AN13184

One PMSM Sensorless FOC and 2-ph Interleaved Boost PFC Control based on MC56F83783

Rev. 1 — 08 March 2022

Application Note

Contents

1 Overview..1
1.1 Introduction.................................1
1.2 Application features and components.................................1
1.3 NXP DSC advantages and features..2
1.4 References..................................3
1.5 Acronyms and abbreviations....... 3
1.6 Glossary of symbols.....................4
2 Control theory.................................5
2.1 3-phase PM Synchronous Motor (PMSM).......................................5
2.2 Mathematical description of PM synchronous motors.....................5
2.3 Vector control of PM synchronous motor...8
2.4 PFC average current control theory.......................................15
3 System concept...........................20
3.1 Motor control and interleaved PFC timing... 20
3.2 Key peripherals setting.............. 21
3.3 Application description...............22
4 Hardware......................................26
5 Software design........................... 27
5.1 Fractional numbers representation 27
5.2 Scaling of analog quantities.......27
5.3 Codes architecture.....................31
5.4 State machines.......................... 32
6 Experiment results........................39
6.1 Hardware setup..........................39
6.2 Test results................................ 40
7 Revision history............................49

1 Overview

1.1 Introduction

This document describes the design of a 3-phase PMSM vector control drive without position sensor and 2-phase interleaved PFC on a single controller. The design is targeted for consumer and industrial applications. This cost-effective, high-efficiency, low-noise, variable-power advanced system solution benefits from NXP's MC56F83783 Digital Signal Controller (DSC) device dedicated to motor control and power conversion applications.

1.2 Application features and components

The system is designed to drive a 3-phase PMSM and 2-phase interleaved PFC. The application features of the system are as below:

- Sensorless 3-phase PMSM speed vector (FOC) with 16 kHz control frequency
- Interleaved PFC with 32 kHz control frequency and 96 kHz PWM frequency
- Motor current sensing with three shunts
- PFC current sensing with two shunts cascaded with each MOSFET
- DC bus voltage sensing
- AC Input voltage sensing
- Based on NXP's MC56F83783 digital signal controller
- Running on High-Voltage Development Platform (HVP-MC3PH) and HVP-56F83783 daughter card
- Input voltage 85–265 V, 47–63 Hz
- Remote SCI control through FreeMASTER

Main application components available for users are:

- Software: Written in C-code using some library algorithms from Real Time Control Embedded Software Motor Control and Power Conversion Libraries (RTCESL). The project is developed in CodeWarrior v11.1 based on SDK project template and config tool v9.0
- Hardware: Based on HVP-MC3PH and HVP-56F83783
- Documentation: See References.

1.3 NXP DSC advantages and features

NXP’s MC56F837xx is very suited for digital power conversion and motor control applications. It combines the calculation of DSP and the features of MCU on a single chip. These controllers offer dedicated peripherals as Pulse Width Modulation (PWM) modules, Analog-to-Digital Converters (ADC), timers, Digital-to-Analog Converters (DAC), Event Generator (EVTG), Direct Memory Access (DMA), cross-bar units (XBAR), communication peripherals (SCI, SPI, I^{2}C), and onboard Flash, RAM, and boot ROM.

MC56F83789 contains the following blocks:

- Core operating at a frequency of 100 MHz
- Up to 2 × 128 kB dual partition flash memory with ECC protection and partition swap function.
- Up to 64 kB data/program RAM
- Both on-chip flash memory and RAM can be mapped into both program and data memory spaces
- 32 kB boot ROM supports boot from SCI, I^{2}C, and CAN
- Two high-resolution eFlexPWM modules with up to 2 × 8 PWM outputs of 312 ps resolution
- 2 × 4 16-bit timers
- Two high-speed 12-bit ADCs with 16 external channels
- Four analog comparators with integrated 8-bit DAC references
- Two 12-bit DACs with automatic waveform generation function
- Event Generator (EVTG) and crossbar for very flexible signals routing
- Three SCI modules with LIN slave functionality
- Two SPI modules
- Two I2C/SMBus ports
- One FlexCAN module, with CAN-FD supported
- One USB2.0 controller with integrated PHY

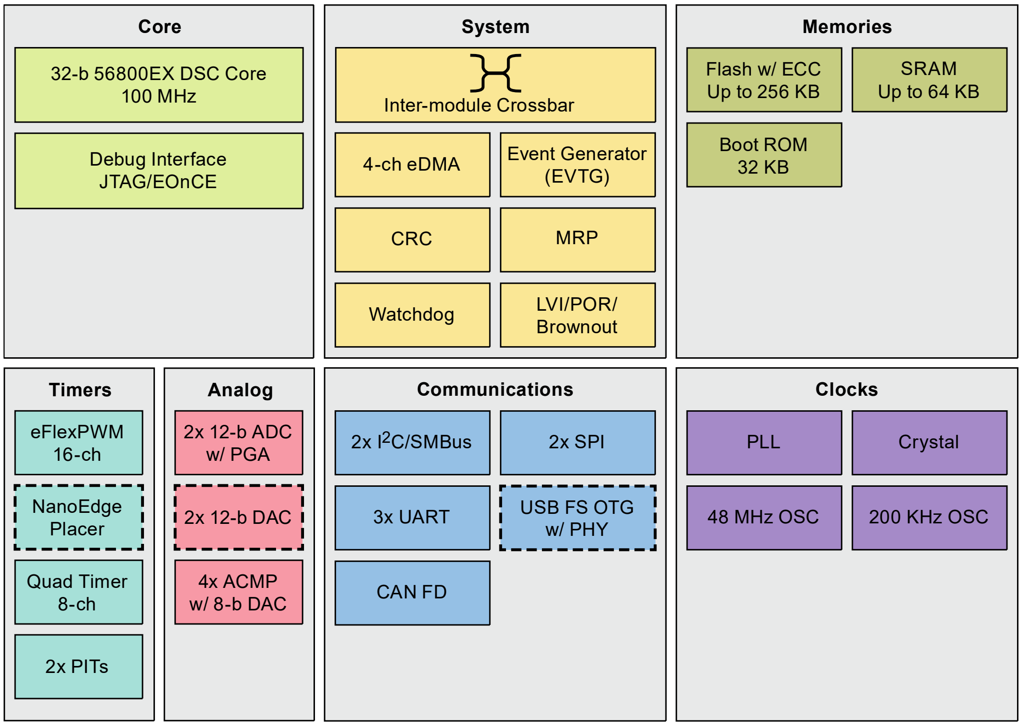


Figure 1. Block diagram

1.4 References

1. *DSP56800E and DSP56800EX Reference Manual* (document DSP56800ERM)
2. *MC56F83xxx Reference Manual* (document MC56F83XXXRM)
3. *Average Current Mode Interleaved PFC Control* (document AN5257)

1.5 Acronyms and abbreviations

Table 1. Acronyms and abbreviations

Term	Meaning
AC	Alternating current
ADC	Analog-to-digital converter
BEMF	Back-electromotive force
CCM	Continuous current mode
DCM	Discontinuous current mode
DAC	Digital-to-analog converter
DC	Direct current

Table continues on the next page...

Table 1. Acronyms and abbreviations (continued)

Term	Meaning
DMA	Direct memory access module
DSC	Digital signal controller
ISR	Interrupt Service Routine
I^2C	Inter-integrated circuit
PFC	Power factor correction
PWM	Pulse-width modulation
RPM	Revolution per minute
SCI	Serial communication interface module: a module that supports asynchronous communication
SPI	Serial peripheral interface module
XBAR	Crossbar unit

1.6 Glossary of symbols

Table 2. Glossary of symbols

Term	Definition
$i_{s\alpha}$, i_{sb}, i_{sc}	Three stator currents in ABC stationary coordinate
$i_{s\alpha}$, $i_{s\beta}$	Two stator currents in Alpha-Beta stationary coordinate
i_d, i_q, u_d, u_q	Currents and voltages in DQ rotating coordinate
ψ_f	PMSM rotor flux
ψ_d, ψ_q	Flux in the air-gap based on DQ rotating coordinate
R_s	Stator resistance
θ	Rotor position – angle between rotor flux vector and A-axis
i_γ, i_δ, u_γ, u_δ	Currents and voltages in γδ rotating coordinate, which is used in BEMF observer
$\theta_{\gamma\delta}$	The estimated rotor position in the BEMF observer
V_{inmax}	AC input voltage amplitude value
V_{in}	AC input voltage instantaneous value
V_o	DC bus voltage
n_p	Pole-pairs of the motor

Table continues on the next page...

Table 2. Glossary of symbols (continued)

Term	Definition
T_e	The electrical torque generated by the motor

2 Control theory

2.1 3-phase PM Synchronous Motor (PMSM)

The PMSM is a rotating electric machine with a classic 3-phase stator, similar to that of an induction motor. The rotor has surface-mounted permanent magnets, as shown in Figure 2.

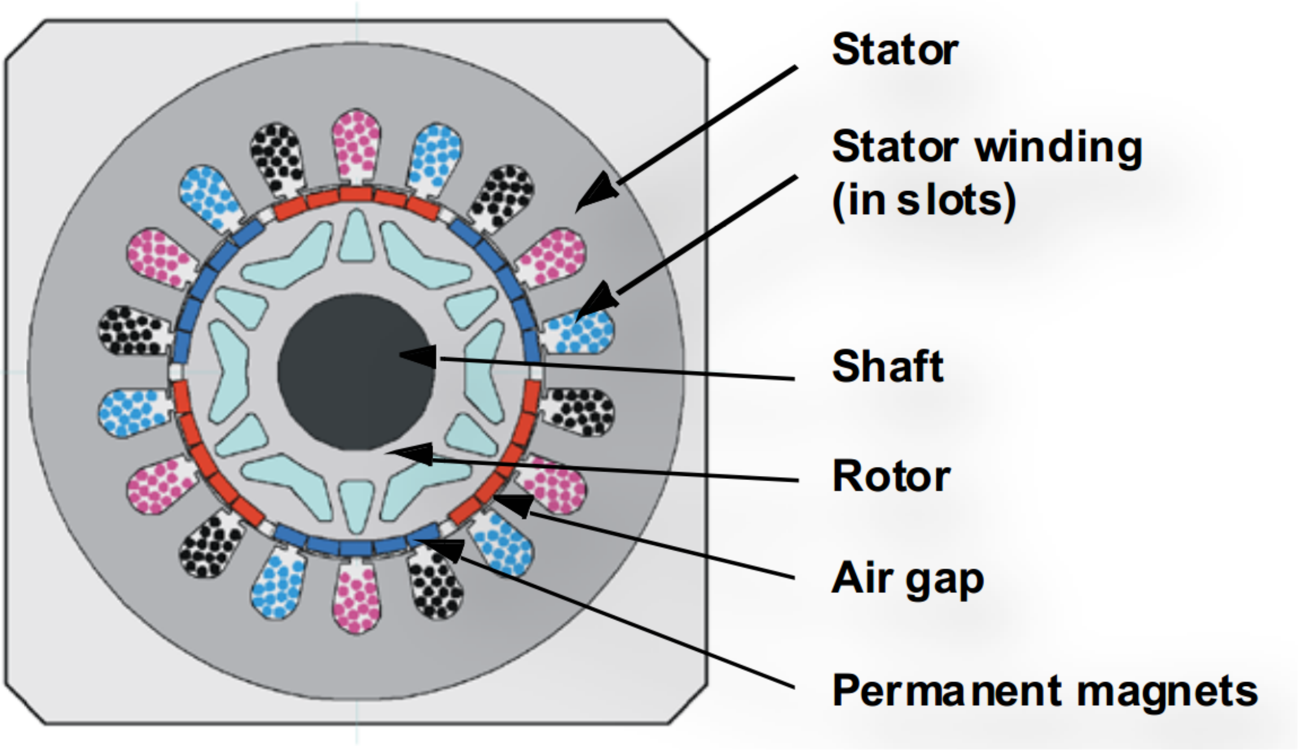


Figure 2. PM synchronous motor cross-section

In this aspect, the PMSM is equivalent to an induction motor, where the air gap magnetic field is produced by a permanent magnet, so the rotor magnetic field is constant. PM synchronous motors offer a number of advantages in designing modern motion control systems. The use of a permanent magnet to generate substantial air gap magnetic flux makes it possible to design highly efficient PM motors.

2.2 Mathematical description of PM synchronous motors

There are a number of PMSM models. The model used for vector control design can be obtained by utilizing space-vector theory. The 3-phase motor quantities (such as voltages, currents, magnetic flux, etc.) are expressed in terms of complex space vectors. Such a model is valid for any instantaneous variation of voltage and current, and adequately describes the performance of the machine under both steady-state and transient operations. Complex space vectors can be described using only two orthogonal axes. Considering the motor as a 2-phase machine motor model reduces the number of equations and simplifies the control design.

2.2.1 Space vector definitions

Assume i_{sa}, i_{sb}, and i_{sc} are the instantaneous balanced 3-phase stator currents:

$$i_{sa} + i_{sb} + i_{sc} = 0$$

Equation 1.

Then the stator current space vector can be defined as in Equation 2.

$$\vec{i}_s = k\left(i_{sa} + a i_{sb} + a^2 i_{sc}\right)$$

Equation 2.

where:

- **a** and **a**^{2} are the spatial operators: $\alpha = e^{j2\pi/3}$, $\alpha^2 = e^{j4\pi/3}$
- **k** is the transformation constant, k=2/3

Figure 3 shows the stator current space vector projection.

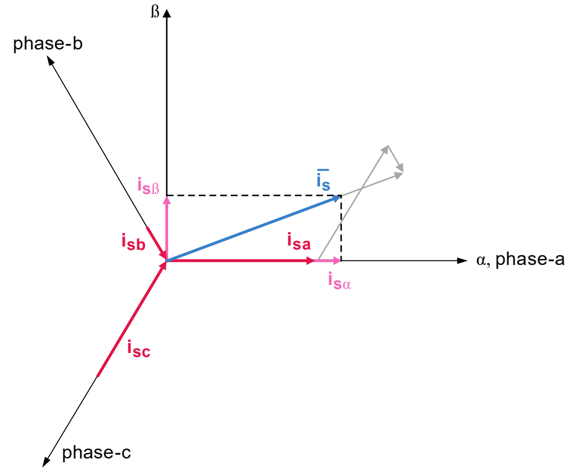


Figure 3. Stator current space vector and its projection

The space vector defined in Equation 2 can be expressed utilizing the two-axis theory. The real part of the space vector equals to the instantaneous value of the direct-axis stator current component, $i_{s\alpha}$, and whose imaginary part is equal to the quadrature-axis stator current component, $i_{s\beta}$. Thus, the stator current space vector in the stationary reference frame attached to the stator can be expressed as:

$$\vec{i}_s = i_{s\alpha} + j i_{s\beta}$$

Equation 3.

In symmetrical 3-phase machines, the direct and quadrature axis stator currents, $i_{s\alpha}$ and $i_{s\beta}$, are fictitious quadrature-phase (2-phase) current components, which are related to the actual 3-phase stator currents as follows.

$$i_{s\alpha} = k\left(i_{sa} - \frac{1}{2} i_{sb} - \frac{1}{2} i_{sc}\right)$$

Equation 4.

$$i_{s\beta} = k\frac{\sqrt{3}}{2}\left(i_{sb} - i_{sc}\right)$$

Equation 5.

where k=2/3 is a transformation constant so that the final equation is:

$$\begin{cases} i_{s\alpha} = i_{sa} \\ i_{s\beta} = \frac{1}{\sqrt{3}}(i_{sb} - i_{sc}) \end{cases} - \mathit{Clarke\ transformation}$$

Equation 6.

NOTE

The vector length is reduced to 2/3 of the original one and it leads to the fact that the amplitude in 2-phase coordinate keeps the same as it is in 3-phase coordinate.

2.2.2 PM synchronous motor model

For the description of a PM synchronous motor, stator winding is sinusoidally distributed.

- When magnets are surface mounted on the rotor, the air-gap length does not change per rotor’s position. There is no saliency, and the 3-phase winding inductances do not change per rotor’s position either.
- When the magnets are inserted into the rotor, the reluctance of 1-phase winding changes per rotor’s position, which means winding inductance is no longer a constant, but changes per rotor’s position. There is saliency in this type of PMSM.

When winding inductance varies with rotor position, voltage equations in stationary 2-phase coordinate is still not decoupled. With the help of Park and Inverse Park transformation, voltage equations can be decoupled in a rotating 2-phase coordinate, which is called **dq** frame. As shown in Figure 4, d-axis is aligned with rotor flux ψ_f. θ is the angle between rotor flux and α-axis of 2-phase stationary frame.

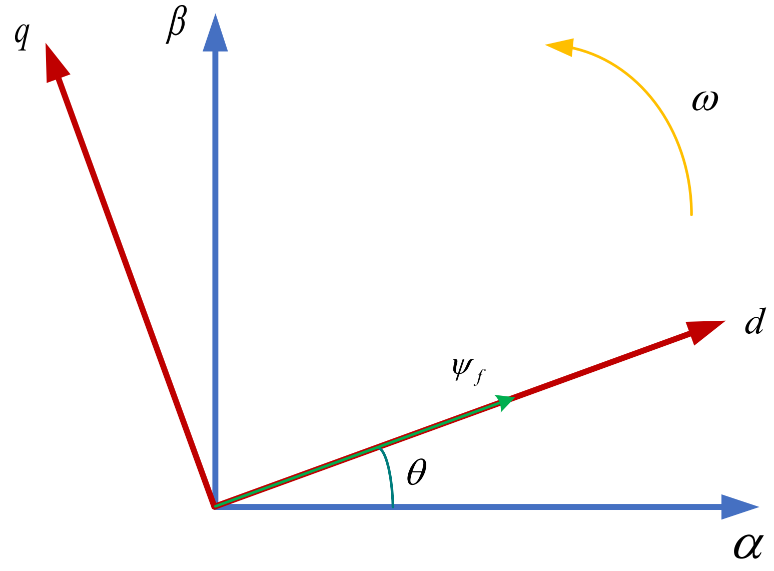


Figure 4. dq rotating frame

In this **dq** frame, flux, stator voltage and torque equations are as follows:

$$\begin{cases} U_d = i_d R_s + \dfrac{d\psi_d}{dt} - \omega\psi_q \\ U_q = i_q R_s + \dfrac{d\psi_q}{dt} - \omega\psi_d \\ \psi_d = L_d i_d + \psi_f \\ \psi_q = L_q i_q \\ T_e = 1.5n_p\left(\psi_d i_q - \psi_q i_d\right) = \underbrace{1.5n_p\psi_f i_q}_{electrical\_torque} + \underbrace{1.5n_p\left(L_d - L_q\right)i_d i_q}_{reluctance\_torque} \end{cases}$$

Equation 7.

Considering that below base speed, $i_d = 0$, there will be no reluctance torque. The total torque can be directly controlled by current i_q only.

2.3 Vector control of PM synchronous motor

2.3.1 Fundamental principle of vector control

High-performance motor control is characterized by smooth rotation over the entire speed range of the motor, full torque control at zero speed, fast accelerations and decelerations. To achieve such control, vector control techniques are used for 3-phase AC motors. The vector control techniques usually refer to Field-Oriented Control (FOC). The basic idea of the FOC algorithm is to decompose a stator current into a magnetic field-generating part and a torque-generating part. Both components can be controlled separately after decomposition. The structure of the motor controller is then as simple as that for a separately excited DC motor.

Figure 5 shows the basic structure of the vector control algorithm for the PM synchronous motor. To perform vector control, follow the steps as below:

1. Measure the motor quantities (phase voltages and currents).
2. Transform them into the 2-phase system (α,β) using Clarke transformation.
3. Calculate the rotor flux space-vector magnitude and position angle.
4. Transform stator currents into the d, q reference frame using Park transformation.
5. The stator current torque (i_{sq}) and flux (i_{sd}) producing components are separately controlled.
6. The output stator voltage space vector is calculated using the decoupling block.
7. The stator voltage space vector is transformed by an inverse Park transformation back from the d, q reference frame into the 2-phase system fixed with the stator.
8. Using space vector modulation, the output 3-phase voltage is generated.

To decompose currents into torque and flux-producing components (i_{sd}, i_{sq}), the position of the motor magnetizing flux must be known. It requires accurate rotor position and velocity information to be sensed. Incremental encoders or resolvers attached to the rotor are naturally used as position transducers for vector control drives. In some applications, the use of speed/position sensors is not desirable either. Then, the aim is not to measure the speed/position directly, but to employ some indirect techniques to estimate the rotor position instead. Algorithms, which do not employ speed sensors, are called sensorless control.

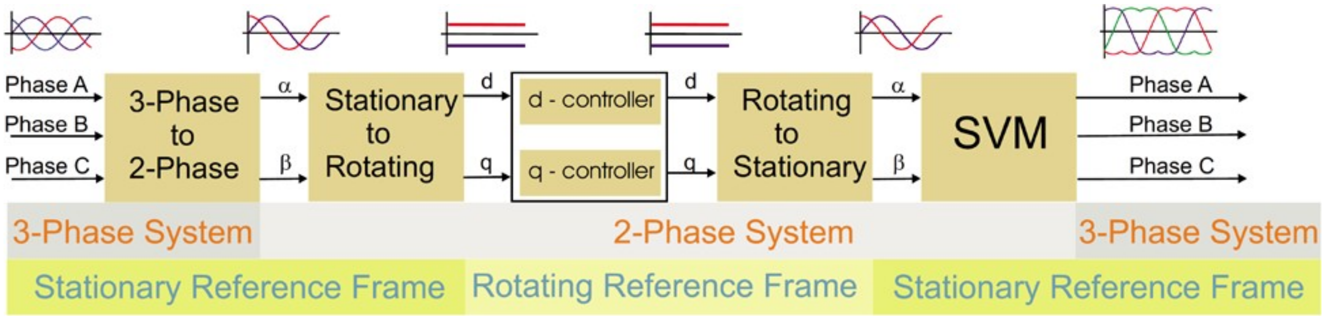


Figure 5. Vector control transformations

2.3.2 Description of the vector control algorithm

Figure 6 shows the overview block diagram of the implemented control algorithm. Similarly, as with other vector control oriented techniques, it is able to control the field and torque of the motor separately. The aim of control is to regulate the motor speed. The speed command value is set by high-level control. The algorithm is executed in two control loops. The fast inner control loop is executed with a 62.5 μs period. The slow outer control loop is executed with a period of 1 millisecond.

For PM synchronous motor control, the algorithm utilizes feedback signals. The essential feedback signals are: 3-phase stator currents and the stator voltage. For the stator voltage, the regulator output is used. For a correct operation, the presented control structure requires a position and speed sensor on the motor shaft, or an advanced algorithm to estimate the position and speed.

The fast control loop executes the following two independent current control loops.

- **Direct-axis current (i_{sd}) PI controllers**: The direct-axis current (i_{sd}) is used to control the rotor magnetizing flux.
- **Quadrature-axis current (i_{sq}) PI controllers**: The quadrature-axis current corresponds to the motor torque.

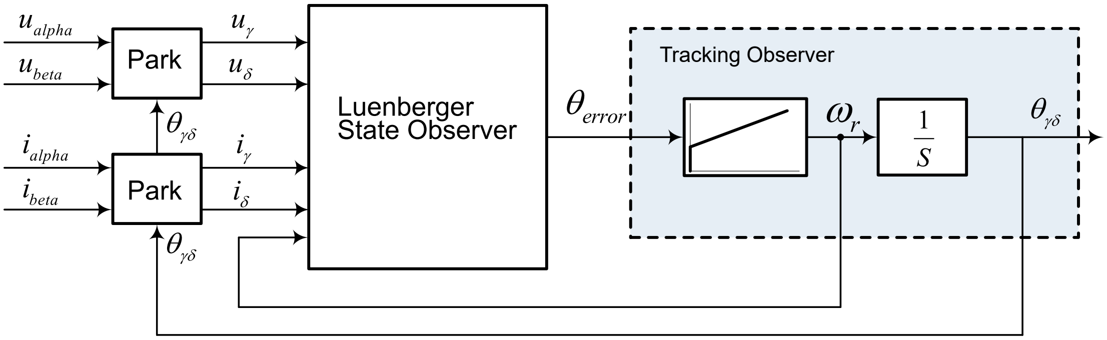


Figure 6. PMSM sensorless field-oriented control overview

Commonly, the current PI controllers' outputs are compensated by DC bus voltage to obtain the desired space-vector for the stator voltage, which is applied to the motor. The fast control loop executes all the following necessary tasks to be able to achieve an independent control of the stator current components.

- 3-phase Current Reconstruction
- Forward Clark Transformation
- Forward and Backward (Inverse) Park Transformations
- DC-Bus Voltage Ripple Elimination
- Space Vector Modulation (SVM)

The slow control loop executes the speed controller and lower priority control tasks. The PI speed controller output sets a reference for the torque-producing quadrature axis component of the stator current (i_{sq}).

2.3.3 Space vector modulation

Space Vector Modulation (SVM) can directly transform the stator voltage vectors from the 2-phase α, β-coordinate system into Pulse Width Modulation (PWM) signals (duty cycle values).

The standard technique of output voltage generation uses an inverse Clarke transformation to obtain 3-phase values. Using the phase voltage values, the duty cycles needed to control the power stage switches are then calculated. Although this technique gives good results, space vector modulation is more straightforward, only valid for transformation from the α, β- coordinate system.

The basic principle of the standard space vector modulation technique can be explained with the help of the power stage schematic diagram depicted in Figure 7. Regarding the 3-phase power stage configuration, as shown in Figure 7, eight possible switching states (vectors) are feasible. They are given by combinations of the corresponding power switches. A graphical representation of all combinations is the hexagon shown in Figure 8. There are six non-zero vectors, U_0, U_{60}, U_{120}, U_{180}, U_{240}, U_{300}, and two zero vectors, O_{000} and O_{111}, defined in α,β coordinates.

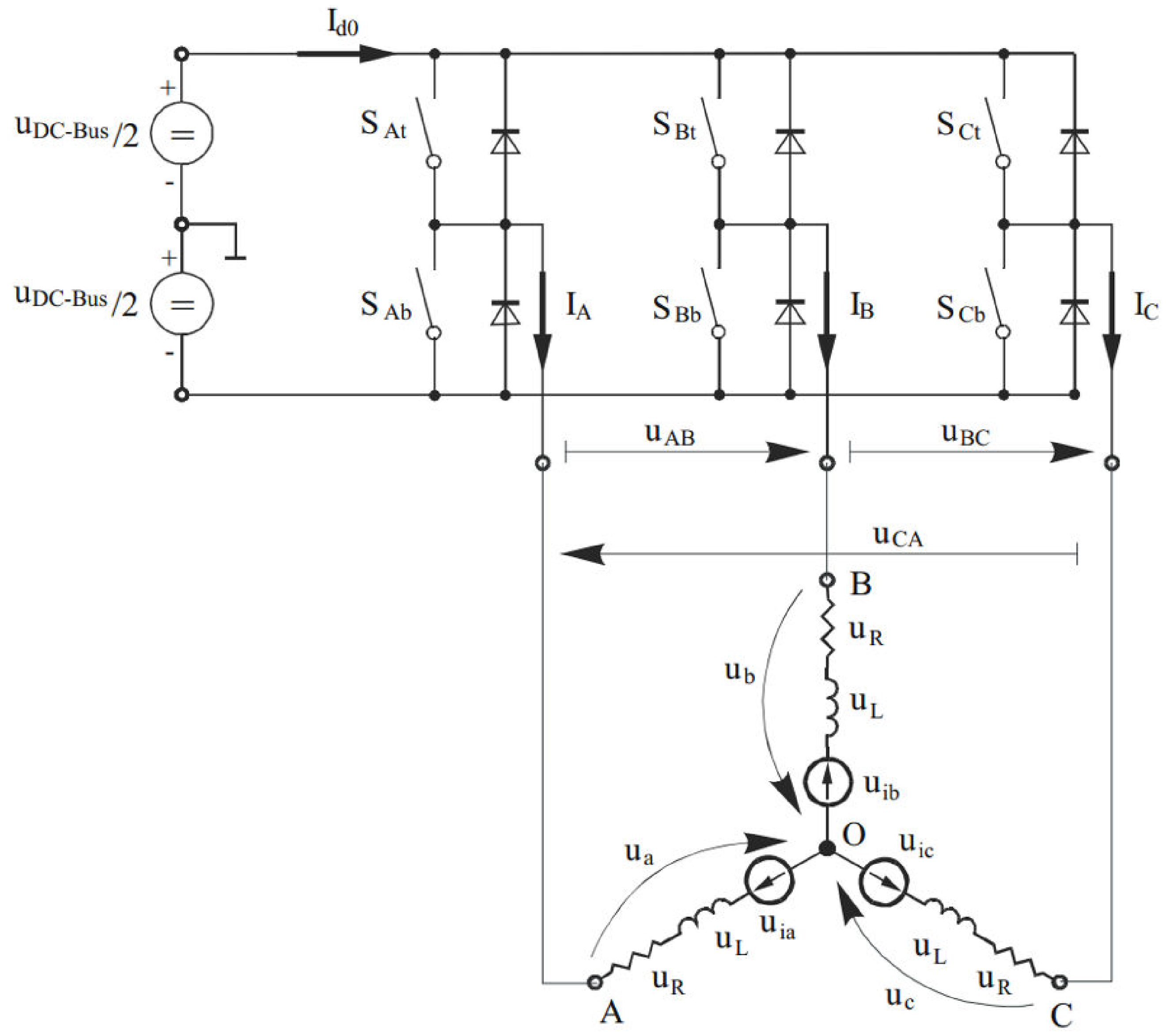


Figure 7. Power stage schematic diagram

The combination of ON/OFF states in the power stage switches for each voltage vector is coded in Figure 8 by the 3-digit number in parentheses. Each digit represents one phase. For each phase, a value of one means that the upper switch is ON and the bottom switch is OFF. A value of zero means that the upper switch is OFF and the bottom switch is ON. These states, together with the resulting instantaneous output line-to-line voltages, phase voltages and voltage vectors, are listed in Table 3.

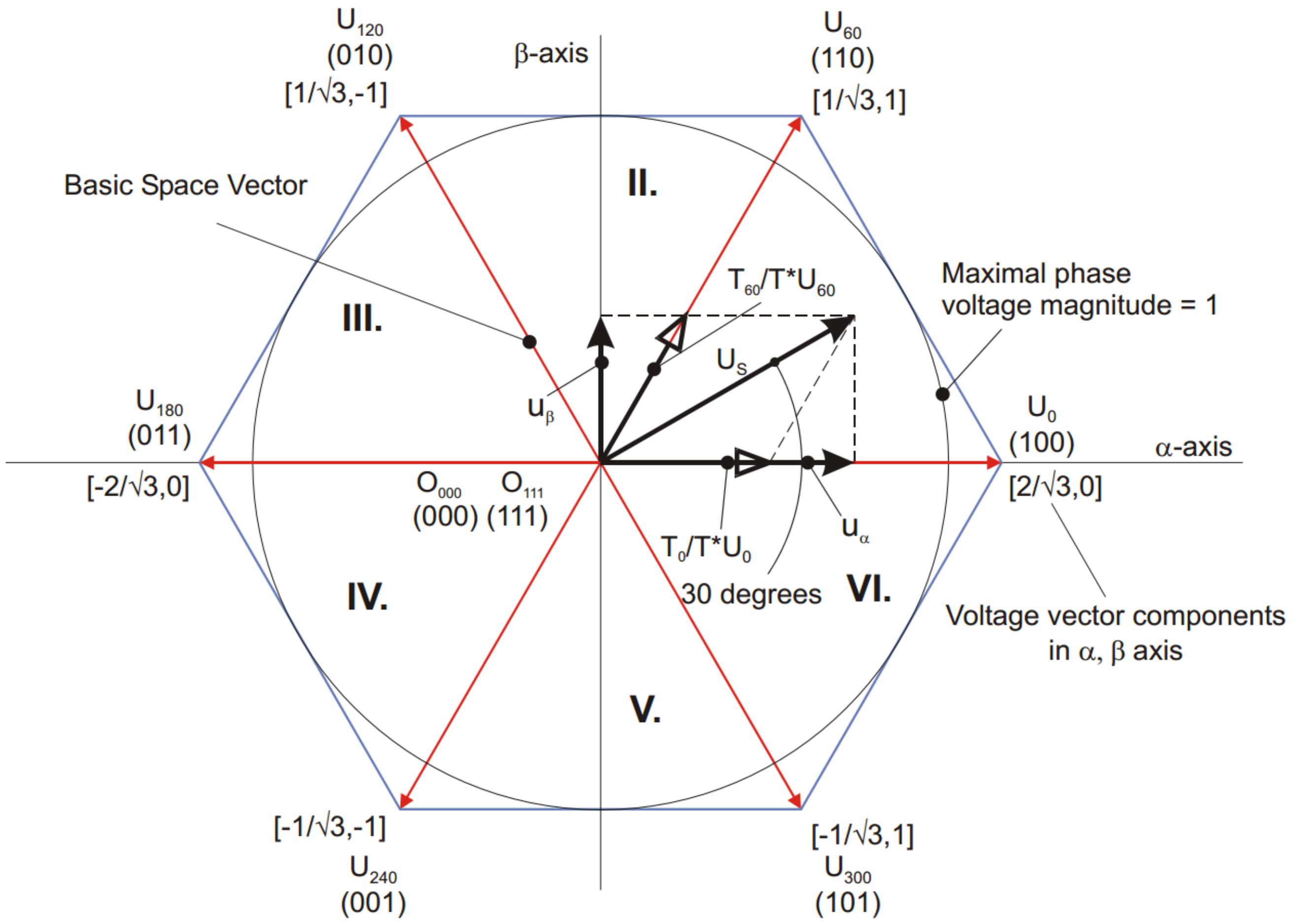


Figure 8. Basic space vectors and voltage vector projection

Table 3. Switching patterns and resulting instantaneous

a	b	c	U_a	U_b	U_c	U_{AB}	U_{BC}	U_{CA}	Vector
0	0	0	0	0	0	0	0	0	O_{000}
1	0	0	$2U_{DC-Bus}/3$	$-U_{DC-Bus}/3$	$-U_{DC-Bus}/3$	U_{DC-Bus}	0	$-U_{DC-Bus}$	U_0
1	1	0	$U_{DC-Bus}/3$	$U_{DC-Bus}/3$	$-2U_{DC-Bus}/3$	0	U_{DC-Bus}	$-U_{DC-Bus}$	U_{60}
0	1	0	$-U_{DC-Bus}/3$	$2U_{DC-Bus}/3$	$-U_{DC-Bus}/3$	$-U_{DC-Bus}$	U_{DC-Bus}	0	U_{120}
0	1	1	$-2U_{DC-Bus}/3$	$U_{DC-Bus}/3$	$U_{DC-Bus}/3$	$-U_{DC-Bus}$	0	U_{DC-Bus}	U_{240}
0	0	1	$-U_{DC-Bus}/3$	$-U_{DC-Bus}/3$	$2U_{DC-Bus}/3$	0	$-U_{DC-Bus}$	U_{DC-Bus}	U_{300}
1	0	1	$U_{DC-Bus}/3$	$-2U_{DC-Bus}/3$	$U_{DC-Bus}/3$	U_{DC-Bus}	$-U_{DC-Bus}$	0	U_{360}
1	1	1	0	0	0	0	0	0	O_{111}

SVM is a technique used as a direct bridge between vector control (voltage space vector) and PWM. The SVM technique consists of the following steps:

1. Sector identification
2. Space voltage vector decomposition into directions of sector base vectors U_x, $U_{x\pm60}$
3. PWM duty cycle calculation

The principle of SVM is the application of the voltage vectors U_{XXX} and O_{XXX} for certain instances in such a way that the **mean vector** of the PWM period T_{PWM} is equal to the desired voltage vector.

This method gives the greatest variability in arranging the zero and non-zero vectors during the PWM period. One can arrange these vectors to lower switching losses and another might want to reach a different result, such as, center-aligned PWM, edge-aligned PWM, minimal switching, etc.

For the chosen SVM, the following rule is defined:

- The desired space voltage vector is created only by applying the sector base vectors: the non-zero vectors on the sector side, (U_x, $U_{x\pm60}$) and the zero vectors (O_{000} or O_{111}).

Equation 8 and Equation 9 define the principle of the SVM:

$$T_{PWM} \cdot U_{S[\alpha,\beta]} = T_1 \cdot U_X + T_2 \cdot U_{X\pm60} + T_0 \cdot (O_{000} \vee O_{111})$$

Equation 8.

$$T_{PWM} = T_1 + T_2 + T_0$$

Equation 9.

To solve the time periods T_0, T_1, and T_2, decompose the space voltage vector $U_{S[\alpha,\beta]}$ into directions of the sector base vectors U_x, $U_{x\pm60}$. Equation 8 splits into Equation 10 and Equation 11.

$$T_{PWM} \cdot U_{SX} = T_1 \cdot U_X$$

Equation 10.

$$T_{PWM} \cdot U_{S(X\pm60)} = T_2 \cdot U_{X\pm60}$$

Equation 11.

By solving this set of equations, obtain the necessary duration for the application of the sector base vectors U_x, $U_{x\pm60}$ during the PWM period, T_{PWM}, to produce the right stator voltages.

$$T_1 = \frac{|U_{SX}|}{|U_X|} T_{PWM} \text{ for vector } U_X$$

Equation 12.

$$T_2 = \frac{|U_{S(X\pm60)}|}{|U_{X\pm60}|} T_{PWM} \text{ for vector } U_{X\pm60}$$

Equation 13.

$$T_0 = T_{PWM} - T_1 - T_2 \quad for\ either\ vector\ O_{000}\ or\ O_{111}\ orthe\ mix\ of\ these\ two\ vectors$$

Equation 14.

2.3.4 Position sensor elimination

The first stage of the proposed overall control structure is alignment algorithm of rotor PM to set an accurate initial position. This allows applying a full startup torque to the motor. In the second stage, to move the motor up to a speed value where the observer provides sufficiently accurate speed and position estimations, the field-oriented control is in open-loop mode (where exact rotor flux position is unknown). As soon as the observer provides appropriate estimates, the rotor speed and position calculation is based on the estimation of a back-EMF in the stationary reference frame using a Luenberger type of observer.

2.3.5 Motor position alignment

In the presented design, there is no rotor position and speed sensor. The exact position of the rotor before the motor is started must be known. One possible and easily implantable method is the rotor alignment to a predefined position. The motor is powered by a selected static voltage vector or a static current vector, and the rotor is aligned to the predefined position, which is usually θ=0 in Figure 4. The alignment is done only once during the first motor start. Figure 9 shows the motor alignment. Before the constant current/voltage vector is applied to the stator, the rotor position is not known. After some stabilization period, the rotor flux must be aligned to the stator flux. In practice, this is true when the external load torque is low enough as compared to the torque produced by the alignment vector.

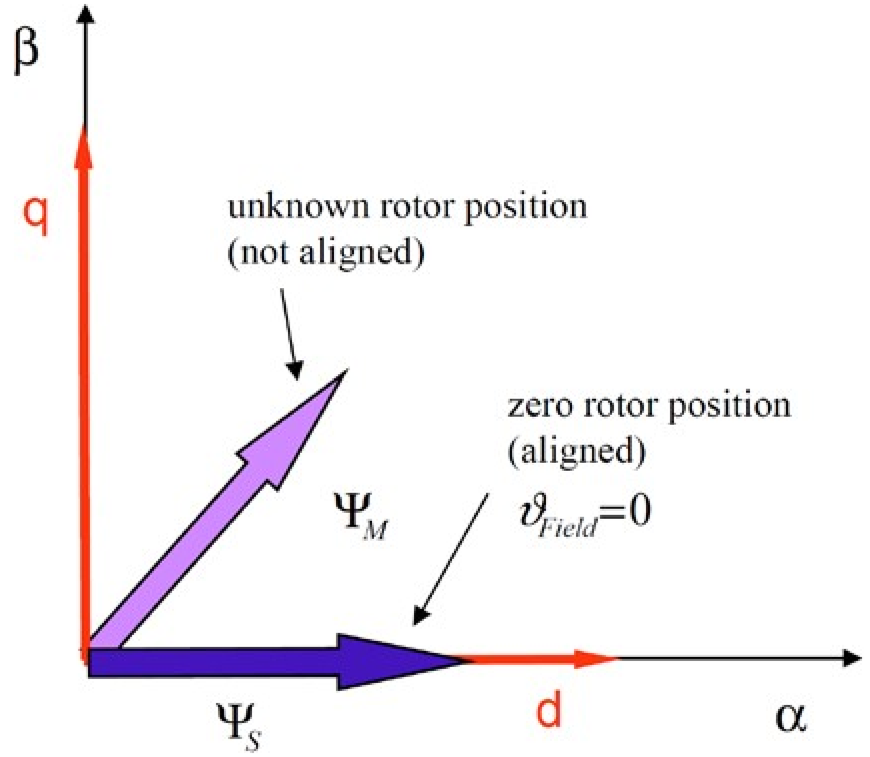


Figure 9. Rotor alignment stabilization: PMSM starting mode

2.3.6 Open-loop startup

After identifying the initial rotor position, the field-oriented control is used in the open-loop mode (only in case of sensorless control). The current set-point is determined by the speed controller, which generates the torque reference current i_{Qref} and the proportional integral controller of speed control loop is initialized to maximum allowable current. The angular speed feedback ω_{FBCK} is kept at zero level during the open loop operation and the vector transformations are fed by a time-varying reference position signal derived by integrating the speed ramp reference. This strategy moves the motor up to the speed values where the observer provides sufficiently accurate speed and position estimates.

The implementation of the open-loop startup is described in Open-loop startup.

2.3.7 Back-EMF observer

When the PMSM reaches a minimum operating speed, a minimum measurable level of back- EMF is generated by the rotor permanent magnets. The back-EMF observer then gradually transits into the closed-loop mode. The feedback loops are then controlled by the estimated angle and estimated speed signals from the back EMF observer.

This estimation method for the position and angular speed is based on the motor mathematical model with an extended electromotive force function. This extended back-EMF model includes both position information from the conventionally defined back-EMF and the stator inductance. It allows to extract the rotor position and velocity information by estimating the extended back-EMF only.

$$\begin{bmatrix} u_\gamma \\ u_\delta \end{bmatrix} = \begin{bmatrix} R_S + sL_D & -\omega_r L_Q \\ \omega_r L_Q & R_S + sL_D \end{bmatrix} \cdot \begin{bmatrix} i_\gamma \\ i_\delta \end{bmatrix} + \left(\Delta L \cdot \left(\omega_r i_D - s i_Q \right) + \psi_f \omega_r \right) \cdot \begin{bmatrix} -sin(\theta_{error}) \\ cos(\theta_{error}) \end{bmatrix}$$

Equation 15.

The observer is applied to PMSM motor with an estimator model excluding the extended back- EMF term. Then the extended back-EMF term can be estimated using the observer shown in Figure 10, which utilizes a simple observer of PMSM stator current. The back-EMF observer presented here is realized within rotating reference frame (dq). The estimator of dq-axis consists of the stator current observer based on RL motor circuit with estimated motor parameters. This current observer is fed by the sum of the actual applied motor voltage, cross-coupled rotational term, which corresponds to the motor q-axis inductance Lq, and compensator corrective output. The observer provides back-EMF signals as disturbance because back-EMF is not included in the observer model.

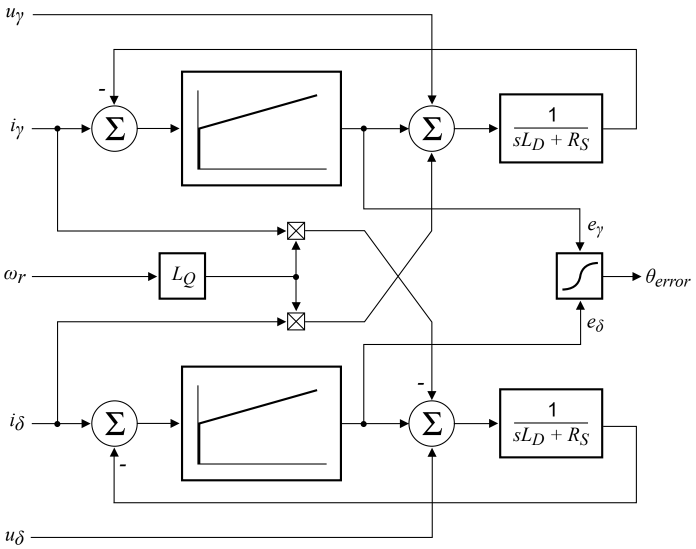


Figure 10. Luenberger type stator current observer acting as state filter for back-EMF

2.3.8 Speed and position extraction

To get the speed and position information from the position error, tracking observer is used. This algorithm adopts the phase-locked loop mechanism. It requires a single input argument as phase error. Such phase tracking observer, with standard PI controller used as the loop compensator, is depicted in Figure 11.

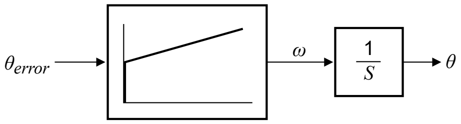


Figure 11. PLL scheme for position estimation

Figure 12 shows the whole picture of Back-EMF observer and tracking observer. $\theta_{\gamma\delta}$ is the estimated rotor position and ω_r is the estimated speed.

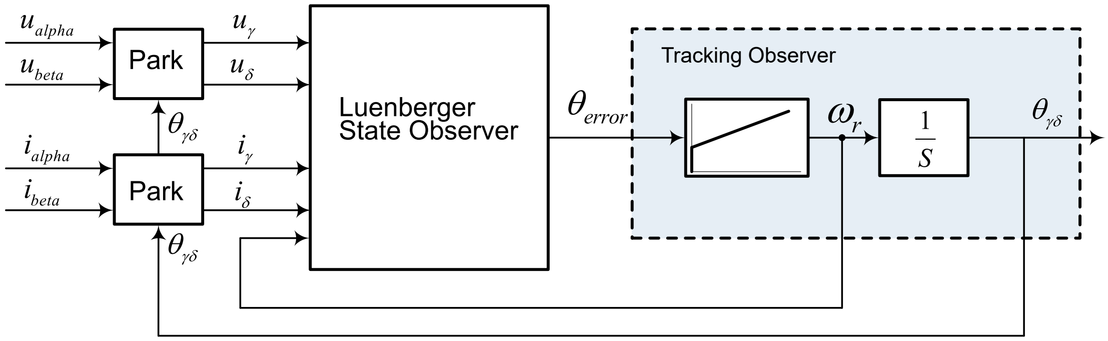


Figure 12. Use observers in αβ frame to get rotor position and speed

2.4 PFC average current control theory

2.4.1 Interleaved boost PFC basis

The interleaved PFC consists of two boost converters connected in parallel. Because the inductor currents are shifted by 180° to one another, as shown in Figure 13, the interleaved operation increases effective current ripple frequency without sacrificing switching loss and also reduces the input current ripple and the output capacitor current ripple. As a result, the overall size of inductor and EMI can be drastically reduced, leading to a higher power density.

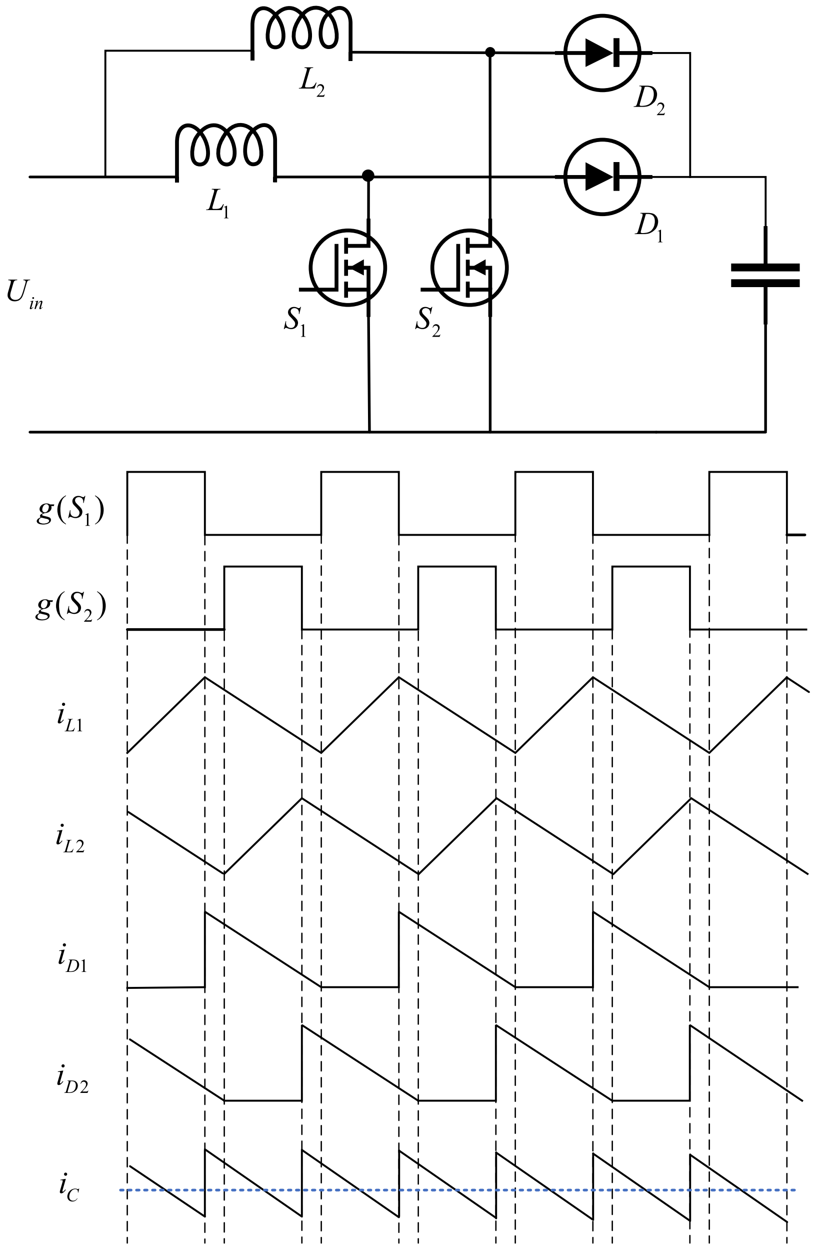


Figure 13. Interleaved Boost PFC operation waveform

However, current balancing between the two interleaved converters is critical, because, due to device and control parameters, the current imbalance can cause more thermal stress on a particular phase and even trigger overcurrent protection. To achieve current balancing, use the same current reference for both phases and control the phases separately. As a result, the closed-loop control will force the current to balance automatically, as shown in Figure 14.

The sensed dc bus voltage, v_o, is compared against the reference bus voltage, v_{ref}. The error signal is input to the voltage controller to regulate the bus voltage at the reference level. The output of voltage controller is proportional to the amount of power transfer by the converter. To eliminate the influence of input voltage change, the square of the AC input peak voltage value is required to compensate the current reference. The rectified AC input voltage also modulates the voltage controller output to ensure the PFC input current has the same phase as that of the AC input voltage. The sensed switch currents, i_{s1} and i_{s2}, are compared against the generated current reference i_{ref} separately. And the error signals are input to two separate current controllers to generate the PWM duty ratio command for each phase.

For the control loops design, see *Average Current Mode Interleaved PFC Control* (document AN5257).

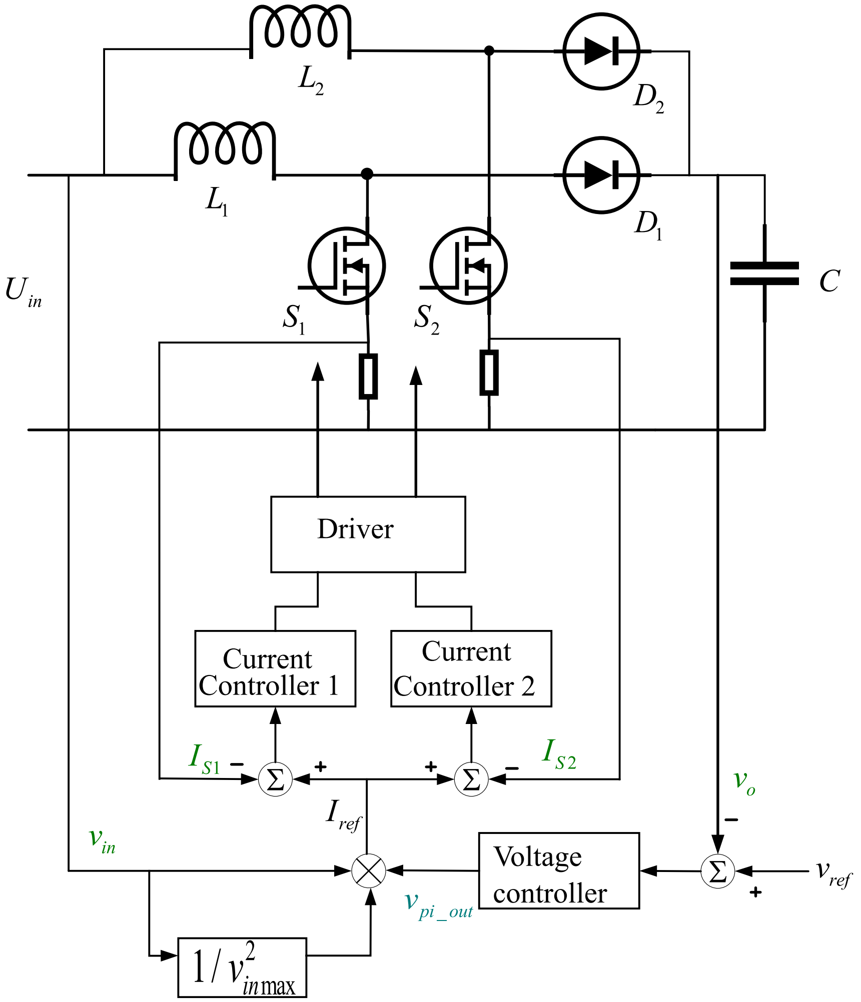


Figure 14. Interleaved Boost PFC control structure

Depending on whether the inductor current is continuous or not, the boost PFC can operate in three modes:

- Continuous Current Mode (CCM)
- Critical Current Mode (CRM)
- Discontinuous Current Mode (DCM)

The converters for higher power range are commonly designed for CCM. And average current control is usually used in CCM boost PFC. However, there is DCM near the zero crossing of input voltage, which will cause input current distortion, especially at light load. There are two main reasons for this distortion: erroneous sampling and different converter dynamics in CCM and DCM.

2.4.2 Feedback current correction due to erroneous sampling

For average current controlled boost PFC, the common choice is to sample the inductor current at the midpoint of its rising edge. When the PFC converter operates in CCM, as shown in Figure 15, the sampled inductor current, i_{l_sample}, equals the average current, i_{l_avg}. So the sampled current can be directly used for average current control.

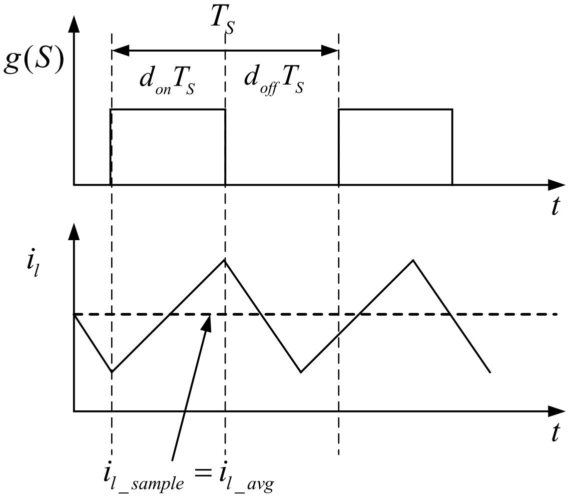


Figure 15. Current waveform for CCM

When the PFC converter operates in DCM, as shown in Figure 16, the sampled inductor current, i_{l_sample}, is larger than the average current, i_{l_avg}. So correction algorithm is needed to get accurate average current.

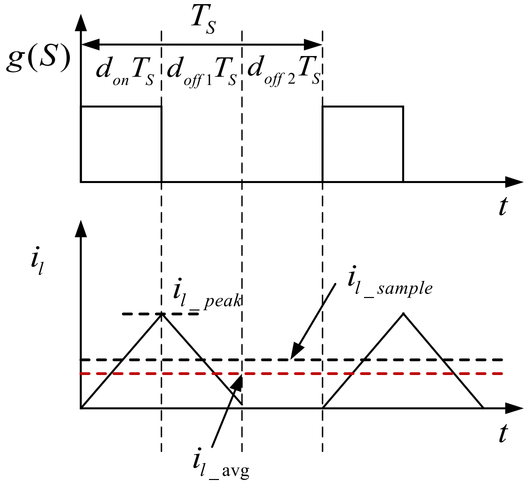


Figure 16. Current waveform for DCM

In a switching cycle, the peak inductor current i_{l_peak} is:

$$i_{l\_peak} = \frac{v_{in} d_{on} T_s}{L}$$

Equation 16.

Where:

- v_{in} is the input voltage.
- d_{on} is the duty cycle.
- T_s is the switching cycle.
- L is the inductor value.

According to the principle of inductor volt-second balance, the inductor current discharging time duty cycle, d_{off1}, is:

$$d_{off1} = \frac{v_{in}d_{on}}{v_o - v_{in}}$$

Equation 17.

Where v_o is the output voltage.

According to Equation 16 and Equation 17, i_{l_avg} can be obtained as:

$$i_{l\_avg} = \frac{1}{2}i_{l\_peak}\left(d_{on} + d_{off1}\right) = \frac{1}{2}i_{l\_peak}\frac{v_o d_{on}}{v_o - v_{in}}$$

Equation 18.

Inductor current is sampled at the midpoint of the rising edge. Equation 19 shows the relation between i_{l_avg} and i_{l_sample}.

$$i_{l\_avg} = i_{l\_sample}\frac{v_o d_{on}}{v_o - v_{in}}$$

Equation 19.

2.4.3 Feedforward duty compensation for different converter dynamics

Since the converter dynamics change abruptly between CCM and DCM, poor inductor current tracking and significant input current distortion appears when same controller is used for both conditions. Without increasing the complexity of the controller design, one way to suppress current distortion is adding the duty-ratio feedforward. In this way, the ideal value of duty-ratio is calculated and added to the output of the current controller to generate the final duty ratio. As a result, the current controller only compensates small values of input current error and a high gain loop is not required. A PI controller designed for converter in CCM can be used for entire operation with low current distortion.

In CCM, the ideal value of duty ratio, d_{CCM}, can be derived as:

$$d_{CCM} = 1 - \frac{v_{in}}{v_o}$$

Equation 20.

In DCM, the input current is directly related to the duty cycle, as calculated in Equation 18, denote d_{DCM} as d_{on}, together with Equation 16. The average current in DCM is:

$$i_{l\_avg} = \frac{1}{2}i_{l\_peak}\frac{v_o d_{DCM}}{v_o - v_{in}} = \frac{1}{2}\frac{v_{in}d_{DCM}T_s}{L}\frac{v_o d_{DCM}}{v_o - v_{in}}$$

Equation 21.

According to the control structure in Figure 14, the required average current, i_{ref}, is:

$$i_{ref} = \frac{v_{pi\_out}v_{in}}{v_{inmax}^2}$$

Equation 22.

Where V_{pi_out} is the output of the voltage controller.

The DCM ideal duty ratio can be calculated as Equation 23, given that the ideal current waveform, $i_{l\_avg} = i_{ref}$.

$$d_{DCM} = \sqrt{\frac{2L}{T_S}\frac{v_{pi\_out}}{v_{inmax}^2}d_{CCM}}$$

Equation 23.

The calculated two duty ratios, d_{CCM} and d_{DCM}, are equal when the converter is expected to work in CRM. When the converter is expected to work in CCM, the calculated d_{DCM} is larger than d_{CCM}. Conversely, the calculated d_{DCM} is smaller than d_{CCM} in DCM. Hence, the required feedforward value of duty ration for operation in mixed mode is a combination of d_{CCM} and d_{DCM}. The correct value is obtained by taking the lowest value of the two duty ratios.

Figure 17 shows the input current controller including feedback current correction and duty-ratio feedforward.

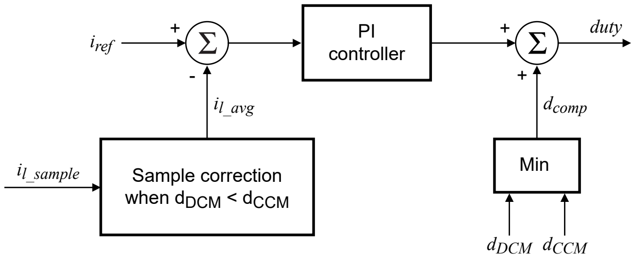


Figure 17. Current controller with sample correction and duty-ratio feedforward

3 System concept

3.1 Motor control and interleaved PFC timing

To use one chip realizing one PMSM FOC and 2-phase interleaved PFC control on 56F83783, the synchronization between motor control PWM waveform and PFC PWM waveform is important. It creates a scanning sequence for ADC that covers all the necessary feedback analog signals for motor and PFC control. Figure 18 shows a detailed system timing with ADC configuration, ISR setting and trigger signals generation.

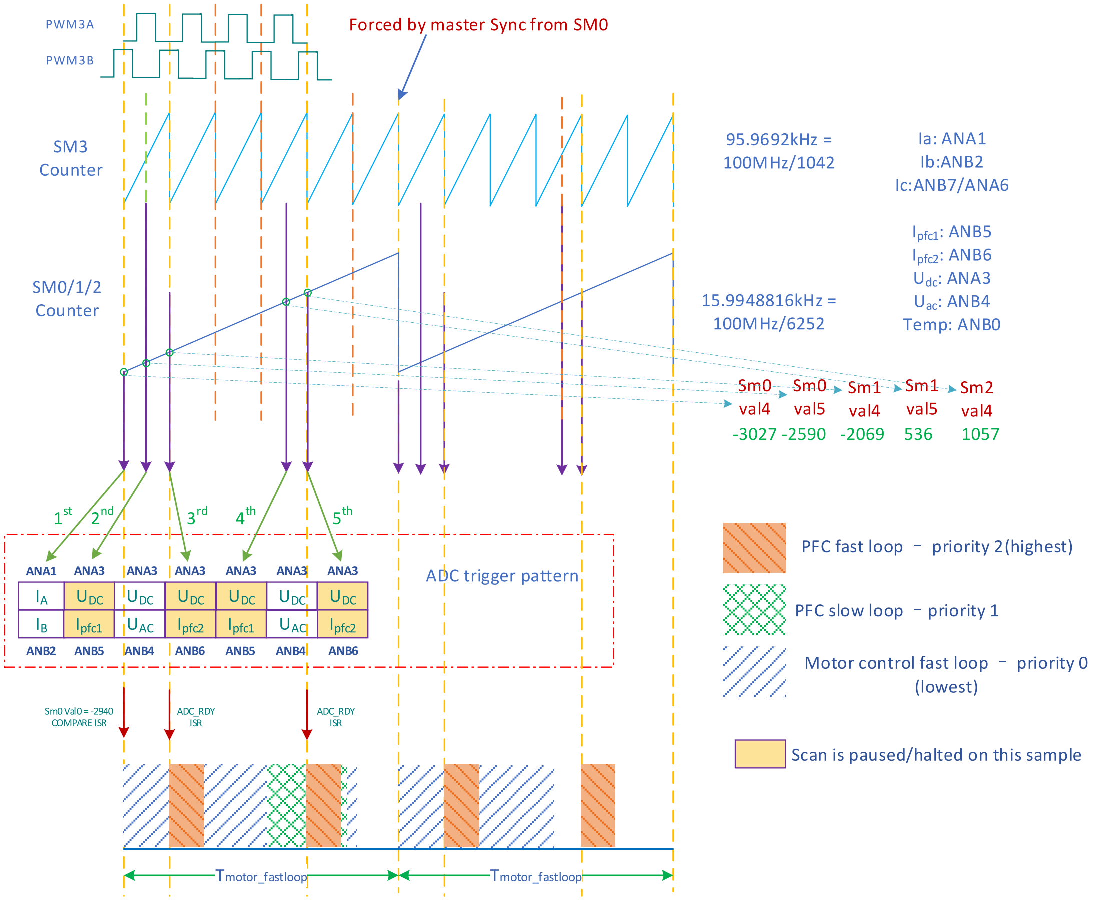


Figure 18. System timing

3.2 Key peripherals setting

Three key peripherals are used to construct one PMSM plus interleaved PFC control. See Figure 18.

- eFlexPWMA: SubModule (SM) 0-2 are for PMSM control, and the two outputs of SM3 are for PFC control. The clock of eFlexPWMA is 100 MHz, which is the same as bus clock. **SM0-2** work at 15.99 kHz while **SM3** works at 95.9692 kHz, which is exactly six times of the frequency of **SM0-2**. **SM3** is synchronized by master sync signal from **SM0** which is the compare match signal of VAL1 in **SM0**. For two-phase interleaved PFC, the outputs of **SM3** have a phase shift of 180°.
- ADC: ADC is configured as simultaneous parallel mode. It is essential to create an **ADC trigger pattern**, as shown in Figure 18, which is actually a scan sequence repeatedly triggered by eFlexPWM and converted by ADC. There are five trigger signals within one SM0 PWM period, generated by VALx compare match signals.
  - — The first trigger signal is at the beginning of SM0 PWM period, which is exactly at the middle of NULL(000) vector for PMSM control. It starts the conversion of the sequence, so the two phase currents IA and IB are sampled/converted simultaneously by ADC. ADC should have continued the conversion of next pair (UDC and Ipfc1) right after the conversion of IA and IB. ADC is configured to be paused on next pair, so it halts and waits for the second trigger signal. Meanwhile a compare match ISR is generated by SM0 VAL0 right after the conversion of IA and IB, where FOC fast loop is implemented.
  - — The second trigger signal is located at the middle of PWM-ON of PFC phase 1 circuit, which is controlled by PWM3A. Upon the occurrence of this trigger signal, DC bus voltage UDC and MOS current Ipfc1 are sampled/converted. There is no pause on next pair, so UDC and input AC voltage UAC are sampled/converted right next to the completion of UDC and Ipfc1. At this point, ADC should have continued the conversion of next pair (UDC and

Ipfc2) right after the conversion of UDC and UAC. ADC is configured to be paused on next pair, so it halts and waits for the third trigger signal.

- — The third trigger signal is located at the middle of PWM-ON of PFC phase 2 circuit, which is controlled by PWM3B. Upon the occurrence of this trigger signal, DC bus voltage UDC and MOS current Ipfc2 are sampled/converted. Meanwhile, an ADC ready ISR is generated after the completion of UDC and Ipfc2, in which PFC fast loops of phase 1 and 2 are implemented. Because there is a pause on the next pair, ADC will wait for the 4th trigger signal.
- — To make sure the control frequency of PFC fast loop is 32 kHz and PMSM fast loop is 16 kHz, the 4th and 5th trigger signals are located in places, as shown in Figure 18. MOS currents of phase 1 and 2 are obtained exactly like what's been done on the second and third trigger signals. ADC ready ISR is generated after the completion of the final pair in the sequence.
- — This sequence (ADC trigger pattern) is executed in every SM0 PWM period. By setting SM0 VAL0 compare ISR the lowest interrupt priority where FOC is implemented, and ADC ready ISR the highest priority where PFC current loop is implemented, CPU loading is allocated reasonably in this fashion. The five trigger signals are OR'ed together through AOI in EVTG.

- PIT0: PIT0 is used to generate a 10 kHz periodic interrupt for PFC voltage loop. The ISR has a priority between FOC fast loop and PFC fast loop.

3.3 Application description

3.3.1 Motor control process

For PMSM sensorless vector control in this application, it needs three major stages: Alignment, open-loop startup, merge and closed-loop spin.

3.3.1.1 Alignment

To start the motor, align the rotor to a known place to make sure there will be a maximum startup torque later. D-axis current reference, i_{dref}, is 0, and q-axis current reference, i_{qref}, is 0.5 A. The rotor position is -90°. There will be a current vector locating at A-axis and the lasting time is 0.8 second. Figure 19 shows the control diagram.

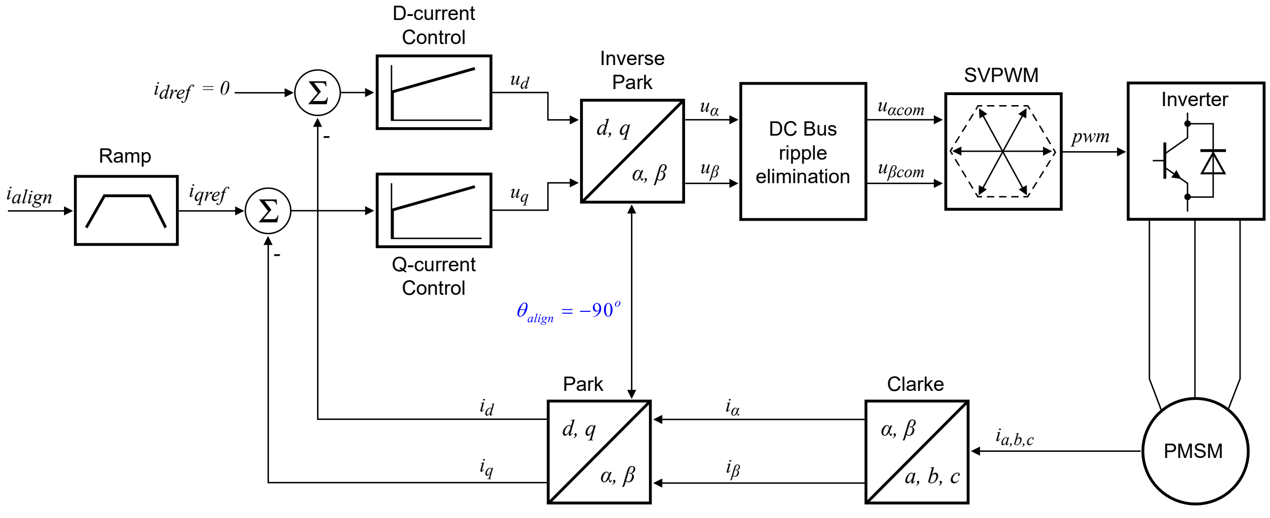


Figure 19. Alignment stage

3.3.1.2 Open-loop startup

After alignment, the rotor has been placed on position 0, which is aligned with A-axis. A current vector is constructed at the position 0 and starts to rotate with a ramping up speed. In this process, the electrical torque increase as the angle between current vector and the rotor flux increases. There will be no loss of synchronization when the angle between current vector and rotor flux stays below 90°. It is obvious that it needs a large current to pull up a heavy load in this fashion. In this application, because there the loading on the shaft is light, i_{qref} is 0.5 A and i_{dref} is 0. In this stage, d-axis position is generated by the integral of the ramping speed. Speed stops increasing when it reaches 500 RPM.

The observer starts to work when current vector speed is higher than 250 RPM.

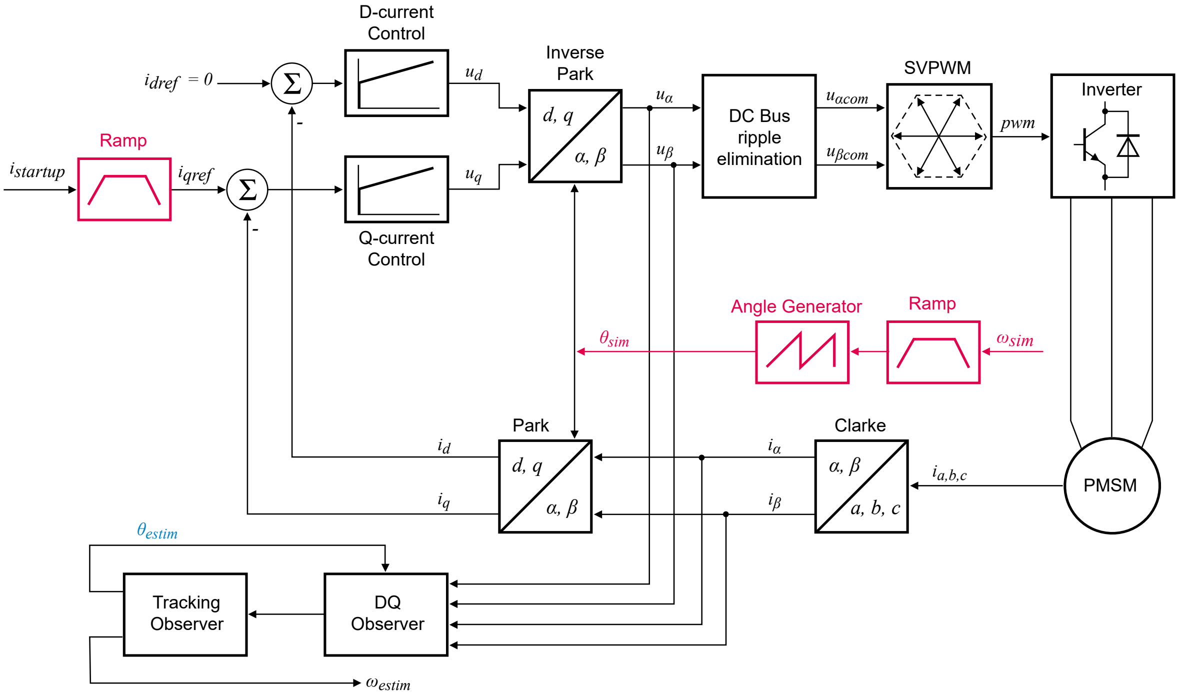


Figure 20. Open-loop startup stage

3.3.1.3 Merge

At the end of open-loop startup stage, as soon as current vector speed reaches 500 RPM, it is assumed that the rotor has reached 500 RPM and the observer is working properly. It is time to switch from manually given rotor position, θ_{sim}, to the observed rotor position, θ_{estim}, as shown in Figure 21.

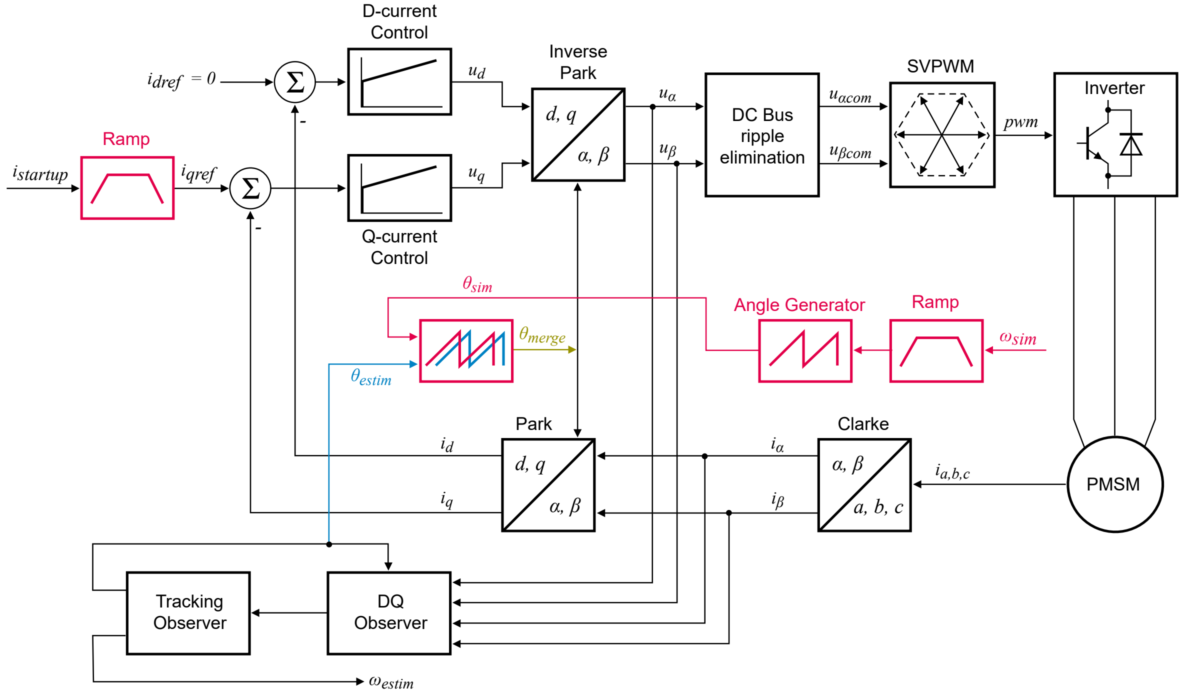


Figure 21. Merge stage

There must be a gap between θ_{sim} and θ_{estim} the moment current vector speed reaches 500 RPM. It is not recommended to switch from θ_{sim} to θ_{estim} abruptly, because it can cause undesired current spikes. A new position, θ_{merge}, is created to make the transition smoothly.

$$\theta_{merge} = \theta_{sim} + Coeff \cdot (\theta_{estim} - \theta_{sim})$$

`Coeff` is a variable that increase from **0** to **1** within 100 fast loops. It will take 6.2 ms to move from manually given position to the observed position.

3.3.1.4 Closed-loop spin

In the end of merge stage, when Coeff reaches **1**, the observed position is completed used for FOC and speed loop control is enabled. The actual q-axis current is set to the integral part of speed controller to make sure there is no step change on the q-axis current reference. Now the speed loop and current loops are fully engaged, and the rotor position and speed feedbacks come from the observer.

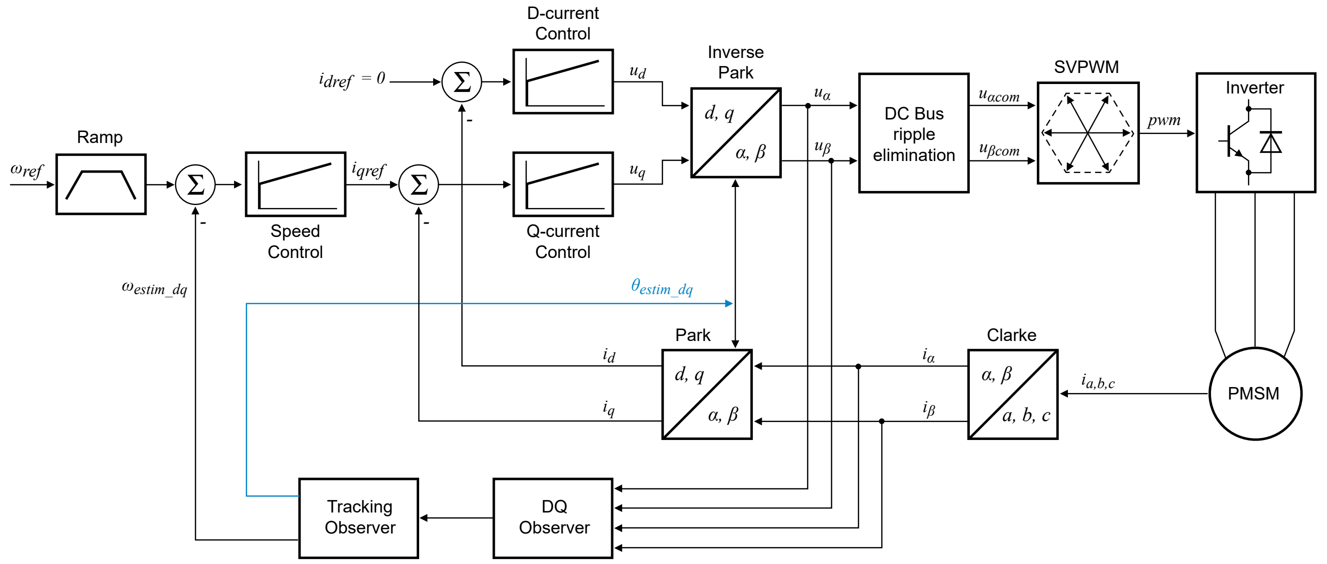


Figure 22. Closed-loop spin stage

3.3.2 PFC control process

Two control loops are used for PFC control:

- The outer loop controls DC bus voltage.
- The inner loop controls inductor current, which makes the current sinusoidal and maintains the same phase as the input voltage.

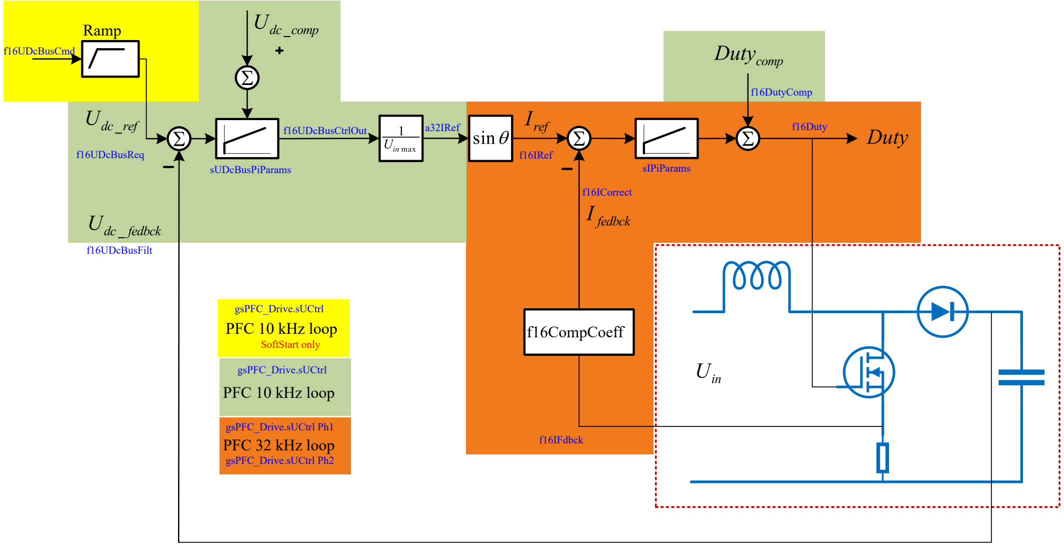


Figure 23. Control diagram of PFC

3.3.3 AC input voltage peak value and phase detection

It requires the latest AC input peak voltage value and phase information to compensate the controller and get the current reference.

- AC input voltage peak value is detected in `PIT0_ISRHANDLER`, which is updated in 10 kHz frequency. Real DC bus voltage is also updated in this ISR. As shown in Figure 20, when voltage-rising trend is detected, the program starts to detect peak value, and the maximum value detected is deemed as peak value when voltage is falling. PFC cannot be enabled until eight AC input voltage peaks are detected.
- AC input voltage phase (zero-crossing) is detected in `ADC_A_IRQHANDLER`, which is calculated in 32 kHz frequency. During AC input peak voltage detection, the program starts to detect rising threshold after voltag-rising trend is confirmed, and the program starts to detect falling threshold after voltage-falling trend is confirmed. Rising and falling thresholds have the same value. The time interval between adjacent detected falling thresholds is half the AC input period. The time point when rising threshold is detected is used to calculate the AC input zero-crossing point. Sinusoidal reference can be calculated with the detected zero-crossing point and AC input period.

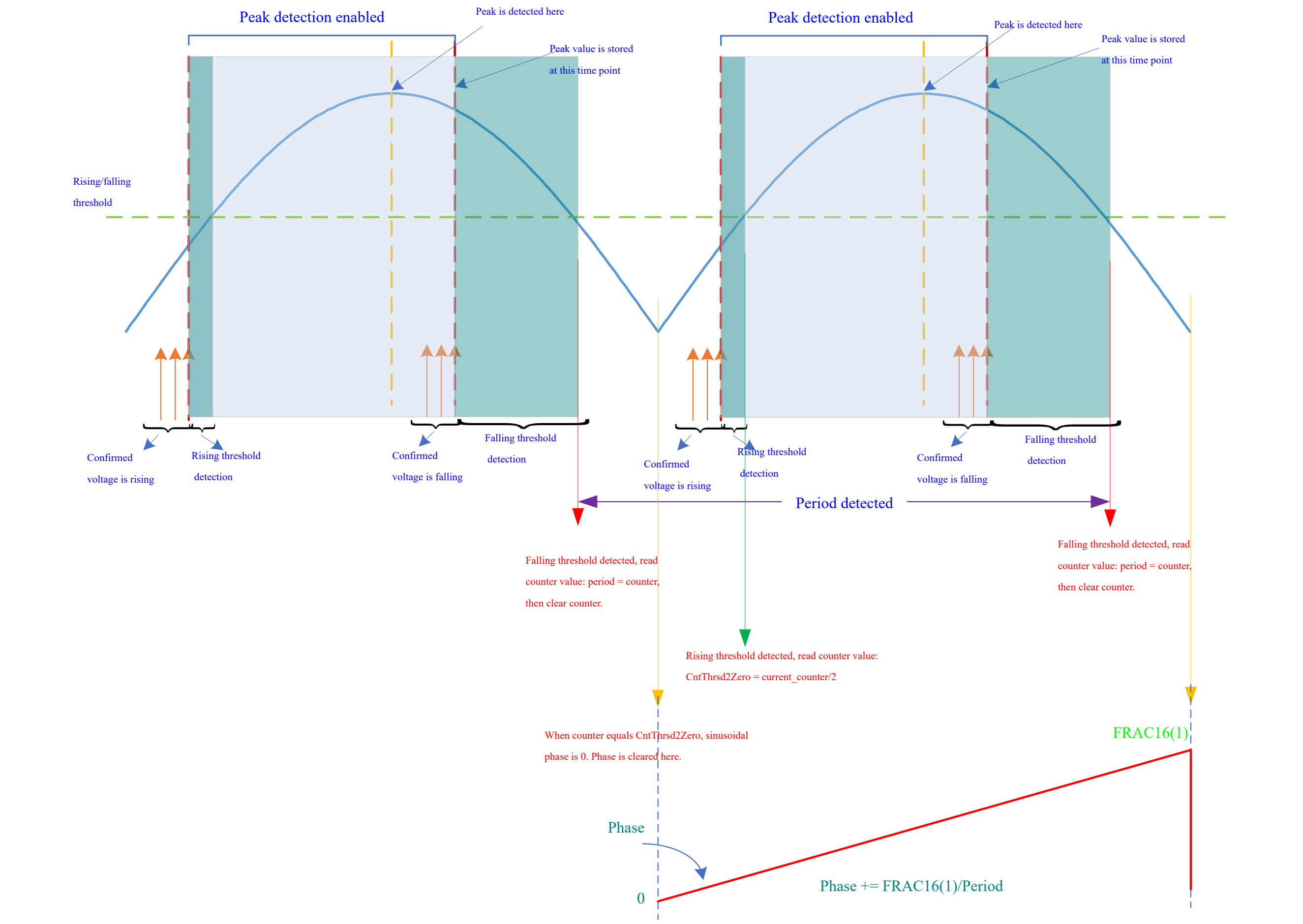

Peak detection enabled
Peak is detected here
Peak value is stored at this time point
Peak detection enabled
Peak is detected here
Peak value is stored at this time point
Rising/falling threshold
Confirmed voltage is rising
Rising threshold detection
Confirmed voltage is falling
Falling threshold detection
Confirmed voltage is rising
Rising threshold detection
Confirmed voltage is falling
Falling threshold detection
Period detected
Falling threshold detected, read counter value: period = counter, then clear counter.
Falling threshold detected, read counter value: period = counter, then clear counter.
Rising threshold detected, read counter value: CntThrsd2Zero = current_counter/2
When counter equals CntThrsd2Zero, sinusoidal phase is 0. Phase is cleared here.
FRAC16(1)
Phase
0
Phase += FRAC16(1)/Period

Figure 24. AC input voltage peak value and phase detection

4 Hardware

This application is realized on High-Voltage Development Platform (HVP-MC3PH) and HVP-56F83783 (a daughter card inserted into HVP-MC3PH). For detailed hardware information regarding this platform, see High-Voltage Development Platform.

The maximum designed loading is 800 W. If wanted, users can directly add loading on the DC bus through J12 on HVP-MC3PH, which is actually a braking resistor connector for the motor.

5 Software design

The whole project is developed within CodeWarrior v11.1, which can be found on CodeWarrior® for MCUs. MCUXpresso config tool v9 is also used to do all the peripheral initializations, including pins and clocks. Config tool provides GUI interface to easily configure pins, clock, peripherals, and middlewares such as FreeMASTER. This application project is developed through below steps:

1. Download project templates package from Project template MC56F83xxx.
2. Unzip the package.
3. Open CodeWarrior, and import the `project_template_MC56F83783` project into workspace. Check **Copy projects into workspace** in the **Import Projects** dialogue.
4. There is a *.mex* file within the imported project, which can be opened directly by the configuration tool.

5.1 Fractional numbers representation

This application uses a fractional representation for most of the quantities. The fractional arithmetic is supported by the NXP DSCs. The application and the algorithms library benefits from this fact.

The N-bit signed fractional format is represented using the `1.[N-1]` format (1 sign bit, N-1 fractional bits). Signed Fractional (SF) numbers lie in the following range:

$-1.0 \leq SF \leq +1.0 - 2^{-[N-1]}$

For words and long-word signed fractions, the most negative number that can be represented is – 1.0, whose internal representation is `0x8000` and `0x80000000` respectively. The most positive word is `0x7FFF` or $1.0 - 2^{-15}$ and the most positive long-word is `0x7FFFFFFF` or $1.0 - 2^{-31}$.

In short, the fractional value that a signed N bit integer represents for `1.[N-1]` format is:

$$\frac{integer}{2^{N-1}}$$

5.2 Scaling of analog quantities

The following shows the relationship between a real and a fractional representation:

$$\text{Fractional value} = \frac{\text{Real Value}}{\text{Real Qauntity Range}}$$

Equation 24.

Where:

- Fractional Value = Fractional representation of quantities [–]
- Real Value = Real quantity in physical units [..]
- Real Quantity Range = Maximum defined quantity value used for scaling in physical units [..]

5.2.1 Scaling for motor control

How physical quantities are scaled in PMSM vector control are provided in the following subsections.

5.2.1.1 Current scale

The current is generally measured as voltage drop on the shunt resistors which is amplified by an operational amplifier, as shown in Figure 25. If both positive and negative currents are needed, the amplified signal has an offset. The offset is generally the half of the ADC range. The maximum current scale is proportional to the half of maximum ADC input voltage range, as shown in Figure 26.

In the circuits shown in Figure 25, one leg of the inverter is demonstrated. Shunt resistor is 0.05 Ω, amplifier gain is 4.121, and the offset added to the output is 1.65 V. The current range -8 - 8A corresponds to 0 - 3.3 V at the input of ADC channel, which in turn points out that current range -8 - 8A corresponds to fractional value -1 - 1 in the code, as shown in Figure 26. So current scale is 8A in this case.

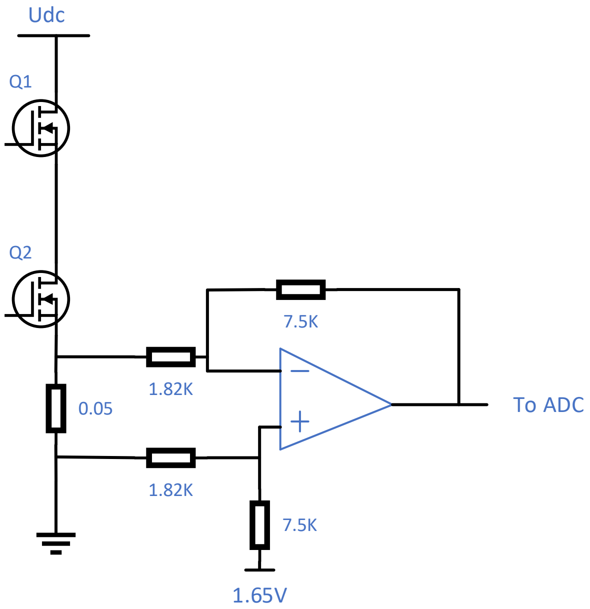


Figure 25. Phase current sampling circuit

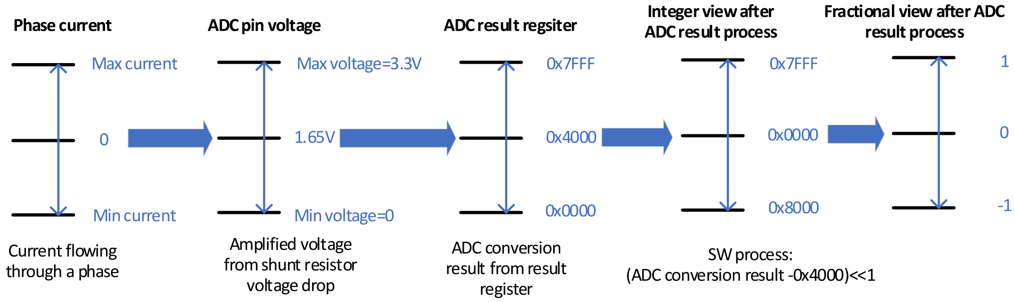

Phase current
ADC pin voltage
ADC result regsiter
Integer view after ADC result process
Fractional view after ADC result process
Max current
0
Min current
Max voltage=3.3V
1.65V
Min voltage=0
0x7FFF
0x4000
0x0000
0x7FFF
0x0000
0x8000
1
0
-1
Current flowing through a phase
Amplified voltage from shunt resistor voltage drop
ADC conversion result from result register
SW process:
(ADC conversion result -0x4000)<<1

Figure 26. Phase current scaling

5.2.1.2 DC bus voltage scale

The voltage is generally measured on the voltage resistor divider by the ADC, so the maximum voltage scale is proportional to the maximum ADC input voltage range. In this application, the maximum voltage is 433 V. See Figure 27 for details.

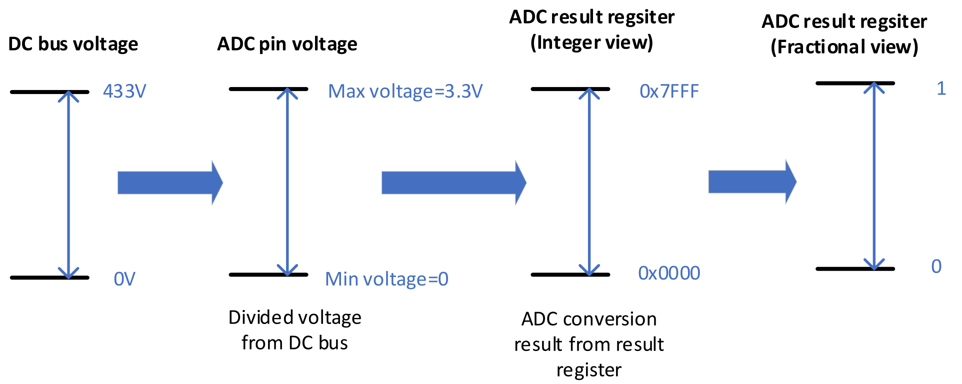

DC bus voltage
433V
0V
ADC pin voltage
Max voltage=3.3V
Min voltage=0
Divided voltage from DC bus
ADC result regsiter (Integer view)
0x7FFF
0x0000
ADC conversion result from result register
ADC result regsiter (Fractional view)
1
0

Figure 27. DC bus voltage scaling

5.2.1.3 Phase voltage scale

For phase current scaling, the maximum physical measurable phase current is chosen as scale. In stable state, phase current is sinusoidal, so the maximum amplitude of phase current is current scale. Because both current and voltage are both fed into an observer to get rotor position and speed, to ensure the accuracy of the observer, the phase voltage should also use the maximum amplitude as scale.

As described in Space vector modulation, the length of six basic voltage vectors is DC bus voltage, so the maximum length of a basic voltage vector is actually DC bus voltage scale, which is 433 V in this application. Because only the inscribed circle is used as shown in Figure 8, the maximum available length of a voltage vector is: $\frac{\sqrt{3}}{2} \cdot \text{DC Bus voltage scale}$

In the Clarke transformation mentioned in Equation 6, the vector length is reduced to its 2/3 to make sure that the vector length is same as the amplitude in stable state. So the phase voltage scale will be: $\frac{2}{3} \cdot \frac{\sqrt{3}}{2} \cdot \text{DC Bus voltage scale} = \frac{1}{\sqrt{3}} \cdot \text{DC Bus voltage scale}$

5.2.1.4 Input voltage scale of SVM

During the implementation of space vector modulation in this application, the inputs are alpha and beta voltages scaled based on the dynamic inscribed circle radius as shown in Figure 8. The radius is treated as unit 1. However, the radius is: $\frac{\sqrt{3}}{2} \cdot \text{real DC Bus voltage}$

so the input voltage scale of SVM is different from the phase voltage scale mentioned in Phase voltage scale.

An algorithm block called **DC bus ripple elimination** is implemented to perform a transformation between different scales, as shown in Figure 28.

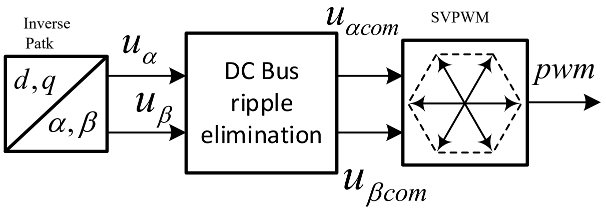


Figure 28. DC bus ripple elimination

- u_α and u_β denote fractional phase voltages, which use $\frac{1}{\sqrt{3}}\text{DC\_Bus\_voltage\_scale}$ as scale.
- $u_{\alpha com}$ and $u_{\beta com}$ denote fractional input voltages to SVM, which use $\frac{\sqrt{3}}{2} \cdot \text{real\_DC\_Bus\_voltage}$ as scale.
- u_{dc} denotes fractional real DC bus voltage, which uses `DC_Bus_voltage_scale` as scale.

To work out the relationship between u_α and $u_{\alpha com}$, follow the rule that the input vector length to SVM should stay the same as the output vector length of SVM. Equation 25 is listed based on this rule.

$$u_\alpha \cdot \frac{1}{\sqrt{3}} DC\_Bus\_voltage\_scale \cdot \frac{3}{2} = u_{\alpha com} \cdot \frac{\sqrt{3}}{2} \mathrm{real\_DC\_Bus\_voltage}$$

Equation 25.

There is a 3/2 at the left side of Equation 25, which is the input of SVM, because vector length is reduced to 2/3 in all the FOC calculations. We need to restore it to its original length in SVM. From Equation 25, we can get:

$$u_{\alpha com} = \frac{u_\alpha \cdot DC\_Bus\_voltage\_scale}{\mathrm{real\_DC\_Bus\_voltage}} = \frac{u_\alpha \cdot DC\_Bus\_voltage\_scale}{u_{dc} \cdot DC\_Bus\_voltage\_scale} = \frac{u_\alpha}{u_{dc}}$$

Equation 26.

Same operation for $u_{\beta com}$:

$$\begin{cases} u_{\alpha com} = \dfrac{u_\alpha}{u_{dc}} \\ u_{\beta com} = \dfrac{u_\beta}{u_{dc}} \end{cases}$$

Equation 27.

5.2.1.5 Angle scale

The angles, such as rotor position, are represented as 16-bit signed fractional values in the range (–1, 1), which corresponds to the angle in the range (–π, π). In a 16-bit signed integer value, the angle is represented as follows.

$$-\pi = 0\mathrm{x}8000$$

Equation 28.

$$\pi = 0\mathrm{x}7\mathrm{FFF}$$

Equation 29.

5.2.2 Scaling for PFC

This section introduces how physical quantities are scaled in PFC control.

5.2.2.1 Current scale

The shunt resistor is cascaded with the MOSFET in the boost circuit, so there is no negative current in the measurement. Similar with the current scaling in motor control, the voltage drop on the shunt is amplified as shown in Figure 29. A very small offset is added to the output considering that the operational amplifier may not output 0 V at the output (namely, not ideal rail-to-rail output).

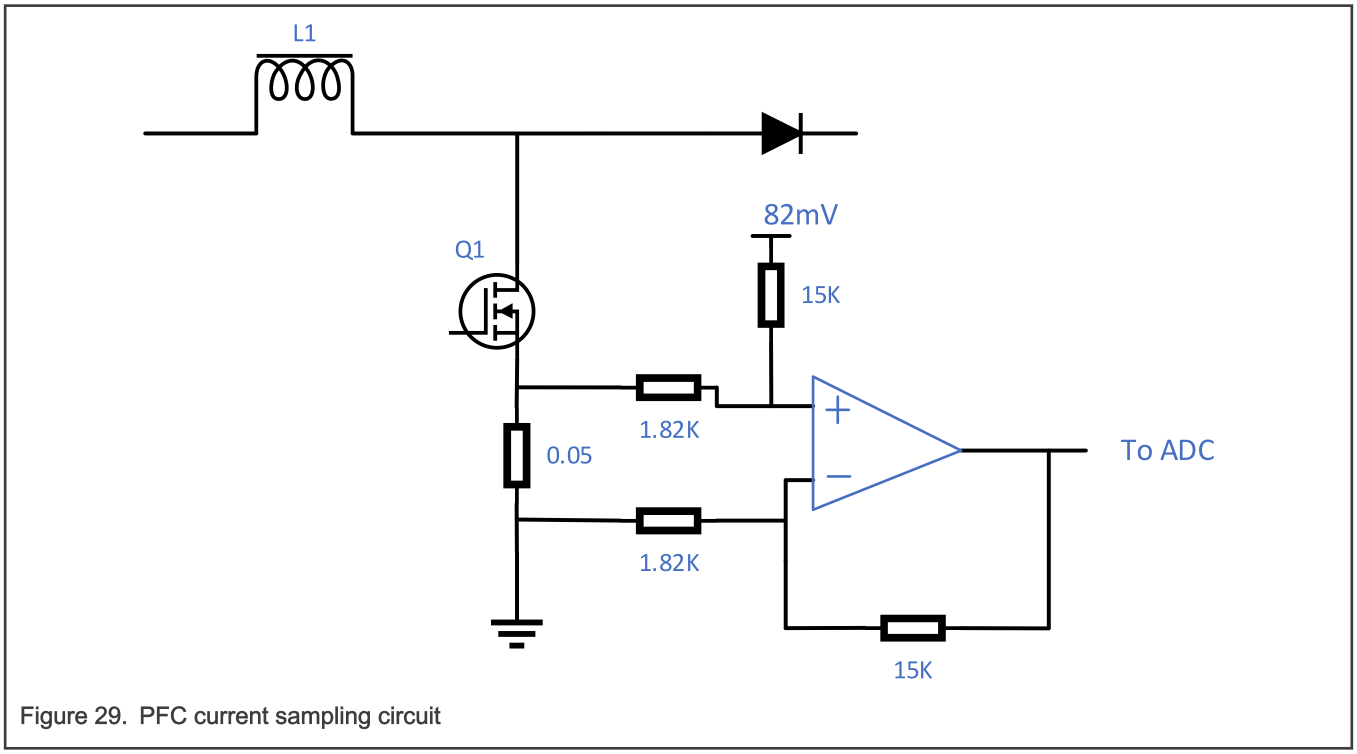


Figure 29. PFC current sampling circuit

In this case, PFC current 0-8A corresponds to fractional 0-1 through the process in Figure 30.

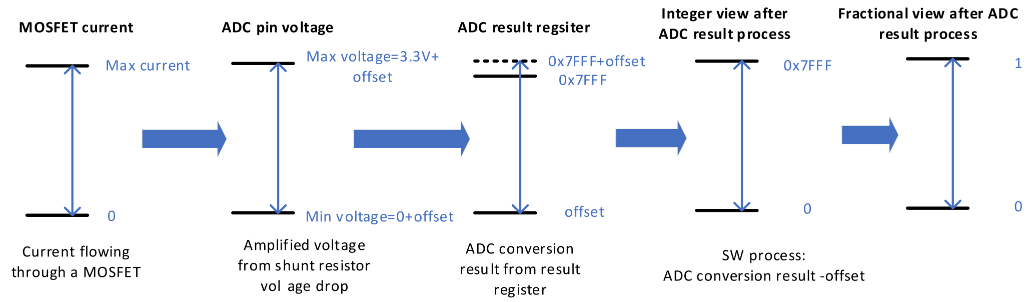

MOSFET current
Max current
0
Current flowing through a MOSFET
ADC pin voltage
Max voltage=3.3V+ offset
Min voltage=0+offset
Amplified voltage from shunt resistor vol age drop
ADC result regsiter
0x7FFF+offset
0x7FFF
offset
ADC conversion result from result register
Integer view after ADC result process
0x7FFF
0
SW process: ADC conversion result -offset
Fractional view after ADC result process
1
0

Figure 30. PFC current scaling

5.2.2.2 Voltage scale

Both motor and PFC control need DC bus voltage, so they share the same circuit as described DC bus voltage scale. AC input voltage is rectified firstly and then sampled with the same resistor divider ladder. The scale is also 433 V.

5.3 Codes architecture

PMSM and PFC control are interrupt driven. There are three ISRs in the system.

- PWMA_COMPARE_0_IRQHANDLER: It is generated by SM0 VAL0 compare match signal with the lowest priority, which is 16 kHz. PMSM FOC fast and slow loops are implemented in this ISR.
- ADC_A_IRQHANDLER: It is generated by ADC sample 11 and 14 ready signal with the highest priority, which is 32 kHz. Input AC voltage phase detection and PFC current loop, realized in a main state machine, is realized in this ISR. PFC current loop is realized in RUN state.
- PIT0_ISRHANDLER: It is generated by PIT0 periodic interrupt with a middle priority, which is 10 kHz. Input AC voltage peak value detection and PFC voltage loop is realized in this ISR. A simple state machine is implemented here to control the behavior of DC bus voltage controller depending on the loading.

5.4 State machines

There are two state machines for motor control, main and sub state machine and same for PFC. Both PFC and motor control share the same code structures for the main state machine.

5.4.1 Motor control state machines and control loops

The main state machine for PMSM control is implemented in `PWMA_COMPARE_0_IRQHANDLER` with a frequency of 16 kHz. The sub state machine is enabled only in RUN of the main states.

5.4.1.1 Motor control main state machine

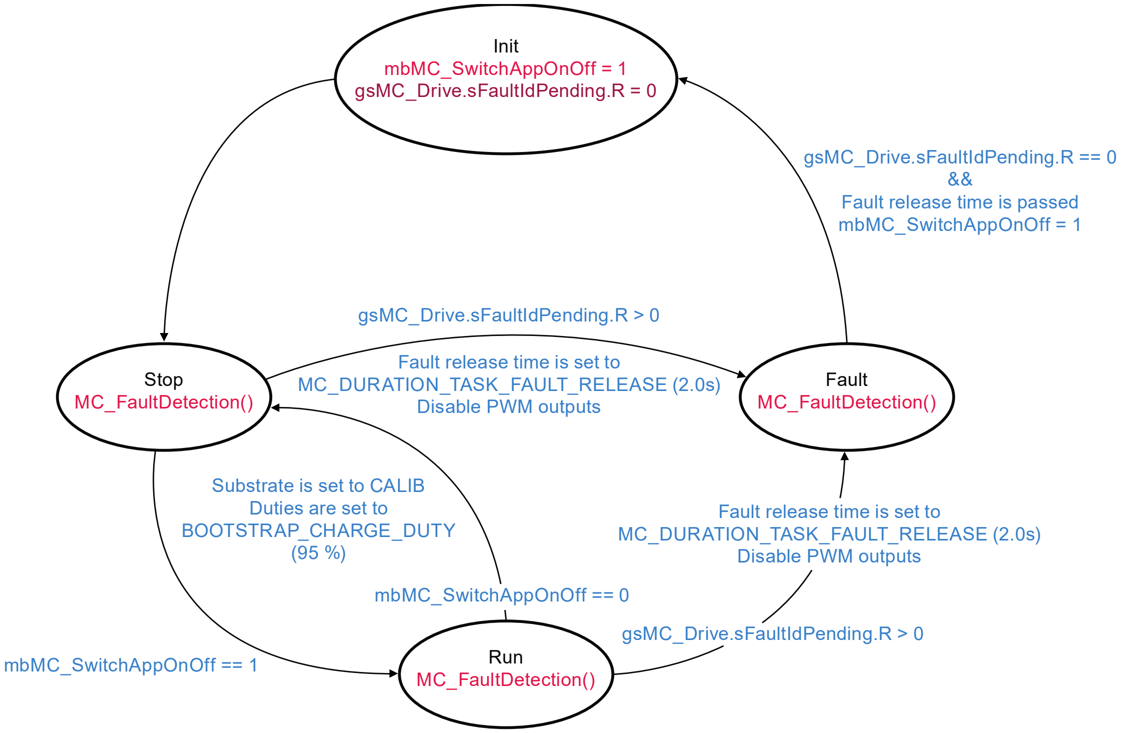

Init
mbMC_SwitchAppOnOff = 1
gsMC_Drive.sFaultIdPending.R = 0
gsMC_Drive.sFaultIdPending.R == 0
&&
Fault release time is passed
mbMC_SwitchAppOnOff = 1
gsMC_Drive.sFaultIdPending.R > 0
Stop
MC_FaultDetection()
Fault release time is set to
MC_DURATION_TASK_FAULT_RELEASE (2.0s)
Disable PWM outputs
Fault
MC_FaultDetection()
Substrate is set to CALIB
Duties are set to
BOOTSTRAP_CHARGE_DUTY
(95 %)
Fault release time is set to
MC_DURATION_TASK_FAULT_RELEASE (2.0s)
Disable PWM outputs
mbMC_SwitchAppOnOff == 0
gsMC_Drive.sFaultIdPending.R > 0
mbMC_SwitchAppOnOff == 1
Run
MC_FaultDetection()

Figure 31. Main state machine for motor control

5.4.1.2 Motor control sub state machine

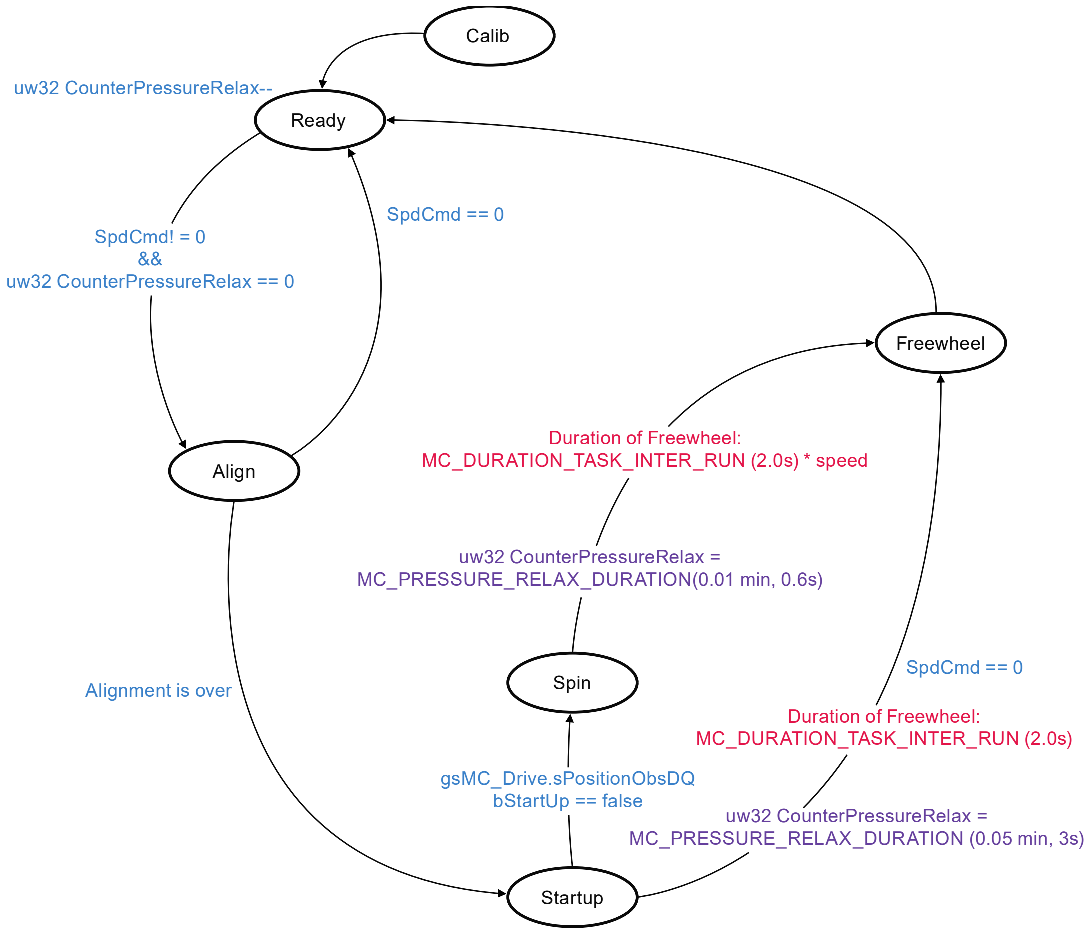

Calib
uw32 CounterPressureRelax--
Ready
SpdCmd == 0
SpdCmd! = 0
&&
uw32 CounterPressureRelax == 0
Freewheel
Duration of Freewheel:
MC_DURATION_TASK_INTER_RUN (2.0s) * speed
Align
uw32 CounterPressureRelax =
MC_PRESSURE_RELAX_DURATION(0.01 min, 0.6s)
Alignment is over
Spin
SpdCmd == 0
Duration of Freewheel:
MC_DURATION_TASK_INTER_RUN (2.0s)
gsMC_Drive.sPositionObsDQ
bStartUp == false
uw32 CounterPressureRelax =
MC_PRESSURE_RELAX_DURATION (0.05 min, 3s)
Startup

Figure 32. Motor control sub state machine

5.4.2 PFC state machines and control loops

A main state machine for PFC is implemented in `ADC_A_IRQHANDLER` ISR with a frequency of 32 kHz. The current loop is realized in RUN state within this ISR, which is called as 32 kHz loop in Figure 23. A sub state machine for PFC voltage controller setting is implemented in `PIT0_ISRHANDLER` with a frequency of 10 kHz. The voltage loop is realized directly within this ISR, which is called 10 kHz loop in Figure 23. Sub state machine is enabled only in RUN of the main states.

5.4.2.1 ADC result ready ISR and PFC main state machine

Within `ADC_A_IRQHANDLER` ISR, execute the below tasks in sequence.

1. Update the two-phase MOSFET currents and AC input voltage.
2. Analyze AC input voltage phase to get the zero-crossing point and the period. The **sinθ** in Figure 23 is calculated based on these information.
3. Execute the main state machine for PFC as shown in Figure 33.

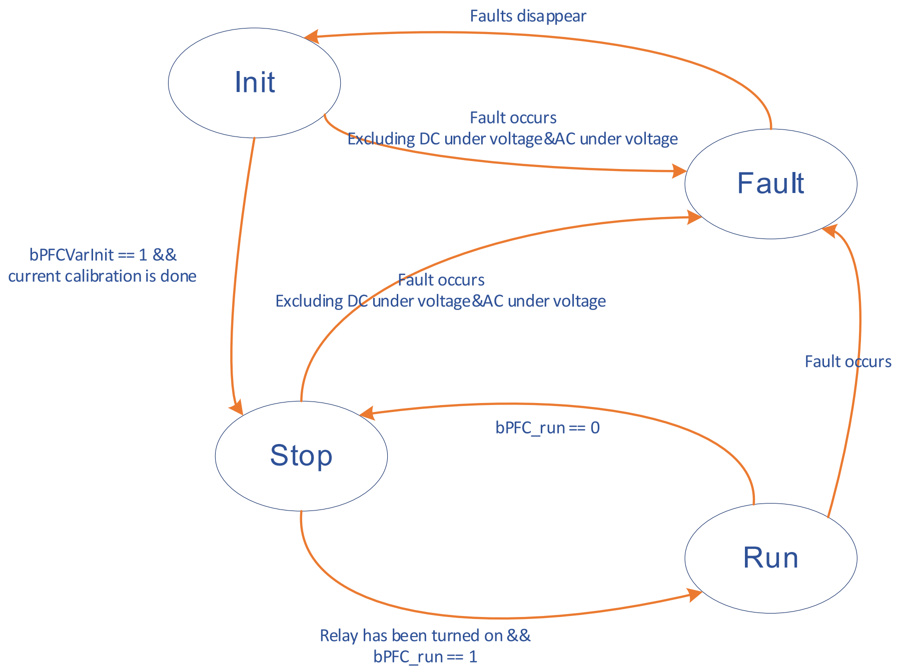

Faults disappear
Init
Fault occurs
Excluding DC under voltage&AC under voltage
Fault
bPFCVarInit == 1 &&
current calibration is done
Fault occurs
Excluding DC under voltage&AC under voltage
Fault occurs
bPFC_run == 0
Stop
Run
Relay has been turned on &&
bPFC_run == 1

Figure 33. PFC main state machine for fast loop

a. **INIT**: This is the first state after power-on reset. All variables are initialized and the current offsets are calibrated within 200 ms. It goes to STOP state after the calibration is done.

b. **STOP**: PFC circuit is turned off in this state. It monitors AC input voltage and DC bus voltage to determine if DC bus relay should be turned on or off. After the relay is turned on, it monitors the value of `bPFC_RUN` to start PFC control. The variable value can be modified through FreeMASTER. See Figure 34 for details.

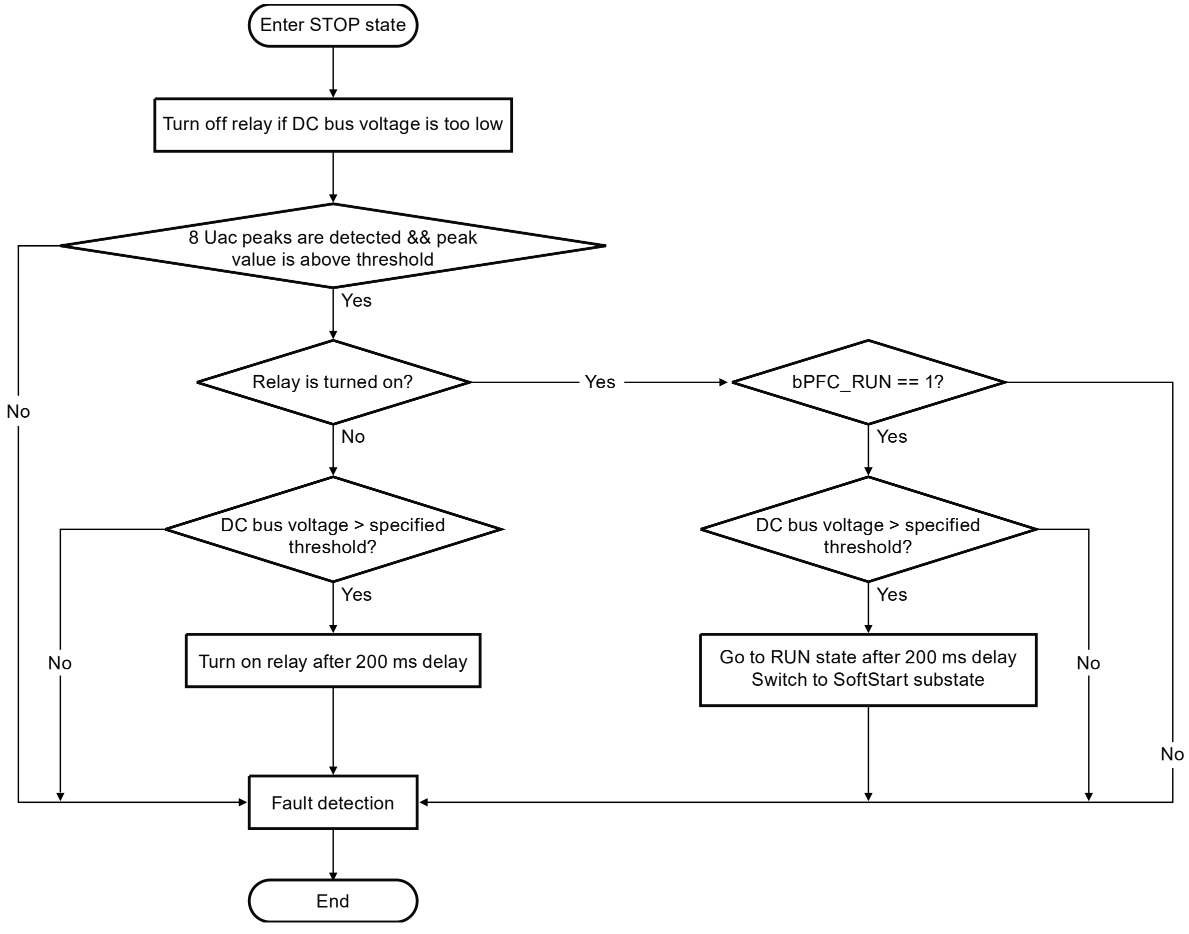

Enter STOP state
Turn off relay if DC bus voltage is too low
8 Uac peaks are detected && peak value is above threshold
Yes
No
Relay is turned on?
Yes
bPFC_RUN == 1?
No
Yes
DC bus voltage > specified threshold?
DC bus voltage > specified threshold?
Yes
Yes
No
Turn on relay after 200 ms delay
Go to RUN state after 200 ms delay
Switch to SoftStart substate
No
No
Fault detection
End

Figure 34. STOP state flowchart for PFC control

c. **RUN**: The state machine goes to this state when relay is turned on and `bPFC_RUN` is 1. Current loop is implemented in this state, and the output duty is compensated depending on Discontinuous Current Mode (DCM) or Continuous Current Mode (CCM). See Figure 35 for details.

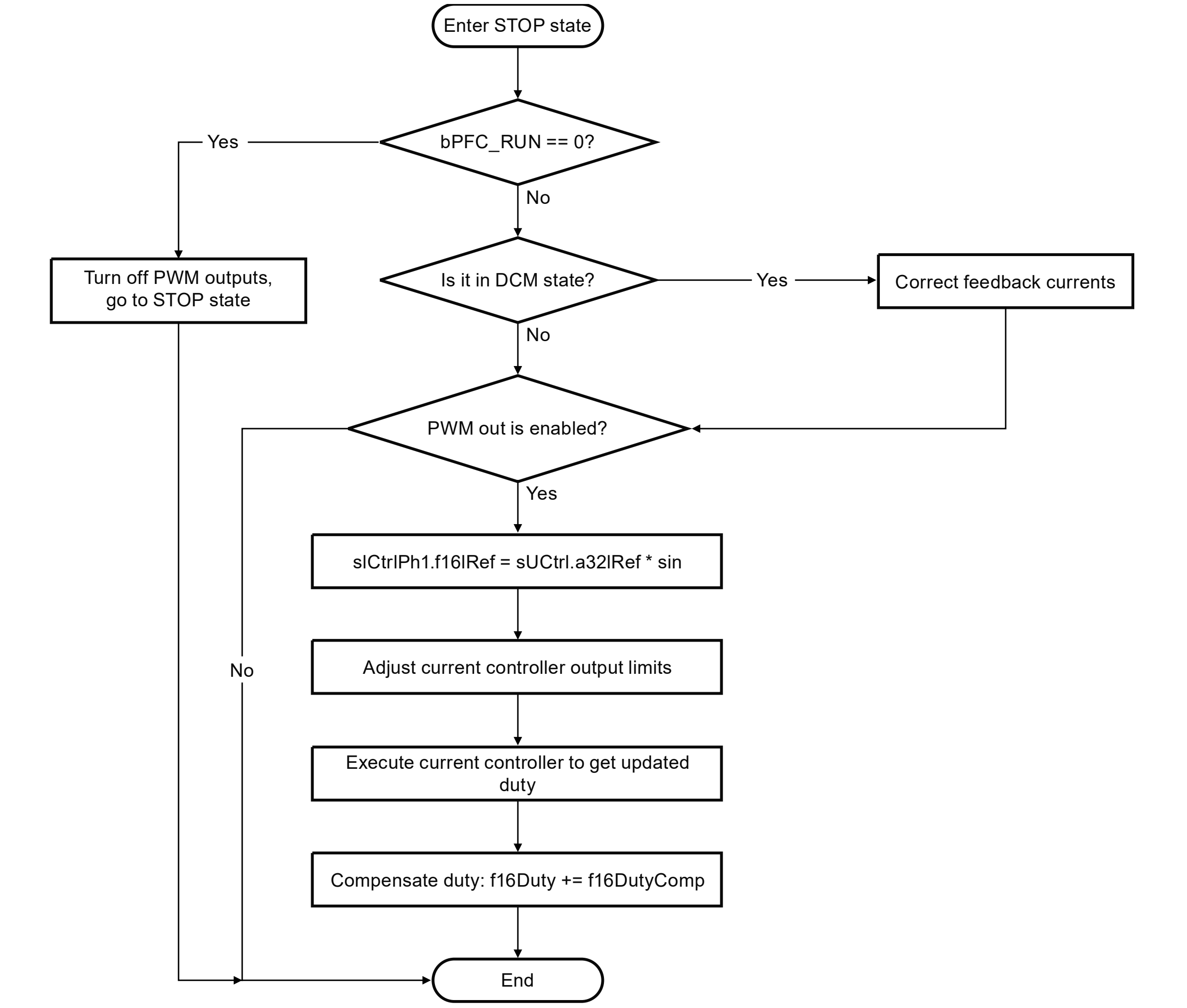

Enter STOP state
bPFC_RUN == 0?
Yes
No
Turn off PWM outputs, go to STOP state
Is it in DCM state?
Yes
Correct feedback currents
No
PWM out is enabled?
Yes
No
slCtrlPh1.f16IRef = sUCtrl.a32IRef * sin
Adjust current controller output limits
Execute current controller to get updated duty
Compensate duty: f16Duty += f16DutyComp
End

Figure 35. RUN state flowchart for PFC control

d. **FAULT**: When there is over/under voltage on DC bus or AC input peak, over current, over/under AC input frequency, the state machine goes to this state and then INIT state after the faults disappear. DC bus under voltage and AC input peak under voltage protection are not enabled when PFC control is not enabled as shown in Figure 33.

5.4.2.2 PIT0 periodic ISR and PFC sub state machine

Within `PIT0_ISRHANDLER` ISR, execute the below tasks in sequence.

1. Update feedback DC bus voltage.
2. Detect AC input voltage peak value.
3. When main state machine is in **RUN** state, execute sub state machine, which is to configure voltage loop controller settings considering the DC bus loading change.

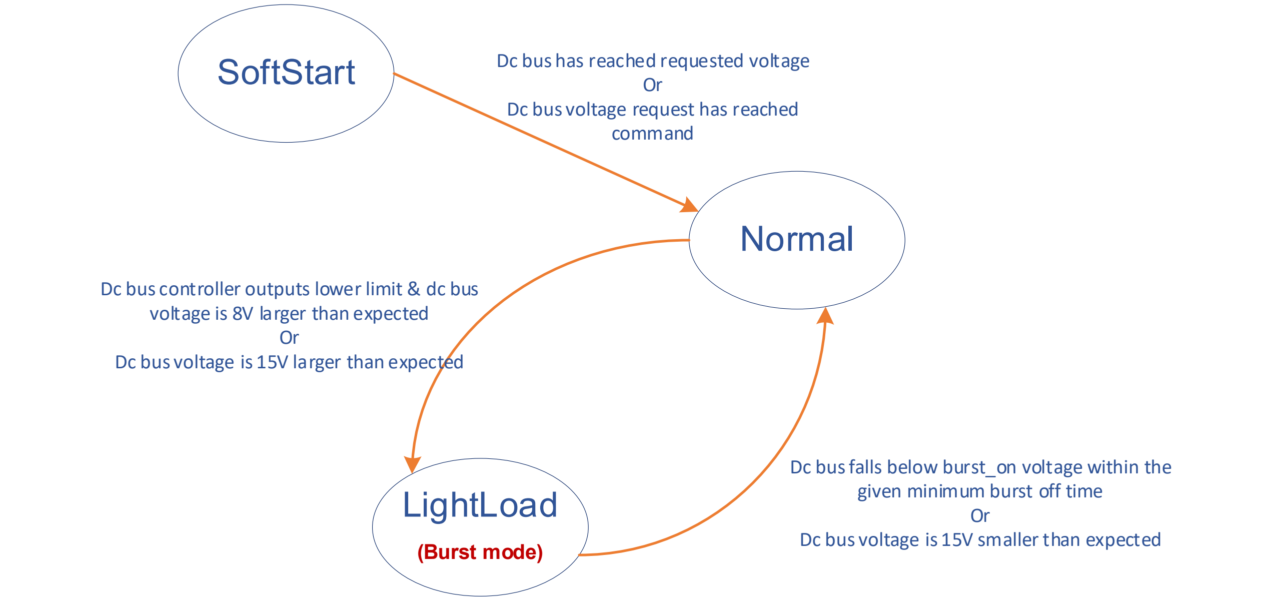

SoftStart
Dc bus has reached requested voltage
Or
Dc bus voltage request has reached command
Normal
Dc bus controller outputs lower limit & dc bus voltage is 8V larger than expected
Or
Dc bus voltage is 15V larger than expected
Dc bus falls below burst_on voltage within the given minimum burst off time
Or
Dc bus voltage is 15V smaller than expected
LightLoad
(Burst mode)

Figure 36. PFC state machine for slow loop

a. **SOFTSTART**: PFC control is just started. Voltage PI controller output lower limit is set to **0** considering that DC bus voltage increase fast through rectifier bridge when the loading is light. A ramp is implemented here for DC bus voltage request. It goes to **NORMAL** state when actual DC bus voltage has reached command voltage during request voltage ramping (loading is light), or request voltage has reached command but the actual DC bus voltage has not reached command yet (loading is heavy). See Figure 23 and Figure 39. Voltage controller output is used as current amplitude reference for current loop.

b. **NORMAL**: DC bus loading is relatively high. Voltage PI controller output lower limit is no longer zero and voltage controller output is used as current amplitude reference for current loop. When DC bus voltage overshoot a lot or the voltage controller always output the lower limit value, the sub state goes to **LIGHTLOAD** state.

c. **LIGHTLOAD**: DC bus loading is relatively low. This state is also referred to as burst mode. Voltage PI controller output lower limit is no longer zero. Because the loading is low, disable PWM output when DC bus is higher than `PFC_U_DCB_BUSRT_OFF` which is called **burst off**, and enable PWM output when DC bus drops below `PFC_U_DCB_BURST_ON` which is called **burst on**. During **burst on** stage, only the voltage controller output lower limit is used as current amplitude reference. The state goes back to **NORMAL** when loading increases in below two cases:

    i. Because the DC bus loading is low, we expect the **burst off** stage should stay for a minimum amount of time. If DC bus voltage drops `PFC_U_DCB_BURST_ON` within this **minimum burst off time**, we assume the loading increases. See Figure 37 for details.

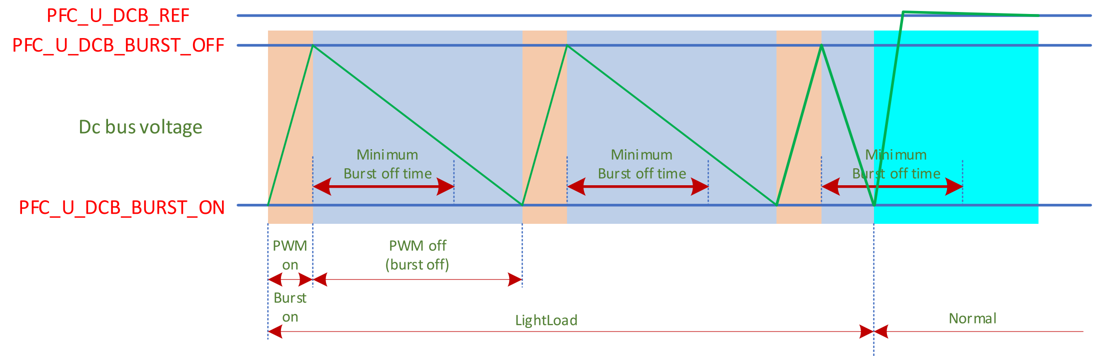


Figure 37. Case 1 - DC bus loading increases and state goes from LIGHTLOAD to NORMAL

ii. When DC bus voltage drops a lot, it goes `PFC_U_DCB_REF - PFC_DC_BUS_BURST_UNDER_DELTA` and the state also goes to NORMAL. See Figure 38 for details.

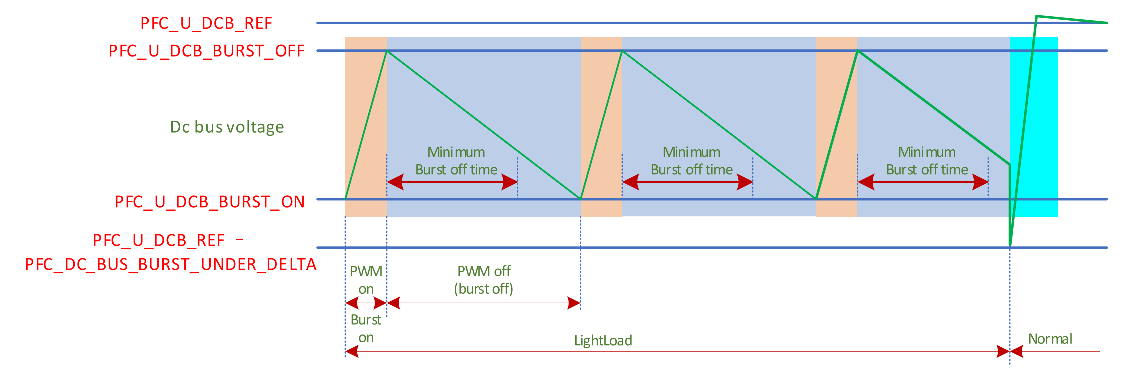
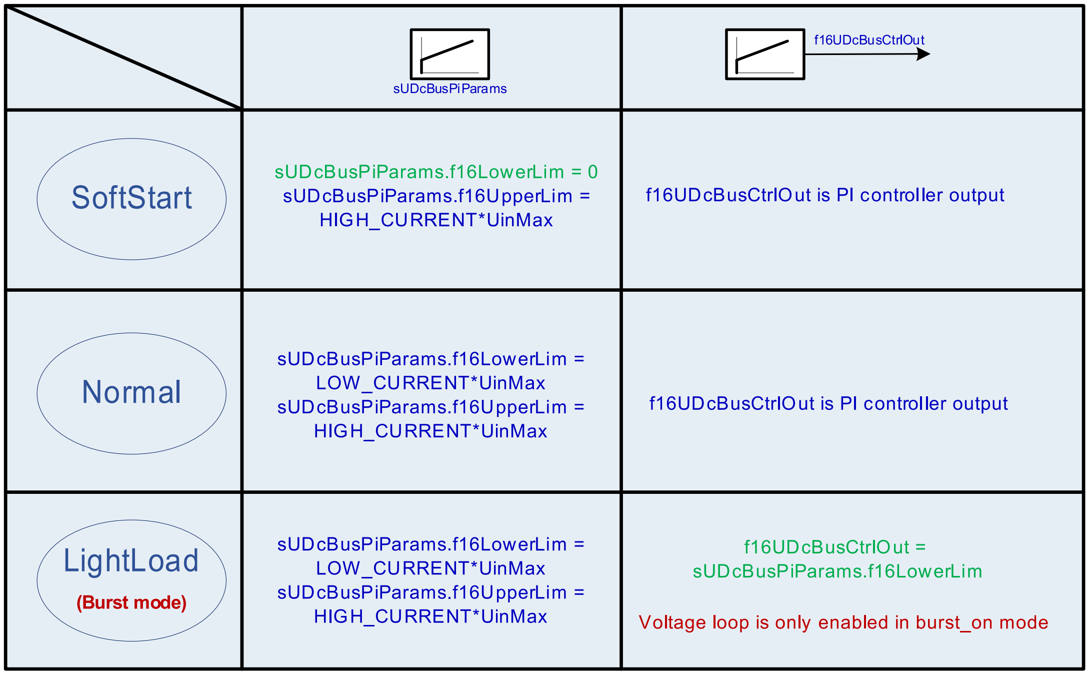


Figure 38. Case 2 - DC bus loading increases and state goes from LIGHTLOAD to NORMAL

d. When main state machine is in RUN state, voltage loop controller is executed when PWM output is enabled. Current amplitude reference is got for current loop. The voltage controller configuration differences, due to this sub state machine, are as shown in Figure 39.



	sUDcBusPiParams	f16UDcBusCtrlOut

SoftStart	sUDcBusPiParams.f16LowerLim = 0
	sUDcBusPiParams.f16UpperLim = HIGH_CURRENT*UinMax
		f16UDcBusCtrlOut is PI controller output

Normal	sUDcBusPiParams.f16LowerLim = LOW_CURRENT*UinMax
	sUDcBusPiParams.f16UpperLim = HIGH_CURRENT*UinMax
		f16UDcBusCtrlOut is PI controller output

LightLoad
(Burst mode)
	sUDcBusPiParams.f16LowerLim = LOW_CURRENT*UinMax
	sUDcBusPiParams.f16UpperLim = HIGH_CURRENT*UinMax
		f16UDcBusCtrlOut = sUDcBusPiParams.f16LowerLim
		Voltage loop is only enabled in burst_on mode

Figure 39. Voltage controller setting and current amplitude reference in sub states

e. DCM and CCM mode is differentiated through circuit model calculation, and the duty for compensation is also calculated.

Figure 40 shows the flowchart in PIT0_ISRHANDLER ISR.

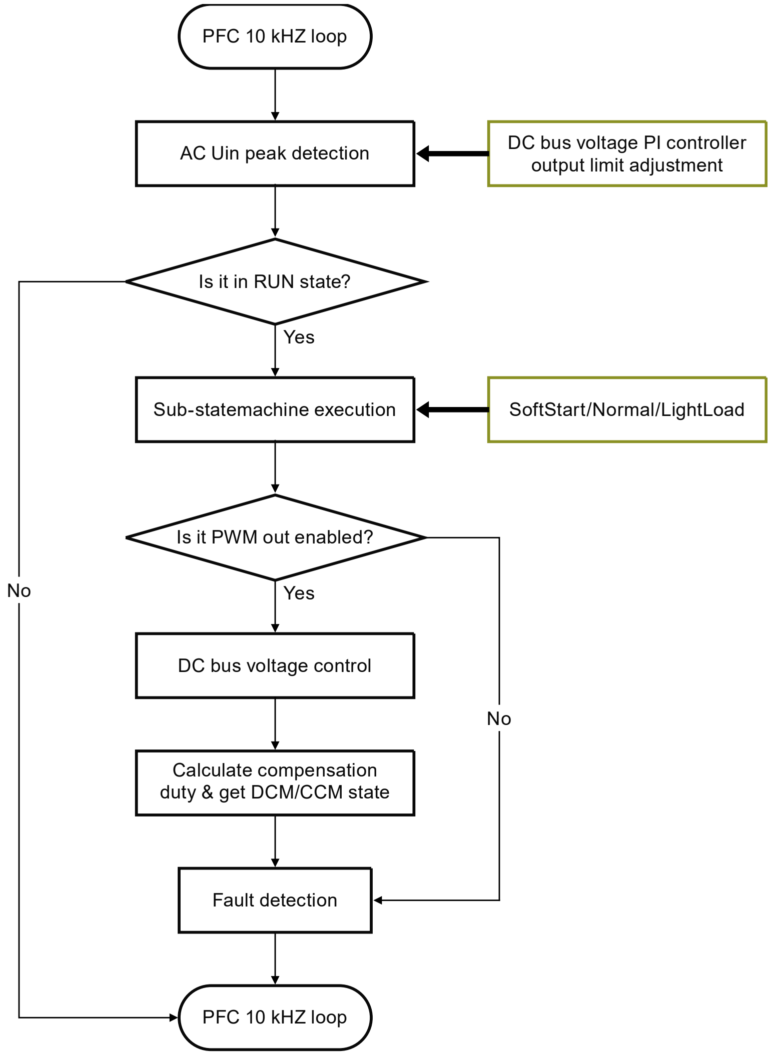

PFC 10 kHZ loop
AC Uin peak detection
DC bus voltage PI controller output limit adjustment
Is it in RUN state?
Yes
No
Sub-statemachine execution
SoftStart/Normal/LightLoad
Is it PWM out enabled?
Yes
No
DC bus voltage control
Calculate compensation duty & get DCM/CCM state
Fault detection
PFC 10 kHZ loop

Figure 40. PIT0 ISR flowchart

6 Experiment results

The full loading is 800 W in this application. DC bus reference is 400 V, several loadings and transitions are tested.

6.1 Hardware setup

The PFC application is built using the High Voltage Motor Control Platform (HVP-MC3PH) with the HVP-56F83783 daughter card. The complete hardware setup is shown in Figure 41:

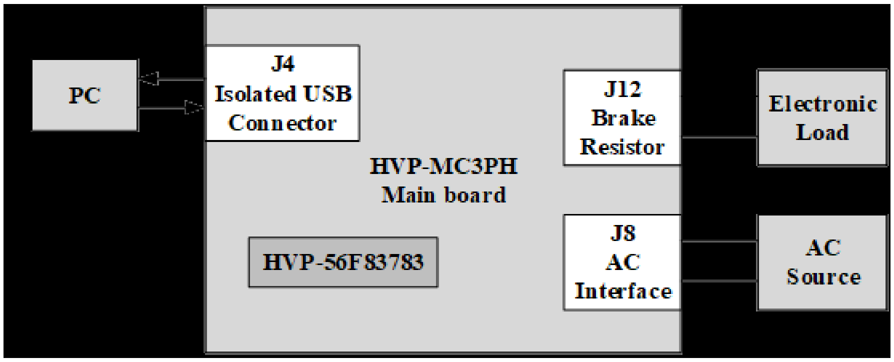


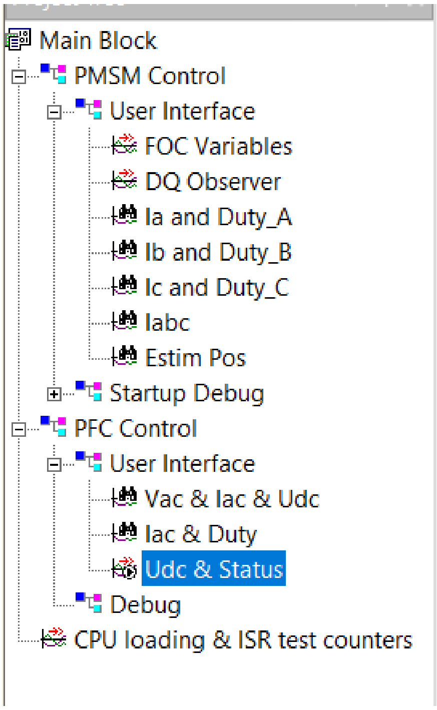
Figure 41. Hardware setup

Both boards are ready for the PFC development in their default configuration, so no jumper setting is needed before running the application. The load is connected through the brake resistor connector J12. The high-voltage power stage can be supplied from the grid or AC source. The supply voltage range is 90-240 VAC. To work properly, note that if an electronic load is used to test the HVP-MC3PH, it is necessary to add an isolated transformer between the grid and the electronic load. The PE on the HVP-MC3PH board should not be connected with the grid earth, such as adding an isolated transformer between the grid and the HVP-MC3PH. If it is necessary to connect the PE on the board with earth for safety consideration, remove the C19 and C20 capacitors from the HVP-MC2PH board.

6.2 Test results

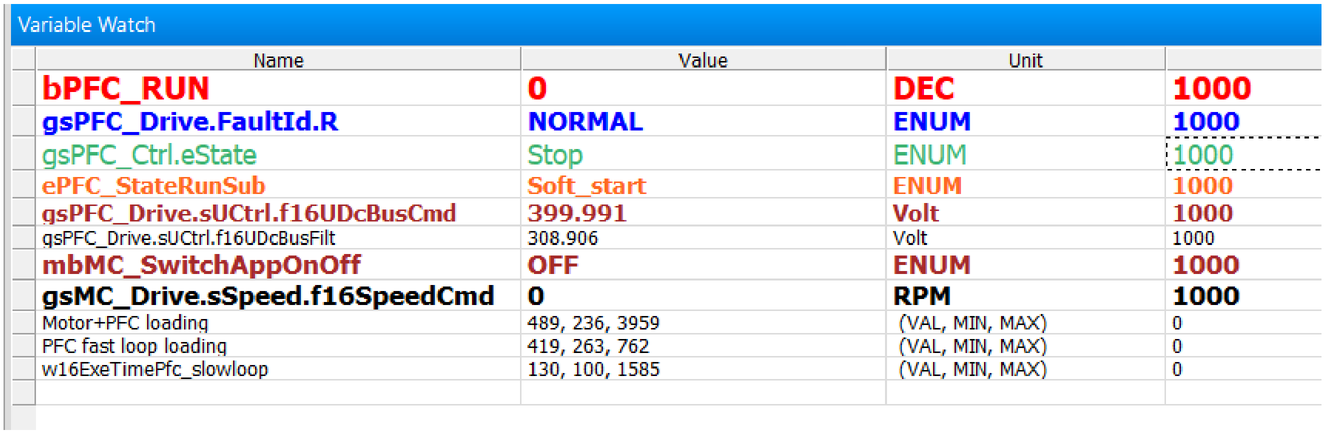
- Channel 1 is input AC voltage.
- Channel 4 is input current.
- Channel 3 is DC bus voltage.
- Freemaster is used as the control and display panel.
- An SCI with 115200 baud rate is the communication port.

To test PFC performance, open *56F83783_HVP_PMSM_PFC_release.pmp* with the latest FreeMASTER version, go to **PFC Control** block, and activate **Udc & Status** scope.



- Main Block
  - PMSM Control
    - User Interface
      - FOC Variables
      - DQ Observer
      - Ia and Duty_A
      - Ib and Duty_B
      - Ic and Duty_C
      - Iabc
      - Estim Pos
    - Startup Debug
  - PFC Control
    - User Interface
      - Vac & Iac & Udc
      - Iac & Duty
      - Udc & Status
    - Debug
  - CPU loading & ISR test counters

Figure 42. PFC and motor control blocks in FreeMASTER project tree

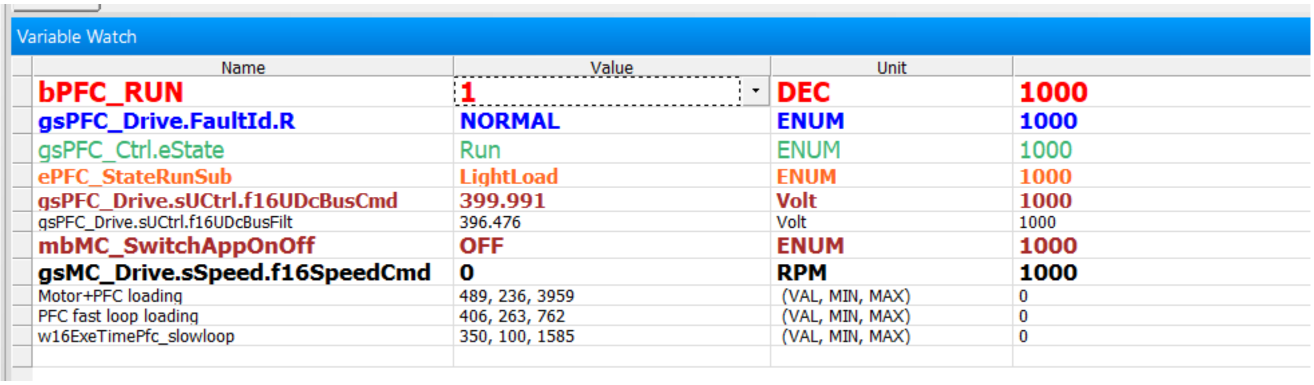


Variable Watch

Name	Value	Unit	
bPFC_RUN	0	DEC	1000
gsPFC_Drive.FaultId.R	NORMAL	ENUM	1000
gsPFC_Ctrl.eState	Stop	ENUM	1000
ePFC_StateRunSub	Soft_start	ENUM	1000
gsPFC_Drive.sUCtrl.f16UDcBusCmd	399.991	Volt	1000
gsPFC_Drive.sUCtrl.f16UDcBusFilt	308.906	Volt	1000
mbMC_SwitchAppOnOff	OFF	ENUM	1000
gsMC_Drive.sSpeed.f16SpeedCmd	0	RPM	1000
Motor+PFC loading	489, 236, 3959	(VAL, MIN, MAX)	0
PFC fast loop loading	419, 263, 762	(VAL, MIN, MAX)	0
w16ExeTimePfc_slowloop	130, 100, 1585	(VAL, MIN, MAX)	0

Figure 43. Variable Watch pane

In the **Variable Watch**, input **1** for **bPFC_RUN** to enable PFC control.

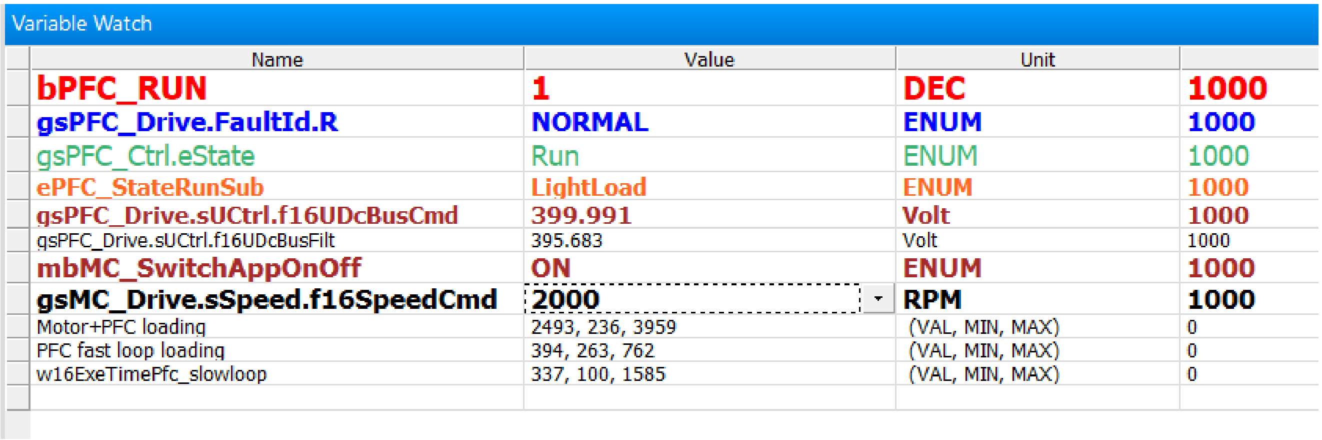


Variable Watch

Name	Value	Unit	
bPFC_RUN	1	DEC	1000
gsPFC_Drive.FaultId.R	NORMAL	ENUM	1000
gsPFC_Ctrl.eState	Run	ENUM	1000
ePFC_StateRunSub	LightLoad	ENUM	1000
gsPFC_Drive.sUCtrl.f16UDcBusCmd	399.991	Volt	1000
gsPFC_Drive.sUCtrl.f16UDcBusFilt	396.476	Volt	1000
mbMC_SwitchAppOnOff	OFF	ENUM	1000
gsMC_Drive.sSpeed.f16SpeedCmd	0	RPM	1000
Motor+PFC loading	489, 236, 3959	(VAL, MIN, MAX)	0
PFC fast loop loading	406, 263, 762	(VAL, MIN, MAX)	0
w16ExeTimePfc_slowloop	350, 100, 1585	(VAL, MIN, MAX)	0

Figure 44. Enable PFC control by setting 1 to bPFC_RUN

Set mbMC_SwitchAppOnOff to ON and set a speed value. gsMC_Drive.sSpeed.f16SpeedCmd will start the motor.



Variable Watch

Name	Value	Unit	
bPFC_RUN	1	DEC	1000
gsPFC_Drive.FaultId.R	NORMAL	ENUM	1000
gsPFC_Ctrl.eState	Run	ENUM	1000
ePFC_StateRunSub	LightLoad	ENUM	1000
gsPFC_Drive.sUCtrl.f16UDcBusCmd	399.991	Volt	1000
gsPFC_Drive.sUCtrl.f16UDcBusFilt	395.683	Volt	1000
mbMC_SwitchAppOnOff	ON	ENUM	1000
gsMC_Drive.sSpeed.f16SpeedCmd	2000	RPM	1000
Motor+PFC loading	2493, 236, 3959	(VAL, MIN, MAX)	0
PFC fast loop loading	394, 263, 762	(VAL, MIN, MAX)	0
w16ExeTimePfc_slowloop	337, 100, 1585	(VAL, MIN, MAX)	0

Figure 45. Enable motor control by setting mbMC_SwitchAppOnOff to ON and giving a non-zero speed command

A programmable AC source, Chroma, model 61704, is used as AC voltage input for below tests.

Input AC voltage (V)	Electric load (W)	Power factor
220	199.85	0.987
220	400.5	0.992
220	600	0.997
220	800	0.997
110	100.3	0.99
110	200	0.996
110	300	0.998
110	400	0.998

6.2.1 220 VAC, 50 Hz, no load

PFC works at burst mode. Because the load is very light, the burst-off time is much longer than the burst-on time.

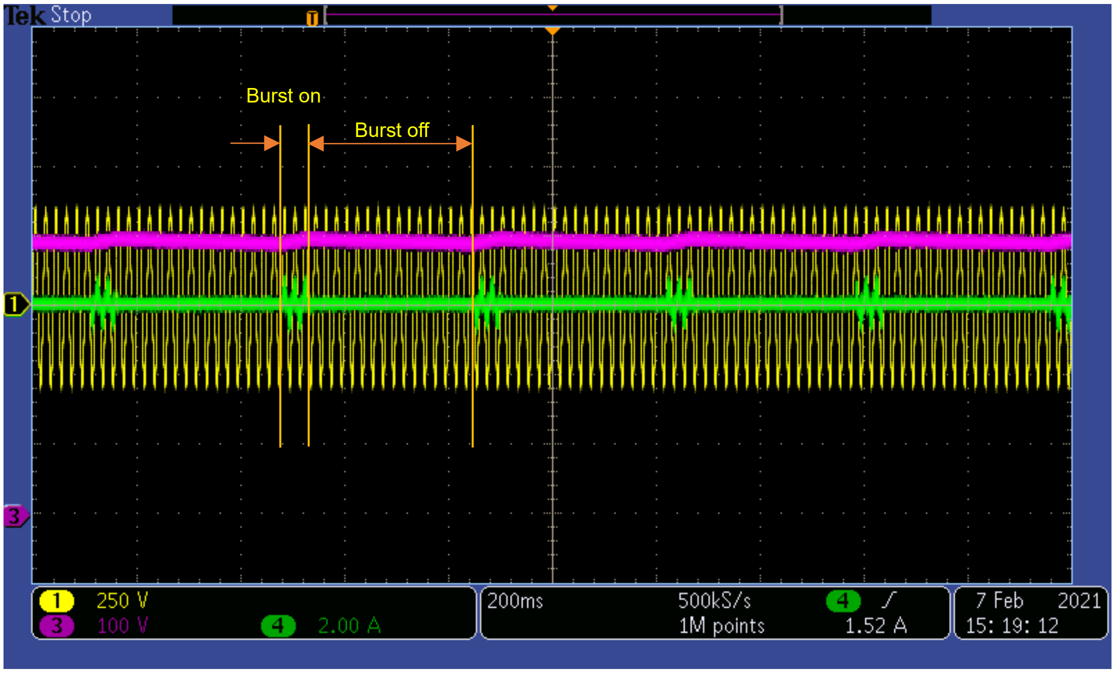


Figure 46. AC input voltage/current and DC voltage with no load

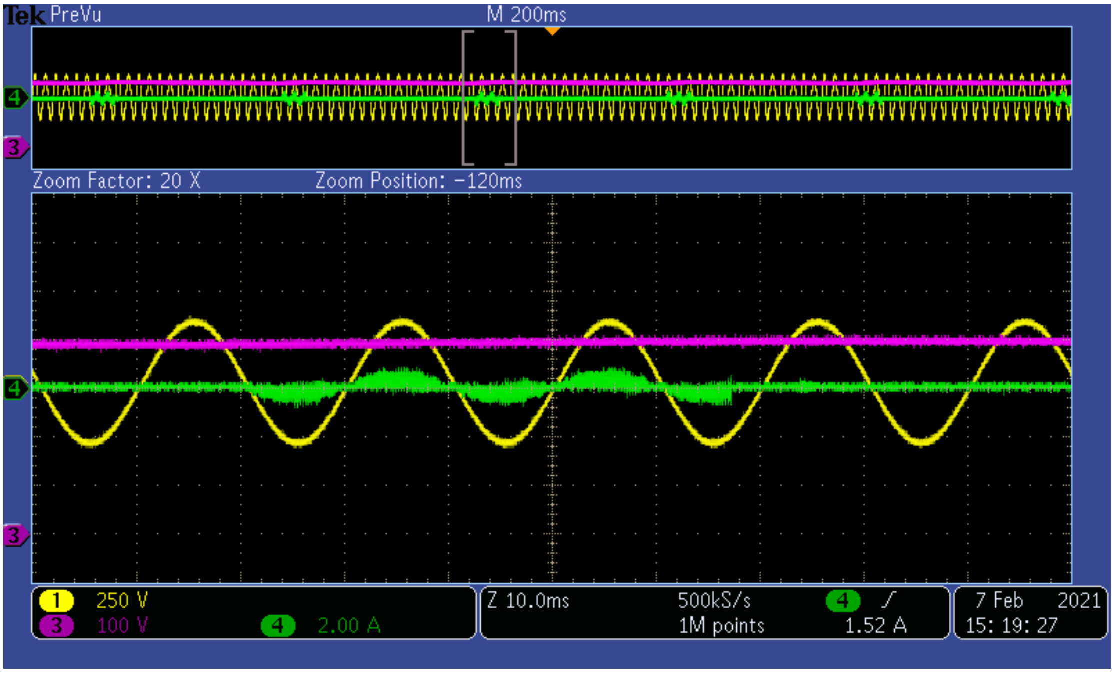


Figure 47. AC input voltage/current and DC voltage with no load – zoom in

6.2.2 220 VAC, 50 Hz, 30 W motor load

The entire load is contributed by the motor running, and there is no load on motor shaft. So the DC bus load is also light, but a bit larger than no load. PFC still works at burst mode. Now burst-on time is longer than burst-off time.

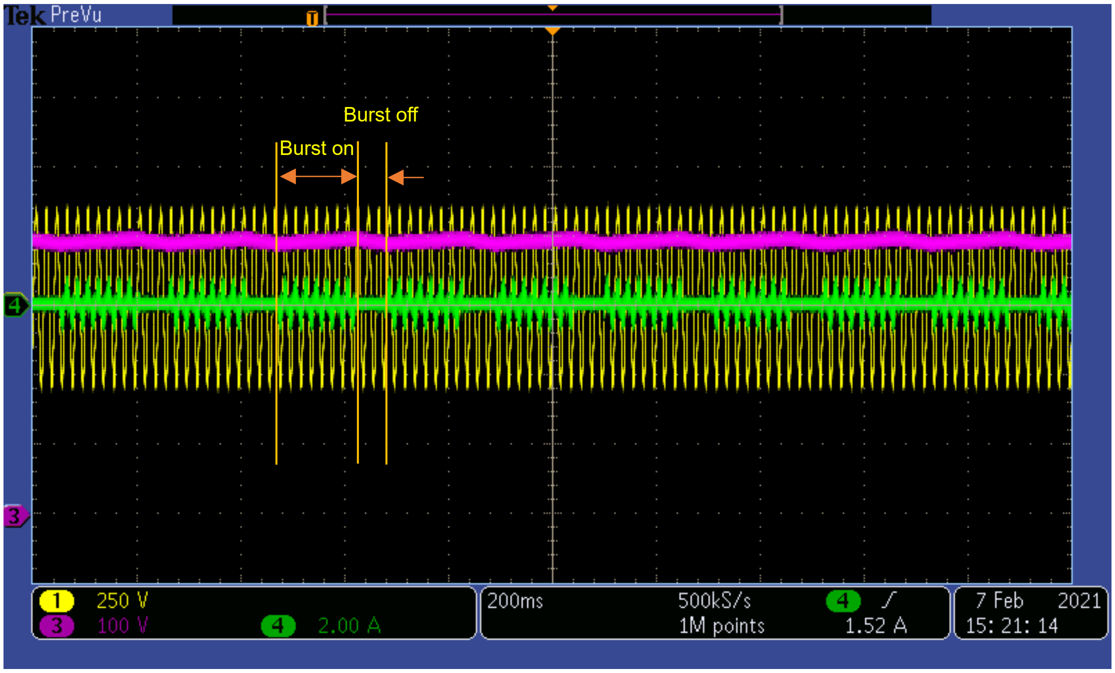


Figure 48. AC input voltage/current and DC voltage with motor load

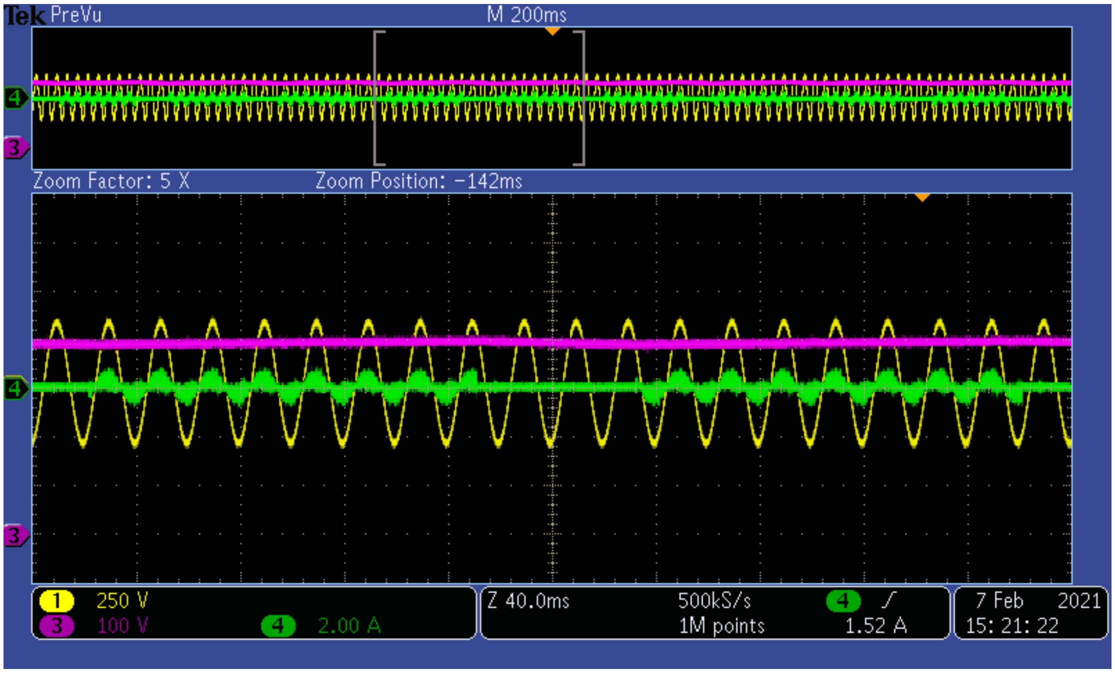


Figure 49. AC input voltage/current and DC voltage with motor load – zoom in

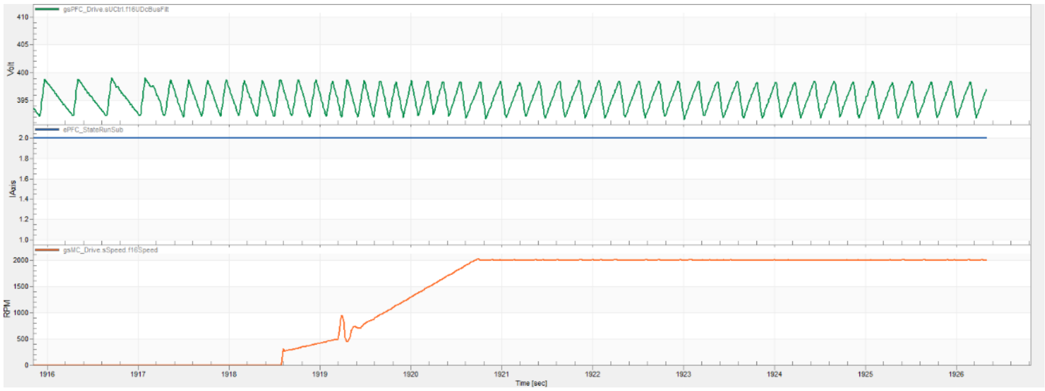


Figure 50. DC voltage – from no load to starting motor

6.2.3 220 VAC, 50 Hz, 200 W electric load

PFC works at normal mode. DC electric load is added through braking resistor connector.

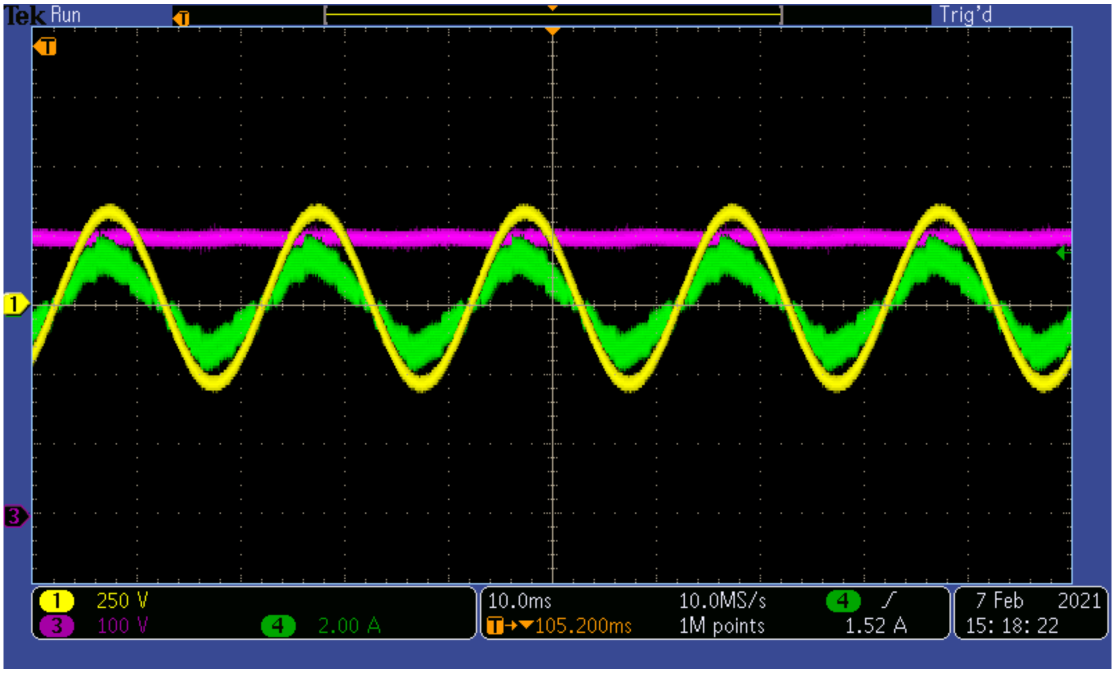


Figure 51. AC input voltage/current and DC voltage with 200 W load

6.2.4 220 VAC, 50 Hz, 400 W electric load

PFC works at normal mode. DC electric load is added through braking resistor connector.

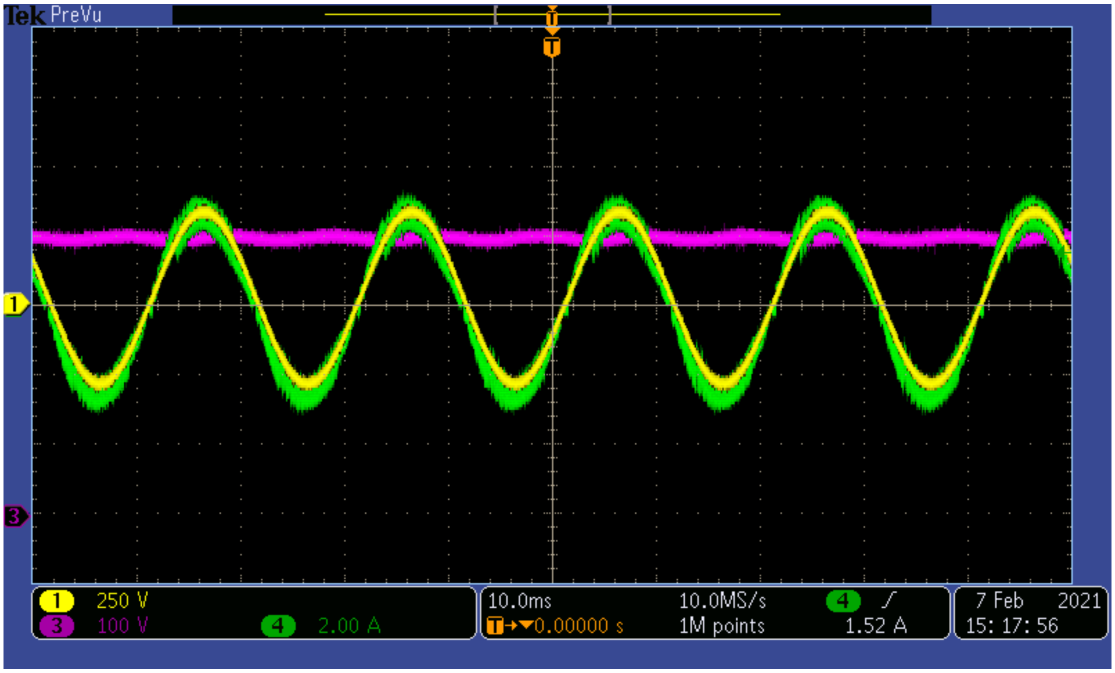


Figure 52. AC input voltage/current and DC voltage with 400 W load

6.2.5 220 VAC, 50 Hz, 800 W electric load

PFC works at normal mode. DC electric load is added through braking resistor connector.

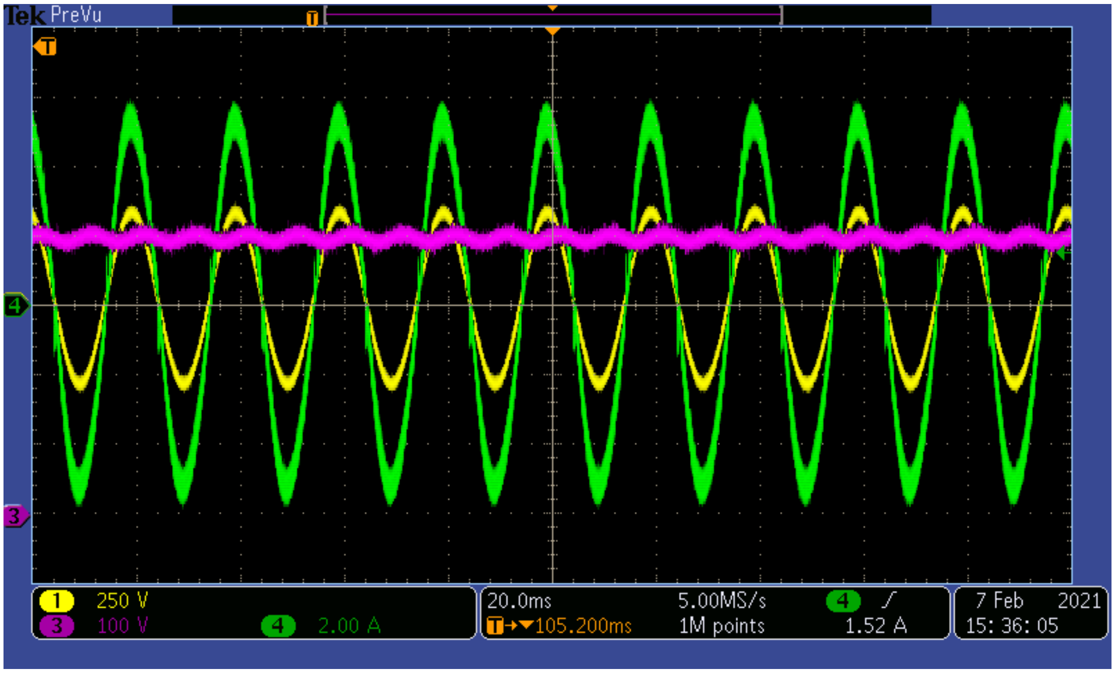


Figure 53. AC input voltage/current and DC voltage with 800 W load

6.2.6 220 VAC, 50 Hz, 0 W to 800 W load transition

When 800 W load is added abruptly, PFC goes from burst mode to normal mode immediately.

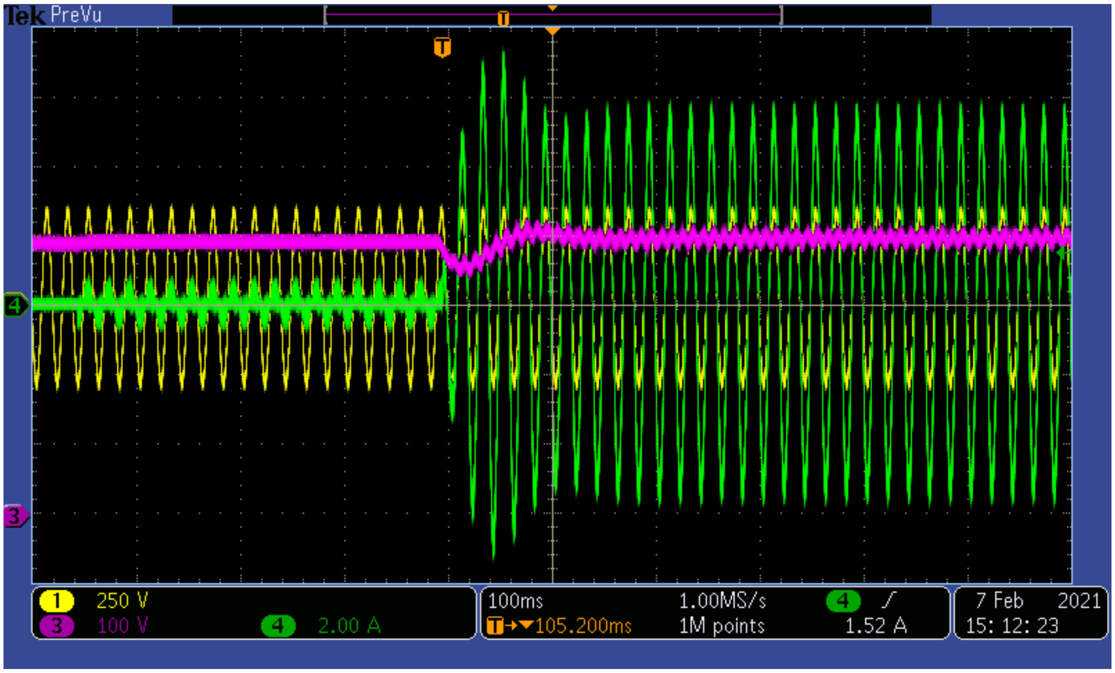


Figure 54. AC input voltage/current and DC voltage with load changing from 0 W to 800 W

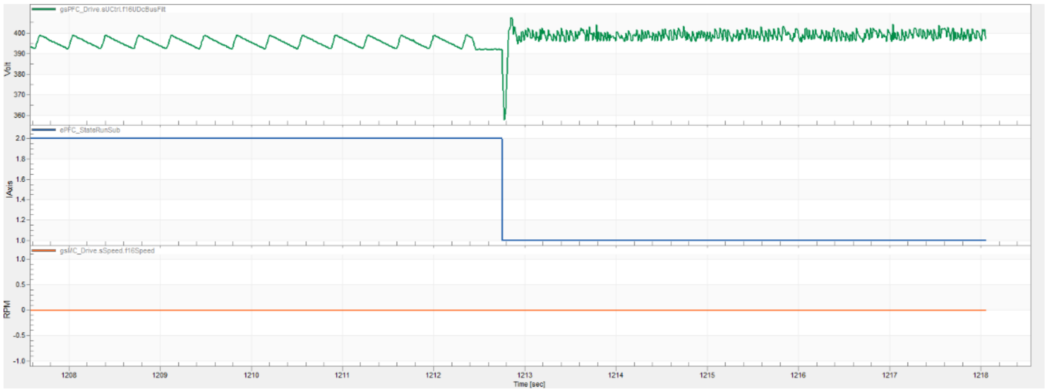


Figure 55. DC voltage – from no load to 800 W

6.2.7 110 VAC, 50 Hz, 800 W to 0 load transition

When 800 W load is taken away abruptly, PFC goes from normal mode to burst mode immediately.

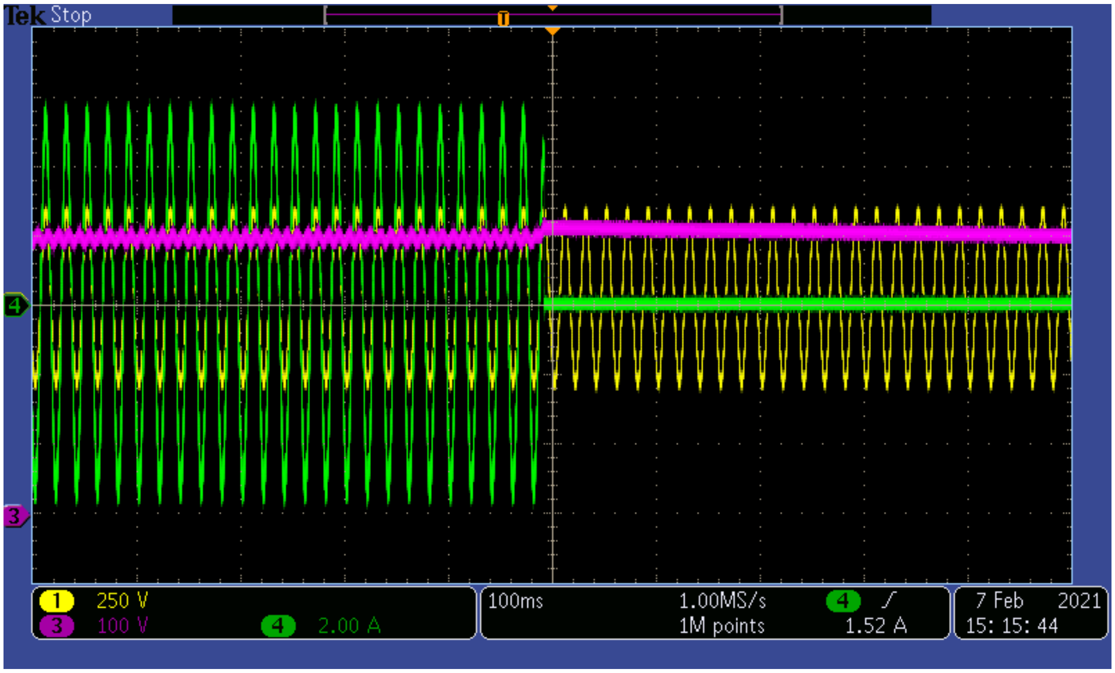


Figure 56. AC input voltage/current and DC voltage with load changing from 800 W to 0

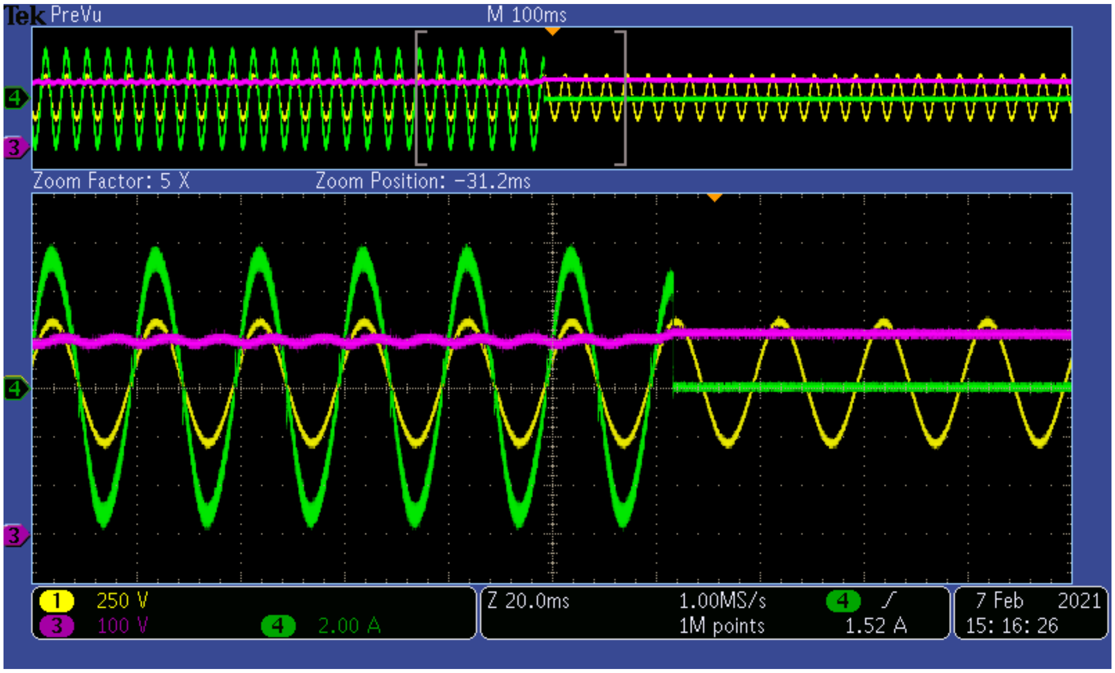


Figure 57. AC input voltage/current and DC voltage with load changing from 800 W to 0 – zoom in

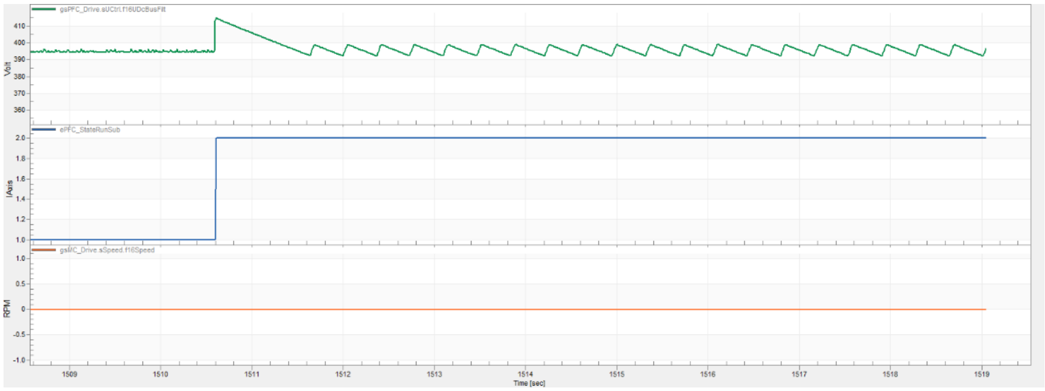


Figure 58. DC voltage – from 800 W to no load

7 Revision history

Table 4. Revision history

Rev.	Date	Description
1	08 March 2022	Updated Figure 17,Software design, and Experiment results.
0	March 2021	Initial release.

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

Right to make changes - NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer’s applications and products. Customer’s responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer’s applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 08 March 2022
Document identifier: AN13184