Motor Control Toolbox Overview

FTF-AUT-F0463

Gavin ZHANG | Automotive Marketing Manager

M A Y . 2 0 1 4
Agenda

• Motor Control Development Toolbox Overview: Library Blocks, FreeMASTER, and Bootloader

• Model Based Design Steps: MIL, SIL, PIL and HIL

• Motor Control Development Toolbox Example
Agenda

- Motor Control Development Toolbox Overview: Library Blocks, FreeMASTER, and Bootloader
- Model Based Design Steps: MIL, SIL, PIL and HIL
- Motor Control Development Toolbox Example
Introduction: Model Based Design (MBD)

• Model Based Design is becoming more common during the normal course of software development to explain and implement the desired behavior of a system. The challenge is to take advantage of this approach and get an executable that can be simulated and implemented directly from the model to help you get the product to market in less time and with higher quality. This is especially true for electric motor controls development in this age of hybrid/electric vehicles and the industrial motor control application space.

• Many companies model their controller algorithm and the target motor or plant so they can use a simulation environment to accelerate their algorithm development.

• The final stage of this type of development is the integration of the control algorithm software with target MCU hardware. This is often done using hand code or a mix of hand code and model-generated code. Motor Control Development Toolbox allows this stage of the development to generate 100% of the code from the model.
Introduction: Motor Control Development Toolbox

• The Motor Control Development Toolbox includes an embedded target supporting Freescale MCUs, Simulink™ plug-in libraries which provide engineers with an integrated environment and tool chain for configuring and generating the necessary software, including initialization routines, device drivers, and a real-time scheduler to execute algorithms specifically for controlling motors.

• The toolbox also includes an extensive Automotive Math and Motor Control Function Library developed by Freescale’s renowned Motor Control Center of Excellence. The library provides dozens of blocks optimized for fast execution on Freescale MCUs with bit-accurate results compared to Simulink™ simulation using single-precision math.

• The toolbox provides built-in support for Software and Processor-in-the-Loop (SIL and PIL), which enables direct comparison and plotting of numerical results.

**MathWorks products required for MC Toolbox:**
- MATLAB (32-Bit or 64-Bit)*
- Simulink
- MATLAB Coder
- Simulink Coder
- Embedded Coder

*Earlier released products only support 32-bit
Introduction: Reduce Development Time With MBD and MC Toolbox

- **System Requirements**: Use software-based model vs. paper-based method, and start testing at very earliest stage.

- **Modeling/Simulation**: Convert model to SIL and now can test ANSI-generated software. Can also use MC library with SIL testing.

- **Rapid Prototype**: With MC library and MC Toolbox, test Model using target MCU and compiler through PIL testing.

- **Target MCU Implementation**: With MC Toolbox, auto-generate code for direct interface of peripherals for target hardware without any manual hand code.

- **HIL Testing**: Now that more testing on target has occurred earlier in the process, HIL testing time is reduced.

- **Functional Testing**: Fewer defects found in this phase of testing, where finding defects is expensive.

Using Freescale’s Motor Control Development Toolbox with Model Based Design and you can reduce development time from this.
Introduction: Reduce Development Time With MBD and MC Toolbox

To This!
## MCD Toolbox: Qorivva Toolbox Library Contents

<table>
<thead>
<tr>
<th>Peripherals</th>
<th>Configuration/Modes</th>
<th>Utility</th>
<th>MCUs Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>• General</td>
<td>• Compiler Options</td>
<td>• FreeMASTER Interface</td>
<td>• MPC5643L/7xK</td>
</tr>
<tr>
<td>– ADC conversion</td>
<td>– CodeWarrior</td>
<td>– Data acquisition</td>
<td>• MPC574xP</td>
</tr>
<tr>
<td>– Digital I/O</td>
<td>– Wind River DIAB</td>
<td>– Calibration</td>
<td>• S12ZVM</td>
</tr>
<tr>
<td>– PIT timer</td>
<td>– Green Hills</td>
<td>– Customize GUI</td>
<td>• KV10Z</td>
</tr>
<tr>
<td>– ISR</td>
<td>– Cosmic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Communication Interface</td>
<td>• RAM/FLASH targets</td>
<td></td>
<td></td>
</tr>
<tr>
<td>– CAN driver</td>
<td>• Simulation Modes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>– SPI driver</td>
<td>– Normal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Motor Control Interface</td>
<td>– Accelerator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>– Cross triggering unit</td>
<td>– Software in the Loop (SIL)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>– PWM</td>
<td>– Processor in the Loop (PIL)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>– eTimer block(s)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Sine wave generation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• ADC Command List</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• GDU (Gate Drive Unit)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PTU (Prog Trigger Unit)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• TIM Hall Sensor Port</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• MCU Option</td>
<td>• Multiple packages</td>
<td></td>
<td></td>
</tr>
<tr>
<td>– Multiple Crystal frequencies</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**MCUs Supported**

- MPC5643L/7xK
- MPC574xP
- S12ZVM
- KV10Z
MCD Toolbox: Auto Math and Motor Control Library Contents

GFLIB

- Trigonometric Functions
  - GFLIB_Sin
  - GFLIB_Cos
  - GFLIB_Tan
  - GFLIB_Asin
  - GFLIB_Acos
  - GFLIB_Atan
  - GFLIB_AtanXY
- Limitation Functions
  - GFLIB_Limit
  - GFLIB_LowerLimit
  - GFLIB_UpperLimit
  - GFLIB_VectorLimit
- PI Controller Functions
  - GFLIB_ControllerPIr
  - GFLIB_ControllerPIrAW
  - GFLIB_ControllerPlp
  - GFLIB_ControllerPlpAW
- Linear Interpolation
  - GFLIB_Lut1D
- Hysteresis Function
  - GFLIB_Hyst
- Signal Integration Function
  - GFLIB_IntegratorTR
- Sign Function
  - GFLIB_Sign
- Signal Ramp Function
  - GFLIB_Ramp

GDFLIB

- Finite Impulse Filter
  - GDFLIB_FilterFIR
- Moving Average Filter
  - GDFLIB_FilterMA
- 1st Order Infinite Impulse Filter
  - GDFLIB_FilterIIR1init
  - GDFLIB_FilterIIR1
- 2nd Order Infinite Impulse Filter
  - GDFLIB_FilterIIR2init
  - GDFLIB_FilterIIR2

GMCLIB

- Clark Transformation
  - GMCLIB_Clark
  - GMCLIB_ClarxInv
- Park Transformation
  - GMCLIB_Park
  - GMCLIB_ParkInv
- Duty Cycle Calculation
  - GMCLIB_SvmStd
- Elimination of DC Ripples
  - GMCLIB_ElimDcBusRip
- Decoupling of PMSM Motors
  - GMCLIB_DecouplingPMSM
MCD Toolbox: RAppID Bootloader Utility

The RAppID Bootloader works with the built-in Boot Assist Module (BAM) included in the Freescale Qorivva and PX series family of parts. The Bootloader provides a streamlined method for programming code into FLASH or RAM on either target EVBs or custom boards. Once programming is complete, the application code automatically starts.

**Modes of Operation**
- The Bootloader has two modes of operation: for use as a stand-alone PC desktop GUI utility, or for integration with different user required tools chains through a command line interface (i.e. Eclipse Plug-in, MATLAB/Simulink, …)

**MCUs Supported**
- MPC5534, MPC5601/2D, MPC5602/3/4BC, MPC5605/6/7B, MPC564xB/C, MPC567xF, MPC567xK, MPC564xL, MPC5604/3P, MPC574XP and S12ZVM

**Graphical User Interface**

**Command**

Status given in two stages: Bootloader download, then application programming
FreeMASTER – Run Time Debugging Tool

• User-friendly tool for real-time debug monitor and data visualization
  – Completely non-intrusive monitoring of variables on a running system
  – Display multiple variables changing over time on an oscilloscope-like display, or view the data in text form
  – Communicates with an on-target driver via USB, BDM, CAN, UART

• Establish a Data Trace on Target
  – Set up buffer (up to 64KB), sampling rate and trigger
  – Near 10-µs resolution

http://www.freescale.com/freemaster
Agenda

• Motor Control Development Toolbox Overview: Library blocks, FreeMASTER, and Bootloader

• Model Based Design Steps: MIL, SIL, PIL and HIL

• Motor Control Development Toolbox Example
Model Based Design Steps: Step 1 (Simulation)

Simulation in PC environment

- ADC
- A/D Conversion
- Analog Sensor Model
- Controller Model
- Electric Motor Model
- Analog Device Model

Idealized simulation of the controller and the motor to refine the control technique. Done on host PC without regard for embedded controller. Can optionally add analog device models for fault detection and signal control.
Model Based Design Steps: Step 2 (SIL)

(SIL) Generated code executes as atomic unit on PC

Still done on host PC without regard for embedded controller. Instead using generated C code that is compiled using a PC-based compiler. Run same test vectors as in simulation for C Code Coverage analysis and verify functionality.
Model Based Design Steps: Step 3 (PIL)

(PIL) Executes generated code on the target MCU

Execute the model on the target MCU and perform numeric equivalence testing. Co-execution with MCU and Model Based Design working together while collecting execution metrics on the embedded controller of control algorithm. Validate performance on MCU.
Model Based Design Steps: Step 4 (Target MCU)*

Generate production code to run on embedded MCU with real motor while collecting execution metrics on the embedded controller of control algorithm. Validate performance on MCU and use FreeMASTER to tune control parameters and perform data logging.

* I/O peripheral driver blocks can be included in the model, providing the analog driver interfaces needed to directly interface to devices external from the MCU.
Model Based Design Steps: Summary

Step 1 – Model in Loop (MIL):
- MBD Simulation Only
- Software requirements
- Control system requirements
- Overall application control strategy
- Modeling style guidelines applied
- Algorithm functional partitioning
- Interfaces are defined here

Step 2 – Software in Loop (SIL):
- MBD Simulation with ANSI C Code using SIL
- Control algorithm design
- Code generation preparation
- Control system design
- Overall application control strategy design
- Start testing implementation approach
- Testing of functional components of algorithm
- Test harness to validate all requirements
- Test coverage of model here
- Creates functional baseline of model

Step 3 – Processor in Loop (PIL):
- MBD Simulation with ANSI C Code using PIL
- Controller code generation
- Determine execution time on MCU
- Verify algorithm on MCU
- See memory/stack usage on MCU
- Start testing implementation approach
- Target testing controls algorithm on MCU
- Refine model for code generation
- Function/File partitioning
- Data typing to target environment done here
- Scaling for fixed point simulation and code gen
- Testing of functional components of algorithm
- Test harness to validate all requirements
- Test coverage of model here
- Creates functional baseline of model
- Equivalence testing

Step 4 – Hardware in Loop (HIL):
- ANSI C Code Running on Target HW & MCU
- Validation/verification phase
- Controller code generation
- Determine execution time on MCU
- Start testing implementation on target ECM
- Code generate control algorithm + I/O drivers. Complete implementation on ECM. Test system in target environment
- Utilize calibration tools for data logging and parameter tuning

Execute code on target MCU
- Functional testing in target environment
- Ensure execution on target is correct as well as code generation on target is performing as desired.
Agenda

• Motor Control Development Toolbox Overview: Library blocks, FreeMASTER, and Bootloader

• Model Based Design Steps: MIL, SIL, PIL and HIL

• Motor Control Development Toolbox Example
Example: FOC_SIL_PIL
Example: FOC Realization
SIL Simulation : SIL Scope
SIL Simulation: Normal Scope
SIL Simulation : Compare Scope
Example: Digital HALL Signal Process

• Interrupt Trigger

Interrupt Triggered

Interrupt Triggered

Interrupt Triggered
Pulse Width Calculation

Calculate Pulse Width and then calculate the speed of Motor
Digital Input

Configure digital input driver, put HALL signal to the proper MCU input PIN
eTimer Configuration

Configure eTimer configuration driver, capture HALL signal rising and falling edge and trigger interrupt

Sub Module 0 Channel 0
Count mode: 001 - Count rising edges of primary source
Output mode: 0000 - Software controlled
PrimarySource: 11111 - IP Bus clock divide by 128 prescaler
SecondarySource: 00010 - Counter #2 input pin

eTimer configuration Hall A

Sub Module 0 Channel 1
Count mode: 001 - Count rising edges of primary source
Output mode: 0000 - Software controlled
PrimarySource: 11111 - IP Bus clock divide by 128 prescaler
SecondarySource: 00011 - Counter #3 input pin

eTimer configuration Hall B

Sub Module 0 Channel 2
Count mode: 001 - Count rising edges of primary source
Output mode: 0000 - Software controlled
PrimarySource: 11111 - IP Bus clock divide by 128 prescaler
SecondarySource: 00100 - Counter #4 input pin

eTimer configuration Hall C
eTimer Capture

Configure eTimer capture, calculate HALL signal pulse width
MCD Toolbox: Summary of Application Support

User Application Software

Application SW
- API
- Algorithm Libraries
- Drivers
- On-Chip Peripherals

System Infrastructure
- MC library set
- GDFLIB: Digital filtering
- GFLIB: General functions
- GMCLIB: Motor Control
- Drivers: Efficient reflecting the chip features

External Hardware
- External Connections
- PINS

Target Platform

Documentation
- User Application Software
- Application SW
- System Infrastructure
- External Hardware

FreeMaster Support
- Boot Loader Support
MBD Demo - System Block Diagram

3Ph Driver Chip (MC33927)

- Phase A
- Phase B
- Phase C

Vb+
Vb-

BLDC Motor

Phase Voltages

Hall Sensors

SPI
FlexPWM

MPC5643L

LinFlex

ADC
eTimer/DI

DO

RApplD BL Utility

FREEMASTER

Freescale™

External Use | 29
Hardware

**TWR-PXS2010**

- PXS20 dual-core
- Nexus interface
- OSJTAG
- CAN
- UART
- ADC
- Potentiometer
- Accelerometer
- TWRPI socket

**TWR-IND-IO**

- USB to Serial
- CAN
- I/O
- ADC

**TWR-MC-LV3PH**

- Input voltage 12-24V DC
- Output current 5-10 Amps
- 3-phase MOSFET inverter
- 3-phase pre-driver MC33937
- Analog sensing
- Motor speed/position sensors interface
- 2 pole-pair BLDC motor with Hall sensors (4000 RPM rated speed)
- On-board power regulation for Tower System (single power supply via TWR-MC-LV3PH power jack)
System Model: Realize BLDC close loop speed control
HIL: FreeMASTER to Monitor and Tune Parameters
Designing with Freescale

Tailored live, hands-on training in a city near you

2014 seminar topics include

• QorIQ product family update
• Kinetis K, L, E, V series MCU product training

freescale.com/DwF