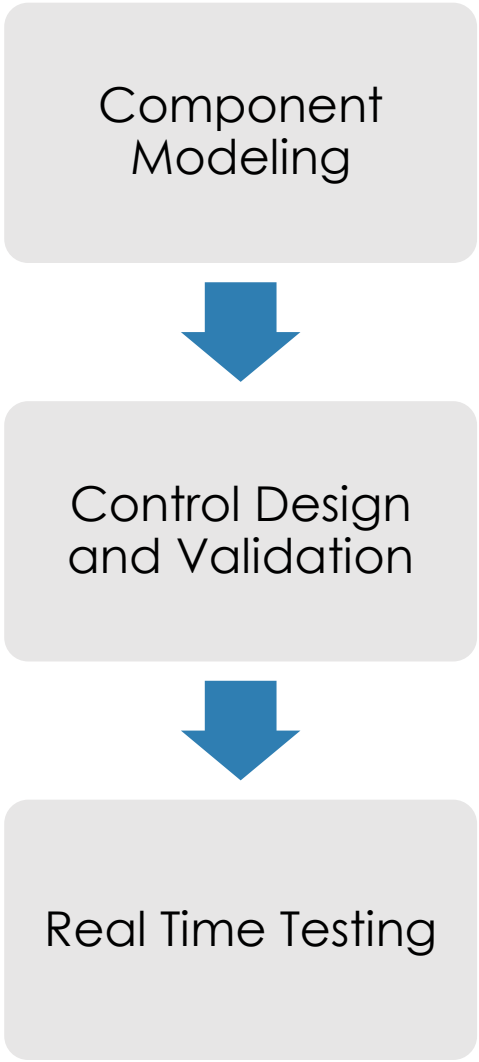
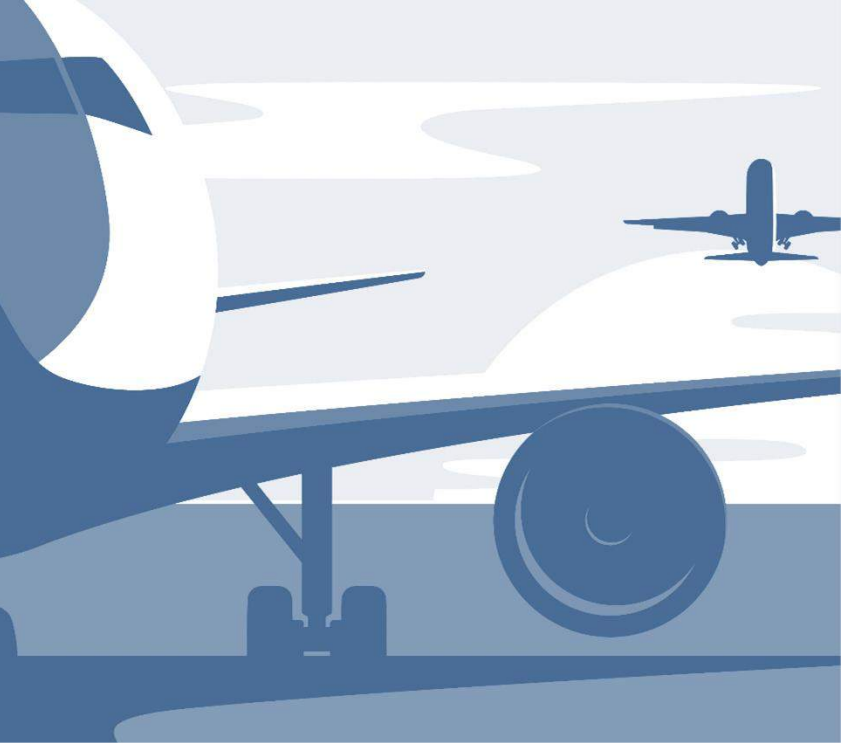
효율적인 수소 전기 추진 차량 개발을 위한 방법론

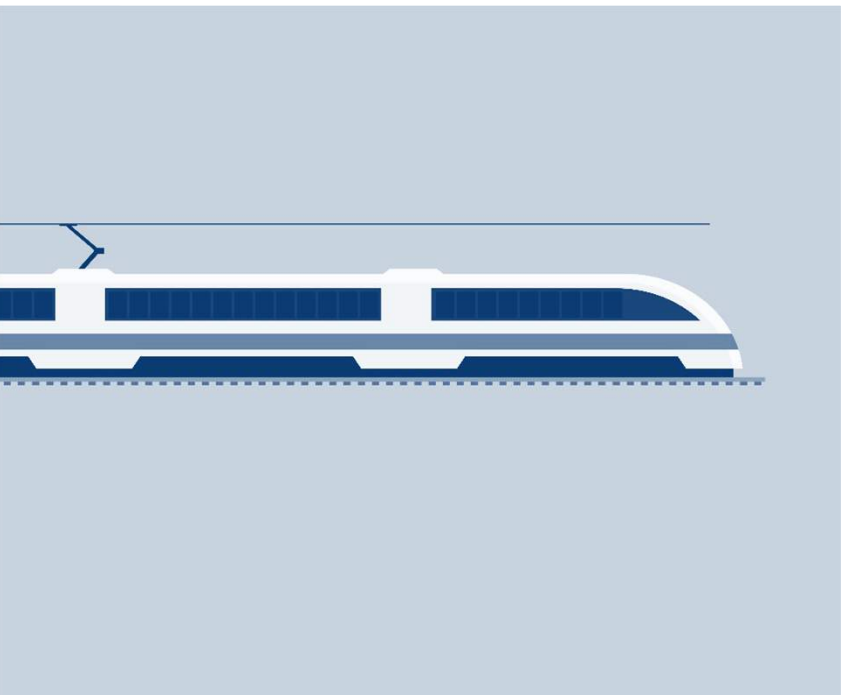
강효석 프로/Ph.D
Application Engineer

Agenda

- Multi-stack fuel cells and battery propulsion system model
- Supervisory logic development, verification and code generation
- Real-time testing and prototyping







Nuvera Develops Hydrogen Fuel Cells That Reduce CO_2 Emissions

User Story

Challenge

Reduce the levels of air pollution caused by emissions from commercial diesel engines at seaports

Solution

Design the control software for a hydrogen fuel cell engine and model the engine with MATLAB and Simulink

Key Outcomes

- The fuel cell produces no exhaust other than heat and water and provides long-range and quick refueling.
- One fuel cell–powered container handler reduces emissions by 128 metric tons of CO_2 annually.
- The system maintains an ideal level of battery charge to handle peak load and manages capacity to reabsorb energy

A Nuvera E-Series Fuel Cell Engine.

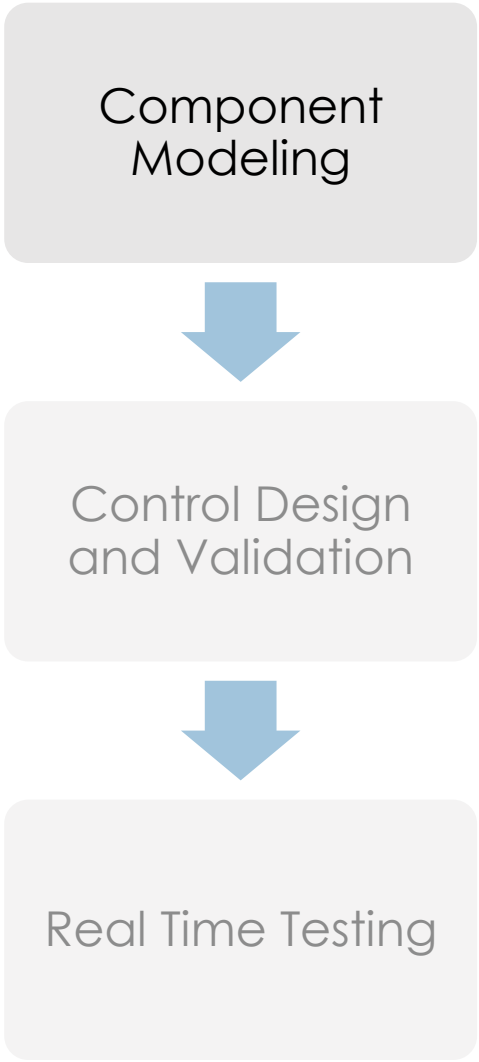
"Fuel cells are better than batteries whenever long range is required, or when battery charging takes too long—making them good for boats, planes, trucks, buses, and emergency response vehicles."

- Gus Block, Nuvera Fuel Cells

Link to user story

Agenda

- **Multi-stack fuel cells and battery propulsion system model**
- Supervisory logic development, verification and code generation
- Real-time testing and prototyping



Fuel Cell and Battery propulsion System

Motivations

Validate behavior against requirements

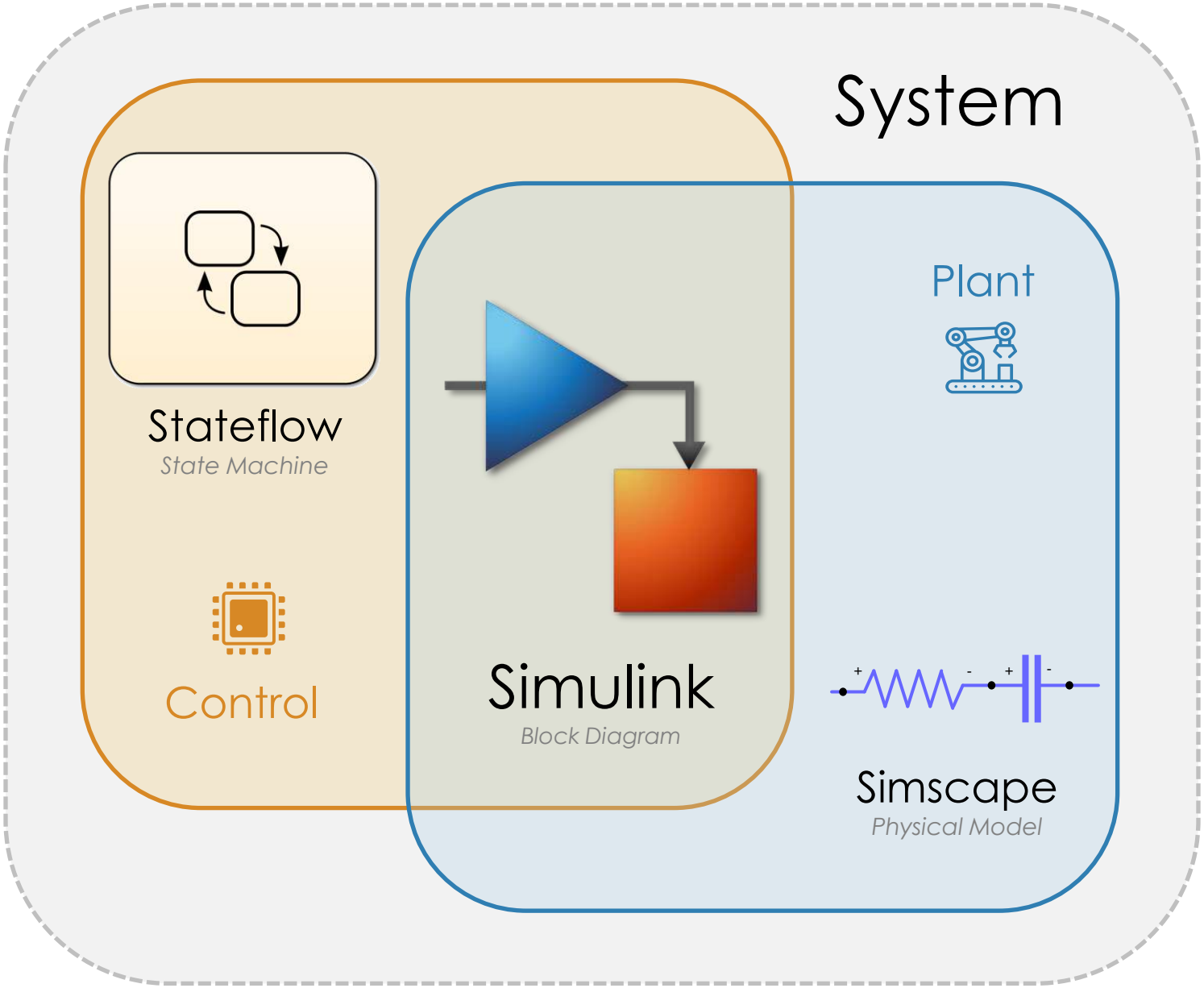
Model simulation
Results analysis

Optimize your design and calibration

Design Space Exploration
Trade Off Analysis

System-Level Model with a single simulation platform

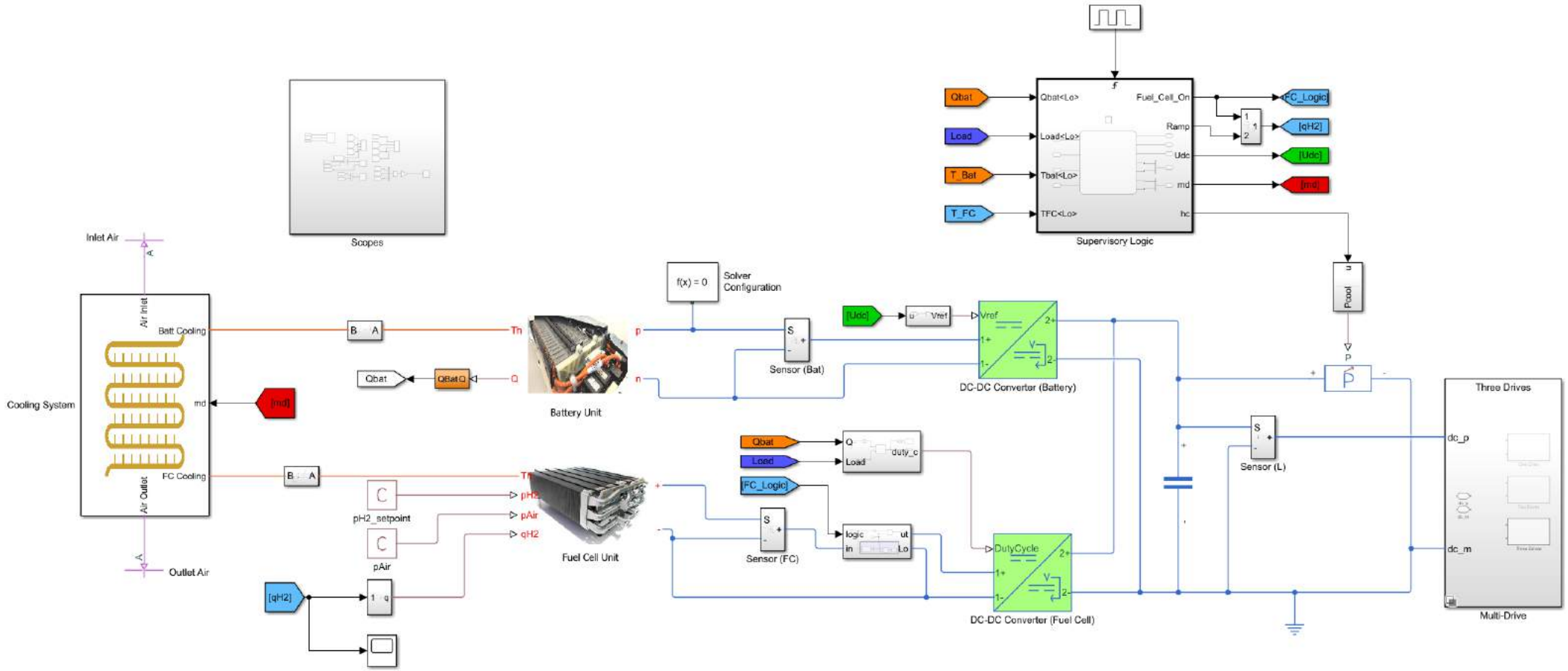
Ease of use


System
Stateflow
State Machine
Control
Simulink
Block Diagram
Plant
Simscape
Physical Model

Run

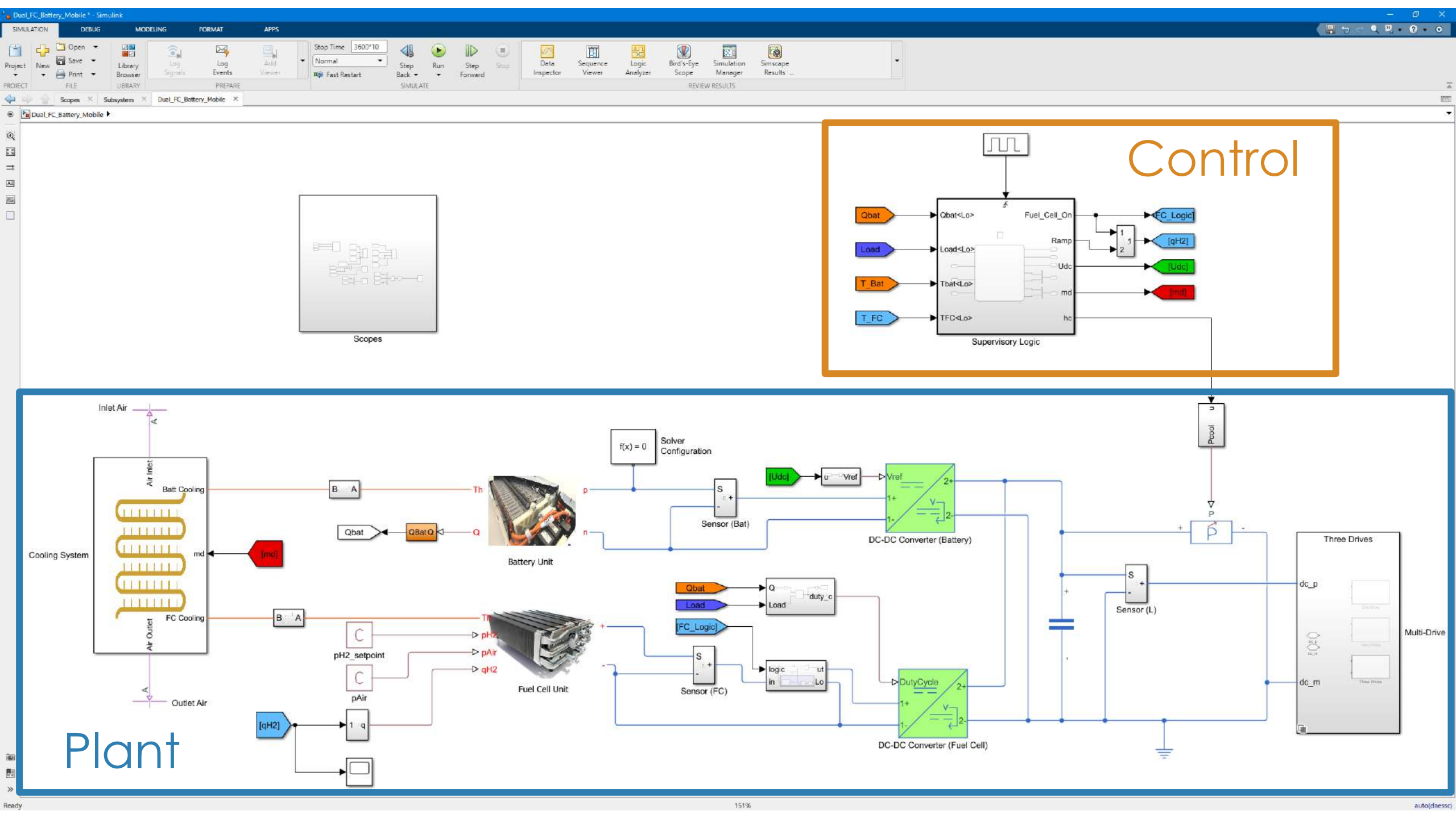
Multi-stack fuel cells and battery propulsion system model

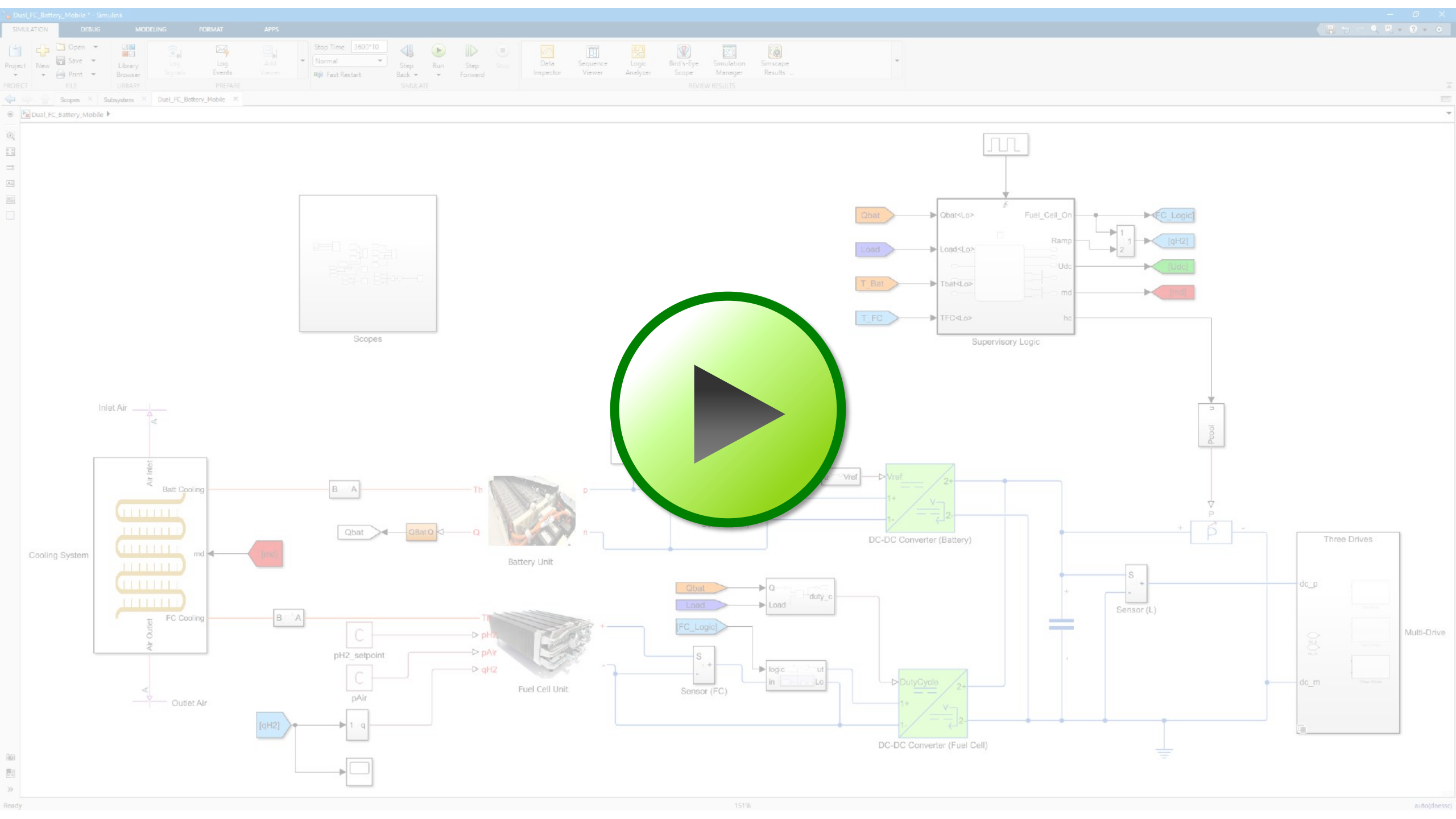
With Simulink

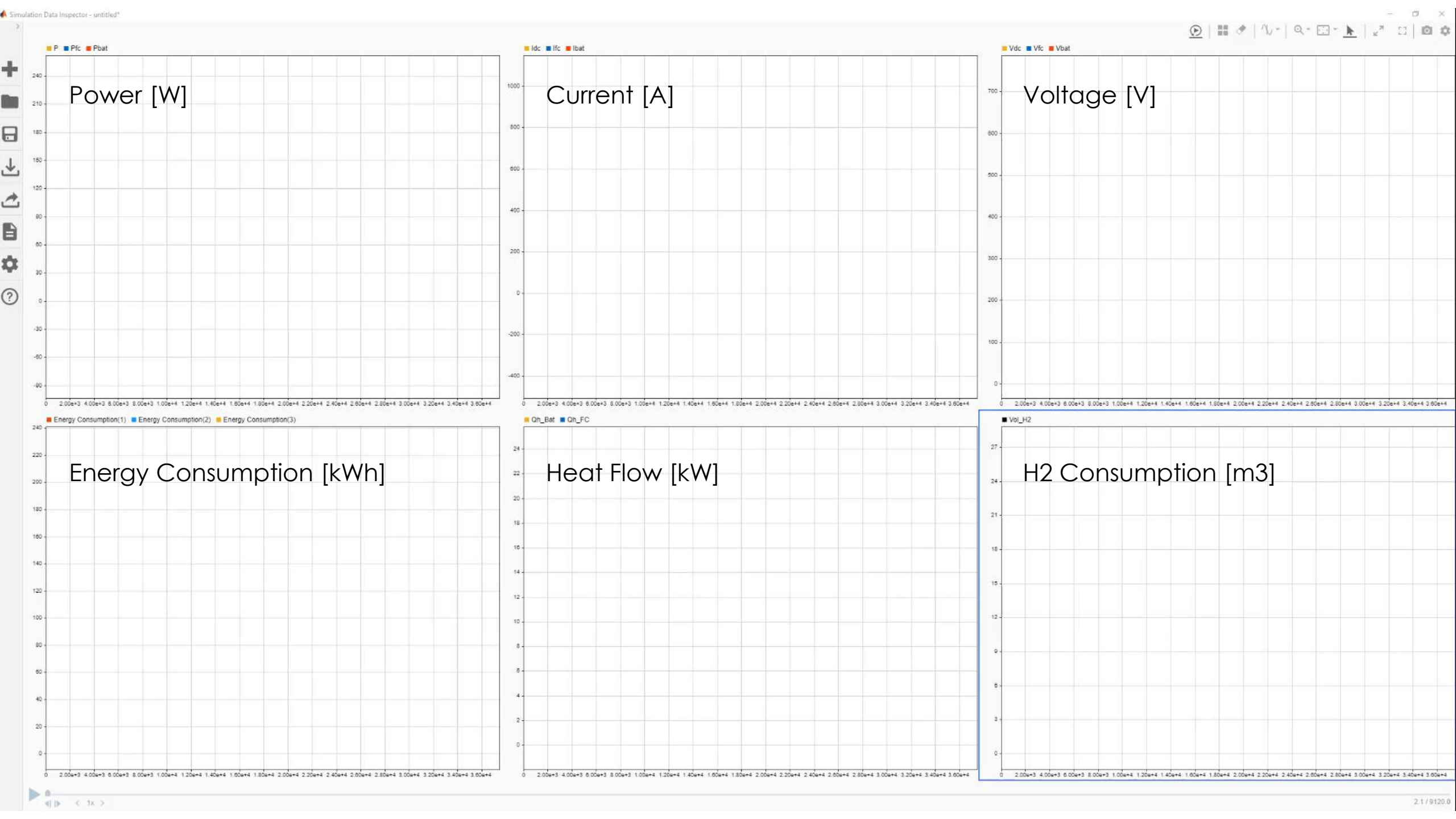


Multi-stack fuel cells and battery propulsion system

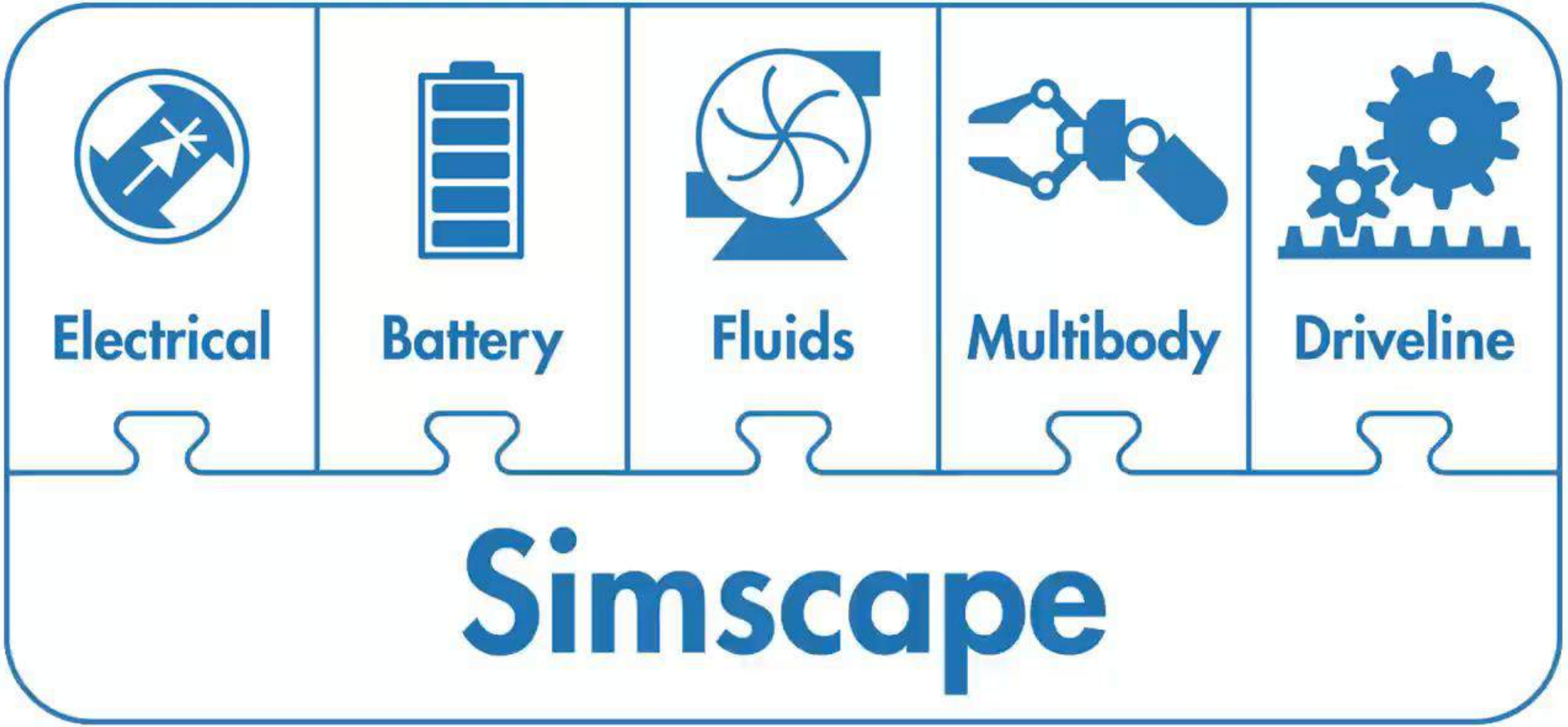
MathWorks - 2022


Dual_FC_Battery_Mobile * - Simulink
SIMULATION
DEBUG
MODELING
FORMAT
APPS
Stop Time
3600*10
Normal
Fast Restart
Step Back
Run
Step Forward
Stop
Data Inspector
Sequence Viewer
Logic Analyzer
Bird's-Eye Scope
Simulation Manager
Simscape Results ...
Scopes
Subsystem
Dual_FC_Battery_Mobile
Scopes
Control
Qbat
Load
T_Bat
T_FC
Qbat<Lo>
Load<Lo>
Tbat<Lo>
TFC<Lo>
Fuel_Cell_On
Ramp
Udc
md
hc
FC_Logic
[qH2]
[Udc]
[md]
Supervisory Logic
Plant
Inlet Air
Air Inlet
Batt Cooling
Cooling System
md
[md]
FC Cooling
Air Outlet
Outlet Air
B A
Th
Qbat
QBat Q
Battery Unit
f(x) = 0
Solver Configuration
p
n
Sensor (Bat)
[Udc]
u Vref
Vref
DC-DC Converter (Battery)
Pcool
P
Three Drives
dc_p
dc_m
Multi-Drive
Sensor (L)
Qbat
Load
[FC_Logic]
Q
Load
duty_c
Sensor (FC)
logic
in
ut
Lo
DutyCycle
DC-DC Converter (Fuel Cell)
pH2_setpoint
pAir
[qH2]
pH2
pAir
qH2
Fuel Cell Unit
Ready
151%
auto(daessc)


Dual_FC_Battery_Mobile * - Simulink
SIMULATION
DEBUG
MODELING
FORMAT
APPS
Scopes
Subsystem
Dual_FC_Battery_Mobile
Scopes
Supervisory Logic
Qbat
Load
T_Bat
T_FC
Qbat<Lo>
Load<Lo>
Tbat<Lo>
TFC<Lo>
Fuel_Cell_On
Ramp
Udc
md
hc
FC_Logic
[qH2]
[Udc]
[md]
Inlet Air
Outlet Air
Cooling System
Air Inlet
Air Outlet
Batt Cooling
FC Cooling
Battery Unit
Fuel Cell Unit
DC-DC Converter (Battery)
DC-DC Converter (Fuel Cell)
Vref
DutyCycle
pH2_setpoint
pAir
Sensor (FC)
Sensor (L)
Three Drives
Multi-Drive
dc_p
dc_m
Pcool
Ready
151%
auto(doesso)


Simulation Data Inspector - untitled*
P
Pfc
Pbat
Power [W]
Idc
Ifc
Ibat
Current [A]
Vdc
Vfc
Vbat
Voltage [V]
Energy Consumption(1)
Energy Consumption(2)
Energy Consumption(3)
Energy Consumption [kWh]
Qh_Bat
Qh_FC
Heat Flow [kW]
Vol_H2
H2 Consumption [m3]

Simscape Products for Physical Modeling

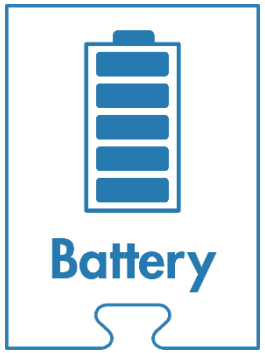


Simscape Products

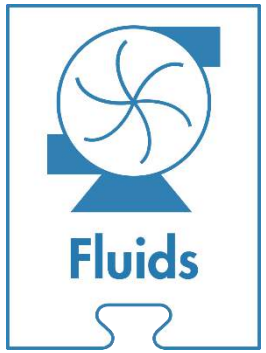
- Components in 10 domains
- Physical modeling language
- Core simulation technology
- Logging and analysis tools
- Model sharing options

- Semiconductors
- Electric Drives
- Power Converters
- Sensors, Logic

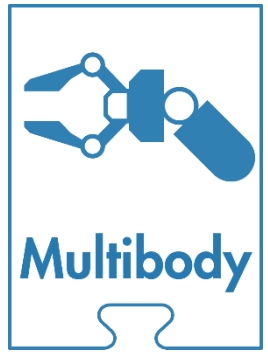
- Gears, Belts
- Tires, Engines
- Clutches
- Transmissions



- Battery Pack Builder
- Cooling Plates
- Battery Management
- Support for HIL Tests



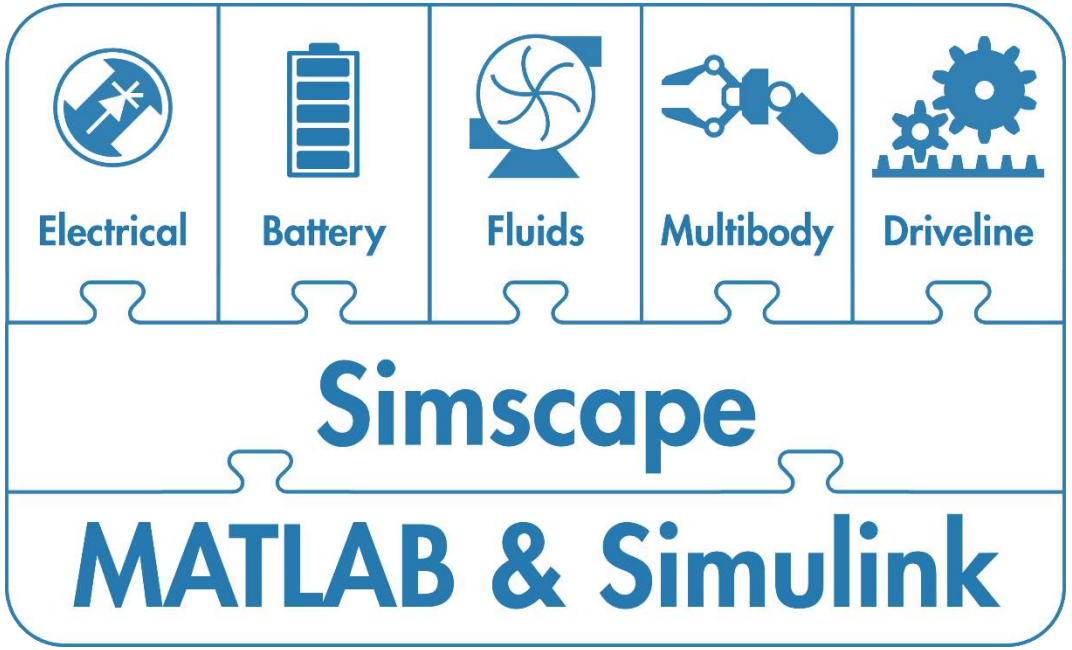
- Fluid Power
- Hydraulic Actuation
- Heating and Cooling
- Fluid Transportation

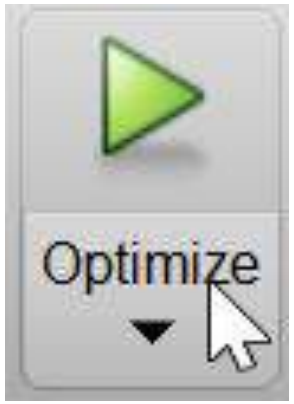


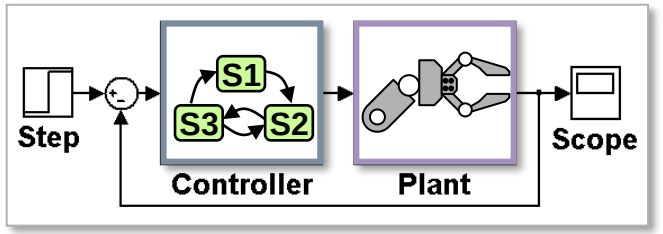
- 3D Dynamics
- Kinematics
- Force Analysis
- CAD Import

MATLAB, Simulink, & Simscape

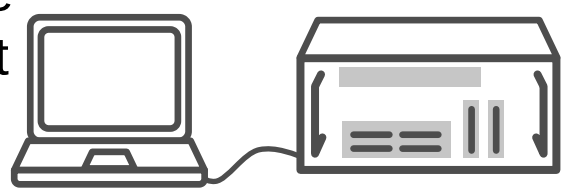
- Get more value from your model of the physical system
- MATLAB
  - Automate any task (build, test, analyze)
  - Streamline testing (parallel computing)
- Simulink
  - Design and test algorithms
  - Test embedded software without hardware prototypes







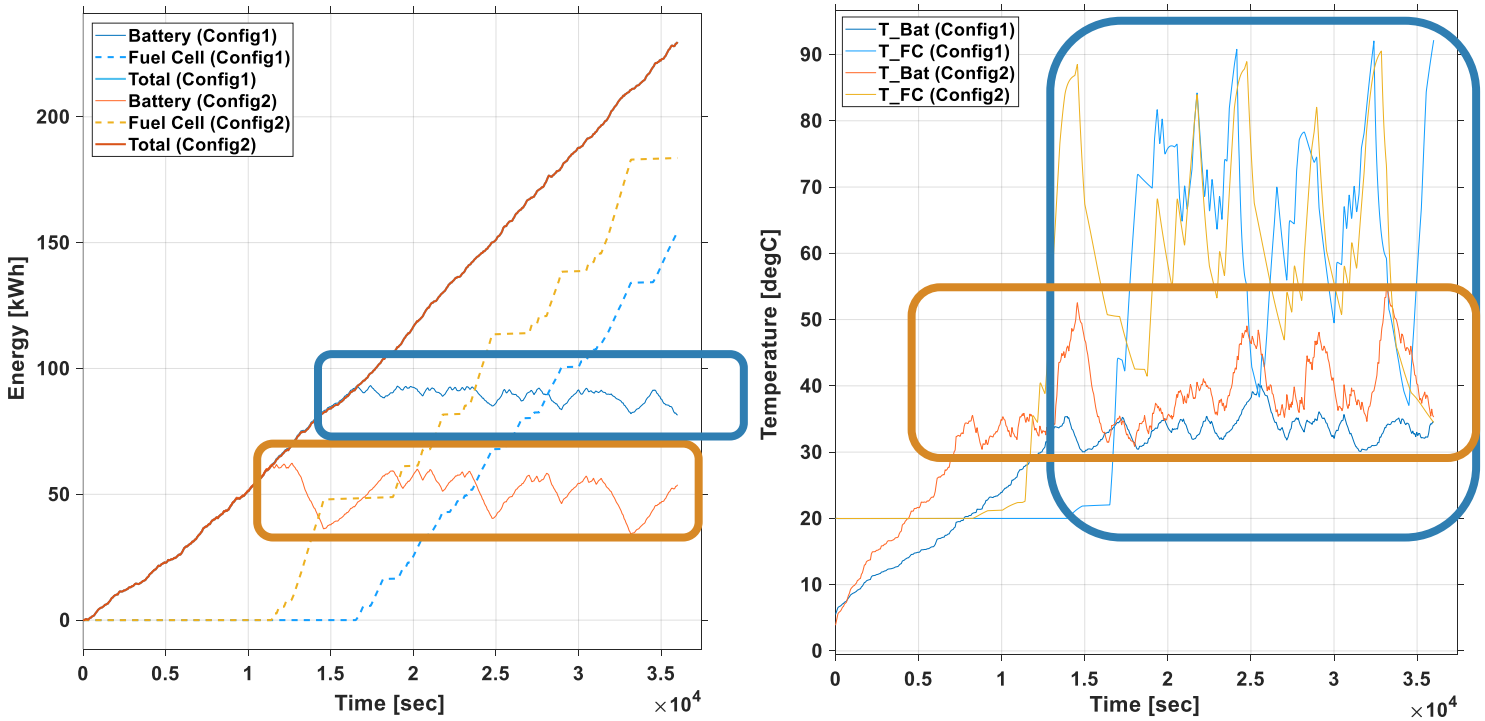
Integrate and Test



Design Activities Based On Simulation

Architecture Evaluation

- Explore Design Space
  - Example: **3 Battery Modules 2 Fuel Cell Stacks** vs **2 Battery Modules 3 Fuel Cell Stacks**

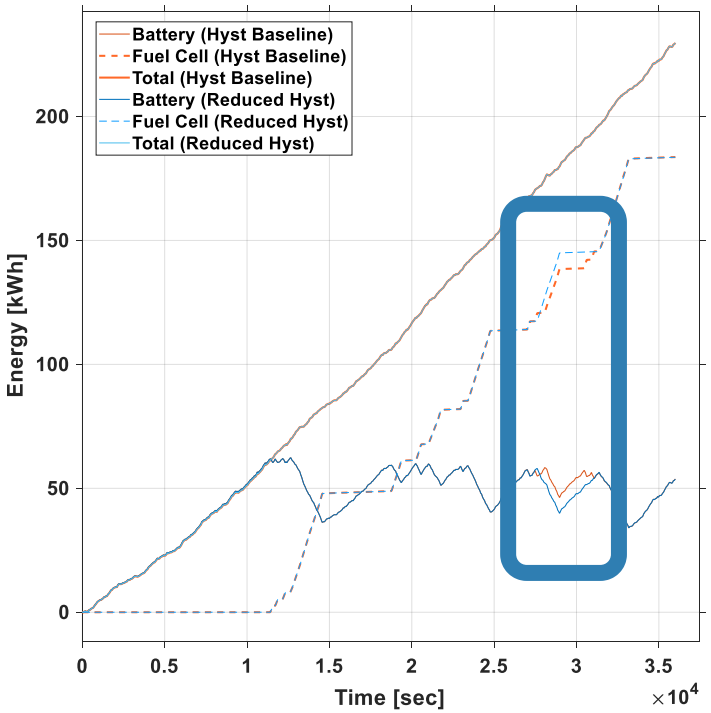


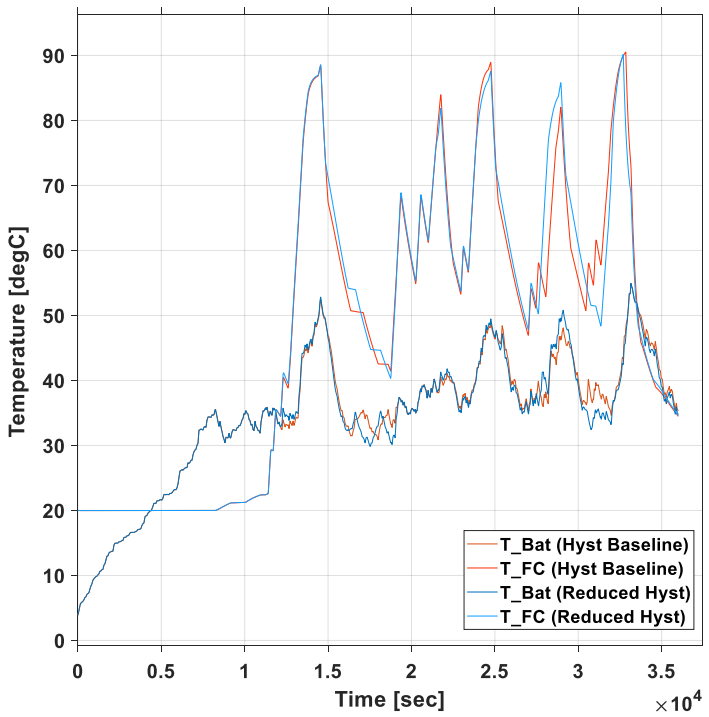
- Architecture Evaluation
- Design Optimization
- Component Requirement
- Component Validation
- System Evaluation
- Controller Calibration
- ...

Design Activities Based On Simulation

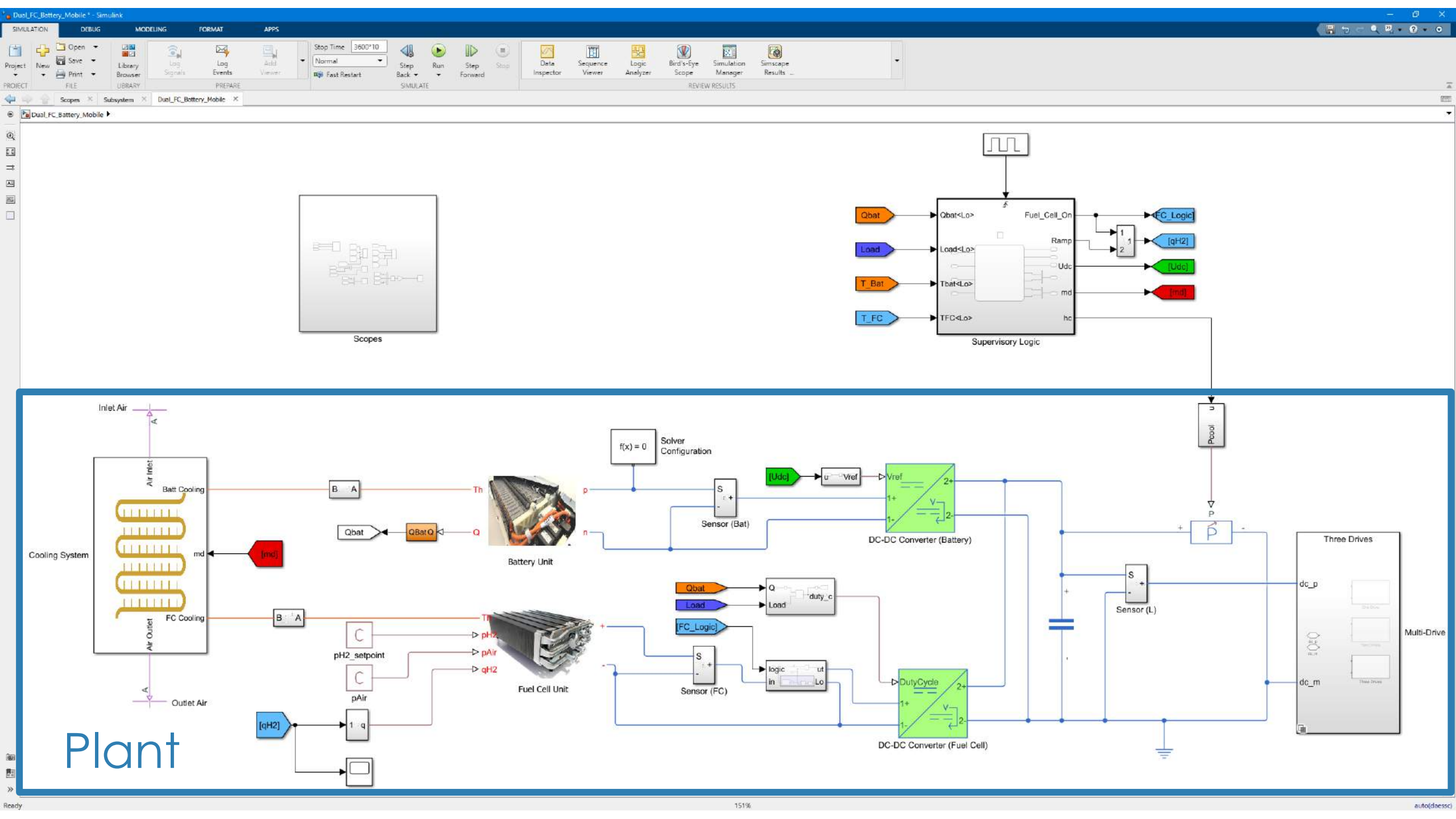
Controller Calibration

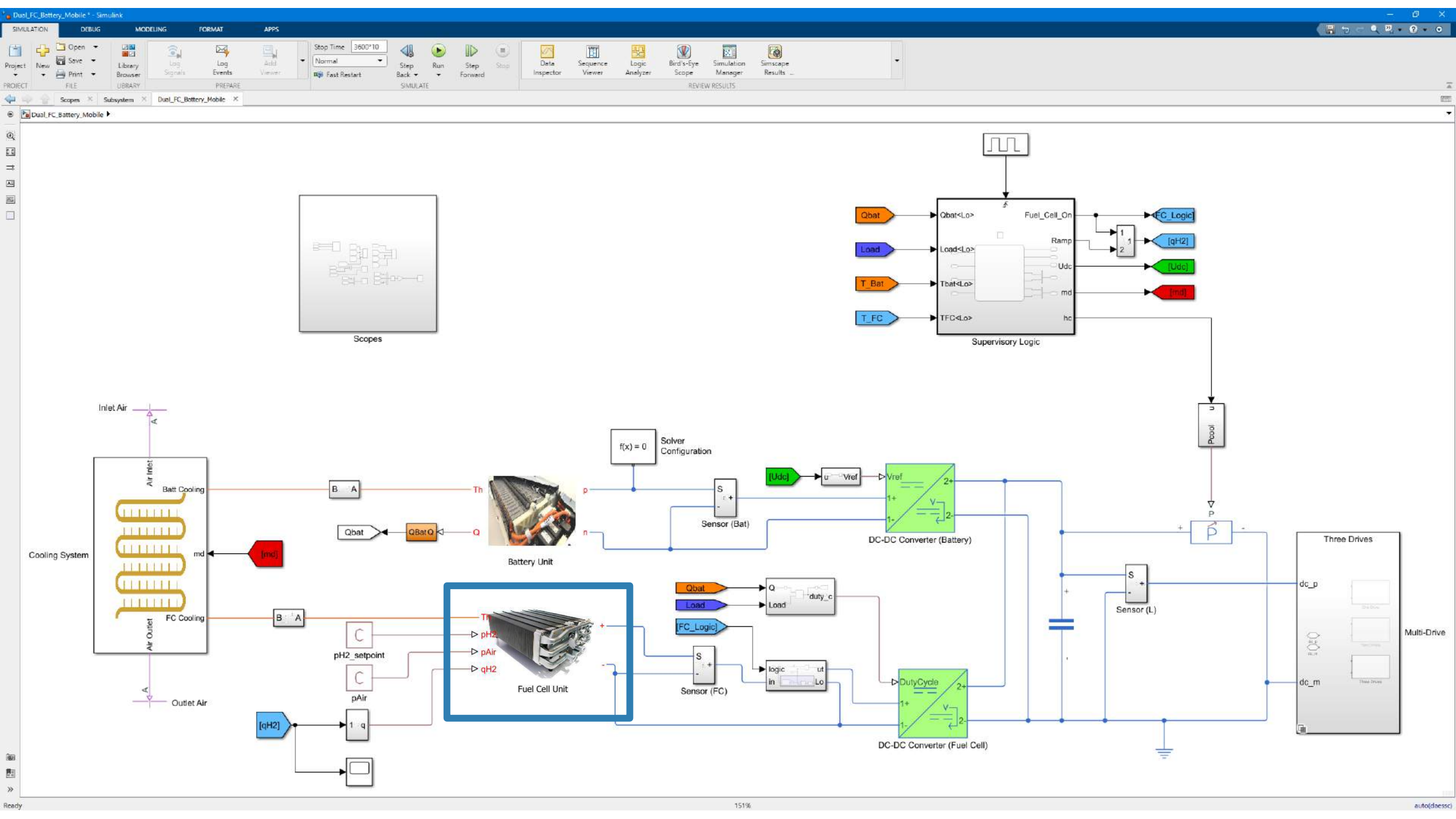
- Compare Cooling Control Settings
  - Example: **Cooling with Baseline Hysteresis** vs **Cooling with Reduced Hysteresis**





- Architecture Evaluation
- Design Optimization
- Component Requirement
- Component Validation
- System Evaluation
- **Controller Calibration**
- ...


Dual_FC_Battery_Mobile * - Simulink
SIMULATION
DEBUG
MODELING
FORMAT
APPS
Scopes
Subsystem
Dual_FC_Battery_Mobile
Scopes
Qbat
Load
T_Bat
T_FC
Qbat<Lo>
Load<Lo>
Tbat<Lo>
TFC<Lo>
Fuel_Cell_On
Ramp
Udc
md
hc
FC_Logic
[qH2]
[Udc]
[md]
Supervisory Logic
Inlet Air
Cooling System
Air Inlet
Batt Cooling
FC Cooling
Air Outlet
Outlet Air
[md]
Th
Qbat
QBat Q
Q
Battery Unit
f(x) = 0
Solver Configuration
Sensor (Bat)
[Udc]
Vref
DC-DC Converter (Battery)
Pcool
P
Sensor (L)
Three Drives
dc_p
dc_m
Multi-Drive
pH2_setpoint
pAir
[qH2]
pH2
pAir
qH2
Fuel Cell Unit
Qbat
Load
[FC_Logic]
duty_c
Sensor (FC)
logic
DutyCycle
DC-DC Converter (Fuel Cell)
Plant
Ready
151%
auto(daessc)


Dual_FC_Battery_Mobile * - Simulink
SIMULATION
DEBUG
MODELING
FORMAT
APPS
Stop Time
3600*10
Normal
Fast Restart
Step Back
Run
Step Forward
Stop
Data Inspector
Sequence Viewer
Logic Analyzer
Bird's-Eye Scope
Simulation Manager
Simscape Results ...
Scopes
Subsystem
Dual_FC_Battery_Mobile
Scopes
Qbat
Load
T_Bat
T_FC
Qbat<Lo>
Load<Lo>
Tbat<Lo>
TFC<Lo>
Fuel_Cell_On
Ramp
Udc
md
hc
FC_Logic
[qH2]
[Udc]
[md]
Supervisory Logic
Pcool
Inlet Air
Air Inlet
Batt Cooling
Cooling System
md
FC Cooling
Air Outlet
Outlet Air
[md]
B A
Th
Qbat
QBat Q
Battery Unit
f(x) = 0
Solver Configuration
p
n
Sensor (Bat)
[Udc]
u Vref
Vref
DC-DC Converter (Battery)
P
Sensor (L)
Three Drives
dc_p
dc_m
Multi-Drive
pH2_setpoint
pAir
[qH2]
Th
pH2
pAir
qH2
Fuel Cell Unit
Qbat
Load
[FC_Logic]
Q
duty_c
Load
Sensor (FC)
logic
ut
in
Lo
DutyCycle
DC-DC Converter (Fuel Cell)
Ready
151%
auto(daessc)

Fuel Cell Model

Composite component with Simscape Language

Block Parameters: Fuel Cell Unit

Fuel Cell Rack

This component represents a fuel cell rack with a given number of parallel modules. It uses the Fuel Cell block (Simscape Electrical) and scalable build (Simscape language).

Source code | Choose source

Settings

Parameters

Open-circuit voltage:	Fuel_Cell.OC_Voltage	V	Compile-time
Tafel slope:	Fuel_Cell.Tafel_Slope	V	Compile-time
Internal resistance:	Fuel_Cell.Resistance	Ohm	Compile-time
Nominal exchange current:	Fuel_Cell.Exchange_Current	A	Compile-time
collapse current:	Fuel_Cell.Collapse_Current	A	Compile-time
Number of cells:	65		Compile-time
Number of units:	10		Compile-time
Nominal pressure (H2):	Fuel_Cell.pH2_nom	Pa	Compile-time
Nominal pressure (O2):	Fuel_Cell.pAir_nom	Pa	Compile-time
Fuel concentration:	99		Compile-time
Oxygen concentration:	21		Compile-time
Humidity:	1		Compile-time
dyn:	int32(0)		Compile-time
Tnom:	293.15	K	Compile-time
Thermal Mass:	Fuel_Cell.ThermalMass	J/K	Compile-time
Initial Temperature:	Fuel_Cell.Theta_Init	K	Compile-time
Number of Parallel Modules:	Nconfig(2)		Compile-time

OK | Cancel | Help | Apply

Simscape Language

C:_MATLAB\09_Webinar\05 - Electrified Propulsion - Fuel Cell\Library\Fuel_Cell_Rack.ssc

```
42    for i=1:Nr
43        components(ExternalAccess = observe)
44          Module(i) = ee.sources.fuel_cell(model_fidelity = i
45        end
46    end
47
48    components(ExternalAccess = observe)
49       PS_Gain = foundation.signal.functions.gain(gain = 3);
50    end
51
52    connections
53       connect(qH2,PS_Gain.I);
54    end
55
56    for i=1:Nr
57       connections
58           connect(dc_p,Module(i).p);
59           connect(dc_m,Module(i).n);
60           connect(Th,Module(i).H);
61           connect(PS_Gain.O,Module(i).qair)
62           connect(qH2,Module(i).qfuel);
63           connect(pH2,Module(i).pfuel);
64           connect(pAir,Module(i).pair);
65       end
66    end
67
68  end
69
```

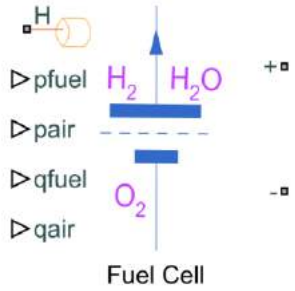
Zoom: 150% | UTF-8 | LF | Simscape model file | Ln 1 | Col 1

Fuel Cell Model

Composite component with Simscape Language


```
for i=1:Nr
     components(ExternalAccess = observe)
       Module(i) = ee.sources.fuel_cell(model_fidelity = i
     end
end

components(ExternalAccess = observe)
   PS_Gain = foundation.signal.functions.gain(gain = 3);
end

connections
   connect(qH2,PS_Gain.I);
end

for i=1:Nr
   connections
       connect(dc_p,Module(i).p);
       connect(dc_m,Module(i).n);
       connect(Th,Module(i).H);
       connect(PS_Gain.O,Module(i).qair)
       connect(qH2,Module(i).qfuel);
       connect(pH2,Module(i).pfuel);
       connect(pAir,Module(i).pair);
   end
end

end
```

- Based on Fuel Cell Block of Simscape Electrical



This chemical reaction defines the electrical conversion:

$$H_2 + \frac{1}{2}O_2 \rightarrow H_2O + heat$$

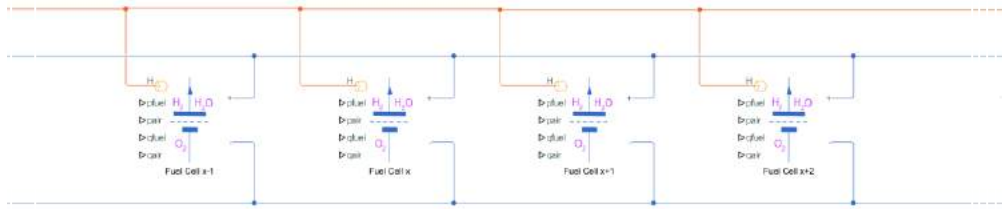
with these catalytic sub-reactions:

$$H_2 \rightarrow 2H^+ + 2e^-$$

$$\frac{1}{2}O_2 + 2e^- \rightarrow O^{2-}$$

Documentation

- Use « components » and « connections » syntax to connect multiple (Nr) Fuel Cell blocks in parallel



Modeling Fuel Cell System

- Thermodynamics Oriented Methodology
  - Electrochemistry & balance of plant
  - Design and optimize FC system
  - Higher fidelity, more details
  - Physical modeling tools

- System-Integration Oriented Methodology
  - Input-output / lookup table / statistics based
  - Integration and supervisory control
  - Fast running, less details
  - Statistical modeling tools

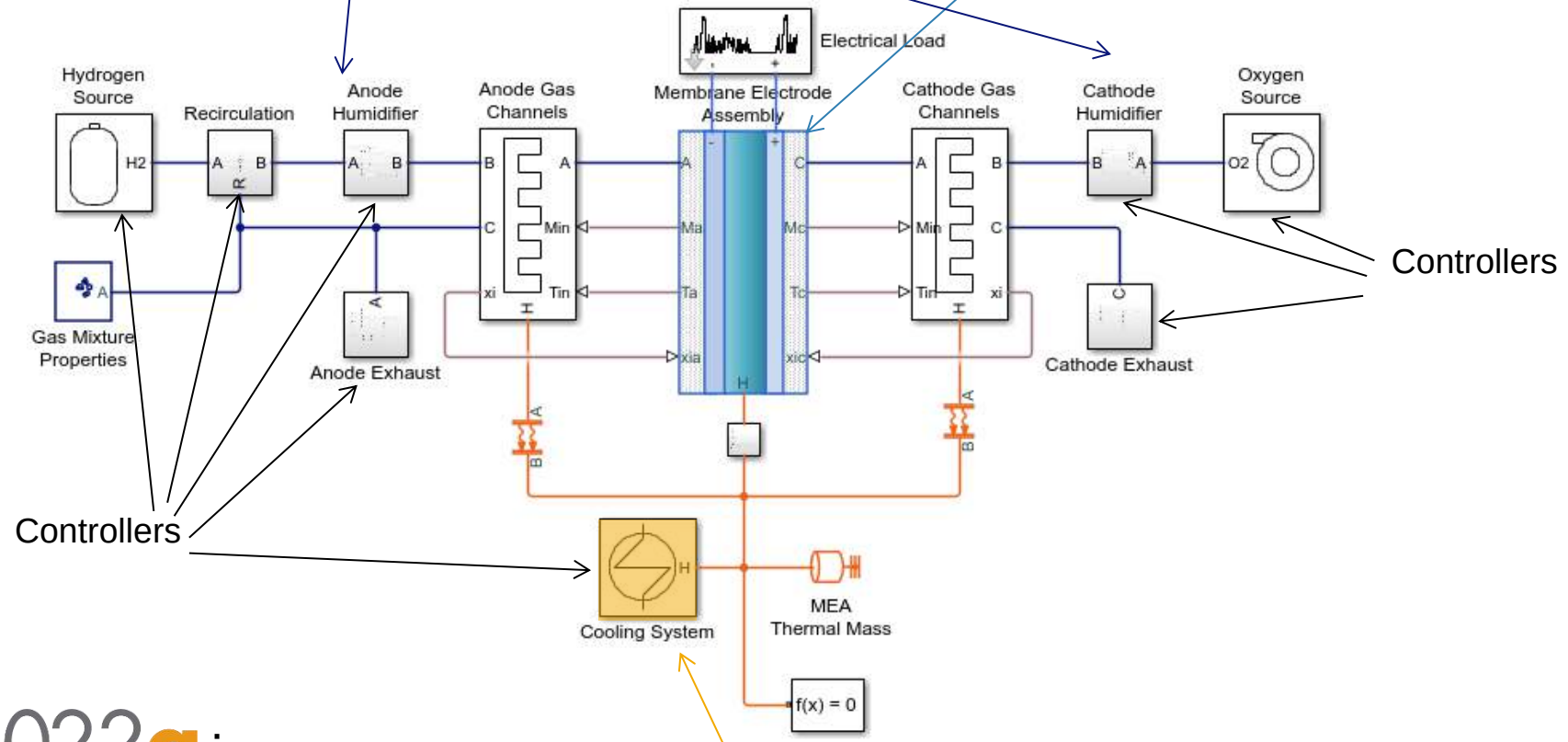
Fuel Cell System Model: Thermodynamics Oriented

Custom Fuel Cell Domain

Multispecies gas network
for N_2,O_2,H_2, H_2O

Membrane

Exchange species and generate elec. power & heat, N_2 diffusion



Liquid cooling system

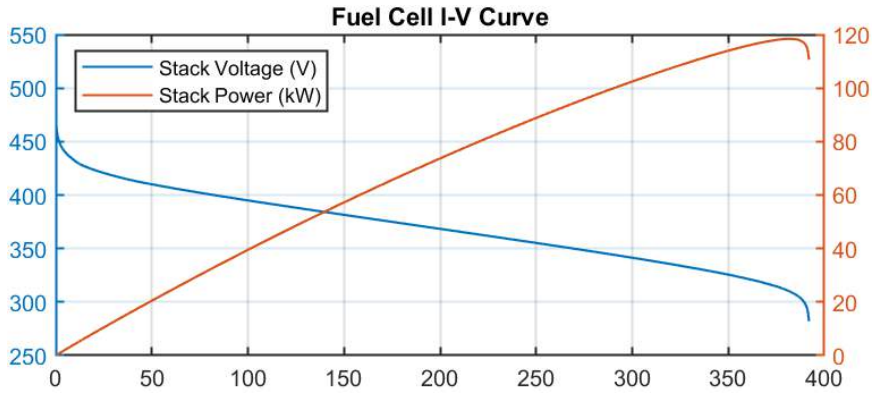
Try it out in R2022a:

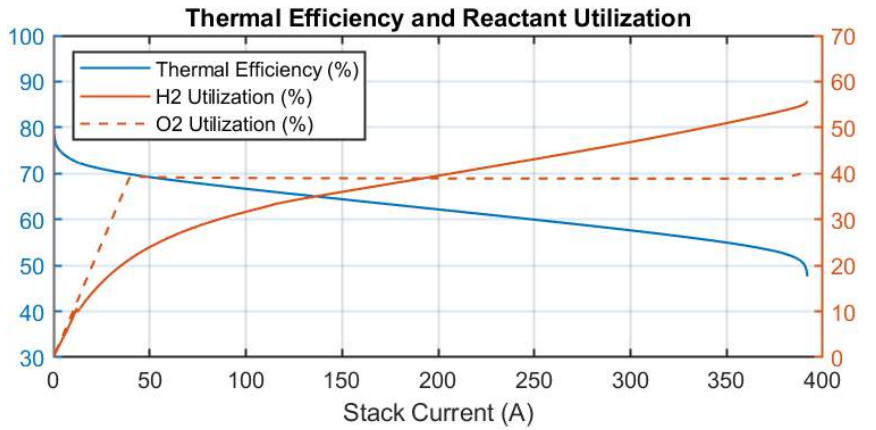
```
>> sscfluids_fuel_cell
```

Example: Characteristic Curves and Drive-Cycle Study

Stack Characteristic Curves

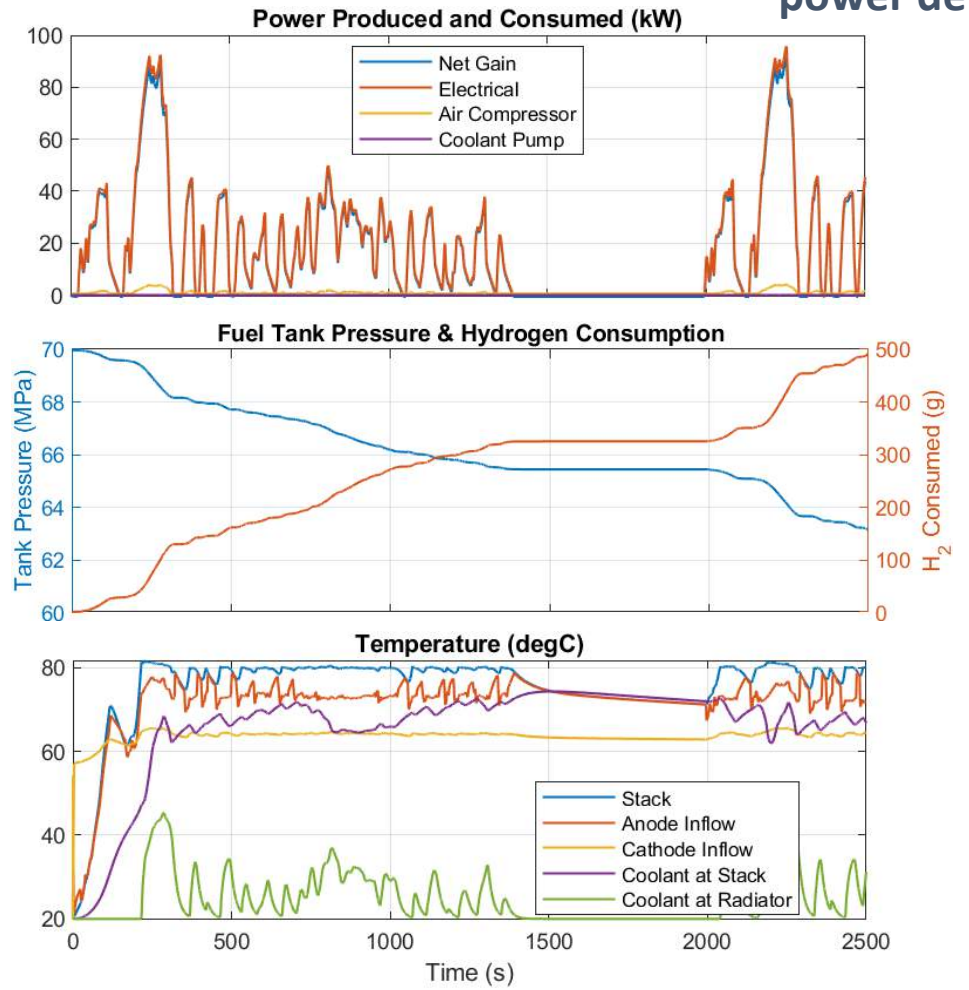
(Current sweep)





Drive-Cycle Response

Based on FTP75 power demand



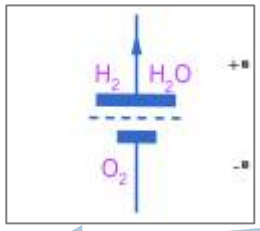
Choose the Appropriate Fidelity Level for Fuel Cell System Modeling

Model what you need …
when you need it

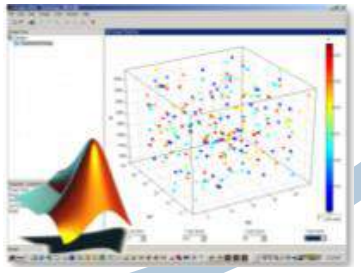
Computational Time

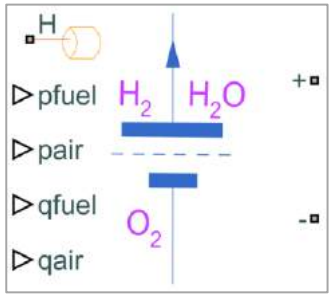
Model Complexity and Details

Voltage vs.
current curve

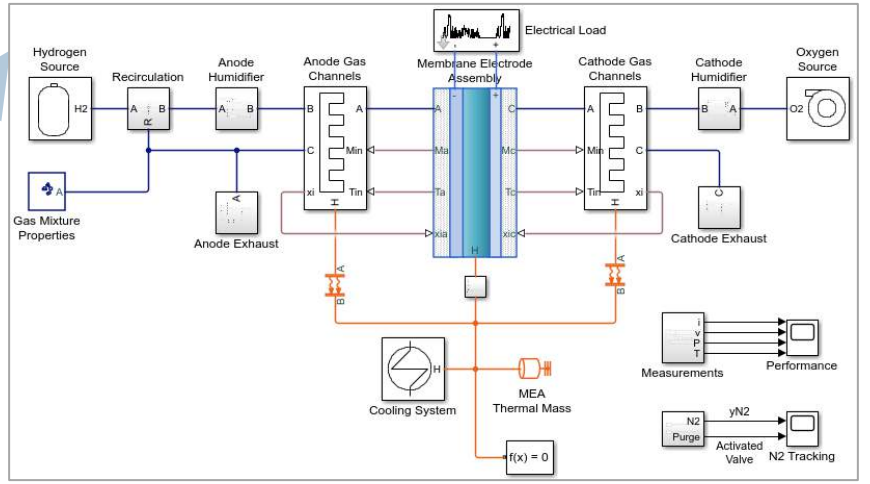


Lookup-table
statistical model
no dynamics





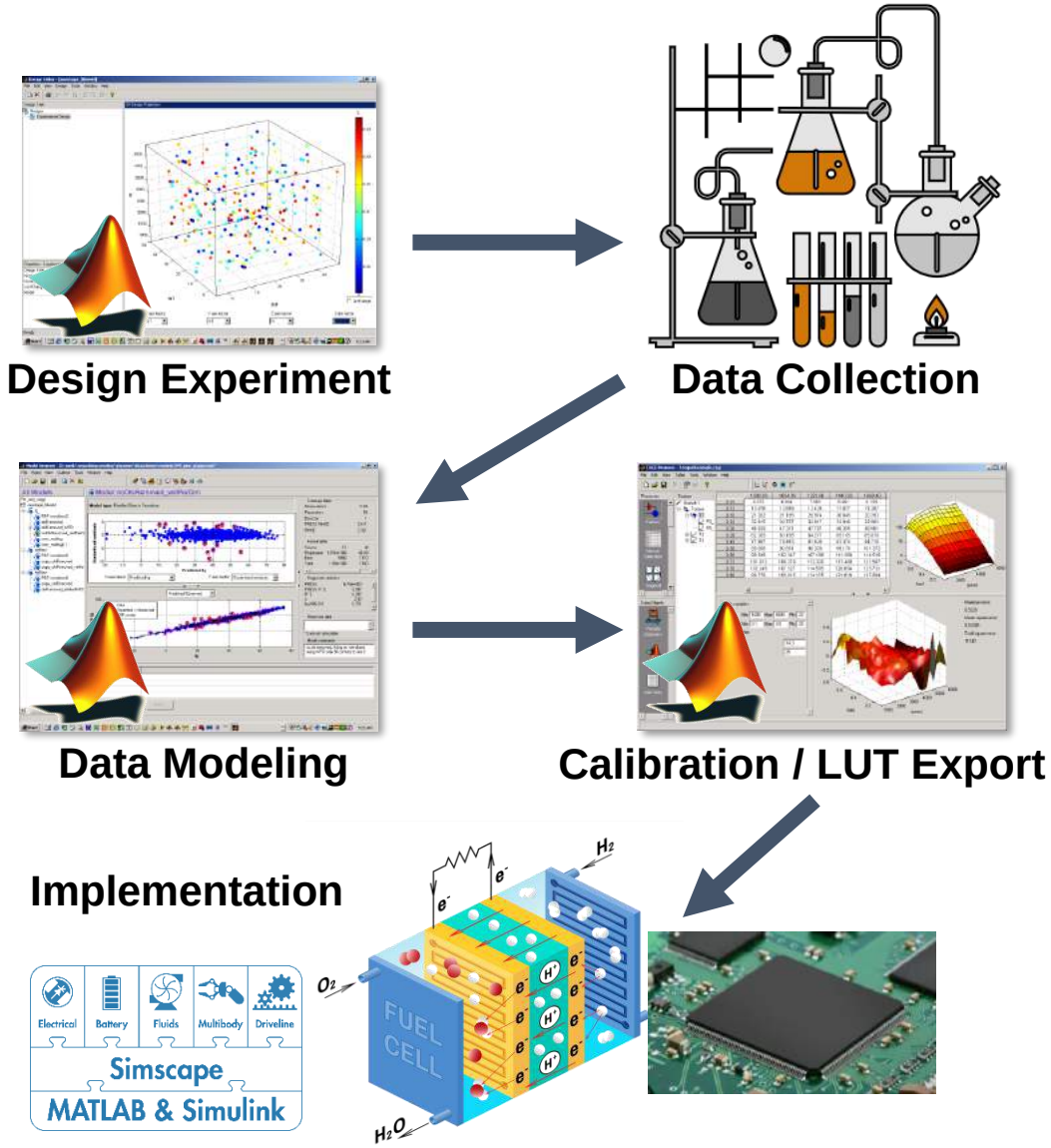
Detailed first principles
model without
gas dynamics



Detailed first principles
model with gas dynamics

Fuel Cell System Model: System-Integration Oriented

- Workflow to build lookup table (LUT), statistics-based model
  - From lab/experimental datasets
  - Or from high-fidelity simulations
  - Fast-running models suitable for integration study and control development
- Model-Based Calibration Toolbox
  - Apps and design tools for modeling and calibrating complex nonlinear systems



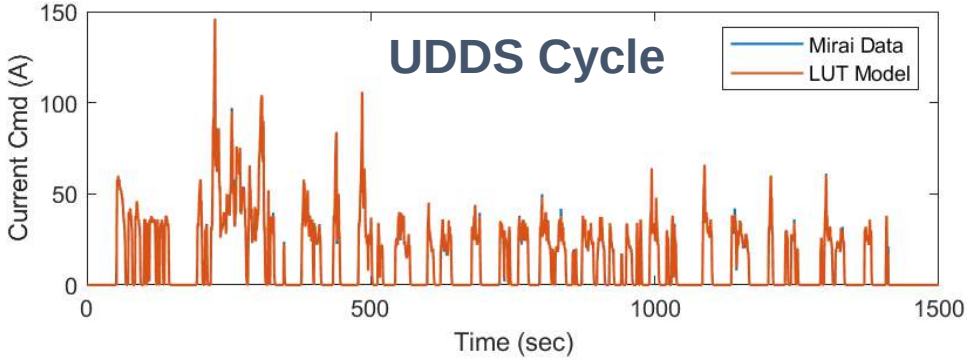
Example: Build Fuel Cell System Model from Lab Data

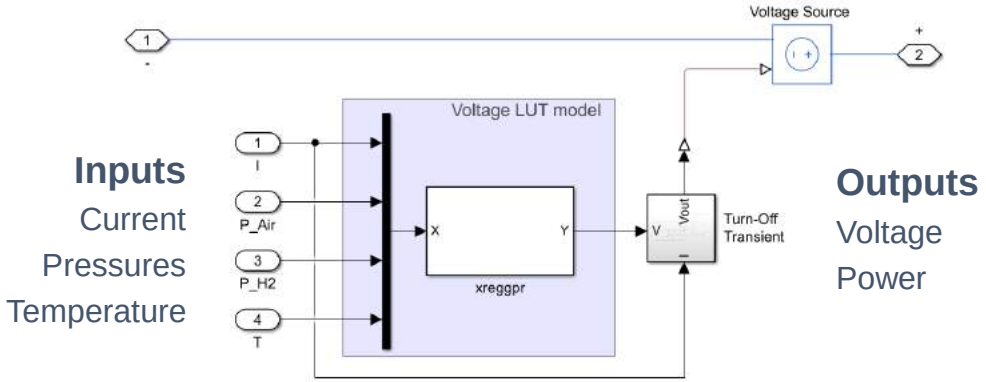
- Lab data collected from an instrumented fuel-cell vehicle

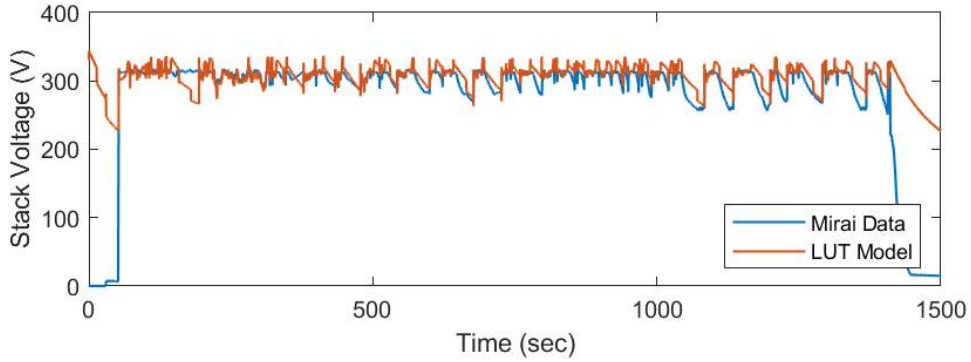
  Argonne National Laboratory, "*Technology Assessment of a Fuel Cell Vehicle: 2017 Toyota Mirai*", ANL/ESD-18/12

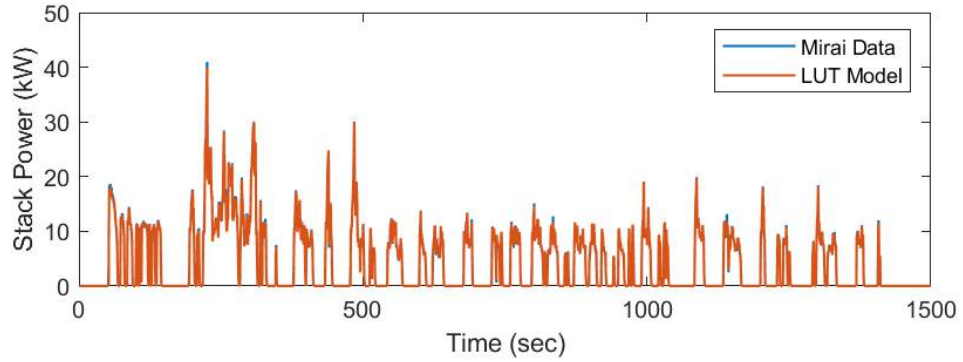
  – MBC model generated and exported for further simulation & validation

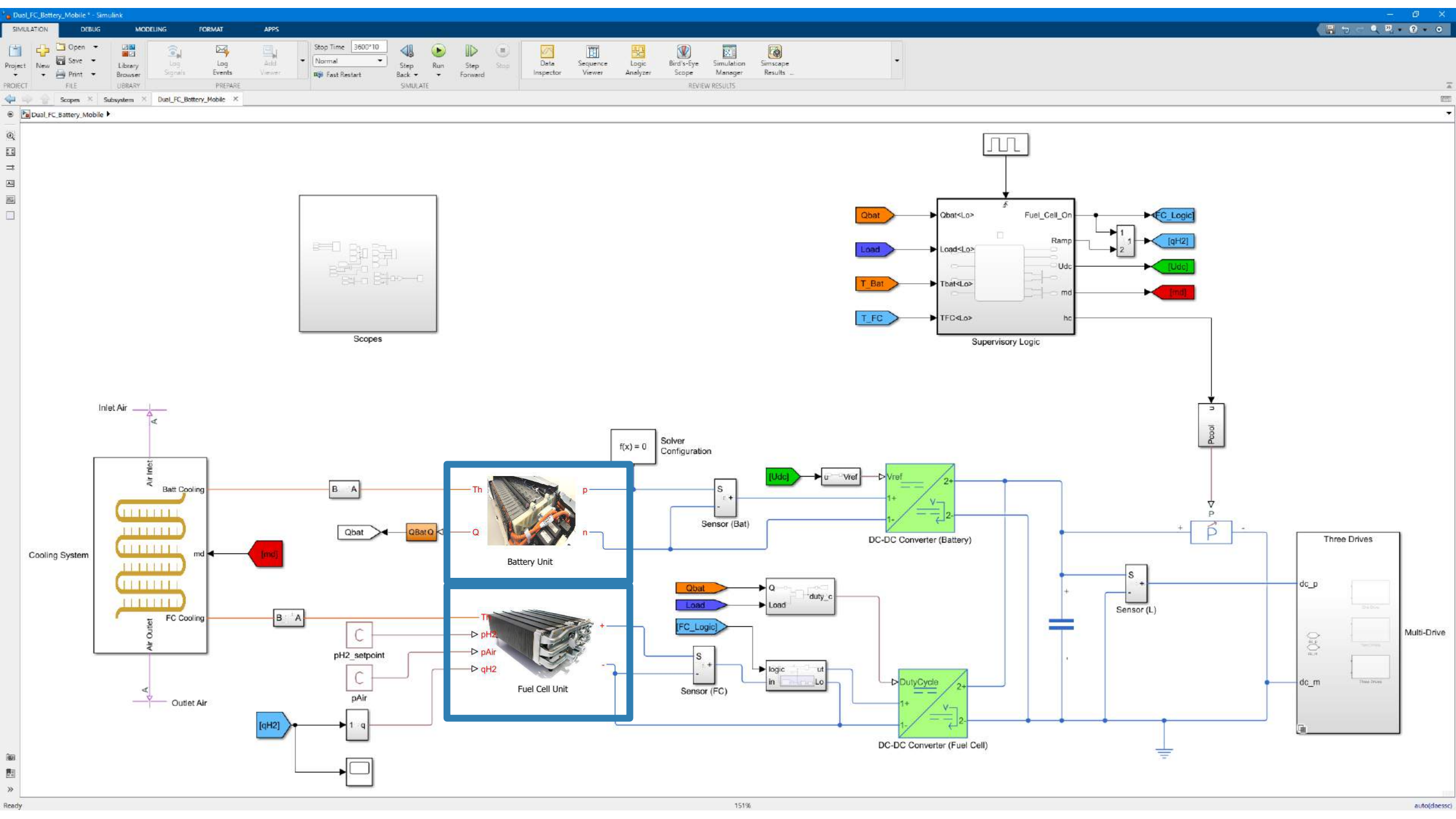
  – Fast-track from lab data to simulation model





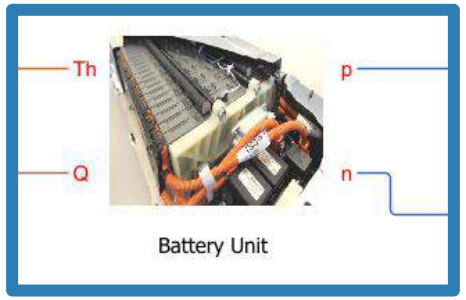





Dual_FC_Battery_Mobile * - Simulink
SIMULATION
DEBUG
MODELING
FORMAT
APPS
Stop Time
3600*10
Normal
Fast Restart
Step Back
Run
Step Forward
Stop
Data Inspector
Sequence Viewer
Logic Analyzer
Bird's-Eye Scope
Simulation Manager
Simscape Results ...
Scopes
Subsystem
Dual_FC_Battery_Mobile
Scopes
Qbat
Load
T_Bat
T_FC
Qbat<Lo>
Load<Lo>
Tbat<Lo>
TFC<Lo>
Fuel_Cell_On
Ramp
Udc
md
hc
FC_Logic
[qH2]
[Udc]
[md]
Supervisory Logic
Inlet Air
Air Inlet
Batt Cooling
Cooling System
md
FC Cooling
Air Outlet
Outlet Air
[md]
B A
Qbat
QBat Q
Th
Q
p
n
Battery Unit
f(x) = 0
Solver Configuration
Sensor (Bat)
[Udc]
u Vref
Vref
DC-DC Converter (Battery)
Pcool
P
Three Drives
dc_p
dc_m
Multi-Drive
Sensor (L)
Qbat
Load
[FC_Logic]
Q
Load
duty_c
Th
pH2
pAir
qH2
Fuel Cell Unit
Sensor (FC)
logic
in
ut
Lo
DutyCycle
DC-DC Converter (Fuel Cell)
pH2_setpoint
pAir
[qH2]
1 q
Ready
151%
auto(daessc)

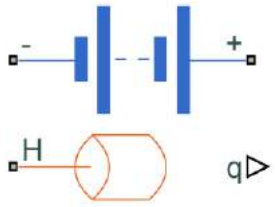
Battery Model

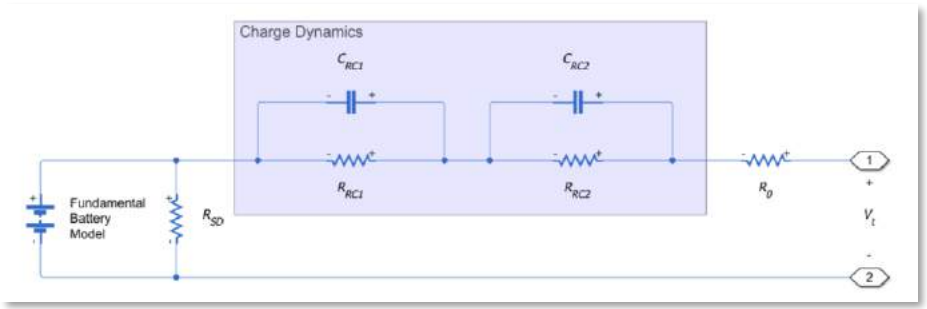
Composite component with Simscape Language




```
C:\_MATLAB\09_Webinar\05 - Electrified Propulsion - Fuel Cell\Library\Battery_Rack.ssc
EDITOR  VIEW
New  Open  Save  Compare  Print  Go To  Find  Bookmark  CODE  ANALYZE  SECTION  Run  Step  Stop
FILE  NAVIGATE  RUN

        v = {250, 'V'}; %Initial Voltage
        charge = {250, 'hr*A'}; %Initial charge
        temperature = {293.15, 'K'}; %Initial temperature
        Nr = {3, '1'}; %Number of Parallel Modules
end

for i=1:Nr
     components(ExternalAccess = observe)
       Module(i) = ee.sources.battery_thermal_instrumented(Vr
     end
end


for i=1:Nr
   connections
       connect(p,Module(i).p);
       connect(n,Module(i).n);
       connect(Th,Module(i).H);
   end
end

connections
    connect(Module(1).q, Q);
end

end

Zoom: 150%  UTF-8  LF  Simscape model file  Ln 1  Col 1
```

- Based on Battery Block of Simscape Electrical



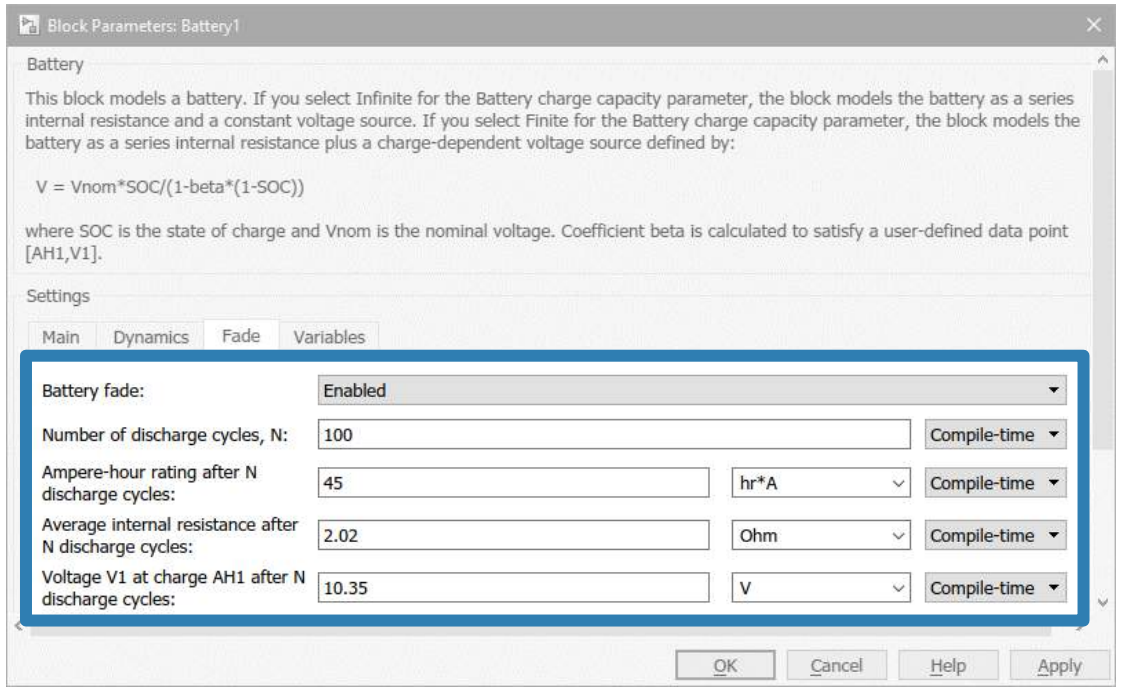


Documentation - Equivalent Model

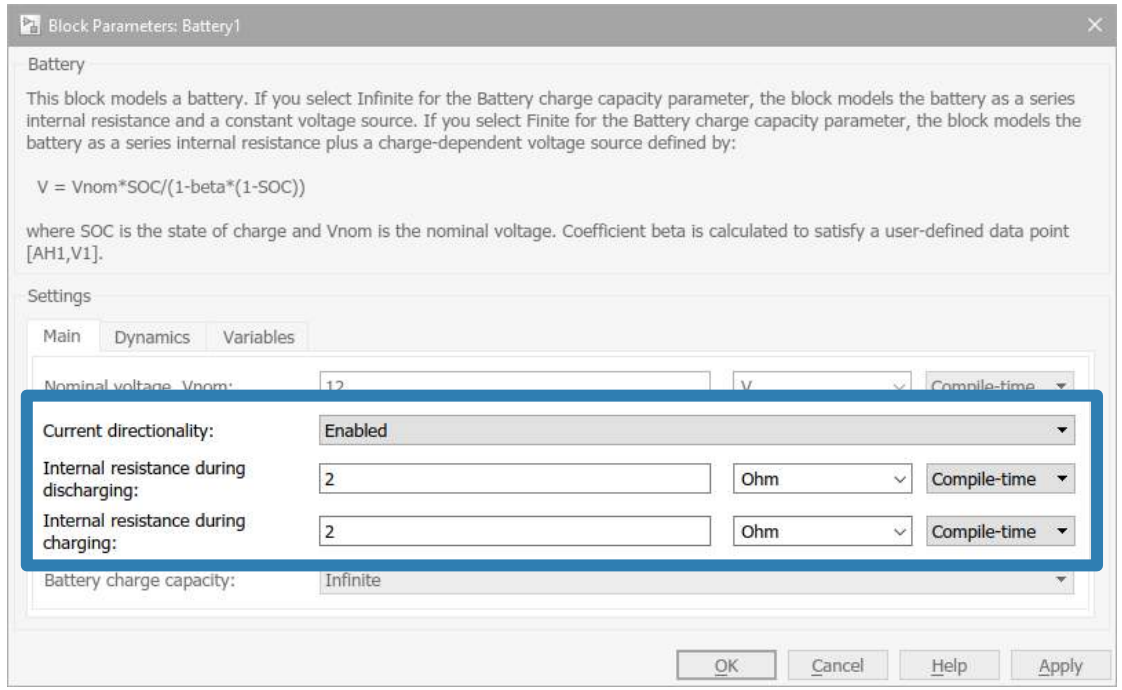
Battery modeling

Directionality and aging

Model advanced **battery fade** through both temperature-independent and temperature-dependent tabulated parameterization



Model the internal **terminal resistance based on the direction** of the current

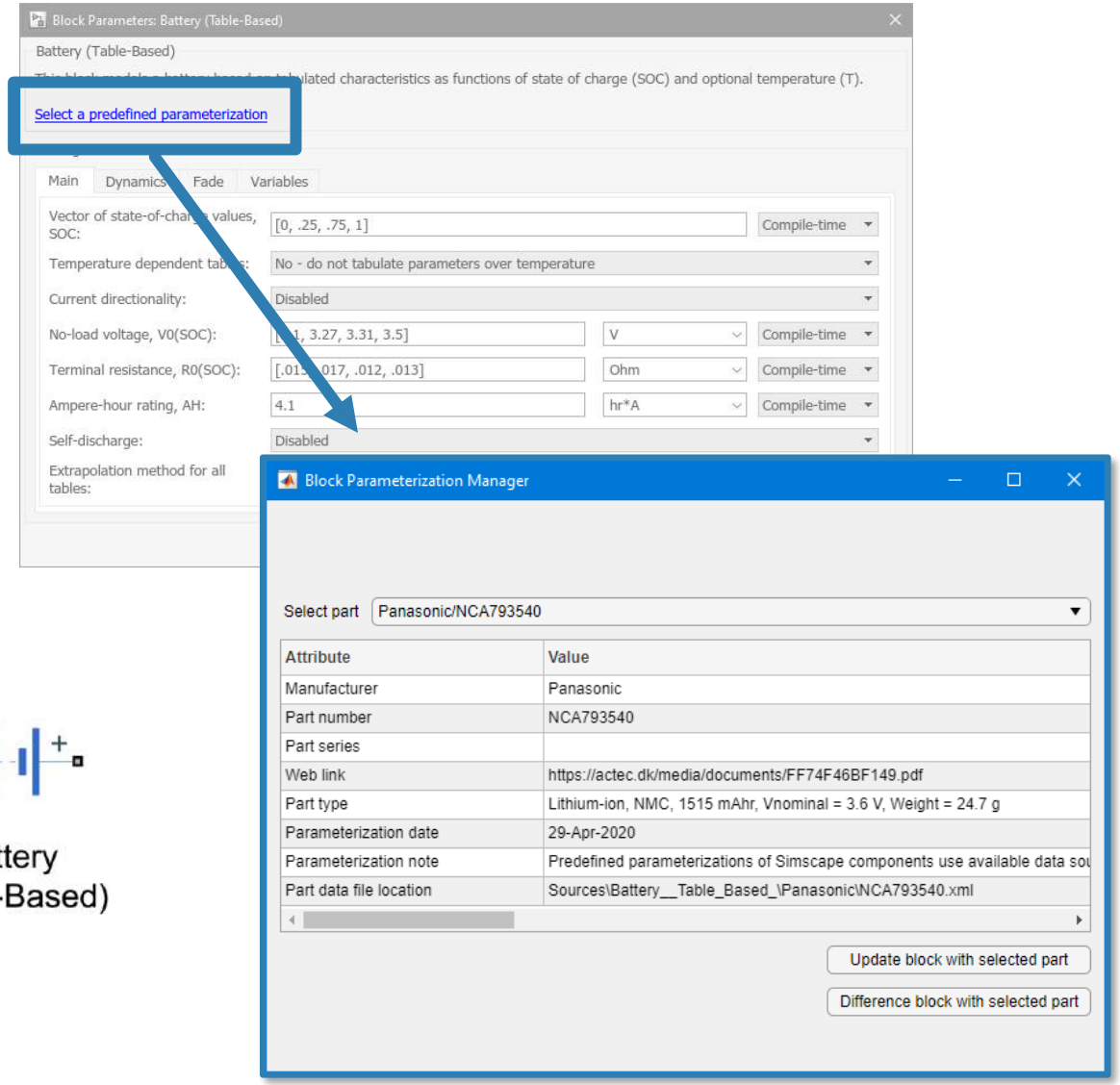


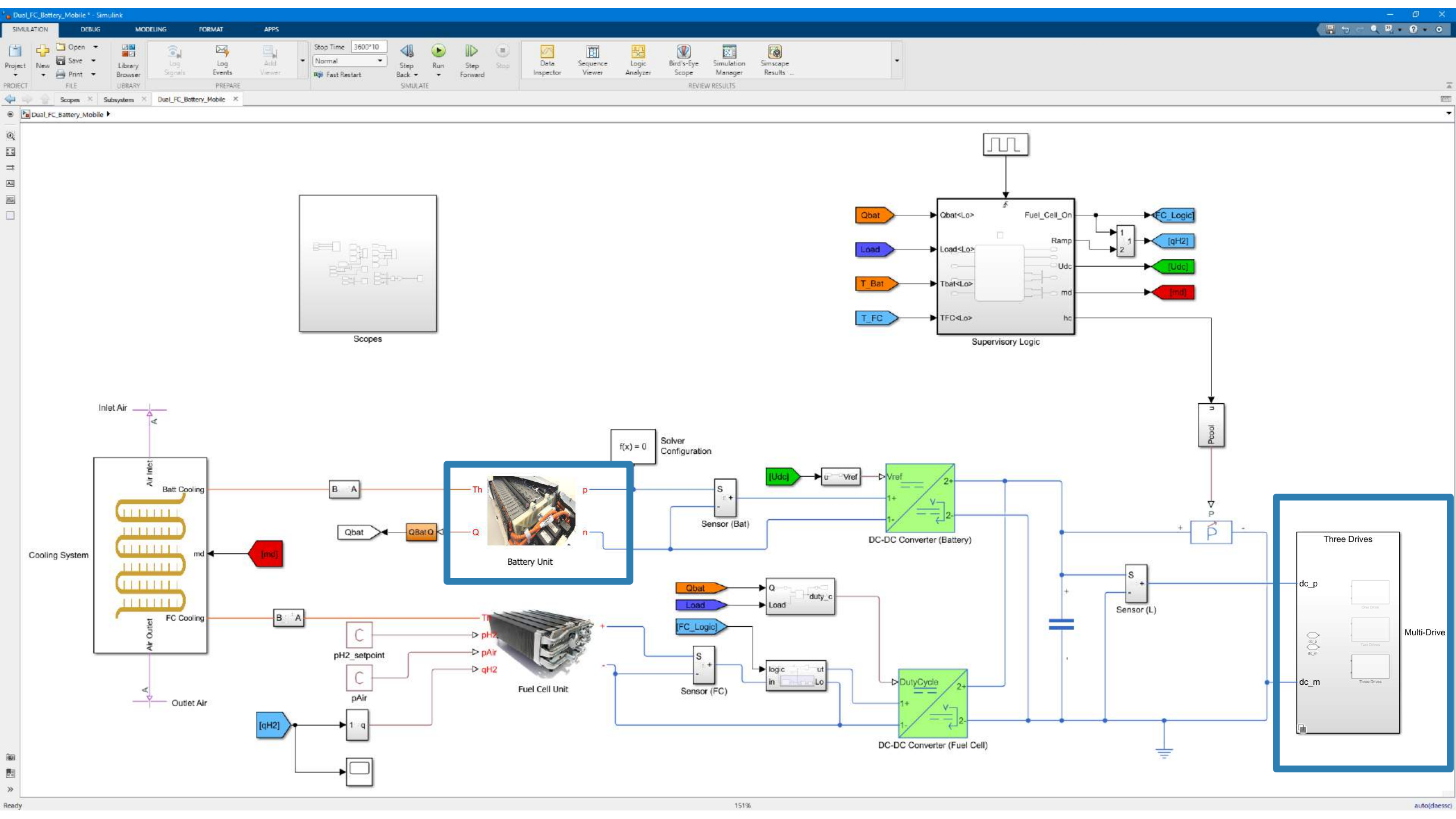
Battery modeling

Predefined Parameterization

- Multiple available built-in parameterizations for the *Battery (Table-Based)* block.
- The parameterizations of these batteries match the manufacturer data sheets.

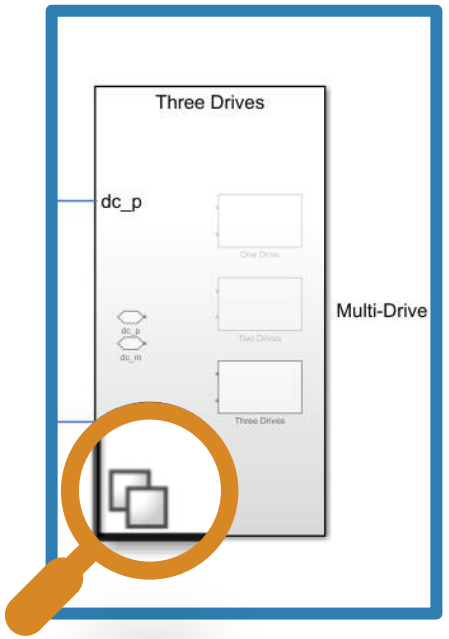
Battery
(Table-Based)

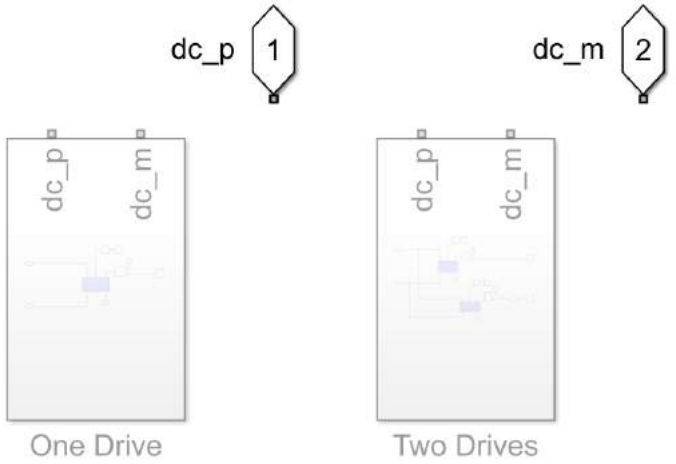


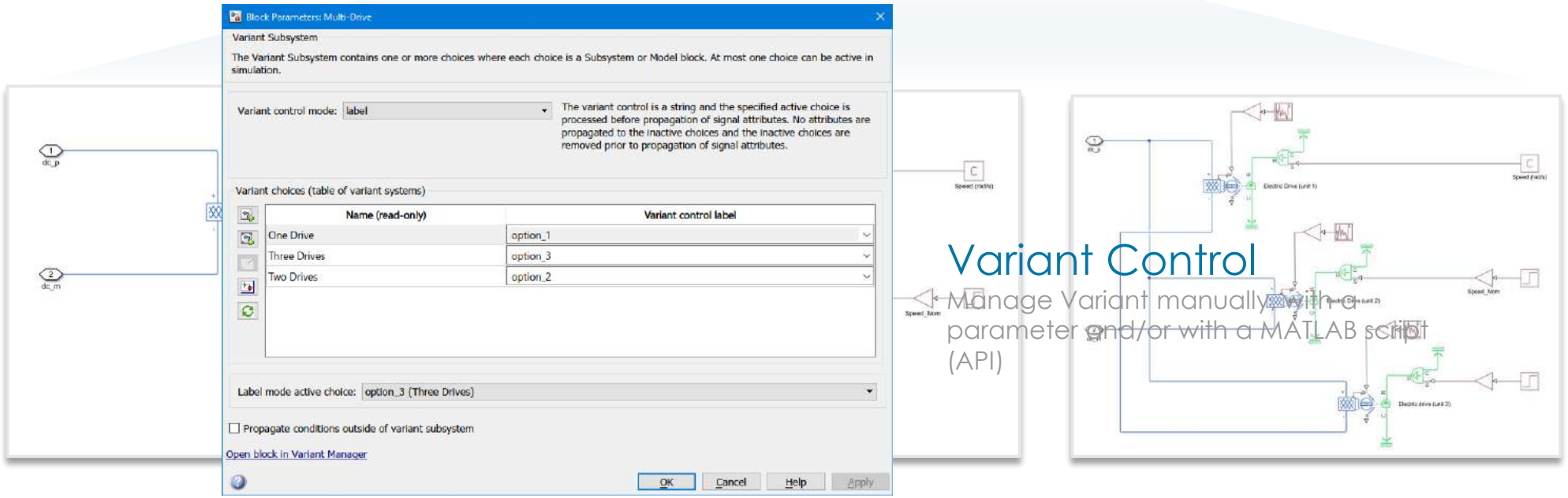

Dual_FC_Battery_Mobile * - Simulink
SIMULATION
DEBUG
MODELING
FORMAT
APPS
Stop Time
3600*10
Normal
Fast Restart
Step Back
Run
Step Forward
Stop
Data Inspector
Sequence Viewer
Logic Analyzer
Bird's-Eye Scope
Simulation Manager
Simscape Results
Scopes
Subsystem
Dual_FC_Battery_Mobile
Scopes
Qbat
Load
T_Bat
T_FC
Qbat<Lo>
Load<Lo>
Tbat<Lo>
TFC<Lo>
Fuel_Cell_On
Ramp
Udc
md
hc
FC_Logic
[qH2]
[Udc]
[md]
Supervisory Logic
Inlet Air
Air Inlet
Batt Cooling
Cooling System
md
FC Cooling
Air Outlet
Outlet Air
[md]
B A
Qbat
QBat Q
Th
Q
p
n
Battery Unit
f(x) = 0
Solver Configuration
Sensor (Bat)
[Udc]
u Vref
Vref
DC-DC Converter (Battery)
Pcool
P
Sensor (L)
Three Drives
dc_p
dc_m
Multi-Drive
pH2_setpoint
pAir
[qH2]
Fuel Cell Unit
Qbat
Load
[FC_Logic]
Sensor (FC)
logic
DutyCycle
DC-DC Converter (Fuel Cell)
Ready
151%
auto(daessc)

Drives

Variant Subsystems





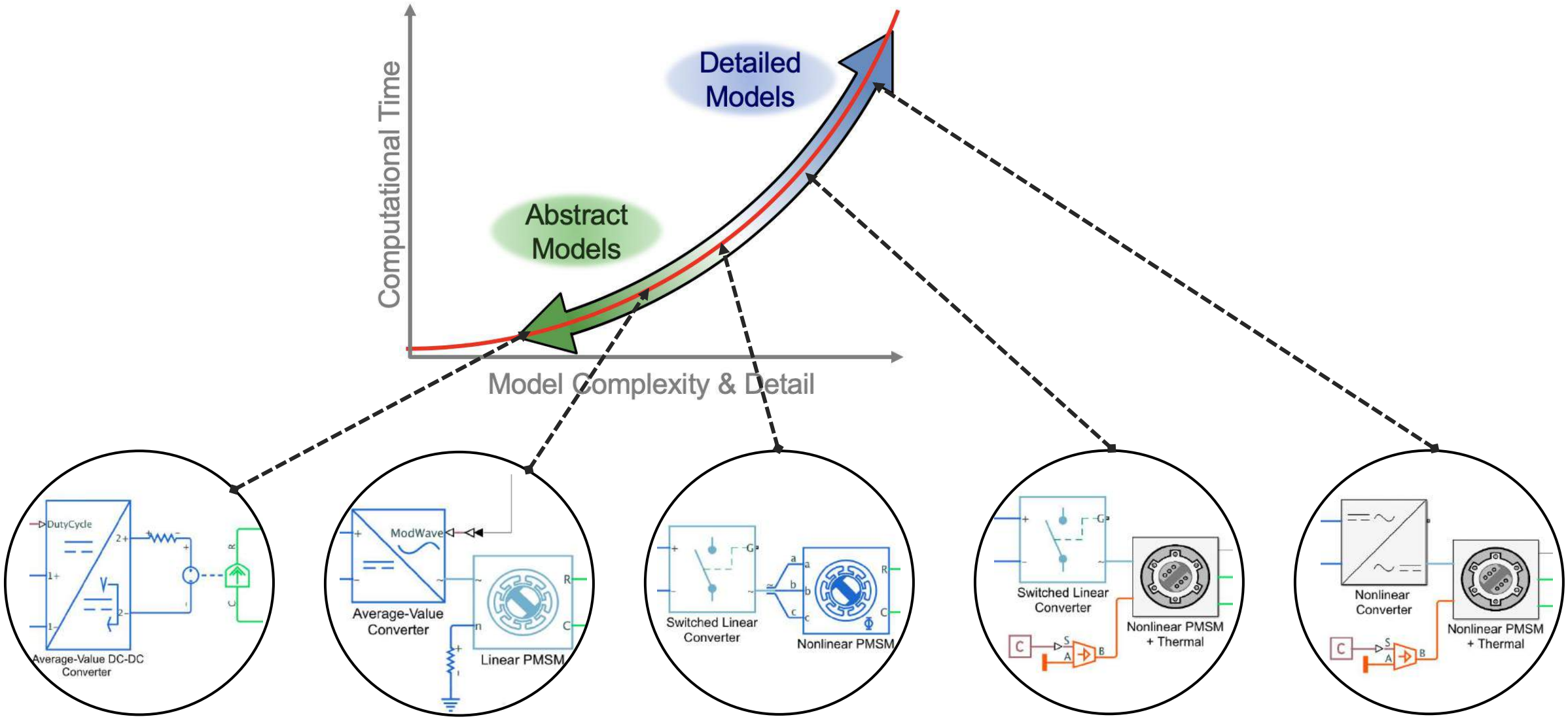


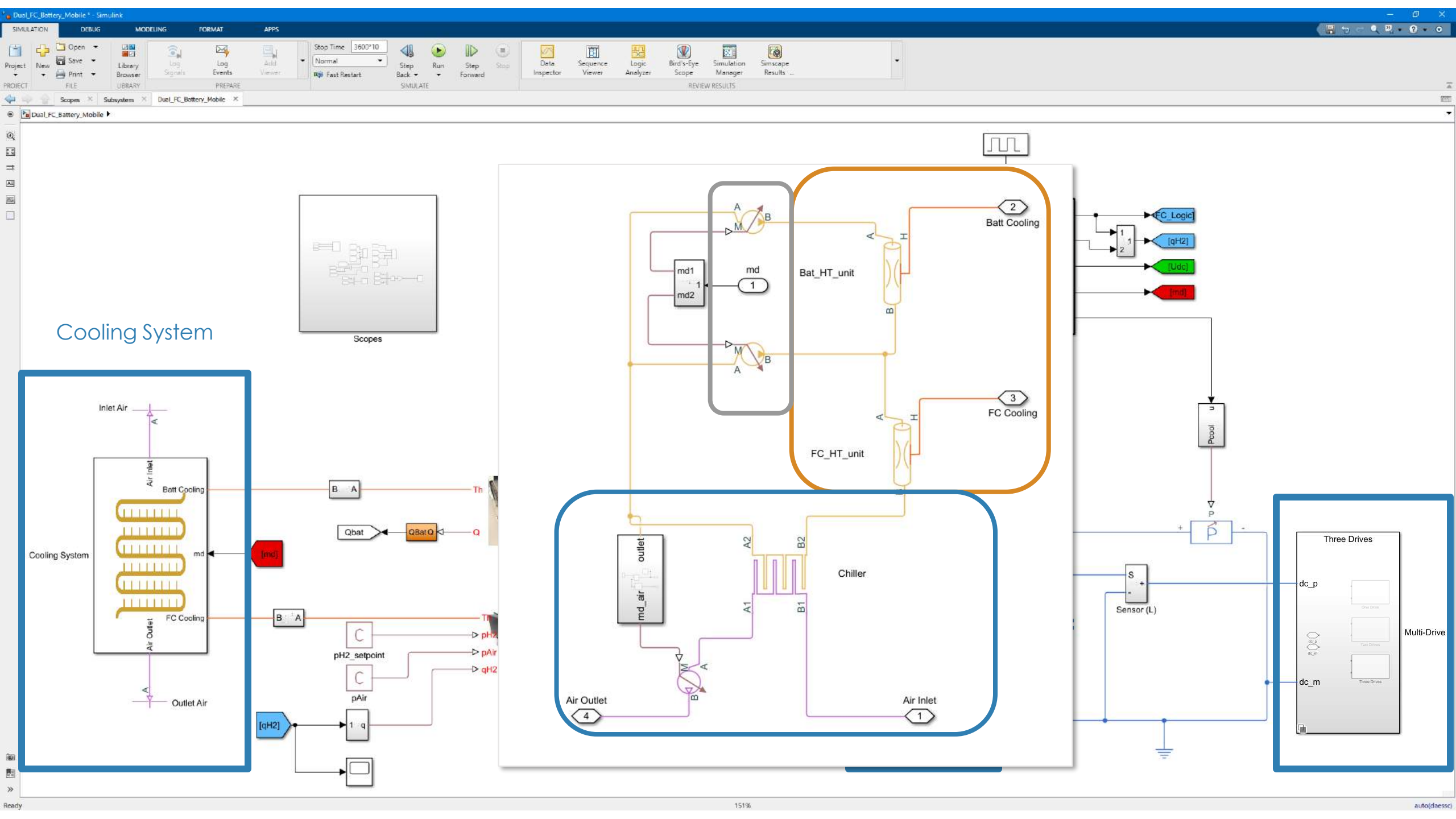
Variant Control

Manage Variant manually with a parameter and/or with a MATLAB script (API)

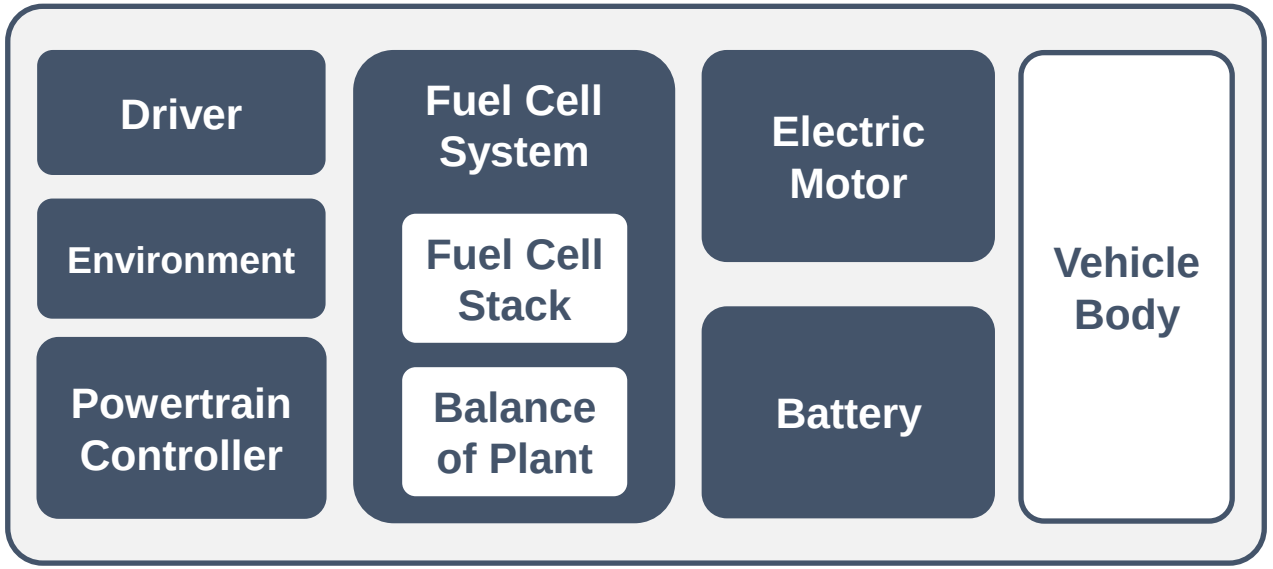
➔ Use variants to switch automatically between different variations of a component

Configure the Model to Balance Model Fidelity and Simulation Speed



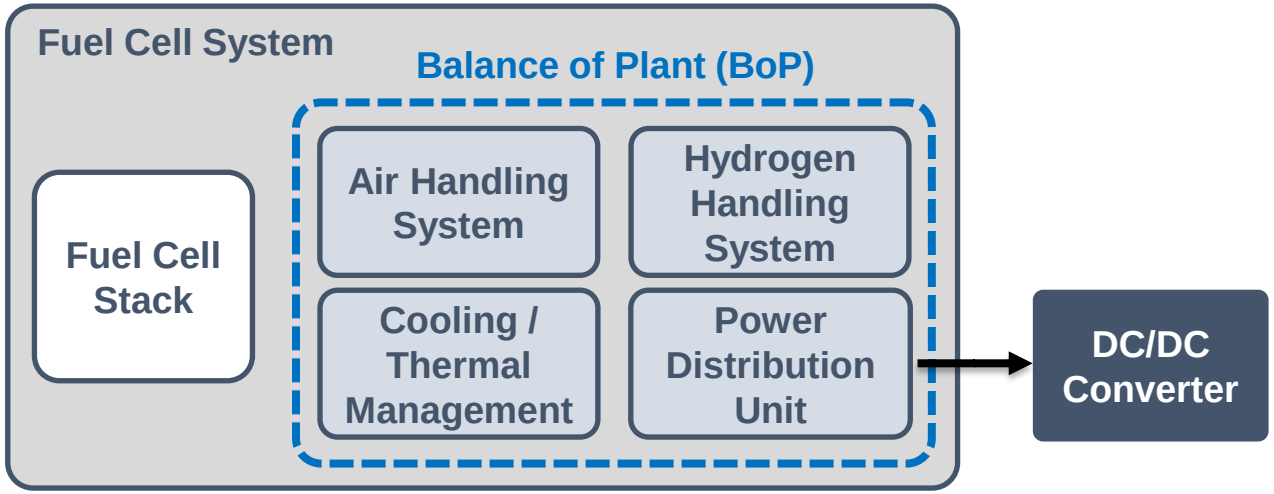

Dual_FC_Battery_Mobile * - Simulink
SIMULATION
DEBUG
MODELING
FORMAT
APPS
Stop Time
3600*10
Normal
Fast Restart
Step Back
Run
Step Forward
Stop
Data Inspector
Sequence Viewer
Logic Analyzer
Bird's-Eye Scope
Simulation Manager
Simscape Results
Scopes
Subsystem
Dual_FC_Battery_Mobile
Cooling System
Scopes
Inlet Air
Air Inlet
Batt Cooling
FC Cooling
Air Outlet
Outlet Air
md
[md]
Qbat
QBat Q
Th
Q
pH2_setpoint
pAir
[qH2]
md1
md2
md
Bat_HT_unit
FC_HT_unit
2
Batt Cooling
3
FC Cooling
outlet
md_air
Chiller
A1
A2
B1
B2
Air Outlet
Air Inlet
4
1
[FC_Logic]
[qH2]
[Udc]
[md]
Pcool
Sensor (L)
Three Drives
dc_p
dc_m
Multi-Drive
Ready
151%
auto(daessc)

Fuel Cell Virtual Vehicle (FCEV) Architecture



Challenge: fuel cell system interacts with the rest of the electrical powertrain

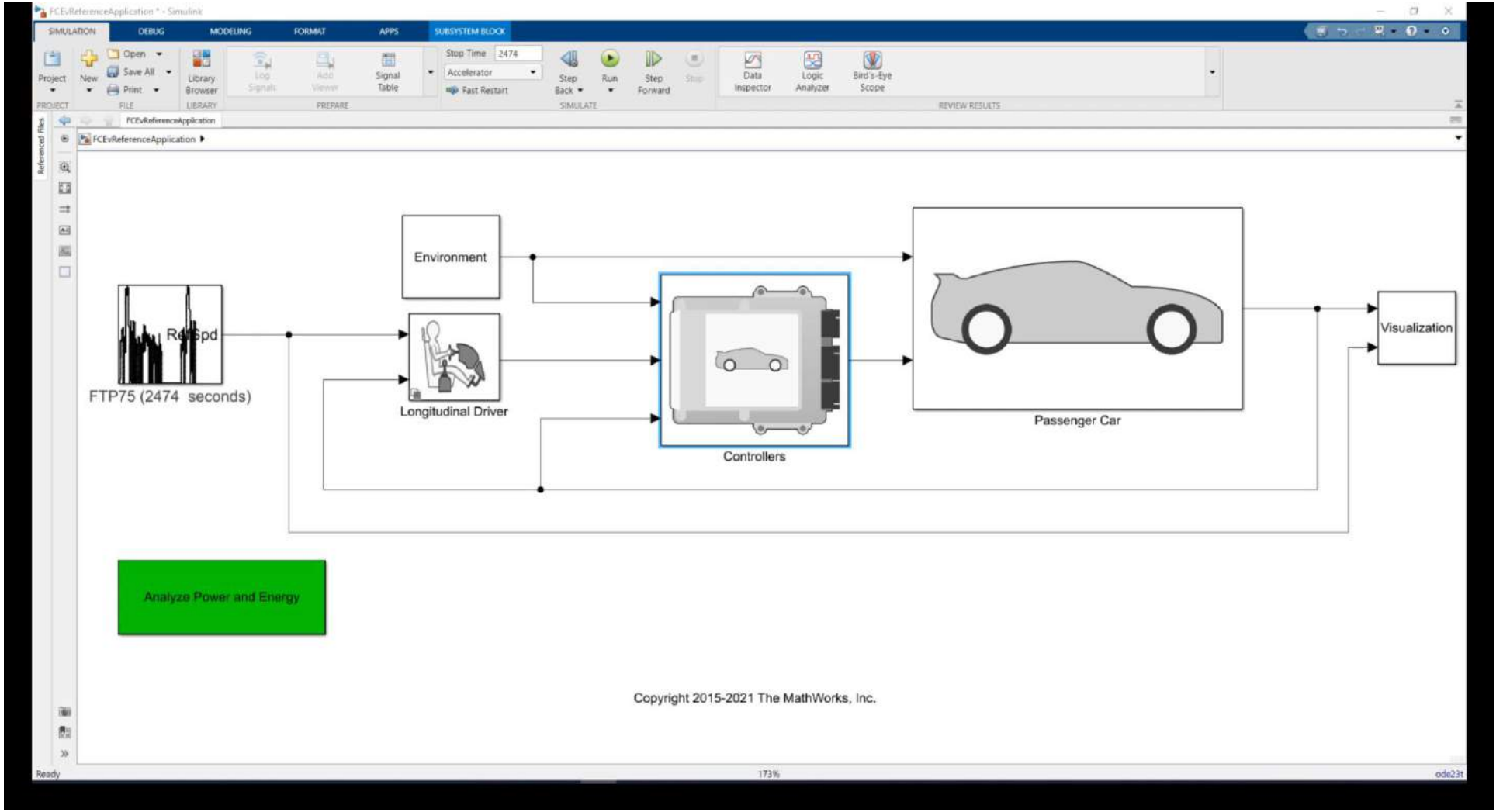
- Drive cycles and operation scenarios
- Motor, battery, DCDC converter, drivetrain
- Supervisory and local control algorithm



Fuel cell system operation in a FCEV

- Determine instantaneous power demand
- Convert power demand to current demand
- Distribute current demand between battery and fuel cell
- Translate current command to H_2 / Air flow commands

Use Case Video: Control Development for Fuel Cell EV



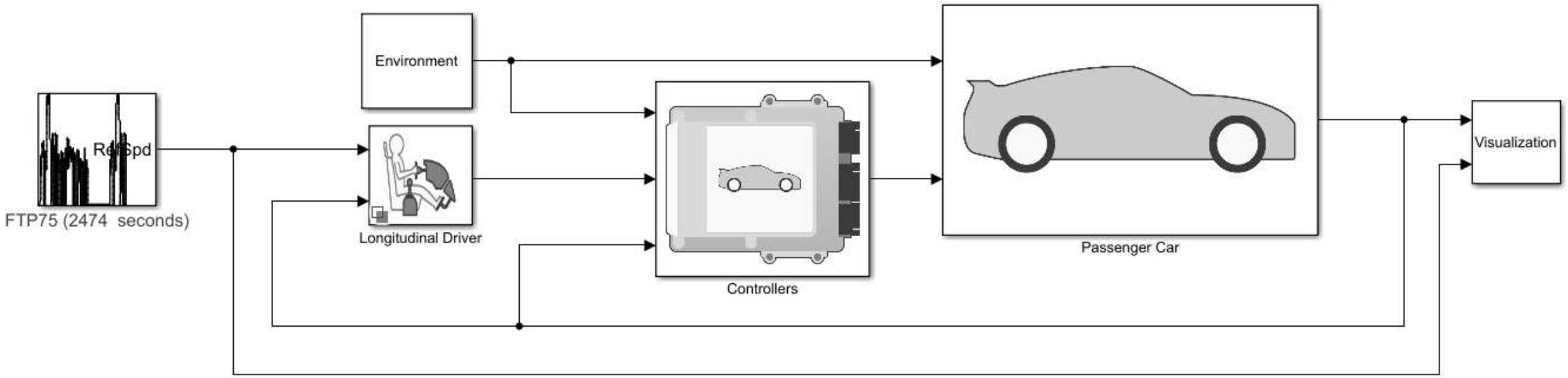
Modeling Fuel Cell Virtual Vehicle

Enable delivery of fuel cell based systems, through vertical integration

model fuel cells, electrified powertrain and virtual vehicle

integrate fuel cell in virtual vehicle models

calibrate and **analyze** fuel cell virtual vehicles



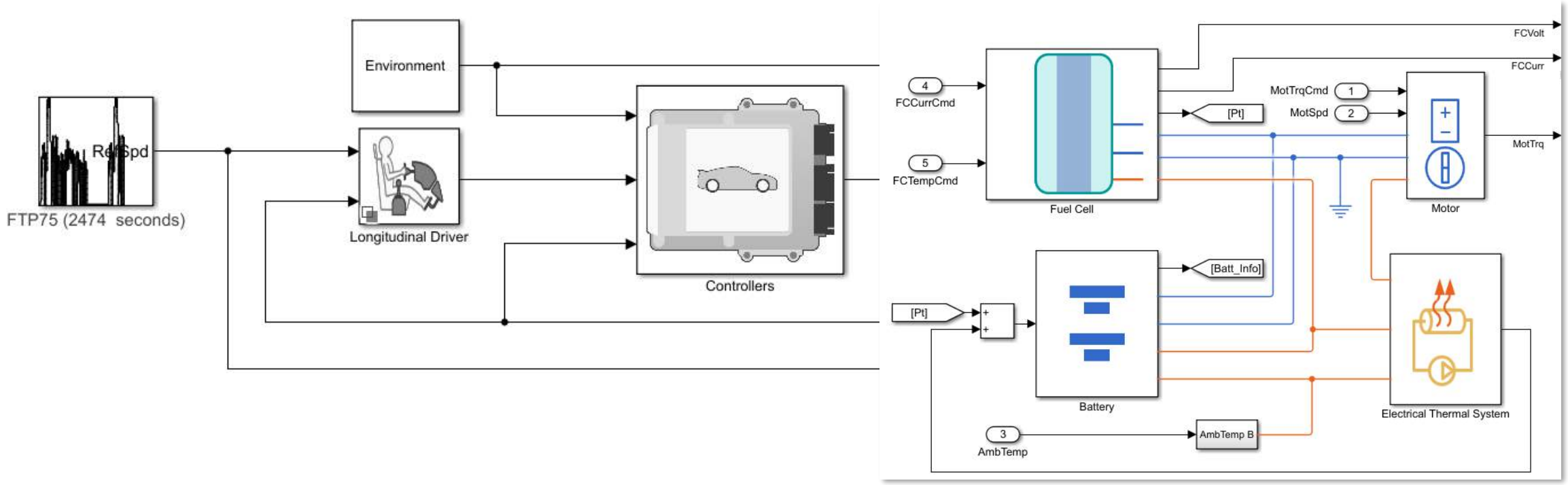
Modeling Fuel Cell Virtual Vehicle

Enable delivery of fuel cell based systems, through vertical integration

model fuel cells, electrified powertrain and virtual vehicle

integrate fuel cell in virtual vehicle models

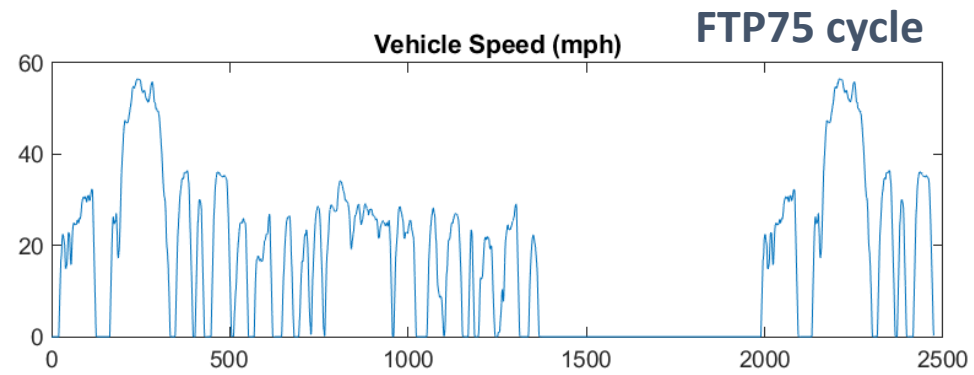
calibrate and **analyze** fuel cell virtual vehicles

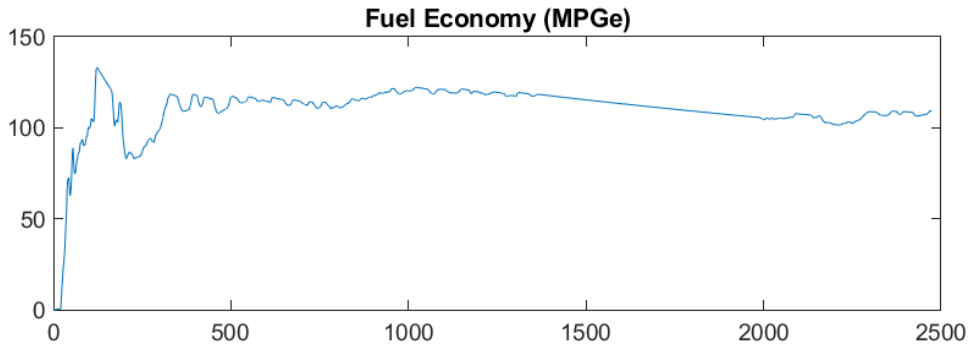


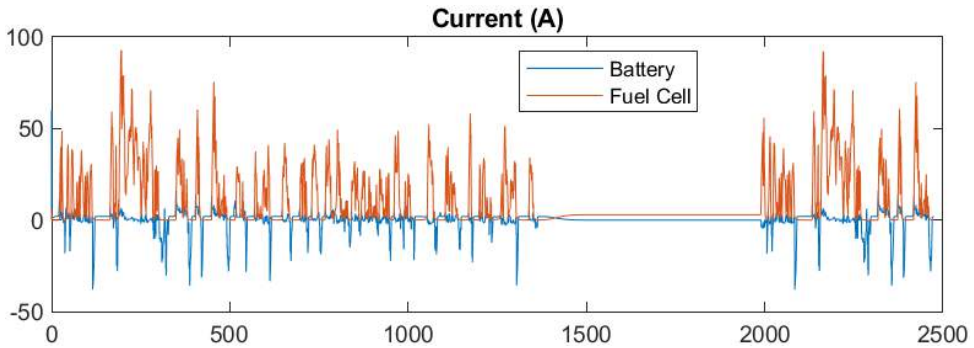
Electric plant model

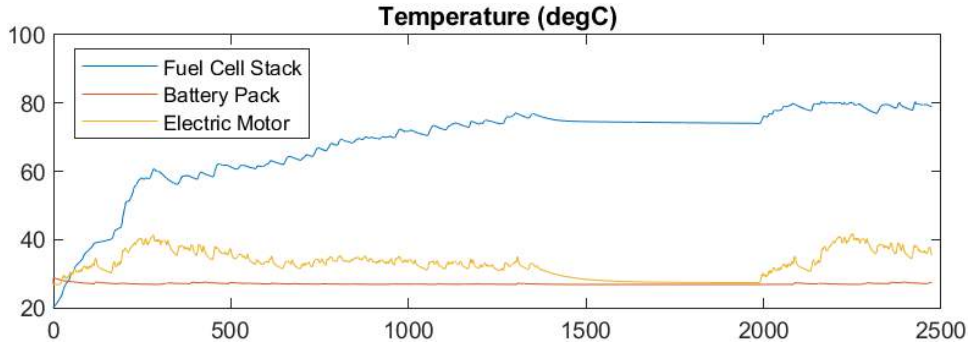
Example: Fuel Cell Virtual Vehicle Simulation

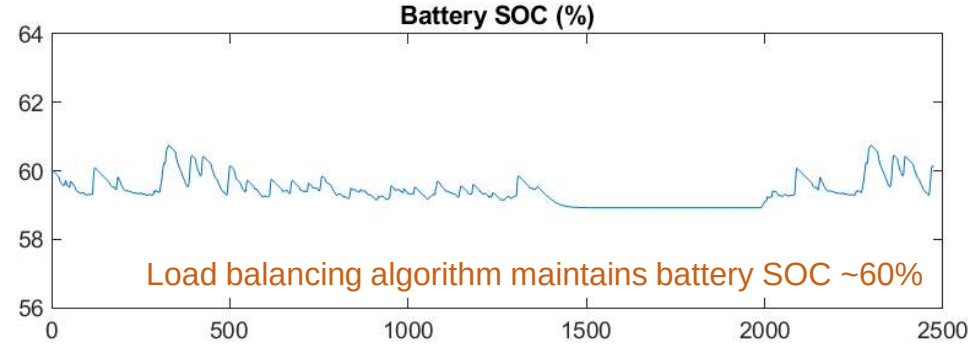
- Full vehicle fuel economy, performance and thermal analysis
- Enables model-based control design





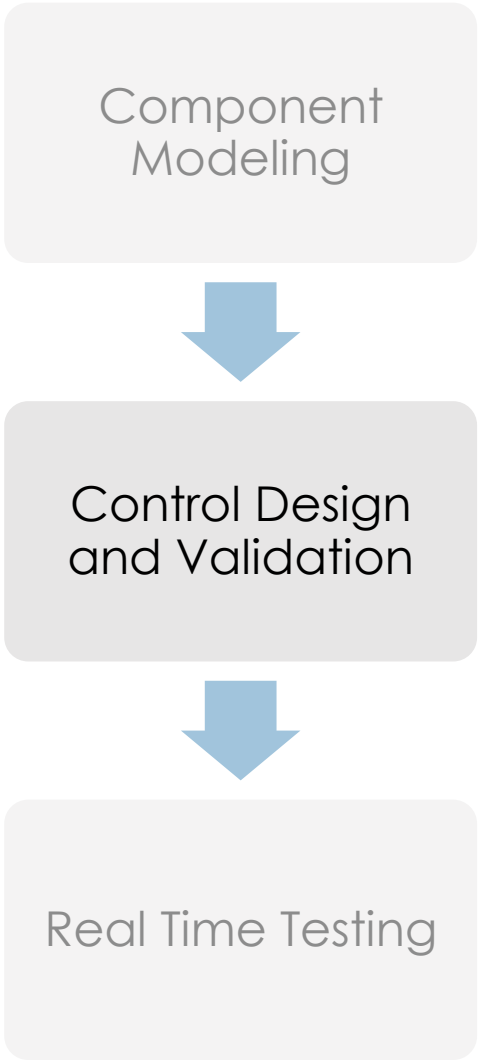




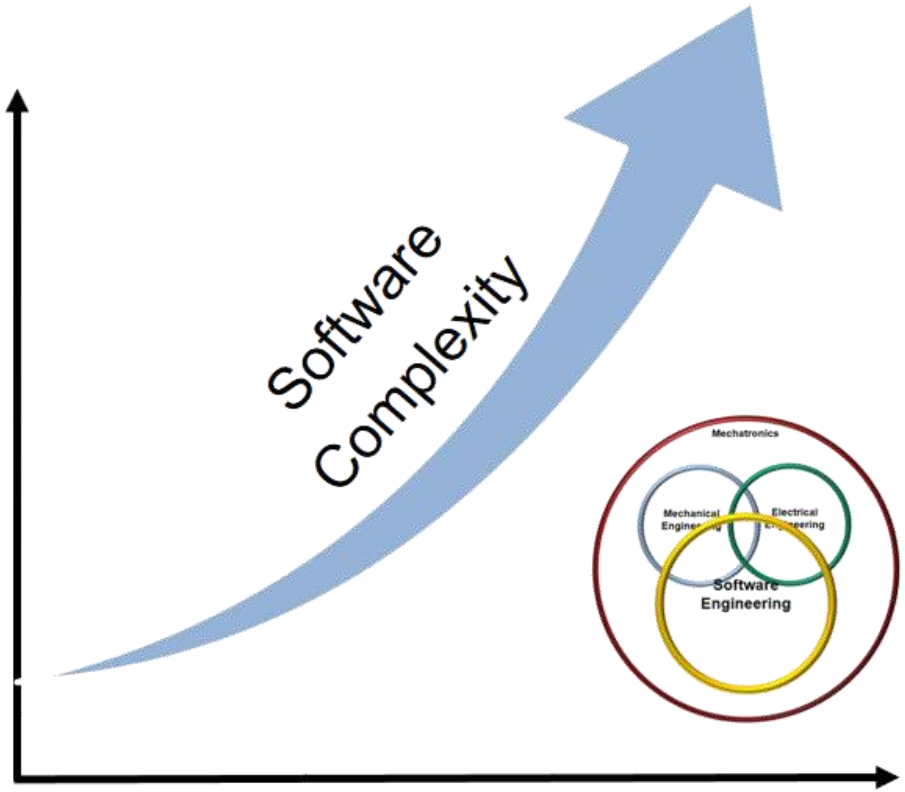


Agenda

- Multi-stack fuel cells and battery propulsion system
- **Supervisory logic development, verification and code generation**
- Real-time testing and prototyping



Growing complexity of **Software**



Growing complexity of fuel cell application systems:

- Modularity and reusability
- More (embedded) software
- Risk of long commissioning times due to insufficient software testing
  - Expensive hydrogen waste

Supervisory Logic Development, Verification and Code Generation

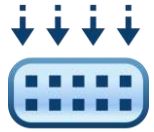
Motivations

Models as Executable Specifications

Simulations to convey and analyze intent, not ambiguous written specification

Measure and Improve Quality

Ensure fully tested models, standards compliance and detect design errors early on



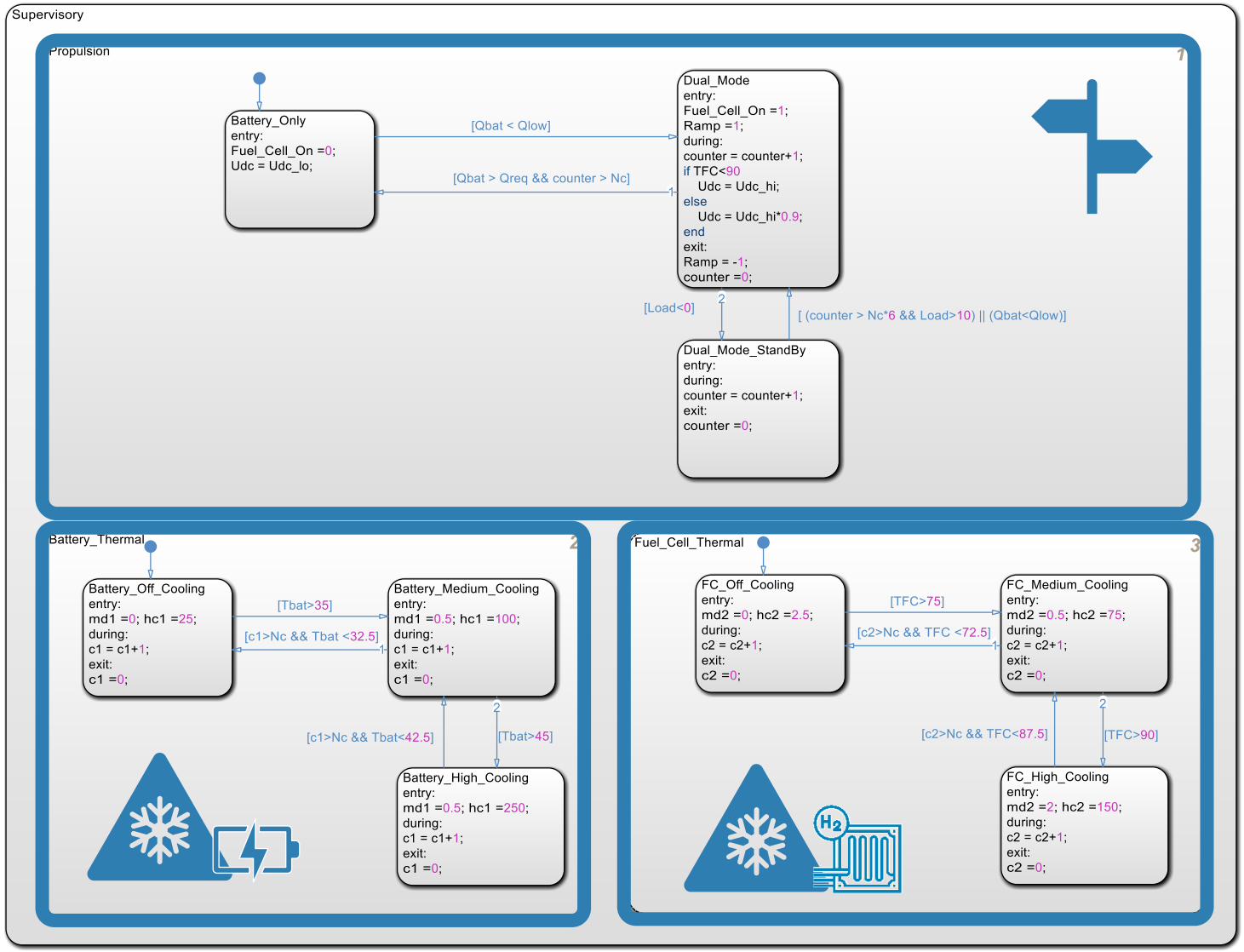
Models to Code

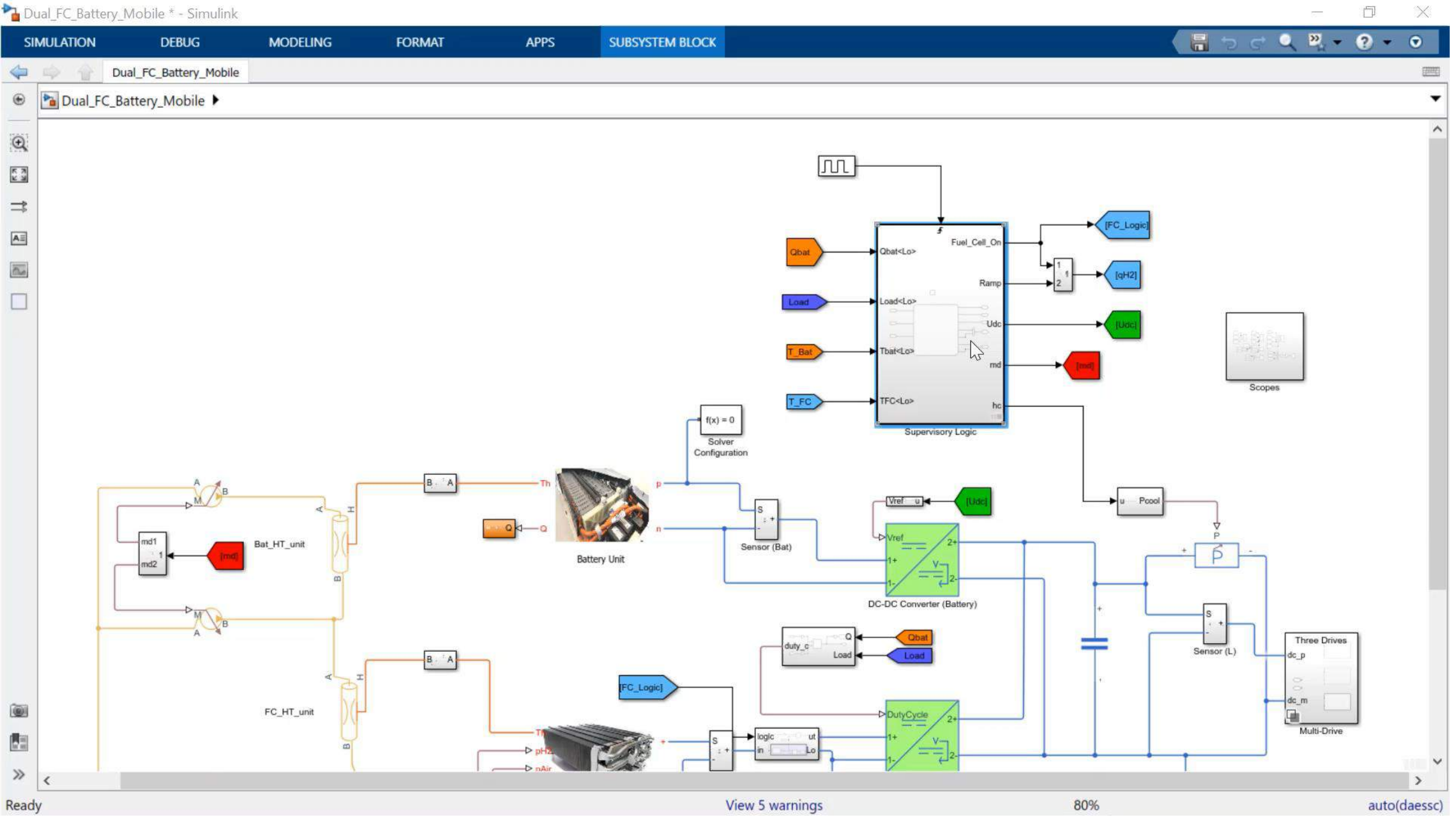
Reuse algorithms through generation of high quality and optimized source code for various systems

Control Algorithms

With Simulink and Stateflow

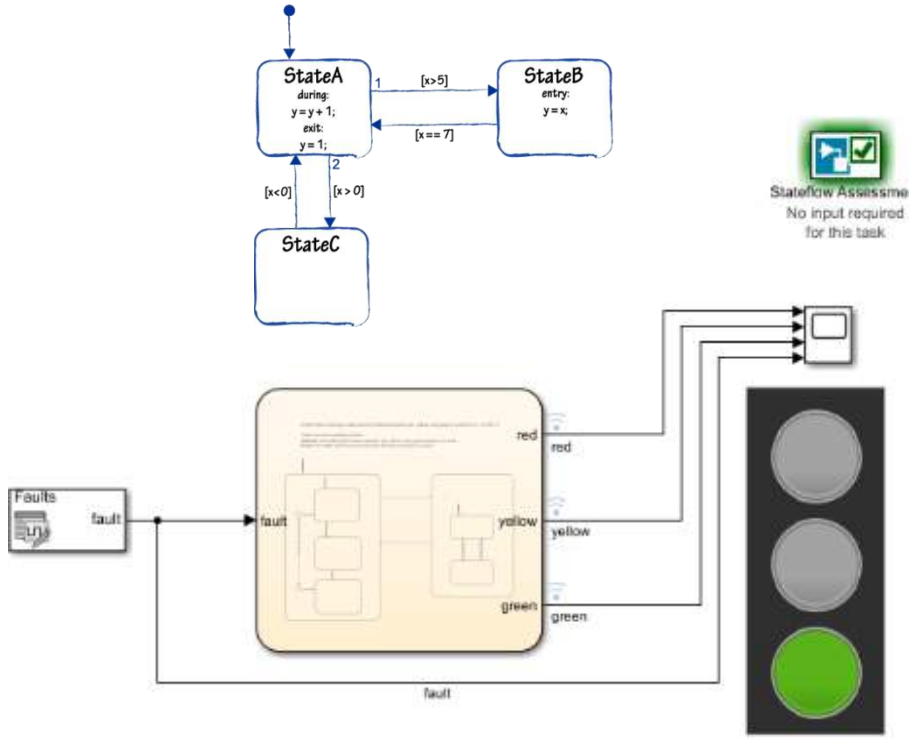
- Propulsion and Cooling Controller
- Based on State Machine:
  1. Propulsion Mode
  2. Battery Cooling Management
  3. Fuel Cell Cooling Management



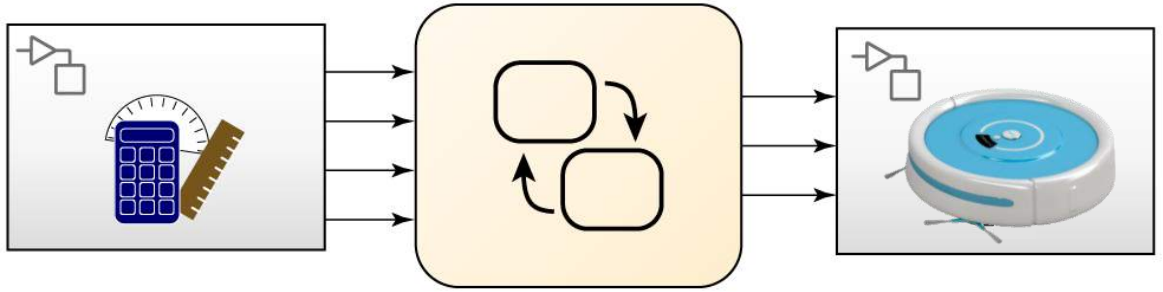

Dual_FC_Battery_Mobile * - Simulink
SIMULATION
DEBUG
MODELING
FORMAT
APPS
SUBSYSTEM BLOCK
Dual_FC_Battery_Mobile
Qbat
Load
T_Bat
T_FC
Fuel_Cell_On
Ramp
Udc
md
hc
Supervisory Logic
[FC_Logic]
[qH2]
[Udc]
[md]
Scopes
f(x) = 0
Solver Configuration
Bat_HT_unit
Battery Unit
Sensor (Bat)
DC-DC Converter (Battery)
Pcool
FC_HT_unit
DutyCycle
Sensor (L)
Three Drives
Multi-Drive
Ready
View 5 warnings
80%
auto(daessc)

Supervisory Logic Development: Stateflow Onramp

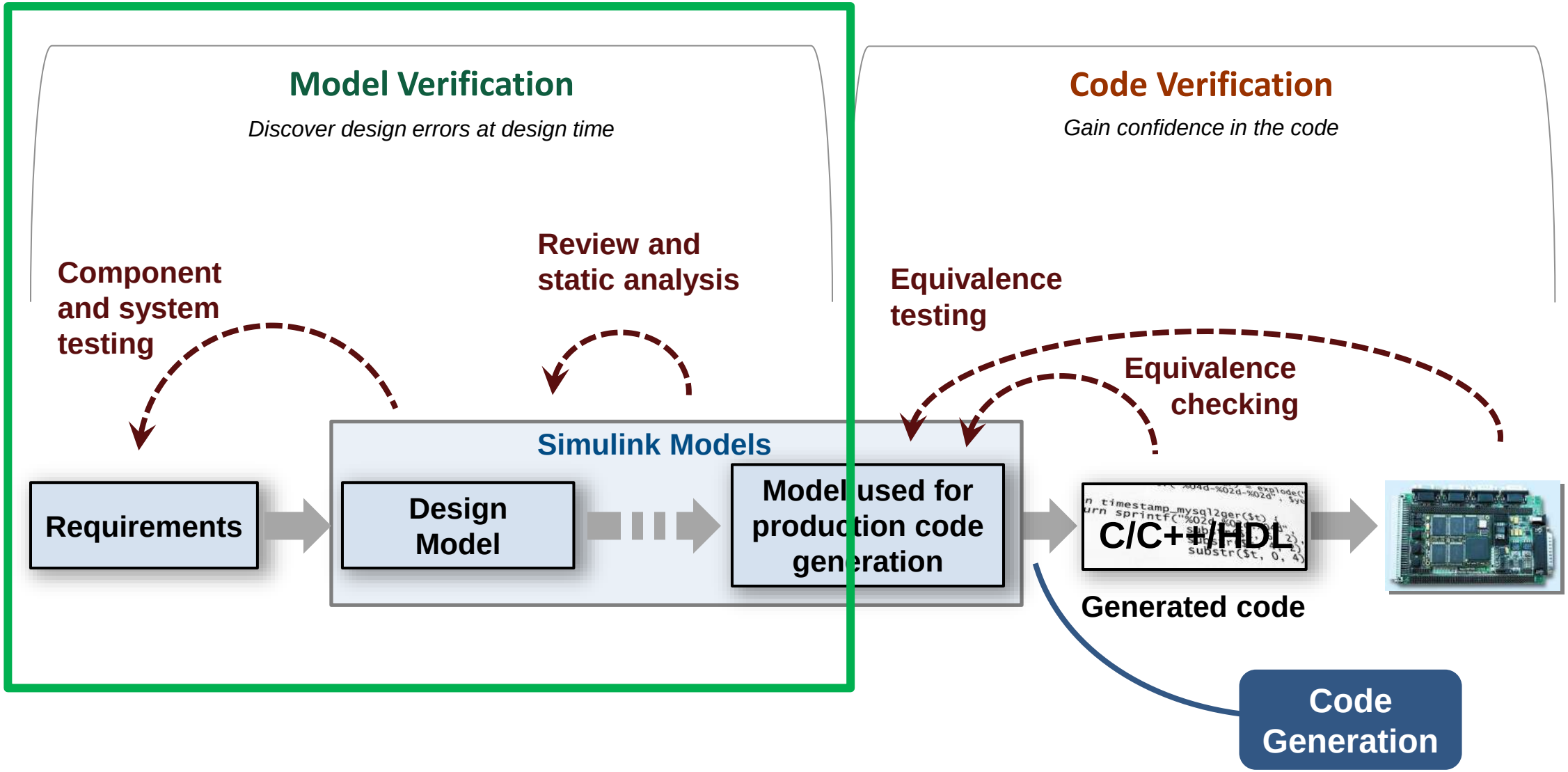
- Free, short course on the basics of using Stateflow in Simulink
- Short videos & hands-on exercises
- Automated assessments & immediate feedback



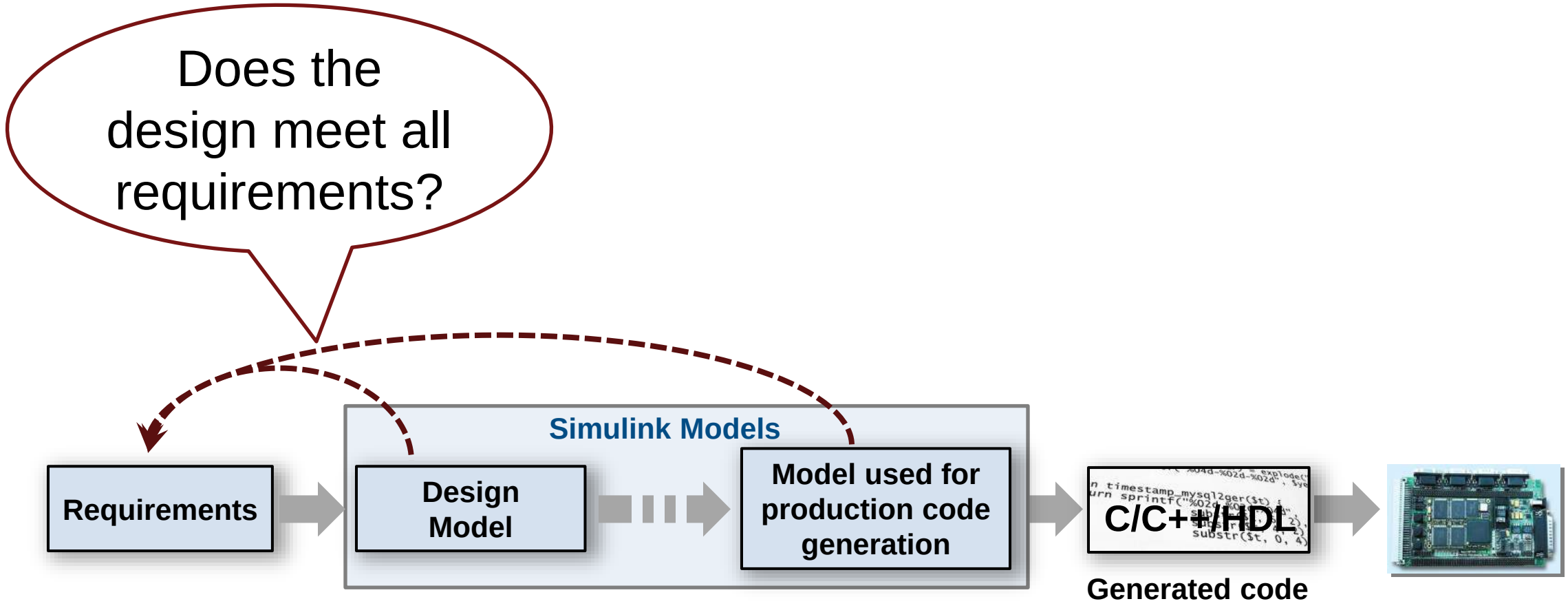
Visit https://www.mathworks.com/learn/tutorials/stateflow-onramp.html for more details!



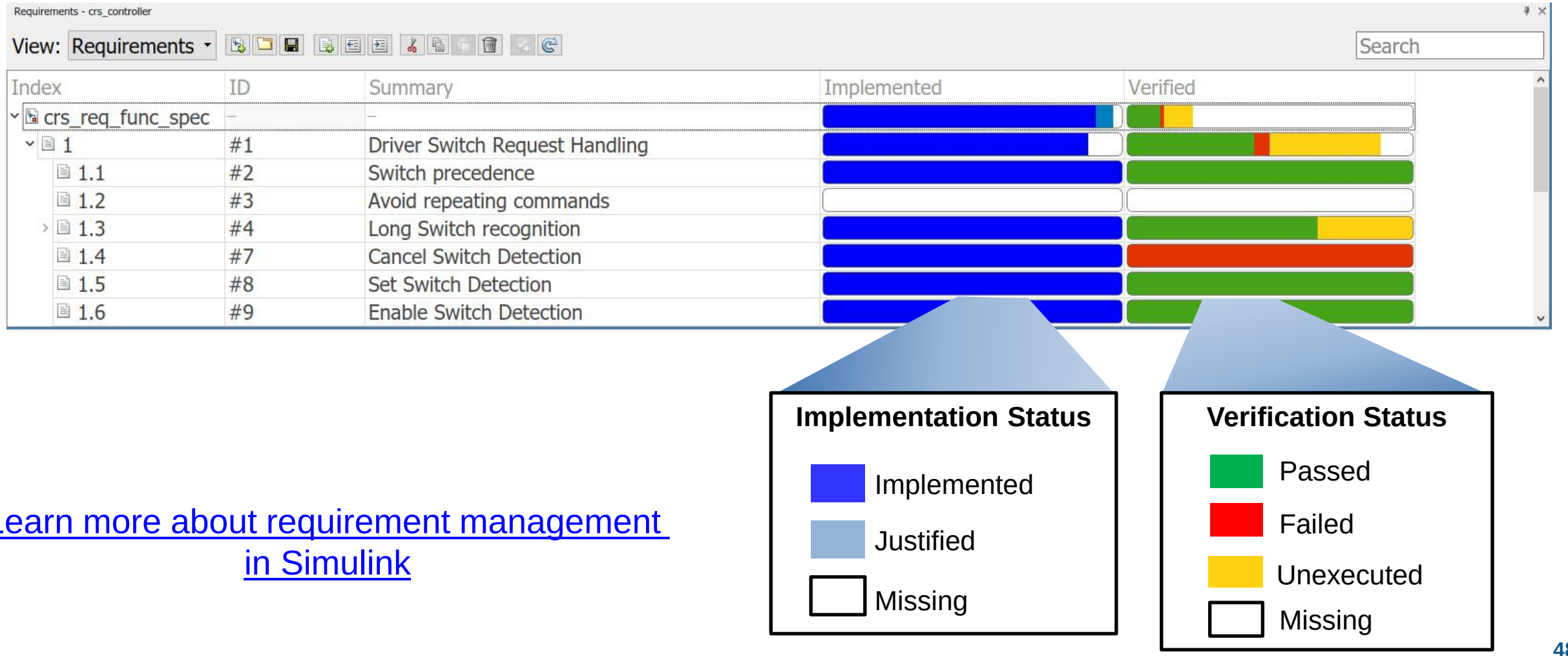
Model-Based Design Verification Workflow



How do we know we meet the requirements? Do requirements need to be updated?

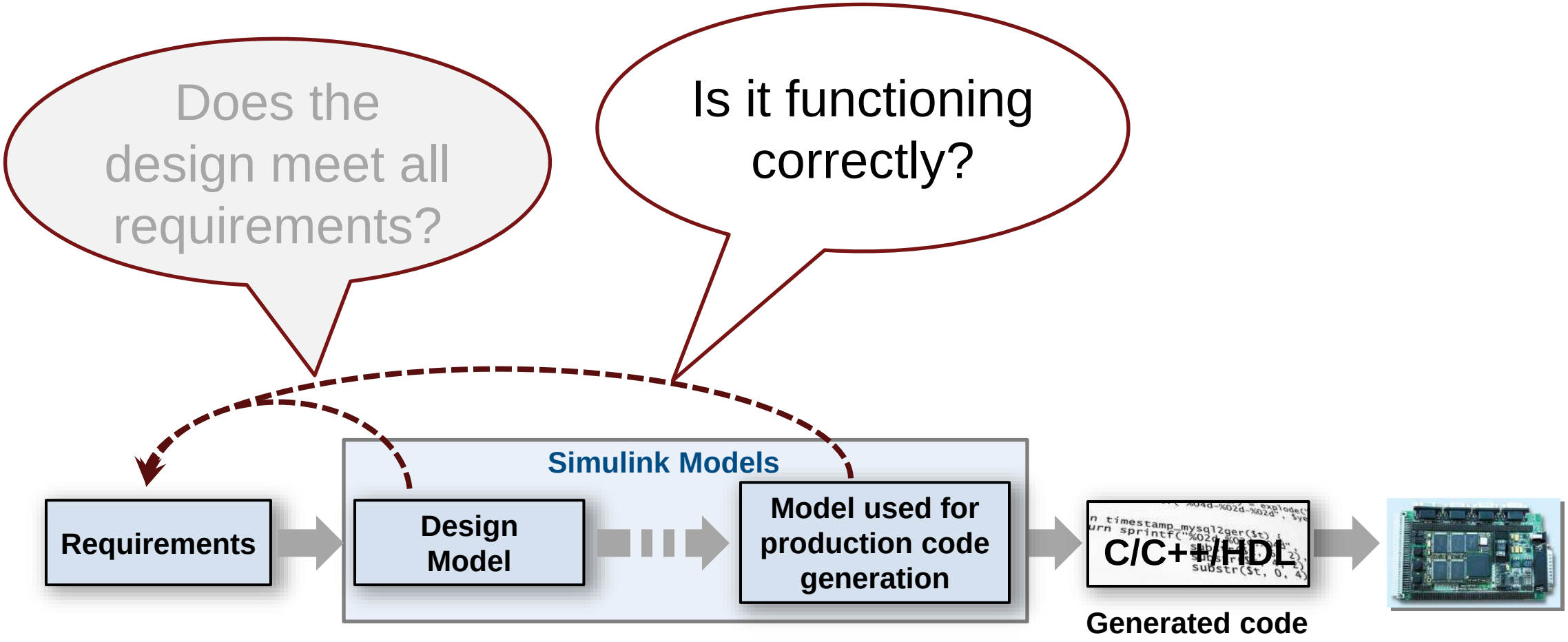


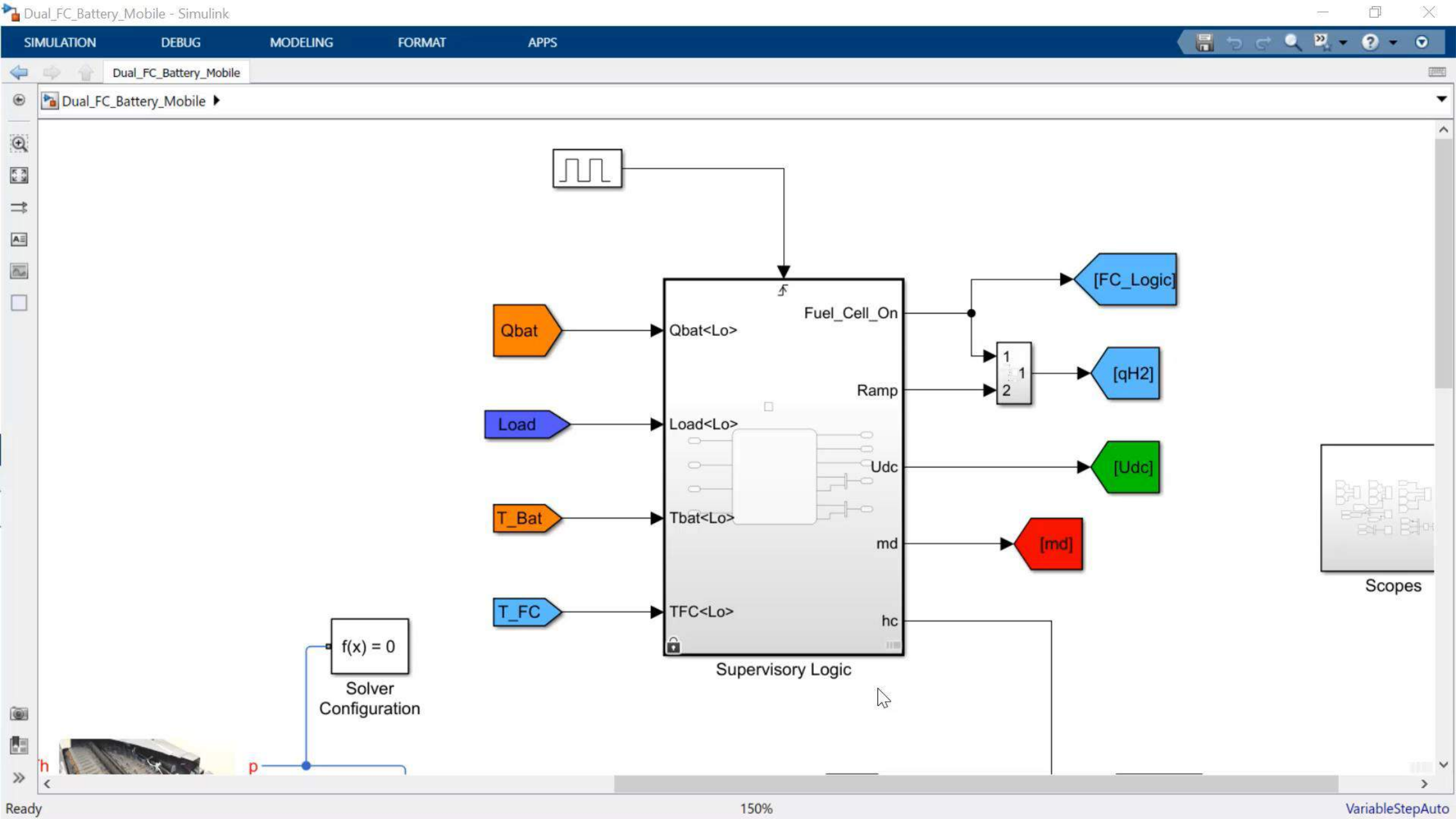
Track Implementation and Verification



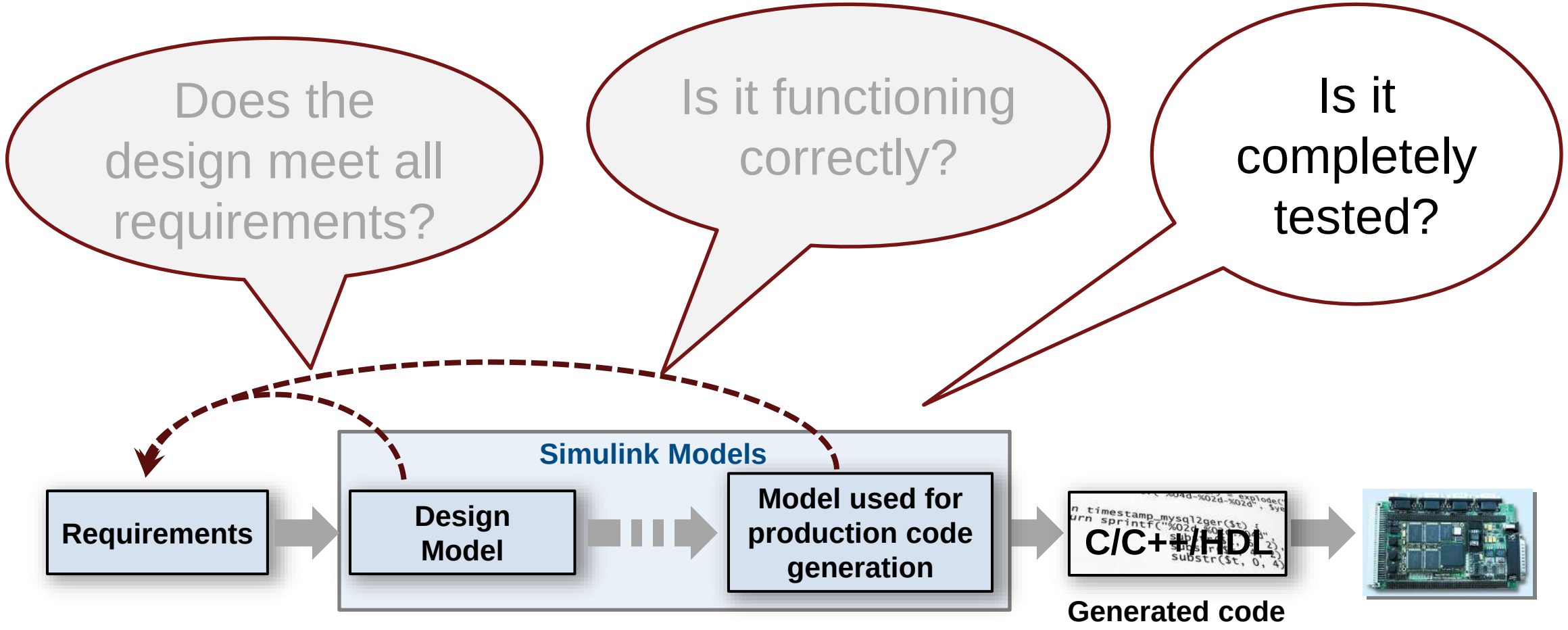
[Learn more about requirement management in Simulink]

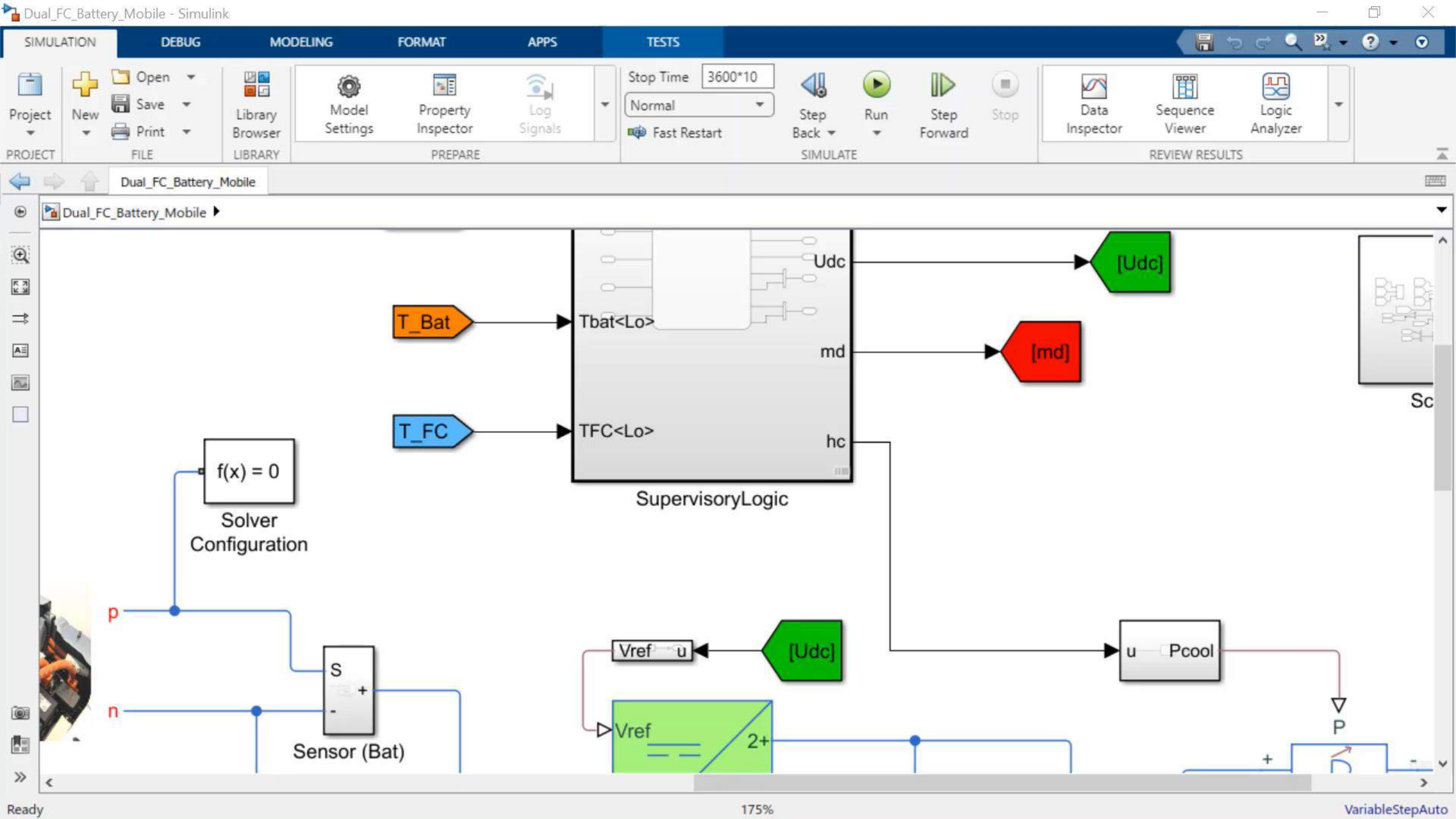
How do we test?



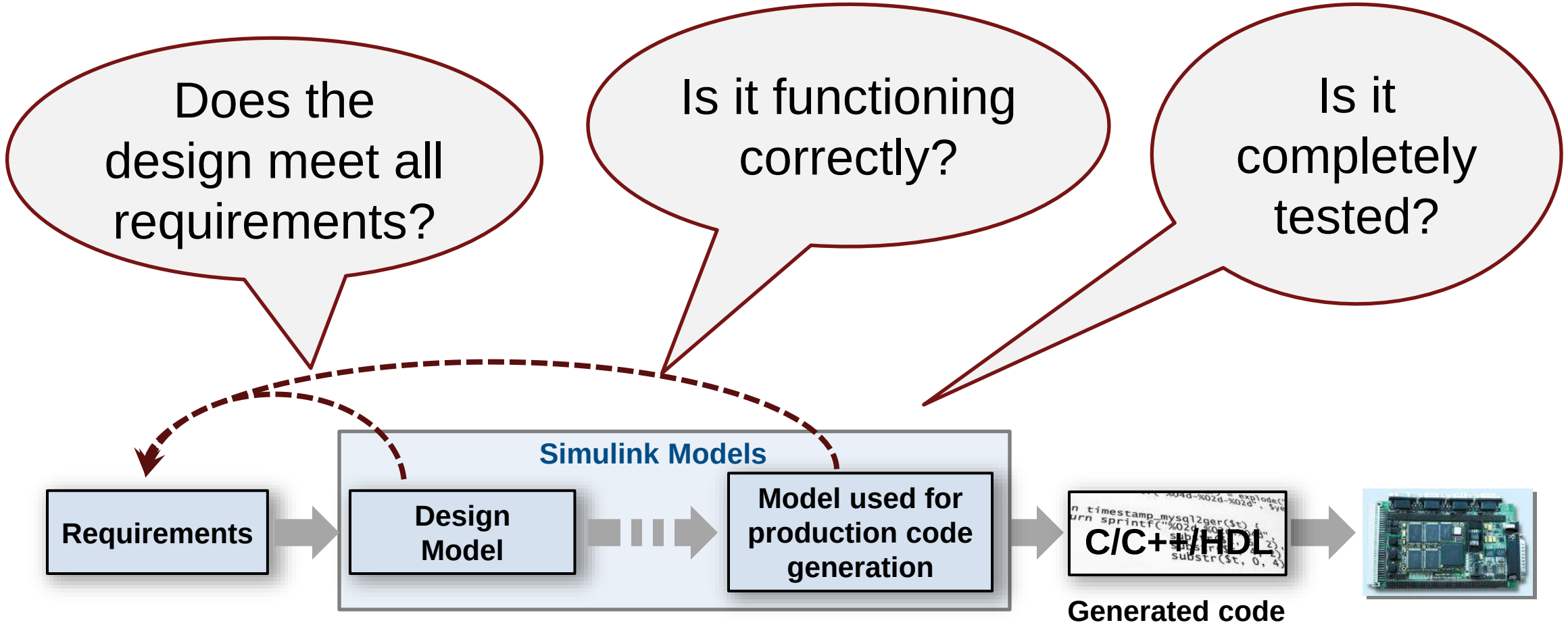

Dual_FC_Battery_Mobile - Simulink
SIMULATION
DEBUG
MODELING
FORMAT
APPS
Dual_FC_Battery_Mobile
Dual_FC_Battery_Mobile
Qbat
Load
T_Bat
T_FC
Qbat<Lo>
Load<Lo>
Tbat<Lo>
TFC<Lo>
Fuel_Cell_On
Ramp
Udc
md
hc
Supervisory Logic
[FC_Logic]
[qH2]
[Udc]
[md]
Scopes
f(x) = 0
Solver Configuration
Ready
150%
VariableStepAuto

How do we know we have enough test?




Dual_FC_Battery_Mobile - Simulink
SIMULATION
DEBUG
MODELING
FORMAT
APPS
TESTS
Project
New
Open
Save
Print
Library Browser
Model Settings
Property Inspector
Log Signals
Stop Time
3600*10
Normal
Fast Restart
Step Back
Run
Step Forward
Stop
Data Inspector
Sequence Viewer
Logic Analyzer
PROJECT
FILE
LIBRARY
PREPARE
SIMULATE
REVIEW RESULTS
Dual_FC_Battery_Mobile
T_Bat
T_FC
Tbat<Lo>
TFC<Lo>
Udc
md
hc
SupervisoryLogic
[Udc]
[md]
f(x) = 0
Solver Configuration
p
n
S
+
-
Sensor (Bat)
Vref
u
[Udc]
Vref
2+
u
Pcool
P
+
-
Ready
175%
VariableStepAuto

Systematic Simulation Testing

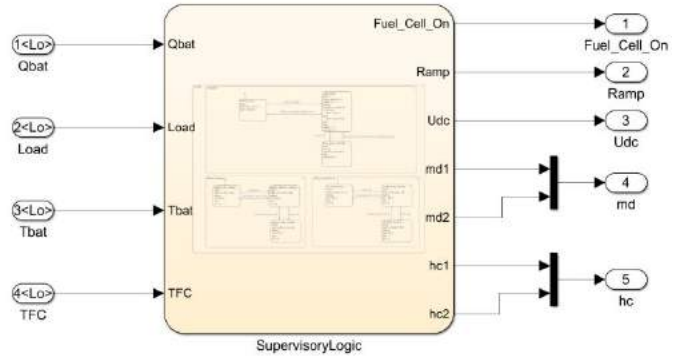


MATLAB and Simulink

For Verification, Validation and Test

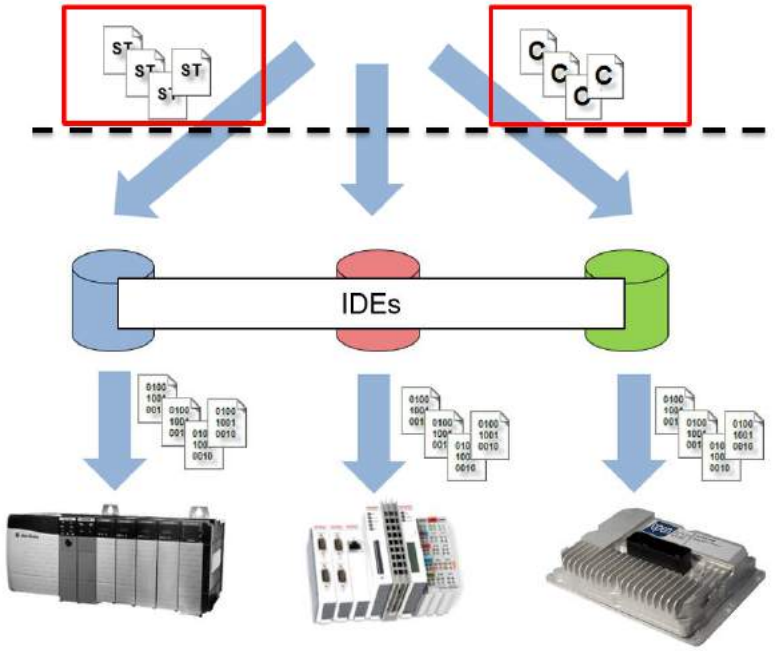
Algorithm model and deployment on real-time controller

- Automatic code generation from models
- Reduced coding time & errors
- Hardware independent source code
- Know-how captured in single source (model)

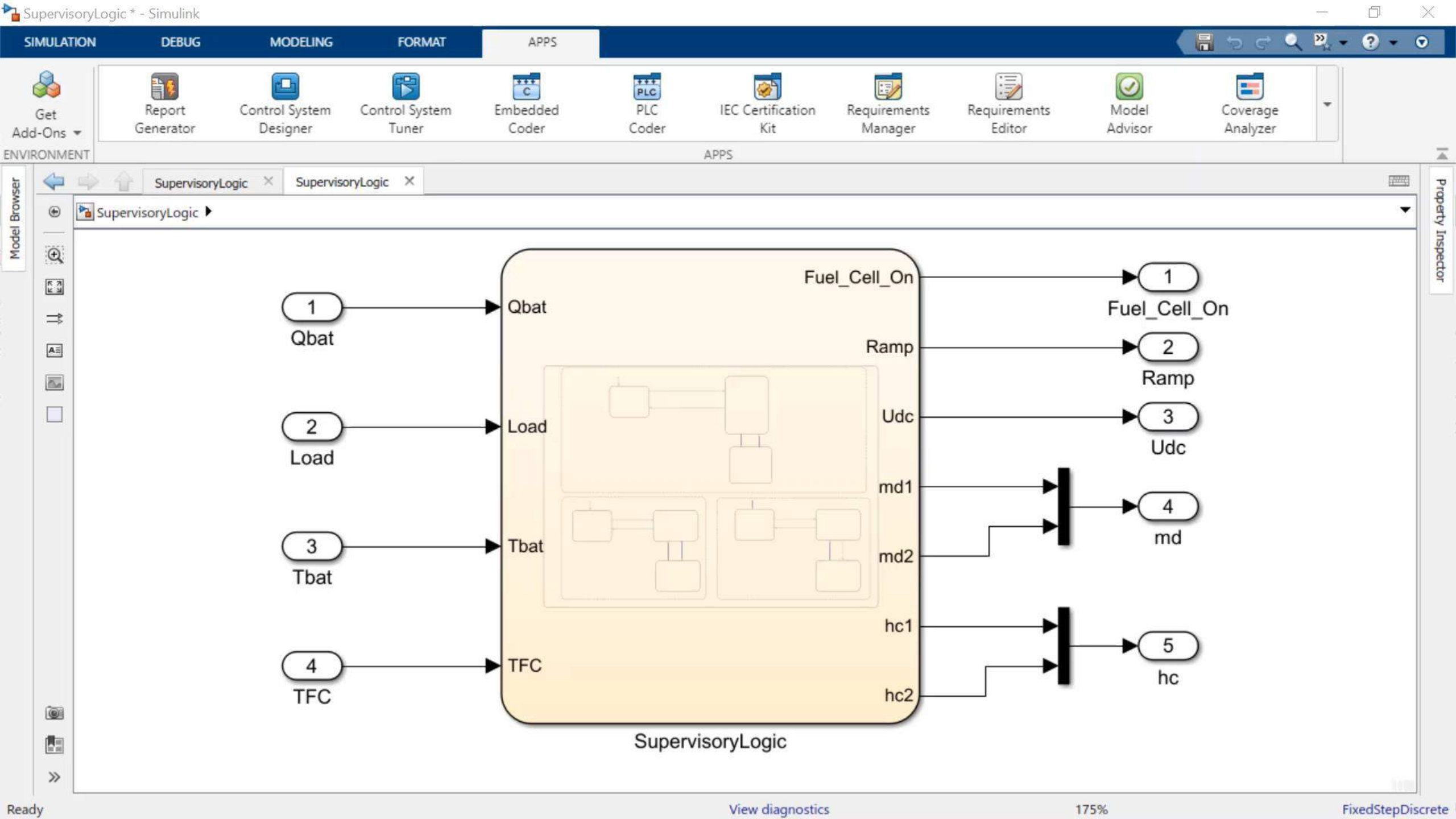


All relevant PLCs supported

Vendor	IDE	IEC 61131-3	C/C++	Connections Partner
3S - Smart Software Solutions	CODESYS	✓		✓
B&R Industrial Automation	Automation Studio	✓	✓	✓
Bachmann Electronic	SolutionCenter	✓	✓	✓
Beckhoff Automation	TwinCAT	✓	✓	✓
Bosch Rexroth	IndraWorks	✓	✓	✓
Mitsubishi Electric	CW Workbench		✓	✓
Omron	Sysmac Studio	✓		✓
Phoenix Contact	PC WORX	✓	✓	✓
Rockwell Automation	RSLogix / Studio 5000	✓		✓
Siemens	TIA Portal / STEP 7	✓	✓	✓

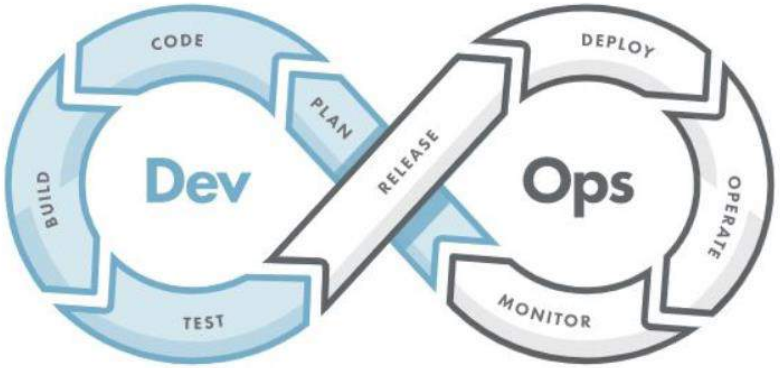


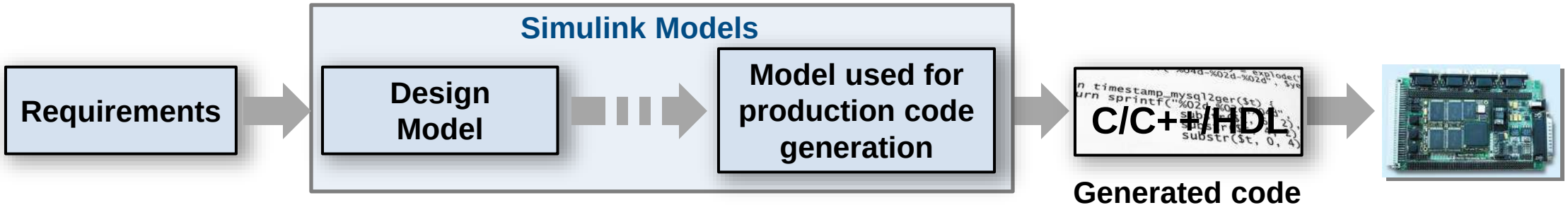
PLCs, ECU, custom hardware


SupervisoryLogic * - Simulink
SIMULATION
DEBUG
MODELING
FORMAT
APPS
Get Add-Ons
ENVIRONMENT
Report Generator
Control System Designer
Control System Tuner
Embedded Coder
PLC Coder
IEC Certification Kit
Requirements Manager
Requirements Editor
Model Advisor
Coverage Analyzer
APPS
Model Browser
SupervisoryLogic
SupervisoryLogic
SupervisoryLogic
1
Qbat
2
Load
3
Tbat
4
TFC
Qbat
Load
Tbat
TFC
Fuel_Cell_On
Ramp
Udc
md1
md2
hc1
hc2
SupervisoryLogic
1
Fuel_Cell_On
2
Ramp
3
Udc
4
md
5
hc
Property Inspector
Ready
View diagnostics
175%
FixedStepDiscrete

Beyond Functional Testing – Scale with Continuous Integration

Can we automate this?

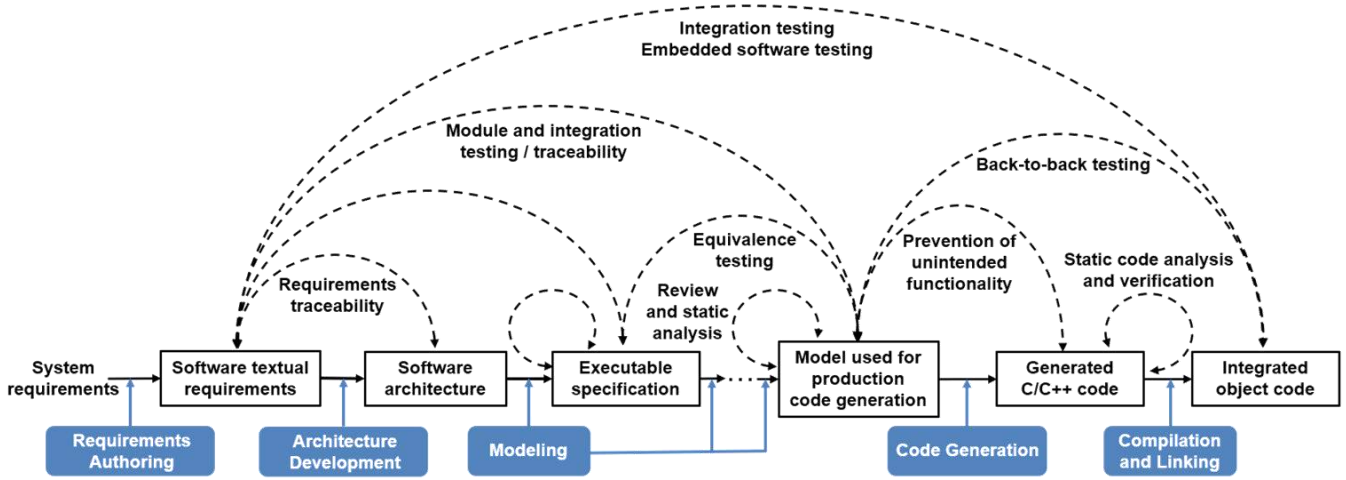




[Continuous Integration with MATLAB and Simulink]

Scalable to certification workflows ensuring highest quality & safety

- IEC 61508 - Safety-related systems
- ISO 26262 - Automotive / Motorcycle
- ISO 25119 - Agriculture and Forestry
- EN 50128 - Rail
- IEC 62304 - Medical
- IEC 61511 - Process Control
- DO-178 & DO-254

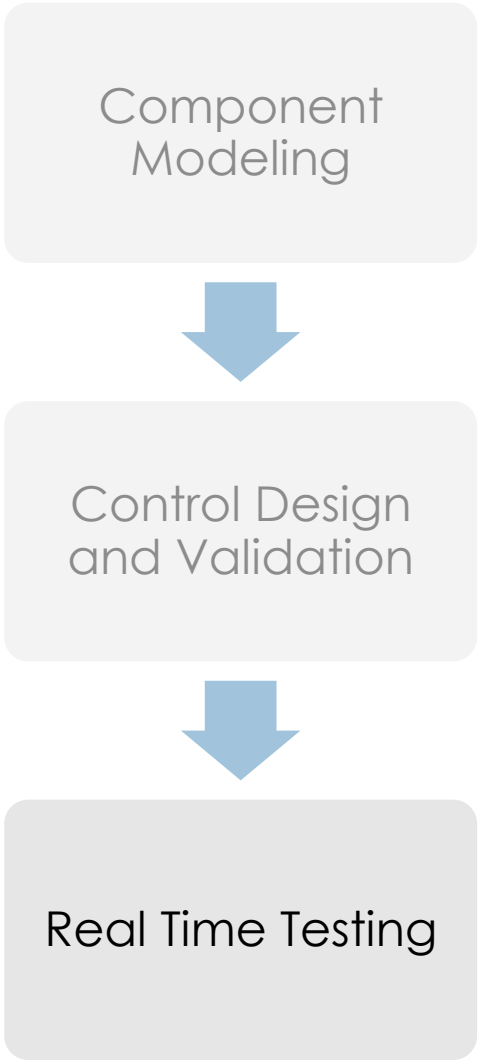


MATLAB and Simulink

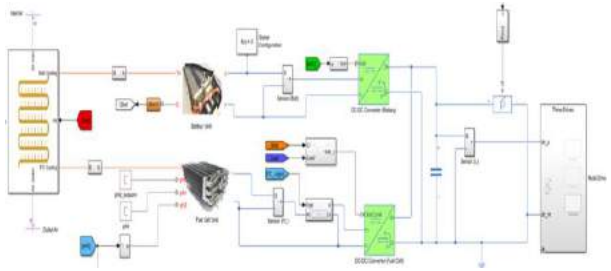
For Verification, Validation and Test

Agenda

- Multi-stack fuel cells and battery propulsion system
- **Supervisory logic development, verification and code generation**
- **Real-time testing and prototyping**



E-mobility - From concept to product



Physical model

Hardware emulator

Final system

Processor-in-the-Loop (PIL)

Hardware-in-the-Loop (HIL)

Productized Solution

Hardware Controller

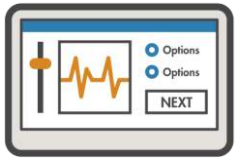
E-mobility / Real-Time Testing – Motivations

Accelerate Innovation (reduce time-to-market)

Prove new concepts (components, algorithms)

Detect Design Pitfalls (as early as possible)

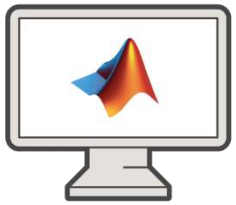
Test challenging conditions (inject faults, harsh environments)



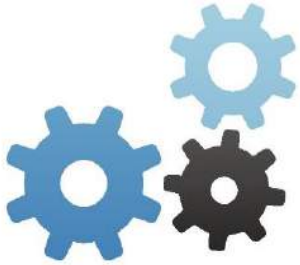
Flexible Testing (extensive, low cost, no risk)

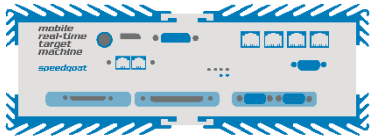
Perform safe tests (no risk for equipment or operators)

E-mobility / Real-Time Testing – Integrated Software-Hardware



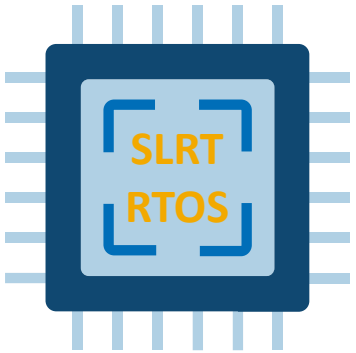
MathWorks
Simulink Real-Time





Speedgoat
Real-time Target Machines

- RT - instrumentation
- Code Gen (C/VHDL)
- Simscape
- Simulink Test



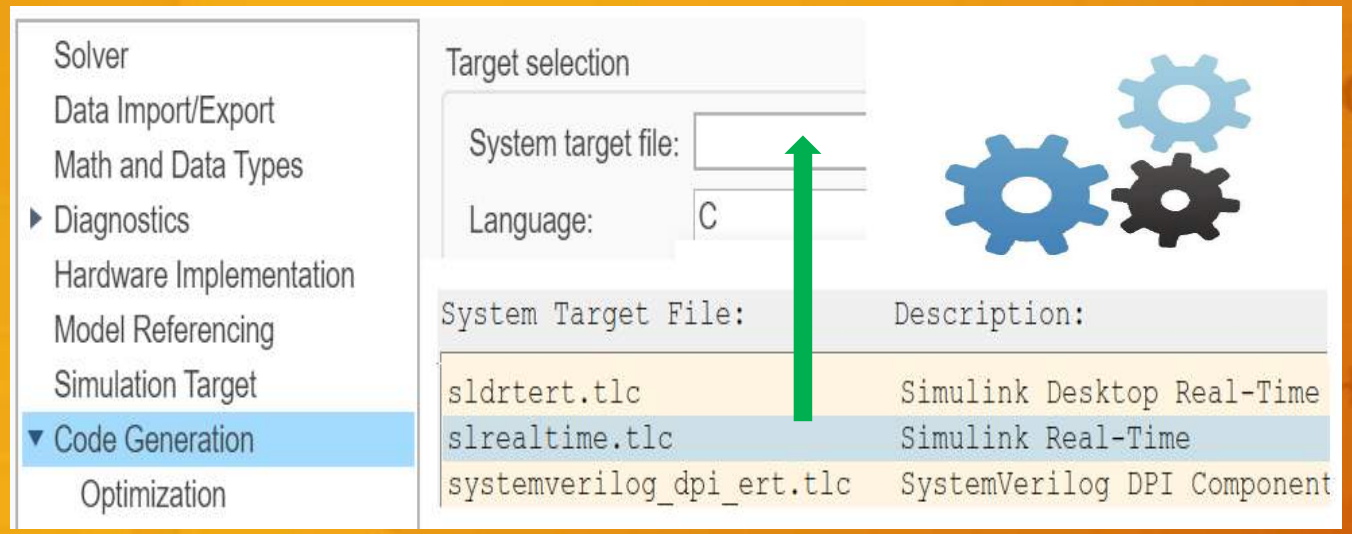
- I/O protocol support
- FPGA-based solutions
- Speedgoat driver library
- Complete HIL-Rigs

E-mobility/Real-time Testing – Value of turnkey solution

One team MathWorks – Speedgoat (support)

Timely software upgrades (functionality)

Smooth on-ramping (productivity)

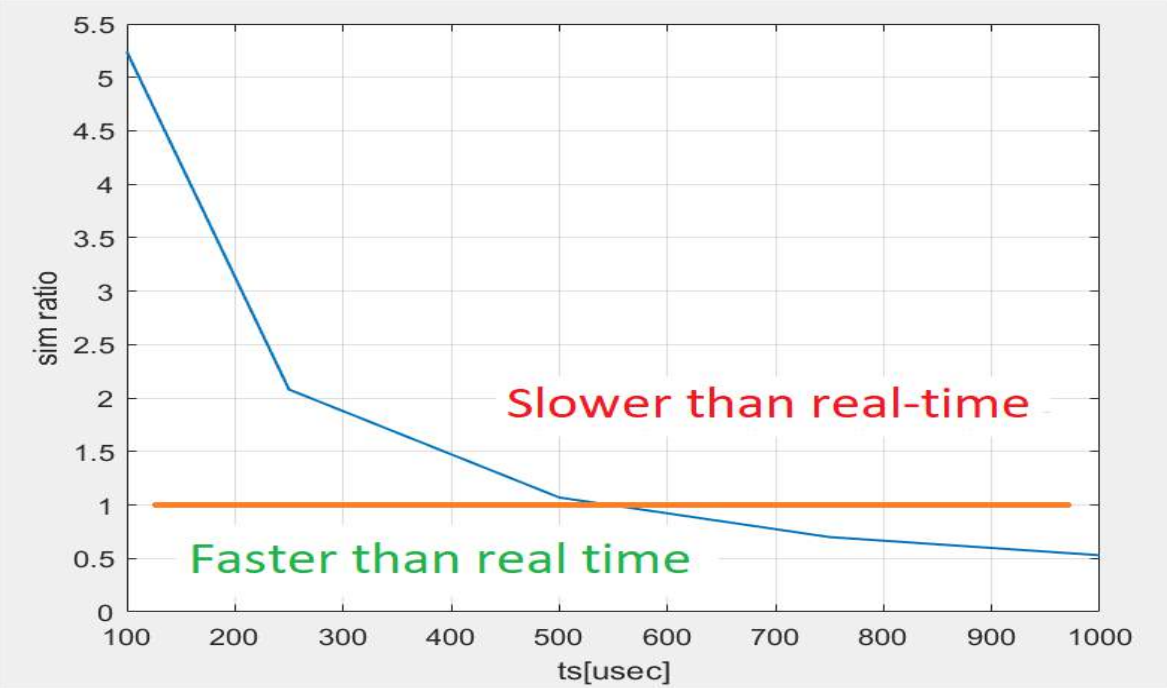


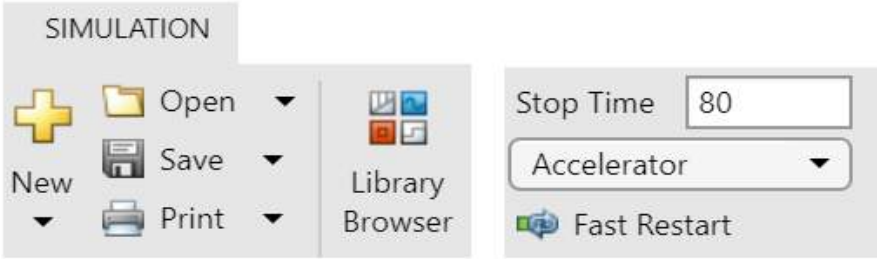
E-mobility / Real time testing – General purpose vs. high end HIL

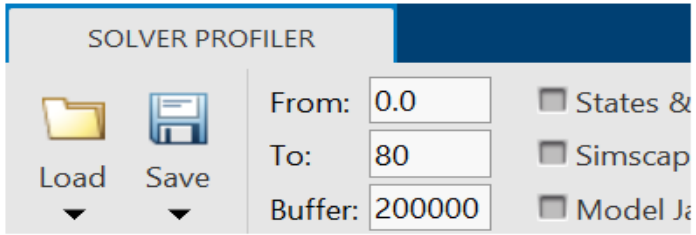
- General purpose (mechatronics, energy management)
  - Relevant physical behaviour (Ts > 50 us)
  - C-based processors are sufficient for HIL hardware
  - Embedded code generation (C)

- High-end applications (focus on power electronics)
  - Modulation, harmonics, faults (Ts < 5us)
  - FPGA devices are needed to execute the physical equations in real-time

E-mobility/Real-time Testing – From desktop to HIL



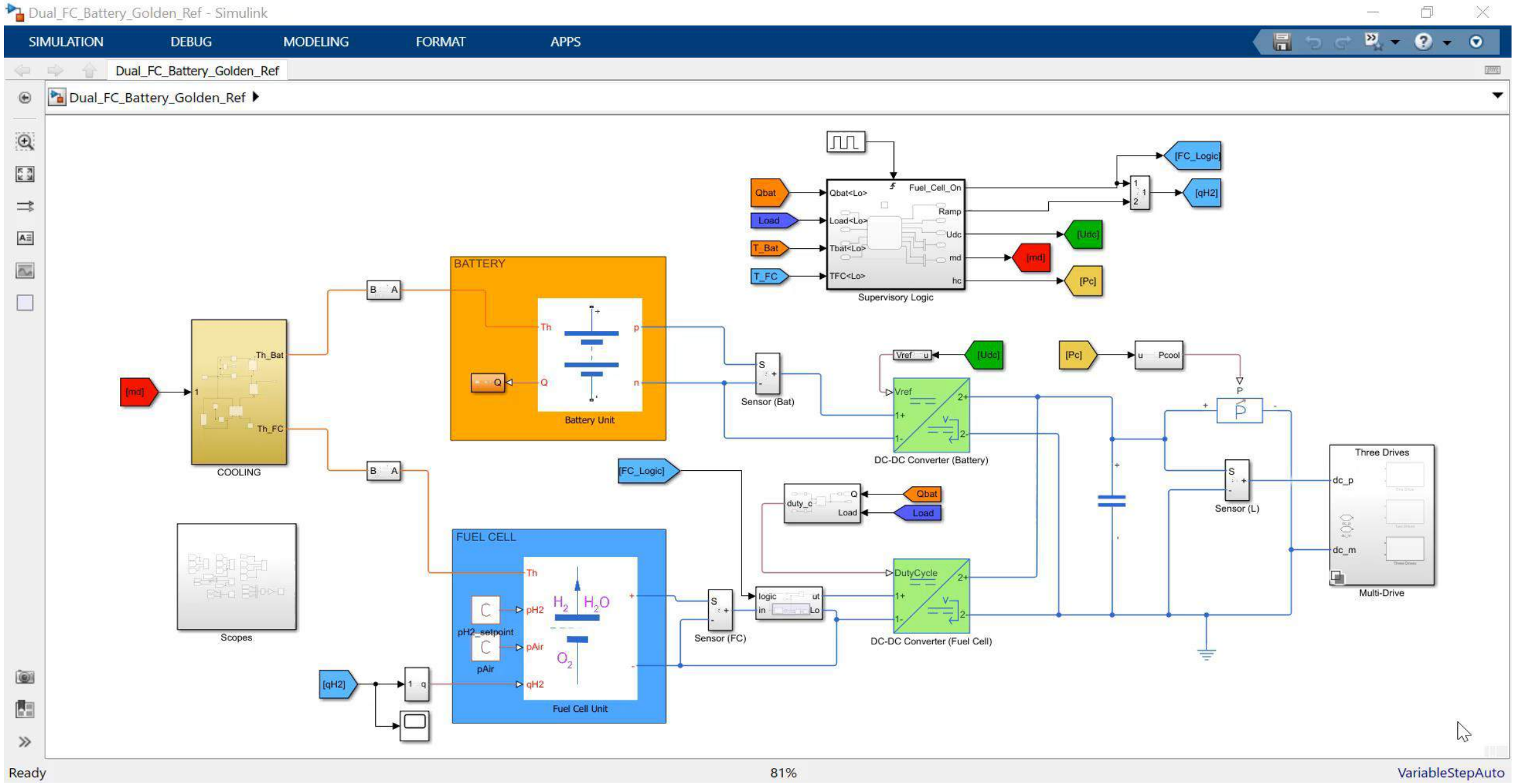




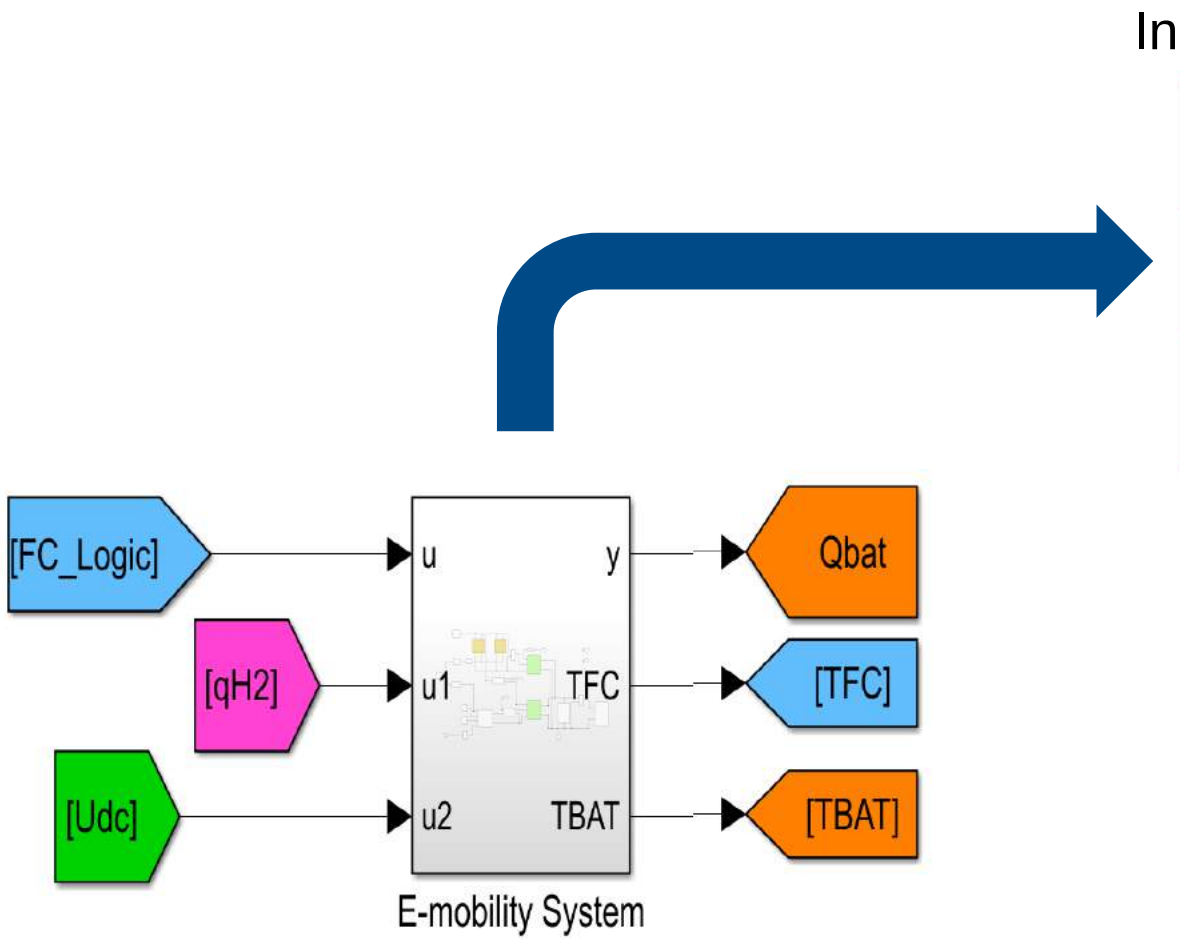
STEP INFORMATION	
Max step size	N/A
Min step size	N/A
Average step size	1.00e-03
Max step size usage(%)	N/A
Total steps	80000
Run time(s)	48.51
Run/sim time ratio	0.61

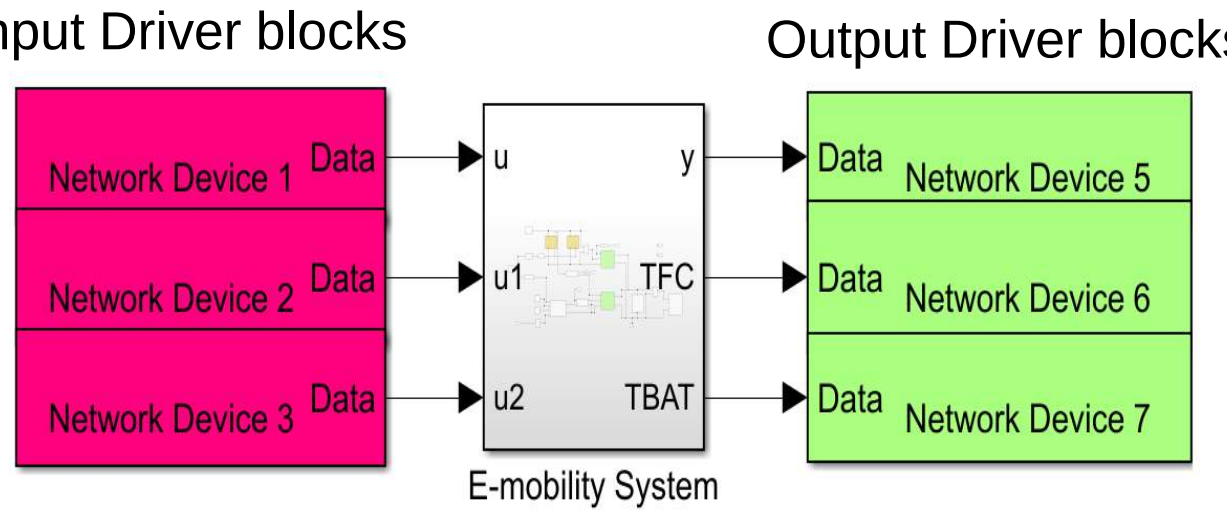
Time step in HIL models
shall enable fast real-time performance
while ensuring robustness
and sufficient accuracy.

E-mobility/Real-time testing – From desktop to HIL



E-mobility/Real-time testing – From desktop to HIL



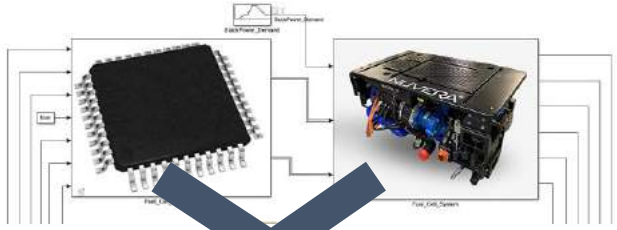


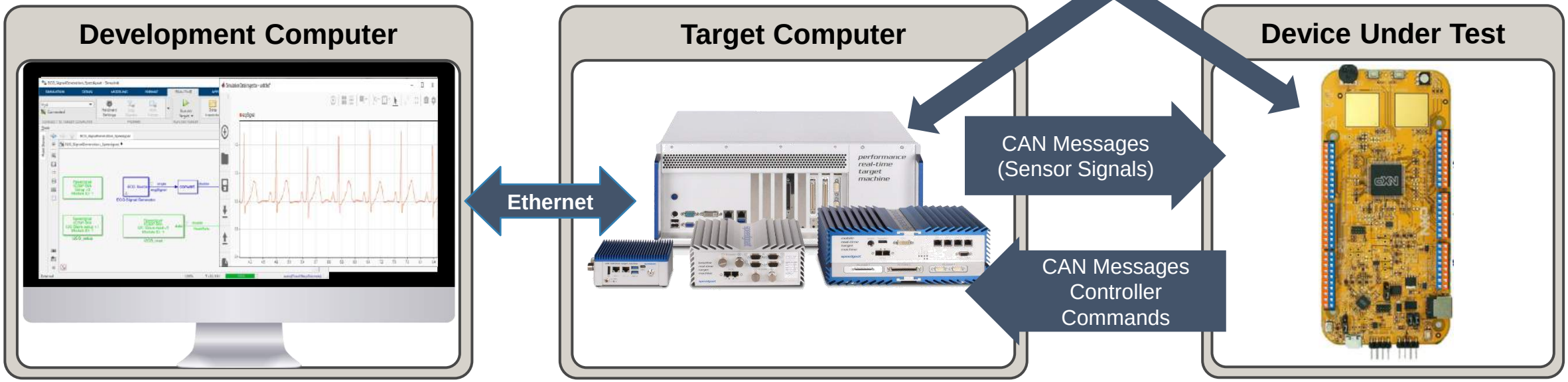
Configure the physical model
with the required I/O driver blocks
meeting your requirements.

….

and generate code
(slrealtime.tlc)

Hardware Setup





- MATLAB
- Simulink
- Simscape …
- Simulink Real-Time
- Code Generation Tools

- Multicore CPU
- IO 613
  - (2 Channel CAN Bus)
- IO397
  - ADCs, DIOs, PWMs
  - I2C, SPI

- NXP S32K142 Eval Board

Nuvera: Hydrogen Fuel Cells Reduce CO_2 Emissions

One of Nuvera's E-Series Fuel Cell Engines. Image credit: Nuvera Fuel Cells.

…. In simulation, Nuvera tests the system in low and high ambient temperatures, and in low- and high-humidity environments..

…. In order to experiment with their algorithms in a more realistic setting, **Nuvera does hardware-in-the-loop testing**: They load their engine model onto a custom computer made by **Speedgoat** that is tailored to have the same inputs and outputs as the physical engine, and can simulate its operation in real time. The same embedded computer that runs the fuel cell engine is connected to the Speedgoat box and is programmed from C code generated from Simulink.
This setup **adds rigor** while enabling Nuvera's engineers to **iterate on their design quickly**. It also allows for experimentation **without putting a real engine at risk**...

Model-Based Design (Simulink) yields strong ROI

Model-Based Design ROI

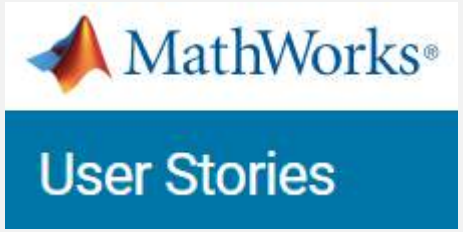


ABB: **Four times more projects** per same headcount

Airbus: **Software testing time cut by two-thirds**

BAE: **One-and-a-half to two times more efficient** than hand-coding

Continental: Six months of effort eliminated; **verification time cut by 50%**

Honeywell: **A five-to-one improvement** in productivity

Lear: Development time cut by 40%; **Zero warranty issues** reported

Lockheed Martin: **Development efficiency doubled**; Design updated in a day

SEGULA Technologies Helps Fuel the Future of Transportation with Model-Based Design

Challenge

Accurately validate hydrogen fuel cell design before testing with hardware prototypes.

Solution

Use Model-Based Design with MATLAB, Simulink, and Simscape to develop models that simulate the physics inside the fuel cell and enable the team to iteratively evolve fuel cell system models.

Key Outcomes

- Reduce development time for each new design by four to six weeks by starting with a Simscape model
- Simulate the fuel cell operation in a typical week or up to its 30,000-hour lifetime with hardware-in-the-loop (HIL) testing and Simulink Real-Time
- Reduce time to market and save money by using system-level simulation in Simulink to validate the design before testing with hardware prototypes

A hydrogen fuel cell stack on a test rig in the SEGULA lab. (Image credit: SEGULA Technologies)

"Starting with a Simscape model shaves four to six weeks off of the initial development time." — Dirk Rensink, technical lead for fuel cell simulation, SEGULA Technologies

Link to feature story

Plug Power Accelerates Fuel Cell Control Development

Challenge

Shorten time to market and reduce operational costs in fuel cell control development

Solution

Use MathWorks tools to model systems and rapidly test new algorithms through simulation

Results

- Shorter development time
- Increased process efficiency
- Reduced operating expenses

Plug Power fuel cell system.

"We don't have time to investigate our algorithms with C or C++. Fortunately, MATLAB lets us test our ideas with just a few lines of code. It saves a lot of time, and moves us toward our goal of creating a commercially viable onsite energy system."

- Rebecca Dinan, Plug Power

Link to user story

Simulating Fuel Cell Hybrid Bus Technology at the University of Delaware

Challenge

Design a low-cost, zero-emission, fuel cell hybrid bus

Solution

Use MATLAB and Simulink to model, simulate, and optimize the design

Results

- Multiple design ideas explored without using hardware
- Simulation results accurate to within 5% of measured data
- Optimal combination of performance and efficiency identified

The University of Delaware series-hybrid fuel cell bus.

"Simulink enables us to answer design questions that would be extraordinarily expensive to answer via trial-and-error hardware iterations."

- Juha Inberg, Ponsse

Link to article

University of Waterloo Develops Award-Winning Fuel Cell Technology

Challenge

- Re-engineer a sport utility vehicle to optimize fuel efficiency without compromising performance

Solution

- Use MathWorks products and Model-Based Design to design and test a fuel cell vehicle propulsion system

Results

- Ease of communication
- Substantial design time savings
- Innovative technology

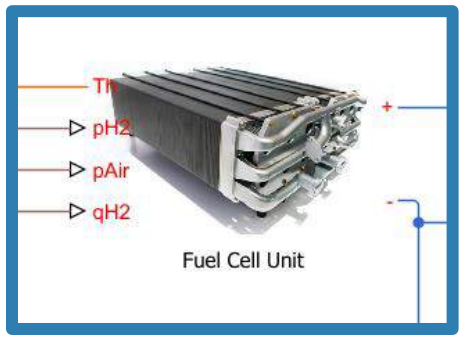
Link to user story

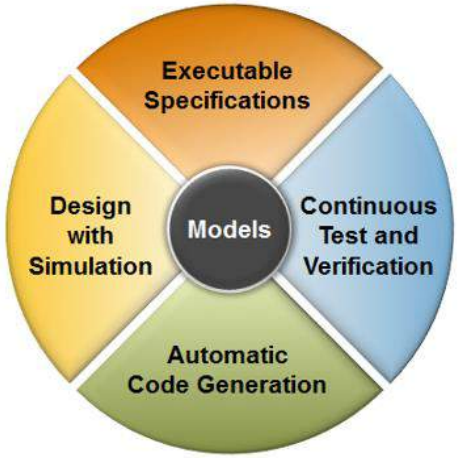
University of Waterloo demonstrating at Challenge X.

"Without Simulink, I don't believe there would have been any way to develop executable specifications to the same level of detail or use Model-Based Design so extensively."

Matthew Stevens
University of Waterloo

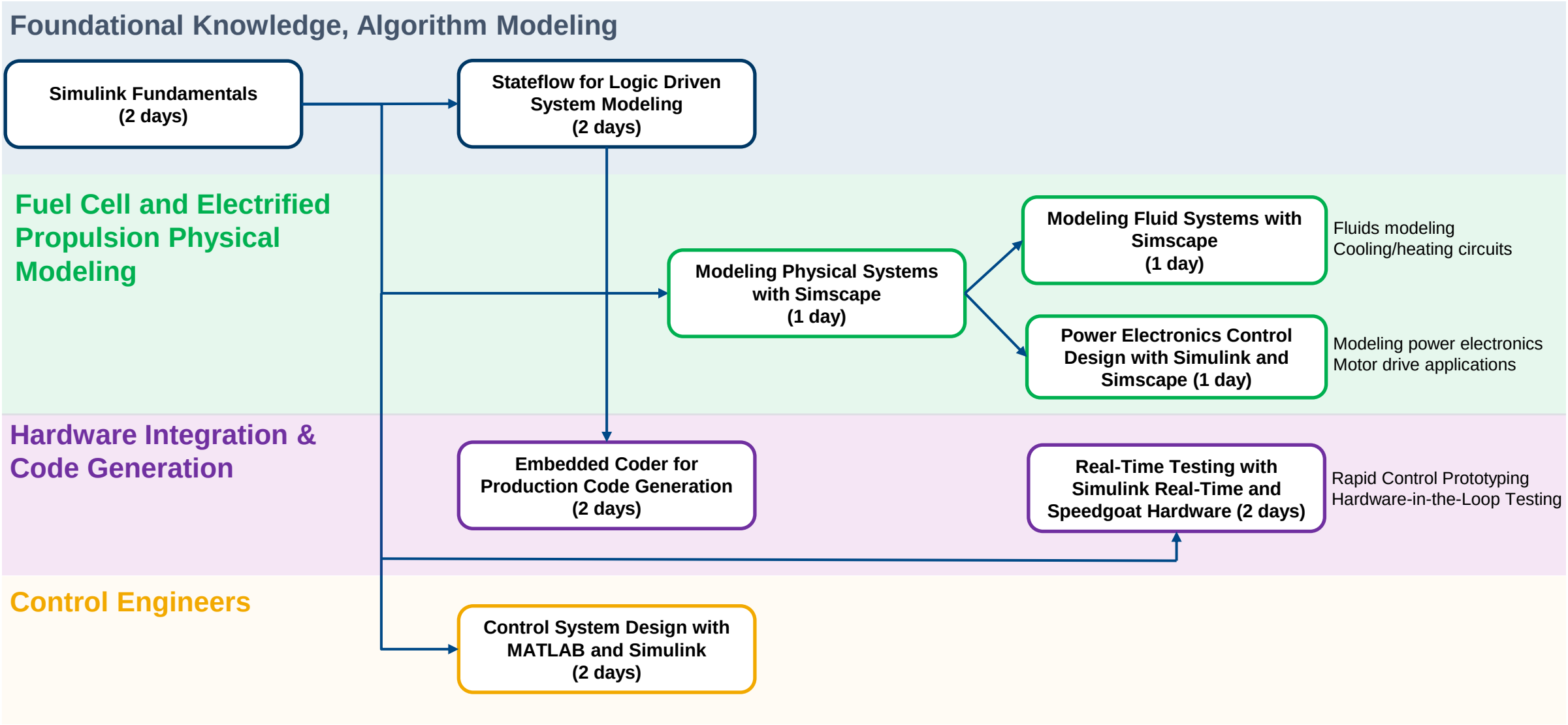
Summary





- **Electrified propulsion system: fuel cells + battery**
  - multi-domain interactions (electric – thermal – mechanic)
  - performance trade-offs (battery - fuel cell stack configuration)
- **Supervisory logic development**
  - agile workflow from concept to certification
  - embedded target support (code generation)
  - systematic validation
- **Real-time testing and prototyping**
  - model once – use twice (virtual testing)
  - add realism without risk

Learning paths: Electrified Propulsion Curriculum



Click on the blocks for more information
For full course schedule: mathworks.com/training

Additional resources at *www.mathworks.com*

- Solutions
  - https://www.mathworks.com/solutions/power-electronics-control
  - https://www.mathworks.com/solutions/power-system-analysis-and-design
  - https://www.mathworks.com/solutions/physical-modeling

- Training
  - https://www.mathworks.com/learn/tutorials/simscape-onramp
  - https://www.mathworks.com/learn/tutorials/circuit-simulation-onramp

- User Testimonies
  - https://www.mathworks.com/company/mathworks-stories/moxie-converts-mars-co2-to-oxygen
  - https://www.mathworks.com/company/user_stories/search.html

Thank you for your attention!

Any questions?