

# Battery Management System For Mobile Robots based on an Extended Kalman Filter Approach

Arezki Abderrahim Chellal<sup>\*1</sup>, José Lima<sup>2</sup>, José Gonçalves<sup>3</sup> and Hicham Megnafi<sup>4</sup>

**Abstract**—Robots are rapidly developing, due to the technology advances and the increased need for their mobility. Mobile Robots can move freely in unconstrained environments, without any external help. They are supplied by batteries as the only source of energy that they could access. Thus, the management of the energy offered by these batteries is so crucial and has to be done properly. Most advanced Battery Management System (BMS) algorithms reported in literature are developed and verified with laboratory-based experiments. The acquired data is then processed either online or offline, using PC-based software. This work consists of developing an on-Chip Extended Kalman Filter based BMS, which can be directly linked in a robot without having to be connected with an external device to process the data. The proposed system is implemented in a low-cost 8 bit microcontroller and results allow to validate the proposed approach.

## I. INTRODUCTION

Nowadays, the significance of storing the energy is taking more and more importance, as for instance, in the field of robotics, which is experiencing unprecedented growth. In order to allow the mobile robots to perform their tasks successfully, it is essential to provide them with the necessary energy to accomplish it, that's why the dimensioning of the battery is an essential part to achieve. In fact, it is also essential to monitor the total remaining energy of the battery throughout its use. That's why a battery management system (BMS) is applied to monitor the state of charge of the battery, but also to ensure the safety of the system.

The BMS is an electronic device that monitors continuously different harmful parameters: Temperature, Voltage and Current, and could at any time isolate it electrically, it must be able to indicate to the robot using the battery to stop getting its energy from it, or to go to the nearest recharging station. A BMS has to be "cell based" in order to be effective, and has to include and take into account different consideration, such as [1]:

- Operation in noisy environment
- Battery cell disconnection capabilities
- Uniform charge/discharge between cells

The management of a Lithium battery in mobile robots requires mainly the knowledge of the state of charge of the battery (SOC), which represents the ratio between the energy stored in (Ah) at a specific time " $t$ " and the total energy storable in the battery also in (Ah) in a reference time " $t_{ref}$ " [2]. It is the main state and it's calculation is a key factor that must be accurately determined, a reliable SOC measurement is essential for ensuring safety and maximizing battery life in rechargeable batteries in general.

The SOC determination is a challenging task due to the lack of any type of sensor that could directly give the SOC, thus it cannot be measured directly, so this is why some basical approaches such as the open circuit voltage (OCV) measure [3] and the Coulomb counting technique [4], or other ones, slightly more complicated such as Sliding Mode [5], [6], Artificial Intelligence [7] and Kalman Filtering [8] are then used, trying to estimate or predict the actual SOC of the battery.

As no specific accuracy requirement are defined, basic approaches are mostly used in smartphones and computers, they certainly use less time to calculate the state of charge, with a higher performance, but they do not produce an accurate calculation and generally give a misleading representation of it and lead to errors. [2], and on the other hand, the complex one, allow indeed to limit the error and the impact of uncertainties on the final result, but requires a computational power that is generally accomplished by the use of powerfull controllers such as dSPACE [9], that are expensive to be implemented on small mobile robots and/or need to be processed with PC-based software such as Matlab [10], [11] which requires a constant connection to a computer, thus losing their autonomy.

This paper has the objective to demonstrate the possibility of implementation of an Extended Kalman Filter (EKF) for the SOC estimation of a lithium battery in an Arduino Uno microcontroller, which is one of the most used microcontrollers on small autonomous robots, without the use of any external device for processing the data providing an on the fly acquisition and processing of data, giving small robots a better long term prediction accuracy of SOC at a reduced price. The paper is divided into 4 sections, the first section describes the way in which the equivalent model of the battery used by the EKF estimator has been deduced and describes the cell characterization tests applied to extract the cell parameter, then, the second outlines the hardware implementation by

<sup>1</sup>Arezki Abderrahim Chellal is with the Research Centre of Digitalization and Intelligent Robotics CeDRI, Instituto Politécnico de Bragança, Bragança, Portugal and the Higher School of Applied Sciences, Tlemcen, Algeria (e-mail: arezki@ipb.pt)

<sup>2</sup>José Lima is with the Research Centre of Digitalization and Intelligent Robotics CeDRI, Instituto Politécnico de Bragança, Bragança, Portugal and Robotics and Intelligent Systems Research Group, INESC TEC, Porto, Portugal (e-mail: jllima@ipb.pt)

<sup>3</sup>José Gonçalves is with the Research Centre of Digitalization and Intelligent Robotics CeDRI, Instituto Politécnico de Bragança, Bragança, Portugal and Robotics and Intelligent Systems Research Group, INESC TEC, Porto, Portugal (e-mail: goncalves@ipb.pt)

<sup>4</sup>Hicham Megnafi is with the Telecommunication Laboratory of Tlemcen LTT, University of Abou Bakr Belkaid, Tlemcen, Algeria and the Higher School of Applied Sciences, Tlemcen, Algeria (e-mail: h.megnafi@essa-tlemcen.dz)

describing in general the algorithm applied as well as the electrical schematics, putting forward the relations linking the various components with the microcontroller, followed by the third section, providing the results and discussion, highlighting them in the form of graphics and explanation, and finally, the fourth section, is a conclusion and further work description.

## II. BATTERY MODELING

### A. Mathematical Equations

Due to its physical and chemical properties, the battery has non-linear characteristics, making the prediction of its behaviour more difficult. The electrical model, which can easily be translated into a mathematical formula can be used by the EKF algorithm, this representation aims to reproduce the behaviour of the electrical quantities of the battery (SOC, Voltage, Current). This model, as shown in Figure 1, use standard electrical components such as resistors, capacitors and voltage source.

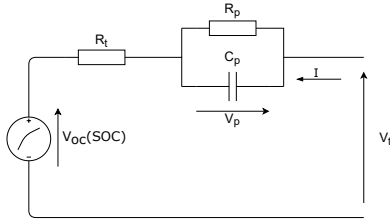


Fig. 1. First Order equivalent Lithium Battery Model

The voltage source denoted by  $V_{oc}$  represents the open circuit voltage (OCV) and is in function with the SOC (Equation 3), the series resistor  $R_t$  represents the voltage drops effect of the cell, the diffusion resistance is represented by the  $R_p$  resistor and  $C_p$  represents the polarization effect of the cell. The behaviour of this model is transcribed into a mathematical equation [2], the terminal voltage is given as in equation (1).

$$V_t = V_{oc}(SOC) + IR_t + V_p \quad (1)$$

Where,  $I$  is the current passing through the battery and  $V_t$  is the terminal voltage of the battery. The mathematical relation of the SOC(t) is expressed as follow:

$$SOC(t) = SOC(t_{ref}) + \int_{t_{ref}}^t \frac{I}{C_{actual}} dt \quad (2)$$

and,

$$V_{oc}(SOC) = a \cdot SOC + b \quad (3)$$

With,  $C_{actual}$  is the total energy storable in the battery in a reference time, and  $a$  and  $b$  variables that fluctuate according to the SOC. The time derivative of equation 2 can be expressed as given by:

$$\dot{SOC} = \frac{I}{C_{actual}} \quad (4)$$

The polarization voltage is given as,

$$\dot{V}_p = -\frac{1}{R_p C_p} V_p + \frac{1}{C_p} I \quad (5)$$

by substituting the equation 3 into equation 1 we obtain:

$$V_t = a \cdot SOC + b + V_p + R_t I \quad (6)$$

The state equation describing the behavior of a cell are then expressed in the form of the following system :

$$\begin{cases} \begin{bmatrix} \dot{SOC} \\ \dot{V}_p \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & \frac{-1}{R_p C_p} \end{bmatrix} \cdot \begin{bmatrix} SOC \\ V_p \end{bmatrix} + \begin{bmatrix} \frac{1}{C_{actual}} \\ \frac{1}{C_p} \end{bmatrix} I \\ V_t = \begin{bmatrix} a & 1 \end{bmatrix} \cdot \begin{bmatrix} SOC \\ V_p \end{bmatrix} + R_t I + b \end{cases} \quad (7)$$

The microcontroller treating current and voltage measurements as discrete events, it must therefore be necessary to retranscribe the previous system in its discrete form, it is then written as follows

$$\begin{bmatrix} SOC \\ V_p \end{bmatrix}_k = \begin{bmatrix} 1 & 0 \\ 0 & \exp \frac{-\Delta t}{R_p C_p} \end{bmatrix} \cdot \begin{bmatrix} SOC \\ V_p \end{bmatrix}_{k-1} + \begin{bmatrix} \frac{1}{C_{actual}} \\ \frac{1}{C_p} \end{bmatrix} I_{k-1} \quad (8)$$

$$V_{t_k} = \begin{bmatrix} a & 1 \end{bmatrix} \cdot \begin{bmatrix} SOC \\ V_p \end{bmatrix}_k + R_t I_k + b$$

A First-order equivalent battery model is preferred, because it represents the best compromise between accuracy and complexity, thus reducing calculation time and at the same time the computing power. Also, the hysteresis effect on the battery is not modeled, since previous research has shown that including hysteresis increases the model accuracy by about 5% [13].

### B. Cell Parameter Extraction

In order to identify the parameter of the first order equivalent model, a constant pulse current discharge test is executed on a fully charged Lithium battery. The test results are shown in Figure 2.

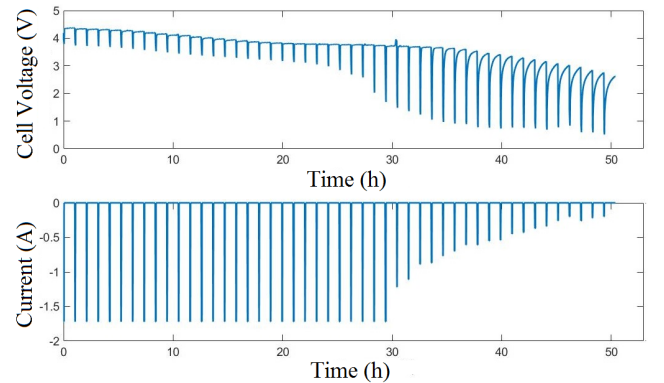


Fig. 2. Voltage and Current for a constant current pulse discharge

This set of test comprised a sequence of constant current discharges of 1.6 A for 180s, followed by a rest sequence of 3600s on an aged battery, the sampling time of the battery voltage and current measure is set at every second [6], since a lower sampling time will not necessarily increase the accuracy of the parameters obtained.

MATLAB offers a good tool for extracting battery parameters from the acquired data, the Simulink design optimization, using optimization techniques to estimate model parameters, which calculates and minimizes the error between the simulated and measured output. To determine these parameters, a simulation of a 1RC cell model was performed on Simulink and run in a Least Squares curve-fitting algorithm to solve for optimal parameter settings, where each circuit element was represented by lookup tables [14], it also provides the relationship between OCV and SOC (Figure. 3).

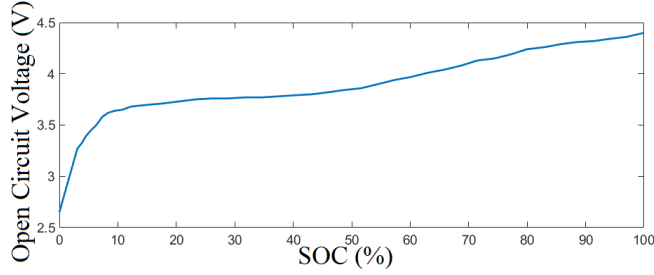


Fig. 3. Open circuit voltage and State of charge relation

Because simulating all the measurements acquired in 50 hours would take a considerable amount of time, a layering technique was applied, consisting of fragmenting the data obtained into several pieces and overlapping each other. The resulting data is summarized in Table I.

SOC	OCV (V)	a	b	$R_0$ ( $\Omega$ )	$R_p$ ( $\Omega$ )	$C_p$ (F)
1.000	4.40	0.800	3.60	0.4686	0.0968	715.600
0.800	4.24	0.800	3.60	0.4920	0.1138	1149.160
0.627	4.00	1.353	3.16	0.4614	0.1010	2718.350
0.517	3.86	1.320	3.17	0.4614	0.1010	604.480
0.431	3.80	0.700	3.50	0.4161	0.3922	283.736
0.346	3.77	0.350	3.64	0.4161	0.3922	563.120
0.232	3.75	0.176	3.71	0.4161	0.3922	19.550
0.122	3.68	0.636	3.60	4.2549	6.3909	85.600
0.037	3.27	4.260	3.16	4.9024	7.6956	91.623

TABLE I

SUMMARY OF THE RESULT OBTAINED BY APPLYING A LEAST SQUARE CURVE-FITTING ALGORITHM

### III. HARDWARE IMPLEMENTATION

#### A. Extended Kalman filter Algorithm

The Kalman filter (KF) provides an optimal estimate for a linear system with no directly measurable states. The state space representation and the measurement equation are represented by :

$$\begin{cases} x_k = A_{k-1} \cdot x_{k-1} + B_{k-1} \cdot u_{k-1} + \omega_k \\ y_k = C_k \cdot x_k + D_k \cdot u_k + v_k \end{cases} \quad (9)$$

Where,  $x_k \in R^{n \times 1}$  is the state vector,  $A_k \in R^{n \times n}$  is the system matrix in discrete time,  $y_k \in R^{m \times 1}$  is the output,  $u_k \in R^{m \times 1}$  is the input,  $B_k \in R^{n \times m}$  is the input matrix in discrete time, and  $C_k \in R^{m \times n}$  is the output matrix in discrete time.  $\omega_k$  and  $v_k$  represents the Gaussian distributed noise of the process and measurement.

After initialization, the KF Algorithm always goes through two steps, prediction and correction. The KF is extensively detailed in many references [15], [11], we will therefore not focus on its detailed description and will limit ourselves to describing its main lines of function.

The first step consists in predicting the states of the system from the previous estimation (or the Initial guess for the first iteration), this step is summarized in the following equations:

$$\begin{aligned} \hat{x}_k &= A_{k-1} \cdot \hat{x}_{k-1} + B_{k-1} \cdot u_{k-1} \\ \hat{P}_k &= A_{k-1} \cdot P_{k-1} \cdot A_{k-1}^t + Q \end{aligned} \quad (10)$$

The correction step, as its name indicates, allows to correct first the Kalman Gain  $K$  and secondly the previously made state prediction in equation 10 ( $\hat{x}_k$ ) and the Covariance Matrix  $P$ , the succession of this step is represented in the following equations:

$$\begin{aligned} \hat{y}_k &= C_k \cdot \hat{x}_k + D_k \cdot u_k \\ K_k &= \hat{P}_k \cdot C_k^t \cdot (C_k \cdot \hat{P}_k \cdot C_k^t + R)^{-1} \\ x_k &= \hat{x}_k + K \cdot (U_k - \hat{y}_k) \\ p_k &= (I - K_k \cdot C_k) \cdot \hat{P}_k \end{aligned} \quad (11)$$

The overall performance of the filters is set by the covariance matrix  $P$ , the process noise matrix  $Q$  and the measurement noise  $R$ . As stated in the literature [8], the determination of these parameters can directly be obtained through experience and empirical experiments.

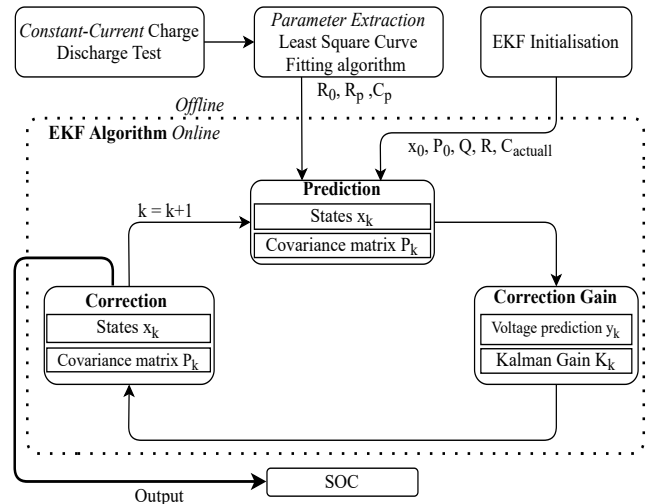


Fig. 4. Flow chart of the EKF algorithm implemented in the Arduino

By overlapping the equation 9 and 8, it is possible to determine the  $A_k$ ,  $B_k$ ,  $C_k$  and  $D_k$  matrices:

$$A_k = \begin{bmatrix} 1 & 0 \\ 0 & \exp\left(\frac{-\Delta t}{R_p C_p}\right) \end{bmatrix}, \quad B_k = \begin{bmatrix} \frac{1}{C_p} \\ \frac{1}{C_p} \end{bmatrix},$$

$$C_k = \begin{bmatrix} a & 1 \end{bmatrix}, \quad D_k = R_t$$

However, in the case of nonlinear systems such as batteries, The equation 9 can not provide an accurate estimation of the states of the battery (SoC and  $V_p$ ), in this case, the Extended Kalman filter (EKF) is used to include the non-linear behaviour and to determine the states of the system [17], [8], the EKF equation have a form of:

$$\begin{cases} x_{k+1} = f(x_k, u_k) + \omega_k \\ y_k = h(x_k, u_k) + v_k \end{cases} \quad (12)$$

where

$$A_k = \frac{\partial f(x_k, u_k)}{\partial x_k}, \quad C_k = \frac{\partial h(x_k, u_k)}{\partial x_k} \quad (13)$$

In order to obtain the most accurate result, the algorithm is written in a way that allows calculation with the smallest possible time interval, the minimum sampling time obtained is 0.0401 seconds.

#### B. BMS Circuit

After implementing the EKF algorithm in the Arduino uno board, it is now necessary to measure the battery current and voltage, which represent the input and output of the model, respectively. To do this, the electronic circuit shown in Figure.6 and Figure.5 is used.

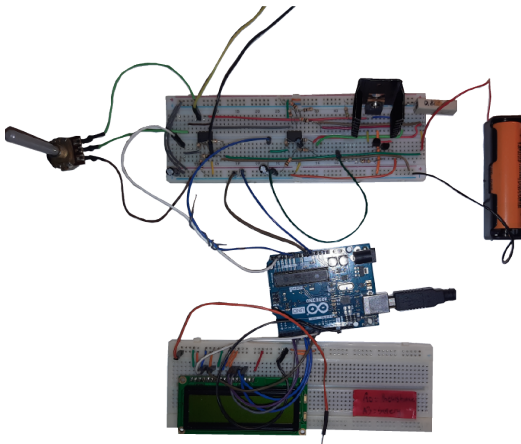


Fig. 5. Electronic circuit for current and voltage measurements

This circuit is composed by different electronic components. The LCD is connected to the microcontroller via the digital pins (PORT4 to PORT9). The current and voltage are gathered respectively through PORT A0 and A3 from two DUAL-Amplifiers LM358P (U2:A & U1:A). The Mosfet IRF1405 (Q1) allows to keep the voltage between the

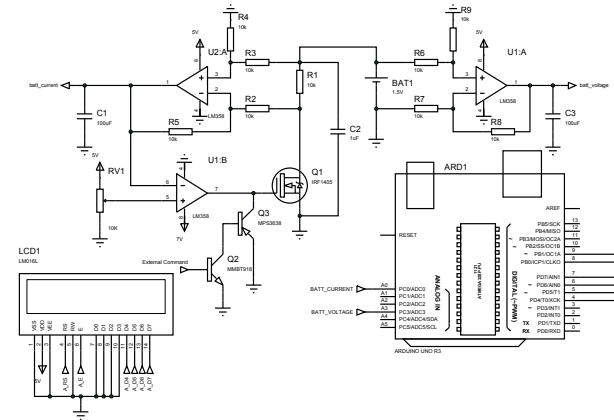


Fig. 6. Schematic diagram of the electronic circuit for measuring current and voltage

terminals of the resistor constant and thus the current, it is controlled by a comparator (U1:B), which compares the current voltage of the resistor with another reference voltage. Because the Mosfet is switched at its maximum, capacitors and resistors (R0, C1, C2 & C3) are added to limit the interferences to the maximum [18], [16].

#### IV. RESULTS AND DISCUSSION

In order to confirm the correct implementation of the estimator and its performance, different 18650 Lithium Ion batteries, with different capacity and aging are tested under room temperature. All the batteries start at 100% cell capacity with an initial guess of the estimator set at 50%, which will show the robustness of this estimator on Arduino. This section regroups the result acquired for one of the batteries tested. Also, in an effort to get as close as possible to a real environment, the discharge test is not done with a constant current, since the mobile robot can ask the battery to supply it with variable power (and therefore variable current) at any time, the test will therefore be done with a constant current for a variable period of time and will change afterwards. In Fig.7, The predicted voltage by the onboard estimator is compared with the real voltage of the Lithium-ion cell.

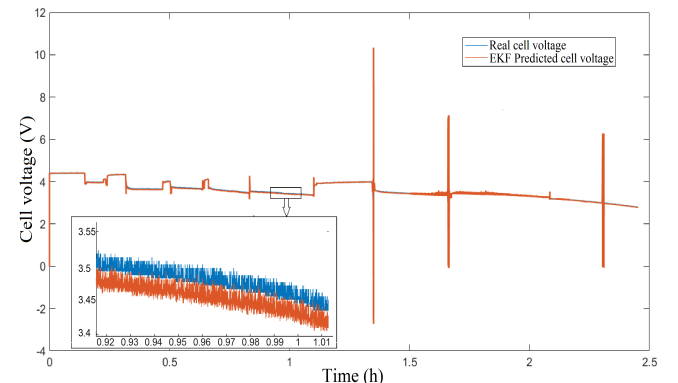


Fig. 7. The measured battery terminal voltage compared with the estimated voltage by the proposed EKF Algorithm

It is clearly seen that apart from a few exceptions due mainly to current measurement error (that can be seen in fig 8), the EKF implemented in the Arduino follows the battery voltage perfectly, minimizing the error as much as possible. The battery data, shown in Table.I, is implemented with the algorithm, the reference SOC used for comparison is calculated in parallel, with a Coulomb counting method, consisting of a current measurement and integration in a specific period of time, this reference SOC following this method is acquired through the use of equation 2. The MATLAB environment is only used for the purpose of collecting the results of each iteration (Voltage, Current, SOC), and it should be specified that no calculation is made through it.

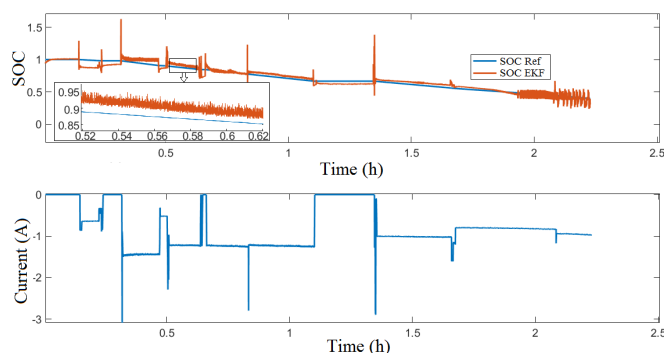


Fig. 8. Estimated and reference SOC comparison with the current measurement

The estimated SOC track the true SOC value during all the test but most especially between 80% and 34% of SOC of the battery, because, between these two values, the battery follows more or less a linear behavior and this can be proved from the Table.I and Figure.3, in order to limit the computing power, the linearization of these curves is done in only 10 points which explains the bad prediction outside this area.

The result of the SOC estimation of the EKF compared with the real SOC, are shown in Fig. 8, the error between them ( $|\Delta SOC|$ ) is shown in Fig. 9.

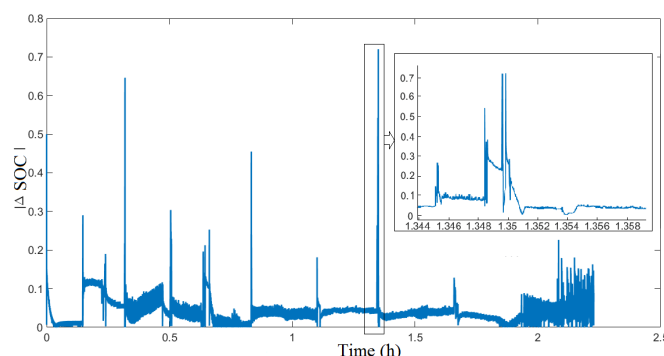


Fig. 9. Absolute Error

During the rough passage from one current to the other, the Mosfet generates interference that affect the correct reading of the current. This results in a false current peak

measure that can be seen in the figure 8, this peak causes the divergence of the estimate during 5 iterations, this interference is kept to allows the validation of the robustness of this estimator on the Arduino, indeed after 200 iterations (equivalent to 8 seconds) the prediction of the SOC converges again. If this brief divergence is not taken into account, this estimator gives an average error of 3.81% in over more than 2 hours of use.

Figure 10 represents the beginning period of Figure 8 (The first three minutes), even if this algorithm can be qualified as robust, it still takes for it 2 minutes to reach the reference

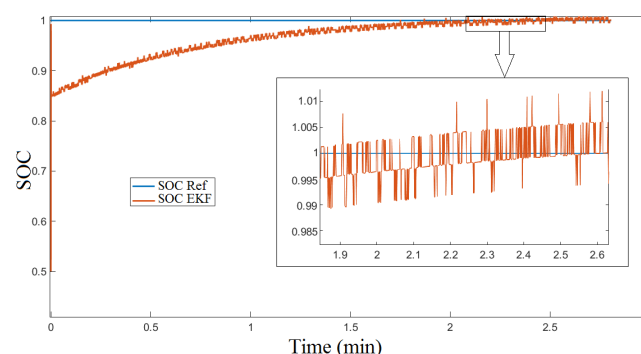


Fig. 10. The reference SOC compared with the SOC estimated within the first 3 minutes

## V. CONCLUSION AND FUTURE WORK

The presented method showed that the Algorithm implemented in the microcontroller can track accurately the SOC in a real environment, and showed its perfect operation in noisy environment. These uncertainties were compensated by the Extended Kalman Filter Algorithm, The performance of this estimator has been verified with a SOC error varying between 10% and 1% in each iteration and with an average value of 3.81%. It showed that this algorithm can be implemented in the most basic microcontroller which can give to mobile robots a better accuracy in prediction of SOC. As the developed model is functional for known battery parameters, further work involves improving this algorithm with the addition of a state of health (SOH) estimator to the previous state of charge estimator, which will continuously update these parameters; these two estimators can work together to give both the state of charge and the ageing of the battery, giving the mobile robot full autonomy in terms of power management during long periods of use.

## ACKNOWLEDGMENT

This paper is supported by ROBOSTEAM Erasmus+ KA201 Project with reference 2018-1-ES01-KA201-050939.

## REFERENCES

- [1] J. Chatzakis, K. Kalaitzakis, N. C. Voulgaris and S. N. Manias. "Designing a new generalized battery management system," *IEEE transactions on Industrial Electronics*, vol. 50, no 5, pp. 990-999, 2003.

- [2] A. Lieve. "Développement d'un système de gestion de batterie lithium-ion à destination de véhicules" mild hybrid": détermination des indicateurs d'état (SoC, SoH et SoF)," Ph.D. dissertation, Claude Bernard - Lyon 1 Univ., Lyon, 2015.
- [3] M. Frivaldsky, J. Sedo, M. Pipiska and M. Danko. "Design of measuring and evaluation unit for multi-cell traction battery system of industrial AGV," *Electrical Engineering*, 2020, pp. 1-13..
- [4] K. -S. Ng, C.-S. Moo, Y.-P. Chen and Y.-C. Hsieh. "Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries," *Applied energy*, vol. 86, no 9, pp. 1506-1511, 2009.
- [5] J. Du, Z. Liu, Y. Wang and C. Wen. "An adaptive sliding mode observer for lithium-ion battery state of charge and state of health estimation in electric vehicles," *Control Engineering Practice*, vol. 54, pp. 81-90, May 2016.
- [6] I.S. Kim. "The novel state of charge estimation method for lithium battery using sliding mode observer," *Journal of Power Sources*, vol. 163, no 1, pp. 584-590, Apr 2006.
- [7] P. Tejaswini and P. Sivraj. "Artificial Intelligence based State of Charge estimation of Li-ion battery for EV applications," in *2020 IEEE. 5th International Conference on Communication and Electronics Systems*, pp. 1356-1361.
- [8] C. Campestrini. "Practical feasibility of Kalman filters for the state estimation of lithium-ion batteries," Ph.D. dissertation, Technische Universität München., Munich, 2017.
- [9] X. Zhou, B. Zhang, H. Zhao, W. Shen and A. Kapoor. "State of charge estimation based on improved Li-ion battery model using extended Kalman filter," in *2013 IEEE 8th Conference on Industrial Electronics and Applications*, pp. 607-612.
- [10] K.-T. Lee, M.-J. Dai and C.-C. Chuang. "Temperature-compensated model for lithium-ion polymer batteries with extended Kalman filter state-of-charge estimation for an implantable charger," in *2017 IEEE Transactions on Industrial Electronics*, vol. 65, no 1, pp. 589-596.
- [11] A. Mouna, B. Abdelilah, N. M'Sirdi, and al. "Estimation of the state of charge of the battery using EKF and sliding mode observer in Matlab-Arduino/LabView," in *2018 4th International Conference on Optimization and Applications*, pp. 1-6.
- [12] I.S. Kim. "A technique for estimating the state of health of Lithium batteries through a dual-sliding-mode observer," *IEEE Transactions on power electronics*, vol. 25, no 4, pp. 1013-1022. Apr 2010.
- [13] . Hu, S. Li and H. Peng. "A comparative study of equivalent circuit models for Li-ion batteries". *Journal of Power Sources*, 2012, vol. 198, pp. 359-367.
- [14] R. Jackey et al. "Battery model parameter estimation using a layered technique: an example using a lithium iron phosphate cell," *SAE Technical Paper*, 2013, vol. 2, pp. 1-14.
- [15] G. Welch, G. Bishop, et al. "An introduction to the Kalman filter," 1995.
- [16] Datasheet, "Automotive MOSFET IRF1405 International Rectifier", 2005.
- [17] M. Mastaliet al. "Battery state of the charge estimation using Kalman filtering," *Journal of Power Sources*, 2013, vol. 239, pp. 294-307.
- [18] A. P. MALVINO and D. BATES. *Principes d'électronique*, DUNOD, France, 2016.