

Accelerate EV Powertrain and Charging Subsystem Design with Virtual Prototyping

Authors

Ben Lorenz
Product Manager
Synopsys

Alan Courtoy
R&D, Sr. Architect
Synopsys

Introduction

Vehicle electrification demands significant efforts and costs across multiple areas and functions, including technological advancements, research and development, supply chain adjustments, manufacturing overhauls, regulatory compliance, and market dynamics. One of the complex EV system examples is vehicle powertrain which requires more rapid development cycles now, with timelines often compressed to 24-36 months, compared to the traditional 5-7 years for internal combustion engines. Overall, while the journey towards vehicle electrification is resource-intensive, it is essential for achieving a sustainable automotive future.

In this white paper, we will delve into the concept of electronics digital twins (eDT) and their applications in virtual prototyping for electric vehicle (EV) subsystems using Synopsys Saber. We will focus on two key applications: powertrain and charging systems.

Before we move further to learn about the specific benefits and use cases eDT for EV subsystems, it is essential to understand what the key design challenges are there in designing electric vehicle's electronics/electrical components.

Key EV Electronics' Design Challenges:

1. Reduce Losses and Improve Efficiency:
 - Thermal Management
 - Vehicle Weight Management
2. Meet Power Quality Norms:
 - Total Harmonic Distortion (THD)
 - Power Factor (PF)
 - Electromagnetic Interference (EMI)
3. Design Control:
 - Software to Test Under Numerous Hardware/System Scenarios
 - Stability (Phase/Gain) Margins
4. Ensure Robustness and Safety:
 - Initial Manufacturing Variability and Aging Degradation
 - ISO 26262 Compliance

Overall, addressing these challenges requires a comprehensive approach that integrates

advanced engineering, rigorous testing, and adherence to regulatory standards, all while balancing cost and performance considerations. Synopsys Saber can help resolve these issues by providing a robust design flow for automotive systems that enables comprehensive simulation and validation under various scenarios.

Saber Application in EV Powertrain System

There are various EV powertrain subsystems for which Saber can help accelerate the development cycle and reduce testing efforts enabling faster time to market. Some of the applications areas include

Power Net: Analysis and optimization of the vehicle's electrical power network.

HVAC (Heating, Ventilation, Air Conditioning): Simulation and validation of HVAC systems for efficient thermal management.

In-Vehicle Network: Physical layer analysis and optimization of in-vehicle communication networks.

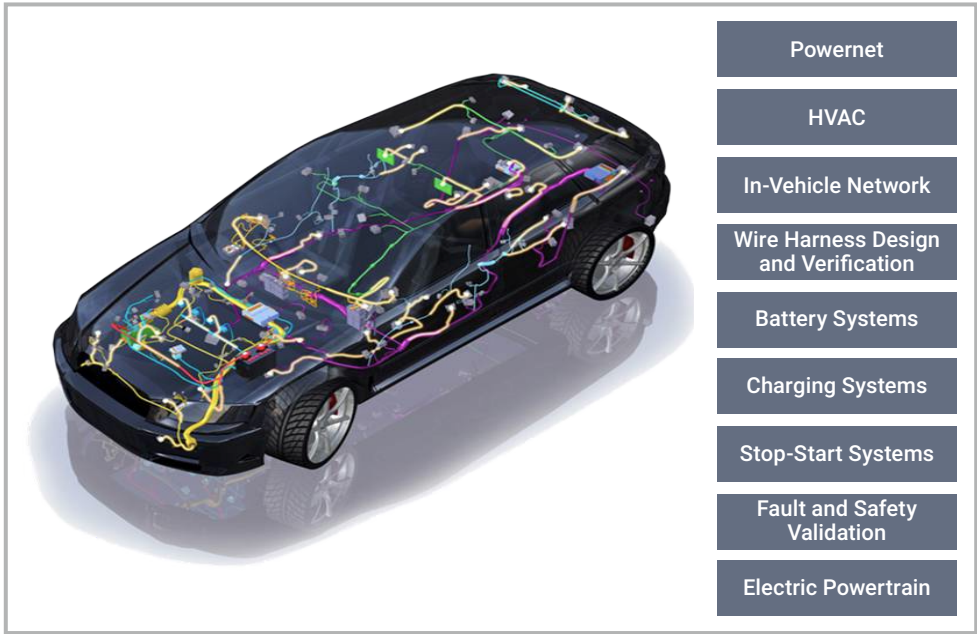


Figure 1: EV Powertrain Subsystems

Wire Harness Design and Verification: SaberES Designer tool for designing and verifying wire harnesses, which are one of the heaviest components in a vehicle.

Battery Systems: Comprehensive analysis and optimization of battery systems, including battery management.

Charging Systems: Simulation and validation of charging systems, both onboard and offboard, ensuring compliance with norms and requirements for grid power quality and efficiency.

Start-Stop Systems: Analysis and optimization of start-stop systems for fuel efficiency and reduced emissions.

Electric Powertrain: Simulation and validation of electric powertrain systems, including motor drives and power electronics.

Virtual Prototyping for Powertrain Systems

Before we walk through some powertrain applications for Saber, it is important to realize that specific problems are best analyzed by models of appropriate abstraction levels or time scales. For example, it would not be practical to simulate hours of driving cycles with semiconductor models requiring nanosecond time steps. Such detailed models are appropriate over a few inverter switching cycles, but not over extended periods.

Virtual Prototyping of Electric Powertrain using Synopsys Saber

Saber is a solution of high-precision virtual prototyping development tools for multi-domain mechatronics and power electronics design. Saber offers abstraction levels according to what is necessary for the required accuracy. To dive into the EV powertrain, we have explained four use cases: drive cycle simulation, electric motor phase current and motor angle analysis, PWM switching in inverters, and semiconductor analysis.

Use Case 1: Driving Cycle Simulation

In this use case, driving cycles are simulated in practical computation times. For example, a sequence of repeated new European driving cycles up to battery depletion (seven hours of actual driving) can be simulated in just one minute. This level of abstraction is practical to analyze factors affecting the driving range, such as the instantaneous efficiency of the motor or inverter. One of the examples at this use case could be optimizing the ratio of the gearbox or adding an extra gear to extend the driving range.

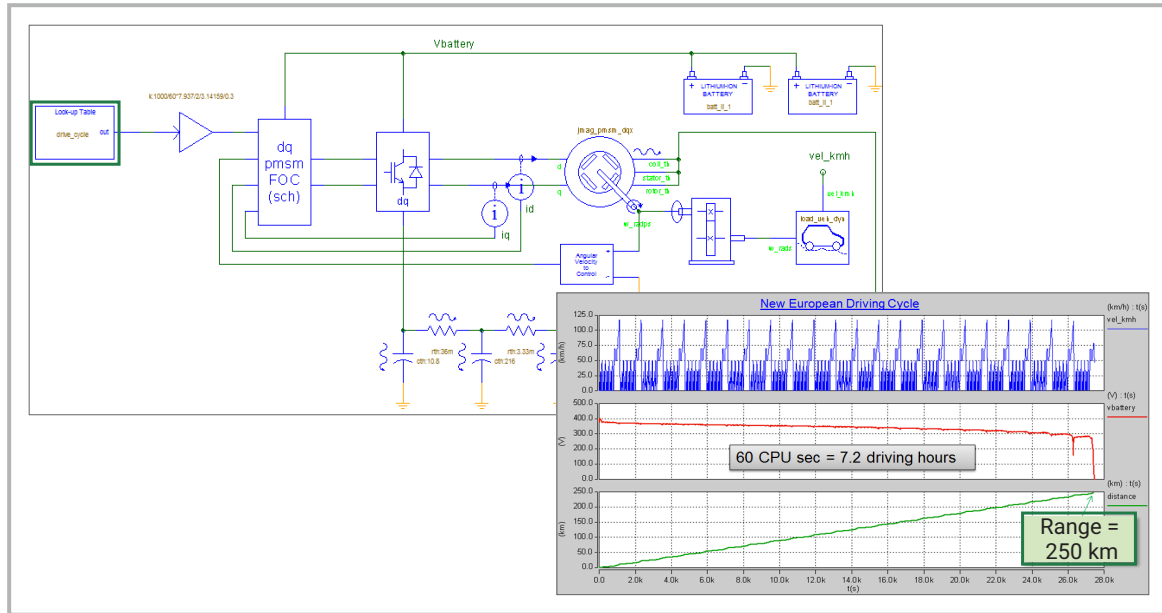


Figure 2: Driving Cycle Simulation

Such studies involve optimizing parameters like gear ratios and transition speeds under certain speed and acceleration performance constraints. Another use case example at this level is the thermal management of the motor and inverter over long thermal time constants. To run fast, the motor model abstracts out the sinusoidal nature of the three-phase currents and only retains in its equations the slow-moving voltage and current in the synchronous DQ reference frame. Despite this abstraction, the motor can accurately account for the speed-dependent motor losses implemented as lookup tables produced by JMAG FEA solver.

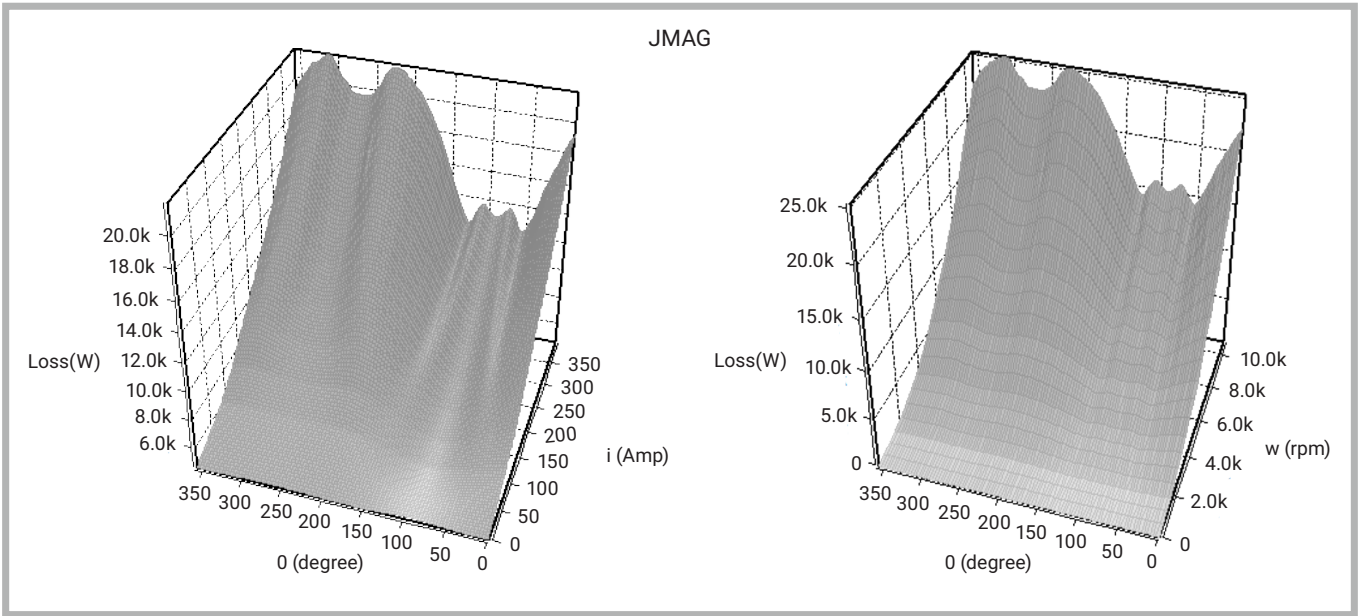


Figure 3: JMAG FEA Solver

Likewise, the inverter model is a fast non-switching model that also operates in the synchronous DQ reference frame and accounts for losses through lookup tables extracted from IGBT or MOSFET data sheets.

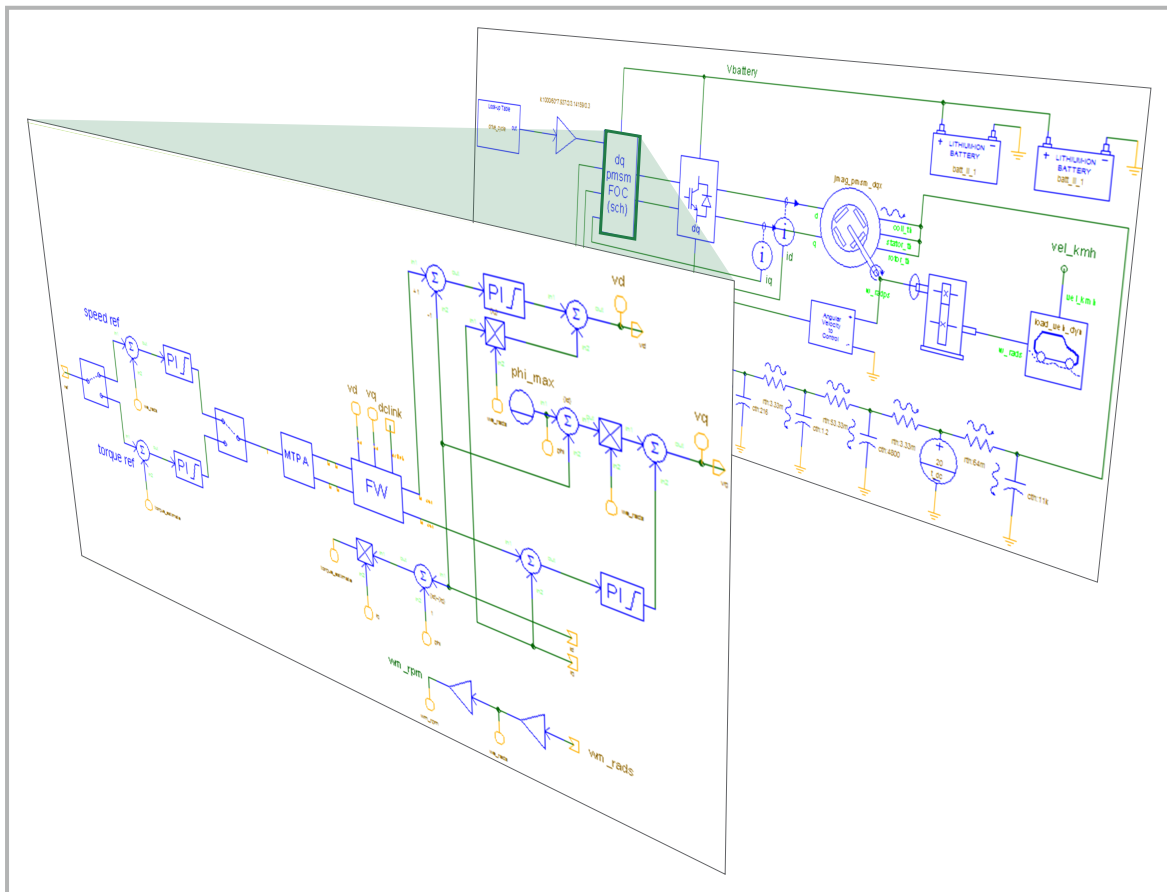


Figure 4: Saber Field Oriented Control

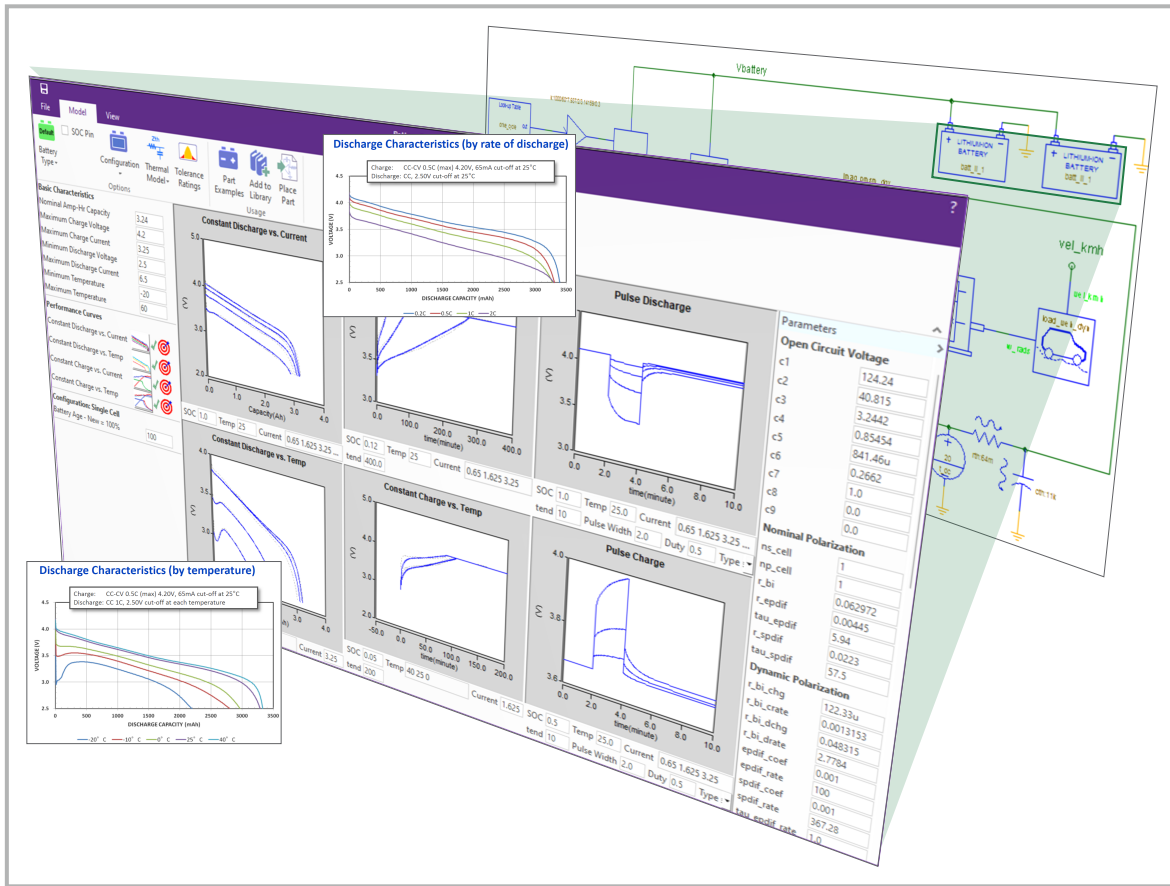


Figure 5: Saber Battery Modeling Tool

Certain powertrain elements don't need different abstraction models due to their relatively slow dynamics or large time constants, such as the vehicle dynamics model or the field-oriented control (vector control) model. There are several possibilities to implement control in Saber. One shown here is a schematic (Figure 4) implementation that leverages building blocks from the Saber control library. Control can also be implemented with custom C/C++ code run through a C block, with imported Simulink models, or with models written in MAST or VHDL-AMS languages (that run natively in Saber). Another model that simulates fast enough and does not necessarily need level-specific incarnations is the battery. Saber includes a battery modeling tool (Figure 5) that by default produces a fast battery pack model defined as a scaled cell. However, there is also an option in the battery tool to generate a pack of distributed battery cells in any series and parallel combination. Such a distributed pack will run slow but is needed to analyze and mitigate possible imbalance between cells. The battery tool supports lithium-ion as well as lead-acid chemistries. It optimizes model parameters to fit charge and discharge curves from data sheets or measurements. The tool also supports basic aging effects.

Use Case 2: Phase Current and Rotor Angle Analysis

This use case accounts for the actual sinusoidal phase currents and the actual rotor angle (instead just the rotor speed in use case 1). It allows motor imperfections such as torque ripples, vibration, and acoustic noise to be analyzed. Saber can simulate the thermal management of the motor and inverter over extended periods, ensuring efficient heat dissipation and preventing overheating.

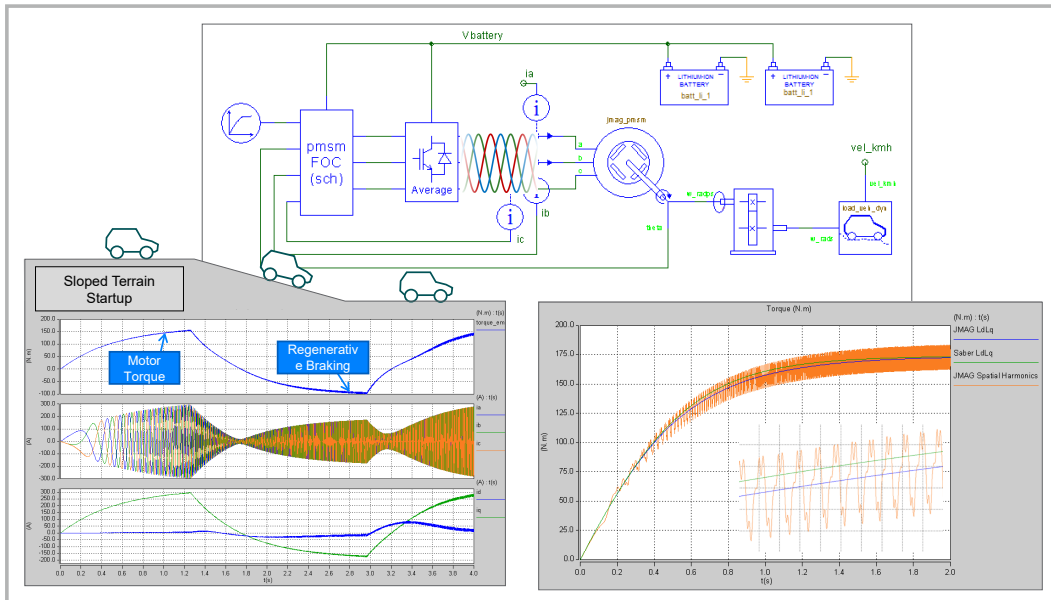


Figure 6: EV Motor Torque Ripple Analysis Using Saber

This helps in designing smoother and quieter motor operations. Motor models at this level of abstraction incorporate surface responses of torque and magnetic flux calculated by finite element analysis tools like ANSYS Maxwell or JMag. Finite element analysis is slow, but the evaluation of lookup tables in Saber is fast.

Use Case 3: PWM Switching in Inverters

This use case includes the PWM switching operation of the inverter. It allows faults in the inverter like opens and shorts across the switches to be analyzed. Fault mitigation by the control can then be evaluated. Ideal switches and diodes are used for faster simulations.

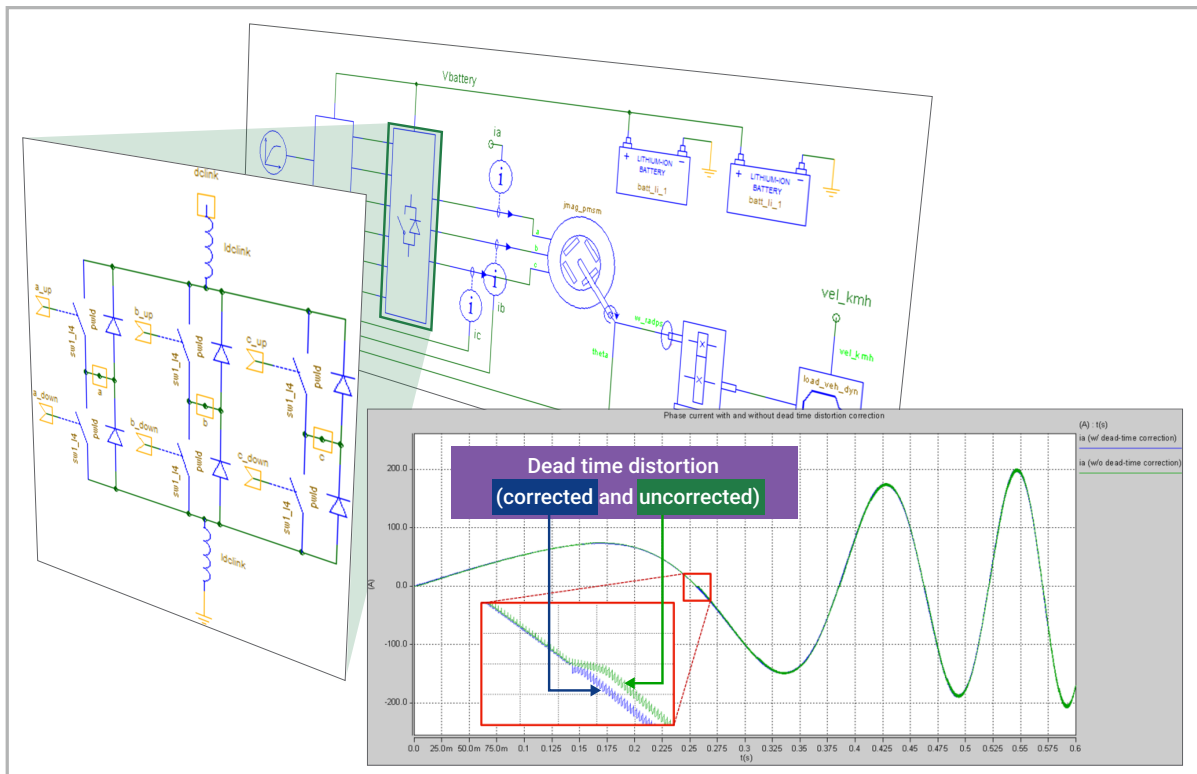


Figure 7: Saber PWM Switching Analysis

Dead time distortion can also be analyzed. If left uncorrected, dead time distortion can contribute to higher THD and iron losses in the motor.

Use Case 4: Semiconductor Analysis

This use case produces information on switching losses, overshoots, and EMI from semiconductors in the inverter. Detailed models of IGBTs or MOSFETs are used to analyze accidental turn-on switching and current imbalance in parallel devices. Accidental turn-on switching in MOSFETs (or IGBTs) through their Miller capacitance between drain (or collector) and gate, or through their parasitic inductance at the source (or emitter) are analyzed and worked on at this level. Also, when FETs are paralleled to reduce stress or increase current handling, current imbalance and even self-sustained oscillations can develop with harmful consequences.

Dedicated modeling tools are available in Saber for IGBTs, diodes, FETs, fuses, and more. Just like in the battery tool, the model parameters are optimized to fit datasheet curves. The models produced can be static or dynamic-thermal, can include statistical tolerances and stress ratings for Monte Carlo and stress analysis. The MOSFET tool has been shown to produce high-fidelity models for wide bandgap devices.

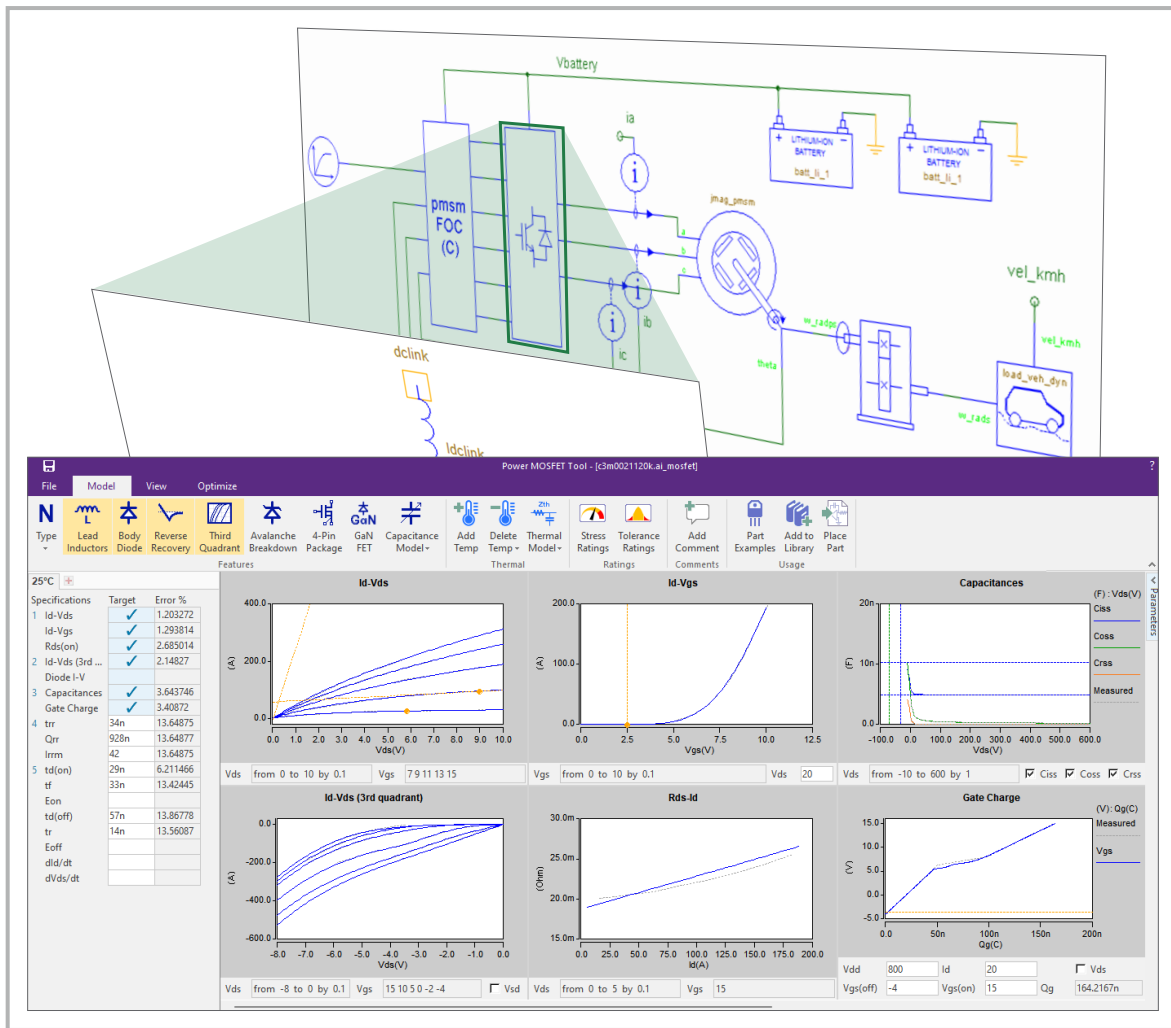


Figure 8: Saber Semiconductor Analysis

Virtual Prototyping for Battery Charging Systems with Synopsys Saber

Battery charging systems contain various components, each presenting design challenges. The complexity depends on the type of power supply, power transfer capability, and the topology selected for power electronic converters. Virtual prototyping helps to monitor battery performance and converter efficiency, allowing for more efficient design changes and parameter optimization.

For battery charging systems, we also distinguish different abstraction levels:

Non-Switching Behavioral Models

The onboard charging module shown below runs fast because of its abstractions. Except for the front-end rectifying bridge, which is made of ideal diodes and commutates at 50 Hz, the other downstream converters are implemented with non-switching models. The PFC Boost is a non-switching behavioral model accounting for closed-loop dynamics. The isolated full-bridge DC-DC converter is a state-space average model. Finally, the output of this converter connects to the battery with its charge controller. The charge controller ensures proper charging without exceeding the battery ratings, first under constant current and then under constant voltage as the battery approaches a fully charged state. The charge control algorithm is implemented as a state machine with the Saber built-in StateAMS modeling tool. The constant charging current is a parameter of this model, and the simulation results show that the current command matches the current delivered to the battery. This abstraction is practical to design and validate the charge control algorithm.

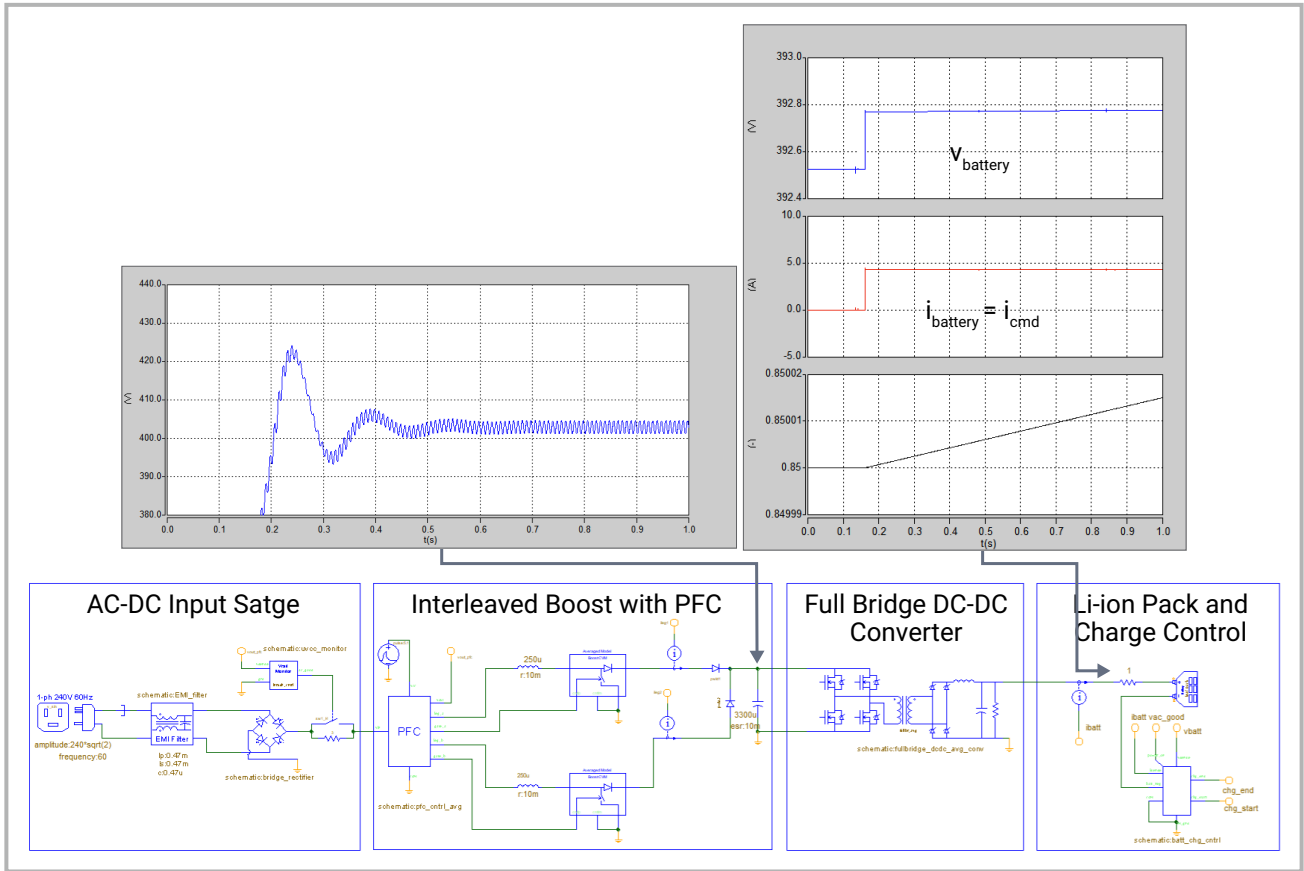


Figure 9: Saber StateAMS Modeling Tool

Higher Abstraction Levels: Detailed Switching Models

This includes detailed switching models for converter to analyze issues like total harmonic distortion (THD) and power factor. The figure below shows an interleaved totem pole PFC (followed by a CLLC resonant DC-DC converter) where the THD is directly measured on the current waveform with the Saber waveform measurement tool. A similar waveform measurement for the power factor shows a value close to 1, indicating that the control is properly synthesizing an input current in phase with the input voltage while regulating the output DC voltage to the demanded value.

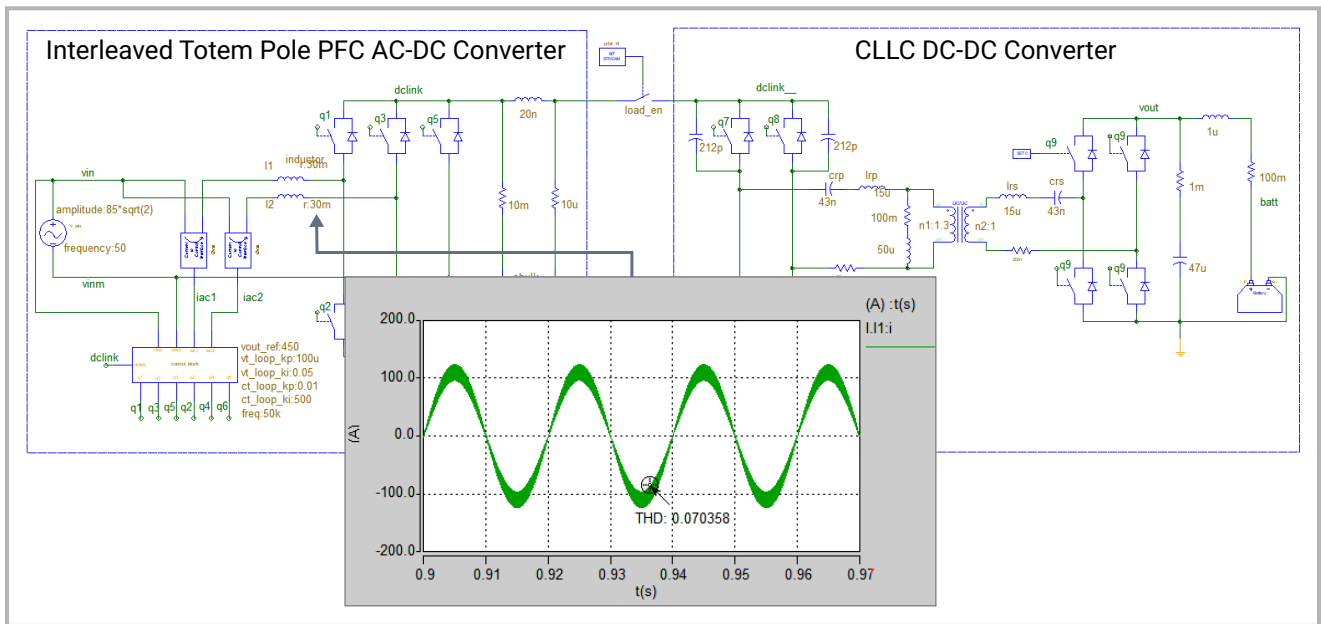


Figure 10: Saber THD and Power Factor Analysis for Converters

This implementation of the control uses continuous PI compensators (proportional integral compensators) and a delay corresponding to half a switching period inserted in the PWM generation for one of the interleaved branches to cancel out input current ripples. An alternative implementation uses a C block to perform the regulations with Z-domain digital filters written in C. The sampling frequency of 500 kHz gives very comparable results but translates into a slower simulation.

Conclusion

EV powertrain development can realize reduced time to market, higher quality, and engineering resource optimization by using virtual prototypes. Virtual prototypes allow for the accurate modeling and representation of the full mechanical, electrical, and magnetic characteristics of these systems.

In this whitepaper we present two use cases of how Saber can accelerate EV development, electric powertrain and charging systems. Saber is a comprehensive solution for developing and validating power electronics and multi-domain physical systems. It contains over 29,000 generic and characterized models, detailed abstraction levels, and specific components based on actual data

For more information Saber for mechatronics and power electronics please visit:
<https://www.synopsys.com/saber>

References:

- [1] <https://ntrs.nasa.gov/citations/20210023699>
- [2] <https://www.marketsandmarkets.com/Market-Reports/electric-vehicle-market-209371461.html>
- [3] <https://www.jabil.com/blog/automotive-industry-trends-point-to-shorter-product-development-cycles.html>
- [4] <https://ieeexplore.ieee.org/document/8449955>