

# **Evaluarea stării de degradare a bateriilor reîncărcabile în sisteme alimentate cu baterii**

Teză destinată obținerii  
titlului științific de doctor inginer  
la  
Universitatea Politehnica Timișoara  
în domeniul CALCULATOARE ȘI TEHNOLOGIA  
INFORMAȚIEI  
de către

**Ing. Lucian UNGUREAN**

Președintele comisiei:	prof.univ.dr.ing Mircea Stratulat
Conducător științific:	prof.univ.dr.habil. ing. Mihai V. Micea
Referenți științifici:	prof.univ.dr. Marius Marcu ș.l. dr.ing. Sebastian Fuicu

Ziua susținerii tezei: .....

Seriile Teze de doctorat ale UPT sunt:

- |   |   |
|---|---|
| 1. Automatică                               | 11. Știința și Ingineria Materialelor                                   |
| 2. Chimie                                   | 12. Ingineria Sistemelor  |
| 3. Energetică                               | 13. Inginerie Energetică  |
| 4. Inginerie Chimică                        | 14. Calculatoare și Tehnologia Informației                              |
| 5. Inginerie Civilă                         | 15. Ingineria Materialelor  |
| 6. Inginerie Electrică                      | 16. Inginerie și Management   |
| 7. Inginerie Electronică și Telecomunicații | 17. Arhitectură   |
| 8. Inginerie Industrială                    | 18. Inginerie Civilă și Instalații                                      |
| 9. Inginerie mecanică                       | 19. Inginerie Electronică, Telecomunicații și Tehnologii Informaționale |
| 10. Știința Calculatoarelor                 |   |

Universitatea Politehnica Timișoara a inițiat seriile de mai sus în scopul diseminării expertizei, cunoștințelor și rezultatelor cercetărilor întreprinse în cadrul Școlii doctorale a universității. Seriile conțin, potrivit H.B.Ex.S Nr. 14 / 14.07.2006, tezele de doctorat susținute în universitate începând cu 1 octombrie 2006.

Copyright © Editura Politehnica – Timișoara, 2021

Această publicație este supusă prevederilor legii dreptului de autor. Multiplicarea acestei publicații, în mod integral sau în parte, traducerea, tipărirea, reutilizarea ilustrațiilor, expunerea, radiodifuzarea, reproducerea pe microfilme sau în orice altă formă este permisă numai cu respectarea prevederilor Legii române a dreptului de autor în vigoare și permisiunea pentru utilizare obținută în scris din partea Universității Politehnica Timișoara. Toate încălcările acestor drepturi vor fi penalizate potrivit Legii române a drepturilor de autor.

România, 300223 Timișoara, Bd. Vasile Pârvan 2B  
Tel./fax 0256 404677  
e-mail: editura@upt.ro

## Cuvânt înainte

Această lucrare este rezultatul unui proces îndelungat, uneori anevoios al activității de cercetare ce am desfășurat-o în cadrul DSPLabs din cadrul universității Politehnica din Timișoara.

Țin să mulțumesc în mod deosebit conducătorului meu de doctorat, prof. dr. habil. Ing. Mihai V. Micea care m-a îndrumat și susținut pe tot parcursul activității doctorale și fără de care nu aș fi luat hotărârea de a începe studiile doctorale.

Doresc să îmi exprim recunoștința și față de prof. dr. ing. Mircea Stratulat, prof. dr. ing. Marius Marcu și ș.l. dr. ing. Sebastian Fuicu pentru îndrumarea și ajutorul oferit în acești ani în cadrul comisiei de îndrumare doctorală.

Mulțumesc întregului colectiv DSPLabs pentru sprijinul acordat, prietenilor și colegilor mei ing. Gabriel Cârstoiu, ș.l. Cristina Stângaciu, ș.l. Valentin Stângaciu, ș.l. Răzvan Cioargă, ing. Ramona Cioargă. De asemenea doresc să mulțumesc ing. Dan Chiciudean pentru inițierea în lumea vastă a sistemelor încorporate.

Nu în ultimul rând doresc să mulțumesc soției mele Olivia pentru suportul și înțelegerea acordată în tot acest timp, copiilor mei, părinților care m-au susținut pe tot parcursul formării educaționale.

Timișoara, octombrie 2021

Lucian Ungurean

Familiei mele.

UNGUREAN, Lucian

**Evaluarea stării de degradare a bateriilor reîncărcabile în sisteme alimentate cu baterii**

Teze de doctorat ale UPT, Seria X, Nr. YY, Editura Politehnica, 200Z, 168 pagini, 39 figuri, 27 tabele.

ISSN:

ISBN:

Cuvinte cheie

Starea de degradare a bateriilor, sisteme de management al bateriilor, rețele neuronale recurente.

Rezumat

Datorită avansului tehnologic numărul sistemelor alimentate cu baterii a suferit o creștere rapidă. Din acest motiv, monitorizarea stării de degradare a bateriei precum și estimarea timpului rămas de utilizare a devenit o problemă importantă.

Sistemele de management al bateriilor joacă un rol important în acest context, ele dezvoltându-se de la simple sisteme de monitorizare a încărcării și descărcării bateriilor până la sisteme complexe care monitorizează multiplii parametri ai bateriei și oferă estimări în legătură cu starea de sănătate a acesteia.

Lucrarea de față abordează problema estimării stării de degradare a bateriilor reîncărcabile. Unul din obiectivele importante al acesteia este prezentarea detaliată a multitudinii de metode de estimare existente, prezentând avantajele și dezavantajele fiecăreia dintre ele în funcție de un set bine definit de criterii mergând de la chimia bateriei până la eroare de estimare și complexitatea computațională.

Dezvoltarea domeniului de inteligență artificială în tandem cu îmbunătățirile din domeniul hardware al procesoarelor a făcut ca implementarea rețelelor neuronale să fie posibilă pe sisteme încorporate.

Unul din cele mai importante obiective ale acestei teze este definirea unei metode noi de estimare a stării de degradare a bateriilor bazate pe rețele neuronale de tip GRU. Evaluarea performanțelor metodei implementată pe PC într-un mediu de simulare este detaliată în comparație cu implementările reale pe diferite sisteme încorporate.

Nu în ultimul rând, un alt obiectiv important este definirea unui sistem cadru pentru evaluarea performanțelor metodelor de estimare a stării de degradare a bateriilor pe sisteme încorporate, în termeni de complexitate computațională, consum de energie și eroare de estimare. Utilitatea acestui sistem cadru este compararea metodelor existente precum și sprijinirea dezvoltării unor metode noi de estimare.



## CUPRINS

Listă de acronime .....	7
Listă tabele .....	8
Lista de figuri .....	9
1. Introducere .....	12
1.1 Domeniile și tema cercetării.....	12
1.2 Motivația cercetării .....	12
1.3 Obiective propuse.....	13
1.4 Structura lucrării .....	13
2. Noțiuni teoretice .....	14
2.1 Sisteme de management al bateriilor.....	14
2.2 Starea de degradare a bateriilor .....	15
2.3 Modelarea bateriilor .....	16
2.3.1 Modelul electric .....	16
2.3.2 Modelul electrochimic.....	17
2.3.3 Modelul matematic .....	17
2.3.4 Modelul bazat pe ciclul de viață a bateriei .....	18
2.4 Rețele neuronale pentru estimarea seriilor de timp .....	19
2.4.1 Taxonomia problemelor legate de estimarea seriilor de timp .....	19
2.4.2 Rețele neuronale de tip MLP.....	20
2.4.3 Rețele neuronale de tip LSTM.....	21
2.4.4 Rețele neuronale de tip GRU .....	26
3. Abordări curente privind metodele pentru evaluarea stării de degradare a bateriilor .....	29
3.1 Discuție asupra metodelor existente de estimare a stării de degradare a bateriilor.....	29
3.1.1 Coulomb Counting .....	30
3.1.2 OCV.....	31
3.1.3 Spectroscopia de impedanță .....	32
3.1.4 Filtre Kalman .....	34
3.1.5 Logică Fuzzy .....	37
3.1.6 Vectori suport .....	38
3.1.7 Filtre particule.....	40
3.1.8 Alte metode .....	43
3.2 Evaluarea comparativă a metodelor existente de estimare a SOH/RUL .....	46
3.3 Concluzii .....	52
4. Metodă de evaluare a stării de degradare a bateriilor bazată pe regresie polinomială .....	54
4.1 Descrierea metodei.....	54
4.2 Implementarea metodei .....	56
5. Metodă de evaluare a stării de degradare a bateriilor folosind rețele neuronale recurente de tip GRU .....	60
5.1 Descrierea metodei.....	60
5.2 Structura rețelei neuronale .....	62
5.3 Antrenarea rețelei .....	63
5.4 Variantă modificată prin adăugarea unui filtru treaptă.....	64
6. Sistem cadru de analiză a performanțelor metodelor de estimare a stării de degradare a bateriilor bazate pe rețele neuronale .....	66
6.1 Obiective.....	66
6.2 Structura sistemului .....	66

6.3 Criterii de evaluare și metrici de performanță.....	67
6.4 Exemplu de evaluare a performanței folosind implementarea PnP.....	68
7. Evaluarea performanțelor.....	71
7.1 Criterii de evaluare .....	71
7.2 Configurarea metodei .....	71
7.3 Rezultatele evaluării metodei pe setul de date NASA PCOE .....	71
7.3.1 Evaluarea performanțelor pentru scenariul 25-DEG-CC.....	72
7.3.2 Evaluarea performanțelor pentru scenariul ALL-DEG-CC .....	73
7.3.3 Evaluarea performanțelor pentru scenariul 25-DEG-CC-RW.....	73
7.3.4 Evaluarea performanțelor pentru scenariul ALL-DEG-CC-RW .....	74
7.3.5 Evaluarea performanțelor pentru scenariul 25-DEG-CC-SF.....	75
7.3.6 Evaluarea performanțelor pentru scenariul ALL-DEG-CC-SF .....	75
7.3.7 Evaluarea performanțelor pentru scenariul 25-DEG-CC-RW-SF.....	76
7.3.8 Evaluarea performanțelor pentru scenariul ALL-DEG-CC-RW-SF .....	76
7.3.9 Analiza repetabilității pentru scenariul ALL-DEG-CC-RW-SF.....	77
7.3.10 Analiza overfit/underfit pentru scenariul ALL-DEG-CC-RW-SF .....	77
7.3.11 Analiza structurii rețelei neuronale folosind scenariul ALL-DEG-CC-RW-SF..	78
7.3.12 Analiza crossvalidare folosind scenariul ALL-DEG-CC-RW-SF.....	78
7.4 Discutarea rezultatelor și studiul comparativ cu alte metode .....	81
8. Concluzii și perspective .....	85
8.1 Concluzii .....	85
8.2 Rezumat al contribuțiilor .....	86
8.3 Perspective de cercetare și dezvoltare.....	86
Referințe bibliografice.....	87
Publicații.....	94
Anexa 1. Descrierea setului de date NASA Prognostics center of excellence .....	95
Anexa 2. Articol 'Battery state of health estimation: a structured review of models, methods and commercial devices' în varianta preprint apărut în , Int J Energ Res Int J Energ Res, vol. 41, no. 2, pp. 151–181, Feb. 2017 .....	98
Anexa 3. Articol , 'Online state of health prediction method for lithium-ion batteries, based on gated recurrent unit neural networks' în variantă pre-print, apărut în, International Journal of Energy Research, vol. 44, no. 8, pp. 6767–6777, Apr. 2020 .....	134
Anexa 4. Articol 'Online State-of-Health Assessment for Battery Management Systems', în variantă pre-print, apărut în Ieee Inst. And Meas., vol. 60, no. 6, pp. 1997–2006, Jun. 2011.....	152

## LISTĂ ACRONIME

ARMA (autoregressive moving average)	37
BMS (Battery management system)	14
DC (direct current)	14
DoD (depth of discharge)	12
EEPROM (electrically erasable programmable read-only memory)	15
EIS (electronic impedance spectroscopy)	32
EKF (extended Kalman Filter)	37
FFNN (feed forward neural network)	20
GRU (Gated recurrent unit)	13
I2C (inter integrated circuit)	15
LCD (liquid crystal display)	15
LED (light emitting diode)	15
LSTM (long short term memory)	21
MLP (multilayer perceptron)	20
OCV (open circuit voltage)	16
PnP (Paper and Pencil)	66
RMSE (root mean square error)	31
RNN (recurrent neural network)	33
RS-232 (recommended standard 232)	15
RUL (remaining useful life)	15
RVM (relevance vector machines)	39
SOC (state of charge)	31
SoH (State of health)	12
SPI (serial peripheral interface)	15
SVR (support vector regression)	36

## LISTĂ TABELE

Tabel 1. Operatori de logica fuzzy	37
Tabel 2. Evaluarea metodelor de estimare a stării de degradare a bateriilor	47
Tabel 3. Configurarea metodei de estimare bazată pe rețele neuronale de tip GRU	71
Tabel 4. Caracteristici de antrenare a rețelei	71
Tabel 5. Analiza structurii rețelei neuronale	78
Tabel 6. Valoarea parametrului de corelație pentru pașii de crossvalidare	79
Tabel 7. Lista platformelor HW și configurarea SW pentru evaluarea implementării metodei pe sisteme încorporate	80
Tabel 8. Evaluarea implementării metodelor pe sisteme încorporate	81
Tabel 9. Comparație între rețele neuronale LSTM și GRU [3]	83

## LISTĂ FIGURI

Fig. 1. Structura generală a unui BMS [2]	14
Fig. 2. Modelul Randles [4]	16
Fig. 3. Modelul Electrochimic [12]	17
Fig. 4. Arhitectura rețelei neuronale MLP [30]	20
Fig. 5. Calea de date și operațiile din cadrul rețelei neuronale MLP [30]	21
Fig. 6. Arhitectura rețelei neuronale LSTM	22
Fig. 7. Căile de date dintr-o rețea LSTM [35]	23
Fig. 8. Starea celulei LSTM [35]	23
Fig. 9. Poartă în cadrul celulei [35]	23
Fig. 10. Controlul memoriei pe termen lung [35]	24
Fig. 11. Controlul memoriei pe termen scurt [35]	25
Fig. 12. Actualizarea stării celulei LSTM [35]	25
Fig. 13. Calea de date pentru ieșire [35]	26
Fig. 14. Arhitectura rețelei neuronale GRU	26
Fig. 15. Structura internă și căile de date din cadrul celulei GRU [3]	27
Fig. 16. Clasificarea metodelor de estimare a stării de degradare a bateriilor	29
Fig. 17. Modelul electric simplificat	31
Fig. 18. Diagrama Niquist obținută prin EIS [16]	32
Fig. 19. Impedanța estimată și cea măsurată partea reală [16]	33
Fig. 20. Variante de filtre Kalman utilizare în sisteme neliniare	34
Fig. 21. Arhitectura Filtrului Kalman Dual	36
Fig. 22. Rezultatele predicției folosind filtru de particule [73]	40
Fig. 23. Transformare MPF a funcției de distribuție de probabilitate Posterioră	42
Fig. 24. Algoritm de estimare RUL propus in [87]	45
Fig. 25. Placa de management al consumului în cadrul unui WIT	57
Fig. 26. Schema de tip high-side pentru măsurarea curentului	57
Fig. 27. Arhitectura software al plăcii de management al consumului	58
Fig. 28. Schema de principiu a metodei de estimare a stării de degradare a Bateriilor bazată pe rețele neuronale de tip GRU	60
Fig. 29. Structura rețelei neuronale cu celule de tip GRU	63
Fig. 30. Efectul de regenerare întâlnit la unele tipuri de baterii	64
Fig. 31. Capacitatea bateriei B01(Panasonic NCR18650A)	65
Fig. 32. Metoda de estimare modificată prin introducerea filtrului treaptă	65
Fig. 33. Structura sistemului cadrul de evaluare a performanțelor metodelor de estimare a stării de degradare a bateriilor	67
Fig. 34. Eroare relativă procentuală pentru cele 8 scenarii de evaluare semnificative	72
Fig. 35. Distribuția erorii de estimare în cadrul scenariului 25-DEG-CC	73
Fig. 36. Distribuția erorii de estimare în cadrul scenariului ALL-DEG-CC	73
Fig. 37. Distribuția erorii de estimare în cadrul scenariului 25-DEG-CC-RW	74
Fig. 38. Distribuția erorii de estimare în cadrul scenariului ALL-DEG-CC-RW	74
Fig. 39. Distribuția erorii de estimare în cadrul scenariului 25-DEG-CC-SF	75
Fig. 40. Distribuția erorii de estimare în cadrul scenariului ALL-DEG-CC-SF	75
Fig. 41. Distribuția erorii de estimare în cadrul scenariului 25-DEG-CC-RW-SF	76
Fig. 42. Distribuția erorii de estimare în cadrul scenariului ALL-DEG-CC-RW-SF	76

Fig. 43. Analiza repetabilității pentru scenariul ALL-DEG-CC-RW-SF	77
Fig. 44. Analiza overfitt/underfitt pentru scenariul ALL-DEG-CC-RW-SF	78
Fig. 45. Parametru de corelație pentru pasul 1 de crossvalidare	79
Fig. 46. Platforme fizice folosite	80
Fig. 47. Acuratețea estimării pentru diferite metode de estimare a stării de degradare a bateriilor [105]	82
Fig. 48. Studiu comparativ a principalelor metode de estimare cu implementare pe sisteme încorporate [3]	83



# 1. INTRODUCERE

## 1.1. Domeniile și tema cercetării

Teza de față tratează probleme dintr-o multitudine de domenii interconectate, dintre care cele mai importante sunt:

- Domeniul bateriilor - un domeniu complex care a cunoscut o evoluție accelerată în ultimul deceniu datorită progresului tehnologiilor de fabricare a compușilor chimici;
- Domeniul sistemelor încorporate alimentate cu baterii reîncărcabile – un domeniu extins ce a cunoscut o creștere exponențială datorită evoluției tehnologice din domeniul bateriilor - având o plajă de aplicabilitate în toate domeniile de activitate: militar, aeronautic, automotive, medical, produse electronice de larg consum.

Având în vedere aceste două domenii principale, cercetarea abordează următoarele subdomenii:

- Subdomeniul sistemelor de management al bateriilor;
- Subdomeniul metodelor de estimare a stării de degradare a bateriilor - un domeniu nou care prezintă un interes deosebit, fapt demonstrat prin numărul mare de publicații recente;
- Subdomeniul inteligenței artificiale - un domeniu vast, cu o vechime de peste 60 de ani care a cunoscut de-a lungul timpului o creștere spectaculoasă susținută de evoluția puterii de calcul.

Tema lucrării de cercetare elaborată în această teză este evaluarea stării de degradare a bateriilor reîncărcabile în sisteme alimentate cu baterii.

## 1.2. Motivația cercetării

În domeniul dispozitivelor încorporate alimentate cu baterii reîncărcabile, sistemele de management al bateriilor joacă un rol important, urmărind îndeplinirea următoarelor funcționalități [1]:

- Asigurarea unui mediu de operare sigur, definit de specificațiile tehnice ale bateriei și tipului acesteia, precum curentul maxim de încărcare, curentul maxim de descărcare, limitele inferioare și superioare ale nivelului DoD, nivele inferioare și superioare ale tensiunii bateriei, etc.
- Monitorizarea continuă a parametrilor bateriilor pentru determinarea și predicția stării acesteia.

Soluțiile de management al bateriilor constau de obicei într-un dispozitiv hardware pe care rulează o aplicație software de management. Algoritmii de determinare și predicție a stării de sănătate a bateriei variază de la cei mai simpli care pot să opereze pe un hardware cu resurse limitate, până la cei mai complecși care necesită resurse hardware suplimentare.

Legat de evaluarea stării de degradare a bateriei, literatura de specialitate menționează următoarele probleme de actualitate:

- Problema neliniarității și a incertitudinii de estimare: datorată modului de utilizare a bateriilor precum și natura parametrilor acesteia.



- Problema complexității algoritmilor de estimare și a resurselor limitate: datorată tipului de algoritm de estimare. Cele mai complexe metode de estimare necesită resurse hardware și putere de procesare pe care microcontrolere de uz general nu le pot oferi, acestea rămânând adeseori la nivel conceptual sau la nivel de simulare în medii precum Matlab.

Activitatea de cercetare sintetizată prin teza de față abordează cele două probleme mai sus menționate și are ca scop ameliorarea acestora prin dezvoltarea unei noi metode de estimare precum și validarea acestei metode într-un cadru complet de analiză a performanțelor bazat pe criterii atent definite.

### 1.3. Obiective propuse

Principalul scop al acestei teze de doctorat este găsirea unor soluții pentru îmbunătățirea sau chiar rezolvarea unor problemele actuale în cadrul metodelor de estimare a stării de degradare a bateriilor. Activitatea de cercetare are ca principale obiective:

- Definirea unor metode de estimare a stării de degradare al bateriilor reîncărcabile care să prezinte un nivel ridicat de performanță și de eficiență, adică:
  - acuratețe de estimare cât mai ridicată în ceea ce privește SoH, respectiv
  - să poată fi implementate și executate pe platforme digitale încorporate, cu resurse limitate (de memorie, de procesare, consum de energie, etc.).
- Definirea unui cadru de evaluare a performanțelor metodelor de estimare a SoH, care să cuprindă:
  - criterii de precizie și acuratețe,
  - criterii de evaluare a complexității algoritmilor,
  - criterii de evaluare a resurselor de microprocesor utilizare precum și a consumului propriu de energie al algoritmului.

### 1.4. Structura lucrării

Structura lucrării este prezentată în cele ce urmează.

Capitolul 2 cuprinde câteva noțiuni teoretice care sunt folosite în domeniul cercetării curente. Printre cele mai importante noțiuni sunt: descrierea și structura unui sistem de management al bateriilor, definirea stării de degradare a bateriilor, modele de baterii și câteva noțiuni despre rețele neuronale folosite în estimarea seriilor de timp. Capitolul 3 conține o descriere detaliată a abordărilor curente în ceea ce privește metodele de evaluare a stării de degradare a bateriilor. Descrierea metodei propuse pentru estimarea SoH pe baza regresiei polinomiale, este cuprinsă în capitolul 4. Capitolul 5 conține detalii despre o nouă metodă de estimare a stării de degradare a bateriilor, bazată pe rețele neuronale de tip GRU. Capitolul 6 descrie sistemul cadru de analiză a performanțelor metodelor de estimare a stării de degradare a bateriilor. În capitolul 7 sunt prezentate rezultatele evaluării performanțelor metodelor de estimare propuse, pentru un set extins de scenarii. Capitolul 8 conține concluziile și perspectivele de cercetare și dezvoltare. Lucrarea se încheie cu referințele bibliografice, respectiv lista de publicații ce au reieșit din cercetarea curentă precum și principalele anexe.

## 2. NOȚIUNI TEORETICE

### 2.1. Sisteme de management al bateriilor

Sistemele de management al bateriilor sunt componentele esențiale din cadrul sistemelor alimentate cu baterii. Principala funcție este asigurarea utilizării bateriei într-un mod cât mai eficient în vederea minimizării stării de degradare. Acest lucru se obține prin monitorizarea parametrilor bateriei în procesul de încărcare și procesul de descărcare al acesteia. De asemenea sistemele de management al bateriilor oferă informații utile despre ciclurile de încărcare/descărcare, timpul de utilizare rămas, etc.

În figura 1 este descrisă structura generală a unui sistem de management al bateriilor (BMS) [2]. Modulul de alimentare este conectat la rețea și asigură energia necesară încărcării bateriei. Acesta poate fi parte integrat pe placa sistemului sau poate fi detașabil. Convertorul DC/DC este folosit pentru a asigura nivelul de tensiune pentru funcționarea consumatorului. Acesta translatează nivelul de tensiune al bateriei la nivelul de tensiune dorit. Bateria este încărcată de la modulul de alimentare. Aceasta este prevăzută cu circuitele de protecție pentru asigurarea unui anumit grad de siguranță. Modulul de monitorizare a stării bateriei are rol de achiziție/determinare a parametrilor bateriei

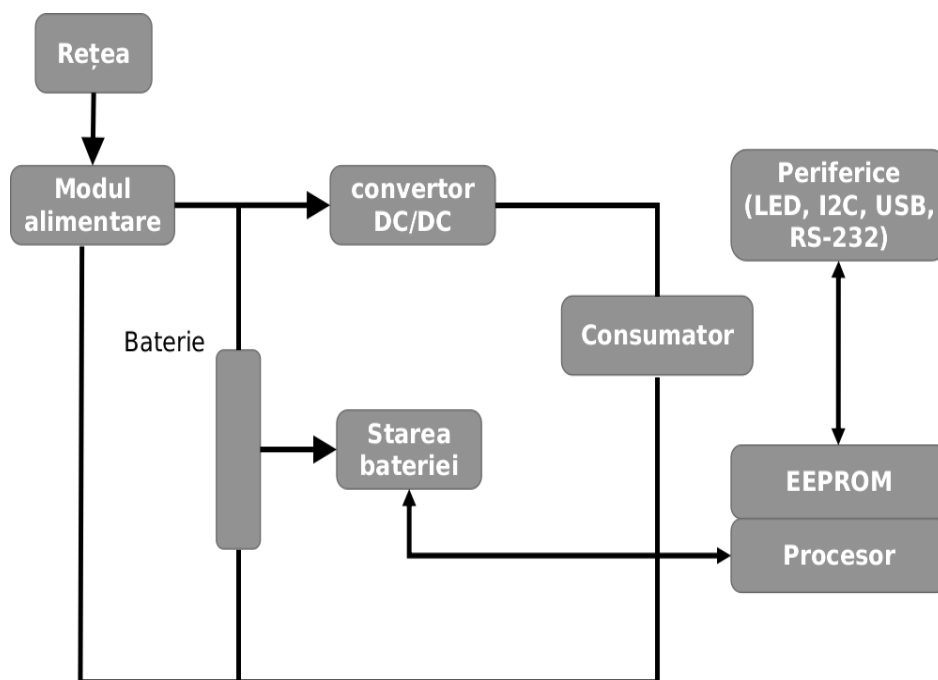


Fig. 1. Structura generală a unui BMS [2]

În cadrul sistemului BMS, sunt monitorizați activ parametri precum tensiunea bateriei, curentul de încărcare/descărcare, temperatura bateriei, etc. Pe baza acestora se pot determina alți parametri precum capacitatea, rezistența internă, starea de degradare (SoH), numărul de cicluri rămăși până bateria este considerată neutilizabilă (RUL).

Modulul de comunicare este format dintr-un procesor/micro controller folosit pentru a salva datele în memoria EEPROM și pentru a transmite mai departe informațiile printr-una dintre interfețele de comunicație (I2C, RS-232, SPI, etc.) sau afișarea stării bateriei printr-un indicator LED sau chiar informații pe un afișaj LCD. Consumatorul transformă energia electrică în energia corespunzătoare destinației acestuia: energie mecanică, sunet, lumină, etc.

## 2.2. Starea de degradare a bateriilor

Starea de degradare a bateriilor se poate reprezenta prin diferite metrici, cele mai importante sunt următoarele: starea de degradare (SoH) și timpul util de operare rămas (RUL).

Starea de degradare (SoH) este o metrică raportată la starea inițială a bateriei. În momentul producerii bateriei, SoH este considerat 100 % – corelat cu valoarea maximă a energiei pe care bateria poate să o furnizeze. Utilizând bateria, apar fenomenele fizice și chimice ireversibile care duc la scăderea energiei pe care bateria poate să o furnizeze. Când energia furnizată scade sub un anumit nivel, bateria este considerată degradată și trebuie schimbată. În acest moment SoH este considerat 0%.

O definiție simplificată a SoH pe care am folosit-o în această lucrare se bazează pe capacitatea bateriei [3]:

$$SOH [\%] = \begin{cases} 100\%, & \frac{C_{bat}}{C_{nom}} > 1 \\ \left( 1 - \frac{1 - \frac{C_{bat}}{C_{nom}}}{SOH_0} \right) \cdot 100 [\%] & \\ 0\%, & \frac{C_{bat}}{C_{nom}} < SOH_0 \end{cases} \quad (1)$$

În care  $C_{bat}$  reprezintă capacitatea curentă a bateriei,  $C_{nom}$  – capacitatea nominală a bateriei,  $SOH_0$  – raportul dintre capacitatea bateriei și capacitatea nominală pentru care bateria este considerată degradată. În mod uzual,  $SOH_0$  are valoarea 0,2 sau 0,3.

În ceea ce privește timpul util de operare rămas (RUL) este strâns legat de SoH și se definește prin numărul de cicluri de încărcare/descărcare sau chiar timpul până când bateria ajunge la SoH 0% și este considerată degradată (neutilizabilă).

## 2.3. Modelarea bateriilor

Modelarea bateriilor este un subiect important în domeniul estimării stării de degradare a bateriilor. Orice metodă de estimare se bazează direct sau indirect pe un model.

Cele mai întâlnite tipuri de modele sunt: modelul electric, modelul electrochimic, modelul matematic, modelul bazat pe ciclul de viață a bateriei.

### 2.3.1. Modelul electric

*Modelul electric* este cunoscut și sub forma modelului de circuit echivalent – modelul Randles sau Thevenin. Este compus dintr-o sursă de tensiune ideală și unul sau mai multe grupuri paralele de rezistențe – condensatori conectați în serie cu o rezistență [4]-[11].

Sursa ideală de tensiune reprezintă tensiunea în gol a bateriei, (*OCV* sau  $U_{OC}$ ) și este un parametru destul de important ce se regăsește în centrul multor metode de estimare. Rezistența în serie  $R_{\Omega}$  modelează rezistența internă a bateriei când la bornele bateriei se aplică un curent constant.

Grupul  $R_p$ - $C_p$  modelează comportamentul dinamic al bateriei, când la bornele bateriei curentul variază

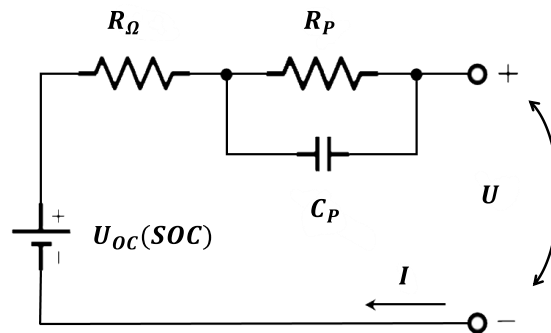


Fig. 2. Modelul Randles [4]

Sistemul de ecuații ce stă la baza acestui model este următorul [4]:

$$\begin{cases} U = U_{OC}(SoC) + IR_{\Omega} + U_p \\ I = \frac{U_p}{R_p} + C_p \frac{dU_p}{dt} \end{cases} \quad (2)$$

în care,  $U_{OC}$  este tensiunea în gol,  $R_{\Omega}$  este rezistența internă a bateriei,  $R_p$ - $C_p$  este grupul paralel rezistență – condensator,  $U$  este tensiunea de alimentare a circuitului,  $I$  este curentul din circuit.

### 2.3.2. Modelul electrochimic

*Modelul electrochimic* are ca scop aproximarea cât mai precisă a proceselor chimice ce au loc în interiorul celulei bateriei în timpul încărcării, descărcării și relaxării. Pentru o baterie de Li-Ion, din punct de vedere chimic, aceasta este compusă din trei elemente: electrodul negativ, separatorul (electrolitul) și electrodul pozitiv. În acest tip de baterie, ionii de Li se deplasează de la electrodul negativ spre cel pozitiv când bateria se descarcă, iar la încărcare de la electrodul pozitiv la cel negativ [12] - [13].

O variantă a modelului electrochimic derivat din modelul Randles de ordin 2 este prezentat în [12]. Impedanța bateriei se poate exprima prin următoarea ecuație:

$$Z_{batt} = j\omega L + \frac{Z_{n,SEI}}{S_n} + R_0 + \frac{Z_p}{S_p} \quad (3)$$

în care,  $Z_{batt}$  este impedanța bateriei,  $Z_{n,SEI}$  este impedanța anodului,  $Z_p$  este impedanța catodului, iar  $S_n$  și  $S_p$  sunt ariile suprafețelor anodului și catodului. Modelul electrochimic poate fi extins și la celelalte tipuri de chimii, cum ar fi cele bazate pe Acid [15], Ni-MH.

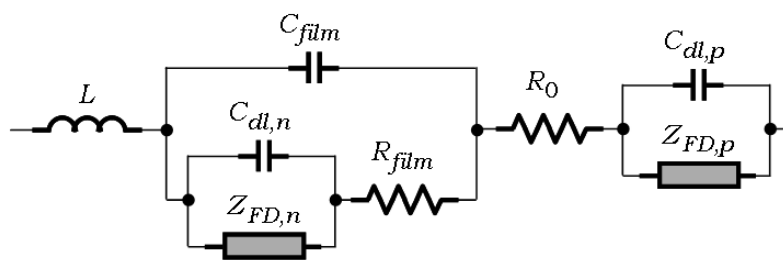


Fig. 3. Modelul Electrochimic [12]

O abordare complexă este prezentată în [17]-[19] în care concentrația de ioni și reacțiile chimice sunt modelate prin ecuații diferențiale.

### 2.3.3. Modelul matematic

*Modelul matematic* folosește în general unul dintre parametri (electrici) ai bateriei pentru a modela degradarea acesteia [20]-[25]. În [20], estimarea degradării bateriei se face prin utilizarea curbelor CC-CV (curent constant – tensiune constantă) și o analiză de tip calendar. Pentru fiecare ciclu de încărcare, curentul poate fi exprimat folosind:

$$I(t) = Ae^{-Bt} + C \quad (4)$$

în care  $A$ ,  $B$ ,  $C$  sunt scalari ce reprezintă parametrii modelului.

Luând în considerare relația dintre parametrii modelului și degradarea capacității bateriei ( $C_{loss}$ ), ecuația se poate rescrie:

$$I(t, C_{loss}) = A(C_{loss})e^{-B(C_{loss})t} + C(C_{loss}) \quad (5)$$

Autorii au arătat că există o relație liniară între capacitatea rămasă și parametrul B al modelului, o scădere a capacității bateriei duce la o scădere a valorii parametrului B.

În [21]-[22], tensiunea în gol a bateriei este folosită pentru a modela degradarea bateriei. Prin intermediul analizei incrementale (ICA) se trasează curbele ce denotă variația capacității cu tensiunea în gol. Ecuația ce stă la baza modelului este:

$$OCV(z) = K_0 - \frac{K_1}{z} - K_2 z + K_3 \ln(z) + K_4 \ln(1-z) \quad (6)$$

în care  $K_{0-4}$ , sunt parametri ai modelului,  $z$  este valoarea normalizată a SoC. Pentru a obține rezultate mai bune se poate folosi o variantă complexă a ecuației:

$$OCV(z) = K_0 - K_1 \frac{1}{1 + e^{a_1(z-\beta_1)}} - K_2 \frac{1}{1 + e^{a_2(z-\beta_2)}} + K_3 \frac{1}{1 + e^{a_3(z-1)}} + K_4 \frac{1}{1 + e^{a_4 z}} + K_5 z \quad (7)$$

În [23], capacitatea bateriei este estimată folosind curbele ridicate din măsurători la fiecare ciclu al bateriei. Sunt evaluate trei modele – cel exponențial, cel polinomial, și cel hibrid:

$$\begin{aligned} C_{Ak} &= C_{Ak1} + C_{Ak2} = a_1 e^{a_2 k} + a_3 e^{a_4 k} \\ C_{Bk} &= C_{Bk1} + C_{Bk2} = \beta_1 k^2 + \beta_2 k + \beta_3 \\ C_{Ck} &= \gamma_1 e^{\gamma_2 k} + \gamma_3 k^2 + \gamma_4 \end{aligned} \quad (8)$$

în care  $C_{Ak}$ ,  $C_{Bk}$  și  $C_{Ck}$  sunt valorile capacității bateriei,  $k$  reprezintă ciclul de descărcare, iar  $a_1, a_2, a_3, a_4; \beta_1, \beta_2, \beta_3; \gamma_1, \gamma_2, \gamma_3$ , sunt parametri modelului. Experimentele arată că modelul hibrid oferă o acuratețe mai bună pentru predicția capacității.

#### 2.3.4. Modelul bazat pe ciclul de viață a bateriei

Modelul bazat pe ciclul de viață a bateriei este un alt tip de model folosit pentru determinarea degradării bateriei. Acest tip de model urmărește parametri bateriei și relația dintre aceștia și procesul de îmbătrânire. Starea de degradare se determină prin observații multiple după teste offline. Acest mod este diferit de celelalte modele care monitorizează parametri în timp real și estimează starea de degradare bazat pe valori instantanee ale parametrilor.

Un astfel de model este utilizat în [26]. Parametrii folosiți sunt: temperatura bateriei, curentul de încărcare, curentul de descărcare, nivelul de descărcare. Este analizat impactul acestor parametri asupra capacității bateriei. Modelul propus a fost dezvoltat și validat în Matlab.

Autorii din [16] propun să monitorizeze următorii parametri ai bateriei: temperatura, capacitatea, numărul de cicluri încărcare/descărcare. Pentru predicția stării de degradare sunt folosiți algoritmi bazați pe vectori suport. Aceste tipuri de

tehnică implică antrenarea pe un set imens de date. Pentru a scurta timpul de observare, autorii au folosit teste de îmbătrânire accelerată. Rămâne discutabil dacă acest teste se mapează direct pe condițiile de utilizare reale ale bateriei.

În [27]-[28], modelul de ciclu de viață se bazează pe modele de fiabilitate. Aceste modele sunt folosite împreună cu celelalte tipuri de modele pentru a îmbunătăți estimarea stării de degradare.

## 2.4. Rețele neuronale pentru estimarea seriilor de timp

### 2.4.1. Taxonomia problemelor legate de estimarea seriilor de timp

Fie un set de date de intrare  $X = \{x_1, x_2, \dots, x_n\}$ , pentru care cunoaștem valorile de ieșire  $Y = \{y_1, y_2, \dots, y_m\}$ . Estimarea seriilor de timp presupune găsirea unei funcții  $f(X) = Y$ , care să realizeze o mapare cât mai fidelă între valorile de intrare și cele de ieșire.

Algoritmii de estimare a seriilor de timp bazați pe învățare automată presupun existența unui set de date cunoscut pentru care se deduce funcția de estimare  $f$ . Performanța algoritmului se obține aplicând funcția obținută pe un set de date necunoscut, și urmărind diferite metrice și criterii de evaluare.

Problemele de predicție a seriilor de timp presupun analizarea mai multor aspecte [29], câteva dintre cele mai importante fiind enumerate în continuare:

- **Intrări vs. ieșiri** – în general o predicție presupune folosirea unor observații anterioare pentru a estima observații viitoare. Intrările sunt datele obținute în observația curentă pentru care se face predicția – valorile de ieșire. Nu trebuie confundate intrările cu datele pentru antrenarea algoritmului.
- **Date endogene vs. date exogene** – variabilele de intrare se consideră endogene dacă sunt afectate de alte variabile și variabilele de ieșire depind de acestea. De exemplu în seriile de timp, o observație la timpul  $t$  este dependentă de cea de la  $t - 1$ , cea de la  $t - 1$  depinde de  $t - 2$  etc. O variabilă de intrare este exogenă dacă este independentă de celelalte variabile din sistem și variabila de ieșire depinde de aceasta.
- **Regresie vs. Clasificare** – în cadrul problemelor de regresie, estimarea obține o valoare. Aceasta este numerică și poate fi asimilată cu un volum, un preț, etc. Problemele de clasificare sunt cele în care valoarea predicției este o categorie. O categorie de date conține etichete bine definite – de exemplu, mare, mic, rece, cald, albastru, verde, etc.
- **Date structurate vs. Nestructurate** – de multe ori este important de analizat setul de date deoarece acesta poate conține tipare – de repetiție, etc. care poate să ușureze într-o anumită măsură algoritmul de predicție. Deci putem spune că datele care urmează un anumit tipar sau ciclicitate sunt date structurate, pe când cele care nu prezintă o astfel de proprietate sunt nestructurate.
- **Predicție univariată vs. Multivariată** – o singură variabilă măsurată în timp se consideră univariată. Predicția univariată presupune predicția unei singure variabile, pe când cea multivariată presupune predicția mai multor variabile.

- **Predicție într-un singur pas vs. multi-pas** – algoritmul de predicție poate să estimeze valoarea corespunzătoare următorului moment de timp sau poate să prezică un set de date ce se întinde pe mai multe instanțe de timp.
- **Model static vs. dinamic** – un model se consideră static dacă acesta nu se modifică între predicții. Un model dinamic este un model care se modifică între predicții, în general își ajustează parametrii cu scopul de a crește acuratețea estimării.
- **Date continue vs. discontinue** – un set de date pentru care observațiile sunt distribuite uniform în timp, se consideră set de date continue. Cel de-al doilea tip presupune existența unor intervale în care datele lipsesc sau sunt corupte. În acest caz este nevoie de formatarea datelor în așa fel încât să poată fi aplicate metode de estimare.

#### 2.4.2. Rețele neuronale de tip MLP

Rețelele neuronale de tip MLP (Multilayer perceptron, sau perceptron multi-strat) sunt o subclasă a rețelelor neuronale de tip *feed-forward* (FFNN) și au o structură simplă în care există o singură cale de date de la intrare spre ieșire prin care se transformă datele de intrare în date de ieșire [30]-[31]. Arhitectura nu permite legături de reacție care să lege o parte din valoarea ieșirilor de intrări după cum se poate observa în figura de mai jos:

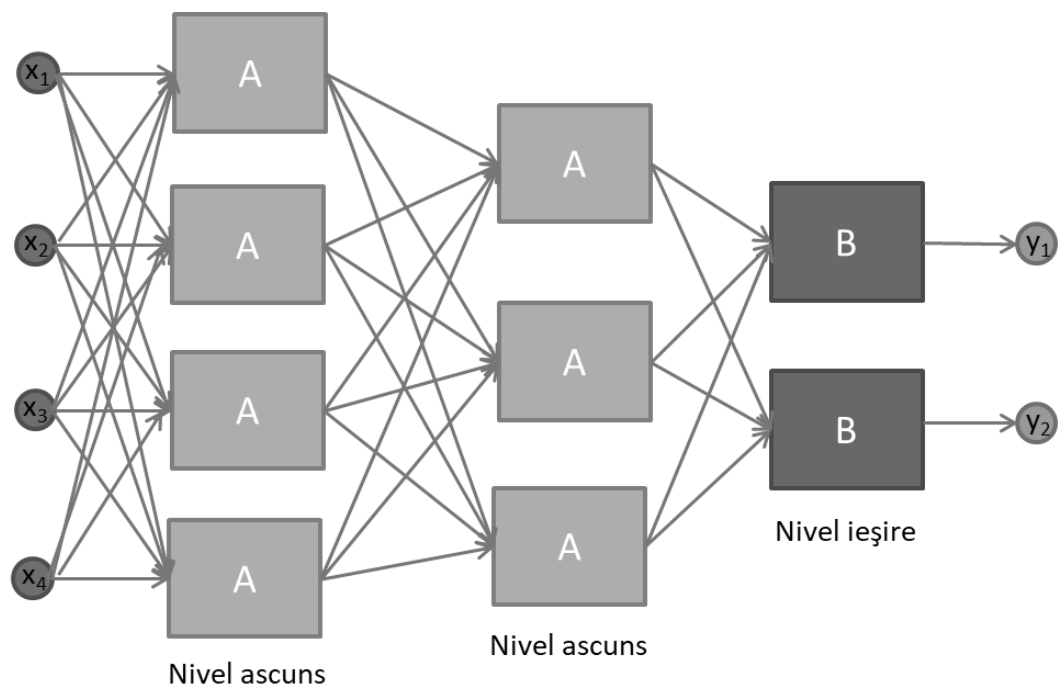


Fig.4. Arhitectura rețelei neuronale MLP [30]



În figura anterioară am reprezentat arhitectura rețelelor de tip FFNN cu două nivele ascunde cu funcția de activare A, un nivel de ieșire cu funcția de activare B, setul de intrări  $x_i, i = 1..4$  și setul de ieșiri,  $y_j, j = 1..2$ .

O astfel de rețea neuronală se poate transforma ușor într-un predictor dacă se aleg funcțiile de activare în mod corespunzător [30].

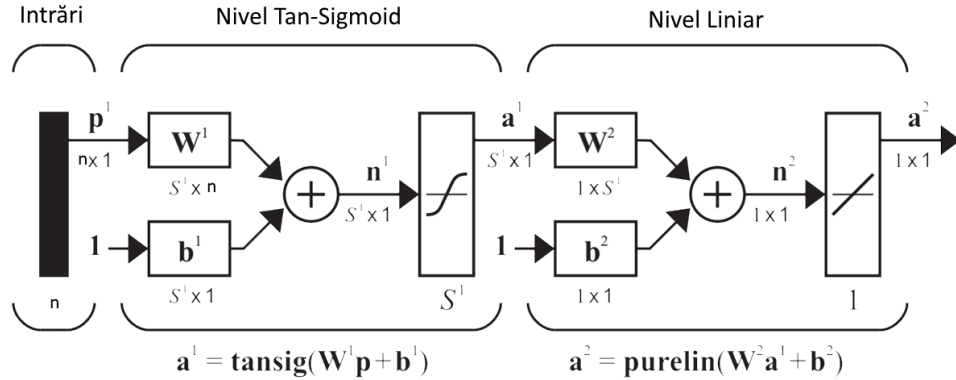


Fig.5. Calea de date și operațiile din cadrul rețelei neuronale MLP [30]

În Fig. 5 se pot distinge următoarele mărimi:  $\mathbf{p}$  este vectorul de intrare,  $\mathbf{W}^1$  este matricea ponderilor pentru nivelul 1,  $\mathbf{b}^1$  este vectorul de bias-uri pentru nivelul 1,  $\mathbf{n}^1$  este vectorul de intrare în funcția de activare al nivelului 1,  $\mathbf{a}^1$  este ieșirea din nivelul 1 și vectorul de intrare pentru nivelul 2,  $\mathbf{W}^2$  este matricea ponderilor pentru nivelul 2,  $\mathbf{b}^2$  este vectorul bias-urilor pentru nivelul 2,  $\mathbf{n}^2$  este vectorul de intrare în funcția de ieșire pentru nivelul 2, iar  $\mathbf{a}^2$  este ieșirea din nivelul 2.

Funcția de activare pentru nivelul ascuns este tan-sigmoid, iar pentru nivelul de ieșire este funcția liniară:

$$\text{tansig}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (9)$$

$$\text{lin}(x) = ax + b$$

### 2.4.3. Rețele neuronale de tip LSTM

Rețelele neuronale de tip LSTM (Long Short-Term Memory) sunt un caz particular de rețele neuronale recurente menite să contribuie la rezolvarea limitărilor MLP [32]. O listă a limitărilor MLP este următoarea:

- Dimensiunea datelor de intrare este fixă;
- Dimensiunea datelor de ieșire este fixă;
- Nu țin cont de dimensiunea temporală;

- Scalare ineficientă în cazul predicției Multivariate;
- Nu țin cont de starea sistemului – FFNN învață o singură funcție fixă și generală pe care o aplică intrărilor.

În figura de mai jos este prezentată o arhitectură de rețea neuronală de tip LSTM cu două nivele ascunde și un nivel de agregare.

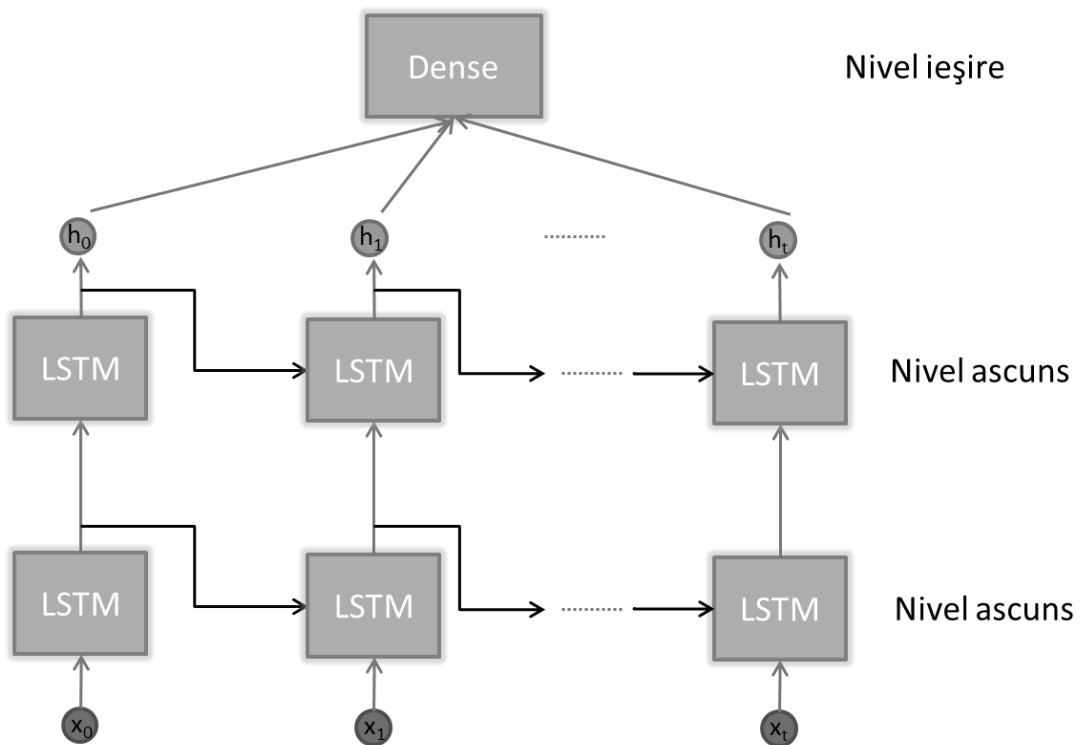


Fig.6. Arhitectura rețelei neuronale LSTM

O particularitate importantă a acestor rețele este că celula de bază a fost modificată în așa fel încât să fie capabilă de a învăța dependențele pe termen lung. Prin intermediul porților se pot controla câtă informație recentă și cu mult anterioară poate fi folosită pentru a genera ieșirea curentă [33]-[34].

Structura celulei și căile de date sunt prezentate în figura următoare [35]:

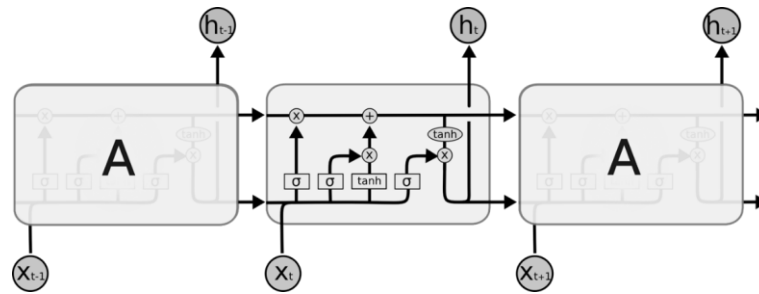


Fig. 7. Căile de date dintr-o rețea LSTM [35]

Una dintre cele mai importante căi este cea a stării celulei. Aceasta este calea de date prezentată în figura de mai jos:

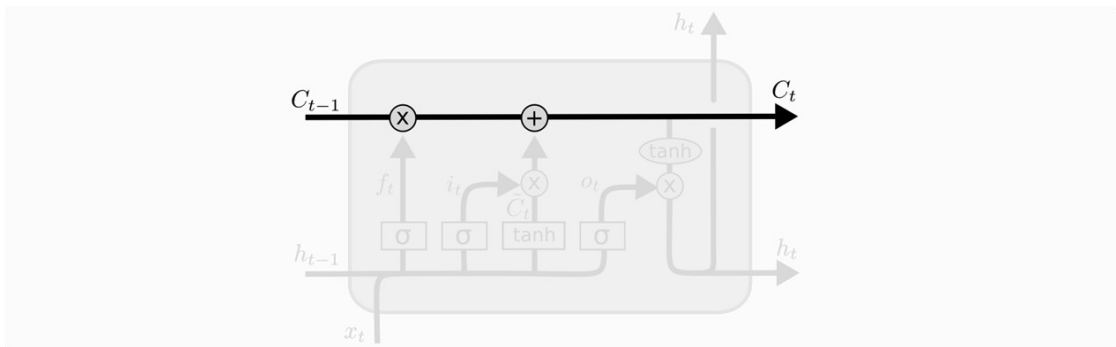


Fig. 8. Starea celulei LSTM [35]

Calea de stare a celulei LSTM este o cale directă liniară ce leagă starea de la momentul  $t$ ,  $C_t$ , cu cea de la momentul de timp  $t - 1$ ,  $C_{t-1}$ . Valoarea stării celulei se actualizează prin adăugarea sau înlăturarea de informații prin intermediul porților. Aceste porți controlează dacă informația este lăsată să treacă sau nu și conțin funcția de activare sigmoid într-un bloc multiplicativ:

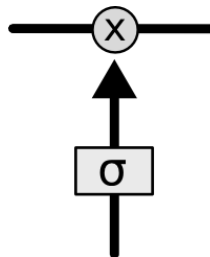


Fig. 9. Poartă în cadrul celulei [35]

Controlul memoriei pe termen lung a celulei LSTM este realizat cu ajutorul porții din figura 9 și a conexiunilor din figura 10. Modelarea matematică a acestui proces este reprezentată prin funcția din (10):

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (10)$$

în care  $f_t$  este funcția „uitării” (*forget function*),  $\sigma$  este funcția sigmoid,  $\mathbf{W}_f$  este matricea ponderilor porții  $f$ ,  $h_{t-1}$  este ieșirea de la momentul de timp  $t - 1$ ,  $x_t$  este intrarea de la momentul  $t$ ,  $b_f$  este biasul porții  $f$ .

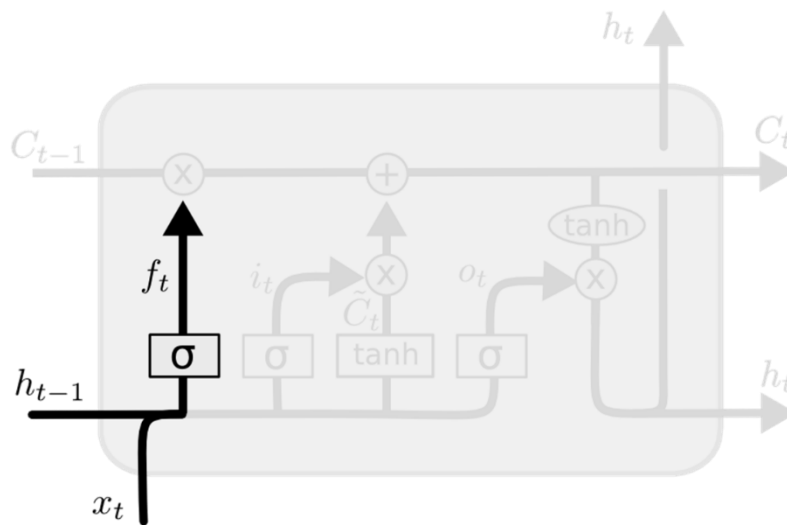


Fig. 10. Controlul memoriei pe termen lung [35]

În Fig 9. este reprezentat modul în care se reține informația ce se păstrează în cadrul celulei. Controlul memoriei pe termen scurt este format dintr-un nivel sigmoid și un vector de valori candidate pentru a fi adăugate în final stării actuale a celulei. Din punct de vedere formal, controlul memoriei pe termen scurt este definit de relațiile din (11):

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \end{aligned} \quad (11)$$

în care,  $i_t$  reprezintă poarta de intrare și controlează dacă valorile candidate intră sau nu în starea actuală a celulei,  $\mathbf{W}_i$  – matricea ponderilor porții de intrare,  $h_{t-1}$  valoarea ieșirii de la momentul  $t - 1$ ,  $x_t$  valoarea de intrare la momentul  $t$ ,  $b_i$  – biasul porții de intrare,  $\tilde{C}_t$  – vectorul valorilor candidate pentru actualizarea stării celulei,  $\mathbf{W}_C$  matricea ponderilor valorilor candidate,  $b_C$  – biasul valorilor candidate.

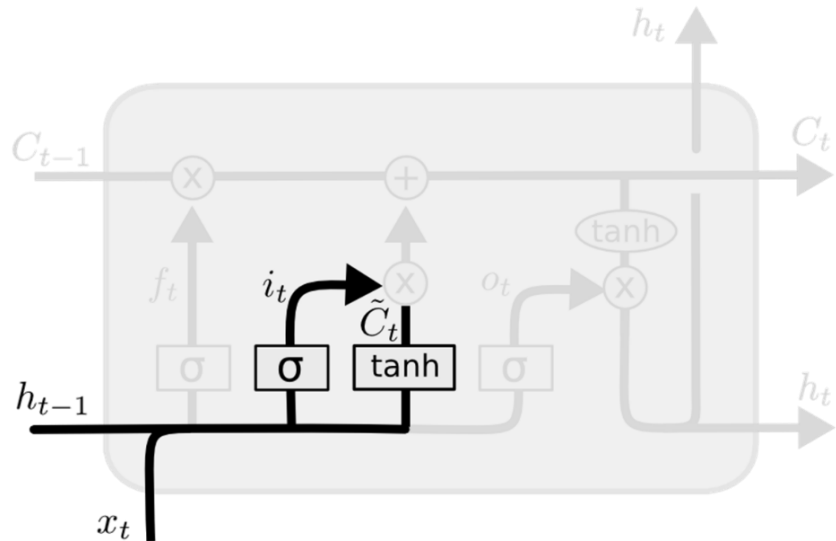


Fig. 11. Controlul memoriei pe termen scurt [35]

Combinând ecuațiile (10) și (11) putem defini ecuația stării celulei LSTM (12), reprezentată grafic în Fig. 12.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (12)$$

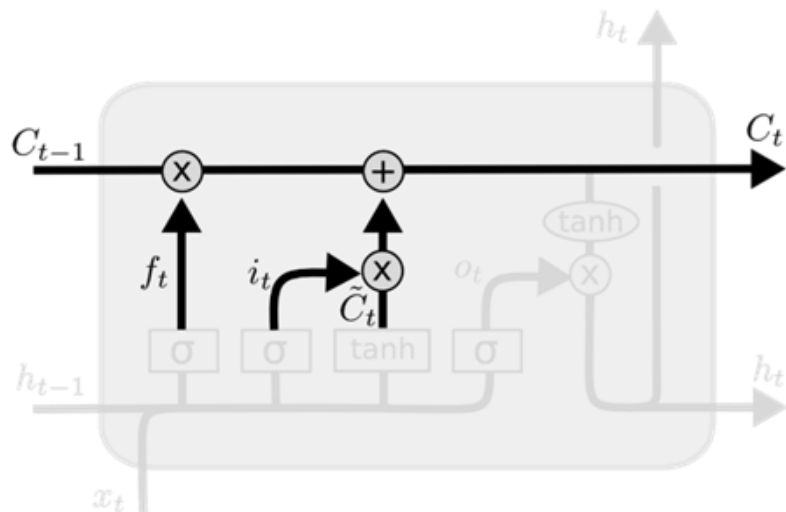


Fig. 12. Actualizarea stării celulei LSTM [35]

Ultima cale de date este cea de ieșire. Printr-un nivel sigmoid se decide ce valori se vor expune la ieșirea celulei (vezi Fig. 13 ). Acestea se calculează conform (13):

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t * \tanh(C_t) \end{aligned} \quad (13)$$

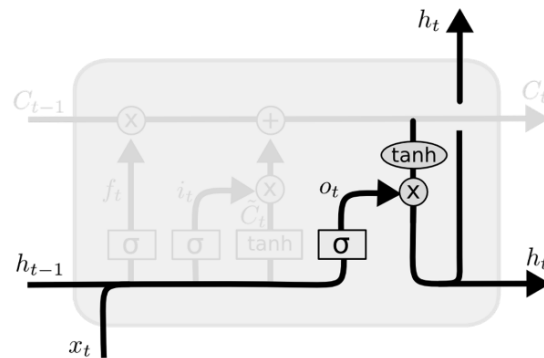


Fig. 13 Calea de date pentru ieșire [35]

#### 2.4.4. Rețele neuronale de tip GRU

În figura de mai jos este prezentată o rețea neuronală formată din două nivele ascunse de tip GRU și un nivel de agregare.

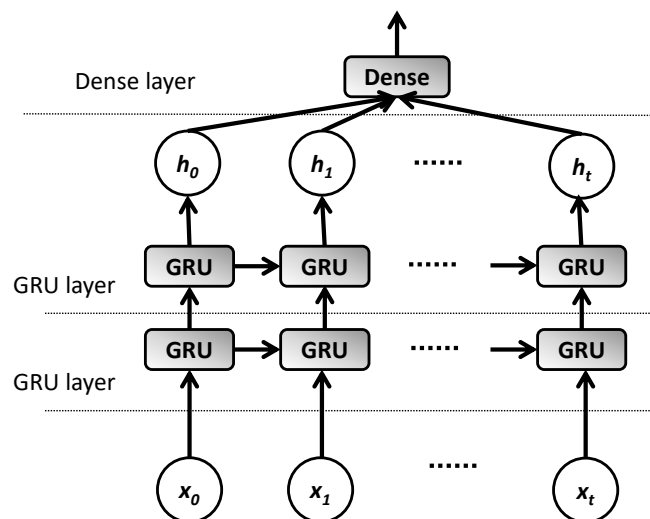


Fig. 14. Arhitectura rețelei neuronale GRU

Celula de bază a rețelelor de tip GRU este asemănătoare cu cea a celulelor de tip LSTM, o modificare importantă este că nu se mai ia în considerare starea celulei. Rețelele neuronale de tip GRU folosesc două porți: poarta de actualizare prin care se controlează cantitatea de informație din trecut ce trebuie luată în considerare și poarta de resetare care controlează câtă informație va fi uitată, respectiv nu va fi luată în considerare.

În figura 15 este detaliată structura internă și calea de date în interiorul unei celule de tip GRU [3].

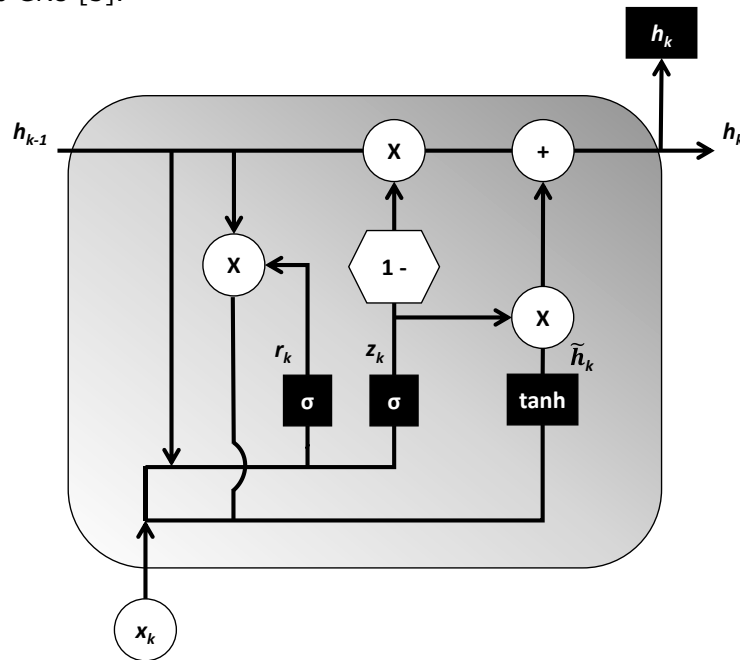


Fig. 25. Structura internă și căile de date din cadrul celulei GRU [3]

Ieșirea nivelului ascuns  $h$ , la instanța de timp  $k$ , depinde de ieșirea de la momentul  $k - 1$  și de o valoare de ieșire candidat, notată cu  $\tilde{h}_k$ , utilizând relația liniară:

$$h_k = (1 - z_k)h_{k-1} + z_k \tilde{h}_k \quad (14)$$

în care  $z_k$  este valoarea la poartă de actualizare, definită prin ecuația de activare:

$$z_k = \sigma(W_z[h_{k-1} \ x_k]) \quad (15)$$

Valoarea  $z_k$  controlează cât de mult ieșirea curentă depinde de ieșirea anterioară sau de ieșirea candidată. Aceasta se definește prin următoarea ecuație:

$$\tilde{h}_k = \tanh(W_h[r_k h_{k-1} \ x_k]) \quad (16)$$

unde  $r_k$  este valoarea la poarta de ieșire:

$$r_k = \sigma(W_r[h_{k-1} \ x_k]) \quad (17)$$

Poarta de resetare controlează cât de mult ieșirea candidată depinde de ieșirea anterioară.  $\mathbf{W}_z$ ,  $\mathbf{W}_h$ ,  $\mathbf{W}_r$  sunt matricile ponderilor pentru poarta de actualizare, pentru cea de resetare, respectiv pentru ieșirea candidat.

În [36] sunt descrise ecuațiile celulelor LSTM și GRU în variantă vectorizată. Aceste ecuații sunt cele mai des întâlnite în implementările de sisteme cadru pentru definirea și modelarea rețelelor neuronale. Aceste ecuații pot fi paralelizate în scopul de a optimiza timpul de învățare, respectiv timpul de rulare.

Astfel, pentru rețelele de tip LSTM avem următorul sistem de ecuații:

$$\begin{cases} c_t = f_t \otimes c_{t-1} + i_t \otimes \tilde{c}_t \\ \tilde{c}_t = g(W_c x_t + U_c h_{t-1} + b_c) \\ h_t = o_t \otimes g(c_t) \\ i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \\ f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \\ o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \end{cases} \quad (18)$$

iar pentru rețelele neuronale de tip GRU, acest sistem de ecuații devine:

$$\begin{cases} h_t = (1 - z_t) \otimes h_{t-1} + z_t \otimes \tilde{h}_t \\ \tilde{h}_t = g(W_h x_t + U_h (r_t \otimes h_{t-1} + b_h)) \\ z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \\ r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \end{cases} \quad (19)$$



### 3. ABORDĂRI CURENTE PRIVIND METODELE PENTRU EVALUAREA STĂRII DE DEGRADARE A BATERIILOR

#### 3.1. Discuție asupra metodelor existente de estimare a stării de degradare a bateriilor

În literatură există o preocupare activă în ceea ce privește problema estimării stării de sănătate a bateriilor dovedit de numărul mare de articole ce tratează probleme din acest domeniu. În figura 16 este prezentată o clasificare a metodelor de estimare a stării de sănătate a bateriilor conform [37]-[46].

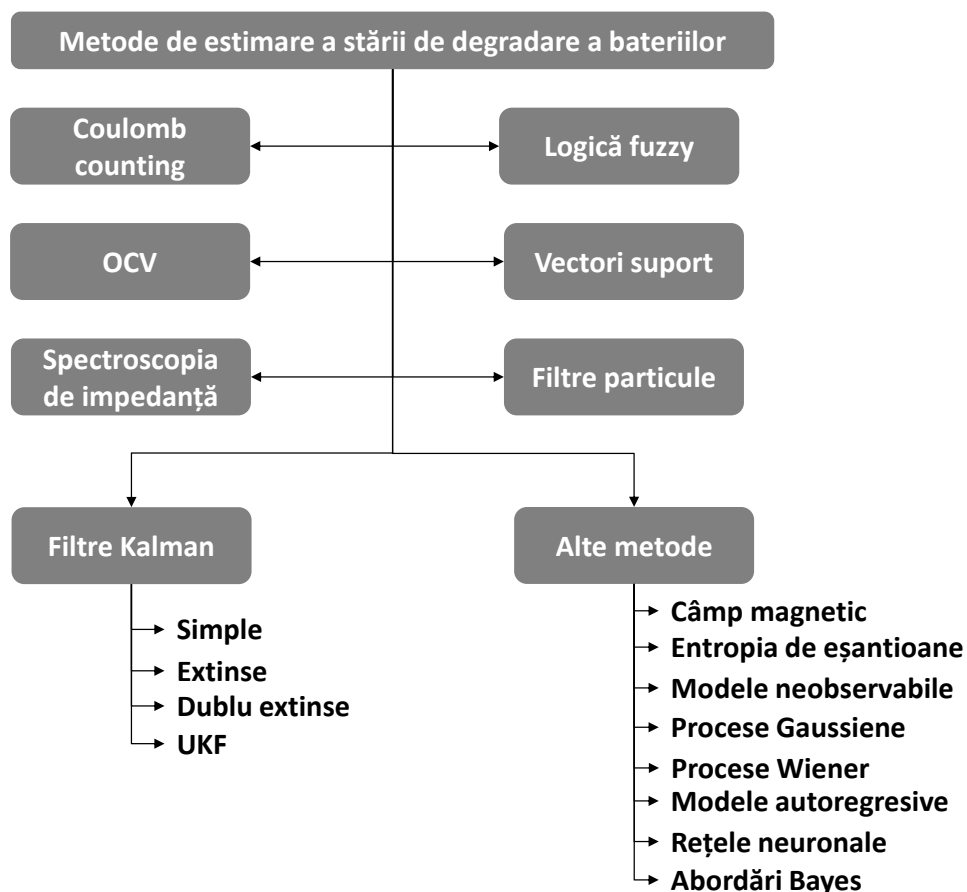


Fig. 36. Clasificarea metodelor de estimare a stării de degradare a bateriilor

### 3.1.1. Coulomb Counting

Este una dintre cele mai comune metode pentru estimarea SOH, estimare ce se desfășoară în doi pași. În primul pas se calculează capacitatea de descărcare a bateriei, prin integrarea curentului de descărcare în timp. În pasul al doilea, se calculează valoarea SOH prin raportarea capacității de descărcare la valoarea capacității nominale a bateriei

$$\begin{cases} Q_{discharge} = \int_0^T I(t)dt \\ SoH [\%] = \frac{Q_{discharge}}{Q_{rated}} \times 100\% \end{cases} \quad (20)$$

Principalii parametri monitorizați ai bateriei sunt: curentul de încărcare/descărcare, tensiunea bateriei, capacitatea bateriei, temperatura bateriei. Metoda poate fi transpusă adaptiv prin calcularea celor doi parametri din ecuație în timpul ciclurilor de încărcare/descărcare.

În [47] autorii au observat că valoarea capacității de descărcare scade o dată cu creșterea ciclului de încărcare/descărcare. În consecință funcția SOH are un trend descrescător, pentru o valoare de sub 80%, bateria se consideră inutilizabilă. Acuratețea metodei depinde foarte mult de instrumentele de măsură care este indicat să fie calibrate periodic pentru a nu introduce erori suplimentare.

Tot în [47] autorii susțin că acuratețea depinde doar de acuratețea cu care se măsoară curentul și capacitatea inițială a bateriei. Se introduce o nouă metrică pentru caracterizarea ciclului de descărcare, și anume nivelul de descărcare (DoD):

$$D_{OD} = \frac{Q_{released}}{Q_{rated}} \times 100\% \quad (21)$$

unde  $Q_{released}$  este capacitatea descărcată din baterie,  $Q_{rated}$  capacitatea nominală a acesteia. În orice moment de timp,  $D_{OD}$  se poate exprima:

$$\begin{cases} D_{OD}(t) = D_{OD}(t) + \eta \Delta D_{OD}(t) \\ \Delta D_{OD}(t) = \frac{-\int_{t_0}^{t_0+\tau} I_b(t)dt}{Q_{rated}} \times 100\% \end{cases} \quad (22)$$

în care  $I_b$  este curentul bateriei (încărcare/descărcare), iar  $\eta$  este factorul de eficiență.

SOH este definit ca valoarea DoD când bateria este degradată. În fiecare ciclu de încărcare/descărcare, valoarea DoD este compensată cu valoarea eficienței la încărcare/descărcare. În paralel, o recalibrare se face la fiecare stare terminală de încărcare/descărcare. Printre avantajele acestei metode se pot aminti costul redus al echipamentelor, putere de procesare redusă.

În [48] este prezentat un sistem de management al bateriilor complet, în care SOC și SOH sunt calculate bazat de coulomb counting. Precizia este comparată cu o versiune modificată a metodei OCV combinată cu filtru Kalman pentru

estimarea SOC. Erorile sunt calculate la diferiți curenți de încărcare/descărcare și se situează în intervalul [-3%, +3%].

### 3.1.2. OCV

Această metodă se bazează pe ideea definirii SOH în funcție de tensiunea în gol a bateriei. Modelul simplificat din [9] definește  $U_{OCV}$  ca în figura de mai jos:

$$U_{OCV} = U + I \cdot R \quad (23)$$

în care  $U_{ocv}$  este tensiunea în gol a bateriei,  $I$  este curentul bateriei,  $R$  este rezistența internă a bateriei și  $U$  este tensiunea la borne.

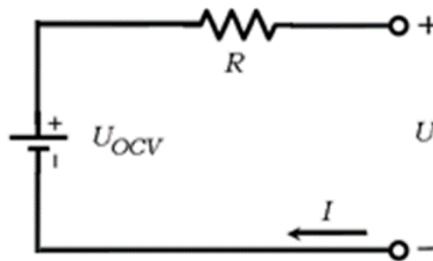


Fig. 4. Modelul electric simplificat

Se pot distinge două variante: metoda în timp real – online – în care predicția se face bazat pe măsurători în timp real al parametrilor bateriei; metoda a priori – offline – se bazează pe o analiză a curbelor OCV-SOH, din care reiese relația dintre OCV și SOH.

Așa cum este prezentat în [21] și [22], autorii au realizat un set extins de teste pentru a obține curbele OCV-SOH.

În [9], autorii s-au concentrat pe analiza influenței temperaturii în construcția curbelor OCV-SOC. Influența temperaturii nu este de neglijat atunci când este nevoie de o precizie cât mai mare. Autorii prezintă o metodă de a determina curbele OCV-SOC în relație cu temperatura. Curbele OCV-SOC se pot transforma cu ușurință pentru a fi utilizate la determinarea SOH. În ciuda unei precizii foarte bune de estimare, acest tip de metodă implică multe teste de laborator.

Varianta în timp real – online – se bazează pe OCV ca și parametru al modelului electric prezentat anterior, determinat prin identificarea sistemelor. Erorile de estimare sunt influențate de SoC inițial al bateriei și de temperatură. Dacă diferența dintre SoC inițial real și cel utilizat în calcule este menținută cât mai redusă, eroarea de estimare este de asemenea mică. O temperatură scăzută introduce o eroare medie pătratică (RMSE) în intervalul 5% - 25%.

Metoda prezentată în [6], este o combinație între parametri electrici pentru estimarea SOC și SOH. Algoritmul este destinat bateriilor cu mai multe celule și este bazat pe măsurarea puterii electrice. Eroarea de estimare depinde doar de perioada pentru care se face predicția, când supratensiunea ce apare la fenomenul de difuzie a bateriei poate influența SOC.

În [4] autorii prezintă câteva dezavantaje al estimării SoH prin intermediul parametrului de tensiune în gol. Aceștia susțin că în condiții reale este foarte de greu de determinat acest parametru, propunând o metodă simplificată în care OCV este eliminat ca parametru, iar estimarea se face bazându-se pe curbele CC-CV (curent constant – tensiune constantă). Această modificare duce la simplificarea algoritmului, reduce puterea de procesare necesară estimării precum și reducerea costurilor hardware.

Autorii din [37] consideră că OCV este foarte eficientă dacă se folosește în combinație cu metoda prezentată anterior: coulomb counting. În [22], autorii folosesc parametri exponențiali în combinație cu OCV pentru estimarea SoH cu ajutorul capacității incrementale. Această combinație produce o eroare de estimare ce se apropie de 1%.

### 3.1.3. Spectroscopia de impedanță

Metoda spectroscopiei impedanței (EIS) permite determinarea cu o precizie foarte mare a impedanței interne a bateriei utilizând un spectru larg de frecvențe la un curent mic al bateriei. În literatură de specialitate este demonstrat că măsurătorile EIS sunt un indicator foarte bun pentru SoC, din care se poate estima SOH.

În figura următoare este prezentat impedanța bateriei în domeniul complex:

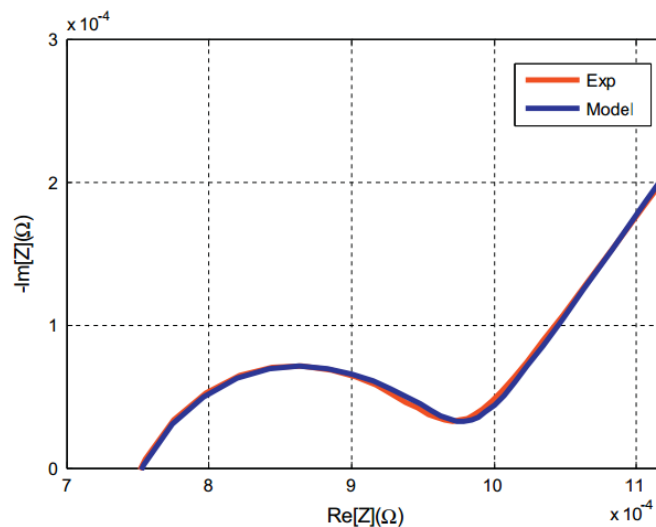


Fig. 58. Diagrama Niquist obținută prin EIS [16][18]

Una dintre cele mai comune utilizări ale EIS este identificarea parametrilor unui model electrochimic. Acest lucru este detaliat în [12], aplicat celulelor de tip Li-Ion. Modelul are 16 parametri din care 6 sunt calculați a-priori și considerați constanți, ceilalți 10 parametri sunt determinați cu ajutorul HMPISO, o metodă de analiză bazată pe filtre de particule.

Măsurătorile EIS sunt făcute pe baterii de timp Li-Ion încărcate la 100% și 50%, folosind un spectru de frecvență de 0.025Hz – 4kHz. Modelul identificat este comparat cu modelul Randles, demonstrându-se o precizie de calcul al parametrilor de 8 ori mai mare. În articol nu este prezentată o metodă completă de estimare a SOH, dar sunt sugerate două modele ce ar putea fi folosite în acest sens.

În [16], autorii prezintă o fuziune între metoda EIS și RNN, în care RNN este folosit exclusiv pentru predicția SoH și RUL a bateriilor. EIS este folosită pentru o analiză offline, pentru calcularea parametrilor circuitului echivalent. Modelul electrochimic propus conține 5 parametri. Doi dintre aceștia sunt folosiți pentru a modela rezistența ohmică  $R_1$  și cea dependentă de SOC,  $R_2$ . Modelul conține și doi parametri ce sunt folosiți la modelarea difuziei: CPE1 și CPE2, care sunt dependenți de SOC. Măsurătorile EIS sunt determinate la 25 grade C, iar spectrul de frecvență folosit este [0,01Hz – 10kHz]. Modelul este validat pentru vehicule electrice hibride, demonstrându-se totodată că este destul de simplu pentru a fi implementat pentru orice sistem de management al bateriei.

Predicția se face folosit RNN având ca intrare măsurătorile RNN. Eroarea medie rezultată de este de 2.1%.

O abordare nouă folosind EIS este prezentată în [20]. Autorii propun o metodă de estimare a RUL bazată pe metoda degradării calendaristice a celulelor Li-Ion. Experimentele au fost realizate la 3 temperaturi diferite și 3 SoC diferite, rezultatele măsurătorilor EIS au demonstrat dependența SOH de temperatură și de SoC. Analizând curbele EIS, autorii au observat că impedanța reală la 0,1 Hz variază exponențial cu SoH, și poate fi folosită pentru predicția acestuia. În figura următoare se prezintă această dependență exponențială:

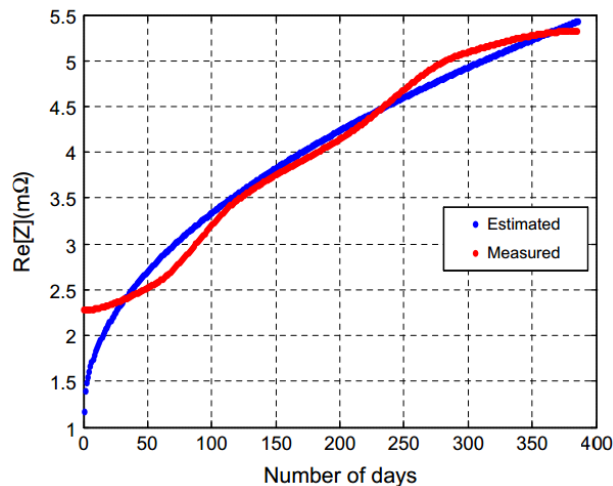


Fig. 19. Impedanța estimată și cea măsurată partea reală [16]

Rezultatele prezentate sugerează că acest parametru identificat prin măsurători EIS poate fi folosit cu succes pentru predicția SoH, modelul sugerat fiind unul mult mai simplu decât cel electrochimic.

### 3.1.4. Filtre Kalman

Filtrele Kalman reprezintă o metodă performantă de estimare a stărilor unui proces. Este compus din două părți: partea de predicție și cea de actualizare. Starea procesului este estimată în faza de predicție. În faza de actualizare, filtrul obține o reacție din măsurători (considerate afectate de zgomot de tip Gaussian) [49].

Ecuatiile standard al unui filtru Kalman sunt următoarele:

$$\begin{cases} \hat{x}(t|t-1) = F_t \hat{x}(t-1|t-1) + B_t u_t \\ P(t|t-1) = F_t P(t-1|t-1) + Q_t \\ \hat{x}(t|t) = \hat{x}(t|t-1) + K_t (y_t - H_t \hat{x}(t|t-1)) \\ K_t = P(t|t-1) H_t^T (H_t P(t|t-1) H_t^T + R_t)^{-1} \\ P(t|t) = P(t|t-1) - K_t H_t P(t|t-1) \end{cases} \quad (24)$$

în care:  $\hat{x}$  este starea estimată,  $F$  este matricea de tranziție a stărilor,  $B$  matricea de control,  $u$  este vectorul de control,  $P$  este matricea de varianță a stărilor,  $Q$  este matricea de varianță a procesului,  $y$  este vectorul de măsurare,  $H$  este matricea de măsurare,  $K$  este amplificarea Kalman și  $R$  este matricea de varianță a măsurătorilor.

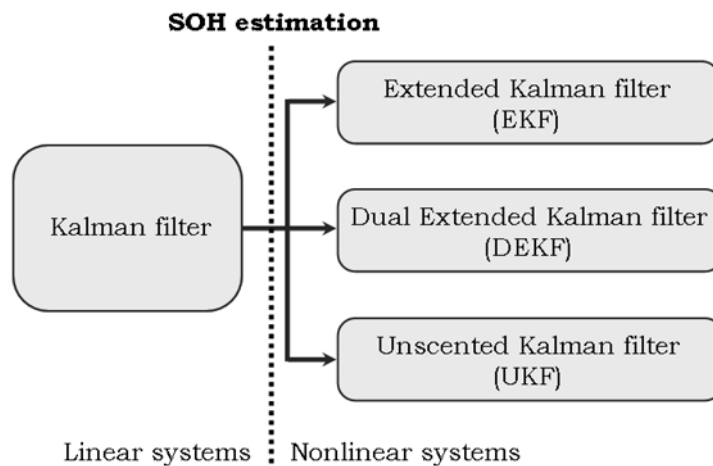


Fig.20. Variante de filtre Kalman utilizare în sisteme neliniare

Filtrele Kalman sunt utilizare de obicei în sistem liniare. Pentru sistemele neliniare, cum e cel pentru estimarea SOH a bateriei, există în literatură câteva versiuni modificate.

Filtrul Kalman extins este folosit pentru predicția stărilor unui proces neliniare, în care ecuațiile de tranziție și de măsurare sunt neliniare, dar trebuie să

fie diferențiabile. Pentru estimare SOH, filtrul Kalman dual extins este format dintr-un filtru Kalman pentru predicția SOC și celălalt pentru predicția capacității bateriei. Filtrul Kalman U cunoscut și sub forma de filtru Kalman bazat pe puncte sigma elimină câteva dezavantaje ale filtrelor Kalman extinse și îmbunătățește performanța. În acest caz, funcția de distribuție a probabilității este estimată prin câteva puncte sigma, care reprezintă un subset de puncte calculate în așa fel să fie apropiate de valoarea mediană. Transformarea neliniară acestor puncte vor genera estimarea covarianței mediane.

În [50] și [51] autorii folosesc un model Randles de ordin 2 pentru a obține ecuațiile de stare a filtrului Kalman extins. În primul caz, SOH este calculat bazat pe SOC, în al doilea caz, SOH este calculat bazat pe impedența internă a bateriei. Ecuațiile modelului sunt:

$$\begin{cases} \begin{pmatrix} SoC_k \\ V_{1,k} \\ V_{2,k} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 - \frac{\Delta t}{R_1 C_1} & 0 \\ 0 & 0 & 1 - \frac{\Delta t}{R_2 C_2} \end{pmatrix} \begin{pmatrix} SoC_{k-1} \\ V_{1,k-1} \\ V_{2,k-1} \end{pmatrix} + \begin{pmatrix} \frac{\Delta t}{C_{bat}} \\ \frac{\Delta t}{C_1} \\ \frac{\Delta t}{C_1} \end{pmatrix} \\ V_{bat,k} = V_{OC}(SoC_k) + V_{1,k} + V_{2,k} + R_0 I_{bat,k} \end{cases} \quad (25)$$

respectiv,

$$\begin{cases} \begin{pmatrix} V_{1,k+1} \\ V_{2,k+1} \\ R_{0,k+1} \end{pmatrix} = \begin{pmatrix} e^{-\frac{T}{R_1 C_1}} & 0 & 0 \\ 0 & e^{-\frac{T}{R_2 C_2}} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} V_{1,k} \\ V_{2,k} \\ R_{0,k} \end{pmatrix} + \begin{pmatrix} R_1 \left( 1 - e^{-\frac{T}{R_1 C_1}} \right) \\ R_2 \left( 1 - e^{-\frac{T}{R_2 C_2}} \right) \\ 0 \end{pmatrix} I_{bat,k} \\ (V_{cell,k} - V_{OC,k}) = (1 \ 1 \ 0)x + R_{0,k} I_{bat,k} \end{cases} \quad (26)$$

în care  $SoC_k$  este SOC la momentul  $k$ ,  $R_0$ ,  $R_1$ ,  $R_2$ ,  $C_1$ , și  $C_2$  sunt parametri modelului electric Randles de ordin 2 care modelează impedența bateriei,  $V_{1,k}$  și  $V_{2,k}$  sunt componente ale tensiunilor modelului Randle la momentul  $k$ ,  $V_{OC}$  este tensiunea în gol a bateriei,  $I_{bat,k}$  este curentul bateriei,  $V_{bat}$  este tensiunea la borne a baterie,  $V_{cell,k}$  este tensiunea în gol per celulă, și  $x = (V_{1,k} \ V_{2,k} \ R_0 \ k)^T$  variabila de stare. În ambele cazuri, eroare de estimare este sub 4%.

În [52], filtrul Kalman extins este folosit pentru a modela parametri unei baterii Li-ion cum ar fi: tensiunea la borne, tensiunea în gol și SOC. Autorii se bazează pe baterii folosite la mașini electrice. Rezultatele experimentale arată o eroare de estimare de aprox. 1%.

Filtrul Kalman dual extins este prezentat în [53] și [54]. Două filtre Kalman sunt folosite: unul pentru estimarea SOC, celălalt pentru estimarea altor parametri ai bateriei, cum ar fi capacitatea. Figura de mai jos prezintă structura unui filtru Kalman dual extins:

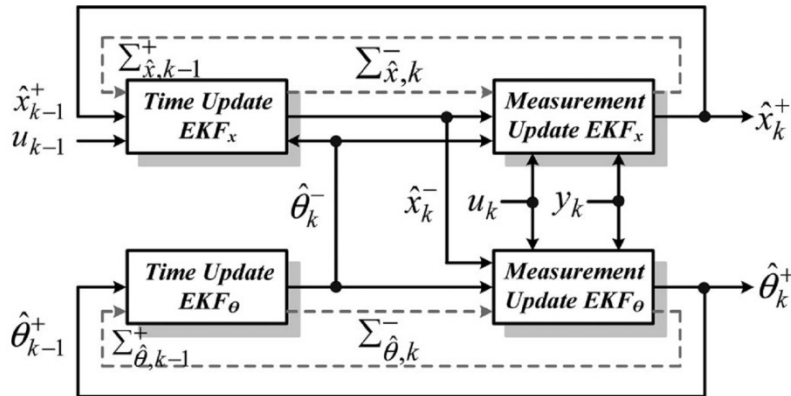


Fig.21. Arhitectura Filtrului Kalman Dual

Filtrul este bazat pe un model electric Randles de ordin 2. Autorii susțin că o dată cu creșterea ordinului modelului electric, estimarea devine greu de realizat datorită complexității în aplicații real-time. În primul articol, erorile de estimare se situează între  $\pm 5\%$ . Eroarea de estimare în cel de-al doilea articol se situează în jurul valorii de 2%.

Filtrul Kalman U este prezentat în [55] și [56] pentru estimarea SoH. Modelul electric al bateriei utilizat este Randles de ordin 2. Acest tip de filtru a fost introdus pentru a aborda câteva probleme pe care le prezintă filtrele Kalman extinse [57]. Printre probleme se numără probabilitatea destul de mare ca filtrul să fie divergent dacă starea inițială nu este specificată în mod corect, rezultând un model eronat al procesului. O altă problemă constituie aproximarea mediei posterioare și a matricei de covarianță care pot să ducă la erori semnificative.

În [55], autorii compară performanța estimării SOH utilizând filtre Kalman U cu cea a coulomb counting. Filtrul Kalman U se adaptează perfect la neliniaritățile bateriei și rezolvă problemele celei de-a doua metode (acumulările de erori), rezultând o estimare a SoH aproape de cea reală. Eroare de estimare se situează între 2.2 și 2.9%.

În [56], filtrul Kalman U este folosit în combinație cu algoritmul SVR. SVR este folosit pentru a determina valoarea capacității inițiale ce se introduce la intrarea filtrului Kalman U. Acest lucru este făcut offline, bazat pe analiza curbelor de îmbătrânire. Performanța algoritmului este sub 1%.

În [58], filtrul Kalman U este aplicat modelului electrochimic. Autorii monitorizează răspunsul tensiunii bateriei la pulsuri de curent și rezistența internă a acesteia. Metoda este adaptivă, cu un timp de calcul inițial de 40 min, până când valorile încep să convergeze la valorile reale a parametrilor.

Autorii din [59] propun o metodă de estimare a SOH obținut prin combinația filtrului Kalman extins și metoda analizei cvadractice. Ecuațiile filtrului Kalman sunt



obținute dintr-un model electric de ordin unu. Parametri bateriei obținuți prin EKF, sunt folosiți în clasificarea: baterii nefolosite, puțin folosite, foarte folosite. Eroarea de clasificare atinge un maxim de 8.3%.

În [60] estimarea SoH folosește un filtru EKF bazat pe capacitatea și rezistența de difuzie. Metoda propusă este validată prin multiple teste de laborator prin măsurarea offline a parametrilor bateriei și ridicarea curbilor de îmbătrânire. Eroarea de estimare este de  $\pm 5\%$ .

În [61], autorii folosesc un filtru Kalman bazat pe diferență centrală (CDKF). Acesta din urmă estimează mai precis neliniaritățile bateriei. CDKF este o variantă de SPKF adaptat pentru utilizarea în implementări practice datorită numărului redus de parametri.

### 3.1.5. Logică Fuzzy

Logica fuzzy este un concept matematic care generalizează logica și mulțimile booleene. ideea de bază este că se permite elementelor din mulțimile fuzzy să aibă grade de apartenență la acea mulțime. O funcție de apartenență este aplicată fiecărui element  $x$  dintr-o mulțime  $A$ , iar valoarea funcției este situată în intervalul  $[0, 1]$ , ceea ce reprezintă cât de mult aparține  $x$  la mulțimea  $A$ . Logica fuzzy se bazează pe operatorii booleeni ȘI, SAU și NEGAT, definiți în tabelul de mai jos:

**Tabel 1.** Operatori de logica fuzzy

Operație	Rezultat
$x$ ȘI $y$	$\min(x, y)$
$x$ SAU $y$	$\max(x, y)$
$x$ NEGAT	$1 - x$

Logica fuzzy se poate aplica și în cazul estimării SOH, dacă se definesc câteva funcții de apartenență pentru SoH, cum ar fi *stare bună*, *stare acceptabilă*, *stare degradată*.

Unul dintre articolele care folosesc această metodă de estimare este [62]. Autorii propun un model derivat din cel electrochimic, iar estimarea se face pentru SOC, SOH se face folosind trei metode: ARMA, rețele neuronale și logică fuzzy. Aceste trei metode sunt comparate, metoda bazată pe logică fuzzy prezentând o eroare apropiată de celelalte metode.

În [63], autorii realizează un set de teste pe un număr de 95 de baterii Li-Co, încărcându-le la curenți constant de 0.5 C și descărcându-le cu diferiți curenți de descărcare. Autorii folosesc 3 parametri pentru a genera mulțimile fuzzy iar funcțiile de apartenență sunt de forma distribuției normale. Erorile de estimare variază între 1.4% și 9.2% dovedind-se că logica fuzzy poate fi o metodă ce se pretează pentru estimarea SOH.

O altă abordare bazată pe logică fuzzy este prezentată în [64]. Rezistența în curenți continuu este folosită ca parametru, fiind măsurată în faza de încărcare. Autorii demonstrează existența unei relații între rezistență și capacitatea bateriei. După calculul rezistenței, se definesc 4 funcții de apartenență care delimitează domeniul intrărilor. Funcțiile de apartenență definesc 4 limite: baterie sănătoasă,

baterie acceptabilă, baterie slabă și baterie degradată. Valoarea actuală a capacității de face prin intermediul relației:

$$C_A = \frac{\sum_{i=1}^4 \mu_i O_i}{\sum_{i=1}^4 \mu_i} \quad (27)$$

în care  $C_A$  este capacitatea actuală estimate,  $\mu_i$  este puterea de activare (grad de îndeplinire) al regulii  $i$ , iar  $O_i$  este puterea de activare a regulii de ieșire. Este calculată capacitatea actuală a mai multă baterii având SoH în intervalul 100% - 20% folosind metoda propusă. Eroarea de estimare se situează sub 5%. Autorii arată că metoda propusă este mult mai bună decât regresia liniară a cărei eroare de estimare ajunge și la 10%.

În [65] și [79] autorii pornesc de la un model dublu exponențial de forma:

$$y_{fit} = a_0 + a_1 e^{-\left(\frac{x}{a_1}\right)^{\beta_1}} + a_2 e^{-\left(\frac{x}{a_2}\right)^{\beta_2}} \quad (28)$$

Parametrii funcției sunt estimați folosind logica fuzzy, generând erori de estimare în intervalul 5%-10%.

### 3.1.6. Vectori suport

Constituie una dintre cele mai populare metode de estimate pentru SOH a bateriei în varianta neliniară.

Noțiunile de bază a SVM sunt prezentate în detaliu în [55] și [66]. Având un set de date de antrenare, notat prin  $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \subset R^n \times R$  - în care  $R^n$  este spațiul datelor de intrare  $x_i$ , iar datele țintă sunt  $y_i$ , scopul este să se determine o funcție  $f(x)$  care să aproximeze datele țintă cu o deviație maximă  $\epsilon$ , pentru toate datele de antrenare. Această funcție are următoarea formă:

$$f(x) = \langle w, x \rangle + b \quad (29)$$

cu  $w \in R^n$ ,  $b \in R$  și  $\langle \cdot, \cdot \rangle$  fiind produsul scalar în  $R^n$ . Există însă situații când această funcție nu există. Pentru a rezolva această problemă se introduc variabilele suplimentare  $\xi$  și  $\xi$ , care odată introduse, oferă posibilitatea definirii unui interval pentru eroarea de măsurare. În acest caz, problema se rezumă la cea de optimizare duală.

Problema optimizării duale se rezolvă folosind tehnici de programare pătratică care presupun aplicarea unei transformate neliniare intrărilor, transformată ce translatează intrările într-un spațiu multidimensional. În acest caz, funcția devine:

$$f(x) = \sum_{k=1}^K (a_k^* - a_k) K(x(k), x) + b \quad (30)$$

în care  $a_k^*, a_k$  sunt variabilele duale obținute prin aplicarea Lagrangianului relativ la  $(w, b, \xi, \xi^*)$ , iar  $K$  este funcția nucleu (funcție de mapare) având forma:

$$K(x(k), x(m)) = \langle \Phi(x(k)), \Phi(x(m)) \rangle \quad (31)$$

în care  $\Phi$  reprezintă transformata neliniară.

Algoritmul RVM [68], [69], reprezintă forma Bayesiană a SVM și diferă de acesta prin introducerea unei interpretări probabilistice ale ieșirilor. Predicția ieșirilor are următoarea formă:

$$t = y(x) + \varepsilon_n \quad (32)$$

în care  $t$  este ținta pentru predicție,  $y(x)$  este funcția ce translatează datele de intrare în cele țintă (de ieșire) conform SVM, iar  $\varepsilon_n$  exprimă zgomotul procesului.

După aplicarea legii lui Bayes, rezultă o funcție de distribuție posterioară:

$$p(w|t, a, \sigma^2) = \frac{p(t|w, \sigma^2)p(w, a)}{p(t|a, \sigma^2)} \quad (33)$$

în care  $w$  este vectorul ponderilor,  $a$  este vectorul hiperparametrilor folosit pentru controlul deviației ponderilor,  $\sigma$  este varianta, și  $p$  este funcția de distribuție posterioară.

Noile predicții se calculează prin integrarea ponderilor și calcularea probabilității marginalia ale hiperparametrilor:

$$p(t|a, \sigma^2) = \int p(t|w, \sigma^2)p(w|a)dw \quad (34)$$

$$= (2\pi)^{N/2} |B^{-1} + \Phi A^{-1} \Phi^T|^{-1/2} \times \exp\left(-\frac{1}{2} t^T (B^{-1} + \Phi A^{-1} \Phi^T)^{-1} t\right)$$

în care  $\mathbf{A} = \text{diag}(a_1, \dots, a_{N+1})$ , și  $\mathbf{B} = \sigma^2 \mathbf{I}$ , iar  $\Phi$  este matricea funcției nucleu, obținută printr-un set de date de antrenare.

Cele două formule de estimare pot fi aplicate seturilor de date de antrenare care conțin parametrii ai bateriei cum ar fi capacitatea.

Algoritmi de tip SVM sunt prezentați în [56], [66]-[72]. Ca o caracteristică generală, acești algoritmi rulează pe seturi extensive de date de antrenare obținute prin ciclarea bateriilor. Variante online cu predicții în timp real solicită resurse computaționale ridicate și de cele mai multe ori sunt prezentate ca simulări în Matlab.

În [56], seturile de date pentru construcția vectorilor sunt obținute prin teste îndelungate de îmbătrânire a bateriei. SVM este comparat cu UKF, prezentând erori de predicție asemănătoare. Dacă se cunoaște capacitatea inițială a bateriei, eroarea de predicție este de 1%. Fără această informație, eroarea poate crește la 20%.

În [69], autorii aplică SVM pe seturi de date rezultate din spectroscopii ale impedanței bateriei. Modelul estimează degradarea capacității bateriei precum și modificarea rezistenței interne ale acesteia, care sunt intrări pentru calcularea SOH și RUL. SVM prezintă o performanță mai mare decât modelul Markov cu stări invizibile, cu o eroare de estimare maximă de 2%.

În [70], algoritmi SVM sunt aplicați pe seturi de date de baterii puse la dispoziție de NASA. Două variante sunt prezentate: online – în timpul ciclurilor de încărcare/descărcare și cea combinată cu analize offline. Erorile medii absolute de estimare este 0.02 în cazul variantei combinate și 0.03 în cazul variantei online.

O variantă a SVM aplicată datelor obținute din teste de durabilitate la bateriile vehiculelor hibride este prezentată în [71]. Vectorii sunt corelați cu valoarea curentului de descărcare pentru a spori performanța algoritmilor.

În [72] SVM este aplicat pentru aproximarea curbelor capacității bateriei obținute prin metoda capacității incrementale. Autorii arată cum se poate aplica această metodă pentru baterii de tip litiu-polimer. Metoda se bazează pe actualizarea parametrilor în funcție de condițiile curente ale bateriei. Eroarea de predicție se situează în jurul de 1%.

Autorii din [67] și [68] folosesc RVM cu setul de date de intrare oferite de NASA. Atenția este îndreptată pe comparația cu SVM, dovedindu-se erori mai mici pentru RVM, sub 0.6%. Cel din urmă articol abordează problema degradării bateriilor în vehicule electrice. RVM este aplicat pentru RUL obținându-se o eroare maximă de 9 cicluri încărcare/descărcare.

### 3.1.7. Filtre particule

Această metodă se potrivește sistemelor neliniare în care parametrii modelelor bateriei sunt necunoscuți sau aceștia se schimbă în timp. Metoda are ca rezultat funcția densitate de probabilitate bazată pe un set de puncte ce aparțin spațiului stărilor.

În [73], autorii folosesc un model electrochimic simplu pentru baterii Li-Ion pentru care parametrii sunt identificați prin RVM. Valorile obținute în pasul anterior sunt folosite ca date inițiale pentru algoritmul bazat pe particule. Particulele sunt recalulate ulterior la fiecare iterație.

În fig. 22 sunt prezentate 2 rezultate tipice pentru filtrele bazate pe particule, funcțiile densitate probabilistică pentru RUL. Se poate observa că unul dintre avantajele folosirii particulelor este că generează o distribuție probabilistică în timp, și nu se bazează doar pe un punct/dată. Acest lucru are ca și consecință reducerea incertitudinilor la nivel de parametri ai modelului precum și cel al măsurătorilor.

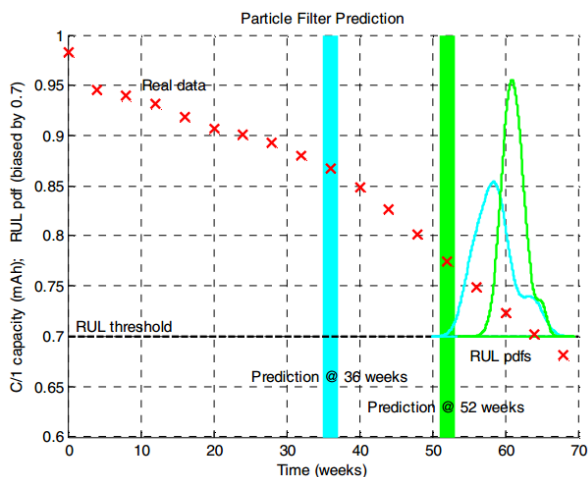


Fig. 22. Rezultatele predicției folosind filtru de particule [73]

În [74], filtrele bazate pe particule sunt folosite pentru estimarea SOC și SOH al bateriilor de tip Li-fosfat (LiFePO<sub>4</sub>). Autorii folosesc o metodă bazată pur pe datele de intrare fără a folosi un model electric al bateriei, parametrii folosiți fiind doar OCV și curentul de încărcare/descărcare. Pasul de reeșantionare al PF, denumit și reeșantionare cu varianță redusă, constă în următoarea regulă: când se generează un nou set de eșantioane, eșantioanele vechi cu ponderi mari sunt înmulțite, pe când cele vechi cu ponderi scăzute sunt eliminate din calcul.

Autorii au validat soluția în două domenii: rețele fotovoltaice pentru consum casnic și vehicule electrice. Rezultatele nu sunt prezentate în cel mai mare amănunt atenția fiind îndreptată spre eroarea medie absolută care este în jurul a 2%.

Autorii din [75] demonstrează faptul că FP este mult mai precis decât metodele bazate pe regresie statistică, comparând erorile de predicție și dimensiunea funcției de distribuție probabilistică. În plus, este prezentat și un avantaj al FP și anume că timpul până la generarea primei predicții este mult mai scurt comparativ cu celelalte metode.

Aceiași autori prezintă în [76] o îmbunătățire a algoritmilor bazați pe FP. Ei au observat că atunci când spațiul stărilor modelului este definit prin vectori de mari dimensiuni, procesul de reeșantionare nu poate reduce varianta erorilor particulelor. Ei propun să înlocuiască acest spațiu de stări cu cel calculat bazat pe metoda Rao-Blackwellized (FPRB). Se demonstrează că FPRB prezintă erori mult mai mici iar funcția de distribuție este mult mai compactă rezultând o predicție mult mai bună.

În [77], autorii propun folosirea a două modele analitice ca intrări pentru FP. Mai precis, aceștia folosesc FP pentru a estima parametri modelului matematic polinomial și cel exponențial. Această soluție poate fi implementată într-un sistem de management al bateriilor cu predicție în timp-real la fiecare ciclu de încărcare/descărcare. O îmbunătățire a acestei soluții este prezentată în [78], în care autorii introduc o nouă abordare în modelarea capacității bateriei. Este o combinație dintre cele două modele anterioare, rezultând o mai bună precizie de estimare.

Filtrele bazate pe particule pot fi folosite ca un mecanism auxiliar pentru predicția RUL, așa cum este prezentat în [79]. Autorii prezintă o metodă complexă care se bazează pe modelul Verhulst. Pentru optimizare se folosește o nouă metodă, prin care se optimizează șiruri de particule. FP este folosit pentru a compensa erorile de predicție prin ajustarea modelului Verhulst.

O soluție originală este prezentată de autori în [79] și [80]. Autorii folosesc un model matematic simplu cu doi termeni exponențiali, pentru a facilita și implementarea într-un sistem de management al bateriei real. Ei folosesc metoda Bayesiană Monte Carlo, similară cu FP, în care rezultatul este tot o funcție de distribuție al probabilității:

$$P(X_k|Q_{0:k}) \approx \sum_{i=1}^N \omega_k^i \delta(X_k - X_k^i) \quad (35)$$

în care  $X_k^i$  este un set independent de eșantioane din  $P$ ,  $\omega_k^i$  ponderea fiecărui eșantion;  $\delta(\cdot)$  funcția impuls Dirac. Reeșantionarea și recalcularea ponderilor se face într-un mod diferit de cel clasic FP. Pentru estimarea valorilor inițiale ale bateriei se folosește metoda Dempster-Shafer, în primii  $n$  cicli de încărcare/descărcare. BMC

este folosit pentru a actualiza parametrii modelului la fiecare ciclu și totodată actualizând SOC și RUL.

În [80] FP este prezentat în comparație cu EKF, scoțându-se în evidență eroarea de estimare a FP care este în jurul de 1.1%, față de EKF unde poate ajunge și la 8.6%. În [81], autorii compară eroarea de predicție a metodei BMC folosind două metode de calcul al parametrilor diferite: DST și metoda mediană. Eroarea de predicție este de 2.1% pentru prima metodă de calcul, respectiv 8.3% pentru cea din urmă.

În [82] autorii prezintă o modificare a FP, precizând și dezavantajele metodei clasice: degenerarea particulelor. Ei propun o soluție pentru a evita acest lucru și anume utilizarea unui filtru Kalman U pentru a genera distribuția particulelor, urmând să se folosească pașii normali de reeșantionare și calcul al ponderilor din FP. Sunt prezentate o serie de teste din care rezultă că metoda propusă asigură o eroare de estimare de sub 5% pe când soluția clasică atinge o valoare a erorii de 7%.

O altă variantă a FP este prezentată în [83] pentru estimarea SOH și RUL. Această metodă își propune să reducă dezavantajele celorlalte metode prezentate anterior: FP clasic, RBFP și FPU. Printre acestea se numără: fenomenul de împușinare a eșantioanelor, lipsa acurateții de determinare a funcției densității de probabilitate. Metoda propusă transformă particulele ce au fost valori ale ponderilor foarte mari, în valori ale ponderilor foarte mici în așa fel încât să urmeze cât mai fidel partea posterioară a densității de probabilitate. Figura 9. prezintă acest algoritm.

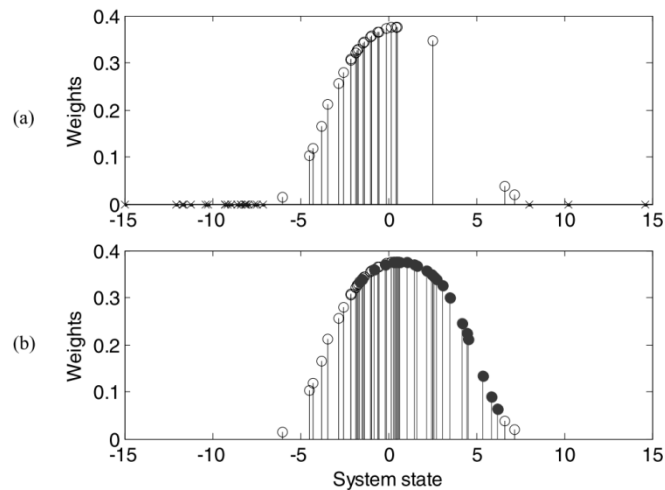


Fig.23. Transformare MPF a funcției de distribuție de probabilitate posterioară

În [84], [86], [87] sunt prezentate variații ale metodei FP implementate în sisteme încorporate și simulate în platforme dezvoltate în Matlab. Autorii se axează pe estimarea RUL precum și detecția și prevenirea erorilor de funcționare a algoritmilor.

### 3.1.8. Alte metode

O nouă metodă pentru estimarea SoH și RUL ce poate fi implementată în sisteme încorporate cu resurse limitate este prezentată în [5]. Estimarea SoH se face ținând cont de SoC iar metoda de predicție este una bazată pe regresie în sensul celor mai mici pătrate.

În [88] autorii folosesc câmpul magnetic pentru a determina dacă o baterie este degradată sau nu. Sunt folosite două bobine, prima pentru a genera câmpul magnetic, iar cea de-a doua pentru măsurarea variației câmpului magnetic datorat concentrației de protoni. Se demonstrează existența unei relații între SoH și variația câmpului magnetic măsurat o dată cu degradarea bateriei.

O combinație între estimarea bazată pe cele mai mici pătrate și cea prin entropia de eșantionare este prezentată în [89]. Autorii folosesc metoda entropiei de eșantionare pentru a modela capacitatea bateriilor Li-Ion folosind teste extensive de laborator. Se folosește o funcție polinomială de ordin 3 pentru a reprezenta corespondența în valoarea capacității bateriei și valoarea entropiei pentru fiecare temperatură.

În [90] este descrisă metoda aproximării entropiei. Aceasta este o variantă a metodei entropiei de eșantionare folosită pentru estimarea SoH la baterii de plumb. Această metodă constă în calcularea funcției  $ApEn(m, r, N)$  pentru un set de date  $x(i) = x(1), x(2), x(3), \dots, x(N)$ , cu  $N$ , reprezentând numărul total de puncte,  $m$  lungimea subsetului de puncte pentru care rulează algoritmul,  $r$  este toleranța – considerată  $0,2DS$  unde  $DS$  este deviația standard al setului de date:

$$DS = \sqrt{\frac{1}{N-1} \sum_{i=1}^N \left[ x(i) - \frac{1}{N} \sum_{i=1}^N x(i) \right]^2} \quad (36)$$

Funcția  $ApEn$  este calculată folosind relațiile:

$$\begin{aligned} ApEn(m, r, N) &= \varphi^m(r) - \varphi^{m+1}(r) \\ \varphi^m(r) &= \frac{1}{N - m + 1} \sum_{i=1}^N \ln \left( \frac{V^m(i)}{N - m + 1} \right) \\ V^m(i) &= \text{No.of } \bar{d}[X(i), X(j)] \leq r \end{aligned} \quad (37)$$

cu  $d$  fiind distanța între vectorii (diferența absolută între elementele scalare din vectori):

$$\begin{aligned} X(i) &= [x(i), x(i+1), \dots, x(i+m-1)], \quad i = 1, N - m + 1 \\ X(j) &= [x(j), x(j+1), \dots, x(j+m-1)], \quad j = 1, N - m + 1 \end{aligned} \quad (38)$$

Autorii din [91] folosesc un model cu stări neobservabile pentru estimarea SOC și SOH a bateriilor de tip Li-Ion. Setul de ecuații al modelului este determinat prin folosirea unui model electric complex alcătuit din două grupuri RC. În plus se prezintă un model pentru auto-descărcarea bateriei. Estimatorul are următoarea formă:

$$\xi = A\xi + BI_B + D\phi(V_{SOC}, I_B) \quad (39)$$

în care  $A$ ,  $B$  și  $D$  sunt matricile de stare și de ieșire,  $\xi$  este vectorul de stări, și  $\phi$  descrie neliniaritățile de estimare. Bazat pe ecuația stărilor se poate calcula rezistența internă  $R_0$  a bateriei și se folosește următoarea formulă pentru calcularea SoH:

$$SOH = \frac{R_{0,EOF} - R_0}{R_{0,EOF} - R_{0,NEW}} \times 100 \quad (40)$$

în care  $R_0$  este rezistența internă curentă a bateriei,  $R_{0,EOF}$  este rezistența internă a bateriei când bateria este considerată inutilizabilă, iar  $R_{0,NEW}$  este rezistența internă a bateriei când bateria este nouă.

O metodă bazată pe funcția densitate de probabilitate este prezentată în [90]. Această metodă este combinată cu analiza incrementală a capacității bateriei pentru a determina evoluția SOH. Se folosesc următoarele serii pentru a calcula funcția de densitate:

$$\begin{cases} V_{d,k} = V_0 + k \cdot \delta V, & k = 0, 1, 2, \dots \\ Q_{d,n} = Q_0 + n \cdot dQ = n \cdot I \cdot t, & n = 0, 1, 2, \dots \end{cases} \quad (41)$$

unde  $V_{d,k}$  este tensiunea de ieșire a bateriei,  $d$  convenția de notație pentru termenul digital,  $k$  numărul eșantionului, și  $Q_{d,k}$  este capacitatea bateriei obținută prin integrare și care corespunde cu  $V_{d,k}$ . Valorile seriilor sunt calculate urmând multiple teste de încărcare/descărcare.

O metodă derivată din cea prezentată anterior și care folosește două funcții de densitate de probabilitate este descrisă în [92], în cadrul unui model Gaussian adaptiv. Prima funcție densitate este folosită pentru caracterizarea seturilor de date din istoric, iar cea de-a doua pentru caracterizarea setului de date curent. Parametrul bateriei folosit pentru estimare este capacitatea bateriei.

În [94] autorii propun regresia funcțională gaussiană pentru estimarea SoH. Capacitatea bateriei este monitorizată pentru a determina evoluția SoH. Un proces de tip Gaussian este definit ca un set de variabile aleatoare  $\{f(x_i) | x_i \in X\}$ , în care  $x$  reprezintă spațiul intrărilor (cicluri de încărcare/descărcare), și funcțiile mediană și covariație:

$$\begin{aligned} m(x) &= E(f(x)) \\ k(x_i, x_j) &= E[(f(x_j) - m(x_j)) \cdot (f(x_i) - m(x_i))] \end{aligned} \quad (42)$$

Funcția de covariație prezintă o componentă funcțională și o componentă ce modelează zgomotul, considerat gaussian. Flexibilitatea algoritmului se poate mări prin introducerea de parametri adiționali prin folosirea funcției polinomiale de ordin 2 pe post de funcție mediană. Erorile de estimare se reduc substanțial în acest caz. Algoritmul necesită un număr ridicat de date de antrenare pentru a atinge performanța sperată, autorii folosesc un minim de 100 de cicluri.

Estimarea RUL prin folosirea unui proces Wiener este propusă în [95]. Funcția de degradare a RUL este exprimată prin următoarea formulă:

$$Y(t) = X(t) + \varepsilon = \lambda t + \sigma_B B(t) + \varepsilon \quad (43)$$



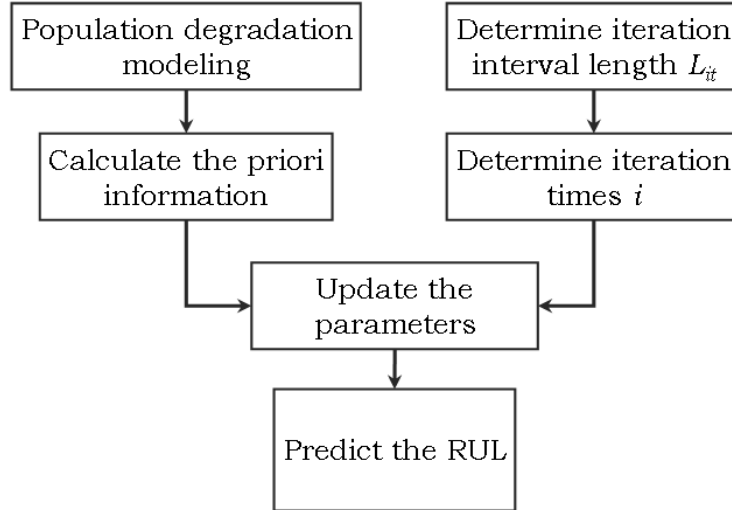


Fig. 64. Algoritm de estimare RUL propus in [87]

În [96], autorii propun o soluție de estimare a SOH bazată pe modelul autoregresiv optimizat:

$$\begin{cases} v_i^{k+1} = w \cdot v_i^k + c_1(pbest_i - x_i^k) + c_2(gbest^k - x_i^k) \\ x_i^{k+1} = x_i^k + d \cdot v_i^{k+1} \end{cases} \quad (44)$$

în care, prima ecuație reprezintă actualizarea vitezei particulelor,  $v_i^{k+1}$  este viteza nouă a particulei  $i$ ,  $w$  ponderea inerției,  $v_i^k$  este viteza precedentă a particulei  $i$ ,  $c_1$  și  $c_2$  sunt parametrii de accelerare,  $d$  este un parametru de ajustare,  $x_i^k$  este locația particulei  $i$  în iterația  $k$ ,  $x_i^{k+1}$  este locația particulei  $i$  în iterația  $k+1$ ,  $pbest_i$  este cea mai bună locație individuală pe care o poate avea particula  $i$ , și  $gbest$  este cea mai bună locație pe care o pot avea particulele ca grup în spațiul de soluții.

În [97] o rețea neuronală este folosită pentru modelarea proceselor electrochimice ale bateriei. Aceasta are funcția de activare pentru fiecare neuron în forma gaussiană:

$$i(r_k) = G(\|r_k - t_i\|) = \exp\left(-\frac{\|r_k - t_i\|^2}{\sigma_i^2}\right), \quad i = 1..M \quad (45)$$

unde,  $r_k = [V_k \ I_k \ SoC_k]^T$  reprezintă vectorul de intrare pentru rețeaua neuronală la momentul  $k$ ,  $t_i$  și  $\sigma_i$  sunt centrul și deviația standard a funcției lui Gauss,  $M$  este numărul de neuroni din nivelul ascuns. Rețeaua neuronală are ca ieșire tensiunea la bornele bateriei la momentul  $(k+1)$ . SOH este determinat folosind curve Voltaj – capacitate. Pentru a obține rezultate bune este nevoie de o antrenare cu un set consistent de date.

În [90] autorii folosesc o rețea neuronală pentru a estima SOH al bateriei. Parametrul bateriei pe care se bazează estimarea este rezistența internă a acesteia.

Rețeaua neuronală este împărțită într-o subrețea una pentru estimarea SOC și cealaltă pentru estimarea parametrilor R-C al modelului electric Randles.

O variantă probabilistă de rețele neuronale sunt folosite în [93]. Acest tip de rețea are avantajul că prezintă o viteză de învățare ridicată, iar datele de antrenare se pot aplica direct fără a fi nevoie de iterații.

Abordări ce se bazează pe teoria Bayesiană sunt prezentate în [93] și [92]. Estimarea RUL se bazează pe predicția curbelor de capacitate ale bateriei. Predicția se bazează pe analize offline și calcule a priori pentru a accelera procesul de estimare.

### **3.2. Evaluarea comparativă a metodelor existente de estimare a SOH/RUL**

Datorită multitudinii metodelor de estimare a stării de degradare a bateriilor și complexitatea acestora – fiecare metodă are particularitățile sale – este foarte greu de comparat aceste metode între ele în lipsa unui set de criterii bine definit.

Am propus următorul set de criterii care facilitează evaluarea comparativă a metodelor:

- Chimia bateriei – există metode specifice care se aplică tuturor tipurilor de chimie, precum există și metode aplicabile doar unui anumit tip
- Complexitatea computațională – acest criteriu definește tipurile de operații prin care sunt implementate metodele care pot să varieze de la cele mai simple, funcții polinomiale de ordin 2, până la cele mai complexe care pot fi operații matriciale, funcții trigonometrice, diferențiale, logaritmice, exponențiale.
- Modul de procesare a datelor – există metode care se aplică datelor primite în timp real (online), precum și metode care se aplică datelor obținute printr-o multitudine de teste de laborator (offline).
- Parametrul estimat: există metode care estimează SoH, sau RUL, sau și SoH și RUL.
- Timpul de procesare pentru prima estimare: timpul de rulare a metodei pe platforma de procesare specificată
- Precizia estimării: este un parametru ce definește cât de precisă este estimarea, este definit prin metrica definită de fiecare autor în parte (unii au autori au optat pentru eroare absolută, alți autori au optat pentru eroarea medie absolută procentuală, etc.)

În continuare am realizat un tabel în care am evaluat toate metodele prezentate la capitolul anterior prin intermediul setului de criterii definit mai sus:

**Tabel 2.** Evaluarea metodelor de estimare a stării de degradare a bateriilor

Metodă	Ref	Chimia Bateriei	Complexitate computațională	Modul de procesare a datelor	SoH/ RUL estim.	Țiimpul de procesare pentru prima estimare	Precizia estimării
Coulomb Counting	[47]	Li-Ion	Funcții și operații liniare	Online	SoH	N/A	~ 9% în ciclul 21; ~ 1% după al 8-lea ciclu
	[48]	LiFePO4	Funcții și operații liniare – metoda inițială ; Operații matriciale (înmulțire, transpusă), Funcții exponențiale (versiunea modificată)	Online	SoH	În combinație cu filtrarea Kalman:: 2000 sec	între +/-3%
OCV	[4]	Li-Ion	Operații liniare, funcții exponențiale	Online cu parametri determinați offline	SoH	N/A	Eroare de estimare: < 1%
	[22]	Li-Ion	Funcții polinomiale cu termeni exponențiali	Parametri identificați offline	SoH	N/A	< 1%
Filtre Kalman	[59]	Aplicabil tuturor tipurilor de chimii; demonstrat pentru NiMH	Operații matriciale (înmulțire, transpusă), funcții logaritmice	Offline	SoH	N/A	Max. 8.3%
	[60][60]	Li-Ion	operații matriciale (înmulțiri, transpusă), funcția radical	Offline	SoH	N/A	Eroare de estimare se situează în intervalul: +/-5%
	[55]	Li-Ion	operații matriciale (înmulțiri, transpusă), funcții polinomiale (de ordin 3)	Online, bazat pe date offline	SoH	N/A	RMS în intervalul 2.2% și 2.9%
	[53]	Li-Ion	operații matriciale (înmulțiri, transpusă)	Online	SoH	N/A	Eroarea de estimare +/- 5%
	[50]	Li-Ion	operații matriciale (înmulțiri, transpusă), funcții exponențiale	Online	SoH	Estimarea se bazează pe date luate pe fereaștră de 30 minute	Eroare de estimare a parametrilor modelului 1%; eroare de estimare a SoH < 4%

	[56]	Li-Ion	operații matriciale (înmulțiri, transpusă), funcții exponențiale	Online	SoH	N/A	eroare de estimare: < 1%
	[51]	Li-Ion	operații matriciale (înmulțiri, transpusă), funcții exponențiale	Offline	SoH	N/A	eroare de estimare: < 4%
	[52]	Li-Ion	operații matriciale (înmulțiri, transpusă), funcții exponențiale	Online cu parametri obținuți offline	SoH	N/A	eroare de estimare: < 1%
	[54]	Lead acid	operații matriciale (înmulțiri, transpusă), funcții exponențiale	Online	SoH	N/A	< 2%
	[58]	Li-Ion	operații matriciale (înmulțiri, transpusă), operații liniare	Online	SoH	Actualizarea parametrilor modelului pentru a avea o eroare acceptabilă este de 40 minute	2%
	[61]	Li-Ion	operații matriciale (înmulțiri, transpusă), funcția radical	Online	SoH	N/A	MSE în intervalul -0.6 și 0.6
Vectori suport (SVM-RVM)	[56]	Aplicabil tuturor tipurilor; Demonstrat pentru NMC (Nickel Manganese Cobalt)	Funcții exponențiale, produs vectorial (multiply and acumulate)	Offline	SoH	N/A	eroare de estimare: < 1%
	[69]	Li-Ion	Funcții exponențiale, produs vectorial (multiply and acumulate)	Offline (poate fi modificat online)	SoH and R	Specifică că este intensiv computațional	eroare de estimare ~ 2%

	[67]	Li	Funcții exponențiale, produs vectorial (multiply and acumulate), operații matriciale (înmulțiri, transpusă),	Offline	SoH	N/A	SVM RMSE între 0.47 și 1.43; RVM RMSE între $5.96 \cdot 10^{-5}$ și 0.5
	[70]	Li	Funcții exponențiale, produs vectorial (multiply and acumulate)	Online combinat cu offline	RUL	N/A	Metodă combinată: MAE 0.02; online: MAE 0.03
	[71]	Li-Ion	Funcții exponențiale, produs vectorial (multiply and acumulate)	Online, bazat pe date offline	SoH and R	N/A	MSE $8 \cdot 10^{-4}$ ; 1LC: MSE $0.1 \cdot 10^{-4}$ ; 3LC: MSE $0.02 \cdot 10^{-4}$
	[68]	Li-Ion	Funcții exponențiale, produs vectorial (multiply and acumulate), operații matriciale (înmulțiri, transpusă)	Online, bazat pe date offline	RUL	N/A	Eroare absolută de predicție după 100 de cicluri: 4 cicluri; Eroare absolută de predicție după 200 de cicluri: 2 cicluri;
	[72]	Li-Polymer	Funcții exponențiale, produs vectorial (multiply and acumulate)	Online, bazat pe date offline	SoH	Specifică că este moderat computațional	Eroare absolută de predicție: 1%
Filtru particule	[79]	Graphite anode Lithium Nickel Cobalt oxide cathode	Funcții exponențiale	Offline (regresie RVM and online (bazat pe filtru particule)	RUL	N/A	9.1% în săptămâna 32 of 64; 4.0% în săptămâna 48 of 64
	[80]	Graphite anode Lithium Nickel Cobalt oxide cathode	Funcții exponențiale, înmulțiri în virgulă flotantă	Offline (regresie RVM) și online (bazat pe filtru particule)	RUL	N/A	3.1% în săptămâna k 48 of 64
	[81]	Li-Ion	Funcții exponențiale (modelul A), înmulțiri simple (model B)	Online	SoH and R	N/A	Model A: 0.4% la ciclul 550 of 780; model B: 11.5% la ciclul 550 of 780

	[82]	Li-Ion	Funcții exponențiale, Înmulțiri în virgulă flotantă	Online	RUL	N/A	Pentru celula #01: 12.1% la ciclul 566 din 849; Pentru celula #02: 1.5% la ciclul 428 din 643; Pentru celula #V4: 0.1% la ciclul 422 din 633
	[83]	Li-Ion	Funcții exponențiale, împărțiri/înmulțiri virgulă flotantă	Online	RUL	N/A	Pentru celula A4: 4.2% la ciclul 27 din 48; pentru celula A2: 1.6% la ciclul 120 din 189
	[84]	Graphite anode Lithium Cobalt oxide cathode	Funcții exponențiale, Înmulțiri în virgulă flotantă, transpoziții vectoriale	Poate fi adaptat online	SoH and R	10 ms pe procesor Intel Core i7 M60 2.67 GB și 4 GB RAM (Matlab)	2.1% la ciclul 18 din 48; 1.1% la ciclul 250 din 610
	[85]	Li-Ion	Funcții exponențiale, Înmulțiri în virgulă flotantă	Offline și online	RUL	N/A	2.0% at la ciclul 18 din 48.
	[74]	LiFePO4	Funcții trigonometrice, Înmulțiri în virgulă flotantă	Offline și online	SoH	N/A	< 2.0%,
	[85]	Graphite anode Lithium Cobalt oxide cathode	Funcții exponențiale, Înmulțiri în virgulă flotantă, transpoziții vectoriale	Offline și online	RUL	N/A	< 5%
Logică fuzzy	[62]	Primary Li Secondary NiCd and NiMH	N/A	Offline și online	SoH	N/A	între 2.0% și 7.9% pentru SoC; N/A pentru SoH
	[63]	Graphite anode Lithium Cobalt oxide cathode	Funcții exponențiale, împărțiri/înmulțiri virgulă flotantă	Online	SoH	15.6 ms, nu se specifică platforma	între 1.4% și 9.2%
	[64]	Li-Ion	Simple înmulțiri / împărțiri	Online	SoH	N/A	Max. 5%

Spectroscopia impedanței	[16]	Graphite anode Lithium Cobalt Manganese Nickel oxide cathode	Funcții exponențiale, înmulțiri în virgulă flotantă	Offline	SoH and R	N/A	medie 2.1%
	[101]	Graphite anode Lithium Cobalt Manganese Nickel oxide cathode	Funcții exponențiale, înmulțiri în virgulă flotantă	Offline	SoH and R	N/A	Max 10% (pentru partea reală a impedanței)
Regresii liniare în sensul celor mai mici pătrate	[5]	NiMH	Ecuatii liniare simple	Online	SoH and R	< 550 $\mu$ s la 14MHz pe o platforma ARM 7	Eroare de estimare: +/- 5 cicli
Câmp magnetic	[88]	SLA	Ecuatii liniare simple	Online	SoH	N/A	N/A
Entropie de eșantionare	[89]	Li-Ion	Operații vectoriale, ecuații liniare, funcții logaritmice	Online, bazat pe date offline	SoH	N/A	Eroare relativă medie 2%
Entropie de aproximare	[90]	Lead Acid	Operații vectoriale, operații matriciale, Funcții logaritmice	Online, bazat pe date offline	SoH	N/A	N/A
Modele neobservabile	[91]	Li-Ion	Operații vectoriale, operații matriciale, Funcții exponențiale	Online, bazat pe date offline	SoH	N/A	Eroare absolută: > 21.2% în primele 681 sec, 6.7% după 1000 sec, 3% după 2000 sec, 0.6 % după 3000 sec, ~ 0% după 4000 sec, Doar puțini cicli au fost luați în considerare
PDF	[92]	LiMn2O4 LiFePO4	Funcții liniare, Funcții diferențiale	Online, bazat pe date offline	SoH	N/A	Eroare de estimare sub 2%
GPR/GPFR	[93]	Li-Ion	Operații matriciale, Funcții diferențiale, Funcții exponențiale	Offline, poate fi modificată pentru online	SoH	N/A	MAPE: < 0.5% RMSE: 1.5 $\div$ 6

Procese Wiener	[94]	Li-Ion	Expresii logaritmice și exponențiale, operații vectoriale și matriciale	Online, bazat pe date offline	RUL	N/A	MSE: < 1 după ciclul 67
Metodă AR cu PSO	[95]	Li-Ion	Operații liniare	Online, cu date minimale de antrenare	RUL	< 2 sec. pentru prima predicție, ~ 2 sec pentru următoarele predicții având 190 de eșantioane de intrare, pe un procesor Intel Core E5800, 3Ghz, 2GB RAM	Eroare de estimare a cicliilor rămași: < 30 ciclii pentru primii 120 ciclii, < 2 cycles pentru ciclii 120 - 200
Rețele neuronale	[97]	VRLA	Funcții exponențiale, operații matriciale și vectoriale	Online	SoH	N/A	Eroare de estimare max. 2%
	[51]	Li-Ion	Funcții exponențiale, operații matriciale și vectoriale	Online, bazat pe date offline	SoH	N/A	Eroare de estimare < 0.5%
	[98]	Li-Ion	Funcții exponențiale, operații matriciale și vectoriale	Online, bazat pe date offline	SoH	N/A	Eroare medie of 0.28% for a pentru un nr. de 2000 eșantioane de antrenare
Abordări Bayes	[99]	Li-Ion	operații vectoriale, Funcții exponențiale	Online	RUL	N/A	Eroare de estimare sub 2%
	[100]	Li-Ion	operații vectoriale, Funcții exponențiale	Online	RUL	N/A	Eroare de estimare între 0.3% și 10%

### 3.3. Concluzii

Tabloul 2 sumarizează o comparație a metodelor de estimare a stării de degradare a bateriilor regăsite în literatură ținând cont de 5 criterii de evaluare: chimia bateriei, complexitatea computațională, timpul de procesare, eroarea de estimare și modelul de procesare al datelor.



În general cam toate metodele chiar dacă au fost definite pentru o chimie de baterii, se pot adapta noilor sau diferitelor chimii de baterii o dată ce acestea sunt produse.

În ceea ce privește complexitatea computațională metodele bazate pe filtru de particule, filtru Kalman, vectori suport, Bayes, rețele neuronale recurente folosesc operații matriciale și funcții matematice complexe în opoziție cu metodele simple OCV, regresii polinomiale, logica fuzzy. Deși eroarea de estimare a metodelor complexe este mult mai mică, acestea rămân de cele mai multe ori la stadiu de concept, fiind aproape imposibil de implementat într-un BMS real.

Metodele simple sunt cele care se regăsesc des implementate în sisteme de management reale, doar că eroarea de estimare a acestora este destul de mare. O categorie aparte o constituie rețelele neuronale care prezintă o complexitate medie spre mare în funcție de modelul ales. Datorită progresului tehnologic din acest domeniu și apariția procesoarelor cu hardware dedicat operațiilor rețelelor neuronale problema implementării acestora în sisteme de management reale nu mai constituie o problemă. În acest context, metodele de estimare bazate pe rețele neuronale pot prezenta erori de estimare foarte bune dacă se utilizează în mod corect seturile de date pentru tipul de baterii pentru care se vrea a face estimarea. Producerea acestor seturi de date rămâne o problemă datorită timpului și condițiilor de laborator necesare pentru o acuratețe cât mai bună.

Un alt aspect observat din analiza metodelor existente este că în general autorii s-au axat pe a prezenta erorile și modul de funcționare a metodelor în condiții de laborator (temperatură 25 de grade Celsius) și curent de descărcare constant. Aceste condiții nu sunt apropiate de condițiile de utilizare obișnuită când curentul de descărcare este unul variabil, iar temperatura este de asemenea diferită de cea de laborator.

## 4. METODĂ DE EVALUARE A STĂRII DE DEGRADARE A BATERIILOR BAZATE PE REGRESIE POLINOMIALĂ

### 4.1. Descrierea metodei

Această metodă a fost dezvoltată în laboratorul DSPLabs din cadrul universității Politehnica din Timișoara. În parte, cercetarea a fost finanțată de către Ministerul Educației și Cercetării prin grantul PNCDI II ID-22/2007-2010, la care am colaborat cu Gabriel N. Cârstoiu.

În cadrul sistemului robotic dezvoltat a existat nevoia predicției stării de degradare a bateriilor, scop urmărit prin implementarea acestei metode.

Metoda este de tip online și se bazează pe regresia liniară de ordin 2, după cum urmează: regresia polinomială încearcă să determine cu acuratețe funcția capacității  $C$  dată prin punctele  $C_k$  corespunzătoare fiecărui ciclu  $k$  de încărcare / descărcare.

Regresia polinomială are următoarea formă:

$$C_k = ak^2 + bk + c, a < 0 \quad (46)$$

Dacă luăm în considerare valorile capacităților a  $n$  cicli de încărcare / descărcare, se pot obține valorile parametrilor funcției prin rezolvarea sistemului de ecuații:

$$\begin{cases} a \sum_k k^2 + b \sum_k k + cn = \sum_k C_k \\ a \sum_k k^3 + b \sum_k k^2 + c \sum_k k = \sum_k k C_k \\ a \sum_k k^4 + b \sum_k k^3 + c \sum_k k^2 = \sum_k k^2 C_k \end{cases} \quad . k = \overline{1, n}, n \geq 3 \quad (47)$$

Acest sistem se poate rezolva prin intermediul determinanților:

$$\Delta = \begin{vmatrix} \sum_k k^2 & \sum_k k & n \\ \sum_k k^3 & \sum_k k^2 & \sum_k k \\ \sum_k k^4 & \sum_k k^3 & \sum_k k^2 \end{vmatrix} \quad (48)$$

$$\Delta_a = \begin{vmatrix} \sum_k C_k & \sum_k k & n \\ \sum_k k C_k & \sum_k k^2 & \sum_k k \\ \sum_k k^2 C_k & \sum_k k^3 & \sum_k k^2 \end{vmatrix} \quad (49)$$

$$\Delta_b = \begin{vmatrix} \sum_k k^2 & \sum_k C_k & n \\ \sum_k k^3 & \sum_k k C_k & \sum_k k \\ \sum_k k^4 & \sum_k k^2 C_k & \sum_k k^2 \end{vmatrix} \quad (50)$$

$$\Delta_c = \begin{vmatrix} \sum_k k^2 & \sum_k k & \sum_k C_k \\ \sum_k k^3 & \sum_k k^2 & \sum_k k C_k \\ \sum_k k^4 & \sum_k k^3 & \sum_k k^2 C_k \end{vmatrix} \quad (51)$$

După calcularea valorilor determinanților se pot obține ușor valorile parametrilor ce determină funcția polinomială aplicând relațiile:

$$a = \frac{\Delta_a}{\Delta}, \quad b = \frac{\Delta_b}{\Delta}, \quad c = \frac{\Delta_c}{\Delta}. \quad (52)$$

Pentru calcularea mai simplă a determinanților se pot folosi formulele recurente care asigură simplificarea computațională, respectiv fac posibilă implementarea pe o platformă de sistem încorporat cu putere de procesare și memorie redusă:

$$S_{k+1} = S_k + (k+1), \quad S_{(k+1)^2} = S_k^2 + (k+1)^2, \quad \text{etc.} \quad (53)$$

Cu ajutorul funcției polinomiale obținute putem calcula primul ciclu  $m$ , de încărcare / descărcare pentru care capacitatea bateriei este sub un anumit nivel,  $F \cdot C_{nominal}$  pentru care SoH este considerat 0%.

Calculul valorii  $m$  se reduce la rezolvarea inecuației:

$$F \cdot C_{nominal} > ak^2 + bk + c \quad (54)$$

Cum  $a < 0$ ,  $m$  devine:

$$m = \left\lfloor \frac{-b - \sqrt{b^2 - 4ac}}{2a} \right\rfloor, \quad (55)$$

în care  $\lfloor x \rfloor$  este partea întreagă a lui  $x$ .

Metoda regresiei polinomiale prezintă o acuratețe mare în cazul în care valorile capacității bateriei urmează un trend liniar. În cazul în care apar nelinearități, acuratețea începe să fie afectată în sens negativ.

## 4.2. Implementarea metodei

Pentru evaluarea metodei am recurs la implementarea acesteia pe hardware-ul aferent plăcii de management al consumului, care face parte din arhitectura unui WIT (wireless intelligent terminal), specificat în proiectul CORE-TX (Collaborative Robotic Environment – The Timișoara Experiment) [102].

Această placă de management al consumului are rol de a asigura energia electrică întregului sistem precum și monitorizarea stării de degradare a bateriei în vederea creșterii duratei de utilizare al acesteia. Versiunea prototip funcțional al plăcii de management al consumului este reprezentată în figura de mai jos.

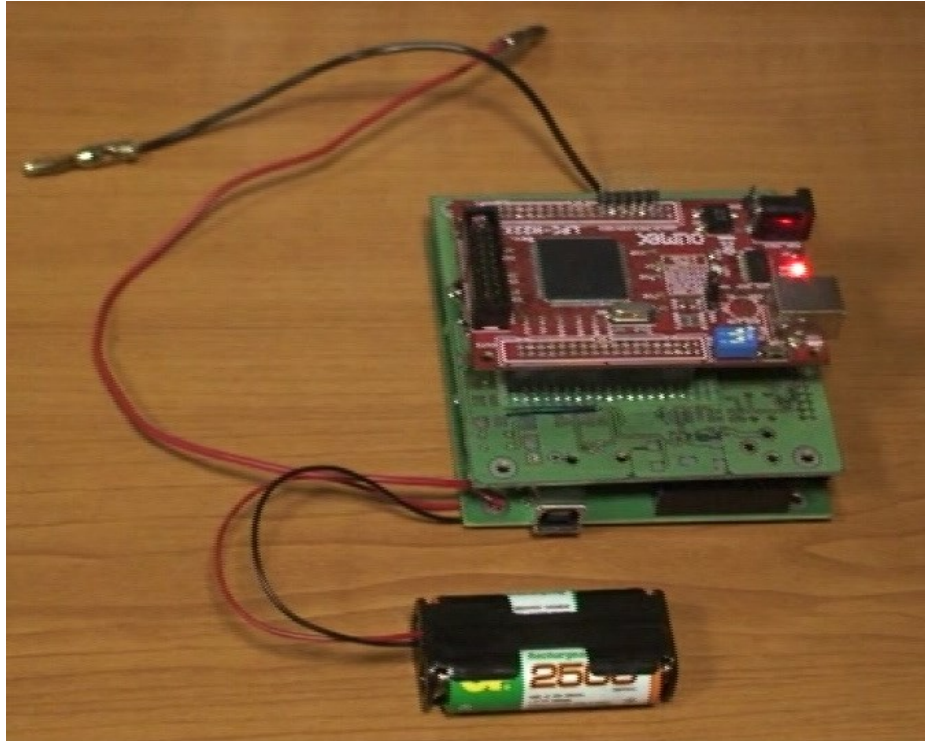


Fig. 25 Placa de management al consumului în cadrul unui WIT

Managementul bateriei se face prin intermediul măsurării cu acuratețe ridicată a tensiunii, curentului și a temperaturii bateriei. Curentul este măsurat prin intermediul unei scheme de tip high side fig. 26, iar temperatura este măsurată cu un senzor de tip TMP101[103]. În acest tip de schemă,  $R_s$  este rezistența consumatorului (în cazul nostru celelalte plăci din sistem),  $I_{circ}$ , curentul prin circuit,  $R_{shunt}$ , rezistența șuntului. Folosind componente de înaltă precizie, implementarea asigură o rezistență bună la zgomot. Metoda folosită pentru calculul capacității este coulomb counting.

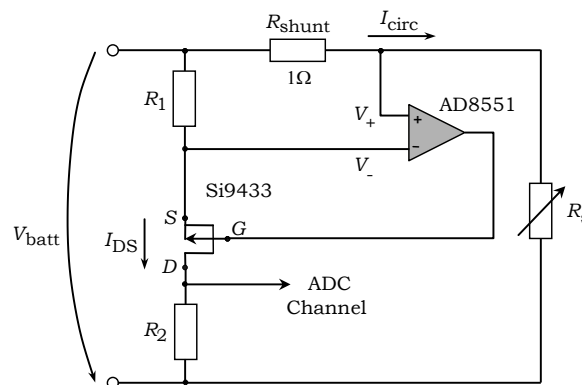


Fig. 76 Schema de tip high-side pentru măsurarea curentului

Din punct de vedere al subsistemului software, a fost aleasă o arhitectură pe nivele, în care distingem trei nivele (fig. 27):

- Managerul stării de degradare al bateriilor (nivel aplicație): asigură măsurarea, stocarea și procesarea parametrilor bateriei precum și tratarea comenzilor de management al consumului.
- Nivelul de management al taskurilor sistem: se ocupă de planificarea și execuția task-urilor hard real time, corespunzător modelului HARETICK (Hard REal-Time Compact Kernel) [104].
- Nivelul de abstractizare hardware: conține driverele și interfețele microcontroler-ului precum și a celorlalte periferice de pe placă.

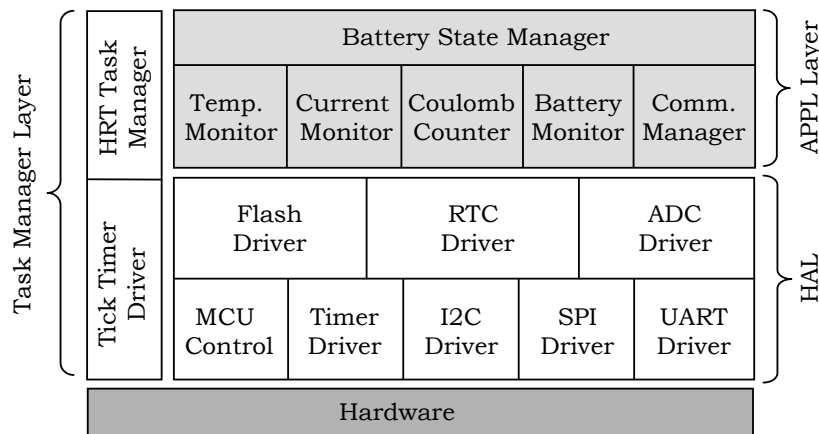


Fig. 27 Arhitectura software al plăcii de management al consumului

Software-ul care rulează predicția propriu-zisă poate fi reprezentat prin intermediul următorului pseudocod:

Codul 1. Secvența de cod pentru predicție

```

1:   while not (battery cycle completed) do
2:       wait
3:   end while
4:   increment  $n$ 
5:   if  $n \geq n_{\min}$  do
6:       Calculate  $Sum_1, Sum_2, \dots, Sum_7$  with (8)
7:       Calculate  $\Delta_1, \Delta_2, \Delta_3$ 
8:       Calculate  $\Delta$ 
9:       Calculate  $\Delta_4, \Delta_5, \Delta_a$ 
10:      Calculate  $\Delta_6, \Delta_b$ 
11:      Calculate  $\Delta_7, \Delta_c$ 
12:      // Scaling of  $\Delta, \Delta_a, \Delta_b, \Delta_c$ 
13:           $\Delta = \Delta / 128$ 
14:           $a = (\Delta_a \text{ left shifted } 20 \text{ bits}) / \Delta$ 

```

---

```
15:           $b = (\Delta_b/\Delta)$  left shifted 20 bits
16:           $c = (\Delta_c/\Delta)$  left shifted 20 bits
17:          Solve (9) and calculate  $m$  according to (10)
18:           $n_{RUL} = m - n$ 
19:          end if
```

---

La fiecare pas (ciclu de descărcare al bateriei), se trece la calculul determinanților urmat de calculul parametrilor funcției de reprezentare a capacității bateriei. Ultimul pas este calcularea numărului de cicli de încărcare / descărcare a bateriei până când aceasta este considerată neutilizabilă.

Algoritmul ales este unul foarte simplu, ales pentru a asigura posibilitatea implementării acestuia în sisteme încorporate cu resurse computaționale și de memorie limitate. Chiar și așa, implementarea pe platforma de 32 biți ARM7 nu a fost lipsită de probleme:

- calculele intermediare depășesc 32 de biți - în acest caz multe din variabile au necesitat o reprezentare pe 64 de biți
- arhitectura microcontrolerului nu oferă suport nativ (hardware) pentru operații în virgulă flotantă - în acest caz am recurs la scalarea și trunchierea variabilelor la numere întregi mari.
- Calculul funcției radical, al valorii de 64 de biți a trebuit optimizată deoarece implementarea standard C oferă o implementare ineficientă din punct de vedere computațional.

## 5. METODĂ DE EVALUARE A STĂRII DE DEGRADARE A BATERIILOR FOLOSIND REȚELE NEURONALE RECURENTE DE TIP GRU

### 5.1. Descrierea metodei

Voi descrie metoda de estimare a stării de degradare a bateriilor pornind de la Fig. 288 ce prezintă structura unui sistem de management al bateriilor tipic ce folosește o metodă de estimare a stării de degradare în timp real (online).

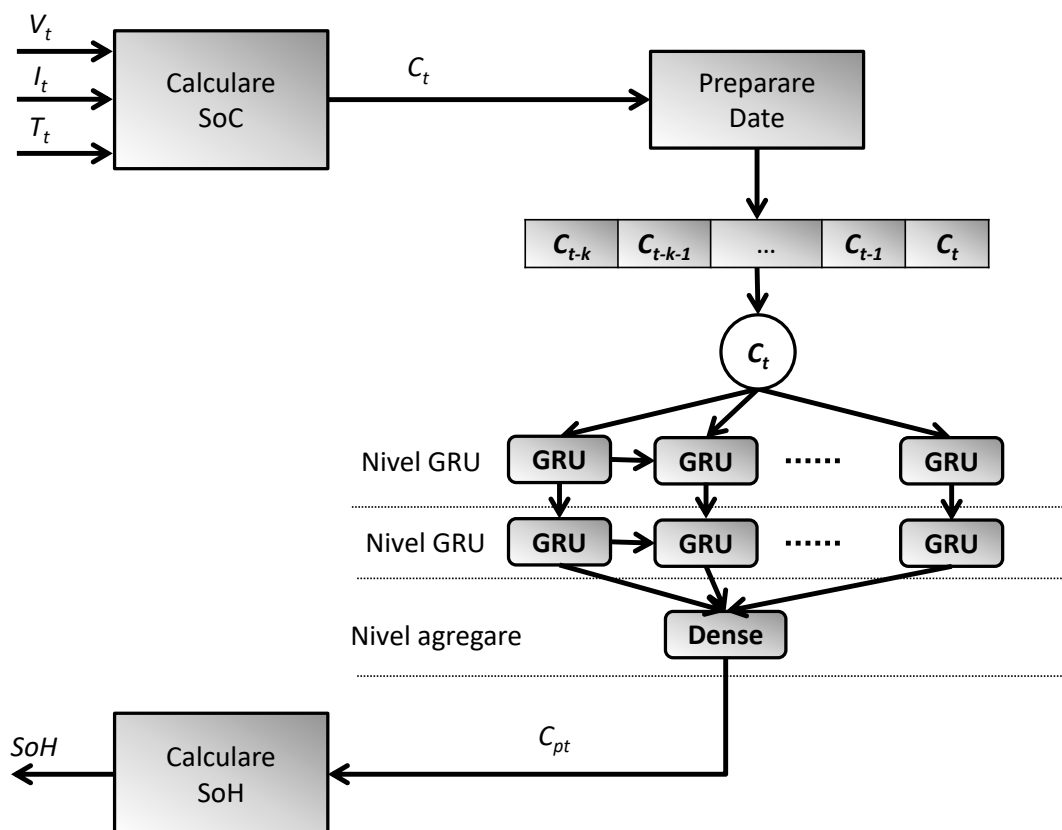


Fig. 288 Schema de principiu a metodei de estimare a stării de degradare a bateriilor bazată pe rețele neuronale de tip GRU

Acest sistem este compus din mai multe subcomponente:

- Blocul ce asigură determinarea SoC – acest bloc asigură calcularea în timp real a parametrului de ieșire (în cazul nostru capacitatea bateriei la momentul  $t$ ),  $C_t$  în funcție de parametrii de intrare, care în exemplul de față



sunt tensiunea la borne, curentul de descărcare și temperatura bateriei, reprezentate prin  $V_t$ ,  $I_t$ ,  $T_t$ . Acest bloc poate fi implementat prin utilizarea oricărei metode cunoscute de determinare a SoC al bateriei de la cele mai simple cum ar fi integrarea în timp a produsului curent de descărcare și tensiune la borne (cunoscută în literatură ca „Coulomb Counting”) până la cele mai complexe. Pentru lucrarea de față am folosit baze de date ce conțin deja acest parametru calculat în condiții de laborator și nu am implementat un astfel de bloc computațional, obiectivul lucrării fiind elaborarea unei metode de estimare a stării de degradare a bateriei.

- Blocul „Preparare date” – asigură pregătirea datelor și ajustarea acestora pentru prezentarea acestora la intrarea rețelei neuronale recurente. Acest stadiu de pregătire a datelor presupune următoarele operații:
  - Crearea secvenței de intrare în rețeaua neuronală. Acest lucru se realizează prin obținerea în fiecare instanță de timp (după fiecare ciclu de descărcare) a unui vector  $C_t$  ce conține valorile  $C_{t-k}$ ,  $C_{t-k-1}, \dots, C_{t-1}$ ,  $C_t$ , a capacităților pe ultimele  $k$  cicluri. De menționat este faptul că în primele  $k$  cicluri, acest bloc nu produce nici o ieșire - cu alte cuvinte nu se realizează nici o estimare a stării de degradare a bateriilor.
  - Limitarea intrărilor în valorile minime și maxime cu care a fost antrenată rețeaua. Acest lucru asigură evitarea funcționării incorecte a rețelei neuronale prin introducerea unor valori în afara intervalului cu care aceasta operează normal.
  - Normalizarea datelor de intrare în intervalul  $[-1, 1]$ , acest lucru fiind necesar pentru funcția de activare din nivelul ascuns.
- Rețeaua neuronală propriu-zisă. Structura rețelei va fi discutată în paragraful următor. Ieșirea rețelei neuronale este capacitatea estimată în formă normalizată.
- Blocul de calculare SoH se ocupă de transformarea capacității estimate a bateriei și calcularea procentuală a stării de degradare a bateriei conform relației:

$$SOH [\%] = \begin{cases} 100\%, & \frac{C_{bat}}{C_{nom}} > 1 \\ \left( 1 - \frac{C_{bat}}{C_{nom}} \right) \cdot 100 [\%] & \\ 0\%, & \frac{C_{bat}}{C_{nom}} < SOH_0 \end{cases} \quad (56)$$

Implementarea metodei de estimare a stării de degradare a bateriei se poate sumariza prin intermediul următorului pseudocod:

Codul 2. Metoda de estimare a stării de degradare a bateriei

---

```
1: while not (battery cycle completed) do
2:     wait
3: end while
4: determine SoC
5: Prepare data  $C_{t-k}, C_{t-k+1}, \dots, C_{t-1}, C_t$ 
6: Calculate  $C_{pt}$ 
7: Calculate  $SOH_t$ 
8: if  $C_{pt}/C_{nom} > 1$ 
9:      $SOH_t[\%] = 100\%$ 
10: else if  $C_{pt}/C_{nom} < SOH_0$ 
11:      $SOH_t[\%] = 0\%$ 
12: else
12:      $SOH_t[\%] = \left( 1 - \frac{1 - \frac{C_{pt}}{C_{nom}}}{SOH_0} \right) * 100 [\%]$ 
13: end if
```

---

## 5.2. Structura rețelei neuronale

Rețeaua neuronală recurentă are structura prezentată în fig. 29 Este formată din două nivele ascunse formate din celule GRU și un nivel de agregare (denumit "dense"). De notat este faptul că pentru simplitate am ales modul de reprezentare al rețelei recurente cel desfășurat (unrolled).

Pentru reprezentarea acestei rețele am folosit mediul Keras 2.4.0 împreună cu Tensorflow 2.1:

Codul 3. Reprezentarea rețelei neuronale propuse

---

```
model = Sequential()
model.add(GRU(nCells, activation='tanh', recurrent_activation='sigmoid', input_shape=(
    nSteps, nFeatures), implementation=1, use_bias=True, reset_after=True,
    return_sequences=True))
model.add(GRU(nCells, activation='tanh', recurrent_activation='sigmoid', input_shape=(
    nSteps, nFeatures), implementation=1, use_bias=True, reset_after=True))
model.add(Dense(1))
```

---

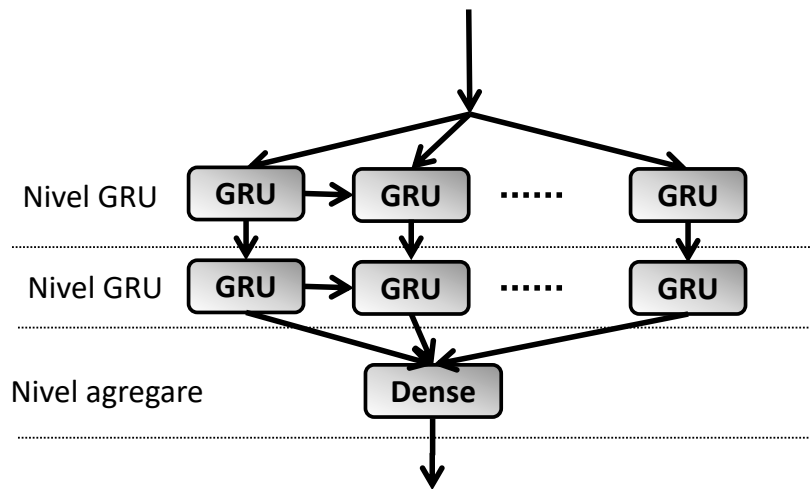


Fig. 99 Structura rețelei neuronale cu celule de tip GRU

### 5.3. Antrenarea rețelei neuronale

Pentru învățare am folosit un algoritm cu gradient descrescător în varianta de optimizare Adam (*adaptive moment estimation*). Acest algoritm combină avantajele introduse de algoritmul Adagrad și RMSProp în sensul că menține rata de învățare per parametru, lucru important din punct de vedere al performanței când gradientii sunt matrici rare. Ratele de învățare per parametru sunt actualizate ținând cont de media celor mai recente valori ale gradientilor.

Algoritmul salvează media mobilă exponențială a pătratului gradientilor anteriori  $v_t$ , asemenea Adadelta și RMSProp, precum și media mobilă exponențială a gradientilor anteriori  $m_t$ , asemenea algoritmului momentum.

Cele două medii mobile exponențiale se pot calcula folosind următoarele relații:

$$\begin{cases} m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \\ v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \end{cases} \quad (57)$$

La inițializare și în cadrul primilor pași se poate observa că valorile  $m_t$  și  $v_t$  înclină spre valoarea 0, în special în momentul în care valorile lui  $\beta_1$  și  $\beta_2$  sunt apropiate de 1. Pentru a evita această situație se iau în considerare valorile corectate a celui de-al doilea momentum:

$$\begin{cases} \hat{m}_t = \frac{m_t}{1 - \beta_1^t} \\ \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \end{cases} \quad (58)$$

Folosind cele două ecuații putem obține ecuația regulii de actualizare a parametrilor:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{V}_t + \varepsilon}} \hat{m}_t \quad (59)$$

În care  $\theta$  este matricea de parametri a modelului care sunt actualizați în fiecare ciclu  $t$ ,  $\eta$  este rata de învățare, și  $\varepsilon$  este o constantă foarte mică pentru a evita împărțirea cu 0. Din punct de vedere al performanței, algoritmul Adam este destul de rapid, având un consum de memorie relative scăzut

#### 5.4. Variantă modificată cu filtru treaptă

Am modificat metoda de estimare a stării de degradare a bateriilor prin introducerea unui filtru de tip treaptă. Acest filtru funcționează după cum urmează:

$$f(x_n) = \begin{cases} x_n, & x_n < x_{n-1} \\ x_{n-1}, & x_n \geq x_{n-1} \end{cases} \quad (60)$$

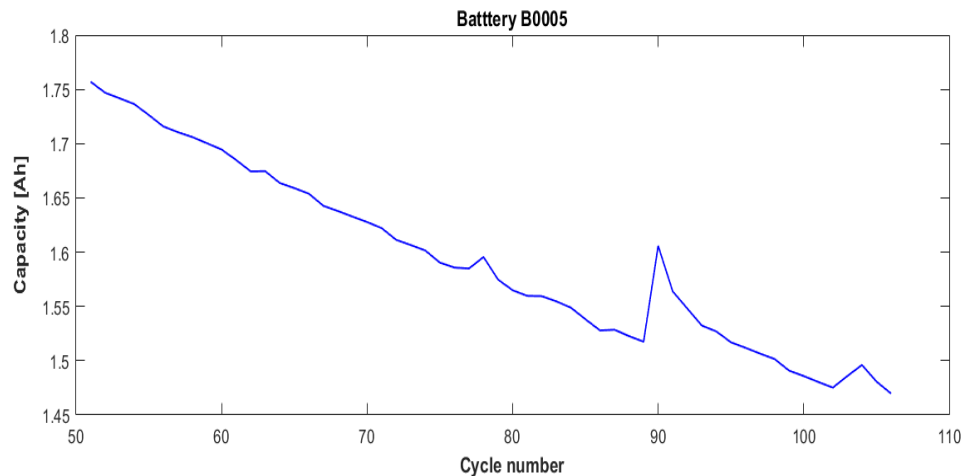


Fig. 30. Efectul de regenerare întâlnit la unele tipuri de baterii

Datorită utilizării acestui filtru se pot crește performanțele estimării, reducându-se astfel erorile de estimare. O analiză detaliată este prezentată în secțiunea de evaluare a performanței.

În figura 31 este prezentată datele aferente capacității bateriei B01 obținută prin efectuarea ciclurilor de încărcare/descărcare folosind aparatul CADEX 7400ER. Se observă că acest tip de baterie nu prezintă efect de regenerare.

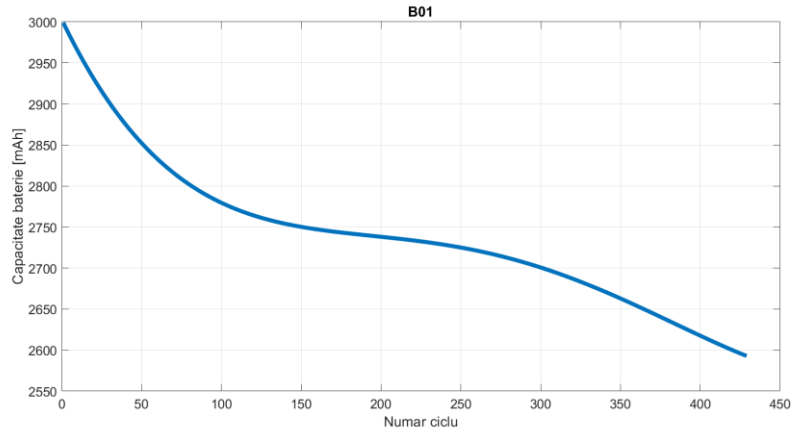


Fig. 31. Capacitatea bateriei B01(Panasonic NCR18650A)

Metoda de estimare modificată este prezentată în figura de mai jos:

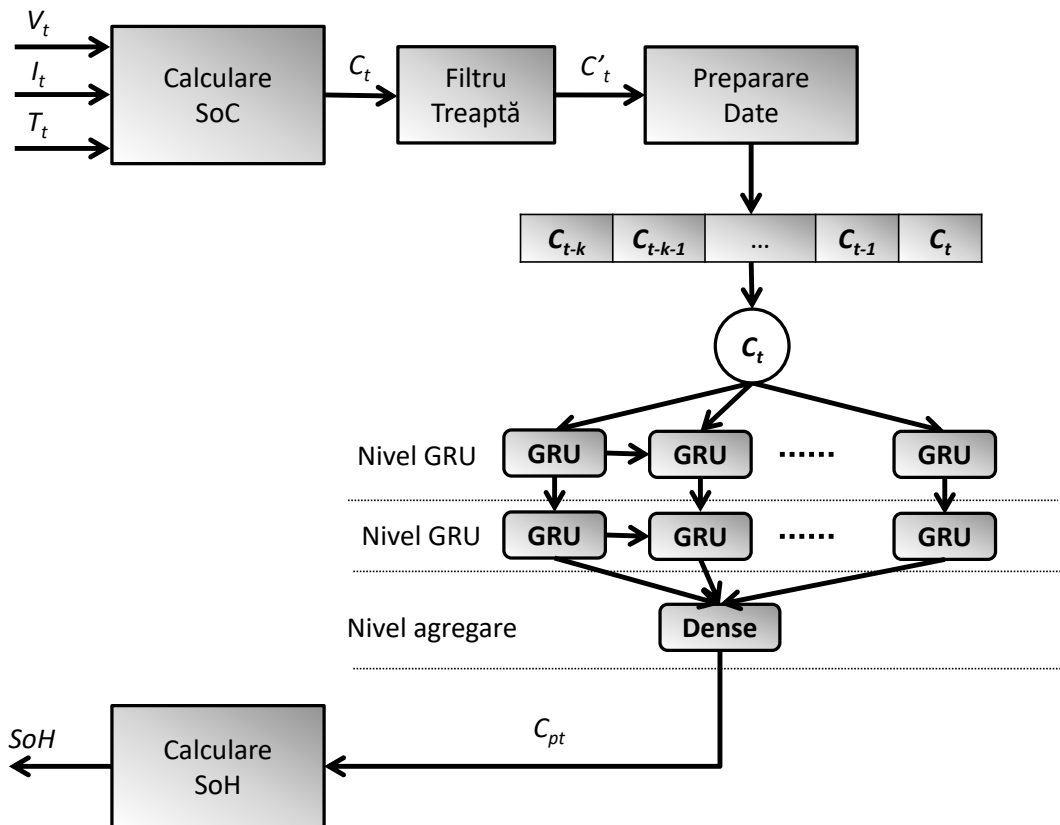


Fig. 32. Metoda de estimare modificată prin introducerea filtrului treaptă

## **6. SISTEM CADRU DE ANALIZĂ A PERFORMANȚELOR METODELOR DE ESTIMARE A STĂRII DE DEGRADARE A BATERIILOR BAZATE PE REȚELE NEURONALE**

### **6.1. Obiective**

Datorită multitudinii metodelor de estimare a stării de degradare a bateriilor, sarcina de a compara aceste metode este destul de grea, mai ales prin prisma implementării acestora în sisteme de management al bateriilor. Multe din metode sunt prezentate de către autori la nivel de concept acompaniat de o simulare pe calculator. În lipsa unui cadru de evaluare al acestor metode pe implementări în sisteme reale este foarte greu să se discearnă care din metode este fezabilă a fi implementată (chiar și într-un sistem comercial).

Prin teza de față propun un sistem de evaluare a metodelor de estimare a stării de degradare a bateriilor care poate fi folosit atât pentru a compara metodele la nivel de simulare pe calculator cât și compararea acestora la nivel de implementare pe diferite platforme de sisteme încorporate cunoscute.

### **6.2. Structura sistemului**

Structura sistemului de analiză a performanțelor este una pe nivele și este cuprinsă în figura de mai jos:

- Nivelul 1 – este nivelul de reprezentare a modelului. Pentru simplitate am ales Keras ca mediu de modelare cu ajutorul limbajului de programare Python. Cu ajutorul particularităților pe care le pune la dispoziție mediul Keras, se pot defini o mulțime de metode de estimare. Acest nivel se adresează simulării metodei pe calculator. Este la latitudinea fiecăruia să configureze metricile de evaluare a performanțelor a metodei. În urma modelării pe calculator a metodei rezultă un set de parametri ce vor fi utilizați pentru implementarea în sisteme încorporate.
- Nivelul 2 – este nivelul de reprezentare computațional pe platforme de sisteme încorporate. Se pot alege pentru comparare mai multe biblioteci de procesare cum ar fi: PnP (Paper and Pencil), Tensorflow (pentru anumite sisteme), biblioteci matematice de algebra liniară precum OpenBlas, Blasfeo. Acest nivel este implementat în C/C++ pentru a ușura portarea pe diferite arhitecturi hardware și bineînțeles pentru a deschide calea spre optimizări computaționale acolo unde este cazul.
- Nivelul 3 – este nivelul de platformei HW pe care se dorește implementarea. Setul de platforme hardware este destul de cuprinzător și conține arhitecturi Arm de la cele mai simple (M3) până la cele mai complexe(A53).

- Nivelul 4 – este nivelul de colectare a metricilor de evaluare a performanței metodei de evaluare a stării de degradare a bateriilor în implementare reală (pe sistem încorporat). Am definit 4 metrici de evaluare a performanței: timpul de procesare, memoria utilizată, consumul de energie precum și eroarea de estimare.

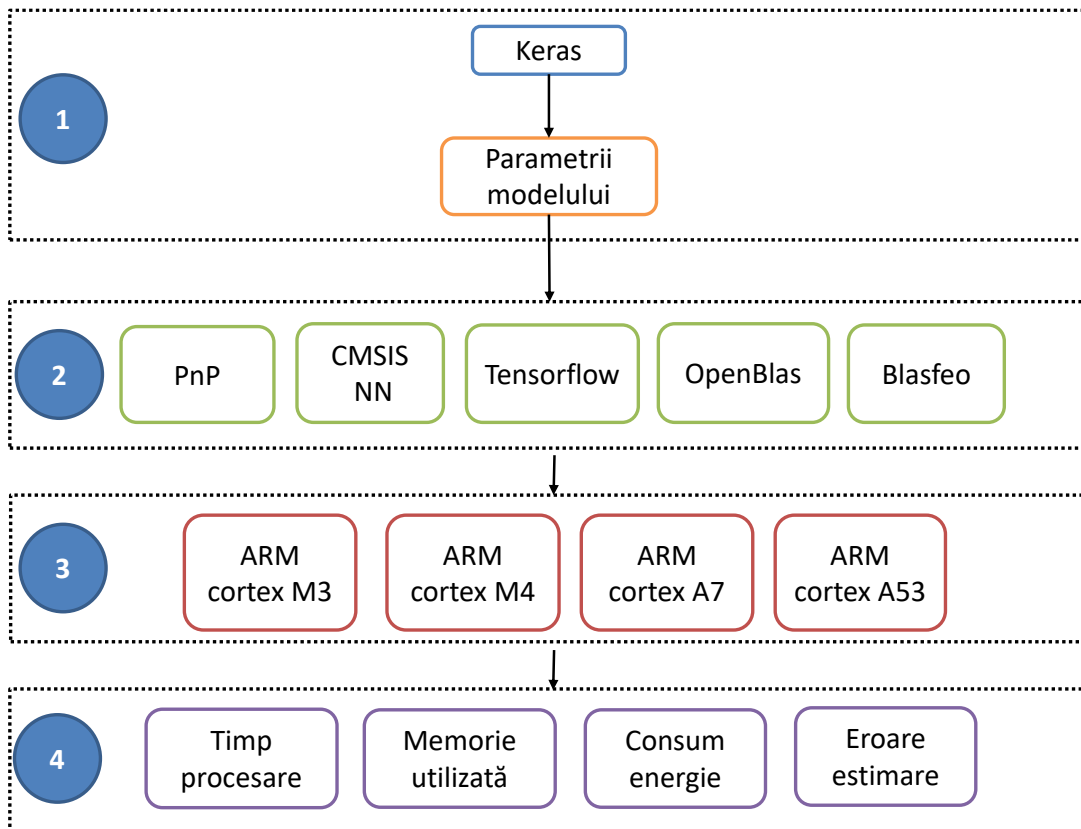


Fig. 33. Structura sistemului cadrul de evaluare a performanțelor metodelor de estimare a stării de degradare a bateriilor

### 6.3. Criterii de evaluare și metrici de performanță

Criteriile de evaluare și metricile de performanță fac parte din nivelul 4 al sistemului cadru descris în acest capitol:

- Timpul de procesare – este timpul definit în ms pentru calcularea estimării stării de degradare a bateriilor la fiecare ciclu de descărcare. Acest timp depinde de frecvența procesorului și numărul de nuclee pe care rulează algoritmul. Am considerat calcularea timpul de procesare prin următoarea formulă: Timpul de procesare =  $T_f - T_i$  [ms], unde  $T_f$  - este instanța de timp la finalul execuției estimării,  $T_i$  - este instanța de timp înainte de execuția estimării.

- Memoria utilizată – în funcție de platforma aleasă avem mai multe tipuri de memorie: flash (pentru constante), SRAM și DRAM pentru algoritmi proprii. Memoria utilizată este raportată în KBytes. Memoria utilizată am calculat-o cu ajutorul hărții de memorie rezultată în urma compilării pentru implementările în limbajul C/C++, iar pentru cele bazate pe limbajul Python am folosit utilitare de calcul al maximumului de memorie alocată dinamic utilizată.
- Consumul de energie – este consumul de energie utilizat pentru estimarea stării de degradare la finalul fiecărui ciclu de descărcare a bateriei. Este raportat în mAh.
- Eroarea de estimare este eroare relativă procentuală rezultată în urma aplicării setului de date de test. Aceasta se poate compara cu eroarea obținută în urma rulării simulării pe calculator pentru a observa dacă există vreo diferență datorită implementării pe sistem încorporat. Această eroare se poate calcula folosind relația:  
$$E = (V_{est} - V_{real}) / 100 [\%],$$
unde  $V_{est}$  – reprezintă valoarea estimată prin aplicarea metodei,  $V_{real}$  - reprezintă valoarea reală a capacității bateriei la ciclul de descărcare dat.

#### 6.4. Exemplu de evaluare a performanței folosind implementarea PnP

Evaluarea performanțelor metodei folosind implementarea PnP presupune translatarea parametrilor GRU obținuți prin antrearea modelului în structura compatibilă PnP. Aceasta presupune conversia din formatul .hd5 (obținut prin antrenare în Keras) într-un format simplu C ce se poate observa în codul următor:

Codul 4. Definierea parametrilor modelului pentru implementarea PnP

---

```
static const float wz[] = {
    -0.6168708, ..., -0.7502499,
};
static const float wr[] = {
    -0.45586815, ..., -0.17601903,
};
static const float wh[] = {
    0.78859043, ..., -0.64399165,
};
static const float bwz[] = {
    -0.17063044, ..., 0.1079701,
};
static const float bwr[] = {
    0.25578716, ..., 0.19392361,
};
static const float bwh[] = {
    -0.14760374, ..., 0.08432186,
};
static const float uz[] = {
    0.33544156, ..., -0.1061423,
```



```
};
static const float ur[] = {
    0.3580518, ..., -0.09428676,
};
static const float uh[] = {
    -0.21631242, ..., 0.07331745,
};
static const float buz[] = {
    -0.17063044, ..., 0.1079701,
};
static const float bur[] = {
    0.25578716, ..., 0.19392361,
};
static const float buh[] = {
    -0.12483689, ..., 0.08706983,
};
static const float wz2[] = {
    0.70549613, ..., -0.08304864,
};
static const float wr2[] = {
    0.02348625, ..., 0.3856362,
};
static const float wh2[] = {
    -0.5259535, ..., 0.52621037,
};
static const float bwz2[] = {
    -0.17133844, ..., 0.10030975,
};
static const float bwr2[] = {
    -0.14853013, ..., 0.07579752,
};
static const float bwh2[] = {
    0.07097511, ..., 0.07575441,
};
static const float uz2[] = {
    -0.30984527, ..., 0.03696784,
};
static const float ur2[] = {
    -0.2557064, ..., 0.22604541,
};
static const float uh2[] = {
    -0.21545862, ..., -0.16208637,
};
static const float buz2[] = {
    -0.17133844, ..., 0.10030975,
};
static const float bur2[] = {
    -0.14853013, ..., 0.07579752,
};
static const float buh2[] = {
    0.08364447, ..., 0.06608906,
};
```

---

```
static const float dense[] = {  
    0.7647076, ..., 0.8833759,  
};  
static const float bias[] = {0.08102331};
```

Înainte de formatarea datelor de intrare, se stabilesc parametrii rețelei neuronale, utilizând parametrii de configurare din codul 5. Datele de intrare și de test trebuie definite conform codului 6.

#### Codul 5. Definirea arhitecturii rețelei neuronale

---

```
// this is the size of input sequence  
#define INPUT_SIZE (5)  
// this is the size for hidden layer  
#define HIDDEN_SIZE (50)  
#define NO_UNROLL INPUT_SIZE
```

---

#### Codul 6. Definirea intrărilor

---

```
static const float inputData[][INPUT_SIZE] = {  
    {0.57, 0.546, 0.519, 0.519, }, ...  
    {0.495, 0.274, 0.091, -0.269, -0.512, },  
};  
static float testData[] = {  
    0.518378626, ...  
    -0.862251519,  
};
```

---

Evaluarea propriu-zisă a implementării se face folosind apelul funcțiilor specifice PnP:

#### Codul 7. Rularea propriu-zisă a metodei

---

```
ModelInit();  
ModelEvaluate();
```

---

## 7. EVALUAREA PERFORMANȚELOR

### 7.1. Criterii de evaluare

În vederea evaluării performanțelor am utilizat criteriile descrise la secțiunea 6. În plus, pentru validarea rețelei neuronale recurente am analizat repetabilitatea, validarea încrucișată și arhitectura rețelei.

### 7.2. Configurarea metodei

Configurarea metodei este sumarizată în tabelul de mai jos.

**Tabel 3.** Configurarea metodei de estimare bazată pe rețele neuronale de tip GRU

Parametru	Valoare
Nr. neuroni in nivelul 1	50
Nr. neuroni in nivelul 2	50
Nr. epoci antrenare	500
Nr. valori în fereastra de timp	5
Procent date antrenare + validare	80%
Procent date test	20%

În urma acestei configurări au rezultat următoarele caracteristici al rețelei:

**Tabel 4.** Caracteristici de antrenare a rețelei

Nr. total de parametri ai modelului	23001
Eroare medie absoluta la antrenare	0.0264

### 7.3. Rezultatele evaluării metodei pe setul de date NASA PCOE

În această secțiune voi prezenta rezultatele experimentale obținute în urma rulării metodei de estimare a stării de degradare a bateriilor pe întreg setul de date NASA PCOE. Specificațiile complete al acestui set de date sunt prezentate în Anexa 1.

Pentru evaluarea metodei propuse de estimare a stării de degradare a bateriilor folosind rețele neuronale recurente de tip GRU, a fost conceput un număr de 8 scenarii de test reprezentative pentru setul de date de baterii amintit mai sus. Alegerea scenariilor a fost făcută ținând cont de caracteristicile acestui set de date: curentul de descărcare, temperature de operare, tipul de chimie care prezintă fenomenul de regenerare. De asemenea au fost luate în considerare și scenariile folosite de alți autori (cele efectuate la curent constant la temperature ambientală) pentru comparație. Cele 8 scenarii de test sunt:

- 25-DEG-CC – setul de date ce conține date ale bateriilor ciclitate în condiții de 25 de grade Celsius, curent de descărcare constant.
- ALL-DEG-CC – setul de date ce conține date ale bateriilor ciclitate în toate condițiile de temperatură puse la dispoziție, curent de descărcare constant

- 25-DEG-CC-RW – setul de date ce conține date ale bateriilor ciclitate în condiții de 25 de grade Celsius, curent de descărcare constant și variabil.
- ALL-DEG-CC-RW – setul de date ce conține date ale bateriilor ciclitate în toate condițiile de temperatură puse la dispoziție, curent de descărcare constant și variabil.
- 25-DEG-CC-SF – setul de date ce conține date ale bateriilor ciclitate în condiții de 25 de grade Celsius, curent de descărcare constant.
- ALL-DEG-CC-SF – setul de date ce conține date ale bateriilor ciclitate în toate condițiile de temperatură puse la dispoziție, curent de descărcare constant
- 25-DEG-CC-RW-SF – setul de date ce conține date ale bateriilor ciclitate în condiții de 25 de grade Celsius, curent de descărcare constant și variabil.
- ALL-DEG-CC-RW-SF – setul de date ce conține date ale bateriilor ciclitate în toate condițiile de temperatură puse la dispoziție, curent de descărcare constant și variabil.

În figura de mai jos sunt prezentate intervalele de eroare rezultate în urma evaluării metodei propuse pentru cele 8 scenarii mai reprezentative

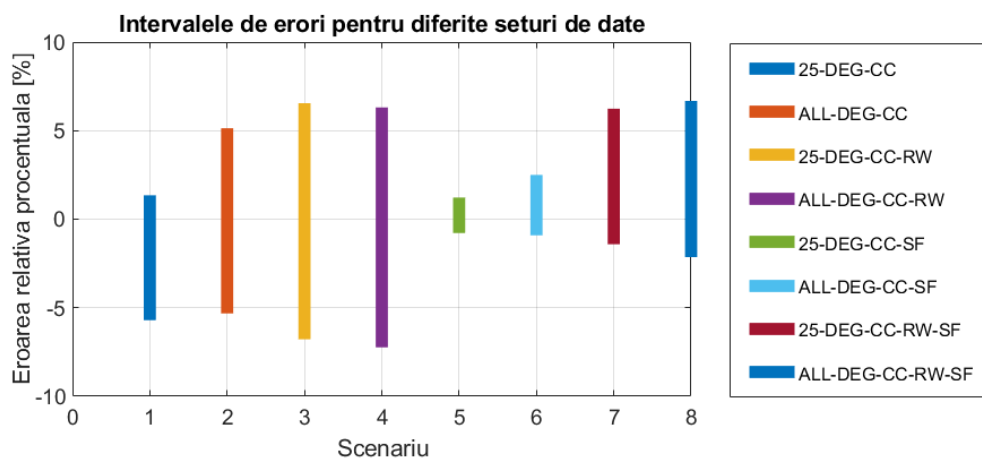


Fig. 10 Eroare relativă procentuală pentru cele 8 scenarii de evaluare semnificative

Datele au fost obținute prin rularea algoritmului în mediul de simulare Keras în versiunea 2.3 și tensorflow 2.1, folosind limbajul de programare Python, pe un PC cu procesor Intel core i7- 6700HQ, 2.6GHz, 8G RAM.

### 7.3.1. Evaluarea performanțelor pentru scenariul 25-DEG-CC

Pentru acest scenariu de test eroarea procentuală relativă de estimare se regăsește în intervalul [-5.7%, 1.35%]. Eroarea se încadrează în limitele uzuale întâlnite pentru metode bazate pe rețele neuronale care se află în intervalul [-5%, 5%]. Acest lucru se datorează faptului că la temperatura ambiantă de 25 grade, curbele de descărcare sunt destul de similare și nu prezintă neliniarități ce nu pot fi prezise cu o eroare inacceptabilă.

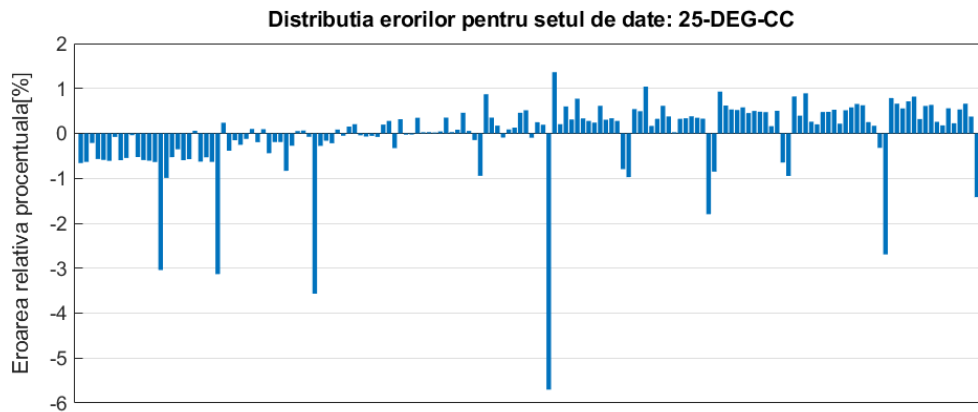


Fig. 11 Distribuția erorii de estimare în cadrul scenariului 25-DEG-CC

### 7.3.2. Evaluarea performanțelor pentru scenariul ALL-DEG-CC

În acest scenariu am adăugat curbe de descărcare la curent constant la toate temperaturile disponibile. În acest caz se observă că eroarea de estimare se situează în intervalul  $[-5.32\%, 5.13\%]$ . Se observă o creștere a erorii de estimare datorită neomogenității curbelor de descărcare pentru diferitele temperaturi luate în considerare.

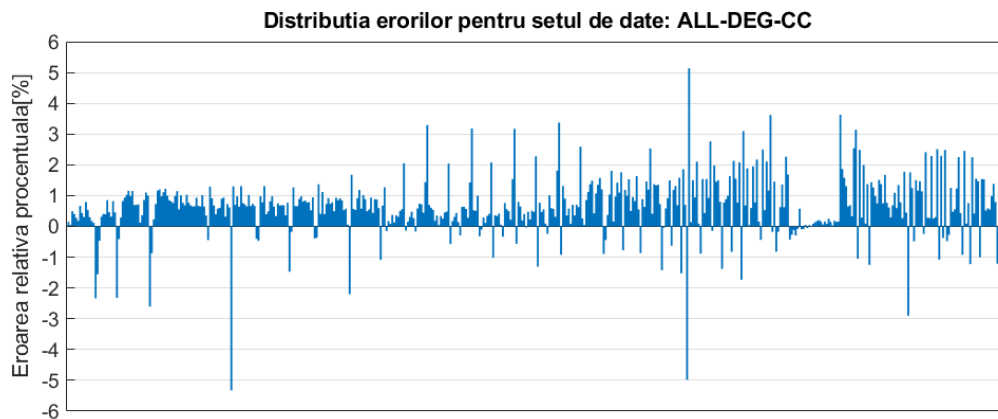


Fig. 12 Distribuția erorii de estimare în cadrul scenariului ALL-DEG-CC

### 7.3.3. Evaluarea performanțelor pentru scenariul 25-DEG-CC-RW

Acest scenariu ia în considerare datele corespunzătoare descărcării cu curent constant la temperatura ambiantă de 25 de grade precum și datele corespunzătoare descărcării bateriilor cu curent variabil. Eroare de estimare se situează în intervalul  $[-6.78\%, 6.55\%]$ . Se observă o creștere a erorii față de setul de date obținut prin

descărcarea la curent constant datorită neliniarităților introduse de procesul de descărcare la curent variabil.

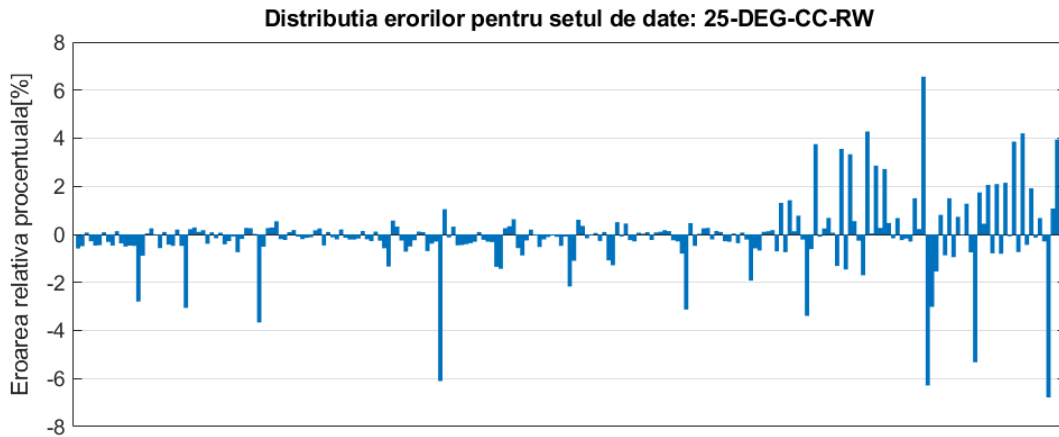


Fig. 13 Distribuția erorii de estimare în cadrul scenariului 25-DEG-CC-RW

#### 7.3.4. Evaluarea performanțelor pentru scenariul ALL-DEG-CC-RW

Setul de date conține date aferente curbelor de descărcare la curent constant și curent variabil la toate temperaturile disponibile. În acest caz degradarea performanțelor de estimare se datorează atât introducerii neliniarităților produse de procesul de descărcare al bateriilor prin curent variabil cât și neomogenitățile introduse de utilizarea datelor pentru temperaturi diferite. Eroarea de estimare se află în intervalul  $[-7.24\%, 6.31\%]$ .

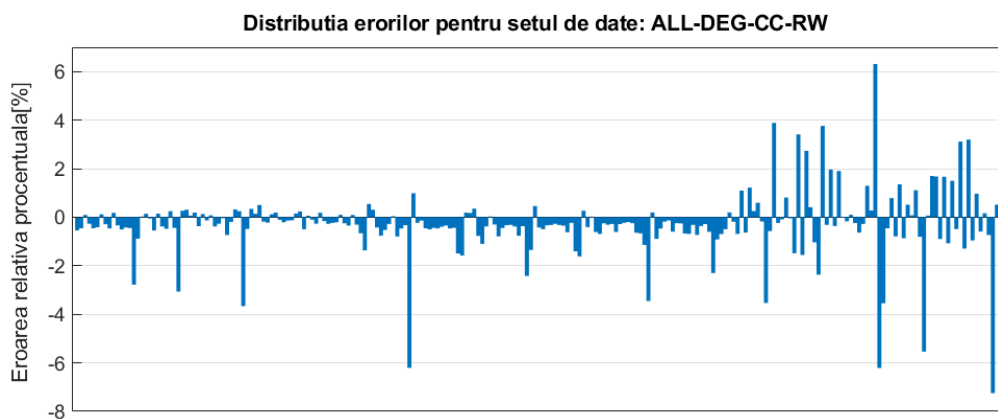


Fig. 14 Distribuția erorii de estimare în cadrul scenariului ALL-DEG-CC-RW

### 7.3.5. Evaluarea performanțelor pentru scenariul 25-DEG-CC-SF

O îmbunătățire considerabilă a erorilor de estimare se observă prin folosirea unui filtru de tip treaptă descris în secțiunea 5. Acest tip de filtru elimină neliniaritățile introduse de fenomenul de regenerare a capacității care se regăsește la unele tipuri de baterii, ca cel folosit în cazul de față. Prin intermediul introducerii filtrului se obține un set de curbe de al capacității strict descrescătoare. Eroare de estimare se regăsește în acest caz în intervalul  $[-0.78\%, 1.22\%]$ . De asemenea datorită modului de operare al fitrului se observă multiple puncte cu aceeași eroare de estimare.

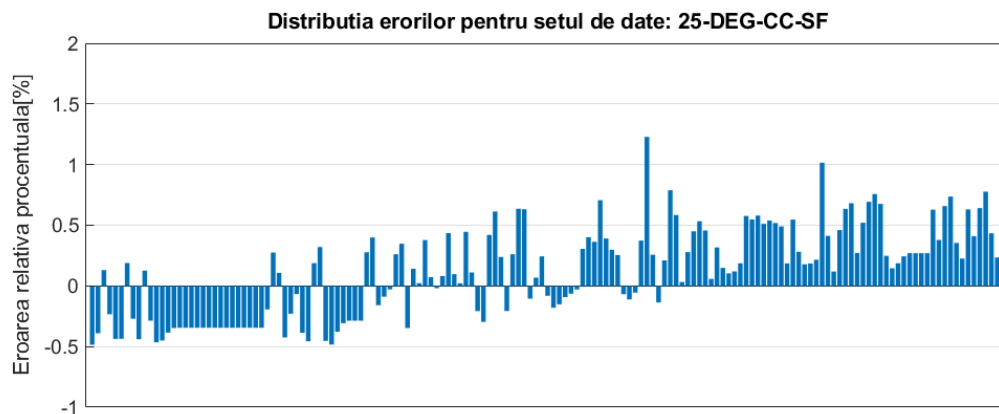


Fig. 15 Distribuția erorii de estimare în cadrul scenariului 25-DEG-CC-SF

### 7.3.6. Evaluarea performanțelor pentru scenariul ALL-DEG-CC-SF

Introducerea filtrului de tip treaptă reduce cu eroare de estimare cu mai mult de 3% față de cazul în care datele nu sunt filtrate. Se obține un interval de eroare situat între limitele de  $-0.91\%$  și  $2.50\%$ .

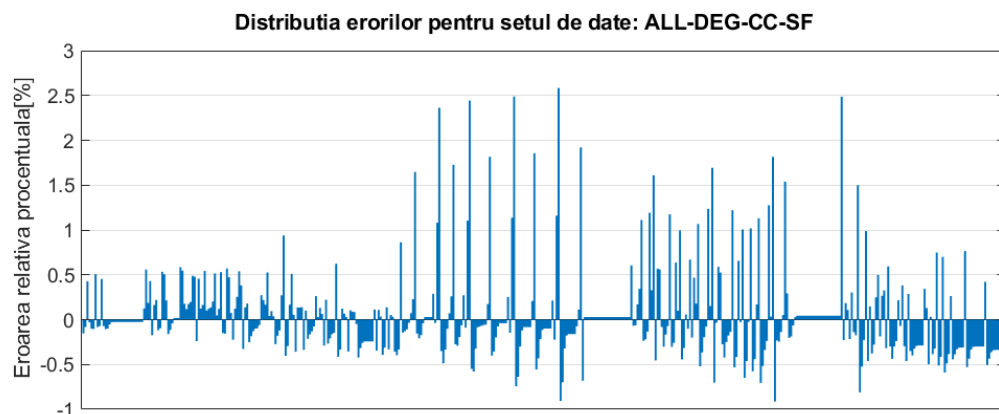


Fig. 16. Distribuția erorii de estimare în cadrul scenariului ALL-DEG-CC-SF

### 7.3.7. Evaluarea performanțelor pentru scenariul 25-DEG-CC-RW-SF

Introducerea filtrului de tip treaptă ajută într-o măsură mai mică în acest scenariu, neliniariitățile introduse de descărcarea cu curent variabil și neomogenitățile datorate diferitelor temperaturi păstrându-se și în datele filtrate. În acest caz eroare de estimare se află în intervalul -1.41% și 6.24%.

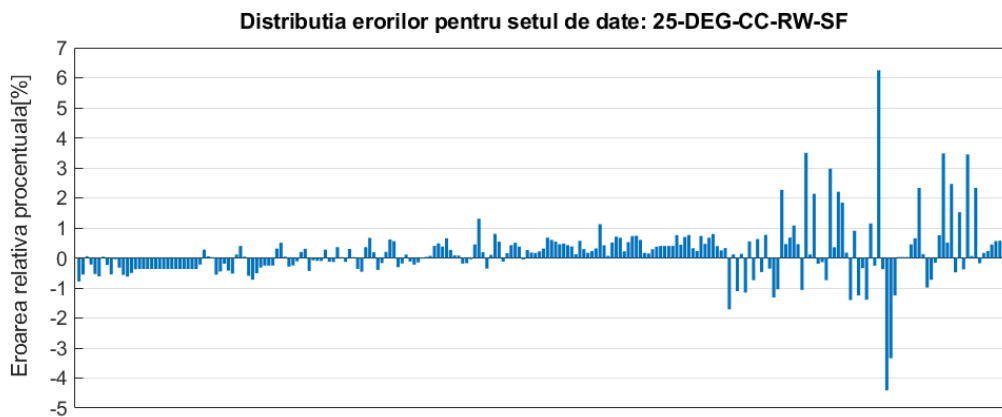


Fig. 17 Distribuția erorii de estimare în cadrul scenariului 25-DEG-CC-RW-SF

### 7.3.8. Evaluarea performanțelor pentru scenariul ALL-DEG-CC-RW-SF

Prin introducerea filtrului de tip treaptă în acest caz se obține un interval pentru eroarea de estimare care se apropie de intervalul de referință de -5%, 5% considerat uzual pentru acest tip de metodă de estimare.

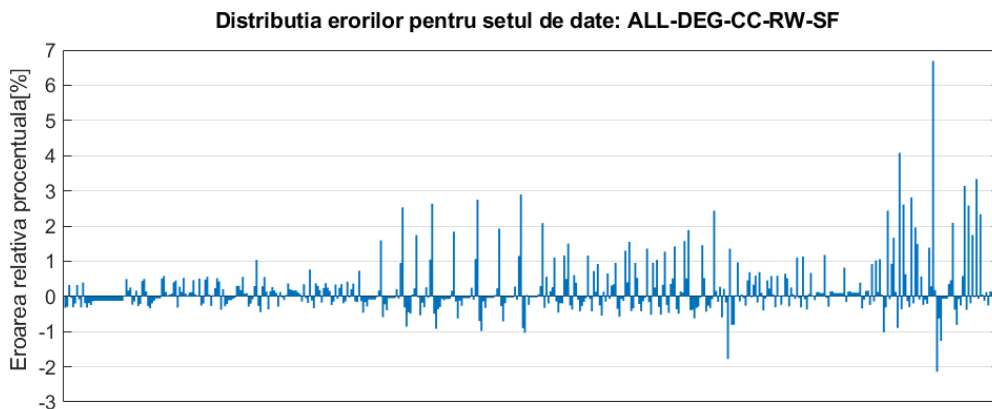


Fig. 18 Distribuția erorii de estimare în cadrul scenariului ALL-DEG-CC-RW-SF



### 7.3.9. Analiza repetibilității pentru scenariul ALL-DEG-CC-RW-SF

Repetabilitatea este una din metricile importante folosite pentru a mă asigura că metoda implementată este una stabilă și nu suferă variații importante de la o rulare la alta. În acest context am executat un număr de 10 repetiții consecutive, reprezentând rezultatele în tabelul de mai jos. Pentru valoarea maximă superioară a erorii am obținut +6.79%, respectiv pentru valoarea maximă inferioară de -3.21%. Deviația maximă este de aproximativ 1% față de rezultatul de referință, fiind o valoare acceptabilă.

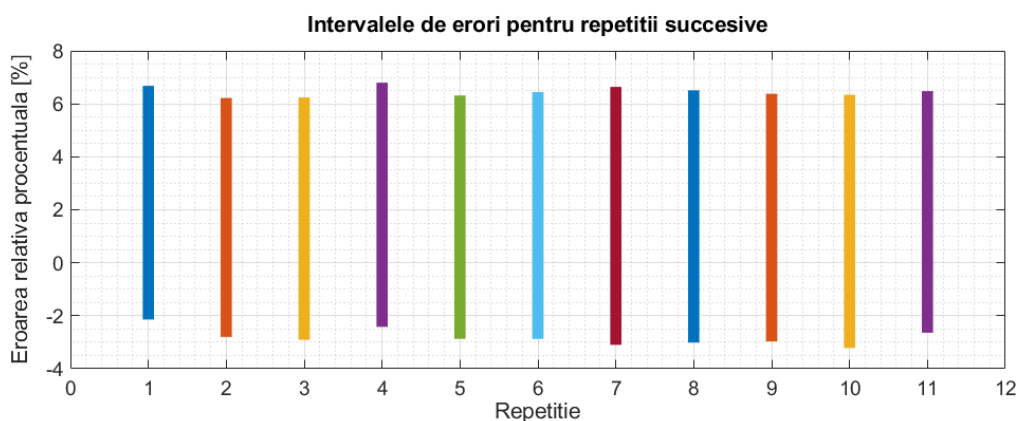


Fig. 19 Analiza repetibilității pentru scenariul ALL-DEG-CC-RW-SF

### 7.3.10. Analiza overfit/underfit pentru scenariul ALL-DEG-CC-RW-SF

Prin overfit se înțelege caracteristica modelului de a obține o bună generalizare strict pentru datele de învățare, iar pentru cele de test performanțe de estimare slabe. În sensul opus, discutăm de underfit, care este caracteristica modelului de a obține o slabă generalizare pentru datele de învățare precum și pentru cele de test.

În figura de mai jos se observă cum eroarea medie pătrată în cazul pasului de învățare urmărește trendul erorii medii pătrare obținute în cazul datelor de test. Acest lucru scoate în evidență o bună generalizare obținută prin rularea modelului.

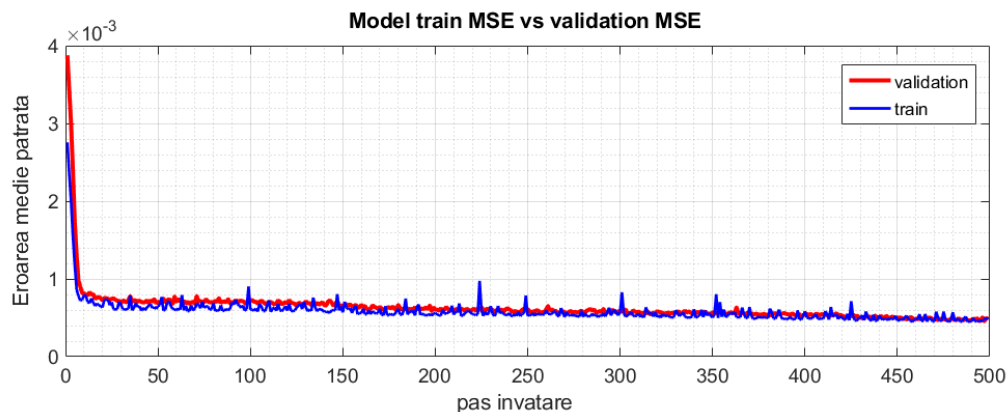


Fig. 20 Analiza overfitt/underfitt pentru scenariul ALL-DEG-CC-RW-SF

### 7.3.11. Analiza structurii rețelei neuronale folosind scenariul ALL-DEG-CC-RW-SF

Pentru determinarea numărului optim de nivele ascunse respectiv numărul de celule pe nivel am derulat o serie de configurații ce conțin combinații ale acestor parametri, urmărind intervalul de eroare relativă procentuală. Tabelul de mai jos sumarizează rezultatele obținute.

Din punct de vedere al structurii rețelei neuronale se poate concluziona că alegerea configurației cu un număr de 2 nivele ascunse și 50 de celule per nivel oferă rezultate bune, introducerea încă unui nivel ascuns neaducând o îmbunătățire semnificativă. De asemenea această configurație conferă posibilitatea implementării pe sisteme încorporate cu memorie RAM și FLASH redusă. Introducerea unui nivel ascuns adițional ar face ca stocarea parametrilor în memoria FLASH de 1MB să nu mai fie posibilă.

**Tabel 5** Analiza structurii rețelei neuronale

Nr. nivele ascunse	Nr celule pe nivel	Interval eroare relativă procentuală
1	50	[-3,33, 6,94]
2	10	[-3.35, 6.96]
2	25	[-3.04, 6.82]
2	50	[-2.14, 6.68]
2	100	[-2.31, 6.7]
3	50	[-2.21, 6.49]

### 7.3.12. Analiza crossvalidare folosind scenariul ALL-DEG-CC-RW-SF

Crossvalidarea sau validarea încrucișată este de asemenea o metrică pentru analiza calității procesului de învățare. De asemenea această metrică poate să confere o imagine de ansamblu asupra gradului de omogenitate a setului de date. Datorită tipului de estimare, și anume a seriilor de timp, am folosit ca și metrică

pătratul coeficientului de corelație  $r$ , întâlnit și sub denumirea de coeficient de determinare.

Pentru analiza validării încrucișate am folosit 10 subdivizări a setului de date. Tabelul următor sintetizează valorile coeficientului de determinare pentru cele 10 rulări:

**Tabel 6.** Valoarea parametrului de corelație pentru pașii de crossvalidare

Pas crossvalidare	Valoare $r^2$
1	0.9949
2	0.9981
3	0.9977
4	0.9974
5	0.9975
6	0.9983
7	0.9976
8	0.9975
9	0.9963
10	0.9982

Pentru exemplificare am afișat graficul corespunzător coeficientului de determinare pentru pasul 1 în cadrul validării încrucișate.

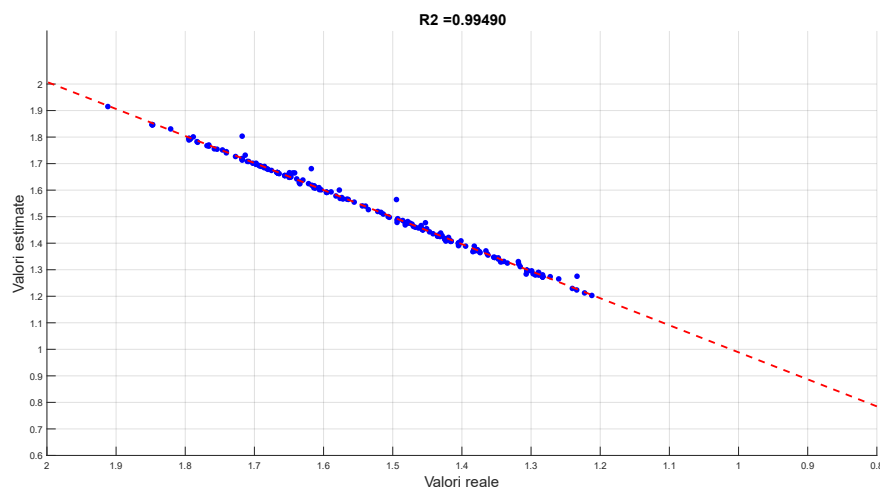


Fig. 21 Parametru de corelație pentru pasul 1 de crossvalidare

### 7.3.13. Evaluarea implementării metodei pe sisteme încorporate

Pentru evaluarea implementării metodei pe sisteme încorporate am ales setul de platforme fizice (vezi Fig. 46) ale căror specificații sunt sintetizate în tabelul următor:

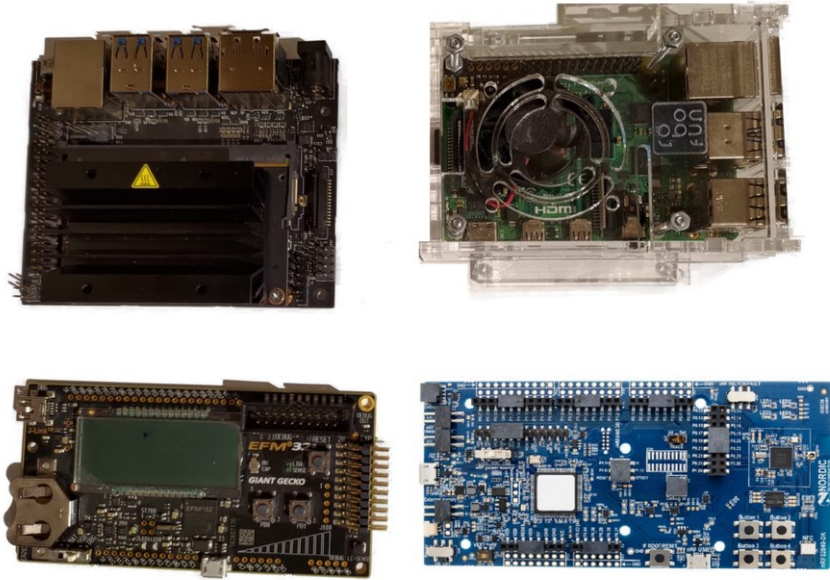


Fig. 22 Platforme fizice folosite

**Tabel 7.** Lista platformelor HW și configurarea SW pentru evaluarea implementării metodei pe sisteme încorporate

Platforma	EFM32GG	Nordic nRF52840	Raspberry pi 4	NVidia jetson nano
Hardware	Silicon Labs EFM32GG990F1024 48 MHz ARM Cortex M3 32 bit 128 KB RAM 1MB flash	Nordic semiconductor nRF52840 64MHz ARM Cortex M4 32 bit cu unitate FPU 256 KB RAM 1 MB flash	Broadcom BCM2711 Quad core 1.5 GHz 64 bit ARM Cortex A-72 4GB LPDDR4-3200 SDRAM	Nvidia GPU 128 – core Maxwell Quas core ARM A57 1.43 GHz 4GB LPDDR4 SDRAM
Software	Mbed OS	Mbed OS	Raspbian	Ubuntu Linux
Compiler / translator	gcc 7.2.1 arm-none- eabi	gcc 7.2.1 arm-none- eabi	gcc 8.3.0 arm-linux ginueabi / python 3.6 + tensorflow 2.1	gcc 8.3.0 arm-linux ginueabi / python 3.6 + tensorflow gpu 2.2

Am folosit două implementări ale metodei: cea bazată pe platforma PnP (C/C++) și cea bazată pe python (Keras) și tensorflow. Ca și criterii de evaluare am ales timpul de execuție, memoria ocupată, consum și eroarea de estimare. Rezultatele complete se pot regăsi în tabelul de mai jos:

**Tabel 8.** Evaluarea implementării metodelor pe sisteme încorporate

Platformă sistem / Implementare	PnP			
	Eroare estimare	Timp estimare [s]	Memorie utilizată [bytes]	Consum energie
EFM32GG	PC	24.537 s	RAM: 125400 FLASH: 291916	0.0825A 5.06V
nRF52840	PC	3.925 s	RAM: 132664 FLASH: 286456	0.107A 5.04V
Raspberry Pi4	PC	0.1042	.text 13392 .data 2464 .rodata 251746 .bss 126145	0.477A 5.02V
Nvidia jetson nano	PC	0.1397 s	.text 14220 .data 2712 .rodata 251939 .bss 127329	0.827A 5.03V
Platformă sistem / Implementare	Keras (tensorflow)			
	Eroare estimare	Timp estimare [s]	Memorie utilizată [bytes]	Consum energie
EFM32GG	N/A	N/A	N/A	N/A
nRF52840	N/A	N/A	N/A	N/A
Raspberry Pi4	PC	0.1042	.text 13392 .data 2464 .rodata 251746 .bss 126145	0.477A 5.02V
Nvidia jetson nano	PC	3 (prima estimare)  0.049 (restul estimărilor)	Max. 29853999	1.02 A 5.04 V

#### 7.4. Discutarea rezultatelor și studiul comparativ cu alte metode

Voi începe discutarea rezultatelor prin prezentarea concluziilor recente referitoare la performanțele de estimare ale metodelor existente în termeni de eroare maximă și minimă 0. Acestea se pot regăsi în figura 45

Metodele bazate pe filtru Kalman și derivatele acestora se caracterizează printr-o eroare de estimare de maxim 5% și minim 2.5%. Aceste rezultate au fost obținute în marea lor majoritate pe medii de simulare pe PC, neexistând o corelație clară cu resursele utilizate și posibilitatea implementării în sisteme reale de management al bateriilor.

Un interval mai larg de erori îl prezintă metodele bazate pe observatori (modele matematice). Erorile de estimare se încadrează între maxim 10% și minim 2.5%. Într-un mod asemănător se comportă și metodele bazate pe regresie în sensul celor mai mici pătrate (polinomială, exponențială), cele din urmă prezentând avantajul resurselor computaționale reduse și totodată facilitatea implementării în sisteme încorporate.

Conform 0 metodele bazate pe logica fuzzy prezintă o eroare de estimare situată în intervalul [2.5%, 5%]. Deși această metodă nu impune un efort computațional substanțial, eroarea de estimare depinde mult de modelul ales și de setul de date, cel mai important dezavantaj fiind cel al necesității unui set de date mult mai mare ca cel necesar metodelor bazate pe rețele neuronale

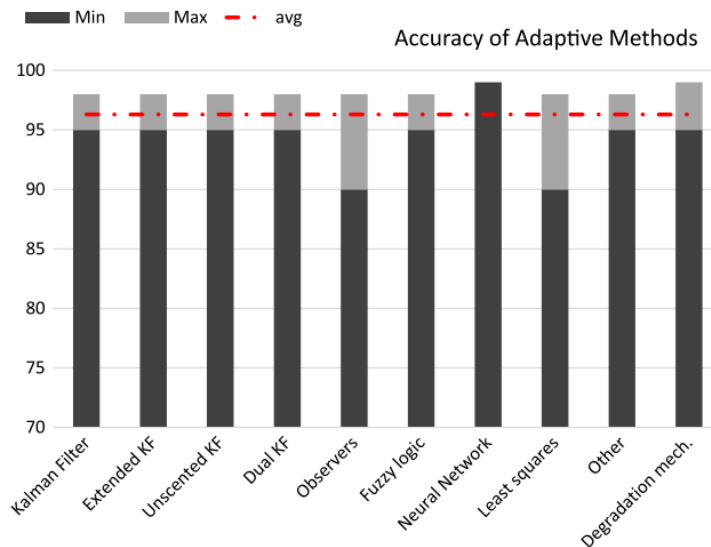


Fig. 23 Acuratețea estimării pentru diferite metode de estimare a stării de degradare a bateriilor 0

Metodele bazate pe ciclul de viață (de degradare) al bateriei prezintă o acuratețe mai mare, eroarea de estimare se situează în intervalul [1.5%, 5%]. În general aceste metode se bazează pe analiza datelor de laborator obținute offline în urma ciclării intensive a bateriilor în diferite condiții.

Metodele bazate pe rețele neuronale prezintă cea mai mare acuratețe conform 0. Eroarea acestor metode se situează în jurul valorii de 1% – 1.5%. Aceste valori au fost obținute în marea majoritate a lor prin folosirea unor seturi de date rezultate din ciclarea bateriilor la temperatura ambiantă de 25 de grade Celsius, încărcarea și descărcarea făcându-se cu un curent constant. Acest lucru l-am confirmat prin rularea scenariului experimental 25-DEG-CC-SF. Totuși prin utilizarea unui set de date ce conține scenarii mai aproape de utilizare reală, cu curenți de descărcare variabili și multiple valori pentru temperatura de operare, am obținut un interval de eroare ce se apropie de valoarea mediană a erorii de estimare a celor mai des întâlnite metode de estimare.

Comparativ cu alte metode bazate pe rețele neuronale cu memorie [3] cum ar fi LSTM, metoda bazată pe celule GRU are avantajul că prezintă o structură mai simplă a celei și de aceea numărul parametrilor modelului de estimare este cu 25% mai redus decât cel al LSTM. Acest lucru este important atât pentru procesul de învățare cât și la implementarea metodei în cadrul unui sistem încorporat cu resurse computaționale reduse.

Tabelul următor sumarizează comparația modelului GRU cu LSTM

**Tabel 9.** Comparație între rețele neuronale LSTM și GRU [3]

Criteria	LSTM	GRU
Nr. de parametri model	30651	23001
Eroare medie absolută învățare	0.270	0.264
Interval eroare relativă procentuală pt. ALL-DEG-CC	[-5.02, 5.24]	[-5.13, 5.32]

Tot în [3] am folosit un set de date asemănător cu ALL\_DEG\_CC, în care am considerat limita bateriei degradate între 60% și 70%. Folosind acest set de date am comparat GRU cu cele mai cunoscute metode bazate pe rețele neuronale, precum și metodele bazate pe regresie în sensul celor mai mici pătrate. Rezultatele sunt sumarizate în figura de mai jos, pentru diferite baterii selectate.

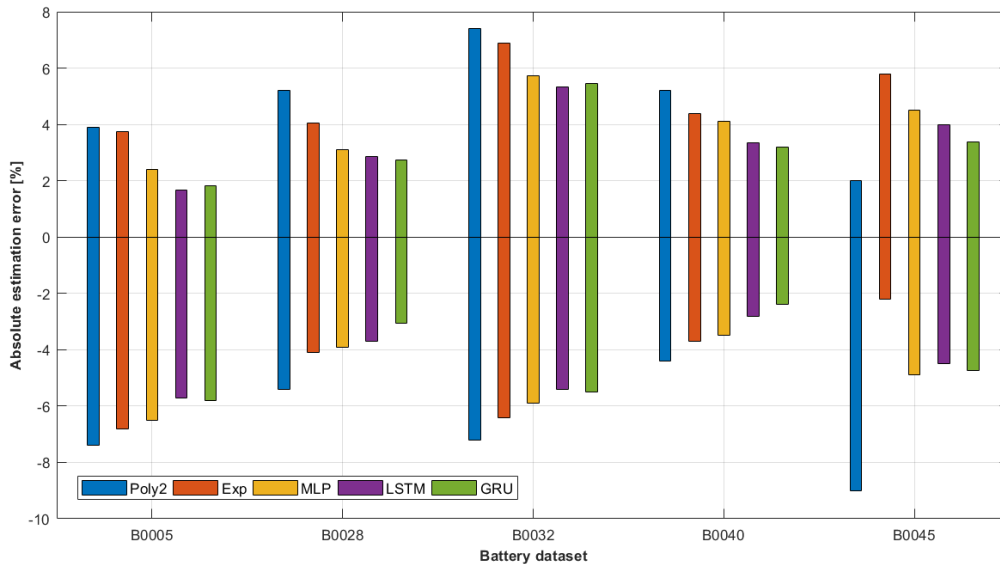


Fig. 24 Studiu comparativ a principalelor metode de estimare cu implementare pe sisteme încorporate [3]

Metodele bazate pe regresie polinomială și exponențială prezintă erori de estimare bune în cazul în care curba capacității urmărește graficul unei funcții polinomiale sau exponențiale. În majoritatea cazurilor, neliniaritățile caracteristice capacității bateriei fac ca eroarea de estimare să crească, de exemplu la un interval de eroare [-7.5%, 4%] respectiv [-7%, 3.8%] pentru bateria B0005. Metodele bazate pe rețele neuronale prezintă o acuratețe de estimare mult mai bună: începând cu MLP, pentru care eroarea de estimare este între -6.5% și 2.5% și continuând cu LSTM și GRU pentru care eroare se situează în intervalul [-5.5%, 2%] pentru bateria B0005, aproape la jumătate ca primele metode amintite. Îmbunătățirea erorii de estimare se datorează structurii celulei rețelelor neuronale

recurente LSTM și GRU care folosesc memorie pe termen lung, respectiv pe termen scurt.

Când capacitatea bateriei prezintă multiple puncte de regenerare, acuratețea de estimare scade și pentru LSTM și GRU, iar diferența dintre metode nu mai e așa de semnificativă. Acest lucru se poate observa la bateria B0032. Chiar și așa, eroarea de estimare se află într-un interval acceptabil.



## 8. CONCLUZII ȘI PERSPECTIVE

### 8.1. Concluzii

Teza de față prezintă activitatea de cercetare și dezvoltare efectuată în cadrul programului de cercetare doctorală "Evaluarea stării de degradare a bateriilor reîncărcabile în sisteme alimentate cu baterii" realizată de către mine sub conducerea științifică a domnului prof. dr. ing. Mihai V. Micea.

Lucrarea abordează probleme curente din domeniul bateriilor și a sistemelor încorporate cu resurse limitate de procesare.

Principalele obiective ale tezei sunt descrise în secțiunea 1.3:

- Definirea unor metode de estimare a stării de degradare al bateriilor reîncărcabile care să prezinte un nivel ridicat de performanță și de eficiență, adică:
  - o acuratețe de estimare cât mai ridicată în ceea ce privește SoH, respectiv
  - să poată fi implementate și executate pe platforme digitale încorporate, cu resurse limitate (de memorie, de procesare, consum de energie, etc.).
- Definirea unui cadru de evaluare a performanțelor metodelor de estimare a SoH, care să cuprindă:
  - criterii de precizie și acuratețe,
  - criterii de evaluare a complexității algoritmilor,
  - criterii de evaluare a resurselor de microprocesor utilizare precum și a consumului propriu de energie al algoritmului.

Primul obiectiv a fost atins prin implementarea metodei de estimare a stării de degradare a bateriilor reîncărcabile descrise la capitolul 5, bazate pe rețele neuronale cu celule GRU. În plus am definit și o optimizare a acestei metode pentru tipuri de baterii ce prezintă efectul de regenerare a capacității. De menționat este faptul că metoda descrisă la capitolul 5 este o îmbunătățire a metodei implementate tot de mine și a cărei specific este descris la capitolul 4, metoda bazată pe regresie polinomială. Metoda descrisă la capitolul 6 se poate implementa pe sisteme încorporate de la cele mai simple ajungând la cele mai complexe în funcție de platforma software aleasă.

Cel de-al doilea obiectiv a fost atins prin descrierea și implementarea cadrului de evaluare a performanței metodelor de estimare a stării de degradare a bateriilor de la capitolul 6. Acest sistem cadru de evaluare a performanței a fost dezvoltat având ca scop obținerea unor metrici prin care să se poată analiza și categorisi orice metodă de estimare a stării de degradare a bateriilor, cu precădere cele bazate pe rețele neuronale. De asemenea în cadrul metodei descrise la capitolul 4 am implementat atât hardware cât și software un BMS complet.

## 8.2. Rezumat al contribuțiilor

Lista completă a contribuțiilor din această lucrare este prezentată mai jos:

- Capitolul 3: am realizat o analiză complexă a principalelor metode de evaluare a stării de degradare a bateriilor scoțând în evidență avantajele și dezavantajele fiecăreia.
- Capitolul 4: Am prezentat descrierea și implementarea unei metode simple și eficiente de estimare a stării de degradare a bateriilor bazate pe regresii polinomiale de ordin 2.
- Capitolul 5: Am detaliat metoda de estimare a stării de degradare a bateriilor bazate pe rețele neuronale recurente de tip GRU. În plus am descris optimizarea necesară pentru baterii care prezintă efectul de regenerare a capacității.
- Capitolul 6: Am descris și implementat un sistem cadru pentru evaluarea performanțelor metodelor de estimare a bateriilor în vederea posibilității comparării acestora.
- Capitolul 7: Am utilizat sistemul cadru descris la capitolul 6 pentru analiza performanțelor metodei de evaluare a stării de degradare a bateriilor în diverse implementări pe sisteme încorporate. În plus am luat în considerare prezentarea unui studiu comparativ cu alte metode existente.

## 8.3. Perspective de cercetare și dezvoltare

Din lucrarea de față se pot evidenția mai multe direcții de dezvoltare ulterioare. Acestea se pot grupa în funcție de obiectivele prezentei teze. În ceea ce privește metode de estimare a stării de degradare a bateriilor se pot distinge o serie de îmbunătățiri:

- Transformarea modelului într-un model dinamic de actualizare a parametrilor – rețeaua neuronală prezentată are parametri ficși calculați pe baza datelor din cadrul procesului de antrenare. Datorită specificului metodei de a rula în timp real este benefic ca acești parametri să poată fi actualizați cu fiecare ciclu de baterie introdus prin utilizarea într-un sistem de management al bateriei.
- Folosirea metodei prezentate împreună cu alte metode mai simple sau complexe în vederea obținerii unui model mai performant.

Ținând cont de cel de-al doilea obiectiv, direcțiile de dezvoltare ulterioară pot fi:

- Lărgirea plajei de sisteme încorporate pe care se poate face evaluarea.
- Adăugarea de suport pentru mai multe biblioteci software de procesare precum și platformelor software
- Definirea unui scor prin intermediul căruia să poată fi comparate metodele.

## REFERINȚE BIBLIOGRAFICE

- [1] L. Ungurean, G. Cârstoiu, M. V. Micea, and V. Groza, 'Battery state of health estimation: a structured review of models, methods and commercial devices', *Int J Energ Res Int J Energ Res*, vol. 41, no. 2, pp. 151-181, Feb. 2017
- [2] V. Pop, H. J. Bergveld, D. Danilov, P. P. L. Regtien, and P. H. L. Notten, *Battery Management Systems. Accurate State-of-Charge Indication for Battery-Powered Applications*. Springer Netherlands, 2008.
- [3] L. Ungurean, M. V. Micea, and G. Cârstoiu, 'Online state of health prediction method for lithium-ion batteries, based on gated recurrent unit neural networks', *International Journal of Energy Research*, vol. 44, no. 8, pp. 6767-6777, Apr. 2020, doi: 10.1002/er.5413.
- [4] Z. Guo, X. Qiu, G. Hou, B.Y. Liaw, C. Zhang, "State of Health Estimation for Lithium Ion Batteries Based on Charging Curves", *J. Power Sources*, 249, 2014, (457-462).
- [5] M.V. Micea, L. Ungurean, G.N. Carstoiu, V. Groza, "Online State-of-Health Assessment for Battery Management Systems", *IEEE Trans. Instrum. Meas.*, 60 (6), 2011, (1997-2006).
- [6] W. Waag, C. Fleischer, D.U. Sauer, "Adaptive On-Line Prediction of the Available Power of Lithium-Ion Batteries", *J. Power Sources*, 242, 2013, (548-559).
- [7] A.H. Ranjbar, A. Banaei, A. Khoobroo, B. Fahimi, "Online Estimation of State of Charge in Li-Ion Batteries Using Impulse Response Concept", *IEEE Trans Smart Grid*, 3 (1), 2012, (360-367).
- [8] K.M. Tsang, L. Sun, W.L. Chan, "Identification and Modelling of Lithium Ion Battery", *Energ. Convers. Managem.*, 51, 2010, (2857-2862).
- [9] Y. Xing, W. He, M. Pecht, K.L. Tsui, "State of Charge Estimation of Lithium-Ion Batteries Using the Open-Circuit Voltage at Various Ambient Temperatures", *Applied Energ.*, 113, 2014, (106-115).
- [10] J. Remmlinger, M. Buchholz, et al., "State-of-Health Monitoring of Lithium-Ion Batteries in Electric Vehicles by On-Board Internal Resistance Estimation", *J. Power Sources*, 196, 2011, (5357-5363).
- [11] Z. Chen, C.C. Mi, Y. Fu, J. Xu, X. Gong, "Online Battery State of Health Estimation Based on Genetic Algorithm for Electric and Hybrid Vehicle Applications", *J. Power Sources*, 240, 2013, (184-192).
- [12] S.E. Li, B. Wang, H. Peng, X. Hu, "An Electrochemistry-Based Impedance Model for Lithium-Ion Batteries", *J. Power Sources*, 258, 2014, (9-18).
- [13] G.K. Prasad, C.D. Rahn, "Model Based Identification of Aging Parameters in Lithium Ion Batteries", *J. Power Sources*, 232, 2013, (79-85).
- [14] M. Ecker, J.B. Gerschler, J. Vogel, S. Kabitz, F. Hust, P. Dechent, D.U. Sauer, "Development of a Lifetime Prediction Model for Lithium-Ion Batteries Based on Extended Accelerated Aging Test Data", *J. Power Sources*, 215, 2012, (248-257).

- [15] L.C. Stevanatto, V.J. Brusamarello, S. Tairov, "Parameter Identification and Analysis of Uncertainties in Measurements of Lead-Acid Batteries", *IEEE Trans. Instrum. Meas.*, 63 (4), 2014, (761-768).
- [16] A. Eddahech, O. Briat, N. Bertrand, J.Y. Deletage, J.M. Vinassa, "Behavior and State-of-Health Monitoring of Li-Ion Batteries Using Impedance Spectroscopy and Recurrent Neural Networks", *Electr. Power Energ. Syst.*, 42, 2012, (487-494).
- [17] S.J. Moura, N.A. Chaturvedi, M. Krstic, "PDE Estimation Techniques for Advanced Battery Management Systems - Part II: SOH Identification", in *Proc. IEEE ACC*, 2012, (566-571).
- [18] S.J. Moura, N.A. Chaturvedi, M. Krstic, "PDE Estimation Techniques for Advanced Battery Management Systems - I: SOC Estimation", in *Proc. IEEE ACC*, 2012, (559-565).
- [19] S. Santhanagopalan, R.E. White, "State of Charge Estimation Using an Unscented Filter for High Power Lithium Ion Cells", *Int. J. Energ. Research*, 34, 2010, (152-163).
- [20] A. Eddahech, O. Briat, J.M. Vinassa, "Determination of Lithium-Ion Battery State-of-Health Based on Constant-Voltage Charge Phase", *J. Power Sources*, 258, 2014, (218-227).
- [21] C. Weng, Y. Cui, J. Sun, H. Peng, "On-Board State of Health Monitoring of Lithium-Ion Batteries Using Incremental Capacity Analysis With Support Vector Regression", *J. Power Sources*, 235, 2013, (36-44).
- [22] C. Weng, J. Sun, H. Peng, "A Unified Open-Circuit-Voltage Model of Lithium-Ion Batteries for State-of-Charge Estimation and State-of-Health Monitoring", *J. Power Sources*, 258, 2014, (228-237).
- [23] Y. Xing, E.W.M. Ma, K.L. Tsui, M. Pecht, "An Ensemble Model for Predicting the Remaining Useful Performance of Lithium-Ion Batteries", *Microelectron. Reliab.*, 53 (6), 2013, (811-820).
- [24] X. Hu, S.E. Li, Z. Jia, B. Egardt, "Enhanced Sample Entropy-Based Health Management of Li-Ion Battery for Electrified Vehicles", *Energy*, 64, 2014, (953-960).
- [25] M. Mastali, J. Vazquez-Arenas, R. Fraser, M. Fowler, S. Afshar, M. Stevens, "Battery State of the Charge Estimation Using Kalman Filtering", *J. Power Sources*, 239, 2013, (294-307).
- [26] N. Omar, M.A. Monem, et al., "Lithium Iron Phosphate Based Battery - Assessment of the Aging Parameters and Development of Cycle Life Model", *Applied Energ.*, 113, 2014, (1575-1585).
- [27] Y. Xing, K.L. Tsui, Q. Miao, M. Pecht, "Prognostics and Health Monitoring for Lithium-Ion Battery", in *Proc. IEEE ISI*, 2011, (242-247).
- [28] J.A.M. Penna, C.L. Nascimento Jr., L.R. Rodrigues, "Health Monitoring and Remaining Useful Life Estimation of Lithium-Ion Aeronautical Batteries", in *Proc. IEEE AERO*, 2012, (1-12).
- [29] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, 'Statistical and Machine Learning forecasting methods: Concerns and ways forward', *PLOS ONE*, vol. 13, no. 3, p. e0194889, Mar. 2018, doi: 10.1371/journal.pone.0194889.

- [30] M. T. Hagan, H. B. Demuth, M. H. Beale, and O. De Jesus, *Neural network design*, 2nd edition, 2016.
- [31] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Cambridge, Massachusetts: The MIT Press, 2016.
- [32] F. A. Gers, J. Schmidhuber, and F. Cummins, 'Learning to forget: continual prediction with LSTM', in 1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470), Sep. 1999, vol. 2, pp. 850–855 vol.2, doi: 10.1049/cp:19991218.
- [33] S. Hochreiter and J. Schmidhuber, 'Long Short-Term Memory', *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
- [34] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, 'LSTM: A Search Space Odyssey', *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017, doi: 10.1109/TNNLS.2016.2582924.
- [35] Christopher Olah, *Understanding LSTM Networks*, <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>, 2015.
- [36] R. Dey and F. M. Salem, 'Gate-variants of Gated Recurrent Unit (GRU) neural networks', in 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Aug. 2017, pp. 1597–1600, doi: 10.1109/MWSCAS.2017.8053243.
- [37] W. Waag, C. Fleischer, D.U. Sauer, "Critical Review of the Methods for Monitoring of Lithium-Ion Batteries in Electric and Hybrid Vehicles", *J. Power Sources*, 258, 2014, (321–339).
- [38] L. Lu, X. Han, J. Li, J. Hua, M. Ouyang, "A Review on the Key Issues for Lithium-Ion Battery Management in Electric Vehicles", *J. Power Sources*, 226, 2013, (272–288).
- [39] B. Price, J. Richardson, E. Dietz, "State-of-Charge and State-of-Health Monitoring: Implications for Industry, Academia, and the Consumer", in *Proc. IEEE EIT*, 2012, (1–6).
- [40] J. Zhang, J. Lee, "A Review on Prognostics and Health Monitoring of Li-Ion Battery", *J. Power Sources*, 196, 2011, (6007–6014).
- [41] X.S. Si, W. Wang, C.H. Hu, D.H. Zhou, "Remaining Useful Life Estimation: A Review on the Statistical Data Driven Approaches", *J. Power Sources*, 213, 2011, (1–14).
- [42] A. Barre, B. Deguilhem, et al., "A Review on Lithium-Ion Battery Ageing Mechanisms and Estimations for Automotive Applications", *J. Power Sources*, 241, 2013, (680–689).
- [43] K.B. Hatzell, A. Sharma, H.K. Fathy, "A Survey of Long-Term Health Modeling, Estimation, and Control of Lithium-Ion Batteries: Challenges and Opportunities", in *Proc. AACC*, 2012, (584–591).
- [44] S.M. Rezvanizani, et al., "Review and Recent Advances in Battery Health Monitoring and Prognostics Technologies for Electric Vehicle (EV) Safety and Mobility", *J. Power Sources*, 256, 2014, (110–124).

- [45] L. Liao, F. Kottig, "Review of Hybrid Prognostics Approaches for Remaining Useful Life Prediction of Engineered Systems, and an Application to Battery Life Prediction", *IEEE Trans. Reliab.*, 63 (1), 2014.
- [46] A. Seaman, et al., "A Survey of Mathematics-Based Equivalent-Circuit and Electrochemical Battery Models for Hybrid and Electric Vehicle Simulation", *J. Power Sources*, 256, 2014, (410–423).
- [47] K.S. Ng, C.S. Moo, Y.P. Chen, Y.C. Hsieh, "Enhanced Coulomb Counting Method for Estimating State-of-Charge and State-of-Health of Lithium-Ion Batteries", *Applied Energ.*, 86, 2009, (1506-1511).
- [48] B.P. Divakar, K.W.E. Cheng, et al., "Battery Management System and Control Strategy for Hybrid and Electric Vehicle", in Proc. PESA, 2009, (1-6).
- [49] A. Saxena, " Subject MI63: Kalman Filter Tank Filling", Cornell Univ., 2008, Online:  
<http://www.cs.cornell.edu/courses/cs4758/2012sp/materials/mi63slides.pdf>.
- [50] B. Rosca, J.T.B.A. Kessels, H.J. Bergveld, P.P.J. van den Bosch, "On-line Parameter, State-of-Charge and Aging Estimation of Li-ion Batteries", in Proc. IEEE VPPC, 2012, (1122-1127).
- [51] D. Andre, A. ], T. Soczka-Guth, D.U. Sauer, "Comparative Study of a Structured Neural Network and an Extended Kalman Filter for State of Health Determination of Lithium-Ion Batteries in Hybrid Electric Vehicles", *Eng. Applic. Artif. Intellig.*, 26, 2013, (951-961).
- [52] M. Daboussy, D. Chrenko, et al., "Characterisation of a Commercial Automotive Lithium Ion Battery Using Extended Kalman Filter", in Proc. IEEE ITEC, 2013, (1-6).
- [53] J. Kim, S. Lee, B.H. Cho, "Complementary Cooperation Algorithm Based on DEKF Combined With Pattern Recognition for SOC/Capacity Estimation and SOH Prediction", *IEEE Trans. Power Electron.*, 27 (1), 2012, (436-451).
- [54] T. Dragicevic, S. Sucic, J.M. Guerrero, "Battery State-of-Charge and Parameter Estimation Algorithm Based on Kalman Filter", in Proc. IEEE EuroCon, 2013, (1519–1524).
- [55] B. Pivlelait, C. Rentel, et al., "An Advanced Battery Management System for Lithium Ion Batteries", in Proc. GVSETS, 2011, (1-7).
- [56] D. Andre, C. Appel, T. Soczka-Guth, D.U. Sauer, "Advanced Mathematical Methods of SOC and SOH Estimation for Lithium-Ion Batteries", *J. Power Sources*, 224, 2013, (20–27).
- [57] E.A. Wan, R. Van Der Merwe, "The Unscented Kalman Filter for Nonlinear Estimation", in Proc. IEEE AS-SPCC, 2000, (153–158).
- [58] B. Bole, C.S. Kulkarni, M. Daigle, "Adaptation of an Electrochemistry-Based Li-Ion Battery Model to Account for Deterioration Observed Under Randomized Use", in Proc. PHM14, 2014, (1-9).
- [59] C. Barlak, Y. Ozkazanc, "A Classification Based Methodology for Estimation of State-of-Health of Rechargeable Batteries", in Proc. IEEE ELECO, 2009, (II.101–II.105).

- [60] J. Kim, B.H. Cho, "State-of-Charge Estimation and State-of-Health Prediction of a Li-Ion Degraded Battery Based on an EKF Combined With a Per-Unit System", *IEEE Trans. Vehicular Tech.*, 60 (9), 2011.
- [61] J. Remmlinger, M. Buchholz, T. Soczka-Guth, K. Dietmayer, "On-Board State-of-Health Monitoring of Lithium-Ion Batteries Using Linear Parameter-Varying Models", *J. Power Sources*, 239, 2013, (689-695).
- [62] J.D. Kozłowski, "Electrochemical Cell Prognostics Using Online Impedance Measurements and Model-Based Data Fusion Techniques", in *Proc. IEEE AERO*, 2003, (3257-3270).
- [63] H.T. Lin, T.J. Liang, S.M. Chen, "The State-of-Health Diagnosis of Li-Co Batteries with Fuzzy Identification", in *Proc. IEEE PEMCC*, 2012, (2678-2682).
- [64] K.M. Tsang, W.L. Chan, "State of Health Detection for Lithium Ion Batteries in Photovoltaic System", *Energ. Convers. Manag.*, 65, 2013, (7-12).
- [65] M. Landi, G. Gross, "Measurement Techniques for Online Battery State of Health Estimation in Vehicle-to-Grid Applications", *IEEE Trans. Instrum. Meas.*, 63 (5), 2014, (1224-1234).
- [66] A.J. Smola, B. Scholkopf, "A Tutorial on Support Vector Regression", *Statistics Comput.*, 14 (3), 2004, (199-222).
- [67] A. Widodo, M.C. Shim, W. Caesarendra, B.S. Yang, "Intelligent Prognostics for Battery Health Monitoring Based on Sample Entropy", *Expert Syst. Applic.*, 38, 2011, (11763-11769).
- [68] D. Wang, Q. Miao, M. Pecht, "Prognostics of Lithium-Ion Batteries Based on Relevance Vectors and a Conditional Three-Parameter Capacity Degradation Model", *J. Power Sources*, 239, 2013, (253-264).
- [69] B. Pattipati, C. Sankavaram, K.R. Pattipati, "System Identification and Estimation Framework for Pivotal Automotive Battery Management System Characteristics", *IEEE Trans. Syst. Man Cybernetics C*, 41 (6), 2011, (869-884).
- [70] J. Zhou, D. Liu, Y. Peng, X. Peng, "Dynamic Battery Remaining Useful Life Estimation: An On-Line Data-Driven Approach", in *Proc. IEEE I2MTC*, 2012, (2196-2199).
- [71] A. Nuhic, T. Terzimehic, et al., "Health Diagnosis and Remaining Useful Life Prognostics of Lithium-Ion Batteries Using Data-Driven Methods", *J. Power Sources*, 239, 2013, (680-688).
- [72] C. Weng, Y. Cui, J. Sun, H. Peng, "On-Board State of Health Monitoring of Lithium-Ion Batteries Using Incremental Capacity Analysis With Support Vector Regression", *J. Power Sources*, 235, 2013, (36-44).
- [73] B. Saha, K. Goebel, S. Poll, J. Christophersen, "An Integrated Approach to Battery Health Monitoring Using Bayesian Regression and State Estimation", in *Proc. IEEE Autotestcon*, 2007, (646-653).
- [74] S. Schwunk, N. Armbruster, S. Straub, J. Kehl, M. Vetter, "Particle Filter for State of Charge and State of Health Estimation for Lithium-Iron Phosphate Batteries", *J. Power Sources*, 239, 2013, (705-710).
- [75] K. Goebel, B. Saha, A. Saxena, J.R. Celaya, J.P. Christophersen, "Prognostics in Battery Health Management", *IEEE Instrum. Meas. Mag.*, 11 (4), 2008, (33-40).

- [76] B. Saha, K. Goebel, S. Poll, J. Christophersen, "Prognostics Methods for Battery Health Monitoring Using a Bayesian Framework", *IEEE Trans. Instrum. Meas.*, 58 (2), 2009, (291-296).
- [77] Y. Xing, E.W.M. Ma, K.L. Tsui, M. Pecht, "A Case Study on Battery Life Prediction Using Particle Filtering", in *Proc. IEEE PHM, 2012*, (1-6).
- [78] Y. Xing, E.W.M. Ma, K.L. Tsui, M. Pecht, "An Ensemble Model for Predicting the Remaining Useful Performance of Lithium-Ion Batteries", *Microelectron. Reliab.*, 53, 2013, (811-820).
- [79] W. Xian, B. Long, M. Li, H. Wang, "Prognostics of Lithium-Ion Batteries Based on the Verhulst Model, Particle Swarm Optimization and Particle Filter", *IEEE Trans. Instrum. Meas.*, 63 (1), 2014, (2-17).
- [80] W. He, N. Williard, M. Osterman, M. Pecht, "Prognostics of Lithium-Ion Batteries Based on Dempster-Shafer Theory and the Bayesian Monte Carlo Method", *J. Power Sources*, 196, 2011, (10314-10321).
- [81] W. He, N. Williard, M. Osterman, M. Pecht, "Remaining Useful Performance Analysis of Batteries", in *Proc. IEEE PHM, 2011*, (1-6).
- [82] S. Schwunk, N. Armbruster, S. Straub, J. Kehl, M. Vetter, "Particle filter for state of charge and state of health estimation for lithium-iron phosphate batteries", *J. Power Sources*, 239, 2013, (705-710).
- [83] D.Z. Li, W. Wang, F. Ismail, "A Mutated Particle Filter Technique for System State Estimation and Battery Life Prediction", *IEEE Trans. Instrum. Meas.*, 63 (8), 2014, (2034-2043).
- [84] G. Li, B. Li, Z. Liu, X. Chen, "Implementation and Optimization of Particle Filter Tracking Algorithm on Multi-DSPs System", in *Proc. IEEE CIS, 2008*, (152-157).
- [85] Q. Miao, L. Xie, H. Cui, W. Liang, M. Pecht, "Remaining Useful Life Prediction of Lithium-Ion Battery With Unscented Particle Filter Technique", *Microelectron. Reliab.*, 53, 2013, (805-810).
- [86] M.E. Orchard, L. Tang, G. Vachtsevanos, "A Combined Anomaly Detection and Failure Prognosis Approach for Estimation of Remaining Useful Life in Energy Storage Devices", in *Proc. PHM11, 2011*, (1-7).
- [87] C. Chen, M. Pecht, "Prognostics of Lithium-Ion Batteries Using Model-Based and Data-Driven Methods", in *Proc. IEEE PHM, 2012*, (1-6).
- [88] N. Khare, P. Singh, J.K. Vassiliou, "A Novel Magnetic Field Probing Technique for Determining State of Health of Sealed Lead-Acid Batteries", *J. Power Sources*, 218, 2012, (462-473).
- [89] X. Hu, S.E. Li, Z. Jia, B. Egardt, "Enhanced Sample Entropy-Based Health Management of Li-Ion Battery for Electrified Vehicles", *Energy*, 64, 2014, (953-960).
- [90] Y.H. Sun, H.L. Jou, J.C. Wu, K.D. Wu, "Auxiliary Health Diagnosis Method for Lead-Acid Battery", *Applied Energ.*, 87, 2010, (3691-3698).
- [91] M. Gholizadeh, F.R. Salmasi, "Estimation of State of Charge, Unknown Nonlinearities, and State of Health of a Lithium-Ion Battery Based on a Comprehensive Unobservable Model", *IEEE Trans. Ind. Electron.*, 61 (3), 2014, (1335-1344).



- [92] X. Feng, J. Li, M. Ouyang, et al., "Using Probability Density Function to Evaluate the State of Health of Lithium-Ion Batteries", *J. Power Sources*, 232, 2013, (209-218).
- [93] J. Yu, "Health Degradation Detection and Monitoring of Lithium-Ion Battery Based on Adaptive Learning Method", *IEEE Trans. Instrum. Meas.*, 63 (7), 2014, (1709-1721).
- [94] D. Liu, J. Pang, J. Zhou, Y. Peng, M. Pecht, "Prognostics for State of Health Estimation of Lithium-Ion Batteries Based on Combination Gaussian Process Functional Regression", *Microelectron. Reliab.*, 53, 2013, (832-839).
- [95] S. Tang, C. Yu, X. Wang, X. Guo, X. Si, "Remaining Useful Life Prediction of Lithium-Ion Batteries Based on the Wiener Process with Measurement Error", *Energies*, 7, 2014, (520-547).
- [96] B. Long, W. Xian, L. Jiang, Z. Liu, "An Improved Autoregressive Model by Particle Swarm Optimization for Prognostics of Lithium-Ion Batteries", *Microelectron. Reliab.*, 53, 2013, (821-831).
- [97] M. Shahriari, M. Farrokhi, "Online State-of-Health Estimation of VRLA Batteries Using State of Charge", *IEEE Trans. Ind. Electron.*, 60 (1), 2013, (191-202).
- [98] H.T. Lin, T.J. Liang, S.M. Chen, "Estimation of Battery State of Health Using Probabilistic Neural Network", *IEEE Trans. Ind. Informatics*, 9 (2), 2013, (679-685).
- [99] G. Jin, D.E. Matthews, Z. Zhou, "A Bayesian Framework for On-Line Degradation Assessment and Residual Life Prediction of Secondary Batteries in Spacecraft", *Reliab. Eng. Syst. Safety*, 113, 2013, (7-20).
- [100] S.S.Y. Ng, Y. Xing, K.L. Tsui, "A Naive Bayes Model for Robust Remaining Useful Life Prediction of Lithium-Ion Battery", *Applied Energ.*, 118, 2014, (114-123).
- [101] A. Eddahech, O. Briat, E. Woirgard, J.M. Vinassa, "Remaining Useful Life Prediction of Lithium Batteries in Calendar Ageing for Automotive Applications", *Microelectr. Reliab.*, 52, 2012, (2438-2442).
- [102] M. V. Micea, G. N. Carstoiu, L. Ungurean, D. Chiciudean, V. I. Cretu, V. Groza, "PARSECS: A Predictable Data Communication System for Smart Sensors and Hard Real-Time Applications", *IEEE Trans. Instrum. Meas.*, 14 pg., 2010
- [103] Texas Instruments, "TMP101: Digital Temperature Sensor with I2CTM Interface", Datasheet Rev. G, Texas Instruments, 2007.
- [104] Mihai V. Micea, V. Cretu, V. Groza, "Maximum Predictability in Signal Interactions with HARETICK Kernel", *IEEE Trans. Instrum. Meas.*, vol. 55, no. 4, pp. (1317-1330), Aug. 2006

## PUBLICAȚII

Publicații cotate ISI:

G. Cârstoiu, M. V. Micea, L. Ungurean, and M. Marcu, 'Novel battery wear leveling method for large-scale reconfigurable battery packs', *International Journal of Energy Research*, vol. 45, no. 2, pp. 1932–1947, Feb. 2021, doi: [10.1002/er.5879](https://doi.org/10.1002/er.5879). – nr. citări 1

L. Ungurean, M. V. Micea, and G. Cârstoiu, 'Online state of health prediction method for lithium-ion batteries, based on gated recurrent unit neural networks', *International Journal of Energy Research*, vol. 44, no. 8, pp. 6767–6777, Apr. 2020, doi: [10.1002/er.5413](https://doi.org/10.1002/er.5413). – nr. citări 16

L. Ungurean, G. Carstoiu, M. V. Micea, and V. Groza, 'Battery state of health estimation: a structured review of models, methods and commercial devices', *Int J Energ Res Int J Energ Res*, vol. 41, no. 2, pp. 151–181, Feb. 2017. – nr. citări 110

M. V. Micea, L. Ungurean, G. N. Carstoiu, and V. Groza, 'Online State-of-Health Assessment for Battery Management Systems', *Ieee T Instrum Meas Ieee T Instrum Meas*, vol. 60, no. 6, pp. 1997–2006, Jun. 2011. – nr. citări 95

M. V. Micea, G. N. Carstoiu, L. Ungurean, D. Chiciudean, V. I. Cretu, and V. Groza, 'PARSECS: A Predictable Data Communication System for Smart Sensors and Hard Real-Time Applications', *Ieee T Instrum Meas Ieee T Instrum Meas*, vol. 59, no. 11, pp. 2968–2981, Nov. 2010.

M. V. Micea, G. N. Carstoiu, L. Ungurean, D. Chiciudean, V. Cretu, and V. Groza, 'Predictable data communication interface for hard real-time systems', Oct. 2008, pp. 98–101. doi: [10.1109/ROSE.2008.4669188](https://doi.org/10.1109/ROSE.2008.4669188).

## ANEXA 1. DESCRIEREA SETULUI DE DATE NASA PROGNOSTICS CENTER OF EXCELLENCE

Cod baterie	Condiții pentru ridicarea curbelor de încărcare/descărcare
B005 B006 B007 B0018	Temperatura de operare: 25 grade C Curent încărcare: 1.5A Tensiune finală de încărcare: 4.2V Curent de descărcare: 2A constant Tensiune finală de descărcare: 2V
B0025 B0026 B0027 B0028	Temperatura de operare: 25 grade C Curent încărcare: 1.5A Tensiune finală de încărcare: 4.2V Curent de descărcare: puls de 0.05Hz cu amplitudine de 4A, factor de umplere 50% Tensiune finală de descărcare: 2V
B0029 B0028 B0030 B0031	Temperatura de operare: 25 grade C Curent încărcare: 1.5A Tensiune finală de încărcare: 4.2V Curent de descărcare: 4A Tensiune finală de descărcare: 2V
B0033 B0034 B0036	Temperatura de operare: 25 grade C Curent încărcare: 1.5A Tensiune finală de încărcare: 4.2V Curent de descărcare: 2A constant Tensiune finală de descărcare: 2V
B0038 B0039 B0040	Temperatura de operare: 25 grade C și 44 grade C Curent încărcare: 1.5A Tensiune finală de încărcare: 4.2V Curent de descărcare: 1A, 2A, 4A, constant Condiția de oprire: până când valoarea capacității a scăzut cu 20% față de valoarea capacității inițiale
B0041 B0042 B0043 B0044	Temperatura de operare: 4 grade C Curent încărcare: 1.5A Tensiune finală de încărcare: 4.2V Curent descărcare: 1A, 4A constant
B0049 B0050 B0051 B0052	Temperatura de operare: 4 grade C Curent încărcare: 1.5A Tensiune finală de încărcare: 4.2V Curent descărcare: 2A constant
B0053 B0054 B0055 B0056	Temperatura de operare: 4 grade C Curent încărcare: 1.5A Tensiune finală de încărcare: 4.2V Curent de descărcare: puls de 0.01Hz-5kHz cu amplitudine de 2A, factor de umplere

	50% Condiția de oprire: până când valoarea capacității a scăzut cu 30% față de valoarea capacității inițiale
RW1 RW2 RW7 RW8	Temperatura de operare: 25 grade C Durată încărcare: 0.5 – 3 ore Profil descărcare: curent aleator 0.5A - 4A Condiția de oprire: tensiunea la borne 3.2V
RW3 RW4 RW5 RW6	Temperatura de operare: 25 grade C Încărcare: valoarea tensiunii la borne 4.2V Profil descărcare: curent aleator 0.5A - 4A Condiția de oprire: tensiunea la borne 3.2V
RW25 RW26 RW27 RW28	Temperatura de operare: 40 grade C Încărcare: valoarea tensiunii la borne 4.2V Profil descărcare: curent aleator 0.5A - 5A Condiția de oprire: tensiunea la borne 3.2V Se folosește următoarea funcție de probabilitate pentru a selecta curentul de descărcare la fiecare minut: 0.5A 2.0% 1.0A 2.4% 1.5A 3.6% 2.0A 6.0% 2.5A 9.2% 3.0A 11.8% 3.5A 17.2% 4.0A 23.4% 4.5A 19.4% 5.0A 5.0%
RW17 RW18 RW19 RW20	Temperatura de operare: 25 grade C Încărcare: valoarea tensiunii la borne 4.2V Profil descărcare: curent aleator 0.5A - 5A Condiția de oprire: tensiunea la borne 3.2V Se folosește următoarea funcție de probabilitate pentru a selecta curentul de descărcare la fiecare minut: 0.5A 2.0% 1.0A 2.4% 1.5A 3.6% 2.0A 6.0% 2.5A 9.2% 3.0A 11.8% 3.5A 17.2% 4.0A 23.4% 4.5A 19.4% 5.0A 5.0%
RW21 RW22 RW23 RW24	Temperatura de operare: 40 grade C Încărcare: valoarea tensiunii la borne 4.2V Profil descărcare: curent aleator 0.5A - 5A Condiția de oprire: tensiunea la borne 3.2V

	<p>Se folosește următoarea funcție de probabilitate pentru a selecta curentul de descărcare la fiecare minut:</p> <p>0.5A 7.2%  1.0A 14.8%  1.5A 19.3%  2.0A 21.6%  2.5A 14.6%  3.0A 10.0%  3.5A 6.5%  4.0A 4.0%  4.5A 1.5%  5.0A 0.5%</p>
RW13 RW14 RW15 RW16	<p>Temperatura de operare: 25 grade C  Încărcare: valoarea tensiunii la borne 4.2V  Profil descărcare: curent aleator 0.5A - 5A  Condiția de oprire: tensiunea la borne 3.2V  Se folosește următoarea funcție de probabilitate pentru a selecta curentul de descărcare la fiecare minut:</p> <p>0.5A 7.2%  1.0A 14.8%  1.5A 19.3%  2.0A 21.6%  2.5A 14.6%  3.0A 10.0%  3.5A 6.5%  4.0A 4.0%  4.5A 1.5%  5.0A 0.5%</p>

## **ANEXA 2. ARTICOL 'BATTERY STATE OF HEALTH ESTIMATION: ...'**

Articol 'Battery state of health estimation: a structured review of models, methods and commercial devices' în varianta preprint apărut în , *Int J Energ Res Int J Energ Res*, vol. 41, no. 2, pp. 151–181, Feb. 2017

## Battery State of Health Estimation: a Structured Review of Models, Methods and Commercial Devices

Lucian UNGUREAN\*, Gabriel CĂRSTOIU\*, Mihai V. MICEA\*, Voicu GROZA\*\*

\* Department of Computer and Software Engineering  
Politehnica University of Timisoara, Romania  
lucian.ungurean@dsplabs.cs.upt.ro, gabriel.carstoiu@dsplabs.cs.upt.ro, mihai.micea@cs.upt.ro

\*\* School of Electrical Engineering and Computer Science  
University of Ottawa, Canada  
vgroza@uottawa.ca

*Abstract*---Estimating the dynamic status parameters of a battery, such as its state of health (SoH) and remaining useful life (RUL), is still a very difficult and complex task. In this paper we perform a structured review of the most relevant state of the art models, algorithms and commercial devices employed in the estimation of the SoH/RUL battery performance figures, in the context of embedded applications. The models and estimation techniques are thoroughly classified and, for each taxonomy class, a presentation of the working principles is made. A comprehensive set of metrics is then introduced for the evaluation of the SoH/RUL estimation techniques from the perspective of their implementation and operation efficiency in embedded systems. These algorithms are then analyzed and discussed in a comparative manner, with concrete figures and results. The capability and the performance of the different types of off-the-shelf fuel gauges to estimate the battery SoH/RUL parameters are also evaluated in this paper.

*Keywords*---Battery state estimation; State of health; Remaining useful life; Battery model; Embedded application; Battery fuel gauge.

### 1. Introduction

#### 1.1. Battery Monitoring in the Context of Embedded Devices and Applications

Efficient gathering, storing and using of electrical energy under severe constraints of volume, weight and cost, is a major topic of interest in current science and technology, with direct impact in every field of modern human activity, due to the omnipresence of embedded devices and applications.

As still the most feasible solution today, rechargeable batteries are complex electrochemical systems, highly sensitive to internal, external and operating conditions, such as temperature, charging current, discharge profiles and depth of discharge (DoD). Moreover, the battery characteristics and behavior are nonlinear and change significantly over time, due to the "aging" phenomenon. Therefore, battery management systems (BMSs) are a vital component of modern battery packs, with two major roles: (a) to ensure the safe operating area (SOA) of the battery, defined by manufacturing and architecture related specifications, such as the maximum charge and discharge currents, maximum DoD, upper and lower boundaries for the voltage of the individual cells, and (b) to continuously measure battery parameters, to determine or predict its status, health and performance figures – generally referred to as *battery monitoring*.

BMS in general, and battery monitoring in particular, require specialized hardware and software to implement precise measurements of parameters such as voltage, current and temperature, as well as complex data processing and prediction algorithms. Two main categories of status and performance figures of the battery are usually determined by such algorithms: instant status (snapshot) parameters and characteristics related to the dynamic status (history) of the battery.

State of charge (SoC) is a key indication of the instant status of the battery, which estimates the percentage amount of energy it currently stores, relative to the most recent fully charged state. The SoC can provide to the user a rough indication on the time the battery will last until full discharge and, in automotive applications for example, this parameter can be used in a similar way the fuel gauge is in conventional cars. Nevertheless, without information on the history of the battery, SoC alone cannot accurately indicate the remaining time until complete discharge. Therefore, the dynamic status figures of the battery are very important in battery monitoring and management systems.

State of health (SoH) is an indication of the dynamic status of a battery, related to the initial condition of the battery. Typically, this initial condition (at the time of manufacture) is considered SoH = 100%. With time and usage of the battery, the SoH will decrease, due to irreversible internal chemical and physical processes generally referred to as battery aging. When the capability of the battery to store and provide energy decreases under a predefined threshold, it is considered depleted (but still further usable) and needs to be replaced. At this point, the SoH is defined as 0%.

Remaining useful life (RUL) is another important parameter, belonging to the same category of dynamic battery status figures. It is directly related to the SoH, as it is generally defined as the number of charge/discharge cycles or the time left until the battery reaches to the SoH = 0%.

Compared to the instant parameter case, estimating the dynamic status parameters of a battery, such as the SoH and RUL, is still a very difficult and complex task. The challenges are further increased by the frequent cases where already used batteries are employed in various applications, such that their previous history is unknown to the battery state and health estimation algorithms. Currently, these techniques still need consistent improvements to meet the efficiency required by most real-life applications, especially in the field of sensors and embedded control.

Aware of the importance and the current problems of this topic, we are investigating and reviewing in this paper the most relevant state of the art models, algorithms and commercial devices employed in the estimation of the SoH/RUL battery parameters and performance figures.

### *1.2. Related Work and Motivation of the Survey*

As an indication of the major interest the embedded, reusable power sources and, in particular, the batteries have gained in late years, stand the impressive number and variety of researches and scientific publications in this area. There are also many surveys which discuss and compare the proposed approaches and results.

An extensive review of the state of the art methods for monitoring and estimating Lithium-Ion (Li-Ion) battery states and parameters is provided in [1]. The approaches from an impressive total of 375 scientific and technical sources are classified into 6 categories, according to the main battery parameters and performance figures: state of charge, battery capacity, impedance, available power, state of health and remaining useful life. The methods employed within each of these categories are then briefly discussed in a critical manner, with emphasis on their advantages and weaknesses. On the other hand, the techniques under review are very briefly presented or not presented at all. Only the Li-Ion batteries (LIBs) are covered regarding battery chemistry. Furthermore, this work takes into consideration only the methods which can be applied to battery electrical vehicles (BEVs) and hybrid electrical vehicles (HEVs). From the same perspective, the authors of [2] review some of the most important advances and challenges regarding BMS for Li-Ion batteries. A presentation of the SoC, SoH and state of function (SoF) battery state figures is made, along with a discussion of the various techniques proposed in the literature for estimating these parameters. Except for the SoC case, this analysis is relatively short and does not provide concrete results and figures. Similar approaches can also be found in [3] - [5].

Battery ageing process for Li-Ion chemistries in automotive application is reviewed in detail in [6] and some techniques for estimating it are then briefly analyzed, mainly from the perspective of their advantages and drawbacks. These methods, which are mainly proposed in the state of the art for the SoH prediction, are roughly classified into five categories: algorithms based on electrochemical and equivalent circuit models, performance-based, analytical methods with empirical fitting and statistical approaches. Other surveys sharing this perspective can be found in the literature, e.g. [7].

With the focus on the same electrical vehicle application area, the authors of [8] review some of the methods for estimating battery SoC and SoH performance figures, as part of the emerging prognostics and health management (PHM) scientific field. Several battery electrical and electrochemical models, such as the Thevenin-based and the runtime-based models, followed by the neural network, support vector machine, fuzzy logic and Kalman filter data-driven methods, are briefly presented in the context of the SoC and SoH estimation. The corresponding approaches in the literature are discussed in a relatively short and general manner, without performance figures and comparisons. From a similar PHM perspective, the authors of [9] review the various prognostics approaches used in the state of the art to predict the RUL of generic engineered systems, by broadly categorizing them into experience-based, data-driven, physics-based and combined (hybrid) approaches. As a case study, such a hybrid prognostics approach has been applied to estimate the RUL of Li-Ion batteries, by combining the physics-based standard particle filter state estimation process with two data-driven methods, i.e. the support vector regression and a similarity based prediction method.

A survey of the most common physics-based battery modeling techniques is given in [10], for electrical and hybrid vehicle battery chemistries such as the Nickel-Metal-Hydrate (NiMH) and Li-Ion.



The electrical, impedance and electrochemical mathematical models are presented and the related approaches in the literature are briefly discussed.

In [11], Moldovan, Weibelzahl and Hava Muntean focus on the energy-aware mobile learning systems and present, in this context, an overview of three main battery state and performance parameters, i.e. SoC, SoH and the remaining run time (RRT). Some of the techniques proposed in the field to estimate these parameters are briefly reviewed, especially for the SoC figure. A short presentation of several battery models is also included, based on a classification into five categories: empirical models, electrochemical, electrical-circuit, mathematical and hybrid models.

Motivated by the shortcomings identified in the reviews presented above, the approach proposed in this survey focuses on improving the following aspects which are incompletely treated or even missing from the related papers in the literature:

- An exhaustive classification and presentation of the working principles of the most relevant battery models and SoH/RUL estimation methods currently found in the state of the art
- Introduction of a comprehensive set of metrics, specially selected for the analysis and evaluation of the SoH/RUL estimation techniques from the perspective of their implementation and operation efficiency in embedded systems
- Detailed discussion and evaluation, in a comparative manner, of the battery state and health estimation methods, based on the proposed taxonomy and set of metrics, with concrete figures and results
- An analysis of the current off-the-shelf commercial solutions for the SoH/RUL estimation.

### 1.3. Synopsis of the Paper and Intended Audience

Using battery models to calculate various parameters and as a base for algorithms which estimate battery health and performance figures is a common approach in the state of the art. In Section 2, we present and discuss the most relevant types of battery models found in the literature. The review is based on a classification of the battery models into four main categories, as shown in Table 1.

**Table 1.** Classification of battery models

Subsection	Battery Models
2.1	Electrical (Randles) models
2.2	Electrochemical models
2.3	Mathematical models
2.4	Lifecycle models

Battery state and health estimation is a non-trivial task and a large number and variety of methods have been proposed in the literature. A survey of the SoH/RUL estimation methods is made in Section 3, based on a classification of the algorithms employed (see Table 2). For each category, we present the underlying theoretical principles, to provide an indication of both the operating foundations of the algorithms, as well as of their required computational complexity. The various methods derived from the basic approach are then discussed, in terms of implementation particularities, accuracy and performance.

**Table 2.** Classification of battery SoH/RUL estimation techniques

Subsection	SoH/RUL Estimation Methods
3.1	Coulomb counting
3.2	Open circuit voltage (OCV)
3.3	Electrochemical impedance spectroscopy (EIS)
3.4	Kalman filter based (KF): EKF, DEKF, UKF, SPKF, CDKF
3.5	Support vector machine (SVM) / Relevance vector machine (RVM)
3.6	Particle filtering: SIR, RBPF
3.7	Fuzzy logic

- 3.8 Other methods:  
 Least squares,  
 Magnetic field,  
 Sample entropy,  
 Approximation entropy,  
 Unobserved model,  
 Probability density function (PDF),  
 Gaussian process functional regression,  
 Wiener process,  
 Autoregressive model (AR) with particle  
 swarm optimization (PSO),  
 Neural network,  
 Bayesian approach

Section 4 contains a structured analysis of all the SoH/RUL estimation methods covered in this survey. We propose a system for the qualitative and quantitative evaluation of these algorithms, composed of six metrics: *battery chemistry*, *computational complexity*, *data processing mode*, *estimation result*, *processing time* and *estimation precision*. The evaluation results are synthetically presented in a head-to-head, detailed comparison table, at the end of this section.

An analysis of the state of the art commercial devices for battery monitoring and state estimation---the battery fuel gauges---is provided in Section 5. A general presentation of the battery fuel gauge is made, followed by a discussion and comparison of such off-the-shelf products, commonly found on the market. The approach is focused on the capability and the performance of the different types of fuel gauges to estimate the battery SoH/RUL figures.

In conclusion, this survey is addressed mainly to scientists and engineers in the field of embedded devices and applications, who are either studying the various existing battery state and health estimation techniques, or searching to select a particular SoH/RUL estimation method for an efficient implementation on embedded targets.

#### 1.4. List of Abbreviations

**Table 3.** List of the abbreviations and acronyms in the paper

Abbreviation	Description
AGMM	Adaptive Gaussian mixture model
AR	Autoregressive model
ARMA	Autoregressive moving-average
BEV	Battery electrical vehicle
BMC	Bayesian Monte Carlo
BMS	Battery management system
CC-CV	Constant current - Constant voltage
CDKF	Central difference Kalman filter
DEKF	Dual extended Kalman filter
DoD	Depth of discharge
DST	Dempster-Shafer theory
ECM	Equivalent circuit model
EIS	Electrochemical impedance spectroscopy
EKF	Extended Kalman filter
ESR	Equivalent series resistance
HEV	Hybrid electrical vehicle
HMPSO	Hybrid multiple particle swarm optimization
ICA	Incremental capacity analysis
KF	Kalman filter
LIB	Li-Ion battery
LiFePO4	Lithium Iron Phosphate
MPF	Mutated particle filter

MAE	Mean absolute error
MAPE	Mean absolute percentage error
MSE	Mean square error
NiMH	Nickel-Metal-Hydride
OCV	Open circuit voltage
PDF	Probability density function
PF	Particle filtering
PHM	Prognostics and health management
PSO	Particle swarm optimization
RAPF	Regularized auxiliary particle filter
RBPF	Rao-Blackwellized particle filter
RMSE	Root-mean-square error
RNN	Recursive neural network
RRT	Remaining run time
RSR	Residual systematic resampling
RUL	Remaining useful life
RVM	Relevance vector machine
SIR	Sampling importance resampling
SLA	Sealed lead acid
SPKF	Sigma point Kalman filter
SOA	Safe operating area
SoC	State of charge
SoF	State of function
SoH	State of health
SVM	Support vector machine
SVR	Support vector regression
UKF	Unscented Kalman filter
UPF	Unscented particle filtering
VRLA	Valve-regulated lead-acid

## 2. Battery Models

A battery model establishes rules and formulas which can be then used to directly calculate the desired output quantities. It can be also considered as an input on which the SoH and RUL estimation methods are based upon.

The process of modeling measured battery parameters in order to obtain estimations of its SoH and RUL has been extensively used in the literature. In this section, we will investigate some of the most relevant battery models in the state of the art.

### 2.1. Electrical Battery Models

An electrical battery model can also be referred to as an *equivalent circuit model (ECM)*, *Randles model* or *Thevenin model*. All ECMs are based on the same structure: an ideal voltage source, a series resistance, and one or several resistor-capacitor groups, the groups being connected in series with the resistance.

The ideal voltage source represents the open-circuit voltage of the battery (*OCV*, denoted as  $U_{oc}$ ) and it is an important input parameter, used in various estimation methods. The series or ohmic resistance (denoted as  $R_i$ ) is the battery internal resistance when applying a constant charge or discharge current.

The  $R_p$ - $C_p$  pairs, in which the resistance is always connected in parallel with the capacitance, are used for modeling the battery behavior in dynamic operating conditions, e.g. when the current is not constant. They are also used for expressing the battery voltage response to a step increase or decrease of the current, as current pulses are used by several parameter estimation methods described in the art.

The simple (classic) ECM, which employs a single R-C pair, is presented in [12] (see Fig. 1).

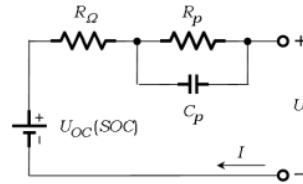


Fig. 1. Simple electrical battery model

In [13] the authors present a solution in which the model parameters are recalculated online at each charge/discharge cycle.

Some simplifying assumptions are also made in [12], to aid the calculation of the SoC:  $R_{\Omega}$ ,  $R_p$  and  $C_p$  are invariant from 3.0 V to 4.2 V, but they do change with the battery SoH. Another assumption states that the  $U_{OC}(SoC)$  is invariant with the temperature and the SoH.

The model can be expressed mathematically as:

$$\begin{cases} U = U_{OC}(SoC) + I R_{\Omega} + U_p \\ I = \frac{U_p}{R_p} + C_p \frac{dU_p}{dt} \end{cases} \quad (1)$$

In (1),  $U_p$  is the voltage drop across the resistor-capacitor group. This simple ECM is used extensively in the art ([14] – [16]).

An interesting approach to modeling the battery voltage can be found in [17]. Here, an over-simplified Randles model is used, which is comprised of only the OCV ideal source and the series ohmic resistance. The authors consider this model to be sufficient for their purposes, because it offers a balance between accuracy and simplicity, and the estimation of the SoC is carried out by using an unscented Kalman filtering approach.

A more complex ECM, which uses two R-C pairs, is described in [18]. The proposed approach is to use one of the R-C pairs for expressing the battery terminal voltage when charging the battery and the other one for modeling the battery voltage during discharge. A similar model has also been described in [19].

## 2.2. Electrochemical Battery Models

As variants of the Randles model, these models try to approximate as accurately as possible the chemical processes which take place inside a battery cell of Li-Ion class during the charge, discharge and the relaxation phases. From the chemical point of view, the Li-Ion battery is comprised of three main components: the negative electrode, the separator and the positive electrode. In this model, the Li ions flow from the negative electrode to the positive electrode during discharge, and vice versa during the charging phase ([20], [21]).

Fig. 2 shows an electrochemical impedance model based on the single particle approach employed in [20]. The impedance of the battery using this model can be expressed as:

$$Z_{batt} = j\omega L + \frac{Z_{n,SEI}}{S_n} + R_0 + \frac{Z_p}{S_p} \quad (2)$$

where  $Z_{n,SEI}$  is the impedance of the anode and the solid-electrolyte interface,  $Z_p$  is the cathode impedance, and  $S_n$  and  $S_p$  are the surface areas of the anode and the cathode, respectively.

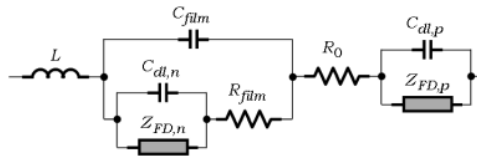


Fig. 2. Electrochemical battery model

Another variant of a second order Randles model is employed in [22] for estimating the capacity fade of Li-Ion batteries, which is essentially the SoH after accelerated aging, not after the actual usage. Although a relatively complex model is used, the authors find that only 3 out of the 8 parameters of the model can be used for the actual estimation.

The electrochemical model can also be applied to other battery chemistries, such as Lead-Acid [23]. In [24] the authors propose a simplified Randles model which they prove to be sufficient for estimating the SoH, based on only one parameter of the model.

A more complicated approach is presented in [25] - [27], where the ion concentration and chemical reactions are modeled using differential functions.

### 2.3. Mathematical Models

The mathematical models generally use one (electrical) battery parameter for the estimation of the SoH. In [28], SoH estimation is done based on CC-CV (Constant Current - Constant Voltage) charge curves and calendar aging analysis. For each charge cycle, in the CV phase, the current can be expressed using:

$$I(t) = Ae^{-Bt} + C \quad (3)$$

where  $A$ ,  $B$  and  $C$  are scalars representing the model parameters.

Taking into consideration the dependency of the model parameters on the capacity loss of the battery, (3) can be rewritten as:

$$I(t, C_{loss}) = A(C_{loss})e^{-B(C_{loss})t} + C(C_{loss}) \quad (4)$$

The authors show there is a linear relationship between the remaining capacity and the value of parameter  $B$ : a decrease in battery capacity follows a decrease in the value of parameter  $B$ . All the measurements and analysis have been done offline on a significant number of cycles.

In [29] and [30], the OCV mathematical model is employed to determine the SoH based on the incremental capacity analysis (ICA) method. The OCV can be defined as:

$$OCV(z) = K_0 - \frac{K_1}{z} - K_2z + K_3 \ln(z) + K_4 \ln(1-z) \quad (5)$$

where  $K_{0-4}$  are the model parameters, and  $z$  the normalized SoC. Better estimation results are obtained using a more complex model:

$$OCV(z) = K_0 + K_1 \frac{1}{1 + e^{\alpha_1(z-\beta_1)}} + K_2 \frac{1}{1 + e^{\alpha_2(z-\beta_2)}} + K_3 \frac{1}{1 + e^{\alpha_3(z-1)}} + K_4 \frac{1}{1 + e^{\alpha_4z}} + K_5z \quad (6)$$

The Incremental Capacity (IC) curves were derived from over 2000 charge/discharge cycles.

In [31], the SoH is predicted from the capacity curves resulted from capacity measurements in each battery cycle. Three prediction models are compared in terms of accuracy: the exponential model (7), the polynomial model (8) and the proposed ensemble model (9):

$$C_{Ak} = C_{Ak1} + C_{Ak2} = \alpha_1 e^{\alpha_2 k} + \alpha_3 e^{\alpha_4 k} \quad (7)$$

$$C_{Bk} = C_{Bk1} + C_{Bk2} = \beta_1 k^2 + \beta_2 k + \beta_3 \quad (8)$$

$$C_{Ck} = \gamma_1 e^{\gamma_2 k} + \gamma_3 k^2 + \gamma_4 \quad (9)$$

where  $C_{Ak}$ ,  $C_{Bk}$  and  $C_{Ck}$  are the values of the battery capacity,  $k$  represents the discharge cycle number, and  $\alpha_1, \alpha_2, \alpha_3, \alpha_4; \beta_1, \beta_2, \beta_3; \gamma_1, \gamma_2, \gamma_3$  are the respective model parameters. The results show that the proposed ensemble model has the best accuracy for the SoH prediction.

In [32], the authors propose an SoH evaluation method based on capacity estimation. The capacity estimation is made using a sample entropy algorithm based on the voltage responses resulting after the hybrid pulses tests, and on static capacity curves. The sample entropy values are taken as input for the capacity estimation. The sample entropy function can be defined as follows:

$$\text{SampleEn}(m, r, N) = -\ln \left[ \frac{A^m(r)}{B^m(r)} \right] \quad (10)$$

where  $N$  is the length of the sample entropy series,  $m$  is the length of a subset of the sample entropy series, for which the difference between any two consecutive elements is bounded by the tolerance  $r$ ,  $A^m(r)$  is the probability that two subsets (vectors) will match  $m + 1$  elements, and  $B^m(r)$  is the probability that two subsets will match  $m$  elements. The algorithm was tested on multicell batteries at different temperatures. The results show satisfactorily robust estimations.

The battery zero-state hysteresis model is introduced in [33], to predict the changing battery voltage profiles for each charge/discharge cycle. In this case the battery voltage is defined as follows:

$$V_k = OCV(SoC_k) - Ri_k - s_k M \quad (11)$$

where  $V_k$  is the battery voltage at battery cycle  $k$ ,  $OCV(SoC_k)$  is the battery open circuit voltage at cycle  $k$ ,  $i_k$  is the battery current at cycle  $k$ ,  $M$  and  $s_k$  are the hysteresis parameters, with  $M$  representing the magnitude and  $s_k$ , which depends on the battery current direction, is defined based on a positive value  $\varepsilon$  as follows:

$$s_k = \begin{cases} 1, & i_k > \varepsilon, \\ -1, & i_k < -\varepsilon, \\ s_{k-1}, & i_k \leq |\varepsilon| \end{cases} \quad (12)$$

#### 2.4. Lifecycle Models

The lifecycle models propose another approach to solve the problem of SoH determination of a battery cell. These battery models monitor battery parameters and their relation with the aging process over the entire lifetime of a battery. The SoH prediction is made based on the observations obtained after extensive offline tests. This approach is different from the other types of models which monitor battery parameters in real time and estimate the SoH based on the instantaneous values of these parameters.

A method for determining the battery aging model has been presented in [34]. Some parameters, such as the operating temperature, depth of discharge, discharge current rate and charge current rate, are analyzed in terms of their impact on the battery health. The proposed model was developed and validated in Matlab.

The authors of [24] propose to monitor battery parameters such as temperature, capacity and number of cycles. In order to predict the SoH, recursive neural network (RNN) and support vector machine (SVM) are used. These techniques involve extensive training and high amounts of data to be processed. To shorten the observation time, the authors used accelerated aging tests. It is questionable if these tests can accurately map the real life operating conditions of a battery.

In [35] and [36], the lifecycle models presented are based on battery reliability and failure models. These models are used in addition to data driven techniques in order to estimate the health of the battery. Matlab is used as support for evaluating the model.

Because of the extensive laboratory tests, the lifecycle models are generally used in extension with other battery models.

### 3. SoH Estimation Methods

The SoH of a battery can be estimated by applying processes and algorithms to its observable parameters, such as the voltage, current and temperature. These processes and algorithms comprise of what we denote as estimation methods, and each method uses a certain battery model as input for transforming the battery observable parameters into SoH and/or RUL. In this section we present some of the most utilized SoH estimation methods that can be found in the art.

### 3.1. Coulomb Counting

Coulomb counting is one of the most common methods used for estimating the SoH. It consists of two steps: in first step the  $Q_{discharge}$  of the battery is determined, corresponding to the value of the SoC of 0%, by integrating the value of discharge current in time; in the second step, the value of the SoH is calculated by dividing the value of  $Q_{discharge}$  with the value of the rated capacity:

$$\begin{cases} Q_{discharge} = \int_0^t I(t)dt \\ SoH [\%] = \frac{Q_{discharge}}{Q_{rated}} \times 100\% \end{cases} \quad (13)$$

The main battery parameters which are measured and monitored are: battery charge/discharge current, battery voltage, battery charge, battery temperature. This method can be easily converted into an adaptive one, by computing the two values in (13) during each discharge cycle. It was observed, e.g. in [37], that the value of  $Q_{discharge}$  is decreasing with the increase of the charge/discharge cycle number. This generates a negative trend for the values of the SoH. Commonly, for a value of the SoH of below 80%, the battery is considered unusable. The accuracy of this method relies on the measurement instruments which should be calibrated periodically.

In [37], authors claim that the accuracy of the coulomb counting method depend primarily on the measurement accuracy of the current and of the initial SoC. A new metric for characterization of the discharge is introduced, depth of discharge ( $D_{OD}$ ), which is similar to the SoC:

$$D_{OD} = \frac{Q_{released}}{Q_{rated}} \times 100\% \quad (14)$$

where  $Q_{released}$  is the discharged battery capacity. At a given instant of time,  $D_{OD}$  can be expressed as:

$$\begin{cases} D_{OD}(t) = D_{OD}(t) + \eta \Delta D_{OD}(t) \\ \Delta D_{OD}(t) = \frac{-\int_{t_0}^{t_0+\tau} I_b(t)dt}{Q_{rated}} \times 100\% \end{cases} \quad (15)$$

where  $I_b$  is the discharge/charge current,  $\eta$  is the operating efficiency.

The SoH is defined as the  $D_{OD}$  when the battery is exhausted. In each discharge and charge cycle, the  $D_{OD}$  is compensated with the charge/discharge efficiency. In addition a recalibration is made at fully charged stage and at fully discharged states.

Among the advantages of this solution can be mentioned the low cost implementation of the measurement devices, very low processing power requirements for the algorithms and easy integration [1]. The estimation accuracy depends on the battery charge/discharge cycle, reaching up to 1% for cycle numbers higher than 28.

A complete BMS is presented in [38], in which the SoC and the SoH are computed using coulomb counting. The performance of the standalone coulomb counting method is compared with a modified version which uses OCV in combination with Kalman filtering for SoC estimation. The SoC and SoH estimation errors are computed at different charge currents and the results range between +/- 3%.

### 3.2. Open Circuit Voltage (OCV)

The OCV based method for estimating the SoH of the battery is oriented on defining the SoH as a function of the open circuit voltage of the monitored battery. This method is found in the literature both in the offline and in the online form.



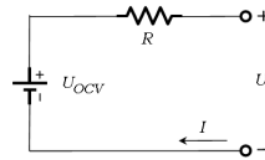


Fig. 3. Simplified electrical battery model

As presented in [17], the  $U_{OCV}$  is defined based on the simplified electrical model of the battery presented in Fig. 3:

$$U_{ocv} = U + IR \quad (16)$$

where,  $U_{ocv}$  is the battery open circuit voltage,  $I$  the battery current,  $R$  is the battery internal voltage, and  $U$  its terminal voltage.

The offline method is based on the mathematical model centered on the OCV value. As presented in the previous section and in [29], [30], extensive tests are conducted on the batteries to obtain the OCV curves and a relationship with the battery SoH.

In [17], the authors prove the influence of the temperature when building the OCV-SoC curves. This influence should not be neglected, when high accuracy of estimation is needed. The authors provide a method to produce the OCV-SoC curves, including temperature influence and provide validation tests.

Despite good accuracy of estimation, the offline approach involves extensive laboratory tests.

The online variant of the method considers OCV as a parameter of the electrical model, which is determined based on system identification. The estimation error is influenced by the initial SoC which is taken into consideration and temperature. When the difference between the real initial SoC and the one used for calculations is kept low, the error is also low. A low battery temperature introduces root-mean-square errors (RMSE) in the range of 5% - 25%.

The online OCV method is used in [14], in combination with the parameters of an electrical model, to estimate the SoC and the SoH of the battery. The algorithm presented is suitable for multicell batteries. The estimation is based on power calculations. The estimation inaccuracy depends only on the forecast period, when battery diffusion overvoltage could influence the computed SoC.

In [12], the authors present some disadvantages of using the open current voltage as a parameter for estimating SoH. They claim that this parameter is difficult to determine in practical use. They focus on using an electrical battery model and, based on the CC-CV curves, they try to eliminate the OCV from the equations. This simplifies the algorithms and, thus, reduces the processing power requirements and the hardware costs.

Other authors, [1], consider the OCV method very efficient and simple to use in combination with Coulomb counting. In [39], the authors propose a new OCV model combined with exponential parameters, (6), to estimate SoH with the help of the incremental capacity method. This combined approach gives an estimation error of about 1%.

### 3.3. Impedance Spectroscopy Method

Electrochemical impedance spectroscopy (EIS) is a method which permits obtaining highly accurate impedance measurements of a battery cell over a large spectrum of frequencies at low currents. The EIS measurements are proven in existing literature to be reliable indicators for a battery SoC and can be used to predict its SoH. Fig. 4 shows a typical real and imaginary impedance plot for a battery.



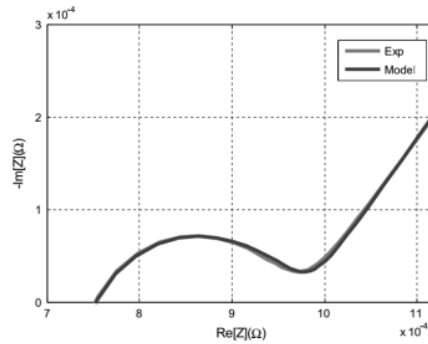


Fig. 4. Typical Nyquist plot obtained using EIS [24]

One of the most common usages of the EIS method is to identify the parameters of an electrochemical battery model. This approach is detailed in [20], as the authors develop such an electrochemical battery model based on the specific physical phenomena on Li-Ion cells. The model has 16 parameters, out of which 6 are pre-calculated and deemed fixed, while the remaining 10 are determined using the hybrid multiple particle swarm optimization (HMPSO) method, out of the measured EIS data.

The EIS measurements are carried out on Li-Ion cells charged at 100% and 50%, in the [0.025 Hz, 4 kHz] frequency range. The identified model is compared against the Randles model using EIS measurements and is demonstrated to be 8 times more accurate than the latter. The paper does not present a complete SoH estimation, but two model parameters are proposed to be used for the estimation.

In [24], the authors couple the EIS method with a RNN in order to predict the SoH and the RUL of the batteries under test. EIS is used as an offline analysis method in this paper as well, to calculate the parameters of the equivalent circuit model proposed by the authors.

The ECM contains 5 elements. Two of them are used to model resistances: the ohmic resistance  $R_1$  and the SoC-dependent resistance  $R_2$ . The model also contains two constant phase elements (CPE), namely  $CPE_1$  and  $CPE_2$ , which are used to model the diffusion, charge transfer and double layer chemical processes. It has been observed that the  $CPE_1$  and  $CPE_2$  parameters are SoC dependent. The EIS measurements have been conducted at 25 °C and the data has been collected in the frequency spectrum of [0.01 Hz, 10 kHz]. Finally, the model has been validated with a standard current load used for HEVs, thus proving that a relatively simple model is sufficiently accurate and can be used in an on-board BMS.

The actual prediction of the remaining battery capacity is performed using a RNN, for which the battery model determined using the EIS method is an input. The algorithm has been used on simulated data, to predict both the battery remaining capacity and its equivalent series resistance (ESR). The average absolute error (calculated from the mean square error that the authors provide) is 2.1% for the predicted remaining capacity, which proves the viability and accuracy of the proposed methods.

Another novel application of the EIS method can be seen in [40]. The authors analyze and propose a RUL prediction method for the calendar ageing phenomenon of Li-Ion cells used in HEVs. The experiments have been performed at 3 different ambient temperatures and at 3 SoCs, the EIS measurements proving that the battery SoH is dependent on the temperature and the SoC at which it is used. By analyzing the EIS plots, the authors have identified one parameter, namely the impedance real part at 0.1 Hz, which varies exponentially with the SoH and can thus be used to accurately predict it. Fig. 5 demonstrates that the impedance real part can be accurately predicted using the proposed exponential fitting function. The results presented in the paper show that the parameter identified using EIS measurements can be used for robust SoH prediction, and that the model and the process are simpler than those using ECM.

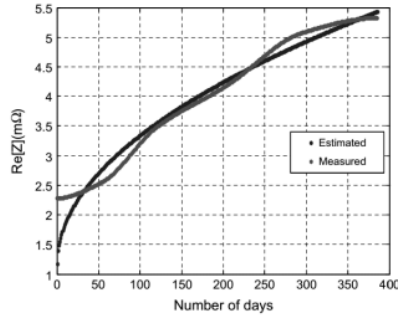


Fig. 5. Predicted and measured impedance real part [24]

### 3.4. Kalman Filtering

The Kalman filter is a powerful method for estimation of the states of a process. It consists of two parts: the prediction phase and the update phase. In the prediction phase, the process state is estimated. In the update phase, the filter obtains feedback from measurements (affected by Gaussian noise) [41]. The equations of a standard Kalman filter are:

$$\begin{cases} \hat{\mathbf{x}}_{(t|t-1)} = \mathbf{F}_t \hat{\mathbf{x}}_{(t-1|t-1)} + \mathbf{B}_t \mathbf{u}_t \\ \mathbf{P}_{(t|t-1)} = \mathbf{F}_t \mathbf{P}_{(t-1|t-1)} + \mathbf{Q}_t \\ \hat{\mathbf{x}}_{(t|t)} = \hat{\mathbf{x}}_{(t|t-1)} + \mathbf{K}_t (\mathbf{y}_t - \mathbf{H}_t \hat{\mathbf{x}}_{(t|t-1)}) \\ \mathbf{K}_t = \mathbf{P}_{(t|t-1)} \mathbf{H}_t^T (\mathbf{H}_t \mathbf{P}_{(t|t-1)} \mathbf{H}_t^T + \mathbf{R}_t)^{-1} \\ \mathbf{P}_{(t|t)} = \mathbf{P}_{(t|t-1)} - \mathbf{K}_t \mathbf{H}_t \mathbf{P}_{(t|t-1)} \end{cases} \quad (17)$$

where  $\hat{\mathbf{x}}$  is the estimated state,  $\mathbf{F}$  is the state transition matrix,  $\mathbf{B}$  is the control matrix,  $\mathbf{u}$  is the control vector,  $\mathbf{P}$  is the state variance matrix,  $\mathbf{Q}$  is the process variance matrix,  $\mathbf{y}$  is the measurement vector,  $\mathbf{H}$  is the measurement matrix,  $\mathbf{K}$  is the Kalman gain and  $\mathbf{R}$  is the measurement variance matrix.

The Kalman filter was initially intended for the use in linear systems. For non-linear systems, such as those used for the battery SoH estimation, modified versions of the Kalman filter are proposed in the literature.

The extended Kalman filter is used for predicting states of a nonlinear process, where the transition and the measurement equations are nonlinear, but need to be differentiable. For battery SoH estimation, the dual extended Kalman filter involves two Kalman filters, one filter is used to predict the SoC and the other one to predict the battery capacity. The unscented Kalman filter, also known as the sigma point Kalman filter (SPKF), eliminates some of the disadvantages of the extended Kalman filter and improves the estimation. In this case, the probability distribution functions will be estimated by a number of sigma points, which represent a subset of weighted points computed in such way they are close to the mean value. The nonlinear transformation of these points will generate the mean covariance estimate.

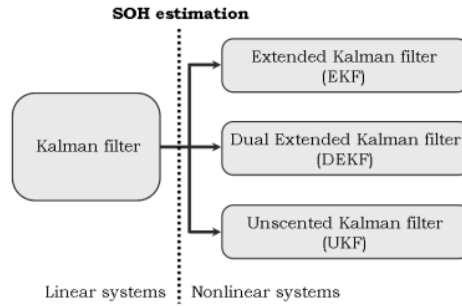


Fig. 6. Modified versions of the Kalman filter, used in nonlinear systems

In [42] and [43], the authors use a second order Randles electrical battery model to obtain the state equations of the extended Kalman filter. If in the first case, the SoH is computed based on the SoC; in the second case, the SoH is computed based on the internal battery impedance. The model equations are:

$$\begin{cases} \begin{pmatrix} SoC_k \\ V_{1,k} \\ V_{2,k} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 - \frac{\Delta t}{R_1 C_1} & 0 \\ 0 & 0 & 1 - \frac{\Delta t}{R_2 C_2} \end{pmatrix} \begin{pmatrix} SoC_{k-1} \\ V_{1,k-1} \\ V_{2,k-1} \end{pmatrix} + \begin{pmatrix} \frac{\Delta t}{C_{bat}} \\ \frac{\Delta t}{C_1} \\ \frac{\Delta t}{C_1} \end{pmatrix} \\ V_{bat,k} = V_{OC}(SoC_k) + V_{1,k} + V_{2,k} + R_0 I_{bat,k} \end{cases} \quad (18)$$

and, respectively,

$$\begin{cases} \begin{pmatrix} V_{1,k+1} \\ V_{2,k+1} \\ R_{0,k+1} \end{pmatrix} = \begin{pmatrix} e^{-\frac{\Delta t}{R_1 C_1}} & 0 & 0 \\ 0 & e^{-\frac{\Delta t}{R_2 C_2}} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} V_{1,k} \\ V_{2,k} \\ R_{0,k} \end{pmatrix} + \begin{pmatrix} R_1 \left( 1 - e^{-\frac{\Delta t}{R_1 C_1}} \right) \\ R_2 \left( 1 - e^{-\frac{\Delta t}{R_2 C_2}} \right) \\ 0 \end{pmatrix} I_{bat,k} \\ (V_{cell,k} - V_{OC,k})' = (1 \ 1 \ 0)x + R_{0,k} I_{bat,k} \end{cases} \quad (19)$$

where  $SoC_k$  is the state of charge at time instance  $k$ ,  $R_0$ ,  $R_1$ ,  $R_2$ ,  $C_1$ , and  $C_2$  are the parameters of the second order Randles which model the battery impedance,  $V_{1,k}$  and  $V_{2,k}$  are the voltage components measured for the Randles model at instance  $k$ ,  $V_{OC}$  is the battery open circuit voltage,  $I_{bat,k}$  the battery current,  $V_{bat}$  is the voltage at the battery terminals,  $V_{cell,k}$  is the battery terminal voltage at cell level, and  $x = (V_{1,k} \ V_{2,k} \ R_0 \ k)^T$  is the state space variable.

In both cases, the reported estimation error has been lower than 4%.

In [44], the extended Kalman filter is used to model Li-Ion battery parameters, such as the output voltage, open circuit voltage and the state of charge. The authors focus on car batteries used in car competitions. The experimental data shows a SoH estimation error below 1%.

The dual extended Kalman filter is presented in [45], [46]. Two Kalman filter are used: one to estimate the SoC and the other to estimate the battery parameters, such as the capacity. Fig. 7 shows the block diagram of the proposed dual extended Kalman filter. The filter is based on the first order Randles

electrical model. The authors claim that increasing the order of the electrical model makes the estimation hard to implement because of the complexity of operation in real-time applications. In the first paper, the authors use the dual filter in combination with pattern matching resulting in a estimation error of  $\pm 5\%$ . The estimation results of the algorithm proposed in the second paper are below 2%.

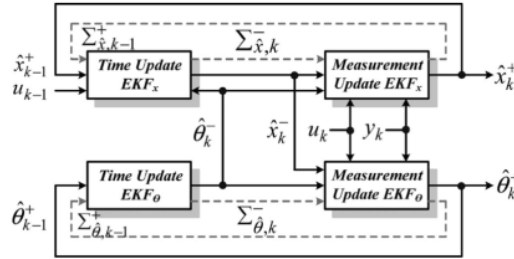


Fig. 7. Architecture of the dual extended Kalman filter [45], [46]

An unscented Kalman filter (UKF) is proposed in [47] and [48], for the estimation of the SoH. A second order Randles model is used as input for the filter equations. The unscented Kalman filter was introduced to address some of the issues of the EKF [49]. Such an issue consists in a high probability of the filter to diverge if the initial state is not specified correctly, resulting in an erroneous model of the process. Another problem is that the approximations of the posterior mean and covariance matrix can introduce large errors and lead to sub-optimal performance.

In [47], the authors compare the performance of the SoH estimation using an unscented Kalman filter with the performance of the coulomb counting method. The UKF adapts perfectly to the nonlinearities of the battery cell characteristics. It overcomes the coulomb counting accumulation errors and the estimated values of the SoC converge approximately to the real value. The estimation error is between 2.2% and 2.9%, depending on the battery cell used.

In [48], the UKF was used in addition with the support vector regression (SVR) algorithm. The purpose of SVR is to determine an initial capacity value used as input for the UKF filter. This initial value is estimated offline, based on ageing curves. The performance of the algorithms shows an estimation error below 1%.

In [50], the UKF is applied to an electrochemical battery degradation model. The authors monitor the battery voltage response to pulse load cycles and battery internal resistance. The proposed model is adaptive with a measured time of 40 minutes until the output values converge to the real parameter measurements.

The authors of [51] propose an SoH estimation technique based on a fused algorithm between EKF and quadratic discriminant analysis. The EKF state equations are developed based on the first order Randles electrical model. The estimated battery parameters obtained by applying the EKF, are further used as classification features in the quadratic analysis method. The method was applied considering three classification classes for the batteries: unused, lightly used and heavily used. The classification error is max. 8.3%.

In [52], the SoH estimation is performed via EKF and a per unit system. Similarly as in [51], the EKF equations are developed using the electrical model of Randles. Three main battery parameters are monitored:  $R_i$ ,  $R_{diff}$ ,  $C_{diff}$  – where  $R_i$  is the internal battery resistance,  $R_{diff}$  is the battery diffusion resistance and  $C_{diff}$  is the battery diffusion capacitance. All these parameters are derived from the current and voltage measurements. The proposed online method for the SoH estimation is validated using extensive offline measurements of the battery parameters and aging curves. The performance of the algorithm is  $\pm 5\%$  compared to the values obtained by ampere-hour counting.

In [53] the authors use a central difference Kalman filter (CDKF) to better estimate the nonlinearities present in the SoH estimation. This is a form of SPKF adapted to practical implementations and therefore has a much lower parameter count.

### 3.5. Support Vector Machine – Relevance Vector Machine

The support vector machine (SVM) algorithms have become a very popular estimation approach, being introduced in many fields of application and supported by a large community. For battery SoH estimation, the non-linear variant of SVM was adopted by many researchers in this field.

The basic idea of the SVM algorithm is presented in detail in [48] and [54]. Having a set of training data,  $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \subset R^n \times R$  - where  $R^n$  is the space of the input data  $x_i$  and the target data is  $y_i$ , the goal is to find a function  $f(x)$  which deviates at most with  $\epsilon$  from the targets, for all the training data and which is also as flat as possible. The function has the following form:

$$f(x) = \langle w, x \rangle + b \quad (20)$$

with  $w \in R^n$ ,  $b \in R$  and  $\langle \cdot, \cdot \rangle$  denotes the dot product in  $R^n$ . There are situations when such a function does not exist. To overcome this issue, a pair of slack variables,  $\zeta$  and  $\zeta^*$ , are introduced, thus allowing a pre-defined interval for the measurement errors. In this case, the problem reduces itself to a dual optimization problem.

The dual optimization problem is solved using quadratic programming techniques, which involve a nonlinear mapping to transform the original input ( $x$ ) to a higher dimensional input feature space. In this case the function becomes:

$$f(x) = \sum_{k=1}^K (a_k^* - a_k) K(x(k), x) + b \quad (21)$$

where  $a_k^*, a_k$  are the dual variables obtained by applying the Lagrangian with respect to the primal variables ( $w, b, \zeta, \zeta^*$ ), and  $K$  is the kernel function (mapping function) with the following form:

$$K(x(k), x(m)) = \langle \Phi(x(k)), \Phi(x(m)) \rangle \quad (22)$$

where  $\Phi$  represents the nonlinear transformation.

The relevance vector machine algorithm (RVM), [55], [56], represents the Bayesian form of the SVM and differs from it by providing a probabilistic interpretation of the outputs. The prediction of the output takes the following form:

$$t = y(x) + \varepsilon_n \quad (23)$$

where  $t$  is the predicted target,  $y(x)$  is function which translates the input data into the target data as defined in the SVM algorithm, and  $\varepsilon_n$  samples the process noise.

After expressing a maximum likelihood function and applying the Bayes rule, the following posterior distribution results:

$$p(w|t, \alpha, \sigma^2) = \frac{p(t|w, \sigma^2)p(w, \alpha)}{p(t|\alpha, \sigma^2)} \quad (24)$$

where  $w$  is a weight vector,  $\alpha$  is a hyperparameter vector used to control the deviation of the weights,  $\sigma$  is the variance, and  $p$  is the posterior distribution function.

New data predictions are obtained using the integration of the weights and computing the marginal likelihood of the hyper parameters:

$$\begin{aligned} p(t|\alpha, \sigma^2) &= \int p(t|w, \sigma^2)p(w|\alpha)dw \\ &= (2\pi)^{N/2} |\mathbf{B}^{-1} + \Phi\mathbf{A}^{-1}\Phi^T|^{-N/2} \times \exp\left(-\frac{1}{2}t^T(\mathbf{B}^{-1} + \Phi\mathbf{A}^{-1}\Phi^T)^{-1}t\right) \end{aligned} \quad (25)$$

where  $\mathbf{A} = \text{diag}(\alpha_1, \dots, \alpha_{N+1})$ , and  $\mathbf{B} = \sigma^2\mathbf{I}$ , and  $\Phi$  the design matrix of kernel functions, obtained from a set of training data.

These two estimation methods, synthesized in (21) and (25) above, can be applied on sets of training data, in which the data represent battery parameters such as the battery capacity.

The support vector machine algorithms are presented in [48], [54]–[60]. As a general characteristic, the algorithms run on high amount of training data gathered offline by cycling the batteries. The online variant of the algorithms are computationally intensive and are presented as Matlab simulations.

In [48], the data used for vector construction is taken from long term battery ageing tests. The SVM algorithm is compared with the UKF, providing almost the same prediction errors. When the initial capacity of the battery is known, the prediction error is 1%. Without having information about the initial capacity, the prediction error increases to 20%.

The authors of [57] apply the SVM technique to battery data gathered from impedance spectroscopy tests. The EIS method is based on the Randles electrical circuit. The model estimates the capacity fading and the internal resistance of the battery, which are input for the SoH and RUL calculations. The SVM has better performance than the hidden Markov model method, with an estimation error of 2%.

In [58], the SVM algorithms are applied to battery data provided by the NASA. Two variants of the algorithms are presented: online and combined (online + offline). In the case of the online SVM, the vectors are dynamically updated. The mean absolute estimation error of the SoH is 0.02 in the case of the combined algorithm and 0.03 in the case of the online variant.

SVM applied to data obtained from automated stress tests of hybrid vehicle batteries, is discussed in [59]. The support vectors are correlated with the battery load to improve the performance of the algorithms. The mean square error obtained is 0.0006, when no load collectors are used, 0.00058 for 1 load collector configuration and 0.0004, when 3 load collectors are used.

Application of the SVM for curve fitting of the battery capacity, based on incremental capacity analysis, is presented in [60]. The authors show how to apply the support vector regression on lithium polymer battery data. They propose an online solution in which the support vector machines are updated based on the current battery data and on the offline data (used in the initial stages of estimation). The prediction accuracy of the algorithm is 1% (absolute error).

In [55] and [56], the implementation of Relevance Vector Machine is presented. In the first paper, RVM is applied to sample entropy and impedance spectroscopy data provided by NASA for Li-Ion batteries during charge and discharge cycles. The authors focus on a comparison with the SVM algorithm. The estimation errors are smaller for the RVM method, i.e. below 0.6% (RMSE). The second paper discusses the capacity degradation data of lithium-ion battery used in electric vehicles. Both offline and online RVM algorithms are covered. Here, the authors emphasize on applying the RVM method for the RUL estimation. The estimation error is given in number of battery cycles. The absolute prediction error is below 9 cycles.

### 3.6. Particle Filtering Method

The method is best suited for nonlinear systems which are based on a battery model with unknown parameters, or the values of the parameters change with time. The output of the method is an estimation of the probability density function (PDF) based on a set of points (particles), which are values sampled from the state space.

One of the most frequently used algorithm for implementing particle filters is the sampling importance resampling (SIR) [61]. It effectively approximates the resulting probability distribution with a set of weighted particles, while the importance weights of these particles can be approximated as their relative probabilities. The importance weights are relative, so the sum of the weights for a probability distribution is always 1.

In [61], the authors employ a simple electrochemical model of a Li-Ion battery cell, for which they identify the parameters using an RVM regression method. The values obtained from the RVM regression are used as the initial values to the particle filtering algorithm. Afterwards, the particles are recalculated, with each measurement iteration.

Fig. 8 shows a plot of 2 typical particle filtering prediction results, the RUL probability density functions. In can be seen that one of the advantages of using the PF method is that it generates a probability distribution spread out over time, rather than just a single date or point of failure. This clearly conveys the inherent uncertainties in the model and measurements.



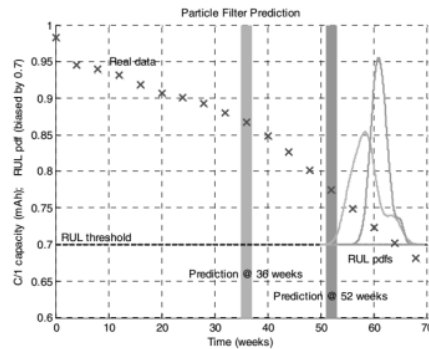


Fig. 8. Particle filtering prediction results [61]

The basic particle filter method described above is employed in [62] for estimating the SoC and SoH of Lithium Iron Phosphate (LiFePO<sub>4</sub>) batteries. The authors use a purely data driven approach, without an explicit battery model, and the only inputs taken into account are the OCV and the current. The resampling step of the PF algorithm is called "low variance resampling", which is a simplified version of the SIR method described above. In essence, when the new set of samples is generated, the old samples with high weights are multiplied, while old samples with low weights are discarded.

The authors validate their algorithm using two battery usage profiles: a photovoltaic off-grid power supply and an electric vehicle driving cycle. The results obtained are not discussed in detail, but the absolute mean error generally falls below 2%. However, the SoH estimation is not described conclusively, because only 5-6 incomplete battery cycles are analyzed.

The particle filtering method is proved to be more accurate than the statistics based regression and probabilistic regression methods in [63], by comparing the prediction accuracy and the width of the PDF. Moreover, it is shown that the PF method provides early predictions, as opposed to the other compared methods, which can be used to prevent failures.

The same authors present in [64] an improvement over the classical PF method. They observe that when the state space of the model has large vectors, the resampling process of the PF method cannot acceptably reduce the variance of particle errors. They thus propose to substitute a part of the state space by one analytically computed from another part, naming the new approach the Rao-Blackwellized particle filter (RBPF). By comparing the variances in errors of the PF and RBPF methods, it is shown that the RBPF error variance is much lower than the PF and consequently the PDF is much narrower, yielding an improvement in prediction accuracy.

In [65], the authors propose using two analytical models as inputs for the PF method. More specifically, they employ the PF for estimating the parameters of a polynomial model and an exponential model. The classical PF algorithm of SIR is used and the authors claim that it can be applied online in a BMS. An improvement of this paper is presented by the same authors in [66], although the same PF algorithm is used. A new battery capacity model is introduced, which the authors call an "ensemble model". It is a fusion of the previous two models and presents a better prediction accuracy than the individual models on which it is based.

The particle filtering method can also be used as an auxiliary mechanism in the prediction of the RUL, as demonstrated in [67]. In this paper, the authors employ a complex data-driven estimation method, based on the Verhulst model. The particle swarm optimization method is used to improve the model fitting accuracy. The PF is used only to compensate the prediction error by adjusting the parameters of the Verhulst model.

A novel approach to the SoH prediction is introduced in [68] and [69]. The authors employ an empirical mathematical battery model with 2 exponential terms, which is suitable also for on board implementations. They propose the Bayesian Monte Carlo (BMC) method which is similar to particle filtering. In this regard, its output is the probability distribution function, which is also represented by a sum of samples with associated weights:

$$P(X_k | Q_{0:k}) \approx \sum_{i=1}^N \omega_k^i \delta(X_k - X_k^i) \quad (26)$$

where  $X_k^i$  is a set of independent samples from  $P$ , which can be selected in practice by using importance sampling;  $\omega_k^i$  is the importance weight of each sample;  $\delta(\cdot)$  is the Dirac (delta) function. The resampling and the weight recalculation methods are different from the classical particle filtering technique.

The proposed SoH prediction method uses the Dempster–Shafer theory (DST), which is a data fusion method, to estimate the initial battery model parameters from the training data (the first  $n$  battery cycles). BMC is then used to update the model parameters at each cycle, thus updating the SoC and RUL estimations.

A comparison of the proposed method with the EKF method is performed in [68] for one battery type and the results show that this approach yields an estimation error of 1.1%, while the EKF method yields an error of 8.6%. In [69], the authors compare the prediction errors using the same BMC method, but with 2 different calculation methods of the model parameters. The first method is DST and the second one uses mean averaging to calculate the initial model parameters. It is shown that the DST method is better than the mean averaging method, as the prediction errors at 18 cycles are 2.1 % and 8.3% respectively.

The unscented particle filtering (UPF) method, using the same empirical mathematical model and the same battery data as [68] and [69], is proposed in [70] for estimating the RUL of Li-Ion batteries. The shortcomings of using the classical PF method (i.e. particle degeneration) are described, along with a theoretical presentation of PF, Kalman filter and UPF. The core idea of the UPF is to use the unscented Kalman filter to generate the proposal distribution of the particles, then to follow the normal PF resampling and weight calculation algorithm. The authors perform a series of tests at different SoH points and conclude that the proposed method yields an absolute error of less than 5%, while applying the standard PF method leads to an error of 7%.

In [71], the authors propose what they call a mutated particle filter (MPF) method for estimating the SoH and RUL. The new method is designed to tackle the shortcomings of the classical, Rao-Blackwellized and unscented particle filtering methods. Among the most important limitations of the afore-mentioned methods are the sample impoverishment and the failure to accurately capture the probability density function.

The proposed method transforms the weights of the outlying particles (i.e. the ones which have either very high or very low weights) so as to represent more accurately the posterior part of the probability density function. The process is presented in Fig. 9, where (a) represents the posterior distribution with the original particle weights and (b) depicts the same distribution after applying the MPF method.

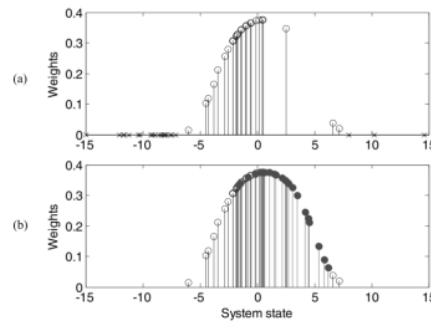


Fig. 9. MPF transformation of the system posterior PDF [71]

Using an electrochemical lumped-parameter model of a Li-Ion cell, the authors evaluate the performance of their proposed method against the classical particle filter and the regularized auxiliary particle filter (RAPF) methods. The reported results indicate that the MPF method achieves a mean



RMSE which is 20% lower and an average standard deviation which is 52% lower than that of the classical particle filter method.

In [72], the authors present an implementation of the particle filter method on an embedded platform which can be applied only by using the mean-shift and RSR (residual systematic resampling) to simplify particle filtering operations. A complete RUL estimation and failure detection framework based on particle filters is described in [74]. The framework validation is done using Matlab. In [75], the RUL estimation is determined using the battery capacity. The capacity fade is modeled using a function based on a sum of exponentials. Then, the particle filtering method is applied to this model.

The authors of [70] use particle filter to estimate both SOC and SOH. The battery is modeled using simple electrical model where open circuit voltage and internal resistance are two of the model parameters. Monitoring SOC and SOH gives a better insight of battery degradation at each cycle.

### 3.7. Fuzzy Logic Method

Fuzzy logic is a mathematical concept which practically generalizes boolean logic and sets. The core idea is that it allows the members of fuzzy sets to have degrees of membership to that set. A membership function is applied to each member  $x$  of a set  $A$  and its output is a number in the  $[0, 1]$  interval, which represents the degree to which  $x$  belongs in the fuzzy set  $A$ . Fuzzy logic uses the boolean operators AND, OR and NOT, but it extends their definitions, as presented in Table 4.

**Table 4.** Definition of fuzzy logic operators

Operation	Result
$x$ AND $y$	$\min(x, y)$
$x$ OR $y$	$\max(x, y)$
NOT $x$	$1 - x$

Fuzzy logic can easily be applied to the battery SoH estimation if we define several output membership functions for the SoH, such as *healthy*, *acceptable* and *not healthy*. However, papers describing the application of the fuzzy logic method to battery SoH estimation have not been encountered in the art until recently.

One of the first papers in which fuzzy logic is used as a viable SoH estimation method is [76]. It proposes a model based approach, utilizing an electrochemical model to extract EIS data and claims that the proposed method can be used for both primary and secondary batteries. The estimation is done for the SoC, SoH and SOL, and is performed using three methods: ARMA (autoregressive moving-average), neural network and fuzzy logic. These methods are compared using the average estimation error and they are also fed into a final decision block, which performs a fusion of the outputs of the three methods and, together with the historical knowledge of the system, outputs the estimation. No detailed description of the fuzzy logic estimator is given, except that it uses 4 of the input signals, much less than the neural network estimator. The fuzzy logic estimator showed good performance and the results obtained are comparable to the ones obtained using the neural network.

In [77], the authors perform an extended life cycle test on 95 Li-Co batteries, by charging them at a constant 0.5 C current and discharging them at different currents. They employ fuzzy identification based only on measured data to perform SoH estimation. For this, the authors have used 3 measurable quantities to form the fuzzy sets and the membership functions are of normal distribution type. The reported average error ranges from 1.4% to 9.2%, which proves that the proposed method is suitable for the SoH prediction.

SoH estimation using fuzzy identification is also employed in [78], but using the DC resistance as the measured quantity. It is measured during the charging phase and the authors show that there exists an almost linear relationship between the capacity and the DC resistance. After having calculated the average and normalized DC resistance, 4 membership functions are used to delimit the input range. The output membership functions are actually 4 thresholds of the health condition: *healthy*, *acceptable*, *weak* and *bad*. The actual calculation of the estimated capacity is done using the defuzzifying formula

$$C_A = \frac{\sum_{i=1}^4 \mu_i O_i}{\sum_{i=1}^4 \mu_i} \quad (27)$$

where  $C_A$  is the estimated actual capacity,  $\mu_i$  is the firing strength of the  $i$ -th rule and  $O_i$  is the firing strength of the output rule. The authors calculate the actual capacity of several batteries having the SoH in the range of 100% to 20% using the proposed fuzzy logic method. They show that the estimation error is less than 5% with the SoH ranging from *healthy* to *weak*, which makes the proposed method better when comparing it to the regression curve method, which is reported to yield estimation errors of around 10%.

A double exponential function, used to model the degradation curve of the battery capacity, is presented in [79]:

$$y_{fit} = a_0 + a_1 e^{-\left(\frac{x}{a_1}\right)^{\beta_1}} + a_2 e^{-\left(\frac{x}{a_2}\right)^{\beta_2}} \quad (28)$$

The fit function parameters are estimated using fuzzy logic method, the authors reporting a SoH estimation error between 5% and 10%. To decrease this error, neural network estimation is proposed, with the result of reducing the estimation error below 5%. On the other hand, the drawback is that training offline data is needed.

### 3.8. Other SoH/RUL Estimation Methods

A novel approach for estimating SOH and RUL, proven suitable for online implementation on resourced constrained embedded platforms is described [13]. The SOH is estimated based on SOC calculation and a simple least squares regression algorithm is used for prognostics.

In [80] the authors use a novel approach to measure the SoH of SLA (sealed lead acid) batteries. They use **magnetic field** to determine if a battery is degraded or healthy. Two coils are used, one to generate the magnetic field and the second coil to measure the magnetic field changes due to the proton concentration in the battery. Instead of giving some SoH measurement error, the article is focused on the determination of the voltage on the second coil and its relation to the SoH when the battery is healthy and to the SoH when the battery is damaged. The authors do not provide analysis on how this method can be influenced by external magnetic fields, nor do they provide a shielding solution.

A least square method fused with **sample entropy** is presented in [81]. The authors use sample entropy to model the Li-Ion battery capacity, based on intensive offline tests and a third degree polynomial function to link the capacity with the corresponding sample entropy value at each temperature.

The **approximation entropy** method is described in [82]. This method is an extension of the sample entropy technique used to estimate the SoH for lead-acid batteries and has a better estimation performance. The method consist of computing an  $ApEn(m, r, N)$  function for a data sequence  $x(i) = x(1), x(2), x(3), \dots, x(N)$ , with  $N$  representing the total number of data points,  $m$  the length of the subset of data points for which the algorithm runs,  $r$  represents the tolerance window, which usually is considered  $0.2SD$ , and  $SD$  represents the standard deviation of the data sequence, defined as

$$SD = \sqrt{\frac{1}{N-1} \sum_{i=1}^N \left[ x(i) - \frac{1}{N} \sum_{i=1}^N x(i) \right]^2} \quad (29)$$

The  $ApEn$  function is computed as follows:

$$\begin{aligned} ApEn(m, r, N) &= \phi^m(r) - \phi^{m+1}(r) \\ \phi^m(r) &= \frac{1}{N-m+1} \sum_{i=1}^N \ln \left( \frac{V^m(i)}{N-m+1} \right) \\ V^m(i) &= No.of \bar{d}[X(i), X(j)] \leq r \end{aligned}$$

and  $d$  is the distance between the following vectors (the maximum absolute difference between the scalar components of the vectors):

$$\begin{aligned} X(i) &= [x(i), x(i+1), \dots, x(i+m-1)], i = 1, N-m+1 \\ X(j) &= [x(j), x(j+1), \dots, x(j+m-1)], j = 1, N-m+1 \end{aligned}$$

The author in [83] uses an **unobserved model** to estimate both the SoC and SoH of a Li-Ion battery. The resulting set of equations is obtained from a complex electrical battery model which uses two RC groups in series. Additionally, a separate electrical model is employed for the battery self-discharge. The SoH estimator has the following form:

$$\dot{\xi} = A\xi + BI_B + D\phi(V_{SOC}, I_B) \quad (30)$$

where  $A$ ,  $B$  and  $D$  are state and output matrixes,  $\xi$  is the state vector, and  $\phi$  denotes the nonlinearities of the estimation. Based on this state equation the authors compute the internal resistance  $R_0$  of the battery and use the following equation to compute the SoH:

$$SOH = \frac{R_{0,EOF} - R_0}{R_{0,EOF} - R_{0,NEW}} \times 100 \quad (31)$$

In which the  $R_0$  is the current battery internal resistance,  $R_{0,EOF}$  is the internal battery resistance when end of life is reached, and  $R_{0,NEW}$  is the battery internal resistance when the battery is new. The end of life internal resistance value is considered 160% of the resistance value when the battery is new.

In [84], the problem of SoH estimation is addressed based on **probability density function** in conjunction with the ICA (incremental capacity analysis). The authors focus on different Li-Ion batteries which have different CV (cyclic voltammogram) curves and where the curve fitting is not accurate because of the measurement noise of voltage/current. Using probability density function plots, the SoH evolution can be monitored with high accuracy. To compute the PDF, the authors use the following series:

$$\begin{cases} V_{d,k} = V_0 + k \cdot \delta V, & k = 0, 1, 2, \dots \\ Q_{d,n} = Q_0 + n \cdot dQ = n \cdot I \cdot t, & n = 0, 1, 2, \dots \end{cases} \quad (32)$$

where  $V_{d,k}$  is the sampled digital output voltage of the battery,  $d$  is the naming convention for digital,  $k$  represents the sample index, and  $Q_{d,k}$  is the integrated discrete capacity of the battery which corresponds to  $V_{d,k}$ . These series of values are computed offline by performing multiple charge/discharge cycles.

An adaptation with two PDF functions is described in [85], in the context of adaptive Gaussian mixture model (AGMM). One PDF function is used for historical dataset characterization and the other one is used for current dataset characterization. The algorithm is applied to the capacity parameter of the battery.

The authors in [86] propose the **Gaussian process functional regression** for SoH prognostics. They monitor the battery capacity parameter in order to determine the SoH evolution. The Gauss process is defined as a finite set of random variables  $\{f(x_i) | x_i \in X\}$ , where  $x$  is the input space (the charge/discharge cycles), and a mean and covariance function:

$$\begin{aligned} m(x) &= E(f(x)) \\ k(x_i, x_j) &= E[(f(x_i) - m(x_i)) \cdot (f(x_j) - m(x_j))] \end{aligned}$$

The covariance function is composed of a functional part and a model noise part, which is considered Gaussian white noise. The flexibility can be increased by introducing additional parameters, when using linear mean function and quadratic polynomial functions as the mean functions. In this case, the estimation errors are reduced significantly. The algorithm needs a high number of training data sets for increased performance; the authors use a minimum of 100 cycles.

A **Wiener Process with Measurement Error** for RUL estimation is proposed in [87]. The RUL degradation model is represented by the following equation:

$$Y(t) = X(t) + \varepsilon = \lambda t + \sigma_B B(t) + \varepsilon \quad (33)$$

where  $Y(t)$  is the degradation process, with the measurement error being taken into consideration,  $X(t)$  represents the degradation model without measurement error,  $\varepsilon$  is the measurement error,  $\lambda$  is the drift parameter,  $\sigma_B$  is the diffusion parameter,  $B(t)$  is the standard Brownian motion. The algorithm uses offline parameter estimation combined with an online parameter update process (see Fig. 10). The online parameter update is improved by using a heuristic framework for updating the parameters. The authors apply this degradation model to the battery capacity parameter and consider that a battery has reached the end of life when the capacity has faded beyond 20%-30% of the nominal capacity.

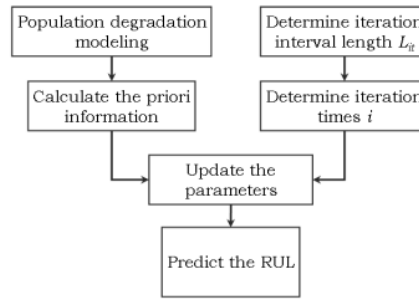


Fig. 10. RUL estimation algorithm proposed in [87]

In [88], the authors propose a data driven method based on **autoregressive model (AR)** fused with **particle swarm optimization (PSO)**. The AR model is optimized by using the PSO algorithm, defined by the following equations:

$$\begin{cases} v_i^{k+1} = w \cdot v_i^k + c_1 (pbest_i - x_i^k) + c_2 (gbest^k - x_i^k) \\ x_i^{k+1} = x_i^k + d \cdot v_i^{k+1} \end{cases} \quad (34)$$

where the first equation represents the velocity update,  $v_i^{k+1}$  is the new velocity of particle  $i$ ,  $w$  the inertia weight coefficient,  $v_i^k$  is the old velocity of particle  $i$ ,  $c_1$  and  $c_2$  represent the acceleration constants,  $d$  is the tuning parameter (having the value of 1 in almost all cases),  $x_i^k$  is the place of particle  $i$  at iteration  $k$ ,  $x_i^{k+1}$  is the place of particle  $i$  at iteration  $k+1$ ,  $pbest_i$  is the best place particle  $i$  can have as individual, and  $gbest$  is the best place of the whole group in the solution space. The PSO is used to determine the AR model order at each step  $k$ . To achieve accurate results a limited number of training data is needed.

In [89] a **neural network** is used to model the battery electrochemical properties, having the following activation functions for the neurons in a Gaussian form:

$$\varphi_i(r_k) = G(\|r_k - t_i\|) = \exp\left(-\frac{\|r_k - t_i\|^2}{\sigma_i^2}\right), \quad i = 1..M \quad (35)$$

where  $r_k = [V_k I_k SoC_k]^T$  represents the input vector of the neural network at sampling time  $k$ ,  $t_i$  and  $\sigma_i$  are the center and, respectively, the standard deviation for the Gaussian function,  $M$  is the number of the neurons in the hidden layer. The neural network outputs the terminal voltage of the time instance  $(k+1)$ . The SoH is determined based on the voltage - capacity slopes, using an additional fuzzy system. Offline battery cycles are required to train the neural network in order to obtain good estimations.

A structured neural network [43] is used to model nonlinearities of SoH estimation using the battery impedance parameter. The neural network is structured in three main subnetworks: one which uses the SoC, current and temperature measurements to estimate battery voltage and two subnetworks to estimate the R-C parameters of the second order Randles electrical battery model. The probabilistic neural network is used in [90]. This type of neural network has the advantage of increased learning speed and the training data can be supplied directly without iterations.

In [91] and [92], other **Bayesian approaches** to RUL estimation are presented. The degradation models use capacity curve predictions in a probabilistic way. The estimation is done online, based on offline data, which increases the estimation performance.

#### 4. Evaluation and Discussion of the SoH/RUL Estimation Methods

##### 4.1. Evaluation Criteria

In the previous sections we have seen that there are many SoH/RUL estimation methods in the art, each of them targeting specific applications and providing more or less information about their performance. In this section we will perform a structured analysis of all the estimation methods that have been presented up to this point.

We thus propose the following six criteria to present a qualitative and quantitative evaluation of the above-mentioned SoH/RUL estimation methods:

- *Battery chemistry*: the chemical structure of the battery cell anode/cathode/electrolyte for which the method is applicable. Although the majority of the presented papers target Li-Ion cells with many different anode and cathode chemistries, some of them can also be applied to NiMH or NiCd, while others can be applied to all secondary battery chemistries.
- *Computational complexity*: how suitable is the proposed algorithm to be implemented in a BMS running on an embedded system. The suitability of implementation is inversely proportional to the complexity of the mathematical transformations of the method.
- *Data processing mode*: denotes when the acquired battery parameters are being processed. Offline processing means that the method is applied retroactively, on data which has been previously acquired, while online processing means that the method is applied in real time, as soon as the data is being acquired. There exists also a combination of the two techniques, whereby some battery parameters are processed offline and the others online.
- *Estimation result*: whether the proposed method can be used for estimating only SoH, only RUL or both quantities.
- *Processing time* (if available): The time required for the method to produce the first estimation, from the moment it has sufficient data. Can also represent the average time needed per estimation result, if the estimation is periodically calculated. It is directly dependent on the hardware on which the method is implemented and executed.
- *Estimation precision*: The degree to which the result obtained by applying the method is closer to the actual result. Different metrics are used in the art for expressing the estimation error or precision, such as the absolute error, absolute mean error or RMSE.

##### 4.2. Evaluation of Estimation Methods

The **Coulomb counting method** is one of the most common methods for the SoH estimation.

It can be applied to all battery chemistries, although the works [37] and [38], referenced in Section 3.1, are focused on Li-Ion and Li-Polymer. The integration of the battery current over time has a low computational complexity, common operations being the addition and the multiplication.

As a result, this method is suited for online implementation, even on low power or resource-constrained microcontrollers. The processing times depend on the operation frequency of the microcontroller as well as on the available computation units (arithmetic/logic units).

Being a lightweight method, the complexity is moved towards the hardware platform, which is focused on the current measurement. This hardware complexity introduces the need for calibration after several battery charge/discharge cycles.

From the estimation point of view, the method shows a strong dependency between the estimation error and the battery cycle used for this estimation. The estimation performance is relatively low, i.e. below 10%. To increase this estimation performance, the coulomb counting method is used in conjunction with other methods such as the Kalman filtering. In this case the estimation error can be improved to 3%, at the expense of a longer processing time (e.g. 2000 seconds for the first estimation, as reported in [38]).

The **OCV based methods** use the relationship between the OCV and the SoH to predict the actual SoH of the battery.

These methods can be applied on a wide range of battery chemistries, but for each chemistry, there is a different relation between the OCV and the SoH of the battery. This relationship varies also with the



battery cycle. This problem is reduced to finding the relationship between the OCV and the SoH, which can be applied in real-life conditions.

The computational complexity is given by the complexity of the function chosen to express this relationship. A higher complexity gives a better estimation with the disadvantage of being hard to implement on an embedded target with low processing power. For polynomial functions, the estimation error varies between 5% and 12%. Using exponential terms for polynomial functions the estimation error can be decreased to less than 1%.

Finding the relationship between the OCV and the SoH can be an extensive process which involves performing offline battery charge/discharge cycles. Due to their nature, these techniques are more suited for offline estimation, the online estimation being based on offline data (table lookups).

The **Electrochemical Impedance Spectroscopy** method is targeted specifically at battery cells having Li-Ion based chemistries. The method is rarely used individually for determining the SoH; it is rather used in combination with other methods.

The computational complexity is low when using the standalone method; however it increases when the method is used in combination with a RNN [24]. Data processing is done exclusively offline, because the hardware needed for impedance measurements is too complex to be found in an embedded BMS implementation. Also, performing the measurements requires applying specific current patterns on the battery, which might not be possible during normal online operation.

By applying this method, both SoH and RUL can be obtained. No processing time information is provided in the presented papers.

The reported precision is relatively low, with absolute estimation errors ranging from 2% to 10%.

The **Particle Filtering** method is a data driven technique, and as such it is suitable for all battery chemistries. However, in this survey we analyze the method based on its applicability to various Li-Ion based cell chemistries. The mathematical apparatus required for implementing this method uses exponential functions, multiplication and vector/matrix transpositions, thus having a high computational complexity.

Data processing is mostly performed online. In other papers the results are obtained by using a combination of offline measurements and parameter identification, as well as online PF processing.

The method is usually employed for determining the RUL, but some papers report estimating the SoH as well. Processing time information is scarce, with only one paper reporting it. The estimation precision varies in the art. Some papers have achieved a very good precision, with results in the sub-1% range. Others have presented 3% to 12% error ranges.

**Fuzzy logic** is a relatively new method to be utilized for SoH/RUL estimation. The method is very versatile, as it has been reported to be used on Li-Ion, NiMH and NiCd cell chemistries. It presents a relatively high computational complexity, because its implementation relies on multiplication, division and exponential function terms.

Online data processing is an advantage of this method, although it can benefit from offline parameter calculations as well.

All the papers which detail the application of this method focus on the SoH estimation. In general, no processing time information is provided, with the exception of [77], which does present some limited performance data. The estimation precision is average, with reported estimation errors in the range of 1.4% to 10%.

**Kalman filtering** can be used on multiple cell battery for estimating the SoH. The nature of the method implies a medium to high computational complexity because of the matrix operations combined with exponential terms.

This is the reason why very limited implementations on embedded targets are present in the literature; most of the implementations are related to Matlab simulations. The estimation error is lower than the estimation error of the Coulomb counting and the OCV based techniques, being in most cases below 3%.

In terms of data processing mode, there are both offline and online variants of the Kalman filtering method.

**Machine Learning methods (SVM-RVM)** have a very low estimation error, with values lower than or equal to 1%, in most cases. This precision is achieved due to high computational complexity:

exponential functions, vector dot products, etc. This method is suited for the Li-Ion battery chemistry, but can also be adapted to other chemistries, depending on the battery model used.

This method can be used both online and offline. Due to its complexity, a very low number of implementations are currently available.

The data driven approach based on **least squares** is suited for embedded platforms with scarce resources. It can be used for the SOH estimation of all battery types. The algorithm complexity is low because it involves only basic multiplications and additions. Another advantage is the low execution time, of the order of hundreds of microseconds per battery cycle on a minimum 10 MHz core. Among the drawbacks of the method is the high estimation error in the case of nonlinear slope of the SOH degradation curve.

**Magnetic field** based method is a new approach to determine the SoH in a noninvasive way. It was intended to be used for lead acid batteries. The main advantage of this kind of method is the computational complexity which is limited to simple linear equations in which the magnetic field is computed.

**Sample entropy** and **approximate entropy** methods are suited for all chemistries. They have a good estimation precision, with an absolute error of about 2% and can be modified to run online, offline or in a combination of the two. The main disadvantage is the high computational complexity needed to run these methods; vector/matrix operations and exponential functions are needed for this type of estimator.

The **unobserved model** is used to estimate the SoH of Li-Ion batteries. This online method uses offline data training to estimate system parameters. The absolute error of the estimation converges to ~0% after 4000 seconds of running the algorithm; in the first 681 seconds it is higher than 21.2% and decreases gradually to 6.7% after 1000 sec, 3% after 2000 sec, and 0.6% after 3000 sec. The estimation error converges to 0 with an acceptable rate.

From the computational complexity point of view, this method uses vector and matrix operations, having parameters in exponential form. This gives a high computational complexity.

**Probability density function (PDF)** can be used to monitor the SoH on Li-Ion batteries. The relation between the SoH and PDF curves are computed offline; in this case an online solution has to be modeled in such way it will use the offline data. The advantage of using PDF to estimate the SoH is given by the simpler way of using the PDF curves which are more accurate than using curve fitting algorithms which need high accuracy measurements of current and voltage yielding to higher costs. The estimation error of this method is below 2% with medium to high computational complexity: function derivatives can be expressed by using subtraction of the function members. The offline characterization of the batteries, based on extensive battery tests, can be mentioned as an important disadvantage of this technique.

**Gaussian process functional regression** for SoH prognostics is one of the accurate estimation methods for Li-Ion batteries. The main advantage is the reduction of cost and complexity of the Particle Filter method, by using a prediction approach. Estimation errors of less than 0.5% are obtained by using medium to high computational complexity given by logarithmic, exponential functions and matrix operations. The method can be fitted to an online and offline approach, extensive training data has to be performed in order to obtain high precision estimations. The flexibility of the method permits estimation of SoH in the presence of the regeneration phenomenon which is usually found in the case of Li-Ion batteries, while other estimation methods can be negatively affected by its presence.

**Wiener process** based method can be applied to predict the RUL of Li-Ion Batteries. The degradation model focuses on battery capacity parameter. A practical implementation was conducted on NASA battery data and the mean square error (MSE) was used to determine the algorithm accuracy. The MSE is high on first cycles when the initial parameter estimation has relative high errors, but it decreases below 1 after 65th – 67th cycle, when the heuristic parameter update has high accuracy.

Despite a very good estimation accuracy, this method has a high computational complexity given by the use of matrix and vector operations combined with logarithmic and exponential functions. Another disadvantage represents the offline parameter computation which requires some training data.

The **AR with PSO** method can achieve very good results for RUL estimation of Li-Ion batteries. It is applied to the capacity parameter of the battery. The estimation errors are less than 10 cycles when the input dataset contains no more than 120 points and can reach 1-2 cycles for larger datasets, e.g. 190 data points. The advantage of using this method is its reduced computational complexity due to the linear equations. It is known that using an autoregressive method implies heavy training data but the combination with the PSO algorithm reduces the quantity of training data at minimum.

The algorithm has also a reduced processing time, e.g. ~2 seconds are needed for the estimation of the RUL on an Intel processor with 3GHz.

SoH estimation using **neural networks** can be used for all types of battery chemistries. The method is very accurate having a maximum estimation of 2% (using 100 training samples). The algorithm is computationally intensive, using exponential functions, vector and matrix operations. The method supports the online mode only, based on offline data.

The **Bayesian approach** used for estimating the RUL of Li-Ion battery has very good estimation results but the implementation is difficult due to its high computational complexity. Very good results can be obtained using this method: the error can be reduced to 0.2% - 0.3% when offline data is used in conjunction with online samples.

A detailed synopsis of the evaluation results for the state of the art battery SoH/RUL estimation methods, according to the previously presented criteria, is provided in Table 5.

**Table 5.** Comparative evaluation of the SoH/RUL estimation methods

Method	Refs	Battery chemistry	Computational complexity	Data processing mode	SoH/RUL estim.	Processing time for first estimation	Estimation precision
Coulomb Counting	[37]	Li-Ion	Linear functions and operations	Online	SoH	N/A	~ 9% in 21th cycle; ~ 1% after 8th cycle
	[38]	LiFePO4	Linear functions and operations (standalone method) ; matrix operation (multiplication, transpose), exponential functions (modified version of method)	Online	SoH	In extension with Kalman filtering process for experimental data: 2000 sec	Between +/-3%
Open circuit voltage (OCV)	[17]	Li-Ion	Linear operations: Matrix operation (multiplication, transpose)	Offline and online	SoH	N/A	< 5% in most of the cases when temperature is taken into account; > 5% when the initial SoC is between 45% and 65%
	[14]	Li-Ion	Linear operations	Online	SoH	Implemented on a low cost 16 bit microcontroller; processor load: 60%	RMSE: 4% ÷ 20% for the constant current case; for constant power: 5% ÷ 12%
	[12]	Li-Ion	Linear operations, exponential functions	Online with parameters obtained offline	SoH	N/A	Estimation error: < 1%
	[39]	Li-Ion	Polynomial functions with exponential terms	Parameter identification offline	SoH	N/A	< 1%
Kalman filtering	[51]	Apply to all; demonstrated for NiMH	Matrix operation (multiplication, transpose), log function	Offline	SoH	N/A	Max. 8.3%
	[52]	Li-Ion	Matrix operation	Offline	SoH	N/A	Estimation error in



		(multiplication, transpose), square root function				range of +/-5%
[47]	Li-Ion	Matrix operation (multiplication, transpose), polynomial functions (3rd order)	Online based on offline data	SoH	N/A	RMS ranges between 2.2% and 2.9%
[45]	Li-Ion	Matrix operation (multiplication, transpose)	Online	SoH	N/A	Estimation error in range of +/-5%
[42]	Li-Ion	Matrix operation (multiplication, transpose), exponential function	Online	SoH	Estimation based on data acquired over 30 minutes interval	Parameter model estimation error 1%; SoH estimation error: < 4%
[48]	Li-Ion	Matrix operation (multiplication, transpose), exponential function	Online	SoH	N/A	Estimation error: < 1%
[43]	Li-Ion	Matrix operation (multiplication, transpose), exponential function	Offline	SoH	N/A	Estimation error: < 4%
[44]	Li-Ion	Matrix operation (multiplication, transpose), exponential function	Online with parameters obtained offline	SoH	N/A	Estimation error: < 1%
[46]	Lead acid	Matrix operation (multiplication, transpose), exponential function	Online	SoH	N/A	< 2%
[50]	Li-Ion	Matrix operation (multiplication, transpose), Linear operations,	Online	SoH	Update of the model parameters in order to have an acceptable estimation error range, requires 40 minutes	2%
[53]	Li-Ion	Matrix operation (multiplication, transpose), square root function	Online	SoH	N/A	Mean error ranges between -0.6 and 0.6
	Machine learning (SVM-RVM)					
[48]	Apply to all; demonstrated for NMC (Nickel Manganese Cobalt)	Exponential functions, vector dot product (multiply and accumulate)	Offline	SoH	N/A	Estimation error: < 1%
[57]	Li-Ion	Exponential functions, vector dot product (multiply and accumulate)	Offline (can be adapted online)	SoH and RUL	Specified that the algorithm is CPU intensive	~ 2%
[55]	Li	Exponential functions, vector dot product (multiply and accumulate), matrix operations (transpose, multiplication)	Offline	SoH	N/A	Missing data for first battery cycle; SVM RMSE between 0.47 and 1.43; RVM RMSE between $5.96 \cdot 10^{-5}$ and 0.5
[58]	Li	Exponential functions, vector dot product (multiply and accumulate)	Online and combined (online + offline)	RUL	N/A	Combined approach: MAE 0.02; online approach: MAE 0.03
[59]	Li-Ion	Exponential	Online based	SoH and	N/A	Conventional use:

			functions, vector dot product (multiply and accumulate)	on offline data	RUL		MSE $8 \cdot 10^{-4}$ ; 1LC: MSE $0.1 \cdot 10^{-4}$ ; 3LC: MSE $0.02 \cdot 10^{-4}$
	[56]	Li-Ion	Exponential functions, vector dot product (multiply and accumulate), matrix operations (transpose, multiplication)	Online based on offline data	RUL	N/A	Absolute prediction error after 100 cycles: 4 cycles; absolute prediction error after 200 cycles: 2 cycles
	[60]	Li-Polymer	Exponential functions, vector dot product (multiply and accumulate) Authors state as "moderate"	Online based on offline data	SoH	Specified moderate computational load	Absolute prediction error: 1%
Particle filtering	[63]	Graphite anode Lithium Nickel Cobalt oxide cathode	Exponential functions	Offline (RVM regression) and online (Particle filtering)	RUL	N/A	9.1% at week 32 of 64; 4.0% at week 48 of 64
	[64]	Graphite anode Lithium Nickel Cobalt oxide cathode	Exponential functions, floating point multiplication	Offline (RVM regression) and online (Particle filtering)	RUL	N/A	3.1% at testing week 48 of 64
	[65]	Li-Ion	Exponential functions (model A), multiplications (model B)	Online	SoH and RUL	N/A	Model A: 0.4% at cycle 550 of 780; model B: 11.5% at cycle 550 of 780
	[66]	Li-Ion	Exponential functions, multiplication	Online	RUL	N/A	For Cell #01: 12.1% at cycle 566 of 849; for Cell #02: 1.5% at cycle 428 of 643; for Cell #V4: 0.1% at cycle 422 of 633
	[67]	Li-Ion	Exponential functions, division	Online	RUL	N/A	For cell A4: 4.2% at cycle 27 of 48; for cell A2: 1.6% at cycle 120 of 189
	[68]	Graphite anode Lithium Cobalt oxide cathode	Exponential functions, floating point multiplication, vector transposition	Can be processed online	SoH and RUL	10 ms on an Intel Core i7 M60 2.67 GB processor and 4 GB RAM (Matlab)	2.1% at cycle 18 of 48; 1.1% at cycle 250 of 610
	[69]	Li-Ion	Exponential functions, multiplication	Offline and online	RUL	N/A	2.0% at cycle 18 of 48.
	[62]	LiFePO4	Trigonometric functions, floating point multiplication	Offline and online	SoH	N/A	< 2.0%, but few cycles are analyzed
	[73]	Graphite anode Lithium Cobalt oxide cathode	Exponential functions, floating point multiplication, vector transposition	Offline and online	RUL	N/A	< 5%
Fuzzy logic	[76]	Primary Li Secondary NiCd and NiMH	N/A	Offline and online	SoH	N/A	Between 2.0% and 7.9% for SoC; N/A for SoH
	[77]	Graphite anode	Exponential functions,	Online	SoH	15.6 ms on an unspecified	Between 1.4% and 9.2%

		Lithium Cobalt oxide cathode	multiplication, division			computer	
	[78]	Li-Ion	Multiplication, division	Online	SoH	N/A	Max. 5%
Impedance spectroscopy	[20]	Li-Ion	*No algorithm for directly determining the SoH is presented	Offline	SoH and RUL	N/A	N/A
	[24]	Graphite anode Lithium Cobalt Manganese Nickel oxide cathode	Exponential function, RNN with 13 neurons	Offline	SoH and RUL	N/A	Average 2.1%
	[40]	Graphite anode Lithium Cobalt Manganese Nickel oxide cathode	*No algorithm for directly determining the SoH is presented	Offline	SoH and RUL	N/A	Max 10% (for the predicted impedance real part)
Data driven least squares regression	[13]	NiMH	Simple linear equations	Online	SoH and RUL	< 550 $\mu$ s at 14MHz on ARM 7 embedded platform	Estimation error: +/- 5 cycles
Magnetic field	[80]	SLA	Simple linear equations	Online	SoH	N/A	N/A (gives only correlation between SoH and magnetic field)
Least squares fused with Sample Entropy	[81]	Li-Ion	Vector operations, logarithmic function	Online, based on offline data	SoH	N/A	Average relative error 2%
Approx. entropy	[82]	Lead Acid	Vector operation, matrix operation, logarithmic functions	Online, based on offline data	SoH	N/A	N/A
Unobserved model	[83]	Li-Ion	Vector and matrix operations, exponential function	Online, based on offline data	SoH	N/A	Absolute error: > 21.2% before 681 sec, 6.7% after 1000 sec, 3% after 2000 sec, 0.6 % after 3000 sec, ~ 0% after 4000 sec, but only a few cycles are considered
PDF	[84]	LiMn <sub>2</sub> O <sub>4</sub> LiFePO <sub>4</sub>	Linear functions, Derivative functions, probabilities	Online based on offline data	SoH	N/A	Estimation error under 2%
GPR/GPFR	[86]	Li-Ion	Matrix operation, function derivatives, Logarithmic and exponential expressions	Offline, can be modified to online, based on offline data	SoH	N/A	MAPE: < 0.5% RMSE: 1.5 $\div$ 6
Wiener Process	[87]	Li-Ion	Exponential and logarithmic expressions, vector and matrix operations	Online, based on offline data	RUL	N/A	MSE: < 1 after cycle 67
AR with PSO	[88]	Li-Ion	Linear expressions	Online, with minimum training data	RUL	< 2 seconds for first prediction, ~ 2 seconds for prediction with 190 samples as	Estimation error of remaining cycles: < 30 cycles in first 120 cycles, < 2 cycles when

						input, on an Intel Core E5800, 3Ghz, 2GB RAM	estimation is done between 120 – 200 cycles
Neural network	[89]	VRLA	Exponential functions, matrix and vector operations	Online	SoH	N/A	Estimation error of max. 2%
	[43]	Li-Ion	Exponential functions, matrix and vector operations	Online, based on offline data	SoH	N/A	Estimation error: < 0.5%
	[90]	Li-Ion	Exponential functions, matrix and vector operations	Online, based on offline data	SoH	N/A	Average error of 0.28% for a number of 2000 training samples
Bayesian approach	[91]	Li-Ion	Vector operations, exponential functions	Online	RUL	N/A	Estimation error below 2%
	[92]	Li-Ion	Vector operations, Exponential functions	Online	RUL	N/A	Estimation error between 0.3% and 10%

### 5. Commercial Solutions for the SoH Estimation

Battery powered devices experienced a tremendous growth in the past years which has determined the boom of a new semiconductor market niche: the battery fuel gauges. Some of the most prominent semiconductor companies, which provide such integrated circuits/solutions, are Texas Instruments, Maxim Integrated, Linear Technology, ST Microelectronics, HDM Systems.

A battery fuel gauge is an integrated circuit which has several features: measurement of battery system parameters such as current, voltage and temperature, SoC determination, and SoH determination. In most of the cases the battery fuel gauges need external circuitry in order to operate. A block schematic of this circuitry is presented in Fig. 11.

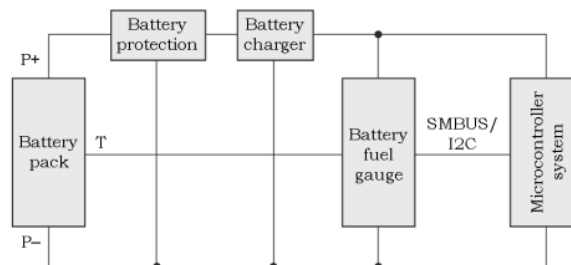


Fig. 11. Battery fuel gauge circuit schematics

Besides the positive and negative terminals (P+ and P-), the battery can have a third terminal (T) if temperature sensor resides inside the pack. An external thermistor can be externally applied, if no internal sensor is presented.

A battery protection circuit is needed to safeguard the battery from overvoltage, overcurrent, or high temperature condition. The protection circuit must decouple the battery pack from the rest of the system in case of an unwanted event is present.

The battery charger is used to provide the charge voltage when the battery pack is in charge mode and system voltage if the battery pack is in discharge mode. It has to be chosen based on the electrical characteristics of the battery pack and the system components.

The battery fuel gauge is the most important component. Based on the battery state, it controls the battery charger and informs periodically the microcontroller system about the battery state via specific interface (usually SMBUS or I2C).

The microcontroller system stores all the information from the battery fuel gauge and controls its mode of operation. It can control all the fuel gauge configuration registers available.

Products like the ST Microelectronics STC3105 [93], or Linear Technology LTC2941 [94] do not expose information related to SoH/RUL of the battery. They only provide information about battery SoC and in this case the user can build up its own SoH estimation algorithm using all the battery parameters which are provided by the circuit. The user can configure the fuel gauge parameters such as charge/discharge thresholds, battery charge termination, battery alerts.

Texas Instruments bq27531 [95] is a fuel gauge designed for Li-Ion batteries which uses Texas Instruments Impedance Track technology for battery management [96]. It implements an electrical battery model which monitors the OCV of the battery. The user is able to configure the gauge parameters based on the technical specification from the battery manufacturer: input voltage, input current, designed capacity, battery limits, charge termination timer intervals, thermal compensation, charging voltage, which can be set as a function of temperature (5 points can be stored in the non-volatile memory), and charging profile for different charging rates (5 steps).

This fuel gauge provides information about the SoH of the battery. The "StateOfHealth" register is available and the data is expressed in percentage (0 – 100%). The manufacturer gives little information about how the value in the register is computed. The value is based on the ratio between the "Full Charge" capacity and the "Designed" capacity. The "Full Charge" capacity used in the calculation is an estimation of the full charge capacity at 25° Celsius and the full charge capacity corresponding to the current SoH at the "LoadI" register. The manufacturer claims an estimation error of 1% in most of the operating conditions. To achieve this, the user should know how to configure the chip. The user should also know the battery parameters from the battery manufacturer. Older versions of Texas Instruments bq product use coulomb counting for the SoC determination, thus supporting a wide range of battery chemistries: NiMH, Lead-Acid, Li-Ion.

Similarly to the previous product, the Maxim Integrated MAX17047 [97] computes the SoH based on SoC. The SoC determination algorithm is a mix between coulomb counting and OCV. The OCV is used to reduce the errors introduced by the need of coulomb counting calibration. The SoH value is given by the "Age" register which is the ratio between the "FullCAP" register and "DesignCap" register. Here, the "FullCAP" register stores the actual value of the charging capacity sampled at the moment of charge termination. The estimation error is under 2%.

The HDM Systems BFG-24-S fuel gauge [98] is designed for Lead-Acid battery chemistry. The SoH is computed based on the ratio between the total Ahr for a fully charged battery and the designed capacity (given by the battery manufacturer). The full charge capacity is computed based on coulomb counting method. The estimation error is 5%.

**Table 6.** Comparison of commercial solutions for battery fuel gauge and SoH estimation

Product	Refs	Battery chemistry	Computational complexity	Data processing mode	SoH/ RUL estimation	Processing time for first estimation	Estimation precision
TI Bq27531	[95] [96]	Li-Ion (older versions support a wide range)	Hardware	Online	SoH	1 sec	1%
MAX17047	[97]	Li-Ion	Hardware	Online	SoH	N/A	< 2%
HDM BFG-24-S	[98]	Lead-Acid	Hardware	Online	SoH	N/A	5%

## 6. Conclusions

There is a multitude of battery models and methods in the field of battery SOH estimation, ranging from the simplest to implement, to very complex ones, which require high computational resources. As a consequence, a large number of papers discussing this topic have been published in the art and some of the most relevant are analyzed in this review.

The presented models and methods are very heterogeneous, making it hard to objectively analyze and compare them. A comprehensive set of clear and measurable criteria has been defined in order to sort out this problem.

Considering the precision of the estimation, we have found that, although the more complex methods generally present a lower estimation error, there are also several relatively simple methods that provide good precision. However, the main drawback of using simple methods is that these lower ranges of errors are achieved only in case of stable operating conditions and low variance of battery parameters. On the other hand, the more complex methods cover almost all possible battery life degradation scenarios. For instance, most of the simple solutions do not take into account the variation of battery temperature or are defined only for a fixed value, e.g. 25 degrees Celsius.

Our evaluation revealed that only a small number of papers which present complex methods have a practical implementation on embedded targets, most of them being only validated in simulation environments, such as Matlab.

The most complex techniques, such as the Particle filtering, Kalman filtering and SVR, have an estimation error in the range of 1% to 2%, while the other present large and very large estimation errors in comparable scenarios.

The existing commercial solutions usually implement the simpler methods and offer a wide range of configurability, making them relatively easy to integrate into a BMS, compared to more theoretical approaches.

### References

- [1] W. Waag, C. Fleischer, D.U. Sauer, "Critical Review of the Methods for Monitoring of Lithium-Ion Batteries in Electric and Hybrid Vehicles", *J. Power Sources*, 258, 2014, (321–339).
- [2] L. Lu, X. Han, J. Li, J. Hua, M. Ouyang, "A Review on the Key Issues for Lithium-Ion Battery Management in Electric Vehicles", *J. Power Sources*, 226, 2013, (272–288).
- [3] B. Price, J. Richardson, E. Dietz, "State-of-Charge and State-of-Health Monitoring: Implications for Industry, Academia, and the Consumer", in *Proc. IEEE EIT*, 2012, (1–6).
- [4] J. Zhang, J. Lee, "A Review on Prognostics and Health Monitoring of Li-Ion Battery", *J. Power Sources*, 196, 2011, (6007–6014).
- [5] X.S. Si, W. Wang, C.H. Hu, D.H. Zhou, "Remaining Useful Life Estimation: A Review on the Statistical Data Driven Approaches", *J. Power Sources*, 213, 2011, (1–14).
- [6] A. Barre, B. Deguilhem, et al., "A Review on Lithium-Ion Battery Ageing Mechanisms and Estimations for Automotive Applications", *J. Power Sources*, 241, 2013, (680–689).
- [7] K.B. Hatzell, A. Sharma, H.K. Fathy, "A Survey of Long-Term Health Modeling, Estimation, and Control of Lithium-Ion Batteries: Challenges and Opportunities", in *Proc. AACC*, 2012, (584–591).
- [8] S.M. Rezvanzaniani, et al., "Review and Recent Advances in Battery Health Monitoring and Prognostics Technologies for Electric Vehicle (EV) Safety and Mobility", *J. Power Sources*, 256, 2014, (110–124).
- [9] L. Liao, F. Kottig, "Review of Hybrid Prognostics Approaches for Remaining Useful Life Prediction of Engineered Systems, and an Application to Battery Life Prediction", *IEEE Trans. Reliab.*, 63 (1), 2014.
- [10] A. Seaman, et al., "A Survey of Mathematics-Based Equivalent-Circuit and Electrochemical Battery Models for Hybrid and Electric Vehicle Simulation", *J. Power Sources*, 256, 2014, (410–423).
- [11] A.N. Moldovan, S. Weibelzahl, C.H. Muntean, "Energy-Aware Mobile Learning: Opportunities and Challenges", *IEEE Commun. Surv. & Tutorials*, 16 (1), 2014, (234–265).
- [12] Z. Guo, X. Qiu, G. Hou, B.Y. Liaw, C. Zhang, "State of Health Estimation for Lithium Ion Batteries Based on Charging Curves", *J. Power Sources*, 249, 2014, (457–462).
- [13] M.V. Micea, L. Ungurean, G.N. Carstoiu, V. Groza, "Online State-of-Health Assessment for Battery Management Systems", *IEEE Trans. Instrum. Meas.*, 60 (6), 2011, (1997–2006).
- [14] W. Waag, C. Fleischer, D.U. Sauer, "Adaptive On-Line Prediction of the Available Power of Lithium-Ion Batteries", *J. Power Sources*, 242, 2013, (548–559).
- [15] A.H. Ranjbar, A. Banaei, A. Khoobroo, B. Fahimi, "Online Estimation of State of Charge in Li-Ion Batteries Using Impulse Response Concept", *IEEE Trans Smart Grid*, 3 (1), 2012, (360–367).
- [16] K.M. Tsang, L. Sun, W.L. Chan, "Identification and Modelling of Lithium Ion Battery", *Energ. Convers. Managem.*, 51, 2010, (2857–2862).
- [17] Y. Xing, W. He, M. Pecht, K.L. Tsui, "State of Charge Estimation of Lithium-Ion Batteries Using the Open-Circuit Voltage at Various Ambient Temperatures", *Applied Energ.*, 113, 2014, (106–115).
- [18] J. Renmlinger, M. Buchholz, et al., "State-of-Health Monitoring of Lithium-Ion Batteries in Electric Vehicles by On-Board Internal Resistance Estimation", *J. Power Sources*, 196, 2011, (5357–5363).
- [19] Z. Chen, C.C. Mi, Y. Fu, J. Xu, X. Gong, "Online Battery State of Health Estimation Based on Genetic Algorithm for Electric and Hybrid Vehicle Applications", *J. Power Sources*, 240, 2013, (184–192).



- [20] S.E. Li, B. Wang, H. Peng, X. Hu, "An Electrochemistry-Based Impedance Model for Lithium-Ion Batteries", *J. Power Sources*, 258, 2014, (9–18).
- [21] G.K. Prasad, C.D. Rahn, "Model Based Identification of Aging Parameters in Lithium Ion Batteries", *J. Power Sources*, 232, 2013, (79–85).
- [22] M. Ecker, J.B. Gerschler, J. Vogel, S. Kabitz, F. Hust, P. Dechent, D.U. Sauer, "Development of a Lifetime Prediction Model for Lithium-Ion Batteries Based on Extended Accelerated Aging Test Data", *J. Power Sources*, 215, 2012, (248–257).
- [23] L.C. Stevanatto, V.J. Brusamarello, S. Tairov, "Parameter Identification and Analysis of Uncertainties in Measurements of Lead–Acid Batteries", *IEEE Trans. Instrum. Meas.*, 63 (4), 2014, (761–768).
- [24] A. Eddahech, O. Briat, N. Bertrand, J.Y. Deletage, J.M. Vinassa, "Behavior and State-of-Health Monitoring of Li-Ion Batteries Using Impedance Spectroscopy and Recurrent Neural Networks", *Electr. Power Energ. Syst.*, 42, 2012, (487–494).
- [25] S.J. Moura, N.A. Chaturvedi, M. Krstic, "PDE Estimation Techniques for Advanced Battery Management Systems - Part II: SOH Identification", in *Proc. IEEE ACC*, 2012, (566–571).
- [26] S.J. Moura, N.A. Chaturvedi, M. Krstic, "PDE Estimation Techniques for Advanced Battery Management Systems - I: SOC Estimation", in *Proc. IEEE ACC*, 2012, (559–565).
- [27] S. Santhanagopalan, R.E. White, "State of Charge Estimation Using an Unscented Filter for High Power Lithium Ion Cells", *Int. J. Energ. Research*, 34, 2010, (152–163).
- [28] A. Eddahech, O. Briat, J.M. Vinassa, "Determination of Lithium-Ion Battery State-of-Health Based on Constant-Voltage Charge Phase", *J. Power Sources*, 258, 2014, (218–227).
- [29] C. Weng, Y. Cui, J. Sun, H. Peng, "On-Board State of Health Monitoring of Lithium-Ion Batteries Using Incremental Capacity Analysis With Support Vector Regression", *J. Power Sources*, 235, 2013, (36–44).
- [30] C. Weng, J. Sun, H. Peng, "A Unified Open-Circuit-Voltage Model of Lithium-Ion Batteries for State-of-Charge Estimation and State-of-Health Monitoring", *J. Power Sources*, 258, 2014, (228–237).
- [31] Y. Xing, E.W.M. Ma, K.L. Tsui, M. Pecht, "An Ensemble Model for Predicting the Remaining Useful Performance of Lithium-Ion Batteries", *Microelectron. Reliab.*, 53 (6), 2013, (811–820).
- [32] X. Hu, S.E. Li, Z. Jia, B. Egardt, "Enhanced Sample Entropy-Based Health Management of Li-Ion Battery for Electrified Vehicles", *Energy*, 64, 2014, (953–960).
- [33] M. Mastali, J. Vazquez-Arenas, R. Fraser, M. Fowler, S. Afshar, M. Stevens, "Battery State of the Charge Estimation Using Kalman Filtering", *J. Power Sources*, 239, 2013, (294–307).
- [34] N. Omar, M.A. Monem, et al., "Lithium Iron Phosphate Based Battery - Assessment of the Aging Parameters and Development of Cycle Life Model", *Applied Energ.*, 113, 2014, (1575–1585).
- [35] Y. Xing, K.L. Tsui, Q. Miao, M. Pecht, "Prognostics and Health Monitoring for Lithium-Ion Battery", in *Proc. IEEE ISI*, 2011, (242–247).
- [36] J.A.M. Penna, C.L. Nascimento Jr., L.R. Rodrigues, "Health Monitoring and Remaining Useful Life Estimation of Lithium-Ion Aeronautical Batteries", in *Proc. IEEE AERO*, 2012, (1–12).
- [37] K.S. Ng, C.S. Moo, Y.P. Chen, Y.C. Hsieh, "Enhanced Coulomb Counting Method for Estimating State-of-Charge and State-of-Health of Lithium-Ion Batteries", *Applied Energ.*, 86, 2009, (1506–1511).
- [38] B.P. Divakar, K.W.E. Cheng, et al., "Battery Management System and Control Strategy for Hybrid and Electric Vehicle", in *Proc. PESA*, 2009, (1–6).
- [39] C. Weng, J. Sun, H. Peng, "A Unified Open-Circuit-Voltage Model of Lithium-Ion Batteries for State-of-Charge Estimation and State-of-Health Monitoring", *J. Power Sources*, 258, 2014, (228–237).
- [40] A. Eddahech, O. Briat, E. Woigard, J.M. Vinassa, "Remaining Useful Life Prediction of Lithium Batteries in Calendar Ageing for Automotive Applications", *Microelectr. Reliab.*, 52, 2012, (2438–2442).
- [41] A. Saxena, "Subject MI63: Kalman Filter Tank Filling", *Cornell Univ.*, 2008, Online: <http://www.cs.cornell.edu/courses/cs4758/2012sp/materials/mi63slides.pdf>.
- [42] B. Rosca, J.T.B.A. Kessels, H.J. Bergveld, P.P.J. van den Bosch, "On-line Parameter, State-of-Charge and Aging Estimation of Li-ion Batteries", in *Proc. IEEE VPPC*, 2012, (1122–1127).
- [43] D. Andre, A. Nuhic, T. Soczka-Guth, D.U. Sauer, "Comparative Study of a Structured Neural Network and an Extended Kalman Filter for State of Health Determination of Lithium-Ion Batteries in Hybrid Electric Vehicles", *Eng. Applic. Artif. Intellig.*, 26, 2013, (951–961).
- [44] M. Daboussy, D. Chrenko, et al., "Characterisation of a Commercial Automotive Lithium Ion Battery Using Extended Kalman Filter", in *Proc. IEEE ITEC*, 2013, (1–6).
- [45] J. Kim, S. Lee, B.H. Cho, "Complementary Cooperation Algorithm Based on DEKF Combined With Pattern Recognition for SOC/Capacity Estimation and SOH Prediction", *IEEE Trans. Power Electron.*, 27 (1), 2012, (436–451).
- [46] T. Dragicevic, S. Sucic, J.M. Guerrero, "Battery State-of-Charge and Parameter Estimation Algorithm Based on Kalman Filter", in *Proc. IEEE EuroCon*, 2013, (1519–1524).
- [47] B. Pivlelait, C. Rentel, et al., "An Advanced Battery Management System for Lithium Ion Batteries", in *Proc. GVSETS*, 2011, (1–7).
- [48] D. Andre, C. Appel, T. Soczka-Guth, D.U. Sauer, "Advanced Mathematical Methods of SOC and SOH Estimation for Lithium-Ion Batteries", *J. Power Sources*, 224, 2013, (20–27).

- [49] E.A. Wan, R. Van Der Merwe, "The Unscented Kalman Filter for Nonlinear Estimation", in *Proc. IEEE AS-SPCC*, 2000, (153–158).
- [50] B. Bole, C.S. Kulkarni, M. Daigle, "Adaptation of an Electrochemistry-Based Li-Ion Battery Model to Account for Deterioration Observed Under Randomized Use", in *Proc. PHM14*, 2014, (1–9).
- [51] C. Barlak, Y. Ozkazanc, "A Classification Based Methodology for Estimation of State-of-Health of Rechargeable Batteries", in *Proc. IEEE ELECO*, 2009, (II.101–II.105).
- [52] J. Kim, B.H. Cho, "State-of-Charge Estimation and State-of-Health Prediction of a Li-Ion Degraded Battery Based on an EKF Combined With a Per-Unit System", *IEEE Trans. Vehicular Tech.*, 60 (9), 2011.
- [53] J. Remmlinger, M. Buchholz, T. Soczka-Guth, K. Dietmayer, "On-Board State-of-Health Monitoring of Lithium-Ion Batteries Using Linear Parameter-Varying Models", *J. Power Sources*, 239, 2013, (689–695).
- [54] A.J. Smola, B. Scholkopf, "A Tutorial on Support Vector Regression", *Statistics Comput.*, 14 (3), 2004, (199–222).
- [55] A. Widodo, M.C. Shim, W. Caesarendra, B.S. Yang, "Intelligent Prognostics for Battery Health Monitoring Based on Sample Entropy", *Expert Syst. Applic.*, 38, 2011, (11763–11769).
- [56] D. Wang, Q. Miao, M. Pecht, "Prognostics of Lithium-Ion Batteries Based on Relevance Vectors and a Conditional Three-Parameter Capacity Degradation Model", *J. Power Sources*, 239, 2013, (253–264).
- [57] B. Pattipati, C. Sankavaram, K.R. Pattipati, "System Identification and Estimation Framework for Pivotal Automotive Battery Management System Characteristics", *IEEE Trans. Syst. Man Cybernetics C*, 41 (6), 2011, (869–884).
- [58] J. Zhou, D. Liu, Y. Peng, X. Peng, "Dynamic Battery Remaining Useful Life Estimation: An On-Line Data-Driven Approach", in *Proc. IEEE I2MTC*, 2012, (2196–2199).
- [59] A. Nuhic, T. Terzimehic, et al., "Health Diagnosis and Remaining Useful Life Prognostics of Lithium-Ion Batteries Using Data-Driven Methods", *J. Power Sources*, 239, 2013, (680–688).
- [60] C. Weng, Y. Cui, J. Sun, H. Peng, "On-Board State of Health Monitoring of Lithium-Ion Batteries Using Incremental Capacity Analysis With Support Vector Regression", *J. Power Sources*, 235, 2013, (36–44).
- [61] B. Saha, K. Goebel, S. Poll, J. Christophersen, "An Integrated Approach to Battery Health Monitoring Using Bayesian Regression and State Estimation", in *Proc. IEEE Autotestcon*, 2007, (646–653).
- [62] S. Schwunk, N. Armbruster, S. Straub, J. Kehl, M. Vetter, "Particle Filter for State of Charge and State of Health Estimation for Lithium-Iron Phosphate Batteries", *J. Power Sources*, 239, 2013, (705–710).
- [63] K. Goebel, B. Saha, A. Saxena, J.R. Celaya, J.P. Christophersen, "Prognostics in Battery Health Management", *IEEE Instrum. Meas. Mag.*, 11 (4), 2008, (33–40).
- [64] B. Saha, K. Goebel, S. Poll, J. Christophersen, "Prognostics Methods for Battery Health Monitoring Using a Bayesian Framework", *IEEE Trans. Instrum. Meas.*, 58 (2), 2009, (291–296).
- [65] Y. Xing, E.W.M. Ma, K.L. Tsui, M. Pecht, "A Case Study on Battery Life Prediction Using Particle Filtering", in *Proc. IEEE PHM*, 2012, (1–6).
- [66] Y. Xing, E.W.M. Ma, K.L. Tsui, M. Pecht, "An Ensemble Model for Predicting the Remaining Useful Performance of Lithium-Ion Batteries", *Microelectron. Reliab.*, 53, 2013, (811–820).
- [67] W. Xian, B. Long, M. Li, H. Wang, "Prognostics of Lithium-Ion Batteries Based on the Verhulst Model, Particle Swarm Optimization and Particle Filter", *IEEE Trans. Instrum. Meas.*, 63 (1), 2014, (2–17).
- [68] W. He, N. Williard, M. Osterman, M. Pecht, "Prognostics of Lithium-Ion Batteries Based on Dempster-Shafer Theory and the Bayesian Monte Carlo Method", *J. Power Sources*, 196, 2011, (10314–10321).
- [69] W. He, N. Williard, M. Osterman, M. Pecht, "Remaining Useful Performance Analysis of Batteries", in *Proc. IEEE PHM*, 2011, (1–6).
- [70] S. Schwunk, N. Armbruster, S. Straub, J. Kehl, M. Vetter, "Particle filter for state of charge and state of health estimation for lithium-iron phosphate batteries", *J. Power Sources*, 239, 2013, (705–710).
- [71] D.Z. Li, W. Wang, F. Ismail, "A Mutated Particle Filter Technique for System State Estimation and Battery Life Prediction", *IEEE Trans. Instrum. Meas.*, 63 (8), 2014, (2034–2043).
- [72] G. Li, B. Li, Z. Liu, X. Chen, "Implementation and Optimization of Particle Filter Tracking Algorithm on Multi-DSPs System", in *Proc. IEEE CIS*, 2008, (152–157).
- [73] Q. Miao, L. Xie, H. Cui, W. Liang, M. Pecht, "Remaining Useful Life Prediction of Lithium-Ion Battery With Unscented Particle Filter Technique", *Microelectron. Reliab.*, 53, 2013, (805–810).
- [74] M.E. Orchard, L. Tang, G. Vachtsevanos, "A Combined Anomaly Detection and Failure Prognosis Approach for Estimation of Remaining Useful Life in Energy Storage Devices", in *Proc. PHM11*, 2011, (1–7).
- [75] C. Chen, M. Pecht, "Prognostics of Lithium-Ion Batteries Using Model-Based and Data-Driven Methods", in *Proc. IEEE PHM*, 2012, (1–6).
- [76] J.D. Kozlowski, "Electrochemical Cell Prognostics Using Online Impedance Measurements and Model-Based Data Fusion Techniques", in *Proc. IEEE AERO*, 2003, (3257–3270).
- [77] H.T. Lin, T.J. Liang, S.M. Chen, "The State-of-Health Diagnosis of Li-Co Batteries with Fuzzy Identification", in *Proc. IEEE PEMCC*, 2012, (2678–2682).
- [78] K.M. Tsang, W.L. Chan, "State of Health Detection for Lithium Ion Batteries in Photovoltaic System", *Energ. Convers. Manag.*, 65, 2013, (7–12).



- [79] M. Landi, G. Gross, "Measurement Techniques for Online Battery State of Health Estimation in Vehicle-to-Grid Applications", *IEEE Trans. Instrum. Meas.*, 63 (5), 2014, (1224–1234).
- [80] N. Khare, P. Singh, J.K. Vassiliou, "A Novel Magnetic Field Probing Technique for Determining State of Health of Sealed Lead-Acid Batteries", *J. Power Sources*, 218, 2012, (462–473).
- [81] X. Hu, S.E. Li, Z. Jia, B. Egardt, "Enhanced Sample Entropy-Based Health Management of Li-Ion Battery for Electrified Vehicles", *Energy*, 64, 2014, (953–960).
- [82] Y.H. Sun, H.L. Jou, J.C. Wu, K.D. Wu, "Auxiliary Health Diagnosis Method for Lead-Acid Battery", *Applied Energy*, 87, 2010, (3691–3698).
- [83] M. Gholizadeh, F.R. Salmasi, "Estimation of State of Charge, Unknown Nonlinearities, and State of Health of a Lithium-Ion Battery Based on a Comprehensive Unobservable Model", *IEEE Trans. Ind. Electron.*, 61 (3), 2014, (1335–1344).
- [84] X. Feng, J. Li, M. Ouyang, et al., "Using Probability Density Function to Evaluate the State of Health of Lithium-Ion Batteries", *J. Power Sources*, 232, 2013, (209–218).
- [85] J. Yu, "Health Degradation Detection and Monitoring of Lithium-Ion Battery Based on Adaptive Learning Method", *IEEE Trans. Instrum. Meas.*, 63 (7), 2014, (1709–1721).
- [86] D. Liu, J. Pang, J. Zhou, Y. Peng, M. Pecht, "Prognostics for State of Health Estimation of Lithium-Ion Batteries Based on Combination Gaussian Process Functional Regression", *Microelectron. Reliab.*, 53, 2013, (832–839).
- [87] S. Tang, C. Yu, X. Wang, X. Guo, X. Si, "Remaining Useful Life Prediction of Lithium-Ion Batteries Based on the Wiener Process with Measurement Error", *Energies*, 7, 2014, (520–547).
- [88] B. Long, W. Xian, L. Jiang, Z. Liu, "An Improved Autoregressive Model by Particle Swarm Optimization for Prognostics of Lithium-Ion Batteries", *Microelectron. Reliab.*, 53, 2013, (821–831).
- [89] M. Shahriari, M. Farrokhi, "Online State-of-Health Estimation of VRLA Batteries Using State of Charge", *IEEE Trans. Ind. Electron.*, 60 (1), 2013, (191–202).
- [90] H.T. Lin, T.J. Liang, S.M. Chen, "Estimation of Battery State of Health Using Probabilistic Neural Network", *IEEE Trans. Ind. Informatics*, 9 (2), 2013, (679–685).
- [91] G. Jin, D.E. Matthews, Z. Zhou, "A Bayesian Framework for On-Line Degradation Assessment and Residual Life Prediction of Secondary Batteries in Spacecraft", *Reliab. Eng. Syst. Safety*, 113, 2013, (7–20).
- [92] S.S.Y. Ng, Y. Xing, K.L. Tsui, "A Naive Bayes Model for Robust Remaining Useful Life Prediction of Lithium-Ion Battery", *Applied Energy*, 118, 2014, (114–123).
- [93] STMicroelectronics, "STC3105: Battery Monitor Ic with Alarm Output for Gas Gauge Applications", *Datasheet*, 2011, Online: <http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/DM00037883.pdf>.
- [94] Linear Technology, "LTC2941 Battery Gas Gauge with I2C Interface", *Datasheet*, 2010, Online: <http://cds.linear.com/docs/en/datasheet/2941fa.pdf>.
- [95] Texas Instruments, "bq27531-G1 Battery Management Unit Impedance Track Fuel Gauge with MaxLife Technology for Use With the bq2419x Charger Controller", *Datasheet*, 2015, Online: <http://www.ti.com/lit/ds/symlink/bq27531-g1.pdf>.
- [96] M. Yu, Y. Barsukov, M. Vega, "Theory and Implementation of Impedance Track Battery Fuel-Gauging Algorithm in bq2750x Family", *Texas Instruments Appl. Report*, SLUA450, 2008, Online: <http://www.ti.com/lit/an/slua450/slua450.pdf>.
- [97] Maxim Integrated, "MAX17047/MAX17050: ModelGauge m3 Fuel Gauge", *Datasheet*, 2015, Online: <https://www.maximintegrated.com/en/products/power/battery-management/MAX17050.html#popppdf>.
- [98] HDM Systems, "BFG-24-S Battery Fuel Gauge Product Specification", *Datasheet*, 2009, Online: [http://www.hdm-sys.com/pdf/hdm\\_manual\\_bfg.pdf](http://www.hdm-sys.com/pdf/hdm_manual_bfg.pdf).

### **ANEXA 3. ARTICOL , 'ONLINE STATE OF HEALTH PREDICTION METHOD FOR LITHIUM-ION BATTERIES... '**

Articol , 'Online state of health prediction method for lithium-ion batteries, based on gated recurrent unit neural networks' în variantă pre-print, apărut în, *International Journal of Energy Research*, vol. 44, no. 8, pp. 6767–6777, Apr. 2020

## Online State of Health Prediction Method for Lithium-Ion Batteries, Based on Gradient Recurrent Unit Neural Networks

Lucian UNGUREAN, Mihai V. MICEA\*, Gabriel CĂRSTOIU

Department of Computer and Information Technology  
Politehnica University of Timisoara, Romania

lucian.ungurean@dsplabs.cs.upt.ro, mihai.micea@cs.upt.ro, gabriel.carstoiu@dsplabs.cs.upt.ro

*Abstract*—Online State of Health (SOH) prediction of Lithium-ion batteries remains a very important problem in assessing the safety and reliability of battery powered systems. Deep learning techniques based on recurrent neural networks with memory, such as the Long-Short Term Memory (LSTM) and Gradient Recurrent Unit (GRU), have very promising advantages, when compared to other SOH estimation algorithms. This work addresses the battery SOH prediction based on GRU. A complete BMS is presented along with the internal structure and configuration parameters. The neural network was highly optimized by adaptive moment estimation (Adam) algorithm. Experimental data show very good estimation results for different temperature values, not only at room value. Comparisons performed against other relevant estimation methods highlight the performance of the recursive neural network algorithms such as GRU and LSTM, with the exception of the battery regeneration points. Compared to LSTM, the GRU algorithm gives slightly higher estimation errors, but within similar prediction error range, while needing significantly fewer parameters (about 25% fewer), thus making it a very suitable candidate for embedded implementations.

*Keywords*—Lithium-ion battery; On-line state prediction; State of health; Recurrent neural networks; Gradient Recurrent Unit; Long-Short Term Memory; Battery management system.

### 1. Introduction

Advances in lithium-ion battery technology make it ubiquitous in battery powered devices, ranging from electric terrestrial and aerial vehicles to consumer electronics. Therefore, battery health assessment becomes a crucial problem in providing safety and reliability of the system [1], [2].

Online estimation of battery health, i.e. State of Health (SOH) or Remaining Useful Life (RUL), is not a trivial task because of its nonlinear character and the computing intensive operations it requires. With all the major scientific focus it gained in the last few years, this topic still raises many challenges [3]-[5]. Mathematical models proposed in the state of the art for the battery cannot precisely estimate the battery dynamics, resulting in considerable estimation errors. Combining mathematical and electrochemical models provides better results but increases complexity, while also requiring more computational resources [6]-[8]. Data driven methods, which try to solve the issues of mathematical and electrochemical models, have their own shortcomings [2]. Algorithms such as SVM/RVM need a lot of data for training, to provide good accuracy. Current literature suggests that such algorithms give

very good results for short data windows, while for long data sequences the results tend to degrade [15]. Further on, the problem of the computational complexity is still present.

In recent years, the evolution of deep learning techniques and hardware optimizations gained attention because real-life implementations become feasible. A lot of work is focusing on Recurrent Neural Networks (RNN). In [9]-[13], LSTM (Long-Short Term Memory) algorithms are used to predict the RUL and SOH of the battery. Hybrid versions of LSTM are also presented in [14] and [15]. LSTM fusion with algorithms such as VARMA or Elman neural networks lead to an increase of LSTM prediction accuracy.

The GRU (Gradient Recurrent Unit) is a simplified long-short memory deep learning algorithm [16]. It gained attention because it addresses the problem of vanishing gradients, like LSTM [17]-[18], but has fewer parameters, which results in a more efficient network training and prediction with respect to the required time. Even with fewer parameters, the estimation accuracy of the GRU networks is kept within a good range.

Compared to other battery health estimation methods, the LSTM/GRU techniques have encouraging advantages. They solve one of the main problems of data driven algorithms: the accuracy is good even for long data windows, while the error remains in range and does not degrade. Further on, the input data is not required to be complete. It works even when some intermediate charge/discharge cycles are missing from the training data. The RNN techniques reduce the computational complexity of the online implementation and push this complexity on the (offline) training phase [2], [5]. Also, the latest evolutions of neural network accelerators make hardware implementations of such techniques possible [20]. Open issues of the LSTM/GRU methods include the non-trivial data preparation phase for the network training (e.g. window splitting – data window size has to be chosen accordingly) and how the model parameters are defined (the training algorithm and training configuration have to be chosen with care to obtain good results).

Our work focuses on developing online battery state of health estimation solutions for embedded implementations. Taking into account the advantages of the LSTM/GRU RNN techniques, this article presents a novel GRU-based method for online SOH estimation. The main highlights of the paper are as follows:

- A complete BMS is presented along with the internal structure and configuration parameters;
- The proposed GRU-based method has been implemented and tested on battery data gathered at various temperature values, not only at room temperature, with very good estimation results, as shown in the experimental section;
- The prediction accuracy of the GRU algorithm has been compared with the ones provided by other relevant battery SOH estimation methods, including a LSTM-based technique, also developed by the authors;
- Overall, the recursive neural network algorithms such as GRU and LSTM show good performances, with the exception of the battery regeneration points;
- Compared to LSTM, the GRU algorithm gives slightly higher estimation errors, but within similar prediction error range, while needing significantly fewer parameters (about 25% fewer), thus making it a very suitable candidate for embedded implementations.

The paper is structured as follows: Section 2 provides a coverage of the GRU unit and how it works; a detailed description of the online SOH estimation method is given in Section 3; the experimental validation of the proposed method is presented, compared and discussed in Section 4, followed by the final conclusions.

## 2. Gated Recurrent Neural Network - GRU

The GRU neural network is based on gated recurrent unit and was introduced in [16] to address the issues of gradient vanishing and gradient explosion of classic recurrent neural network unit when dealing with long-term dependencies between neurons. The cell structure was simplified compared to LSTM by removing the cell state and therefore it uses the hidden state for data transfer.

GRU neural network uses two gates – the update gate to control the amount of past information which should be taken into consideration, and the reset gate – to control the amount of information which will be left behind. The general structure of the GRU unit is presented in Fig. 1.

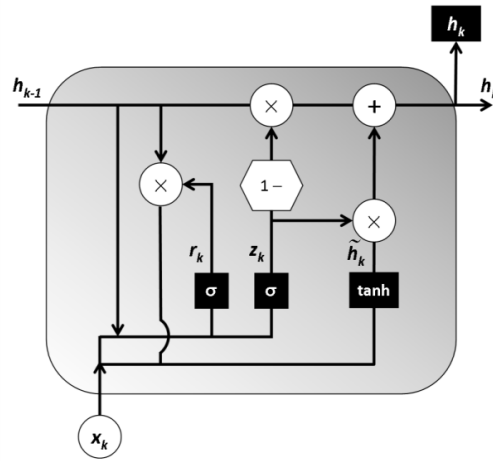


Fig. 1. GRU structure

The hidden state at one instance of time  $k$  depends on the previous state at  $k-1$  time instance and a candidate state,  $\tilde{h}_k$ , using a linear relation [19]:

$$h_k = (1 - z_k) \cdot h_{k-1} + z_k \cdot \tilde{h}_k \quad (1)$$

where  $z_k$  is the update gate defined by the following activation equation:

$$z_k = \sigma(\mathbf{W}_z \cdot [h_{k-1} \ x_k]) \quad (2)$$

The value of  $z_k$  controls how much the current state depends on the candidate state or on the previous state. The candidate state of a GRU is defined by the following equation:

$$\tilde{h}_k = \tanh(\mathbf{W}_h \cdot [r_k \cdot h_{k-1} \ x_k]) \quad (3)$$

where  $r_k$  is the reset gate:

$$r_k = \sigma(\mathbf{W}_r \cdot [h_{k-1} \ x_k]) \quad (4)$$

The reset gate controls how much the candidate state depends on the previous hidden state.  $\mathbf{W}_z$ ,  $\mathbf{W}_h$ ,  $\mathbf{W}_r$  are the weights matrices for update gate, candidate state and reset gate [21].

### 3. Online SOH Estimation Method Based on GRU

#### 3.1. SOH Definition

In literature there are several definitions for battery state of health. For simplicity we used the SOH [%] defined as:

$$SOH [\%] = \begin{cases} 100\%, & \frac{C_{bat}}{C_{nom}} > 1 \\ \left(1 - \frac{C_{bat}}{C_{nom}}\right) \cdot 100 [\%] & \\ 0\%, & \frac{C_{bat}}{C_{nom}} < SOH_0 \end{cases} \quad (5)$$

where  $SOH$  [%] is the battery state of health,  $C_{bat}$  is the current battery capacity,  $C_{nom}$  is the battery nominal capacity,  $SOH_0$  is the chosen ratio related to battery nominal capacity for which SOH is 0 [%]. A common value for  $SOH_0$  is 0.2 or 0.3.

#### 3.2. SOH Estimation Method

The neural network architecture proposed for SOH prediction is presented in Fig. 2. It consists of an input layer, two GRU layers and a dense layer which is a regular neural network layer of connected neurons. The GRU layers are stacked and each GRU unit is interconnected with the neighbor units. The dense layer gathers all the outputs from the last GRU layer and outputs a single value which is the prediction value. The input of the network is denoted with  $C_t$ , which is the battery capacity at cycle  $t$ . With  $C_{pt}$  we have denoted the predicted capacity for the next cycle ( $t+1$ ). The network is designed for univariate predictions.

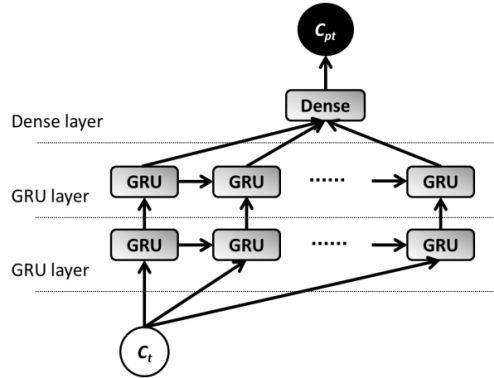


Fig. 2. Architecture of the proposed neural network for SOH estimation

We used the Adam (*Adaptive moment estimation*) optimization algorithm [22] for training the described network, because it combines the advantages of both AdaGrad and RMSProp [23]: it keeps a per-parameter learning rate which is important (performance wise) when the gradients are sparse. The per-parameter learning rates are updated based on the average of recent values of the gradients.

The algorithm stores an exponentially decaying average of past squared gradients  $v_t$  like Adadelta and RMSProp, and also keeps an exponentially decaying average of past gradients  $m_t$ , similar to the momentum algorithm [23].

The decaying averages of past gradients and past squared gradients are the following:

$$\begin{cases} m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \\ v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \end{cases} \quad (6)$$

where  $m_t$  and  $v_t$  are the estimates of the first moment and the second moment of the gradients. During the initial time steps and at initialization, it can be observed that the values of  $m_t$  and  $v_t$  are biased towards zero, especially when  $\beta_1$  and  $\beta_2$  are close to 1 (i.e. the exponentially decaying rates of the Adam algorithm are small).

To address these issues, we can compute the bias-corrected second moment estimates:

$$\begin{cases} \hat{m}_t = \frac{m_t}{1 - \beta_1^t} \\ \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \end{cases} \quad (7)$$

Using these equations, we can obtain the Adam update rule as following:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t. \quad (8)$$

where  $\theta$  represents, generically, the model parameters, which are updated at every cycle  $t$ ,  $\eta$  is the learning rate and  $\varepsilon$  is a very small constant to avoid division by zero (usually  $10^{-8}$ ).

Concerning performance, the Adam algorithm excels at speed and relatively low memory consumption.

A typical BMS system with an online SOH estimator using GRU is presented in Fig. 3 and is composed of the following main blocks:

- Battery SoC determination block – is used to compute the current battery capacity based on battery parameters such as battery voltage, battery current and battery temperature. It provides the input for data preparation block. This can be implemented with one of the available methods and it is out of the scope of this paper;
- Data preparation block – prepares data to be used in the neural network block: normalization, data sequencing;
- Neural network block – outputs a predicted battery capacity value for the next cycle;
- Battery SOH determination block – transforms the predicted battery capacity value into the predicted SOH value using denormalization and based on the battery nominal capacity.

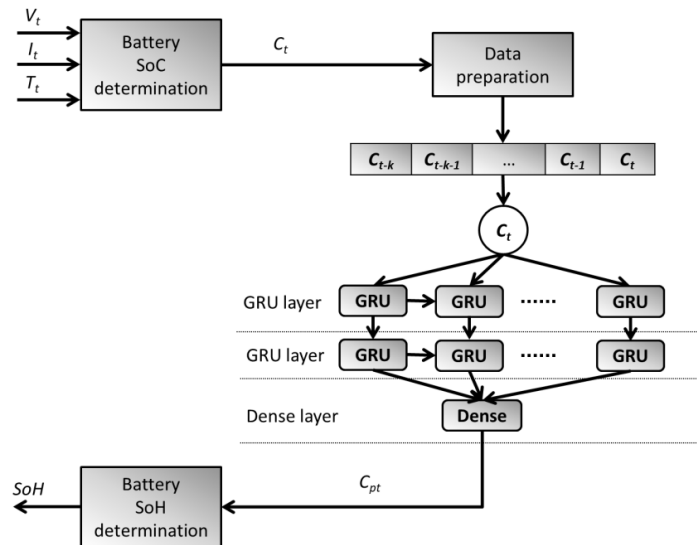


Fig. 3. Battery management system based on GRU neural network

The algorithm for computing the online SOH[%] at each cycle  $t$  is summarized below:



```

1: while not (battery cycle completed) do
2:   wait
3: end while
4: determine SoC
5: Prepare data ( $C_{t-k}, C_{t-k+1}, \dots, C_{t-1}, C_t$ )
6: Calculate  $C_{pt}$ 
7: Calculate  $SOH_t$ 
8: if  $C_{pt}/C_{nom} > 1$ 
9:    $SOH_t[\%] = 100\%$ 
10: else if  $C_{pt}/C_{nom} < SOH_0$ 
11:    $SOH_t[\%] =$ 
11: else
12:    $SOH[\%] = \begin{cases} 100\%, \frac{C_{bat}}{C_{nom}} > 1 \\ \left(1 - \frac{1 - \frac{C_{bat}}{C_{nom}}}{SOH_0}\right) * 100[\%] \\ 0\%, \frac{C_{bat}}{C_{nom}} < SOH_0 \end{cases}$ 
13: endif

```

Fig. 4. Pseudocode to compute online SOH

## 4. Experimental Results

### 4.1. Network Training

Our network configuration is presented in the following table. It has a number of 50 units in hidden layer and one dense unit in the regular layer.

To train and test the neural network, we used the experimental battery data set provided by the Prognostics Data Repository at NASA Ames Research Center, considered one of the reference sources for Li-ion batteries in the field. More specifically, we used the data from the batteries in the Annex section at the end of the paper (Table VI).

For example, the data for battery B0005 corresponds to the following experimental setup: the battery has been run on 3 operational profiles, in a cyclic manner (charge, impedance, discharge), until the battery reached the end-of-life criteria – a 30% drop of its capacity (i.e. from 2 Ahr to 1.4 Ahr). Charging has been performed with constant current (CC) at 1.5 A, until the battery voltage reached 4.2 V and then continued with constant voltage (CV), until the charge current dropped to 20 mA. Each discharge cycle was executed at a constant current (CC) of 2 A, until the battery voltage decreased to 2.7 V. The impedance measurement cycles are based on the electrochemical impedance spectroscopy (EIS) method, using a frequency sweep from 0.1 to 5000 Hz.

In addition to battery data at normal operating temperature (24-25 degrees Celsius), our evaluations consider also different operating temperature ranges available on the NASA data set (see Table VI): 44 degrees Celsius and 4 degrees Celsius.

First, we have prepared the data set, splitting it into training (80%) and test data (20%). Then we spitted the data sequences using a window size of 5. On the training dataset, we performed k-fold cross-validation with parameter  $k = 10$ . We used the R-squared coefficient (Pearson flavor) to analyze the dataset homogeneity. Table II presents all the R-squared values for each fold. The detailed R-squared plots for folds 3 and 4 are illustrated in Fig. 5 and Fig. 6, respectively. Network training was performed in a number of 150 of epochs on a PC with Intel core i7 processor and Nvidia GTX1050T, with 8GB RAM. To ensure better regularization we used a dropout value of 0.5 [25].

**Table I.** Neural network training parameters

Network parameter	Value
No. of units in layer 1	50
No. of units in layer 2	50
No. of epochs	150
History window size	5
Dropout	0.2
Training data	80%
Test data	20%

**Table II.** R-squared values of the 10-fold cross-validation of training dataset

Fold no.	R2 value
1	0.9728
2	0.9737
3	0.9377
4	0.9835
5	0.9813
6	0.9456
7	0.9912
8	0.9812
9	0.9791
10	0.9731

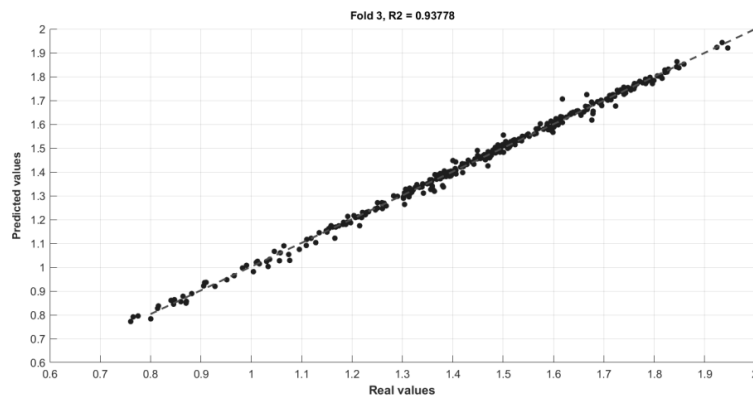


Fig. 5. R-squared plot for fold 3 of the 10-fold cross-validation of the training dataset

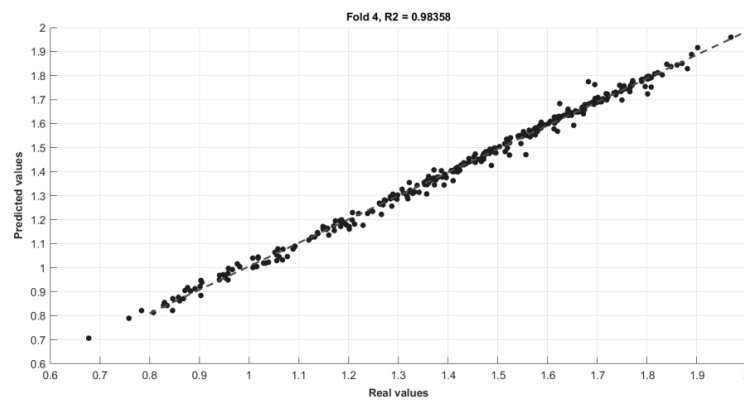


Fig. 6. R-squared plot for fold 4 of the 10-fold cross-validation of the training dataset

For comparison purposes, we also developed a network based on LSTM units. In the training phase, the GRU and LSTM performance are close, with same value of MAE (Mean Absolute Error).

Table III. Training MAE of LSTM versus GRU networks

LSTM vs. GRU		
Total no. of parameters	30651	23001
Training MAE	0.0270	0.0264

It can be observed that the number of parameters for GRU is with 25% less than the number of the parameters for LSTM. This enables the training speedup of the neural network and its prediction results.

#### 4.2. Estimation Error and Comparison with LTSM

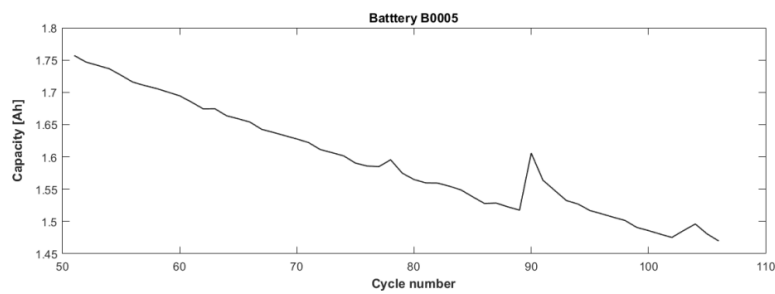
For error estimation we used percentage absolute error as a metric for comparison. The error range for GRU and LSTM is presented in Table IV. We have observed that the peak errors are due to the regeneration phenomenon which affects this type of batteries [26]. For example, one regeneration point in the B0005 battery dataset is presented in Fig. 7. If exclude these errors are excluded, a very good estimation error range can be obtained, as presented in Table V.

**Table IV.** Absolute estimation error range

Absolute error [%] range		
LSTM	-5.70	5.32
GRU	-5.79	5.46

**Table V.** Absolute estimation error range (excluding regeneration points)

Absolute error [%] range (excluding regeneration points)		
LSTM	-2.45	2.56
GRU	-2.91	2.73



**Fig. 7.** Regeneration point of capacity for battery B0005

Fig. 8 to Fig. 11 present in details the percentage absolute error of the SOH prediction for batteries B0005 and B0028, at each charge-discharge cycle. Battery datasets B0005 and B0028 were used for testing the neural network.

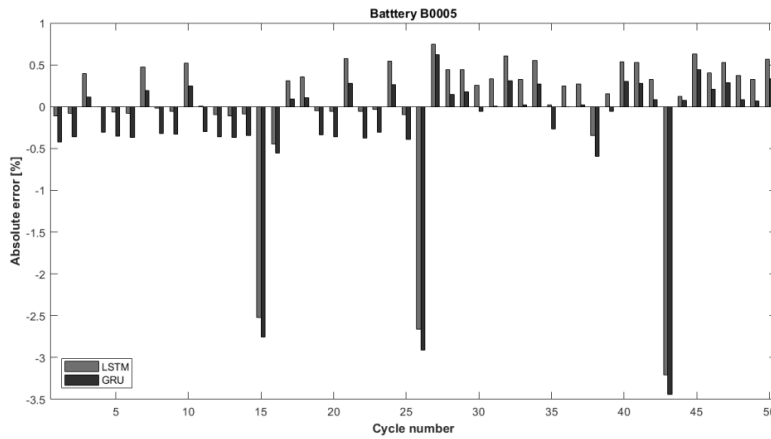


Fig. 8. Battery B0005 absolute estimation error for cycles 1-50

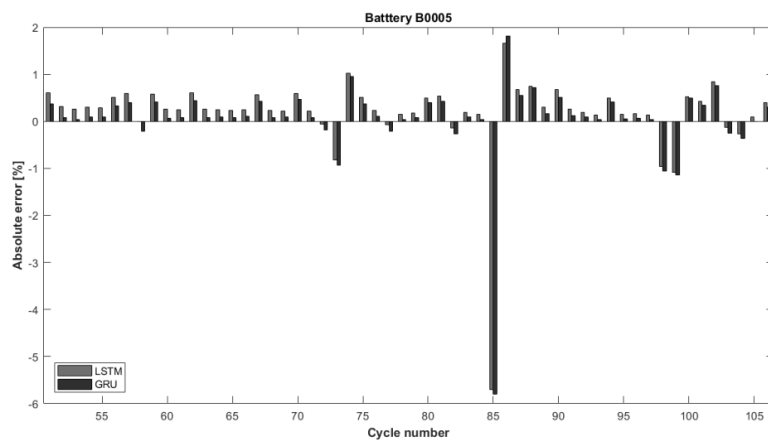


Fig. 9. Battery B0005 absolute estimation error for cycles 51 - 106

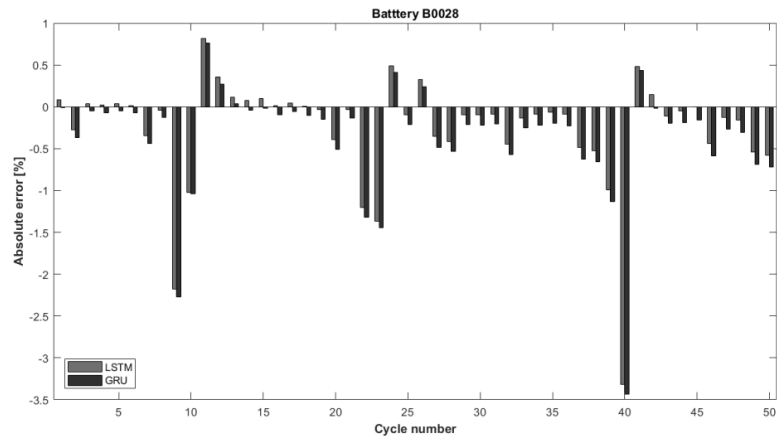


Fig. 10. Battery B0028 absolute estimation error for cycles 1-50

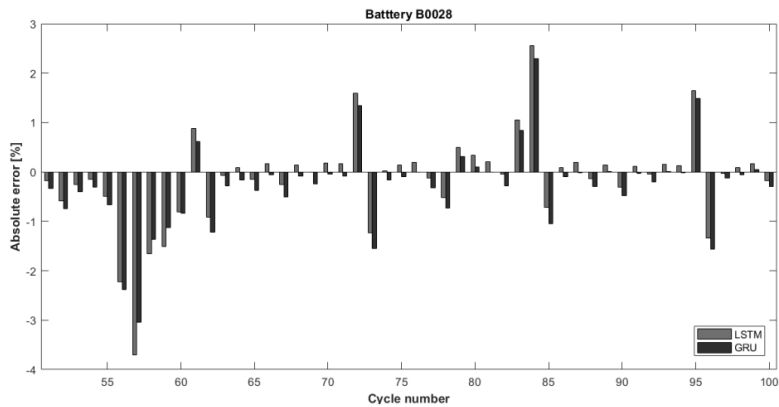


Fig. 11. Battery B0028 absolute estimation error for cycles 51 - 100

We can observe that in most of the cases, the GRU based estimation errors are slightly higher than the ones obtained by using the LSTM algorithm, but within the usual prediction error range for these types of algorithms.

#### 4.3. Comparison with Other SOH Estimation Techniques

To further evaluate the performance of the proposed GRU/LSTM algorithms, we performed a set of extensive comparisons with other state of the art battery SOH estimation methods: second order polynomial regression ("Poly2") [4], exponential regression ("Exp") and multilayer perceptron ("MLP") [27]. For the aforementioned online versions. On each cycle, the parameters for the second order polynomial regression and exponential regression were computed based on the formulas in [28] and [29]. Then the next cycle was predicted based on the obtained polynomial and exponential function. The MLP neural network was trained, validated and tested on the same dataset as used for LSTM and GRU. The architecture of the MLP neural network consists of 2 hidden layers with 50 neurons. The network train method was bayesian regularization. The prediction of the battery capacity for each cycle was performed in the same manner as in the case of LSTM and GRU.

Fig. 12 shows the absolute percentage estimation error ranges obtained for the batteries B0005, B0028, B0032, B0040 and B0045.

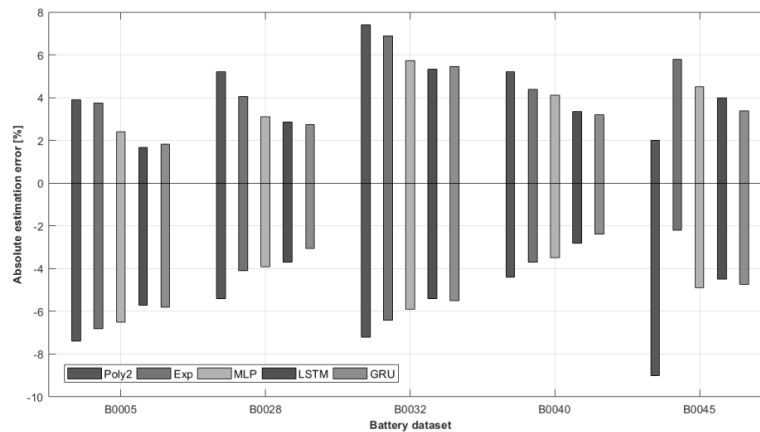


Fig. 12. Comparisons of LSTM and GRU with other relevant battery SOH estimation techniques

Polynomial and exponential regression generate good estimation errors when the battery capacity curve follows a polynomial or exponential trend. But in most of the cases the nonlinearities of the battery capacity outperforms these algorithms, giving an error range of [-7.5%, 4%] respectively [-7%, 3.8%] for battery B0005. The neural network based algorithms show better performance. Starting with the multilayer perceptron, the error falls between -6.5% and 2.5% for battery B0005. The more complex recursive neural network algorithms such as LSTM and GRU estimate the SOH with an absolute percentage error which in most of the cases is half the polynomial and exponential algorithm generated errors. This gives an error range of [-5.5%, 2%] for battery B0005. The improvement of the estimation results is due to the cell structure of the LSTM and GRU cells, which use the long-term and short-term memory.

When the battery capacity curve presents a lot of regeneration points the estimation performance of the LSTM and GRU drops and the performance difference between the other algorithms is not that significant. This is the case of battery B0032. Still, the estimation errors fall into an acceptable range for these types of algorithms [5].

## 5. Conclusions

In this work we present an online SOH estimation method based on GRU neural network units. The proposed method has been implemented and tested on an extended battery dataset, for three different temperature ranges (4, 25 and 44 degrees Celsius).

To assess the estimation performance of the algorithm we used the percentage absolute error, such that the simulation of the method would be closer to real scenarios. Comparisons performed against other relevant estimation methods show the good performances of the recursive neural network algorithms such as LSTM and GRU, with the exception of the battery regeneration points. Such cases must be treated distinctly, with special care.

The estimation errors of the GRU method are very close to the ones obtained with the similar algorithm, LSTM, but on the other hand, it has a significantly smaller number of parameters (~25% less parameters). This makes it a very convenient solution when it comes to training the neural network, due to the major decrease of the necessary processing time. Another advantage is the reduced memory footprint and the acceptable processing power. All these advantages make the proposed GRU-based estimation method a very suitable candidate for embedded implementations.

## 6. References

- [1] C. Zhang, Y. Zhu, G. Dong, J. Wei, "Data-driven lithium-ion battery states estimation using neural networks and particle filtering", *International Journal of Energy Research*, vol. 43, no. 14, pp. 8230-8241, 2019.
- [2] Y. Li, K. Liu, A.M. Foley, A. Zulke, M. Bercebar, E. Nanini-Maury, J. Van Mierlo, H.E. Hoster "Data-driven health estimation and lifetime prediction of lithium-ion batteries: A review", *Renewable and Sustainable Energy Reviews*, vol. 113, 2019.
- [3] M. S. H. Lipu *et al.*, "A review of state of health and remaining useful life estimation methods for lithium-ion battery in electric vehicles: Challenges and recommendations", *Journal of Cleaner Production*, vol. 205, pp. 115-133, Dec 20 2018.
- [4] L. Ungurean, G. Carstoiu, M. V. Micea, V. Groza, "Battery state of health estimation: a structured review of models, methods and commercial devices", *International Journal of Energy Research*, vol. 41, no. 2, pp. 151-181, 2017.
- [5] M. Bercebar, I. Gandiaga, I. Villarreal, N. Omar, J. Van Mierlo, P. Van den Bossche, "Critical review of state of health estimation methods of Li-ion batteries for real applications", *Renewable and Sustainable Energy Reviews*, vol. 56, pp. 572-587, 2016.
- [6] W. Waag, C. Fleischer, D.U. Sauer, "Critical review of the methods for monitoring of lithium-ion batteries in electric and hybrid vehicles", *Journal of Power Sources*, vol. 258, pp. 321-339, 2014.
- [7] A. Seaman, T.-S. Dao, J. McPhee, "A survey of mathematics-based equivalent-circuit and electrochemical battery models for hybrid and electric vehicle simulation", *Journal of Power Sources*, vol. 256, pp. 410-423, 2014.
- [8] Y. Xing, E.W.M. Ma, K.-L. Tsui, M. Pecht, "An Ensemble Model for Predicting the Remaining Useful Performance of Lithium-Ion Batteries", *Microelectronics Reliability*, vol. 53, no. 6, pp. 811-820, 2013.
- [9] Y. Z. Zhang, R. Xiong, H. W. He, M. G. Pecht, "Long Short-Term Memory Recurrent Neural Network for Remaining Useful Life Prediction of Lithium-Ion Batteries", *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 5695-5705, Jul 2018.
- [10] J. T. Qu, F. Liu, Y. X. Ma, J. M. Fan, "A Neural-Network-Based Method for RUL Prediction and SOH Monitoring of Lithium-Ion Battery", *IEEE Access*, vol. 7, pp. 87178-87191, 2019.



- [11] Y. Z. Zhang, R. Xiong, H. W. He, Z. R. Liu, "A LSTM-RNN method for the lithium-ion battery remaining useful life prediction", *2017 Prognostics and System Health Management Conference (Phm-Harbin)*, pp. 1059-1062, 2017.
- [12] C. S. Wang, N. Y. Lu, S. L. Wang, Y. H. Cheng, B. Jiang, "Dynamic Long Short-Term Memory Neural-Network-Based Indirect Remaining-Useful-Life Prognosis for Satellite Lithium-Ion Battery", *Applied Sciences-Basel*, vol. 8, no. 11, Nov 2018.
- [13] Y. T. Wu, M. Yuan, S. P. Dong, L. Lin, Y. Q. Liu, "Remaining useful life estimation of engineered systems using vanilla LSTM neural networks", *Neurocomputing*, vol. 275, pp. 167-179, Jan 31 2018.
- [14] A. C. Caliwag, W. Lim, "Hybrid VARMA and LSTM Method for Lithium-ion Battery State-of-Charge and Output Voltage Forecasting in Electric Motorcycle Applications", *IEEE Access*, vol. 7, pp. 59680-59689, 2019.
- [15] X. Y. Li, L. Zhang, Z. P. Wang, P. Dong, "Remaining useful life prediction for lithium-ion batteries based on a hybrid model combining the long short-term memory and Elman neural networks", *Journal of Energy Storage*, vol. 21, pp. 510-518, Feb 2019.
- [16] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation", Doha, Qatar, Association for Computational Linguistics, 2014, pp. 1724-1734.
- [17] S. Hochreiter, J. Schmidhuber, "Long short-term memory", *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, Nov 15 1997.
- [18] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, J. Schmidhuber, "LSTM: A Search Space Odyssey", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222-2232, Oct 2017.
- [19] F. F. Yang, W. H. Li, C. Li, Q. Miao, "State-of-charge estimation of lithium-ion batteries based on gated recurrent neural network", *Energy*, vol. 175, pp. 66-75, May 15 2019.
- [20] M. Modasshir, A.Q. Li, I. Rekleitis, "Deep Neural Networks: A Comparison on Different Computing Platforms", *15th IEEE Conference on Computer and Robot Vision (CRV)*, Toronto, Canada, 2018.
- [21] R. X. Zhao, P. J. Kollmeyer, R. D. Lorenz, T. M. Jahns, "A Compact Unified Methodology via a Recurrent Neural Network for Accurate Modeling of Lithium-Ion Battery Voltage and State-of-Charge", *2017 IEEE Energy Conversion Congress and Exposition (ECCE)*, pp. 5234-5241, 2017.
- [22] D. Kingma, J. Ba, "Adam: A Method for Stochastic Optimization", *International Conference on Learning Representations*, San Diego, USA, arXiv:1412.6980, Dec 2014.
- [23] S. Ruder, "An overview of gradient descent optimization algorithms", arXiv:1609.047470, Sep 2016.
- [24] B. Saha, K. Goebel, "Battery Data Set", NASA Ames Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA, 2007, [Online: <https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/>], Accessed: Nov. 2019].
- [25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting", *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929-1958, 2014.
- [26] D. Liu, J. Pang, J. Zhou, Y. Peng, M. Pecht, "Prognostics for state of health estimation of lithium-ion batteries based on combination Gaussian process functional regression", *Microelectronics Reliability*, vol. 53, no. 6, pp. 832-839, 2013.
- [27] J. Kim, J. Yu, M. Kim, K. Kim, S. Han, "Estimation of Li-ion Battery State of Health based on Multilayer Perceptron: as an EV Application", *IFAC-PapersOnLine*, vol. 51, no. 28, pp. 392-397, 2018.
- [28] E.W. Weisstein, "Least Squares Fitting-Polynomial", MathWorld - A Wolfram Web Resource, [Online: <http://mathworld.wolfram.com/LeastSquaresFittingPolynomial.html>], Accessed: Nov 2019].
- [29] E.W. Weisstein, "Least Squares Fitting-Exponential", MathWorld - A Wolfram Web Resource, [Online: <http://mathworld.wolfram.com/LeastSquaresFittingExponential.html>], Accessed: Nov. 2019].

## 7. Annex

Table VI. Battery data used for the training and testing of the proposed neural network

Battery ID	Operating conditions
B0005, B0006, B0007, B0018	Room temp., charge current 1.5 A, Final charge voltage 4.2 V Constant discharge current 2 A, Final discharge voltage 2 V
B0025, B0026, B0027, B0028	Room temp., charge current 1.5 A, Final charge voltage 4.2 V Discharge square wave 0.05 Hz, amplitude 4 A, duty cycle 50% Final discharge voltage 2 V
B0029, B0030, B0031	Room temp., charge current 1.5 A, Final charge voltage 4.2 V Constant discharge current 4 A, Final discharge voltage 2 V
B0033, B0034, B0036	Room temp., charge current 1.5 A, Final charge voltage 4.2 V Constant discharge current 2 A, Final discharge voltage 2 V
B0038, B0039, B0040	Room temp. and 44 deg Celsius, charge current 1.5 A, Final charge voltage 4.2 V Constant discharge current 2 A, Final discharge voltage 2 V
B0041, B0042, B0043, B0044	4 deg. Celsius., charge current 1.5 A, Final charge voltage 4.2 V Constant discharge current 1 A, 4 A
B0049, B0050, B0051, B0052	4 deg. Celsius., charge current 1.5 A, Final charge voltage 4.2 V Constant discharge current 2 A
B0053, B0054, B0055, B0056	4 deg. Celsius, charge current 1.5 A, Final charge voltage 4.2 V Discharge square wave 0.01 - 5000 Hz, amplitude 2 A, duty cycle 50% Discharge until final capacity is 30% bellow nominal capacity

## **ANEXA 4. ARTICOL 'ONLINE STATE-OF-HEALTH ASSESSMENT FOR BATTERY MANAGEMENT SYSTEMS'**

Articol 'Online State-of-Health Assessment for Battery Management Systems', în variantă pre-print, apărut în *Ieee Inst. And Meas.*, vol. 60, no. 6, pp. 1997-2006, Jun. 2011

## Battery Management System with Online State-of-Health Estimation

Mihai V. MICEA, *Member, IEEE*, Lucian UNGUREAN, Gabriel N. CĂRSTOIU, Voicu GROZA,  
*Senior Member, IEEE*

**Abstract**—Battery powered embedded systems have known a rapid evolution in recent years, as Nickel-metal hydride (Ni-MH) battery technology has enabled important reductions in size and proportional increases in total capacity over older battery technologies such as Nickel-Cadmium (Ni-Cd) and Lead-Acid. This paper addresses the problem of State-of-Health (SoH) estimation and prediction for use in resource-constrained Ni-MH battery powered embedded systems. We propose a novel SoH prediction methodology, presenting both a theoretical analysis of the estimation algorithm as well as the detailed description of a hardware and software implementation. Two versions of estimation algorithms are proposed, along with the analysis of their performance in terms of prediction accuracy and required processing power, as the SoH prediction is designed to run online, being part of an embedded battery management system (BMS).

**Index Terms**—Battery management, State-of-Health prediction, Ni-MH, battery powered device.

### I. INTRODUCTION

THE number of battery powered mobile and portable devices has seen an exponential growth in recent years, among the most obvious examples being: laptops, mobile phones, PDAs, smart sensors [1], [2], etc. The processing power of these devices has also been rapidly growing, thus determining the development and evolution of both primary and secondary battery technologies needed to support longer usage times between recharges and/or replacements of batteries.

NiMH technology offers a lower cost compared to Lithium-ion (Li-Ion) based solutions in many applications, because it provides a high storage capacity (up to 2700 mAh for standard AA cells) and high energy density, it contains chemical elements which do not have a bad environmental impact, and features good charging and discharging rate capabilities [3], [4]. Another advantage of Ni-MH batteries worth mentioning is the good tolerance to fast charging (using charging currents up to 1C). The main disadvantage of using this technology is its low robustness when it comes to extreme conditions which affect the battery storage capacity and therefore its performance [5]. Another problem remains the self discharge rate which isn't negligible but it can be quantified and used in calculations.

All of the afore-mentioned advantages favor the usage of Ni-MH cells in battery powered embedded systems which are resource-constrained in terms of size, processing power and battery capacity. Although most consumer handheld and portable devices use Li-Ion batteries as their energy source, Ni-MH batteries can still be found in devices such as power tools, walkie-talkies, DECT phones, GPS receivers, digital cameras, etc. Furthermore, Ni-MH is the preferred energy source for smart sensor nodes used in monitoring and surveillance applications.

#### A. Previous work

Previous research papers in the field of battery management systems for Ni-MH cells have targeted areas such as charge termination techniques [6], accurate State-of-Charge (SoC) determination [4], innovative charger designs [7], [8], and discharge coordination algorithms to improve battery lifetime [9].

Considering the area of charge termination techniques, many of the novel developments that have been published thus far have focused on improving certain characteristics of the already-known and accepted techniques in common use for Ni-MH batteries

Manuscript received July 21, 2010. This work was supported in part by the Romanian Ministry of Education and Research under Grant PNCDI II ID-22/2007-2010.

M. V. Micea, G. N. Carstoiu, and L. Ungurean are with the Department of Computer and Software Engineering, "Politehnica" University of Timisoara, Timisoara 300223, Romania (e-mail: mihai.micea@cs.upt.ro).

V. Groza is with the School of Information Technology and Engineering, University of Ottawa, Ottawa, ON K1N 6N5, Canada.

[6], such as: the rate of change of the battery terminal voltage ( $dV$  slope and voltage plateau detection), maximum rate of change in battery temperature ( $dT/dr$ ).

Among the notable ideas are the ones presented by Diaz and his co-authors in [8], in which they describe a charge termination technique based on detection of the battery voltage second slope, effectively stopping the charging process as soon as the time windows between consecutive increases in terminal voltage grew over a certain time threshold.

SoC determination is one of the key issues relating to battery powered devices, and, throughout the history of the development of secondary battery cells and their applications, several systems and techniques have been specified and developed [4]. Many recent developments in the field have concentrated on batteries with Alkaline [10], Lead-Acid [11] and Li-Ion chemistries. The authors of [12] and [13] have proposed and demonstrated an ingenious solution for improving the accuracy of SoC determination by combining the classical coulomb counting method with a SoC correction algorithm, based on a second order Randle model of the batteries. However, the proposed algorithm is based on offline parameter identification and assumes a fixed lookup table relating the open-circuit voltage (OCV) to the SoC, which is scarcely true for real batteries.

Research work focusing on smart charger designs for Ni-MH batteries has been growing in recent years [7], [8], as these types of batteries are very sensitive to charge and discharge termination conditions, and also as a result of more demanding requirements for reduced charging times and increased charging efficiency. The approach in [6] focuses on the fast charging process, providing a limited degree of intelligence by determining what amplitude of charge current is required (fast or trickle) by using two levels of discrimination for the battery terminal voltage ( $V_{batt}$ ), at 10% of  $V_{batt}$ , and 80% of  $V_{batt}$ . The design proposed in [8] implements the charging algorithm using an 8-bit microcontroller and is based on generating current pulses and observing the battery terminal voltage. It offers a certain degree of flexibility by allowing the user to select between 4 predefined nominal capacities for the batteries used.

A complete charger system with SoC measurements and a battery capacity learning feature is presented in [14]. The authors focus on minimizing the charging system power consumption and succeed in attaining a power envelope of under 100  $\mu$ A. However, the reported accuracy of measurements is not supported by experimental results and the hardware implementation is costly due to its large number of discrete analog components.

#### B. Proposed design

The main idea of this paper is to present in detail a complete battery management system, which can be utilized standalone or as part of a battery powered embedded device. We will discuss the novel features introduced by our design and will evaluate the validity of our implementation by presenting a case study and detailed experimental results.

Our solution provides in-system charging of the attached battery pack, SoC calculation and SoH prediction of the remaining number of cycles or the remaining useful life (RUL), as defined by [15]. Similar to other intelligent charger designs [7], our charge controller module makes use of a combination of charge termination techniques, relying mainly on voltage and temperature based methods, as well as on the accurate measurement of the total supplied energy via a hybrid coulomb-counting algorithm.

One of the novelties introduced by this paper is the integration of the battery pack, battery charge and discharge controller, SoC measurement hardware and the battery management software into a single module, thus providing a configurable black-box solution ready to be integrated into battery powered devices. We will present the implementation details of our complete hardware-software solution, as well as a case study implementation in Section III of the paper.

The ability of an intelligent battery management system to accurately measure the total energy supplied to and removed from the battery pack is mandatory for correct SoH determination. Our design implements SoC calculation based on a reduced version of the mixed algorithm proposed by Codeca et al. in [12], in which the authors demonstrate that very accurate SoC measurements can be obtained when combining the classical coulomb-counting approach with feedback from an electrical battery model, based on the OCV-SoC dependency. The novelty introduced by our solution, as detailed in sections II and III, consists in the online recalculation of the OCV-SoC model parameters, as opposed to the static and offline model identification presented in [12]. This idea is based on the observation that the battery model parameters need to be modified as each battery cycle is consumed, to take into account the subsequent SoH deterioration caused by battery ageing.

The SoH prediction capability is central to our design and represents another novel feature. It provides the users with a vital piece of information about the health and the remaining operating cycles of the integrated battery pack, allowing them to know in advance when to replace the used cells. Although several SoH estimation algorithms have been reported in the art ([15], [16]), their complexity makes them hardly suitable to be implemented in a low cost and low power embedded system, and, to our knowledge, no such attempts have been reported.

Our proposed SoH estimation and prediction solution is based on a second order parabolic regression algorithm, and in sections III and IV we will elaborate on the numerical methods that we have found suitable to be implemented on an embedded processor. We will analyze two proposed implementation concepts in terms of relative estimation error versus actual battery data, and CPU and memory usage of the embedded implementation case study.

Section V draws the conclusions and summarizes the obtained results.

II. THEORY OF OPERATION

A. SoC calculation principles

The state of charge is an important parameter for all battery powered devices, as it is used to provide an indication of the remaining operating time for the current discharge cycle or the remaining time until the batteries are charged, in the charging phase. Although many methods exist for SoC indication [4], [10], [12], the most widely studied and most used ones are: coulomb counting, which is relatively simple to implement and can be used with all battery types; impedance spectroscopy, which can also be generally used, but is expensive to implement; fuzzy logic approaches, which can provide accurate estimates but present large memory requirements; Kalman filters, which are robust and adaptable to multiple battery types, but require a complex battery model and are difficult to implement.

There are multiple ways in which SoC can be defined, but the most commonly accepted definition is the following: the SoC of a battery pack is the maximum remaining useful capacity stored in the respective battery pack. SoC can also be described as a percentage value relative to the nominal capacity of the battery pack, expressed in Ampere-hours (Ah):

$$SoC(t) = \frac{Q_{nom} - \int_0^t I(t)dt}{Q_{nom}} \cdot 100 \quad (1)$$

In (1), the state of charge is denoted by  $SoC(t)$  and it can hold a value from 0 to 100, the nominal capacity is  $Q_{nom}$ , and the current flowing to/from the batteries is denoted by  $I(t)$ , and can have a positive value in the charge phase and a negative value during discharge.

A direct consequence of the SoC definition is that the SoC can be determined by accumulating the changes in the charge level, effectively integrating the current flow over the operating time. This book-keeping method is known as coulomb counting. Although its implementation is simple and suitable for a microcontroller based battery management system (BMS), this method presents some drawbacks: its performance is directly dependent on the accuracy of current measurements, requiring very precise current sensing, and significant offset errors can be introduced if the initial SoC ( $SoC(0)$ ) is not known precisely [13]. A more accurate solution is presented in [12]. Here, the authors make use of a second order resistor-capacitor Randle model. The model inputs are the load current and an OCV-SoC lookup table. The model parameters can be identified by measuring the voltage levels and settling times, after applying charge and discharge current pulses. The model identification and the OCV-SoC lookup table determination are done offline using a PC-controlled test-bench.

Our proposed SoH estimation algorithms need accurate SoC measurements as inputs; therefore, we have implemented an adaptation of the mixed algorithm described above. The applications targeted by our BMS design will use currents in the range of 200 to 1000 mA. In the field of battery engineering, this range corresponds to an interval of 0.1 to 0.5 C of a 2000 mAh battery pack, for which the rated discharge current of 1 C is equal to 2000 mA. As a consequence, according to [5], the maximum available SoC does not depend on the charge or discharge current applied to the batteries. Furthermore, the OCV-SoC curve changes as the batteries advance in age, leading to the observation that the OCV and the maximum available SoC are functions of the battery SoH.

Thus we propose an adaptation of the mixed algorithm for SoC determination described in [11]. Its most important and novel feature is the possibility of determining and recalculating the battery model parameters online, as part of the system operating time. We have considered a simplified first order Randle model for the Ni-MH batteries, shown in Fig. 1, in order to make it feasible to be implemented as part of an embedded BMS and to shorten the testing times necessary for identifying its parameters.

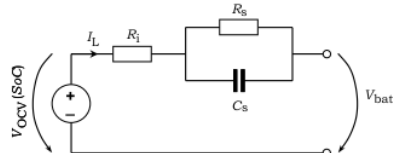


Fig. 1 Proposed Ni-MH electrical battery model.

The proposed model expresses the battery terminal voltage  $V_{batt}$ , considering a load current  $I_L$ , as a function of the battery open circuit voltage  $V_{OCV}(SoC)$ . The battery overpotential ( $V_{OCV}(SoC) - V_{batt}$ ) is modeled by the voltage drop across its internal resistance  $R_i$  and the response to a step current signal is handled by the parallel  $R_s$ - $C_s$  group.

$$V_{\text{batt}}(t) = V_{\text{OCV}}(\text{SoC}) - I_L \left( R_1 + R_s \left( 1 - e^{-\frac{t}{R_s C_s}} \right) \right) \quad (2)$$

In the Laplace domain, (2) can be represented with the following relation:

$$V_{\text{batt}}(s) = V_{\text{OCV}}(\text{SoC}) - I_L \left( R_1 + \frac{R_s}{1 + s R_s C_s} \right) \quad (3)$$

The aforementioned relations link the directly measurable parameters,  $V_{\text{batt}}$  and  $I_L$ , to the model parameters and the OCV, where  $t$  represents the elapsed time of the current cycle. This relation will be used by the BMS software implementation to periodically recalculate the model parameters as needed.

#### B. SoH estimation

The State of Health (SoH) is a metric which indicates the battery condition related to a new battery. SoH determination is not a simple task. In most cases several parameters are involved in this process, like cycle numbers, accurate SoC determination, etc. There is a strong dependency between SoH and both the battery and the type of applications running on the battery powered device. This degree of unpredictability can be controlled using adaptive systems based on neural networks, Kalman filtering or fuzzy logic [17]. The problem with these systems is the high computational power which is needed for implementation, unfeasible for low cost consumer smart chargers.

A simple approach is presented in [17]. The SoH indication is based on the stored maximum capacity function,  $C_k$ , for some charge/discharge cycle,  $k$ . This paper focuses on the improvement of this paradigm by introducing curve modeling and estimation using polynomial regression. We consider that this simple algorithm is well suited for consumer smart chargers with small computational power, as we show in the following sections.

A least-square approach is proposed to implement the second order polynomial regression which estimates the  $C_k$  function.

The polynomial representation of  $C_k$  is given by:

$$C_k = ak^2 + bk + c, \quad a < 0 \quad (4)$$

The constraint imposed to  $a$ , ensures a better curve approximation, because of its monotonically decreasing part.

Consider a battery operating scenario with a total of  $n$  charge/discharge cycles completed up to the current moment. As a result, the maximum capacity value  $C_k$  for each successive battery cycle can be obtained. The curve estimation using least squares algorithms is determined by solving the following system of equations:

$$\begin{cases} a \sum_k k^2 + b \sum_k k + cn = \sum_k C_k \\ a \sum_k k^3 + b \sum_k k^2 + c \sum_k k = \sum_k k C_k \quad .k = \overline{1, n}, n \geq 3 \\ a \sum_k k^4 + b \sum_k k^3 + c \sum_k k^2 = \sum_k k^2 C_k \end{cases} \quad (5)$$

To solve the system and to obtain the  $a$ ,  $b$ , and  $c$  parameters, a simple algorithm, such as Cramer's, can be used. It consists of computing the following determinants:

$$\Delta = \begin{vmatrix} \sum_k k^2 & \sum_k k & n \\ \sum_k k^3 & \sum_k k^2 & \sum_k k \\ \sum_k k^4 & \sum_k k^3 & \sum_k k^2 \end{vmatrix} \quad (6)$$

$$\Delta_a = \begin{vmatrix} \sum_k C_k & \sum_k k & n \\ \sum_k k C_k & \sum_k k^2 & \sum_k k \\ \sum_k k^2 C_k & \sum_k k^3 & \sum_k k^2 \end{vmatrix}$$



$$\Delta_b = \begin{vmatrix} \sum_k k^2 & \sum_k C_k & n \\ \sum_k k^3 & \sum_k k C_k & \sum_k k \\ \sum_k k^4 & \sum_k k^2 C_k & \sum_k k^2 \end{vmatrix}$$

$$\Delta_c = \begin{vmatrix} \sum_k k^2 & \sum_k k & \sum_k C_k \\ \sum_k k^3 & \sum_k k^2 & \sum_k k C_k \\ \sum_k k^4 & \sum_k k^3 & \sum_k k^2 C_k \end{vmatrix}$$

In this case the values of the  $a$ ,  $b$ , and  $c$  parameters are expressed using:

$$a = \frac{\Delta_a}{\Delta}, b = \frac{\Delta_b}{\Delta}, c = \frac{\Delta_c}{\Delta}. \quad (7)$$

For calculating the sums in the determinants specified in (6), the following recurrent formulae are suited for embedded implementation:

$$S_{k+1} = S_k + (k+1), S_{(k+1)^2} = S_{k^2} + (k+1)^2, \text{ etc.} \quad (8)$$

Using the polynomial function obtained, we can compute the first cycle,  $m$ , for which the discharged capacity is under a threshold value, denoted as  $F \cdot C_{\text{nominal}}$ , and the SoH is considered 0%.

The computation of  $m$  is reduced to solving the inequality:

$$F \cdot C_{\text{nominal}} > ak^2 + bk + c \quad (9)$$

Since  $a < 0$ ,  $m$  becomes:

$$m = \left\lfloor \frac{-b - \sqrt{b^2 - 4ac}}{2a} \right\rfloor, \quad (10)$$

where  $\lfloor x \rfloor$  (floor) represents the largest integer less than or equal to  $x$ .

The least squares algorithm is well suited for linear systems where the approximation has the highest accuracy. Of course, if the values considered for interpolation present nonlinearities, the accuracy decreases. However, the applications targeted by the proposed BMS design use discharge currents between 200 and 1000 mA, which are well below the maximum rated discharge current (specified by the battery manufacturer datasheets). Thus, nonlinearities relating to the degradation of the battery capacity have a low probability of appearance [5].

The SoH estimation principle that we have adopted for integration in the BMS is based on previously stored measurements of the total available discharge capacity of the battery pack. As a consequence, we have identified two methods for SoH estimation using the available battery data and the number of elapsed battery cycles: the *time-window algorithm* and the *history-based algorithm*.

The *history-based algorithm* for SoH estimation has been devised starting from the observation that the typical Ni-MH cell discharge curve presents a relatively constant decreasing curve over its expected lifetime [18]. This method makes use of the whole charge-discharge history of the cell, up to the current charge/discharge cycle. The basic idea supporting this algorithm is that the estimation will be more accurate as more inputs are used.

The *time-window algorithm* is based on the following idea: considering that the current battery cycle is denoted by  $C_n$ , then the measured capacities of the previous  $n - 1$  cycles ( $C_1$  through  $C_{n-1}$ ) will be used to estimate the remaining  $u$  useful cycles. The value of  $u$  is defined according to the afore-mentioned requirements. As the number of the battery cycles advances, so do both boundaries of the considered time window, providing the estimation algorithm with a moving time-window of the previous  $n - 1$  cycles. Among the advantages of this approach, we can note that in the case of a new battery pack,  $n \ll u$ , and therefore, relatively few logged discharge cycle data are needed to estimate the remaining useful life. For example, experimental results presented in Section IV show that a number of up to  $u = 60-70$  future cycles can be estimated by considering only  $n = 25$  recorded discharge capacities.



Considering the two data gathering approaches for the SoH estimation algorithm, the BMS designer must choose the best suited implementation, taking into consideration such factors as: nominal cell capacity, the number of cells/ pack and the available computing power and memory storage (for keeping the measured battery capacity logs).

### III. BMS IMPLEMENTATION CASE STUDY

The algorithms presented in the previous section have been implemented on an embedded power management module called the PMBoard (Power Management Board). This board is part of the WIT architecture (Wireless Intelligent Terminal), specified by the CORE-TX project (Collaborative Robotic Environment – The Timisoara Experiment) [19], [2]. The PMBoard has several functions: to supply power to the entire system, to measure and manage the power consumption [20] of the entire WIT, to increase the life and state of health of the battery pack, to provide battery system parameters to the upper levels of the system power management (SPM) software.

The management of the power consumption is based on accurate measurements of voltage, current and temperature values. The temperature is measured using a TMP101 sensor [21], the voltage by using a 10 bit SAR ADC [22]; for current measurement a high side monitor schematic has been implemented, as depicted in Fig. 2 for a particular load channel which corresponds to an individual component board of the WIT.

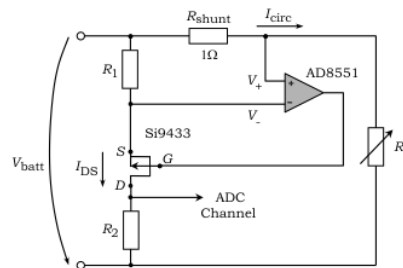


Fig. 2. High side current monitor schematic.

In the above figure,  $R_s$  is the load resistance,  $I_{circ}$  the current through the circuit and  $R_{shunt}$  the shunt resistor. Using high precision components, the implementation of this schematic provides very accurate measurements and reduces the influence of noise. Software averaging [23] is also used for increasing current measurement precision.

Details concerning the power consumption of the monitor schematic are given below. The variable resistor  $R_s$  presented in Fig. 2 denotes the resistive load. The resistor divider composed of  $R_1$  and  $R_2$  is used to adapt the voltage of the ADC channel, which has very high impedance, thus its current draw is in the order of  $\mu A$ . The AD8551 op-amp draws  $750 \mu A$ . The power consumption of the shunt resistor is proportional to the current through the load, and has a maximum value of  $500 mW$ , considering a  $0.5 \Omega$  shunt and the maximum load current of  $1000 mA$ . This power figure can be reduced by decreasing the value of the shunt resistor. ~~N-ar trebui aici macar o valoare totala, in mW?~~ Am introdus consumul pt. shunt. Consumul total al circuitului este aprox. acelasi, pt. ca restul componentelor consuma foarte putin, dupa cum am scris.

SoC determination is done using the model presented in Fig. 1. The system has the ability to identify the model parameters and to recalculate them every time the batteries are replaced. The identification of the system parameters is done by utilizing additional hardware for adjusting the amplitude of charge and discharge pulse currents. ~~The details of this additional logic are beyond the scope of this paper. Ar trebui incluse detaliile cerute de Rwr 3 (?). Nu le vom include, ca sa nu dam de "gandit" celorlalti 2 revieweri si pt. ca nu sunt relevante. I-am explicat in response to reviewer, mai ales ca ceilalti doi nu au cerut asemenea detalii, doar al treilea.~~

The power management software, as a key part of the PMBoard BMS implementation, is schematically represented in Fig. 3. It consists of a modular embedded application which must ensure three layers of functionality:

- Battery State Manager (APPL layer): provides the measurement, storage and processing of the main battery parameters, as described above, as well as the SPM command and response handling.
- Task manager layer: provides the scheduling and execution of the HRT (hard real-time) tasks, according to the HARETICK (Hard REal-Time Compact Kernel) model [24].
- Hardware abstraction layer: contains the drivers and interfaces of the microcontroller unit (MCU) and board peripherals.

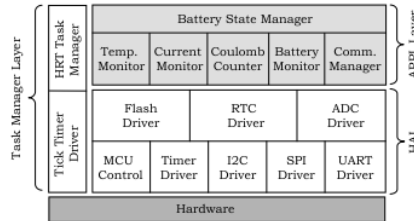


Fig. 3. BMS software layered architecture.

The estimation software component, which is part of the battery state management layer, handles both the estimation of the remaining operating time of the current discharge cycle as well as our novel SoH prediction algorithm. As detailed in Section II, the prediction of the RUL is achieved by first calculating the polynomial coefficients via the parabolic regression, and then solving the second order inequality of (9).

Although this algorithm was chosen as a tradeoff between the implementation complexity and the available computing power, its execution on the 32 bit ARM7 platform had to be designed to tackle some implementation-specific problems such as:

- the magnitude of the intermediate calculation results exceeds the 32-bit boundary and must be represented as 64-bit numbers;
- the MCU architecture offers no native support for floating point operations; therefore the algorithm inputs (e.g. the battery charge measurements) must be scaled and truncated to integers;
- the calculation of the square root of a 64 bit number is required for solving the final inequality, while the standard C library provides a slow, inefficient version of the `sqrt()` function.

One of the main requirements of the BMS software is that its components must be designed to be executed within real-time constraints and that its duty cycle must be minimized in order to reduce the system power consumption. For these reasons, the SoH estimation component must be implemented using all available optimizations, to reduce its execution time and memory footprint.

The identification of the main optimization aspects has been carried out by analyzing the mathematical relations detailed in section II. We have found that the best method of calculating the four third-order determinants, in terms of favoring additions and subtractions over multiplications, is the recursive expansion in cofactors using the first row [25], rather than the classical Sarrus or triangle rules. Using this method, a total of 12 second-order determinants will be generated. Another key observation that we have made is that only 7 of the afore-mentioned determinants are distinct, yielding a significant reduction of the required calculations.

In Section II we have mentioned that an immediate optimization would be to calculate the summation terms recursively, without the need to retain all the previously measured battery charge values, thus reducing memory space. In addition, a significant optimization of the processing power and memory space has been achieved by observing that, out of the total of 24 summation terms implied by the second-order determinants, only 7 are distinct.

As to the square root calculation, we have implemented an iterative successive approximation method, based on the observation that for a given integer number of  $b$  bits, its square root is represented by at most  $b/2$  bits.

Considering all these observations, the implementation of the proposed SoH estimation algorithm for the calculation of the number of RUL cycles ( $n_{RUL}$ ) is presented in Algorithm 1. Here,  $n$  denotes the current number of measured battery cycles,  $n_{min}$  the minimum number of cycles which are necessary for the startup of the algorithm, and  $\Delta_1$  through  $\Delta_7$  are the seven distinct second-order determinants previously mentioned. This requirement applies to both variants of the algorithm and implies that the SoH estimation cannot be performed until at least a number of  $n_{min}$  battery cycles have been recorded. The minimum number of battery cycles is equal to the time-window size and different values can be chosen, as detailed in the next section.

Algorithm 1. RUL estimation algorithm.

```

1: while not (battery cycle completed) do
2:   wait
3: end while
4: increment  $n$ 
5: if  $n \geq n_{min}$  do
6:   Calculate  $Sum_1, Sum_2, \dots, Sum_7$  with (8)
7:   Calculate  $\Delta_1, \Delta_2, \Delta_3$ 
8:   Calculate  $\Delta$ 

```

```

9: Calculate  $\Delta_4, \Delta_5, \Delta_6$ 
10: Calculate  $\Delta_6, \Delta_b$ 
11: Calculate  $\Delta_7, \Delta_c$ 
12: // Scaling of  $\Delta, \Delta_a, \Delta_b, \Delta_c$ 
13:    $\Delta = \Delta/128$ 
14:    $a = (\Delta_s \text{ left shifted } 20 \text{ bits})/\Delta$ 
15:    $b = (\Delta_n/\Delta) \text{ left shifted } 20 \text{ bits}$ 
16:    $c = (\Delta_e/\Delta) \text{ left shifted } 20 \text{ bits}$ 
17: Solve (9) and calculate  $m$  according to (10)
18:  $m_{RUL} = m - n$ 
19: end if

```

The differences between the time-window and the history based algorithms reside in the calculation of the 7 summation terms ( $Sum_1$  to  $Sum_7$ ) and will be discussed in the following sections. Additional optimizations have been introduced and are shown in the lines 12 to 16 of the algorithm. They address the numerical adaptation of the polynomial coefficients ( $a$ ,  $b$  and  $c$ ), due to the fact that we are dealing with large numbers, which need to fit 64 bits, without compromising the accuracy of the calculations.

#### IV. PERFORMANCE EVALUATION

For the performance evaluation we have considered two sets of data. The first set contains the estimated values for the cycle in which the battery capacity drops under a fraction of the nominal capacity, value for which the state of health of the battery is considered 0%. This value is specific for each type of battery and usually it can be found in the documentation provided by the manufacturers. The second set of data contains the real cycle number for which the capacity reaches the terminal SoH value specified earlier.

The data has been obtained from cycling two types of NiMH battery packs: a GP pack with a nominal capacity of approximately 2400 mAh and an aged Sanyo pack designed for 2200 mAh. For evaluation, a factor of 80% of the nominal capacity is considered to be the threshold for which the terminal SoH is reached. Our experimental data shows the respective condition occurs at the 85-th cycle for the GP pack and at the 69-th cycle for the Sanyo pack.

The cycling of the batteries through multiple charge and discharge phases has been accomplished using an automated test setup, constructed using the PMBoard presented in Section III, and the gathered data has been used as input for the aforementioned datasets. The batteries have been configured in packs of two cells connected in series, at a nominal voltage of 2.4V. The average discharge current for the GP battery packs has been 210 mA or 0.09 C (at the rated capacity of 2400 mAh) and 420 mA or 0.19 C (at the rated capacity of 2200 mAh) for the Sanyo packs.

In charge mode, a total of 2300-2600 mAh have been supplied to the battery pack in each cycle, using the constant current charging strategy and terminating the charge at the activation of one of the classical termination techniques, such as the "V plateau" [14]. In discharge mode the battery pack has been automatically switched to supply a programmable resistive DC load, using specialized hardware embedded into the PMBoard to simulate the typical current consumption of a WTT.

##### A. Evaluation of the SoH estimation algorithms

The maximum discharge capacities for the tested battery packs have been recorded and the cycle number at which the 0% SoH point is reached has been noted. A comparative depiction of this data is shown in Fig 4. The cycle number represents the number of the test cycles elapsed since the start of the experiment and not the absolute discharge cycle of the pack, elapsed since first use. Thus we can see that for the Sanyo pack the SoH degradation trend line is more abrupt indicating that the batteries are aged. (? : la ciclul 70: 85.89% pt. Sanyo, vs. 87.55% pt. GP) => la ciclul 70 pt. Sanyo valoarea este de 1760 mAh, care este exact 80% din 2200. Pt. GP am gresit valoarea de 2300, de fapt la ciclul 85 mai avem 1920 mAh, care este 80% din 2400 mAh, nu din 2300. Am corectat si in articol.

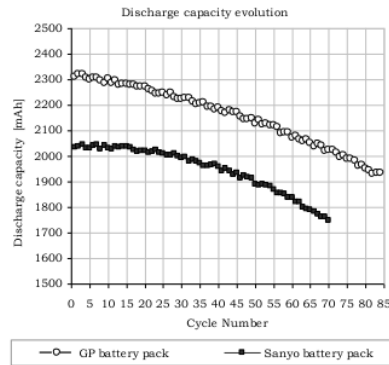


Fig. 4. Discharge capacity evolution for GP and Sanyo battery packs.

Two types of SoH estimation algorithms have been implemented: the time-window algorithm, with a window of size  $W = 25$  and  $W = 30$  cycles, and the history-based technique. In the first case, the estimation has been made using the values recorded in the last 25 and last 30 cycles. For the history based algorithm the estimation is made based on all the recorded values, up to the current cycle, on which the estimation is computed.

In the following performance evaluation of the algorithms, we compare the theoretical results obtained from executing the algorithms in a Matlab environment, with the experimental results obtained from running the embedded implementation.

The experimental results obtained during the performance evaluation of the embedded algorithm implementation on the GP batteries are represented in figures 5 and 6. The embedded implementation is different from Matlab because of the integer arithmetic which causes rounding errors. There is also the problem of number representation. To suppress the rounding errors, the algorithms on the embedded side are implemented using 64 bit integer representation.

As shown in Fig. 5, the history based algorithm estimation error approaches zero starting from the 60th execution cycle. However, the algorithm occasionally produces gross estimation errors, visible for example between cycles 35 and 50. A solution to improve the robustness of the estimation is to filter out the errors at the application level, by replacing the out-of-range values with the average of the previous cycles. The RMSD of the embedded estimator is specified in Fig. 6. The deviation is higher than the theoretical one, and has a maximum value of 10.5 cycles.

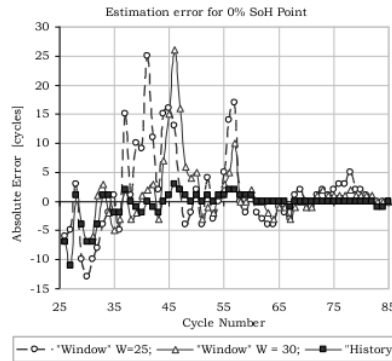


Fig. 5. GP batteries: Absolute estimation error for the embedded implementation.

We have also evaluated the estimation errors introduced by implementing the algorithm on the embedded platform, as compared to the Matlab implementation. For the first 20 estimation cycles the absolute error varies between -6 and 4 cycles. For the next estimation cycles, the theoretical and the experimental results generate similar values. Fig. 7 shows the absolute error between the Matlab and the embedded implementations.

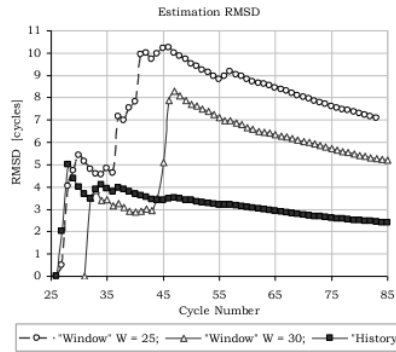


Fig. 6. Estimation RMSD for the embedded implementation.

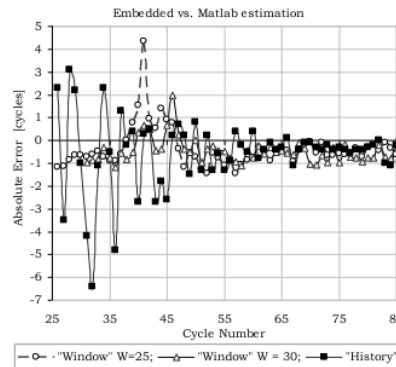


Fig. 7. Absolute error comparison between the Matlab and the embedded implementations.

Figure 8 shows the absolute estimation error of the embedded implementation for the Sanyo battery pack. Compared to Fig 5 we can observe that the initial estimation errors are smaller but the algorithm prediction accuracy is the highest maximized starting approximately from the 55-th cycle. Thus we can observe that the proposed SoH estimation approach is viable even for aged battery packs.

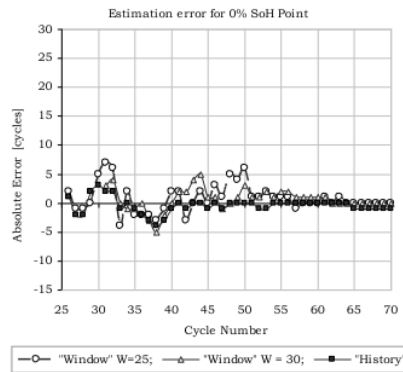


Fig 8. Sanyo batteries: Absolute estimation error for the embedded implementation

**B. Evaluation of the embedded resource requirements**

As shown in Section III, we have integrated the embedded software implementation of our proposed SoH estimation algorithms into the battery management layer. We have designed and implemented the SoH estimation calculations observing the real-time requirements imposed by the WIT architecture while minimizing the additional memory and processing power overhead.

Table I shows the memory requirements of the SoH estimation module in comparison with the total memory footprint required by the BMS software. The implementation contains the routines for both types of algorithms, in order to offer increased flexibility to the system designer. The three rows represent the comparison metrics: the total ROM (or Flash memory) used for code storage, the ROM used for constant data and the total RAM for runtime data storage. All values are expressed in bytes and the third row gives percentage values for the memory requirements of the SoH module implementation with respect to the memory usage of the entire BMS software.

TABLE I  
MEMORY REQUIREMENTS OF THE BMS AND THE SoH ESTIMATION SOFTWARE.

Memory	BMS [B]	SoH module [B]	% of BMS
Code ROM	35936	3034	8.44
Constant ROM	2056	32	1.56
Data RAM	8624	176	2.04

These comparisons clearly show that we have succeeded in implementing the SoH estimation algorithms with a low memory overhead, while also considering all the optimization aspects identified in Section III. We can also conclude that a complete BMS implementation, which provides SoH estimation, is entirely feasible on an ARM7-TDMI platform similar to the one we have used, because all such MCU platforms integrate a minimum of 64 KB Flash and 16 KB of RAM, thus becoming a very cost effective solution.

The power consumption of the BMS is another important evaluation factor. The hardware design has been done with clear low power principles in mind: using PMOS gates as switching devices for the charge current, low power and high efficiency buck boost regulators (e.g.: LTC3401 with shutdown capability) and component integration. For the current BMS implementation, the minimum available CPU frequency of 14.745 MHz is used and the BMS CPU time is 31%, yielding a total power consumption of 13.95 mW. This power envelope could be reduced if lower power alternatives are considered for future development, such as the TI MSP430 or the Energy Micro EFM32 MCUs.

Concerning the required processing power of our proposed implementation, we have done a series of tests to evaluate whether

the resulting calculation durations fit within certain real-time boundaries. For this we have varied the CPU frequency of the used MCU in steps starting from 14.745 MHz up to 58.980 MHz. We have also varied the window size of the SoH estimation algorithm, from  $W = 25$  to  $W = 40$  cycles, in order to see the evolution of the calculation times for different window sizes and to offer the system designer an area of solutions, in the CPU frequency and window size domains.

After measuring the execution durations for the time-window algorithm for different CPU frequencies, we have observed that the maximum execution duration does not exceed 550  $\mu\text{s}$  even in the worst case, i.e. at 14.745 MHz. This conclusion is important for a system designer who wishes to integrate the SoH estimation into the BMS, because it proves that the SoH algorithms add little to the real-time constraints of the system, allowing for more relaxed timing configurations.

To compare the history based and the time-window algorithms, a similar test has been conducted, emphasizing the effect of the recursive accumulation optimization discussed in sections II and III. We have considered a fixed CPU frequency of 29.490 MHz and we have varied the window size as in the previous test. The results are presented in Fig. 7 and it should be noted that, in the case of the history based algorithm implementation, the window size parameter expresses the actual number of cycles of the battery pack, while with each increment a recursive accumulation of the summation terms takes place. Thus, we have shown the history based algorithm offers a significant improvement in processing time, due to the fact that the summation terms are only calculated once (at the 25th cycle) and the subsequent calculations rely on recursive accumulation, which is more computationally efficient.

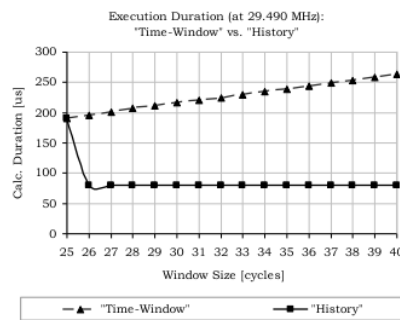


Fig. 7. Time-window vs. History-based execution durations.

## V. CONCLUSION

In this paper we have proposed two algorithms based on the least squares regression method for battery SoH estimation and we have adapted them for ease of integration into an embedded battery management solution. We have described in detail several important optimizations required for implementing the mathematical algorithms on a MCU, powered by the ARM7 architecture.

The SoH prediction algorithms have been successfully integrated into a BMS solution developed at the DSPLabs (the Digital Signal Processing Labs in Timisoara), as a central part of the CORE-TX platform.

We have conducted extensive tests in order to evaluate the accuracy and real-time feasibility of the proposed solutions. The results show the history based algorithm ensures a more accurate estimation than the time-window technique, although both types present a very good accuracy of estimation, starting from approx. 50% of the remaining useful life of the battery.

Concerning the required execution times, the recursive accumulation optimization produces a dramatic reduction in the execution time of the history based algorithm.

## REFERENCES

- [1] O. Kanoun, H. R. Trankler, "Sensor technology advances and future trends", *IEEE Trans. Instrum. Meas.*, vol. 53, no.6, pp. (1497-1501), Dec. 2004.
- [2] M. V. Micea, G. N. Carstoiu, L. Ungurean, D. Chiciudean, V. I. Cretu, V. Groza, "PARSECS: A Predictable Data Communication System for Smart Sensors and Hard Real-Time Applications", *IEEE Trans. Instrum. Meas.*, 14 pg., 2010.
- [3] R. T. Crompton, "Battery Reference Book", 3rd Edition, Reed Educational and Professional Publishing, Ltd., 2000.
- [4] V. Pop, H. J. Bergveld, P. H. L. Notten, P. P. L. Regtien, "State-of-the-art of battery state-of-charge determination", *Meas. Science Technol. J.*, vol. 16, no. 12, pp. (R93-R110), Dec. 2005.
- [5] J. C. Viera, M. Gonzalez, J. C. Anton, J. C. Campo, F. J. Ferrero, M. Valledor, "NiMH vs NiCd Batteries under High Charging Rates", in *Proc. 28th Annual Intl. Telecom. Energy Conf., INTELEC '06, USA*, pp. (1-6), Sep. 2006.



- [6] T. S. Mundra, A. Kumar, "An Innovative Battery Charger for Safe Charging of NiMH/NiCd Batteries", *IEEE Trans. Consumer Electron.*, vol.53, no.3, pp. (1044–1052), Aug. 2007.
- [7] M. Gonzalez, F. J. Ferrero, J. C. Anton, M. A. Perez, "Considerations to improve the practical design of universal and full-effective NiCd/NiMH battery fast-chargers", in *Proc. Applied Power Electron. Conf. and Exposition, APEC '99*, Texas, USA, pp. (167–173), 1999.
- [8] J. Diaz, J. A. Martin-Ramos, A. M. Pernia, F. Nuno, F. F. Linera, "Intelligent and universal fast charger for Ni-Cd and Ni-MH batteries in portable applications", *IEEE Trans. Ind. Electron.*, vol. 51, no. 4, pp. (857–863), Aug. 2004.
- [9] S. Sastry, O. Gimdogmus, T. Hartley, R. Viellette, "Coordinated Discharge of a Collection of Batteries", *Journal of Power Sources*, no. 166, pp. (284-296), 2007.
- [10] A. B. da Cunha, B. R. de Almeida, D. C. da Silva, Jr., "Remaining Capacity Measurement and Analysis of Alkaline Batteries for Wireless Sensor Nodes", *IEEE Trans. Instrum. Meas.*, vol. 58, no. 6, pp. (1816–1822), Jun. 2009.
- [11] J. Alvarez, J. Marcos, A. Lago, A. A. Nogueiras, J. Doval, C. M. Penalver, "A fully digital smart and fast lead-acid battery charge system", in *Proc. 34th Annual IEEE Power Electron. Specialist Conf., PESC '03*, vol.2, Acapulco, Mexico, pp. (913–917), Jun. 2003.
- [12] F. Codeca, S. M. Savaresi, G. Rizzoni, "On battery State of Charge estimation: A new mixed algorithm", in *Proc. IEEE Intl. Conf. Contr. Apps., CCA 2008*, pp. (102–107), Sep. 2008.
- [13] F. Codeca, S. M. Savaresi, V. Manzoni, "The mix estimation algorithm for battery State-of-Charge estimator - Analysis of the sensitivity to measurement errors", in *Proc. 48th IEEE Conf. Decision and Contr., CDC/CCC 2009*, Shanghai, China, pp. (8083–8088), Dec. 2009.
- [14] T.S. Mundra, A. Kumar, "Micro power battery state-of-charge monitor", *IEEE Trans. Consumer Electron.*, vol. 54, no. 2, pp. (623–630), May 2008.
- [15] B. Saha, K. Goebel, S. Poll, J. Christophersen, "Prognostics Methods for Battery Health Monitoring Using a Bayesian Framework", *IEEE Trans. Instrum. Meas.*, vol. 58, no. 2, pp. (291–296), Feb. 2009.
- [16] A. Saxena, J. Celaya, E. Balaban, K. Goebel, B. Saha, S. Saha, M. Schwabacher, "Metrics for evaluating performance of prognostic techniques", in *Proc. Intl. Conf. Prognostics and Health Manage., PHM 2008*, Colorado, USA, pp. (1–17), Oct. 2008.
- [17] H. J. Bergveld, W. S. Kruijt, P. H. L. Notten, "Battery management systems: Design by modelling", *Kluwer Academic Publishers*, 2002.
- [18] R. H. Somogye, "An aging model of Ni-MH batteries for use in hybrid-electric vehicles", MSc. Thesis, *Department of Electrical Engineering*, Ohio State University, USA, 2004.
- [19] R. D. Cioarga, M. V. Micea, B. Ciubotaru, D. Chiucudean, D. Stanescu, "CORE-TX: Collective Robotic Environment - the Timisoara Experiment", in *Proc. 3rd Romanian-Hungarian Joint Symp. Applied Computational Intellig., SACI 2006*, Timisoara, Romania, pp. (495–506), May 2006.
- [20] A. Pianegiani Boni, D. F. Petri, "Low-Power and Low-Cost Implementation of SVMs for Smart Sensors", *IEEE Trans. Instrum. Meas.*, vol. 56, no. 1, pp. (39–44), Feb. 2007.
- [21] Texas Instruments, "TMP101: Digital Temperature Sensor with I2C™ Interface", Datasheet Rev. G, Texas Instruments, 2007.
- [22] Philips Semiconductors, "LPC2131/2/4/6/8 User manual", Rev. 02, Phillips Semiconductors, Jul. 2006.
- [23] D. Macii, D. Petri, "Accurate Software-Related Average Current Drain Measurements in Embedded Systems", *IEEE Trans. Instrum. Meas.*, vol. 56, no. 3, pp. (723–730), Jun. 2007.
- [24] M. V. Micea, V. Cretu, V. Groza, "Maximum Predictability in Signal Interactions with HARETICK Kernel", *IEEE Trans. Instrum. Meas.*, vol. 55, no. 4, pp. (1317–1330), Aug. 2006.
- [25] S. T. Karris, "Numerical Analysis Using MATLAB and Excel", 3rd Edition, *Orchard Publications*, 2007.



