Advanced AC Motor Control S/W Development

Sang-Ho Yoon
Senior Application Engineer
The MathWorks
Agenda

- Develop ECUs with Model-Based Design
- Generate Application Code for Prototyping
- Generate Algorithm Code for Production
- Case Study: Brushless Motor Control Application
- Questions
ECU Development Process with Model-Based Design

- System Requirements
- System Design
- Software Design
- Software Integration
- Coding
- System Integration and Tuning
- Hardware/Software Integration

- Sim: Simulation
- RP: Rapid Prototyping
- PCG: Production Code Generation
- SIL: Software-in-the-Loop Testing
- PIL: Processor-in-the-Loop Testing
- HIL: Hardware-in-the-Loop Testing
Simulation with Simulink

- Simulating plant and controller in one environment allows engineers to optimize system-level performance.

- Controls engineers and domain specialists can work together to detect integration issues in simulation.
Rapid Prototyping

Generate, deploy, and tune code for a component (algorithm or controller) on a real-time simulator connected to system hardware.
Rapid Prototyping on Embedded Processors

Run the generated code in real time, tune parameters, and monitor real-time data on the same processor you plan to use in mass production, or a close equivalent to it.
Processor-in-the-Loop Testing

Use processor-in-the-loop PIL to evaluate the behavior of a candidate algorithm on the target processor.
Hardware-in-the-Loop Testing

Final test before integration using simulated plant executing in real time.
Agenda

- Develop ECUs with Model-Based Design
- **Generate Application Code for Prototyping**
- Generate Algorithm Code for Production
- Case Study: Brushless Motor Control Application
- Questions
xPC Target Focusing Areas

- **System Requirements**
- **System Design**
- **Software Design**
- **Software Integration**
- **Coding**
- **System Integration and Tuning**
- **Hardware/Software Integration**

**Sim**: Simulation  
**RP**: Rapid Prototyping  
**PCG**: Production Code Generation  
**SIL**: Software-in-the-Loop Testing  
**PIL**: Processor-in-the-Loop Testing  
**HIL**: Hardware-in-the-Loop Testing
What is xPC Target?

- An environment that allows the real-time execution of Simulink models on a separate xPC Target Kernel compatible PC.
What is xPC Target?

- An Environment that provides interactive access between the real-time application and the host computer
- Allows live parameter tuning, control from the original Simulink model and offline analysis support in MATLAB.
What is xPC Target?

- An Environment that provides interactive access between the real-time application and the host computer
- Allows live parameter tuning, control from the original Simulink model and offline analysis support in MATLAB.
What is xPC Target?

- An Environment that provides numerous I/O device driver blocks
- Blocks are easily configurable within the Simulink model and communicate with actual hardware in real-time.
What is xPC Target?

- An Environment that provides numerous I/O device driver blocks
- Blocks are easily configurable within the Simulink model and communicate with actual hardware in real-time.
Mobile Real-Time Target Machine

- **Performance:**
  - 2GHz Dual Core Intel processor
  - 2GB RAM, 100GB HD
  - Gigabit Ethernet

- **Expandable:** 5 PCI Slots

- **Digital and Analog I/O, CAN, Serial, etc**

- **FPGA Expansion Card:**
  - High-speed PWM and Quadrature generation and decoding
  - High-frequency control loops
  - Programmable with Mathworks HDL Coder
xPC Target Turnkey

- Target Hardware partnership with Speedgoat
- Hardware customized to fit your performance and I/O needs

- Saves you time and hassle of acquiring, installing, configuring, and testing

Agenda

- Develop ECUs with Model-Based Design
- Generate Application Code for Prototyping
- **Generate Algorithm Code for Production**
- Case Study: Brushless Motor Control Application
- Questions
Considerations for Embedded S/W Development

### Mapping Application Requirements to the Optimization Pane

<table>
<thead>
<tr>
<th>Configuration Parameter</th>
<th>Settings for Building Code</th>
<th>Factory Default</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Debugging</td>
<td>Traceability</td>
</tr>
<tr>
<td>Block reduction</td>
<td>Off (GRT)</td>
<td>Off</td>
</tr>
<tr>
<td></td>
<td>No impact (ERT)</td>
<td></td>
</tr>
<tr>
<td>Implement logic signals as Boolean data (vs. double)</td>
<td>No impact</td>
<td>No impact</td>
</tr>
<tr>
<td>Inline parameters</td>
<td>Off (GRT)</td>
<td>On</td>
</tr>
<tr>
<td></td>
<td>On (ERT)</td>
<td></td>
</tr>
<tr>
<td>Conditional input branch execution</td>
<td>No impact</td>
<td>On</td>
</tr>
<tr>
<td>Signal storage reuse</td>
<td>Off</td>
<td>Off</td>
</tr>
<tr>
<td>Application lifespan (days)</td>
<td>No impact</td>
<td>No impact</td>
</tr>
<tr>
<td>Enable local block outputs</td>
<td>Off</td>
<td>No impact</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ignore integer downcasts in folded expressions</td>
<td>Off</td>
<td>No impact</td>
</tr>
<tr>
<td>Eliminate superfluous local variables (Expression folding)</td>
<td>Off</td>
<td>No impact (GRT)</td>
</tr>
<tr>
<td>Minimize data copies between local and global variables</td>
<td>Off</td>
<td>Off</td>
</tr>
</tbody>
</table>
Code Generation Products: Simulink Coder and Embedded Coder

**Simulink Coder**
- Generates code for use in simulation and prototyping applications
- Comes with Generic Real-Time (GRT) based targets

**Embedded Coder**
- Generates efficient code that can be customized to look like hand code for production
- Comes with Embedded Real-Time (ERT) based targets
Agenda

- Develop ECUs with Model-Based Design
- Generate Application Code for Prototyping
- Generate Algorithm Code for Production
- **Case Study: Brushless Motor Control Application**
- Questions
Brushed DC vs. AC Motor Control

Brushed DC Motor Control: Mechanical Commutation

Brushless DC Motor: Electrical Commutation
Example PMSM Controller

Controller Algorithm Attributes:
- Outer Loop Velocity Controller @100Hz
- Inner Loop Current Controller @25kHz
- Coordinate transforms (ABC to/from dq)
- Multiple PI Controllers

Controller Outputs:
Three PWM Duty Cycles

Rotor Phase Currents (AC Waveforms)
Optimize Performance with Simulation

- Simulating plant and controller in one environment allows engineers to optimize system-level performance.
- Controls engineers and domain specialists can work together to detect integration issues in simulation.
Rapid Prototyping

System Model

Controller Model

Motor Model

Code Generation

Controller C Code

Motor Hardware

Real Time Simulator

Rapid Prototyping
Hardware High-Level Block Diagram

Simulink Host

Real-Time Target

xPC Target Turnkey Mobile Machine

Application Hardware

Power Electronics

PMSM Motor

Encoder

Encoder
Embedded Code Generation

System Model

Controller Model

Motor Model

Code Generation

Controller C Code

Motor Hardware

Embedded Software

Embedded System
Integrating Generated Controller Code with an Embedded Software Project

Embedded Software Project

Execute at 25kHz

Controller

Command

ADC

Encoder

PWM
Integrating Generated Controller Code with an Embedded Software Project

Embedded Software Project

```c
Main()
{
    Commands_Init
    PWM_Init
    ADC_Init
    Encoder_Init

    Controller_Init

    while(1) {
    }
}

ADC_InterruptServiceRountine_25kHz()
{
    Commands_Read
    ADC_Read
    Encoder_Read

    Controller_Step

    PWM_Write
}
```
Assessing behavior and execution time using Processor-In-the-Loop (PIL)
Questions?