

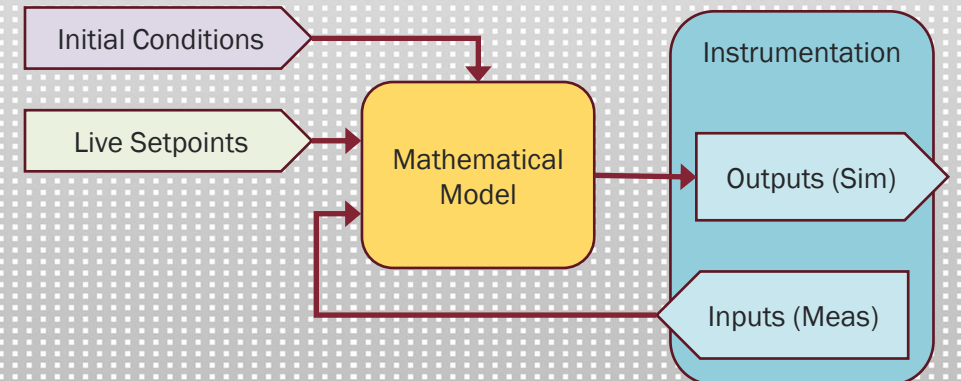


Demystifying BMS Hardware-In-the-Loop (HIL) Testing

Grant Gothing, Chief Technology Officer

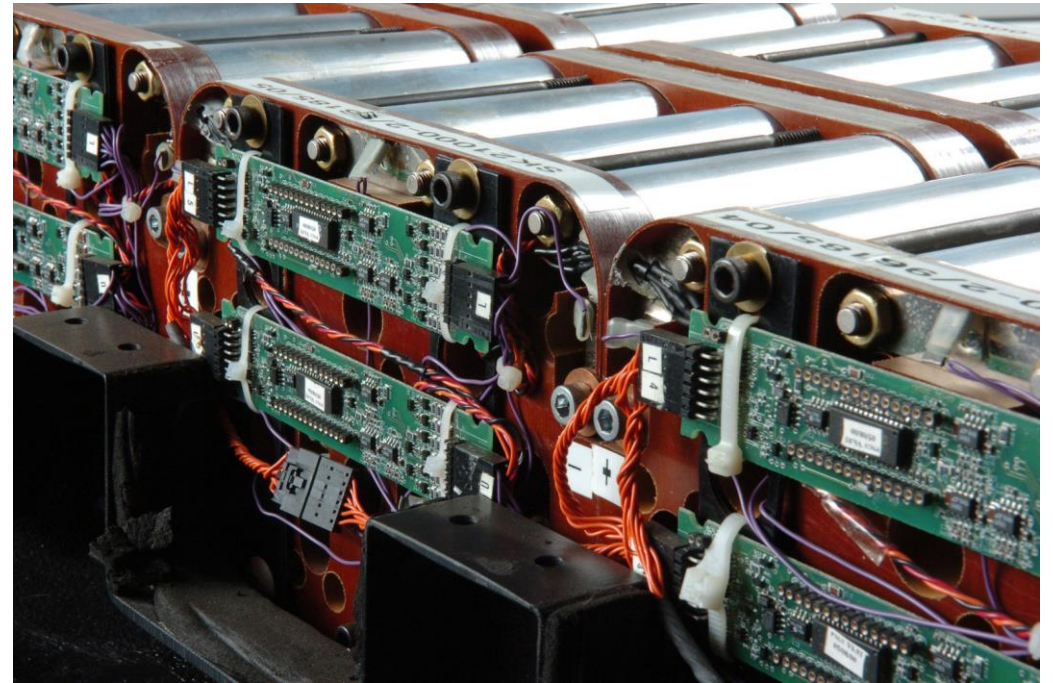
Peter Blume, President

June 26, 2020



Overview

- Introduction
- BMS HIL background
- Test methodologies
- Instrumentation and connectivity
- HIL test software for IO control
- User interfaces
- Test scripting & data logging
- BMS HIL test scenarios
- Q&A



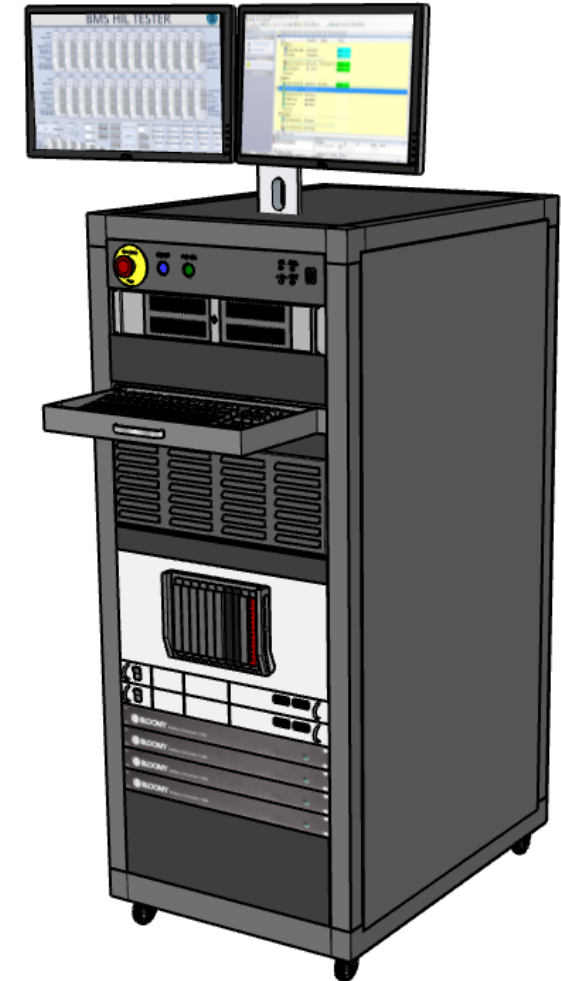
BLOOMY QUICK FACTS

- Founded in 1992
- Automated testing equipment
 - Battery test & simulation (BTS)
 - Trans/Aero/Def Simulation systems (SIMS)
 - Universal manufacturing electronics functional test (EFT)
- NI Platinum Alliance Partner
 - Published “The LabVIEW Style Book” © 2007, Prentice Hall
 - “BMS HIL Test System Helps JLR Shorten Time-to-Market” Graphical Systems Design Award in 2016



Bloomy BMS HIL Tester

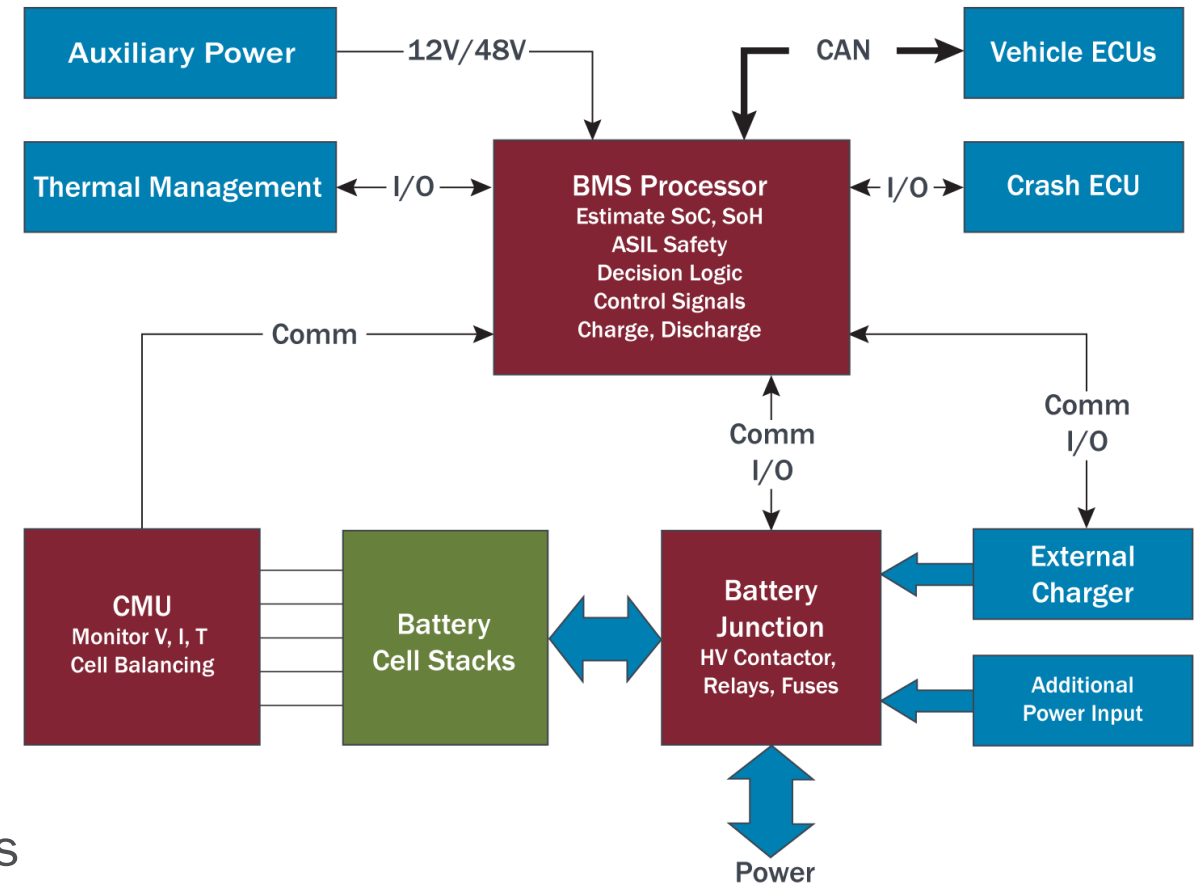
- Modular hardware and software
- Open platform
 - Based on COTS components
 - Bloomy BS1200 Cell Simulator
 - Maintainable & modifiable
 - Design documentation & source code provided
- Built on over a decade of BMS testing experience



What is the BMS?

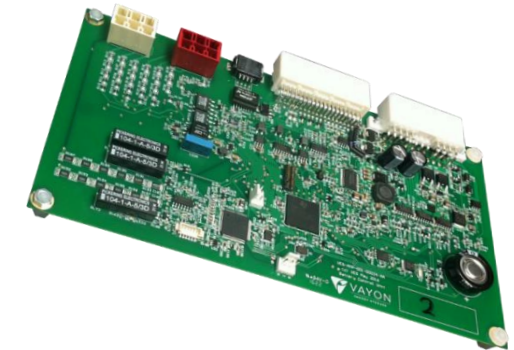
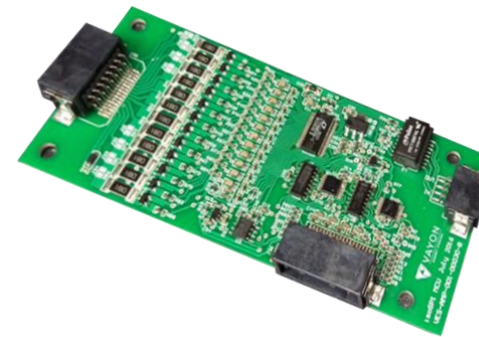
Battery Management System

- Monitor cell V, I and T
- Estimating the SoC and SoH
- Controlling the rate of charging and discharging
- Cell balancing
- Monitoring alarm levels and pack voltage
- Managing critical safety functions
- Controlling the contactor(s)
- Communicating to the ECM and other ECUs



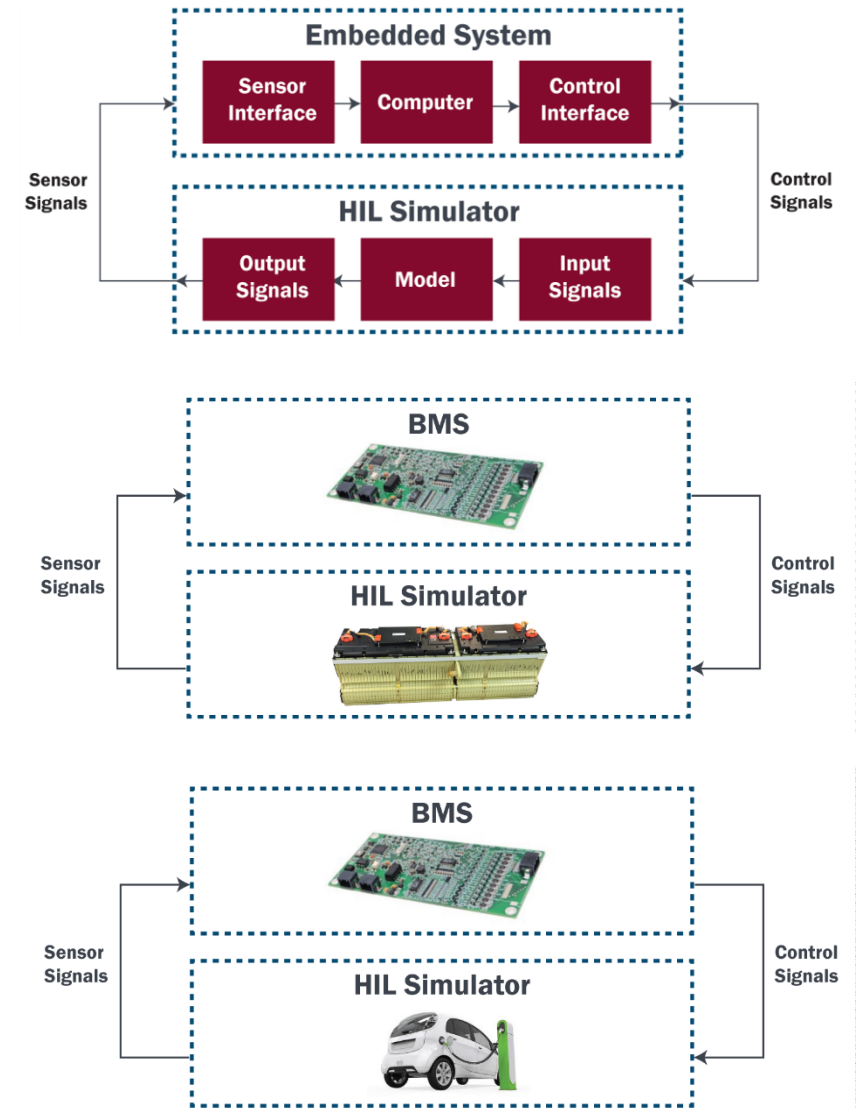
BMSs come in many different flavors

- Module and pack configuration
- Topology, series/parallel
- BMS optimized for end user profile
 - EV, grid, UPS, consumer electronics



What is HIL testing?

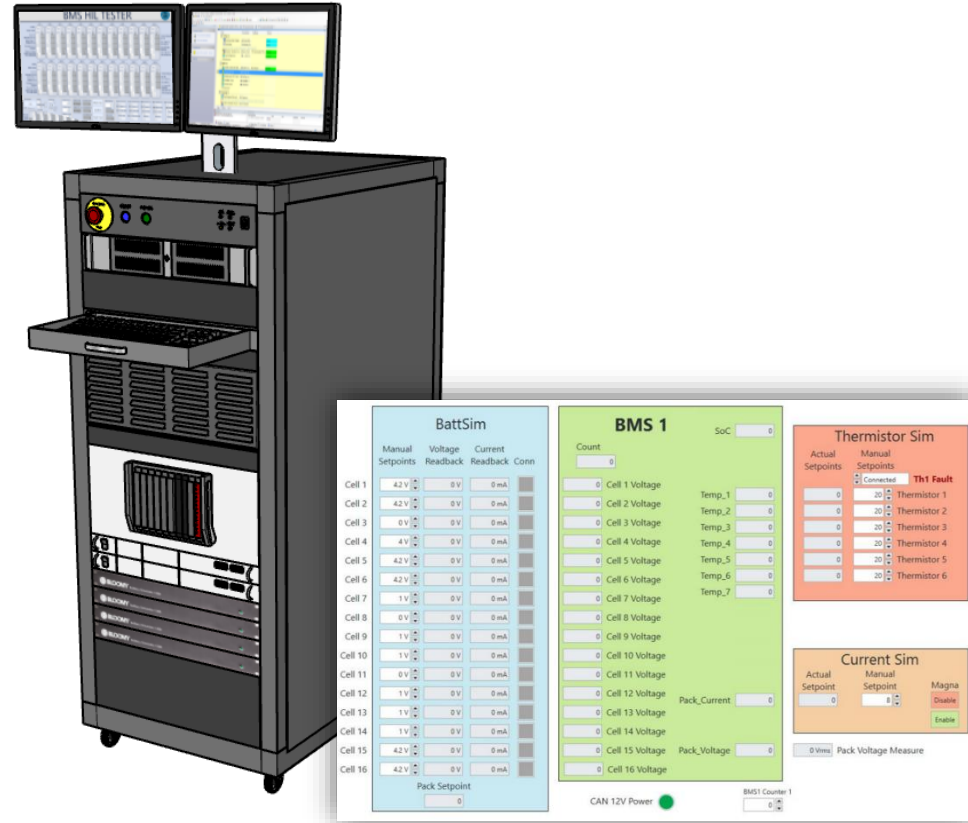
- HIL = Hardware In the Loop
- Technique for development and test of embedded control systems
 - Mathematical model of processes
 - Simulation of sensor and control signals
 - Measurements fed back into model
- Facilitates testing electronic controls without the actual plant
- BMS HIL Simulator simulates battery and other EV subsystems
- Cell model is derived through characterization of real cells



What does a BMS HIL system consist of?

Hardware

- Computing
 - Real-time
 - Windows
- Instrumentation
 - Simulation
 - Measurements
 - Comms



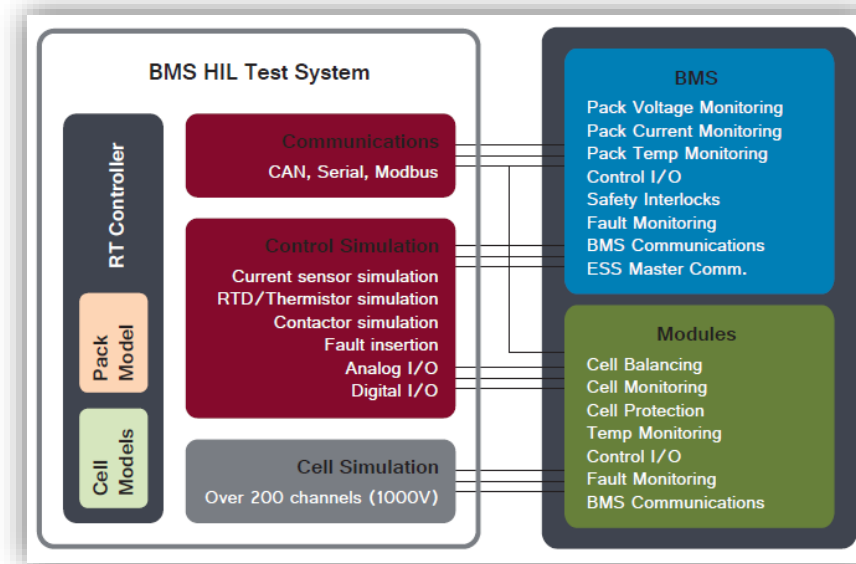
Software

- Real-Time deterministic
 - IO & comms control
 - Model execution
 - High speed data logging
- Windows
 - Configuration
 - User Interfaces
 - Test Scripting

BMS HIL Test Objectives & Benefits

Objectives

- Validate functionality
 - Hardware & Firmware
- Tune algorithms
- Ensure BMS safety
- Check response timing
- Test edge cases



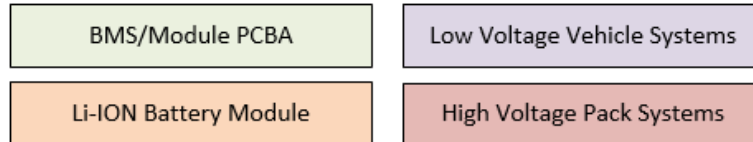
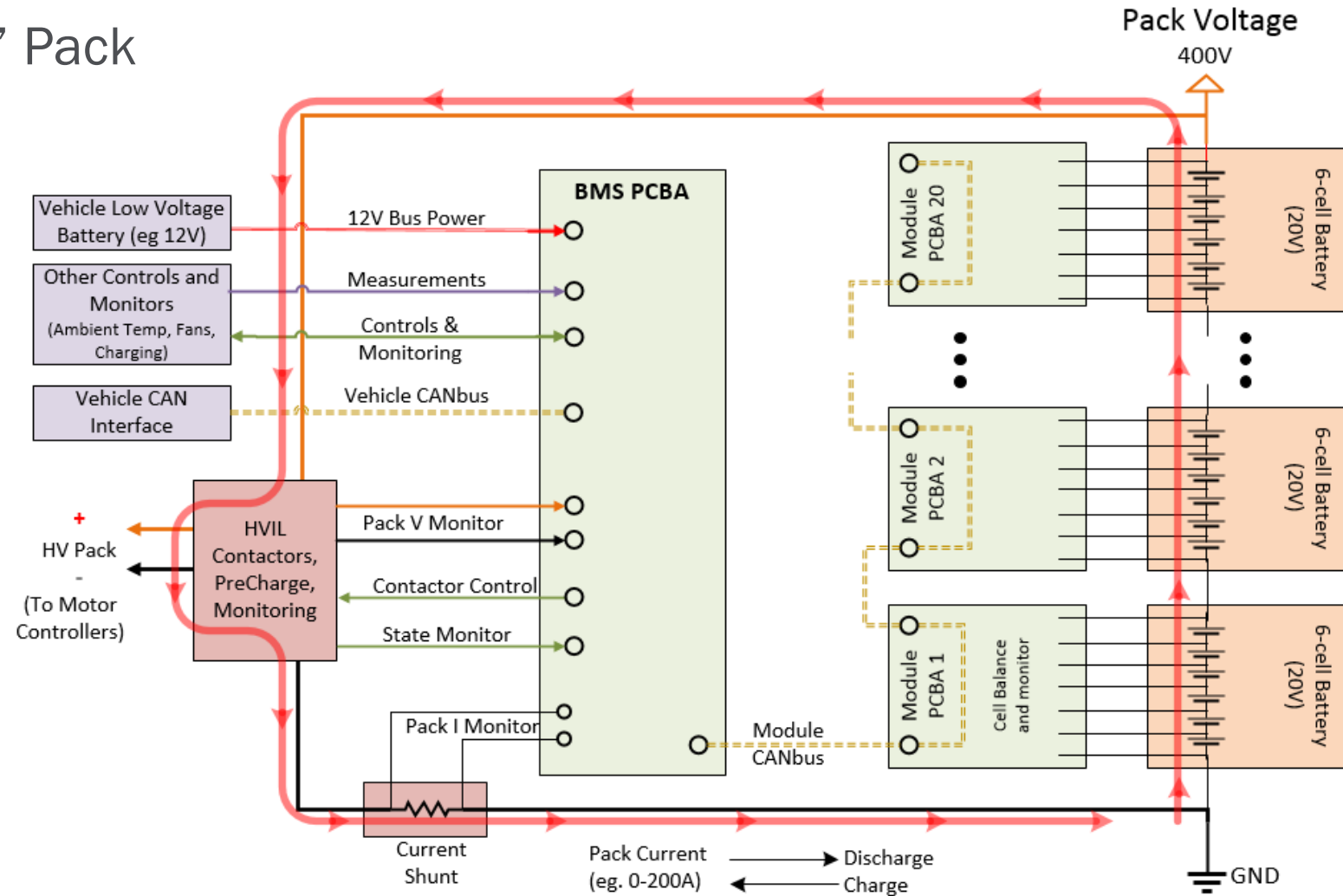
Benefits

- Operator & product safety
- Explicit repeatability
- Reduced test time
- Extended test coverage
- Faster time to market

BMS HIL METHODOLOGY

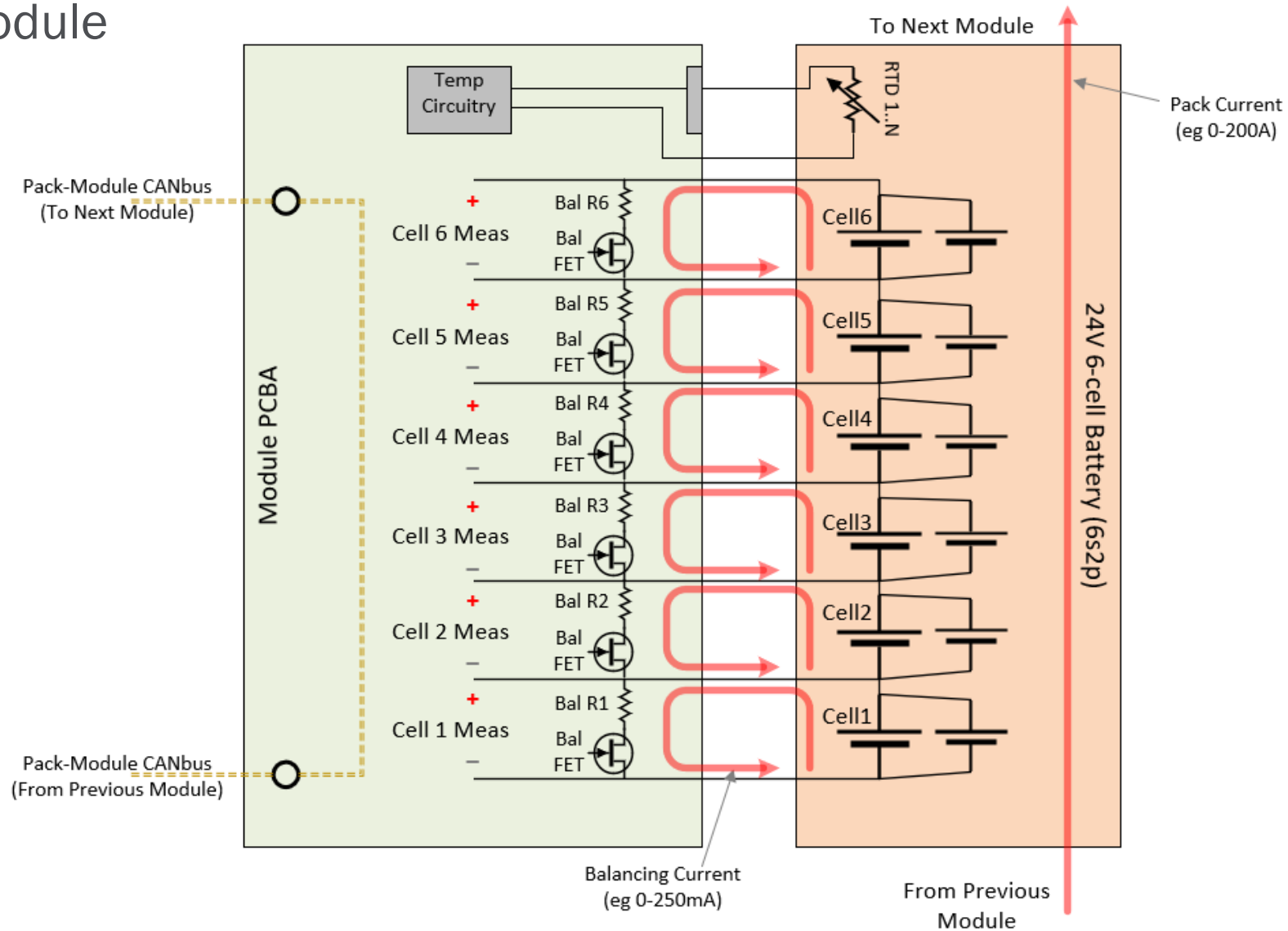
BMS and Module Interaction

“Real World” Pack



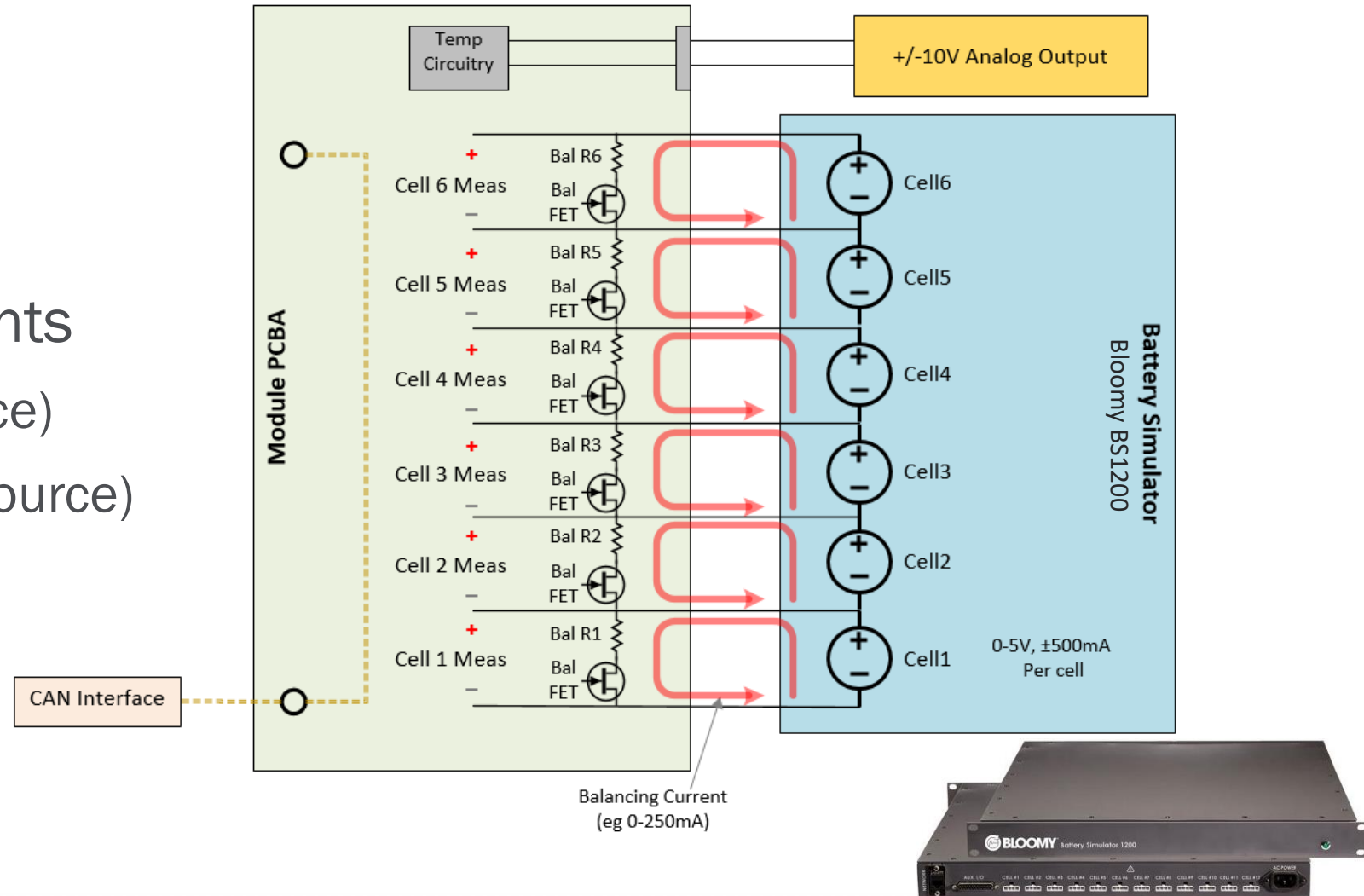
BMS and Module Interaction

“Real World” Module



HIL Testing at the Module Level

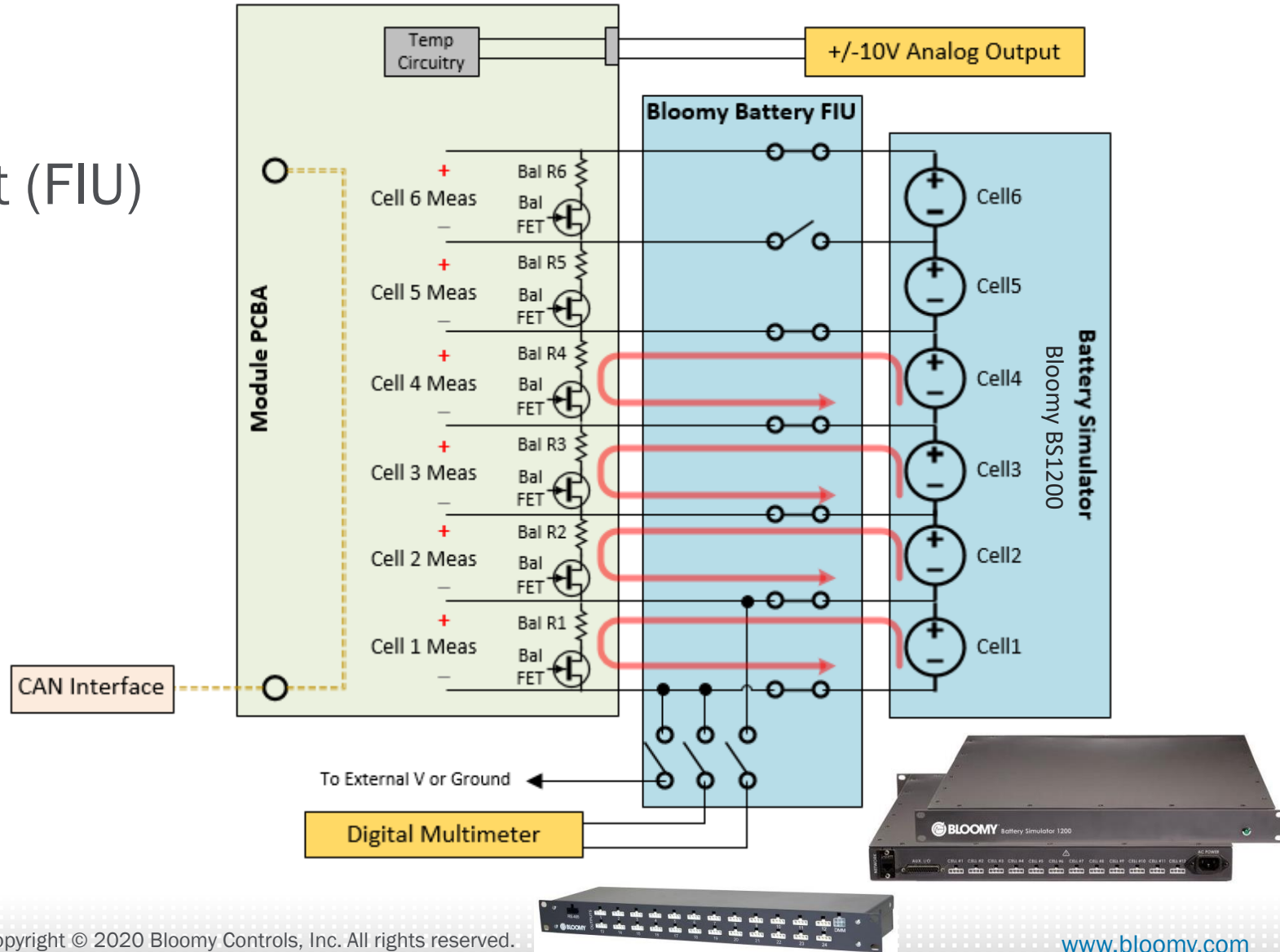
- Simulate series cells
- Output cell voltages
- Provide balancing currents
 - Passive balancing (source)
 - Active balancing (sink/source)
- Simulate thermistors



HIL Testing at the Module Level

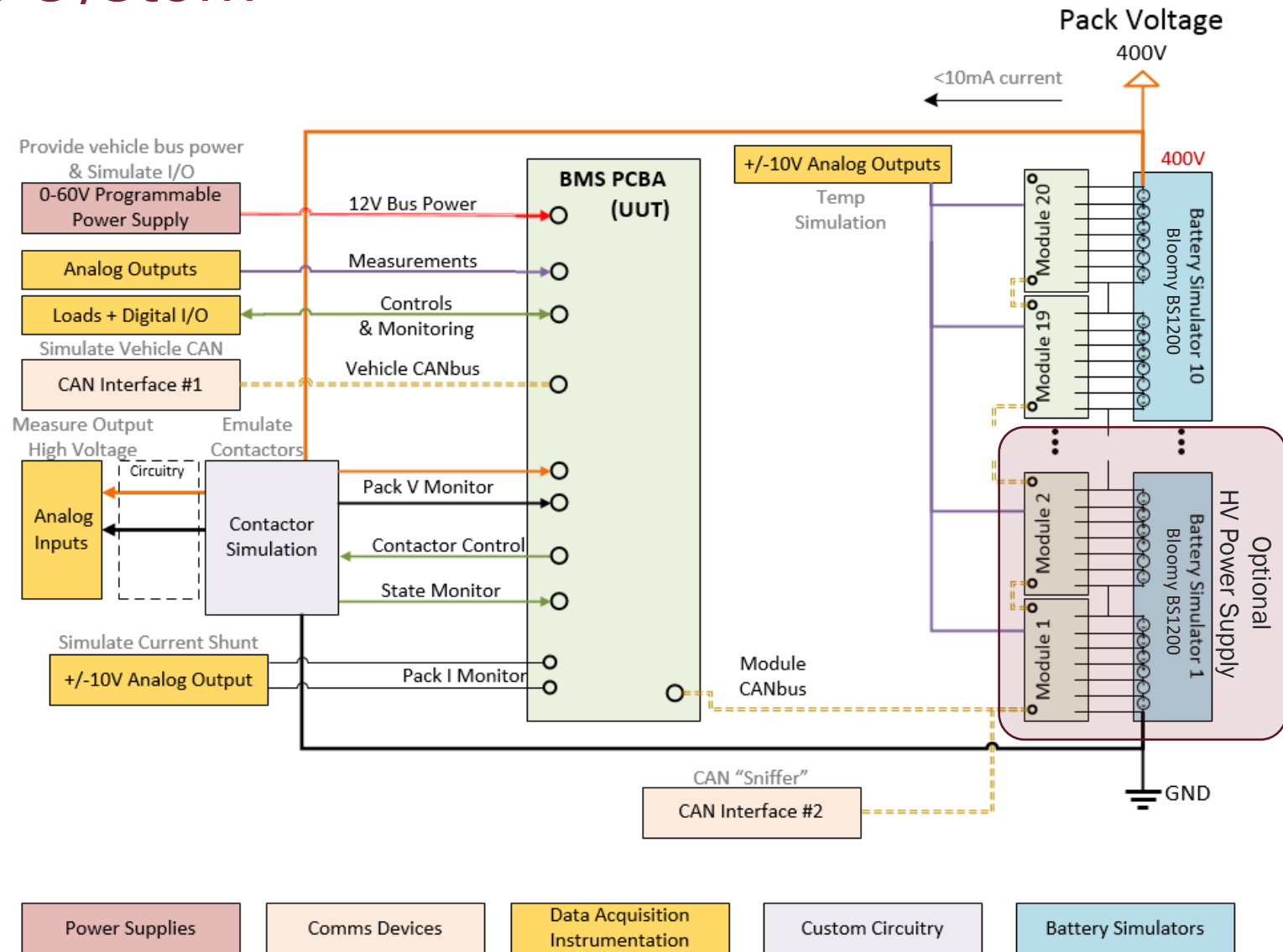
With Fault Insertion

- In-line Fault Insertion Unit (FIU)
- Open/broken wire faults
- Short to rail faults
- Connect to DMM
- Shorted cell (BS to 0V)



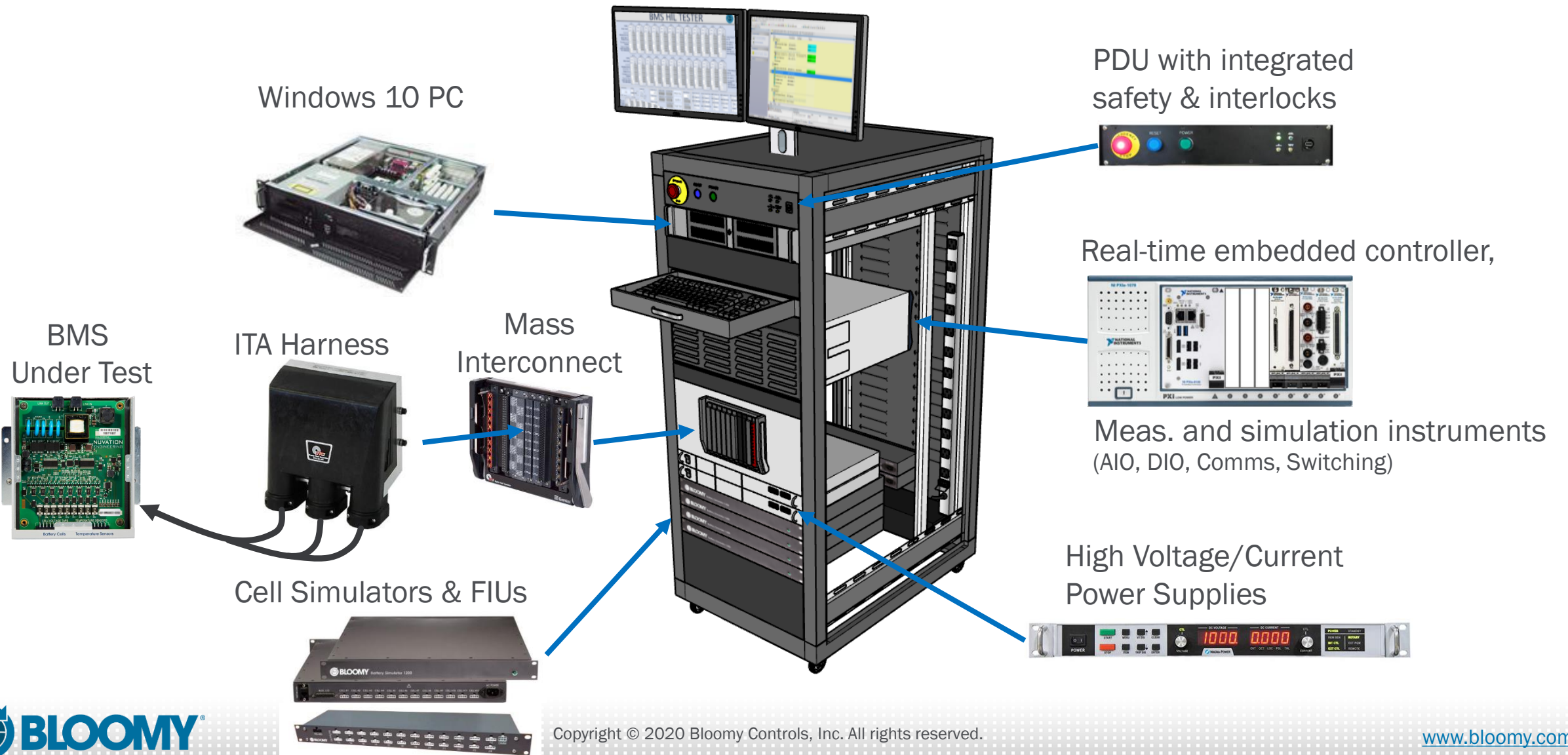
HIL Testing of the entire system

- BMS in the “real world”
- Simulate/monitor all IO
 - Cells
 - Pack V & I
 - Contactors
 - Environmental
 - Vehicle/system
 - Communications
- Low power, inherently safe

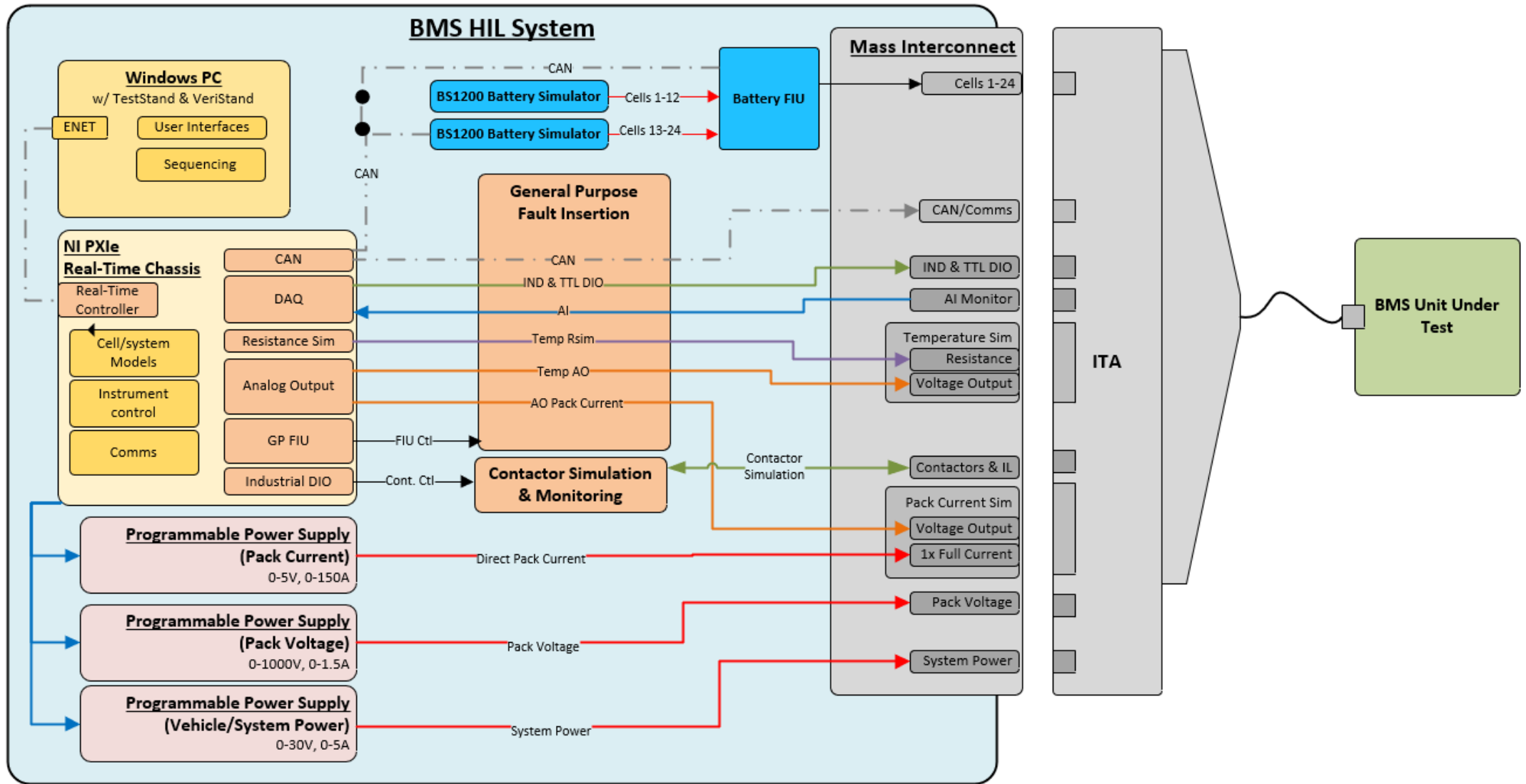


BMS HIL INSTRUMENTATION AND CONNECTIVITY

BMS HIL Instrumentation Overview



BMS HIL Hardware Block Diagram

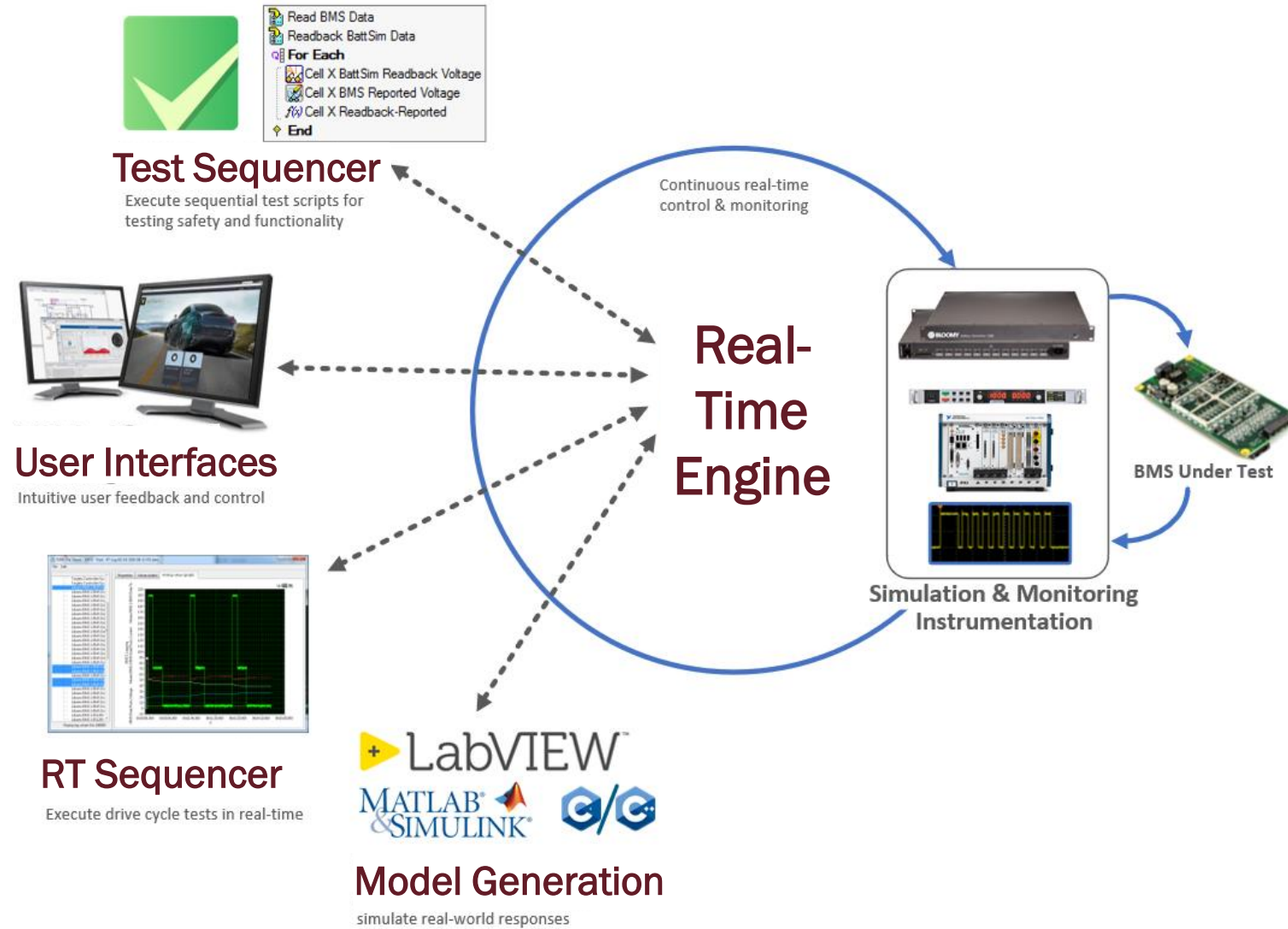


BMS HIL Signals

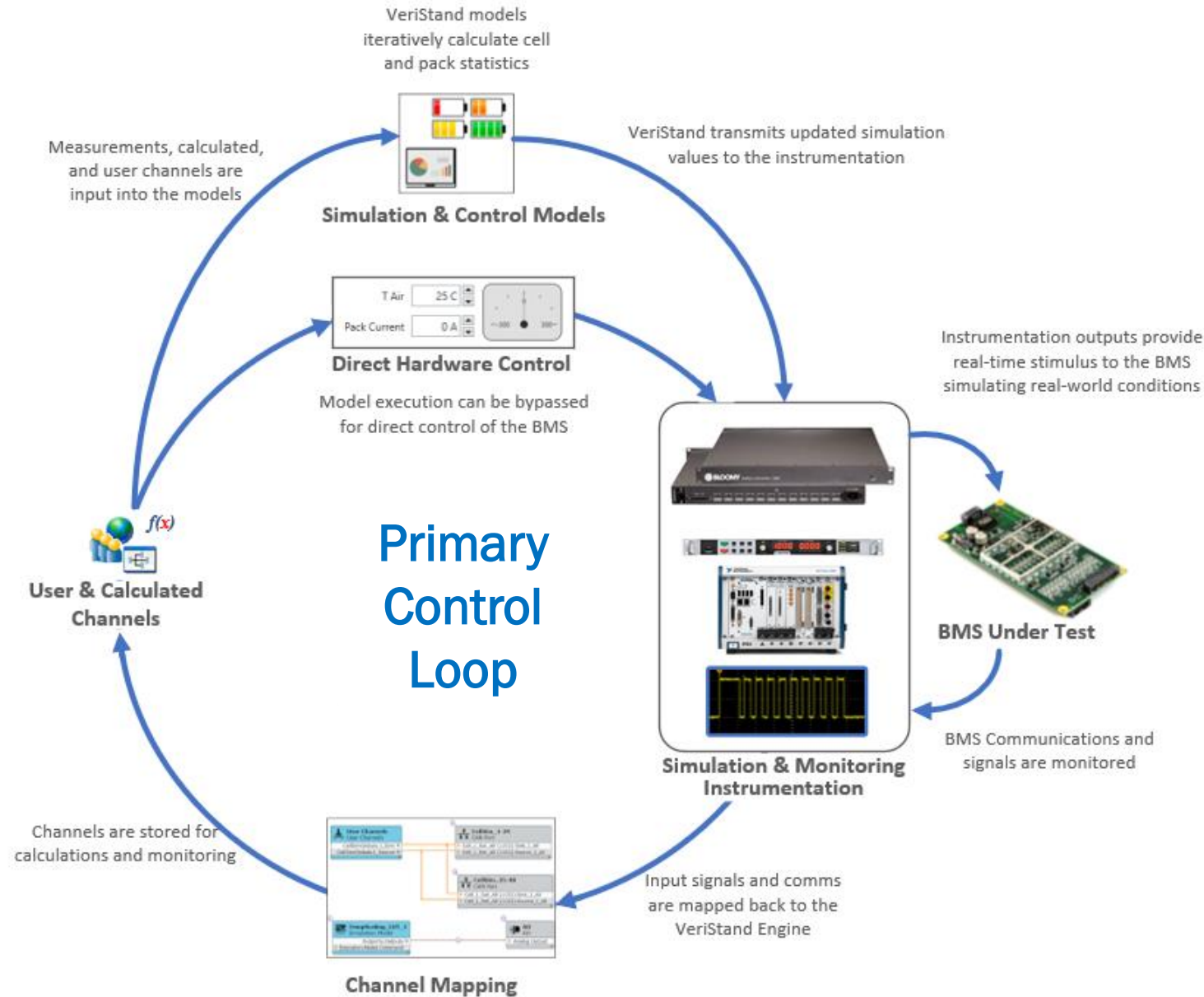
Signal Type	Qty	Description	Details
Vehicle/CAN power & interlocks	1-3	12V or 24V main power	Programmable Power Supply
Cell Simulation	12-200	Simulation of individual cells	0-5VDC Sink/Source 500mA
Temperature Simulation	8-32	Simulation of cell/pack temperatures	Analog Voltage (+/-10V) Resistance (0-65kΩ)
Pack Current Simulation	1	Simulation of full pack current (eg 400A)	Analog Voltage (+/-10V) Programmable PS (up to 150A)
Vehicle Discrettes	2-4	Controlling ignition, interlock, enable, CAN power	Fixed power supply with industrial digital output control
Comms	1-4	BMS-Vehicle, Module, etc	CAN, LIN, Serial, IsoSPI
Contactors	2-4	Simulation of contactors to verify BMS control	Fixed resistance with industrial digital inputs for monitoring
Insulation Resistance sim	1-2	Simulating HV resistance to test BMS monitor circuits	Fixed resistors with high voltage switching
Misc BMS IO	4-12	Misc analog/digital inputs and outputs to/from BMS	Analog/digital inputs and outputs for simulation/monitoring

BMS HIL SOFTWARE

BMS HIL Software Components

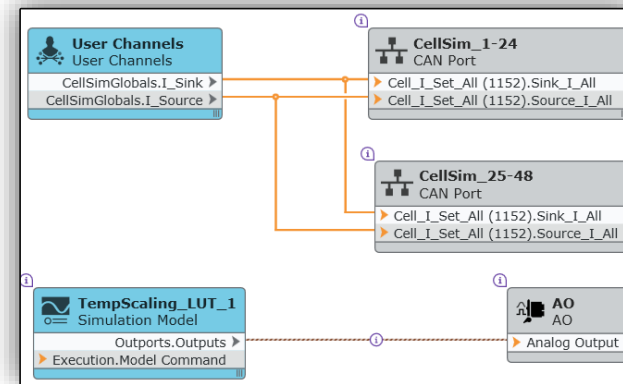
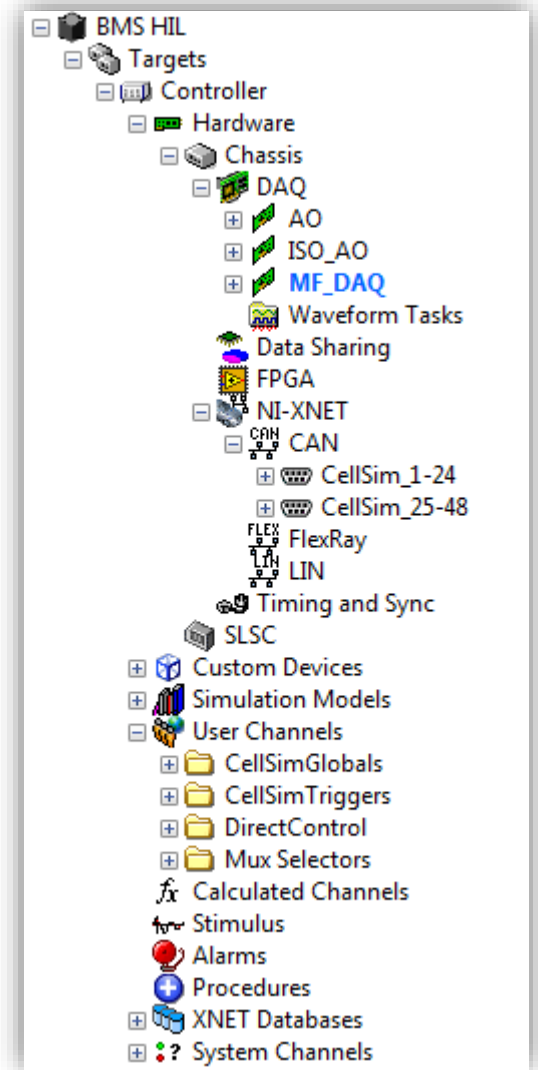
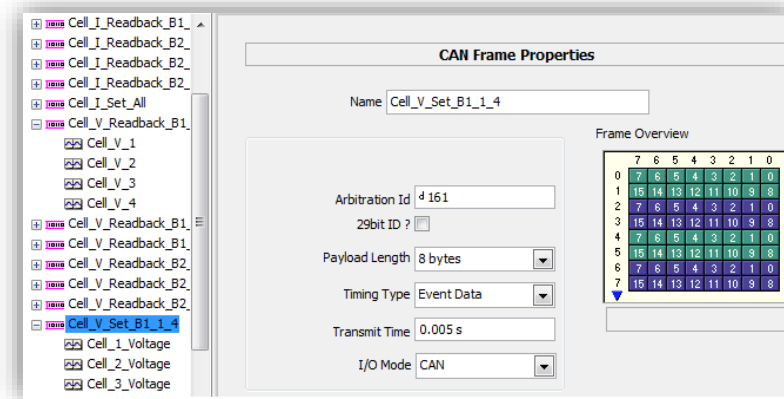


BMS HIL Software – Real-Time Engine Execution



BMS HIL Software – Real-Time Engine Configuration

- Instrumentation Interfaces
 - DAQ, BattSim, FIU
 - Switching, Power Supplies
- User and calculated channels
- BMS Communications
 - Serial, CAN Databases
- Models
 - Cell, vehicle, contactors
- Procedures
 - Startup, alarm, safety
- Channel mapping



User & System Channels

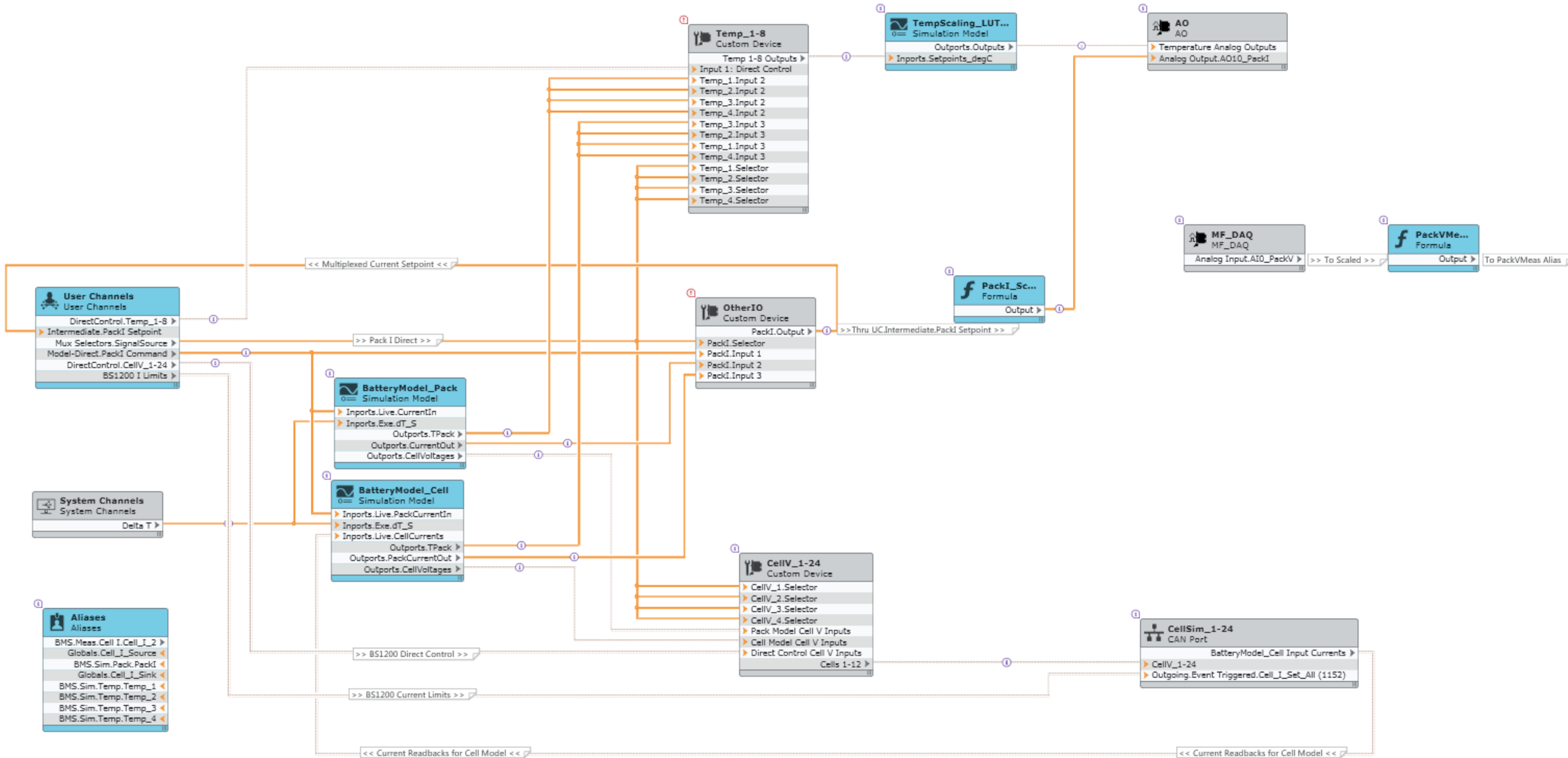
Battery Models

Channel Multiplexing

Output Scaling

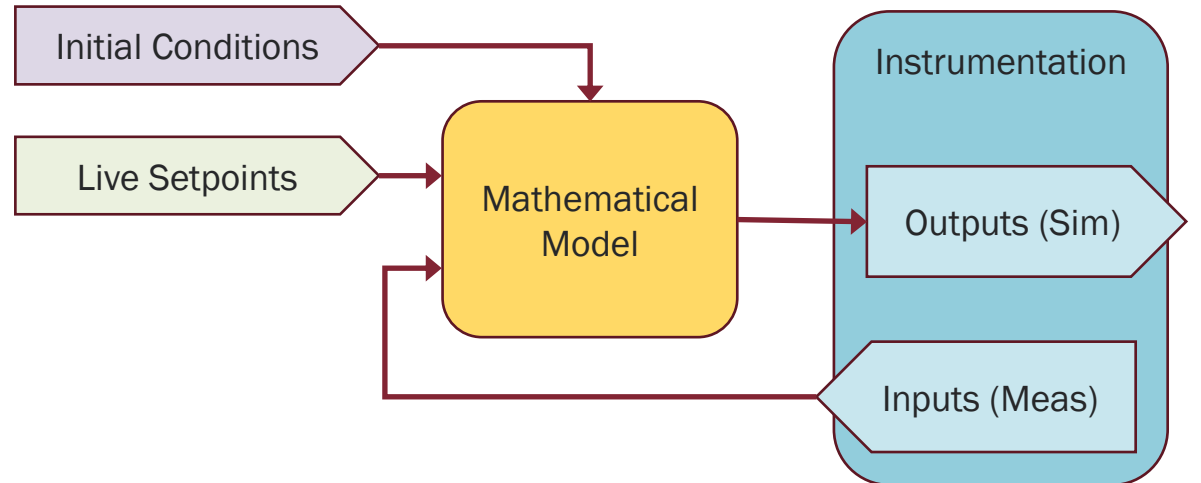
Instrumentation IO

Input Scaling

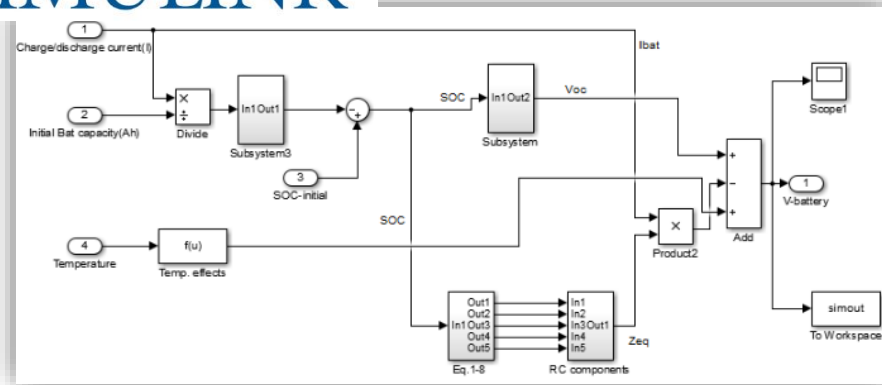


BMS HIL Software – Model Integration

- Models define behavior of outside world
 - Cells, thermal, pack
 - Contactors, vehicle/system
- Models close the loop
- Provide “real world” testing
- Mimic the response of real batteries
- Tune algorithms and response of BMS



Model Development Options



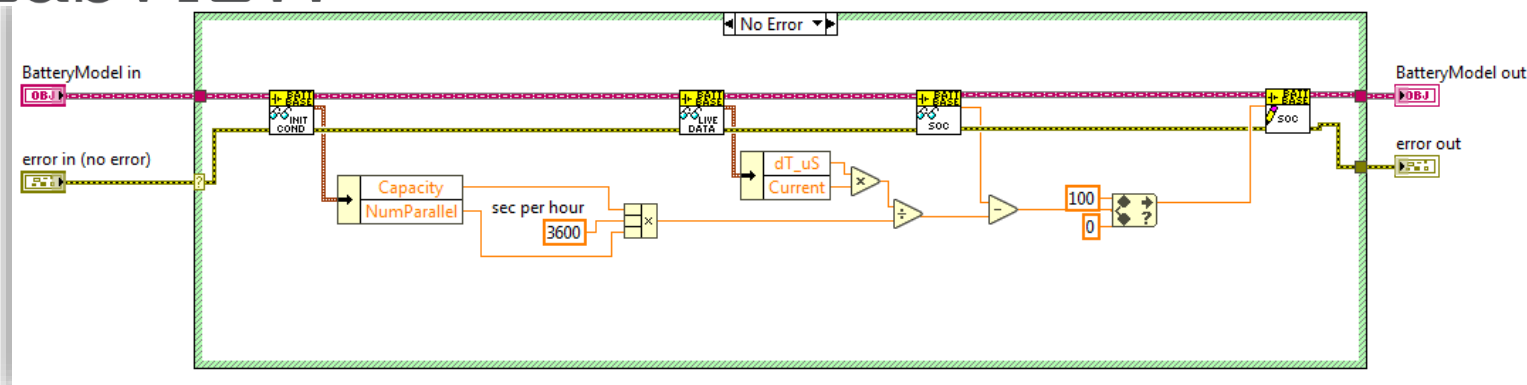
```
int32_t state_normal(double timestamp) override {
    static double current[currentLen];
    static int loop;
    switch((ModelMode)(inports->Mode)) {
        case FIXED_CURRENT:
            {
                double cellVoltage[16] = {0};
                int numCells = inports->Num_Series;
                double PackV = 0;
                double IPackFactor = 0;
                bool thresholdreached = false;

                //update SoC based on pack current
                signals->SoC = (((signals->SoC/100.0)*inports->Capacity) - (
                    inports->Current*1e-5))/inports->Capacity)*100.0;

                //Limit SoC to 0-100
                signals->SoC = (signals->SoC < 100) ? signals->SoC : 100;
                signals->SoC = (signals->SoC > 0) ? signals->SoC : 0;

                //update cell voltages based on SoC
                for (int i = 0; i < 16; i++) {
                    if (i <= numCells-1) {
                        cellVoltage[i] = -2e-7*std::pow(signals->SoC,4)+5e-5*std
                            ::pow(signals->SoC,3)-0.004115*std::pow(signals->SoC,2)+
                            0.1335*signals->SoC+2.5;
                    }
                }
            }
    }
}
```

Sheikh, Muhammad & Baglee, David & Knowles, Michael & Elmarakbi, Ahmed & Al-Hariri, Mohammad. (2015). A Novel Approach for Predicting Thermal Runaway in Electric Vehicle Batteries When Involved in a Collision. 10.1115/IMECE2015-51781.

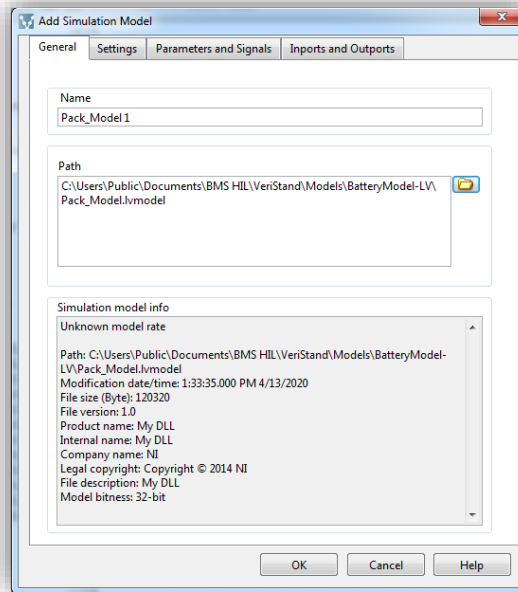


Model Integration into HIL Environment

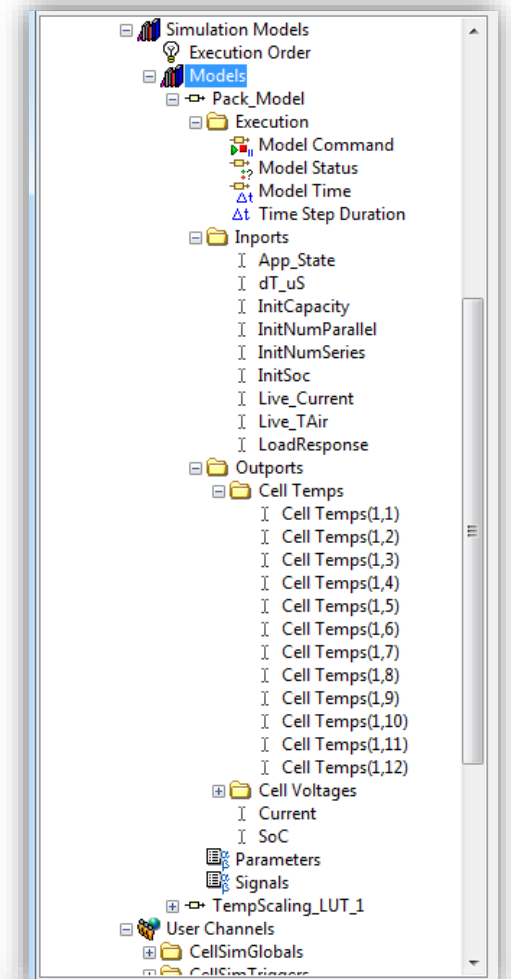
Develop & compile model



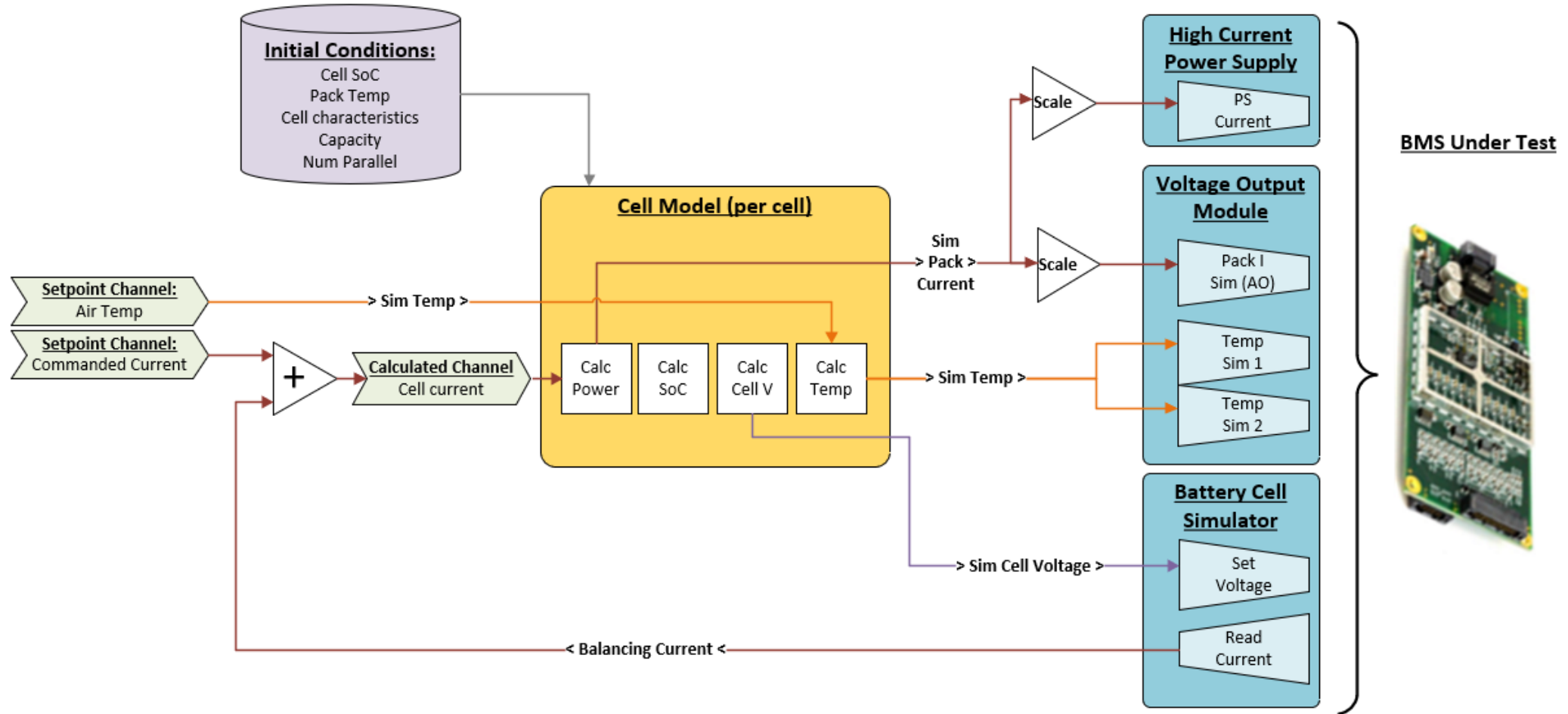
Load into environment



Map channels

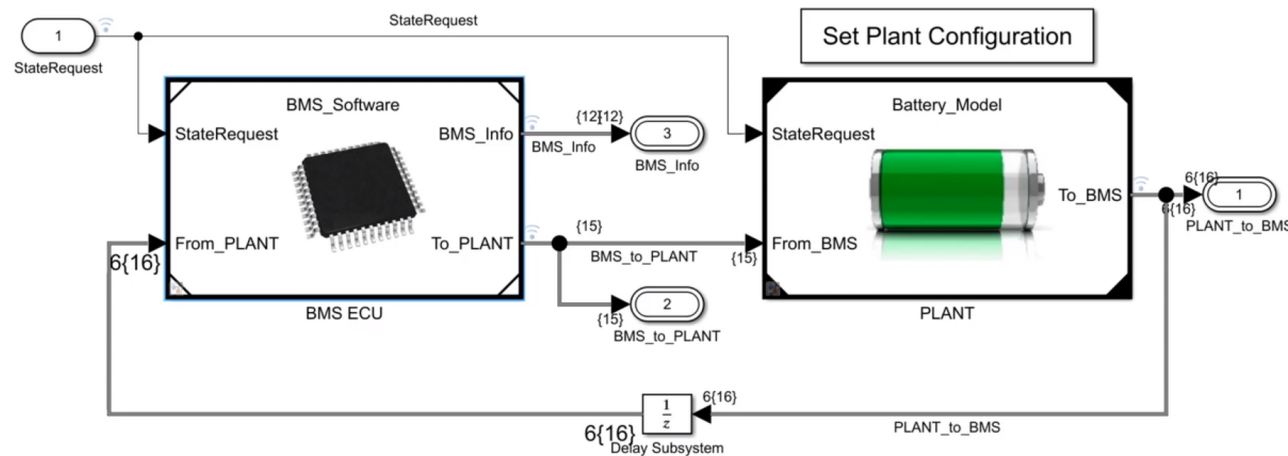


BMS HIL Software – Cell Model Example



Advantages of Model-Based BMS Testing

- Evaluate different cell chemistries/suppliers
- Evaluate different BMS architectures and tune algorithms
- Incorporate interaction with other vehicle systems
- Repeatably test real-world and edge condition scenarios
 - Impossible, impractical, too dangerous, or too expensive



<https://www.mathworks.com/solutions/power-electronics-control/battery-models.html>

Copyright © 2020 Bloomy Controls, Inc. All rights reserved.

BMS HIL USER INTERFACES

BMS HIL Software – Engine User Interfaces

Engine User Interfaces

- Windows-side client
- Socket into RT execution
- Peek/poke system channels
- Intuitive view into system state
- Direct hardware control
- BMS interface control
- Model-based control

The collage displays several key components of the BMS HIL software interface:

- BS1200 #1 / BS1200 #1:** A control panel for individual cells (Cell 1 to Cell 5) showing voltage and current readbacks with manual setpoint controls.
- FIU#1:** A panel for channel monitoring, showing 'All Channels' status (Normal) and an 'Update All' button.
- BattSim:** A detailed view of battery simulation parameters for seven cells, including manual setpoints, voltage readbacks, and current readbacks.
- BMS 1:** A panel for Battery Management System status, showing 'Count' (0) and 'SoC' (0), along with individual cell voltage and temperature (Temp_1 to Temp_7) readbacks.
- Thermistor Sim:** A panel for simulating thermistor data, showing 'Actual Setpoints' and 'Manual Setpoints' for six thermistors, with a 'Th1 Fault' indicator.
- Main Control Panel:** A large central panel for 'Model Execution Settings' and 'Live Control Parameters'. It includes:
 - Model Execution Settings:** Output Data Source (Cell Model), App State (Run), and a Run button.
 - Live Control Parameters:** T Air (25 C), Pack Current (0 A).
 - Generic Initial Conditions:** dt (2500 uS), BMS Num (2), Parallel Cells (1), NumCells (16).
 - System Status:** HP Count (38), Model Count (12433), Successful File Load (0).
- Cell Grid (Cells 1-16):** A grid of individual cell control panels, each showing 'Model Command' (Start), 'Initial Conditions' (24.90 Ah, 25 C, 92.00%), 'SoC / T Pack' (92%, 25 C), and 'Voltage / Current' (e.g., 4.103 V, 0.013 A).
- BMS Readback Data:** A table providing detailed readback data for 16 cells, including cell voltages, temperatures, and pack-level metrics like Balancing Pack, Pack Current, SoC, and Pack Voltage.
- Additional Panels:** Partial views of 'Fault' status panels and 'Magna' simulation controls.

BMS HIL Software – Automated Test Operator Interfaces

Automated Test User Interface

- Windows-side client
- Intuitive automated test selection
- Simplifies test execution
- Trace execution
- Advanced debugging
 - Single stepping, looping
 - Pause, terminate

The screenshot displays the 'Universal Test System' v2.2.5.7 interface. The main window is titled 'TestStand User Interface - Test UUTs -- Test Socket Entry Point - Multi-BMS Launcher.seq [5]'. It features a menu bar (File, Edit, Execute, Debug, Configure, Tools, Setup, Help) and a toolbar with 'Stop', 'Resume', and 'Pause' buttons. The 'Serial Number' field is set to 'test 1'. The 'Trace Execution' table shows the following steps:

Step	Description	Settings	Status
Setup (0)			
Main (14)			
Disable BMS CA...	Call __BMS CAN - Enable-Disa...	Disable Tracing	Passed
Set Interlock Dea...	Call __Set Interlock Enable Sta...	Disable Tracing	Passed
Set Manual Control	Call __Set Output Manual-Mod...	Disable Tracing	Passed
Wait	TimeInterval(3)		Done
Battery Simulation...			Done
Set BMS 2 All Cel...	Call __Set All Cell Voltages to S...	Pre Expression, Di	Passed
Set BMS 2 All Th...	Call __Set All Thermistor Temps...	Pre Expression, Di	Passed
Set BMS 2 Pack ...	Call __Set BMS Pack Current t...	Pre Expression, Di	Passed
Wait for setting	TimeInterval(2)		Passed
CAN Setup			
Enable BMS CAN...	Call __BMS CAN - Enable-Disa...	Disable Tracing	
Set Interlock Ena...	Call __Set Interlock Enable Sta...	Disable Tracing	
<End Group>			
Cleanup (0)			

The 'UUT Information' panels on the right show the following details:

- Main (Running):** UUT Part Number: MainThread, Test Sequence: [empty], Serial Number: MainThread. Buttons: Stop All, Terminate All, Abort All, Exit Testing.
- Socket 1 (Waiting):** UUT Part Number: Embedded 24-Cell eBMS, Test Sequence: [empty], Serial Number: [empty]. Buttons: OK, Stop.
- Socket 2 (Waiting):** UUT Part Number: Distributed 48-Cell BMS, Test Sequence: [empty], Serial Number: [empty]. Buttons: OK, Stop.

A dropdown menu is open over the 'Socket 2' panel, listing the following test scenarios:

- BMS HIL Test Scenarios.seq
- Full UUT Test.seq
- BMS HIL ATP Support Tests.seq
- Model Test - Pack.seq
- Model Test - Cell.seq
- EEPROM Tests.seq



BMS HIL TEST SCRIPTING AND DATA LOGGING

BMS HIL Software – Automated Sequencing

- Python, NI TestStand, etc
- Test script development & execution
 - Sequential scripts, flow control
 - Pass/fail determination
 - Results collection
- Interact with real-time system
 - Get/Set channel values
 - Control model setpoints
- Automate common tasks/tests
- Tests are non real-time

The image displays the NI TestStand software interface. The main window shows a test sequence titled "BMS HIL Test Scenarios.seq" with a "Steps: CAN 12V Test" view. The steps include: "Disable BMS CAN Comms", "Wait", "Set CAN 12V OFF", "Look for CAN comms to stop", "For" loop (containing "Query BMS Data", "Wait", "Query BMS Data 2", "If", "Goto", "End"), "Done Waiting Disabled", "Verify CAN 12V Off Disables...", "Set CAN 12V ON", "Wait", "Enable BMS CAN Comms", "Wait", and "Look for CAN comms to rest...". The "Settings" column shows various configurations like "Disable Tracing", "Result Recording: Disabled", and "Post Action".

Overlaid on the TestStand interface is a Python code editor window. The code defines a class "Demo" with attributes, methods, and a function:

```
##Start##
doubleAttribute = 0.0
intAttribute = 0
boolAttribute = False
stringAttribute = ''
tupleAttribute = ()
listAttribute = [0,0,0,0,0]

def Definition_1 (X, Y):
    print("Test 1")
    return X

def Definition_2 (X, Y):
    print("Test 2")
    return Y

class Demo(object):
    Test_1 = 0
    Test_2 = 1

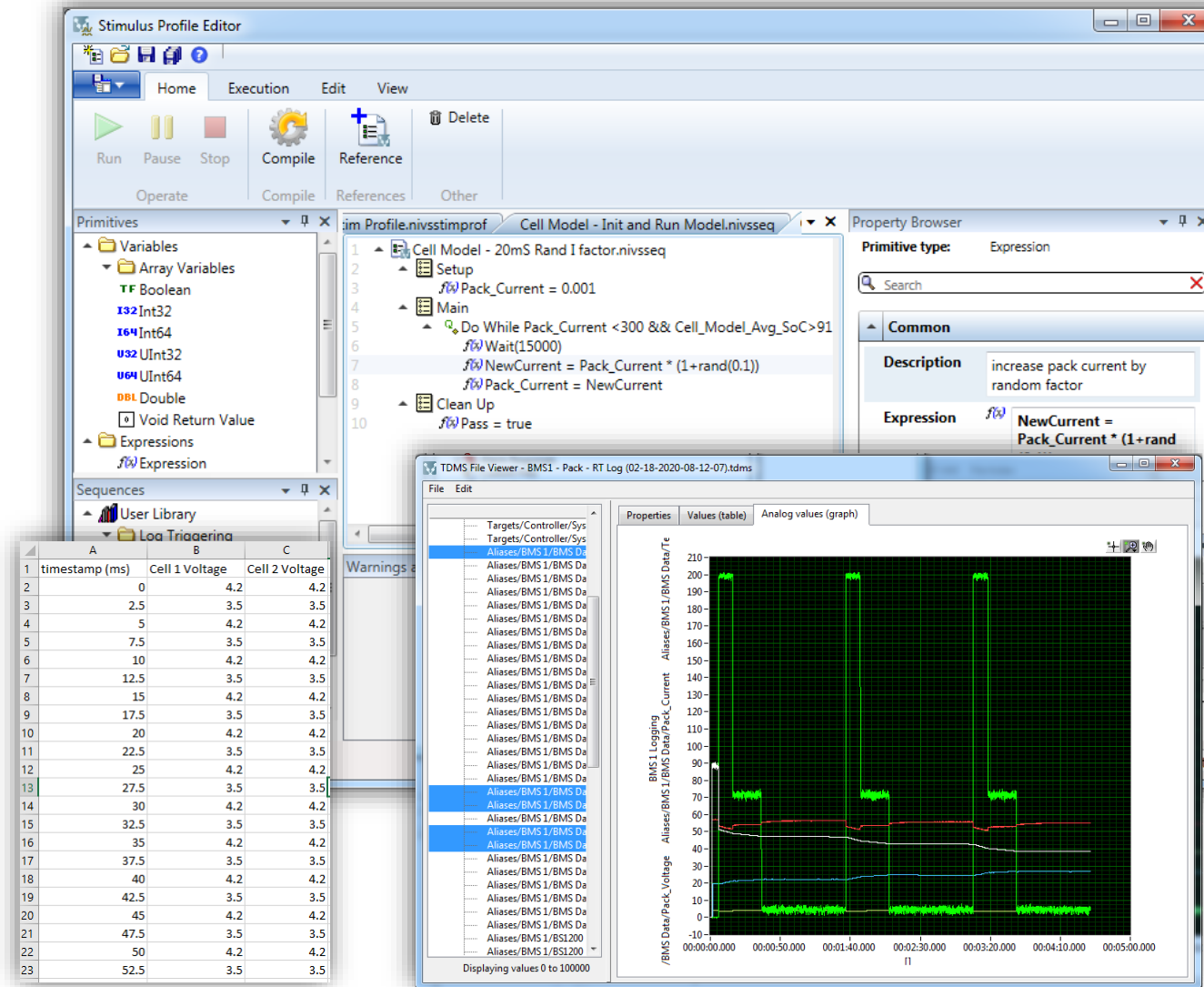
    def __init__(self, first, last, population):
        self.first = first
        self.last = last
        self.population = 3
        self.population += 1
        self.population2 += 1

    def Demo_function(self, example):
        example.Test_1 = 0
        Test_2 = 1

Test_3 = 1
```


BMS HIL Software – Real-Time Sequencing

- RT engine-based
- Stimulus profiles
 - High speed drive cycle playback
 - CSV-based channel control
- Real-time sequencing
 - Set/read channels
 - Pass/fail determination
 - Flow control
- Execute in Real-Time
- High-speed channel data logging



BMS HIL Software – Data Logging

Automated test script results

- Discrete measurements
- Pass/fail data
- XML, CSV, or Excel files
- Database storage

Step	Status	Measurement	Units	Limits			
				Nominal Value	Low Limit	High Limit	Comparison Type
Verify +15V Input Voltage	Failed	50.00221359436	VDC		14	16	GELE(>= <=)
Verify -15V Input Voltage	Failed	50.00221359436	VDC		-16	-14	GELE(>= <=)
Verify 5V Input Voltage	Failed	50.00221359436	VDC		4	6	GELE(>= <=)
Verify +15V Input Current	Passed	0.005			0.001	0.25	GELE(>= <=)
Verif						0.25	GELE(>= <=)
Verif						0.25	GELE(>= <=)

Real-time channel logging

- At PCL rate (400 – 1000Hz)
- Any real-time channel
- CSV, Excel, TDMS

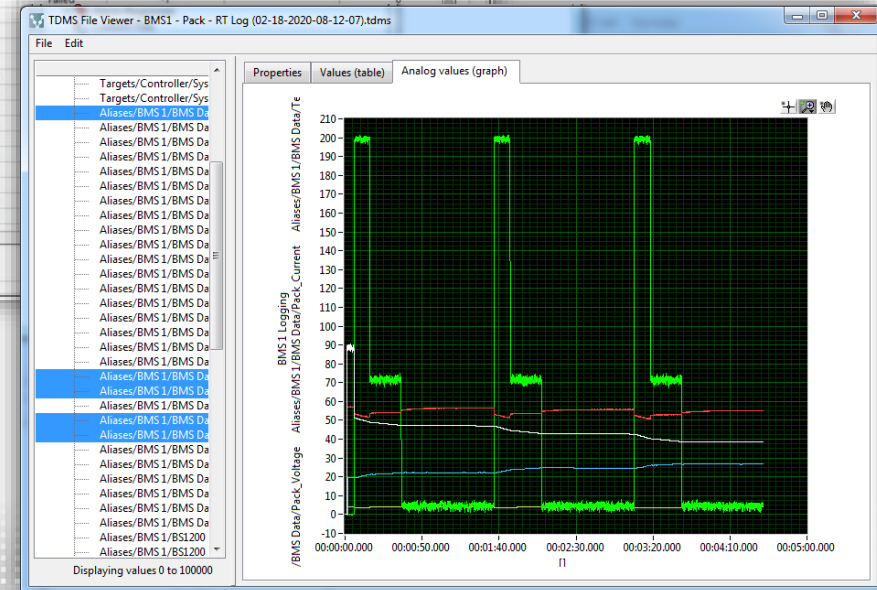
EFT Module for TestStand SQL Database Report Generator v1.1.0.21

Database Setup Results Single UUT Report SQL Queries Database Connected

TestRun ID: 1182

Part Number	Revision	Serial Number	Work Order/Lot	Unit	Unit ID	Sequence	Seq. Checksum	Limit File	Limit Checksum	TestType	Result	Timestamp	ExecutionTime (s)	Exit Code	Exit Description	Station	Fixture ID	Test Socket	Operator	Notes
16 Cell BMS		cell model		16 Cell BMS	154	Model Test - CellSeq	05ac2760226441aac99c31e104f7f406				Failed	2/17/2020 10:15:31 AM	91.0647259	Failed	Failed	BLOOMY-121521		3	administrator	

StepName	StepSubName	StepType	Result	Measurement	Units	Limit_Hi
Set Up Model Execution.Download Configuratio		SequenceCall	Passed	0		0
Set Up Model Execution.Download Configuratio		PassFailTest	Passed	1		0
Set Up Model Execution.Verify CSV Param File L		NumericLimitTest	Failed	-1		0
Set BMS to Cell Model Mode		SequenceCall				
Turn on CAN Comms		SequenceCall				
Verify Balancing On		NumericLimitTest				
Health Alert (1)		NumericLimitTest				
Cell 7 Current (1)		NumericLimitTest				
Health Alert (2)		NumericLimitTest				
Cell 7 Current (2)		NumericLimitTest				
Health Alert (3)		NumericLimitTest				
Cell 7 Current (3)		NumericLimitTest				
Health Alert (4)		NumericLimitTest				
Cell 7 Current (4)		NumericLimitTest				
Health Alert (5)		NumericLimitTest				
Cell 7 Current (5)		NumericLimitTest				



BMS HIL TEST SCENARIOS

Cell Voltage Verification and Balancing Test

- Set one cell at a time to balancing voltage
- Verify Balancing V & I from BS1200
- Verify BMS response (status & meas)

Setup (5)

- BMS Cell Balancing Test
 - Set all cells to nominal voltage
 - set one cell to balancing voltage
 - verify all cell currents & query balancing states from BMS

if TestChannel = 0, prompt for channel to test

if

- Select cell to test

End

<End Group>

Main (18)

- Set All Cell Voltages to Nominal
- Set Cell y to Balancing
- Wait for settling

Verify currents

Readback all cell V & I from simulators

For

- if this is the balancing channel, use balancing limits

if

- Verify Cell x Balancing Current
- otherwise use nominal limits

Else

- Verify Cell y Nominal Current with Cell x Balancing

End

End

Verify BMS Data Balancing States

Query Health Alerts

Verify Balancing On

<End Group>

Cleanup (3)

- Set BMS x All Voltages to Nominal
- Query Health Alerts
- Verify Balancing returns OFF

<End Group>

BattSim

	Manual Setpoints	Actual Setpoints	Voltage Readback	Current Readback	Conn
Cell 1	4.2 V	4.2 V	4.202 V	15.1 mA	
Cell 2	4.2 V	4.2 V	4.201 V	14.9 mA	
Cell 3	4.25 V	4.25 V	4.251 V	278 mA	
Cell 4	4.2 V	4.2 V	4.200 V	14.8 mA	
Cell 5	4.2 V	4.2 V	4.199 V	15.2 mA	
Cell 6	4.2 V	4.2 V	4.201 V	14.9 mA	
Cell 7	4.2 V	4.2 V	4.198 V	15.1 mA	
Cell 8	4.2 V	4.2 V	4.201 V	15.1 mA	
Cell 9	4.2 V	4.2 V	4.199 V	14.9 mA	
Cell 10	4.2 V	4.2 V	4.199 V	14.8 mA	
Cell 11	4.2 V	4.2 V	4.201 V	15.1 mA	
Cell 12	4.2 V	4.2 V	4.200 V	15.3 mA	

Pack Setpoint: 50.4

BMS

Balancing Pack

Onboard Temp: 22.4

- 4.205 Cell 1 Voltage
- 4.201 Cell 2 Voltage
- 3.994 Cell 3 Voltage
- 4.199 Cell 4 Voltage
- 4.199 Cell 5 Voltage
- 4.200 Cell 6 Voltage
- 4.202 Cell 7 Voltage
- 4.199 Cell 8 Voltage
- 4.203 Cell 9 Voltage
- 4.204 Cell 10 Voltage
- 4.204 Cell 11 Voltage
- 4.205 Cell 12 Voltage

Temp_1: 19.8
Temp_2: 20.1
Temp_3: 20.2
Temp_4: 19.9
Temp_5: 20.1
Temp_6: 20.1

Contactors: C1 (red), C2 (red), C3 (red)

SoC: 92.8

Pack_Current: 0

Pack_Voltage: 50.101

CAN 12V Power: ●

Ignition: ●

Thermistor Sim

Actual Setpoints	Manual Setpoints	
20	20	Thermistor 1
20	20	Thermistor 2
20	20	Thermistor 3
20	20	Thermistor 4
20	20	Thermistor 5
20	20	Thermistor 6

Th1 Fault

C1 Meas: ●
C2 Meas: ●
C3 Meas: ●

Current Sim

Actual Setpoint	Manual Setpoint
0	0

50.398 Vrms Pack Voltage Measure

Over/Under Voltage Protection Test

- Set one cell at a time to over/under voltage
- Verify BMS contactor outputs
- Verify BMS response (status & meas)

Setup (5)

- Overvoltage test
 - Set all cells to nominal voltage
 - set one cell to overvoltage voltage
 - verify overvoltage alarm comes from BMS

if TestChannel = 0, prompt for channel to test

?-3 If

- Select cell to test
- End

<End Group>

Main (7)

- Set BMS x All Voltages to Nominal
- Set BMS x Cell y to Overvoltage

Verify BMS Data

- Query Health Alerts
- Verify CELL_VOLTAGE_TOO_HIGH

<End Group>

Cleanup (3)

- Set BMS x All Voltages to Nominal
- Query Health Alerts
- Verify CELL_VOLTAGE_TOO_HIGH cleared

<End Group>

The screenshot displays a simulation interface for a Battery Management System (BMS). It is divided into several panels:

- BattSim (Blue Panel):** A table showing the status of 12 battery cells. Cell 5 is highlighted in red, indicating an overvoltage condition. The table includes columns for Manual Setpoints, Actual Setpoints, Voltage Readback, Current Readback, and Conn.
- BMS (Green Panel):** Shows the overall BMS status. An "Over Voltage Alarm" is active, indicated by a red box. Other parameters include Onboard Temp (22.4), SoC (92.8), Pack Current (0), and Pack Voltage (50.101).
- Thermistor Sim (Orange Panel):** Shows the status of 6 thermistors. Thermistor 1 is highlighted in red, indicating a "Th1 Fault".
- Current Sim (Yellow Panel):** Shows the status of 3 current sensors. C1 Meas is highlighted in red, indicating a fault.
- Control Panels:** Includes "CAN 12V Power" (green indicator), "Ignition" (blue indicator), and "Contactors" (C1, C2, C3) with C1 being red.

Cell	Manual Setpoints	Actual Setpoints	Voltage Readback	Current Readback	Conn
Cell 1	4.2 V	4.2 V	4.202 V	15.1 mA	
Cell 2	4.2 V	4.2 V	4.201 V	14.9 mA	
Cell 3	4.2 V	4.2 V	4.199 V	15.2 mA	
Cell 4	4.2 V	4.2 V	4.200 V	14.8 mA	
Cell 5	4.6 V	4.6 V	4.589 V	282.1 mA	
Cell 6	4.2 V	4.2 V	4.201 V	14.9 mA	
Cell 7	4.2 V	4.2 V	4.198 V	15.1 mA	
Cell 8	4.2 V	4.2 V	4.201 V	15.1 mA	
Cell 9	4.2 V	4.2 V	4.199 V	14.9 mA	
Cell 10	4.2 V	4.2 V	4.199 V	14.8 mA	
Cell 11	4.2 V	4.2 V	4.201 V	15.1 mA	
Cell 12	4.2 V	4.2 V	4.200 V	15.3 mA	

Temperature Trip Test

- Set one thermistor at a time to over/under Temp
- Verify BMS contactor outputs
- Verify BMS response (status & meas)

Setup (5)

- Temperature Trip - High
 - Set all theristors to nominal temperature
 - set one thermistor to overtemp resistance
 - verify TEMPERATURE_TOO_HIGH alarm conr

if TestChannel = 0, prompt for channel to test

?-> **If**

- Select thermistor to test

◆ **End**

<End Group>

Main (8)

- Set BMS x All Thermistors to Nominal
- Set BMS x Thermistor y to High Temp

- Verify BMS Data Temp
- Query Health Alerts
- Verify TEMPERATURE_TOO_HIGH
- Thermistor X From BMS

<End Group>

Cleanup (3)

- Set BMS x All Thermistors to Nominal
- Query Health Alerts
- Verify TEMPERATURE_TOO_HIGH Cleared

<End Group>

The screenshot displays a BMS simulation interface with four main panels:

- BattSim (Light Blue):** A table showing cell voltages and currents.

Cell	Manual Setpoints	Actual Setpoints	Voltage Readback	Current Readback	Conn
Cell 1	4.2 V	4.2 V	4.202 V	15.1 mA	
Cell 2	4.2 V	4.2 V	4.201 V	14.9 mA	
Cell 3	4.2 V	4.2 V	4.199 V	15.2 mA	
Cell 4	4.2 V	4.2 V	4.200 V	14.8 mA	
Cell 5	4.2 V	4.2 V	4.200 V	15.2 mA	
Cell 6	4.2 V	4.2 V	4.201 V	14.9 mA	
Cell 7	4.2 V	4.2 V	4.198 V	15.1 mA	
Cell 8	4.2 V	4.2 V	4.201 V	15.1 mA	
Cell 9	4.2 V	4.2 V	4.199 V	14.9 mA	
Cell 10	4.2 V	4.2 V	4.199 V	14.8 mA	
Cell 11	4.2 V	4.2 V	4.201 V	15.1 mA	
Cell 12	4.2 V	4.2 V	4.200 V	15.3 mA	
- BMS (Light Green):** Shows system status and temperature.
 - Temperature Too High (highlighted in red)
 - Onboard Temp: 22.4
 - Temp_1: 155 (highlighted in red)
 - Temp_2: 20.1
 - Temp_3: 20.2
 - Temp_4: 19.9
 - Temp_5: 20.1
 - Temp_6: 20.1
 - SoC: 92.8
 - Pack_Current: 0
 - Pack_Voltage: 50.101
- Thermistor Sim (Light Orange):** Shows thermistor settings and status.
 - Actual Setpoints: 160, 20, 20, 20, 20, 20
 - Manual Setpoints: 160, 20, 20, 20, 20, 20
 - Thermistor 1: Th1 Fault (highlighted in red)
 - Thermistor 2-6: Normal
- Current Sim (Light Orange):** Shows current measurements.
 - Actual Setpoint: 0
 - Manual Setpoint: 0
 - 50.398 Vrms Pack Voltage Measure

Additional indicators at the top right: CAN 12V Power (green dot), Ignition (blue dot), and Contactors (C1, C2, C3) with their respective measurement status (C1 Meas, C2 Meas, C3 Meas).

Pack Current Trip Test

- Set pack current over threshold
- Verify BMS contactor outputs
- Verify BMS response (status & meas)

Main (15)

- Nominal Verification
- Set Pack to Nominal Current
- Enable Magna Supply
- Log Nominal Current Setpoint
- Wait for Settling
- Read back BMS Data
- Verify BMS Reported Pack I - Nominal

OverCurrent Verification

- Set Pack to OverCurrent
- Log OverCurrent Setpoint
- Wait for Settling
- Read back BMS Data
- Verify BMS Reported Pack I - Over
- Verify EXCESSIVE PACK CURRENT Fault

<End Group>

Cleanup (5)

- Set Pack to Nominal Current
- Disable Magna Supply
- Wait for Settling
- Read back BMS Data
- Verify EXCESSIVE PACK CURRENT Resets

<End Group>

The screenshot displays a simulation interface for a Battery Management System (BMS) during a pack current trip test. The interface is divided into several panels:

- BattSim (Blue Panel):** A table showing cell data for 12 cells. Each row includes Manual Setpoints, Actual Setpoints, Voltage Readback, Current Readback, and a Conn column. The Pack Setpoint is 50.4.
- BMS (Green Panel):** Shows an "Excessive Pack Current" warning. It lists cell voltages (4.205V to 4.200V), onboard temperature (22.4), and temperatures for 6 thermistors (19.8 to 20.1). It also shows SoC (92.8) and Pack Voltage (50.101).
- Thermistor Sim (Orange Panel):** Shows manual setpoints and connected status for 6 thermistors. A "Th1 Fault" is indicated.
- Current Sim (Yellow Panel):** Shows Actual Setpoint (520) and Manual Setpoint (520) for current, with a Pack Voltage Measure of 50.398 Vrms.
- Control Elements:** Includes CAN 12V Power (green dot), Ignition (blue dot), and three contactors (C1, C2, C3) with their respective measurement indicators (C1 Meas, C2 Meas, C3 Meas).

Fault Response Tests

Main (15)

- Set All Thermistors to Nominal
- Set No Faults
- Wait
- Query BMS Data
- Verify No initial Faults
- Fault Thermistor to GND
- Wait
- Query BMS Data
- Verify GND Fault
- Fault Thermistor
- Wait
- Query BMS Data
- Verify Shorted F
- <End Group>

Main (10)

- Set BMS x All Voltages to Nominal
- Open Fault Cell
- wait for health alert to clear
- Query Health Alerts
- Verify CELL_VOLTAGE_TOO_LOW
- Verify Cell X Voltage Readback when Open
- Read Back BS1200 V&I
- Verify Cell X Current When Open
- <End Group>

Cleanup (4)

- Set No Faults
- Wait
- Query BMS Data
- Verify No Faults
- <End Group>

Cleanup (7)

- Reconnect Cell
- Query Health Alerts
- Verify CELL_VOLTAGE_TOO_LOW cleared
- Verify Cell X Voltage Readback when Fault Cleared
- Read Back BS1200 V&I
- Verify Cell X Current after Open Fault Cleared
- <End Group>

- Inject hardware faults
- Verify BMS response (status & meas)
- Verify contactor outputs
- Faults include:
 - Cell (open, short, short to rail/gnd)
 - Thermistor (open, short, short to rail/gnd)
 - Comms (bad data, short to rail/gnd)
 - Insulation resistance

Charge/Discharge Test

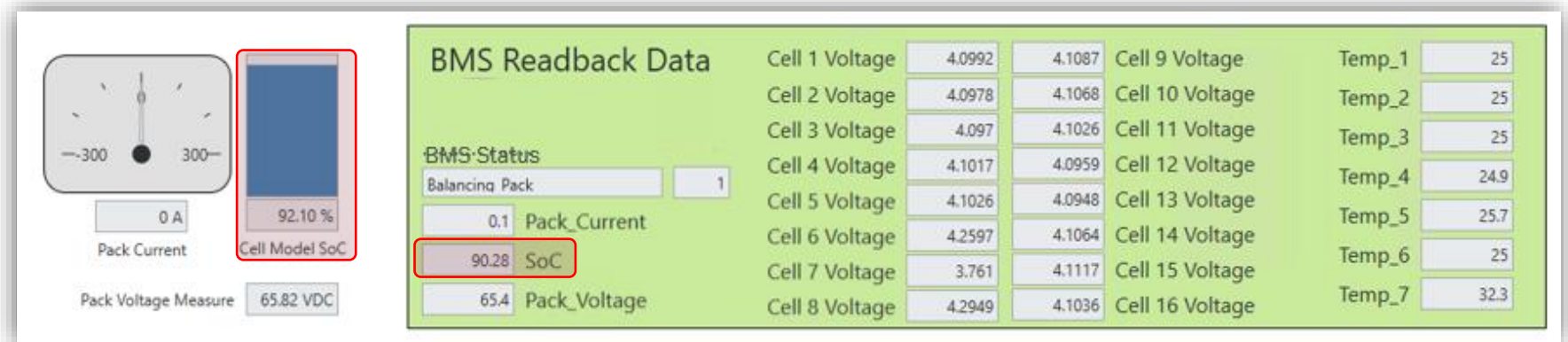
- Ideally with model in the loop
- Set pack current to 1C, 2C, 3C, etc
 - Discharge, then charge
- Monitor BMS from 95% → 5% → 95% SoC
- Compare against model/expected SoC
- Monitor BMS outputs and alarms

```
f(x) Set Start Time
Read back BMS Data
Log Initial SoC

Set Pack to Discharge Current
f(x) Init interval count
While
  Wait Log Interval
  Read back BMS Data
  Log SoC (Interval x)

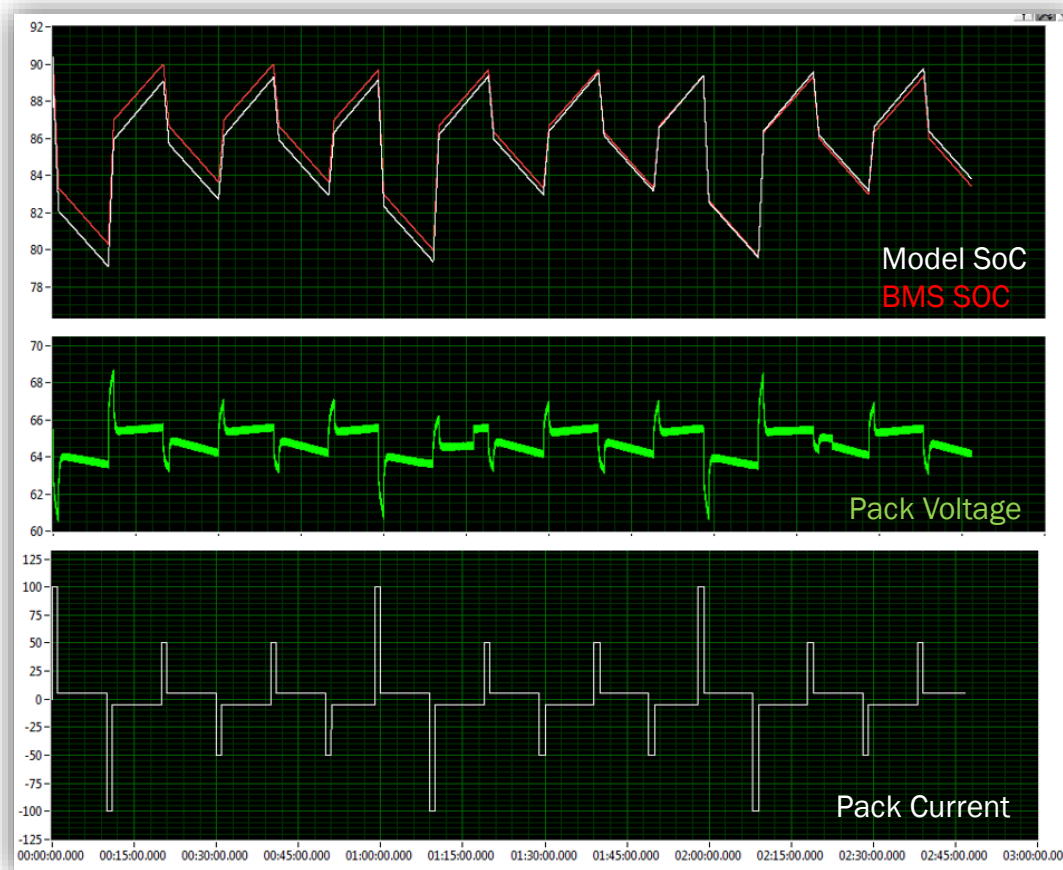
If SoC hit
  If
    Goto DoneLooping
  End
  f(x) Increment Count
End

DoneLooping
f(x) Set End Time
Shut Off Pack Current
Read back BMS Data
Final SoC
Discharge Time
```



SoC Estimation Tuning

- Control pack current through drive cycle test
- Compare model output SoC with BMS algorithm



BMS HIL CASE STUDIES

LG xEV BMS SOFTWARE VALIDATION



- Proprietary BMS software algorithms developed, tested, validated by LG
- Simulate a wide range of battery systems
 - 500V packs for EV
 - 200-300V full hybrid, PHEV
 - 48-200V micro, mild hybrids
 - 12V start-stop systems
- Able to reconfigure HIL simulator for different xEV programs without new capital appropriations



JAGUAR LAND ROVER



- Evaluated 3 cell chemistries
- Validated 4 BMS architectures
- Accelerated firmware validation from more than 1 month to less than 1 week

“Major BMS firmware releases can now be rolled out within one week with good confidence of success, whereas similar projects in the past used to take well over a month, and with substantially less confidence.”

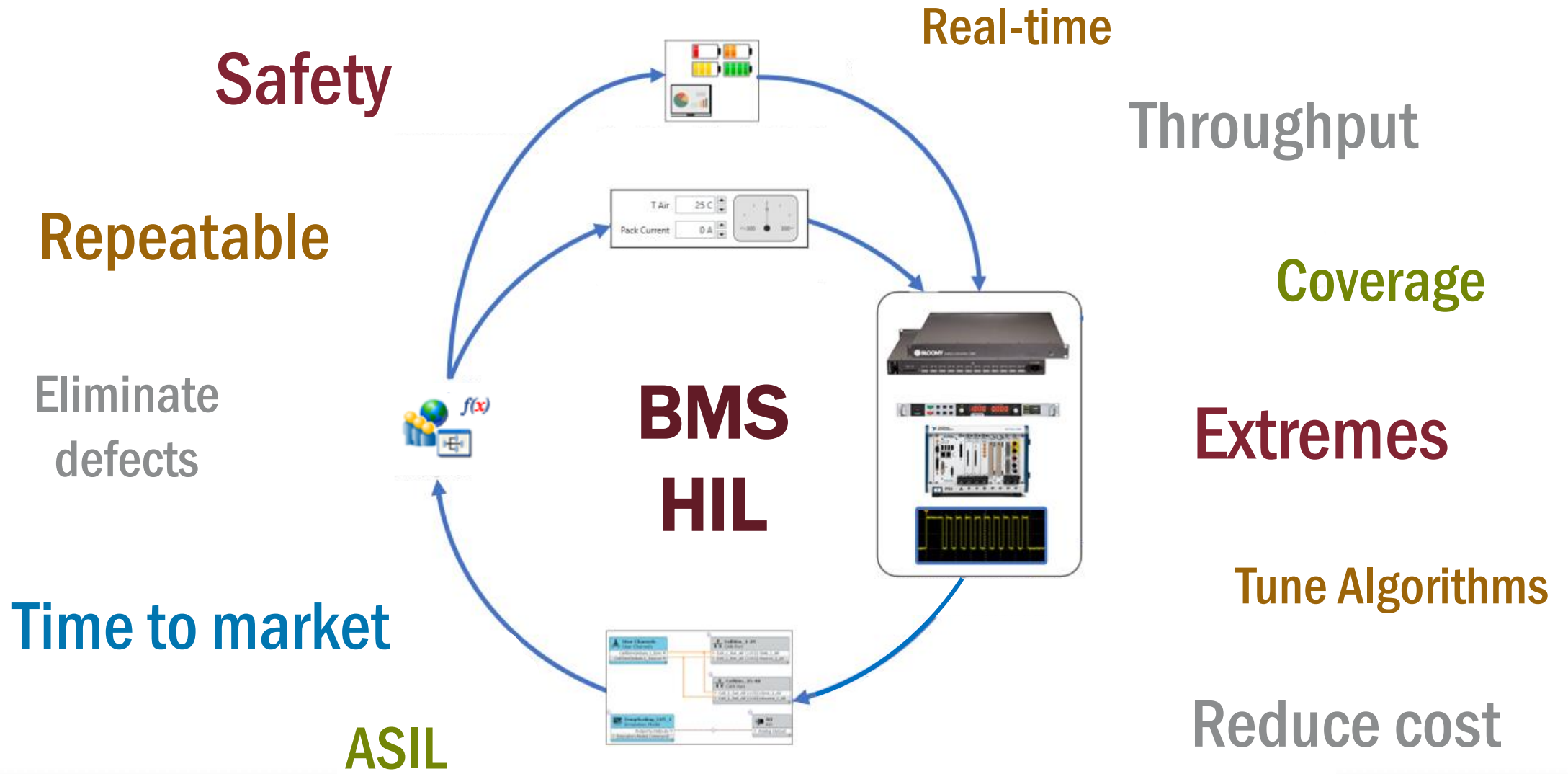
Miguel Angel Gama-Valdez, Principle Engineer, Jaguar Land Rover

- Download whitepaper

<https://www.bloomy.com/support/resources/white-paper/bms-hil-test-system-helps-jaguar-land-rover-shorten-time-market-hybrid>

SUMMARY

Battery Management System HIL Testing



MORE INFORMATION

Grant Gothing

grant.gothing@bloomy.com

Peter Blume

peter.blume@bloomy.com

www.bloomy.com

