

Document downloaded from:

<http://hdl.handle.net/10251/47216>

This paper must be cited as:

Lajara Vizcaino, JR.; Pérez Solano, JJ.; Pelegrí Sebastián, J. (2015). A method for modeling the battery state of charge in wireless sensor networks. *IEEE Sensors Journal*. 15(2):1186-1197. doi:10.1109/JSEN.2014.2361151.



The final publication is available at

<http://dx.doi.org/10.1109/JSEN.2014.2361151>

Copyright Institute of Electrical and Electronics Engineers (IEEE)

A method for modeling the battery state of charge in wireless sensor networks

Lajara R., Perez-Solano J.J., and Pelegri-Sebastia J., *Member, IEEE*

Abstract— In this paper we propose a method for obtaining an analytic model of the battery State-of-Charge (SoC) in wireless sensor nodes. The objective is to find simple models that can be used to estimate accurately the real battery state and consequently the node lifetime. Running the model in the network nodes, we can provide the motes with the required information to implement applications that can be considered as battery-aware. The proposed methodology reduces the computational complexity of the model avoiding complicated electrochemical simulations and treating the battery as an unknown system with an output that can be predicted using simple mathematical models. At a first stage, during a setup period, the method starts with the measurement of several battery parameters under different environmental and operational conditions. After that, the method uses the previous collected data for building mathematical models, considering the linear regression or the multilayer perceptron as the most appropriated. Finally, the models are validated experimentally with new measures. Results show the suitability of the method that produces accurate and simple models, capable of being implemented even in low-cost and very constrained real motes.

Index Terms—Batteries modeling, batteries State-of-Charge, energy, wireless sensor networks

I. INTRODUCTION

WIRELESS Sensor Networks (WSN) has become one of the most important and most active research topics in last years. The network's nodes (motes) are composed of low-cost and low power components, including a wireless communications transceiver, a microcontroller, some sensors and a battery. The kind of work a mote does is usually very simple, due to its low computational capabilities, and it involves the data sampling process and its subsequent transmission through the wireless link to a central base station. The motes have very strict constraints in terms of size,

This paragraph of the first footnote will contain the date on which you submitted your paper for review. It will also contain support information, including sponsor and financial support acknowledgment. For example, "This work was supported in part by the U.S. Department of Commerce under Grant BS123456".

The next few paragraphs should contain the authors' current affiliations, including current address and e-mail. For example, F. A. Author is with the National Institute of Standards and Technology, Boulder, CO 80305 USA (e-mail: author@boulder.nist.gov).

S. B. Author, Jr., was with Rice University, Houston, TX 77005 USA. He is now with the Department of Physics, Colorado State University, Fort Collins, CO 80523 USA (e-mail: author@lamar.colostate.edu).

T. C. Author is with the Electrical Engineering Department, University of Colorado, Boulder, CO 80309 USA, on leave from the National Research Institute for Metals, Tsukuba, Japan (e-mail: author@nrim.go.jp).

computing capacity or energy. The mote's available energy sources are normally batteries and in some cases an additional energy-harvesting hardware that can recharge the battery or any other alternative energy storage device. When the mote is equipped only with batteries, they are usually of the lithium or alkaline type. The mote can capture energy from the environment using several sources like solar panels, piezoelectric generator, RF harvester, etc and the storage element can be a battery, a supercapacitor or a combination of them [1].

One important issue in the design and implementation of WSN applications is the optimization of the nodes' power consumption. This problem has been studied from different points of view: the optimization of the communications protocols, the improvement of the semiconductor components, the optimization of algorithms, etc. In [2], it is stated that a higher optimization of the power consumption can be achieved through the reduction of the device power consumption rather than optimizing only the network protocols, although they are not mutually exclusive methods.

In this context, some papers highlight the necessity of developing battery models that allow a precise estimation of the available battery capacity with the objective of improving the power consumption efficiency and the network lifetime. This estimation is essential for establishing a suitable duty cycle at the MAC layer, sustainable routing paths, scheduling the workload, etc. But battery modeling, and in particular for Lithium batteries, is still an open problem, and to the best of our knowledge, there is very little work about the development of lightweight rechargeable battery models that can be run on a network mote. So, the main objective of this paper is to propose a method for obtaining simple battery models that can estimate precisely the state of charge of the mote's battery, using exclusively the original mote's hardware. In the article five mathematical models based on linear regression and the multilayer perceptron (MLP) are studied, evaluating the goodness of the fit between the model and the actual battery behaviour.

The paper is organized as follows. In section 2, we provide the state of art. Section 3 summarizes the steps to be carried out for building models. The hardware used for collecting the experimental data is shown in section 4. Section 5 presents the mathematical models. In section 6 the validation of the models is carried out. Finally, section 7 is devoted to present the conclusions and future work.

II. STATE OF ART

Wireless sensor networks have been conceived as a versatile technology that can be applied to a large quantity of different scenarios. But there is a common limitation in the majority of these applications that is the power consumption. This constrain is even more pronounce in outdoors deployments, where the motes are placed at distant or remote locations. Because energy efficiency is of primary importance in WSN, the network's motes should be aware of the charge that is left in the battery. In some cases, this matter is addressed modeling the power consumption of the motes. In references [3-5] the model is focused on determining the consumption of the wireless transceiver, because it is the element with highest impact in the total power consumption, whereas in [6] the microcontroller and the sensors effects are also taken into account. Typical WSN applications put the mote in three different states during its operation: sensing (acquiring the data), data transmission (using the wireless transceiver) and some power down mode (sleep state). The mote passes from one state to another following the scheduler commands, changing the state periodically and completing the orders programmed in the software. The mote's current consumption varies depending on each specific power state, but a global model can be developed considering the consumption at each state and the transitions between them [7, 8]. The consumption models can be applied to estimate the life expectancy of the batteries' nodes or to make better decisions, such as: to optimize the network protocols [2,3,9-11] and algorithms [12,13]. In [14] the WSN platform is parameterized with a constant average current for each action that it can do. The SoC estimation is performed by a software component that accumulates the current values over the period of time at which each action is carried out; thus, this method is similar to a Coulomb counter by software. However, these models are not completely precise because they do not include the battery behaviour.

The most common and simple way of estimating the battery state of charge (SoC) of a sensor node is to measure the battery voltage [15]. In this case the management system needs only to measure the battery voltage either in open circuit or with a load connected. After that, the voltage-based method tries to estimate the charge level relating the SoC with the voltages measured. However, the battery voltage of a lithium battery is nonlinear and its voltage drops very fast in the last moments of its lifetime. This is because there are several different types of polarization in the battery operation. During the ohmic polarization the battery response is quasi-linear, but after the concentration polarization there is a quick voltage drop [16]. An example that follows this approach is presented in [17]. This method generates a SoC model from a voltage trace acquired during a constant current discharge. The model estimates the SoC using a polynomial function that is adjusted to the battery voltage trace. In order to avoid the dependence with the current value, a second adjustment is carried out over several experimental initial traces with different currents to generate the polynomial coefficients with distinct loads. This model is the only one that we have found in the bibliography

which fits well the WSN constrains in terms of computational complexity, but the duty cycle operation of the motes that produces an irregular current consumption can negatively affect its accuracy. There are other methods that estimate the SoC measuring the discharge current. In this case, the remaining charge contained in the battery can be determined subtracting the charge that is extracted. Nevertheless, these methods can accumulate errors, require periodic recalibration and usually need additional hardware to perform the coulomb counting [18]. Other parameters, which can modify the battery effective capacity and are not considered in the previous methods, are: temperature, load percentage, number of charge/discharge cycles, pulsed cycles [13] and aging that produces a modification of the internal structure of the electrodes due to losses of the active material dissolution [19].

In the literature, there are also many battery models that can be applied to the SoC estimation. The first kind of models treats the battery as a complex electrochemical system and they simulate the internal chemical processes of the battery in a detailed way [20]. As a result, the electrochemical models present a high estimation precision but a huge computational cost that makes them too complex to be executed on a microcontroller [21]. The second approach is represented by analytical methods that avoid the underlying chemical process, modeling the battery at a higher abstraction level. In this group, the most popular method is based on the Peukert's law that predicts the SoC of a lead-acid battery measuring its discharging rate. However, this formula was developed for lead-acid batteries, and it is not directly applicable for lithium batteries [22]. Moreover, Peukert's law assumes a constant load and it does not fit well with the WSN constrains. Other methods take into account other parameters or estimate them in some way, for example the internal resistance of the battery [23,24]. All batteries have internal impedance. This impedance can be measured in AC or DC, but in both cases it has a high correlation with aging and the capacity loss. This impedance is related to the electric resistance of internal materials and the ionic resistance caused by electrochemical conductivity factors [25]. There are different methods of measurement. For example in vehicle applications [26] the DC resistance is calculated draining different current levels and sampling the battery voltage. The battery voltage decreases with a higher current demand [27] and thus the resistance can be calculated as:

$$R = \frac{V_2 - V_1}{I_2 - I_1} \quad (1)$$

Other analytic models attempt to provide an equivalent representation of the battery and estimate the SoC using neural networks combined with extended Kalman filters [28]. Unfortunately, the implementation of the extended Kalman filters is too complex for WSN. As a conclusion, it is clear that current battery SoC models are not suitable to be implemented in motes with low computational capabilities and new approaches are needed in this sense. On the other hand, simple models developed specifically for WSN do not cover all the battery parameters and the particular conditions that appear in

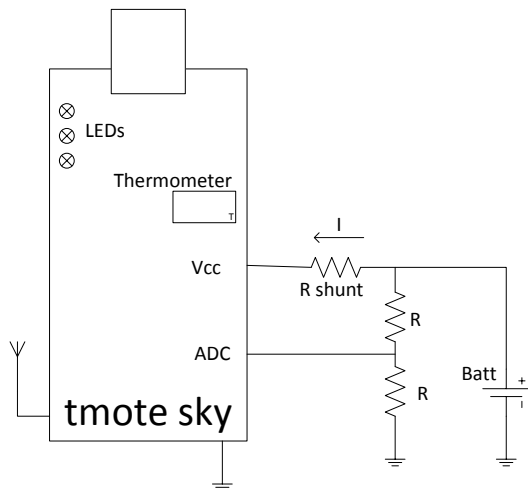


Fig. 1. Schematic of the setup used for measuring voltage and current.

real WSN applications. Moreover, the development of better consumptions models and battery SoC estimators can benefit other research aspects. For example, it has been found that simulators lack good power consumption models [2], which can negatively affect research on new protocols and architectures [7]. Another potential application is to detect or predict when maintenance would be necessary to change or recharge batteries.

III. METHODOLOGY

In the article, we have followed a sequence of steps to complete the generation and validation of the battery SoC models. These steps are explained below to illustrate the process with the aim that it can be followed by other researches interested in the method. The process to develop the models is:

1). Measurement. It is an initial acquisition stage that consists of several monitoring experiments of the battery connected to the mote during its normal operation. The device sends the sampled data, e.g.: voltage, current and temperature to a base station. The session will begin with a battery totally charged, and ends when the node stops working. Each session will have particular conditions, such as: temperature or number of charge/discharge cycles.

2). Modeling. First of all a filter is applied to reduce the noise. After that, new parameters are calculated from the original data. Conditions of the measurements are also introduced as new parameters. The SoC is calculated for every sample using the total duration of each session and the time of each sample, the SoC is the percentage of remain time from the beginning of the session. Finally, a set of representative data of each session is extracted to form a final data set, which will be used to build a model. During the model generation the parameters that provide more information are selected and the others are discarded. The selected parameters are used in several data mining algorithms to predict the SoC.

3). Validation. This section is included for testing the models. First of all, some results on the prediction accuracy using the same data files are provided. But since these models

are generated using the same data, they could depend on them, making the test invalid. So in this step, new data measurements sessions are used to study the prediction accuracy with completely new data.

IV. MEASUREMENT

In this section we describe the experimental setup used for acquiring the data traces during the initialization phase. This experimental data include some parameters taken during the discharge of a rechargeable battery under different environmental conditions. The setup hardware collects the parameters that are relevant to the battery behaviour. This acquisition of the data involves a sequence of charge and discharge cycles performed automatically in a stress testbench to capture the evolution of the battery SoC. In order to make our work open to other researches and easily reproducible, we include a detailed description of the hardware and provide all the schematics and the software programs through this link (<http://www.uv.es/jjperezs/models/materials-models.zip>). In addition, we have made use of open hardware, that is, devices that can be easily built or commercially purchased because they are well known and widely used.

A. Measusement setup

A.1. Mote's hardware

Now we describe the hardware used in the acquisition of the battery current and voltage when it is connected to the mote. With these hardware elements, we will build and validate the battery SoC models that are proposed in the next section. The batteries that we have used during all this work are the PRT-11316 [29], of 40 mAh form xtra-power that are rechargeable and provide a sufficient current to supply a mote. We have selected this model because it has a relative low capacity, a feature that allows us to carry out the measures more quickly, although this method could be applied to any another model of battery.

The WSN mote considered in the experimental setup is the well-known tmote Sky [30]. This platform has a TI MSP430F1611 microcontroller running at 8 MHz, a CC2420 wireless transceiver, USB connection, 3 LEDs, Flash and RAM memory and three kind of sensors: temperature, humidity and light. The LEDs have a resistor with values 470, 220 and 100 Ω (considering $V_{cc}=3.7$ V it represents 4.3, 7.3 and 16 mA respectively). The microcontroller has several low-power modes. The active mode needs 500 μ A when the microcontroller runs with a 1 MHz clock, and 2.6 μ A in standby mode. The transceiver [31] consumes 18.8 mA in reception mode, 17.4 mA in transmission, 426 μ A in sleep, 20 μ A in power down, and 0.02 μ A in shut down. The temperature sensor is a Sensirion SHT11 [32], which consume 2 μ W in sleep mode and 3 mW during the conversion.

We have used a shunt resistor in order to measure the current drained by the mote. The voltage in the first terminal of the shunt resistor is the power line of the microcontroller. This voltage is measured internally using the microcontroller's ADC with a reference voltage of 2.5 V. The other terminal is the battery voltage, which can be measured using a basic

circuit connected to other ADC input, as it represented in Fig. 1. If the value of the divisor resistors is low, the current drained will be higher. On the other hand, if the value of these resistors is high, it will be necessary an operational amplifier to convert the analogue signal properly. Considering both cases and establishing a trade-off between them, we take a value of 1 k Ω for each resistor. The shunt resistor that we have selected is of 32 Ω for extending the measures voltage range in the model generation phase. This is especially appropriate with low currents, because it allows us to obtain better quality data.

A.2. Software of the mote

The code of the mote constitutes a typical WSN application that performs a periodic sensor sampling and a subsequent wireless transmission of the data to the central base station. After sending the data, the node goes to sleep and put all the elements in a power down mode. The tmote Sky application is programmed in TinyOS [33], which is open source and widely used due to its low power consumption [34]. The sampling period that is established in the program is one second. The detailed list of actions that the program carries out is:

1. Measures the battery voltage and current (V_{low} , I_{low}) during the low power mode.
2. Measures the temperature (Temp).
3. Wakes up the transceiver and builds the data frame.
4. Sends the data.
5. Switches-on the LEDs with the purpose of increasing the current consumption (high state).
6. Measures the battery voltage and current (V_{high} , I_{high}).
7. Shuts-down LEDs, transceiver and microcontroller.

The battery measurements are performed with the analogue to digital converter ADC integrated in the microcontroller and it is configured with an internal reference of 2.5 V for measuring V_{cc} . This action is carried out using the VoltageC component of TinyOS library. The Fig. 2 shows a graph of the current drained during the mote operation. There are different current levels between *low* (sensing) and *high* (transmitting and LEDs on) states. The measurement corresponds with the shunt voltage. There are 625 μ A during 272 ms in the low state, and 28 mA during 28 ms in the high state.

The reception mote is another tmote Sky running the BaseStation TinyOS application. It retransmits all the data through the serial port. This board is connected to a PC, where a LabVIEW software receives and saves the data. The measurement ends when the transmitter mote depletes the battery and cannot send more data frames.

In this moment, we calculate the real SoC using the

timestamp of each sample and the total time. We use a pre-processing stage in order to calculate others parameters. The pre-processing stage recovers or eliminates measurement errors and reduces the noise applying a IIR filter to the data.

The number of charge/discharge cycles can be calculated by the microcontroller in the same way presented in [17]. This method is built as a finite state machine that checks the voltage level and the slope. The original method can be simplified if we are only interested in the charge/discharge phases. Basically, the microcontroller monitorizes continuously the battery voltage; if there is an over-voltage ($V > V_{float}$), the battery is charging and it can increment the counter of cycles. But, when the voltage is equal or inferior to the nominal value and decreases, the battery is discharging. This method can be seen in Fig. 3 the system begins from an unknown state (init). If the battery voltage decreases, the battery is discharging and the system goes to this state. On the other hand, if battery voltage increases over V_{float} , the battery is charging and the system changes in this way its state. This method could be

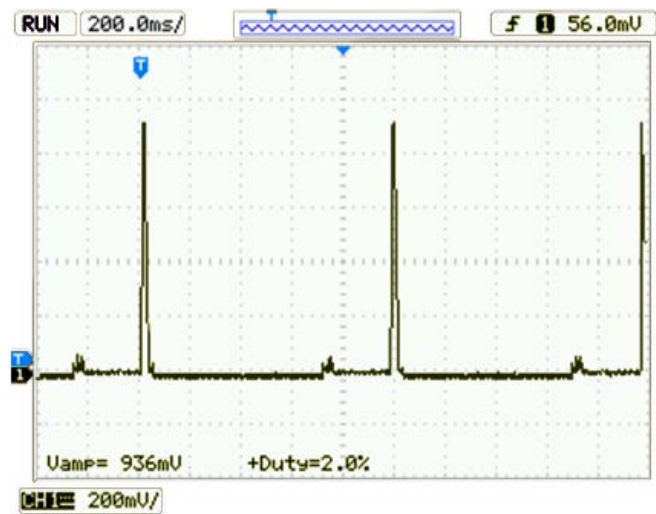


Fig. 2. Consumption of tmote Sky. Rshunt = 32 Ω .

extended by tracking the duration and/or current of the charge cycle.

A.3. Stress testbench

We have developed a system for accelerating the wear of the batteries that is shown in Fig. 4. This system applies consecutives cycles of charge and discharge automatically. The system has a LiPo battery charger, with reference MCP73831 [35], which regulates the voltage and current in the charge phase. The maximum charge current is configured to the nominal current for this battery.

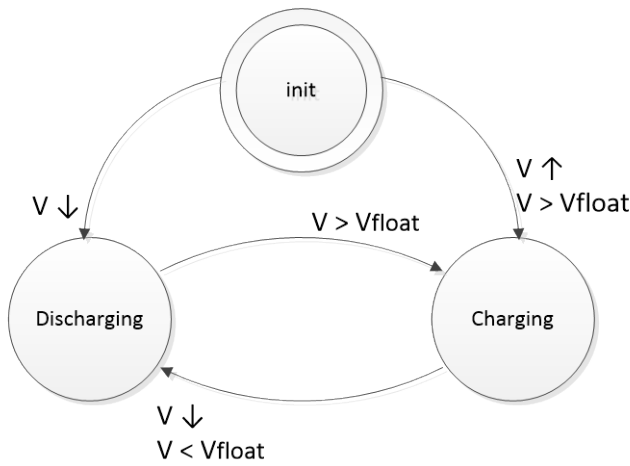


Fig. 3. State machine for monitoring battery status.

The load resistor simulates the node consumption, but the current has been increased. The wear depends on the discharge ratio [36], but we are going to assume that it is a constant and discharges completely the battery for simplicity. This assumption is extracted from the results presented in reference [36] since it is not appreciated a noticeable dependence between the Deep-of-Discharge (DoD) and the capacity. The discharge current is selected using a load of 100 Ω, being the nominal current value of the battery 40 mA.

The system has two MOS-switches, the first one enables the battery charge, and the second one enables the discharge across the load resistor. A control system opens and closes the switches, and it monitors the battery status: charged or uncharged, then, it makes a cycle of charge or discharge. When the voltage drops below 2.7 V is considered uncharged, and when the voltage is above 3.7 V and the STAT signal of charger chip is enabled, the battery is considered fully

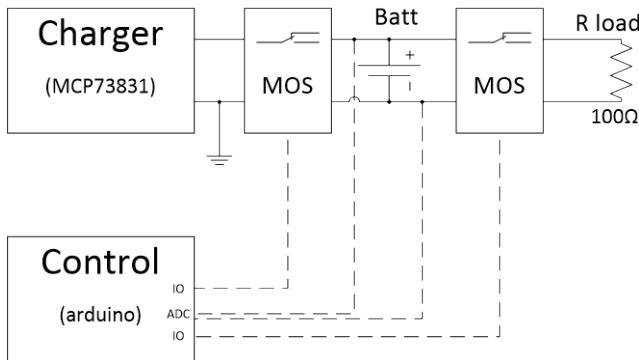


Fig. 4. Wear system.

charged. The control system that is constituted by an arduino platform is able to manage three independent wear testbenches. The evolution of a battery voltage can be seen in Fig. 5.

B. Measurement results

We have performed the measurements with two tmote Sky with different temperatures (low temperature 0-5 °C and ambient temperature 20-30 °C, controlled by a thermostat) and cycles (low wear 0-10 cycles, medium wear 40-50, high wear 200-250). We have repeated the experiments 6 times in each condition. The measurement process acquires V_high, V_low, I_high, I_low, temperature and a timestamp. All measures are transmitted using channel 16 (2.48 GHz), power 0 dBm and a distance between node and base station of 1-2 meters. The measurements are summarized in table 1, and the autonomy in Fig. 6.

The autonomy follows a curve that is similar to the logarithmic function. Therefore we will use the logarithm of the number of cycles as a parameter for building the models. All traces are represented in the Fig. 7; histograms of wear and temperature are on the left, and the voltage on the right.

A first approximation for estimating the SoC considers the voltage extreme points and the temperature. Using V_low, the battery has 3.94±0.04 V when it is fully charged, and 2.53±0 V when it is discharged. This data allow us to calculate (2):

$$SoC(\%) = 70.92 \cdot V - 179.43 \quad (2)$$

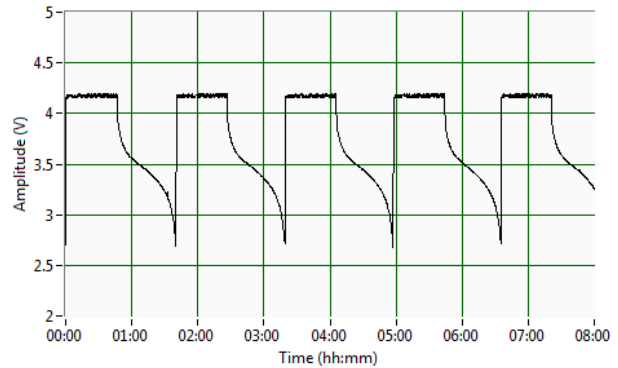


Fig. 5. Evolution of battery voltage.

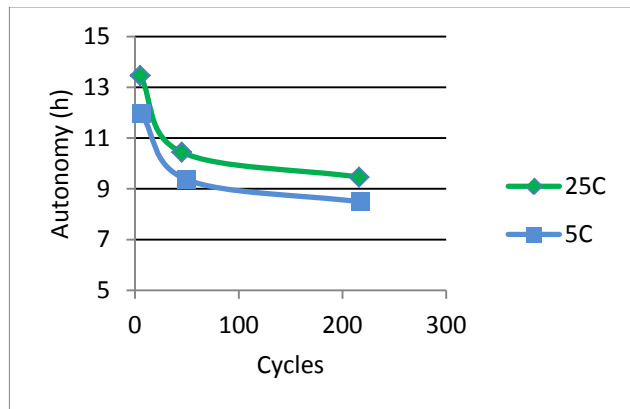


Fig. 6. Evolution of battery autonomy.

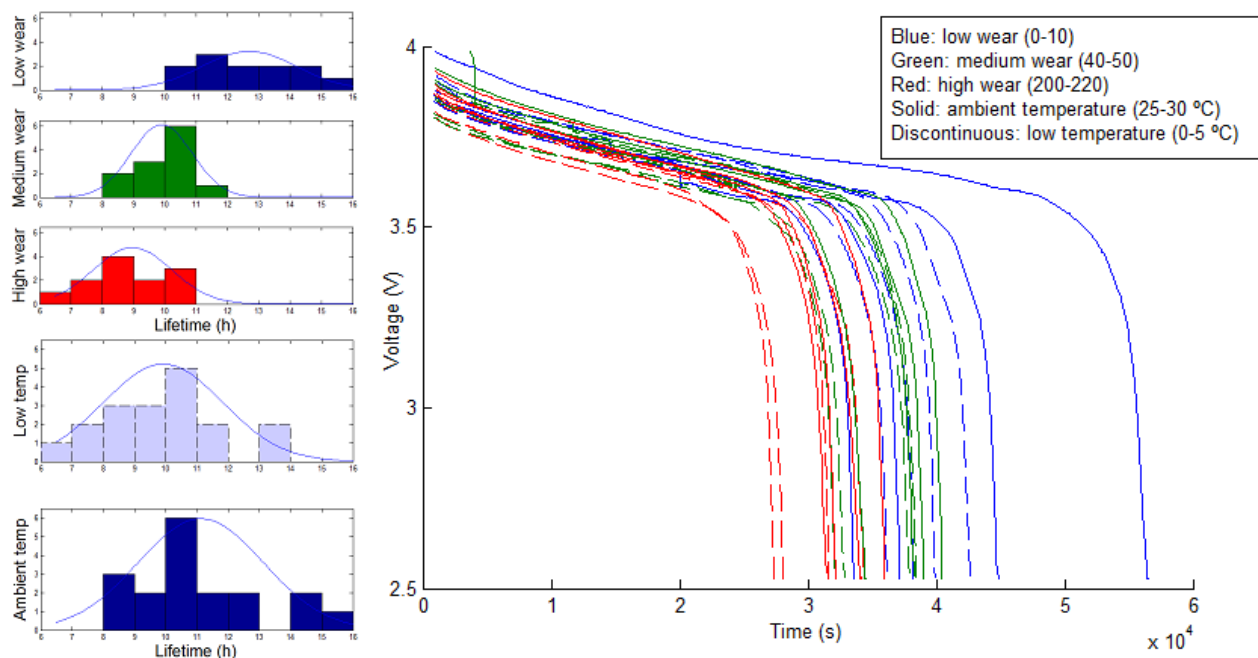


Fig. 7. Measurement distributions & evolution of battery voltage.

V. MODELING

The first step in this section is the pre-processing of the data traces obtained in the stress testbench to eliminate noise and measurement errors. The data is filtered using a second order IIR filter, with a cut-off frequency of 0.02 Hz. Once the data is filtered, several additional parameters extracted from the original data (V, I and temperature) are calculated for each sample. Some of these parameters may not provide relevant information and they will be discarded at a subsequent step. So, the complete list of parameters is:

- V_{high} , V_{low} , I_{high} , I_{low} and temperature. These are the original parameters measured directly in the experiments.
- Exponentiation of V_{low} , with different values: 2 to 5 (V_{low}^2 , V_{low}^3 , etc). These parameters are needed to test polynomial fittings.
- Slope of V_{low} , calculated in a window of 100 samples.
- Exponentiation of slope, from 2 to 5.
- Ten to the power of V_{low} ($10^{V_{low}}$).
- Difference between V_{high} and V_{low} .
- Relationship between voltages and currents (R) at high and low power states as expressed in (1). This parameter is related to the internal resistance, but it should be noted that it is not the real internal resistance since it is not measured in open circuit.
- Cycles of charge/discharge and the logarithm of the value.
- Relative time from the starting time of the trace (this parameter is used to estimate the real SoC and it is not used during the modeling process).

During the experiment, data traces were taken at two different temperatures (low temperature: 0-5 °C; ambient temperature: 25-30 °C) and three different battery states in terms of number of charge/discharge cycles (low wear: 0-10; medium wear: 40-

TABLE I
SUMMARY OF LIFETIME (HOURS) FOR EACH MEASUREMENT. MEAN \pm STANDARD DEVIATION (STANDARD ERROR)

Charge/discharge Cycles	Low temperature (0-5 °C)	Ambient temperature (20-30 °C)
Low wear (0-10 cycles)	11.96 h \pm 1.15 (0.436)	13.47 h \pm 1.45 (0.593)
Medium wear (40-50 cycles)	9.47 h \pm 1.02 (0.415)	10.44 h \pm 0.59 (0.242)
High wear (200-250 cycles)	8.49 h \pm 1.43 (0.584)	8.91 h \pm 1.36 (0.554)

50; high wear: 200-220). Each experiment was repeated six times, and we selected four of them giving a total of 24 data traces. From every trace, 1000 random samples are selected providing a final file with 24000 samples and including 20 parameters for each sample. In each trace, 100 samples out of 1000 are reserved for the last part of the trace, so we can consider that this part is over represented in the final file. All these samples are included because the slope of this part is very high and the duration of this interval is very short. So, it is necessary to take a major density of points in the last segment to have good global results and a good fitting in this part.

It has been used an algorithm based on correlations [37] to determine which parameters provide more relevant information. These parameters will be used to build the models. We have applied the data mining software Weka [38] to evaluate two families of algorithms: regressions and neural networks.

A. Regressions

We have evaluated different linear regression models using one or more parameters to compare different approaches. The simpler case is based on a linear regression that uses only the

voltage level (V_{low}). In this case, the expression obtained from the combination of the experimental data is:

$$SoC(\%) = 88.6 \cdot V - 273.3 \quad (3)$$

A first variant of this equation would divide the traces in segments, due to the different slopes that we have at the first part of the trace and after the cut-off voltage. In the first flat segment of the trace, one fit is calculated and a different regression covers the fast decreasing part of the curve. A decision tree algorithm can be employed in order to select the threshold voltage. This can be calculated with the Weka software using a Decision Stump. The result for our experimental data was 3.6 V. If we divide the samples into two groups with the previous criteria, the following equations are obtained:

$$SoC(\%) = \begin{cases} V > 3.6 \rightarrow 253.7 \cdot V - 884.3 \\ V \leq 3.6 \rightarrow 23.5 \cdot V - 67.2 \end{cases} \quad (4)$$

However, the battery voltage is not necessarily the best parameter. Evaluating subsets of attributes as [37], we obtained that the best parameters are: power of ten, logarithm of cycles and temperature. The regression model using these parameters is:

$$SoC(\%) = -0.18 \cdot Temp + 0.016 \cdot 10^V + 1.13 \cdot \log(Cycles) - 26 \quad (5)$$

Finally, just for the sake of comparing with different possibilities, we built a model using all the parameters and another one using only the 5 main components obtained after a principal component analysis (PCA) transformation. But, in both cases the results were similar or worse than in the previous case, with the side effect of the increment of the computational cost.

B. Neural Networks

The second group of algorithms that we have evaluated are based on the multilayer perceptron (MLP) [39], since it is a model that has been previously used in electric vehicles [40, 41]. In this case, the experimental data must be normalized and the output optionally de-normalized. The neural network can be trained on the PC, and then it can be deployed on the nodes. The neural networks are configured with two layers, 500 maximum epochs to train through, and sigmoid as activation function. The training phase consists on a learning method which minimizes a signal error, its outputs are the neural network's coefficients; this process is performed with weka.

We begin as in the previous case evaluating only V_{low} . In this model, named as MPL1, a two-neurons network is used and it can be expressed as:

$$SoC(norm) = th_2 + k_2 \cdot \frac{1}{1 + e^{-(k_1 \cdot V + th_1)}} \quad (6)$$

Where the coefficients are $k_1 = -10.38$; $th_1 = 5.96$; $k_2 = -1.85$; $th_2 = 0.95$.

Also, we have used a four-neurons network, named as MPL5, which uses the number of cycles (C), temperature (T), relationship between voltages and currents (R), and slope (S). The expression of this model is:

$$\begin{aligned} n_1(norm) &= \frac{1}{1 + e^{-(k_{11} \cdot R + k_{12} \cdot V + k_{13} \cdot T + k_{14} \cdot S + k_{15} \cdot \log(C) + th_1)}} \\ n_2(norm) &= \frac{1}{1 + e^{-(k_{21} \cdot R + k_{22} \cdot V + k_{23} \cdot T + k_{24} \cdot S + k_{25} \cdot \log(C) + th_2)}} \\ n_3(norm) &= \frac{1}{1 + e^{-(k_{31} \cdot R + k_{32} \cdot V + k_{33} \cdot T + k_{34} \cdot S + k_{35} \cdot \log(C) + th_3)}} \end{aligned} \quad (7)$$

$$SoC(norm) = th_4 + k_{41} \cdot n_1 + k_{42} \cdot n_2 + k_{43} \cdot n_3$$

Where the coefficients are:

$$\begin{aligned} k_{11} &= -3.6; k_{12} = -8.98; k_{13} = -0.07; k_{14} = -0.17; k_{15} = -0.04; \\ th_1 &= 6.61 \\ k_{21} &= 0.73; k_{22} = 5.22; k_{23} = -0.41; k_{24} = -0.52; k_{25} = -5.62; \\ th_2 &= -6.7 \\ k_{31} &= -4.4; k_{32} = 8.55; k_{33} = 0.14; k_{34} = -0.55; k_{35} = -6.15; \\ th_3 &= -6.76 \\ k_{41} &= -2.03; k_{42} = -1.33; k_{43} = 1.04; th_4 = 1.07 \end{aligned}$$

VI. VALIDATION AND RESULTS

In the last step, we evaluate the models proposed in the previous section. First of all, we are going to assess the models using the same experimental data that comprises the file with the 24000 samples. This assessment can give an indication of the goodness of each model. Results can be seen in table 2.

Fig. 8 shows the graphical representation of the SoC fitting, where the blue line is the real SoC, calculated from the total duration and the timestamp of each sample, and the remaining lines correspond to each proposed model.

The line calculated from extreme points in (2) has the biggest error, this model only has accurate values at the origin and the end of the line: near 0 and 100%. The line for the V_{low} regression in (3) presents a similar shape than the case before, so it has identical values for correlation, but it has less absolute error because the estimations are bounded in the central part of the graph. The segmented regression in (4) has a very good correlation for most of the range, however the error in the last section is bigger. And it has another main problem, because there may be a discontinuity when voltage crosses the threshold. The two neural networks fit the desired line better in the entire range.

TABLE II.
MEAN RESULTS OF THE PROPOSED MODELS ON THE ORIGINAL MEASURES.

Model	Correlation	Absolute error (SoC)	Squared error (SoC)	Relative abs error (%)	Relative squared error (%)
Eq 2	0.79	30.19	35.11	121.55	121.92
Eq 3 (reg1)	0.79	14.67	17.62	59.08	61.18
Eq 4 (reg segm)	0.97	5.72	7.11	23.04	24.69
Eq 5 (reg3)	0.93	8.63	10.30	34.75	35.75
Eq 6 (MLP1)	0.98	5.20	6.46	20.93	22.42
Eq 7 (MLP5)	0.99	3.41	4.59	13.71	15.93

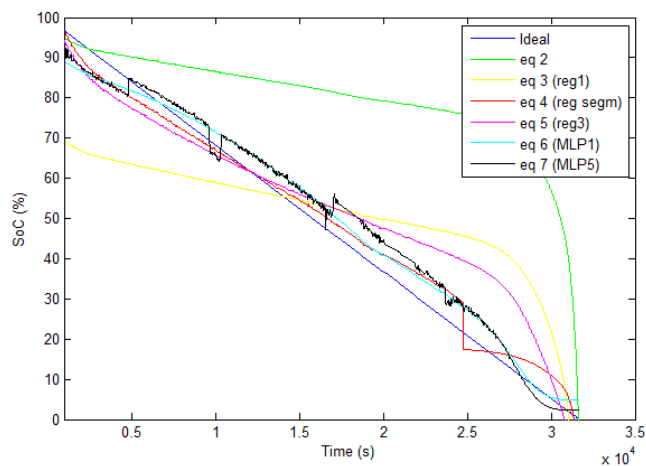


Fig 8. Comparison of fitting between estimations for a representative full operating session.

The estimation of SoC only uses instantaneous measurements, i.e., it does not take into consideration the

previous samples with only the exception of the slope, which uses a window of one hundred samples of the voltage. So the generated models could be used without the knowledge of the previous battery state.

Finally, we carry out a new validation using a different set of measures for studying the accuracy and generality of the models. The new dataset was acquired with different batteries and conditions:

- A new battery of the same type.
- A different battery of 110 mAh.
- Different temperatures (10 and 35 °C) and charge/discharge cycles numbers (20 and 720).

In addition, we compare the results of the new dataset with a commercial battery monitor, a MAX17043 fuel gauge [42]. This IC is a lithium batteries monitor that it does not need a calibration step, it eliminates the current measure and it has temperature compensation.

TABLE III.
RESULTS OF THE PROPOSED MODELS USING A NEW MEASURES DATASET.

Session	Model	Correlation	Absolute error (SoC)	Squared error (SoC)	Relative abs error (%)	Relative squared error (%)
40 mAh, 10 °C, 3 cycles	Eq 2	0.83	32.66	38.61	135.01	138.20
	Eq 3 Reg1	0.83	17.13	19.58	70.81	70.10
	Eq 4 segm	0.98	5.82	7.70	24.08	27.56
	Eq 5 Reg3	0.95	9.89	11.89	40.87	42.57
	Eq6 MLP 1	0.99	7.41	9.04	30.62	32.37
	Eq7 MLP 5	0.99	4.72	5.30	19.61	19.04
	MAX	0.99	7.96	9.87	31.83	34.20
40 mAh, 35 °C, 21 cycles	Eq 2	0.85	33.45	38.64	139.93	139.96
	Eq 3 Reg1	0.85	16.01	18.36	66.97	66.52
	Eq 4 segm	0.99	7.71	8.71	32.27	31.55
	Eq 5 Reg3	0.97	7.61	8.84	31.83	32.02
	Eq6 MLP 1	0.97	9.33	10.60	39.05	38.39
	Eq7 MLP 5	0.99	8.55	9.44	36.17	34.57
	MAX	0.95	6.51	9.38	26.05	32.48
40 mAh, 10 °C, 726 cycles	Eq 2	0.87	31.87	36.40	134.11	132.61
	Eq 3 Reg1	0.87	14.09	16.06	59.28	58.53
	Eq 4 segm	0.99	6.23	6.76	26.23	24.64
	Eq 5 Reg3	0.97	8.11	9.73	34.13	35.45
	Eq6 MLP 1	0.98	6.63	7.76	27.89	28.25
	Eq7 MLP 5	0.99	6.41	7.45	26.96	27.16
	MAX	0.99	5.18	6.80	20.71	23.56
110 mAh, 10 °C, 1 cycles	Eq 2	0.92	35.19	40.28	141.90	140.64
	Eq 3 Reg1	0.93	16.00	18.94	64.53	66.13
	Eq 4 segm	0.98	13.55	14.52	54.64	50.71
	Eq 5 Reg3	0.96	12.76	15.01	52.49	53.43
	Eq6 MLP 1	0.98	12.81	13.82	51.64	48.27
	Eq7 MLP 5	0.99	3.10	4.33	12.76	15.41
	MAX	0.99	4.55	5.02	18.20	17.39

As we can see, all models improve the baseline case (2), and the results are better with more complicated models. The models with better results are the MLP, being comparable to the MAX17043 in accuracy but with the advantage of not requiring an additional hardware.

The only related models that we have found in the bibliography are [14] and [17]. Other models are not suitable for WSN due to the computational cost. The method [14] is Coulomb counter estimator that does not use direct measurements of voltage or current during the mote operation

and its error over a particular application is about 30% of the real SoC. The method proposed in [17] is more comparable. This method is described in section 2 and its result is a quadratic equation, whose coefficients are obtained from a previous fitting. The model only uses voltages (and currents for obtaining the coefficients in the training step), but it does not take into account the temperature or wear cycles directly. Although the method assumes a constant load, and this is not strictly true in our case, we are going to apply this method for comparing with our results. We can simplify the model obtaining the best regression using only the voltages on the low state (V_{low}) from our original traces. The best fitting we have obtained is:

$$DoD = -142.9557 \cdot V_{low}^2 + 861.4243 \cdot V_{low} \quad (8)$$

$$SoC = 100 - DoD \quad (9)$$

This model achieves a correlation over our measurements of 0.93 and a relative square error of 41.8 %. As it was expected, the results are better than in the case of the simple regression, but worse than models with MLP, which can adjust better non-linear behaviours.

We have implemented the above models on TinyOS and the tmote Sky using the floating-point math.h library to assess their computational cost. We have included the current calculation, slope, and other parameters when it is necessary. Also, the normalization of input data has been included in the MLP algorithms. But we have not included the data acquisition, filters and other pre-processing operations, or de-normalization for MLP output because it is an optional step. The measurement was made using the Counter32khz32C component from TinyOS. This component implements a 32 kHz clock using a microcontroller's timer. The computational cost using lineal regression models is 17, 20, and 734 cycles respectively; and with MLP is 249 and 664 cycles respectively. The high cost of 3-parameter regression is mainly due to the float-point exponentiation function. These times could be translated into current consumption in particular conditions. We begin with the actual average current, 954 μ A. If we assume a current consumption of 2.5 mA (MSP430 @ 4 MHz) during the computation of the models in the microcontroller, and we assume that the models are executed each second, the average current would be incremented +1%, +1%, +4.8%, +2% and +4% respectively, as it is shown in table 4. Obviously, in a practical implementation the period for calculating the SoC prediction would be greater and still lower consumptions can be achieved.

TABLE IV.
COMPUTATIONAL AND ENERGY COST OF THE MODELS.

Model	Time (ms)	I average (mA)
Eq 2&Eq 3 Reg1	0.531	0.955
Eq 4 segm	0.625	0.955
Eq 5 Reg3	23	1
Eq6 MLP 1	7.8	0.973
Eq7 MLP 5	16.6	0.994

Finally, we have also evaluated the computational cost of (8). Its implementation using the same previous conditions produces an execution time of 19 cycles (0.593 ms and 0.955 mA), which is similar to the cost of models based on linear regression. With this, we can conclude that our models can accurately predict the battery SoC without a too high computational cost and any additional hardware following list outlines the different types of graphics published in IEEE journals.

VII. CONCLUSION

It has been found that the proposed method is valid for determining the battery SoC in sensor network nodes. The method involves several stages and uses open hardware and software. The first step includes the acquisition of the temperature, the battery current and its voltage during the operation of a tmote Sky under different conditions. The number of charge/discharge cycles of the battery has been taken into account and a testbench for applying a certain number of cycles to the battery in an autonomous way has been developed. Once the data is sampled, the processing stage includes an initial filtering for eliminating errors and noise and the model generation using linear regression and MLP. The models have been posteriorly validated with a new set of measurements and batteries. The conducted experiments have demonstrated the validity of the generated models. In the best cases, with models based on MLP, the results have demonstrated the accuracy of the adjustment, achieving a correlation of 0.99 and an absolute error around 5. Most of the models have a low computational cost and they are suitable for being executed on a microcontroller.

The models have been able of predicting the battery SoC for different batteries and conditions not included during the model training. The influence of the number of charge/discharge cycles has been studied in very extreme values and it is included as a significant parameter in some models. The algorithms proposed can be especially useful for simple motes, which have not battery monitors, like tmote, micaz, etc. This method can be used with other batteries, conditions or specific applications. In addition, it can be part of a simulation algorithm and it can allow an improvement in the prediction of the network lifetime. As it is shown in the results section, the two first models have the worst performance. These models are the simplest and they are widely used, although they may be unsuitable in cases in which a high precision in the SoC estimation is required.

In the future, this work can be extended by adding more parameters and considering others models or algorithms. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

ACKNOWLEDGMENT

REFERENCES

- [1] .Alberola, J., Pelegri, J., Lajara, R., & Perez, J. J. Solar inexhaustible power source for wireless sensor node. In *Instrumentation and Measurement Technology Conference Proceedings*, 2008. IMTC 2008. 657-662.
- [2] .Wang, Q., & Yang, W. Energy consumption model for power management in wireless sensor networks. In *Sensor, Mesh and Ad Hoc Communications and Networks*, 2007. SECON'07. 4th Annual IEEE Communications Society Conference on (pp. 142-151). IEEE.
- [3] .Wang, Q., Hempstead, M., & Yang, W. A realistic power consumption model for wireless sensor network devices. In *Sensor and Ad Hoc Communications and Networks*, 2006. SECON'06. 2006 3rd Annual IEEE Communications Society on (Vol. 1, pp. 286-295).
- [4] .Baoqiang, K., Li, C., Hongsong, Z., & Yongjun, X. Accurate energy model for WSN node and its optimal design. *Journal of Systems Engineering and Electronics*, 2008, 19(3), 427-433.
- [5] .Kan, B., Cai, L., Zhao, L., & Xu, Y. Energy efficient design of WSN based on an accurate power consumption model. In *Wireless Communications, Networking and Mobile Computing*, 2007. WiCom 2007. 2751-2754.
- [6] .Zhao, L., Zhang, W., Xu, C., Xu, Y., & Li, X. Energy-aware system design for wireless sensor network. *ACTA Automatica Sinica*, 2006, 32(6), 892.
- [7] .Zhou, H., Luo, D., Gao, Y., & Zuo, D. C. Modeling of Node Energy Consumption for Wireless Sensor Networks. *Wireless Sensor Network*, 2011. 3(1), 18-23.
- [8] .Wang, Q., & Yang, W. Energy consumption model for power management in wireless sensor networks. In *Sensor, Mesh and Ad Hoc Communications and Networks*, 2007. SECON'07. 4th Annual IEEE Communications Society Conference on (pp. 142-151). IEEE.
- [9] .Norouzi, A., & Serbas, A. An Integrated Survey in Efficient Energy Management for WSN Using Architecture Approach. *International Journal of Advanced Networking and Applications*, 2011, 3(1), 968-977.
- [10] .Bougard, B., Cathoor, F., Daly, D. C., Chandrakasan, A., & Dehaene, W. Energy efficiency of the IEEE 802.15. 4 standard in dense wireless microsensor networks: Modeling and improvement perspectives. *InDesign, Automation, and Test in Europe*, 2008, 221-234. Springer Netherlands.
- [11] .Camilo, T., Carreto, C., Silva, J. S., & Boavida, F. An energy-efficient ant-based routing algorithm for wireless sensor networks. In *Ant Colony Optimization and Swarm Intelligence*, 2006, 49-59. Springer Berlin Heidelberg.
- [12] Tan, H. P., Lee, P. W., Seah, W. K. G., & Eu, Z. A. Impact of power control in wireless sensor networks powered by ambient energy harvesting (WSN-HEAP) for railroad health monitoring. In *Advanced Information Networking and Applications Workshops*, 2009. WAINA'09. International Conference on (pp. 804-809). IEEE.
- [13] .Reddy, B. V., & Reddy, D. R. K. Maximizing the usable battery capacity in WSN with load harmonization. *International Journal of Instrumentation Electrical Electronics Engineering* 2013, 1, 22-25.
- [14] .Castagnetti, A.; Pegatoquet, A.; Belleudy, C.; Auguin, M., An efficient state of charge prediction model for solar harvesting WSN platforms, *Systems, Signals and Image Processing (IWSSIP)*, 2012 19th International Conference on , vol., no., pp.122,125, 11-13 April 2012.
- [15] Kim, N., Choi, S., & Cha, H. (2008, September). Automated sensor-specific power management for wireless sensor networks. In *Mobile Ad Hoc and Sensor Systems*, 2008. MASS 2008. 5th IEEE International Conference on (pp. 305-314). IEEE.
- [16] .Winter, M., & Brodd, R. J.. What are batteries, fuel cells, and supercapacitors? *Chemical reviews*, 2004, 104(10), 4245-4270.
- [17] .Buchli, B., Aschwanden, D., & Beutel, J. Battery state-of-charge approximation for energy harvesting embedded systems. In *Wireless Sensor Networks, Springer Lecture Notes in Computer Science Volume 7772*, 2013, pp 179-196
- [18] .Evanczuk, S. Advanced ICs Simplify Accurate State-of-Charge Measurement for Lithium-Ion Batteries, 2013. Available online: <http://www.digikey.com/us/en/techzone/energy-harvesting/resources/articles/advanced-ics-simplify-state-of-charge-measurement.html> (accessed on 7 1 2014).
- [19] .Barré, A., Deguilhem, B., Grolleau, S., Gérard, M., Suard, F., & Riu, D. A review on lithium-ion battery ageing mechanisms and estimations for automotive applications. *Journal of Power Sources*, 2013, 241, 680-689.
- [20] . Doyle, M., Fuller, T. F., and Newman, J. (1993). Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell. *Journal of the Electrochemical Society*, 140(6):1526.
- [21] . Rohner, C., Feeney, L. M., & Gunningberg, P. Evaluating Battery Models in Wireless Sensor Networks. In *Wired/Wireless Internet Communication*, 2013, 29-42. Springer Berlin Heidelberg.
- [22] .Doerffel, D., & Sharkh, S. A. A critical review of using the Peukert equation for determining the remaining capacity of lead-acid and lithium-ion batteries. *Journal of Power Sources*, 2006, 155(2), 395-400.
- [23] .Barreras, J. V., Schaltz, E., Andreassen, S. J., & Minko, T. Datasheet-based modeling of Li-Ion batteries. In *Vehicle Power and Propulsion Conference (VPPC)*, 2012, 830-835.
- [24] .Rao, R., Vrudhula, S., & Rakhmatov, D. N.. Battery modeling for energy aware system design. *Computer*, 2003, 36(12), 77-87.
- [25] .Technical bulletin Energizer. Battery Internal Resistance, 2005. Available online: <http://data.energizer.com/PDFs/BatteryIR.pdf> (accessed on 18 12 2013)
- [26] .Remmlinger, J., Buchholz, M., Meiler, M., Bernreuter, P., & Dietmayer, K. State-of-health monitoring of lithium-ion batteries in electric vehicles by on-board internal resistance estimation. *Journal of Power Sources*, 2011, 196(12), 5357-5363.
- [27] .Williard, N. D. Degradation Analysis and Health Monitoring of Lithium Ion Batteries. Doctoral dissertation. University of Maryland, 2011.
- [28] .Charkhgard, M.; Farrokhi, M., State-of-Charge Estimation for Lithium-Ion Batteries Using Neural Networks and EKF, *Industrial Electronics, IEEE Transactions on* , vol.57, no.12, pp.4178,4187, Dec. 2010
- [29] .Xtra-power. Rechargeable 3.7V Li-ion Polymer Battery Specification, 40 mAh. Available online: [http://www.xtra-power.com.hk/Products/Li-polymer/Standard%20Type/Xtra-power%20051213\(40mAh\).pdf](http://www.xtra-power.com.hk/Products/Li-polymer/Standard%20Type/Xtra-power%20051213(40mAh).pdf) (accessed on 1 12 2013)
- [30] .Moteiv Corporation. Datasheet tmote sky (2006). Available online: <http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmot-e-sky-datasheet.pdf> (accessed on 1 12 2013)
- [31] .Texas Instruments. Datasheet CC2420 RF transceiver. Available online: <http://www.ti.com/lit/ds/symlink/cc2420.pdf> (accessed on 1 12 2013)
- [32] .Sensirion. Datasheet SHT11 temperature and humidity sensor. Available online: http://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/Humidity/Sensirion_Humidity_SHT25_Datasheet_V2.pdf (accessed on 1 12 2013)
- [33] .Levis, P., Madden, S., Polastre, J., Szewczyk, R., Whitehouse, K., Woo, A., ... & Culler, D.. TinyOS: An operating system for sensor networks. In *Ambient intelligence*, 2005, 115-148.
- [34] .Lajara, R.; Pelegri-Sebastia, J.; Perez-Solano, J.J. Power consumption analysis of operating systems for wireless sensor networks. *Sensors*, 2010, 10, 5809-5826.
- [35] .Microchip. Datasheet MCP73831/2 battery charger. Available online: <http://ww1.microchip.com/downloads/en/DeviceDoc/21984F.pdf> (accessed on 7 12 2013)
- [36] .Choi, S. S., & Lim, H. S. Factors that affect cycle-life and possible degradation mechanisms of a Li-ion cell based on LiCoO2. *Journal of Power Sources* 2002, 111(1), 130-136.
- [37] .M. A. Hall. Correlation-based Feature Subset Selection for Machine Learning. Phd thesis. University of Waikato. Hamilton, New Zealand, 1999.
- [38] .Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H.. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 2009, 11(1), 10-18.
- [39] .Rumelhart, D. E., Hinton, G. E., & Williams, R. J. Learning internal representations by error propagation (No. ICS-8506). ICS Report. California Univ San Diego La Jolla inst for cognitive science, 1985.
- [40] .Gong, Q., Li, Y., & Peng, Z. Power management of plug-in hybrid electric vehicles using neural network based trip modeling. In *American Control Conference*, 2009. ACC'09, 4601-4606. IEEE.
- [41] .Sheikhan, M., Pardis, R., & Gharavian, D.. State of charge neural computational models for high energy density batteries in electric vehicles. *Neural Computing and Applications*, 2013, 1-10.
- [42] .Maxim Integrated. Datasheet fuel gauge MAX17044. Available online: <http://www.maximintegrated.com/datasheet/index.mvp/id/6546> (accessed on 27 12 2013)