



HAL
open science

Temperature-based models of batteries for the simulation of Wireless Sensor Networks

Nabil Kadjouh, Ahcène Bounceur, Abdelkamel Tari, Reinhardt Euler, Loïc Lagadec, Abdelkader Laouid

► **To cite this version:**

Nabil Kadjouh, Ahcène Bounceur, Abdelkamel Tari, Reinhardt Euler, Loïc Lagadec, et al.. Temperature-based models of batteries for the simulation of Wireless Sensor Networks. 3rd International Conference on Future Networks and Distributed Systems, Jul 2019, Paris, France. pp.1-6, 10.1145/3341325.3342013 . hal-02303250

HAL Id: hal-02303250

<https://ensta-bretagne.hal.science/hal-02303250v1>

Submitted on 14 Jan 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Temperature-based models of batteries for the simulation of Wireless Sensor Networks

Nabil Kadjouh
LIMED laboratory
University of Bejaia
Bejaia, Algeria
nabil.kadjouh@gmail.com

Ahcène Bounceur
Lab-STICC CNRS UMR 6285
Université de Bretagne Occidentale
Brest, France
Ahcene.Bounceur@univ-brest.fr

Abdelkamel Tari
LIMED laboratory
University of Bejaia
Bejaia, Algeria
tarikamel59@gmail.com

Reinhardt Euler
Lab-STICC CNRS UMR 6285
Université de Bretagne Occidentale
Brest, France
Reinhardt.Euler@univ-brest.fr

Loïc Lagadec
Lab-STICC CNRS UMR 6285
ENSTA Bretagne
Brest, France
loic.lagadec@ensta-bretagne.fr

Abdelkader Laouid
LIMED Laboratory
University of El-Oued
El-Oued, Algeria
abdelkader-laouid@univ-eloued.dz

ABSTRACT

The main parameter studied in a simulation of a Wireless Sensor Network is the lifetime of the network. In other words, the state of the battery of each node. That is why modelling correctly the battery is very important to obtain realistic results. In real applications, many types of batteries can be considered, where their lifetime depends on the weather variations and on the type of the considered sensor node. In this paper, we present some models of batteries simulated with the *CupCarbon* simulator. The models are obtained by estimating the consumption of real batteries. This is done by studying series of discharging current values with respect to different voltage values and different temperatures. Furthermore, we implement a new module in the *CupCarbon* simulator to allow testing the proposed models and to implement new personal models.

KEYWORDS

Battery modelling, *CupCarbon* simulator, energy consumption, simulation, Internet of things, Wireless sensor network

1 INTRODUCTION

The evolution of the Internet in recent years does not only concern computers but also the objects. These objects are equipped by sensors to collect information about the real world and with the communication antennas to connect to the Internet. The connected objects can include traffic lights, factories, dams, gas meter, phones, etc. The necessity to supervise the environment with the internet gives birth to the field of wireless sensor networks (WSN) which is an interface between the physical environment and the numerical world. The use of the sensors makes possible to monitor the environment and

provide sensed data. It is then possible to follow the state of the objects, their temperature, the presence of the gases, the bacteria, moistures, insects, etc. The appearance of WSN leads to the appearance of new approaches and new challenges. The management of the energy consumption of the sensors and the maximization of their lifetime are the most important challenges since the sensors are supplied by batteries with very limited capacities which are generally not replaceable and non-refillable. Batteries' lifetime depends on several factors such as their design and the weather variations. It is then essential to simulate the batteries before the real deployment of the network in order to study its lifetime according to the environment where it will be deployed. In this context, we will present mathematical models of batteries starting from the study of the discharging current values according to various tension and temperature values in order to calculate their lifetime under climatic conditions similar to the real environment where they will be used. These models are based on the regression method. We have developed and integrated a new module in the *CupCarbon* simulator. It allows to simulate the sensors real batteries. Thus, *CupCarbon* becomes a platform to test the mathematical models of the sensors' batteries under different climatic conditions. This document is divided as follows: Section 2 describes WSNs. In Section 3, we present different battery modelling techniques. In Section 4 we present the *CupCarbon* simulator and the structure of the new module. In Section 5 we describe a case study and we conclude.

2 WIRELESS SENSOR NETWORKS

The progress in the field of microelectronics and the revolution in the field of networks generated a new type of networks called wireless sensor network. They are performant communicative networks with low cost and low energy consumption. They are embedded, autonomous, and self-configurable systems used in various applications (industrial, medical, military and environmental).

A sensor is an electronic component that contains four units [1]:

- (1) **Sensing unit** Generally it is composed of two components: the sensor itself and an analogue-to-digital converter. The sensing unit provides the collected information to the processing unit.
- (2) **Processing unit** Contains a microcontroller equipped with a small memory of storage, which works by a special operating system conceived for microsensors (Tiny OS, for example), it is used to execute the communication protocols which allow collaboration between the network nodes.
- (3) **Transmission unit** The role of this unit is the transmission and the reception of the data.
- (4) **Energy control unit** It is responsible for distributing the necessary energy to the other units.

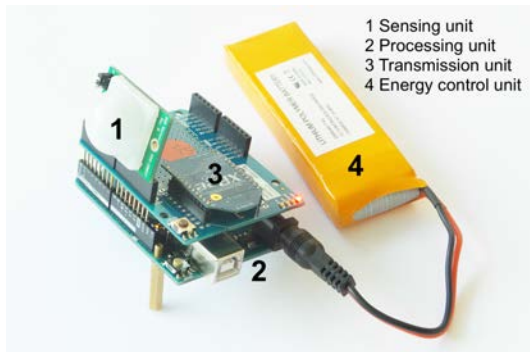


Figure 1: A sensor node Arduino/XBee with a motion sensor unit.

A wireless sensors network (WSN) is composed of a set of nodes of sensors. The position of the nodes is not necessarily predetermined, they can be dispersed randomly in a zone called *interest zone*, or *sensing field*. Each node of the network is able to sense, to collect and to transfer the data by a multi-jump routing to a specific node considered as a point of collection called *sink*. Then the sink transmits these data using Internet or a satellite to a host computer to analyse it and to take a decision.

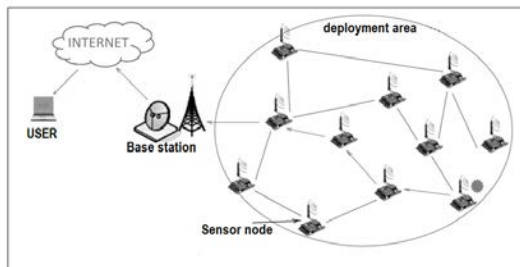


Figure 2: Structure of a Wireless sensor network

The common point between the different WSN applications is the need to have a reliable network while minimizing the energy

costs in order to extend the lifetime of the network. The deployment of the nodes of sensors is generally within inaccessible zones and in most of the cases the replacement or the recharging of the batteries is impossible or difficult, which makes the calculation of power consumption one of the most interesting metric performances of the network, and to minimize energy consumption will be the most important challenge in this technology. In a WSN, energy consumption is an important parameter to take into account during the deployment of the network which makes the simulation of this type of network an essential task especially if the number of the nodes is very important. It is therefore necessary to study the environment in which our network will be deployed and to observe the behaviour of the nodes in the simulator before the real deployment. In order to facilitate the design and the study of a WSN, several tools for simulation have been created. There exist a big number of tools of simulation and modelling of the WSN, the authors in reference [1][2][14] present a set of 36 simulators and emulators, such as Castalia, TOSSIM, COOJA/MSPSim, OMNET++, J-Sim MiXiM, NesCT, PAWiS, SENSIM, Viptos, VisualSense, Ataraya, WSNet, J-Sim, and *CupCarbon*. Simulation makes it possible to recreate a real and complex physical scenario by the execution of a computer program on a computer or a network, whereas the emulation seeks to substitute material by software [4].

3 BATTERY MODELING IN WSN

The lifetime of the battery depends mainly on the speed of consumption of apparatus energy; in the periods of strong consumption of energy the battery loses its effectiveness and its capacity but in the periods of rest the battery can recover a part of its capacity loss: thus the battery lifetime will be extended [5].

3.1 Battery modelling

A battery is composed of one or more cells that are connected in parallel or series, where cells contain a chemically stored energy which is converted into electrical energy by an electrochemical reaction. A cell is composed of an anode, a cathode and an electrolyte, which separates the two electrodes.

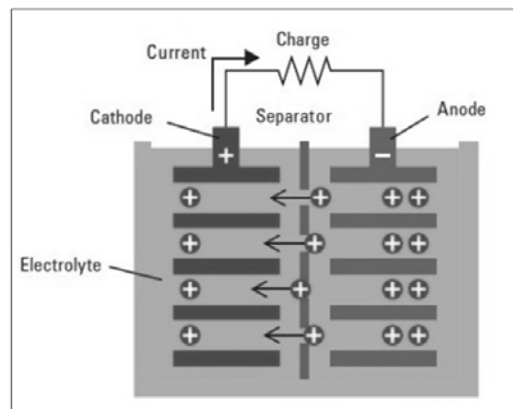


Figure 3: Structure of a battery

Various batteries are available today on the market, which proves the technological progress in this field. Battery technology has advantages and disadvantages and it is adaptable to a specific application, where each type differs from the other by the chemical component. Among these types we find [6]: Acid lead, Valve-Regulated Lead Acid (VRLA), Nickel Cadmium (Ni-Cd), Hydride Nickel-Metal (Nor-MH), Lithium Ion (Li-Ion), Sodium Sulfur (Na-S), Mono-sulfide Lithium-Iron, Nickel Zinc (Refillable Alkaline System), Metal Air Zinc-Bromine, Sodium-Beta and others. The following electric properties are generally employed to characterize a battery, its capacity, its tension, the maximum current that it can support and its lifetime. One of the most important characteristics of a battery is its capacity, which represents the quantity of current which can be extracted from an electrode via the electrolyte and active materials of the electrodes, on a discharge. This capacity is expressed in ampere-hour, Ah (1 Ah = 3600 coulombs). We often use another unit, the watt-hour Wh (1 Wh = 3600 J).



Figure 4: Characteristics of a battery

Figure 4 shows a Li-ion battery, having a capacity of 500mAh and a Tension of 3.7V. If a battery capacity is given by C (Ah) and its current by I (A) then its autonomy in hours (h) is given by T as follows:

$$T = \frac{C}{I} \quad (1)$$

Example

If $C=800\text{mA}$ and $I=10\text{mAh}$, the autonomy = 80h (less than 4 days).

In the literature, many models of batteries can be found; various approaches were used to model the properties of the battery varying from electrochemical very detailed models to the high level stochastic models. We found: electrochemical models, circuit-electric models, analytical models, Kinetic models, Stochastic models [5] [6] [7] [8], modeling by fuzzy logic [9], The Kalman filter [10], and the artificial neurological network [11].

3.2 The regression

The study of the various values of current according to the temperature and the tension leads us to use one of the techniques of predicting future value on the basis of the existing data; it is the

technique of regression. The most known regressions are Linear, Logarithmic, Exponential, Polynomial and Power [13].

3.2.1 Linear regression. Linear regression is a straight line which generally represents an increase or a regular reduction:

$$y = mx + b \quad (2)$$

where m is the slope and b is the ordinate.

3.2.2 Polynomial regression. Polynomial regression is generally used to represent fluctuations of data. The order of the polynomial curve can be given by the number of fluctuations in the data or by the number of curvatures of the curve

$$y = b + c_1x + c_2x^2 + \dots + c_nx^n \quad (3)$$

where b, c_1, c_2, \dots, c_n are constants.

3.2.3 Logarithmic regression. Logarithmic regression is the most exact possible curve which proves to be the most useful when the frequency of modification of the data increases or decreases quickly and then stabilizes itself.

$$y = c \ln x + b \quad (4)$$

where c and b are constants and \ln is the Neperian logarithm function.

3.2.4 Power regression. Power regression a curve particularly adapted to the data set that compares measures which increase at a specific speed. You cannot create such a curve if your data contain zero or negative values.

$$y = cx^b \quad (5)$$

where c and b are real constants.

3.2.5 Exponential regression. Exponential regression is used when the data values increase or decrease always more quickly. You cannot create a curve of the exponential tendency for data containing the zero or negative values.

$$y = c e^{bx} \quad (6)$$

where c and b are constants.

3.2.6 The coefficient of determination. Regression is more precise when its coefficient of determination (r^2) is equal or close to 1. r is the correlation coefficient of Pearson and given by Equation (7).

$$r = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sqrt{\sum(x - \bar{x})^2(y - \bar{y})^2}} \quad (7)$$

where \bar{x}, \bar{y} are samples averages.

3.3 Modelling by regression

3.3.1 The used data. We are interested in certain types of sensors which are used in the design of Smart city projects such as:

Concertina, MygaleS4, SPI-108, MCam, Oldham OLCT 50, Oldham OLCT 80 and Oldham TX12. The data are results obtained by tests for various types of sensors. Tests for each sensor were carried out in various temperatures under various tensions. Each sensor has a card which shows the various values of the tension and the values of the temperature as well as the values of the correspondent measured current.

3.3.2 *Work proceedings.* We took as sample the average curve obtained by the calculation of the median values of currents in the various temperatures; we calculated the equations of the various regressions and among these curves, we have chosen the best by the calculation of coefficients of determination r^2 . We obtain the equation of the curve of tendency in the form:

$$I = f_1(V) \quad (8)$$

For the moment we have only a function which turns a value of the current according to the tension given in average temperature. To calculate the measured value of the current in an unspecified temperature x it is necessary to calculate the current distance between this temperature x and the average temperature. This distance is given by:

$$Dist_x = DU * (t_x - t_{moy}) \quad (9)$$

where t_x is the given temperature, t_{moy} is the average temperature and DU is the unit distance, i.e., the difference of current for two successive temperatures:

$$DU = \frac{average((max - min)_i)}{t_{max} - t_{min}} \quad (10)$$

Finally, we formulate the models in the following form:

$$I = f_1(V) \quad (11)$$

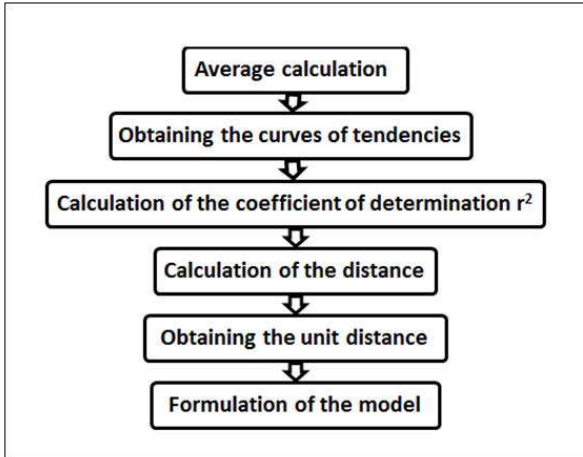


Figure 5: Work proceedings

3.3.3 *Results.* By the application of this technique we obtain the models represented in Table 3.

3.3.4 *Discussion.* Concerning the modelling of the Concertina sensor we found the best model by use of polynomial regression with a total of errors equal to 11.03 for the 30 cases and an average error of 0.37. In the case of Mygale4S sensor, we found the best model by the use of power regression with a total of errors equal to 21.80 for the 50 cases and an average error of 0.44. For the Oldham OLCT 50 sensor, we found the best model by the use of power regression with a total of errors equal to 29.01 for the 120 cases

Sensor type	Model
Concertina	$I = 0.0197 * V^2 - 1.4764 * V + 44.588 + (T - 17.5) * 0.0273$
MyGale4S	$I = 221.5 * V^{-0.632} + (T - 17.5) * 0.041$
Oldham Olct 50	$I = 644.72 * V^{-0.831} - ((390.26 * V^{-1.707}) * (T - 16.5) / 75)$
Oldham Olct 80	$I = 0.0724 * V^2 - 4.0504 * V + 94.723 + (T - 17.5) * 0.025$

Table 1: Results.

and an average error of 0.24. Finally, for the modelling of Oldham OLCT 80 sensor we found the best model by the use of polynomial regression with a total of errors equal to 71.47 for the 91 cases and an average error of 0.79.

4 INTEGRATION IN CUPCARBON

4.1 Description of the CupCarbon simulator

CupCarbon is an open source platform for designing and simulating Smart-City and Internet of Things Wireless Sensor Networks. It is programmed in the Java language. Figure 6 shows the main interface of this simulator.

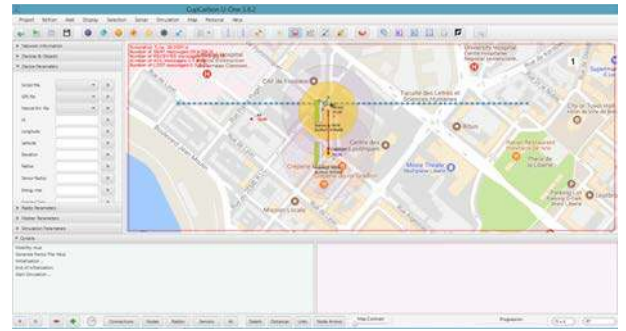


Figure 6: Main IHM of the CupCarbon simulator

Its objective is to design, visualize and validate distributed algorithms for monitoring, environmental data collection, etc. and to create environmental scenarios such as fires, gas, mobiles. It can be used for educational and scientific projects. It offers two simulation environments: A multi-agent environment [15], which enables the design of mobility scenarios and the generation of events such as fires and gas as well as the simulation of mobiles such as vehicles and flying objects [14]. A discrete event simulation of wireless sensor networks takes into account the scenario designed on the basis of the first environment. Networks can be designed on a geographical map using the OpenStreetMap (OSM) framework to deploy sensors directly on the map. For programming and configuring each sensor node individually, a scripting language called SenScript is included in the simulator. *CupCarbon* is more focused on the application layer compared to the other layer which makes it a real complement to the other simulators [16]. In order to test and validate the battery models in the *CupCarbon* simulator we improved its architecture by adding a new module named "Weather event" and also some modifications in the other modules especially on "BatteryModel" and "Discrete Event Simulation".

4.1.1 *Weather event module.* To simulate the batteries under climatic variations we need a temperature files management system, for which we built a system useful to import, affect, validate, modify and create a temperature file. This file contains temperature information about the environment during the simulation. It must be composed of two fields: the duration in seconds and the temperature value in °C. The duration represents the time difference between two events of temperature: previous and current events. The temperature file takes as extension .wsc. Figure 7 shows an example of a fragment of a temperature file. Our system is an inter-

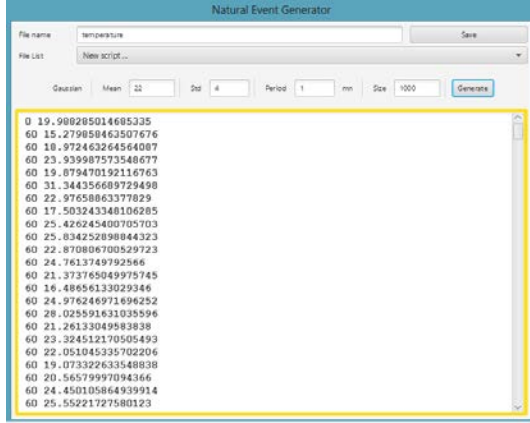


Figure 7: A fragment of temperature file

face between the process of simulation and the file of temperature. From the value of the simulation current time it calculates the corresponding temperature and it classifies the temperature change event in the list of events to execute them.

4.1.2 *Battery model module.* In the *CupCarbon* simulator we used files for battery models. Such a file contains information about battery characteristics such as model name, battery capacity (in mAh), battery tension in Volts and the discharge current formula which is of the form: $I = f(T, V)$ as shown in the modelling section. A model file of the battery must respect this last format and also take the extension .bmf. We have developed a battery model system manager to create, remove, modify, import and assign it to the corresponding sensor. This system is an interface between the process of simulation and the various models. Starting from the discharge duration calculated in the simulation and the current temperature it calculates the power consumption based on the battery model characteristics and includes the discharge current formula.

4.1.3 *Discret event simulation module.* In the Discrete Event Simulation module we have added the option of simulation under climatic conditions. This option uses the services offered by the two studied managers: battery model manager and temperature file manager to calculate the sensors battery lifetime under weather variations of the environment in which will be installed. The temperature change was included in the events list as an environmental event.

4.2 How to calculate consumption at a given temperature ?

The capacity of a battery is the quantity of the electrical energy after reception of a full charge. It is usually expressed in milli-amperes-hours (mAh). If the battery capacity equals 1000mAh, this signifies that this battery will be exhausted after 3600 seconds (1 hour) under a discharge current equal to 1000mA. The power consumption depends on the discharge current, the data size and the duration of the transmission, which means that

$$Global_e(joule) \longrightarrow Capacity(mAh) \quad (12)$$

$$Consumed_e(joule) \longrightarrow Current(mA) * Datasize * Trans-time(second) \quad (13)$$

which gives by the application of the rule of three on (12) and (13)

$$Consumed_e = \frac{Global_e * Current * Datasize * Trans - time}{Capacity} \quad (14)$$

According to the models found, the current is a function of the temperature and the voltage knowing that the voltage is constant and depends on each type of battery by replacing the variables with the corresponding values which are:

- Global energy = 19160 Joule
- Data size = 1024 bits
- Transmission time = 0.004 second
- Capacity = 2200 mAh
- Voltage = 4.7 volts

Table 2 represents the models of energy consumption we obtained for each sensor type.

Sensor type	Model
Concertina	$E = 0.000273 * (Temperature - 17.5) + 0.38084$
MyGale4S	$E = 0.00041 * (Temperature - 17.5) + 0.83293$
Oldham Olct 50	$E = 0.00028 * (Temperature - 17.5) + 1.78179$
Oldham Olct 80	$E = 0.00025 * (Temperature - 17.5) + 0.77285$

Table 2: Models of energy consumption.

5 CASE STUDY

The example below shows the energy consumption of each of these models by comparison with the classic model integrated into the simulator.(cf. Figure 8). For this, we have chosen five sensors of the same type to execute the same script at the same time, and for each sensor we have assigned a different consumption model among the obtained models. Every 200 seconds we have calculated the energy consumption for each model, and the results obtained are represented in Table 3 whose corresponding graph is given in Figure 9.

In general, the models that depend on the temperature are more consumed than the default model integrated with the simulator and the consumption of each model follows a growing line. The Classical model consumes 4 joules in the first 200 seconds and for each time an increase of 3 joules every 200 seconds for the Concertina model, the consumption in the first 200 seconds equal to 17 joules and each time an increase average of 15 joules each

200 seconds, the consumption of the two models MyGaleS4 and OldHamOLCT50 are very close, in the first 200 seconds the consumption is between 33 and 36 joules and each time an increased average between 30.6 and 32.8 joules every 200 seconds. On the other hand, the OldHamOCT50 model is the most consumed among the other models with an increased average of consumption between 66 and 77 joules. This different energy consumption between the study models means that energy consumption depends on many factors which are:

- Climatic conditions such as temperature and humidity.
- The operating conditions: the size of transmitted data, speed and transmission distance and power of the transmission.
- Factors depends on the battery itself like: state of the battery, current of discharge, capacity, operating voltage and the battery ageing.

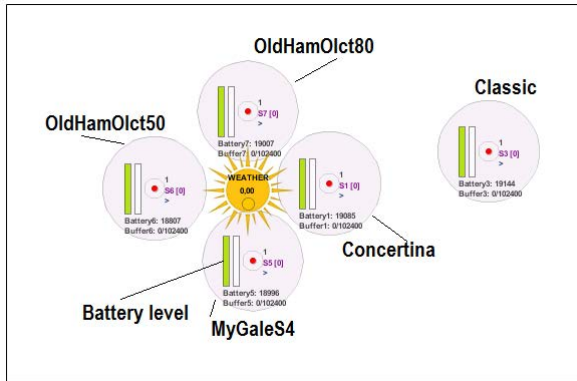


Figure 8: Simulation with different energy consumption models

Time(s)	Classic	Concertina	MyGaleS4	OldHamOLCT50	OldHamOLCT50
0	0	0	0	0	0
221	4	17	36	77	33
424	7	31	69	147	64
625	10	46	101	216	94
830	13	61	134	287	124
1020	16	75	164	353	153

Table 3: Results of simulation of different energy consumption models.

6 CONCLUSION

In this work, we tried to clarify the manner we presented wireless sensor real batteries simulation under different climatic conditions by the *CupCarbon* simulator. For that, we have built models of real batteries by the study of the discharge current values according to different tension and temperature values. We used the of regression technique to build the mathematical models and we found a minimal average error equal to 0.24 in the battery of sensors Oldham OLCT 50, but the maximum average error is 0.97 in the battery of sensors Oldham OLCT 80. After that we have developed a new module and

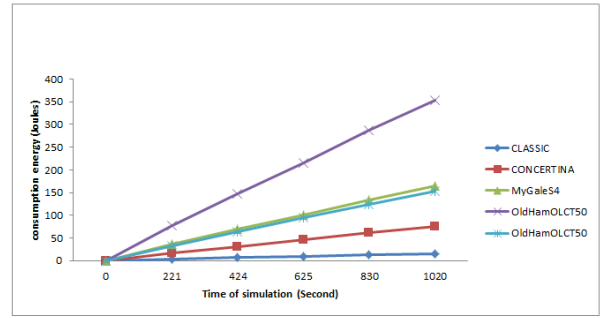


Figure 9: Simulation results of different energy consumption models

integrated into the *CupCarbon* simulator to test and validate the real sensor battery models under meteorological variations.

REFERENCES

- [1] PRESS, C. R. C. Wireless sensor networks: Current status and future trends. CRC press, 2016.
- [2] M. Jevtic, N. Zogovic, Evaluation of Wireless Sensor Network Simulators , In 17th Telecommunications forum TELFOR 2009, Serbia, Belgrade, November 24-26, 2009.
- [3] P. Chimwal, D. Singh Rai, D. Rawat, Comparison between Different Wireless Sensor Simulation Tools, IOSR Journal of Electronics and Communication Engineering (IOSR-JECE), PP 54-60, 2013.
- [4] A. Bounceur, Modèles et Simulation pour le Test de Circuits Mixtes, la Fouille de Données et les Réseaux de capteurs sans fil, HDR / University of Brest, 2014.
- [5] M.R. Jongerden, B.R. Haverkort, Battery Modeling, 2008.
- [6] A. Davoudi, R.S. Balog, Batteries, Battery Management, and Battery Charging Technology, Springer, 2013.
- [7] B. Buchli, D. Aschwanden, J. Beutel, Battery State-of-Charge Approximation for Energy Harvesting Embedded Systems, Springer, 2013.
- [8] Christian Rohner, Laura Marie Feeney, and Per Gunningberg, Evaluating Battery Models in Wireless Sensor Networks, Springer, 2013.
- [9] Jingliang Zhang, Jay Lee, A review on prognostics and health monitoring of Li-ion battery, ScienceDirect, 2011.
- [10] Dinh Vinh Do, Diagnostic de batteries Lithium ion dans des applications embarquées, Université de Technologie de Compiègne, 2010.
- [11] Chang, Wen-Yeau, Estimation of the state of charge for a LFP battery using a hybrid method that combines a RBF neural network, an OLS algorithm and AGA, Sciencedirect, 2014.
- [12] <http://www.cvc.u-psud.fr/spip.php?article171>, 2016
- [13] <https://support.office.com/en-us/article/Add-change-or-remove-a-regression-in-a-chart>, 2016.
- [14] K. Mehdi, M. Lounis, A. Bounceur, and T. Kechadi, Cupcarbon: A multi-agent and discrete event wireless sensor network design and simulation tool, In IEEE 7th International Conference on Simulation Tools and Techniques (SIMUTools'14), Lisbon, Portugal, March 17-19, 2014.
- [15] M. Lounis, K. Mehdi, A. Bounceur. A CupCarbon Tool for Simulating Destructive Insect Movements. International Conference on Information and Communication Technologies for Disaster Management, Algeria, Mar 2014.
- [16] CupCarbon simulator, <http://www.cupcarbon.com>