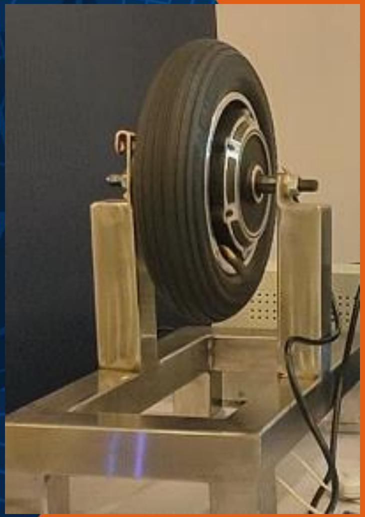
MathWorks
AUTOMOTIVE CONFERENCE 2023
India

Master class - Driving Efficiency and Performance using Motor Control Workflow for Electric Vehicle

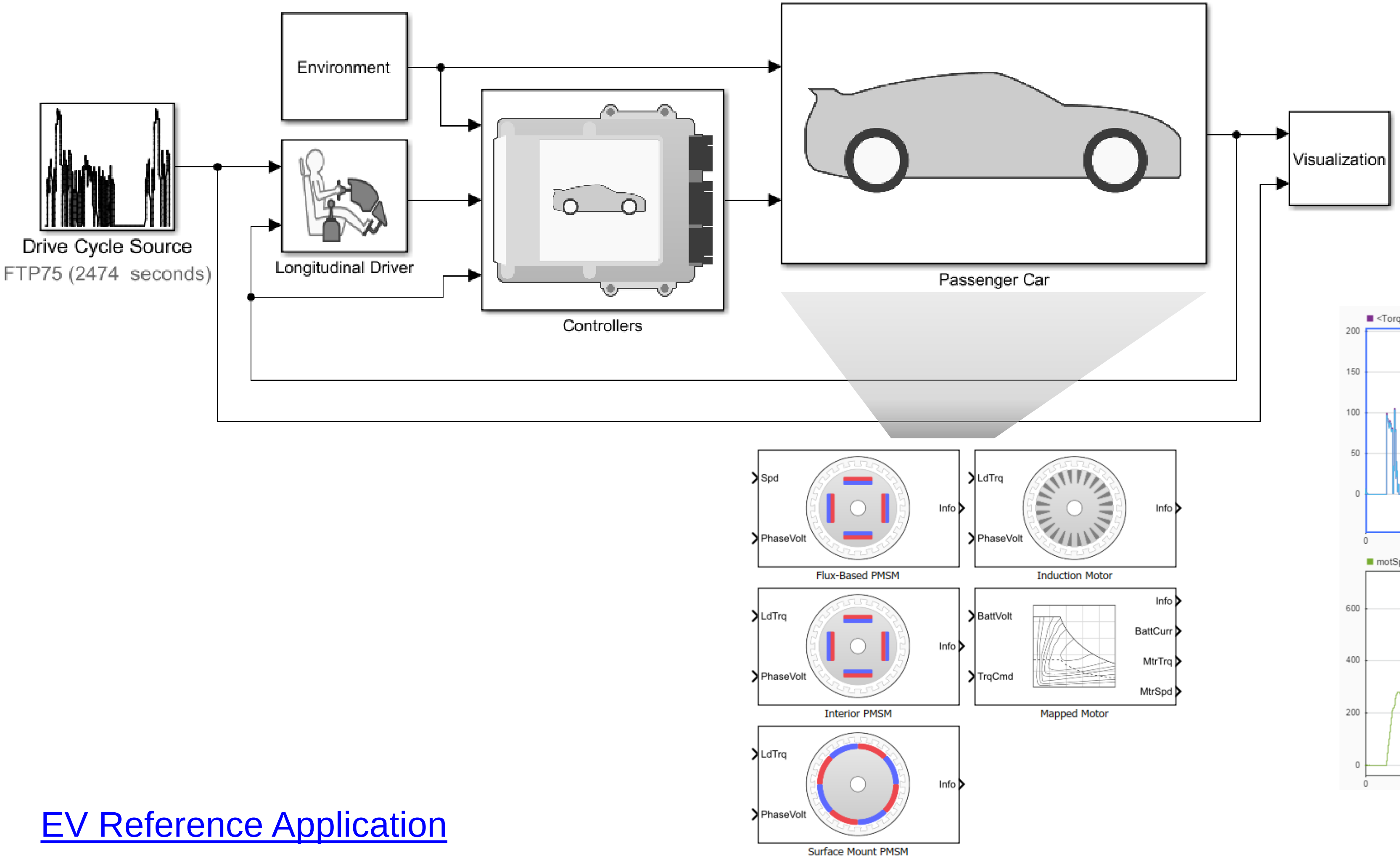
Rahul Choudhary, MathWorks

Ananth Kumar, MathWorks



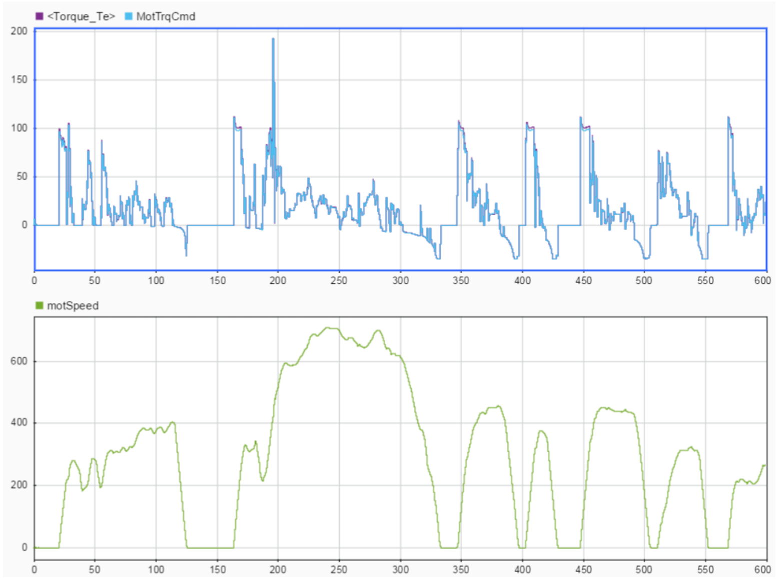
MathWorks

The Heart of the Electric Powertrain: Motors



- Component Selection?
- Component Sizing?
- Trade-off Studies?

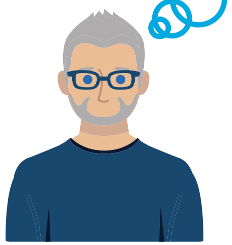
- Detailed Component Modeling and Control Design?



[EV Reference Application]()

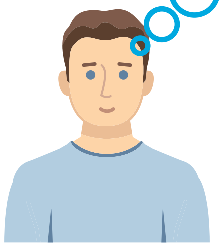
Challenges in Motor Control Development for Electric Vehicle

What steps I should follow to implement efficient field-weakening control?



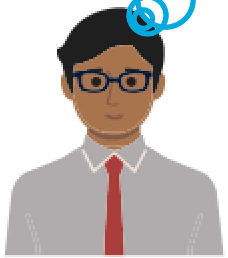
EV Start-up Engineer

How do I get motor parameters to develop my motor model?



Modeling Engineer

How do I generate efficient code and port to different hardware ?

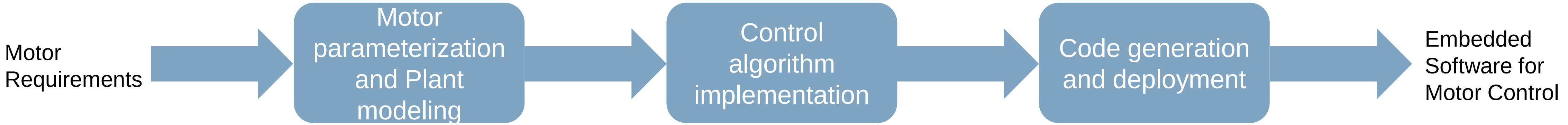


Embedded Engineer

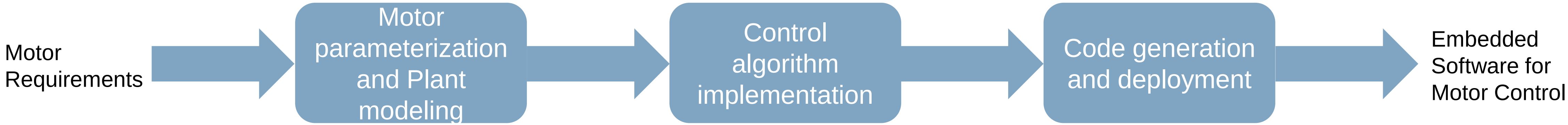
How do I tune my Control loop gains to achieve the performance?

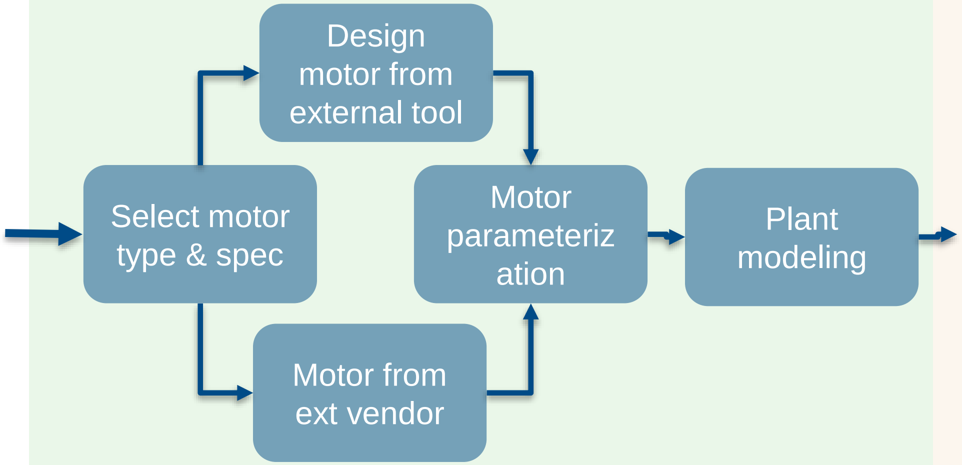
Control Engineer

Proposed motor control software development workflow for EV



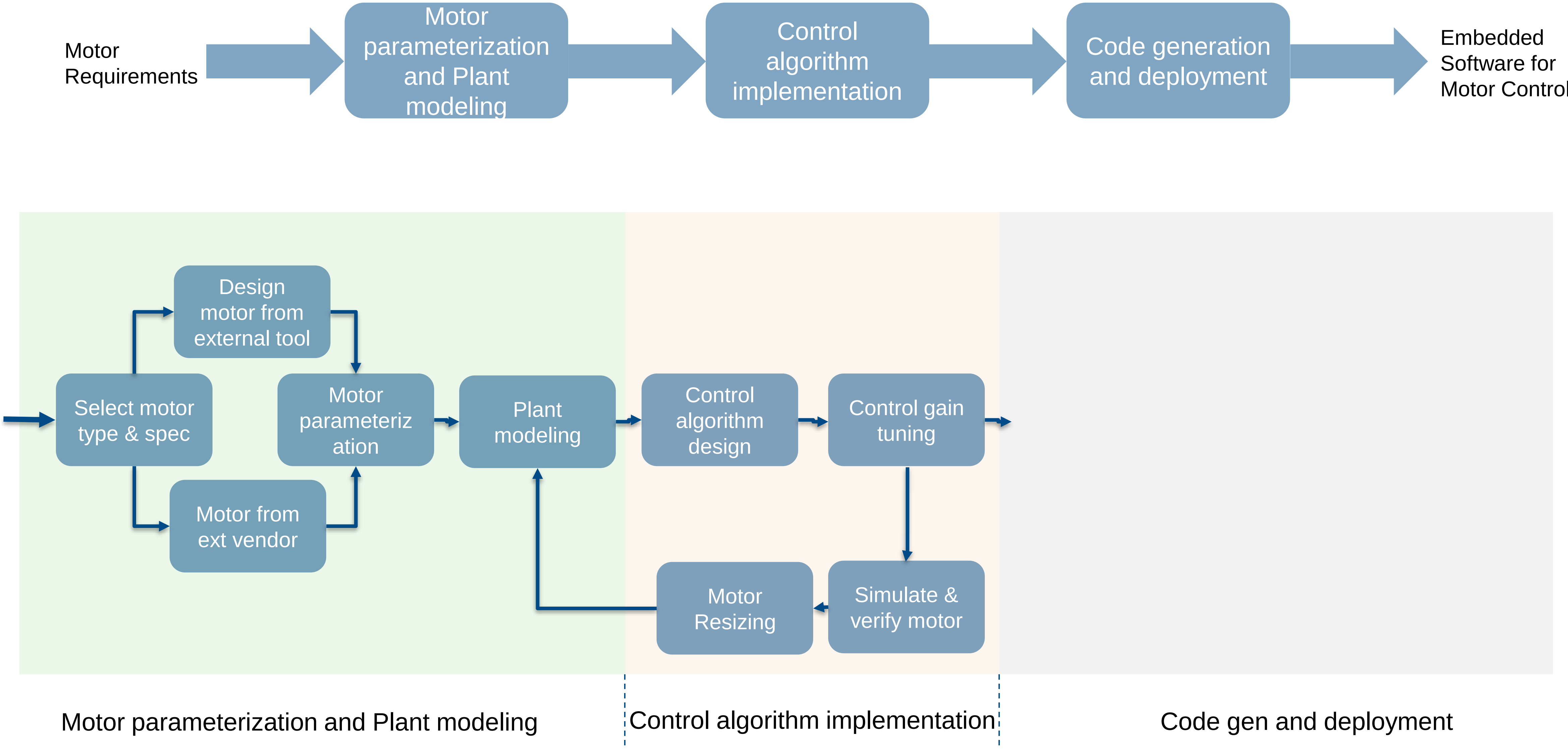
Proposed motor control software development workflow for EV



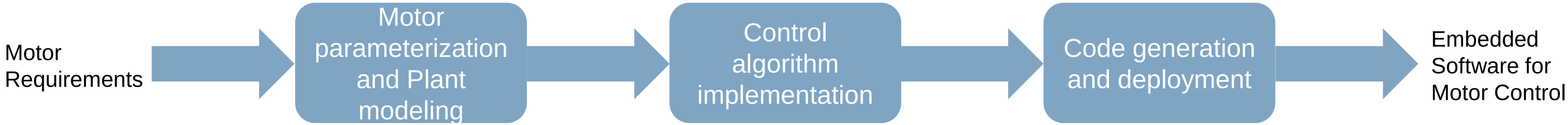


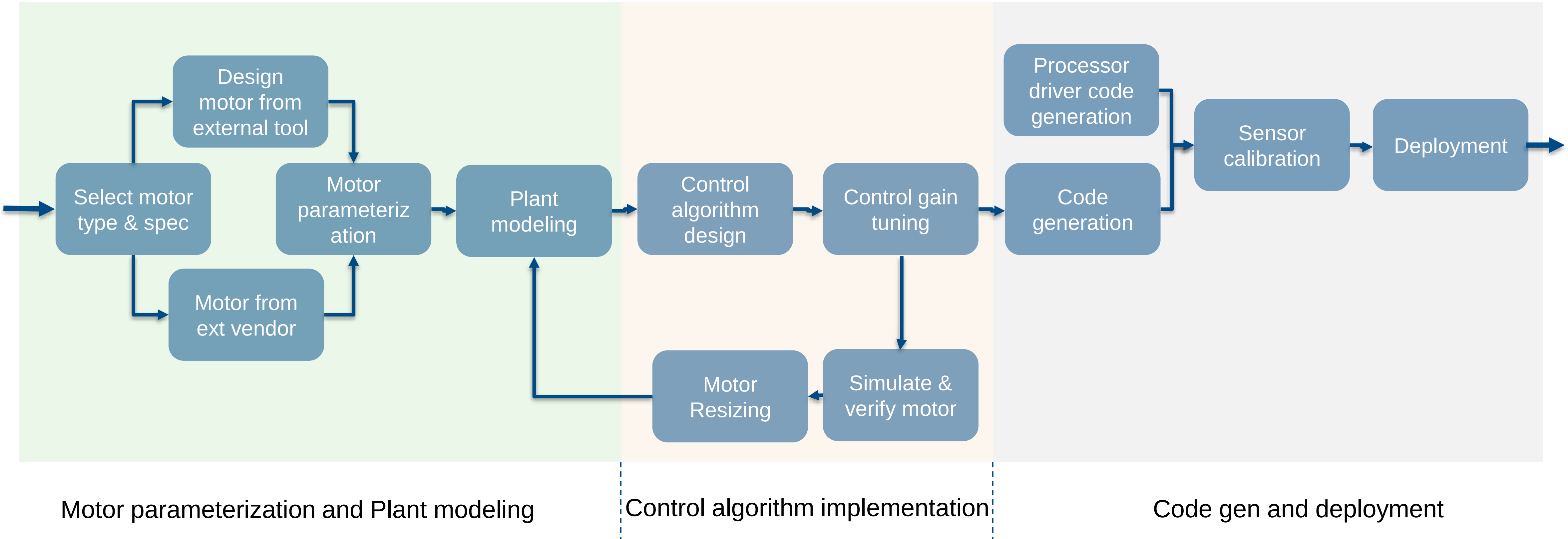
Motor parameterization and Plant modeling | Control algorithm implementation | Code gen and deployment

Proposed motor control software development workflow for EV



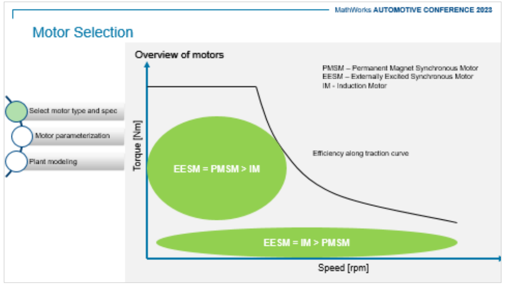
Proposed motor control software development workflow for EV



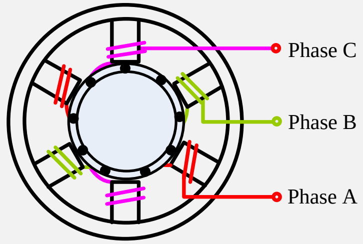


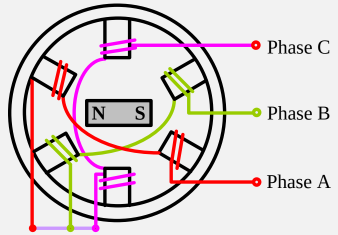
Motor Selection

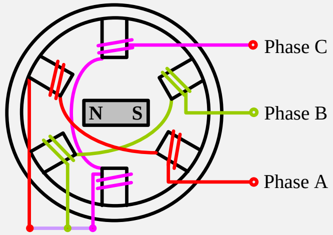
- Select motor type and spec
- Motor parameterization
- Plant modeling

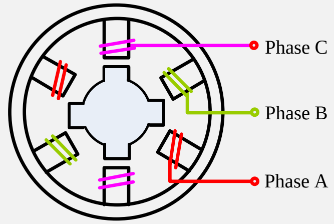


Overview of motors



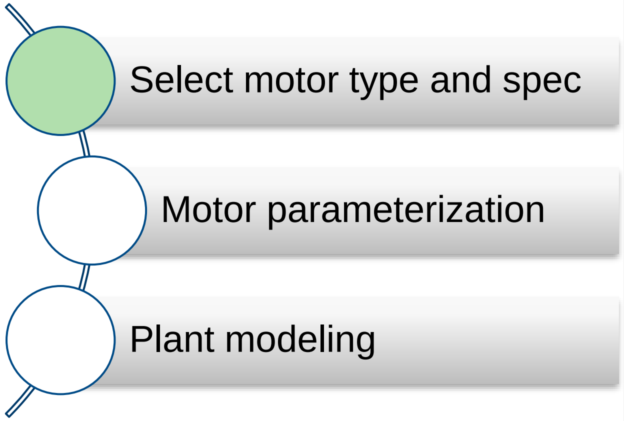






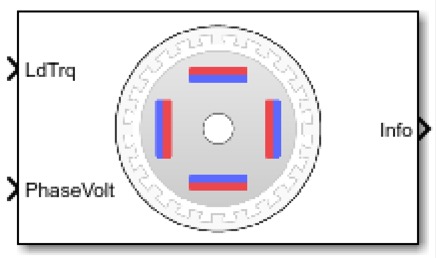
Squirrel Cage Induction Motor (IM)	Brushless DC Motor (BLDC)	Permanent Magnet Synchronous Motor (PMSM)	Switched Reluctance Motor (SRM)
Advantages			
– Low material cost/ kg – Robust, reliable, Simple Control – Less maintenance	– Efficient, reliable – High Power Density and High torque at low speed	– Low noise, smooth operation – High performance and efficiency over operating range	– Simple and robust construction – Low cost, long constant power range
Disadvantages			
– Low efficiency, low power factor specially at light loads, Heavy, High Copper loss	– Higher torque ripples and cogging torque leads to vibration, noise and poor position control	– High Cost – Demagnetizing risk – Requires complex control	– High noise, torque ripples – Complex Control – High Switching losses
Industry Examples			
Tesla model X, Toyota RAV4 EV, Renault Zoe	Hero Electric, Yamaha EC-03, Bounce	Tesla Model S, Chevrolet Bolt, Hyundai Kona Electric	Jaguar I-PACE Concept, Ford Fiesta EV Prototype

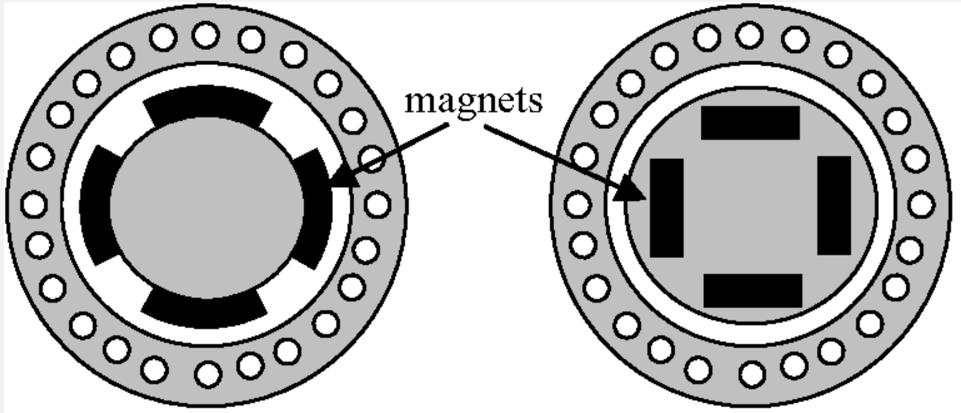
Motor Selection



In this presentation, we will focus on Interior PMSM because of:

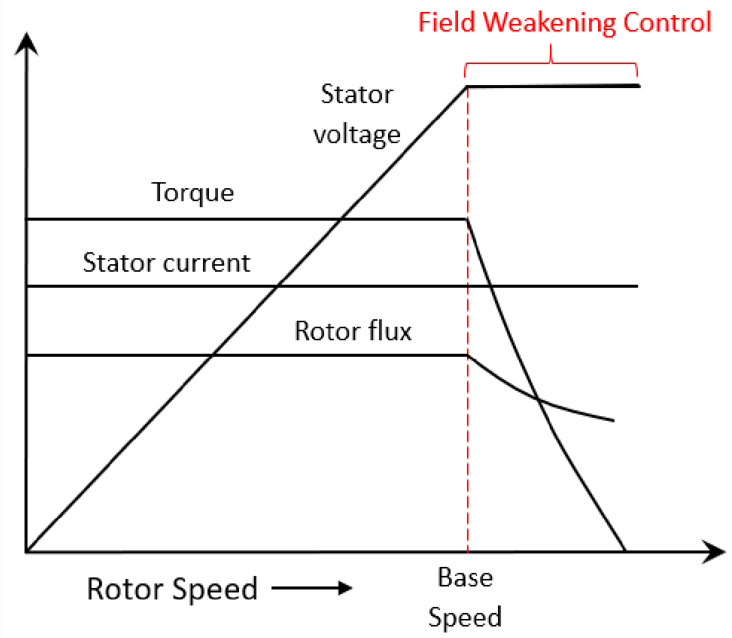
- High power density and efficiency
- Motor can spin more than base speed





Surface Mount PMSM

Interior PMSM



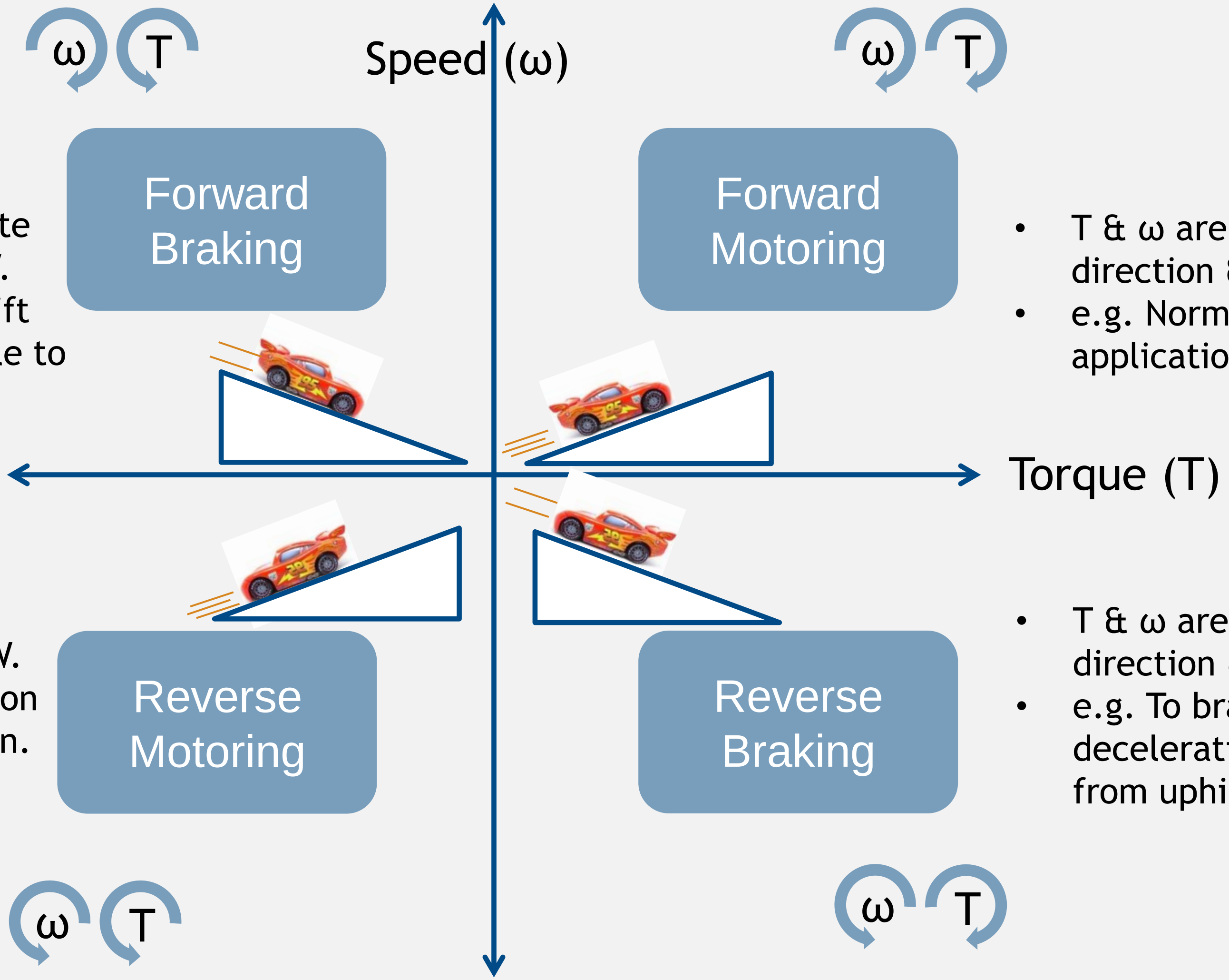
Motor characteristics

Motor Selection

- Select motor type and spec
- Motor parameterization
- Plant modeling

Motor functionality – Four quadrant operations

- T & ω are in opposite direction & ω is CW.
- e.g. Stop a falling lift
- e.g. Stopping vehicle to avoid collision with some object.



- T & ω are in same direction & are CW.
- e.g. Normal motoring application.

- T & ω are in same direction & are CCW.
- e.g. Reverse direction motoring application.

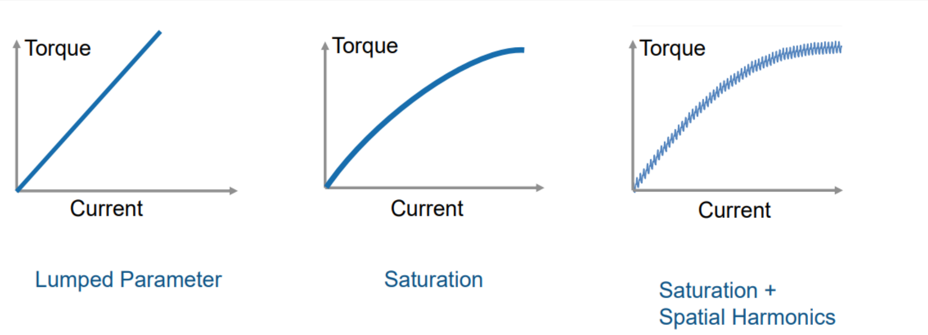
- T & ω are in opposite direction & ω is CCW.
- e.g. To brake a decelerating vehicle from uphill slope.

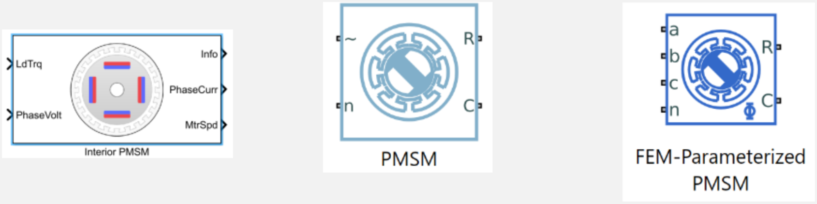
- Note: CW direction of T & ω is taken as +ve ref for explanation only.

Motor Parameterization and Model Fidelity

- Select motor type and spec
- Motor parameterization
- Plant modeling

Motor Model Fidelity Level

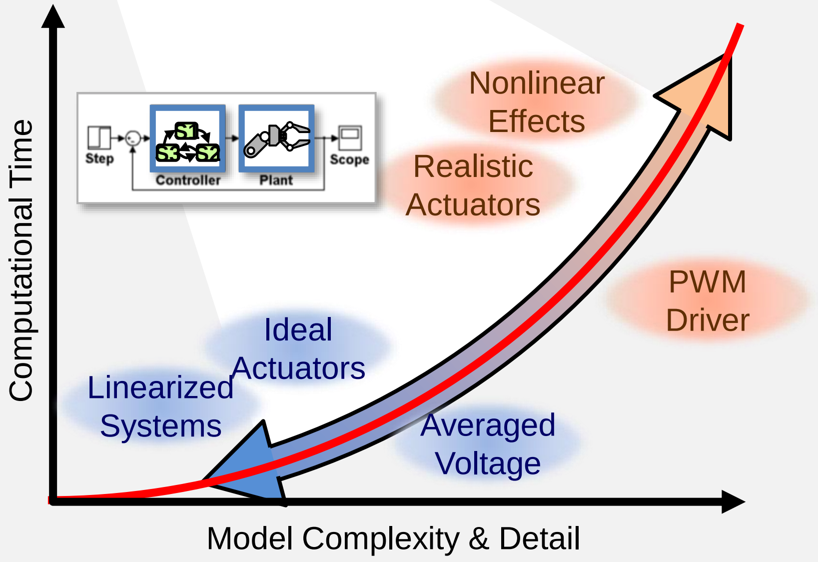




Motor Control Blockset

Simscape Electrical

Computational Time vs. Model Complexity

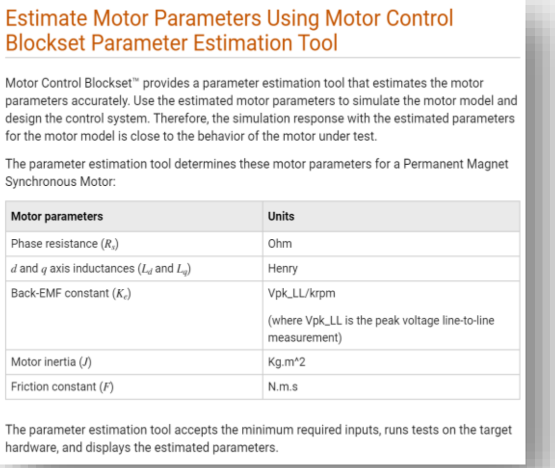


Motor Parameterization and Plant Modeling

- Select motor type and spec
- Motor parameterization
- Plant modeling

Motor Parameterization

If motor exist	If motor and dyno test setup are available	Motor doesn't exist but FEA data from motor design tool is available



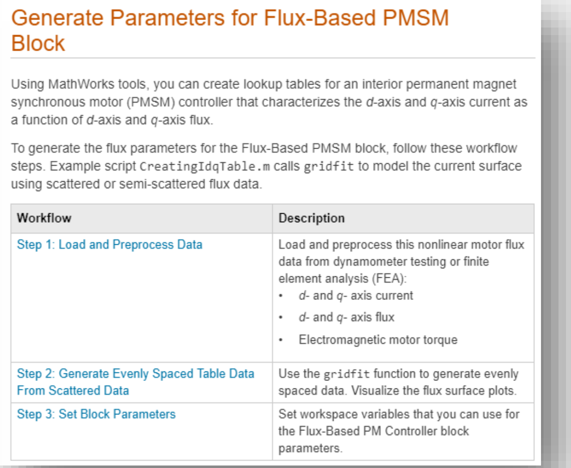
Estimate Motor Parameters Using Motor Control Blockset Parameter Estimation Tool

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

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

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

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

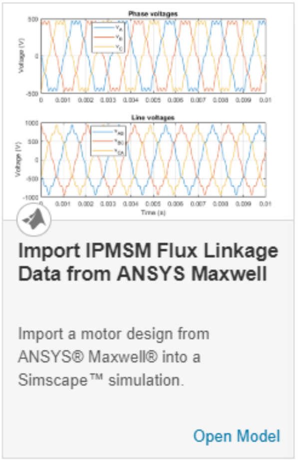


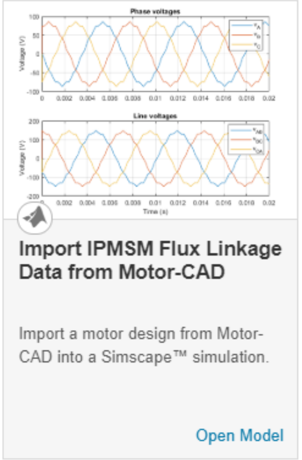
Generate Parameters for Flux-Based PMSM Block

Using MathWorks tools, you can create lookup tables for an interior permanent magnet synchronous motor (PMSM) controller that characterizes the *d*-axis and *q*-axis current as a function of *d*-axis and *q*-axis flux.

To generate the flux parameters for the Flux-Based PMSM block, follow these workflow steps. Example script `CreatingIdqTable.m` calls `gridfit` to model the current surface using scattered or semi-scattered flux data.

Workflow	Description
Step 1: Load and Preprocess Data	Load and preprocess this nonlinear motor flux data from dynamometer testing or finite element analysis (FEA): • *d*- and *q*- axis current • *d*- and *q*- axis flux • Electromagnetic motor torque
Step 2: Generate Evenly Spaced Table Data From Scattered Data	Use the `gridfit` function to generate evenly spaced data. Visualize the flux surface plots.
Step 3: Set Block Parameters	Set workspace variables that you can use for the Flux-Based PM Controller block parameters.





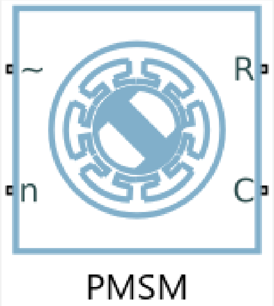
From datasheet or Instrumented tests running on the Hardware	From Dyno test data	From FEA tools such as ANSYS Maxwell, JMAG, Motor-CAD
Lumped Parameters	Saturation	Saturation + Spatial Harmonics

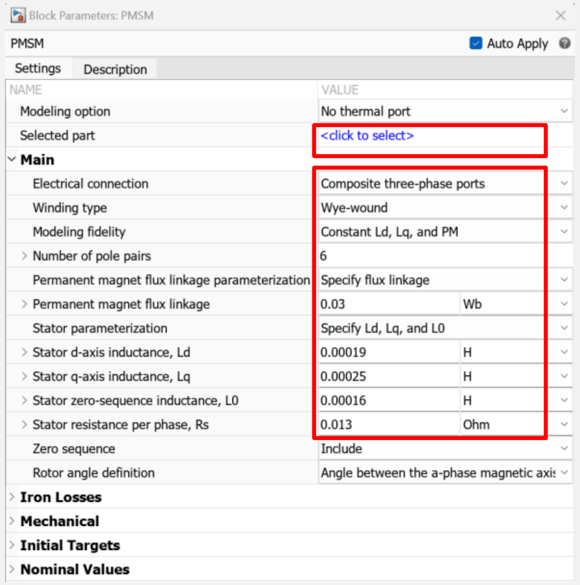
Parameterization helps with Motor Modeling which captures motor dynamics and helps us with Control Design

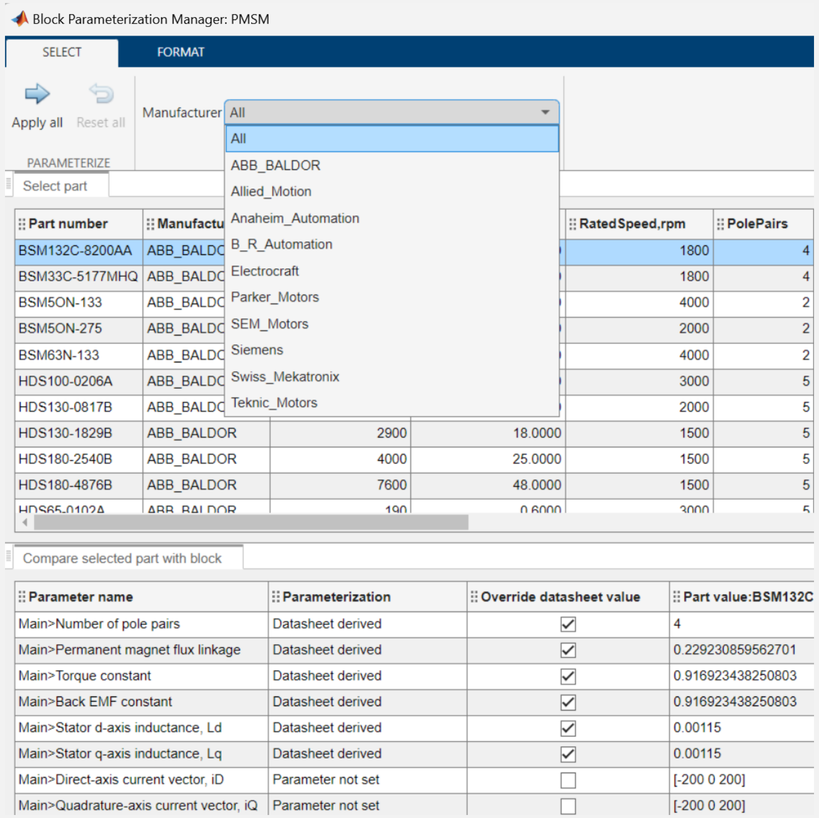
Motor Parameterization and Plant Modeling

- Select motor type and spec
- Motor parameterization
- Plant modeling

Motor Parametrization using Datasheet



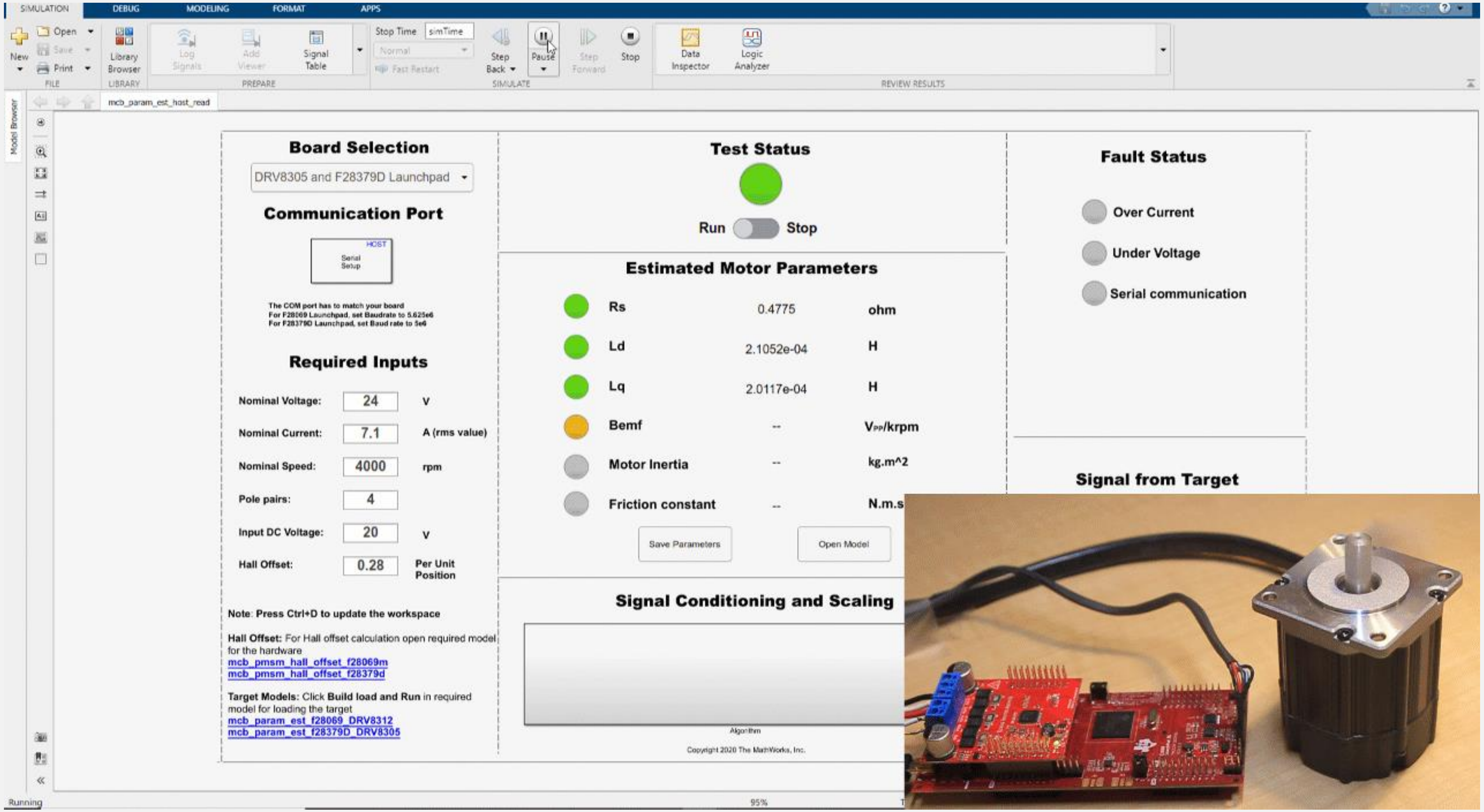




Motor Parameterization and Plant Modeling

- Select motor type and spec
- Motor parameterization
- Plant modeling

Motor parameters estimation from Instrumented tests

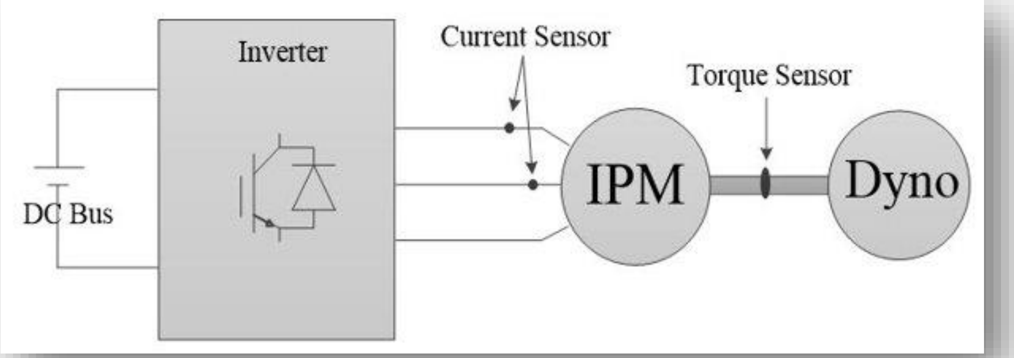


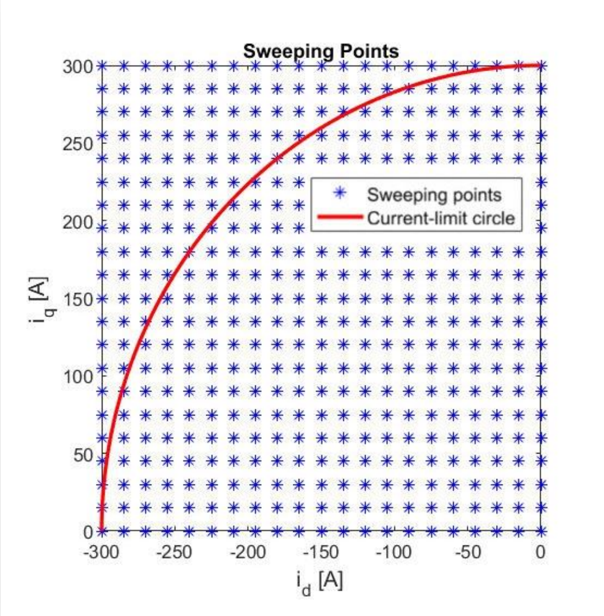
- Instrumented tests running on the target
- Sensor-based and Sensor-less modes available
- Supports PMSM and Induction Motor

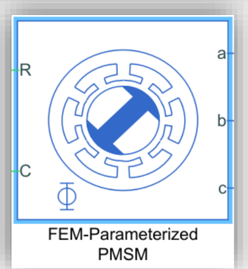
Motor Parameterization and Plant Modeling

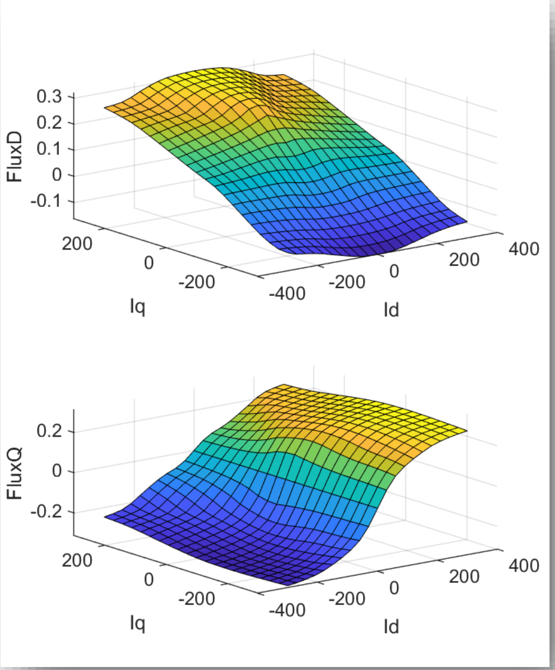
- Select motor type and spec
- Motor parameterization
- Plant modeling

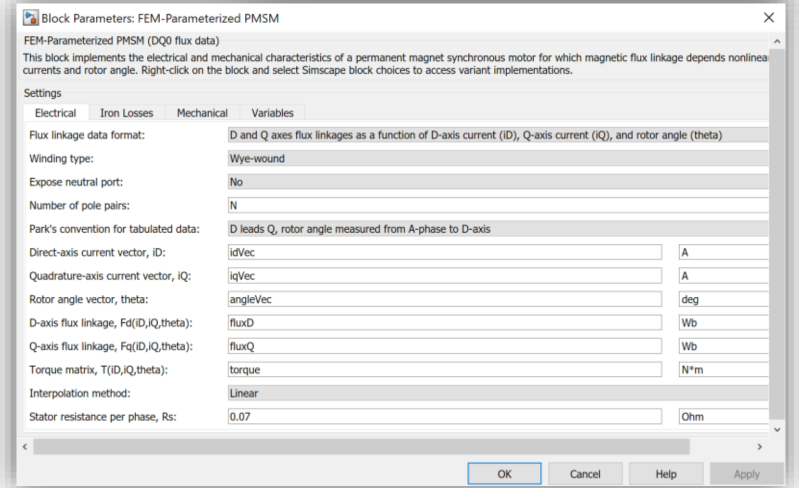
Motor parameters from dyno test







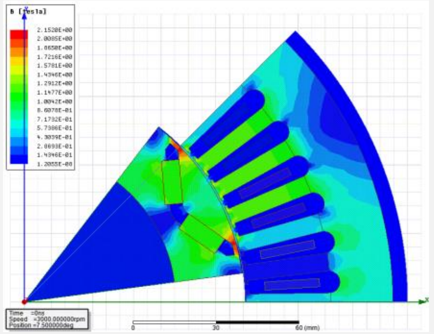


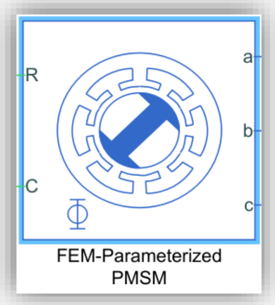


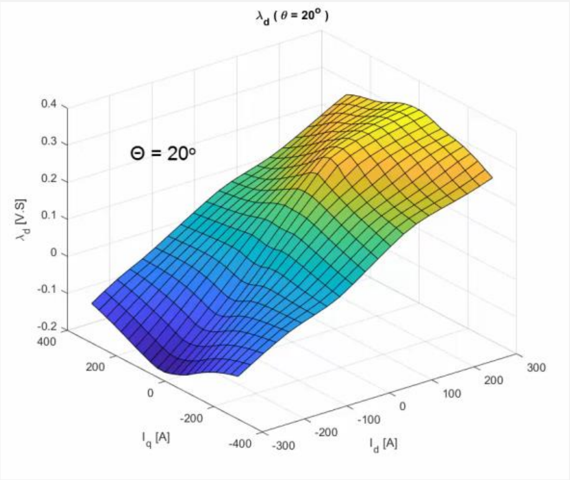
Motor Parameterization and Plant Modeling

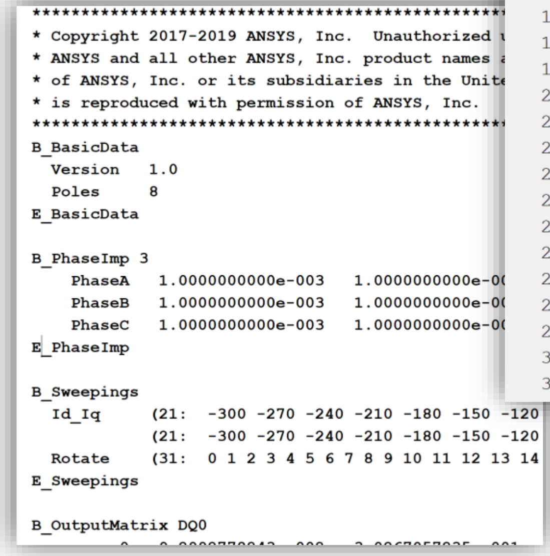
- Select motor type and spec
- Motor parameterization
- Plant modeling

Motor parameters from motor design tool

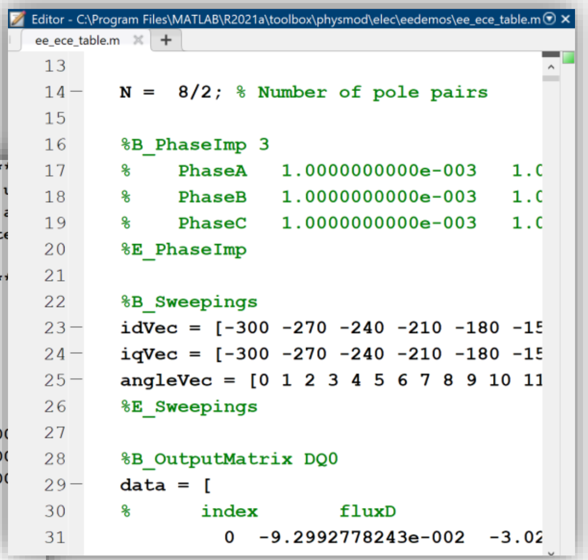








```
****************************************************
* Copyright 2017-2019 ANSYS, Inc.  Unauthorized u
* ANSYS and all other ANSYS, Inc. product names a
* of ANSYS, Inc. or its subsidiaries in the Unite
* is reproduced with permission of ANSYS, Inc.
****************************************************
B_BasicData
  Version    1.0
  Poles      8
E_BasicData

B_PhaseImp 3
    PhaseA   1.0000000000e-003   1.0000000000e-0
    PhaseB   1.0000000000e-003   1.0000000000e-0
    PhaseC   1.0000000000e-003   1.0000000000e-0
E_PhaseImp

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

B_OutputMatrix DQ0
```



Editor - C:\Program Files\MATLAB\R2021a\toolbox\physmod\elec\eedemos\ee_ece_table.m

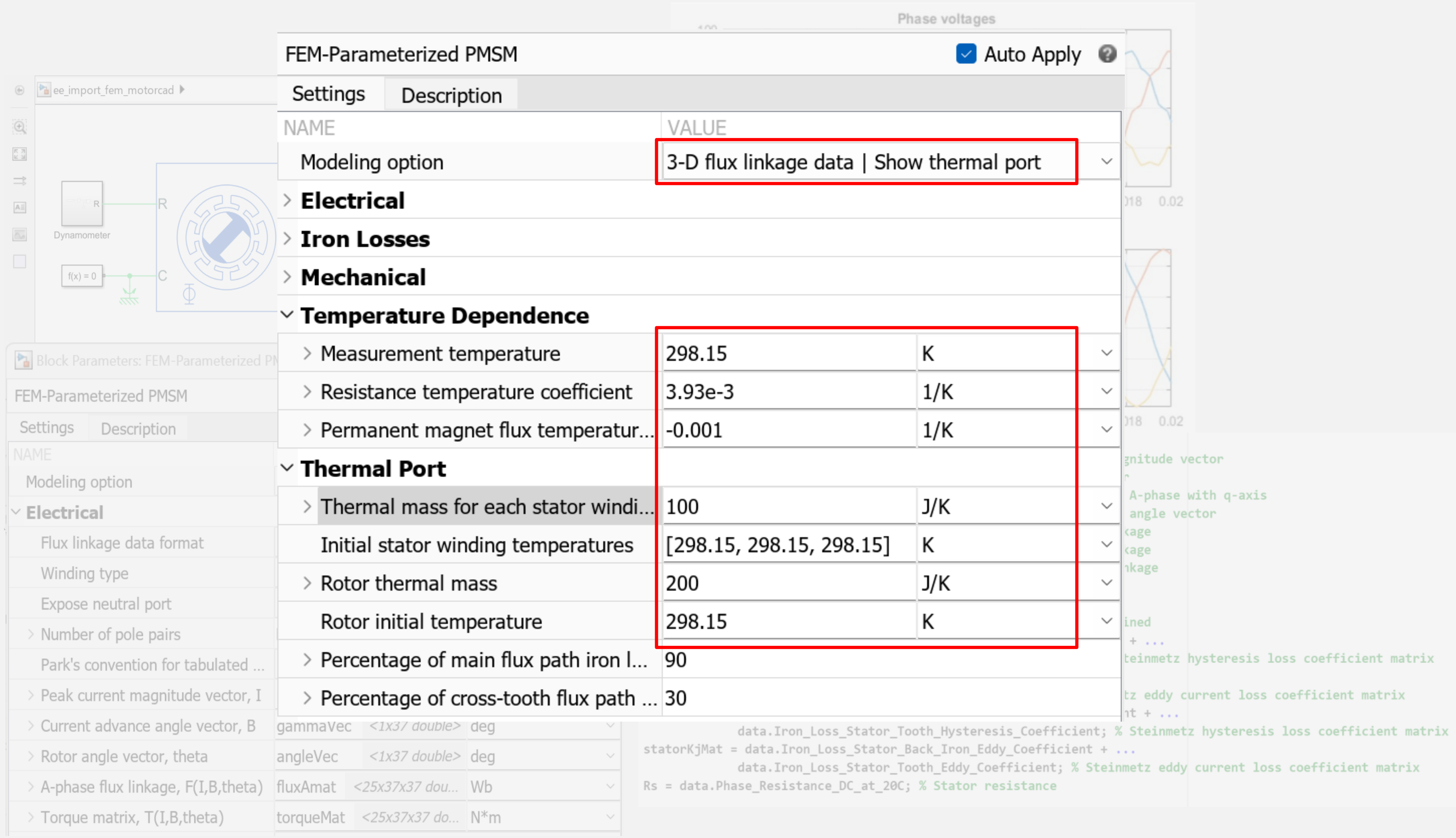
ee_ece_table.m

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

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

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

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

Motor Parameterization and Plant Modeling

- Select motor type and spec
- Motor parameterization
- Plant modeling

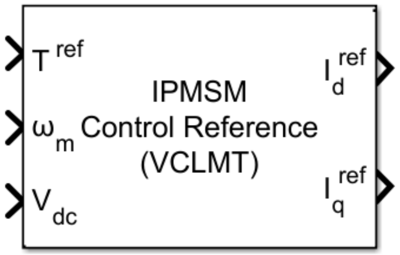
Motor parameters from motor design tool



Motor Constraint Curves and Characteristics

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

- Exploration Motor Characteristics



LUT based PMSM Control Reference

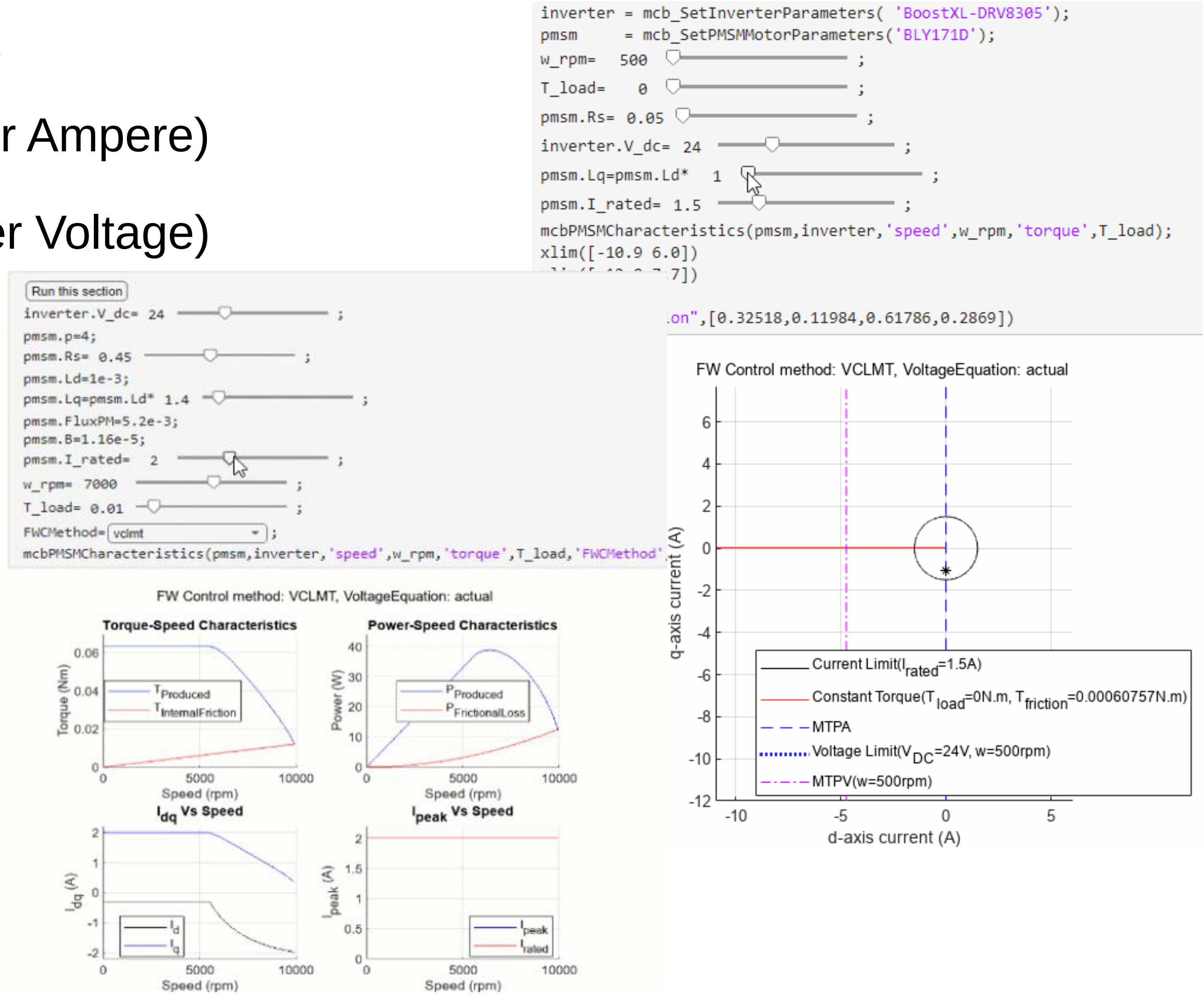


Figure : Dependency of curves and characteristics on parameters changes

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

Motor Parameterization and Plant Modeling

- Select motor type and spec
- Motor parameterization
- Plant modeling

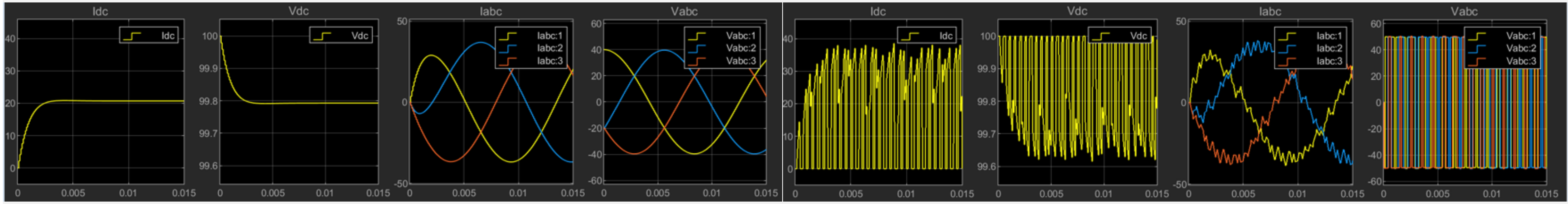
Power Converter Model Fidelity

System-Level Behavior	Component Validation	Component Design

Low Fidelity — **Composition** — High Fidelity

Abstracted | Waveform Controlled | PWM Controlled | Individual Switches

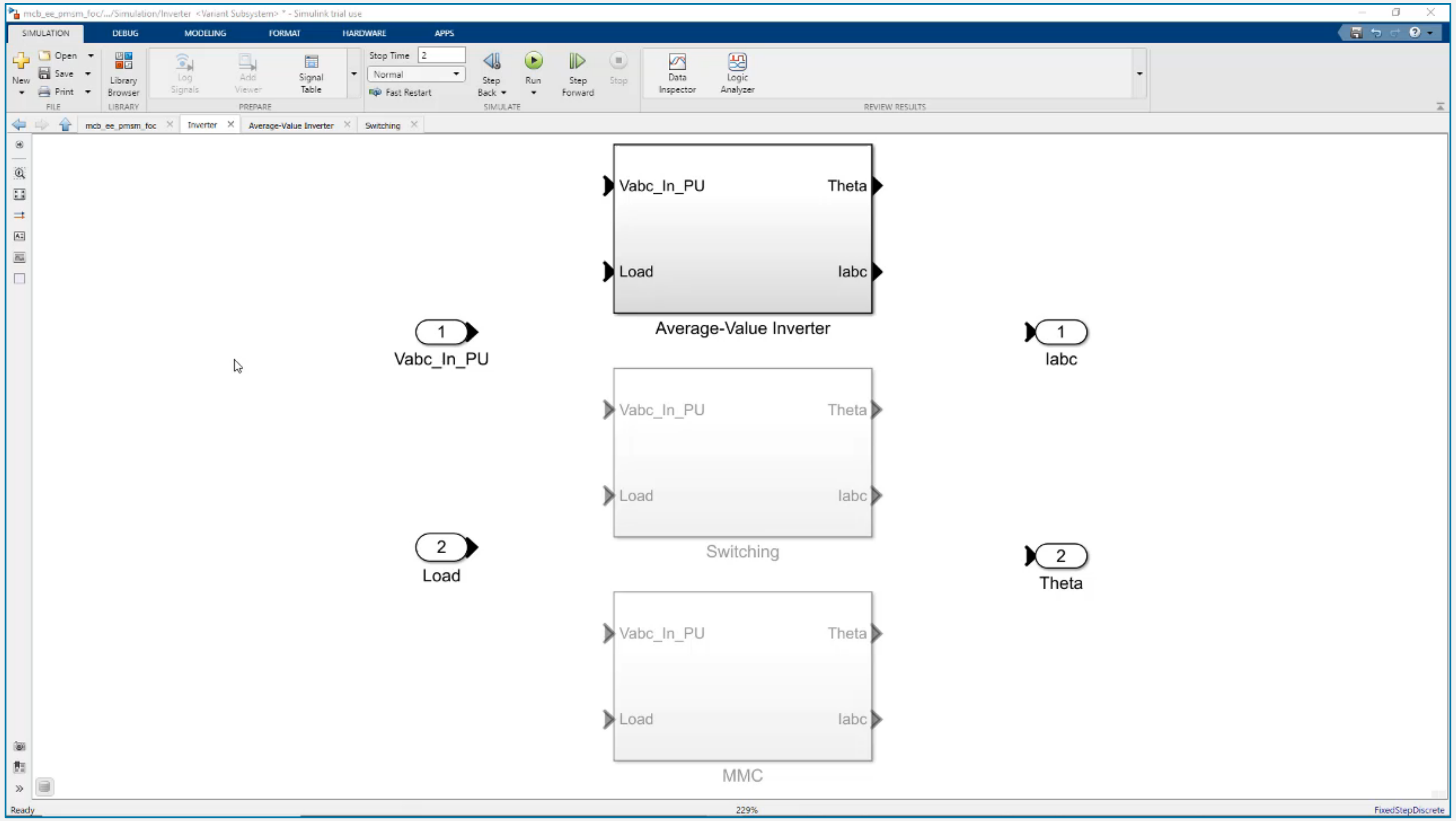
Semiconductor Converter Models



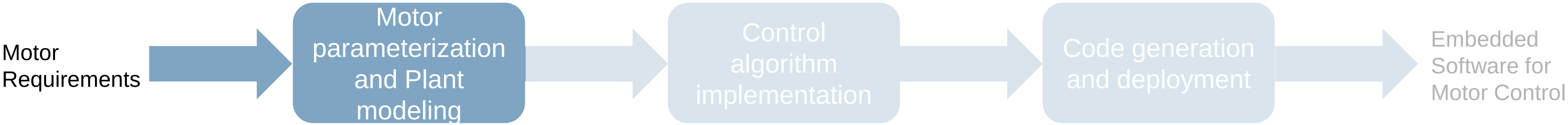
Motor Parameterization and Plant Modeling

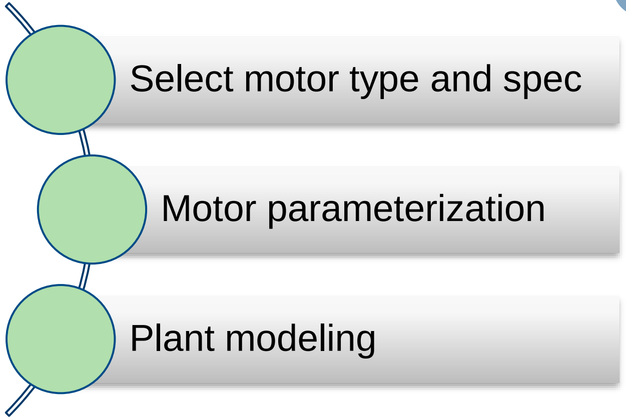
- Select motor type and spec
- Motor parameterization
- Plant modeling

Variant approach for different plant fidelity level



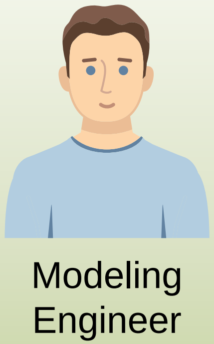
Motor Parameterization and Plant Modeling





Summary

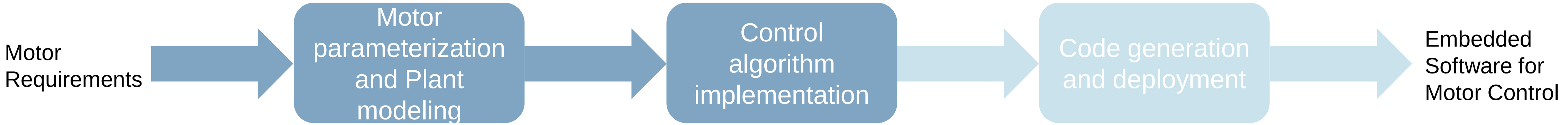
- Identify motor
- Workflow is defined for Interior PMSM
- Steps to parameterize Motor using datasheet, running instrumented tests, FEA tool or dyno test data
- Model motor and inverter with different fidelity levels

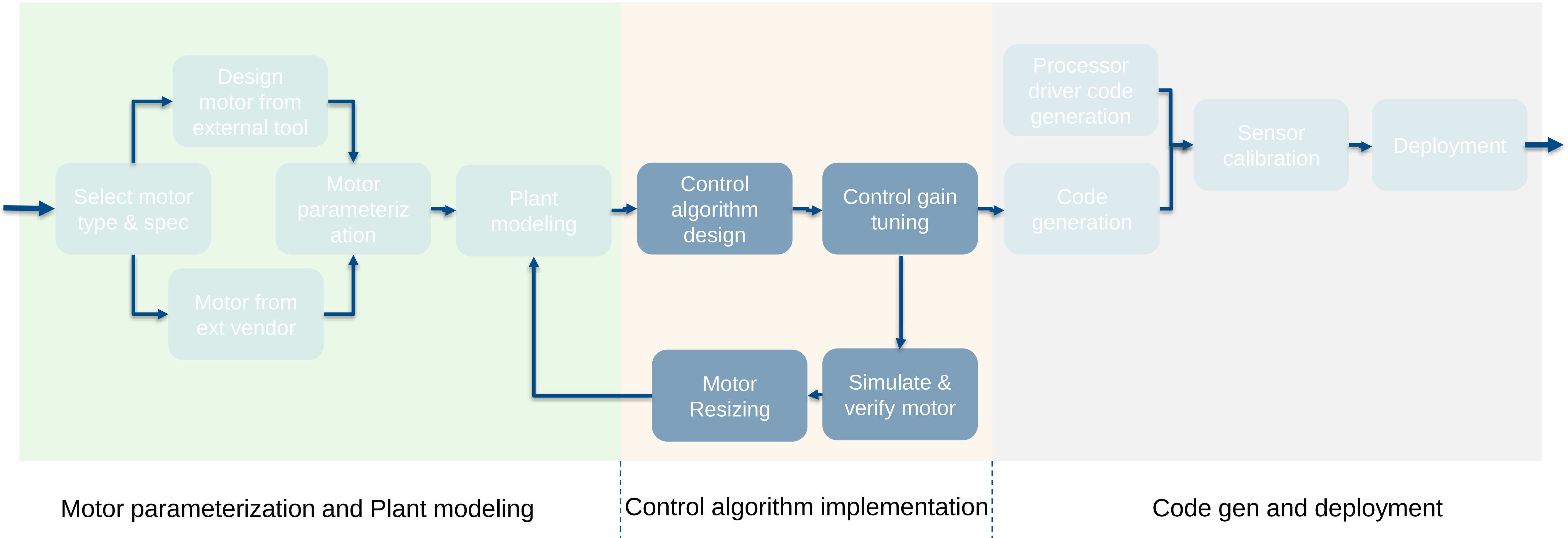


How do I get motor parameters to develop my motor model?

Use instrumented test or dyno test or FEA data for parameterizing the motor

Proposed motor control software development workflow for EV

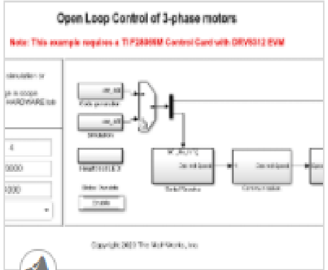




Control Algorithm Implementation

- Control algorithm design
- Control-loop gain tuning
- Simulate & Verify
- Motor resizing

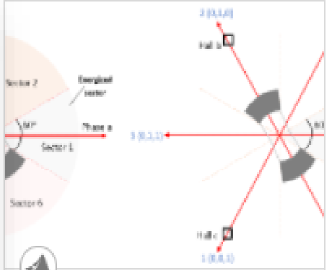
Control algorithm design - Different control strategies



Run 3-Phase AC Motors in Open-Loop Control and Calibrate ADC Offset

Uses open-loop control (also known as scalar control or Volts/Hz control) to run a motor. This technique varies the stator voltage

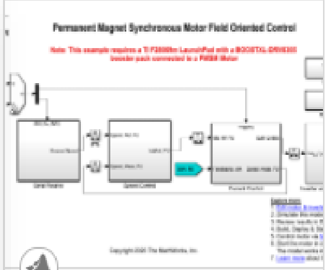
V/F control



Six-Step Commutation of BLDC Motor Using Sensor Feedback

Uses 120-degree conduction mode to implement the six-step commutation technique to control speed and direction of rotation of a

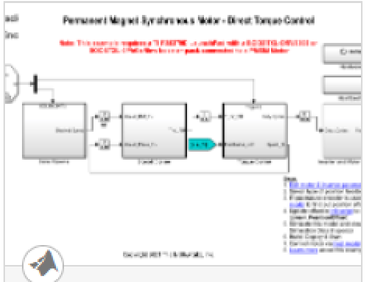
Six-step commutation for BLDC with Hall sensor



Sensorless Field-Oriented Control of PMSM

Implements the field-oriented control (FOC) technique to control the speed of a three-phase permanent magnet synchronous

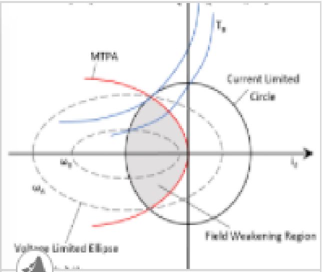
Field-oriented control



Direct Torque Control of PMSM Using Quadrature Encoder or Sensorless Flux...

Implements direct torque control (DTC) technique to control the speed of a three-phase permanent magnet synchronous motor (PMSM). Direct

Direct torque control



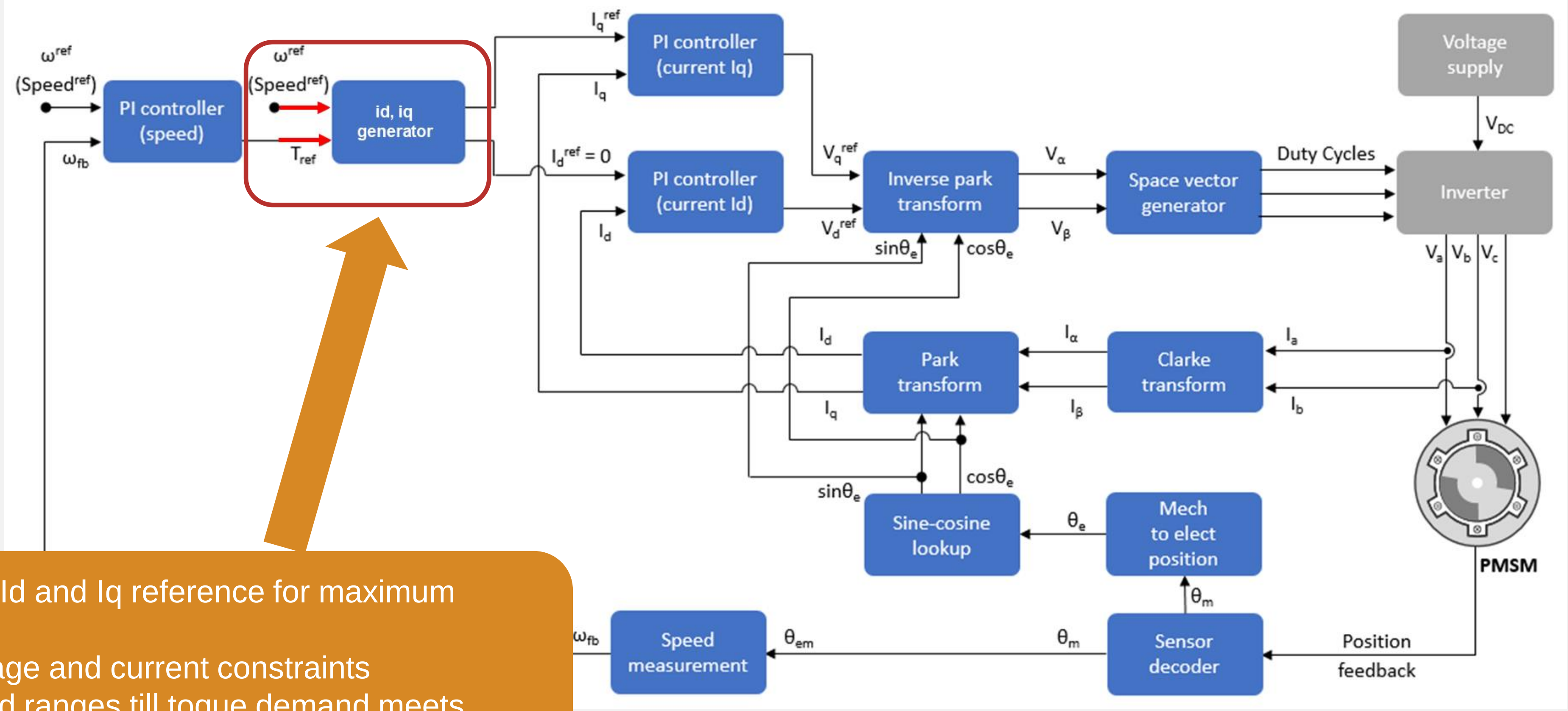
Field-Weakening Control (with MTPA) of PMSM

Implements the field-oriented control (FOC) technique to control the torque and speed of a three-phase permanent magnet

Field-weakening control

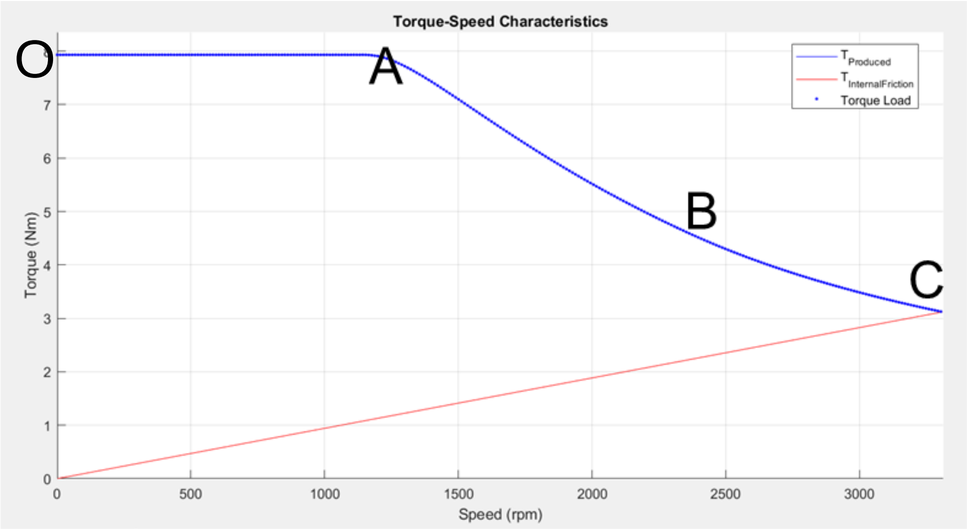
Field-weakening control architecture with block to generate optimum control reference

- Control algorithm design
- Control-loop gain tuning
- Simulate & Verify
- Motor resizing

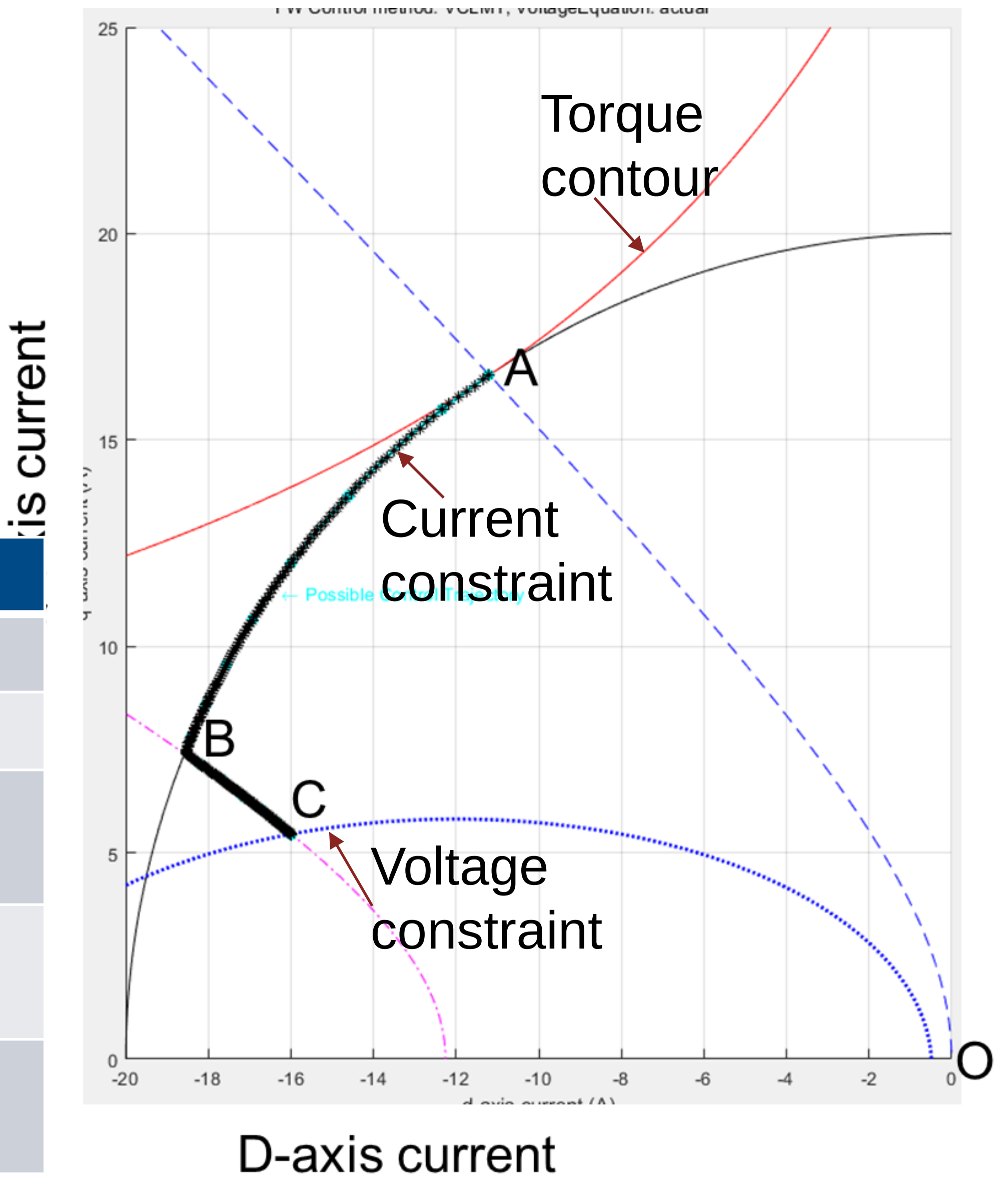


1. Output right Id and Iq reference for maximum torque
2. Honour voltage and current constraints
3. Output speed ranges till toque demand meets frictional torque

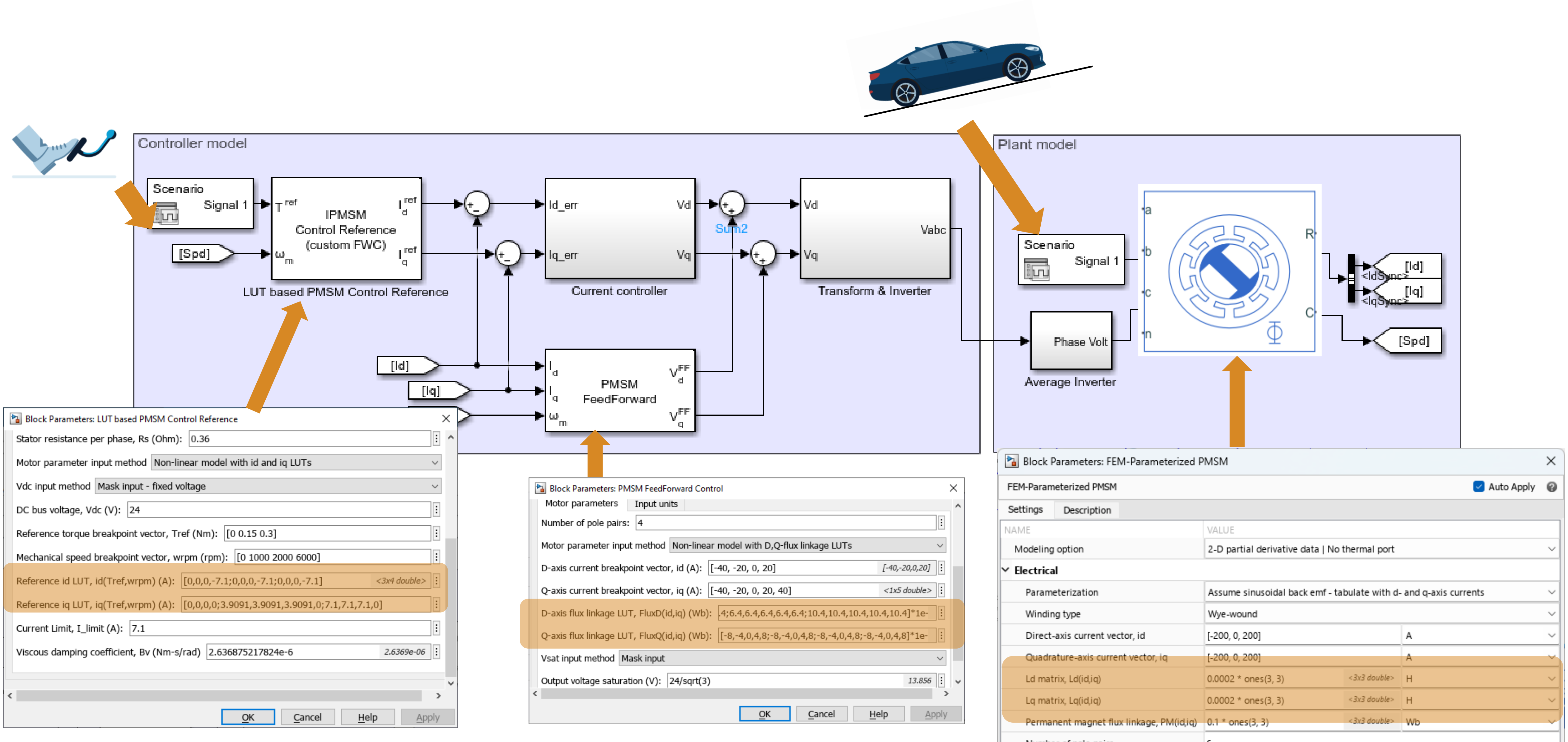
Compare critical operating points in Torque-Speed characteristic curve and Id_ref, Iq_ref plot



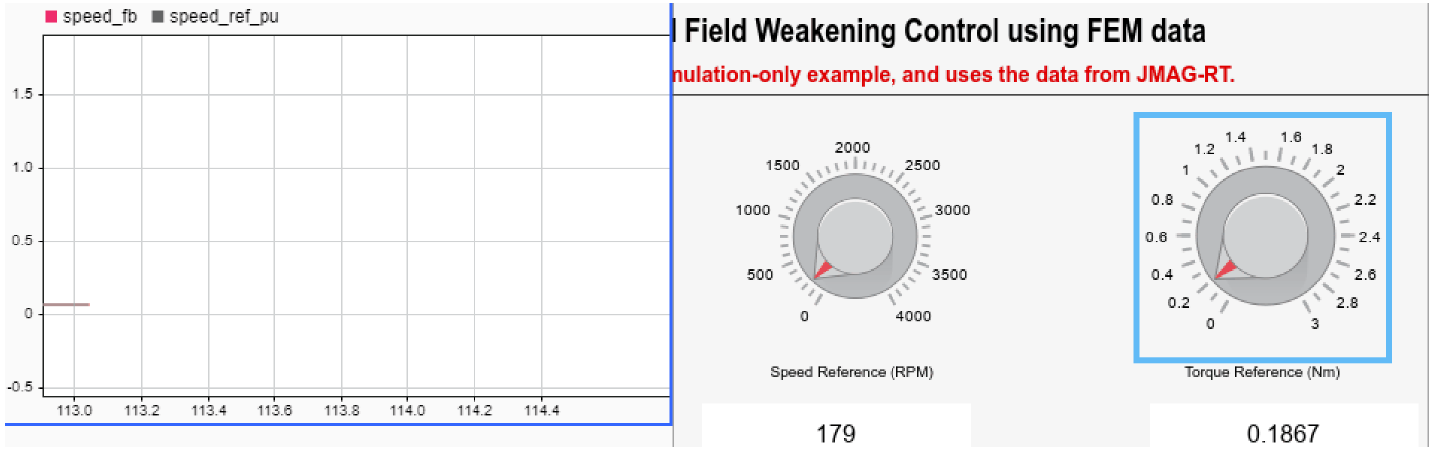
Region	
OA	Maximum torque per Amphere (MTPA)
AB	Field-weakening control (Beyond base speed)
BC	Field-weakening control honoring Maximum Torque per volt (MTPV)
CO	Field-weakening control honoring Voltage constraint limit
C	C is max achievable speed where frictional torque equals motor deliverable torque



Implementing Field-weakening control with motor-dyno test data



Field-weakening control for Interior PMSM



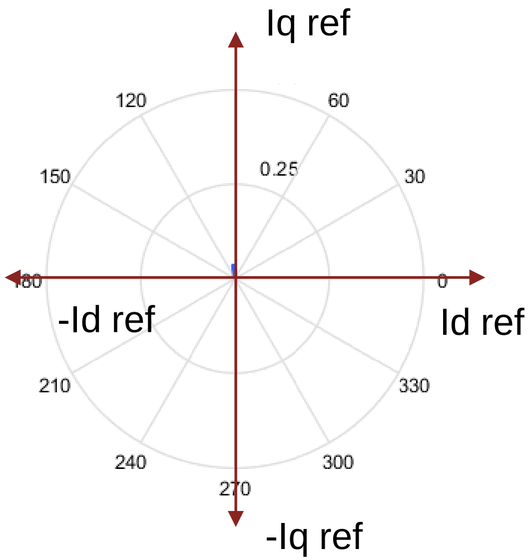
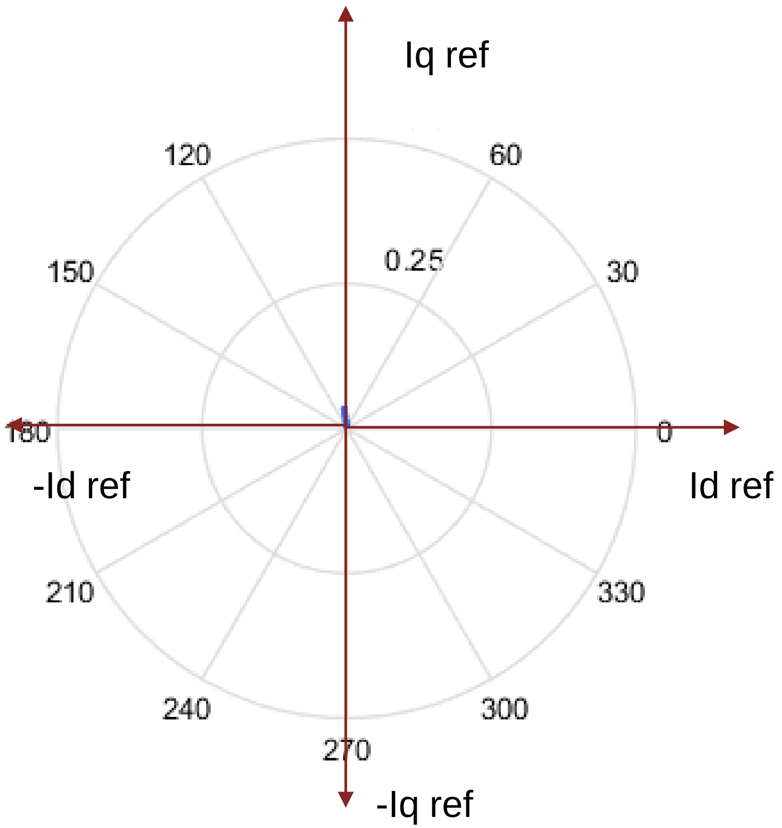
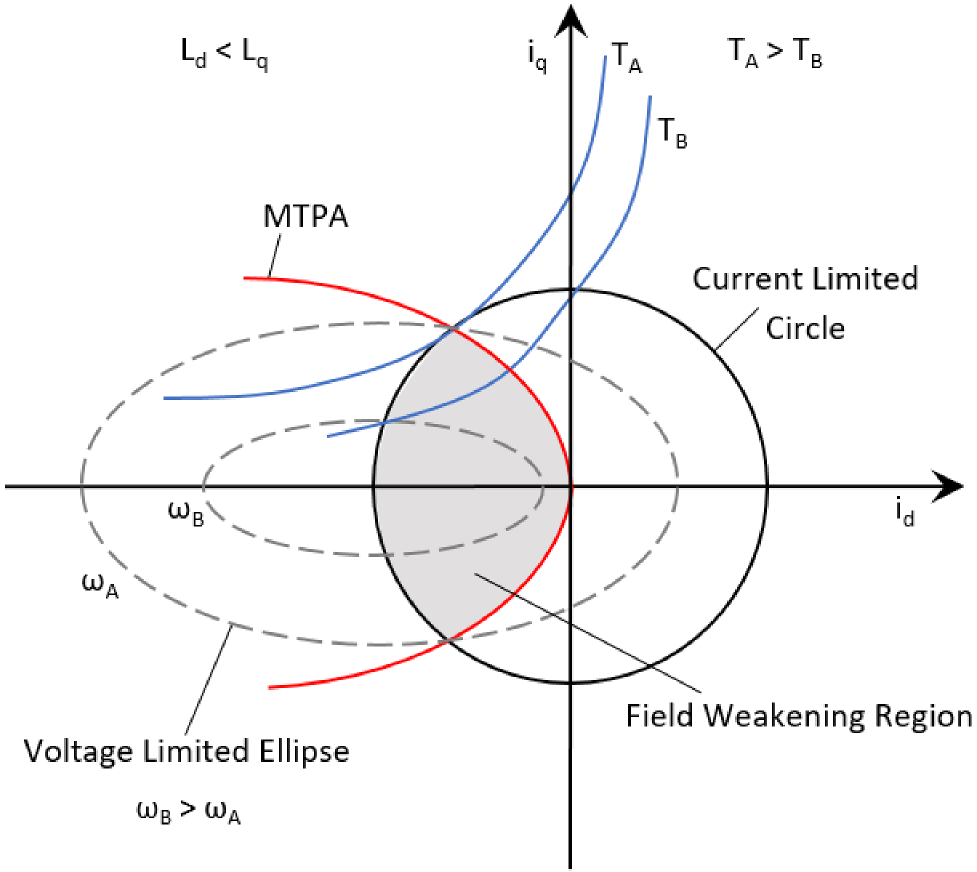


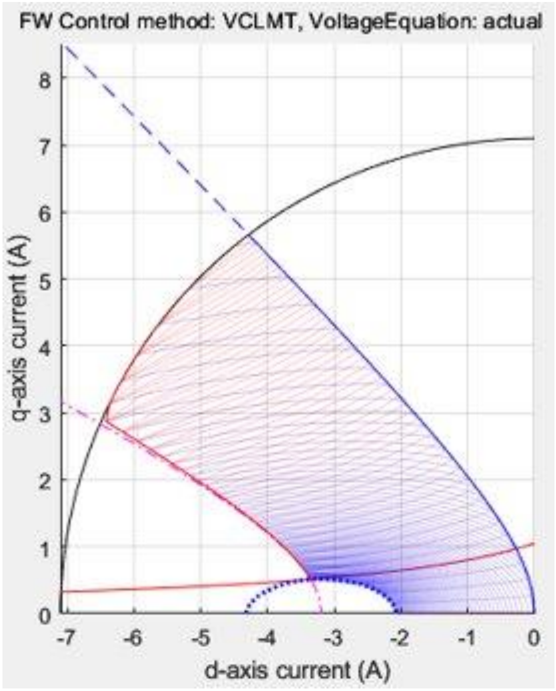
Fig. shows speed ref and measured speed with varying load and speed ref in Dashboard simulation

Id and Iq plot in dq axis shows the current trajectory with increase in speed

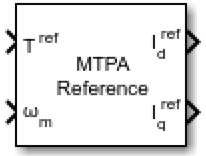
Field-weakening control for Interior PMSM





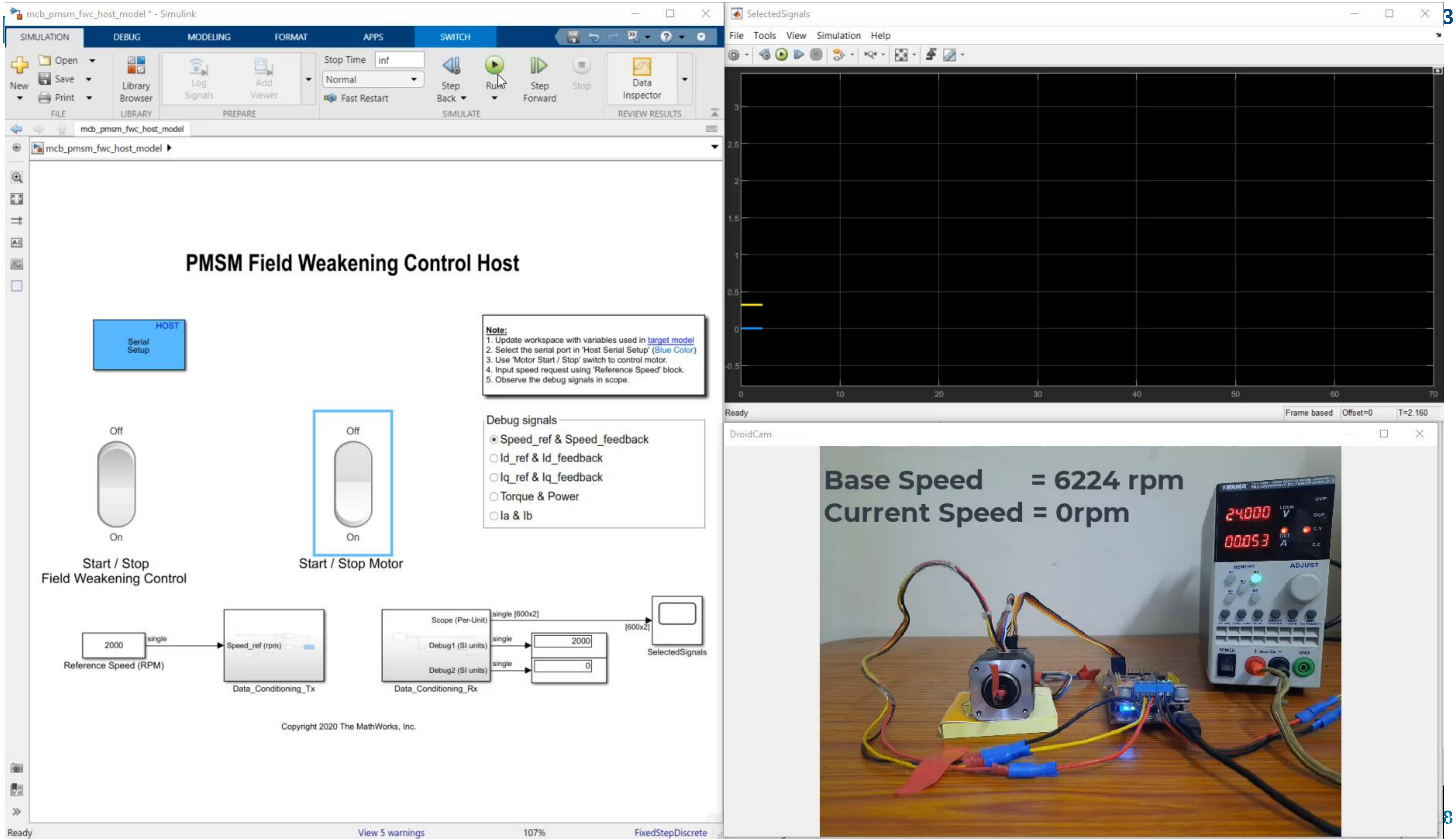


Id and Iq plot in dq axis shows the MTPA path



MTPA Control Reference

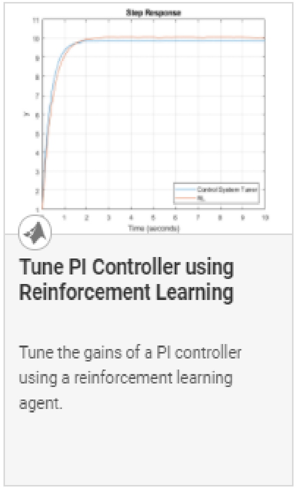
Generates Id_ref and Iq_ref for MTPA, field-weakening control and MTPV


mcb_pmsm_fwc_host_model * - Simulink
SIMULATION
DEBUG
MODELING
FORMAT
APPS
SWITCH
New
Open
Save
Print
FILE
Library Browser
LIBRARY
Log Signals
Add Viewer
PREPARE
Stop Time
inf
Normal
Fast Restart
Step Back
Run
Step Forward
Stop
SIMULATE
Data Inspector
REVIEW RESULTS
mcb_pmsm_fwc_host_model
PMSM Field Weakening Control Host
HOST
Serial Setup
Note:
1. Update workspace with variables used in target model
2. Select the serial port in 'Host Serial Setup' (Blue Color)
3. Use 'Motor Start / Stop' switch to control motor.
4. Input speed request using 'Reference Speed' block.
5. Observe the debug signals in scope.
Off
On
Start / Stop
Field Weakening Control
Start / Stop Motor
Debug signals
Speed_ref & Speed_feedback
Id_ref & Id_feedback
Iq_ref & Iq_feedback
Torque & Power
Ia & Ib
2000
single
Reference Speed (RPM)
Speed_ref (rpm)
Data_Conditioning_Tx
Scope (Per-Unit)
Debug1 (SI units)
Debug2 (SI units)
Data_Conditioning_Rx
single [600x2]
[600x2]
SelectedSignals
0
Copyright 2020 The MathWorks, Inc.
Ready
View 5 warnings
107%
FixedStepDiscrete
SelectedSignals
File
Tools
View
Simulation
Help
Ready
Frame based
Offset=0
T=2.160
DroidCam
Base Speed = 6224 rpm
Current Speed = 0rpm
24.000
00.053
ADJUST

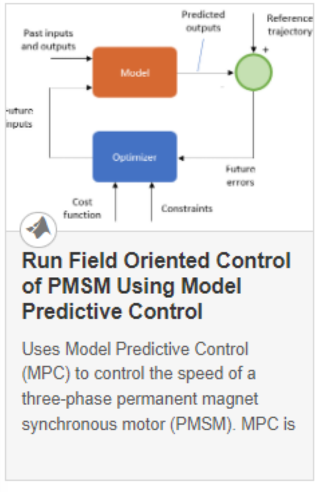
Control Algorithm Implementation

Control algorithm design
Control-loop gain tuning
Simulate & Verify
Motor resizing

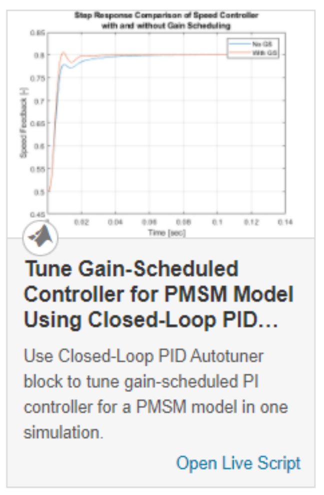
Control algorithm design - Novel control strategies



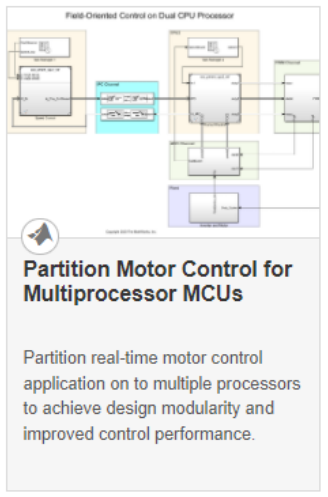
Reinforcement learning for PMSM



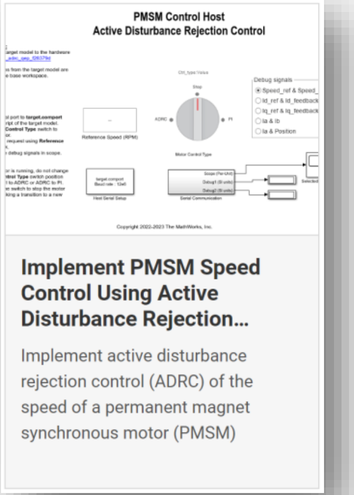
Model predictive control



Gain scheduling example



Motor control simulation for Multiprocessor MCUs

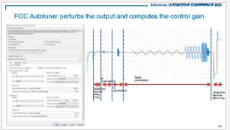


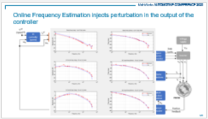
Active Disturbance Rejection Control for PMSM

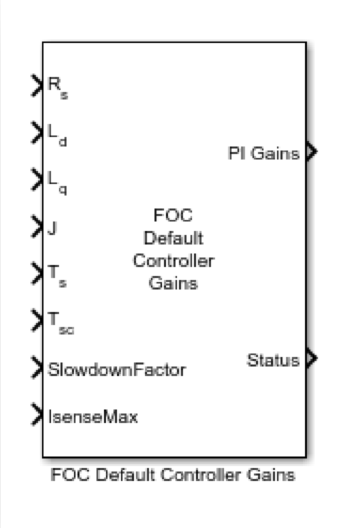
Control Algorithm Implementation

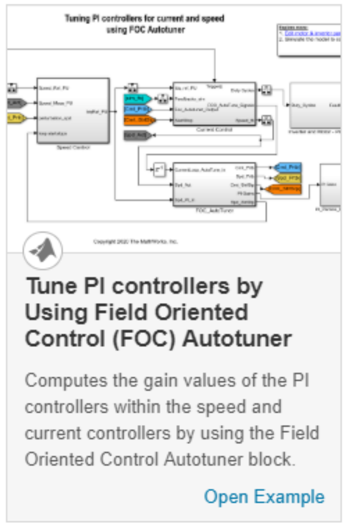
- Control algorithm design
- Control-loop gain tuning
- Simulate & Verify
- Motor resizing

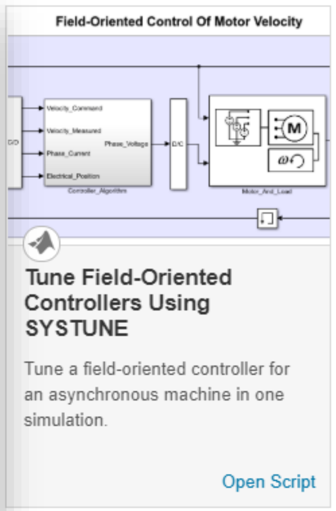
Control-loop gain tuning











Empirical Computation

FOC Autotuner

Online frequency estimation and PID tuner app

Classical Control Theory

Motor Control Blockset

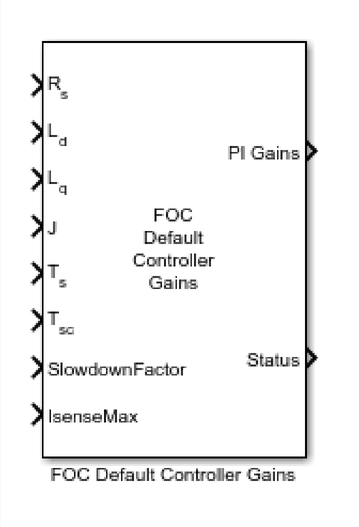
Motor Control Blockset and Simulink Control Design

Simulink Control Design

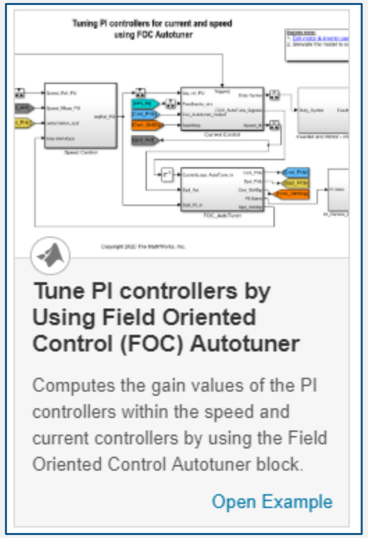
Control Algorithm Implementation

Control-loop gain tuning

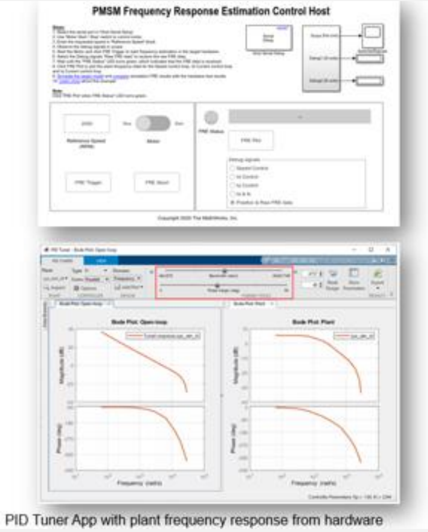
- Control algorithm design
- Control-loop gain tuning
- Simulate & Verify
- Motor resizing



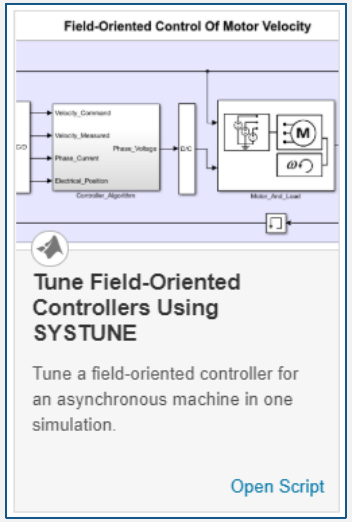
Empirical Computation



FOC Autotuner



Online frequency estimation and PID tuner app



Classical Control Theory

ntrol Blockset

Motor Control Blockset and Simulink Control Design

Simulink Control Design

Control Engineer

How do I tune my Control loop gains to achieve the performance?

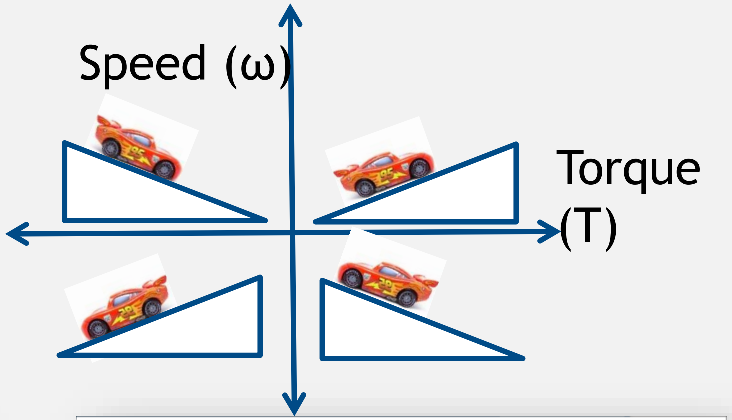
Use any of the methods listed here for tuning the control gains

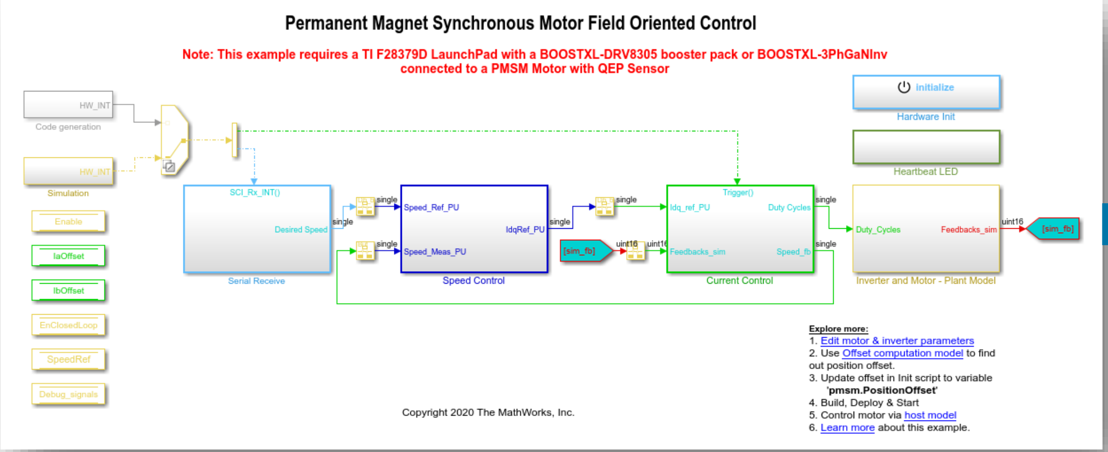
Control Algorithm Implementation

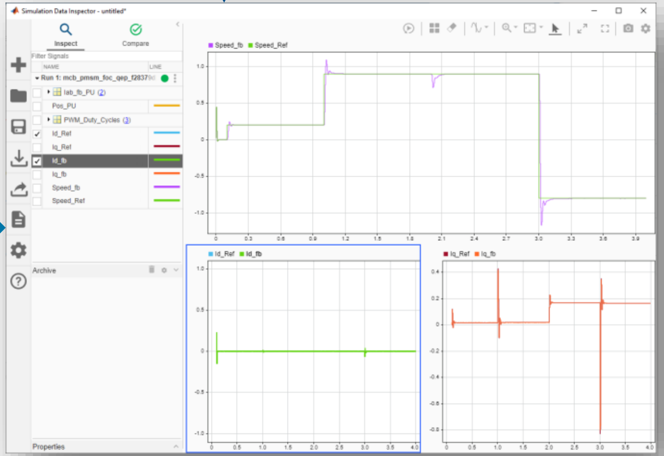
- Control algorithm design
- Control-loop gain tuning
- Simulate & Verify
- Motor resizing

Simulate & Verify

Simulate on 4-quadrant operations and validate the motor and control characteristics.



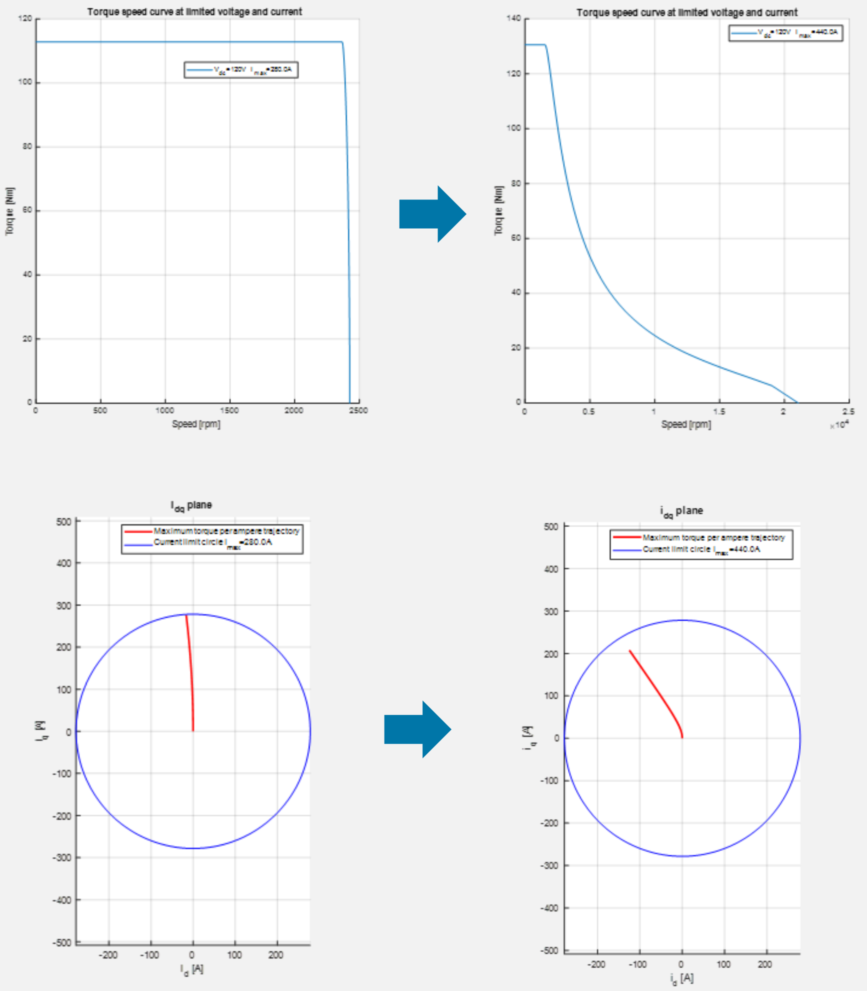




Control Algorithm Implementation

- Control algorithm design
- Control-loop gain tuning
- Simulate & Verify
- Motor resizing

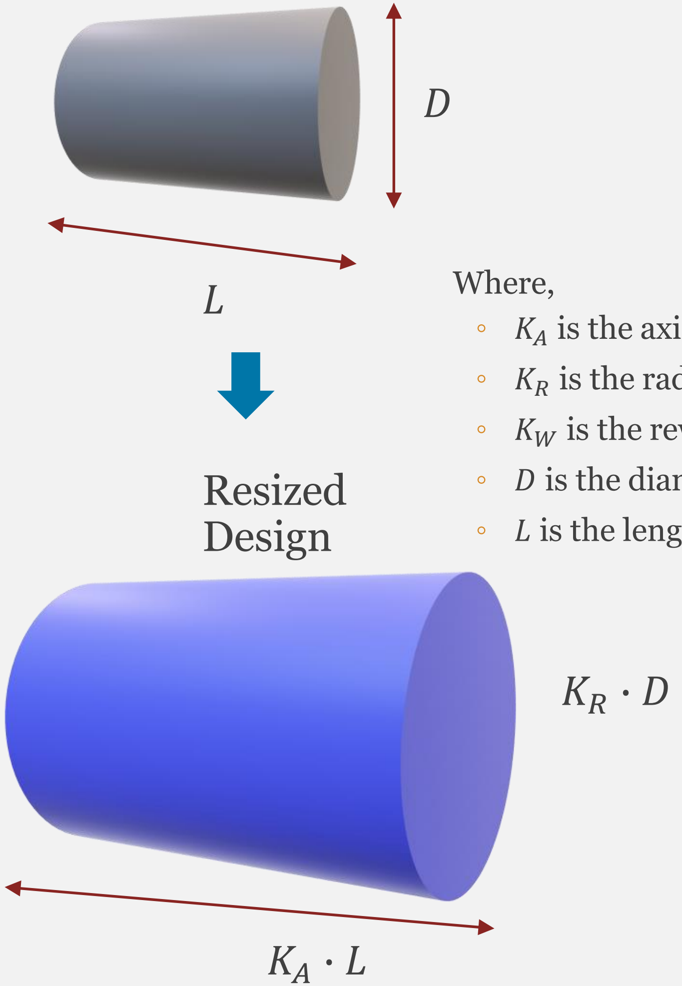
Motor Resizing



Existing characteristics

Proposed characteristics

Initial Design



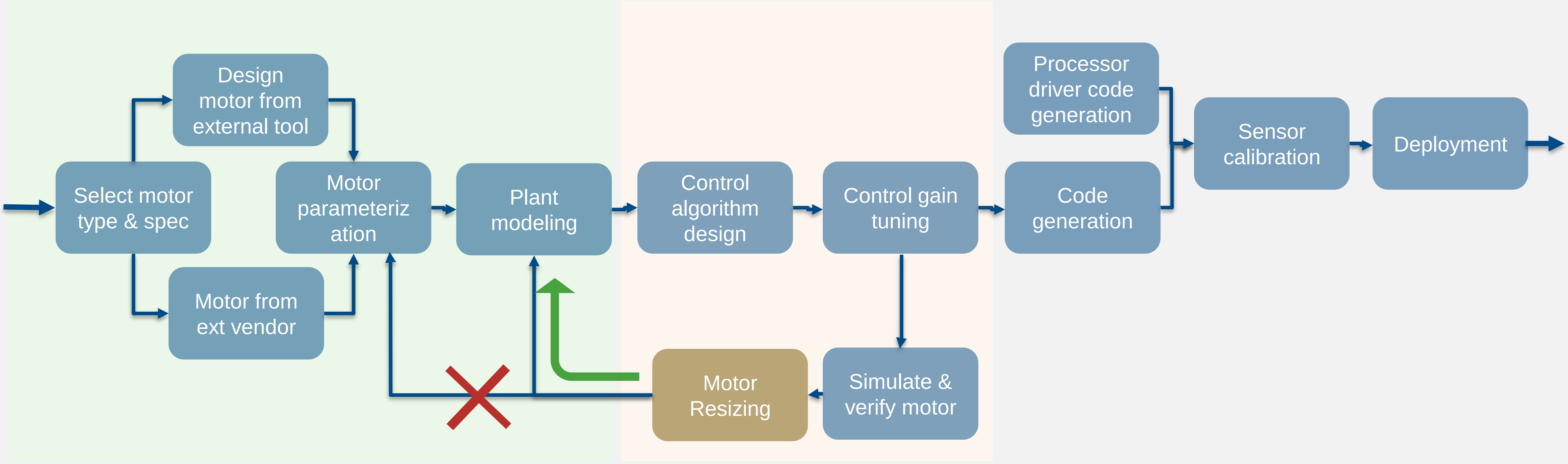
Resized Design

Where,

- K_A is the axial resize factor
- K_R is the radial resize factor
- K_W is the rewinding factor
- D is the diameter of motor
- L is the length of motor

Control Algorithm Implementation

Motor Resizing

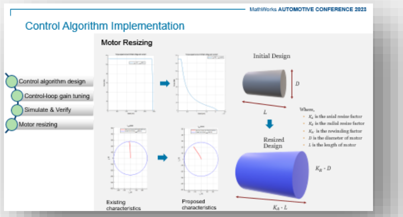


Motor parameterization and Plant modeling

Control algorithm implementation

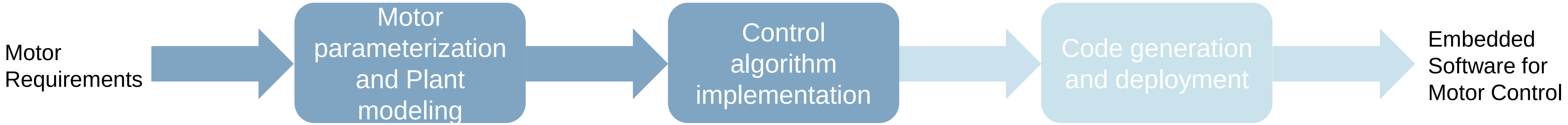
Code gen and deployment

- For resizing motor, No need to run dyno test again.
- Reuse the motor parameters from dyno test with a factor.
- This saves laboratory test time and money.



Motor Resizing

Control Algorithm Implementation

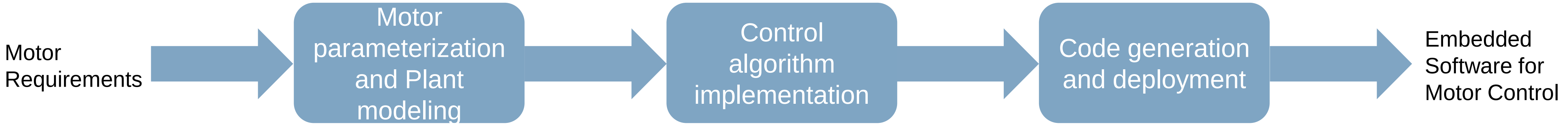


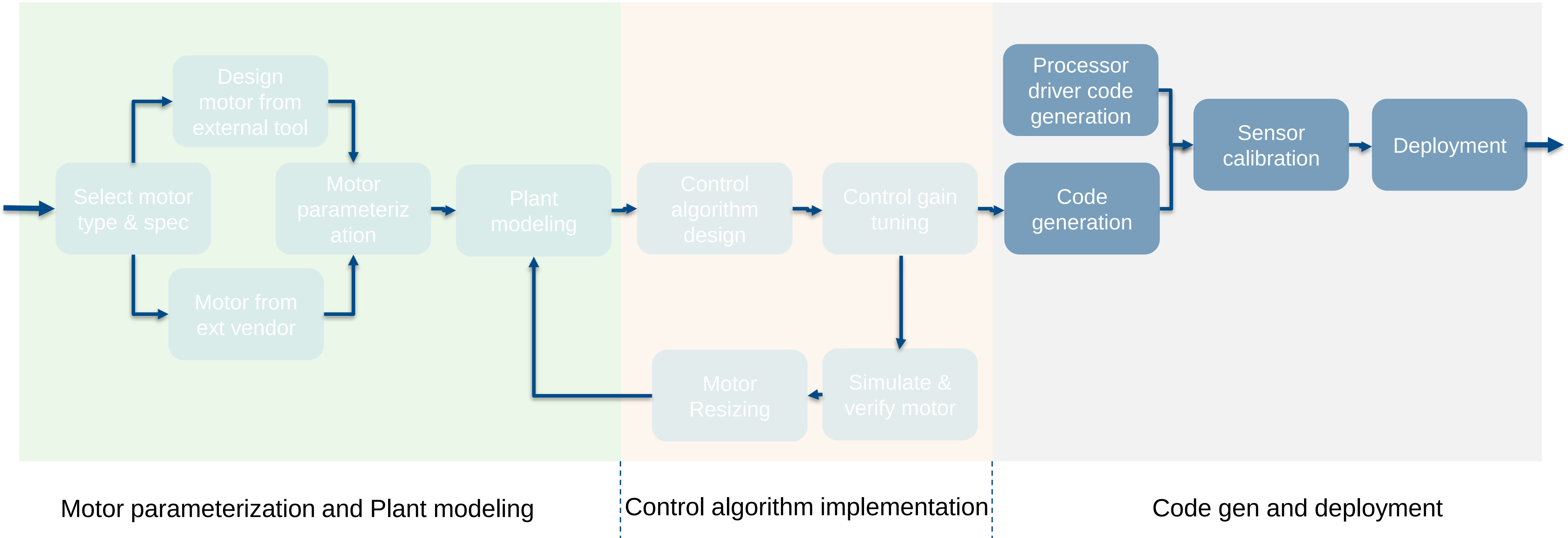
- Control algorithm design
- Control-loop gain tuning
- Simulate & Verify
- Motor resizing

Summary

- Understand the different control strategies and novel control strategies
- Discussed the Field-weakening control (MTPA, MTPV)
- Discussed the control gain tuning strategies
- Discussed the motor resizing

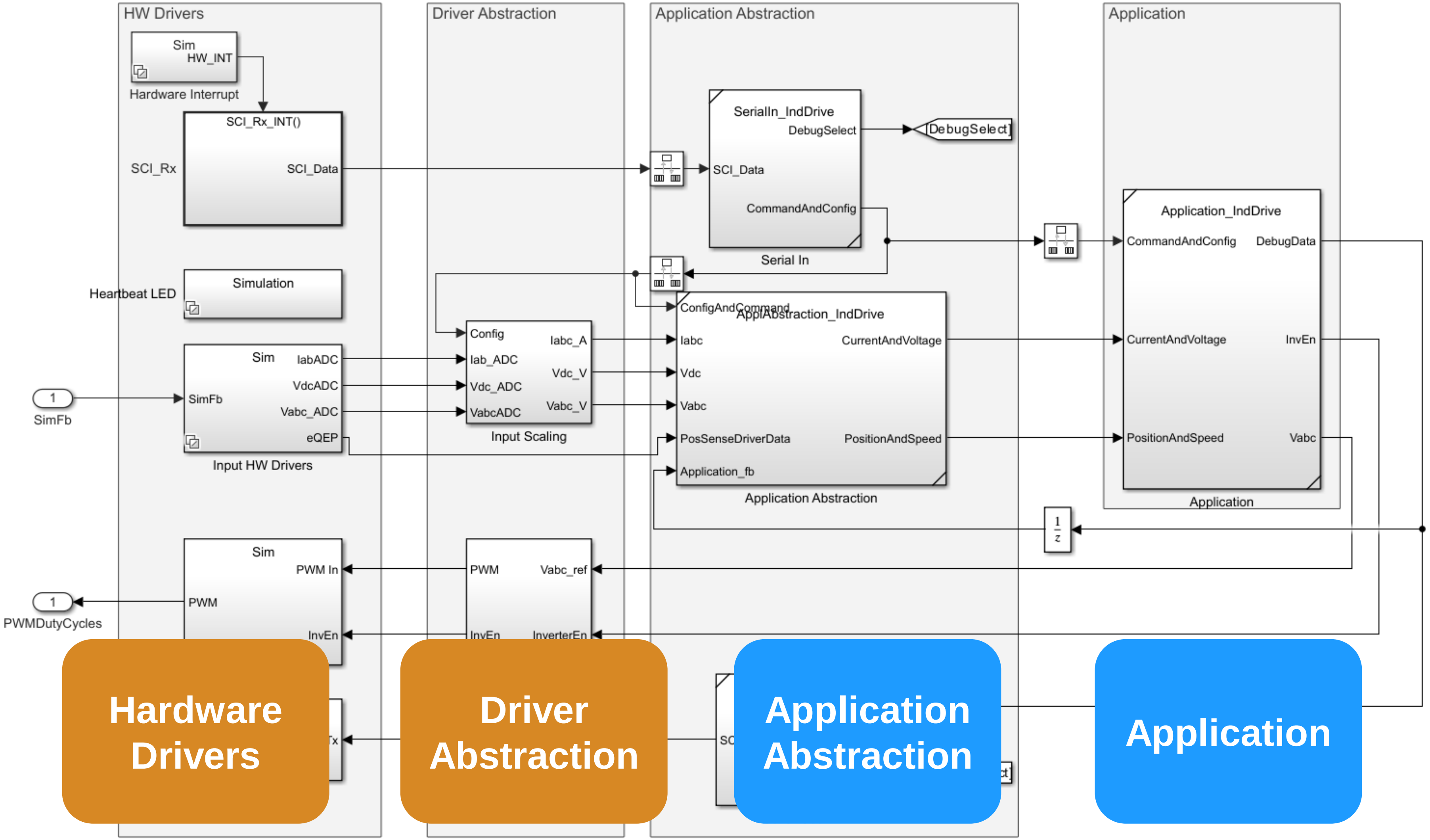
Proposed motor control software development workflow for EV





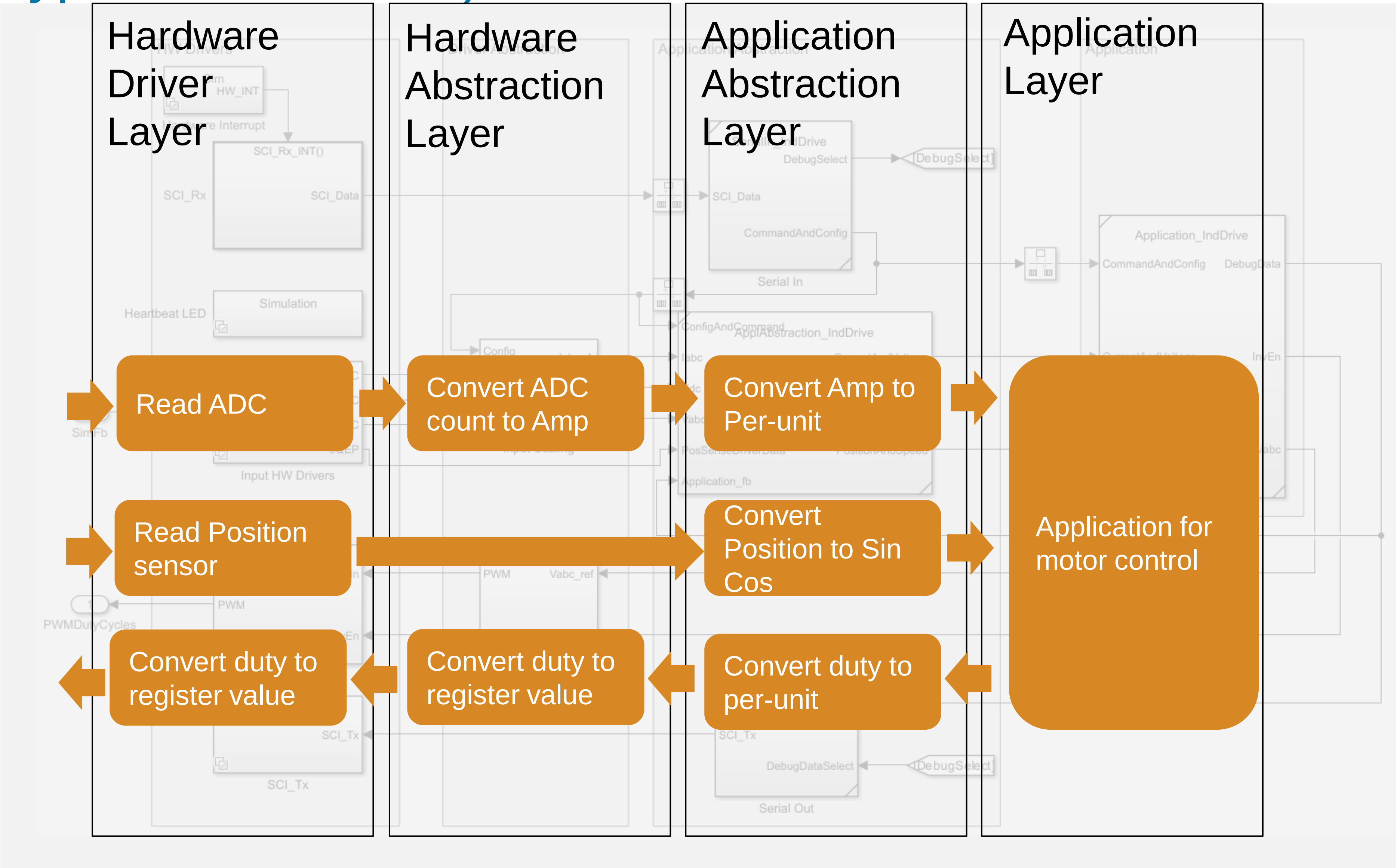
Model architecture with layers to assist porting algorithm between different hardware (Proto-type to Production)

- Model architecture
- Processor driver code gen
- Deployment
- Static code analysis
- Sensor calibration
- Profiling (PIL)
- AUTOSAR Integration



Model architecture to assist porting algorithm between different hardware (Proto-type to Production)

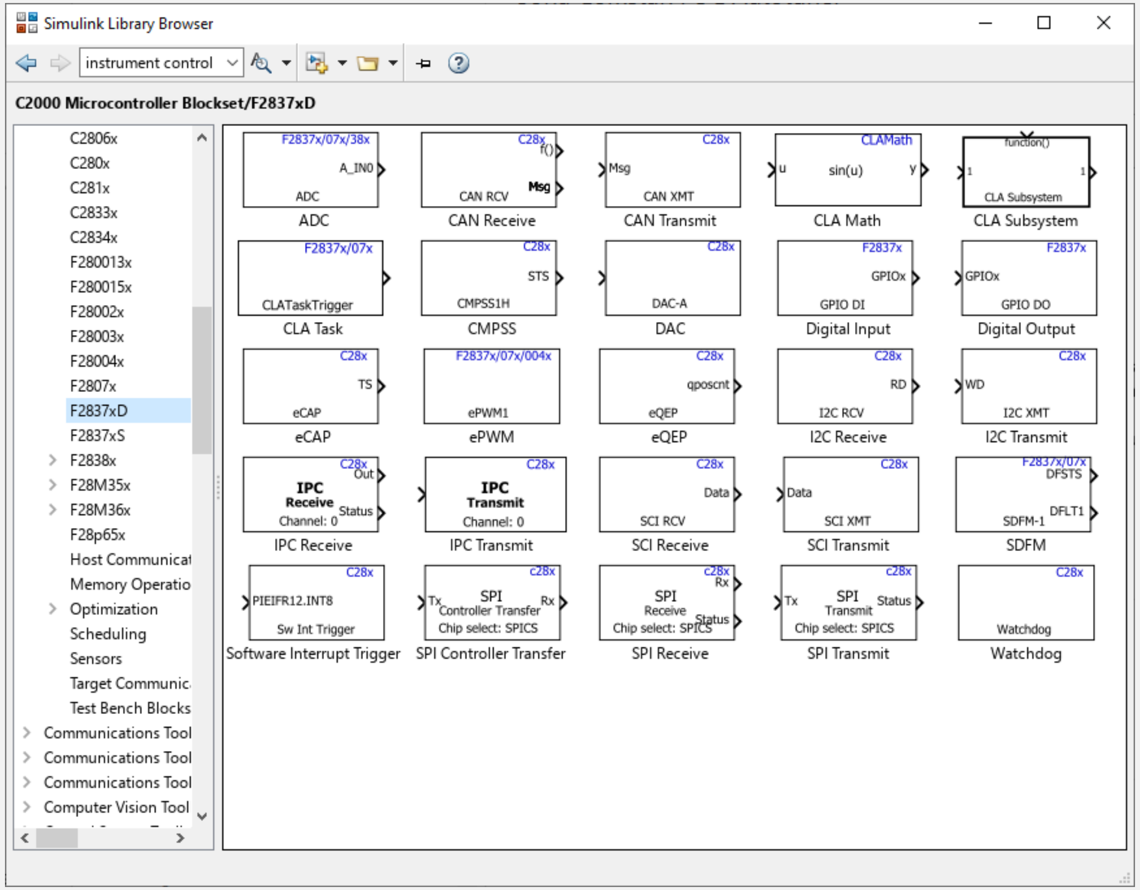
- Model architecture
- Processor driver code gen
- Deployment
- Static code analysis
- Sensor calibration
- Profiling (PIL)
- AUTOSAR Integration

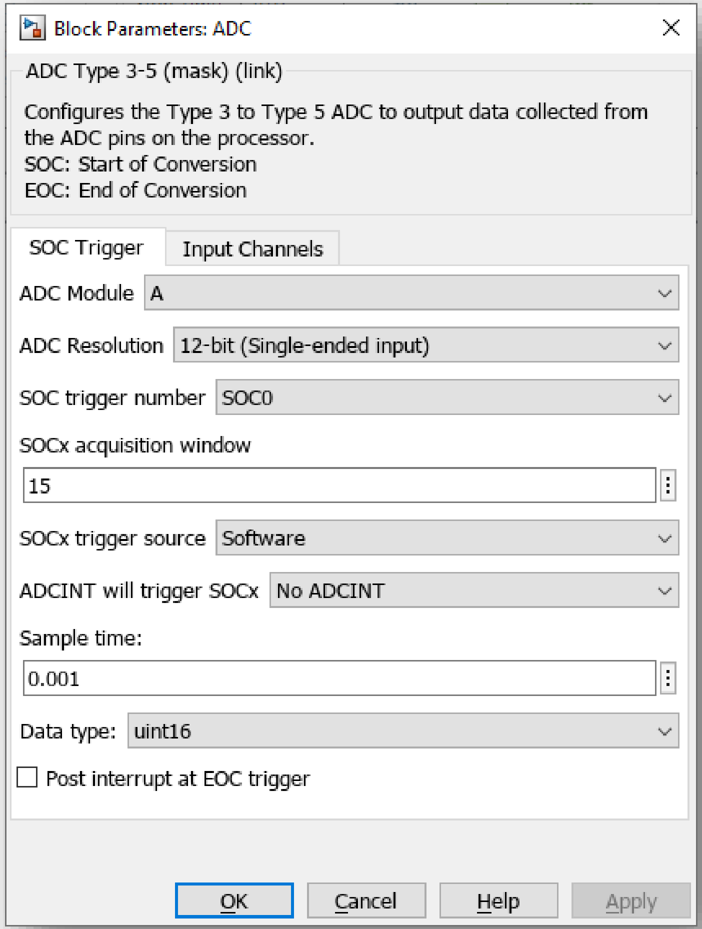


Processor driver code from Hardware support package or Driver block

ADC, PWM, Interrupt trigger, Serial communication blocks

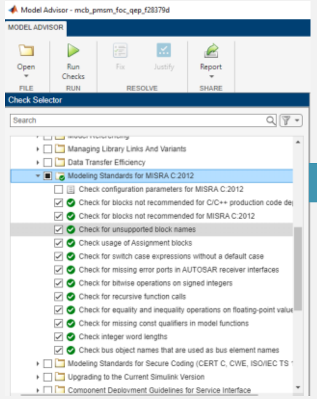
- Model architecture
- **Processor driver code gen**
- Deployment
- Static code analysis
- Sensor calibration
- Profiling (PIL)
- AUTOSAR Integration



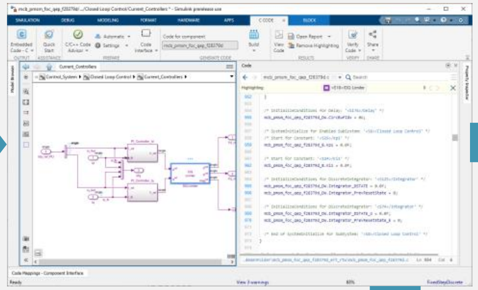


Generate c-code or hdl-code and integrated with peripheral driver code

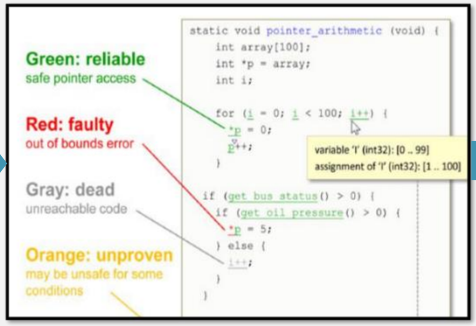
- Model architecture
- Processor driver code gen
- Deployment
- Static code analysis
- Sensor calibration
- Profiling (PIL)
- AUTOSAR Integration



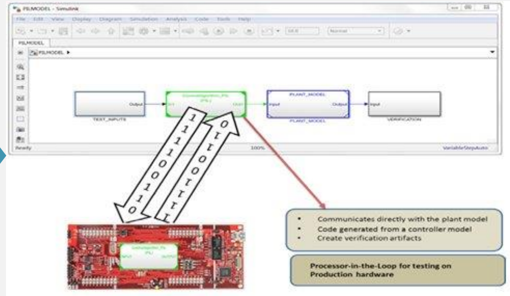
Run model advisor and ensure Misra-C and ISO26262



Generate code using Embedded Coder

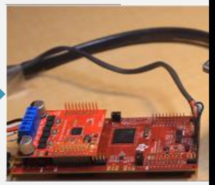


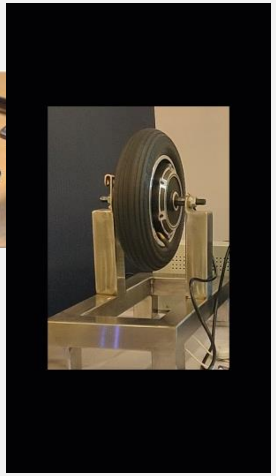
Run static code analysis using Polyspace



Processor-in-loop testing to validate execution time in hardware

Device driver code (Hand-written or Code generated from Hardware support package)





Processor driver code from Hardware support package or Driver block

- Model architecture
- Processor driver code gen
- Deployment
- Static code analysis
- Sensor calibration
- Profiling (PIL)
- AUTOSAR Integration

Processor peripheral driver code generation (ADC, PWM, Interrupt trigger, Serial communication)

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

Generate code optimized for C2000 MCU.

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

TI

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

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

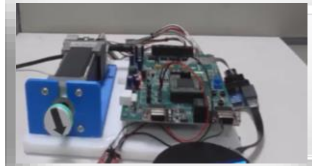
FOC-of-PMSMField-Oriented Control of Permanent Magnet Synchronous Motor Using NX S32K144 Development kitThis example implements a motor control system using the NX MCSPTE1AK144 hardware. The

NXP

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

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

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



Microchip

Embedded Coder Support Package for STMicroelectronics STM32 Processors by MathWorks Embedded Coder Team STAFF

Generate code optimized for STMicroelectronics STM32 Processor boards

STM

Embedded Coder Support Package for Infineon AURIX TC4x Microcontrollers by MathWorks Embedded Coder Team STAFF

Generate code optimized for Infineon AURIX TC4x Microcontrollers

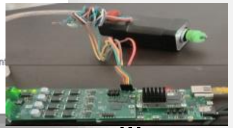
Infineon

HDL Coder Support Package for Xilinx Zynq Platform by MathWorks HDL Coder Team STAFF

Generate code for the FPGA portion of the Zynq-7000 SoC.

HDL Coder™ Support Package for Xilinx®Zynq®-7000 Platform supports the generation of IP cores that can be in FPGA designs using Xilinx Vivado® or Xilinx ISE. When used in combination

Hardware Support



Xilinx FPGA

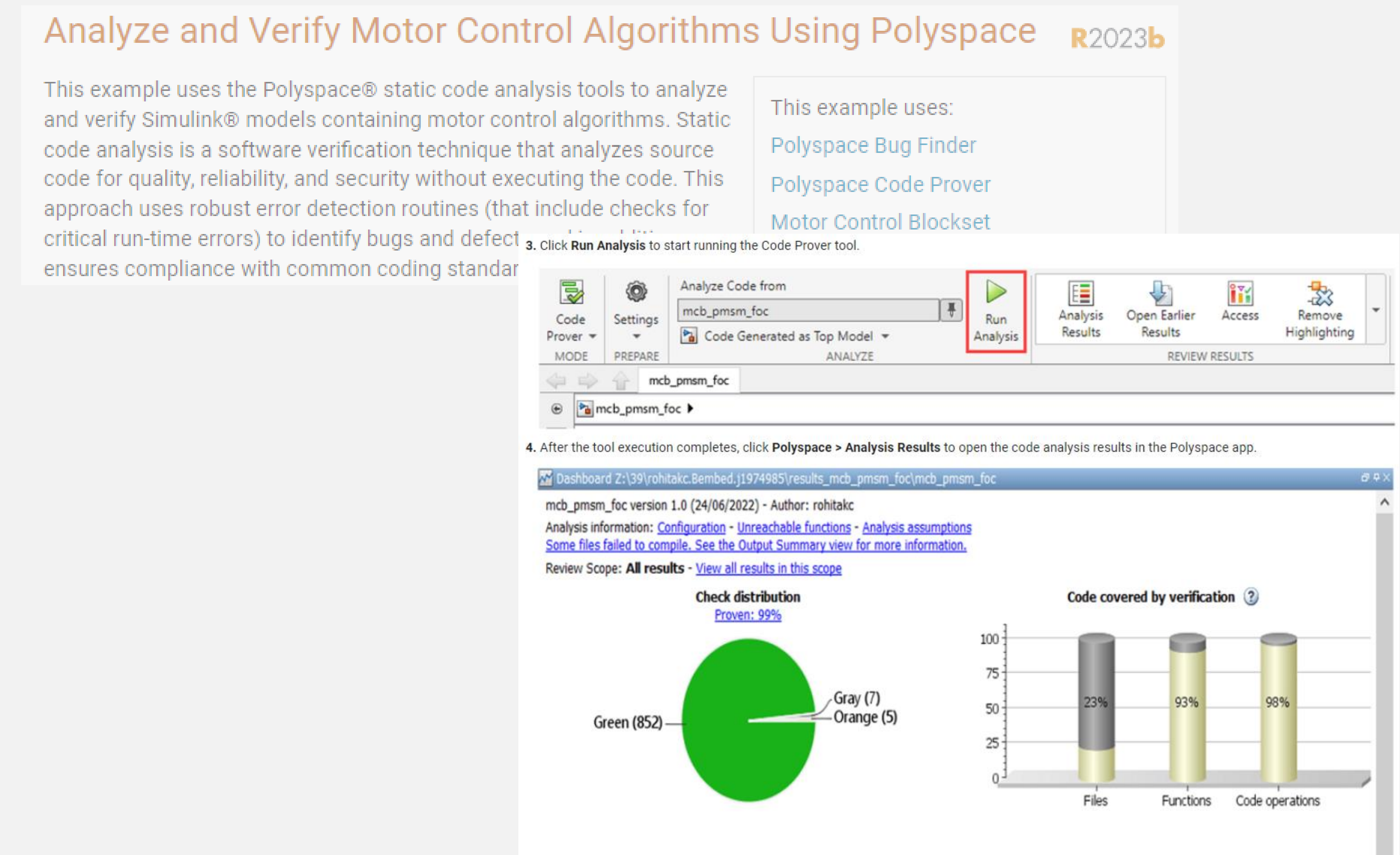
Simulink Real-Time Target Support Package by MathWorks Simulink Real Time Target Team STAFF

Tools to compile a real-time application that runs on a Speedgoat target computer

Speedgoat

Perform static-code analysis tool using Polyspace

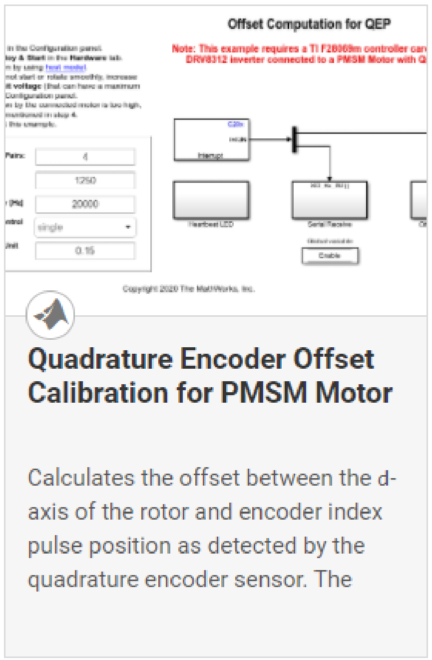
- Model architecture
- Processor driver code gen
- Deployment
- Static code analysis
- Sensor calibration
- Profiling (PIL)
- AUTOSAR Integration



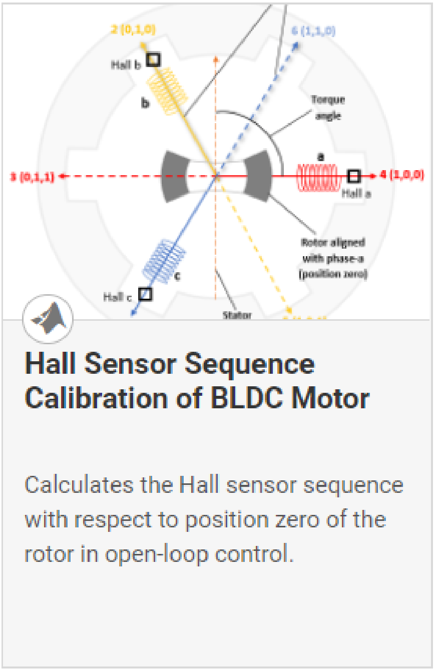
Analyze and Verify Motor Control Algorithms Using Polyspace - MATLAB & Simulink Example - MathWorks India

Calibrate position sensors by spinning the motor in hardware using reference examples

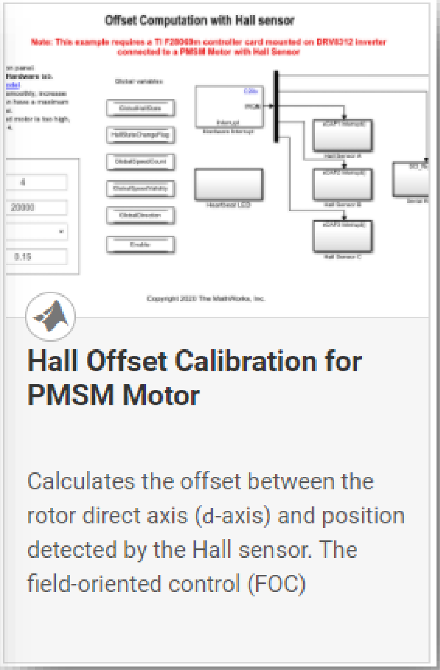
- Model architecture
- Processor driver code gen
- Deployment
- Static code analysis
- Sensor calibration
- Profiling (PIL)
- AUTOSAR Integration



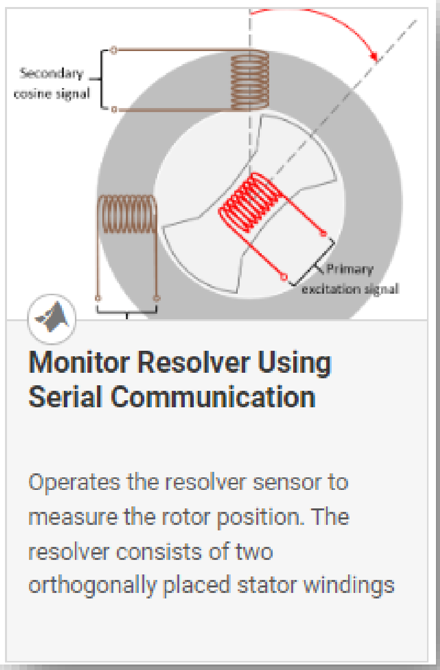
Quadrature encoder calibration



Identify Hall sensor sequence



Calibrate Hall sensor



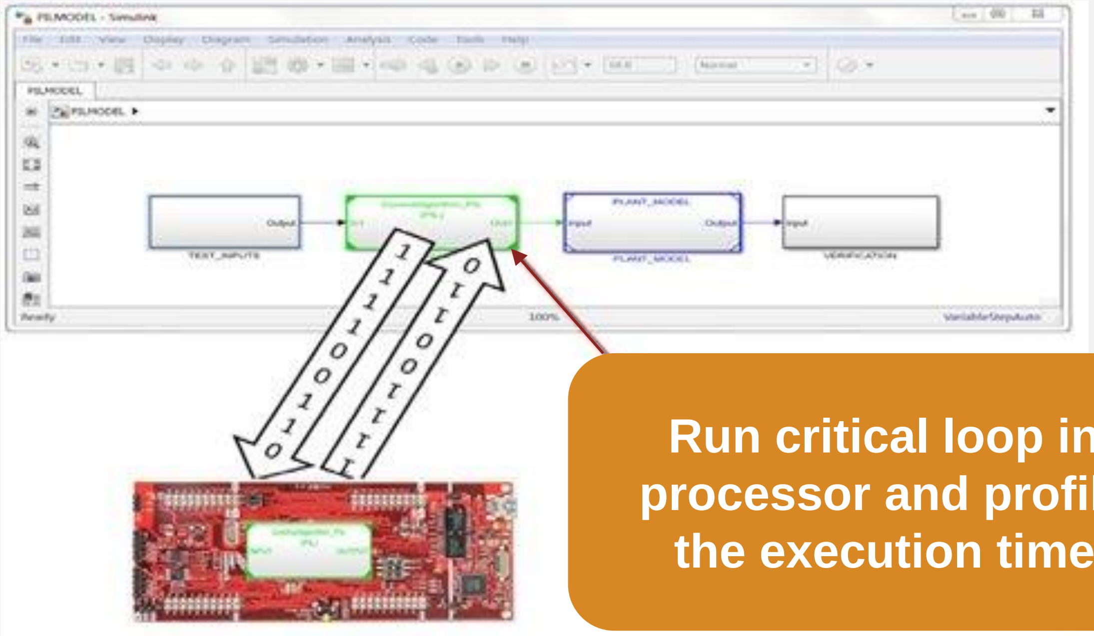
Resolver with Square-pulse carrier frequency

https://in.mathworks.com/help/mcb/gs/quadrature-encoder-offset-calibration-pmsm-motor.html

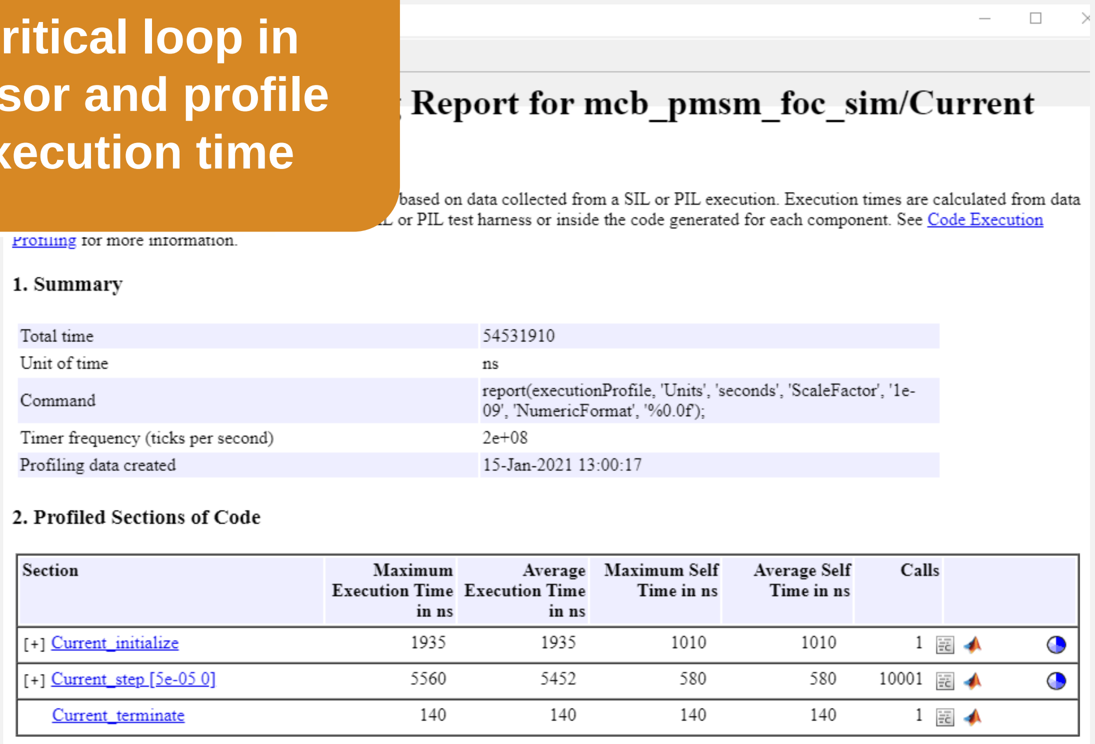
https://in.mathworks.com/help/mcb/gs/hall-sensor-sequence-calibration-bldc-motor.html

Perform profiling test and measure the execution time of critical control loop

- Model architecture
- Processor driver code gen
- Deployment
- Static code analysis
- Sensor calibration
- Profiling (PIL)
- AUTOSAR Integration



Run critical loop in processor and profile the execution time



Report for mcb_pmsm_foc_sim/Current

based on data collected from a SIL or PIL execution. Execution times are calculated from data or PIL test harness or inside the code generated for each component. See Code Execution Profiling for more information.

1. Summary

Total time	54531910
Unit of time	ns
Command	report(executionProfile, 'Units', 'seconds', 'ScaleFactor', '1e-09', 'NumericFormat', '%0.0f');
Timer frequency (ticks per second)	2e+08
Profiling data created	15-Jan-2021 13:00:17

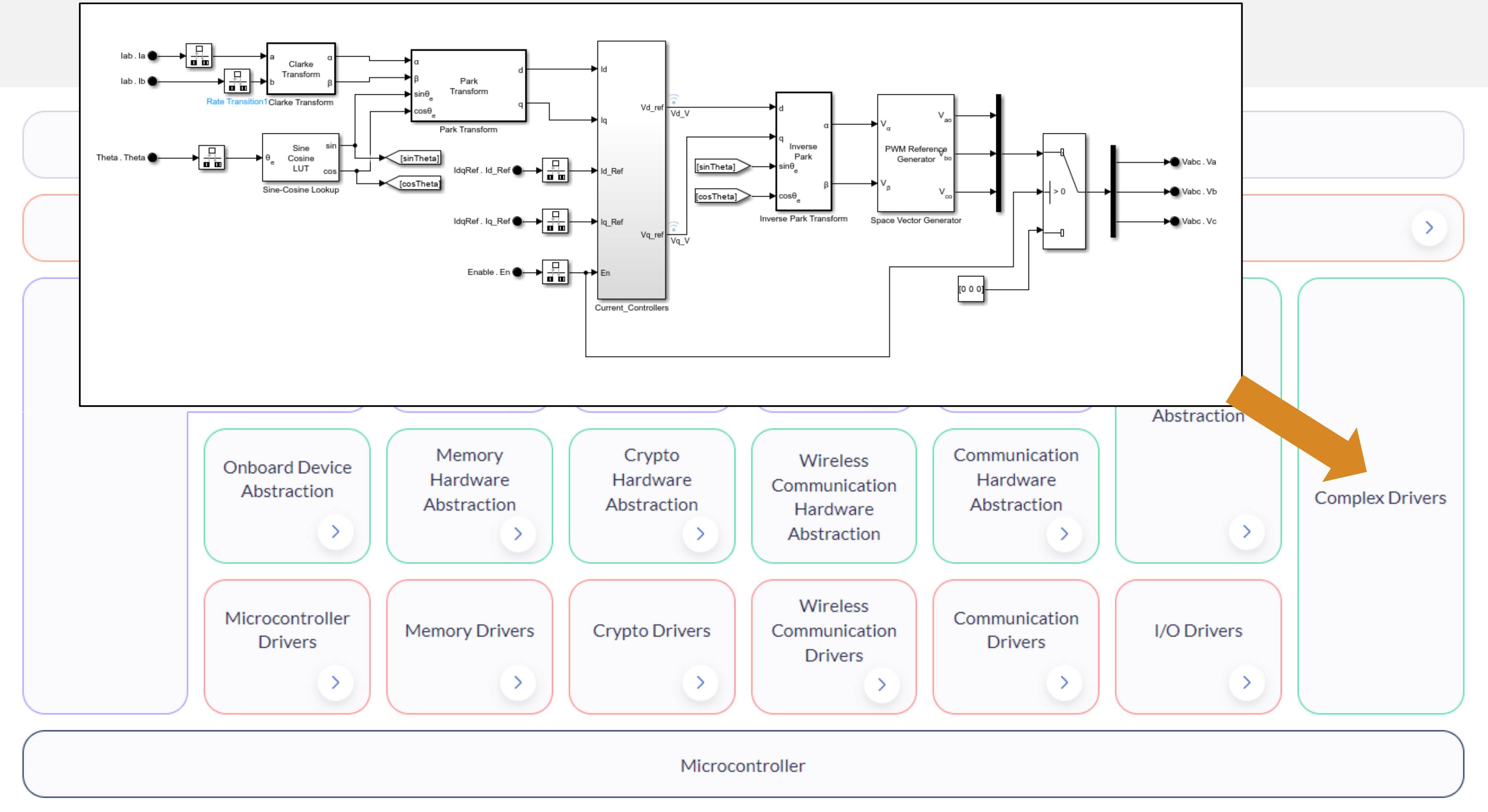
2. Profiled Sections of Code

Section	Maximum Execution Time in ns	Average Execution Time in ns	Maximum Self Time in ns	Average Self Time in ns	Calls
[+] Current_initialize	1935	1935	1010	1010	1
[+] Current_step [5e-05 0]	5560	5452	580	580	10001
Current_terminate	140	140	140	140	1

Code Verification and Profiling Using PIL Testing - MATLAB & Simulink Example - MathWorks India

Integrate motor control loop in AUTOSAR architecture

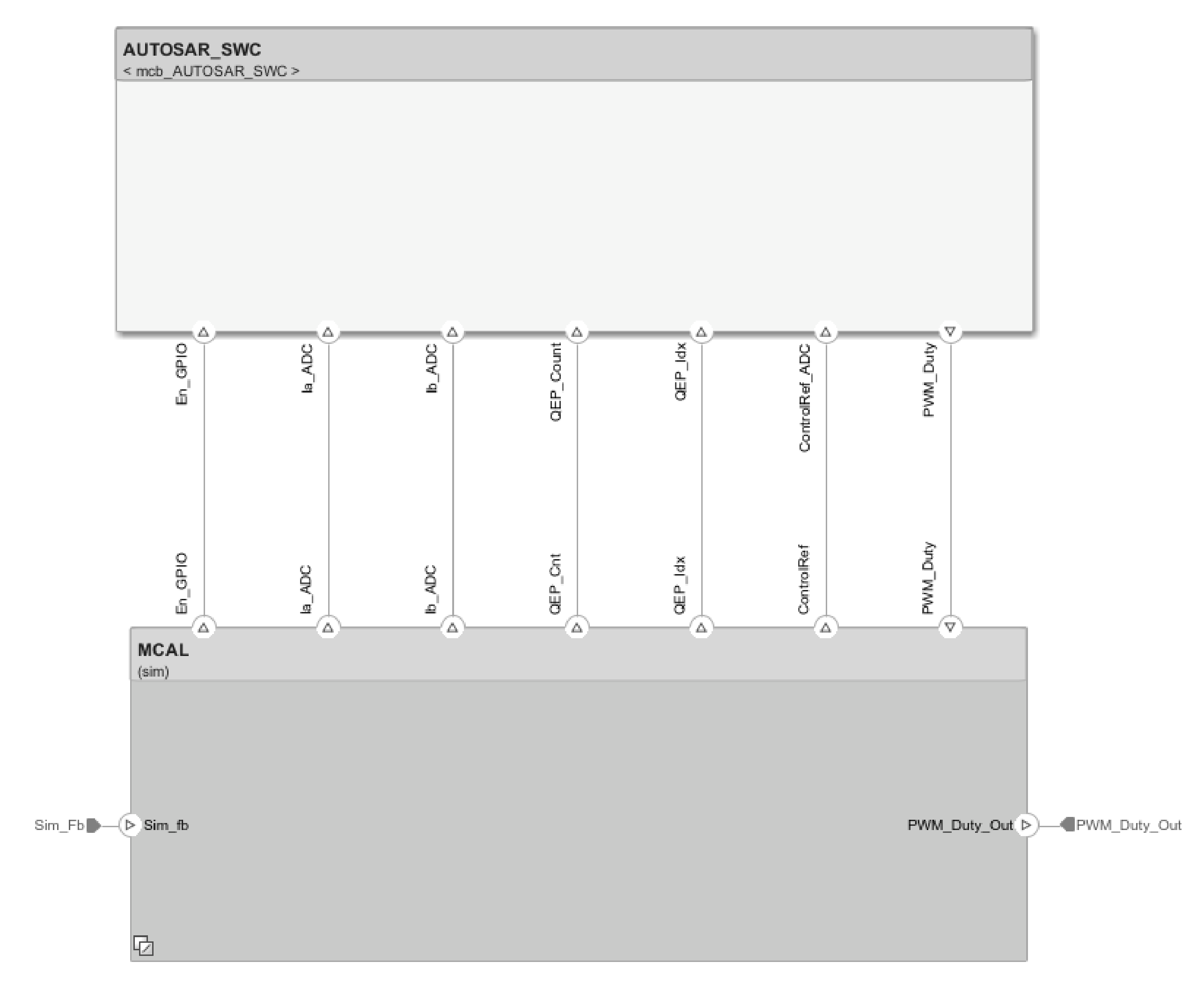
- Model architecture
- Processor driver code gen
- Deployment
- Static code analysis
- Sensor calibration
- Profiling (PIL)
- AUTOSAR Integration



Reference: https://www.autosar.org/standards/classic-platform

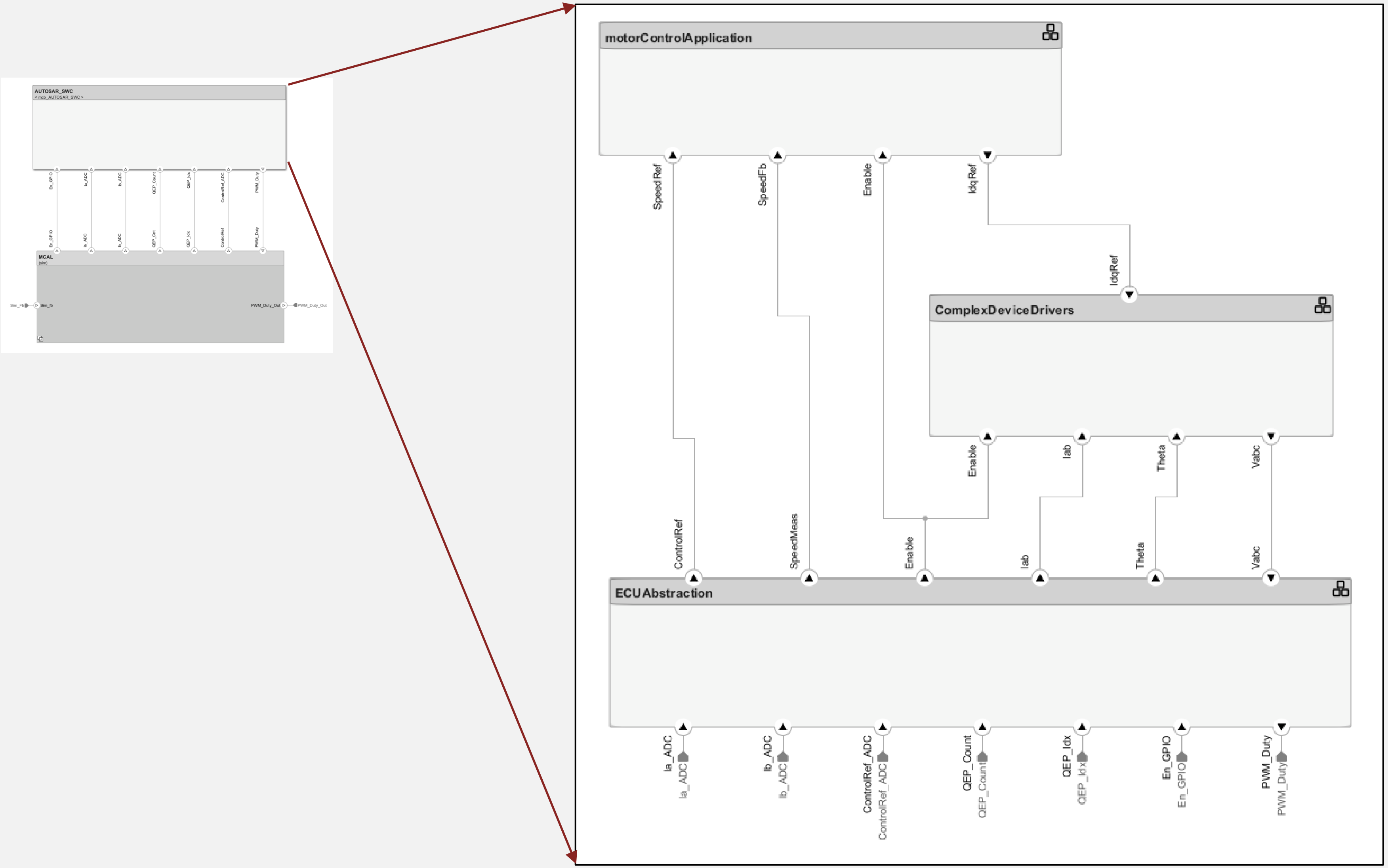
Integrate motor control loop in AUTOSAR architecture

- Model architecture
- Processor driver code gen
- Deployment
- Static code analysis
- Sensor calibration
- Profiling (PIL)
- AUTOSAR Integration



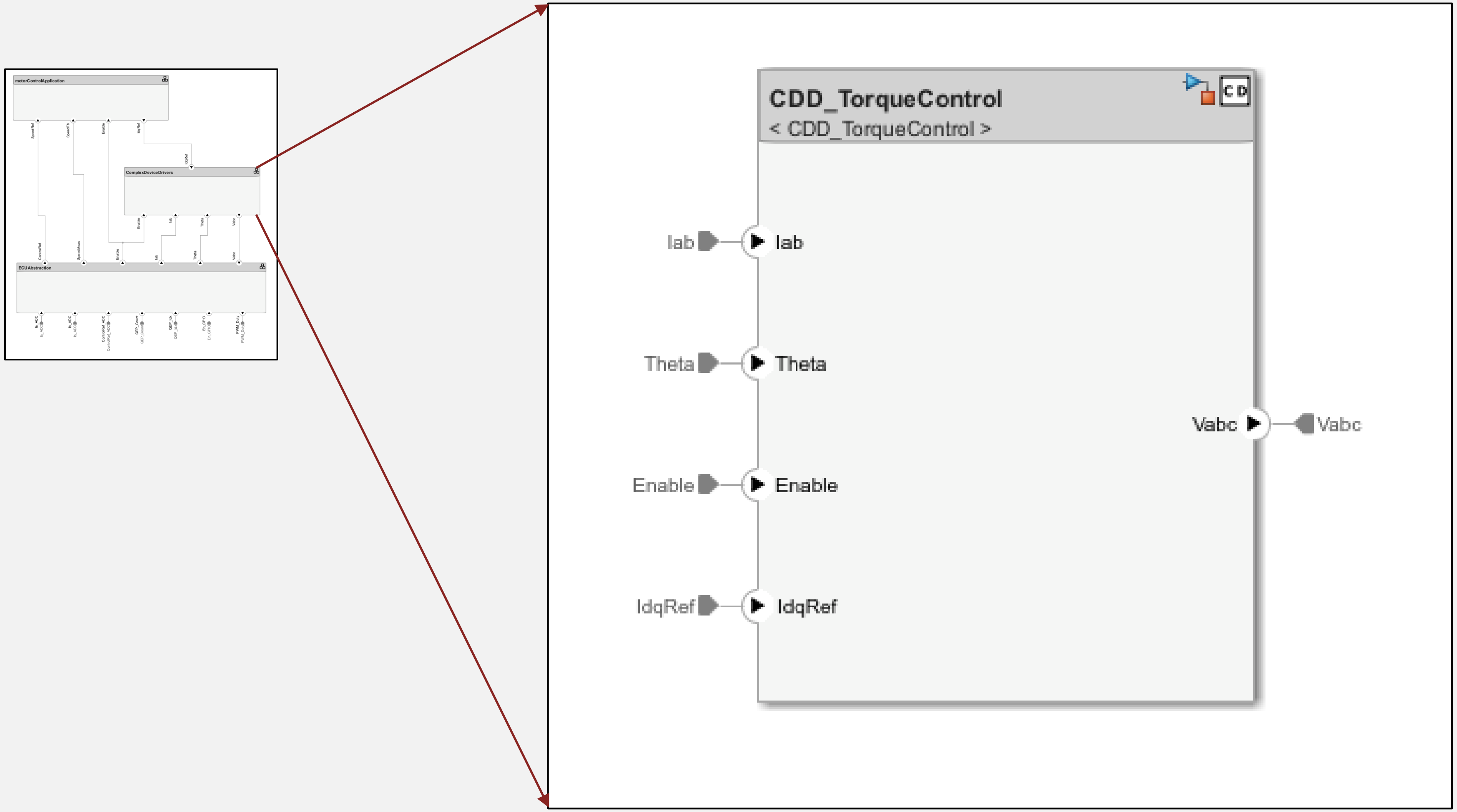
Integrate motor control loop in AUTOSAR architecture

- Model architecture
- Processor driver code gen
- Deployment
- Static code analysis
- Sensor calibration
- Profiling (PIL)
- AUTOSAR Integration



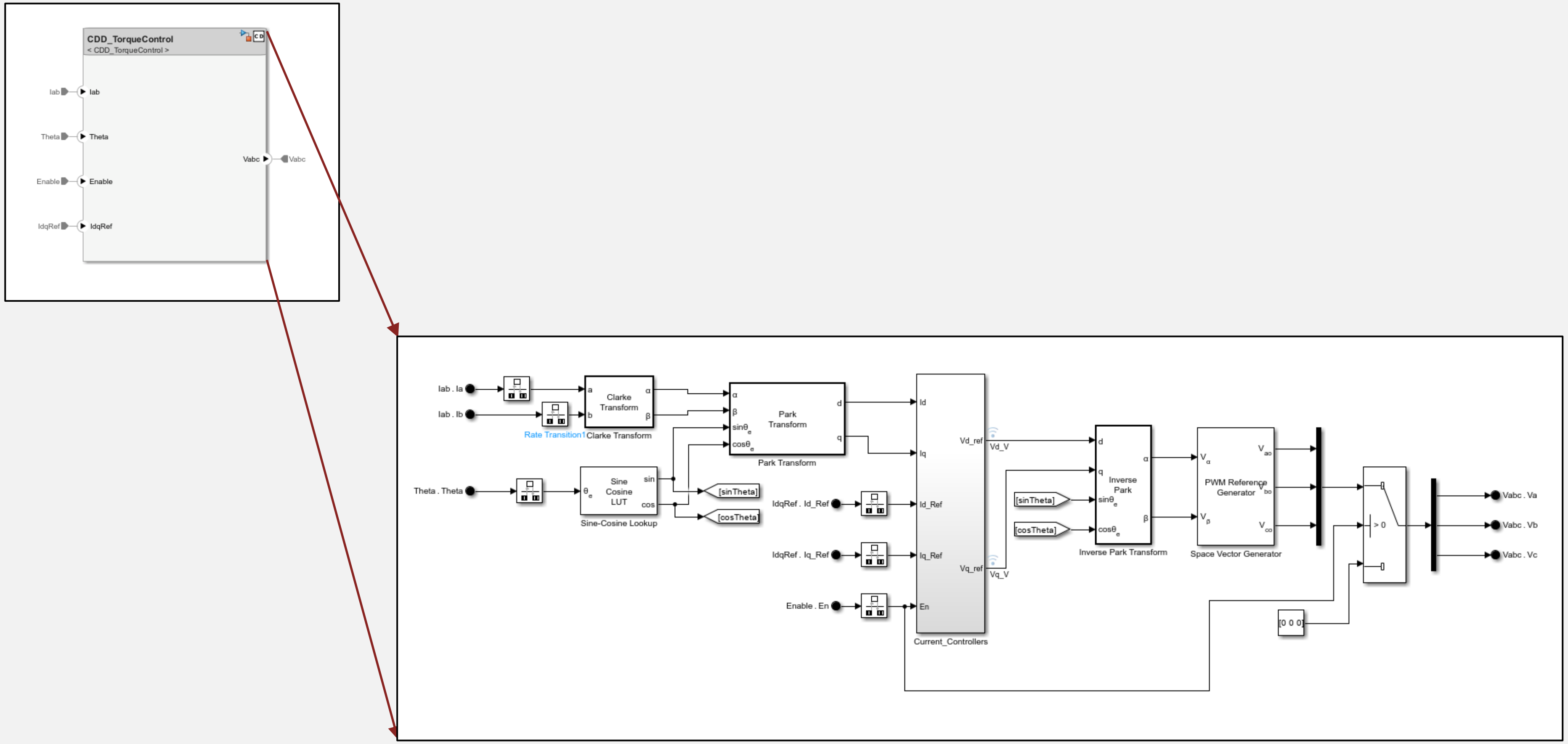
Integrate motor control loop in AUTOSAR architecture

- Model architecture
- Processor driver code gen
- Deployment
- Static code analysis
- Sensor calibration
- Profiling (PIL)
- AUTOSAR Integration



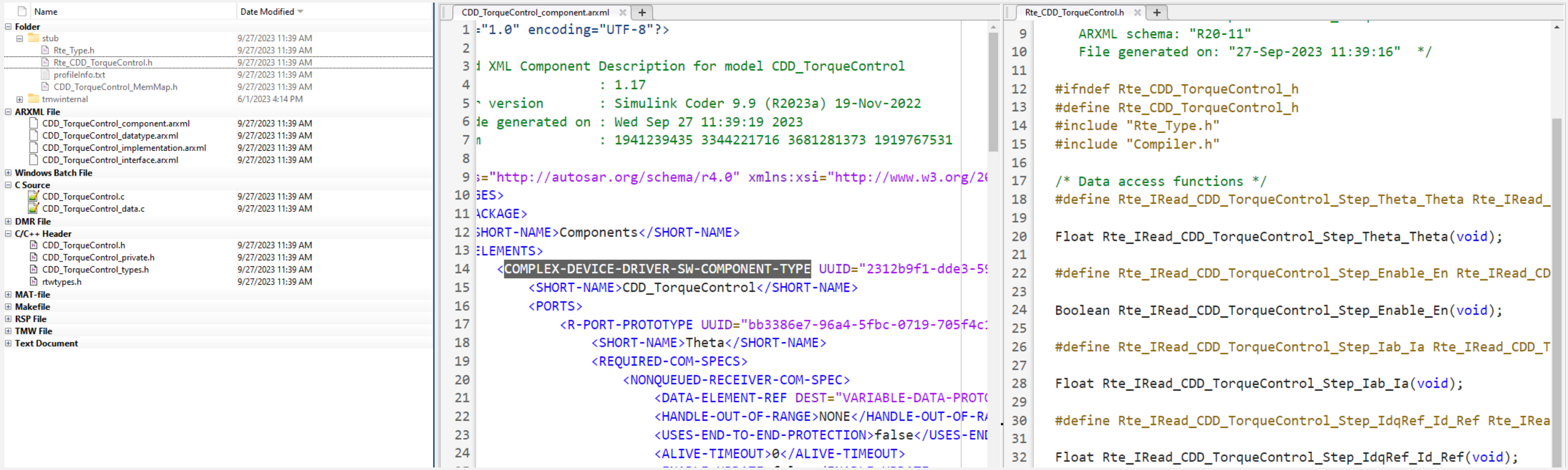
Integrate motor control loop in AUTOSAR architecture

- Model architecture
- Processor driver code gen
- Deployment
- Static code analysis
- Sensor calibration
- Profiling (PIL)
- AUTOSAR Integration

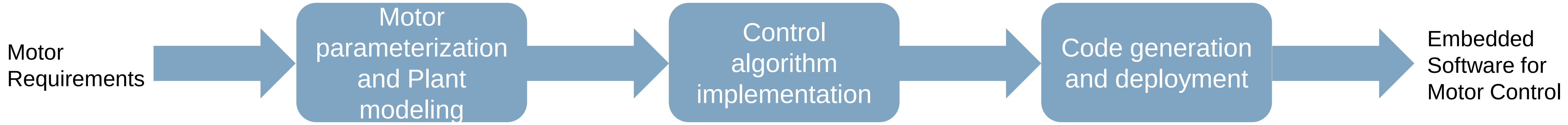


Generate AUTOSAR compliant code and ARXML file for motor control component

- Model architecture
- Processor driver code gen
- Deployment
- Static code analysis
- Sensor calibration
- Profiling (PIL)
- AUTOSAR Integration



Summary



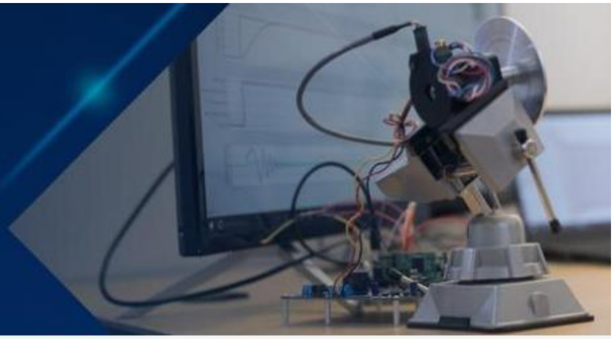
- ✓ Discussed on the motor control workflow and reference examples available for each of the steps
- ✓ Different methods to parameterize the motor and different fidelity levels in modeling
- ✓ Different control strategies and its control loop gain tuning
- ✓ Deploy to the hardware and Validate

Learn More

MATLAB and Simulink for Motor Drives and Traction Motors

Develop algorithms and embedded software for motor-inverter control systems

Free trial



How It Works

Design motor control algorithms

Test motor control algorithms

Implement motor control algorithms on hardware

Simulate Motor Control Algorithms

Use MATLAB® and Simulink® to build motor models from libraries of motors, inverters, sources, and loads. Choose the level of fidelity in motor and inverter modeling based on your requirements and simulate motor control algorithms.

- Implement linear lumped-parameter motor models and use average value inverters with Motor Control Blockset™ for fast simulations
- Model and simulate nonlinear motor dynamics and ideal or detailed switching in the inverter using Power Systems Simulation Onramp
- Parametrize motor models to capture motor dynamics with the help of instrumented tests or import parameters from a database or finite element analysis

Motor Control Design with MATLAB and Simulink

MathWorks Videos

Suggest a video

Understanding Field-Oriented Control | Motor Control, Part 4

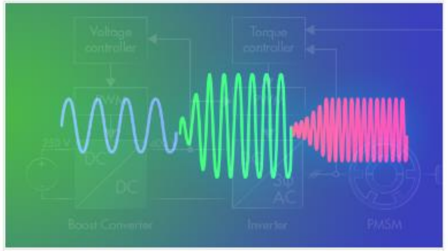
Reinforcement Learning for Developing Field-Oriented Control

Motor Control, Part 3: BLDC Speed Control Using PWM

» View all MathWorks videos

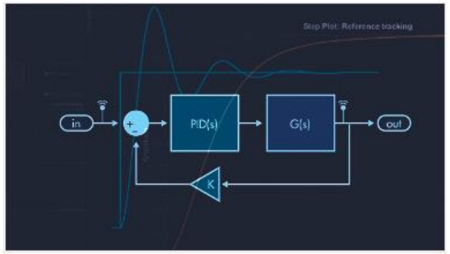
https://www.mathworks.com/solutions/electrification/motor-drives-traction-motors.html

Enable Your Team on Motor Control



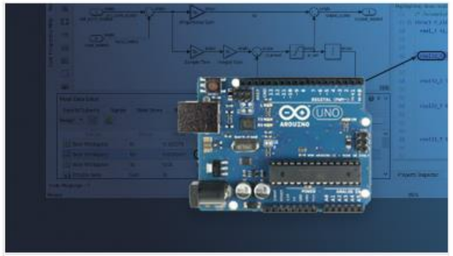
Power Electronics Control Design with Simulink and Simscape

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



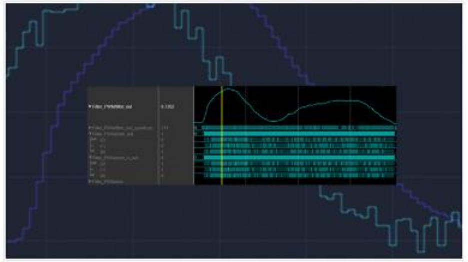
Control System Design with MATLAB and Simulink

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



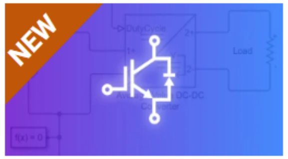
Embedded Coder for Production Code Generation

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



Generating HDL Code from Simulink

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



Power Electronics Simulation Onramp

5 modules | 1 hour | Languages

Learn the basics of simulating power electronics converters in Simscape.

Circuit Simulation Onramp

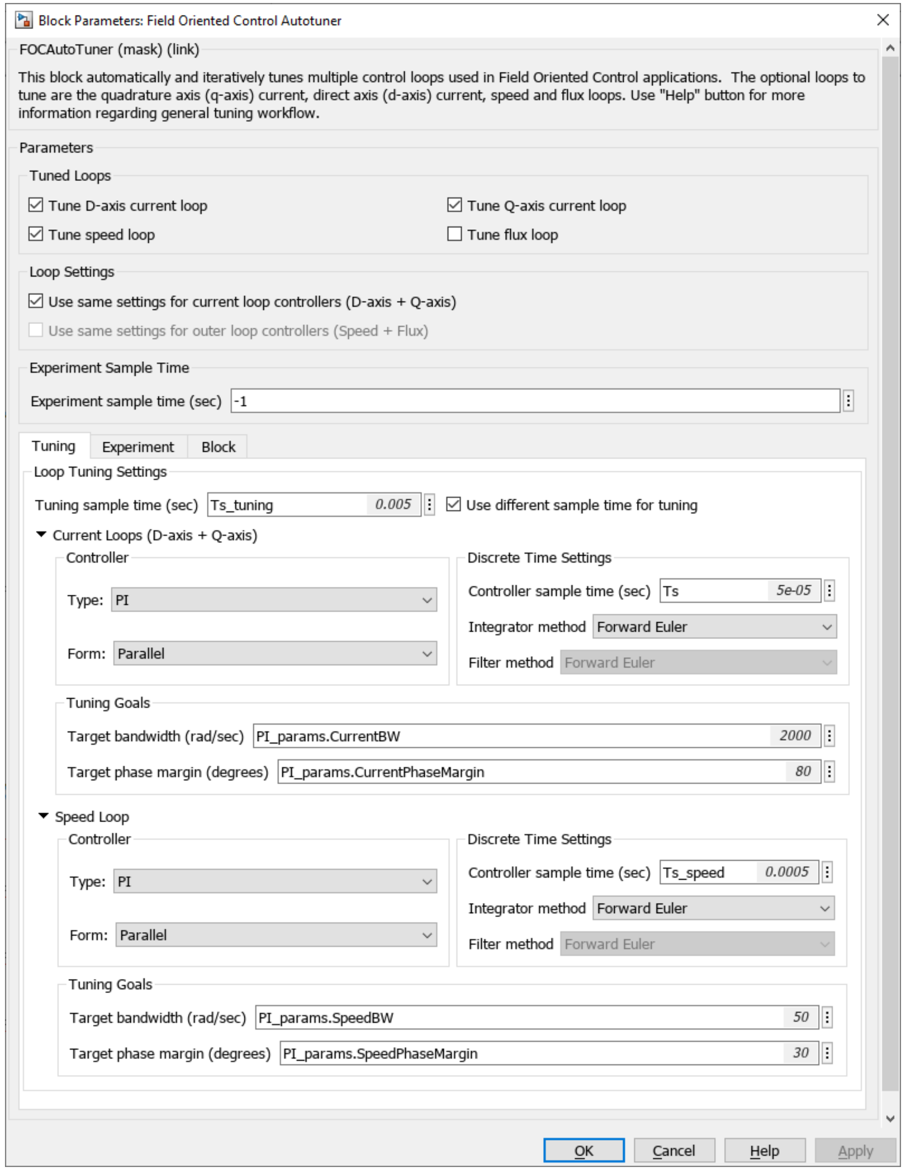
7 modules | 2 hours | Languages

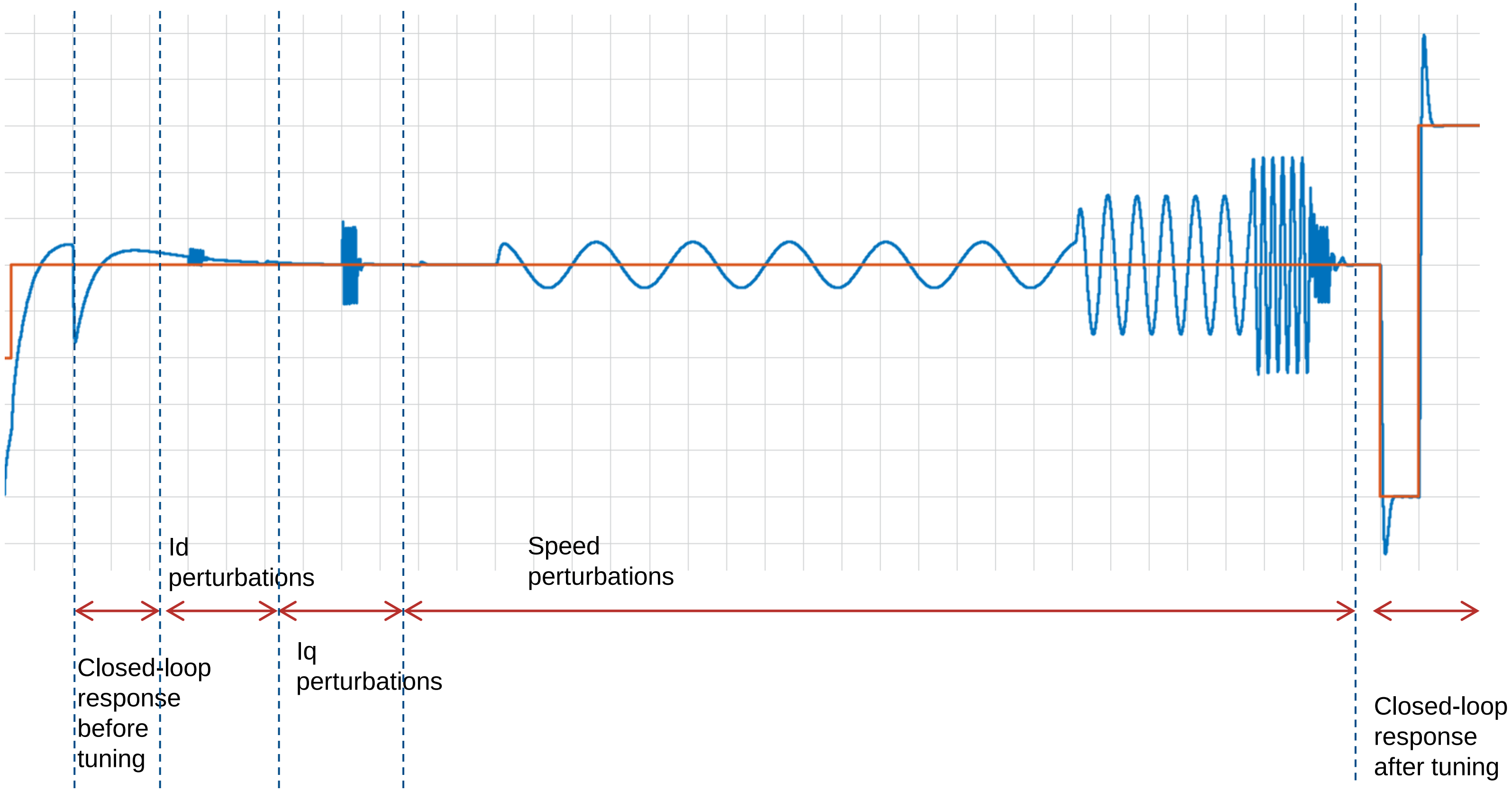
Learn the basics of simulating electrical circuits in Simscape.

MathWorks
AUTOMOTIVE CONFERENCE 2023
India

Thank you

FOC Autotuner perturbs the output and computes the control gain





Online Frequency Estimation injects perturbation in the output of the controller