

# Modelling the Virtual Laboratory for Control System Stability Simulation with Variable Speed Drives



Filip Filipovic, Milutin Petronijevic, Nebojša Mitrovic, Bojan Bankovic  
University of Niš  
14 Aleksandra Medvedeva  
18000 Niš, Serbia  
{filip.filipovic@elfak.ni.ac.rs}, {milutin.petronijevic@elfak.ni.ac.rs}  
{nebojsa.mitrovic@elfak.ni.ac.rs} {bojan.bankovic@elfak.ni.ac.rs}

**ABSTRACT:** *In this work we modelled the virtual laboratory for control system stability simulation with variable speed drives. We have deployed the Pi Raspberry as the web server for this work. We have used Modbus master and Arduino Due which are used for data acquisition with various affordable sensors.*

**Keywords:** Raspberry Pi, Arduino, Python, Internet of Things

**Received:** 25 August 2020, Revised 29 November 2020, Accepted 7 December 2020

**DOI:** 10.6025/ed/2021/10/1/17-24

**Copyright:** With Authors

## 1. Introduction

Modern laboratories for electrical engineer education often have constraints not in the lack of equipment for basic experiments, but in the lack of time each student has the equipment on its disposal for experimentation. On the other hand, traditional teaching of subjects such as control theory or electrical machines involves pictures, graphs and formulas. Possibility of doing experiments on the laboratory equipment and observing the results in real time without actually being in the laboratory can be a valuable addition for both aforementioned cases. Leveraging ideas present in the internet of things concept, but inverting usual structure that involves many sensors and less users is foundation for the base idea “The possibility of remotely conducting or observing created experiments using low-cost hardware and analysing results, all within current access level?”. Users of the laboratory have accounts and different access levels controlled by the administrator, which can be modified to include experiment observation, stability simulation, experiment setup or any combination of the previous.

## 2. Materials and Methods

Computing power of widely accessible hardware has increased to the point of which web servers, data acquisition systems and even fully functional computers can be bought for a price just over 40€ (price of Raspberry Pi 3 model B). Very popular as an experimentation platform, Raspberry Pi can be seen as a web server of virtual laboratory [1], an assistant in control education

[2], [3] and [4] or in various monitoring systems [5] and those are just some of the examples. In order to maintain low cost of the laboratory, cheap open source hardware was used (Raspberry Pi and Arduino Due), and all used libraries are free of charge for both individual and commercial application. Structure of the system is shown in Figure 1.

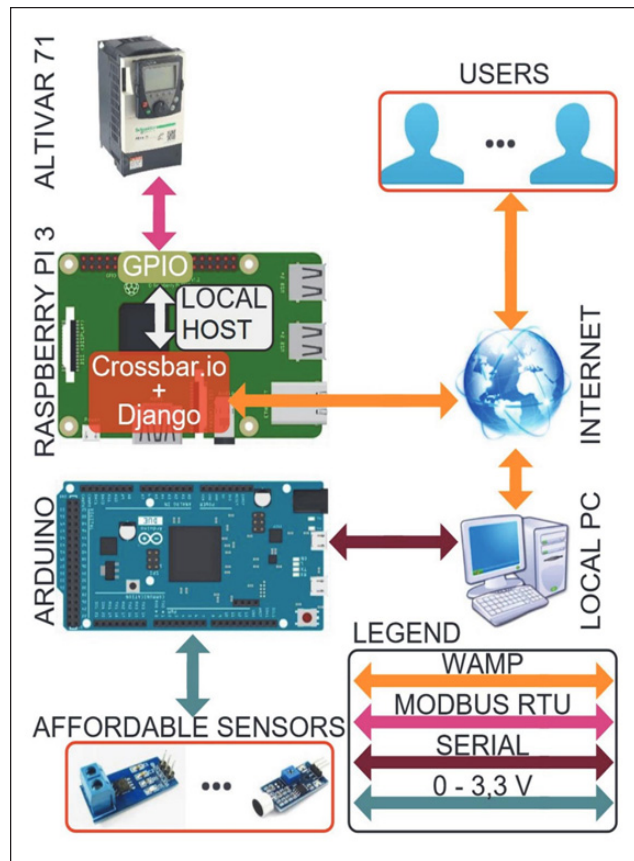


Figure 1. Overview of the virtual laboratory

Existing variable frequency drive, Altivar 71 [6] was used for virtual laboratory, only preconfigured to accept speed reference via Modbus RTU communication protocol. Problem of communication instability with software based Universal Asynchronous Receiver Transmitter (UART), on Raspberry Pi's General-Purpose Input Output (GPIO) pins, was solved by processor frequency throttling that led to stable communication. Bidirectional real-time communication is achieved with crossbar.io software library [7] that uses Web Application Messaging Protocol (WAMP), a WebSocket subprotocol. Its Router is paired with Django framework [8] to achieve user administration, security and page organization of web-based laboratory.

Since faster sampling rates were desirable (as low as 10  $\mu$ s) 32-bit SAM3X8E ARM Cortex-M3 CPU based Arduino Due was chosen instead of Arduino Uno. It uses 5 of its 12-bit analog inputs (pins A0 to A4) to accept the signal from any sensor, as long as the signal voltage is in range 0 – 3,3 V. It is configured to record any number of points per channel (up to 40000 total) and then transmit them back. There is a trigger option, where users can define if they want a trigger, will it be on rising or falling edge, what value, and lastly how many points before trigger should be recorded. Lack of Ethernet connection on this device introduced another device that serves as an internet gateway, or in this case old PC with internet connection. The Arduino Due was not connected directly to the Raspberry for the purpose of demonstrating internet of things concept, where data can be acquired on one place, and then transmitted to other devices for processing.

Users can examine a control system stability of arbitrary single input single output (SISO) system by its Bode amplitude and phase and Nyquist plot. After a system model is made the transfer function of each block is sent back to Raspberry for calculation. It uses specialized libraries, Numpy [9] and Scipy [10], for stability calculation and arrays of numbers are returned to users only to be visualized on the device. In an equivalent way user can view system response, and all graph visualization in

the virtual laboratory are done using plot.ly JavaScript library. Python was chosen as a predominant programming language, due to its syntactical simplicity and coding speed. The virtual laboratory components written using it are webserver, program for Modbus RTU communication and program on Arduino connected PC that, besides gateway purpose, optionally calculates Fourier transform and/or does signal filtering (using Kalman filter). In addition, JavaScript is used for web page behaviour and CSS for graphical design.

### 3. Main Results

Web-based virtual laboratory can be accessed over any device with modern browser (like Firefox, Chrome or Edge). Bootstrap framework [11] ensures optimal component layout for any screen size and aspect ratio. Default page displays only the result of the most recently conducted experiment, and for dynamic system simulation access or experimentation setup access, user must be logged in and appropriate privileges granted.

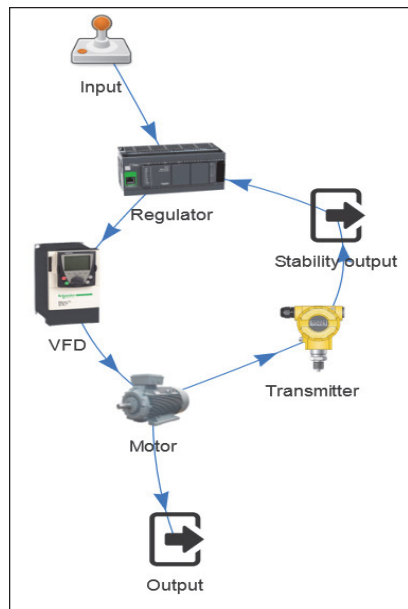
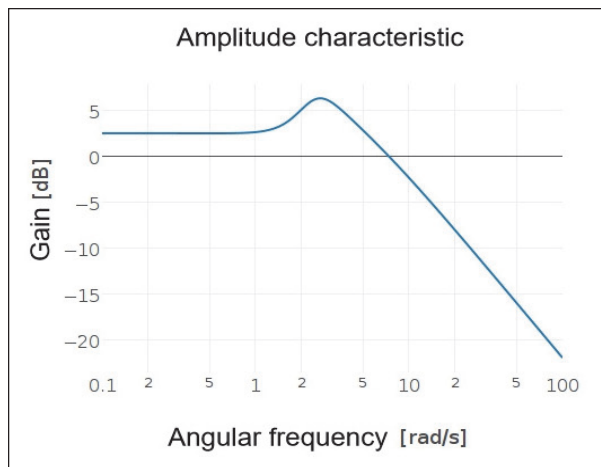


Figure 2. Overview of the control system simulation example

Figure 2. shows overview of a simulated system where each picture represents one block with a transfer function, except predefined blocks “Input”, “Output” and “Stability output” which are used for input signal selection, system response plot and system stability plot, respectively.



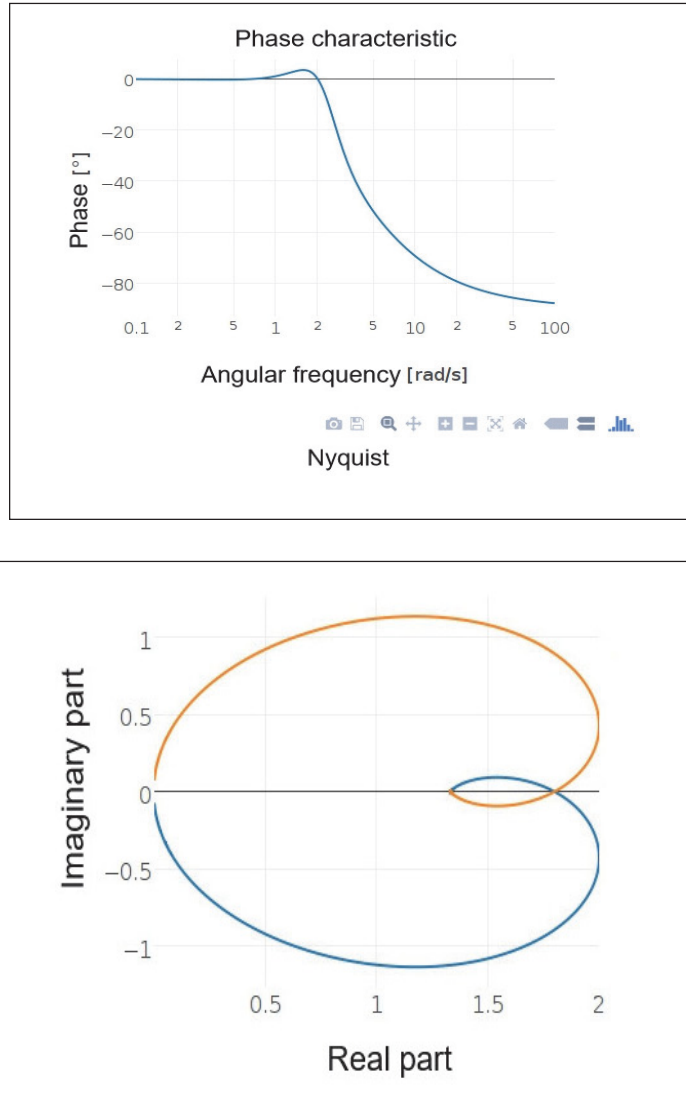


Figure 3. Simulated system stability plots

System stability plots, available on double-click of the icon “Stability output” are shown in Fig. 3. This part of the virtual laboratory provides possibility of modelling and simulating system before conducting experiments. That can be useful when, for example, parameters of regulators are calculated, simulated for theoretical output and then the results with the real-world experiment are compared. Choosing input signal on the icon “Input” and observing output on the icon “Output” can be seen in Figure 4.

Experimentation part of the virtual laboratory involves one variable frequency drive and a few cheap Hall effect based current sensors for Arduino, although further expansion is possible. In order to start the experiment, initial setup has to be done. Variable frequency drive is connected to the Raspberry via Modbus RTU, providing a way for a parameter setup if needed by writing appropriate values in drive’s registers of interest. In the initial setup, drive has its parameters configured to use the Modbus reference as speed reference and to provide voltage on its analog output proportional to motor’s voltage frequency. After drive setup, the user chooses channels of interest, defines sampling time and optional trigger and/or filter details for each channel individually. At last, duration of the experiment and shape of the reference are defined. Referent value can be created from three signals, with constant, linear and quadratic dependence of experiment time by defining value of each signal and time signal is activated and deactivated. Process of experiment configuration represented by flowchart can be seen in Fig. 5.

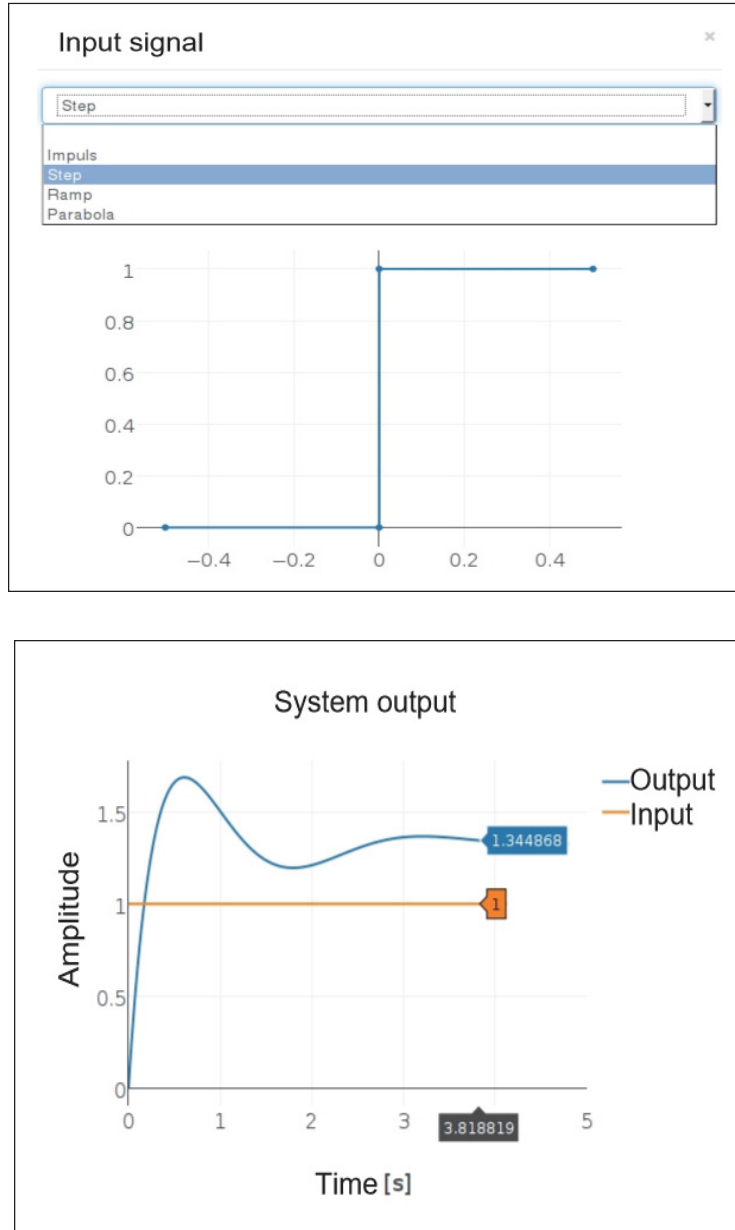


Figure 4. Selection of system input signal (upper figure) and appropriate output (bottom figure)

Users of the virtual laboratory can have insights in results after the experiment, because in general, sampling times of Arduino channels are too short for real time data transfer, or in another word, serial communication is the bottleneck of the system. Results are distributed to all users who have the laboratory's web page open. Fig. 6. shows current waveform over 20 ms period, and its spectrum analysis. The current waveform represents noise present during acquisition while no current passes through the current sensor. Although it might seem as a large amplitude of noise, the current sensor has rated current of 30 A, and noise of up to 0,3 A represents 1 % of rated current error, and manufacturer defines this error to be 1, 5 % at 25 °C for used current sensor ACS712 [12].

There is also a possibility of viewing live-stream video of the experiment using a USB camera attached to the used PC, but its usefulness can be a subject of discussion. For stream functioning, YouTube streaming service is used, and embedded inside the laboratory's web-page. This enables recording of experiment's video besides graphs, but this video has unpredictable lag, from 10 to 20 seconds, which makes it difficult to connect exact moments in the conducted experiment on video and graphs.

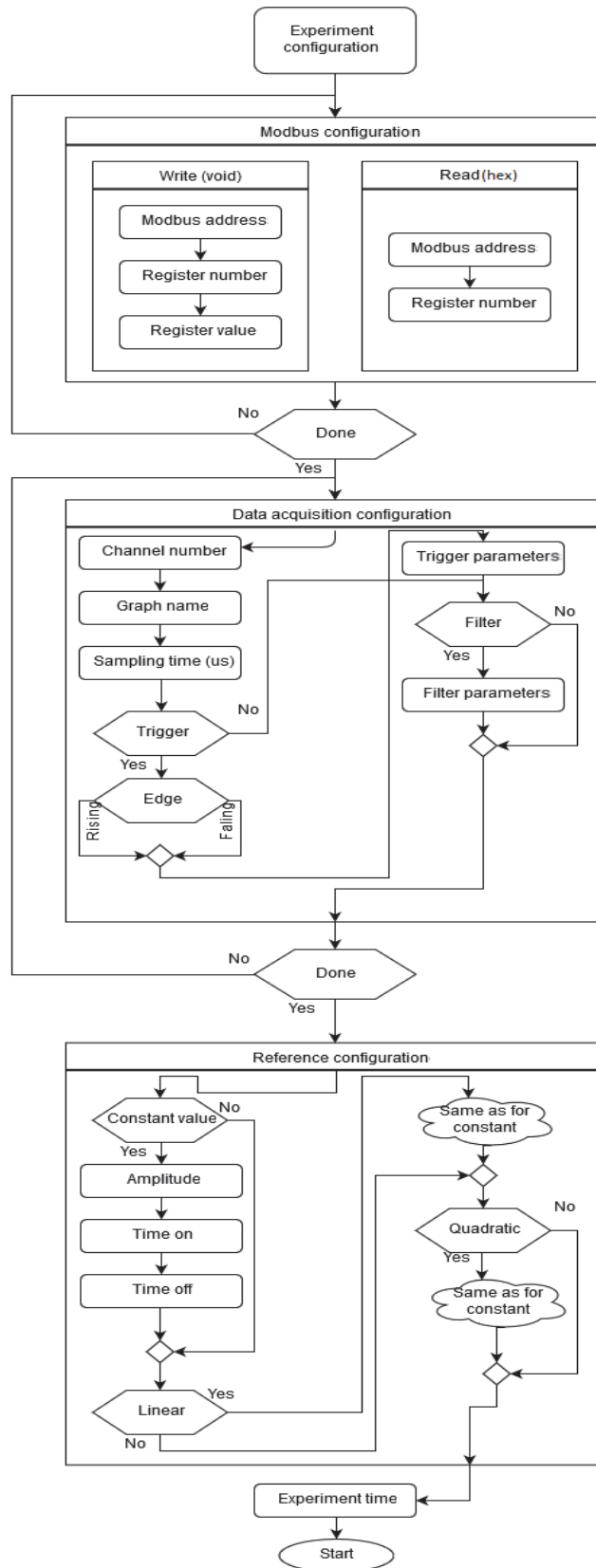


Figure 5. Process of experiment configuration

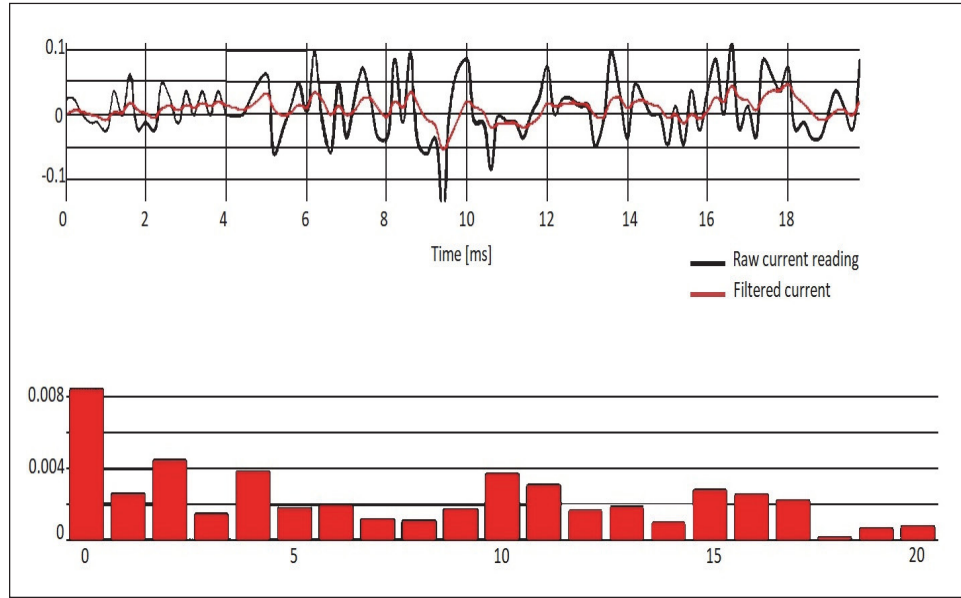


Figure 6. Waveform of recorded current (raw and filtered) and spectral analysis of filtered current

The WAMP lacks disconnect fall-back procedures and that requires from the user to manually refresh the web page after internet connection problems. Although experiments now have defined running time given on their setup, if later virtual laboratory expansions include real time commands over the internet, this could mean a time period of lost control over system when proper internet connection is not established. In that case, further safety procedures must be taken for ensuring equipment safe operation. Modern browsers are required for establishing WAMP connection (browser like Internet Explorer 9 and earlier versions do not satisfy this requirement) but this is not a problem for common users.

#### 4. Conclusion

Conducting experiments on the real equipment over the internet can be beneficial for students in a way that they can test newly formed knowledge without paying attention to motor wiring, frequency converters and measurement equipment. Laboratories can be more accessible to all students, and live experiments can even be conducted during teaching classes. Their realization is not too expensive, and in this case, added hardware parts for virtual laboratory realization (Raspberry Pi and Arduino with sensors) represent the cheapest part (with combined price just over 80€). Laboratory precision of the measured data is not primary concern, and if greater precision or faster sampling times are required, it can be achieved using dedicated microcontroller boards or cards for data acquisition or by using sensors of greater quality. The laboratory itself can be a subject of experimentation, whether with improving existing software or hardware.

Does this mean that traditional approach with laboratory experimentation is obsolete? No, because this type of laboratories can provide easier access to endpoints of experiment (initial parameter setup and results), but not to physical creation of experiment. Physical setup of some experiment is part of the education. This type of laboratories can be a supplement to an engineer's practical education but not its central part.

#### References

- [1] Bermúdez-Ortega, J., et al. (2015). Remote Web-based Control Laboratory for Mobile Devices based on EJS, Raspberry Pi and Node.js, *IFAC-PapersOnLine*, 48 (29) 158-163.
- [2] Sobota, J., et al. (2013). Raspberry Pi and Arduino Boards in Control Education, *IFAC Proceedings*, 46 (17) 7-12.
- [3] Reguera, P., et al. (2015). A Low-cost Open Source Hardware in Control Education. Case Study: Arduino-Feedback MS-150, *IFAC-PapersOnLine*, 48 (29) 117-122.

- [4] Jamieson, P., Herdtner, J. (2015). More Missing the Boat— Arduino, Raspberry Pi, and Small Prototyping Boards and Engineering Education Needs Them, 2015 *IEEE Frontiers in Education Conference (FIE)*.
- [5] Ferdoush, S., Li, X. (2014). Wireless Sensor Network System Design using Raspberry Pi and Arduino for Environmental Monitoring Applications, *Procedia Computer Science*, 34, p. 103-110.
- [6] Variable Speed Drives - Altivar 71 | Schneider Electric, *Schneider-electric.com*, 2017. [Online]. Available: <http://www.schneider-electric.com/en/product-range/1155-altivar-71/>. [Accessed: 01- Apr- 2017].
- [7] Crossbar.io, *Crossbar.io*, 2017. [Online]. Available: <http://crossbar.io>. [Accessed: 01- Apr- 2017].
- [8] The Web framework for perfectionists with deadlines | Django, *Djangoproject.com*, 2017. [Online]. Available: <https://www.djangoproject.com/>. [Accessed: 03- Apr- 2017].
- [9] NumPy — NumPy, *Numpy.org*, 2017. [Online]. Available: <http://www.numpy.org/>. [Accessed: 03- Apr- 2017].
- [10] SciPy.org — SciPy.org, *Scipy.org*, 2017. [Online]. Available: <https://www.scipy.org/>. [Accessed: 03- Apr- 2017].
- [11] Mark Otto, Bootstrap. The world's most popular mobile-first and responsive front-end framework, *Getbootstrap.com*, 2017. [Online]. Available: <http://getbootstrap.com/>. [Accessed: 10- Apr- 2017].
- [12] *Allegromicro.com*, 2017. [Online]. Available: [http://www.allegromicro.com/~media/Files/Datasheets/ACS71\\_2-Datasheet.ashx](http://www.allegromicro.com/~media/Files/Datasheets/ACS71_2-Datasheet.ashx). [Accessed: 11- Apr- 2017].