FPGA Implementation of the Mix Algorithm for State-of-Charge Estimation of Lithium-Ion Batteries

Federico Baronti*, Roberto Roncella*, Roberto Saletti*, Walter Zamboni§
*Dip. di Ingegneria dell'Informazione - Università of Pisa, Italy
§Dip. di Ingegneria dell'Informazione, Ingegneria Elettrica e Matematica Applicata, Università degli Studi di Salerno, Italy
Email: f.baronti@iet.unipi.it

***Abstract*—This paper describes the hardware implementation of a model-based State-of-Charge (SoC) estimation algorithm for Lithium-ion batteries. SoC estimation is essential to evaluate the remaining runtime of the battery, as well as to enhance its safety and life expectancy. Model-based SoC estimation is a good solution to the problem, but only offline tests have been presented so far. In this work, the SoC estimation algorithm is implemented on an FPGA device, following an innovative and automatic development flow, which starts from a MATLAB/Simulink model of the algorithm. The SoC estimation hardware block is combined with a soft-core processor to form a System on a Programmable Chip. Experimental results obtained exerting the battery with a current profile that simulates its operation in an electric vehicle are presented and discussed.**

I. Introduction

Lithium-ion (Li-ion) batteries are becoming an attractive choice for energy storage in many industrial fields, such as electric transportation and utility grids [1]. The Li-ion batteries based on a Nickel Manganese Cobalt (NMC) cathode provide higher energy and power densities if compared to other variants making them particularly suitable for electric vehicles (EVs), at the expense of a higher initial cost and greater fragility. These batteries are always provided with a Battery Management System (BMS). A fundamental function of a BMS with advanced features is State-of-Charge (SoC) estimation [2], [3].

SoC indicates the residual charge of the battery and is usually expressed as a percentage of the battery capacity. SoC knowledge makes it possible to maintain the battery inside the operating condition optimal range and to evaluate the runtime of the system powered by the battery (*e.g.*, the residual driving range of an EV). The simplest approach to estimate the SoC is considering the battery an ideal charge reservoir (*i.e.*, a capacitor). In this way, the charge stored in the battery can be tracked by integrating the current flowing in or out of the battery (Coulomb Counting method). This approach, however, is very sensitive to measurement errors. Particularly, any offset of the current sensor may lead to large SoC errors over time, because of the current integration. In addition, Coulomb Counting does not account for the non unitary charge efficiency of the battery and requires to be initialized with the correct initial SoC value.

Another simple method to estimate the SoC is to make use of the relationship between the SoC and the Open Circuit Voltage (OCV). The SoC-OCV relationship is indeed almost invariant in a wide operating temperature range and with battery aging [4]. Unfortunately, the OCV measurement requires the battery to be in the steady state, a condition that is reached only after a long time (often many minutes or even hours) with no load current. Thus, this approach is not suited for real time SoC estimation, when the battery is continuously charged or discharged at high currents, as it happens in an EV. Model-based algorithms (such as Extended Kalman and Particle filters [5]–[7] and the Mix algorithm [8], [9]) have been introduced to take into account the battery dynamics and proved to be suitable for online SoC estimation in an EV. However, almost all the works in this field have been carried out with lab experiments, by measuring the voltage and current of the battery under test with dedicated equipment and then by processing the acquired data offline with software environments such as MATLAB/Simulink or LabVIEW.

The aim of this work is to report on the preliminary implementation of an enhanced version of the SoC estimation Mix algorithm on a standalone platform consisting of a Field Programmable Gate Array (FPGA)-based board. Thanks to the intrinsic hardware parallelism and the deterministic "execution" of concurrent tasks, an FPGA is a viable solution for the implementation of SoC estimation of an EV battery consisting of many series-connected cells. In fact, SoC can vary from cell to cell because of the differences in the cell characteristics. Thus, SoC estimation at cell level rather than at battery level is desirable for an accurate evaluation of the battery runtime and charge balancing of the battery cells. The hardware block that implements the SoC estimation can also be combined with a processor inside the same FPGA, creating a System on a Programmable Chip (SoPC) that provides an effective platform for the realization of a BMS with advanced functions.

This paper is organized as follows. Section II presents the Mix algorithm for SoC estimation and the cell model used in the algorithm. The FPGA implementation of the algorithm is discussed in the next Section. Experimental results are then described in Section IV before drawing some conclusions.

II. Model-Based SoC Estimation

The basic idea underlying a model-based SoC estimation algorithm is to use a *cell model* in which SoC is one of the state variables that can be estimated by an observer, as it is shown in Fig. 1. The crucial element of this approach is the

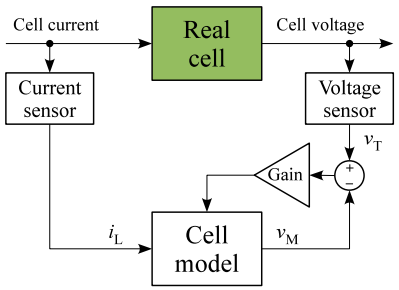

Figure 1. Schematic representation of a model-based SoC estimation algorithm.

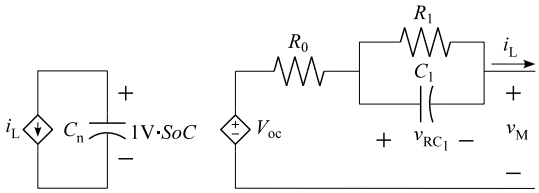

Figure 2. Equivalent electrical model with one R-C group to model the cell relaxation phenomenon.

cell model, which must be capable of accurately reproducing the cell behavior in a wide range of its operating conditions.

A. Cell Model

Equivalent electrical models maintain a connection to the physical behavior of the battery, while keeping the computational complexity affordable, if compared to purely mathematical or electrochemical models [10]. Fig. 2 shows a possible representation of an electrical equivalent model, which consists of two sections. On the left-hand side, a linear capacitor accumulates the charges flowing in or out of the battery. The numerical value of its capacitance C_n is equal to the cell capacity (expressed in Coulomb) divided by 1 V. SoC is thus the numerical value of the voltage on the capacitor. On the right-hand side, the model output voltage v_M is obtained as the sum of three terms (with the appropriate signs): the open circuit voltage V_{OC}, a purely resistive voltage $R_0 i_L$ (where i_L is the cell current, as shown in Fig. 2), and a relaxation voltage v_{RC_1} (with time constant $\tau_1 = R_1 C_1$).

This model is capable of faithfully reproducing the dynamic cell behavior assuming that the model parameters are properly identified for the specific cell in the full range of the operating conditions, *i.e.,* SOC, temperature and load current of the cell [11]. In fact, the model parameters significantly vary with the operating condition. Figure 3 shows the SoC-OCV curve and the model parameter measured on a 1.5 A h NMC cell, using pulsed current tests [11], [12]. Specifically, each charge or discharge current pulse has 1.5 A amplitude (or equivalently 1 C–rate) and 3 min duration, thus determining a 5 % variation of the cell SoC. The tests were performed at room temperature.

B. Mix Algorithm

The *Mix* algorithm is a simple implementation, compared to Extended Kalman filters, of a model-based SoC estimation method. Its block diagram is shown in Fig. 4. The light blue

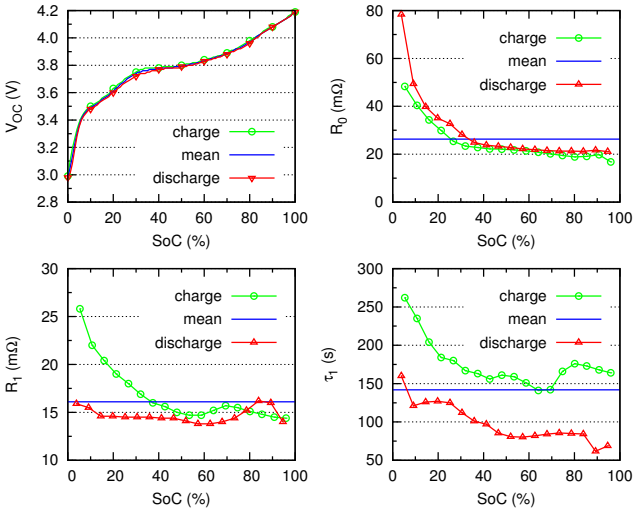

Figure 3. SoC-OCV curve and model parameters measured on a 1.5 A h NMC cell at room temperature.

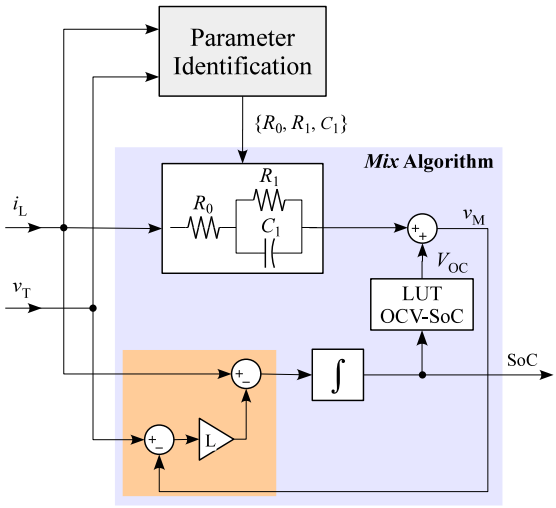

Figure 4. Block diagram of the *Mix* algorithm.

box encloses the cell model [*i.e.*, the electric model shown in Fig. 2, in which the V_{OC} generator is replaced by an OCV-SoC Look-Up Table (LUT)]. The orange box contains the block comparing the model output v_M and the measured cell voltage v_T. The generated error signal is amplified by the observer *gain* L and subtracted to the measured cell current i_L. The resulting current signal is then integrated over time to produce SoC as in the conventional Coulomb Counting method.

Consequently, the *Mix* algorithm can be seen as the enhancement of the Coulomb Counting method by adding the SoC estimation through OCV, where the latter is dynamically obtained from the cell equivalent model. Thanks to the feedback loop, the sensitivity to uncertainties over the SoC initial value and current measurements (mainly the drifting offset of the current sensor) affecting the Coulomb Counting method can be reduced [8]. However, such a valuable result relies on

Table I
ERROR RESPONSE TO DIFFERENT ERROR SOURCES

Error source	Error response E(s)	Steady-state Error
$\mathrm{SoC_{err}}$	$\frac{C_\mathrm{n} SoC_\mathrm{err}}{C_\mathrm{n} s + L\alpha_1}$	0
V_err	$\frac{V_\mathrm{err}}{s} \frac{L}{C_\mathrm{n} s + L\alpha_1}$	$\frac{V_\mathrm{err}}{\alpha_1}$
I_err	$\frac{I_\mathrm{err}}{s} \frac{LZ(s)-1}{C_\mathrm{n} s + L\alpha_1}$	$I_\mathrm{err} \frac{L(R_0+R_1)-1}{L\alpha_1}$

the capability of the model to reproduce the cell behavior in an accurate way. This is shown in [9], where the model parameters variation with the cell operating conditions, namely SoC, temperature and current, is considered with 3-Dimensional LUTs. Unfortunately, using LUTs has disadvantages. In fact, these LUTs are determined with time-consuming offline tests (as those used to extract the parameters shown in Fig. 3). The tests should be repeated for any cell of the battery to account for variations in cell manufacturing. Furthermore, constant value LUTs cannot model the variation of the parameters with battery aging.

An attractive alternative to the use of LUTs or fitting functions to model parameter variations is to implement the cell model with constant parameters and then update their values in real-time by means of a *Parameter Identification* method (see Fig. 4) [13]–[15]. Thus, the hardware realization of this approach consists of the implementation of the *Mix* algorithm, with constant values of the cell model parameters, and the *Parameter Identification* block. The following sections describe the implementation in a low-cost FPGA-based board and the related testing of the *Mix* algorithm with constant values of the cell model parameters. Before moving on to the FPGA design, let us briefly discuss the choice of the observer gain *L*.

C. Choice of the Observer Gain L

The observer gain *L* shown in Fig. 4 can be calculated following the procedure described by Codeca et al. in [16], which is based on the linearization of the OCV-SoC relationship (*i.e.*, $V_\mathrm{OC} = \alpha_1 \mathrm{SoC} + \alpha_0$), and the evaluation of the system step response *E*(*s*) in the Laplace domain with respect to different error sources, such as a bad SoC initialization value $\mathrm{SoC_{err}}$, an error in the cell voltage and current measurements, V_err and I_err, respectively. The steady-state value of the SoC errors caused by the different error sources is obtained from *E*(*s*) using the final value theorem. This is shown in Table I, where $Z(s) = R_0 + \frac{R_1}{1+R_1 C_1 s}$ is the small-signal output impedance of the linearized cell model of Fig. 2.

It is worth noting that the *Mix* algorithm is capable of fully correcting a bad SoC initialization independently of *L*. In contrast, an error in the cell voltage measurement (or equivalently in the output of the cell model) leads to a non-zero steady-state error, which is independent of *L* being inversely proportional to the slope α_1 of the OCV-SoC curve in the SoC operating point. Finally, the SoC steady-state error due to a static offset in the cell current measurement can be cancelled with an appropriate choice L_opt of the *gain*.

$$L_\mathrm{opt} = \frac{1}{R_0 + R_1} \tag{1}$$

According to (1), L_opt is determined by the value of the resistive components of the cell model and thus is not constant but varies when new values of the model parameters are identified.

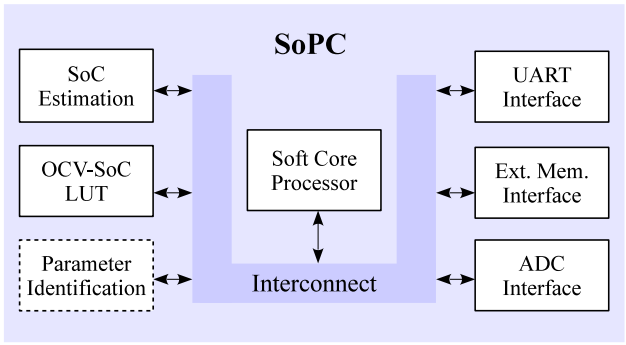

Figure 5. Block diagram of the System on a Programmable Chip (SoPC) implemented in the FPGA device.

III. FPGA IMPLEMENTATION

FPGA devices provide the valuable support for a hardware/software partitioning of the BMS functions, when compared to discrete microcontrollers or digital signal processors (DSPs). Specifically, more computationally-intensive tasks, such as SoC estimation and parameter identification, can be implemented by dedicated hardware blocks that exploit the power of customized logic. These blocks can be combined with an embedded processor core that can either be integrated (hard-core) in the FPGA device or hardware programmed using the programmable logic resources (soft-core). In this way, rather complex SoC estimation algorithms can be implemented with very optimized hardware modules acting as co-processors of the embedded CPU. The processor is thus released from heavy computation loads and can reliably perform the other BMS functions, such as battery monitoring, battery protection, communicating with other systems, etc.

Figure 5 shows the architecture of the SoPC programmed in an Altera Cyclone IV family FPGA. The *SoC Estimation* module implements the *Mix* algorithm described in the previous section. The tabular values of the OCV-SoC relationship are stored in an on-chip ROM (*OCV-SoC LUT*), whereas the cell model parameters R_0, R_1 and C_1 are generated by the *Parameter Identification* block. This block simply provides a constant value for each parameter in this paper. It will be replaced by the full complex identification function in future developments. The constant parameter values are computed by averaging the values measured in the entire SoC range. The *OCV-SoC LUT* contains 100 OCV values (1 % SoC resolution) obtained by averaging and interpolating the measured charge/discharge OCV values. The used values for OCV, R_0, R_1, and C_1 are the *mean* curves in Fig. 3.

The modules composing the SoPC communicate to each other with standard memory-mapped interfaces connected by means of the *Interconnect* infrastructure as shown in Fig. 5. In this way, the values of the cell voltage and current acquired by the *ADC Interface* are available to the *SoC Estimation*, the *Parameter Identification,* and the *Soft Core Processor* blocks. The latter is the economy variant of the Altera Nios II core. As the aim of this work is to implement and validate the *Mix* algorithm, the processor functions are basically configuring and supervising the various hardware modules. It also communicates to a host PC via the *UART Interface* for configuration and data logging purposes. The *Ext. Mem. Interface* is used to connect to an SDRAM external memory.

An important aspect of the applied FPGA design flow is the automatic generation of the hardware description (HDL code) of the *SoC Estimation* block carried out with the Altera DSP Builder tool, starting from a Simulink model of the algorithm. Calculations are performed using single-precision floating-point arithmetic. Then, the generated HDL code has been provided with a standard interface and used as custom component of the SoPC system. As all the components have standard interfaces, the SoPC system can easily be composed with the Altera Qsys tool, which automatically generates the HDL files related to the used components and their interconnections. Finally, the HDL description is synthesized using the Altera Quartus II tool and the conventional FPGA back-end flow. This approach significantly speeds up the hardware implementation of an algorithm, as the latter can be developed using a high level tool such as Simulink, from which the FPGA programming bitstream can be obtained with automatic steps. In this way, it is possible to take advantage of the hardware parallelism provided by FPGAs, without the need of time-consuming efforts to manually translate an algorithm into an HDL code.

IV. Validation

A. Experimental test-bed

The above described SoPC system has been programmed in an Altera Cyclone IV EP4CE22 FPGA device mounted on a low-cost development board (Terasic DE0-Nano), which also includes an 8-channel 12-bit A/D converter. Two channels of the ADC are used to acquire the voltage and the current of the Lithium-ion cell under test, a 1.5 A h NMC cell (Kokam SLPB723870H4), which was preliminary characterized by performing pulsed current tests as described in Section II-A. The cell current is sensed by an off-the-shelf Hall sensor (DHAB s/25), commonly used in automotive applications. To increase its sensitivity and make it suitable for the used 1.5 A h cell, ten windings of the conductor carrying the cell current have been sensed. The sampling rate of the *SoC Estimation* block inputs is set to 10 Hz, while the system clock frequency is 50 MHz (all the SoPC blocks are synchronous with the system clock).

The current flowing in or out the cell is imposed by a highly accurate source-meter unit (Keithley SMU 2420), which is controlled by a LabVIEW application running on a PC. The

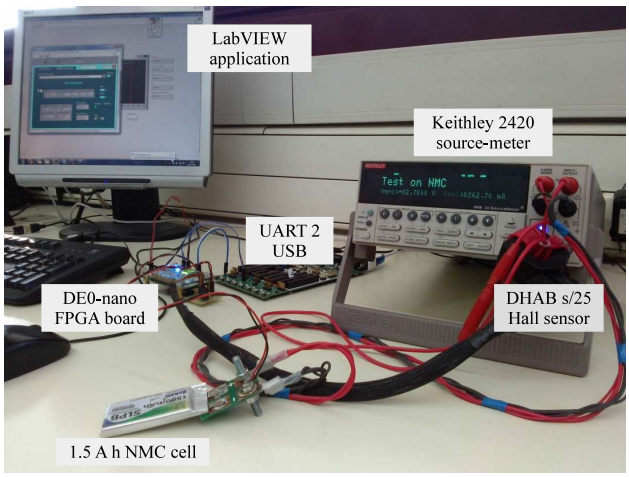

Figure 6. Photograph of the experimental test-bed.

cell current measured by the source-meter is integrated over time and the resulting SoC is used as the reference value to which compare the SoC calculated by the implemented estimation algorithm. Figure 6 shows a photograph of the experimental test-bed. The DE0-nano board is also connected to the PC via an UART link, so that the cell voltage and current values acquired by the ADC, as well as the output of the *SoC Estimation* block, are available to the PC. In this way, it is possible to compare the estimated SoC with the reference one. Further, we can use the logged voltage and current values as input of the Simulink model and verify that it produces the same SoC value as that generated in the FPGA.

B. Test Current Profile

A crucial aspect in evaluating the performance of a SoC estimation algorithm is to exert the battery with a current profile relevant for the target application of the BMS. For this purpose, we considered the Urban Dynamometer Driving Schedule (UDDS) [17], *i.e.*, the speed profile defined by the U.S. Environmental Protection Agency that simulates a urban route of 12.07 km with frequent stops. Given the speed profile of the EV, we computed the current flowing in or out the battery using a simple model of the vehicle, as described in [14]. As the battery under test is a small size unit, the UDDS derived current profile has been expressed in C-rate of the battery, so that the current values are properly scaled to the battery size available. Another simplification of the experiment comes from the Keithley 2420 source-meter that does not allow continuous changes of the current as the calculated profile would require. The highly variable *Simulated* current profile has been *Simplified* into a step-wise profile by averaging the current values in 30 s windows, so that it can be generated by the source-meter. The UDDS speed profile and the *Simulated* and *Simplified* current profiled (expressed in C-rate) are shown in Fig. 7.

C. Experimental Results

As each UDDS cycle determines approximately a 7.5 % variation of the SoC, 12 consecutive UDDS cycles are per-

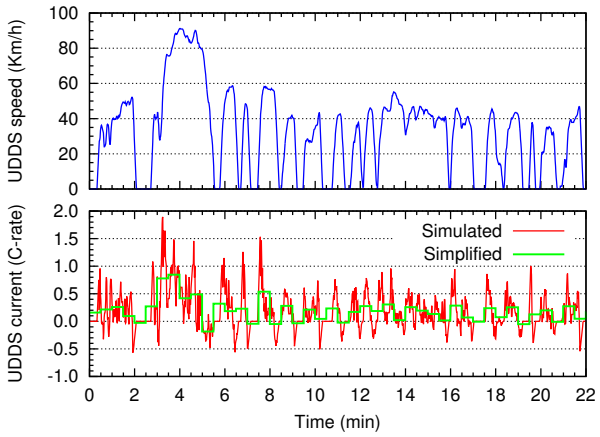

Figure 7. UDDS speed and battery current profiles expressed in C-rate.

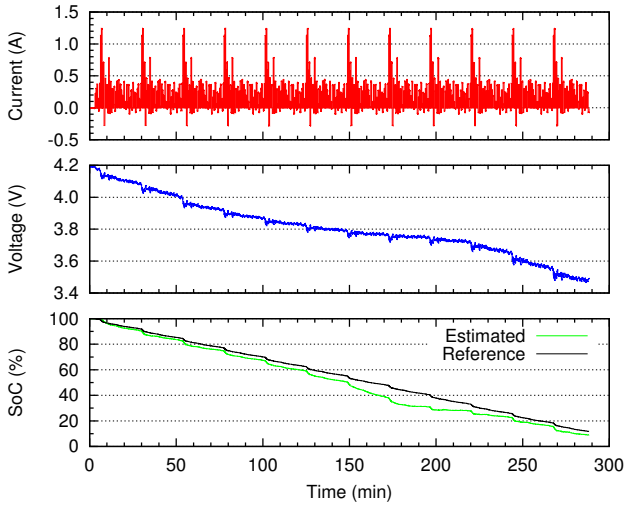

Figure 8. Signals logged during the test with 12 consecutive UDDS cycles.

formed to span the SoC range from 100 % to 10 %. Each test starts after a full charge of the cell. Figure 8 shows the result of applying 12 consecutive UDDS simplified cycles to the cell. Specifically, the upper plots show the cell current and voltage, as acquired by the FPGA, while the bottom plot compares the *estimated* SoC from the FPGA system with the *reference* one calculated from the source-meter data. We note that the *estimated* SoC is in good accordance with the *reference* one. Indeed, the SoC rms error is 5 %, as reported in Table II. In more detail, the absolute SoC error is below 5 % except in the SoC interval 50 % to 20 %, where the absolute error goes up to 11.5 %. This behavior can be ascribed to the slope of OCV-SoC curve that is very low in the 25 % to 50 % SoC range (see Fig. 3). In fact, the error in SoC estimation caused by errors in the cell voltage measurement or in the model output is inversely proportional to the slope of the OCV-SoC curve (see Table I).

To further investigate this phenomenon, we performed another test in which the UDDS cycles are separated by 1 h

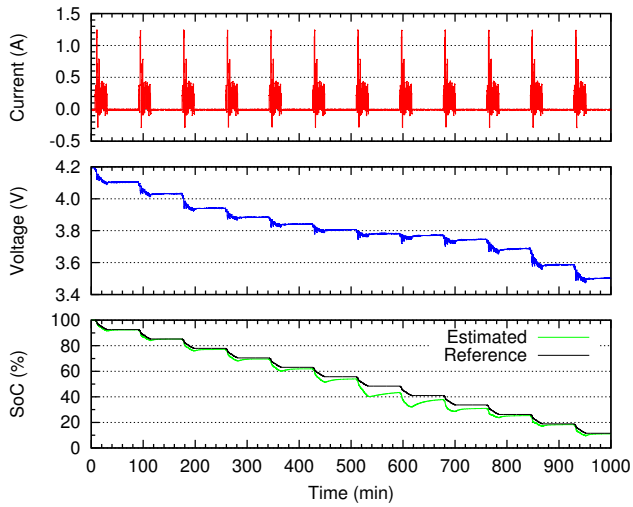

Figure 9. Signals logged during the test with 12 consecutive UDDS cycles separated by 1 h pauses.

pauses in which the cell current is zero. As shown in Fig. 9, the SoC estimation error decreases during the pauses leading to lower rms and maximum SoC error values compared to those obtained in the test without pauses (see Table II). This implies that the SoC error is mainly caused by a dynamic error of the cell model output. As we used a constant value for the cell model parameters, the capability of the model to accurately reproduce the dynamic behavior of the cell voltage is reduced. Such a voltage error is highly emphasized in the SoC region with the lowest value of the OCV-SoC curve slope. Thus, we are confident that the performance of the online implementation of the *Mix* algorithm could be improved by the online identification of the cell model parameters. However, as shown in Fig. 9, there is also a residual static error, which is related to the static error of the cell model caused by the fact that the OCV LUT is filled with the mean value of the charge/discharge OCV values. In fact, even if in NMC cell the hysteresis is not pronounced as in other Li-ion batteries [18], the charge curve lies slightly above the discharge one. Consequently, the output of the cell model during discharge overestimates the OCV during discharge, which causes the *Mix* algorithm to underestimates the SoC especially when the slope of the OCV-SoC curve is low.

Finally, we repeated the test with 12 consecutive UDDS cycles after introducing a 100 mA offset in the current sensor. Figure 10 shows that the SoC estimated by the *Mix* algorithm in presence of an offset is very similar (see also Table II) to that achieved without offset (*i.e.*, when the intrinsic offset of the used Hall sensor is zeroed before starting the test). This experiment proves the expected capability of the algorithm to cancel the current offset effects. Instead, we note that the introduced offset causes the conventional Coulomb Counting method (CC in Fig. 10) to produce very unreliable results.

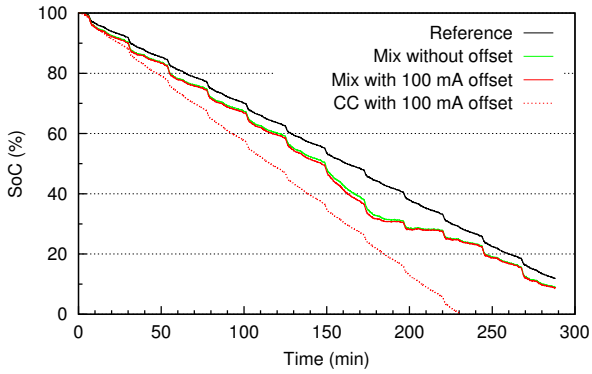

Figure 10. SoC estimation in presence of an offset in the current measurement.

Table II
SoC errors in the different tests

Test	rms error	Max error
12x UDDS cycles	5.0 %	11.5 %
12x UDDS cycles w/pause	3.1 %	9.4 %
12x UDDS cycles w/ current offset	5.6 %	12.8 %

V. Conclusions

The FPGA implementation of a model-based SoC estimation algorithm, specifically the *Mix* algorithm, has been described in this paper. The SoC estimation algorithm has been translated into a hardware block starting from its Simulink model applying a tool for automatic hardware description language (HDL) code generation, thus speeding up the development process. The SoC estimation block is combined with a soft-core processor to potentially realize a battery management system (BMS) with advanced estimation functions as System on a Programmable Chip (SoPC). The SoC estimation hardware block can be used in time-multiplexing to perform the SoC estimation of multiple cells of a battery, being the sampling frequency of the SoC estimation inputs much lower than the system clock frequency.

The SoPC has been fitted on an Altera Cyclone IV FPGA device mounted on a low-cost development board, and subjected to several tests to verify the performance of the SoC estimation algorithm. First, the results obtained by the hardware block overlap the output of the Simulink model when it is fed with the same cell voltage and current signals, as acquired by the FPGA. Second, the performances achieved by the embedded system implementation of the *Mix* algorithm are encouraging and comparable with those obtained in previous works, when SoC estimation was performed offline on a PC. This time the SoC estimation is running on an FPGA and its inputs are acquired by off-the-shelf sensors. Future work will be the implementation of the parameter identification hardware block, which can improve the accuracy of SoC estimation by updating the cell model parameters in real time, with the final aim of obtaining a low-cost portable BMS for Lithium-ion batteries with advanced functions.

Acknowledgement

The authors would like to acknowledge the valuable contribution of Francesco Mengali and Sandro Rosi to the work described in this paper, and of Pierluigi Lo Muzio for the fruitful discussions on the Altera tools utilization.

References

[1] M. S. Whittingham, "History, Evolution, and Future Status of Energy Storage," *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1518–1534, May 2012.

[2] H. Rahimi-Eichi, U. Ojha, F. Baronti, and M.-Y. Chow, "Battery Management System: An Overview of Its Application in the Smart Grid and Electric Vehicles," *IEEE Ind. Electron. Mag.*, vol. 7, no. 2, pp. 4–16, Jun. 2013.

[3] L. Lu, X. Han, J. Li, J. Hua, and M. Ouyang, "A review on the key issues for lithium-ion battery management in electric vehicles," *Journal of Power Sources*, vol. 226, pp. 272–288, Mar. 2013.

[4] L. Zhong, C. Zhang, Y. He, and Z. Chen, "A method for the estimation of the battery pack state of charge based on in-pack cells uniformity analysis," *Applied Energy*, vol. 113, no. 0, pp. 558–564, Jan. 2014.

[5] G. L. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 2. Modeling and identification," *Journal of Power Sources*, vol. 134, no. 2, pp. 262–276, 2004.

[6] R. Restaino and W. Zamboni, "Comparing particle filter and extended kalman filter for battery State-Of-Charge estimation," in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*. IEEE, Oct. 2012, pp. 4018–4023.

[7] ——, "Rao-blackwellised particle filter for battery state-of-charge and parameters estimation," in *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, Nov. 2013, pp. 6783–6788.

[8] F. Codeca, S. M. Savaresi, and G. Rizzoni, "On battery State of Charge estimation: A new mixed algorithm," in *2008 IEEE International Conference on Control Applications*. IEEE, Sep. 2008, pp. 102–107.

[9] F. Baronti, G. Fantechi, L. Fanucci, E. Leonardi, R. Roncella, R. Saletti, and S. Saponara, "State-of-Charge Estimation Enhancing of Lithium batteries through a Temperature-Dependent Cell Model," in *2011 Int. Conf. Appl. Electron.*, Pilsen, 2011, pp. 29–33.

[10] M. Chen and G. Rincon-Mora, "Accurate Electrical Battery Model Capable of Predicting Runtime and I-V Performance," *IEEE Transactions on Energy Conversion*, vol. 21, no. 2, pp. 504–511, Jun. 2006.

[11] F. Baronti, G. Fantechi, E. Leonardi, R. Roncella, and R. Saletti, "Enhanced model for Lithium-Polymer cells including temperature effects," in *IECON 2010 - 36th Annu. Conf. IEEE Ind. Electron. Soc.* IEEE, Nov. 2010, pp. 2329–2333.

[12] S. Abu-Sharkh and D. Doerffel, "Rapid test and non-linear model characterisation of solid-state lithium-ion batteries," *Journal of Power Sources*, vol. 130, no. 1-2, pp. 266–274, May 2004.

[13] H. Rahimi-Eichi, F. Baronti, and M.-Y. Chow, "Modeling and online parameter identification of Li-Polymer battery cells for SOC estimation," in *2012 IEEE International Symposium on Industrial Electronics*. IEEE, May 2012, pp. 1336–1341.

[14] F. Baronti, W. Zamboni, N. Femia, H. Rahimi-Eichi, R. Roncella, S. Rosi, R. Saletti, and M.-Y. Chow, "Parameter identification of Li-Po batteries in electric vehicles: A comparative study," in *2013 IEEE Int. Symp. Ind. Electron.* IEEE, May 2013, pp. 1–7.

[15] H. Rahimi-Eichi, F. Baronti, and M.-Y. Chow, "Online Adaptive Parameter Identification and State-of-Charge Coestimation for Lithium-Polymer Battery Cells," *IEEE Trans. Ind. Electron.*, vol. 61, no. 4, pp. 2053–2061, Apr. 2014.

[16] F. Codeca, S. M. Savaresi, and V. Manzoni, "The mix estimation algorithm for battery State-of-Charge estimator- Analysis of the sensitivity to measurement errors." in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. IEEE, Dec. 2009, pp. 8083–8088.

[17] "Urban Dynamometer Driving Schedule (UDDS)." [Online]. Available: http://www.epa.gov/nvfel/testing/dynamometer.htm

[18] M. Petzl and M. A. Danzer, "Advancements in OCV Measurement and Analysis for Lithium-Ion Batteries," *IEEE Transactions on Energy Conversion*, vol. 28, no. 3, pp. 675–681, Sep. 2013.