International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 02 Issue: 09 | Dec-2015 www.irjet.net p-ISSN: 2395-0072

# Design of State of Charge of Battery Estimator Using Locally Linear Model Tree for Use in Hybrid Vehicles

**Amin Rezaei** [1]

[1] Amin Rezaei, Electrical Engineering Department, University of Bonab, East Azerbayjan, Iran

---------------------------------------------------------------------***---------------------------------------------------------------------

**Abstract -** ***State of charge (SOC) estimation is one of the essential segments of hybrid electrical vehicles (HEV). By monitoring of SOC, we can optimize the consumption of fuel and decrease the pollution of air, in HEV. In this works considers the state of charge (SOC) estimation problem for lead-acid batteries for use in HEV. In this paper, the online state of charge estimation is worked using a locally linear model tree (LOLIMOT) which is a Nero-fuzzy network. The training data of LOLIMOT contain measured voltage, current and SOC data in different temperature in which voltage, current and temperature are used as inputs and SOC is output. In this paper, the SOC estimation results using LOLIMOT is compared with the results of ANFIS [1].***

***Key Words:*** *SOC, LOLIMOT, HEV*

## 1. INTRODUCTION

In HEV, a key parameter is the state of charge of the battery that is a measure of the amount of electrical energy stored in it. It is analogous to a fuel gauge on a conventional internal combustion (IC) car. To define the state of charge, consider a completely discharged battery. With $I_b(t)$ the charging current, the amount of charge delivered to the battery is $\int_{t_0}^{t} I_b(\tau)d\tau$. Thus, the total of charge that the battery can hold is $Q_0 = \int_{t_0}^{\infty} I_b(\tau)d\tau$ and the state of charge (SOC) of the battery is given by

$$SOC(t) = \frac{\int_{t_0}^{t} I_b(\tau)d\tau}{Q_0} \times 100 \qquad (1)$$

Typically, it is desired that the state of charge of the battery be kept within appropriate limits, for example 20% < SOC (%) < 95%. As a consequence, it is essential to be able to estimate the state of charge of the battery to maintain the state of charge within safe limits. Estimating the battery state of charge (SOC) is not an easy task because the SOC depends on many factors such as temperature, battery capacitance and internal resistance.

There are many different techniques of battery SOC estimation, which are summarized in the recent paper [3], and each of them has its most suitable field due to complex nonlinear behavior of battery. The reference [3] points out that at present, the most widely used technique for all systems is Ah counting because it can be easily understood, implemented and it gives satisfyingly accurate results. But actually a starting point SOC is required for the method. Due to complex nonlinear behavior of battery, it is quite difficult for most existed mathematical model to obtain satisfactory prediction accuracy. Therefore traditional methods of battery management systems are incapable for SOC estimation of HEV Batteries.

Artificial neural network (ANN) which is able to predict SOC online if trained before use, can be implemented for any battery and battery system, and it is a new promising method. Recently, soft computing techniques in the form of adaptive neuro-fuzzy inference system known as ANFIS begin to be utilized to estimate SOC under different operation condition such as constant current discharging (CCD) and random current discharging (RCD) [1].

In this paper is used the new neuro-fuzzy network (LOLIMOT) for SOC estimation of batteries in HEV. The schematic of LOLIMOT SOC estimator is pictured in Fig -1 where V, I, T, is as inputs of LOLIMOT estimator and SOC is as output of LOLIMOT estimator. For implementation of this estimator, we can sense V (terminal voltage of battery), I (load current) and T (combination of environment and battery temperature) by sensors when HEV is working.

## 2. STRUCTURE OF LOLIMOT

In this paper, a new algorithm (LOLIMOT) for nonlinear dynamic system identification (battery system) with local linear models is used [7]. The local linear model tree

(LOLIMOT) is based on the idea to approximate a nonlinear function with piece-wise linear models.

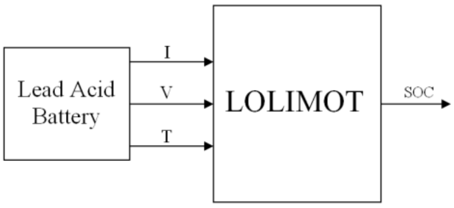


**Fig -1**: The schematic of LOLIMOT SOC estimator

The algorithm has an outer loop (upper level) that determines the input partitions (structure) where the local linear models are valid and an inner loop (lower level) that estimates the parameters of those local linear models. However, the partitions where the linear models are valid are not crisp but fuzzy, i.e. the local linear models are interpolated by weighting functions. In this paper normalized Gaussian weighting functions are applied. The LOLIMOT output y is calculated by summing up the contributions of all M linear models (hyper-planes)

$$y=\sum_{i=1}^{M}(w_{0i}+w_{1i}x_1+\ldots+w_{ni}x_n).\Phi_i(\underline{x},\underline{c}_i,\underline{\sigma}_i) \qquad (2)$$

where $w_{ij}$ are the parameters of the i[th] linear regression model, & $\underline{x}$ is the input vector and $\Phi_i$ is the normalized Gaussian weighting function for the ith model with centre coordinates $\underline{c}_i$ and standard deviations $\underline{\sigma}_i$ .

$$\Phi_i(\underline{x},\underline{c}_i,\underline{\sigma}_i)=\frac{z_i}{\sum_{j=1}^{M} z_j} \qquad (3)$$

$$z_j=\exp\left(-\frac{1}{2}\left(\frac{(x_1-c_{1j})^2}{\sigma_{1j}^2}+\frac{(x_2-c_{2j})^2}{\sigma_{2j}^2}+\ldots+\frac{(x_n-c_{nj})^2}{\sigma_{nj}^2}\right)\right)$$

Equation (2) can be interpreted as a Sugeno-Takagi fuzzy system with Gaussian membership functions and product operator as t-norm. It is also equivalent to the local model network based on a radial basis function network in and therefore an extension to radial basis function networks.

Assume the weighting functions would have been already determined. Then the parameters of each linear model are estimated separately (local estimation) by a weighted least squares technique. With the data matrix X (each row represents one measurement of $x^T$ at time instant k, i.e. $x_k^T$. The diagonal weighting matrix $Q_i$ (each entry $q_k$ is the weighting function value of the corresponding data $x_k^T$ ) and desired outputs y the optimal parameters % of the i[th] model are

$$\underline{w}_i=\left(X^TQ_iX\right)^{-1}X^TQ_i\underline{y} \qquad (4)$$

The overlapping of the weighting functions is ignored in this local estimation approach. This may lead to interpolation errors that grow with increasing standard deviations of the weighting functions. On the other hand local estimation is very fast and robust. Instead of optimizing all M(n+l) parameters in (2) globally, only a (n+l) parameter estimation is performed M times. Since an LS estimation has cubic complexity (e.g. matrix inversion with singular value decomposition) the global estimation approach is of O(M3) while the local estimation is of O(M). Furthermore, it is shown in that local estimation is more robust in the case of small noisy data sets, since it forces the linear models to represent the local surface of the unknown function and therefore avoids compensation effects like the "balancing of weights".

For optimization of each model all data are taken into account. The data points are weighted with the corresponding weighting function value. This is consistent with the fuzzy logic point of view, where the weighting function value is the degree of rule fulfillment that the corresponding model is true. This means that the closer the points are to the weighting function's centre the more significant they are for the estimation of the hyper-plane. Therefore each weighting function centre can be interpreted as an operating point for the corresponding linear model.

The following algorithm exploits ideas from other tree construction algorithms like CART, basis function trees and MARS to determine the centers and standard deviations of the weighting functions. The LOLIMOT algorithm partitions the input space in hyper-rectangles. Each local linear model belongs to one hyper-rectangle in which centre the weighting function is placed. The standard deviations are set proportional to the size of the hyper rectangle. This makes the size of the validity region of a local linear model proportional to its hyper-rectangle extension. A model may be valid over a wide operating range of one input variable but only in a small area of another one. Fig -2 illustrates position and orientation of the non-normalized Gaussians, where the ellipses characterize the contours and the crosses mark the centers.

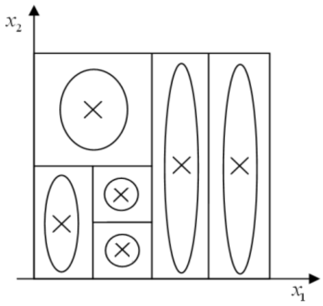

**Fig -2**: Possible partitioning of the input space with contour of the non-normalized weighting functions

The algorithm is the following:

1. Set the first hyper-rectangle in such a way that is contains all data points. Estimate a global linear model $y = w_0 + w_1 x_1 + \ldots + w_n x_n$.
2. For all input dimensions j := l...n:
   2a. cut the hyper-rectangle into two halves along dimension j.
   2b. Estimate local linear models for each half.
   2c. calculate the global approximation error (output error) for the model with this cut.
3. Determine which cut has led to the smallest approximation error.
4. Perform this cut. Place a weighting function within each centre of both hyper-rectangles. Set standard deviations of both weighting functions proportional to the extension of the hyper-rectangle in each dimension. Apply the corresponding estimated local linear models (from 2b).
5. Calculate the local error measures ***J*** on basis of a parallel running model for each hyper-rectangle.
6. Choose the hyper-rectangle with the largest local error measure ***J***.
7. If the global approximation error on a parallel model (output error) is too large go to step 2.
8. Convergence. Stop.

Fig -3a shows six iterations of the LOLIMOT algorithm for a two dimensional input space (n=2). It can be represented in a tree structure, see Fig -3b.

At each iteration, the worst local linear model is subdivided into two new ones. The cuts in all dimensions are tested and the one with the highest performance improvement is chosen. Note that the evaluation of the model's performance involves generalization since it is run with a parallel model while parameter tuning applies a series-parallel model. In order to control the complexity of the model and generate more parsimonious models a pruning step can be included between step 4 and 5. That pruning step 4 can merge two models if the loss of accuracy due to this merging is smaller than the gain of accuracy from the previous cut. Another very important extension of this algorithm is to allow different input spaces for the weighting functions in (4) and the local linear models in (2). The input space of the weighting functions should contain only those variables that may influence the nonlinear process behavior, while the input space of the local linear models should include all variables on which the process dynamics may depend on. By choosing the variables of these input spaces prior knowledge can be incorporated to speed up the training.

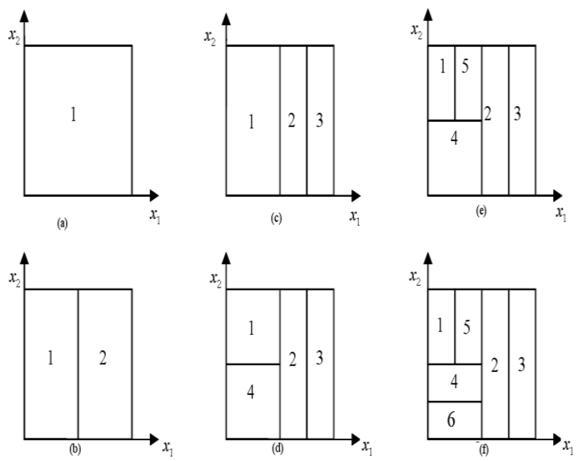

**Fig -3a:** Six iterations of LOLIMOT

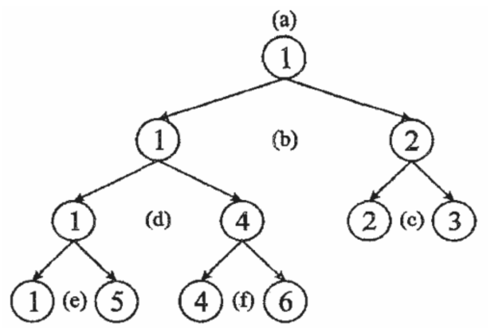

**Fig -3b:** Tree structure for Fig -3a

## 3. LOLIMOT SOC ESTIMATOR AND ITS RESULTS

LOLIMOT network is trained with experimental data (voltage, current, temperature and SOC) of lead acid battery that used in HEV. For training of this LOLIMOT network, first we specified the optimal number of neurons with using of computing of square error summation for different number of neurons (Fig -3). According to Fig -3, the number of the optimal neuron obtained 14 neurons. Then we trained network for experimental data by using LOLIMOT algorithm. Results of this training are shown in Fig4. For these experimental data, the training error with the optimal number of neuron is 3.0218 % (Fig -4).

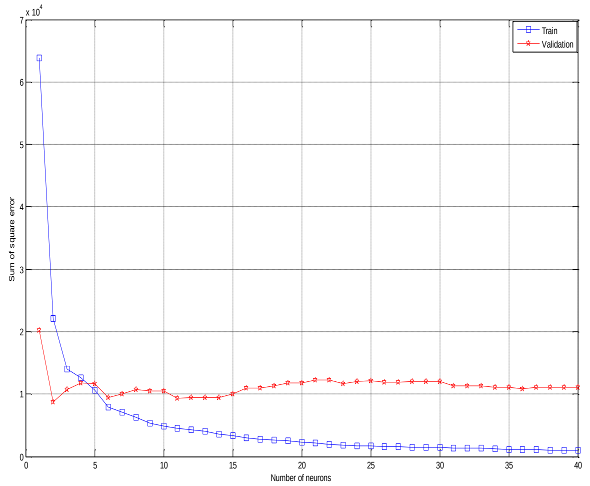


**Fig -4:**The square error summation according the Number of neurons

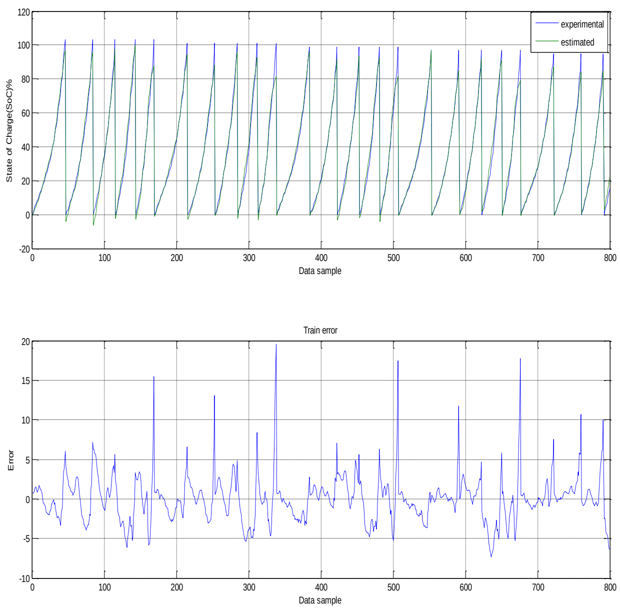


**Fig -5:** Estmated SOC and training error for LOLIMOT SOC estimator

For testing of this trained LOLIMOT, we used 214 data that the error is 7.4823%. the error of test data is bigger than the error of training data because the number of training data is bigger than the number of test data. Results of testing of LOLIMOT network are shown Fig -5. In Fig -7 actual values and estimated values for test data is shown.

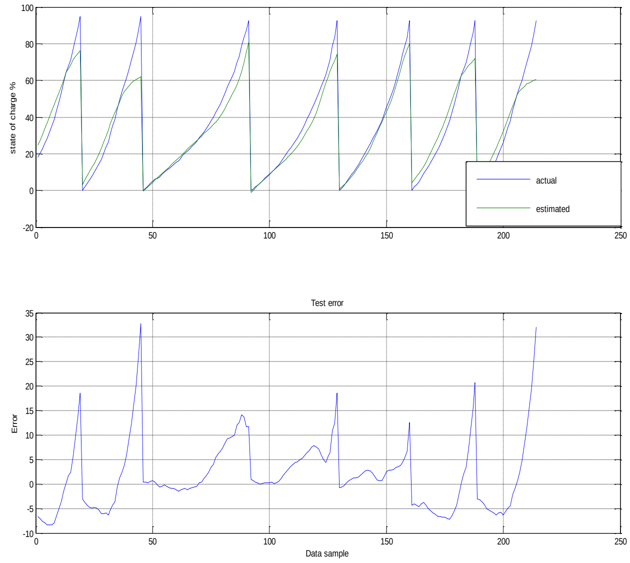


**Fig -6:** Error and output of LOLIMOT network for test data

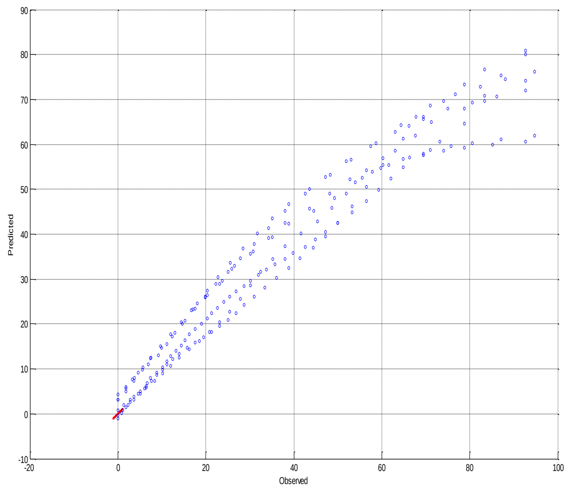


**Fig -7:** Actual values and estimated values for test data

## 4. COMPARISON OF LOLIMOT AND ANFIS SOC ESTIMATORS

For the same experimental data, we use from ANFIS GUI in MATLAB software for estimation of SOC [1]. Used ANFIS network in this paper, contains three member functions for each the inputs therefore the ANFIS network contains 27 rules that it is rather than the neurons of LOLOMOT network (14 neurons). Error of ANFIS network training is shown in Fig8.Training error for ANFIS network is 4.6829% for the same data. Training of ANFIS estimator for the same data is shown in Fig -9 and the results of test data for ANFIS network is shown in Fig -10. The error of ANFIS network for test data is 15.7438% that is bigger than the error of test data in LOLIMOT network. Therefore the LOLIMOT network is better than ANFIS network for SOC estimation of lead acid battery in HEV.

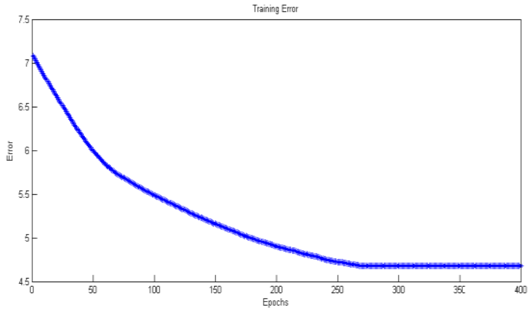


**Fig -8:** Error of ANFIS network training

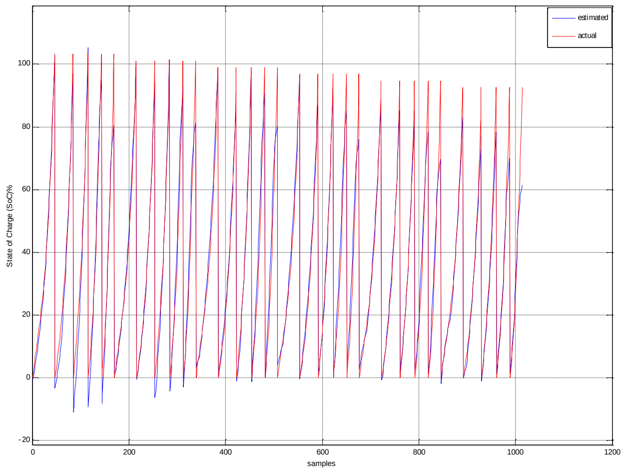


**Fig -9:** Training of ANFIS estimator for the same data

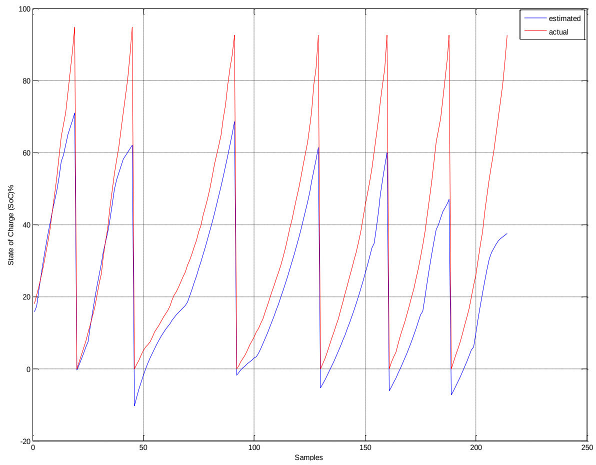


**Fig -10:** Results of test data for ANFIS network

## 5. CONCLUSION

In this paper, a new online soc estimator for lead acid batteries is designed and simulated. For design of this estimator, LOLIMOT is used. Results of the LOLIMOT soc estimator compared with ANFIS soc estimator and according to error and number of neuron in the LOLIMOT soc estimator, we find out this estimator is better than ANFIS estimator.

## REFERENCES

[1] C.H. Cai, D. Du, Z.Y. L iu, *Battery State-of-Charge Estimation Using Adaptive Neuro-Fuzzy Inference System (ANFIS)*, 2003 IEEE conference.

[2] W. X. Shen, C. C. Chan, E. W. C. Lo,, and K. T. Chau, *Adaptive Neuro-Fuzzy Modeling of Battery Residual Capacity for Electric Vehicles*, IEEE Transactions On Industrial Electronics, Vol. 49, No. 3, June 2002

[3] S. Piller, Marion Perrin, A. Jossen, *Methods for state-of-charge determination and their application*, Journal of Power Sources, 96(2001) 113-120.

[4] J.S. Roger Jang. *ANFIS: Adaptive-Network-based Fuzzy Inference Systems*. IEEE Trans. on Systems, Man, and Cybernetics, 23(03): 665-685, May 1993.

[5] Pascoe Phillip E., Anhuky, Adnan H. *Adaptive fuzzy coup de fouet based VRLA battery capacity estimation*. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Vol. 4, pp. 2157-2162, Oct. 2001.

[6] C.H. Cai, D. Du, et al. *Artificial neural network in estimation of battery state-of-charge (SOC) with nonconventional input variables selected by correlation analysis*. The First IEEE International Conference on Machine Learning and Cybernetics 2002 (ICMLC'OZ),Beijing, pp. 1619-1625, Nov. 4-5 2002.

[7] Oliver Nelles and Rolf Isermann, *Basis Function Networks for Interpolation of Local Linear Modells*, Proceedings of the 35th WA18 10:40 Conference on Decision and Control Kobe, Japan * December 1996

[8] Murray-Smith R., *A Local Model Network Approach to Nonlinear Modelling*, Ph.D. Thesis, University of Strathclyde, UK, 1994

[9] Jinchun Peng, Yaobin Chen, R. Eberhart. *Battery Pack state of Charge Estimator Design Using Computational Intelligence Approaches*,15' AMud Battery Conference on Applications and Advances (CAT. No. 00TH8490), IEEE, 1999, pp. 173-7. Piscataway, NJ, USA.

[10] Rezaei.P.R A., Salmasi FR. *State of charge estimation for batteries in HEV using locally linear model tree (LOLIMOT)*. International Conference on Electrical Machines and Systems; Seoul. 2007 Oct 8-11. p. 2041–5.