Systems Dynamics

Course ID: 267MI – Fall 2023

Thomas Parisini
Gianfranco Fenu

University of Trieste
Department of Engineering and Architecture

267MI –Fall 2023

Course Overview

Course Administration

Lecturers & examiners

- Thomas Parisini (parisini@units.it)
- Gianfranco Fenu (fenu@units.it)

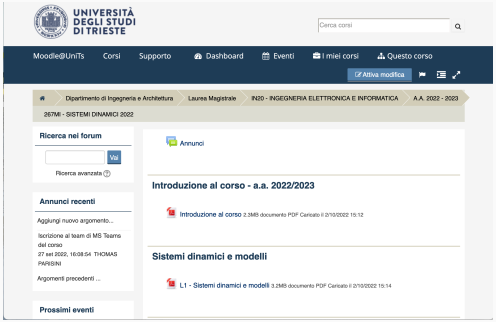
Course home page

- slides, exercises and computer code examples
- old exams

Systems Dynamics homepage

Course credits

- 9 CFU



Aggiungi nuovo argomento...

Iscrizione al team di MS Teams del corso
27 set 2022, 16:08:54 THOMAS PARISINI

Argomenti precedenti ...

Introduzione al corso - a.a. 2022/2023

Introduzione al corso 2.3MB documento PDF Caricato il 2/10/2022 15:12

Sistemi dinamici e modelli

L1 - Sistemi dinamici e modelli 3.2MB documento PDF Caricato il 2/10/2022 15:14

Examination: Intermediate Tests

Partial tests (highly recommended)

- Three **partial tests** are carried out to evaluate the competences acquired by the students enrolled in the course.
- The partial tests have an **open-book format** and are carried out by the students during the course timetable.
- The duration of each partial test is of one hour.
- The partial tests are strictly personal and group work is not allowed.

- Each partial evaluation will consist in the solution of a specific problem via the creation of a Matlab *livescript*.
- The *livescript* will consist of the code solving the problem and an explanation of the solution method.

Examination: Intermediate Tests (cont.)

Partial tests mark & cumulative mark

- Each partial test gets an individual mark P_{test_i} ranging from 1 to 10.
- Given

$$P_{\text{cum}} = \sum_{i=1}^{3} P_{\text{test}_i}$$

the cumulative mark P_{tot} is computed according to the formula:

$$P_{\text{tot}} = \begin{cases} P_{\text{cum}} & \iff \quad P_{\text{cum}} \leq 24 \\ 24 + 0.5 \cdot \left(P_{\text{cum}} - 24\right) & \iff \quad P_{\text{cum}} > 24 \end{cases}$$

Examination: Intermediate Tests (cont.)

In case P_{tot} is not an integer, P_{tot} is rounded up.

Partial tests & final grade

- A positive mark on each test ($P_{test_1} \geq 6$, $P_{test_2} \geq 6$ and $P_{test_3} \geq 6$) allows to register the final mark P_{tot} at the student discretion.

In case the final mark $P_{tot} \geq 18$, the following options are available at the student discretion:

- **Register** the final mark P_{tot} ;
- Carry out an **additional (open book) partial written examination** with a **maximum mark** equal to 5 points and the registration of the cumulative mark at the student discretion;
- Carry out the **full examination**.

Examination: Intermediate Tests (cont.)

Partial Tests Grade Expiration

- Expiration: using the partial tests cumulative mark P_{tot} is allowed during the current academic year (**for the academic year 2023/2024** until the end of the examination session in **February 2025**).
- When the exam sessions of the current academic year are over, the cumulative mark P_{tot} expires.

Examination: Full Final Examination

Full final examination

- **Full final exam**: a preliminary written examination followed by oral questions.
- The **final grade** depends on both the written part and the outcome of the oral discussion.
- Written examination and oral discussion usually usually take during the same exam session.

Full Final Examination

Written examination

The test consists of $2-3$ essay questions:

- usually, the student is asked to solve numerical application problems.
- The written examination has an **open-book format**.

Oral questions

Oral questions deal with any possible topic, discussed and analyzed in the lectures.

- A short discussion about the written examination results generally also takes place.

Exam Sessions

Examination timetable

- 3 sessions in January–February
- 2 sessions in June–July
- 2 session in September

How to sign up for examinations

- In order to participate to the exam session you must sign up/register for the exam **(compulsory)**
- To sign up, use the students university career management system Esse3 to access to the on-line University Services.
- Please, **pay attention** to the dates of the registration periods and the examination periods!

Course Information

Prerequisites

- Linear algebra, calculus and complex analysis
- Course 034IN “Fundamentals of automatic control” (or equivalent for students enrolled from other universities/programs)
- Basic knowledge of probability and statistics is not mandatory, but highly helpful

Course organization

- Lectures
- Hands-on exercises using Matlab *live scripts*
  - partly discussed in an interactive way during the lectures
  - partly made available as stand-alone exercises
- Exercise sessions

Desiderata

Students who pass the course should be able to:

- carry out a complete and comprehensive analysis of the main properties of deterministic and stochastic discrete-time dynamic systems;
- design and implement parametric estimation and identification, and state estimation algorithms that use available data or data collected in real-time with reference to engineering application scenarios;
- . . .

Desiderata (cont.)

Students who pass the course should be able to

- evaluate, among several options, what’s the best choice of parametric estimation and identification, and state estimation algorithms starting from requirements and considering technological constraints;
- describe in a clear and plain way the functionalities of a parametric estimation and identification, and state estimation algorithm in the context of discrete-time dynamic systems and with the correct use of technical terminology

Lectures Plan

Lect.	Content
1	Course overview. Generalities: systems and models (defs, props, problems).
2	Sampling and discrete-time representation of linear continuous-time dynamic systems.
3	Time-evolution of state and output of linear dynamic systems.
4	Stability of discrete-time dynamic systems.
5	Model identification from data.
6	A glimpse on prob. theory, random vars and discrete-time stochastic processes.
7	Definitions and properties of the estimation and prediction problems.
	. . .

Lectures Plan (cont.)

Lect.	Content
8	Dynamic models of stationary discrete-time stochastic processes.
9	Least-squares estimation.
10	Bayes estimation.
11	Solution of the prediction problem.
12	Identification Based on Prediction Error Minimization (PEM).
13	Batch PEM Identification Algorithms.
14	State estimation from observed data.

References

References on dynamic systems analysis:

P. J. Antsaklis and A. N. Michel.
Linear Systems.
Birkhäuser, 2006.

G. Calafiore.
Elementi di Automatica.
CLUT, Torino, 2007.
(in Italian).

G. Marro.
Teoria dei Sistemi e del Controllo.
Zanichelli, 1989.
(in Italian).

S. Rinaldi.
Teoria dei Sistemi.
CLUP, Milano, 1977.
(in Italian).

References (cont.)

References on data-based estimation and identification:

T. Söderström. P. Stoica.
System Identification.
Prentice Hall, 1989.

L. Ljung.
System Identification – Theory for the User.
Prentice Hall, 1999.

S. Bittanti.
Model identification and data analysis.
John Wiley & Sons, 2019.

J. Schoukens, R. Pintelon, and Y. Rolain.
Mastering System Identification in 100 ***Exercises.***
IEEE Press, 2012.

References (cont.)

References on data-based estimation and identification (cont.):

S. Bittanti.
Teoria della predizione e del filtraggio.
Pitagora Editrice, Bologna, 2000.
(in Italian).

S. Bittanti.
Identificazione dei Modelli e Controllo Adattativo.
Pitagora Editrice, Bologna, 1997.
(in Italian).

S. Bittanti, M. Campi.
Raccolta di Problemi di Identificazione, Filtraggio, Controllo Predittivo.
Pitagora Editrice, Bologna, 1996.
(in Italian).

Motivating Application Examples

Prediction of Temperature of Melted Glass at the Spout of a Glass Furnace Feeder

Prediction of Temperature at the Spout of a Glass Furnace Feeder

Prediction of Melted Glass Temperature

- A feeder is the final part of a plant used for melting glass.
- Main purpose: to realize a homogeneous temperature distribution of the glass at some absolute level that allows shaping of the glass.
- Structure: it is divided into several sections in which energy can be supplied to or extracted from the glass, using burners and cooling air.

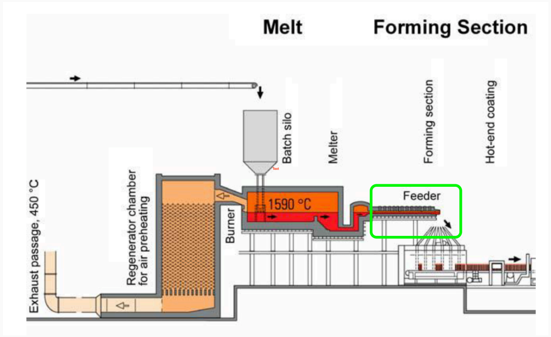


Figure 1: Typical industrial glass production plant, with in evidence the feeder [Source: https://www.glassallianceeurope.eu/]

Prediction of Temperature at the Spout of a Glass Furnace Feeder (cont.)

- The **model** shall capture and describe the **dynamic relations** between the **inputs** of the feeder (burners and cooling air in the various sections) and the **outputs** (some glass temperatures close to the outlet).

Figure 2: Detail of a glass furnace. [Source: Lehigh University, USA]

Important remarks

- The **process** is a **distributed parameters** dynamical **system**, but the model we are looking for is a **lumped parameters model**.
- The **model set** we choose for process identification (ARX model family) **does not contain** the mathematical description of the process (the so-called **true parametrization**). The present situation is one of the possible scenarios when facing system identification.

Prediction of Temperature at the Spout of a Glass Furnace Feeder (cont.)

Data description

- Available data: 1247 samples for each input and output variable;
- 3 inputs:
  - input 1: gas input of the first burner section of the feeder;
  - input 2: cooling air input into the feeder;
  - input 3: gas input of the second burner section of the feeder;
- 6 outputs: temperatures of the glass in a cross section of the feeder, close to the outlet (actually, temperature variations around the steady state temperature at each considered location).
- Data source: DaISy - Database for the Identification of Systems, KU Leuven, BE.
- Please, note: the authors have preprocessed the data: they removed the trend and the average value both from input and output data and finally, they scaled all the signals by dividing each signal by its standard deviation (i.e., the squared root of the computed variance of each signal).

Prediction of Temperature at the Spout of a Glass Furnace Feeder (cont.)

Matlab live script

The **code** for loading and visualizing
the data, applying the chosen system identification algorithm, and showing
the performance of the identified model is available as a **Matlab live script**.

Steps to retrieve the live script:

- Download as a ZIP archive the whole contents of the folder named ”**L0_Glass_Temperature_Predictor**,” available in the ”**Class Materials**” file area of the MS Teams course team.
- Uncompress the archive into a preferred folder.
- Add the chosen folder and subfolders to the Matlab path.
- Open the live script using the Matlab command:

```
open('glass_temperature_predictor.mlx');
```

- Explore the live script and run it.

Prediction of Temperature at the Spout of a Glass Furnace Feeder (cont.)

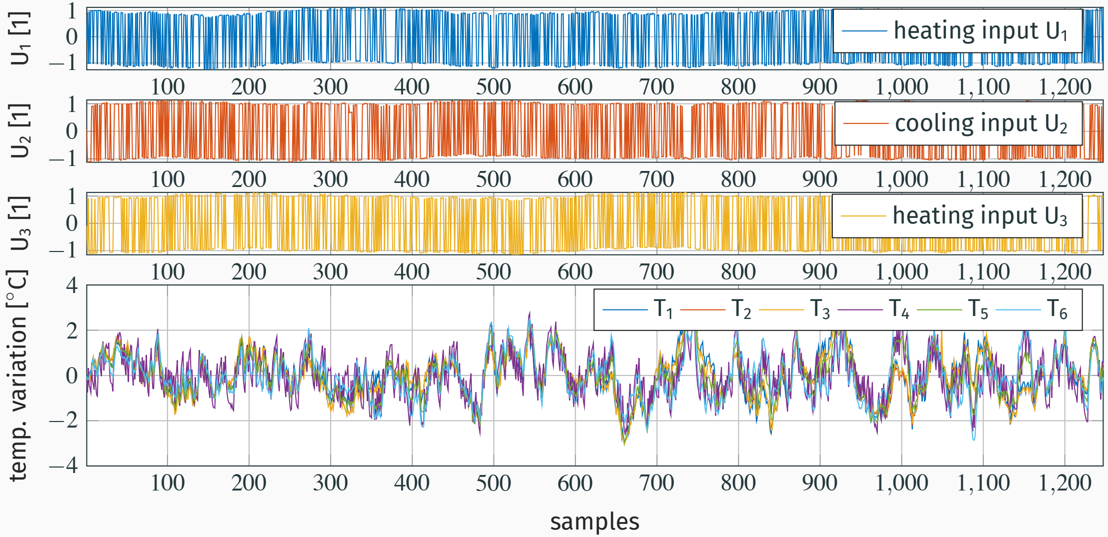


Figure 3: The *glass furnace* dataset.

Prediction of Temperature at the Spout of a Glass Furnace Feeder (cont.)

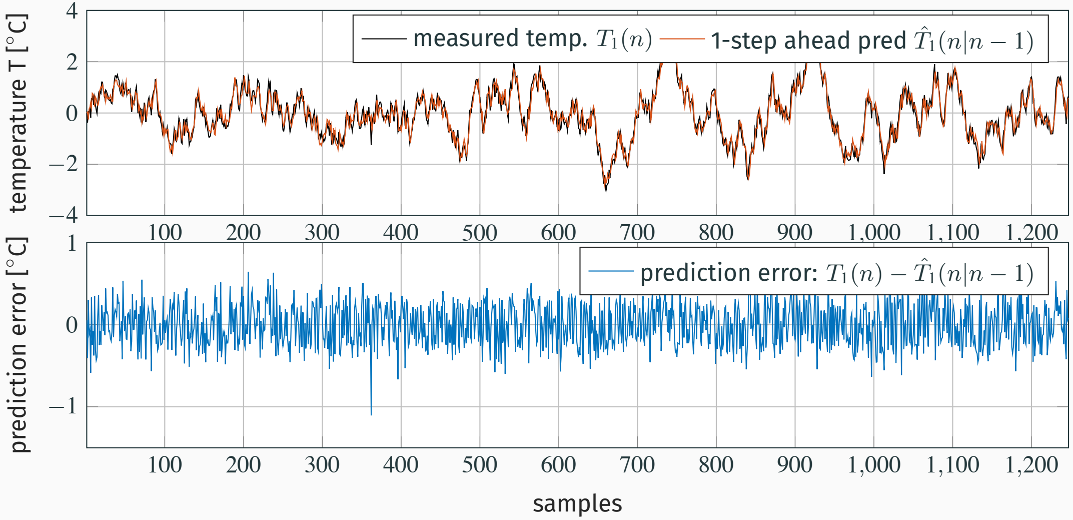


Figure 4: Prediction of the temperature T_1 using the best ARX model according the Cross-validation criterion.

Prediction of Temperature at the Spout of a Glass Furnace Feeder (cont.)

The Model for the temperature T_1

Discrete-time ARX model: $A(z)y(t) = B(z)u(t) + e(t)$

Polynomial orders: $n_a = 7$, $n_b = [7,\ 5,\ 4]$, $n_k = [1,\ 2,\ 5]$

$$A(z) = 1 - 0.6969z^{-1} - 0.05691z^{-2} - 0.06532z^{-3} - 0.02166z^{-4} - 0.1276z^{-5} - 0.0871z^{-6} + 0.1011z^{-7}$$

$$B_1(z) = 0.02755z^{-1} + 0.07005z^{-2} + 0.05956z^{-3} + 0.03304z^{-4} + 0.02933z^{-5} + 0.03725z^{-6} + 0.02344z^{-7}$$

$$B_2(z) = -0.1006z^{-2} - 0.09078z^{-3} - 0.06914z^{-4} - 0.03442z^{-5} - 0.01503z^{-6}$$

$$B_3(z) = -0.04104z^{-5} - 0.04249z^{-6} - 0.01118z^{-7} - 0.008236z^{-8}$$

Extended Kalman Filter Based Estimation of the State of Charge of a Li-ion Battery

A Kalman Filter Based Battery SOC Estimation

SOC Estimation

- A battery management system is responsible for monitoring the **state-of-charge (SOC)** of the battery, among other features.
- The SOC can be estimated using a **Kalman Filter**.

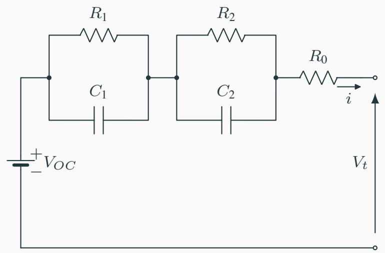


Figure 5: 2^{nd} order RC ECM battery model.

- common **battery model**: 2-RC Equivalent Circuit Model (ECM);
- parameters: the battery Open Circuit Voltage (OCV) as voltage source V_{OC} , the internal resistance R_0 and two parallel RC pairs;
- the ECM parameters depend on the actual SOC value and the battery temperature

A Kalman Filter Based Battery SOC Estimation (cont.)

Appropriate battery tests should be done to obtain data for OCV as a function of SOC for a desired range of battery temperatures.

The used battery data (namely, data of a Turnigy Graphene 5000mAh Li-ion Battery) is publicly available at https://data.mendeley.com.

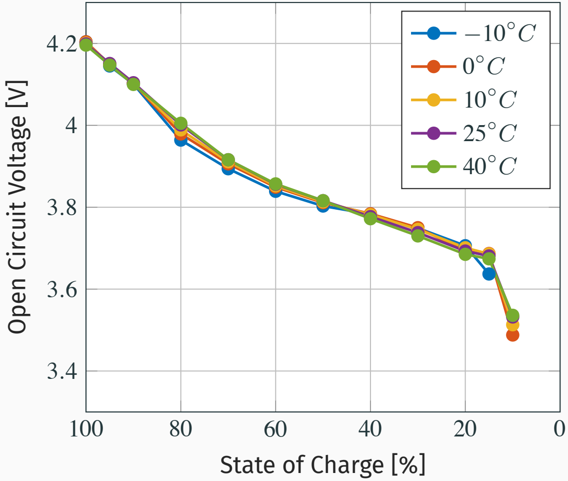


Figure 6: OCV vs. SOC of a Turnigy Graphene 5000mAh Li-ion Battery.

A Kalman Filter Based Battery SOC Estimation (cont.)

The tests data are used also to estimate the EMC parameters R_0, R_1, R_2, C_1 and C_2 at different temperatures as functions of SOC.

A model-based parameter optimization approach has to be applied.

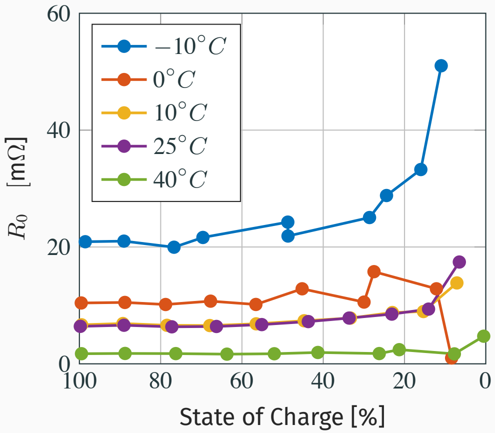


Figure 7: Resistance R_0 vs. SOC of a Turnigy Graphene 5000mAh Li-ion Battery. The battery data is publicly available at https://data.mendeley.com.

A Kalman Filter Based Battery SOC Estimation (cont.)

Battery as dynamic system: the ECM dynamics is described by

$$
\begin{cases}
\mathrm{SOC}(k+1) &= \mathrm{SOC}(k) - \dfrac{\eta\,\Delta\, i(k)}{C_n} \\[2ex]
V_1(k+1) &= e^{\frac{-\Delta}{R_1 C_1}} V_1(k) + R_1 \left(1 - e^{\frac{-\Delta}{R_1 C_1}}\right) i(k) \\[2ex]
V_2(k+1) &= e^{\frac{-\Delta}{R_2 C_2}} V_2(k) + R_2 \left(1 - e^{\frac{-\Delta}{R_2 C_2}}\right) i(k) \\[2ex]
V_{\mathrm{OC}}(k) &= g\Big(\mathrm{SOC}(k)\,,\ T(k)\Big) \\
V_t(k) &= V_{\mathrm{OC}}(k) - V_1(k) - V_2(k) - R_0\, i(k)
\end{cases}
$$

where Δ is the sampling time interval, k indicates the k-th time instant, η is the Coulombic efficiency, C_n is the battery nominal capacity, i is the battery current (negative when charging, positive when discharging), T is the battery temperature.

The equations are in **state space form** and describe a **non-linear system**: not only V_{OC} depends on the actual SOC and T values, but also the ECM parameters R_0, R_1 etc.

A Kalman Filter Based Battery SOC Estimation (cont.)

An Extended Kalman Filter as SOC Estimator

Assuming that both the state transition and the observation equation sets are affected by additive zero mean multivariate Gaussian noise

$$\begin{cases} x(k+1) &= f\big(x(k), u(k)\big) + w(k) \\ z(k) &= h\big(x(k), u(k)\big) + \nu(k) \end{cases}$$

where $x(k) = \big[\mathsf{SOC}(k)\,,\ V_1(k)\,,\ V_2(k)\big]^T,\ u(k) = i(k)\,,\ z(k) = V_t(k)$

an **Extended Kalman Filter** may be successfully applied to estimate the SOC value at the k-th time instant, using the measurements of input and output at the same time instant.

A Kalman Filter Based Battery SOC Estimation (cont.)

An Extended Kalman Filter as SOC Estimator

The EKF structure

$$
\begin{aligned}
e(k) &= \Big[z(k) - h\Big(f\left(\hat{x}\left(k-1|k-1\right)\,,\; u(k)\right)\,,\; u(k)\Big)\Big]\\
\hat{x}\left(k|k\right) &= f\left(\hat{x}\left(k-1|k-1\right)\,,\; u(k)\right) + \mathbf{K}_0(k)e(k)\\
\mathbf{K}_0(k) &= \mathbf{P}(k)\,\mathbf{H}^T(k)\Big[\mathbf{H}(k)\,\mathbf{P}(k)\,\mathbf{H}^T(k) + \mathbf{R}\Big]^{-1}\\
\mathbf{P}(k+1) &= \mathbf{F}\left\{\mathbf{P}(k) - \mathbf{K}_0(k)\,\mathbf{H}(k)\,\mathbf{P}(k)\right\}\mathbf{F}^T(k) + \mathbf{Q}
\end{aligned}
$$

where

$$
\begin{aligned}
\mathbf{F}(k) &= \frac{\partial f}{\partial x}\Big|_{\hat{x}(k-1|k-1)\,,\,u(k)} \qquad \mathbf{Q} = \mathbf{cov}(w)\\
\mathbf{H}(k) &= \frac{\partial h}{\partial x}\Big|_{\hat{x}(k-1|k-1)\,,\,u(k)} \qquad \mathbf{R} = \mathbf{cov}(\nu)
\end{aligned}
$$

A Kalman Filter Based Battery SOC Estimation (cont.)

Data description

- Reference paper:

  F. Khanum, E. Louback, F. Duperly, C. Jenkins, P. J. Kollmeyer and A. Emadi,
  "A Kalman Filter Based Battery State of Charge Estimation MATLAB Function,"
  2021 IEEE Transportation Electrification Conference & Expo (ITEC), Chicago, IL, USA, 2021, pp. 484-489,
  doi: 10.1109/ITEC51675.2021.9490163
- The code is publicly available at https://it.mathworks.com/matlabcentral/fileexchange/.
- The battery data (a Turnigy Graphene 5000mAh Li-ion Battery) is publicly available at https://data.mendeley.com.
- Sampling frequency: $1.00\,\mathrm{Hz}$

A Kalman Filter Based Battery SOC Estimation (cont.)

Matlab live script

The **code** for loading and visualizing the data, applying the EKF-based estimation algorithm, and showing the results is available as a set of **live scripts** and **live functions**.

Steps to retrieve the live script:

- Download as a ZIP archive the whole contents of the folder named "**L0_EKB_based_Battery_SOC_Estimator**," available in the "**Class Materials**" file area of the MS Teams course team.
- Uncompress the archive into a preferred folder.
- To start using the MATLAB live script, please open the previously chosen folder and select the subfolder named "**EKF_SOC_Estimation_v3.1**" as the working folder directly in MATLAB. Finally, open the live script, using for example the MATLAB command:

```
open('main.mlx');
```

- Explore the live script and run it.

A Kalman Filter Based Battery SOC Estimation (cont.)

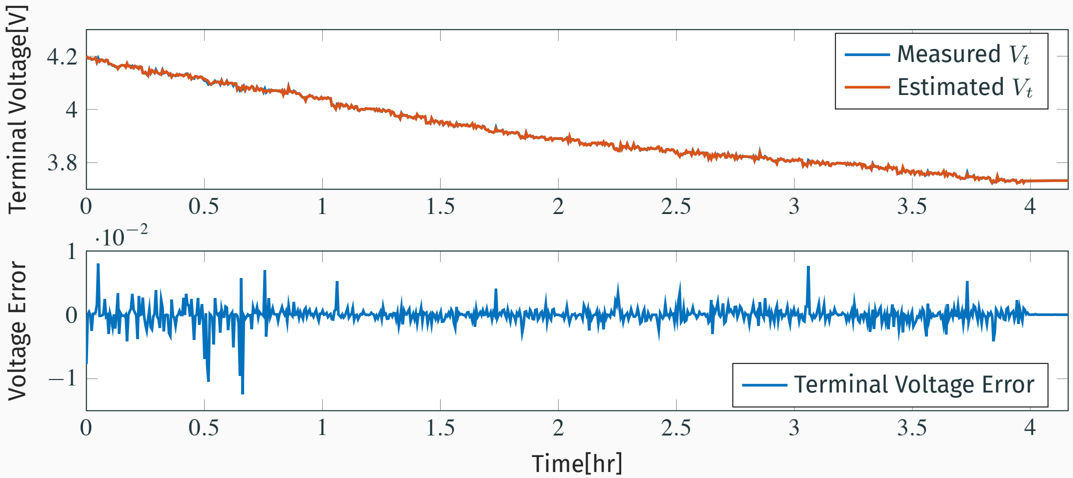


Figure 8: Measured vs. estimated terminal voltage V_t at $0\,^{\circ}$C (upper plot); voltage error (lower plot). Battery SOC estimation using EKF filter, applied to a Turnigy Graphene 5000mAh Li-ion battery.

A Kalman Filter Based Battery SOC Estimation (cont.)

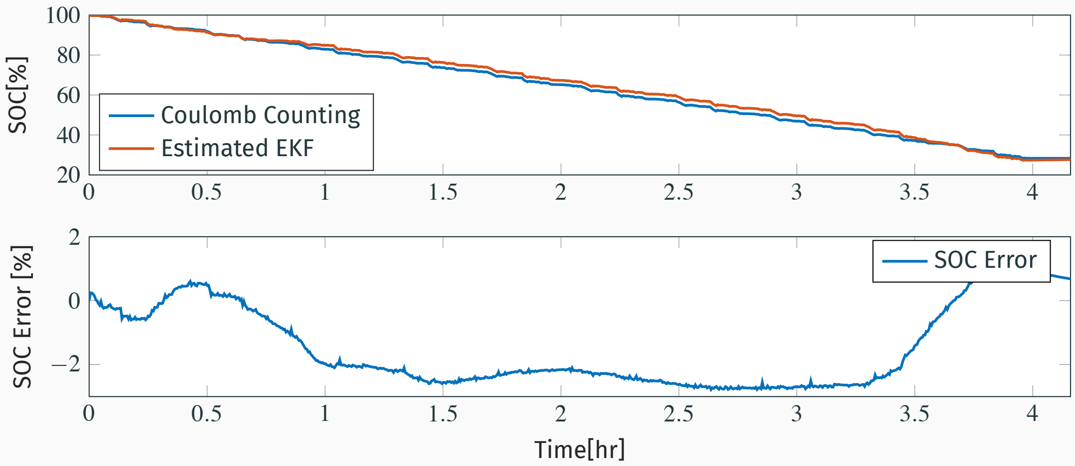
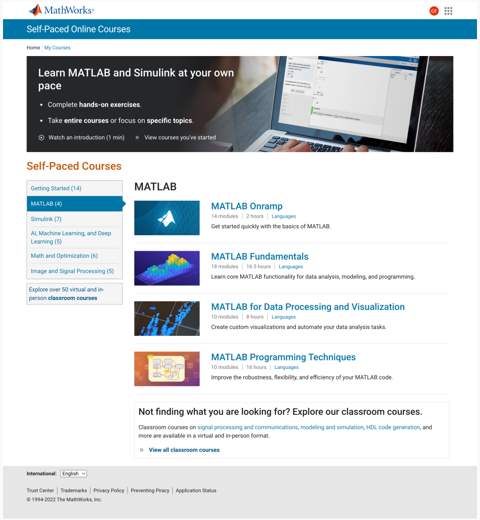


Figure 9: Coulomb Counting vs. SOC Estimated at $0\,^\circ$C (upper plot); SOC estimation error using EKF filter (lower plot). Battery SOC estimation using EKF filter, applied to a Turnigy Graphene 5000mAh Li-ion battery.

Online Tutorials and Courses on MATLAB

Online courses and tutorials for learning MATLAB fundamentals and programming techniques with online courses:
MATLAB Academy



MathWorks

Self-Paced Online Courses

Home | My Courses

Learn MATLAB and Simulink at your own pace

- Complete **hands-on exercises**.
- Take **entire courses** or focus on **specific topics**.

Watch an introduction (1 min) » View courses you've started

Self-Paced Courses

- Getting Started (14)
- MATLAB (4)
- Simulink (7)
- AI, Machine Learning, and Deep Learning (5)
- Math and Optimization (6)
- Image and Signal Processing (5)

Explore over 50 virtual and in-person **classroom courses**

MATLAB

MATLAB Onramp
14 modules | 2 hours | Languages
Get started quickly with the basics of MATLAB.

MATLAB Fundamentals
18 modules | 16.5 hours | Languages
Learn core MATLAB functionality for data analysis, modeling, and programming.

MATLAB for Data Processing and Visualization
10 modules | 8 hours | Languages
Create custom visualizations and automate your data analysis tasks.

MATLAB Programming Techniques
10 modules | 16 hours | Languages
Improve the robustness, flexibility, and efficiency of your MATLAB code.

Not finding what you are looking for? Explore our classroom courses.

Classroom courses on signal processing and communications, modeling and simulation, HDL code generation, and more are available in a virtual and in-person format.

» **View all classroom courses**

International: English

Trust Center | Trademarks | Privacy Policy | Preventing Piracy | Application Status
© 1994-2022 The MathWorks, Inc.

267MI –Fall 2023

Course Overview

END