Universidad de Alcalá
BIBLIOTECA

Document downloaded from the institutional repository of the University of Alcala: http://ebuah.uah.es/dspace/

This is a postprint version of the following published document:

Paredes Larroca, F., Settone Olschewski, E., Quino Favero, J., Rosales Huamani, J. & Castillo Sequera, J.L. 2020, "Development of a fuzzy logic-based solar charge controller for charging lead-acid batteries", in Proceedings of CSEI 2019, Advances and Applications in Computer Science, Electronics and Industrial Engineering, part of the AISC, vol. 1078, pp. 168-183

Available at https://doi.org/10.1007/978-3-030-33614-1_12

© 2020 Springer

(Article begins on next page)

This work is licensed under a

Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

International Conference on Computer Science, Electronics and Industrial Engineering (CSEI)

CSEI 2019: Advances and Applications in Computer Science, Electronics and Industrial Engineering pp 168-183 | Cite as

Development of a Fuzzy Logic-Based Solar Charge Controller for Charging Lead–Acid Batteries

Authors | Authors and affiliations

Fabricio Paredes Larroca, Erich Saettone Olschewski, Javier Quino Favero, Jimmy Rosales Huamaní, José Luis Castillo Sequera

Conference paper
First Online: 24 October 2019

133
Downloads

Part of the Advances in Intelligent Systems and Computing book series (AISC, volume 1078)

Abstract

The design and implementation of a solar charge controller for lead–acid batteries is intended to supplement a component of the water purification module of the water treatment unit for natural disaster relief. This unit contains a solar panel system that supplies power to the module by charging batteries through a controller comprising an Atmega 328 processor. The solar panel feeds voltage to the batteries through fuzzy logic-based software, which allows up to 6 A DC to pass through the controller's power circuit. Consequently, the battery was charged in less time (an average of 7 h to reach maximum capacity), wherein battery lifespan is related to the charge wave frequency. Thus, our software may be adapted in different control algorithms without having to change hardware.

Keywords

Fuzzy logic Controller Battery charger

CSEI: International Conference on Computer Science, Electronics and Industrial Engineering (CSEI)

Advances and Applications in Computer Science, Electronics and Industrial Engineering

Proceedings of the Conference on Computer Science, Electronics and Industrial Engineering (CSEI 2019)

Editors (view affiliations)

Jyrki Nummenmaa, Federico Pérez-González, Bruno Domenech-Lega, Jean Vaunat,

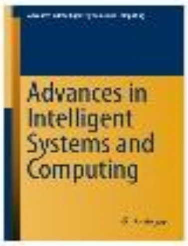
Félix Oscar Fernández-Peña

Conference proceedings
CSEI 2019

3 Citations 2 Mentions 3.1k Downloads

Part of the Advances in Intelligent Systems and Computing book series (AISC, volume 1078)

SpringerLink



Advances in Intelligent Systems and Computing

Book Series
There are 972 volumes in this series
Published 2012 - 2020

About this series

The series "Advances in Intelligent Systems and Computing" contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing such as: computational intelligence, soft computing including neural networks, fuzzy systems, evolutionary computing and the fusion of these paradigms, social intelligence, ambient intelligence, computational neuroscience, artificial life, virtual worlds and society, cognitive science and systems, Perception and Vision, DNA and immune based systems, self-organizing and adaptive systems, e-Learning and teaching, human-centered and human-centric computing, recommender systems, intelligent control, robotics and mechatronics including human-machine teaming, knowledge-based paradigms, learning paradigms, machine ethics, intelligent data analysis, knowledge management, intelligent agents, intelligent decision making and support, intelligent network security, trust management, interactive entertainment, Web intelligence and multimedia.

The publications within "Advances in Intelligent Systems and Computing" are primarily proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

**** Indexing: The books of this series are submitted to ISI Proceedings, EI-Compendex, DBLP, SCOPUS, Google Scholar and Springerlink ****

Development of a Fuzzy Logic-based Solar Charge Controller for Charging Lead–Acid Batteries.

Fabricio Paredes Larroca[1] [0000-0001-8857-9253] and Erich Saettone Olschewski [1][0000-0003-4849-3038] and Javier Quino Favero[1] [0000-0003-0073-2925] and Jimmy Rosales Huamaní[2] [0000-0002-3737-8694] and José Luis Castillo Sequera[3] [0000-0002-9131-1618]

[1] Universidad de Lima, Lima, 15023, Lima, Perú.
fparedes@ulima.edu.pe, Esaetton@ulima.edu.pe, jquinof@ulima.edu.pe.
[2] Universidad Nacional de Ingeniería, Lima, 15333, Lima, Perú.
jrosales@uni.edu.pe
[3] Universidad de Alcalá, Madrid, E-28871, Madrid, España.
jluis.castillo@uah.es

Abstract. The design and implementation of a solar charge controller for lead–acid batteries is intended to supplement a component of the water purification module of the water treatment unit for natural disaster relief. This unit contains a solar panel system that supplies power to the module by charging batteries through a controller comprising an Atmega 328 processor. The solar panel feeds voltage to the batteries through fuzzy logic-based software, which allows up to 6 A DC to pass through the controller's power circuit. Consequently, the battery was charged in less time (an average of 7 h to reach maximum capacity), wherein battery lifespan is related to the charge wave frequency. Thus, our software may be adapted in different control algorithms without having to change hardware.

Keywords: Fuzzy Logic, Controller, Battery Charger.

1 Introduction

The extensive use of microprocessor technology [1] yields simple solutions to complex problems through electronics, which allows both complex and moderately advanced algorithms to be quickly and easily programmed into new environments, such as MatLab. Thus, these algorithms may effectively contribute to solving problems from other fields. This hardware–software combination emerges as an embedded system used to address problems in multiple areas of technology, such as charging batteries through solar panels.

Presently, two types of solar energy charging systems are used: the maximum power point tracking that maximizes solar panel power output and the power width modulation (PWM), which slowly lowers the amount of power supplied to batteries when almost fully charged and is used herein. Feeding power to autonomous system components such as water purifications systems for disaster relief requires a system designed to supply large voltage amounts. Therefore, the battery charging system required must be arranged in two separate banks and powered using two different charging systems, in which one system actually charges the battery and the other supplies the necessary power to prevent halting of the water treatment system. Further, this arrangement of two individual intelligent exchange systems is intended to provide operating autonomy and safety to the autonomous water treatment unit. Moreover, the charging system requires a controller to efficiently manage the energy from the battery charging systems, which are usually solar panels. A solar charge controller is a device that quickly and efficiently charges batteries using solar panel, thereby extending battery lifespan and managing voltage consumption throughout the charging process. Fast charging of batteries is important for providing sufficient energy to the water treatment module regularly, thereby guaranteeing continuous water treatment operations 24 h a day. Conversely, the method used for charging lead–acid batteries such as pulse charging may reduce charging times and increase battery lifespan [2].

This proposal focuses on building a custom battery charger based on the features described above using electronic components available in the local market. This development method includes the following advantages: an *ad hoc* design suitable for outdoor operation in local weather conditions; easy access for the replacement of electronic components such as the control board and power board, which are different and independent herein; and the considerable reduction in importing costs for a finished product with these characteristics. In fact, this development may considerably contribute to the decrease in operating costs associated with off-grid rural electrification recorded in Perú since 1993 [3].

In this paper, Section 1 provides background information, Section 2 presents a conceptual framework for fuzzy logic, Section 3 describes the materials and methods used, Section 4 details the design and development of the proposed controller, Section 5 denotes our results, Section 6 discusses our conclusion, and finally, Section 7 suggests possible future works.

2 Fuzzy Logic Controller

Herein, the Mamdani model is used. This system features the following components (figure 1):

2.1 The Fuzzifier

The value incoming from the voltage sensor [4] is converted to a language that can be processed by the Atmega 328 processor. Further, the data converted into fuzzy values will be processed by the Inference System. These fuzzy values correspond to the universe of input variables.

2.2 The Fuzzy Inference System

This structure contains the different membership levels originated in the fuzzifier. These are the values processed to create a fuzzy output.

2.3 The Fuzzy Rule-Based System

It is a set of rules that constitute the system’s engine (Figure 2). These rules are based on the information provided via daily work procedures used by the system operator. Further, this procedure is interpreted using IF-THEN rules with a precedent and a result.

2.4 The Defuzzifier

The output generated through the Inference System is a fuzzy output that cannot be interpreted by the actuator of the system, indicating that the output must be converted to an analog value to achieve an adequate response. This analog value is obtained through the center of gravity for all possible answers, according to their membership degree [5].

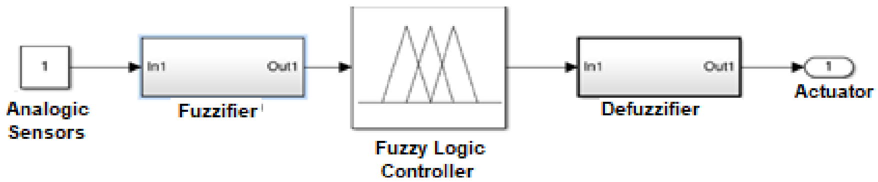
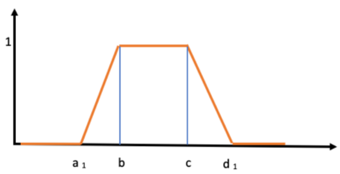


Figure 1- Fuzzy Logic System.



$$\mu(x)=\begin{cases} 0. & si\, x<=a_1 \\ \frac{x-a_1}{b-a_1} & si\, x\in[a_1,b] \\ 1. & si\, x\in[b,c] \\ \frac{d_1-x}{d_1-c} & si\, x\in[c,d_1] \end{cases}$$

Figure 2- Fuzzy Control Trapezoidal functions.

The program uses four membership functions [6] for the fuzzifier and defuzzifier, which are trapezoidal [7].

3 Materials and Methods

The circuit was designed using the Eagle® software from Autodesk. The electronic board was manufactured via a Roland CNC Milling Machine using a 0.3–mm-diameter drill bit to carve circuit tracks in the ceramic board.

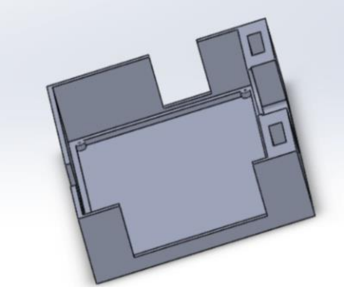
Currently, the Atmega 328 processors support the creation of new electronic devices, in which the power of the processor may generate quite interesting solutions when coupled with an adequate software. The joint development of the power board and electronic board, which host the control software, creates unparalleled capacities for materializing the design of an electronic device.

Following the board development stage, our purpose is enclosing these electronics in an optimized model. Further, our initial sketches and designs are poured through electronic drawing tablets, which supplement the computers wherein the design software is framed, to create the new shapes that will contain the electronic boards designed in the previous stage. Eventually, these designs may be improved to craft something tangible for the first time through 3D printing, such that these prototypes closely resemble a final product. Hence, the Solidworks® software was used to design the packaging of the electronics, and the model was printed in polylactic acid.

3.1. Power System

To develop this solar charge controller for lead–acid batteries, the circuit features two protection systems: a fuse at the solar panel output and another at the battery input. Moreover, the circuit includes an MBR 2545 Schottky Barrier Rectifier diode as protection from reversal polarity. Additionally, the NTE 4941 diode is available as a protection measure against input overloads. The circuit developed has three IRF9540 Power MOSFET gates. The first gate is operated by a button, which disconnects the solar panels. The second gate drives the PWM signal [8], which charges the battery at the frequency that has been previously hardcoded into the microprocessor. The third gate can be used to disconnect the feeding load at the operator's discretion. The circuit also features a few free pins for connecting a set of additional sensors. For example, a voltage or a current sensor can be connected for future analysis through a datalogger.

Figures 3–7 and 12–16 display the designed parts, along with their electronic components.



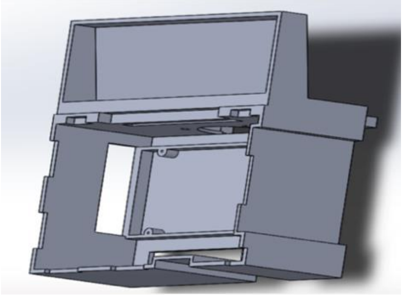
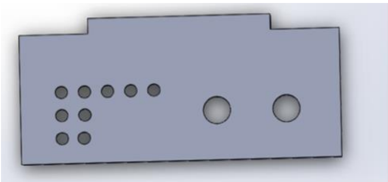


Figure 3 - Bottom view of the controller. Figure 4 - Top view of the controller.

Figure 3 shows the controller box, which has been designed in two sections to facilitate assembly. In this box, we may observe the location where the power board will be placed, as the inside area must also allow cables to pass through to the other boards. Figure 4 denotes the box where the controller is located, and it will be bolted to the top section of the box.



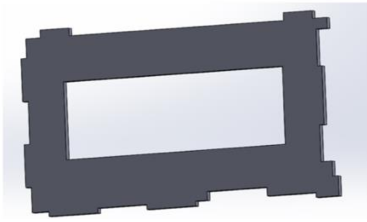


Figure 5 - LED Indicator Cover. Figure 6 - Upper mask of the controller.

Figure 5 shows the indicator cover, which houses the LED assembly, and the buttons used to connect the panel and feeding load. Figure 6 shows the frame that contains the LCD screen, which features 20 characters and 4 lines. This screen provides information about solar panel voltage, whether the panel is currently connected or disconnected and whether charging is enabled.

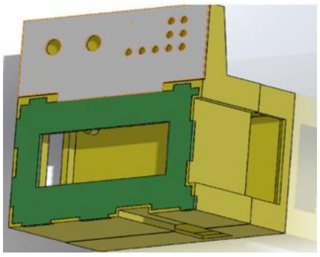
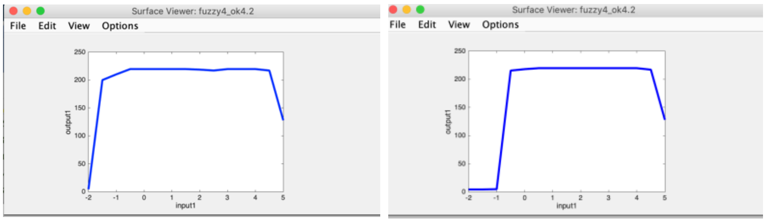


Figure 7 - Assembly modeled in SolidWorks.

Before printing the assembly, the software may be used to verify whether the controller will fit perfectly, as the proposal herein was to design a system that suits a particular need. Therefore, calculations were made for a simple controller, which may properly charge the battery pack, uses components locally available in the market and that does not consume more than 0.6 A. The initial works were not as successful because the peripherals consumed considerable voltage; however, this problem was solved using the processor described herein at 5 V. Further work may include developing new electronic boards containing 3.3 V processors for lower energy consumption during operation. The power board was developed using the generic MOSFET gates to quickly and easily replace them using conventional welding tools.

The fuzzifier’s rule base was optimized by changing the values of the fuzzy system coefficients of the fuzzifier’s rule base through MatLab® simulations using the Surface Viewer tool (Figure 8-a). When changing the coefficients, the error calculation chart becomes more smooth (Figure 8-b) and load behavior stabilizes.



(a) (b)

Figure 8 - MatLab-Surface Viewer tool.

Subsequently, these new coefficients were transferred to the fuzzy logic library in the comprehensive IDE-Atmega328 development environment.

4 Controller Design and Development

The design foundations addressed are based on the robustness of the power board, its easy construction, the easy loading of the firmware and developed control program because the whole system must be replicable at a low cost. The control rule base comprises the error variable, which is given by the difference between the setpoint and voltage of the battery. The aggregated error was not considered for the solution because the processor slowed down during tests. Other points considered for the development were the use of the free access “Embedded Fuzzy Logic” library, a processor widely available in the local market, easy integration in the development board, and the algorithm supporting the proposed rules.

The software programmed for this process was initially developed using the MatLab® environment through the fuzzy logic library [9]. Once the coefficients were identified through the simulation, these values were used in the fuzzy logic library of the Atmega 328 processor [10]. The calculated error is then transferred to the four trapezoidal input functions of the fuzzy system [11].

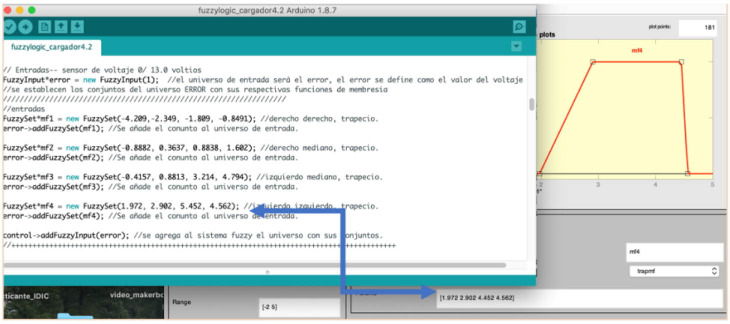


Figure 9 - MatLab values for the Fuzzifier transferred to the Fuzzy Logic library of the processor.

Only four optimized rules [12] are used for this controller because the entire program that controls peripherals demands excessive processing time and memory consumption, thereby slowing down the operation [13].

Table 1 - Control rules transferred to the fuzzy logic library of the processor.

Fuzzifier Rules	Fuzzy Rule Base System	Defuzzifier Rules
(Input)		(Outputs)
Left Input	If X then output Y, then ΔZ^1	Left output
Middle Left Input	If M then output N, then ΔZ^2	Middle Left Output
Middle Right Input	If L then output F, then ΔZ^3	Middle Right Output
Right Input	If O then output P, then ΔZ^4	Right Output

The output is the pulse-width modulation (PWM) signal, which is the response obtained through the membership rules [14] resulting from the calculation of the center of gravity given by the mathematical expression (1) [15].

$$Z^* = \frac{\sum_{i=1}^{M} \bar{Z} W_i}{\sum_{i=1}^{M} W_i} \qquad (1)$$

where Wi=μ(Mi)=min {μ1, μ2… μk} (2)

In Equation (1), Z* represents the locus of the gravity center for all the possible rules involved in the solution, where M represents the number of fuzzy outputs. In Equation (2), the term μk represents the fuzzy value involved in the antecedent of the i-th rule, where M represents the number of fuzzy outputs, with the corresponding W-weights for the maximum values of the results involved.

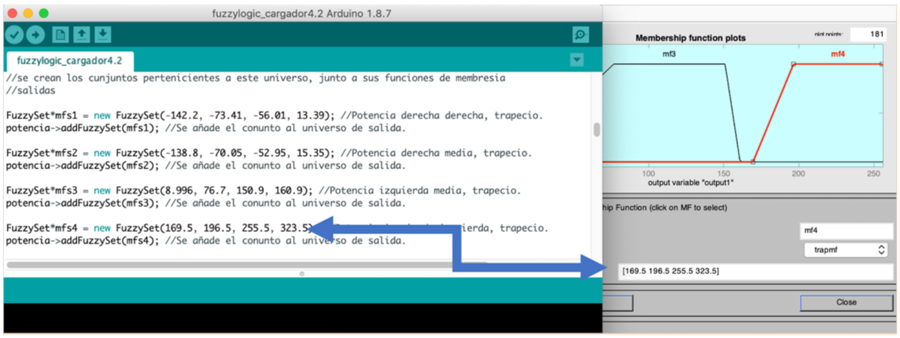


Figure 11 - MatLab values transferred to the fuzzy logic library of the processor.

In Figure 11, the MatLab simulated and tested values [16] are transferred to the microprocessor’s library, wherein the actual operation of the MatLab simulation is verified.

Through these development tools, the programming can take less time, which reduces the implementation hours required for a system developed in C++. The difference between the input voltage from the solar panel and battery voltage is the parameter used as a reference for the algorithm. Therefore, the board design incorporated voltage dividers to determine the voltage values for the battery and solar panel. Under low battery voltage conditions, the controller has supplied up to 6 A according to laboratory tests. With battery voltage at its peak, the work cycle decreases, and the system maintains battery voltage for the proposed setpoint.

4.1. Controller Implementation

The electronic boards are located inside the solar charger. One of them is the power board comprising three power transistors. The first transistor allows the PWM to charge the battery, the second transistor connects and disconnects the solar panel, and the third transistor activates or deactivates the load (consumption). This power board is prepared to support a workload of up to 8 A.

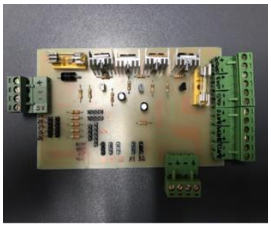


Figure 12 - Power Board.

To manage the charging process for the lead–acid battery [17], an Atmega 328 microprocessor, where the control software is installed, is used. The voltage dividers serve as voltage sensors so that the battery controller can perform the work as programmed. At first, the voltage starts charging the battery through the PWM control, which allows the voltage to overcome the impedance of the battery. When the setpoint is reached, the controller disconnects the voltage from the solar panel.

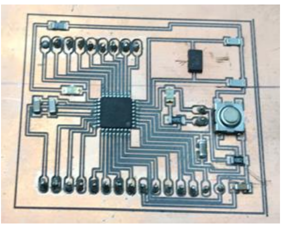


Figure 13 - Control board.

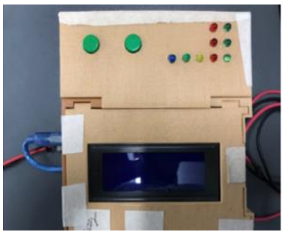


Figure 14 - Solar Charge Controller.

Figure 15 - Charging LED Indicators.

Figure 13 displays the control board using an Atmega328P-AU [18] with a 16 Mhz crystal, where two 22 pF ceramic capacitors are used via the resonator. The software developed was installed on board with superficial mounted devices elements. Figure 14 denotes the prototype of a fully functional solar charge controller. The LEDs in Figure 15 indicate whether the battery is being charged, fully charged, or completely drained. This figure also shows two buttons: the left button disconnects the solar panels and the right button deactivates the load that has been fed.

Figure 16 - Ventilation System.

Figure 16 displays the forced ventilation system, which is integrated into the system so that MOSFETs may work at a lower temperature, thereby extending their lifespan. The fan is powered at 12 V and is easy to replace and maintain. The system uses lead–acid batteries, which are not ideal for photovoltaic systems because they are not capable of supporting deep discharges. However, they are the most affordable and easily available in rural areas. The solar charge controller designed accounts for the inherent disadvantages of these batteries and adapts them based on their needs.

5 Results

Based on the tests performed during the two-month period in which the lead–acid battery was fully charged and drained for 60 cycles, the following results were obtained:

- The battery takes less time to fully charge than when using commercially available chargers. An approximate average of 7 h is obtained against the 12–14 h, which was recorded when using locally sold chargers.
- When the battery reaches its maximum charge, the system disconnects and stops charging the battery, thereby preventing voltage overloads.

The charts below show the working states for the controller in different charging regimes.

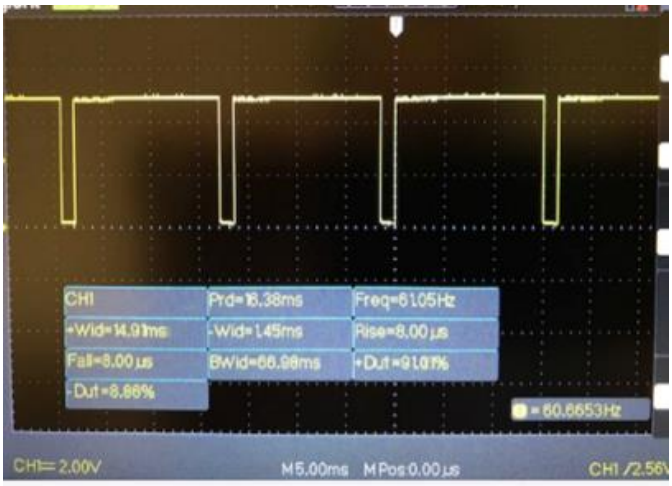


Figure 17 - PWM signal when the battery is drained.

When the MOSFET generates the PWM signal [19], its temperature increases and the forced air fan turns on.

The PWM is generated by the frequency that has been previously established in the microprocessor. The time percentage in which the pulse is at peak level with respect to the complete pulse period is known as the Duty Cycle, and it is expressed as a percentage. For example, a Duty Cycle of 91% at a frequency of 61.05 Hz, corresponds to a pulse of 16.40 ms, wherein maximum current is supplied for 14.91 ms before cutting supply off at 1.49 ms (Figure 17). Herein, the processor has applied a correspondence rule to operate at full load because the battery is below or equal to 10.5 V (Figure 18).

Figure 18 - Battery Current.

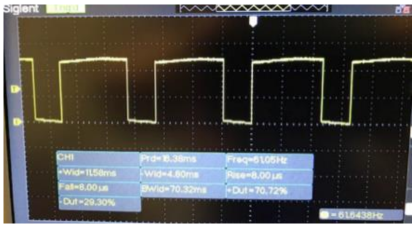


Figure 19 - PWM signal, the battery is reaching its maximum charge.

Conversely, when the battery charge reaches approximately 80%, the Duty Cycle changes to 70%, which means that the controller has applied another correspondence rule to start charging the accumulator when it is at approximately 12.8 V. The voltage supplied to the battery is controlled by the MOSFET, whose operation depends on the orders from the microprocessor exerting the fuzzy control. The voltage supplied continuously decreases as the battery is being charged. For example, if the battery is reaching maximum charge, the current decreases to 0.5 A, while the control algorithm disconnects the system and stops charging the battery. At this point, the current and voltage from the panels is cut off, leaving the fan on so that the MOSFETs, which are still at high temperature, may cool down.
The Duty Cycle or period fraction decreases as the battery is charged up to the setpoint, where it cuts off the voltage supply, and the battery has been fully charged.

Table 2 - Tests performed when charging a lead–acid battery. Controller behavior during a typical full charging cycle.

Hour	Duty Cycle (%)	Voltage (V)	Current (A)
8:00	94.19	9.76	6.11
8:15	80.02	12.25	4.76
9:55	82.78	12.85	4.96
10:15	85.52	12.98	5.06
10:45	82.78	13.18	5.09
11:00	86.30	13.23	5.28
12:00	89.04	13.53	4.49
12:30	91.04	13.70	4.06
1:00	93.84	13.84	3.86
3:15	60.66	15.78	1.57

The data displayed in Table 2 correspond to a charging Duty Cycle, wherein the lead–acid battery with a rated voltage of 12 V and a capacity of 53 Ah took approximately 7 h to complete the charging process. About 60 charging and discharging cycles were tested.

6 Conclusions

- The proposed system has charged the lead–acid battery in 7 h, which allows high flexibility considering the amount of sunlight available in the area.
- However, it was also determined that small changes in the fuzzifier rules cause large changes in the Duty Cycle of the battery, resulting in increased charging times.
- In rural areas with very low outdoor temperatures, changing the fuzzifier's rule base guarantees a suitable battery charging temperature.
- Because the lead–acid batteries do not support deep discharges as their lifespan drastically decreases, the controller prevents discharges from dropping below 10.5 V.
- Further, the controller was manufactured using components, which are readily available in the local market to ensure that it may be replicated.
- The robustness of the system was proven after performing laboratory tests for about four months without any difficulty.

7 Future Work

Using a single type of hardware for several types of algorithms allows the system developed herein to be deemed as a benchmark (Figure 20). This system currently uses three different control algorithms, external voltage sensors in the solar panel and battery, and current sensors to determine the amperes consumed by the load. Additionally, temperature sensors are present in the solar panel and battery, which collect data for future analysis.

Overall, the 16 sensors originate around 5500 data per day, which are transferred to a PC through a 16 analog channel National Instruments board for subsequent analysis of the algorithms contained in each load controller. This will help us to demonstrate the efficiency of our algorithms through Deep Learning techniques, which will allow us to find the best charging algorithm for the batteries. This test bank is valuable when determining the most effective algorithm contained in the three controllers for charging a lead–acid battery.

Figure 20 - Test bank using three solar panels and three controllers with different algorithms.

This test bank includes controllers with three different algorithms for future battery charging analyses. The first algorithm is the fuzzy logic algorithm, the second is an adaptive PID, and the third uses logical and comparison operators.

8 References.

1. Jaisin, C., Intaniwet, A., Nilkhoa, T., Maneechukate, T., Mongkon, S., Kongkraphan, P., & Polvongsri, S. (2019). A prototype of a low-cost solar-grid utility hybrid load sharing system for agricultural DC loads. *International Journal of Energy and Environmental Engineering*, *10*(1), 137-145.
2. Lam, L. T., Ozgun, H., Lim, O. V., Hamilton, J. A., Vu, L. H., Vella, D. G., & Rand, D. A. J. (1995). Pulsed-current charging of lead/acid batteries—a possible means for overcoming premature capacity loss?. Journal of Power Sources, 53(2), 215-228.
3. Javadi, F. S., Rismanchi, B., Sarraf, M., Afshar, O., Saidur, R., Ping, H. W., & Rahim, N. A. (2013). *Global policy of rural electrification. Renewable and Sustainable Energy Review*s, 19, 402-416.
4. Patil, A. R., Atar, K. D., Potdar, A. A., & Mudholkar, R. R. (2013). Embedded fuzzy module for battery charger control. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, *2*(8), 4072-4078.
5. Paliwal, P., Patidar, N. P., & Nema, R. K. (2012). Fuzzy logic based determination of battery charging efficiency applied to hybrid power system. *J. World Acad. Eng. Sci. Technol*, *71*, 1164-1168.
6. Chevrie, F., & Guely, F. (1998). Fuzzy logic-Cahier technique no 191. first issue, December.
7. *Sowah, R., Ampadu, K. O., Ofoli, A., Koumadi, K., Mills, G. A., & Nortey, J. (2016, October). Design and implementation of a fire detection and control system for automobiles using fuzzy logic. In 2016 IEEE industry applications society annual meeting (pp. 1-8). IEEE.*
8. Govidan, N., & Rajasekaran Indra, M. (2018). Smart fuzzy-based energy-saving photovoltaic burp charging system. *International Journal of Ambient Energy*, *39*(7), 671-677.
9. Sivanandam, S. N., Sumathi, S., & Deepa, S. N. (2007). *Introduction to fuzzy logic using MATLAB* (Vol. 1). Berlin: Springer.
10. Bawa, D., & Patil, C. Y. (2013). Fuzzy control based solar tracker using Arduino Uno. *International Journal of Engineering and Innovative Technology*, *2*(12), 179-187.
11. Çelik, B., Birtane, S., Dikbıyık, E., & Erdal, H. (2015, November). Liquid level process control with fuzzy logic based embedded system. In 2015 9th International Conference on Electrical and Electronics Engineering (ELECO) (pp. 874-878). IEEE.
12. Lara-Rojo, F., Sánchez, E. N., Zaldívar-Navarro, D., & Sur, P. L. M. (2001). Real-time fuzzy microcontroller for a didactic level system. In Electro (Vol. 2001, pp. 153-158).

13. Malkhandi, S. (2006). Fuzzy logic-based learning system and estimation of state-of-charge of lead-acid battery. *Engineering Applications of Artificial Intelligence*, *19*(5), 479-485.
14. Thao, N. G. M., & Uchida, K. (2017). *A two-level control strategy with fuzzy logic for large-scale photovoltaic farms to support grid frequency regulation. Control Engineering Practice, 59, 77-99.*
15. Velázquez-González, R., Gómez-Lemus, T., & Rodríguez-Reséndiz, J. (2017, May). *A pH process control embedded on a PLC using Fuzzy Logic. In 2017 XIII International Engineering Congress (CONIIN) (pp. 1-6). IEEE.*
16. Driankov, D., & Saffiotti, A. (Eds.). (2013). *Fuzzy logic techniques for autonomous vehicle navigation* (Vol. 61). Physica.
17. Bandara, G. E. M. D. C., Ivanov, R., & Gishin, S. (1999). Intelligent fuzzy controller for a lead-acid battery charger. In *IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 99CH37028)* (Vol. 6, pp. 185-189). IEEE.
18. Atmel Mega 328p Datasheet (s.f.). *En Atmel.com. Recuperado el 1 de julio de 2019, de: Atmel 8-bit AVR Microcontrollers ATmega328/P Datasheet Summarywww.atmel.com/.../Atmel-42735-8-bit-AVR-Microcontroller-AT...*
19. Silva, S., Soares, S., Valente, A., & Marcelino, S. T. (2015, July). *Digital sound processing using Arduino and MATLAB. In 2015 Science and Information Conference (SAI) (pp. 1184-1191). IEEE.*