DOI: 10.48011/asba.v2i1.1366

Tool for Control Systems Development based on IoT (IoTControl)

MSc. Danilo Oliveira Martins*
PhD. Prof. Almir Fernandes**
PhD. Prof. Alexandre Brincalepe Campo***

**IFSP – Instituto Federal de Educação, Ciência e Tecnologia de São Paulo.*
Rua Pedro Vicente, 625 - Canindé, São Paulo - SP, 01109-010 (e-mail:
dmartins.iotcontrol@gmail.com);
***IFSP – Instituto Federal de Educação, Ciência e Tecnologia de São Paulo.*
Rua Pedro Vicente, 625 - Canindé, São Paulo - SP, 01109-010
****IFSP – Instituto Federal de Educação, Ciência e Tecnologia de São Paulo.*
Rua Pedro Vicente, 625 - Canindé, São Paulo - SP, 01109-010 (e-mail:
brinca@ifsp.edu.br)}

Abstract: This work presents the development of a tool for control systems, capable to configure the control parameters without process stop and allow remote access operation based on the architecture of IoT. Perform the control of processes, monitoring and update algorithms, solution embedded into the physical device connected to the control plant (DC motor), equipped with data network management program, allow the configuration of parameters and graphical analysis of the controller in real time. Tools that allow remote control and monitoring to people geographically distant from large educational centers is an advantage, also relevant and essential due to the challenges we face on COVID-19 pandemic.

Resumo: Este trabalho apresenta o desenvolvimento de uma ferramenta para sistemas de controle, capaz de configurar parâmetros de controle sem parar o processo e permite operação com acesso remoto baseado na arquitetura de internet das coisas (IoT). Realizar controle de processos, monitoramento e atualização de algoritmos, solução embarcada em um dispositivo físico conectado a uma planta de controle (motor CC), equipado com um programa de gerenciamento de dados em rede, permite configurar parâmetros de controle e realizar análises gráficas do controlador em tempo real. Ferramentas que permitem executar comandos e monitoramento remotos a pessoas geograficamente distantes dos grandes centros de ensino oferecem vantagem, sendo relevantes e essenciais devido aos desafios causados pela pandemia de coronavírus COVID-19.

Keywords: Control; Data Acquisition; embedded controller; Internet of Things, Remote lab.

Palavras-chaves: Controle; Aquisição de dados; controlador embarcado; Internet das Coisas, Laboratórios remotos.

1. INTRODUCTION

Control Systems integration is an essential need to provide data access and to optimize processes. Modern technologies of internet created new possibilities in the world of connected machines but how to solve the integration problem with the diversity of controllers that we know as of today? Worldwide many studies are investigating the IoT architectures as the solution.

In the article from J. Sáenz et al (2015), was presented an open course, offering several virtual and remote laboratories on automatic control, accessible to anyone. Was used others materials compared to this experiment but with similar goal, interesting solution was the integration with a learning management system (Moodle).

An experiment with architecture based in the internet technologies would be able to demonstrate the solution to facilitate systems integration providing the remote access to the laboratory experiment. From Y. Liang et al (2018), was presented a method to design high-quality 3D equipment for virtual laboratories, designed on large-scale educational purpose with merits of saving expenses for universities and to provide more opportunities for students. Their conclusion is that simulations with extensible web-based control units can offer a flexible way to provide experiment for large-scale users. In the work of Barber, Ramon et al (2013), was described a Simulink lab practice using Arduino as low cost hardware, the real plant, consist of a DC motor.

In an industrial plant where many control processes and machines are operating in the production lines, why not get all them connected thru the network with the implementation of database server to allow security data access and data manipulation. We can control machines just manipulating data thru the network. On a railway traffic control, would be possible to implement such embedded controller in the train motor, based on IoT solution, that we could control each train speed, distance control and acceleration control offering passengers comfort and safety? This seems impossible, but the safety depends on the implementation concept. Consider the example of the train controller getting an instant message from the central server to keep moving otherwise the brakes will be automatic applied as safety protection. It could keep traffic control safe, the trains can't move out of the commanded distance. This approach may affect performance and the solution is to combine sources of network gaining system availability (heterogeneous network).

This work brings a new approach to facilitate the architecture by implementing a tool for control systems development based on IoT that allows easy data access. It demonstrate that was possible to implement, in the Laboratory of Automation and Control (LAC) at Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP), the control plant of the DC motor based on an IoT architecture.

Tools that allow remote control and monitoring to people geographically distant from large educational centers is an advantage, also are essential; to give remote access and control to the universities laboratories. Nowadays, due to the COVID-19 pandemic, the new challenges require to reinvent the ways of working while the social distancing is mandatory, it gained relevance as key solution to keep schooling programs progress, a situation in which is impossible to group people are demanding new solutions.

2. GOAL

Our goal is to demonstrate and analyze the topology concept using an experimental Control Systems Development to verify the feasibility of this proposed tool as the solution to facilitate systems integration and to provide remote access to the laboratory.

3. DEVELOPMENT

The experiment consists of the control plant development. Application software development that is able to control the hardware GPIO. We developed an electronic circuit interface between the GPIO and the motor. The motor was attached to a wheel to improve dynamics of the system. An embedded CPU was connected in the local area network by wireless device.

The web based software was developed to allow the human machine interface to the system.

This block diagram represents the proposed solution:

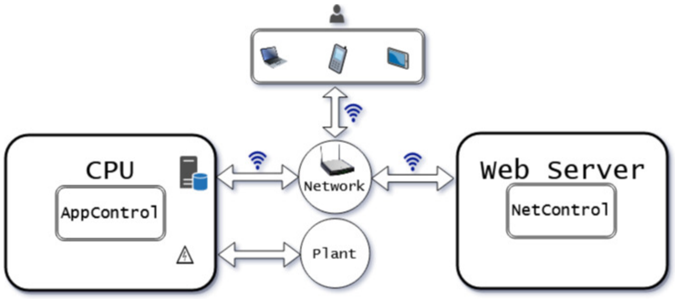


Fig. 1. Solution diagram block.

For the network interface solution, we applied a database management system to allow the application software to store and read data making the connection between embedded device and the external world. The application software is a Python script, it execute the control algorithm using the electronic signals from GPIO and store results as data into the database.

The interface human machine is a webpage, developed using HTML and PHP technologies.

During the experiments an oscilloscope was used to verify the hardware interface between the GPIO channel PWM output and the electronic circuit as well to verify its output to the motor.

We can represent the software architecture as:

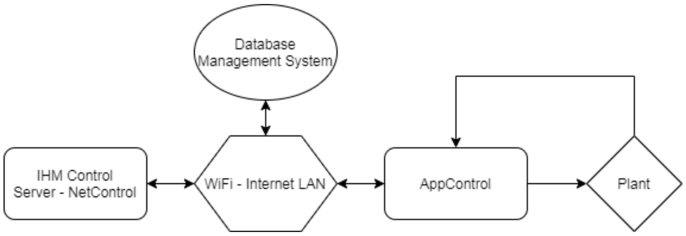


Fig. 2. Software architecture.

Where 'IHM Controller Server' do interface with the database 'c-treeACE®' thru wifi internet connection and the 'Controller Client' is responsible to run the control algorithms and do interface with database to read and store data. For this experiment, we implemented three of the most common controllers: PID, Fuzzy and open loop.

In the work of Kiam Heong Ang et al (2005), PID control system analysis, design, and technology we can find that the designing and tuning of proportional-integral-derivative (PID) controller appears to be conceptually intuitive, but can be hard in practice, if multiple (and often conflicting) objectives such as short transient and high stability are to be achieved.

The Equation 1 describes PID parallel:

$$f(t) = Kp\ e(t) + Ki \int_0^t e(t)\ dt + Kd \frac{de(t)}{dt} \qquad (1)$$

PID parallel diagram block is represented as:

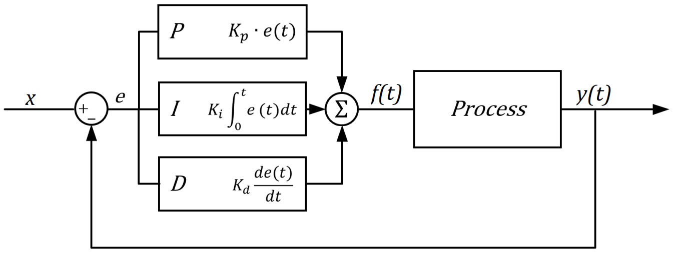


Fig. 3. PID parallel block diagram.

Where ‘x’ is the desired speed, ‘e’ is the error between output feedback control and ‘y(t)’ is the output signal. The implemented experiment:

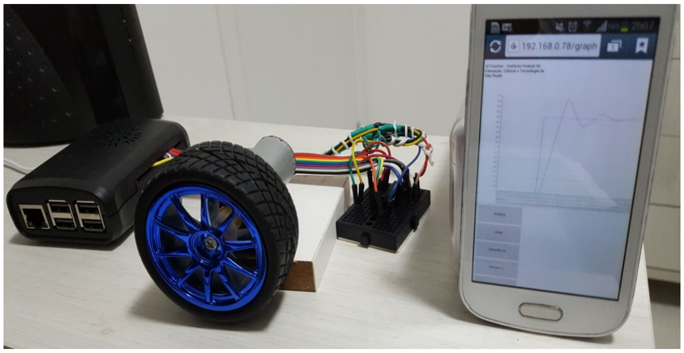


Fig. 4. Picture of the experiment showing the IHM running in the smartphone, the wheel under DC motor in PID control.

System components:

TABLE I. Hardware

Short Name	***Description***
PC	Computer.
RaspPi	Raspberry Pi, model 3.
wLAN	Wireless router.
dcMotor	DC motor with hall effect encoder
L293D	Driver L293D, electronic component.
+5V	Power Supply, +5V USB.

TABLE II. Software

Short Name	***Description***	***Release***
Raspbian	Operational system, Debian variant.	Jessie
App Control	Software developed in Python, part of IoTControl.	07/2017
Net Control	Mix of webpages developed in html5 and PHP.	07/2017
c-tree	Database management system.	V11
Apache	Webserver.	2.2.27

The DC motor is modelled using the own tool, where we can in open loop set the output value of the PWM and measure the speed of the motor counting the pulses. The speed calculation is based on interruption, executed each rising edge detected on pins 16 and 20, the signal phase will determine the direction of motor rotation, the command speed can be positive (clockwise) or negative (anti-clockwise), the PWM value is always greater or equal to zero. The pins 24 and 25 are used to command the direction.

The voltage applied in the motor is determined by the PWM value generated by the GPIO connected to the driver L293D, the DC voltage level is supplied by the power source. The PWM output from 0 to 100% in steps of 10 maps its relation to speed in pulses per second, as in the graphic show this:

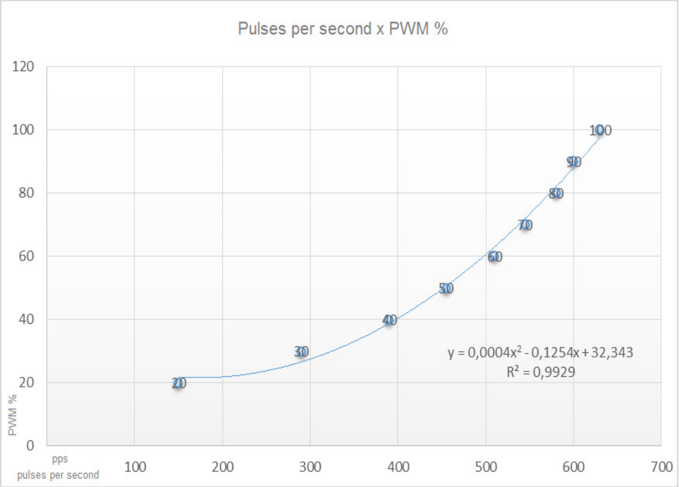
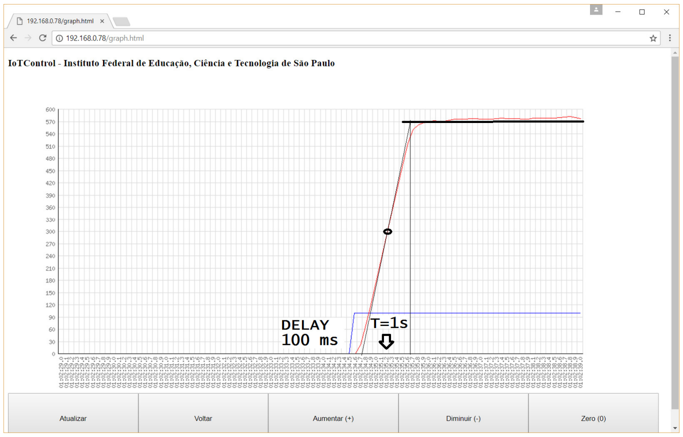


Fig. 5. PWM x Speed.

The modelled equation is:

$$y(t) = 0{,}0004x^2 - 0{,}1254x + 32{,}343 \qquad (2)$$

For tuning purposes, in open loop we can determine the factors Kp, Ki and Kd to be used in the PID controller, it is possible by applying the Ziegler-Nichols step response method, as described in the work of Ashwaq Abdulameer et al (2016), is a recommended example of method for tuning PID controller to DC motor speed control. In this experiment by using the graphical tool under the webserver (NetControl), is possible to get the values of T, L and the gain. Represented below the graphical analysis of step response:



IoTControl - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo

Fig. 6. Step Response.

The L is 100ms, T is 1s and plant gain is 6,3. This correspond to Kp = 1,9; Ki = 0,2 and Kd = 0,05.

4. ELECTRONIC

The electronic is the DC motor connected to L293D driver which is controlled by the Raspberry Pi pins 23, 24 and 25. The power supply is coming from Raspberry Pins, 2 and 6 (+5V and GND). The encoder supply +3V3 come from Raspberry pin 1, the encoder pulses are capture by Raspberry inputs on pins 16 and 20.

The circuit diagram presented gives a connection perspective figure of the electronic wirings.

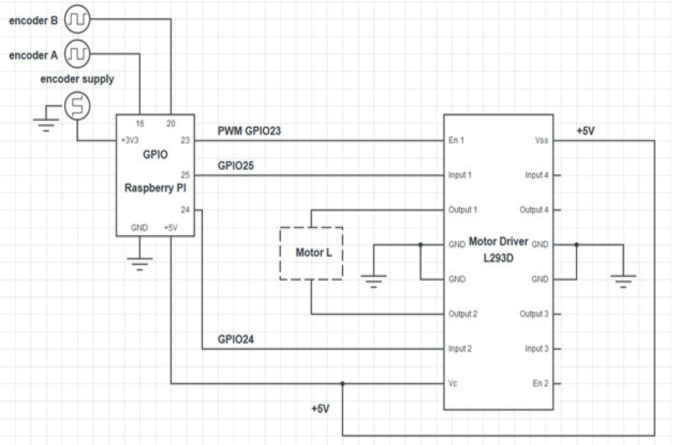


Fig. 7. Electronic.

The control signal was verified by using an oscilloscope to measure the range from 0 to 100%, down below is the measurement of the PWM at 80%:

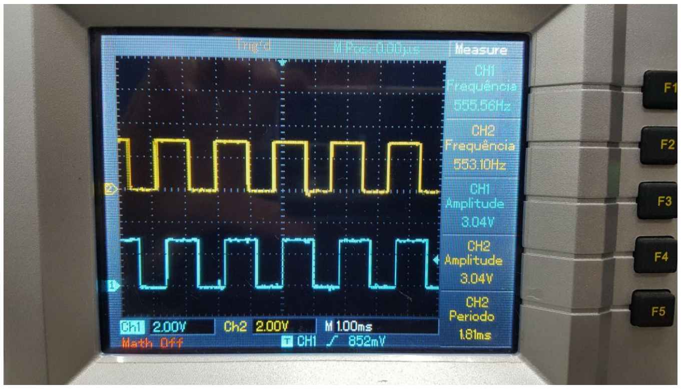


Fig. 8. Oscilloscope.

5. SOFTWARE

The operational system, Raspbian with installation of webserver Apache, driver unixODBC, PHP, Python and the database c-treeACE is the main software base. The software development to create the real experiment, consist of the Application (AppControl) and Network (Net Control) software that was developed to demonstrate the concept idea. These software components form the software system we used to build the platform of IoT for Control System Development.

These components are disposed and represented in the Figure 2. Software Architecture, during the experiments it was identified the possibility to increase the system performance by using ISAM database technology which offers sequential access methods for insertion of data then facilitating the CPU hardware performance. An improvement of the IHM can be achieved by applying modern techniques of webpage programming like HTML5 combined with JavaScript but this was not the goal of this work, rudimentary webpage look was implemented to facilitate the code and to reduce programming code hours. The controller selection, initial screen:

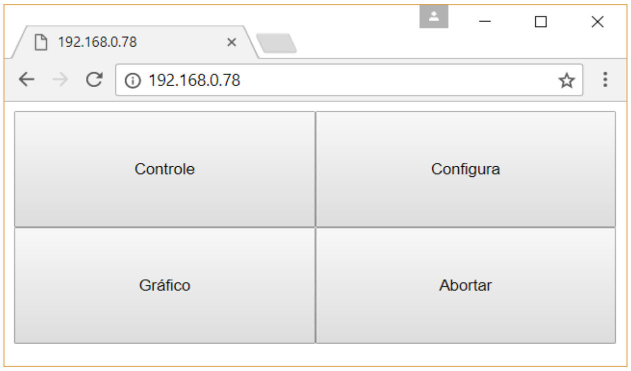


Fig. 9. Initial Screen.

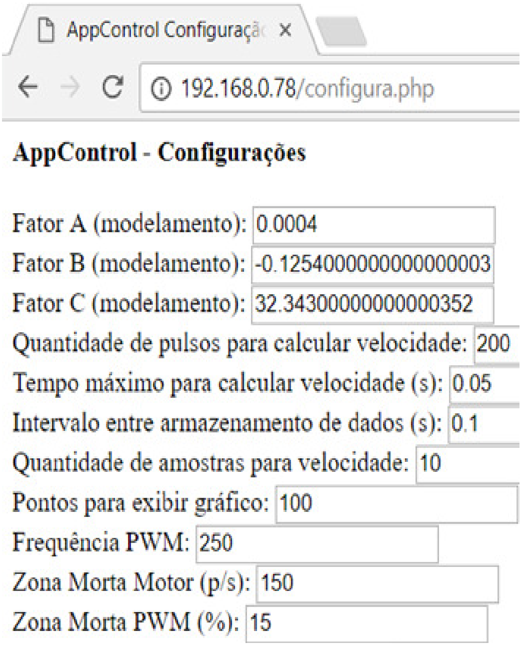
AppControl, is a script running in Python, implement the controller algorithms using the database c-tree to store and read data such as parameters for the system configuration, internal timings and others. These parameters can be managed by the user running the NetControl thru its webpages.

The goal was to let the system most configurable as possible to allow usage as the web experiment, making possible remote users to connect in this platform and make their experiment in a real controller. From the system configuration is possible to change factors a, b and c which are represented in the equation (2). The quantity of pulses and the time interval between calculations of speed is also configurable. The DC motor used do have a dead zone in lower voltages, we allow this configuration as parameter in the minimum frequency generated to move the motor. The PWM frequency of the system output is also configurable, the user may experiment different motor behavior due to these settings changes.

The parameters are also stored in the database system, the algorithm controls the stages for the data refresh upload as per event request. A level of production is applied to make able the system to switch from an algorithm to another allowing the user to replace the controller without stop the process. The protection allows the users to operate safe while the system is running.

The PHP request does interface with the database, this access is done thru ODBC. The database can be changed as per user preference because the ODBC isolate the software by abstracting the database.

The webpages was developed, the system configuration screen:



AppControl - Configurações

Fator A (modelamento): 0.0004
Fator B (modelamento): -0.12540000000000003
Fator C (modelamento): 32.343000000000000352
Quantidade de pulsos para calcular velocidade: 200
Tempo máximo para calcular velocidade (s): 0.05
Intervalo entre armazenamento de dados (s): 0.1
Quantidade de amostras para velocidade: 10
Pontos para exibir gráfico: 100
Frequência PWM: 250
Zona Morta Motor (p/s): 150
Zona Morta PWM (%): 15

Fig. 10. System Configuration.

The NetControl allow user to get a graphical view of control plant data, user can get a plot of motor speed in time.

The following figure is the system response while the PID configured with parameters from Ziegler-Nichols step response method:

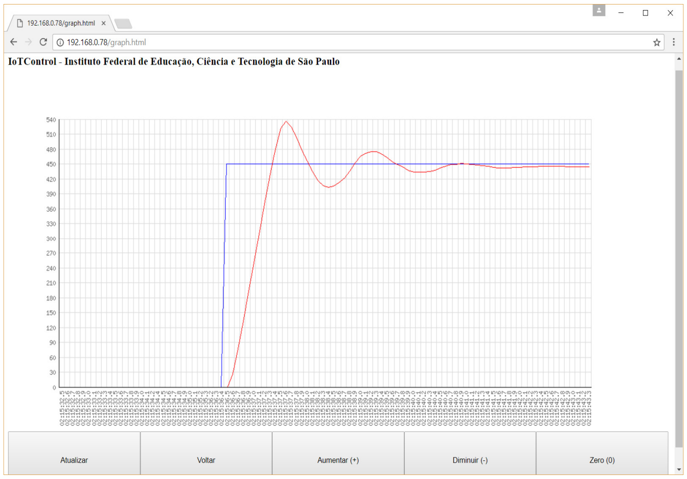


Fig. 11. PID graphic.

6. RESULTS

With the experiment built was possible to use the own tool to plot graphic then execute results analysis, it was also possible to use an external tool for this purpose and execute analytics outside, the architecture offer flexibility to use others tools.

Our focus is to demonstrate what is possible to do with the tool developed for control systems based on IoT. The results presented in this chapter were obtained from the controller developed, running algorithms and hosting the webserver in the same hardware.

The result below was obtained from the tool, is a screenshot of the graphical view when the user commanded the desired speed to the value 500 pulses/s, after a while it was commanded to invert the direction by setting the speed to the value of -500 pulses/s in the PID controller:

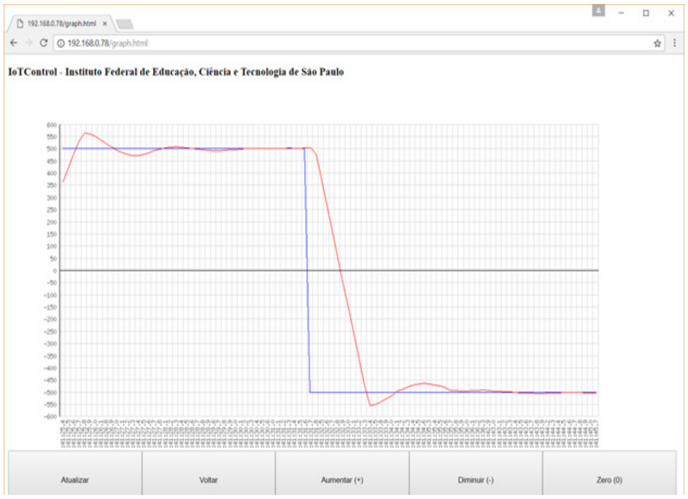
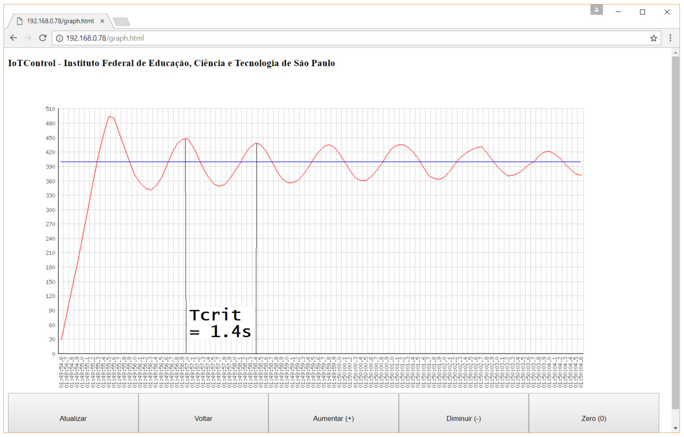


Fig. 12. Direction Change.

This experiment may help initial students to understand how controllers work, as example; we can set the parameters to obtain PID factors using Ziegler-Nichols Marginal Stability method:



IoTControl - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo

Fig. 13. Marginal Stability.

For learning purposes there are experiments to be explored, the tool also offers others algorithms like open loop, PI and Fuzzy:

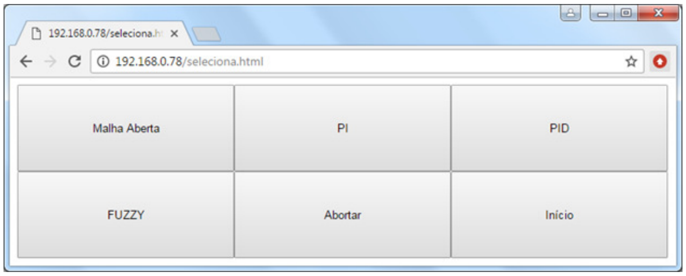
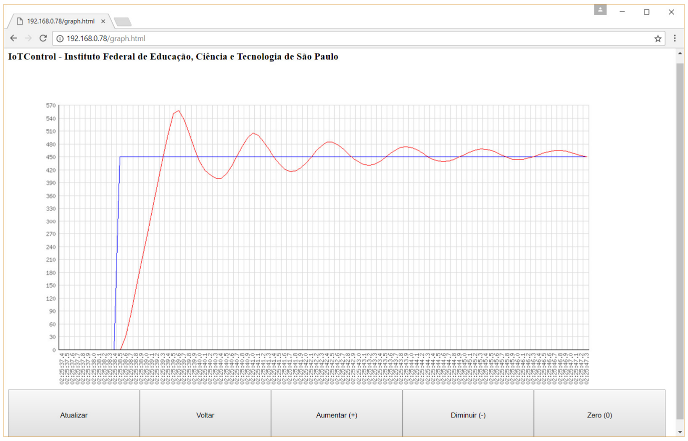


Fig. 14. Algorithms.

Plant running random parameters values:



IoTControl - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo

Fig. 15. Kp = 2,4 Ki = 0,7 Kd = 0168.

7. CONCLUSION

The developed experiment presents the results of the IoT application in the control system development of minimalist plant of DC motor speed control. This approach facilitate the architecture by implementing a tool for control systems development based on IoT architecture, it was the solution but there are challenges as we can find described in the work from S. Wilson et al. (2020).

This topology concept, using an experimental Control Systems Development, verifies the proposed tool as the solution to facilitate the systems integration. It creates possibilities, the experiment can be accessed using a cell phone, or a tablet, any device able to run the web browser and connect into the wifi network. The experiment of the speed control for DC motor facilitates the demonstration and execution from anywhere, just requiring network access, the results can be monitored remotely.

The platform as a tool for control system development creates new possibilities, for future work we could integrate this system with others systems. It could be expanded to a bigger IoT application, reducing human interaction, expandable by scale of the elements quantities, 10, more than 100 or even more than 1000 DC motors in a way to benefit the students learning. Would be great to work on features for: integration with others plant processes; multi-thread server to manipulate multiple client access; implement control over cloud computing with fail-safe strategy for communication lost.

ACKNOWLEDGMENTS

This work was supported by the IFSP – Instituto Federal de Educação, Ciência e Tecnologia de São Paulo. The database server ‘c-treeACE®’ was courtesy of FairCom, the others materials used in the experiment was provided by the authors.

REFERENCES

Ashwaq Abdulameer, Marizan Sulaiman, MSM Aras, Dawood Saleem (2016). Tuning Methods of PID Controller for DC Motor Speed Control. *Indonesian Journal of Electrical Engineering and Computer Science* 3.2: 343-349.

Barber, Ramon & Köllmer, Mario & Crespo, Jonathan. (2013). Control Practices Using Simulink with Arduino As Low Cost Hardware. 250-255.

J. Sáenz, J. Chacón, L. De La Torre, A. Visioli and S. Dormido (2015). Open and Low-Cost Virtual and Remote Labs on Control Engineering, in *IEEE Access*, vol. 3, pp. 805-814.

Kiam Heong Ang, Gregory Chong, and Yun Li (2005). PID control system analysis, design, and technology. *IEEE transactions on control systems technology* 13.4: 559-576.

S. Wilson et al. (2020). The Robotarium: Globally Impactful Opportunities, Challenges, and Lessons Learned in Remote-Access, Distributed Control of Multirobot Systems, *IEEE Control Systems Magazine*, vol. 40, no. 1, pp. 26-44.

Y. Liang and G. Liu (2018). Design of Large Scale Virtual Equipment for Interactive HIL Control System Labs, in *IEEE Transactions on Learning Technologies*, vol. 11, no. 3, pp. 376-388, 1.