

MATLAB EXPO

Deploying Motor Control Algorithms to a TI C2000™ Dual-core Microcontroller

Shang-Chuan Lee, MathWorks



Peeyush Kumar, MathWorks





MathWorks 

@MathWorks

Share the EXPO experience
#MATLABEXPO



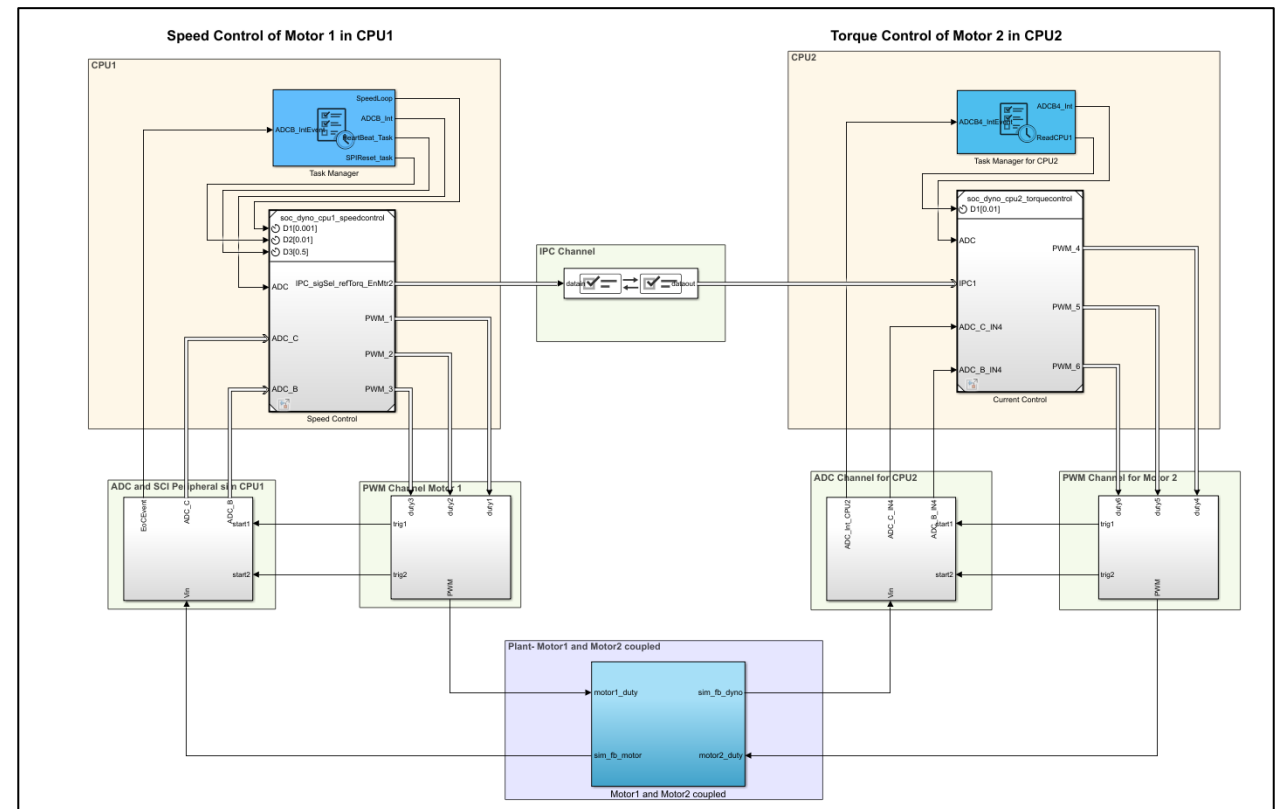
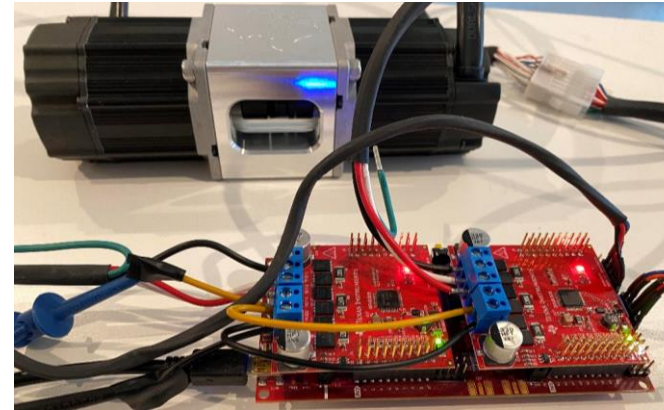
Poll Question

What is the most challenging part in your Motor Control Development process?

1. Motor control algorithm design
2. Write efficient code for embedded processors
3. Identify hardware and software architecture for multicore processors

Key Takeaways

- Simulate sensorless field-oriented control (FOC) on a dyno setup
- Complete Model-Based Design workflow for multicore microcontroller
 - Hardware component and device driver behavior simulation
 - Enhanced on-device profiling



Dual CPUs PMSM Dyno Testing

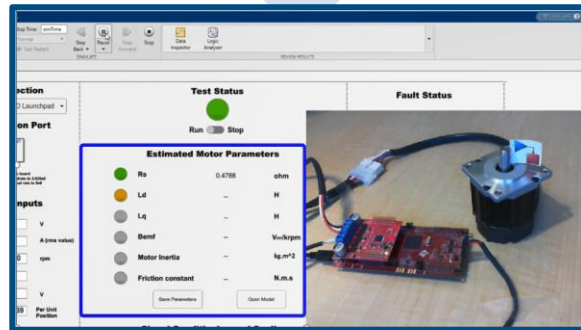
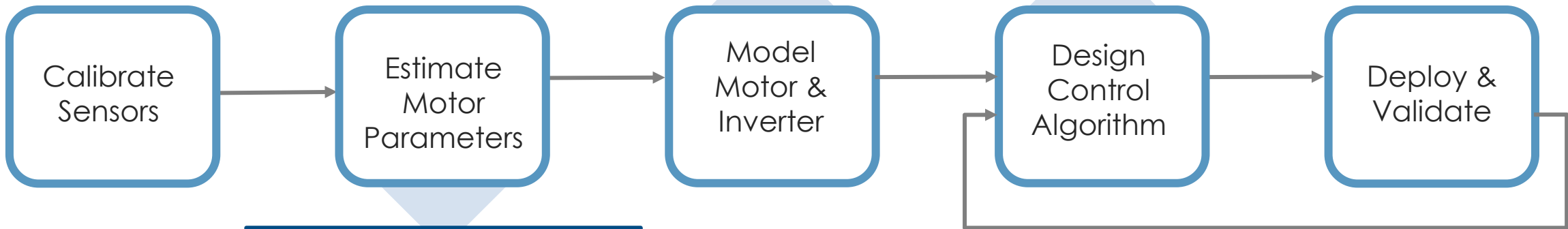
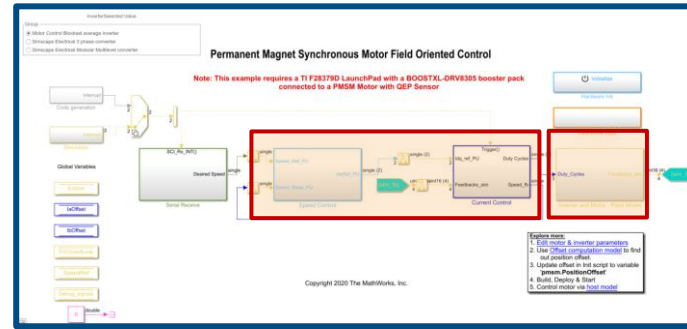
The screenshot displays the MATLAB/Simulink environment for a PMSM Dual CPU Dyno Control Host. The main workspace shows the control model with the following components:

- Inputs:** Two rotary knobs for **SpeedRef.Motor1** (range -4000 to 4000) and **Iq_ref.Motor2** (range -6 to 6).
- Motor Parameters:** Radio buttons for Motor1 and Motor2, with **Speed** selected for Motor1 and **Torque** selected for Motor2.
- Control Elements:** A **Serial Setup** block, a **Motor1** block with a **Stop/Start** toggle switch, and a **Scope (Per-Unit)** block.
- Scope Window:** Shows data from **Serial Communication/1:1** and **Serial Communication/1:2**.
- Physical Hardware:** A photograph of the motor and power supply is overlaid on the bottom right. The power supply display shows **0.056 A** and **24.00 V**.

Copyright 2021 The MathWorks, Inc. T=68.550

Workflow for Implementing Field-Oriented Control

- Motor Control Blockset



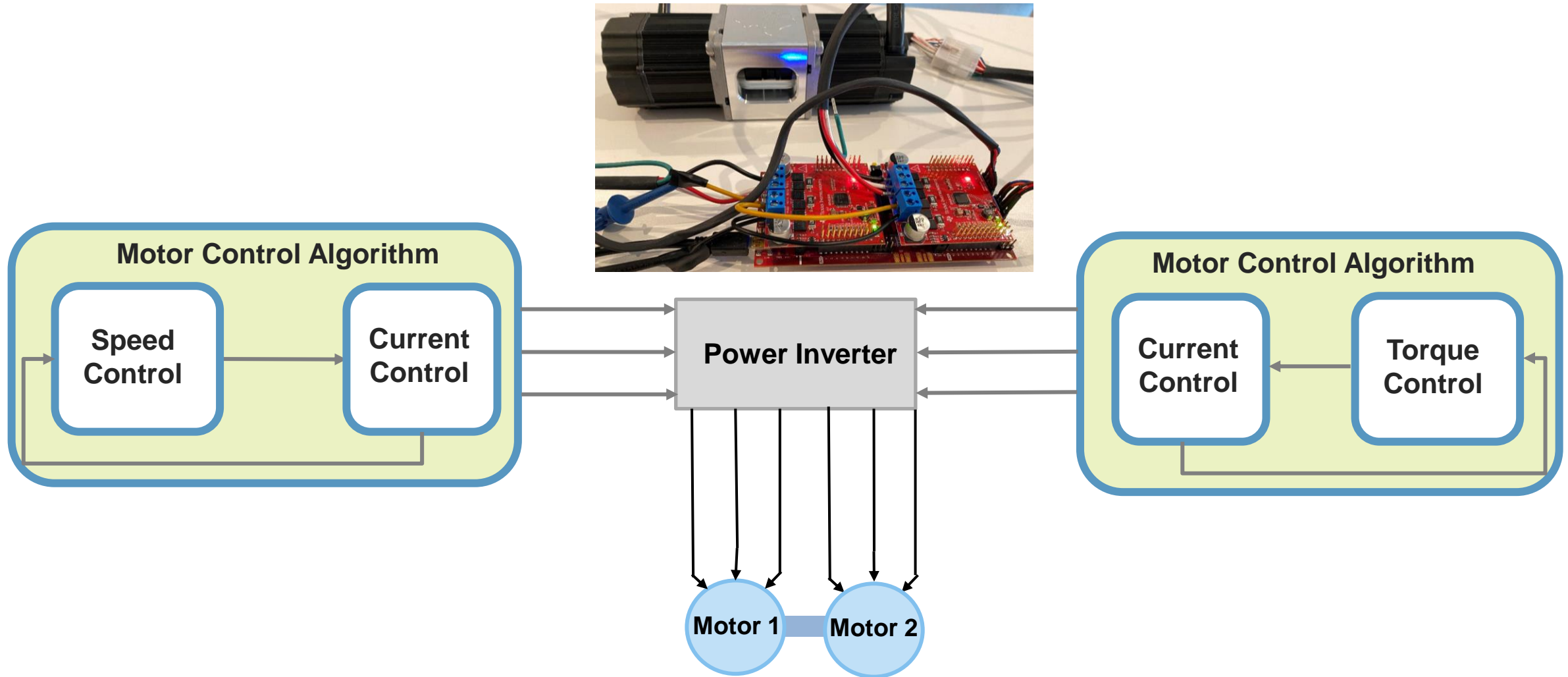
```

/* Product: '<S26>/Product1' */
mcb_pmsm_foc_hall_f28379d_B.Product1_j =
mcb_pmsm_foc_hall_f28379d_B.Saturation_*
mcb_pmsm_foc_hall_f28379d_B.Saturat

/* Sum: '<S26>/Sum2' */
mcb_pmsm_foc_hall_f28379d_B.Sum2_i =
+ mcb_pmsm_foc_hall_f28379d_B.Produ

/* Sqrt: '<S26>/Sqrt' */
mcb_pmsm_foc_hall_f28379d_B.Sqrt = (r
(mcb_pmsm_foc_hall_f28379d_B.Sum2_i
        
```

Simulate Motor Control System



Demo – PMSMs Dyno Model in FOC Sensorless Control

Position Estimator

- Sliding mode observer
- Flux observer

PMSM Motor-Dyno-Sensorless Field Oriented Control

Note: This example requires a TI F28379D LaunchPad with a two BOOSTXL-DRV8305 booster pack connected to a PMSM Motor

Select the value of 'EstimatorSelect' using 'Position Estimator' radio button

Explore more:

1. [Edit motor & inverter parameters](#)
2. Simulate this model
3. Review results in Data Inspector
4. Build, Deploy & Start
5. Control motor via [host model](#)
6. Start the motor in open loop and transition to close loop. The model works in open loop for speed ref below 0.1pu.

Copyright 2021-2022 The MathWorks, Inc.

Challenges of Deployment on the Embedded Systems..

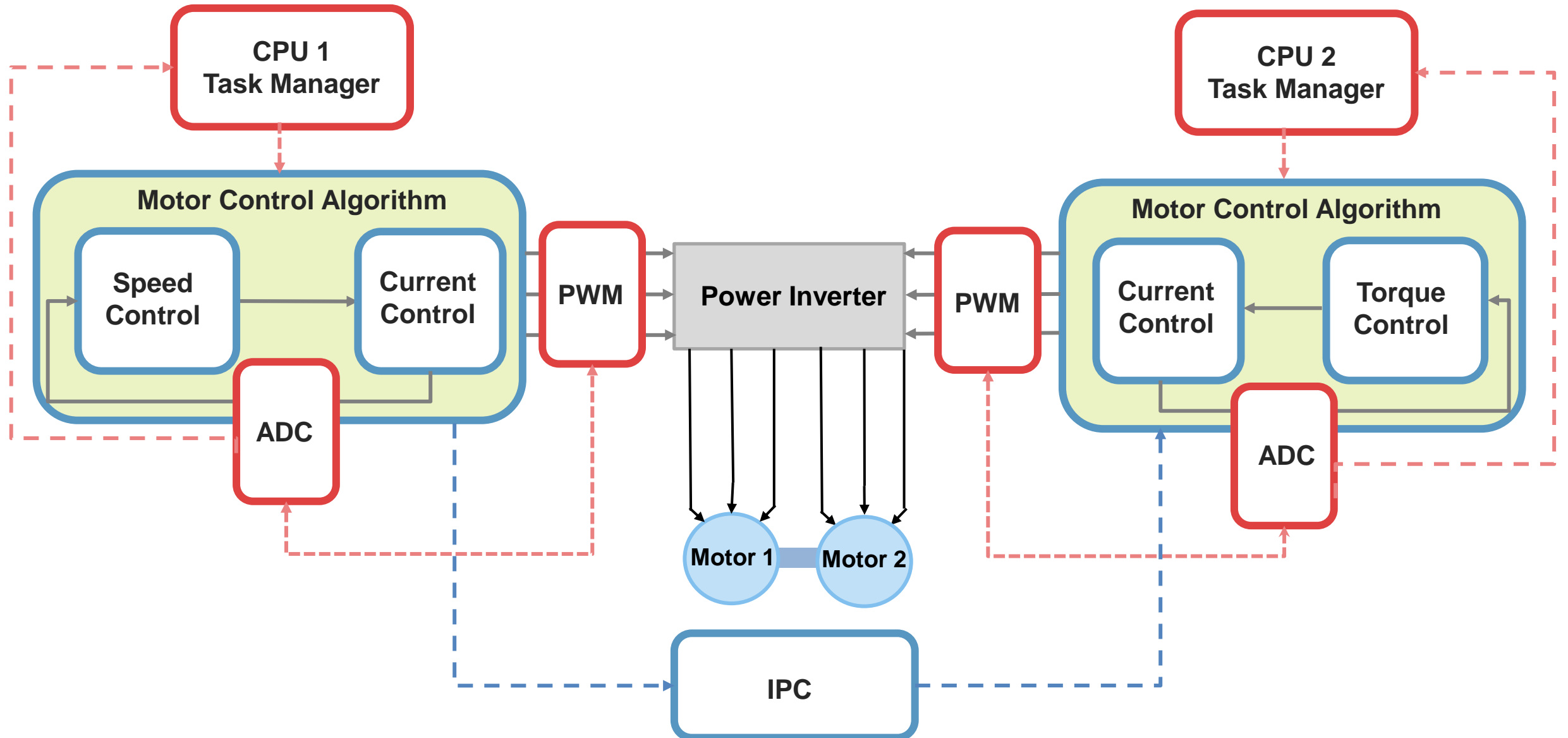
Model

- System requirement
 - TI C2000 dual-core processors
- Controller sample rate is 20kHz
 - Field-oriented control (FOC)
 - Sensorless control
 - Dyno setup (2 motors)
- No sensor delays in my model
- ADC-PWM synchronization

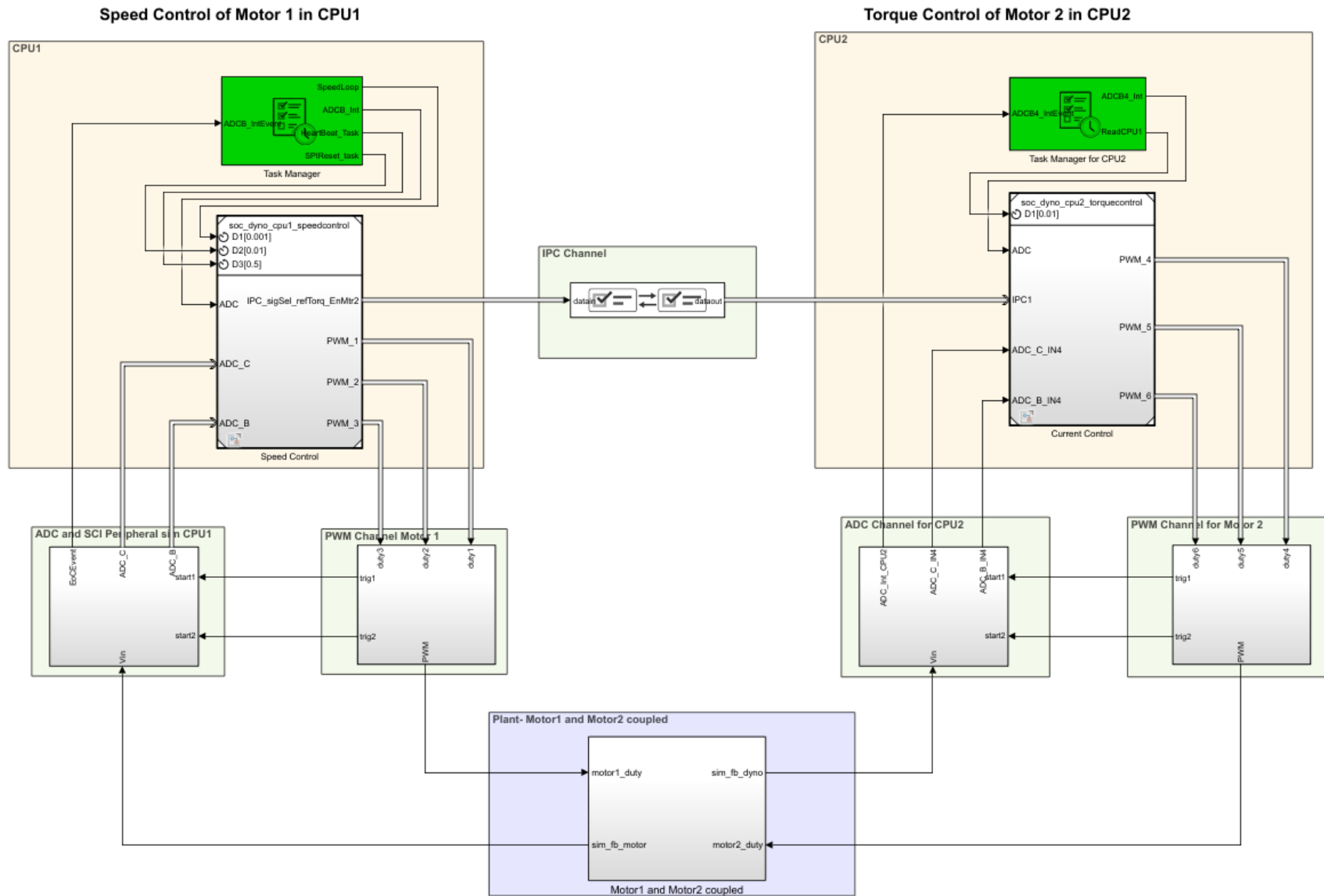
Multicore Processor

- How to implement and partition controls into two separated cores?
- How to communicate between CPU1 and CPU2?
- How to make sure task execution meets software requirement?

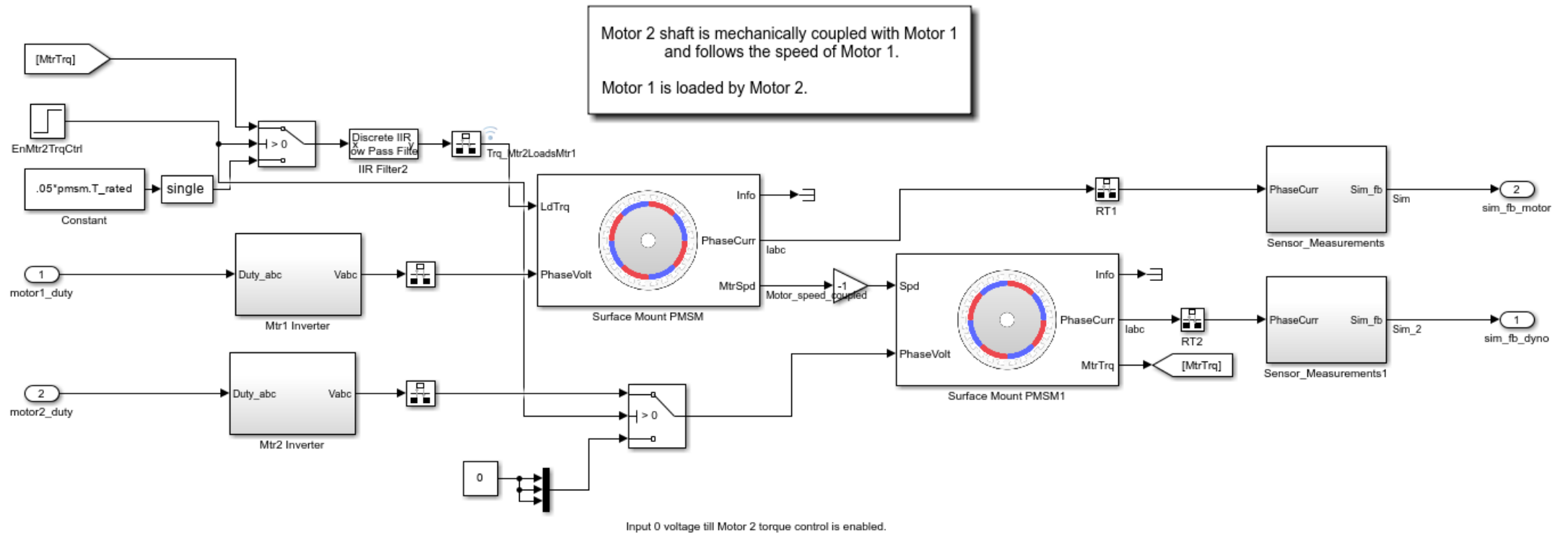
Simulate Motor Control System with Peripherals and Task Execution



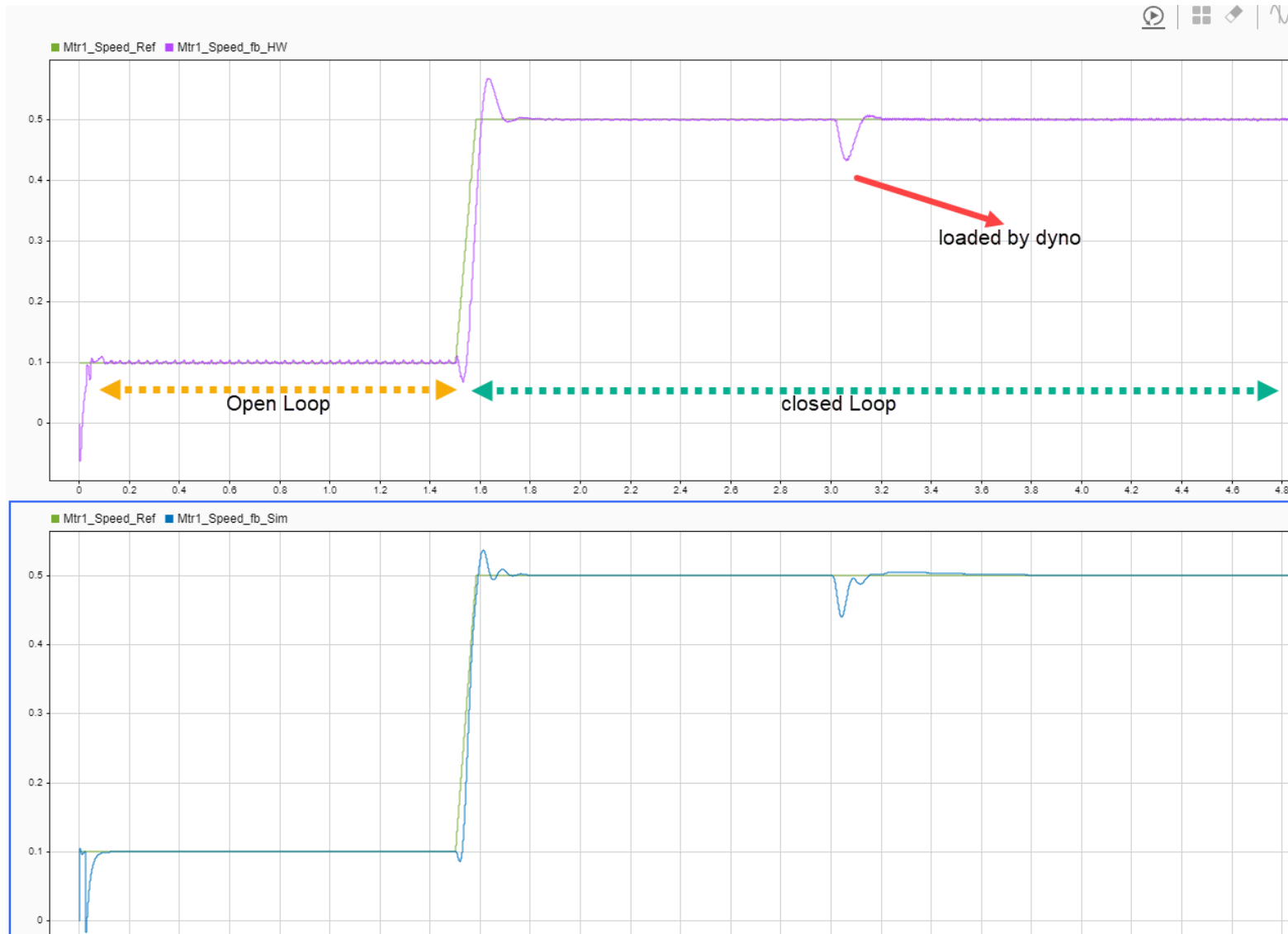
Model Multicore application Using SoC Blockset



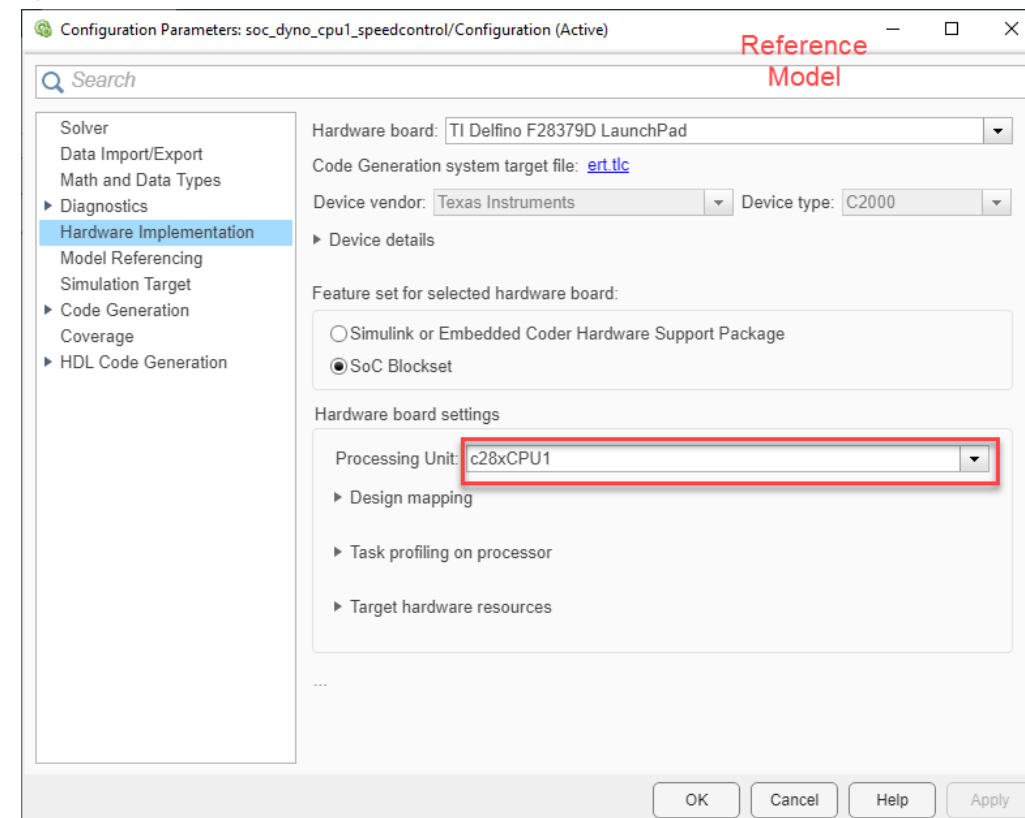
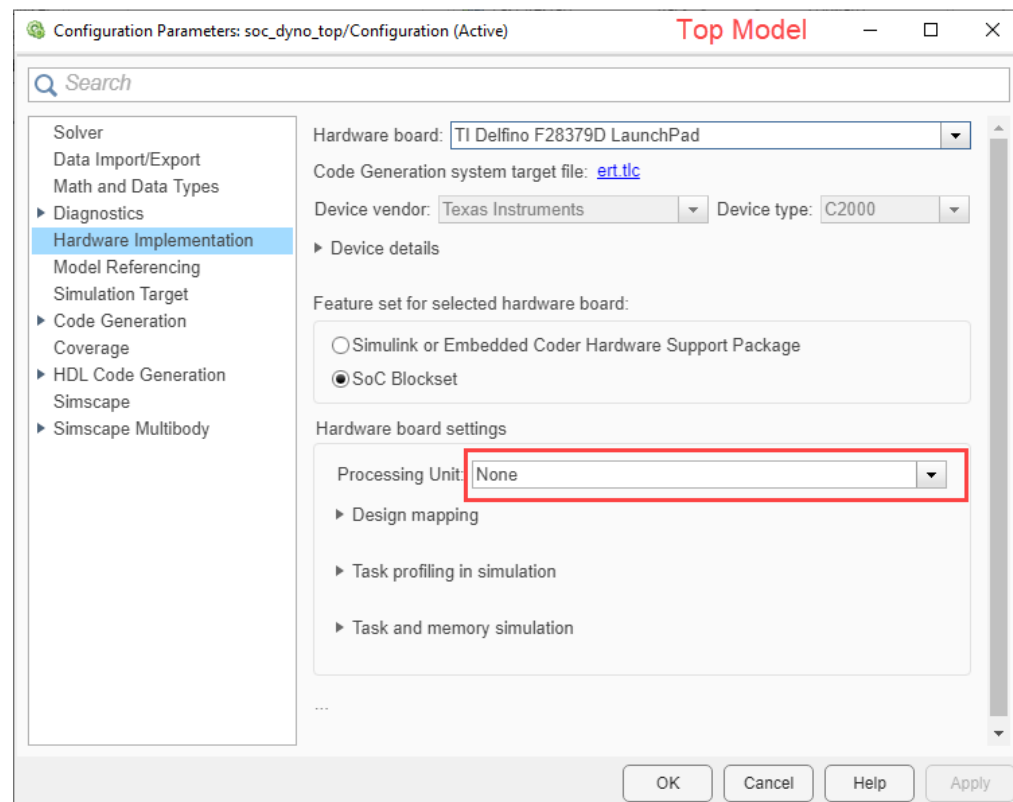
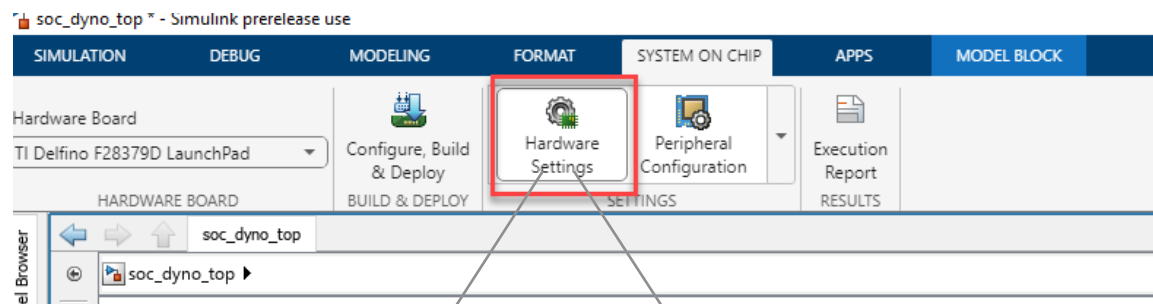
Plant Subsystem



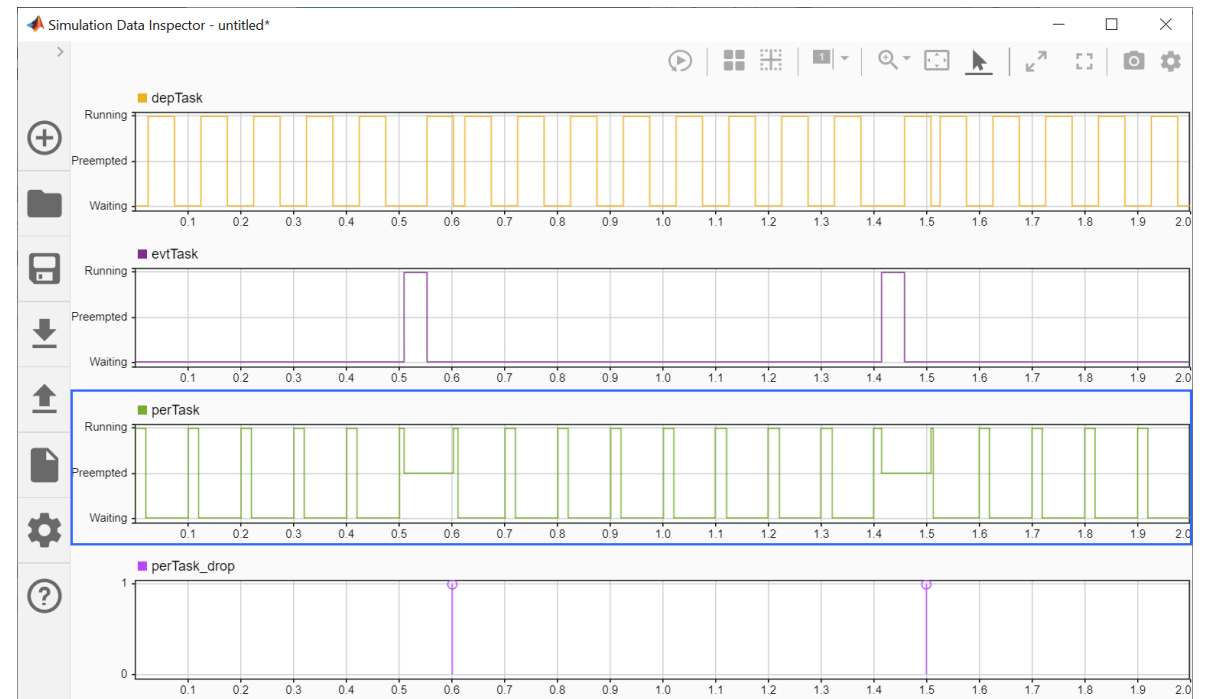
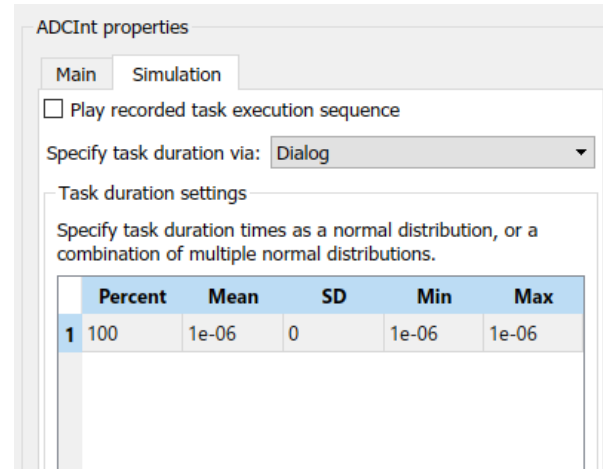
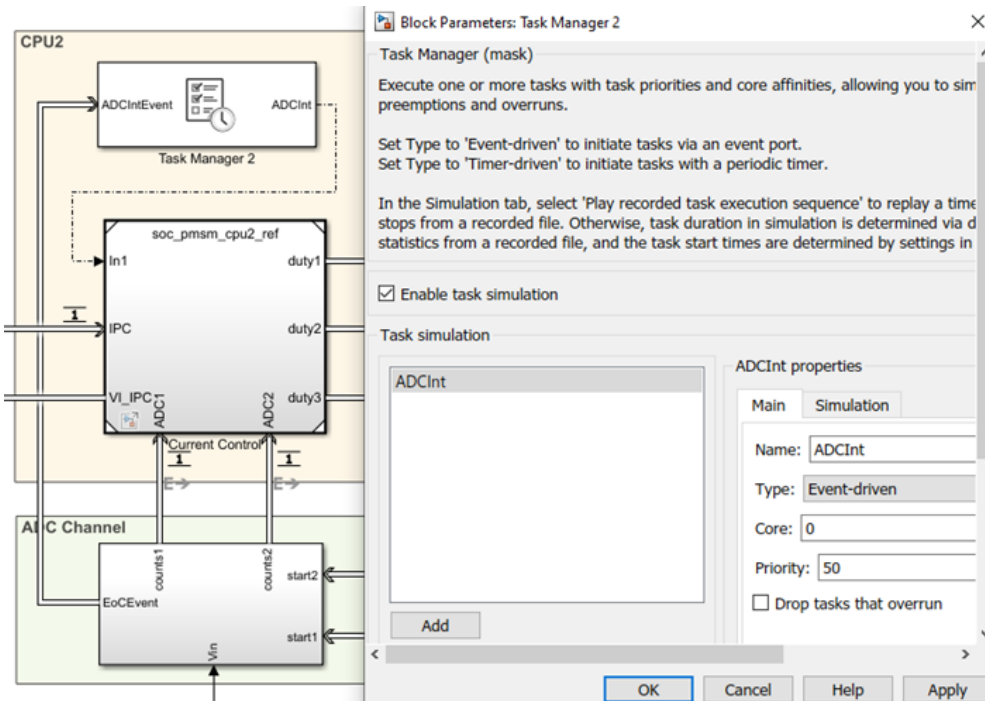
Hardware vs Simulation Analysis



Model Configuration



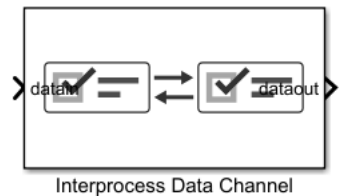
Task Manager



- Model/Simulate
 - Periodic/async tasks
 - Task priorities
 - Latencies
 - Duration as normal distribution

Inter-Processor Communication with IPC Blocks

- Model the communication buffering and delay



Block Parameters: Interprocess Data Channel 1

Interprocess Data Channel
Model data channel between two processes.

Interprocess Data Write block writes the data to the channel from one process.

Interprocess Data Read block reads the data from the channel from another process.

Set Show event port to output a task trigger event when data is written to the channel.

Parameters

Main Statistics

Number of buffers: 1

Propagation delay: 2e-6

Show event port

Define buffer size and timing delay

OK Cancel Help Apply

Block Parameters: Interprocess Data Channel 1

Interprocess Data Channel
Model data channel between two processes.

Interprocess Data Write block writes the data to the channel from one process.

Interprocess Data Read block reads the data from the channel from another process.

Set Show event port to output a task trigger event when data is written to the channel.

Parameters

Main Statistics

Show number of used buffers

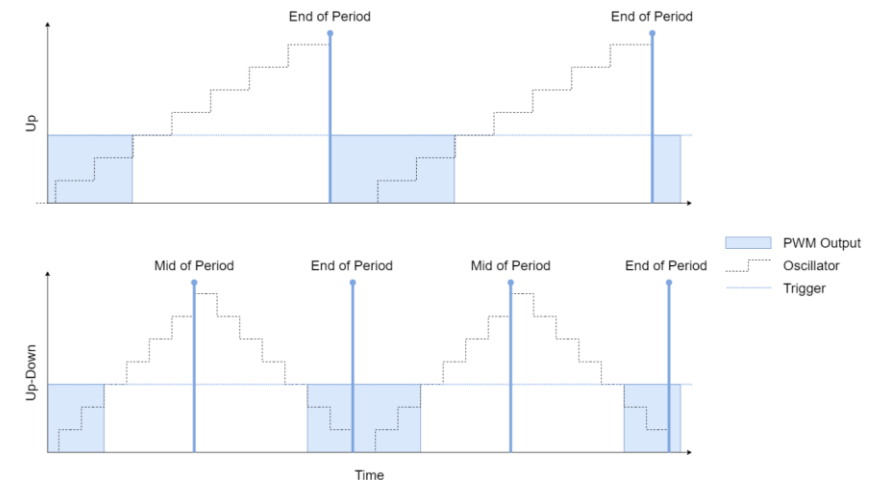
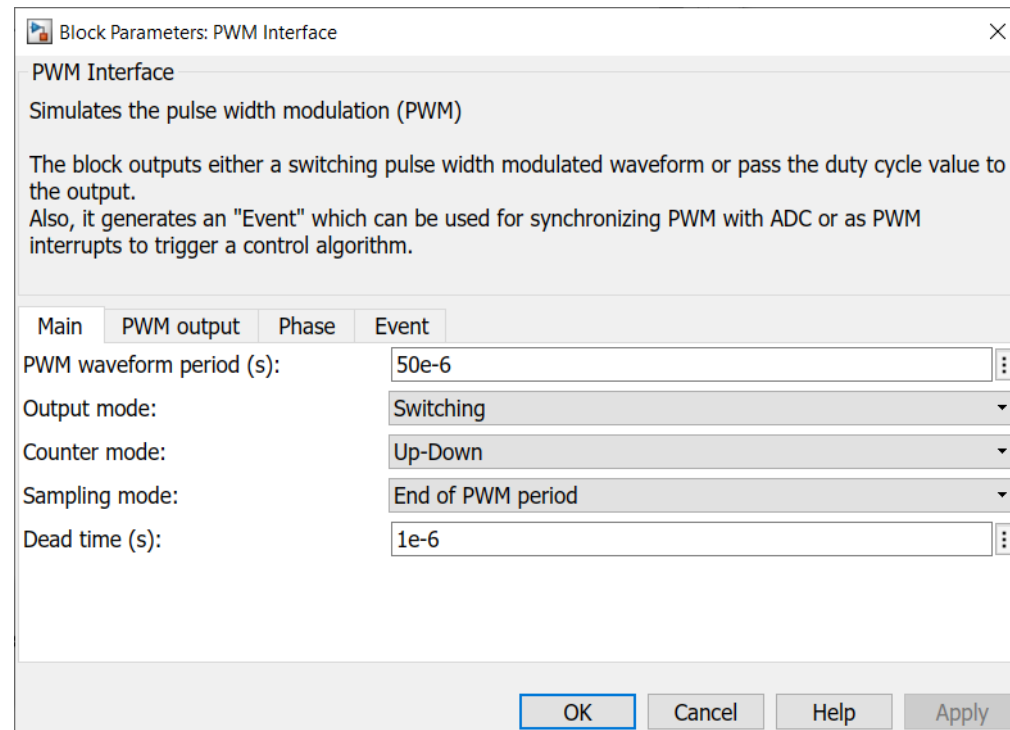
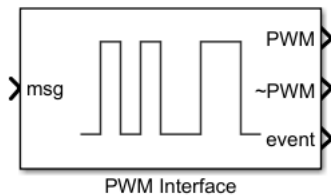
Show when buffer is overwritten

Visualize buffer consumption and overwrites

OK Cancel Help Apply

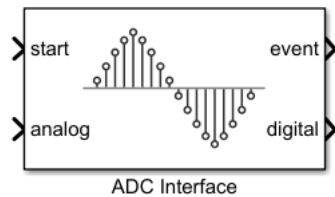
PWM Modeling

- PWM waveform simulation
- Event to synchronize with ADC or schedule a task



ADC Modeling

- Convert analog values to digital counts
- Model acquisition/conversion delays and trigger events



Block Parameters: ADC Interface

ADC Interface
Simulates the analog-to-digital conversion (ADC)

The block samples the analog input based on a start event or sample time and outputs a representative digital value in counts. Also, it generate events which can be used for scheduling an algorithm.

Acquisition time and Conversion time parameter values sets the delays in the conversion.

Main Multi channel Event

Resolution (bits): 12

Voltage reference (V): 3

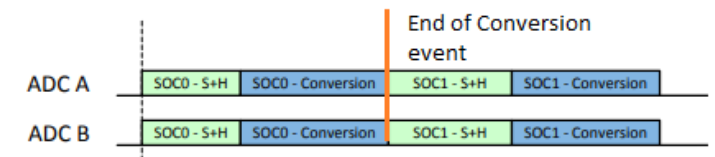
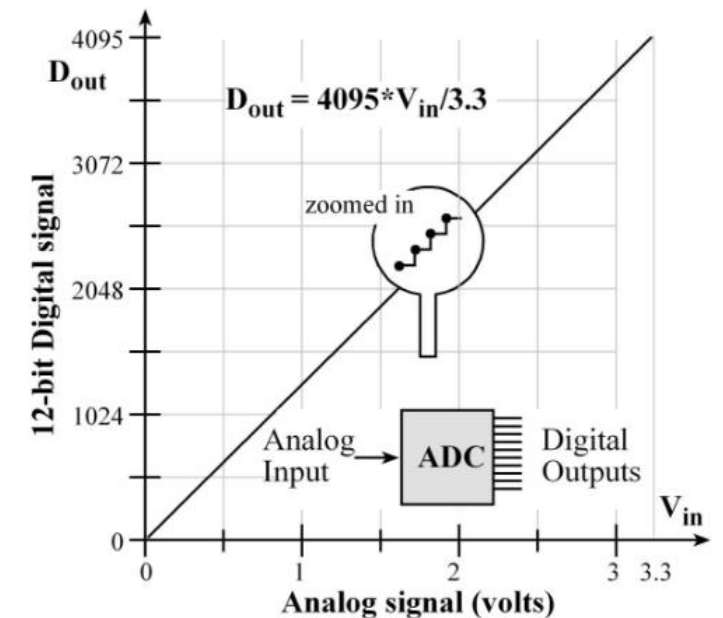
Acquisition time (s): 320e-9

Conversion time (s): 240e-9

Charge/discharge time constant (s): 0

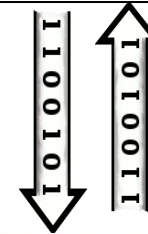
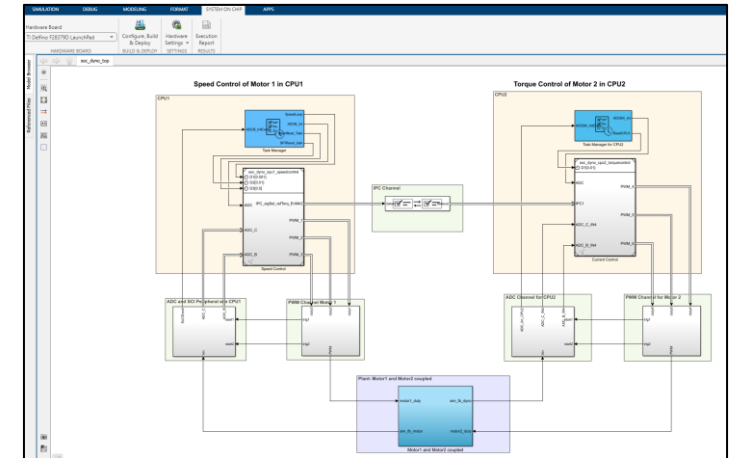
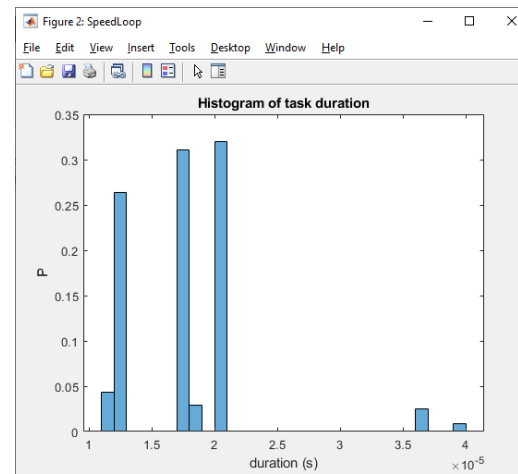
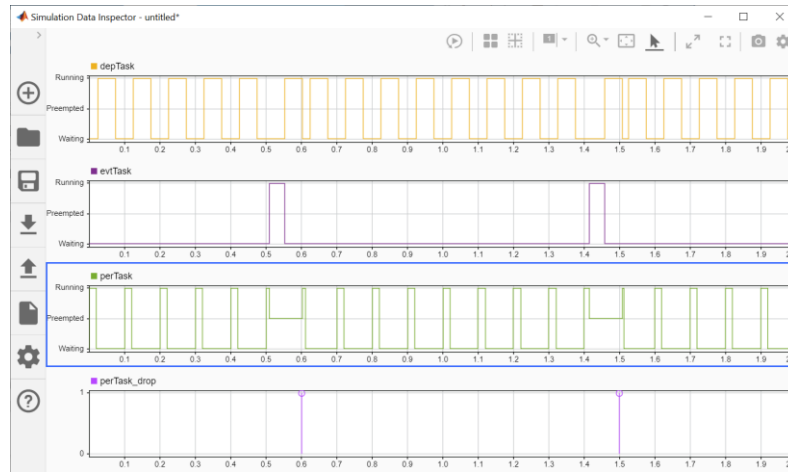
Software triggered

OK Cancel Help Apply

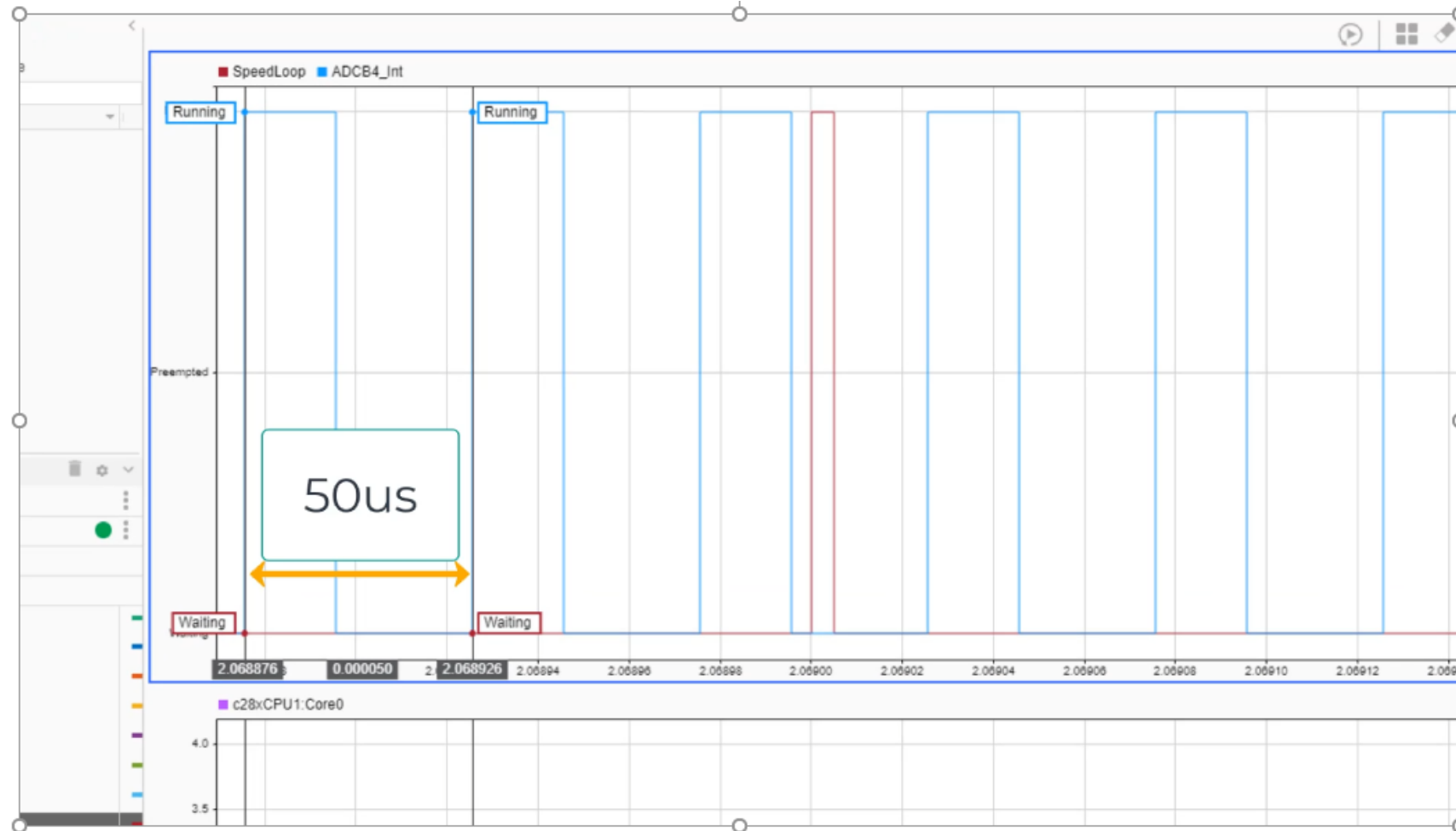


SoC Block set Key Functionalities – On-Device Profiling

- Real-time performance profiling on hardware, including
 - Task execution
 - CPU utilization
 - Communication buffering and delay
 - Real-time SDI view
 - Analysis report



Profile Tasks in Simulation and Hardware



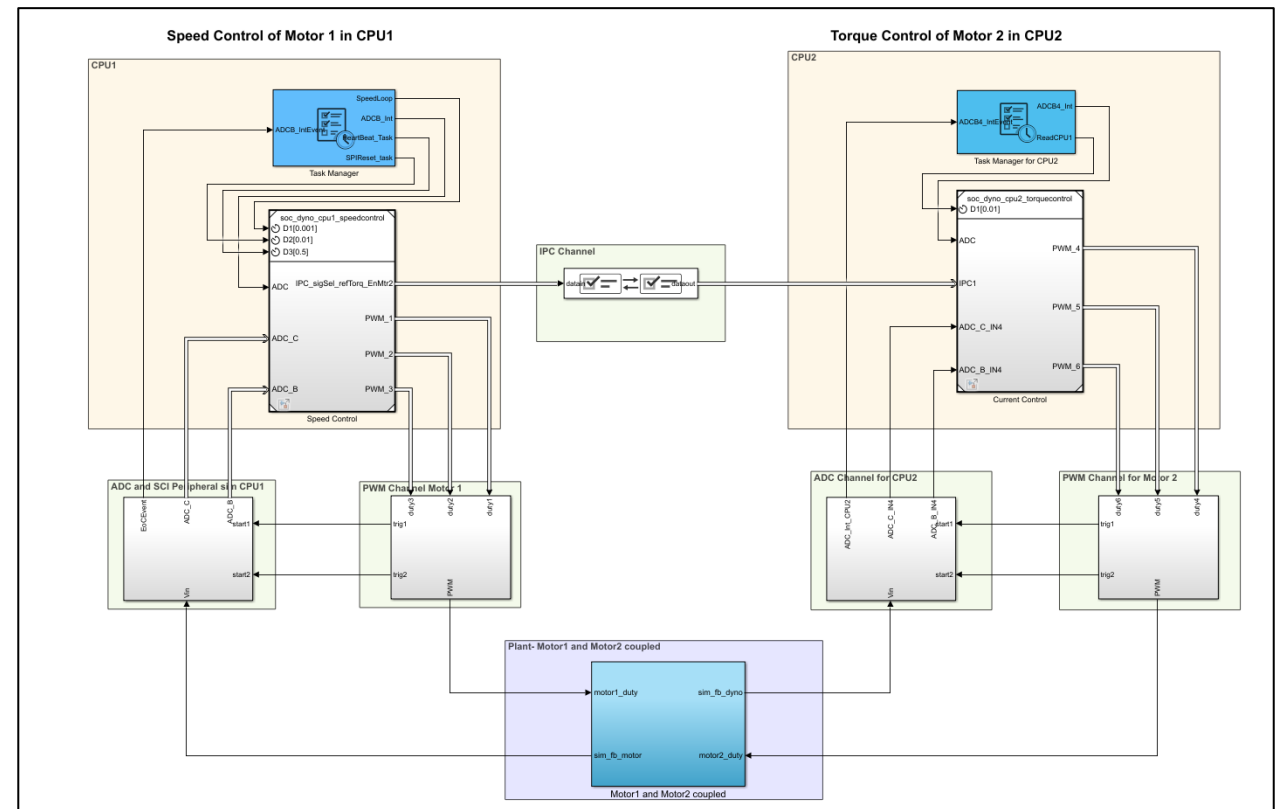
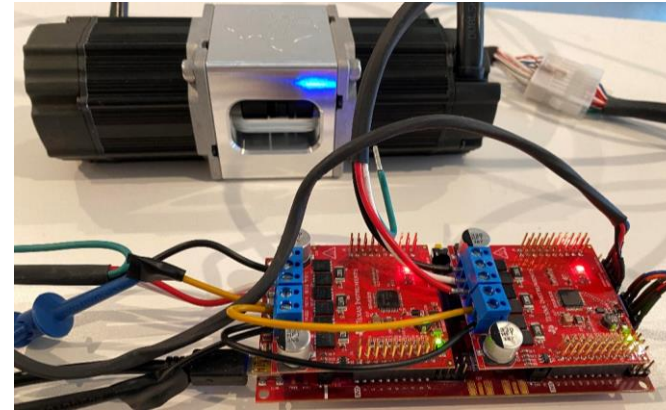
Poll Question

Which topics do you want to discuss more?

1. Motor control design
2. Partition Multicore Processors
3. Simulate PWM/ADC and perform device profiling

Wrap Up

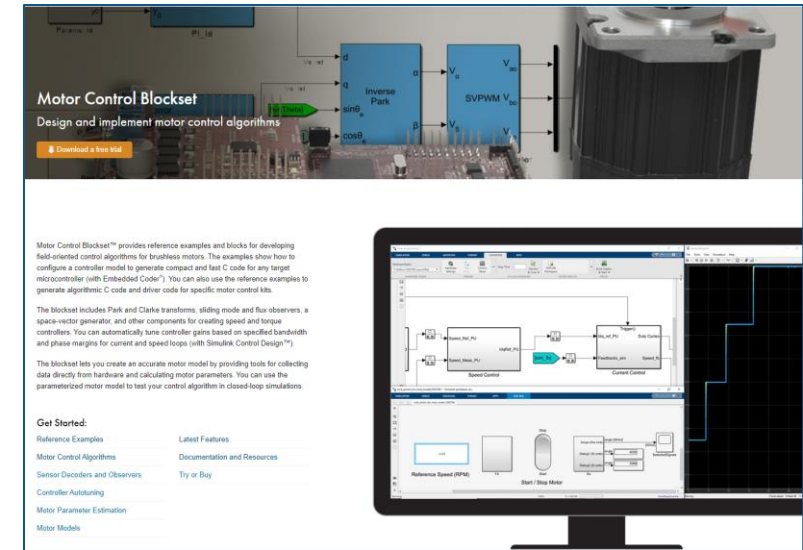
- Simulate sensorless field-oriented control (FOC) on a dyno setup
- Complete Model-Based Design workflow for multicore microcontroller
 - Hardware component and device driver behavior simulation
 - Enhanced on-device profiling



Learn More

- Recording webinars

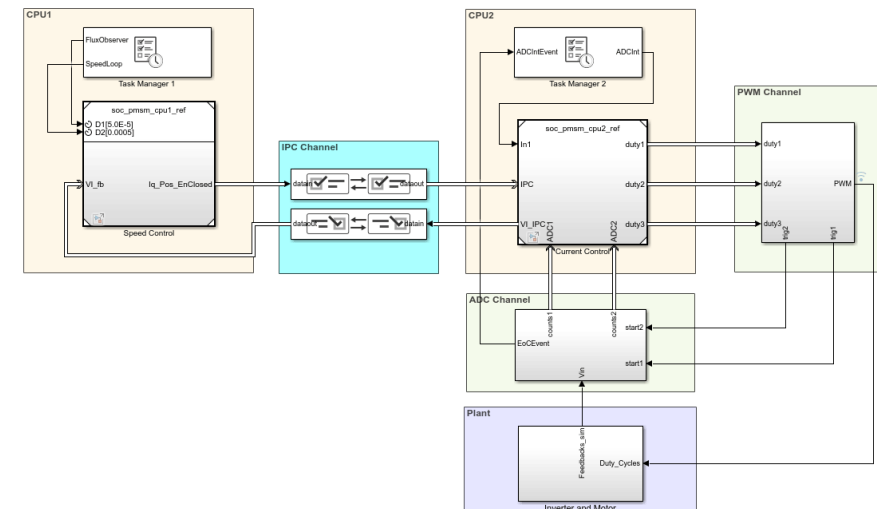
- [Field Oriented Control Made Easy](#)
- [Motor Control with TI Multicore MCUs Using Simulink](#)
- [Implementing Motor and Power Electronics Control on an FPGA-Based SoC](#)



- Shipping Demos

- [Partition Motor Control for Multiprocessor MCUs](#)
- [Control PMSM Loaded with Dual Motor \(Dyno\)](#)
- [Integrate MCU Scheduling and Peripherals in Motor Control Application](#)

Field-Oriented Control on Dual CPU Processor



Copyright 2020 The MathWorks, Inc.

MATLAB EXPO

Thank you



© 2022 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [mathworks.com/trademarks](https://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.