Design of a Simpler Ampere-Hourmeter

Application in State-of-Charge Estimation in Lead-Acid Batteries through a Microcontroller

Lluís Farré*[1], Lluís Closas[2], Pau Casals[3]

Electronics Engineering Department, Electrical Engineering Department, Facultat de Nàutica de Barcelona, Universitat Politècnica de Catalunya – Barcelona TECH. Barcelona, Spain.

*[1]lluisfarre@yahoo.es; [2]lluis.closas@upc.edu; [3]p.casals@upc.edu

Abstract

In any application in which working with batteries is a must, the knowledge of the batteries' state-of-charge (SOC) is a fundamental parameter for anyone, so it determines the remaining capacity in the battery. There exist several methods for the estimation of this SOC in Lead-acid batteries; however, when the requisites of the measuring method must offer, besides precision and reliability, the possibility to integrate the results into an automatized system, the Coulomb's counter is the method that prevails. This paper presents, then, the design of a simpler Ampere-hourmeter based lead-acid battery SOC estimating system. Supported by previous studies in the field of SOC estimation in Hybrid Electric Vehicles and by experimental tests carried out by the researchers, the modelling of the simpler measuring system has been parameterized by following Peukert's Equation, and afterwards it has been integrated into a data acquisition and processing system designed through a CSNX25 current sensor and a low-cost and low-consume 16F877 microcontroller. The fundamental conclusion of this paper is that obtaining an accurate result while estimating the SOC of a lead-acid battery with an simpler Ampere-hourmeter developed through a microcontroller based system is achievable but only if the used microcontroller holds enough memory to include the whole data treatment algorithms.

Keywords

Battery; SOC; State-of-Charge; Microcontroller; Ampere-Hourmeter

Introduction

In any application related to batteries, predicting the state of charge (SOC) is necessary in order to determine the remaining capacity in the battery.

In every ship, as in any industrial facility, batteries supply the emergency power to back up systems. They represent, though, a key component of the vessel's engine and navigation systems. Batteries must always be well maintained, fully charged and ready for any emergency occurring. Knowing the SOC of a ship's battery is the basis of any batteries' bank maintenance; and achieving an automatic method to predict this SOC and to apply controlled charging cycles to the batteries, fulfils the whole battery maintenance problem.

There are several methods to estimate the SOC in lead-acid batteries [1][2][3][4]: electrolyte's density measuring, voltage measuring, Coulomb counting, internal resistance measuring… The first of these methods, the electrolyte's specific gravity measuring, is the only method that can be named as a non-estimating method because the electrolyte's density reflects the true SOC of the lead-acid battery, so it shows the available rate of the chemicals inside the battery. On the other hand, this method would be annoying to be integrated into an automatic system due to the need to acquire, or even design, a suitable electronic densitometer. The other three mentioned methods, despite being estimating methods which only lead to an approximate SOC value, are more prone to be automatized.

Focusing on the Coulomb counting method, as suggested by relevant studies in the field of battery SOC measuring [5], specifically in Hybrid Electric Vehicles (HEV) battery technology [6][7][8], this paper presents the development of an Ampere-hourmeter solution to the estimation of the SOC in lead-acid batteries by designing the complete structure of the SOC estimator in a much simpler way than the previous citations; obtaining a more restricted SOC measuring tool that, however, reveals the key for a further development.

First of all, the mathematical path, followed to determine the SOC, will be presented. Then, the design of the electronic circuits needed to acquire the data from the batteries, besides some laboratory test results, will be described. The next point will focus on the development of the software programmed in the

microcontroller. To conclude this paper, a critical revision of the results obtained from the prototype's testing in the laboratory, will be presented.

The Mathemathical Model

In 1897, W. Peukert exposed the results of his investigations on lead-acid batteries. In the published paper, it was revealed that his tests had led him to an empirical equation which linked the discharge of the batteries with their real available capacity [9].

This formula is known as the Peukert's Equation:

$$C_t = I^n t \tag{1}$$

Being C_t the real discharged capacity, I the discharge intensity, t the discharge time at the I rate and n the Peukert number.

The Peukert's Equation offered the possibility to work with a mathematical model which follows the behaviour of batteries in a discharge cycle closer than the direct extrapolation from just I and t. This n number, then, is a coefficient, unique for each battery and slightly variable through its living and environmental conditions, which breaks the faux linearity in the battery's model and approaches its behaviour to a more exponential work.

Following the Peukert's Equation, so, it becomes possible to obtain an evaluation of the SOC just by a few easy calculus steps.

Knowing, first of all, that the efficiency (E) of a battery is function of C_t and the utile capacity (C) [3]:

$$E = \frac{C}{C_t} = \frac{I t}{I^n t} = \frac{I}{I^n} = I^{1-n} \tag{2}$$

Then, the SOC for a discharging cycle can be put on the formulae:

$$SOC = 100\frac{C_o - C_t}{C_m} = 100\frac{C_o - I^n t}{C_m} \tag{3}$$

In percentage. Being C_m the maximum available capacity from the battery and C_o the capacity before starting the discharge.

Whereas in a charging cycle, the same path must be followed but making a few additional considerations.

$$C_{car} = (1+(1-E))C = (2-E)I t \tag{4}$$

So there are considered the charging loses during the process.

$$SOC = 100\frac{C_o + C_{car}}{C_m} = 100\frac{C_o + (2-E)C}{C_m}$$
$$SOC = 100\frac{C_o + (2-E)I t}{C_m} \tag{5}$$

Being C_o, in this equation, the capacity before starting the charging process.

Finally, then, the main equations to calculate the SOC in charging and discharging processes through the development of the Peukert's Equation have been obtained. Recapitulating the SOC shows to be in function of I, t, n, C_m, C_o and E.

And by measuring and calculating these parameters it is how the SOC value will be found in this paper.

However, it should be warned that n, as a number that parameterizes the chemical behaviour of the battery, is strongly dependent of environmental conditions as temperature, as well as of the aging of the battery. So, an accurate calculus of the SOC should include correcting factors for, at least, temperature and aging [3][10][11][12][13].

Due to technical boundaries, these corrections will not finally be implemented in this design. n will be, then, calculated for each one of the batteries and treated as a constant. However, what the effects of temperature and battery aging involve should always be put into consideration.

The Hardware Design

It is assumed, then, that to reach a SOC value the operating parameters in (3) and (5) must be previously known, and the key process to achieve the evaluation of the SOC is the procedure of measure and calculation of these parameters.

If the attention is focused on (3), it can be easily found that both C_m and n can be considered as given constants for a specific battery, whereas C_o is a direct result of the immediately previous charging/discharging process. Then, the only two parameters which do really must be measured are I and t.

While evaluating (5), the final conclusion becomes the same as above.

Therefore, the design of the SOC evaluator must follow two paths of development: On one side, a device which is able to measure the values of I and t; On the other side, a processor whose task will be the integration of all the given constants and measured

values into the equations of SOC calculation, (3) and (5).

To fulfil the first of these goals, it becomes necessary a sensing device. In order to process all the signals and data values, there is a compulsory need of a microcontroller.

PIC 16F877

The PIC 16F877 is a low-cost and low-consume microcontroller [14] that can easily be integrated into a system like this.

With its corresponding development board, it can receive and transmit data, so it has a 8 gates Analogical/Digital (A/D) converter, a LCD controlling port and a port to control external interruptions.

The PIC 16F877 can, then, develop the main desired tasks. Just by receiving the data of *I* through its A/D module, it could theoretically be programmed to calculate internally the elapsed time, *t*, while the measuring process occurs. Then, all necessary measured data can be obtained.

CSNX25

In order to measure the current and transmit the *I* data to the microcontroller, there is a need to use a current sensing device.

In this specific design, the chosen sensor is the CSNX25 [15][16], a bidirectional magnetoresistive sensor that delivers an output current proportional to a measured *I* up to 25 A. A range which goes from –25 to +25 A is a desired range for a prototype version of the design, and the bidirectionality of the sensor suits perfectly to a SOC calculating procedure which segregates the charging cycle from the discharge.

The 16F877 A/D module, however, only works with positive voltage signal values, so it turns necessary a first conversion from the proportional current signal emitted by the CSNX25 to a proportional tension value.

By following the datasheet indications [15], this first step can be done just by integrating a resistance between the output pins of the sensor.

Knowing that with a parallel connection of the primary pins of the sensor, the nominal output current, with 25 A in the primary, will be I_{Nout}=12.5 mA, then with the suggested 80 Ω resistance, the output tension value reaches 1 V.

However, this only solves part of the problem. The bidirectionality of the sensor is still a problem, so it gives an output range of –1 to +1 V. And negative values cannot be processed by the 16F877.

In order to obtain a positive response to all the measured current values, it is necessary the development of a signal processing circuit able to add a voltage level to the CSNX25 tension output.

Signal Processing Circuit

The incompatibility between the output signals of the current sensor and the analogical inputs of the microcontroller has restricted the design's bidirectional operation. However, this boundary can be broken by the integration of an intermediate signal processing circuit able to convert the CSNX25 emissions into a valuable input for the 16F877.

Due to the main problem which is the negative values of tension while using the sensor to measure negative currents, the solution is the addition of a voltage level to this tension output that makes all the outputs positive.

In order to obtain this signal process, the more suitable circuit is an instrumentation amplifier with offset on the ground. By adjusting this offset value to the desired, working with no negative values on the output becomes possible. Moreover, the gain (*G*) of the instrumentation amplifier can be also adjusted to increase the output resolution.

So, if the desired maximum output range is fixed between 0 and 5 V, the ideal offset becomes 2.5 V.

$$V_s = 2.5V + GV_o \tag{6}$$

And by knowing, as told before, that the original tension output value was up to 1 V, then, *G* can be set up to 2, to obtain a working range from 0.5 to 4.5 V.

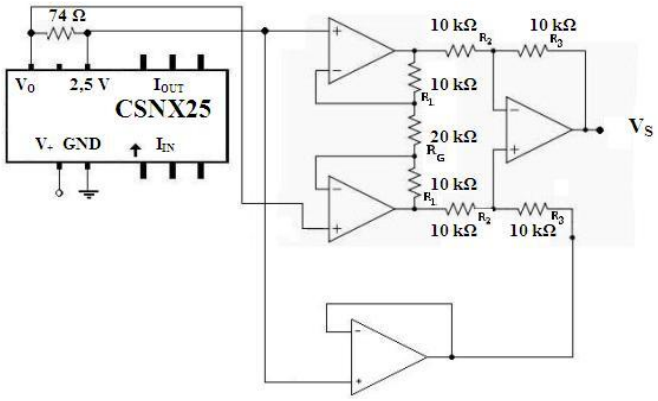

FIG. 1 SIGNAL PROCESSING CIRCUIT

$$G = \frac{V_s}{V_2 - V_1} = \left(1 + \frac{2R_1}{R_G}\right)\frac{R_3}{R_2} \tag{7}$$

$$G = \frac{V_s}{V_2 - V_1} = \frac{V_s}{V_o - 2.5V} = \left(1 + \frac{2 \times 10k\Omega}{20k\Omega}\right) = 2$$

Notice that a follower of the 2.5 V reference voltage has also been included.

And so, the final results of the sensing circuit, as tested in the laboratory, lead to the following graph.

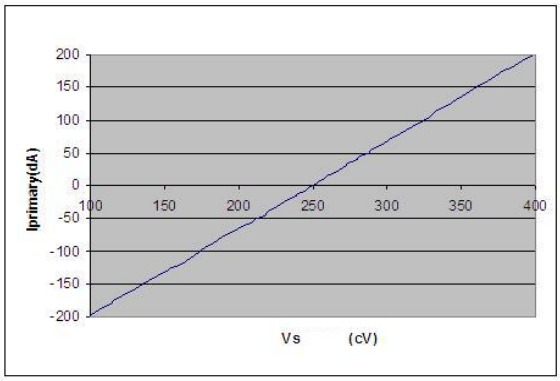

FIG. 2 SENSING CIRCUIT CHARACTERIZATION GRAPH

Which shows linearity between –20 A to 20 A by following the next equations:

$$I_{primary} = 1.35 \times V_s - 337.63 \quad (8)$$

$$-I_{primary\_negative} = -1{,}314 \times V_s + 328.79 \quad (9)$$

It is only necessary to be considered that (9) result must be always treated as a positive value, because of 16F877, as in the trending shown below:

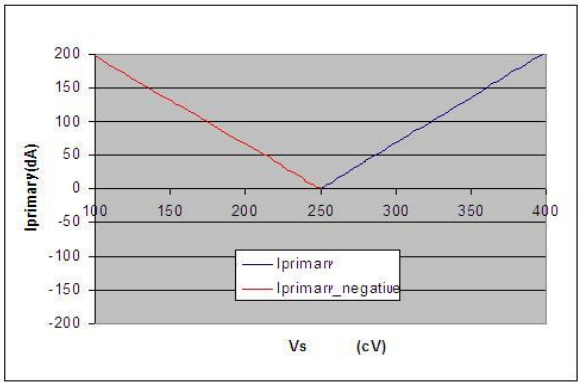

FIG. 3 SENSING CIRCUIT CHARACTERIZATION GRAPH IN PIC16F877

Becoming (9), then, in the forthcoming calculus:

$$I_{primary\_negative} = -1.314 \times V_s + 328.79 \quad (10)$$

The Microcontroller's Program

Presuming that the measurement of *I* has been solved through the design of the presented measuring circuit, the next step to be taken is to obtain the missing values to solve (3) and (5).

Recapitulating C_m and n are specific constant values for any battery, C_o must be calculated in situ after a charging or a discharging cycle, while E is calculated through (2) and, finally, t is to be obtained through a internal microcontroller calculus by using an internal timer of the 16F877 [14]. Once all these values have been either acquired or read as constants in the microcontrollers program, they are being operated in order to obtain a final result which leads to a SOC value readable for the operator.

These are the main tasks of the program, as shown in the following table:

TABLE 1 MICROCONTROLLER'S PROGRAM MAIN TARGETS AND TASKS

Objectives	Action
Receive voltage input signal through analogical inputs	Activate A port Identify one A port's pin as analogical input
Activate timer cycle Define constant values Data process	Internal program of the microcontroller
Show results through LCD	Activate D port Identify output pins to deliver output message

These actions to be taken in order to achieve the objectives must be included in the design of the software programmed in the 16F877. In order to develop this programming design, it is used the software for visual programming of microcontrollers Niple v5.2. [17], which allows a visual development of the solution through flow diagrams prior to the conversion to assembler code.

A flow diagram scheme of the program is presented next:

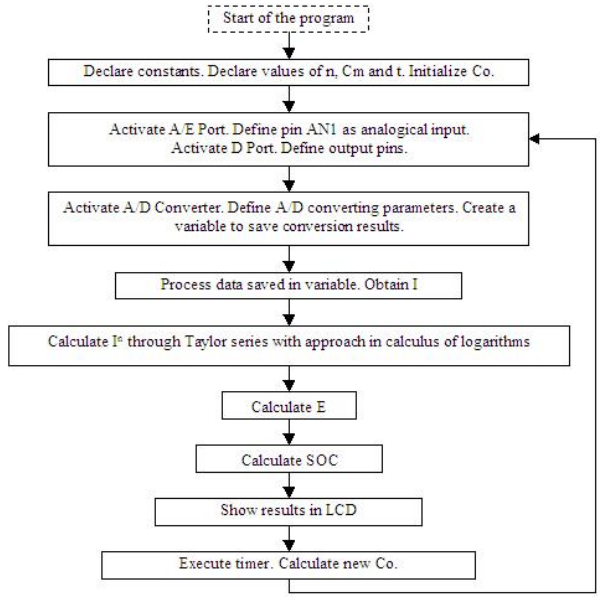

FIG. 4 MICROCONTROLLER'S PROGRAM BASIC SCHEME

Notice that some intermediate actions have been added in order to approach the scheme to the real structure of the program. One of the most important points is the calculus of I^n so it must be done through Taylor series in the microcontroller:

$$I^n = 1 + n\ln I + \frac{(n\ln I)^2}{2!} + \frac{(n\ln I)^3}{3!} + \ldots \quad (11)$$

And therefore there must also be calculated values for these logarithms. However, in this case, the choice made is to assign values in a matrix type table of variables for different values of I between the operating range.

The program, then, must estimate the value of I^n by using a double approach calculus: First of all, a Taylor development, with six terms; secondly, logarithm values which don't fit exactly the true values. These estimations necessarily lead to an error on the SOC calculation.

Moreover, as well as the 16F877 cannot calculate logarithms, or work with negative numbers stored in variables, it also cannot process decimals, so working with multiples of the desired numbers becomes necessary in order to achieve a minimum resolution. However, these numbers cannot be too high, either, in order to fit registers. For instance, the I calculus as a result of (8) and (10) becomes the following:

$$90 \times I_{primary} = 30(1.35 \times V_s - 337.63) \approx (40 \times V_s - 10000) \quad (12)$$

$$90 \times I_{primary\_neg} = 30(-1.31 \times V_s + 32879) \approx (-40 \times V_s + 10000) \quad (13)$$

Another focus of error.

Results and Error Analysis

In the previous points of this paper, the different parts, or modules, of the SOC estimator have been presented. Finally, in this last item, it is to be shown the running behaviour of the assembled prototype during the tests carried out on the laboratory by using a scheme such as this:

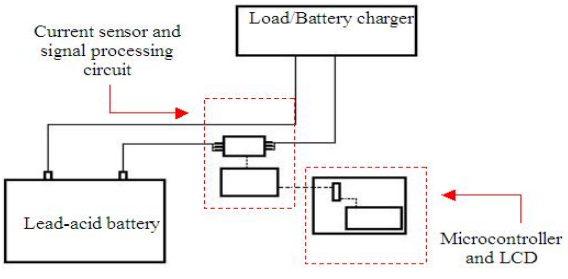

FIG. 5 SOC ESTIMATOR WORKING SCHEME

In this real case, while developing the last stages of the 16F877's program, the first boundary is found: a memory limit which forces a limit in the number of program lines. This problem is due to the microcontroller's construction, but it could be solved by using a programming software which allowed programming by interruptions. Rather than the Niple v5.2 case.

The main consequence of this memory limit is observed in the calculus of I^n and, more strongly, in the approximation of the logarithms. As it has been exposed, these mathematical operations need lots of factors to obtain an estimated value close to the real one. And with a memory limit it becomes impossible to develop the calculus.

Keeping up with the laboratory tests, then, for a 95 A·h battery, with n=1.26, the results obtained in the internal calculus of the microcontroller of SOC reveal the following working graph for a discharge cycle:

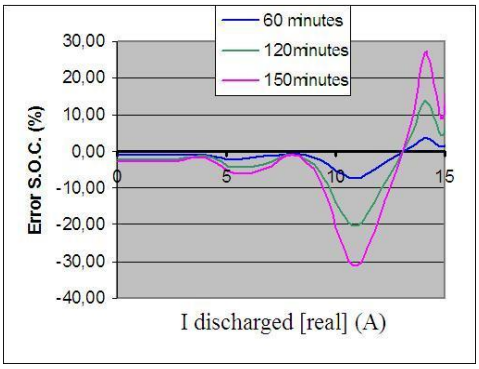

FIG. 6 SOC ERROR IN A DISCHARGE CYCLE

Notice that the error grows as the time of running increases due to the addition of the error in the calculus of the intermediate capacities; as well as it grows for higher current values, which is a easy predictable result.

And in the graph for a charge cycle:

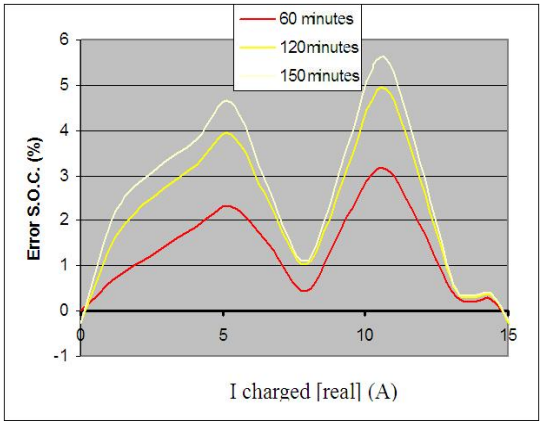

FIG. 7 SOC ERROR IN A CHARGE CYCLE

Again, errors appear, but with lower values in this case, due to the nature of (3) and (5). It is observed that in (3) I^n has a more relevant function while in (5) its weight is lightened by other factors.

In other words, by focusing the lab tests on the effects of this I^n error, it is found that it has a remarkable incidence in the calculus of the C_t (1) but lesser in C_{car} (4). In comparison with the errors in I, I^n and C_t and C_{car}, respectively, the key evidence to comprehend this wrong behaviour of the SOC calculator's prototype is found.

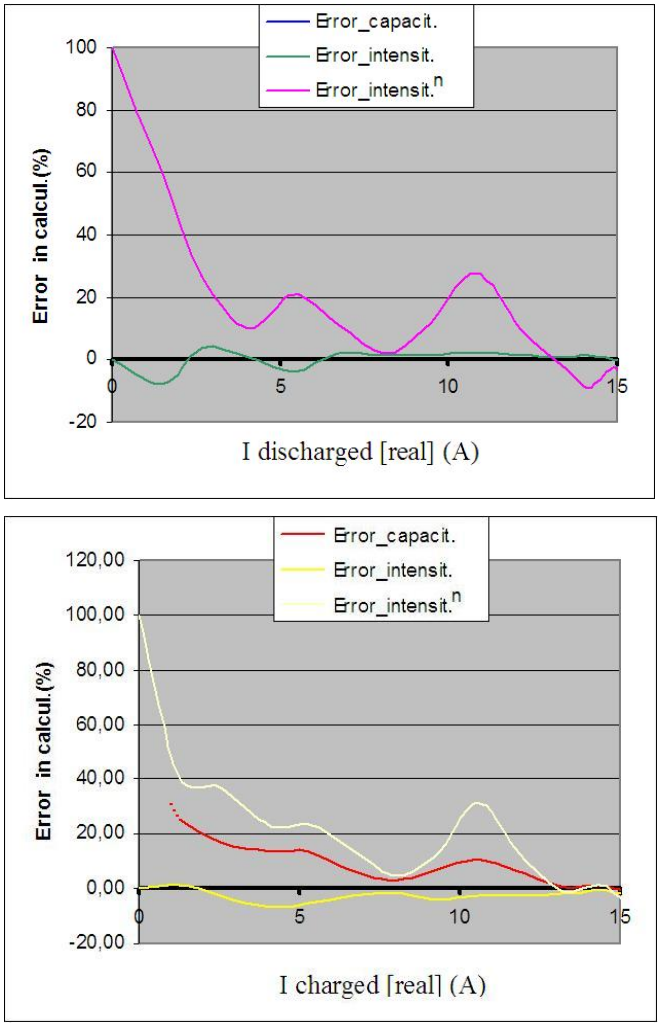

FIG. 8 COMPARISON BETWEEN INTERMEDIATE ERRORS

Notice that there is an insignificant error in the calculus of I through (12) and (13), an error which means almost nothing till it is operated in (11) to obtain I^n through a limited approach.

To prove this hypothesis, a solution of the main precision problem is found just by executing tests in which the I^n calculus error is reduced to an insignificant value by inducing an equivalent data signal to the microcontroller.

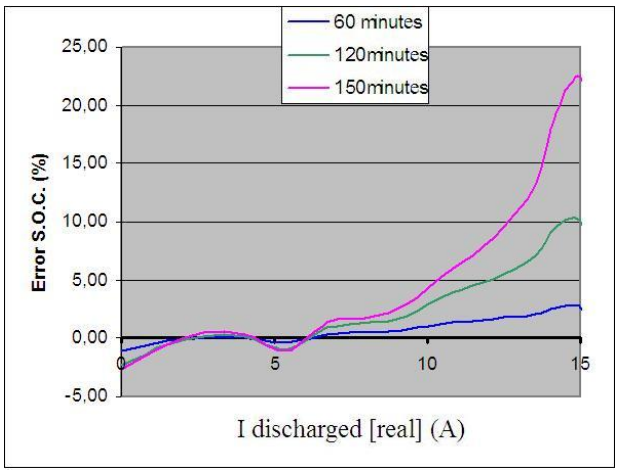

FIG. 9 SOC ERROR IN A DISCHARGE CYCLE. SOLUTION APPLIED.

The error is almost eliminated for intensities lower than 10 A, and it performs predictably so it could be corrected.

In a charging cycle, the results are also positive, reducing the error in SOC value:

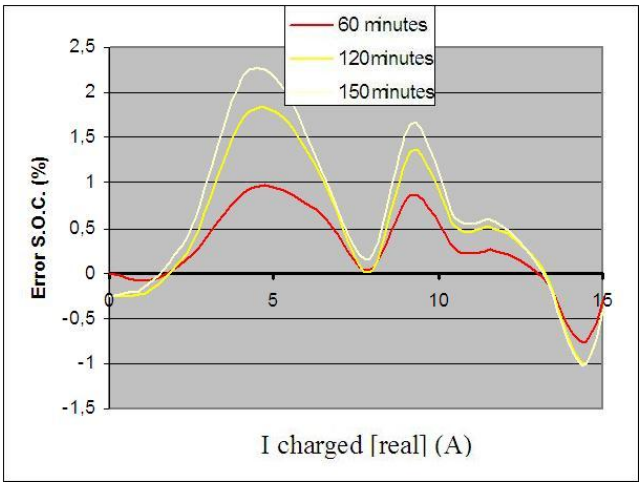

FIG. 10 SOC ERROR IN A CHARGE CYCLE. SOLUTION APPLIED.

Proving that without such restrictive boundaries in the number of program lines, the SOC estimator performs in a very positive way.

Conclusions

Test results which have been presented in this paper show that it is possible to develop a simpler SOC estimating system by designing an Ampere-hourmeter based structure through a microcontroller which controls both a data acquisition and a data processing system.

However, in order to achieve good accurate results in this SOC estimator, it becomes necessary the development of not only an accurate data acquisition

system, by the choice and design of a robust current sensing circuit, but also a data processing system which allows the necessary internal mathematical operations needed to obtain a non corrupted result.

Obtaining an accurate result of a lead-acid battery's SOC with an Ampere-hourmeter based system such as the presented in this paper is only possible if the used microcontroller holds enough memory to include the totality of the needed program lines. A good memory reservoir would even permit a future improvement of the presented results, by the addition of correcting factors for unbalancing disturbances such as the mentioned *I* calculations, the temperature effect on the battery or the ageing.

REFERENCES

Albércorp, "Predicting battery performance using internal cell resistance," Boca Raton, FL, 2003.

I. Buchmann, "Una mirada a las tecnologías emergentes capaces de estimar la capacidad de reserva de una batería," Cadex Electronics Inc., February 2004.

L. Closas, "Bateries i emmagatzemament d'energia elèctrica," Facultat de Nàutica de Barcelona, Universitat Politècnica de Catalunya, 2011, unpublished.

R. Pérez, "Lead-Acid Battery State of Charge vs. Voltage," Home Power 36, pp. 66-69, 1993.

N. Kong-Soon, H. Yao-Feng, M. Chin-Sien, and H. Yao-Ching, "An enhanced coulomb counting method for estimating state-of-charge and state-of-health of lead-acid batteries", 31st International Telecommunications Energy Conference, pp. 1-5, 2009.

D. Haifeng, W. Xuezhe, and S. Zechang, "Online SOC estimation of high-power lithium-ion batteries used on HEVs", IEEE International Conference on Vehicular Electronics and Safety, pp. 342-347, 2006.

C.R. Gould, C.M. Bingham, D.A. Stone, and P. Bentley, "New battery model and state-of-health determination through subspace parameter estimation and state-observer techniques", IEEE Transactions on Vehicular Technology, vol. 58, Issue 8, pp. 3905-3916, 2009.

B.S. Bhangu, P. Bentley, D.A. Stone, and C.M. Bingham, "Nonlinear observers for predicting state-of-charge and state-of-health of lead-acid batteries for hybrid-electric vehicles", IEEE Transactions on Vehicular Technology, vol. 54, Issue 3, pp. 783-794, 2005.

W. Peukert, "Elektrotechnische Zeitschrift" 20, pp. 20-21, 1897.

J. Denker, "An overview of the chemical reactions in a lead-acid battery and how they reputedly work" John S. Denker, 2004.

D. Doerffel, and S. Abu Sharkh, "A critical review of using the Peukert equation for determining the remaining capacity of lead-acid and lithium-ion batteries". Journal of Power Sources 155, pp. 395-400, 2006.

W. Guoliang, L. Rengui, Z. Chunbo, and C.C. Chan, "Apply a piece-wise Peukert's equation with temperature correction factor to NiMH battery state of charge estimation." Journal of Asian Electric Vehicles, (8) 2, pp. 1419-1423, 2010.

R. Nelson, "The basic chemistry of gas recombination in lead-acid batteries", The Minerals, Metals & Materials Society, 2001.

Microchip Technology Inc., "PIC 16F87X Data Sheet", 2001.

Honeywell Sensing Systems Ltd. "CSN series magnetorresistive closed loop current sensor", 2003.

Honeywell " Hall-effect open-loop current sensor application", 2007.

Niple Software – http://www.niplesoft.net.