MATLAB EXPO

Integrating AI-based Virtual Sensors into Model-Based Design

Emelie Andersson, MathWorks

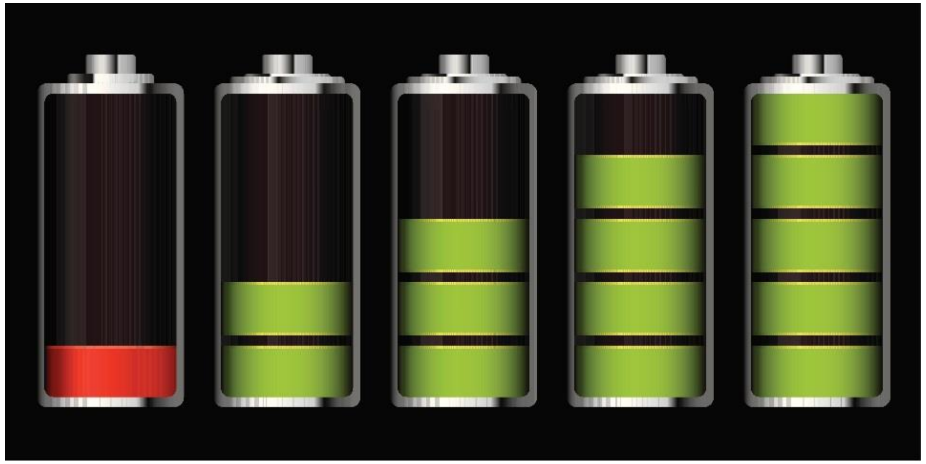
Javier Gazzarri, MathWorks

MathWorks

Why Virtual Sensors?

When estimating a quantity that is not measurable

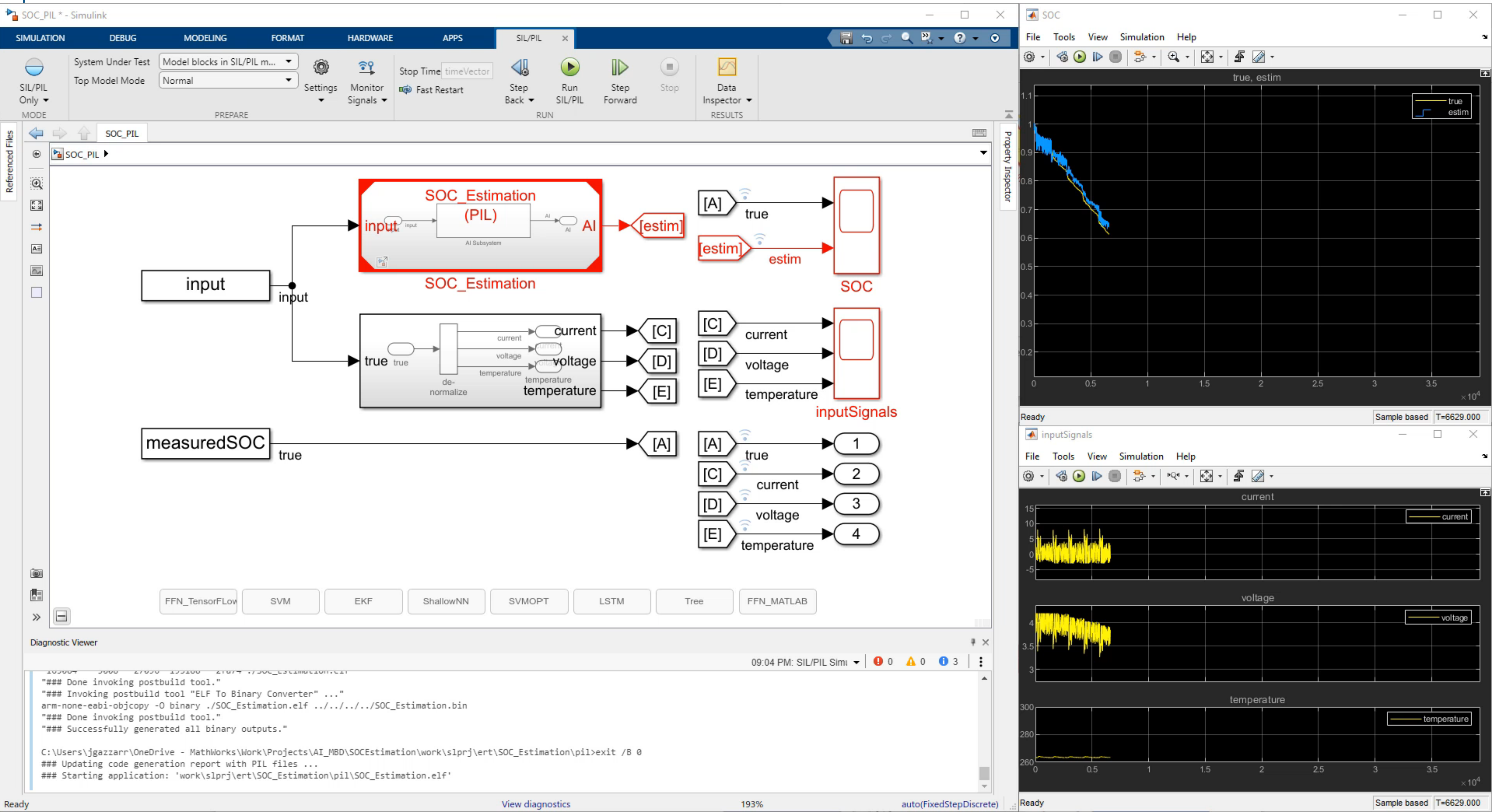
Battery State of Charge (SOC)



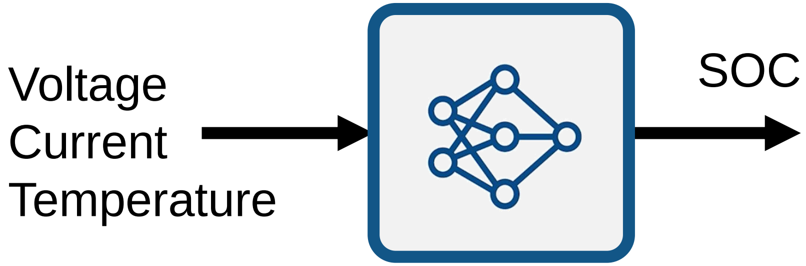
Not directly measurable

We <u>measure</u> voltage, current, temperature and <u>calculate</u> SOC

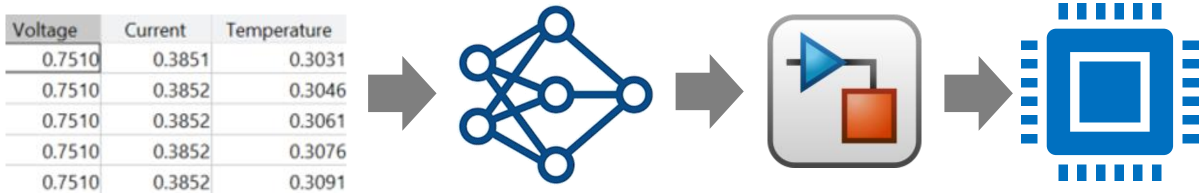
MATLAB EXPO


SOC_PIL * - Simulink
SIMULATION
DEBUG
MODELING
FORMAT
HARDWARE
APPS
SIL/PIL
SIL/PIL Only
MODE
System Under Test
Model blocks in SIL/PIL m...
Top Model Mode
Normal
Settings
Monitor Signals
PREPARE
Stop Time
timeVector
Fast Restart
Step Back
Run SIL/PIL
Step Forward
Stop
RUN
Data Inspector
RESULTS
SOC_PIL
SOC_Estimation
(PIL)
input
AI
AI Subsystem
[estim]
input
true
current
voltage
temperature
de-normalize
[C]
[D]
[E]
measuredSOC
[A]
estim
SOC
inputSignals
1
2
3
4
FFN_TensorFLow
SVM
EKF
ShallowNN
SVMOPT
LSTM
Tree
FFN_MATLAB
Diagnostic Viewer
09:04 PM: SIL/PIL Simu
"### Done invoking postbuild tool."
"### Invoking postbuild tool "ELF To Binary Converter" ..."
arm-none-eabi-objcopy -O binary ./SOC_Estimation.elf ../../../../SOC_Estimation.bin
"### Done invoking postbuild tool."
"### Successfully generated all binary outputs."
C:\Users\jgazzarr\OneDrive - MathWorks\Work\Projects\AI_MBD\SOCEstimation\work\slprj\ert\SOC_Estimation\pil>exit /B 0
Updating code generation report with PIL files ...
Starting application: 'work\slprj\ert\SOC_Estimation\pil\SOC_Estimation.elf'
Ready
View diagnostics
193%
auto(FixedStepDiscrete)
SOC
File
Tools
View
Simulation
Help
true, estim
true
estim
Sample based
T=6629.000
inputSignals
current
voltage
temperature

Agenda



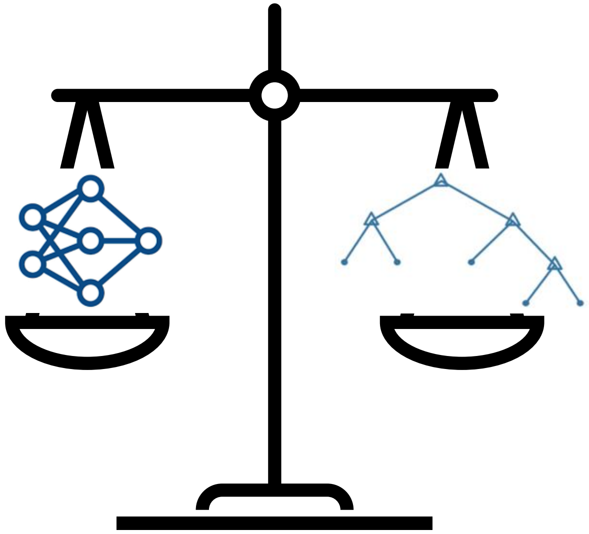
- Develop AI-based virtual sensor for battery SOC estimation



Voltage	Current	Temperature
0.7510	0.3851	0.3031
0.7510	0.3852	0.3046
0.7510	0.3852	0.3061
0.7510	0.3852	0.3076
0.7510	0.3852	0.3091

- Workflow - From data acquisition to hardware deployment

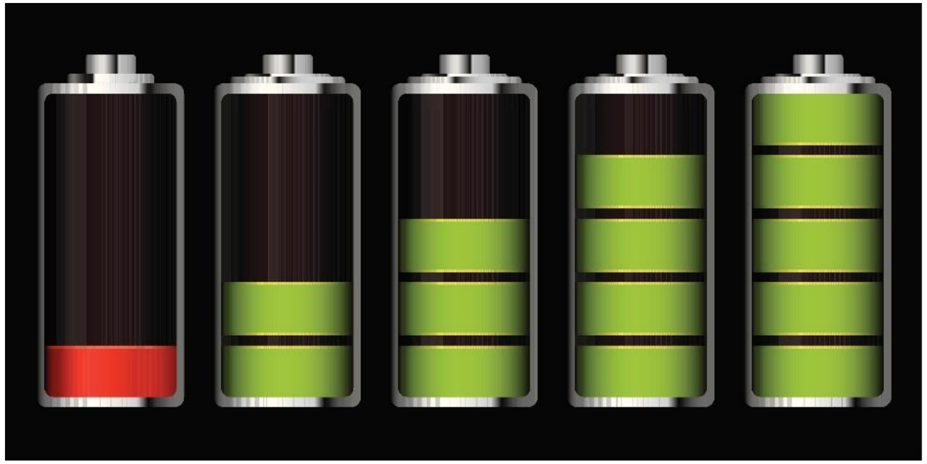
- Compare different AI methods



Battery State of Charge (SOC)

$$SOC(t) = \frac{1}{C}\int_0^t I(p)dp$$

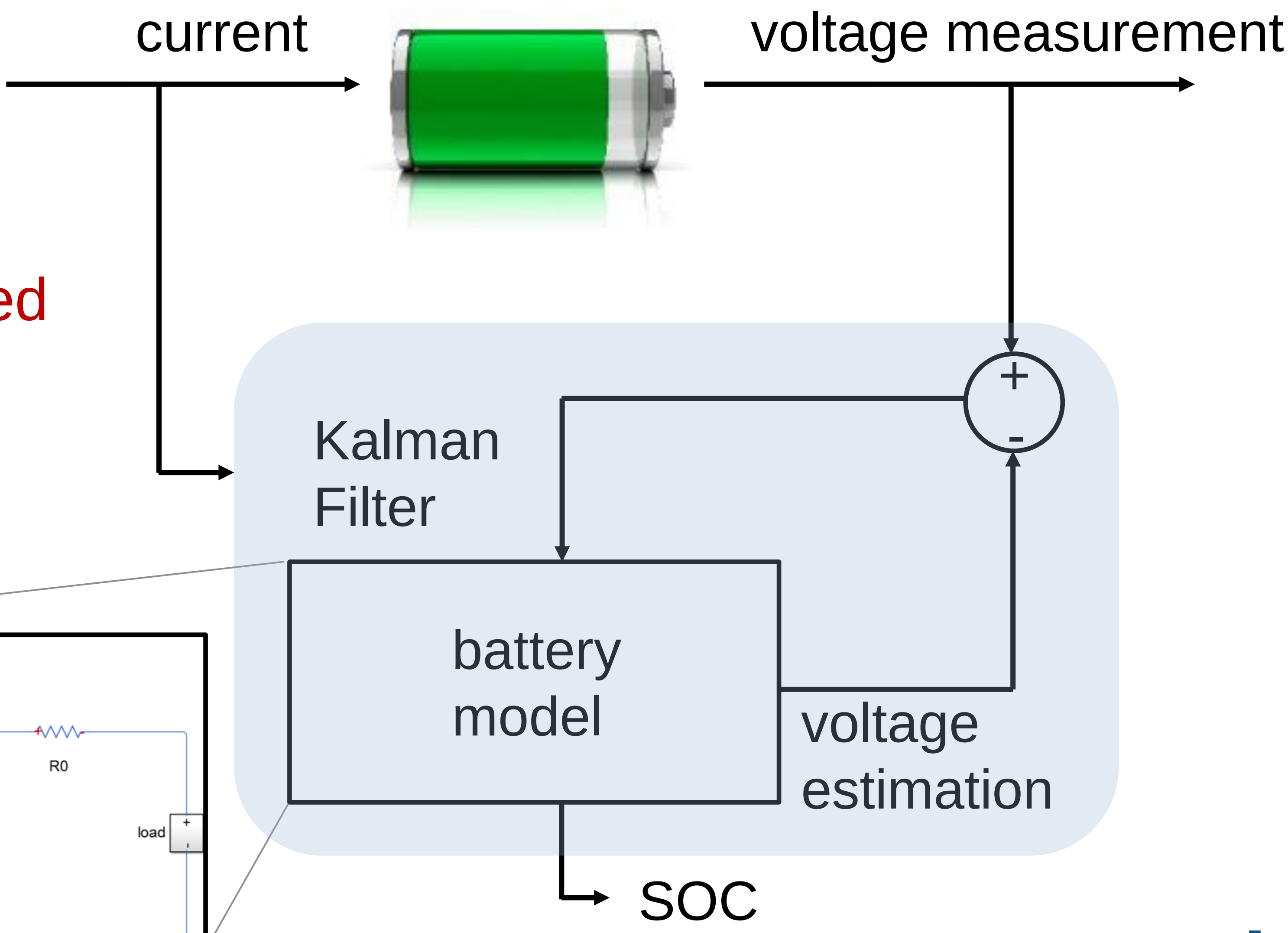
capacity current

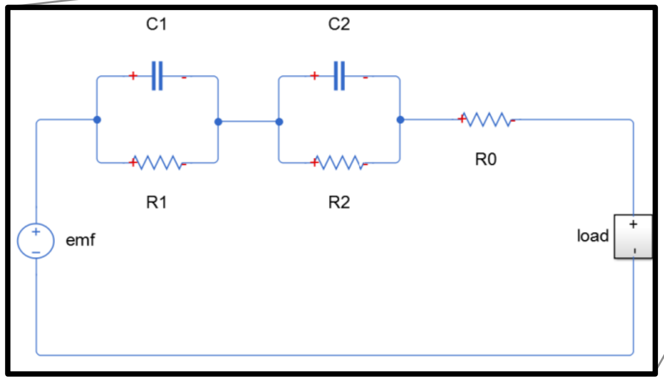


Affected by sensor error

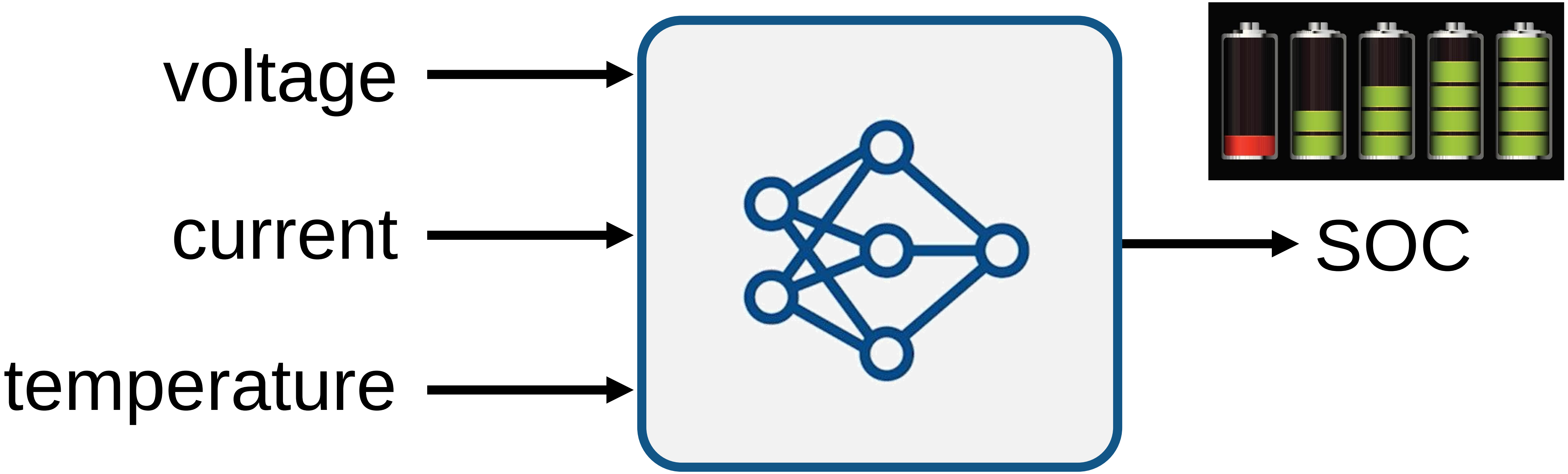
Extended Kalman Filter

- Well established
- Accurate
- Detailed battery model required
  - Operating condition range
- Computationally intensive





How About…



Instead of creating a physics-based model –
Train a Statistical Model

Comparison

Extended Kalman Filter

- Well established
- Accurate
- Detailed battery model required
  - Operating condition range
- Computationally intensive

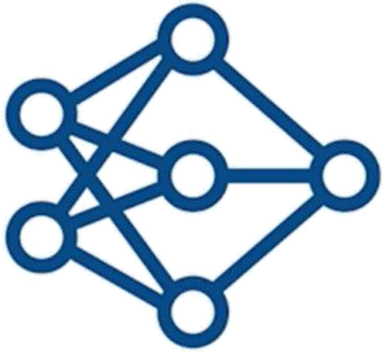
AI

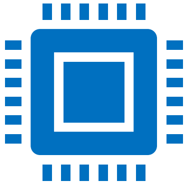
- Training on real data
- Capture very complex data relationships
- No need for battery model
- Interpretability
- Computationally intensive

AI-driven System Design

Data Preparation | **AI Modeling** | **Simulation & Test** | **Deployment**

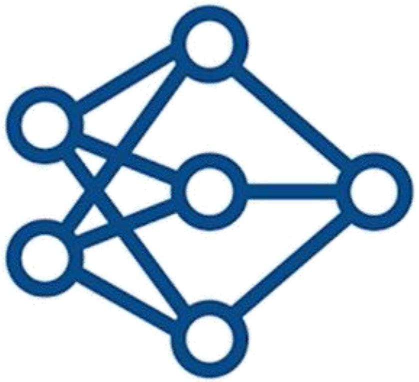
Voltage	Current	Temperature
0.7510	0.3851	0.3031
0.7510	0.3852	0.3046
0.7510	0.3852	0.3061
0.7510	0.3852	0.3076
0.7510	0.3852	0.3091

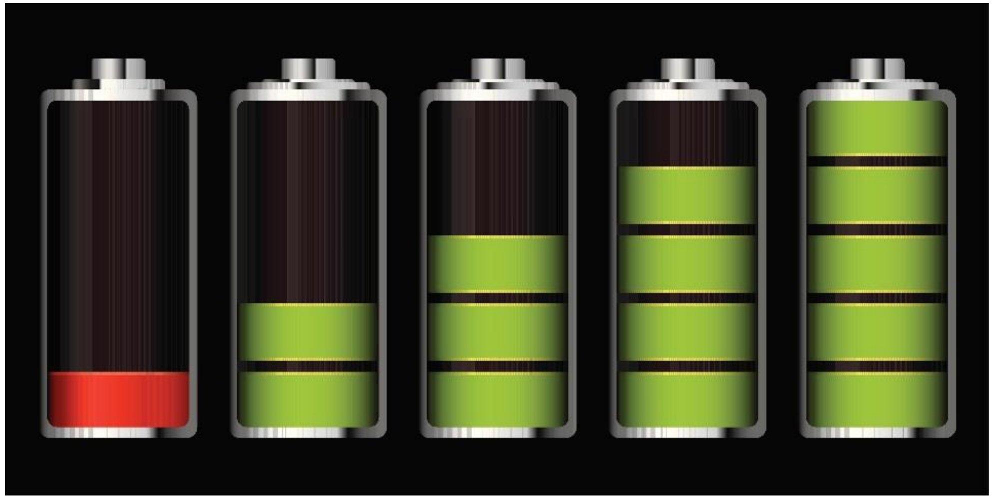




Steps involved in creating an AI-based virtual sensor

Back to SOC estimation





2020-01-1181 Published 14 Apr 2020

Robust xEV Battery State-of-Charge Estimator Design Using a Feedforward Deep Neural Network

Carlos Vidal, Phillip Kollmeyer, and Mina Naguib McMaster Automotive Res. Centre

Pawel Malysz and Oliver Gross FCA US LLC

Ali Emadi McMaster University

Citation: Vidal, C., Kollmeyer, P., Naguib, M., Malysz, P. et al., "Robust xEV Battery State-of-Charge Estimator Design Using a Feedforward Deep Neural Network," SAE Technical Paper 2020-01-1181, 2020, doi:10.4271/2020-01-1181.

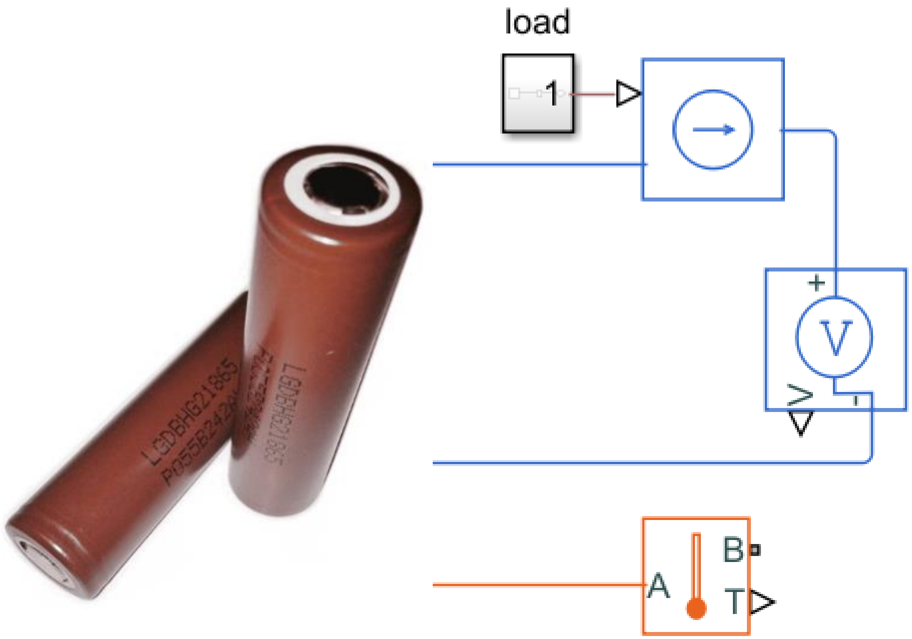
Abstract

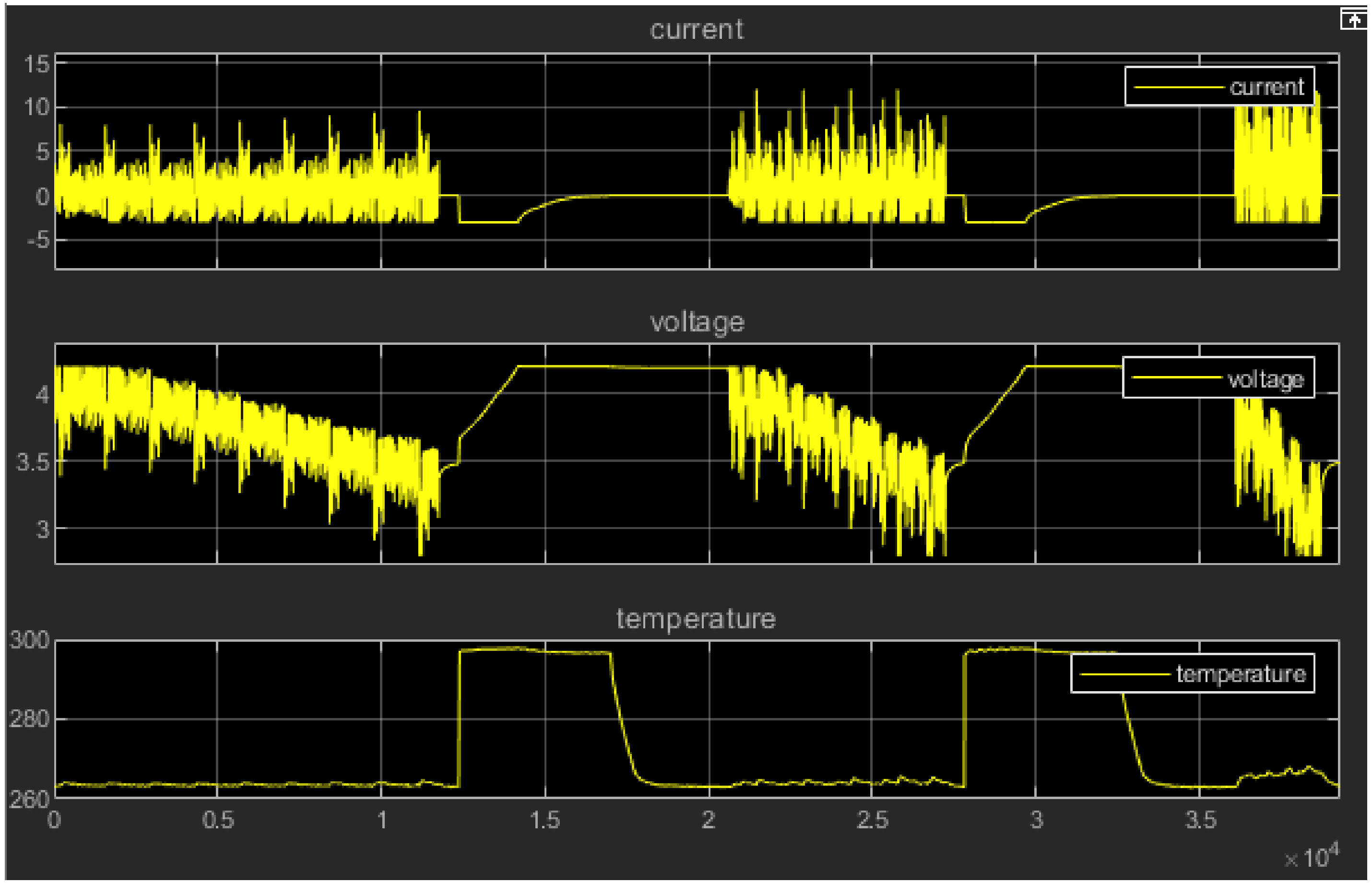
Battery state-of-charge (SOC) is critical information for the vehicle energy management system and must be accurately estimated to ensure reliable and affordable electrified vehicles (xEV). However, due to the nonlinear temperature, health, and SOC dependent behaviour of Li-ion

(FNN) approach. The method includes a description of data acquisition, data preparation, development of an FNN, FNN tuning, and robust validation of the FNN to sensor noise. To develop a robust estimator, the FNN was exposed, during training, to datasets with errors intentionally added to the data, e.g. adding cell voltage variation of ±4mV, cell current

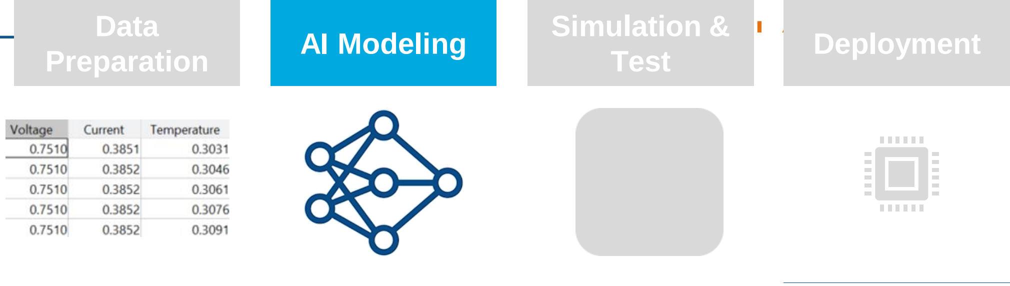
Voltage	Current	Temperature
0.7510	0.3851	0.3031
0.7510	0.3852	0.3046
0.7510	0.3852	0.3061
0.7510	0.3852	0.3076
0.7510	0.3852	0.3091

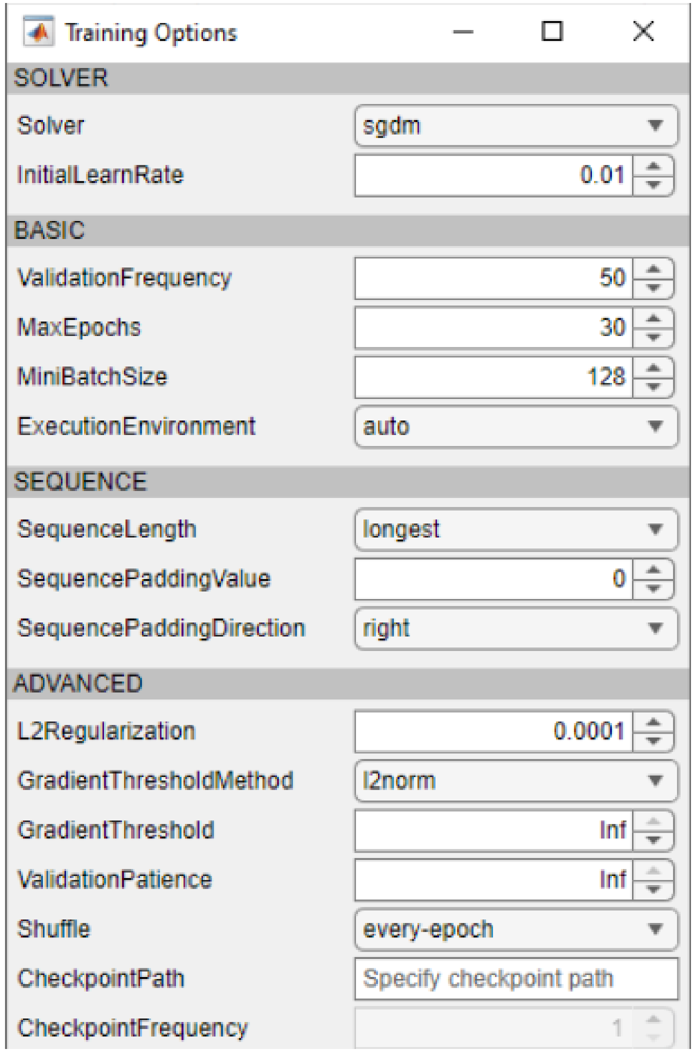
Read data

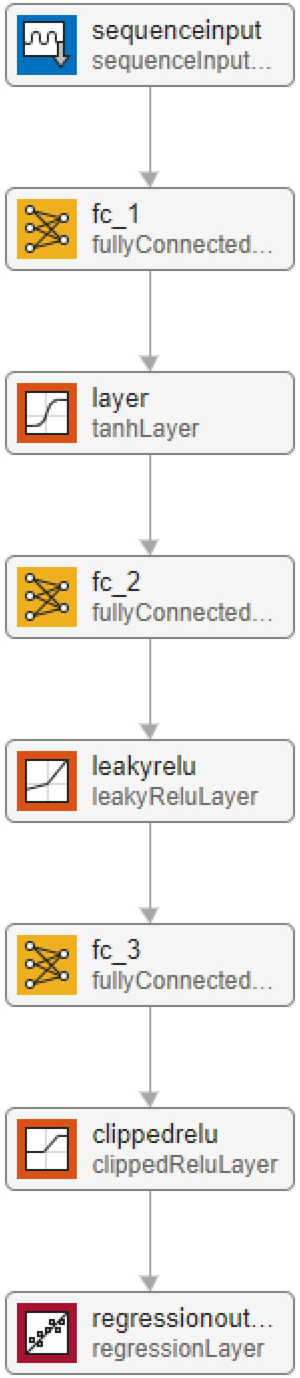


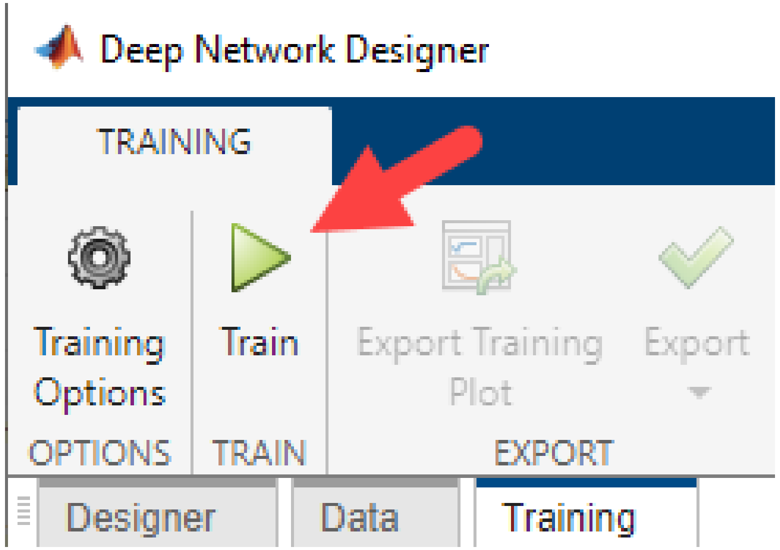


- Configure AI model
- Train and Test AI model

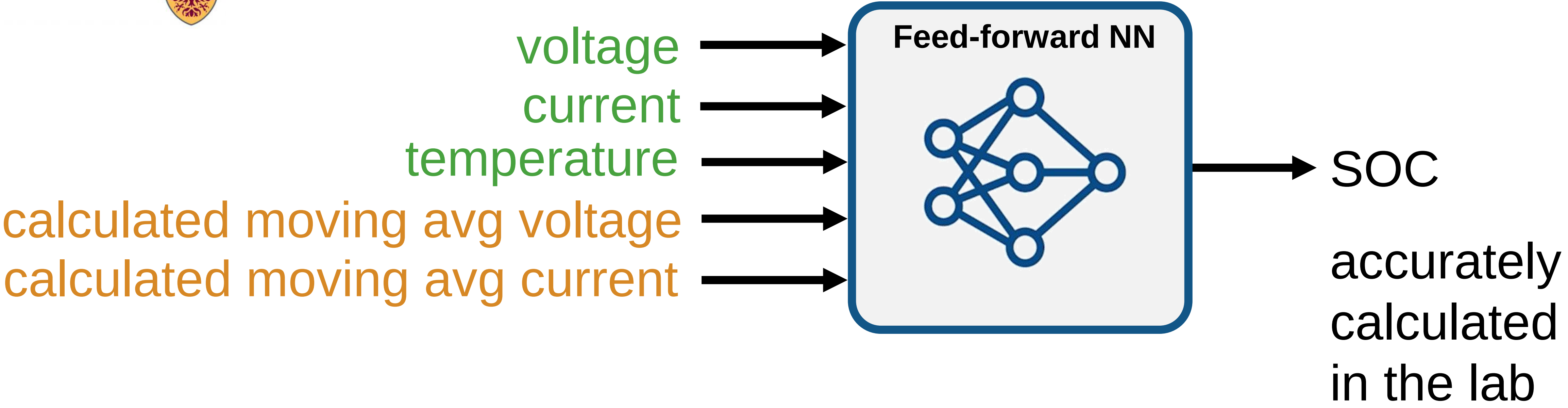








McMaster University

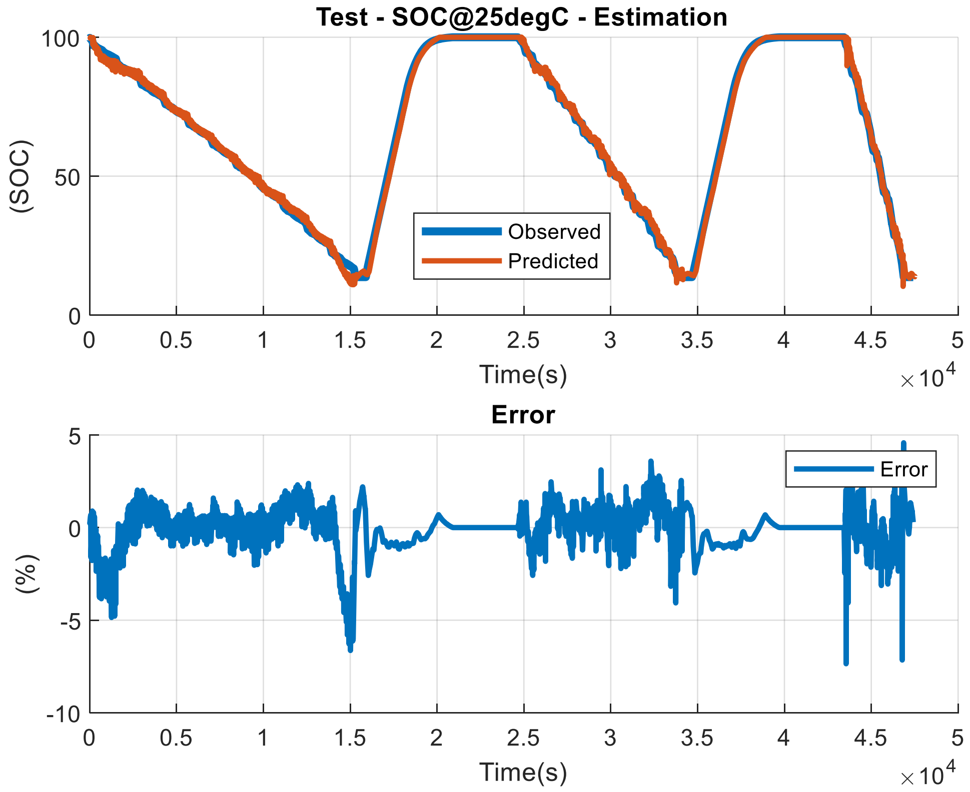


Feed Forward NN is simple – but it has no memory

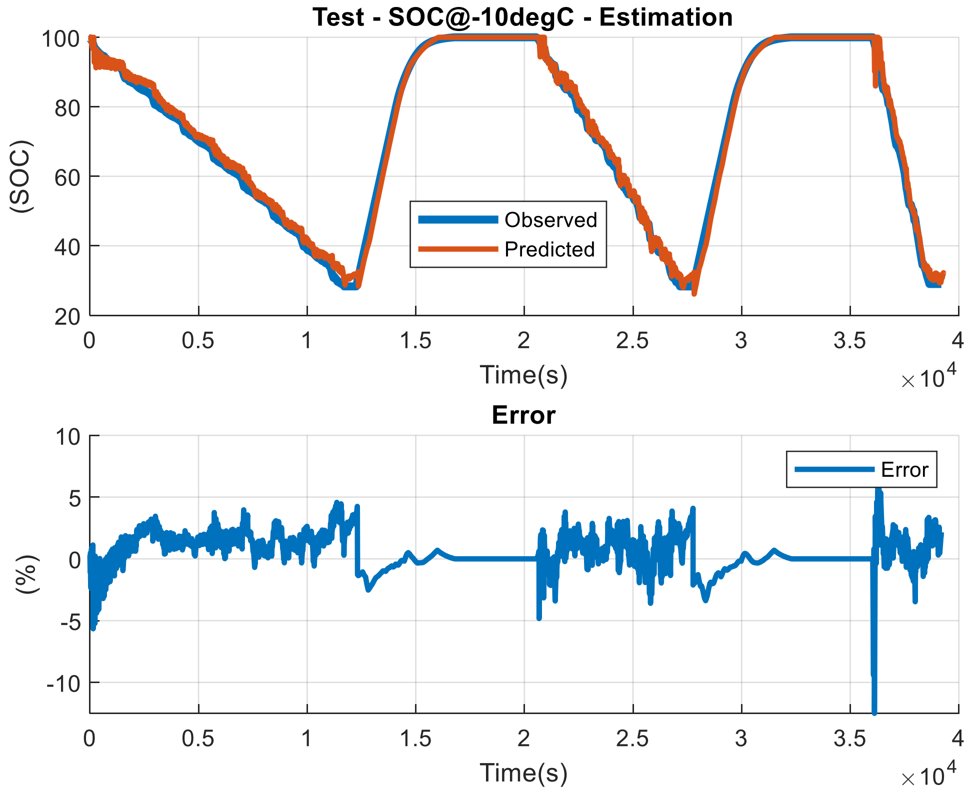
Moving average added to the input signals

Results

25ºC



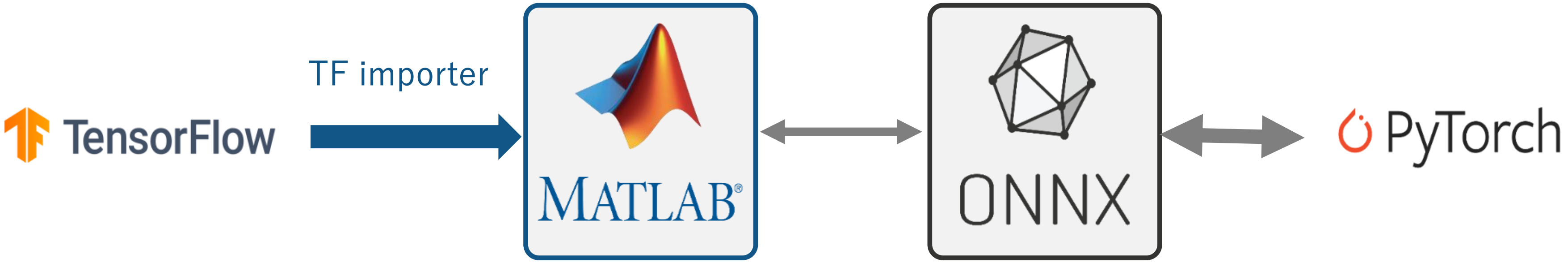
-10ºC



Prediction is good even at low temperatures

prediction
ground truth

Import Pre-Trained Model

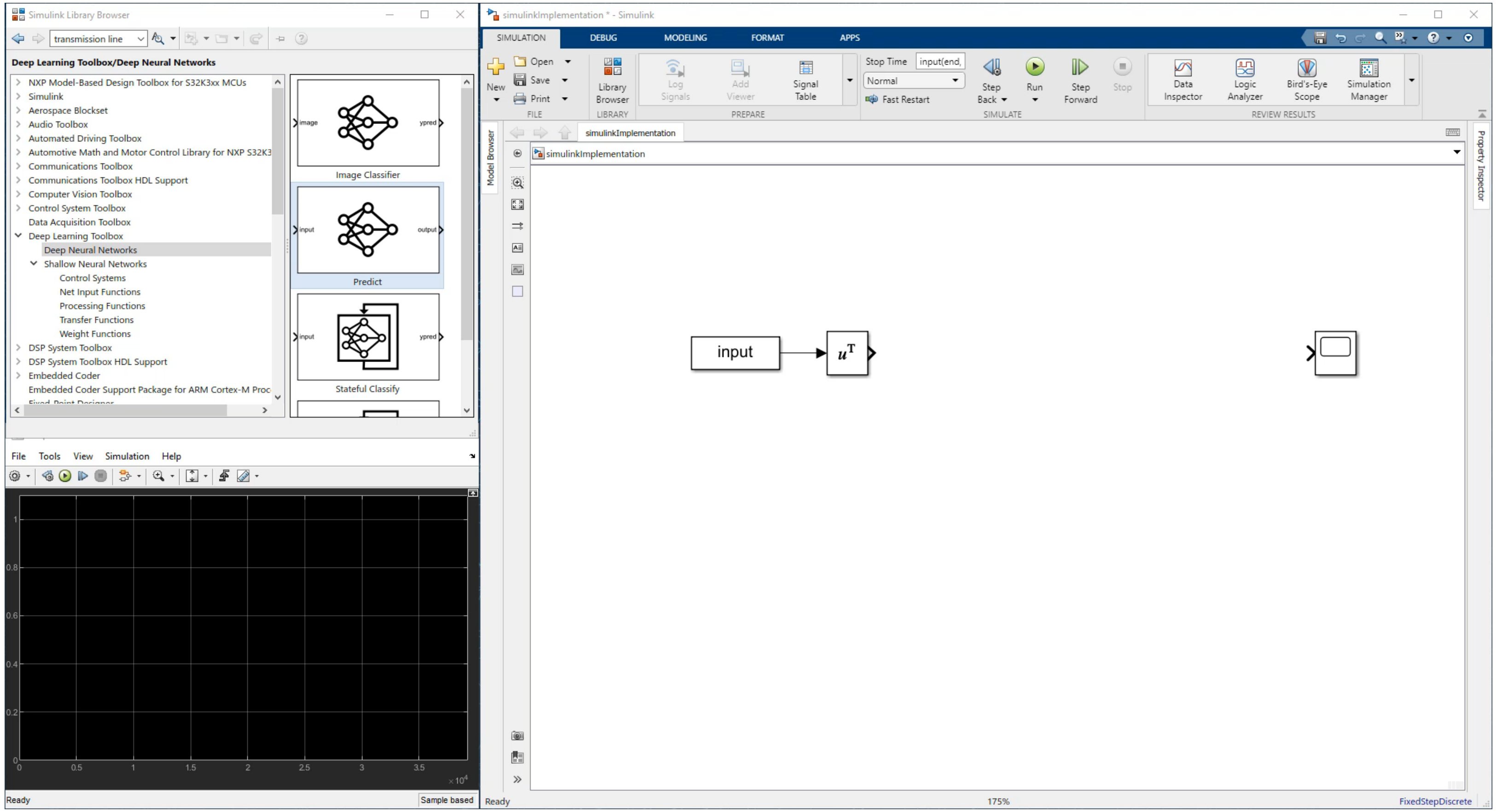


You can also import an AI model trained outside of the MathWorks ecosystem into MATLAB

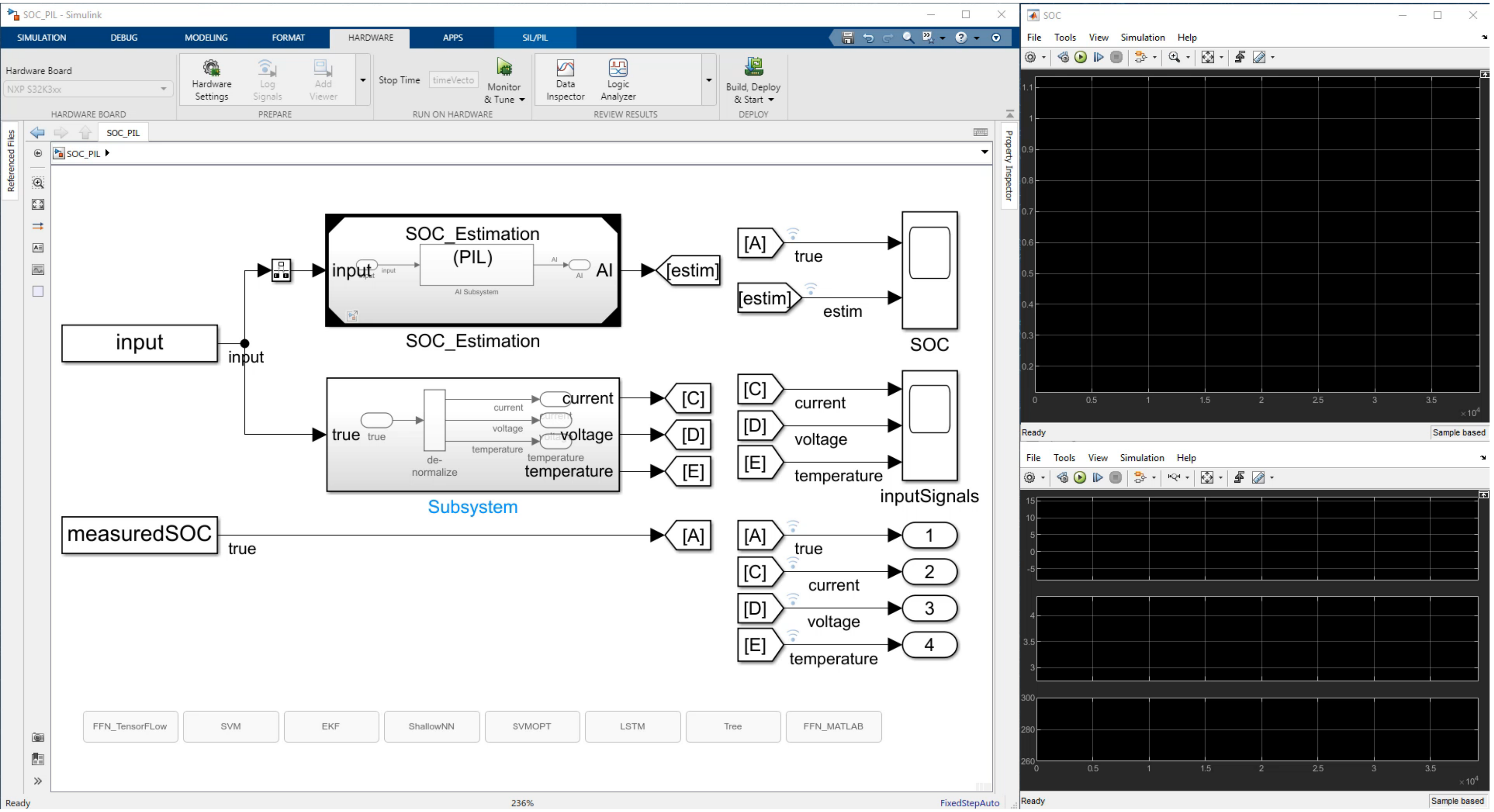
Simulink Integration

Data Preparation | AI Modeling | **Simulation & Test** | Deployment

Voltage	Current	Temperature
0.7510	0.3851	0.3031
0.7510	0.3852	0.3046
0.7510	0.3852	0.3061
0.7510	0.3852	0.3076
0.7510	0.3852	0.3091



Simulink provides blocks with different AI functions We just parameterize them with the AI function name and feed them with signals with the predictors

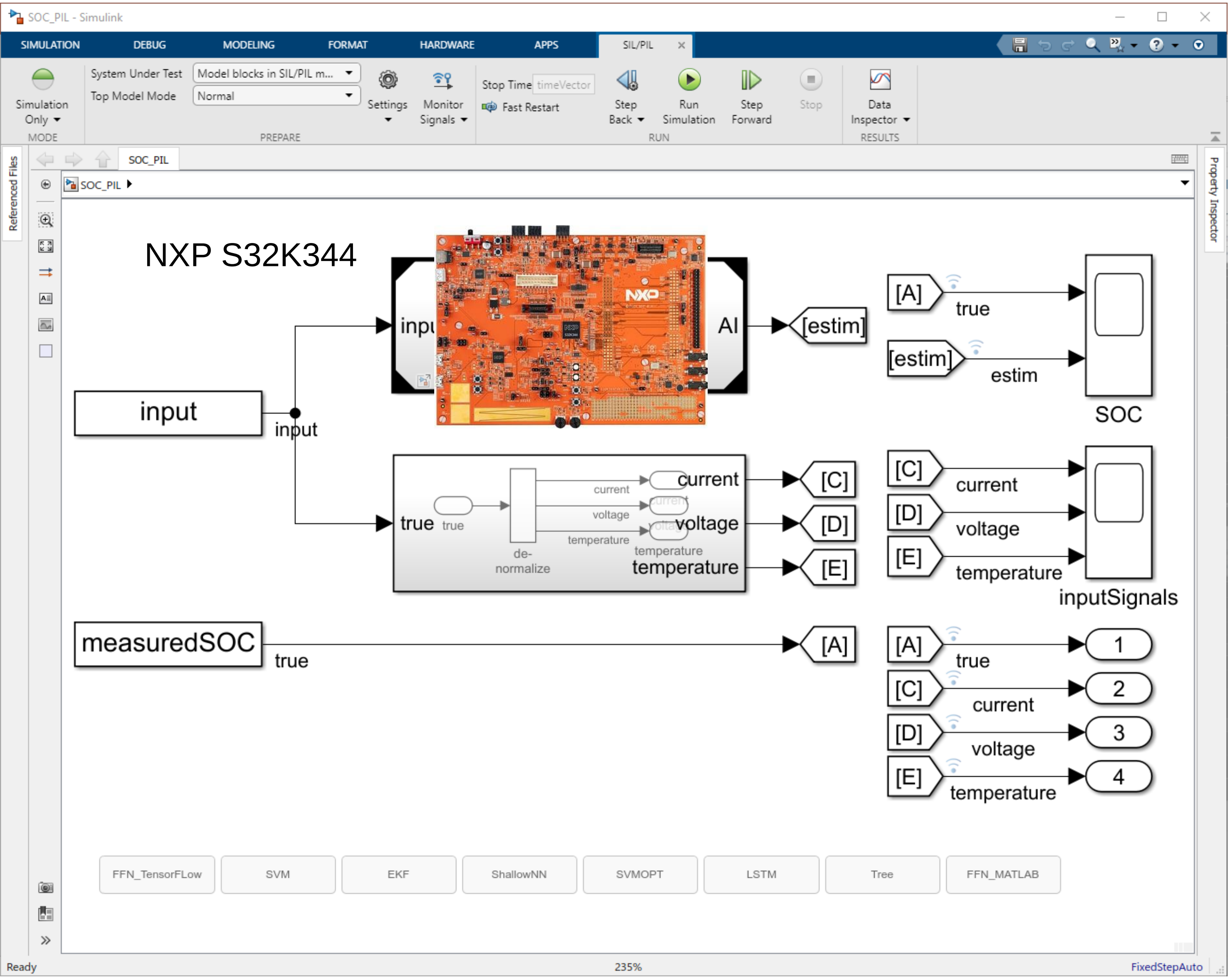


With Variant Subsystems we can implement several AI functions in the same model and try them one at a time

Data Preparation	AI Modeling	Simulation & Test	Deployment

Voltage	Current	Temperature
0.7510	0.3851	0.3031
0.7510	0.3852	0.3046
0.7510	0.3852	0.3061
0.7510	0.3852	0.3076
0.7510	0.3852	0.3091

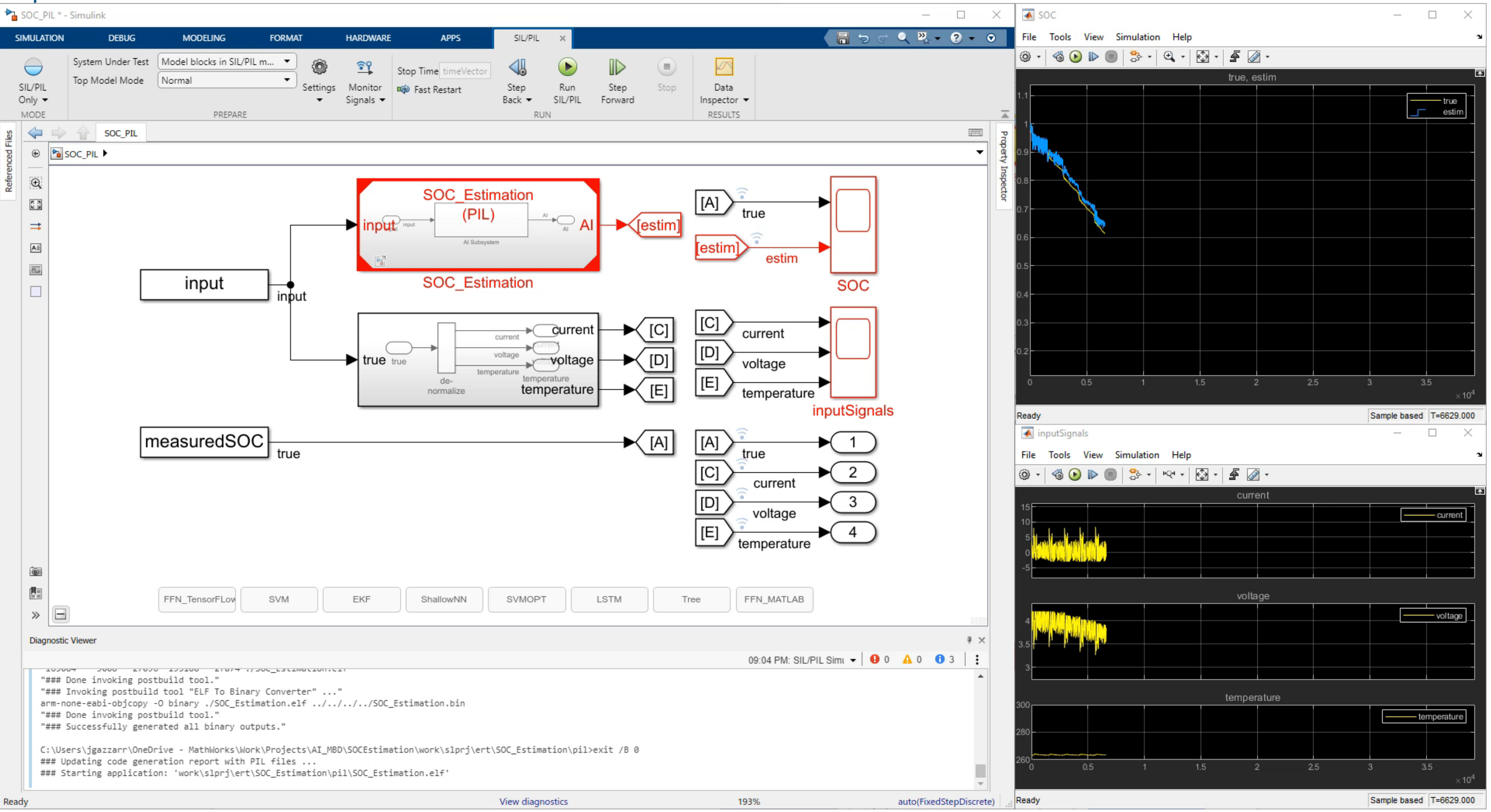
Processor-in-the-Loop (PIL) Testing on ARM Cortex-M7 Processor



Automatic Library-Free C Code

Any CPU
Inc. ARM Cortex-M

life.augmented



Finally, we can configure the model for Processor in the Loop execution

1- Configure hardware and communication ports

2- Select PIL execution

3- Code is generated for the AI function subsystem and downloaded onto the evaluation board

4- The algorithm now runs on target

Tradeoffs and Benchmark

	EKF Extended Kalman Filter	**Tree** Fine Regression Tree	**FFN** 1-hidden layer Feedforward Network	**LSTM** Stacked Long Short-Term Memory Network
Training Speed	N/A	🟢	🟠	🔴
Interpretability	🟢	🟢	🔴	🔴
Inference Speed *	🟠	🟢	🟢	🔴
Model Size *	🟢	🟠	🟢	🔴
Accuracy (RMSE)	🟢	🟠	🟢	🟢

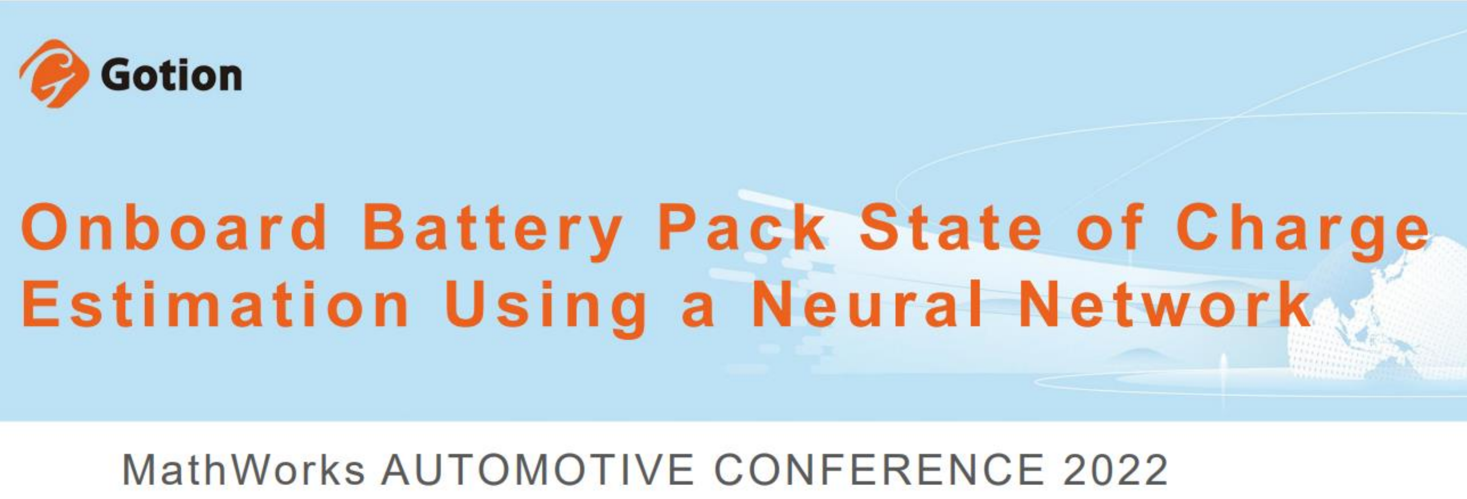
Results are specific to this example

Here is a comparison among AI methods and the EKF benchmark

There is a trade-off among training effort, predictive accuracy, and on-target execution time

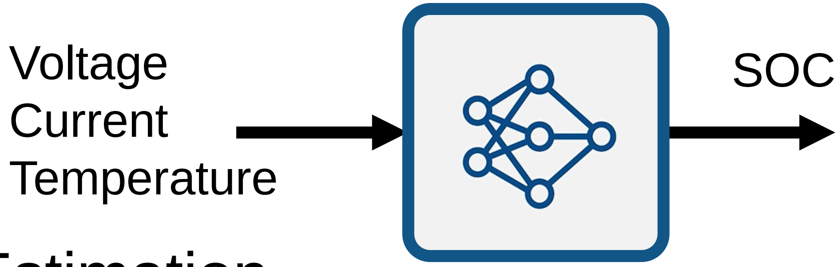
** NXP S32K344 board*

User Stories



MathWorks Automotive Conference 2022 videos available on demand

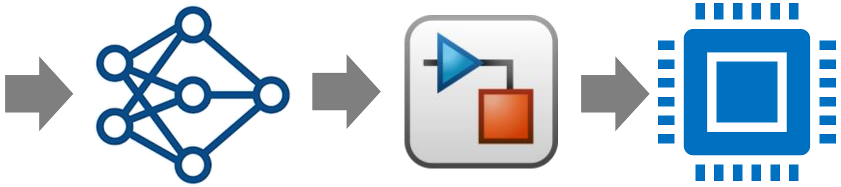
Summary



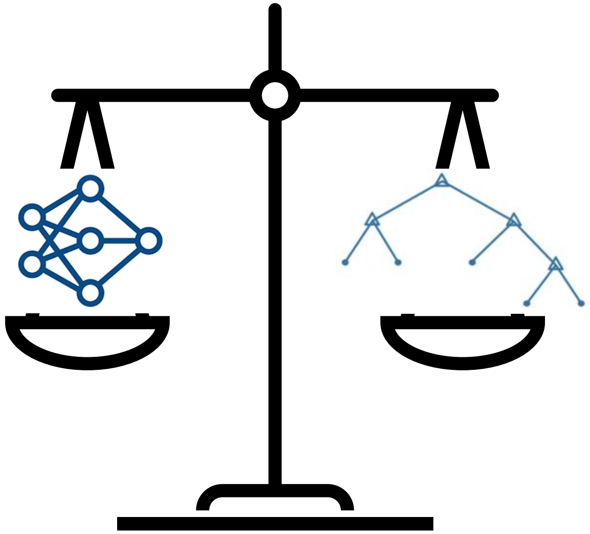
- Develop AI-based Virtual Sensor for Battery SOC Estimation

- Workflow - From Data Acquisition to Hardware Deployment

Voltage	Current	Temperature
0.7510	0.3851	0.3031
0.7510	0.3852	0.3046
0.7510	0.3852	0.3061
0.7510	0.3852	0.3076
0.7510	0.3852	0.3091



- Compare Different AI Methods



MATLAB EXPO

Thank you

MathWorks®