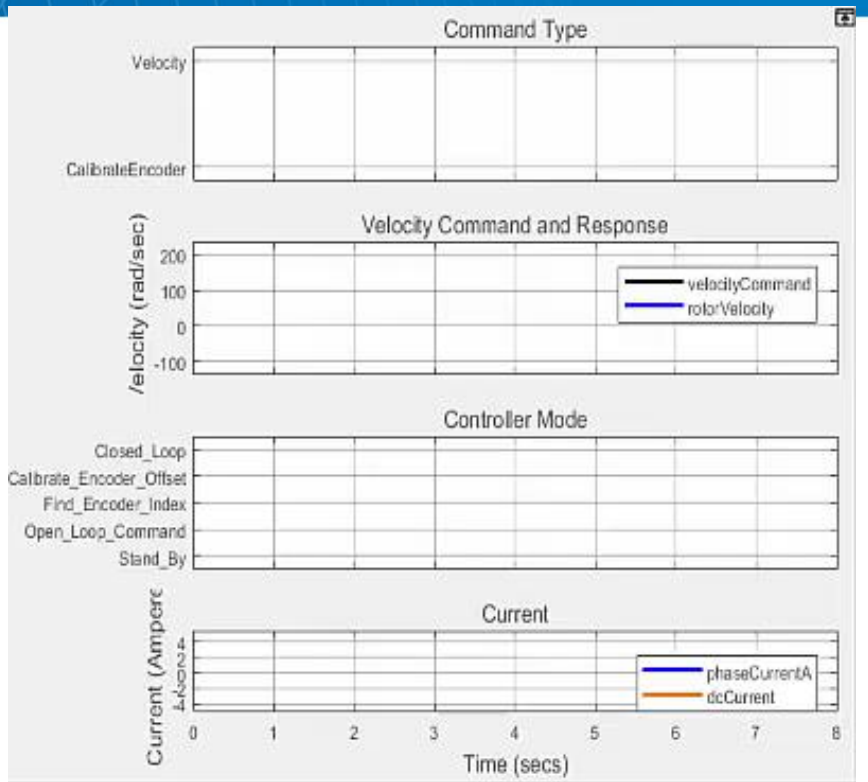
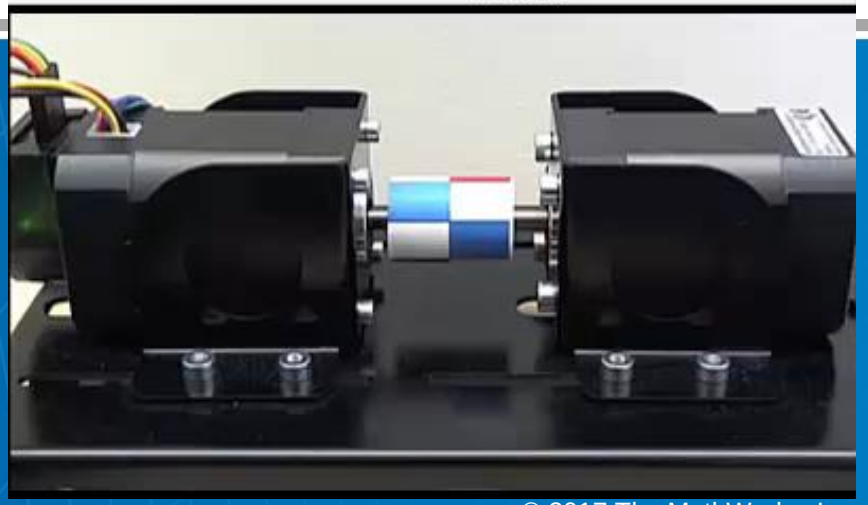
Targeting Motor Control Algorithms to System-on-Chip Devices

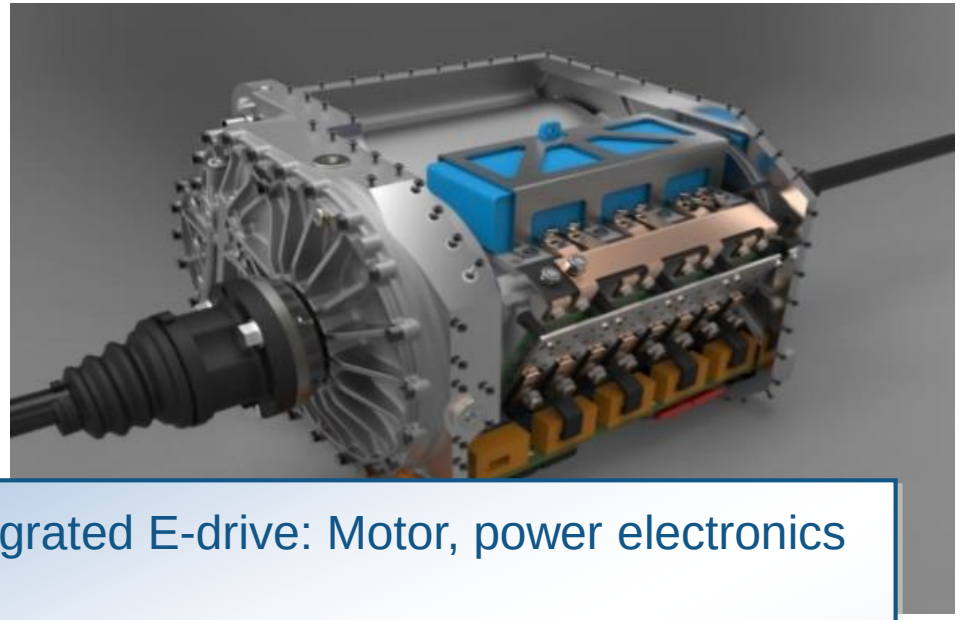
Eric Cigan





Punch Powertrain develops complex SoC-based motor control

- Powertrains for hybrid and electric vehicles
- Need to increase power density and efficiency at a reduced cost
  - Integrate motor and power electronics in the transmission
- New switched reluctance motor
  - Fast: 2x the speed of their previous motor
    - Target to a Xilinx® Zynq® SoC 7045 device
  - Complex: 4 different control strategies
- Needed to get to market quickly
- No experience designing FPGAs!



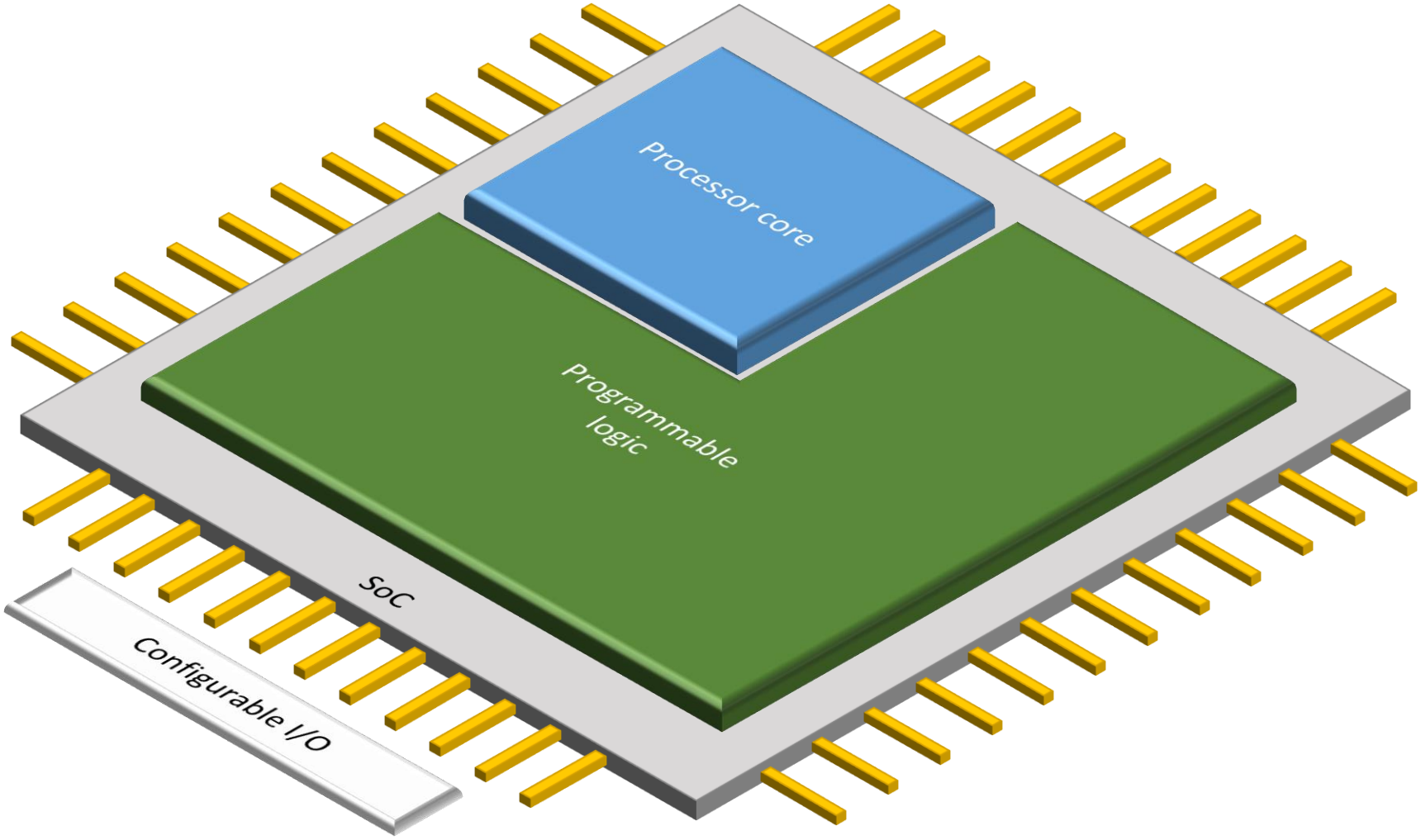
- ✓ Designed integrated E-drive: Motor, power electronics and software
- ✓ 4 different control strategies implemented
- ✓ Done in 1.5 years with 2FTE's
- ✓ Models reusable for production
- ✓ Smooth integration and validation due to development process – thorough validation before electronics are produced and put in the testbench

Link to video

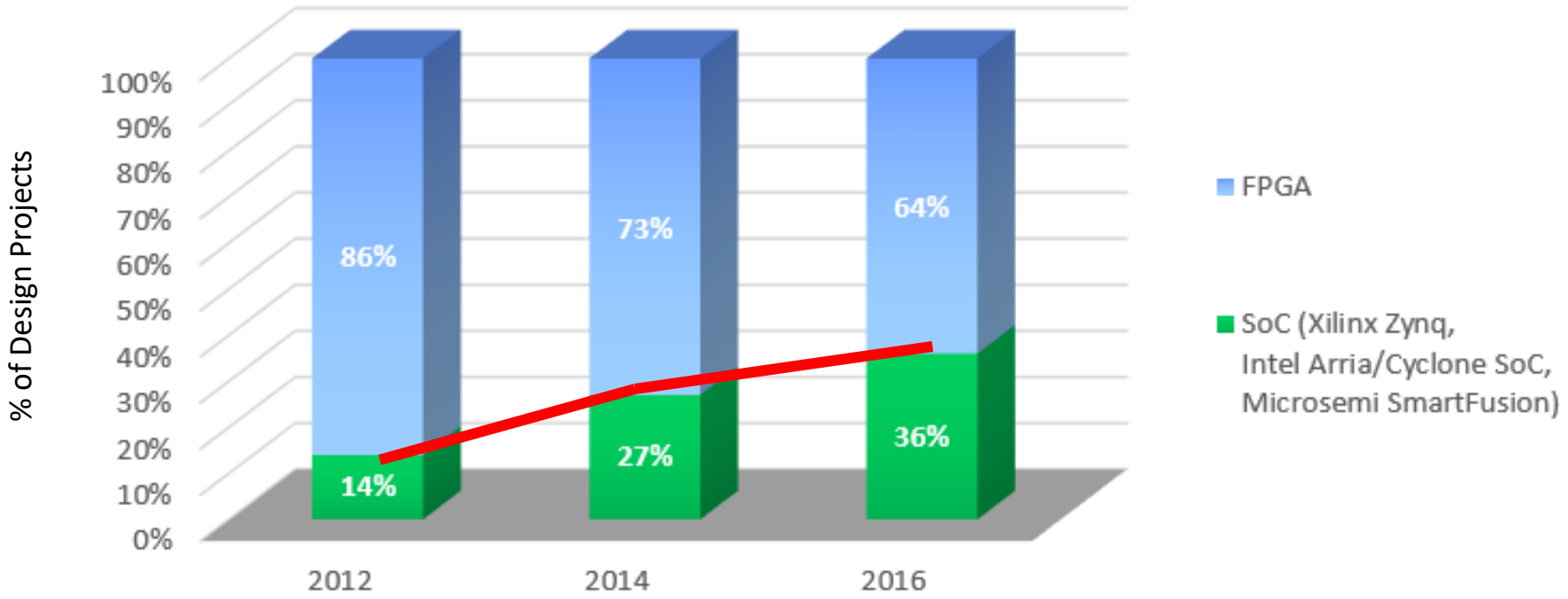
Key trend: Increasing demands from motor drives

- Advanced algorithms require faster computing performance.
  - Field-Oriented Control
  - Sensorless motor control
  - Vibration detection and suppression
  - Multi-axis control

What's an SoC?



Key Trend: SoCs are now used in 36% of new FPGA projects



Source: Wilson Research Group and Mentor Graphics, 2016 Functional Verification Study

Challenges in using SoCs for Motor and Power Control

- Integration requires collaboration
- Validation of design specifications with limits on access to test hardware
- How to make design decisions?

Why use Model-Based Design to develop motor control applications on SoCs?

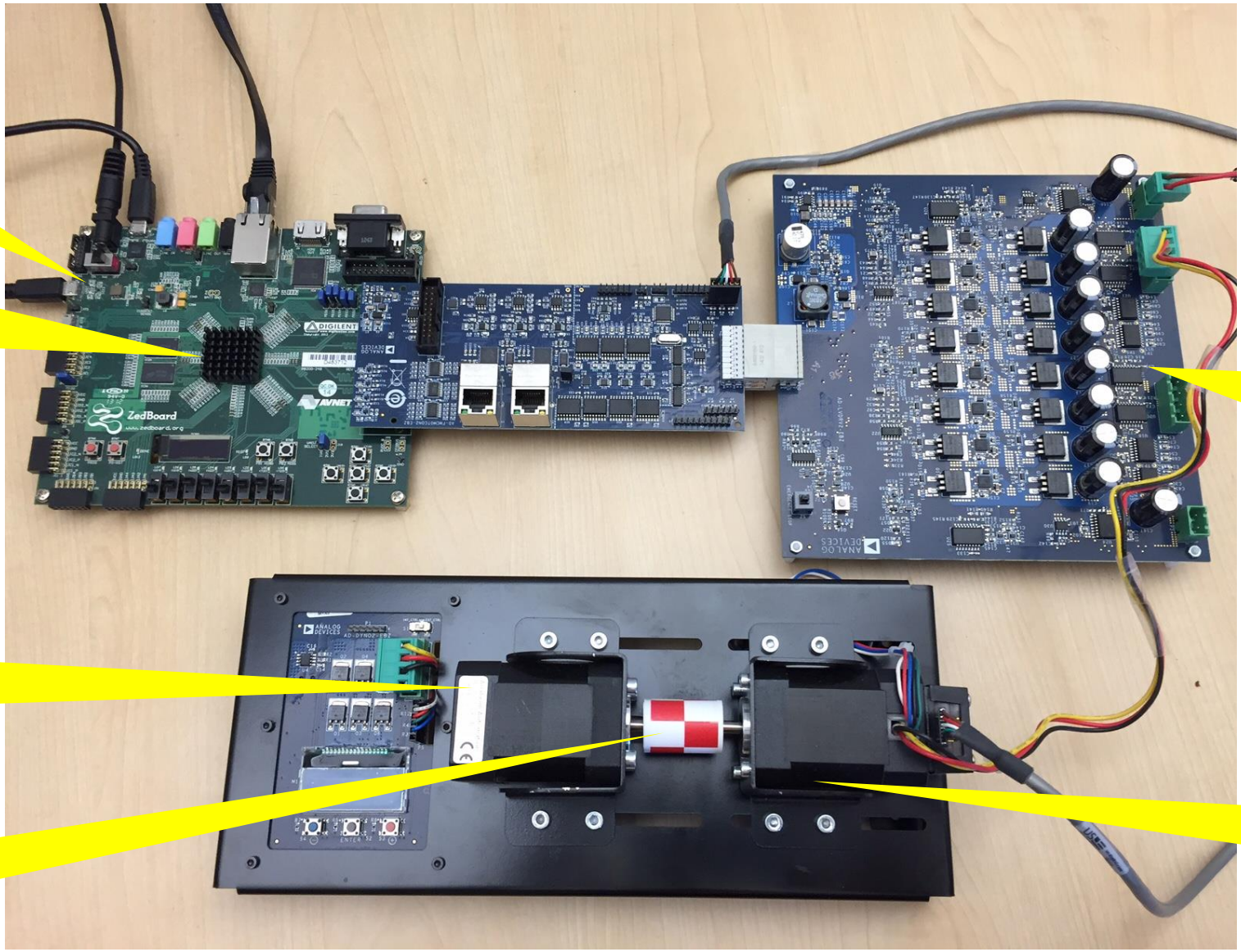
- Enables early validation of specifications using simulation months before hardware is available.
- Dramatically improves design team collaboration and designer productivity by using a single design environment.
- Reduces hardware testing time by 5x by shifting design from lab to the desktop

ZedBoard

Zynq SoC
(XC7Z020)

Load motor

Mechanical coupler



FMC module: control board + low-voltage board

Motor under test (with encoder)

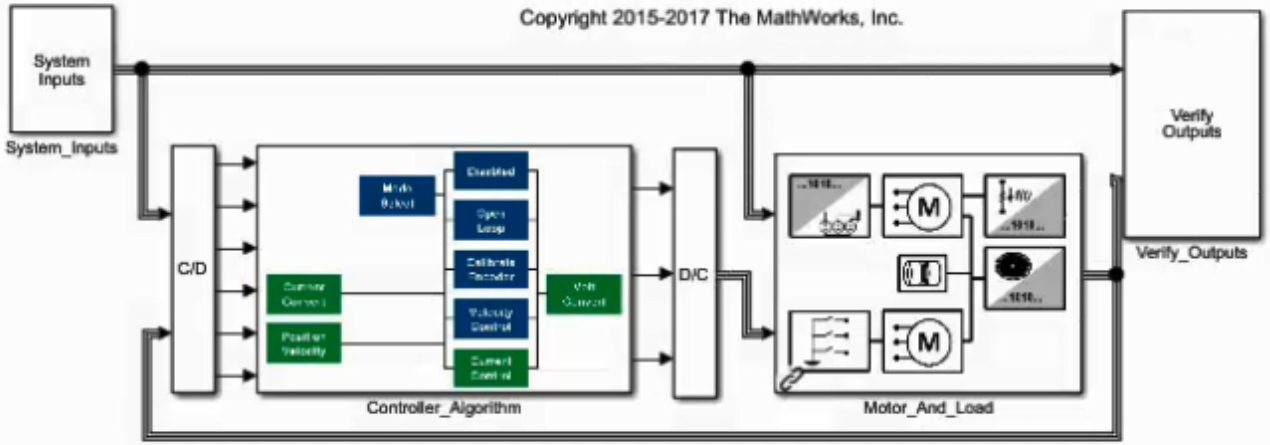
focZynqTestBench - Simulink

File Edit View Display Diagram Simulation Analysis Code Tools Help

focZynqTestBench

focZynqTestBench

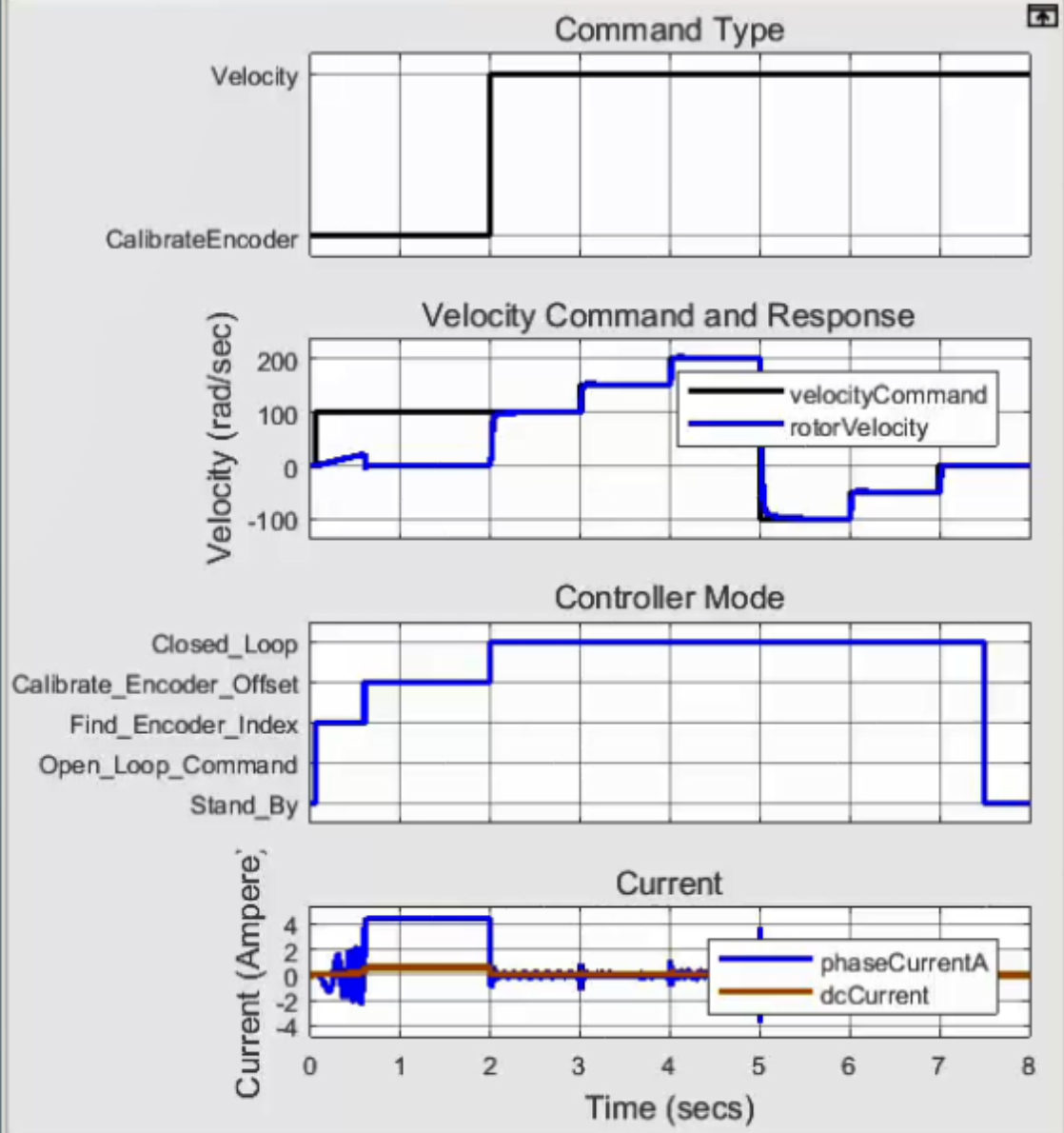
Field-Oriented Control of Velocity
Hardware/Software Test Bench



Ready | View 1 warning 68% | auto(ode45)

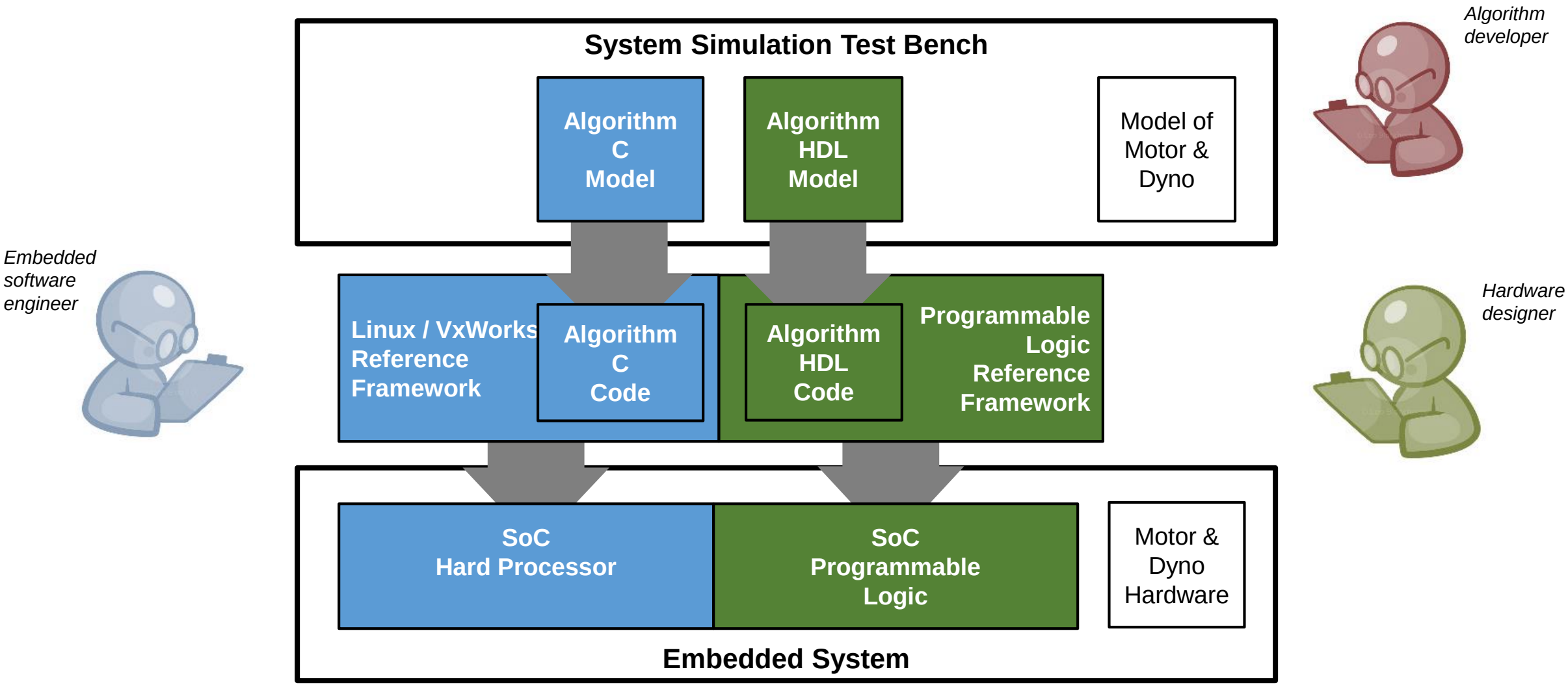
System_Response

File Tools View Simulation Help

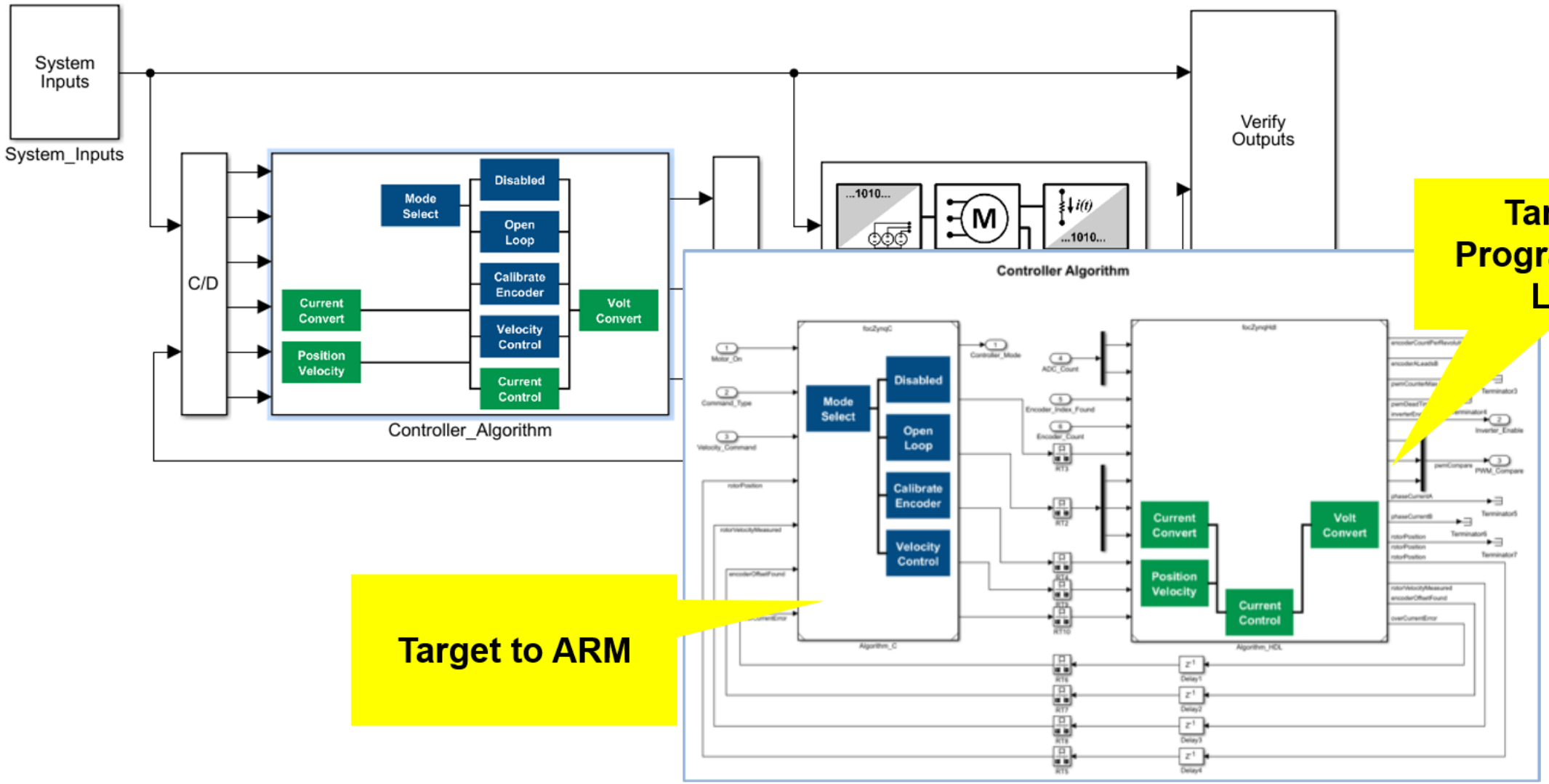


Ready | Sample based | T=8.000

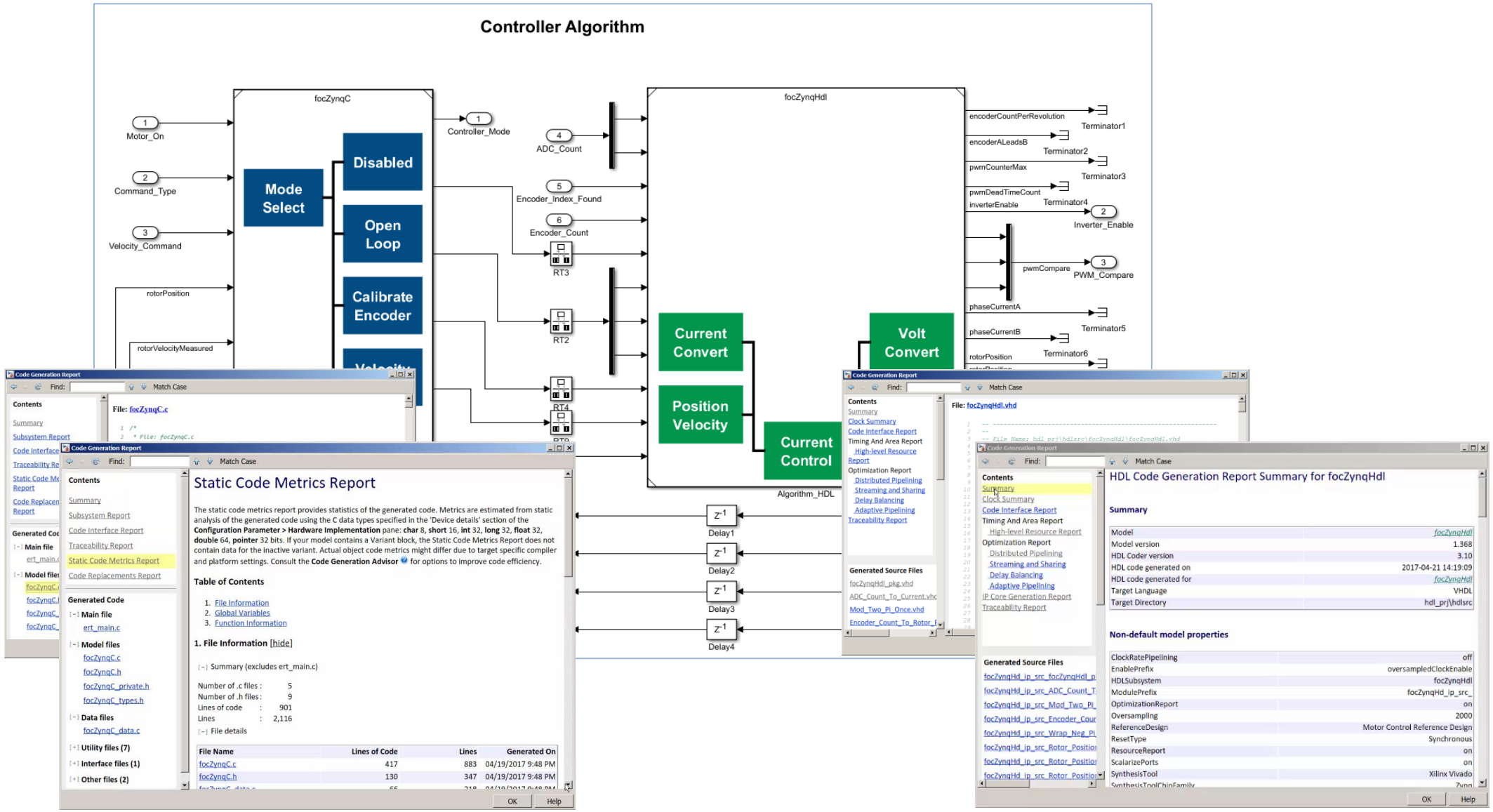
Conceptual workflow targeting SoCs

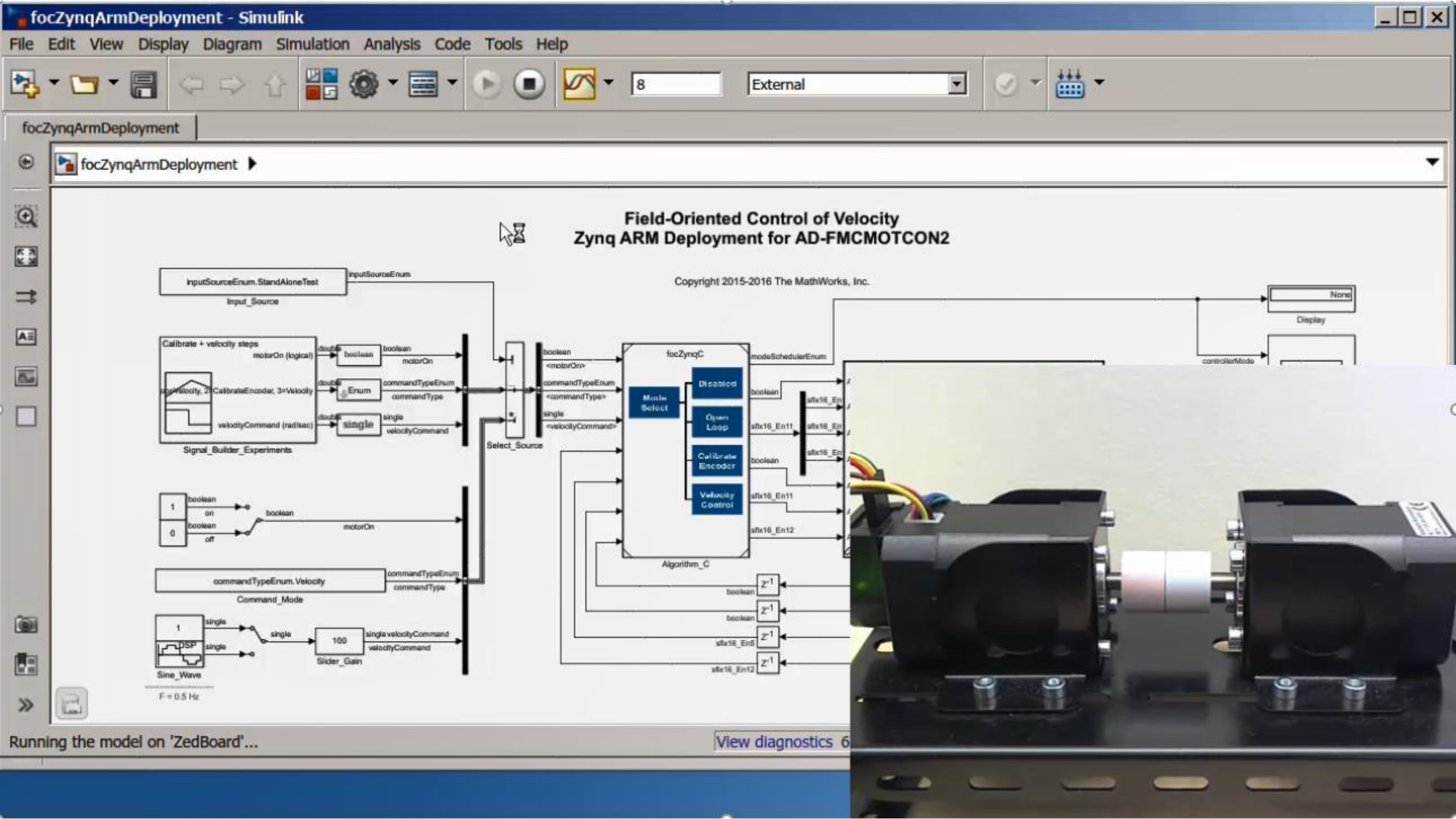


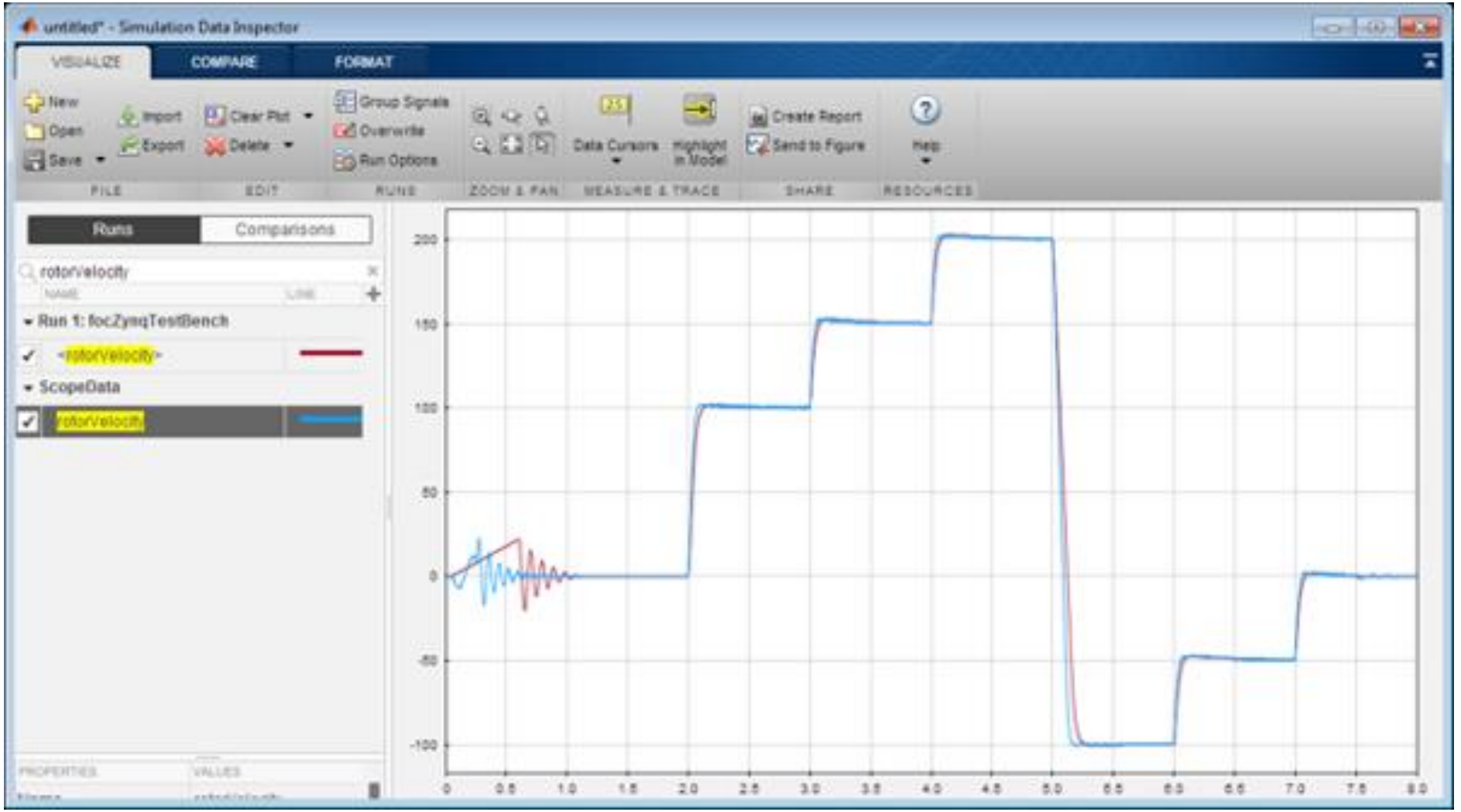
Hardware/software partitioning



Code Generation




focZynqArmDeployment - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
8
External
focZynqArmDeployment
Field-Oriented Control of Velocity
Zynq ARM Deployment for AD-FMCMOTCON2
Copyright 2015-2016 The MathWorks, Inc.
inputSourceEnum.StandAloneTest
Input_Source
Calibrate + velocity steps
Signal_Builder_Experiments
Select_Source
focZynqC
Mode Select
Disabled
Open Loop
Calibrate Encoder
Velocity Control
Algorithm_C
commandTypeEnum.Velocity
Command_Mode
Sine_Wave
Slider_Gain
Display
Running the model on 'ZedBoard'...
View diagnostics



Simulation

Hardware test

3T Develops Robot Emergency Braking System with Model-Based Design

Challenge

Design and implement a robot emergency braking system with minimal hardware testing

Solution

Model-Based Design with Simulink and HDL Coder to model, verify, and implement the controller

Results

- Cleanroom time reduced from weeks to days
- Late requirement changes rapidly implemented
- Complex bug resolved in one day

A SCARA robot.

“With Simulink and HDL Coder we eliminated programming errors and automated delay balancing, pipelining, and other tedious and error-prone tasks. As a result, we were able to easily and quickly implement change requests from our customer and reduce time-to-market.”

Ronald van der Meer

3T

Link to user story

Why use Model-Based Design to develop motor control applications on SoCs?

- Enables early validation of specifications using simulation months before hardware is available.
- Dramatically improves design team collaboration and designer productivity by using a single design environment.
- Reduces hardware testing time by 5x by shifting design from lab to the desktop

Learn More

- Get an in-depth demo in the Technology Showcase
  - New: see award-winning Native Floating Point in HDL Coder!
- Videos
  - HDL Coder: Native Floating Point
- Webinars
  - Prototyping SoC-based Motor Controllers on Intel SoCs with MATLAB and Simulink
  - How to Build Custom Motor Controllers for Zynq SoCs with MATLAB and Simulink
- Articles
  - How Modeling Helps Embedded Engineers Develop Applications for SoCs (MATLAB Digest)
  - MATLAB and Simulink Aid HW-SW Codesign of Zynq SoCs (Xcell Software Journal)
- Tutorials:
  - Define and Register Custom Board and Reference Design for SoC Workflow
  - Field-Oriented Control of a Permanent Magnet Synchronous Machine on SoCs