

# 博士論文

## リチウムイオン蓄電池の高精度等価回路モデル およびそれを用いた最適電池制御システムに関する研究 (Research for highly accurate equivalent circuit model of lithium-ion storage battery and its application to optimal battery management system)

2017年3月

立命館大学大学院理工学研究科  
電子システム専攻博士課程後期課程

LIN Lei



立命館大学審査博士論文

リチウムイオン蓄電池の高精度等価回路モデル

およびそれを用いた最適電池制御システムに関する研究

(Research for highly accurate equivalent circuit model of lithium-ion storage  
battery and its application to optimal battery management system)

2017年3月

March 2017

立命館大学大学院理工学研究科

電子システム専攻博士課程後期課程

Doctoral Program in Electronic and Computer systems

Graduate School of Science and Engineering

Ritsumeikan University

LIN Lei

リン ライ

研究指導教員：福井 正博 教授

Supervisor : Professor FUKUI Masahiro



# 概要

近年、再生可能エネルギーの普及に伴い、エネルギーの貯蔵技術が重要な課題となっている。その中でも、リチウムイオン蓄電池は電気エネルギーを高密度に貯蔵できるデバイスとして注目されている。蓄電池管理において、残量予測、劣化診断、電池の熱管理は重要な3つの課題である。

リチウムイオン蓄電池は比較的劣化が少ない蓄電池とされるが、高価であるため、劣化を抑えた使用が望まれる。蓄電池劣化には様々な要因があるが、温度上昇による劣化は蓄電池劣化の主要因である。そのため、蓄電池の温度上昇を抑える技術が重要である。そのため、蓄電池の熱回路モデルを用いた蓄電池の熱挙動を解析する。熱回路モデルは規模が大きいので、並列計算による高速化が求められる。そのために、GPGPUを用いた回路の効率的な並列計算について述べ、熱回路モデルへの展開手法を明らかにした。高精度な後退オイラー法を解くための疎行列の効率的なデータ構造を提案し、従来の疎行列表現方法に対して約63%の計算量で実現することを示した。また、クロック周波数3GHzのCPU(E8400)に比較してコア周波数1.54GHz 並列数512のGPU(GTX580)で、最大74倍の計算の高速化を実現した。これは、GPU-CPU間通信ロスなどを除外し、すべてのプログラムがコア数の数だけ並列度があるとした場合の理論的な高速化上界に比べ、約50%の性能である。通信ロスが約30%であること、データ転送や条件分岐等の並列化できない部分が含まれることを考えると革新的な結果である。

蓄電池は内部抵抗の劣化に伴い、起電力や内部抵抗などの内部パラメータが変化する。そのため、蓄電池の内部パラメータをリアルタイムに把握するのは重要な課題である。本研究では蓄電池のOCVと電極表面抵抗の同定手法を解明した。SOCとOCVの関係に着目し、端子電圧と端子電流の比率に基づいて最適な忘却係数を算出する手法を解明した。従来の固定忘却係数手法に比べ、SOCの最大誤差を半分以下に抑えることができ、有効性を示した。また、温度や電流条件の変動に対しても良好な推定結果を示すことを実験により確認した。

リチウムイオン蓄電池の状態管理において、残量予測は最も重要かつ基本的な技術である。蓄電池残量は、蓄電池充電率(SOC: State of Charge)といい、負極側にリチウムイオンが集中する度合いを数値化する重要な指標である。本研究では、拡張カルマンフィルタを用いた高精度残量予測手法を提案し、実装と評価を行った。また、カルマンフィルタが用いる観測方程式とノイズ分散を自動的に推定する手法を明らかにした。従来技術では、最適なノイズ分散を計算するためには総当たりの膨大な計算機実験(約1000回程程度のカルマンフィルタ処理)が必要であったが、それを自動算出することにより計算量を大幅に削減した。また、評価の結果、従来の総当たり手法以上のSOC推定精度の向上が得られた。これは処理途中にノイズを可変的に最適化する効果にもよるものであり、革新性の高いものである。









## 関連発表論文

### 論文誌等採録論文

- [1] L. Lin, H. Ono, M. Fukui, and K. Takaba, "An in situ full charge capacity estimation algorithm for Li-ion batteries using recursive least-squares identification with adaptive forgetting factor tuning," *ECS Transaction*, vol.75, no.20, pp.111-119, Jan. 2017.
- [2] M. Fukui, Y. Hayakawa, and L. Lin, "A Li-ion battery pack level degradation simulator with consideration of thermal and electrical conditions," *ECS Transaction*, vol.75, no.20, pp.103-110, Jan. 2017.
- [3] 林磊, 石崎龍, 鷹羽浄嗣, 福井正博, "適応的忘却係数調整によるリチウムイオン蓄電池の逐次最小 2 乗同定," 電子情報通信学会和文論文誌 B 第 99 巻, 第 7 号, pp.481-489, 2016 年 7 月.
- [4] 大矢将輝, 鷹羽浄嗣, 林磊, 石崎龍, 河原林直紀, 福井正博, "パラメータ依存状態空間モデルに基づく蓄電池の SoC 推定," システム制御情報学会論文誌, vol.29, no.10, pp.433-440, 2016 年 5 月.
- [5] R. Ishizaki, L. Lin, and M. Fukui, "An accurate SOC estimator for Lithium-ion batteries which considers thermal variation," *Journal of Electrochemistry*, vol. 83, no. 10, pp.852-854, Oct. 2015.
- [6] 林磊, 塩野隼人, 福井正博, "GPGPU による電源回路シミュレーションに対する後退オイラー法の実装と評価," 電子情報通信学会和文論文誌 A 第 96 巻, 第 5 号, pp.215-224, 2013 年 5 月.

### 国際会議における発表

#### 査読付き

- [1] H. Ono, L. Lin, M. Fukui, and K. Takaba, "A precise FCC estimation algorithm based on recursive least squares identification of Li-ion batteries with adaptive forgetting factor tuning," in *Proc. PRiME 2016/230th ECS Meeting*, A03-0358, Oct. 2016. (Honolulu, USA)
- [2] Y. Hayakawa, L. Lin, and M. Fukui, "A practical degradation simulator for assembled Li-ion batteries with calibration functions," in *Proc. PRiME 2016/230th ECS Meeting*, A03-0357, Oct. 2016. (Honolulu, USA)
- [3] L. Lin, M. Fukui, and K. Takaba, "An accurate SOC estimation system for Lithium-ion batteries by EKF with dynamic noise adjustment, .inproc. *International Symposium on Communications and Information Technologies*," pp.33-36, Oct. 2015. (Nara, Japan)

- [4] L. Lin, N. Kawarabayashi, M. Fukui, and I. Shirakawa, "Live Demonstration: A battery smart sensor for smart grid," *in proc. The 12th Asia Pacific Conference on Circuits and Systems*, pp.173-174, Nov. 2014. (Ishigaki Island, Okinawa, Japan)
- [5] L. Lin, N. Kawarabayashi, M. Fukui, S. Tsukiyama, and I. Shirakawa, "An accurate SOC estimation system for lithium ion batteries," *in proc. IEEE Vehicle Power and Propulsion Conference*, pp. 33-36, Oct. 2014. (Coimbra, Portugal)
- [6] Y. Yamamoto, K. Kato, L. Lin, and M. Fukui, "A thermal management system for lithium-ion battery in mobile systems," *in proc. European Conference on Circuit Theory and Design*, pp.1-4, Sep. 2013. (Dresden, Germany)
- [7] L. Lin, M. Fukui, and S. Tsukiyama, "A GPGPU implementation of parallel backward Euler algorithm for power grid circuit simulation," *in proc. IEEE International NEWCAS Conference*, Jun.2013. (Paris, France)
- [8] L. Lin, K. Takaba, and M. Fukui, "Practical and accurate SOC estimation system for Lithium-ion batteries by EKF with adaptive noise covariance estimation," *in proc. Workshop on Synthesis and System Integration of Mixed Technologies*, pp. 152-157, Oct. 2016. (Kyoto, Japan)
- [9] R. Ishizaki, L. Lin, N. Kawarabayashi, and M. Fukui, "An SOC estimation system for Lithium-ion batteries considering thermal characteristics," *in proc. Workshop on Synthesis and System Integration of Mixed Technologies*, pp.16-21, Mar. 2015. (Yilan, Taiwan)
- [10] N. Kawarabayashi, L. Lin, R. Ishizaki, M. Fukui, and I. Shirakawa, "A battery smart sensor and its SOC estimation function for assembled Lithium-ion batteries," *in proc. Workshop on Synthesis and System Integration of Mixed Technologies*, pp.28-33, Mar. 2015. (Yilan, Taiwan)
- [11] A. Segawa, Y. Yamamoto, L. Lin, and M. Fukui, "A development and evaluation of variable speed charger system for lithium-ion battery," *in proc. Workshop on Synthesis and System Integration of Mixed Technologies*, pp.201-202, Oct. 2013. (Sapporo, Japan)
- [12] Y. Yamamoto, K. Kato, L. Lin, and M. Fukui, "A cool charger for lithium-ion battery," *in proc. Workshop on Synthesis and System Integration of Mixed Technologies*, pp.123-128, Oct. 2013. (Sapporo, Japan)
- [13] L. Lin, H. Shiono, M. Yokota, and M. Fukui, "GPGPU implementation of parallel backward Euler algorithm for power grid circuit simulation," *in proc. Workshop on Synthesis and System Integration of Mixed Technologies*, pp.460-321, Mar. 2012. (Beppu, Oita, Japan)

## 国内学会・シンポジウム等における発表

### 査読付き

- [1] 加藤啓路, 山本祐介, 林磊, 福井正博, “リチウムイオン蓄電池の熱特性解析モデルの構築と内部温度推定手法, 第 54 回電池討論会,” pp.89, 2013 年 10 月.
- [2] 石崎龍, 河原林直紀, 林磊, 福井正博, “温度変動を考慮したリチウムイオン蓄電池の高精度残量推定システムの開発,” 第 55 回電池討論会, pp.75, 2014 年 11 月.

### 査読なし

- [3] 大矢将輝, 鷹羽浄嗣, 林磊, 石崎龍, 福井正博, “パラメータ依存状態空間モデルに基づく蓄電池の SoC 推定,” 第 59 回システム制御情報学会研究発表講演会(SCI15), pp.59-63, 2015 年 5 月.
- [4] 渡邊正剛, 大村崇, 北川友貴, 林磊, 孟林, 福井正博, “GPU を用いた LSI 基板の 3 次元発熱・伝熱シミュレータの構築,” 電子情報通信学会 総合大会(シンポジウム CS-5), pp.1-6, 2015 年 3 月.
- [5] 河原林直紀, 林磊, 福井正博, “蓄電池スマートセンサーの実装と評価, 情報処理学会 DA シンポジウム 2014,” pp.33-38, 2014 年 8 月.
- [6] 河原林直紀, 林磊, 福井正博, “スマートグリッドのための蓄電池センサー技術, 第 58 回システム制御情報学会研究発表講演会,” 324-2, 2014 年 5 月.
- [7] 林磊, 河原林直紀, 福井正博, “リチウムイオン蓄電池の高精度残量予測システムの開発と実装,” 情報処理学会 システムと LSI 設計技術研究会(SLDM), pp.1-6, 2014 年 5 月.
- [8] 大村崇, 林磊, 孟林, 福井正博, “GPGPU を用いた VLSI チップ熱解析の一手法,” 電子情報通信学会 VLSI 設計技術研究会(VLD), pp.209-214, 2013 年 11 月.
- [9] 瀬川明大, 山本祐介, 林磊, 福井正博, “低温動作を目的とした可変速充電システムの实装と評価,” 電子情報通信学会 回路とシステム研究会(CAS), pp.147-152, 2013 年 7 月.
- [10] 林磊, 横田誠, 塩野隼人, 福井正博, “GPGPU によるバックワードオイラー法回路シミュレーションの高速化,” 電子情報通信学会ソサイエティ大会, pp.58-59, 2011 年 9 月.



## 目次

第 1 章	序章 .....	1
第 2 章	蓄電池モデル .....	5
2.1	蓄電池の構造と等価回路モデル .....	5
2.2	結言 .....	8
第 3 章	GPU の並列による計算高速化 .....	9
3.1	蓄電池熱解析モデルと熱解析 .....	9
3.2	GPU の構造と並列化による高速計算の仕組み .....	14
3.3	GPU による並列化プログラムの仕組み .....	17
3.4	結言 .....	25
第 4 章	内部パラメータの推定 .....	27
4.1	内部パラメータの推定のための蓄電池モデル .....	27
4.2	蓄電池内部パラメータ推定 .....	29
4.3	同定実験とその結果 .....	33
4.4	結言 .....	38
第 5 章	蓄電池残量推定 .....	39
5.1	蓄電池残量推定アルゴリズム .....	41
5.2	自動適応ノイズ推定アルゴリズム .....	47
5.3	残量推定実験と評価 .....	49
5.4	結言 .....	56
第 6 章	結論 .....	57
謝辞	.....	59
参考文献	.....	61



# 第1章 序章

近年，再生可能エネルギーの普及に伴い，エネルギーの貯蔵技術が重要な課題となっている．その中でも，リチウムイオン蓄電池は電気エネルギーを高密度に貯蔵できるデバイスとして注目されている．リチウムイオン蓄電池は昔から使われている鉛蓄電池，ニッケルカドミウム電池，ニッケル水素電池などの蓄電池より軽くて，エネルギー密度が大きい利点を持つ．図 1-1 は現在普及されている蓄電池のエネルギー密度の比較である [1]．リチウムイオン蓄電池はほかの蓄電池より優位性がある．

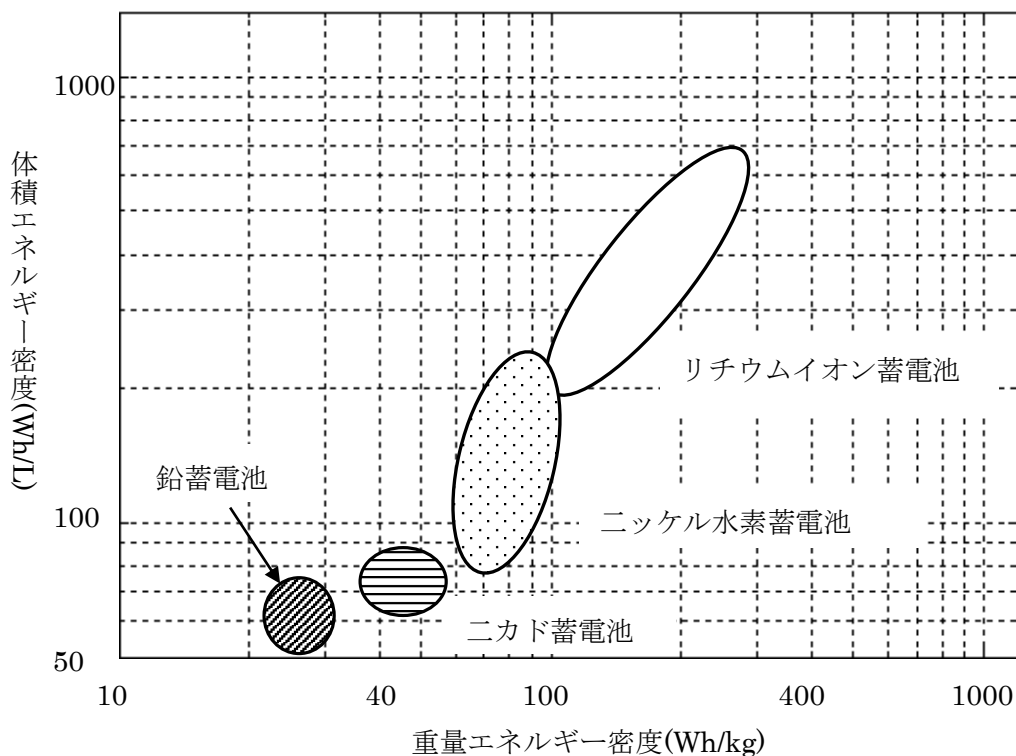


図 1-1 蓄電池のエネルギー密度の比較

リチウムイオン蓄電池の用途は商品化したとき（1991年ソニー社）の携帯電子機器のエネルギー源から電動工具，ハイブリッド車，EV（電動自動車），エネルギー貯蔵に使われていく傾向がある．そのために，安全かつ効率的な蓄電池の管理システム BMS（Battery Management System） [2, 3, 4] 必要性が高まっている．蓄電池管理システムにおいては，蓄電池熱管理，蓄電池内部パラメータ推定かつ劣化診断，残量予測重要な三つの課題になっている．本論文では，その三つの課題について述べる．

蓄電池熱管理の課題は蓄電池の熱上昇する問題を解析することが目的である。蓄電池劣化には様々な要因があるが、温度上昇による劣化は蓄電池劣化の要因の一つである。そのため、蓄電池の温度上昇を抑える技術は重要になる。そうすると、蓄電池の熱モデルが必要である。本論文では、熱回路を用いた蓄電池の熱挙動を解析する。熱回路網法は物体の伝熱を表現する方法である [5, 6]。熱回路モデルの規模は大きいので、並列計算を行わないと、膨大な時間がある。その問題を解決するために、GPGPU を用いた電気回路の並列計算を転用して、熱回路モデルの解析の高速化手法を提案する。従来法では、前進オイラー法 (forward Euler's rule) で解析するので、高速な並列処理できるが、精度はよろしくない欠点がある [7]。シミュレーション精度を上昇するために、後退オイラー (backward Euler's rule) を用いた手法は有効な手段と認められている [8]。本論文はそれに対して GPU による回路モデルの解析の高速化手法を述べる。

蓄電池内部パラメータ推定は電池の内部抵抗、OCV などの内部パラメータのリアルタイム推定を目的に設定されている。蓄電池は内部抵抗の劣化に伴い変化するため、起電力や内部抵抗などの内部パラメータ変化する。その原因で、蓄電池の内部パラメータをリアルタイム的に把握するのは重要な課題になる。本論文では蓄電池の OCV と電極表面抵抗の同定を目的とし、SOC と OCV の関係に着目することにより、端子電圧と端子電流の比率に基づき最適な忘却係数を算出する発見的手法を提案する [9]。従来の同定手法 [10]では、電池の起電力 (OCV) の短時間のふるまいをコンデンサで近似することにより、簡便なモデル化を行っているが、一般に、OCV は SOC に対して非線形性を有し、放電カーブがコンデンサとは大きく異なるため、同定精度に支障をきたす。別の同定手法 [11, 12]では、短時間の起電力 (OCV) を電圧源として表現する。これらの手法では固定的な忘却係数  $\lambda$  を用いて、計測時間中に OCV が変動することによって生じる誤差の問題を解消している。しかし、一般に、蓄電池の OCV の変動量は、電流や SOC の影響を受けるため、一定ではない。このような同定問題に対して忘却係数  $\lambda$  を固定とするのは適切ではない。本論文では、この問題を解決するため、最適な忘却係数の与え方を論じる。

蓄電池残量予測は最も重要かつ基本的な技術である。蓄電池残量は利用可能な容量としてどのぐらい貯蔵しているか、という重要な電池指標である。しかし、それは内部抵抗による電圧降下、分極等の影響を受ける為、簡単ではない。古典的な蓄電池残量予測方法は電流積算法 [13]、端子電圧法 [14]と内部抵抗法 [15]が提案されている。電流積算法は高い予測精度を実現するが、初期状態を知る必要がある。端子電圧法は充放電による電圧降下、分極の影響で高予測精度を実現することは難しい。内部抵抗法は簡単であるが、電池の内部抵抗と残量との依存性が少ないために、高予測精度を確保することが難しい。端子電圧、電流、内部抵抗、分極による電池端子電圧の過度現象をモデル化して、拡張カルマンフィルタで処理する残量予測方法は高精度かつリアルタイム性の良い予測方法と認められている [16]。そのほか、Plett ら [17, 18, 19]は蓄電池の内部パラメータと SOC 同時に推定を行う研究して SOC 推定高精度化を実現できる。そのほか、Sigma-Point カルマンフィルタによる SOC



推定の高精度化 [20, 21]やニューラル・ネットワークとカルマンフィルタを併用する研究 [22]があるが, 拡張カルマンフィルタを応用するときのノイズの自動定期に推定を論じていない. 本論文では, 拡張カルマンフィルタを用いた高精度残量予測手法を提案し, 実装と評価を行う, そして, カルマンフィルタに置いてノイズ分散値の設定方法を議論する. 提案手法はイノベーションによる最適なノイズ分散を自動に推定する.



## 第2章 蓄電池モデル

### 2.1 蓄電池の構造と等価回路モデル

蓄電池は電気エネルギーを化学エネルギーに貯蔵し，電気エネルギーに変換するデバイスである．構造 [23]としては図 2-1 のように正負極集電体，正極，負極，電解液，セパレーターで構成される．正極と電解液の間に電気二重層がある．リチウムイオン  $\text{Li}^+$ が負極に移動すると，電子は正極に移動する，その動きによる蓄電池は充電状態になる．逆になると，放電状態になる．リチウムイオン蓄電池の正極は結晶構造であり，負極は多層構造である．

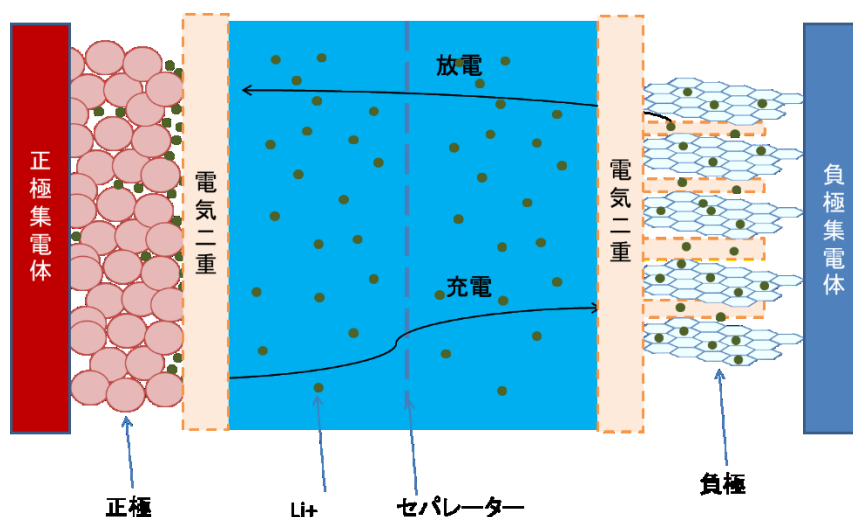


図 2-1 リチウムイオン蓄電池の内部構造

蓄電池の充放電反応は複雑なので，電池を抽象的なモデルを作る必要である．モデルとして，電気化学モデルや等価回路モデルが提案されているが．本論文では，観測が容易な端子電圧と端子電流との関係が明確で扱いやすい等価回路モデルを使用する．電気化学モデルは精度がいいが，複雑である，等価回路モデルは簡単かつ扱いやすい．電気化学モデルと等価回路モデルの特徴は以下の表で与えられる [24].

表 2-1 電気化学モデルと等価回路モデルの特徴

	等価回路モデル	電気化学モデル
内部構造	交流インピーダンス特性 起電力と SOC 関係	Butler-Volmer の式 Newman の多孔体電極
長所	単純で取り扱いが簡単	物理的なパラメータが詳細
短所	特性の合わせ込みが困難	式が複雑

以上の説明で、本論文は等価回路モデルを選択し、蓄電池の残量と内部パラメータ推定を行う。

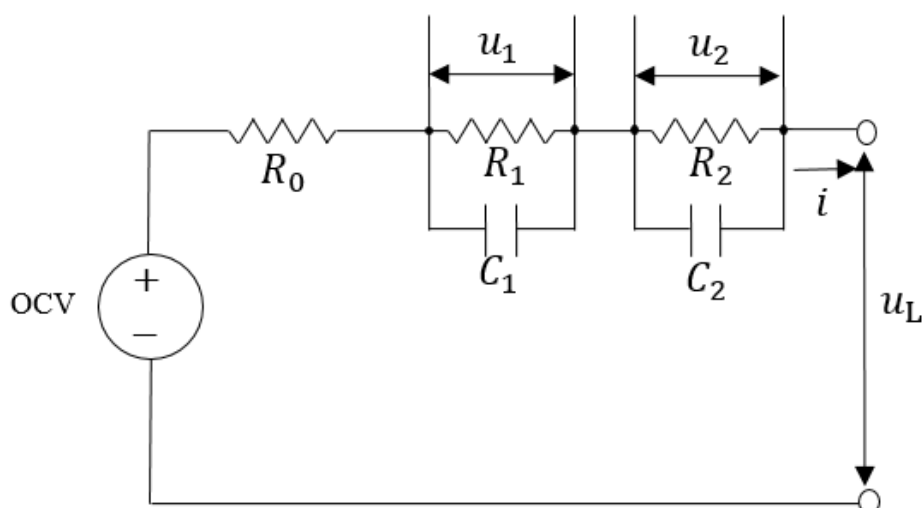


図 2-2 蓄電池の等価回路モデル

図 2-2 にリチウムイオン蓄電池の等価回路モデルを示す。同モデルでは、起電力を表す電圧源 OCV (Open Circuit Voltage), 溶液抵抗を表す抵抗  $R_0$ , 電極表面の電気二重層, 電極内部の拡散現象を表現する抵抗とコンデンサ並列回路 ( $R_1, C_1, R_2, C_2$ ) から成る [10] [11] [12].  $u_1$  は蓄電池の端子間電圧,  $i$  は端子電流である。電極表面の電気二重層を表す抵抗とコンデンサ並列回路 ( $R_1, C_1$ ) の時定数は小さく (数ミリ秒~数秒), 一方で並列回路 ( $R_2, C_2$ ) の時定数は前述に比べて相当大きい (数秒~数時間)。

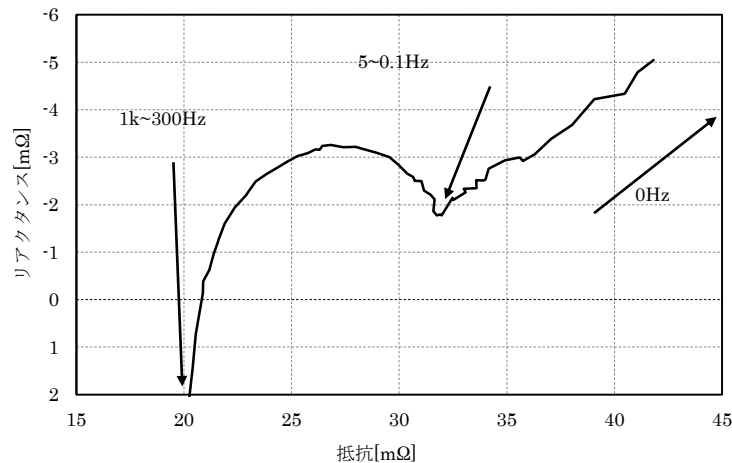


図 2-3 18650 型蓄電池のナイキスト線図

評価する 18650 型リチウムイオン蓄電池を、交流インピーダンス法 [25]により解析したナイキスト線図を図 2-3 に示す。18650 型リチウムイオン蓄電池のスペックは以下の表で与えられる。

表 2-2 18650 型リチウムイオン蓄電池のスペック

公称容量	2250[mAh]
公称電圧	3.6[V]
カットオフ電圧	3.0[V]
充電電圧	4.2[V]

蓄電池の応答周波数領域は広く、一番応答が速い溶液抵抗の反応周波数は 1k~300Hz、電極表面の電気二重層抵抗の反応周波数は 300~0.1Hz、電極内部活物質の拡散抵抗の反応周波数は 0.1Hz 以下となる。各部分の応答周波数の差は 1000 倍以上であるため、全領域をカバーするシステム同定を行うのは難しい。そのために、特定の応答周波数領域に限定し、モデルを構築する必要がある。

また、OCV は SOC に依存する物理量であり、その関係を表す OCV-SOC 曲線は SOC の推定に重要な役割を果す。この OCV-SOC 曲線は 0.02C の微小電流を印加して充電および放電したときの端子電圧値の平均値で近似した曲線である。その曲線は蓄電池の劣化によってもほとんど変動しないため、放電中の 2 ポイントの OCV とその間の放電量から、蓄電池容量を算出できる。

OCV の測定方法は微小電流で充放電して得た充電電圧曲線と放電電圧曲線の平均値で実際の OCV 曲線と近似することができる。具体的な方法は、残量 SOC=0 の時から 0.04C の微小電流で 25 時間充電した際の電圧変化曲線と、SOC=1 の時から 0.04C の微小電流で 25 時間放電した際の電圧変化曲線の中心値が、各 SOC 状態での OCV 曲線と近似する [24]。

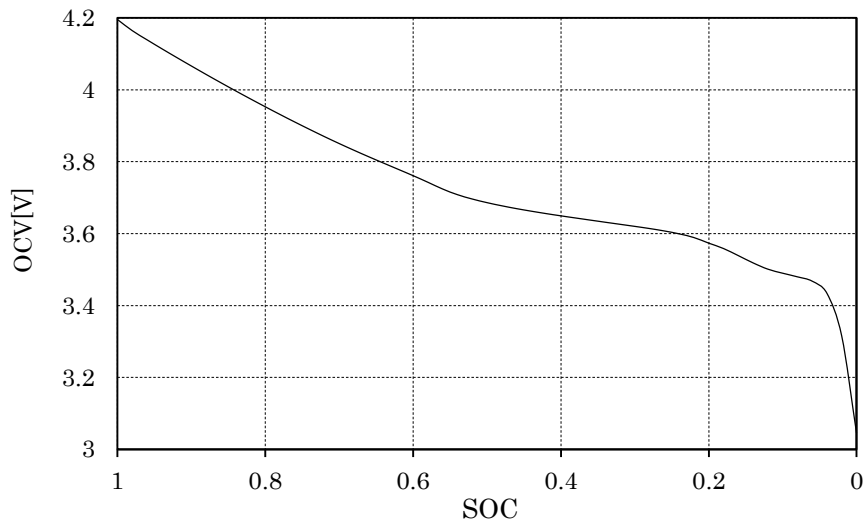


図 2-4 蓄電池の OCV-SOC 曲線

SOC-OCV 曲線は複雑であるので、線形関数で表すのは難しい。我々は複雑な初等関数 (exponential 関数等) を避けるために、相対的に簡単な多項式で回帰方法を選択した。式 (3.1) で表す。我々は 12 次多項式で回帰することで、回帰精度は最大誤差 9.233 [mV]、平均誤差は 1.071 [mV] である。

$$\text{OCV}(\text{SOC}) = \sum_{i=0}^{12} a_i \text{SOC}^i \quad (3.1)$$

## 2.2 結言

本章では、リチウムイオン蓄電池の構造と等価回路モデルを紹介して、等価回路モデルは精度的な損失があるが、単純で取り扱いが簡単な利点で、等価回路モデルを採用した。以降の蓄電池の内部パラメータ推定と残量推定では、等価回路モデルに基づいて行う。

## 第3章 GPU の並列による計算高速化

本章では、熱回路を用いた蓄電池の熱挙動を高速に解析する手法について述べる。熱回路モデルの規模は大きいので、並列計算を行わないと、膨大な時間がある [26]。その問題を解決するために、GPGPU を用いた電気回路の並列計算を用いる。図 3-1 は熱回路モデルの解析の高速化手法の仕組みである。熱回路生成、圧縮、結果保存は CPU で処理して、計算部を GPU で並列化処理するのは手案手法の特徴である。以下では GPU の並列計算の仕組みを説明する。

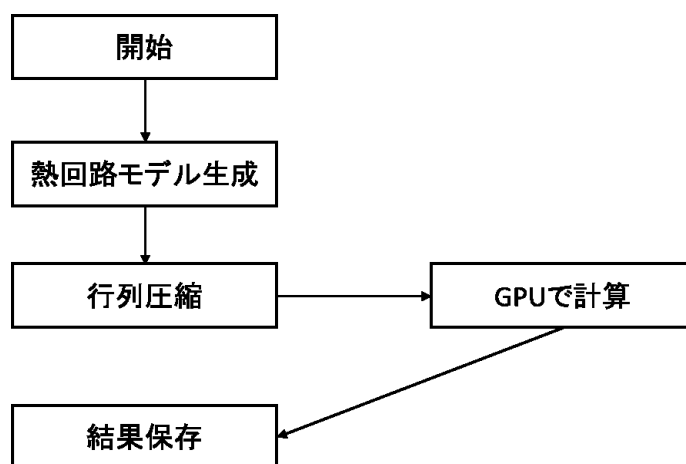


図 3-1 熱回路モデルの解析の高速化手法の仕組み

### 3.1 蓄電池熱解析モデルと熱解析

リチウムイオン蓄電池は比較的劣化が少ない蓄電池とされるが、高価であるため、劣化を抑えた使用が望まれる。蓄電池劣化には様々な要因があるが、負極活物質表面における SEI 生成や、正極活物質における立方晶化などの化学現象が温度により加速されることから、蓄電池の温度管理の重要性が求められている [27, 28]。そのため、蓄電池の熱モデルが必要である [29, 30]。熱回路を基づいた蓄電池の発熱や放熱の挙動は熱回路で表すことができる有効かつ簡単なモデルである [28]。論文 [28]ではマイクロモデルに関する蓄電池を熱回路で表し、発熱の理論や、各パラメータの解析を実験により各パラメータを測定し、空冷を用いることによって温度上昇が抑えられる理論を実験により検証したが、蓄電池の内部の熱挙動は表すことができない。よりよく精密な蓄電池の内部の熱挙動を解析するために、熱回路を用いて蓄電池熱挙動を解析するマイクロモデルの構築が必要である [26]。

熱回路網法は物体の伝熱を、ノードの温度 $T$ 、および、ノード間の熱の流れを熱流量 $Q_T$ 、ノード間の熱の伝わりにくさを熱抵抗 $R_T$ 、により、伝熱経路を表現する方法である [5, 6]。また、ノードに温度あたりの蓄熱できる量を熱容量 $C_T$ 、熱量を $H$ であらわされる。熱回路において、温度 $T$ 、熱流量 $Q_T$ 、熱抵抗 $R_T$ 、熱容量 $C_T$ 、熱量 $H$ は、電気回路における電圧 $V$ 、電流

$I$ , 電気抵抗 $R_E$ , 静電容量 $C_E$ , 電荷量 $Q_E$ に対応づけられ, オームの法則や過渡応答などが同様に成立する. 熱回路と電気回路物理量の対応関係は表 3-1 に載せている. 図 3-2 は二次元の熱回路網である. 図 3-3 は頂点 $(i, j)$ における熱回路モデルである. 以上の説明で, 熱回路モデルと電気回路の仕組みが同じであるので, 回路のシミュレーション技法は熱回路モデルに転用できる.

蓄電池劣化には様々な要因があるが, 温度上昇による劣化は蓄電池劣化の要因の一つである. そのため, 蓄電池の温度上昇を抑える技術は重要になる. そうすると, 蓄電池の熱モデルが必要である.

表 3-1 熱回路と電気回路物理量の対応表

熱回路		電気回路	
物理量	単位	物理量	単位
温度 $T$	[K]	電圧 $V$	[V]
熱流量 $Q_T$	[J/S]	電流 $I$	[A]
熱抵抗 $R_T$	[K/W]	電気抵抗 $R_E$	[ $\Omega$ ]
熱容量 $C_T$	[J/S]	静電容量 $C_E$	[F]
熱量 $H$	[J]	電荷量 $Q_E$	[C]

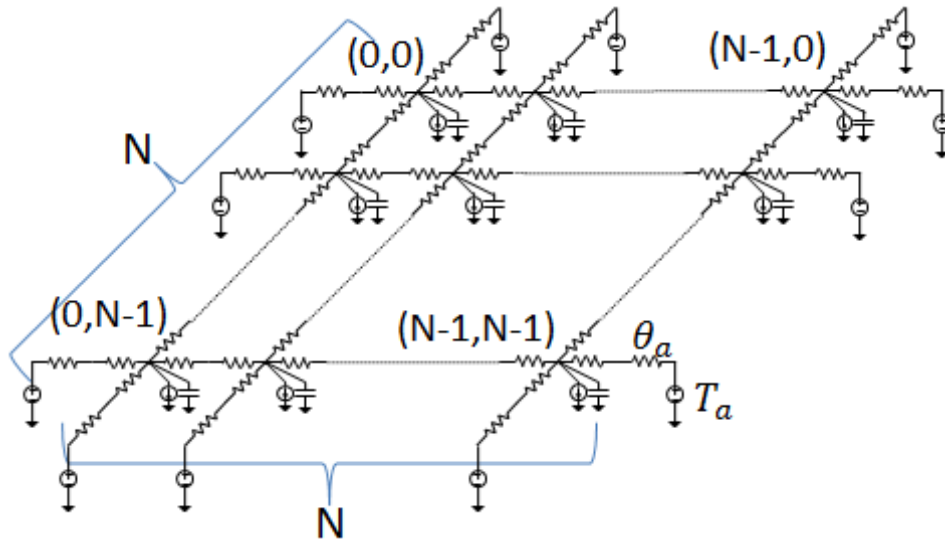


図 3-2 二次元熱回路網



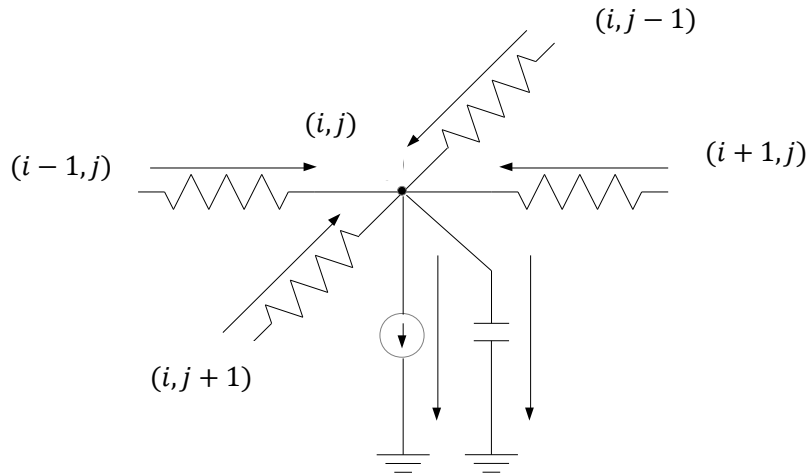


図 3-3 頂点 $(i, j)$ における熱回路モデル

一般に回路解析手法としては前進公式と後退公式が考えられる。しかし、前進オイラー法解の収束性(安定性)が悪く低信頼であるため、計算精度が低かった。その弱点を改善するために、本研究では解の収束性がよりよい後退オイラー法(Backward Euler's Method, BE法)を解析手法 [31, 32]として採用した。

後退オイラー法とは、微分方程式の数値解法の一つである。時間に依存する変数に関する微分項 $dx(t)/dt = \dot{x}(t)$ を式(3.1)のように差分化することを考える。ここで、 $\Delta t$ は、微小な時間刻み幅である。なお、 $\Delta t$ は0より大きい微小な数字である。

$$\frac{dx}{dt} = \lim_{\Delta t \rightarrow 0} \frac{x(t + \Delta t) - x(t)}{\Delta t}, \quad \Delta t > 0 \quad (3.1)$$

微分項 $\frac{dx}{dt}$ を時刻 $t + \Delta t$ における微分値 $\frac{dx(t+\Delta t)}{dt}$ を利用して、整理することで、式(3.1)は式(3.2)のように書き改められる。

$$\frac{dx(t + \Delta t)}{dt} = \lim_{h \rightarrow 0} \frac{x(t + \Delta t) - x(t)}{\Delta t} \approx \frac{x(t + h) - x(t)}{\Delta t}, \quad \Delta t > 0 \quad (3.2)$$

$x(t + \Delta t)$ を左側に移動して、微分項 $\frac{dx(t+\Delta t)}{dt}$ 右側に移動して、整理することで、式(3.3)を得ることが出来る。 $\dot{x}(t + \Delta t)$ は $\frac{dx(t+\Delta t)}{dt}$ である。

$$x(t + \Delta t) = x(t) + \Delta t \dot{x}(t + \Delta t), \quad \Delta t > 0 \quad (3.3)$$

図 3-4 は式(3.3)を幾何学の観点から表したグラフである。式(3.3)では両辺に未知変数 $x(t + \Delta t)$ を含んでいるので、式(3.3)を時刻 $t$ に関して反復すると、各ステップにおいて方程式の解を求めることが必要である。だから、大規模な熱回路に対して、データ量、計算量が増大する。

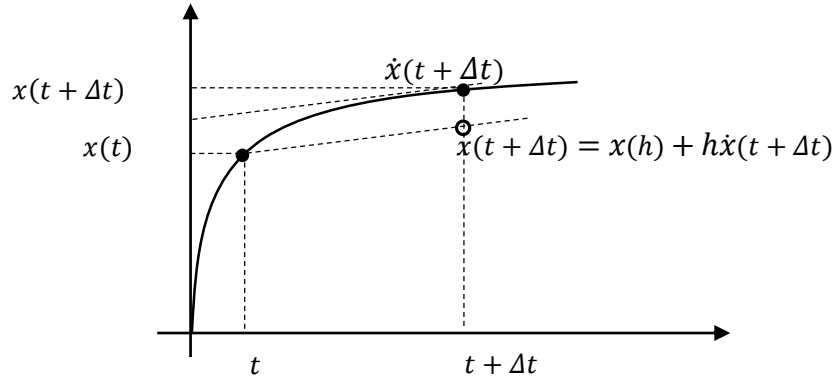


図 3-4 後退オイラー法による解の更新

熱量 $H^{(i,j)}(t)$  はノードの熱容量 $C^{(i,j)}$ に流出入する熱流量の合計によって決まる．単位時間に流出する熱流量の合計を $Q_{all}^{(i,j,k)}(t + \Delta t)$  とすると，単位時間に変化するノードの熱量  $H^{(i,j)}(t + \Delta t)$ は式(2.4)で表せる． $Q_{all}^{(i,j,k)}(t + \Delta t)$  は，あるノード $(i,j)$ の4方向から流出入する熱流量と発熱および垂直方向の熱収支を表す $Q_s^{(i,j)}$  によって決まる．ノード間の熱抵抗を $\theta$ とすると $Q_{all}^{(i,j)}(t + \Delta t)$ は式(2.5)で表わせる．式(3.3)，(3.4)，(3.5)を整理し，(3.6)式が得られる [33].

$$H^{(i,j)}(t + \Delta t) = H^{(i,j)}(t) + Q_{all}^{(i,j)}(t + \Delta t) \times \Delta t \quad (3.4)$$

$$Q_{all}^{(i,j)}(t + \Delta t) = \sum_{\substack{(x,y)=(i-1,j),(i+1,j), \\ (i,j-1),(i,j+1)}} \frac{T^{(i,j)}(t + \Delta t) - T^{(x,y)}(t + \Delta t)}{R_{(i,j)-(x,y)}} - Q_s \quad (3.5)$$

$$\frac{\Delta t}{C} \begin{bmatrix} -\frac{1}{\theta} & -\frac{1}{\theta} & -\frac{1}{\theta} & -\frac{1}{\theta} \\ \frac{4\Delta t + CR}{\Delta t\theta} & & & \\ & & & \\ & & & \end{bmatrix} \begin{bmatrix} T^{(i-1,j)}(t + \Delta t) \\ T^{(i+1,j)}(t + \Delta t) \\ T^{(i,j-1)}(t + \Delta t) \\ T^{(i,j+1)}(t + \Delta t) \\ T^{(i,j)}(t + \Delta t) \end{bmatrix} = T^{(i,j)}(t) - \frac{\Delta t}{C} Q_s \quad (3.6)$$

全てのノードに対する式(3.6)の行列を解き，温度を求める．熱回路網のノード間接続の規則性から係数行列は対角優位の疎行列となる．回路サイズを $3 \times 3$ ノードとしたときの例を式(3.7)に示す．式(3.7)を行列 $A$ ，ベクトル $x$ ， $b$ で表現すると， $Ax = b$ になる．回路網のシミュレーション問題は $Ax = b$ の解 $x$ になる．

$$\begin{bmatrix}
\frac{2\Delta t}{C\theta} + \frac{2\Delta t}{C\theta_s} + 1 & -\frac{\Delta t}{C\theta} & 0 & -\frac{\Delta t}{C\theta} & 0 & 0 & 0 & 0 & 0 \\
-\frac{\Delta t}{C\theta} & \frac{3\Delta t}{C\theta} + \frac{1\Delta t}{C\theta_s} + 1 & -\frac{\Delta t}{C\theta} & 0 & -\frac{\Delta t}{C\theta} & 0 & 0 & 0 & 0 \\
0 & -\frac{\Delta t}{C\theta} & \frac{2\Delta t}{C\theta} + \frac{2\Delta t}{C\theta_s} + 1 & 0 & 0 & -\frac{\Delta t}{C\theta} & 0 & 0 & 0 \\
-\frac{\Delta t}{C\theta} & 0 & 0 & \frac{3\Delta t}{C\theta} + \frac{1\Delta t}{C\theta_s} + 1 & -\frac{\Delta t}{C\theta} & 0 & -\frac{\Delta t}{C\theta} & 0 & 0 \\
0 & -\frac{\Delta t}{C\theta} & 0 & -\frac{\Delta t}{C\theta} & \frac{4\Delta t}{C\theta} + 1 & -\frac{\Delta t}{C\theta} & 0 & -\frac{\Delta t}{C\theta} & 0 \\
0 & 0 & -\frac{\Delta t}{C\theta} & 0 & -\frac{\Delta t}{C\theta} & \frac{3\Delta t}{C\theta} + \frac{1\Delta t}{C\theta_s} + 1 & 0 & 0 & -\frac{\Delta t}{C\theta} \\
0 & 0 & 0 & -\frac{\Delta t}{C\theta} & 0 & 0 & \frac{2\Delta t}{C\theta} + \frac{2\Delta t}{C\theta_s} + 1 & -\frac{\Delta t}{C\theta} & 0 \\
0 & 0 & 0 & 0 & -\frac{\Delta t}{C\theta} & 0 & -\frac{\Delta t}{C\theta} & \frac{3\Delta t}{C\theta} + \frac{1\Delta t}{C\theta_s} + 1 & -\frac{\Delta t}{C\theta} \\
0 & 0 & 0 & 0 & 0 & -\frac{\Delta t}{C\theta} & 0 & -\frac{\Delta t}{C\theta} & \frac{2\Delta t}{C\theta} + \frac{2\Delta t}{C\theta_s} + 1
\end{bmatrix}
\begin{bmatrix}
T^{(0,0)}(t + \Delta t) \\
T^{(0,1)}(t + \Delta t) \\
T^{(0,2)}(t + \Delta t) \\
T^{(1,0)}(t + \Delta t) \\
T^{(1,1)}(t + \Delta t) \\
T^{(1,2)}(t + \Delta t) \\
T^{(2,0)}(t + \Delta t) \\
T^{(2,1)}(t + \Delta t) \\
T^{(2,2)}(t + \Delta t)
\end{bmatrix}
=
\begin{bmatrix}
T^{(0,0)}(t) - \frac{\Delta t}{C}(Q_s - 2\frac{T_s}{\theta_s}) \\
T^{(0,1)}(t) - \frac{\Delta t}{C}(Q_s - \frac{T_s}{\theta_s}) \\
T^{(0,2)}(t) - \frac{\Delta t}{C}(Q_s - 2\frac{T_s}{\theta_s}) \\
T^{(1,0)}(t) - \frac{\Delta t}{C}(Q_s - \frac{T_s}{\theta_s}) \\
T^{(1,1)}(t) - \frac{\Delta t}{C}Q_s \\
T^{(1,2)}(t) - \frac{\Delta t}{C}(Q_s - \frac{T_s}{\theta_s}) \\
T^{(2,0)}(t) - \frac{\Delta t}{C}(Q_s - 2\frac{T_s}{\theta_s}) \\
T^{(2,1)}(t) - \frac{\Delta t}{C}(Q_s - \frac{T_s}{\theta_s}) \\
T^{(2,2)}(t) - \frac{\Delta t}{C}(Q_s - 2\frac{T_s}{\theta_s})
\end{bmatrix}
\quad (3.7)$$

$$Ax = b \quad (3.8)$$

後退オイラー法の行列計算に適した緩和法の 1 つであるヤコビ法を用いる。計算の順序の依存性がないため並列計算に適しており、対角優位の場合、収束することが保証される。式(3.9)により、行列 $A$ は、対角成分からなる対角行列 $D$ と、行列 $A$ の対角行列を除いた行列 $R$ に分割することができる。式(3.8)を式(3.9)に代入することで式(3.10)が得られる。さらに、式(3.10)を展開し係数 $Rx$ の項を移項することで(3.11)式が得られる。

$$A = D + R \quad (3.9)$$

$$(D + R)x = b \quad (3.10)$$

$$Dx = b - Rx \quad (3.11)$$

式(3.11)を満たす $x$ は、任意の初期値 $x^0$ に対し、 $k$ 回の反復で得られる解を $x^{(k)}$ で近似できる。式(3.11)の両辺に $D^{-1}$ を掛けて整理すると、式(3.12)が得られる。さらに、式(3.12)を $i$ 番目の要素 $x_i$ について書き改めると、式(3.13)が得られる。

$$x^{(k)} = D^{-1}(b - Rx^{(k-1)}) \quad (3.12)$$

$$x_i^{(k)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j \neq i}^N a_{ij} x_j^{(k-1)} \right) \quad (3.13)$$

ヤコビ法のアルゴリズムは以下の通りである。

---

**Algorithm:** Jacobi Method for  $A\mathbf{x} = \mathbf{b}$

---

$k \leftarrow 0$ ;

Guess  $x^0$

repeat {

$k \leftarrow k + 1$

    foreach( $i \in \{1, \dots, n\}$ )

$$x_i^{(k)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j \neq i}^N a_{ij} x_j^{(k-1)} \right)$$

    } until  $(\sum_k^N |x_i^{(k)} - x_i^{(k-1)}| \leq \varepsilon)$

---

ここで、 $\varepsilon$ は最大許容誤差である。数式に改めると式(3.14)で表すことができる。

$$\sum_k^N |x_i^{(k)} - x_i^{(k-1)}| \leq \varepsilon \quad (3.14)$$

式(3.14)を満たさない限り、式(3.13)の反復を行う。つまり、反復回数 $k$ は最大許容誤差 $\varepsilon$ によって決定される。

回路網の規模が大きくなると、行列のサイズが大きくなる、計算時間が大分かかる。そのために、高速化が必要である。並列処理は高速化する有効な手段である。近年、GPU (Graphic Processing Unit) [34]を一般科学技術計算に適用する GPGPU (General Purpose GPU)技術に注目された。GPU は従来グラフィック処理専用プロセッサであり、CUDA(Compute Unified Device Architecture)という総合開発環境の発展を伴い、科学計算の領域も利用されることができるようになった。GPUは複数の計算単位が含まれているので、各計算単位は独立になるので、GPU 上では高並列度のプログラミングが可能となった。

### 3.2 GPU の構造と並列化による高速計算の仕組み

本研究では、NVIDIA 社の GPU GeForce GTX260 と GTX580 を用いていた。第 2 世代アーキテクチャ(GT200 アーキテクチャ) [34]を採用している GTX260 と、第 3 世代アーキテクチャ(Fermi アーキテクチャ)を採用している GTX580 を使って評価実験を行う。本文では Fermi アーキテクチャを中心に説明する。図 3-5 には Fermi アーキテクチャを示す。

Fermi アーキテクチャは、図 3-6 のように 16 個の Streaming Multiprocessor(SM)と、L2 キャッシュ、DRAM、GigaThread グローバルスケジューラからなる。GigaThread グローバルスケジューラは SM スレッドスケジューラにスレッドがブロックを配布しています。各 SM は、2 つのワーブスケジューラ、ディスパッチユニットと、32 個の CUDA

Core, 共有メモリ等から成る. しかし, GTX260 に対しては, 216 コアしか含めていない. 各 CUDA コアはパイプライン化された整数演算論理ユニット (ALU) と浮動小数点ユニット (FPU) 持っている. ワープスケジューラは, 32 本の並列スレッドをグループ

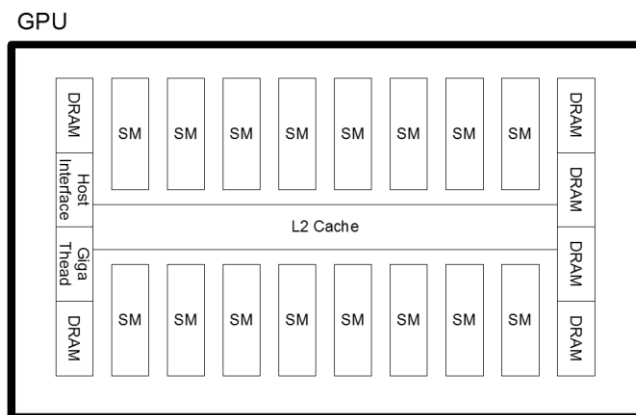


図 3-5 Fermi アーキテクチャ

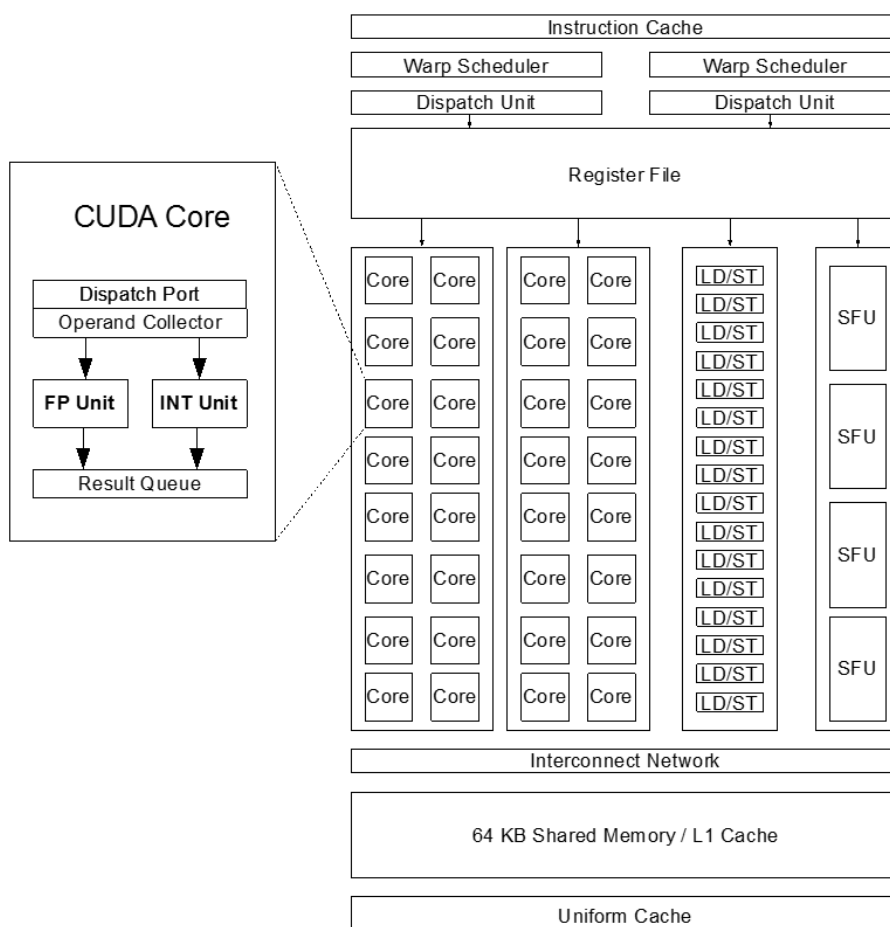


図 3-6 Streaming Multiprocessor(SM)構造

化したワープという単位のスケジューリングを行い、ディスパッチユニットは命令ディスパッチを行う。これらによって、並列実行がスケジューリングされている。Fermi ではワープスケジューラ、ディスパッチユニット 2 つずつあるので、2 つのワープを並列実行することができる。しかし、GTX260 に対しては、1 つのワープしか並列実行することができない。Fermi アーキテクチャは単精度と倍精度乗算加算 (FMA) 算術命令が提供しているので、1 つのステップで倍精度の乗算加算を計算できる。GTX260 では、整数 ALU が乗算演算用の 24 ビット精度に制限されていたので、同じ倍精度の乗算加算に対してはマルチエミュレーション命令シーケンスが必要である。

GPU を動かすために、並列計算統合開発環境は不可欠である。NVIDIA 社の GPU 並列計算統合開発環境は CUDA(Compute Unified Device Architecture)と呼ばれる。CUDA は C 言語の統合開発環境を提供しているだけでなく、C++, Fortran などの言語も提供している。本研究では、C 言語を使って、シミュレーションのプロを行う。CUDA プログラミングの演算処理単位は CUDA カーネルと呼ばれている。CUDA カーネルが上からグリッド、ブロック、スレッドという三階層構造となっている。図 3-7 に処理単位の概略図を示す。

実行時は、並列ブロックで構成されたグリッド上に GPU がカーネルプログラムのインスタンスを生成し、ブロックを構成するスレッド 1 つ 1 つそれぞれがカーネルのインスタンスを実行する。なお、各ブロックは独立になるので、各ブロック間ではデータの移動と同期することができない。一つのブロック内のスレッド間ではデータの共有と同期することができる。スレッド間でデータの共有は共有メモリで実現するので、共有メモリを活用したら、高速化を実現する重要なポイントである。

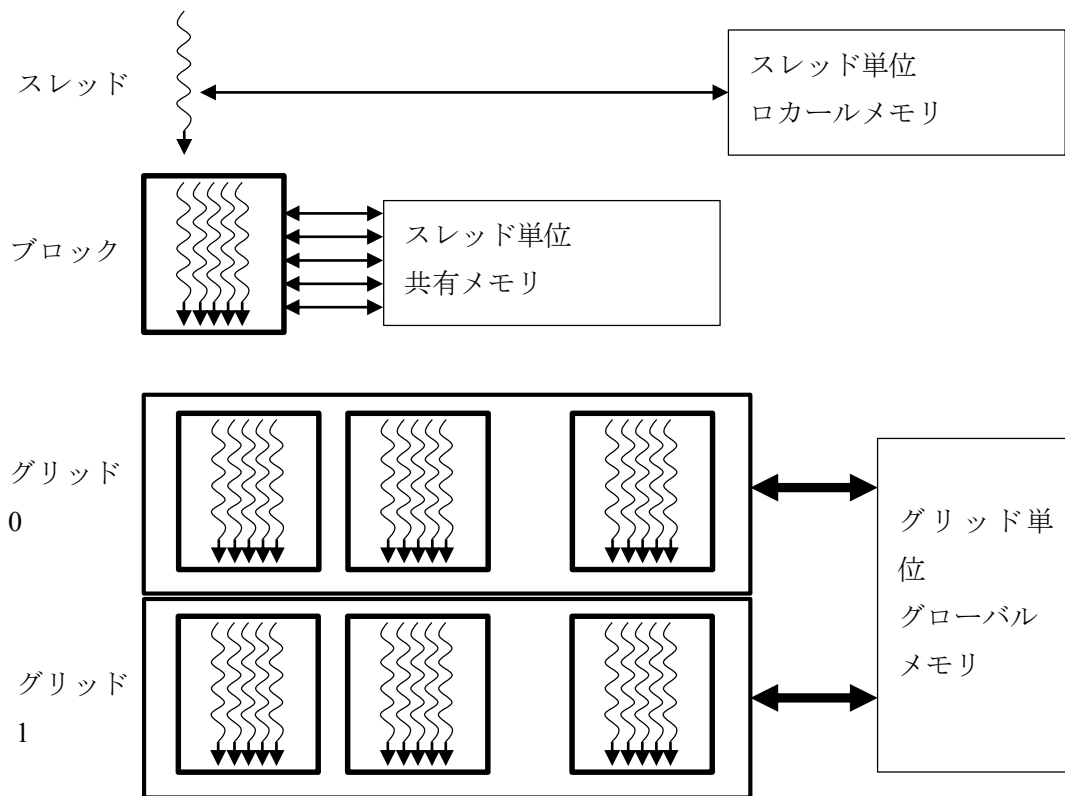


図 3-7 CUDA 処理単位カーネル概略図

### 3.3 GPU による並列化プログラムの仕組み

バックワードオイラー法による熱回路網解析手法はメモリの使用量は増大になるので、GPU にアルゴリズムを実装する際に、並列度の確保と並び重要なのが使用データ量の削減することが必要である。この問題を解決する手段として、疎行列の効率的な表現を用いるメモリ領域の削減手法を作る。従来例としては CSR(Compressed Sparse Row)法があるので、本研究では熱回路網行列の特徴を活用する CSG 手法(Compressed Sparse for Grid structure)を用いた使用データ量を削減する手法を示す [8, 35].

疎行列の表現として従来用いられている CSR (Compressed Sparse Row) [36, 37]は行列の中のノンゼロの値を抽出する圧縮表現手法である。具体的には図 3-8 を例として説明する。行列 $M$ は  $4 \times 4$  サイズの疎行列である。これを、ノンゼロ成分の要素の値を並べたベクトル  $val$ 、 $val$  のそれぞれの値が位置する列番号  $col\_idx$ 、さらに、 $col\_idx$  において行が変わる位置を示す  $row\_ptr$ 、という 3 つの行ベクトルへ変換する。この方法は、一般的な回路構造に対する疎行列には有効な表現方法である。今、 $N \times N$  サイズの回路に対しノンゼロ成分を  $n_{no\_zero}$  とすると、CSR 非適用時に必要なメモリ領域が  $N^2$  であるのに対し、CSR の適用によって  $2n_{no\_zero} + N$  のメモリ領域消費で済む。なお、この一般例は CSR 非適用時の

方がメモリ消費は少ないが、これは回路サイズが小さいためであり、大きくなればなるほどこの手法は有効である。

$$M = \begin{bmatrix} 1 & 3 & 0 & 0 \\ 0 & 6 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 2 & 0 & 0 & 4 \end{bmatrix}, \quad \begin{array}{l} \text{val} = [1 \ 3 \ 6 \ 5 \ 2 \ 4] \\ \text{col\_idx} = [1 \ 2 \ 2 \ 3 \ 1 \ 4] \\ \text{row\_ptr} = [1 \ 3 \ 4 \ 5] \end{array}$$

図 3-8 CSR の一般例

説明した CSR 法は一般的な回路構造に対する疎行列には有効な表現方法であるが、熱路の場合、非対角成分は上、下、右、左の四つの隣頂点への接続のみであるという規則性があるので、より効率のよい表現方法が可能である。熱回路網行列で用いる疎行列の表現方法 CSG (Compressed Sparse for Grid structure) を、図 3-9 を用いて説明する。熱回路網行列を表現する行列は式(3.7)を見ながら説明すると理解しやすい。行番号、列番号はそれぞれ熱回路網の各頂点に対応しており、行列の対角成分は各頂点そのものに対応している。さらに、各行のノンゼロ要素は、対角要素が頂点に対応し、それ以外が、同頂点から上下左右に接続する辺に対応している。前述のヤコビ法において、熱回路網モデルの対象行列  $A$  は、対角行列  $D$  と疎行列  $R$  に分割されるが、CSG において、対角行列  $D$  は、対角成分のみを行ベクトル  $d$  として保存する。頂点から上下左右に接続する辺の値は、頂点に対応する行に存在するので、その値と列位置を  $R\_value$ ,  $R\_position$  に格納する。このデータを用いれば、式(3.13)の計算において、必要なデータ  $j$  (最大 4 個) に直接アクセスでき、GPU の並列性を損なわず、かつ、必要最小限の効率の良い計算が行える。もし、CSR で同様のことを行うとすれば、行成分の抽出、行の各要素に対する列成分の抽出が必要なので、各スレッドにデータを割り振るための計算の手間が増大する。

$d$		R_value					R_position			
		Top	Left	Right	Down		Top	Left	Right	Down
$a_{11}$	1	*	*	$a_{12}$	$a_{14}$	1	*	*	2	4
$a_{22}$	2	*	$a_{21}$	$a_{23}$	$a_{25}$	2	*	1	3	5
$a_{33}$	3	*	$a_{32}$	*	$a_{36}$	3	*	2	*	6
$a_{44}$	4	$a_{41}$	*	$a_{45}$	$a_{47}$	4	1	*	5	7
$a_{55}$	5	$a_{52}$	$a_{54}$	$a_{56}$	$a_{58}$	5	2	4	6	8
$a_{66}$	6	$a_{63}$	$a_{65}$	*	$a_{69}$	6	3	5	*	9
$a_{77}$	7	$a_{74}$	*	$a_{78}$	*	7	4	*	8	*
$a_{88}$	8	$a_{85}$	$a_{87}$	$a_{89}$	*	8	5	7	9	*
$a_{99}$	9	$a_{96}$	$a_{98}$	*	*	9	6	8	*	*

図 3-9 熱回路網行列向き疎行列表現 CSG



$$\text{data} = \begin{bmatrix} * & * & a_{11} & * & * \\ * & a_{21} & a_{22} & a_{12} & * \\ * & a_{32} & a_{33} & a_{23} & * \\ a_{41} & * & a_{44} & * & a_{14} \\ a_{52} & a_{54} & a_{55} & a_{45} & a_{25} \\ a_{63} & a_{65} & a_{66} & a_{56} & a_{36} \\ a_{74} & * & a_{77} & * & a_{47} \\ a_{85} & a_{87} & a_{88} & a_{78} & a_{58} \\ a_{96} & a_{98} & a_{99} & a_{89} & a_{96} \end{bmatrix}$$

$$\text{Offset} = [-3 \quad -1 \quad 0 \quad 1 \quad 3]$$

図 3-10 DIA で表現する熱回路網行列

CSR より対角優位疎行列のベクトル積を高速に処理できる DIA 形式 [35]の先例があるので、それが一般的な対角優位構造に対する疎行列の計算を高速化できるが、CSR と同様に疎行列での位置と熱回路網モデルでの頂点位置の変換処理が必要となるため、提案する CSG に比較すると熱回路網モデルでの頂点位置の変換処理を減らすことができる。

具体例を示すと、式(3.9)の行列  $A$  は DIA 形式で表現すると、図 3-10 のようになる。対角成分および対角成分と平行（すなわち斜め）に位置する変数をいずれも、垂直に配置し、対角成分をオフセット 0，そこから右にずれる距離をプラス，左にずれる距離をマイナスのオフセット値で表現する。このような表現によりコンパクトな表現が可能となる。

反復計算は式(3.13)の計算が必要なので、 $x_j^{(k-1)}$ と $b_i$ の参照を効率よく行うことが必要である。DIA では、各頂点 $(i, j)$ から $x_j^{(k-1)}$ と $b_i$ の参照は以下の手順で行うことになる。回路の各頂点 $(i, j)$ に対応した行列のノード番号は $n = iN + j$ である。 $n$ の計算は行列演算の前に実行される。ノード  $n$  に隣接する非対角成分の温度値 $x_*^{(k-1)}$ と隣接ノード番号の参照は、 $(i - 1) \times N + j$ ,  $(i + 1) \times N + j$ ,  $i \times N + j - 1$ ,  $i \times N + j + 1$ の計算で実施できる。したがって、乗算 4 回，加算 6 回，減算 2 回，共有メモリへの参照が 10 回の手数が必要となる。

これに対して，CSG での検索手順は，図 3-11 になる。回路の各頂点 $(i, j)$ に対応した行列のノード番号は $n = iN + j$ なので，対角成分 $d_n$ と $b_n$ へは直接参照できる。（図 3-11 中①）同様にノード $n$ に隣接する非対角成分の R\_value の $a_{n,*}$  および隣接ノード番号 R\_position も直接参照できる。（図 3-11 中②）さらに，接続するノードの温度値 $x_*^{(k-1)}$ は，R\_position から参照できる。（図 3-11 中③）以上により， $x_j^{(k-1)}$ と $b_i$ の参照に必要な手数は，共有メモリへの参照が 14 回である。以上のことから， $x_j^{(k-1)}$ と $b_i$ の参照を約 63%の時間で実行できるため，全体の処理時間もこの割合で減らすことが期待できる。

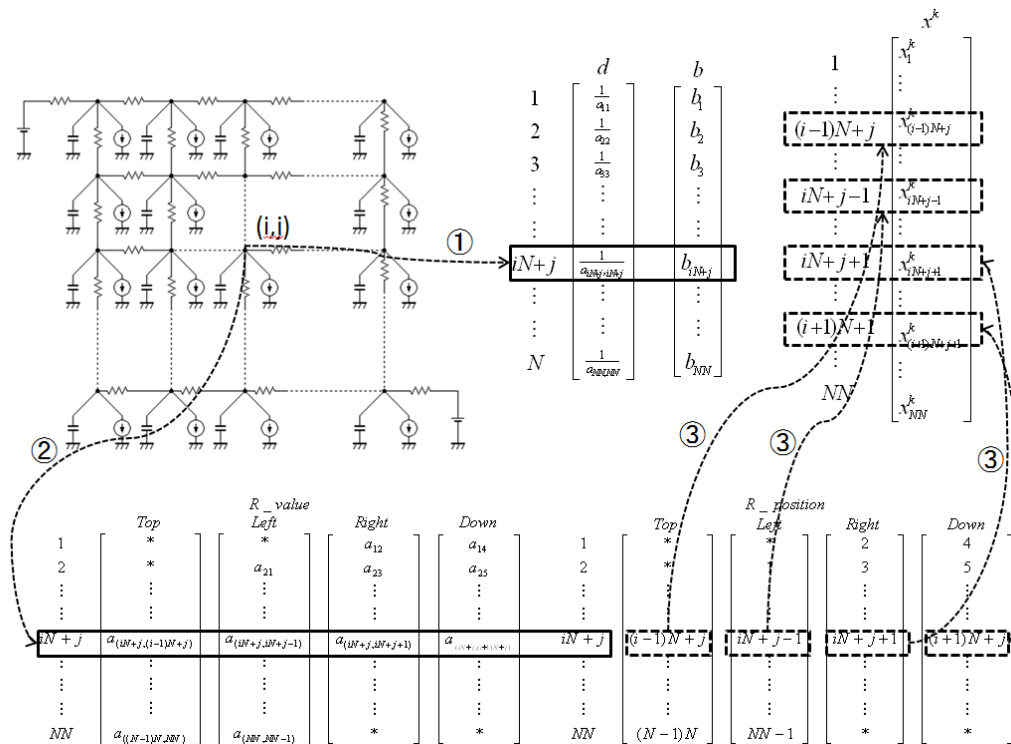


図 3-11 CSG 法のメモリ対応関係

具体的な回路シミュレーションの処理の流れを図 3-12 に示す．はじめに，熱回路網モデルを構築する．次に示した CSG により回路を表現する疎行列を圧縮する．次に，温度の初期分布（ヤコビ法で用いる初期解  $x^0$ ）を生成する．これらのデータを元に， $\Delta t$ 後の頂点温度  $T(t + \Delta t)$ を導く．このときの演算を GPU に実装することで処理の高速化を図る．実装には，NVIDIA 社から提供されている開発環境 CUDA(Compute Unified Device Architecture)を用いる．各頂点の  $x$  座標を CUDA スレッドブロック内のスレッド `theadIdx` の番号として設定し， $y$  座標を CUDA スレッドブロック `BlockIdx` の番号として設定することで，並列処理の領域を分割した．

GPU に実装する部分は図 3-13 で示される．この段階では，初期解  $x^0$  が与えられている．その後，式(3.12)を用いて  $x^{(k-1)}$  から  $x^k$  を導出する手続き(Xnew Calculation Kernal)と，その結果から式(3.13)を用いて算出誤差の総和を求める手続き(Error Calculation Kernal)を，式(3.13)の最大許容誤差  $\epsilon$  より小さくなるまで順次繰り返す．これらの手続きの中で， $x^k$  を保存する Xnew, CSG を表現する D, R\_value, R\_position, b はグローバルメモリに保存する．これらのデータは，GPU が処理を行うときに CPU のホストメモリからグローバルメモリに書き込む．また，各カーネルは，グローバルメモリから CSG のデータを取りこみ，内部の計算は共有メモリを使用する．

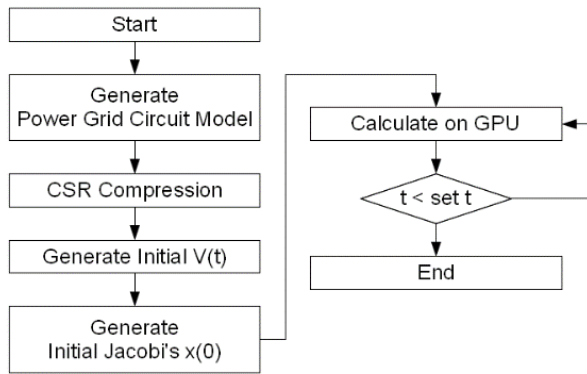


図 3-12 回路シミュレーションの流れ

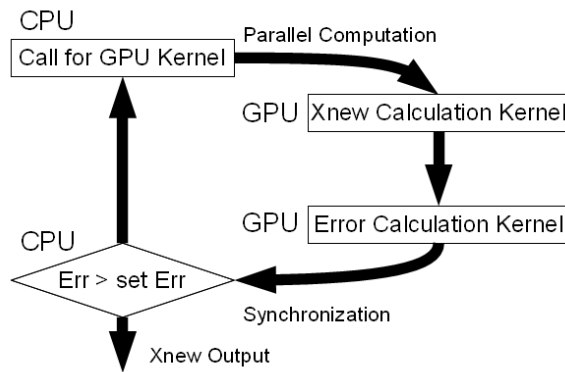


図 3-13 ヤコビ法実装アルゴリズム

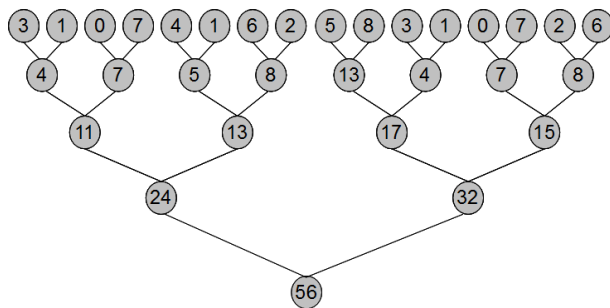


図 3-14 二分木による加算処理

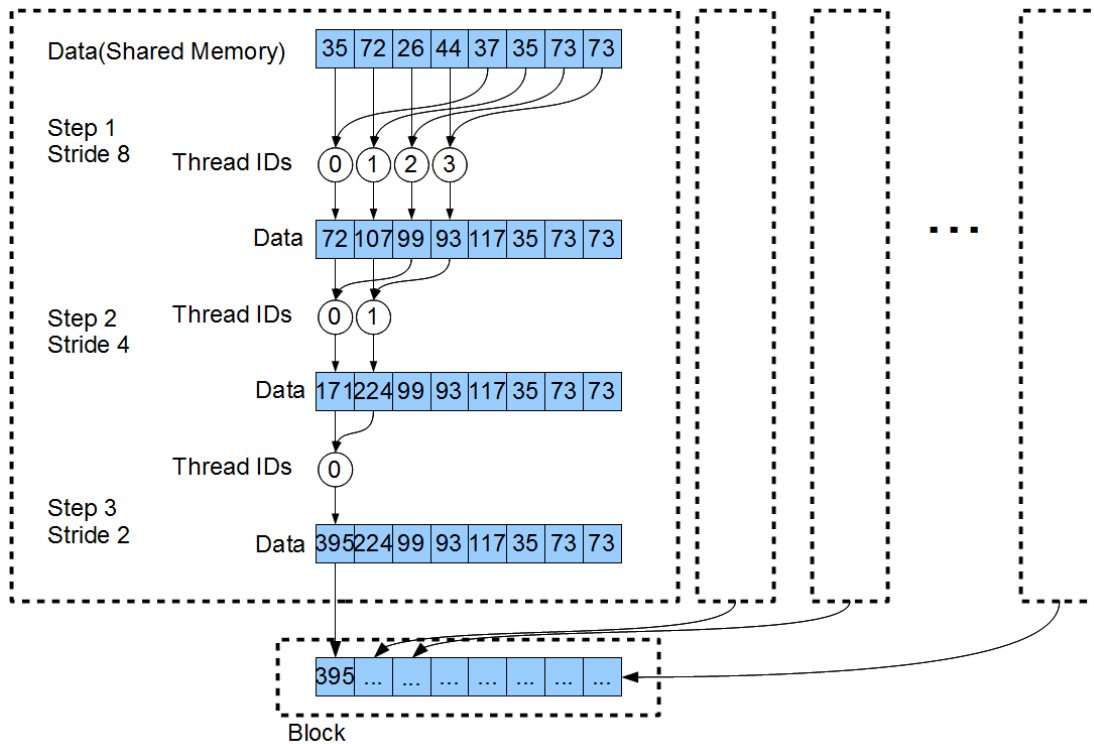


図 3-15 二分木による加算の GPU 実装例

ヤコビ法の反復判定のため、式(3.13)の $x_i^{(k-1)}$ と $x_i^{(k)}$ の差の和を求める必要がある。この計算を GPU で効率よく行うため、一般に図 3-14 に示すような 2 分木構造の処理順で加算を行う方法が知られている。十分な並列処理コアが存在する場合、要素数 $N$ に対して、計算オーダーを $O(N)$ から $O(\log_2 N)$ に減らすことが出来る。大規模なデータの加算は CUDA の処理単位であるブロックに分割して処理される。2 分木の加算はリーフからルートに向かって順に 2 個ずつの加算が行われる。計算の処理順に矛盾しないように、リーフからの距離の近いものを最初のブロックに割り当て、順次、計算順序に矛盾しないように別のブロックに割り当てる。GPU は各ブロック間でデータ移動ができないので、図 3-15 により、算出関数の第 1 ステップでは $x_i^{(k-1)}$ と $x_i^{(k)}$ のそれぞれに関して、各ブロックで差の絶対値を求め、それぞれを共有メモリに渡して二分木シグマ計算を行い、その結果をグローバルメモリに格納する。次に第 2 ステップでは各ブロックで計算した結果をグローバルメモリから一つのブロックに移動した後、第 1 ステップと同様に、それぞれを共有メモリに渡して二分木シグマ計算を行う。以上により、すべての差の絶対値の和を求める。

高速をしめすために、以下のように GPU を用いた CUDA プログラミングのシミュレーションの評価を行う。動作環境を表 3-2 で示す。時間刻み幅 $\Delta t$ を 0.1 [ns]とし、終了時刻を 1 [us]と設定し、解析演算を 10000 回のタイムステップ分繰り返した。

表 3-2 動作環境

	GTX260	GTX580	E8400
OS	Windows 7 Professional(32bit)		
CUDA Cores	216	512	-
Clock(GHz)	1.36	1.54	3.00
Memory Bus Width(Bit)	448	384	64
Memory Clock rate(MHz)	1000	2004	800
Coding Environment	CUDA4.0		Visual Studio 2008

注) GTX = GeForce GTX  
E8400 = Core2Duo E8400

CPU では、GPU と同様のアルゴリズムを用い、速度検証を GPU と比較した。

実験は回路サイズの順に  $32 \times 32$ ,  $64 \times 64$ ,  $128 \times 128$ ,  $256 \times 256$ ,  $512 \times 512$  に対して行った。  $n \times n$  の回路サイズに対して、スレッドブロック数とスレッドブロック内のスレッド数は、それぞれ  $n$  に設定した。CPU に対する速度向上のグラフを図 3-16 に示す。縦軸に速度比率、横軸に回路サイズをとる。速度比率は、CPU の解析時間 (CPU time) を GPU の解析時間 (GPU time) で割ったものである。GTX260 で約 34.9 倍、GTX580 で最大約 74.0 倍の高速化を達成した。上述の計算速度の上界に対して、約半分程度の性能を示し、ほぼ妥当な結果であるといえる。次に、図 3-17 に Visual Profiler を用いて、GPU(GTX260) での処理時間の内容を分析したものを示す。図 3-17 から、Xnew Calculation, Error Calculation 等のそれぞれの処理時間比率が読み取れる。これらの比率は、アルゴリズムから予想できるものになっており妥当である。白い部分は、CPU-GPU 間の通信による時間である。更なる速度向上の可能性を探るとすれば、この通信時間の削減や CPU-GPU 間のデータ転送の削減などの工夫が必要であろう。

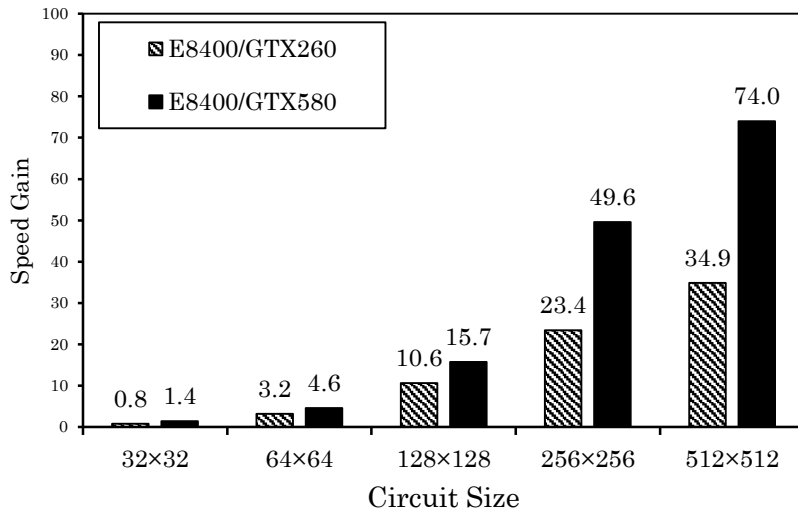


図 3-16 CPU に対する速度向上

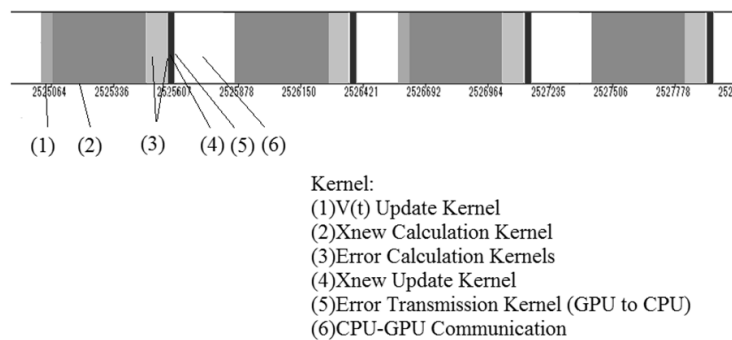


図 3-17 Visual Profiler による計算時間の分析

次に、解析手法による GPU 計算速度の違いについて述べる。Feng ら [36]はマルチグリッド法を GeForce8800 Ultra GPU を使って、500 Relaxation Time の計算を実施した時間を示している。グリッドサイズ 127K に対して、CPU で 2.4(s)、GPU で 0.24(s)、グリッドサイズ 851K に対して、CPU で 16.1(s)、GPU で 0.80(s) であった。磯田ら [7]は、前進オイラー法を実装し、グリッドサイズ 262K(512×512)に対して、GTX260 で、2.5(s)であった。これらの結果と、本手法の計算時間の比較を表 4 にまとめる。表中の「BE 法」は本手法の後退オイラー法による計算時間である。「FE 法」は [7]の前進オイラー法による結果である。いずれも GPU は GTX260 による結果である。「MG 法」は [36]のマルチグリッド法の結果、「MG 法\*」は、比較のため、使用 GPU の性能差を補正したものである。

前進オイラー法と本提案の後退オイラー法は、いずれも刻み幅、解析演算のタイムステッ

プは同じに設定し、直接比較を行った。後退オイラー法は、1 個の時間刻み幅  $\Delta t$  の解析に対して、小規模回路では 3 回、大規模回路では 2 回の Relaxation の計算を要した。10000 回のタイムステップに対して 30000 回の Relaxation 計算を行ったことに相当する。マルチグリッド法 [36] では 500 Relaxation Time の結果が示されているが、本手法と同じ Relaxation 回数に基準を合わせるためには、同文中に書かれている計算時間を 60 倍すればよいので、その値を表の「MG 法」に記載した。マルチグリッド法は、詳細のシミュレーションは後退オイラー法を用いているので、時間刻み幅の条件を合わせた場合、Relaxation に要する回数は同等であると予想できるためである。

さらに、文献 [36] は GeForce8800 Ultra で実装が行われているが、これは、クロック周波数 1.5GHz、コア数が 128 個である。それに対して、「BE 法」 [7]、「FE 法」で用いた GPU(GTX260) は 1.3GHz、コア数 196 個である。いずれも、アクセス速度が同じ 2G バイトのメモリを搭載しているので、メモリアクセスによる差が生じないとすると、使用 GPU の最高性能比は GeForce8800 Ultra が GTX260 に対して約 75% である。表 3-2 の「MG 法\*」は、「MG 法」に対して、この補正を行ったものである。

これら比較結果から、本提案の後退オイラー法は前進オイラー法の約 5 倍程度の計算時間で実現されており、後退オイラー法が前進オイラー法に比べて、高精度で発散しにくい点を考えると、有効性の高い結果であるといえる。また、他の後退オイラー法との比較では、[36] のマルチグリッド法に比較して明らかな優位性を示すことができた。

表 3-3 GPU 計算時間(s)の手法による違い

Grid Size	16K	64K	127K	256K	851K
BE 法	2.8	4.5	8.0	13.3	(33.9)
FE 法	-	1.1	1.5	2.5	7.6
MG 法*			10.8		36.0
MG 法	-	-	14.4	-	48.0

\*は GPU の性能差の補正を行った

### 3.4 結言

本論文では、熱回路を用いた蓄電池の熱挙動を解析する。熱回路網法は物体の伝熱を表現する方法である。熱回路モデルの規模は大きいので、並列計算を行わないと、膨大な時間がある。その問題を解決するために、GPGPU を用いた電気回路の並列計算を転用して、熱回路モデルの解析の高速化手法を提案する。熱回路の特徴を活かした疎行列の新しい表現方法 CSG と、それを用いた、ヤコビ法による後退オイラー法を GPU に実装し評価した。高速化率は約 74 倍のスピードアップが実現した。また、使用メモリ領域も、CSG を用いることで大幅に削減することが出来た。





## 第4章 内部パラメータの推定

本章では、蓄電池の OCV と電極表面抵抗の同定を目的とし、SOC と OCV の関係に着目することにより、端子電圧と端子電流の比率に基づき最適な忘却係数を算出する発見的な手法を説明する。

### 4.1 内部パラメータの推定のための蓄電池モデル

前述したように、本文で扱うシステム同定は蓄電池管理において重要な残量予測や劣化診断を目的とする。蓄電池の劣化は蓄電池容量の減少（負極表面の SEI 形成）と電極表面の内部抵抗の増大が代表的なものである [38]。一般にリチウムイオン蓄電池の内部抵抗は、溶液抵抗、電極表面抵抗、電極活物質内部の拡散抵抗に分離され、それぞれ、ミリ秒以下、ミリ秒から数 10 ミリ秒、分～時間オーダーの時定数を持つ。また、内部抵抗変動のなかで正極表面の増加が占める割合が支配的であるため、1Hz のサンプリング周波数に対する反応を見ることにより、OCV と正極表面抵抗の変動が観測できる。このことより、図 2-2 のようにモデルの高周波部分を一つの RC 回路に合成し、図 4-1 に示す等価回路モデルを使用する。 $R_a$  は図 2-2 の溶液抵抗  $R_0$  と電気二重層 ( $R_1, C_1$ ) の合成抵抗を表す。 $R_b$  と  $C_b$  は電極内部の拡散現象を表す抵抗  $R_2$  とコンデンサ  $C_2$  である。システム同定の対象は、OCV と  $R_a$  である。本論文では、これら OCV と  $R_a$  の推定方法とその評価を主な目的とする。尚、リチウムイオン蓄電池の劣化モードには、上記以外にデンドライトの形成による劣化が含まれる。これは、負極表面のインピーダンス変化を正極表面の評価と分離して観測することにより解明できる、すなわち、より詳細化した等価回路と高い周波数でのサンプリングが必要であるが、本論文では対象外とする。

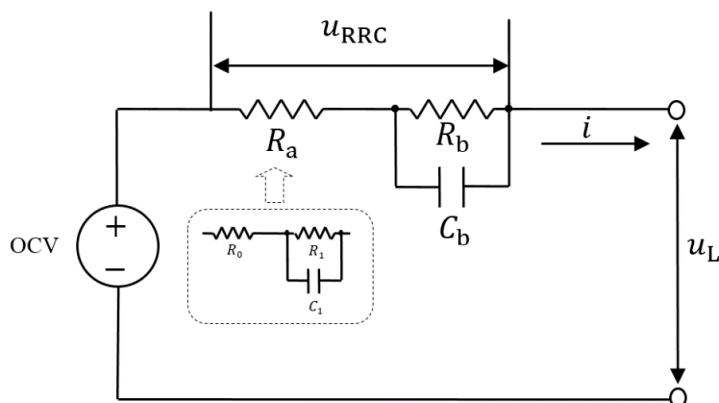


図 4-1 蓄電池同定に使う等価回路モデル

本節では、OCV を電圧源で表現し、蓄電池内部パラメータ同定モデルを検討する。図 4-1 中の OCV の値を  $u_{OCV}$ 、電池の端子電流を  $i$ 、RC 回路 ( $R_a$ ,  $R_b$ ,  $C_b$ ) の両端電圧を  $u_{RRC}$  とする。等価回路モデルの、 $R_b$  および  $C_b$  の両端電圧を  $u_b$ 、電流  $i$  が  $R_b$  および  $C_b$  に分流される電流をそれぞれ、 $i_{R_b}$ 、 $i_{C_b}$  とすると、 $u_b = u_{RRC} - iR_a$ 、 $i_{C_b} = C_b \frac{du_b}{dt}$ 、 $i_{R_b} = i - i_{C_b}$  で与えられる。これを自明な式、 $u_{RRC} = iR_a + i_{R_b}R_b$  に適用すると  $u_{RRC}$  と  $i$  に関する以下の微分方程式が導出できる。

$$u_{RRC} = (R_a + R_b)i + R_a R_b C_b \frac{di}{dt} - R_b C_b \frac{du_{RRC}}{dt} \quad (4.1)$$

ここで、前進オイラー法で差分化すると、式(4.1)中の  $u_{RRC}$ 、 $i$  をそれぞれ  $u_{RRC}(k-1)$ 、

$i(k-1)$  とおき、 $\frac{du_{RRC}}{dt} \cong \frac{u_{RRC}(k) - u_{RRC}(k-1)}{T_s}$ 、 $\frac{di}{dt} \cong \frac{i(k) - i(k-1)}{T_s}$  を適用すれば、

$$\begin{aligned} u_{RRC}(k-1) &= (R_a + R_b)i(k-1) + R_a R_b C_b \frac{i(k) - i(k-1)}{T_s} \\ &\quad - R_b C_b \frac{u_{RRC}(k) - u_{RRC}(k-1)}{T_s} \end{aligned} \quad (4.2)$$

を得る。これを整理すると次式を得る。

$$u_{RRC}(k) = R_a \cdot i(k) + \left( \frac{T_s R_a}{R_b C_b} + \frac{T_s}{C_b} - R_a \right) i(k-1) - \left( \frac{T_s}{R_b C_b} - 1 \right) u_{RRC}(k-1) \quad (4.3)$$

式(4.3)に  $u_L(k)$  の関係式  $u_L(k) = u_{RRC}(k) + u_{OCV}(k)$  を代入すると次式を得る。

$$u_L(k) = R_a i(k) + \left( \frac{T_s R_a}{R_b C_b} + \frac{T_s}{C_b} - R_a \right) i(k-1) - \left( \frac{T_s}{R_b C_b} - 1 \right) u_{RRC}(k-1) + u_{OCV}(k) \quad (4.4)$$

さらに、 $u_{RRC}(k-1) = u_L(k-1) - u_{OCV}(k)$  を代入すると、

$$\begin{aligned} u_L(k) &= R_a i(k) + \left( \frac{T_s R_a}{R_b C_b} + \frac{T_s}{C_b} - R_a \right) i(k-1) + \left( \frac{T_s}{R_b C_b} - 1 \right) u_{OCV}(k) \\ &\quad - \left( \frac{T_s}{R_b C_b} - 1 \right) u_L(k-1) + u_{OCV}(k) \end{aligned} \quad (4.5)$$

を得る。システム同定するために、

$$b_0 = R_a, \quad b_1 = \frac{T_s R_a}{R_b C_b} + \frac{T_s}{C_b} - R_a, \quad (4.6)$$

$$a_1 = \frac{T_s}{R_b C_b} - 1, \quad f = (1 + a_1)u_{OCV}$$

式(4.6)のように変数変換し、観測ノイズ  $w(k)$  の項を加えて上式を書き換えることにより、次式を得る [39].

$$y(k) = u_L(k) = \boldsymbol{\varphi}^T(k)\boldsymbol{\theta} + w(k), \quad \boldsymbol{\varphi}(k) = \begin{bmatrix} i(k) \\ i(k-1) \\ -u_L(k-1) \\ 1 \end{bmatrix}, \quad \boldsymbol{\theta} = \begin{bmatrix} b_0 \\ b_1 \\ a_1 \\ f \end{bmatrix} \quad (4.7)$$

式(4.7)において、パラメータ $\boldsymbol{\theta} = [b_0 \ b_1 \ a_1 \ f]^T$ は未知パラメータとして扱う。そうすると、回帰式(4.7)は線形回帰式である。蓄電池の充電放電により $u_{OCV}$ や $R_a$ ,  $R_b$ ,  $C_b$ 等の内部パラメータは時間変化するの、 $\boldsymbol{\theta}$ は時変パラメータである。本文では忘却係数付き逐次最小 2 乗法を用いてパラメータ $\boldsymbol{\theta}$ を推定する。パラメータ $\boldsymbol{\theta}$ の値が得られれば、 $R_a$ ,  $R_b$ ,  $C_b$ ,  $u_{OCV}$ は次式を使って逆算することができる。

$$R_a = b_0, \quad R_b = \frac{b_1 - a_1 b_0}{1 + a_1}, \quad C_b = \frac{T_s}{b_1 - a_1 b_0}, \quad u_{OCV} = \frac{f}{1 + a_1} \quad (4.8)$$

線形回帰式(4.7)において、 $f = (1 + a_1)u_{OCV}$ は、開回路電圧 $u_{OCV}$ を含んでいる。 $u_{OCV}$ は SOC に依存して変化する物理変数であるが、本論文では $u_{OCV}$ を時変パラメータとみなし、忘却係数付き逐次最小 2 乗法を用いて推定する。なお、式(4.8)により $u_{OCV}$  が得られれば、OCV-SOC 曲線から SOC の推定値を得ることができる。

## 4.2 蓄電池内部パラメータ推定

システム同定による蓄電池内部パラメータ推定のブロック線図を図 4-2 に示す [40]。同図の適応機構の目的は、同じ入力 $i(k)$ を現実システム（実際の蓄電池）の出力 $u_L(k)$ と蓄電池数学モデルの予測値 $\hat{u}_L(k)$ の各時刻誤差 $e(k)$ の二乗総和を最小化するように蓄電池等価回路モデルの最適なパラメータを求める(システム同定)ことである。

蓄電池内部パラメータは、温度や、SOC、電流、劣化などにより変動する。劣化による内部パラメータの変動を知ることは本研究の最終目的ではあるが、劣化による変動は数日～数か月の長期間の化学反応であるため、長期間での複数回のシステム同定によるパラメータ推定結果の変化により把握することが可能である。本論文では、最大 1 放電サイクルの時間範囲で同定手法と精度検証を主旨として述べる。前述のように劣化速度が遅いので、1 放電サイクルにおいては、内部抵抗の変化が少なく無視できる程度である。同定精度に大きな影響を与える要因は SOC の変化による OCV 電圧変化しかない。それに対する対策は本文の着目点である。

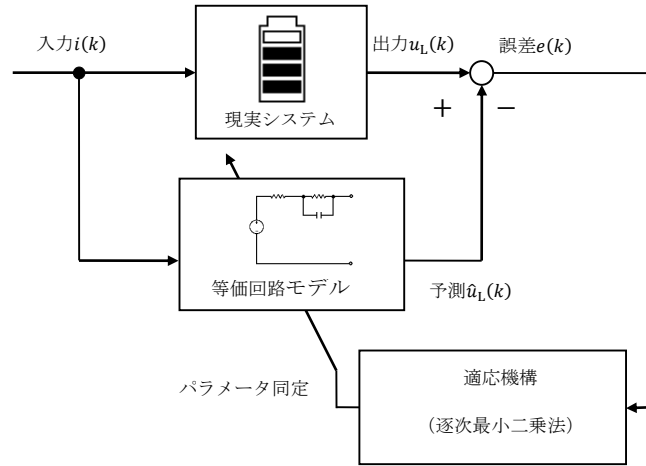


図 4-2 蓄電池内部パラメータ同定のブロック図

本論文では、ほぼ一定の平均電流で放電する場合を想定し、時間の経過とともに、SOC がほぼ一定割合で減少することを条件とする。また、SOC と OCV および内部抵抗の関係式は未知であるとする。そうすると、時間変化に伴い SOC が変動し、SOC との関係が未知である OCV と内部抵抗を、観測可能な出力電圧と出力電流から、等価回路に基づいてシステム同定する問題として定式化される。

この問題は、同定対象の OCV と内部抵抗が時変パラメータと定義できる。そのため、過去のデータを忘却する方法が有効であるため、忘却係数付き逐次最小二乗法（Recursive Least-Squares method）を用いる。前述の図 4-2 における適応機構（逐次最小二乗法）の評価関数は式(4.9)で示される [41]。式(4.9)では、 $\lambda$ は忘却係数である。ここで、忘却係数 $\lambda$ は 1 以下の正数であり、 $\lambda = 1$ のとき通常最小二乗法に帰着される。

$$J(k) = \sum_{i=1}^k \lambda^{k-i} e^2(i) \quad (4.9)$$

リチウムイオン蓄電池の OCV-SOC 曲線において、SOC に対する OCV の変化率は一定ではないと放電電流による蓄電池の端子電圧 $u_L$ の変化量も違うため、忘却係数を一定に設定するのは不適切である。そのため、忘却係数を、以下の評価関数を用いて適応調整する。各時刻の忘却係数を設定する評価式を構築する。式(4.10)では、 $\lambda(i)$ は $i$ 時刻における忘却係数である [42]。

$$J(k) = \sum_{j=1}^{k-1} \left( \prod_{i=j+1}^k \lambda(i) \right) e^2(j) + e^2(k) \quad (4.10)$$

式(4.10)の評価関数を最小化する適応機構（逐次最小 2 乗法）のアルゴリズムを以下に示す。 $\hat{\boldsymbol{\theta}}(k)$ は時刻 $k$ におけるパラメータの推定値、 $\mathbf{P}(k)$ は時刻 $k$ における誤差共分散行列、 $\mathbf{I}$ は単位行列である。パラメータの初期値 $\hat{\boldsymbol{\theta}}(0)$ 、および、誤差共分散行列の係数 $\boldsymbol{\gamma}$ の設定は、 $\hat{\boldsymbol{\theta}}(0) = [0 \ 0 \ 0 \ 0]^T$ 、 $\boldsymbol{\gamma} = 10^2$ とする。

初期値：

$$\hat{\boldsymbol{\theta}}(0) = \mathbf{0}$$

$$\mathbf{P}(0) = \gamma \mathbf{I} \quad \gamma \text{は大きな正数}$$

$$0 \ll \lambda(k) < 1 \quad \text{忘却係数}$$

逐次処理：

$$\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \frac{\mathbf{P}(k-1)\boldsymbol{\varphi}(k)}{\lambda(k) + \boldsymbol{\varphi}^\top(k)\mathbf{P}(k-1)\boldsymbol{\varphi}(k)} e(k) \quad (4.11)$$

$$e(k) = y(k) - \boldsymbol{\varphi}^\top(k)\hat{\boldsymbol{\theta}}(k-1) \quad (4.12)$$

$$\mathbf{P}(k) = \frac{1}{\lambda(k)} \left\{ \mathbf{P}(k-1) - \frac{\mathbf{P}(k-1)\boldsymbol{\varphi}(k)\boldsymbol{\varphi}^\top(k)\mathbf{P}(k-1)}{\lambda(k) + \boldsymbol{\varphi}^\top(k)\mathbf{P}(k-1)\boldsymbol{\varphi}(k)} \right\} \quad (4.13)$$

本論文では端子電圧 $u_L$ と一定時間内の平均電流の比率により忘却係数を適応的に調整するアルゴリズムを提案する。SOCを直接測定することはできないが、SOCの時間微分が電流 $i$ に正比例する、すなわち

$$\frac{d\text{SOC}}{dt} \propto i \quad (4.14)$$

が成り立つので、

$$\frac{du_{\text{OCV}}}{d\text{SOC}} = \frac{du_{\text{OCV}}}{dt} / \frac{d\text{SOC}}{dt} \propto \frac{d\text{SOC}}{dt} / i \quad (4.15)$$

を得る。一方、前進オイラー法近似によれば、一定のサンプリング周期 $T_s$ の下で、

$$\frac{du_{\text{OCV}}(k)}{dt} \cong \frac{u_L(k) - u_L(k-1)}{T_s} \quad (4.16)$$

が成り立つ。ただし、( $R_b$ ,  $C_b$ )回路部分の応答は収束(時定数)が速いので、これを無視して $du_{\text{OCV}}/dt \cong du_L/dt$ と近似した。

したがって、上式と式(4.15)を合わせれば

$$\frac{du_{\text{OCV}}(k)}{d\text{SOC}} \propto \frac{u_L(k) - u_L(k-1)}{i(k)} \quad (4.17)$$

となり、 $u_L(k) - u_L(k-1)$ と $i(k)$ を測定すれば、 $du_{\text{OCV}}/d\text{SOC}$ を評価できることになる。ただし、測定ノイズの影響を除去するために、以下の方法で移動平均をとる。

$$u_L(k) - u_L(k-1) \cong \frac{1}{M-1} \sum_{i=k}^{k-M+1} u_L(i) - u_L(i-1) \quad (4.18)$$

$$i(k) \cong \frac{1}{M} \sum_{i=k}^{k-M+1} i(i) \quad (4.19)$$

式(4.18), (4.19)の左辺をそれぞれ $m_{u_L}(k)$ ,  $i_{\text{avg}}(k)$ と定義する。 $m_{u_L}$ は長さMのFIFOバッファに保存されている端子保存されている電流値電圧 $u_L$ の平均変化、 $i_{\text{avg}}$ は長さMのFIFOバッファに $i$ の平均値である。

以上の準備の下で、忘却係数 $\lambda$ の調整側を式(4.20)のように提案する。ここで、 $G$ は調整ゲイン（定数）である。SOCに対して $u_{OCV}$ の変化が大きいときには忘却係数 $\lambda(k)$ を小さくし、 $u_{OCV}$ の変化が小さいときには $\lambda(k)$ を大きくすることを表している。

$$\lambda(k) = 1 - G \left| \frac{m_{u_L}(k)}{i_{avg}(k)} \right| \quad (4.20)$$

逐次最小二乗法で計算した後、パラメータ計算エラーを除去する作業を行う必要がある。まずは分母検査、最小二乗法で算出した結果をパラメータの逆算式(4.8)に代入すると、分母は非常に小さい値になる恐れがあるので、分母のチェックが必要である。分母のチェック式は以下になる。

$$\begin{aligned} 1 + a_1 &> 0.025, \\ b_1 - a_1 b_0 &> 0.001 \end{aligned} \quad (4.21)$$

分母検査の結果、式(4.21)が満たされれば、得られた $\hat{\theta}(k)$ を採用する。逆に、満たされなければ、 $\hat{\theta}(k)$ を棄却して、1ステップ前の推定値 $\hat{\theta}(k-1)$ を現時刻での推定値とする。このようにして得られた $\hat{\theta}(k)$ から、 $R_a(k)$ 、 $R_b(k)$ 、 $C_b(k)$ 、 $u_{OCV}(k)$ の推定値を式(8)により算出する。 $R_a(k)$ 、 $R_b(k)$ 、 $C_b(k)$ 、 $u_{OCV}(k)$ の推定値はノイズの影響で大きな動的変動を含んでいる。その動的変動を軽減するために、現時刻 $k$ から前の時刻 $k-63$ までの64ステップ分の移動平均を計算することにより平滑化処理を行う。たとえば $R_a(k)$ を平滑化した値 $\bar{R}_a(k)$ は式(22)で与えられる。 $R_b(k)$ 、 $C_b(k)$ 、 $u_{OCV}(k)$ についても同様である。

$$\bar{R}_a(k) = \frac{1}{64} \sum_{i=k-63}^k R_a(i) \quad (4.22)$$

以上をまとめると、蓄電池内部パラメータ推定アルゴリズムは、図4-3のフローチャートに示す通りとなる。

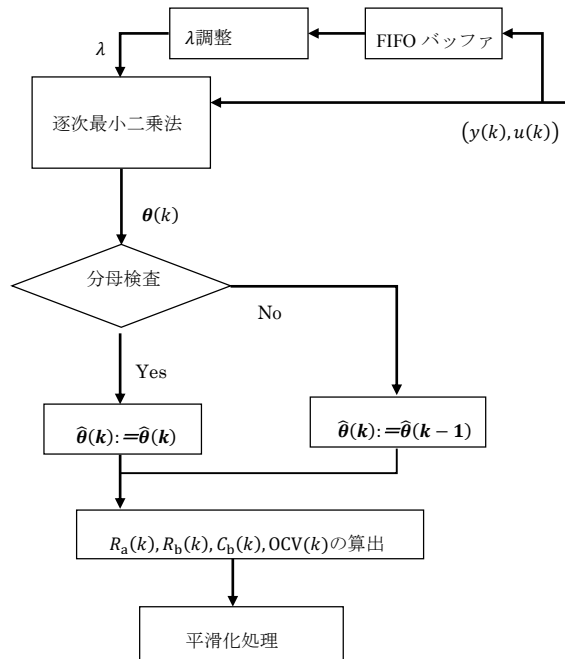
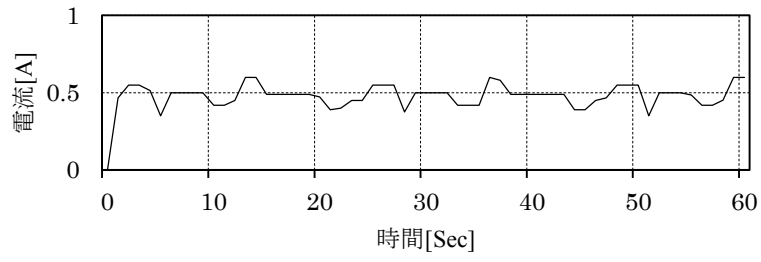


図 4-3 蓄電池パラメータ推定のフローチャート

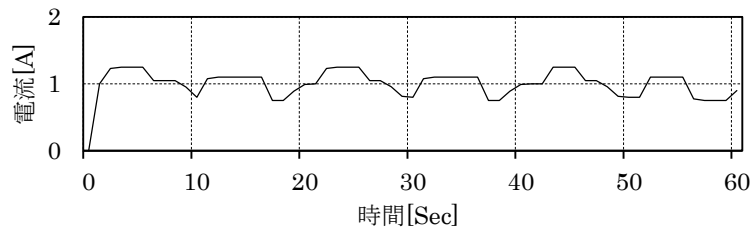
### 4.3 同定実験とその結果

本節では、図 4-4 のように疑似ランダム放電による蓄電池パラメータ推定を行う。実験では内部抵抗の放電電流による変化を調査するために、(a)0.5A 付近、(b)1A 付近、(c)2A 付近の疑似ランダム放電の放電パターンをそれぞれ用意する。環境温度は 15°C、25°C、35°C、45°C の 4 通りの温度設定でそれぞれの実験を行う。

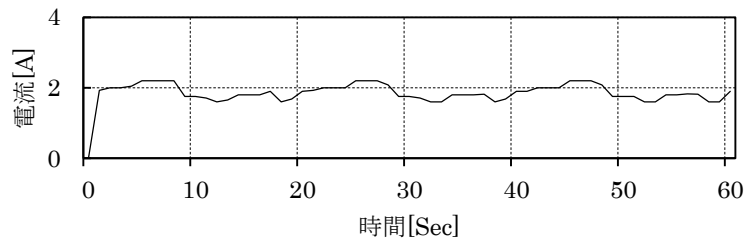
まず、実験により適切なゲイン  $G$  を検証する。実験は  $G$  を 500 から 1600 に設定して同定電流を 0.5A、1A、2A の 3 通りのパターンで行う。図 4-5 は式(20)の忘却係数  $\lambda(k)$  を用いて OCV 推定を行い、前述の OCV-SOC 曲線により SOC を求め、その誤差を評価した結果である。SOC の真値は、実験装置により電流積算法で与えられるものを用いた。疑似ランダム放電実験によると  $\frac{m_{uL}}{i_{avg}}$  の最大値は  $6.95 \times 10^{-4}$  であり、 $\lambda(k)$  が負にならない  $G$  の最大値は 1438 である。図 4-5 からわかるように電流値によらず SOC 推定誤差は  $G$  が 1100~1300 付近で最小となる。



(a)



(b)



(c)

図 4-4 疑似ランダム放電パターン

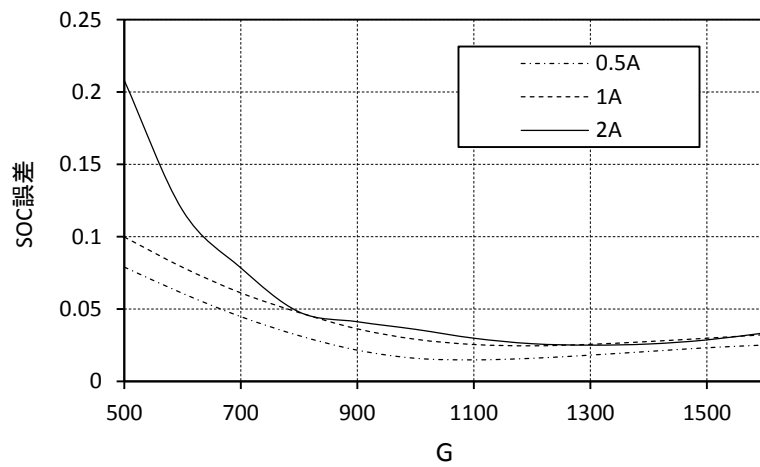


図 4-5 G の設定値に対する SOC 推定誤差



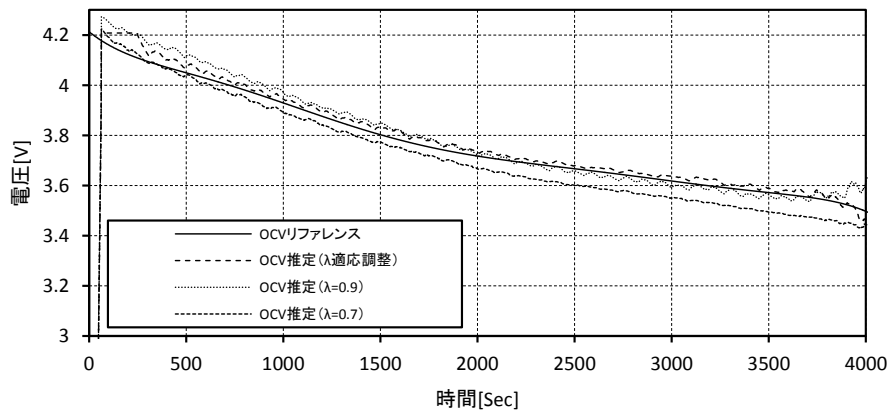


図 4-6 2A 疑似ランダム放電による OCV 推定結果 (環境温度 25°C)

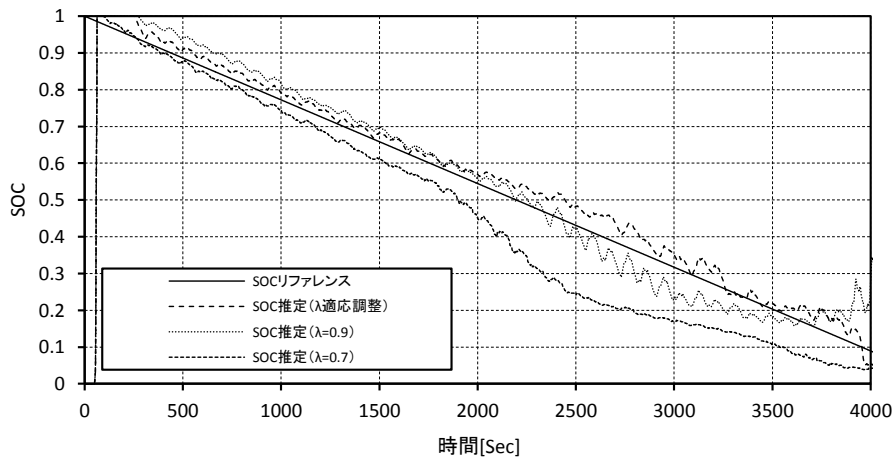


図 4-7 2A 疑似ランダム放電による SOC 推定結果 (環境温度 25°C)

図 4-6, 図 4-7 は  $\lambda$  固定と適応推定による環境温度 25°C で 2A 疑似ランダム放電パターンに対する  $u_{OCV}$  と SOC の推定結果である。SOC の推定値は前段落と同じ方法で OCV-SOC 曲線を使って求めた値である。  $\lambda = 0.9$  に固定すると、SOC が低いとき  $u_{OCV}$  の推定値が真値から大きく外れる。  $\lambda = 0.7$  に固定すると SOC が 0.7 より小さい領域での推定誤差が大きくなる。  $\lambda$  を式(20)により適応的に調整した場合の SOC 推定結果は、  $\lambda$  固定の場合よりも良好である。実験では、  $G$  の値は 1200 に設定した。

図 4-8 は 2A 放電の場合の忘却係数  $\lambda$  の時間変化を示す。図 4-8 を見ると、  $u_{OCV}$  の変化が小さい所 (2500~3500[Sec]) の忘却係数  $\lambda$  は大きい区間である。その原因は、OCV-SOC 曲線を見ると、SOC が 0.4 から 0.2 のところ、  $u_{OCV}$  が 3.7-3.6[V] 付近、SOC による  $u_{OCV}$  の変化率が一番小さい。忘却係数  $\lambda$  は大きい区間は 2500~3500[Sec] の付近である。

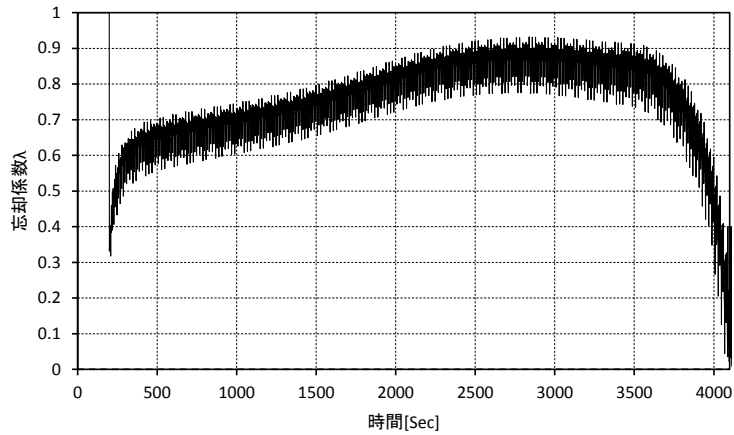


図 4-8 2A 疑似ランダム放電による $\lambda$ の時間変化

表 4-1 は各電流パターンと忘却係数に対する SOC 推定値の平均誤差である。電流が高くなると、SOC の推定誤差が高くなる。その原因は、電流が高くなると単位時間あたりの $u_{ocv}$ 変化が速くなる、 $\lambda$ の算出値は小さくなり、忘却速度は速くなる。

表 4-1 忘却係数による SOC 推定平均誤差

忘却係数 $\lambda$	適応調整	0.9	0.7
2A 疑似ランダム放電	0.026	0.061	0.073
1A 疑似ランダム放電	0.025	0.036	0.050
0.5A 疑似ランダム放電	0.016	0.035	0.039

温度と電流別の SOC 推定誤差（忘却係数 $\lambda$ 適応調整の場合）の平均値は表 4-2 に示す通りである。表 4-2 を見ると、平均推定誤差の最大値は 0.050 である。低温の場合では SOC の推定誤差は高温より大きくなる傾向がある。

表 4-2 温度と電流別 SOC 推定誤差平均値

	15°C	25°C	35°C	45°C
0.5A	0.033	0.016	0.012	0.017
1A	0.044	0.025	0.023	0.031
2A	0.037	0.026	0.046	0.050

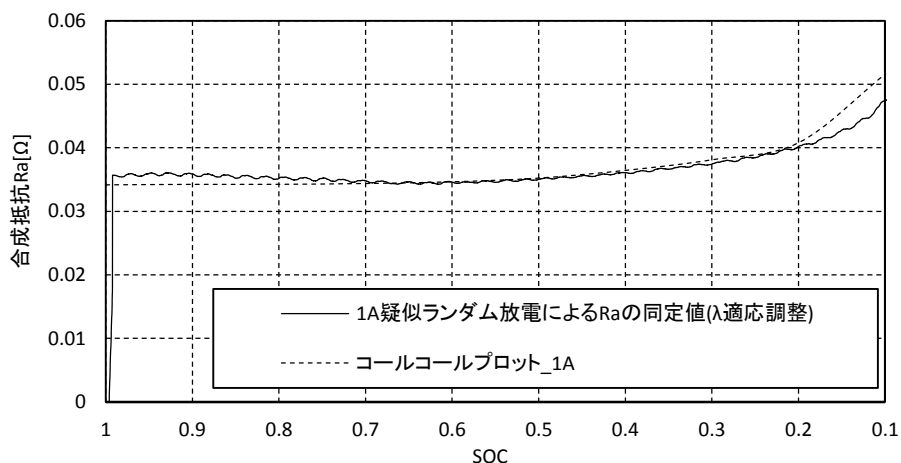


図 4-9 合成抵抗 $R_a$ コールコールプロットとの比較

表 4-3 合成抵抗 $R_a$ コールコールプロットとの誤差

15°C	25°C	35°C	45°C
2.99%	2.22%	4.96%	9.27%

$R_a$ の同定精度を比較ために、蓄電池のオフライン手法コールコールプロット（蓄電池のナイキスト特性測定）との比較を行う。コールコールプロットの実験結果は各温度において、 $\pm 8\%$ の交流成分を上乗せた平均値 1A の電流で放電させる状態で測定したインピーダンスである。測定した溶液抵抗 $R_0$ と電気二重層抵抗 $R_1$ の合成抵抗を同定手法で同定した $R_a$ と比較した。

図 4-9 は 25°Cの環境温度において 1A 疑似ランダム放電による $R_a$ の同定値（ $\lambda$ 適応調整）との比較である。表 4-3 は各温度合成抵抗 $R_a$ の推定値とコールコールプロットの比較結果である。高温（45°C）以外に 5%以内の推定誤差であり、良好な推定精度であることが分かる。

#### 4.4 結言

本章では、蓄電池の OCV と電極表面抵抗の同定を目的とし、SOC と OCV の関係に着目することにより、端子電圧と端子電流の比率に基づき最適な忘却係数を算出する発見的手法を説明する。最適な忘却係数を算出する方法による蓄電池の内部抵抗、OCV を同定することが可能となり、OCV から算出した SOC の推定精度は 5%以内を抑え、内部抵抗 $R_a$ の同定精度は 10%以内を抑えることが可能になる。

## 第5章 蓄電池残量推定

本章では、蓄電池の拡張カルマンフィルタを基づいた残量推定アルゴリズムを説明する。さらに、実験により、システムノイズと観測ノイズの最適値の設定方法の検討も行う。

蓄電池の残量 SOC (State of Charge) とは式(5.1)のように今残っている容量 $Q_R$ と電池の総容量 $FCC_{max}$ の割合である [3]。電池の総容量は蓄電池の放電可能な最大容量である。蓄電池容量の単位は Ah である。1[Ah]は  $3600[Sec] \times 1A = 3600[C]$  である。[C]は電荷量の単位クーロンである。放電可能な最大容量 $FCC_{max}$ は蓄電池を上限電圧から下限電圧まで最大の放電容量である。

$$SOC(T) = \frac{Q_R(T)}{FCC_{max}} \quad (5.1)$$

蓄電池の残量推定の従来法は三つある [43]。

電流積算法 [14]は電流を積算することにより電池の残量を算出する手法である、式(5.2)のようになる。電流積算法は高い予測精度を実現するが、初期状態の精度に高く依存する。

$$SOC(T) = SOC(T_0) + \frac{1}{FCC_{max}} \int_{T_0}^T i(t) dt \quad (5.2)$$

また、端子電圧法は充放電による電圧降下、分極の影響で高予測精度を実現することは難しい。図 5-1 は実験で使う 18650 の各放電レートの電圧変化図である。図 5-1 を見ると、放電レートによる電圧変化が違う、その原因は放電レートによる電池の分極の影響が違う。

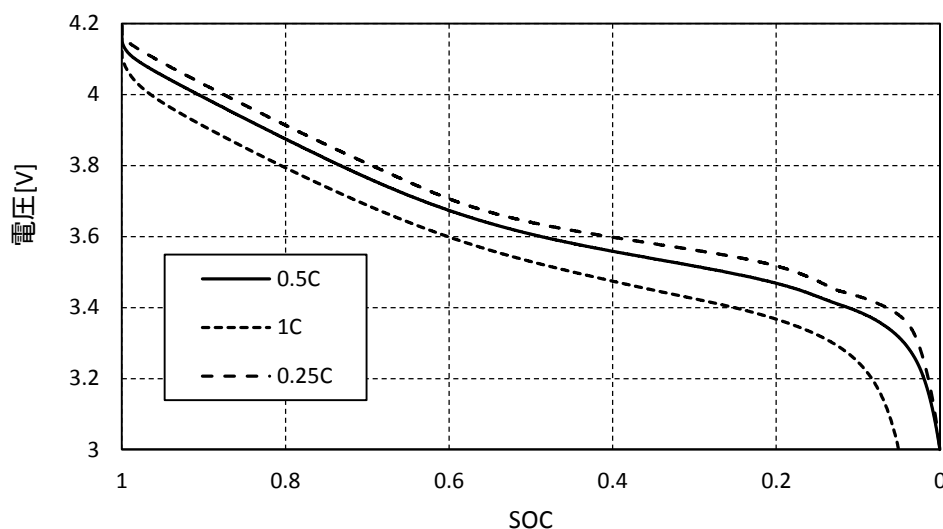


図 5-1 各放電レートの電圧変化図

開回路法 [13]は簡単であるが、電池は電気化学反応が平衡状態じゃないと、十分精度を出せない。蓄電池は充放電状態から電気化学反応平衡状態まで回復時間がかかる。つまり、蓄電池が充放電をすると、電圧を安定な状態に回復時間が長い。図 5-2 は蓄電池を一定の電流 (1C) を放電した後の電圧回復カーブである。カーブを見ると、電池の電圧が 2000 秒を待ちないと、電圧が安定しない。

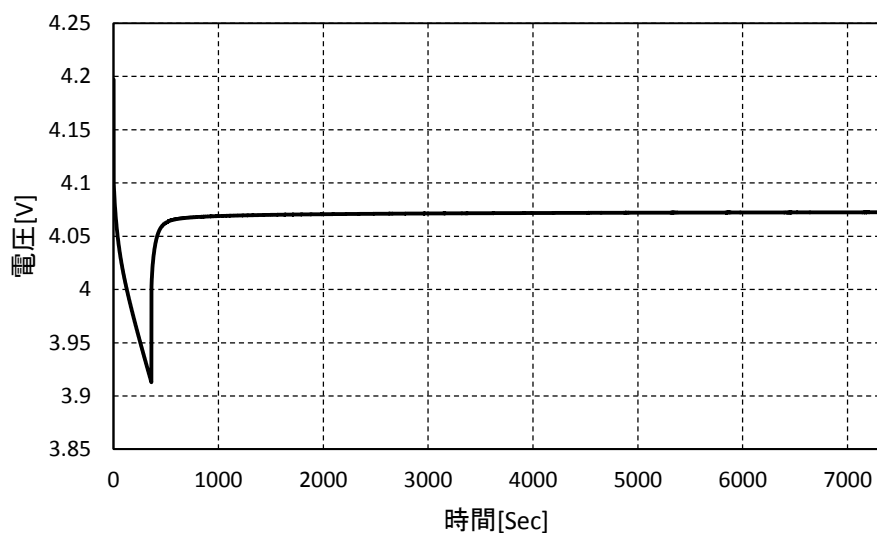


図 5-2 蓄電池を一定の電流 (1C) を放電した後の電圧回復カーブ

## 5.1 蓄電池残量推定アルゴリズム

蓄電池の等価回路モデルを第 2 章の図 2-2 に示す。同モデルは前に紹介された蓄電池モデルと同じ、内部抵抗、正極と負極の分極による過度現象を表す RC 回路、および、無負荷状態の蓄電池電圧を意味する開放電圧 OCV (Open Circuit Voltage) から構成される。同モデルの中で OCV は理想電源で模擬され、内部抵抗は抵抗素子  $R_0$  で模擬され、分極は 2 つの RC 回路 ( $R_1, C_1, R_2, C_2$ ) で模擬される。 $u_1, u_2$  はそれぞれの RC 回路間の電位差、 $u_L$  は蓄電池の端子間電圧、 $i$  は蓄電池から外部に流れる電流である。RC 回路の両端電圧  $u_1, u_2$  を微分方程式で表示すると、式 (5.3)、式 (5.4)、となる。式 (5.5) は蓄電池の端子間電圧と  $u_1, u_2$ 、電流  $i$ 、内部抵抗  $R_0$  で表現した式である。

$$C_1 \frac{du_1}{dt} + \frac{u_1}{R_1} = i \quad (5.3)$$

$$C_2 \frac{du_2}{dt} + \frac{u_2}{R_1} = i \quad (5.4)$$

$$u_L = u_1 + u_2 + iR_0 + u_{ocv} \quad (5.5)$$

蓄電池の場合は、蓄電池の端子電圧  $u_L$  と端子電流  $i$  しか観測できない。蓄電池の端子電圧  $u_L$  と端子電流  $i$  の観測による蓄電池の残量推定問題は状態空間モデルの内部状態推定である。そのために、蓄電池の等価回路モデルに関する状態空間モデルの構築が必要である [44]。蓄電池残量推定では、デジタル計算機に便利に使われる離散状態空間モデルを選択する。以下は線形状態空間モデルの定義式である。式(5.6)は状態方程式、式(5.7)は観測方程式である。その中、 $x(k)$ は状態ベクトル、 $y(k)$ は観測ベクトル、 $w(k)$  (システムノイズ) と  $v(k)$  (観測ノイズ) はガウスノイズベクトルである。電池の観測値は端子電圧しかないので、こちらの  $y(k)$  はスカラーで定義する。 $b$ は駆動行列である。 $F$ は遷移行列、 $H$ は観測行列、時刻  $k$  に依存する行列である。 $u(k)$ は制御入力、 $C$ と  $D$ は状態方程式と観測方程式における制御入力  $u(k)$  のゲインである。

$$x(k+1) = Fx(k) + Cu(k) + bw(k) \quad (5.6)$$

$$y(k) = Hx(k) + Du(k) + v(k) \quad (5.7)$$

非線形の場合は、状態空間モデルは以下の式に変わる。遷移行列  $F$ 、観測行列  $H$  を  $f(x(k))$ 、 $h(x(k))$  に変わるのは線形状態空間モデルと非線形状態空間モデルの変化点である。蓄電池の回路モデルでは、OCV-SOC カーブは非線形関数なので、非線形状態空間モデルに対応できる。

$$x(k+1) = f(x(k)) + Cu(k) + bw(k) \quad (5.8)$$

$$y(k) = h(x(k)) + Du(k) + v(k) \quad (5.9)$$

蓄電池の場合では、状態ベクトル  $x(k) = [\text{SOC}(k) \ u_1(k) \ u_2(k)]^T$ 、観測ベクトル  $y(k) = u_L(k)$  である。 $x(k+1)$  と  $x(k)$  の関係を表現する状態方程式は以下になる。 $\Delta t$  はサンプリング周期である。

$$x(k+1) = Fx(k) + Ci(k) + bw(k) \quad (5.10)$$

$$F = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \left(1 - \frac{\Delta t}{R_1 C_1}\right) & 0 \\ 0 & 0 & \left(1 - \frac{\Delta t}{R_2 C_2}\right) \end{bmatrix}, \quad C = \begin{bmatrix} \frac{\Delta t}{Q} \\ \frac{\Delta t}{C_1} \\ \frac{\Delta t}{C_2} \end{bmatrix}$$

その理由は時刻 $k$ のSOC( $k$ )と時刻 $k+1$ のSOC( $k+1$ ), 時刻 $k$ の $u_1(k)$ と時刻 $k+1$ の $u_1(k+1)$ , 時刻 $k$ の $u_2(k)$ と時刻 $k+1$ の $u_2(k+1)$ の関係式は式(5.11-5.13)になる。 $Q$ は電池の総容量である。式(5.11), (5.12)になる理由は式(5.3), (5.4)を前進オイラー法で離散化した結果である。具体の展開は式(5.14)のように $du_1$ を $u_1(k+1) - u_1(k)$ に,  $dt$ をサンプリング周期 $\Delta t$ に,  $u_1$ と $i$ を $u_1(k)$ と $i(k)$ に変換して,  $u_1(k+1)$ を左辺に移動し, それ以外の項を右辺に移動し, 整理すると, 式(5.14)になる。式(5.15)も同じ方法で, 導出することができる。

$$\text{SOC}(k+1) = \text{SOC}(k) + \frac{\Delta t}{Q} i(k) \quad (5.11)$$

$$u_1(k+1) = \left(1 - \frac{\Delta t}{R_1 C_1}\right) u_1(k) + \frac{i}{C_1} \Delta t \quad (5.12)$$

$$u_2(k+1) = \left(1 - \frac{\Delta t}{R_2 C_2}\right) u_2(k) + \frac{i}{C_2} \Delta t \quad (5.13)$$

$$C_1 \frac{u_1(k+1) - u_1(k)}{\Delta t} + \frac{u_1(k)}{R_1} = i(k) \quad (5.14)$$

$$C_2 \frac{u_2(k+1) - u_2(k)}{\Delta t} + \frac{u_2(k)}{R_2} = i(k) \quad (5.15)$$

$u_1(k)$ ,  $u_2(k)$ ,  $i(k)R_0$ と $u_{OCV}(k)$ を式(5.3)に代入すると, 観測方程式は式(5.16)のようになる。

$$y(k) = u_{OCV}(\text{SOC}) + u_1(k) + u_2(k) + R_0(\text{SOC})i(k) + v(k) \quad (5.16)$$

以上の説明で, 蓄電池の離散状態モデルが構築することができる。

カルマンフィルタは自動適応フィルタの一つであり, 観測データからシステム状態ベクトルを逐次推定するアルゴリズムである [44] [45]。具体のアルゴリズムは以下になる。その中では,  $\hat{x}^-(k)$ は時刻 $k$ のときの状態 $x(k)$ の事前推定値,  $\hat{x}(k)$ は時刻 $k$ のときの状態 $x(k)$ の事後推定値,  $P^-(k)$ は時刻 $k$ のときの事前推定誤差共分散行列,  $P(k)$ は時刻 $k$ のときの事後推定誤差共分散行列である。 $\sigma_w^2(k)$ はシステムノイズの分散,  $\sigma_v^2(k)$ は観測ノイズの分散である。事前推定値 $\hat{x}^-(k)$ 実際は観測値の時系列 $\{y(k-1), y(k-2), \dots, y(1)\}$ 集合の条件付き期待値であり, 事前推定値 $\hat{x}(k)$ は観測値の時系列 $\{y(k), y(k-1), \dots, y(1)\}$ 集合の条件付き期待値である。 $P^-(k)$ は時刻 $k$ のときの真値 $x(k)$ と事前推定値 $\hat{x}^-(k)$ の誤差の分散 $\mathbf{E}[x(k) - \hat{x}^-(k)][x(k) - \hat{x}^-(k)]^T$ であり,  $P(k)$ は時刻 $k$ のときの真値 $x(k)$ と事前推定値 $\hat{x}^-(k)$ の誤差の分散 $\mathbf{E}[x(k) - \hat{x}(k)][x(k) - \hat{x}(k)]^T$ である。カルマンフィルタは誤差共分散を最小化するフィルタである。



初期値推定

$$\hat{x}(0) = \mathbf{E}[x(0)], \quad P(0) = \mathbf{E}[x(0) - \hat{x}(0)][x(0) - \hat{x}(0)]^T$$

事前推定

$$\begin{aligned} \hat{x}^-(k+1) &= F\hat{x}(k) + Cu(k) \\ P^-(k+1) &= FP(k)F^T + bb^T\sigma_w^2(k) \end{aligned}$$

事後推定

$$\begin{aligned} G(k) &= \frac{P^-(k)H^T}{HP^-(k)H^T + \sigma_v^2(k)} \\ \hat{x}(k) &= \hat{x}^-(k) + G(k)\{y(k) - H\hat{x}^-(k) - Du(k)\} \\ P(k) &= \{I - G(k)H\}P^-(k) \end{aligned}$$

カルマンフィルタの動作の仕組み（ブロック図）は図 5-3 になる。カルマンフィルタは  $k$  時刻の事前推定値  $\hat{x}^-(k)$  と観測値  $y(k)$  の残差（イノベーションとよぶ、真値  $x(k)$  と推定値  $\hat{x}(k)$  の誤差）を使って事後推定値  $\hat{x}(k)$  を導く。そして、 $k$  時刻の事後推定値  $\hat{x}(k)$  を使って  $k+1$  時刻の事前推定値  $\hat{x}^-(k+1)$  を推定する。図 5-4 のようにその繰り返しはカルマンフィルタの特徴である。

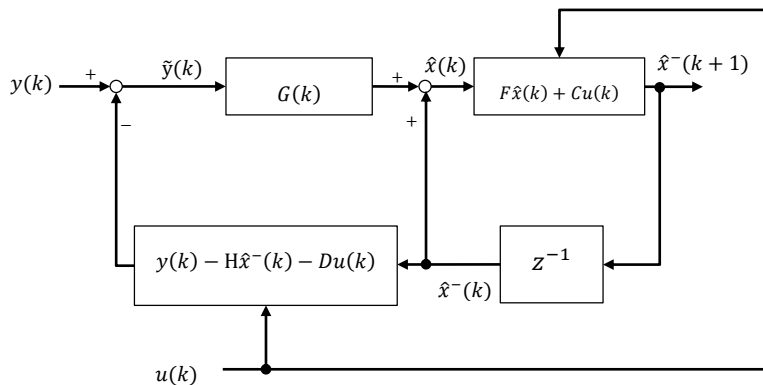


図 5-3 カルマンフィルタのブロック図

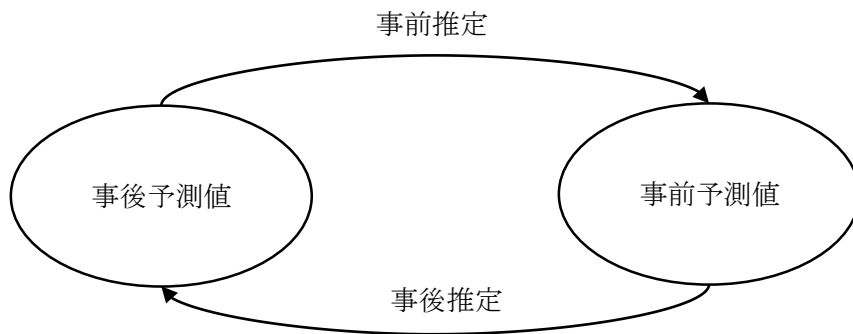


図 5-4 カルマンフィルタの事前推定と事後推定の関係

しかし、電池のモデルは線形ではなく、カルマンフィルタは線形モデルだけに対応できるので、非線形モデルに適応される拡張カルマンフィルタを用いる残量推定アルゴリズムを構築する発揚である。拡張カルマンフィルタは一次テイラー展開近似による等価線形化カルマンフィルタの一つである。拡張カルマンフィルタの考え方は非線形関数 $f(x(k))$ と $h(x(k))$ を各時刻の状態 $x(k)$ に線形近似する [44, 46]。図 5-5 は拡張カルマンフィルタの線形近似のイメージである。図 5-5 は 4 つのグラフを分けっている、その上半分は真実の確率変数の遷移図である、下半分は等価線形化確率変数の遷移図である。左側は状態方程式の状態変数 $x(k)$ の確率遷移図である、右側は状態観測方程式の状態変数 $x(k)$ から観測値 $y(k)$ に遷移する確率遷移図である。前に紹介された状態値 $x(k)$ を正規分布と考えると、 $x(k)$ をある非線形関数 $f(x(k))$ と $h(x(k))$ に入力すると、その出力の分布の形は変わる可能性がある。出力の分布は正規分布で表せないので、フィルタの推定値（条件付き期待値）と誤差分散の導出するとき、分布の形も考えなければならない。そのために、フィルタの計算量は大幅に増大する。精度を落ちることを容赦できると、非線形関数 $f(x(k))$ と $h(x(k))$ の導関数で線形近似することはよい。その利点は入力 $x(k)$ の確率分布と出力 $x(k+1)$ と $y(k)$ の確率分布の形は同じである。つまり、正規分布を入力すると、出力 $x(k+1)$ と $y(k)$ の確率分布の形も正規分布である。こちらは拡張カルマンフィルタ線形近似の特徴である。

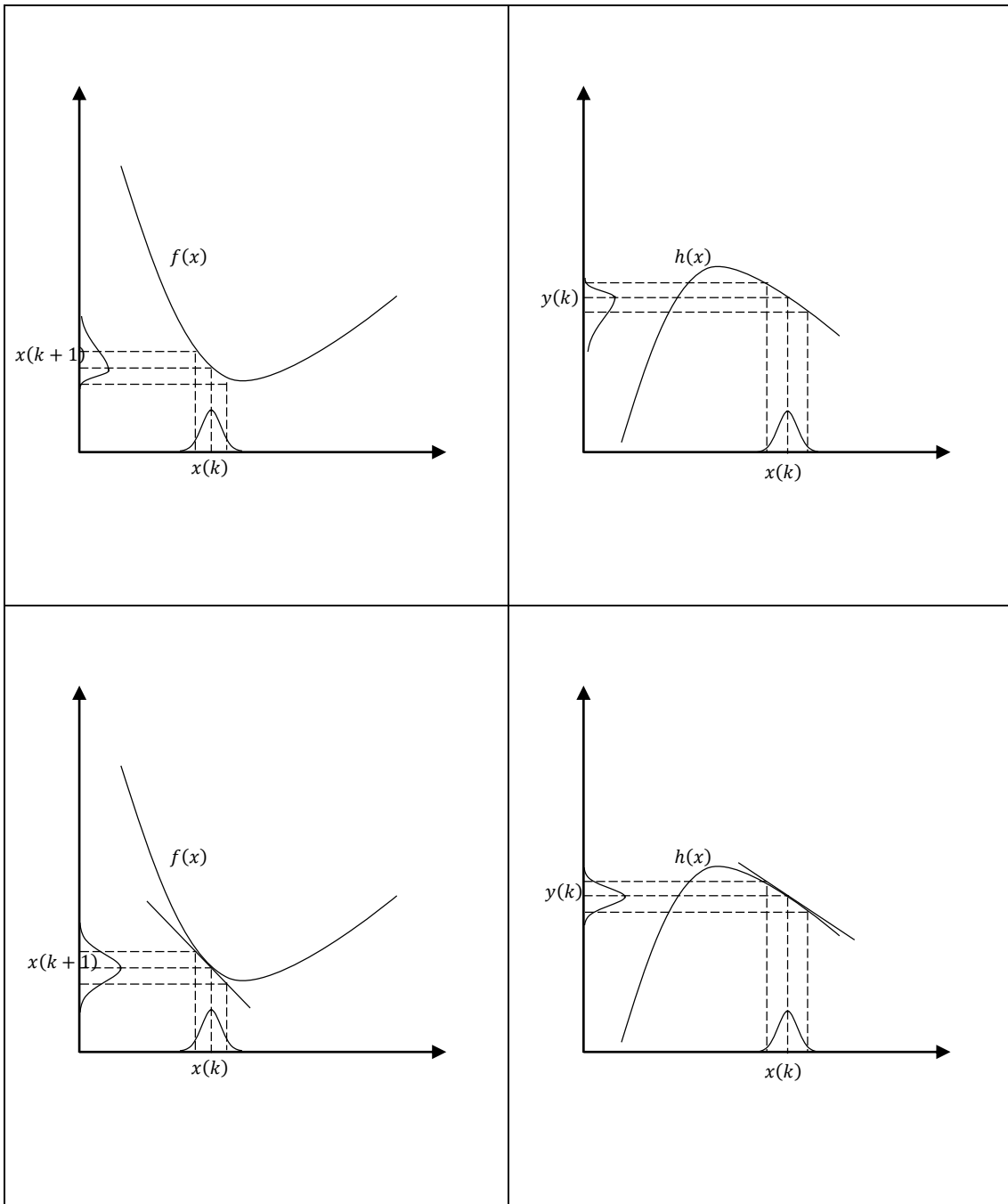


図 5-5 拡張カルマンフィルタの線形近似のイメージ

具体的なアルゴリズムは以下になる．その中， $\hat{F}(k)$ は $f(x(k))$ のヤコビアン（導関数）

$$\left. \frac{\partial f(x(k))}{\partial x(k)} \right|_{x(k)=\hat{x}^-(k)}$$

であり， $\hat{H}(k)$ は $h(x(k))$ のヤコビアン $\left. \frac{\partial h(x(k))}{\partial x(k)} \right|_{x(k)=\hat{x}(k)}$ である．

初期値推定

$$\hat{x}(0) = \mathbf{E}[x(0)], \quad P(0) = \mathbf{E}[x(0) - \hat{x}(0)][x(0) - \hat{x}(0)]^\top$$

事前推定

$$\begin{aligned} \hat{x}^-(k+1) &= f(\hat{x}(k)) + Cu(k) \\ P^-(k+1) &= \hat{F}(k)P(k)\hat{F}(k)^\top + bb^\top\sigma_w^2(k) \end{aligned}$$

事後推定

$$\begin{aligned} G(k) &= \frac{P^-(k)\hat{H}(k)^\top}{\hat{H}(k)P^-(k)\hat{H}(k)^\top + \sigma_v^2(k)} \\ \hat{x}(k) &= \hat{x}^-(k) + G(k)\{y(k) - h(\hat{x}^-(k)) - Du(k)\} \\ P(k) &= \{I - G(k)\hat{H}(k)\}P^-(k) \end{aligned}$$

蓄電池の状態方程式(5.10)は線形であり、 $u_{\text{OCV}}$ (SOC)関数は非線形であるので、観測方程式(5.16)は非線形である。つまり、観測方程式(5.11)だけ線形近似が必要である。観測方程式(5.17)のヤコビアン式は以下になる。蓄電池に応用される拡張カルマンフィルタは以下になる。状態方程式は線形であるので、事前推定は線形カルマンフィルタと同じである。

$$\hat{H}(k) = \left. \frac{\partial h(x(k))}{\partial x(k)} \right|_{x(k)=\hat{x}^-(k)} = \left[ \left. \frac{du_{\text{OCV}}}{d\text{SOC}} \right|_{\text{SOC}=\widehat{\text{SOC}}^-(k)}, 1, 1 \right] \quad (5.17)$$

初期値推定

$$\hat{x}(0) = \mathbf{E}[x(0)], \quad P(0) = \mathbf{E}[x(0) - \hat{x}(0)][x(0) - \hat{x}(0)]^\top$$

事前推定

$$\begin{aligned} \hat{x}^-(k+1) &= F\hat{x}(k) + Cu(k) \\ P^-(k+1) &= FP(k)F^\top + bb^\top\sigma_w^2(k) \end{aligned}$$

事後推定

$$\begin{aligned} G(k) &= \frac{P^-(k)\hat{H}(k)^\top}{\hat{H}(k)P^-(k)\hat{H}(k)^\top + \sigma_v^2(k)} \\ \hat{x}(k) &= \hat{x}^-(k) + G(k)\{y(k) - h(\hat{x}^-(k)) - Du(k)\} \\ P(k) &= \{I - G(k)\hat{H}(k)\}P^-(k) \end{aligned}$$

## 5.2 自動適応ノイズ推定アルゴリズム

本節は自動適応ノイズ推定アルゴリズムを述べる. カルマンフィルタの応用ではノイズの設定は難しい課題であるので, ノイズの設定による推定精度は変わる. 最適なノイズ推定が重要な課題となっている. 本論文では, イノベーション (残差) による最適なノイズ推定を行って, よりよい SOC 推定アルゴリズムを構築する [47]. 具体的に, 式(5.18)のように観測データ  $y(k)$  と  $h(\hat{x}^-(k)) - Du(k)$  の誤差 (イノベーション) から観測のいずを予測する.

$$\tilde{y}(k) = y(k) - h(\hat{x}^-(k)) + Du(k) \quad (5.18)$$

非線形関数  $h(x(k)) + Du(k)$  を  $\hat{x}^-(k)$  の周りにテイラー展開すると, 次式になる. 式中の... は高次項 (2 次以上) を表せる.

$$h(x(k)) + Du(k) = h(\hat{x}^-(k)) + Du(k) + \hat{H}(k)(x(k) - \hat{x}^-(k)) + \dots$$

その中の  $\hat{H}(k)$  は式(5.19)のようにヤコビ行列である.

$$\hat{H}(k) = \left. \frac{\partial h(x(k))}{\partial x(k)} \right|_{x=\hat{x}^-(k)} \quad (5.19)$$

高次項を無視できれば,  $y(k)$  を次式に書き直せる.

$$y(k) = h(\hat{x}^-(k), u(k)) + \hat{H}(k)(x - \hat{x}^-(k)) + v(k) \quad (5.20)$$

式(5.14)の通りに, イノベーション  $\tilde{y}(k)$  の式は以下になる.

$$\tilde{y}(k) = \hat{H}(k)(x(k) - \hat{x}^-(k)) + v(k) \quad (5.21)$$

イノベーション  $\tilde{y}(k)$  の分散  $\mathbf{E}\{\tilde{y}(k)\tilde{y}^T(k)\}$  と観測ノイズの分散  $\sigma_v^2(k)$  の関係式は式(5.22)のように書く.

$$\begin{aligned} \mathbf{E}\{\tilde{y}(k)\tilde{y}^T(k)\} &= \hat{H}(k)\mathbf{E}\{(x - \hat{x}^-(k))(x - \hat{x}^-(k))^T\}\hat{H}^T(k) + \sigma_v^2(k) \\ &= \hat{H}(k)P^-(k)\hat{H}^T(k) + \sigma_v^2(k) \end{aligned} \quad (5.22)$$

$\mathbf{E}\{\tilde{y}(k)\tilde{y}^T(k)\}$  を  $Y(k)$  に定義し,  $Y(k)$  と観測ノイズの分散  $\sigma_v^2(k)$  の関係式は(5.23)になる. 信号の時間平均とサンプル平均が同じ (エルゴード性) になると,  $Y(k)$  を式(5.24)に定義できる.

$$Y(k) = \hat{H}(k)P^-(k)\hat{H}^T(k) + \sigma_v^2(k) \quad (5.23)$$

$$Y(k) \stackrel{\text{def}}{=} \frac{1}{k+1} \sum_{i=0}^k \tilde{y}(i)\tilde{y}^T(i) \quad (5.24)$$

(5.23)と(5.24)によって,  $\sigma_v^2(k)$  を逆算することができる. その逆算した結果は  $\sigma_v^2(k)$  の推定値  $\hat{\sigma}_v^2(k)$  を式(5.25)のように定義する.

$$\hat{\sigma}_v^2(k) = Y(k) - \hat{H}(k)P^-(k)\hat{H}^T(k) \quad (5.25)$$

イノベーション  $\tilde{y}$  の分散の逐次推定式, 式(5.26)のように書く.

$$Y(k) = \frac{1}{k+1} \tilde{y}(k) \tilde{y}^T(k) + \frac{k}{k+1} Y(k-1) \quad (5.26)$$

実際に、観測ノイズは定常ではない場合が多い。つまり、観測ノイズは時間によって変化する  
 ことである。その問題を解決するために、スライディングウィンドウを使って、時間変化  
 する観測ノイズを推定することは提案手法の特徴である。式(5.27)はスライディングウィ  
 ンドウを用いた $Y(k)$ の定義式である。 $M$ はスライディングウィンドウの長さである。実際実現  
 すると、長さ $M$ の FIFO メモリが必要である。逐次的な算出を考えると、 $Y(k)$ の導出式は式  
 (5.28)になる。しかし、インデックス $k$ は FIFO メモリの長さ $M$ より小さくなると、つまり、FIFO  
 メモリはフルの状態ではないとき、式(5.28)を対応できない。そのため、式(5.24)で $Y(k)$ を  
 算出する。また、長さ $M$ の設定値は相当に小さくなると、 $Y(k)$ の算出値は $C(k)P^-(k)C^T(k)$ よ  
 り小さくなる可能性がある。観測ノイズ分散予測値 $\hat{\sigma}_v^2(k)$ はマイナスになる可能性がある。  
 分散予測値は正に決まらるので、マイナスになるとエラーの推定状態になる。その状態を  
 避けるために、長さ $M$ の下限を決まらなければならない。蓄電池の残量推定アルゴリズムの  
 場合は、30 は長さ $M$ の設定値の下限である。自動適応ノイズ推定アルゴリズムの全体ブロ  
 ック図は図 5-6 で示す [48].

$$Y(k) \stackrel{\text{def}}{=} \frac{1}{M} \sum_{i=0}^{M-1} \tilde{y}(k-i) \tilde{y}^T(k-i) \quad (5.27)$$

$$Y(k) = Y(k-1) + \frac{1}{M} \tilde{y}(k) \tilde{y}^T(k) - \frac{1}{M} \tilde{y}(k-M) \tilde{y}^T(k-M) \quad (5.28)$$

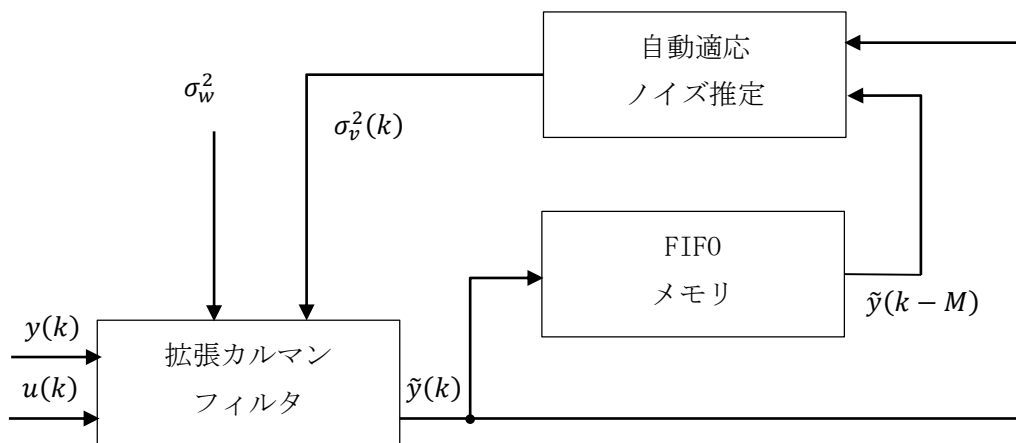


図 5-6 自動適応ノイズ推定のブロック図

### 5.3 残量推定実験と評価

提案した SOC 推定方法の適用性を検証するために、我々は図 5-7 から図 5-9 のように 3 つのパターンの放電実験を用意する。パターン 1 は、図 5-7 に示すパルスによる放電である。パターン 2 は図 5-8 に示すように、周期性あり放電である。パターン 3 は図 5-9 に示す擬似ランダム放電である。各放電パターン図では、上半分は電圧波形（蓄電池の端子電圧）、下半分電流波形（蓄電池の端子電流）である。実験で評価基準として使われる基準 SOC はベテリテスタの SOC 測定値である。バッテリーテスタは高性能クーロンカウンタを使用して SOC を算出し、バッテリーテスタが充放電サイクルを完了すると自動的に容量をリセットする機能を持つため、十分いい精度 SOC と信用できる。

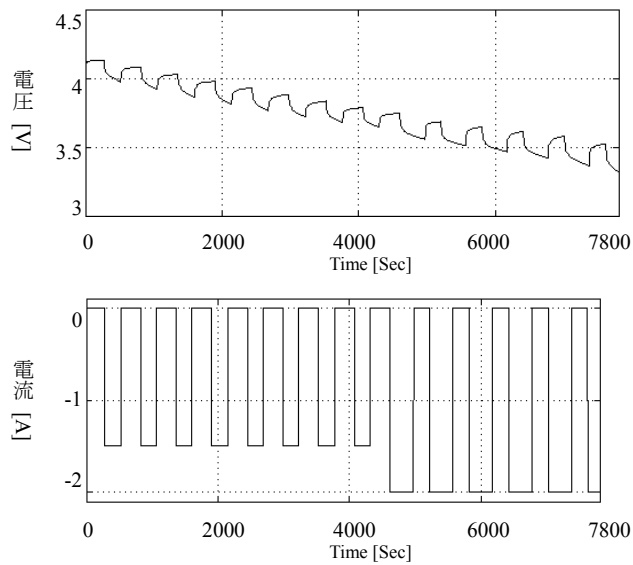


図 5-7 パターン 1 の電流電圧波形

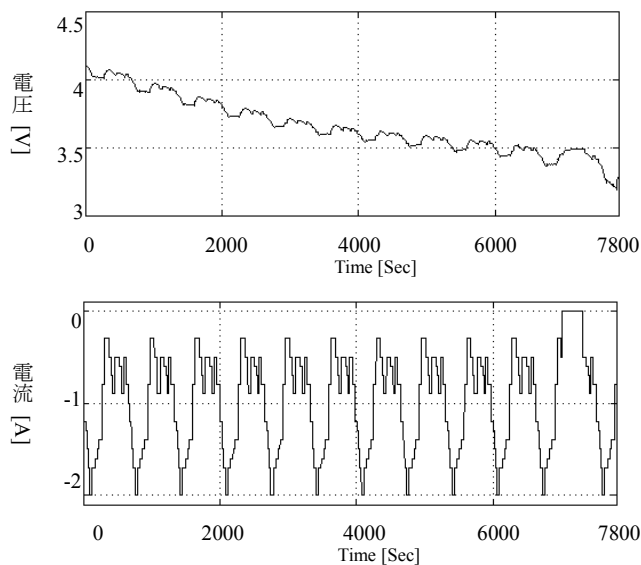


図 5-8 パターン 2 の電流電圧波形

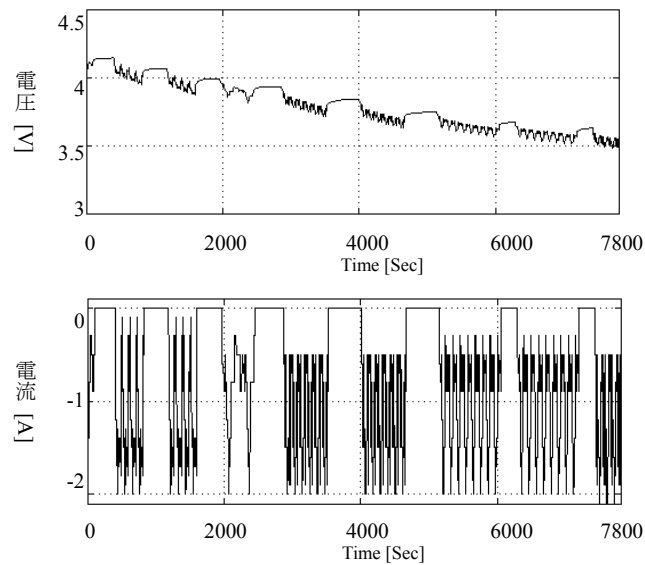


図 5-9 パターン 3 の電流電圧波形

自動適応ノイズ共分散推定の効果を評価するに、テストの実験から得られた結果を用いて最適なノイズ共分散設定値を探索して、その探索した結果と比べる。最適なノイズ共分散設定値を探索するために、式(5.29)のようにRMS値を評価基準として探索する。探索領域は  $10^{-7} \leq \sigma_w \leq 10^{-1}$ ,  $10^{-3} \leq \sigma_v \leq 10$  の対数グリッドである。式(5.29)では、 $SOC_{est}$  が推定SOCであり、 $SOC_{true}$  はバッテリーテスタによるSOCの真値であり、Nは推定の数である。その結果をパターン1例としてパターン1の最適ノイズ分散設定値の探索図は図5-10に示している。各パターンの最適なノイズ分散設定値は表5-1の通りに示されている。

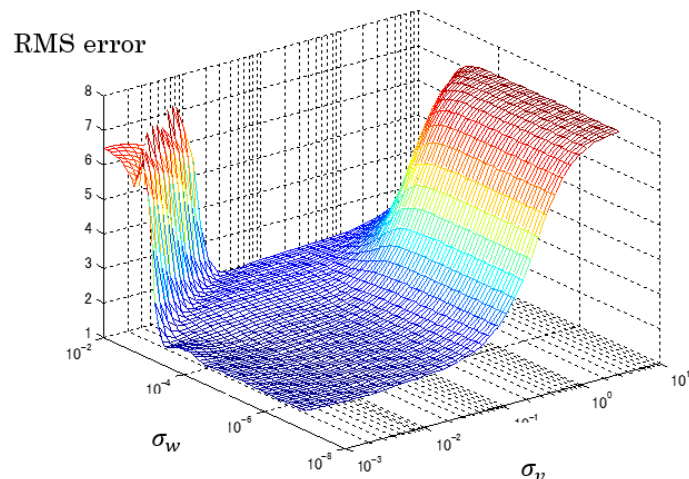


図 5-10 パターン 1 の最適ノイズ分散設定値の探索図



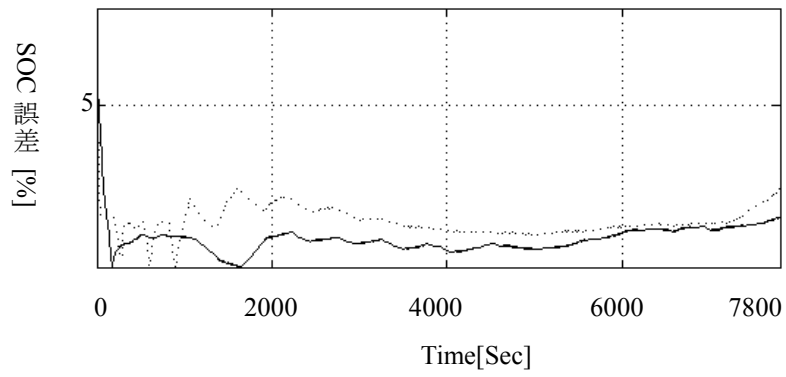
$$E_{\text{RMS}} = \sqrt{\frac{1}{N} \sum_{i=1}^N [\text{SOC}_{\text{est}}(i) - \text{SOC}_{\text{true}}(i)]^2} \quad (5.29)$$

表 5-1 最適ノイズ分散設定値の探索結果

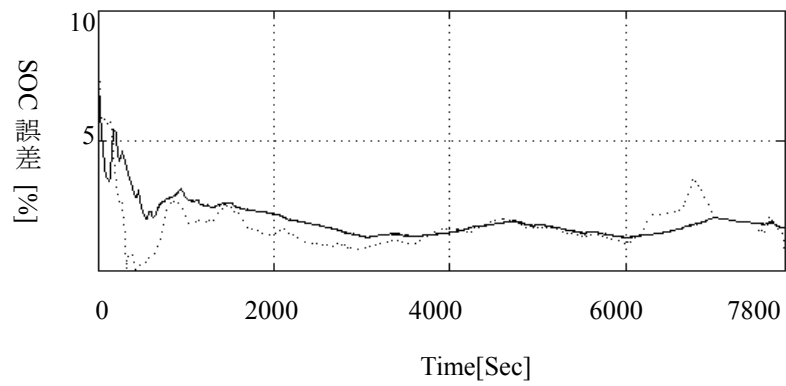
	パターン 1	パターン 2	パターン 3
システムノイズ分散 $\sigma_w$	$4.08 \times 10^{-6}$	$1.94 \times 10^{-5}$	$3.35 \times 10^{-6}$
観測ノイズ分散 $\sigma_v$	0.00522	0.002031	0.00662

自動適応ノイズ分散推定 (Adaptive noise covariance estimation) の比較対象は表 5-1 のように静的最適ノイズ分散設定値 (Stationary noise covariance estimation) の探索結果である。自動適応ノイズ分散推定はシステムノイズを含めていないので、我々は $\sigma_w = 10^{-5}$  で設定する。Y(k)の初期値Y(0)は0に設定する。テスト実験では、FIFO メモリの長さMは50, 200, 500, 1000を用意し、長さMの設定による推定精度を確認する。前に説明した通りに、30は長さMの設定値の下限であるので、長さMの設定値は30以上になる。その推定誤差は図 5-11 と

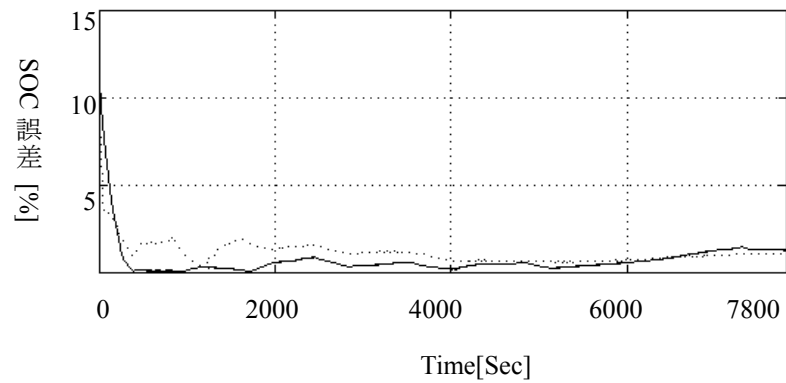
図 5-12 になる。図 5-11 はMを50に設定するとき、推定誤差である。図 5-11 はMを50に設定するとき、観測ノイズ標準偏差 $\sigma_v$ の推定値である。図 5-13 はMを50設定するとき、推定誤差である。図 5-14 はMを1000に設定するとき、観測ノイズ標準偏差 $\sigma_v$ の推定値である。各図の中では、点線は最適ノイズ分散設定値による推定誤差と観測ノイズ標準偏差 $\sigma_v$ の設定値。図 5-15 は各パターンの RMS 誤差と平均誤差のまとめである。



(a)



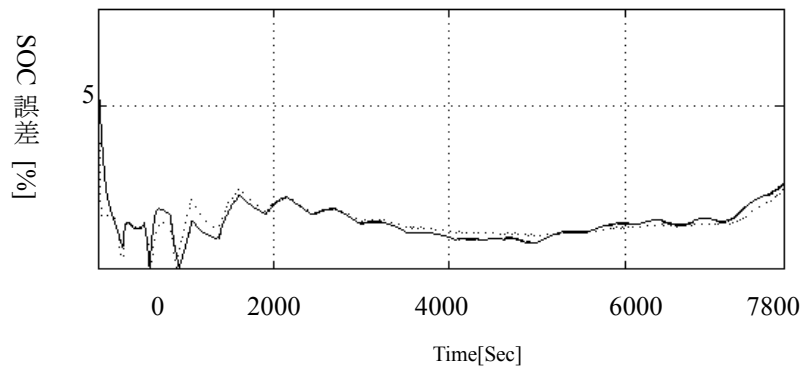
(b)



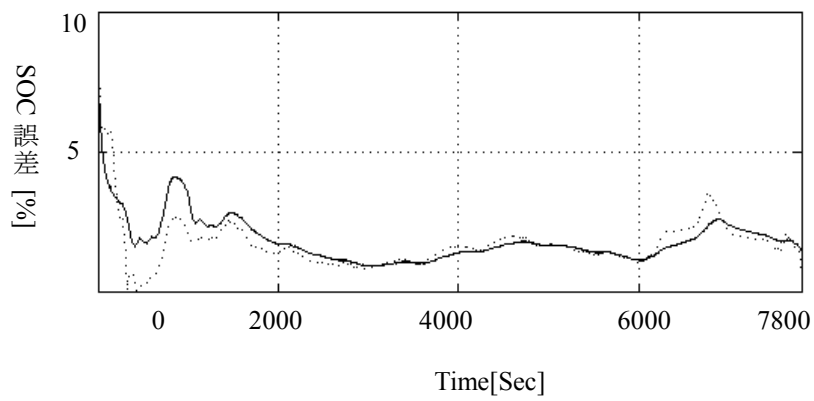
(c)

——— 自動適応  
 ノイズ分散推定
 
 ..... 静的最適ノイズ  
 分散設定

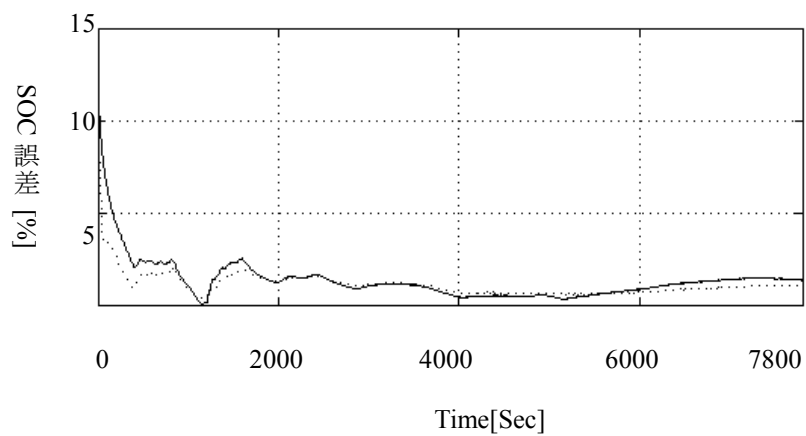
図 5-11 各パターンの推定誤差 ( $M = 50$ )  
 (a) パターン 1, (b) パターン 2, (c) パターン 3



(a)



(b)



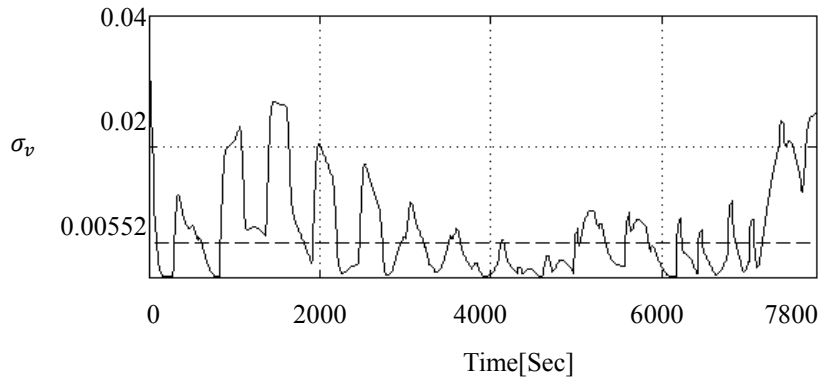
(c)

——— 自動適応  
 ノイズ分散推定

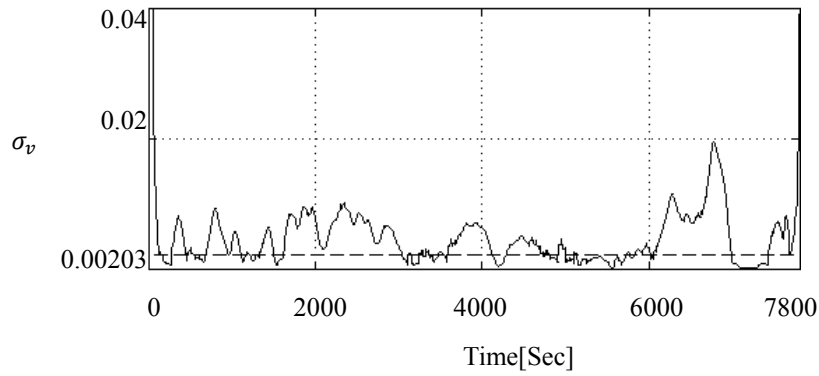
..... 静的最適ノイズ  
 分散設定

図 5-12 各パターンの推定誤差 ( $M = 1000$ )

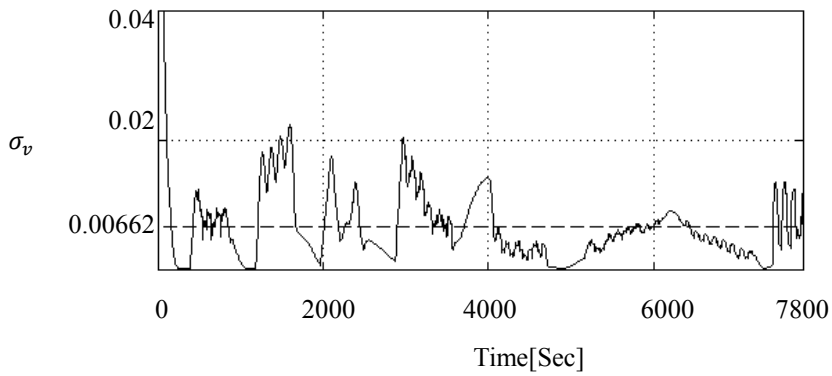
(a) パターン 1, (b) パターン 2, (c) パターン 3



(a)



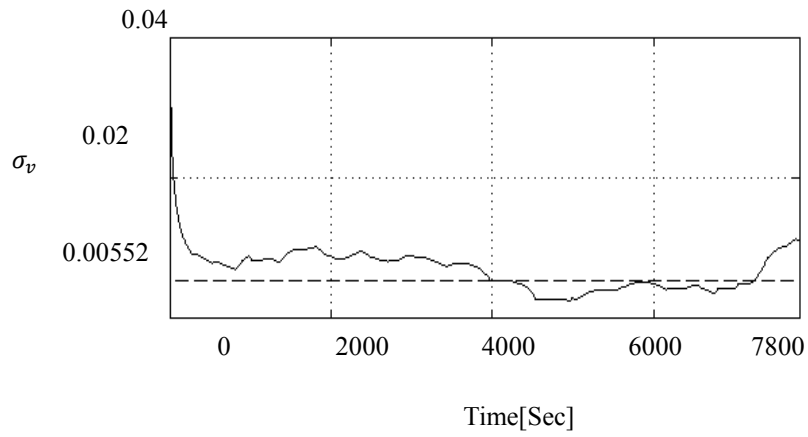
(b)



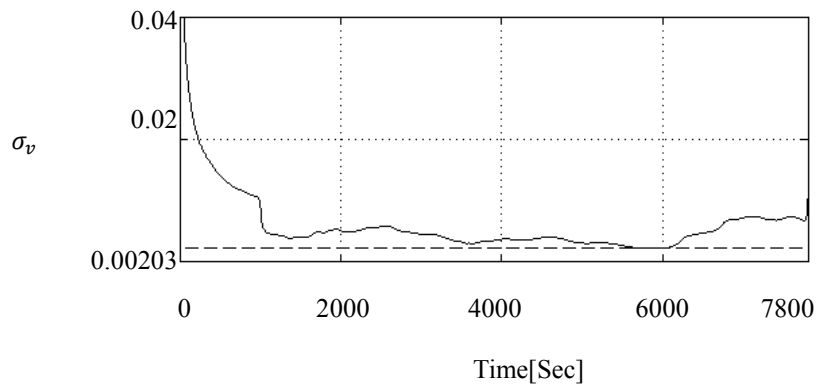
(c)

自動適応  
 ノイズ分散推定
     
  静的最適ノイズ  
 分散設定

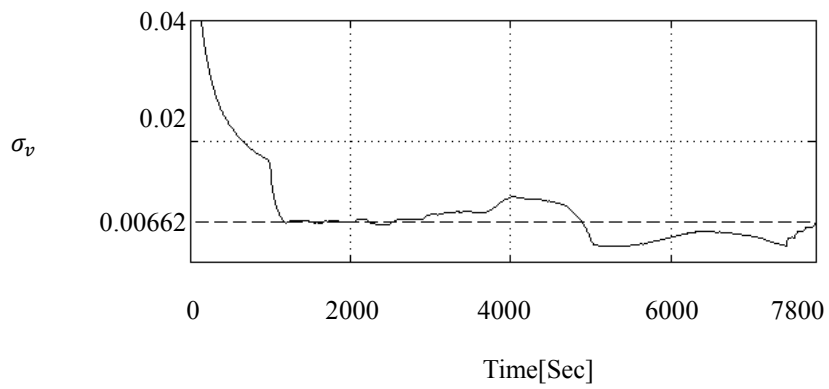
図 5-13 各パターンの観測ノイズ標準偏差 $\sigma_v$ の推定値 ( $M = 50$ )  
 (a) パターン 1, (b) パターン 2, (c) パターン 3



(a)



(b)



(c)

自動適応  
 静的最適ノイズ  
 ノイズ分散推定                      分散設定

図 5-14 各パターンの観測ノイズ標準偏差 $\sigma_v$ の推定値 ( $M = 1000$ )  
 (a) パターン 1, (b) パターン 2, (c) パターン 3

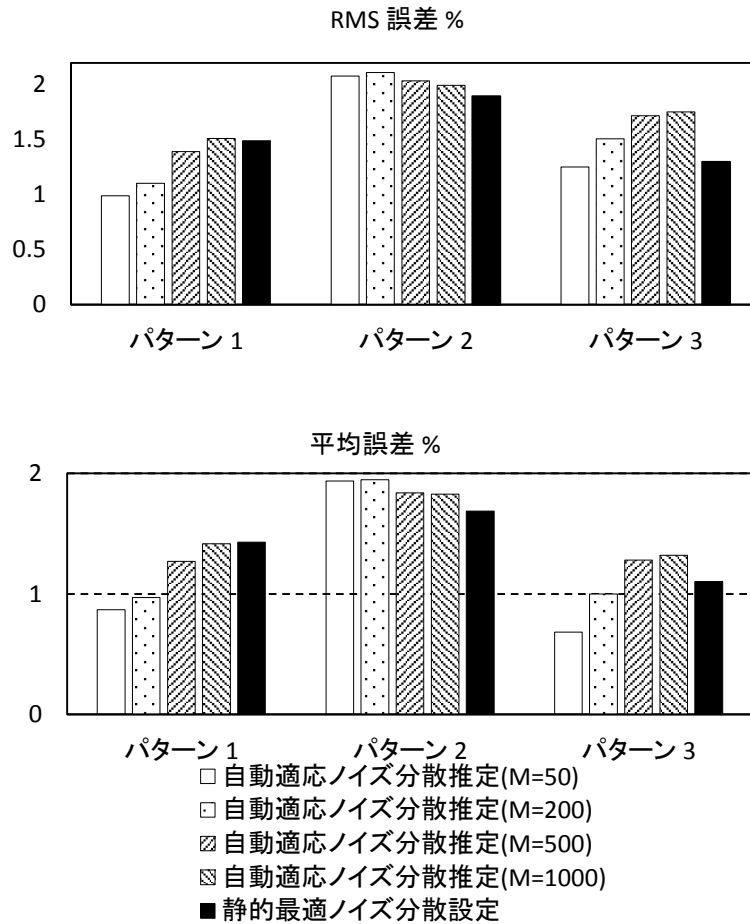


図 5-15 各パターンの RMS 誤差と平均誤差

## 5.4 結言

本章では、蓄電池の残量推定の問題点を説明し、その問題に対してカルマンフィルタは有効な解決手段である。そして、カルマンフィルタの仕組みと蓄電池の拡張カルマンフィルタを基づいた残量推定アルゴリズムを説明し、観測値と推定値の残差（イノベーション）による観測ノイズの最適推定を行う。観測ノイズの放電実験データによって、最適推定によるSOC推定精度がよくなることが確認した。静的な観測ノイズ設定方法と比べると、3つの放電パターンのうちに、2放電パターンの推定精度は明確に改善することができた。

## 第6章 結論

蓄電池管理において、残量予測、劣化診断、電池の熱管理は重要な3つの課題として、本論文はその3つの課題の解決方法を述べる。第2章は準備知識として蓄電池のモデルを説明する。

第3章では、熱回路を用いた蓄電池の熱挙動を解析する。熱回路網法は物体の伝熱を表現する方法である。熱回路モデルの規模は大きいので、並列計算を行わないと、膨大な時間がある。その問題を解決するために、GPGPUを用いた電気回路の並列計算を転用して、熱回路モデルの解析の高速化手法を提案する。従来法では、前進オイラー法(forward Euler's rule)で解析するので、高速な並列処理できるが、精度はよろしくない欠点がある。シミュレーション精度を上昇するために、後退オイラー(backward Euler's rule)を用いた手法は有効な手段と認められている。それに対してGPUによる回路モデルの解析の高速化手法を述べる。電源配線の回路の特徴を活かした疎行列の新しい表現方法CSGと、それを用いた、ヤコビ法による後退オイラー法をGPUに実装し評価した。その結果、GPUでの解析時間はCPU比で約74.0倍の高速化を実現することが出来た。また、使用メモリ領域も、CSGを用いることで大幅に削減することが出来た。

第4章では、蓄電池は内部抵抗の劣化に伴い変化するため、起電力や内部抵抗などの内部パラメータ変化する。その原因で、蓄電池の内部パラメータをリアルタイム的に把握するのは重要な課題になる。本論文では、リチウムイオン蓄電池を対象として、忘却係数自動設定逐次最小二乗法を用いた動的な蓄電池内部パラメータ推定手法を提案した。従来の手法では固定的な忘却係数 $\lambda$ を用いて、計測時間中にOCVが変動することによって生じる誤差の問題を解消している。しかし、一般に、蓄電池のOCVの変動量は、電流やSOCの影響を受けるため、一定ではない。このような同定問題に対して忘却係数 $\lambda$ を固定とするのは適切ではない。提案手法は電池端子電圧の変動による忘却係数を自動算出し、 $u_{ocv}$ を推定する手法を検討する。その結果、 $u_{ocv}$ と内部抵抗を同定することが可能であり、実験結果から $u_{ocv}$ によるSOC推定精度5%以内に抑えた提案する方法の有効性を示すために、 $u_{ocv}$ 既知の推定結果とオフライン手法コールルーコールプロットとの比較を行った。その結果、内部抵抗 $R_a$ は高い精度を求める可能となる。

第5章では、蓄電池残量予測の高精度化手法について述べた。古典的な蓄電池残量予測方法は電流積算法、端子電圧法と内部抵抗法が提案されている。古典的な蓄電池残量予測方法は精度や初期状態の問題で、高性能残量推定することが難しい。本論文では、拡張カルマンフィルタを用いた高精度残量予測手法を提案し、実装と評価を行う、そして、カルマンフィルタに置いてノイズ分散値の設定方法を議論する。提案手法はイノベーションによる最適なノイズ分散を自動的に推定する。静的なノイズ設定と比べると、SOCの推定誤差は改善できたことが確認した。

今後、蓄電池のモデリング技術と制御技術の発展に伴って、以下に展望する。

蓄電池のモデルとその解析では、実電池の熱挙動と比較し、熱パラメータキャリブレーションを行う。実用できる蓄電池内部温度推定システムを構築する。

蓄電池の内部パラメータ推定と蓄電池残量予測では、蓄電池の容量、寿命も含める推定システムを構築する。現在の等価回路モデルより正確な電気化学モデルを用意し、蓄電池の物理的なパラメータが詳細に把握できるようになる。そのモデルの応用に応じて、今よりよく残量かつ劣化状態を把握できるようにする。そして、より高度な制御技術を応用し、蓄電池の内部状態と内部パラメータを同時に推定することを実現する。最新の情報技術を応用し、大量な蓄電池の使用状況、劣化状態を管理するビッグデータ技術を目指す。



# 謝辞

本論文を執筆するあたり多くの方々の世話になりましたので、ここに記して感謝の意を表します。

本研究の全過程を通じて、始終ご懇切な指導を頂いた立命館大学理工学研究科福井正博教授に深謝の意を上げます。

立命館大学理工学研究科鷹羽浄嗣教授は本研究を進めるに当たり、ご指導、貴重なアドバイスを賜りました。感謝もう上げます。

立命館大学 VLSI 最適化学研究室の方々に感謝いたします。VLSI 最適化学研究室の卒業生も本研究に有益なアドバイスを賜りました。困難に挑み成果を出し続ける VLSI 最適化学研究室の仲間により感謝いたします。

本研究の一部は、科学研究費補助金 基盤研究(C)(2) 23500071(2011～2013 年度)「100 億トランジスタ規模集積回路のチップレベルタイミング解析技術に関する研究」。

および、NEDO イノベーション実用化ベンチャー支援事業（立命館大学、シンセシス株式会社共同プロジェクト）「蓄電池の性能低下・劣化を逐次計測するスマートセンサー」により行われた。

ここに感謝の意を示します。



## 参考文献

- [1] 小久見善八, 西尾晃治, 図解革新型蓄電池のすべて, 工業調査会, 2010.
- [2] P. Weicker, *A Systems Approach to Lithium-Ion Battery Management*, Artech House, 2014.
- [3] V. Pop, J. Bergveld, and , D. Danilov, *Battery Management Systems: Accurate State-of-Charge Indication for Battery-Powered Applications*, Springer, 2008.
- [4] K. Cheng, B. Divakar, H. Wu, K. Ding , H. F. Ho, *Battery-management system (BMS) and SOC development for electrical vehicles*, Vol. 60, No. 1, pp. 76-88: *IEEE Trans. Vehicular Technology*, 2011.
- [5] 国峯尚樹, 中村篤, 熱設計と数値シミュレーション, オーム社, 2015.
- [6] 国峰尚樹, 電子機器の熱流体解析入門(第2版), 日刊工業新聞, 2015.
- [7] 磯田有哉, 横田誠, 福井正博, “GPUによる電源配線シミュレーションの高速化と評価,” 電子情報通信学会和文論文誌 D, vol.J95-D, no.3, pp.366-375, 2012.
- [8] 林磊, 塩野隼人, 福井正博, “GPGPUによる電源回路シミュレーションに対する後退オイラー法の実装と評価,” 電子情報通信学会和文論文誌 A, 第96巻, 第5号, pp.215-224, 2013.
- [9] 林磊, 石崎龍, 鷹羽浄嗣, 福井正博, 適応的忘却係数調整によるリチウムイオン蓄電池の逐次最小2乗同定, vol.J99-B, no.7, pp. 481-489: 電子情報通信学会論文誌 B, 2016.
- [10] 板橋欣之介, 岩鼻利幸, 寺西望, 福永おり, 川口貴弘, 足立修一, “等価回路を用いたリチウムイオン2次電池のパラメータ推定,” 第53回自動制御連合講演会, pp.577-580, 2010.
- [11] C. Fleischera, W. Waaga, H. Heyna, and , D. U. Sauera, “On-line adaptive battery impedance parameter and state estimation considering physical principles in reduced order equivalent circuit battery models: Part 1. Requirements, critical review of methods and modeling,” *J. Power Sources*, vol.260, no. 15, pp. 276-291, 2014.
- [12] C. Fleischera, W. Waaga, H. Heyna, and , D. U. Sauera, “On-line adaptive battery impedance parameter and state estimation considering physical principles in reduced order equivalent circuit battery models: Part 2. Parameter and state estimation,” *J. of Power Sources*, vol.262, no.15, pp.457-482, 2014.
- [13] G. Wu, R. Lu, C. Zhu, and , C. C. Chan, “State of charge estimation for NiMH

- battery based on electromotive force method,” in Proc. IEEE Vehicle Power and Propulsion, pp.1-5, 2008.
- [14] F. Baronti, G. Fantechi, L. Fanucci, E. Leonardi, R. Roncella, R. Saletti, and S. Saponara, “State-of-charge estimation enhancing of lithium batteries through a temperature-dependent cell model,” in Proc. Applied Electronics, pp.1-5, 2011.
- [15] 藤本政士, 井上達也, 林磊, 福井正博, PSoC を用いたリチウムイオン蓄電池の残量予測システムの実装と評価, 電子情報通信学会リコンフィギャラブルシステム研究会, 2012.
- [16] Z. Chen, Y. Fu, C. C. Mi, State of charge estimation of lithium-ion batteries in electric drive vehicles using extended kalman filtering, Vol. 62, No. 3, PP. 1020-1030: IEEE Tran. vehicular technology, 2013.
- [17] G. L. Plett, Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 3. State and parameter estimation, Vol. 134, No. 2, pp. 277-292: Journal of Power Sources, 2004.
- [18] G. L. Plett, Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 1. Background, Vol. 134, No. 2, pp. 252-261: Journal of Power Sources, 2004.
- [19] G. L. Plett, Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 2. Modeling and identification, Vol. 134, No. 2, pp. 262-276: Journal of Power Sources, 2004.
- [20] G. L. Plett, Sigma-point Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 1: Introduction and state estimation, Vol. 161, No. 2, pp. 1356-1368: Journal of Power Sources, 2006.
- [21] G. L. Plett, Sigma-point Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 2: Simultaneous state and parameter estimation, Vol. 161, No. 2, pp. 1369-1384: Journal of Power Sources, 2006.
- [22] M. Charkhgard, M. Farrokhi, State-of-charge estimation for lithium-ion batteries using neural networks and EKF, Vol. 57, No. 12, pp. 4178-4187: IEEE Trans. Industrial Electronics, 2010.
- [23] 堀江明英, リチウムイオン電池「基礎と応用」, 培風館, 2010.
- [24] 重松浩一, “バッテリーのシミュレーション技術モデリングと応用,” 電子情報通信学会総合大会, ABCT-1-3, 2015.
- [25] 板垣昌幸, 電気化学インピーダンス法, 丸善出版, 2011.
- [26] 大村崇, 林磊, 孟林, 福井正博, “GPGPU を用いた VLSI チップ熱解析の一手法,”

- 電子情報通信学会 VLSI 設計技術研究会(VLD), pp.209-214, 2013.
- [27] M. Fukui, I. Tnaka, R. Miyahara, and , S. Gora, Cost-effective power management for smart house considering degradation of batteries, pp.1-6: in Proc. IEEE PES POWERCON, 2012.
- [28] Y. Kitagawa, K. Kato, and , M. Fukui, “Analysis and experimentation for effective cooling of Li-ion battery modules,” J. Procedia Technology, vol.18, pp.63-67, 2014.
- [29] 加藤啓路, 山本祐介, 林磊, 福井正博, “蓄電池温度特性および温度管理システム構築の一考察,” 電子情報通信学会電子通信エネルギー技術研究会(EE), pp.59-64, 2013.
- [30] 加藤啓路, 山本祐介, 林磊, 福井正博, “リチウムイオン蓄電池の熱特性解析モデルの構築と内部温度推定手法,” 第 54 回電池討論会, pp.89, 2013.
- [31] 牛田明夫, 田中衛, 電子回路シミュレーション, コロナ社, 2002.
- [32] 浅井秀樹, 渡辺高之, 電子回路シミュレーション技法, 科学技術出版, 2003.
- [33] T. Omura, L. Lin, L. Meng, and , M. Fukui, “A two-dimensional thermal analysis system for LSI packages by GPGPU,” in Proc. International Technical Conference on Circuits/Systems, Computers and Communications, 2014.
- [34] J. Sanders, and , E. Kandrot , CUDA by Example 汎用 GPU プログラミング入門, インプレスジャパン, 2011.
- [35] N. Bell, and , M. Garland, “Efficient sparse matrix-vector multiplication on CUDA,” NVIDIA Technical Report, 2008.
- [36] Z. Feng, and , P. Li, “Multigrid on GPU: Tackling Power Grid Analysis on Parallel SIMT Platforms,” in proc. IEEE International Conference on Computer-Aided Design, pp.647-654, 2008.
- [37] A. Maringanti, V. Athavale, and , S. B. Patkar, “Acceleration of conjugate gradient method for circuit simulation using CUDA,” in Proc. HiPC, pp.438-444, 2009.
- [38] 福井正博, “リチウムイオン蓄電池の劣化抑制と制御技術,” 技術情報協会エネルギーデバイス, vol.2, no.2, pp.63-67, 2014.
- [39] 足立修一, システム同定の基礎, 東京電機大学出版局, 2009.
- [40] 足立修一, MATLAB による制御のためのシステム同定, 東京電機大学出版局, 1996.
- [41] 片山徹, システム同定入門, 朝倉書店, 1994.
- [42] L. Ljung, System Identification Theory for The User, Prentice-Hall, 1987.
- [43] S. Piller, M. Perrin, and , A. Jossen, “Methods for state-of-charge determination and their applications,” J. Power Sources, Vol. 96, pp.113-120, 2001.
- [44] 片山徹, 非線形カルマンフィルタ, 朝倉書店, 2011.

- [45] 片山徹, 新版カルマンフィルタ, 朝倉書店, 2000.
- [46] 足立修一, カルマンフィルタの基礎, 東京電機大学出版局, 2012.
- [47] R. K. Mehra, “On the identification of variances and adaptive Kalman filtering,”  
IEEE Trans. Automatic Control, VOL. 15, NO. 2, pp. 175-184, 1970.
- [48] L. Lin, K. Takaba, and , M. Fukui, “Practical and Accurate SOC Estimation  
System for Lithium-Ion Batteries by EKF with Adaptive Noise Covariance  
Estimation,” in proc. The 20th Workshop on Synthesis And System Integration of  
Mixed Information technologies, pp. 33-36, 2016.
- [49] Y. Yamamoto, K. Kato, L. Lin, and , M. Fukui, “A thermal management system  
for lithium-ion battery in mobile systems,” in Proc. ECCTD, pp.1-4, 2013.
- [50] R. Farber, CUDA Application Design and Development, Morgan Kaufmann, 2011.
- [51] L. Lin, M. Fukui, and , S. Tsukiyama, “A GPGPU implementation of parallel  
backward Euler algorithm for power grid circuit simulation,” in proc. IEEE  
International NEWCAS, 2013.