NiCd battery type parameter estimation using a hybrid neuronal approach

L MERAD*, M. BEKHTI*, N LARBI* and A. BOUTTE*
*Division of Space Instrumentation
National Centre of Space Techniques
PO 13, 01 Avenue of Palestine, Arzew (31200)
ALGERIA
Fax: +213 41 47 36 65

Abstract – In this study, our principal objective is the estimation of the parameters of a NiCd battery type by using the artificial neural networks. The idea consists in using a hybrid training based on the evolutionary algorithms and the method of Levenberg-Marquardt. The results of simulation show the good founded of the used technique.
Keywords – battery, microsatellite, temperature, voltage, estimation, neural networks.

1 Introduction

Many problems, where a system must react to the measures done, necessitate the presentation of the measured phenomena in the form of a mathematical model dependant of a certain number of parameters. The form of the model is deduced from the physical aspect of the studied phenomena and takes usually into account simplifications so that to avoid the design of implementations excessively complex. In general, the model does not reflect the reality because the measurements are in many cases affected with noise and the use of a more complicated model is not practically interesting regarding a simplified model.

To estimate a physical system parameters, many methods were developed such as the minimization quadratic criteria algorithm [1], Kalman estimator [2] and the artificial neural networks [3].

The artificial neural networks allow not only to establish important analytical relations for the estimation phase, but also a great flexibility, since there is no restriction with regard to the number of the system's parameters as input and output.

The estimation phase becomes then possible because of the explicit relation given by the neural networks [3]. The interesting thing about the use of neural networks resides in their capability to estimate the battery parameters taking into account the predefined general criteria.

In this paper, we will estimate the temperature and the battery voltage of the Alsat-1 microsatellite over a short period using the artificial neural networks. For this, it is possible to construct the neural networks during the learning phase basing on a set of solutions obtained from the telemetry data of Alsat-1.

In our application, we used a hybrid neuronal approach based on evolutionary algorithms so that to do a pre-learning on the weights and biais. The solution obtained provides an initialization point for a learning algorithm minimizing the error function by steepest descent.

2 Battery

The batteries revolutionized the way of storage of the electric power and they allowed a very great mobility and independence of the sector fixes, the portables, systems statement to the space equipment (technology of point) [4].

They represent a device which converts chemical energy, stored in these active materials, in an electric power, by the means of an electrochemical reaction of oxydoreduction (redox).

The battery designed for space applications is called battery "without maintenance", whose electrochemical cell is closed.

It comprises:

- a positive electrode made up **of nickel hydroxide**. This active matter being located in reception facilities out of sintered nickel (the most current solution currently) also playing the role of collector of current.
- a negative electrode made up **of cadmium hydroxide.**
- an electrolyte made up of **potash** immobilized in the form of freezing (silica Addition on high specific surface), or retained in a glass fiber separator to high capillary capacity (**MGA, Absorptive Glass Chechmate)**.

  The gases produced during **the gassing** remain prisoners in freezing and are recombined during the discharge. The water consumption and the gas emission are thus extremely low.

The batteries Ni-Cd present a rather constant voltage until a great depth of discharge, and their capacity like their internal impedance, depend little on the temperature and their state of load. One determines the performances of the battery through these parameters has to know: the voltage, capacity, duration of discharge, temperature, ect… [4].

The battery of the microsatellite Alsat-1 comprises 22 cells **SANYO N-400drl** available in the trade at **SANYO**. Its capacity is of 4Ah. The cells are selected so that their voltages do not differ much during the cycles from load and discharge.

To ensure the thermal control of the cells, they are gathered in an aluminum case which makes 250 mm length, 180mm of width and 80mm height.

This case consists of 4 lines, two lines consist of 5 cells and the two others made up of 6 cells inter-connected in series such as figure 1 shows it. In order to allow the compensation of temperature, 2 thermistors are stuck on the aluminum case.

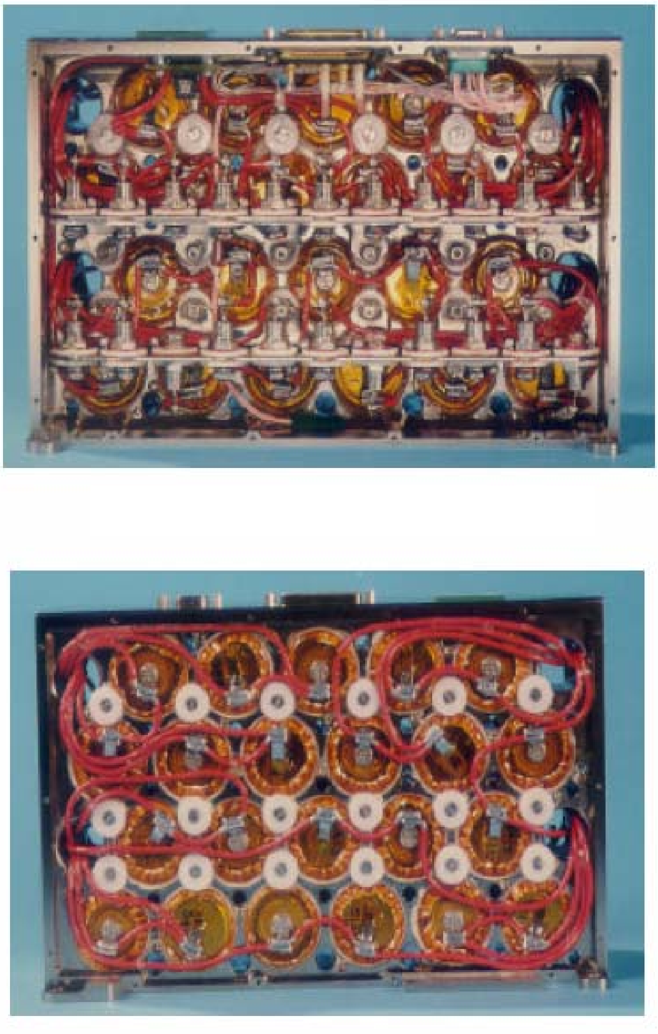

Fig. 1. Alsat–1 Battery

3. Artificial Neural Networks

The artificial neural networks (ANN) are mathematical tools able to approach the nonlinear relations to significant degrees of complexity. The space of input X of dimension n (Level of the input) is connected to the space of the output Y of dimension m (Level of the output) by the intermediary of a hidden level. This level has a fixed number of neurons, but which varies from a study to another according to the complexity of the problem (figure 2) [3].

In the present study, an architecture is considered; multi-layer perceptron networks with backpropagation of the gradient. They are probably one architectures most current and simplest non-linear networks. The capacities of modeling of these networks are analyzed.

The multi-layer perceptron networks are composed of a input layer whose neurons code the information presented at the network, of a variable number of internal layers called "hidden" and of a input layer (figure. 2) containing as many neurons as of desired responses. The neurons of the same layer are not connected between them. The training of these networks is supervised. The algorithm used during this training is known under the name of method of Backpropagation learning *BPL*. This method of training is divided into two phases:

A phase of propagation, which consists in presenting a configuration of input at the network then to propagate this input gradually layer of input to the output layer while passing by the hidden layers.

A phase of backpropagation (figure 3), which consists, after the process of propagation, to minimize the error made on the whole of the examples presented, error considered as a function of the synaptic weights. This error represents the sum of differences squared between the calculated responses and those desired for all the examples contained as a whole of training.

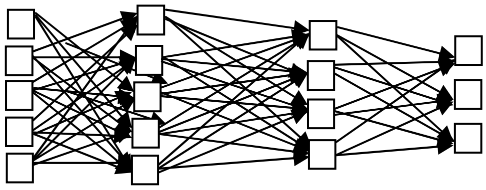

Fig. 2. Multi-layer networks

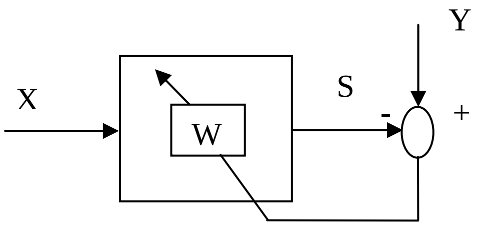

Fig. 3. Training by backopropagation of gradient BPL

3.1 Formalism

One considers a network of ***n*** input neurons and ***m*** output neurons with an unspecified number of hidden layers.

The following notations are used:

$X = (X_1, X_2 ....., X_n)$ is the vector of the inputs.

$Y = (Y_1, Y_2 ....., Y_m)$ is the vector of the desired output.

$S = (S_1, S_2 ......, S_m)$ is the vector of the output actually obtained.

$e(t)$ is the step of the gradient at the stage t.

The modification of the synaptic weights, is given by the following formula [5]:

$$w_{ij}(t) = w_{ij}(t-1) + \Delta w_{ij}(t) \qquad (1)$$

In the case of *an off-line* adaptation the modification of the weights is done only after having completely presented the whole of training:

$$\Delta w_{ij}(t) = -e(t) \cdot \frac{1}{N_s} \sum_{k=1}^{N_s} \frac{\partial E^k}{\partial w_{ij}} \qquad (2)$$

with

$$E^k = (Y^k - S^k)^2$$

N_S: a number of examples contained in the data base of the training.

However, *the on-line* adaptation adjusts the weights of the network as the vectors of training are presented:

$$\Delta w_{ij}(t) = -e(t) \frac{\partial E^k}{\partial w_{ij}} \qquad (3)$$

There are other methods allowing the calculation of the synaptic weights connecting each neuron such as the evolutionary algorithms (EA) [6]. The EA make it possible to carry out a nondeterministic total research. The algorithms carrying out a local research are more inclined to converge towards no optimal solutions but requires less memory allocations. Moreover, if the point of current calculation is far away from the solutions locally optimal and located apart from a plate (on a plate, the function varies very little) the information of the gradient makes it possible to know quickly and with precision the position of an optimal solution. In our application, we used the differential evolution for pre-learning. The method consists in making evolve a population of real vectors. A news solutions are created with each generation by using a mechanism of mutation, recombination and selection [6].

Generally, the stages of construction and validation of the neural networks are divided into four phases:

- choice of the input of the networks,
- choice of the outpu of the networks,
- choice of the total architecture of the studied networks,
- tests of the networks selected on new examples close to the examples of training.

In our applications, we called upon the Matlab software, "neural networks Toolbox" in order to carry out these stages [7].

4 Applications

In our study, we carried out a pretreatment of the data in order to select, in the space of representation "information" necessary to the application. This selection passes by the date conversion into a numerical value (exp: October 18, 2005 corresponds has 1, October 19 corresponds has 2, so on until January 12 2006).

We built a data file corresponding to our study, for each day, corresponds the temperature and the voltage of the battery taken to 12h00.

4.1 Training phase

Our network is a network multi-layer composed of four layers (figure 4):

- A input layer: it includes only one neuron representing the day.
- Two hidden layers: composed each one respectively of 18 and 12 neurons.
- A input layer: composed of 2 neurons representing the temperature and the voltage of the battery

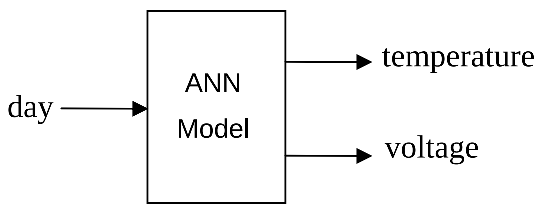

Fig. 4. *Representation of the estimation model by the neural networks with 1 input and 2 outputs*

Once that the network architecture was decided, the learning phase makes it possible to calculate the synaptic weights driving with each formal neuron. For this application, we used the combination of two algorithms (Differential Evolution and Levenberg-Marquardt) whose source code in Matab is available on Internet [8]. The DE will be used as algorithm of pre-learning for Levenberg-Marquardt. The DE tries to obtain a good solution by reducing the space of research gradually. The algorithm of Levenberg-Marquardt uses this solution to find a solution optimal more precise.

The essential objective here is to find the best training which makes it possible to give a good model. For that, several tests are necessary, while acting on the parameters influencing the training or learning. The parameters obtained are as follows for the two algorithms:

- **Differential evolution**

- the sigmoid function is affected like function of activation to the hidden layers and the linear function with the output layer.
- recombining constant: 0.8
- mutation constant: 0.55
- number of epochs: 500
- final Error: 10^{-4}

- **Levenberg-Marquardt**

- the sigmoid function is affected like function of activation to the hidden layers and the linear function with the output layer.
- learning rate: 0.15
- number of epochs: 500
- final Error: 10^{-4}

The training phase is illustrated by figure 5. This figure represents the evolution of the mean square error between the output of the ANN and the samples given according to the number of epochs. The final error obtained is 9.7569 10^{-5} after 352 epochs.

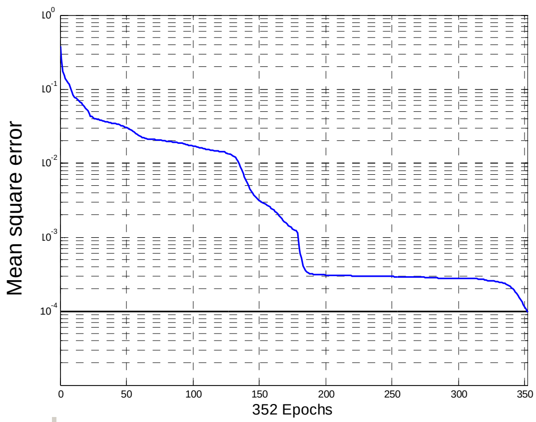

Fig. 5. Training phase of ANN

4.2 Use Phase

Once the training phase is finished, we plotted the evolution of the temperature and the voltage of the battery during the period (October 18, 2005 until January 12, 2006). One notices a clear agreement between the neuronal approach and the measurements obtained by telemetry (figures 6 and 7).

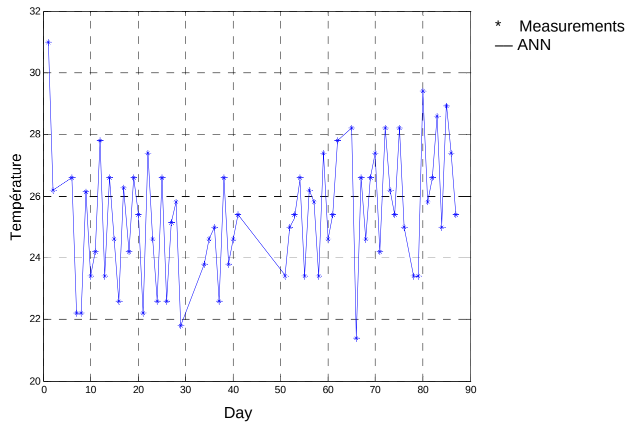

Fig. 6. Evolution of the temperature

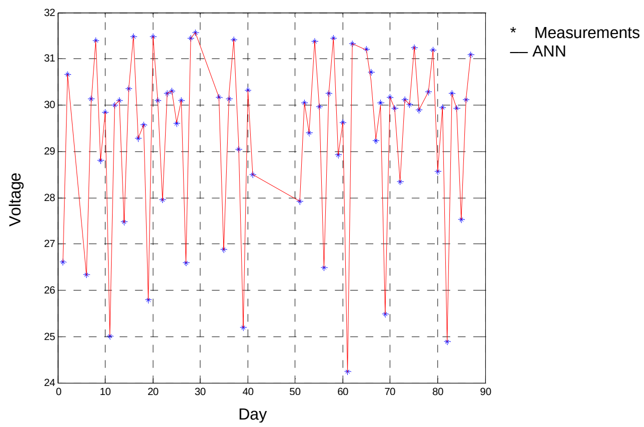

Fig. 7. Evolution of the voltage

5 Conclusion

In our study, we have examined the possibilities of the estimation of the parameters of a NiCd battery type by using the neuronal approach. The results obtained by this approach are encouraging.

We have used an evolutionary algorithm allowing a pre-learning with an aim of obtaining a solution constituting an effective point of initialization for the algorithm of Levenberg-Marquardt. Applied to the training of the multi-layer networks for the estimation and the battery parameters, our approach was shown very powerful.

The results are satisfactory and show the interest of the application of the neural networks in the field of the estimation of the parameters of the battery. This interest comes from their capacity of approximation, training, modelling and optimization of the nonlinear models. However, the multi-layer neural networks with backpropagation of the gradient present the disadvantage of slowness due to the phase of training, which depends on the number of inputs and examples used.

In spite of these disadvantages and once the finished training, the neural networks allows to redure computing time for the estimation of the battery parameters at the use phase.

The neural network can of course be built and enriched with experimental data.

References

[1] J. Le Roux, Estimation des paramètres d'un modèle par la méthode des moindres carrés récursifs, (2000).
[2] G. Welch and G. Bishop, Introduction to the Kalman Filtres, University of North Carolina at Chapel Hill, (2002).
[3] S. Haykin, Neural networks a comprehensive foundation, Prentice Hall, 1999.
[4] D.Linden, Handbook of Batteries, 2end edition, MacGraw-Hill Edition, (1995).
[5] E. Davalo et P. Naim, Des réseaux de neurones , Eyrolles, 1993.
[6] N.G. Pavlidis, D.K. TASOULIS and M.N. Vrahatis, Financial Forecasting Through Unsupervised Clustering and Evolutionary Trained Neural Networks ,2003 Congress on Evolutionary Computation, Canberra Australia, 2003.
[7] H. Demuth and M. Beale, Neural network toolbox for use with Matlab, *Users Guides*, *Mathworks*, 1997.
[8] http://www.it.lut.fi/project/nngenetic/