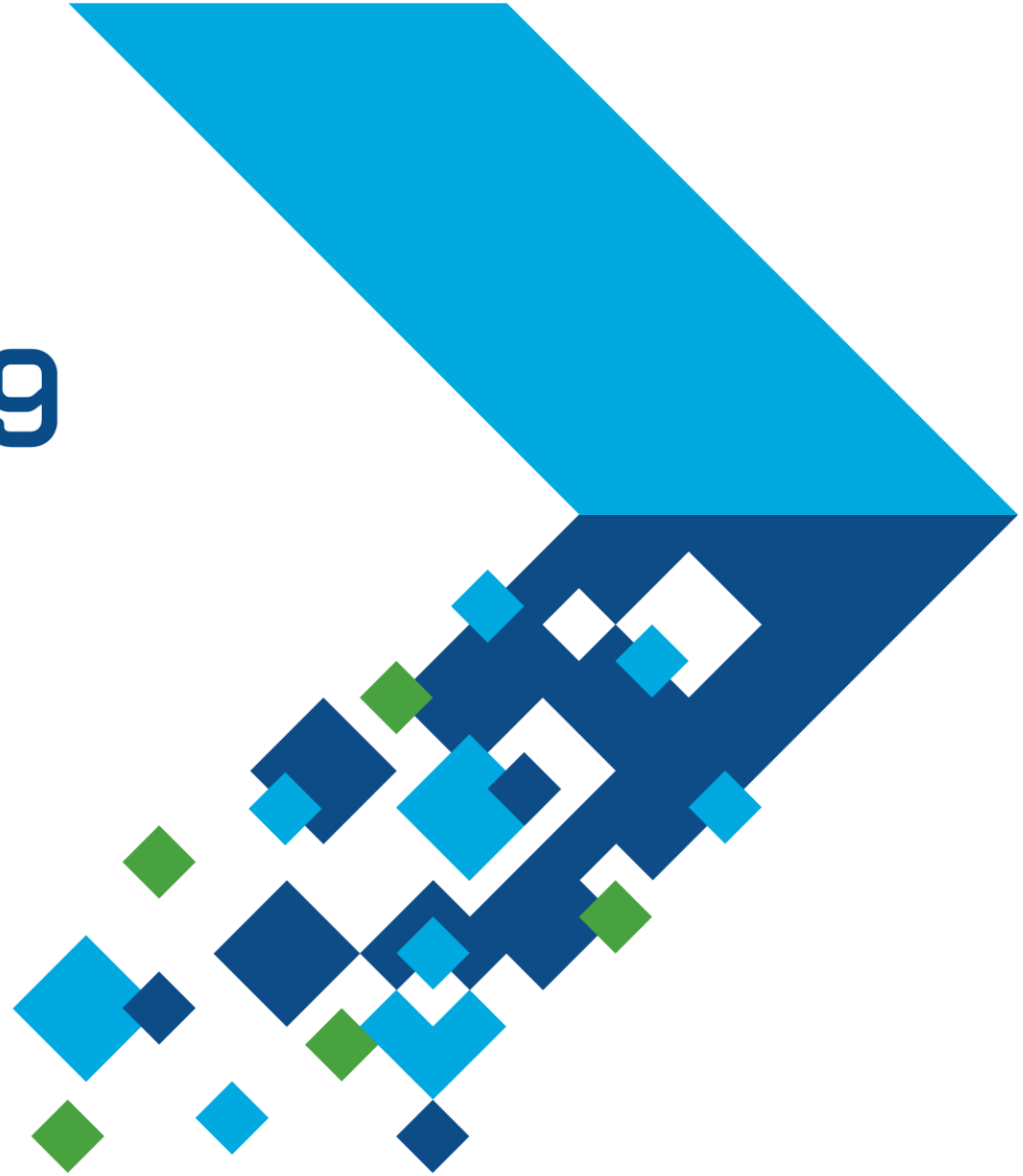
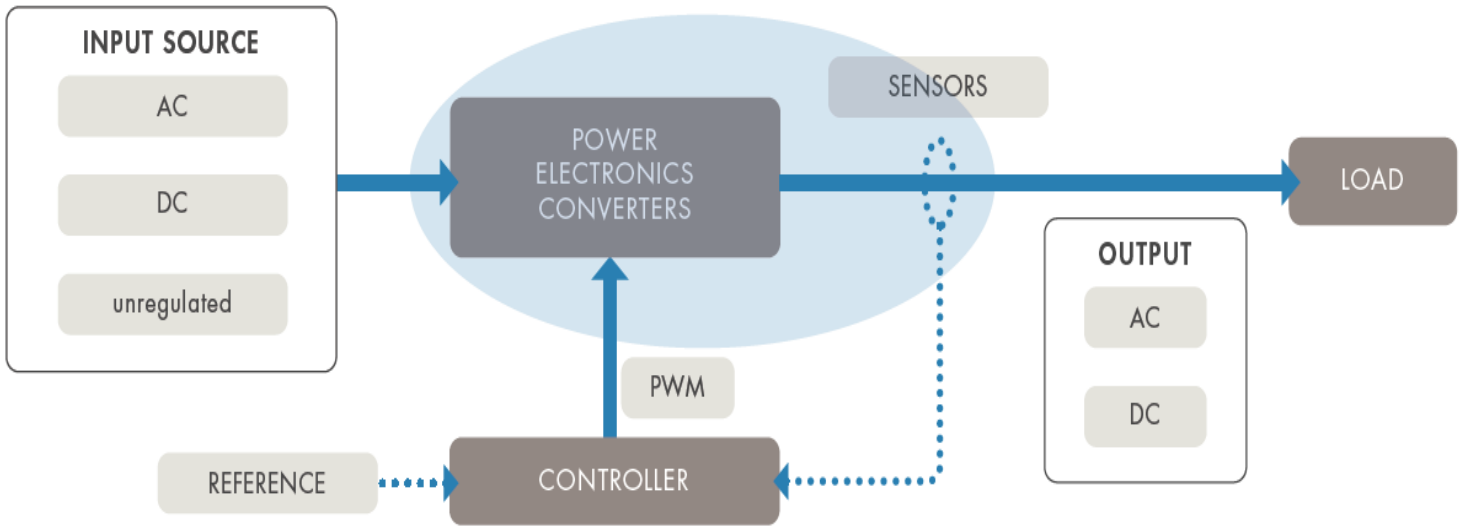
MATLAB EXPO 2019

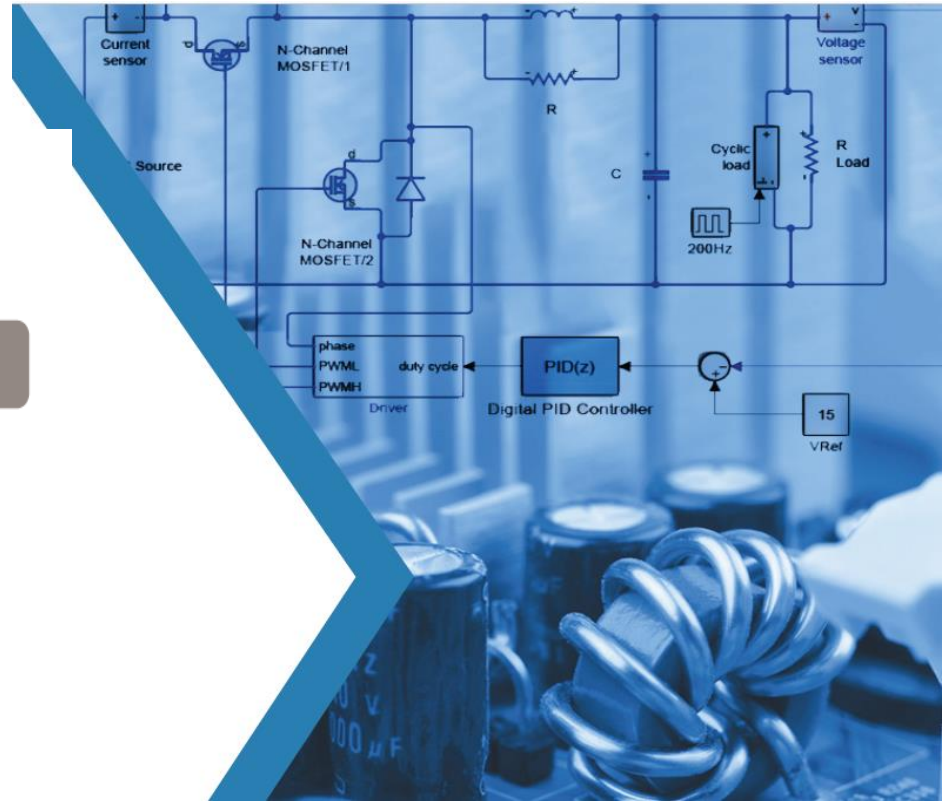
功率变换器数字控制系统开发

周前程



电力电子


INPUT SOURCE
AC
DC
unregulated
POWER ELECTRONICS CONVERTERS
SENSORS
LOAD
OUTPUT
AC
DC
PWM
REFERENCE
CONTROLLER


Current sensor
N-Channel MOSFET/1
R
Voltage sensor
Source
Cyclic load
R Load
C
N-Channel MOSFET/2
200Hz
phase
PWML
PWMH
duty cycle
PID(z)
Driver
Digital PID Controller
15
VRef

电力电子应用

电动客车

充电站

火车

新能源

LED灯

为什么要用Simulink做电力电子设计?

- 丰富的电源、负载模型
  - 光伏电池板, 电池, 电机等
- 多种电力电子器件模拟方式
  - 平均值模型, 理想开关模式, 物理模型
- 先进控制设计能力
  - 自动调参
- 自动生成可读、稳定和高效代码
  - 用于嵌入式处理器的C、C++代码
  - 用于FPGA的HDL代码
  - 用于PLC的结构化文本

村田使用Simulink进行EMS系统建模、仿真和代码生成

挑战

缩短公司首个能量管理系统产品上市时间

挑战

采用基于模型的设计方法开发产品，使用Simulink进行电力电子和控制算法建模，利用仿真进行算法设计，采用代码生成将算法模型生成可以布置到TI on Piccolo™ 和 Delfino™ 的32位处理器。

结果

- 控制软件开发时间缩减了50%以上
- 生成的代码没有任何错误
- 项目启动时间被缩短了

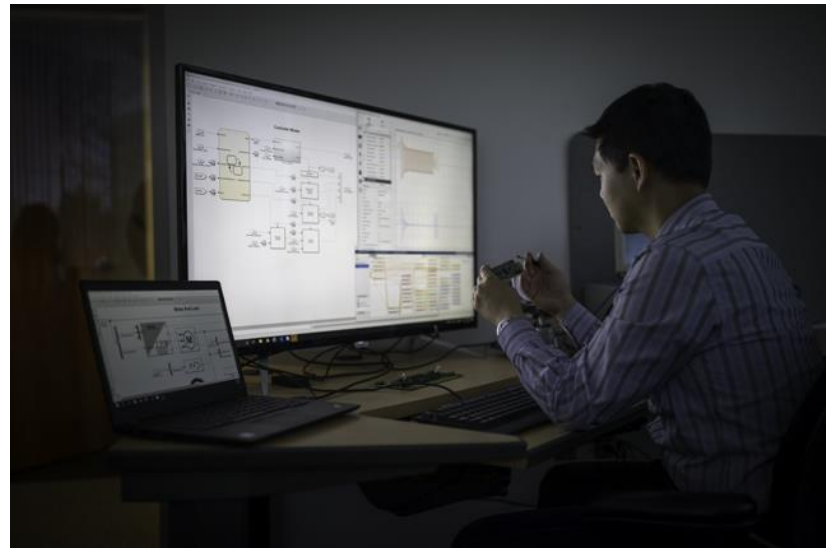
Murata flexible three-phase energy management system with lithium-ion battery.

Model-Based Design with Simulink enabled us to reduce time-to-market, which was a significant advantage for us. Because we were not expert programmers, modeling and simulating our control design and then generating quality C code from our models was essential to produce a working system as quickly as possible."

- Dr. Yue Ma, Murata Manufacturing Co., Ltd.

电力电子工程师面临的挑战

- 评估电源和负载变化对功率变流器的影响
- 所有操作和故障条件下嵌入式软件的测试
- 在软硬件集成后才能测试并发现错误
- 衡量设计是否满足效率、电能质量和安全性要求的相关标准



Our Project Today

DRV8312-C2-KIT 三相电机控制套件

功率变换器设计流程

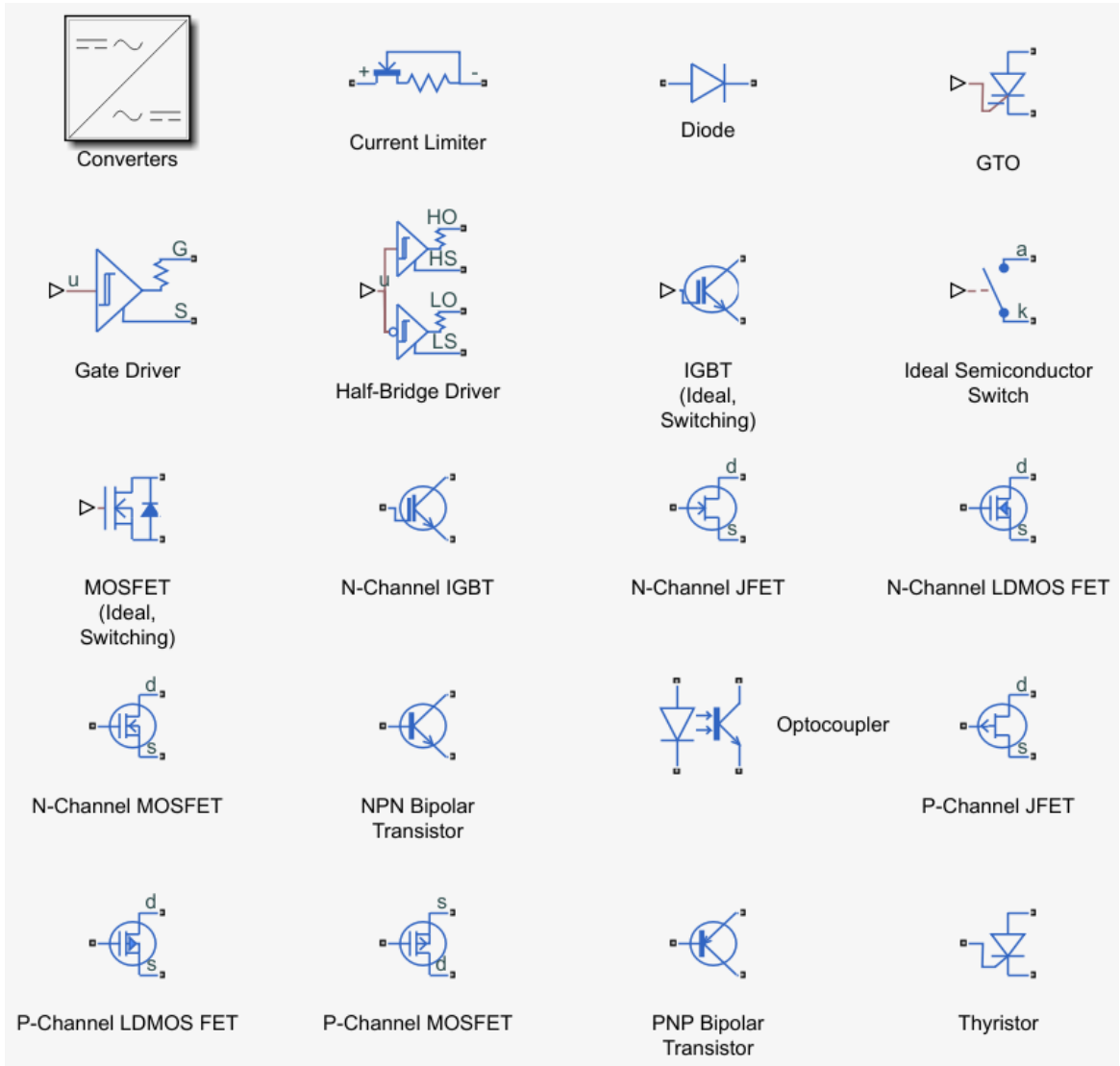
- 利用仿真验证设计
- 根据时域和频域需求设计控制算法
- 面向产品的算法设计
- 将电力电子控制布置到嵌入式处理器

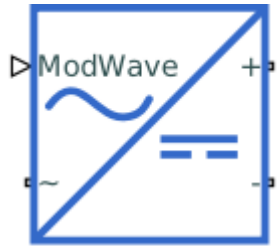
Let’s get to it!

功率变换器设计流程

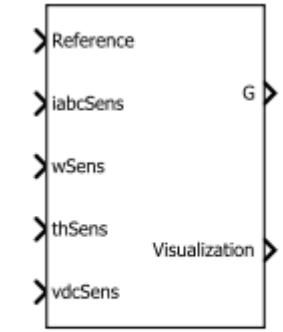
- 利用仿真验证设计
- 根据时域和频域需求设计控制算法
- 面向产品的算法设计
- 将电力电子控制布置到嵌入式处理器

丰富的参考设计

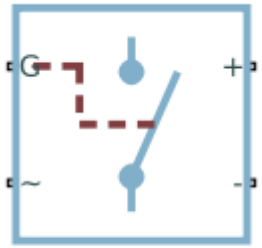




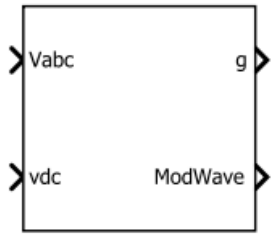
Average-Value
Voltage Source
Converter
(Three-Phase)



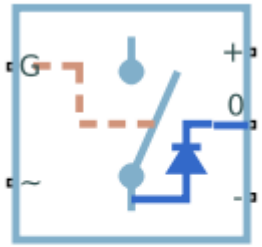
PMSM Field-Oriented
Control



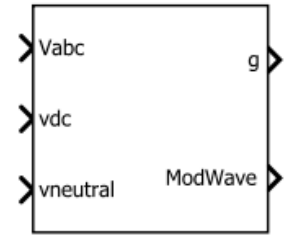
Converter
(Three-Phase)



PWM Generator
(Three-phase, Two-level)

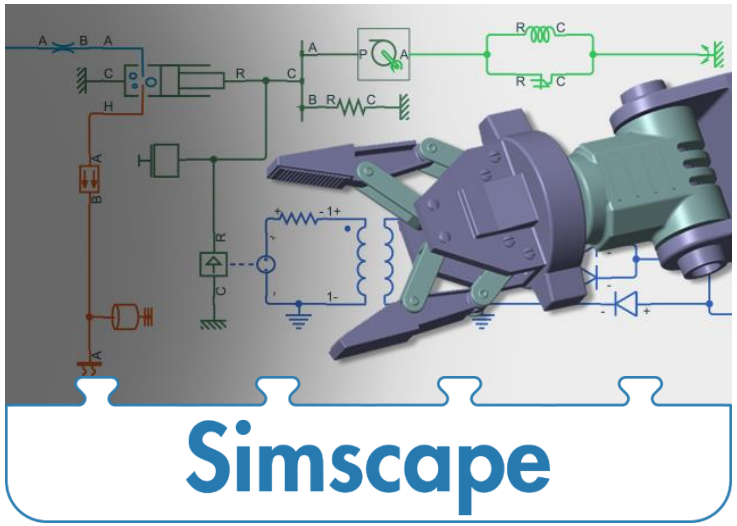


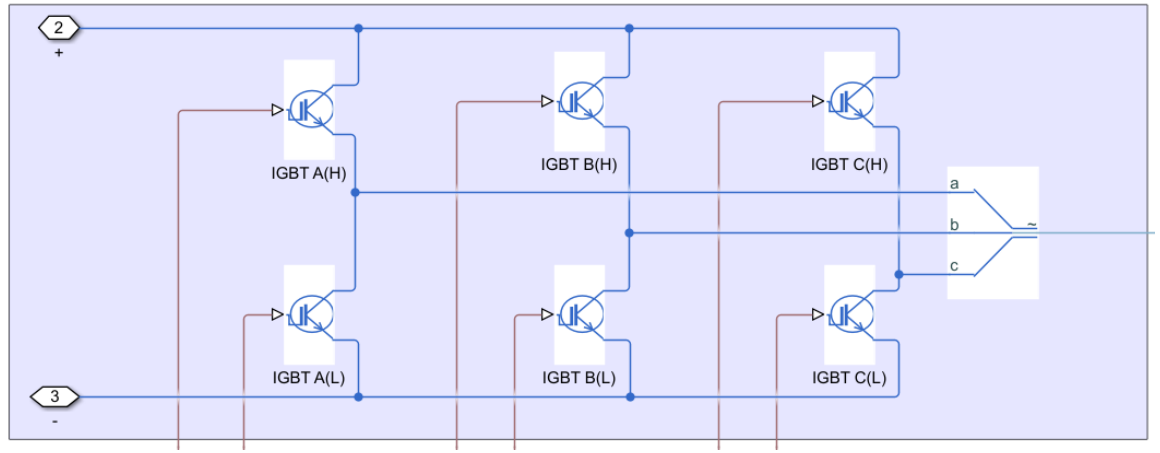
Three-Level
Converter
(Three-Phase)

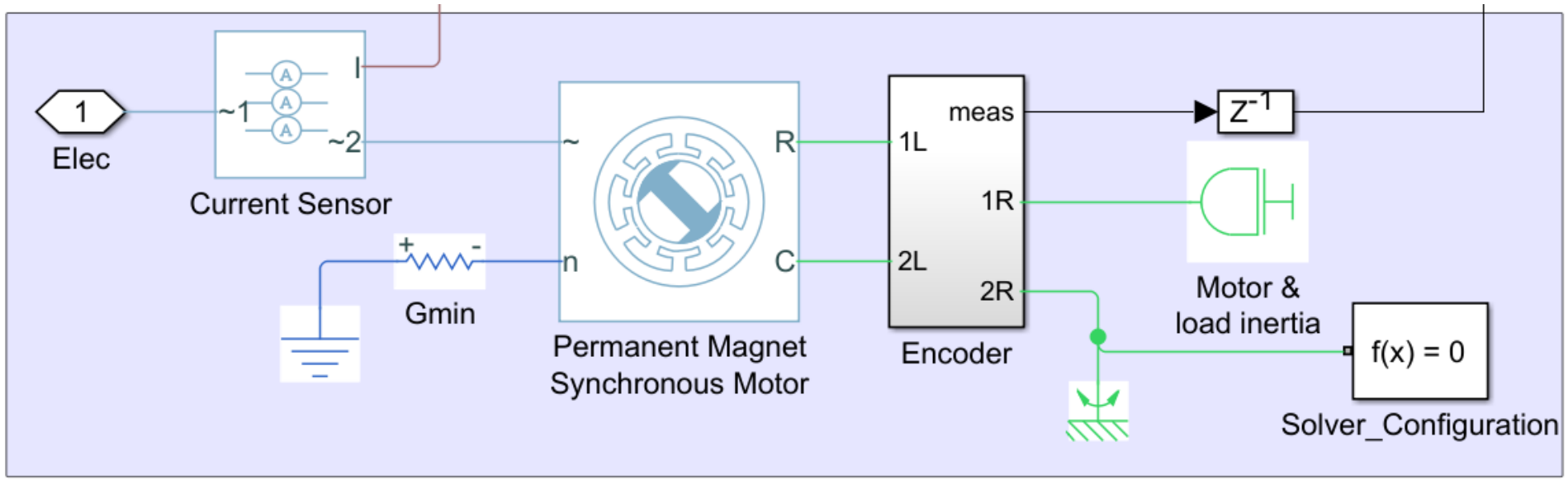


PWM Generator
(Three-phase, Three-level)

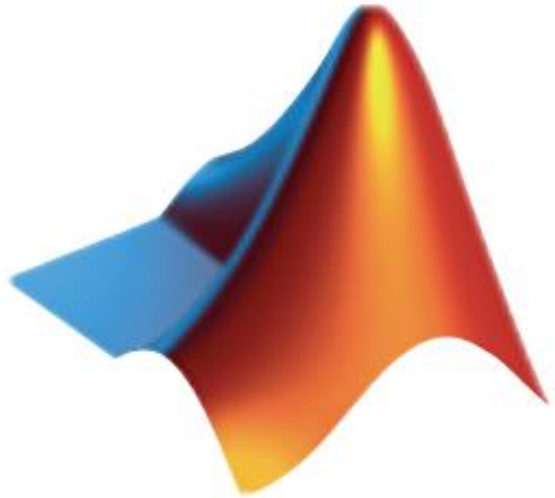
构建你自己的设计

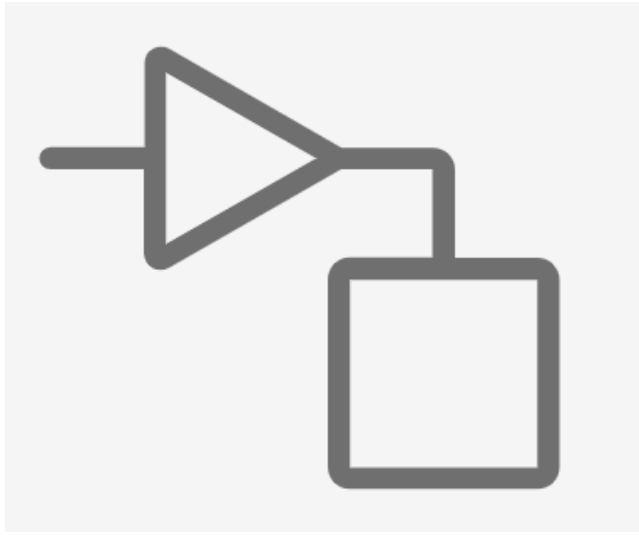

Simscape

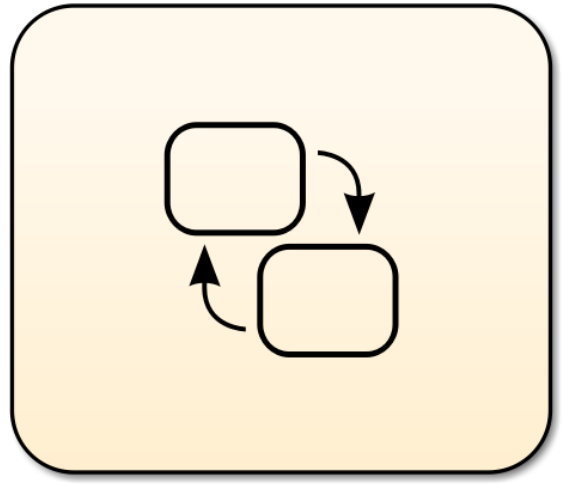

2
+
IGBT A(H)
IGBT B(H)
IGBT C(H)
a
b
c
IGBT A(L)
IGBT B(L)
IGBT C(L)
3
-


1
Elec
~1
~2
I
Current Sensor
Gmin
~
n
R
C
Permanent Magnet
Synchronous Motor
meas
1L
1R
2L
2R
Encoder
Z-1
Motor &
load inertia
f(x) = 0
Solver_Configuration

构建你自己的设计





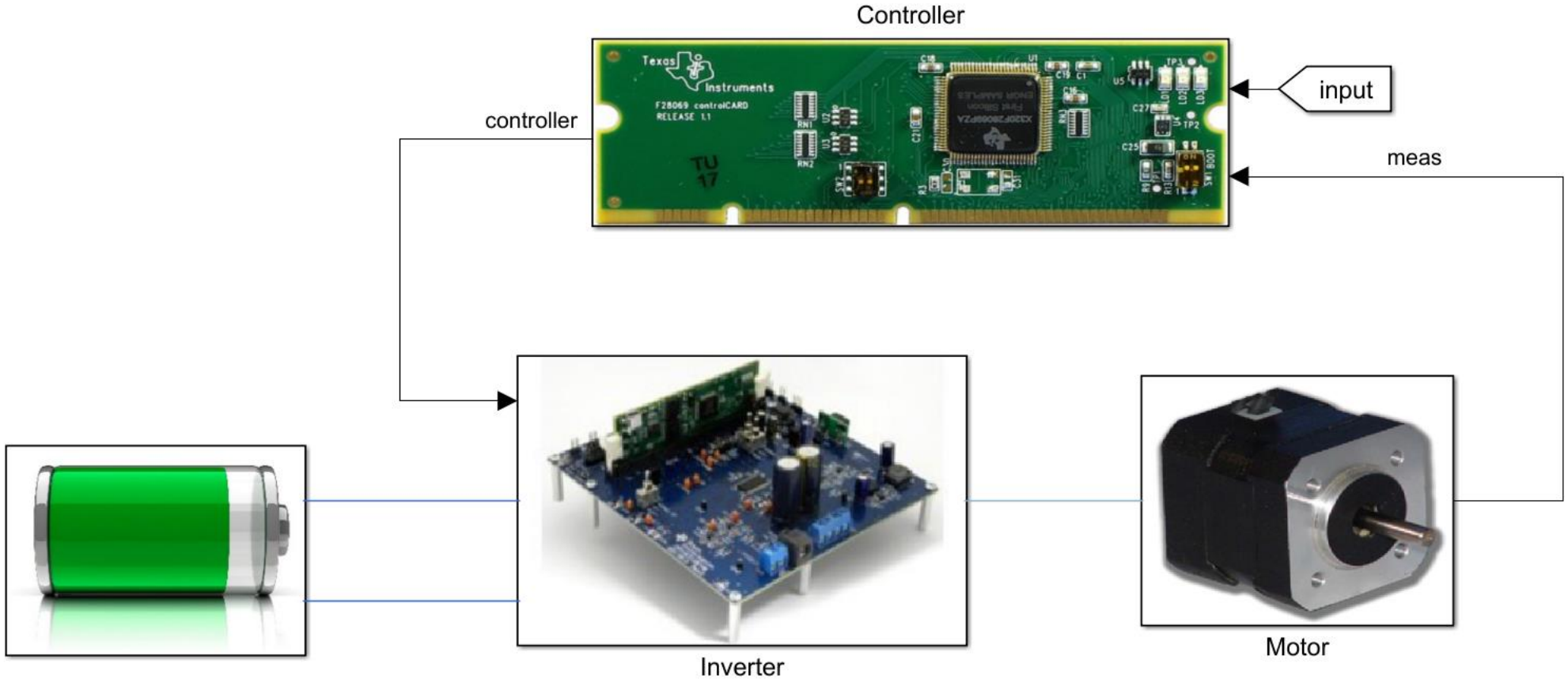


pmsm_sim_expo_ct * - Simulink

File Edit View Display Diagram Simulation Analysis Code Tools Help

pmsm_sim_expo_ct

pmsm_sim_expo_ct



Ready View 1 warning 158% FixedStepAuto

功率变换器设计流程

- 利用仿真验证设计
- 根据时域和频域需求设计控制算法
- 面向产品的算法设计
- 将电力电子控制布置到嵌入式处理器

pmsm_sim_expo_ct * - Simulink

File Edit View Display Diagram Simulation Analysis Code Tools Help

pmsm_sim_expo_ct

pmsm_sim_expo_ct

Controller

input

meas

controller

InputAndOutput

Inverter

Motor

Ready 158% auto(FixedStepDiscrete)

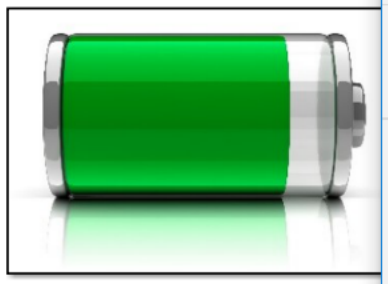
pmsm_sim_expo_ct * - Simulink

File Edit View Display Diagram Simulation Analysis Code Tools Help

pmsm_sim_expo_ct

pmsm_sim_expo_ct

电流环：Kp、Ki
转速环：Kp、Ki



Simulation Data Inspector - untitled*

<rotorVelocity>

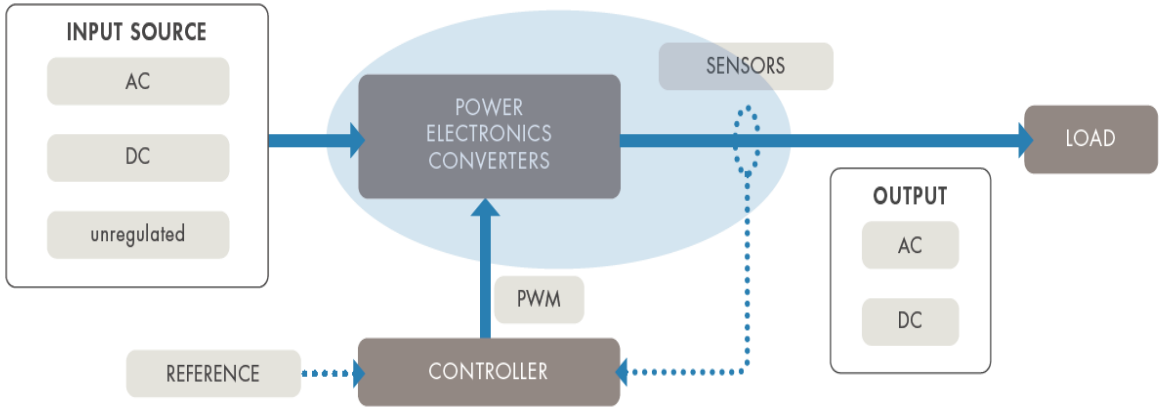
100
50
0
-50
0 0.05 0.10 0.15 0.20 0.25 0.30 0.35 0.40 0.45 0.50 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00

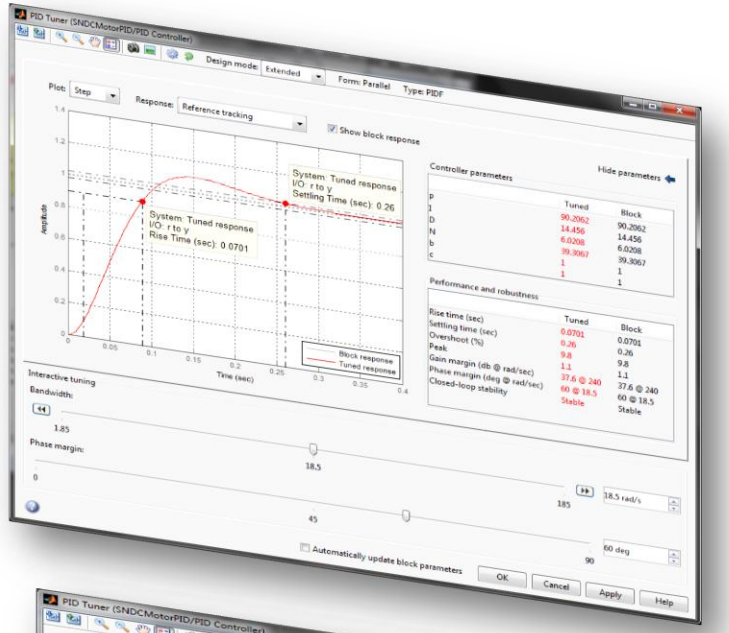
velocityCommand

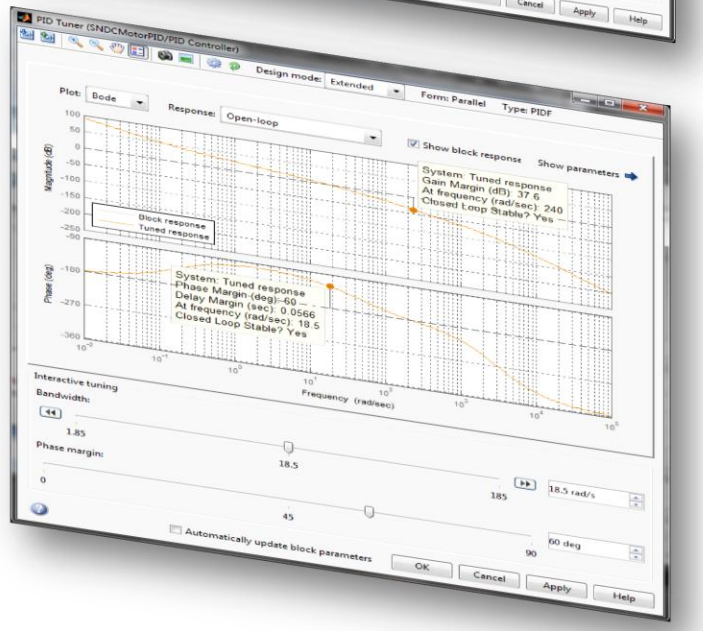
100
80
60
40
20
0
0 0.05 0.10 0.15 0.20 0.25 0.30 0.35 0.40 0.45 0.50 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00

Ready View 1 warning 158% auto(FixedStepDiscrete)

自动化控制参数设计







多种手段

- PID Tuner
- Control System Tuner
- Response Optimization

Response Optimization

- 设定调节目标
- 选择参数
- 开始优化

pmsm_sim_expo_ct * - Simulink

File Edit View Display Diagram Simulation Analysis Code Tools Help

pmsm_sim_expo_ct

pmsm_sim_expo_ct

Controller

controller

input

meas

InputAndOutput

Inverter

Motor

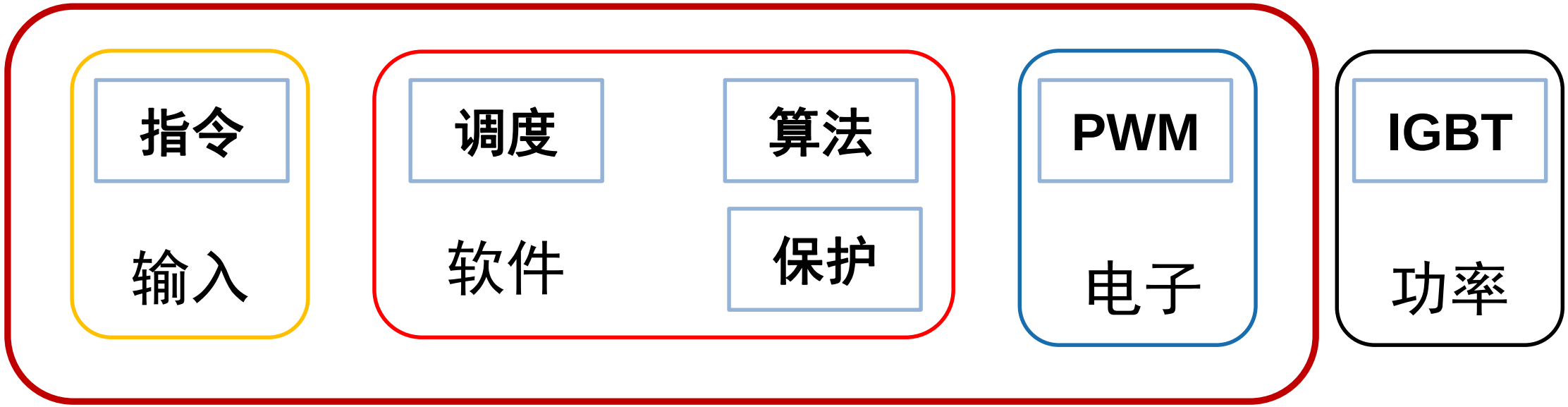
Ready View 1 warning 158% auto(FixedStepDiscrete)

功率变换器设计流程

- 利用仿真验证设计
- 根据时域和频域需求设计控制算法
- 面向产品的算法设计
- 将电力电子控制布置到嵌入式处理器

电力电子控制系统特点

- 层级结构

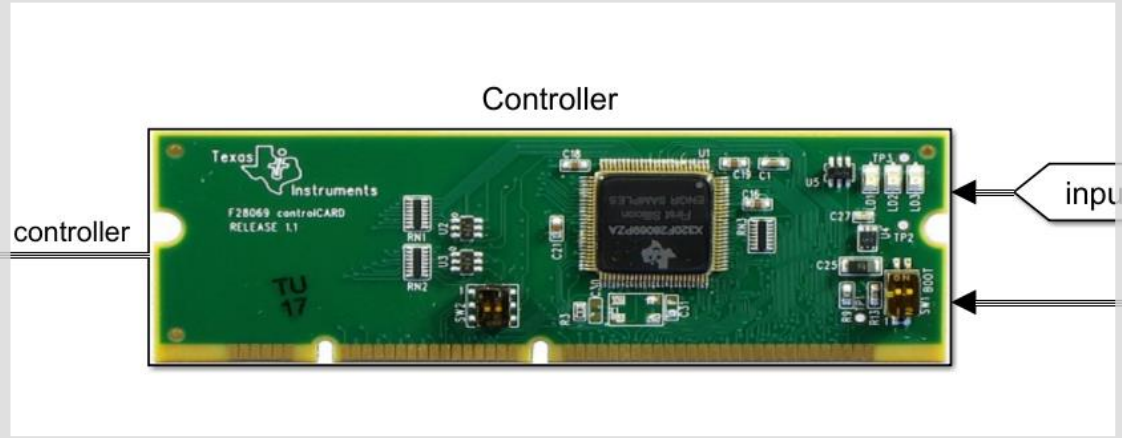


pmsm_sim_expo_ct * - Simulink

File Edit View Display Diagram Simulation Analysis Code Tools Help

pmsm_sim_expo_ct

pmsm_sim_expo_ct



InputAndOutput

输入预处理

核心控制算法

PWM

Inverter

Motor

启停控制

功能调度

装置保护

功能实现

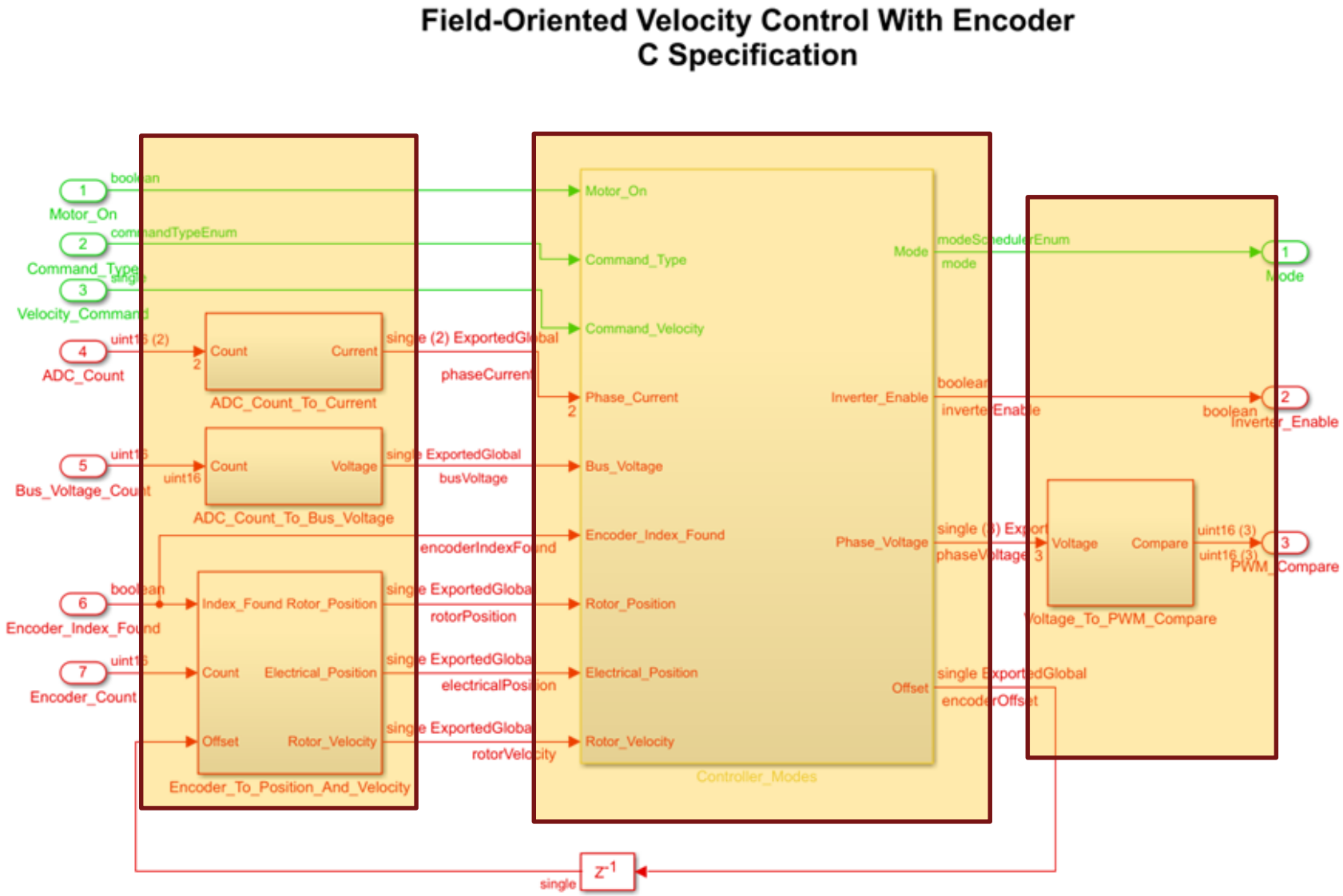
Ready

View 1 warning

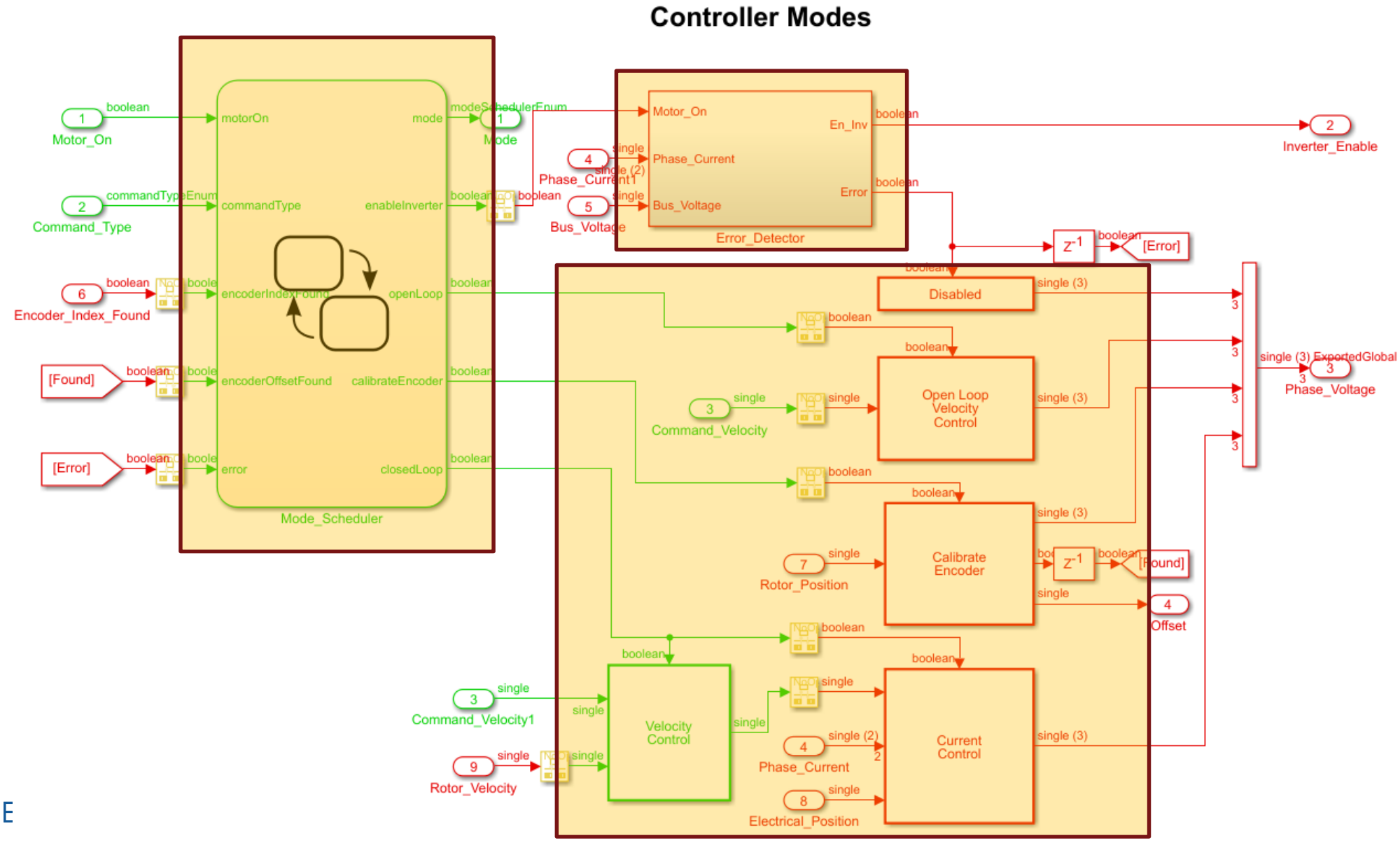
158%

auto(FixedStepDiscrete)

控制算法模型

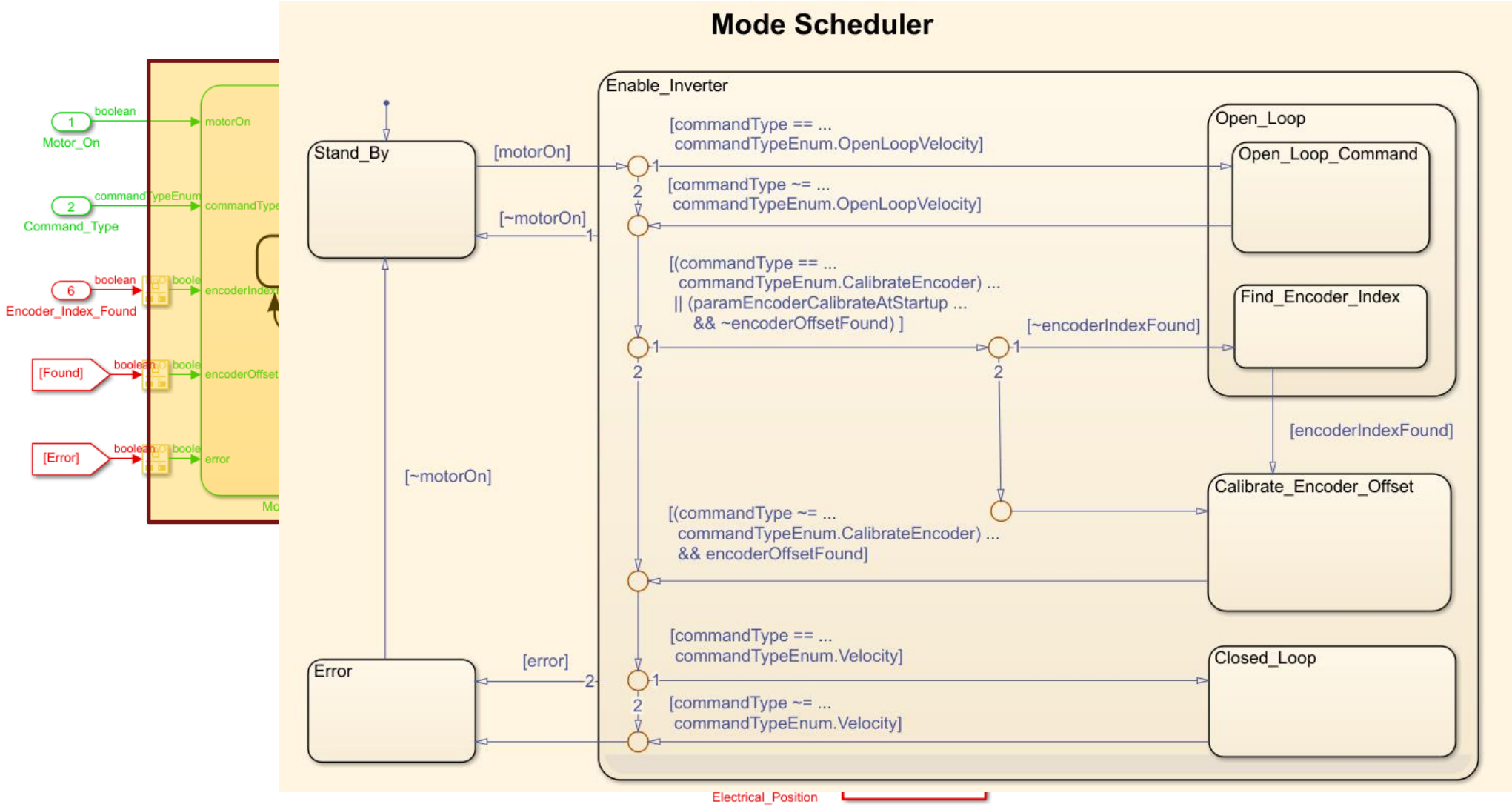

Field-Oriented Velocity Control With Encoder
C Specification
1
Motor_On
2
Command_Type
3
Velocity_Command
4
ADC_Count
5
Bus_Voltage_Count
6
Encoder_Index_Found
7
Encoder_Count
Count
Current
ADC_Count_To_Current
Count
Voltage
ADC_Count_To_Bus_Voltage
Index_Found Rotor_Position
Count
Electrical_Position
Offset
Rotor_Velocity
Encoder_To_Position_And_Velocity
phaseCurrent
busVoltage
encoderIndexFound
rotorPosition
electricalPosition
rotorVelocity
Motor_On
Command_Type
Command_Velocity
Phase_Current
Bus_Voltage
Encoder_Index_Found
Rotor_Position
Electrical_Position
Rotor_Velocity
Mode
Inverter_Enable
Phase_Voltage
Offset
Controller_Modes
modeSchedulerEnum
mode
boolean
inverterEnable
phaseVoltage
encoderOffset
Voltage
Compare
Voltage_To_PWM_Compare
1
Mode
2
boolean
Inverter_Enable
3
PWM_Compare
single
z-1

核心控制算法建模方式


Controller Modes
1
Motor_On
boolean
2
Command_Type
commandTypeEnum
6
Encoder_Index_Found
boolean
[Found]
boolean
[Error]
boolean
motorOn
commandType
encoderIndexFound
encoderOffsetFound
error
mode
enableInverter
openLoop
calibrateEncoder
closedLoop
Mode_Scheduler
modeSchedulerEnum
1
Mode
boolean
4
Phase_Current1
single (2)
5
Bus_Voltage
single
Motor_On
Phase_Current
Bus_Voltage
En_Inv
Error
Error_Detector
2
Inverter_Enable
Disabled
Open Loop Velocity Control
Calibrate Encoder
Velocity Control
Current Control
3
Command_Velocity
single
7
Rotor_Position
3
Command_Velocity1
9
Rotor_Velocity
4
Phase_Current
8
Electrical_Position
Offset
single (3)
single (3) ExportedGlobal
3
Phase_Voltage
Z-1

核心控制算法建模方式

使用Stateflow来调度各种电机控制算法的运行模式



功率变换器设计流程

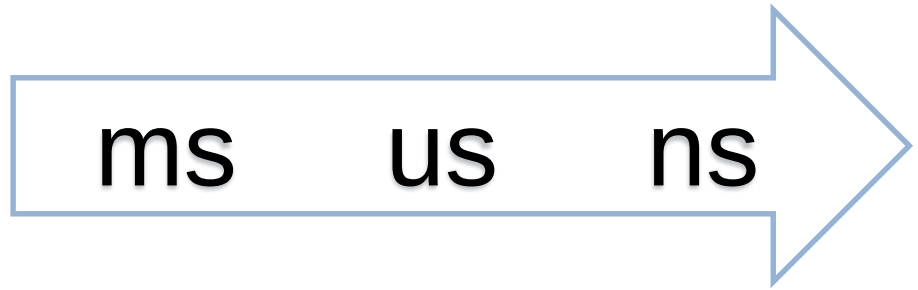
- 利用仿真验证设计
- 根据时域和频域需求设计控制算法
- 面向产品的算法设计
- 将电力电子控制布置到嵌入式处理器

电力电子控制系统特点

- 嵌入式架构

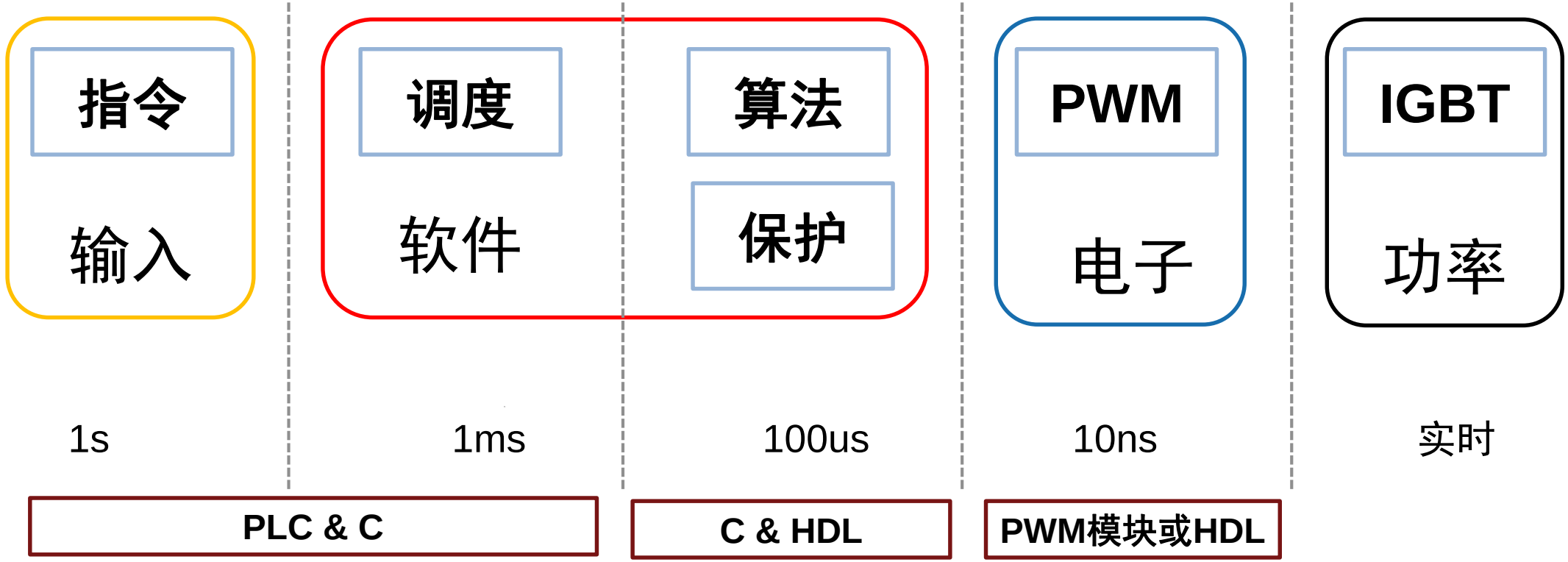
DSP Arm FPGA SoC

- 实时性高

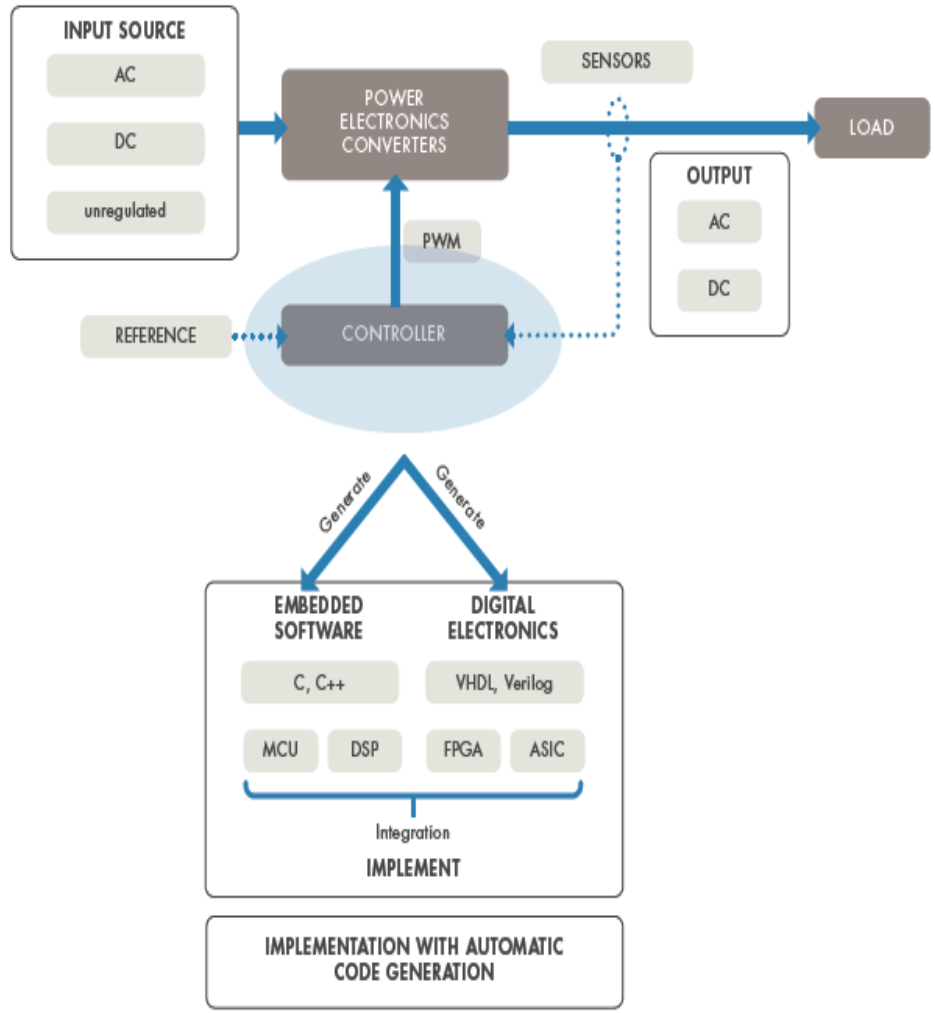


电力电子控制系统特点

- 层级控制



将控制布置到嵌入式处理



任务:

- 按计算周期划分结构
- 从MATLAB和Simulink生成优化的代码

从模型到代码

C

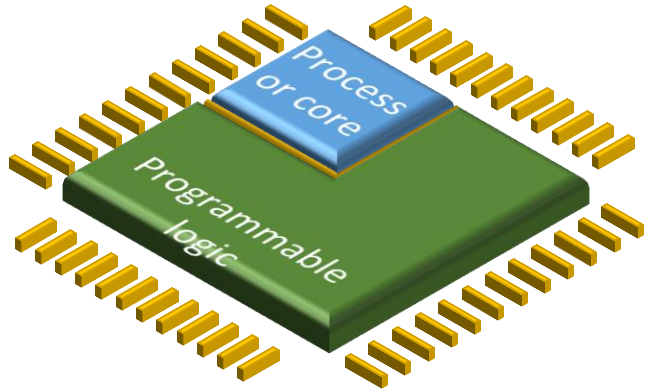
编码规范: AUTOSAR, MISRA C®, and ASAP2
代码与模型的相互追溯
代码生成报告
自动代码验证
符合DO-178、IEC 61508和ISO 26262中的软件开发要求

HDL

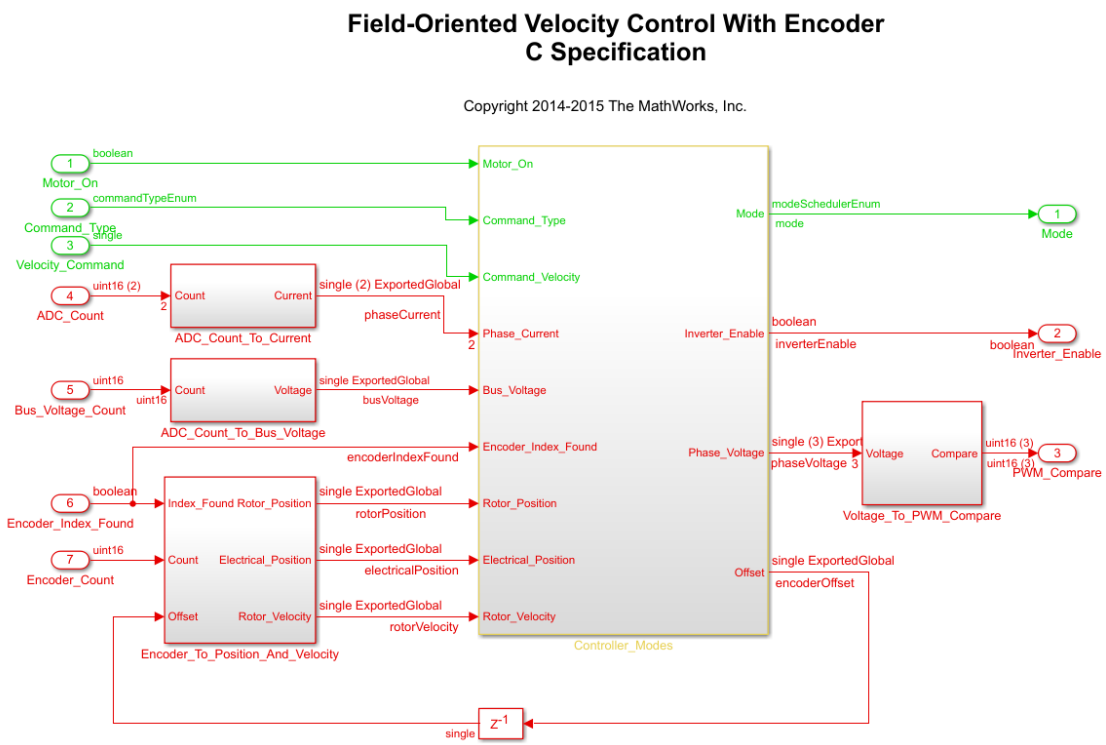
支持定点和浮点代码生成
配合定点化工具箱完成算法定点化设计
与Xilinx、Microsemi、Intel FPGAs设计工具无缝集成
符合DO 254标准要求

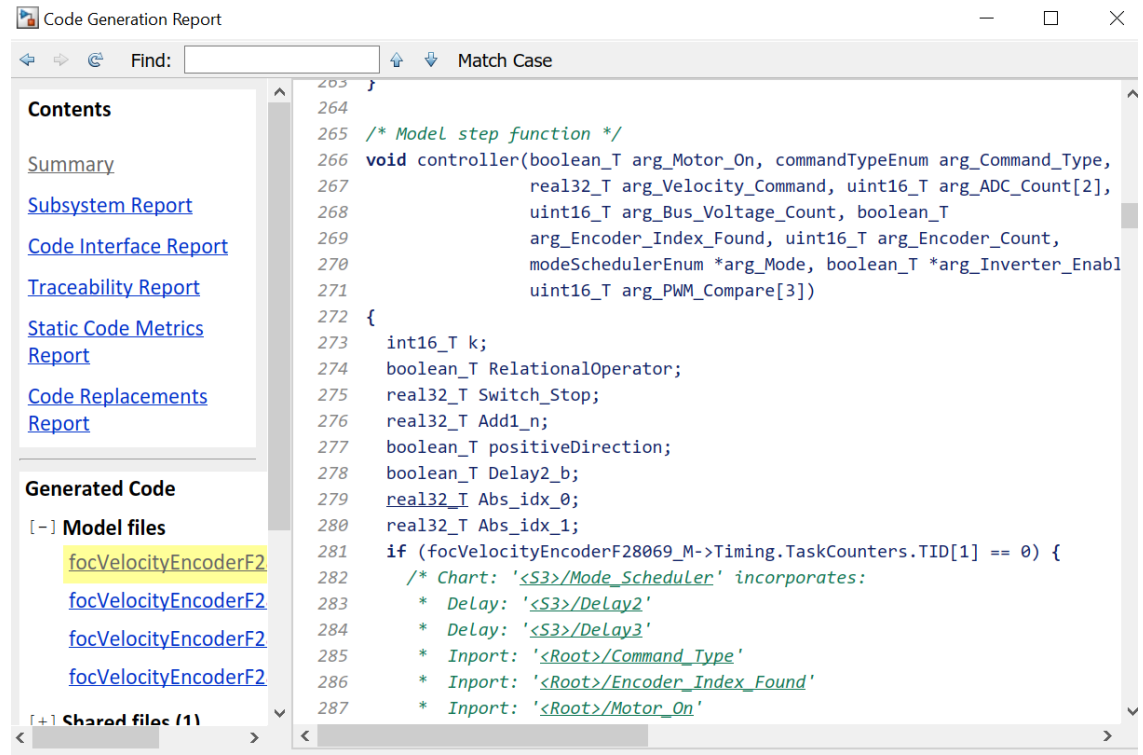
PLC

IEC 61131-3标准结构化文本和梯形图
支持CODESYS、Rockwell、Siemens、Omron等公司产品
符合 IEC 61508和 IEC61511

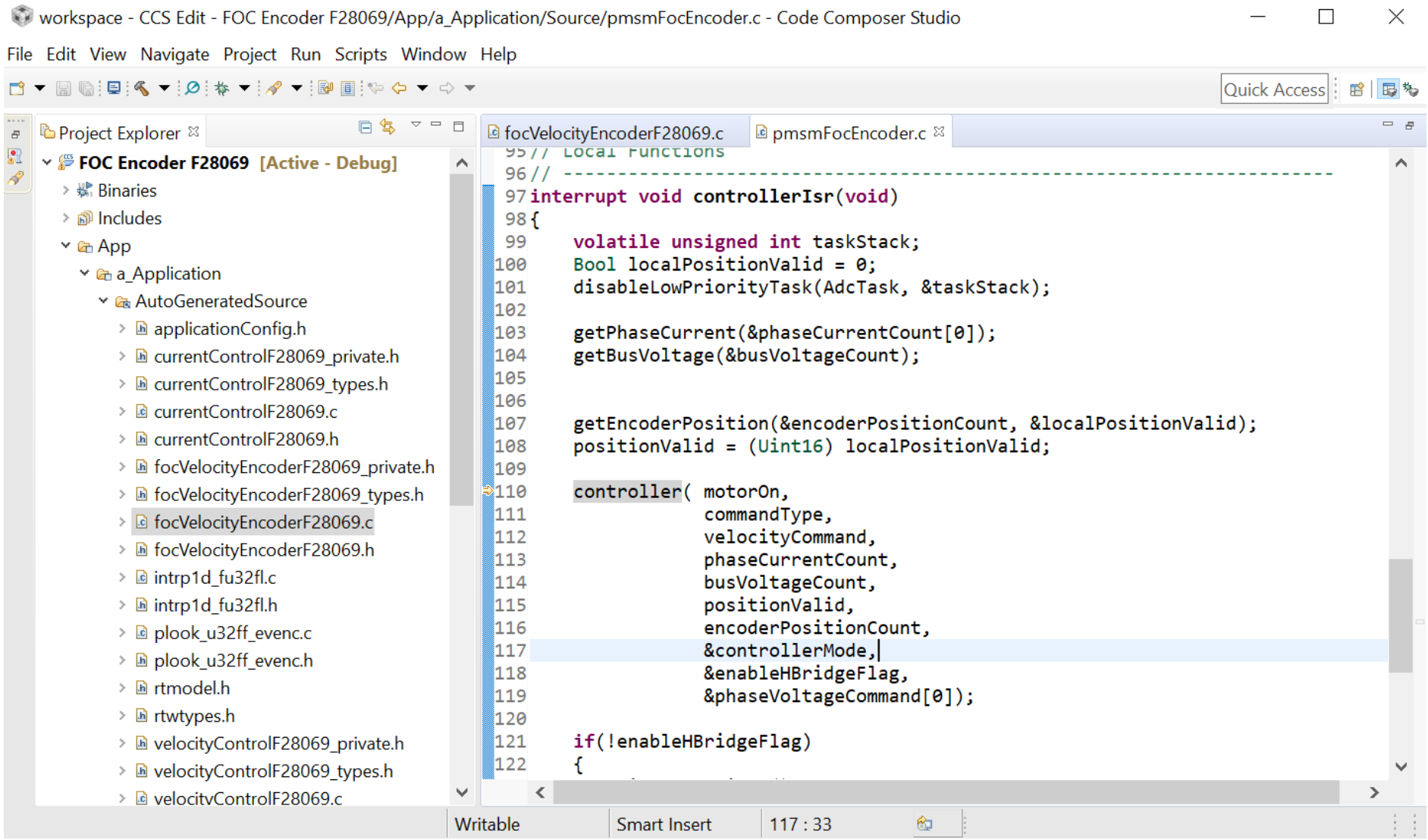


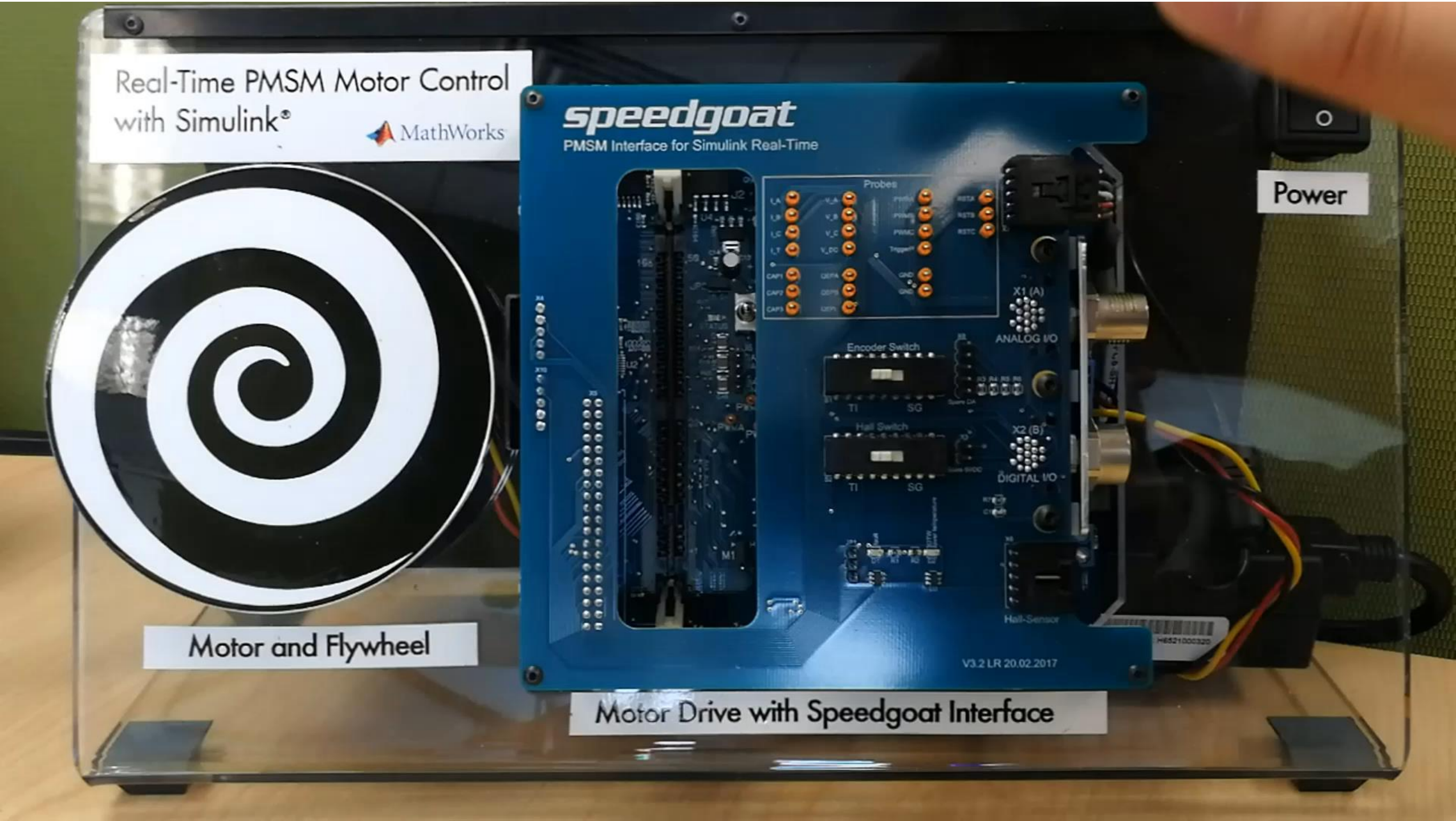
从模型到代码


Field-Oriented Velocity Control With Encoder
C Specification
Copyright 2014-2015 The MathWorks, Inc.
Motor_On
Command_Type
Command_Velocity
Phase_Current
Bus_Voltage
Encoder_Index_Found
Rotor_Position
Electrical_Position
Rotor_Velocity
Mode
Inverter_Enable
Phase_Voltage
Offset
Controller_Modes
ADC_Count_To_Current
ADC_Count_To_Bus_Voltage
Encoder_To_Position_And_Velocity
Voltage_To_PWM_Compare
modeSchedulerEnum
mode
inverterEnable
phaseCurrent
busVoltage
encoderIndexFound
rotorPosition
electricalPosition
rotorVelocity
encoderOffset
single
boolean
commandTypeEnum
uint16
uint16 (2)
single (2) ExportedGlobal
single ExportedGlobal
single (3) Export
uint16 (3)
Motor_On
Command_Type
Velocity_Command
ADC_Count
Bus_Voltage_Count
Encoder_Index_Found
Encoder_Count
Mode
Inverter_Enable
PWM_Compare


Code Generation Report
Find:
Match Case
Contents
Summary
Subsystem Report
Code Interface Report
Traceability Report
Static Code Metrics Report
Code Replacements Report
Generated Code
[-] Model files
focVelocityEncoderF2
focVelocityEncoderF2
focVelocityEncoderF2
focVelocityEncoderF2
[+] Shared files (1)
263 }
264
265 /* Model step function */
266 void controller(boolean_T arg_Motor_On, commandTypeEnum arg_Command_Type,
267 real32_T arg_Velocity_Command, uint16_T arg_ADC_Count[2],
268 uint16_T arg_Bus_Voltage_Count, boolean_T
269 arg_Encoder_Index_Found, uint16_T arg_Encoder_Count,
270 modeSchedulerEnum *arg_Mode, boolean_T *arg_Inverter_Enabl
271 uint16_T arg_PWM_Compare[3])
272 {
273 int16_T k;
274 boolean_T RelationalOperator;
275 real32_T Switch_Stop;
276 real32_T Add1_n;
277 boolean_T positiveDirection;
278 boolean_T Delay2_b;
279 real32_T Abs_idx_0;
280 real32_T Abs_idx_1;
281 if (focVelocityEncoderF28069_M->Timing.TaskCounters.TID[1] == 0) {
282 /* Chart: '<S3>/Mode_Scheduler' incorporates:
283 * Delay: '<S3>/Delay2'
284 * Delay: '<S3>/Delay3'
285 * Inport: '<Root>/Command_Type'
286 * Inport: '<Root>/Encoder_Index_Found'
287 * Inport: '<Root>/Motor_On'

软件集成


workspace - CCS Edit - FOC Encoder F28069/App/a_Application/Source/pmsmFocEncoder.c - Code Composer Studio
File Edit View Navigate Project Run Scripts Window Help
Quick Access
Project Explorer
FOC Encoder F28069 [Active - Debug]
Binaries
Includes
App
a_Application
AutoGeneratedSource
applicationConfig.h
currentControlF28069_private.h
currentControlF28069_types.h
currentControlF28069.c
currentControlF28069.h
focVelocityEncoderF28069_private.h
focVelocityEncoderF28069_types.h
focVelocityEncoderF28069.c
focVelocityEncoderF28069.h
intrp1d_fu32fl.c
intrp1d_fu32fl.h
plook_u32ff_evenc.c
plook_u32ff_evenc.h
rtmodel.h
rtwtypes.h
velocityControlF28069_private.h
velocityControlF28069_types.h
velocityControlF28069.c
focVelocityEncoderF28069.c
pmsmFocEncoder.c
// Local Functions
// ---
interrupt void controllerIsr(void)
{
volatile unsigned int taskStack;
Bool localPositionValid = 0;
disableLowPriorityTask(AdcTask, &taskStack);
getPhaseCurrent(&phaseCurrentCount[0]);
getBusVoltage(&busVoltageCount);
getEncoderPosition(&encoderPositionCount, &localPositionValid);
positionValid = (Uint16) localPositionValid;
controller(motorOn,
commandType,
velocityCommand,
phaseCurrentCount,
busVoltageCount,
positionValid,
encoderPositionCount,
&controllerMode,
&enableHBridgeFlag,
&phaseVoltageCommand[0]);
if(!enableHBridgeFlag)
{
Writable
Smart Insert
117 : 33


Real-Time PMSM Motor Control with Simulink®
MathWorks
speedgoat
PMSM Interface for Simulink Real-Time
Probes
Encoder Switch
Hall Switch
X1 (A)
ANALOG I/O
X2 (B)
DIGITAL I/O
Hall-Sensor
V3.2 LR 20.02.2017
Power
Motor and Flywheel
Motor Drive with Speedgoat Interface

基于模型的设计加速ABB 大规模并网逆变器开发

挑战

加速大规模并网逆变器设计与上市

解决方案

使用基于模型的设计进行功率模块和装置的建模、仿真和代码生成

结果

- 两周内完成原型样机的开发，再次之前需要三周
- 自动生成没有bug的优化的代码
- 减少了测试带来的设备损失

A cabinet of Power Electronic Building Blocks (PEBBs).

“Simulink and Embedded Coder enabled us to open the door to new markets. With increased productivity from extensive simulation and efficient code generation, we have confidence in our ability to produce the systems that larger customers are asking for in the time frames they want.”

- Dr. Robert Turner, ABB

为什么要用Simulink做电力电子设计?

- 丰富的电源、负载模型
  - 光伏电池板, 电池, 电机等
- 多种电力电子器件模拟方式
  - 平均值模型, 理想开关模式, 物理模型
- 先进控制设计能力
  - 自动调参
- 自动生成可读、稳定和高效代码
  - 用于嵌入式处理器的C、C++代码
  - 用于FPGA的HDL代码
  - 用于PLC的结构化文本