Sensorless Field Oriented Control of a PMSM Motor Using S12 MagniV S12ZVM Mixed-Signal MCUs

FTF-AUT-F0232

Branislav Zigmund | Motor Control Applications
Anita Maliverney | Field Applications

APR. 2014

Freescale
Session Objectives

After completing this session you will be able to:

- Name the key motor control features on the new MagniV S12ZVM family
- Describe the basic idea of Field Oriented Control
- Outline the main elements of a sinusoidal PMSM control technique
Agenda

- S12ZVM Motor Control Family Overview
- Special Motor Control Features
  - Supporting digital modules and ADC
  - Integrated high voltage analog modules
- Sensorless PMSM Motor Control
  - Introduction
  - Field oriented control basics and design
  - Sensorless PMSM control by position estimation using saliency based back-EMF
  - Output voltage generation: DC bus ripple elimination & space vector modulation performance
  - Current measurement
  - Possible Enhancements of the Algorithm
- Performance
- Summary
Agenda

- S12ZVM Motor Control Family Overview
  - Special Motor Control Features
    - Supporting digital modules and ADC
    - Integrated high voltage analog modules
  - Sensorless PMSM Motor Control
    - Introduction
    - Field oriented control basics and design
    - Sensorless PMSM control by position estimation using saliency based back-EMF
    - Output voltage generation: DC bus ripple elimination & space vector modulation performance
    - Current measurement
    - Possible Enhancements of the Algorithm
  - Performance
  - Summary
S12ZVM - Single Chip Solution for BLDC Motor Control

Discrete Solution:
- VREG (8pin)
- LIN phy (8pin)
- MCU or DSC (48pin)
- Gate Driver (48pin)
- Op-amps

Optimize system cost:
- ~ 50 fewer solder joints
- - 4 to 6 cm² PCB space

Optimize system efficiency:
- Vector Control

S12ZVM Solution:
- ~ 50 fewer solder joints
- - 4 to 6 cm² PCB space

4cm ~1 ½ in.
BLDC/PMSM Motors: Market Segmentation

200+W motors
- Cooling Fan
- Sliding doors
- Fuel pump

50-200W motors
- Water pump
- Oil pump

S12ZVML/C
Features:
- 32 to 128 KB flash
- Vreg + boost
- 100-150 nC gate drive unit
- Charge pump
- Current sense OpAmp
- LIN transceiver or CAN transceiver
- Supply

S12ZVM
Features:
- 16 to 32KB flash
- Vreg + boost
- 50-70 nC gate drive unit
- Charge pump
- Current sense OpAmp
- 1 HVI for PWM bus
S12ZVM Ecosystem – The Complete Solution

Customer Application Software

- MC ToolBox: Rapid prototyping with Matlab Simulink

- FreeMASTER: -Graphical User Interface -Instrumentation

- MCAT Tuning Tool

- MC Dev Kit Reference Software

- Autosar OS

Math and Motor Control Libraries:
- Standard optimized math functions and motor control algorithms
- Includes Matlab Simulink Models

Compiler and Debugger

Graphical Init Tool

Hardware (Evaluation board, target application)

- FSL production Software

- FSL enablement Software

- 3rd Party production Software

Watch out at FTF:
- FTF-AUT-F0233Freescale Automotive Motor Control Enablement Solutions
- FTF-AUT-F0076: Hands-On Workshop: Motor Control Toolbox Overview
- FTF-AUT-F0014: Overview of FreeMASTER and RAppID Bootloader
Agenda

• S12ZVM Motor Control Family Overview

• Special Motor Control Features
  − Supporting digital modules and ADC
    − Integrated high voltage analog modules

• Introduction to Sensorless PMSM Motor Control

• Sensorless PMSM Motor Control
  − Introduction
  − Field oriented control basics and design
  − Sensorless PMSM control by position estimation using saliency based back-EMF
  − Output voltage generation: DC bus ripple elimination & space vector modulation performance
  − Current measurement
  − Possible Enhancements of the Algorithm

• Performance

• Summary
Autonomous Motor Control Loop Implementation

- **PMF** (Pulse Width Modulation With Fault)
  - Commutation Event
  - PWM signals
  - Fault Inputs

- **PTU** (Programmable Trigger Unit)
  - Trigger
  - Reload

- **ADC0** (Analog Digital Converter)
  - Conversion Command
  - Conversion Result

- **ADC1** (Analog Digital Converter)
  - Conversion Command
  - Conversion Result

- **GDU** (Gate Drive Unit)
  - DC Bus Volt.
  - BackEMF
  - DC Bus Curr.

- **RAM / NVM**
  - Trigger List(s) (<=32)
  - Command List(s) (<=64)
  - Result List(s) (<=64)

- **NO CPU involvement & interrupt during motor control cycle**
Pulse Width Modulator Module (PMF)

- 6 PWM channels, 3 independent counters
  - Up to 6 independent channels or 3 complementary pairs
- Based on core clock (max. 100MHz)
- Complementary operation:
  - Dead time insertion
  - Top and Bottom pulse width correction
  - Double switching
  - Separate top and bottom polarity control
- Edge- or center-aligned PWM signals
- Integral reload rates from 1 to 16
- 6-step BLDC commutation support, with optional link to TIM Output Compare
- Individual software-controlled PWM outputs (+ easy masking feature per output)
- Programmable fault protection
2 x 12-bit Analog Digital Converter

**DMA integrated**

**Automatic Trigger**
- Can be triggered by PTU, for accurate synch with PWM
- Up to 32 triggers per control cycle per ADC

**From PTU**
- LoadOK
- Seq_abort
- Restart

**From External or OpAmp**
- ANx 8
- ANx 2
- ANx 1
- ANx 0

**External & OpAmp Inputs**
- 9 external channels (5 to ADC0 and 4 to ADC1)
- OpAmp output shared with ADC external channel

**Monitoring Internal Signals**
- DC link, phase voltages, Vsup
- Vreg & ADC temp sensors
- Bandgap voltage

**Sample Time**
- Selectable: 480 ns to 2.88 µs

**Conversion Time**
- 1.8 µs @ max. ADC clock for 12 bit

**Command and Result List**
- Double buffered
- Flexible conversion sequence and oversampling
- Fully autonomous motor control cycle which unloads CPU

**DMA**
- Takes commands from SRAM /NVM and stores results back into SRAM

**Command and Result List**
- Double buffered
- Flexible conversion sequence and oversampling
- Fully autonomous motor control cycle which unloads CPU

**Sys Clock**
- PRSCLR

**ADC clock**
- DMA

**Interrupt**
- Abort

**Automatic Trigger**
- Can be triggered by PTU, for accurate synch with PWM

**Up to 32 triggers per control cycle per ADC**

**Internal ADC channels**

**Monitoring Internal Signals**
- DC link, phase voltages, Vsup
- Vreg & ADC temp sensors
- Bandgap voltage
Programmable Trigger Unit (PTU)

Completely avoids CPU involvement to trigger ADC during the control cycle

- One 16-bit counter as time base
- Two independent trigger generators (TG)
- Up to 32 trigger events per trigger generator
- Trigger Value List stored in system memory
- Double buffered list, so that CPU can load new values in the background
- Software generated “Reload” & trigger event
- Synchronized with PMF and ADC to guarantee coherent update of all control loop modules
S12Z Core: An Optimized Powerful Machine

24-bit address bus maps up to 16MB (no paging needed!)

- Harvard Architecture – Parallel data & code accesses
- CPU operates at 100 MHz
- Fractional math support
- Instructions/addressing optimized for C programming
- Multiple length register set optimized for less memory access
- 32-Bit data paths, ALU, data registers
- 24-bit address bus, stack pointer, program counter and X/Y index registers
- It has a 16-bit I/O data path
- Handles 8-Bit data and indices

<table>
<thead>
<tr>
<th>Attribute</th>
<th>S12Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shifter</td>
<td>32-Bit multi-bit</td>
</tr>
<tr>
<td>Multiplier</td>
<td>32*32</td>
</tr>
<tr>
<td></td>
<td>16*16</td>
</tr>
<tr>
<td>Divider</td>
<td>32 = 32/32</td>
</tr>
<tr>
<td>MAC</td>
<td>32 += 32*32</td>
</tr>
<tr>
<td>Fractional math</td>
<td>Yes</td>
</tr>
<tr>
<td>Bus speed</td>
<td>50MHz</td>
</tr>
</tbody>
</table>
Autonomous PMSM Application Timing

Two shunts current sensing

PMF
- PMF MOD VAL0
- PWM counter
- PMF phA out (A_top A_bot)
- PMF phB out (B_top B_bot)
- PMF half cycle reload every fourth opportunity
- deadtime
- PMF half cycle reload every fourth opportunity

PTU
- PTU counter
- PTU triggers
- Trig. 0
- Trig. 1
- Delay in order for DMA to load ADC list

ADC0
- DMA0
- Phase current A sampling
- U_{DCBus} sampling
- ADC0 conversion
- ADC0 conversion

ADC1
- DMA1
- Phase current B sampling
- Temp sampling
- ADC1 conversion

CPU
- ISR service routine

SW Serviced Hardware Events

Free for application use

FOC Calculations
application

EOC interrupt
EOC interrupt

External Use | 14
Agenda

- S12ZVM Motor Control Family Overview
- **Special Motor Control Features**
  - Supporting digital modules and ADC
  - Integrated high voltage analog modules
- Sensorless PMSM Motor Control
  - Introduction
  - Field oriented control basics and design
  - Sensorless PMSM control by position estimation using saliency based back-EMF
  - Output voltage generation: DC bus ripple elimination & space vector modulation performance
  - Current measurement
  - Possible Enhancements of the Algorithm
- Performance
- Summary
## Operating Voltage Ranges

<table>
<thead>
<tr>
<th>Vsup</th>
<th>MCU</th>
<th>GDU</th>
<th>Vsup</th>
<th>MCU</th>
<th>GDU</th>
</tr>
</thead>
<tbody>
<tr>
<td>20V…40V</td>
<td>Full</td>
<td>Disabled</td>
<td>20V … 40V</td>
<td>Full</td>
<td>Disabled</td>
</tr>
<tr>
<td>7V…20V</td>
<td>Full</td>
<td>Enabled Vgs &gt; Vsup – 2*Vbe (5V min)</td>
<td>9.5V…20V</td>
<td>Full</td>
<td>Boost OFF for Vsup &gt; 11V Vgs = 9.6V</td>
</tr>
<tr>
<td>6V … 7V</td>
<td>Full</td>
<td>Disabled</td>
<td>6V…9.5V</td>
<td>Full</td>
<td>Boost ON Vgs &gt;9V</td>
</tr>
<tr>
<td>3.5V … 6V</td>
<td>Full Iddx = 25mA max if no external PNP</td>
<td>Disabled</td>
<td>3.5V … 6V</td>
<td>Full Iddx = 25mA max if no external PNP</td>
<td>Boost ON Vgs &gt;9V</td>
</tr>
<tr>
<td>&lt;3.5V</td>
<td>Reset</td>
<td>Disabled</td>
<td>&lt;3.5V</td>
<td>Reset</td>
<td>Disabled</td>
</tr>
</tbody>
</table>
Power Domain Overview

Flexible Boost & Bypass Option

5V Reg.Bypass option
Can optionally be used to manage power dissipation

5V Regulator
70mA @ Vsup >6V

5V CAN Regulator
Supply of external CAN transceiver (bond option)
Used with external bypass transistor

Low Voltage Monitor

Boost Converter
Supports GDU operation at min. 3.5V
Vsup boost turn on trip point typ. 10.1V

Vsup Sense
Low / high voltage monitoring
Selectable warning levels

11V LDO Regulator
Supplies GDU
Typ. 77mA current limit

Undervoltage Monitoring
Monitors all internal domains

ADC
Core RAM PLL OSC
Flash
GPIO

VREG_AUTO

Bypass option

Boost option

Reverse Protected VBAT

VSUP
BST

VDDX
BCTL

CPS

GLVLSF
VLS_OUT
VLS
LG
LS
GDU

LG
LS

External Use | 17
Boost Converter

 Coil Size
High frequency option allows small coils down to 10 µH

Step-up frequency /duty cycle
Wide range selectable
Optimizes \( dI/dt \) and power dissipation
Frequency range: 62.5kHz .. 12.5MHz @ 50MHz \( f_{\text{bus}} \)
Duty cycle: 25%, 50% or 75%

Coil Current limitation
threshold selectable between typ. 190 mA, 380 mA and 560 mA

Boost Function
Optional
guarantees a \( V_{GS} > 9V \) at low \( V_{\text{BAT}} \) conditions

Coil

Reverse Battery FET

D1

C1

C2

BST

Boost Converter Clock

Clock Frequency & Duty Cycle

Bus Clock Input

Boost Converter Enable

V_{\text{BAT}}<9.5V \rightarrow \text{Boost Converter enabled}
Gate Driver Unit (GDU) Topology

11V LDO
- Supplies the LS drivers
- Charges bootstrap cap for the HS drivers

Voltage Monitoring
- HD High Voltage Monitor @ typ. 21/27.3 V
- VLS Low Voltage trip point: 6.2 .. 7V

Integrated Dividers
- HD: divider 12 ; HS : divider 6

Phase Comparators
- Compares HS against DCbus/2 in HW

Phase Multiplexer
- Switched in each sector

Slew Rate Control
- Output current limitation of Iout via selectable Iref
- 8 selectable slew rates

Drive Strength
- Typ 100-150nC

Turn-off Resistor
- 80 kOhm pull down integrated

Charge Pump
- Supplies HS Gate Drive
- Supports 100% duty cycle

HS Topology
- Bootstrap for HS Gate supply

Max. PWM Frequency
- Min. driver pulse width = 2µs

HG / HS / LG / LS
- Max. rating: -5 .. 42V

Charge Pump
- Supplies HS Gate Drive
- Supports 100% duty cycle

HS Topology
- Bootstrap for HS Gate supply

Max. PWM Frequency
- Min. driver pulse width = 2µs

HG / HS / LG / LS
- Max. rating: -5 .. 42V

Charge Pump
- Supplies HS Gate Drive
- Supports 100% duty cycle

HS Topology
- Bootstrap for HS Gate supply

Max. PWM Frequency
- Min. driver pulse width = 2µs

HG / HS / LG / LS
- Max. rating: -5 .. 42V

Charge Pump
- Supplies HS Gate Drive
- Supports 100% duty cycle

HS Topology
- Bootstrap for HS Gate supply

Max. PWM Frequency
- Min. driver pulse width = 2µs

HG / HS / LG / LS
- Max. rating: -5 .. 42V
**V_{DS} Monitoring & Overcurrent Protection**

- After turning on (any) high-side or low-side transistor, the HSx voltage is monitored.
- In case of de-saturation error:
  - LS/HS switched off
  - Optional interrupt

- 6 Desaturation Comparators
  - One HS and one LS per phase
- Programmable blanking time
  - From 80ns..4.84 µs @ 50MHz

Desaturation voltage

- Programmable from 0.3V to 1.35V in 8 steps (150mV steps)
Current Measurement & Overcurrent Protection

**Overcurrent Comparator**
Flexible programmable with 6-bit DAC
Voltage range: 3.82V .. VDDA

**Current measurement**
- Two current sense amplifiers
- Supports 2-shunt systems
- Measures voltage across shunt

**Output voltage**
- Range 0 V .. VDDA
- Measured on ADC via external channel AMP(0/1)

**Gain**
- Selectable via external resistors

**Offset compensation**
- +/-5mV to +/-15mV software selectable

**Measure negative currents**
- By adding an external offset

**Overcurrent Condition**
\[ aV_{sense} + V_{ref} < V_{oct} \]

**Output Voltage to ADC**
\[ V_{AMP} = aV_{sense} + V_{ref} \]

**Gain**
Select via external resistors

**Output voltage range**
0 V .. VDDA
measured on ADC via external channel AMP(0/1)
Agenda

- S12ZVM Motor Control Family Overview
- Special Motor Control Features
  - Supporting digital modules and ADC
  - Integrated high voltage analog modules
- Sensorless PMSM Motor Control
  - Introduction
    - Field oriented control basics and design
    - Sensorless PMSM control by position estimation using saliency based back-EMF
    - Output voltage generation: DC bus ripple elimination & space vector modulation performance
    - Current measurement
    - Possible Enhancements of the Algorithm
- Performance
- Summary
How to Control a PMSM Motor

1. Measure and obtain state, phase currents and DC bus voltage

2. With measured currents, determine current stator flux vector

3. With measured currents, estimate the actual rotor position

4. With rotor position, determine ideal stator flux vector which is oriented at 90° with respect to rotor flux

5. Calculate 3 phase voltages to be applied to achieve this stator flux vector

6. Apply voltages to the 3 phases with associated PWM signals
Why Field Oriented Control?

For a PMSM motor, the three sinusoidal phase currents have to be controlled to create a flux vector which is perpendicular to the rotor flux current.

• To control the three sinusoidal currents independently would be a very complex mathematical task

• FOC simplifies the math by transforming the 3 phase system to a DC motor system viewing angle

• It decomposes the stator current into:
  - A magnetic field-generating part
  - A torque generating part

• Both components can be easily controlled separately after decomposition
Sensorless PMSM Motor Control Principle on S12ZVM

MC9S12ZVM Board

Power line

3-phase Inverter

ADC Module 0
ADC Module 1

Current Calculation

Position Estimation

Forward Transformation
Torque Control
Reverse Transformation

FOC

Ripple Elimination & Modulation

GDU

CPMU

PhaseA Current
DC Bus Voltage
PhaseB Current
TempSense

ADC Triggers

u_{DC Bus}

u_a

u_b

u_c

i_a

i_b

i_c

Rotor Angle

Actual Speed

Required Speed

I_q^*

I_d^* = 0

Speed PI Controller

BDM

LIN Transc.
LIN Drv

On-Chip Run-time Debugging Freemaster

Superior System

S12ZVM

u_{DC Bus}

u_a

u_b

u_c

u_d

u_q

PTU

Sensorless System

Sensorless PMSM Motor Control Principle on S12ZVM

MC9S12ZVM Board

Power line

3-phase Inverter

ADC Module 0
ADC Module 1

Current Calculation

Position Estimation

Forward Transformation
Torque Control
Reverse Transformation

FOC

Ripple Elimination & Modulation

GDU

CPMU

PhaseA Current
DC Bus Voltage
PhaseB Current
TempSense

ADC Triggers

u_{DC Bus}

u_a

u_b

u_c

i_a

i_b

i_c

Rotor Angle

Actual Speed

Required Speed

I_q^*

I_d^* = 0

Speed PI Controller

BDM

LIN Transc.
LIN Drv

On-Chip Run-time Debugging Freemaster

Superior System

S12ZVM

u_{DC Bus}

u_a

u_b

u_c

u_d

u_q

PTU

Sensorless System
Agenda

- S12ZVM Motor Control Family Overview
- Special Motor Control Features
  - Supporting digital modules and ADC
  - Integrated high voltage analog modules
- Sensorless PMSM Motor Control
  - Introduction
  - Field oriented control basics and design
    - Sensorless PMSM control by position estimation using saliency based back-EMF
    - Output voltage generation: DC bus ripple elimination & space vector modulation performance
    - Current measurement
    - Possible Enhancements of the Algorithm
- Performance
- Summary
Sensorless PMSM Motor Control Principle on S12ZVM

MC9S12ZVM Board

3-phase Inverter

ADC Module 0
ADC Module 1

Current Calculation

Position Estimation

BDM
LIN Transc.
LIN Drv

On-Chip Run-time Debugging Freemaster

Superior System

ADC Triggers

ADC Triggers

PhaseA Current
PhaseB Current
DC Bus Voltage
TempSense

Volt. Divider

CPMU

PMF

PTU

GDU

FOC

Forward Transformation

Torque Control

Reverse Transformation

Ripple Elimination & Modulation

Sensorless PMSM Motor

Power line

Sensorless PMSM Motor Control Principle on S12ZVM

MC9S12ZVM Board

3-phase Inverter

ADC Module 0
ADC Module 1

Current Calculation

Position Estimation

BDM
LIN Transc.
LIN Drv

On-Chip Run-time Debugging Freemaster

Superior System

ADC Triggers

ADC Triggers

PhaseA Current
PhaseB Current
DC Bus Voltage
TempSense

Volt. Divider

CPMU

PMF

PTU

GDU

FOC

Forward Transformation

Torque Control

Reverse Transformation

Ripple Elimination & Modulation

Sensorless PMSM Motor

Power line
FOC Transformation Sequencing

Phase A
Phase B
Phase C

3-Phase to 2-Phase
Stationary to Rotating
Control Process
Rotating to Stationary
SVM

3-Phase System
2-Phase System
3-Phase System

Stationary Reference Frame
Rotating Reference Frame
Stationary Reference Frame

AC
DC
AC
Creation of Rotating Magnetic Field

- The space-vectors can be defined for all motor quantities

\[ \bar{i}_s = i_A e^{j0} + i_B e^{j120^\circ} + i_C e^{j240^\circ} \]
Creating Space Vectors

- Because a space vector is defined in a plane (2D), it is sufficient to describe a space vector in a 2-axis (α,β) coordinate system.

\[
\vec{i}_s = i_A e^{j\theta} + i_B e^{j12\theta} + i_C e^{j24\theta}
\]
Transformation to 2-ph Stationary Frame

3ph currents / MMF

3ph quantities

Stationary 2ph quantities
Transformation to 2-ph Synchronous Frame

- Position and amplitude of the stator flux/current vector is fully controlled by two DC values.
Transformation to 2-ph Synchronous Frame

• Position and amplitude of the stator flux/current vector is fully controlled by two DC values.
Application Using Forward Clarke Transformation

Motor Control Library Function:

```c
void GMCLIB_Clarke_F16(SWLIBS_2Syst_F16 *const pOut, const SWLIBS_3Syst_F16 *const pln);
```

<table>
<thead>
<tr>
<th>input</th>
<th>Pointer to the structure containing data of the three-phase stationary system (f16A-f16B-f16C). Arguments of the structure contain fixed point 16-bit values.</th>
</tr>
</thead>
<tbody>
<tr>
<td>output</td>
<td>Pointer to the structure containing data of the two-phase stationary orthogonal system (β-α). Arguments of the structure contain fixed point 16-bit values.</td>
</tr>
</tbody>
</table>

Called in ATD interrupt:
```c
static tBool focFastLoop(pmsmDrive_t *ptr)
{
    ...
    GMCLIB_Clarke(&ptr->iAlBeFbck,&ptr->iAbcFbck);
    ptr->thTransform.f16Arg1 = GFLIB_Sin(ptr->pospeControl.thRotEl);
    ptr->thTransform.f16Arg2 = GFLIB_Cos(ptr->pospeControl.thRotEl);
    GMCLIB_Park(&ptr->iDQFbck,&ptr->thTransform,&ptr->iAlBeFbck);
    ...
    return (true);
}
```
Application Using Forward Park Transformation

Rotor position

Is_beta -> Forward Park Transformation -> Is_d
Is_alpha -> Is_q

Motor Control Library Function:

```c
void GMCLIB_Park_F16
(SWLIBS_2Syst_F16 *pOut,
 const SWLIBS_2Syst_F16 *const pInAngle,
 const SWLIBS_2Syst_F16 *const pIn);
```

Called in ATD interrupt:
```
static tBool focFastLoop(pmsmDrive_t *ptr)
{
    ...  
    GMCLIB_Clar(&ptr->iAlBeFbck,&ptr->iAFCbck);
    ptr->thTransform.f16Arg1 = GFLIB_Sin(ptr->pospeControl.thRotEl);
    ptr->thTransform.f16Arg2 = GFLIB_Cos(ptr->pospeControl.thRotEl);
    GMCLIB_Park(&ptr->iDQFbck,&ptr->thTransform,&ptr->iAlBeFbck);
    ...
    return (true);
}
```
FOC Transformation Sequencing

Phase A
Phase B
Phase C

3-Phase to 2-Phase
Stationary to Rotating
Control Process
Rotating to Stationary
SVM

From measurement

AC
DC
AC

Stationary Reference Frame
Rotating Reference Frame
Stationary Reference Frame
FOC Design - 3-phase PMSM Model

• Considering sinusoidal 3-phase distributed winding and neglecting effect of magnetic saturation and leakage inductances

Stator voltage equations

\[
\begin{bmatrix}
  u_A \\
  u_B \\
  u_C
\end{bmatrix}
= R \begin{bmatrix}
  i_A \\
  i_B \\
  i_C
\end{bmatrix} + \frac{d}{dt} \begin{bmatrix}
  \psi_A \\
  \psi_B \\
  \psi_C
\end{bmatrix}
\]

Forward Clarke

Stator linkage flux

\[
\begin{bmatrix}
  \psi_A \\
  \psi_B \\
  \psi_C
\end{bmatrix} = \begin{bmatrix}
  L_{aa} & L_{ab} & L_{ac} \\
  L_{ba} & L_{bb} & L_{bc} \\
  L_{ca} & L_{cb} & L_{cc}
\end{bmatrix} \begin{bmatrix}
  i_A \\
  i_B \\
  i_C
\end{bmatrix} + \Psi_{PM}
\]

Internal motor torque

\[
T_i = \frac{P_i}{\omega_m} = \frac{P_p}{\omega_e} (u_{iA}i_A + u_{iB}i_B + u_{iC}i_C)
\]

\[
T_i = P_p( -\Psi_{PM}i_A \sin(\theta_e) - \Psi_{PM}i_B \sin(\theta_e - \frac{2}{3}\pi) - \Psi_{PM}i_C \sin(\theta_e + \frac{2}{3}\pi))
\]
FOC Design: 2-phase PMSM Model

- Considering sinusoidal 2-phase distributed winding and neglecting effect of magnetic saturation and leakage inductances

Stator voltage equations

\[
\begin{bmatrix}
  u_{\alpha} \\
  u_{\beta}
\end{bmatrix} = \begin{bmatrix}
  R \\
  \frac{d}{dt}
\end{bmatrix} \begin{bmatrix}
  i_{\alpha} \\
  i_{\beta}
\end{bmatrix} + \begin{bmatrix}
  \psi_{\alpha} \\
  \psi_{\beta}
\end{bmatrix}
\]

Stator linkage flux

\[
\begin{bmatrix}
  \Psi_{s\alpha} \\
  \Psi_{s\beta}
\end{bmatrix} = \begin{bmatrix}
  L_S & 0 \\
  0 & L_S
\end{bmatrix} \begin{bmatrix}
  i_{\alpha} \\
  i_{\beta}
\end{bmatrix} + \Psi_{PM} \bigg|_{i_{sa}=0} \begin{bmatrix}
  \cos \theta_{re} \\
  \sin \theta_{re}
\end{bmatrix}
\]

Internal motor torque

\[
T_i = \frac{3}{2} \frac{p_p}{\omega_e} (u_{i\alpha} i_{\alpha} + u_{i\beta} i_{\beta}) = \frac{3}{2} p_p \left( \psi_{\alpha} i_{\beta} - \psi_{\beta} i_{\alpha} \right)
\]
Sinusoidal PM Motor Model in \( dq \) Synchronous Frame

Salient machine model in \( dq \) synchronous frame aligned with the rotor

- **Stator Voltage Equations**
  \[
  \begin{bmatrix}
    u_d \\
    u_q
  \end{bmatrix} = \begin{bmatrix}
    i_d \\
    i_q
  \end{bmatrix} + \begin{bmatrix}
    s & \omega_e \\
    -\omega_e & s
  \end{bmatrix} \begin{bmatrix}
    \psi_d \\
    \psi_q
  \end{bmatrix}
  \]

- **Stator Flux Linkages of Salient Machine**
  \[
  \begin{bmatrix}
    \psi_d \\
    \psi_q
  \end{bmatrix} = \begin{bmatrix}
    L_d & 0 \\
    0 & L_q
  \end{bmatrix} \begin{bmatrix}
    i_d \\
    i_q
  \end{bmatrix} + \psi_{PM} \begin{bmatrix}
    1 \\
    0
  \end{bmatrix}
  \]

- **Resulting stator voltage equations**
  \[
  \begin{bmatrix}
    u_d \\
    u_q
  \end{bmatrix} = R_s \begin{bmatrix}
    i_d \\
    i_q
  \end{bmatrix} + \begin{bmatrix}
    sL_d & 0 \\
    0 & sL_q
  \end{bmatrix} \begin{bmatrix}
    i_d \\
    i_q
  \end{bmatrix} + \omega_e \begin{bmatrix}
    -L_q \\
    L_d
  \end{bmatrix} \begin{bmatrix}
    i_q
  \end{bmatrix} + \omega_e \psi_{PM} \begin{bmatrix}
    0 \\
    1
  \end{bmatrix}
  \]

R-L circuit  cross-coupling backEMF

- **Internal motor torque**
  \[
  T_i = \frac{3}{2} \frac{p_p}{\omega_e} (u_{id} i_d + u_{iq} i_q) = \frac{3}{2} p_p (\psi_d i_q - \psi_q i_d) = \frac{3}{2} p_p \cdot \psi_{PM} i_q
  \]
PMSM Current Control

\[
\begin{bmatrix}
   u_d \\
   u_q
\end{bmatrix} = R_s \begin{bmatrix}
   i_d \\
   i_q
\end{bmatrix} + \begin{bmatrix}
   sL_d & 0 \\
   0 & sL_q
\end{bmatrix} \begin{bmatrix}
   i_d \\
   i_q
\end{bmatrix} + \omega_e \begin{bmatrix}
   -L_q \\
   L_d
\end{bmatrix} \begin{bmatrix}
   i_q \\
   i_d
\end{bmatrix} + \omega_e \psi_{PM} \begin{bmatrix}
   0 \\
   1
\end{bmatrix}
\]

- **R-L circuit**
- **Cross-coupling backEMF**

**Independent control of DQ currents**

Transfer functions of PI controller and RL model in ‘s’ domain

**Two axis components of required current vector**

```
G(s) = \frac{K_p s + K_i}{s^2 + \omega_n^2}
```

Resulting Transfer function in ‘s’ domain
Zero Cancelation

- Controller gain design can be done by matching coefficients of characteristic polynomial with those of an ideal 2\textsuperscript{nd} order system

Transfer function of current loop

\[
G(s) = \frac{\frac{K_P}{L} s + \frac{K_I}{L}}{s^2 + \left(\frac{K_P + R}{L}\right)s + \frac{K_I}{L}} = \frac{\frac{K_I}{L} \left(\frac{K_P}{K_I} s + 1\right)}{s^2 + \left(\frac{K_P + R}{L}\right)s + \frac{K_I}{L}}
\]

Transfer function of ideal 2\textsuperscript{nd} order system

\[
G_{ideal}(s) = \frac{\omega_0^2}{s^2 + 2\xi\omega_0 s + \omega_0^2}
\]

- “Zero” introduced by PI controller at \(-\frac{K_P}{K_I}\) adds derivative behavior to the closed loop, creating overshoot during step response

\[\xi\] – is damping factor
\[\omega_0\] – is natural frequency
Zero Cancellation

- Zero Cancellation placed in the feed-forward path will be designed to compensate the closed loop zero with unity DC gain.

\[
G(s) = \frac{1}{\left(\frac{K_P}{K_I} s + 1\right)} \times \frac{K_I \left( \frac{K_P}{K_I} s + 1 \right)}{s^2 + \left( \frac{K_P + R}{L} \right) s + \frac{K_I}{L}} = \frac{K_I}{L} \frac{1}{s^2 + \left( \frac{K_P + R}{L} \right) s + \frac{K_I}{L}}
\]
PI Controller Gain Calculation

- Implementation of Zero Cancellation allows precise matching of characteristic polynomial coefficients
- Enables simple tuning of the current loop bandwidth and attenuation

\[ G(s) = \frac{K_I}{L} \frac{L}{s^2 + \left(\frac{K_P + R}{L}\right)s + \frac{K_I}{L}} \]

\[ G_{\text{ideal}}(s) = \frac{\omega_0^2}{s^2 + 2\xi\omega_0 s + \omega_0^2} \]

**PI controller gains**

\[ K_I = \omega_0^2 L \]
\[ K_P = 2\xi\omega_0 L - R \]
Application Using PI Controller

Motor Control Library Function:

tFrac16 GFLIB_ControllerPlrAW_F16
(tFrac16 f16InErr, 
GFLIB_CONTROLLER_PIAW_R_T_F16 * pParam)

Called in ATD interrupt:
static tBool focFastLoop(pmsmDrive_t *ptr)
{
    ...
    ptr->uDQReq.f16Arg1 = GFLIB_ControllerPlrAW(ptr->iDQErr.f16Arg1,&ptr->dAxisPI);
    ...
    return (true);
}
FOC SW Implementation

CPU control task decoupled from synchronous control loop

PMF
- PMF phA out
- PMF phB out

PTU
- PTU counter
- PTU triggers

ADC
- Phase current A sampling
  - U_{DCBus} sampling
  - ADC0 conversion
- Phase current B sampling
  - Temp sampling
  - ADC1 conversion

DMA 0

CPU
- ISR service routine

FOC calculations

ADC DMA

ISRADC1()
[Run_state]

Read measured I_{PhaseAB}, U_{DCBus} and Temp from ADC Result List

Calculate I_{PhaseC}

Calculate actual position
BEMF Observer

Forward Clarke Transformation
3-phase stationary to 2-phase stationary

Forward Park Transformation
2-phase stationary to rotational frame

Control d- and q- current component
d- and q- PI control

Reverse Park Transformation
Od d- and q- stator voltage to α,β frame

Filter DCBus voltage

DCBus ripple elimination on U_{α,β}

Space vector modulation on U_{α,β}

Update duty cycle

Update

return
Agenda

- S12ZVM Motor Control Family Overview
- Special Motor Control Features
  - Supporting digital modules
  - Integrated high voltage analog modules
- Sensorless PMSM Motor Control
  - Introduction
  - Field oriented control basics and design
  - Sensorless PMSM control by position estimation using saliency based back-EMF
    - Output voltage generation: DC bus ripple elimination & space vector modulation performance
    - Current measurement
    - Possible Enhancements of the Algorithm
- Performance
- Summary
Sensorless PMSM Motor Control Principle on S12ZVM

MC9S12ZVM Board

Power line

3-phase Inverter

ADC Module 0
ADC Module 1

Current Calculation

Position Estimation

Current

ADC Triggers

PhaseA Current
DC Bus Voltage
PhaseB Current
TempSense

Vol. Divider
Current Amp.

GDU

CPMU

PMF

PTU

On-Chip
Run-time
Debugging
Freemaster

Superior
System

BDM

LIN
Transc.

LIN
Drv

TempSense

Actual
Speed

Required
Speed

Speed
PI Controller

Forward
Transformation

FOC

Torque
Control

Reverse
Transformation

Ripple
Elimination
& Modulation

S12ZVM

u_a

u_b

i_a

i_b

i_c

u_DC Bus

u_a

u_b

i_d

i_q

I_q^*

I_d^*=0

u_DC Bus

u_DC Bus

u_DC Bus

u_DC Bus

u_DC Bus

u_DC Bus

u_DC Bus

u_DC Bus

u_DC Bus

u_DC Bus

u_DC Bus

u_DC Bus

u_DC Bus

u_DC Bus
Rotor Position Sensor Elimination: Introduction

- **FOC** requires accurate position and velocity signals

- Sensorless FOC application uses
  - Estimated position
  - Estimated speed

- Position/speed is estimated from measured currents and measured/estimated voltages
Classification of Sensorless Methods

- **Model-based methods**
  - Based on the electrical model of PM
  - Presets good results in medium and high speed operation (starting from 5% of nominal speed)
  - Broadly commercialized for low-end applications

- **Methods relaying on magnetic saliency**
  - Based on inherent characteristic of PM motor called magnetic saliency
  - Presets good results in standstill or very low speed region (up to 10% of nominal speed)
  - Commercialized for certain types of PM motors designed accordingly
Model-based Methods Classification

- Back EMF estimation methods
  - Utilize simplified model of PM in d-q estimated frame or so called extended BEMF (Freescale approach)

- Voltage/current model methods
  - Difference between the estimated voltage/current and the actual voltage/current is used to extract the position error

- Fundamental voltage feedback methods
  - Output of the current controller is directly used to obtain position error
Saliency Based Back-EMF Observer

- Saliency based back-EMF voltage is generated due to $L_d \neq L_q$
- Because back-EMF term is not modeled, observer actually acts as a back-EMF state filter
- Observer is designed in synchronous reference frame; i.e. all observer quantities are DC in steady state, making the observer accuracy independent of rotor speed

\[
\begin{bmatrix}
    u_\gamma \\
    u_\delta
\end{bmatrix} =
\begin{bmatrix}
    R_s + sL_d & -\omega_\gamma L_q \\
    \omega_\gamma L_q & R_s + sL_d
\end{bmatrix}
\begin{bmatrix}
    i_\gamma \\
    i_\delta
\end{bmatrix} +
E_{sal}
\begin{bmatrix}
    -\sin(\theta_{err}) \\
    \cos(\theta_{err})
\end{bmatrix}
\]

\[
\frac{dL}{d\theta} \text{ causes } \frac{d\lambda}{d\theta}, \text{ which when combined with } \frac{d\theta}{dt}, \text{ causes } \frac{d\lambda}{dt} = \text{ voltage}
\]
Position Estimation Using Saliency Based Back-EMF

Position estimation steady state error at constant speed

\[
\theta_{err,ss} = \lim_{s \to 0} \left[ \frac{\theta_c(s)s^3}{s^2 + K_ps + K_i} \right] = 0
\]

Position estimation steady state error during speed ramp change

\[
\theta_{err,ss} = \lim_{s \to 0} \left[ \frac{s^2}{s^2 + K_ps + K_i s^2} A \right] = \frac{A}{K_i}
\]
Application using E-BEMF Observer

Advanced Motor Control Library Function (not part of standard library):

```c
void ACLIB_PMSMBemfObsrvDQ
(&sensorless->wRotEl, &sensorless->thRotEl,
iAlBeFbck, uAlBeReq,
&sensorless->bEMFObs);
```

Called in ADC interrupt:

```c
void stateRun( )
{
    ...
    ACLIB_PMSMBemfObsrvDQ(&sensorless->wRotEl, &sensorless->thRotEl, iAlBeFbck, uAlBeReq, &sensorless->bEMFObs);
    ...
    stateRunStatus = focFastLoop(&drvFOC);
    ...
    Pmf_updateDutycycle(&drvFOC.pwm16);
}
```
Agenda

• S12ZVM Motor Control Family Overview
• Special Motor Control Features
  - Supporting digital modules
  - Integrated high voltage analog modules
• Sensorless PMSM Motor Control
  - Introduction
  - Field oriented control basics and design
  - Sensorless PMSM control by position estimation using saliency based back-EMF
  - Output voltage generation: DC bus ripple elimination & space vector modulation performance
  - Current measurement
  - Possible Enhancements of the Algorithm
• Performance
• Summary
Sensorless PMSM Motor Control Principle on S12ZVM

MC9S12ZVM Board

3-phase Inverter

ADC Module 0

ADC Module 1

Current Calculation

Position Estimation

Current

ADC Triggers

PhaseA Current

PhaseB Current

Volt. Divider

GDU

Reactive Power

PMF

PTU

Current Amp.

DC Bus Voltage

DC Bus

On-Chip Debugging Freemaster

BDM

LIN Transc.

LINDrv

Superior System

Required Speed

Actual Speed

Rotor Angle

Forward Transformation

Torque Control

Reverse Transformation

Ripple Elimination & Modulation

Speed PI Controller

I_q^*

I_d^* = 0

S12ZVM
DC-bus Ripple Compensation

• Compensates the ripple of the output voltages from power stage caused by DC-bus voltage ripples

• Improves performance of the drive

• Compensation uses moving average filtered DCBus voltage and a fixed index

\[
u_{k}^a = \begin{cases} \frac{f32\text{ModIndex} \cdot u_{a}}{f32\text{ArgDCBusMsr}/2} & \text{if } \left| f32\text{ModIndex} \cdot u_{a} \right| < \frac{f32\text{ArgDCBusMsr}}{2} \\ \text{sign}(u_{a}) & \text{otherwise} \end{cases}
\]

\[
u_{k}^\beta = \begin{cases} \frac{f32\text{ModIndex} \cdot u_{\beta}}{f32\text{ArgDCBusMsr}/2} & \text{if } \left| f32\text{ModIndex} \cdot u_{\beta} \right| < \frac{f32\text{ArgDCBusMsr}}{2} \\ \text{sign}(u_{\beta}) & \text{otherwise} \end{cases}
\]
Application Using DCBus Ripple Elimination

Motor Control Library Function:

```c
void GMCLIB_ElimDcBusRip_F16
(SWLIBS_2Syst_F16 *const pOut,
const SWLIBS_2Syst_F16 *const pIn,
const GMCLIB_ELIMDCBUSRIP_T_F16 *const pParam);
```

```c
drvFOC.elimDcbRip.f16ModIndex = FRAC16(0.866025403784439);
tBool Meas_UdcVoltageMeasure(measModule_t *ptr, GDFLIB_FILTER_MA_T *uDcbFilter)
{
    ptr->measured.f16Udcb.raw = ADC0ResultList[1]>>1;
    ptr->measured.f16Udcb.filt = GDFLIB_FilterMA(ptr->measured.f16Udcb.raw, uDcbFilter);
    return(1);
}
static tBool focFastLoop(pmsmDrive_t *ptr)
{
    ...
    ptr->elimDcbRip.f16ArgDcBusMsr = meas.measured.f16Udcb.filt;
    GMCLIB_ElimDcBusRip(&ptr->uAIBeReqDCB, &ptr->uAIBeReq, &ptr->elimDcbRip);
    ptr->svmSector = GMCLIB_SvmStd(&(ptr->pwm16), &ptr->uAIBeReqDCB);
    return (true);
}
```
Three Phase Output Voltage Generation

- The average output voltage is proportional to the duty cycle of the switch PWM.
- It is regulated to form a sinusoidal shape on all three phases to achieve optimum torque.

![Diagram of three-phase output voltage generation](image)

Many kind of modulations
SVM selected from alph bet.

Source: Strategic Technology Group, India
Sinusoidal Modulation: Limited in Amplitude

- In sinusoidal modulation, the amplitude is limited to half of the DC-bus voltage.
- The phase to phase voltage is then lower than the DC-bus voltage (although such voltage can be generated between the terminals).

Can such a modulation technique be found that would generate full phase-to-phase voltage?
Full Phase-to-Phase Voltage Generation

- Full phase-to-phase voltage can be generated by continuously shifting the 3-phase voltage system.
- The amplitude of the first harmonic can be then increased by 15.5%.
How to Increase Modulation Index

• Modulation index is increased by adding the “shifting” voltage $u_0$ to first harmonic

• “Shifting” voltage $u_0$ must be the same for all three phases, thus it can only contain $3^r$ harmonics!
Standard Space Vector Modulation Output Waveforms

• Output voltage vector is created by switching continuously between the adjacent base vectors and the “NULL” vectors so that the vectorial time-average of the asserted base SVs is equal to the commanded voltage.

• Generates maximum phase voltage 0.5773 VDC.

• Both nulls O000 and O111 are generated at each cycle.

Input & Output Waveforms
Application Using SVM

Motor Control Library Function:

```
tU32 GMCLIB_SvmStd_F32(SWLIBS_3Syst_F32 *pOut, const SWLIBS_2Syst_F32 *const pIn);
```

<table>
<thead>
<tr>
<th>Input, output</th>
<th>Pointer to the structure containing calculated duty-cycle ratios of the 3-Phase system.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>Pointer to the structure containing direct $U_\alpha$ and quadrature $U_\beta$ components of the stator voltage vector.</td>
</tr>
</tbody>
</table>

```
static tBool focFastLoop(pmsmDrive_t *ptr)
{
    GMCLIB_Clark(&ptr->iAlBeFbck,&ptr->iAbcFbck);
    ...
    GMCLIB_ParkInv(&ptr->uAlBeReq,&ptr->thTransform,&ptr->uDQReq);
    ptr->elimDcbRip.f16ArgDcBusMsr = meas.measured.f16Udcb.filt;
    GMCLIB_ElimDcBusRip(&ptr->uAlBeReqDCB,&ptr->uAlBeReq,&ptr->elimDcbRip);
    ptr->svmSector = GMCLIB_SvmStd(&(ptr->pwm16),&ptr->uAlBeReqDCB);
    return (true);
}
```
Agenda

- S12ZVM Motor Control Family Overview
- Special Motor Control Features
  - Supporting digital modules
  - Integrated high voltage analog modules
- Sensorless PMSM Motor Control
  - Introduction
  - Field oriented control basics and design
  - Sensorless PMSM control by position estimation using saliency based back-EMF
  - Output voltage generation: DC bus ripple elimination & space vector modulation performance
- Possible Enhancements of the Algorithm
- Performance
- Summary
Sensorless PMSM Motor Control Principle on S12ZVM

MC9S12ZVM Board

3-phase Inverter

ADC Module 0
ADC Module 1

Current Calculation

Position Estimation

Forward Transformation
Torque Control
Reverse Transformation

ROCR Angle

FOC

Ripple Elimination & Modulation

GDU

CPMU

PMF

PTU

ADC Triggers
PhaseA Current
PhaseB Current
TempSense
Volt. Divider
Current Amp.

Power line

3-phase Inverter

ADC Triggers

On-Chip Run-time Debugging Freemaster

Superior System

BDM
LIN Transc.
LIN Drv

Required Speed

Actual Speed

Forward
Reverse

Linearization

Ripple Elimination & Modulation

Speed PI Controller

I_d
I_q
I_d=0

Sensorless PMSM Motor

S12ZVM

MC9S12ZVM Board

3-phase Inverter

ADC Module 0
ADC Module 1

Current Calculation

Position Estimation

Forward Transformation
Torque Control
Reverse Transformation

ROCR Angle

FOC

Ripple Elimination & Modulation

GDU

CPMU

PMF

PTU

ADC Triggers
PhaseA Current
PhaseB Current
TempSense
Volt. Divider
Current Amp.

Power line

3-phase Inverter

ADC Triggers

On-Chip Run-time Debugging Freemaster

Superior System

BDM
LIN Transc.
LIN Drv

Required Speed

Actual Speed

Forward
Reverse

Linearization

Ripple Elimination & Modulation

Speed PI Controller

I_d
I_q
I_d=0

Sensorless PMSM Motor

S12ZVM
PMSM Sensorless Application Example Timing

Two shunts current sensing

- PWM counter
- PMF phA out
- PMF phB out
- PMF half cycle reload every fourth opportunity
- deadtime
- PMF half cycle reload every fourth opportunity
- PTU counter
- PTU triggers
- Delay in order for DMA to load ADC list
- ADC0 conversion
- ADC0 sampling
- Phase current A sampling
- \( U_{DCBus} \) sampling
- ADC1 conversion
- ADC1 sampling
- Temp sampling
- EOC interrupt
- \( EOC \) interrupt
- ISR service routine
- FOC calculations
- application
- Free for application use

ADC0

ADC1

CPU

PMF

PTU

SW Serviced Hardware Events

Preset Hardware Events

Autonomous Hardware Events
Current Sensing

- Bottom transistor must be switched on at least for a critical pulse width to get stabilized current shunt resistor voltage drop
- At any time, this rule needs to be accomplished for the legs where the shunts are located.
- Minimum pulse width defined by system delays and ADC sampling time
Delays Involved in PWM Driven Closed Loops

- Delays are chained and are caused by:
  - Dead time insertion
  - Opto-coupler propagation delay
  - MOSFET driver propagation delay
  - MOSFET turn ON/OFF times
  - Amplifier slew rate
  - Low-pass filter delay
  - ADC delays

Overall delay: ~0.4 ÷ 6 us
Delays Involved in PWM Driven Closed Loops

Current Sensing Shunts in Inverter Legs

- Internal counter
- Desired PWM
- Complementary pair with dead time inserted (signals at pins)
- IGBT/MOSFET gate signals top / bottom
- Motor phase terminal positive / negative current
- Motor phase current positive / negative - shunts in legs
- Real feedback signal at ADC pin

Mid point shifts
Delay1 = $T_{\text{opto}} + T_{\text{preDRV}} = \sim 300 \text{ ns} + \sim 50 \text{ ns}$

Delay2 = $T_{\text{opto}} + T_{\text{preDRV}} + T_{\text{on}}/2 + \text{DT}$
Delay2 = $\sim 350 \text{ ns} + (0.1\sim 1 \text{ us})/2 + (0.3\sim 4 \text{ us})$
Delay2 = $\sim 0.5 \sim 5 \text{ us}$

Delay3 = $T_{\text{opto}} + T_{\text{preDRV}} + T_{\text{on}}/2 = 0.1\sim 1 \text{ us}$

Mid point shifts
$= T_{\text{opto}} + T_{\text{preDRV}} + (T_{\text{on}} + T_{\text{off}})/4 + (\text{DT1 or DT2})/2$
$= \sim 300 \text{ ns} + \sim 50 \text{ ns} + (0.2\sim 2 \text{ us})/4 + (0.3\sim 4 \text{ us})/2$
$= \sim 0.4 \sim 5 \text{ us}$

Low pass filter delay + $T_{\text{opto}}$: \sim 1 \text{ us}

FSL note - in our ref designs this is close to 0

Amplifier slew rate: <1 \text{ us/ full range}
Agenda

- S12ZVM Motor Control Family Overview
- Special Motor Control Features
  - Supporting digital modules
  - Integrated high voltage analog modules
- Sensorless PMSM Motor Control
  - Introduction
  - Field oriented control basics and design
  - Sensorless PMSM control by position estimation using saliency based back-EMF
  - Output voltage generation: DC bus ripple elimination & space vector modulation performance
  - Current measurement

- Possible Enhancements of the Algorithm

- Performance
- Summary
Possible enhancements of the Algorithm

- Following code enhancements can be added to the SW:

**Advanced motor control & sensor enhancement:**

- Field Weakening
- Single-shunt 3-phs currents reconstruction
- Max. Torque per Amp (MTPA)
- Torque Ripple elimination
- Zero rpm
Agenda

• S12ZVM Motor Control Family Overview
• Special Motor Control Features
  - Supporting digital modules
  - Integrated high voltage analog modules
• Sensorless PMSM Motor Control
  - Introduction
  - Field oriented control basics and design
  - Sensorless PMSM control by position estimation using saliency based back-EMF
  - Output voltage generation: DC bus ripple elimination & space vector modulation performance
  - Current measurement
  - Possible Enhancements of the Algorithm

• Performance

• Summary
Core Utilization on S12ZVM

- System Setup:
  - Two shunt current sensing
  - Running at 100[MHz] core clock
  - PWM frequency of 20[kHz] with reload every second opportunity
  - Current loop calculated every 100[µs]
  - Speed loop calculated every 1[ms]
  - Cosmic compiler employed

<table>
<thead>
<tr>
<th></th>
<th>Execution time [µs]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AVG</td>
</tr>
<tr>
<td>Two shunt measurement including post processing</td>
<td>1.7</td>
</tr>
<tr>
<td>DC-bus voltage measurement including Filtering</td>
<td>2.2</td>
</tr>
<tr>
<td>FOC algorithm current loop (fast loop 100µs) including SVM and DC-bus ripple elimination</td>
<td>28.8</td>
</tr>
<tr>
<td>FOC speed loop (slow loop 1ms)</td>
<td>3.8</td>
</tr>
<tr>
<td>E-BEMF observer</td>
<td>27.7</td>
</tr>
<tr>
<td>Fault checking</td>
<td>5</td>
</tr>
<tr>
<td>SUM</td>
<td>69</td>
</tr>
</tbody>
</table>
Agenda

• S12ZVM Motor Control Family Overview
• Special Motor Control Features
  – Supporting digital modules
  – Integrated high voltage analog modules
• Sensorless PMSM Motor Control
  – Introduction
  – Field oriented control basics and design
  – Sensorless PMSM control by position estimation using saliency based back-EMF
  – Output voltage generation: DC bus ripple elimination & space vector modulation performance
  – Current measurement
  – Possible Enhancements of the Algorithm

• Performance

• Summary
Summary: Simplify Your Design

- Minimized system cost: Small PCB, Minimum external components

- Scalable approach (memory, boost, bypass, communication interface)

- CPU offloaded from motor control timing tasks due to autonomous motor control peripherals

- Complete Software & Hardware Enablement with ready to use

- Higher reliability
For Further Information

• Visit www.freescale.com/motorcontrol

• Welcome to related FTF motor control enablement sessions:

<table>
<thead>
<tr>
<th>Session no.</th>
<th>title</th>
<th>Duration [hrs]</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTF-AUT-F0233</td>
<td>Freescale Automotive Motor Control Enablement Solutions</td>
<td>2</td>
<td>Tue, 2pm</td>
</tr>
<tr>
<td>FTF-AUT-F0014</td>
<td>Overview of FreeMASTER and RAppID Bootloader</td>
<td>1</td>
<td>Wed, 4.15 pm</td>
</tr>
<tr>
<td>FTF-AUT-F0076</td>
<td>Hands-On Workshop: Motor Control Toolbox Overview (Reserved Seat Required)</td>
<td>3</td>
<td>Fri, 9.30 am</td>
</tr>
</tbody>
</table>