

DISCRETE EVENT SIMULATION FOR TEACHING IN CONTROL SYSTEMS

Leonardo das Dores Cardoso

Instituto Federal Fluminense
273 Dr. Siqueira St, Pq. Dom Bosco
Campos dos Goytacazes, RJ, 28030-130,
BRAZIL

João José de Assis Rangel

Universidade Candido Mendes
100 Anita Pessanha St, Pq. São Caetano
Campos dos Goytacazes, RJ, 28030-335,
BRAZIL

Ariel Carvalho Nascimento

Instituto Federal Fluminense
273 Dr. Siqueira St, Pq. Dom Bosco
Campos dos Goytacazes, RJ, 28030-130,
BRAZIL

Quézia Manuela Gonçalves Laurindo
Jhonathan Correa Camacho

Universidade Candido Mendes
100 Anita Pessanha St, Pq. São Caetano
Campos dos Goytacazes, RJ, 28030-335,
BRAZIL

ABSTRACT

The objective of this work is to demonstrate the use of discrete event simulation to be employed as a didactic resource in automatic control systems classes. The application was made with the free and open-source software Ururau integrated to programmable logic controllers of manufacturing didactic stations. The result produced an environment for learning (simulation model and controllers). In this environment, the student can experience different practical situations with the simulation model and have direct contact with the real equipment.

1 INTRODUCTION

In recent work, Cardoso, Rangel and Bastos (2013) presented a hybrid environment for control system testing. The hybrid environment employed a Discrete Event Simulation (DES) model integrated to didactic cells of manufacturing processes used in training of students attending courses of industrial automation. The authors aimed to study the mechanism of integration between the DES model and the manufacturing cells so that it became clear for the Programmable Logic Controller (PLC) of the process to distinguish the real part from the virtual part (simulation model). However, after testing, the authors noted, among the results, the possibility to build different simulation models and connect them to didactic stations of manufacturing. Such fact drew attention because it could be thus created more alternatives and, consequently, a greater variety of practical situations in the training environment of the lab. That is, the student could have direct contact with the real equipment as well as be placed under other operating conditions typical of control systems with simulation models integrated with didactic stations. Another issue to evaluate is in the works of King and Harrison (2013) and Rangel *et al.* (2013). The first presents the Free and Open-Source Software (FOSS) called JaamSim. This software, according to the authors, is the first simulation software of open code in Java and has a number of facilities for the development of simulation models.

On the other hand, the second paper, authored by Rangel *et al.* (2013), introduces the FOSS named Ururau, which was applied to a simulation model to assess the gas emissions of carbon monoxide from a fleet of trucks in a logistics system. What is highlighted here is the emergence of FOSS to be applied to the construction of DES. That is, within this perspective, such software could also be applied, in the future, as didactic resources in the area of education providing practical lessons, mainly in technical schools, and furnishing lower costs for training environments.

Thus, the objective of this work is to demonstrate the application of a DES model built with FOSS to be used as a didactic resource for classes on automatic control systems. The paper demonstrates the application through a practical example where some typical situations of a class on PLC programming are presented. The main advantage of using a FOSS is that schools can adopt an interactive learning environment at lower costs.

2 FOSS: JAAMSIM AND URURAU

In Swain's paper (2007), the author presents a descriptive and well detailed list on more than fifty commercial software that can be used in the development of DES models. There are many benefits in using commercial software of simulation. We do not want to raise any questions regarding these advantages or even disadvantages concerning the commercial or free software, just highlight that there is still much to do regarding FOSS, since the citations found about it are scarce.

From this perspective, the main characteristics of two FOSS, Ururau and JaamSim, that can be used in the construction of DES will be summarized here.

Ururau is a software used to elaborate DES models, based on Java Simulation Library (Rossetti, 2008). The Ururau simulator is open source (GPL - General Public License) multiplatform software, which allows the user to develop his models in a graphical interface or implement them directly in the source code. The use of the Ururau permits to understand the source code and the internal structure of an environment of discrete event simulation, adding specific algorithms to the source code of simulation models or, even, adding new features to the simulator.

For the development and use of software Ururau, the reader is referred to <https://bitbucket.org/ururau>. After downloading the zip file, he/she must unzip it into a directory and run `ururau.jar`. It is necessary to have installed, on the computer, Environment Java Runtime (JRE) version 6.0 or higher, which can be obtained at <http://java.com/getjava>.

To build the model using the source code, it is required a development environment - Integrated Development Environment (IDE), preferably NetBeans 7.0, which can be obtained at <http://www.netbeans.org>.

The Ururau environment is internally composed by one or more process commands, which extend functionalities of the JSL. At the own address in the software, the file README.TXT, which presents an explanation on how to operate each command, can also be downloaded.

JaamSim is a FOSS developed by Ausenco, a global engineering company. JaamSim includes a modern graphical user interface (GUI), which is equivalent to those provided by commercial off-the-shelf simulation software. The software is written in the Java programming language and can be downloaded from Github.

The executable can be copied directly to the user's computer. No special installation program is required. JaamSim runs on most modern computers that support OpenGL graphics version 3.0 or later, including laptop computers with Intel Core i5 and i7 series processor, which rely on integrated graphics (Second Generation, "Sandybridge", and later versions).

The GUI for JaamSim provides all the necessary tools for model building:

- Control Panel – the main interface to JaamSim that controls the execution of models and provides access to the other GUI components
- Views – one or more windows showing 3D views of the model universe
- Model Builder – tool for dragging and dropping model components

- Object Selector – provides access to each object in the model
- Input Editor – displays the inputs to the selected object and permits editing
- Output Viewer – displays the present value for the outputs from the selected object
- Property Viewer – debugging tool that displays all the internal properties for the selected object

The software chosen for the implementation of this work was Ururau. The selection criterion was only based on the fact that the software includes a component for communication with PLCs easy to use.

3 COMMUNICATION OF URURAU WITH PLC

The access can be accomplished through the Tools menu and option OPC (OLE for Process Control) Communication. OPC is a software interface standard that allows Windows programs to communicate with industrial hardware devices.

It is necessary to create a Windows user account with access permissions used for DCOM components of the OPC server chosen. The communication will be successful when, clicking ‘Connect’, the text box lists the predefined tags in the OPC server selected as shown in Figure 1.

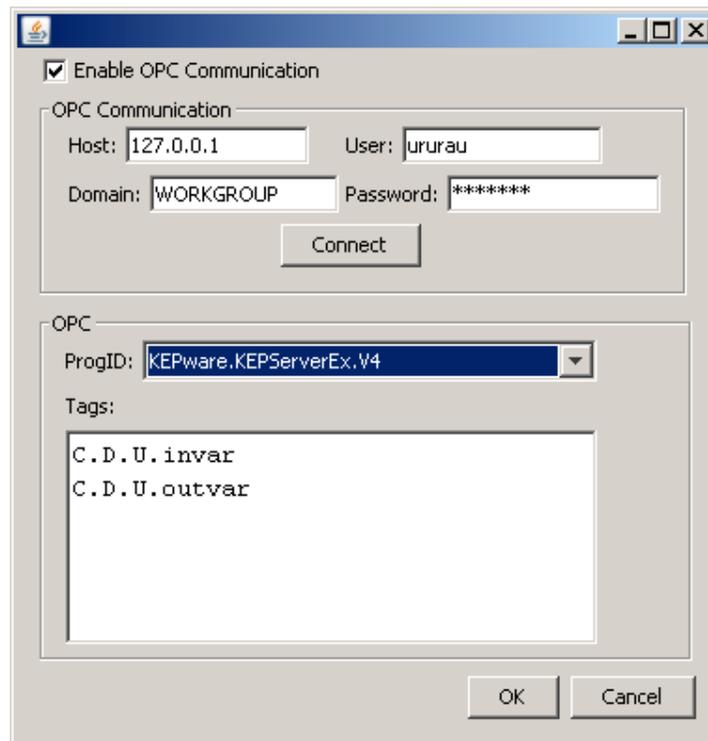


Figure 1: OPC Communication of the Ururau.

4 SCRIPT FOR PRACTICE LESSONS WITH THE PROPOSED ENVIRONMENT

The work of Chiou *et al.* (2011) was used as the basis to prepare a test script for classes in industrial automation technical courses. In this work, the authors also developed a learning environment composed of a DES model integrated to a PLC. The corresponding environment was used as a didactic support tool for teaching modeling methods and verification of automated processes. With that, students were able to develop their own virtual environment to test a program to be adopted in a control system.

Thus, we developed a teaching proposal with the environment formed by the DES model in Ururau and an automatic control system with 8 steps, organized as follows:

- Presentation of a control system with a possible operational error;
- discussion with the class about possible alternatives to solve the corresponding error;
- solution to the problem caused by the error;
- construction of a control system;
- development of a DES model to represent the physical part of the new system using the proposed solution;
- test of the control system employing the DES model;
- test of the control system applying the real physical system;
- operation of the control system in the real physical system.

All steps can be performed in an automation or robotics laboratory, equipped with PLC and some mechanism involving automatic control systems.

5 DEMONSTRATION OF THE USE OF THE LEARNING ENVIRONMENT PROPOSED

This item presents the application of the proposed environment in a laboratory used by classes of industrial automation.

The learning environment was built with four didactic stations of manufacturing systems, manufactured by the company Sintered Metal Corporation (SMC). Each station has a PLC (Omron), model CP1H, and uses programming software CX-One of the same PLC manufacturer. The simulation model was developed in Ururau software version 0.5. The simulations were performed on a computer with the following configuration: Intel Core 2 Duo processor E7600 of 3.06 Ghz, 4Gb RAM memory, video card GForce 8400GS 1774Mb and Windows 7 Professional operating system.

The methodology, by Banks (2010), was followed to prepare the simulation model, according to the next steps: Formulation and analysis of the problem; construction of the conceptual model; construction of the simulation model; verification and validation; and experimentation. It was used, in addition, the methodology proposed by Sargent (2013) for verification and validation of the model.

5.1 Step 1

The script of the class starts with the presentation of the existing system, composed by Stations 1, 3 and 4, shown in Figure 2. Station 2 is not yet described, due to being part of the process related to the error, which will be the subject of student's learning. Arrow A represents the input of parts and B, the output.

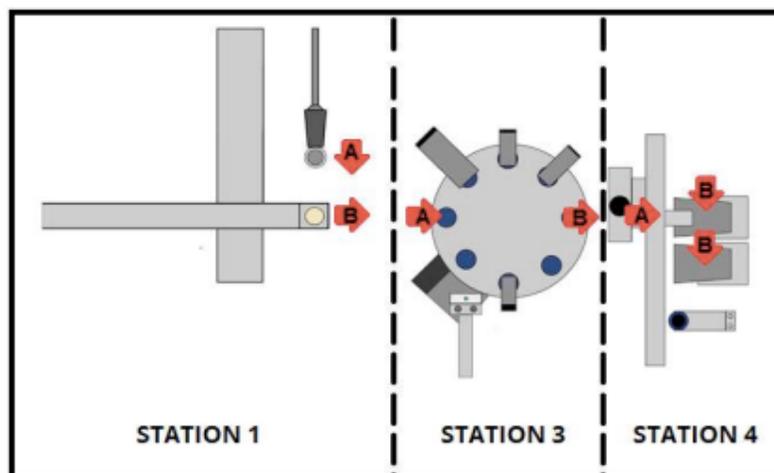


Figure 2: Station 1 (Supply), Station 3 (Process) and Station 4 (Storage).

The first Station, called Supply, is responsible for providing inputs to other stations. It is composed of a movement system and a supplier of parts. The movement system has two associated linear actuators that allow both horizontal and vertical displacements toward the next Station.

The parts sent by Station 1 are deposited on the rotary table of Station 3 and displaced up to the actuator of hole inspection. If the part in question has any hole, the actuator moves until it reaches the end of the path, activating the sensor. In this case, the piece is moved up to the grinding in order to suffer a process of polishing and, then, be moved to the position of delivery for Station 4.

The parts not drilled are considered defective, being discarded.

5.2 Step 2

It is discussed with the students a way to solve a hypothetical error. The solution for this issue is conditioned to the creation and insertion of another Station to the system originally presented in Figure 2 (in this case, Station 2). This Station should allow small parts to be deposited directly in Station 3.

5.3 Step 3

Figure 3 shows an illustration of Station 2. This Station must operate together with other Stations at the end of the task. The graphic animation of Station 2 was elaborated applying the Wizcon software.

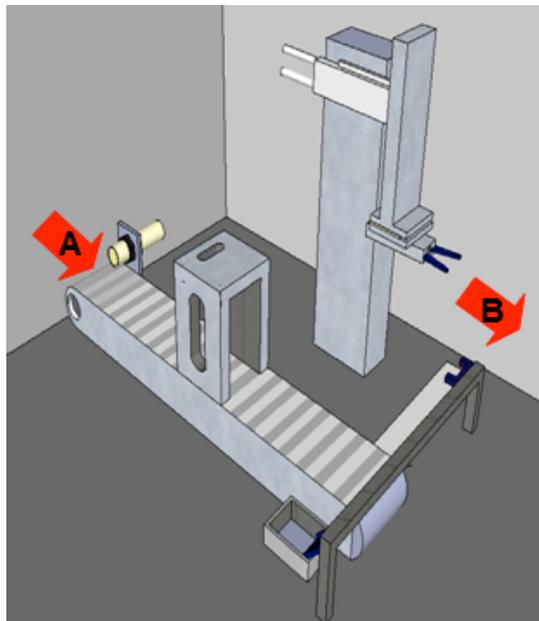


Figure 3: Transport and Separation Station.

5.4 Step 4

After the presentation of the Station, it should be discussed the logic of the system control. The first step consists of the survey of the I/O, which will be used in that logic. Table 1 describes the elements of the system.

Table 1: I/O elements.

ESTAÇÃO	
ADDRESS	TAG's
0.00	X_0
0.01	X_INT
0.02	X_1
0.03	Y_0
0.04	Y_1
0.05	Z_0
0.06	Z_1
0.07	OPEN_CLAW
0.08	PART_BEGINNING_CONVEYOR
0.10	BIG_PART
0.11	GUIDE_PART
1.00	START
100.00	ADVANCE_X
100.01	RETREAT_X
100.02	ADVANCE_Y
100.03	ADVANCE_Z
100.04	CLOSE_CLAW
101.00	START_CONVEYOR
101.01	L_GREEN
101.02	L_YELLOW

5.5 Step 5

Figure 4 shows the structure of the proposed learning environment with the simulation model and the control system. The computer on the right is with the logic of the control system at Ladder diagram. The computer on the left is with the simulation model of Station 2.

Figure 5 shows, in detail, the simulation model in Ururau. This simulation model can also be seen on the right part of the screen, to the left of Figure 4. On the left part of the same screen, to the left of Figure 4, the animation of the Station 2, which is detailed in Figure 3, can be seen.



Figure 4: Test environment of the control system with the simulation model. Computer on the left: simulation model in Ururau; computer on the right: Ladder diagram of the control system.

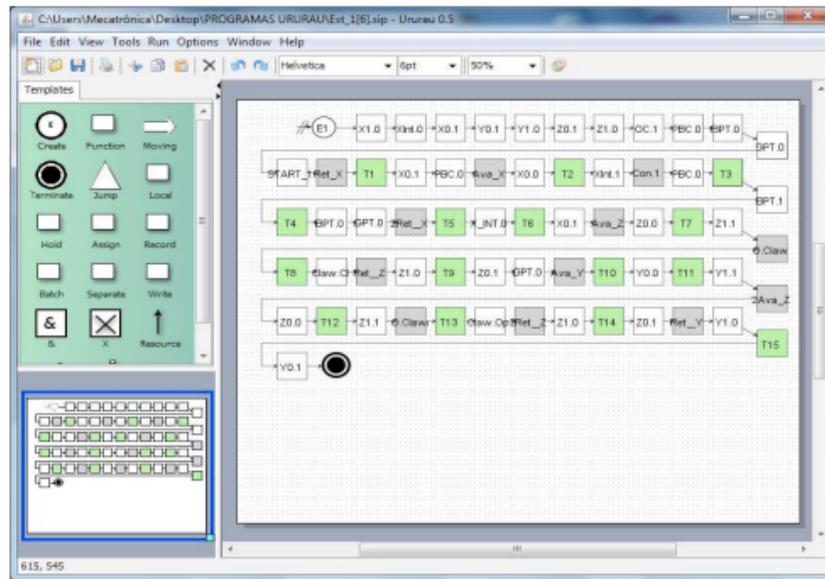


Figure 5: Simulation Model in the Ururau software.

Figure 6 presents the Ladder diagram of the logic to be programmed in the PLC of the control system that operates in Station 2. Similarly, it can also be seen on the monitor, to the right of the same Figure 4, the logic of the control system, which is detailed in Figure 6.

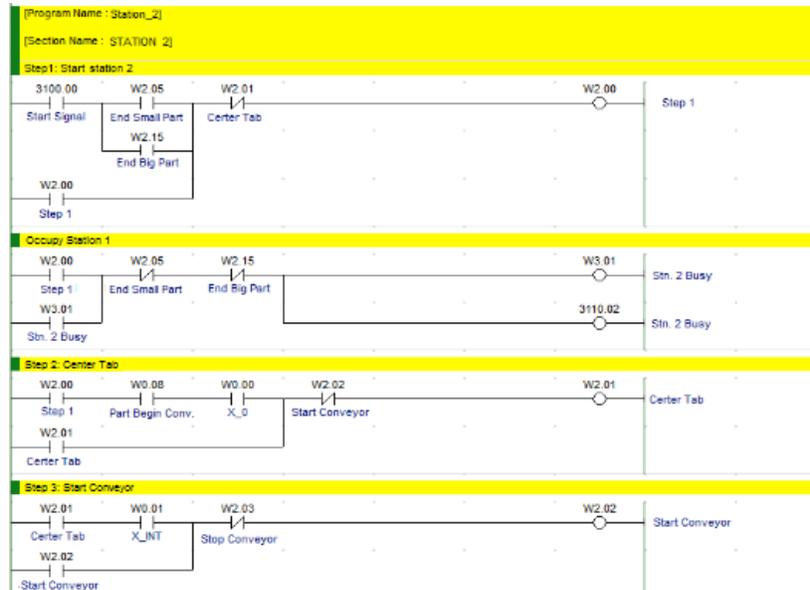


Figure 6: Ladder diagram with the logic of the control system of Station 2.

5.6 Step 6

Similarly, Figure 7 shows the structure of the learning environment proposed, with the real Station 2 and the control system. Note that the computer on the right with the control system remains the same way as shown in Figure 4. Also, the computer on the left, which was previously (Figure 4) with the simulation model of Station 2, was replaced by Station 2 real.

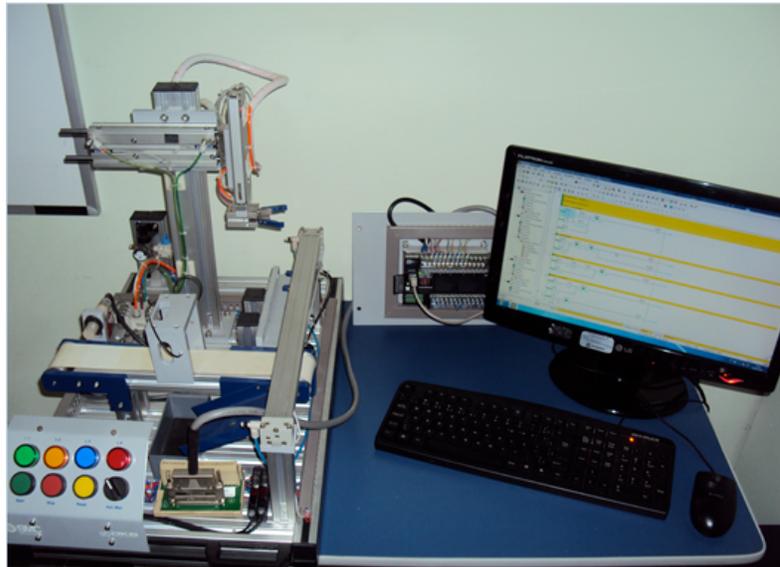


Figure 7: Station 2 on the left, computer on the right with the control system.

Thus, the way it is presented, there is the possibility of combining simulation models with FOSS Ururau integrated to control systems. Notice that the control system runs on PLC served both in the simulation model of the real Station as in the simulation model of the referred Station. In other words, one can idealize different control logic and different simulation models and combine them generating a large number of training opportunities for students. It can be drawn attention to the fact that there is, also, the option to integrate or not the simulation model and the didactic Stations of manufacturing in order to have a hybrid system, as described in Figure 2. Thus, Station 2, real or virtual, may occupy the place and, this way, form a single operational set.

The simultaneous combination of didactic stations and simulation models can further extend the possibilities for different operating situations and control systems for the practice of the students. Also, another issue that can draw attention to is the fact, as it was shown, that students have the possibility of using the real station and, thus, handle the equipment itself. That is important because the contact with the real system allows the improvement of other experiences that students need to develop to facilitate subsequent performance when in an industry. The use of simulation models in FOSS environments enables cost reduction for the application in the laboratories of schools.

6 CONCLUDING REMARKS

The script for an industrial automation class presented in this study demonstrated the possibility of applying FOSS Ururau as a didactic resource at low cost. The use of simulation models integrated to manufacturing cells formed a hybrid environment. Therefore, the student was able to perform several tests combining the options available by the didactic cells and DES models. It has been demonstrated that there is the possibility of expanding the options for the different types of practical situations for learning students. Moreover, by the fact of being in a hybrid environment (manufacturing cell and DES model), there is the possibility of expanding the learning environment while providing the learner a chance to be placed towards the real

equipment. Although this work has focused on the use of Ururau software, the methodology can easily be applied to another simulation software that has a channel of communication with PLC.

ACKNOWLEDGMENTS

The authors would like to thank the National Council of Technological and Scientific Development - CNPq and the Foundation for Research Support of Rio de Janeiro - FAPERJ for financial support for this research. They also thank Maria Marta Garcia for the suggestions on the text in English.

REFERENCES

- Banks, J., J. S. Carson, B. L. Nelson, and D. M. Nicol. 2010. “*Discrete-Event System Simulation*”. 5th ed. Upper Saddle River, New Jersey: Prentice-Hall, Inc.
- Cardoso, L. D., J. J. A. Rangel, and P. J. T. Bastos. 2013. “Discrete Event Simulation for Integrated Design in the Production and Commissioning of Manufacturing Systems”. In: *Proceedings of the Winter Simulation Conference*, edited by R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl.
- Chiou, R., Y. Kwon, R. Kizirian, M. Dordai, and A. D. Bret. 2011. “Modeling and Experimental Verification of PLC Code in a Robotics and Mechatronics Course”. In: *ASEE Annual Conference & Exposition. 118th ed.* Vancouver, B. C. Canada.
- King, D.H., and H. S. Harrison. 2013. “Open-Source Simulation Software JAAMSIM”. In: *Proceedings of the Winter Simulation Conference*, edited by R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl.
- Rangel, J. J. A., G. L. Oliveira, T. A. Peixoto, L. D. Cardoso, I. O. Matias, E. Shimoda. 2013. “Simulation Model in a Free and Open-Source Software for Carbon Monoxide Emissions Analysis”. In *Proceedings of the Winter Simulation Conference*. edited by R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl.
- Rosseti, M. D. 2008. “Java Simulation Library (JSL): an open-source object-oriented library for discrete-event simulation in Java”. *Int. J. Simulation and Process Modeling*, Vol. 4, No. 1: 69-87.
- Sargent, R. G. 2013. “Verification and Validation of Simulation Models”. *Journal of Simulation*, Vol.7:12-24.
- Swain, J. J. 2007. “Discrete Event Simulation Software: New Frontiers in Simulation”, *OR/MS Today - INFORMS*, Vol. 34, No. 5: 32-43.

AUTHOR BIOGRAPHIES

LEONARDO DAS DORES CARDOSO is a teacher at Federal Fluminense Institute (IFF). He is an Industrial Engineering degree from Candido Mendes University (UCAM-Campos) in Campos dos Goytacazes, RJ - Brazil, in 2011, and also a Control and Automation Engineering degree from Federal Fluminense Institute (IFF-Campos) in Campos dos Goytacazes, RJ - Brazil, in 2012. He received a Master degree in Industrial Engineering at UCAM-Campos in 2013. His e-mail is leonardo.cardoso@iff.edu.br.

JOÃO JOSÉ DE ASSIS RANGEL is a professor at Federal Fluminense Institute (IFF) and Candido Mendes University (UCAM-Campos) in Campos dos Goytacazes, RJ - Brazil. He received a doctoral degree in Materials Engineering at UENF in 1998. His research interests include Simulation on Logistics and Manufacturing, Simulation on Super Hard Materials Process and Simulation Education. His e-mail address is joao@ucam-campos.br.

ARIEL CARVALHO NASCIMENTO is a Control and Automation Engineering student at Federal Fluminense Institute (IFF-Campos) in Campos dos Goytacazes, RJ – Brazil. His e-mail address is ariel.arilius@gmail.com.

Cardoso, Rangel, Nascimento, Laurindo and Camacho

QUÉZIA MANUELA GONÇALVES LAURINDO is a Mechanical Engineering student at Candido Mendes University (UCAM-Campos). Her e-mail address is manulaurindo92@hotmail.com.

JHONATHAN CORREA CAMACHO is a Computer Science student at Candido Mendes University (UCAM-Campos). His e-mail address is jhonathancorreacamacho@gmail.com.