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

Shang-Chuan Lee, MathWorks

Peeyush Kumar, 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
Workflow for Implementing Field-Oriented Control

- Motor Control Blockset

1. Calibrate Sensors
2. Estimate Motor Parameters
3. Model Motor & Inverter
4. Design Control Algorithm
5. Deploy & Validate

Code snippet:
```c
/* Product: 'c3240/Product1' */
mb_pemx_foc_hall_f28379d_b Produk1 =
mb_pemx_foc_hall_f28379d_b_saturate(q_d, q_i, q_p, q_r);
mb_pemx_foc_hall_f28379d_b_saturate(q_d, q_i, q_p, q_r);

/* Sum: 'c3240/Sum2' */
mb_pemx_foc_hall_f28379d_b_Sum2_I =
mb_pemx_foc_hall_f28379d_b_Sum2_D; Prod;

/* Sort: 'c3240/Sort' */
mb_pemx_foc_hall_f28379d_b_Sort =
mb_pemx_foc_hall_f28379d_b_Sort; Prod;```

Sensors
Simulate Motor Control System

Motor Control Algorithm

- Speed Control
- Current Control

Power Inverter

Motor Control Algorithm

- Current Control
- Torque Control

Motor 1

Motor 2
Demo – PMSMs Dyno Model in FOC Sensorless Control
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

Motor 2 shaft is mechanically coupled with Motor 1 and follows the speed of Motor 1. Motor 1 is loaded by Motor 2.

Input 2 voltage till Motor 2 torque control is enabled.
Hardware vs Simulation Analysis
Model Configuration

Top Model

Reference Model
- **Model/Simulate**
  - Periodic/async tasks
  - Task priorities
  - Latencies
  - Duration as normal distribution
Inter-Processor Communication with IPC Blocks

- Model the communication buffering and delay

Define buffer size and timing delay

Visualize buffer consumption and overwrites
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
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

50us
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
Thank you