

BMS (バッテリーマネジメントシステム) の 先進アルゴリズム開発

MathWorks Japan

アプリケーションエンジニアリング部

アジェンダ

- **機械学習による予測モデルの作成の概要とMATLAB[®]ソリューション**
 - 例題: 機械学習によるバッテリーの劣化予測モデルの作成
- **非線形カルマンフィルターによるSOC推定**

アジェンダ

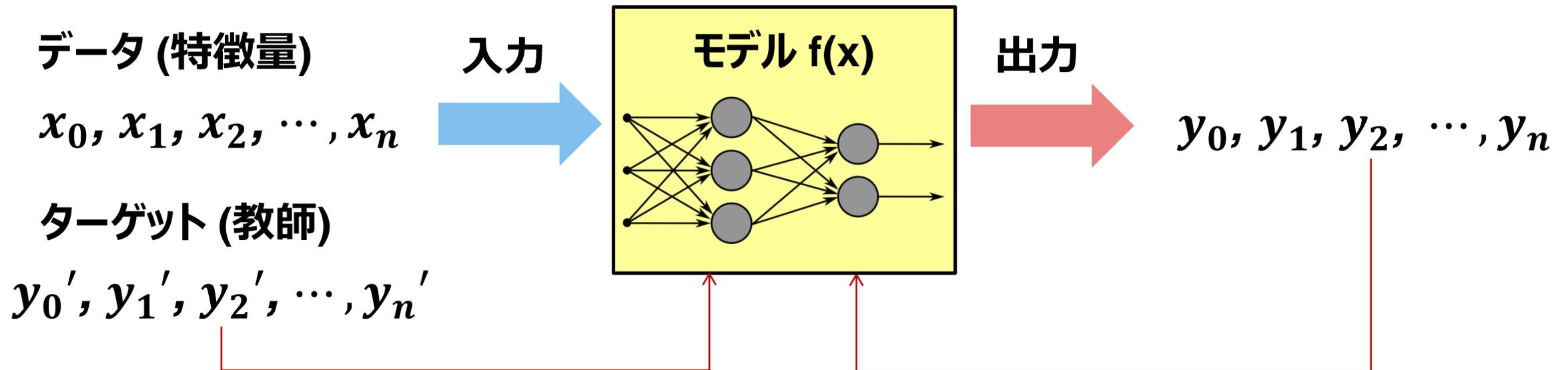
- **機械学習による予測モデルの作成の概要とMATLAB®ソリューション**
 - 例題: 機械学習によるバッテリーの劣化予測モデルの作成
- **非線形カルマンフィルターによるSOC推定**

機械学習による予測モデルの作成の概要

機械学習は、人間ではなく機械が、データに基づいてルールを自動的に決定する技術

(例) ニューラルネットワーク (脳の神経回路の仕組みを模倣したモデル)

$$y = f(x_0, x_1, x_2, \dots, x_n)$$



- ターゲットが項目: 分類
- ターゲットが実数値: 回帰

ターゲットと出力が
一致するように学習

機械学習による予測モデル作成のMATLABソリューション

1

実験による
データ計測と前処理

2

機械学習による
予測モデルの作成

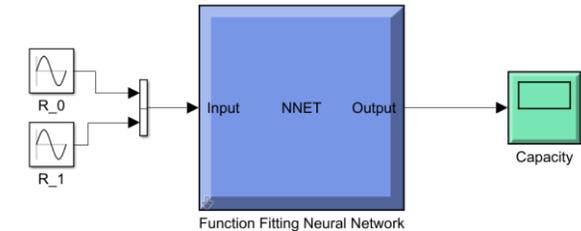
3

予測モデルを
システムモデルに統合

	newTime	1	2	3	4	5	6	7
		Cycle	Type	AmbientTemp	Capacity	RO	RI	TestData
1	2008/04/02 13:08:17	1	charge	24	NaN	NaN	NaN	789x6 table
2	2008/04/02 15:25:41	2	discharge	24	2.0353	NaN	NaN	197x6 table
3	2008/04/02 16:37:51	3	charge	24	NaN	NaN	NaN	940x6 table
4	2008/04/02 19:43:48	4	discharge	24	2.0251	NaN	NaN	196x6 table
5	2008/04/02 20:55:40	5	charge	24	NaN	NaN	NaN	937x6 table

機械学習による回帰

- ニューラルネットワーク
- 線形、非線形回帰
- 決定木、など



計測器からデータ取得

Data Acquisition Toolbox™
Instrument Control Toolbox™



統計・機械学習

Statistics and ...
Machine Learning Toolbox™



ブロック線図

Simulink®



データ前処理

MATLAB®
Signal Processing Toolbox™



深層学習ネットワーク

Deep Learning Toolbox™

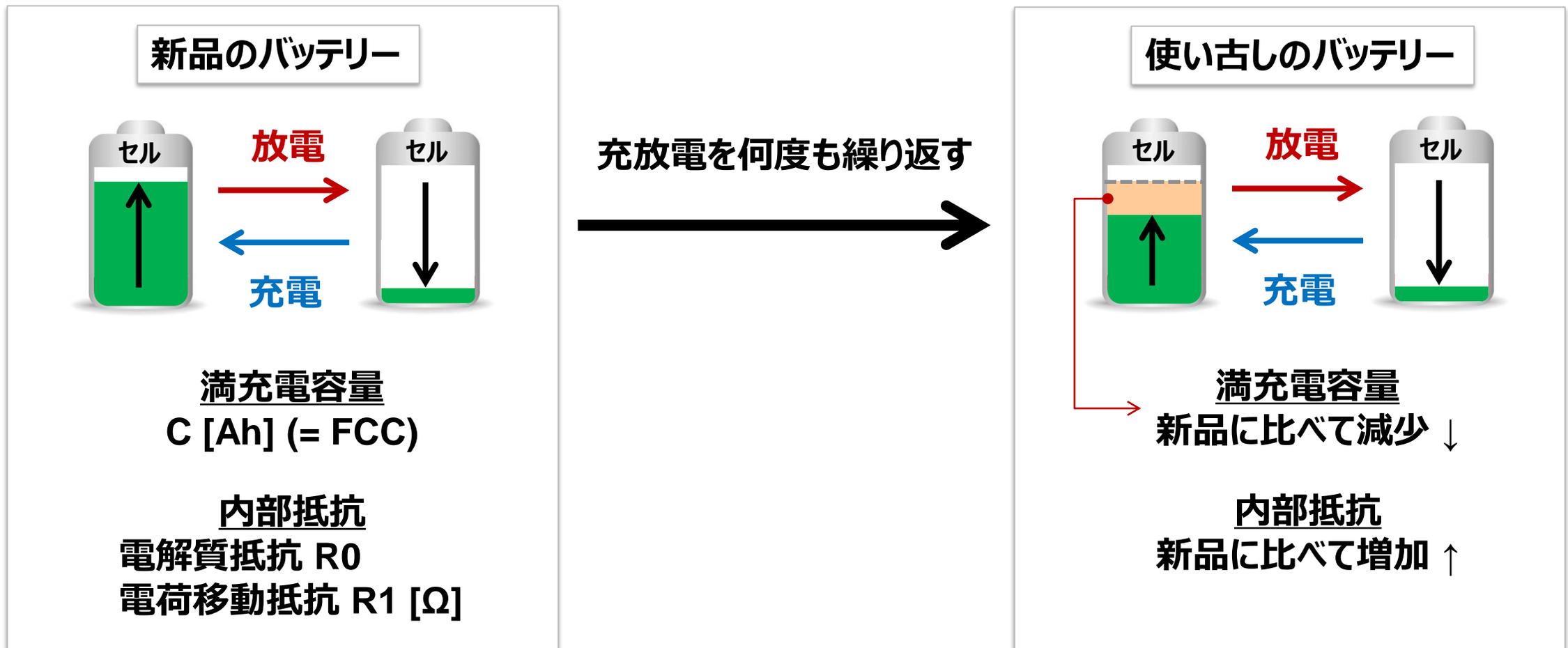
アジェンダ

- **機械学習による予測モデルの作成の概要とMATLAB®ソリューション**
 - 例題: 機械学習によるバッテリーの劣化予測モデルの作成
- **非線形カルマンフィルターによるSOC推定**

バッテリーの劣化は様々な要因が考えられる

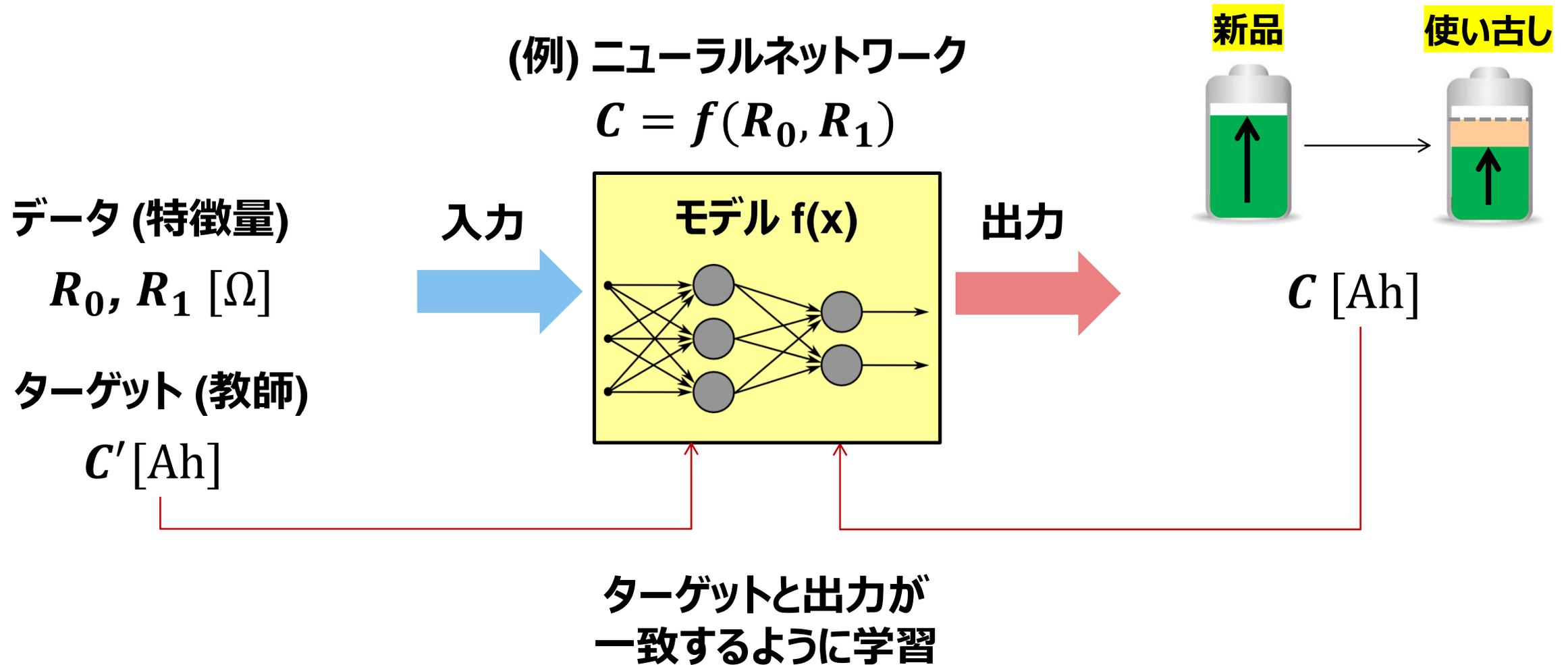
充放電の繰り返しによるサイクル劣化、高温・低温劣化、放置による保存劣化など

(例) 充放電の繰り返しによるサイクル劣化



例題: 機械学習によるバッテリーの劣化予測モデルの作成

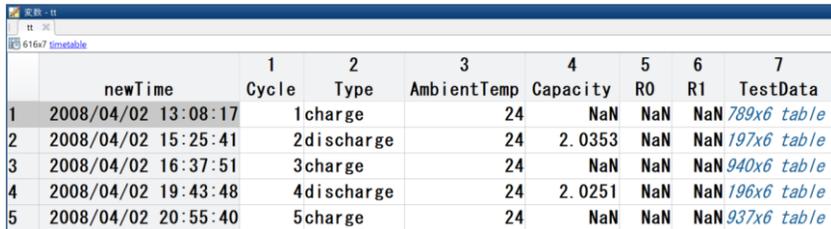
バッテリーの内部抵抗 (R_0 , R_1) から、満充電容量 C の劣化を予測するモデルを構築する



機械学習による予測モデル作成のMATLABソリューション

1

実験による データ計測と前処理



	newTime	1 Cycle	2 Type	3 AmbientTemp	4 Capacity	5 R0	6 R1	7 TestData
1	2008/04/02 13:08:17	1	charge	24	NaN	NaN	NaN	789x6 table
2	2008/04/02 15:25:41	2	discharge	24	2.0353	NaN	NaN	197x6 table
3	2008/04/02 16:37:51	3	charge	24	NaN	NaN	NaN	940x6 table
4	2008/04/02 19:43:48	4	discharge	24	2.0251	NaN	NaN	196x6 table
5	2008/04/02 20:55:40	5	charge	24	NaN	NaN	NaN	937x6 table



計測器からデータ取得

Data Acquisition Toolbox™
Instrument Control Toolbox™



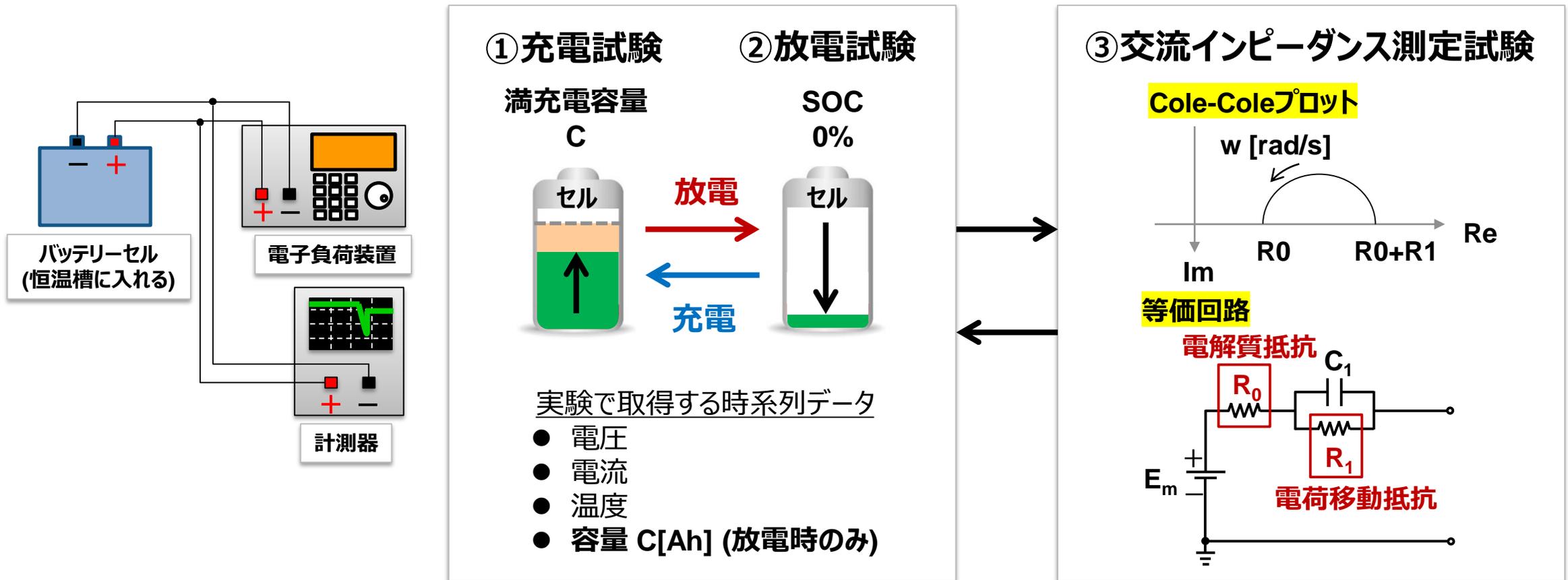
データ前処理

MATLAB®
Signal Processing Toolbox™

バッテリーのサイクル劣化を調べるための実験をする

充電、放電、交流インピーダンス測定の実験を満充電容量がX%低下するまで実施

(例) NASAが実施したリチウムイオンバッテリーのサイクル劣化試験 (以下の3つの試験を繰り返し実施)



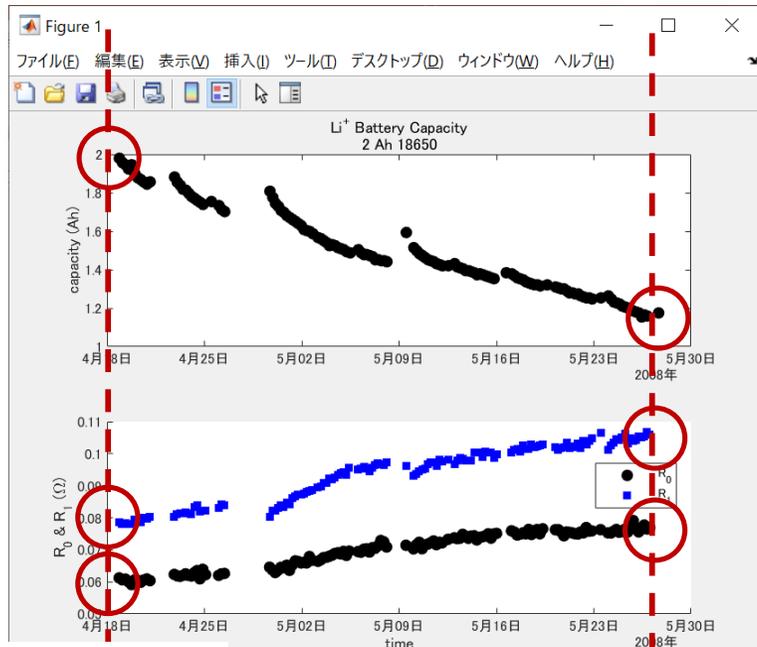
サイクル劣化の実験結果から満充電容量の低下に関係するものを調

充放電の繰り返し回数が増える毎に、満充電容量が低下して内部抵抗が増加する

(例) 充放電の繰り返しによるサイクル劣化

満充電容量
C [Ah]

内部抵抗
R0, R1 [Ω]

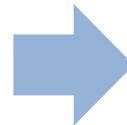


満充電容量
C (新品)



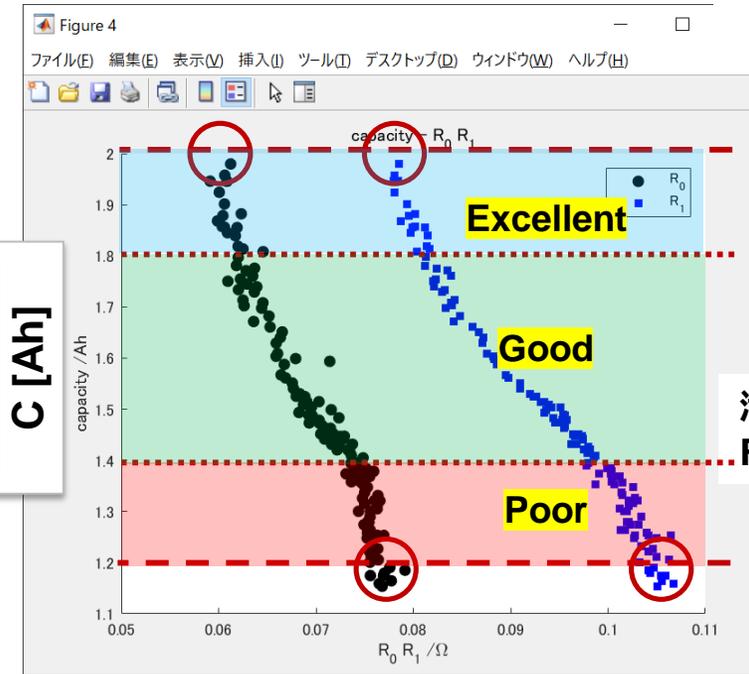
時間
t [日時]

満充電容量
C (劣化)



ターゲット (教師)

満充電容量
C [Ah]



満充電容量
C (新品)



満充電容量
FCC (劣化)



内部抵抗
R0, R1 [Ω]

データ (特徴量)

MATLABのTips: データの取り扱い、グラフ表示がしやすい

table 配列

- 大規模データの管理がしやすい
- ワークスペースに変数が散らばらない
- 異なるデータ型を混在することが可能

```

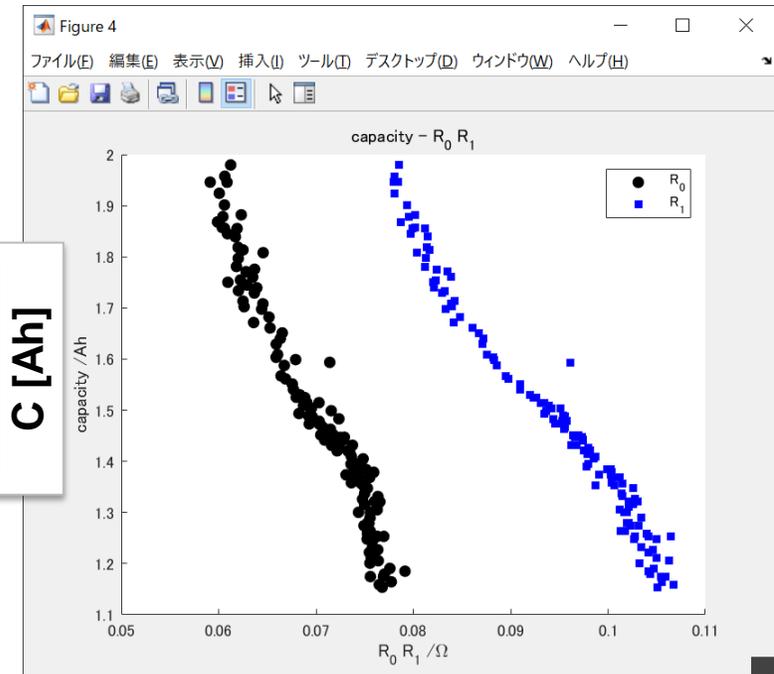
>> tt
tt =
616 × 7 timetable
    newTime      Cycle      Type      AmbientTemp      Capacity      R0      R1      TestData
    _____  _____  _____  _____  _____  _____  _____  _____
    2008/04/02 13:08:17      1      charge      24      NaN      NaN      [ 789 × 6 table]
    2008/04/02 15:25:41      2      discharge  24      2.0353      NaN      NaN      [ 197 × 6 table]
    2008/04/02 16:37:51      3      charge      24      NaN      NaN      NaN      [ 940 × 6 table]
    2008/04/02 19:43:48      4      discharge  24      2.0251      NaN      NaN      [ 196 × 6 table]
    2008/04/02 20:55:40      5      charge      24      NaN      NaN      NaN      [ 937 × 6 table]
    2008/04/03 00:01:06      6      discharge  24      2.0133      NaN      NaN      [ 195 × 6 table]
    2008/04/03 01:12:38      7      charge      24      NaN      NaN      NaN      [ 933 × 6 table]
    2008/04/03 04:16:37      8      discharge  24      2.0133      NaN      NaN      [ 194 × 6 table]
    2008/04/03 05:27:49      9      charge      24      NaN      NaN      NaN      [ 937 × 6 table]
    2008/04/03 08:33:25     10      discharge  24      2.0005      NaN      NaN      [ 194 × 6 table]
    2008/04/03 09:44:35     11      charge      24      NaN      NaN      NaN      [ 952 × 6 table]
    2008/04/03 12:55:10     12      discharge  24      2.0139      NaN      NaN      [ 195 × 6 table]
    
```

日付 サイクル数 タイプ 温度 C[F] R0[Ω] R1[Ω] 諸々の時系列データ

グラフ表示 (2D、3D)

ターゲット (教師)

満充電容量 C [Ah]



内部抵抗 R0、R1 [Ω]

データ (特徴量)

機械学習による予測モデル作成のMATLABソリューション

2

機械学習による 予測モデルの作成

機械学習による回帰

- ニューラルネットワーク
- 線形、非線形回帰
- 決定木、など

**統計・機械学習**

Statistics and ...

Machine Learning Toolbox™

**深層学習ネットワーク**

Deep Learning Toolbox™

モデルを作成する学習用データとモデルを検証するテスト用データに分ける

データ (特徴量)、ターゲット (教師) を設定する

```
x = [R0 R1];
y = C;
```

学習用データとテスト用データに分ける (学習用データ: 80%、テスト用データ: 20%)

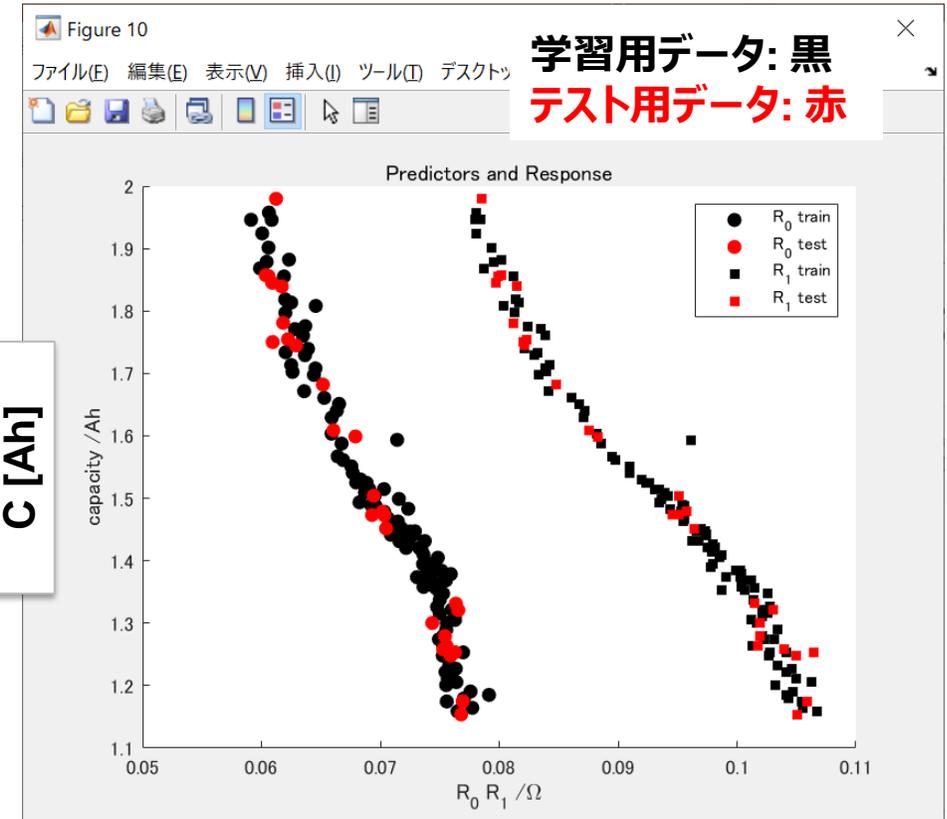
```
testFraction = 0.2;
dataPartition ...
= cvpartition(length(x), 'HoldOut', testFraction);
```

```
Xtrain = x(dataPartition.training,:); % R1 and R0
Ytrain = y(dataPartition.training,:); % Capacity
```

```
Xtest = x(dataPartition.test,:); % R1 and R0
Ytest = y(dataPartition.test,:); % Capacity
```

ターゲット (教師)

満充電容量
C [Ah]



内部抵抗
R0、R1 [Ω]

データ (特徴量)

機械学習によるバッテリーの劣化予測モデルを作成する (ニューラルネットワークを活用)

ニューラルネットワークのモデルを作成して、
学習用データを使い、出力とターゲットが一致するように、
モデル内の重みを調整する。

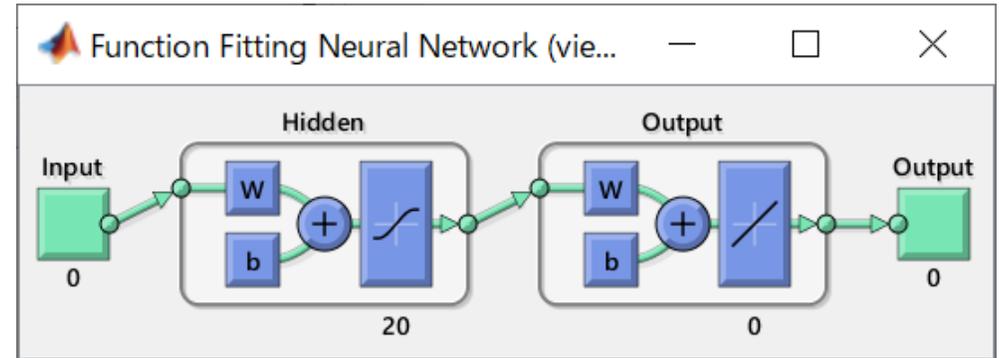
```
trainFcn = 'trainlm';
hiddenLayerSize = 20;
net = fitnet(hiddenLayerSize,trainFcn);
view(net)
mdl_net = train(net,Xtrain',Ytrain', ...
'UseParallel','no');
```

テスト用データとモデルの出力において、
平均絶対パーセント誤差を求めて、モデルの精度を確認する。

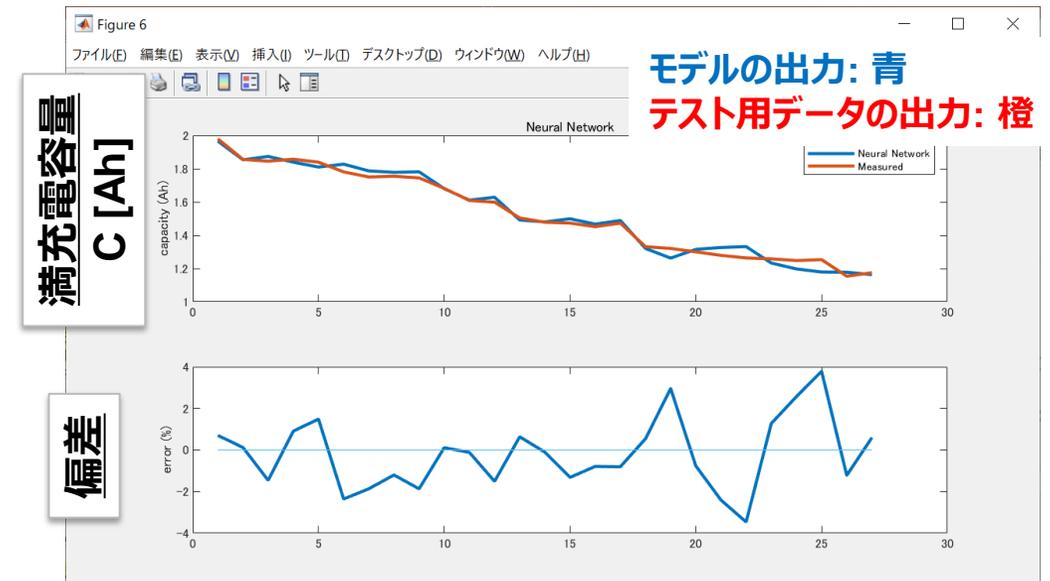
```
fprintf('Neural Net Training set MAPE: %0.2f%%\n', ...
mape(Ytest, mdl_net(Xtest'))*100);
```

バッテリーの劣化予測モデルの精度が良好かどうかを、
テスト用データを使い、可視化して検証する。

```
Y_nn = mdl_net(Xtest)';
f6 = plotPrediction(Y_nn,Ytest,6,'Neural Network');
```



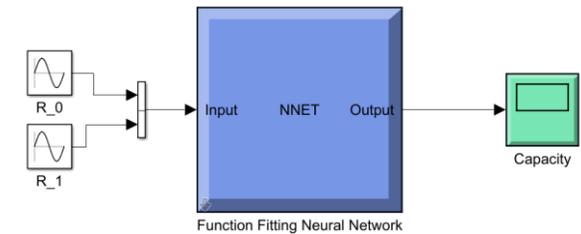
Neural Net Training set MAPE: 1.98%



機械学習による予測モデル作成のMATLABソリューション

3

予測モデルを
システムモデルに統合

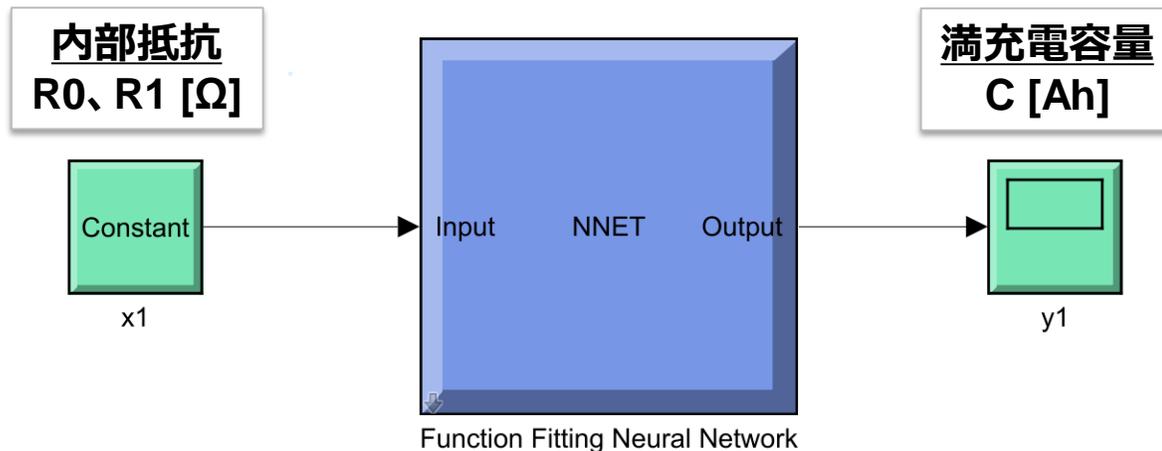


 **ブロック線図**
Simulink®

バッテリーの劣化予測モデルをSimulink環境にインポート

ニューラルネットワークのモデルを
Simulinkモデルに変換する。

gensim mdl_net)



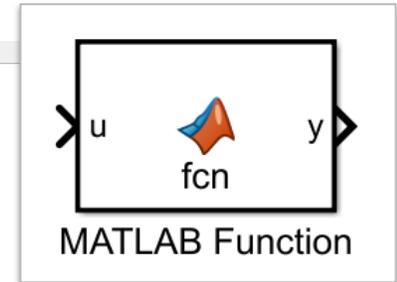
ニューラルネットワークのモデルを
MATLAB関数に変換する。

genFunction mdl_net)

```

1 function [Y,Xf,Af] = neural_function(X,~,~)
2 %NEURAL_FUNCTION neural network simulation function.
3 %
4 % Auto-generated by MATLAB, 29-Aug-2019 16:14:24.
5 %
6 % [Y] = neural_function(X,~,~) takes these arguments:
7 %
8 % X = 1xTS cell, 1 inputs over TS timesteps
9 % Each X[1,ts] = 2xQ matrix, input #1 at timestep ts.
10 %
11 % and returns:
12 % Y = 1xTS cell of 1 outputs over TS timesteps.
13 % Each Y[1,ts] = 1xQ matrix, output #1 at timestep ts.
14 %
15 % where Q is number of samples (or series) and TS is the number of timesteps.

```



活用例

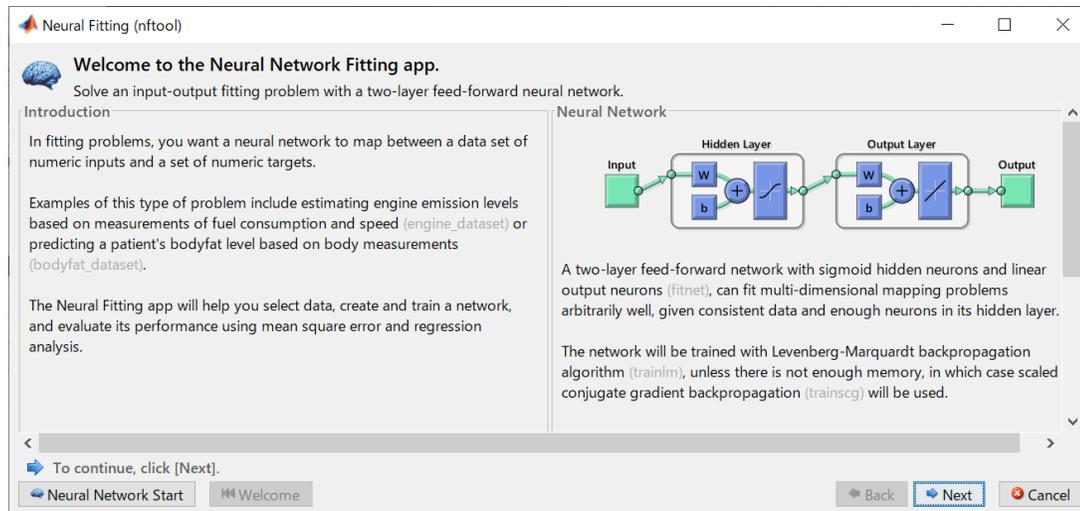
バッテリーの電圧V、電流Iなどの情報から、内部抵抗 R_0 、 R_1 を推定し、劣化予測モデルで満充電容量 (C) を求めて、バッテリーがどれぐらい劣化しているかを調べる。

前半まとめ

機械学習を活用したバッテリーの劣化予測アルゴリズムもMATLABで開発し、それをSimulinkを使ったシステム検証用のモデルにシームレスに展開可能です。

機械学習を活用したアルゴリズム開発

MATLABコマンドベースのアプローチ以外に、GUIベースのアプローチもあります。



<https://www.mathworks.com/help/deeplearning/gs/fit-data-with-a-neural-network.html>

機械学習とMATLAB活用例



機械学習は、近年大きな注目を集めている「AI」、「人工知能」、「ディープラーニング」といった研究分野と深い関わりがあります。機械学習は、人間や動物が経験を通して自然に学習することをコンピューターにさせようとするデータ解析テクニックです。機械学習アルゴリズムは所定の方程式をモデルとして用いることなく、データから直接的に情報を「学習」するコンピューティング手法です。アルゴリズムは、学習に利用可能なサンプル数が増加するにつれて適応的にその性能を改善します。

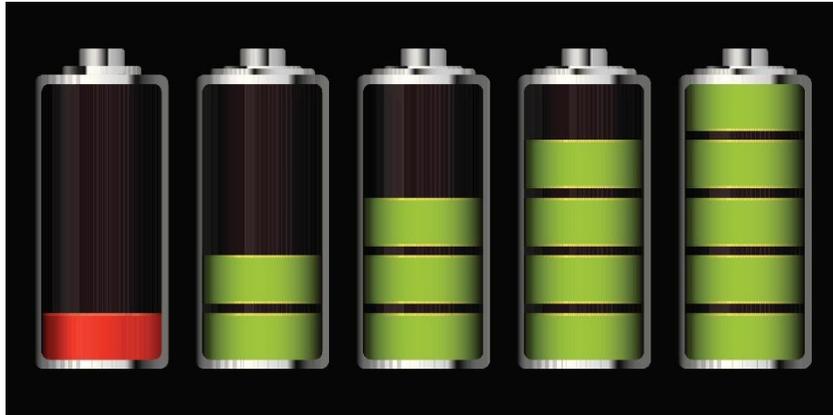
<https://jp.mathworks.com/discovery/machine-learning.html>

アジェンダ

- **機械学習による予測モデルの作成の概要とMATLAB®ソリューション**
 - 例題: 機械学習によるバッテリーの劣化予測モデルの作成
- **非線形カルマンフィルターによるSOC推定**

目的

- バッテリーのSOC (State Of Charge) を推定する

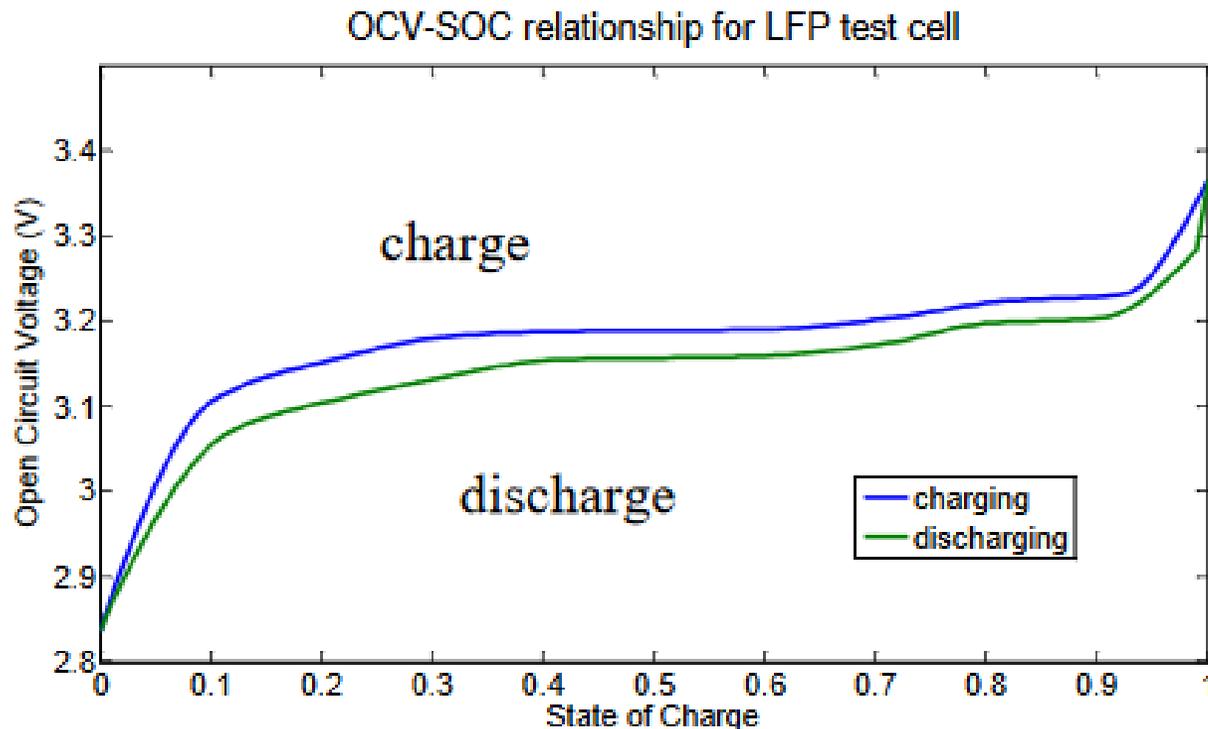


SOC

SOC (State Of Charge) とは、電池の残容量を表す指標

$$SOC = \frac{C_r}{C_f} \times 100[\%]$$

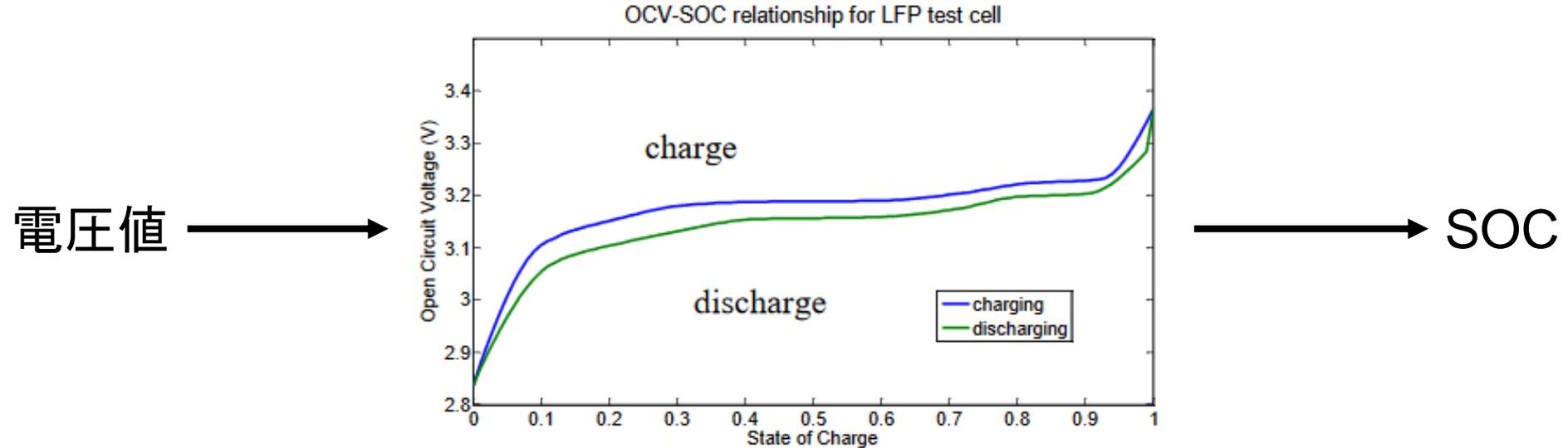
C_r は残容量[Ah]、 C_f は満充電容量[Ah]



SOCの特性は非線形
かつ
直接測れない

SOCの推定方式

- 電圧測定方式
 - 電圧-SOC特性から残量を求める

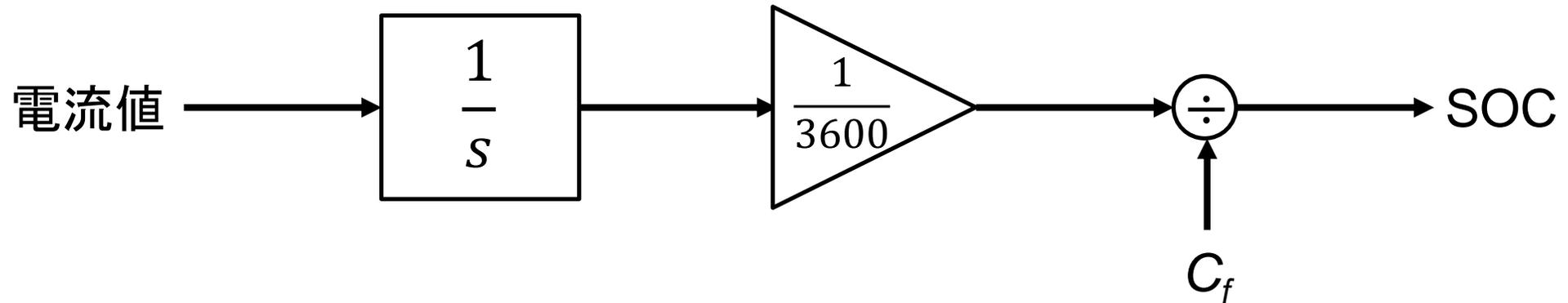


デメリット: 精度が低い

(電流や温度によってバッテリーの特性は変化するため)

SOCの推定方式

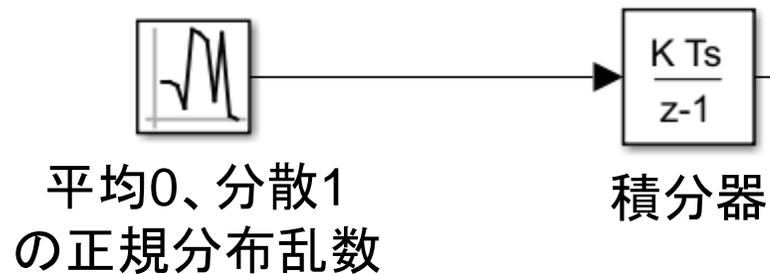
- クーロン・カウンタ方式
 - 電流を積分して残量を求める



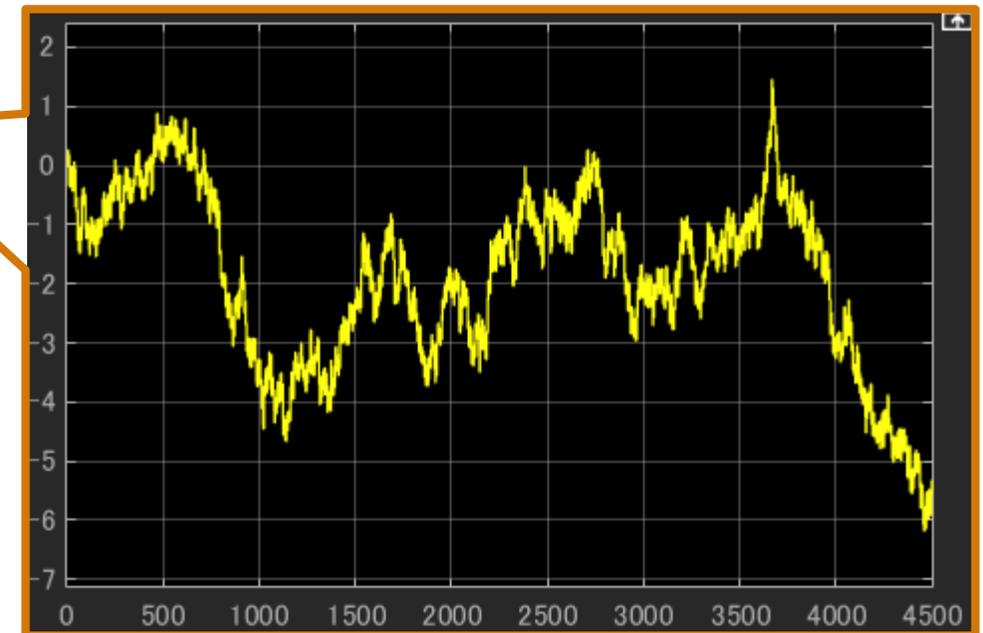
デメリット: 短時間の精度は高いが、長時間ではドリフトが発生する

(参考)ランダムウォーク

- クーロン・カウンタ方式では、電流を積分してSOCを求める。このとき、電流値が完全に一致していればドリフトは発生しないが、そこにノイズが加わるだけでもドリフトは発生する。これはランダムウォークの作用によるものである。
- 単純にノイズを乱数で表現し、それを積分すると以下のようなになる

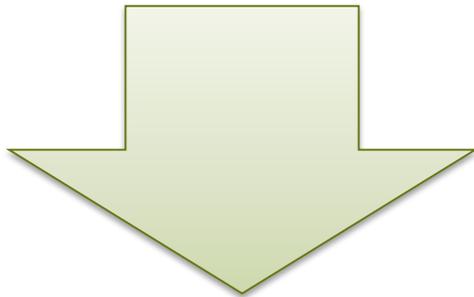


乱数は平均的には0だが、
積分後は0でない値に至っている



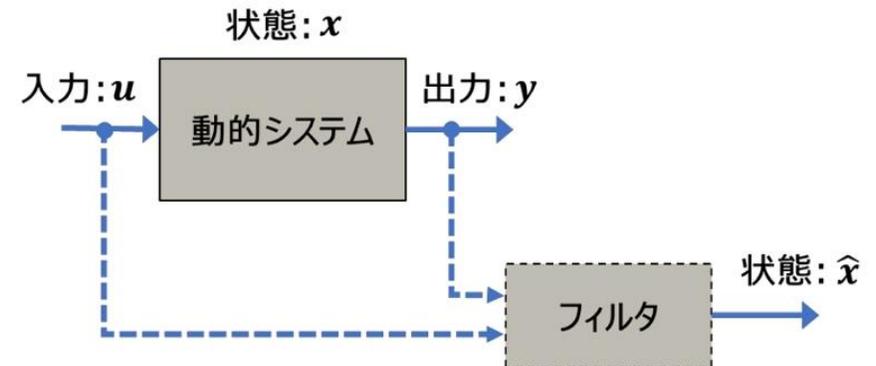
モデルを使ってSOCを推定する

- 電圧、電流、加えて温度も計測できるとして、バッテリーモデルを使って精度よく推定したい



- カルマンフィルターによる推定

- 逐次ベイズフィルターの一種であり、測定データからシステムの状態を実時間で推定するアルゴリズム
- コンピュータービジョン、誘導・航法システム、計量経済学、信号処理などで広く使われており、多くの実績がある



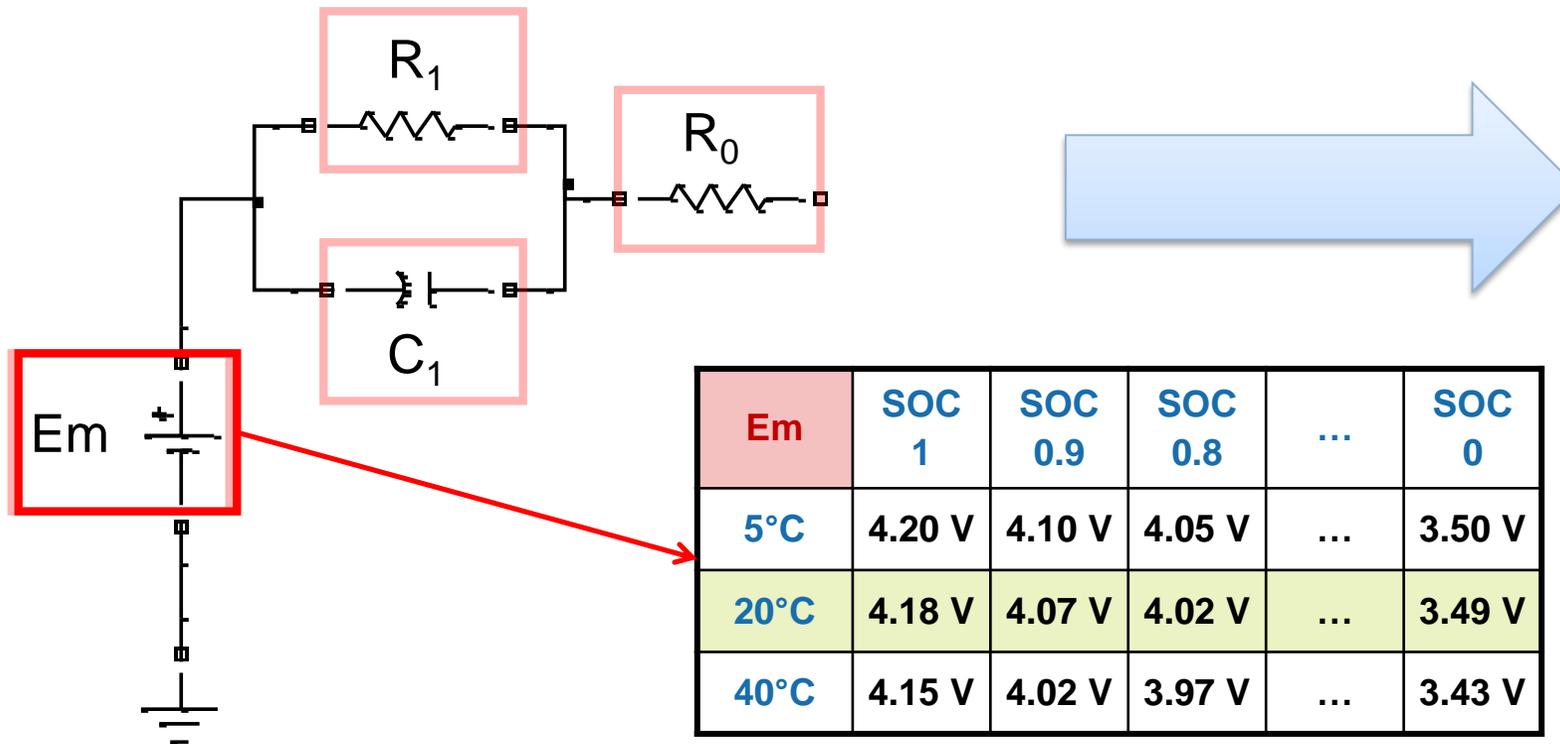
カルマンフィルター構築のワークフロー

1. モデル化(定式化)
2. カルマンフィルター用の状態方程式、出力方程式を立てる
3. カルマンフィルター計算式を実装(Simulinkモデル化)
4. パラメータチューニング
5. 量産用に実装・検証

バッテリーモデル

- 最初に入力 u 、状態 x 、出力 y を決める

バッテリーセルの等価回路



入力 u : 電流 I 、温度 T_b
 状態 x : C_1 にかかる電圧 V_{C1} 、
 充電量SOC
 出力 y : セル端電圧 E

状態方程式と出力方程式

- 状態方程式

$$\begin{bmatrix} SOC[k+1] \\ V_{C1}[k+1] \end{bmatrix} = \begin{bmatrix} SOC[k] \\ V_{C1}[k] \end{bmatrix} + \begin{bmatrix} -\frac{I}{3600C_q} \\ \frac{I}{C_1(SOC[k], T_b[k])} - \frac{V_{C1}[k]}{R_1(SOC[k], T_b[k])C_1(SOC[k], T_b[k])} \end{bmatrix} Ts + W_p$$

- 出力方程式

$$E[k] = E_m(SOC[k], T_b[k]) - V_{C1}[k] - IR_0(SOC[k], T_b[k]) + W_o$$

Iは電流、 T_b は温度、 V_{C1} は C_1 にかかる電圧、SOCは充電量、Eはセル端電圧、 C_q はバッテリーの容量、 C_1 、 R_1 、 R_0 、 E_m はSOCと温度に依存して変化するキャパシタ、抵抗、内部電圧、 W_p はプロセスノイズ、 W_o は観測ノイズ、Tsはサンプリングタイムステップ。

(参考)カルマンフィルターの計算式

- 拡張カルマンフィルターの一般式

$$\underline{\hat{\mathbf{x}}^- [k - 1] = \mathbf{f}(\mathbf{x}[k - 1])}$$

$$\underline{\mathbf{A}[k - 1] = \left. \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}[k-1]}, \quad \mathbf{C}[k] = \left. \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}^- [k]}}$$

推定するシステムごとに
変わる計算式

$$\mathbf{P}^- [k] = \mathbf{A}[k - 1]\mathbf{P}[k - 1]\mathbf{A}^T [k - 1] + \mathbf{BQB}$$

$$\mathbf{G}[k] = \mathbf{P}^- [k]\mathbf{C}^T [k](\mathbf{C}[k]\mathbf{P}^- [k]\mathbf{C}^T [k] + \mathbf{R})$$

$$\hat{\mathbf{x}}[k] = \hat{\mathbf{x}}^- [k] + \mathbf{G}[k](\mathbf{y}[k] - \underline{\mathbf{h}(\hat{\mathbf{x}}^- [k])})$$

$$\mathbf{P}[k] = (\mathbf{I} - \mathbf{G}[k]\mathbf{C}[k])\mathbf{P}^- [k]$$

変わらない式はライブラリで計算できる

非線形カルマンフィルター

- 状態方程式、出力方程式が非線形であるため、非線形に対応したカルマンフィルターを使う

拡張カルマンフィルター (EKF)

- 計算効率が良い
- 非線形性が強いと不安定になりやすい
- 方程式を状態 x で微分した係数行列が必要



Control System Toolboxの拡張カルマンフィルターブロックでは、係数行列を用意しなくても逐次計算で求めることができます。

アンセンテッドカルマンフィルター (UKF)

- 計算効率が良い
- 非線形性が強くても安定
- 方程式を微分する必要なし

ちなみに計算時間は
「係数行列を用いたEKF」 < 「UKF」
「逐次計算のEKF」 \doteq 「UKF」

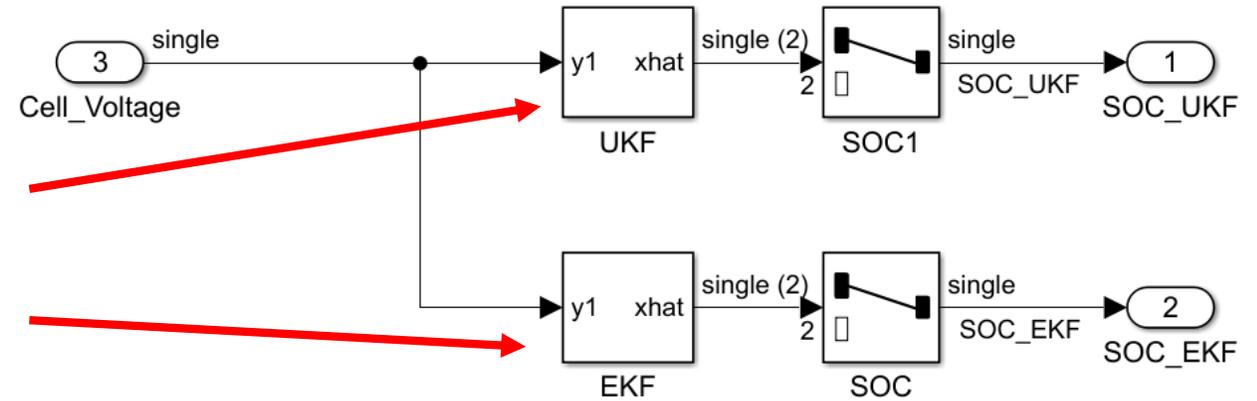
実装 (Simulinkモデル化)

※比較のために二つとも実装しています。最終的にはどちらか一つでよいです。



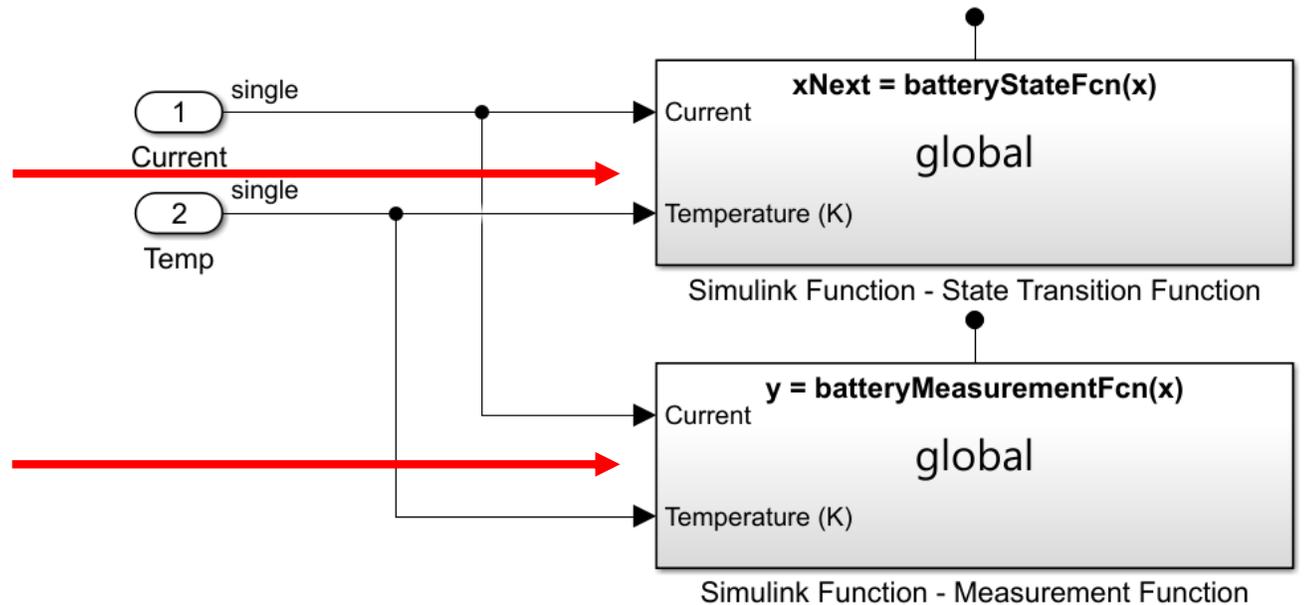
UKFのブロック

EKFのブロック



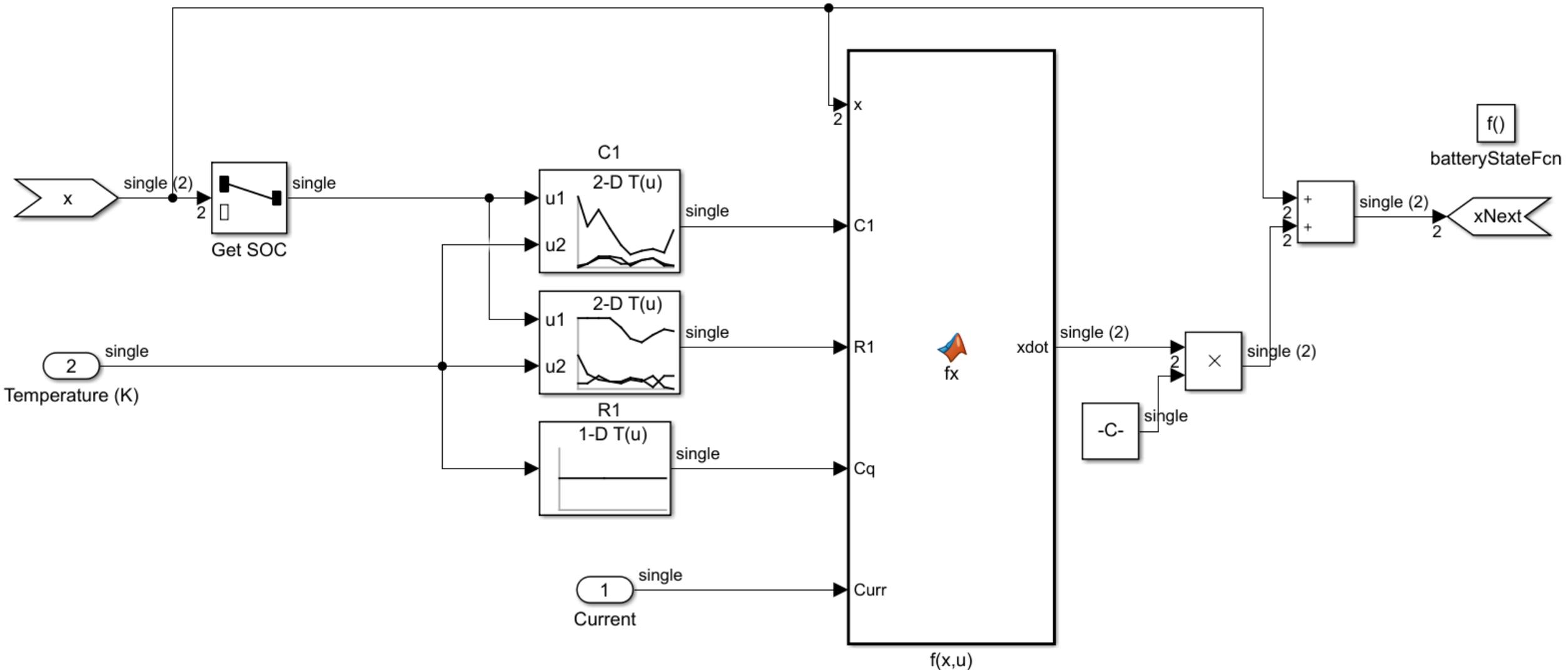
状態方程式

出力方程式



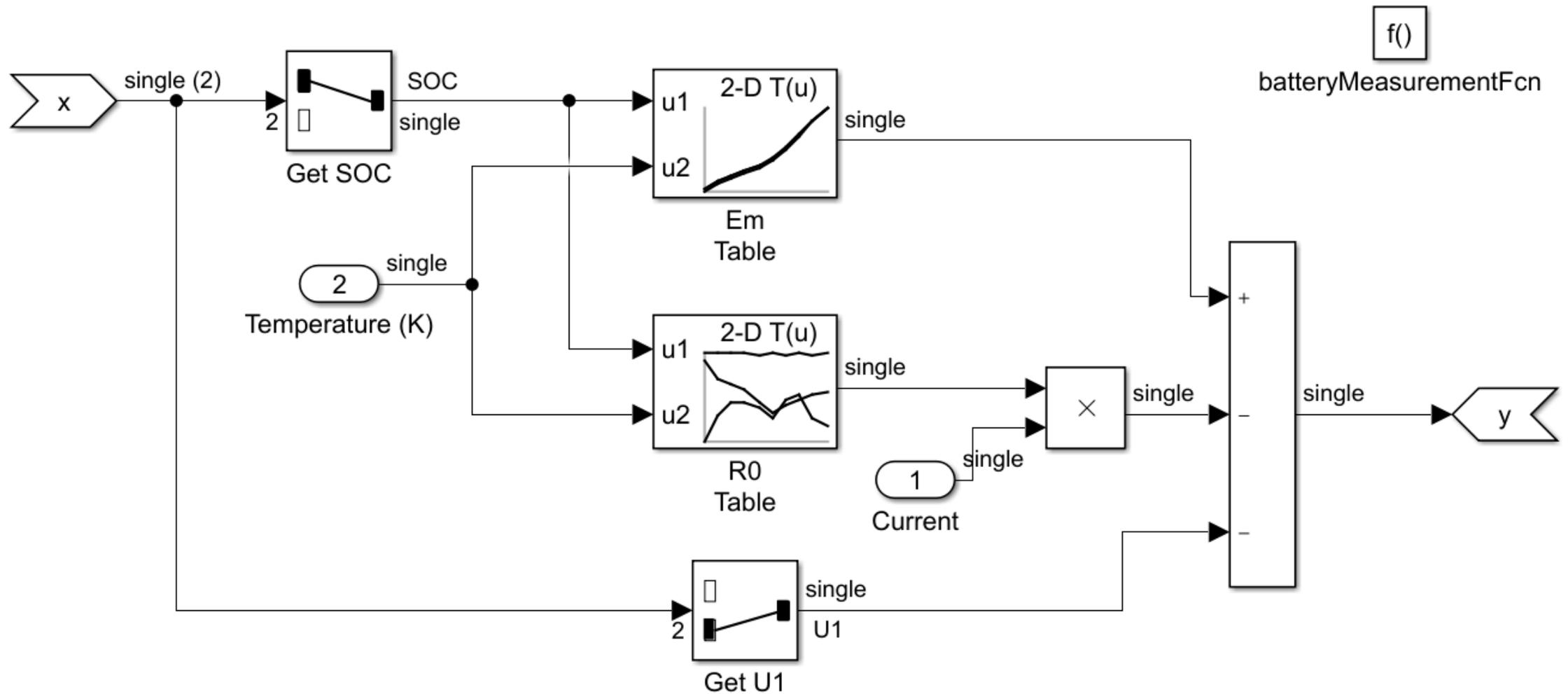
実装 (Simulinkモデル化)

状態方程式



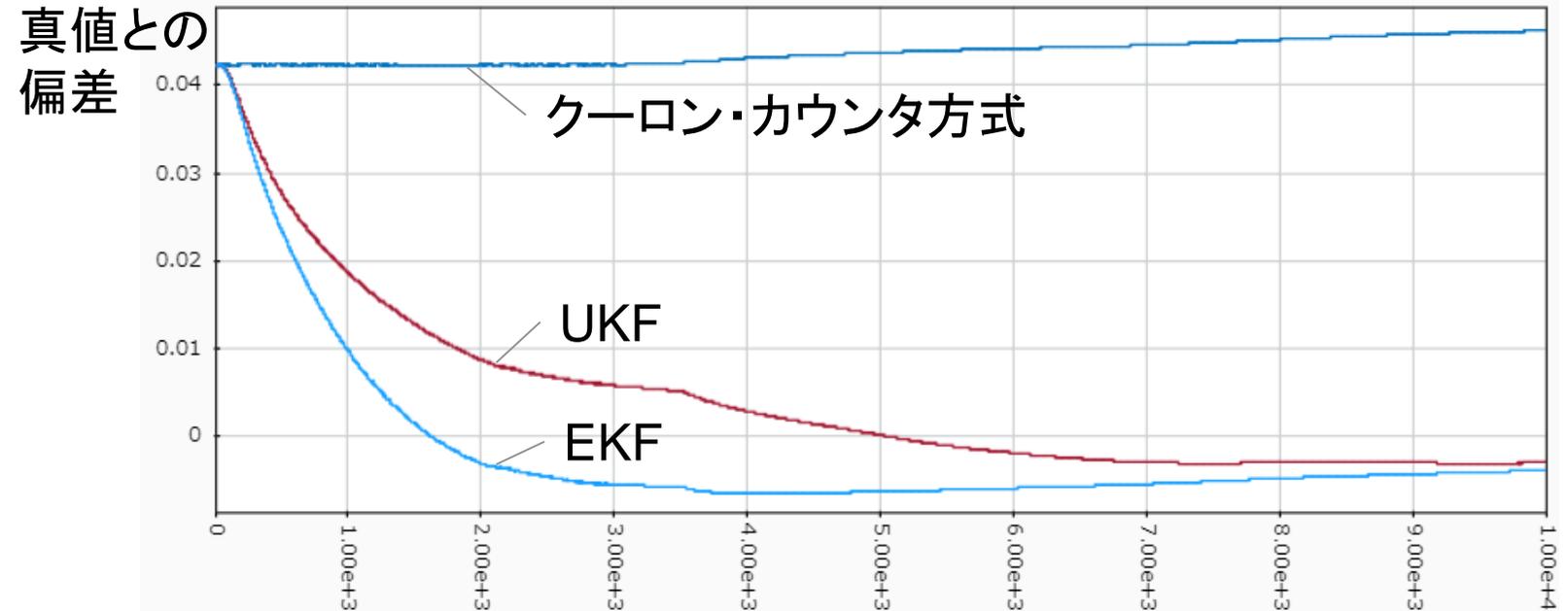
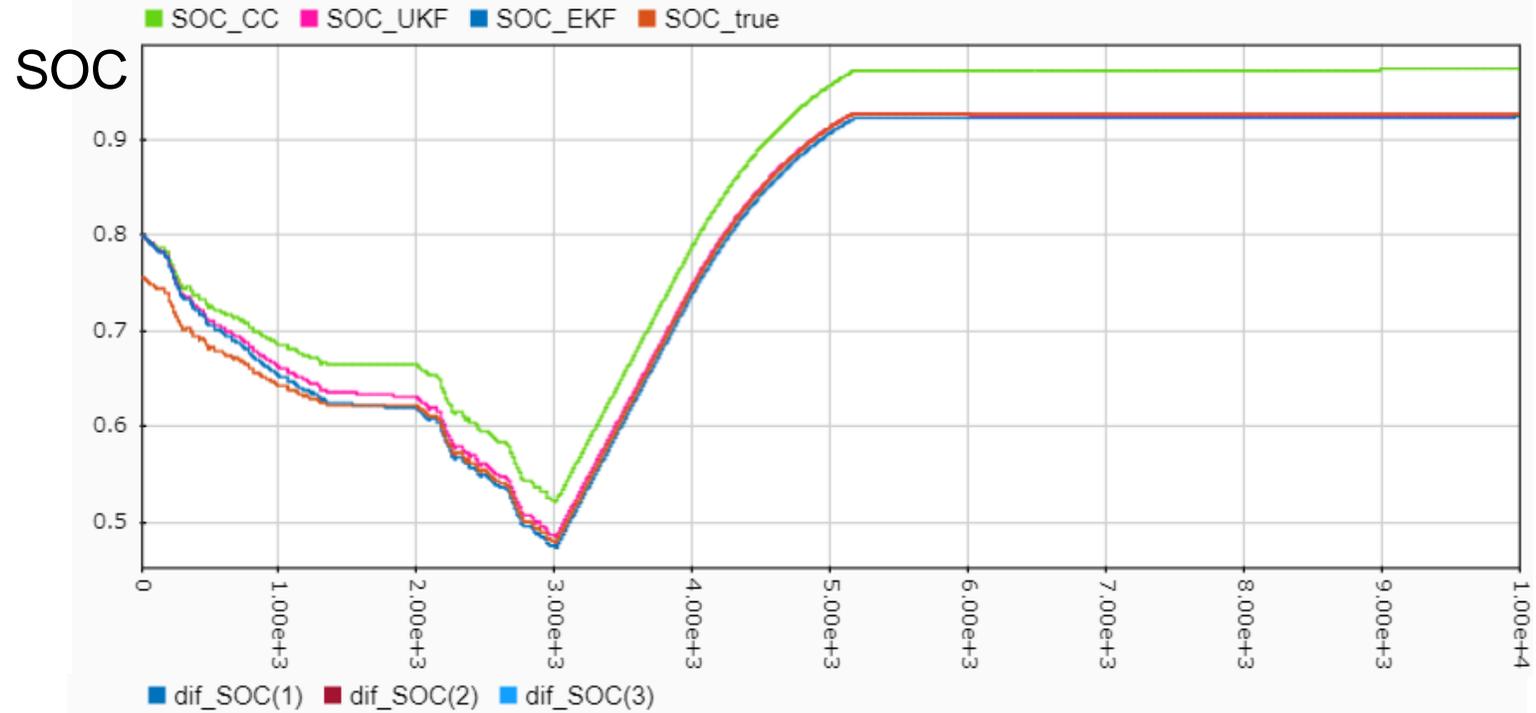
実装 (Simulinkモデル化)

出力方程式



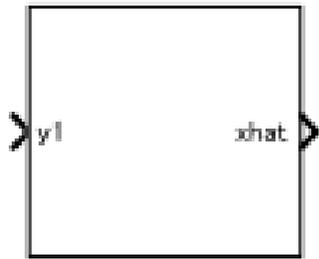
結果

- UKF、EKF共に偏差はSOC1%以内に収束

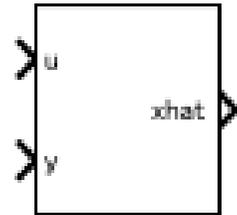


実装 (量産コード生成)

- Control System Toolboxのカルマンフィルタブロックは、Embedded CoderによるC/C++コード生成に対応しています



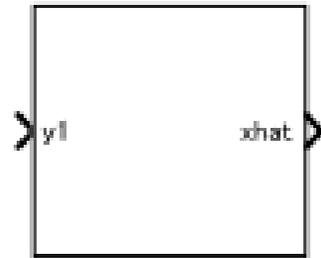
Extended Kalman Filter



Kalman Filter



Particle Filter



Unscented Kalman Filter

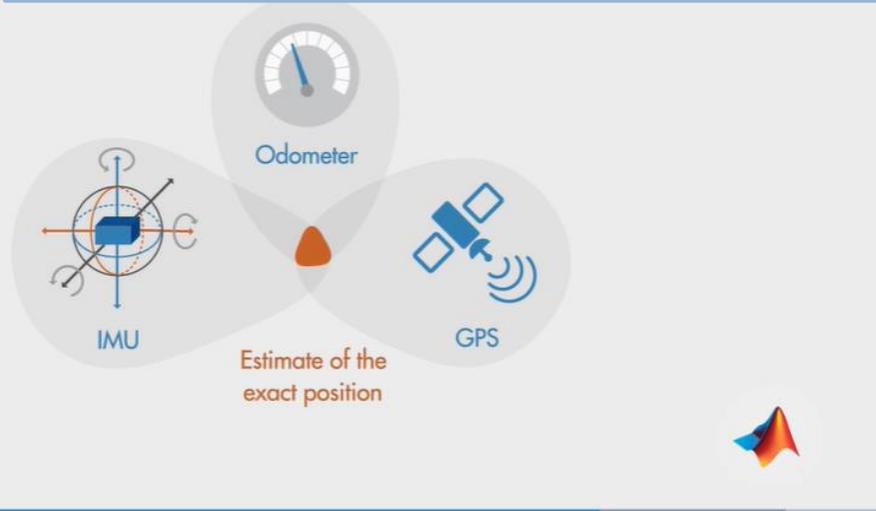
```
コード
Estimator.c 検索
162
163 /* Outputs for Enabled SubSystem: '<S5>/Correct1' incorporates:
164  * EnablePort: '<S53>/Enable'
165  */
166 /* MATLAB Function: '<S53>/Correct' incorporates:
167  * Constant: '<S5>/R1'
168  * DataStoreRead: '<S53>/Data Store ReadX'
169  * DataStoreWrite: '<S53>/Data Store WriteP'
170  * Inport: '<Root>/y1'
171  */
172 UKFCorrectorAdditive_getPredict(0.001, Estimator_DW.x, Estimator_DW.P, &y,
173     gain, &Pyy);
174 y = Estimator_U.y1 - y;
175 gain[0] /= Pyy;
176 yCov = gain[1] / Pyy;
177 rtb_Akxhatk1 = gain[0] * Pyy;
178 Estimator_DW.P[0] -= rtb_Akxhatk1 * gain[0];
179 Pyy *= yCov;
180 Estimator_DW.P[1] -= Pyy * gain[0];
181
182 /* DataStoreWrite: '<S53>/Data Store WriteX' incorporates:
183  * DataStoreRead: '<S53>/Data Store ReadX'
184  * MATLAB Function: '<S53>/Correct'
185  */
186 Estimator_DW.x[0] += gain[0] * y;
187
```

Ln 171 Col 3

後半まとめ

非線形特性を持つSOCを推定するカルマンフィルタを設計し、Simulinkでシミュレーション検証を行い、量産Cコード生成が可能です。

カルマンフィルタをより学びたい人には、Tech Talksビデオがお勧めです。また、Discoveryページでは、様々な実例を見ることができます。



Description

Full Transcript

Code and Resources

Understanding Kalman Filters, Part 1: Why Use Kalman Filters?

<https://jp.mathworks.com/videos/series/understanding-kalman-filters.html>

カルマンフィルタ

サイト内検索

Examples and How To

- インターンがモデルベースデザインでQuadcopter用GNCソフトウェアを開発 - NASA
- モデルベースデザインによる近距離撮影のための無人航空システム開発 - Airnamics
- モデルベースデザインによる仮想センサーアルゴリズム、サーボドライブのパフォーマンスを向上 - B&R Industrial Automation
- モデルベースデザインによるX-43Aスクラムジェットのマッハ10速度達成 - NASA

Software Reference

- カルマンフィルタを用いた物体トラッキングの紹介 (13:58) - ビデオ
- MATLABによるアルゴリズム開発 ～ 粒子フィルタの応用 ～ (45:35) - ビデオ
- 自動運転・ADASの開発・検証プラットフォーム : Automated Driving Toolbox (24:21) - ビデオ

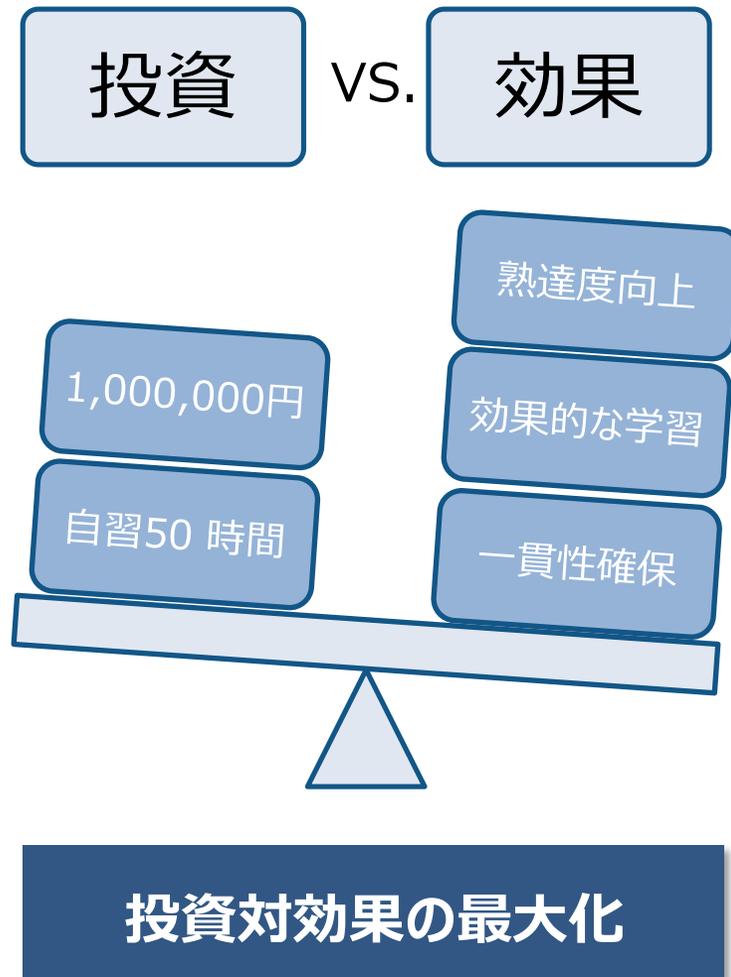
<https://jp.mathworks.com/discovery/kalman-filter.html>



Accelerating the pace of engineering and science

© 2020 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

トレーニング・コンサルティングサービス



トレーニングサービス

MathWorks製品の機能の理解や使い方を、短期間で最大限に習得して頂けるような、多種多様な教育カリキュラムを提供します。

<http://jp.mathworks.com/services/training/>

コンサルティングサービス

お客様のプロジェクトを成功させるため、お客様のモデル・データを使った、コンサルティングサービスを提供します。

<http://jp.mathworks.com/services/consulting/>