Module 15: C28x Digital Motor Control

32-Bit-Digital Signal Controller
TMS320F2812

Texas Instruments Incorporated
European Customer Training Centre
University of Applied Sciences Zwickau (FH)
Motor Classification:

- Direct Current Motors (DC)
- Alternating Current Motors (AC)
  - Asynchronous Induction Motor (ACI)
  - Permanent Magnet Synchronous Motor (PMSM)
  - Synchronous Brushless DC Motor (BLDC)
For most three phase machines, the winding is stationary, and magnetic field is rotating.

Three phase machines have three stator windings, separated 120° apart physically.

Three phase stator windings produce three magnetic fields, which are spaced 120° in time.

\[
\begin{align*}
    i_a &= I_s e^{j \omega t} \\
    i_b &= I_s e^{j \omega t - \frac{2\pi}{3}} \\
    i_c &= I_s e^{j \omega t - \frac{4\pi}{3}} \\
    e_a &= E e^{j p \Omega t} \\
    e_b &= E e^{j p \Omega t - \frac{2\pi}{3}} \\
    e_c &= E e^{j p \Omega t - \frac{4\pi}{3}}
\end{align*}
\]
Three stationary pulsating magnetic fields

- The three phase winding produces three magnetic fields, which are spaced 120° apart physically.
- When excited with three sine waves that are a 120° apart in phase, there are three pulsating magnetic fields.
- The resultant of the three magnetic fields is a rotating magnetic field.
Rotor is carrying a constant magnetic field created either by permanent magnets or current fed coils.

The interaction between the rotating stator flux, and the rotor flux produces a torque which will cause the motor to rotate.

- The rotation of the rotor in this case will be at the same exact frequency as the applied excitation to the rotor.
- This is synchronous operation.

Rotor speed (rad/s) \( \Omega = \frac{\omega}{p} \) gives \( \frac{60.f}{p} \) (r.pm)

- \( f \): AC supply frequency (Hz)
- \( p \): motor poles pair per phase

Example: a 2 poles pair synchronous motor will run at 1500 r.pm for a 50Hz AC supply frequency
Both (typically) have permanent-magnet rotor and a wound stator

BLDC (Brushless DC) motor is a permanent-magnet brushless motor with trapezoidal back EMF

PMSM (Permanent-magnet synchronous motor) is a permanent-magnet brushless motor with sinusoidal back EMF
3 – Phase Voltage Inverter

Six PWM signals to control Power Switches

DC - Voltage

Upper & lower devices can not be turned on simultaneously (dead band)

3 - phase outputs to motor terminals

Power Switching Devices
Scalar Control Scheme ("V/f")

- Simple to implement: All you need is three sine waves feeding the motor
- Position information not required (optional).
- Doesn’t deliver good dynamic performance.
- Torque delivery not optimized for all speeds
**Field Oriented Control (FOC)**

- Field Oriented Control (FOC) or Vector Control, is a control strategy for 3-phases induction motors where the torque producing and magnetizing components of the stator flux are separately controlled.

- The approach consists in imitating the DC motors’ operation

- FOC will be possible with system information: currents, voltages, flux and speed.
FOC control scheme

Some key mathematical components are required!
PARK transform (1929):

- \((V_s)\): voltage vector applied to motor stator (index \(s\))

- Park transform is a referential change

\[
\begin{bmatrix}
\vec{v}_{s1} \\
\vec{v}_{s2} \\
\vec{v}_{s3}
\end{bmatrix}
= \begin{bmatrix}
\cos \theta_s & -\sin \theta_s & 1 \\
\cos(\theta_s - \frac{2\pi}{3}) & -\sin(\theta_s - \frac{2\pi}{3}) & 1 \\
\cos(\theta_s - \frac{4\pi}{3}) & -\sin(\theta_s - \frac{4\pi}{3}) & 1 \\
\end{bmatrix}
\begin{bmatrix}
\vec{v}_{sd} \\
\vec{v}_{sq} \\
\vec{v}_{so}
\end{bmatrix}
\]

\[
\begin{bmatrix}
\vec{v}_{s1} \\
\vec{v}_{s2} \\
\vec{v}_{s3}
\end{bmatrix}
= [P(\theta_s)]
\begin{bmatrix}
\vec{v}_{sd} \\
\vec{v}_{sq} \\
\vec{v}_{so}
\end{bmatrix}
\]

\[
\begin{bmatrix}
\vec{v}_{s1} \\
\vec{v}_{s2} \\
\vec{v}_{s3}
\end{bmatrix}
= [P(\theta_s)]^\dagger
\begin{bmatrix}
\vec{v}_{s1} \\
\vec{v}_{s2} \\
\vec{v}_{s3}
\end{bmatrix}
\]

\(\theta_s = \omega_s t\)

\(v_{so} = 0\)
(\(v_{sd}, v_{sq}, v_{so}\)) are called the Park coordinates

- \(v_{sd}\): direct Park component
- \(v_{sq}\): squaring Park component
- \(v_{so}\): homo-polar Park component

\(v_{so}\) is null for a three-phases balanced system

Each pair of components is perpendicular to each other

\[\begin{align*}
\vec{v}_{s1} + \vec{v}_{s2} + \vec{v}_{s3} &= \vec{0} \quad \text{(tri - phases balanced system)} \\
\vec{v}_{sd} \cdot \vec{v}_{sq} &= 0 \\
\vec{v}_{sd} \cdot \vec{v}_{so} &= 0 \\
\vec{v}_{sq} \cdot \vec{v}_{so} &= 0
\end{align*}\]
Transform is usually split into CLARKE transform and one rotation

CLARKE converts balanced three phase quantities into balanced two phase orthogonal quantities

\[\theta_S = \omega_S t\]

\[v_\alpha = v_1\]

\[v_\beta = \frac{2v_2 + v_1}{\sqrt{3}}\]

\[
\begin{bmatrix}
  v_d \\
  v_q \\
\end{bmatrix} = \begin{bmatrix}
  \cos(\theta_S) & \sin(\theta_S) \\
  -\sin(\theta_S) & \cos(\theta_S)
\end{bmatrix} \begin{bmatrix}
  v_\alpha \\
  v_\beta
\end{bmatrix}
\]
PARK Transform summary

- **Stator phase current example**: $i_s$ is moving at $\theta_s$ and its PARK coordinates are constant in (d,q) rotating frame.

- Can be applied on any three-phase balanced variables (flux...)

Three phase rotating domain \[ v_1 \rightarrow v_2 \rightarrow v_3 \]

Two phase rotating domain \[ v_\alpha \rightarrow v_\beta \]

Stationary domain \[ V_d \rightarrow V_q \]

\[ \begin{align*}
  v_1 & = v_{1s} \\
  v_2 & = v_{2s} \\
  v_3 & = v_{3s} \\
  v_\alpha & = v_{\alpha s} \\
  v_\beta & = v_{\beta s} \\
  V_d & = V_{ds} \\
  V_q & = V_{qs} \\
  \theta_s & = \omega_s t
\end{align*} \]
“C2000 - Digital Motor Control Library (DMC)” :

- Single Phase ACI Motor Control Using Constant V/Hz
- 3-Phase ACI Motor Constant V/Hz Control
- 3-Phase ACI Motor Field Oriented Control
- 3-Phase Sensored Field Oriented Control (PMSM)
- 3-Phase Sensorless Field Oriented Control (PMSM)
- 3-Phase Sensored Trapezoidal Control (BLDC)
- 3-Phase Sensorless Trapezoidal Control (BLDC)
Digital Motor Control Library (DMC-Lib)

- The DMC-Library is a collection of most commonly used algorithms and function blocks for motor control systems.

- For every algorithm and function:
  - Essential theoretical background information
  - Data types for input/output parameters with numerical range and precision
  - Function prototypes and calling conventions
  - code size (program and data memory)
  - Build Level based code examples
The DMC-Lib contains

- PID regulators,
- Clarke transformers,
- Park transformers,
- Ramp generators,
- Sine generators,
- Space Vector generators,
- Impulse generators,

and more...
Laboratory: FOC for PMSM

TMS320F28x controller

\[ v_{\alpha_s}^* \quad v_{\beta_s}^* \]

Inv.

Space Vector Gen.

PWM Driver

PWM1
PWM2
PWM3
PWM4
PWM5
PWM6

ADCIN1
ADCIN2
ADCIN3

QEP

THETA

DRV

QEP_A
QEP_B
QEP_inc

Encoder

PMSM

Voltage Source Inverter

SPEED
FRQ

\[ \omega_r \]

QEP
THETA
DRV

\[ \theta_r \]

\[ \theta_e \]

dir

\[ i_{qs}^* \quad i_{ds}^* \quad v_{qs}^* \quad v_{ds}^* \]

PI

PI

PI

\[ T_a \quad T_b \quad T_c \]

Park

Clarke

Ileg2_
Bus
Driver

\[ i_{\alpha_s} \quad i_{\beta_s} \quad i_{qs} \quad i_{ds} \]

ADCIN1
ADCIN2
ADCIN3

PWM1
PWM2
PWM3
PWM4
PWM5
PWM6

Voltage Source Inverter

TMS320F28x controller

15 - 18
TI Library Solution “PMSM 3-1” (sprc129)

Hardware for Laboratory setup:

- Spectrum Digital eZdsp TMS320F2812
- Spectrum Digital DMC550 drive platform
- 3-phase PMSM with a QEP encoder
  - Applied Motion 40mm Alpha Motor
  - Type: A0100-104-3-100
- 24V DC power supply (DC bus voltage)
- load, e.g. DC Motor as Generator
- PC parallel port to JTAG
- 5V DC (eZdsp)
- RS232 (optional)
- Oscilloscope
TI Library Solution “PMSM 3-1” (sprc129)

- PMSM 3-1
- Load
- 24V
- DMC 550
- eZdsp
- 5V DC
- RS232 (optional)
- PC- parallel port
PMSM 3-1 Laboratory

Student Workbench
Variable transformation from stationary $\alpha\beta$-axis to synchronously rotating $dq$-axis

Variable transformation from synchronously rotating $dq$-axis to stationary $\alpha\beta$-axis

Variable transformation from phase $ab$-axis to stationary $\alpha\beta$-axis

Space-vector generator producing the duty cycle ratio of PWM signals
TI DMC Library Modules

Proportional-Integral controller

QEP driver and shaft position and rotor direction

Speed estimation from rotor position and rotor direction

PWM generation

ADC driver for two line currents and DC-bus voltage measurement
Build Level 1

TMS320F28x controller

Vq_testing

Vd_testing

Inv. Park

Space Vector Gen.

PWM Driver

T_a

T_b

T_c

PWM1

PWM2

PWM3

PWM4

PWM5

PWM6

speed_ref

Ramp control

Ramp Gen.

rmp_out

key modules under test

TMS320F28x controller
Code Composer Studio with Real Time Mode

1. Load a workspace file ‘pmsm3_1.wks’
2. In build.h, #define BUILDLEVEL LEVEL1
3. Rebuild all
4. Load program to target
   (..\pmsm3_1.out)
Code Composer Studio with Real Time Mode

5. In Debug menu, “Reset CPU” and then set “Real-time Mode”. Then, click “Yes” when the message box pops up.

6. Click “Run” icon
7. Right click on watch window. Then, check “Continuous Refresh”.

8. Set “enable_flg” to 1 in watch window. (to enable T1UF interrupt and PWM drive on DMC550)
Results: Build Level 1

Graph Property Dialog

- Display Type: Dual Time
- Graph Title: buff1 & buff2
- Interleaved Data Sources: No
- Start Address - upper display: _DLOG_4CHBuff1
- Start Address - lower display: _DLOG_4CHBuff2
- Page: Data
- Acquisition Buffer Size: 400
- Index Increment: 1
- Display Data Size: 400
- DSP Data Type: 16-bit signed integer
- Q-value: 0
- Sampling Rate (Hz): 1
- Plot Data From: Left to Right
- Left-shifted Data Display: Yes
- Autoscale: Off
- DC Value: 0
- Maximum Y-value: 40000
- Axes Display: On
- Time Display Unit: s
- Status Bar Display: On
- Magnitude Display Scale: Linear
- Data Plot Style: Line
- Grid Style: Zero Line
- Cursor Mode: Data Cursor

rmp_out

Ta

buff1 & buff2

Time

Lin Fixed Scale
Build Level 2 – Current verification

**Key components:**
- **Vq_testing**
- **Vd_testing**
- **Inv. Park**
- **Space Vector Gen.**
- **PWM Driver**
- **Voltage Source Inverter**
- **Ileg2_Bus Driver**
- **Encoder**
- **PMSM**

**Variables:**
- \( v_{\alpha s}^* \)
- \( v_{\beta s}^* \)
- \( t_a \)
- \( t_b \)
- \( t_c \)
- \( i_{ds} \)
- \( i_{qs} \)
- \( i_{\alpha s} \)
- \( i_{\beta s} \)
- \( i_a \)
- \( i_b \)
- \( \theta_e \)
- \( \text{Speed_ref} \)
- \( \text{Ramp control} \)
- \( \text{Ramp Gen.} \)
- \( \text{rmp_out} \)
- \( \text{TMS320F28x controller} \)

**Connections:**
- From **Vq_testing** to **Inv. Park**
- From **Vd_testing** to **Inv. Park**
- From **Inv. Park** to **Space Vector Gen.**
- From **Space Vector Gen.** to **PWM Driver**
- From **PWM Driver** to **Voltage Source Inverter**
- From **Voltage Source Inverter** to **Ileg2_Bus Driver**
- From **Ileg2_Bus Driver** to **Encoder**
- From **Encoder** to **PMSM**

**Notes:**
- The key module under test is highlighted.
- The diagram includes various components and their interactions.
Instructions: Build Level 2

1. Tune the 24V Power Supply to 10 Volts with 1 Amp limit

2. Load a workspace file ‘pmsm3_1.wks’

3. In build.h, #define BUILDLEVEL LEVEL2

4. Reset CPU, Compile, Load, start RTM and Run

5. Switch on 24V Power Supply

6. Set variable “enable_flg” to 1 in watch window.

7. Try to change motor speed by setting “speed_ref” (p.u.) in watch window. Then, motor should change its speed accordingly.
1. PMSM should run open-loop smoothly
2. The currents in the motor phases should be sinusoidal.
Build Level 3 - Tuning of dq-axis current closed loops

Key modules under test:

- Iq_ref
- Id_ref
- Ramp control
- Ramp Gen.
- Inv. Park
- Space Vector Gen
- PWM Driver
- Park
- Clarke
- Ileg2_Bus Driver
- Voltage Source Inverter
- ADCIN1
- ADCIN2
- ADCIN3
- Encoder
- PMSM
- TMS320F28x controller

Symbols:
- vqs
- vαs
- vβs
- Tα
- Tβ
- Tγ
- rmp_out
- ids
- vds
- iαs
- iβs
- ia
- ib
- ids
- vqs
- Iq_ref
- Id_ref
- Speed_ref
Instructions: Build Level 3

1. In `build.h`, define `BUILDLEVEL LEVEL3`

2. Compile, Load, start RTM and Run

3. Set “enable_flg” to 1 in watch window.

4. Tune-up the PI
   - Observe dq-axis current regulations at PI inputs (i.e., reference and feedback). For example, “pid1_iq.pid_ref.reg3” and “pid1_iq.pid_fdb.reg3”.
   - Try to change motor speed by setting “speed_ref” (p.u.) in watch window. Then, observe dq-axis current regulations.
Build Level 4 – Encoder verification

Iq_ref → PI → Inv. Park → Space Vector Gen. → PWM Driver → PWM1
Id_ref → PI → Park → Clarke → Ileg2_Bus Driver

Speed_ref → Ramp control → Ramp Gen

Theta_elec → QEP THETA DRV

TMS320F28x controller
Instructions: Build Level 4

1. In `build.h`, `#define` BUILDLEVEL LEVEL4
2. Compile, Load, start RTM and Run
3. Set the DC-bus to 24 Volts 1 Amp
4. Set “enable_flg” to 1 in watch window.
Results: Build Level 4

- Emulated angle VS sensed angle
Results: Build Level 4

- Speed reference VS real speed
Build Level 5 – close speed loop

- **PWM Driver**
- **Space Vector Gen.**
- **Inv.**
- **PMSM**
- **ADCIN1**
- **ADCIN2**
- **ADCIN3**
- **PWM1**
- **PWM2**
- **PWM3**
- **PWM4**
- **PWM5**
- **PWM6**
- **Voltage Source Inverter**

- **QEP_A**
- **QEP_B**
- **QEP_inc**

- **TMS320F28x controller**

- **ADCIN1**
- **ADCIN2**
- **ADCIN3**

- **θ_e**
- **dir**

- **SPEED FRQ**

- **θ_λr**

- **i_αs**

- **i_ds**

- **i_qs**

- **i_βs**

- **PWM1**
- **PWM2**
- **PWM3**
- **PWM4**
- **PWM5**
- **PWM6**

- **Encoder**

- **PMSM**
Instructions: Build Level 5

1. In build.h, `#define BUILDLEVEL LEVEL5`
2. Compile, Load, start RTM and Run
3. Set the DC-bus to 24 Volts
4. Set “enable_flg” to 1 in watch window.
Results: Build Level 5

- Fastest response time with the closed loop!
Application of PMSM 3-1: