Nonlinear Filtering Techniques Comparison for Battery State Estimation

Aspasia Papazoglou, Stefano Longo[*]*, Daniel Auger, Francis Assadian*
Centre for Automotive Engineering, Cranfield University, Bedfordshire, UK
e-mail: s.longo@cranfield.ac.uk

Cite as: Papazoglou, A., Longo, S., Auger, D., Assadian, F., Nonlinear Filtering Techniques Comparison for Battery State Estimation, J. sustain. dev. energy water environ. syst., 2(3), pp 259-269, 2014, DOI: http://dx.doi.org/10.13044/j.sdewes.2014.02.0021

ABSTRACT

The performance of estimation algorithms is vital for the correct functioning of batteries in electric vehicles, as poor estimates will inevitably jeopardize the operations that rely on un-measurable quantities, such as State of Charge and State of Health. This paper compares the performance of three nonlinear estimation algorithms: the Extended Kalman Filter, the Unscented Kalman Filter and the Particle Filter, where a lithium-ion cell model is considered. The effectiveness of these algorithms is measured by their ability to produce accurate estimates against their computational complexity in terms of number of operations and execution time required. The trade-offs between estimators' performance and their computational complexity are analyzed.

KEYWORDS

Battery-management system, Estimation algorithms, Lithium-ion cells, State of Health, Computational complexity.

INTRODUCTION

Accurate battery estimation algorithms are considered to be of great importance due to their applications in electrified transportation and energy storage systems. Battery systems constitute the core source of energy for electrified vehicles hence their safe and reliable operation has to be ensured. Their degradation over time and the limited amount of information related to their internal state justifies the need of controlling batteries by advanced estimation algorithms.

Battery-management systems (BMSs) are embedded computers that have the ability to execute real time algorithms in order to estimate, control and communicate with other components. Implemented into Electric Vehicles (EVs), BMSs manage the battery by monitoring its state, protecting the battery and controlling its environment [1]. Estimating battery's state provides information that is essential for obtaining critical variables such as the State of Health (SoH) and State of Charge (SoC) of the battery. The estimated state of the battery determines the control actions that will be taken by the BMS and enables a more aggressive utilization of the battery, while ensuring its safe and reliable operation [2].

Over the past decades, different versions of battery models have been applied in combination with a number of estimation algorithms. The first category includes equivalent circuit models (ECMs), which consider the battery as if it was an electrical network. The majority of battery models are based on ECMs as they represent complex relationships in a simplified way, aiding the calculations and the analysis. Methods for

[*] Corresponding author

estimating states and parameters based on ECMs by using the Extended Kalman Filter (EKF) [3] and the Sigma-point (or Unscented) Kalman Filter (UKF) have been proposed [4]. An adaptive unscented Kalman filter based on an ECM to develop an online estimation of the SoC of EV's battery has also been proposed [2]. Other approaches include a nonlinear ECM to estimate the SoC of the battery model by applying the Particle Filter (PF) [5] and the use of ECMs to compare the effectiveness of EKF and PF based on state estimations [6].

The second category consists of the electrochemical models, battery models that are represented by partial differential equations and provide an explanation of the fundamental physics of the batteries. Electrochemical principles to model a Lithium-ion (Li-ion) battery with a reduced-order model are used in [7]. In [8], the same authors, based on the analytical expressions of a Li-ion battery, present a simpler electrochemical model, which is used to design an observer. The performance and effectiveness of the PF to specific algorithms in an electrochemical model is presented in [9]. Such modeling method is complicated, as it is difficult to obtain model's parameters and it is not computationally suitable for online estimation.

This paper considers Li-ion batteries, which are recognised as the most promising technology [8]. Except from belonging among the batteries with the best energy-to-weight ratios, they have the privilege of lacking memory effect and when they are not in use their self-discharge rate is low. Additionally, their decreasing cost enlists them as a leading candidate for the next automotive generation [7]. The battery model that is chosen to represent the Li-ion battery belongs to the ECMs category and is known as Thevenin model. The estimation algorithms that have been compared here to estimate the states of the Li-ion battery are the EKF, UKF and PF. The algorithm's accuracy is tightly coupled with its computational complexity. Unfortunately, the complexity that characterizes each algorithm increases with its accuracy, making the choice of the estimation technique more challenging.

The aim of this paper is to analyze the complexity of the EKF, UKF and PF based on their number of operations and the execution time (i.e. their computational effort). The trade-off between estimators' performance and computational complexity is shown. This could be used for the selection of the most appropriate estimation technique for a particular application.

LITHIUM-ION CELL MODEL

ECMs have been widely used in different type of modeling and simulation for EVs and BMSs. They are based on the dynamic characteristics and working principles of batteries and they use capacitors, resistances and voltage sources to compose circuit networks. The battery cell model that is used in this paper is a discrete-time, nonlinear model, based on the Thevenin circuit. The Thevenin model consists of a parallel RC network connected in series, which includes a polarization resistance R_{Th} and an equivalent capacitance C_{Th}. Symbol R_{o} represents an ohmic resistance and U_{L} is the terminal voltage. I_{L} is the current (positive for charging, negative for discharging) and U_{oc} is an ideal voltage source, chosen to represent an open-circuit voltage [10]. The schematic diagram of the circuit is illustrated in Figure 1.

The cell model can be presented by the nonlinear set of state space equations:

$$x_{\mathrm{k}} = f(x_{k-1}, u_{k-1}, w_{k-1}, k-1) \tag{1}$$

$$y_{\mathrm{k}} = h(x_{\mathrm{k}}, u_{\mathrm{k}}, v_{\mathrm{k}}, k) \tag{2}$$

where, in our case, x_k = [SoH SoC U_{Th} Temperature]^T is the state vector, $y_k = U_L$ is the output of the system, $u_k = I_L$ is the current input to the system, w_k is the process noise (zero-mean Gaussian), v_k is the measurement noise (zero-mean Gaussian). The nonlinear function *f* is given by the relationship:

$$SoH_k = SoH_{k-1} - \frac{|I_{L_k}|}{2L_T 3600 Q_{cell}} \tag{3}$$

$$SoC_k = SoC_{k-1} - \frac{I_{L_k}}{3600 Q_{cell}} \tag{4}$$

$$U_{Th_k} = -\frac{U_{Th_{k-1}}}{R_{Th} C_{Th}} + \frac{I_{L_{k-1}}}{C_{Th}} \tag{5}$$

The temperature *T* is assumed, at this stage, to be constant. Although the estimation algorithms used in the paper focus on state estimation, temperature is in this case a parameter and it is included in the state vector so that an estimate can be readily obtained. Symbols L_T, Q_{cell} and $C_{Th,}$ which represent the lifetime, the capacity and the capacitance of the cell, respectively, are constant values. Further details regarding equation (3) are included in [16]. The nonlinear measurement function *h* is given by:

$$U_{L_k} = -U_{Th_k} - I_{L_k} R_o + U_{oc} \tag{6}$$

Resistances R_o, R_{Th} and the Open Circuit Voltage (OCV) are considered to be parameters and are functions of the SoH, SoC, current, temperature and charge/discharge conditions. They are obtained from experimental results and the procedure to determine them is based on offline methods and explained in [15].

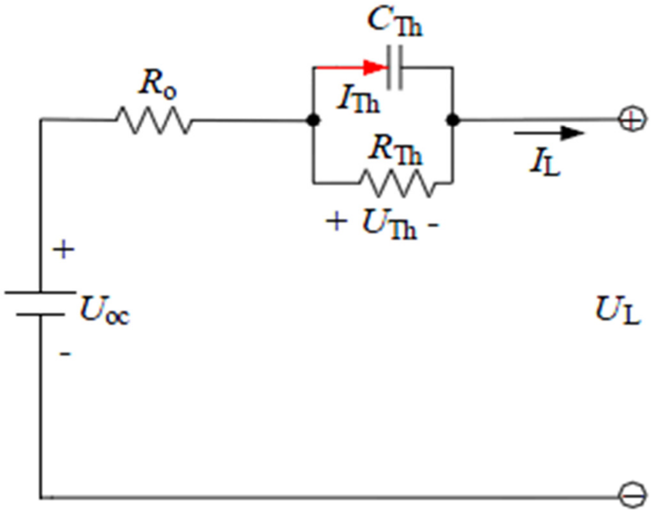

Figure 1. Thevenin model

ESTIMATION ALGORITHMS

Estimation algorithms are mathematical techniques used to compute the optimal estimates of states and parameters of a dynamical system [11]. In this paper, they are used to estimate the states of the Li-ion battery cell, which cannot be directly measured. The EKF, UKF and PF are considered here. The algorithms are presented in sequential stages. At each stage, the number of operations (additions, subtractions, multiplications, divisions and square roots) required is determined. This is given as a function of the number of states *n*, the number of inputs *m* and the number of outputs *p*.

Extended Kalman Filter

In the EKF, the nonlinear dynamical system is linearized at every time step by a first-order Taylor-series expansion approximation. The linearized system is then used to compute the estimation of the states and the error covariance matrices [11, 12]. A summary of the algorithm is presented in Table 1. (Notice that in the table, the superscript '-' indicates an estimate at time *k* without the information about the prediction at time *k*. The superscript '+' indicates an estimate at time *k* given the information about the prediction at *k*.)

The EKF is easy to implement in terms of complexity, especially when the linearized system matrices (Jacobians) can be computed analytically. Its simplicity makes it a tempting choice among the variety of nonlinear estimation algorithms. However, the EKF's performance deteriorates when models are highly nonlinear. The UKF, on the other hand, can better deal with harsh nonlinearities.

Table 1. Summary of the Extended Kalman Filter

Stages	Computations required	Number of FLOPs
Initialization		
For $k = 0$ set Initial state estimation	$\hat{x}_o^{+}$	-
Initial error covariance matrix	P_o^{+}	-
For $k = 1, 2, \ldots$ compute: Linearized matrices $\hat{\mathrm{A}}_k, \hat{B}_k, \hat{C}_k, \hat{D}_k$	$\hat{\mathrm{A}}_k = \left.\frac{\partial f(x_k, u_k, w_k, k)}{\partial x_k}\right\rvert_{x_k=\hat{x}_k^{+}}$	Derived analytically
	$\hat{B}_k = \left.\frac{\partial f(x_k, u_k, w_k, k)}{\partial w_k}\right\rvert_{w_k=\tilde{w}_k}$	
	$\hat{C}_k = \left.\frac{\partial h(x_k, u_k, w_k, k)}{\partial x_k}\right\rvert_{x_k=\hat{x}_k^{-}}$	
	$\hat{D}_k = \left.\frac{\partial h(x_k, u_k, w_k, k)}{\partial v_k}\right\rvert_{v_k=\tilde{v}_k}$	
Main loop		
State estimate time update	$\hat{x}_k^{-} = f(\hat{x}_{k-1}^{+}, u_{k-1}, \tilde{w}_{k-1}, k-1)$	$40n^2 + 1740n$
Error covariance time update	$P_k^{-} = A_{k-1} + \Sigma_{\tilde{x},k-1}^{+} A_{k-1}^{T} + \Sigma_w$	$2n(n^2+n+1)+98+$ $mn(4m+4n+3)$
Output estimate	$\hat{y}_k = C_k \hat{x}_k^{-} + D_k u_k$	$492n$
Estimator gain matrix	$K_k = \Sigma_{\tilde{x},k}^{-} C_k^T \left[C_k \Sigma_{\tilde{x},k}^{-} C_k^T + \Sigma_v\right]^{-1}$	$pn(2n+3p-1)+120+$ $p(n-1)(2n+p)+p$
State estimate measurement update	$\hat{x}_k^{+} = \hat{x}_k^{-} + L_k(y_k - \hat{y}_k)$	$p(2n-1)$
Error covariance measurement update	$P_k^{+} = (I - L_k C_k)\Sigma_{\tilde{x},k}^{-}$	$n^3 + 2n^2$

Unscented Kalman Filter

The filters that belong to the Kalman Filter category attempt to propagate the mean and the covariance values of a system using the methods of time update and measurements update [13]. According to this, when systems under consideration are highly nonlinear, the method of linearization based on the EKF does not seem to be the ideal solution. The UKF determines a set of sigma points and transforms them nonlinearly to a new set of points. Due to this, the mean and covariance value of the sigma points matches the mean and covariance value of the estimated value. The states of the system and the covariance matrices are computed based on those sigma points [14]. A summary of the UKF algorithm is presented in Table 2.

Table 2. Summary of the Unscented Kalman Filter

Stages	Computations required	Number of FLOPs
Initialization		
For $k = 0$ set		
Initial state estimation	$\hat{x}_o^+$	-
Initial error covariance matrix	P_o^+	-
Main loop		
For $k = 1,2,\ldots$ compute:		
State estimate time update	$x_{k-1}^{(i)} = \hat{x}_{k-1}^{+} + \tilde{x}^{(i)}$ $x_k^{(i)} = f(x_{k-1}^{(i)}, u_{k-1}, w_{k-1}, k-1)$ $\hat{x}_k^{-} = \frac{1}{2n}\sum_{i=1}^{2n} x_k^{(i)}$	$\frac{n^3}{3} + 3n^2$ $40n^2 + 1740n$ $2n^2$
Error covariance time update	$P_k^- = \frac{1}{2n}\sum_{i=1}^{2n}\left(x_k^{(i)} - \hat{x}_k^-\right)\left(x_k^{(i)} - \hat{x}_k^-\right)^T + Q_k$	$4n^3+6n^2+98+2$ $(m+n+1)(mn+n)$
Output estimate	$y_k^{(i)} = h(x_k^{(i)}, u_k, v_k, k)$ $\hat{y}_k = \frac{1}{2n}\sum_{i=1}^{2n} y_k^{(i)}$	$492n$ $2np$
Estimator gain matrix	$P_y = \frac{1}{2n}\sum_{i=1}^{2n}\left(y_k^{(i)} - \hat{y}_k^-\right)\left(y_k^{(i)} - \hat{y}_k^-\right)^T + R_k$ $P_{xy} = \frac{1}{2n}\sum_{i=1}^{2n}\left(x_k^{(i)} - \hat{x}_k^-\right)\left(y_k^{(i)} - \hat{y}_k^-\right)^T$ $K_k = P_{xy} P_y^{-1}$	$4n(p^2+p)+p^2$ $np(4n+3)+$ $2n^2+n+p$ $2p^2n-pn$
State estimate measurement update	$\hat{x}_k^+ = \hat{x}_k^- + K_k(y_k - \hat{y}_k)$	$2pn-n+p$
Error covariance measurement update	$P_k^+ = P_k^- - K_k P_y K_k^T$	$pn(2p+2n-1)$

Table 3. Summary of the Particle Filter

Stages	Computations required	Number of FLOPs
Initialization		
For k = 0 and I = 1,..,N Generation of N initial randomly particles	$x_{o,i}^{\;+}$	$3Nn$
Main loop		
For k = 1, 2,… compute: *A priori N* particles	$x_{k,i}^{-}= f_{k-1}(x_{k-1,i}^{+},u_{k-1},w_{k-1}^{i})$	$N(43n^2+1739n)$
Importance weights	$q_{k,i}=p(y_k \vert x_{k,i}^{\;-})$	$N(2p^2+p+498n)$
Normalization of the weights	$q_{k,i}=\frac{q_{k,i}}{\sum_{j=1}^{N} q_{k,j}}$	Np
A posterior N particles	$x_{k,i}^{+}$ different resampling techniques used	$2N-1$
Statistical measurements	mean $(x_{k,i}^{+})$ or covariance $(x_{k,i}^{+})$	$n(N+1)$

Particle Filter

Both the EKF and UKF work under the assumption of a Gaussian posterior, limiting the range of their applications and characterizing them as inappropriate for non-Gaussian dynamical systems [5, 17]. The PF algorithm (presented in Table 3) uses on a different philosophy. Being part of the sequential Monte Carlo methods, the PF estimate states and parameters of a nonlinear dynamical system using a number of sample points based on Bayesian estimators [9].

ESTIMATION ALGORITHMS COMPARISON

The description of the estimation algorithms is followed by a comparison of the results of the estimated states. At this point, it should be noticed that the cell model used to simulate the 'real' battery cell is also used within the estimation algorithms to compute the estimated values. This provides an advantage in the entire estimation procedure, as the estimators operate under ideal modeling conditions. However, in the model for the 'real' battery, a Gaussian noise with variance equal to 4 is injected to the input and output signal.

The analysis of the algorithms' performance focuses on the estimated values of SoH, SoC, temperature and terminal voltage (see Figure 2). The sampling time was selected to be 1s, which is appropriate given the relatively slow dynamic response of the battery. Each graph contains the estimated values from the three filters (the PF is implemented for 100 particles).

As it can be seen in the plots, the SoC and voltage estimations seem to have a faster convergence rate than SoH and temperature, which also converge but at a slower rate. The reason may be that slowly changing states (as the SoH) and parameters (as the temperature in this case) are generally more difficult to track. Overall, the EKF appears to have a better performance and be more accurate when compared to the other filters.

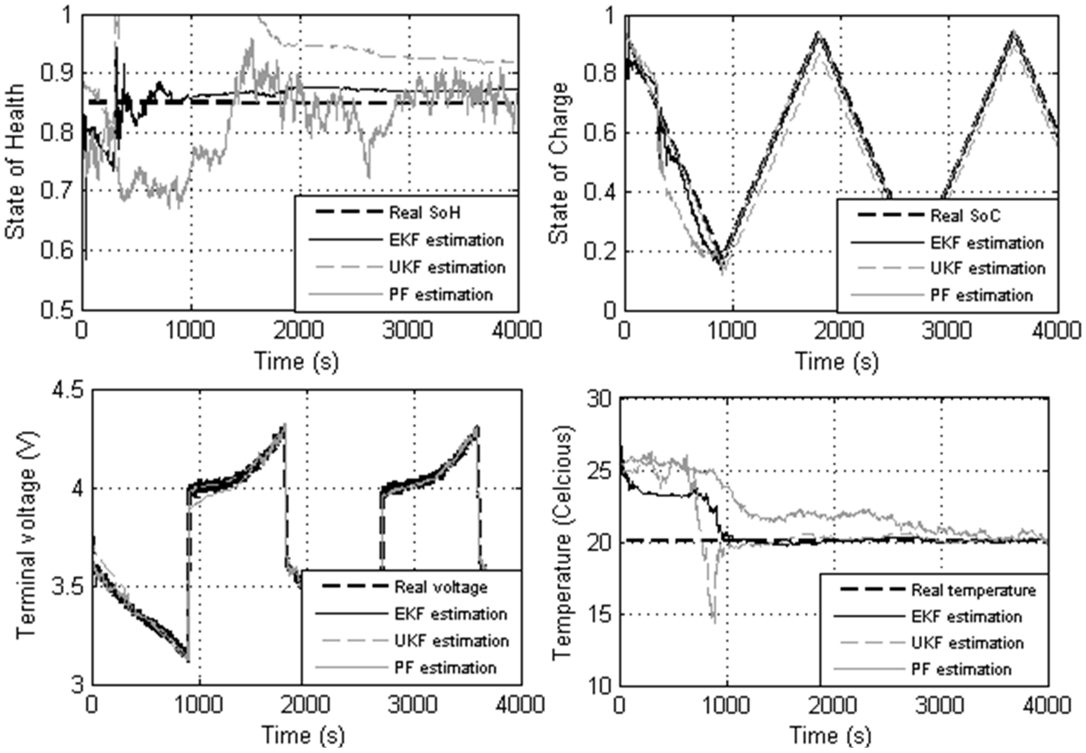

Figure 2. Estimation results

However, at steady state, the EKF and the UKF converge to the same estimate. The superiority of the EKF towards UKF has to do with the type of nonlinearities that characterizes the cell model and the fact that the noise affecting the system is precisely known. The model for the Li-ion cell is only moderately nonlinear. The UKF approximates the Gaussian noise distribution rather than the model. With a mildly nonlinear model and known noise distribution it is not surprising that the EKF, which approximates the model instead of the noise distribution, performs better than the UKF. The PF seems to have higher accuracy, as far as SoC and voltage are concerned. The reduced performance in SoH and temperature estimation may be due again to the noise distribution. PFs perform better for non-Gaussian systems but the noise added to this particular model is Gaussian. Further work will include experiments with models corrupted by non-Gaussian noise.

Figure 3. shows the estimation errors for the EKF, the UKF and the PF. The estimation error is defined as the difference between the 'real' value and the estimated one. The PF seems to have the lowest SoC and voltage estimation error followed by the EKF and the UKF. But overall the EKF seems to be more accurate for this particular case.

COMPUTATIONAL COMPLEXITY ANALYSIS

The performance of the estimation algorithms is significant for the safe and efficient operation of the battery, hence accuracy is one of the most important criteria upon which the choice of the algorithm is based. However, accuracy and computational complexity are correlated and both of them have to be taken into account when evaluating the algorithms' performance. Computational complexity will be analyzed in terms of the number of operations performed in the algorithm and the corresponding execution time.

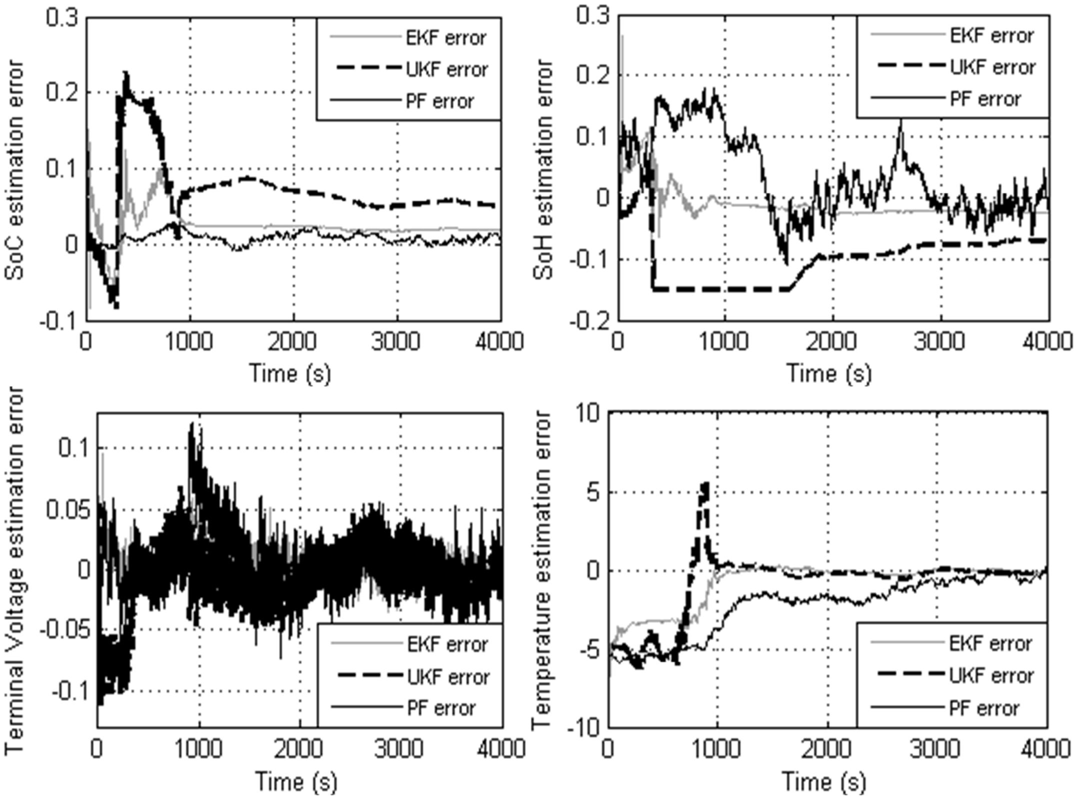

Figure 3. Estimation errors

Number of operations

In order to determine the number of operations executed in the algorithms, a new measure will be introduced, the number of FLoating-point OPerations (FLOPs) [15]. The FLOPs are indicative of the complexity of each algorithm and constitute a measure of comparison.

Taking into account that every embedded computer has a different range of capabilities and that for a typical processor more than one step is needed to execute one operation, it is assumed that all the operations have the same weight and each operation (addition, subtraction, multiplication, division and square root) corresponds to one FLOP. Tables 1, 2 and 3 also show the number of FLOPs needed for each algorithm's step. Keeping as a constant the number of inputs $m = 1$ and the number of outputs $p = 1$ and varying the number of states (for example by adding more capacitors in the Thevenin model for higher accuracy) the graph in Figure 4 is presented.

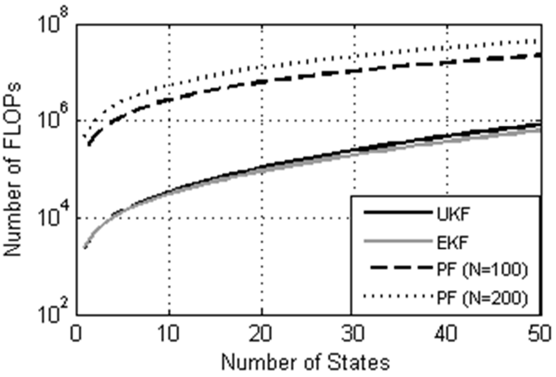

Figure 4. Number of FLOPs for the EKF, UKF and PF

It can be seen that as the number of states increases, the number of FLOPs increases rapidly, especially for the PF. For $n = 4$, which is the number of states of the proposed cell model, the total number of FLOPs for the EKF is 10,217, for the UKF is 10,414 and for the PF is 964,603 (N = 100) and 1,929,203 (N = 200). It should be noted that the complexity of the PF, even for N = 100, is approximately two orders of magnitudes higher than the one of the EKF, making it challenging to use the PF for real time estimation.

It would be useful to be able to combine the time needed to execute an algorithm with the number of FLOPs that this algorithm needs. There are a variety of processors with different capabilities and an example of a common processor will be used in order to understand how FLOPs relate to time. Assuming that a processor has a clock frequency of 80 MHz and that it executes one FLOP per clock cycle, then the processor has a theoretical performance of 80,000.000 FLOPs/second, and the actual computational time for an algorithm can be easily calculated.

Execution time

The analysis presented earlier (FLOP counting) is theoretical and only gives an indication of the algorithms' complexity. In practice, the actual execution time is needed, and this can often only be obtained experimentally. Hence, the actual execution time of the three filters was measured experimentally and presented in Table 4. The computation was performed on a desktop PC with a 2.1 GHz CPU, 4 GB of RAM and running MATLAB's version 7.12.0 (R2011a).

The execution time for the UKF and the execution time for the PF are 2.4 and 35.2 times longer than the one for the EKF, respectively. The time difference between the PF and the other two algorithms is very large, considering that these estimations should be executed online and within a typical sampling interval of 1s. This result confirms the high complexity of the PF's that was predicted, theoretically, by counting the number of FLOPs.

Table 4. Experimental computational time of EKF, UKF and PF

	EKF	UKF	PF
Time for one iteration (s)	0.0017	0.0041	0.0599
Proportion of execution time	1x	2.4x	35.2x

CONCLUSION

This paper compared the performance of three algorithms for state estimation of a nonlinear Li-ion battery cell model. The comparison was done in terms of the estimates accuracy and the computational complexity required. The nonlinear cell model was based on an equivalent Thevenin circuit. The states estimated were SoH, SoC, voltage and temperature. Results showed that the EKF, although based on an approximation of the nonlinear dynamics, was generally more accurate than the UKF and PF. This was probably due to the fact that the UKF, which is based on an approximation of the noise distribution rather than process dynamics, performs better for highly nonlinear systems and the cell model considered was only mildly nonlinear. The performance of the PF was unsatisfactory regarding the SoH and temperature estimations. This has to be attributed to the fact that the PF outperforms the other algorithms when the system is corrupted by non-Gaussian noise (while simulations were performed only with Gaussian noise). The computational complexity of the PF is up to two orders of magnitudes higher than the other two methods and its practical applicability would be difficult to justify for the model used here. In order to fully understand the capabilities and limitations of the algorithms considered, test should be done with complex nonlinear models corrupted by non-Gaussian noise. This will be subject of future research.

REFERENCES

1. Yinjiao, X., Kwok, M., Eden W., Pecht, M., Battery Management Systems in Electric and Hybrid Vehicles, *Energies,* pp 1840-1857, 2011, http://dx.doi.org/10.3390/en4111840
2. Fengchun, S., Yuan Z., Siguang, L., Adaptive Unscented Kalman Filtering for State of Charge Estimation of a Lithium-ion Battery for Electric Vehicles, *Energy,* Vol. 36*,* pp 3531-3540, 2011, http://dx.doi.org/10.1016/j.energy.2011.03.059
3. Plett, G., Sigma-point Kalman Filtering for Battery Management Systems of LiPB-based HEV Battery Packs Part 1: Introduction and State Estimation, *Journal of Power Sources,* Vol. 161, pp 1356-1368, 2006, http://dx.doi.org/10.1016/j.jpowsour.2006.06.003
4. Plett, G., Extended Kalman Filtering for Battery Management Systems of LiPB-based HEV Battery Packs Part 2: Modeling and Identification, *Journal of Power Sources,* Vol. 134, pp 262-276, 2004, http://dx.doi.org/10.1016/j.jpowsour.2004.02.032
5. Gao, M., Liu, Y., He, Z., Battery State of Charge online Estimation based on Particle Filter, *4th International Congress on Image and Signal Processing,* 2011, http://dx.doi.org/10.1109/CISP.2011.6100603
6. Restaino, R., Zamboni, W., Comparing Particle Filter and Extended Kalman Filter for Battery State-Of-Charge Estimation, *38th Annual Conference on IEEE Industrial Electronics Society,* 2012, http://dx.doi.org/10.1109/IECON.2012.6389247
7. Chaturvedi, N., Klein, R., Christensen, J., Ahmed, J., Kojic, A., Algorithms for Advanced Battery-Management Systems, *IEEE Control systems magazine,* pp 49-68, 2010, http://dx.doi.org/10.1109/MCS.2010.936293
8. Klein, R., Chaturvedi, N., Christensen, J., Ahmed, J., Findeisen, R., Kojic, A., Electrochemical Model Based Observer Design for a Lithium-Ion Battery, *IEEE Transactions on Control Systems Technology,* 2012, http://dx.doi.org/10.1109/TCST.2011.2178604
9. Samadi, M., Mahdi Alavi, A., Saif, M., An Electrochemical Model-based Particle Filter Approach for Lithium-ion Battery Estimation, *51st IEEE Conference on Decision and Control,* 2012, http://dx.doi.org/10.1109/CDC.2012.6426009
10. Hongwen, H., Xiong, R., Fan, J., Evaluation of Lithium-Ion Battery Equivalent Circuit Models, *Energies,* Vol. 4, pp 582-598, 2011., http://dx.doi.org/10.3390/en4040582
11. Chan, D. S. H.L., A New Battery Model for use with Battery Energy Storage Systems and Electric Vehicles Power Systems, *IEEE,* 2000, http://dx.doi.org/10.1109/PESW.2000.850009
12. Rapid STM32 Team, Developing a Tilt Sensor System Using Rapid STM32 Blockset, (Part 1 Design and Simulation), Rapid STM32 Team, [Online], Available: www.rapidstm32.com. [Accessed: 22-July-2013]
13. Simon, D., *Optimal State Estimation*, John Wiley & Sons, New Jersey, 2006, http://dx.doi.org/10.1002/0470045345
14. Wei, H., Williard, N., Chen, C., Pecht, M., State of Charge Estimation for Electric Vehicle Batteries using Unscented Kalamn Filter, *Microelectronics Reliability,* 2012, http://dx.doi.org/10.1016/j.microrel.2012.11.010
15. Antaloae, C., Marco, J., Assadian, F., A Novel Method for the Parameterization of a Li-Ion Cel Model for EV/HEV Control Applicatons, *IEEE Transactions on Vehicular Technology*, 2012, http://dx.doi.org/10.1109/TVT.2012.2212474

16. Auger, D., Groff, M., Mohan, G., Longo, S., Assadian, F., The impact of Battery Ageing on an EV Powertrain Optimisation, *8th Conference in Sustainable Development of Energy, Water and Environment Systems*, Croatia, 2013.
17. Schwunk, S., Particle Filter for State of Charge and State of Health estimation for Lithium-ion Phosphate Batteries, *Journal of Power Sources,* Vol. 239, pp 705-710, 2012, http://dx.doi.org/10.1016/j.jpowsour.2012.10.058

Paper submitted: 24.02.2014
Paper revised: 09.05.2014
Paper accepted: 15.05.2014