MATLAB EXPO

Developing Electrified Propulsion Systems for a Sustainable Future

Rahul Choudhary, MathWorks

Ramanuja Jagannathan, MathWorks

MathWorks®

MathWorks
@MathWorks

Share the EXPO experience
#MATLABEXPO

www.linkedin.com/in/rch193

https://www.linkedin.com/in/jrvinayak/

Megatrend: Electrification of Everything

From Grid to Vehicle

Clean Energy

Low-carbon fuels

Energy efficiency and electrification

Battery storage

Design Challenges

- Every bit of energy wasted reduces efficiency/ range
- Improperly sized components can add weight or waste energy
- Overly complex solutions may reduce efficiency if not optimized
- Analysis becomes important to predict issues

On the Grid Side

IEEE1676 – Control Architecture for High Power Electronic

Control Transients

Hours

- Peak Shaving/Load Leveling
- Energy Markets
- Integration of Renewables
- Islanding Operation (No Grid)

Seconds

- Voltage/Frequency Regulation
- Transient Smoothing
- Reactive Power Control

m/µ-Seconds

- Switched-Mode Control
- Harmonic Analysis

System Control

Application Control

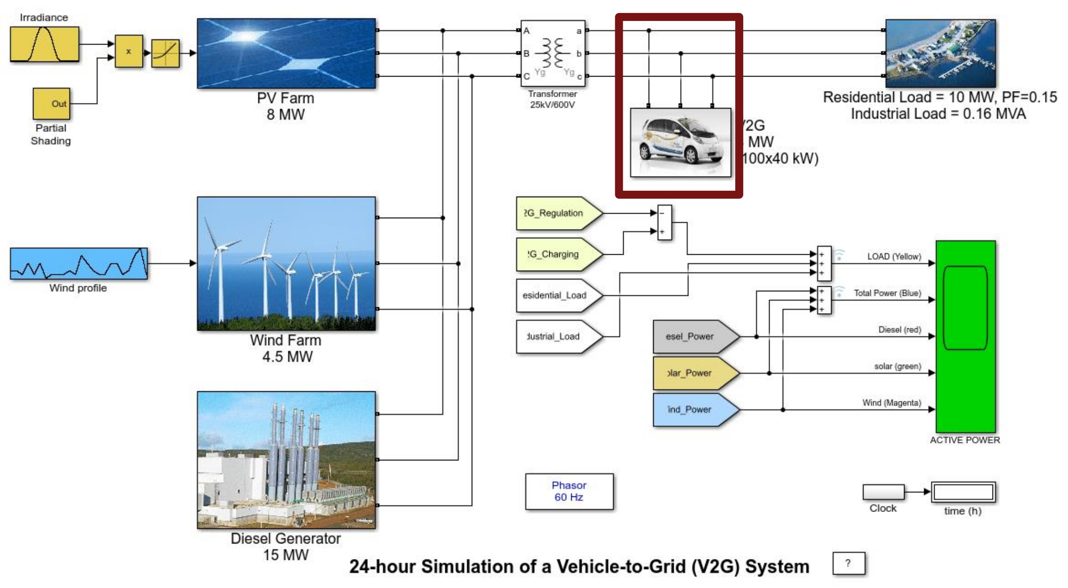
Converter Control

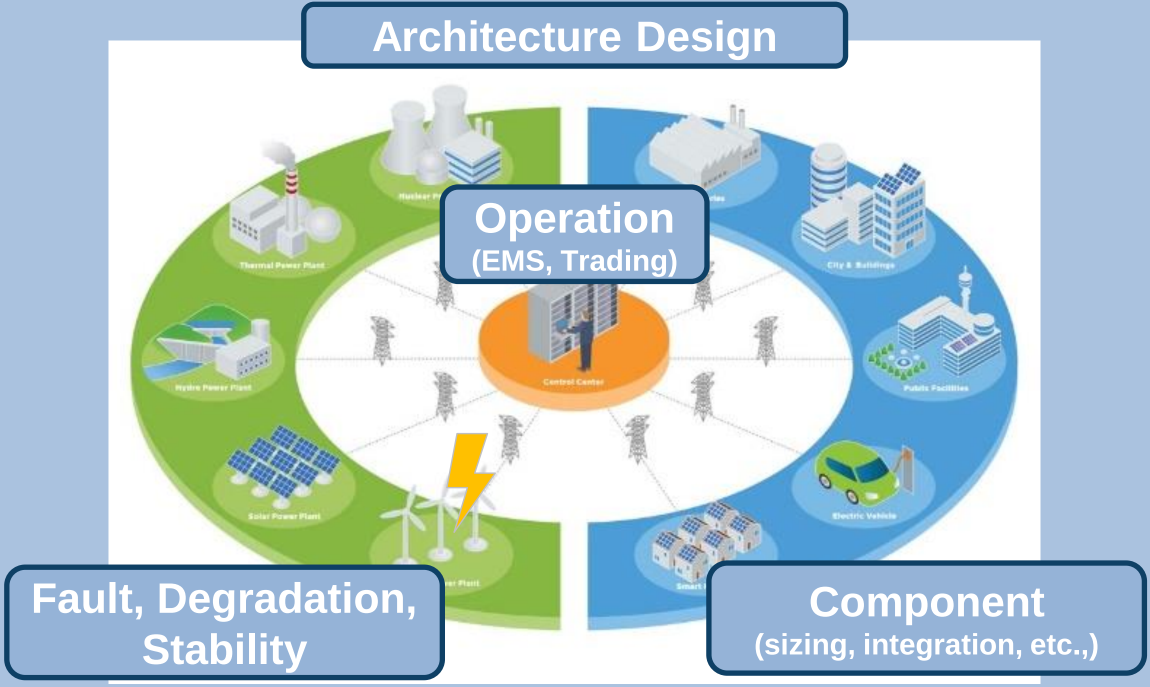
Switching Control

Hardware Control

Design Flexibility

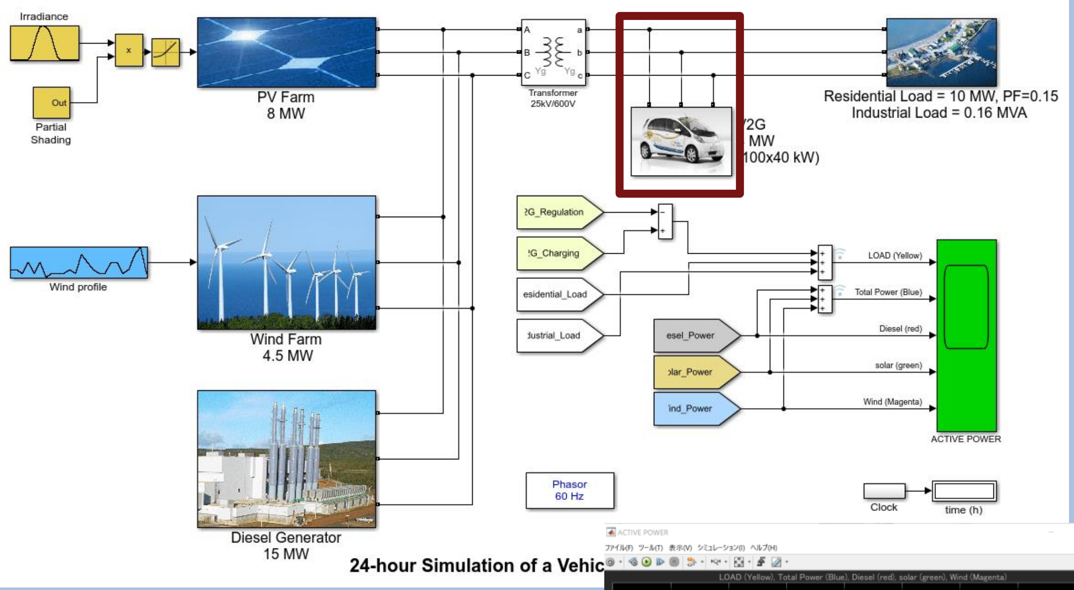
Low Fidelity



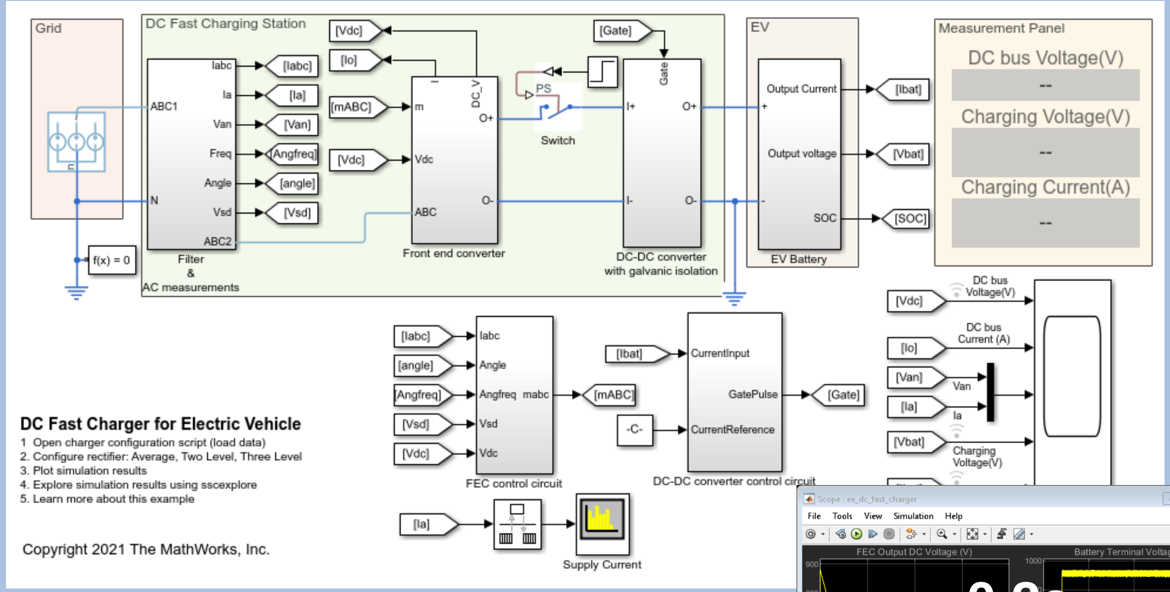


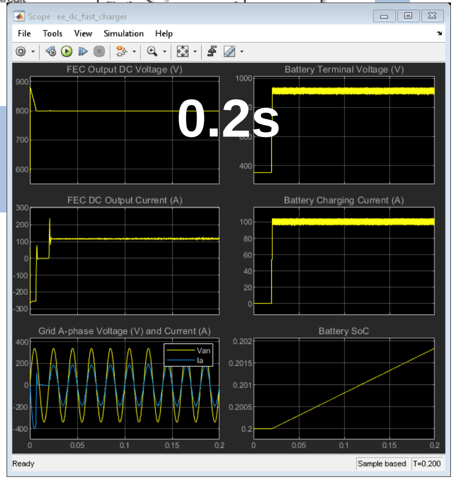
Design Flexibility

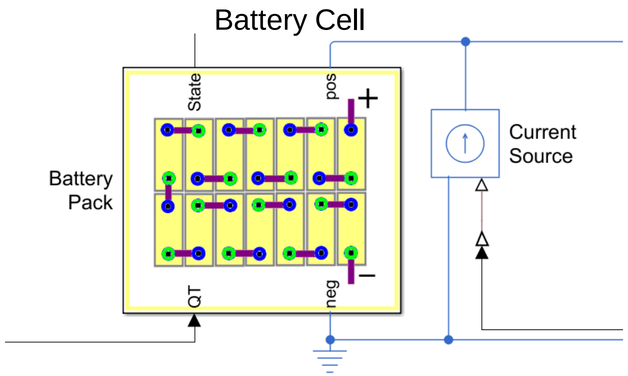
Low Fidelity



High Fidelity

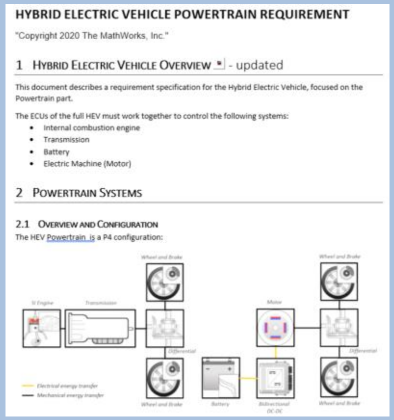






Systems Engineering: System Composer

Requirement


HYBRID ELECTRIC VEHICLE POWERTRAIN REQUIREMENT

"Copyright 2020 The MathWorks, Inc."

1 Hybrid Electric Vehicle Overview - updated

This document describes a requirement specification for the Hybrid Electric Vehicle, focused on the Powertrain part.

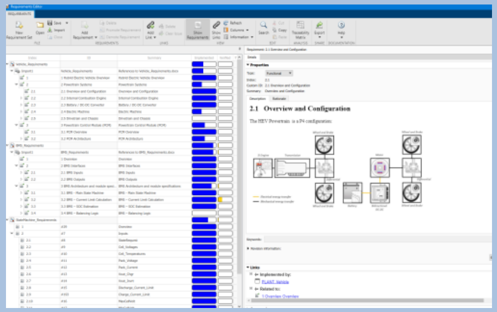
The ECUs of the full HEV must work together to control the following systems:

- Internal combustion engine
- Transmission
- Battery
- Electric Machine (Motor)

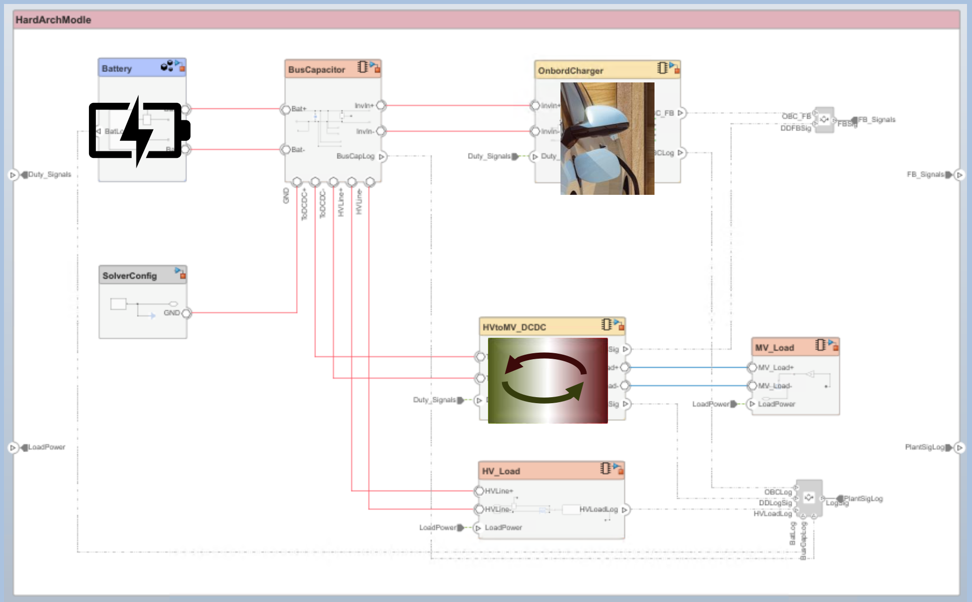
2 Powertrain Systems

2.1 Overview and Configuration

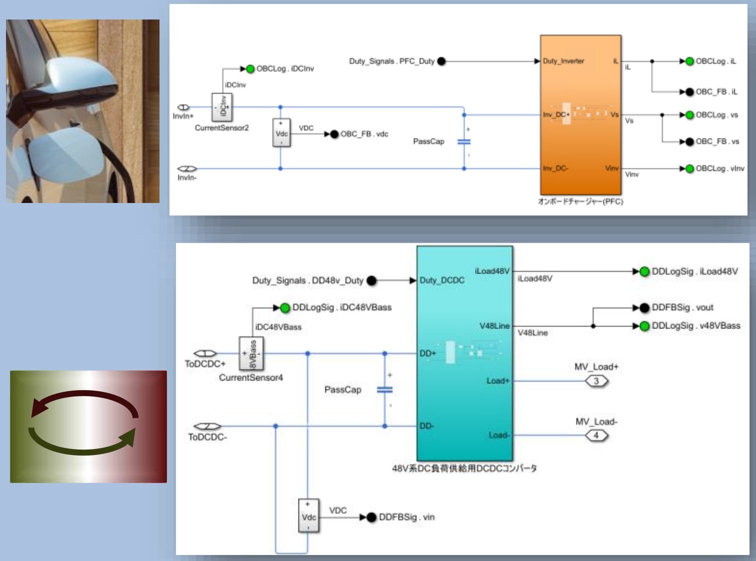
The HEV Powertrain is a P4 configuration:

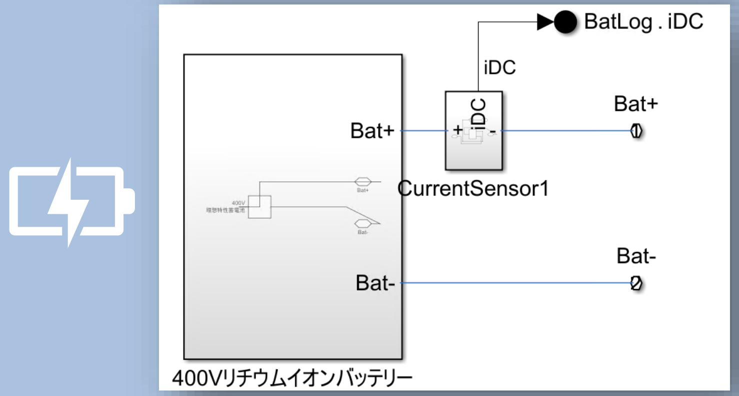


System Architecture

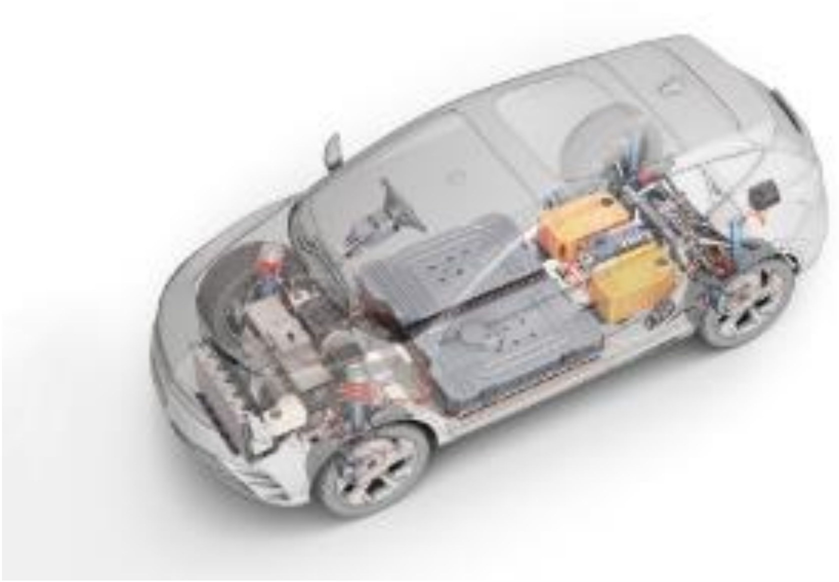


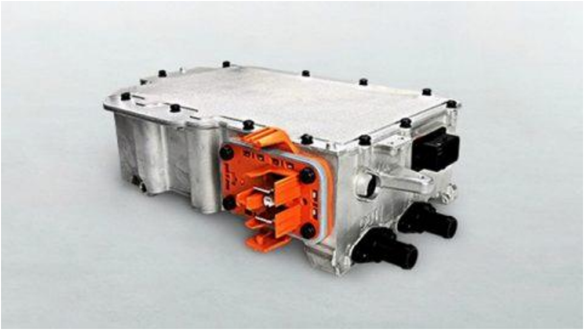
Component



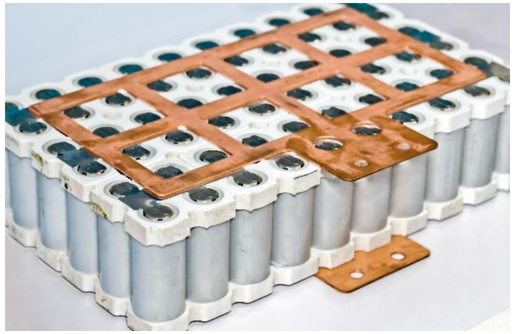


On e-Mobility Side

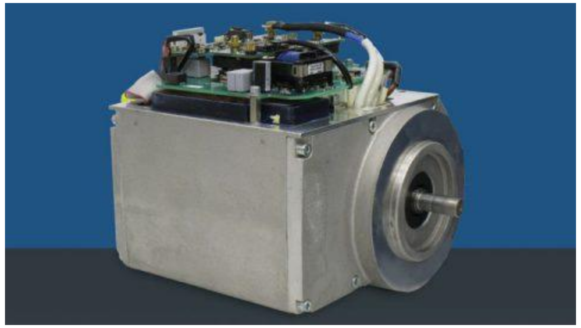




Power Electronics

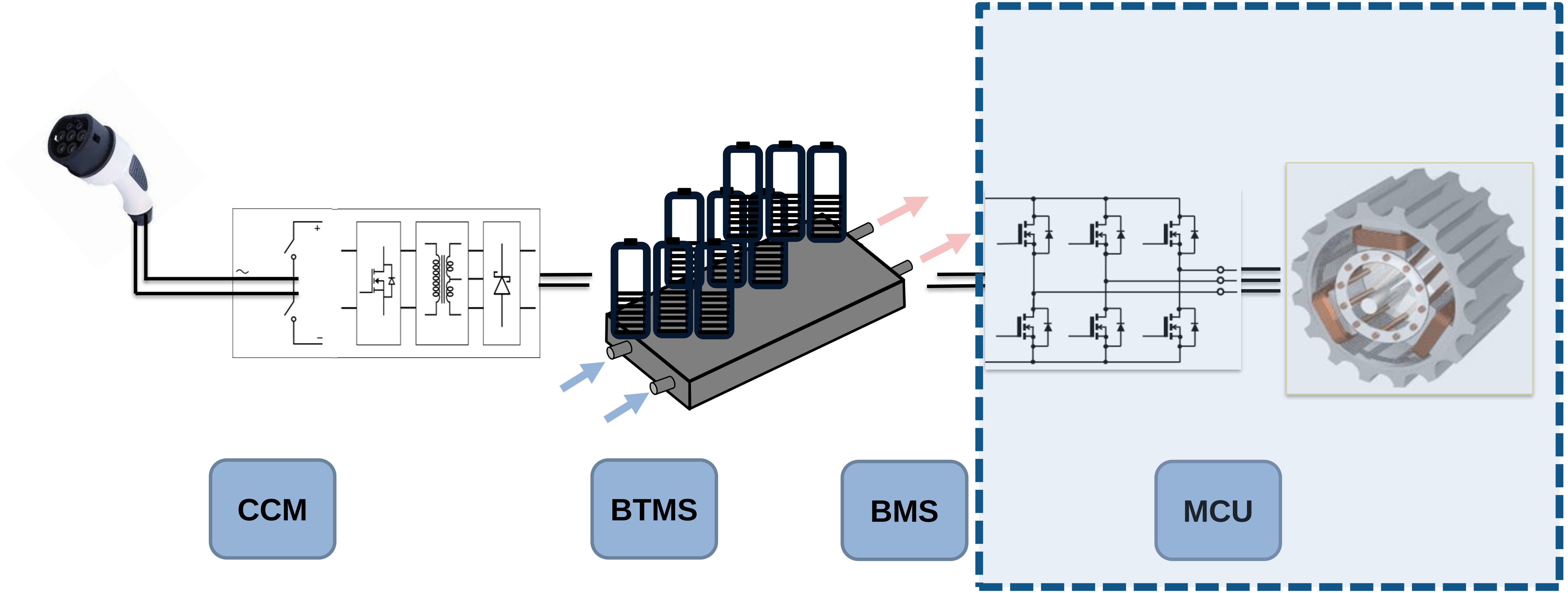


Battery Management



Motor Control

Electric Vehicle System



Common Challenges with Motor Control Development

- Developing accurate motor models with varying model fidelity
- Motor constraint curves and Identifying optimal operating condition
- Tune Control loop gains to get the desired performance
- Code generation and Deployment to a target hardware

ebm-papst Develops Electric Auxiliary Oil Pump for Automatic Transmissions Using Model-Based Design

ebm-papst's automotive auxiliary oil pump without a pressure sensor.

Challenge

Develop, verify, and calibrate an automotive auxiliary oil pump without a pressure sensor

Solution

Use Model-Based Design to model and simulate the controller, and use Simulink Real-Time to validate the design and automate system identification and calibration

Results

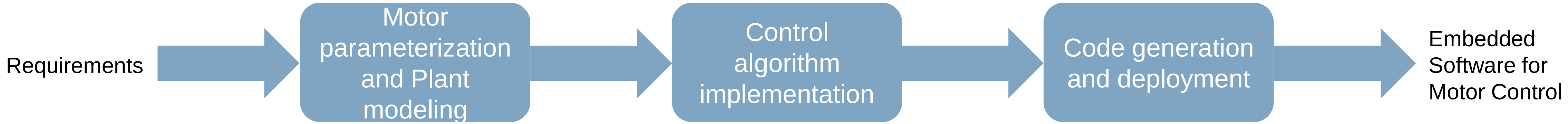
- Overall development time halved
- System investigation time cut by 60%
- Deployment on specified microcontroller supported

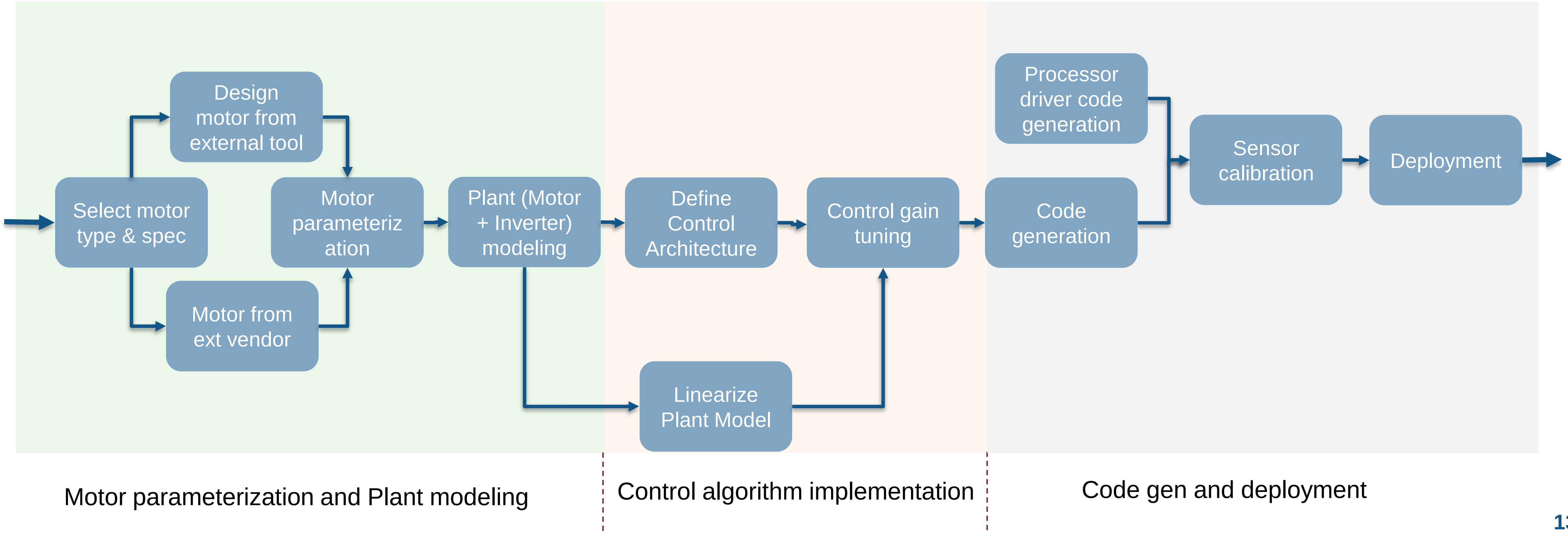
"Given the time pressure we faced, Model-Based Design was our only chance to design a controller that satisfied our customer's pressure regulation requirements, build a test rig to automate labor-intensive tests, and rapidly set up a calibration process in production to maximize performance."

– *Jens Löffler, ebm-papst*

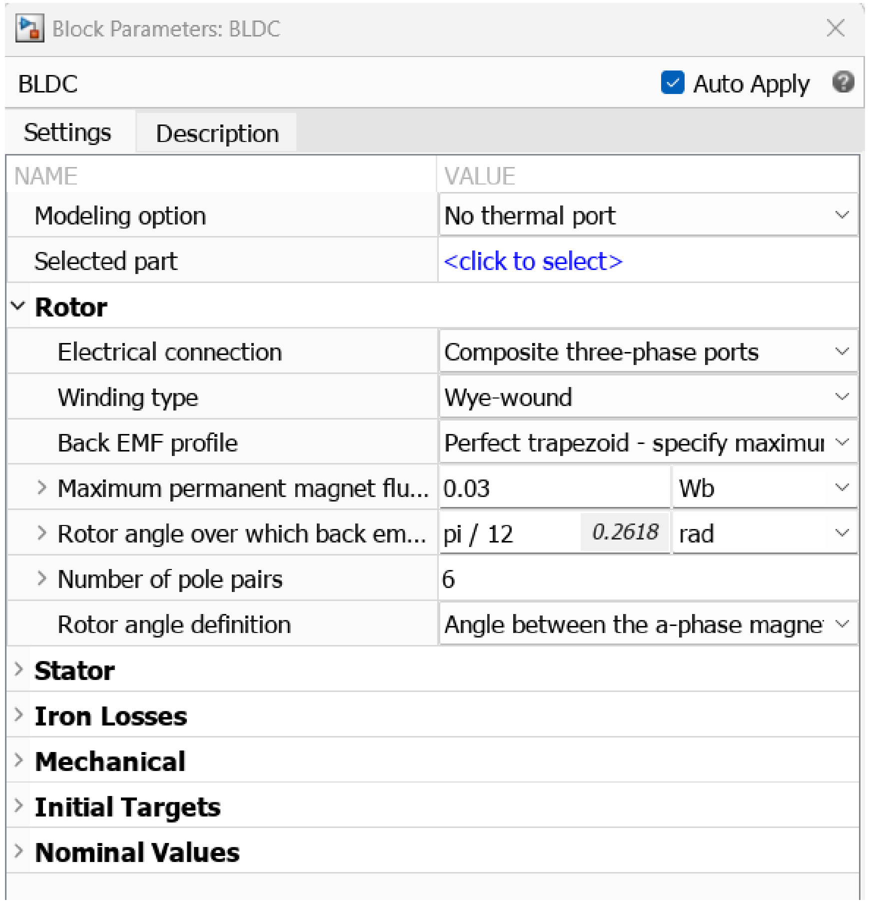
Link to user story

Workflow for Motor Control Development

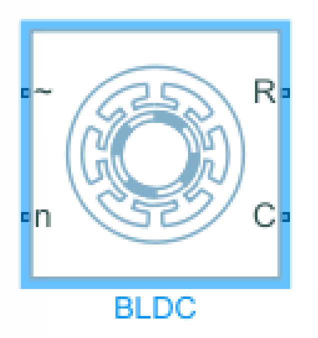


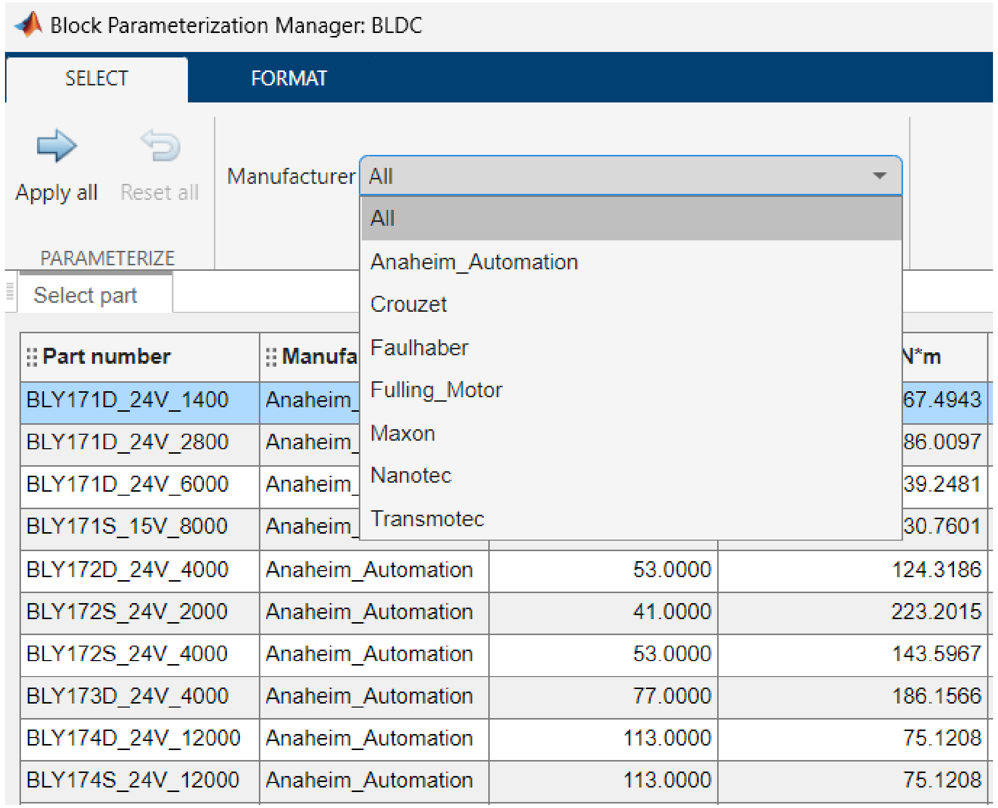


Motor Parametrization



From Motor Datasheet





From Pre-parametrized Motors

Parameter Estimation by Running Instrumented Tests

- Instrumented tests running on the target
- Sensor-based and Sensorless modes available
- Supports PMSM and Induction Motor

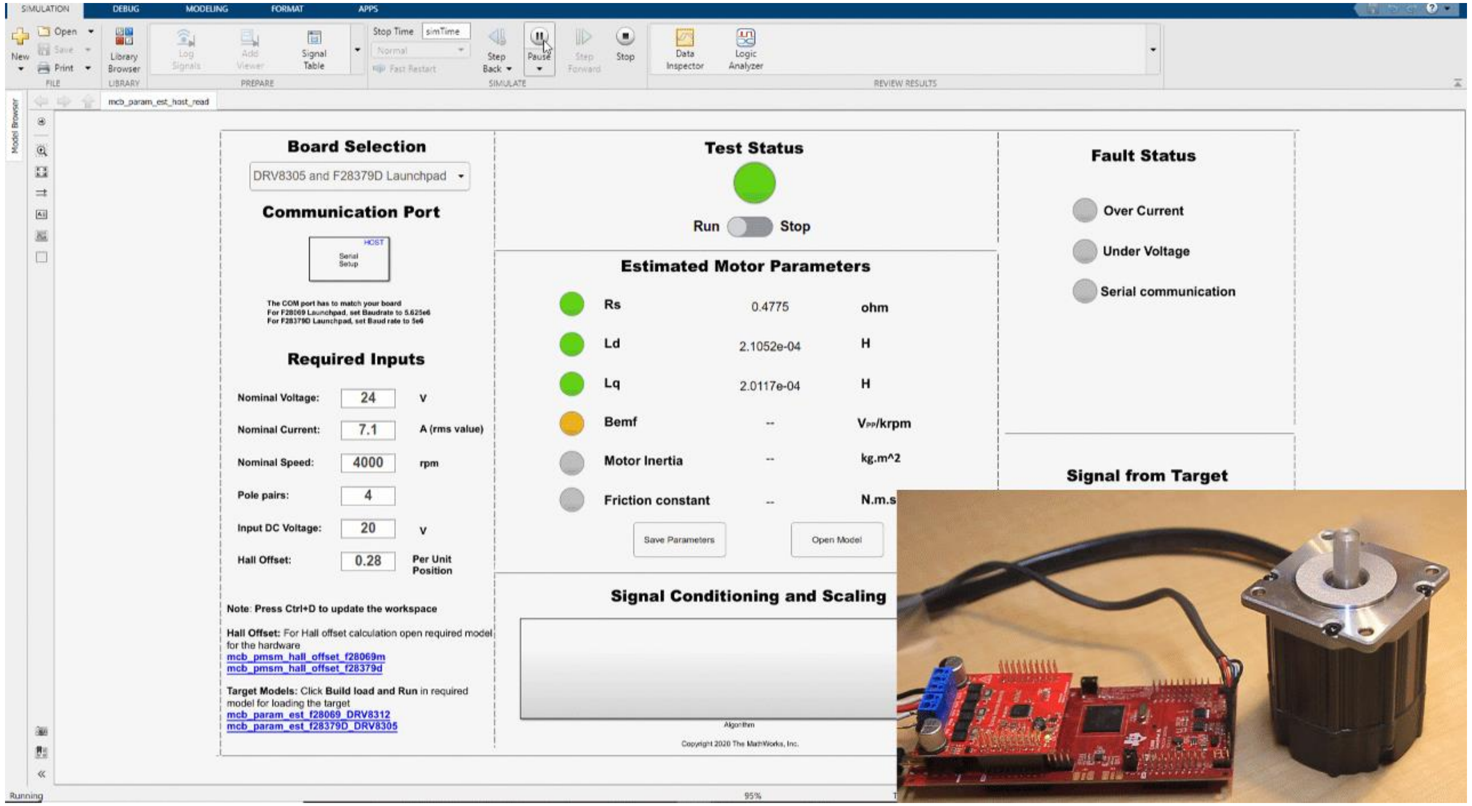
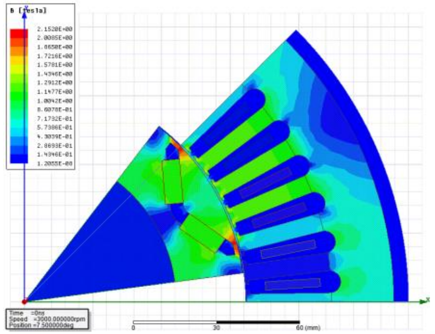
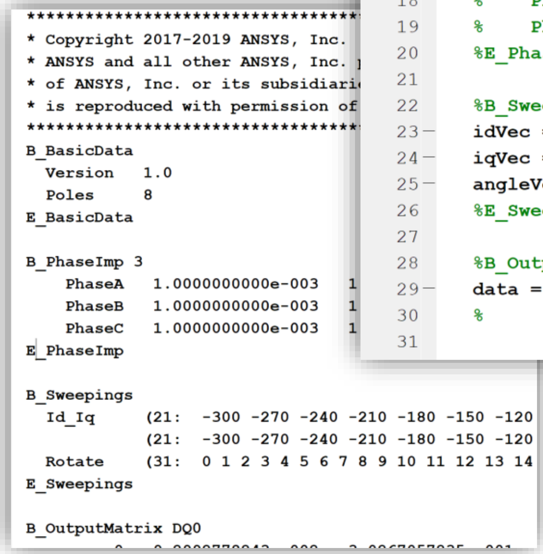


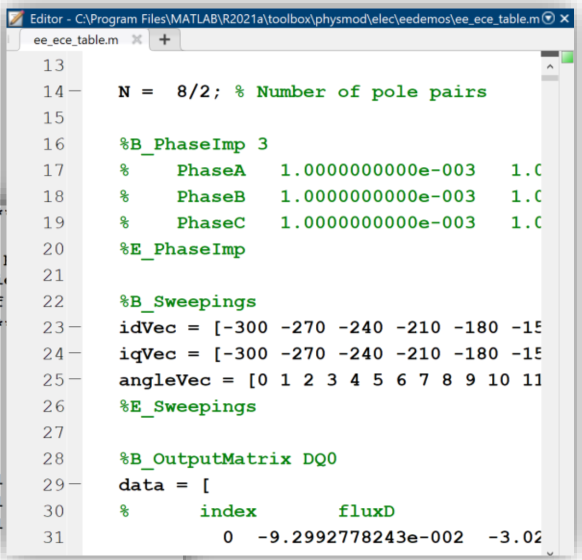
Figure : gif showing parameter estimation capability with TI C2000 hardware

Motor Parametrization using FEA tools

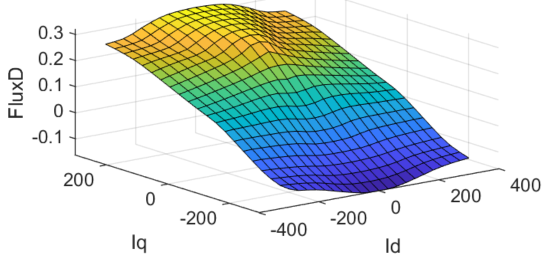


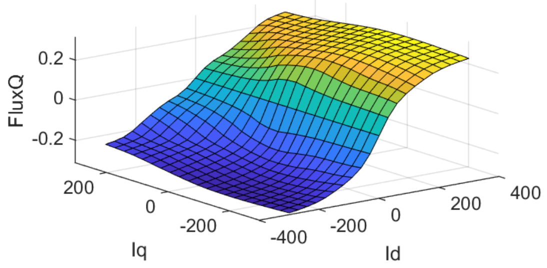


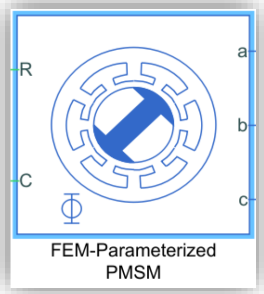
```
**********************************
* Copyright 2017-2019 ANSYS, Inc.
* ANSYS and all other ANSYS, Inc. 
* of ANSYS, Inc. or its subsidiari
* is reproduced with permission of
**********************************
B_BasicData
  Version    1.0
  Poles      8
E_BasicData

B_PhaseImp 3
    PhaseA    1.0000000000e-003    1
    PhaseB    1.0000000000e-003    1
    PhaseC    1.0000000000e-003    1
E_PhaseImp

B_Sweepings
  Id_Iq      (21:  -300 -270 -240 -210 -180 -150 -120
             (21:  -300 -270 -240 -210 -180 -150 -120
  Rotate     (31:  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
E_Sweepings

B_OutputMatrix DQ0
```

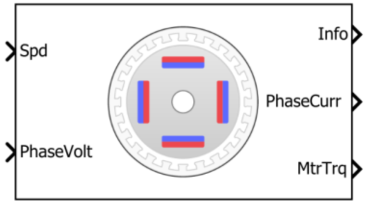



```
N =  8/2; % Number of pole pairs

%B_PhaseImp 3
%     PhaseA    1.0000000000e-003    1.0
%     PhaseB    1.0000000000e-003    1.0
%     PhaseC    1.0000000000e-003    1.0
%E_PhaseImp

%B_Sweepings
idVec = [-300 -270 -240 -210 -180 -15
iqVec = [-300 -270 -240 -210 -180 -15
angleVec = [0 1 2 3 4 5 6 7 8 9 10 11
%E_Sweepings

%B_OutputMatrix DQ0
data = [
%       index        fluxD
           0  -9.2992778243e-002   -3.02
```



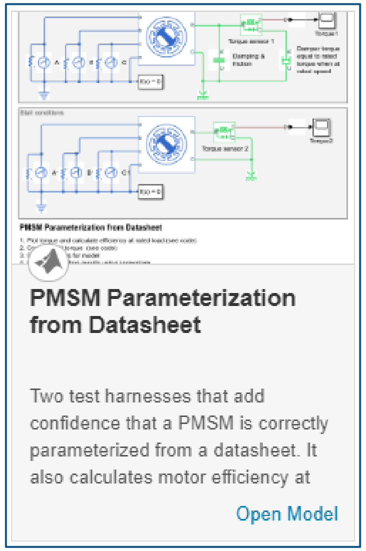


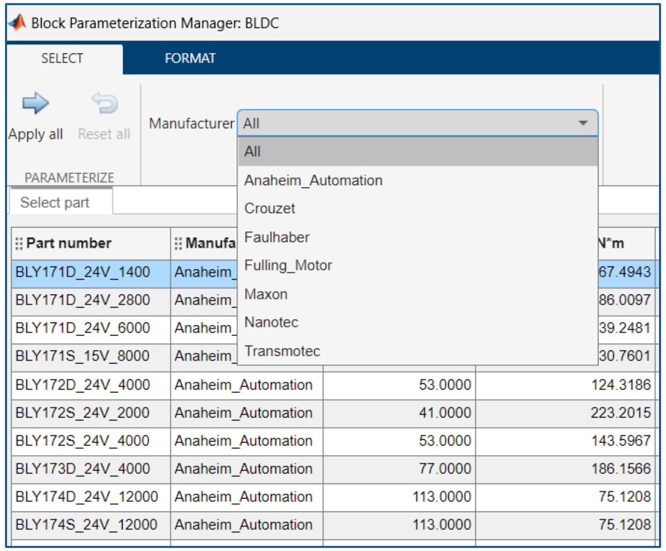


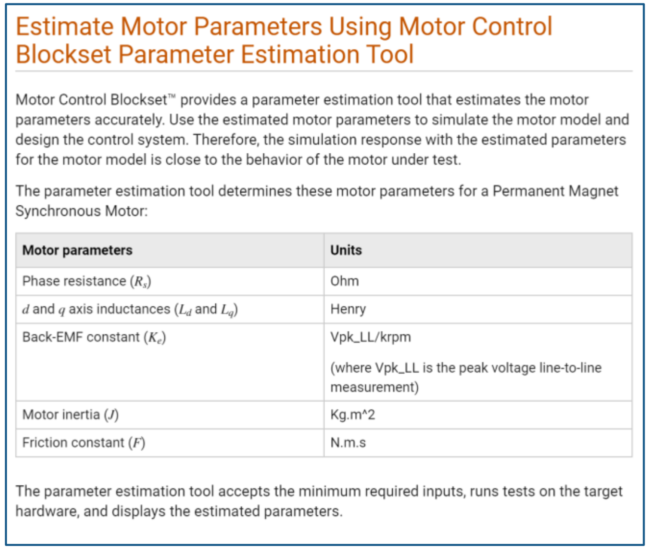


Flux-Based PMSM

Parameterizing the motor models





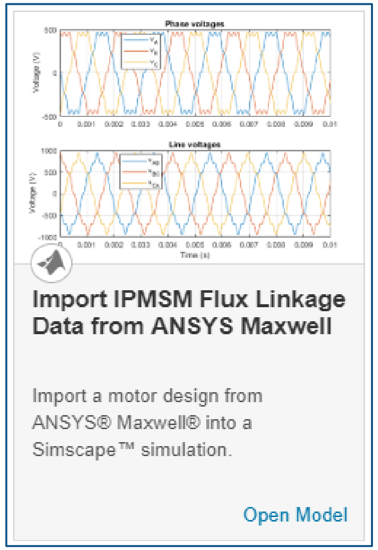

Estimate Motor Parameters Using Motor Control Blockset Parameter Estimation Tool

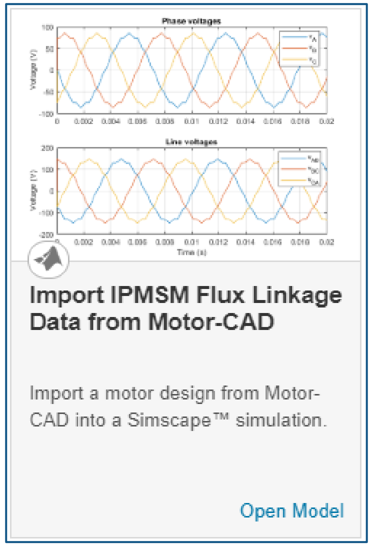
Motor Control Blockset™ provides a parameter estimation tool that estimates the motor parameters accurately. Use the estimated motor parameters to simulate the motor model and design the control system. Therefore, the simulation response with the estimated parameters for the motor model is close to the behavior of the motor under test.

The parameter estimation tool determines these motor parameters for a Permanent Magnet Synchronous Motor:

Motor parameters	Units
Phase resistance (R_s)	Ohm
d and q axis inductances (L_d and L_q)	Henry
Back-EMF constant (K_e)	Vpk_LL/krpm (where Vpk_LL is the peak voltage line-to-line measurement)
Motor inertia (J)	Kg.m^2
Friction constant (F)	N.m.s

The parameter estimation tool accepts the minimum required inputs, runs tests on the target hardware, and displays the estimated parameters.





From datasheet

From Pre-parametrization

From Instrumented tests running on the Hardware

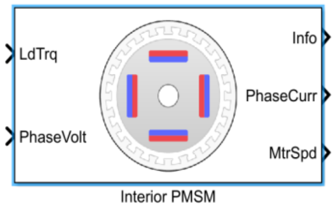
From FEA tools such as ANSYS Maxwell, JMAG, Motor-CAD

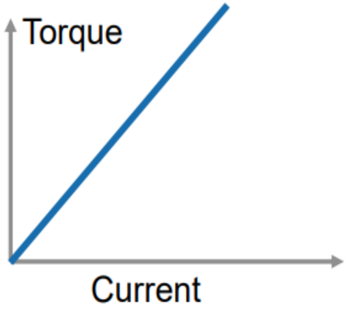
Lumped Parameters

Saturation + Spatial Harmonics

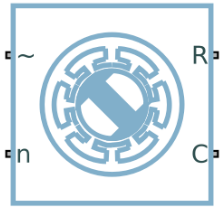
17

Right model fidelity for Motor & Inverter Model

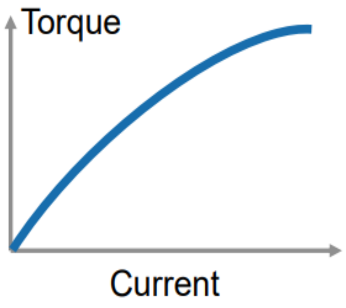




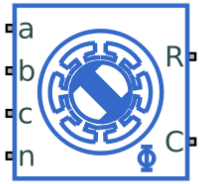
Lumped Parameter



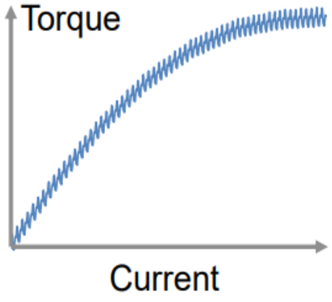
PMSM



Saturation

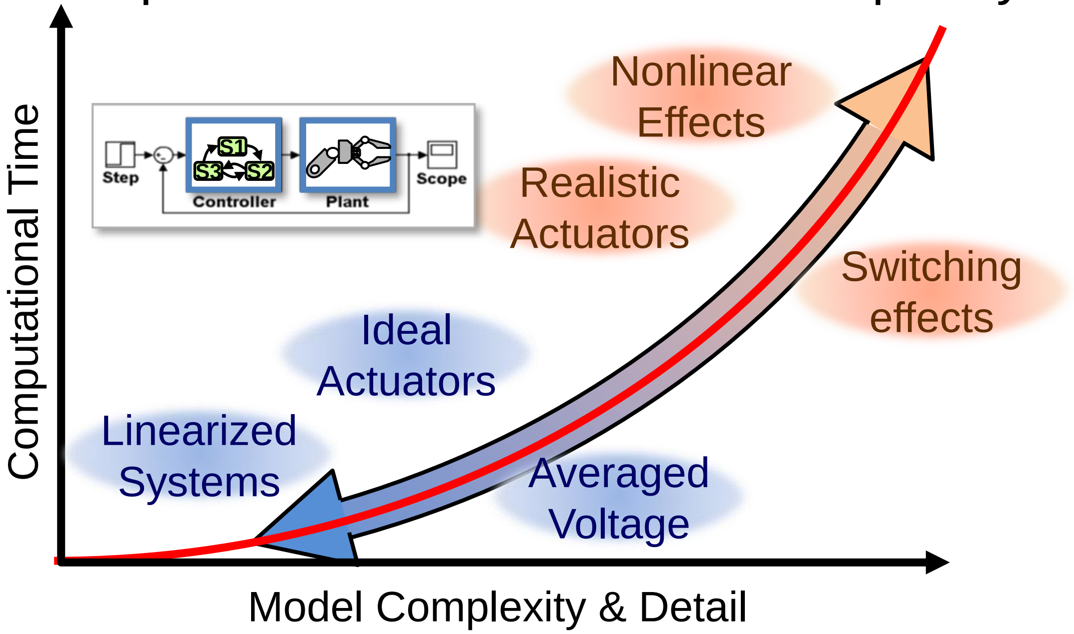


FEM-Parameterized PMSM



Saturation +
Spatial Harmonics

Computational Time vs. Model Complexity



Configure your model to balance simulation speed and model fidelity.

Motor Constraint Curves and Characteristics

- Display Motor Constraints such as
  - MTPA curve (Maximum Torque per Ampere)
  - MTPV curve (Maximum Torque per Voltage)
  - Voltage Limit curve
  - Current Limit
- Exploration Motor Characteristics

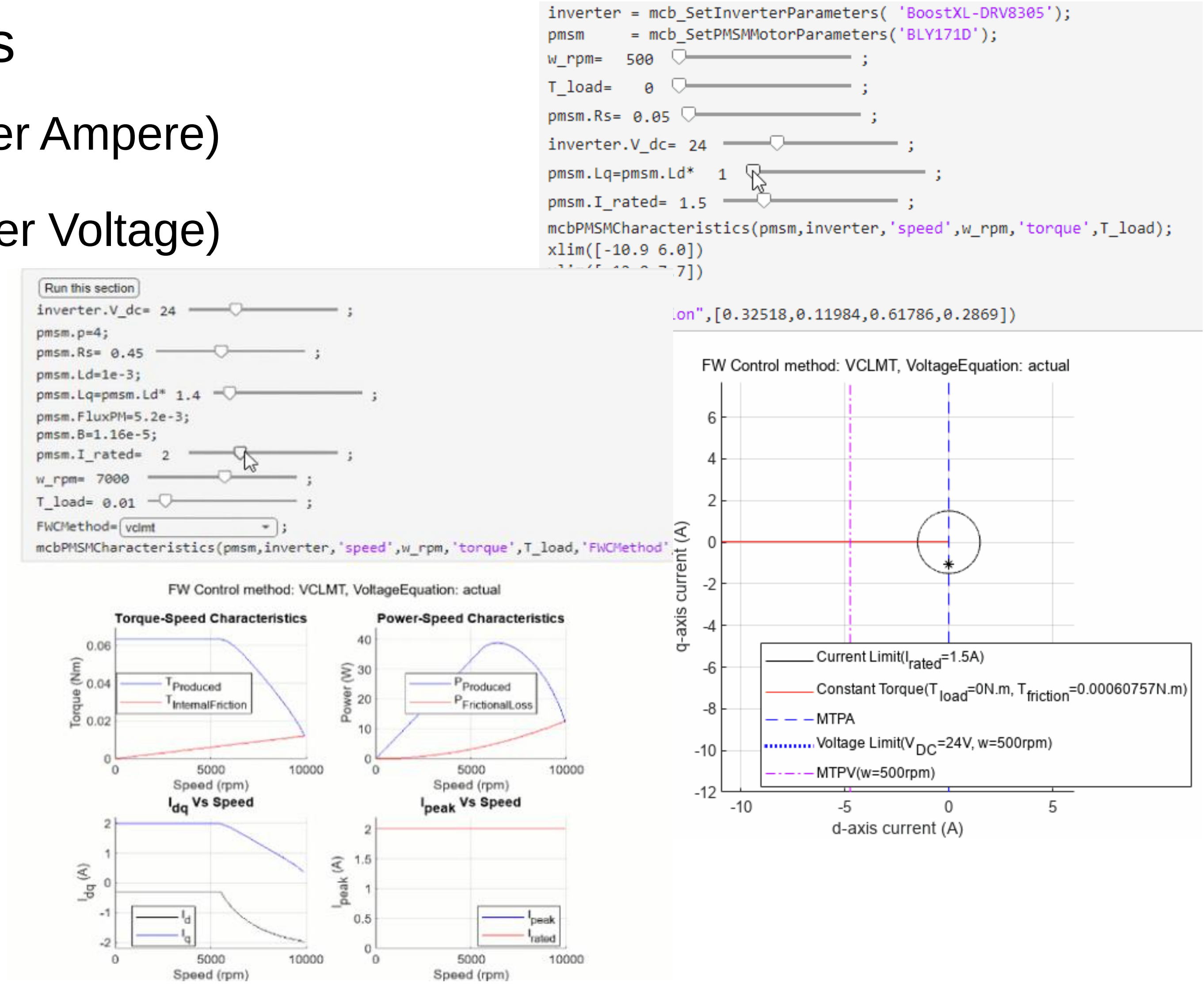
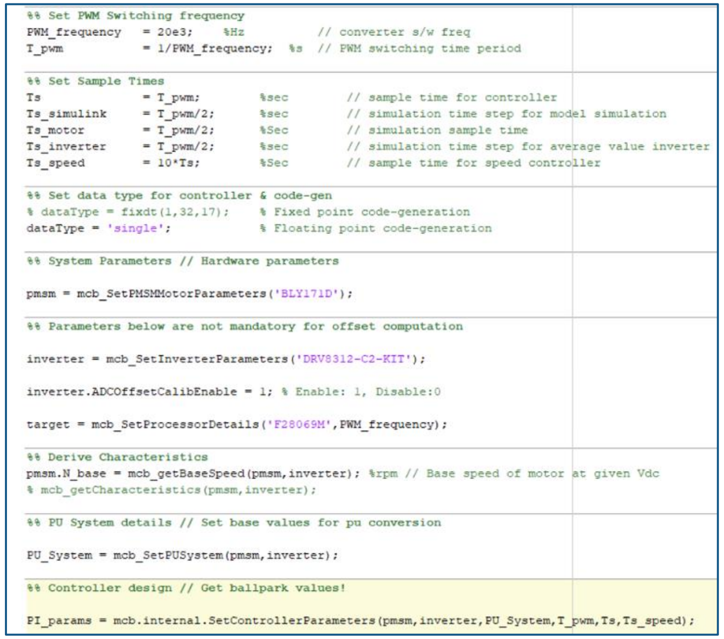


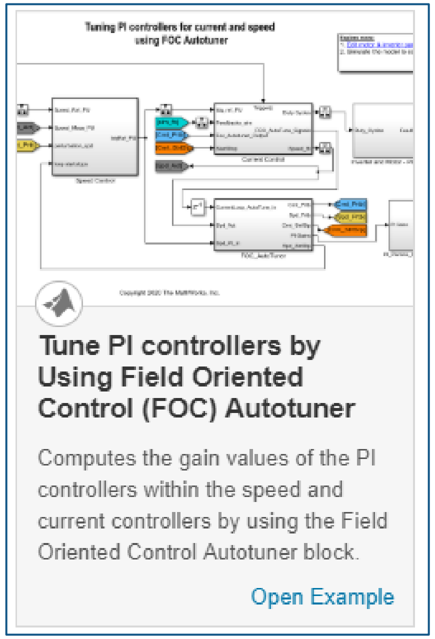
Figure : Dependency of curves and characteristics on parameters changes

Ref: https://www.mathworks.com/help/mcb/ug/pmsm-characteristics-constraint-curves.html

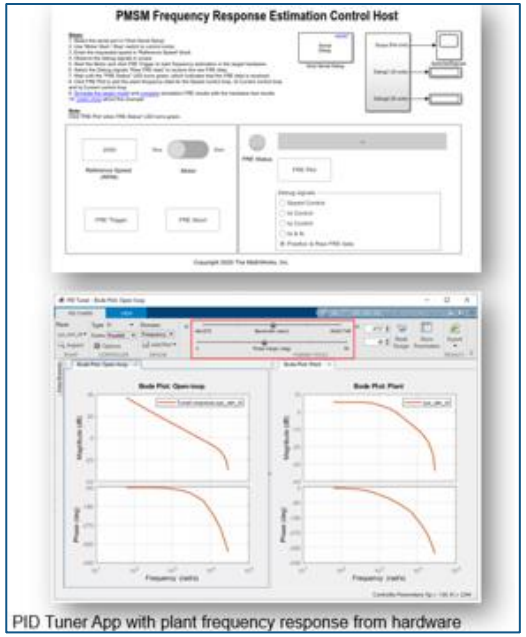
Control loop gain tuning


```
%% Set PWM Switching frequency
PWM_frequency   = 20e3;     %Hz          // converter s/w freq
T_pwm           = 1/PWM_frequency;  %s   // PWM switching time period

%% Set Sample Times
Ts              = T_pwm;        %sec       // sample time for controller
Ts_simulink     = T_pwm/2;      %sec       // simulation time step for model simulation
Ts_motor        = T_pwm/2;      %Sec       // simulation sample time
Ts_inverter     = T_pwm/2;      %sec       // simulation time step for average value inverter
Ts_speed        = 10*Ts;        %Sec       // sample time for speed controller

%% Set data type for controller & code-gen
% dataType = fixdt(1,32,17);    % Fixed point code-generation
dataType = 'single';            % Floating point code-generation

%% System Parameters // Hardware parameters

pmsm = mcb_SetPMSMMotorParameters('BLY171D');

%% Parameters below are not mandatory for offset computation

inverter = mcb_SetInverterParameters('DRV8312-C2-KIT');

inverter.ADCOffsetCalibEnable = 1; % Enable: 1, Disable:0

target = mcb_SetProcessorDetails('F28069M',PWM_frequency);

%% Derive Characteristics
pmsm.N_base = mcb_getBaseSpeed(pmsm,inverter); %rpm // Base speed of motor at given Vdc
% mcb_getCharacteristics(pmsm,inverter);

%% PU System details // Set base values for pu conversion

PU_System = mcb_SetPUSystem(pmsm,inverter);

%% Controller design // Get ballpark values!

PI_params = mcb.internal.SetControllerParameters(pmsm,inverter,PU_System,T_pwm,Ts,Ts_speed);
```

Empirical Computation

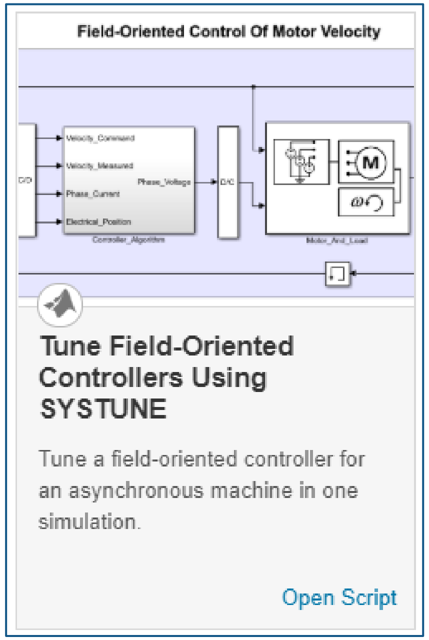


FOC Autotuner



PID Tuner App with plant frequency response from hardware

Online frequency estimation and PID tuner app



Classical Control Theory

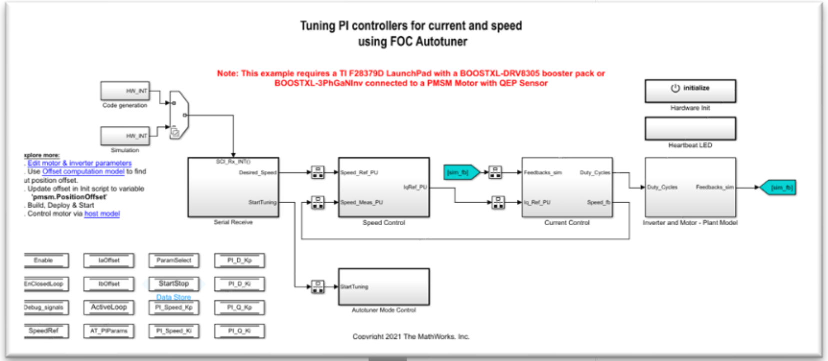
Motor Control Blockset

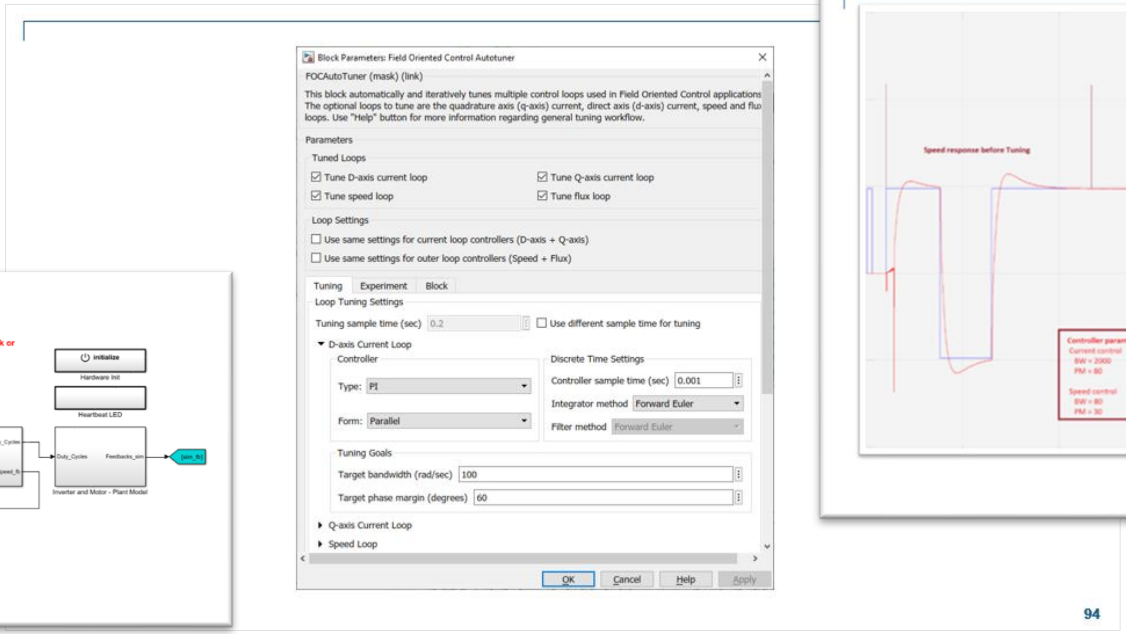
Motor Control Blockset and Simulink Control Design

Simulink Control Design

Control loop gain tuning with FOC Autotuner

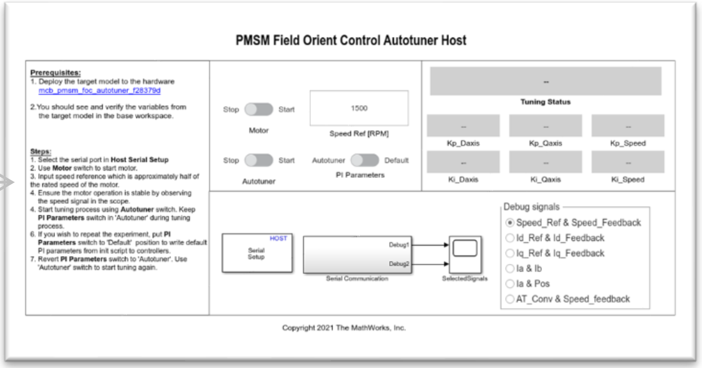
Target model deployed to the hardware





Deployed in hardware

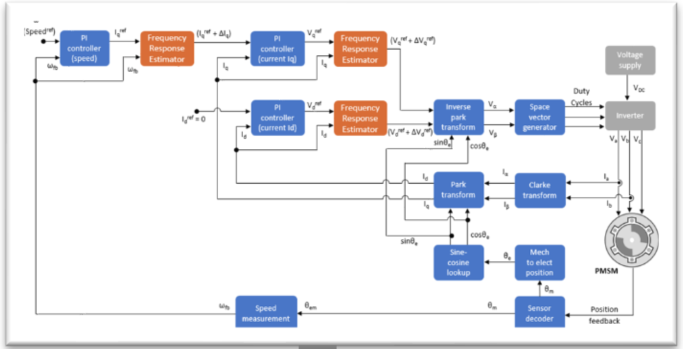
Motor control kit with evaluation board



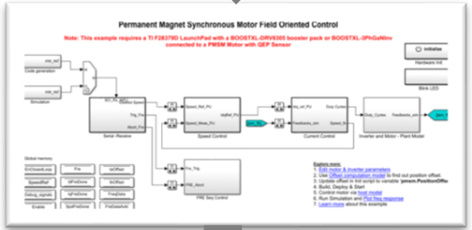
Host model reads the tuned gain values from hardware

Control loop gain tuning Online Frequency Estimation

Online plant frequency response

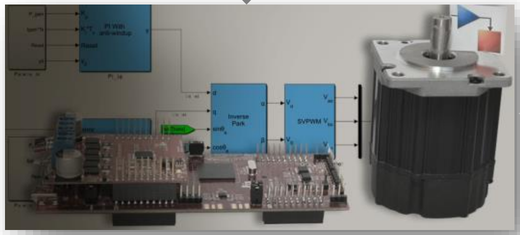


Target model deploy in target

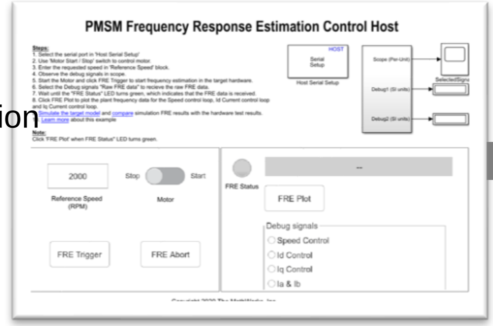


Deployed in hardware

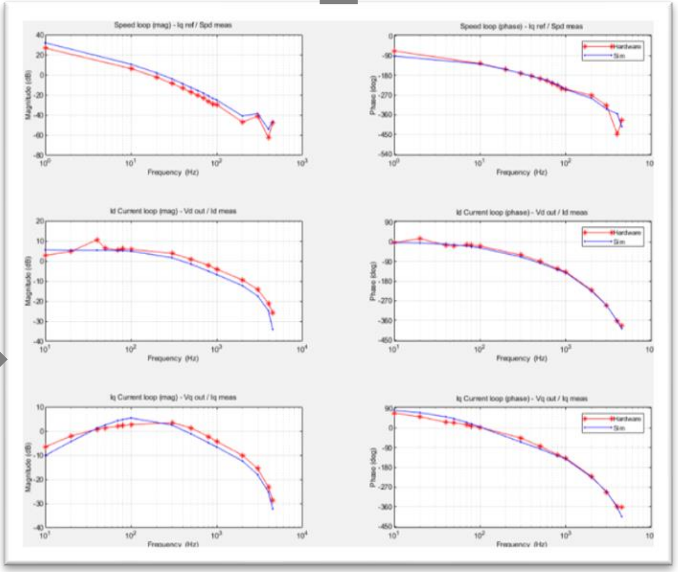
Motor control kit with evaluation board



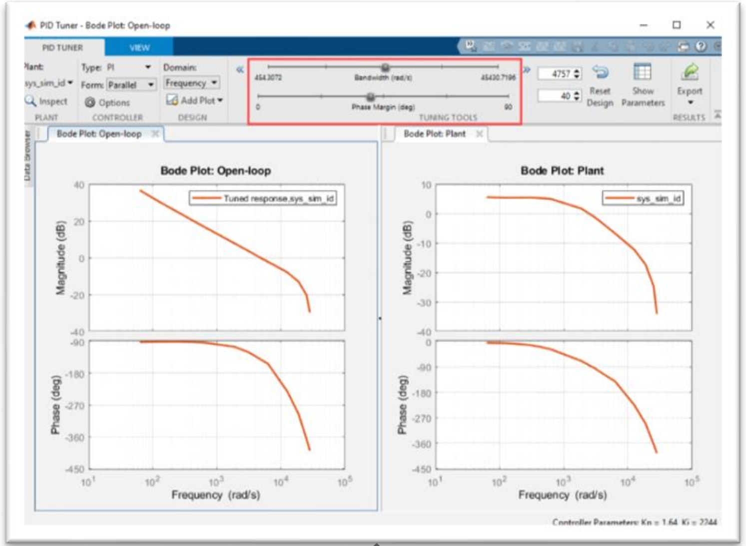
Serial communication



Host model reads the frequency spectrum data from hardware



Compare frequency response data in Sim and hardware testing



Input frequency response data to PID Tuner App

Code Generation and Deployment on embedded device

Hardware Support Package for Driver code

Embedded Coder Support Package for Texas Instruments C2000 Processors by MathWorks Embedded Coder Team STAFF

Generate code optimized for C2000 MCU.

Embedded Coder® Support Package for Texas Instruments C2000™ Processors enables you to run Simulink® models on TI C2000 MCUs. Embedded Coder automatically generates C code for your algorithms and

Field-Oriented Control of PMSM Using NXP™ S32K144 Kit version 1.0 by Shivaprasad Narayan STAFF

The workflow demonstrates Field Oriented Control of a Permanent Magnet Synchronous Motor using NXP™ MCSPTE1AK144: S32K144 Development Kit

FOC-of-PMSMField-Oriented Control of Permanent Magnet Synchronous Motor Using NXP™ S32K144 Development kitThis example implements a motor control system using the NXP™ MCSPTE1AK144 hardware. The

Demo for Motor Control Deployment on Microchip Controllers version 1.0.0 by Brian McKay STAFF

Demo used in MathWorks-Microchip joint webinar: Deploying Motor Control Algorithms on Microchip dsPIC, PIC32, and SAM Controllers.

which includes a dsPIC33E Digital Signal Controller.The demo also require you to download and install the free add-on MPLAB Device Blocks for Simulink: dsPIC, PIC32, and SAM MCU's.View the webinar for a

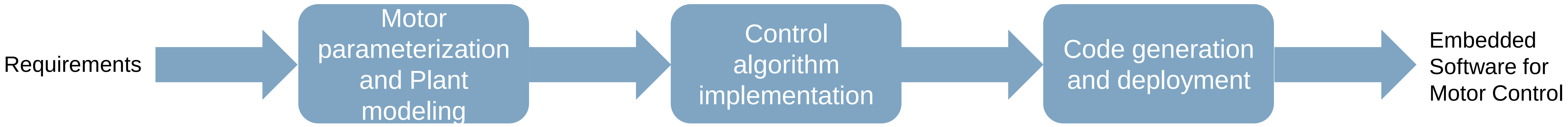
Hand-written Driver code for peripherals

Algorithm-Export Workflows for Custom Hardware

Enables you to use any custom motor-control hardware (hardware not used in the Motor Control Blockset™ examples) to run a three-

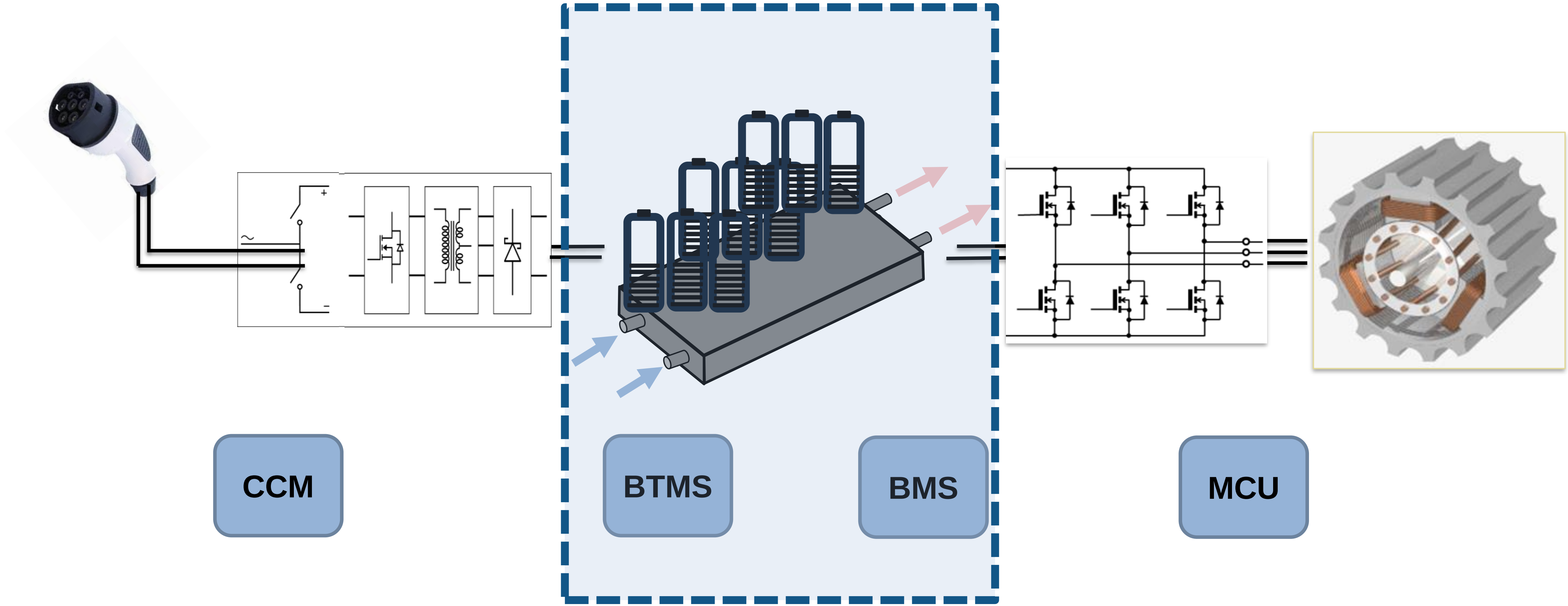
Open Example

Summary



- ✓ Discussed various methods to parameterize the motor and achieve different fidelity levels
- ✓ Utility to display constraints curves as a function of motor parameters
- ✓ Discussed systematic approaches to tune loop gains
- ✓ Deploy to the hardware and validate

Electric Vehicle System

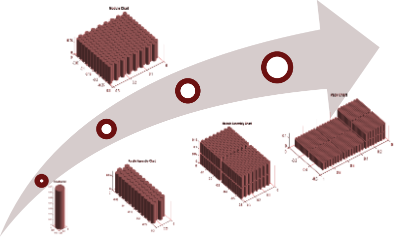


Challenges in developing a BMS

Collaboration Gap

Multi-Domain Modeling Environment

System Analysis



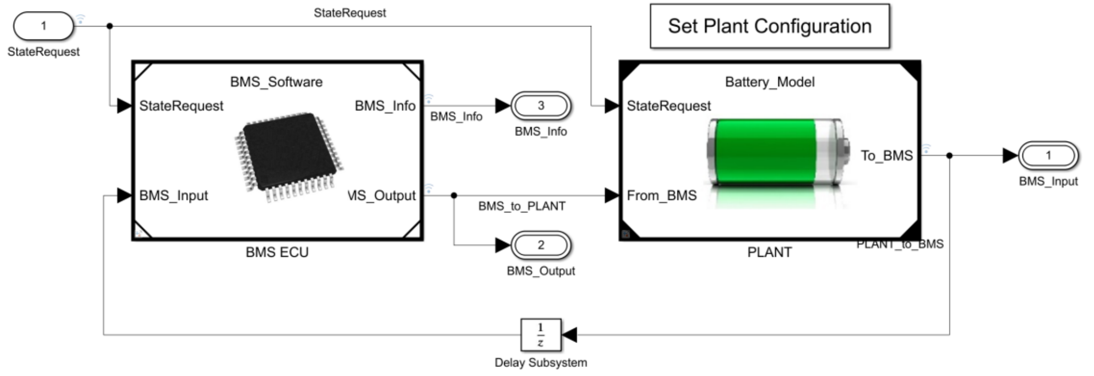
Scaling up from cell to pack using automation

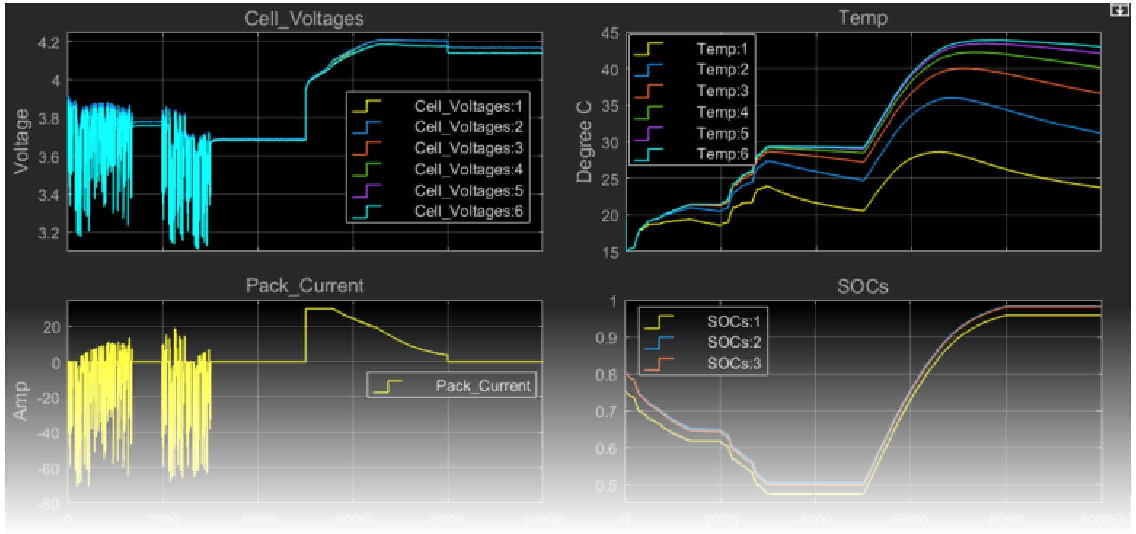
Long Iteration Cycles

Simulations and Code Generation

Safety Critical System

Model V&V and Hardware-In-Loop Testing



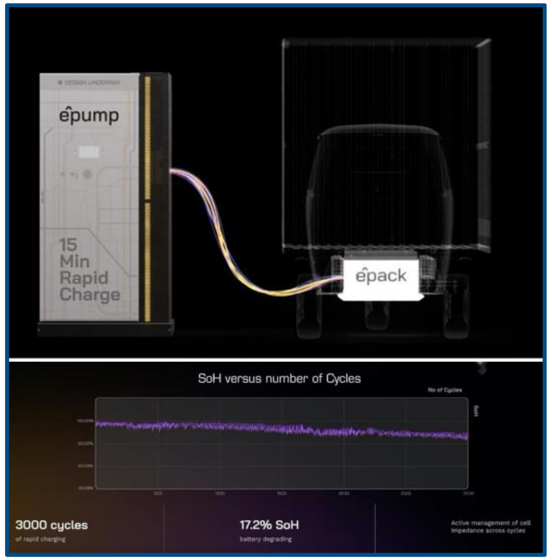


Exponent Energy Develops 15-Minute Fast-Charging Battery System for Electric Vehicles Using Model-Based Design

Using Simulink, Stateflow, and Embedded Coder, Exponent Energy designed a battery system with a fast charger that accounts for state of charge, state of health, and impedance faster and more accurately than existing systems. After 3,000 cycles, its residual capacity was still over 80%

Key Outcomes/Advantages

- Accelerated the development, testing, and verification of the entire solution by at least five times using Model-Based Design
- Accelerated product prototyping through code generation with Embedded Coder and TSP for TI C2000
- Developed an accurate and responsive BMS with improved thermal management, resulting in more than 80% state of health after 3,000 fast-charging cycles



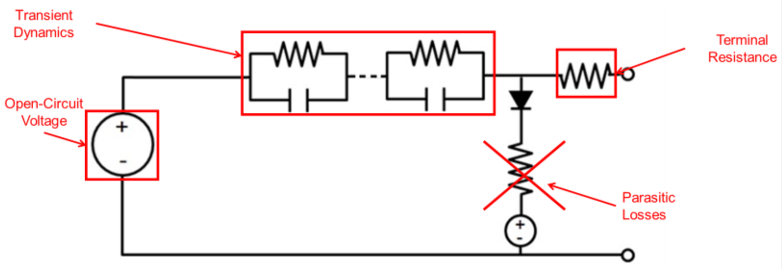
(Above) Exponent Energy's flexible energy stack with the e^pump fast charger and e-pack battery in the vehicle. (Below) A graph measuring battery health across fast charging cycles.

> *"In just a month, Exponent Energy successfully integrated the battery with the vehicle and conducted multiple charging and discharging tests on the road. We reduced development time significantly using Model-Based Design and MATLAB and Simulink products."*
>
> *- Richard Davis, lead system architect, Exponent Energy*

Link to user story

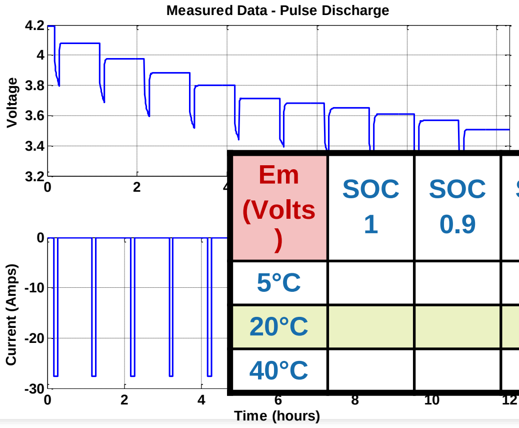
Battery Pack and BMS Design Workflow Tasks

1. Cell Modeling



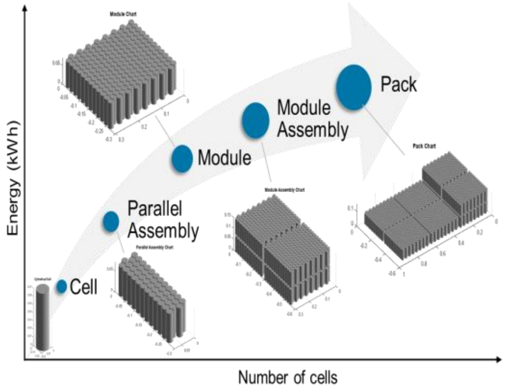
$[E_m\ R_x\ C_x] = f(SOC, Temperature\ldots)$

2. Cell Parametrization

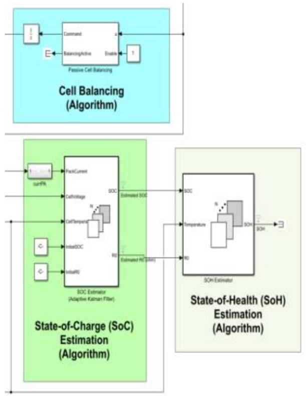


Em (Volts)	SOC 1	SOC 0.9	SOC 0.8	…	SOC 0
5°C					
20°C					
40°C					

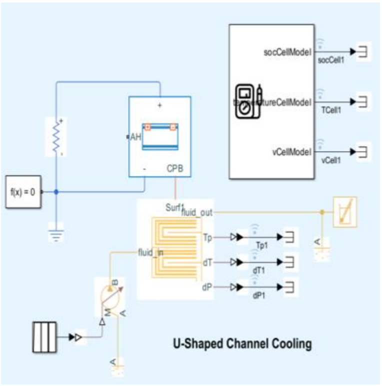
3. Battery Pack Design



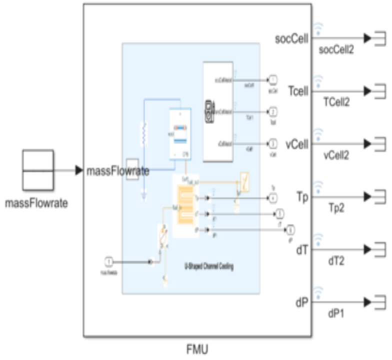
4. Battery Management System Design & Deployment



5. Thermal Management System Design



6. Deployment and HIL

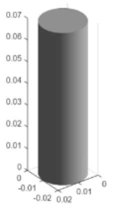


Where do we start?

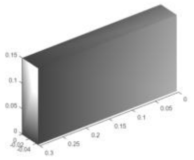
Gain insight into cell behavior and model it

Cell Geometry

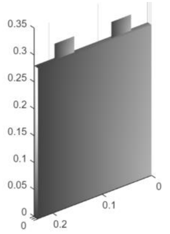
Available geometries



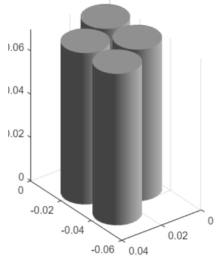
Cylindrical Cell



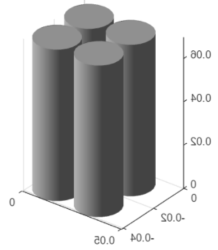
Prismatic Cell



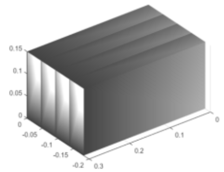
Pouch Cell



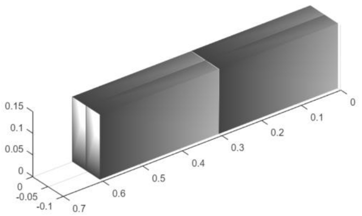
Hexagonal



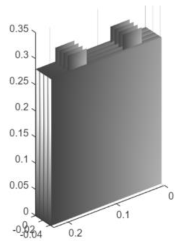
Square



Single Stack



Multiple Stack



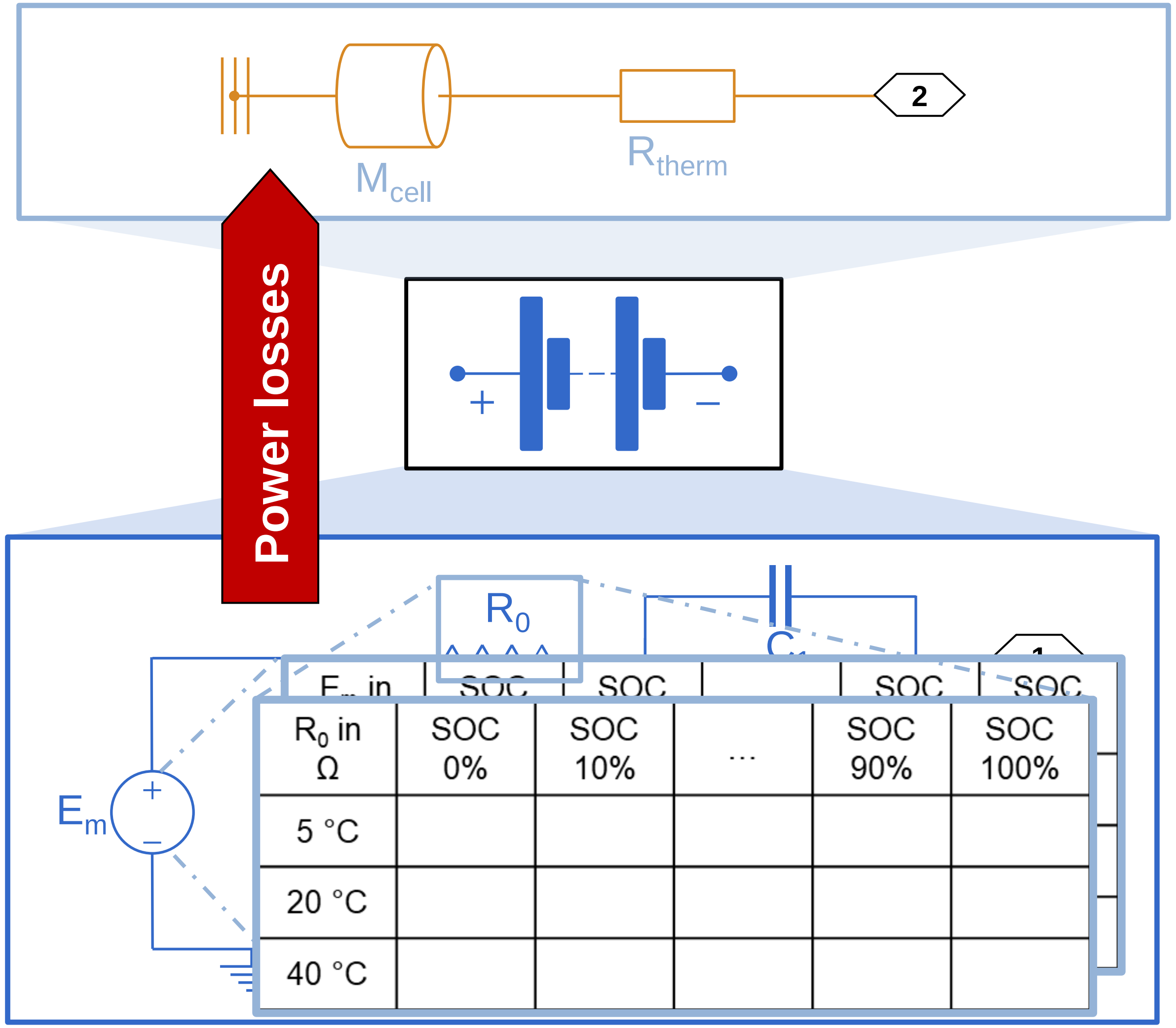
Single Stack

- The cells can be disposed in different topologies, depending on their geometry
- Battery pack volume, mass, and dimensions are scaled automatically as new cells are added

Understanding the Cell Model

Electrical and thermal model

- Electrical model for RC circuit
  - Look-up table based
  - Dependency from SOC and Temperature
- Thermal lumped model
- Thermal resistances that account for different thermal paths
- Power losses calculated from Ohmic losses

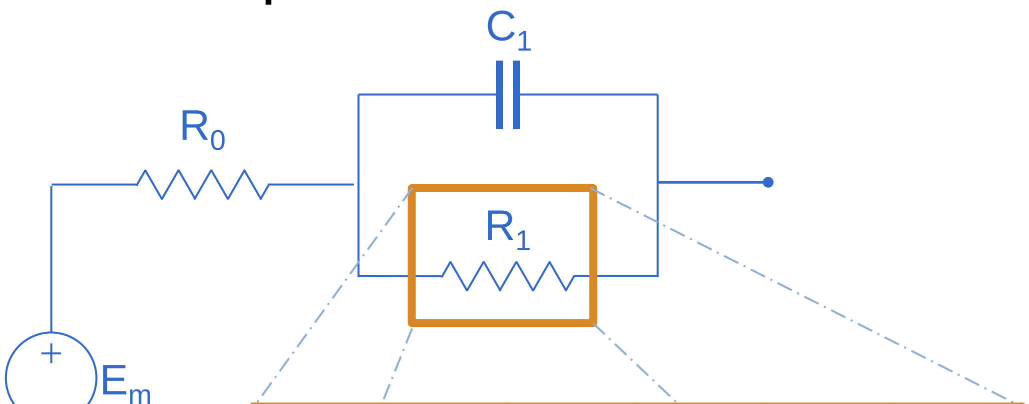


Parametrizing the Cell Model

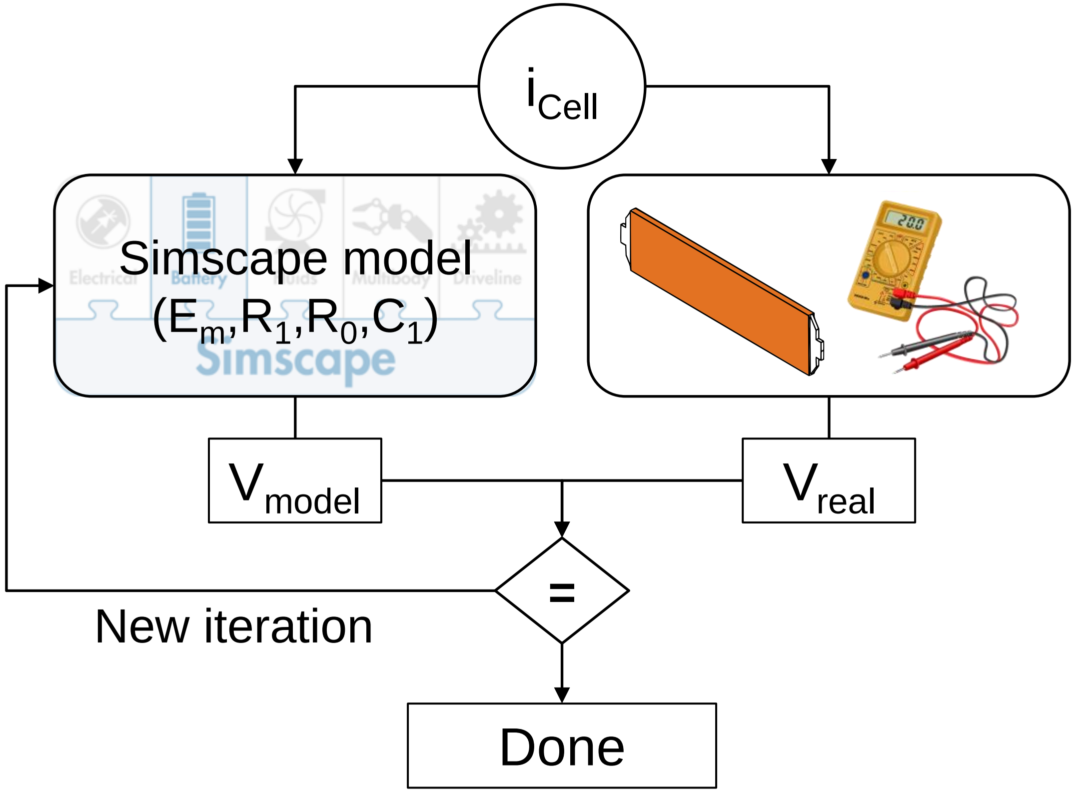
RC circuit parametrization

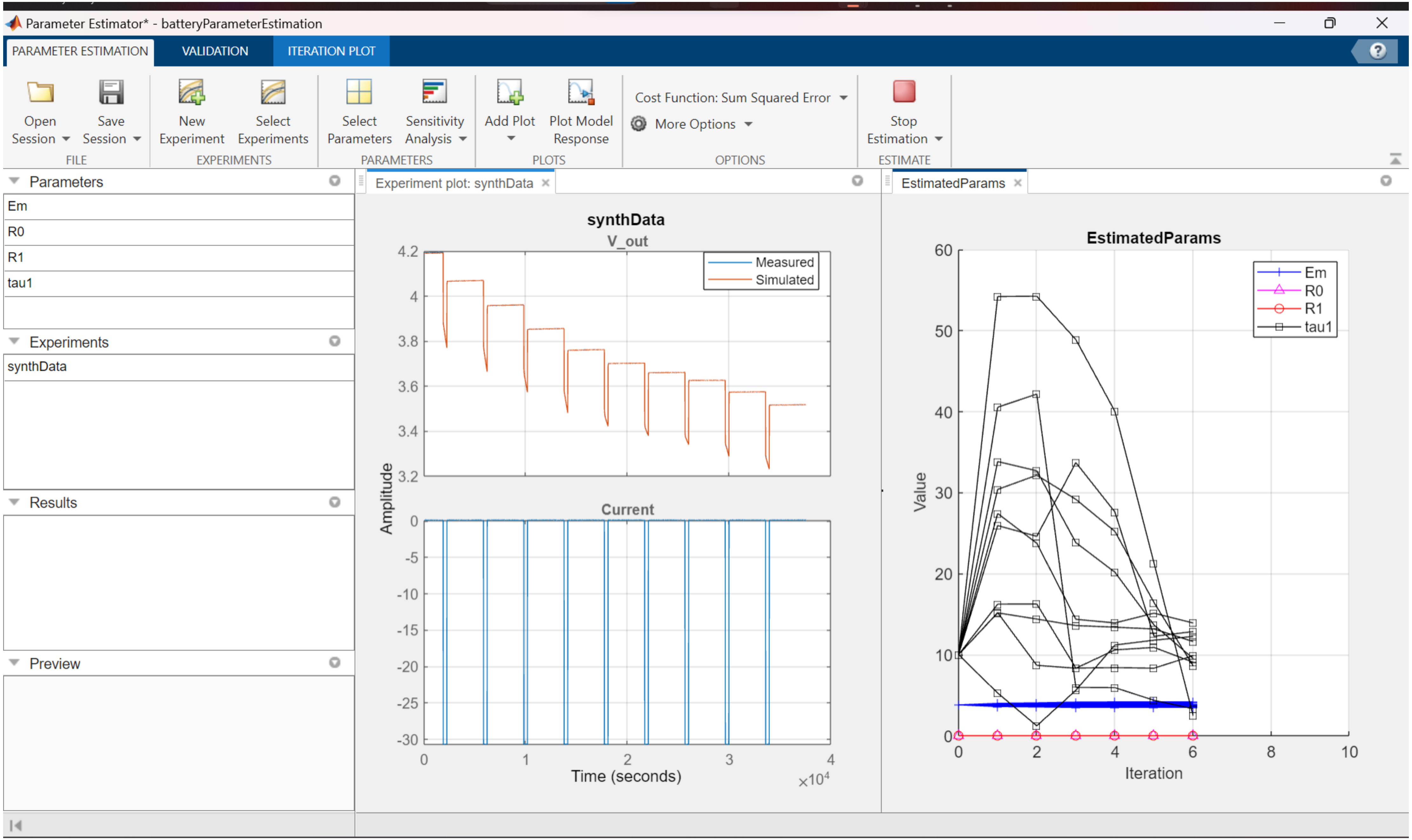
Two possible cases for RC circuit (E_m, R_0, R_1…) parametrization:

- Look-up tables are known
- Look-up tables must be estimated

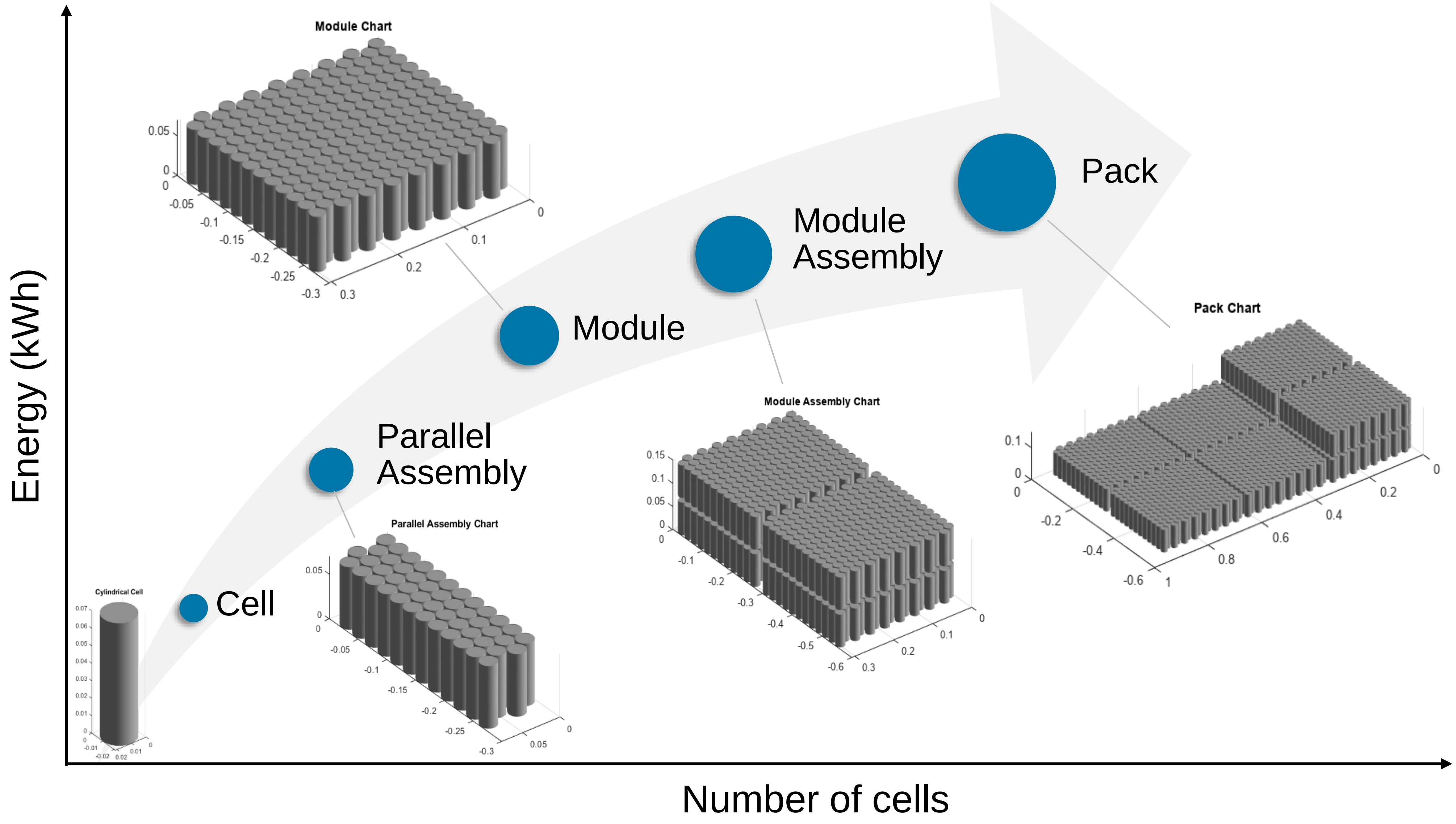


R1 in Ω	SOC 100%	SOC 90%	SOC 80%	…	SOC 0%
5°C					
20°C					
40°C					



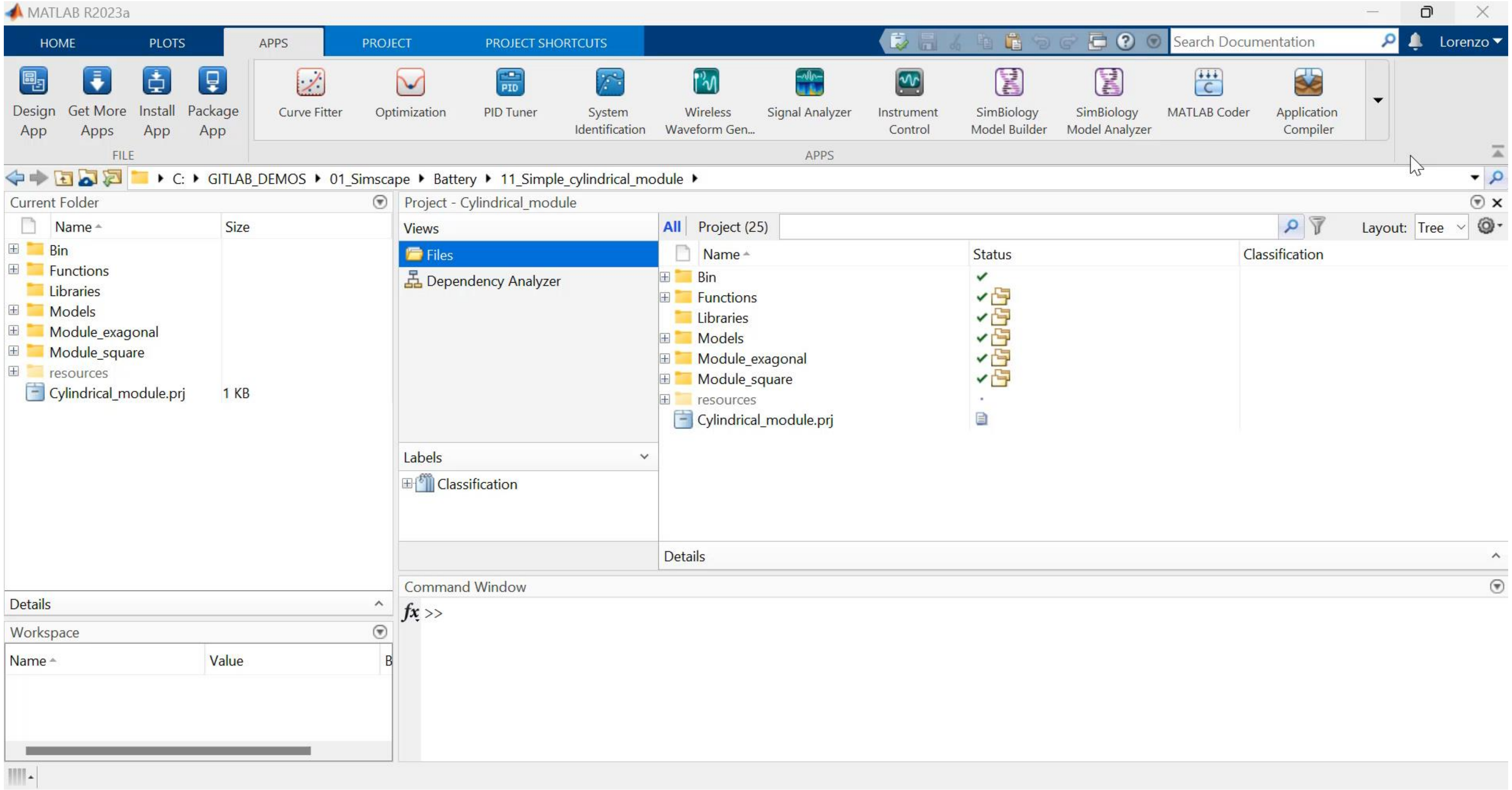

Parameter Estimator* - batteryParameterEstimation
PARAMETER ESTIMATION
VALIDATION
ITERATION PLOT
Open Session
Save Session
FILE
New Experiment
Select Experiments
EXPERIMENTS
Select Parameters
Sensitivity Analysis
PARAMETERS
Add Plot
Plot Model Response
PLOTS
Cost Function: Sum Squared Error
More Options
OPTIONS
Stop Estimation
ESTIMATE
Parameters
Em
R0
R1
tau1
Experiments
synthData
Results
Preview
Experiment plot: synthData
synthData
V_out
Measured
Simulated
Current
Amplitude
Time (seconds)
EstimatedParams
Em
R0
R1
tau1
Value
Iteration

Electric Cell to Battery Pack



Battery Builder

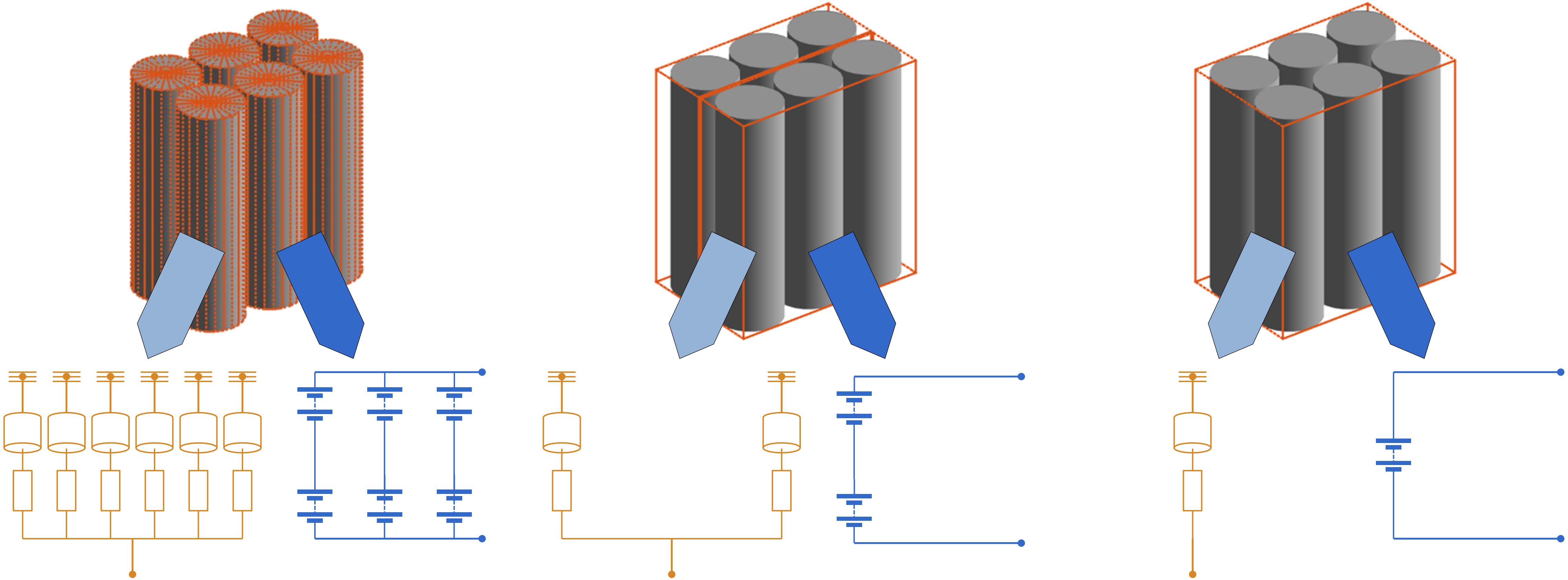
App-based battery pack generation



Choose Model Fidelity

Find tradeoff between calculation speed and precision

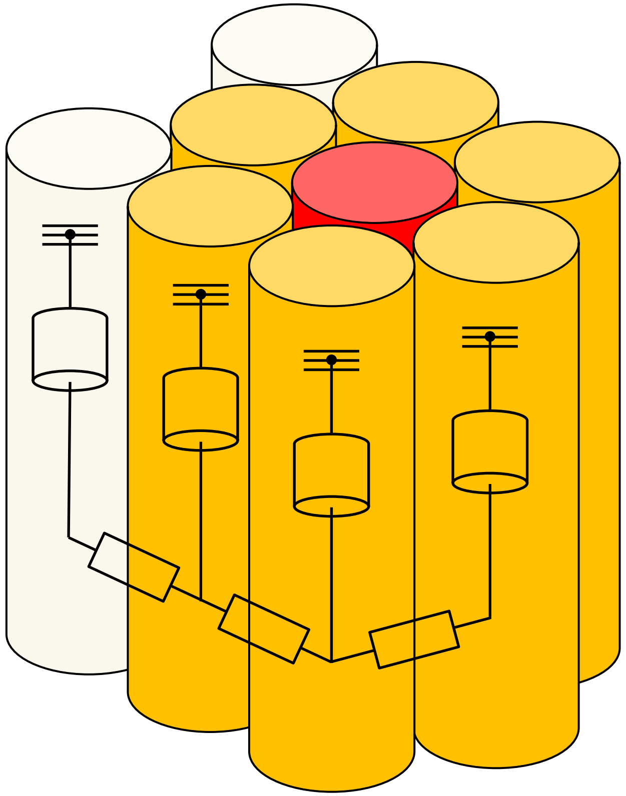
Detailed | **Grouped** | **Lumped**



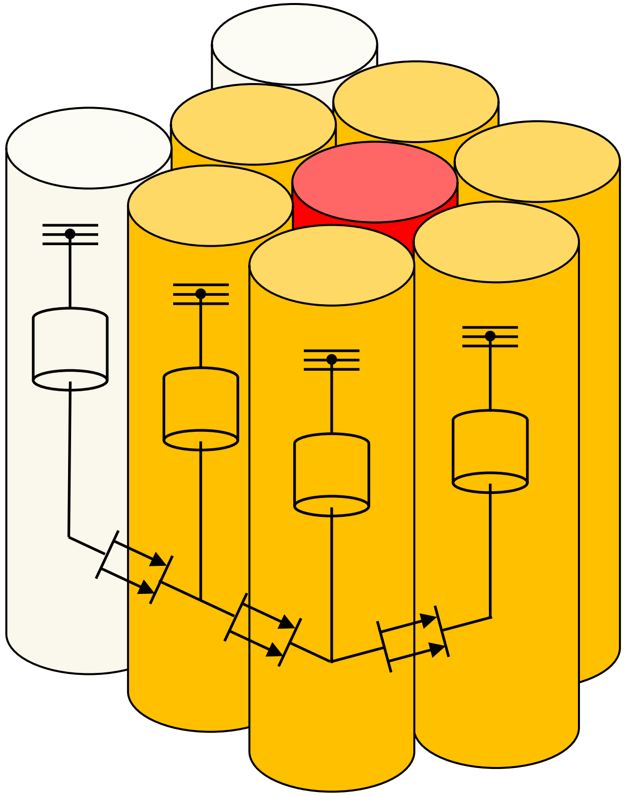
Battery pack: Thermal Paths WITHIN the Pack

Model conduction, convection, and radiation between cells

Inter cell path



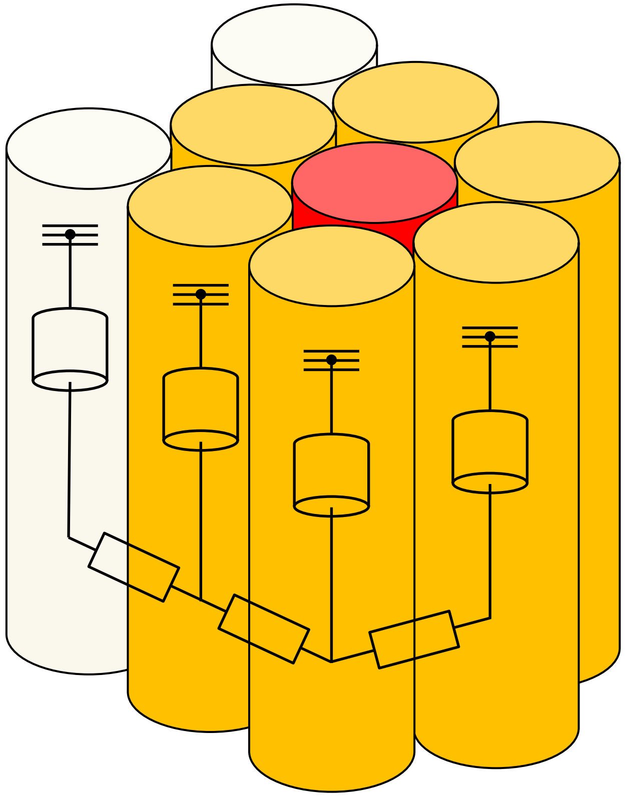
Inter cell radiative path

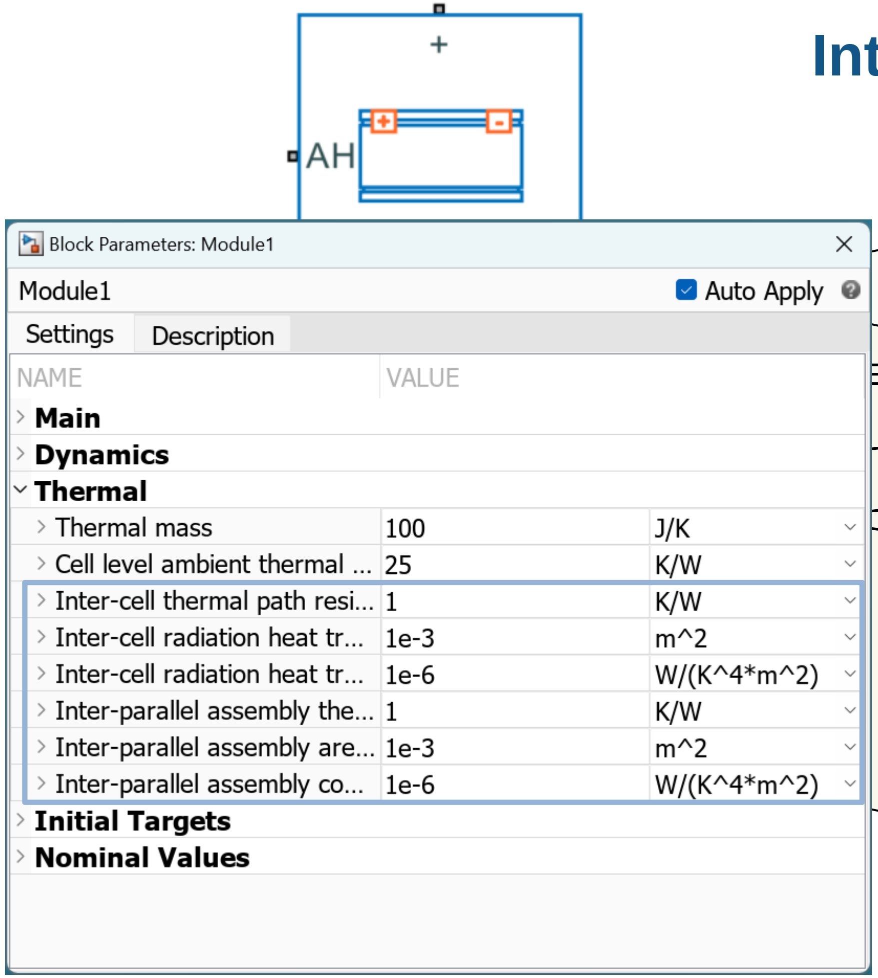


Battery pack: Thermal Paths WITHIN the Pack

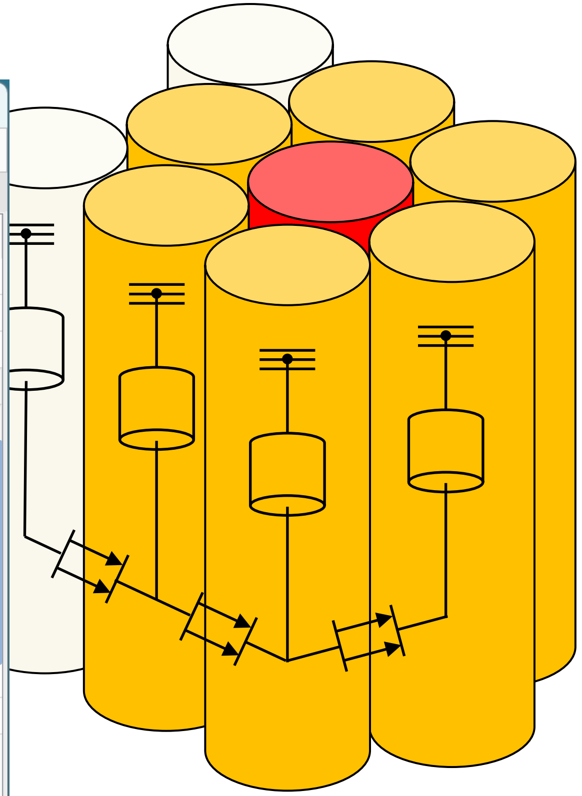
Model conduction, convection, and radiation between cells

Inter cell path





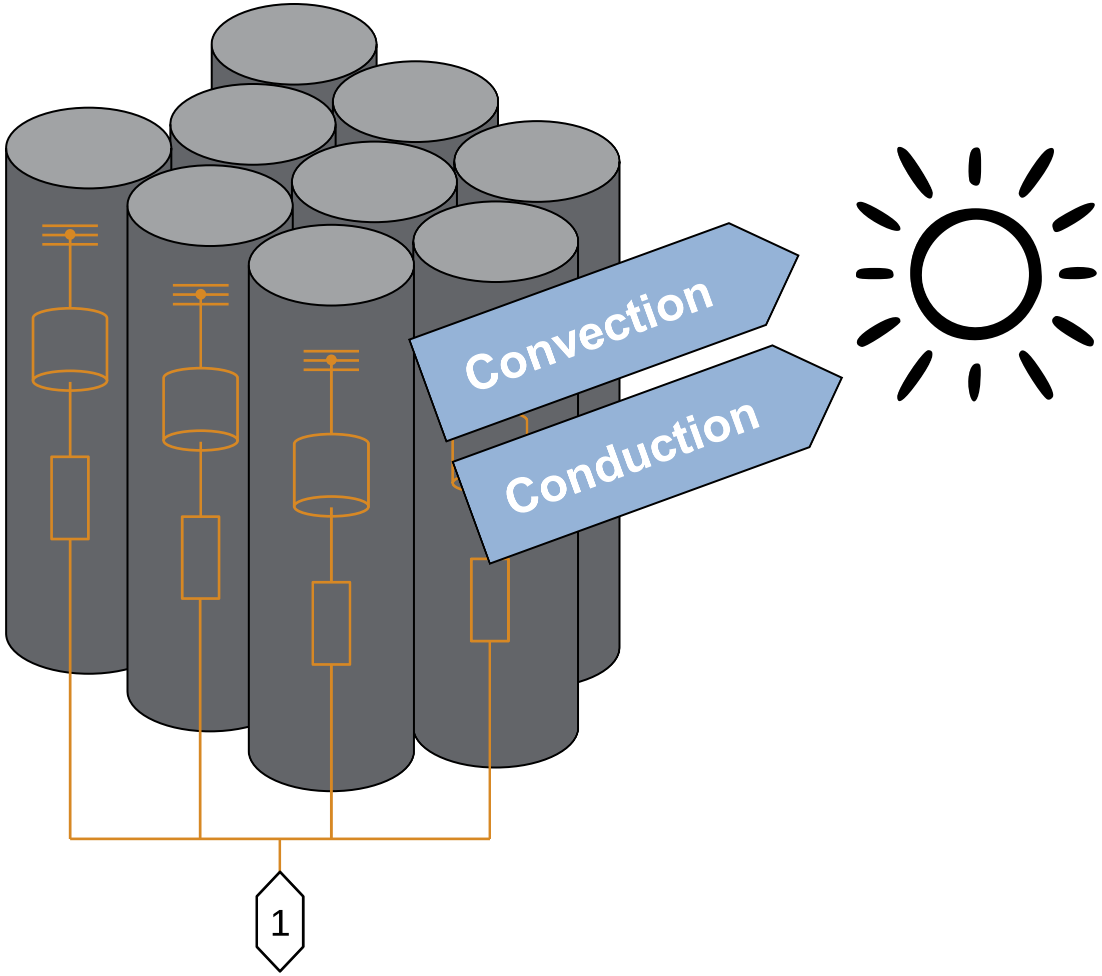
Inter cell radiative path



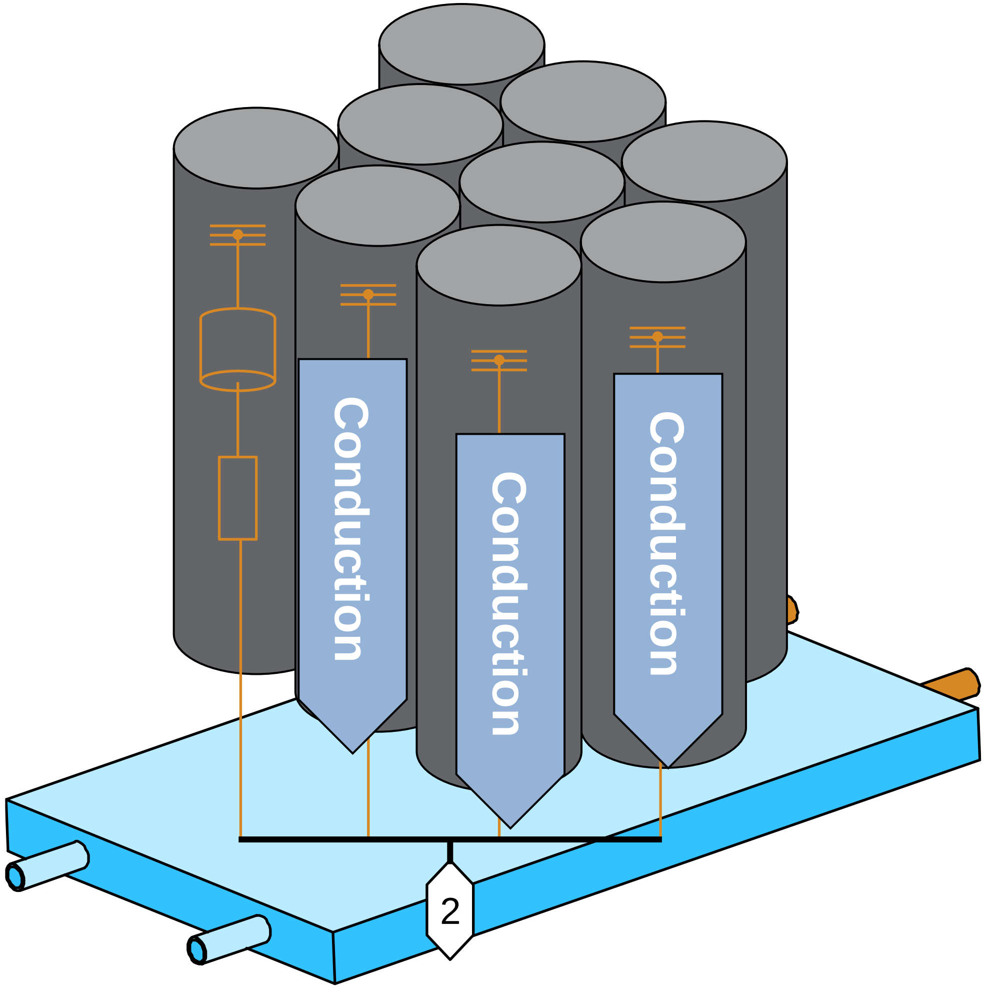
Battery pack: Thermal Paths OUTSIDE of the Pack

Model conduction, convection between pack and environment

Thermal path to ambient



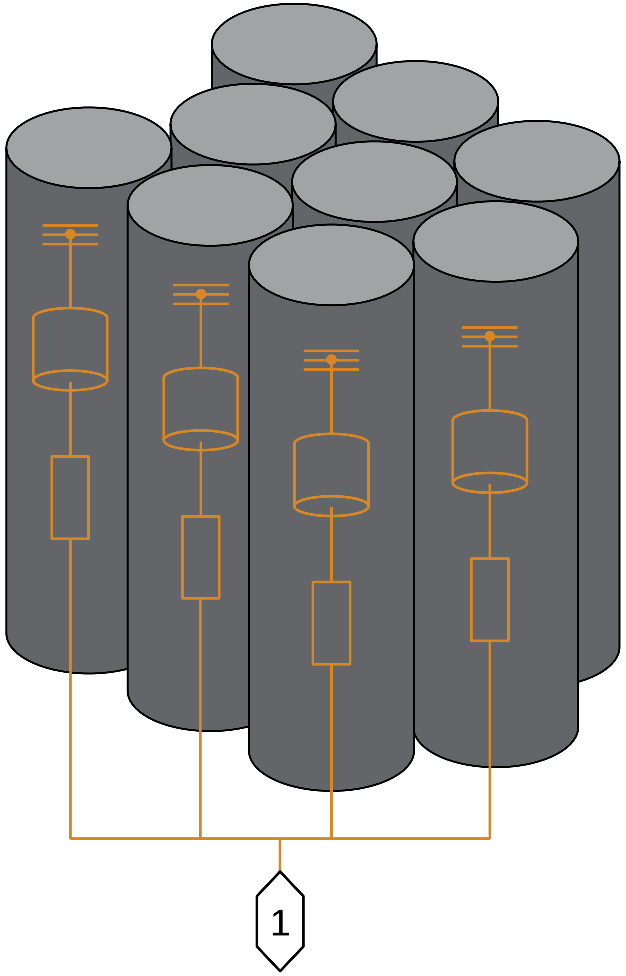
Thermal path to cooling plate

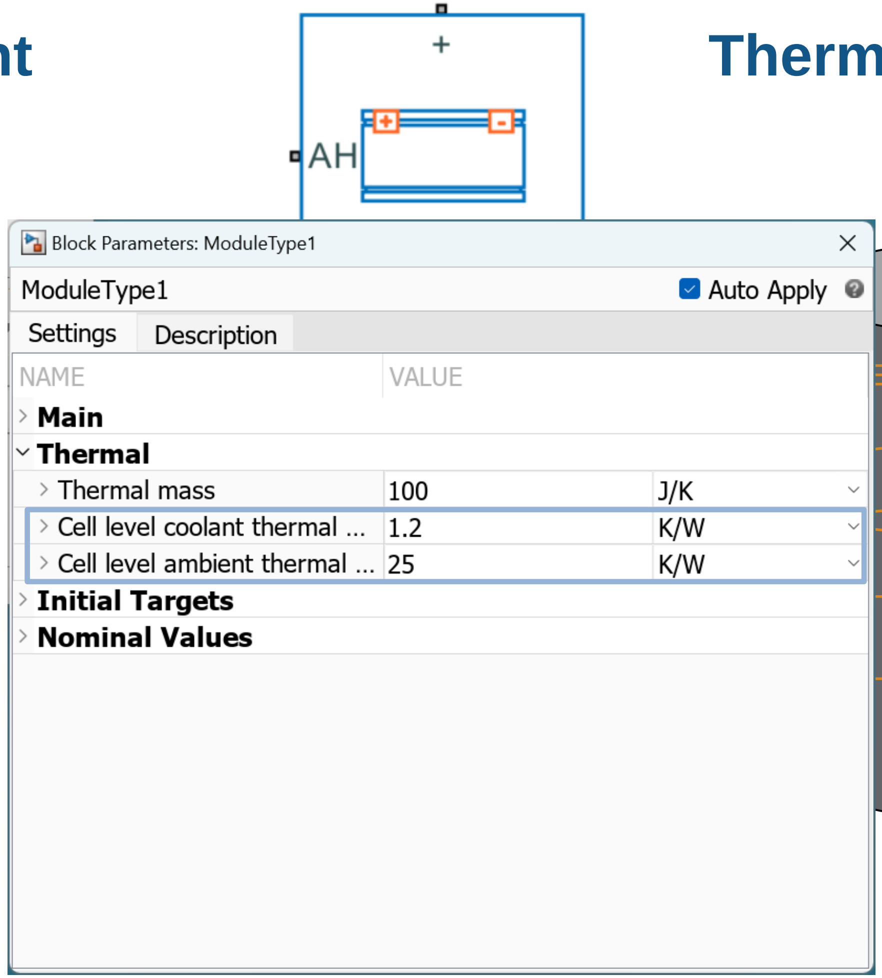


Battery pack: Thermal Paths OUTSIDE of the Pack

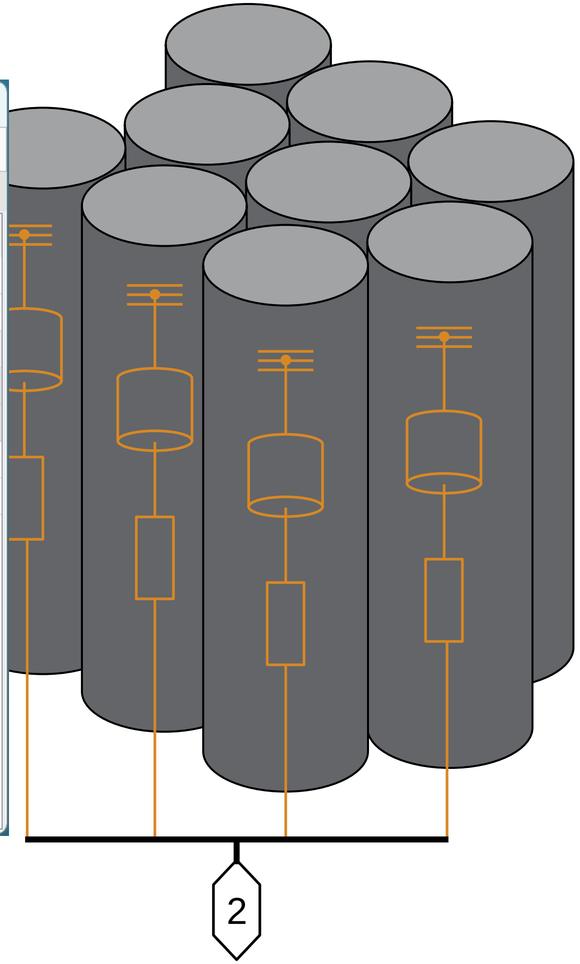
Model conduction, convection between pack and environment

Thermal path to ambient



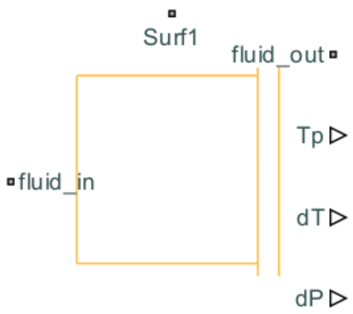


Thermal path to cooling plate

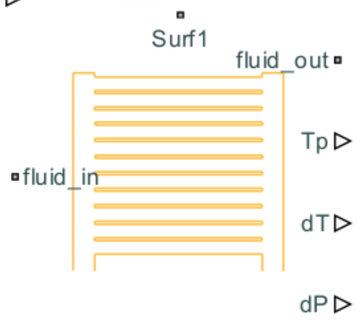


Battery pack active cooling
Cooling plate topologies

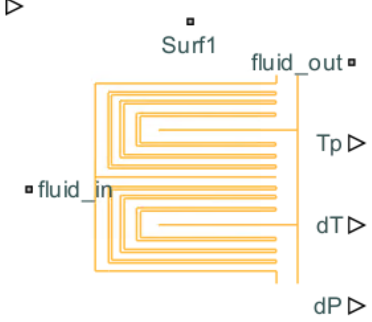
- Model heat transfer between battery, liquid cooling system, and environment
  - Control cell-to-cell temperature variation
  - Tradeoff of pumping costs and cooling efficiency
- Different cooling plate topologies
  - Edge, parallel channel, U-shaped channel
  - Single- and double-sided plates
- Adjust resolution of thermal model
  - Define quantity and placement of nodes



Edge Cooling



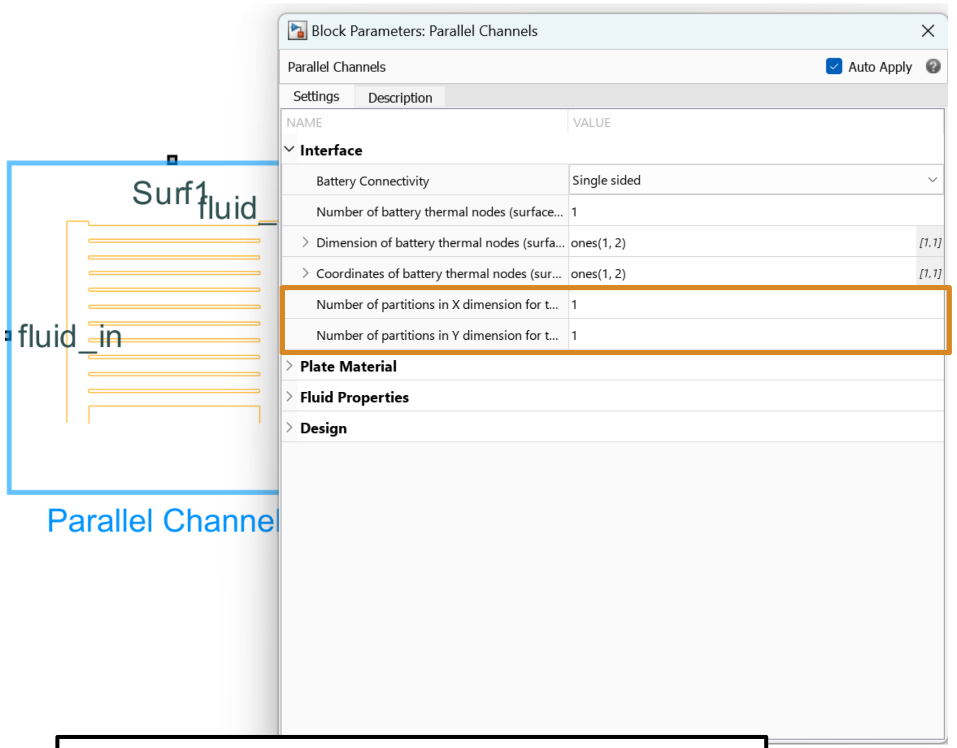
Parallel Channels



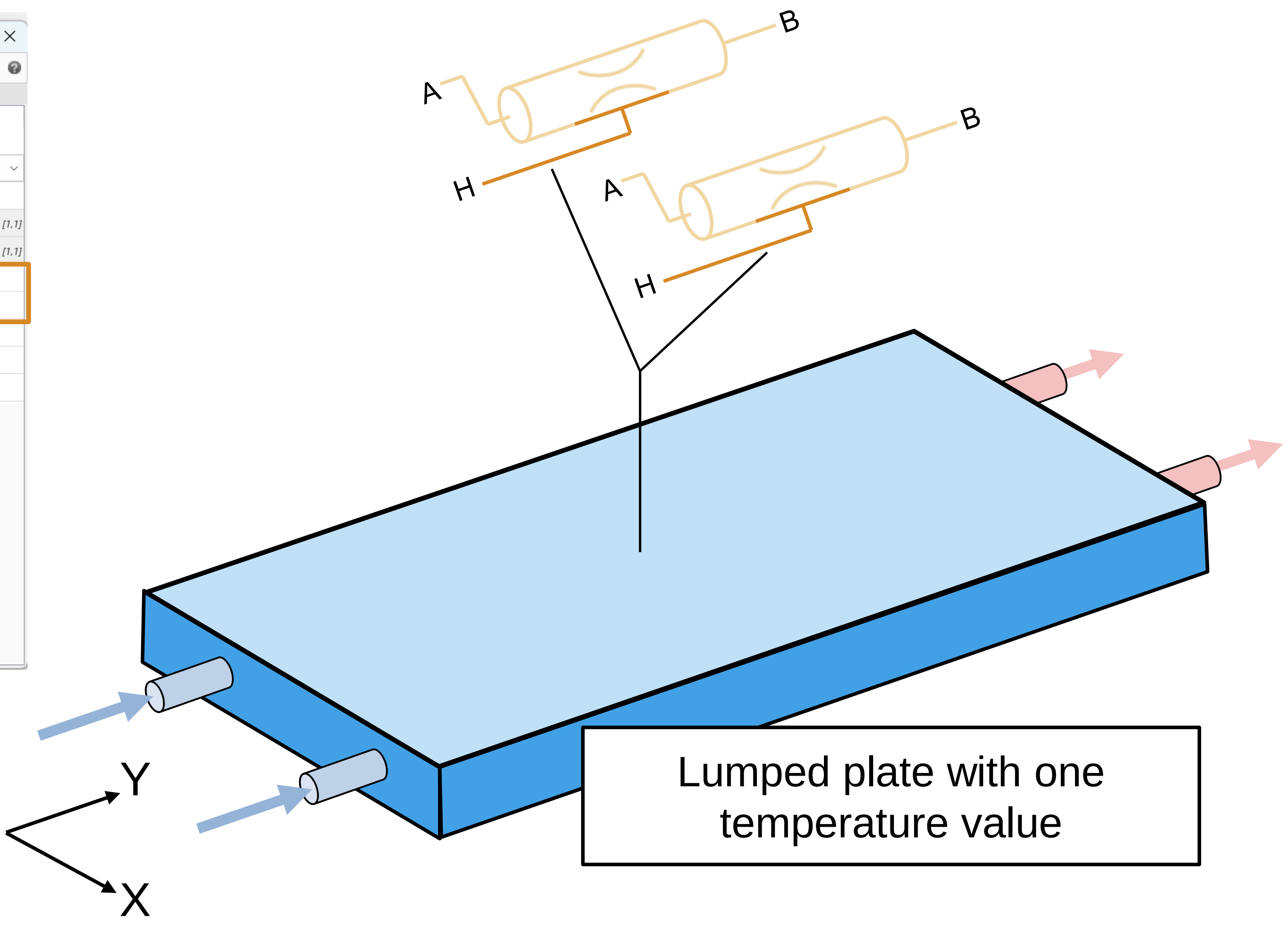
U-shaped Channels

Cooling plate model

Parallel Channels plate with two channels, lumped



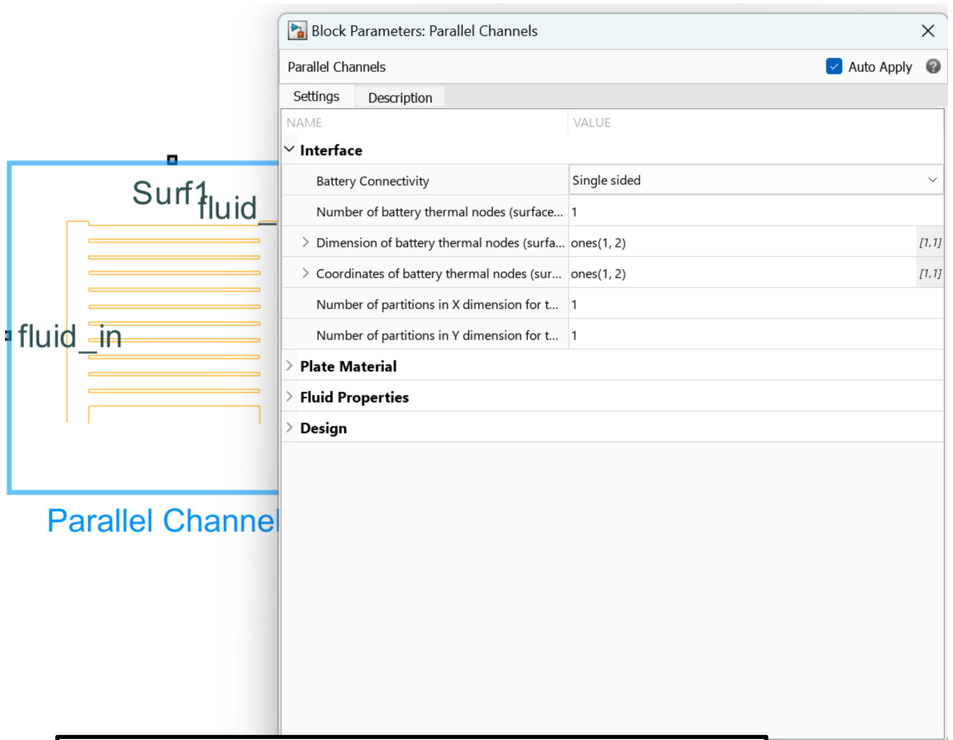
Number of partitions in X: 1
Number of partitions in Y: 1



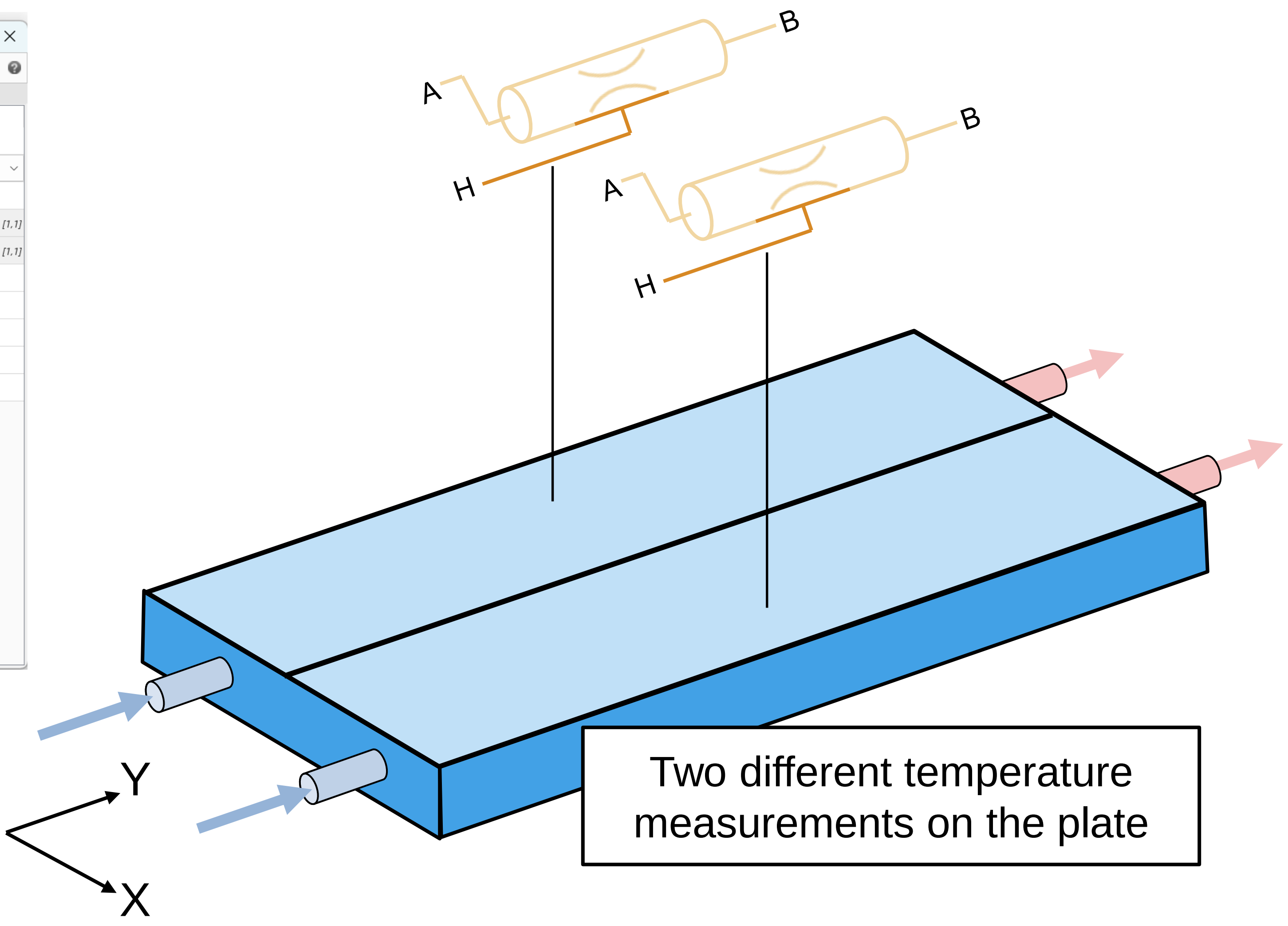
Lumped plate with one temperature value

Cooling plate model

Parallel Channels plate with two channels, two plate elements



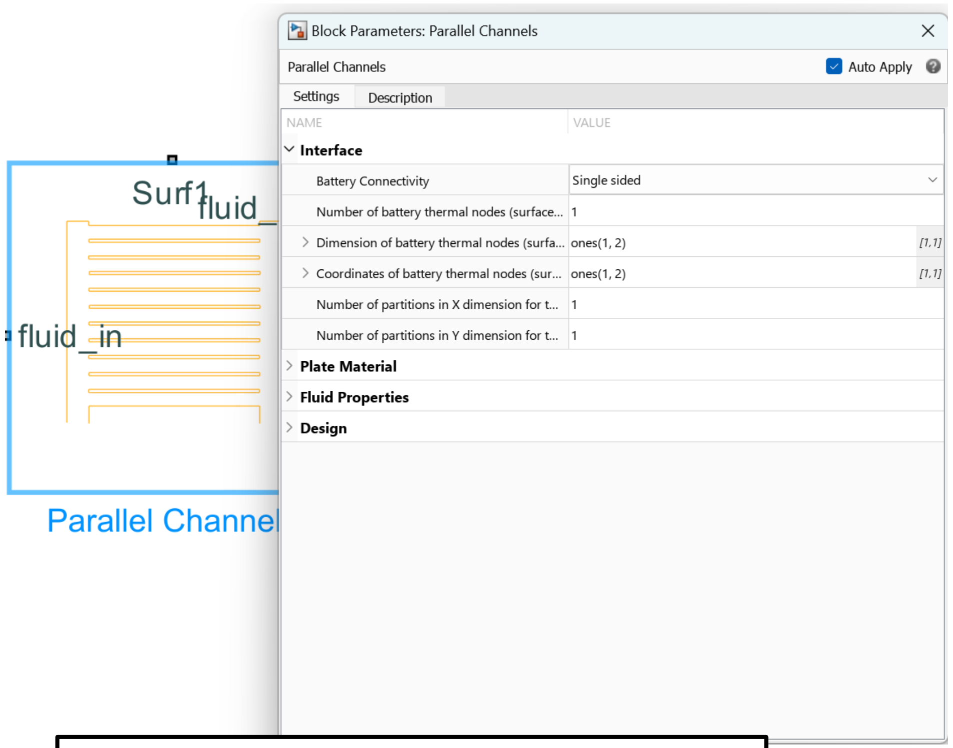
Number of partitions in X: 2
Number of partitions in Y: 1



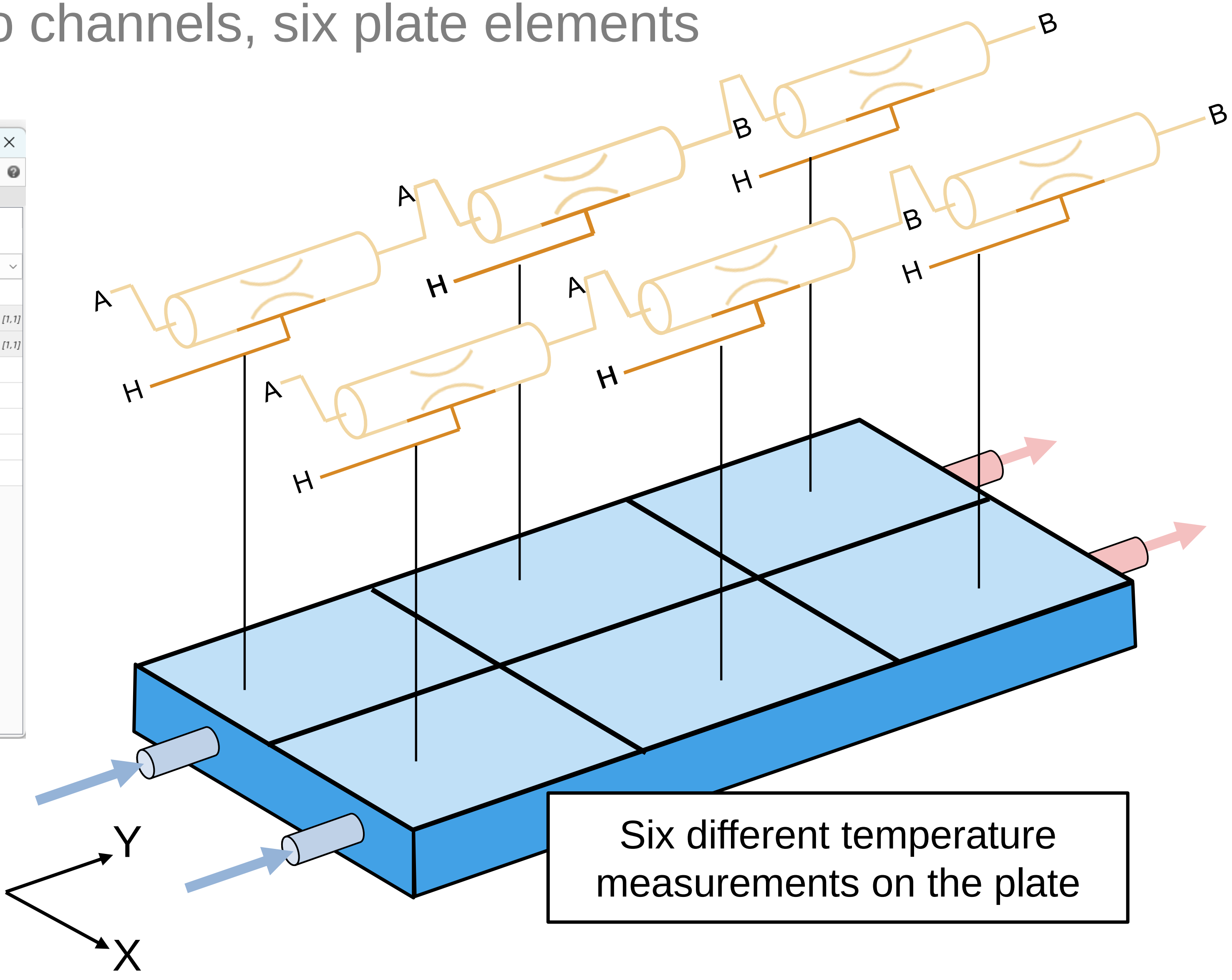
Two different temperature measurements on the plate

Cooling plate model

Parallel Channels plate with two channels, six plate elements



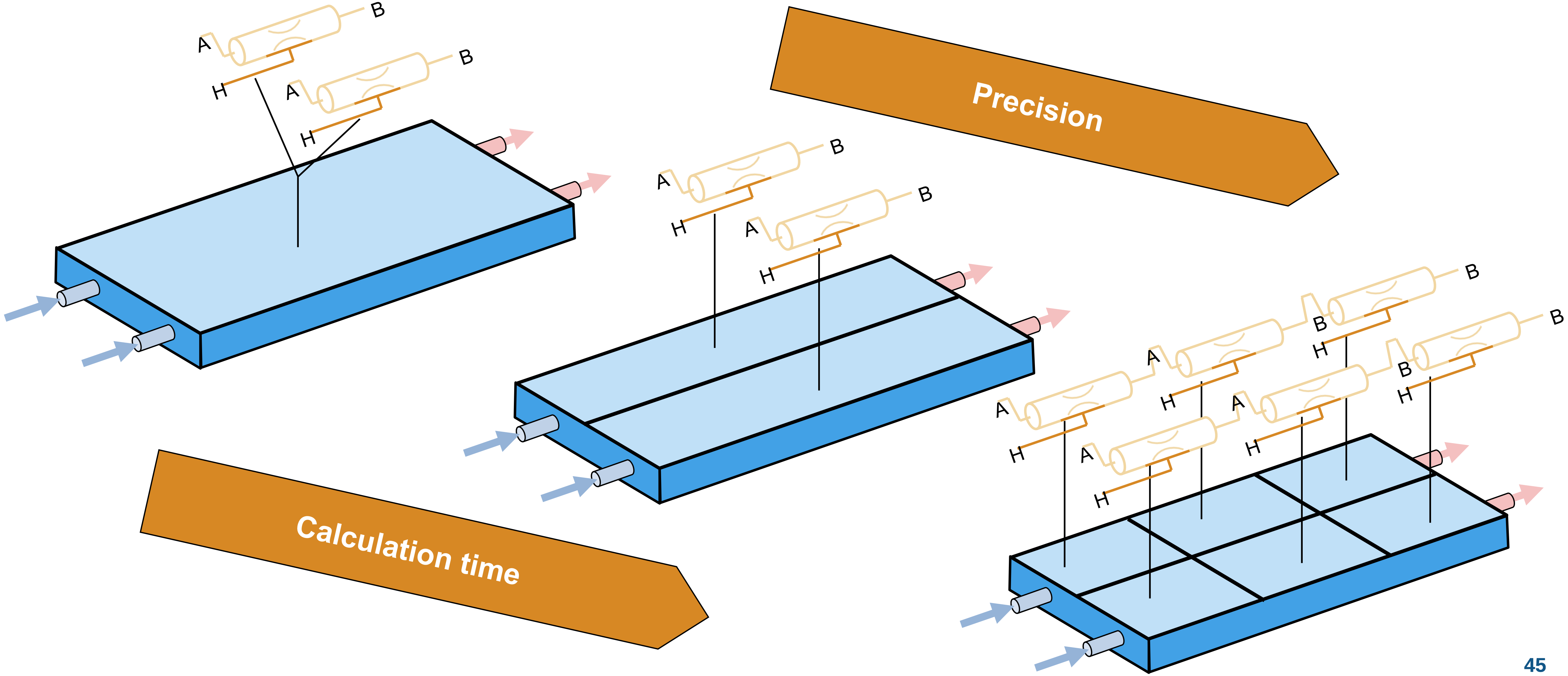
Number of partitions in X: 2
Number of partitions in Y: 3



Six different temperature measurements on the plate

Cooling plate model

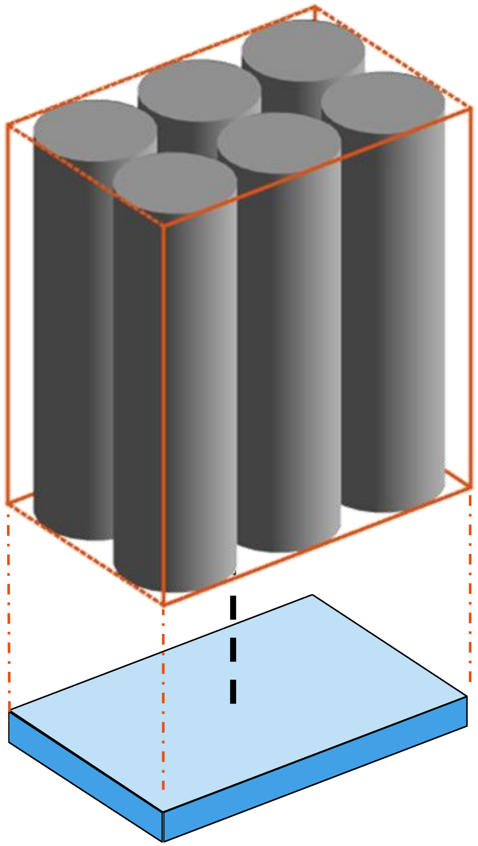
Find tradeoff between calculation speed and precision



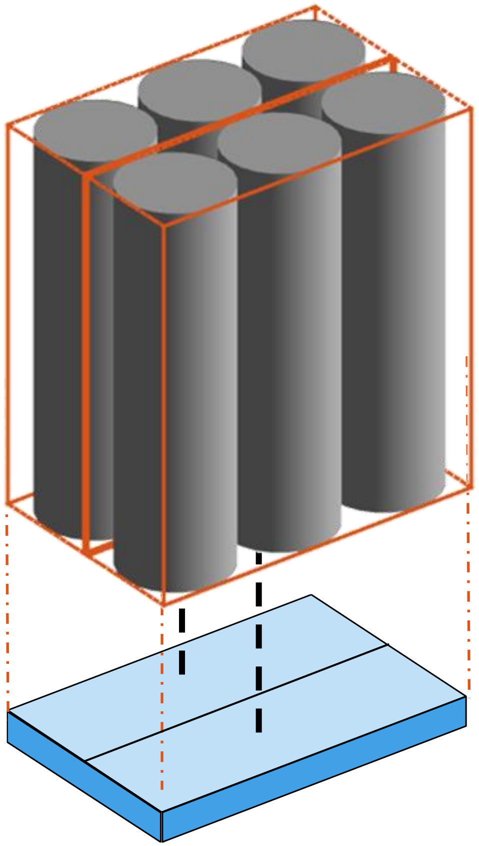
Connecting Pack and Plate

Combine plate and pack module fidelities

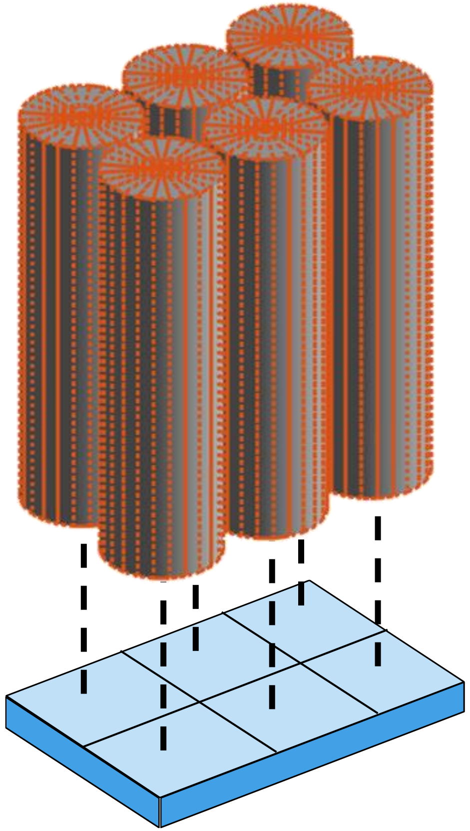
Lumped module
Lumped plate



Grouped module
Discretized plate



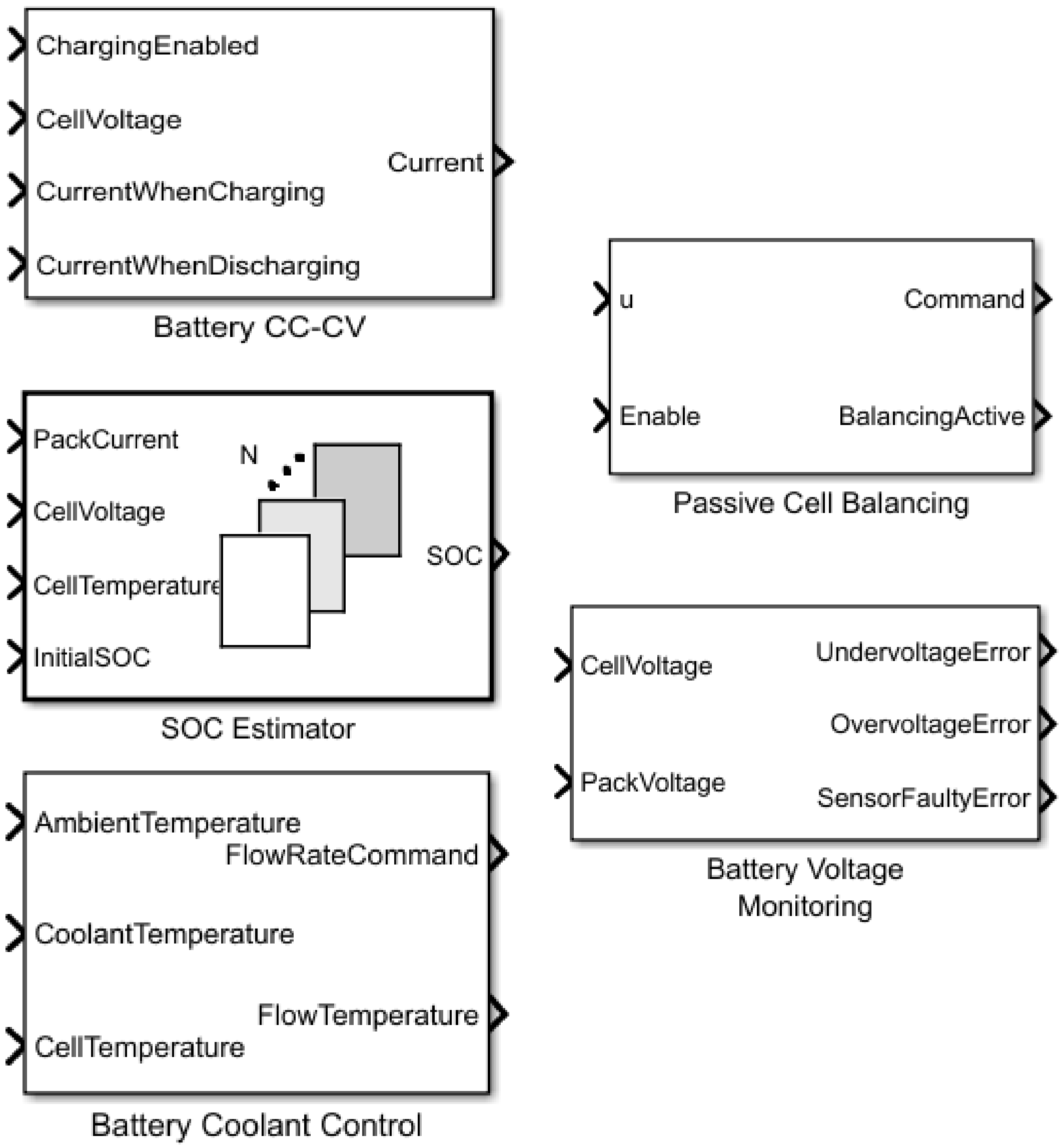
Detailed module
Discretized plate



Battery Management Algorithms

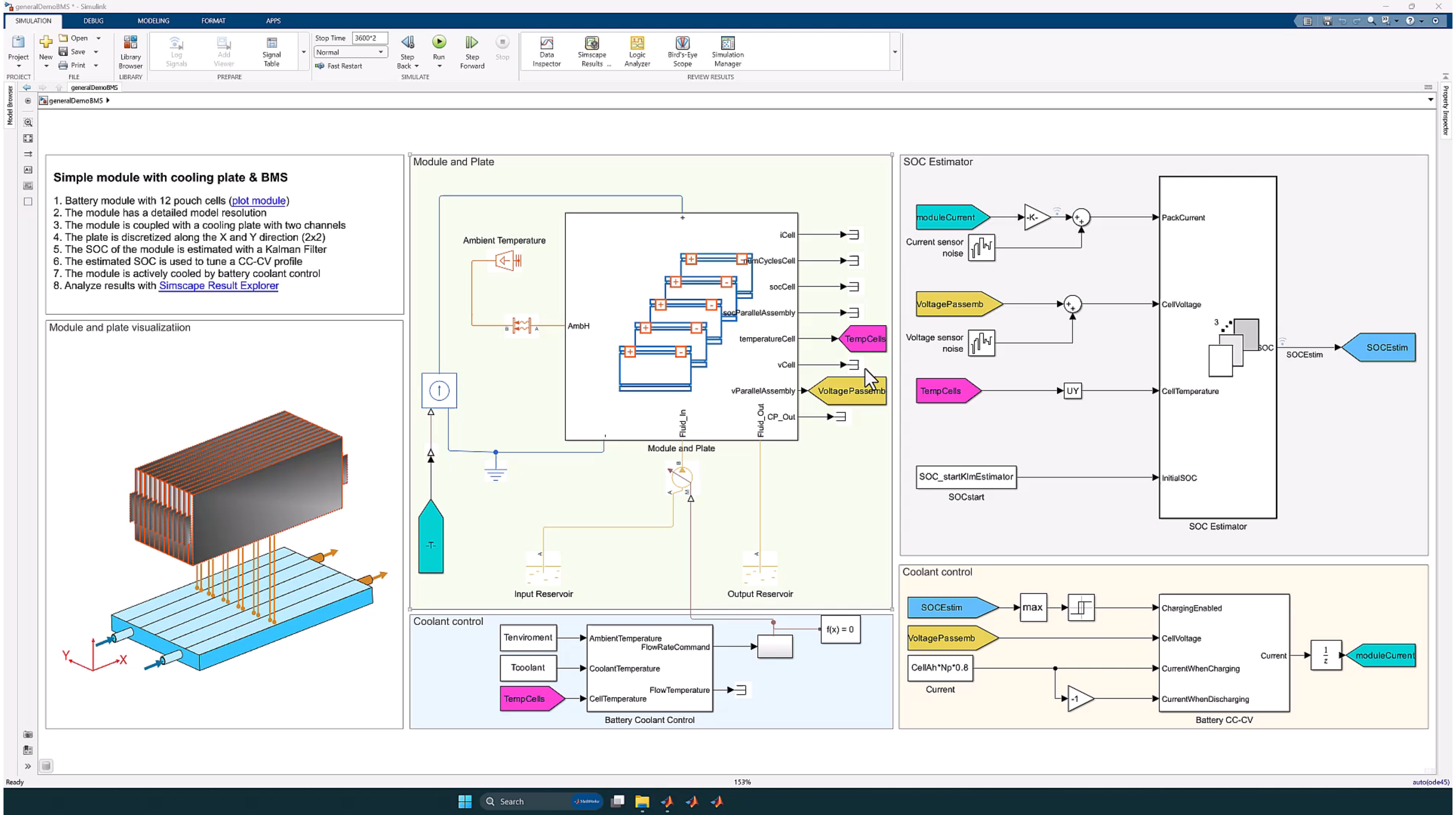
Block overview

- Charge and discharge
  - CC-CV, current limits
- Passive cell balancing
- Estimators
  - SOC, SOH
- Protection
  - Current, voltage, temperature monitor
  - Fault qualification
- Thermal management
  - Coolant and heater control



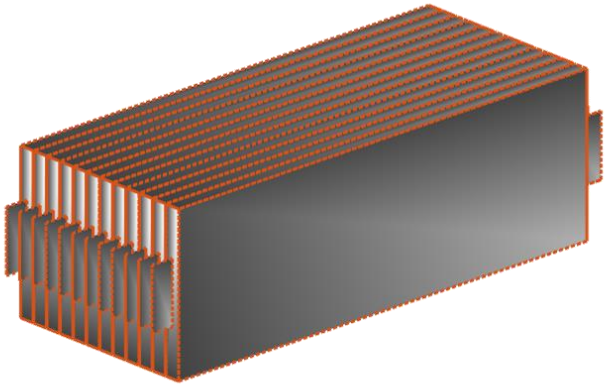
Pouch Module with Cooling Plate & BMS Blocks

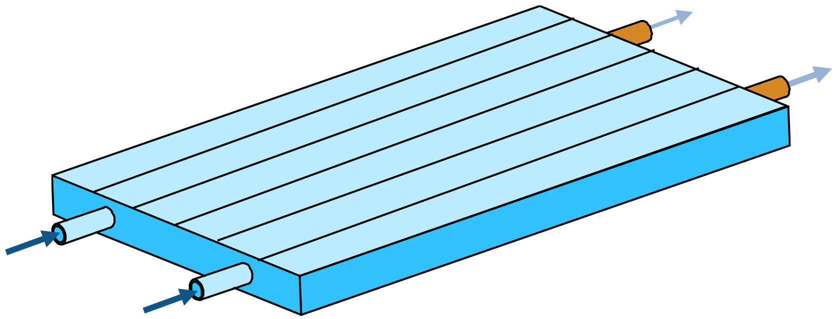
Demo implementation



Pouch Module with Cooling Plate & BMS Blocks

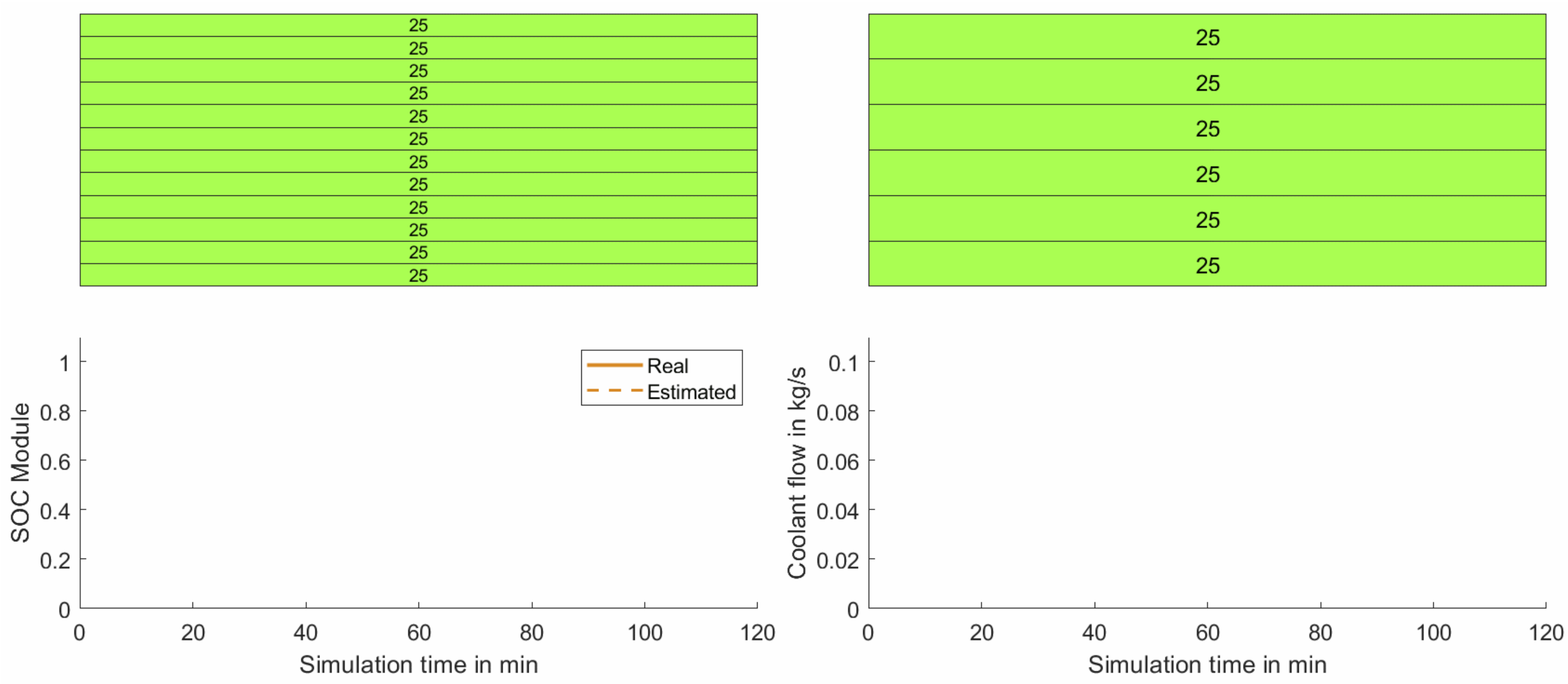
Simulation results

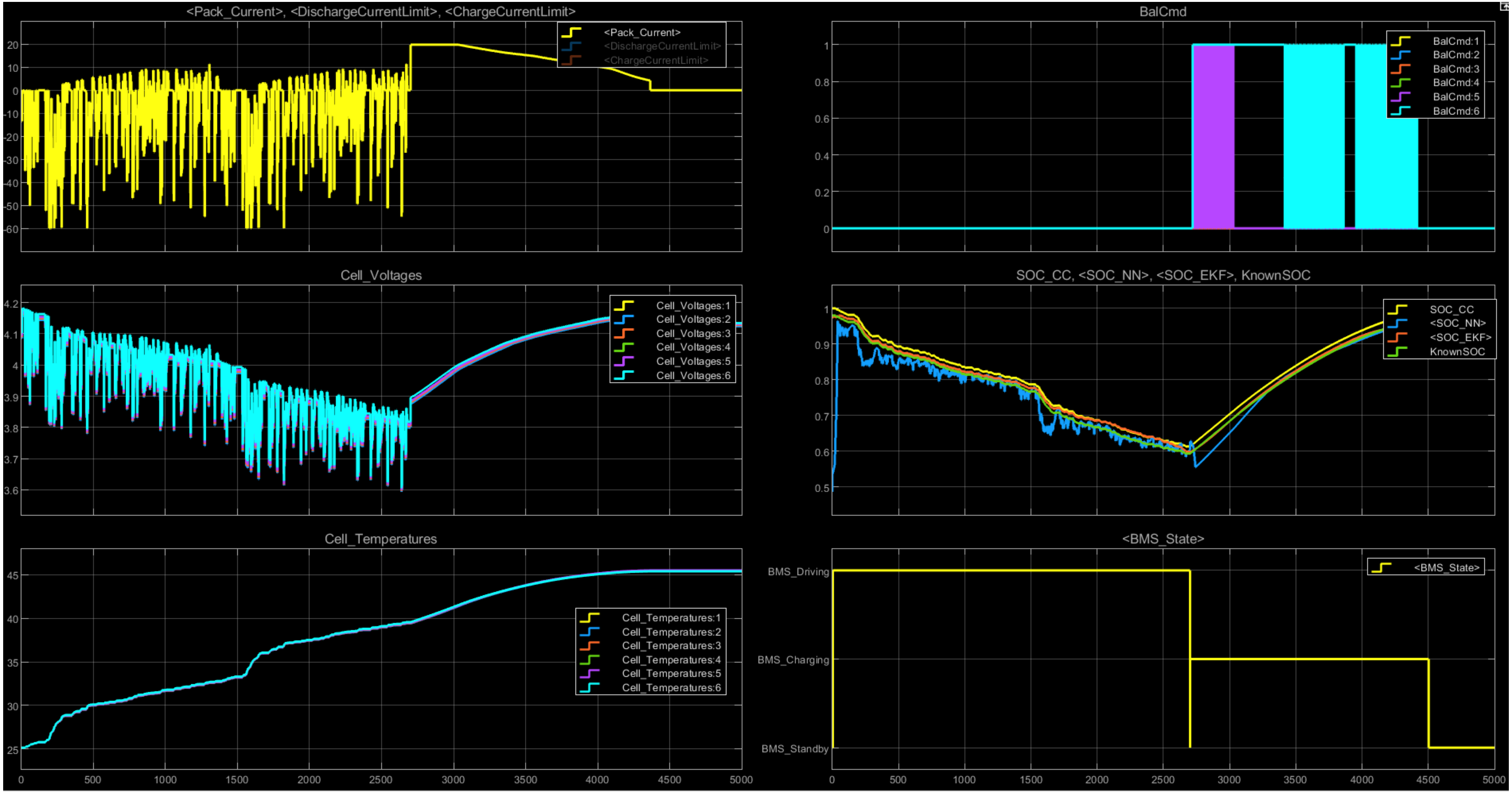




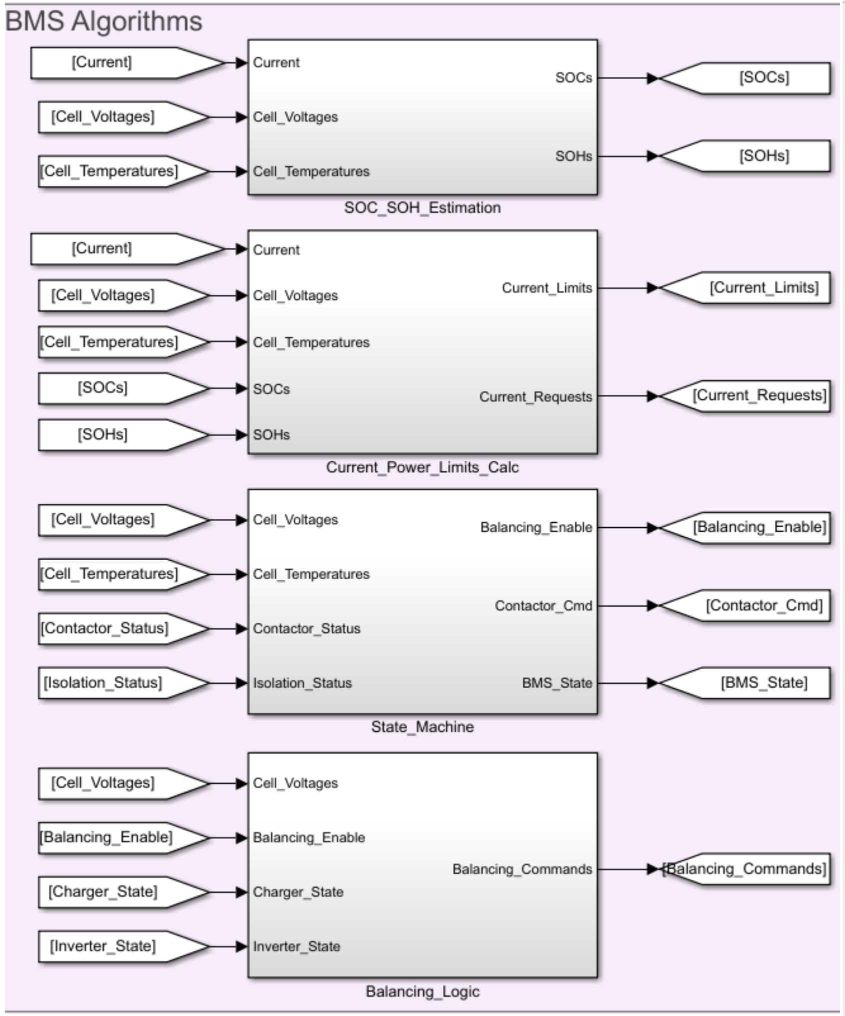
Pouch Module with Cooling Plate & BMS Blocks

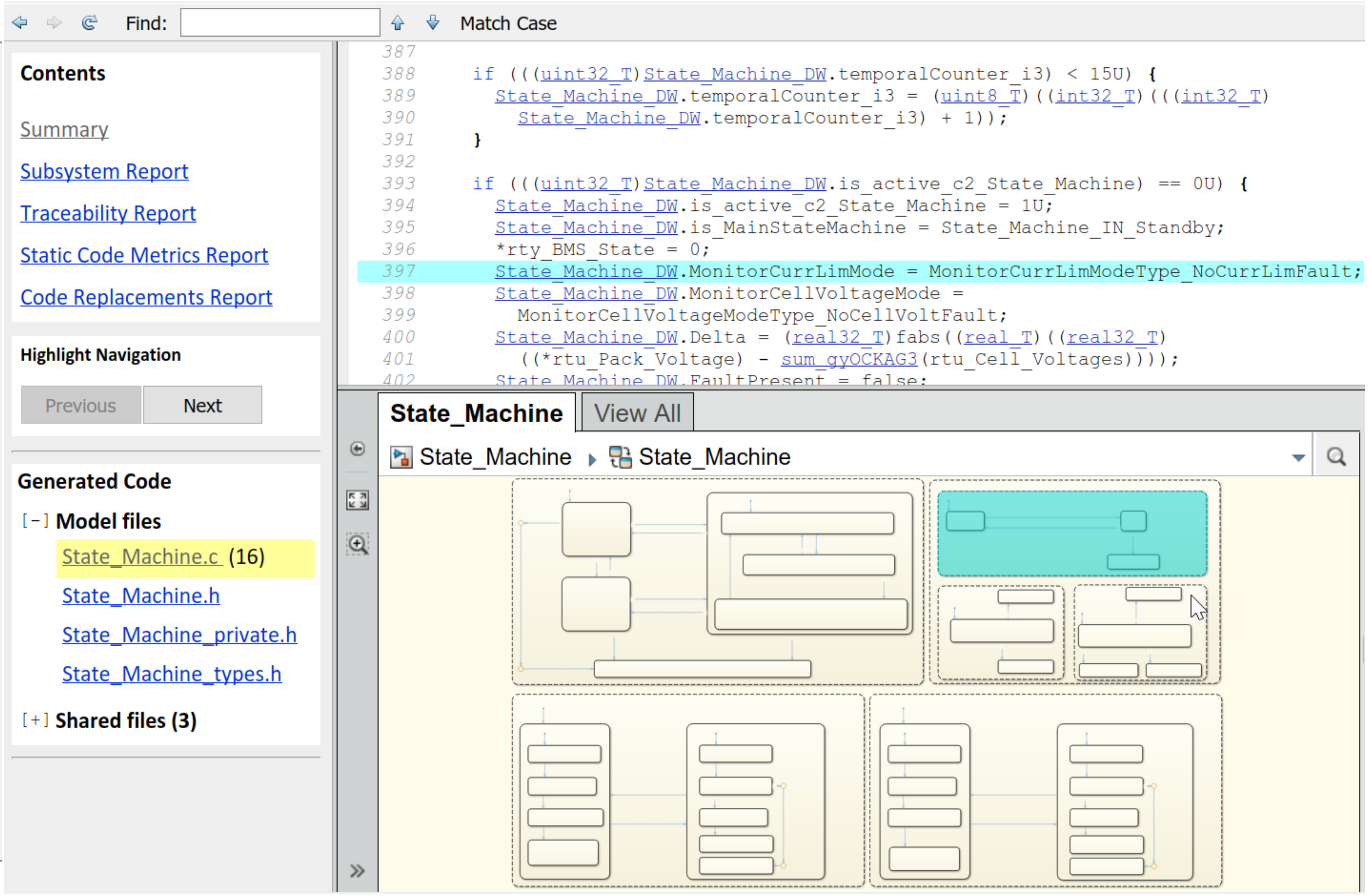
Simulation results




<Pack_Current>, <DischargeCurrentLimit>, <ChargeCurrentLimit>
<Pack_Current>
<DischargeCurrentLimit>
<ChargeCurrentLimit>
BalCmd
BalCmd:1
BalCmd:2
BalCmd:3
BalCmd:4
BalCmd:5
BalCmd:6
Cell_Voltages
Cell_Voltages:1
Cell_Voltages:2
Cell_Voltages:3
Cell_Voltages:4
Cell_Voltages:5
Cell_Voltages:6
SOC_CC, <SOC_NN>, <SOC_EKF>, KnownSOC
SOC_CC
<SOC_NN>
<SOC_EKF>
KnownSOC
Cell_Temperatures
Cell_Temperatures:1
Cell_Temperatures:2
Cell_Temperatures:3
Cell_Temperatures:4
Cell_Temperatures:5
Cell_Temperatures:6
<BMS_State>
BMS_Driving
BMS_Charging
BMS_Standby
0 500 1000 1500 2000 2500 3000 3500 4000 4500 5000

Generate C/C++ Code From BMS Algorithm Models

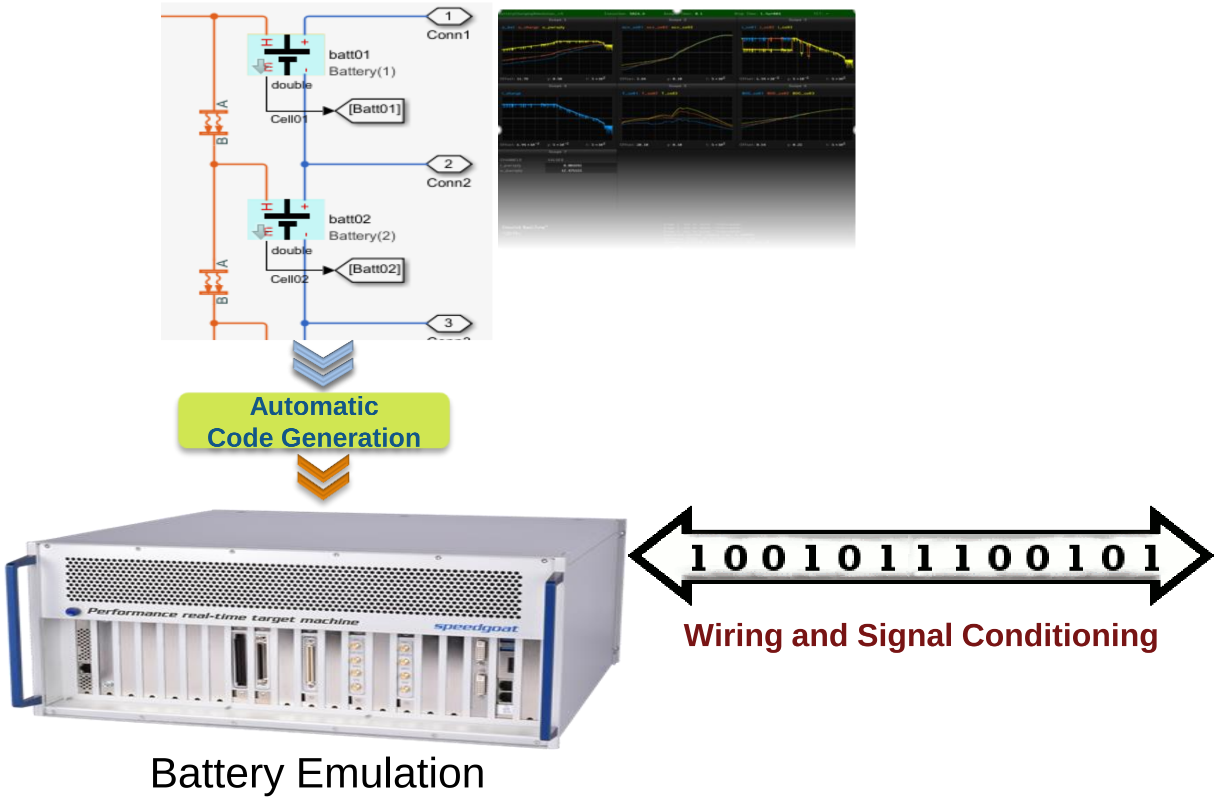


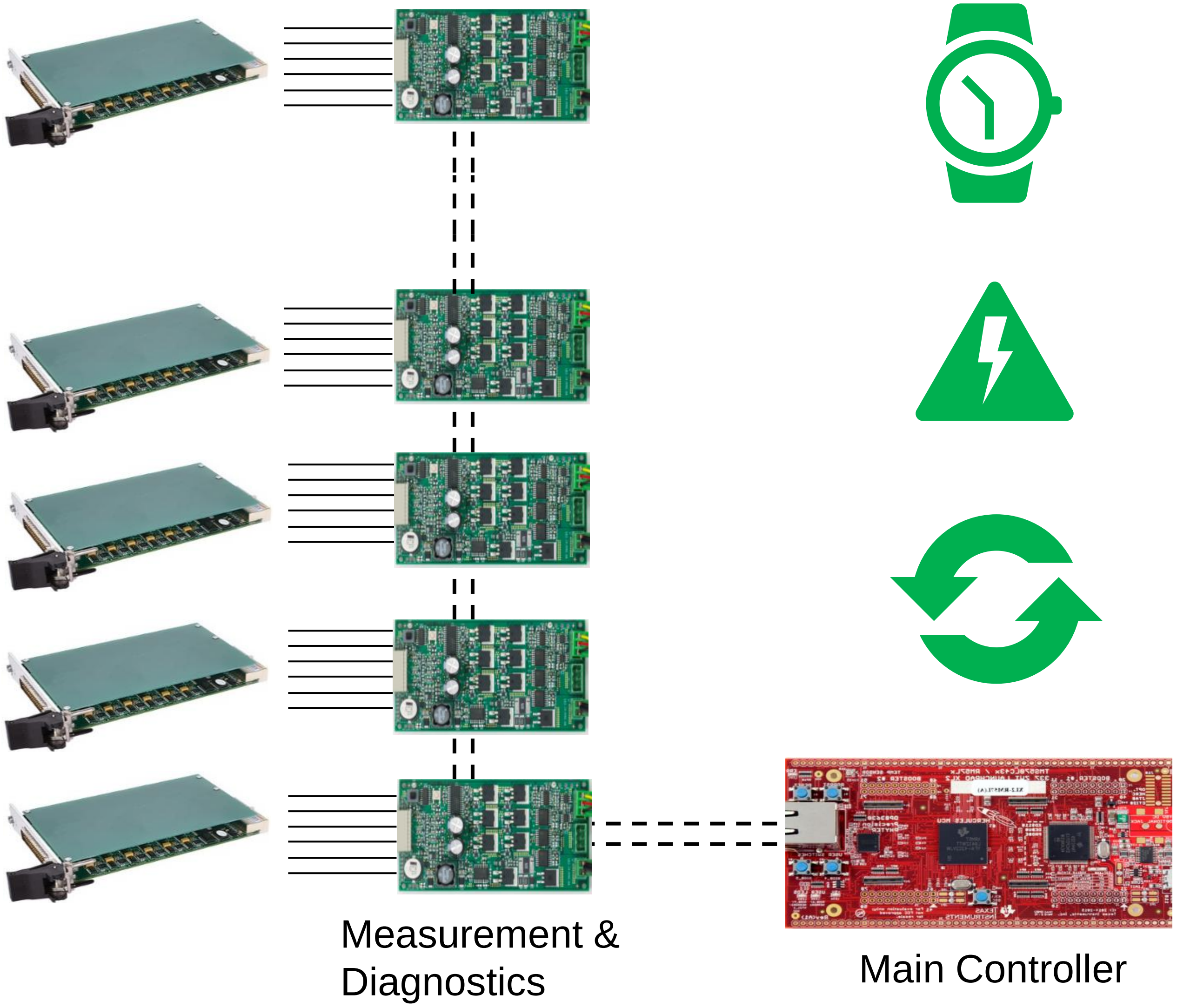


Hardware-In-Loop Testing of Battery Management System

Testing BMS with Emulated Battery Cells

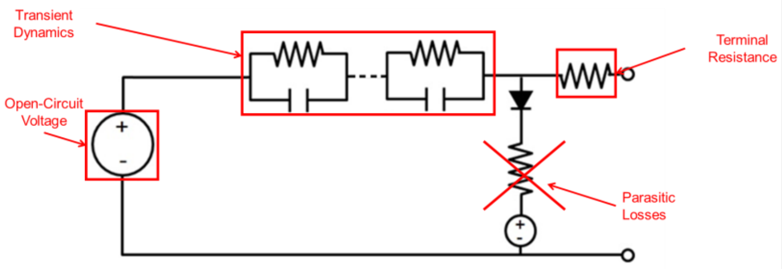
- Reduce testing time
- Test fault conditions safely
- Automate testing





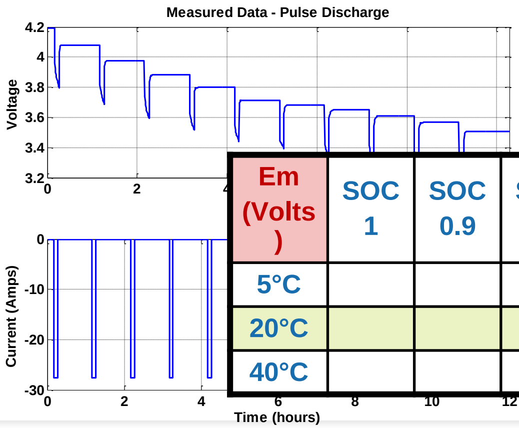
Battery Pack and BMS Design Workflow Tasks

1. Cell Modeling



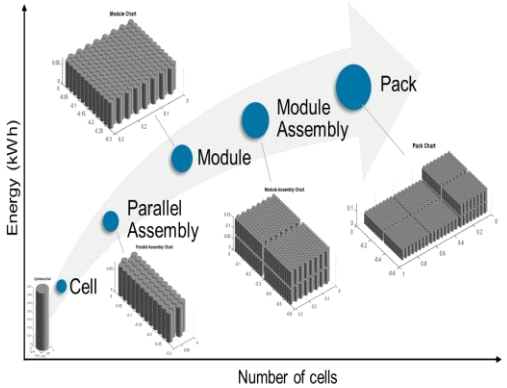
$[E_m\ R_x\ C_x] = f(\text{SOC, Temperature...})$

2. Cell Parametrization

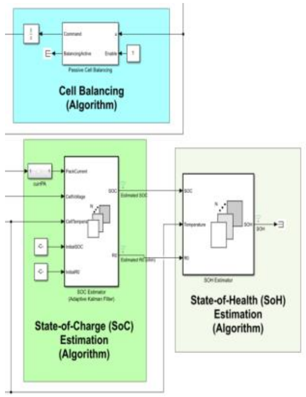


Em (Volts)	SOC 1	SOC 0.9	SOC 0.8	…	SOC 0
5°C					
20°C					
40°C					

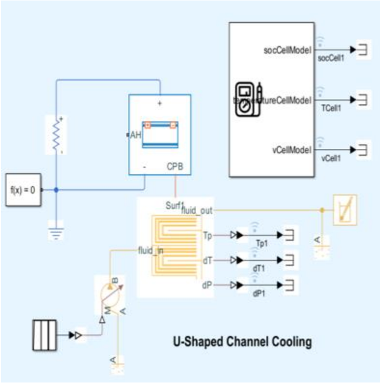
3. Battery Pack Design



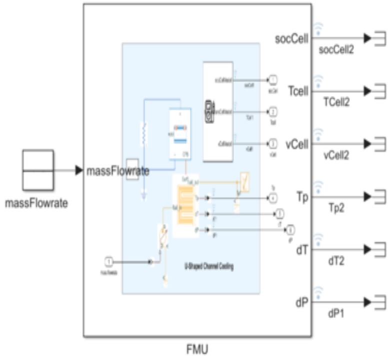
4. Battery Management System Design & Deployment



5. Thermal Management System Design



6. Deployment and HIL

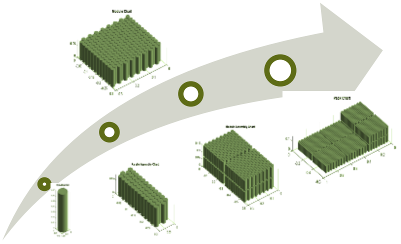


Challenges in developing a BMS

Collaboration Gap

Leverage models to communicate technical specifications, design implementation, results and maintain traceability

System Analysis



Use Simscape Battery framework that is assembled specifically to create a bridge between cell and system.

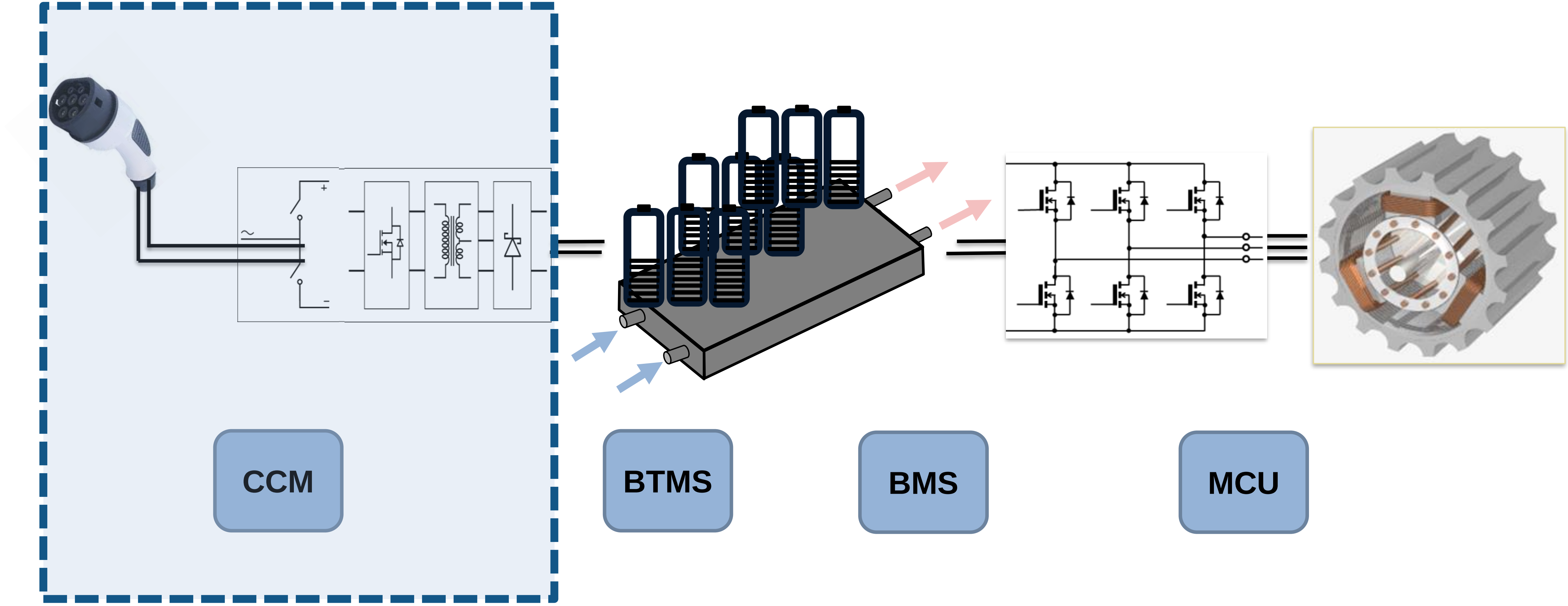
Long Iteration Cycles

Test your design iterations every step of the way through simulations and Hardware-In-Loop testing

Safety Critical System

Gain confidence in design and work towards safety certification

Electric Vehicle System



Challenges for Power Electronics Engineer

- Understand the impact of the power source and load
- Testing for a complete range of operating and fault conditions
- Designing and implementing digital controls using *only* SPICE simulator tools
- Catching errors during software-hardware integration testing
- Compliance to industry standards
- Development Time

VONSCH Speeds the Development of Control Systems for Solar Inverters and Battery Chargers

Challenge

- Develop solar inverter and battery charger control systems amid frequently shifting market requirements

Solution

- Use Model-Based Design with MATLAB and Simulink to model power electronics and control systems, run simulations, and generate embedded code for a TI microcontroller

Results

- Product development time reduced by one year
- New product R&D accelerated via model reuse
- Number of hardware prototypes reduced

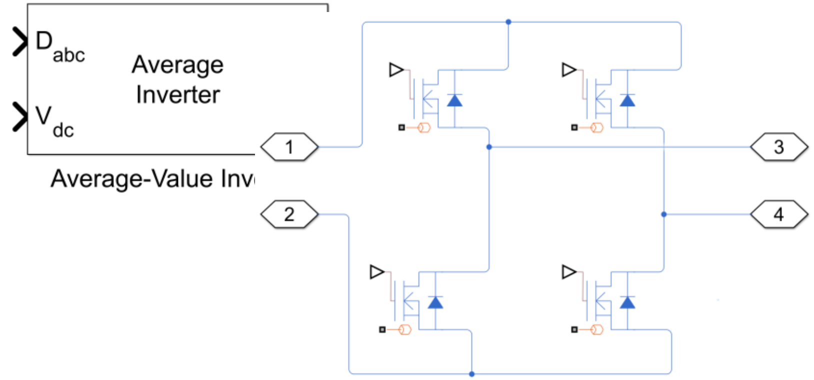
Link to user story

Development and testing of FOTO CONTROL 1f and FOTO CHARGER products.

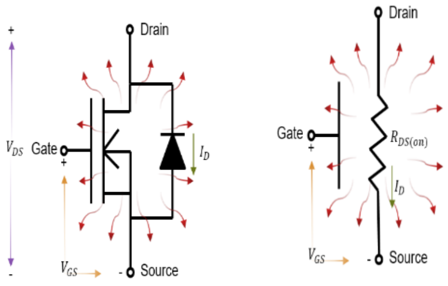
"Model-Based Design enabled us to quickly adapt to changing legislation and requirements. Before prototype hardware was available, we designed and simulated the entire system in Simulink and generated embedded code for the controller, which was working on the prototype within a day or two after the hardware was available."

Dr. Jakub Vonkomer
VONSCH

Power Converter Control Design Workflow Tasks



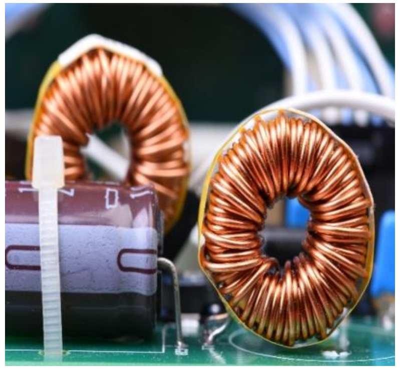
Model Fidelity



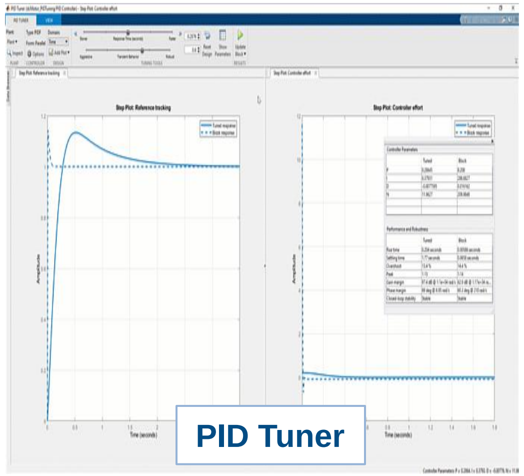
Switching loss

Conduction loss

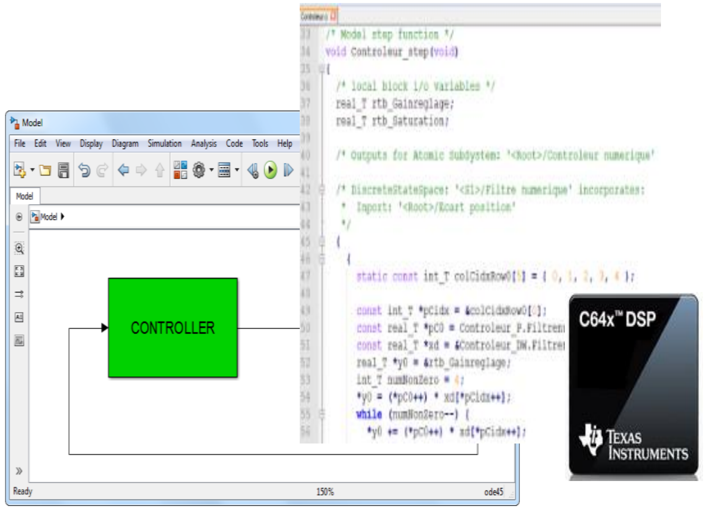
Losses Analysis



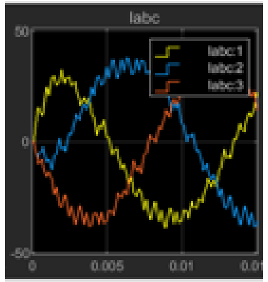
Size Components



Controller Design

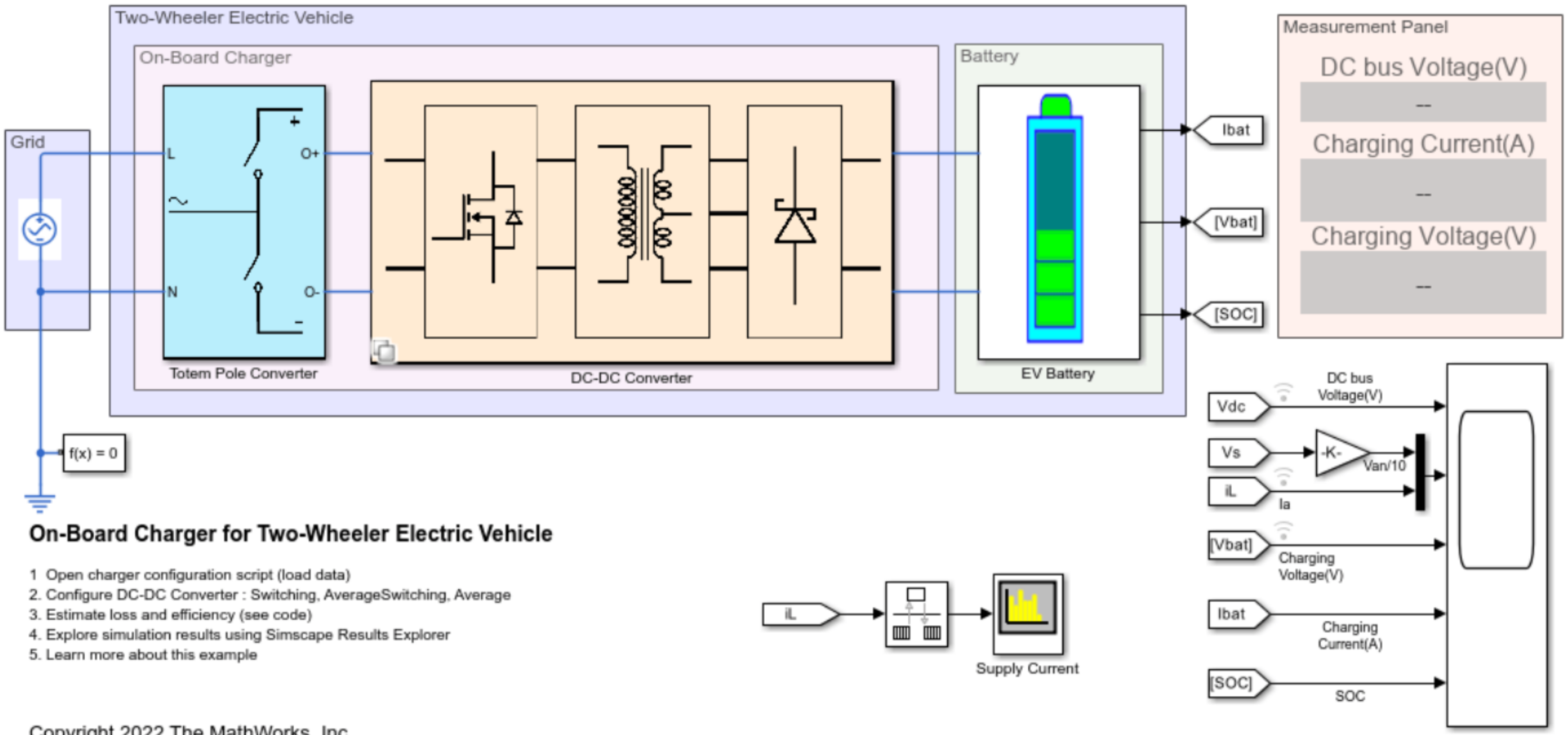


One-Click Embedded Deployment



Analyze Harmonics

On-Board Charger for Two-Wheeler Electric Vehicle



On-Board Charger for Two-Wheeler Electric Vehicle

1. Open charger configuration script (load data)
2. Configure DC-DC Converter : Switching, AverageSwitching, Average
3. Estimate loss and efficiency (see code)
4. Explore simulation results using Simscape Results Explorer
5. Learn more about this example


```
>> ee_onboard_charger_two_wheeler
```

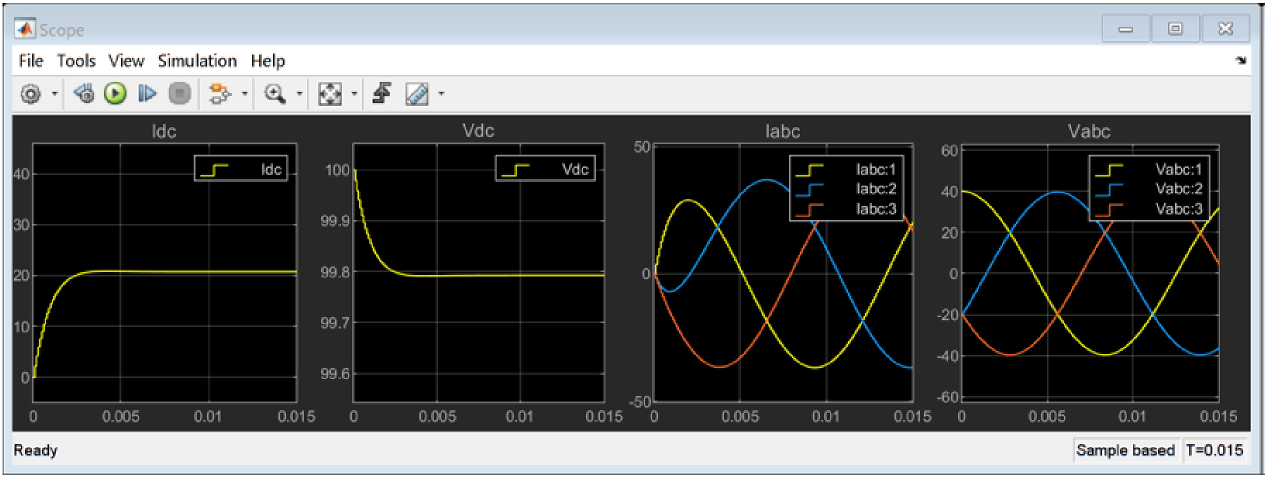
Power Converter Model Fidelity

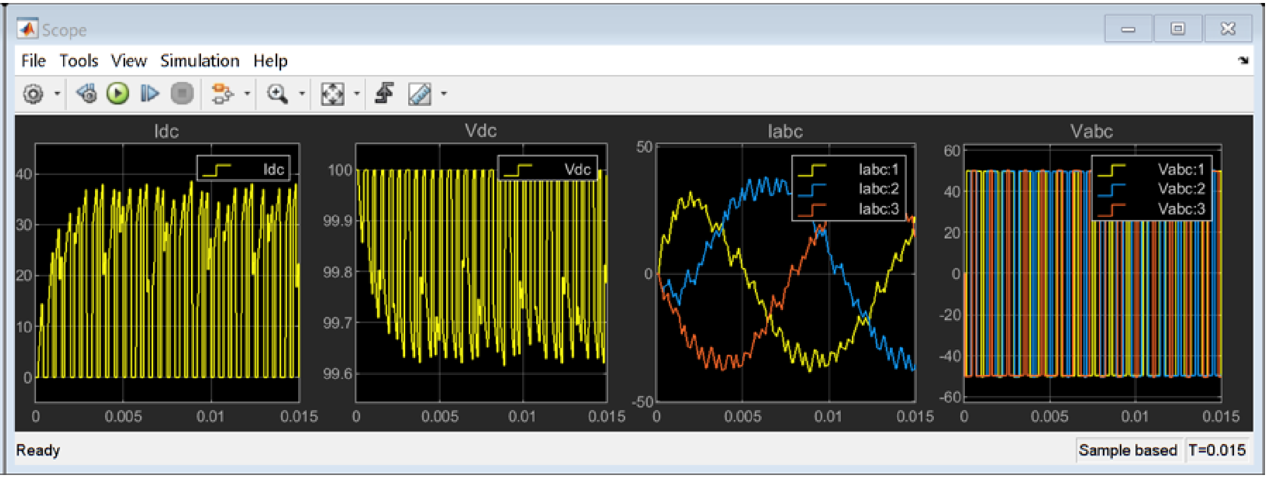
System-Level Behavior	Component Validation	Component Design

Low Fidelity **Composition** High Fidelity

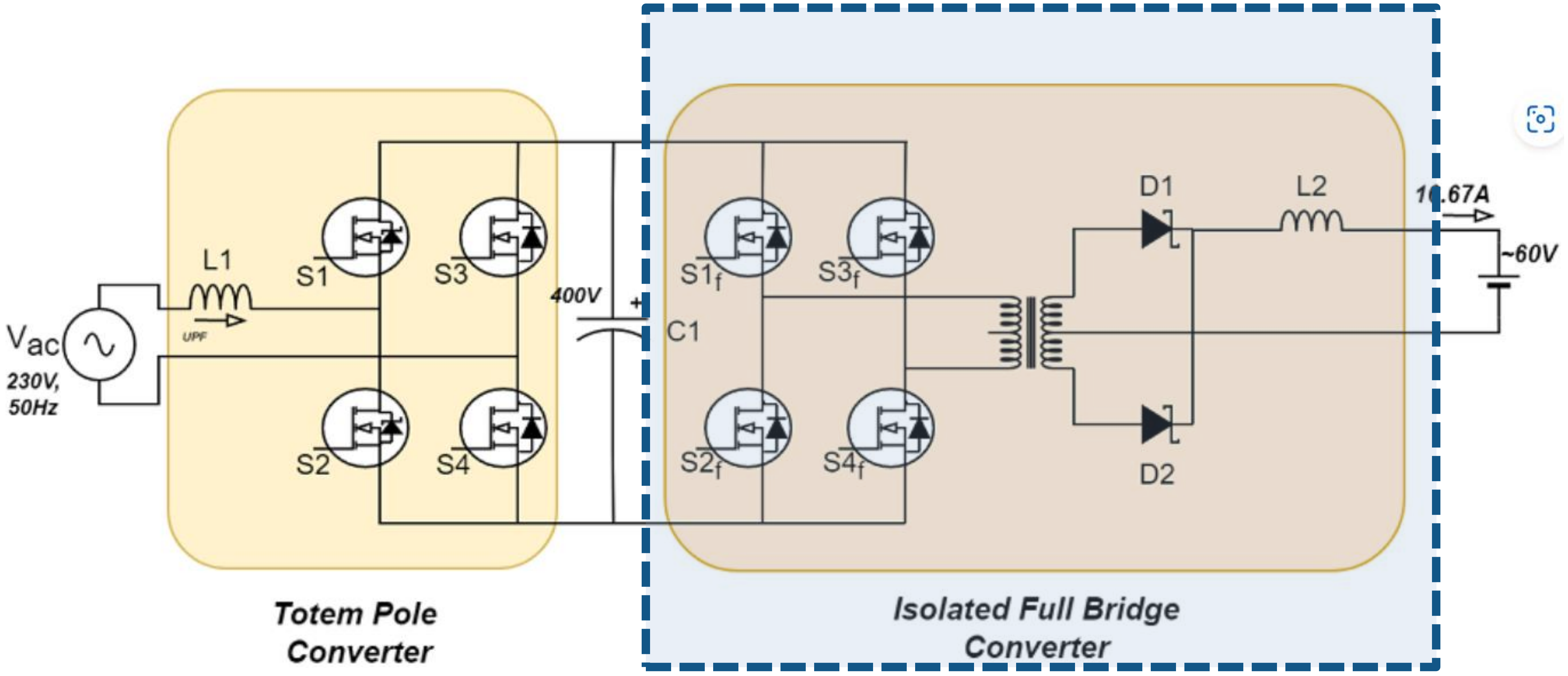
Abstracted | Waveform Controlled | PWM Controlled | Individual Switches

Semiconductor Converter Models

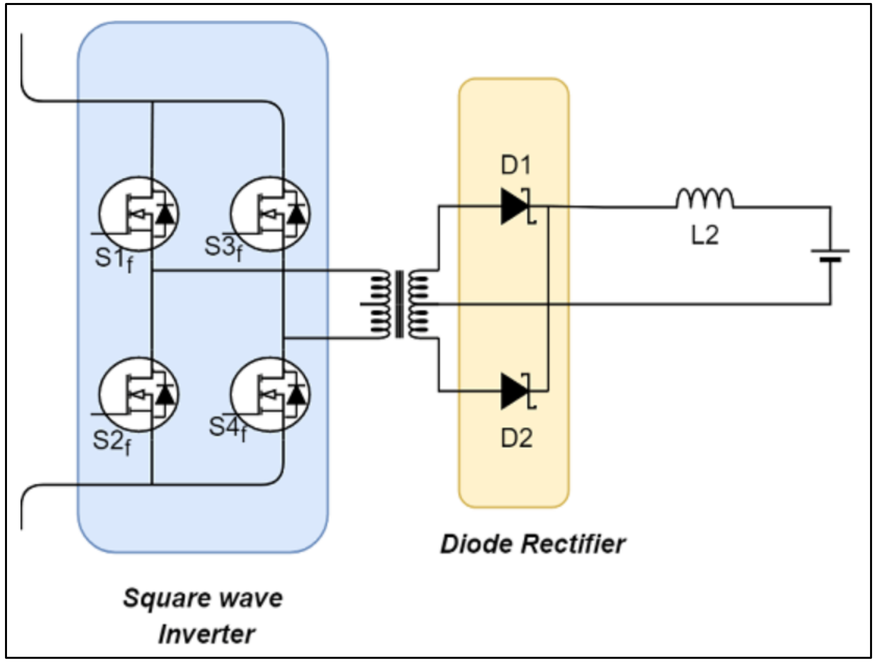


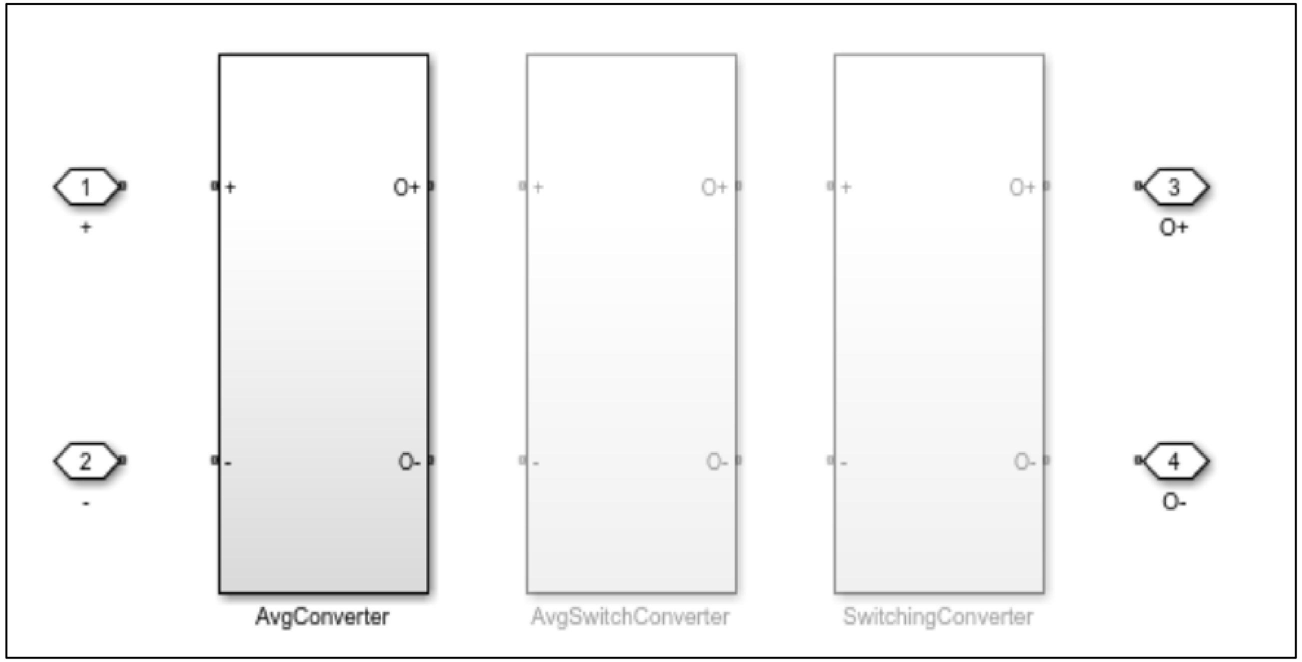


On-Board Charger for Two-Wheeler Electric Vehicle



DC-DC Converter - Variant Subsystem

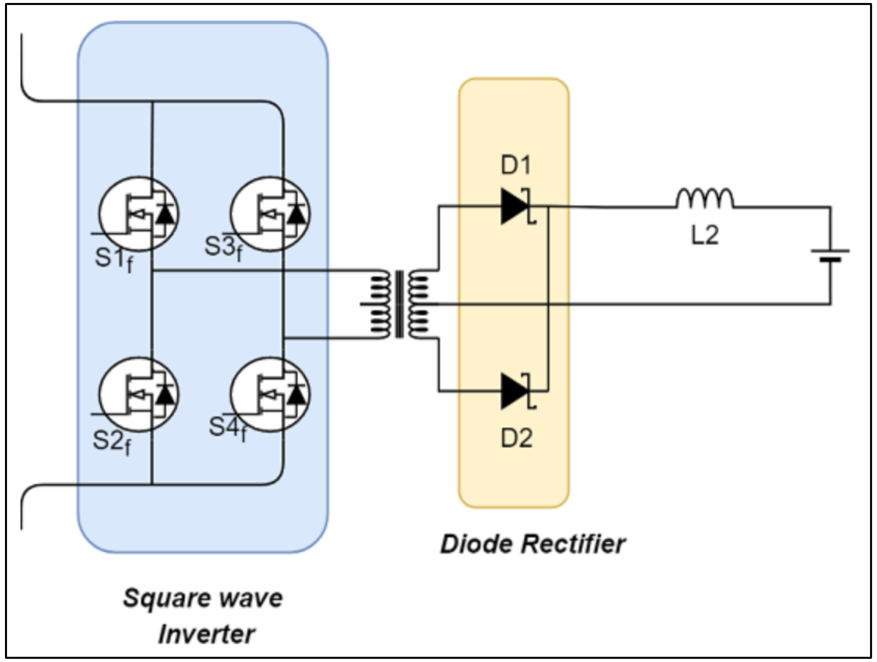


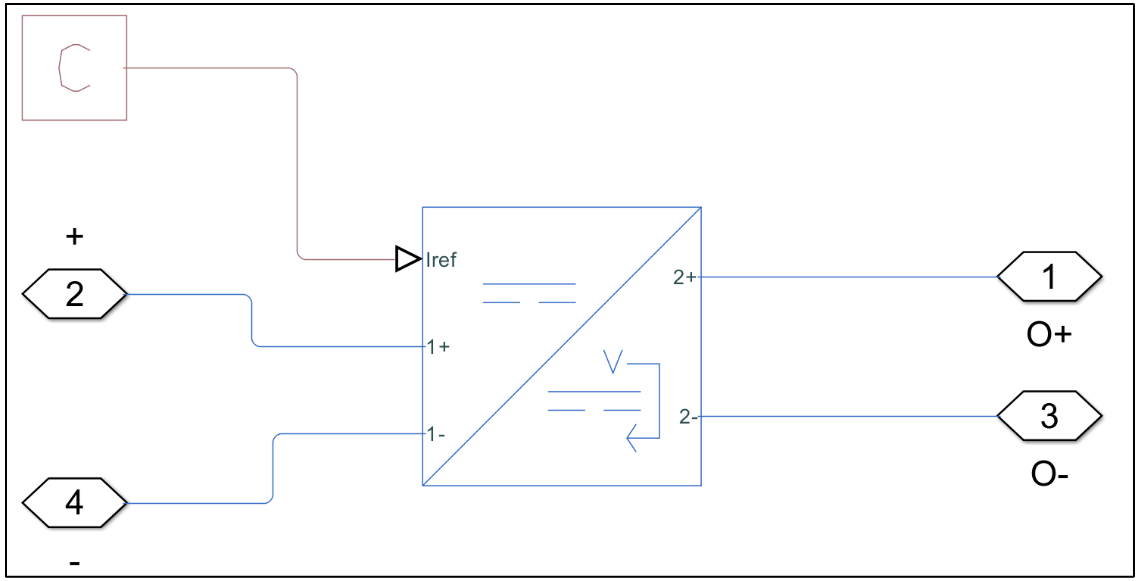


Three Variant Implementation:

- Average Converter
- Averaged Switching Converter
- Switching Converter

DC-DC Converter - Variant Subsystem

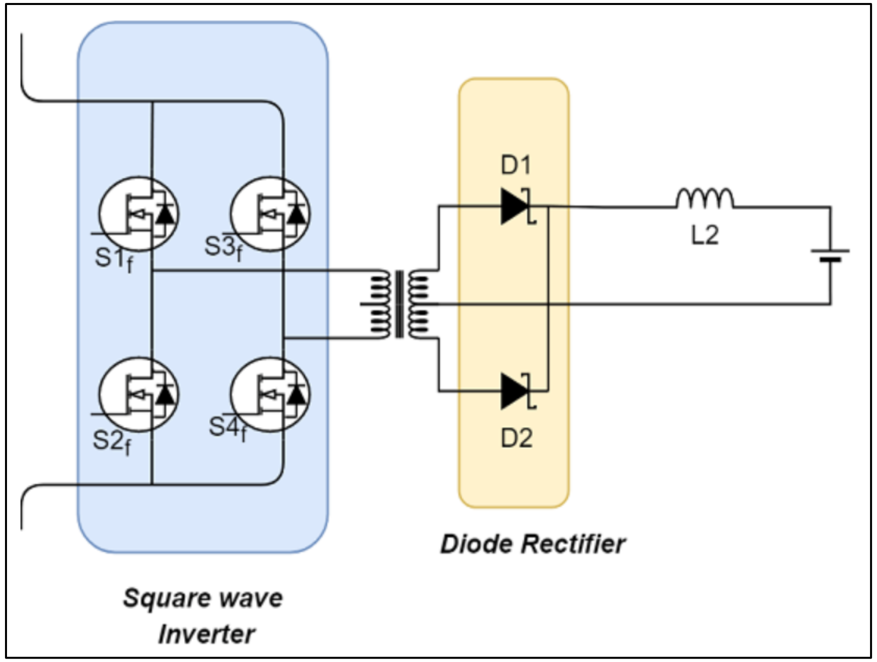


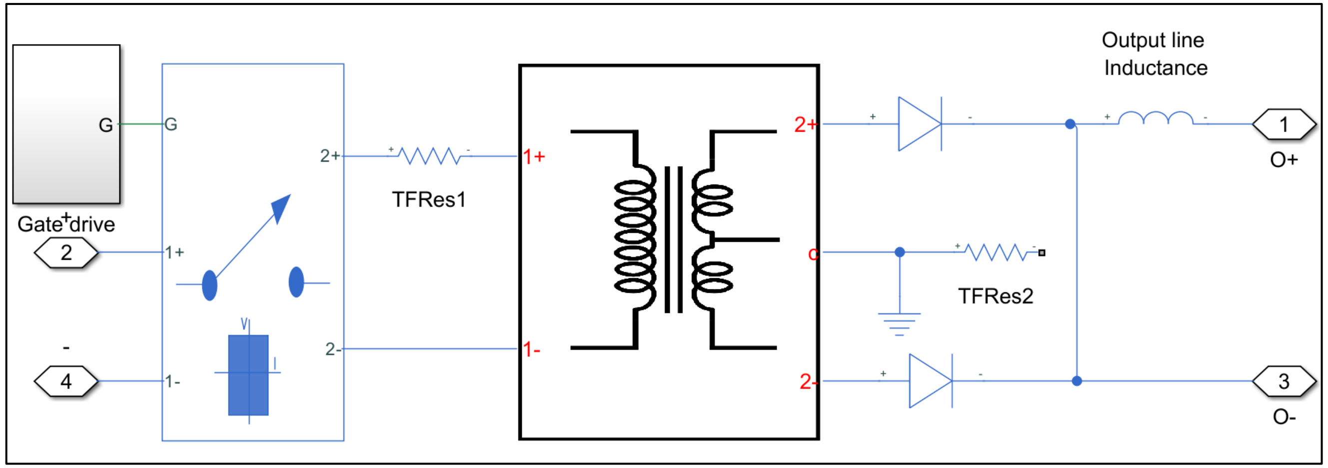


Average Converter:

- Fastest simulation
- Slight reduction in accuracy during transient
- Capture system level behavior

DC-DC Converter - Variant Subsystem

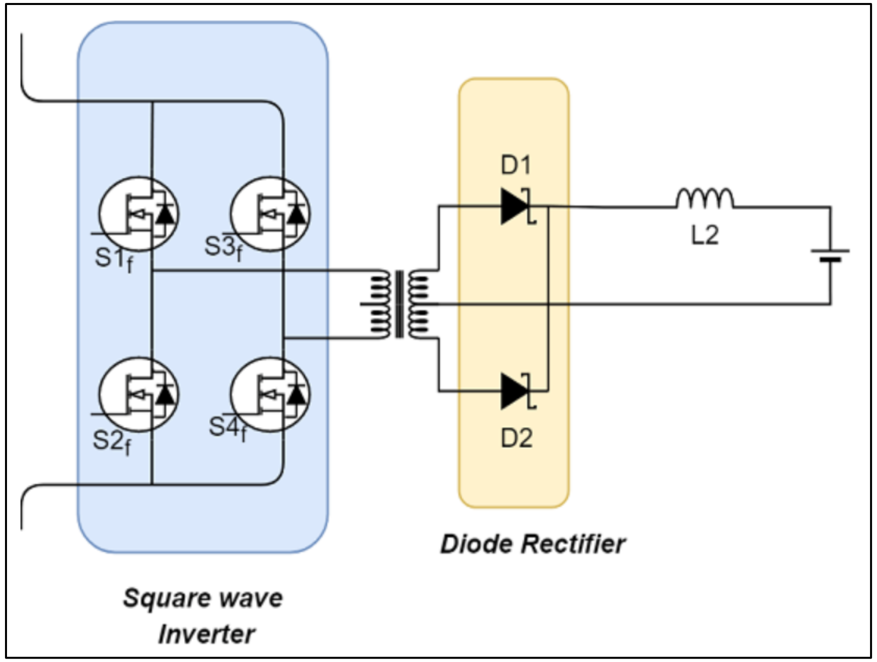


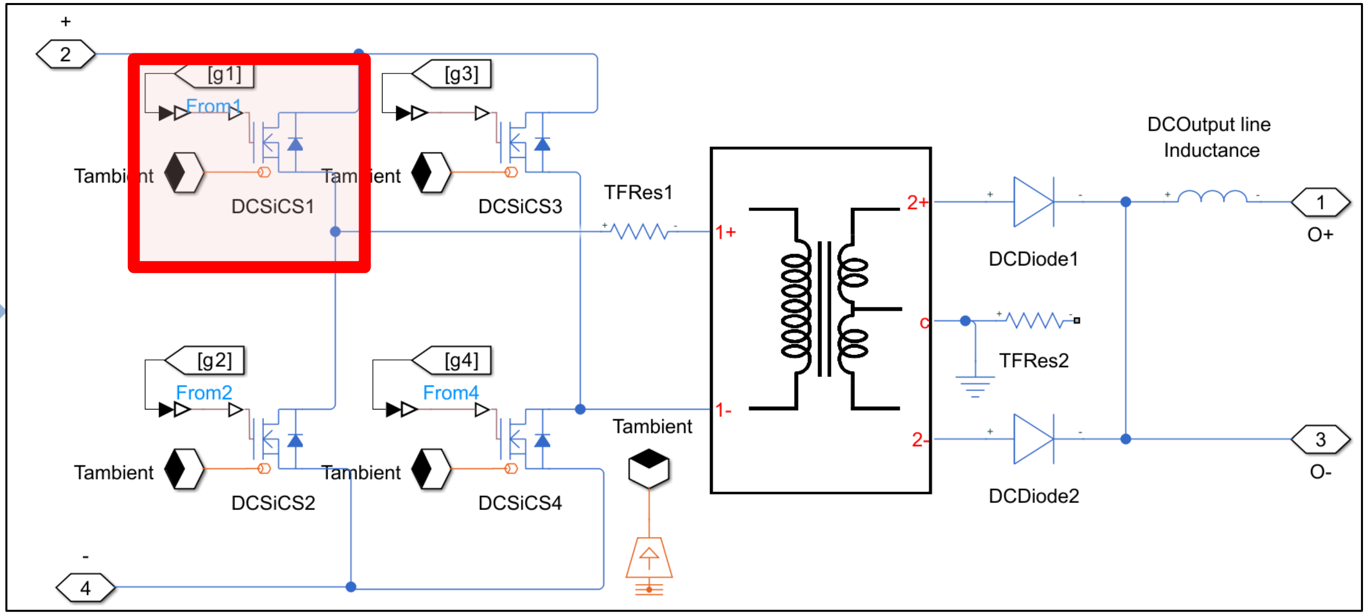


Averaged Switching Converter:

- Faster simulation time
- Better accuracy and capture switching events
- Verify the operation of the converter

DC-DC Converter - Variant Subsystem





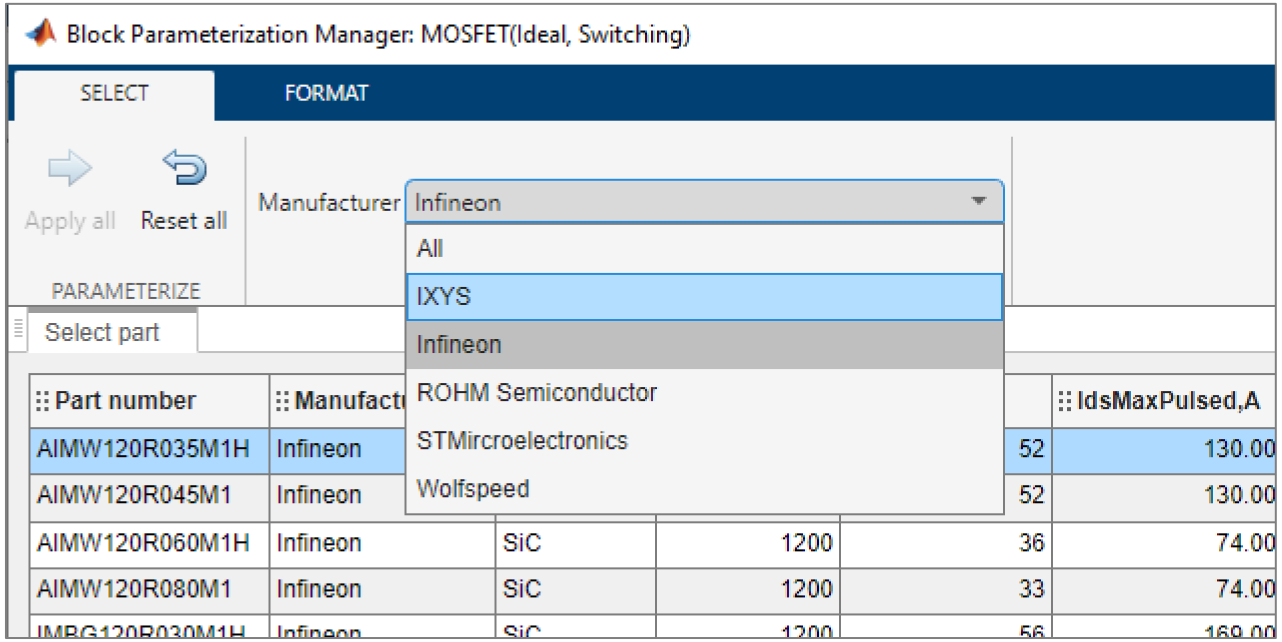
Fully Switching Converter:

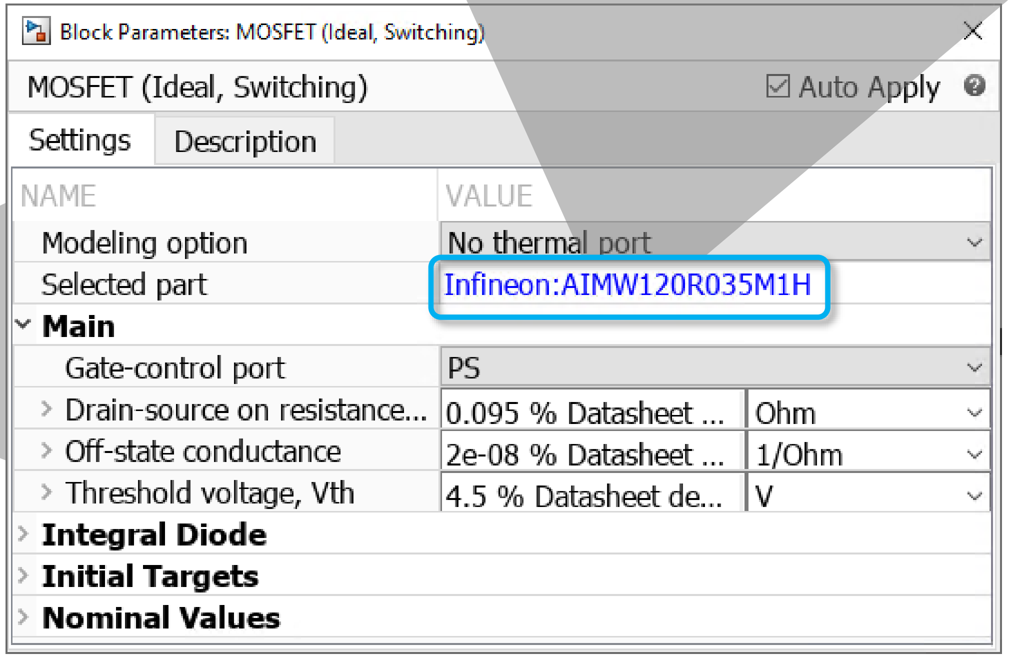
- Slower simulation
- Capture complete switching details
- Calculate converter losses and to estimate its efficiency

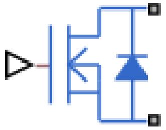
MOSFET Parametrization

Manufacturer Specific SiC MOSFETs

- Database with parameter values to match vendor-specific motors







MOSFET
(Ideal, Switching)

Import Spice Subcircuit

Buck Converter Subcircuit Import

```
>>subcircuit2ssc(BUCK_SUBCKT.CIR)
```

BUCK_SUBCKT.CIR

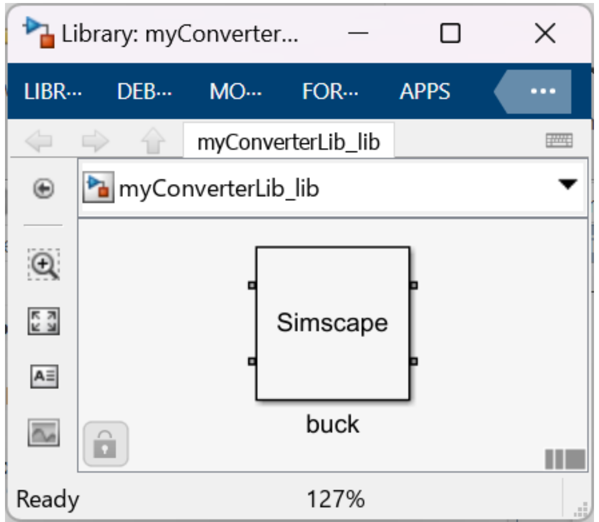
```
.subckt buck Vcc Gate Out Gnd

M1  Vcc Gate 2 2 N1
D1  Gnd 2   D
L1  2   Out 50UH
C1  Out Gnd 25UF

.MODEL N1 NMOS LEVEL=1 L=1e-6 W=1
.MODEL D D IS=0.0002 RS=0.05 CJO=5e-10

.ends
```

subcircuit2ssc

buck.ssc

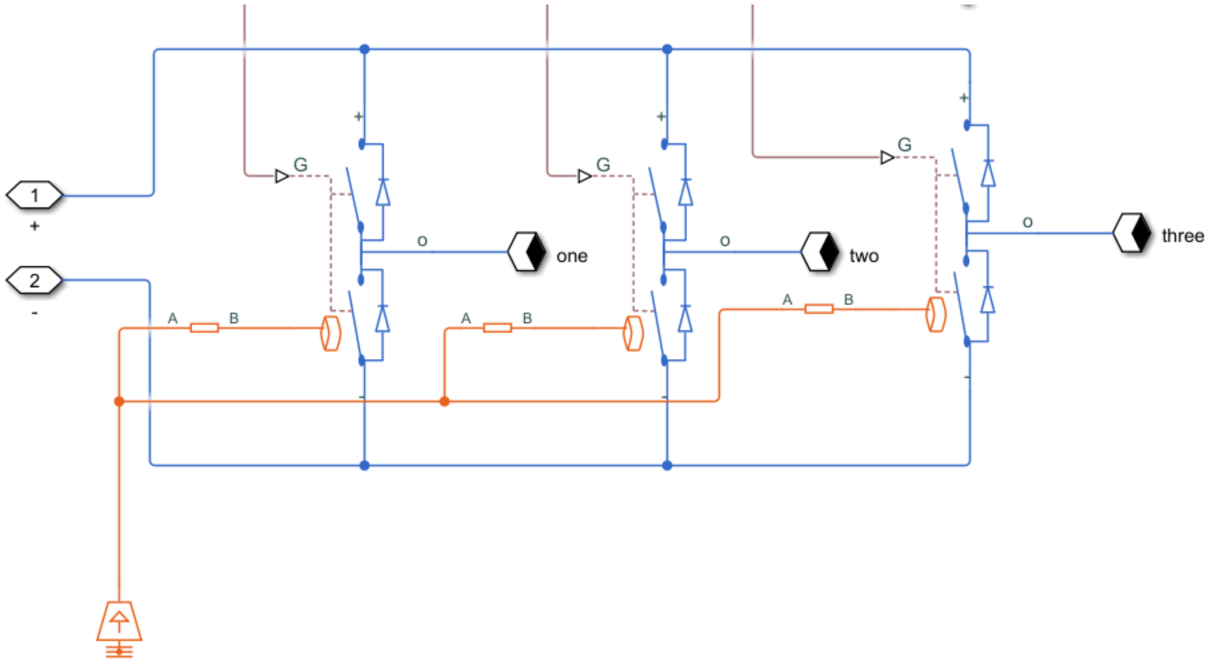
```
    components(ExternalAccess=observe)
        M1 = ee.additional.spice_semiconductors.spice_nmos(AD={0,'m^2'},AS={0,'m^2'},CBD=
            CGBO={0,'F/m'},CGDO={0,'F/m'},CGSO={0,'F/m'},CJ={0,'F/m^2'},CJSW={0,'F/m'},DE
            FC={0.5,'1'},GAMMA={nan,'V^0.5'},IS={1e-14,'A'},JS={0,'A/m^2'},KAPPA={0.2,'1'
            LD={0,'m'},LENGTH={1e-06,'m'},LEVEL={1,'1'},SCALE={1,'1'},MJ={0.5,'1'},MJSW={
            NFS={0,'1/cm^2'},NSS={nan,'1/cm^2'},NSUB={nan,'1/cm^3'},NRD={0,'1'},NRS={0,'1
            PHI={nan,'V'},PS={0,'m'},RD={nan,'Ohm'},RS={nan,'Ohm'},RSH={nan,'Ohm'},THETA=
            TPG={1,'1'},UCRIT={10000,'V/cm'},UEXP={0,'1'},U0={600,'cm^2/(V*s)'},VMAX={0,'
            WIDTH={1,'m'},Ci_param={1,'1'},Cov_param=2,C_param=2);
        D1 = ee.additional.spice_semiconductors.spice_diode(AREA={1,'m^2'},SCALE={1,'1'},
            IKF={Inf,'A/m^2'},ISR={0,'A/m^2'},N={1,'1'},NR={2,'1'},M={0.5,'1'},VJ={1,'V'}
            CJO={5e-10,'F/m^2'},FC={0.5,'1'},TT={0,'s'},RevBrk={2,'1'},BV={Inf,'V'},IBV={
            NBV={1,'1'},NBVL={1,'1'},XTI={3,'1'},EG={1.11,'eV'},TIKF={0,'K^-1'},TRS1={0,'
            TBV2={0,'K^-2'});
```

ssc_build



Parametrizing from datasheet

IGBT Module



Fuji Electric

http://www.fujielectric.com/products/semiconductor/

6MBI450V-170-50

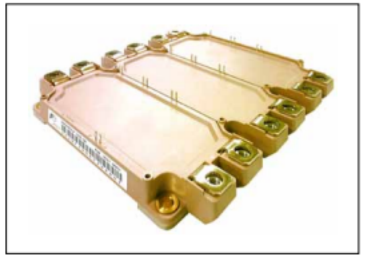
IGBT Modules

IGBT MODULE (V series)

1700V / 450A / 6 in one package

Features

- Compact Package
- P.C.Board Mount
- Low V_{CE} (sat)



Applications

- Inverter for Motor Drive
- AC and DC Servo Drive Amplifier
- Uninterruptible Power Supply
- Industrial machines, such as welding machines

Maximum Ratings and Characteristics

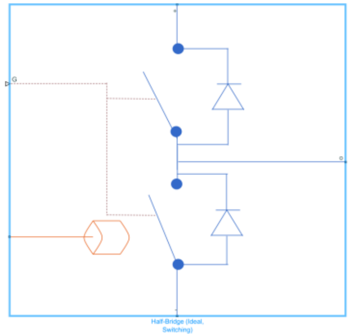
● **Absolute Maximum Ratings (at T_c=25°C unless otherwise specified)**

Items		Symbols	Conditions		Maximum ratings	Units
Inverter	Collector-Emitter voltage	V_{CES}			1700	V
	Gate-Emitter voltage	V_{GES}			±20	V
	Collector current	I_C	Continuous	T_c=25°C	600	A
				T_c=100°C	450	
		$I_{C\,pulse}$	1ms		900	
		$-I_C$			450	
		$-I_{C\,pulse}$	1ms		900	
	Collector power dissipation	P_C	1 device		2500	W
Junction temperature		T_j			175	°C
Operating junction temperature (under switching conditions)		T_{jop}			150	
Case temperature		T_C			125	
Storage temperature		T_{stg}			-40 ~ 125	
Isolation voltage	Between terminal and copper base (*1)	V_{iso}	AC : 1min.		3400	VAC
	Between thermistor and others (*2)					
Screw torque	Mounting (*3)	-			3.5	N m
	Terminals (*4)	-			4.5	

Note *1: All terminals should be connected together during the test.
Note *2: Two thermistor terminals should be connected together, other terminals should be connected together and shorted to base plate during the test.
Note *3: Recommendable Value : 2.5-3.5 Nm (M5)
Note *4: Recommendable Value : 3.5-4.5 Nm (M6)

Parametrizing from datasheet

IGBT Module

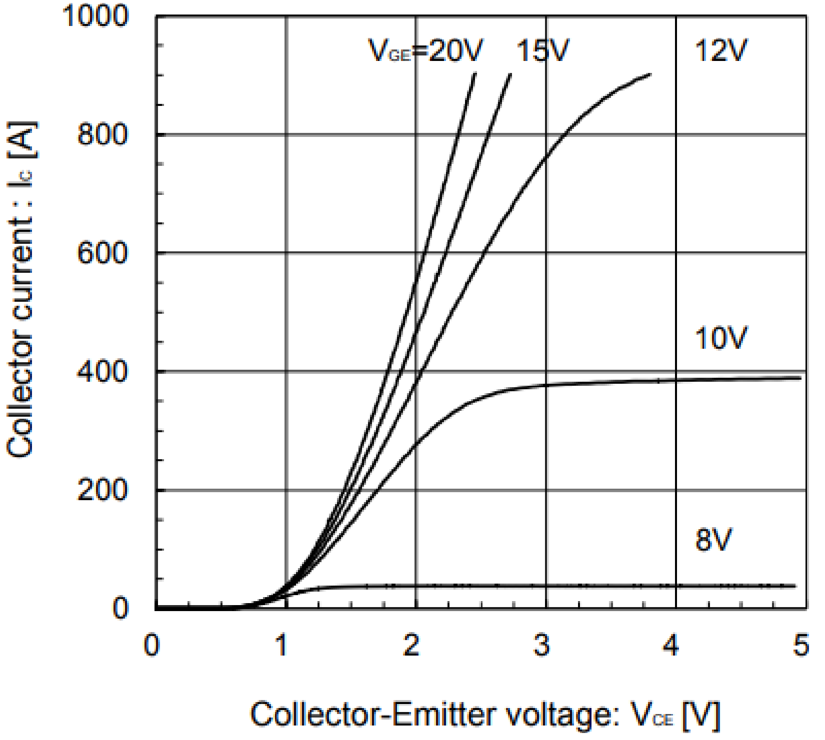


Main			
Gate-control port	PS		
Switching device	IGBT		
Threshold voltage, Vth	Vge_th	6.5	V
On-state behavior and losses	Tabulate with temperature and current		
On-state voltage, Vce(Tj,Ice)	Vce	<2x9 double>	V
Temperature vector, Tj	[25 150]		degC
Collector-emitter current vector, Ice	Ids	<1x9 double>	A
Off-state conductance	1e-4		1/Ohm

[Inverter]

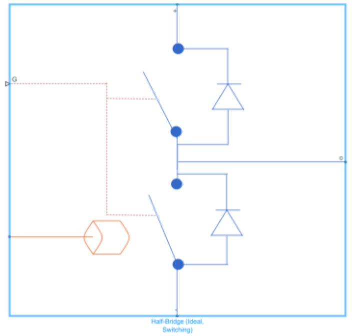
Collector current vs. Collector-Emitter voltage (typ.)

T_j= 25°C / chip

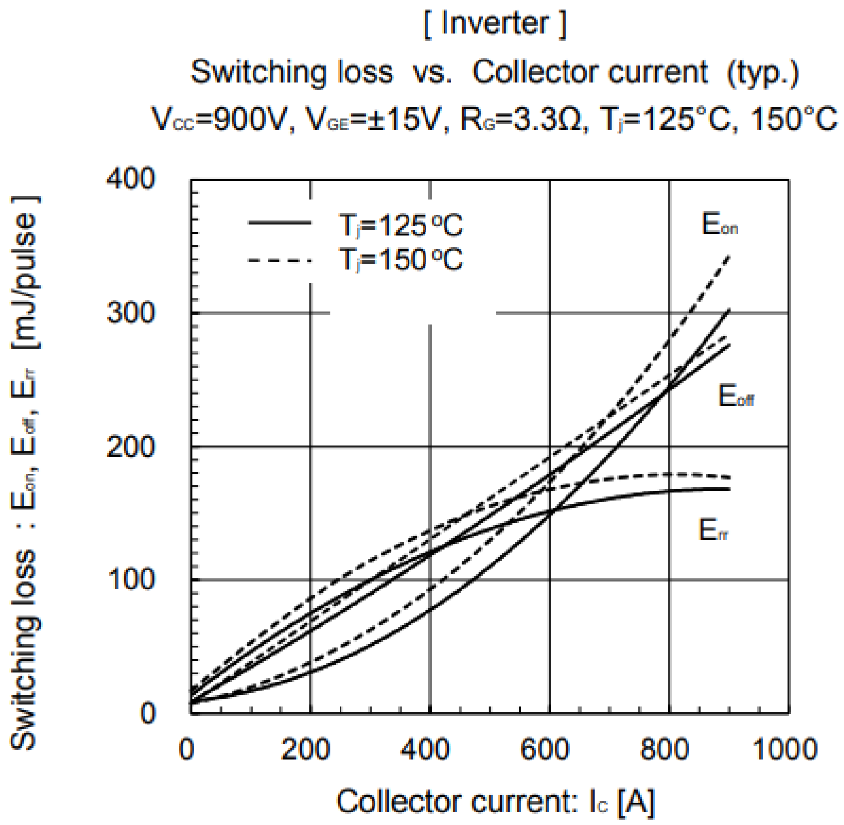


Parametrizing from datasheet

IGBT Module

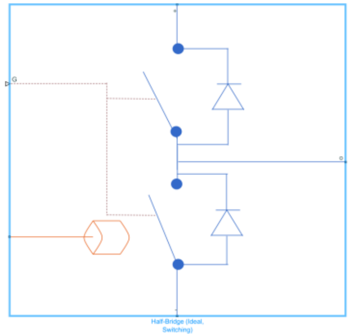


Losses			
Switch-on loss, Eon(Tj,Ice)	Eon	<2x9 double>	mJ
Switch-off loss, Eoff(Tj,Ice)	Eoff	<2x9 double>	mJ
Diode reverse recovery loss, Erec(Tj,Ice)	Err	<2x9 double>	mJ
Temperature vector for losses, Tj	[125 150]		degC
Collector-emitter current vector for losses,...	Ids	<1x9 double>	A
Off-state voltage for loss data	900		V



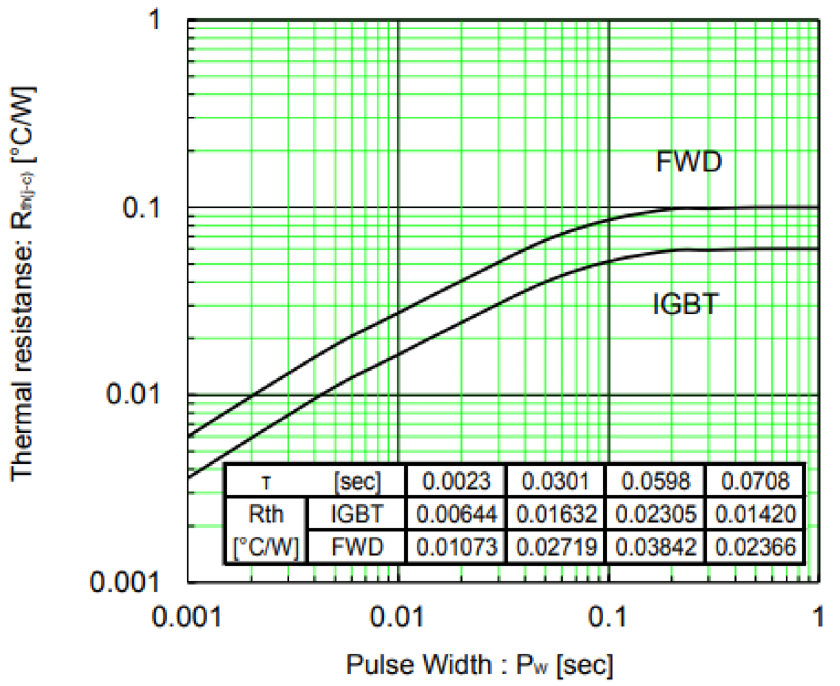
Parametrizing from datasheet

IGBT Module



Thermal Port

Thermal network	Cauer model	
Thermal resistances, [R1 R2 ... Rn]	Rt [0.00644,0.01632,0.02305,0.0...	K/W
Thermal mass parameterization	By thermal time constants	
Thermal time constants, [t1 t2 ... tn]	tou [0.0023,0.0301,0.0598,0.0708]	s
Thermal masses initial temperatures, [T1 T...	[25, 25, 25,25]	degC



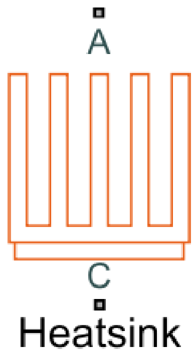
τ	[sec]	0.0023	0.0301	0.0598	0.0708
Rth	IGBT	0.00644	0.01632	0.02305	0.01420
[°C/W]	FWD	0.01073	0.02719	0.03842	0.02366

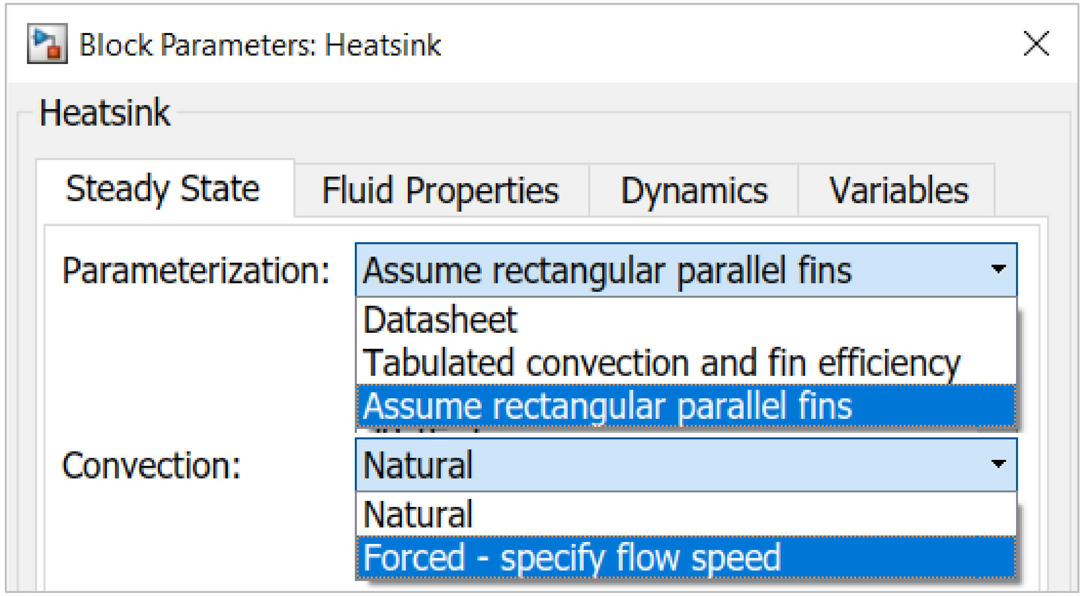
FWD safe operating area (max.)

T_j=150°C

Heatsink Block

- Model a heatsink used to dissipate heat from power electronics
  - Model conduction through fins and convection to environment
- Flexible parameterization
  - Datasheet
  - Tabulated heat transfer properties
  - Geometry assuming an empirical convection correlation



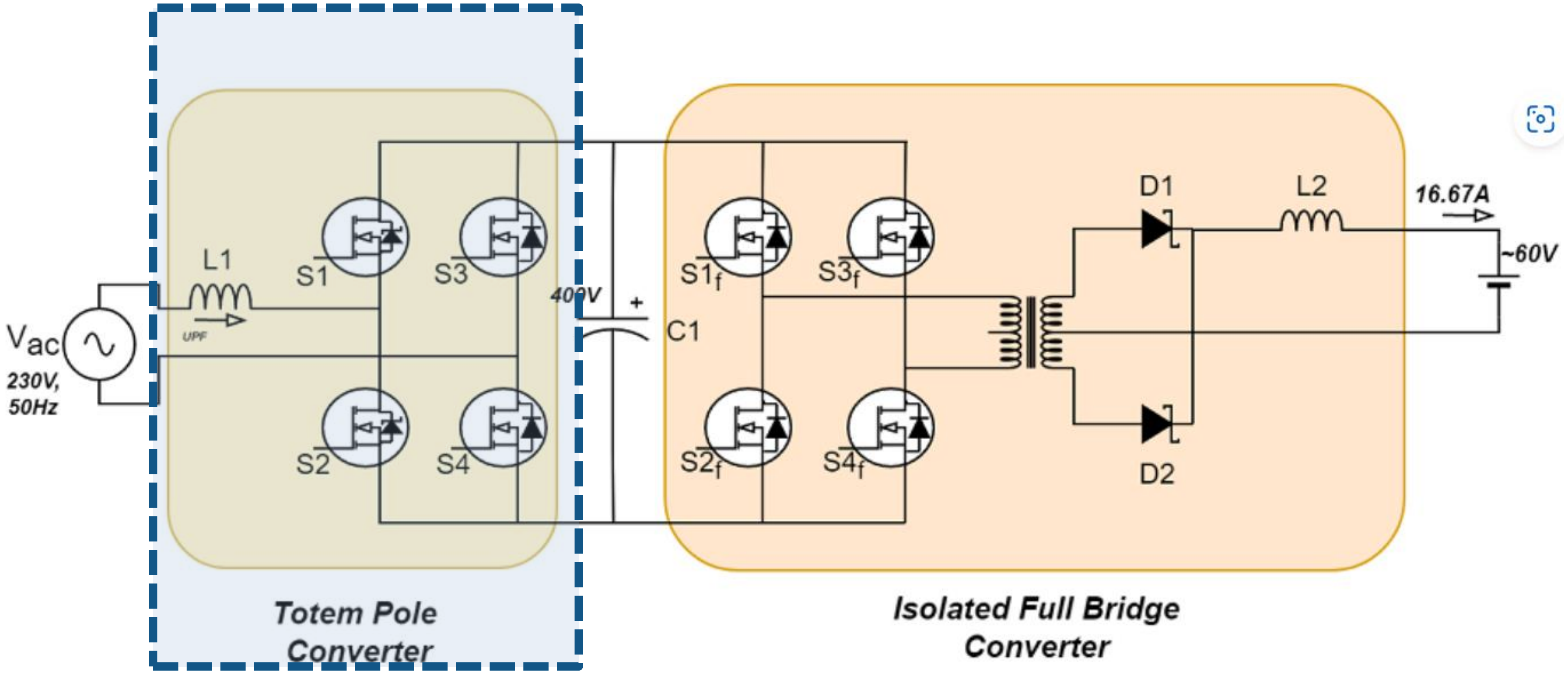


Calculate Power Loss and Efficiency

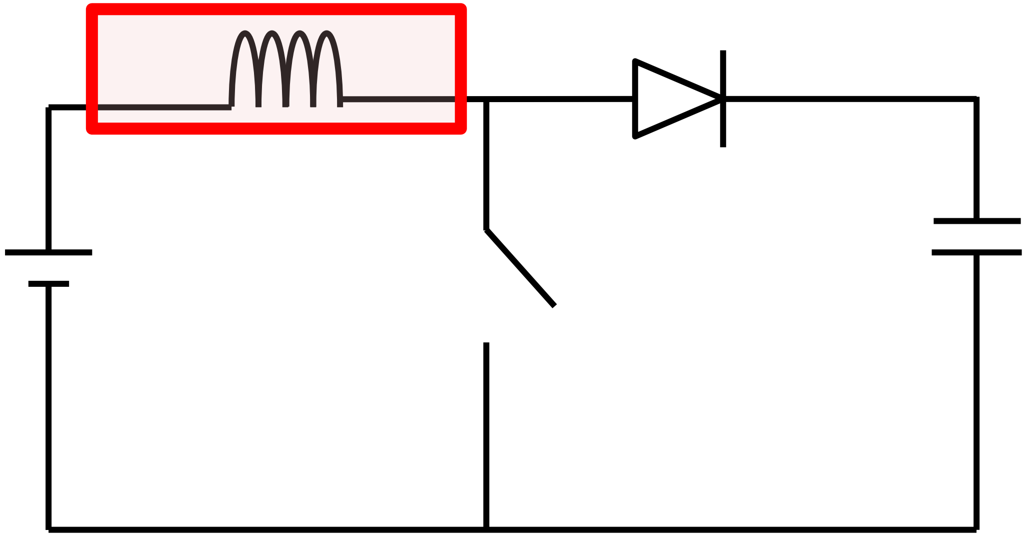
>> ee_getEfficiency
>> ee_getPowerLossSummary
>> ee_getPowerLossTimeSeries

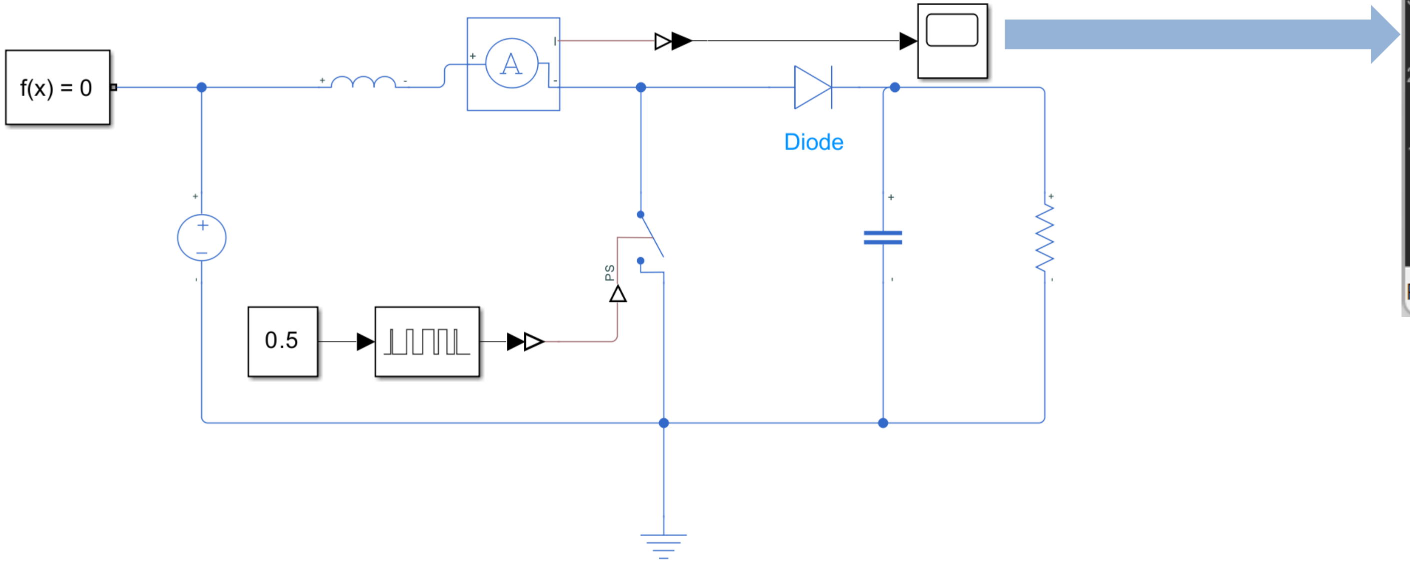
```
>> ee_getPowerLossSummary(simlog_ee_onboard_charger_two_wheeler)

ans =

  16×3 table

                                       LoggingNode                                            Power      SwitchingLosses
    ___________________________________________________________________________________    _________    _______________

    {'ee_onboard_charger_two_wheeler.DC_DC_Converter.SwitchingConverter.TFRes1'            }     74.369              0
    {'ee_onboard_charger_two_wheeler.DC_DC_Converter.SwitchingConverter.DCDiode2'          }      27.44              0
    {'ee_onboard_charger_two_wheeler.DC_DC_Converter.SwitchingConverter.DCDiode1'          }     23.925              0
    {'ee_onboard_charger_two_wheeler.DC_DC_Converter.SwitchingConverter.TFRes2'            }     19.237              0
    {'ee_onboard_charger_two_wheeler.Totem_Pole_Converter.Converter.TPSiMOSL'              }     5.2336              0
    {'ee_onboard_charger_two_wheeler.Totem_Pole_Converter.Converter.TPSiMOSH'              }     5.1749              0
    {'ee_onboard_charger_two_wheeler.Totem_Pole_Converter.Converter.TPSiCMOSL'             }     4.7752        0.56326
    {'ee_onboard_charger_two_wheeler.Totem_Pole_Converter.Converter.TPSiCMOSH'             }     4.6683        0.56565
    {'ee_onboard_charger_two_wheeler.DC_DC_Converter.SwitchingConverter.SiC_Inverter.DCSiCS3'     }     2.8173        0.63248
    {'ee_onboard_charger_two_wheeler.DC_DC_Converter.SwitchingConverter.SiC_Inverter.DCSiCS1'     }     2.5285        0.63496
    {'ee_onboard_charger_two_wheeler.DC_DC_Converter.SwitchingConverter.SiC_Inverter.DCSiCS2'     }     1.7666         0.6308
    {'ee_onboard_charger_two_wheeler.DC_DC_Converter.SwitchingConverter.DCOutput_line_Inductance'}     1.6031              0
    {'ee_onboard_charger_two_wheeler.DC_DC_Converter.SwitchingConverter.SiC_Inverter.DCSiCS4'     }     1.4778     0.00088757
    {'ee_onboard_charger_two_wheeler.Totem_Pole_Converter.Converter.TPInductor'            }    0.37979              0
    {'ee_onboard_charger_two_wheeler.EV_Battery.EV_Battery.Battery_Table_Based1'           }   0.074153              0
    {'ee_onboard_charger_two_wheeler.Totem_Pole_Converter.C'                               }  0.0022549              0
```

On-Board Charger for Two-Wheeler Electric Vehicle

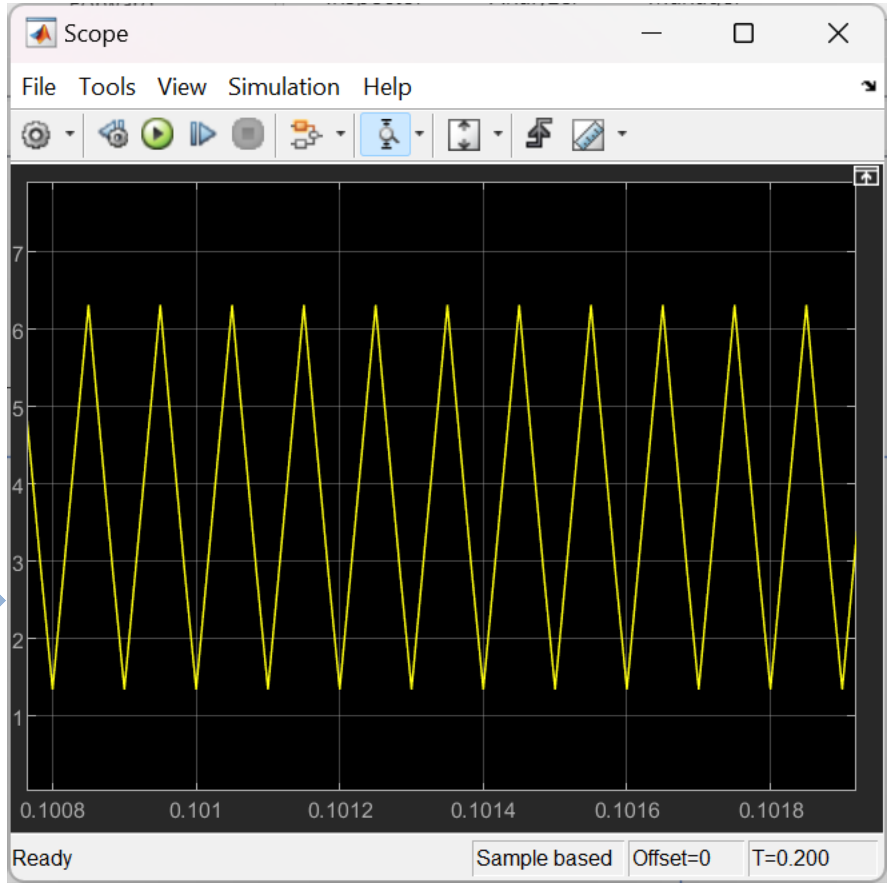


Component Sizing using Simulation

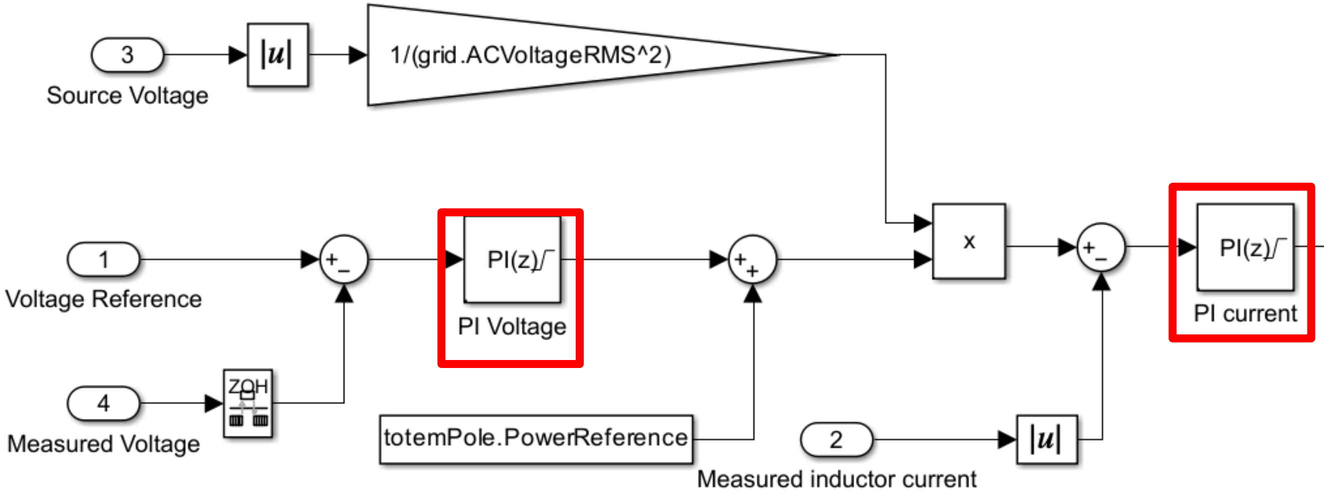


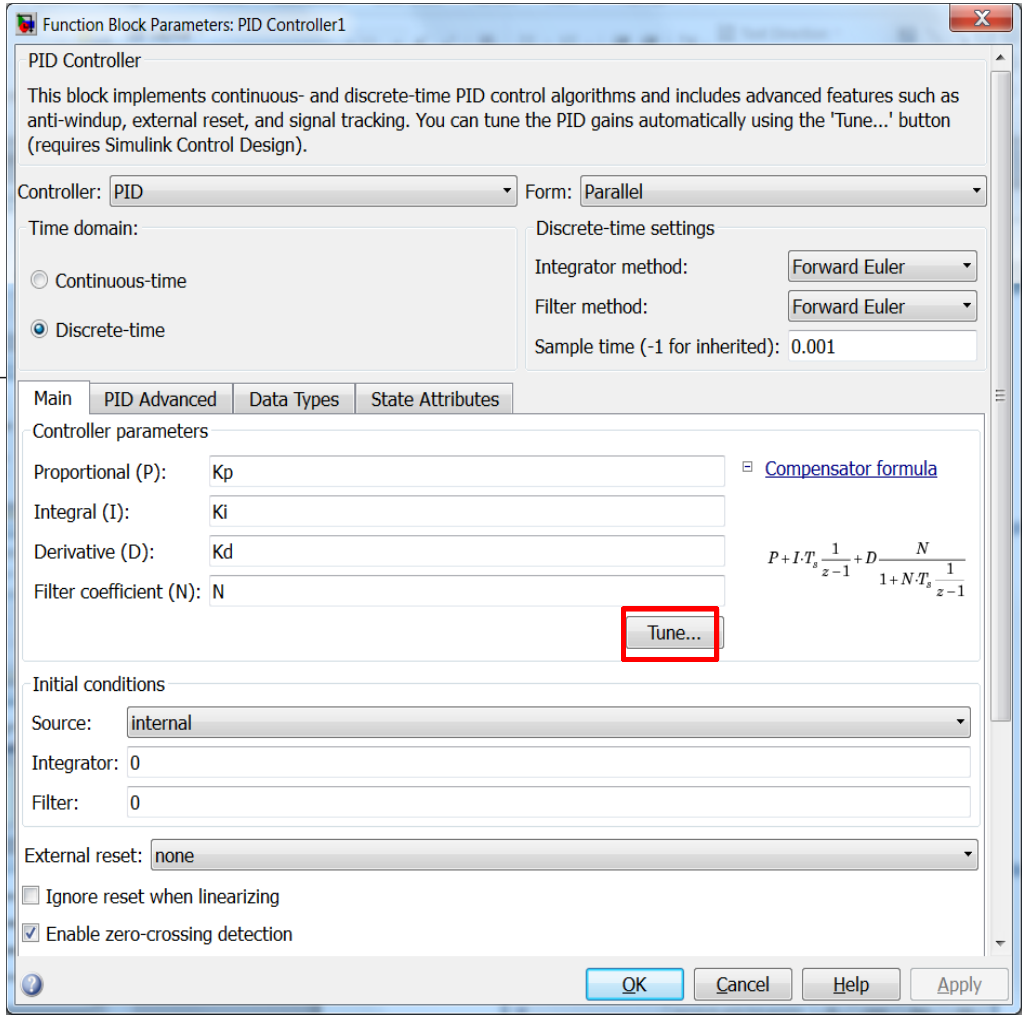


DisContinuous Conduction Mode



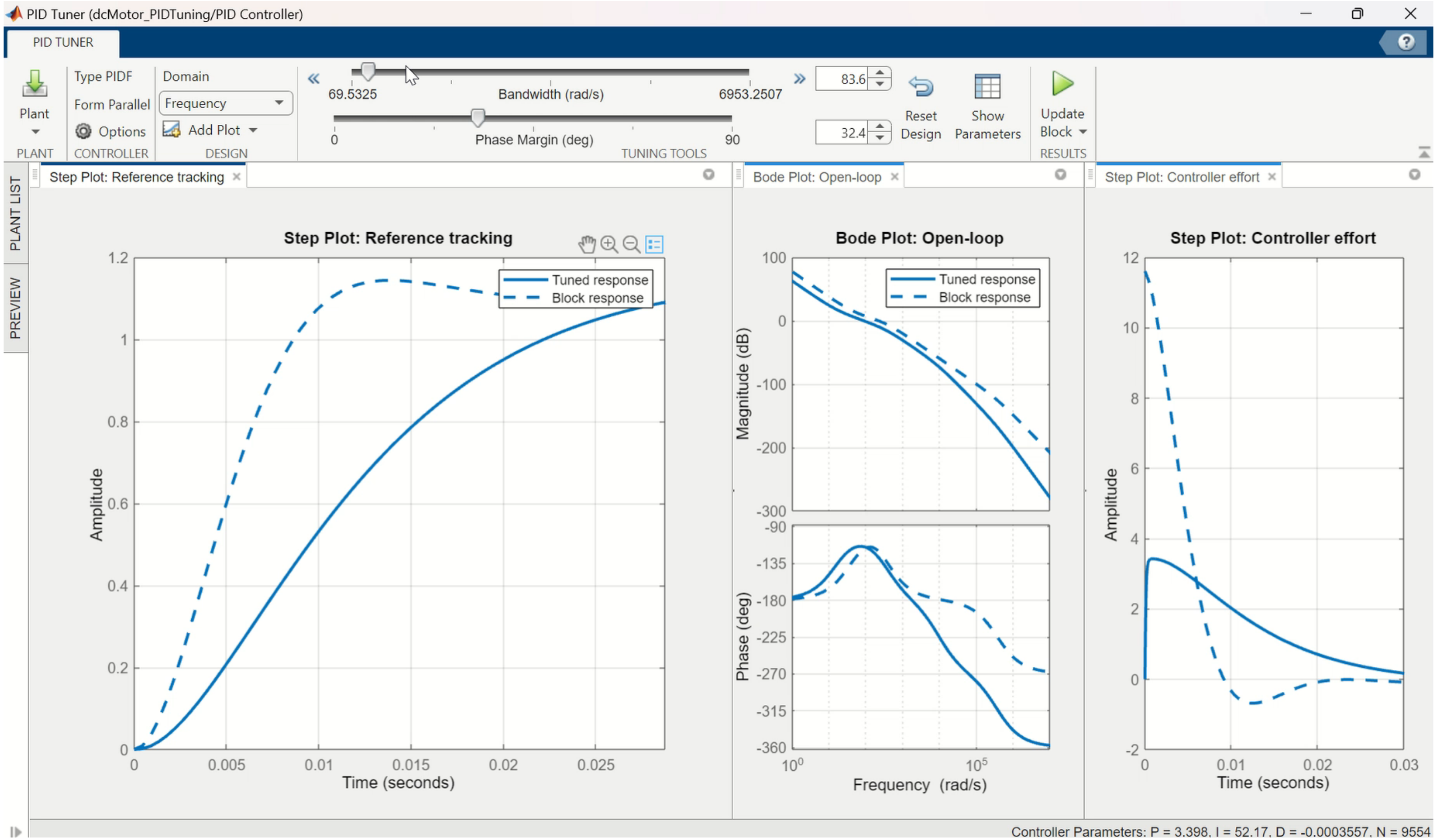
Totem Pole Converter for Power Factor Correction



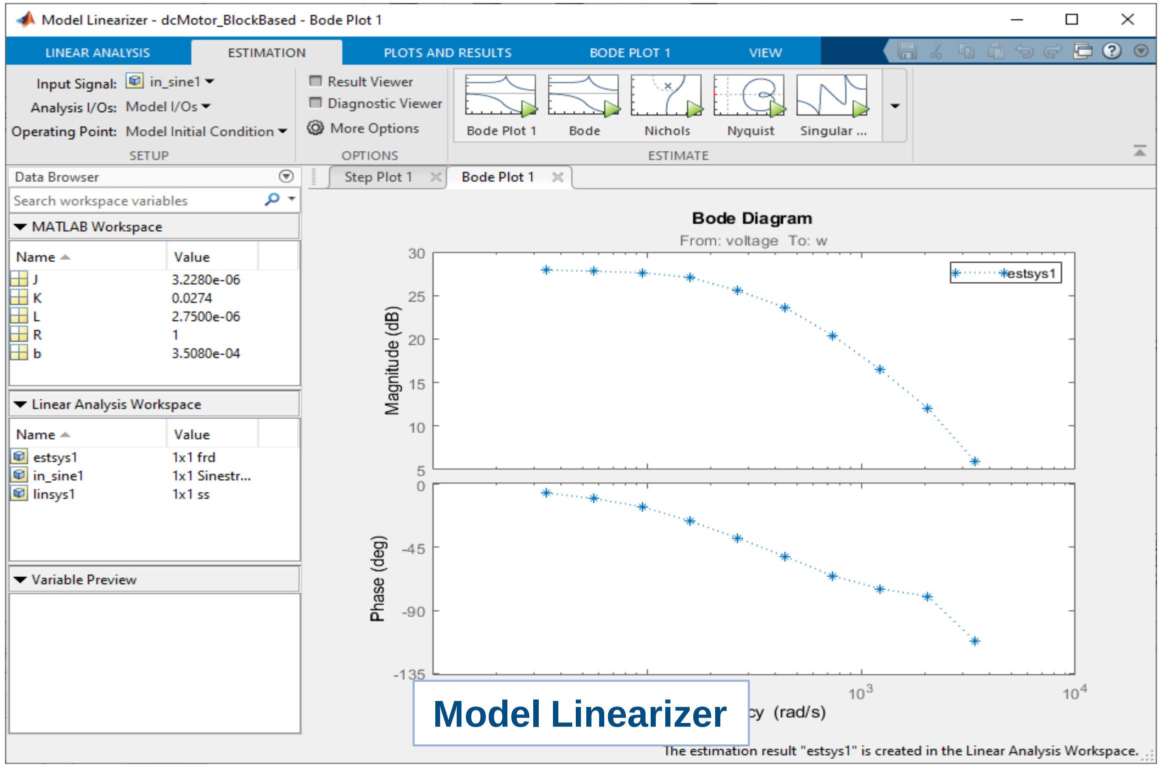


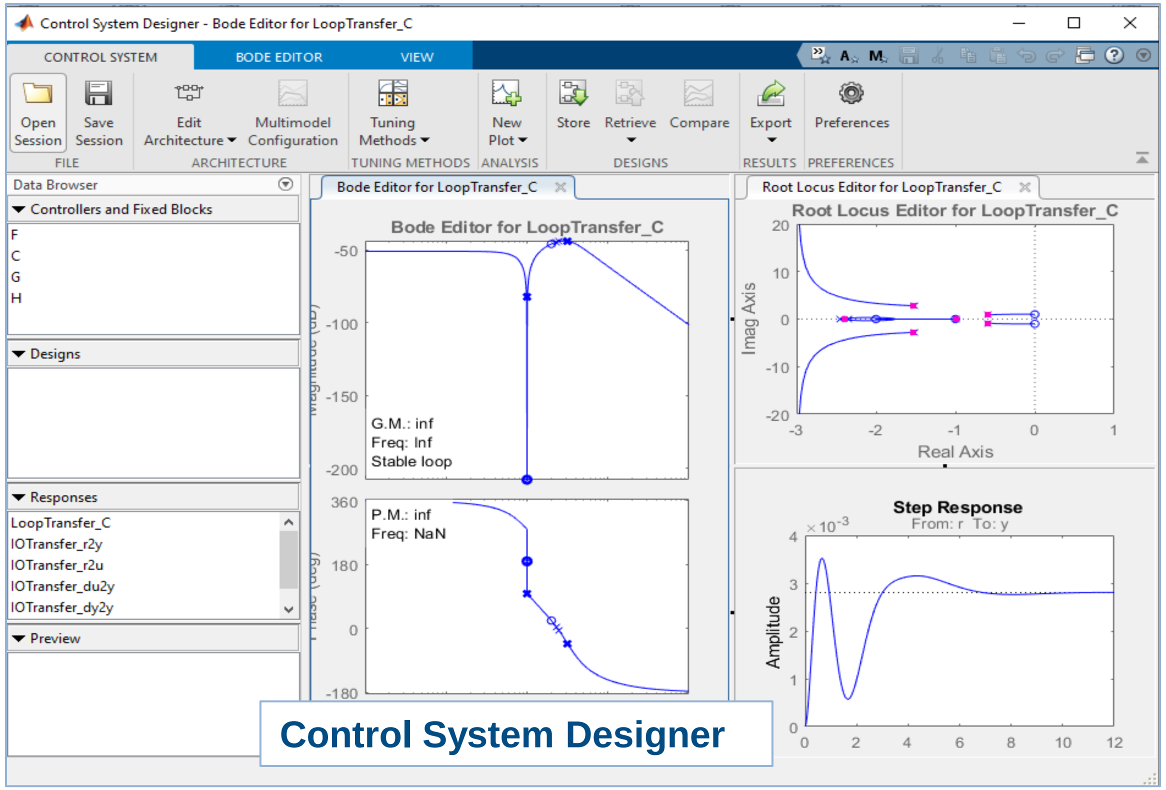
MATLAB EXPO

PID Tuning

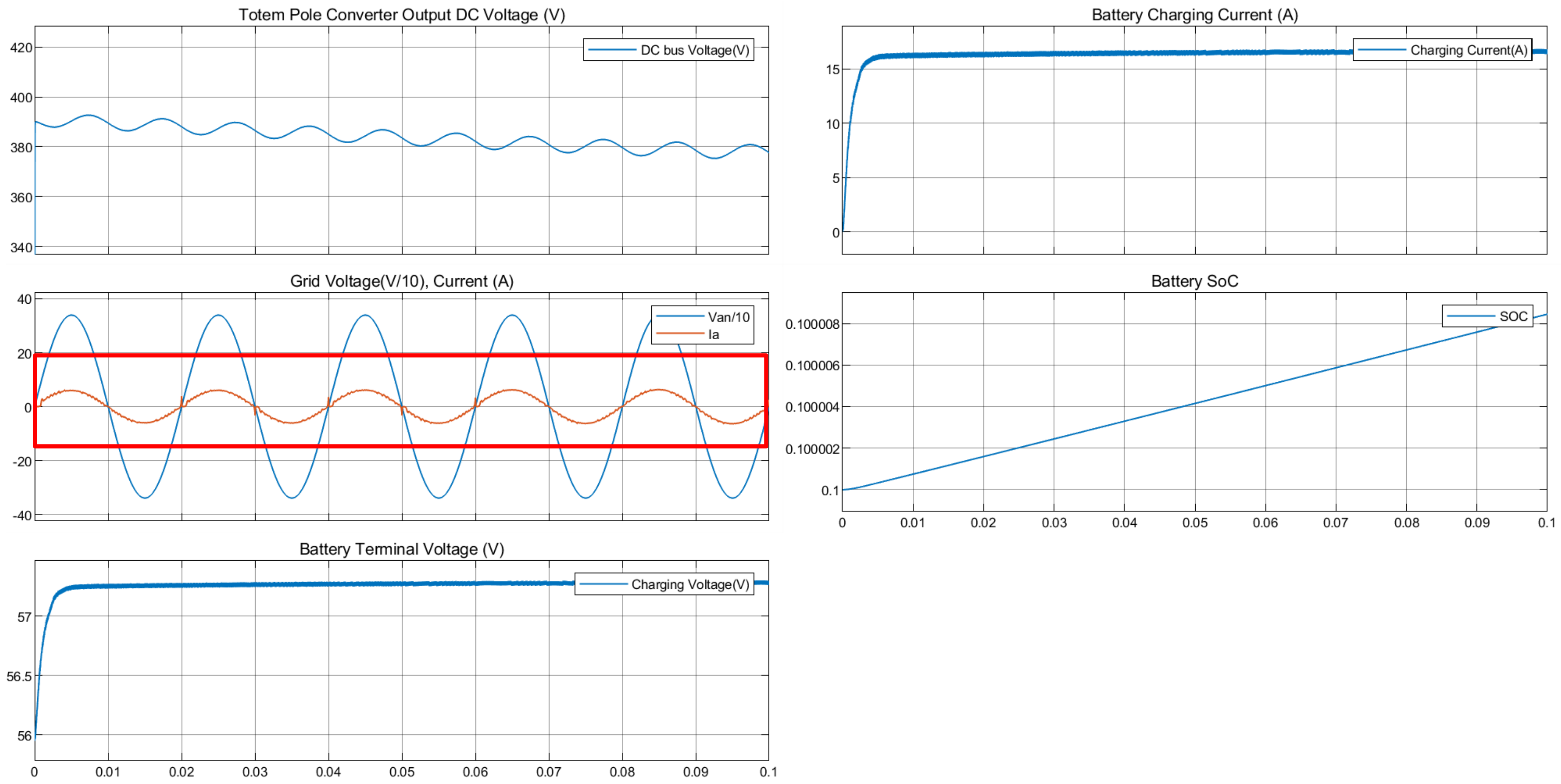


Compensator Design

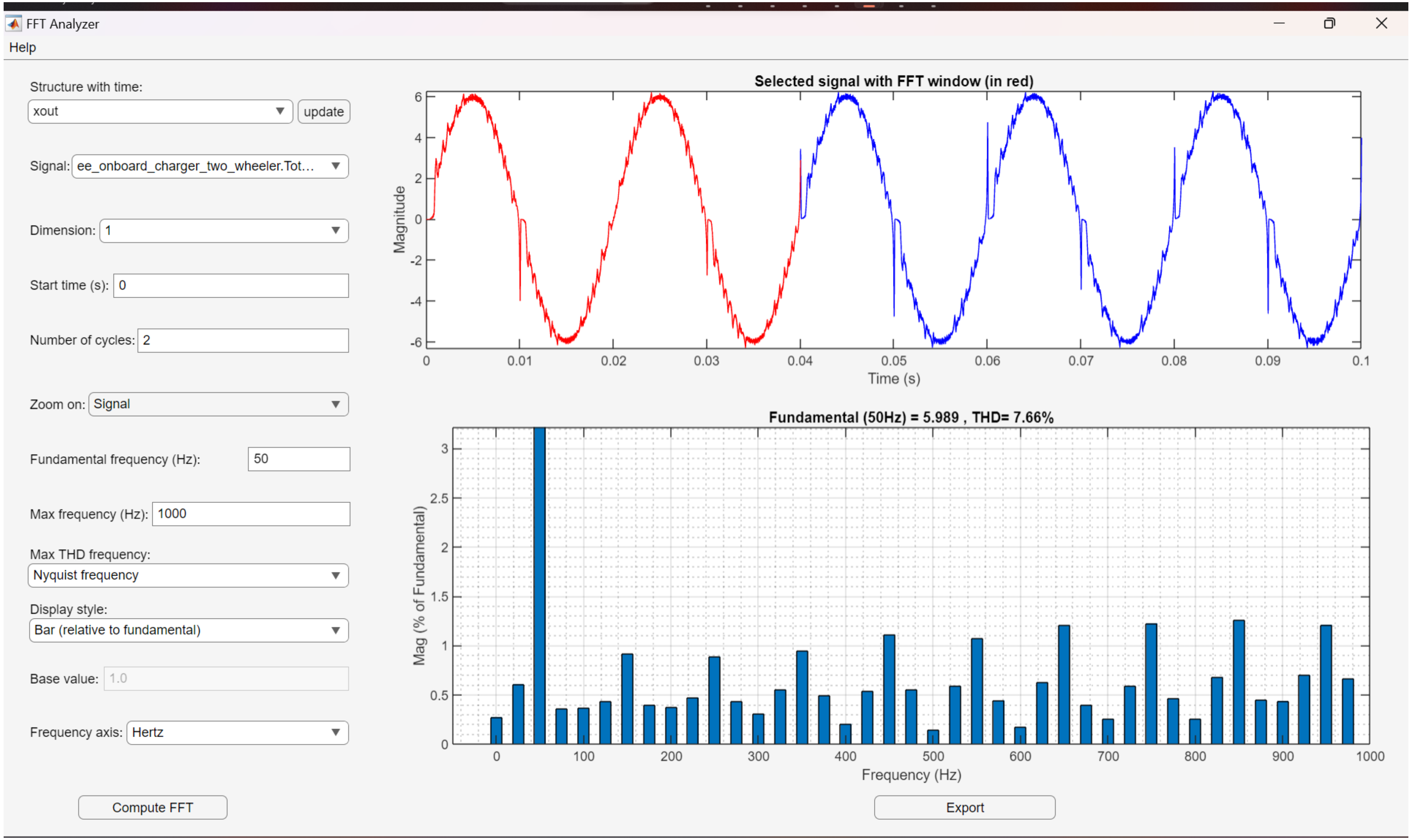




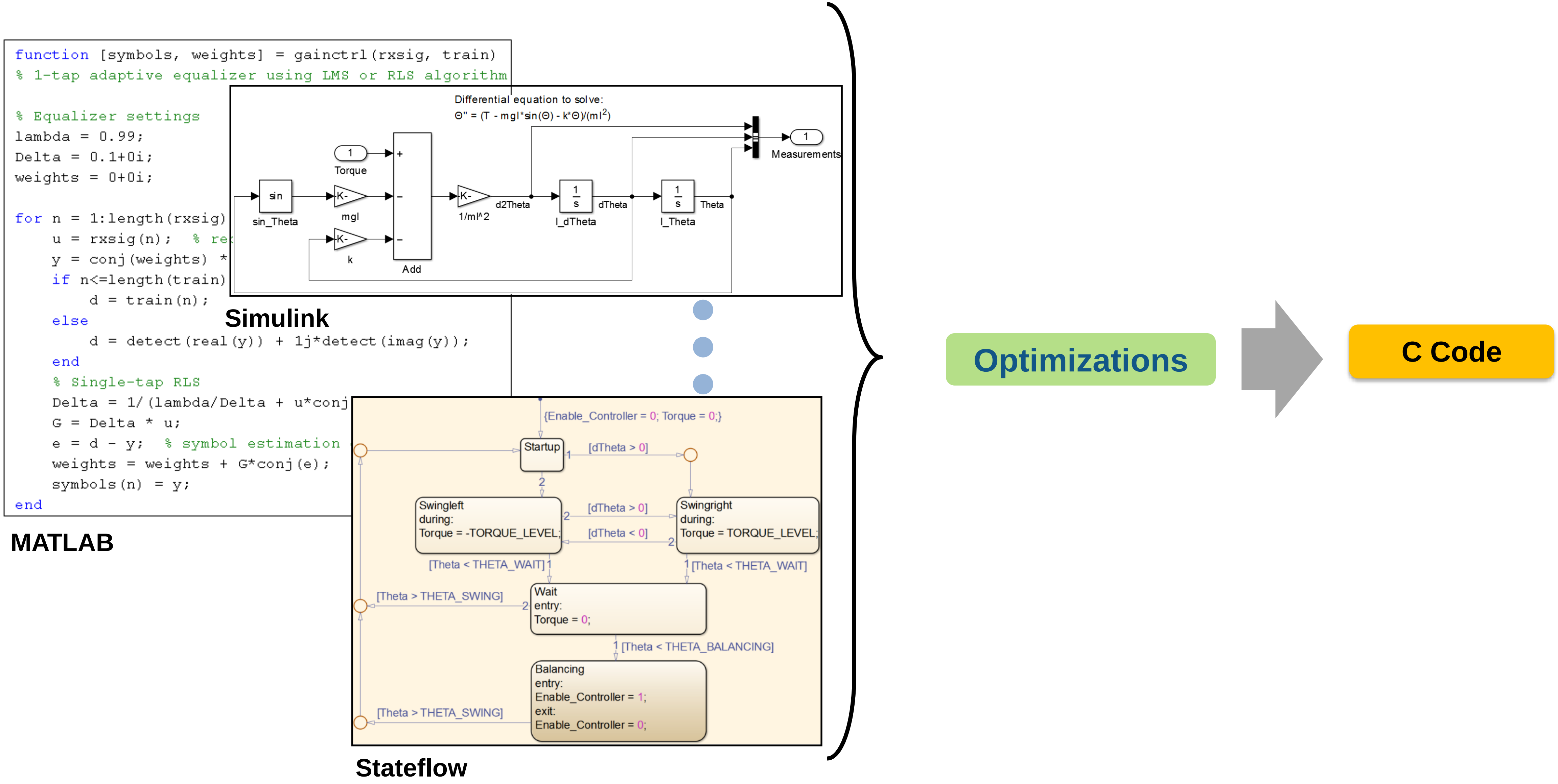
Simulation Output



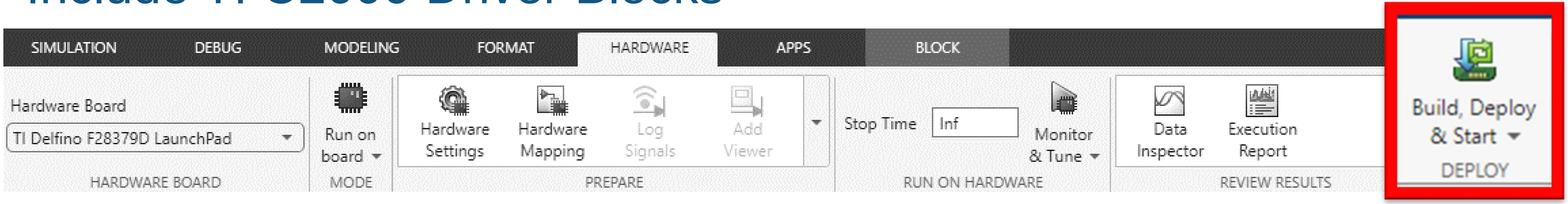
FFT Analyzer



Replace Hand Coding with Code Generation



Include TI C2000 Driver Blocks

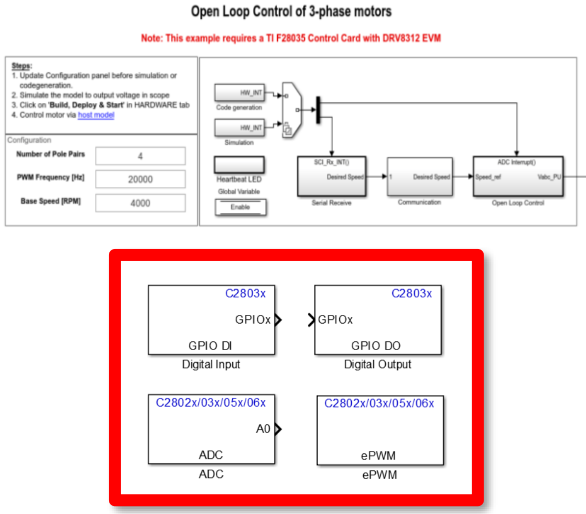


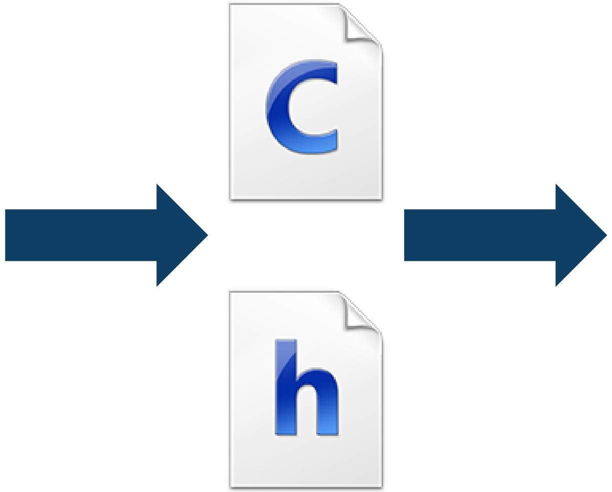
Simulink Model

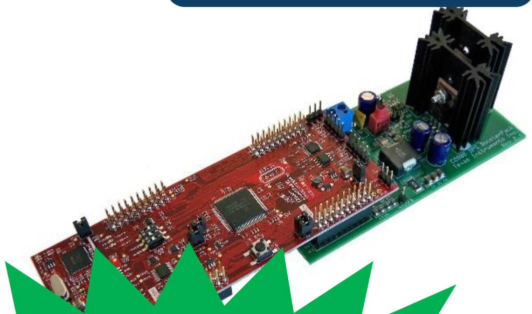
Embedded Coder

CCS Project

Run on C2000



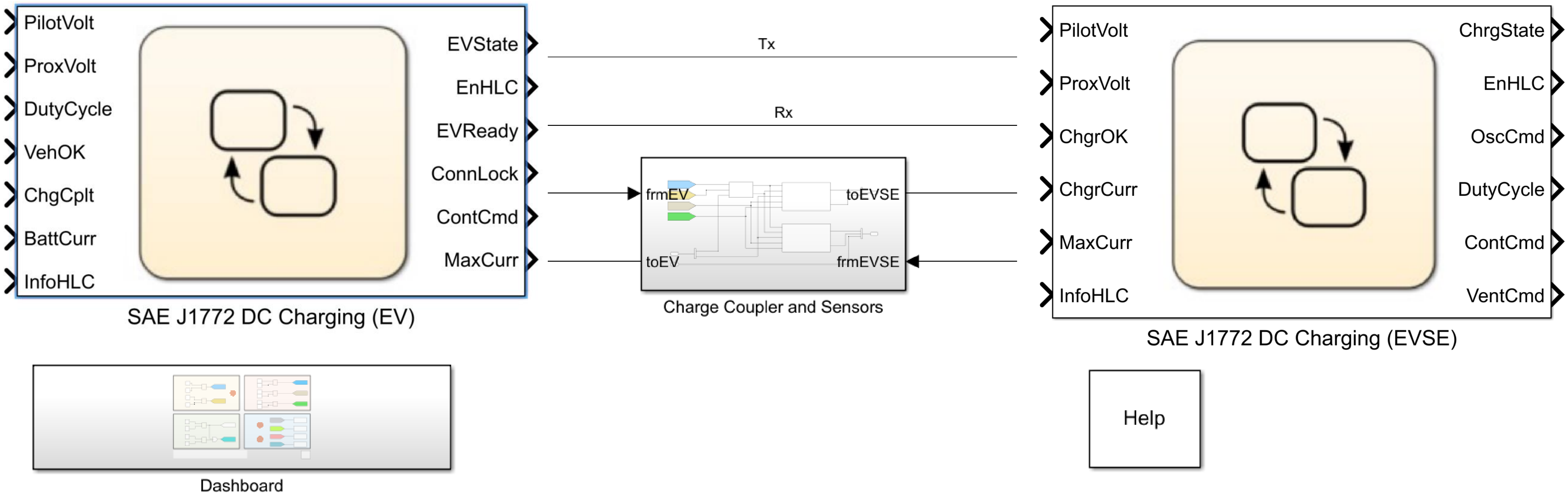




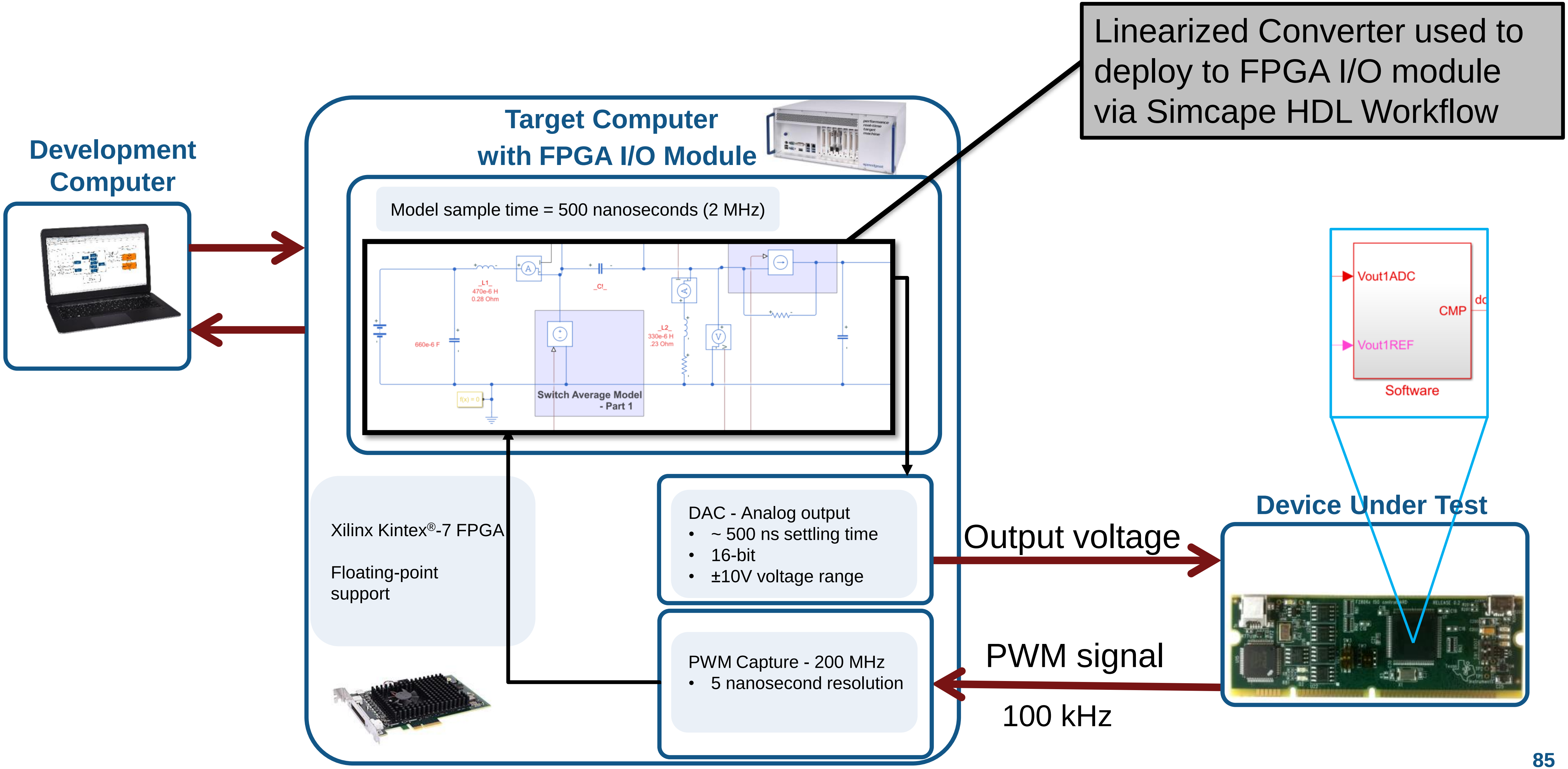
C2000 Blockset Introduced in R2023a

Test Digital Communication Protocols

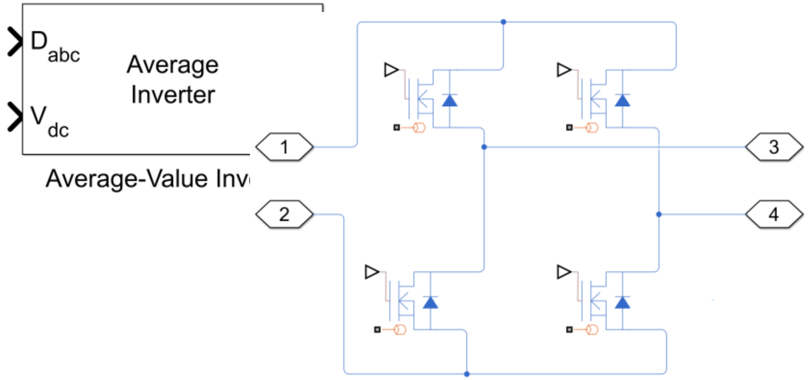
SAE J1772 protocol for basic charging and ISO15118 protocol for high level communication (HLC)



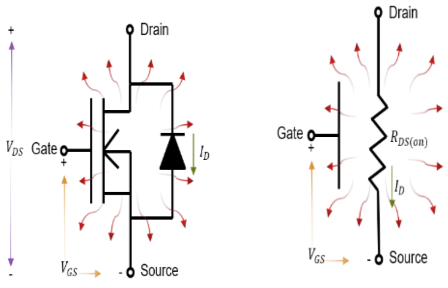
Hardware-in-the-Loop Simulation of Power Converter



Power Converter Control Design Workflow Tasks



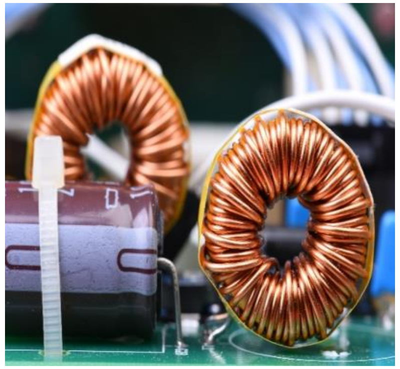
Model Fidelity



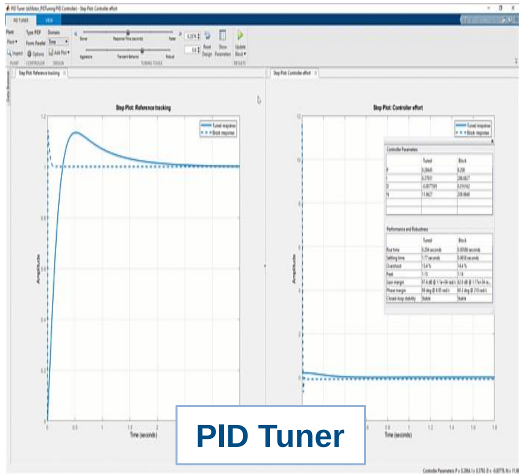
Switching loss

Conduction loss

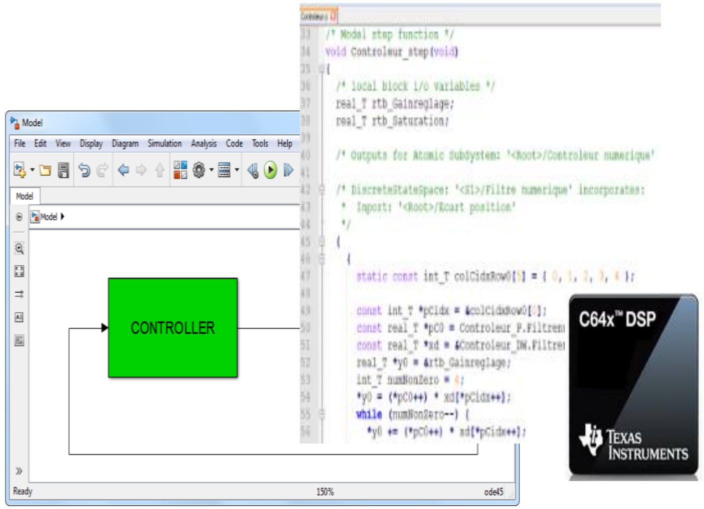
Losses Analysis



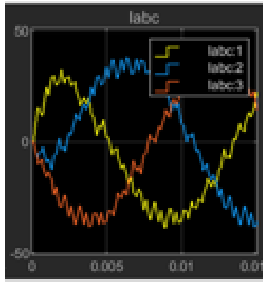
Size Components



Controller Design



One-Click Embedded Deployment

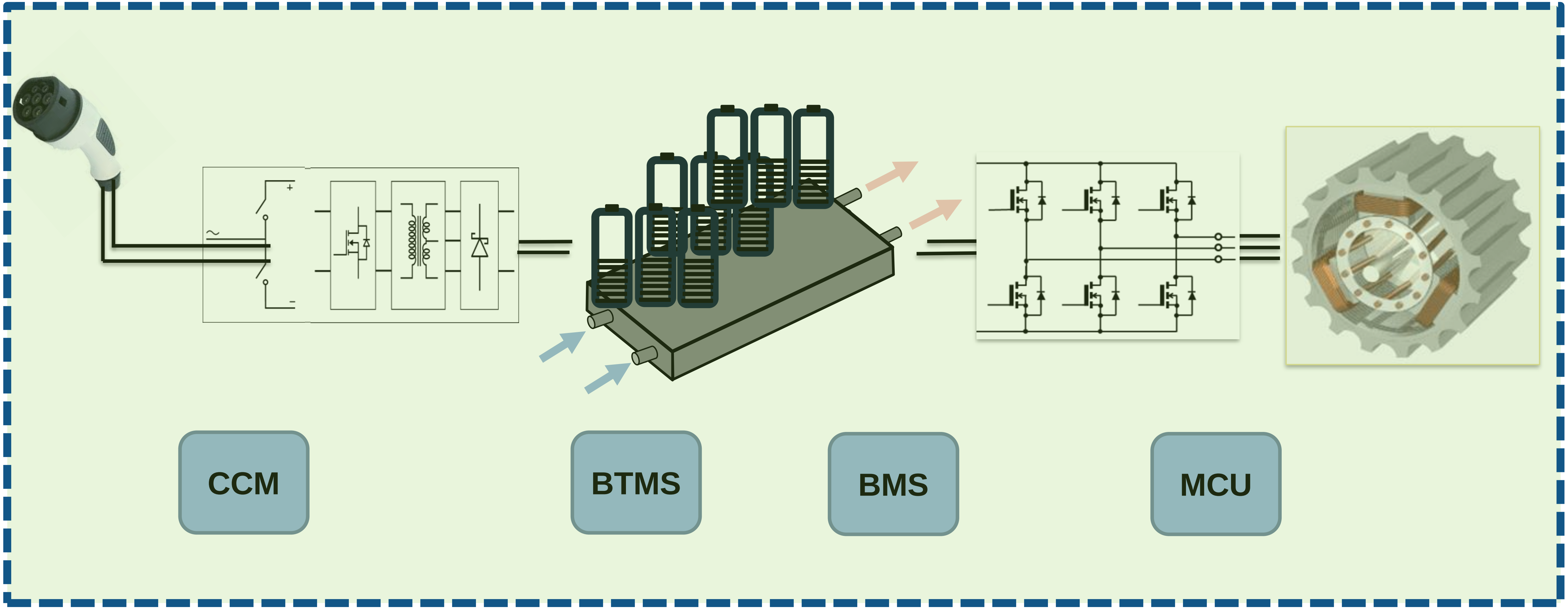


Analyze Harmonics

Challenges for Power Electronics Engineer

- Understand the impact of the power source and load
- Testing for a complete range of operating and fault conditions
- Designing and implementing digital controls using *only* SPICE simulator tools
- Catching errors during software-hardware integration testing
- Compliance to industry standards
- Development Time

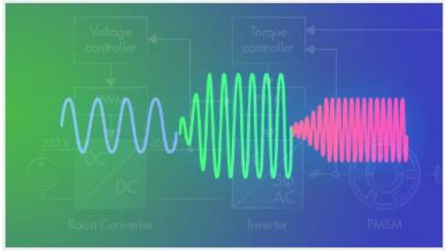
Electric Vehicle System



Key Takeaways

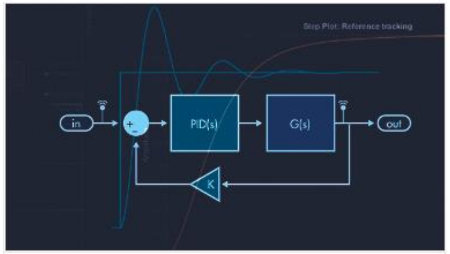
- **Speeding up journey from an idea to implementation!**
- **Iterating on new ideas faster using Model Based Design**
  - Design with simulation
  - Prototype on real-time hardware
  - Generate code for production
- **Varying model fidelity based on your needs**

Enable Your Team For Electrification



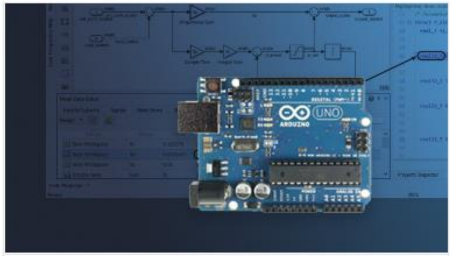
Power Electronics Control Design with Simulink and Simscape

Learn to model power electronic systems in the Simulink environment using Simscape Electrical™ and to design control with Simulink Control Design.



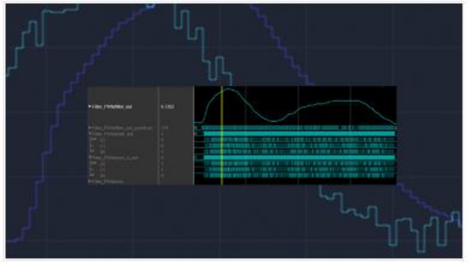
Control System Design with MATLAB and Simulink

Learn to design and model control systems with Simulink. Topics include system identification, parameter estimation, control system analysis, and response optimization.



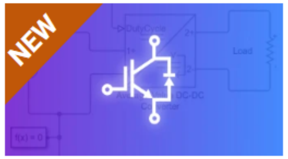
Embedded Coder for Production Code Generation

Develop Simulink models for deployment in embedded systems. Topics include code structure and execution, code generation options and optimizations, and deploying code to target hardware.



Generating HDL Code from Simulink

Learn to prepare Simulink models for HDL code generation, generate HDL code and testbench for a compatible Simulink model, and perform speed and area optimizations.



Power Electronics Simulation Onramp

5 modules | 1 hour | Languages

Learn the basics of simulating power electronics converters in Simscape.

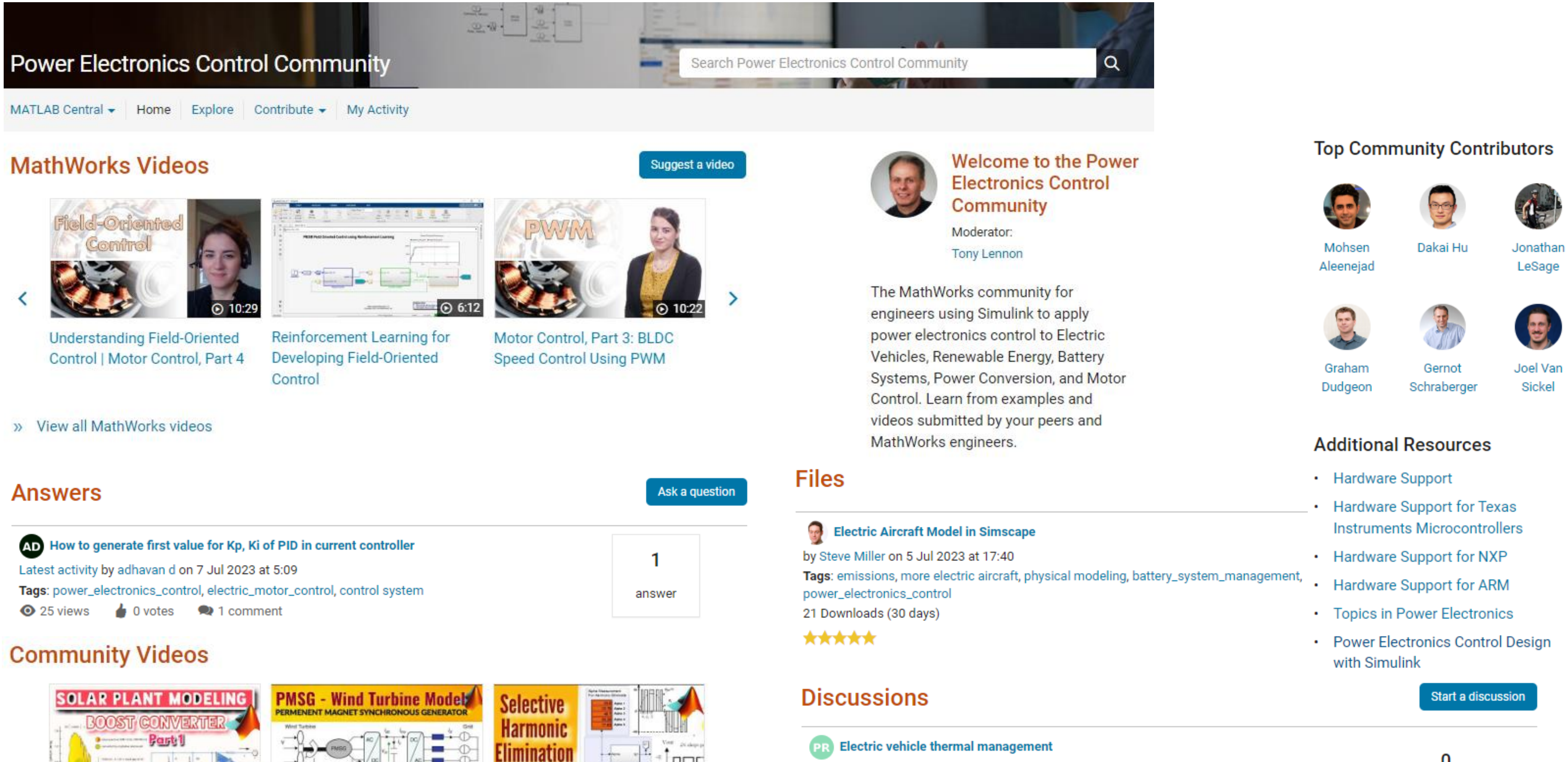
Circuit Simulation Onramp

7 modules | 2 hours | Languages

Learn the basics of simulating electrical circuits in Simscape.

Visit the Power Electronics Control Community on MATLAB Central to find Models, Answers, and How-to Videos

https://www.mathworks.com/matlabcentral/topics/power-electronics-control.html



MATLAB EXPO

Thank you

MathWorks®