System on Chip Battery State Estimator: E-Bike Case Study

Rocco Morello, Roberto Di Rienzo, Federico Baronti, Roberto Roncella and Roberto Saletti
Dip. di Ingegneria dell'Informazione, Università di Pisa, Pisa, Italy

***Abstract*—This paper discusses the hardware implementation and experimental validation of a model-based battery state estimator. The model parameters are identified online using the moving window least squares method. The estimator is implemented in a field programmable gate array device as a hardware block, which interacts with the embedded processor to form a system on a chip battery management system (BMS). As a case study, the BMS is applied to the battery pack of an e-bike. Road tests show that the implemented estimator may provide very good performance in terms of maximum and rms estimation errors. This work also proposes a new methodology to assess the performance of a battery state estimator.**

I. Introduction

Lithium-ion batteries allow the implementation of light on-board energy storage systems (ESS), thanks to their high power and energy densities. The ESS must be equipped with a battery management system (BMS) in order to ensure a reliable and safe operation of the battery. The BMS monitors and controls the charge and discharge phases by estimating the battery state. The most important state variables are the battery state of charge (SOC) and state of health (SOH). SOC is related to the residual charge stored in the battery and SOH quantifies its degradation in terms of capacity fading and the increase of the internal resistance [1], [2].

The BMS executes algorithms that use quantities directly observable on the cells, *i.e.*, their voltage, current and temperature. Many algorithms have been proposed for SOC estimation, starting from the Coulomb Counting (CC), which is based on the integration over time of the battery current. The integration result is normalized with respect to the cell capacity. The CC must be initialized with the correct value of the SOC and the current must be measured by a high accuracy sensor, in order to avoid the intrinsic divergence in the integration. Another simple method is based on the open circuit voltage (V_{OC}) measurement. In fact, V_{OC} mainly depends on SOC and slightly on other factors, such as temperature, ageing and current rate [3]. However, the cell voltage relaxes to V_{OC} a long time after the cell current is interrupted. Thus, this technique cannot be used in highly dynamic systems [4].

High accuracy and reliability are required in automotive applications, in order to guarantee the safety standard and a good estimation of the vehicle autonomy. Furthermore, the algorithms must be executed in real-time in an embedded system, like the BMS. Model-based algorithms, such as the Extended Kalman Filter (EKF) [5], [6], the Mix Algorithm [7] and the Adaptive Mix Algorithm (AMA) [4] are adopted. A model-based technique is a closed-loop method in which the model is used to correct the estimated SOC, by comparing the voltage predicted by the model with the measured one. Various kinds of model can be used [8], but an Electrical Circuit Model (ECM) has some benefits when used in a real-time embedded system [9]. The accuracy of the SOC estimate depends on the ECM capability to predict the cell voltage, in all the operating conditions and for the entire battery life. A good approach is to track the variation of the operating condition online, which is the solution adopted in the EKF and AMA techniques.

The AMA uses an enhanced version of the Mix Algorithm, in which SOC estimation is corrected by using an ECM where the parameters are updated online by using the Moving Window Least Squares (MWLS) method [10], [11], instead of taking them constant, like in the original algorithm. The performance of this algorithm has been evaluated by comparing it to the EKF method [4], [12], which is the most used in this field. The comparison, carried out with current and voltage profiles obtained from the Urban Dynamometer Driving Schedule (UDDS) driving cycle, defined by the U.S. Environmental Protection Agency [13], shows that both methods provide good estimation of both parameters and SOC. Both EKF and AMA are suitable for an embedded system implementation.

An attractive solution for industrial and automotive applications is the use of a low-cost field-programmable gate array (FPGA). The estimators can be implemented on the FPGA as hardware accelerators beside the processor, which is fully dedicated to the other BMS functionalities. A first hardware implementation of the non adaptive version of the Mix algorithm is reported in [14]. A hardware-in-the-Loop (HiL) simulation framework used to test and compare the AMA and EKF, implemented in hardware on an FPGA, is presented in [15]. In this framework, a multi-cell battery is simulated using an accurate model.

An experiment not common in literature that can provide final results is to test the estimators in a real application. This allows an effective measurement of the estimator performance, as well as the analysis of the issues arising from its integration in a real system. The aim of this paper is to present the results obtained using the AMA in a real application, *i.e.*, on an e-bike. A simple BMS provided with the AMA battery state estimator has been implemented as a system on chip on an Altera Cyclone® V. The algorithm details and the hardware/software partitioning of the system are discussed in Section II. The experimental testbed description is reported In Section III, while Section IV describes the methodology applied to assess

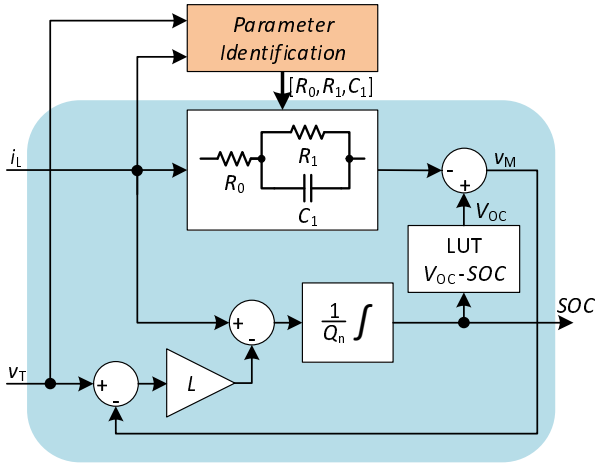


Fig. 1. Adaptive Mix Algorithm block diagram.

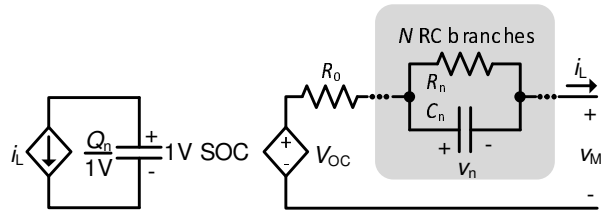


Fig. 2. Electric circuit model.

the estimator performance. The results obtained from road tests are discussed in Section V and finally conclusions are drawn in Section VI.

II. State Estimator

A. The Adaptive Mix Algorithm

The AMA is based on the combination of the Mix Algorithm and the MWLS method used with the AutoRegressive eXogenous (ARX) structure of the ECM. A basic block diagram of AMA is shown in Fig. 1. The Mix Algorithm is a simple model-based SOC estimation method (blue square in Fig. 1), which uses a constant parameter ECM in order to correct the estimated SOC variable. A general representation of the ECM is shown in Fig. 2 and a single RC branch version is used in this work. The voltages v_{n} on the RC branches model the relaxation effects. A single RC group reduces complexity maintaining good accuracy, particularly when fast transients are dominant. The cell terminal voltage v_{M} is generated as a sum of the voltage v_1, the open-circuit voltage V_{OC} and the voltage across the internal resistance $R_0 i_{\mathrm{L}}$. The V_{OC}-SOC relationship is modelled as a controlled generator $V_{\mathrm{OC}} = f(SOC)$, implemented with a Look-up Table (LUT). The left-hand side of the ECM models the cell capacity and SOC. The latter is expressed as Q/Q_{n}, where Q is the residual cell charge and Q_{n} is the normalization value corresponding to the maximum charge that can be stored in the cell, expressed in Coulomb.

The model cell voltage v_{M} is subtracted to the measured cell voltage v_{T} yielding an error signal, which is firstly amplified by the observer gain L and then used to correct the measured cell current i_{L}. Finally, SOC is the result of the integration over time of the corrected current signal. The observer gain L is chosen to strongly reduce the effects due to bad SOC initialization and current measurement uncertainty [14].

The estimation accuracy depends on the capability of the ECM to reproduce the cell behaviour. For this reason, the ECM parameters are updated online by the Parameter Identification block, which implements the MWLS algorithm. In this way, the algorithm takes into account the parameter variations due to the battery ageing, the operation conditions (as cell SOC and temperature) and the manufacturing process tolerances.

The ARX structure can be obtained starting from the cell time-domain state space model:

$$\begin{cases} \dfrac{dSOC}{dt} = -\dfrac{i_{\mathrm{L}}}{Q_{\mathrm{n}}} \\ \dfrac{dv_1}{dt} = -\dfrac{v_1}{\tau_1} + \dfrac{i_{\mathrm{L}}}{C_1} \\ v_{\mathrm{M}} = V_{\mathrm{OC}} - R_0 i_{\mathrm{L}} - v_1 \end{cases} \tag{1}$$

where $\tau_1 = R_1 C_1$. The cell operating point slowly changes over time, so that the model can be linearised around it, considering the parameters constant in the identification window. The V_{OC}-SOC relationship is thus approximated by a piecewise linear curve $V_{\mathrm{OC}} = \alpha_0 + \alpha_1 SOC$, where α_0 and α_1 depend on the operating point. The discrete-time transfer function is obtained by the application of the bilinear transform to the transfer function from the current input to the cell voltage output, as obtained from (1). Then, the discrete-time relationship between the input and output samples becomes:

$$\begin{aligned} y(k) = &- a_1 y(k-1) - a_2 y(k-2) + \alpha_0(1 + a_1 + a_2) \\ &+ b_0 u(k) + b_1 u(k-1) + b_2 u(k-2) \end{aligned} \tag{2}$$

which is a second order ARX model of the cell. The coefficients of the ARX model depend on the ECM parameters and we can simply verify that $1 + a_1 + a_2 = 0$, so (2) is reduced to:

$$\begin{aligned} y(k) - y(k-2) = &a_1(y(k-2) - y(k-1)) + b_0 u(k) \\ &+ b_1 u(k-1) + b_2 u(k-2) \end{aligned} \tag{3}$$

Eq. (3) is used to build an overdetermined system by using the current and voltage samples within a given time window. This window is shifted in time to track the parameter variations during the cell operation. In this way, a different system is obtained at every shift, which is solved by the Least Squares (LS) method. Finally, the resulting vector $[a_1, b_0, b_1, b_2]$ is used to compute the ECM parameters $[R_0, R_1, C_1]$.

B. Hardware implementation

The AMA is fully developed as a hardware module and implemented on an Altera Cyclone® V SoC 5CSXFC6D6F31C6N device, by using the Altera SoCkit development board. The hardware implementation enables a high

TABLE I
ESTIMATOR RESOURCE USAGE AND PERFORMANCE

Logic utilization (in ALMs)	17244/41910 (41 %)
Variable-precision DSP Block	36/112 (32 %)
Memory bits	723 Kb/5530 Kb (13 %)
Execution time	61 µs (@50 MHz)

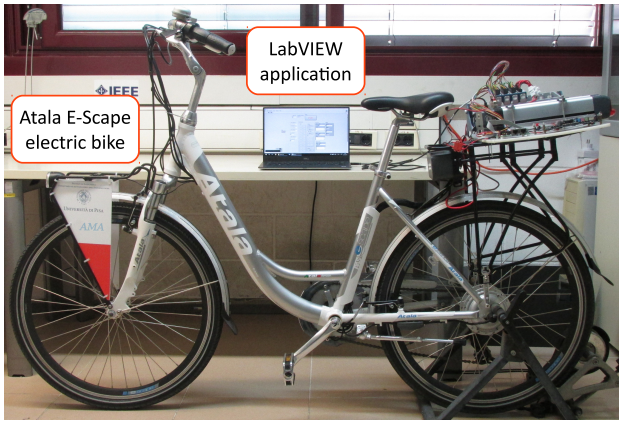
throughput and low latency estimation in an embedded system, in spite of the computational complexity of the algorithm.

The algorithm is described as a Simulink model and then automatically synthesized to a low-level hardware description, by using the Altera DSP Builder tool. This development environment optimizes the hardware module performance and resource usage depending on the tool parameter settings. The module is also provided with a Memory Mapped (MM) interface. The latter consists of input and output registers which are used by the companion processor to write the algorithm inputs and to read the computed results. The estimator FPGA resource usage and execution time are shown in Table I. The time required to estimate both SOC and the parameters, for a clock frequency of 50 MHz, is very short. Therefore, the BMS is capable of estimating the state of a large number of cells by using the estimator in Time Division Multiplexing (TDM) across multiple cells. This feature is very useful in high-power applications, where the batteries consist of hundreds of cells.

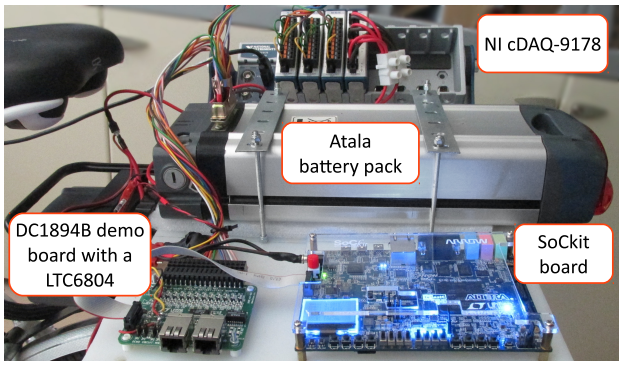
III. CASE STUDY

The demonstration system is built on an electrically assisted bike, equipped with a 10 cell battery pack. This solution allows us to verify the estimator on more than one cell in real road tests. Two photos of the demonstrator are shown in Fig. 3. The electric bike is an Atala E-Scape (see Fig. 3(a)), a city bike with aluminum frame. The bike is equipped with a brushless DC motor with a nominal power of 250 W. The battery pack consists of 10 Lithium-ion series-connected cells, yielding a nominal voltage of 36 V and a rated capacity 10 A h. The entire setup, including the battery pack, was arranged on the bike rack, as shown in Fig. 3(b). A 14-pin connector allows us to access the battery cells' terminals. A Hall effect current sensor from Allegro Microsystems is used to measure the battery current. Its full scale range is ±12.5 A, with an internal resistance of 1.2 mΩ and a sensitivity of 110 mV/A (for a supply voltage of 3.3 V). The cells and current sensor output voltages are acquired by a DC1894B demo board. This Linear Technology board is equipped with an LTC6804 battery stack monitor capable of measuring up to 12 series connected cells with a total measurement error of less than 1.2 mV. The LTC6804 is also provided with two auxiliary analog inputs, which are used to accurately acquire the output and supply voltages of the current sensor. All the measurements take 11 ms to complete.

The battery was preliminary characterized in the laboratory with a Pulsed Current Test (PCT) profile at 25 °C [4]. The input current and the output voltages during the test are used to determine the AMA initialization parameters, *i.e.*,



(a)



(b)

Fig. 3. Photographs of the demonstrator setup.

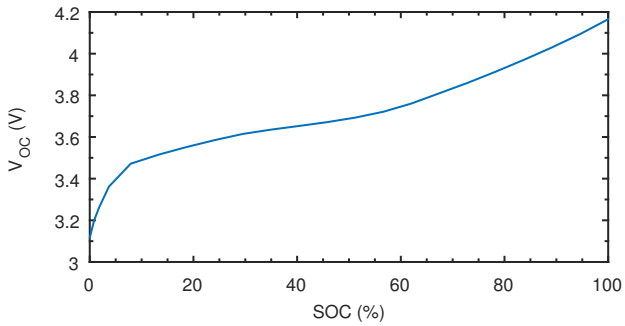


Fig. 4. Open circuit voltage V_{OC}, as a function of SOC.

the V_{OC}-SOC relationship, the capacity of the cells and the initialization values of the ECM parameters. The V_{OC}-SOC curve is considered invariant for every cell [3] and it is shown in Fig. 4. The extracted cell capacities are very similar to each other and are around the rated value of 10 A h, except for the second cell which has a 9.4 A h capacity. This result, together with the average values of the ECM parameters ($R_0 = 24.5\,\text{m}\Omega$, $R_1 = 13.3\,\text{m}\Omega$ and $C_1 = 6945\,\text{F}$), are used to initialize the algorithm implemented on the FPGA.

The Allegro current sensor, the DC1894B and SoCkit boards form a basic BMS, useful to test the hardware estimator. Fig. 5 shows a block diagram of it. The SoCkit Cyclone V SoC FPGA contains a dual-core ARM® Cortex™-A9 Hardware

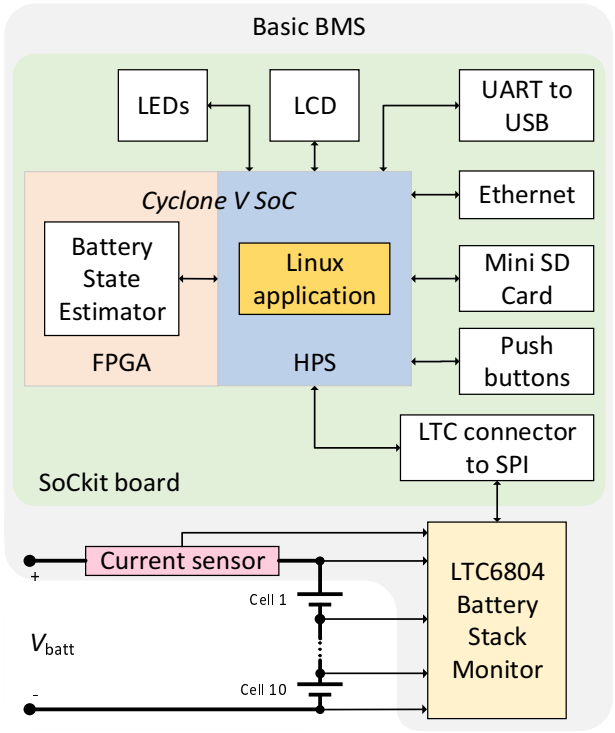


Fig. 5. Basic BMS block diagram.

Processor System (HPS). When the board is turned on, the Linux Ångström distribution, based on a real-time kernel, is booted from a partition on the mini SD card and the main application is launched. The application manages the acquisition of the cell data, their logging and the communication with the user interface. The current and voltage samples are read from the battery stack monitor connected to an SPI port. The samples are then sent to the estimator, implemented on the FPGA, via the MM interface, and the computation is started. The operation is cyclically executed every $0.1\,\mathrm{s}$, yielding an acquisition sample rate of $10\,\mathrm{Hz}$. When all the cell states are computed, the application saves the samples and the estimation results in a log file, which can be downloaded directly from the SD card or by using the Secure Copy Protocol (SCP) through the Ethernet port. The user interacts with the system via the push buttons on the board. Other data are shown on the LCD display and LEDs on the development board. The bike speed, useful for a better analysis of the data, is captured by a GPS unit.

IV. Validation Methodology

As a valuable contribution of this work, the implemented SOC estimator is validated in field tests, *i.e.*, riding the e-bike as in every day use. To evaluate the SOC estimation error, the SOC estimate needs to be compared with a reference value [16]. The common approach is to use the CC technique to obtain the SOC reference value, as shown in (4).

$$SOC_{\mathrm{ref}}(t) = SOC_0 - \frac{1}{Q_{\mathrm{n}}}\int_0^t i_{\mathrm{L}}(\tau)d\tau \qquad (4)$$

Eq. (4) can provide a reliable SOC reference SOC_{ref}, assuming that the initial SOC value SOC_0, the normalization capacity Q_{n} and the battery current i_{L} are known precisely.

The battery current should therefore be acquired by a highly accurate current sensor, different from the one used within the BMS, which feeds the SOC estimator. This is an important aspect, often neglected in literature, which makes it possible to assess the robustness of the estimator against errors in the current measurement effectively. In this work, i_{L} in (4) is measured by a National Instrument (NI) 9227 module mounted on a NI cDAQ-9178 chassis. The latter is also equipped with three NI 9215 modules, which are used to acquire the voltage of the 10 cells and to verify the LTC6804 stack monitor behavior (see Fig. 3(b)). The cDAQ-9178 is powered by a supplementary battery and connected to a laptop, carried by the cyclist in a backpack, which runs a LabVIEW application for data acquisition and logging (see Fig. 3(a)).

SOC_0 is taken equal to $100\,\%$, as tests are started after a full charge of the battery. Q_{n} is assumed to be equal to either the rated battery capacity or the capacity measured in a preliminary characterization test. These assumptions however may lead to inaccuracies in the computation of SOC_{ref}. Inhomogeneities in the cells of the battery pack can cause not all the cells to be fully charged to $100\,\%$ SOC. Moreover, the maximum charge that can be extracted from each cell during a validation test (we refer to this value as the *real* capacity) may differ from that measured during a characterization test, because of different discharge rates and temperature.

To overcome these problems, we propose to combine the CC and V_{OC} estimation techniques for computing SOC_{ref}. These two techniques are used to define two alternative metrics for assessing the SOC estimator errors [16], whereas here they are merged in a unique definition of SOC_{ref}. The basic idea is to set SOC_0 and Q_{n} in (4), so that SOC computed by (4) coincides with the estimate obtained through V_{OC} at the beginning and end of the validation test. This does not require any specific condition during the execution of the test, apart from the fact that the test starts with the battery in a rested state and ends in an SOC region where it is possible to reliably extract SOC from the V_{OC}-SOC relationship, once the battery has reached the steady state after discharge.

Let $V_{\mathrm{OC}} = f(SOC)$ express the V_{OC}-SOC relationship shown in Fig. 4. For each cell, we can compute SOC before and after the validation test by inverting the V_{OC}-SOC relationship, *i.e.*, $SOC_0 = f^{-1}(V_{\mathrm{OC}}^{\mathrm{start}})$ and $SOC_{\mathrm{end}} = f^{-1}(V_{\mathrm{OC}}^{\mathrm{end}})$, where $V_{\mathrm{OC}}^{\mathrm{start}}$ and $V_{\mathrm{OC}}^{\mathrm{end}}$ are the cell voltages measured, when the battery is in a steady state, before and after the validation test, respectively. Given the above definition, we can compute the *real* cell capacity related to a validation test, which coincides with the desired normalization capacity to be used in (4), by the following equation [17].

$$Q_{\mathrm{real}} = \frac{\int_{t_0}^{t_{\mathrm{end}}} i_{\mathrm{L}}(t)dt}{SOC_0 - SOC_{\mathrm{end}}} \qquad (5)$$

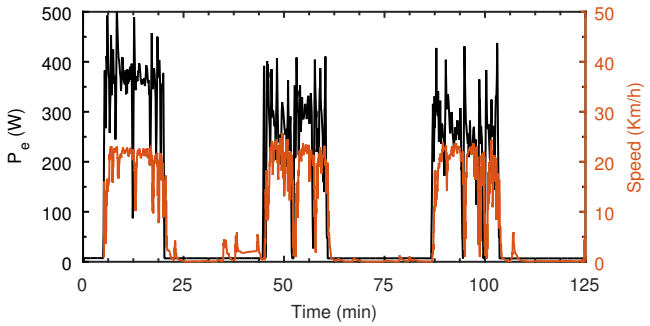


Fig. 6. Electric power and speed during the discussed validation test.

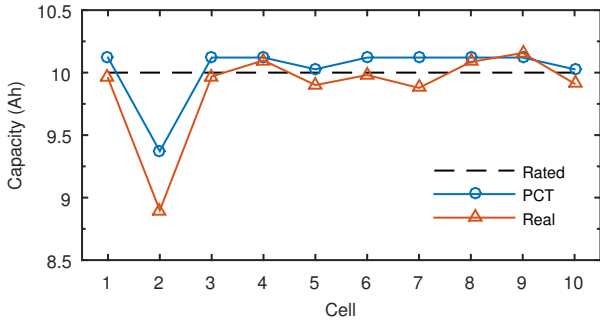


Fig. 7. Capacities of the cells extracted using the PCT and the discussed validation test.

V. Experimental Results

Several road tests have been carried out, proving both the functionality of the basic BMS provided with the AMA estimator and the effectiveness of the designed experimental set-up. As a representative example, we discuss the results obtained during a validation test, consisting of the repetitions of a cycling period followed by a rest pause. The electric power P_e provided by the battery pack and the speed measured by the GPS data logger during the test are shown in Fig. 6.

First of all, we compute the initial SOC and the *real* capacity of each cell of the battery pack from the data acquired by the NI cDAQ-9178 chassis. Then, we evaluate the corresponding SOC reference using (4). Fig. 7 shows the *real* cell capacity, for the discussed test, and compares it to the value obtained in the PCT characterization test. Apart from the second cell, which presents a pronounced degradation of the cell capacity, there is only a slight difference between the PCT, rated and *real* cell capacities, as expected by the high Coulombic efficiency provided by the Lithium-ion battery technology. The computed $SOC_{\mathrm{ref}}(t)$ for each cell is reported in Fig. 8, where it is evident that the SOC of the second cell corresponds to the battery SOC, as the second cell is the first cell to reach the discharge cut-off voltage.

With the availability of the SOC reference, we can calculate the estimation error introduced by the AMA. The AMA SOC estimate for the second cell is shown in Fig. 9 and is very close to the reference value during all the test. On the contrary, the pure uncompensated CC drifts over time. This

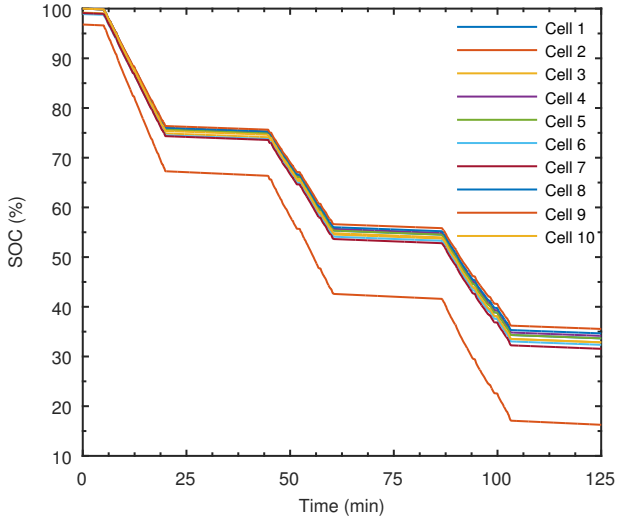


Fig. 8. SOC behavior of the 10 cells during the discussed validation test.

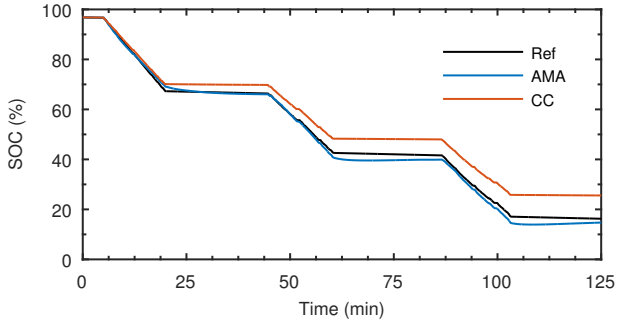


Fig. 9. Comparison of the SOC estimation results for the second cell.

clearly demonstrates that the AMA is capable of correcting the uncertainties in the current measurement (especially the offset of the current sensor) and in the real capacity of the cell.

Table II shows the maximum and rms errors of the AMA and CC SOC estimations. The AMA provides a more reliable estimation than CC, being its rms error below $2\,\%$ and the maximum absolute error around $3\,\%$, for all the battery cells. This is a valuable achievement, as it is obtained in road tests, when the battery is subjected to the real application load. In fact, if we consider the SOC estimation errors reported in [6] for a similar application, AMA outperforms EKF, as the latter introduces a maximum SOC error well above $5\,\%$. Better performance in terms of maximum error is obtained in [6] using an Adaptive EKF, at the expense of higher complexity. It is worth noting that the results reported in [6] refer to laboratory tests and not to road tests, as in this work.

Finally, Fig. 10 shows the R_0 and τ_1 ECM parameters identified by the AMA for the second cell. While the R_0 identification is robust against the choice of the window length, τ_1 identification seems to be sensitive to this choice. The results reported in Fig. 10 refer to a window length of $20\,\mathrm{min}$ and an LS matrix with 30 rows. This implies that the ECM

TABLE II
SOC ESTIMATION ERRORS

	AMA		CC	
	Max (%)	rms (%)	Max (%)	rms (%)
Cell 1	2.4	1.2	2.7	1.5
Cell 2	3.0	1.7	9.3	5.9
Cell 3	3.2	1.7	2.8	1.6
Cell 4	2.2	1.1	3.1	1.8
Cell 5	3.2	1.5	2.4	1.3
Cell 6	3.4	1.7	3.1	1.8
Cell 7	3.1	1.6	3.4	2.0
Cell 8	2.6	1.3	2.5	1.4
Cell 9	2.7	1.3	2.1	1.1
Cell 10	2.5	1.2	3.2	1.9

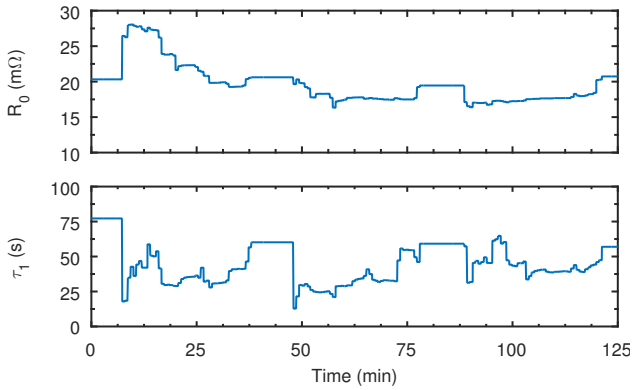


Fig. 10. ECM parameters estimated online by the AMA for the second cell.

parameters are updated every 40 s. Even if the windows in which the battery current is almost constant are discarded, the identification of τ_1 is less reliable, when the window overlaps a rest period.

VI. Conclusions

This work has demonstrated the implementation of a battery state estimator in an industrial FPGA. The estimator is based on the AMA and is implemented as a hardware accelerator block, which interacts to the processor embedded in the FPGA to form a system on a chip BMS. Thanks to its effective design, the estimator module can be used to estimate the state of all the cells in the battery pack in a time multiplexing fashion. As a case study, we have integrated the BMS into the battery pack of an e-bike. This allowed us to validate the estimator in a real application scenario, which represents a significant achievement, as almost all the literature on this topic reports on experimental results obtained in laboratory tests and for a single cell. The road tests carried out have shown the effectiveness of the designed BMS and experimental set-up, and that the AMA estimator provides a very good estimation comparable or even better than the well established EKF.

Acknowledgment

The authors would like to thank Alessio Tarabori and the System Solution Engineering group of Altera now part of Intel for the support and fruitful discussion on the topic of this paper. This work has been funded by the ECSEL Joint Undertaking under grant agreement No. 662192 (3CCar project).

References

[1] 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.

[2] Dai Haifeng, Wei Xuezhe, and Sun Zechang, "A new SOH prediction concept for the power lithium-ion battery used on HEVs," in *2009 IEEE Veh. Power Propuls. Conf.* IEEE, Sep. 2009, pp. 1649–1653.

[3] B. Pattipati, B. Balasingam, G. Avvari, K. Pattipati, and Y. Bar-Shalom, "Open circuit voltage characterization of lithium-ion batteries," *J. Power Sources*, vol. 269, pp. 317–333, Dec. 2014.

[4] R. Morello, W. Zamboni, F. Baronti, R. D. Rienzo, R. Roncella, G. Spagnuolo, and R. Saletti, "Comparison of State and Parameter Estimators for Electric Vehicle Batteries," in *IECON 2015 - 41st Annu. Conf. IEEE Ind. Electron. Soc.*, 2015, pp. 5433–5438.

[5] G. L. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 3. State and parameter estimation," *J. Power Sources*, vol. 134, no. 2, pp. 277–292, Aug. 2004.

[6] C. Taborelli and S. Onori, "State of charge estimation using extended Kalman filters for battery management system," in *2014 IEEE Int. Electr. Veh. Conf.* IEEE, Dec. 2014, pp. 1–8.

[7] F. Codeca, S. M. Savaresi, and G. Rizzoni, "On battery State of Charge estimation: A new mixed algorithm," in *2008 IEEE Int. Conf. Control Appl.* IEEE, Sep. 2008, pp. 102–107.

[8] V. Ramadesigan, P. W. C. Northrop, S. De, S. Santhanagopalan, R. D. Braatz, and V. R. Subramanian, "Modeling and Simulation of Lithium-Ion Batteries from a Systems Engineering Perspective," *J. Electrochem. Soc.*, vol. 159, no. 3, pp. R31–R45, Jan. 2012.

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

[10] 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 Int. Symp. Ind. Electron.* IEEE, May 2012, pp. 1336–1341.

[11] ——, "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.

[12] 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.

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

[14] F. Baronti, R. Roncella, R. Saletti, and W. Zamboni, "FPGA Implementation of the Mix Algorithm for State-of-Charge Estimation of Lithium-Ion Batteries," in *IECON 2014 - 40th Annu. Conf. IEEE Ind. Electron. Soc.*, 2014, pp. 5641–5646.

[15] R. Morello, F. Baronti, X. Tian, T. Chau, R. D. Rienzo, R. Roncella, B. Jeppesen, W. H. Lin, T. Ikushima, and R. Saletti, "Hardware-in-the-Loop Simulation of FPGA-based State Estimators for Electric Vehicle Batteries," in *to be Present. ISIE 2016*, 2016.

[16] B. Balasingam, G. Avvari, K. Pattipati, and Y. Bar-Shalom, "Performance analysis results of a battery fuel gauge algorithm at multiple temperatures," *J. Power Sources*, vol. 273, pp. 742–753, Jan. 2015.

[17] M. Einhorn, F. V. Conte, C. Kral, and J. Fleig, "A Method for Online Capacity Estimation of Lithium Ion Battery Cells Using the State of Charge and the Transferred Charge," *IEEE Trans. Ind. Appl.*, vol. 48, no. 2, pp. 736–741, Mar. 2012.