A Virtual PLC Environment for Assisting Automation Teaching and Learning

https://doi.org/10.3991/ijim.v11i5.7066

L. Brito Palma (✉), V. Brito, J. Rosas and P. Gil
Universidade NOVA de Lisboa, Lisbon, Portugal
lbp@fct.unl.pt

Abstract—In this paper, a virtual PLC environment aiming at assisting automation teaching, learning and e-learning is proposed and discussed. The main contributions are a virtual PLC environment propped on a local and remote applications, along with the comparison of two teaching and learning methodologies, one local, while the other offers a remote interaction with the system. The proposed framework fits in the emerging technologies on the Internet of Everything, having the potential of promoting the integration and communication of virtual and real systems.

Keywords—Remote laboratory, automation, control, e-learning, programmable logic controller, structured text language.

1 Introduction

The current wave towards full globalization, along with the evolution of the web standards have fostered the development of remote teaching and learning/e-learning methodologies, which are paramount in these days. Remote applications allow students and researchers to have access from distant locations to essential laboratory environments, without attending practical classes or physical laboratories on-Campus, which is undoubtedly a great advantage to students or even researchers with lack of time or resources [1], [2].

This work presents a remote laboratory framework focused on automation courses, which provides users with a simulation environment for dynamic models (ARX) connected to a virtual Programable Logic Controller (PLC) and its programming based on Structured Text (ST).

2 State of the art

2.1 Remote Laboratories

The muddle surrounding remote labs and their definition has been passing on through the years. The nomenclature and concepts are extremely straightened in [3] and defined by Table 1.

Table 1. Laboratory environments

Nature of equipment **User access**	**Local**	**Distant**
Physical (Real)	Hands-on lab	Remote lab
Virtual (Modelled)	Virtual lab	Distributed virtual lab

Experimentation in a physical laboratory is expensive due to the need of a number of similar equipment items, and in addition it is costly to maintain. Simulators, virtual laboratories and remote laboratories provide a solution to some of these problems, while being available most of the time, and avoiding possible dangerous failure events. The remote lab concept also provides a tool for shifting towards a student-centric teaching approach, as recommended in the Bologna Process [4].

One of the resulting impacts of information and communication technologies has been the establishment of large educational networks, such as the European Schoolnet, MIT OpenCourseWare, iLab, and PROLEARN, along with many other individual and collaborative distance laboratory efforts all over the world [5].

The Internet and the World Wide Web allow an effective support for collaborative work, either on-line or in real-time. Most universities and some other institutions have their own e-learning environments ready to be remotely accessed through the Web, mainly on subject areas of Electronics, Mechanics, Physics, Chemistry and Control [4]-[9].

On the World Wide Web, few Web-Labs are found in the area of Automation, particularly in the field of remote PLC programming. In this section, two different types of solutions are described and referenced, to exemplify, namely one using remote desktop (RDP) services, and others lacking RDP services. Most of the existing laboratories are based on Remote Desktop Terminal Services using proprietary software from PLC manufacturers, and usually using a Virtual Private Network (VPN) for security reasons. One of the published works, using a graphical desktop with remote access via the Remote Desktop Protocol (RDP) supported by a Remote Desktop (Terminal Services) client [10], allows a user to remotely work with the Ladder (LD) programming language on a PLC Micrologix 1200. Another work [11], also using RDP services, allows users to remotely program a PLC M340 using five IEC programming languages (LD, IL, SFC, FBD, ST). Without relying on RDP services, a solution should have the PLC programming software on both sides (client side and server side), along with other software packages such as ".net framework", "vb.net", etc, as mention in [12]. Another solution without RDP services is described in [13], using Java language and LabView.

For years, WebLabs have been developed and maintained by university teachers/instructors and researchers, mainly physicists or control/electronic engineers. In the development of WebLabs not only technical issues should be taken into account, but also security, accessibility, usability, remote collaboration and multilingualism; so, the informatics and social aspects are both very important.

A comparison of different approaches for remote laboratories, with different software technologies on the server side and on the client side (HTML, Javascript, AJAX, PHP, Java, Matlab, Labview, Webservices, etc), can be found for instance in [11].

There are significant contributions regarding simulation for Industrial Automation. In the past, many simulators, virtual labs and remote Labs were developed around the world, mainly in universities [5], [14]–[17]. The five PLC programming languages according to IEC 61131-3 are [18]: Ladder (LD), Instruction List (IL), Sequential Function Chart (SFC), Function Block Diagram (FBD) and Structured Text (ST). There are a fair number of modelling and simulation tools for these languages. However, few software programs exist for the generation and simulation of the ST language. One of them is the Simulink PLC Coder, which generates ST code from Simulink models, state-flow charts and embedded Matlab functions. Others are proprietary PLC programming software provided by PLC manufacturers, such as Schneider Electric, Siemens, Omron, Rockwell, just to name a few.

In this paper, the main focus is on the Structured Text Programming language, where two PLC approaches are discussed: one based on a local application (client-side) and the other relying on a remote application (server-side).

3 Proposed Virtual PLC Framework

Two applications compose the proposed virtual PLC framework, namely a local environment and remote application.

The client's local application high-level architecture consists of the elements shown in Figure 2 within the Application container. The ST Parser is where the code is interpreted and translated into C-code via a Dynamic-Link Library (DLL), the PLC chart allows to plot inputs and outputs from the PLC, and the System Graphical User Interface (GUI) is where the ARX dynamic model is defined.

The Web application takes advantage of the local application running the ST Parser implemented in it. The Web interface will be equipped with an authentication manager to allow only one user at a time, since only one local app is available and running on the server. It will save information about the users' code in order to better understand the most common errors among students and also manages the input and output ports.

The Web programming follows the Model View Controller (MVC) architecture. The goal of the MVC design pattern is to isolate the user interface from the original data represented by the user interface.

In MVC the Views display the information to a user, the Controller is where the users' interactions are processed, and the Model is where the information displayed in the View is contained and the logic that changes that information responding to the user interaction [19], [20]. Figure 1 shows this architecture.

With respect to the client local environment, this application allows simulating discrete-time ARX dynamic models, visualizing both analogue and digital input and output signals and simulating a virtual PLC (TSX3721 Schneider Electric) with support to the ST programming language. All the interface is programed in C#.

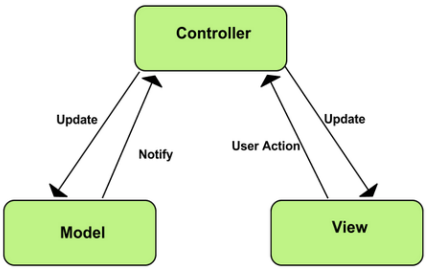


Fig. 1. MVC architecture.

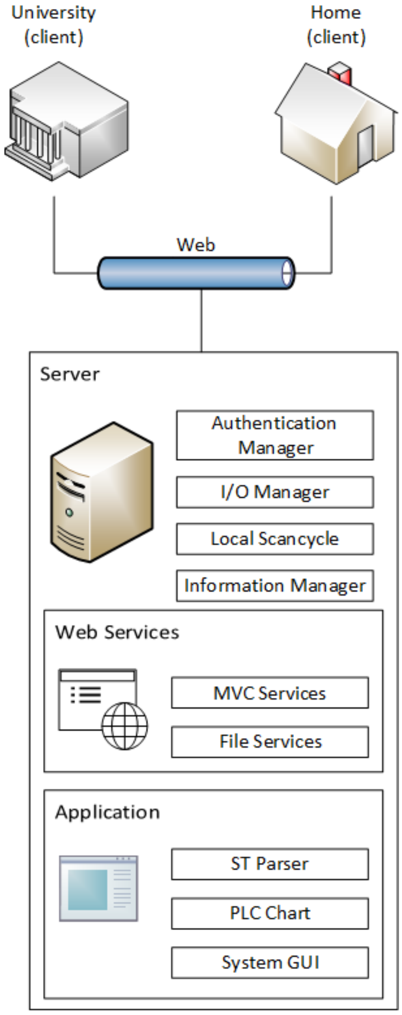
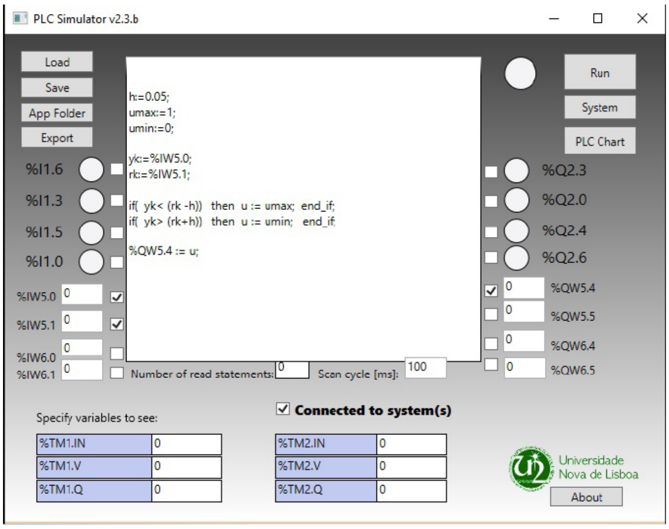


Fig. 2. High-level architecture.

The virtual PLC allows the implementation of supervision tasks, enabling as well to record input and output data. The ST language supports the basic identifiers (constants, variables, I/O addresses, memories, timers, etc.), expressions (basic operators, logical operators, transitions detection, etc.) and statements (set/reset memories, start/stop timers, output assignments, if-then-else, etc.). The PLC GUI is shown in Figure 3 [21], [22].

As a PLC, the routine of the application is running repeatedly, commonly known as scan cycle. Figure 4 illustrates the scan cycle, where inputs are read into the PLC memory, the logical statements are executed and the results are provided to the outputs.

The architecture of the simulator is presented in Figure 5 where all the components are connected and interoperable.



```
h:=0.05;
umax:=1;
umin:=0;

yk:=%IW5.0;
rk:=%IW5.1;

if(  yk< (rk -h))   then  u := umax;  end_if;
if(  yk> (rk+h))   then  u := umin;   end_if;

%QW5.4 := u;
```

%IW5.0 0 ☑
%IW5.1 0 ☑
%IW6.0 0 □
%IW6.1 0 □

Number of read statements: 0   Scan cycle [ms]: 100

☑ %QW5.4 0
□ %QW5.5 0
□ %QW6.4 0
□ %QW6.5 0

☑ **Connected to system(s)**

Specify variables to see:

%TM1.IN	0	%TM2.IN	0
%TM1.V	0	%TM2.V	0
%TM1.Q	0	%TM2.Q	0

Fig. 3. Local virtual PLC framework graphical user interface.

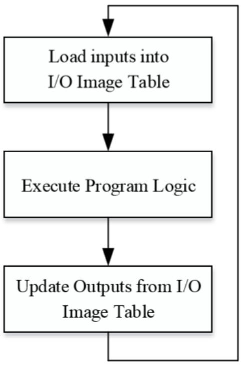

Load inputs into I/O Image Table
Execute Program Logic
Update Outputs from I/O Image Table

Fig. 4. Application scan cycle [21].

The remote application is based on a Web client/server architecture, and presents similar functionalities to the local application, although it is still in a development stage and some of the functionalities are not ready yet, the System GUI and the login routine. This virtual PLC environment is an appropriate tool not only for local/remote teaching, but also for local/remote self-learning. Its main advantages are the ability to save feedback information collected from remote users' simulations and experiments, and can also be used as an interface with real remote processes and systems.

The implementation of the Web interface calls upon a MVC structure mentioned ealier and uses Html, Css and Bootstrap for the GUI, and its functionalities are implemented using JavaScript, JQuery and AJAX. Figure 6 shows the web interface with the PLC code input and the I/O ports.

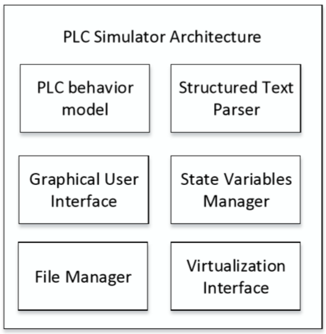
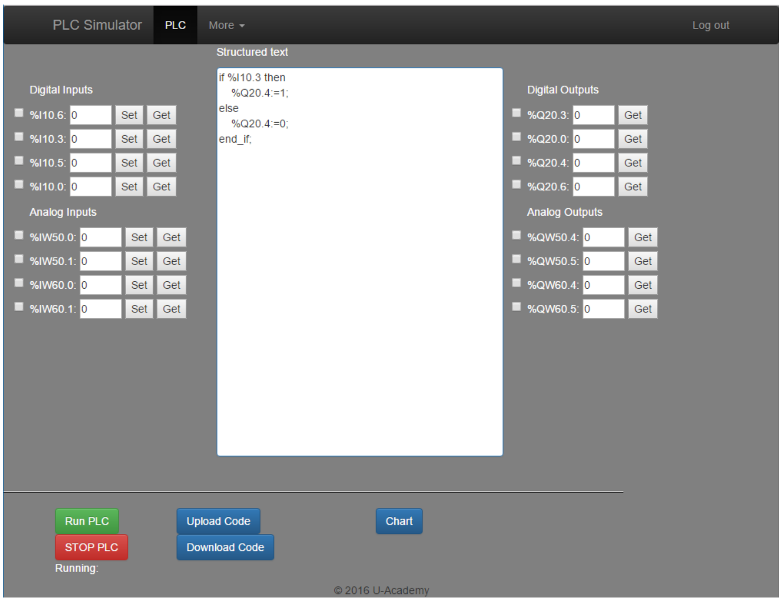


Fig. 5. Local PLC application architecture.



Structured text

```
if %I10.3 then
   %Q20.4:=1;
else
   %Q20.4:=0;
end_if;
```

Digital Inputs

□ %I10.6: 0 Set Get
□ %I10.3: 0 Set Get
□ %I10.5: 0 Set Get
□ %I10.0: 0 Set Get

Analog Inputs

□ %IW50.0: 0 Set Get
□ %IW50.1: 0 Set Get
□ %IW60.0: 0 Set Get
□ %IW60.1: 0 Set Get

Digital Outputs

□ %Q20.3: 0 Get
□ %Q20.0: 0 Get
□ %Q20.4: 0 Get
□ %Q20.6: 0 Get

Analog Outputs

□ %QW50.4: 0 Get
□ %QW50.5: 0 Get
□ %QW60.4: 0 Get
□ %QW60.5: 0 Get

Running:

© 2016 U-Academy

Fig. 6. Virtual PLC web interface framework.

The implemented virtual PLC environment not only allows users to learn Automation concepts and ST programming, but also intuitively enables users to better understand dynamic systems and control theory/algorithms in open or closed loop *modus operandi*. For instance, Figure 7 illustrates a supervision/control architecture, which can be implemented in the virtual PLC [22].

The next section presents these concepts subsequently implemented and tested, where the virtual PLC environment and a control structure are simulated on a dynamic system.

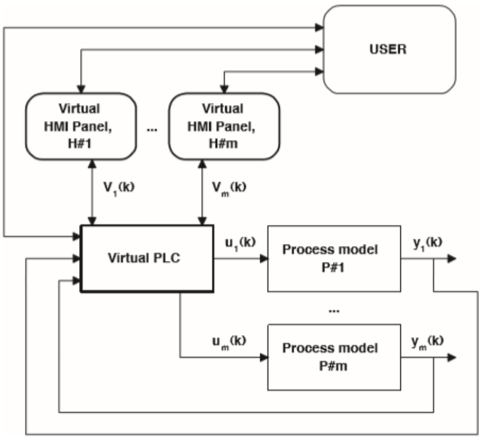
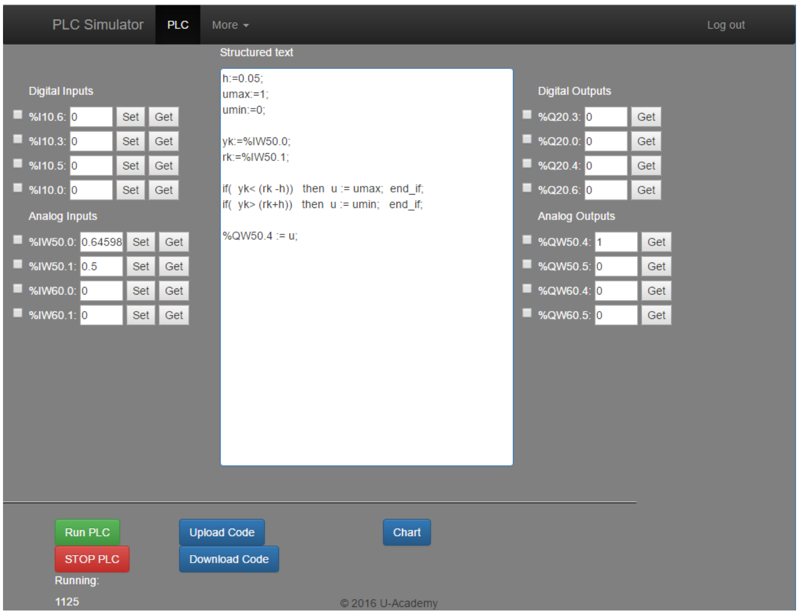


Fig. 7. Example of supervision/control architecture of the PLC simulator.



```
h:=0.05;
umax:=1;
umin:=0;

yk:=%IW50.0;
rk:=%IW50.1;

if(  yk< (rk -h))   then  u := umax;  end_if;
if(  yk> (rk+h))   then  u := umin;   end_if;

%QW50.4 := u;
```

□ %I10.6: 0
□ %I10.3: 0
□ %I10.5: 0
□ %I10.0: 0
□ %IW50.0: 0.64598
□ %IW50.1: 0.5
□ %IW60.0: 0
□ %IW60.1: 0
□ %Q20.3: 0
□ %Q20.0: 0
□ %Q20.4: 0
□ %Q20.6: 0
□ %QW50.4: 1
□ %QW50.5: 0
□ %QW60.4: 0
□ %QW60.5: 0

Running:
1125

Fig. 8. Web PLC application simulation.

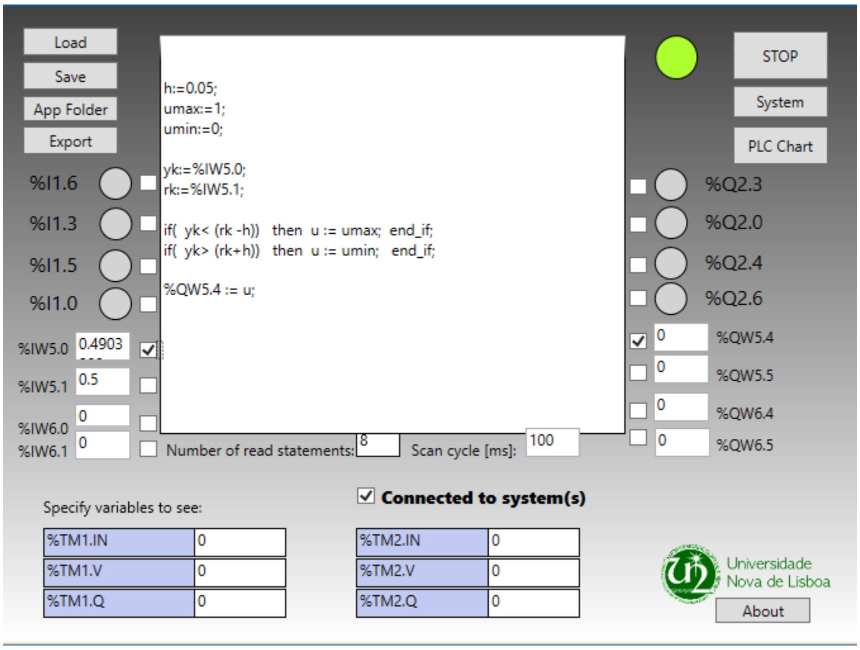
4 Simulation Results

4.1 Local Application Simulation Results

The simulations carried out on the local application consisted in designing a simple ST code with a relay controller to control an ARX(2,2,1) model. The sampling time is 100ms.

Error! Reference source not found. Fig. 10Fig. 11show the simulation of the virtual PLC, as well as its components, namely the PLC chart and the system simulation. In the virtual PLC GUI running, one can observe the analogue ports changing according to the relay controller, which was implemented, as well as the number of read statements and the scan cycle loop time. The PLC chart (Fig. 10) allows the user to plot selected I/O ports of the virtual PLC. The system simulation (Fig. 11) contains the generic ARX(Na, Nb, Nk) parameters (following equation 1) with a maximum architecture size of (3, 2, 1) and the I/O ports of the system. The interface also allows a user to start and stop the simulation.

$$\begin{aligned} & y(k) = a_1 y(k-1) - \cdots - \\ & -a_{Na} y(k-Na) + b_1 u(k-Nk) + \cdots + \\ & +b_{Nb} u(k-Nk-Nb+1) \end{aligned} \tag{1}$$



```
h:=0.05;
umax:=1;
umin:=0;

yk:=%IW5.0;
rk:=%IW5.1;

if( yk< (rk -h))  then  u := umax;  end_if;
if( yk> (rk+h))  then  u := umin;   end_if;

%QW5.4 := u;
```

%IW5.0 0.4903 ☑
%IW5.1 0.5 □
%IW6.0 0 □
%IW6.1 0 □

☑ 0 %QW5.4
□ 0 %QW5.5
□ 0 %QW6.4
□ 0 %QW6.5

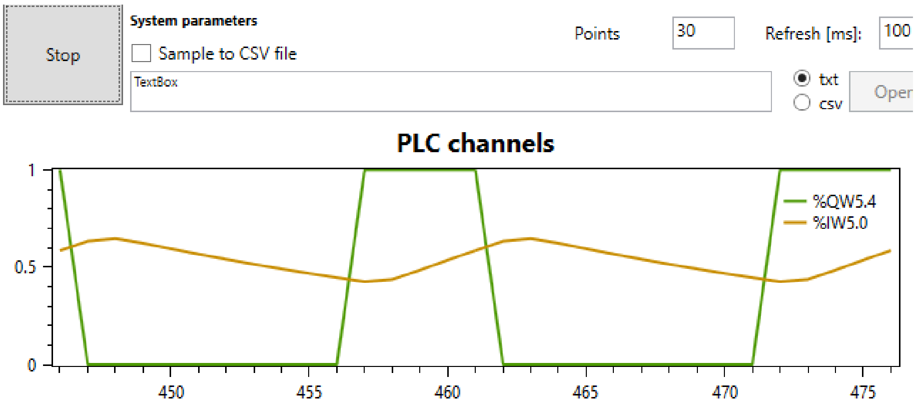
Number of read statements: 8 Scan cycle [ms]: 100

☑ **Connected to system(s)**

Specify variables to see:

%TM1.IN	0	%TM2.IN	0
%TM1.V	0	%TM2.V	0
%TM1.Q	0	%TM2.Q	0

Fig. 9. PLC in Run mode.



System parameters

☐ Sample to CSV file

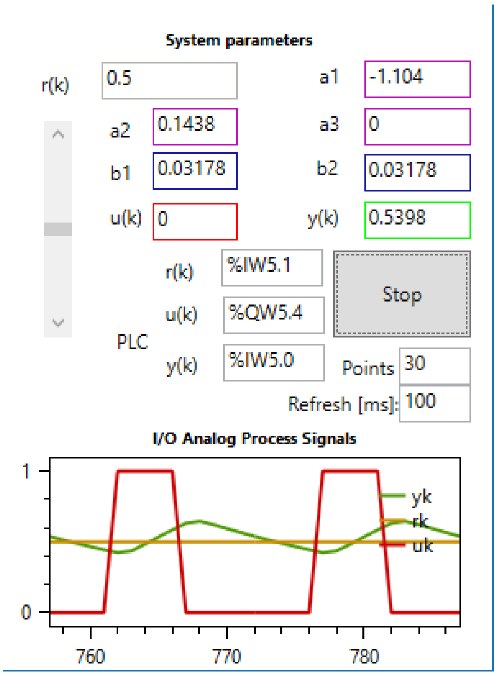
Points: 30

Refresh [ms]: 100

TextBox

● txt ○ csv

Fig. 10. PLC chart showing the output and input of the system.



System parameters

r(k)	0.5	a1	-1.104
a2	0.1438	a3	0
b1	0.03178	b2	0.03178
u(k)	0	y(k)	0.5398

PLC

r(k): %IW5.1

u(k): %QW5.4

y(k): %IW5.0

Points: 30

Refresh [ms]: 100

Fig. 11. PLC system representing the ARX(2,2,1) and the I/O.

4.2 Web Application Simulation Results

The web application was also simulated using the same relay controller programmed in the local simulation, and applied to the same system. A remote user can have access to a similar GUI (Figure 8) with start/stop buttons to manipulate the virtual PLC state, the upload/download code, in order to send or receive code from the local application running in the server, and can observe the I/O ports being manipulated. A user can also observe the system's I/O chart represented in Fig. 12, similarly to the graph presented in the PLC system GUI, by pressing the corresponding button.

The Web interface also incorporates a scan cycle timer, allowing the user to check whether the local application in the server is running the downloaded code.

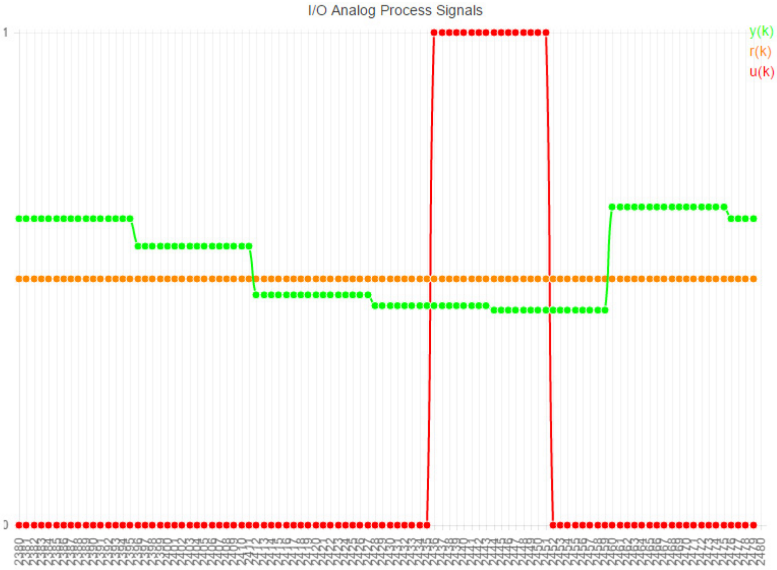


Fig. 12. PLC chart regarding the system simulation.

5 Conclusions

A virtual PLC environment for assisting automation teaching and learning considering discrete-event systems and ARX models, HMI panels, in a multitasking paradigm, was proposed and discussed. Architectures of the remote PLC Simulator (version w2.3) were described, emphasizing the parser for the ST language, and the PLC behaviour model, as well as the e-learning capabilities. The environment allows a user to implement open-loop or closed-loop controllers, while types of controllers can be programmed and tested. Performed simulations show that the PLC simulator works well with different kinds of systems, and has a great potential to be a relevant tool in learning/e-learning/teaching environments, as well as in industry. It is also intended to develop the tool up to a level in which it can be used in professional contexts. Additionally, this tool could be further used for research purposes in the areas of Dynamic Systems, Control and Automation.

Although the work is far from complete, the client-side local application is already in tests, and being included in a lecture program. The Web interface is currently in a test phase and will, in a near future, be available to the scientific and industrial community.

6 Acknowledgment

This work has been supported by Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, by Uninova-CTS research unit, by U-Academy project and by national funds through FCT Fundação para a Ciência e a Tecnologia within the research unit CTS - Centro de Tecnologia e Sistemas (project UID/EEA/00066/2013). The authors would like to thank all the institutions.

7 References

[1] M. Latorre, A. Robles-Gómez, L. Rodríguez, P. Orduña, E. San Cristóbal, A. C. Caminero, L. Tobarra, I. Lequerica, S. Ros, R. Hernández, M. Castro, D. Lopez-de-Ipiña, and J. García-Zubia, "A review of webapp authoring tools for e-learning," *IEEE Glob. Eng. Educ. Conf. EDUCON*, April 2014, pp. 770–777, 2014.

[2] F. Lerro and S. Marchisio, "Preferences and uses of a remote lab from the students' viewpoint," *Proc. 2015 Int. Conf. Interact. Collab. Learn. ICL 2015*, vol. 12, no. 3, pp. 854–857, 2015.

[3] D. Muller and H.-H. Erbe, "Collaborative Remote Laboratories in Engineering Education: Challenges and Visions," *Adv. Remote Lab. e-learning Exp.*, 2007.

[4] C. Samoila, S.G. Cosh and D. Ursutiu, "Competences, Remote Labs and Bologna Process," in *Advances on remote laboratories and e-learning experiences*, 2007, pp. 63–96.

[5] L. F. D. S. Gomes and J. García Zubía, *Advances on remote laboratories and e-learning experiences*, vol. 6, 2007.

[6] J. Garcia-Zubia, D. López-de-Ipiña, and P. Orduña, "Mobile devices and remote labs in engineering education," in *Proceedings - The 8th IEEE International Conference on Advanced Learning Technologies, ICALT 2008*, 2008, pp. 620–622.

[7] M. T. Restivo and M.G. Silva, "Portuguese Universities Sharing Remote Laboratories," *Int. J. Online Eng.*, vol. 5, 16-19, 2009. https://doi.org/10.3991/ijoe.v5s2.1090

[8] C. Ferrater-Simon, L. Molas-Balada, O. Gomis-Bellmunt, N. Lorenzo-Martinez, O. Bayo-Puxan, and R. Villafafila-Robles, "A Remote Laboratory Platform for Electrical Drive Control Using Programmable Logic Controllers," *IEEE Trans. Educ.*, vol. 52, no. 3, pp. 425–435, 2009. https://doi.org/10.1109/TE.2008.930095

[9] M. T. Restivo and A. Cardoso, "Online Experimentation in Education and Training," *Int. J. Eng. Pedagog.*, vol. 4, no. 2, pp. 52–56, 2014. https://doi.org/10.3991/ijep.v4i2.3481

[10] K. C. V. Lasky, D. Liu, S. Murray, "A Remote PLC system for e-Learning," *Proc. ASEE / AAEE 4th Glob. Colloq. Eng. Educ. Aust.*, 2005.

[11] L. F. F. Brito Palma, F. Vieira Coito, A. Gomes Borracha, and J. Francisco Martins, "A Platform to Support Remote Automation and Control Laboratories," *1st Exp. Int. Conf.*, pp. 17–18, 2011.

[12] I. Colak and A. Efe, "Design and implementation of a remote access PLC training set," in *International Aegean Conference on Electrical Machines and Power Electronics and Electromotion, Joint Conference*, 2011, pp. 425–429. https://doi.org/10.1109/ACEMP.2011.6490636

[13] A. Safavi, A. A. Safavi, and P. Veisi, "A remote and virtual PLC laboratory via smartphones," in *4th International Conference on e-Learning and e-Teaching (ICELET 2013)*, 2013, pp. 63–68. https://doi.org/10.1109/ICELET.2013.6681647

[14] L. Gomes and S. Bogosyan, "Current Trends in Remote Laboratories," *Ind. Electron. IEEE Trans.*, vol. 56, no. 12, pp. 4744–4756, 2009.

[15] J. Garcia-Zubia and G. Alves, *Using Remote Labs in Education*. 2011.

[16] J. García-Zubía and O. Dziabenko, *IT Innovative Practices in Secondary Schools: Remote Experiments*. 2013.

[17] R. Langmann, "E-Learning & Doing in Training for Automation Engineers," 2011.

[18] J. Martins, C. Lima, H. Martinez, and A. Grau, "PLC Control and Matlab/Simulink Simulations – A Translation Approach," *Matlab - Model. Program. Simulations*, 2010.

[19] A. Leff and J. Rayfield, "Web-application development using the model/view/controller design pattern," *2001. EDOC'01, Proc. 5th IEEE*, 2001. https://doi.org/10.1109/EDOC.2001.950428

[20] G. Krasner and S. Pope, "A description of the model-view-controller user interface paradigm in the smalltalk-80 system," *J. object oriented*, 1988.

[21] L. B. Palma, J. A. Rosas, J. Pecorelli, and P. S. Gil, "Simulation of structured text language for PLC programming," *exp.at 2015 - 3rd Exp. Int. Conf. Online Exp.*, pp. 296–301, 2015.

[22] L. F. F. Brito Palma, J. A. Rosas, J. Pecorelli, and P. Sousa Gil, "Structured Text Simulator for PLC in Learning Environment," 2015.

8 Authors

L. Brito Palma (corresponding author) received his PhD in Electrical Engineering in 2007 from NOVA University of Lisbon (UNL) - Faculty of Sciences and Technology (FCT), where is currently Professor at the Department of Electrical Engineering, and Researcher at Uninova-CTS Research Institute, Caparica-Lisboa, Portugal. He worked at INETI Institute in the optoelectronics and laser areas, at INESC Institute in the electronics and image processing areas, and in the Physics Department at UNL-FCT in the electronics area. Nowadays, his research interests are in automation, fault detection / diagnosis, intelligent fault tolerant control systems, industrial process control, aeronautical control systems and distributed systems. He has more than 100 publications, in international journals, conference proceedings and book chapters. (e-mail: lbp@fct.unl.pt).

V. Brito received the MSc degree in electrical and computers engineering from NOVA University of Lisbon (UNL) - Faculty of Sciences and Technology (FCT), Portugal in 2016. In 2013 he has worked in the electronics area, for instance the development of audio tube power amplifiers. In 2014 he developed and implemented PID industrial controllers embedded in microcontrollers. In 2015 he participated in a national Siemens contest with a project entitled "Hybrid System of Distributed Automation", regarding distributed automation and fault tolerant control systems, reaching the final stage. He is currently a Researcher at Uninova-CTS Research Institute, Caparica - Lisboa, Portugal, in the areas of dynamical signals and systems, intelligent fault tolerant control systems, aeronautical systems, multi-rotors drones and more recently Internet of Things. (v.brito@campus.fct.unl.pt).

J. Rosas received his PhD in Electrical Engineering in 2010 from NOVA University of Lisbon, Portugal, where is currently Professor at the Department of Electrical Engineering. His research interests are in Real-time Systems, Robotics, Internet of

Things, and Simulation of Distributed Manufacturing systems. He has several publications in international journals, conference proceedings and book chapters. He has been participating as team member in several European Commission funded research projects. (jrosas@fct.unl.pt).

P. Gil received his PhD in Electrical Engineering in 2004 from NOVA University of Lisbon, Portugal, where is currently Professor at the Department of Electrical Engineering. His research interests are in distributed systems, intelligent control systems, fault tolerant control and resilient networked control systems over heterogeneous networks. He has more than 120 publications, in international journals, conference proceedings and book chapters, and is active on program committees of many of the well-established conferences and workshops. Additionally, he is a Fellow of the IEEE Control Systems Society. (psg@fct.unl.pt).

Article submitted 27 April 2017. Published as resubmitted by the authors 05 June 2017.