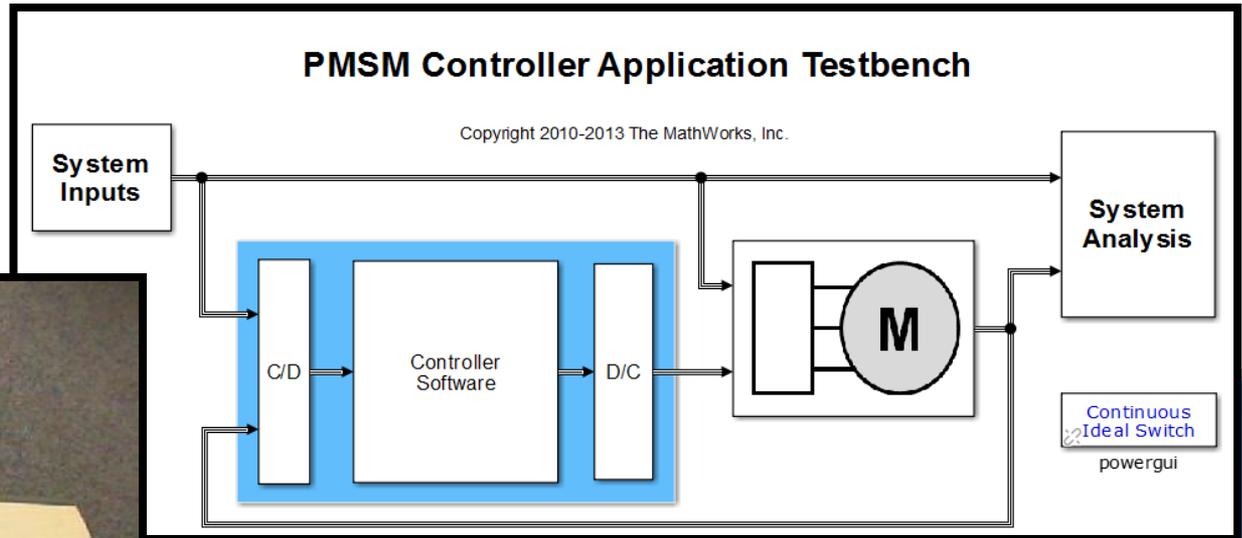
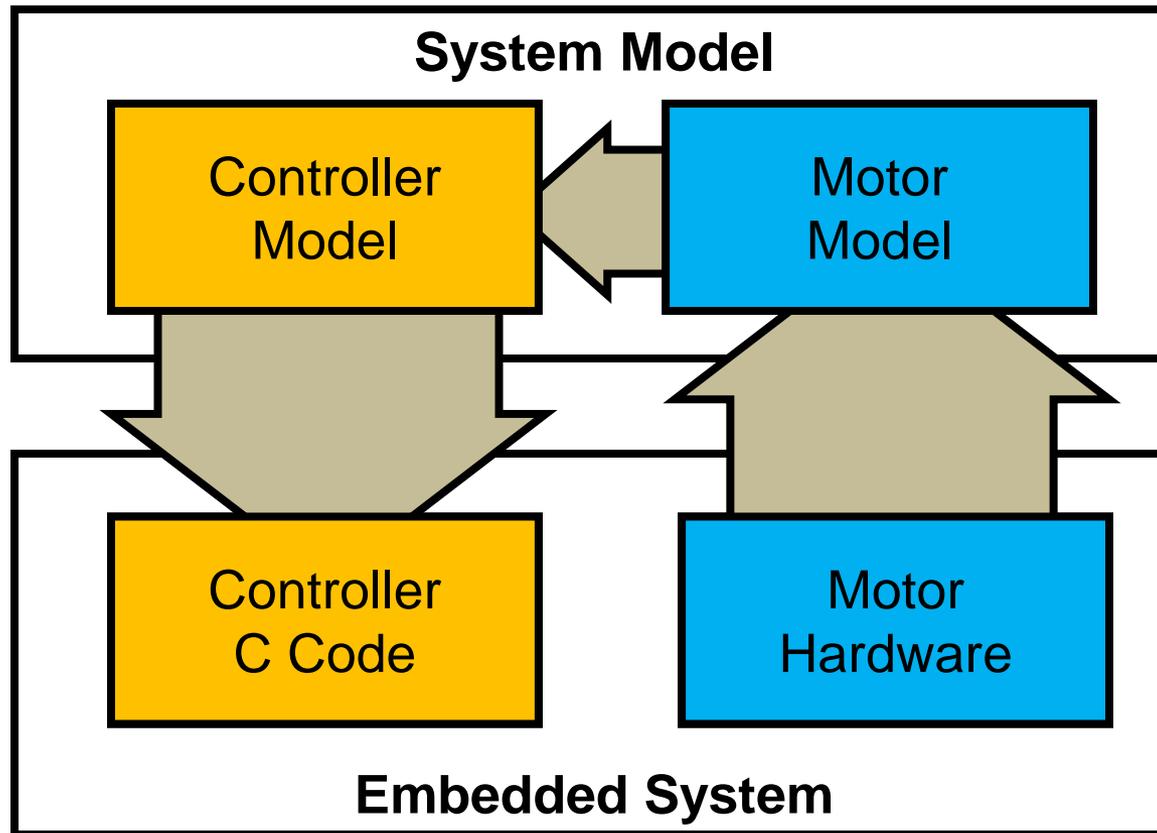


Introduction to Production Code Generation in Simulink: An AC motor control application



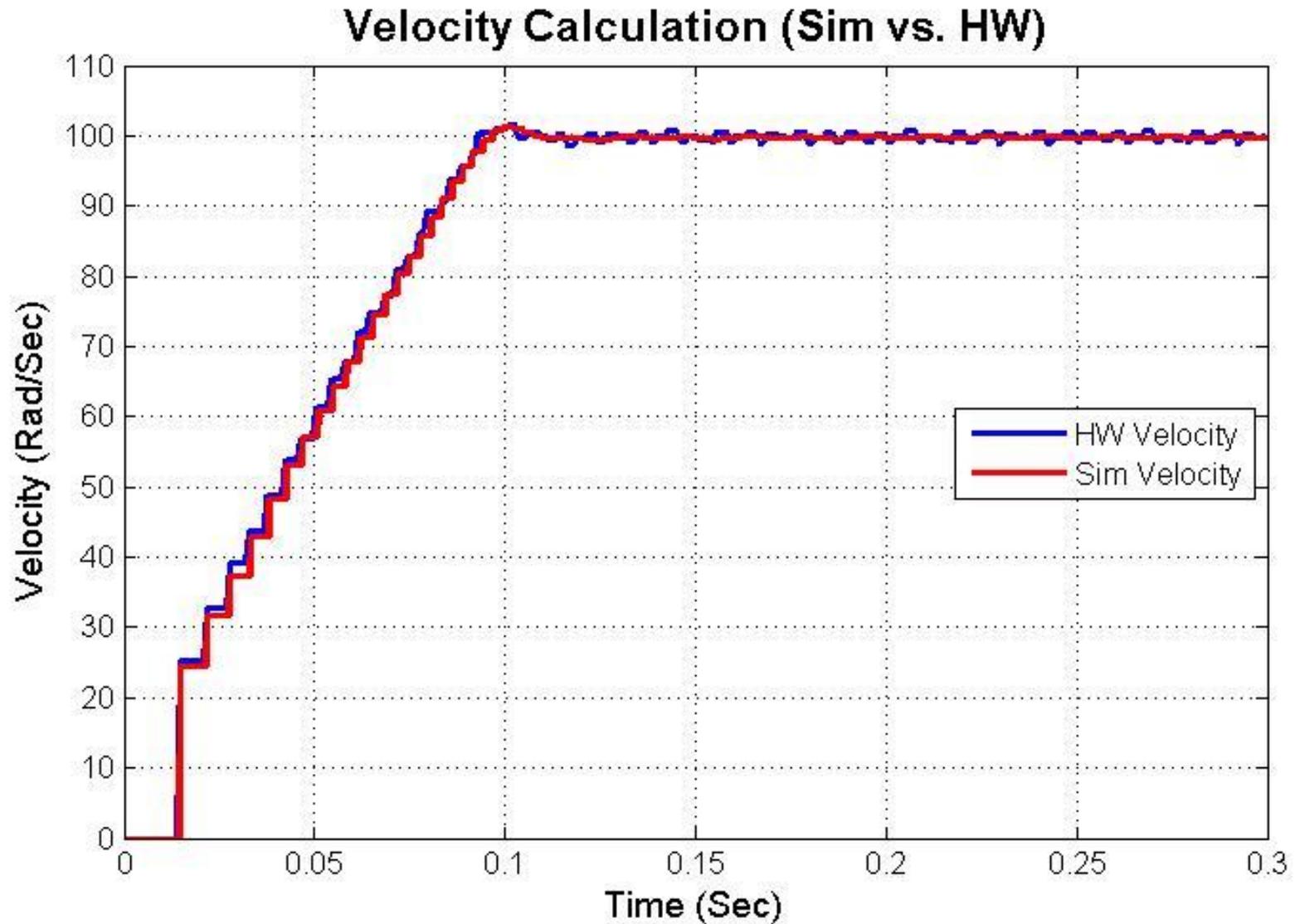
Jeff Tackett, Application Engineering

Model your system, design your controller, and generate code using Simulink



**Let's explore the AC Motor Control
application...**

Simulation vs. Hardware Measurements



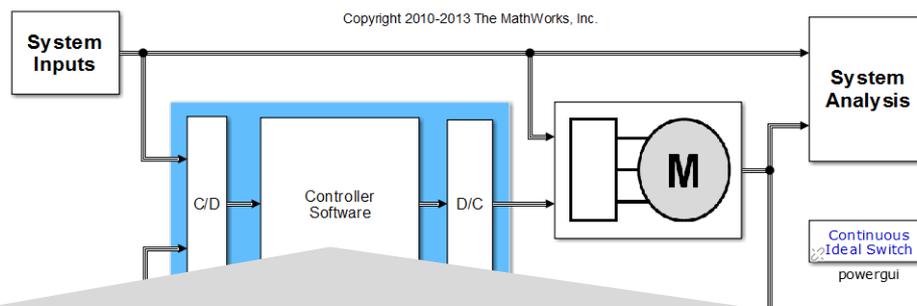
Introduction to Code Generation

- Configuring model settings
- Specifying the function prototype
- Exporting parameters as global variables
- Packaging the source and header files

Where do you start when adopting Model-Based Design?

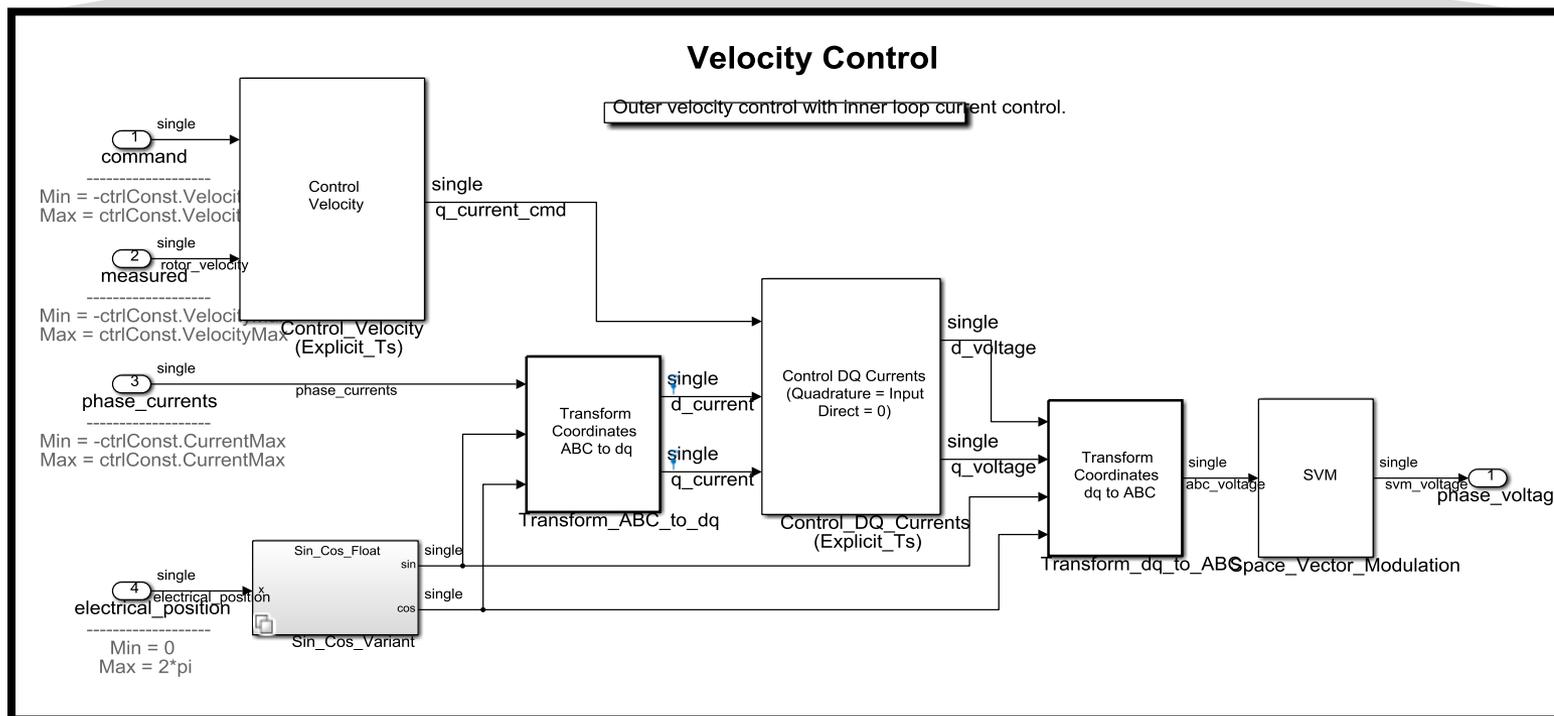
Start with a core controller component

PMSM Controller Application Testbench



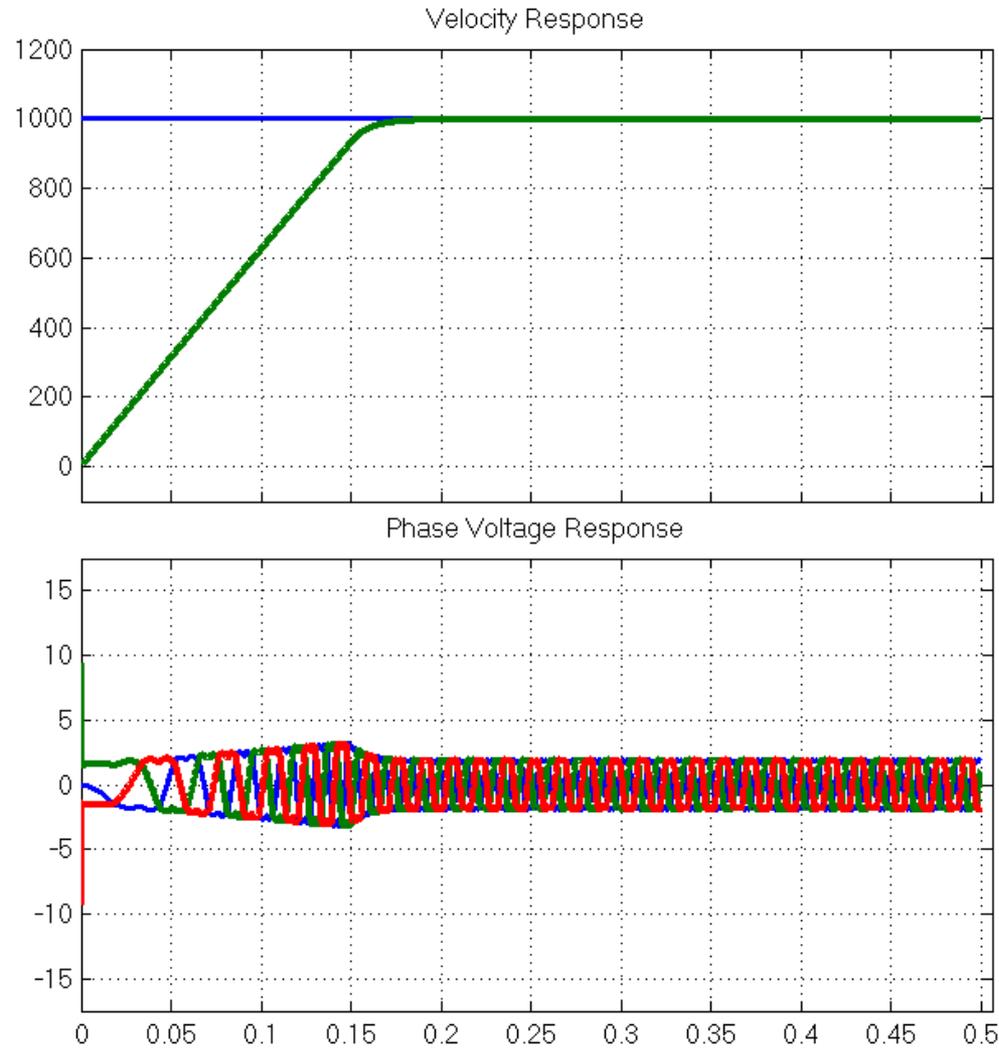
Velocity Control

Outer velocity control with inner loop current control.

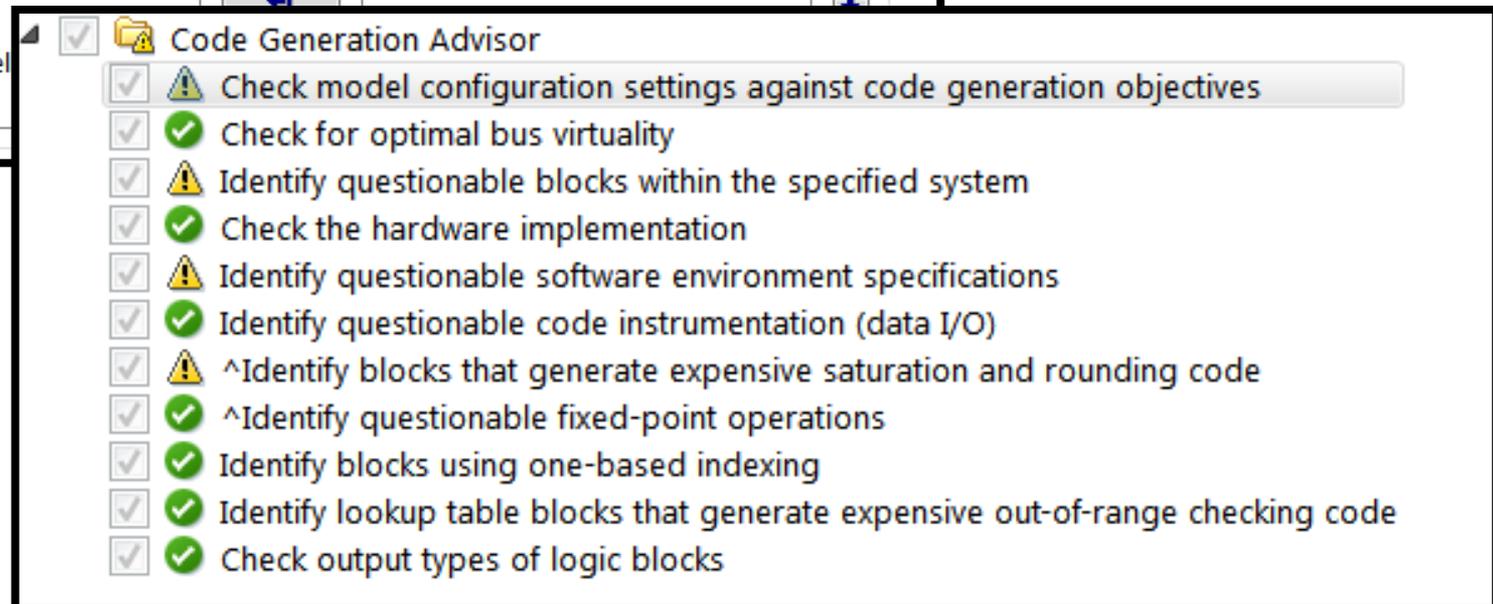
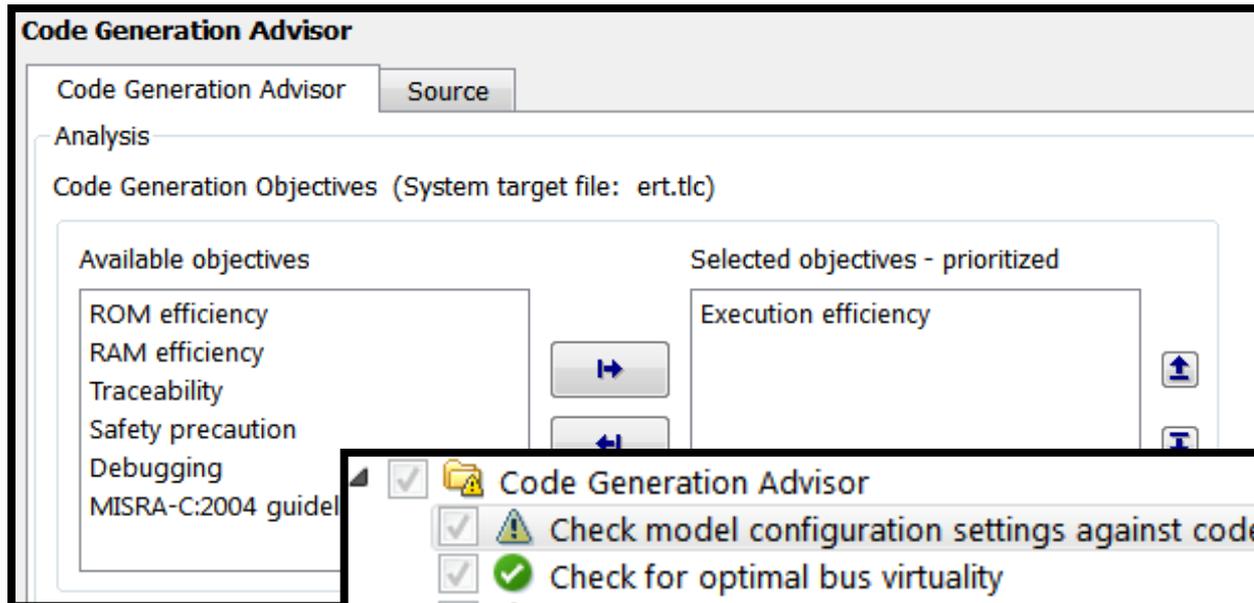


Let's explore this component...

Simulate to Understand Behavior



Prep Model for Code Generation with “Code Generation Objectives”



Control Interface Elements

Desired Function Prototype:

```
void Controller(real32_T arg_command, real32_T arg_measured,
real32_T arg_phase_currents[2], real32_T arg_electrical_position,
real32_T arg_phase_voltages[3])
```

Configuration:

Configure model initialize and step functions

Initialize function name:

Step function name:

Step function arguments:

Order	Port Name	Port Type	Category	Argument Name	Qualifier
1	command	Inport	Value	arg_command	none
2	measured	Inport	Value	arg_measured	none
3	phase_curr...	Inport	Pointer	arg_phase_curre...	none
4	electrical_...	Inport	Value	arg_electrical_po...	none
5	phase_volt...	Outport	Pointer	arg_phase_volta...	none

Step function preview

```
Controller ( arg_command, arg_measured, * arg_phase_currents, arg_electrical_position, *
arg_phase_voltages )
```

Exporting global tuning parameters

Specify data object in model explorer:

Simulink.Parameter: paramVelocityPICurrentSat

Value:

Data type:

Dimensions: Complexity:

Minimum: Maximum:

Units:

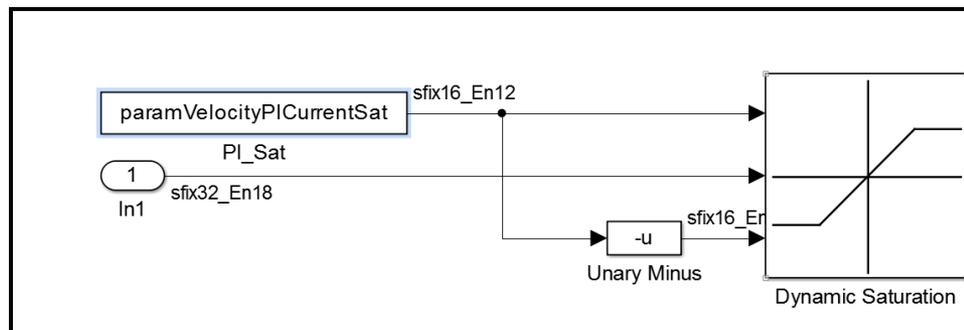
Code generation options

Storage class:

Alias:

Alignment:

Use data object for constant value:



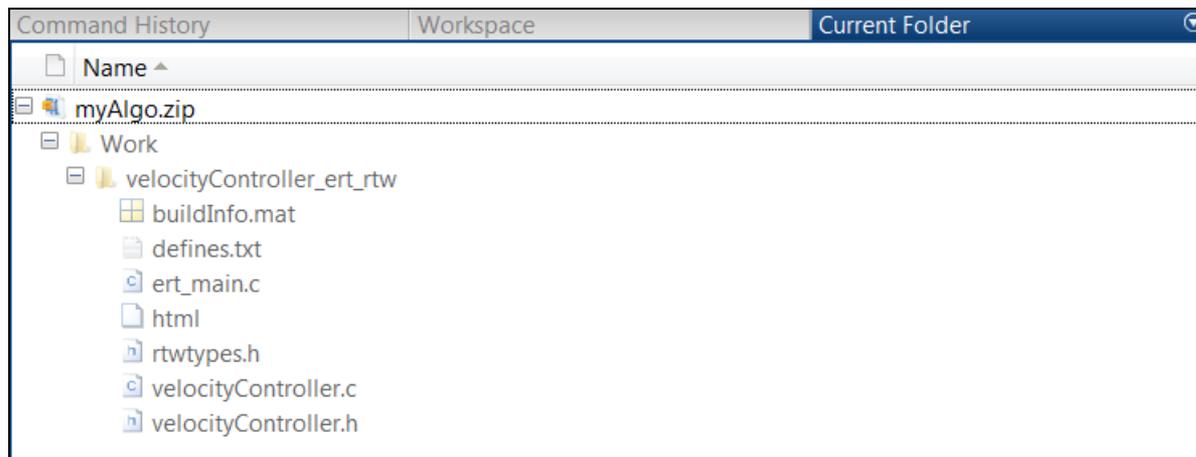
Package up the generated code and distribute

Enable and specify filename:

Generate code only
 Package code and artifacts

Zip file name:

Zip file created with all required source files:



Navigating between the model and generated code

Code Generation Report

Back Forward Search...

Contents

- Summary
- Subsystem Report
- Code Interface Report
- Traceability Report
- Static Code Metrics Report
- Code Replacements Report

Generated Code

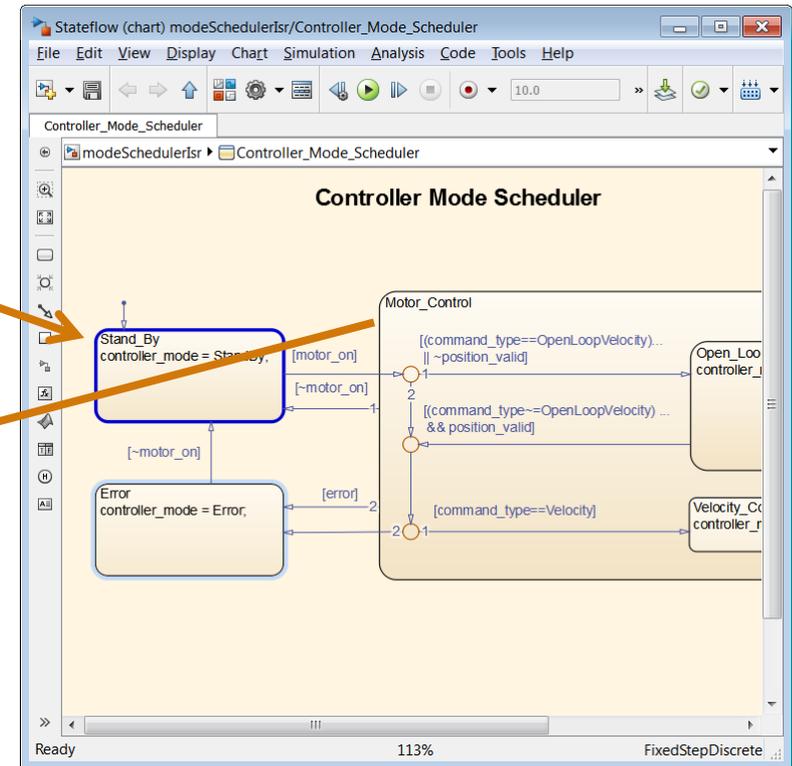
- (-) Main file
 - [ert_main.c](#)
- (-) Model files
 - [modeSchedulerIsr.c](#)
 - [modeSchedulerIsr.h](#)
- (+) Utility files (1)

```

66  /* Entry: Controller_Mode_Scheduler */
67  modeSchedulerIsr_DW.is_active_c1_modeSchedulerIsr = 1U;
68
69  /* Entry Internal: Controller_Mode_Scheduler */
70  /* Transition: '<S1>:9' */
71  modeSchedulerIsr_DW.is_c1_modeSchedulerIsr = modeSchedulerIsr_
72
73  /* Entry 'Stand By': '<S1>:154' */
74  modeSchedulerIsr_DW.controller_mode = StandBy;
75  } else {
76  switch (modeSchedulerIsr_DW.is_c1_modeSchedulerIsr) {
77  case modeSchedulerIsr_IN_Error:
78  /* During 'Error': '<S1>:374' */
79  if (!(arg_motor_on != 0U)) {
80  /* Transition: '<S1>:375' */
81  modeSchedulerIsr_DW.is_c1_modeSchedulerIsr =
82  modeSchedulerIsr_IN_Stand_By;
83
84  /* Entry 'Stand By': '<S1>:154' */
85  modeSchedulerIsr_DW.controller_mode = StandBy;
86  }
87  break;
88
89  case modeSchedulerIsr_IN_Motor_Control:
90  /* During 'Motor_Control': '<S1>:372' */
91  if (!(arg_motor_on != 0U)) {
92  /* Transition: '<S1>:474' */
93  /* Exit Internal 'Motor_Control': '<S1>:372' */
94  modeSchedulerIsr_DW.is_Motor_Control = modeSchedule_IN_NO_
95  modeSchedulerIsr_DW.is_c1_modeSchedulerIsr =
96  modeSchedulerIsr_IN_Stand_By;
97
98  /* Entry 'Stand By': '<S1>:154' */
99  modeSchedulerIsr_DW.controller_mode = StandBy;
100  } else if (arg_error != 0U) {
101  /* Transition: '<S1>:559' */

```

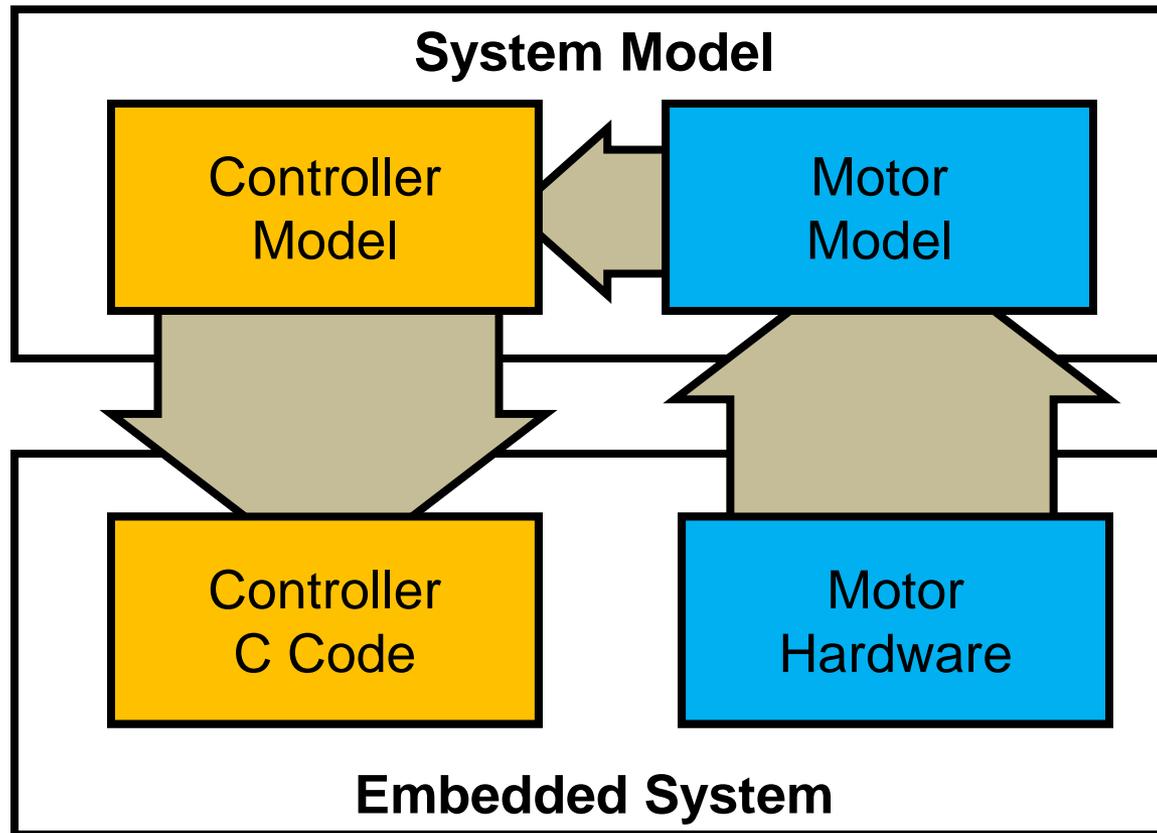
OK Help



What did we learn from this example?

- Initial steps for production code generation
 - Code generation advisor
 - Function prototype control
 - Exporting global tuning parameters
 - Navigating between the model and generated code
 - Package up code and artifacts

Model your system, design your controller, and generate code using Simulink



Where can I find more information?

- Explore a demo on generating C code from a controller and verifying its compiled behavior and execution time

```
>> docsearch "Field-Oriented Control of Permanent Magnet Synchronous Machine"
```


http://mathworks.com/products/demos/shipping/rtw/rtwdemo_pmsmfoc_script.html
- Watch recorded webinar: Embedded Code Generation for AC Motors
<http://www.mathworks.com/company/events/webinars/wbnr61549.html>
- Watch 2012 MathWorks Automotive Conference recording of: Parameterizing and Verifying a Permanent Magnet Synchronous Motor Model
<http://www.mathworks.com/company/events/conferences/automotive-conference-michigan/2012/proceedings/registration.html?video=3>

