BATTERY STATE-OF-CHARGE ESTIMATION

MATLAB DESIGN TOOL BASED ON DATA DRIVEN LEAST-SQUARES SUPPORT VECTOR MACHINE METHOD

Luca Cavanini, Pawel Majecki, Mike Grimble, Gerrit van der Molen

L. Cavanini, P. Majecki, M. J. Grimble and G. M. van der Molen are with
Industrial Systems and Control Ltd (www.isc-ltd.com), Glasgow.
M. J. Grimble is also with the University of Strathclyde, Glasgow

Acknowledgements: Research supported by the Scottish Enterprise Can Do Innovation Challenge programme for Zero Emissions Heavy Duty Vehicles

Scottish Enterprise

Contents

Introduction

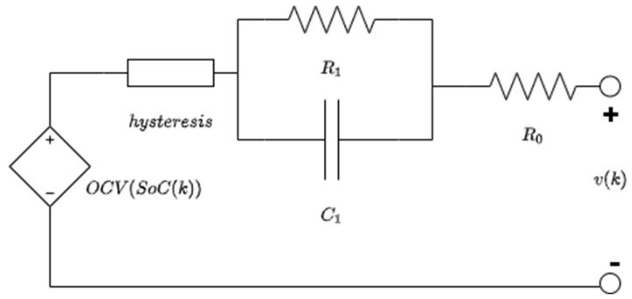
- Hybrid HDVs with electric power zero-emissions
  - Involve battery energy storage supplying electric vehicle
  - In conjunction with other power sources, e.g. Fuel Cells
  - Green revolution highlights importance of batteries and Battery Management Systems (BMS) in transport applications
- Development of estimators is critical aspect of BMS design
  - Algorithms compute estimates of battery characteristics that are unmeasurable
  - E.g. for battery equivalent capacity, battery resistance & battery State-of-Charge (SoC)
- BMS estimator design is complex and time-consuming task
  - Due to variety of battery technologies, architectures, and uncertainties
  - Several data-driven and combined data-driven/model-based methods have been proposed to solve BMS SoC estimation problem
  - Including Neural Network (NN), Fuzzy Logic, KF-based algorithms, combining ML and model-based approaches

Philosophy

Industrial Systems and Control Ltd.

- Purely data-driven design methodology to perform battery estimation
  - Focus is on battery State-of-Charge (SoC) estimation problem
  - Method combines modern data-driven Least-Squares Support Vector Machine (LS-SVM) with data pruning procedure for efficient computations
  - Method uses Particle Swarm Optimization (PSO) to tune algorithm
- Pruning method selects important samples from training dataset
  - Reduces computational complexity and memory footprint of algorithm
  - Whilst maintaining performance
- Dataset collected by emulating battery using Enhanced Self-Correcting (ESC) battery simulation model
  - Baseline estimator was also developed for comparison with model-based approach
    - Extended Kalman Filter (EKF)
  - Estimators compared in simulation
    - In terms of performance and computational complexity

Battery Model



Battery State-of-Charge

SoC of *i*-th battery cell is defined as:

$$SoC_i(k) = \frac{\theta(k) - \theta_{0\%}}{\theta_{100\%} - \theta_{0\%}}$$

where $\theta(k)$ is average lithium concentration stoichiometry at discrete-time k defined as:

$$\theta(k) = \frac{c_{s,avg,k}}{c_{s,max}}$$

This should remain between *0%* and *100%*

- Although possible to violate limits in over-discharge or over-charge situation
- Presently no way to measure concentrations to calculate stoichiometries and SoC
- Therefore necessary to infer or estimate SoC using only measurements of cell terminal voltage, current, and temperature

Data-Driven SoC Estimator

Pruning Procedure

One of main issues in LS-SVM identification is size of training dataset

Estimate is computed iteratively by comparing training dataset information with measurements from real system

Two issues:

1) Computational burden increases with dataset size; when training dataset is large, iterative computation of estimate can be prohibitive in real-time;
2) To use LS-SVM on real systems, training data must be stored in memory, reducing possibility of porting algorithm onto hardware with limited resources

Possible solution for reducing training dataset size is so-called *pruning method*

Method involves iteratively performing LS-SVM identification, reducing training dataset size at each iteration

By gradually omitting least significant training data

- Lowest Lagrangian multipliers

Method allows a priori definition of maximum size of data subset to consider

- Or equivalently, acceptable value of identification performance degradation

Data-Driven SoC Estimator

Industrial Systems and Control Ltd.

Pruning Procedure

1) Considering original dataset of size N & train LS-SVM
2) Remove number of points (e.g. $\Delta N = 5\%$ of N) corresponding to smallest values in $|\alpha_k|$ (data point 'influence') spectrum
3) Train LS-SVM with new reduced dataset
4) Return to point 2 until identification performance degradation threshold is exceeded

Data-driven SoC Estimator

Data-Driven Design Procedure

Data-driven estimator design is based on ML and statistical methods

To limit computational complexity, following design procedure was defined:

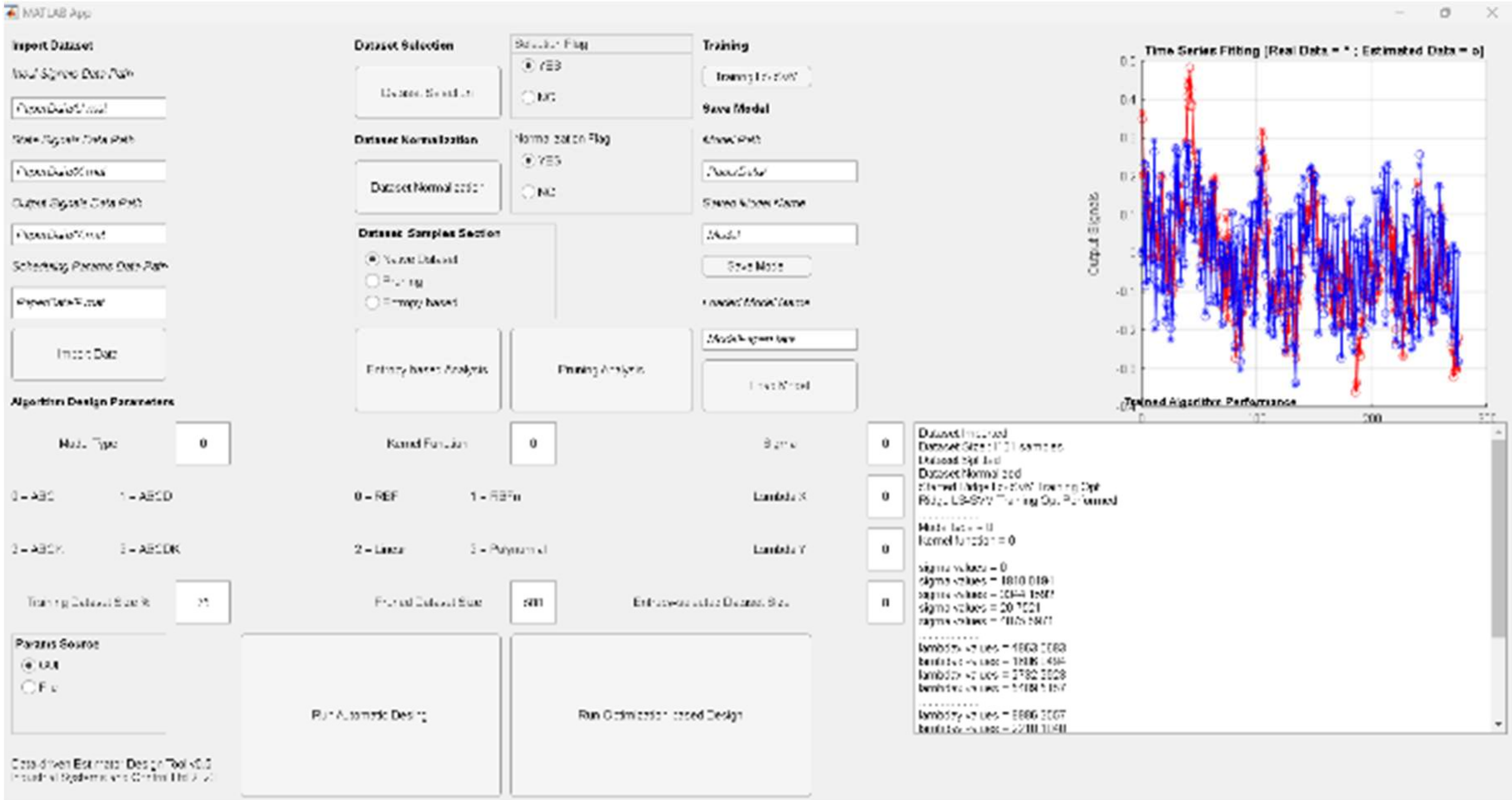
- Phase 1: Perform first LS-SVM training based on PSO & original training dataset to compute estimator calibration parameters
- Phase 2: Perform Pruning Analysis with calibration parameters of Phase 1 & select pruned dataset size to consider for final estimator training
- Phase 3: Perform final LS-SVM training

Data-driven SoC Estimator Design Tool

Industrial Systems and Control Ltd.

Design approach is implemented in Data-driven Estimator Design Tool

- Developed by ISC to simplify development
- Software generates data-driven estimator
  - Includes tuning parameters for training and pruning procedure
- GUI provides feedback to user
  - Estimation/pruning graphical results & identification of performance statistics
- GUI shows inclusion of set of common data manipulation methods e.g. data normalization & Entropy-based Analysis



Performance Criteria

Performance indices to compare estimators:

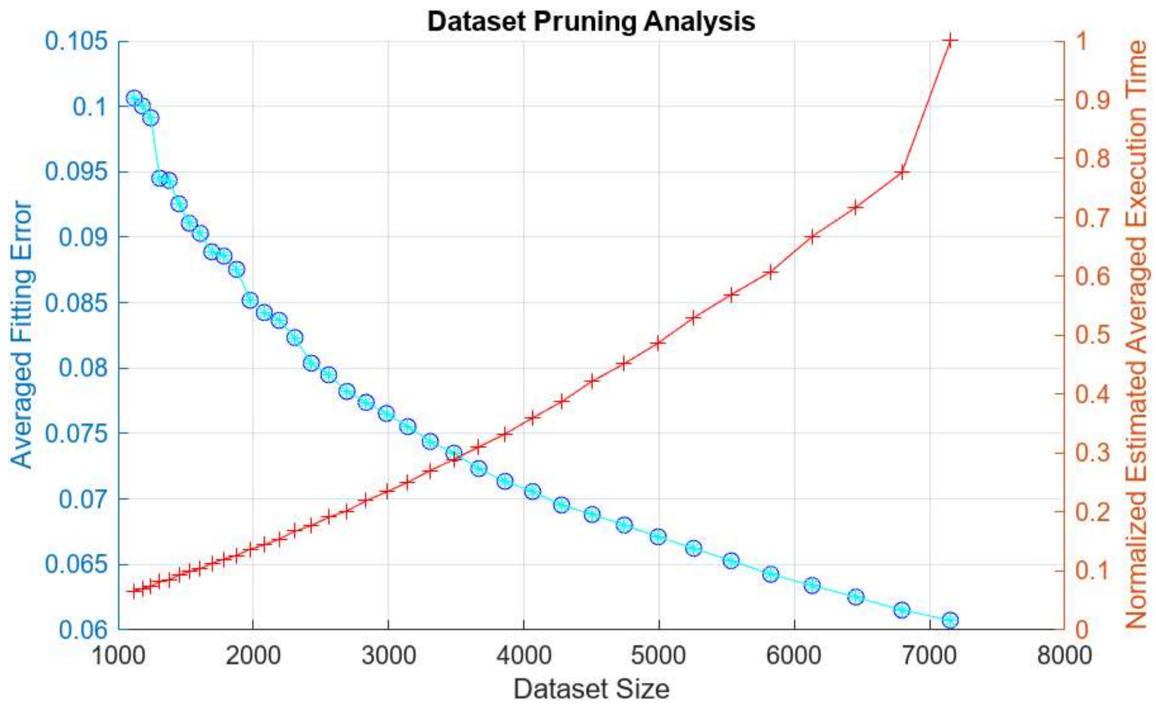
- **Mean Absolute Error** (MAE)
- **Root Mean Square Error** (RMSE)
- **Execution Time** (ET) in seconds

Estimators Comparison

Data-driven estimator design: Phase 2

Industrial Systems and Control Ltd.

- In Phase 2, pruning was applied to dataset from Phase 1
- Figure: Result of analysis with averaged fitting error with respect to size of selected dataset, plus normalised averaged execution time
- Analysis enables evaluation of trade-off between fitting error increase and related execution time reduction
- From this, size of pruned dataset for Phase 3 is selected as 5000 samples
- Reduces computational burden by 50%, with expected 12% increase of fitting error

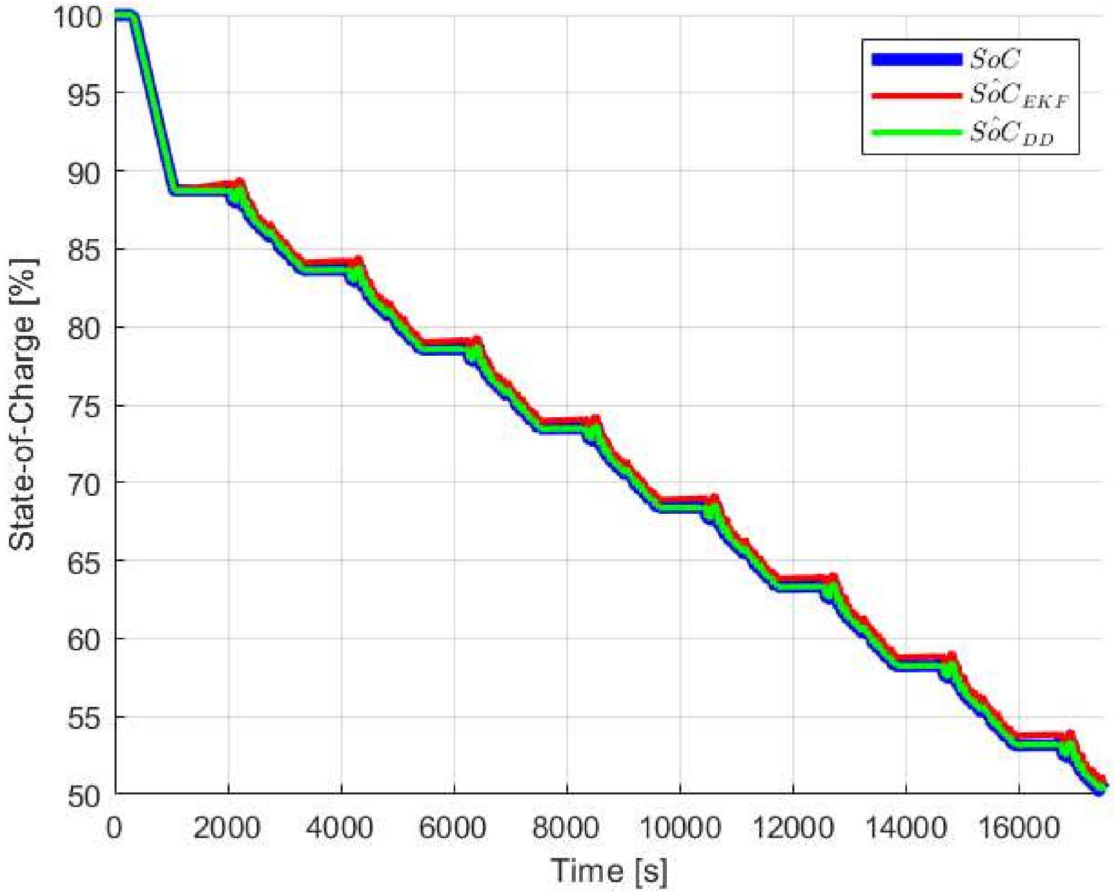


Estimators Comparison

Industrial Systems and Control Ltd.

Performance of data-driven estimator vs. baseline EKF

Figure: estimator performance for scenario with SoC varying from 100% to 50%



Estimator	RMSE	MAE
EKF	5.1049×10^{-3}	4.8496×10^{-3}
LS-SVM	4.7922×10^{-3}	2.9855×10^{-3}

Conclusions

Industrial Systems and Control Ltd.

- Battery State-of-Charge (SoC) estimation method developed
  - Exploiting capabilities of data-driven Machine Learning (ML) techniques
- Approach/tool can be used for more general applications
  - Combines Least-Squares Support Vector Machine (LS-SVM) identification technique with Pruning Dataset Selection procedure
  - Uses Particle Swarm Optimization (PSO) method
- **Automatic procedure combining these techniques has been developed and implemented within Data-driven Estimator Design Tool**
  - **Developed to reduce effort in battery estimation**
- Estimator approach achieves good performance with limited computational complexity
- Performance of algorithm compared against baseline EKF estimator
  - Results demonstrated that proposed estimator overcame limitations of model-based methods whilst reducing computational burden