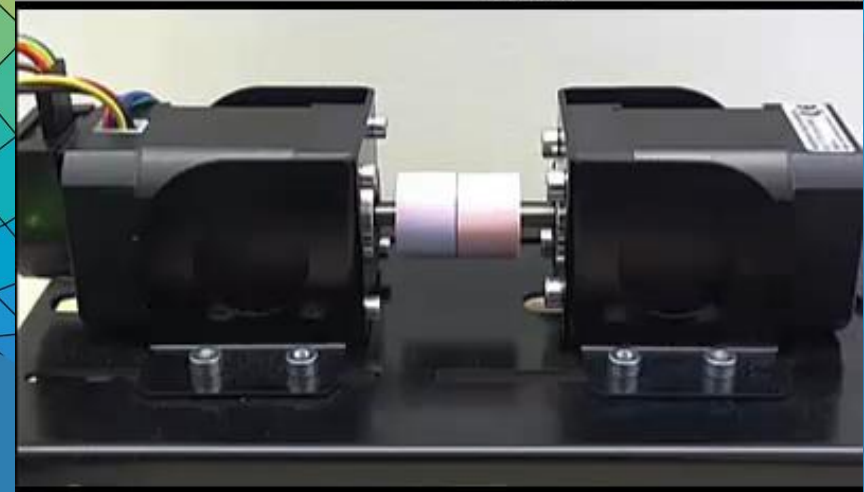
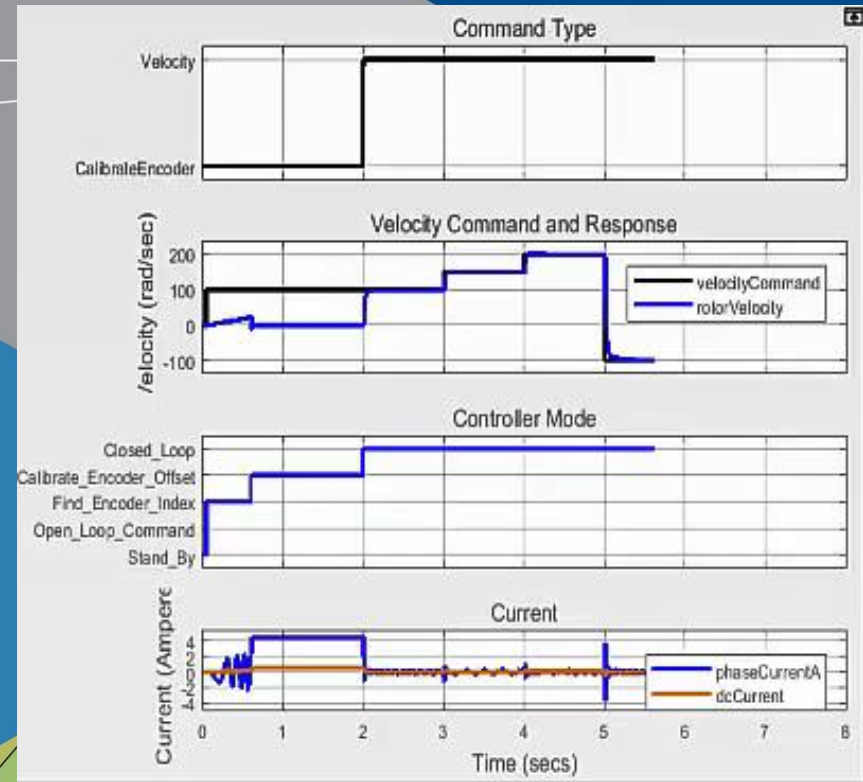


MATLAB EXPO 2017

Motor Controls Implementation on Systems-On-Chip

Jorik Caljouw

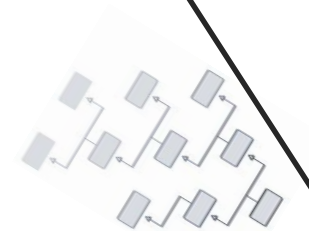
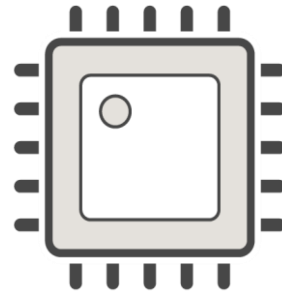


Key Takeaways

Meet stringent requirements
and reduce costs



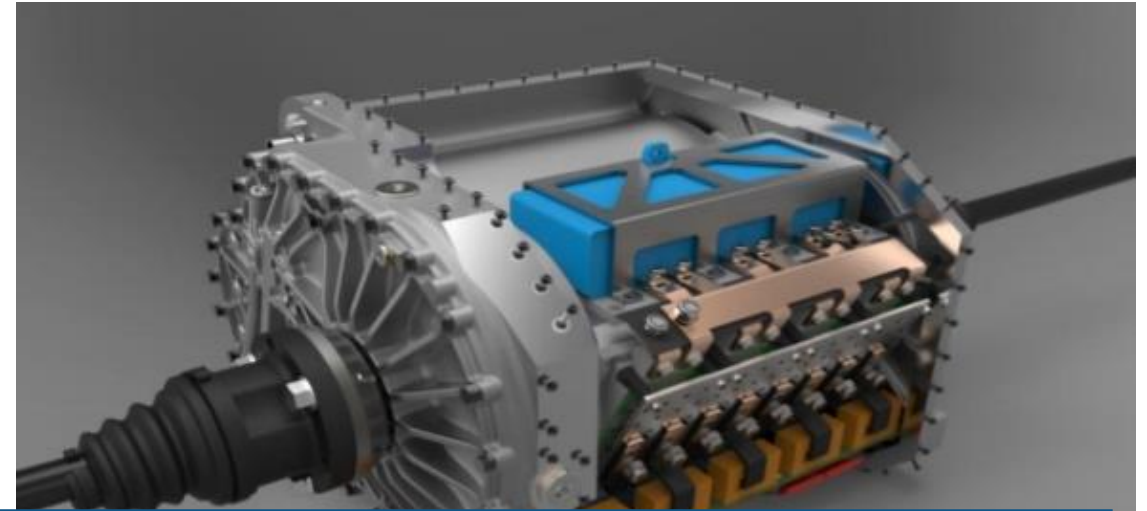
Reduce hardware testing
time up to 5x



Manage design complexity and improve team collaboration

Punch Powertrain develops complex SoC-based motor control

- Powertrains for hybrid and electric vehicles
- Hardware choice through simulations
- Traditional microcontroller too slow
- No experience designing FPGAs!

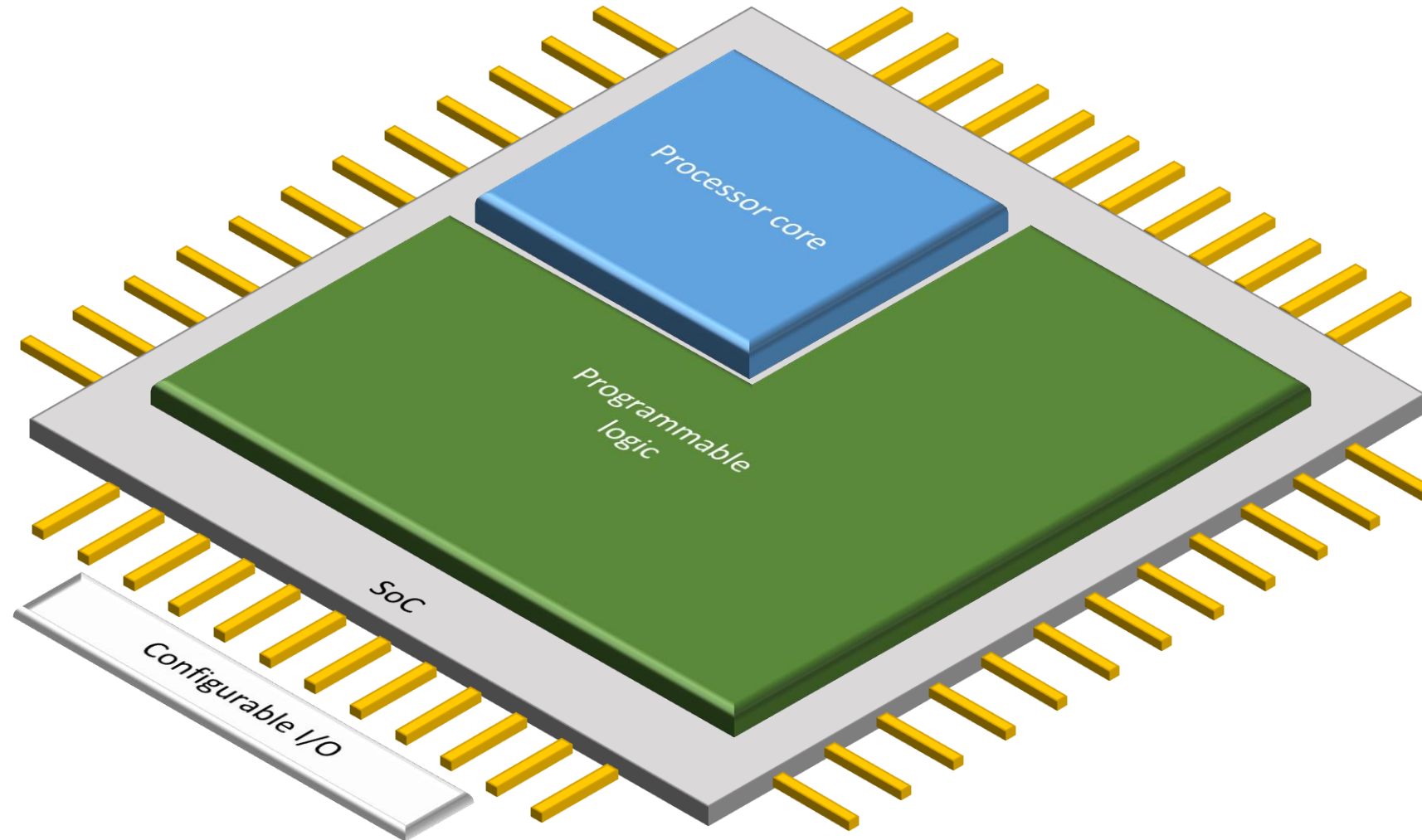


- ✓ Designed integrated E-drive: Motor, power electronics and software
- ✓ 4 different control strategies implemented
- ✓ Done in 1.5 years with 2FTE's
- ✓ Models reusable for production
- ✓ Smooth integration and validation due to development process

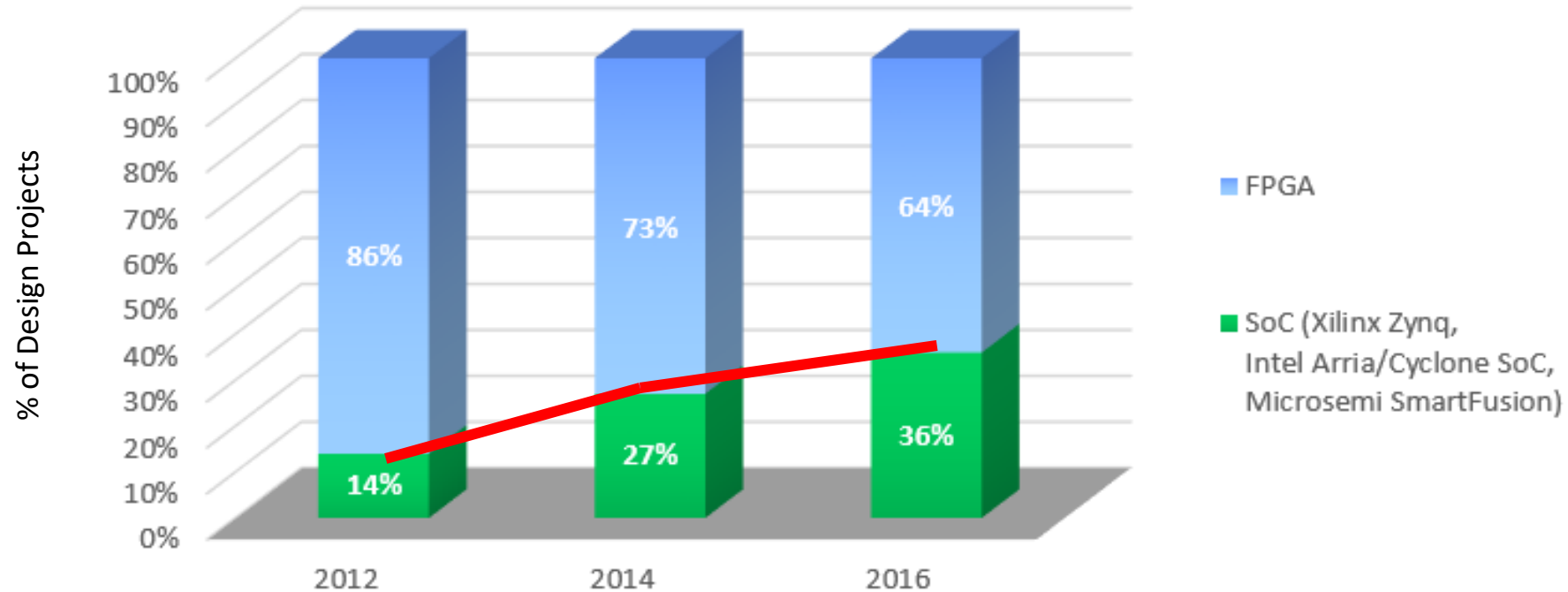
Key trend: Increasing demands from motor drives



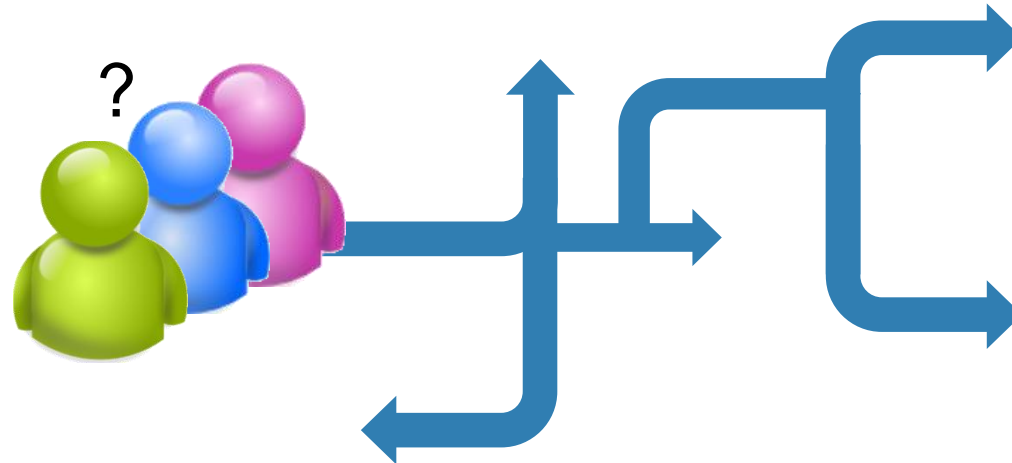
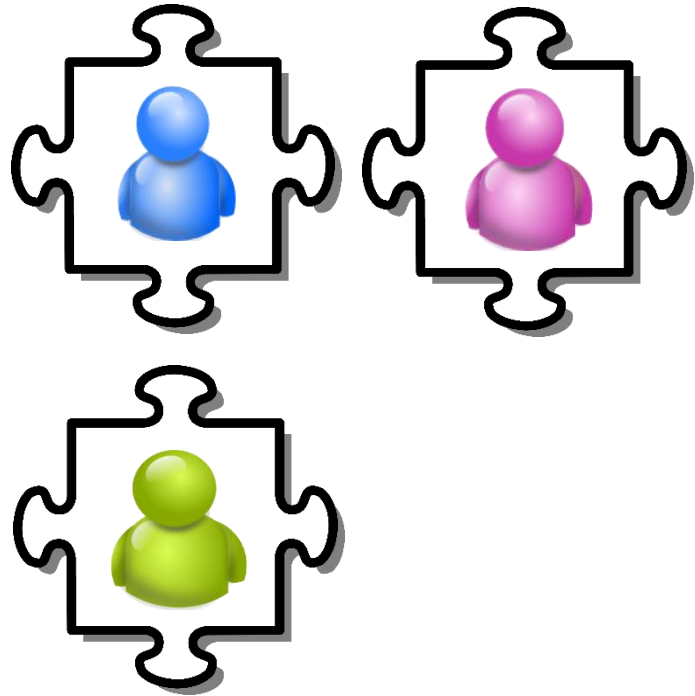
Systems-on-Chip for motor control



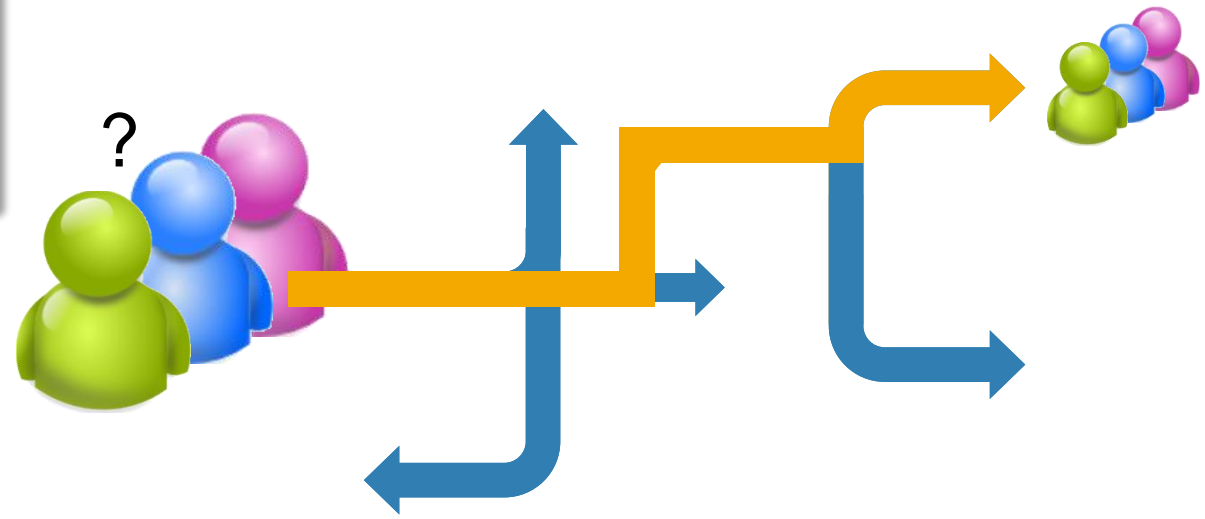
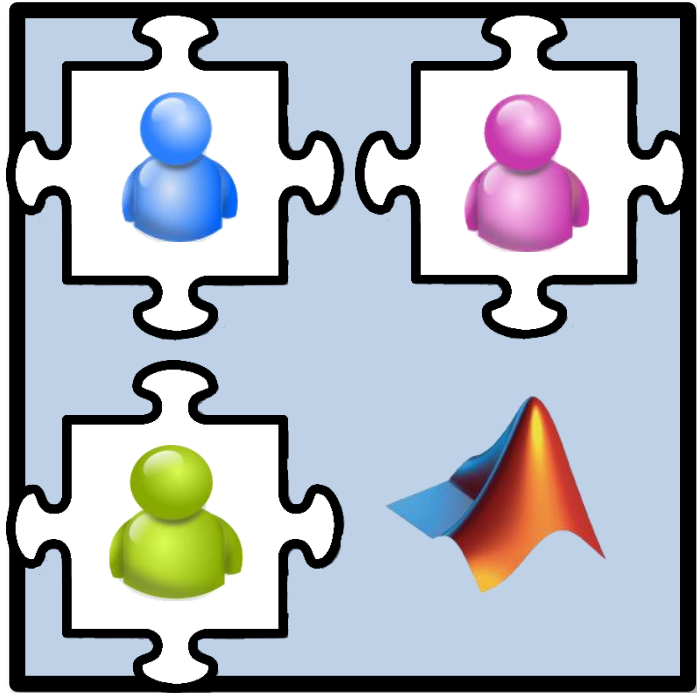
Key Trend: SoCs are now used in 36% of new FPGA projects



Challenges in using SoCs for Motor and Power Control



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



ZedBoard

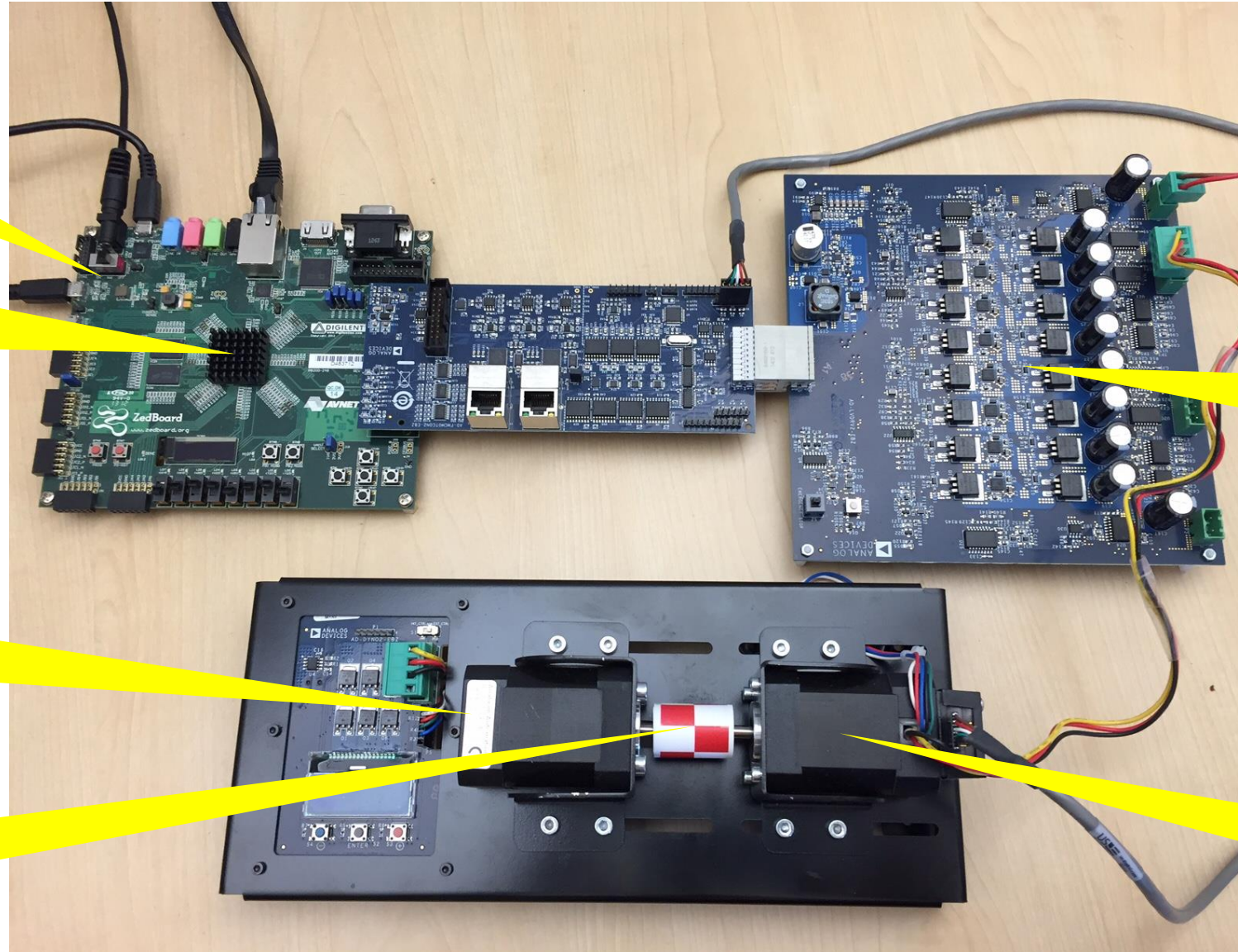
**Zynq SoC
(XC7Z020)**

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

Field-Oriented Control of Velocity Hardware/Software Test Bench

Copyright 2015-2017 The MathWorks, Inc.

Running View diagnostics 68% T=6.239 auto(ode45)

System_Response

File Tools View Simulation Help

Command Type

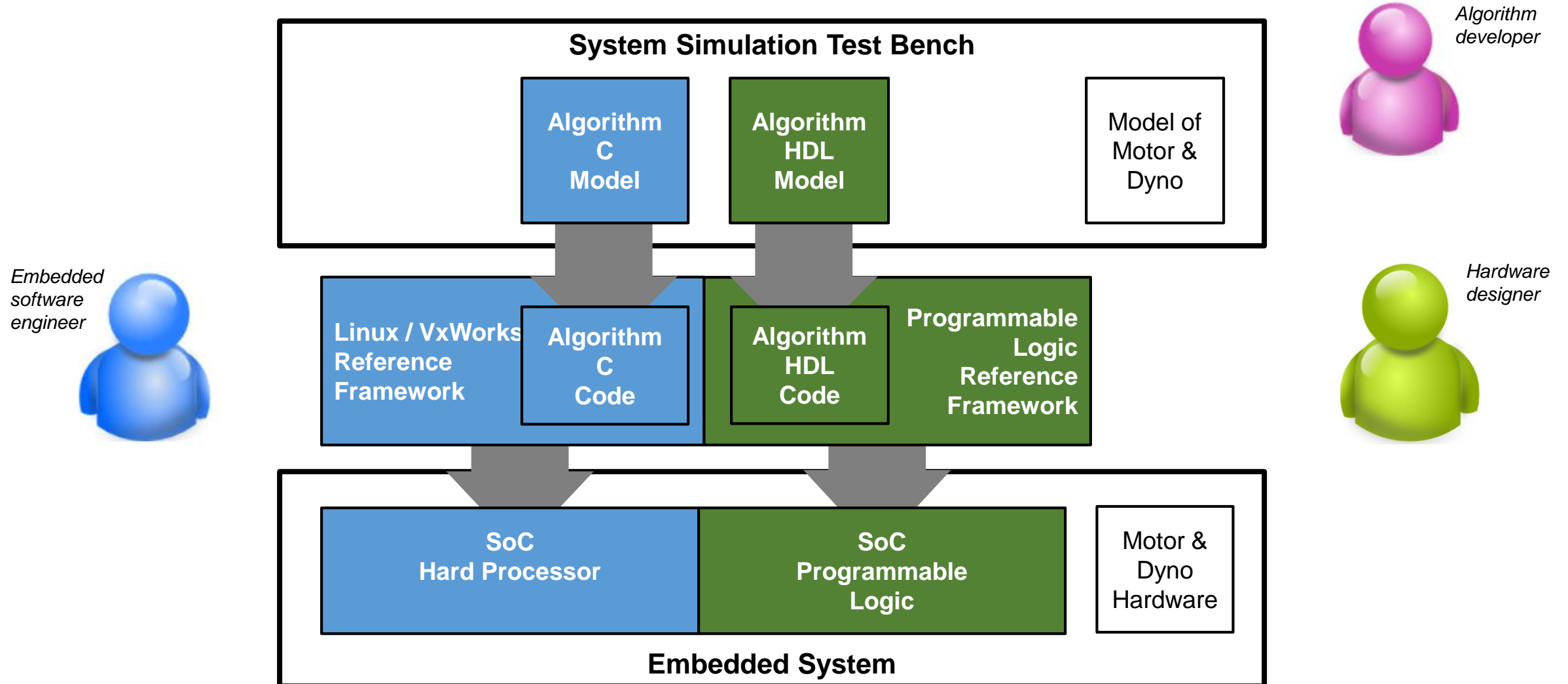
Velocity Command and Response

Controller Mode

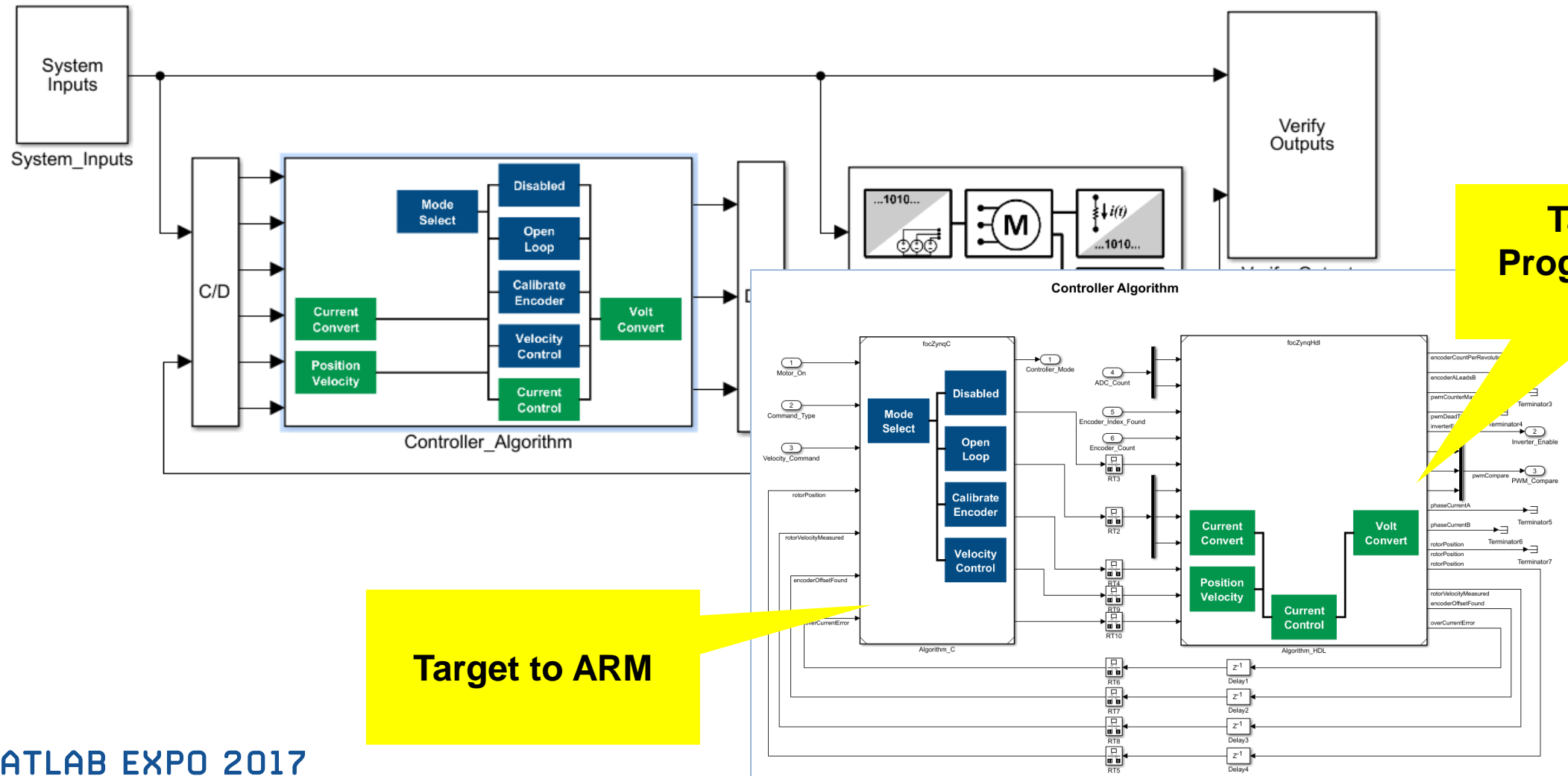
Current

Running Sample based T=6.220

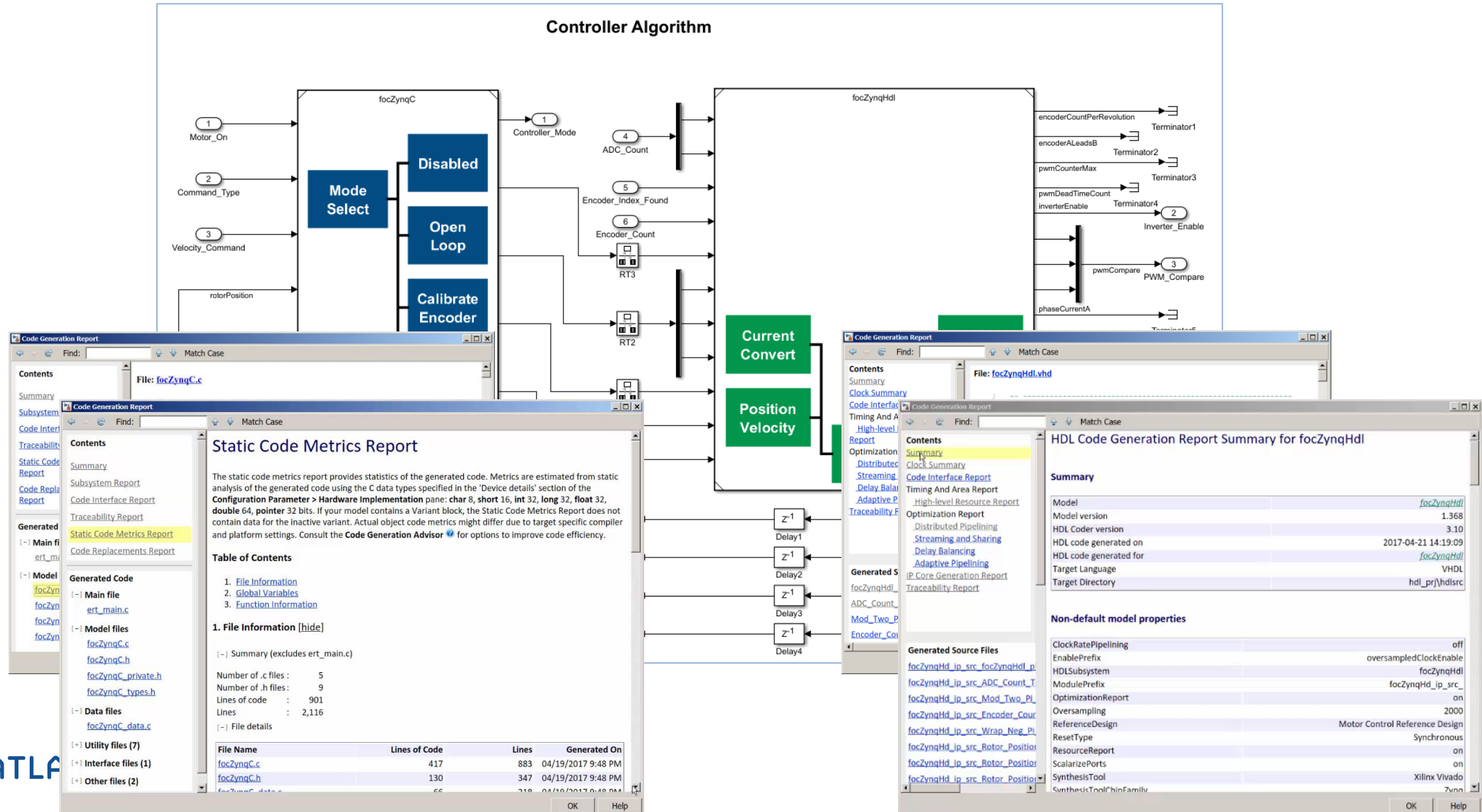
Conceptual workflow targeting SoCs



Hardware/software partitioning



Code Generation



Simulation Data Inspector - untitled*

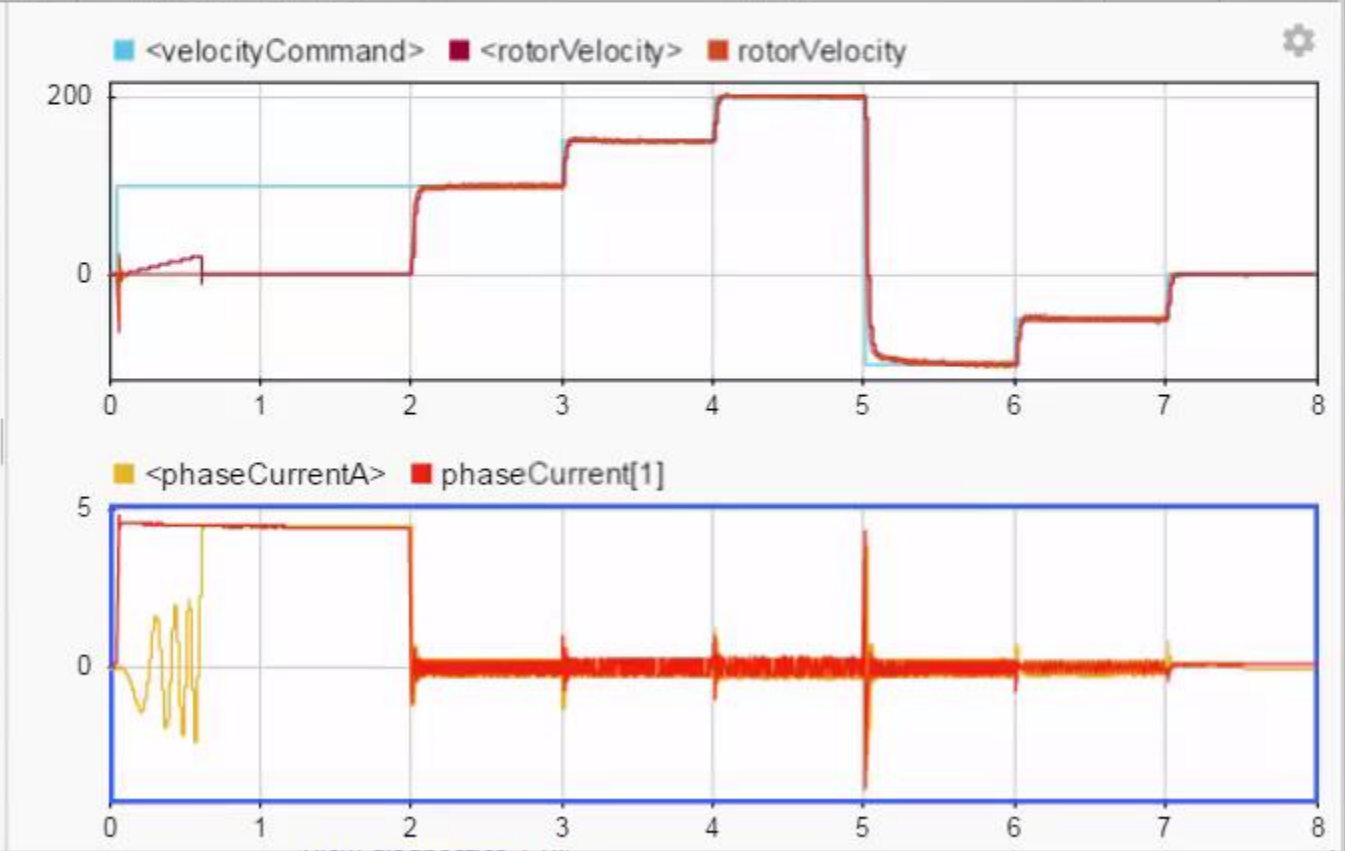
Visualize toolbar with icons for New, Import, Open, Export, Save, Preferences, Delete, Data Cursors, Highlight in Model, Subplots, Clear Subplot, Legend, Saved Views, SHARE, and RESOUR.

Inspect panel with 'Filter Signals' search bar and a list of signals:

NAME	LINE
<input type="checkbox"/> controllerMode	---
<input checked="" type="checkbox"/> phaseCurrent[1]	---
<input type="checkbox"/> phaseCurrent[2]	---
<input type="checkbox"/> rotorPosition	---
<input type="checkbox"/> rotorVelocity	---
▼ Run 6: focZynqArmDeployment	
<input type="checkbox"/> qCurrentCommand	---

Properties table for phaseCurrent[1]:

PROPERTIES	VALUES
Name	phaseCurrent[1]
Line	---



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

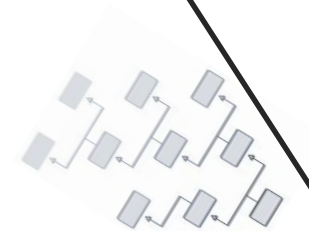
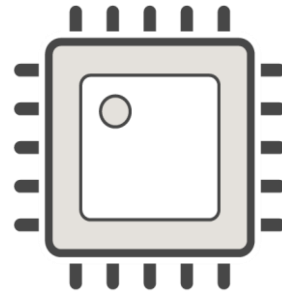
3T

Key Takeaways

Meet stringent requirements
and reduce costs



Reduce hardware testing
time up to 5x



Manage design complexity and improve team collaboration

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](#)

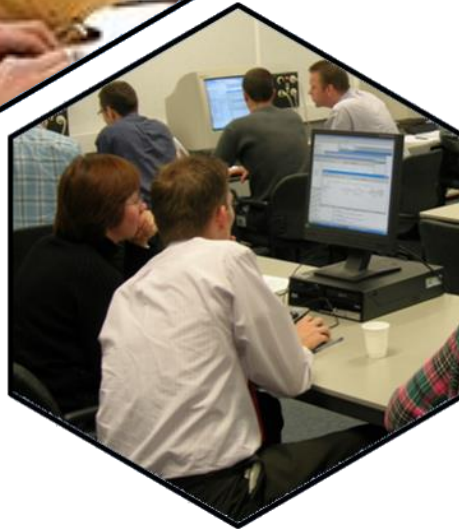


How to get started?

Public



On-Site



- Embedded Systems
- FPGA Design
- Xilinx Zynq SoCs

MATLAB[®]

Data Analytics

Data Processing and Visualization
 Statistics
 Machine Learning
 Optimization Techniques
 Parallel Computing

Application Development

Programming Techniques
 Building Interactive Applications
 Object-Oriented Programming

Code Generation

MATLAB Coder
 Interfacing with C-code

Application-Specific

Control System Design
 Signal Processing
 Communication Systems
 LTE Systems

Computational Finance

Risk Management
 Time-Series Modelling

Signal Processing

Using MATLAB
 Using Simulink

Image and Video Processing

Image Processing
 Computer Vision

SIMULINK[®]

Model-Based Design

Implementing MBD Workflow
 Model Management and Architecture
 Verification and Validation

Stateflow[®]

Event-Based Modeling

Simscape[™]

General Simscape[™]
 Simscape Multibody[™]
 Simscape Driveline[™]
 Simscape Fluids[™]
 Simscape Power Systems[™]

Code Generation

Rapid Prototyping and HIL-Simulation
 Embedded Systems
 FPGA Design
 Generating HDL Code
 Xilinx Zynq SoCs
 AUTOSAR

Code Integration

Integrating C and MATLAB

Polyspace[®]

Polyspace Code Prover[™]

<https://nl.mathworks.com/services/training.html>



© 2017 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.