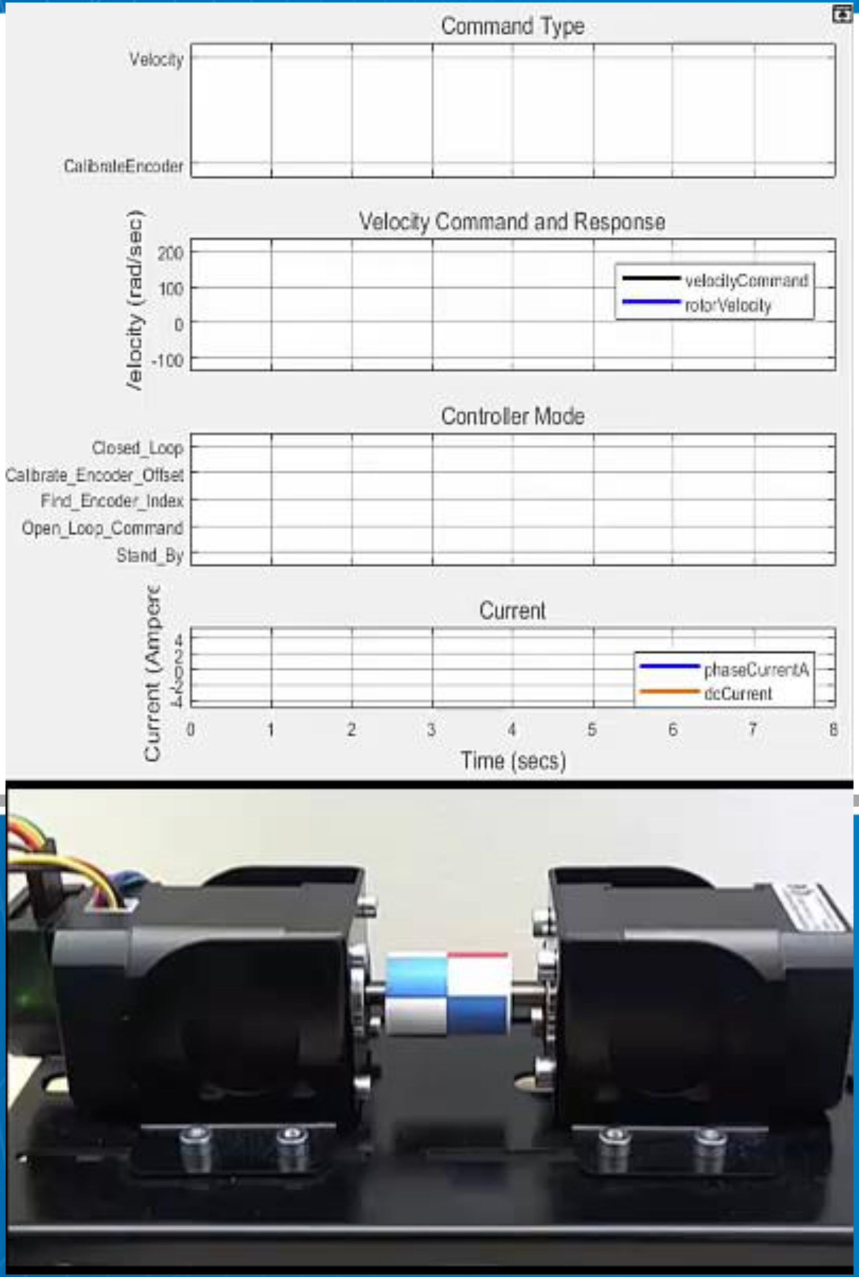
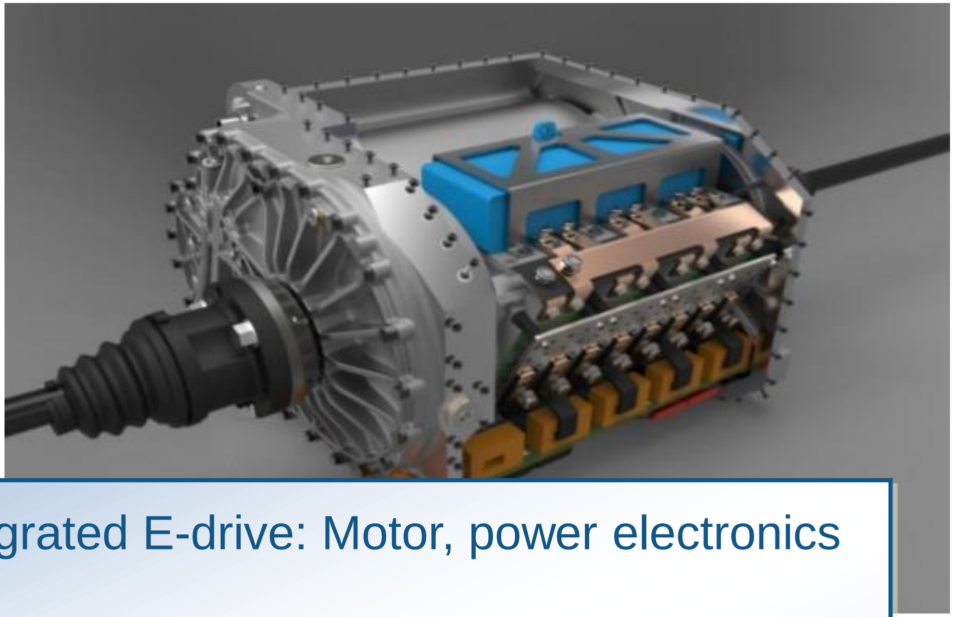
# Targeting Motor Control Algorithms to System-on-Chip Devices

**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!



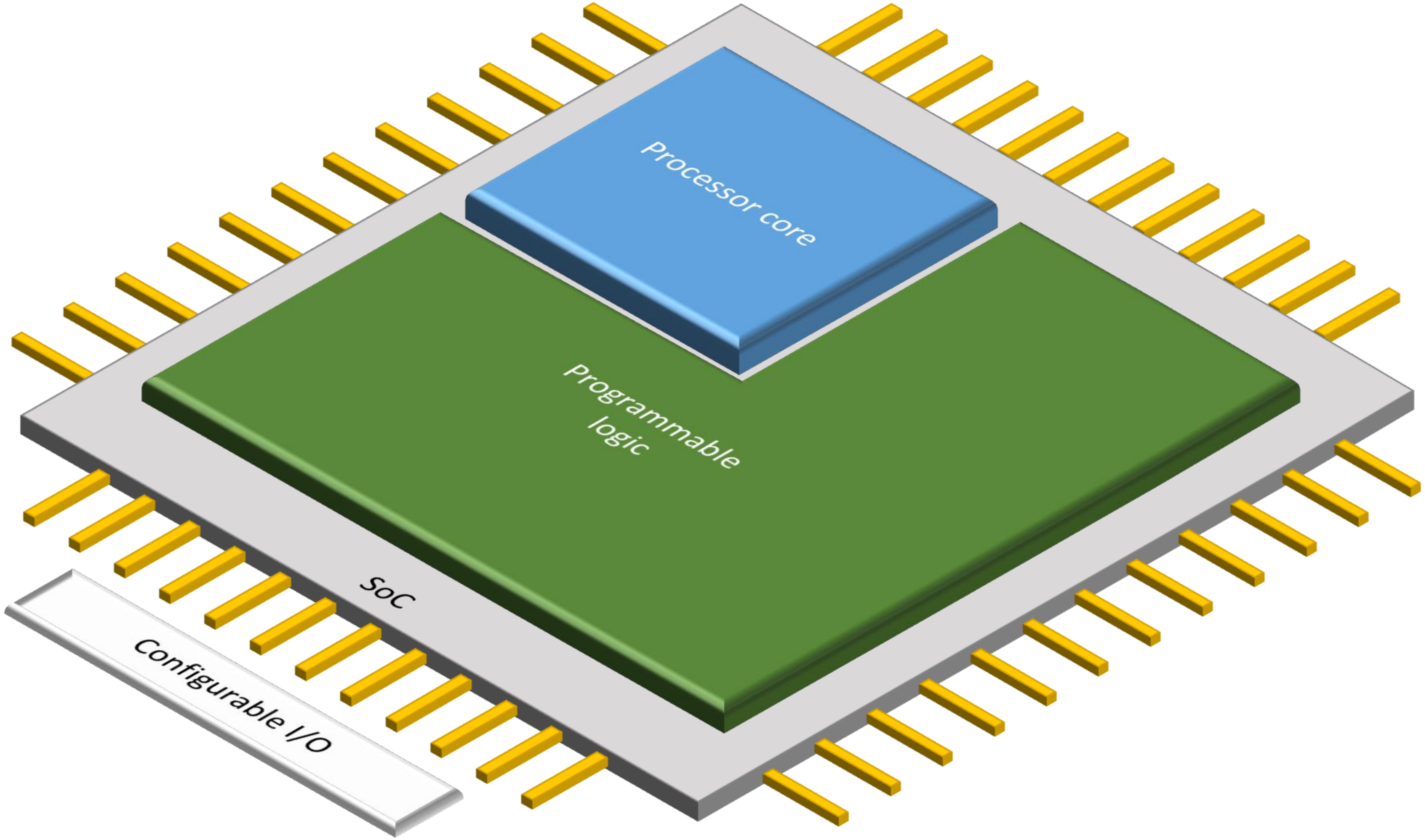
- ✓ Designed integrated E-drive: Motor, power electronics and software
- ✓ 4 different control strategies implemented
- ✓ Done in 1.5 years with 2FTE's
- ✓ Models reusable for production
- ✓ Smooth integration and validation due to development process – thorough validation before electronics are produced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

Link to video

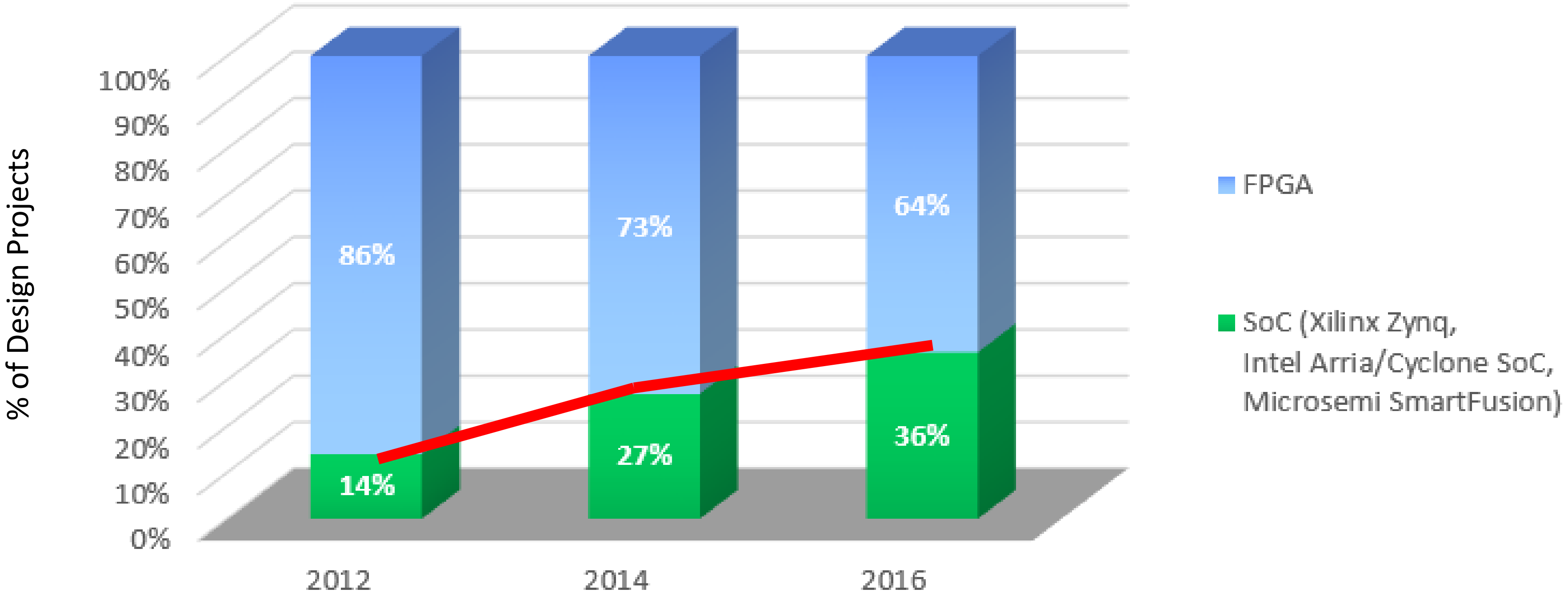
# Key trend: Increasing demands from motor drives

- 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

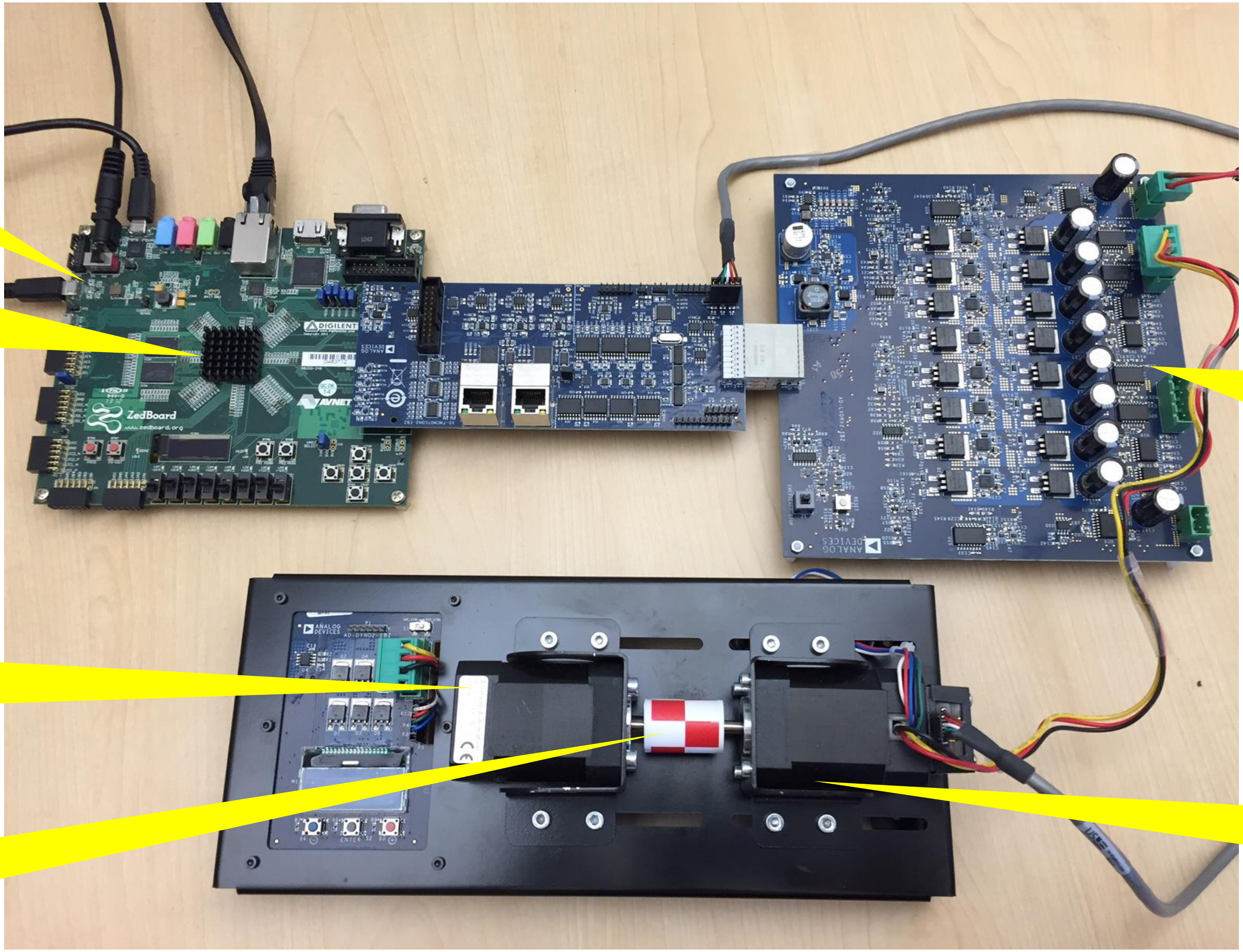
- How to make design decisions?

# 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

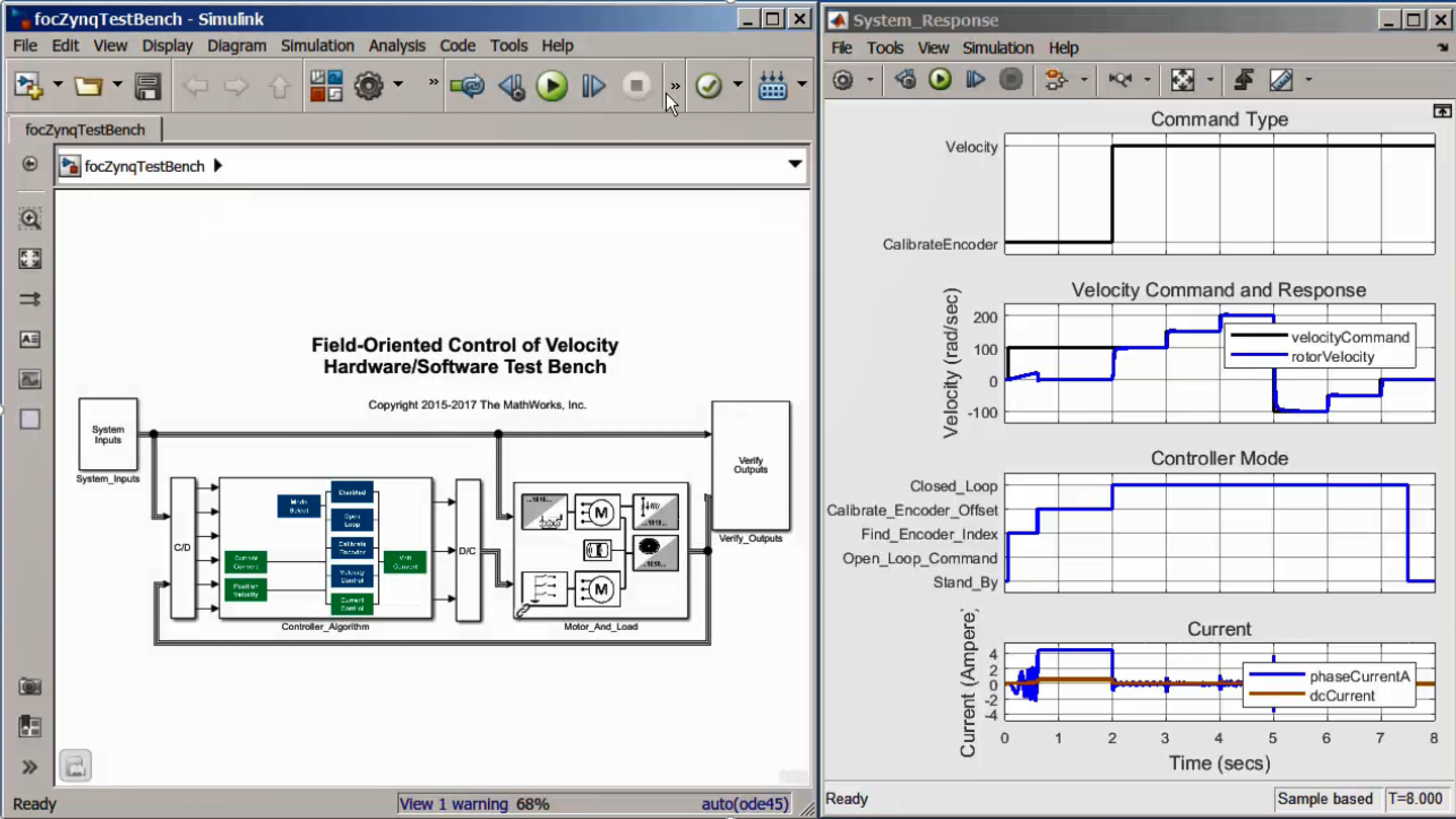
Zynq SoC
(XC7Z020)

FMC module:
control board +
low-voltage board

Load motor

Mechanical
coupler

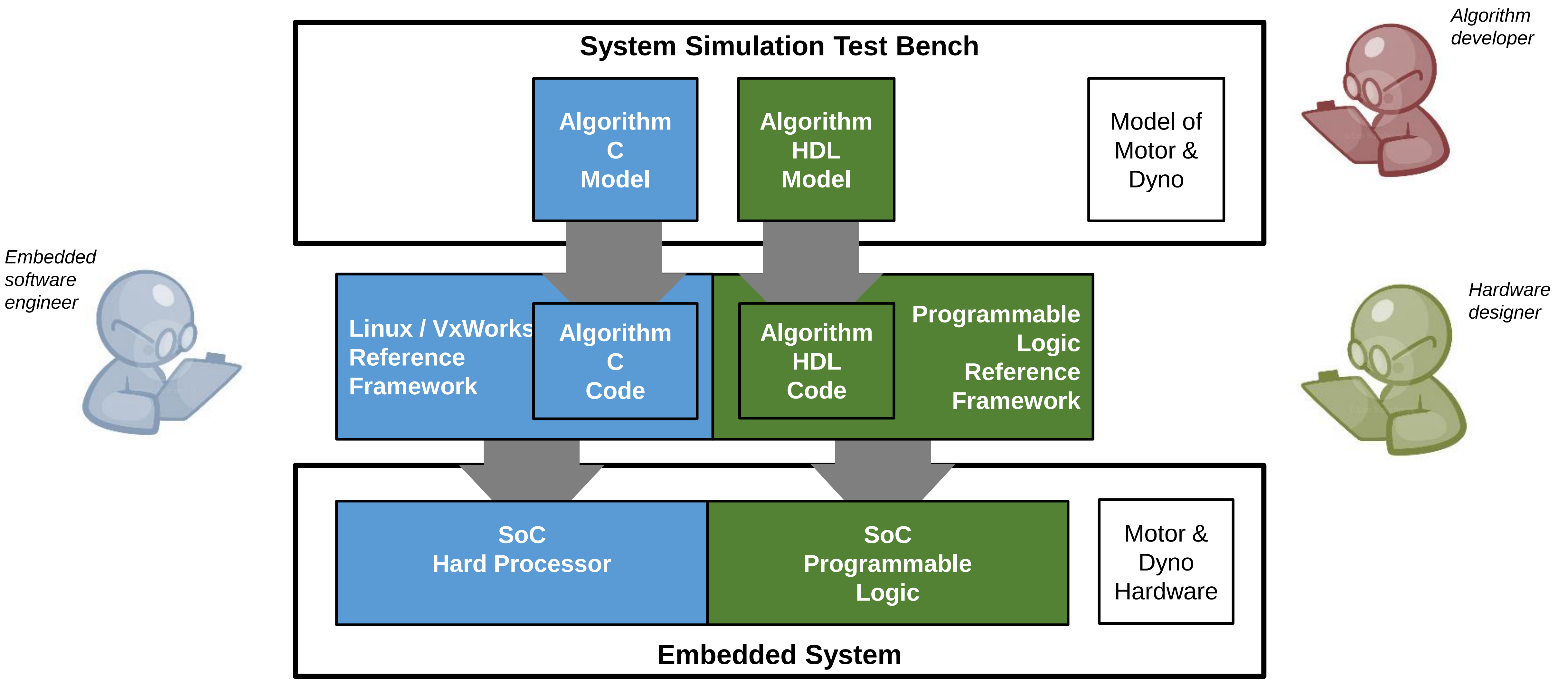
Motor under test
(with encoder)



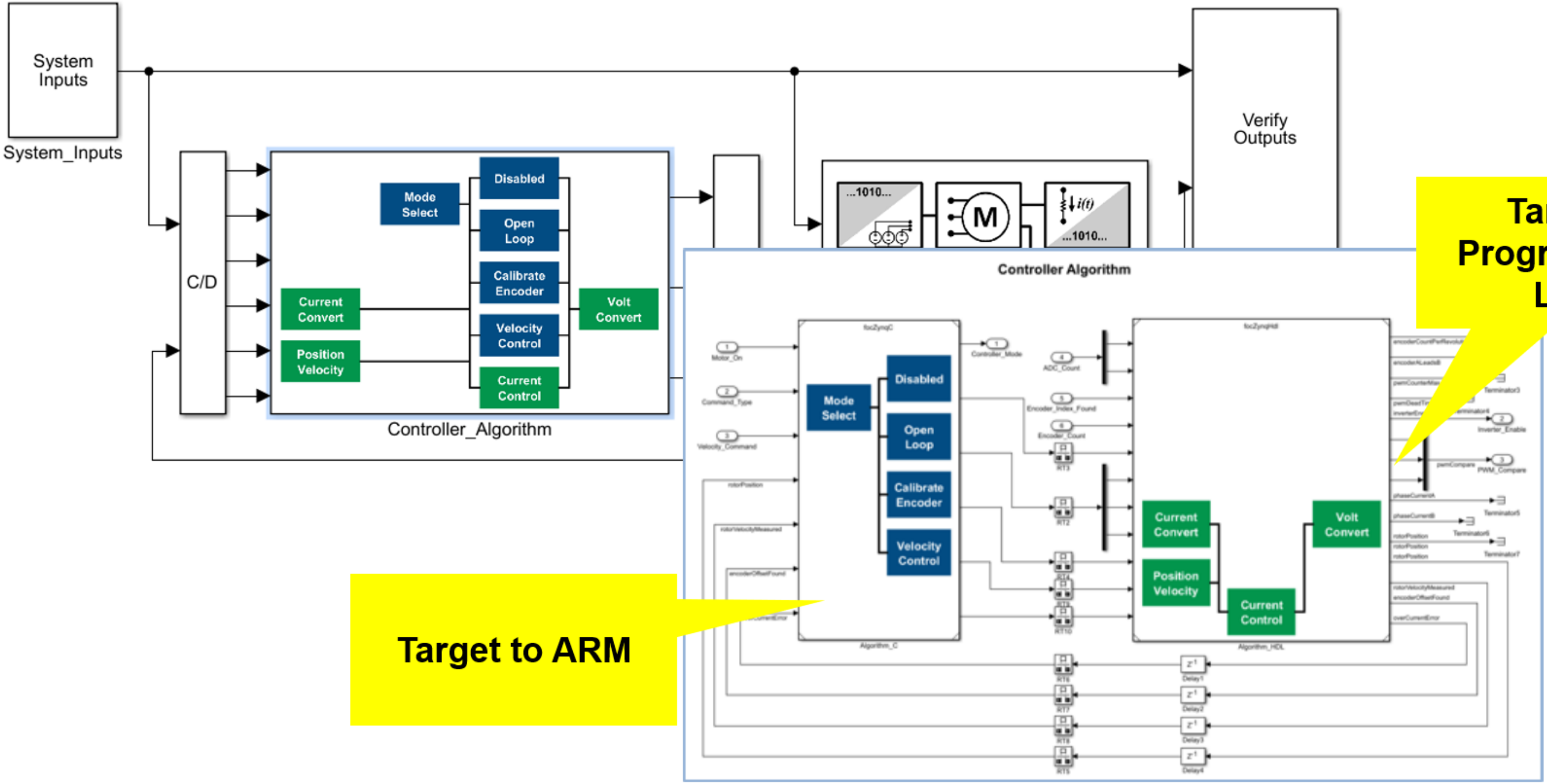
**Field-Oriented Control of Velocity**
**Hardware/Software Test Bench**

Copyright 2015-2017 The MathWorks, Inc.

# Conceptual workflow targeting SoCs



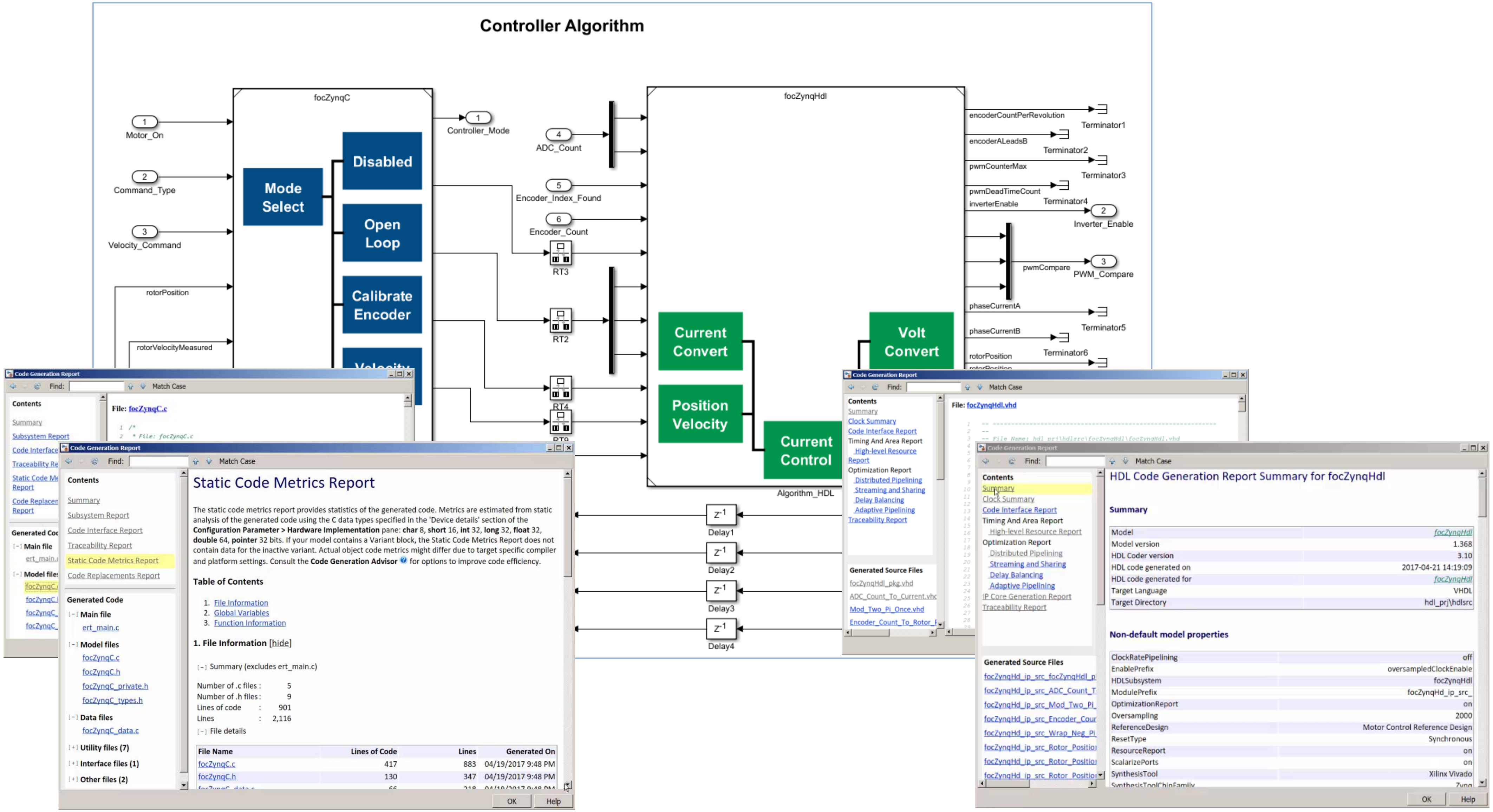
# Hardware/software partitioning



Target to Programmable Logic

Target to ARM

# Code Generation



```
/*
 * File: focZynqC.c
```

```
-- ----------------------------------------------------------------
--
-- File Name: hdl_prj\hdlsrc\focZynqHdl\focZynqHdl.vhd
```

## Static Code Metrics Report

The static code metrics report provides statistics of the generated code. Metrics are estimated from static analysis of the generated code using the C data types specified in the 'Device details' section of the **Configuration Parameter > Hardware Implementation** pane: **char** 8, **short** 16, **int** 32, **long** 32, **float** 32, **double** 64, **pointer** 32 bits. If your model contains a Variant block, the Static Code Metrics Report does not contain data for the inactive variant. Actual object code metrics might differ due to target specific compiler and platform settings. Consult the **Code Generation Advisor** for options to improve code efficiency.

**Table of Contents**

1. File Information
2. Global Variables
3. Function Information

**1. File Information** [hide]

[-] Summary (excludes ert_main.c)

| | | |
|---|---|---|
| Number of .c files | : | 5 |
| Number of .h files | : | 9 |
| Lines of code | : | 901 |
| Lines | : | 2,116 |

[-] File details

| File Name | Lines of Code | Lines | Generated On |
|---|---|---|---|
| focZynqC.c | 417 | 883 | 04/19/2017 9:48 PM |
| focZynqC.h | 130 | 347 | 04/19/2017 9:48 PM |

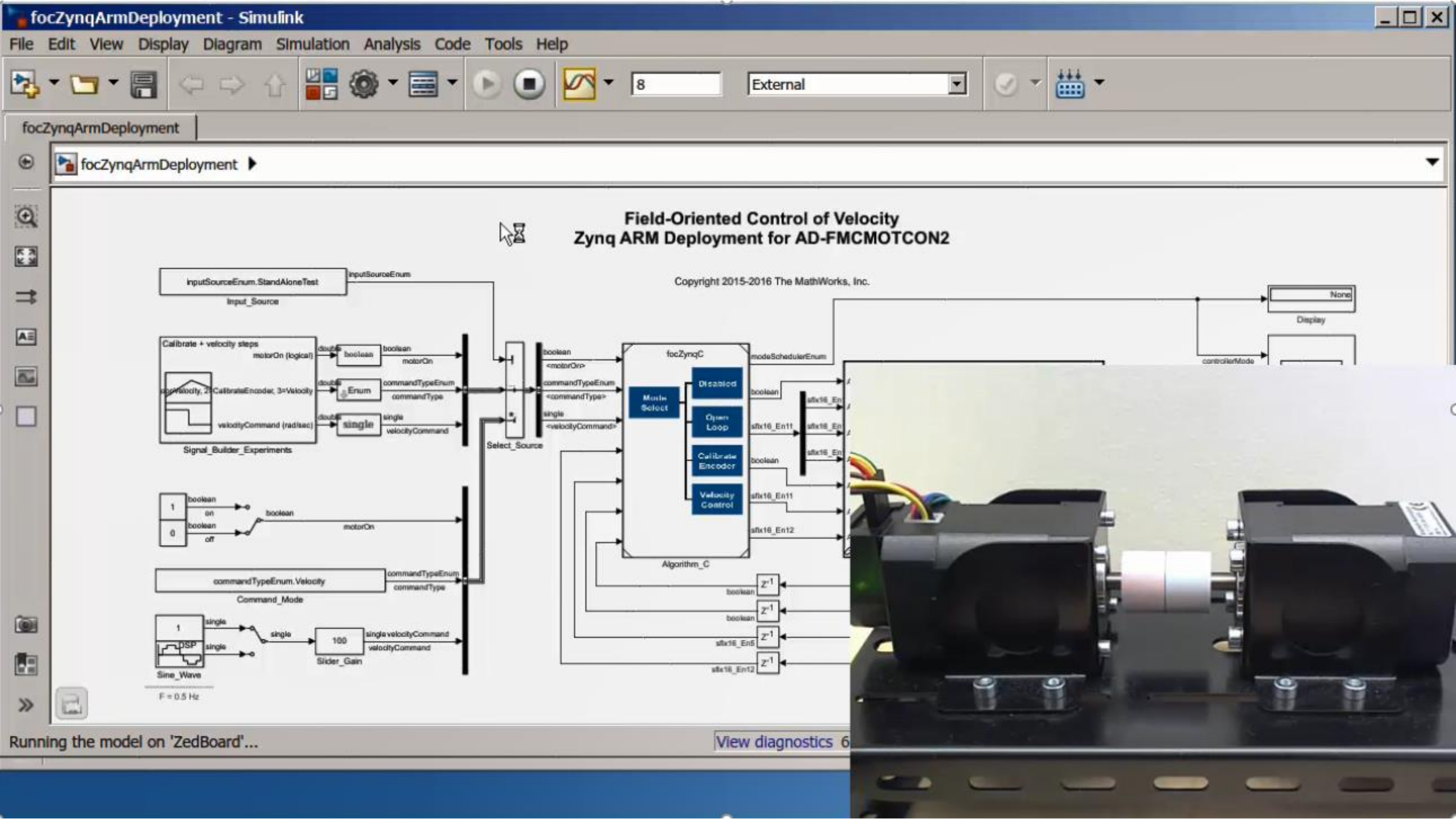
## HDL Code Generation Report Summary for focZynqHdl

**Summary**

| | |
|---|---|
| Model | focZynqHdl |
| Model version | 1.368 |
| HDL Coder version | 3.10 |
| HDL code generated on | 2017-04-21 14:19:09 |
| HDL code generated for | focZynqHdl |
| Target Language | VHDL |
| Target Directory | hdl_prj\hdlsrc |

**Non-default model properties**

| | |
|---|---|
| ClockRatePipelining | off |
| EnablePrefix | oversampledClockEnable |
| HDLSubsystem | focZynqHdl |
| ModulePrefix | focZynqHd_ip_src_ |
| OptimizationReport | on |
| Oversampling | 2000 |
| ReferenceDesign | Motor Control Reference Design |
| ResetType | Synchronous |
| ResourceReport | on |
| ScalarizePorts | on |
| SynthesisTool | Xilinx Vivado |
| SynthesisToolChipFamily | Zynq |

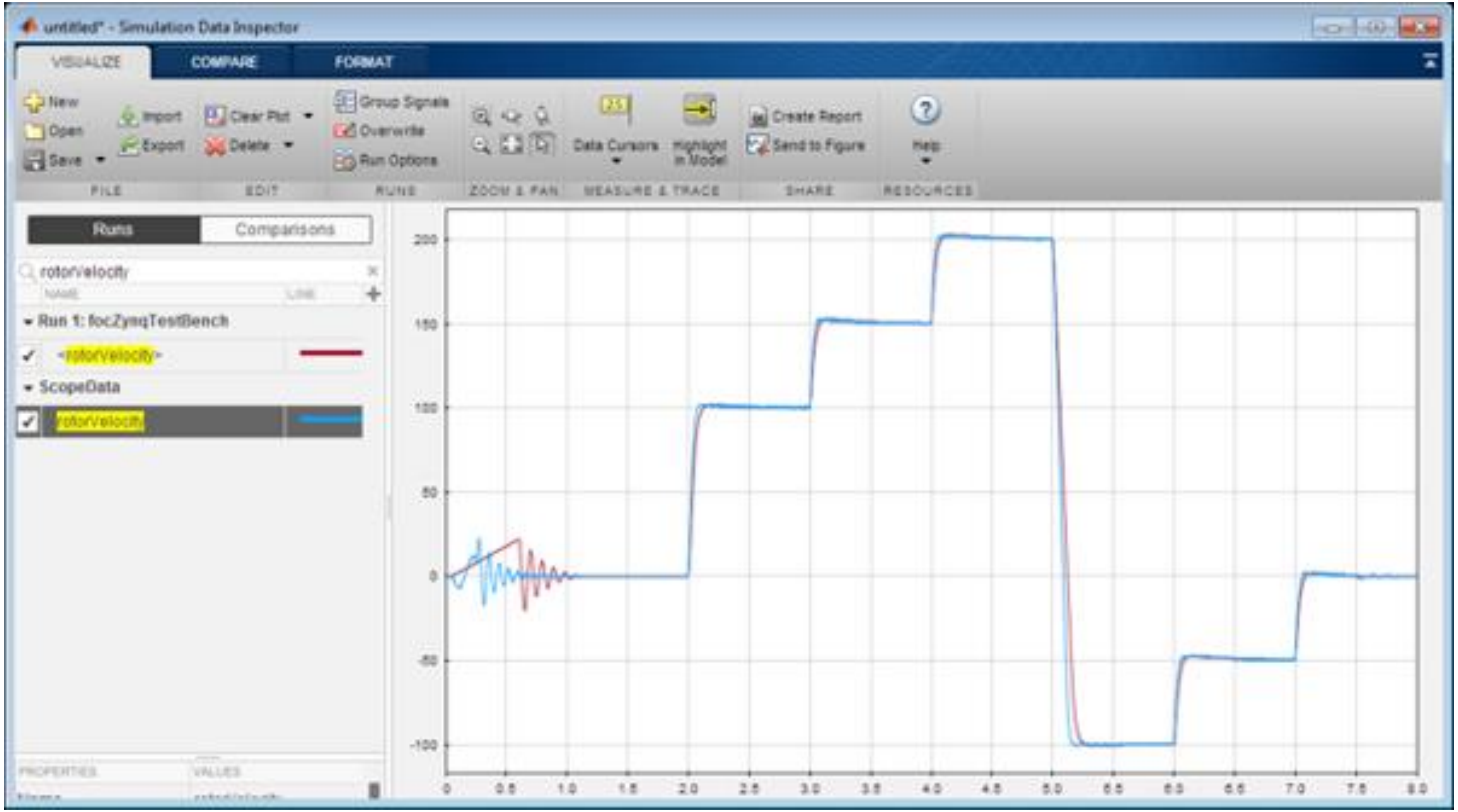


focZynqArmDeployment - Simulink

Field-Oriented Control of Velocity
Zynq ARM Deployment for AD-FMCMOTCON2

Copyright 2015-2016 The MathWorks, Inc.

Running the model on 'ZedBoard'...



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

MathWorks
58.426 volgers
16 d

MathWorks is honored to receive the Embedded World Award 2017 in the Tools Category for HDL Coder. http://owl.li/nBzd309XYxW

288 interessant • 6 commentaren