NNT : 2015SACLS148

THESE DE DOCTORAT
DE L'UNIVERSITE PARIS-SACLAY
préparée à Université Paris-Sud

ÉCOLE DOCTORALE N° 574
ÉCOLE DOCTORALE DE MATHÉMATIQUES HADAMARD

Spécialité : Mathématiques Appliquées

Par

Mme Jana KALAWOUN

MODÉLISATION STATISTIQUE DE L'ÉTAT DE CHARGE DES BATTERIES ÉLECTRIQUES

Thèse présentée et soutenue à Orsay, le 30 novembre 2015 :

Composition du Jury :

Mme. GASSIAT Elisabeth	Professeur, Université Paris-Sud	Présidente
M. CAPPÉ Olivier	Directeur de recherche, Télécom ParisTech	Rapporteur
M. LE GLAND François	Directeur de recherche, INRIA	Rapporteur
M. MENSLER Michel	Ingénieur R&D, Renault	Examinateur
M. CELEUX Gilles	Directeur de recherche, INRIA	Directeur de thèse
M. MONTARU Maxime	Ingénieur R&D, CEA	Co-encadrant de thèse
M. PAMPHILE Patrick	Maître de conférence, Université Paris-Sud	Co-encadrant de thèse

Remerciements

Ce manuscrit est le résultat de trois années de travaux de recherche, effectuées au sein du CEA Saclay, de l'INES Chambéry et l'Université Paris-Sud.

Durant ma thèse, j'ai été aidée, accompagnée, encadrée et encouragée par toute une équipe de chercheurs, scientifiques et experts. Qu'ils en soient ici remerciés.

M. Gilles CELEUX, pour son appui scientifique et son aide dans le développement de mon esprit de recherche.

MM. Olivier CAPPE et François LE GLAND, qui m'ont fait l'honneur et le plaisir d'être rapporteurs sur ma thèse, pour leurs remarques et pertinentes propositions.

Mme Elisabeth GASSIAT et M. Michel MENSLER, pour avoir accepté de faire partie de mon jury.

M. Patrick PAMPHILE, pour son accompagnement, ses précieux conseils tout au long de ma thèse, et sa grande disponibilité.

M. Maxime MONTARU et l'équipe de l'INES, pour leur accueil et leur apport scientifique.

Mme Krystyna BILETSKA, M. Yohan PETETIN et l'équipe du CEA Saclay – LADIS, pour leur accueil et la qualité de l'environnement de recherche dont j'ai bénéficié.

Mes amis, qui ont apporté de la légèreté à ces trois années de travail, et Lucile qui fut ma compagne de route à quelques jours près.

À Tamim

Contents

Le manuscrit est rédigé en anglais. Toutefois, chaque chapitre est doté d'un résumé en français. L'introduction et la conclusion sont également rédigées en français.

The manuscript is written in English. However, each chapter has a French summary. The introduction and the conclusion are written in French.

Abbreviations

A	**A**mpere
Ah	**A**mpere-**h**our
AIC	**A**kaike **I**nformation **C**riterion
ANN	**A**rtificial **N**eural **N**etwork
ARMA	**A**uto**R**egressive **M**oving **A**verage
BIC	**B**ayesian **I**nformation **C**riterion
BMS	**B**attery **M**anagment **S**ystem
CVL	**C**ross-**V**alidation **L**ikelihood
EM	**E**xpectation **M**aximization
ESS	**E**ffective **S**ample **S**ize
HMM	**H**idden **M**arkov **M**odel
iid	**i**ndependent and **i**dentically **d**istributed
IR	**I**nternal **R**esistance
KLD	**K**ullback-**L**eibler **D**ivergence
LGSSM	**L**inear **G**aussian **S**tate **S**pace **M**odel
MAP	**M**aximum ***a Posteriori***
MC	**M**onte **C**arlo
MLE	**M**aximum **L**ikelihood **E**stimate
MSE	**M**ean **S**quare **E**rror
NEDC	**N**ew **E**uropean **D**riving **C**ycle
OCV	**O**pen **C**ircuit **V**oltage
OEHS	**O**dd **E**ven **H**alf **S**ampling
OSL	**O**ne **S**tep **L**ate
pdf	**p**robability **d**ensity **f**unction
SMSSM	**S**witching **M**arkov **S**tate **S**pace **M**odel

SSM	**S**tate **S**pace **M**odel
SoC	**S**tate **o**f **C**harge
SoH	**S**tate **o**f **H**ealth
SVM	**S**upport **V**ector **M**achine
V	**V**olts
W	**W**att

Mathematical notations

$\mathcal{N}(z; m, \sigma^2)$	Gaussian pdf with mean m and variance σ^2 taken at point z
$Z \sim \mathcal{N}(m, \sigma^2)$	Z follows a Gaussian distribution with mean m and variance σ^2
$\{Z_t\}_{t\geq 0}$ or $(Z_t)_{t\geq 0}$	Stochastic process
$z_{0:T}$	$\{z_0, z_1, \ldots, z_T\}$
$\mathbb{E}_Y[f(X)]$	Conditional expectation: $\mathbb{E}[f(X)\|Y]$
$\{1, \ldots, \kappa\}^T$	T-ary product: it is a set of T-tuples with cardinality equal to κ^T
X_t	Random continuous state of interest of a LGSSM/SMSSM
S_t	Random discrete state of a SMSSM
Y_t	Known observation of a LGSSM/SMSSM
u_t	Input of a SMSSM
Γ	Set of unknown parameters of a LGSSM
Θ	Set of unknown parameters of a SMSSM
Π	Vector of initial probability of the Markov chain $\{S_t\}_{t\geq 0}$
P	Transition matrix of the Markov chain $\{S_t\}_{t\geq 0}$
κ	Number of discrete states of the Markov chain

In this study, we use the standard convention whereby capital letters denote random variables, and lower letters are used for their corresponding realizations.

Electrochemical definitions

Battery Basics

- **Principle of operation of an electric battery** An electric battery converts stored chemical energy into electrical energy. This conversion is reversible and can be performed multiple times for *secondary batteries*, whereas *primary batteries* cannot be reliably recharged since their chemical reactions are not easily reversible. In this study, we are interested in secondary batteries. For simplicity sake, hereafter the term "battery" designates a "secondary battery". A battery generally consists of three primary parts: two terminals, called *electrodes*, electrically connected through a substance, called *electrolyte*, allowing ions to move between the electrodes. The operation of a battery is based on two types of chemical reactions, namely *reduction* and *oxidation*, commonly referred to as *redox* reactions. During the battery charge, an oxidation reaction occurs at the positive electrode, as the active material releases electrons. On the other side, a reduction reaction occurs at the negative electrode, as the active material gains electrons. During the battery discharge, electrodes switch roles: the positive electrode is reduced, and the negative one is oxidized. The released and gained electrons generate current in the external circuit, which flows from the negative to the positive electrode during charge, and in the opposite direction during discharge (cf. Figure 1).

- **Cell, module, and pack** - *Cellule, module et pack* - A cell is a single unit performing the functions of the battery. A module is formed by connecting several cells in series and/or in parallel. A pack is formed by connecting several modules in series and/or in parallel.

- **Amount of charge**, expressed in Ampere-hour (Ah) - *Quantité de charge* - The integrated current that can be removed from the battery in a time period:

$$Q = \int_{t_1}^{t_2} I(t)\,\mathrm{d}t.$$

- **Charge/Discharge current rate** nC - *Régime de courant de charge/décharge* - An expression of the speed according to which a battery is charged/discharged.

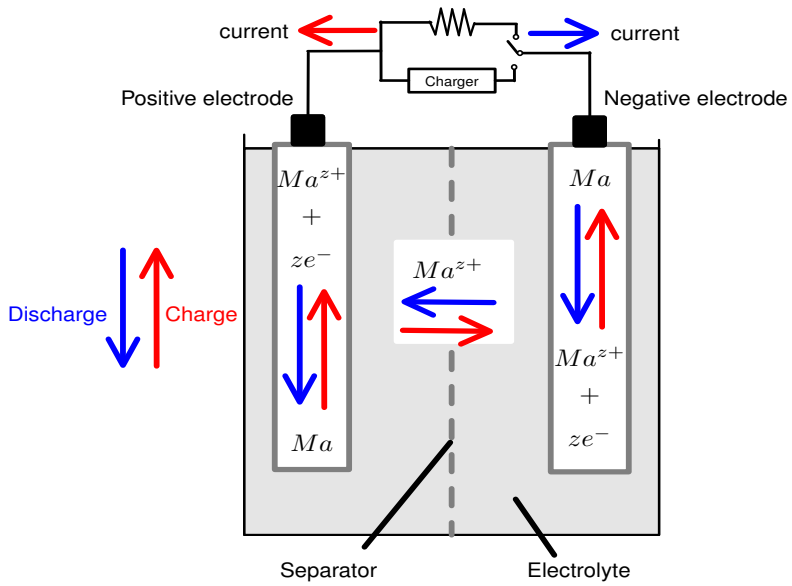

FIGURE 1: Illustration of battery components and operation: Ma designates an active material, Ma^{z+} an ion with positif charge (cation), and e^- an electron

The charge/discharge time is inversely proportional to the current value. Let us consider a battery with nominal capacity equal to $15Ah$. A $1C$ rate means that the battery can provide a current of $15A$ for $1h$. However, a $C/2$ rate means that the battery can provide, theoretically, a current of $7.5A$ for $2h$.

- **Charge/Discharge power rate** nE - *Régime de puissance de charge/décharge* - Similarly to a current rate, a power rate describes the discharge power. A $1E$ rate is the discharge power to completely discharge a fully charged battery in $1h$.

Battery Technical Specifications

This section provides definitions of some specifications found in battery technical sheets.

- **Nominal capacity**, expressed in Ah - *Capacité nominale* - The maximum amount of charge that a new and fully charged battery can provide under a nominal discharge regime.
- **Nominal charge/discharge regime** - *Régime de charge/décharge nominale* - The set of charge/discharge conditions specified by the manufacturer for which the nominal capacity of a new battery is equal to its actual capacity: ambient temperature (generally equal to $25°C$) and current rate (generally equal to $C/2$).
- **Nominal voltage**, expressed in Volts (V) - *Tension nominale* - For a constant discharge current, it is the voltage reported on the largest part of the discharge curve (generally when the state of charge belongs to $[20\%, 80\%]$).

- **Float voltage**, expressed in V - *Tension flottante* - The maximum allowed voltage at which the battery is maintained after a complete charge ($SoC = 100\%$) until the current becomes close to zero, thus guaranteeing efficient charge.
- **Cut-off voltage**, expressed in V - *Tension minimale* - The minimum allowed voltage for which the battery is considered as "empty".
- **Cycle** - *Cycle* - A complete discharge of a fully charged battery followed by a complete charge.
- **Life cycle** - *Cycle de vie* - The number of cycles that the battery can perform while meeting specific performance criteria. It is determined for specific charge/discharge conditions.
- **Impedance** - *Impedance* - The opposition exhibited by a battery to an alternative current (AC) of a specific frequency. It consists of resistance (expressed in Ω), inductance (expressed in Henry, H), and capacitance (expressed in Faraday, F). It corresponds to the response of the battery to an excitation of a small amplitude.
- **Resistance**, expressed in Ω - *Résistance* - The opposition exhibited by a battery to a direct current (DC).

Battery Condition

This section provides definitions of some variables used to describe the actual conditions of a battery.

- **Remaining capacity**, expressed in Ah - *Capacité disponible* - The amount of charge that a battery can provide under specific discharge conditions before becoming "empty".
- **Actual capacity**, expressed in Ah - *Capacité réelle* - The maximum amount of charge that a fully charged battery can provide under specific discharge conditions. It depends, amongst other factors, on the usage history of the battery and the discharge conditions.
- **State of Charge** (SoC), expressed in % - *État de charge* - The ratio between remaining and actual capacities.
- **State of Health** (SoH), expressed in % - *État de santé* - The ratio between the maximum amount of charge that a fully charged battery can provide under its nominal discharge regime, and its nominal capacity.
- **Open Circuit Voltage** (OCV), expressed in V - *Tension à vide* - The battery voltage when no load is applied ($I = 0A$). An accurate measurement requires a long rest period ($\simeq 30$ min).

- **Coulombic efficiency** - *Rendement faradique* - The ratio of the amount of charge that exits the battery during the discharging cycle over the one entering it during the charging cycle. The losses that reduce Coulombic efficiency are primarily due to secondary reactions. It depends, amongst other factors, on the ambient temperature, the current rate, and the state of health of the battery.

- **Active power**, expressed in Watt W - *Puissance active* - The active power averaged over a period of time is given by

$$P_{\text{active}} = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} V_t \cdot I_t \, dt,$$

  where V_t is the battery voltage and I_t is the current that flows into the battery.

- **Self-discharge** - *Auto-décharge* - During the rest period, secondary reactions lead to current leakages and therefore reduce the remaining capacity. The self-discharge depends primarily on the ambient temperature, the rest period, and the state of charge and health of the battery.

- **Battery autonomy** - *Autonomie d'une batterie* - The time for which the battery will support a load before becoming "empty". It depends, amongst other factors, on the load, the SoC and SoH of the battery, and the ambient temperature.

Chapitre 1

Introduction

1.1 Contexte et problématique

Les batteries électriques jouent un rôle important dans la transition énergétique que nous vivons actuellement. Elles sont utilisées dans des applications stationnaires comme le stockage de l'énergie solaire pour lutter contre la raréfaction des énergies fossiles, ainsi que dans des applications temps réel comme les véhicules électriques pour lutter contre la pollution.

Une indication précise de l'état de charge d'une batterie (*State of Charge – SoC*) permet de garantir une utilisation sûre de la batterie en évitant une surcharge ou une surdécharge pouvant l'endommager. Cette indication est également indispensable pour calculer l'énergie disponible de la batterie ainsi que son autonomie. Cela est particulièrement important pour des applications cruciales comme les véhicules électriques.

Cependant, la batterie étant un système électrochimique complexe, ni son SoC ni sa capacité disponible ne peuvent être mesurés directement par un capteur. De plus, la dynamique de la batterie dépend non seulement de ses caractéristiques internes, comme sa composition chimique et son état de santé, mais aussi de ses conditions d'usage, souvent non contrôlables, comme la température ambiante et le profil de courant (qui caractérise le type de conduite, la dénivellation de la route, etc.) (cf. Figure 1.1). Cela rend difficile la conception d'un estimateur fiable du SoC, notamment pour des applications temps réel. En effet, deux principales contraintes limitent les performances d'un estimateur temps réel du SoC : la puissance de calcul limitée du système de contrôle de la batterie (*Battery Management System - BMS*) et le changement aléatoire des conditions d'usage externes de la batterie.

De nombreuses études ont cherché à obtenir une estimation fiable du SoC. Nous les classifions en quatre catégories :

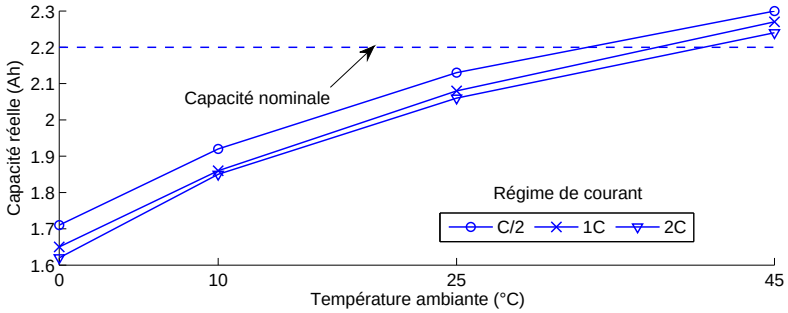

FIGURE 1.1: Variation de la capacité réelle d'une batterie électrique en fonction de la température ambiante et la vitesse de décharge

1. Les méthodes basées sur une relation directe entre le SoC et une quantité physique telle que la tension à vide ou l'impédance interne de la batterie. Cette relation est souvent décrite par une table de correspondance ou un modèle de régression comme dans Huet [1998] et Peled et al. [1988]. Ce type de méthodes est simple à implémenter et fournit une indication précise du SoC. Cependant, l'utilisation en ligne de ces méthodes est très limitée. En effet, les quantités physiques utilisées ici en entrée ne peuvent pas être mesurées en ligne.

2. Les méthodes basées sur la modélisation électrochimique de la dynamique de la batterie permettant, par exemple, de modéliser la diffusion des particules entre les deux électrodes de la batterie comme dans Di Domenico et al. [2008] et Pop et al. [2006]. Ces méthodes permettent de suivre de près la dynamique de la batterie. Cependant, le calibrage du modèle nécessite un accès à des paramètres de bas niveau (par exemple la constante de diffusion temporelle des ions dans l'électrolyte), ce qui peut être compliqué, notamment dans un pack de batteries.

3. Les méthodes basées sur la modélisation physique de la dynamique de la batterie, dont l'une des plus utilisées est la méthode coulométrique, dite aussi de comptage d'ampère-heure (*Ampere-hour counting - Ah-counting*) (Ng et al. [2009], Wang et al. [2007]). Celle-ci consiste à faire une somme pondérée du courant entrant et sortant de la batterie. Toutefois, cette méthode présente de nombreuses limites : le cumul de l'erreur du capteur de courant, la variation de la capacité de la batterie, etc. D'autres méthodes physiques sont basées sur la modélisation de la batterie par un circuit électrique équivalent. Leur principe est soit d'estimer la tension à vide et par la suite utiliser des méthodes de la première catégorie pour calculer le SoC (Hirai et al. [2008], Moo et al. [2007], Pang et al. [2001]), soit d'estimer la tension de la batterie et de la comparer avec la tension mesurée afin d'ajuster le SoC coulométrique par un système à boucle fermée, comme le contrôleur ou le filtre de Kalman (Codeca et al. [2008], Plett [2004]). Ces méthodes tiennent compte de l'erreur des capteurs et sont simples à intégrer dans le BMS afin d'indiquer le SoC en temps réel.

4. Les méthodes basées sur l'apprentissage statistique d'un modèle de régression, comme les réseaux de neurones (*Artificial Neural Network - ANN*), les machines à vecteurs de support (*Support Vector Machine - SVM*) et les modèles autorégressifs et moyenne mobile (*Autoregressive moving average models - ARMA*) (Affanni et al. [2003], Bo et al. [2008], Hansen and Wang [2005], Shen [2010]). Le principe de ces méthodes est d'estimer les paramètres d'un modèle de régression prédéfini à partir d'une base d'apprentissage formée de données collectées lors des tests de charge et de décharge d'une batterie. Ces méthodes donnent une indication précise du SoC lorsque les conditions d'usage et les caractéristiques internes de la batterie sont proches de celles de la base d'apprentissage. En revanche, certaines de ces méthodes, les SVMs par exemple, ne tiennent pas compte de l'évolution temporelle du SoC.

1.2 Motivations

Ces différentes méthodes décrites ne tiennent pas compte de la variabilité de la dynamique de la batterie dépendante de ses caractéristiques internes et de ses conditions d'usage souvent non contrôlables. Des améliorations ont été introduites afin de permettre aux modèles de SoC de s'adapter à cette variabilité. En effet, les paramètres d'un modèle de SoC peuvent être identifiés pour plusieurs températures ambiantes et SoCs comme dans Hirai et al. [2008]. Ils peuvent également être intégrés dans le vecteur d'état du filtre de Kalman comme dans Pang et al. [2001]. La variabilité peut être aussi représentée par un modèle de régression entre les paramètres et la température ambiante comme dans Plett [2004]. Néanmoins, ces solutions ne tiennent pas compte des limitations de ressources disponibles dans des applications temps réel et/ou ne peuvent pas être généralisées pour des configurations inconnues de la dynamique de la batterie lors de la phase d'apprentissage du modèle.

1.3 Objectifs de la thèse

Dans cette thèse, nous cherchons à réaliser une modélisation statistique du SoC à partir des mesures instantanées de courant I_t et de tension U_t.

L'objectif est d'obtenir un modèle statistique générique tenant compte des erreurs de capteurs de courant et de tension ainsi que de la variabilité de la dynamique de la batterie selon ses conditions d'usage et ses caractéristiques internes. Il est également important que ce modèle soit adapté à des applications temps réel et considère la capacité de calcul limitée du BMS. Le SoC étant inconnu, nous utiliserons des algorithmes d'apprentissage adaptés aux modèles à structure latente.

Enfin, dans le but d'identifier les points forts et les limitations du modèle développé, nous allons l'appliquer sur des données issues de tests de charge/décharge de plusieurs types de batteries et sous différents conditions d'usage soit contrôlables (tests dans le laboratoire), soit non contrôlables (roulages réels d'un véhicule électrique).

1.4 Démarche

La démarche consiste, dans un premier temps, à analyser les méthodes existantes d'estimation du *SoC*, à identifier leurs avantages et limitations et à évaluer leurs performances pour des applications temps réel. Dans ce cadre, nous réalisons une nouvelle classification de ces méthodes basée sur les types des variables d'entrée (estimées et mesurées), de modèle du *SoC* (physique, chimique et statistique) et de traitement des données (non récursif, récursif à boucle ouverte et récursif à boucle fermée).

Dans un deuxième temps, nous proposons une modélisation statistique du *SoC* à partir de mesures instantanées de courant et de tension. Cette modélisation est basée sur un modèle à espaces d'états gouverné par une chaîne de Markov cachée (*Switching Markov State-Space Model - SMSSM*). En effet, nous considérons que la variation aléatoire de la dynamique de la batterie et l'évolution du *SoC* peuvent être modélisées par plusieurs modèles à espaces d'états linéaires et gaussiens. Cela nous permet de modéliser les différents "régimes" de fonctionnement de la batterie.

Définition 1

Par un régime de fonctionnement d'une batterie, nous entendons une relation linéaire entre la tension, le courant et l'état de charge. Étant donné la nature non-linéaire d'une relation entre ces trois grandeurs physiques, cette notion de "régime" n'a pas nécessairement une signification électrochimique.

Dans le cas des SMSSMs, la commutation entre les régimes est indexée par une chaîne de Markov. À chaque instant t, trois variables aléatoires sont ainsi considérées par cette modélisation :

1. l'état continu inconnu X_t, décrivant le *SoC* à estimer,
2. l'observation Y_t, décrivant la tension mesurée de la batterie,
3. l'état discret caché d'une chaîne de Markov S_t, indexant le régime de fonctionnement de la batterie.

Nous considérons le courant traversant la batterie comme une entrée connue, notée u_t.

Un SMSSM décrit l'évolution jointe de ces variables par deux équations : la première, dite *équation de transition*, décrit l'évolution temporelle de X_t, et la deuxième, dite

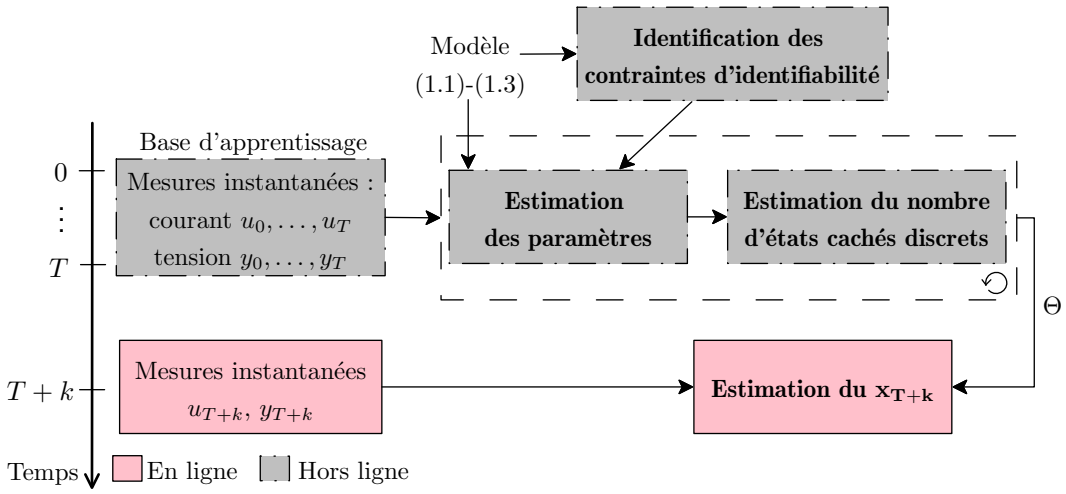

FIGURE 1.2: Phases de conception d'un modèle de *SoC* basé sur les SMSSMs

équation d'observation, décrit l'influence de l'état inconnu X_t sur les observations Y_t. Dans cette étude, ces deux équations sont obtenues à partir de modèles physiques. La forme générale du modèle est décrite de la manière suivante :

$$(S_t)_{t\geq 0} \quad \text{chaîne de Markov, avec } p_{\Theta_1}(s_t \,|s_{0:t-1}) = p_{\Theta_1}(s_t \,|s_{t-1}) \tag{1.1}$$

$$X_t = f_{\Theta_2}(S_t, X_{t-1}, u_t) \tag{1.2}$$

$$Y_t = g_{\Theta_3}(S_t, X_t, u_t), \tag{1.3}$$

où $t = 0, 1, \ldots, T$, $\Theta = \{\Theta_1, \Theta_2, \Theta_3\}$ est l'ensemble des paramètres du modèle et f et g sont des fonctions linéaires.

L'utilisation d'un SMSSM pour estimer le *SoC* induit quatre problématiques essentielles : la vérification de l'identifiabilité du modèle, l'estimation hors ligne de l'ensemble de paramètres Θ, l'estimation du nombre adéquat d'états cachés de la chaîne de Markov et enfin l'estimation en ligne du *SoC*. La Figure 1.2 présente un schéma résumant ces quatres problématiques.

Identifiabilité - Un modèle est dit identifiable lorsque la relation entre l'ensemble des paramètres et la vraisemblance est bijective. Un SMSSM n'est pas nécessairement identifiable et ses paramètres ne peuvent pas être ainsi estimés d'une façon unique (Walter and Lecourtier [1981]).

Dans cette étude, nous démontrons que *l'identifiabilité générale* de l'ensemble des paramètres du SMSSM peut être assurée en imposant des contraintes simples et naturelles sur quelques paramètres du modèle.

Dans un premier temps, nous supposons que l'ensemble de paramètres Θ est connu et nous cherchons à estimer en ligne X_t.

Estimation en ligne du SoC - Le jeu de paramètres estimés Θ est stocké dans le BMS et utilisé afin de donner une indication du *SoC* au cours du fonctionnement de la batterie. Nous cherchons ainsi à estimer x_t de façon récursive à partir d'une estimation de x_{t-1}, de la nouvelle observation y_t et de la nouvelle entrée u_t. Cependant, une estimation optimale de X_t, au sens de l'erreur quadratique moyenne, est un problème NP-complet connu (Tugnait [1982]).

Dans cette étude, nous utilisons une méthode de Monte-Carlo pour estimer X_t. Une étude est réalisée pour trouver le nombre de particules à utiliser pour assurer à la fois une estimation précise du *SoC* et tenir compte des limitations des ressources d'un système embarqué.

Estimation hors ligne des paramètres du modèle - Nous considérons une base de données d'apprentissage formée d'un jeu d'observations $y_{0:T} = \{y_0, \dots, y_T\}$ et d'entrées $u_{0:T}$. Nous cherchons à estimer l'ensemble de paramètres du modèle Θ à partir de cette base d'apprentissage par une approche bayésienne et une approche du type maximum de vraisemblance.

L'approche bayésienne considère que ces paramètres sont des variables aléatoires et cherche à minimiser l'erreur quadratique moyenne entre les vrais paramètres et ceux estimés. Nous considérons ainsi une connaissance *a priori* du jeu de paramètres du modèle Θ. Cette approche est motivée par le fait que le modèle du *SoC* proposé est basé sur des équations physiques. De ce fait, les paramètres sont physiquement interprétables et nous disposons d'informations *a priori*. Une solution analytique étant impossible, nous avons recours à l'échantillonneur de Gibbs spécialement utilisé dans le cadre d'une inférence bayésienne. Ici, la k-ème itération de l'échantillonneur de Gibbs consiste à simuler successivement un jeu de paramètres $\Theta(k)$, une séquence $s_{0:T}(k)$ des états discrets et une séquence $x_{0:T}(k)$ des états continus par les lois *a posteriori* $p(\Theta|y_{0:T}, s_{0:T}(k-1), x_{0:T}(k-1))$, $p_{\Theta(k)}(s_{0:T}|y_{0:T}, x_{0:T}(k-1))$ et $p_{\Theta(k)}(x_{0:T}|y_{0:T}, s_{0:T}(k))$. Pour assurer une simulation exacte de Θ, nous choisissons la loi *a priori* $p(\Theta)$ de façon à ce que la loi *a posteriori* $p(\Theta|y_{0:T}, s_{0:T}, x_{0:T})$ soit une loi conjuguée de $p(\Theta)$. Plusieurs algorithmes ont été développés pour tirer efficacement de $p_{\Theta(k)}(s_{0:T}|y_{0:T}, x_{0:T}(k-1))$ (Carter and Kohn [1994], Chib [1996]) et $p_{\Theta(k)}(x_{0:T}|y_{0:T}, s_{0:T}(k))$ (Carter and Kohn [1994], De Jong and Shephard [1995], Durbin and Koopman [2002], Frühwirth-Schnatter [1994]). Toutefois, les deux séquences latentes $s_{0:T}$ et $x_{0:T}$ sont fortement corrélées, ce qui rend l'exploration de l'espace de paramètres lente. Une solution consiste à tirer la séquence d'états discrets en ne considérant que les observations $y_{0:T}$ (Carter and Kohn [1996]). Cependant, un tirage selon la loi initiale $p_\Theta(s_{0:T}|y_{0:T})$ étant difficile, nous utilisons une technique d'échantillonnage d'importance pour l'estimer. Cette version modifiée d'échantillonneur de Gibbs est appelée un *échantillonneur de Gibbs particulaire* (Andrieu et al. [2010]). Afin de calibrer cette méthode, nous réalisons une étude de sensibilité basée

sur des données simulées. Plus précisément, l'impact du choix de la loi *a priori* et du nombre de particules est étudié.

La deuxième approche est basée sur la maximisation de la vraisemblance. Une solution analytique de l'estimateur de maximum de vraisemblance étant impossible, nous utilisons l'algorithme EM (Dempster et al. [1977]) particulièrement adapté pour des modèles à structure latente : Rabiner [1989] pour les modèles de Markov cachés (*Hidden Markov Model - HMM*) et Shumway and Stoffer [1982] pour les modèles à espaces d'états (*State Space Model - SSM*). Cependant, l'algorithme EM repose sur le calcul des espérances conditionnelles $\mathbb{E}_{Y_{0:T},\Theta}[X_{0:T}]$ et $\mathbb{E}_{Y_{0:T},\Theta}[S_{0:T}]$, qui est un problème NP complet connu dans le cas des SMSSMs (Tugnait [1982]). Plus précisément, pour un modèle à κ états discrets cachés et un jeu d'observations $y_{0:T}$ de taille $T+1$, l'étape E (espérance) de l'algorithme EM nécessite une somme sur les κ^{T+1} séquences possibles de $s_{0:T}$. Afin de surmonter cette difficulté, nous utilisons une version stochastique de l'algorithme EM, nommée *Monte-Carlo EM* (MCEM). Ainsi, la somme sur les κ^{T+1} séquences est approchée par une méthode de Monte-Carlo à partir d'un ensemble de séquences simulées. De plus, les méthodes du type maximum de vraisemblance peuvent se révéler inefficace lorsque le modèle n'est pas identifiable comme dans le cas des SMSSMs. Pour surmonter cette problématique, nous développons deux variantes pénalisées du MCEM (Tanner [1996], Wei and Tanner [1990]). La première est pénalisée par les contraintes d'identifiabilité trouvées. Quant à la deuxième, elle est pénalisée par une loi *a priori* sur les paramètres du modèle (Green [1990]). Afin de calibrer ces deux algorithmes pénalisés, nous réalisons une étude détaillée de leur sensibilité en se basant sur des données simulées. Plus précisément, l'impact de la stratégie d'initialisation, du nombre de particules et du choix de la loi *a priori* est étudié.

Estimation du nombre adéquat d'états cachés discrets - Les états cachés de la chaîne de Markov visent à modéliser les différents régimes de la dynamique de la batterie. Cependant, aucune information sur le nombre de ces régimes n'est disponible et le nombre adéquat d'états cachés doit être correctement estimé afin d'avoir une estimation précise du *SoC*. Dans cette étude, différents critères de sélection du modèle, initialement basés sur la vraisemblance pénalisée, sont testés : les critères d'information d'Akaike (*Akaike information criterion - AIC*, Akaike [1998]) et de Bayes (*Bayesian information criterion - BIC*, Schwarz [1978]), l'heuristique de pente (*slope heuristics criterion - SHC*, Birgé and Massart [2007]) et la vraisemblance croisée (*cross-validated Likelihood- CVL*, Celeux and Durand [2008]).

Application du modèle sur des données réelles - Afin d'évaluer la robustesse du modèle de *SoC* développé, nous utilisons des données collectées lors d'expérimentations de charge/décharge de batteries aux caractéristiques internes variées et sous différentes conditions d'usage. À cette fin, nous considérons trois types de batteries : une cellule

rechargeable lithium-ion de type S, un module de batteries appelé “M60” et composé de 60 cellules rechargeables lithium-ion de type L et un pack de batteries composé de 10 modules “M60” équipant un véhicule électrique.

Pour chaque type de batteries, nous montrons des résultats d'expérimentations de charge/décharge permettant de caractériser la variabilité du comportement de la batterie en fonction de ses conditions d'usage externes et de ses caractéristiques internes. L'influence du régime de courant, de la température ambiante et du *SoC* sur la capacité réelle, le rendement faradique, l'OCV et la résistance interne est particulièrement étudiée.

Ensuite, nous appliquons le modèle sur des données collectées lors d'expérimentations de décharge en laboratoire sous des températures ambiantes variées et différents profils de courant simulant l'usage réel d'un véhicule électrique. Afin d'identifier les avantages et les limitations du modèle de *SoC* proposé, nous considérons toutes les combinaisons possibles d'apprentissage/validation. Dans le cas du pack de batteries, nous validons le modèle de *SoC* en se basant sur des données issues de roulages réels d'un véhicule électrique.

1.5 Communiqués scientifiques

Ce travail a fait l'objet de quatres publications :

- **Articles de revues internationales avec comité de lecture**
  (1) J. Kalawoun, K. Biletska, F. Suard and M. Montaru, *From a novel classification of the battery state of charge estimators toward a conception of an ideal one*, Journal of Power Sources, volume 279, pages 694-706, April 2015.
  (2) J. Kalawoun and P. Pamphile, *How to deal with non-linearity using a Switching Markov State-Space model ?*, Computational Statistics and Data Analysis (submitted).

- **Article de conférence internationale avec actes et comité de lecture**
  (3) J. Kalawoun, P. Pamphile, G. Celeux, K. Biletska and M. Montaru, *Estimation of the state of charge of an electric battery : Switching Markov State-Space Model*, European Signal Processing Conference'15, Aug 2015, Nice, France, pages 1995-1999.

- **Article de conférence nationale avec actes et comité de lecture**
  (4) J. Kalawoun, P. Pamphile and G. Celeux, *Identifiability of a Switching Markov State-Space model*, GRETSI'15, Sep 2015, Lyon, France, pp.4.

1.6 Structure du manuscrit

Le manuscrit est organisé de la façon suivante. Le Chapitre 2 passe en revue les méthodes d'estimation du *SoC* et présente une nouvelle classification de celles-ci.
Dans le Chapitre 3, nous développons un nouveau modèle de *SoC* basé sur les SMSSMs, et nous identifions les contraintes assurant l'identifiabilité de ses paramètres. Pour un jeu de paramètres fixés, nous présentons une méthode de Monte-Carlo pour estimer en ligne x_t à partir d'une estimation de X_{t-1}, une nouvelle observation y_t et une nouvelle entrée u_t. L'algorithme développé est calibré à partir des données simulées.
Les Chapitres 4 et 5 sont consacrés à l'estimation des paramètres du modèle respectivement par une approche bayésienne et du type de maximum de vraisemblance. Les algorithmes développés sont calibrés à partir des données simulées.
Dans le Chapitre 6, nous présentons les résultats théoriques de quatres critères de sélection de modèle, soit AIC, BIC, SHC et CVL. En se basant sur des données simulées, nous testons ensuite leur capacité à estimer le vrai nombre d'états cachés discrets.
Dans le Chapitre 7, le modèle de *SoC* est mis en oeuvre pour des données réelles de charge/décharge de différents types de batteries (cellules, modules et packs) : estimation des paramètres inconnus et du nombre d'états discrets cachés. La sensibilité du choix de la base d'apprentissage sur les performances du modèle est particulièrement étudiée.
Le Chapitre 8 de conclusion et perspectives évoque en particulier les limitations du modèle et la possibilité d'adapter automatiquement ses paramètres au cours du fonctionnement de la batterie à travers des techniques d'apprentissage en ligne.

Chapter 2

Overview and classification of the methods of estimation of the state of charge of an electric battery*

Ce chapitre passe en revue les méthodes d'estimation de *SoC* proposées dans la littérature et discute leurs avantages et inconvénients en termes de leurs performances pour des applications temps réel. Dans ce contexte, nous développons une nouvelle classification de ces méthodes basée sur trois caractéristiques (cf. Figure 2.1) :

Le type d'entrée - Nous distinguons trois types de variables d'entrée : *mesurables en temps réel* comme le courant et la tension, *non mesurables en temps réel* comme la tension à vide et l'impédance d'une batterie et *estimées* par un modèle physique, électrochimique ou statistique qui peut être à la fois mesurable ou non mesurable en temps réel ;

Le type de modèle de SoC - Nous distinguons quatre types de modèle de *SoC* : *tables de correspondance*, *modèle physique* comme l'équation coulométrique et la modélisation de la batterie par un circuit électrique équivalent, *modèle électrochimique* comme la modélisation de la diffusion des particules actives entre les deux électrodes et *modèle statistique* comme les ANN et les SVM.

La procédure d'estimation du SoC - Trois procédures d'estimation de *SoC* sont considérées : *non récursive* où le *SoC* à l'instant t est estimé à partir des mesures de l'entrée à l'instant t uniquement, *récursive à boucle ouverte* où le *SoC* à l'instant t est estimé à partir du *SoC* à l'instant $t-1$ et *récursive à boucle fermée* où le *SoC* à l'instant

*The English sections of this chapter are already published in Journal of Power Sources (Kalawoun et al. [2015a]).

t est estimé à partir du SoC à l'instant $t-1$ et ajusté par la suite à partir d'une autre variable observée comme dans le cas d'un filtre de Kalman.

Cette classification nous a permis d'identifier des pistes encore inexplorées dans le domaine de l'estimation du SoC ainsi que des méthodes qui peuvent être améliorées pour obtenir un estimateur "idéal". Nous appelons un modèle de SoC "idéal" lorsqu'il permet d'estimer précisément le SoC, indépendamment des caractéristiques internes de la batterie et de ses conditions d'usage externes. Cette estimation doit aussi pouvoir être réalisée en temps réel par un BMS embarqué.

Toutefois, les caractéristiques d'un modèle idéal dépendent de l'objectif, du contexte d'utilisation de la batterie et même des spécifications du BMS. Premièrement, un modèle idéal doit être précis, en particulier lorsque le SoC est proche de 0% et de 100% pour éviter une surcharge ou une sousdécharge pouvant endommager la batterie. De plus, il doit fournir une indication linéaire du SoC en évitant tout changement brutale de cette indication afin que l'utilisateur ait confiance en la prédiction de l'autonomie de la batterie. Deuxièmement, un modèle idéal doit être robuste à l'erreur des capteurs, particulièrement dans le cas des produits commerciaux comme les véhicules électriques. En effet, plus le capteur de courant est précis, plus il est coûteux. De ce fait, tenir compte de son imprécision par le modèle SoC diminue le coût de fabrication d'un véhicule électrique. Cependant pour les expériences dans un laboratoire, les capteurs sont généralement très précis et les erreurs des capteurs peuvent ne pas être considérées. Troisièmement, un modèle idéal doit estimer le SoC dans des applications temps réel. Dans ce contexte, la batterie est souvent partiellement chargée, et les points de référence assurant le calibrage du SoC sont rarement disponibles. En plus, l'estimation du SoC d'un pack batteries, souvent utilisé dans des contextes réels, induit une complexité supplémentaire due à l'hétérogénéité des cellules formant le pack (Zheng et al. [2013]). Le temps d'affichage et de calcul de SoC est également un facteur important dans les applications temps réel. Par exemple, dans le cas des téléphones mobiles ou des PC portables la variation du SoC est moins rapide que celle dans le cas d'un véhicule électrique. De ce fait, l'algorithme d'estimation de SoC doit être plus rapide pour les applications dans lesquelles le SoC change vite. Quatrièmement, un modèle idéal doit tenir compte de la capacité de calcul du BMS. Enfin, il doit considérer l'influence des conditions d'usage sur le comportement de la batterie. Ses paramètres doivent s'adapter au cours du fonctionnement de la batterie pour suivre le changement de sa dynamique. Il doit également être capable d'alerter l'utilisateur lorsqu'il s'avère non précis et que l'apprentissage d'un nouveau modèle est indispensable.

Nous étudions toutes les méthodes d'estimation de SoC à travers le prisme de deux principaux défis : la conception d'un modèle pour estimer le SoC en tenant compte de la variabilité de la dynamique de la batterie, et l'estimation des paramètres inconnus de ce modèle.

Au sein de la batterie, de nombreuses réactions électrochimiques se superposent et compliquent la modélisation de la variabilité de sa dynamique et donc une modélisation précise du *SoC*.

Un modèle de *SoC* peut être estimé comme le rapport entre le nombre de particules actives dans l'anode et le nombre maximal que peut contenir celui-ci. Cette estimation est initialement basée sur une modélisation des réactions électrochimiques. Ces réactions sont couramment étudiées à partir du phénomène de diffusion des particules actives entre les deux électrodes comme dans Di Domenico et al. [2008], Pop et al. [2006]. Le comportement de la batterie est ainsi décrit par un système d'équations différentielles faisant intervenir des constantes électrochimiques de bas niveau (par exemple, la constante de diffusion temporelle des ions dans l'électrolyte) souvent difficilement accessibles. Cependant, le phénomène de diffusion dépend de la chimie des électrodes et de l'électrolyte et la modélisation des réactions électrochimiques doit donc être réalisée pour chaque type de batterie. Ce type de méthode est détaillé dans la Section 2.3.4.

Une méthode plus simple, dite coulométrique (Ah-counting), consiste à estimer le *SoC* par le rapport entre la quantité de charge disponible et la quantité de charge maximale que peut délivrer la batterie. Contrairement au nombre de particules actives, le calcul de la quantité de charge ne nécessite pas une modélisation des réactions électrochimiques. Cette quantité peut être calculée par le cumul du courant entrant et sortant de la batterie. Toutefois, cette méthode présente de nombreuses limitations comme le cumul de l'erreur du capteur de courant et la variation de la quantité de charge maximale que peut délivrer la batterie en fonction de ses conditions d'usage. Les limitations de la méthode coulométrique ainsi que les améliorations proposées sont présentées dans la Section 2.3.1.

D'autres méthodes consistent à approximer le comportement de la batterie par un circuit électrique équivalent pour estimer la tension à vide et indiquer la valeur du *SoC* par une table de correspondance (cf. Section 2.3.3.2), ou estimer la tension de la batterie et utiliser l'erreur entre la tension estimée et celle mesurée pour ajuster le *SoC* coulométrique à travers des outils de contrôle à boucle fermée comme un contrôleur (Codeca et al. [2008]) ou un filtre de Kalman (Plett [2004]) (cf. Section 2.3.3.1). Les circuits électriques de 1er et de 2ème ordre de Randle sont parmi les plus utilisés (Codeca et al. [2008], Dai et al. [2006], Hirai et al. [2008], Moo et al. [2007]). D'autres modèles plus complexes tiennent compte du phénomène d'hystérèsis entre la charge et la décharge (Feng and Sun [2008]) et des effets de relaxation de la tension de la batterie (Plett [2004]).

Le modèle de *SoC* peut aussi être construit en faisant abstraction des modèles physiques de la batterie. Le *SoC* est ainsi estimé des modèles statistiques de régression linéaire comme les ARMAs (Kozlowski [2003]), ou non-linéaire comme les ANNs (Affanni et al. [2003], Bo et al. [2008], Charkhgard and Farrokhi [2010], Shen [2010]) et les SVMs (Hansen and Wang [2005]). L'utilisation de l'apprentissage statistique pour estimer le *SoC* est détaillée dans la Section 2.3.5.

Suivant le modèle de SoC, les paramètres inconnus du modèle peuvent être soit identifiés à partir des expériences physiques, soit estimés grâce aux méthodes d'apprentissage statistique. Dans les deux cas, des tests de charge/décharge sont effectués sous des conditions d'usage connues. Parmi les expériences physiques, l'analyse temporelle du signal de la tension sous des tests d'impulsion de courant (Codeca et al. [2008], Hirai et al. [2008]) et l'analyse fréquentielle de la mesure d'impédance par des diagrammes de Nyquist (Lee et al. [2008]) sont les plus utilisées. Les méthodes d'apprentissage statistique consistent à estimer les paramètres à partir d'une base d'apprentissage formée des données collectées lors de ces tests de charge/décharge. La méthode la plus utilisée dans ce cadre est la méthode des moindres carrés (Dai et al. [2006], Plett [2004]).

Une fois identifiés/estimés, ces paramètres sont ensuite utilisés en ligne par le BMS pour estimer le SoC. Cependant, la variation des conditions d'usage de la batterie et/ou de ses caractéristiques internes induit un changement de son comportement. Par conséquent, les paramètres identifiés/estimés hors-ligne pour des conditions d'usage bien définies pourraient ne plus être optimaux, ce qui diminue les performances du modèle de SoC. Prenons l'exemple de la résistance interne de la batterie : la valeur de cette résistance est affectée par les conditions d'usage externes comme la température ambiante (la résistance augmente lorsque la température diminue) mais aussi par les caractéristiques internes de la batterie comme son état de santé (la résistance augmente lorsque l'état de santé diminue). Il convient de noter que cette problématique est fortement liée à la modélisation de la variabilité du comportement de la batterie.

Pour surmonter cette difficulté, plusieurs études réalisées proposent de mettre à jour les paramètres du modèle de SoC au cours du fonctionnement de la batterie. Hu et al. [2010] modifient les paramètres en temps réel grâce à une table de correspondance entre le SoC et ces paramètres. Pang et al. [2001] intègrent les paramètres du modèle dans le vecteur d'état du filtre de Kalman ; ils sont alors mis à jour automatiquement à chaque instant. Plett [2004] développe un modèle de régression polynomial qui met en correspondance les paramètres du modèle et la température ambiante.

Dans ce cadre, les méthodes d'apprentissage statistique permettent de créer des modèles de SoC à partir des tests d'usage de la batterie sans besoin d'une description physique de celle-ci. Elles peuvent être utilisées pour apprendre les paramètres d'un modèle de régression statistique ou même d'un modèle physique prédéfini à travers une base d'apprentissage issue de tests de charge/décharge comme le courant, la tension et la température de la batterie. Le modèle appris est ainsi valide lorsque les conditions d'usage de la batterie sont proches de celles de sa base d'apprentissage. De ce fait, afin d'avoir un modèle générique, indépendant de la composition chimique de la batterie et de ses conditions d'usage, une base d'apprentissage exhaustive formée de toutes les combinaisons possibles de conditions d'usage et de caractéristiques internes est nécessaire. Les difficultés de choix d'une base d'apprentissage sont discutées dans la Section 2.3.5.5. Cependant dans des applications temps réel, le comportement de la batterie change au

cours de son fonctionnement en fonction de ses conditions d'usage non contrôlables. Par conséquence, un seul modèle de SoC qui ne tient pas compte de ces changements ne peut pas être valide tout au long du fonctionnement de la batterie. Pour surmonter cette problématique, deux solutions peuvent être envisagées. La première consiste à mettre à jour en ligne les paramètres du modèle (cf. Section 8.2). Cette solution est contrainte par la limitation de capacité de calcul du BMS et l'absence d'une vraie valeur de SoC. La seconde solution consiste à tenir compte du changement de comportement de la batterie lors de la conception du modèle de SoC. La suite de cette étude vise à mettre en oeuvre la deuxième solution. Quant à la première, elle est maintenue en tant que perspective à considérer.

Contents

The state of charge of an electric battery (SoC) is essential to calculate its autonomy and its available energy. An accurate SoC is fundamental to obtain an efficient control strategy to manage energy, as well as to guarantee a safe utilization of the battery by preventing under or over-charge that may lead to permanent damage. Indeed, the energy management plays a significant role in extending and optimizing the lifetime of the battery.

The electric battery being a complex electrochemical system, neither its remaining capacity nor its SoC can be directly measured through a sensor. In addition, the dynamics of the battery depend not only on its usage conditions like the current profile and the ambient temperature, but also on its internal characteristics like its state of health and its internal resistance. This makes the establishment of a reliable SoC estimator difficult, especially in real-time applications.

Two difficulties constrain the performances of a real-time SoC estimator. The first comes from the limited hardware resources of the Battery Management System (BMS). The second lies in the fact that the dynamics of the battery depend on its internal characteristics and its usage conditions.

Hence, we point out the need for an efficient model able to estimate the SoC of any battery, regardless of its internal characteristics and its usage conditions in real-time applications. Such a model will be referred hereafter to as *"ideal" SoC estimator*.

By taking a closer look at the existing SoC estimation methods, it is clear that none possess the characteristics of this "ideal" estimator. In order to obtain it, a suitable approach must be identified among the large number of the existing ones. This identification can be achieved through a comprehensive classification of the existing methods.

The SoC estimation methods can be classified with respect to different criteria. The first one is the type of the input variables, either measured or estimated. The second one is the type of the SoC model, which can be a physical, an electrochemical or a statistical regression model. The third criterion deals with the temporal dimension: static methods like those based on lookup tables and dynamic methods like those based on state-space models. Also the methods can be classified according to the battery technology, namely Li-ion, Ni-MH, Lead-acid and so on. Finally, the classification can be made based on the mathematical tools used by the estimation method, such as a Kalman filter, an

artificial neural network, a fuzzy logic, etc. It is important to distinguish between the tools applied to the *SoC* estimation and those used to estimate the input variables. Indeed, the more the classification criteria are relevant, the more easily we can identify the methods that can be improved in order to provide an ideal *SoC* estimator and flesh out new ways of developing it.

Several reviews of the existing *SoC* estimation methods are available in the literature. Lu et al. [2013], Piller et al. [2001], Zhang and Lee [2011] and Waag et al. [2014] give an overview of the methods without classifying them. The drawbacks and advantages of each method are presented, but this is not sufficient to provide an exhaustive and well structured vision on the path to be followed to develop an "ideal" *SoC* estimator. Pop et al. [2005] give a chronological review of the estimation methods before classifying them under three categories: direct measurement methods, book-keeping systems that involve basic and modified Ah-counting, and adaptive systems which are supposed to be self-designed and automatically adjusted according to the battery aging and the battery dynamics. The Kalman filter, the artificial neural network and the fuzzy logic approaches were allocated to this category, but the authors acknowledge that these methods have strong limits and cannot adapt to all usage conditions. Chang [2013] gives a similar classification while adding to it a fourth category of hybrid methods including methods combining the three categories distinguished by Pop et al. [2005].

Hence, the classification of Pop et al. [2005] and Chang [2013] doesn't make a distinction between the type of *SoC* model and that of the input variables model, as it focuses on the temporal and technological criteria.

Subsequently, the above classifications do not strictly abide by all earlier mentioned criteria, thus rendering difficult the identification of the aspects to be improved.

In this chapter, we introduce a novel classification of the *SoC* estimation methods based on their concept, their adaptability and their performances in real-time applications. The importance of machine learning methods in providing an "ideal" *SoC* estimator is also stressed.

2.1 State of Charge of an electric battery

2.1.1 Definition of the State of Charge

The *SoC* of a battery is defined as the ratio between the *remaining* and the *actual capacities*. The actual capacity is the maximum capacity that can be withdrawn from the fully charged battery under specific discharge conditions. It depends, among other factors, on the current profile, the ambient temperature and the state of health of the battery. A battery being a chemical energy storage system, there is no sensor that

directly measures the SoC. Consequently, these remaining and actual capacities must be estimated.

2.1.2 Challenges in estimating the battery capacity

One way to compute the remaining capacity of a battery is the *discharge test*. It consists of discharging the battery under the reference conditions to reach the end of the discharge criterion (i.e., the cutoff voltage). However, the discharge test cannot be applied in real-time applications, as well as in offline application as it leads to a loss of energy.

Accordingly, the SoC is generally calculated based on the Ah-counting equation given by:

$$SoC_t = SoC_{t_0} + \frac{\int_{t_0}^{t} I_\tau \, \mathrm{d}\tau}{C_{\text{actual}}}, \tag{2.1}$$

where SoC_{t_0} is the initial SoC at time t_0, I_τ the algebraic current measurement: positive for a charge current and negative for a discharge current and C_{actual} the actual capacity. A numerical implementation of this equation requires a temporal discretization. The SoC is then calculated as follows:

$$SoC_t = SoC_{t-\Delta t} + \frac{I_t \times \Delta t}{C_{\text{actual}}}, \tag{2.2}$$

where Δt is the sampling interval, which can be constant or variable.

It is clear that the precision of this method depends on the accuracy of the current sensor and the sampling interval. Nevertheless, the actual capacity is not constant during the battery charge/discharge; it depends on the internal characteristics and the usage conditions. In a real-time context such as an electric vehicle, the usage conditions are uncontrolled as they depend on the user's behavior, weather conditions, road conditions, etc. Accordingly, in some situations, the state of charge can be lower than 0 or higher than 100. The establishment of a deterministic function to provide a reliable value of the actual capacity is a challenging issue.

In the next section, a novel classification of the SoC estimators is presented.

2.2 Novel classification of the SoC estimation methods

From a global point of view every estimation method can be characterized by the input variables, the SoC model and the type of the SoC estimation processing (see Figure 2.1).

The input variables can be either directly measured by a sensor, or estimated through a physical, electrochemical or statistical regression model. For instance, the current is

measured online, but the OCV can be measured offline or estimated online using an suitable model. Thereby, the identification of the type of the input variables is the first step when determining if a SoC estimation method can be improved to become "ideal".

Three types of SoC models can be distinguished, namely a lookup table, a physical or a statistical regression model. The lookup table represents a direct relation between the SoC and a measured physical quantity like the OCV, the internal impedance and the electrolyte density. The physical model is based on the SoC definition; two physical models can be identified: the macroscopic model where the SoC is calculated using the Ah-counting and the microscopic model where the SoC is defined as the ratio between the available and the maximum concentration of the active material in the anode. The statistical regression model describes a linear or a nonlinear relationship between the SoC and the input variables, and does not necessarily have a physical interpretation. The type of the SoC model is the most important classification criterion. It is important not to confuse the SoC model with the models used to estimate other battery variables.

Finally, the SoC estimation methods are characterized by the type of processing: open- and closed-loop processing. In closed-loop processing, the gap between the measured and the predicted value of a given physical quantity is used to adjust the estimated value of the state of interest. We consider that the use of a lookup table for the SoC estimation is not related to any category.

The rest of this chapter is organized in five sections: the improved Ah-counting methods, the methods based on directly measured input variables, the methods based on inputs estimated with physical models, the methods based on inputs estimated with electrochemical models, and the SoC estimation based on the machine learning methods.

There could as many families and/or sections as there are combinations of types of input variables, SoC models, and processing methods (see Figure 2.1). We could have comprehensively listed all existing and/or possible families; and allocated a section to each family. However, this leads to a size disparity between the sections. Some parts would even be empty. Consequently, we have structured the sections, except for the Ah-counting, according to the type of the input variables models. As a reading aid, at the beginning of each section, we have introduced a table summarizing the novel classification characteristics of the presented SoC estimation methods.

2.3 State of Charge estimation methods

2.3.1 Improved Ah-counting

The SoC definition given by (2.2) offers a generic SoC estimation method, called *Ampere hour counting* (Ah-counting). This method is suitable for all battery types and used as a reference method for the evaluation of the performance of any other estimation method.

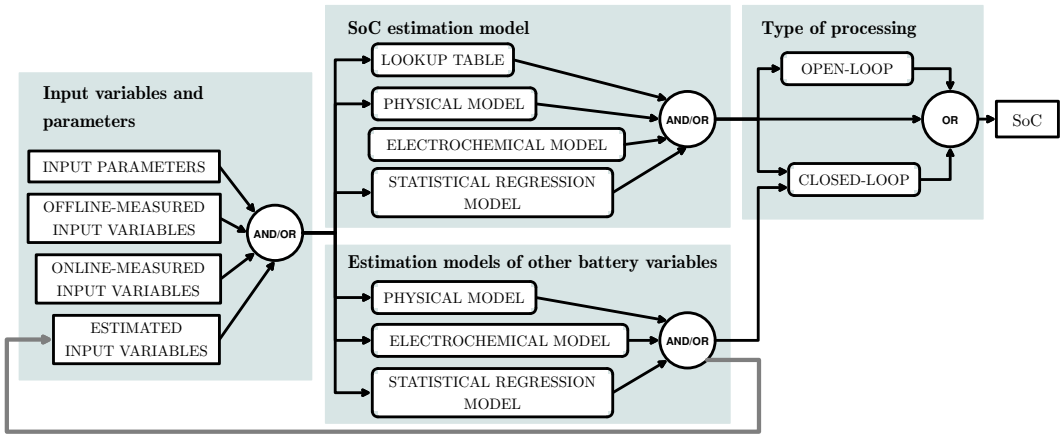

FIGURE 2.1: Novel generalized classification of the *SoC* estimation methods based on the type of the *SoC* model, the type of the input variables and parameters and the processing type

Indeed, given an actual capacity, this method requires only a measure of the current. This optimizes the volume of information exchanged between the battery and the BMS, and reduces the allocated memory space as well as the computational time.

Despite its apparent simplicity, this method holds a number of limitations. Some of them have been successfully overcome, but some remain unsolved largely because the Ah-counting does not consider the complexity of the electrochemical system. Below we discuss the sources of the imprecisions of the Ah-counting model (2.2) and the improvements that were brought to it until now.

2.3.1.1 Calculation of SoC_{t_0}

The Ah-counting approach supposes that the initial value SoC_{t_0} is known, but even if the SoC is equal to 100% for a fully charged battery, and to 0% for a fully discharged battery, questions remain about the initial SoC of a partially charged battery. To overcome this problem, the SoC_{t_0} is estimated using a Kalman filter in Wang et al. [2007], or using a lookup table OCV/SoC after a long rest period in Ng et al. [2009].

2.3.1.2 Current leakages in rest period

During the rest period, secondary reactions lead to current leakages and therefore reduce the remaining capacity. This phenomenon is known as self-discharge (Iliev and Pavlov [1982], Vetter et al. [2005], Wu and White [2000]). The rate of the self-discharge depends, among other factors, on the duration of the rest period, the ambient temperature, the SoC and the state of health (Broussely et al. [2001], Zachlin [1942]). The Ah-counting method does not take into account these current leakages, and the SoC remains constant when the current is equal to 0. To overcome this problem, Wang et al. [2007] establish

a linear relationship between the rate of the self-discharge and the duration of the rest period.

2.3.1.3 Coulombic efficiency

Undesired reactions, which lead to current loss, can also consume or produce an electric charge at either the positive or the negative electrode. The Coulombic efficiency measures the delivered/ withdrawn current loss (cf. Smith et al. [2010]). However, its value depends essentially on the ambient temperature, the current profile, the state of health and even on the SoC of the battery (cf. Bond et al. [2013]). The establishment of a deterministic function to provide the Coulombic efficiency remains a challenging problem. For an accurate estimation of the SoC, most algorithms include the Coulombic efficiency η in the formula of Ah-counting:

$$SoC_t = SoC_{t-\Delta t} + \frac{\eta \cdot I_t \cdot \Delta t}{C_{\text{actual}}}. \tag{2.3}$$

Usually two values of the Coulombic efficiency are considered: η_c for the charge and η_d for the discharge (cf. Dai et al. [2006], Liao et al. [2011], Wang et al. [2007]). Ng et al. [2009] use a variable Coulombic efficiency depending on the SoC. Alzieu et al. [1997] integrate the influence of the ambient temperature and the current rate on the Coulombic efficiency: $\eta = K_{\Delta T} \cdot K_{\Delta I}$ where $K_{\Delta T}$ and $K_{\Delta I}$ represent the influence of the variation of the temperature ΔT and the current ΔI respectively. Malkhandi [2006] uses a fuzzy logic model to estimate the Coulombic efficiency based on the current and the temperature values.

2.3.1.4 Error of the current sensor

When integrating the current over time the Ah-counting does not use a feedback loop to offset an eventual estimation error. From a control theory perspective, the system is referred to as *open*: the estimation error is accumulated over time and increases the bias of the estimator. This error is mainly a result of the inaccuracy of the current measurement which can be caused by the sensor error and the sampling frequency.

Calibration points are often used to adjust the estimated SoC. Hence, these points are available for specific states of the battery: 0% for a fully discharged battery, 100% for a fully charged battery, and the SoC estimated by an OCV measurement after a long rest period. However, these calibration points are seldom available in a real-time application where the battery is often partially charged and no OCV measurement is available.

2.3.1.5 Variation of the battery capacity

The remaining capacity as well as the actual capacity change during the charge/discharge of the battery according to its internal characteristics and its usage conditions, such as the current profile, the ambient temperature and the state of health (cf. Chubb and Harner [1935], Linden [1947]). Gaddam et al. [2000] use a fuzzy logic system to estimate the actual capacity based on the current rate and the ambient temperature.

Furthermore, the alteration of the remaining capacity affects not only the remaining capacity at instant t but also the one at $t-1$. This is a fundamental issue in the Ah-counting method, as it is based on a recursive computation.

In summary, the Ah-counting is a simple and generic SoC model suitable for all battery technologies and can be used in real-time applications. However, this model does not take into account the calculation of the initial state of charge, the precision of the current sensor, the Coulombic efficiency, and the variation of the actual capacity. Multiple improvements are developed to overcome these difficulties. Nevertheless, they remain contestable, especially for an online SoC estimation, due to the complexity of the battery dynamics which depends strongly on the uncontrolled usage conditions.

In the next section, the SoC estimation methods using measured input variables are discussed.

2.3.2 SoC estimator based on directly measured input variables

Another concept of the SoC estimation is based on a direct relation between the SoC and a measured physical quantity like the OCV and the internal impedance. This relation can be described using a lookup table or a statistical regression model. Unlike the Ah-counting, this relation is not generic and depends on the battery technology, its state of health and its usage conditions.

A *lookup table* is a one-to-one relation obtained through an empirical way. Several laboratory experiments need to be realized in order to construct one table; various tables are required depending on the usage conditions and the inherent battery properties. In this case, the accuracy of the SoC depends on the precision of the measured physical quantity, and the quality and the wealth of the lookup tables.

In the following we describe and analyze the SoC estimation methods taking as input variables a measure of the OCV, the impedance Z and other measurements specific to certain battery technologies. Table 2.1 summarizes these methods according to the novel classification (see Figure 2.1).

Measured input variables	OCV, Z, V_t, electrolyte density
SoC estimation model	Lookup table, statistical model
Other models	-
Type of processing	Non-recursive

TABLE 2.1: Classification criteria for SoC estimator based on directly measured input variables

2.3.2.1 Open circuit voltage measurement

The term *voltage* refers to the electrical potential difference between two electrodes of an electrochemical cell. The *open circuit voltage (OCV)* is the measured terminal voltage for zero current (Kurzweil [2009]). Thus, it gives an indication of the available energy and is directly proportional to the SoC (see for instance Christianson and Bourke [1976], Parfitt et al. [2010], Pop et al. [2006]). Peled et al. [1988] use the OCV and the ambient temperature as input parameters for a lookup table to obtain the SoC.

However, the lookup table utilization/application remains limited due to several reasons. First, the OCV cannot be directly measured in a real-time application as an accurate value of the OCV requires an extended rest period (i.e., $I = 0A$) (Petzl and Danzer [2013]). Some techniques can be used to overcome the problem of the rest duration like the interpolation of the OCV data during the rest time to get a higher resolution of the OCV curve, and the extrapolation of the relaxation behavior using a battery modeling (Petzl and Danzer [2013]). Second, the OCV/SoC relationship presents a hysteresis behavior according to the charge/discharge history even if the rest time is very long. This behavior is well known but needs specific experimental tests to be deeply characterized (Roscher and Sauer [2011], Srinivasan et al. [2001]). Third, the OCV/SoC relation is not generic; it depends on the battery technology. For example, there is a slight change in the OCV of a LiFePO4 battery when the $SoC \in [20\%, 80\%]$. A small error on the OCV measurements induces a large error on the estimated SoC. Moreover, the OCV/SoC relationship is not constant: it is influenced by the state of health of the battery (Pop et al. [2007]) and the ambient temperature (Peled et al. [1988]).

Hence, to cover all conditions, a lot of lookup tables should be built. This requires a lot of laboratory experiments and a large memory to store all these tables in a real-time application.

2.3.2.2 Internal impedance measurements

The electrochemical impedance of a battery characterizes its dynamics behavior, that is, its response to an excitation of a small amplitude (Rodrigues et al. [2000], Willihnganz [1941]). Two categories of methods for the impedance measurement can be distinguished: active and passive methods.

The *electrical impedance spectroscopy* (EIS), the major example of the active methods, involves an excitation of the battery with a small AC (for alternative current) signal over a wide frequency range typically from 10kHz to 10^{-5}Hz. This frequency range depends on the battery chemistry (Buller et al. [2005], Diard et al. [1997], Rodrigues et al. [2000]). In addition, the response of the battery is strongly influenced by the ambient temperature (Deng et al. [2013]). This measurement method cannot be used in a real-time application due to its hardware complexity and its high cost.

The passive methods search an impulse in the current profile. Once found, this current impulse and the corresponding output voltage are used to estimate the internal impedance. These methods are not reliable since a perfect impulse can hardly be detected in real conditions.

A review of the investigations for the applicability of impedance measurements as a test for the *SoC* of lead-acid and nickel-cadmium batteries is detailed by Huet [1998]. These methods attempt to establish a relationship between the real and/or imaginary impedance components and the *SoC*. Thereby, the presented models are not generic and remain specific to a particular technology. Huet concludes that the impedance measurement is strongly influenced by the ambient temperature, particularly at low frequencies. This technique seems to be more suitable for lead-acid than nickel-cadmium batteries. Indeed for nickel-cadmium batteries, the ohmic resistance (i.e., real impedance component) variations with the *SoC* are much lower than the variations for lead-acid batteries.

Salkind et al. [1999] use a fuzzy logic model to establish a relationship between the *SoC* and the impedance measured based on an EIS at different frequencies. Thus the fuzzy logic model limits the impact of the unreliability of the impedance measurement.

In short, it is difficult to measure the impedance online. In addition, the measured values of the impedance depend greatly on the measurement method and they are sensitive to measurement conditions. Furthermore, the type of relationship between this impedance and the *SoC* is not generic for all battery technologies.

2.3.2.3 Other measurements

Some *SoC* estimation techniques are suitable for particular battery technologies. For example, the lead-acid batteries present a specific behavior called *coup de fouet* which is the initial voltage drop when discharging a fully charged battery (Delaille et al. [2006]). It can be used to calibrate the *SoC* to 100% and provides information regarding the state of health of the battery.

A second specific method for the lead-acid battery is based on a direct relation between the *SoC* and the electrolyte density measurement. However, this method is very sensitive to the temperature and the impurities present in the electrolyte (Piller et al. [2001]).

All of the *SoC* estimation methods mentioned and analyzed in this section need a direct measurement of an inherent physical quantity of the battery. Nevertheless, their measurement is difficult, even impossible, to carry out especially in a real-time context. To get around this difficulty, a chosen physical quantity can be estimated using a physical, an electrochemical or a statistical regression model (see Figure 2.1).

2.3.3 SoC estimator based on inputs estimated using physical models

The estimation of input variables is usually based on a physical representation of the dynamics of the battery. It consists of establishing a relationship between an input variable, such as the output voltage or the *OCV*, and the battery model parameters derived from its physical representation.

Two cases are generally considered when using estimated input variables. In the first one, the input variable can be estimated by a physical model as well as measured in real-time conditions. The difference between the estimated and measured input variable is then used to improve the *SoC* model via a closed-loop processing. In the second case, the input variable cannot be directly measured but is estimated in real-time by a physical model. The *SoC* is then estimated via a predefined relationship (lookup table or regression model) using this estimated input variable.

In this section, we present several *SoC* estimation methods where the estimated input variables are either the voltage or the *OCV*. Table 2.2 summarizes the variables and models involved in these methods according to the novel classification (see Figure 2.1).

Estimated input variables	Output voltage	*OCV*
SoC estimation model	Ah-counting	Lookup table, regression model
Other models	Physical model of V_t	Physical model of *OCV*
Type of processing	Closed-loop	Closed- & Open-loop

TABLE 2.2: Classification criteria for *SoC* obtained from inputs estimated with physical models

2.3.3.1 Voltage estimation

The battery voltage can be measured and estimated at the same time. This is why this input variable is very suitable for a closed-loop processing: the gap between the measured and the estimated voltages is used in real-time to adjust the *SoC* obtained by the Ah-counting method. Several battery voltage models are described below. Examples of closed-loop processing techniques, such as a controller and a Kalman Filter, are also presented.

The Ah-counting method can be improved using a controller as in Codeca et al. [2008]. Figure 2.2 shows a flow diagram of the SoC adjustment using a controller. Accordingly, the voltage is estimated using the 2nd order Randle model. The difference between the estimated $\widehat{V}_t$ and the measured V_t voltages is then integrated in the feedback of the controller in order to adjust the Ah-counting value $\hat{SoC}_t^{AH}$:

$$\hat{SoC}_t = \hat{SoC}_t^{AH} + K_p(V_t - \widehat{V}_t), \tag{2.4}$$

where K_p is the controller parameter.

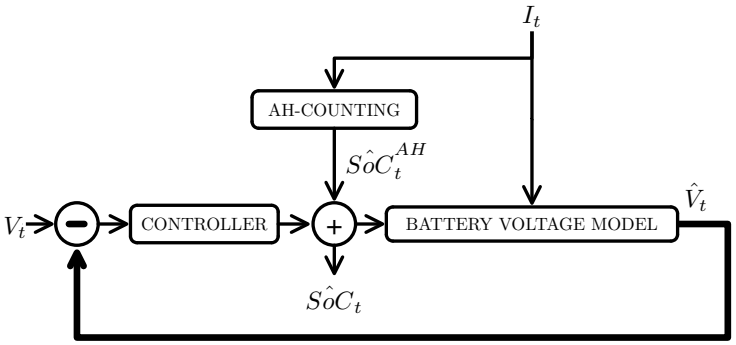

FIGURE 2.2: Concept of SoC adjustment through a controller

More efficient than a controller, a Kalman filter takes into account the imprecisions of the voltage measurement and those of the SoC model. Indeed, a state-space model is defined by a *transition equation* describing the dynamics of the state of interest of the system X_t, and an *observation equation* specifying how the observations Y_t is influenced by X_t:

$$X_{t+1} = A \cdot X_t + B \cdot u_t + W_t, \qquad \text{(Transition equation)}$$
$$Y_t = C \cdot X_t + D \cdot u_t + \Sigma_t, \qquad \text{(Observation equation)}$$

where A and B represent the transition and observation matrix respectively, C and D model the influence of the input u_t on X_t and Y_t, and W_t and Σ_t are white-noise processes used to model the error of the state and observation equations respectively.

The Kalman filter provides an optimal estimation of X_t, in a mean squared error sense, when the state and observation models are linear and all noises are Gaussian (Kalman [1960]). When one of these models is nonlinear, a sub-optimal extension of the Kalman filter, called Extended Kalman Filter (EKF), is used (Haykin [2001]).

The performance and efficiency of an EKF shall be determined by the reliability of the state and observation equations. Figure 2.3 shows a flow diagram of the SoC estimation using an EKF. Accordingly, the unknown state of interest X_t is predicted using the state equation given the observation y_{t-1}. The observation Y_t is then predicted using the observation equation, based on the predicted value $\widehat{x}_t$. Finally, the predicted $\widehat{x}_t$ is

adjusted based on the gap between the predicted value and the measured value of the observation. The power of the EKF is that this gap is weighed by an adaptive Kalman gain that depends on the correlation between the observed Y_t and the unknown X_t variables.

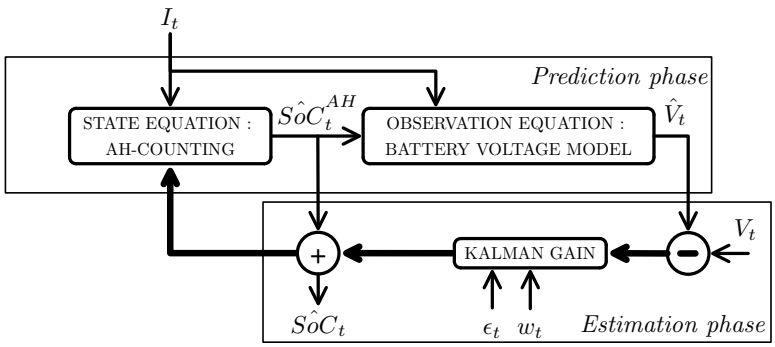

FIGURE 2.3: SoC estimator through an extended Kalman filter, where I_t is the input u_t, SoC_t the state variable X_t and V_t the observed variable Y_t

In battery applications, the state and observation models depend on the battery model itself. A review of several types of battery models used on electric vehicles is given in Chen and Lin [2005].

Plett [2004] develops and analyzes several observation models. The SoC is considered to be the unknown state of interest X_t, and the battery voltage the observed variable Y_t. To describe the relationship between the SoC and the voltage, Plett [2004] creates a *combined model* by merging three existing voltage models: Shepherd, Unnewehr universal and Nernst models:

$$V_t = K_0 - \frac{K_1}{SoC_t} - K_2 SoC_t + K_3 \ln(SoC_t) + K_4 \ln(1 - SoC_t) - RI_t, \tag{2.5}$$

where $\{K_0, \ldots, K_4\}$ are the model parameters to be estimated, R is the internal resistance, and I_t is the current. This combined model estimates the voltage efficiently but suffers from computation problems when the SoC is near to 0% and 100%, because of the terms $1/SoC_t$, $\log(SoC_t)$, etc. Then, Plett [2004] considers that these terms can be reduced to an $OCV(SoC_t)$ function. In this *simple model*, the relationship between the battery voltage and its SoC is described as follows:

$$V_t = OCV(SoC_t) + R \cdot I_t, \tag{2.6}$$

where R is the internal resistance of the battery. Plett [2004] does not give a theoretical description of the $OCV(SoC_t)$ function, but a graphical one; the ratio $\partial OCV_t / \partial SoC_t$ is calculated using this graphical representation.

Dai et al. [2006] use the 2nd order Randle model to physically represent a Li-ion battery pack:

$$V_t = OCV(SoC_t) + V_t^{RC1} + V_t^{RC2} + R \cdot I_t, \tag{2.7}$$

where V_t, the observed variable, is the pack terminal voltage, V_t^{RC1} and V_t^{RC2} are the voltages across the two RC circuits estimated using the EKF, R is the internal resistance of the battery and $OCV(SoC_t)$ is the relationship between the OCV and SoC. As in Plett [2004], this relationship is not clearly described.

However, the precision of the voltage estimation depends essentially on the quality of $OCV(SoC_t)$. We have seen earlier (Section 2.3.2) that this relationship depends not only on the internal characteristics of the battery but also on the usage conditions. In addition, in certain cases, the OCV has a limited variation on a broad SoC range, thus calling into question the performance of the EKF. As a result, this relationship cannot be described using a single equation or a single lookup table.

Also, the voltage depends on the polarization of the battery. This is highlighted by the hysteresis effect occurring in the battery. In order to describe the voltage hysteresis effect, Plett [2004] develops the *zero-state hysteresis model*, an improvement of the simple model (Eq. 2.6) discussed above. A switching factor is thus introduced on the state vector of the model. This factor depends on the sign of the current, and the difference between the maximum positive and the minimum negative of the hysteresis. However, this hysteresis factor flips immediately when the current sign changes. A slow transition may be modeled considering that the battery voltage contains an unknown hysteresis voltage that is also estimated using the EKF. For a Ni-MH battery, Feng and Sun [2008] represent the evolution of the hysteresis factor using an exponential function depending on the current value, a parameter derived from the hysteresis voltage test and the maximum value of the hysteresis voltage. For a Li-ion polymer battery, Plett [2004] considers the evolution of the hysteresis voltage as an exponential function of the SoC, and the current value.

In a more advanced model called *enhanced self-correcting model*, Plett [2004] considers that the relaxation effect during pulsed current events and rest periods is described by a low-pass filter on the current. The model forces the estimated voltage to converge to the OCV after a rest period.

Hence, the EKF is a popular tool to estimate the SoC; but several issues remain unsolved. First, the EKF is not an optimal estimator, and it leads to an inaccurate estimation when the state or observation equations are strongly nonlinear. Second, the EKF strongest limitation is the need to initialize different filter parameters, like the state of interest X_t and the covariance matrix. Indeed, the EKF may diverge quickly if its parameters are inadequately initialized. To solve this problem Han et al. [2009] use an adaptive Kalman filter that estimates automatically the initial covariance matrix.

2.3.3.2 Open circuit voltage estimation

The open circuit voltage cannot be measured in real-time but may be estimated based on a suitable battery voltage model. In Pang et al. [2001], an OCV model is developed using an equivalent-electric battery model. Then this model is used as a transition equation to estimate the OCV through an EKF. Hirai et al. [2008] and Moo et al. [2007] calculate the OCV_t directly based on the current I_t and the voltage V_t using the Randle equivalent model. The OCV can also be calculated using a state observer as in Chen et al. [2012], Li et al. [2013] and Hu et al. [2010]. A state observer estimates the state of interest $\widehat{x}_t$ (i.e., the unknown OCV) as follows:

$$\widehat{x}_{t+1} = f(\widehat{x}_t) + G \cdot (\widehat{y}_t - y_t). \tag{2.8}$$

In battery applications, $f(\cdot)$ is the transition equation modeling the battery behavior, Y_t is usually the observed voltage, $\widehat{y}_t$ is the estimated voltage based on an electric model, and G is the observer gain which can be constant as in Chen et al. [2012], Li et al. [2013] or variable as in Hu et al. [2010].

In Hu et al. [2010], the OCV is estimated using an adaptive Luenburger observer. The observer gain is adaptively adjusted using a stochastic gradient approach so as to reduce the error between estimated and measured battery voltages (see Figure 2.4).

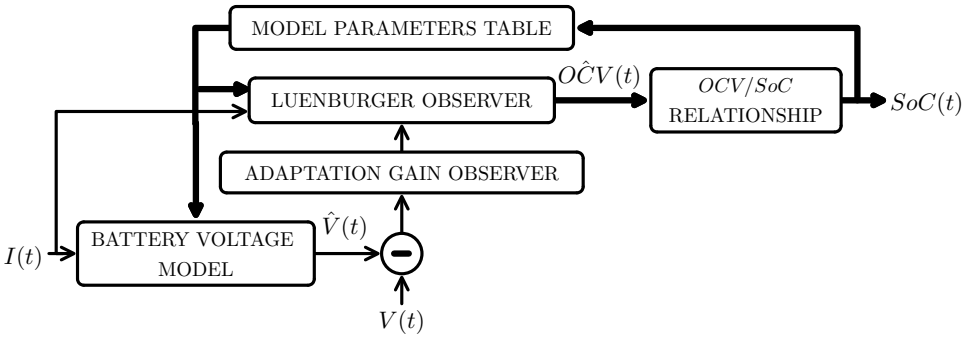

FIGURE 2.4: SoC estimator through an adaptive Luenburger observer

Once the OCV is estimated, the SoC is then calculated using a predefined relationship which is considered as linear in Pang et al. [2001] and Moo et al. [2007] for Lead-acid batteries, and exponential in Hirai et al. [2008] for Li-ion ones.

To model the hysteresis effect between the SoC and the OCV, Tang et al. [2008] use the Preisach operator to decompose this hysteresis in a N elementary hysteresis called hysteron. Each hysteron i is characterized by: a SoC_i and a portion w_i. The SoC is then estimated as follows:

$$\widehat{SoC} = \sum_{i=1}^{N} w_i \cdot SoC_i. \tag{2.9}$$

In the next section, the identification of the unknown parameters of the presented models is discussed.

2.3.3.3 Parameters identification

The physical identification of the parameters of the state and observation models can generally be classified into two categories:

1. The time domain which analyzes the voltage and the current signals under pulse current tests as in Codeca et al. [2008], Hirai et al. [2008], Lee et al. [2008].
2. The frequency domain like the analysis of the impedance measurement of the battery using Nyquist plots as in Lee et al. [2008]. More efficient than time domain analysis methods, the impedance measurement suffers from several limitations mentioned in Section 2.3.2.

In both categories, the parameters are identified offline, and do not change according to the usage conditions. As a result, the model gives an accuarate SoC when the usage conditions are close to the experimental conditions.

However, these parameters must change during the charge/discharge of the battery as they depend on the usage conditions (Dong et al. [2011], Moss et al. [2008], Parfitt et al. [2010], Pavlov and Petkova [2002]). Attempts have been made to overcome this problem so as to integrate the possibility of changes over time. Hu et al. [2010] create a lookup table to identify the model parameters for a specific SoC. This method is based on an estimated and therefore inaccurate SoC, and on a lookup table whose limitations are presented in Section 2.3.2. Pang et al. [2001] incorporate all model parameters in the vector state of the state-space model; so they are estimated at each time step using an EKF. This method can use a lot of BMS resources despite the fact that these parameters change slowly and do not need to be frequently estimated. Hirai et al. [2008] identify the model parameters for a specific temperature and SoC. This model cannot be efficiently generalized to situations not seen previously. Plett [2004] develops a more advanced model by performing joint optimization over the temperature range where every parameter is represented by a continuous polynomial (4th order) of temperature.

The model parameters can also be estimated using machine learning methods. In battery applications, the most used method is the least-squares method as in Dai et al. [2006] and Plett [2004].

In more sophisticated methods, the estimated parameters can be updated during the charge/discharge of the battery and adapted according to the usage conditions. These techniques are designated as *online learning* and will be discussed in the perspectives of this work (cf. Section 8.2).

In short, the presented methods are generally used either to improve the Ah-counting method, or to estimate the OCV and benefits from the strong link between the OCV and the SoC. A proper identification of the unknown model parameters is crucial to provide an efficient model. However, these parameters change during charge/discharge of the battery, and depend on uncontrolled usage conditions. The proposed improvements (Hirai et al. [2008], Hu et al. [2010], Pang et al. [2001], Plett [2004]) remain contestable as they do not take into account the limited hardware resources. In addition, the improved model cannot be efficiently generalized to cases not previously seen.

2.3.4 SoC estimator based on inputs estimated using electrochemical models

Several phenomena occur alongside the main redox reaction, like the diffusion and the migration of ions, the transition phase and the ohmic resistance. Thus, the SoC is influenced by all these phenomena and cannot be directly measured. A micro modeling of these phenomena can be achieved through an electrochemical model, such as the single particle model (Guo et al. [2011]) and porous electrode model (Newman and Tiedemann [1975]). This electrochemical model can be used to estimate the OCV or the concentration of the active material in the anode, thus allowing the computation of the SoC using a suitable estimation model. In the following, we present several SoC estimation methods where the input variables are estimated using electrochemical models. Table 2.3 describes the SoC estimation methods of this Section according to the novel classification (see Figure 2.1).

Estimated input variables	OCV, overvolatege, Lithium concentration
SoC estimation model	Physical
Other models	Electrochemical models of OCV, overpotential
Type of processing	Closed-loop

TABLE 2.3: Main classification criteria for SoC obtained from inputs estimated via electrochemical models

2.3.4.1 Open circuit voltage estimation

Based on an electrochemical model, two physical quantities, namely the OCV and the overvoltage, are estimated in Pop et al. [2006]. Two SoC estimation methods are then presented. The first one is based on an OCV/SoC electrochemical relationship and the second one consists in improving the Ah-counting considering the overpotential. Indeed, due to the overpotential, the remaining charge cannot be withdrawn from the battery, because the battery voltage would drop below the cutoff voltage. To incorporate

the influence of the ambient temperature, a linear relationship between it and the other model parameters is considered. The main limitation of the two models is that the values of the parameters depend on the determination method and experimental conditions.

2.3.4.2 Estimation of the lithium concentration

In a Li-ion battery, the SoC is defined as the ratio between the average and the maximum concentration of lithium inside the anode. Santhanagopalan and White [2006] model the battery by a single spherical particle. Thus, each electrode is modeled by a single spherical particle. The behavior of each electrode is modeled by a diffusion equation governed by Fick's laws, and then an EKF is used to solve these diffusion equations and estimate the concentration of lithium inside the two electrodes.

A porous battery model is used in Di Domenico et al. [2008]. Thus, each electrode is modeled by a solid matrix immerged in the electrolyte. Unlike the previous method, each electrode is decomposed into M parts, with a diffusion equation being calculated per part. A Kalman filter with $M-1$ state equations is used to estimate the concentration of lithium inside the anode.

In both methods, the observation equation of the state-space model is based on a relationship between the battery voltage and the concentration of lithium in the anode.

Nevertheless, the SoC estimation based on electrochemical models has several limitations. First, the electrochemical processes depend widely on the battery technology and the fabrication process. Therefore, even for an identical technology and provider, a small disparity can greatly affect the low-level battery parameters. Second, the battery state of health, which affects largely the physical and the chemical properties of any battery, is not considered in these models: the aging of the battery decreases the number of active particles and modifies the conductivity of the electrolyte (Armenta-Deu and Donaire [1996], Barré et al. [2013], Vetter et al. [2005]). Moreover, electrochemical models require knowledge of low-level battery parameters values (for instance, the electrolyte diffusion coefficient) which is impossible in some contexts.

2.3.5 SoC estimator based on machine learning methods

The majority of SoC models presented above have a set of parameters that need to be identified, for instance, the parameters of

- the relationship between the OCV and the SoC as in Hirai et al. [2008], Pang et al. [2001] and Moo et al. [2007],
- the state and the observation models of an EKF as in Lee et al. [2008].

The estimation of these parameters can be done using machine learning methods. The least squares method, one of the simplest machine learning methods, is already used in the batteries domain in order to estimate the parameters of the battery voltage models as in Dai et al. [2006] and Plett [2004]. However, the machine learning domain contains more advanced methods that can efficiently estimate the parameters of a physical model.

In the following, we present the principle of machine learning methods, before we discuss the ones used to estimate the SoC.

2.3.5.1 Principle of the machine learning methods

Let us consider

$$z_t = f(r_t, \mathbf{w}), \tag{2.10}$$

where z_t is the system output, r_t the known input vector and $\mathbf{w}$ the model parameters to be estimated. The machine learning methods comprise three major phases: the construction of the training dataset phase, the learning phase, and the estimation phase (see Figure 2.5). The first and second phases are often performed offline, while the third phase is achieved online.

In the first phase, the input and the output variables are considered to be available (directly measured or calculated). A *training dataset* $\mathcal{D} = \{(r_i, z_i) \mid i = 1, \ldots, N\}$, where N is the size of the dataset, is collected through experimental tests. In a battery application, z represents the SoC and r the input vector that can be a set of measured variables like the current, the voltage and the temperature, or estimated variables like the OCV. A training dataset is generally collected in a laboratory under controlled conditions: specific ambient temperature, specific current profile, etc. The SoC values in the training dataset are obtained using the Ah-counting method. This is rendered reliable as the current sensor is very efficient, the initial value of SoC is known, and the actual capacity can be calculated by integrating the current that flows through the battery during the experiment.

During the *learning phase*, one looks for the model parameters $\mathbf{w}$ in order to have function $f(r, \mathbf{w})$ fitting the output z, given the inputs r. This problem is generally referred to as the *regression problem*. In other words, the goal is to attain a quality criterion while estimating the parameters based on the training dataset. In the case of the regression problem, this often means minimizing the residual Lp-norm:

$$\min_{\mathbf{w}} \left(\sum_{i=1}^{N} |f(r_i, \mathbf{w}) - z_i|^p \right)^{1/p}. \tag{2.11}$$

In the most cases, the $L2$-norm is used. It is the sum of squares of the residuals.

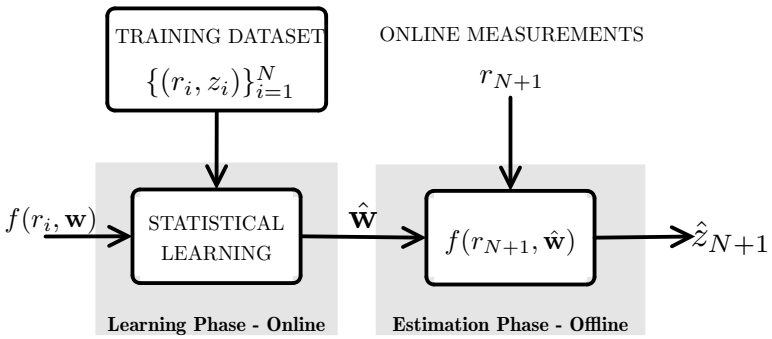

FIGURE 2.5: Estimation of *SoC* based on machine learning methods

In the *estimation phase*, a new measured input vector r_t is considered, the output $\widehat{z}_t$ is thus estimated using the model provided by the learning phase.

The intent of this technique is not to find an exact model but an approximation of it. Therefore, this approximated model is not always physically interpretable. The main virtue of the machine learning technique is its capacity to be easily extrapolated in different situations as soon as a corresponding training dataset is available.

In the following, we present several battery *SoC* estimators based on machine learning methods: the autoregressive moving average, the support vector regression and the artificial neural network. The elements of the *SoC* estimation methods presented in this section are listed in Table 2.4 according to the novel classification (see Figure 2.1).

Input variables	Online measured
SoC estimation model	Statistical regression
Other models	-
Type of processing	Open-loop

TABLE 2.4: Main classification criteria for *SoC* estimator based on machine learning methods

2.3.5.2 Autoregressive moving average (ARMA)

In a statistical analysis of time series, ARMA models provide a description of a stationary stochastic process. The model consists of two parts: an autoregressive (AR) part and a moving average (MA) part. Kozlowski [2003] uses these models to estimate the *SoC*:

$$SoC_t = w_1 \cdot r_t + w_2 \cdot r_{t-1} + w_3 \cdot SoC_{t-1}, \tag{2.12}$$

where r_t is the input vector including the real and imaginary components of the internal impedance, identified online based on a pattern recognition of the current and voltage signals. Here, the model parameters w_1, w_2 and w_3 are estimated in the learning phase using the least squares method. Multiple training datasets were collected during the charge/ discharge experiments, under various temperatures and current profiles. Four

types of batteries are considered: primary lithium and alkaline, and nickel-cadmium and lead-acid secondary batteries. Thus, there are 300 datasets across the different types and sizes of batteries. The model parameters are then identified separately for each usage conditions and type of battery. Accordingly, prior to an online estimation of the SoC, suitable model amongst the 300 models must be chosen based on the usage conditions, the type and the size of the battery.

Unfortunately, the least squares approach is strongly dependent on the training dataset. The learned model fits very well the training dataset but is not generic enough to fit new data that bears slight difference compared to training data. This problem is referred to as *overfitting*. To avoid it there are numerous regularization techniques enabling smooth models (see Arlot and Celisse [2010] for more information). The regularization can be usually achieved by:

- adding a penalty term to the cost function (2.11) like in SVR (Vapnik [1995]), RVM (Tipping [2001]) and LASSO (Hastie et al. [2001]),
- implementing a cross-validation techniques.

The former technique gives a smooth model (*i.e.*, some of the parameters w_i are equal to zero). The latter technique provides an unbiased estimator of the model parameters and is generally coupled with the first technique. Both techniques remain limited when there is a significant difference between the training dataset and the real-time data. For a better understanding of the cross-validation techniques, we describe here the K-fold technique which is one of the most widely used ones. It consists in splitting the training dataset into K subsets. At the i-th iteration, two sets are considered: a *learning set*, involving all subsets except the i-th one used to learn the model parameters, and a *testing set*, also called *validation set*, including the i-th subset and used to test the predictive performance of the model. Hence, the best model of the K learned models is retained. Figure 2.6 shows a detailed flow diagram of the K-fold cross-validation. The main drawback of this technique is that it has a high computation cost as K models need to be learned.

2.3.5.3 Support vector regression (SVR)

The SVR algorithm estimates the parameters of a nonlinear regression model which describes the system behavior as follows:

$$z_t = \sum_{i=1}^{N} w_i \cdot \kappa(r_t, r_i), \tag{2.13}$$

where N is the size of the training dataset, w_i the model parameters estimated by SVR and $\kappa(\cdot, \cdot)$ the kernel function. Thus, the regression model is based on a kernel function

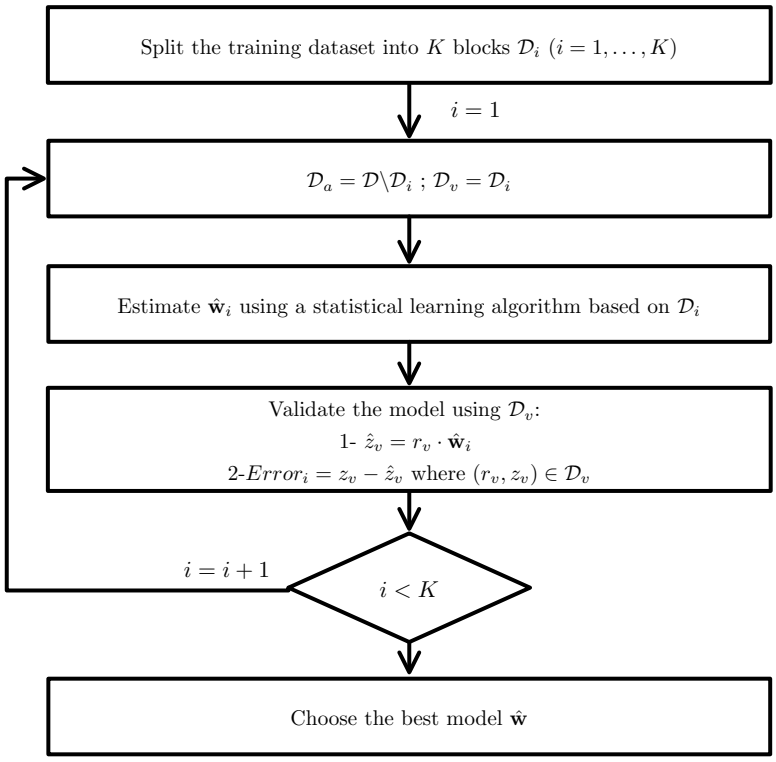

FIGURE 2.6: Principle of K-fold cross-validation

that measures the similarity between the new sample r_t and the training sample r_i. The SVR algorithm rules out some of the w_i as it adds a penalty constraint on the model parameters. Accordingly, a significant number of w_i are equal to zero. The remaining w_i corresponds to specific samples from the training dataset called *support vectors*. Only these vectors are then used in equation (2.13) to describe the behavior of the modeled system (see Vapnik [1995] for more information).

The SVR model is used to estimate the SoC based on a polynomial kernel as in Hansen and Wang [2005] and on an exponential kernel as in Álvarez Antóna et al. [2013]. Hansen and Wang [2005] use the current and the voltage as input vector r_t, whereas Álvarez Antóna et al. [2013] add the battery temperature to r_t.

In both studies, the training dataset contains only experimental tests under a constant current rate and a single ambient temperature. Accordingly, the estimated model is valid under similar usage conditions and it cannot be efficiently generalized to cases not seen in the learning phase. In addition, the SVR model is unsuitable for time series processing since it does not take into account the time dimension.

2.3.5.4 Artificial neural network (ANN)

In machine learning, the ANN is used to describe complex processes by using a nonlinear regression model. An ANN consists of a sequence of layers connected through connection

weights w_i. In most cases, a three layers system is used: the input layer which includes the input r_t, the hidden layer which describes the nonlinearity of the process via M intermediary functions $g_i(r_t)$ $(1 \leq i \leq M)$, and the output layer providing the estimated output z_t via an activation function $K(\cdot)$:

$$z_t = K\left(\sum_{i=1}^{M} w_i g_i(r_t)\right). \tag{2.14}$$

Based on a training dataset, the connection weights w_i and the parameters of $g_i(\cdot)$ are estimated in the learning phase using such methods as the least squares method or back propagation algorithm (see for instance Haykin [1998]).

The ANN is widely used to create a SoC model. Hereafter, several ways of using the ANN in the battery domain are presented. Charkhgard and Farrokhi [2010] model the battery voltage via an ANN as follows:

$$V_t = \sum_{i=1}^{M} w_i \exp\left(-\frac{\|r_t - t_i\|}{\sigma_i^2}\right)^2 + w_0, \tag{2.15}$$

where t_i and σ_i are the parameters of $g_i(\cdot)$ and $r_t = \{U_{t-1},\, I_t,\, SoC_t\}$. This mathematical representation of the battery voltage is then used as an observation equation of a state-space model. When using an EKF to estimate the state of interest SoC, this model raises computational problems to calculate $\partial V_t / \partial SoC_t$, as it is a sum of M strongly nonlinear functions.

Shen [2010] develops a more complex model using two successive ANNs. The first one provides a prediction of the battery voltage. The inputs of this ANN are defined based on the combined voltage model of Plett [2004] (i.e., the temperature T_t, the current I_t and the SoC_t are used as inputs of the first ANN). The second ANN is used to update the parameters of a PID (for proportional, integrated and derivative) controller that estimates the SoC as follows:

$$SoC_t = K_p \cdot (e_t - e_{t-1}) + K_I \cdot e_t + K_D \cdot (e_t - 2 \cdot e_{t-1} + e_{t-2}) + SoC_{t-1}, \tag{2.16}$$

where K_P, K_I, K_D are respectively the proportional, integral and derivative gain of the controller, and e_t is the gap between the predicted and measured voltage. In this case, the PID controller is called *neuro-controller* as K_p, K_I and K_D are calculated using an ANN. Accordingly, the SoC model is adapted online when the usage conditions change and provides an accurate estimation of the SoC. Although the model can be generalized more efficiently than the previously presented models, it remains limited when the usage conditions are very different from the training dataset or when the state of health of the battery decreases. In these cases, the model parameters of both ANNs must be updated. However, the use of two successive ANNs increases the complexity of the model and the embedded model requires a lot of the BMS computational resources.

The choice of the input variables directly affects the quality of the ANN. Thus, Bo et al. [2008] use partial least squares regression (PLSR) to select the input variables. Affanni et al. [2003] use the internal impedance, the extracted charge and the OCV as inputs of the ANN. However, these parameters cannot be directly measured; they are estimated using their physical definition. This induces an important error on the input values and affects the performance of the ANN.

2.3.5.5 Difficulties in choosing the training dataset

As shown above, in machine learning, the estimated model depends strongly on the training dataset, the learning method and the quality criterion. Thus, in order to obtain a generic model that can provide a reliable SoC estimation for any battery type under any usage conditions, as many possible situations must be included in an exhaustive training dataset. Such a dataset requires a large number of experimental tests, since several types of battery behavior can be identified according to the internal characteristics and the external usage conditions. Indeed, the battery behavior varies following the charge/discharge, the SoC ranges (generally decomposed into $[0\%, 20\%]$, $[20\%, 80\%]$, $[80\%, 100\%]$), the battery technologies, the types of current profile (constant, dynamic, etc.), the state of health, the ambient temperature and so on. Thus, machine learning methods are able to provide a generic SoC estimator for all behaviors existing in this training dataset. Another approach consists in associating a battery model to each battery behavior as in Kozlowski [2003] or, for example, a model for each voltage rate as in Kim et al. [2011]. The problem of the latter technique is the need for a large storage space that is often limited in real-time contexts.

In conclusion, the machine learning methods provide powerful techniques to estimate the parameters of a SoC model but they are not well explored. A penalty on the model parameters and a cross validation can provide a simple and smooth model. The main drawback of these methods is the need of an exhaustive training dataset including all possible usage conditions and battery types. Collecting such a dataset is difficult, even impossible, to carry out. More advanced techniques, like online learning, can detect the variations in the battery dynamics and adapt the model to new situations. Accordingly, these techniques can provide an "ideal" SoC model, reliable for all battery types and under all usage conditions. In addition, it is important to note that having a reliable SoC in the training dataset is sometimes difficult; particularly in real-time applications like electric vehicles.

2.4 Discussion

The battery being a complex electrochemical system, it is very difficult to provide an exact SoC model. Moreover, the electrochemical phenomena of each type of battery are very specific and their description requires the knowledge of the low-level battery parameters which is not possible in some contexts.

Thus, in order to create an approximate and generalized SoC model, an infinite number of experimental tests must be conducted to estimate its parameters.

Therefore all existing estimation models and methods are not generic and have some limitations. Several researchers develop SoC estimators for a specific battery technology, for instance, the method based on electrolyte density for acid batteries and the one based on modeling the migration of materials for Li-ion batteries. In addition, many developed models do not take into account the influence of the usage conditions on the battery behavior. For example, the SoC model is learned for a fixed temperature and current rate as in Hansen and Wang [2005], and for a fixed state of health as in Hu et al. [2010]. Several researchers improve the SoC model by incorporating the influence of ambient temperature as in Plett [2004] and the variations of the battery characteristics according to the SoC as in Hirai et al. [2008]. More advanced models can be adapted online as in Pang et al. [2001] and Shen [2010], but they require a lot of BMS resources as the model parameters are updated at each time step. All of these methods remain contestable when the usage conditions are very different from those of the training data. A summary of all the existing methods and their types, based on the novel classification, is presented in Table 2.5.

Accordingly, a complete and generic model that provides a reliable SoC estimation for any battery technology under any usage conditions, is needed. The characteristics of this "ideal" model depend on the goal and the context of the battery usage, and even on the specifications of the system. First, an "ideal" model must be efficient specifically when the SoC is near to 0% and 100% to avoid over-charge/discharge that cause permanent damage and fire risks like for Li-ion batteries. Also, it must give a linear SoC indication avoiding brutal change in order to give confidence on autonomy prediction. Second, an "ideal" model must be robust to the sensor noise, specifically in commercial products, such as an electric vehicle. Indeed, an efficient current sensor is very expensive. However in a laboratory, sensors are generally very reliable and there is no need to complicate the model with sensor noise treatment. Third, this model has to estimate the SoC in a real-time application. In this case, the battery is often partially charged and reference points that provide a SoC calibration are seldom available. In addition, the estimation of the SoC of batteries pack, usually used in real-life contexts, introduces more complexity due to the heterogeneity of the elementary cells constituting the pack (see for instance Zheng et al. [2013]). The display rate time of the SoC is also an important factor in real-time applications. In cell phones and laptops, the variation of the SoC is slower than the one in electric vehicles. Accordingly, the estimation algorithm must adapt more quickly in applications where the SoC changes rapidly. Fourth, a complete model must take into account the hardware limitations of the Battery Management System (BMS). Finally, this model has to incorporate the variation of the battery characteristics like the state of health, the impedance and the remaining capacity, as well as the influence of the usage conditions on the battery dynamics. It must be updated online to keep track of the

variation of the battery properties. It must also be able to indicate when it becomes inapplicable and when the learning of a new model is necessary.

In light of the above, machine learning methods are able to acquire the behavior of the battery through real data, without the absolute need to have a physical description of the battery. They can be used to identify the parameters of either predefined physical or statistical regression models. The dataset collected during an experimental test like the current, the temperature and the voltage, is called training data. Thus, the learned model provides an accurate SoC estimation for an identical battery technology with experimental conditions close to those of its corresponding training data. Therefore, in order to provide a valid model for all battery technologies under any usage conditions, an exhaustive training data that includes a large variety of situations is necessary. However, the behavior of the battery changes and the usage conditions are uncontrolled and so, no single model is valid at all times. Hence an online update of the SoC model is crucial. Updating a model online faces two constraints: the first one being hardware limitations, and the second being the absence of a reliable SoC values. Once these problems have been solved, the machine learning methods can provide an "ideal" SoC estimator.

2.5 Conclusion

In this chapter, we have introduced a novel classification of the existing SoC estimation methods. The main benefit of this classification is that it enables the identification of the techniques that can be improved to potentially reach an "ideal" SoC model. Such a model can estimate the SoC efficiently in a real-time context without being impacted by the battery chemistry or the usage conditions. We found that the methods based on directly measured physical quantities (presented in Section 2.3.2) cannot be generalized in a real-time context, as these quantities can only be measured offline. Also, the methods based on an open-loop processing do not take into account the sensor noises which can greatly decrease the reliability of the estimation, particularly in mainstream commercial products. Moreover, the methods based on closed-loop processing, like the Kalman filter and the controller, are promising candidates on the way to obtain an "ideal" SoC estimator. The main difficulty of these methods resides in the parameters identification. Clearly, the machine learning methods provide powerful techniques allowing the estimation of these parameters. However, these methods need an exhaustive training dataset to produce a generalized and smooth model. Two possible solutions can be adopted. The first one consists in taking into account the variability of the battery behavior during the conception of the SoC model. The second one consists in updating the model online when the characteristics of the battery or the usage conditions change. In this study, the former solution is studied, whereas the latter is discussed in the perspectives section.

SoC model	**Input variables and parameters**					**Processing type**		References
	Measured online	Measured offline	Estimated using physical model	Estimated using electrochemical model	Estimated using statistical model	Open loop	Closed loop	
Lookup table	✓	✓				✓		Huet [1998], Peled et al. [1988], Salkind et al. [1999]
	✓					✓		Piller et al. [2001]
	✓		✓				✓	Hu et al. [2010]
Physical model	✓		✓				✓	Chen et al. [2012], Li et al. [2013]
	✓		✓			✓		Alzieu et al. [1997], Kutluay et al. [2005], Ng et al. [2009]
	✓		✓		✓	✓		Gaddam et al. [2000], Malkhandi [2006]
	✓		✓		✓		✓	Avvari et al. [2015], Charkhgard and Farrokhi [2010], He et al. [2011], Kim et al. [2011], Plett [2004], Wang et al. [2007]
	✓		✓				✓	Codeca et al. [2008], Dai et al. [2006], Feng and Sun [2008], Han et al. [2009], Lee et al. [2008], Liao et al. [2011]
	✓		✓	✓		✓		Pop et al. [2006]
	✓		✓	✓			✓	Di Domenico et al. [2008], Santhanagopalan and White [2006]
Statistical model	✓		✓		✓		✓	Pang et al. [2001]
	✓		✓		✓	✓		Hirai et al. [2008], Moo et al. [2007], Tang et al. [2008]
	✓				✓	✓		Hansen and Wang [2005], Kozlowski [2003], Álvarez Antóna et al. [2013]
	✓				✓		✓	Affanni et al. [2003], Bo et al. [2008], He et al. [2014], Shen [2010]

TABLE 2.5: Summary of the existing *SoC* models and their types based on the novel classification

Chapter 3

Design of a battery State of Charge estimator using Switching Markov State-Space Models

Dans ce chapitre, nous développons un modèle de SoC tenant compte de la variabilité de la dynamique de la batterie selon ses caractéristiques internes et ses conditions externes de charge et de décharge souvent non-contrôlables.

L'évolution du SoC est principalement décrite par l'équation coulométrique :

$$SoC_t = SoC_{t-1} + \frac{\eta I_t \Delta t}{C_{\text{réelle}}}, \tag{3.1}$$

où η est la rendement faradique, I_t est le courant traversant la batterie, Δt le pas d'échantillonnage et $C_{\text{réelle}}$ la capacité réelle de la batterie. Pour faire face aux nombreuses limitations de cette équation présentées dans le Chapitre 2, nous considérons également une relation linéaire reliant la tension de la batterie à son SoC. Cette relation résulte de la modélisation de la batterie par un circuit électrique équivalent appelé *modèle à résistance interne* (cf. Figure 3.1). Ainsi, le SoC est décrit par un modèle à espaces d'états linéaires et gaussiens (*Linear Gaussian State-Space Model - LGSSM*) : l'équation de transition correspond à l'équation coulométrique et celle d'observation à la relation entre la tension et le SoC. Pour ce LGSSM, le courant traversant la batterie représente l'entrée, la tension aux bornes de la batterie représente l'observation et le SoC est l'état continu à estimer. À un instant t donné, le SoC peut être récursivement estimé par le filtrage de Kalman (Kalman [1963]) à partir du SoC à l'instant $t-1$ et de nouvelles mesures d'observation et d'entrée.

Cependant, le comportement de la batterie n'est pas linéaire et un seul modèle linéaire ne suffit pas pour décrire ses différents régimes de fonctionnement (cf. Figures 3.2 et 3.3). Nous considérons ainsi que chacun de ces régimes est décrit par un LGSSM et que la

transition entre ces différents LGSSMs est aléatoire et décrite par une chaîne de Markov cachée. Cette modélisation induit alors, à un instant donnée, trois variables aléatoires :

1. l'état discret caché de la chaîne de Markov, noté S_t, indexant le LGSSM actif à l'instant t. $(S_t)_{t\geq 0}$ est un processus de Markov d'ordre 1 sur $\{1,\dots,\kappa\}$ de vecteur de probabilité initial Π et de matrice de transition P ;
2. l'état continu inconnu, noté X_t, décrivant le SoC à estimer. Ici, $X_t \in [0, 100]$;
3. l'observation, notée Y_t, correspondant à la tension de la batterie.

Nous proposons un nouvel modèle de SoC basé sur les modèles à espaces d'états linéaires et gaussiens gouvernés par une chaîne de Markov, que nous appellerons plus simplement un modèle à espaces d'états gouverné par une chaîne de Markov (*Switching Markov State-Space Model - SMSSM*) (cf. Figure 3.4). L'évolution jointe de ces trois variables aléatoires est représentée par :

$$p_\Theta(s_t \,|s_{0:t-1}) = p_\Theta(s_t|s_{t-1}) = P(s_t,\, s_{t-1}), \tag{3.2}$$

$$X_t = X_{t-1} + B(S_t)u_t\Delta t + W_t, \tag{3.3}$$

$$Y_t = C(S_t)X_t + D(S_t)u_t + D_0(S_t) + \Sigma_t, \tag{3.4}$$

avec $p(w_t|s_t) = \mathcal{N}(w_t;\, 0, \sigma_X^2(s_t))$, $p(\varepsilon_t|s_t) = \mathcal{N}(\varepsilon_t;\, 0, \sigma_Y^2(s_t))$, $\{W_t\}_{t\geq 0}$ et $\{\Sigma_t\}_{t\geq 0}$ sont deux suites iid et W_i et Σ_j sont indépendants quels que soient (i, j).

L'équation (3.2) décrit l'évolution markovienne de $S_t \in \{1,\dots,\kappa\}$. L'équation (3.3), représentant l'équation de transition, résulte de l'équation coulométrique où u_t est le courant, Δt le pas d'échantillonnage et W_t le bruit du processus. L'équation (3.4), représentant l'équation d'observation, décrit l'influence de X_t sur l'observation Y_t où Σ_t modélise le bruit du capteur. Nous supposons que la loi de X_0, et le vecteur de probabilité initiale de S_0, noté Π, sont connus. L'ensemble de paramètres inconnus du modèle est noté Θ :

$$\Theta = \{P,\, \Gamma\}, \tag{3.5}$$

avec Γ l'ensemble de paramètres des deux équations de transition et d'observation :

$$\Gamma = \{B(s),\, \sigma_X(s),\, C(s),\, D(s),\, D_0(s),\, \sigma_Y(s)\}_{s=1,\dots,\kappa}. \tag{3.6}$$

Pour un SMSSM, la densité de probabilité jointe de la séquence d'observations $Y_{0:T}$, d'états continus $X_{0:T}$ et d'états discrets $S_{0:T}$ peut être décrite comme suit (Cappé et al. [2005]) :

$$\begin{aligned} p_\Theta(y_{0:T}, x_{0:T}, s_{0:T}) =& p_\Theta(y_0 \mid x_0, s_0) \cdot p_\Theta(x_0 \mid s_0) \cdot p_\Theta(s_0) \\ & \cdot \prod_{t=1}^{T} p_\Theta(y_t \mid s_t, x_t) \cdot p_\Theta(x_t \mid s_t, x_{t-1}) \cdot p_\Theta(s_t \mid s_{t-1}). \end{aligned} \tag{3.7}$$

Le modèle proposé ainsi que les hypothèses posées sont décrits d'une façon détaillée dans la Section 3.3.

Pour compléter la description du modèle, nous vérifions l'identifiabilité de l'ensemble de ses paramètres inconnus. Un modèle est dit identifiable lorsque la relation entre l'ensemble de paramètres Θ et la vraisemblance est bijective :

$$\Theta \neq \Theta^* \Leftrightarrow p_\Theta(y_{0:T}) \neq p_{\Theta^*}(y_{0:T}). \tag{3.8}$$

Si nous considérons une transformation linéaire de X_t telle que $X_t^* = k \cdot X_t$, nous remarquons que le SMSSM n'est pas nécessairement identifiable. En effet, les paramètres du modèle peuvent être ajustés de façon à ce que la loi des observations reste invariable :

$$X_t^* = X_{t-1}^* + k \cdot B(S_t)u_t\Delta t + k \cdot W_t, \tag{3.9}$$

$$Y_t = \frac{C(S_t)}{k}X_t^* + D(S_t)u_t + D_0(S_t) + \Sigma_t. \tag{3.10}$$

Dans la Section 3.4 de ce chapitre, nous démontrons que lorsque la chaîne de Markov est irréductible et apériodique, une information *a priori* reliant les observations et l'état continu inconnu à un instant t_0 suffit pour assurer l'identifiabilité de l'ensemble de paramètres du modèle. Cette information *a priori* s'écrit de la manière suivante :

$$y_{t_0} = C(s)x_{t_0} + D(s)u_{t_0} + D_0(s), \tag{3.11}$$

$\forall s = 1, \ldots, \kappa$, où y_{t_0} et x_{t_0} sont connus.

Dans le cadre de l'estimation du SoC, ces deux contraintes sont naturellement vérifiées. En effet, $\{S_t\}$ indexant le régime de fonctionnement de la batterie est irréductible et apériodique du fait que la charge et la décharge sont réversibles et que la variation du régime est aléatoire selon les conditions d'usage externes. De plus, dans la plupart des cas, la batterie est initialement au repos (i.e., $u_0 = 0$). De ce fait, son SoC peut être calculé d'une manière efficace à partir de la mesure de sa tension à vide, donnant ainsi la relation suivante :

$$y_0 = C(s)x_0 + D_0(s), \ \forall s = 1, \ldots, \kappa. \tag{3.12}$$

où y_0 est la mesure de la tension à vide et x_0 est le SoC correspondant.

La Section 3.5 est dédié à l'estimation en ligne du SoC dans le modèle estimé. La problématique revient à estimer X_t de manière récursive à partir d'une estimation de X_{t-1} et des nouvelles mesures y_t et u_t en considérant que l'ensemble de paramètres Θ est connu. Cependant, une estimation optimale de X_t, au sens de l'erreur quadratique moyenne, est un problème NP-complet (Tugnait [1982]). En effet, cette estimation nécessite d'utiliser un filtre de Kalman pour toutes les séquences d'états discrets possibles, soit κ^{t+1} séquences. Une solution basée sur une méthode de Monte-Carlo consisterait à tirer aléatoirement N séquences $\{s_{0:t}^i\}_{i=1}^N$, appelées *particules*, selon la loi

$p_\Theta(s_{0:t}|y_{0:t})$. Pour chaque séquence $s^i_{0:t}$, un filtre de Kalman est utilisé pour calculer

$$\mathbb{E}_{y_{0:t},s^i_{0:t}}[X_t] \triangleq x^i_t.$$

Dans ce cas, un estimateur de X_t est donné par la moyenne empirique de $\{x^i_t\}_{i=1}^N$. Cependant, le tirage selon $p_\Theta(s_{0:t}|y_{0:t})$ étant difficile (Doucet et al. [2001b]), nous utilisons une technique d'*échantillonnage d'importance séquentiel*. Les séquences $\{s^i_{0:t}\}_{i=1}^N$ sont ici tirées selon une loi d'importance, notée $q_\Theta(s_{0:t}|y_{0:t})$, s'écrivant de la manière suivante :

$$q_\Theta(s_{0:t}|y_{0:t}) = q_\Theta(s_{0:t-1}|y_{0:t-1})q_\Theta(s_t|s_{0:t-1}, y_{0:t}). \tag{3.13}$$

de manière à simuler d'une façon séquentielle et à coût fixe à chaque instant. Ainsi, à l'instant $t-1$, nous disposons de N particules $\{s^i_{0:t-1}\}_{i=1}^N$ et de leur poids d'importance $\{w^i_{t-1}\}_{i=1}^N$ obtenus à partir de l'échantillonnage d'importance séquentiel et faisant intervenir la loi d'importance $q_\Theta(s_{0:t-1}|y_{0:t-1})$. À l'instant t, les séquences et leurs poids $\{s^i_{0:t}, w^i_t\}_{i=1}^N$ sont obtenus à partir de la nouvelle observation y_t et des séquences et poids précédents $\{s^i_{0:t-1}, w^i_{t-1}\}_{i=1}^N$ en tirant $s^i_t \sim q_\Theta(s_t|s^i_{0:t-1}, y_{0:t})$, $1 \leq i \leq N$. Dans cette étude, nous choisissons $q_\Theta(s_t \mid s^i_{0:t-1}, y_{0:t}) = p_\Theta(s_t \mid s^i_{0:t-1}, y_{0:t})$ de telle manière à minimiser la variance conditionnelle des poids $\{w^i_t\}_{i=1}^N$ (Doucet et al. [2001b]). L'algorithme proposé pour estimer x_t en ligne est présenté dans l'algorithme 1.

Afin de valider cette méthode d'estimation en ligne et de calibrer le nombre de particules, nous avons généré une base de données de taille $T = 500$ à partir d'un SMSSM à $\kappa = 3$ états cachés (cf. Figure 3.5). Les paramètres de ce modèle sont choisis de façon à ce que la base de données soit proche des tests d'usage d'une batterie électrique (cf. Table 3.1). La séquence d'états continus $X_{0:T}$ est estimée pour différentes valeurs de N. Les résultats montrent que lorsque N est bien choisi ($N \simeq 10$, pour $\kappa = 3$), l'erreur quadratique moyenne entre la vraie séquence et celle estimée est relativement faible (cf. Figures 3.6 et 3.7).

En conclusion, nous avons proposé un nouveau modèle de *SoC* basé sur les SMSSMs afin de tenir compte de la variabilité du comportement de la batterie lors de la phase de conception du modèle de *SoC*. Selon ce nouveau modèle, le *SoC* est estimé à travers un ensemble de modèles à espaces d'états linéaires indexés par une chaîne de Markov reflétant le régime de fonctionnement de la batterie. Néanmoins, les paramètres de ce modèle ne sont pas nécessairement identifiables. Pour assurer leur identifiabilité, nous avons imposé des contraintes simples et physiquement vérifiables. Nous avons également montré que l'estimation récursive de X_t à partir d'une estimation de X_{t-1}, une nouvelle observation y_t et entrée u_t peut être effectuée par une technique d'échantillonnage d'importance séquentiel. Le nombre de particules nécessaire pour assurer une estimation précise semble être relativement faible ($N \simeq 10$, pour $\kappa = 3$), ce qui rend cette technique adaptée à un calculateur embarqué.

Contents

The aim of this chapter is to develop a *SoC* model taking into account the random change of battery dynamics during its charge/discharge. Indeed, the inherent behavior of the battery depends on its internal characteristics like its resistance and its state of health, as well as on the usage conditions like the ambient temperature and the current profile.

As a first step, we recall the benefits and the limitations of the Ah-counting model to provide a *SoC* indication, and show the importance of the battery voltage to improve this leading model. Hence, a Linear Gaussian State-Space Model (LGSSM) is established: the Ah-counting models the transition equation, whereas the observation equation is based on an equivalent-electric circuit relating the voltage to the *SoC*. For real data, we show that a single LGSSM cannot be reliable throughout the usage of the battery, and different potential LGSSMs should be used instead.

Hence, we propose a new *SoC* model using Switching Markov State-Space Models (SMSSM): the battery dynamics are described by a set of LGSSMs switching randomly according to a Markov chain. This new model includes two latent states: a continuous one, the *SoC* and a discrete one, the finite state of the Markov chain.

Once the model is developed and the used assumptions are fixed, its identifiability must be verified to guarantee its relevance. For SMSSMs, we prove that the identifiability can be achieved by imposing straightforward and natural constraints on the model parameters.

We then show how to estimate online the state of interest efficiently for fixed model parameters. Since an optimal estimation is at a prohibitive computational cost, a sub-optimal algorithm based on Monte Carlo (MC) methods is developed. The performance of this algorithm in estimating the state of interest and the influence of the number of particles are studied based on simulated data.

3.1 Coulomb Counting Model

The leading method for estimating the SoC is based on the Coulomb counting equation:

$$SoC_t = SoC_0 + \int_0^t \frac{\eta \cdot I_s}{C_{\text{actual}}} ds, \tag{3.14}$$

where η is the Faraday efficiency, C_{actual} the actual capacity and I_s the algebraic current measurement: positive for a charge and negative for a discharge.

A numerical implementation of this equation requires a temporal discretization. The SoC is then calculated as follows:

$$SoC_t = SoC_{t-\Delta t} + \frac{\eta I_t \Delta t}{C_{\text{actual}}}, \tag{3.15}$$

where Δt is the sampling time.

As discussed in Section 2.3.1, this method offers a generic SoC model, suitable for all battery types and used as a reference method for the evaluation of the performance of any other estimation method. Indeed given the actual capacity, the SoC estimation requires only a measure of the current. This minimizes the volume of information exchanged between the battery and the battery management system and reduces the computation time.

Therefore, this method holds a number of limitations, such as the calculation of η, the current leakages in rest period and the variation of the battery capacity. In addition, this method suffers from error accumulation over time which may introduce a bias to the estimated SoC. The limitations and the proposed improvements of this model are thoroughly detailed in Section 2.3.1.

In this study, the voltage of the battery is considered in order to improve the Ah-counting model. Indeed, an accurate voltage sensor is not costly contrary to a current one. Accordingly, a voltage model depending on the SoC completes the Coulomb counting in order to establish the so-called Linear Gaussian State-Space Model (LGSSM).

3.2 Linear Gaussian State-Space Model

In the following, the joint evolution of the SoC_t and the observed voltage V_t is modeled using a state-space framework (Anderson and Moore [1979]). Let us denote X_t and Y_t the SoC and the voltage at time t respectively. In this study, we use the standard convention whereby capital letters denote random variables, and lower letters their corresponding realizations.

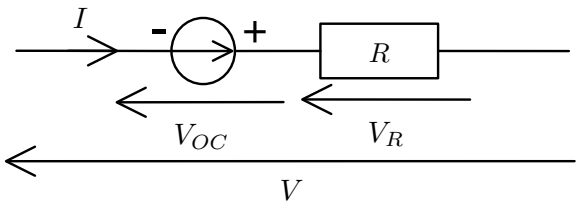

FIGURE 3.1: Battery equivalent circuit diagram: Internal Resistance model

3.2.1 Observation equation

To describe the relation between the voltage and the SoC, we use the Internal Resistance (IR) model which is the only linear model linking the battery voltage and its SoC (Plett [2004]). As shown in Figure 3.1, this model implements an ideal voltage source V_{OC} representing the Open Circuit Voltage (OCV) of the battery and an ohmic resistance R describing its internal resistance, with

$$V_{R_t} = R \cdot I_t, \tag{3.16}$$

$$V_{OC_t} = C \cdot \mathrm{SoC}_t + D_0, \tag{3.17}$$

where C and D_0 have to be estimated. The terminal voltage is then the sum of the resistance voltage V_R and the voltage source V_{OC}:

$$V_t = V_{R_t} + V_{OC_t}. \tag{3.18}$$

This voltage model gives the *observation equation*:

$$Y_t = C \cdot X_t + D \cdot u_t + D_0 + \Sigma_t, \tag{3.19}$$

where C, D and D_0 are constants with physical interpretation, $u_t = I_t$ is the exogenous input, and $\Sigma_t \sim \mathcal{N}(\varepsilon_t; 0, \sigma_Y^2)$ models the voltage sensor error with $\{\Sigma_t\}_{t>0}$ iid. Indeed, D corresponds to the internal resistance, C corresponds to the slope of the OCV-SoC curve, and D_0 is also related to the OCV-SoC relationship.

3.2.2 Transition equation

The description of the model is completed by the *transition equation* based on the Coulomb counting model:

$$X_t = X_{t-1} + B\, u_t\, \Delta t + W_t, \tag{3.20}$$

with $B = \frac{\eta}{C_{\text{actual}}}$, and $W_t \sim \mathcal{N}(w_t; 0, \sigma_X^2)$ modeling the random fluctuations of the SoC where $\{W_t\}_{t>0}$ is iid. The Gaussian white noises W_t and Σ_t are assumed to be independent: W_i and Σ_j are independent $\forall(i, j)$.

Thus, the LGSSM of the SoC relates the unobserved variable X_t and the observed variable Y_t through the linear equations (3.19) and (3.20).

The Kalman filter provides an optimal estimation of X_t, in a mean square error sense, given observations $y_{0:t} = \{y_0, \ldots, y_t\}$ and inputs $u_{0:t}$ up to time t (cf. Appendix A).

In practice at time $t = 0$, the battery is often in a resting state (i.e., the current is zero on a long duration $\simeq 30$ mins), hence x_0 can be efficiently calculated through the OCV measurement (see Section 2.3.2.1). Thus, the following physical constraint on the observation equation is available at $t_0 = 0$:

$$y_0 = C \cdot x_0 + D_0, \tag{3.21}$$

where y_0 is the OCV measurement and x_0 is the corresponding SoC.

However in a real context, each set of usage conditions should be described by specific observation and transition equations. Indeed, the battery dynamics change during the charge/discharge according to uncontrolled internal and external conditions. Let us consider the case of a driving of an electric vehicle at constant ambient temperature. In order to monitor the relevance of the Kalman filter, attention has been given to the comparison between the observed and estimated voltage. Figure 3.2 shows that a single LGSSM cannot estimate accurately the voltage throughout the whole time interval, and different potential LGSSMs should be used instead.

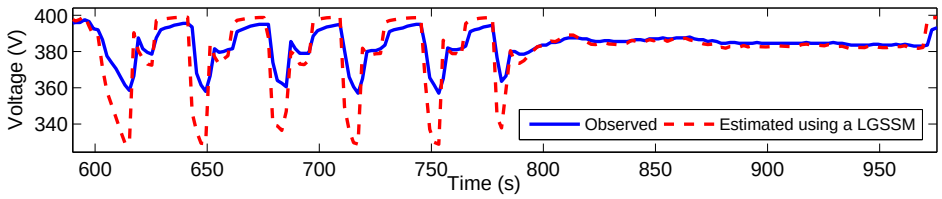

FIGURE 3.2: Estimation of the voltage using a LGSSM: Data from driving of an electric vehicle

3.3 Switching Markov State-Space Model

For real electric vehicle data, as shown in Figure 3.3, the relation between the voltage, the current and the SoC is non-linear. Therefore, the objective is to model these non-linearities by a linear-piecewise model. Accordingly, we suppose that the observation equation (3.19) is rather linear piecewise, and that the pair $\{X_t, Y_t\}$ is described by different potential LGSSMs. The number of these potential LGSSMs controls then the degree of nonlinearity of battery dynamics. The changes between these LGSSMs are supposed random according to an unobservable Markov chain, denoted $\{S_t\}$. The SoC is then modeled by a Switching Markov State-Space Model (SMSSM). This model has

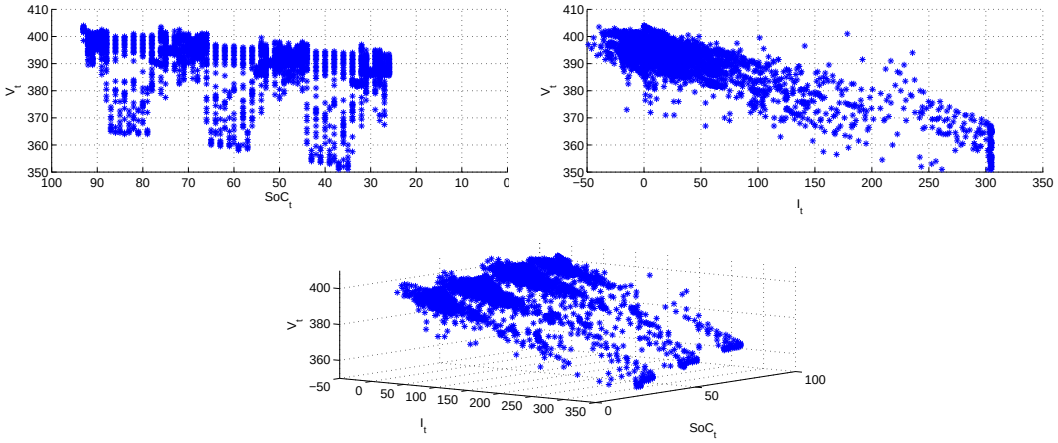

FIGURE 3.3: Voltage V_t *versus* SoC and current I_t: Data from driving of an electric vehicle

been well studied in Ackerson and Fu [1970], Akashi and Kumamoto [1977], Frühwirth-Schnatter [2006], Kim [1994], Tugnait [1982]. A SMSSM can be seen as a LGSSM with parameters indexed by a Markov chain. A graphical representation of a SMSSM is provided in Figure 3.4.

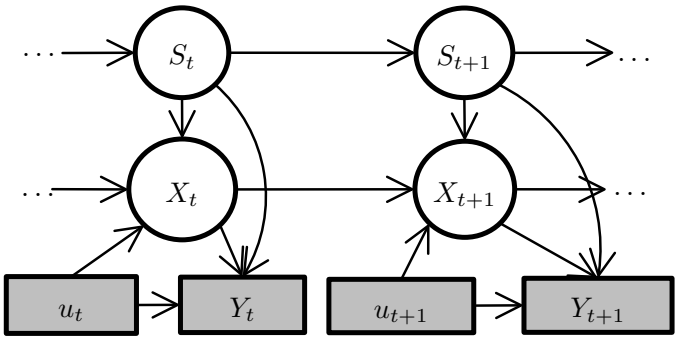

FIGURE 3.4: Graphical representation of a SMSSM, where $\{Y_t\}$ and $\{u_t\}$ are observed, and $\{X_t\}$ and $\{S_t\}$ are latent variables

3.3.1 Modeling

Let S_t denote a discrete irreducible and aperiodic Markov chain on $\{1, \ldots, \kappa\}$, with initial distribution Π and transition matrix P. The proposed SoC model is given by a natural extension of the LGSSM (3.19)-(3.20) with parameters indexed by S_t:

$$X_t = X_{t-1} + B(S_t)u_t\Delta t + W_t, \tag{3.22}$$

$$Y_t = C(S_t)X_t + D(S_t)u_t + D_0(S_t) + \Sigma_t, \tag{3.23}$$

for $t = 0, \ldots, T$, where $p(\varepsilon_t|s_t) = \mathcal{N}(\varepsilon_t; 0, \sigma_Y^2(s_t))$ and $p(w_t|s_t) = \mathcal{N}(w_t; 0, \sigma_X^2(s_t))$. In addition, it is assumed that conditionally on S_t, $\{W_t\}_{t>0}$ and $\{\Sigma_t\}_{t>0}$ are iid, and W_i

and Σ_j are independent $\forall(i,j)$. Let Θ denote the set of model parameters:

$$\Theta = \{P,\ \Gamma\}\,, \tag{3.24}$$

where Γ denote the set of parameters of the transition and observation equations:

$$\Gamma = \{B(s),\ \sigma_X(s),\ C(s),\ D(s),\ D_0(s),\ \sigma_Y(s))\}_{s=1,\dots,\kappa}\,. \tag{3.25}$$

It is assumed that the distribution of X_0 is Gaussian and known, the initial distribution Π is known, and the distributions $p_\Theta(x_t \mid x_{t-1}, s_t)$ and $p_\Theta(y_t \mid x_t, s_t)$ are Gaussian with parameters calculated respectively from (3.22) and (3.23):

$$p_\Theta(x_t \mid x_{t-1}, s_t) = \mathcal{N}(x_t; x_{t-1} + B(s_t)u_t\Delta t, \sigma_X^2(s_t)), \tag{3.26}$$

$$p_\Theta(y_t \mid x_t, s_t) = \mathcal{N}(y_t; C(s_t)x_t + D(s_t)u_t + D_0(s_t), \sigma_Y^2(s_t)). \tag{3.27}$$

3.3.2 Model properties

A SMSSM can be viewed as a hierarchical model. This point of view is very useful to describe its structure. On a first level, the model specifies the conditional distribution $p_\Theta(y_{0:T} \mid s_{0:T}, x_{0:T})$ of the observation process $Y_{0:T}$ given the whole state process $\{S_{0:T}, X_{0:T}\}$. On a second level, the model specifies the conditional distribution $p_\Theta(x_{0:T} \mid s_{0:T})$ of the states of interest $X_{0:T}$ given the whole Markov sequence $S_{0:T}$. On a third level, the model specifies the distribution $p_\Theta(s_{0:T})$ of the state of the Markov chain.

The following conditional independent relations arise from the model (3.22)-(3.23). The observations $Y_{0:T}$ are independent given $\{S_{0:T}, X_{0:T}\}$, and Y_t is independent of $\{(S_0, X_0), \dots, (S_{t-1}, X_{t-1}), (S_{t+1}, X_{t+1}), \dots, (S_T, X_T)\}$ given (S_t, X_t). This can be formulated as follows:

$$p_\Theta(y_{0:T} \mid s_{0:T}, x_{0:T}) = \prod_{t=0}^{T} p_\Theta(y_t \mid s_t, x_t). \tag{3.28}$$

Given $S_{0:t}$, the state of interest X_t is a first-order hidden Markov process, hence independent of $\{X_0, \dots, X_{t-2}, X_{t+1}, \dots, X_T\}$ given X_{t-1}. In addition, X_t is independent of $\{S_0, \dots, S_{t-1}, S_{t+1}, \dots, S_T\}$ given S_t. This can be formulated as follows:

$$p_\Theta(x_{0:T} \mid s_{0:T}) = p_\Theta(x_0 \mid s_0) \prod_{t=1}^{T} p_\Theta(x_t \mid s_t, x_{t-1}). \tag{3.29}$$

The Markov chain $\{S_t\}$ satisfies:

$$p_\Theta(s_{0:T}) = p_\Theta(s_0) \prod_{t=1}^{T} p_\Theta(s_t \mid s_{t-1}). \tag{3.30}$$

Thus, the augmented state $\{S_t, X_t\}$ is a Markov chain, satisfying in addition:

$$p_\Theta(s_t, x_t \mid s_{t-1}, x_{t-1}) = p_\Theta(s_t \mid s_{t-1}) p_\Theta(x_t \mid s_t, x_{t-1}). \tag{3.31}$$

Indeed, the SMSSM can also be viewed as a Hidden Markov Model (HMM) with two latent states, namely S_t and X_t.

Based on this hierarchical view, the complete-likelihood can be written as follows:

$$\begin{aligned} p_\Theta(y_{0:T}, x_{0:T}, s_{0:T}) =& p_\Theta(y_0 \mid x_0, s_0) \cdot p_\Theta(x_0 \mid s_0) \cdot p_\Theta(s_0) \\ & \cdot \prod_{t=1}^{T} p_\Theta(y_t \mid s_t, x_t) \cdot p_\Theta(x_t \mid s_t, x_{t-1}) \cdot p_\Theta(s_t \mid s_{t-1}). \end{aligned} \tag{3.32}$$

It is noteworthy that conditionally to $S_{0:t}$, the pair (X_t, Y_t) is a LGSSM. Accordingly, given a specific sequence of Markov states $s_{0:T}$, the likelihood $p_\Gamma(y_{0:T} \mid s_{0:T})$ is a $(T+1)$-Gaussian distribution with parameters recursively calculated as follows:

$$p_\Gamma(y_{0:T}|s_{0:T}) = p_\Gamma(y_0|s_0) \prod_{t=1}^{T} p_\Gamma(y_t \mid y_{0:t-1}, s_{0:t}), \tag{3.33}$$

where $p_\Gamma(y_t \mid y_{0:t-1}, s_{0:t}) = \mathcal{N}(y_t; y_{t|t-1}, \Omega_{t|t-1})$. The variables $y_{t|t-1}$ and $\Omega_{t|t-1}$ are respectively the expectation and the variance of the observation y_t given the discrete states $s_{0:t}$, and the observations $y_{0:t-1}$ up to time $t-1$. They are calculated from (3.23) as follows:

$$y_{t|t-1} = C(s_t) x_{t|t-1} + D(s_t) u_t + D_0(s_t), \tag{3.34}$$

$$\Omega_{t|t-1} = C(s_t)^2 \sigma^2_{t|t-1} + \sigma^2_Y(s_t), \tag{3.35}$$

where the variables $x_{t|t-1}$ and $\sigma^2_{t|t-1}$ are respectively the expectation and the variance of the continuous state X_t given the discrete states $s_{0:t}$, and the observations $y_{0:t-1}$ up to $t-1$. They are calculated from (3.22) during the prediction step of a Kalman filter:

$$x_{t|t-1} = x_{t-1|t-1} + B(s_t) u_t \Delta t, \tag{3.36}$$

$$\sigma^2_{t|t-1} = \sigma^2_{t-1|t-1} + \sigma^2_X(s_t). \tag{3.37}$$

The variables $x_{t-1|t-1}$ and $\sigma^2_{t-1|t-1}$ are respectively the expectation and the variance of the continuous state X_{t-1} given the discrete states $s_{0:t-1}$, and the observations $y_{0:t-1}$. They are calculated by the correction step of a Kalman filter:

$$x_{t-1|t-1} = x_{t-1|t-2} + K_{t-1}(y_{t-1} - y_{t-1|t-2}), \tag{3.38}$$

$$\sigma^2_{t-1|t-1} = (1 - K_{t-1} C(s_{t-1}))^2 \, \sigma^2_{t-1|t-2} + K^2_{t-1} \sigma^2_Y(s_{t-1}), \tag{3.39}$$

where K_{t-1} is the Kalman gain (Kalman [1963]) given by

$$K_{t-1} = \sigma^2_{t-1|t-2} C(s_{t-1}) \Omega^{-1}_{t-1|t-2}. \tag{3.40}$$

The prediction and correction steps of the Kalman filter are detailed in Appendix A.

In order to guarantee the relevance of the SMSSM, its identifiability must be ensured. In the next section, we show that the SMSSM (3.22)-(3.23) is non-identifiable, and we identify the constraints to be imposed in order to ensure its identifiability.

3.4 Identifiability of a SMSSM

The SMSSM, among many other mathematical models, is used to describe the dynamics of a given system using experimental data and to understand observed phenomena. Indeed, it is beneficial that each parameter of the model has a physical interpretation in order to easily integrate any prior knowledge and interpret the results of this modeling. This can be ensured when the model is *identifiable*; i.e., two different set of parameters Θ and Θ^* lead to two different models (i.e., the transformation which maps Θ to the likelihood $p_\Theta(y_{0:T})$ should be one-to-one). This can be formulated as follows:

$$p_\Theta(y_{0:T}) = p_{\Theta^*}(y_{0:T}) \quad \Rightarrow \quad \Theta = \Theta^*. \tag{3.41}$$

Moreover, the identifiability of a subset of parameters can be formulated as follows.

Definition 1

A subset of parameters $F \in \Theta$ is said globally structurally (g.s.) identifiable if

$$p_{\Theta^*}(y_{0:T}) = p_\Theta(y_{0:T}) \quad \Rightarrow \quad F^* = F. \tag{3.42}$$

However, it is well-known that a LGSSM (see for instance Walter and Lecourtier [1981]), and even more a SMSSM, suffers from identifiability problems. Indeed, let us consider that X^*_t and X_t are related by a linear transformation $X^*_t = H(s_t)X_t$. This transformation leads to an equivalent model with the same observations distribution:

$$\left\{ \begin{array}{rcccl} B^*(s) & = & H(s) & \cdot & B(s) \\ C^*(s) & = & C(s) & \cdot & H^{-1}(s) \\ {\sigma^2_X}^*(s) & = & H(s) & \cdot & \sigma^2_X(s) \cdot H(s), \end{array} \right. \tag{3.43}$$

where $H(s) \in \mathbb{R}^*$. It has to be noted that $D(s)$, $D_0(s)$ and $\sigma_Y(s)$ are always g.s. identifiable. They are invariant under any linear transformation $H(s)$. To ensure the identifiability of the model parameters, constraints could be imposed on these ones.

It is noteworthy that the Markov states S_t can be relabeled without changing the distribution of the observations $p_\Theta(y_{0:T})$ (Stephens [2000]). Thus, the identifiability of the SMSSM is considered up to a state switching.

3.4.1 Case of a Linear Gaussian State-Space Model

First of all, we address the identifiability of a LGSSM (i.e., SMSSM with $\kappa = 1$). The following prior information is considered at t_0:

$$y_{t_0} = Cx_{t_0} + Du_{t_0} + D_0, \tag{3.44}$$

with $x_{t_0} \neq 0$. Any equivalent model should have the same prior information at t_0:

$$y_{t_0} = C^* x_{t_0} + D^* u_{t_0} + D_0^*. \tag{3.45}$$

Based on the system of equations (3.43), this prior can be written as follows:

$$y_{t_0} = CH^{-1} x_{t_0} + Du_{t_0} + D_0. \tag{3.46}$$

Under (3.44) and (3.46), the only solution of the system (3.43) is $H = 1$. Thus, the parameters of a SMSSM with $\kappa = 1$ are g.s. identifiable under the constraint (3.44).

3.4.2 Case of a known sequence of Markov states

It is assumed that at t_0,

$$y_{t_0} = C(s_{t_0})x_{t_0} + D(s_{t_0})u_{t_0} + D_0(s_{t_0}), \tag{3.47}$$

for any hidden state $s_{t_0} = 1, \ldots, \kappa$. When the Markov sequence $s_{0:T}$ is known, the model can be transformed into linear piecewise models. This stems from the fact that the Markov chain $\{S_t\}$ is irreducible and aperiodic. Accordingly, the identifiability results of the previous section can be extended, and the set Γ of parameters of these LGSSMs is g.s. identifiable under constraints (3.47). This can be formulated by the following relation:

$$p_\Gamma(y_{0:T}|s_{0:T}) = p_{\Gamma^*}(y_{0:T}|s_{0:T}) \quad \Rightarrow \quad \Gamma = \Gamma^*. \tag{3.48}$$

3.4.3 Case of an unknown sequence of Markov states

When the Markov sequence $s_{0:T}$ is unknown, the marginal likelihood of the SMSSM (3.22)-(3.23) is given by

$$p_\Theta(y_{0:T}) = \sum_{s_{0:T} \in \mathcal{S}} p_P(s_{0:T}) \cdot p_\Gamma(y_{0:T}|s_{0:T}), \tag{3.49}$$

where $\mathcal{S} = \{1, \dots, \kappa\}^{T+1}$ is the set of all possible Markov sequences, and $p_\Gamma(y_{0:T}|s_{0:T})$ is a $(T+1)$-Gaussian distribution whose parameters are recursively calculated using a Kalman filter (as presented in Section 3.3.2). Thus, $p_\Theta(y_{0:T})$ is a finite convex combination of Gaussian distributions. Yakowitz and Spragins [1968] prove that a family of finite mixture distributions is identifiable, iff the members of the underlying distribution family are linearly independent over the field of real numbers. Based on this theorem, they prove that an m-dimensional Gaussian distribution ($m > 1$) generates identifiable finite mixtures. Let us consider Θ and Θ^* such as

$$\begin{gathered} p_\Theta(y_{0:T}) = p_{\Theta^*}(y_{0:T}) \\ \sum_{s_{0:T}\in\mathcal{S}} p_P(s_{0:T}) \cdot p_\Gamma(y_{0:T}|s_{0:T}) = \sum_{s_{0:T}\in\mathcal{S}} p_{P^*}(s_{0:T}) \cdot p_{\Gamma^*}(y_{0:T}|s_{0:T}). \end{gathered} \tag{3.50}$$

Following the theorem of Yakowitz and Spragins [1968], we have the following equations:

$$\begin{aligned} &\text{(i)} \;\; p_\Gamma(y_{0:T}|s_{0:T}) = p_{\Gamma^*}(y_{0:T}|s_{0:T}), \\ &\text{(ii)} \; p_P(s_{0:T}) = p_{P^*}(s_{0:T}). \end{aligned}$$

Under the constraints (3.47), the equation (i) implies that $\Gamma = \Gamma^*$. Moreover, the Markov chain has a unique stationary distribution, as it is an irreducible aperiodic Markov chain. Accordingly, the equation (ii) implies that $P = P^*$ (cf. Lemma 2 in Leroux [1992]). As a result, we have the following proposition.

Proposition 1

The parameters of the SMSSM (3.22)-(3.23) are g.s. identifiable if the following constraints are verified

1. *Prior information at t_0 is available:*

$$y_{t_0} = C(s)x_{t_0} + D(s)u_{t_0} + D_0(s) \text{ with } s = 1, \dots, \kappa \text{ and } x_{t_0} \neq 0, \tag{3.51}$$

2. $\forall (i,j),\ 1 \leq i, j \leq \kappa,\ P(i,j) \neq 0.$

Condition 2 implies that the finite hidden Markov chain is irreducible and aperiodic.

It is noteworthy that this proposition is easily extended in the case of a SMSSM with X_t and Y_t belong to $\mathbb{R}$, and u_t belongs to $\mathbb{R}^n$ (see Kalawoun et al. [2015b]).

3.4.4 Interpretation in the SoC case

In the *SoC* case, the conditions of Proposition 1 are naturally verified. Indeed, as explained in Section 3.2, at $t_0 = 0$ the battery is often in a resting state (i.e., the current is equal to zero on a long duration: $u_0 = 0$), hence the following prior information is

available:

$$y_0 = C(s_0)x_0 + D_0(s_0), \tag{3.52}$$

where y_0 is the OCV measurement and x_0 the corresponding SoC.

In addition, the Markov chain is naturally irreducible in the case of an electric battery. Indeed, the hidden Markov states reflect the different regimes of the underlying dynamics of the battery. Since the behavior of an electric battery is reversible, the transition from a Markov state to another is reversible too. The Markov chain is also aperiodic as the dynamics of the battery randomly change according to uncontrolled internal and external usage conditions.

3.5 Online state inference on SMSSMs

The filtering aims at estimating the states X_t and S_t given the observations $y_{0:t}$ up to time t, considering that the model parameters Θ is known.

An optimal estimation of X_t and S_t is a well-known NP problem and requires a prohibitive computational cost. A closed form solution would require running Kalman filters for each possible Markov sequence $s_{0:t}$; i.e., κ^{t+1} filters where κ is the number of hidden Markov states. Indeed, the expectation $\mathbb{E}_{y_{0:t},\Theta}[X_t]$ of X_t given the observations $y_{0:t}$ cannot be directly calculated, and the Markov sequence $s_{0:t}$ should be considered:

$$\mathbb{E}_{y_{0:t},\Theta}[X_t] = \sum_{s_{0:t}\in\mathcal{S}} p_\Theta(s_{0:t} \mid y_{0:t}) \cdot \mathbb{E}_{y_{0:t},s_{0:t},\Theta}[X_t], \tag{3.53}$$

where $\mathcal{S} = \{1, \ldots, \kappa\}^{t+1}$ and $\mathbb{E}_{y_{0:t},s_{0:t},\Theta}[X_t]$ is calculated using a Kalman filter.

To overcome this computational problem, a variety of suboptimal estimation algorithms has already been proposed.

Most of these algorithms are based on merging the components at time t like the Interacting Multiple Model (IMM) (see Blom and Bar-Shalom [1988] for more information). The IMM uses only κ parallel Kalman filters at each time t, and then merges their κ estimated x_t^i $(1 \leq i \leq \kappa)$ using a deterministic finite Gaussian mixture approximation.

Another possible suboptimal strategy is to approximate the continuous process X_t by a finite state process with fixed states. The SMSSM is thus reduced to an HMM with two discrete latent states, and the estimation algorithms of HMMs can be easily implemented (cf. Rabiner [1989]).

3.5.1 Filtering using Importance Sampling

An alternative method would use a Monte Carlo method to estimate $\mathbb{E}_{y_{0:t},\Theta}[X_t]$: a set of N independent *particles* $s^i_{0:t}$ $(1 \leq i \leq N)$ is simulated from the posterior density $p_\Theta(s_{0:t} \mid y_{0:t})$, and $\mathbb{E}_{y_{0:t},\Theta}[X_t]$ is then estimated by

$$\widehat{x}_t = \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{y_{0:t},s^i_{0:t},\Theta}[X_t]. \tag{3.54}$$

Unfortunately, it is difficult to simulate directly from the *target* distribution $p_\Theta(s_{0:t} \mid y_{0:t})$. Therefore, we use the well-known *importance sampling* method (Hammersley and Handscomb [1964]). The basic idea of this method is to sample from an *instrumental distribution* $q_\Theta(s_{0:t} \mid y_{0:t})$ from which it is easy to obtain samples. Then, *importance weights* are introduced to take into account that $S_{0:t}$ is simulated using $q_\Theta(s_{0:t} \mid y_{0:t})$ rather than $p_\Theta(s_{0:t} \mid y_{0:t})$. The importance weights are given by

$$w^i_t = w(s^i_{0:t}) = \frac{p_\Theta(s^i_{0:t} \mid y_{0:t})}{q_\Theta(s^i_{0:t} \mid y_{0:t})}, \tag{3.55}$$

where $s^i_{0:t} \overset{iid}{\sim} q_\Theta(s_{0:T} \mid y_{1:T})$. Accordingly, $\mathbb{E}_{y_{0:t},\Theta}[X_t]$ is estimated as follows:

$$\widehat{x}_t = \sum_{i=1}^{N} \widehat{w}^i_t \, \mathbb{E}_{y_{0:t},s^i_{0:t},\Theta}[X_t], \tag{3.56}$$

where $\widehat{w}^i_t$ denotes the normalized version of w^i_t: $\sum_{i=1}^{N} \widehat{w}^i_t = 1$. Convergence results of (3.56) are detailed in Doucet et al. [2001b].

However, sampling from $q_\Theta(s_{0:t} \mid y_{0:t})$ would have a computational complexity increasing at least linearly with t (Doucet et al. [2001a]). Hence, since S_t is a Markov chain, we naturally resort to the *sequential importance sampling* that admits a fixed computational complexity at each time step t.

3.5.1.1 Sequential Importance Sampling

At time t, the instrumental distribution $q_\Theta(s_{0:t} \mid y_{0:t})$ can be written as follows:

$$q_\Theta(s_{0:t} \mid y_{0:t}) = q_\Theta(s_0) \prod_{k=1}^{t} q_\Theta(s_k \mid s_{0:k-1}, y_{0:t}). \tag{3.57}$$

Therefore at time t, the simulated sequences $s^i_{0:t-1}$ will be modified as the observation y_t is considered. The sequential sampling allows the simulation of s^i_t without modifying the previously simulated sequences $s^i_{0:t-1}$. This is possible if the instrumental distribution

has the following form:

$$q_\Theta(s_{0:t} \mid y_{0:t}) = q_\Theta(s_0) \prod_{k=1}^{t} q_\Theta(s_k \mid s_{0:k-1},\, y_{0:k}). \tag{3.58}$$

Importance weights - When the instrumental distribution $q_\Theta(s_{0:t} \mid y_{0:t})$ has the form (3.58), the importance weights can be recursively evaluated as follows:

$$w_t^i = w_{t-1}^i \cdot \tilde{w}_t^i, \tag{3.59}$$

where

$$\begin{aligned} \tilde{w}_t^i =& \frac{p_\Theta(y_t \mid s_{0:t}^i, y_{0:t-1}) p_\Theta(s_t^i \mid s_{t-1}^i)}{p_\Theta(y_t \mid y_{0:t-1}) q_\Theta(s_t^i \mid s_{0:t-1}^i, y_{0:t})} \\ \propto& \frac{p_\Theta(y_t \mid s_{0:t}^i, y_{0:t-1}) p_\Theta(s_t^i \mid s_{t-1})}{q_\Theta(s_t^i \mid s_{0:t-1}^i, y_{0:t})}. \end{aligned} \tag{3.60}$$

Choice of the instrumental distribution - The necessary condition for selecting the instrumental distribution is that $q_\Theta(s_{0:t} \mid y_{0:t}) = 0$ implies $p_\Theta(s_{0:t} \mid y_{0:t}) = 0$. This condition ensures that in equation (3.55) the denominator is not equal to zero.

Here we choose $q_\Theta(s_t \mid s_{0:t-1}, y_{0:t}) = p_\Theta(s_t \mid s_{0:t-1}, y_{0:t})$ which is the distribution that minimizes the variance of the importance weights given $s_{0:t-1}$ and $y_{0:t}$ (Doucet et al. [2001b]). In addition, it is easy to obtain samples from $p_\Theta(s_t \mid s_{0:t-1}, y_{0:t})$ as it can be computed using κ Kalman filters; hence the computational complexity fixed at each time t:

$$p_\Theta(s_t \mid s_{0:t-1}, y_{0:t}) = \frac{p_\Theta(s_t \mid s_{t-1}) \cdot p_\Theta(y_t \mid s_{0:t}, y_{0:t-1})}{p_\Theta(y_t \mid s_{0:t-1}, y_{0:t-1})}, \tag{3.61}$$

with

$$p_\Theta(y_t \mid s_{0:t-1}, y_{0:t-1}) = \sum_{s_t=1}^{\kappa} p_\Theta(s_t \mid s_{t-1}) \cdot p_\Theta(y_t \mid s_{0:t}, y_{0:t-1}), \tag{3.62}$$

where $p_\Theta(y_t \mid s_{0:t}, y_{0:t-1})$ is calculated using a Kalman filter.

The prior distribution $p_\Theta(s_t|s_{t-1})$ can also be used as an instrumental distribution. In this case, based on (3.60), the computation of importance weights requires only a one step Kalman filter. However, the variance of the importance weights becomes larger (Doucet et al. [2001b]).

3.5.1.2 Sequential Importance Resampling

The entire path up to t of the target distribution $p_\Theta(s_{0:t} \mid y_{0:t})$ and the instrumental distribution $q_\Theta(s_{0:t} \mid y_{0:t})$ could be far apart. Therefore after few simulation iterations, a lot of importance weights could be very close to zero. To avoid this *degeneracy phenomenon*, a resampling step is generally added to the sequential importance sampling

algorithm (Doucet and Johansen [2009]). It consists in discarding the particles $s_{0:t}^i$ with low weights and duplicating the ones with high weights. This resampling step can be done at each time t, or when this is considered necessary given the variance of the importance weights. The Effective Sample Size (ESS) criterion is related on this variance (Kong et al. [1994]). It is given at time t by

$$ESS = \frac{1}{\sum_{i=1}^{N} (\widehat{w}_t^i)^2}. \tag{3.63}$$

The ESS takes values between 1 and N: it is maximal when all particles have the same weights

$$ESS = N \text{ when } \widehat{w}_t^i = \frac{1}{N} \, (1 \leq i \leq N),$$

and minimal when the weight of one particle is not equal to zero

$$ESS = 1 \text{ when } \exists j \text{ such as } \widehat{w}_t^i = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{otherwize.} \end{cases}$$

In practice, it is more reasonable to resample when the ESS is lower than a predefined threshold. Here, we use the multinomial sampling procedure which is the most popular resampling one (Gordon et al. [1993]). This procedure associates a number of offspring N_t^i with each particle $s_{0:t}^i$, where N_t^i $(1 \leq i \leq N)$ are sampled from a multinomial distribution of parameters $(N, \widehat{w}_t^{1:N})$. The weights of the resampled particles are then set to $\frac{1}{N}$.

The proposed method for the estimation of X_t using a sequential importance sampling is presented in Algorithm 1. It is noteworthy that x_t^i is not simulated but calculated using a Kalman filter given the particle $s_{0:t}^i$ and $y_{0:t}$.

Algorithm 1 Sequential Importance Sampling for state inference on SMSSMs

Input $\leftarrow \Theta, x_0$

Init $\leftarrow s_0^i \overset{iid}{\sim} \Pi \ (\forall i = 1:N)$

for $t = 1:T$ and $i = 1:N$ **do**

1. Sample $s_t^i \sim p_\Theta(s_t \mid s_{0:t-1}^i, y_{0:t})$, thus $s_{0:t}^i = \left(s_{0:t-1}^i, s_t^i\right)$
2. Given $s_{0:t}^i$ and $y_{0:t}$, calculate $x_t^i \overset{\triangle}{=} \mathbb{E}_{y_{0:T}, s_{0:T}^i}[X_t]$ using a Kalman filter
3. Calculate importance weights w_t^i
4. Calculate $\widehat{x}_t$
5. Selection step: if necessary sample $s_{0:t}^i \overset{iid}{\sim} \sum_{i=1}^{N} w_t^i \delta(s_{0:t} - s_{0:t}^i)$, where $\delta(\cdot)$ is a Dirac function with mass at zero, and set $w_t^i = \frac{1}{N}$

end for

3.5.2 Sensitivity Analysis using simulated data

In this section, we validate the presented estimation algorithm of the state of interest. To this purpose, we simulate a learning dataset with $T = 500$ using a SMSSM with $\kappa = 3$. Figure 3.5 shows the simulated dataset, and Table 3.1 summarizes the values of the model parameters. We suppose that Δt is equal to 1.

	B	C	D	D_0	σ_X^2	σ_Y^2
$s = 1$	-1.20	0.11	-350	375	10^{-4}	20
$s = 2$	-1.35	0.15	-400	380	10^{-4}	1
$s = 3$	-1.30	0.20	-420	385	10^{-4}	1

TABLE 3.1: Parameters values of the SMSSM with $\kappa = 3$ used to generate the simulated dataset

The main difficulty in this simulated model is that C is well below 1, with C describing the influence of the state of interest X_t on the observation Y_t. When C has small values, meaning that X_t has a very limited impact on Y_t, it is then difficult to estimate X_t based solely on Y_t. It is noteworthy that this simulated dataset matches the behavior of a battery.

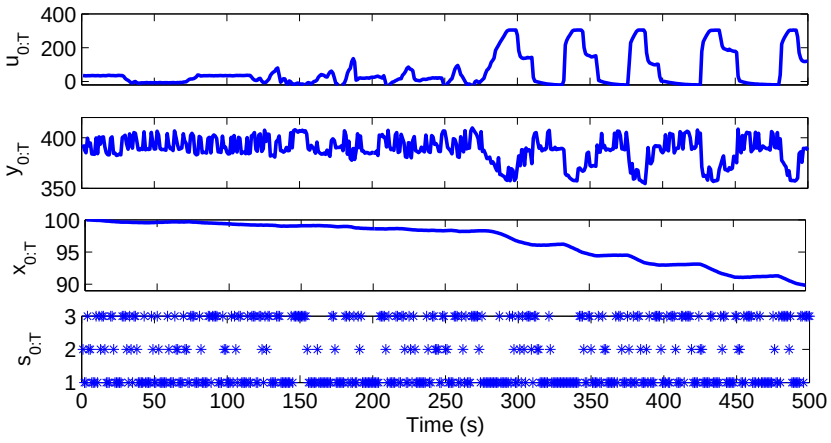

FIGURE 3.5: [Simulated data] Learning dataset generated using a SMSSM with $\kappa = 3$

Figures 3.6 and 3.7 show the estimation error using different numbers of particles. Indeed, the number of particles N to be used for the online estimation of $\widehat{x}_t$ is constrained by the limited hardware resources in embedded applications. Based on the simulated data, the numerical results show that a small number of particles ($N \simeq 10$) is sufficient to accurately estimate $\widehat{x}_t$.

3.6 Conclusion

In this chapter, we have proposed a new *SoC* model using a switching Markov state-space model which is a linear Gaussian state-space model with parameters indexed by

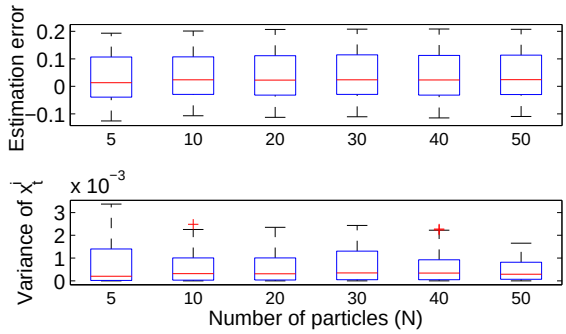

FIGURE 3.6: [Simulated data] Influence of the number of particles N on the online estimation of $\widehat{x}_t$: Boxplots of the estimation error and the variance of x_t^i

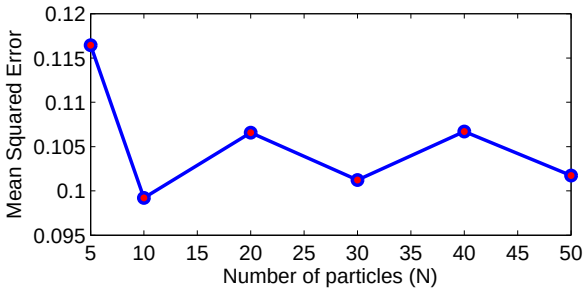

FIGURE 3.7: [Simulated data] Mean squared error of the online estimated x_t *vs.* the number of particles N

a Markov chain. This model could take into account the random variation of battery dynamics during its charge/discharge according to the usage conditions. The transition and observation equations are based on physical models: the Ah-counting and the IR models respectively.

The identifiability of this model was then verified to guarantee its relevance, particularly when its parameters will be estimated. We have proved that if a prior information relating the observations Y_t to the unknown continuous state X_t is available at time t_0, and if the Markov chain is irreducible and aperiodic, the set of the model parameters will be globally structurally identifiable.

For fixed model parameters, the state of interest X_t was estimated using a suboptimal algorithm based on a sequential importance sampling technique, as achieving a closed solution of an optimal estimation is at a prohibitive computational cost. The numerical results based on a simulated dataset have shown the relevance of the proposed algorithm, particularly in embedded applications with limited hardware capacity. Indeed, a small number of particles ($N \simeq 10$ for $\kappa = 3$) is sufficient to provide an accurate estimation of X_t.

In the next chapter, we consider the problem of estimation of the unknown parameters of the considered SMSSM using a Bayesian approach, particularly *via* a *Gibbs sampler*.

Chapter 4

Bayesian parameters inference for a SMSSM

Ce chapitre est dédié à l'estimation de l'ensemble des paramètres Θ par une approche bayésienne, étant donné une base d'apprentissage $\{y_{0:T},\, u_{0:T}\}$ et sous contrainte que les deux séquences d'états continus $X_{0:T}$ et discrets $S_{0:T}$ sont inconnues. Rappelons que Θ est un ensemble de cardinal $n_\Theta = 6\kappa + 1$:

$$\Theta = \left\{P,\, \Gamma = \{B(s),\, \sigma_X(s),\, C(s),\, D(s),\, D_0(s),\, \sigma_Y(s)\}_{s=1,\dots,\kappa}\right\}, \tag{4.1}$$

et notons Θ_i le i-ème élément de Θ et $\Theta_{-i} = \{\Theta_j\}_{j\neq i}$ l'ensemble des éléments restants.

Dans ce type d'approche, on considère les paramètres comme des variables aléatoires de loi *a priori* $p(\Theta)$. Ces lois reflètent les informations que l'on a sur les paramètres. On cherche alors l'estimateur $\widehat{\Theta}_{\text{Bayes}}$ minimisant l'erreur quadratique moyenne entre les vrais paramètres et ceux estimés. Cet estimateur correspond à l'espérance de Θ suivant la loi *a posteriori* $p(\Theta|y_{0:T})$ (Robert and Casella [2004]) :

$$\begin{aligned}\widehat{\Theta}_{\text{Bayes}} &= \int_\Theta \Theta \cdot p(\Theta \mid y_{0:T}) d\Theta \\ &= \int_\Theta \Theta \cdot \sum_{s_{0:T}} \int_{x_{0:T}} p(\Theta, s_{0:T}, x_{0:T} \mid y_{0:T}) dx_{0:T}\, d\Theta.\end{aligned} \tag{4.2}$$

L'approximation de cette intégrale par une méthode de Monte-Carlo directe est difficile à réaliser dans la mesure où l'expression exacte de $p(\Theta, s_{0:T}, x_{0:T} \mid y_{0:T})$ n'est pas disponible :

$$p(\Theta, s_{0:T}, x_{0:T} \mid y_{0:T}) \propto p(\Theta) \cdot p_\Theta(s_{0:T}|y_{0:T}) \cdot p_\Theta(x_{0:T}|s_{0:T}, y_{0:T}). \tag{4.3}$$

Nous avons alors recours à l'échantillonneur de Gibbs (Gelfand and Smith [1990]) qui est un choix naturel dans le cas du SMSSM compte tenu des relations d'indépendance conditionnelle présentes dans la loi jointe (4.3).

L'échantillonneur de Gibbs est une méthode itérative de Monte-Carlo qui consiste à :

1. **Initialisation**
   - $k = 0$
   - $\Theta(k) \sim p(\Theta)$
   - $\{s_{0:T}(k), x_{0:T}(k)\} \sim p_{\Theta(k)}(\cdot \mid y_{0:T})$
   - $k = k + 1$
2. **Tant que non convergence**
   (a) $\Theta_1(k) \sim p(\cdot \mid \Theta_{-1}, s_{0:T}(k-1), x_{0:T}(k-1), y_{0:T})$
   $\vdots$
   $\Theta_{n_\Theta}(k) \sim p(\cdot \mid \Theta_{-n_\Theta}, s_{0:T}(k-1), x_{0:T}(k-1), y_{0:T})$
   (b) $s_{0:T}(k) \sim p_{\Theta(k)}(\cdot \mid y_{0:T}, x_{0:T}(k-1))$
   (c) $x_{0:T}(k) \sim p_{\Theta(k)}(\cdot \mid y_{0:T}, s_{0:T}(k))$
   (d) $k = k + 1$

   Fin tant que

Les Θ_i peuvent être tirés directement selon la loi *a posteriori* $p(\Theta_i \mid \Theta_{-i}, s_{0:T}, x_{0:T}, y_{0:T})$ lorsque cette loi est la loi conjuguée de la loi *a priori* correspondante $p(\Theta_i)$; i.e., la loi *a posteriori* appartient a la même famille que celle de la loi *a priori*. Plusieurs algorithmes ont été développés pour tirer efficacement selon $p_\Theta(s_{0:T} \mid y_{0:T}, x_{0:T})$ (Carter and Kohn [1994], Chib [1996]) et $p_\Theta(x_{0:T} \mid y_{0:T}, s_{0:T})$ (Carter and Kohn [1994], De Jong and Shephard [1995], Durbin and Koopman [2002], Frühwirth-Schnatter [1994]). De ce fait, une application directe de l'échantillonneur de Gibbs peut être réalisée.

Cependant, les deux séquences latentes $X_{0:T}$ et $S_{0:T}$ sont fortement corrélées, ce qui rend l'exploration de l'espace des paramètres, et par la suite la convergence de l'algorithme, très lente (Carter and Kohn [1996]). Dans certains cas, cet algorithme peut rester bloqué dans un état lorsque l'une des variances $\sigma^2_X(s)$ est égale à 0 (Carter and Kohn [1996]). Afin de surmonter cette difficulté, la séquence d'états discrets $s_{0:T}$ peut être tirée selon une loi conditionnelle aux observations seulement. L'étape (b) de l'algorithme de Gibbs est donc remplacée par :

(b') $s_{0:T}(k) \sim p_{\Theta(k)}(s_{0:T} \mid y_{0:T})$

Cependant, le tirage selon $p_{\Theta(k)}(s_{0:T} \mid y_{0:T})$ n'étant pas évident, nous réutilisons la technique d'échantillonnage d'importance séquentiel du Chapitre 3 pour obtenir une approximation empirique de cette loi :

$$\widehat{p}_{\Theta(k)}(s_{0:T}|y_{0:T}) = \sum_{i=1}^{N} w_T^i \delta(s_{0:T} - s_{0:T}^i), \tag{4.4}$$

où N est le nombre de particules, $s_{0:T}^i \overset{iid}{\sim} q_{\Theta(k)}(s_{0:T} \mid y_{0:T})$ et w_T^i, $1 \leq i \leq N$, sont les poids d'importance associés à $s_{0:T}^i$ avec

$$w_T^i = \frac{p_{\Theta(k)}(s_{0:T}^i \mid y_{0:T})}{q_{\Theta(k)}(s_{0:T}^i \mid y_{0:T})}.$$

Le tirage selon $p_{\Theta(k)}(s_{0:T} \mid y_{0:T})$ est ainsi remplacé par un tirage selon $\widehat{p}_{\Theta(k)}(s_{0:T}|y_{0:T})$. Cette version modifiée de l'échantillonneur de Gibbs est appelée *échantillonneur particulaire de Gibbs*. Elle a été développée par Andrieu et al. [2010] pour les SSMs (state-space models) et améliorée par Whiteley et al. [2010] pour les SSSMs (switching space-space models).

Afin de calibrer cet algorithme, nous avons généré une base de données de taille $T = 500$ à partir d'un SMSSM à $\kappa = 3$ états cachés. L'impact de la loi *a priori* et du nombre de particules est étudié.

Les lois *a priori* de $B(s)$, $C(s)$, $D(s)$ et $D_0(s)$ sont des lois gaussiennes : leurs lois *a posteriori* sont aussi des gaussiennes. Quant aux lois *a priori* des variances $\sigma_X^2(s)$ et $\sigma_Y^2(s)$, elles correspondent à des inverse-gamma. Les paramètres des différentes lois *a posteriori* sont fournis dans la Section 4.3.2. Nous avons testé trois configurations de lois *a priori* avec un nombre de particules $N = 200$, 10^4 itérations de préchauffement suivies de 4×10^4 itérations. Les trois configurations des lois *a priori* sont comme suit :

1. informative : les moyennes des lois sont égales aux vrais paramètres et leur variance est relativement grande (cf. Table 4.2),
2. non-informative : les états ne sont *a priori* pas différenciés. Pour chaque paramètre, nous assignons la même moyenne à tous les états avec une grande variance (cf. Table 4.3),
3. incorrecte : les moyennes des lois sont très différentes des vrais paramètres et leur variance est relativement petite (cf. Table 4.4).

Les résultats montrent que dans le cas d'une loi *a priori* non-informative, l'algorithme exploite l'espace de paramètres mieux que dans le cas de loi *a priori* informative du fait que la variance de la loi non-informative est plus grande que celle de la loi informative (cf. Figures 4.1 and 4.2). Toutefois, l'estimation de B est la moins précise dans les deux cas. Ce résultat est attendu vu que B décrit l'évolution de l'état inconnu X_t. Il convient de noter aussi que dans le cas d'une loi *a priori* incorrecte, les paramètres estimés ne convergent pas vers les vrais paramètres même après 4×10^4 itérations (cf. Figure 4.3).

Afin de tester l'impact du nombre de particules sur les performances de l'algorithme, nous avons testé plusieurs valeurs de N avec une loi *a priori* non-informative, 10^4 itérations de préchauffement suivies de 4×10^4 itérations. Les résultats montrent que les fonctions d'auto-corrélation de B, D, σ_X^2 et σ_Y^2 diminuent nettement en utilisant seulement 10

particules. Concernant C et D_0, les performances s'améliorent lorsque N augmente, mais il paraît nécessaire d'utiliser plus que 200 particules pour que leurs fonctions d'auto-corrélation diminuent considérablement (cf. Figure 4.4).

Pour conclure, l'approche bayésienne permet d'intégrer les connaissances des experts dans la phase d'estimation des paramètres du modèle, et de ce fait faire face au problème d'identifiabilité. Cependant, dans le cas des SMSSMs, l'échantillonneur particulaire de Gibbs est très coûteux. En effet, pour garantir la convergence de l'algorithme pour un SMSSM à 3 états cachés, il paraît nécessaire d'utiliser plus que 200 particules avec 4×10^4 itérations.

Contents

In the previous chapter, we have developed a new *SoC* estimator using SMSSMs:

$$X_t = X_{t-1} + B(S_t)u_t\Delta t + W_t, \tag{4.5}$$
$$Y_t = C(S_t)X_t + D(S_t)u_t + D_0(S_t) + \Sigma_t, \tag{4.6}$$

where $p(w_t|s_t) = \mathcal{N}(w_t; 0, \sigma_X^2(s_t))$, $p(\varepsilon_t|s_t) = \mathcal{N}(\varepsilon_t; 0, \sigma_Y^2(s_t))$, $s_t \in \{1, \ldots, \kappa\}$, and Θ is the set of unknown parameters of cardinality $n_\Theta = 6\kappa + 1$:

$$\Theta = \{P,\ \Gamma\}\,, \text{ where } \Gamma = \{B(s),\ \sigma_X(s),\ C(s),\ D(s),\ D_0(s),\ \sigma_Y(s)\}_{s=1,\ldots,\kappa}\,.$$

The state of interest X_t was estimated using the sequential importance sampling, considering that the set of model parameters Θ is known.

The aim of this chapter is to perform Bayesian inference for the SMSSM (3.22)-(3.23), conditional on a learning dataset $\{y_{0:T}, u_{0:T}\}$ and considering that both latent sequences $X_{0:T}$ and $S_{0:T}$ and the set of parameters Θ are unknown.

The principle of this approach is to consider the parameters as random variables with prior distribution $p(\Theta)$. The Bayesian inference then relies on the posterior joint distribution $p(\Theta, s_{0:T}, x_{0:T} \mid y_{0:T})$. Since an efficient application of MCMC algorithms to this joint distribution is difficult, we resort to the *Gibbs sampling* (Gelfand and Smith [1990]). This choice is naturally justified by the fact that the SMSSMs present a specific conditional independence relations, and the conditional distribution of each variable given all remaining ones has a simple form.

For the SMSSM (4.5)-(4.6), the k-th iteration of a direct application of the Gibbs sampler would consist of successively sampling $\Theta(k)$, $s_{0:T}(k)$ and $x_{0:T}(k)$ from the posterior probability distributions $p(\Theta|y_{0:T}, s_{0:T}(k-1), x_{0:T}(k-1))$, $p_{\Theta(k)}(s_{0:T}|y_{0:T}, x_{0:T}(k-1))$ and $p_{\Theta(k)}(x_{0:T}|y_{0:T}, s_{0:T}(k))$ respectively.

It is possible to efficiently sample from $p(\Theta|y_{0:T}, s_{0:T}, x_{0:T})$ when its is a conjugate of the prior $p(\Theta)$. A variety of efficient algorithms have been developed to sample from $p_\Theta(x_{0:T}|y_{0:T}, s_{0:T})$ (see for instance Carter and Kohn [1994], De Jong and Shephard [1995], Durbin and Koopman [2002], Frühwirth-Schnatter [1994]). Sampling from $p_\Theta(s_{0:T}|y_{0:T}, x_{0:T})$ can also be efficiently performed as $\{S_t\}$ is a Markov chain (see for instance Carter and Kohn [1994], Chib [1996]).

However, the latent sequences $X_{0:T}$ and $S_{0:T}$ are usually strongly correlated which leads to a very slow exploration of the variables space by the Gibbs algorithm. To overcome this problem, the discrete state sequence $s_{0:T}$ is simulated given only $y_{0:T}$, where $x_{0:T}$ is marginalized out (see for instance Carter and Kohn [1996], Gerlach et al. [2000]). This is practically feasible since, given both Θ and $s_{0:T}$, the conditional distribution of the continuous states $X_{0:T}$ is fully characterized using the forward/backward Kalman techniques (Shumway and Stoffer [1982]).

However, as shown in Chapter 3, it is impossible to directly sample from the target distribution $p_\Theta(s_{0:T}|y_{0:T})$. Therefore, importance sampling is used to estimate it. This modified version of the Gibbs sampler is referred to as *particle Gibbs sampler* which was developed by Andrieu et al. [2010] in the state-space models context, and improved by Whiteley et al. [2010] in the switching state-space models context.

In this chapter, we use the particle Gibbs sampler to estimate the unknown parameters of the SMSSM (4.5)-(4.6). In addition, in order to calibrate this algorithm, a sensitivity analysis is performed based on simulated data. Specifically, the influence of the prior probability distribution $p(\Theta)$ and the number of particules is studied.

4.1 Principle of the Bayesian parameters inference

Let us consider a loss function $L : \Upsilon \times \Upsilon \rightarrow \mathbb{R}_+$, which takes the true model parameters Θ and its estimate $\widehat{\Theta}$ as inputs, and returns a real-valued positive loss. The Bayes' estimator $\widehat{\Theta}_{\text{Bayes}}$ of the model parameters is the argument minimizing the expected posterior loss:

$$\widehat{\Theta}_{\text{Bayes}} = \arg\min_{\widehat{\Theta}} \mathbb{E}_{Y_{0:T}} \left[L\left(\Theta, \widehat{\Theta}\right) \right], \tag{4.7}$$

where this expected loss is given by

$$\mathbb{E}_{Y_{0:T}} \left[L\left(\Theta, \widehat{\Theta}\right) \right] = \int_\Theta L\left(\Theta, \widehat{\Theta}\right) p(\Theta \mid y_{0:T})\, d\Theta. \tag{4.8}$$

Let us consider that the loss function $L(\Theta, \widehat{\Theta})$ is equal to the quadratic difference between the true and the estimated parameters

$$L(\Theta, \widehat{\Theta}) = \left(\Theta - \widehat{\Theta}\right)^T \left(\Theta - \widehat{\Theta}\right).$$

The estimate minimizing this quadratic loss function is then the expected value of the posterior given by (Robert and Casella [2004])

$$\widehat{\Theta}_{\text{Bayes}} = \mathbb{E}_{Y_{0:T}}[\Theta] = \int_\Theta \Theta \cdot p(\Theta \mid y_{0:T}) d\Theta. \tag{4.9}$$

This parameters estimator (4.9) is called the *minimum mean square error* estimator.

The posterior parameters distribution is obtained by the Bayes' theorem from the parameters prior distribution $p(\Theta)$ and the marginal likelihood $p_\Theta(y_{0:T})$

$$\begin{aligned} p(\Theta|y_{0:T}) =& \frac{p(\Theta) \cdot p_\Theta(y_{0:T})}{p(y_{0:T})} \\ \propto& p(\Theta) \cdot p_\Theta(y_{0:T}). \end{aligned} \tag{4.10}$$

4.2 Gibbs sampler for SMSSM

The aim is to approximate the minimum mean square error estimator (4.9) using a Monte Carlo method. Unfortunately, it is not obvious to sample from the target distribution $p(\Theta \mid y_{0:T})$ without considering the latent variables $X_{0:T}$ and $S_{0:T}$. The Bayesian inference then relies on the joint posterior distribution:

$$p(\Theta, s_{0:T}, x_{0:T} \mid y_{0:T}) \propto p(\Theta) \cdot p_\Theta(s_{0:T}|y_{0:T}) \cdot p_\Theta(x_{0:T}|s_{0:T}, y_{0:T}). \tag{4.11}$$

We shall denote by Θ_{-i} the i-th element of Θ, and by $\Theta_{-i} = \{\Theta_j\}_{j \neq i}$ the set of its remaining elements. Sampling directly from the joint distribution (4.11) is difficult as a closed-form expression is unavailable. Instead, the Gibbs sampler could be used. Indeed, the distribution of interest $p(\Theta|s_{0:T}, x_{0:T}, y_{0:T})$ is multivariate ($n_\Theta = 6\kappa + 1$), and the conditional distribution of each component $p(\Theta_i \mid \Theta_{-i}, s_{0:T}, x_{0:T}, y_{0:T})$ has a simple form when it is a conjugate of the prior. In addition, due to the hierarchical structure of the SMSSM, the joint distribution is easily decomposed into conditional independent distributions as shown in (4.11).

Accordingly, the Gibbs sampler is an iterative Monte Carlo technique that, in its standard version, successively samples from the conditional distributions $p(\Theta_i \mid \Theta_{-i}, s_{0:T}, x_{0:T}, y_{0:T})$ (for $i = 1, \ldots, n_\Theta$), $p_\Theta(s_{0:T} \mid y_{0:T}, x_{0:T})$ and $p_\Theta(x_{0:T} \mid y_{0:T}, s_{0:T})$. To be more specific, the Gibbs sampler consists of:

1. **Initialization**
   - $k = 0$
   - $\Theta(k) \sim p(\Theta)$
   - $\{s_{0:T}(k), x_{0:T}(k)\} \sim p_{\Theta(k)}(\cdot \mid y_{0:T})$
   - $k = k + 1$
2. **While non convergence**
   (a) $\Theta_1(k) \sim p(\cdot \mid \Theta_{-1}, s_{0:T}(k-1), x_{0:T}(k-1), y_{0:T})$
   $\vdots$
   $\Theta_{n_\Theta}(k) \sim p(\cdot \mid \Theta_{-n_\Theta}, s_{0:T}(k-1), x_{0:T}(k-1), y_{0:T})$

(b) $s_{0:T}(k) \sim p_{\Theta(k)}\left(\cdot \mid y_{0:T}, x_{0:T}(k-1)\right)$

(c) $x_{0:T}(k) \sim p_{\Theta(k)}\left(\cdot \mid y_{0:T}, s_{0:T}(k)\right)$

(d) $k = k + 1$

end while

Hence, the Gibbs sampler requires the full knowledge of all these conditional distributions, and the ability to generate samples from these ones. Sampling from $p(\Theta_i \mid \Theta_{-i}, s_{0:T}, x_{0:T}, y_{0:T})$ is feasible when it is a conjugate of the prior. Moreover, sampling from $p_\Theta(x_{0:T}|y_{0:T}, s_{0:T})$ and $p_\Theta(s_{0:T}|y_{0:T}, x_{0:T})$ can also be performed efficiently since the closed-form expression of $p_\Theta(x_{0:T}|y_{0:T}, s_{0:T})$ can be computed using the forward/backward Kalman techniques and $\{S_t\}$ is a Markov chain (see Chapter 13 of Frühwirth-Schnatter [2006] for detailed formulas).

However, using this standard approach, the exploration of the variables space and thus the convergence speed might be very slow due to the close association between $X_{0:T}$ and $S_{0:T}$. The algorithm may even be strapped to a state when one of the process variances $\sigma_X^2(s)$ is assumed to be exactly 0 (Carter and Kohn [1996]), Chapter 6 of Cappé et al. [2005]).

4.3 Particle Gibbs sampler for SMSSM

This lack of convergence can be overcome by sampling a discrete sequence without considering the continuous states(Carter and Kohn [1996]). At the k-th iteration, the discrete state sequence $s_{0:T}(k)$ is sampled from $p_{\Theta(k)}(s_{0:T}|y_{0:T})$. The step (b) of the previous algorithm is replaced by

(b') $s_{0:T}(k) \sim p_{\Theta(k)}\left(s_{0:T} \mid y_{0:T}\right)$

However as shown in Section 3.5.1, it is not possible to directly simulate from the target distribution $p_{\Theta(k)}(s_{0:T}|y_{0:T})$. Therefore, we resort to the importance sampling method to estimate $p_{\Theta(k)}(s_{0:T}|y_{0:T})$ as in Chapter 3. The target distribution $p_{\Theta(k)}(s_{0:T}|y_{0:T})$ is thus estimated as follows:

$$\widehat{p}_{\Theta(k)}(s_{0:T}|y_{0:T}) = \sum_{i=1}^{N} \widehat{w}_T^i \cdot \delta(s_{0:T} - s_{0:T}^i), \tag{4.12}$$

where $\delta(\cdot)$ is a Dirac function with mass at zero, N is the number of particles, and $\widehat{w}_T^i$ is the normalized version of the importance weights associated with particle $s_{0:T}^i$ and given by (3.55). This method is referred to as *particle Gibbs sampler* (Andrieu et al. [2010], Whiteley et al. [2010])). Here, the discrete state sequences $\{s_{0:T}^i\}_{i=1}^N$ are

sequentially simulated as in Section 3.5.1. The algorithm of the particle Gibbs sampler for the inference of the parameters of the SMSSM (4.5)-(4.6) is presented in Algorithm 2.

Algorithm 2 Particle Gibbs sampler algorithm for the SMSSM (4.5)-(4.6)

Input: $y_{0:T}$, $u_{0:T}$, κ, N, and hyperparameters of the prior distribution

Initialisation: Set $\widehat{\Theta}(0)$ and $k = 0$

while stop criterion is not verified **do**

1. Sample $\{s^i_{0:T}\}_{i=1}^N$ using particle filter, given $\widehat{\Theta}(k)$ and $y_{0:T}$
2. Sample $s_{0:T}(k) \sim \widehat{p}_\Theta(s_{0:T}|y_{0:T})$
3. Calculate $x_{0:T}(k)$ using a Kalman filter, given $\widehat{\Theta}(k)$, $y_{0:T}$ and $s_{0:T}(k)$
4. Sample $\widehat{\Theta}(k+1) \sim p(\Theta \mid s_{0:T}(k), x_{0:T}(k), y_{0:T})$
5. Set $k = k + 1$

end while

Output: $\widehat{\Theta}(k)$

4.3.1 Choice of the prior distribution of the state-space parameters

One of the main questions in Bayesian inference is the choice of the prior $p(\Theta)$. A sensible choice is the *conjugate priors* which enables us to compute a closed-form expression of the posterior $p(\Theta|s_{0:T}, x_{0:T}, y_{0:T})$. To avoid confusion, the parameters involved in the prior distribution are called *hyperparameters*. The parameters are assumed to be *a priori* independent:

$$p(\Theta) = p(P) \prod_{s=1}^{\kappa} p\left(B(s)\right)\, p(C(s))\, p(D(s))\, p(D_0(s))\, p(\sigma_X^2(s))\, p(\sigma_Y^2(s)). \tag{4.13}$$

Prior and posterior of B(s) - Assuming $B(s)$ is to be estimated for $s = 1, \ldots, \kappa$. The prior associated to the conjugate posterior is $\mathcal{N}(B(s); \alpha(s), \beta(s))$. Thus, the posterior $p(B(s) \mid \Theta_{-B(s)}, s_{0:T}, x_{0:T}, y_{0:T})$ is proportional to the pdf of a normal distribution with mean $m_B(s)$ and variance $\sigma_B^2(s)$ (cf. Appendix B), where

$$m_B(s) = \frac{\sum\limits_{\substack{t=1 \\ s_t=s}}^{T} u_t \Delta t \left(x_t - x_{t-1}\right) + \frac{\alpha(s)}{\beta(s)} \sigma_X^2(s)}{\sum\limits_{\substack{t=1 \\ s_t=s}}^{T} u_t^2 \Delta t^2 + \frac{\sigma_X^2(s)}{\beta(s)}}, \tag{4.14}$$

$$\sigma_B^2(s) = \frac{\sigma_X^2(s)}{\sum\limits_{\substack{t=1 \\ s_t=s}}^{T} u_t^2 \Delta t^2 + \frac{\sigma_X^2(s)}{\beta(s)}}. \tag{4.15}$$

Prior and posterior of C(s) - Assuming $C(s)$ is to be estimated for $s = 1, \dots, \kappa$. The prior associated to the conjugate posterior is $\mathcal{N}(C(s); \gamma(s), \mu(s))$. Thus, the posterior $p(C(s) \mid \Theta_{-C(s)}, s_{0:T}, x_{0:T}, y_{0:T})$ is proportional to the pdf of a normal distribution with mean $m_C(s)$ and variance $\sigma_C^2(s)$ (cf. Appendix B), where

$$m_C(s) = \frac{\sum\limits_{\substack{t=1 \\ s_t=s}}^{T} x_t \left(y_t - D(s)u_t - D_0(s)\right) + \frac{\gamma(s)}{\mu(s)}\sigma_Y^2(s)}{\sum\limits_{\substack{t=1 \\ s_t=s}}^{T} x_t^2 + \frac{\sigma_Y^2(s)}{\mu(s)}}, \tag{4.16}$$

$$\sigma_C^2(s) = \frac{\sigma_Y^2(s)}{\sum\limits_{\substack{t=1 \\ s_t=s}}^{T} x_t^2 + \frac{\sigma_Y^2(s)}{\mu(s)}}. \tag{4.17}$$

Prior and posterior of D(s) - Assuming $D(s)$ is to be estimated for $s = 1, \dots, \kappa$. The prior associated to the conjugate posterior is $\mathcal{N}(D(s); \zeta(s), \epsilon(s))$. Thus, the posterior $p(D(s) \mid \Theta_{-D(s)}, s_{0:T}, x_{0:T}, y_{0:T})$ is proportional to the pdf of a normal distribution with mean $m_D(s)$ and variance $\sigma_D^2(s)$ (cf. Appendix B), where

$$m_D(s) = \frac{\sum\limits_{\substack{t=1 \\ s_t=s}}^{T} u_t \left(y_t - C(s)x_t - D_0(s)\right) + \frac{\zeta(s)}{\epsilon(s)}\sigma_Y^2(s)}{\sum\limits_{\substack{t=1 \\ s_t=s}}^{T} u_t^2 + \frac{\sigma_Y^2(s)}{\epsilon(s)}}, \tag{4.18}$$

$$\sigma_D^2(s) = \frac{\sigma_Y^2(s)}{\sum\limits_{\substack{t=1 \\ s_t=s}}^{T} u_t^2 + \frac{\sigma_Y^2(s)}{\epsilon(s)}}. \tag{4.19}$$

Prior and posterior of $\mathbf{D_0}$(s) - Assuming $D_0(s)$ is to be estimated for $s = 1, \dots, \kappa$. The prior associated to the conjugate posterior is $\mathcal{N}(D_0(s); \zeta_0(s), \epsilon_0(s))$. Thus, the posterior $p(D_0(s) \mid \Theta_{-D_0(s)}, s_{0:T}, x_{0:T}, y_{0:T})$ is proportional to the pdf of a normal distribution with mean $m_{D_0}(s)$ and variance $\sigma_{D_0}^2(s)$ (cf. Appendix B), where

$$m_{D_0} = \frac{\sum\limits_{t=1}^{T} \mathbb{1}(s_t = s) \left(y_t - C(s)x_t - D(s)u_t\right) + \frac{\zeta_0(s)}{\epsilon_0(s)}\sigma_Y^2(s)}{\sum\limits_{t=1}^{T} \mathbb{1}(s_t = s) + \frac{\sigma_Y^2(s)}{\epsilon_0}}, \tag{4.20}$$

$$\sigma_{D_0}^2 = \frac{\sigma_Y^2(s)}{\sum\limits_{t=1}^{T} \mathbb{1}(s_t = s) + \frac{\sigma_Y^2(s)}{\epsilon_0(s)}}. \tag{4.21}$$

Prior and posterior of $\sigma^2_{\mathbf{X}}(\mathbf{s})$ - Assuming $\sigma^2_X(s)$ is to be estimated for $s = 1, \ldots, \kappa$. The prior associated to the conjugate posterior is the inverse-gamma distribution $IG(\sigma^2_X;\, \varrho(s), \tau(s))$, with density

$$p(\sigma^2_X(s)) = \frac{\tau(s)^{\varrho(s)}}{\Gamma\left(\varrho(s)\right)\left(\sigma^2_X(s)\right)^{\varrho(s)+1}} \exp\left(\frac{-\tau(s)}{\sigma^2_X(s)}\right).$$

Hence, the posterior distribution of $\sigma^2_X(s)$ is (cf. Appendix B)

$$IG\left(\sigma^2_X(s);\, \varrho(s) + \frac{\sum_{t=1}^{T} \mathbb{1}(s_t = s)}{2}, \tau(s) + \sum_{t=1}^{T} \mathbb{1}(s_t = s)\left(x_t - x_{t-1} - B(s)u_t\Delta t\right)^2\right). \tag{4.22}$$

Prior and posterior of $\sigma^2_{\mathbf{Y}}(\mathbf{s})$ - Assuming $\sigma^2_Y(s)$ is to be estimated for $s = 1, \ldots, \kappa$. The prior associated to the conjugate posterior is the inverse-gamma distribution with shape parameter $\upsilon(s)$ and scale parameter $\psi(s)$). Hence, the posterior distribution of $\sigma^2_Y(s)$ is (cf. Appendix B)

$$IG\left(\sigma^2_Y(s);\, \upsilon(s) + \frac{\sum_{t=1}^{T} \mathbb{1}(s_t = s)}{2}, \psi(s) + \sum_{t=1}^{T} \mathbb{1}(s_t = s)\left(y_t - C(s)x_t - D(s)u_t - D_0(s)\right)^2\right). \tag{4.23}$$

4.3.2 Choice of the prior distribution of the Markov chain parameters

Cappé et al. [2005] prove that the Dirichlet prior is a conjugate distribution for the transition probability matrix P of the Markov chain $\{S_t\}$. Let p_{ij} denotes $p(s_t = j | s_{t-1} = i)$. Assume that each row of P has a Dirichlet prior distribution

$$(p_{i1}, \ldots, p_{i\kappa}) \sim \mathrm{Dir}_\kappa\left(\eta_1, \ldots, \eta_\kappa\right), \tag{4.24}$$

with the rows being independent *a priori*, and that the distribution Π of S_0 is either fixed or parameterized by a separate parameter. Then, given the Markov chain, the rows of P are conditionally independent and

$$p\left(p_{i1}, \ldots, p_{i\kappa} \mid s_{0:T}\right) \sim \mathrm{Dir}_\kappa\left(\eta_1 + n_{i1}, \ldots, \eta_\kappa + n_{i\kappa}\right), \tag{4.25}$$

where n_{ij} denotes the number of transitions from i to j in the sequence $s_{0:T}$:

$$n_{ij} = \sum_{t=1}^{T} \mathbb{1}(s_{t-1} = i, s_t = j). \tag{4.26}$$

A standard choice is to set equal η_j.

4.4 Sensitivity analysis using simulated data

In this section, the influence of the prior distributions and the choice of a sensible number of particles are discussed using a dataset generated according to the SMSSM (4.5)-(4.6) described in Chapter 3 with a number of hidden discrete states $\kappa = 3$. Table 4.1 recalls the parameters values of this model.

	B	C	D	D_0	σ_X^2	σ_Y^2
$s = 1$	-1.20	0.11	-350	375	10^{-4}	20
$s = 2$	-1.35	0.15	-400	380	10^{-4}	1
$s = 3$	-1.30	0.20	-420	385	10^{-4}	1

TABLE 4.1: Parameters values of the SMSSM with $\kappa = 3$ used to generate the simulated dataset

4.4.1 Influence of the prior distributions

The prior distribution often express subjective belief about a parameter without overwhelming precision. Hence, the mean of the prior is often specified by an experienced expert, whereas its variance gives the expert's confidence. Three types of prior distribution are tested hereafter:

1. informative: the expected value of each parameter is close to the true one, and the variance is relatively large (cf. Table 4.2);
2. non-informative: the states are not differentiated. For each parameter, we assign the same expected value to all states with a large variance (cf. Table 4.3);
3. improper: the expected value of each parameter is significantly different from the true one, and the variance is relatively small (cf. Table 4.4).

	$B : (\alpha, \beta)$	$C : (\gamma, \mu)$	$D : (\zeta, \epsilon)$	$D_0 : (\zeta_0, \epsilon_0)$	$\sigma_X^2 : (\varrho, \upsilon)$	$\sigma_Y^2 : (\tau, \psi)$
$s = 1$	$(-1.20,\ 1)$	$(0.10,\ 0.01)$	$(-350, 50)$	$(375,\ 50)$	$(3,\ 0.5)$	$(3,\ 1)$
$s = 2$	$(-1.40,\ 1)$	$(0.15,\ 0.01)$	$(-400, 50)$	$(380,\ 50)$	$(3,\ 0.5)$	$(3,\ 1)$
$s = 3$	$(-1.30,\ 1)$	$(0.20,\ 0.01)$	$(-420, 50)$	$(385,\ 50)$	$(3,\ 0.5)$	$(3,\ 1)$

TABLE 4.2: Informative priors for the particle Gibbs sampler

	$B : (\alpha, \beta)$	$C : (\gamma, \mu)$	$D : (\zeta, \epsilon)$	$D_0 : (\zeta_0, \epsilon_0)$	$\sigma_X^2 : (\varrho, \upsilon)$	$\sigma_Y^2 : (\tau, \psi)$
$s = 1$	$(-1.50,\ 1)$	$(0.20,\ 0.1)$	$(-380, 10^3)$	$(380, 10^3)$	$(3, 0.5)$	$(3, 1)$
$s = 2$	$(-1.50,\ 1)$	$(0.20,\ 0.1)$	$(-380, 10^3)$	$(380, 10^3)$	$(3, 0.5)$	$(3, 1)$
$s = 3$	$(-1.50,\ 1)$	$(0.20,\ 0.1)$	$(-380, 10^3)$	$(380, 10^3)$	$(3, 0.5)$	$(3, 1)$

TABLE 4.3: Non-informative priors for the particle Gibbs sampler

	$B:(\alpha,\beta)$	$C:(\gamma,\mu)$	$D:(\zeta,\epsilon)$	$D_0:(\zeta_0,\epsilon_0)$	$\sigma_X^2:(\varrho,\upsilon)$	$\sigma_Y^2:(\tau,\psi)$
$s=1$	$(-2.0,\ 0.5)$	$(1.0,\ 0.01)$	$(-450, 50)$	$(200,\ 50)$	$(3,\ 0.5)$	$(3,\ 1)$
$s=2$	$(-1.0,\ 0.5)$	$(1.2,\ 0.01)$	$(-500, 50)$	$(250,\ 50)$	$(3,\ 0.5)$	$(3,\ 1)$
$s=3$	$(-0.7,\ 0.5)$	$(0.7,\ 0.01)$	$(-550, 50)$	$(300,\ 50)$	$(3,\ 0.5)$	$(3,\ 1)$

TABLE 4.4: Improper priors for the particle Gibbs sampler

In all cases, the parameters of the Dirichlet distribution are set to 4 (Frühwirth-Schnatter [2011]):

$$\eta_s = 4, \text{ for } s = 1, 2, 3. \tag{4.27}$$

We run the particle Gibbs sampler with $N = 200$ for 4×10^4 iterations after an initial burn-in of 10^4 iterations. Figures 4.1, 4.2 and 4.3 show the estimates of the posterior densities for the parameters using the informative, non-informative and improper priors respectively. Most noticeably in the case of informative prior, it can be observed that the sampler has not explored the parameters support as thoroughly as in the case of the non-informative prior. Nevertheless in all cases, it appears that the parameter B is less well-estimated than the other parameters. This result is expected as B relates on the transition equation describing the evolution of the unknown continuous state X_t. Moreover using an improper prior, the results indicate that despite the long run the algorithm fails to efficiently estimate the unknown parameters.

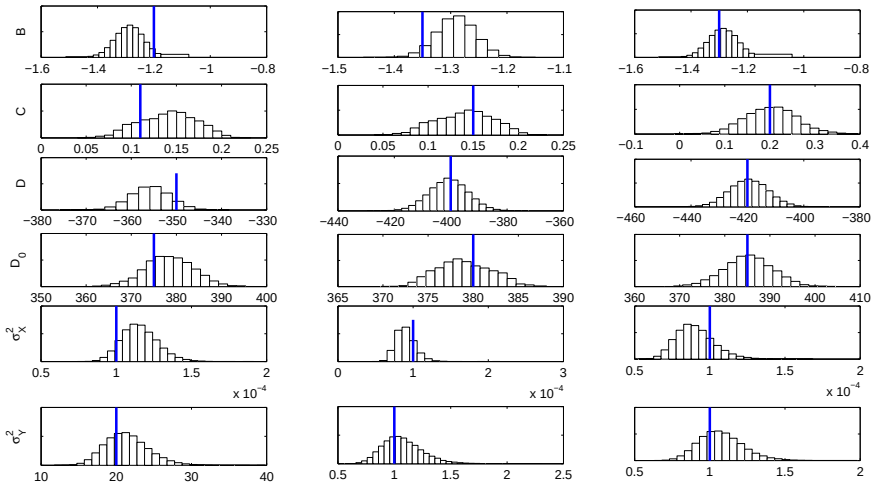

FIGURE 4.1: Histogram estimates of posterior densities for the SMSSM (4.5)-(4.6) using informative priors of Table 4.2

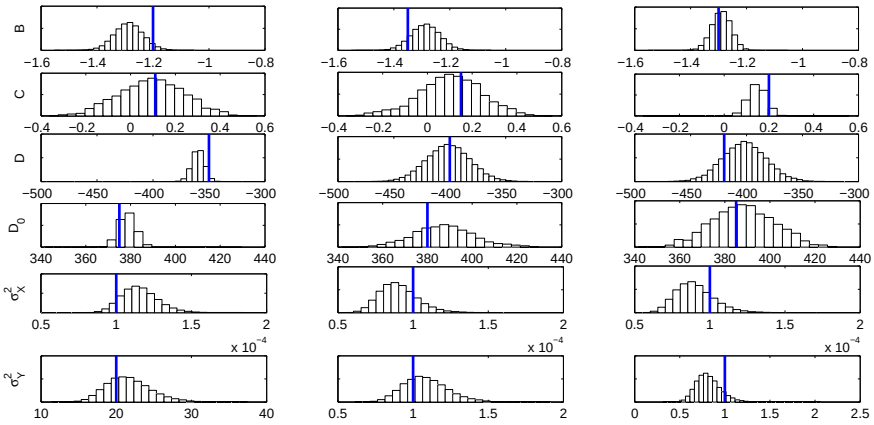

FIGURE 4.2: Histogram estimates of posterior densities for the SMSSM (4.5)-(4.6) using non-informative priors of Table 4.3

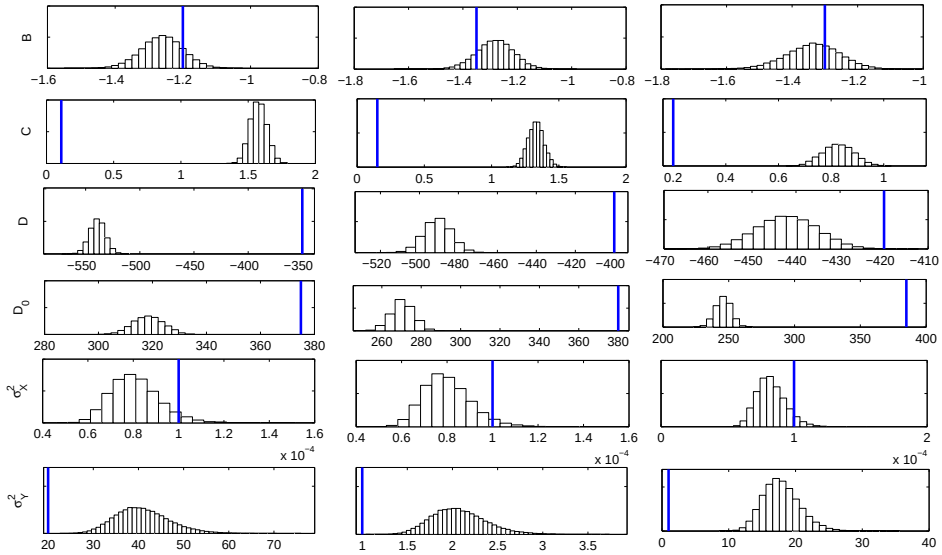

FIGURE 4.3: Histogram estimates of posterior densities for the SMSSM (4.5)-(4.6) using improper priors of Table 4.4

4.4.2 Choice of the number of particles

We present in Figure 4.4 the Auto-Correlation Function (ACF) for $\{B(s), C(s), D(s), D_0(s), \sigma_X^2(s), \sigma_Y^2(s)\}$, with $s = 1, 2, 3$, for the particle Gibbs sampler with various numbers of particles, namely $N = 10, 50, 100, 200$. We also run the algorithm for 4×10^4 iterations after an initial burn-in of 10^4 iterations. For B, D, σ_X^2 and σ_Y^2, it can be observed that the ACF drops sharply for $N = 10$. For C and D_0, the performance improves clearly when N increases; but it appears necessary to use more that 200 particles to make the ACF drop sharply.

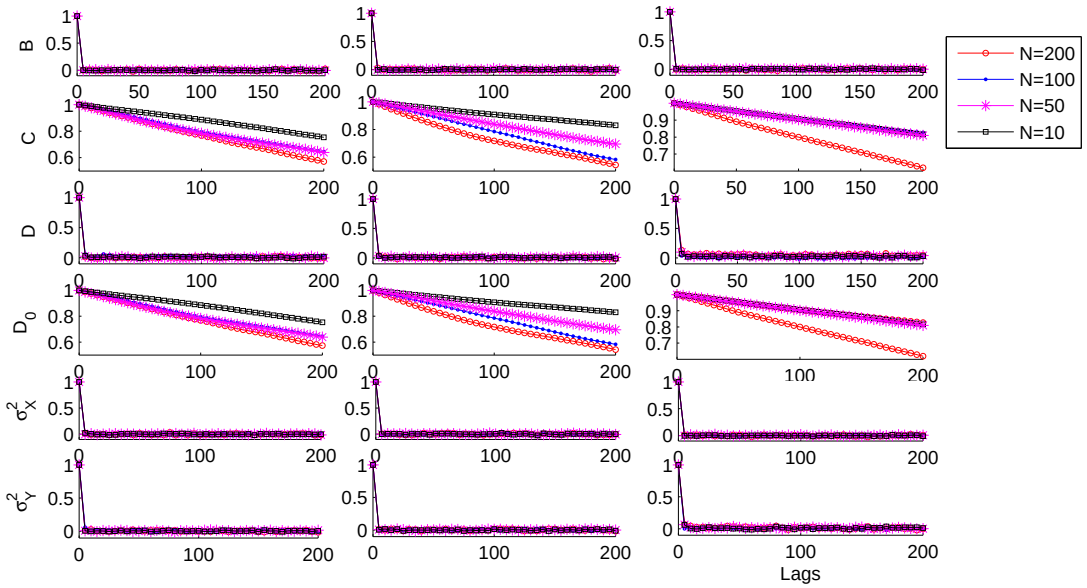

FIGURE 4.4: Auto-correlation function of the parameters B, C, D, D_0, σ_X^2 and σ_Y^2 respectively from top to bottom with $s = 1, 2$ and 3 from left to right: particle Gibbs sampler with $\circ$ 200 particles, $\bullet$ 100 particles, $*$ 50 particles and $\square$ 10 particles.

4.5 Conclusion

In this chapter, we have developed a particle Gibbs sampler to estimate the set of unknown parameters Θ of the considered SMSSM (4.5)-(4.6). Two major points should be emphasized concerning this algorithm. The first one is that at the k-th iteration, $s_{0:T}(k)$ is simulated by considering only the simulated $\Theta(k)$ and $y_{0:T}$, since $S_{0:T}$ and $X_{0:T}$ are strongly correlated. Given $\Theta(k)$, $y_{0:T}$ and $s_{0:T}(k)$, the sequence $x_{0:T}(k)$ is then calculated using a Kalman filter. The second point is that instead of sampling from $p_{\Theta(k)}(s_{0:T}|y_{0:T})$ which is a NP problem, the particle Gibbs sampler uses $\widehat{p}_{\Theta(k)}(s_{0:T}|y_{0:T})$ which is an estimation of $p_{\Theta(k)}(s_{0:T}|y_{0:T})$ using a particle filter.

To calibrate this algorithm, the impact of the prior distribution and the number of particles is particulary studied. The results show that this algorithm estimates accurately the unknown parameters even when using a non-informative prior for which the states are not differentiated and the variance is large. However, it appears necessary to use more than 200 particles, for a SMSSM with $\kappa = 3$, to ensure that the ACF for all parameters drop sharply. This would be too expensive especially in real contexts as the number of hidden discrete states is unknown and large values of κ might be tested.

In the next chapter, another approach is used to estimate the set of unknown parameters of the considered SMSSM, namely the maximum likelihood approach.

Chapter 5

Parameters inference using the maximum likelihood approach

L'objectif du travail présenté dans ce chapitre est d'estimer l'ensemble de paramètres inconnus Θ par une approche du type maximum de vraisemblance, conditionnellement à une base d'apprentissage $\{y_{0:T}, u_{0:T}\}$ et en considérant que les deux séquences d'états continus $X_{0:T}$ et discrets $S_{0:T}$ sont inconnues :

$$\widehat{\Theta}_{\mathrm{ML}} = \arg\max_{\Theta} \log p_{\Theta}(y_{0:T}). \tag{5.1}$$

La Section 5.1 présente le principe de cette approche. Le calcul exact de cet estimateur est difficile du fait que $p_{\Theta}(y_{0:T})$ est inconnu. Nous avons alors recours à l'algorithme EM (Dempster et al. [1977]) spécialement adapté aux modèles à structure latente (Rabiner [1989] pour les HMMs et Shumway and Stoffer [1982] pour les SSMs). Cependant, appliqué directement pour des SMSSMs, cet algorithme nécessite le calcul des espérances conditionnelles $\mathbb{E}_{Y_{0:T},\Theta}[X_{0:T}]$ et $\mathbb{E}_{Y_{0:T},\Theta}[S_{0:T}]$, qui est un problème NP-complet comme nous l'avons vu dans le Chapitre 3 (Tugnait [1982]). Plus précisément, pour un SMSSM à κ états cachés et une base d'apprentissage de taille $T+1$, l'étape E de l'algorithme EM nécessite une somme sur toutes les séquences possibles $s_{0:T}$, soit κ^{T+1} séquences :

$$\mathcal{Q}(\Theta, \Theta') = \sum_{s_{0:T} \in \mathcal{S}} p_{\Theta'}(s_{0:T} \mid y_{0:T}) \mathbb{E}_{S_{0:T}, Y_{0:T}, \Theta'}[\log p_{\Theta}(X_{0:T}, S_{0:T}, Y_{0:T})], \tag{5.2}$$

où $\mathcal{S} = \{1, \ldots, \kappa\}^{T+1}$ est l'ensemble des séquences $s_{0:T}$ possibles. Les équations résultant de la maximisation directe de l'équation (5.2) présentées dans (5.14)-(5.20) font également intervenir une somme sur les κ^{T+1} séquences.

Pour surmonter cette difficulté, une solution consiste à approcher la somme de (5.2) sur les κ^{T+1} séquences par une somme sur N séquences, appelées des *particules*, simulées

par une méthode de Monte-Carlo :

$$\widehat{\mathcal{Q}}(\Theta, \Theta') = \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{s^i_{0:T}, Y_{0:T}, \Theta'}[\log p_\Theta(X_{0:T}, s^i_{0:T}, Y_{0:T})], \tag{5.3}$$

où $s^i_{0:T} \overset{iid}{\sim} p_{\Theta'}(s_{0:T} \mid y_{0:T})$. Cette version stochastique de l'algorithme EM est appelée Monte-Carlo EM (MCEM) (Wei and Tanner [1990]). Cependant, le tirage selon $p_{\Theta'}(s_{0:T} \mid y_{0:T})$ n'étant pas évident, nous réutilisons la technique d'échantillonnage d'importance séquentiel du Chapitre 3. Ainsi, l'espérance conditionnelle est estimée par :

$$\widehat{\mathcal{Q}}(\Theta, \Theta') = \sum_{i=1}^{N} w^i_T \, \mathbb{E}_{s^i_{0:T}, Y_{0:T}, \Theta'}[\log p_\Theta(X_{0:T}, s^i_{0:T}, Y_{0:T})], \tag{5.4}$$

avec $s^i_{0:T} \overset{iid}{\sim} q_{\Theta'}(s_{0:T} \mid y_{0:T})$, $q_{\Theta'}(s_{0:T} \mid y_{0:T})$ la loi d'importance et w^i_T les poids d'importance. L'application de cet algorithme pour les SMSSMs est détaillée dans la Section 5.3.

D'autre part, comme nous l'avons présenté dans le Chapitre 3, le SMSSM n'est pas nécessairement identifiable. Dans ce cas, les méthodes du type maximum de vraisemblance pourraient ne pas estimer convenablement Θ. En effet, si le modèle n'est pas identifiable, pour un jeu d'observations donné $y_{0:T}$, deux ensembles de paramètres pourraient donner la même vraisemblance, et ainsi il est impossible de distinguer ces deux ensembles candidats en se basant uniquement sur le jeu d'observations donné. Par conséquent, il est courant d'introduire un terme de pénalisation assurant l'identifiabilité des paramètres inconnus (Casella and Berger [2002]). Dans ce chapitre, nous avons développé deux versions pénalisées de l'algorithme MCEM. La première est pénalisée par les contraintes d'identifiabilité (Proposition 1, Section 3.4). Quant à la seconde, elle est pénalisée par une loi *a priori* sur Θ (Green [1990]).

Pour le premier type de pénalisation, l'étape de maximisation consiste à chercher l'ensemble de paramètres qui maximise l'espérance conditionnelle estimée $\widehat{\mathcal{Q}}(\Theta, \Theta')$ pénalisée par les contraintes d'identifiabilité (cf. Proposition 1, Section 3.4), en y ajoutant aussi les contraintes sur la matrice de transition P.

$$\begin{aligned} y_{t_0} &= C(s)x_{t_0} + D(s)u_{t_0} + D_0(s), \\ \sum_{j=1}^{\kappa} P(s,j) &= 1, \end{aligned}$$

pour $s = 1, \ldots, \kappa$. Par conséquent, le lagrangien correspondant à cette maximisation est de la forme suivante :

$$\mathcal{L}'(\Theta, \lambda, \mu) = \widehat{\mathcal{Q}}(\Theta, \Theta') + \sum_{i=1}^{\kappa} \lambda_i \left(1 - \sum_{j=1}^{\kappa} P(i,j)\right) + \sum_{i=1}^{\kappa} \mu_i \left(y_{t_0} - C(i)x_{t_0} - D(i)u_{t_0} - D_0(i)\right), \tag{5.5}$$

où λ_i et μ_i sont les multiplicateurs de Lagrange. La Section 5.4.1 fournit les détails de calcul de cet algorithme.

Le deuxième type de pénalisation correspond à la recherche du maximum *a posteriori* (Green [1990]). L'étape de maximisation consiste à chercher l'ensemble de paramètres qui maximise l'espérance conditionnelle estimée $\widehat{\mathcal{Q}}(\Theta, \Theta')$ pénalisée par $p(\Theta)$ une loi *a priori* sur Θ :

$$\frac{\partial}{\partial \Theta} \widehat{\mathcal{Q}}(\Theta, \Theta') + \frac{\partial}{\partial \Theta} p(\Theta) = 0. \tag{5.6}$$

Dans le cas de SMSSM, cette équation induit la résolution d'un système d'équations non linéaires. Pour éviter cette difficulté, une légère modification est introduite à cette équation :

$$\frac{\partial}{\partial \Theta} \widehat{\mathcal{Q}}(\Theta, \Theta') + \frac{\partial}{\partial \Theta} p(\Theta)|_{\Theta=\Theta'} = 0. \tag{5.7}$$

La seule différence entre ces deux équations réside dans le fait que la dérivée de la loi *a priori* $p(\Theta)$ est évaluée en Θ'. Cette version modifiée de l'algorithme EM développée par Green [1990] est appelé *one step late*. Les détails concernant cet algorithme ainsi que le choix de la loi *a priori* sont présentés à la Section 5.4.2.

Nous avons calibré ces deux algorithmes pénalisés en se basant sur une base de données générée à partir d'un SMSSM à $\kappa = 3$. Plus précisément, le calibrage concerne l'influence de la stratégie d'initialisation (cf. Section 5.5.1.1), du nombre de particules (cf. Section 5.5.1.2) et du choix de la loi *a priori* (cf. Section 5.5.2).

L'estimateur donné par l'algorithme EM est influencé par l'initialisation des paramètres du modèle. Selon cette initialisation, il peut converger vers un maxima local. Pour éviter ce problème, cet algorithme est généralement lancé plusieurs fois avec différentes initialisations. La solution ayant le maximum de vraisemblance est ensuite retenue. Dans cette étude, nous avons testé deux types d'initialisation : indépendante et emboîtée. L'initialisation indépendante consiste à initialiser chaque paramètre indépendamment des autres selon une loi normale de moyenne donnée par un expert et d'une variance relativement grande pour assurer une initialisation différente à chaque lancement de l'algorithme. L'initialisation emboîtée, récemment développée par Baudry and Celeux [2015], est adaptée aux modèles de mélange. Elle consiste à initialiser un modèle à $\kappa + 1$ états cachés à partir des résultats d'estimation des paramètres du modèle à κ états cachés. Ainsi, pour le modèle à κ états cachés, l'état ayant la plus grande entropie est divisée en deux états (cf. Figure 5.2). Les résultats montrent que l'initialisation emboîtée est plus adaptée aux SMSSMs que l'initialisation indépendante. En effet, dans les modèles de mélange lorsque le nombre de composantes augmente, le maximum de vraisemblance (*Maximum Likelihood* - ML) doit aussi augmenter dans la mesure où un modèle à $\kappa + 1$ composantes peut être équivalent à un modèle à κ composantes en divisant l'une des composantes en deux. Cependant, en utilisant l'initialisation indépendante, le ML estimé diminue parfois lorsque κ augmente (cf. Figure 5.1). Un tel comportement n'est pas optimal.

L'algorithme MCEM dépend également du nombre de particules N utilisées pour approcher la somme sur les κ^{T+1} séquences (Biscarat [1994], Celeux et al. [1996]). Pour étudier l'influence de N sur le ML estimé, nous avons lancé l'algorithme MCEM 20 fois avec une initialisation identique des paramètres pour différentes valeurs de N. Dans ce cas, la variabilité du ML estimé dépend uniquement de l'approximation par Monte-Carlo, soit de N. Ensuite, nous avons lancé cet algorithme 20 fois avec différentes initialisations et pour différentes valeurs de N. La variabilité du ML estimé dépend, dans ce cas, de l'initialisation des paramètres et du choix de N. Les résultats montrent que lorsque le nombre de particules est bien choisi ($N \simeq 100$ pour $\kappa = 3$ et $T = 500$), la dispersion des MLs estimés est relativement faible (cf. Figures 5.5 et 5.6).

Enfin, pour l'algorithme MCEM pénalisé par une loi *a priori* sur Θ, l'influence du choix de cette loi est étudiée. Il convient de noter que cette loi doit être bien choisie pour assurer l'identifiabilité des paramètres inconnus. De ce fait, si une loi uniforme est utilisée par exemple, le problème d'identifiabilité persiste et l'algorithme pourrait ne pas estimer efficacement Θ. Dans cette étude, nous choisissons des lois *a priori* gaussiennes pour $B(s)$, $C(s)$, $D(s)$ et $D_0(s)$, et des inverse-gamma pour les variances $\sigma^2_X(s)$ et $\sigma^2_Y(s)$. Nous avons testé trois configurations de ces lois *a priori* :

1. très informative : les moyennes des lois sont égales aux vrais paramètres et leur variance est relativement faible (cf. Table 5.2),
2. informative : les moyennes des lois sont égales aux vrais paramètres mais leur variance est relativement grande (cf. Table 5.3),
3. non-informative : les états ne sont pas différenciés. Pour chaque paramètre, nous assignons la même moyenne à tous les états avec une variance relativement grande (cf. Table 5.4).

Les résultats montrent que la pénalisation de l'algorithme MCEM par une loi *a priori* permet de résoudre le problème d'identifiabilité et que les paramètres sont estimés convenablement dans les trois configurations de cette loi (cf. Figures 5.9, 5.10 et 5.11).

Pour conclure, en utilisant une initialisation emboîtée et un nombre de particules approprié, les deux versions pénalisées de l'algorithme MCEM estiment convenablement l'ensemble de paramètres inconnus Θ. Toutefois, il convient de noter que la pénalisation par une loi *a priori* nécessite un grand nombre d'hyperparamètres (soit $12 \cdot \kappa$ paramètres) qui doivent être correctement choisis pour éviter le problème d'identifiabilité. Ce choix peut s'avérer compliqué, surtout lorsque ces paramètres sont dépourvus de toute signification physique. Quand ils en ont une, les deux algorithmes peuvent être appliqués et l'ensemble estimé ayant la plus grande vraisemblance sera ensuite retenu.

Contents

In the previous chapter, we have estimated the set of unknown parameters Θ using a Bayesian approach. The aim of this chapter is to estimate Θ of the considered SMSSM (3.22)-(3.23) using the ML approach. Since a direct evaluation of the ML estimate is analytically intractable, we resort to the well-known *EM algorithm* (Dempster et al. [1977]). This is commonly known to be adapted to latent models such as SMSSMs.

However, the EM algorithm is based on the computation of the conditional expectation of two latent variables X_t and S_t, $\mathbb{E}_{Y_{0:T},\Theta}[X_t]$ and $\mathbb{E}_{Y_{0:T},\Theta}[S_t]$, which is a NP-hard problem in SMSSMs (Tugnait [1982]). Precisely, when the number of hidden discrete states is equal to κ and the size of the learning dataset is equal to $T+1$, the expectation step of the EM algorithm requires a summation over up to the κ^{T+1} possible sequences $s_{0:T}$. To overcome this computational difficulty, we use a stochastic version of the EM algorithm called *Monte Carlo EM-algorithm* (MCEM) (Tanner [1996], Wei and Tanner [1990]). The summation over up to the κ^{T+1} sequences is thus approximated by only N sequences simulated by a Monte Carlo method.

In addition, as shown in Chapter 3, the SMSSM (3.22)-(3.23) is not necessarily identifiable. Thus, ML methods could fail to properly estimate the model parameters without considering a penalization term. To overcome this numerical issue, we develop two penalized versions of the MCEM algorithm. The first one is penalized with the identifiability constraints of Proposition 1 (Section 3.4); whereas the second one is penalized with a prior probability distribution on the set of unknown parameters Θ (cf. Green [1990]).

In order to calibrate these two penalized algorithms, a sensitivity analysis is performed based on simulated data. Specifically, the influence of the initialization procedure, the number of particles, and the prior distribution is thoroughly studied.

5.1 Principle of maximum likelihood parameters inference

Let us consider a learning dataset $\{y_{0:T}, u_{0:T}\}$ where $y_{0:T}$ is observed and $u_{0:T}$ is an input. Here, both latent variables $X_{0:T}$ and $S_{0:T}$ are unknown. The maximum likelihood

estimate (MLE) $\widehat{\Theta}_{\text{ML}}$ of the set of unknown parameters Θ is written as follows:

$$\widehat{\Theta}_{\text{ML}} = \arg\max_{\Theta} \log p_{\Theta}(y_{0:T}), \tag{5.8}$$

where $p_{\Theta}(y_{0:T})$ is the marginal likelihood. For a SMSSM, the marginal likelihood is given by

$$p_{\Theta}(y_{0:T}) = \sum_{s_{0:T} \in \mathcal{S}} \int_{x_{0:T}} p_{\Theta}(x_{0:T}, s_{0:T}, y_{0:T}) dx_{0:T}, \tag{5.9}$$

where $\mathcal{S} = \{1, \ldots, \kappa\}^{T+1}$ is the set of all possible sequences $s_{0:T}$, and $p_{\Theta}(x_{0:T}, s_{0:T}, y_{0:T})$ is the completed-likelihood given by (3.32):

$$\begin{aligned} p_{\Theta}(x_{0:T}, s_{0:T}, y_{0:T}) =& p_{\Theta}(y_0 \mid x_0, s_0) \cdot p_{\Theta}(x_0 \mid s_0) \cdot p_{\Theta}(s_0) \\ & \cdot \prod_{t=1}^{T} p_{\Theta}(y_t \mid s_t, x_t) \cdot p_{\Theta}(x_t \mid s_t, x_{t-1}) \cdot p_{\Theta}(s_t \mid s_{t-1}). \end{aligned}$$

It is clear that a direct evaluation of (5.9) is analytically intractable, since it involves a summation over up to κ^{T+1} Markov sequences. Therefore, the EM algorithm developed by Dempster et al. [1977] is used to iteratively approximate the MLE $\widehat{\Theta}_{ML}$. This algorithm is a popular and often efficient approach to ML estimation for incomplete data problems as in the SMSSMs context where the sequence of observations $y_{0:T}$ and inputs $u_{0:T}$ are known but their corresponding $x_{0:T}$ and $s_{0:T}$ are unknown.

The EM iteration alternates between an expectation step (E-step), which calculates the conditional expectation of the log-likelihood for the complete data under the current set of parameters, and a maximization step (M-step) which computes the parameters maximizing the conditional expectation found on the E-step. These new estimated parameters are then used in the E-step of the next iteration.

5.2 EM algorithm for SMSSMs

Instead of $\log p_{\Theta}(y_{0:T})$, the EM algorithm considers the conditional expectation of the complete log-likelihood:

$$\mathcal{Q}(\Theta, \Theta') = \mathbb{E}_{Y_{0:T}, \Theta'} \left[\log p_{\Theta} \left(X_{0:T}, S_{0:T}, Y_{0:T} \right) \right], \tag{5.10}$$

where Θ' is the current estimated parameters. The substitution by the conditional expection is theoretically justified by the Jensen's inequality showing that

$$\log p_{\Theta}(y_{0:T}) - \log p_{\Theta'}(y_{0:T}) \geq \mathcal{Q}(\Theta, \Theta') - \mathcal{Q}(\Theta', \Theta') \geq 0. \tag{5.11}$$

Hence, maximizing $\mathcal{Q}(\Theta, \Theta')$ with respect to Θ corresponds to maximizing $\log p_{\Theta}(y_{0:T})$. Accordingly, the MLE is iteratively approached by replacing the current estimate Θ' by

Θ maximizing $\mathcal{Q}(\Theta, \Theta')$. This iteration is repeated until the convergence is apparent (cf. Dempster et al. [1977]).

5.2.1 Expectation Step

Based on the interaction (3.31) between the two latent variables S_t and X_t, the conditional expectation $\mathcal{Q}(\Theta, \Theta')$ can be written as follows:

$$\begin{aligned}\mathcal{Q}(\Theta, \Theta') &= \sum_{s_{0:T} \in \mathcal{S}} p_{\Theta'}(s_{0:T} \mid y_{0:T}) \mathbb{E}_{S_{0:T}, Y_{0:T}, \Theta'}[\log p_\Theta(X_{0:T}, S_{0:T}, Y_{0:T})] \\ &= \sum_{s_{0:T} \in \mathcal{S}} p_{\Theta'}(s_{0:T} \mid y_{0:T}) \{\mathbb{E}_{S_{0:T}, Y_{0:T}, \Theta'}[\log p_\Theta(S_{0:T})] + \mathbb{E}_{S_{0:T}, Y_{0:T}, \Theta'}[\log p_\Theta(X_{0:T}|S_{0:T})] \\ &\quad + \mathbb{E}_{S_{0:T}, Y_{0:T}, \Theta'}[\log p_\Theta(Y_{0:T}|S_{0:T}, X_{0:T})]\} . \end{aligned} \tag{5.12}$$

Since $p_\Theta(x_{0:T}|s_{0:T})$ and $p_\Theta(y_{0:T}|s_{0:T}, x_{0:T})$ are Gaussian distributions, the conditional expectations of their logarithm are evaluated using the forward and backward techniques of a Kalman filter (Shumway and Stoffer [1982]).

5.2.2 Maximization Step

Our aim is to maximize $\mathcal{Q}(\Theta, \Theta')$ w.r.t Θ, under the constraints on the transition matrix $\sum_{j=1}^{\kappa} P(i,j) = 1$ $(1 \leq i \leq \kappa)$. The Lagrangian associated to these constraints is given by

$$\mathcal{L}(\Theta, \lambda, \mu) = \mathcal{Q}(\Theta, \Theta') + \sum_{i=1}^{\kappa} \lambda_i \left(1 - \sum_{j=1}^{\kappa} P(i,j)\right), \tag{5.13}$$

where λ_i, $1 \leq i \leq \kappa$, are the Lagrangian coefficients.

The transition matrix P is estimated by solving $\partial \mathcal{L}(\Theta, \lambda, \mu)/\partial P = 0$. This leads to the following equation:

$$\widehat{P}(i,j) = \frac{\sum_{s_{0:T} \in \mathcal{S}} \sum_{t=1}^{T} \mathbb{1}(s_{t-1} = i, s_t = j) p_{\Theta'}(s_{t-1} = i\,,\, s_t = j \,|\, y_{0:T})}{\sum_{s_{0:T} \in \mathcal{S}} \sum_{t=1}^{T} \mathbb{1}(s_{t-1} = i) p_{\Theta'}(s_{t-1} = i \,|\, y_{0:T})}, \tag{5.14}$$

where $1 \leq i, j \leq \kappa$.

Let us consider $\widehat{x}_{t|T} = \mathbb{E}_{s_{0:T}, y_{0:T}, \Theta'}[X_t]$, and $\widehat{x}_{t,r|T} = \mathbb{E}_{s_{0:T}, y_{0:T}, \Theta'}[X_t \cdot X_r]$ (note that these quantities depend on $s_{0:T}$). Solving $\partial \mathcal{L}(\Theta, \lambda, \mu)/\partial \Gamma = 0$ leads to the following system of

$6 \cdot \kappa$ differential equations:

$$\sum_{s_{0:T}\in\mathcal{S}} \sum_{t:s_t=s} u_t \left[\widehat{x}_{t-1|T} + u_t \Delta t B(s) - \widehat{x}_{t|T}\right] \Pi_{t|T}(s) = 0, \tag{5.15}$$

$$\sum_{s_{0:T}\in\mathcal{S}} \sum_{t:s_t=s} \left[C(s)\widehat{x}_{t,t|T} + D(s)\widehat{x}_{t|T}u_t + D_0(s)\widehat{x}_{t|T} - \widehat{x}_{t|T}y_t\right] \Pi_{t|T}(s) = 0, \tag{5.16}$$

$$\sum_{s_{0:T}\in\mathcal{S}} \sum_{t:s_t=s} u_t \left[C(s)\widehat{x}_{t|T} + u_t D(s) + D_0(s) - y_t\right] \Pi_{t|T}(s) = 0, \tag{5.17}$$

$$\sum_{s_{0:T}\in\mathcal{S}} \sum_{t:s_t=s} \left[C(s)\widehat{x}_{t|T} + D(s)u_t + D_0(s) - y_t\right] \Pi_{t|T}(s) = 0, \tag{5.18}$$

$$\begin{aligned}\sum_{s_{0:T}\in\mathcal{S}} \sum_{t:s_t=s} &\left[\sigma_X^2(s) + 2\widehat{x}_{t,t-1|T} - 2\widehat{x}_{t-1|T}B(s)u_t\Delta t - \widehat{x}_{t,t|T} - \widehat{x}_{t-1,t-1|T} - B(s)^2u_t^2\Delta t^2\right.\\ &\left.+2B(s)u_t\Delta t\widehat{x}_{t|T}\right] \Pi_{t|T}(s) = 0,\end{aligned} \tag{5.19}$$

$$\begin{aligned}\sum_{s_{0:T}\in\mathcal{S}} \sum_{t:s_t=s} &\left[\sigma_Y^2(s) - y_t^2 - C^2(s)\widehat{x}_{t,t|T} - D(s)^2u_t^2 - D_0(s)^2 + 2y_tD_0(s)\right.\\ &\left.+2C(s)\widehat{x}_{t|T}\left(y_t - D(s)u_t - D_0(s)\right) + 2D(s)u_t\left(y_t - D_0(s)\right)\right] \Pi_{t|T}(s) = 0,\end{aligned} \tag{5.20}$$

where $t = 1, \ldots, T$, $1 \leq s \leq \kappa$, $\Pi_{t|T}(s) = p_{\Theta'}(s_t = s \mid y_{0:T})$.

However, an exact computation of equations (5.14)-(5.20) relies on the computation of κ^{T+1} values of $s_{0:T}$; i.e., all possible sequences $s_{0:T}$. Even for modest values of T and κ, this requires a prohibitive computational cost.

5.3 Monte Carlo-EM algorithm for SMSSMs

To overcome this computational difficulty, we resort to the particle filters method to approximate the conditional expectation $\mathcal{Q}(\Theta, \Theta')$ given by (5.12). This stochastic version of the EM algorithm is referred to as MCEM algorithm (Tanner [1996], Wei and Tanner [1990]).

A basic idea would consist in using a set of N iid particles $\{s_{0:T}^i\}_{i=1}^N$ simulated from $p_{\Theta'}(s_{0:T} \mid y_{0:T})$ such that the conditional expectation $\mathcal{Q}(\Theta, \Theta')$ given by (5.12) is estimated as follows:

$$\widehat{\mathcal{Q}}(\Theta, \Theta') = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{s_{0:T}^i, Y_{0:T}, \Theta'}[\log p_\Theta(X_{0:T}, s_{0:T}^i, Y_{0:T})]. \tag{5.21}$$

5.3.1 Importance sampling for the MCEM algorithm

As shown is Section 3.5.1, it is not possible to sample directly from the target distribution $p_{\Theta'}(s_{0:T} \mid y_{0:T})$. Therefore, we use the importance sampling method to do it. Instead of $p_{\Theta'}(s_{0:T} \mid y_{0:T})$, we use an arbitrary instrumental distribution $q_{\Theta'}(s_{0:T} \mid y_{0:T})$ from which it is easy to obtain samples.

The conditional expectation $\mathcal{Q}(\Theta, \Theta')$ is thus estimated as follows:

$$\widehat{\mathcal{Q}}(\Theta, \Theta') = \sum_{i=1}^{N} \widehat{w}_T^i \mathbb{E}_{s_{0:T}^i, Y_{0:T}, \Theta'}[\log p_\Theta(X_{0:T}, s_{0:T}^i, Y_{0:T})], \tag{5.22}$$

where $\widehat{w}_T^i$ is the normalized version of the importance weight given by (3.55), and the sequences $s_{0:T}^i$ is obtained as in Section 3.5.1 (cf. Algorithm 1). Accordingly, the equations (5.14)-(5.20) are estimated as follows:

$$\sum_{i=1}^{N} \sum_{t:s_t^i=s} u_t \left[\widehat{x}_{t-1|T}^i + u_t \Delta t B(s) - \widehat{x}_{t|T}^i\right] \widehat{w}_T^i = 0, \tag{5.23}$$

$$\sum_{i=1}^{N} \sum_{t:s_t^i=s} \left[C(s)\widehat{x}_{t,t|T}^i + D(s)\widehat{x}_{t|T}^i u_t + D_0(s)\widehat{x}_{t|T}^i - \widehat{x}_{t|T}^i y_t\right] \widehat{w}_T^i = 0, \tag{5.24}$$

$$\sum_{i=1}^{N} \sum_{t:s_t^i=s} u_t \left[C(s)\widehat{x}_{t|T}^i + u_t D(s) + D_0(s) - y_t\right] \widehat{w}_T^i = 0, \tag{5.25}$$

$$\sum_{i=1}^{N} \sum_{t:s_t^i=s} \left[C(s)\widehat{x}_{t|T}^i + D(s)u_t + D_0(s) - y_t\right] \widehat{w}_T^i = 0, \tag{5.26}$$

$$\begin{aligned}\sum_{i=1}^{N} \sum_{t:s_t^i=s} \Big[&\sigma_X^2(s) + 2\widehat{x}_{t,t-1|T}^i - 2\widehat{x}_{t-1|T}^i B(s) u_t \Delta t - \widehat{x}_{t,t|T}^i - \widehat{x}_{t-1,t-1|T}^i - B(s)^2 u_t^2 \Delta t^2 \\ &+2B(s)u_t \Delta t \widehat{x}_{t|T}^i\Big] \widehat{w}_T^i = 0,\end{aligned} \tag{5.27}$$

$$\begin{aligned}\sum_{i=1}^{N} \sum_{t:s_t^i=s} \Big[&\sigma_Y^2(s) - y_t^2 - C^2(s)\widehat{x}_{t,t|T}^i - D(s)^2 u_t^2 - D_0(s)^2 + 2y_t D_0(s) \\ &+2C(s)\widehat{x}_{t|T}^i \left(y_t - D(s)u_t - D_0(s)\right) + 2D(s)u_t \left(y_t - D_0(s)\right)\Big] \widehat{w}_T^i = 0.\end{aligned} \tag{5.28}$$

The k-th iteration of the proposed method for estimation the unknown parameters of the SMSSM (3.22)-(3.23) using the presented MCEM algorithm is described in Algorithm 3. It has a computational complexity equal to $\mathcal{O}(N\,T)$.

5.3.2 Approximation of the marginal likelihood

This particle filter provides an approximation of

$$\begin{aligned} p_\Theta(y_t \mid y_{0:t-1}) &= \sum_{s_{0:t-1} \in \mathcal{S}} p_\Theta(y_t, s_{0:t-1} \mid y_{0:t-1}) \\ &= \sum_{s_{0:t-1} \in \mathcal{S}} p_\Theta(y_t \mid s_{0:t-1}, y_{0:t-1}) \cdot p_\Theta(s_{0:t-1} \mid y_{0:t-1}) \\ &\approx \sum_{i=1}^{N} \widehat{w}_{t-1}^i \cdot p_\Theta(y_t \mid s_{0:t-1}^i, y_{0:t-1}), \end{aligned} \tag{5.29}$$

Algorithm 3 k-th iteration of the MCEM algorithm

Input $\leftarrow \Theta = \Theta^{(k)}, y_{0:T}, u_{0:T}$

Init $\leftarrow s_0^i \overset{iid}{\sim} \Pi \ (\forall i = 1:N)$

for $t = 1:T$ and $i = 1:N$ **do**

1. Sample $s_t^i \overset{iid}{\sim} p_\Theta(s_t \mid s_{0:t-1}^i, y_{0:t})$, thus $s_{0:t}^i = \left(s_{0:t-1}^i, s_t^i\right)$
2. Given $s_{0:t}^i$ and $y_{0:t}$, calculate $\widehat{x}_{t|t}^i \overset{\triangle}{=} \mathbb{E}_{s_{0:t}^i, y_{0:t}, \Theta}[X_t]$ using a Kalman filter
3. Calculate importance weights w_t^i
4. Selection step: sample $s_{0:t}^i \overset{iid}{\sim} \sum_{i=1}^{N} w_t^i \delta(s_{0:t} - s_{0:t}^i)$, where $\delta(\cdot)$ is a Dirac function with mass at zero

end for
for $t = 1:T$ and $i = 1:N$ **do**

Calculate $\widehat{x}_{t|T}^i$ and $\widehat{x}_{t,t|T}^i$ using a Kalman smoother

end for
Estimate $\Theta^{(k+1)}$ by solving $\partial \widehat{\mathcal{Q}}(\Theta, \Theta')/\partial\Theta = 0$

with

$$p_\Theta(y_t \mid s_{0:t-1}^i, y_{0:t-1}) = \sum_{s=1}^{\kappa} p_\Theta(s_t = s | s_{t-1}^i) p_\Theta(y_t \mid s_{0:t-1}^i, s_t = s, y_{0:t-1}), \tag{5.30}$$

where $p_\Theta(y_t \mid s_{0:t-1}^i, s_t = s, y_{0:t-1})$ is computed using a Kalman filter. An estimation of the marginal likelihood $p_\Theta(y_{0:T})$ is then given by

$$\widehat{p}_\Theta(y_{0:T}) = \widehat{p}_\Theta(y_0) \prod_{t=1}^{T} \widehat{p}_\Theta(y_t \mid y_{0:t-1}). \tag{5.31}$$

This estimation is of particular interest in the sequel.

However, since the SMSSM (3.22)-(3.23) is not necessary identifiable, ML methods could fail to efficiently estimate the unknown model parameters without a penalization term (Casella and Berger [2002]).

5.4 Penalized Monte Carlo-EM algorithms for SMSSMs

To overcome this numerical issue, we develop two penalized versions of the MCEM algorithm. The first one is penalized with the identifiability constraints of the Proposition 1 (Section 3.4); whereas the second one is penalized with a prior distribution on the set of unknown parameters Θ (Green [1990]).

5.4.1 Penalization with identifiability constraints

In the maximization step of the EM algorithm, we consider the following constraints:

1. the constraints on the transition matrix $\sum_{j=1}^{\kappa} P(i,j) = 1$ $(1 \leq i \leq \kappa)$,
2. the constraints ensuring the identifiability of the model (cf. Proposition 1, Section 3.4):

$$y_{t_0} = C(s)x_{t_0} + D(s)u_{t_0} + D_0(s), \text{ for any hidden state } s = 1, \ldots, \kappa.$$

The Lagrangian associated to these constraints is as follows:

$$\begin{aligned}\mathcal{L}'(\Theta, \lambda, \mu) =& \widehat{\mathcal{Q}}(\Theta, \Theta') + \sum_{i=1}^{\kappa} \lambda_i \left(1 - \sum_{j=1}^{\kappa} P(i,j)\right) \\ &+ \sum_{i=1}^{\kappa} \mu_i \left(y_{t_0} - C(i)x_{t_0} - D(i)u_{t_0} - D_0(i)\right), \end{aligned} \tag{5.32}$$

where λ_i and μ_i, $1 \leq i \leq \kappa$, are the Lagrangian coefficients.

Solving $\partial \mathcal{L}'(\Theta, \lambda, \mu)/\partial \Gamma = 0$ leads to the same system of equations of Section 5.3. Only the differential equations relating to the observation equation, namely (5.24)-(5.26), change:

$$\sum_{i=1}^{N} \sum_{t:s_t^i = s} \left[C(s)\widehat{x}_{t,t|T}^i + D(s)\widehat{x}_{t|T}^i u_t + D_0(s)\widehat{x}_{t|T}^i - \widehat{x}_{t|T}^i y_t\right] \widehat{w}_T^i + \mu_s \sigma_Y^2(s) x_{t_0} = 0, \tag{5.33}$$

$$\sum_{i=1}^{N} \sum_{t:s_t^i = s} u_t \left[C(s)\widehat{x}_{t|T}^i + u_t D(s) + D_0(s) - y_t\right] \widehat{w}_T^i + \mu_s \sigma_Y^2(s) u_{t_0} = 0, \tag{5.34}$$

$$\sum_{i=1}^{N} \sum_{t:s_t^i = s} \left[C(s)\widehat{x}_{t|T}^i + D(s)u_t + D_0(s) - y_t\right] \widehat{w}_T^i + \mu_s \sigma_Y^2(s) = 0. \tag{5.35}$$

for $t = 1, \ldots, T$, and $1 \leq s \leq \kappa$.

5.4.2 Penalization with a prior probability distribution

In this section, we suppose that we have a prior probability distribution $p(\Theta)$ on the set of the unknown parameters Θ. Instead of looking for the expected value of the posterior as in the Bayesian inference (Chapter 4), the aim is to find the maximum *a posteriori* estimate (MAP) (Green [1990]). This can be formulated as follows:

$$\begin{aligned}\tilde{\Theta} &= \arg\max_{\Theta} \log p(\Theta \mid y_{0:T}) \\ &= \arg\max_{\Theta} \log p_{\Theta}(y_{0:T}) + \log p(\Theta), \end{aligned} \tag{5.36}$$

where $\tilde{\Theta}$ is the MAP, and $p(\Theta \mid y_{0:T})$ is the *a posteriori* distribution. The EM algorithm relating to equation (5.36) consists in iteratively replacing a trial estimate Θ' by Θ maximizing the conditional expectation $\mathcal{Q}(\Theta, \Theta')$ of the completed-likelihood penalized with the prior probability distribution $p(\Theta)$:

$$\mathcal{Q}'(\Theta, \Theta') = \mathcal{Q}(\Theta, \Theta') + \log p(\Theta). \tag{5.37}$$

Hence, the maximization step of the MCEM algorithm is achieved by solving:

$$\frac{\partial}{\partial \Theta} \widehat{\mathcal{Q}}(\Theta, \Theta') + \frac{\partial}{\partial \Theta} \log p(\Theta) = 0. \tag{5.38}$$

In case of a SMSSM, this leads to a very complicated system of non-linear differential equations. To avoid this difficulty, it is possible to modify the algorithm quite simply. The maximization step is then replaced by solving:

$$\frac{\partial}{\partial \Theta} \widehat{\mathcal{Q}}(\Theta, \Theta') + \frac{\partial}{\partial \Theta} \log p(\Theta)|_{\Theta=\Theta'} = 0. \tag{5.39}$$

The only difference from equation (5.38) is that the derivatives of the prior are evaluated at the current value of Θ', rather than the new value Θ. This algorithm is referred to as *One-Step-Late* (OSL) algorithm (see Green [1990] for more information). This modification is motivated by the fact that if the algorithm converges slowly, then $\partial \log p(\Theta)/\partial \Theta$ and $\partial \log p(\Theta')/\partial \Theta'$ will not be much different. In addition, if the algorithm converges, equations (5.38) and (5.39) have then the same limit, namely the same MAP estimate. Green [1990] proves that the OSL algorithm converges at least as quickly as the EM algorithm.

An important but difficult question to be solved when using the OSL to avoid identifiability issue is the choice of the prior distribution. For instance, if a uniform prior is used, the non-identifiability of the parameters persists and the OSL could fail to efficiently estimate the MAP.

In this study, the following informative prior is used: the prior of $B(s)$ (resp. $C(s)$, $D(s)$ and $D_0(s)$), $1 \leq s \leq \kappa$, is a normal distribution with mean $\alpha(s)$ (resp. $\gamma(s)$, $\zeta(s)$ and $\zeta_0(s)$), and variance $\beta(s)$ (resp. $\mu(s)$, $\epsilon(s)$ and $\epsilon_0(s)$); whereas the prior of $\sigma_X^2(s)$ (resp. $\sigma_Y^2(s)$), $1 \leq s \leq \kappa$, is an inverse-gamma distribution with shape parameter $\varrho(s)$ (resp. $\upsilon(s)$), and scale parameter $\tau(s)$ (resp. $\psi(s)$).

Solving equation (5.39) leads to the following system of differential equations:

$$\sum_{i=1}^{N} \sum_{t:s_t^i=s} u_t \left[\widehat{x}_{t-1|T}^i + u_t \Delta t B(s) - \widehat{x}_{t|T}^i \right] \widehat{w}_T^i = -\sigma_X^2(s) \frac{B'(s) - \alpha(s)}{\beta(s)}, \tag{5.40}$$

$$\sum_{i=1}^{N} \sum_{t:s_t^i=s} \left[C(s) \widehat{x}_{t,t|T}^i + D(s) \widehat{x}_{t|T}^i u_t + D_0(s) \widehat{x}_{t|T}^i - \widehat{x}_{t|T}^i y_t \right] \widehat{w}_T^i = -\sigma_Y^2(s) \frac{C'(s) - \gamma(s)}{\mu(s)}, \tag{5.41}$$

$$\sum_{i=1}^{N} \sum_{t:s_t^i=s} u_t \left[C(s)\widehat{x}_{t|T}^i + u_t D(s) + D_0(s) - y_t \right] \widehat{w}_T^i = -\sigma_Y^2(s) \frac{D'(s) - \zeta(s)}{\epsilon(s)}, \tag{5.42}$$

$$\sum_{i=1}^{N} \sum_{t:s_t^i=s} \left[C(s)\widehat{x}_{t|T}^i + D(s)u_t + D_0(s) - y_t \right] \widehat{w}_T^i = -\sigma_Y^2(s) \frac{D'(s) - \zeta_0(s)}{\epsilon_0(s)}, \tag{5.43}$$

$$\begin{aligned} \sum_{i=1}^{N} \sum_{t:s_t^i=s} \Big[& \sigma_X^2(s) + 2\widehat{x}_{t,t-1|T}^i - 2\widehat{x}_{t-1|T}^i B(s) u_t \Delta t - \widehat{x}_{t,t|T}^i - \widehat{x}_{t-1,t-1|T}^i - B(s)^2 u_t^2 \Delta t^2 \\ & + 2B(s) u_t \Delta t \widehat{x}_{t|T}^i \Big] \widehat{w}_T^i = -2\sigma_X^4(s) \left(\frac{\tau(s)}{\varrho(s)} - \frac{\varrho(s)+1}{{\sigma'_X}^4(s)} \right), \end{aligned} \tag{5.44}$$

$$\begin{aligned} \sum_{i=1}^{N} \sum_{t:s_t^i=s} \Big[& \sigma_Y^2(s) - y_t^2 - C^2(s)\widehat{x}_{t,t|T}^i - D(s)^2 u_t^2 - D_0(s)^2 + 2y_t D_0(s) \\ & + 2C(s)\widehat{x}_{t|T}^i \left(y_t - D(s)u_t - D_0(s) \right) + 2D(s)u_t \left(y_t - D_0(s) \right) \Big] \widehat{w}_T^i \\ & = -2\sigma_Y^4(s) \left(\frac{\psi(s)}{\upsilon(s)} - \frac{\upsilon(s)+1}{{\sigma'_Y}^4(s)} \right). \end{aligned} \tag{5.45}$$

In the next section, these two penalized version of the MCEM algorithm are calibrated through a sensitivity analysis performed using a simulated dataset.

5.5 Sensitivity analysis using simulated data

The accuracy of the presented parameters inference algorithms depends, on the one hand, on the initialization of the set of unknown parameters as the EM may converge to a local maxima of likelihood and, on the other hand, on the number of particles since the EM algorithm is approximated using a Monte Carlo method. In addition, in the case of penalization with a prior probability distribution, the influence of the choice of this prior should be tested.

The dataset used in this section is simulated through the SMSSM with number of hidden discrete states $\kappa = 3$ described in Chapter 3. Table 5.1 recalls the parameters values of this model.

	B	C	D	D_0	σ_X^2	σ_Y^2
$s=1$	-1.20	0.11	-350	375	10^{-4}	20
$s=2$	-1.35	0.15	-400	380	10^{-4}	1
$s=3$	-1.30	0.20	-420	385	10^{-4}	1

TABLE 5.1: Parameters values of the SMSSM with $\kappa = 3$ used to generate the simulated dataset

5.5.1 MCEM algorithm penalized with identifiability constraints

5.5.1.1 Initialization procedure

The EM algorithm may converge to a local maxima of likelihood depending on the starting value. To avoid this difficulty, the EM algorithm is repeated several times with different initializations of Θ. However, when the number of parameters to be estimated is large, the initialization should be repeated many times. For mixture models, as SMSSMs, Baudry and Celeux [2015] develop a more sophisticated procedure, called *nested initialization*. Hereafter, we test two types of initialization procedure, the popular independent initialization and the nested one.

In order to validate the MCEM algorithm, the ML is estimated for $\kappa = \{2, \ldots, 7\}$. These estimated MLs should not be decrease when κ increases. Indeed, the estimated ML for a model of order $\kappa+1$ should be at least equal to the estimated ML for a model of order κ by duplicating one the hidden discrete state.

Independent initialization - Each unknown parameter is initialized independently through a Gaussian distribution with mean computed based on its physical interpretation and/or expert knowledge. The variance should be relatively large to ensure that the initialization varies each time the MCEM algorithm is run. The variance of the process noise W_t and measurement noise Σ_t are set to 10^3, as the transition and observation equations are unreliable at this stage. In addition, the transition matrix P is initialized by assigning random values to each row which is then normalized.

Figure 5.1 shows the results of the estimated ML for $\kappa = \{2, \ldots, 7\}$ based on the simulated data. It can be observed that some estimations may be far from being optimal. Indeed, contrary to the results obtained for $\kappa = 5$ and 7, the maximum of likelihood must not decrease when the model order κ increases. In order to overcome this problem, a different initialization strategy, called *nested initialization*, is used (Baudry and Celeux [2015]).

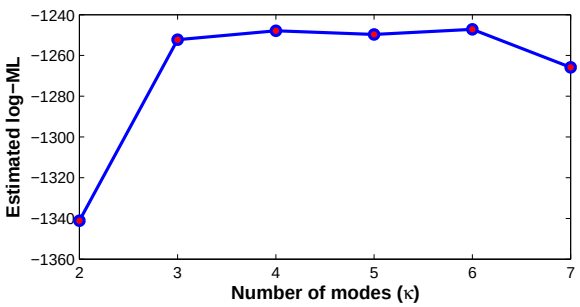

FIGURE 5.1: [Simulated data] Estimated log ML for $\kappa = \{2, \ldots, 7\}$. For each κ, 20 independent initializations of the MCEM algorithm penalized with identifiability constraints

Nested initialization - This strategy is also called $Km1$ (for K *minus* 1). For $\kappa \geq 2$, suppose that the set of parameters Θ of the model of order $\kappa - 1$, denoted $\Theta_{(\kappa-1)}$, is available. Then, to initialize $\Theta_{(\kappa)}$, the state κ_0 having the maximum entropy is divided into two states. Figure 5.2 illustrates this strategy. The entropy of a state r is computed as follows

$$\mathrm{E}(r) = -\sum_{t=0}^{T} p_\Theta(s_t = r \mid y_{0:T}) \log p_\Theta(s_t = r \mid y_{0:T}), \tag{5.46}$$

where $r = arg\max_s \; p_\Theta(s_t = s \mid y_{0:T})$.

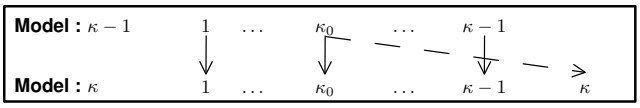

FIGURE 5.2: Nested initialization strategy ($Km1$ strategy) for the EM algorithm

In Figure 5.3 the parameters vector $\Theta_{(2)}$ is estimated based on the simulated data. To initialize the parameters for $\kappa = 3$, the entropy of the two states are computed. Figure 5.3 shows that the observations under the state having the highest entropy (*e.g.* state 1) are more dispersed than those under the other states (*e.g.* state 2). Thus for $\kappa = 3$, the state 1 is divided into two states and the parameters vector of state 1 and 3 is initialized as follows

$$\Theta_{(3)} = \{\Theta_{(2)}(s = 1) + random_1,\ \Theta_{(2)}(s = 2),\ \Theta_{(2)}(s = 1) + random_2\}.$$

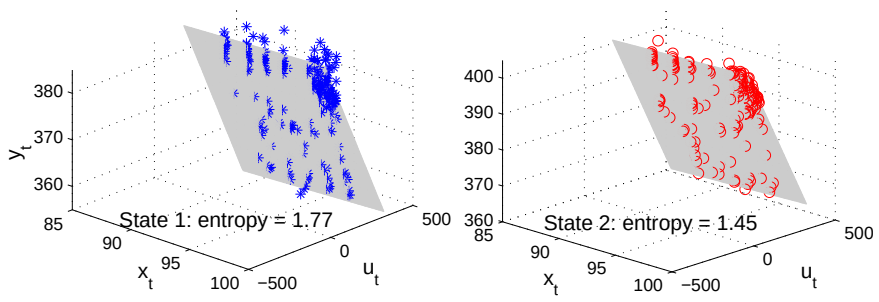

FIGURE 5.3: [Simulated data] Distribution of the observations around the estimated parameters per state: the observations under the State 1 are more dispersed than those under the State 2. State 1 should be divided into two states

In Figure 5.4, it can be observed that the problem of suboptimal ML solutions is resolved, and the estimated MLs with nested initializations increase with respect to the model order κ.

5.5.1.2 Number of particles

In order to study the influence of the number of particles on the estimated ML, we run 20 EM with identical initialization for $N = 100$ and $N = 1000$ where $\kappa = 3$. Figure

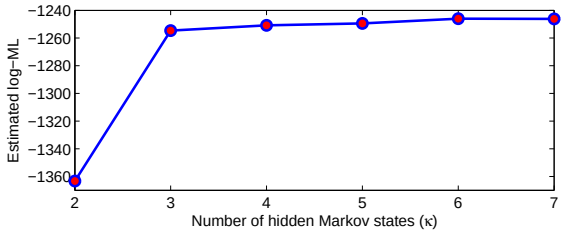

FIGURE 5.4: [Simulated data] Estimated log ML with $\kappa = \{2, \ldots, 7\}$. For each κ, 20 nested initializations of the MCEM algorithm penalized with the identifiability constraints

5.5 shows that the mean of the estimated ML is almost equal in both cases. However, the dispersion of the estimated ML for $N = 100$ is relatively higher than the one for $N = 1000$.

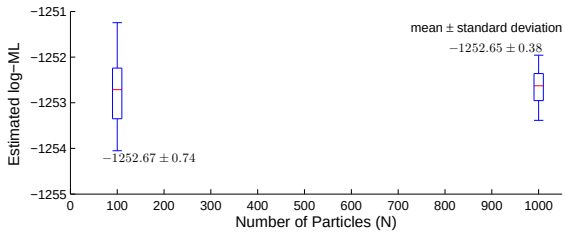

FIGURE 5.5: [Simulated data] Estimated log-ML for $N = 100$ and 1000 with $\kappa = 3$. For each case, 20 identical initializations of the MCEM algorithm penalized with identifiability constraints

Figure 5.6 shows the estimated MLs for different number of particles N with $\kappa = 3$. For each N, the EM algorithm is repeated 20 times with different independent initializations. For each N the dispersion of the estimated MLs has two sources: the initialization of the EM algorithm and the particle filter. We note that, in almost all cases, this dispersion remains relatively small. Hence, the choice of the likelihood as an accuracy indicator of a SMSSM seems pertinent.

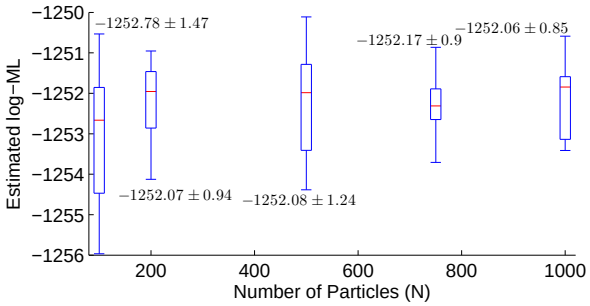

FIGURE 5.6: [Simulated data] Estimated log-ML for $N = \{100, 200, 500, 750, 1000\}$ with $\kappa = 3$. For each case, 20 different initializations of the MCEM algorithm penalized with identifiability constraints

5.5.1.3 Validation of the proposed MCEM algorithm

In this section, we compare the true parameters of the simulated model with $\kappa = 3$ and estimated parameters for $\kappa = 3$ and 5. For $\kappa = 3$, Figure 5.7 highlights that $B(s)$ is less well-estimated than the parameters of the equation of the observed variable Y_t. It is not surprising since $B(s)$ describes the evolution of the unobserved state X_t. For $\kappa = 5$,

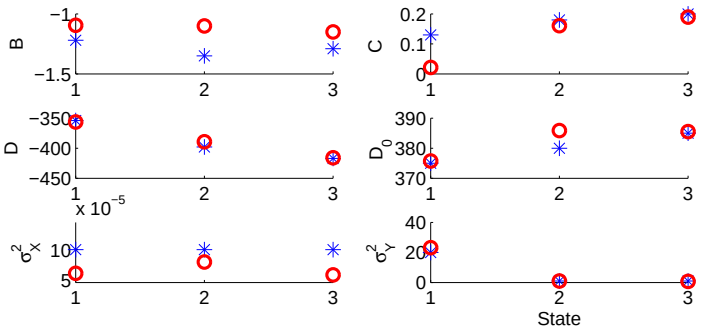

FIGURE 5.7: [Simulated data] Comparaison between true ($*$) and estimated ($\circ$) parameters with $\kappa = 3$. The parameters are estimated using a MCEM algorithm penalized with identifiability constraints

Figure 5.8 highlights that State 1 is duplicated into 2 states: State 3 and 5. In this case, for $\kappa > 3$, the EM algorithm duplicates a state of the true model $\kappa = 3$ in order to find the set of parameters that maximizes the likelihood $p_\Theta(y_{0:T})$. This can be clearly seen in Figure 5.4 where the estimated likelihood is almost stable for $\kappa \geq 3$.

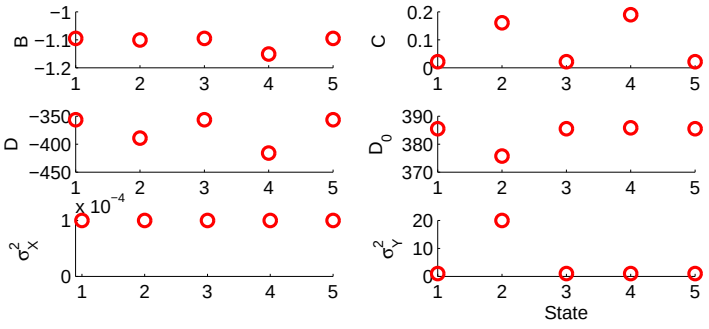

FIGURE 5.8: [Simulated data] Estimated parameters for SMSSM with $\kappa = 5$. The parameters are estimated using a MCEM algorithm penalized with identifiability constraints

5.5.2 MCEM algorithm penalized with a prior probability distribution

In this section, the influence of the prior probability is studied using three different types of prior:

1. very informative: the expected value of each parameter is equal to the true one, and the variance is small (cf. Table 5.2),

2. informative: the expected value of each parameter is equal to the true one, but the variance is large (cf. Table 5.3),

3. non-informative: the states are not differentiated. For each parameter, we assign the same expected value to all states with a large variance (cf. Table 5.4).

	$B : (\alpha,\ \beta)$	$C : (\gamma,\ \mu)$	$D : (\zeta,\ \epsilon)$	$D_0(\zeta_0,\ \epsilon_0)$	$\sigma_X^2 : (\varrho,\ \upsilon)$	$\sigma_Y^2 : (\tau,\ \psi)$
$s = 1$	$(-1.20,\ 1)$	$(0.11,\ 0.1)$	$(-350,\ 50)$	$(375,\ 50)$	$(3,\ 0.5)$	$(3,\ 1)$
$s = 2$	$(-1.35,\ 1)$	$(0.15,\ 0.1)$	$(-400,\ 50)$	$(380,\ 50)$	$(3,\ 0.5)$	$(3,\ 1)$
$s = 3$	$(-1.30,\ 1)$	$(0.20,\ 0.1)$	$(-420,\ 50)$	$(385,\ 50)$	$(3,\ 0.5)$	$(3,\ 1)$

TABLE 5.2: Very informative priors

	$B : (\alpha,\ \beta)$	$C : (\gamma,\ \mu)$	$D : (\zeta,\ \epsilon)$	$D_0(\zeta_0,\ \epsilon_0)$	$\sigma_X^2 : (\varrho,\ \upsilon)$	$\sigma_Y^2 : (\tau,\ \psi)$
$s = 1$	$(1.20,\ 10)$	$(0.11,\ 1)$	$(-350,\ 10^3)$	$(375,\ 10^3)$	$(3,\ 0.5)$	$(3,\ 1)$
$s = 2$	$(1.35,\ 10)$	$(0.15,\ 1)$	$(-400,\ 10^3)$	$(380,\ 10^3)$	$(3,\ 0.5)$	$(3,\ 1)$
$s = 3$	$(1.30,\ 10)$	$(0.20,\ 1)$	$(-420,\ 10^3)$	$(385,\ 10^3)$	$(3,\ 0.5)$	$(3,\ 1)$

TABLE 5.3: Informative priors

	$B : (\alpha,\ \beta)$	$C : (\gamma,\ \mu)$	$D : (\zeta,\ \epsilon)$	$D_0(\zeta_0,\ \epsilon_0)$	$\sigma_X^2 : (\varrho,\ \upsilon)$	$\sigma_Y^2 : (\tau,\ \psi)$
$s = 1$	$(-2,\ 10)$	$(0.5,\ 1)$	$(-400,\ 10^3)$	$(350,\ 10^3)$	$(3,\ 0.5)$	$(3,\ 1)$
$s = 2$	$(-2,\ 10)$	$(0.5,\ 1)$	$(-400,\ 10^3)$	$(350,\ 10^3)$	$(3,\ 0.5)$	$(3,\ 1)$
$s = 3$	$(-2,\ 10)$	$(0.5,\ 1)$	$(-400,\ 10^3)$	$(350,\ 10^3)$	$(3,\ 0.5)$	$(3,\ 1)$

TABLE 5.4: Non-informative priors

We run the algorithm 20 times with a nested initialization. Figures 5.9, 5.10 and 5.11 show the estimated parameters when using a very informative, informative and non-informative priors respectively. It can be observed that the identifiability issue is resolved, and the parameters are efficiently estimated in the three cases. Nevertheless when using a non-informative prior, B is less well estimated than the other parameters as for the MCEM penalized with the identifiability constraints (cf. Figure 5.7).

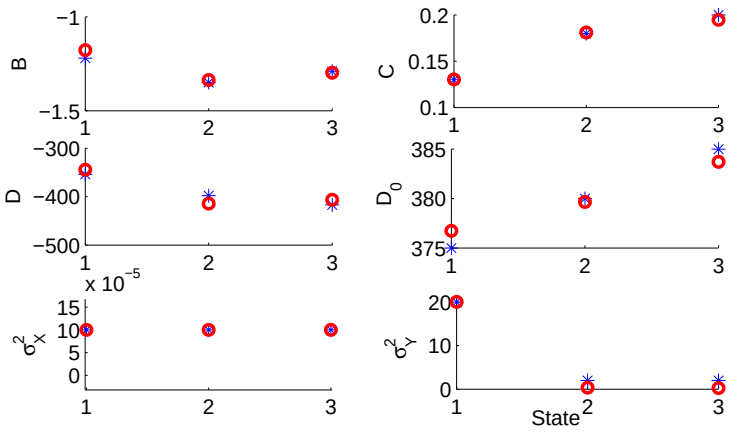

FIGURE 5.9: [Simulated data] Comparaison between true ($\star$) and estimated ($\circ$) parameters with $\kappa = 3$ using a very informative priors

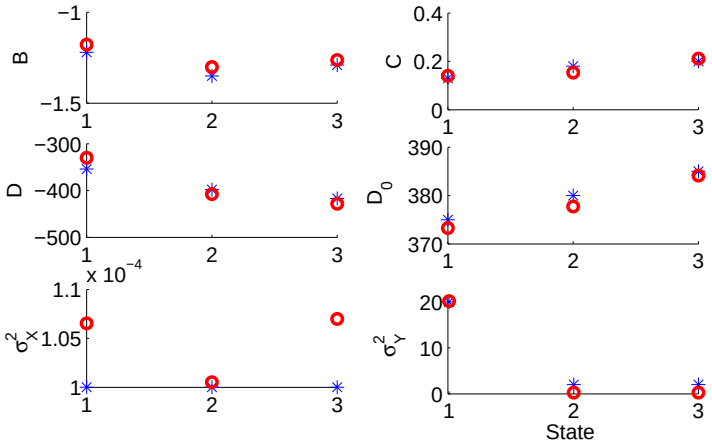

FIGURE 5.10: [Simulated data] Comparaison between true ($\star$) and estimated ($\circ$) parameters with $\kappa = 3$ using an informative priors

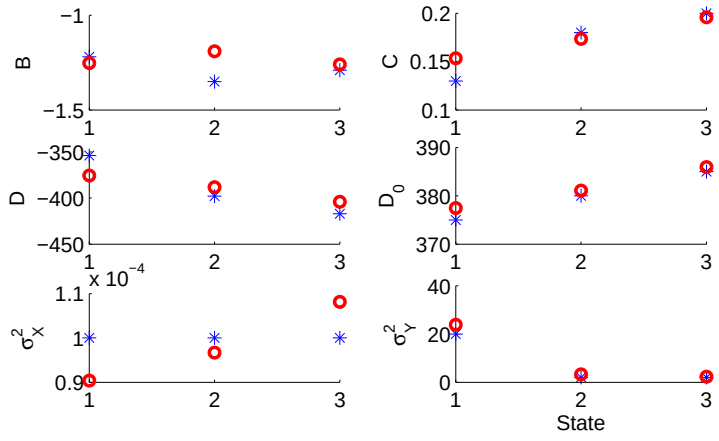

FIGURE 5.11: [Simulated data] Comparaison between true ($\star$) and estimated ($\circ$) parameters with $\kappa = 3$ using a non-informative priors

5.6 Conclusion

In this chapter, we have developed two penalized MCEM algorithms to estimate the set of unknown parameters Θ of the considered SMSSM (3.22)-(3.23). Indeed, as the model is non-identifiable, ML methods fail to properly estimate the unknown parameters Θ without a penalization term. Two types of penalization have been used: penalization with constraints ensuring the identifiability of the model, and penalization with a prior probability distribution.

These two penalized algorithms are calibrated using simulated data. For this purpose, the influence of the initialization procedure, the number of particles, and the prior probability distribution has been studied.

The convergence of the EM algorithm depend primarily on the starting position. It may converge to a local ML depending on the initialization of the parameters. In this study, two initialization procedures have been tested: independent and nested initializations. The results show that the nested initialization is more adapted to SMSSMs than the independent one. Indeed, using an independent initialization, the estimated MLs decrease sometimes when the model order increases. On the other hand, the nested initialization is suitable to mixture models: the initialization of the model with order $\kappa + 1$ is based on the estimated ML of the model with order κ. The state having the largest entropy is then divided into two states, thus ensuring a successful initialization.

An exact EM algorithm being at a prohibitive computational cost, a MCEM algorithm is used instead. The estimated ML is thus affected by the number of the simulated particles. The results show that when an appropriate number of particles is used ($N \simeq 100$ for $\kappa = 3$ and $T = 500$), the dispersion of the estimated MLs is relatively small. Hence, the choice of the likelihood as an accuracy indicator of a SMSSM seems pertinent.

For the MCEM algorithm penalized with a prior distribution, an appropriate choice of this prior is important to avoid the identifiability issue. The use of Gaussian priors for B, C, D and D_0 and inverse-gamma priors for σ_X^2 and σ_Y^2 seems pertinent, even when the states are not *a priori* differentiated.

To conclude, the two penalized MCEM algorithms seem estimate properly the set of unknown parameters Θ. Nevertheless, the use of prior distributions introduces a large number of hyperparameters (i.e., $12 \cdot \kappa$) that must be properly chosen to avoid identifiability issue. This can be complicated, especially when these unknown parameters lack physical interpretation. When they do have one, these two algorithms can be applied, and the estimated Θ having the maximum likelihood would be retained.

In the next chapter, we consider the problem of the estimation of the proper number of the hidden Markov states using different model selection criteria.

Chapter 6

Estimation of the number of hidden Markov states

Ce chapitre est consacré à l'estimation du nombre d'états cachés κ, considéré comme connu dans les chapitres précédents. Soit un ensemble de modèles candidats formé de SM-SSMs avec un nombre variable d'états cachés κ, et noté $\mathcal{M} = \{M_\kappa = (\kappa, \Theta_\kappa);\ \kappa_{\min} \leq \kappa \leq \kappa_{\max}\}$. L'ensemble d'estimateurs correspondant est noté $\widehat{\mathcal{M}} = \{\widehat{M}_\kappa = (\kappa, \widehat{\Theta}_\kappa);\ \kappa_{\min} \leq \kappa \leq \kappa_{\max}\}$. Notons M le meilleur modèle au sens d'une fonction de perte qui sera pour nous la divergence de Kullback-Leibler (KL) entre la vrai distribution inconnue $p^*(y_{0:T})$ et le modèle.

Les états discrets cachés visent à modéliser les différents régimes de fonctionnement de la batterie. Cependant, aucune information n'est *a priori* disponible concernant le nombre de ces régimes, et le nombre d'états discrests cachés doit donc être estimé.

Une simple méthode de sélection de modèle consiste à choisir le modèle minimisant une fonction de risque donnée dépendant de la précision du modèle et de son attache à la base d'apprentissage. Ce risque est généralement représenté par la vraisemblance $p_\Theta(y_{0:T})$ dans le cas où les paramètres du modèle sont estimés par une méthode de type ML. Parmi l'ensemble $\widehat{\mathcal{M}}$, le modèle ayant la plus grande vraisemblance est ainsi sélectionné. Suivant cette logique, le modèle à $\kappa_{\max}$ états cachés serait toujours choisi, et un surapprentissage serait introduit. Pour éviter ce biais "d'optimiste" lors de l'évaluation de performance, la complexité du modèle est généralement prise en compte. Elle correspond au nombre de paramètres à estimer. Dans le cas du SMSSM (3.22)-(3.23), cette complexité est égale à

$$K(\kappa) = (\kappa - 1)^2 + 6 \cdot \kappa = \kappa^2 + 4\kappa + 1. \tag{6.1}$$

De ce fait, l'enjeu d'un critère de sélection de modèle est de rechercher le meilleur compromis entre la précision du modèle et sa complexité. Quand la vraisemblance est utilisée comme un indicateur de précision, ce critère correspond à cette vraisemblance pénalisée

par une fonction de la complexité du modèle. Plusieurs fonctions de pénalité ont été étudiées afin de fournir un critère de sélection de modèle optimal dans un sens donné.

La suite de ce chapitre passe en revue les résultats théoriques des critères d'information d'Akaike (AIC, Akaike [1998]) et de Bayes (BIC, Schwarz [1978]), de l'heuristique de pente (Birgé and Massart [2007]) et de la validation croisée (Celeux and Durand [2008], Ripley [1996]).

AIC et BIC sont parmi les critères de sélection de modèle les plus utilisés. AIC est basé sur une fonction de pénalité fixe dépendant de la compléxité du modèle, tandis que la fonction de pénalité associée à BIC dépend en plus de la taille de la base d'apprentisage. Toutefois, l'AIC et le BIC traitent différemment le problème de sélection de modèle. Les résultats théoriques de ces deux critères sont détaillés dans les Sections 6.1 et 6.2, et résumés ci-après.

L'AIC choisit le modèle minimisant la divergence de KL entre la vraie distribution $p^*(y_{0:T})$ et celle estimée par M_κ. La vraie distribution étant inconnue, cette divergence est estimée. Akaike [1998] démontre que la minimisation de cette divergence correspond à la minimisation du log-ML de $\widehat{M}_\kappa$ pénalisé par le nombre de paramètres à estimer :

$$\kappa^*(\text{AIC}) = \arg\min_\kappa -2\log p_{\widehat{\Theta}_\kappa}(y_{0:T}) + 2(\kappa^2 + 4\kappa + 1). \tag{6.2}$$

Le BIC se place dans un contexte bayésien de sélection de modèle. Il cherche à sélectionner le modèle $\widehat{M}_\kappa$ qui maximise la probabilité *a posteriori* $p(\widehat{M}_\kappa|y_{0:T})$. Schwarz [1978] démontre que lorsque T tend vers l'infini et que κ et $y_{0:T}$ sont fixés, la probabilité *a posteriori* est estimé par le log-ML de $\widehat{M}_\kappa$ pénalisé par une fonction de pénalité dépendant de T la taille de la base d'apprentissage et de la complexité du modèle. Le modèle sélectionné par BIC correspond à

$$\kappa^*(\text{BIC}) = \arg\min_\kappa -2\log p_{\widehat{\Theta}_\kappa}(y_{0:T}) + (\kappa^2 + 4\kappa + 1)\log T. \tag{6.3}$$

Contrairement à AIC et BIC auxquels sont associés à des fonctions de pénalité fixes, l'heuristique de pente récemment développé par Birgé and Massart [2007] utilise une fonction de pénalité variable dépendant du comportement de la vraisemblance des modèles de grand ordre. Ce critère choisit le modèle minimisant l'espérance de la divergence de KL entre la vraie distribution $p^*(y_{0:T})$ et celle estimée par $\widehat{M}_\kappa$. La vraie distribution étant inconnue, il cherche à identifier la fonction de pénalité optimale qui permet d'approcher cette espérance par une vraisemblance pénalisée. Birgé and Massart [2007] démontre que la fonction de pénalité optimale est égale à

$$2w_{\text{opt}}(\kappa^2 + 4\kappa + 1), \tag{6.4}$$

où w_{opt} est une constante à estimer dépendant du comportement du ML des modèles de grand ordre. Une condition nécessaire pour pouvoir utiliser ce critère est d'observer un comportement linéaire du ML des modèles de grand ordre. Si tel est le cas, w_{opt}

est estimé par la pente de la partie linéaire du maximum de vraisemblance. La théorie associée à ce critère est présentée dans la Section 6.3.

La validation croisée est utilisée dans le cadre de la sélection de modèle pour évaluer la capacité de prévision d'un modèle $\widehat{M}_\kappa$, et fournit une estimation de la divergence de KL entre la vraie distribution $p^*(y_{0:T})$ et celle estimée par $\widehat{M}_\kappa$ (Ripley [1996]). Cependant, dans le cas des SMSSMs l'application de la validation croisée fait face à des difficultés techniques. En effet, les données sont dépendantes et le retrait de certaines observations pourrait rompre la dépendance markovienne entre les états cachés s_t et l'évolution des états continus x_t. Une solution proposée par Celeux and Durand [2008] dans le cadre des HMMs consiste à découper d'une façon déterministe la base d'apprentissage en deux parts : elles correspondent respectivement aux observations à indices pairs et impairs de la base d'origine et elles sont notées y^{pair} et y^{impair}. Celeux and Durand [2008] démontre que cette répartition permet de maintenir la dépendance entre les observations de chaque partie. Le rôle de ces deux parties est ensuite permuté : lorsque y^{pair} (resp. y^{impair}) est utilisé pour estimer Θ_κ, sa vraisemblance est calculée en y^{impair} (resp. y^{pair}). La divergence de KL est alors estimée par la moyenne empirique de ces deux vraisemblances. Le modèle sélectionné par la validation croisée est celui ayant la plus petite divergence estimée. Cependant, cette divergence estimée peut être biaisée du fait que y^{pair} et y^{impair} sont dépendants. Par conséquent, nous proposons un autre découpage également déterministe de la base d'apprentissage : les deux parts correspondent respectivement à la première et la seconde moitié de celle-ci. Le calcul de la vraisemblance croisée et la sélection du κ sont effectués de la même manière que le découpage pair/impair.

En se basant sur une base de données générée par un SMSSM à $\kappa = 3$ états cachés, nous avons testé la capacité de ces critères à estimer le vrai nombre d'états cachés (cf. Section 6.5). Les résultats montrent que tous les critères ont un comportement satisfaisant et permettent d'estimer le vrai nombre d'états cachés. Néanmoins, AIC et la validation croisée pair/impair ont tendance à surestimer κ.

Toutefois, l'une des principales différences entre ces critères est leur coût de calcul. L'AIC et le BIC sont simples à implémenter : ils sélectionnent, parmi les modèles candidats, ceux qui les minimisent respectivement. L'heuristique de pente nécessite l'estimation de la vraisemblance des modèles de grand ordre, ne figurant parfois pas *a priori* parmi les modèles candidats, pour vérifier un comportement linéaire du ML. La validation croisée consomme le plus de temps de calcul : les modèles estimés $\widehat{M}_\kappa$ ne peuvent pas être utilisés directement, et pour chaque κ deux modèles doivent être de nouveau estimés pour les deux parts de la base d'apprentissage.

De plus, le travail présenté dans ce chapitre met en évidence une autre problématique étroitement liée au problème d'estimation du nombre d'états cachés : le choix du pas d'échantillonnage. En effet, le coût de calcul des algorithmes d'estimation des paramètres du modèle augmente linéairement avec la taille de la base d'apprentissage. Lorsque

cette base est formée de plusieurs heures de charge et de décharge d'une batterie, un rééchantillonnage peut réduire considérablement le temps d'estimation des paramètres. Cependant, le pas d'échantillonnage pourrait affecter le nombre d'états cachés estimés par l'un des critères de sélection de modèle présentés. Pour illustrer cette influence, considérons une base d'apprentissage générée par un SMSSM à $\kappa = 2$ états cachés. Pour cette base, nous considérons que $s_t = 1$ pour ses indices pairs et $s_t = 2$ pour ceux impairs. Avec un rééchantillonnage à pas $T_e = 2$, l'un des deux états est perdu et le nombre estimé d'états cachés est égal à 1. L'influence du rééchantillonnage sur le choix de κ est également illustrée à travers des données simulées (cf. Figure 6.5). D'un point de vue statistique, cette question reste ouverte notamment dans le cadre des modèles de mélange.

En conclusion, les quatres critères présentés traitent différemment la problématique de sélection de modèle et peuvent induire des choix différents. Si tel était le cas, on peut soit choisir le modèle le plus parcimonieux (ayant le plus petit κ) nécessitant ainsi un espace de stockage plus petit et une capacité de calcul plus faible, soit réaliser plus de tests de validation afin d'évaluer la capacité de prédiction de chacun de modèles sélectionnés.

Contents

Up to now, we have considered that the number of hidden Markov states κ is known. In this chapter, several model selection criteria are used to select the model with "proper" κ when a set of candidate models is considered.

The hidden Markov states aim to model the different regimes of the battery dynamics. However, no indications are *a priori* available with regards to the number of these regimes. Consequently, the number of hidden Markov states κ should be properly estimated in order to provide an accurate *SoC* model.

Let us consider a countable set of models $\mathcal{M} = \{M_\kappa = (\kappa, \Theta_\kappa); \kappa_{\min} \leq \kappa \leq \kappa_{\max}\}$, where M_κ denotes a SMSSM of order κ and set of parameters Θ_κ. The corresponding estimator set is denoted $\{\widehat{M}_\kappa = (\kappa, \widehat{\Theta}_\kappa); \kappa_{\min} \leq \kappa \leq \kappa_{\max}\}$. The model of interest, denoted as M, is related to the unknown distribution of the sequence of observations $p^*(y_{0:T})$.

A straightforward method for model selection consists of optimizing an appropriate risk function that depends on the losses emerging when a model $\widehat{M}_\kappa$ is selected. Since $p^*(y_{0:T})$ is unknown, this risk is estimated through a learning dataset. It typically measures the capacity of the estimated model to fit the data. This empirical risk corresponds to the likelihood when the model parameters are estimated using a ML approach. Amongst the candidate models, the model maximizing the likelihood is thus selected. Nevertheless, this selection may introduces an undesirable optimistic bias due to overfitting. Thus, the *statistical complexity* of the model should be introduced to avoid this selection bias during the performance evaluation. This complexity corresponds typically to the number of free parameters. For the SMSSM (3.22)-(3.23), it is equal to

$$K(\kappa) = (\kappa - 1)^2 + 6 \cdot \kappa = \kappa^2 + 4\kappa + 1. \tag{6.5}$$

Accordingly, an appropriate model selection criterion is commonly determined by a "trade-off" between accuracy requirements and model complexity. When the likelihood is used as an accuracy indicator, the selected model is the one optimizing the likelihood penalized with its complexity.

Several studies have been made to provide a suitable penalty function. Some well-known penalized criteria with fixed penalties, such as the Bayesian Information Criterion (BIC)

(Schwarz [1978]) and the Akaike Information Criterion (AIC) (Akaike [1998]), have been widely studied and used in real-world applications (see Burnham and Anderson [2002] for more information). The Slope Heuristics Criterion (SHC), developed recently by Birgé and Massart [2007], uses a variable penalty depending on the behavior of the ML for large model orders. Other studies propose to estimate the number of components in a mixture model by maximizing the cross-validation likelihood (CVL), for instance, Smyth [2000] in an independent data context, and Celeux and Durand [2008] in a HMM context.

In this chapter, we provide an overview of the theoretical results of these criteria and verify their ability to estimate the proper order of the SMSSM (3.22)-(3.23) using simulated data.

6.1 Akaike Information Criterion

The AIC developed by Akaike [1998] consists of selecting the model that minimizes the Kullback-Leibler divergence $\mathrm{KL}(M, \widehat{M}_\kappa)$ between $p^*(y_{0:T})$ the true pdf of $y_{0:T}$ and $p_{\widehat{\Theta}_\kappa}(y_{0:T})$ the pdf of $y_{0:T}$ with respect to the candidate models $\widehat{M}_\kappa$ ($\kappa_{\min} \leq \kappa \leq \kappa_{\max}$). It is defined as follows:

$$\begin{aligned}
\mathrm{KL}(M, \widehat{M}_\kappa) &= \mathbb{E}_{Y_{0:T}}\left[\log\left(\frac{p^*(y_{0:T})}{p_{\widehat{\Theta}_\kappa}(y_{0:T})}\right)\right] \\
&= \int \log\left(\frac{p^*(y_{0:T})}{p_{\widehat{\Theta}_\kappa}(y_{0:T})}\right) p^*(y_{0:T})\,\mathrm{d}y_{0:T} \\
&= \mathbb{E}_{Y_{0:T}}\left[\log\left(p^*(y_{0:T})\right)\right] - \mathbb{E}_{Y_{0:T}}\left[\log\left(p_{\widehat{\Theta}_\kappa}(y_{0:T})\right)\right]. \qquad (6.6)
\end{aligned}$$

Minimizing (6.6) with respect to κ is equivalent to minimizing

$$-\mathbb{E}_{Y_{0:T}}\left[\log\left(p_{\widehat{\Theta}_\kappa}(y_{0:T})\right)\right] = -\int \log p_{\widehat{\Theta}_\kappa}(y_{0:T}) p^*(y_{0:T})\,\mathrm{d}y_{0:T}. \qquad (6.7)$$

Since the true pdf $p^*(y_{0:T})$ is unknown, (6.7) must be estimated. A natural estimation is the log-likelihood $\log p_{\widehat{\Theta}_\kappa}(y_{0:T})$. However, $\widehat{\Theta}_\kappa$ and $y_{0:T}$ are dependent, and such an estimate will introduce an optimistic bias. Akaike [1998] proves that this bias is equal to the number of free parameters of the candidate model $\widehat{M}_\kappa$ under regularity conditions.

Accordingly, the AIC of a candidate model $\widehat{M}_\kappa$ is given by

$$\mathrm{AIC}(\kappa) = -2 \cdot \log p_{\widehat{\Theta}_\kappa}(y_{0:T}) + 2 \cdot K(\kappa), \qquad (6.8)$$

where $K(\kappa) = \kappa^2 + 4\kappa + 1$ is the number of free parameters of the SMSSM (3.22)-(3.23). The first term in (6.8) retains the model that fits the data, whereas the second term penalizes large-order models.

The proper number of hidden states κ with respect to AIC is then the one that minimizes AIC:

$$\kappa^*_{\text{AIC}} = \underset{\kappa_{\min} \leq \kappa \leq \kappa_{\max}}{\arg\min} \ \text{AIC}(\kappa). \tag{6.9}$$

It is worth mentioning that AIC performs well in many real-world applications (Burnham and Anderson [2002]) but not so good to estimate the order of a mixture model (McLachlan and Peel [2000]).

6.2 Bayesian Information Criterion

The BIC developed by Schwarz [1978] studies the model selection within a Bayesian framework: Θ_κ and M_κ are considered as random variables with prior distributions respectively $p(\Theta_\kappa)$ and $p(M_\kappa)$. BIC consists of selecting the model having the maximum *a posteriori* (MAP) given by

$$\kappa^*_{\text{BIC}} = \underset{\kappa_{\max} \leq \kappa \leq \kappa_{\max}}{\arg\min} \ p(M_\kappa \,|\, y_{0:T}). \tag{6.10}$$

From the Bayes' rule, the posterior distribution can be written as follows:

$$\begin{aligned} p(M_\kappa \,|\, y_{0:T}) =& \frac{p(y_{0:T}|M_\kappa) \cdot p(M_\kappa)}{p(y_{0:T})} \\ \propto& p(y_{0:T}|M_\kappa) \cdot p(M_\kappa). \end{aligned} \tag{6.11}$$

The prior $p(M_i)$ are assumed non-informative, thus there is no preference for either model:

$$p(M_1) = p(M_2) = \ldots = p(M_\kappa). \tag{6.12}$$

Based on (6.11) and (6.12), the model selected with BIC is that maximizing the marginal likelihood $p(y_{0:T}|M_\kappa)$. This marginal likelihood can be written as follows:

$$\begin{aligned} p(y_{0:T}|M_\kappa) &= \int p(y_{0:T}, \Theta_\kappa|M_\kappa)\, d\Theta_\kappa \\ &= \int p(y_{0:T}|\Theta_\kappa, M_\kappa) \cdot p(\Theta_\kappa|M_\kappa)\, d\Theta_\kappa. \end{aligned} \tag{6.13}$$

Since an exact computation of (6.13) is difficult, it is estimated using a Laplace approximation (the theoretical formulas are thoroughly detailed in Lebarbier and Mary-Huard [2004]). When T tends to infinity, the error terms of this approximation are omitted, then BIC corresponding to the model M_κ is given by

$$\text{BIC}(\kappa) = -2 \log p_{\widehat{\Theta}_\kappa}(y_{0:T}) + K(\kappa) \log T, \tag{6.14}$$

where $\widehat{\Theta}_\kappa$ is the ML estimate of Θ_κ. The proper number of hidden states κ with respect to BIC is then the one that minimizes BIC:

$$\kappa^*_{\text{BIC}} = \underset{\kappa_{\min}\leq\kappa\leq\kappa_{\max}}{\arg\min} \ \text{BIC}(\kappa). \tag{6.15}$$

BIC and AIC are derived from distinct perspectives and address differently the trade-off between accuracy and complexity. Indeed, BIC is "consistent" in the sense of asymptotically selecting M (i.e., the probability of selecting M by BIC approaches to one as $T \to \infty$), whereas AIC is "efficient" in the sense of asymptotically minimizing the mean-squared prediction error (Burnham and Anderson [2002]). In addition, BIC is more parsimonious than AIC (i.e., $\kappa^*_{\text{AIC}} \geq \kappa^*_{\text{BIC}}$. This is not surprising as large-order models are more penalized in BIC case.

6.3 Slope Heuristics Criterion

The slope heuristics method is introduced and proved for the first time by Birgé and Massart [2007]. This method is based on the existence of a *contrast* function γ fulfilling the fundamental property that

$$M = \underset{M_\kappa\in\mathcal{M}}{\arg\min}\ \mathbb{E}_{y_{0:T}}[\gamma(M_\kappa, y_{0:T})]. \tag{6.16}$$

In practice, the contrast is substituted by the empirical contrast γ_T. A suitable estimator of M is then the one minimizing the expectation of γ_T over $\mathcal{M}$.

In model parameters inference framework, the maximum likelihood is a minimum contrast estimator, i.e. $\gamma_T(M_\kappa) = p_{\Theta_\kappa}(y_{0:T})$, and the corresponding loss function is the KL divergence between the unknown model M and the trial model M_κ. The *optimal* model is thus the one minimizing the risk

$$\mathbb{E}[\text{KL}(M, \widehat{M}_\kappa)]. \tag{6.17}$$

However, this optimal model is intractable as M is unknown. The slope heuristics method tries to find an optimal penalty function for which the model selection criterion is as close as possible to the risk $\mathbb{E}[\text{KL}(M, \widehat{M}_\kappa)]$. The loss function can be decomposed as follows:

$$\begin{aligned}\text{KL}(M, \widehat{M}_\kappa) &= \int \log\left(\frac{p^*(y)}{p_{\Theta_\kappa}(y)}\right) p^*(y)dy + \int \log\left(\frac{p_{\Theta_\kappa}(y)}{p_{\widehat{\Theta}_\kappa}(y)}\right) p^*(y)dy \\ &= b_\kappa + V_\kappa,\end{aligned} \tag{6.18}$$

where b_κ is the bias representing the difference between M_κ and M, and V_κ is the variance reflecting the variability of the estimator $\widehat{M}_\kappa$ around its expected value M_κ due to the

learning dataset on which it is inferred. Accordingly, the risk (6.17) is given by

$$\mathbb{E}[\mathrm{KL}(M, \widehat{M}_\kappa)] = b_\kappa + \mathbb{E}[V_\kappa]. \tag{6.19}$$

The selected model $\widehat{M}_\kappa$ is the one minimizing the following criterion with respect to κ:

$$\gamma_T(\widehat{M}_\kappa) + \mathrm{pen}(M_\kappa), \tag{6.20}$$

where $\mathrm{pen}(M_\kappa)$ is the penalty function to be identified. Since $\gamma_T(M)$ does not depend on κ, this criterion can be written as follows:

$$\begin{aligned}\gamma_T(\widehat{M}_\kappa) - \gamma_T(M) + \mathrm{pen}(M_\kappa) &= \gamma_T(\widehat{M}_\kappa) - \gamma_T(M_\kappa) + \gamma_T(M_\kappa) - \gamma_T(M) + \mathrm{pen}(M_\kappa) \\ &= \widehat{b}_\kappa - \widehat{V}_b + \mathrm{pen}(M_\kappa),\end{aligned} \tag{6.21}$$

where $\widehat{b}_\kappa := \gamma_T(M_\kappa) - \gamma_T(M)$, and $\widehat{V}_b := \gamma_T(M_\kappa) - \gamma_T(\widehat{M}_\kappa)$.

Introducing (6.18) into (6.21) leads to

$$\gamma_T(\widehat{M}_\kappa) - \gamma_T(M) + \mathrm{pen}(M_\kappa) = \mathrm{KL}(M, \widehat{M}_\kappa) + (\widehat{b}_\kappa - b_\kappa) - (V_\kappa + \widehat{V}_\kappa) + \mathrm{pen}(M_\kappa). \tag{6.22}$$

For large-order models, the estimated model $\widehat{M}_\kappa$ cannot be significantly improved; hence it is sensible to assume that $\widehat{b}_\kappa - b_\kappa \approx 0$. In addition, based on concentration arguments, Birgé and Massart [2007] suppose that $\mathrm{KL}(M, \widehat{M}_\kappa)$ is close to its expectation (6.19). The selection criterion (6.21) can hence be approximated as follows:

$$\mathbb{E}[\mathrm{KL}(M, \widehat{M}_\kappa)] - (V_\kappa + \widehat{V}_\kappa) + \mathrm{pen}(M_\kappa). \tag{6.23}$$

In order to obtain a selection criterion close to the risk $\mathbb{E}[\mathrm{KL}(M, \widehat{M}_\kappa)]$, an optimal choice of the penalty function is

$$\mathrm{pen}_{\mathrm{opt}}(M_\kappa) = V_\kappa + \widehat{V}_\kappa. \tag{6.24}$$

The key assumption of the slope heuristics method is to consider $\widehat{V}_\kappa \approx V_\kappa$. The optimal penalty can then be approximated as follows:

$$\mathrm{pen}_{\mathrm{opt}}(M_\kappa) \approx 2 \cdot \widehat{V}_\kappa. \tag{6.25}$$

In this study, the complexity of the model is considered in the penalty function. Accordingly, the optimal penalty can be defined by

$$\mathrm{pen}_{\mathrm{opt}}(M_\kappa) = 2 \cdot w_{\mathrm{opt}} \cdot K(\kappa), \tag{6.26}$$

where w_{opt} is a constant to be estimated, and $K(\kappa)$ is the number of free parameters.

In order to use the slope heuristics to calibrate this constant, a required condition is to observe a linear behavior of the contrast function $\gamma_T(\widehat{M}_\kappa)$ for large-order models. If this condition is satisfied, w_{opt} can be estimated by the slope of the linear part of

$\gamma_T(\widehat{M}_\kappa)$. Algorithm 4 presents a data-driven algorithm proposed by Baudry et al. [2012] to implement the slope heuristics criterion.

Algorithm 4 Slope Heuristics algorithm

Input $\leftarrow \left\{\mathrm{ML}(\kappa) = \log p_{\widehat{\Theta}_\kappa}(y_{0:T});\ \kappa = \kappa_{\min}, \ldots, \kappa_{\max}\right\}$
if $\exists(\kappa, \kappa') \mid \kappa < \kappa'$ and $\mathrm{ML}(\kappa) = \mathrm{ML}(\kappa')$ **then**
  Remove $\mathrm{ML}(\kappa')$; *%To make the reading easier, the indexation is not modified*
end if
for $\kappa = \kappa_{\min} : \kappa_{\max}$ **do**
  $\widehat{\omega}(\kappa) \leftarrow$ Slope of the robust regression of $\{(ML(\kappa'), K(\kappa'))\,;\ \kappa' \geq \kappa\}$
  $\mathrm{Sl}(\kappa) = \underset{\kappa_{\min} \leq \kappa' \leq \kappa_{\max}}{\arg\min} \; - \mathrm{ML}(\kappa') + 2 \cdot \widehat{\omega}(\kappa)\, K(\kappa')$
end for
$\kappa^*_{SHC} = \underset{\kappa_{\min} \leq \kappa \leq \kappa_{\max}}{\arg\max} \sum_{\kappa'=\kappa_{\min}}^{\kappa_{\max}} \mathbb{1}\left(Sl(\kappa') = \kappa\right)$

6.4 Cross-Validation likelihood criterion

The cross-validation aims to evaluate the predictive performance of a trial model, which makes it a natural approach in model selection. Indeed, it provides an estimation of the KL divergence between $p^*(y_{0:T})$ the true pdf of $y_{0:T}$ and $p_{\widehat{\Theta}_\kappa}(y_{0:T})$ the pdf of $y_{0:T}$ with respect to a trial model $\widehat{M}_\kappa$ (cf. (6.6) and (6.7)) (Ripley [1996]).

This criterion consists of randomly splitting the learning dataset into two-fold. The cross-likelihood is then computed by permuting the role of these two subsequences: when the first (resp. second) subsequence is used to estimate Θ_κ, its corresponding log-likelihood is estimated on the second (resp. first) subsequence. The KL divergence is then estimated by averaging the two estimated log-likelihood.

However in the SMSSM context, the data are dependent, and removing some of the data from the learning dataset breaks the Markovian dependencies between the hidden Markov states S_t and the evolution of the continuous states X_t, which makes difficult the use of the cross-validation.

One straightforward solution consists of partitioning the original data sequence of the learning dataset into long continuous subsequences to maintain the dependencies between the data. Nevertheless, some states might never be reached into these subsequences which is harmful to assess the proper number of hidden states.

To deal with this difficulty, Celeux and Durand [2008] propose a particular half-sampling procedure in the HMM context. It consists of splitting the learning dataset into two-fold chosen in a deterministic way to maintain the dependencies between the data of each subsequence: these two subsequences correspond to the observations with respectively

the odd and the even indices of the original dataset. They are noted hereafter y^{odd} and y^{even} respectively.

Accordingly, the CVL is computed by permuting the role of these two subsequences: when y^{odd} (resp. y^{even}) is used to estimate Θ_κ, its log-likelihood is estimated on the other subsequence y^{even} (resp. y^{odd}), noted $\log p_{\widehat{\Theta}_\kappa(y^{\text{odd}})}(y^{\text{even}})$ (resp. $\log p_{\widehat{\Theta}_\kappa(y^{\text{even}})}(y^{\text{odd}})$). The proper number of hidden Markov states κ with respect to this odd-even half-sampling (OEHS) criterion is that maximizing

$$-\frac{1}{2}\left(\log p_{\widehat{\Theta}_\kappa(y^{\text{odd}})}(y^{\text{even}}) \;+\; \log p_{\widehat{\Theta}_\kappa(y^{\text{even}})}(y^{\text{odd}})\right). \tag{6.27}$$

However,y^{even} and y^{odd} are dependent which may introduce an optimistic bias on the estimated KL. We propose an alternative deterministic half-sampling procedure, called independent half-sampling (IHS): the two subsequences correspond respectively to the first and second half of the learning dataset. We shall denote $y^1 \triangleq \{y_0, y_1, \ldots, y_{\lfloor T/2 \rfloor}\}$ and $y^2 \triangleq \{y_{\lfloor T/2 \rfloor+1}, y_{\lfloor T/2 \rfloor+2}, \ldots, y_T\}$. The CVL is computed as in the OEHS case. The proper number of hidden Markov states κ with respect to this IHS criterion is that maximizing

$$-\frac{1}{2}\left(\log p_{\widehat{\Theta}_\kappa(y^1)}(y^2) \;+\; \log p_{\widehat{\Theta}_\kappa(y^2)}(y^1)\right). \tag{6.28}$$

6.5 Validation using simulated data

The dataset used in this section is simulated through the SMSSM with number of hidden discrete states $\kappa = 3$ described in Chapter 3. Table 6.1 recalls the parameters values of this model.

	B	C	D	D_0	σ_X^2	σ_Y^2
$s=1$	-1.20	0.11	-350	375	10^{-4}	20
$s=2$	-1.35	0.15	-400	380	10^{-4}	1
$s=3$	-1.30	0.20	-420	385	10^{-4}	1

TABLE 6.1: Parameters values of the SMSSM with $\kappa = 3$ used to generate the simulated dataset

6.5.1 BIC and AIC

Based on the simulated data, both BIC and AIC find the true number of hidden Markov states ($\kappa = 3$) as shown in Figure 6.1. By examining each criterion closely, it is observed that BIC chooses sharply $\kappa = 3$, but AIC hesitates between $\kappa = 3$ and $\kappa = 4$. This is because AIC tends to overestimate the model order. This will emerge clearly when using real data (cf. Chapter 7).

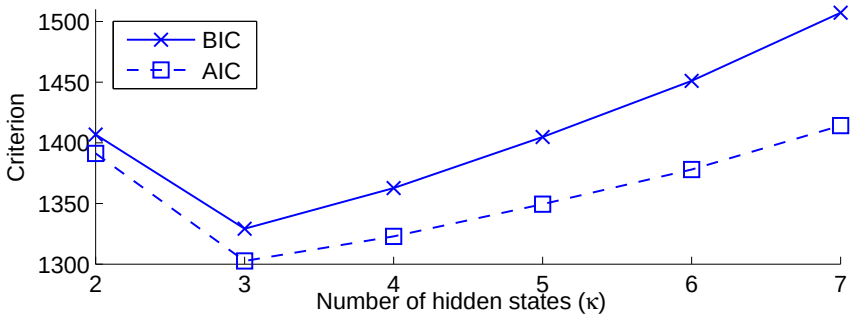

FIGURE 6.1: [Simulated data] BIC and AIC with $\kappa = \{2, \dots, 7\}$. The values of the criteria are represented as a function of the number of hidden states κ

6.5.2 Slope heuristics criterion

Figure 6.2 shows that the slope heuristics criterion gives the true order, *i.e.* $\kappa = 3$. The

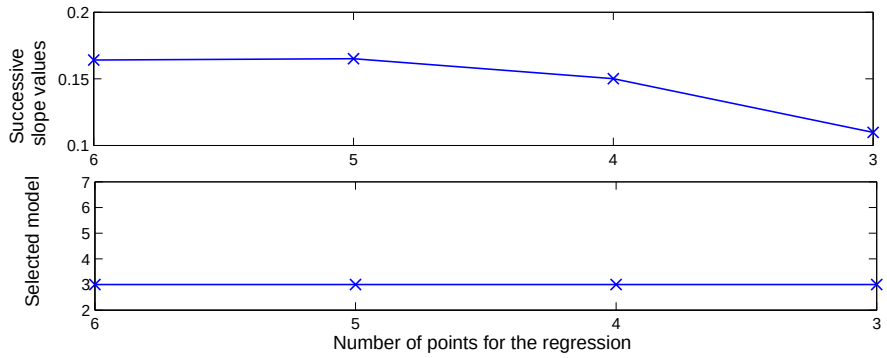

FIGURE 6.2: [Simulated data] The top graph gives the estimated slope as a function of the number of couples $(K(\kappa), \log p_{\hat{\Theta}_\kappa}(y_{0:T}))$ used for the linear regression. The bottom graph gives the selected model with respect to the successive slope values.

slope values are very close (top Figure 6.2). This is expected as the estimated MLs have one abrupt changes in slope (for $\kappa = 3$), before the estimated MLs (for $\kappa = \{4, \dots, 7\}$) become quasi-stable (see Figure 5.4). Figure 6.3 shows the slope criterion with respect to the successive slope values. We note that all slope values choose $\kappa = 3$, and that the heuristics slope criterion is close to being the maximum likelihood model selection criterion (cf. black curve ($\times$) in Figure 6.3) when the slope is small (slope with $5 : 7$ equal to 0.11).

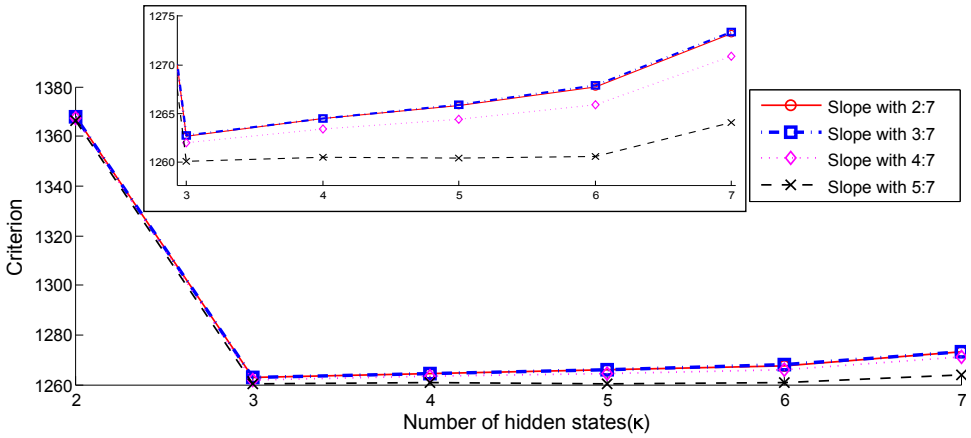

FIGURE 6.3: [Simulated data] The slope heuristics criterion with respect to the successive slope values

6.5.3 CVL criterion

Figure 6.4 shows the results of estimation of the number of hidden states using two CVL criteria: the OEHS and the IHS. It can be observed that both criteria give the true order, *i.e.* $\kappa = 3$. Nevertheless, the IHS chooses sharply $\kappa = 3$, whereas the OEHS tends to under-penalize model complexity. This behaviour will emerge clearly when dealing with real-life data.

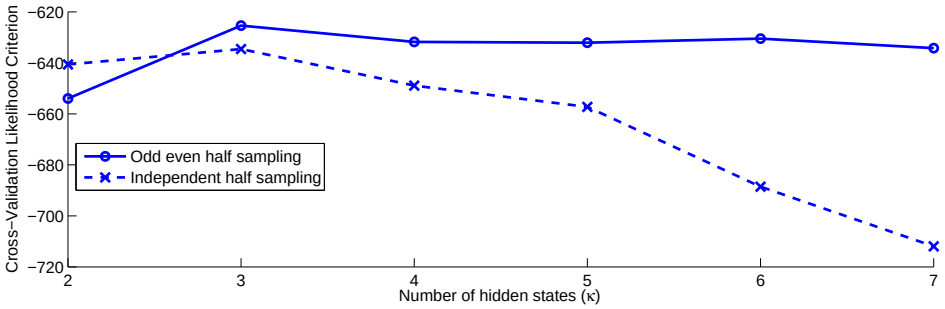

FIGURE 6.4: [Simulated data] OEHS and IHS criteria with $\kappa = \{2, \ldots, 7\}$. The values of CVL are represented as a function of the number of hidden states κ

6.6 Choice of the sampling interval

In real-world applications, the sampling interval can be constant or variable. Indeed, the sensor measurements are communicated to the BMS at fixed sampling times or when the variation of a given physical quantity (for instance, the voltage) exceeds a defined threshold. The later sampling procedure is generally used to optimize the volume of information exchanged between the battery and the BMS, and thus to reduce the allocated memory space as well as the calculation time.

Moreover, the computational complexity of the parameters inference algorithm increases linearly with T the size of the learning dataset. When the learning dataset comprises a test of battery charge and discharge of several hours, an appropriate resampling significantly reduces the parameters inference time.

However, the sampling interval may influence the estimated number of hidden Markov states. Indeed, let us consider a simulated dataset generated from a SMSSM with $\kappa = 2$. For this simulated dataset, we consider that $s_t = 1$ for odd indices and $s_t = 2$ for even ones. If the sampling time is equal to 2, State 2 will be lost, and the estimated number of Markov states will be equal to 1.

To test the impact of the choice of the sampling interval T_s on the estimated number of hidden Markov states κ, a dataset with size $T = 4000$ is generated according to the SMSSM (3.22)-(3.23) with $\kappa = 5$. Figure 6.5 shows the selected κ using BIC and AIC for sampling interval equal to 1, 5 and 10. It can be observed that BIC and AIC select the same κ. For a sampling interval equal to 1 and 5, the selected κ is 5; whereas $\kappa = 3$ is selected when $T_s = 10$. It is noted that for $T_s = 5$, BIC hesitates between $\kappa = 4$ and 5. These results are expected: for small learning dataset size, some states might not be often, if not never, reached.

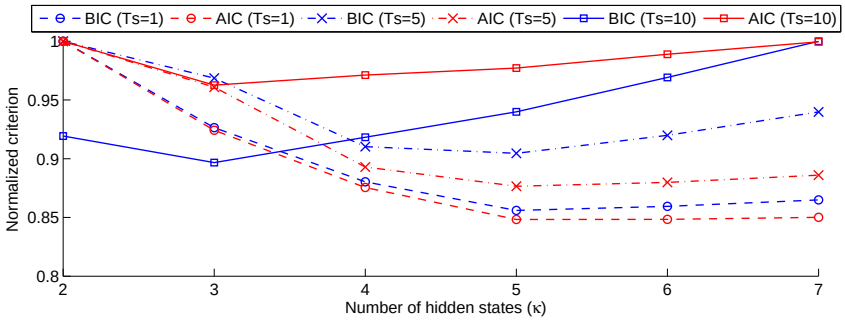

FIGURE 6.5: [Simulated data] Model selection criterion for a sampling interval Te equal to 1, 5, and 10. The values of the criteria BIC and AIC are normalized to make the reading easier, and represented as a function of the number of hidden states κ

From a statistical point of view, this question remains an issue particulary when dealing with a finite mixture model such as SMSSMs. In this study, the parameters are learned using the whole learning dataset with a sampling step equal to one seconde. A careful study of the impact of the sampling step should be conducted to reduce the size of the learning dataset and thus the learning time.

6.7 Discussion

In this chapter, we have provided an overview of the theoretical results of four model selection criteria, namely BIC, AIC, SHC and the CVL criterion. Next, we have verified

their ability to select the proper number of hidden states of the SMSSM (3.22)-(3.23) using a simulated dataset.

The results show that all tested criteria seem to have a satisfactory behavior. However, some of them have a tendency to overestimate the model order (i.e. underpenalize the complexity of a model). This will clearly emerge when using real data (Chapter 7).

One of the main differences between these criteria is the computational complexity. Indeed, BIC and AIC are based on the estimated likelihood of the candidate models penalized with a fixed function depending on the number of free parameters. The slope heuristics criterion aims at finding an optimal penalty function considering the behavior of the ML for large-order models. A linear behavior of the ML with respect to the model order is required to calibrate this penalty function. This could be expensive - in terms of computational time and resources - because in some cases, the ML of very large-order models, *a priori* not included in the set of candidate models, should be calculated to observe this linear behavior. As for AIC, the cross-validation is a promising model selection criterion as it evaluate the predictive performance of the candidate models. In the case of SMSSMs, the use of this approach faces with technical difficulties. Indeed, the data are dependent, and the random selecting of subsequences might break the dependencies between the observations. To overcome this problem, the dataset was divided into two subsequences by a deterministic way, so that the data dependency is maintained. Moreover, the cross-validation is too expensive. Indeed for $\kappa_{\min} \leq \kappa \leq \kappa_{\max}$, a cross-likelihood should be calculated in two steps: the model parameters are estimated using the first subsequence (resp. the second one), and then the likelihood is evaluated for the second subsequence (resp. the first one). The selected model is that maximizing this cross-likelihood. Once the number of hidden states is selected, the corresponding set of model parameters is calculated using the whole learning dataset.

This chapter has also presented a challenging issue for model selection in mixture models context, namely the choice of the sampling interval. Indeed, increasing the sampling interval reduces the computational complexity of parameters inference algorithms, but this affects the selection of the number of hidden states. This is due to the fact that for small dataset size, some states might not be often, if not ever, reached. This challenging issue is will be discussed in the perpectives (cf. Chapter 8).

In the next chapter, the *SoC* of an electric battery is estimated through the SMSSM (3.22)-(3.23). A real-life electric battery data is used to validate the model developed in Chapter 3, as well as the developed algorithms to online estimate the *SoC* (Chapter 3), offline estimate the model parameters (Chapters 4 and 5), and offline estimate the number of hidden states κ (Chapter 6).

Chapter 7

Validation of the SoC switching Markov state-space model using real-life battery data

Ce chapitre est consacré à l'application du modèle proposé de SoC sur des données réelles de charge/décharge des batteries électriques. Cette application comprend l'estimation des paramètres inconnus du modèle, l'estimation du nombre d'états cachés et l'analyse de l'influence de la base d'apprentissage sur les performances du modèle de SoC. En effet, des expérimentations de charge/décharge sont utilisées afin de tester la robustesse du modèle de SoC pour différentes conditions d'usage externes et caractéristiques internes. Dans ce cadre, nous considérons trois types de batteries : cellule, module (constitué de cellules) et pack (constitué de modules).

Pour le premier type, une cellule rechargeable lithium-ion de type S est utilisée. Sa capacité nominale est de $2.2\,Ah$ et sa tension nominale de $3.6\,V$. Plusieurs expérimentations de charge/décharge sont effectuées sur deux cellules neuves de type S pour caractériser la variabilité de leur comportement en fonction de leurs conditions d'usage externes et de leurs caractéristiques internes. Les variations des caractéristiques internes (capacité réelle, OCV, résistance interne) en fonction de divers facteurs (régime de courant, température ambiante, SoC) sont particulièrement étudiées.

Ces expérimentations montrent que la capacité réelle de la cellule augmente lorsque la température ambiante augmente et diminue lorsque le régime de courant augmente. Des tests d'impulsion de courant montrent un léger effet d'hystérésis de l'OCV entre la charge et la décharge (i.e., la relation entre l'OCV et le SoC change selon le signe de courant). Cette relation est donc difficile à identifier et modéliser par des tests en laboratoire. De plus, ces expérimentations montrent l'influence de la température ambiante et du régime de courant sur la relation OCV/SoC est limitée pour cette cellule de type S. Concernant la résistance interne, les expérimentations mettent en évidence qu'elle augmente lorsque

la température ambiante diminue, et lorsque le SoC est inférieur à 20%. De plus, la résistance interne diminue lorsque le régime de courant augmente.

Pour plus de détails sur les spécifications nominales et les tests de caractérisation de la cellule de type S, se référer à la Section 7.1.1.

Pour analyser l'impact du choix de la base d'apprentissage sur les performances du modèle de SoC, nous considérons deux profils élémentaires de puissance, appelés "autoroutier" et "urbain", visant à simuler les conditions d'utilisation d'un pack batteries dans un véhicule électrique roulant respectivement sur une autoroute et dans un milieu urbain. Les caractéristiques électriques et les signaux de courant et de tension mesurés au cours de ces deux profils élémentaires sont présentés dans la Table 7.1 (page 126) et la Figure 7.7. Il convient de noter que la décharge par un profil autoroutier ($\simeq 1h$) est dix fois plus rapide que celle par un profil urbain ($\simeq 10h$).

Afin de simuler plusieurs situations de conduites, quatre profils de courant sont envisagés pour chacun des deux profils élémentaires :

1. décharge complète : le profil de courant élémentaire est répété jusqu'à la décharge complète de la cellule ;
2. décharge partielle : le profil de courant élémentaire est répété jusqu'à ce que 50% de la quantité de charge maximale soit retirée de la cellule ;
3. décharge partielle en deux itérations : pour chaque itération, le profil de courant élémentaire est répété jusqu'à ce que 50% de la quantité de charge maximale soit retirée de la cellule. Ensuite, la cellule est maintenue au repos ($I = 0\,A$) pendant une heure ;
4. décharge partielle en dix itérations : pour chaque itération, le profil de courant élémentaire est répété jusqu'à ce que 10% de la quantité de charge maximale soit retirée de la cellule. Ensuite, la cellule est maintenue au repos ($I = 0\,A$) pendant une heure.

Des expérimentations de décharge sont réalisées en utilisant ces huit profils de courant (4 types $\times$ 2 profils élémentaires) sous deux températures différentes 0 et $25°C$. Ainsi, 16 expérimentations de décharge sont effectuées (8 profils de courant simulés $\times$ 2 températures ambiantes). Pour toutes ses expérimentations, la cellule est initialement complètement chargée sous son régime nominal de charge (température ambiante $25°C$ et régime de courant $C/2$, la cellule est ensuite maintenue à sa tension flottante jusqu'à ce que le courant soit inférieure à C/25). Après l'application du profil de courant la cellule est complètement déchargée aussi sous son régime nominal de décharge (température ambiante $25°C$ et régime de courant $C/2$). La Figure 7.9 montre une représentation schématique de ces quatre types de profils.

Les données (i.e., mesures instantanées de courant et de tension) issues de ces 16 tests sont utilisées comme des bases d'apprentissage pour estimer les paramètres du modèle de SoC à nombre d'états cachés $\kappa = 2, \ldots, 10$.

Nous appliquons l'algorithme MCEM pénalisé par les contraintes d'identifiabilité pour estimer ces paramètres. Pour chaque base d'apprentissage, nous utilisons ensuite les critères BIC, AIC et SHC pour sélectionner le nombre d'états cachés parmi les neuf modèles candidats.

Pour chaque critère de sélection de modèle, ces 16 expérimentations de décharge nous fournissent ainsi 16 modèles de SoC. Finalement, chacun de ces modèles est validé sur les 16 expérimentations. De ce fait, il y a 256 combinaisons possibles.

À cause du nombre important de combinaisons, nous choisissons d'utiliser uniquement le modèle sélectionné par BIC. De plus, pour évaluer les performances d'un modèle donné, nous nous basons sur l'écart entre le SoC estimé par ce modèle et le SoC coulométrique. En effet, pour les expérimentations réalisées en laboratoire, nous considérons le SoC coulométrique comme une valeur de référence pour les raisons suivantes :

- la valeur initiale de SoC est connue : $SoC_0 = 100\%$;
- le capteur de courant est très précis ;
- la capacité réelle est connue : calculée hors ligne à la fin de l'expérimentation.

Le choix de la base d'apprentissage a une influence sur les performances du modèle estimé. En termes de capacité de prédiction, les 256 tests de validation du modèle sur la cellule de type S mettent en évidence les résultats suivants :

- un modèle appris à partir d'une décharge partielle est moins performant qu'un modèle appris à partir d'une décharge complète ;
- Le modèle appris avec une température donnée est plus performant pour estimer le SoC à la même température qu'avec un autre ;
- le modèle appris à partir d'une expérimentation de décharge complète par un profil de courant de type 4 paraît le plus robuste.

Pour une description complète des phases d'apprentissage et de validation, se référer à la Section 7.1.3.

Pour valider le modèle de SoC sur un module, nous considérons un module de batteries, noté "M60", composé de 60 cellules rechargeables lithium-ion de type $LiFeO_4$, noté par la suite type L. Sa capacité nominale est de $75\,Ah$ et sa tension nominale de $38.4\,V$.

Comme dans le cas d'une cellule, le module est soumis à plusieurs expérimentations de charge/décharge pour illustrer la variation de la capacité réelle, l'OCV et la résistance interne en fonction du régime de courant, la température ambiante et le SoC.

Les résultats montrent que la capacité réelle augmente lorsque le régime de courant et la température ambiante augmentent. Toutefois, il convient de noter que l'influence de la température ambiante est plus importante que celle du régime de courant. Les tests à impulsions de puissances mettent en évidence un fort effet d'hystérésis de l'OCV entre la charge et la décharge, et un faible changement de l'OCV lorsque le $SoC \in [20\%, 80\%]$. Ce comportement est lié à la technologie des cellules formant le module ($LiFeO_4$). Les méthodes d'estimation de SoC basées sur un OCV mesuré/estimé pourraient ainsi se révéler inefficaces pour ce module. Concernant la résistance interne, les résultats montrent que pendant la charge (resp. décharge), la résistance interne du module a tendance à augmenter (resp. diminuer) lorsque le SoC augmente.

Pour plus de détails sur les spécifications nominales et les tests de caractérisation du "M60", se référer à la Section 7.2.1.

Pour valider le modèle de SoC, nous considérons deux profils élémentaires de courant appelés "NEDC" pour New European Driving Cycle et "Michelin". Le premier profil représente la puissance qui serait appliquée à un pack batteries lors d'un cycle européen d'homologation NEDC. Le second profil est extrait d'un contrôle au cours d'essais sur un véhicule électrique "WILL" réalisé dans le cadre du projet Forewheel en partenariat avec Michelin. Ce profil a été ajusté selon les spécifications électriques du module "M60". Les caractéristiques électriques et les signaux de courant et de tension mesurés au cours de ces deux profils sont présentés dans la Table 7.4 (page 140) et Figure 7.27. Douze expérimentations de décharge sont réalisées pour simuler plusieurs situations de conduite :

1. décharge complète : le profil NEDC (resp. Michelin) est répété jusqu'à la décharge complète du module sous trois différentes températures ambiantes -5, 5, et $25°C$;
2. décharge partielle : le profil NEDC (resp. Michelin) est répété jusqu'à ce que 30% de la quantité de charge maximale soit retirée du module sous trois différentes températures ambiantes -5, 5, et $25°C$.

Pour toutes ses expérimentations, le module est initialement complètement chargé sous son régime nominal de charge (température ambiante $25°C$ et régime de courant $C/2$). Après l'application du profil de courant le module est complètement déchargé aussi sous son régime nominal de décharge (température ambiante $25°C$ et régime de courant $C/2$). La validation est réalisée de la même façon que celle de la cellule, à une différence près : pour chaque critère de sélection de modèle, nous avons 144 combinaisons possibles. En termes de capacité de prédiction, les 144 tests de validation du modèle sur le module "M60" mettent en évidence les résultats suivants :

- un modèle appris à partir d'une décharge partielle est moins performant qu'un modèle appris à partir d'une décharge complète ;
- le modèle est robuste à la variation de la température ambiante de la base d'apprentissage ;
- le modèle appris à partir d'une expérimentation de décharge complète à $5°C$ paraît le plus robuste.

Pour une description complète des phases d'apprentissage et de validation, se référer à la Section 7.2.3.

Pour le cas d'un pack, nous considérons un véhicule électrique, appelé "WILL", équipé d'un pack batteries composé de 10 modules de type "M60", soit 120 étages connectés en série. Sa capacité nominale est de $75\,Ah$ et sa tension nominale de $400\,V$.

Des expérimentations ont été effectuées sur 23 mois comprenant à la fois des périodes de stockage et d'usage du pack. Le processus global de ces expérimentations est constitué de trois différents modes d'utilisation :

1. roulage du WILL sur un circuit privé afin de contrôler les conditions de la route et le cycle de roulage. Un même profil de vitesse est répété par un conducteur professionnel. Ce profil contient toutes les phases typiques d'une conduite réelle : démarrage, arrêt, accélération, décélération, vitesse constante, etc. Ce roulage se déroule en 5 périodes ;
2. stockage de longue durée du pack sans charge ni décharge ;
3. test en laboratoire de charge/décharge du pack en utilisant un profil reproduisant le même profil de puissance du roulage réel du mode 1.

Des contrôles (checkups) ont été effectués d'une façon régulière pour suivre l'évolution des performances du pack, soit 9 checkups sur les 23 mois. La Figure 7.36 présente le déroulement global des expérimentations sur ce pack. Il est clair que l'état de santé du pack se dégrade au fil du temps : lors du checkup 1, le $SoH = 100\%$, alors que lors du checkup 9, le $SoH = 85\%$.

Ces checkups mettent en évidence l'hétérogénéité des états des cellules élémentaires composant le pack. En effet, pendant les expérimentations de décharge, la tension de certains étages atteint son minimum alors que la tension d'autres étages est encore élevée (cf. Figure 7.38). De ce fait, le SoC n'est pas le même pour tous les étages (cf. Figure 7.40). Un calcul hors-ligne de la capacité réelle de chaque étage met en évidence l'hétérogénéité des capacités réelles des étages. De plus, l'écart entre ces capacités augmente lorsque l'état de santé du pack se dégrade (cf. Figure 7.39). Par conséquent, cette hétérogénéité rend plus compliquée l'estimation du SoC dans le cas du pack de batteries.

Pour l'apprentissage et la validation du modèle de *SoC* proposé, nous procédons de la manière suivante : pour chaque checkup, les paramètres inconnus de neuf modèles de *SoC* à $\kappa = 2, \ldots, 10$ sont estimés par l'algorithme MCEM à partir d'une base d'apprentissage issue d'une expérimentation de décharge complète du pack ; le critère BIC est ensuite utilisé pour sélectionner le modèle le plus adéquat parmi ces neuf candidats ; ce modèle sélectionné est enfin validé sur des données issues des cinq périodes de roulage réel du "WILL". Pour la période 5, nous considérons deux roulages : le premier effectué le matin et le second effectué le soir pour tester l'effet de la température intérieure du pack sur les performances du modèle. En effet, le soir la température du pack est supérieure à celle du matin, du fait que le véhicule a effectué plusieurs tours sur le circuit. Nous avons ainsi 54 (i.e., 9 checkups $\times$ 6 roulages) combinaisons possibles. En termes de capacité de prédiction, ces 54 tests de validation montrent que la variation de l'état de santé du pack a un effet limité sur les performances du modèle de *SoC* développé et que l'impact du changement des températures ambiante et interne est plus important.

Pour une description complète des phases d'apprentissage et de validation, se référer à la Section 7.3.2.

La validation du modèle de *SoC* proposé à partir des données d'usage réel de trois types de batteries (soit cellule, module et pack) met en évidence les avantages potentiels et l'utilité pratique des modèles à espaces d'états gouvernés par une chaîne de Markov pour estimer le *SoC* sous des conditions d'usage non contrôlables. Toutefois, le choix de la base d'apprentissage influe sur la capacité de prédiction du modèle. En effet, un modèle appris à partir d'une expérimentation de décharge complète est meilleur en terme de performance de prédiction qu'un modèle appris à partir d'une expérimentation de décharge partielle. De plus, il apparaît que le type du profil de courant affecte les performances du modèle. Par exemple, le modèle appris à partir du profil de type 4 dans le cas d'une cellule et du profil Michelin dans le cas d'un module est plus robuste aux changements des conditions d'usage que les autres modèles. De ce fait, le choix de la meilleure base d'apprentissage reste une question ouverte. Néanmoins, lorsqu'un ensemble de bases est disponible, une validation croisée comme celle effectuée dans ce chapitre permet d'identifier le modèle ayant la meilleure capacité de prédiction. Cependant, le temps nécessaire pour effectuer une telle procédure est relativement long.

Contents

The aim of this chapter is to apply and validate the proposed SoC model using real-life battery data. Indeed, charge/discharge experiments are performed to test the robustness of the SoC model under different external usage conditions and internal characteristics. For this purpose, three types of electric batteries are used: cell, module and pack.

For the cell battery, a lithium-ion rechargeable cell of type S is used. Its nominal capacity is equal to $2.2\,Ah$ and its nominal voltage $3.6\,V$. Several charge/discharge experiments are carried out to characterize the variability of the cell behavior according to its external usage conditions and its internal characteristics. The variation of the actual capacity, OCV and internal resistance according to the current rate, ambient temperature and SoC is particularly tested. In order to validate the SoC model, two elementary profiles, called "highway" and "urban", simulating the discharge current profile of an electric vehicle when driving respectively on a highway and an urban area, are considered.

Four types of current profiles, formed by each of these elementary profiles, are then designed to simulate different driving situations. Discharge experiments are performed using the 8 simulated current profiles (i.e., 4 types $\times$ 2 elementary profiles) at two different ambient temperatures (0 and $25°C$). Hence, 16 discharge experiments (8 simulated current profiles $\times$ 2 ambient temperatures) are carried out. These experiments are thoroughly described in Section 7.1. The data (i.e., current and voltage measurements) collected during each experiment is used as a learning dataset to estimate the parameters of the SoC model with number of hidden states $\kappa = 2, \ldots, 10$. The parameters are estimated using the MCEM algorithm penalized with the identifiability constraints. BIC, AIC and SHC are then used to select a model amongst the 9 candidate ones. Finally, this learned SoC model is validated for each of the 16 discharge experiments. Given a model selection criterion, there are thus 256 (i.e., 16×16) possible combinations. Since the number of combinations is high, we choose to use the model selected with BIC. Moreover, to evaluate the performance of the learned model, the SoC estimated using the learned SMSSM is compared with the SoC calculated using the Ah-counting method.

For the module batteries, a module, called "M60", formed by connecting 60 lithium-ion rechargeable cells of type L is used. Its nominal capacity is equal to $75Ah$ and its nominal voltage $38.4\,V$. As for the cell of type S, several charge/discharge experiments are carried out to characterize the variation of its actual capacity, internal resistance and OCV according to the current rate, ambient temperature and SoC. Then, two elementary profiles, called "NEDC" for New European Driving Cycle and "Michelin",

are considered. The former aims at simulating a typical usage of a car in Europe; whereas the latter is provided by the Michelin company to evaluate the performances of the pack batteries in laboratory conditions. Twelve discharging experiments using these two elementary profiles at different ambient temperatures are performed. These experiments are fully described in Section 7.2. The validation is then achieved as in the cell case. A particular difference is that the number of possible combinations is equal to 144 (i.e., 12×12) given a model selection criterion.

For the pack batteries, we consider an electric vehicle, called "WILL", equipped with a pack batteries formed by connecting 10 "M60" modules. Its nominal capacity is equal to 75 Ah and its nominal voltage 400 V. Experiences have been conducted during 23 months comprising usage and storage periods. The overall experiment process consists of three modes: driving the vehicle on a private circuit, long period of storage, and laboratory tests of charge/discharge. During these 23 months, 9 checkups are regularly performed to monitor the evolution of the electric performances of the pack. These checkups highlight the heterogeneity of the stages constituting the pack making the estimation of the SoC more complicated. These experiments are fully described in Section 7.3. The validation is then achieved as in the two previous cases. Only several experiments are considered in the validation phase due to the big size of data collected during the overall process, namely 27 Go.

Based on the validation using these three types of batteries, the advantages and limitations of the proposed SoC model are thoroughly detailed in Section 7.4.

7.1 Cell battery of type S

7.1.1 Cell performances

The experiments presented in this section have been performed by CEA/LITEN/DTS/S3E/LSEC.

7.1.1.1 Nominal specifications

In order to validate the proposed SoC model for a cell battery, a lithium-ion rechargeable cell of type S, with the following nominal specifications, is used:

- Nominal capacity: 2.2 Ah,
- Nominal voltage: 3.6 V,
- Float voltage: 4.2 V,
- Cut-off voltage: 2.75 V,
- Nominal charge/discharge current rate: $C/2$,
- Nominal ambient temperature: 25°C.

7.1.1.2 Complete discharge performance

Several laboratory experiments are performed to illustrate the variability of the behavior of this cell according to its internal characteristics and external usage conditions. As a first step, two new and fully-charged cells of type S are completely discharged with current rates equal to $C/2$, $1C$ and $2C$ at four different ambient temperatures, namely at 0, 10, 25 and 45°C. Accordingly, the number of experiments to be carried out is 32 (number of cells × number of different current rates × number of different ambient temperatures). Figure 7.1 shows the actual cell capacity during these discharge experiments. Apart from discharging with $C/2$ at 0°C, the two cells appear to have the same actual capacity. The empirical mean of these actual capacities is presented in Figure 7.2. It can be observed that the maximum amount of charge that can be removed from the cell increases when the ambient temperature increases; whereas this maximum amount of charge decreases when the current rate increases (i.e., when the battery is discharged more rapidly). The actual capacity is even larger than the nominal capacity for discharges at 45°C. These results emphasize that the estimation of the SoC using solely the Ah-counting method might be inefficient, especially when neither the ambient temperature nor the current rate are controlled, as in real-life contexts. In addition, the estimated SoC with respect to the nominal capacity might be $< 0\%$ or $> 100\%$.

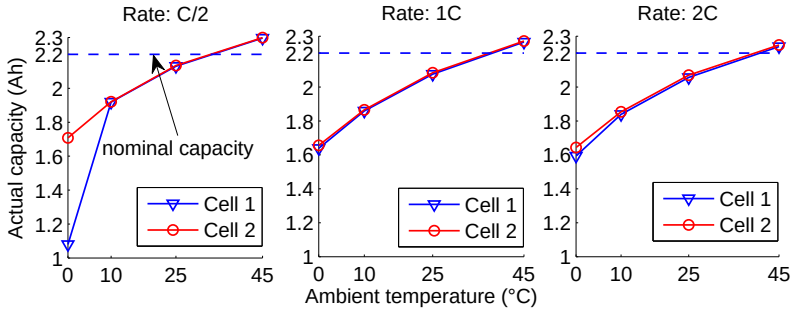

FIGURE 7.1: Maximum amount of charge that can be removed from two new cells of type S with discharge current rate equal to $C/2$, $1C$ and $2C$ at different ambient temperatures (0, 10, 25, 45°C)

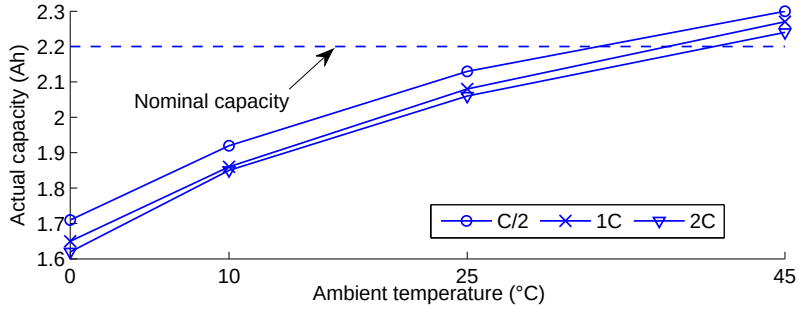

FIGURE 7.2: Empirical mean of the maximum amount of charge that can be removed from the two new cells of type S with discharge current rate equal to $C/2$, $1C$ and $2C$ at different ambient temperatures (0, 10, 25, 45°C)

7.1.1.3 Open circuit voltage

As a second step, a new cell of type S is discharged with an impulse current of rate $1C$ and $2C$ at 0, 10, 25, and $45°C$. After each current impulse, the SoC is calculated and the cell is maintained at rest (i.e., $I = 0A$) in order to calculate its corresponding OCV and internal resistance. Figure 7.3 shows the OCV as a function of the SoC for these different impulse discharge tests. Apart from discharging with $1C$ at $25°C$, it can be observed that when the $SoC \in [20\%, 90\%]$ the impact of the current rate and the ambient temperature on the OCV is limited. It is noteworthy that some values of the SoC are larger than 100 as the SoC is calculated with respect to the nominal capacity.

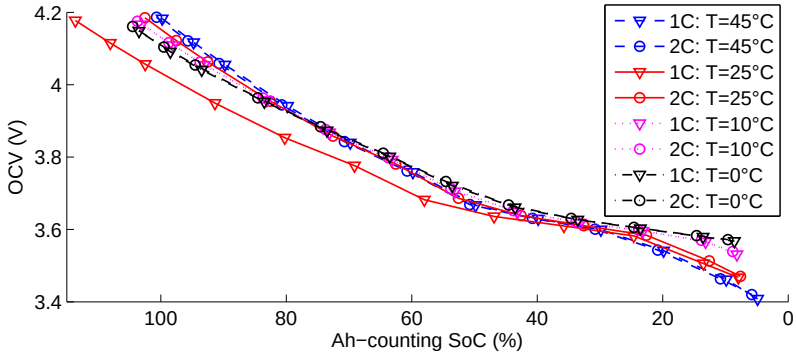

FIGURE 7.3: OCV vs. SoC of a new cell of type S with impulse discharge current of $1C$ and $2C$ at different ambient temperatures (0, 10, 25, $45°C$)

Figure 7.4 presents the OCV according to the SoC, for impulse charge and discharge current of rate $1C$ at 0, 10, 25, and $45°C$. These results show a minor hysteresis effect between charge and discharge. Nevertheless, this is more visible at 0, 10, and $25°C$ than at $45°C$. This hysteresis effect depends widely on the battery type, and would be stronger as shown in the case of module batteries presented in Section 7.2.

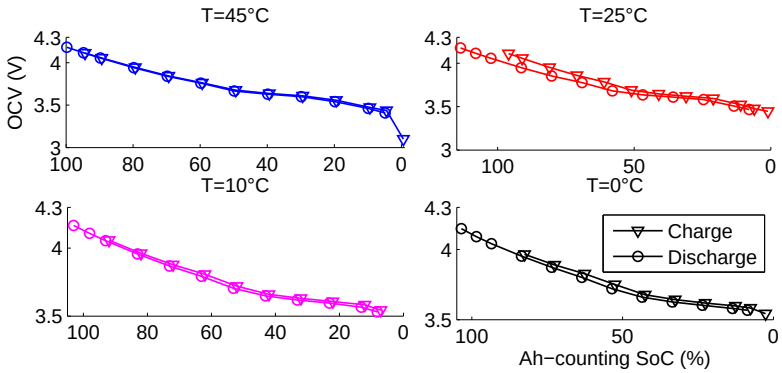

FIGURE 7.4: Hysteresis effect between charge ($\triangledown$) and discharge ($\circ$) of a new cell of type S with impulse charge and discharge of current rate $1C$ at different ambient temperatures (0, 10, 25, $45°C$)

7.1.1.4 Internal resistance

In Figure 7.5, the internal resistance is presented with respect to the SoC for $1C$ and $2C$ impulse discharges at 0, 10, 25, and $45°C$. Three major points emerge from these results:

1. the internal resistance increases when the SoC decreases, especially when the $SoC \in [0\%, 20\%]$;
2. the internal resistance increases when the ambient temperature decreases;
3. the internal resistance decreases when the current rate increases. This phenomenon can be described based on the Butler-Volmer equation (see for instance Montaru [2009] for real cell data).

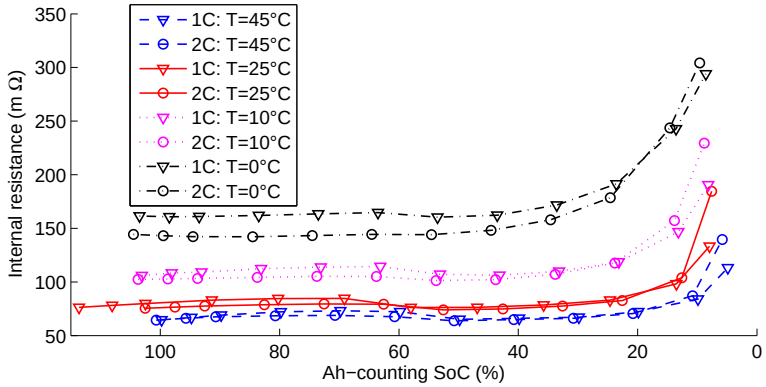

FIGURE 7.5: Internal resistance vs. SoC of a new cell of type S with impulse discharge current of $C/2$, $1C$ and $2C$ at different ambient temperatures (0, 10, 25, $45°C$)

The impact of the charge and discharge on the internal resistance is presented in Figure 7.6. It can be observed that when the $SoC \in [20\%, 100\%]$ the internal resistance is almost equal for charge and discharge. However when the $SoC \in [0\%, 20\%]$, the internal resistance during discharge is larger than the one during charge.

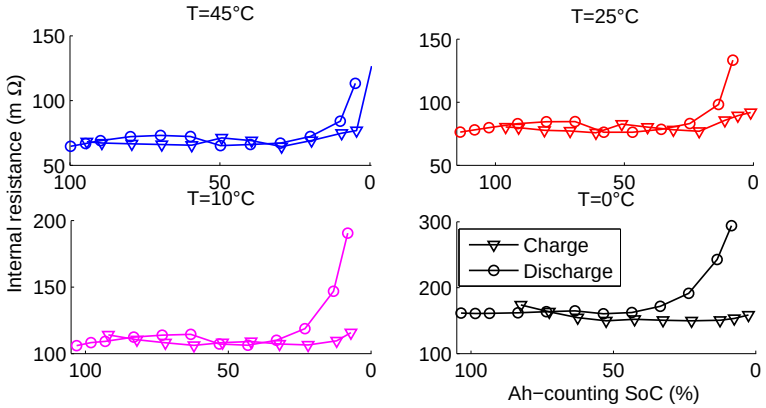

FIGURE 7.6: Internal resistance vs. SoC of a new cell of type S with impulse charge ($\triangledown$) and discharge ($\circ$) current of $1C$ at different ambient temperatures (0, 10, 25, $45°C$)

7.1.2 Validation protocol

The validation protocol aims at identifying the advantages as well as the limitations of the proposed SoC model. In order to test the robustness of the SoC model for different types of current profile, two elementary current profiles of 15.3 minutes are simulated. The first one, called "highway profile", aims at modeling the discharge current profile of an electric vehicle when driving on a highway. The second one, called "urban profile", aims at modeling the discharge of an electric vehicle during an urban drive. The electrical characteristics of these two profiles are summarized in Table 7.1. Accordingly, a complete discharge of a cell of type S with the highway profile (duration $\simeq 1h$) is faster than that with the urban profile (duration $\simeq 10h$) as the highway mean power is much greater than the urban one. Moreover, the highway current peaks, thus its maximum discharged power, are larger than the urban current peaks (cf. Figure 7.7). Another characteristic of the highway profile is that almost all the time the current is negative; whereas the current of the urban one changes signs relatively quickly due to accelerations and decelerations during an urban drive. Figure 7.8 shows the current and the voltage measurements during a complete discharge of a cell of type S with the highway and urban profiles.

	Highway profile	Urban profile
Mean power (W)	−8.1	−0.8
Maximum recovered power (W)	9.1	7.6
Maximum discharge power (W)	−19.6	−11.1
Cumulated energy (Wh)	−2.1	−0.2
Profile duration (min)	15.3	15.3
Duration of a complete discharge (h)	0.9	9.3
Number of profiles to complete discharge	3.7	36.6

TABLE 7.1: Electrical characteristics of the highway and urban profiles

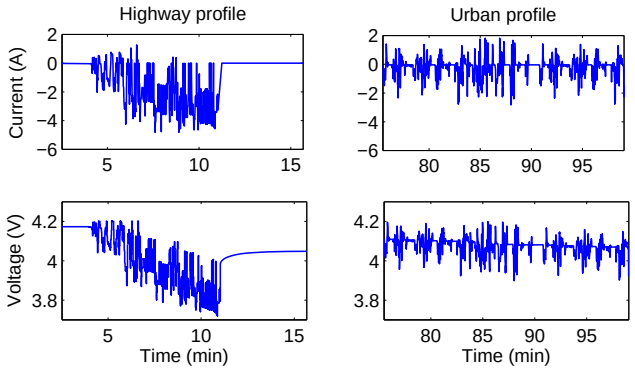

FIGURE 7.7: Highway and urban profiles: current and voltage measurements

These two elementary profiles are used to design four types of current profiles representing different driving situations:

1. complete discharge using the elementary profile;

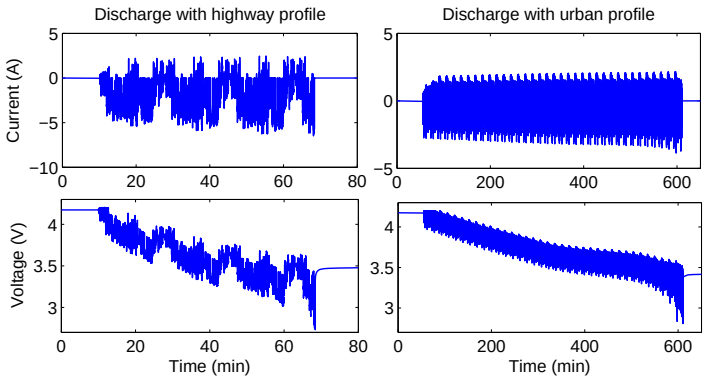

FIGURE 7.8: Discharge with the highway and urban profiles: current and voltage measurements

2. partial discharge: 50% of the maximum amount of charge is removed from the cell using the elementary profile;
3. two iterations of partial discharge: at each iteration, 50% of the maximum amount of charge is removed from the cell using the elementary profile. The cell is then maintained in rest for one hour;
4. ten iterations of partial discharge: at each iteration, 10% of the maximum amount of charge is removed from the cell using the elementary profile. The cell is then maintained in rest for one hour.

For all types of current profiles, the cell is initially charged under its nominal charge regime (i.e., ambient temperature: $25°C$, and current rate: $C/2$). Then, after the end of the profile, it is completely discharged also under its nominal discharge regime (i.e., ambient temperature: $25°C$, and current rate: $C/2$). A schematic representation of these types is provided in Figure 7.9. Hereafter, Hx (resp. Ux) denotes a current profile of type x with highway (resp. urban) elementary profile.

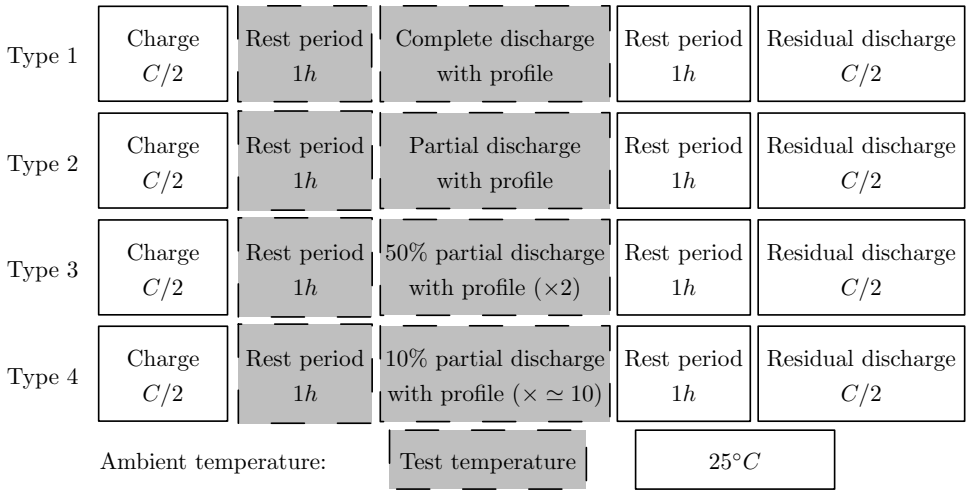

FIGURE 7.9: Four types of current profiles formed by the highway and urban elementary profiles. For type 2, the partial discharge with an elementary profile aims to remove approximately 50% of the maximum amount of charge. For type 3 (resp. 4), each partial discharge with an elementary profile aims to remove approximately 50% (resp. 10%) of the maximum amount of charge.

Different cells of type S are then discharged using these 8 current profiles (4 types $\times$ 2 elementary profiles) at two different ambient temperatures (0 and 25°C). Thus, 16 discharge experiments (8 current profiles $\times$ 2 ambient temperatures) are conducted.

7.1.3 Application of the SoC SMSSM using cell data

Figure 7.10 shows the global procedure of learning and validation of the *SoC* SMSSM developed hereafter.

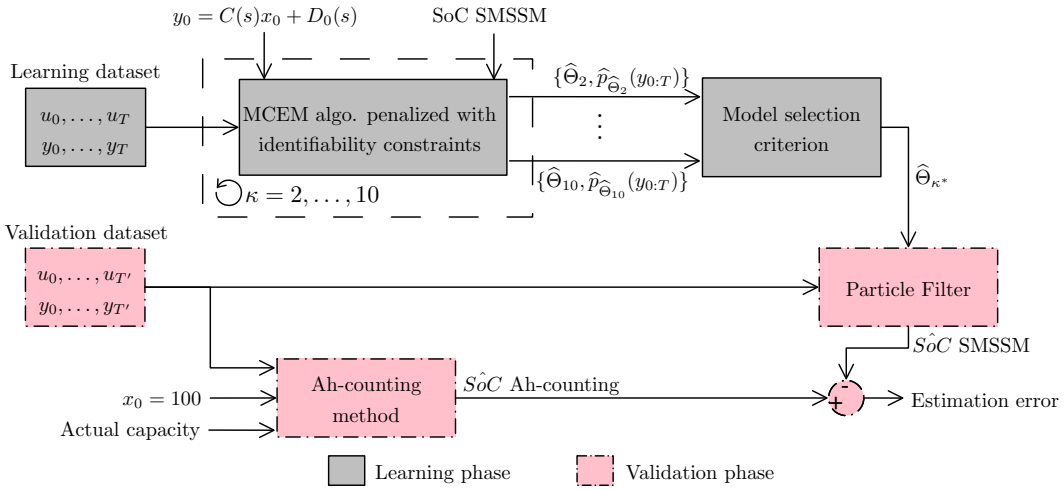

FIGURE 7.10: Validation protocol: learning and validation phases

7.1.3.1 Learning datasets

The data (i.e., voltage and current) collected, with sampling time $T_s = 1s$, during each of these 16 experiments is considered as a learning dataset. The size of these datasets are reported in Table 7.2.

	H1	H2	H3	H4	U1	U2	U3	U4
0°C	2250	2000	2500	3000	8400	5800	8355	8520
25°C	3500	1904	3377	3701	10832	5746	10690	10605

TABLE 7.2: Size of the dataset collected during the 16 discharge experiments. Hx (resp. Ux) denotes a current profile of type x with the highway (resp. urban) elementary profile.

7.1.3.2 Estimation of the unknown parameters

For each of these learning datasets, the unknown parameters of *SoC* models with $\kappa = 2, \ldots, 10$ are estimated using the MCEM algorithm penalized with identifiability

constraints. Indeed, for these experiments, the initial value of the SoC and its corresponding voltage are known (i.e., $SoC_0 = 100\%$ and $y_0 = 4.2\,V$) since the cell is initially fully-charged under its nominal regime. Accordingly, the following relationship is available at $t_0 = 0$

$$4.2 = 100 \times C(s) + D_0(s), \tag{7.1}$$

where $s = 1, \ldots, \kappa$.

In addition, to avoid a local Maxima of Likelihood, for a given learning dataset and number of Markov states κ, this algorithm is repeated 20 times. The estimated model having the maximum likelihood is then adopted.

Moreover, as shown in Section 5.5.1.2 the estimated ML is affected by the number N of simulated particles. For these experiments we use $N = 200$ for $\kappa = 2, 3, 4$; $N = 500$ for $\kappa = 5, 6, 7$; and $N = 750$ for $\kappa = 8, 9, 10$. This choice is determined based on the dispersion of the estimated ML of the largest learning dataset (i.e., $T \simeq 10^4$) amongst the 16 conducted experiments. Indeed, for each κ, tests similar to those described in Section 5.5.1.2 are performed to choose the appropriate number of particles.

7.1.3.3 Estimation of the number of hidden discrete states

As a result of the above, and for each learning dataset, there are nine candidate SMSSMs ($\kappa = 2, \ldots, 10$). The criteria BIC, AIC and SHC are then used to select an adequate model among the nine models. Table 7.3 summarizes the selected number of hidden states for the 16 learning datasets. It is noteworthy that for the SHC, the corresponding selected κ is not reported if a linear behavior of the ML is not observed. It can be seen that in almost all cases, these three criteria lead to the selection of different κ. In addition, it is complicated to establish a relation between the number of hidden states and the ambient temperature or the type of the current profile alike.

$T = 0°C$	H1	H2	H3	H4	U1	U2	U3	U4
BIC	6	6	6	3	7	5	7	3
AIC	7	6	9	8	7	5	8	8
SHC	7	6	-	3	-	-	-	-
$T = 25°C$	H1	H2	H3	H4	U1	U2	U3	U4
BIC	7	5	9	6	4	8	8	4
AIC	7	9	10	10	5	9	9	6
SHC	7	5	3	6	5	4	3	6

TABLE 7.3: Selected number of hidden states κ for the SoC SMSSMs of the cell of type S with AIC, BIC and SHC. There are 16 models learned using the 16 datasets collected during the discharge experiments with the 8 different current profiles at ambient temperatures equal to 0 and $25°C$.

To explain these selection results, we focus on two cases:

- H4 at $0°C$ for which BIC and SHC select $\kappa = 3$, whereas AIC selects $\kappa = 8$. These two SMSSMs with $\kappa = 3$ and 8 are validated using the 16 experiments (the validation phase is detailed in the next section). In Figure 7.11, it can be observed that for almost all validations, the mean absolute error of the model selected with AIC is smaller than that selected with BIC, which is consistent with the theoretical results. However, the prediction gain is too small (maximum 1%), and selecting $\kappa = 3$ makes more sense as it requires less computational capacity which is in line with embedded applications requiremnts.
- H3 at $25°C$ for which BIC and AIC select respectively $\kappa = 9$ and $\kappa = 10$, whereas SHC select $\kappa = 3$. Figure 7.12 shows the estimated ML with respect to the number of hidden discrete states. Despite the fact that a linear behavior of the ML with respect to κ is not clearly observed, it seems that the SHC detects the linear piece for $\kappa = 3, 4, 5$.

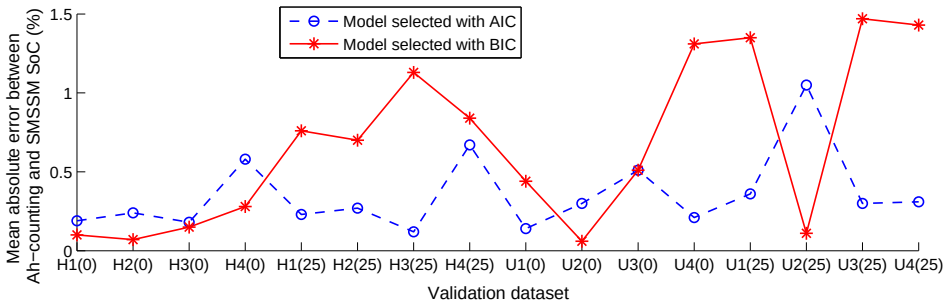

FIGURE 7.11: Mean absolute error between Ah-counting and SMSSMs *SoC* learned using H4 at $0°C$ and selected with AIC and BIC. The ambient temperature of each experiment is written in parenthesis.

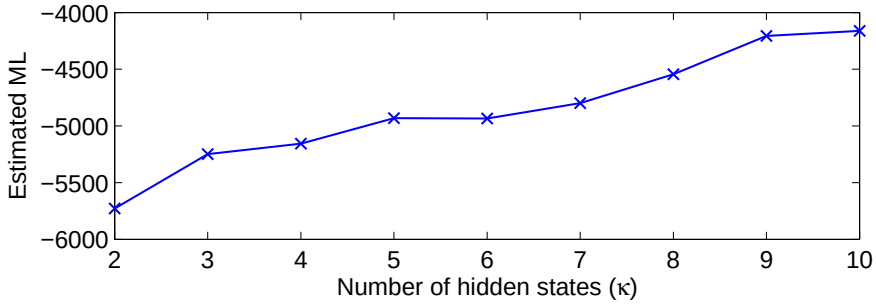

FIGURE 7.12: Estimated ML for SMSSMs with $\kappa = 2, \ldots, 10$ learned using a discharge experiment with H3 at $25°C$

7.1.3.4 Validation phase

A learned *SoC* model is finally validated for each of the 16 learning datasets. Accordingly, given a model selection criterion, there are 16×16 (i.e., 256) possible combinations.

Due to the high number of possible combinations, validation work is limited to the model selected with BIC.

For these laboratory experiments, the Ah-counting SoC is considered as the reference SoC for the following reasons:

- the initial SoC is known: $SoC_0 = 100\%$;
- the current sensor is very accurate;
- the actual capacity is known: it is calculated offline at the end of the experiment by integrating the current that flows through the battery.

Learning using highway profiles - Figures 7.13 and 7.14 show the results of the validation of the models learned using the 8 discharge experiments with H1, H2, H3 and H4 at 0 and 25°C. Figure 7.13 presents the maximum of the estimation error between the Ah-counting SoC and the SoC estimated using these learned SMSSMs; whereas Figure 7.14 presents the mean absolute error.

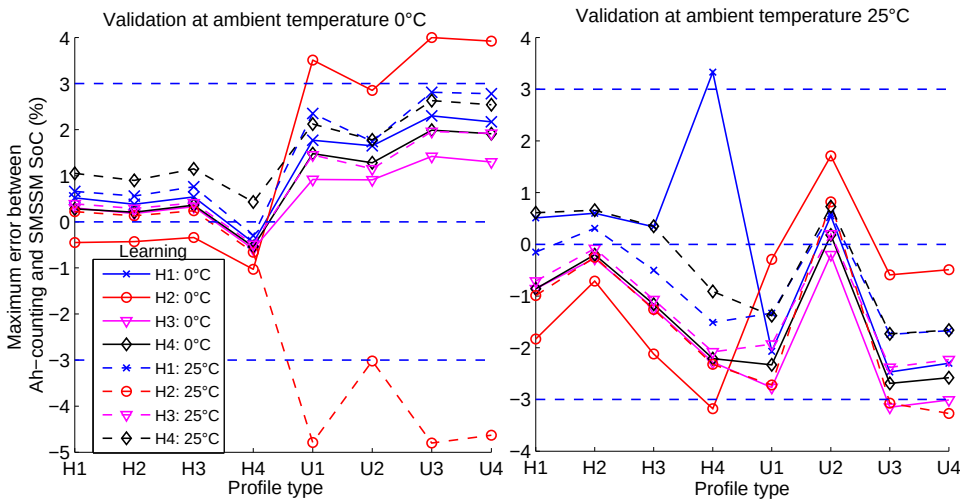

FIGURE 7.13: Maximum error between the Ah-counting and the SMSSM SoC: learning with the four simulated current profiles formed of the highway elementary profile at ambient temperature equal to 0°C and 25°C, and validation using the eight simulated current profiles at ambient temperature equal to 0°C and 25°C. The learned model is selected with BIC. Horizontal dot-lines represent +3%, 0% and−3%.

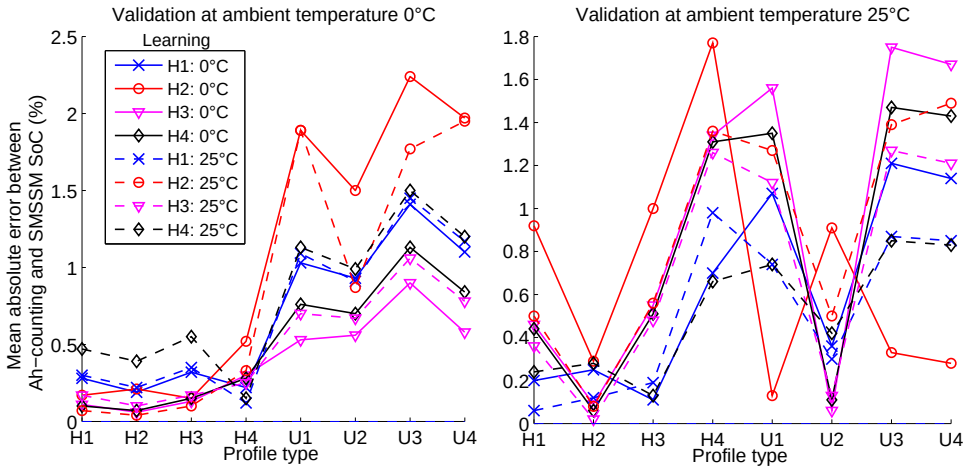

FIGURE 7.14: Mean absolute error between the Ah-counting and the SMSSM SoC: learning with the four simulated current profiles formed of the highway elementary profile at ambient temperature equal to $0°C$ and $25°C$, and validation using the eight simulated current profiles at ambient temperature equal to $0°C$ and $25°C$. The learned model is selected with BIC.

It can be observed that the prediction capacity of the model learned with a partial discharge (i.e., H2 at 0 and $25°C$) is less powerful than that of the other models. This was expected as for these two experiments only 50% of the maximum amount of charge is removed from the cell. Indeed, the particular behavior of the battery at a low voltage and SoC is not integrated in the learning dataset H2 (cf. Figure 7.8 and Figure 7.19 -left-).

Moreover, these results show that when the model is learned using the highway elementary profile, the estimated SoC when discharging with highway current profiles is more accurate than that estimated when discharging with urban current profiles.

In addition, these results emphasize that the behavior of the battery for $SoC \in [50\%, 100\%]$ is well-estimated in all cases. Indeed, the estimated SoC corresponding to partial discharge experiments with current profiles of type 2 (i.e., H2 and U2 at 0 and $25°C$) is more accurate than that in the other experiments.

Learning using urban profiles - Figures 7.15 and 7.16 show the results of the validation of the models learned using the 8 discharge experiments with U1, U2, U3 and U4 at 0 and $25°C$. Figure 7.15 presents the maximum of the estimation error between the Ah-counting SoC and the SoC estimated using these learned SMSSMs; whereas Figure 7.16 presents the mean absolute error.

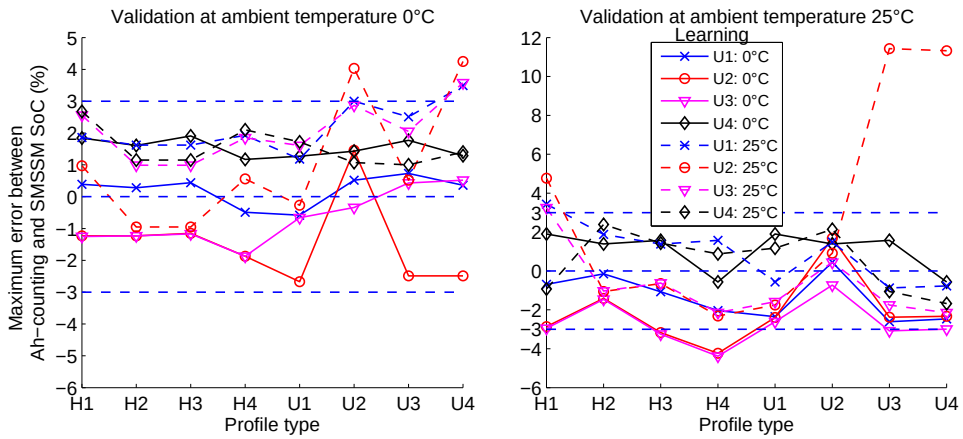

FIGURE 7.15: Maximum error between the Ah-counting and the SMSSM SoC: learning with the four simulated current profiles formed of the urban elementary profile at ambient temperature equal to $0°C$ and $25°C$, and validation using the eight simulated current profiles at ambient temperature equal to $0°C$ and $25°C$. The learned model is selected with BIC.

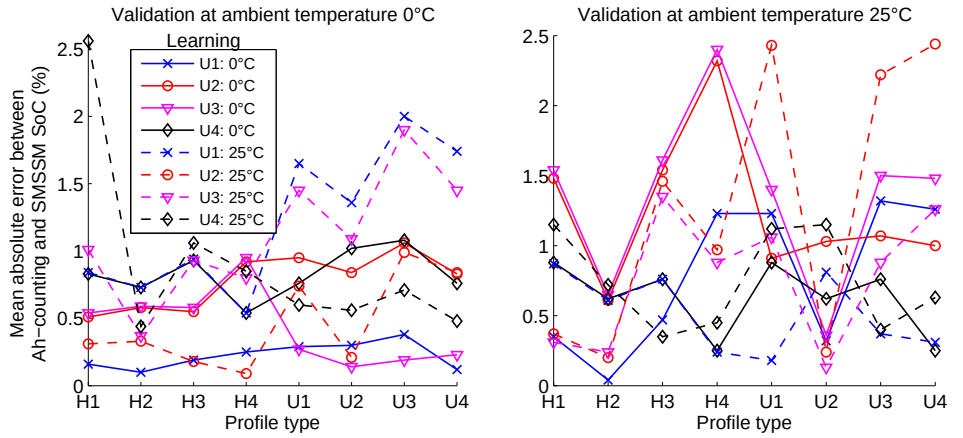

FIGURE 7.16: Mean absolute error between the Ah-counting and the SMSSM SoC: learning with the four simulated current profiles formed of the urban elementary profile at ambient temperature equal to $0°C$ and $25°C$, and validation using the eight simulated current profiles at ambient temperature equal to $0°C$ and $25°C$. The learned model is selected with BIC.

As for the highway elementary profile, it can be observed that the forecast capacity of models learned using partial discharge experiments (i.e., U2 at 0 and $25°C$) is less powerful than that of the other models (cf. Figure 7.19 -right-).

Also, these results highlight that the most powerful model in terms of forecast accuracy (mean absolute and maximum error) is the one learned using discharge experiment with U4 (cf. Figure 7.19 -right-).

In addition, validations occurring at $25°C$ show that the estimated SoC corresponding to partial discharge experiments with current profiles of type 2 (i.e., H2 and U2 at $25°C$) is more accurate than the other ones. This trend confirms that the behavior of the battery is well-learned for $SoC \in [50\%, 100\%]$.

7.1.3.5 Summary

To illustrate these validations, the Ah-counting SoC and the SMSSM SoC are shown for two cases:

1. learning using a discharge experiment with H4 at $25°C$ and validation using a discharge experiment with U1 at $0°C$ (cf. Figure 7.17);
2. learning using a discharge experiment with U2 at $25°C$ and validation using a discharge experiment with U3 at $25°C$ (cf. Figure 7.18). It can be observed that the estimation error of $SoC \in [0\%, 50\%]$ is large. This is du to the fact that the learning dataset is performed using a partial discharge ($SoC \in [50\%, 100\%]$).

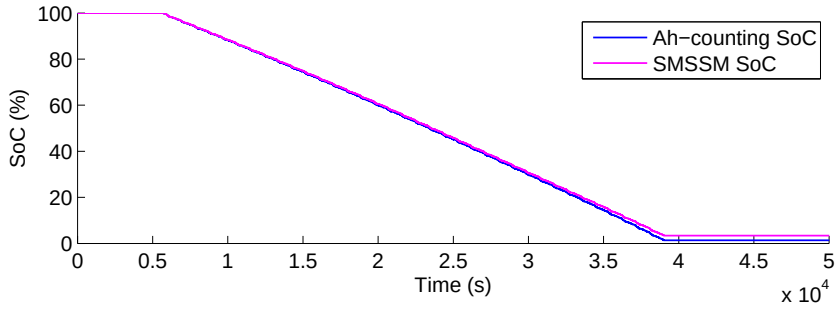

FIGURE 7.17: Learning using a discharge experiment with H4 at $25°C$ and validation using a discharge experiment with U1 at $0°C$

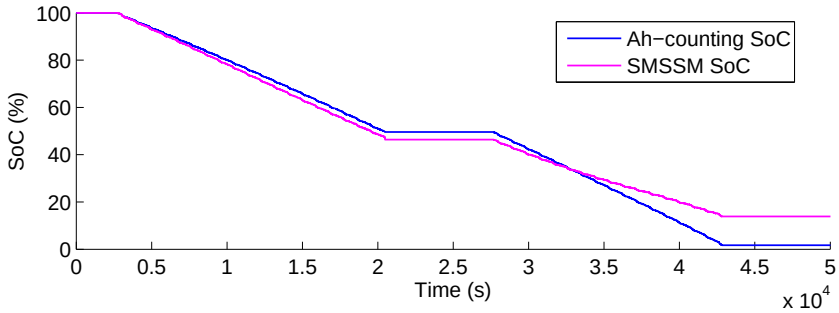

FIGURE 7.18: Learning using a discharge experiment with U2 at $25°C$ and validation using a discharge experiment with U3 at $25°C$

Considering the overall results, three key points emerge in terms of forecast capacity (cf. 7.19):

1. learning using discharge experiments with type 2 current profiles (i.e., partial discharge) is less performant than by using a complete discharge of the cell,
2. learning and validation at $0°C$ (resp. $25°C$) is more performant than learning at $25°C$ (resp. $0°C$) and validating at $0°C$ (resp. $25°C$),

3. learning using discharge experiments with current profiles of type 4 seems to be the most robust.

Tables C.1 and C.2 of Appendix C summarize the mean absolute, the standard deviation and the maximum estimating error of these 256 validations.

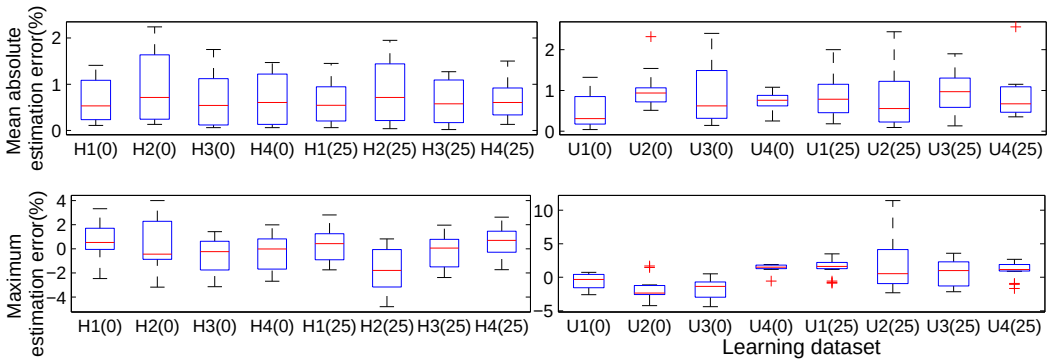

FIGURE 7.19: Boxplots of the mean absolute estimation (top) and maximum estimation error (bottom). The estimation error is given by SoC(Ah-counting)$-SoC$(SMSSM). The ambiant temperature of each experiment is written in parenthesis.

7.1.4 Physical interpretation of the parameters

In this section, we verify the physical interpretation of the parameters of the SoC SMSSM. Let us recall that the transition and observation equations are based on physical models, and those parameters correspond to physical variables (cf. Section 3.2). Figures 7.20 and 7.22 show the parameters of the model learned using H4 respectively at $25°C$ and $0°C$ and selected with BIC. It can be observed that the values of the estimated parameters are close to the physical variables:

- $B(s_t)$ close to $\frac{100}{C_{\text{nominal}}} = 45.45$ Ah^{-1}: the nominal capacity is equal to 2.2 Ah, and the estimated capacity corresponding to $\frac{100}{B(s)}$ varies between 2 and 2.3 (Figure 7.21);
- the variation of $C(s_t)$ (cf. Figure 7.20) corresponds to the variation of the slope of OCV-SoC curve (cf. Figure 7.3);
- $D(s_t)$ varies between 60 and 80 (Figure 7.20), it is close to the internal resistance R varying between 60 and 100 $m\Omega$ at $25°C$ (Figure 7.5): $D(s = 4)$ might correspond to the internal resistance of $SoC < 20\%$.

In addition, by comparing the values of D estimated at $25°C$ (Figure 7.20) and those estimated at $0°C$ (Figure 7.22), it is can be observed that D increases when the ambient temperature decreases which corresponds to the physical evolution of the internal resistance.

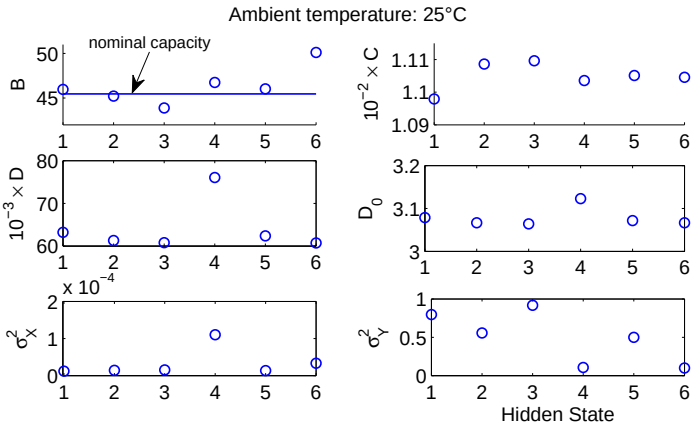

FIGURE 7.20: Estimated parameters of the model learned using H4 at $25°C$ and selected with BIC

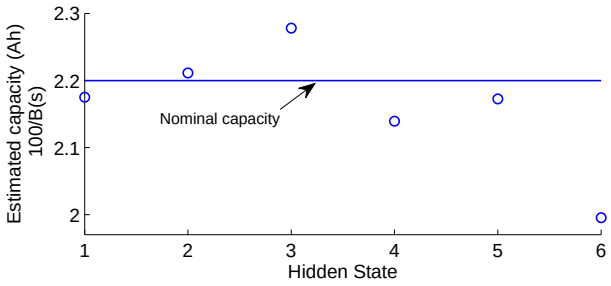

FIGURE 7.21: Estimated capacity corresponding to $\frac{100}{B(s)}$: the model is learned using H4 at $25°C$ and selected with BIC

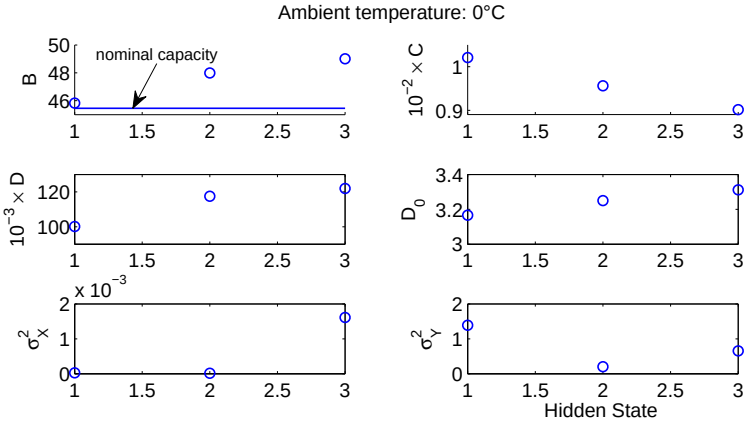

FIGURE 7.22: Estimated parameters of the model learned using H4 at $0°C$ and selected with BIC

Moreover, $C(s_t)X_t + D_0(s_t)$ corresponds to the voltage source V_{OC} (cf. Figure 3.1). Figure 7.23 shows that the estimated and laboratory V_{OC} with respect to the SoC are close for $SoC \in [20\%, 100\%]$, and otherwise the estimated V_{OC} is smaller than the laboratory one. The estimated V_{OC} is calculated for a discharge experiment using H1 at

$25°C$, whereas the laboratory V_{OC} is calculated during a 2C impulse discharge current at $25°C$ (cf. Figure 7.3).

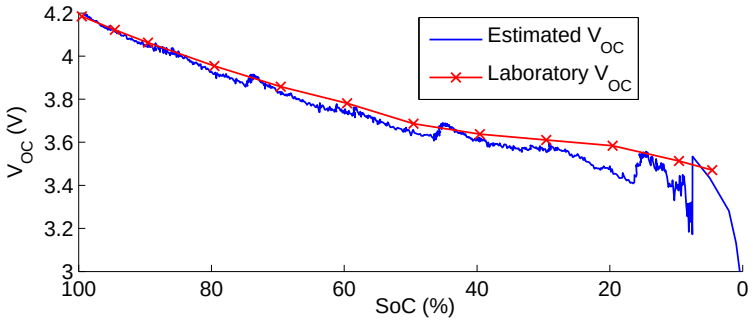

FIGURE 7.23: Estimated $V_{OC} = \sum_{i=1}^{N} \widehat{w}_t^i \left(C(s_t^i)\hat{x}_t^i + D_0(s_t^i) \right)$ calculated for a discharge experiment using H1 at $25°C$, and laboratory V_{OC} calculated during a 2C impulse discharge current at $25°C$

7.2 Module batteries of type L

7.2.1 Module performances

The experiments presented in this section have been performed by CEA/LITEN/DTS/S3E/LSEC in collaboration with Michelin.

7.2.1.1 Nominal specifications

In order to validate the *SoC* SMSSM for a module batteries, a module, called "M60", formed by connecting 60 lithium-ion rechargeable cells of type L is used. Its nominal specifications are listed in the following:

- Nominal capacity: $75\,Ah$,
- Nominal voltage: $38.4\,V$,
- Float voltage: $43.8\,V$,
- Cut-off voltage: $24.0\,V$,
- Nominal charge/discharge current rate: $C/2$,
- Nominal ambient temperature: $25°C$.

7.2.1.2 Complete discharge performance

The variability of the behavior of this module according to its internal characteristics and external usage conditions is characterized through several laboratory tests. As a first step, the module is completely discharged with constant current rates equal to $C/2$ and $1C$ at five different ambient temperatures, namely at -5, 5, 25 and $35°C$. A supplementary discharge experiments is performed at $15°C$ with $1C$ constant current rate. It can be observed that the actual capacity increases when the ambient temperature increases, and that this capacity with $C/2$ discharge current rate (i.e., the nominal discharge current rate) is at least greater that the one with $1C$ discharge current rate (cf. Figure 7.24). It is noteworthy that for all tests, the actual capacity is less than the nominal one. This is due to the dispersion of the capacity of the 60 cells which compose the module. The impact of this heterogeneity is detailed in the pack case.

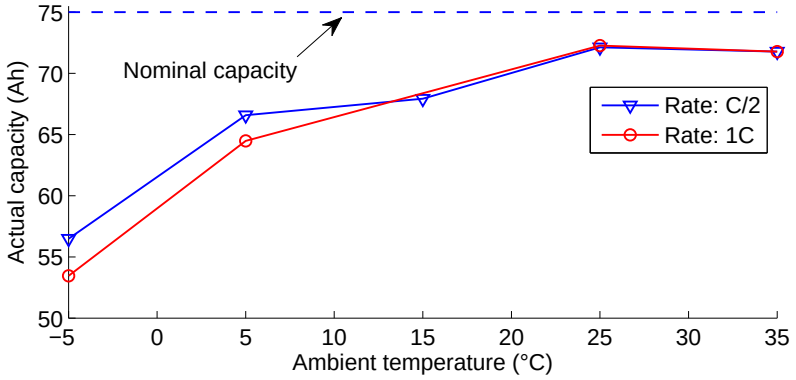

FIGURE 7.24: Maximum amount of charge that can be removed from the module of type L with discharge current rate equal to $C/2$ and $1C$ at different ambient temperatures (-5, 5, 15, 25, $35°C$)

7.2.1.3 Open circuit voltage

As a second step, "M60" is discharged with an impulse power of rate 1.83E and charged with an impulse power of rate 1.3E at $25°C$. After each impulse, the SoC is calculated and the cell is maintained at rest (i.e., $I = 0A$) in order to calculate its corresponding OCV and internal resistance. Figure 7.25 shows the OCV as a function of the SoC. These results show that the hysteresis effect between the charge and discharge is stronger. Nevertheless, the OCV is almost constant when $SoC \in [20\%, 80\%]$. From an electrical point of view, this is beneficial as the battery is able to deliver a constant power within this large SoC interval. For the SoC estimation, this hysteresis effect as well as this slight change of the OCV limit the performances of methods based on a measured and/or estimated OCV.

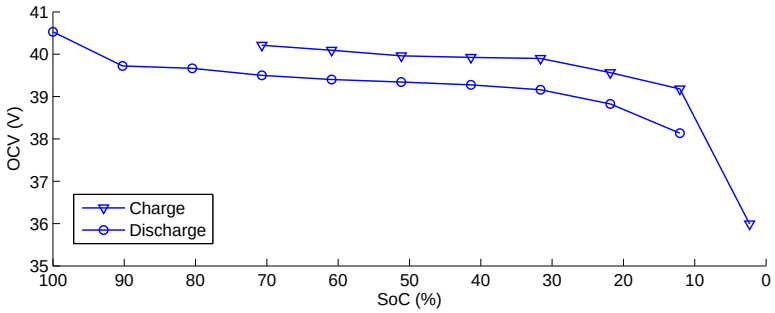

FIGURE 7.25: Hysteresis effect between charge ($\triangledown$) and discharge ($\circ$) of "M60": impulse power charge of rate 1.3E and impulse power discharge of rate 1.883E at ambient temperature equal to $25°C$

7.2.1.4 Internal resistance

The impact of the charge and discharge on the internal resistance is presented in Figure 7.26. It can be observed that for charge (resp. discharge), the internal resistance tends to increase (resp. decrease) when the SoC increases.

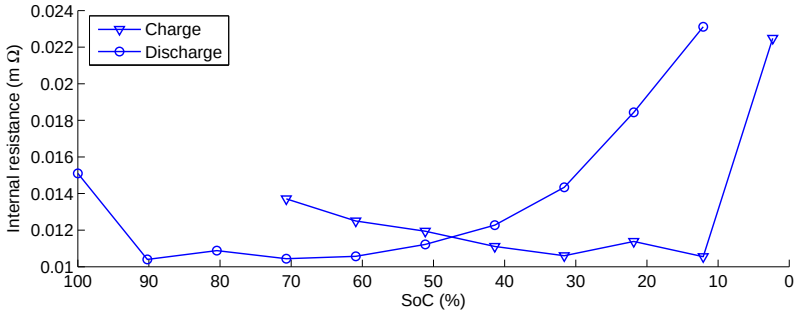

FIGURE 7.26: Internal resistance vs. SoC of the module of type L with impulse charge and discharge at ambient temperature equal to $25°C$

7.2.2 Validation protocol

The validation protocol aims at identifying the advantages and limitations of the proposed SoC model for a module batteries. For this purpose, two different elementary profiles are considered. The first one, called "NEDC" for New European Driving Cycle, represents a discharge power profile simulating a typical usage of a car in Europe. The second one, called "Michelin", is obtained through monitoring experiments of an electric vehicle within the Forewheel project in partnership with Michelin. This profile is adjusted to the "M60" specifications. The electrical characteristics of these two profiles are summarized in Table 7.4. Accordingly, a complete discharge of "M60" with the Michelin profile (3.2h) is faster than the NEDC profile (4.2h) as the Michelin mean power is greater than the NEDC one. Moreover, the maximum Michelin current peaks are more frequent than the NEDC ones (cf. Figure 7.27): this maximum is equal to $-180A$ for

the two profiles. Figure 7.28 shows the current and the voltage measurements during a complete discharge of the "M60" with the Michelin and NEDC profiles.

	NEDC	Michelin
Mean power (kW)	−6.4	−10.2
Mean recovered power (kW)	4.2	7.2
Mean discharge power (kW)	−7.7	−14.1
Profile duration (min)	20.3	91.3
Duration of a complete discharge (h)	4.2	3.2
Number of profiles to complete discharge	12.5	2.1

TABLE 7.4: Electrical characteristics of the NEDC and Michelin profiles in pack scale

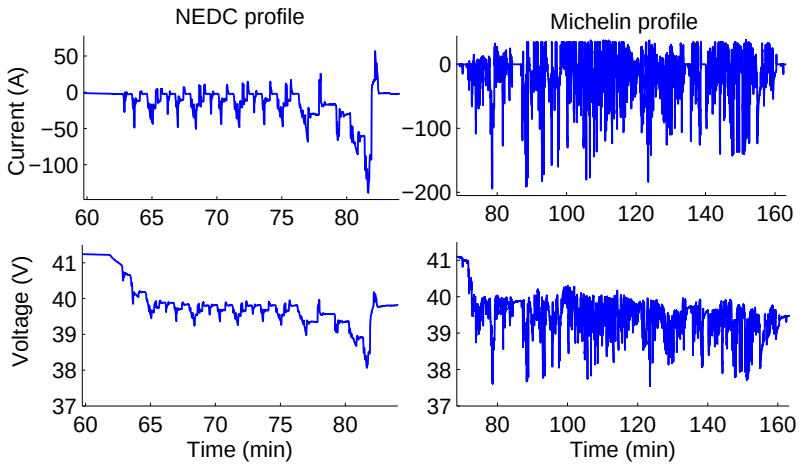

FIGURE 7.27: NEDC and Michelin profiles: current and voltage measurements

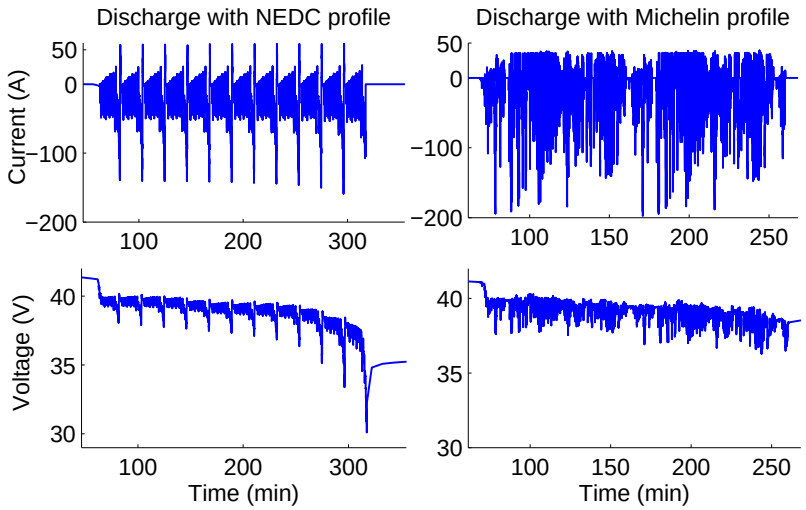

FIGURE 7.28: Complete discharge with the NEDC and the Michelin profiles: current and voltage measurements

Twelve laboratory experiments are performed using these two profiles:

- complete discharge of "M60" with each profile at -5, 5 and $25°C$;
- partial discharge of "M60" by removing 30% of its maximum amount of charge with each profile at -5, 5 and $25°C$.

For all experiments, the module is initially charged under its nominal charge regime (i.e., ambient temperature: $25°C$, and current rate: $C/2$). Then, after the end of each experiment, the module is completely discharged also under its nominal discharge regime (i.e., ambient temperature: $25°C$, and current rate: $C/2$).

7.2.3 Application of the SoC SMSSM using module data

7.2.3.1 Learning datasets

The data (i.e., voltage and current) collected, with sampling time $T_s = 1s$, during each of these 12 experiments is considered as a learning dataset. The size of these datasets are reported in Table 7.5.

	Complete NEDC	Complete Michelin	Patrial NEDC	Patrial Michelin
$-5°C$	11698	6511	4258	3547
$+5°C$	11628	6379	5243	4021
$25°C$	13946	7327	5721	4579

TABLE 7.5: Size of the dataset collected during the 12 discharge experiments of "M60"

7.2.3.2 Estimation of the unknown parameters

For each of these learning datasets, the unknown parameters of SoC models with $\kappa = 2, \ldots, 10$ are estimated using the MCEM algorithm penalized with identifiability constraints. Indeed, for these experiments, the initial value of the SoC and its corresponding voltage are known (i.e., $SoC_0 = 100\%$ and $y_0 = 43.8\,V$) since the module is initially fully-charged under its nominal regime. Accordingly, the following relationship is available at $t_0 = 0$

$$43.8 = 100 \times C(s) + D_0(s), \tag{7.2}$$

where $s = 1, \ldots, \kappa$.

In addition, to avoid a local Maxima of Likelihood, for a given learning dataset and number of Markov states κ, this algorithm is repeated 20 times; then the estimated model having the maximum likelihood is adopted.

Moreover, as shown in Section 5.5.1.2 the estimated ML is affected by the number N of simulated particles. For these experiments we use $N = 200$ for $\kappa = 2, 3, 4$; $N = 500$

for $\kappa = 5, 6, 7$; and $N = 750$ for $\kappa = 8, 9, 10$. This choice is determined based on the dispersion of the estimated ML of the largest learning dataset (i.e., $T \simeq 10^4$) amongst the 12 conducted experiments. Indeed, for each κ, tests similar to those described in Section 5.5.1.2 are performed to choose the appropriate number of particles.

7.2.3.3 Estimation of the number of hidden discrete states

As a result of the above, and for each learning dataset, there are 9 candidate SMSSMs ($\kappa = 2, \ldots, 10$). The criteria BIC, AIC and SHC are then used to select an adequate model among the nine models. Table 7.6 summarizes the selected number of hidden states for the 12 learning datasets.

It is noteworthy that for the SHC, the corresponding selected κ is not reported if a linear behavior of the ML is not observed. It can be observed that on almost all cases, these three criteria lead to the selection of different κ. The number of hidden discrete states selected with BIC is relatively stable ($k = 3, 4, 5$). However, it is complicated to establish a relation between the number of hidden states selected with AIC and SHC and the ambient temperature or the type of the current profile alike.

Nevertheless, it is surprising that in some cases the number of hidden discrete states selected with AIC for partial discharge is larger than the one for complete discharge (NEDC at $-5°C$, and Michelin at -5 and $+5°C$). This behavior confirms that AIC is not appropriate for mixture models (McLachlan and Peel [2000]). Indeed, Figure 7.29 shows that the SMSSM with $\kappa = 4$ is a good candidate, but AIC selects the SMSSM with $\kappa = 10$ as its complexity is not sufficiently penalized.

	Complete discharge			Partial discharge		
NEDC	$-5°C$	$+5°C$	$25°C$	$-5°C$	$+5°C$	$25°C$
BIC	4	5	4	3	4	3
AIC	5	8	9	10	5	7
SHC	6	5	4	3	4	4
Michelin	$-5°C$	$+5°C$	$25°C$	$-5°C$	$+5°C$	$25°C$
BIC	4	5	3	3	3	4
AIC	4	5	7	7	8	7
SHC	5	5	7	-	-	4

TABLE 7.6: Selected number of hidden states for the *SoC* SMSSMs of "M60" with AIC, BIC and SHC. There are 12 models learned using the 12 datasets collected during the 12 discharge experiments of the module.

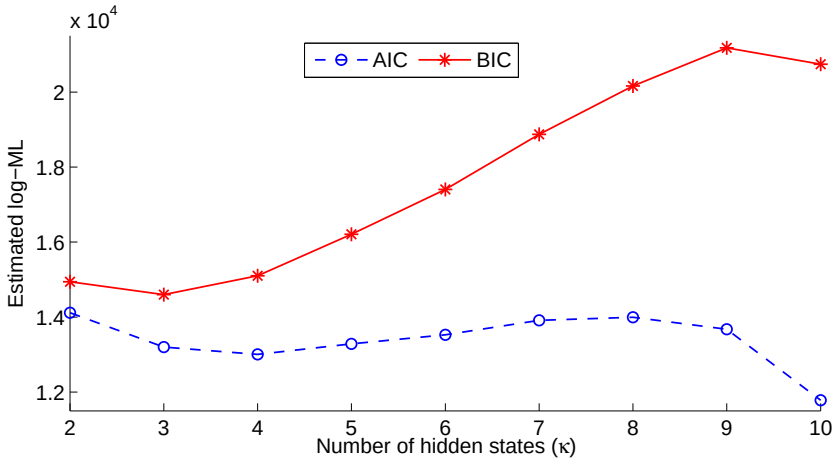

FIGURE 7.29: AIC and BIC with $\kappa = 2, \ldots, 10$: the models are learned using a patrial NEDC discharge at $-5°C$

7.2.3.4 Validation phase

A learned *SoC* model is finally validated for each of the 12 learning datasets. Accordingly, given a model selection criterion, there are 12×12 (i.e., 144) possible combinations. Due to the high number of possible combinations, validation work is limited to the model selected with BIC.

For these laboratory experiments, the Ah-counting *SoC* is considered as the reference *SoC*. This is justified by the facts that the initial value of the *SoC* and the actual capacity are known, and that the current sensor is accurate. Indeed, the actual capacity is calculated offline at the end of the experiment by integrating the current that flows through the battery.

Learning with complete discharges - Figures 7.30 and 7.31 show the results of the validation of the models learned using the 6 complete discharge experiments with NEDC and Michelin at -5, 5 and $25°C$. Figure 7.30 presents the maximum of the estimation error between the Ah-counting *SoC* and the *SoC* estimated using these learned SMSSMs; whereas Figure 7.31 presents the mean absolute error.

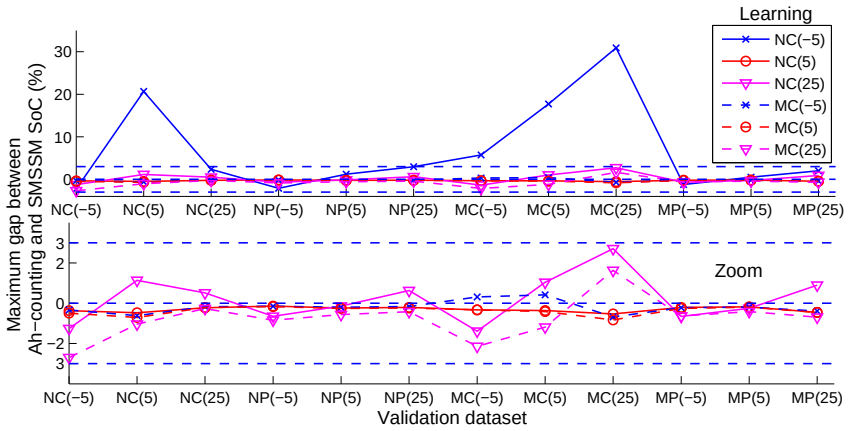

FIGURE 7.30: Maximum error between the Ah-counting and the SMSSM SoC: learning with complete discharge experiments with NEDC and Michelin at ambient temperature equal to $-5°C$, $+5°C$ and $25°C$, and validation based on the 12 conducted experiments. NC and NP (resp. MC and MP) denote complete discharge with NEDC (resp. Michelin) and partial discharge with NEDC (resp. Michelin). The ambient temperature of each experiment is written in parenthesis. The learned model is selected with BIC.

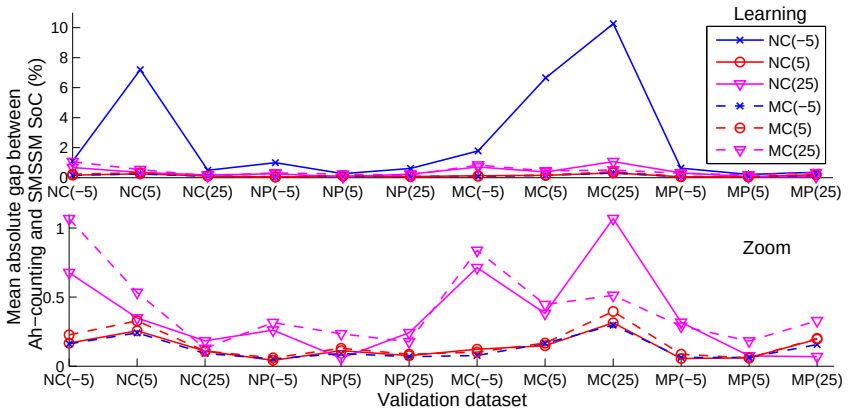

FIGURE 7.31: Mean absolute error between the Ah-counting and the SMSSM SoC: learning with complete discharge experiments with NEDC and Michelin at ambient temperature equal to $-5°C$, $+5°C$ and $25°C$, and validation based on the 12 conducted experiments. NC and NP (resp. MC and MP) denote complete discharge with NEDC (resp. Michelin) and partial discharge with NEDC (resp. Michelin). The ambient temperature of each experiment is written in parenthesis. The learned model is selected with BIC.

It can be observed that the prediction capacity of the model learned with complete NEDC discharge at $-5°C$ is less powerful than that of the other models. Also, these results highlight that the models learned at $5°C$ are the most powerful in terms of forecast capacity (cf. Figure 7.35).

In addition, these results emphasize that the behavior of the battery for $SoC \in [70\%; 100\%]$ is well-estimated in all cases. Indeed, the estimated SoC corresponding to partial discharge experiments is more accurate than that in the complete discharge experiments.

Learning with partial discharges - Figures 7.32 and 7.33 show the results of the validation of the models learned using the 6 partial discharge experiments with NEDC and Michelin at -5, 5 and $25°C$. Figure 7.32 presents the maximum of the estimation error between the Ah-counting SoC and the SoC estimated using these learned SMSSMs; whereas Figure 7.33 presents the mean absolute error.

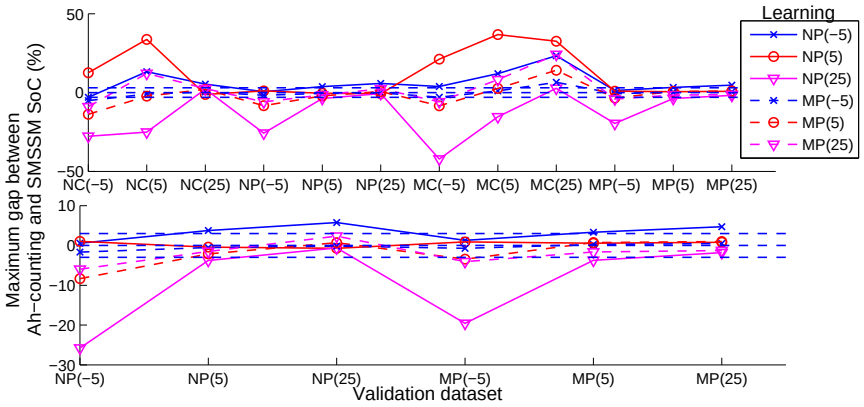

FIGURE 7.32: Maximum error between the Ah-counting and the SMSSM SoC: learning with partial discharge experiments with NEDC and Michelin at ambient temperature equal to $-5°C$, $+5°C$ and $25°C$, and validation based on the 12 conducted experiments. NC and NP (resp. MC and MP) denote complete discharge with NEDC (resp. Michelin) and partial discharge with NEDC (resp. Michelin). The ambient temperature of each experiment is written in parenthesis. The learned model is selected with BIC.

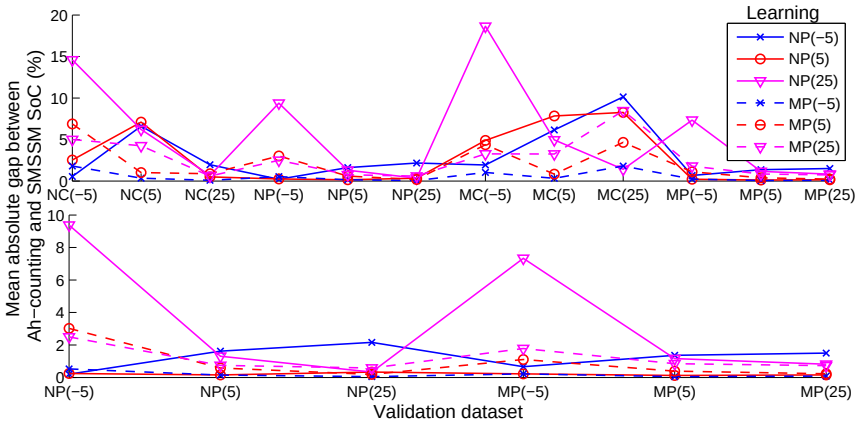

FIGURE 7.33: Mean error between the Ah-counting and the SMSSM SoC: learning with partial discharge experiments with NEDC and Michelin at ambient temperature equal to $-5°C$, $+5°C$ and $25°C$, and validation based on the 12 conducted experiments. NC and NP (resp. MC and MP) denote complete discharge with NEDC (resp. Michelin) and partial discharge with NEDC (resp. Michelin). The ambient temperature of each experiment is written in parenthesis. The learned model is selected with BIC.

It can be observed that the prediction capacity of these models learned with partial discharge is less powerful than that of the models learned with complete discharge (cf. Figure 7.34). This was expected as for the partial discharge experiments only 30% of the maximum amount of charge is removed from the cell. Indeed, the behavior of the

battery for $SoC \in [0\%, 70\%]$ is not integrated in the learning dataset. However, the model learned with partial discharge with Michelin profile at $-5°C$ seems to have best performance prediction than the other models learned with partial discharge.

7.2.3.5 Summary

Considering the overall results, three key points emerge in terms of forecast capacity (cf. Figures 7.34 and 7.35):

1. learning using partial discharge experiments is less performant than by using a complete discharge of the module,
2. the impact of the ambient temperature on the performances of the learned model is limited in the "M60" case,
3. learning using complete discharge experiments at $+5°C$ seems to be the most robust.

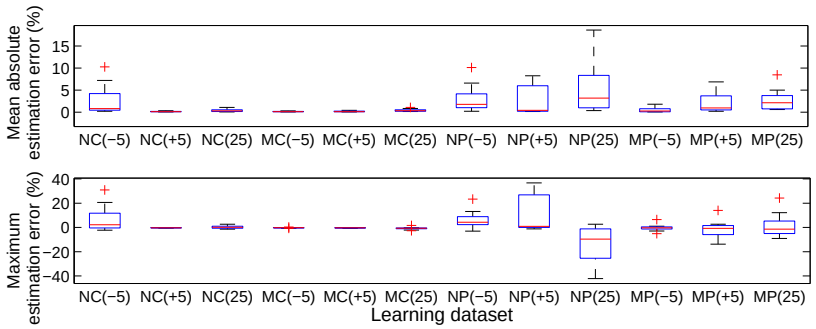

FIGURE 7.34: Boxplots of the mean absolute estimation (top) and maximum estimation error (bottom). The estimation error is given by SoC(Ah-counting)$-SoC$(SMSSM). NC and NP (resp. MC and MP) denote complete discharge with NEDC (resp. Michelin) and partial discharge with NEDC (resp. Michelin). The ambiant temperature of each experiment is written in parenthesis.

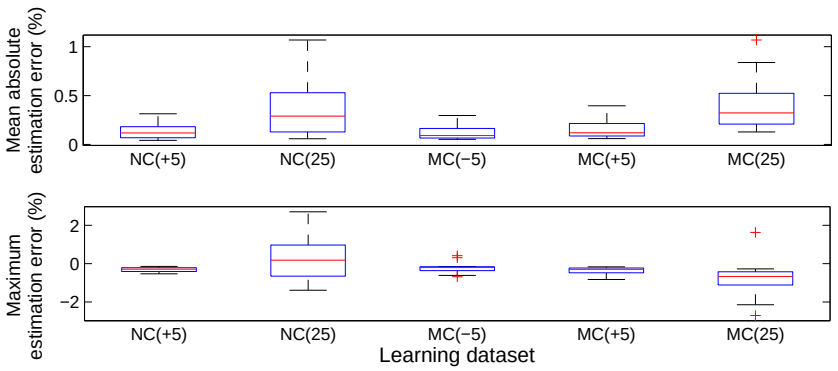

FIGURE 7.35: Boxplots of the mean absolute estimation (top) and maximum estimation error (bottom): focus on the complete discharge datasets. The estimation error is given by SoC(Ah-counting)$-SoC$(SMSSM). NC and NP (resp. MC and MP) denote complete discharge with NEDC (resp. Michelin) and partial discharge with NEDC (resp. Michelin). The ambiant temperature of each experiment is written in parenthesis.

Table C.3 of Appendix C summarizes the mean absolute, the standard deviation and the maximum estimating error of these 144 validations.

7.3 Pack batteries of type L

7.3.1 Pack performances

The experiments presented in this section have been performed by CEA/LITEN/DTS/S3E/LSEC in collaboration with Michelin.

7.3.1.1 Nominal specifications

In order to validate the SoC model for a pack batteries, we consider an electric vehicle, called "WILL", equipped with a pack batteries formed by connecting 10 modules "M60". This pack is thus made up of 120 stages connected in series. Its nominal specifications are listed in the following:

- Nominal capacity: $75\,Ah$,
- Nominal voltage: $384\,V$,
- Float voltage: $438\,V$,
- Cut-off voltage: $240\,V$,
- Nominal charge/discharge current rate: $C/2$,
- Nominal ambient temperature: $25°C$.

7.3.1.2 Characterization procedure

To evaluate the performance of the pack under a real usage, experiments have been performed during 23 months including both charge/discharge and storage periods. The overall process of these experiments consists of three operating modes (cf. Figure 7.36):

- Drive 1, 2, 3, 4 and 5: driving the vehicle on a private circuit to control the road conditions. A professional driver repeats a specific speed profile comprising all phases of a real-life drive: start, stop, acceleration, deceleration, constant speed, etc. There are 5 different periods of drive;
- Storage: long period of storage without charging/discharging;
- Test 1 and 2: laboratory charge/discharge tests using a profile reproducing the same power profile of the real drive in mode 1.

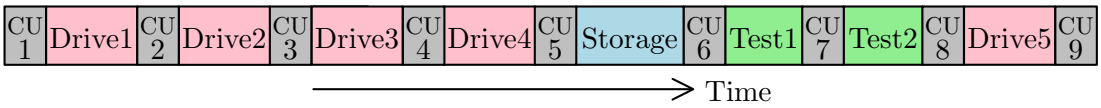

FIGURE 7.36: Illustration of the global experiments progress over time. CUx denote the x-th checkup.

During these 23 months, 9 checkups have been regularly conducted to achieve a complete characterization of the pack and monitor the evolution of its electric performances, namely its state of health SoH, its actual capacity, its internal resistance, etc. All of these checkups follow the same test protocol and have been performed at $25°C$. The pack is maintained in rest for one hour, then it is charged with C/9 constant current. The fully-charged pack is then discharged with three different current profils: C/2 constant current and two dynamics current profiles, namely NEDC and ARTEMIS Urbain (André [2004]).

7.3.1.3 Experimental results

These checkups show that the SoH of the pack decreases from 100% at the first checkup to 87% at the last one (cf. Figure 7.37). Indeed, the evolution of the SoH depends primarily on the usage conditions (Barré et al. [2013]).

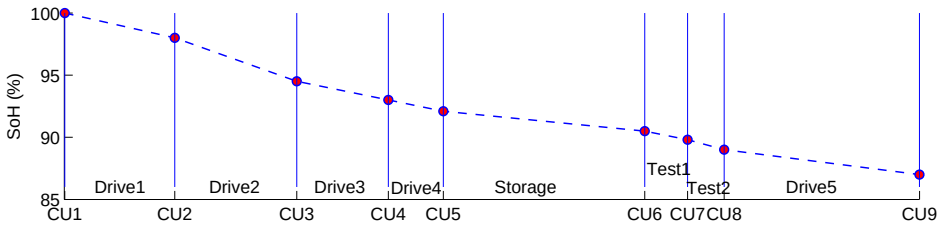

FIGURE 7.37: Evolution of the state of health of the pack during the 23 month

In addition, they emphasize the heterogeneity of the electric performances of the 120 stages. Indeed, during a discharge experiment it can be observed that the voltage of certain stages reaches the cut-off one, whereas the voltage of another stages is still larger than the cut-off one (cf. Figure 7.38) due to the dispersion of the cell performances (SoC, capacity, resistance). An offline computation of the actual capacity of each branch highlights the electric heterogeneity of the stages. Figure 7.39 shows the boxplots of these actual capacities for the 9 checkups. It can be observed that the maximum gap between these capacities increases when the SoH decreases. Based on these actual capacities, the SoC of each branch can be calculated (cf. Figure 7.40). The issue that arises from this heterogeneity is how to define the SoC of the pack. It is clear that for security reasons, the discharge should be stopped when the voltage of one of the 120 stages reaches the cut-off one. It seems obvious that this cannot be taken into account by a single Ah-counting method. Instead, a combination of Ah-counting SoCs of the 120

stages or of the extremum values of the stage SoCs (maximum for charge and minimum for discharge) should be considered.

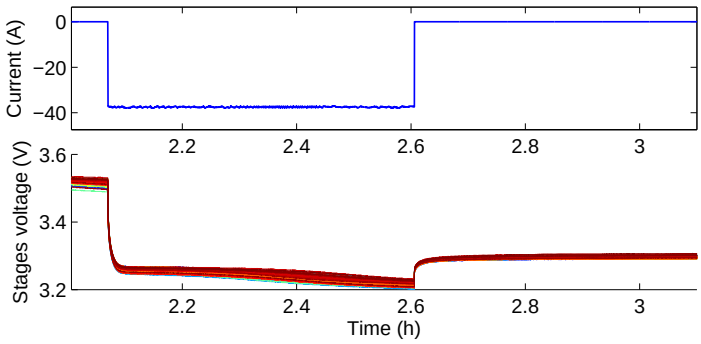

FIGURE 7.38: Complete discharge of the pack using constant current: current measurement (top) and voltage measurements of the 120 stages (bottom)

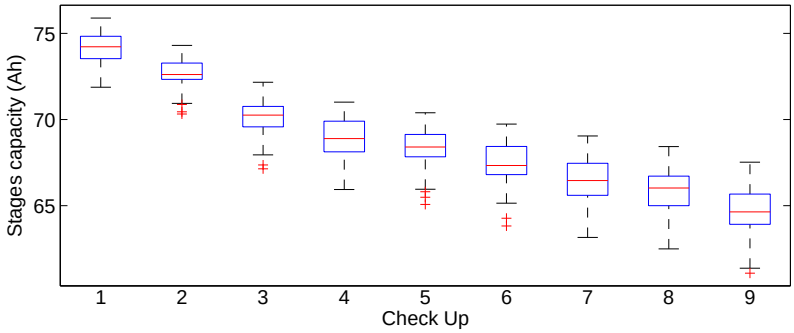

FIGURE 7.39: Boxplots of the actual capacity of the 120 stages of the pack calculated during the 9 checkup periods

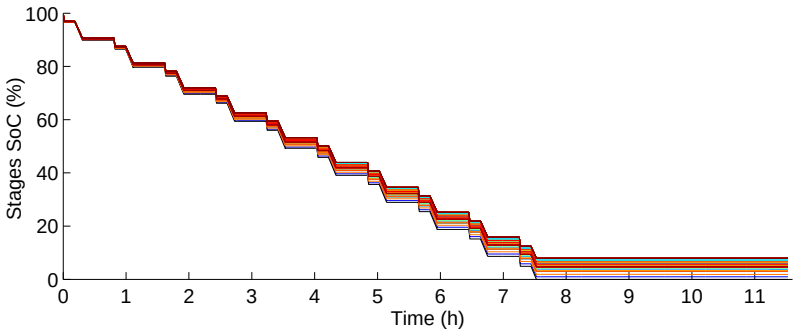

FIGURE 7.40: The SoC of the 120 stages of the pack during a discharge experiment. The SoC of each branch is calculated offline using its actual capacity.

7.3.2 Application of the SoC SMSSM using pack data

7.3.2.1 Learning datasets

Due to the big size of experiments conducted during the overall process (i.e., 27 Go), we choose to consider, for each checkup, the data (i.e., current and voltage) collected during the complete discharge experiment with NEDC profile at $25°C$ as a learning dataset. The key motivation for this choice is that the current and voltage sensors are accurate, and that this might reflect a real-life situation for which the SoC model is learned during the regular checkups of the vehicle.

7.3.2.2 Estimation of the unknown parameters

There are thus 9 learning datasets corresponding to different states of health of the pack and heterogeneities of its stages. For each learning dataset, 9 SoC SMSSMs with $\kappa = 2, \dots 10$ are learned using the MCEM algorithm penalized with identifiability constraints. Indeed, the initial value of SoC and its corresponding voltage are known (i.e., $SoC_0 = 100\%$, $y_0 = 438\,V$) since the pack is fully-charged with C/9 constant current before the discharge experiment. The following constraint is thus available at $t_0 = 0$:

$$438 = 100 \times C(s) + D_0(s), \tag{7.3}$$

where $s = 1, \dots, \kappa$.

As in the cell and module cases, the MCEM algorithm is repeated 20 times to avoid local maxima likelihood, then the estimated model having the maximum likelihood is adopted. Moreover, the choice of the number of particles N is identical to those in the cell and module cases since the size of the learning datasets is of the same order.

7.3.2.3 Estimation of the number of hidden discrete states

Accordingly, for each learning dataset (i.e., checkup), there are 9 candidate SMSSMs ($\kappa = 2, \dots, 10$). The criteria BIC, AIC and SHC are also used here to select an adequate model among the 9 ones. Table 7.7 summarizes the selected number of hidden states for the 9 learning datasets. For CU2, CU3, CU5 and CU9, the three criteria select the same model order. For CU1, AIC select the most complex model (i.e., $\kappa = 10$). For CU7 and CU8, SHC select the most simple model (i.e., $\kappa = 2$). It can observed that in almost all cases the criteria select $\kappa = 5$ or 6, but it is complicated to establish a relation between the selected number of hidden states and the checkup.

	CU1	CU2	CU3	CU4	CU5	CU6	CU7	CU8	CU9
BIC	5	6	6	5	5	6	3	5	4
AIC	10	6	6	6	5	8	5	5	4
SHC	5	6	6	6	5	5	2	2	4

TABLE 7.7: Selected number of hidden states for the SoC SMSSMs of the pack batteries with BIC, AIC and SHC. There are 9 models learned using the 9 datasets collected during the 9 checkups. CUx denote the x-th checkup.

7.3.2.4 Validation phase

The validation of these learned models is performed based on data collected during the 5 periods of drive. For the 5-th period, we consider two drives in the morning and in the evening to test the impact of the internal temperature of the pack on the performance of the estimated SMSSM. Indeed, the professional driver repeats the speed profile several times in the day. Its internal temperature in the evening is thus larger than that in the morning.

Accordingly, given a model selection criterion, there are 54 (i.e., 9 models learned during the 9 checkups $\times$ 6 drives) possible combinations. Due to high number of possible combinations, validation work is limited to models selected with BIC.

It is noteworthy that for the periods of drive, no reference SoC is available as the current sensor is not very accurate and the actual capacity of the pack is unknown since the pack is partially discharged. However, for the validation, the estimated SoC using SMSSMs is compared to the Ah-counting SoC. Accordingly, the difference between these two estimated SoC is not an estimation error but a simple gap to be analyzed.

Also, it is important to note that Drive 2 is conducted during the summer (i.e., ambient temperature larger than $25°C$), the ambient temperature during Drive 5 in the morning is around $0°C$, and the ambient temperature of the other drives is around $15°C$.

Figures 7.41 and 7.42 show the results of the validation of the models learned during the 9 checkups. Figure 7.41 presents the maximum gap between the Ah-counting SoC and the SoC estimated using these learned SMSSMs, whereas Figure 7.42 presents the mean absolute gap.

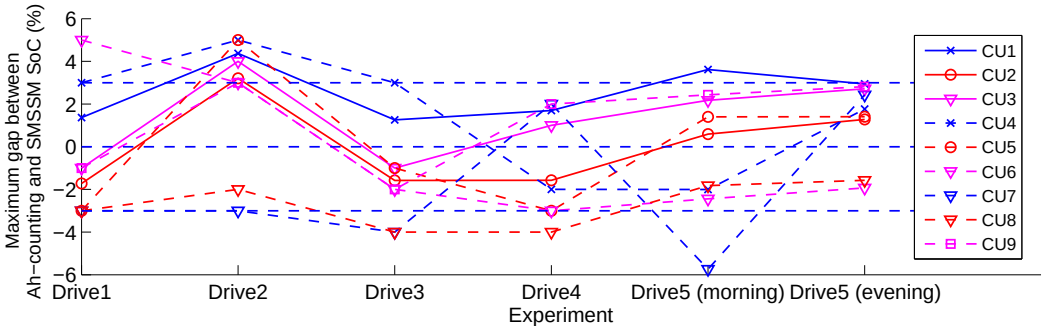

FIGURE 7.41: Maximum gap between the Ah-counting and the SMSSM SoC: learning during the 9 checkups at ambient temperature equal to $25°C$, and validation with the 6 drives. The learned model is selected with BIC.

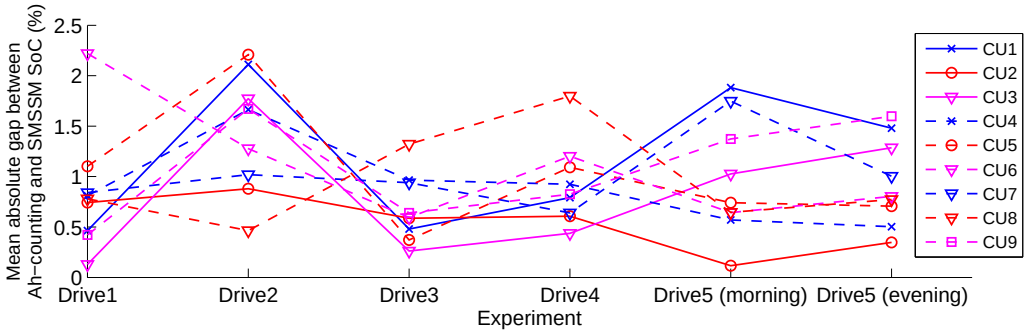

FIGURE 7.42: Mean absolute gap between between the Ah-counting and the SMSSM SoC: learning during the 9 checkups at ambient temperature equal to $25°C$, and validation with the 6 drives. The learned model is selected with BIC.

It can be observed that the impact of the state of health on the performance of the model is limited. The gap corresponding to Drive 2 and Drive 5 in the morning is larger than that of the other drives. It can be due to the fact that the ambient temperature of these two drives ($30°C$ for Drive 2 and $0°C$ for Drive 5 in the morning) is very different from that of the other drives (i.e., $15°C$). Considering the overall results, the learning during the CU2 seems to be the most robust. In addition, these results emphasize that the SMSSM can take into account the dispersion of the capacity of the 120 stages, and thus there no need to define a SoC per stage.

7.3.3 Interpretation of the discrete hidden states

Figure 7.43 shows the variation over time of the estimated hidden Markov state, the current, the voltage, the speed and the acceleration. The presented s_t corresponds to the one that appears the most in the N particles s_t^i. It can be observed that the hidden Markov states might reflect a specific usage of the pack. However, the interpretation of these states is not obvious: in this case, the States 2 and 6 might correspond to a constant speed, whereas the other states correspond to accelerations and decelerations.

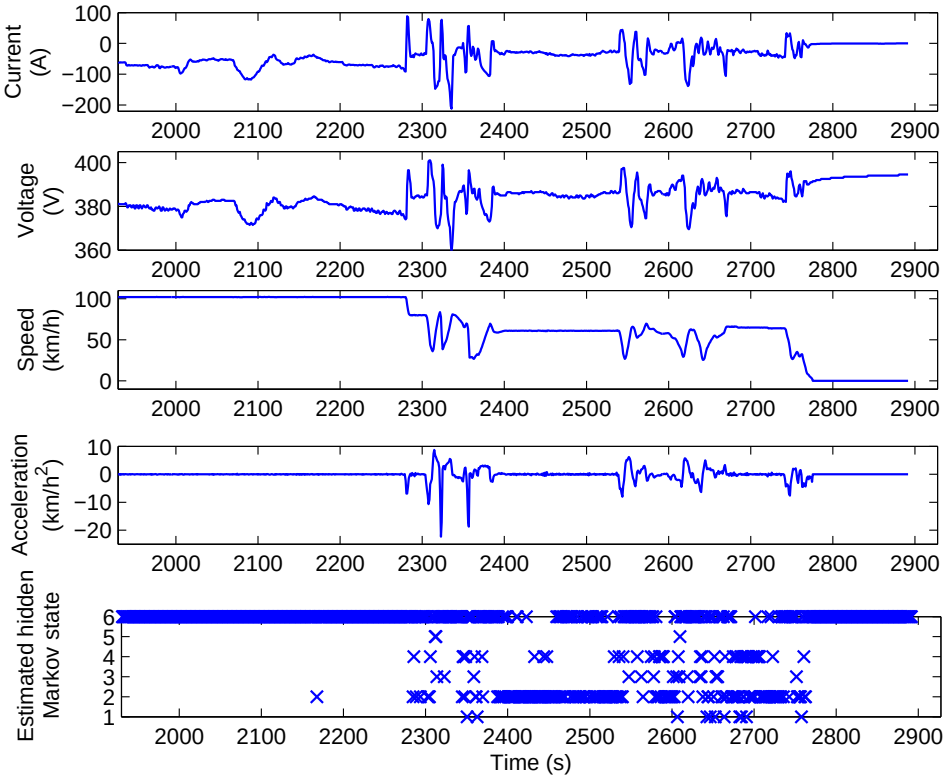

FIGURE 7.43: Variation over time of the current, voltage, speed, acceleration and estimated hidden state. The presented s_t corresponds to the one that appears the most in the N particules s_t^i. The data are collected during Drive 5 in the evening.

7.4 Discussion

In this chapter, we have applied and validated the proposed SoC model using real-life battery data. Three types of electric batteries have been considered: cell, module and pack. For each type, the impact of the usage conditions on the actual capacity, the OCV and the internal resistance is shown. In the module and pack cases, it is noteworthy that the heterogeneity of the electric characteristics of their elementary cells makes the estimation of their SoC more complicated.

In the cell case, we have considered height types of current profiles simulating different driving situations. The data (i.e., current and voltage) collected during discharge experiments with these height profiles at $0°C$ and $25°C$ are used to learn the SMSSMs with $\kappa = 2\ldots,10$. In this chapter, the unknown parameters are estimated using the MCEM algorithm penalized with identifiability constraints. Indeed, for all experiments the battery is initially fully charged, x_0 is thus known (i.e., $SoC_0 = 100\%$). For these 16 discharge experiments, a cross-validation is then conducted using the models selected with BIC.

In the module case, two power profiles have been considered. The data collected during complete and partial discharge experiments at $-5°C$, $+5°C$ and $25°C$ are used to learn the SMSSMs with $\kappa = 2 \ldots, 10$. For the same reasons as above, the unknown parameters are estimated using the MCEM algorithm penalized with identifiability constraints. A cross-validation has been also conducted to identify the advantages and limitations of the learned models.

Finally, we have considered an electric vehicle, called "WILL", to validate the proposed model in the pack case. Experiments comprising real drive, storage, laboratory tests and regular checkups have been carried out to evaluate the performances of the pack. These experiments highlights the electric heterogeneity of the pack stages (i.e., different actual capacities), making it more difficult to estimate the SoC. The data collected during the checkups are used as learning datasets. The unknown parameters of the SMSSMs with $\kappa = 2 \ldots, 10$ have been estimated as in the cell and module cases. The validation of the models selected with BIC has been performed using the data collected during the drives of WILL.

In almost all cases, it is complicated to establish a relation between the selected number of hidden Markov states and the usage conditions. However, the model selected with the criterion BIC often makes more sense. In addition, the size of the learning datasets is large, thus the choice of BIC is pertinent. Moreover, the criterion AIC tends to overestimates the number of hidden Markov states which confirms its limitation in a mixture models context (McLachlan and Peel [2000]). The use of the criterion SHC is limited to when a linear behavior of the ML is observed.

In addition, we have verified the physical interpretation of the estimated values of the unknown parameters, since the transition and observation equations are based on physical models. In the cell case, it can be observed that these estimated parameters are close to their corresponding physical variables. The values of $D(s)$ corresponds perfectly to the internal resistance of the battery. The physical dependence between this resistance and the ambient temperature is even visible: the estimated values of $D(s)$ using a learning datatset collected during a discharge at $0°C$ is larger than that estimated using a learning dataset collected during a discharge at $25°C$. The variations of the parameter $C(s)$ corresponds also to the slope of the OCV-SoC curve.

The validation using these three types of batteries highlights the potential benefits and the practical usefulness of the switching Markov state-space models to estimate the SoC. Unlike existing methods, the proposed SoC model does not require a complex modeling of the physical and electrochemical battery phenomena. In addition, the parameters of this model are automatically estimated based on a learning dataset that maintains their cross-dependence. Indeed, the parameters of the existing methods are physically identified based on laboratory tests. The limitations of such an approach are detailed in Chapter 2.

Nevertheless, the prediction capacity of the proposed SoC model depends on the choice of the leaning dataset. Indeed, a model learned based on a complete discharge experiment is more performant than that learned based on a partial discharge experiment. In addition, it seems that the type of the current profile affects also the prediction capacity. Indeed, the models learned with highway profile of type 4 in the cell case and Michelin in the module case are more performant than the other models. Moreover, the results in the pack case show that the degradation of the state of health of the battery has a limited impact on the prediction capacity of the proposed model. This can not be generalized as the SoH interval tested in this study is limited to $[87\%, 100\%]$.

Accordingly, the choice of the best learning dataset still an open question. When a set of datasets is available, a cross-validation, such as that performed in this chapter, helps to identify the best learned model. However, the required time to carry out this procedure is time consuming.

Chapitre 8

Conclusions et Perspectives

Les batteries électriques jouent un rôle primordial dans la transition énergétique et elles sont omniprésentes dans notre vie quotidienne. Cependant, l'estimation de leur état de charge reste une problématique majeure, surtout lorsque les conditions d'usage de la batterie ne sont pas contrôlables comme dans le cas des véhicules électriques. Dans des telles applications, un calcul précis de l'état de charge est crucial pour fournir une indication de l'autonomie et gérer efficacement l'énergie.

Dans cette étude, nous avons développé un estimateur statistique de l'état de charge basé sur des mesures instantanées de courant et de tension. Cet estimateur tient compte de la variabilité du comportement de la batterie selon ses conditions d'usage et ses caractéristiques internes et il est également adapté aux contraintes d'un calculateur embarqué.

8.1 Bilan des travaux

Nouvelle classification des méthodes de SoC - Nous avons classifié les méthodes de *SoC* existantes selon trois caractéristiques : le type de variable d'entrée (non mesurable en temps réel, mesurable en temps réel et estimée), le type de modèle de *SoC* (table de correspondance, physique, électrochimique et mathématique) et la procédure d'estimation du *SoC* (non récursive, récursive à boucle ouverte et récursive à boucle fermée). Cette nouvelle classification nous a permis d'identifier des pistes prometteuses à explorer dans le domaine de l'estimation du *SoC* :

- la prise en compte de la variabilité du comportement de la batterie lors de la conception du modèle de *SoC*. En général, cette variabilité est traitée d'une façon heuristique. Par exemple, plusieurs jeux de paramètres du modèle de *SoC* sont estimés pour plusieurs températures ambiantes.

- l'utilisation de techniques avancées d'apprentissage statistique pour estimer les paramètres du modèle de *SoC*. En général, la méthode des moindres carrés est la plus utilisée dans ce cadre.

Conception d'un modèle de SoC tenant compte de la variabilité du comportement de la batterie - Nous avons mis en place une modélisation statistique du *SoC* à partir des mesures instantanées de courant et de tension basée sur les modèles à espaces d'états gouvernés par une chaîne de Markov cachée (SMSSM). Les équations de transition et d'observation correspondent à des modèles physiques : l'équation de transition est basée sur le modèle coulométrique et celle d'observation sur la modélisation de la batterie par un circuit électrique équivalent. Pour un SMSSM, les paramètres de ces deux équations dépendent de l'état discret caché S_t indexant le régime de fonctionnement de la batterie.

Ce modèle développé n'étant pas nécessairement identifiable, nous avons démontré que *l'identifiabilité générale* de l'ensemble de ses paramètres peut être assurée si la chaîne de Markov est irréductible et apériodique et une relation entre les observations Y_t et l'état continu inconnu X_t à un instant t_0 est disponible.

Étant donné un jeu de paramètres Θ, nous avons présenté une méthode pour estimer X_t d'une façon récursive à partir d'une estimation de X_{t-1}, d'une nouvelle observation y_t et d'une nouvelle entrée u_t. Cette méthode est basée sur le filtrage particulaire du fait qu'une estimation optimale de X_t au sens de l'erreur quadratique moyenne est un problème NP-complet. Nous avons validé cette méthode à partir d'une base de données de taille $T = 500$ simulée par un SMSSM à $\kappa = 3 : S_t \in \{1, 2, 3\}$. Les résultats montrent que le nombre de particules nécessaire à l'obtention d'une estimation précise de X_t est relativement faible ($N \simeq 10$ pour $\kappa = 3$), rendant cette méthode adaptée aux contraintes d'un calculateur embarqué.

Estimation des paramètres du modèle - Nous avons estimé les paramètres du modèle à partir d'une base d'apprentissage $\{y_{0:T}, u_{0:T}\}$ par deux approches :

1. une approche bayésienne utilisant l'échantillonnage de Gibbs comportant un filtrage particulaire,
2. une approche du type maximum de vraisemblance à travers un algorithme Monte-Carlo EM.

Concernant le filtrage particulaire de Gibbs, les résultats de validation sur des données simulées montrent que cet algorithme converge très lentement, probablement dû au fait que les nombreux paramètres sont estimés à partir des données cachées $X_{0:T}$ et $S_{0:T}$

L'algorithme MCEM, comme tout autre algorithme de type maximum de vraisemblance, pourrait ne pas estimer correctement Θ du fait que le SMSSM n'est pas forcément

identifiable. Pour surmonter cette difficulté, un terme de pénalisation dépendant des paramètres est ajouté à la vraisemblance afin d'obtenir un problème à solution unique. Dans cette étude, nous avons développé deux versions pénalisées de cet algorithme : MCEM pénalisé par les contraintes d'identifiabilité et MCEM pénalisé par une loi *a priori* $p(\Theta)$.

L'algorithme MCEM étant sensible à l'initialisation des paramètres, nous avons testé deux stratégies d'initialisation : indépendante et emboîtée. Les résultats de validation sur des données simulées montrent que l'initialisation emboîtée est plus adaptée aux SMSSMs que l'initialisation indépendante. De plus, lorsqu'un nombre approprié de particules est choisi ($N \simeq 100$ pour $\kappa = 3$ et $T = 500$), la dispersion des MLs estimés est relativement faible. Ainsi, les deux algorithmes pénalisés estiment convenablement l'ensemble de paramètres inconnus Θ. Toutefois, il convient de noter que la pénalisation par une loi *a priori* nécessite un choix adéquat des $12 \cdot \kappa$ hyperparamètres afin d'éviter le problème d'identifiabilité. Ce choix peut s'avérer compliqué, surtout lorsque les paramètres du modèle sont dépourvus de toute signification physique et que le nombre d'hyperparamètres est important (120 hyperparamètres pour $\kappa = 10$). Cependant en imposant l'échangeabilité des paramètres (i.e., les états ne sont pas différenciés), on peut réduire le nombre d'hyperparamètres à 12.

Pour la validation du modèle à partir des données réelles, nous avons utilisé uniquement l'algorithme MCEM pénalisé par les contraintes d'identifiabilité.

Estimation du nombre d'états cachés du modèle - À partir d'une base de données simulée par un SMSSM à $\kappa = 3$, nous avons testé la capacité de quatre critères de sélection de modèle à estimer le vrai nombre d'états de Markov, soit AIC, BIC, l'heuristique de pente et la validation croisée. Les résultats montrent que ces quatre critères ont un comportement satisfaisant et arrivent à sélectionner le vrai nombre d'états cachés pour cette base simulée. Toutefois, une des principales différences entre ces critères réside dans leur coût de calcul. L'AIC et le BIC sélectionnent, parmi les modèles candidats, respectivement celui qui les minimise. L'application de l'heuristique de pente exige un comportement linéaire de la vraisemblance des modèles à grand ordre. Si tel n'est pas le cas, la vraisemblance des modèles ne figurant pas *a priori* parmi les modèles candidats devrait être calculée. La validation croisée a le coût de calcul le plus élevé. En effet, pour chaque κ, la vraisemblance du modèle candidat n'étant pas utilisée directement, une vraisemblance croisée de deux modèles à nouveau estimés à partir de deux parts de la base d'apprentissage doit être calculée.

De plus, ces critères traitent différemment la problématique de sélection du modèle et peuvent choisir différents modèles. Dans ce cas, lorsque l'espace de stockage et/ou la capacité de calcul du système de contrôle sont limités, comme dans les applications embarquées, le modèle le plus parcimonieux est choisi. Lorsque ces contraintes sont

relâchées, des tests de validation supplémentaires sont réalisés afin d'évaluer les performances prédictives de chacun de ces modèles sélectionnés.

Validation du modèle proposé sur des batteries électriques : cellule, module et pack - Pour chacun de ces trois types de batteries, nous disposons de plusieurs bases de données issues des expérimentations à des températures ambiantes variées et des profils de courant simulant différentes situations de décharge.

À partir de ces bases de données, nous avons estimé les paramètres inconnus des modèles de SoC à $\kappa = 2, \ldots, 10$ par l'algorithme MCEM pénalisé par des contraintes d'identifiabilité. En effet, pour toutes ces expérimentations, la batterie est initialement complètement chargée. Son état de charge initial est ainsi connu : $x_0 = 100$. L'algorithme MCEM est initialisé d'une façon emboîtée et répété 20 fois avec différentes initialisations pour éviter les maxima locaux. Le nombre de particules utilisées est soigneusement choisi pour garantir que la dispersion du ML estimé soit relativement faible.

Pour chaque base d'apprentissage, nous avons ainsi neuf modèles candidats correspondant à $\kappa = 2, \ldots, 10$. Nous avons ensuite utilisé le critère BIC pour sélectionner le modèle le plus adéquat. En effet, la taille des bases d'apprentissage est relativement grande ; le choix de BIC paraît ainsi pertinent. De plus, les résultats montrent que le critère AIC surestime le nombre d'états cachés. Ce comportement confirme que ce critère n'est pas adapté aux modèles de mélange (cf. McLachlan and Peel [2000]). Quant au critère SHC, son choix est limité aux cas où le comportement du ML estimé en fonction de l'ordre du modèle est linéaire. Dans certains cas, il sous-estime même le nombre d'états cachés lorsque le comportement du ML estimé est linéaire par morceaux.

Dans le cas de la cellule et du module, nous avons effectué une validation croisée en réalisant l'apprentissage du modèle à partir d'une base de données et validant le modèle sélectionné par BIC sur toutes les autres bases.

Pour le pack, nous avons considéré des bases de données collectées sur 23 mois lors des expérimentations sur un véhicule électrique. Plus précisément, nous avons utilisé les expérimentations des 9 checkups réalisés d'une façon régulière comme des bases d'apprentissage pour estimer les paramètres inconnus des modèles de SoC à $\kappa = 2, \ldots, 10$. Nous avons validé ces modèles sur des données issues de cinq périodes de roulage réel du véhicule. Ces roulages ont été effectués sous des températures ambiantes et internes variées. De plus, l'état de santé du pack change d'une période de roulage à l'autre.

Les résultats de validation sur ces trois types de batteries mettent en évidence les avantages potentiels et l'utilité pratique des modèles à espaces d'états gouvernés par une chaîne de Markov dans le cadre d'estimation du SoC sous des conditions d'usage non contrôlables.

Toutefois, nous avons montré que la performance de prédiction du modèle SMSSM dépend de la base d'apprentissage. Pour l'ensemble des validations réalisées, nous avons observé que la meilleure base d'apprentissage est celle collectée lors d'une décharge complète par un profil de courant de type 4 dans le cas d'une cellule, d'une décharge complète par un profil Michelin à $5°C$ pour le module et d'une décharge complète par un profil NEDC à $25°C$ lors du 2ème checkup dans le cas d'un pack.

Dans cette étude, les paramètres sont appris à partir de la totalité de la base d'apprentissage avec un pas d'acquisition de mesures de courant et de tension d'une seconde. Une étude approfondie de l'influence du pas d'échantillonnage devrait être effectuée pour réduire la taille de la base d'apprentissage et ainsi le coût de la phase d'apprentissage.

De plus, nous avons vérifié l'interprétation physique des paramètres estimés du fait que les équations de transition et d'observation sont basées sur des modèles physiques. Les résultats montrent que les valeurs de ces paramètres sont proches des variables physiques correspondants. Les valeurs de $D(s)$ correspondent parfaitement à la résistance interne de la batterie. Nous retrouvons même la dépendance physique entre cette résistance et la température ambiante : les valeurs de $D(s)$ estimées à partir d'une base d'apprentissage issue d'une décharge à $0°C$ sont supérieures à celles estimées à partir d'une base d'apprentissage issue d'une décharge à $25°C$. Nous pouvons également subordonner que les changements d'état correspondent à des changements de comportement et de sollicitations de la batterie.

8.2 Perspectives

La problématique majeure des méthodes d'estimation de SoC réside dans l'estimation des paramètres inconnus du modèle de SoC. Il est clair que les méthodes d'apprentissage statistique offrent des techniques puissantes permettant d'estimer ces paramètres en se basant sur une base d'apprentissage formée des tests de charge/décharge d'une batterie. Cependant, ces méthodes nécessitent que cette base d'apprentissage soit exhaustive afin de pouvoir construire un modèle lisse et générique. Prenons l'exemple des véhicules électriques : les conditions d'usage de la batterie ne sont pas contrôlables et elles dépendent principalement du comportement du conducteur, de la météo et du type de trajet (urbain, routier, autoroutier). Pour construire un modèle générique de SoC, la base d'apprentissage devrait contenir des données collectées lors d'expérimentations de charge/décharge sous toutes les situations possibles de conditions d'usage et de caractéristiques internes. La formation d'une telle base est difficile voire impossible du fait du très grand nombre de combinaisons de situations à considérer. Il convient de noter que si on arrive à récupérer les données de roulage, on aura une base de données massive, représentative des différentes situations de conduite d'un véhicule. Pour cela, il serait nécessaire que le véhicule soit connecté afin de pouvoir stocker et traiter les données à distance.

Les tests de validation réalisés dans cette étude montrent que pour certaines bases d'apprentissage, le modèle de *SoC* proposé est robuste au changement de la température, à la variation du profil de courant et à la dégradation de l'état de santé de la batterie. Il convient de noter que ce résultat est observé pour l'ensemble d'expérimentations disponibles et jusqu'à présent nous ne pouvons pas le généraliser pour toutes les conditions d'usage et caractéristiques internes. En effet, lorsque la dynamique de la batterie est très différente de celle observée lors de l'apprentissage, le modèle serait incapable de s'adapter à cette nouvelle dynamique et sa capacité prédictive serait limitée. L'apprentissage à partir des décharges partielles met en avant cette limitation.

Toutefois, les méthodes d'apprentissage offrent des techniques prometteuses pour adapter en ligne les paramètres du modèle lorsque cela s'avère nécessaire. Ces paramètres devront ainsi être mis à jour pour s'adapter aux changements imprévus de l'évolution temporelle de données comme le courant et la tension. Pour ce faire, deux stratégies peuvent être distinguées : l'apprentissage adaptatif et l'apprentissage automatique en ligne.

L'apprentissage adaptatif consiste à mettre à jour les paramètres du modèle à chaque instant. Ce principe a été utilisé pour l'estimation du *SoC* par Pang et al. [2001] (cf. Section 2.3.3). Seulement, il consomme trop de ressources du BMS, et s'avère ainsi inadapté aux applications temps réel, d'autant plus que les paramètres du modèle varient peu pendant la charge/décharge, et ne nécessitent pas de mises à jour aussi fréquentes.

L'apprentissage automatique en ligne se déroule en deux étapes principales : l'étape de surveillance et l'étape de mise à jour. La Figure 8.1 présente ces deux étapes dans le cas d'estimation du *SoC* par un SMSSM.

Lors de la première étape, celle de surveillance, on s'intéresse à la détection de la "dérive" des données ; i.e. l'évolution de la distribution qui génère les données. Dans le cas d'une batterie, cette dérive pourrait apparaître lorsque les conditions d'usage de la batterie sont différentes de celles de la base apprentissage ou bien lorsque son état de santé se dégrade significativement. Ainsi, des mécanismes de détection de dérive sont mis en place pour fournir des informations concernant la date, la vitesse et la gravité de la dérive détectée. Ils sont basés soit sur des indicateurs de performances du modèle, soit sur des paramètres de la distribution de données.

Pour le modèle de *SoC* basé sur les SMSSMs, un indicateur de la performance du modèle est sa capacité à estimer la tension mesurée de la batterie. Ainsi, une dérive pourrait être détectée lorsque l'erreur d'estimation de la tension dépasse un seuil prédéfini.

Une autre méthode pour détecter la dérive du modèle consiste à comparer la distribution statistique de la tension de la batterie dans un intervalle de temps présent à celle dans un intervalle de temps prédéfini. Lorsque la différence entre les deux distributions est significative, on pourrait affirmer qu'une dérive est détectée.

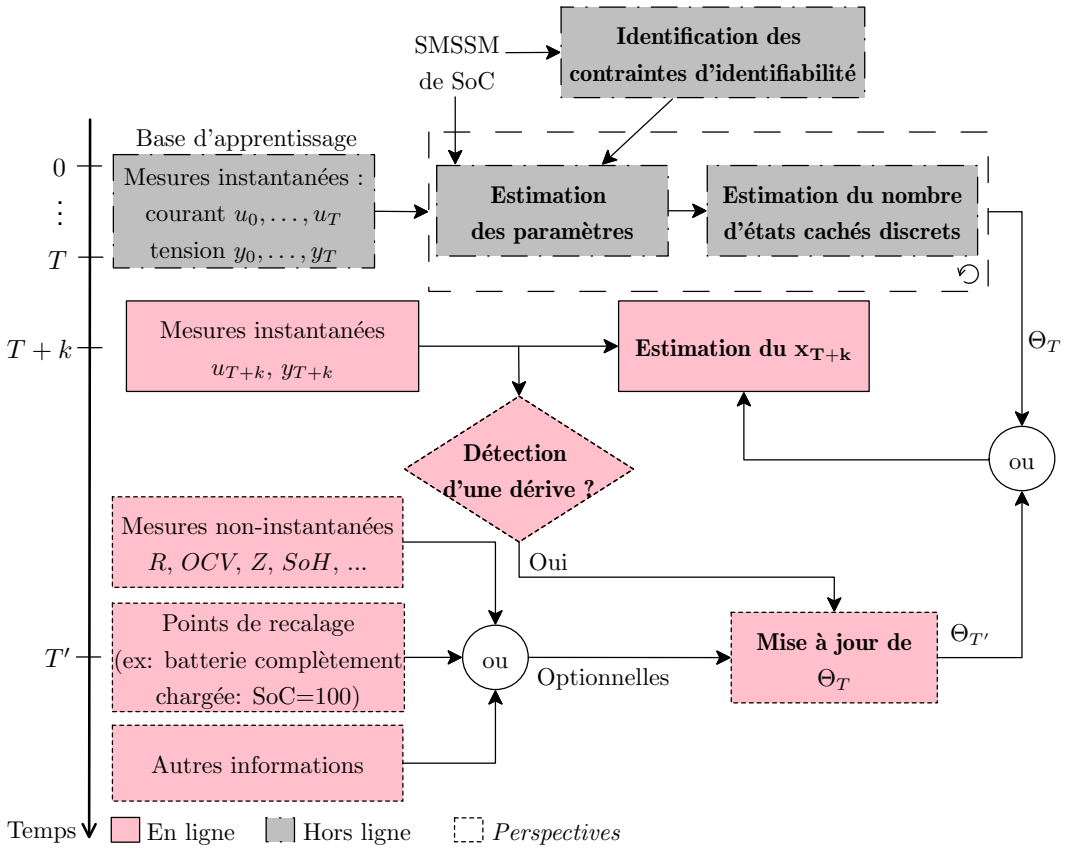

FIGURE 8.1: Conception d'un modèle de *SoC* "idéal"

Néanmoins, il est important de distinguer le bruit des capteurs d'un changement réel des données. Une méthode efficace devrait être robuste au bruit et s'adapter aux changements des données. Pour répondre à cette exigence, la détection de la dérive est généralement basée sur l'évolution des performances du modèle dans le passé récent, et non pas sur une mesure à un seul instant.

La deuxième étape, celle de mise à jour, consiste à estimer de nouveau les paramètres du modèle à partir de données récentes. La principale difficulté de cette étape est le choix de la taille de la fenêtre de données à utiliser. Trois possibilités ont été traitées dans la littérature : fixe (Sobhani and Beigy [2011]), dynamique (Gama and Castillo [2006]) et combinaison de plusieurs fenêtres de tailles différentes (Lazarescu et al. [2004]). La fenêtre à taille fixe est généralement utilisée lorsque la vitesse et la gravité de la dérive sont *a priori* connues. Quant à la fenêtre de taille dynamique, elle peut s'adapter à la nature de la dérive détectée. La combinaison de plusieurs fenêtres de tailles différentes permet d'adapter les paramètres du modèle aux changements progressifs de la distribution de données.

Toutefois, la limitation de la capacité de calcul et de stockage du BMS ainsi que les contraintes des applications temps réel doivent être prise en compte dans le cas des batteries électriques. De plus, il est à noter qu'en temps réel un vrai *SoC* n'est pas disponible. De prime abord, cela peut paraître comme étant une forte limitation de ces méthodes,

mais en pratique des informations supplémentaires comme la résistance interne, la capacité actuelle, l'état de santé et l'OCV peuvent être disponibles occasionnellement. Si tel est le cas, elles peuvent être intégrées dans la procédure d'apprentissage automatique en ligne et contribuent ainsi à l'amélioration de la mise à jour du modèle.

Ces deux étapes de surveillance et de mise à jour sont nécessaires pour la conception d'un modèle de SoC "idéal".

Appendix A

Kalman filter and smoother

Let us consider the following linear Gaussian state-space model (LGSSM):

$$X_t = X_{t-1} + Bu_t + W_t, \tag{A.1}$$
$$Y_t = CX_t + Du_t + D_0\Sigma_t, \tag{A.2}$$

where W_t and Σ_t are independent zero-mean Gaussian noises with variance respectively equal to σ_X^2 and σ_Y^2, and $\{W_t\}_{t\geq 0}$ and $\{\Sigma_t\}_{t\geq 0}$ are iid. As for the proposed *SoC* model, we consider that X_t, Y_t and u_t belong to $\mathbb{R}$. In this case, B, C, D, D_0, σ_X^2 and σ_Y^2 also belong to $\mathbb{R}$. In addition, we consider that $X_0 \sim \mathcal{N}(m_0, \sigma_0^2)$. For a general LGSSM, refer to Shumway and Stoffer [1982].

First of all, we recall a key property of the conditional distributions in a Gaussian context.

Theorem 1

Let us consider that

$$(Z_1,\, Z_2) \sim \mathcal{N}\left(\begin{bmatrix}\mu_1\\ \mu_2\end{bmatrix}\,;\,\begin{bmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{21} & \Sigma_{22}\end{bmatrix}\right), \tag{A.3}$$

then, the distribution of Z_1 conditional on $Z_2 = z_2$ is a Gaussian distribution given by

$$p(Z_1|Z_2 = z_2) = \mathcal{N}\left(\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(z_2 - \mu_2); \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right). \tag{A.4}$$

A.1 Kalman filter

The filtering aims at recursively estimating X_t given $y_{0:t}$ and $u_{0:t}$. It can be performed in two steps: prediction and correction.

Prediction step - a prior estimation

Given the sequence of observations $y_{0:t-1}$ up to $t-1$, $p_\Theta(x_t \mid y_{0:t-1})$ is a Gaussian distribution of mean $\widehat{x}_{t|t-1}$ and variance $\sigma^2_{t|t-1}$, where

$$\begin{aligned}\widehat{x}_{t|t-1} &= \mathbb{E}_{y_{0:t-1},\Theta}\left[X_t\right] = \mathbb{E}_{y_{0:t-1},\Theta}\left[X_{t-1} + B\,u_t + W_t\right] \\ &= \mathbb{E}_{y_{0:t-1},\Theta}\left[X_{t-1}\right] + B\,u_t \\ &= \widehat{x}_{t-1|t-1} + B\,u_t, \end{aligned} \tag{A.5}$$

and

$$\sigma^2_{t|t-1} = \mathrm{Var}\left(X_t |\, y_{0:t-1}, \Theta\right) = \mathrm{Var}\left(X_{t-1} + B\,u_t + W_t |\, y_{0:t-1}, \Theta\right). \tag{A.6}$$

Since X_{t-1} and W_t are independent, $\sigma^2_{t|t-1}$ can be calculated as follows:

$$\begin{aligned}\sigma^2_{t|t-1} &= \mathrm{Var}\left(X_{t-1} |\, y_{0:t-1}, \Theta\right) + \mathrm{Var}\left(W_t |\, y_{0:t-1}, \Theta\right) \\ &= \sigma^2_{t-1|t-1} + \sigma^2_X. \end{aligned} \tag{A.7}$$

Correction - a posterior estimation

Given the sequence of observations $y_{0:t}$ up to t, $p_\Theta(x_t \mid y_{0:t})$ is a Gaussian distribution of mean $\widehat{x}_{t|t}$ and variance $\sigma^2_{t|t}$. Based on Theorem 1, $\widehat{x}_{t|t}$ is calculated as follows:

$$\begin{aligned}\widehat{x}_{t|t} &= \mathbb{E}_{y_{0:t},\Theta}\left[X_t\right] = \mathbb{E}_{y_{0:t-1},y_t,\Theta}\left[X_t\right] \\ &= \widehat{x}_{t|t-1} + K_t \cdot \left(y_t - \mathbb{E}_{y_{0:t-1},\Theta}\left[Y_t\right]\right), \end{aligned} \tag{A.8}$$

where K_t is the "Kalman gain" given by

$$K_t = \frac{\mathrm{Cov}\left(X_t, Y_t \mid y_{0:t-1}, \Theta\right)}{\mathrm{Var}\left(Y_t \mid y_{0:t-1}, \Theta\right)}, \tag{A.9}$$

and

$$\mathbb{E}_{y_{0:t-1},\Theta}\left[Y_t\right] = \mathbb{E}_{y_{0:t-1},\Theta}\left[C \cdot X_t + D \cdot u_t + D_0 + \Sigma_t\right]. \tag{A.10}$$

Since X_t and Σ_t are independent, $\mathbb{E}_{y_{0:t-1},\Theta}\left[Y_t\right]$ is given by

$$\mathbb{E}_{y_{0:t-1},\Theta}\left[Y_t\right] = C \cdot \widehat{x}_{t|t-1} + D \cdot u_t + D_0. \tag{A.11}$$

The covariance of X_t and Y_t given the previous observations sequence $y_{0:t-1}$ can be calculated as follows:

$$\mathrm{Cov}\left(X_t\,;\, Y_t \mid y_{0:t-1}, \Theta\right) = \mathrm{Cov}\left(X_t\,;\, C \cdot X_t + D \cdot u_t + D_0 + \Sigma_t \mid y_{0:t-1}, \Theta\right). \tag{A.12}$$

Since X_t and Σ_t are independent, $\mathrm{Cov}\,(X_t\,;\,Y_t \mid y_{0:t-1}, \Theta)$ is given by

$$\begin{aligned}\mathrm{Cov}\,(X_t\,;\,Y_t \mid y_{0:t-1}, \Theta) &= C \cdot \mathrm{Var}\,(X_t \mid y_{0:t-1}, \Theta) \\ &= C \cdot \sigma^2_{t|t-1}. \end{aligned} \tag{A.13}$$

Similarly, the variance of Y_t given the previous observation sequence $y_{0:t-1}$ can be calculated as follows:

$$\begin{aligned}\mathrm{Var}\,(Y_t \mid y_{0:t-1}, \Theta) =& \mathrm{Var}\,(C \cdot X_t + D \cdot u_t + D_0 + \Sigma_t \mid y_{0:t-1}, \Theta) \\ =& C^2 \cdot \mathrm{Var}\,(X_t \mid y_{0:t-1}, \Theta) + \sigma^2_Y \\ =& C^2 \cdot \sigma^2_{t|t-1} + \sigma^2_Y. \end{aligned} \tag{A.14}$$

Replacing (A.13) and (A.14) in (A.9), the "Kalman gain" is given by

$$K_t = \frac{C \cdot \sigma_{t|t-1}}{C^2 \cdot \sigma^2_{t|t-1} + \sigma^2_Y}. \tag{A.15}$$

Based on the equations (A.15) and (A.11), $\widehat{x}_{t|t}$ can then be estimated as follows:

$$\widehat{x}_{t|t} = \widehat{x}_{t|t-1} + K_t \cdot \left(y_t - C \cdot \widehat{x}_{t|t-1} - D \cdot u_t - D_0\right). \tag{A.16}$$

Similarly, $\sigma^2_{t|t}$ is obtained using Theorem 1:

$$\begin{aligned}\sigma^2_{t|t} &= \mathrm{Var}\,(X_t \mid y_{0:t}, \Theta) = \mathrm{Var}\,(X_t \mid y_t, y_{0:t-1}, \Theta) \\ &= \mathrm{Var}\,(X_t \mid y_{0:t-1}, \Theta) - K_t \cdot \mathrm{Cov}\,(X_t; Y_t \mid y_{0:t-1}, \Theta) \\ &= \sigma^2_{t|t-1} - K_t \cdot \mathrm{Cov}\,(X_t;\, C \cdot X_t + D \cdot u_t + D_0 + \Sigma_t \mid y_{0:t-1}, \Theta) \\ &= \sigma^2_{t|t-1} - K_t \cdot C \cdot \sigma^2_{t|t-1}. \end{aligned} \tag{A.17}$$

A.2 Kalman smoother

The smoothing aims at recursively calculating x_t given the whole sequence of observations $y_{0:T}$, where $p_\Theta(x_t|y_{0:T})$ is a Gaussian distribution of mean $\widehat{x}_{t|T}$ and variance $\sigma^2_{t|T}$. Hence, the aim is to recursively calculate $\widehat{x}_{t|T}$ and $\sigma^2_{t|T}$, for $t = T-1, \ldots, 1$, considering that $\widehat{x}_{t|t}$ and $\sigma^2_{t|t}$ $(t = 1, \ldots, T)$ are obtained by filtering. Using the law of total expectation, $\widehat{x}_{t|T}$ can be recursively calculated as follows:

$$\widehat{x}_{t|T} = \mathbb{E}_{y_{0:T},\Theta}\left[X_t\right] = \mathbb{E}_{y_{0:T},\Theta}\left[\mathbb{E}_{X_{t+1},y_{0:T},\Theta}\left[X_t\right]\right]. \tag{A.18}$$

Based on Theorem 1, $\mathbb{E}_{X_{t+1},y_{0:T},\Theta}[X_t]$ is given by

$$\begin{aligned}\mathbb{E}_{X_{t+1},y_{0:T},\Theta}[X_t] &= \mathbb{E}_{X_{t+1},y_{0:t},\Theta}[X_t] \\ &= \mathbb{E}_{y_{0:t},\Theta}[X_t] + G_t\left(X_{t+1} - \mathbb{E}_{y_{0:t},\Theta}[X_{t+1}]\right) \\ &= \widehat{x}_{t|t} + G_t\left(X_{t+1} - \widehat{x}_{t+1|t}\right), \end{aligned} \tag{A.19}$$

where

$$G_t = \frac{\mathrm{Cov}\left(X_t; X_{t+1} \mid y_{0:t}, \Theta\right)}{\mathrm{Var}\left(X_{t+1} \mid y_{0:t}, \Theta\right)}. \tag{A.20}$$

The covariance of X_t and X_{t+1} given $y_{0:t}$ can be calculated as follows:

$$\mathrm{Cov}\left(X_t\,;\, X_{t+1} \mid y_{0:t}, \Theta\right) = \mathrm{Cov}\left(X_t\,;\, X_t + B \cdot u_t + W_t\right) \mid y_{0:t}, \Theta\right). \tag{A.21}$$

Since X_t and W_t are independent, $\mathrm{Cov}\left(X_t\,;\, X_{t+1} \mid y_{0:t}, \Theta\right)$ is given by

$$\mathrm{Cov}\left(X_t\,;\, X_{t+1} \mid y_{0:t}, \Theta\right) = \mathrm{Var}\left(X_t \mid y_{0:t}, \Theta\right) = \sigma^2_{t|t}. \tag{A.22}$$

Accordingly, G_t is given by:

$$G_t = \frac{\sigma^2_{t|t}}{\sigma^2_{t+1|t}}. \tag{A.23}$$

Replacing (A.23) and (A.19) in (A.18), $\widehat{x}_{t|T}$ is then estimated as follows:

$$\begin{aligned}\widehat{x}_{t|T} &= \mathbb{E}\left[X_t \mid y_{0:t}, \Theta\right] + G_t\left(\mathbb{E}\left[X_{t+1} \mid y_{0:T}, \Theta\right] - \mathbb{E}\left[X_{t+1} \mid y_{0:t}, \Theta\right]\right) \\ &= \widehat{x}_{t|t} + G_t \cdot \left(\widehat{x}_{t+1|T} - \widehat{x}_{t+1|t}\right). \end{aligned} \tag{A.24}$$

Using the law of total variance, the variance of X_t given the whole sequence of observations $y_{0:T}$ is given by

$$\begin{aligned}\sigma^2_{t|T} &= \mathrm{Var}\left(X_t \mid y_{0:T}, \Theta\right) \\ &= \mathbb{E}_{y_{0:T},\Theta}\left[\mathrm{Var}\left(X_t \mid X_{t+1}, y_{0:T}, \Theta\right)\right] + \mathrm{Var}\left(\mathbb{E}_{X_{t+1},y_{0:T},\Theta}[X_t] \,|y_{0:T}, \Theta\right) \\ &= \mathbb{E}_{y_{0:T},\Theta}\left[\mathrm{Var}\left(X_t \mid X_{t+1}, y_{0:t}, \Theta\right)\right] + \mathrm{Var}\left(\mathbb{E}_{X_{t+1},y_{0:t},\Theta}[X_t] \,|y_{0:T}, \Theta\right) \\ &= \sigma^2_{t|t} - G_t\sigma^2_{t|t} + G_t^2\sigma^2_{t+1|T} \\ &= \sigma^2_{t|t} + G_t^2(\sigma^2_{t+1|T} - \sigma^2_{t+1|t}). \end{aligned} \tag{A.25}$$

Appendix B

Choice of the conjugate prior of the state-space parameters

B.1 Prior and posterior of $B(s)$

Assuming $B(s)$ is to be estimated, the conjugate prior associated with this distribution is normal $\mathcal{N}(\alpha(s), \beta(s))$, as then $p(B(s) \mid \Theta_{-B(s)}, s_{0:T}, x_{0:T}, y_{1:T})$ is proportional to

$$
\begin{aligned}
&\propto p(B(s)) \cdot p_\Theta(s_{0:T}, x_{0:T}, y_{1:T}) \qquad \text{(B.1)}\\
&\propto p(B(s)) \cdot p_\Theta(s_0, x_0) \cdot \prod_{t=1}^{T} p_\Theta(s_t \mid s_{t-1}) \cdot p_\Theta(x_t \mid x_{t-1}, s_t) \cdot p_\Theta(y_t \mid x_t, s_t)\\
&\propto p(B(s)) \cdot \prod_{t=1}^{T} p_\Theta(x_t \mid x_{t-1}, s_t)\\
&\propto p(B(s)) \cdot \prod_{\substack{t=1\\ s_t=s}}^{T} \exp\left(\frac{-1}{2\sigma_X^2(s)} \left(x_t - x_{t-1} - B(s) u_t \Delta t\right)^2\right)\\
&\propto \exp\left(\frac{-1}{2\beta(s)}(B(s) - \alpha(s))^2\right) \cdot \exp\left(\frac{-1}{2\sigma_X^2(s)} \sum_{\substack{t=1\\ s_t=s}}^{T} \left(x_t - x_{t-1} - B(s) u_t \Delta t\right)^2\right)\\
&\propto \exp\left\{-\frac{1}{2}\left[B(s)^2 \left(\frac{1}{\beta(s)} + \frac{\sum_{\substack{t=1\\ s_t=s}}^{T} u_t^2 \Delta t^2}{\sigma_X^2(s)}\right) - 2B(s)\left(\frac{\alpha(s)}{\beta(s)} + \frac{\sum_{\substack{t=1\\ s_t=s}}^{T} u_t \Delta t (x_t - x_{t-1})}{\sigma_X^2(s)}\right)\right]\right\}.
\end{aligned}
$$

Inspecting the right-hand side shows that it is proportional, in $B(s)$, to the density of a normal distribution with mean $m_B(s)$ and variance $\sigma_B^2(s)$, where:

$$m_B(s) = \frac{\sum_{\substack{t=1 \\ s_t=s}}^{T} u_t \Delta t (x_t - x_{t-1}) + \frac{\alpha(s)}{\beta(s)} \sigma_X^2(s)}{\sum_{\substack{t=1 \\ s_t=s}}^{T} u_t^2 \Delta t^2 + \frac{\sigma_X^2(s)}{\beta(s)}} \tag{B.2}$$

$$\sigma_B^2(s) = \frac{\sigma_X^2(s)}{\sum_{\substack{t=1 \\ s_t=s}}^{T} u_t^2 \Delta t^2 + \frac{\sigma_X^2(s)}{\beta(s)}} \tag{B.3}$$

B.2 Prior and posterior of $C(s)$

Assuming $C(s)$ is to be estimated, the conjugate prior associated with this distribution is normal $\mathcal{N}(\gamma(s), \mu(s))$, as then $p(C(s) \mid \Theta_{-C(s)}, s_{0:T}, x_{0:T}, y_{1:T})$ is proportional to

$$\begin{aligned}
&\propto p(C(s)) \cdot p_\Theta(s_{0:T}, x_{0:T}, y_{1:T}) \qquad \text{(B.4)}\\
&\propto p(C(s)) \cdot \prod_{t=1}^{T} p_\Theta(y_t \mid x_t, s_t) \\
&\propto p(C(s)) \cdot \prod_{\substack{t=1 \\ s_t=s}}^{T} \exp\left(\frac{-1}{2\sigma_Y^2(s)} \left(y_t - C(s)x_t - D(s)u_t - D_0(s)\right)^2 \right) \\
&\propto \exp\left(\frac{-1}{2\mu(s)} (C(s) - \gamma(s))^2 \right) \cdot \exp\left(\frac{-1}{2\sigma_Y^2(s)} \sum_{\substack{t=1 \\ s_t=s}}^{T} \left(y_t - C(s)x_t - D(s)u_t - D_0(s)\right)^2 \right) \\
&\propto \exp\left\{ -\frac{1}{2} \left[C(s)^2 \left(\frac{1}{\mu(s)} + \frac{\sum_{\substack{t=1 \\ s_t=s}}^{T} x_t^2}{\sigma_Y^2(s)} \right) - 2C(s) \left(\frac{\gamma(s)}{\mu(s)} + \frac{\sum_{\substack{t=1 \\ s_t=s}}^{T} x_t (y_t - D(s)u_t - D_0(s))}{\sigma_Y^2(s)} \right) \right] \right\}.
\end{aligned}$$

Inspecting the right-hand side shows that it is proportional, in $C(s)$, to the density of a normal distribution with mean $m_C(s)$ and variance $\sigma_C^2(s)$, where:

$$m_C(s) = \frac{\sum_{\substack{t=1 \\ s_t=s}}^{T} x_t \left(y_t - D(s)u_t - D_0(s)\right) + \frac{\gamma(s)}{\mu(s)} \sigma_Y^2(s)}{\sum_{\substack{t=1 \\ s_t=s}}^{T} x_t^2 + \frac{\sigma_Y^2(s)}{\mu(s)}} \tag{B.5}$$

$$\sigma_C^2(s) = \frac{\sigma_Y^2(s)}{\sum_{\substack{t=1 \\ s_t=s}}^{T} x_t^2 + \frac{\sigma_Y^2(s)}{\mu(s)}} \tag{B.6}$$

B.3 Prior and posterior of $D(s)$

Assuming $D(s)$ is to be estimated, the conjugate prior associated with this distribution is normal $\mathcal{N}(\zeta(s), \epsilon(s))$, as then $p(D(s) \mid \Theta_{-D(s)}, s_{0:T}, x_{0:T}, y_{1:T})$ is proportional to

$$
\begin{aligned}
&\propto p(D(s)) \cdot p_\Theta(s_{0:T}, x_{0:T}, y_{1:T}) \qquad \text{(B.7)}\\
&\propto p(D(s)) \cdot \prod_{t=1}^{T} p_\Theta(y_t \mid x_t, s_t)\\
&\propto p(D(s)) \cdot \prod_{\substack{t=1\\ s_t=s}}^{T} \exp\left(\frac{-1}{2\sigma_Y^2(s)}\left(y_t - C(s)x_t - D(s)u_t - D_0(s)\right)^2\right)\\
&\propto \exp\left(\frac{-1}{2\epsilon(s)}(D(s)-\zeta(s))^2\right) \cdot \exp\left(\frac{-1}{2\sigma_Y^2(s)} \sum_{\substack{t=1\\ s_t=s}}^{T} \left(y_t - C(s)x_t - D(s)u_t - D_0(s)\right)^2\right)\\
&\propto \exp\left\{-\frac{1}{2}\left[D(s)^2\left(\frac{1}{\epsilon(s)} + \frac{\sum_{\substack{t=1\\ s_t=s}}^{T} u_t^2}{\sigma_Y^2(s)}\right) - 2D(s)\left(\frac{\zeta(s)}{\epsilon(s)} + \frac{\sum_{\substack{t=1\\ s_t=s}}^{T} u_t(y_t - C(s)x_t - D_0(s))}{\sigma_Y^2(s)}\right)\right]\right\}.
\end{aligned}
$$

Inspecting the right-hand side shows that it is proportional, in $D(s)$, to the density of a normal distribution with mean $m_D(s)$ and variance $\sigma_D^2(s)$, where:

$$
m_D(s) = \frac{\sum_{\substack{t=1\\ s_t=s}}^{T} u_t\left(y_t - C(s)x_t - D_0(s)\right) + \frac{\zeta(s)}{\epsilon(s)}\sigma_Y^2(s)}{\sum_{\substack{t=1\\ s_t=s}}^{T} u_t^2 + \frac{\sigma_Y^2(s)}{\epsilon(s)}} \qquad \text{(B.8)}
$$

$$
\sigma_D^2(s) = \frac{\sigma_Y^2(s)}{\sum_{\substack{t=1\\ s_t=s}}^{T} u_t^2 + \frac{\sigma_Y^2(s)}{\epsilon(s)}} \qquad \text{(B.9)}
$$

B.4 Prior and posterior of $D_0(s)$

Assuming $D_0(s)$ is to be estimated, the conjugate prior associated with this distribution is normal $\mathcal{N}(\zeta_0(s), \epsilon_0(s))$, as then $p(D_0(s) \mid \Theta_{-D_0(s)}, s_{0:T}, x_{0:T}, y_{1:T})$ is proportional to

$$\begin{aligned}
&\propto p(D_0(s)) \cdot p_\Theta(s_{0:T}, x_{0:T}, y_{1:T}) \qquad \text{(B.10)}\\
&\propto p(D_0(s)) \cdot \prod_{t=1}^{T} p_\Theta(y_t \mid x_t, s_t)\\
&\propto p(D_0(s)) \cdot \prod_{\substack{t=1\\ s_t=s}}^{T} \exp\left(\frac{-1}{2\sigma_Y^2(s)}\left(y_t - C(s)x_t - D(s)u_t - D_0(s)\right)^2\right)\\
&\propto \exp\left(\frac{-1}{2\epsilon_0(s)}(D_0(s) - \zeta_0(s))^2\right) \cdot \exp\left(\frac{-1}{2\sigma_Y^2(s)} \sum_{\substack{t=1\\ s_t=s}}^{T} \left(y_t - C(s)x_t - D(s)u_t - D_0(s)\right)^2\right)\\
&\propto \exp\left\{-\frac{1}{2}\left[D_0(s)^2\left(\frac{1}{\epsilon_0(s)} + \frac{\sum_{t=1}^{T} \mathbb{1}(s_t = s)}{\sigma_Y^2(s)}\right) - 2D_0(s)\left(\frac{\zeta_0(s)}{\epsilon_0(s)} + \frac{\sum_{\substack{t=1\\ s_t=s}}^{T} (y_t - C(s)x_t - D(s)u_t)}{\sigma_Y^2(s)}\right)\right]\right\}.
\end{aligned}$$

Inspecting the right-hand side shows that it is proportional, in $D_0(s)$, to the density of a normal distribution with mean $m_{D_0}(s)$ and variance $\sigma^2_{D_0}(s)$, where:

$$m_{D_0}(s) = \frac{\sum_{t=1}^{T} \mathbb{1}(s_t = s)\left(y_t - C(s)x_t - D(s)u_t\right) + \frac{\zeta_0(s)}{\epsilon_0(s)}\sigma_Y^2(s)}{\sum_{t=1}^{T} \mathbb{1}(s_t = s) + \frac{\sigma_Y^2(s)}{\epsilon_0(s)}} \qquad \text{(B.11)}$$

$$\sigma^2_{D_0}(s) = \frac{\sigma_Y^2(s)}{\sum_{t=1}^{T} \mathbb{1}(s_t = s) + \frac{\sigma_Y^2(s)}{\epsilon_0(s)}} \qquad \text{(B.12)}$$

B.5 Prior and posterior of $\sigma_X^2(s)$

Assuming $\sigma_X^2(s)$ is to be estimated, the conjugate prior is instead the inverse Gamma distribution $IG(\sigma_X^2(s); \varrho(s), \tau(s))$, with density

$$p(\sigma_X^2(s)) = \frac{\tau(s)^{\varrho}(s)}{\Gamma(\varrho(s))\left(\sigma_X(s)\right)^{\varrho(s)+1}} e^{\frac{-\tau(s)}{\sigma_X(s)}}.$$

Indeed, with this prior, $p(\sigma_X^2(s) \mid \Theta_{-\sigma_X^2(s)}, s_{0:T}, x_{0:T}, y_{1:T})$ is proportional to

$$\begin{aligned}
&\propto p(\sigma_X^2(s)) \cdot \prod_{t=1}^{T} p_\Theta(x_t \mid x_{t-1}, s_t) \qquad \text{(B.13)}\\
&\propto p(\sigma_X^2(s)) \cdot \prod_{\substack{t=1\\ s_t=s}}^{T} \frac{1}{\sqrt{\sigma_X^2(s)}} \exp\left(\frac{-1}{2\sigma_X^2(s)} \left(x_t - x_{t-1} - B(s)u_t\Delta t\right)^2\right)\\
&\propto \left(\sigma_X^2(s)\right)^{\varrho(s)-1} \left(\sigma_X^2(s)\right)^{-\frac{1}{2}\sum_{t=1}^{T} \mathbb{1}(s_t=s)} \exp\left(\frac{-1}{\sigma_X^2(s)} \left[\tau(s) + \sum_{t=1}^{T} \mathbb{1}(s_t=s) \left(x_t - x_{t-1} - B(s)u_t\Delta t\right)^2\right)\right].
\end{aligned}$$

Hence, the posterior distribution of $\sigma_X^2(s)$ is the inverse gamma distribution

$$IG\left(\varrho(s) + \frac{\sum_{t=1}^{T} \mathbb{1}(s_t=s)}{2}, \tau(s) + \sum_{t=1}^{T} \mathbb{1}(s_t=s) \left(x_t - x_{t-1} - B(s)u_t\Delta t\right)^2\right). \qquad \text{(B.14)}$$

B.6 Prior and posterior of $\sigma_Y^2(s)$

Assuming $\sigma_Y^2(s)$ is to be estimated, the conjugate prior is instead the inverse Gamma distribution $IG(\sigma_Y^2(s); \upsilon(s), \psi(s))$, with density

$$p(\sigma_Y^2(s)) = \frac{\psi(s)^{\upsilon}(s)}{\Gamma(\upsilon(s)) \left(\sigma_Y(s)\right)^{\upsilon(s)+1}} e^{\frac{-\psi(s)}{\sigma_Y(s)}}.$$

Indeed, with this prior, $p(\sigma_Y^2(s) \mid \Theta_{-\sigma_Y^2(s)}, s_{0:T}, x_{0:T}, y_{1:T})$ is proportional to

$$\begin{aligned}
&\propto p(\sigma_Y^2(s)) \cdot \prod_{t=1}^{T} p_\Theta(y_t \mid x_t, s_t) s \qquad \text{(B.15)}\\
&\propto p(\sigma_Y^2(s)) \cdot \prod_{\substack{t=1\\ s_t=s}}^{T} \frac{1}{\sqrt{\sigma_Y^2(s)}} \exp\left(\frac{-1}{2\sigma_Y^2(s)} \left(y_t - C(s)x(t) - D(s)u_t - D_0(s)\right)^2\right)\\
&\propto \left(\sigma_Y^2(s)\right)^{\upsilon(s)-1} \left(\sigma_Y^2(s)\right)^{-\frac{1}{2}\sum_{t=1}^{T} \mathbb{1}(s_t=s)} \exp\left(\frac{-1}{\sigma_Y^2(s)} \left[\psi(s) + \sum_{t=1}^{T} \mathbb{1}(s_t=s) \left(y_t - C(s)x_t - D(s)u_t - D_0(s)\right)^2\right)\right].
\end{aligned}$$

Hence, the posterior distribution of $\sigma_Y^2(s)$ is the inverse gamma distribution

$$IG\left(\upsilon(s) + \frac{\sum_{t=1}^{T} \mathbb{1}(s_t=s)}{2}, \psi(s) + \sum_{t=1}^{T} \mathbb{1}(s_t=s) \left(y_t - C(s)x_t - D(s)u_t - D_0(s)\right)^2\right). \qquad \text{(B.16)}$$

Appendix C

Results of validation of the SoC model using real-life battery data

C.1 Cell battery of type S

Validation \ Learning		$T_{\mathbf{amb}} = 0°C$							
		H1	H2	H3	H4	U1	U2	U3	U4
$T_{\mathbf{amb}} = 0°C$	H1	0.28 ± 0.13 (0.52)	0.17 ± 0.13 (−0.45)	0.11 ± 0.07 (0.30)	0.10 ± 0.07 (0.28)	0.16 ± 0.10 (0.39)	0.51 ± 0.34 (−1.23)	0.54 ± 0.35 (−1.25)	0.83 ± 0.51 (1.84)
	H2	0.19 ± 0.12 (0.38)	0.21 ± 0.10 (−0.43)	0.06 ± 0.07 (0.18)	0.07 ± 0.08 (0.21)	0.10 ± 0.10 (0.28)	0.58 ± 0.33 (−1.23)	0.59 ± 0.33 (−1.23)	0.73 ± 0.49 (1.61)
	H3	0.32 ± 0.15 (0.54)	0.15 ± 0.10 (−0.34)	0.13 ± 0.09 (0.33)	0.15 ± 0.10 (0.36)	0.19 ± 0.13 (0.44)	0.55 ± 0.35 (−1.15)	0.58 ± 0.35 (−1.17)	0.93 ± 0.57 (1.91)
	H4	0.22 ± 0.23 (−0.45)	0.52 ± 0.40 (−1.03)	0.30 ± 0.26 (−0.59)	0.28 ± 0.24 (−0.54)	0.25 ± 0.22 (−0.49)	0.92 ± 0.65 (−1.87)	0.95 ± 0.66 (−1.87)	0.54 ± 0.29 (1.17)
	U1	1.03 ± 0.44 (1.77)	1.89 ± 0.96 (3.51)	0.53 ± 0.19 (0.92)	0.76 ± 0.36 (1.48)	0.29 ± 0.33 (−0.58)	0.95 ± 1.11 (−2.67)	0.27 ± 0.27 (−0.66)	0.76 ± 0.43 (1.27)
	U2	0.93 ± 0.43 (1.65)	1.50 ± 0.78 (2.85)	0.56 ± 0.23 (0.91)	0.70 ± 0.33 (1.28)	0.30 ± 0.13 (0.52)	0.84 ± 0.44 (1.46)	0.14 ± 0.16 (−0.34)	1.02 ± 0.38 (1.43)
	U3	1.41 ± 0.63 (2.30)	2.24 ± 1.13 (4.00)	0.90 ± 0.36 (1.42)	1.13 ± 0.53 (1.99)	0.38 ± 0.25 (0.73)	1.06 ± 1.15 (−2.49)	0.19 ± 0.17 (0.43)	1.08 ± 0.50 (1.77)
	U4	1.10 ± 0.64 (2.17)	1.97 ± 1.16 (3.92)	0.58 ± 0.37 (1.30)	0.84 ± 0.57 (1.91)	0.12 ± 0.12 (0.36)	0.83 ± 0.99 (−2.49)	0.23 ± 0.21 (0.52)	0.76 ± 0.37 (1.29)
$T_{\mathbf{amb}} = 25°C$	H1	0.20 ± 0.15 (0.51)	0.92 ± 0.54 (−1.83)	0.46 ± 0.24 (−0.86)	0.44 ± 0.24 (−0.86)	0.35 ± 0.19 (−0.69)	1.48 ± 0.84 (−2.86)	1.54 ± 0.88 (−2.95)	0.88 ± 0.57 (1.90)
	H2	0.25 ± 0.16 (0.60)	0.29 ± 0.18 (−0.71)	0.09 ± 0.06 (−0.27)	0.06 ± 0.04 (−0.20)	0.04 ± 0.03 (−0.15)	0.61 ± 0.37 (−1.41)	0.65 ± 0.39 (−1.46)	0.62 ± 0.39 (1.39)
	H3	0.11 ± 0.10 (0.34)	1.00 ± 0.69 (−2.12)	0.55 ± 0.42 (−1.24)	0.51 ± 0.39 (−1.16)	0.47 ± 0.36 (−1.07)	1.54 ± 1.01 (−3.17)	1.61 ± 1.04 (−3.25)	0.76 ± 0.42 (1.57)
	H4	0.70 ± 0.34 (3.33)	1.77 ± 0.86 (−3.18)	1.34 ± 0.62 (−2.28)	1.31 ± 0.60 (−2.21)	1.23 ± 0.55 (−2.04)	2.32 ± 1.17 (−4.23)	2.40 ± 1.22 (−4.38)	0.25 ± 0.29 (−0.57)
	U1	1.07 ± 0.73 (−2.07)	0.13 ± 0.14 (−0.29)	1.59 ± 0.94 (−2.77)	1.35 ± 0.78 (−2.33)	1.23 ± 0.86 (−2.35)	0.91 ± 0.98 (−2.39)	1.40 ± 0.88 (−2.61)	0.88 ± 0.57 (1.90)
	U2	0.36 ± 0.14 (0.55)	0.91 ± 0.49 (1.71)	0.06 ± 0.07 (−0.20)	0.11 ± 0.03 (0.18)	0.32 ± 0.10 (0.45)	1.03 ± 0.55 (1.68)	0.36 ± 0.21 (−0.73)	0.62 ± 0.39 (1.39)
	U3	1.21 ± 0.93 (−2.47)	0.33 ± 0.35 (−0.59)	1.75 ± 1.14 (−3.15)	1.47 ± 0.97 (−2.69)	1.32 ± 1.02 (−2.61)	1.07 ± 1.14 (−2.38)	1.50 ± 0.96 (−3.05)	0.76 ± 0.42 (1.57)
	U4	1.14 ± 0.86 (−2.30)	0.28 ± 0.30 (−0.49)	1.67 ± 1.07 (−3.01)	1.43 ± 0.91 (−2.58)	1.26 ± 0.97 (−2.47)	1.00 ± 1.07 (−2.33)	1.48 ± 0.96 (−2.99)	0.25 ± 0.29 (−0.57)

TABLE C.1: Learning with the height current profiles at ambient temperature equal to $0°C$ and validation using these height current profiles at ambient temperature equal to $0°C$ and $25°C$: mean absolute error ± standard deviation error (maximum error). The model is learned using a MCEM algorithm penalized with identifiability constraints and selected with BIC.

	Learning / Validation	$T_{\mathbf{amb}} = 25°C$							
		H1	H2	H3	H4	U1	U2	U3	U4
$T_{\mathbf{amb}} = 0°C$	H1	0.30 ± 0.18 (0.66)	0.07 ± 0.05 (+0.22)	0.17 ± 0.10 (0.39)	0.47 ± 0.29 (1.05)	0.84 ± 0.52 (1.86)	0.31 ± 0.27 (0.97)	1.01 ± 0.71 (2.56)	1.07 ± 0.74 (2.68)
	H2	0.22 ± 0.18 (0.56)	0.04 ± 0.05 (0.13)	0.10 ± 0.10 (0.29)	0.39 ± 0.28 (0.90)	0.73 ± 0.50 (1.62)	0.33 ± 0.18 (−0.95)	0.37 ± 0.32 (0.99)	0.44 ± 0.35 (1.15)
	H3	0.35 ± 0.23 (0.76)	0.10 ± 0.06 (0.24)	0.17 ± 0.12 (0.41)	0.55 ± 0.34 (1.15)	0.94 ± 0.58 (1.93)	0.18 ± 0.16 (0.56)	0.93 ± 0.60 (1.85)	1.06 ± 0.67 (2.10)
	H4	0.12 ± 0.14 (−0.30)	0.33 ± 0.28 (−0.66)	0.25 ± 0.22 (−0.49)	0.15 ± 0.11 (0.43)	0.54 ± 0.29 (1.18)	0.09 ± 0.11 (−0.27)	0.80 ± 0.47 (1.61)	0.85 ± 0.51 (1.72)
	U1	1.09 ± 0.59 (2.35)	1.89 ± 1.73 (−4.79)	0.70 ± 0.32 (1.45)	1.13 ± 0.53 (2.12)	1.65 ± 0.80 (3.00)	0.74 ± 1.01 (4.03)	1.45 ± 0.76 (2.87)	0.60 ± 0.64 (1.07)
	U2	0.91 ± 0.45 (1.74)	0.87 ± 1.02 (−3.02)	0.67 ± 0.29 (1.15)	0.99 ± 0.47 (1.78)	1.36 ± 0.68 (2.51)	0.21 ± 0.13 (0.52)	1.09 ± 0.57 (2.05)	0.56 ± 0.31 (1.00)
	U3	1.45 ± 0.75 (2.81)	1.77 ± 1.75 (−4.80)	1.06 ± 0.49 (1.96)	1.50 ± 0.70 (2.63)	2.00 ± 0.97 (3.49)	0.99 ± 0.94 (4.25)	1.90 ± 0.88 (3.58)	0.71 ± 0.58 (1.40)
	U4	1.17 ± 0.80 (2.78)	1.95 ± 1.60 (−4.63)	0.78 ± 0.53 (1.92)	1.20 ± 0.73 (2.54)	1.74 ± 1.00 (3.43)	0.84 ± 1.27 (4.77)	1.45 ± 0.96 (3.25)	0.48 ± 0.54 (−0.93)
$T_{\mathbf{amb}} = 25°C$	H1	0.06 ± 0.03 (−0.15)	0.50 ± 0.28 (−0.99)	0.36 ± 0.19 (−0.71)	0.24 ± 0.18 (0.61)	0.87 ± 0.56 (1.88)	0.37 ± 0.28 (−1.06)	0.31 ± 0.34 (−1.03)	1.15 ± 0.92 (2.39)
	H2	0.12 ± 0.08 (0.31)	0.08 ± 0.06 (−0.25)	0.02 ± 0.02 (−0.08)	0.28 ± 0.18 (0.66)	0.61 ± 0.38 (1.38)	0.20 ± 0.24 (−0.65)	0.24 ± 0.28 (−0.62)	0.72 ± 0.60 (1.43)
	H3	0.19 ± 0.19 (−0.50)	0.56 ± 0.42 (−1.26)	0.48 ± 0.36 (−1.07)	0.13 ± 0.10 (0.35)	0.76 ± 0.42 (1.57)	1.46 ± 0.59 (−2.32)	1.35 ± 0.53 (−2.14)	0.35 ± 0.41 (0.88)
	H4	0.98 ± 0.40 (−1.51)	1.36 ± 0.63 (−2.32)	1.26 ± 0.56 (−2.08)	0.66 ± 0.25 (−0.91)	0.24 ± 0.28 (−0.57)	0.97 ± 0.43 (−1.75)	0.88 ± 0.41 (−1.58)	0.45 ± 0.45 (1.17)
	U1	0.74 ± 0.41 (−1.33)	1.27 ± 1.10 (−2.72)	1.12 ± 0.60 (−1.93)	0.74 ± 0.46 (−1.38)	0.18 ± 0.19 (−0.49)	2.43 ± 3.60 (11.69)	1.06 ± 0.50 (−1.87)	1.12 ± 0.79 (−2.41)
	U2	0.30 ± 0.15 (0.59)	0.50 ± 0.27 (0.82)	0.13 ± 0.04 (0.20)	0.42 ± 0.20 (0.73)	0.81 ± 0.42 (1.49)	0.24 ± 0.29 (0.91)	0.13 ± 0.16 (0.44)	1.15 ± 0.53 (2.14)
	U3	0.87 ± 0.59 (−1.74)	1.49 ± 1.32 (−3.27)	1.27 ± 0.80 (−2.38)	0.85 ± 0.65 (−1.73)	0.37 ± 0.40 (−0.87)	2.22 ± 3.44 (11.43)	0.88 ± 0.59 (−1.75)	0.40 ± 0.46 (−1.06)
	U4	0.85 ± 0.55 (−1.67)	1.39 ± 1.25 (−3.07)	1.21 ± 0.74 (−2.23)	0.83 ± 0.60 (−1.66)	0.31 ± 0.35 (−0.77)	2.44 ± 3.59 (11.32)	1.26 ± 0.58 (−2.15)	0.63 ± 0.61 (−1.68)

TABLE C.2: Learning with the height current profiles at ambient temperature equal to $25°C$ and validation using these height current profiles at ambient temperature equal to $0°C$ and $25°C$: mean absolute error ± standard deviation error (maximum error). The model is learned using a MCEM algorithm penalized with identifiability constraints and selected with BIC.

C.2 Module batteries of type L

Learning \ Validation		NEDC						Michelin					
		100% $-5°C$	100% $+5°C$	100% $25°C$	30% $-5°C$	30% $+5°C$	30% $25°C$	100% $-5°C$	100% $+5°C$	100% $25°C$	30% $-5°C$	30% $+5°C$	30% $25°C$
NEDC	100% $-5°C$	1.12 ± 0.78 (−2.29)	0.18 ± 0.08 (−0.35)	0.64 ± 0.36 (−1.21)	0.55 ± 0.69 (−3.01)	2.53 ± 3.20 (12.58)	14.59 ± 8.96 (−27.75)	0.17 ± 0.10 (−0.35)	0.23 ± 0.12 (−0.49)	1.07 ± 0.75 (−2.70)	1.80 ± 1.45 (−5.07)	6.87 ± 4.77 (−13.79)	4.99 ± 2.82 (−9.03)
	100% $+5°C$	7.13 ± 7.33 (20.69)	0.32 ± 0.16 (−0.58)	0.35 ± 0.41 (1.15)	6.60 ± 4.47 (13.20)	7.10 ± 9.00 (33.68)	6.22 ± 5.76 (−25.05)	0.31 ± 0.18 (−0.64)	0.33 ± 0.19 (−0.71)	0.53 ± 0.25 (−1.04)	0.33 ± 0.36 (−0.99)	1.01 ± 1.16 (−2.32)	4.24 ± 4.73 (12.18)
	100% $25°C$	0.49 ± 0.68 (2.27)	0.09 ± 0.04 (−0.19)	0.17 ± 0.14 (0.50)	1.96 ± 1.55 (5.34)	0.50 ± 0.43 (−1.19)	0.67 ± 0.67 (2.23)	0.10 ± 0.05 (−0.20)	0.11 ± 0.06 (−0.21)	0.13 ± 0.07 (−0.27)	0.09 ± 0.09 (0.37)	0.88 ± 0.53 (2.06)	0.59 ± 0.83 (2.56)
	30% $-5°C$	1.02 ± 0.71 (−2.16)	0.04 ± 0.05 (−0.14)	0.27 ± 0.21 (−0.67)	0.21 ± 0.16 (0.55)	0.24 ± 0.24 (1.02)	9.37 ± 8.31 (−25.78)	0.03 ± 0.04 (−0.10)	0.06 ± 0.05 (−0.17)	0.32 ± 0.26 (−0.84)	0.53 ± 0.52 (−1.67)	3.02 ± 2.80 (−8.34)	2.50 ± 1.88 (−5.97)
	30% $+5°C$	0.28 ± 0.36 (1.27)	0.09 ± 0.06 (−0.19)	0.06 ± 0.05 (−0.14)	1.62 ± 1.10 (3.75)	0.15 ± 0.15 (−0.43)	1.31 ± 1.27 (−3.79)	0.08 ± 0.06 (−0.20)	0.13 ± 0.08 (−0.26)	0.23 ± 0.18 (−0.57)	0.15 ± 0.15 (−0.52)	0.58 ± 0.73 (−2.14)	0.75 ± 0.39 (−1.51)
	30% $25°C$	0.64 ± 0.89 (3.01)	0.07 ± 0.04 (−0.17)	0.23 ± 0.17 (0.62)	2.16 ± 1.64 (5.76)	0.33 ± 0.25 (−0.72)	0.33 ± 0.20 (−0.61)	0.07 ± 0.05 (−0.19)	0.09 ± 0.06 (−0.24)	0.18 ± 0.13 (−0.42)	0.05 ± 0.06 (−0.17)	0.19 ± 0.23 (0.68)	0.59 ± 0.80 (2.33)
Michelin	100% $-5°C$	1.79 ± 2.32 (5.72)	0.06 ± 0.07 (0.23)	0.70 ± 0.42 (−1.37)	1.94 ± 1.11 (3.86)	4.89 ± 5.60 (21.18)	18.63 ± 11.29 (−42.20)	0.08 ± 0.09 (0.44)	0.10 ± 0.11 (−0.33)	0.84 ± 0.59 (−2.14)	1.03 ± 0.81 (−2.92)	4.37 ± 2.92 (−8.44)	3.27 ± 1.77 (−5.84)
	100% $+5°C$	6.54 ± 6.56 (17.59)	0.09 ± 0.10 (−0.99)	0.42 ± 0.42 (1.08)	6.16 ± 4.01 (12.02)	7.84 ± 9.67 (36.79)	4.96 ± 3.59 (−15.41)	0.09 ± 0.10 (0.26)	0.17 ± 0.14 (−0.41)	0.45 ± 0.26 (−1.19)	0.32 ± 0.39 (1.06)	0.83 ± 0.99 (2.61)	3.25 ± 3.48 (8.13)
	100% $25°C$	10.73 ± 10.91 (31.99)	0.29 ± 0.12 (−0.60)	1.05 ± 0.90 (2.75)	10.14 ± 7.98 (23.38)	8.26 ± 9.74 (32.52)	1.40 ± 1.55 (2.67)	0.35 ± 0.16 (−0.75)	0.40 ± 0.17 (−0.83)	0.51 ± 0.65 (1.63)	1.80 ± 2.28 (6.52)	4.65 ± 5.05 (14.11)	8.47 ± 9.06 (24.20)
	30% $-5°C$	0.65 ± 0.45 (−1.26)	0.07 ± 0.06 (−0.23)	0.34 ± 0.24 (−0.71)	0.66 ± 0.39 (1.30)	0.22 ± 0.26 (0.89)	7.34 ± 6.81 (−19.58)	0.04 ± 0.05 (−0.17)	0.09 ± 0.07 (−0.26)	0.29 ± 0.22 (−0.63)	0.22 ± 0.22 (−0.71)	1.10 ± 1.16 (−3.45)	1.78 ± 1.41 (−4.08)
	30% $+5°C$	0.21 ± 0.21 (−0.47)	0.06 ± 0.07 (−0.19)	0.07 ± 0.08 (−0.25)	1.37 ± 1.03 (3.30)	0.13 ± 0.17 (0.55)	1.16 ± 1.11 (−3.75)	0.04 ± 0.05 (−0.16)	0.06 ± 0.07 (−0.19)	0.18 ± 0.08 (−0.42)	0.05 ± 0.06 (0.22)	0.39 ± 0.19 (0.74)	0.85 ± 0.47 (−1.65)
	30% $25°C$	0.36 ± 0.48 (2.05)	0.21 ± 0.13 (−0.47)	0.07 ± 0.09 (0.86)	1.50 ± 1.22 (4.70)	0.15 ± 0.14 (0.79)	0.82 ± 0.61 (−1.80)	0.18 ± 0.10 (−0.39)	0.20 ± 0.12 (−0.46)	0.33 ± 0.21 (−0.70)	0.09 ± 0.07 (0.49)	0.23 ± 0.12 (0.99)	0.73 ± 0.39 (−1.28)

TABLE C.3: Leaning with using complete (100%) and partial (30%) NEDC and Michelin discharge experiments at -5, 5 and $25°C$ and croos-validation with all these experiments: mean absolute error ± standard deviation error (maximum error). The model is learned using a MCEM algorithm penalized with identifiability constraints and selected with BIC.

C.3 Pack batteries of type L

Validation \ Learning	CU1	CU2	CU3	CU4	CU5	CU6	CU7	CU8	CU9
Drive 1	0.46 ± 0.39 (1.38)	0.79 ± 0.47 (−1.80)	0.13 ± 0.35 (−1.00)	0.81 ± 1.05 (3.00)	1.10 ± 1.09 (−3.00)	2.22 ± 1.39 (5.00)	0.84 ± 1.06 (−3.00)	0.78 ± 1.04 (−3.00)	0.42 ± 0.62 (−1.00)
Drive 2	2.08 ± 1.11 (4.31)	0.89 ± 0.90 (3.28)	1.77 ± 0.91 (4.00)	1.66 ± 1.73 (5.00)	2.21 ± 1.20 (5.00)	1.28 ± 0.59 (3.00)	1.02 ± 1.13 (−3.00)	0.46 ± 0.76 (−2.00)	1.67 ± 0.55 (3.00)
Drive 3	0.49 ± 0.34 (1.27)	0.61 ± 0.45 (−1.61)	0.26 ± 0.44 (−1.00)	0.96 ± 1.15 (3.00)	0.37 ± 0.60 (−1.00)	0.59 ± 0.82 (−2.00)	0.94 ± 1.27 (−4.00)	1.32 ± 1.26 (−4.00)	0.64 ± 0.88 (−2.00)
Drive 4	0.77 ± 0.40 (1.66)	0.66 ± 0.51 (−1.69)	0.44 ± 0.50 (1.00)	0.92 ± 0.77 (−2.00)	1.09 ± 1.16 (−3.00)	1.20 ± 1.08 (−3.00)	0.64 ± 0.75 (2.00)	1.80 ± 1.27 (−4.00)	0.82 ± 0.43 (2.00)
Drive 5 (morning)	1.91 ± 1.14 (3.64)	0.11 ± 0.13 (0.46)	1.03 ± 0.48 (2.18)	0.57 ± 0.73 (−2.01)	0.74 ± 0.32 (1.40)	0.64 ± 0.83 (−2.44)	1.75 ± 1.91 (−5.74)	0.65 ± 0.75 (−1.82)	1.37 ± 0.55 (2.43)
Drive 5 (evening)	1.47 ± 0.92 (2.93)	0.34 ± 0.36 (1.26)	1.29 ± 0.66 (2.71)	0.50 ± 0.63 (1.76)	0.71 ± 0.34 (1.41)	0.80 ± 0.70 (−1.93)	0.58 ± 0.38 (1.53)	0.77 ± 0.61 (−1.57)	1.60 ± 0.69 (2.82)

TABLE C.4: Learning using complete NEDC discharge experiments performed during the 9 checkups and validation using data collected during the 5 periods of drive: mean absolute error ± standard deviation error (maximum error). The model is learned using a MCEM algorithm penalized with identifiability constraints and selected with BIC.

Bibliographie

Ackerson, G. A. and Fu, K. S. (1970). On state estimation in switching environments. *IEEE Transactions on Automatic Control*, 15(1) :10–17. *Cited on page 51*

Affanni, A., Bellini, A., Concari, C., Franceschini, G., Lorenzani, E., and Tassoni, C. (2003). EV battery state of charge : neural network based estimation. In *IEEE International Conference on Electric Machines and Drives*, volume 2, pages 684–688. *Cited on pages 3, 13, 38, and 41*

Akaike, H. (1998). Information theory and an extension of the maximum likelihood principle. In *Selected Papers of Hirotugu Akaike*, pages 199–213. Springer Series in Statistics. *Cited on pages 7, 100, and 104*

Akashi, H. and Kumamoto, H. (1977). Random sampling approach to state estimation in switching environments. *Automatica*, 13(4) :429 – 434. *Cited on page 51*

Alzieu, J., Smimite, H., and Glaize, C. (1997). Improvement of intelligent battery controller : state-of-charge indicator and associated functions. *Proceedings of the Fifth European Lead Battery Conference*, 67(1-2) :157–161. *Cited on pages 21 and 41*

Anderson, B. D. O. and Moore, J. B. (1979). *Optimal filtering*. Englewood Clifss, N.J. Prentice-Hall. *Cited on page 48*

André, M. (2004). The {ARTEMIS} european driving cycles for measuring car pollutant emissions. *Science of The Total Environment*, 334–335 :73 – 84. *Cited on page 148*

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 72(3) :269–342. *Cited on pages 6, 65, 68, and 70*

Arlot, S. and Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics surveys*, 4 :40–79. *Cited on page 35*

Armenta-Deu, C. and Donaire, T. (1996). Determination of an ageing factor for lead/acid batteries. 1. Kinetic aspects. *Journal of Power Sources*, 58(2) :123 – 133. *Cited on page 32*

Avvari, G. V., Balasingam, B., Pattipati, K. R., and Bar-Shalom, Y. (2015). A battery chemistry-adaptive fuel gauge using probabilistic data association. *Journal of Power Sources*, 273 :185 – 195. *Cited on page 41*

Barré, A., Deguilhem, B., Grolleau, S., Gerard, M., Suard, F., and Riu, D. (2013). A review on lithium-ion battery ageing mechanisms and estimations for automotive applications. *Journal of Power Sources*, 241 :680 – 689. *Cited on pages 32 and 148*

Baudry, J.-P. and Celeux, G. (2015). EM for mixtures - Initialization requires special care. *Statistics and Computing*. (to appear, https ://hal.inria.fr/hal-01113242). *Cited on pages 81 and 92*

Baudry, J.-P., Maugis, C., and Michel, B. (2012). Slope heuristics : overview and implementation. *Statistics and Computing*, 22(2) :455–470. *Cited on page 108*

Birgé, L. and Massart, P. (2007). Minimal penalties for gaussian model selection. *Probability Theory and Related Fields*, 138(1-2) :33–73. *Cited on pages 7, 100, 104, 106, and 107*

Biscarat, J. C. (1994). Almost sure convergence of a class of stochastic algorithms. *Stochastic Processes and their Applications*, 50(1) :83 – 99. *Cited on page 82*

Blom, H. A. P. and Bar-Shalom, Y. (1988). The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33(8) :780–783. *Cited on page 57*

Bo, C., Zhifeng, B., and Binggang, C. (2008). State of charge estimation based on evolutionary neural network. *Energy Conversion and Management*, 49(10) :2788–2794. *Cited on pages 3, 13, 38, and 41*

Bond, T. M., Burns, J. C., Stevens, D. A., Dahn, H. M., and Dahn, J. R. (2013). Improving precision and accuracy in Coulombic efficiency measurements of Li-ion batteries. *Journal of The Electrochemical Society*, 160(3) :A521–A527. *Cited on page 21*

Broussely, M., Herreyre, S., Biensan, P., Kasztejna, P., Nechev, K., and Staniewicz, R. J. (2001). Aging mechanism in Li-ion cells and calendar life predictions. *Journal of Power Sources*, 97-98 :13 – 21. *Cited on page 20*

Buller, S., Thele, M., De Doncker, R. W. A. A., and Karden, E. (2005). Impedance-based simulation models of supercapacitors and Li-ion batteries for power electronic applications. *IEEE Transactions on Industry Applications,*, 41(3) :742–747. *Cited on page 24*

Burnham, K.-P. and Anderson, D.-R. (2002). *Model selection and multimodel inference : a practical information-theoretic approach*. Springer-Verlag New York. *Cited on pages 104, 105, and 106*

Cappé, O., Moulines, E., and Rydén, T. (2005). *Inference in hidden Markov models*. Springer Series in Statistics. Springer-Verlag New York. *Cited on pages 44, 70, and 73*

Carter, C. K. and Kohn, R. (1994). On Gibbs sampling for state space models. *Biometrika*, 81(3) :541–553. *Cited on pages 6, 64, and 67*

Carter, C. K. and Kohn, R. (1996). Markov Chain Monte Carlo in conditionally Gaussian state space models. *Biometrika*, 83(3) :589–601. *Cited on pages 6, 64, 68, and 70*

Casella, G. and Berger, R. L. (2002). *Statistical Inference*. Duxbury advanced series in statistics and decision sciences. Thomson Learning. *Cited on pages 80 and 88*

Celeux, G., Chauveau, D., and Diebolt, J. (1996). Stochastic versions of the EM algorithm : an experimental study in the mixture case. *Journal of Statistical Computation and Simulation*, 55(4) :287–314. *Cited on page 82*

Celeux, G. and Durand, J.-B. (2008). Selecting hidden Markov model state number with cross-validated likelihood. *Computational Statistics*, 23(4) :541–564. *Cited on pages 7, 100, 101, 104, and 108*

Chang, W.-Y. (2013). The state of charge estimating methods for battery : A review. *ISRN Applied Mathematics*, (ID 953792, 7 pages). doi :10.1155/2013/953792. *Cited on page 17*

Charkhgard, M. and Farrokhi, M. (2010). State-of-charge estimation for lithium-ion batteries using neural networks and EKF. *IEEE Transactions on Industrial Electronics*, 57(12) :4178–4187. *Cited on pages 13, 37, and 41*

Chen, Q. and Lin, C. (2005). Summarization of studies on performance models of batteries for electric vehicle. *Automobile Technology*, 3(3) :1–5. *Cited on page 27*

Chen, X., Shen, W., Cao, Z., and Kapoor, A. (2012). A comparative study of observer design techniques for state of charge estimation in electric vehicles. In *7th IEEE Conference on Industrial Electronics and Applications (ICIEA'12)*, pages 102–107. *Cited on pages 29 and 41*

Chib, S. (1996). Calculating posterior distributions and modal estimates in Markov mixture models. *Journal of Econometrics*, 75(1) :79 – 97. *Cited on pages 6, 64, and 67*

Christianson, C. C. and Bourke, R. F. (1976). Battery state of charge gauge. US Patent 3946299. *Cited on page 23*

Chubb, M. F. and Harner, H. R. (1935). The effect of temperature and rate of discharge on the capacity of Lead-Acid storage batteries. *Transactions of The Electrochemical Society*, 68(1) :251–259. *Cited on page 22*

Codeca, F., Savaresi, Sergio, M., and Rizzoni, G. (2008). On battery state of charge estimation : A new mixed algorithm. In *IEEE International Conference on Control Applications (CCA'8)*, pages 102 –107. *Cited on pages 2, 13, 14, 26, 30, and 41*

Dai, H., Wei, X., and Sun, Z. (2006). Online SOC estimation of high-power lithium-ion batteries used on HEVs. In *IEEE International Conference on Vehicular Electronics and Safety (ICVES'6)*, pages 342 –347. *Cited on pages 13, 14, 21, 27, 30, 33, and 41*

De Jong, P. and Shephard, N. (1995). The simulation smoother for time series models. *Biometrika*, 82(2) :339–350. *Cited on pages 6, 64, and 67*

Delaille, A., Perrin, M., Huet, F., and Hernout, L. (2006). Study of the coup de fouet of lead-acid cells as a function of their state-of-charge and state-of-health. *Journal of Power Sources*, 158(2) :1019–1028. *Cited on page 24*

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1) :1–38. *Cited on pages 7, 79, 83, 84, and 85*

Deng, Z., Zhang, Z., Lai, Y., Liu, J., Li, J., and Liu, Y. (2013). Electrochemical Impedance Spectroscopy study of a Lithium/Sulfur battery : modeling and analysis of capacity fading. *Journal of The Electrochemical Society*, 160(4) :A553–A558. *Cited on page 24*

Di Domenico, D., Fiengo, G., and Stefanopoulou, A. (2008). Lithium-ion battery state of charge estimation with a Kalman filter based on a electrochemical model. In *IEEE International Conference on Control Applications (CCA'08)*, pages 702–707. *Cited on pages 2, 13, 32, and 41*

Diard, J.-P., Le Gorrec, B., Montella, C., and Landaud, P. (1997). Constant load vs constant current EIS study of electrochemical battery discharge. *Electrochimica Acta*, 42(23–24) :3417 – 3420. *Cited on page 24*

Dong, T. K., Kirchev, A., Mattera, F., Kowal, J., and Bultel, Y. (2011). Dynamic modeling of Li-ion batteries using an equivalent electrical circuit. *Journal of The Electrochemical Society*, 158(3) :A326–A336. *Cited on page 30*

Doucet, A., De Freitas, N., and Gordon, N. (2001a). *Sequential Monte Carlo methods in practice*. Information Science and Statistics. Springer-Verlag New York. *Cited on page 58*

Doucet, A., Gordon, N. J., and Krishnamurthy, V. (2001b). Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing,*, 49(3) :613–624. *Cited on pages 46, 58, and 59*

Doucet, A. and Johansen, A. M. (2009). A tutorial on particle filtering and smoothing : Fifteen years later. *Handbook of Nonlinear Filtering*, 12 :656–704. *Cited on page 60*

Durbin, J. and Koopman, S. J. (2002). A simple and efficient simulation smoother for state space time series analysis. *Biometrika*, 89(3) :603–615. *Cited on pages 6, 64, and 67*

Feng, X. and Sun, Z. (2008). A battery model including hysteresis for state-of-charge estimation in Ni-MH battery. In *IEEE Conference on Vehicle Power and Propulsion*, pages 1–5. *Cited on pages 13, 28, and 41*

Frühwirth-Schnatter, S. (2011). *Dealing with Label Switching under Model Uncertainty*. Mixtures estimation and applications. John Wiley & Sons, Ltd. *Cited on page 75*

Frühwirth-Schnatter, S. (1994). Data augmentation and dynamic linear models. *Journal of Time Series Analysis*, 15(2) :183–202. *Cited on pages 6, 64, and 67*

Frühwirth-Schnatter, S. (2006). *Finite Mixture and Markov Switching Models*. Springer Series in Statistics. Springer-Verlag New York. *Cited on pages 51 and 70*

Gaddam, V. R., Wang, X. Q., Arey, S., Yang, Z. J., and Singh, P. (2000). Fuzzy Logic-Based Internal and External State Of Charge Meters for LI/SO2 Cells. *39th Power Sources Conference*. *Cited on pages 22 and 41*

Gama, J. and Castillo, G. (2006). Learning with local drift detection. In *Advanced Data Mining and Applications*, Lecture Notes in Computer Science, pages 42–55. Springer Berlin Heidelberg. *Cited on page 163*

Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410) :398–409. *Cited on pages 63 and 67*

Gerlach, R., Carter, C., and Kohn, R. (2000). Efficient Bayesian inference for dynamic mixture models. *Journal of the American Statistical Association*, 95(451) :819–828. *Cited on page 68*

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2) :107–113. *Cited on page 60*

Green, P. J. (1990). On use of the EM for penalized likelihood estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, 52(3) :443–452. *Cited on pages 7, 80, 81, 83, 88, 89, and 90*

Guo, M., Sikha, G., and White, R. E. (2011). Single-particle model for a lithium-ion cell : Thermal behavior. *Journal of The Electrochemical Society*, 158(2) :A122–A132. *Cited on page 31*

Hammersley, J. M. and Handscomb, D. C. (1964). *Monte Carlo Methods*. Monographs on Applied Probability and Statistics. Springer Netherlands. *Cited on page 58*

Han, J., Kim, D., and Sunwoo, M. (2009). State-of-charge estimation of lead-acid batteries using an adaptive extended Kalman filter. *Journal of Power Sources*, 188(2) :606 – 612. *Cited on pages 28 and 41*

Hansen, T. and Wang, C.-J. (2005). Support vector based battery state of charge estimator. *Journal of Power Sources*, 141(2) :351–358. *Cited on pages 3, 13, 36, 39, and 41*

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc. *Cited on page 35*

Haykin, S. (1998). *Neural Networks : A Comprehensive Foundation*. Prentice Hall PTR, 2nd edition. *Cited on page 37*

Haykin, S. (2001). *Kalman filtering and neural networks*, volume 47. John Wiley & Sons, Inc. *Cited on page 26*

He, H., Xiong, R., and Fan, J. (2011). Evaluation of lithium-ion battery equivalent circuit models for state of charge estimation by an experimental approach. *Energies*, 4(4) :582–598. *Cited on page 41*

He, W., Williard, N., Chen, C., and Pecht, M. (2014). State of charge estimation for Li-ion batteries using neural network modeling and unscented Kalman filter-based error cancellation. *International Journal of Electrical Power & Energy Systems*, 62 :783 – 791. *Cited on page 41*

Hirai, T., Ohnishi, A., Nagaoka, N., Mori, N., Ametani, A., and Umeda, S. (2008). Automatic equivalent-circuit estimation system for lithium-ion battery. In *43rd International Universities Power Engineering Conference (UPEC'08)*, pages 1 –5. *Cited on pages 2, 3, 13, 14, 29, 30, 31, 32, 39, and 41*

Hu, X., Sun, F., and Zou, Y. (2010). Estimation of state of charge of a lithium-ion battery pack for electric vehicles using an adaptive Luenberger observer. *Energies*, 4(12) :1586–1603. *Cited on pages 14, 29, 30, 31, 39, and 41*

Huet, F. (1998). A review of impedance measurements for determination of the state-of-charge or state-of-health of secondary batteries. *Journal of Power Sources*, 70(1) :59–69. *Cited on pages 2, 24, and 41*

Iliev, V. and Pavlov, D. (1982). Self-Discharge and Passivation Phenomena in Lead-Acid Batteries during Storage. *Journal of The Electrochemical Society*, 129(3) :458–464. *Cited on page 20*

Kalawoun, J., Biletska, K., Suard, F., and Montaru, M. (2015a). From a novel classification of the battery state of charge estimators toward a conception of an ideal one. *Journal of Power Sources*, 279 :694–706. *Cited on page 11*

Kalawoun, J., Pamphile, P., and Celeux, G. (2015b). Identifiability of a Switching Markov State-Space model. *Gretsi 2015*. *Cited on page 56*

Kalman, R. (1963). Mathematical description of linear dynamical systems. *Journal of the Society for Industrial and Applied Mathematics Series A Control*, 1(2) :152–192. *Cited on pages 43 and 54*

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME-Journal of Basic Engineering*, 82(Series D) :35–45. *Cited on page 26*

Kim, C.-J. (1994). Dynamic linear models with Markov-switching. *Journal of Econometrics*, 60(1–2) :1–22. *Cited on page 51*

Kim, J., Lee, S., and Cho, B. H. (2011). Discrimination of Li-ion batteries based on Hamming network using discharging-charging voltage pattern recognition for improved state-of-charge estimation. *Journal of Power Sources*, 196(4) :2227–2240. *Cited on pages 38 and 41*

Kong, A., Liu, J. S., and Wong, W. H. (1994). Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89(425) :278–288. *Cited on page 60*

Kozlowski, J. D. (2003). Electrochemical cell prognostics using online impedance measurements and model-based data fusion techniques. *IEEE Aerospace Conference*, 7 :3257–3270. *Cited on pages 13, 34, 38, and 41*

Kurzweil, P. (2009). BATTERIES — Nomenclature. In *Encyclopedia of Electrochemical Power Sources*, pages 381–394. Elsevier. *Cited on page 23*

Kutluay, K., Çadirci, Y., Özkazanç, Y. S., and Çadirci, I. (2005). A new online state-of-charge estimation and monitoring system for sealed lead-acid batteries in telecommunication power supplies. *IEEE Transactions on Industrial Electronics*, 52(5) :1315–1327. *Cited on page 41*

Lazarescu, M. M., Venkatesh, S., and Bui, H. H. (2004). Using multiple windows to track concept drift. *Intelligent Data Analysis*, 8(1) :29–59. *Cited on page 163*

Lebarbier, E. and Mary-Huard, T. (2004). Le critère BIC : fondements théoriques et interprétation. Research Report RR-5315, INRIA. *Cited on page 105*

Lee, S., Kim, J., Lee, J., and Cho, B. H. (2008). State-of-charge and capacity estimation of lithium-ion battery using a new open-circuit voltage versus state-of-charge. *Journal of Power Sources*, 185(2) :1367–1373. *Cited on pages 14, 30, 32, and 41*

Leroux, B.-G. (1992). Maximum-likelihood estimation for hidden Markov models. *Stochastic Processes and their Applications*, 40(1) :127 – 143. *Cited on page 56*

Li, J., Barillas, J. K., Guenther, C., and Danzer, M. A. (2013). A comparative study of state of charge estimation algorithms for LiFePO4 batteries used in electric vehicles. *Journal of Power Sources*, 230 :244 – 250. *Cited on pages 29 and 41*

Liao, Y., Huang, J. H., and Zeng, Q. (2011). A novel method for estimating state of charge of lithium-ion battery packs. *Advanced Materials Research*, 152-153 :428–435. *Cited on pages 21 and 41*

Linden, D. (1947). Capacity as a Function of Temperature and Discharge Rate of Certain Military Types of Lead-Acid Batteries. *Transactions of The Electrochemical Society*, 92(1) :223–228. *Cited on page 22*

Lu, L., Han, X., Li, J., Hua, J., and Ouyang, M. (2013). A review on the key issues for lithium-ion battery management in electric vehicles. *Journal of Power Sources*, 226 :272 – 288. *Cited on page 17*

Álvarez Antóna, J., García Nietob, P., De Cos Juezc, F., F., S. L., González Vegaa, M., , and Roqueñí Gutiérrezc, M. (2013). Battery state-of-charge estimator using the SVM technique. *Applied Mathematical Modelling*, 37(9) :6244–6253. *Cited on pages 36 and 41*

Malkhandi, S. (2006). Fuzzy logic-based learning system and estimation of state-of-charge of lead-acid battery. *Engineering Applications of Artificial Intelligence*, 19(5) :479 – 485. *Cited on pages 21 and 41*

McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc. *Cited on pages 105, 142, 154, and 160*

Montaru, M. (2009). *Contribution à l'evaluation du vieillissement des batteries de puissance utilisées dans les véhicules hybrides*. PhD thesis, Institut Polytechnique de Grenoble. *Cited on page 125*

Moo, C. S., Ng, K. S., Chen, Y. P., and Hsieh, Y. C. (2007). State-of-charge estimation with open-circuit-voltage for lead-acid batteries. In *Power Conversion Conference-Nagoya (PCC'07)*, pages 758–762. *Cited on pages 2, 13, 29, 32, and 41*

Moss, P. L., Au, G., Plichta, E. J., and Zheng, J. P. (2008). An electrical circuit for modeling the dynamic response of Li-ion polymer batteries. *Journal of The Electrochemical Society*, 155(12) :A986–A994. *Cited on page 30*

Newman, J. and Tiedemann, W. (1975). Porous-electrode theory with battery applications. *AIChE Journal*, 21(1) :25–41. *Cited on page 31*

Ng, K. S., Moo, C.-S., Chen, Y.-P., and Hsieh, Y.-C. (2009). Enhanced Coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries. *Applied energy*, 86(9) :1506–1511. *Cited on pages 2, 20, 21, and 41*

Pang, S., Farrell, J., Du, J., and Barth, M. (2001). Battery state of charge estmation. *American Control Conference*, 2 :1644–1649. *Cited on pages 2, 3, 14, 29, 30, 31, 32, 39, 41, and 162*

Parfitt, C. E., Crofts, W. E., and Buckle, R. (2010). An adaptable lithium-ion battery model. *ECS Transactions*, 28(22) :21–33. *Cited on pages 23 and 30*

Pavlov, D. and Petkova, G. (2002). Phenomena that limit the capacity of the positive lead-acid battery plates : II. Electrochemical Impedance Spectroscopy and Mechanism of Discharge of the Plate. *Journal of The Electrochemical Society*, 149(5) :A654–A661. *Cited on page 30*

Peled, E.and Yamin, H., Reshef, I., Kelrich, D., and Rozen, S. (1988). Method and apparatus for determining the state-of-charge of batteries particularly lithium batteries. US 4725784. *Cited on pages 2, 23, and 41*

Petzl, M. and Danzer, M. A. (2013). Advancements in OCV Measurement and Analysis for Lithium-Ion Batteries. *IEEE Transactions on Energy Conversion,*, 28(3) :675–681. *Cited on page 23*

Piller, S., Perrin, M., and Jossen, A. (2001). Methods for state-of-charge determination and their application. *Journal of Power Sources*, 96(1) :113–120. *Cited on pages 17, 24, and 41*

Plett, L. G. (2004). Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs. *Journal of Power Sources*, 134 :262–276. *Cited on pages 2, 3, 13, 14, 27, 28, 30, 31, 33, 37, 39, 41, and 49*

Pop, V., Bergveld, H. J., Notten, P. H. L., and Regtien, P. P. L. (2005). State-of-the-art of battery state-of-charge determination. *Measurement Science and Technology*, 16(12) :R93–R110. *Cited on page 17*

Pop, V., Bergveld, H. J., Op het Veld, J. H. G., and Regtien, P. P. L. (2006). Modeling battery behavior for accurate state-of-charge indication. *Journal of the Electrochemical Society*, 153(11) :A2013–A2022. *Cited on pages 2, 13, 23, 31, and 41*

Pop, V., Bergveld, H. J., Regtien, P. P. L., Op het Veld, J. H. G., Danilov, D., and Notten, P. H. L. (2007). Battery aging and its influence on the electromotive force. *Journal of The Electrochemical Society*, 154(8) :A744–A750. *Cited on page 23*

Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2) :257–286. *Cited on pages 7, 57, and 79*

Ripley, B. D. (1996). *Pattern recognition and neural networks*. Cambridge university press. *Cited on pages 100, 101, and 108*

Robert, C. P. and Casella, G. (2004). *Monte Carlo Sattisticals Methods*. Springer Series in Statistics. Springer-Verlag New York. *Cited on pages 63 and 68*

Rodrigues, S., Munichandraiah, N., and Shukla, A. K. (2000). A review of state-of-charge indication of batteries by means of a.c. impedance measurements. *Journal of Power Sources*, 87(1-2) :12 – 20. *Cited on pages 23 and 24*

Roscher, M. A. and Sauer, D. U. (2011). Dynamic electric behavior and open-circuit-voltage modeling of LiFePO4-based lithium ion secondary batteries. *Journal of Power Sources*, 196(1) :331 – 336. *Cited on page 23*

Salkind, A. J., Fennie, C., Singh, P., Atwater, T., and Reisner, D. E. (1999). Determination of state-of-charge and state-of-health of batteries by fuzzy logic methodology. *Journal of Power Sources*, 80(1-2) :293–300. *Cited on pages 24 and 41*

Santhanagopalan, S. and White, R. E. (2006). Online estimation of the state of charge of a lithium ion cell. *Journal of Power Sources*, 161(2) :1346–1355. *Cited on pages 32 and 41*

Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2) :461–464. *Cited on pages 7, 100, 104, and 105*

Shen, Y. (2010). Adaptive online state-of-charge determination based on neuro-controller and neural network. *Energy Conversion and Management*, 51(5) :1093–1098. *Cited on pages 3, 13, 37, 39, and 41*

Shumway, R. H. and Stoffer, D. S. (1982). An approach to time series smoothing and forecasting using the EM algorithm. *Journal of Time Series Analysis*, 3(4) :253–264. *Cited on pages 7, 68, 79, 85, and 165*

Smith, A.-J., Burns, J.-C., Trussler, S., and Dahn, J.-R. (2010). Precision measurements of the Coulombic efficiency of lithium-ion batteries and of electrode materials for lithium-ion batteries. *Journal of The Electrochemical Society*, 157(2) :A196–A202. *Cited on page 21*

Smyth, P. (2000). Model selection for probabilistic clustering using cross-validated likelihood. *Statistics and Computing*, 10(1) :63–72. *Cited on page 104*

Sobhani, P. and Beigy, H. (2011). New drift detection method for data streams. In *Adaptive and Intelligent Systems*, Lecture Notes in Computer Science, pages 88–97. Springer Berlin Heidelberg. *Cited on page 163*

Srinivasan, V., Weidner, J. W., and Newman, J. S. (2001). Hysteresis during cycling of nickel hydroxide active material. *Journal of the Electrochemical Society*, 148(9) :A969–A980. *Cited on page 23*

Stephens, M. (2000). Dealing with label switching in mixture models. *Journal of the Royal Statistical Society, Series B*, 62(4) :795–809. *Cited on page 55*

Tang, X., Zhang, X., Koch, B., and Frisch, D. (2008). Modeling and estimation of nickel metal hydride battery hysteresis for SOC estimation. In *Proceedings of International conference on prognostics and health management*, pages 1–12. *Cited on pages 29 and 41*

Tanner, M. A. (1996). *Tools for Statistical Inference : Methods for the Exploration of Posterior Distributions and Likelihood Functions*. Springer Series in Statistics. Springer-Verlag New York. *Cited on pages 7, 83, and 86*

Tipping, M.-E. (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1 :211–244. *Cited on page 35*

Tugnait, J. K. (1982). Adaptive estimation and identification for discrete systems with Markov jump parameters. *IEEE Transactions on Automatic Control*, 27(5) :1054–1065. *Cited on pages 6, 7, 45, 51, 79, and 83*

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Information Science and Statistics. Springer-Verlag New York. *Cited on pages 35 and 36*

Vetter, J., Novák, P., Wagner, M. R., Veit, C., Möller, K.-C., Besenhard, J. O., Winter, M., Wohlfahrt-Mehrens, M., Vogler, C., and Hammouche, A. (2005). Ageing mechanisms in lithium-ion batteries. *Journal of Power Sources*, 147(1 - 2) :269 – 281. *Cited on pages 20 and 32*

Waag, W., Fleischer, C., and Sauer, D.-U. (2014). Critical review of the methods for monitoring of lithium-ion batteries in electric and hybrid vehicles. *Journal of Power Sources*, 258 :321–339. *Cited on page 17*

Walter, E. and Lecourtier, Y. (1981). Unidentifiable compartmental models : what to do ? *Mathematical Biosciences*, 56(1–2) :1 – 25. *Cited on pages 5 and 54*

Wang, J., Cao, B., Chen, Q., and Wang, F. (2007). Combined state of charge estimator for electric vehicle battery pack. *Control Engineering Practice*, 12 :1569–1576. *Cited on pages 2, 20, 21, and 41*

Wei, G. C. G. and Tanner, M. A. (1990). A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(411) :699–704. *Cited on pages 7, 80, 83, and 86*

Whiteley, N., Andrieu, C., and Doucet, A. (2010). Efficient Bayesian inference for switching state-space models using discrete particle Markov chain Monte Carlo methods. *arXiv preprint arXiv :1011.2437*. *Cited on pages 65, 68, and 70*

Willihnganz, E. (1941). A bridge for measuring storage battery resistance. *Transactions of The Electrochemical Society*, 79(1) :253–258. *Cited on page 23*

Wu, B. and White, R. E. (2000). Self-Discharge Model of a Nickel-Hydrogen Cell. *Journal of The Electrochemical Society*, 147(3) :902–909. *Cited on page 20*

Yakowitz, S. J. and Spragins, J. D. (1968). On the identifiability of finite mixtures. *The Annals of Mathematical Statistics*, 39(1) :209–214. *Cited on page 56*

Zachlin, A. C. (1942). The effect of Temperature on the Rate of Self Discharge of Lead-Acid Storage Batteries. *Transactions of The Electrochemical Society*, 82(1) :365–373. *Cited on page 20*

Zhang, J. and Lee, J. (2011). A review on prognostics and health monitoring of Li-ion battery. *Journal of Power Sources*, 196(15) :6007–6014. *Cited on page 17*

Zheng, Y., Ouyang, M., Lu, L., Li, J., Han, X., Xu, L., Ma, H., Dollmeyer, T. A., and Freyermuth, V. (2013). Cell state-of-charge inconsistency estimation for LiFePO4 battery pack in hybrid electric vehicles using mean-difference model. *Applied Energy*, 111 :571–580. *Cited on pages 12 and 39*

université PARIS-SACLAY | École doctorale de mathématiques Hadamard (EDMH)

Titre : Modélisation statistique de l'état de charge des batteries électriques

Mots clés : Apprentissage statistique, État de charge d'une batterie électrique, Modèle à espaces d'états gouverné par une chaîne de Markov, Filtrage particulaire, Algorithme EM, Sélection de modèle

Résumé : Les batteries électriques sont omniprésentes dans notre vie quotidienne : ordinateur, téléphone, etc. Elles jouent un rôle important dans le défi de la transition énergétique : anticiper la raréfaction des énergies fossiles et réduire la pollution, en développant le stockage des énergies renouvelables et les transports électriques. Cependant, l'estimation de l'état de charge (State of Charge – SoC) d'une batterie est difficile et les modèles de prédiction actuels sont peu robustes. En effet, une batterie est un système électrochimique complexe, dont la dynamique est influencée non seulement par ses caractéristiques internes, mais aussi par les conditions d'usages souvent non contrôlables : température, profil d'utilisation, etc. Or, une estimation précise du SoC permet de garantir une utilisation sûre de la batterie en évitant une surcharge ou sous-décharge ; mais aussi d'estimer son autonomie. Dans cette thèse, nous utilisons un modèle à espaces d'états gouverné par une chaîne de Markov cachée. Ce modèle est fondé sur des équations physiques et la chaîne de Markov cachée permet d'appréhender les différents « régimes de fonctionnement » de la batterie. Pour garantir l'unicité des paramètres du modèle, nous démontrons son identifiabilité à partir de contraintes simples et naturelles sur ses paramètres «physiques ». L'estimation du SoC dans un véhicule électrique doit être faîte en ligne et avec une puissance de calcul limitée. Nous estimons donc le SoC en utilisant une technique d'échantillonnage préférentiel séquentiel. D'autre part l'estimation des paramètres est faîte à partir d'une base d'apprentissage pour laquelle les états de la chaîne de Markov et le SoC ne sont pas observés. Nous développons et testons trois algorithmes adaptés à notre modèle à structure latente : un échantillonneur particulaire de Gibbs, un algorithme de Monte-Carlo EM pénalisé par des contraintes d'identifiabilité et un algorithme de Monte-Carlo EM pénalisé par une loi a priori. Par ailleurs les états cachés de la chaîne de Markov visent à modéliser les différents régimes du fonctionnement de la batterie. Nous identifions leur nombre par divers critères de sélection de modèles. Enfin, à partir de données issues de trois types de batteries (cellule, module et pack d'un véhicule électrique), notre modèle a permis d'appréhender les différentes sollicitations de la batterie et donne des estimations robustes et précises du SoC.

Title: Statistical modeling of the state of charge of electric batteries

Keywords: Statistical learning, State of charge of an electric battery, Switching Markov State Space Model, Particle filter, EM algorithm, Model selection

Abstract: Electric batteries are omnipresent in our daily lives: computers, smartphones, etc. Batteries are important for anticipating the scarcity of fossil fuels and tackling their environmental impact. Therefore, estimating the State of Charge (SoC) of a battery is nowadays a challenging issue, as existing physical and statistical models are not yet robust. Indeed, a battery is a complex electrochemical system. Its dynamic depends not only on its internal characteristics but also on uncontrolled usage conditions: temperature, usage profile, etc. However, the SoC estimation helps to prevent overcharge and deep discharge, and to estimate the battery autonomy. In this thesis, the battery dynamics are described by a set of physical linear equations, switching randomly according to a Markov chain. This model is referred to as switching Markov state space model. To ensure the unicity of the model parameters, we prove its identifiability by applying straightforward and natural constraints on its physical parameters. Embedded applications, like electric vehicles, impose online estimated with hardware and time constraints. Therefore, we estimate the SoC using a sequential importance sampling technique. Furthermore, the model includes two latent variables: the SoC and the Markov chain state. Thus, to estimate the parameters, we develop and test three algorithms adapted to latent structure models: particle Gibbs sampler, Monte Carlo EM penalized with identifiability constraints, and Monte Carlo EM penalized with a prior distribution. The hidden Markov states aim to model the different "regimes" of the battery dynamics. We identify their number using different model selection criteria. Finally, when applied to various data from three battery types (cell, module and pack of an electric vehicle) our model allows us to analyze the battery dynamics and to obtain a robust and accurate SoC estimation under uncontrolled usage conditions.