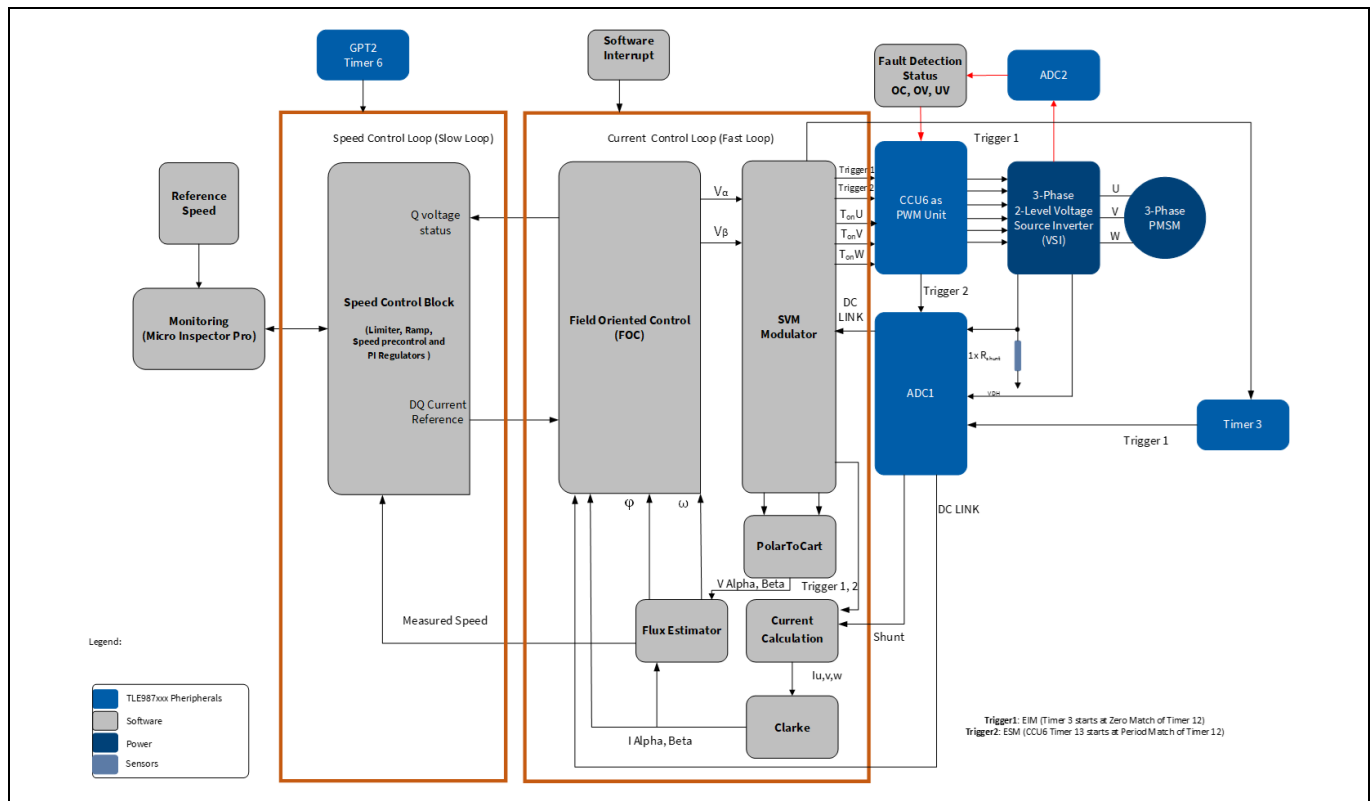


MOTIX™ TLE987x Motor Control FOC Demo

SW Version 1.3.0



About this document

Scope and purpose

This document describes the implementation of the MOTIX™ TLE987x Motor Control FOC Demo (called also “BASIC SW”), capable to drive 3-phase BLDC motors using the Infineon MOTIX™ TLE987x device. This document shall be used along with the “Motor Control Software Libraries BASIC for TLE987x Getting started” (included in [SW package](#)), “[TLE987x User Manual](#)”, “[TLE986x/7x BootROM User Manual](#)”, “[TLE987x/6x Hardware Design Application Note](#)”, “[Sensorless Field Oriented Control with TLE987x](#)”.

Intended audience

This document is intended for users already familiar with FOC as motor control solution, and who are going to use the BASIC SW in combination with the TLE987x microcontrollers for BLDC applications.

Table of contents

About this document.....	1
Table of contents.....	2
1 Terms and abbreviations	4
2 Introduction	5
2.1 Content of the Application Note	5
2.2 FOC as motor control technique.....	5
3 FOC implementation in the BASIC SW	7
3.1 Initialization of the parameters	7
3.2 Main endless loop.....	8
3.2.1 Start Angle Identification.....	9
3.3 Rotor alignment	10
3.4 FOC mode	10
3.4.1 Open loop during start-up	11
3.4.2 Transitions from open to closed and from closed to open loop	13
3.4.3 Closed loop control.....	15
3.5 VToF mode.....	20
3.6 TLE987x features used in the BASIC SW	21
3.6.1 PWM generation	21
3.6.2 ADC	22
3.6.2.1 ADC triggering with PWM	23
3.6.2.2 ADC read with PWM.....	25
3.6.3 Bridge Driver.....	25
3.6.4 Shunt resistor and CSA	26
4 Structure of the BASIC SW	27
4.1 MCTRL Solution (“FOC Solution”).....	28
4.2 MCTRL Drive Algorithm	28
4.3 MCTRL Actuator Sensor.....	30
4.4 MCTRL Hardware Abstraction Layer.....	30
4.5 MCTRL Math.....	32
4.6 MCU HW Peripherals	33
5 Configuration of the static parameters	34
5.1 Application layer	34
5.2 MCTRL solution.....	35
5.2.1 Base values.....	35
5.2.2 System parameters	37
5.2.3 PI controllers	39
5.2.4 General operation settings	41
5.3 MCTRL drive algorithm.....	44
5.3.1 Flux estimator	44
5.3.2 FOC controller	46
5.3.3 I/f software component	48
5.3.4 V/f software component	48
5.3.5 Start Angle Identification.....	49
5.4 MCTRL actuator sensor	51
5.4.1 System parameters	51
5.4.2 Measurement trigger.....	54
5.4.3 Fault options	56
5.5 MCTRL Hardware Abstraction.....	57

Table of contents

5.5.1	Measurement ADC.....	57
5.5.2	Pattern generator.....	58
5.5.3	Bridge driver	59
5.6	Math library.....	60
5.7	TLE987x peripherals.....	60
6	Dynamic parameters for the user	62
6.1	Functional flow chart	62
7	Actions needed to run the new motor with MOTIX software	63
7.1	Minimum parameters configuration to spin a motor	64
7.2	Example of a Nanotec motor with no load.....	65
7.3	Example with MOTIX™ 1 kW Engine Cooling Fan reference design.....	66
7.3.1	Challenges when turning a motor for the first time.....	68
7.4	Methods to determine the motor parameters	70
7.4.1	Phase resistance.....	71
7.4.2	Phase inductance.....	71
7.4.3	Pole pairs	71
8	Fault types, configuration and indication	73
8.1	Types of fault	73
8.1.1	Software fault.....	73
8.1.2	Hardware fault	73
8.2	Fault configuration.....	74
8.3	Indication of faults	75
8.3.1	Indication for faults in the debug watch window	75
8.3.2	Indication for fault in the Micro Inspector Pro	75
9	Execution time and memory consumption	76
9.1	Execution time of the motor control software	76
9.2	Memory consumption of the motor control software	78
	References.....	79
	Revision history.....	80
	Disclaimer.....	81

1 Terms and abbreviations

Table 1 Terms and abbreviations

Acronym or term	Definition or explanation
3ph	Three phases
ADC	Analog to Digital Converter
BLDC	Brushless Direct Current Motor
BDRV	Bridge driver
CCU	Capture Compare Unit
CSA	Current Sense Amplifier
DQ	D (direct) and Q (quadrature), used to describe the rotational coordinates
FOC	Field Oriented Control
HSS	High Side Switch
HW	Hardware
IToF	Current to frequency
LLD	Low Level Driver
LSS	Low Side Switch
MCTRL	Motor control
PI	Proportional-Integral controller
PLL	Phase Locked Loop
PMSM	Permanent Magnet Synchronous Motor
PWM	Pulse Width Modulation
RAM	Random Access Memory
SDK	Software Development Kit
SVM	Space Vector Modulator
SW	Software
UART	Universal Asynchronous Receiver Transmitter
VToF	Voltage to frequency

2 Introduction**2 Introduction**

The Infineon BASIC SW is capable to drive BLDC motors in combination with the TLE987x device, implementing FOC in sensorless single shunt configuration.

2.1 Content of the Application Note

The BASIC SW, available in the [Infineon Developer Center](#), is a **demo SW** to be evaluated in combination with the TLE987x devices. If the SW is going to be used in a productive environment, a **licensed version** of the BASIC SW is available from Infineon.

Note: for additional information about the licensed version, please visit the [Infineon link](#) or contact your Infineon counterpart

The BASIC SW has been developed according to MISRA-C guidelines. The BASIC SW can also fulfill ASPICE Level 1 requirements: all the related information can be provided by Infineon, together with the licensed version.

This application note will focus on the following topics:

- Description of FOC implementation in the BASIC SW
- Description of the structure of the BASIC SW
- Description of the parameters
- Making a motor spin with the BASIC SW and TLE987x
- Fault types and troubleshooting
- Execution time and memory consumption

2.2 FOC as motor control technique

Field oriented control (FOC) is a method of motor control to generate three sinusoidal phase signals, which can be controlled with frequency and amplitude in order to minimize the phase currents, which in turn means to maximize the efficiency. The basic idea is to transform three phase signals into two rotor-aligned signals and vice versa.

According to the specific implementation, FOC can be realized with or without sensor to detect the rotor position, and with single, double or triple shunt to measure the motor current.

For a general description of FOC in sensorless single shunt configuration, the user can refer to the Infineon Application Note "[Sensorless Field Oriented Control with TLE987x](#)".

The Figure 1 shows a simplified block diagram of FOC, as implemented in the BASIC SW.

Note: the SW mentioned in the Application Note "[Sensorless Field Oriented Control with Embedded Power SoC](#)" is the "BLDC Sensorless FOC (TLE9879 EvalKit)" example SW, available till 2022 in the Embedded Power SDK. This demo SW has been replaced by the BASIC SW.

2 Introduction

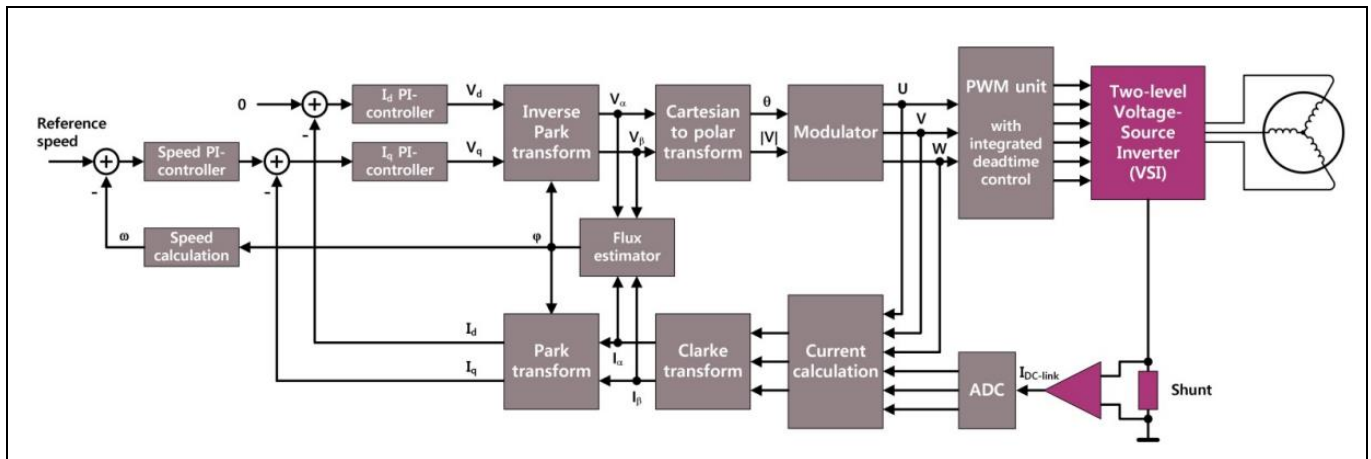


Figure 1 General block diagram of FOC motor control

At start-up, since it is difficult to measure the back electromotive force (BEMF) signal at low speeds, the IToF method (current to frequency open loop) is used. After the speed exceeds the threshold when the BEMF signal can be clearly measured, the flux estimation and the speed closed loop are enabled.

There are two control loops in the system: the speed control loop (known as “slow sloop”) and the current control loop (known as “fast sloop”).

The motor current is sampled by the low side single shunt resistor, using the current sensor amplifier and the ADC integrated in the TLE987x. By measuring the current of two phases, the current of the 3rd phase can be obtained using Kirchhoff’s law. The stator-aligned rotor current I_α , I_β are the output of Clarke transformation, while the rotating current I_d , I_q is derived from the Park transformation.

Being stator currents I_α , I_β and stator voltages V_α , V_β , the motor rotor speed and angle can be calculated through flux estimation. In order to make the speed and angle estimation more stable, SW modules like the angle PLL observer and low pass filter are implemented.

The reference speed is derived from the speed ramp controller. The output of the speed PI controller is the reference for the I_q axis current PI controller, while the reference for the I_d axis current PI controller is 0.

The outputs of the current PI controllers are converted into polar coordinates and fed to the space vector modulator (SVM), which calculates the timings for the pattern generator needed as inputs for the CCU6 module of the TLE987x.

The CCU6 block translates the timings into PWM signals, which are fed into the TLE987x bridge drivers to switch on and off the MOSFETs of the 3-phase inverter.

3 FOC implementation in the BASIC SW

In this section, the FOC implementation in the BASIC SW is described, referring to the simplified flow-chart shown in Figure 2.

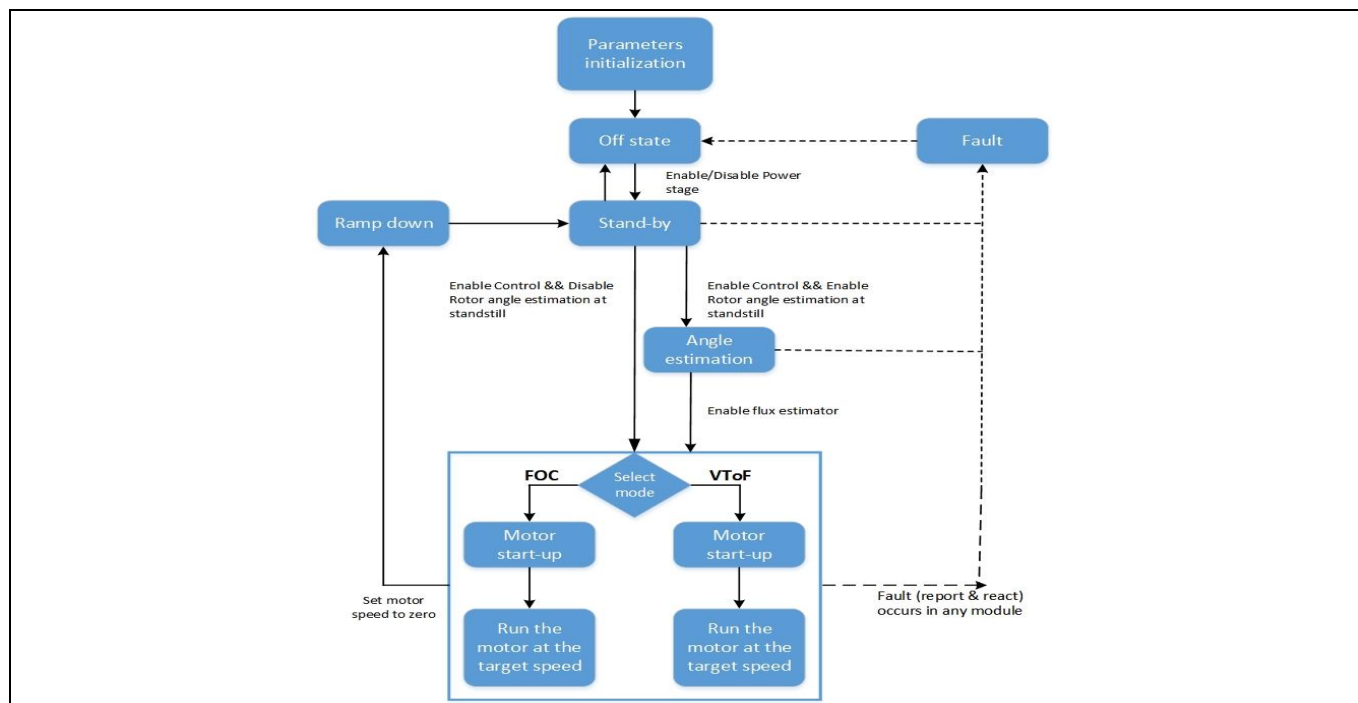


Figure 2 FOC solution – state machine of the speed loop (simplified version)

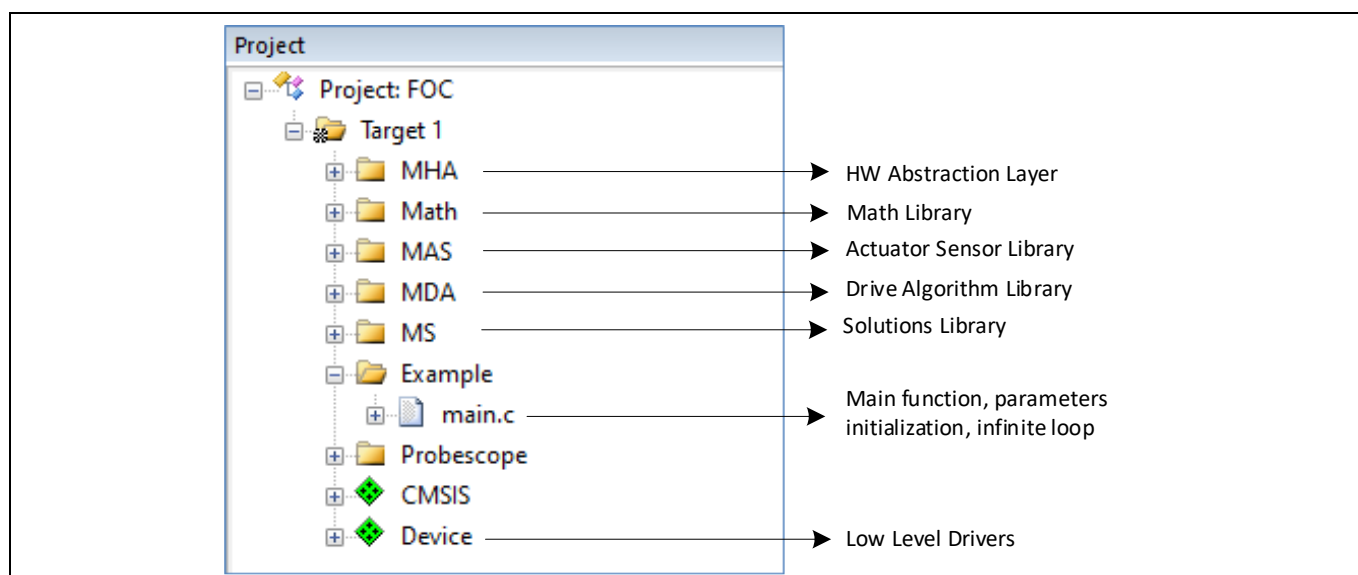


Figure 3 BASIC SW framework in KEIL

3.1 Initialization of the parameters

At the beginning of the SW main.c function (Figure 4), the following initializations are performed:

- The TLE987x modules are initialized by the TLE_Init() instruction
- The TIMER3 of the TLE987x, used to trigger the ADC measurement of the shunt current, starts

Sensorless FOC motor control using Infineon MOTIX™ TLE987x

3 FOC implementation in the BASIC SW

- The Timer T6 of the GPT2, used as timer for the speed loop, starts
- The Timer12, used as timer for the CCU6 block, starts
- The parameters of the “FOC Solution” function, which include the state machines of the SW and speed loop, are initialized

```

161 int main(void)
162 {
163     /** initialization of the hardware modules based on the configuration done **
164     ** by using the IFXConfigWizard **
165     *****/
166
167     /** place your application code here **
168     *****/
169
170     /* Initialize Hardware */
171     TLE_Init();
172
173     /* Start Timer3 high and low byte timer */
174     TIMER3_T3HL_Start();
175
176     /* Start speed loop timer */
177     GPT12E_T6_Start();
178     #if (IFX_MS_FOCsolutionF16_CFG_CURRENT_LOOP_FACTOR > 1)
179
180     /* Set PendSV interrupt priority */
181     NVIC_SetPriority(PendSV_IRQn, 2);
182     #endif
183
184     /* Start CCU6 T12 timer */
185     CCU6_StartTmr_T12();
186     // ProbeScope_Init(10000);
187
188     /* Initialize FOC */
189     Ifx_MS_FocSolutionF16_init(&FocDemoClosedLoop);
190
191

```

Figure 4 Initializations in the main.c function

The configuration of the BASIC SW parameters can be classified in two categories:

- Parameters that can be only configured before code compilation, called “**static-configuration**”. These static parameters, also defined as “constants”, are stored in the data flash of the TLE987x device.
- Parameters that can be tuned during the code execution (runtime), called “**dynamic-configuration**”. These dynamic parameters are physically implemented as variables in RAM. The configuration may be performed either by the application, or by the user by using a specific parameter handler, like a PC-Tool ([Micro Inspector Pro](#), MOTIX™ [Solution Designer](#)) that is able to send and receive back data from the target device.

The user shall configure the static parameters of the BASIC demo SW using the [Config Wizard](#) tool, available in the Infineon Developer Center Launcher.

3.2 Main endless loop

In the main endless loop (Figure 5), the mandatory service function of the TLE987x watchdog (WDT1) is executed. The “FOC Solution” statemachine is then enabled: after aligning the rotor, the motor will start and the selected control mode will be applied. By default, the FOC mode is selected in the BASIC demo SW.

Note: the Start Angle Identification function is disabled by default in the BASIC SW V1.3

3 FOC implementation in the BASIC SW

```

193  /* ***** */
194  ** main endless loop
195  *****
196  for ( ; ;)
197  {
198      /* Service Watchdog */
199      (void)WDT1_Service();
200
201      /* Call clear fault API if requested */
202      if (clrFaultFoc == 1)
203      {
204          Ifx_MS_FocSolutionFl6_clearFault(&FocDemoClosedLoop);
205          clrFaultFoc = 0;
206      }
207
208      /* Call FOC enable */
209      Ifx_MS_FocSolutionFl6_enablePowerStage(&FocDemoClosedLoop, (bool)enablePowerStage);
210      Ifx_MS_FocSolutionFl6_enableControl(&FocDemoClosedLoop, (bool)enableControl);
211      Ifx_MS_FocSolutionFl6_setControlMode(&FocDemoClosedLoop, (Ifx_MS_FocSolutionFl6_ControlMode)controlMode);
212      Ifx_MS_FocSolutionFl6_enableDirectInterface(&FocDemoClosedLoop, (bool)enableDirectInterface);
213
214      /* Call Bridge Driver (fault handling) */
215      Ifx_MHA_BridgeDrv_TLE987_execute(&FocDemoClosedLoop.bridgeDrvTLE987);
216      Ifx_MS_FocSolutionFl6_enableSpeedPreControl(&FocDemoClosedLoop, (bool)enableSpeedPreControl);
217
218      if (enableStartAngleIdent == 1)
219      {
220          Ifx_MS_FocSolutionFl6_enableStartAngleIdent(&FocDemoClosedLoop);
221          // enableControl = 1;
222          enableStartAngleIdent = 0;
223      }

```

Figure 5 Main endless loop

3.2.1 Start Angle Identification

Note: The Start Angle Identification functions is disabled by default in the BASIC SW V1.3.0. A detailed description of this module is available in the BASIC SW User Manual, available with the licensed version.

In order to prevent unwanted rotation in the opposite intended direction during start-up, it is important to know the initial position of the rotor to impose a starting voltage vector accordingly. However, with the absence of a position sensor there is no direct feedback about the initial rotor angle. In addition to that, model-based speed and position estimators only work starting from a high enough speed level, thus other methods need to be used to estimate the initial rotor position.

Permanent magnet synchronous motors (PMSM) have a structural saliency, since the flux path is different in the d axis compared to the q axis (i.e. the inductance L_d and L_q are not equal, $L_d < L_q$). This characteristic can be used to detect the position of the rotor by putting a pulse of constant voltage amplitude for a very short time and observing the change of current.

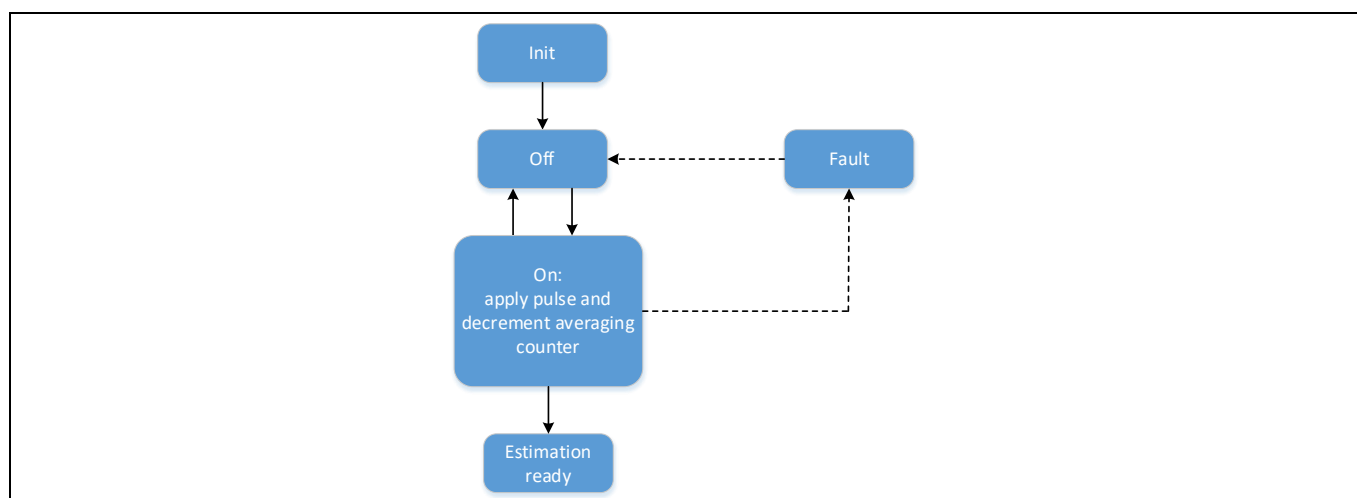


Figure 6 Start Angle Identification state machine (simplified)

3 FOC implementation in the BASIC SW

Ideally it is possible to excite the stator windings with a voltage vector with a small amplitude and to rotate at a high frequency. Due to the rotor-load inertia, the rotor then does not rotate. By measuring all three phase currents at this moment it is possible to estimate the position of the rotor.

3.3 Rotor alignment

The rotor alignment is implemented in the main function with open loop control. The alignment is done by setting the reference speed to zero for some fixed steps, so that the currents will be applied to the motor continuously as the control is enabled. During these fixed steps, the flux is generated and the magnetic fields will get aligned.

3.4 FOC mode

FOC requires feedbacks on rotor position and rotor speed. These can come from a sensorless mechanism or from dedicated sensors: the sensorless FOC derives the rotor position and rotor speed based on motor modeling, the voltage applied to the motor phases and the current of the three motor phases. The feedback of the phase currents can be sensed with a shunt at the low-side MOSFET.

The Figure 7 shows the simplified block diagram for FOC, where the current sensing with single shunt at low side is implemented.

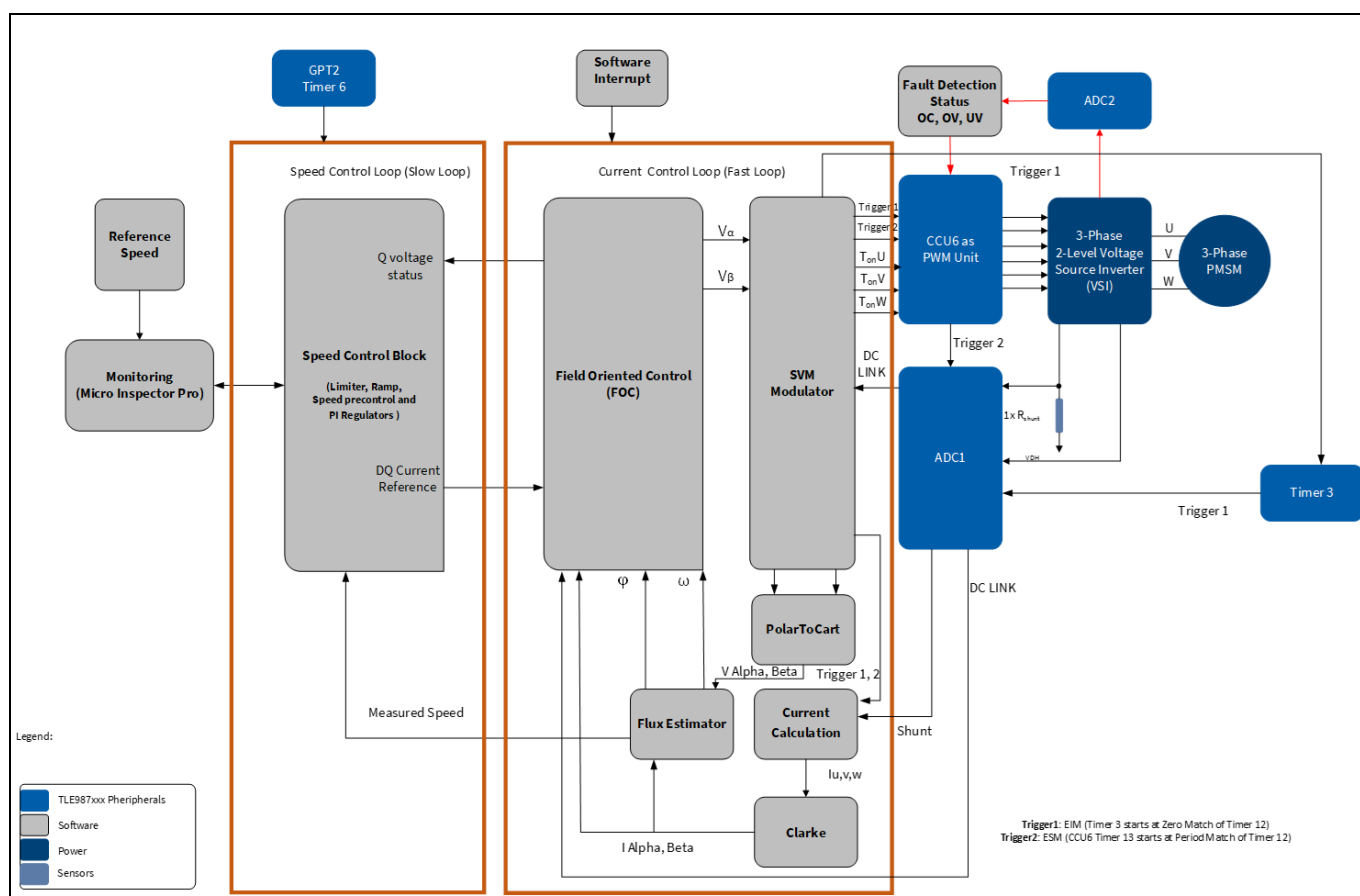


Figure 7 Block diagram of PMSM FOC motor control

From start-up until the motor reaches the reference speed, two control stages can be identified (Figure 8):

- **Open loop control:** it is active from start-up (i.e. at zero speed) until a specific limit (transition speed) is reached. Vice versa, it is enabled if the speed drops below the specific limit (Transition speed down). “Open loop” means that the control does not consider neither the motor speed estimated by the flux estimator as

3 FOC implementation in the BASIC SW

feedback, nor the estimated angle for the Park transformation. The reason why open loop control is used at start-up, is that closed-loop methods do not work reliably at low speeds

- **Closed loop control:** it is active from the moment when the transition speed is reached until the speed drops below the lower limit of the transition speed. "Closed loop" means that the control considers the motor speed estimated by the flux estimator as feedback to calculate the reference I_q current, and then uses it for estimating the angle in the Park transformation.

Both control stages of the FOC mode are called “sensorless”, since they work without a feedback from sensor to detect the motor angle.

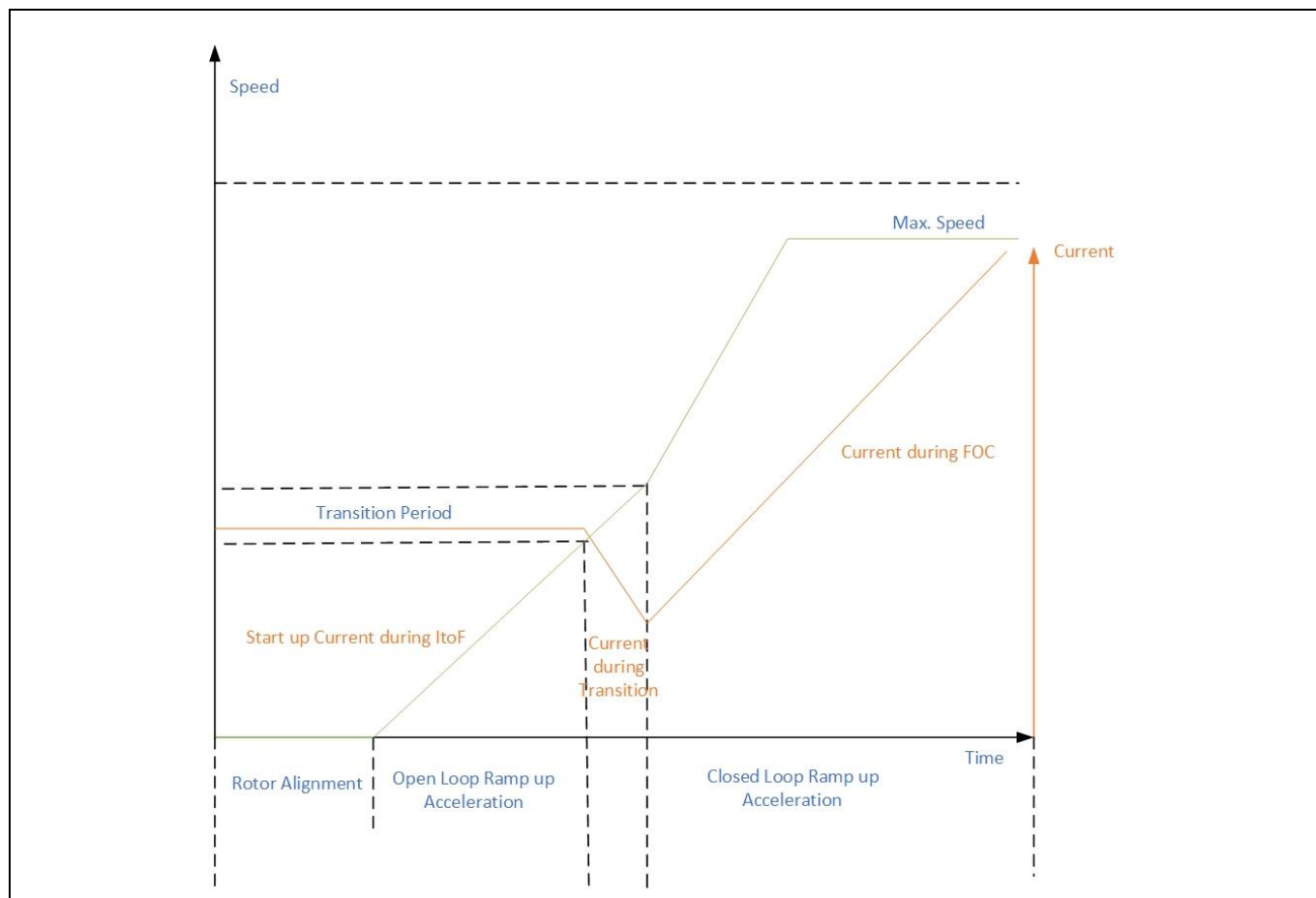


Figure 8 Start-up sequence – smooth transition

3.4.1 Open loop during start-up

The IToF (current to frequency) method is used as open loop control technique to turn the motor at start-up and to run it at lower speeds. The IToF control advantages are the following:

- The IToF uses the same current controller with PI regulators as FOC, so the transition from IToF to FOC is easier compared to other methods
- The IToF method is better suited under load conditions because of its torque regulation compared to VToF

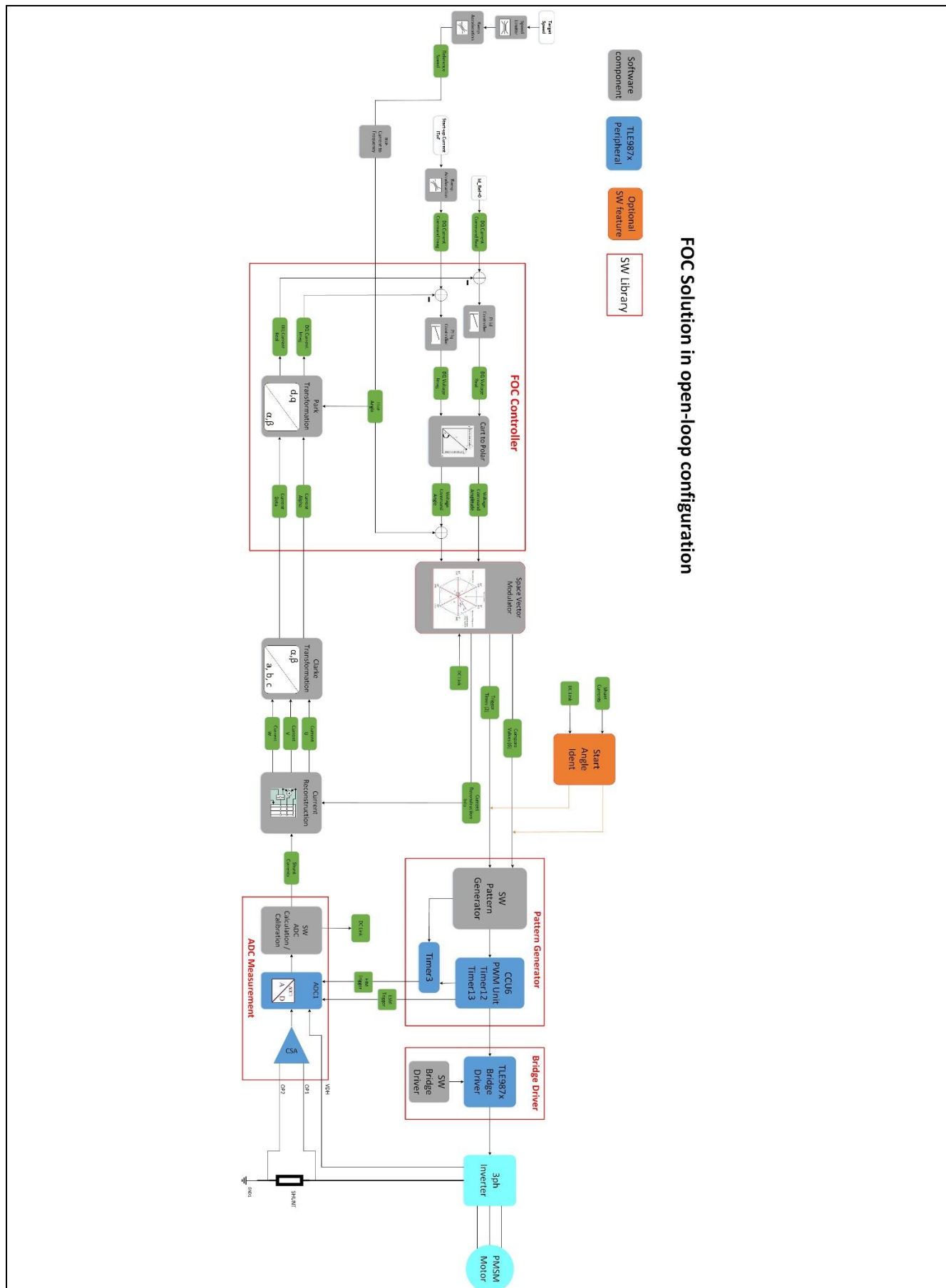


Figure 9 Block diagram for FOC in open loop configuration

3 FOC implementation in the BASIC SW

Working principle

The amplitude of the start-up current has to be configured by the user to start the motor initially (Start-up Current I_{ToF} in Figure 9). The start-up current is used as a constant electro-mechanical torque reference to accelerate the motor. The value, to be adjusted according to the load, shall be big enough to start the motor.

In fact, when turning a motor for the first time with the BASIC SW, the motor might not start spinning as expected. One of the reasons could be the value of the start-up current, which has to be adequately increased.

The start-up current ramp-up rate is used for smooth acceleration during start-up. Without the ramp-up rate, a constant start-up current can also be used. Depending upon the reference speed, the IToF algorithm calculates the related angle used by the Park transformation and the modulator.

3.4.2 Transitions from open to closed and from closed to open loop

The transition from open to closed loop (and vice versa) is not trivial, because the angle generated by the IToF open loop and the real rotor position are not aligned (Figure 10). The reason is that in open loop the I_q reference current is the start-up current, which is usually different from the closed loop FOC reference current required to accelerate then the load.

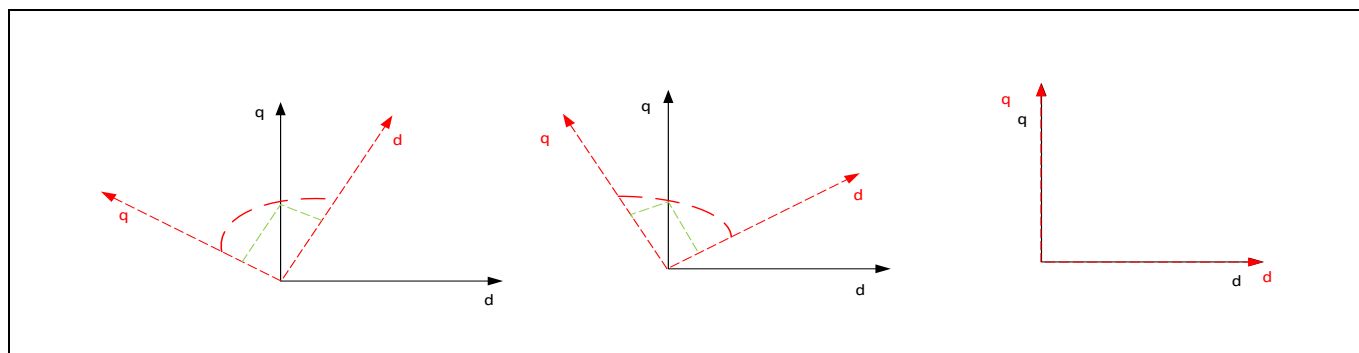


Figure 10 Transition angle from open (red) to closed loop (black)

Transition from open loop to closed loop, also called “transition up”

The IToF open loop will be executed until the limit of transition speed (set in the Config Wizard - MCTRL solution) is reached. Then the mode will be switched to closed loop.

Transition from closed loop to open loop, also called “transition down”

The closed loop will be executed until the limit of transition speed (set in the Config Wizard - MCTRL solution) is reached. Then the mode will be switched to open loop.

Smooth transition and direct transition

Two types of transitions can be selected (Figure 11):

- smooth transition (soft switching): the transition is done following a ramp during a configurable period of time
- direct transition (hard switching): the transition is done instantly

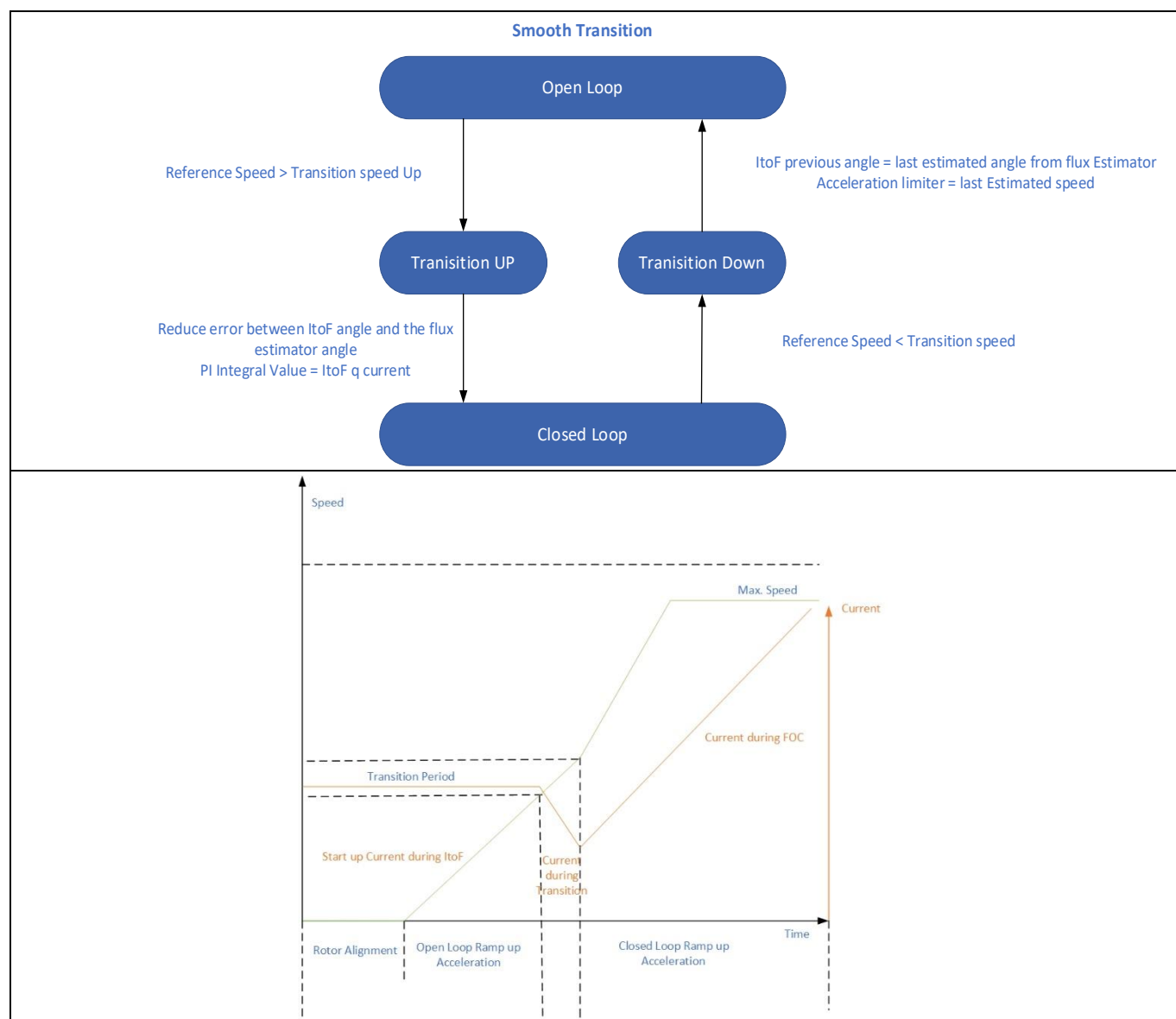


Figure 11 Smooth transition flow

3 FOC implementation in the BASIC SW

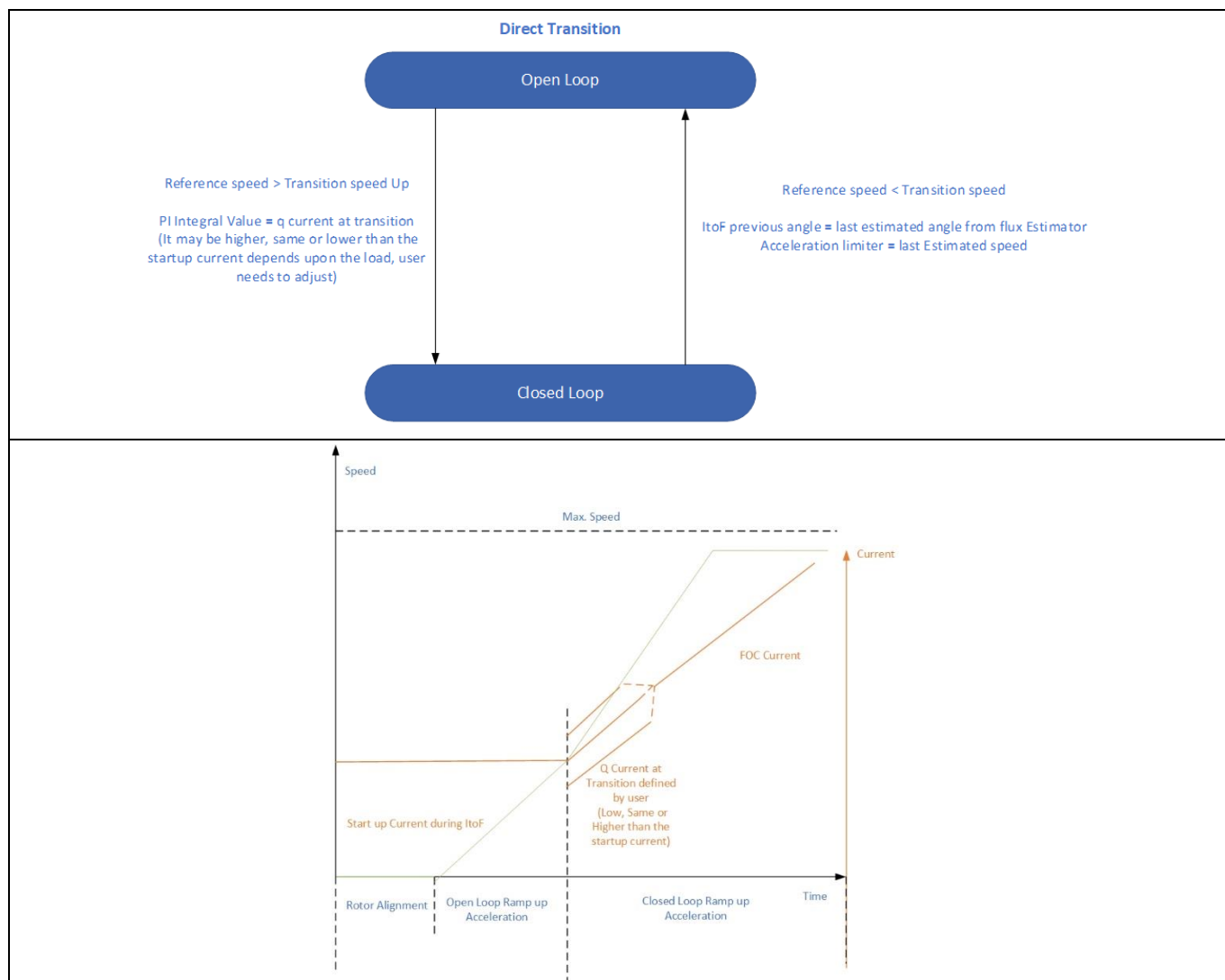


Figure 12 Direct transition flow

3.4.3 Closed loop control

Field oriented control is all about controlling the stator currents represented by a vector. This control depends upon the projections of a 3-phase time and speed dependent system converted into a two-coordinate (d and q frame) time invariant system. These projections and transformations result in a structure that is similar to that of a DC machine control. FOC controller require two constants as input references: the torque component (aligned with the q coordinate) and the flux component (aligned with d coordinate).

The speed control scheme, shown in Figure 7, represents a closed loop configuration. This scheme uses cascaded speed and current control structures, where the dynamic requirement for the speed control loop is lower compared to the current loop. In fact, the speed control loop is called **slow loop** and the current control loop is called **fast loop**. The time to execute the fast loop and the slow loop can be configured in the Config Wizard – Application section.

The Figure 13 shows a simplified representation of the FOC in closed loop configuration, as implemented in the Infineon BASIC SW.

3 FOC implementation in the BASIC SW

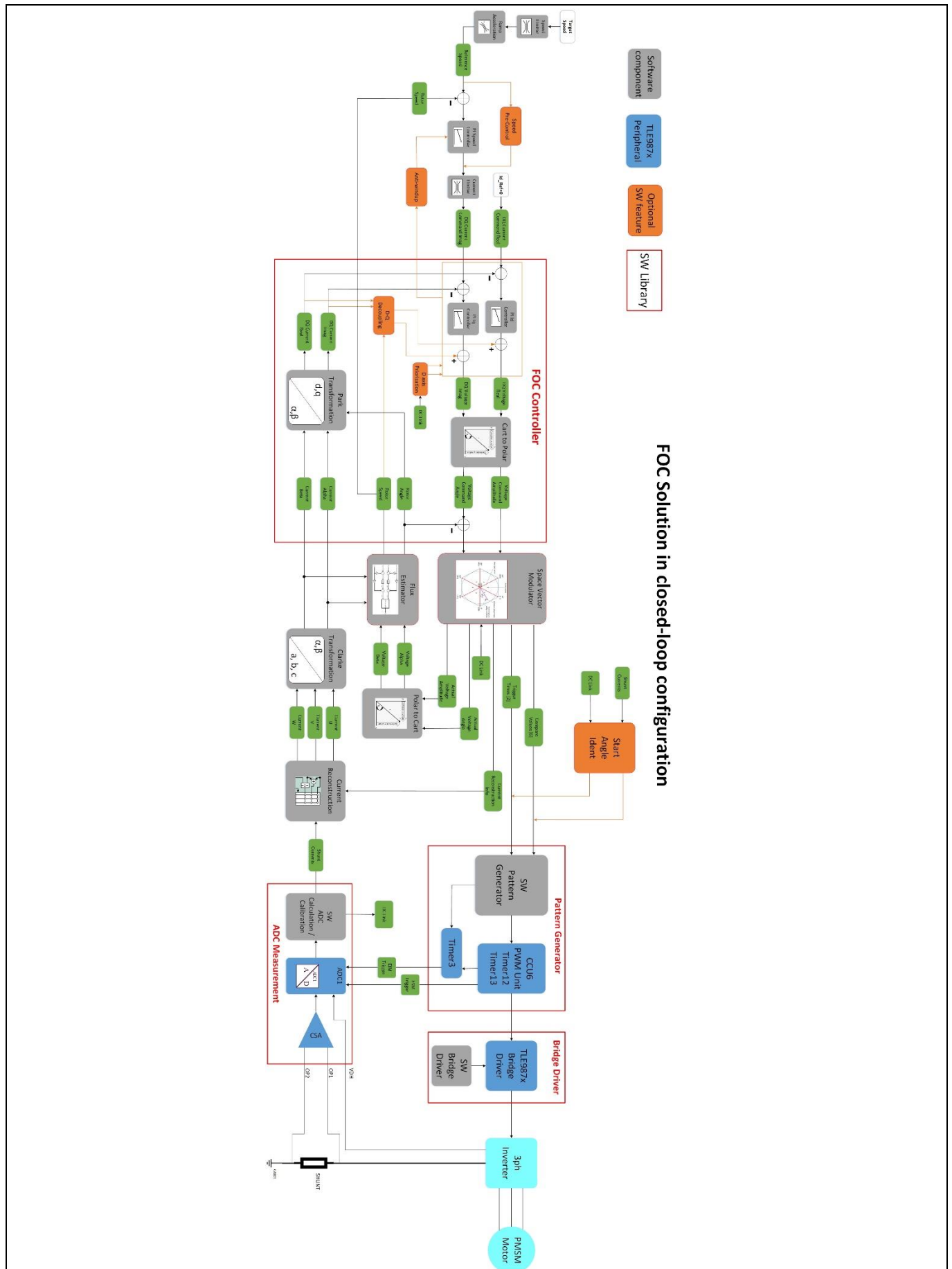


Figure 13 **Block diagram for FOC with closed loop control**

3 FOC implementation in the BASIC SW

The working principle of FOC in closed loop control is as follows.

1. The desired motor speed in steady state is defined as **target speed** (Figure 14Figure 13): this is used as input for the **ramp generator**, which is managing the transition from zero to nominal speed during the start-up phase. The output of the ramp generator is the reference of the speed controller.
2. The difference between the reference speed and the estimated rotor (motor) speed, calculated by the flux estimator, is the input of the **speed PI controller**. As SW options, speed pre-control and anti-windup can be enabled.
3. The speed proportional-integral (PI) controller calculates the motor currents (direct and quadrature, I_d and I_q) used as reference for the current loop. The PI speed loop compensator acts on the error between the reference speed and the current speed: the integral term forces the steady state error to zero, while the proportional term improves the high frequency response. The PI compensator gains can be adjusted depending on the motor and load characteristics to meet the target dynamic performance.

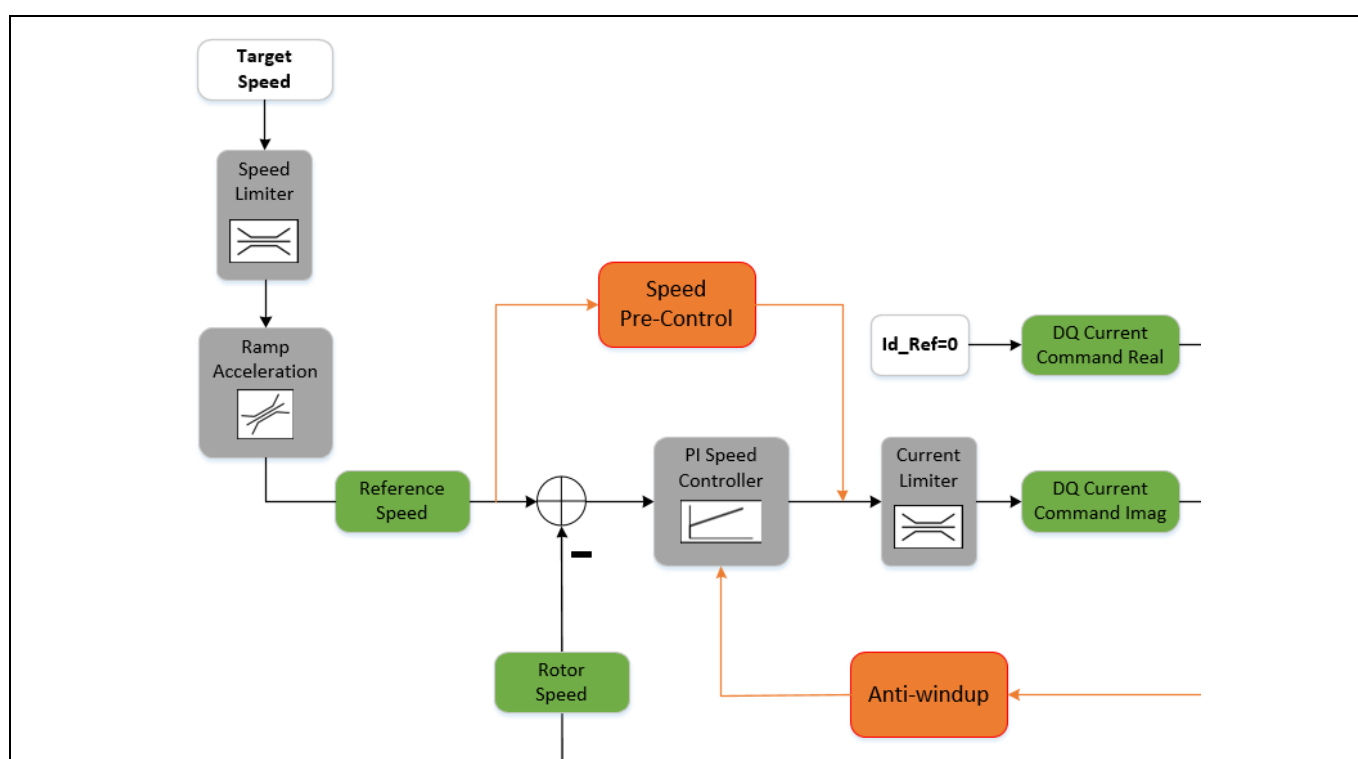


Figure 14 Target speed and PI speed controller

4. The current loop in the FOC controller block (Figure 15) calculates the inverter voltages to drive the motor currents needed to generate the desired torque. Two separate current PI controllers (for I_d and I_q) have as inputs the difference of the **I_d and I_q reference currents** (DQ Current Commands from the speed controller) and the **I_d and I_q motor currents** (DQ Currents calculated with Clarke and Park transformations from the phase currents measured on the shunt). The FOC uses the Clarke and Park transformations to convert the motor phase currents into two quasi DC components: An I_d component that increases or weakens the rotor field, and an I_q component that generates motor torque.
5. The outputs of the current PI controllers are the output voltages V_d and V_q , used as input for the Cartesian to polar function, which convert them as voltage reference amplitude and reference angle for the space vector modulator (SVM) in Figure 15.

3 FOC implementation in the BASIC SW

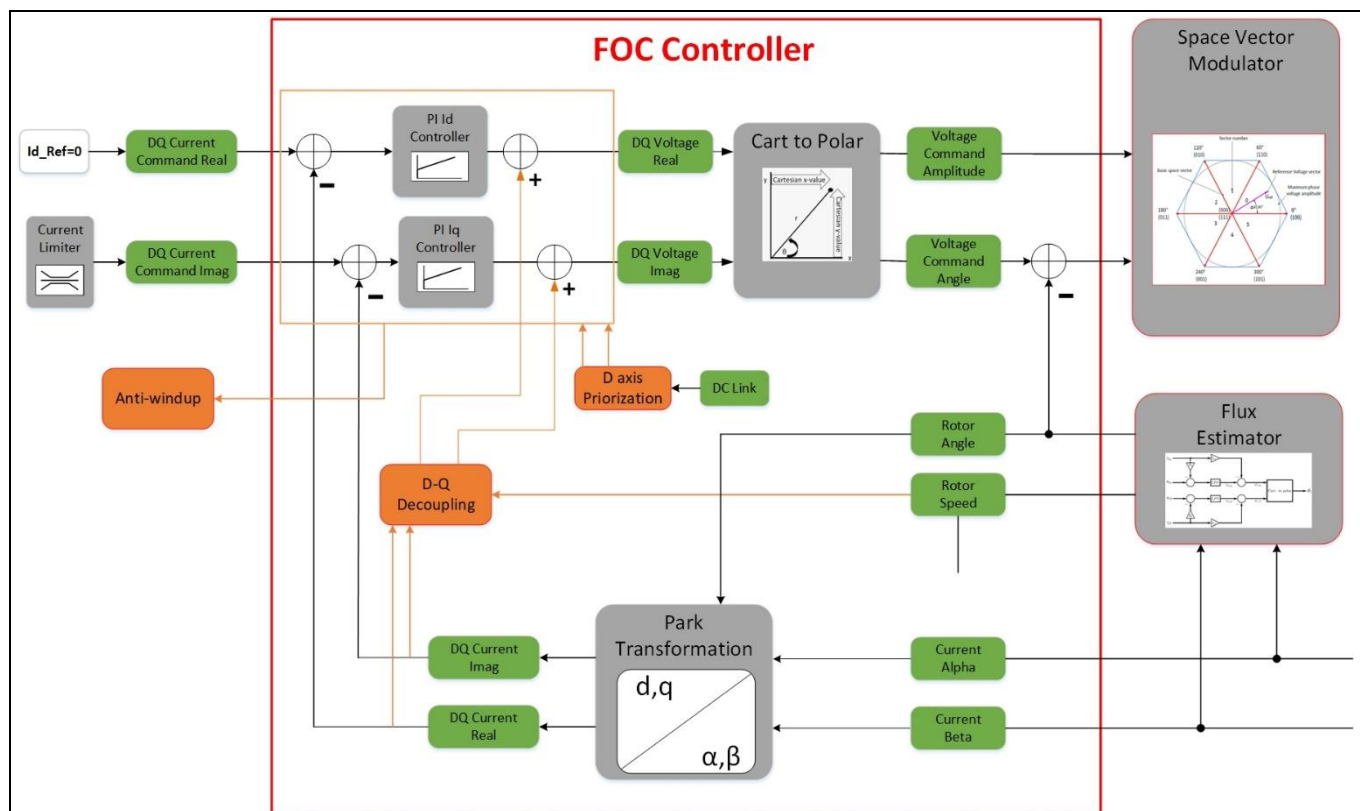


Figure 15 FOC Controller

6. The SVM block (Figure 16) calculates the opening times of the switches for the voltage source inverter that is connected to the stator of a three-phase BLDC motor. The compare values and trigger times (output signals of the SVM block) are fed into the pattern generator. The current reconstruction information is used by the homonymous module in order to obtain the phase currents of the motor, calculated by the ADC measurement block. The currents are then converted into alpha/beta domain by the Clarke transformation.
7. The flux estimator (Figure 16) calculates the position and the speed of the rotor, taking as inputs the angle and amplitude voltage from the SVM (after a polar to Cartesian conversion) and the alpha/beta currents from the Clarke transformation.

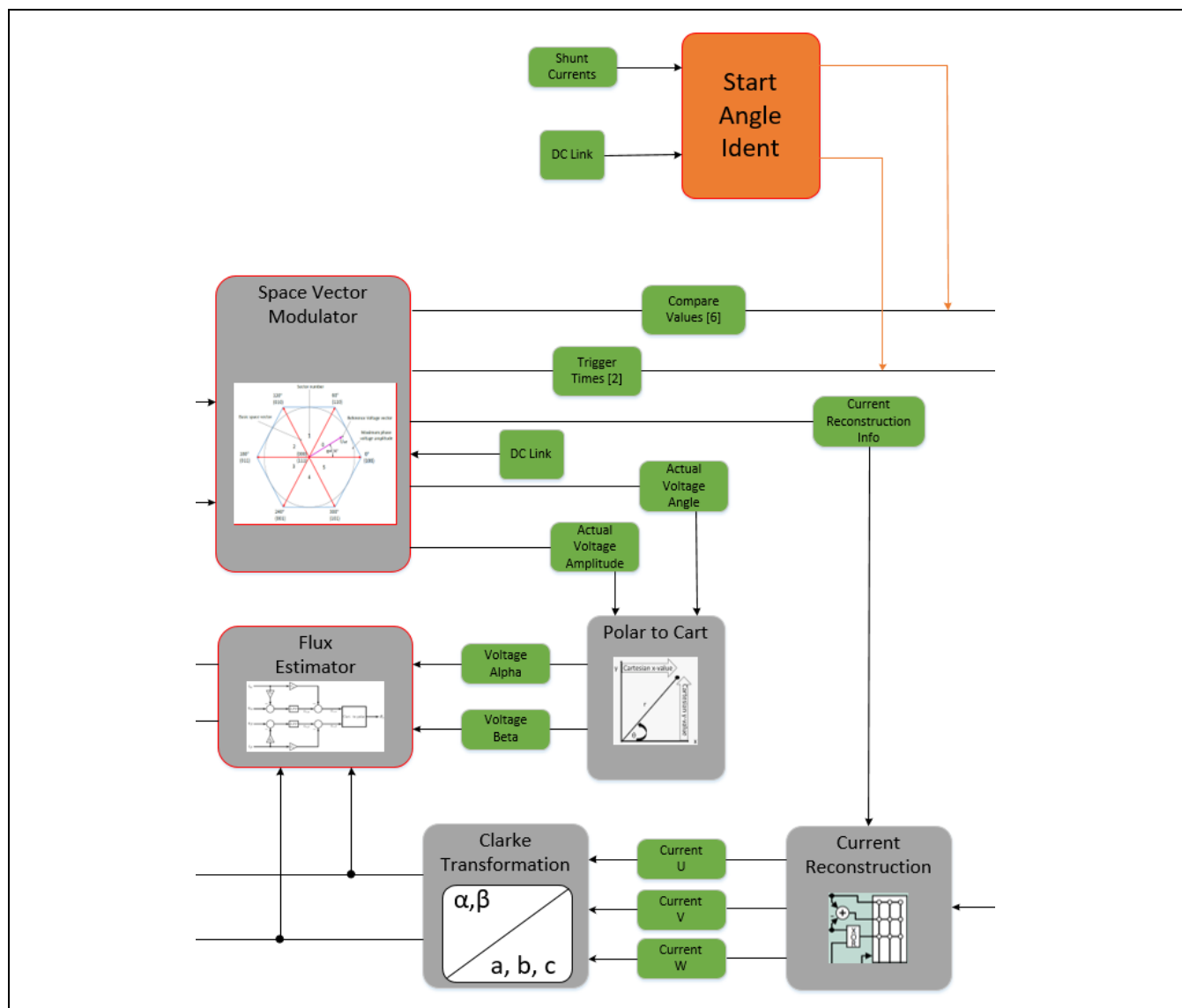


Figure 16 SVM and Flux Estimator blocks

8. The pattern generator block (Figure 17) provides the PWM signals to drive the 3ph inverter, based on the reference voltage and angle inputs from the SVM module. The CCU6 module of the TLE987x generates the PWM signals in center-aligned mode, using the timers T12 and T13. The Timer3 module is used to generate the EIM (exceptional interrupt measurement) event for the ADC measurement.
9. In the ADC measurement block (Figure 17), the CSA (current sense amplifier) of the TLE987x is used to sense the current of the motor, fed into the TLE987x ADC1 converter that takes also care of measuring the motor supply voltage (VDH). The ADC measurement is triggered by EIM and ESM (exceptional sequence measurement) events, generated by the CCU6 module and Timer3. The SW of the ADC measurement is used to calibrate the current offset during the initialization phase.
10. Finally, the bridge driver integrated in the TLE987x (Figure 17) drives the MOSFETS of the 3-phase inverter, using the PWM signals from the CCU6 module. The bridge driver SW manages the fault conditions detected by the hardware.

3 FOC implementation in the BASIC SW

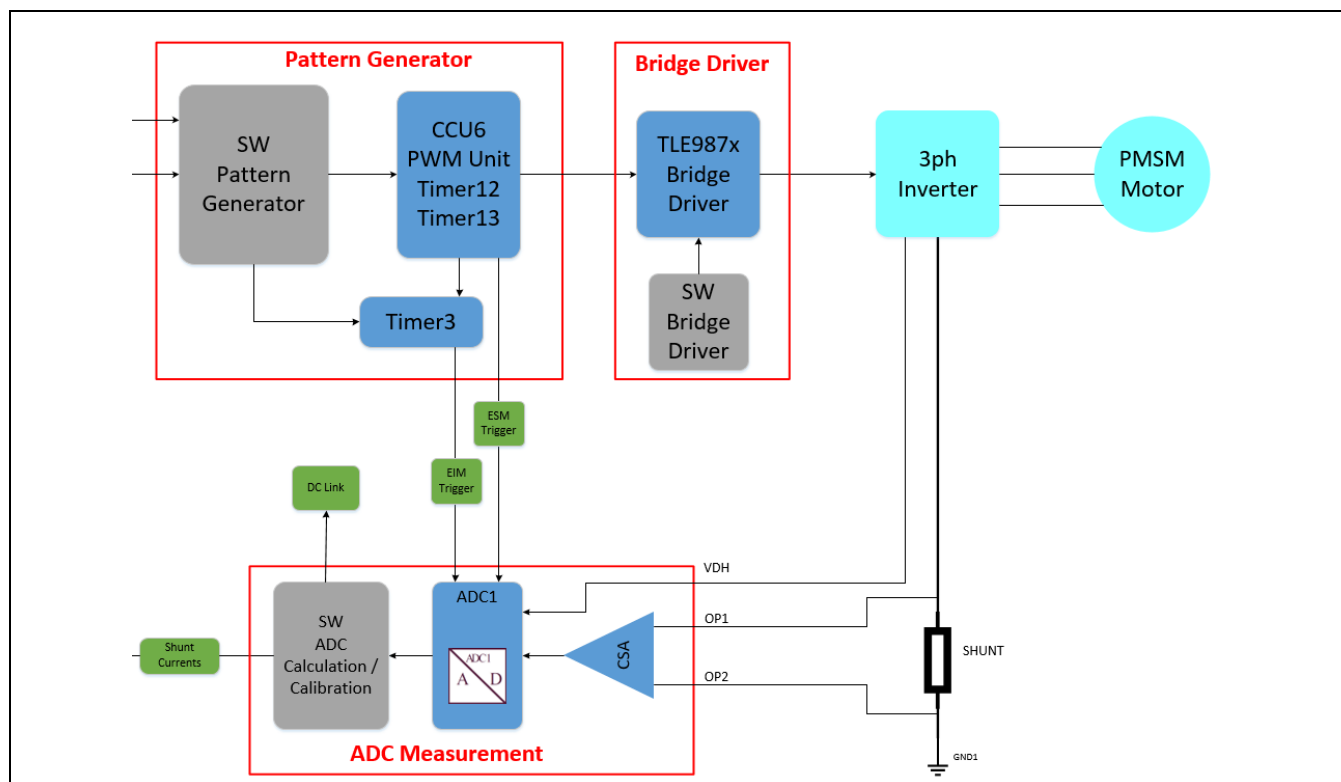


Figure 17 **Pattern generator, Bridge Driver and ADC measurement blocks**

3.5 VToF mode

In the VToF (voltage to frequency) control mode, the rotational voltage command in alpha/beta coordinate system is generated based on the input reference speed command (Figure 18). The characteristics of the motor are configured in the voltage to speed look-up-table. If the dead-time compensation feature is enabled, the current measurement is still used for dead-time compensation, but it is not used as feedback. In VToF control mode it is not possible to switch to closed-loop operation.

The parameters of the VToF scalar control structure are tuned based on the motor parameters (rated voltage, rated speed) and the mechanical characteristics of the load (load torque, load inertia, range of variation) in the specific application.

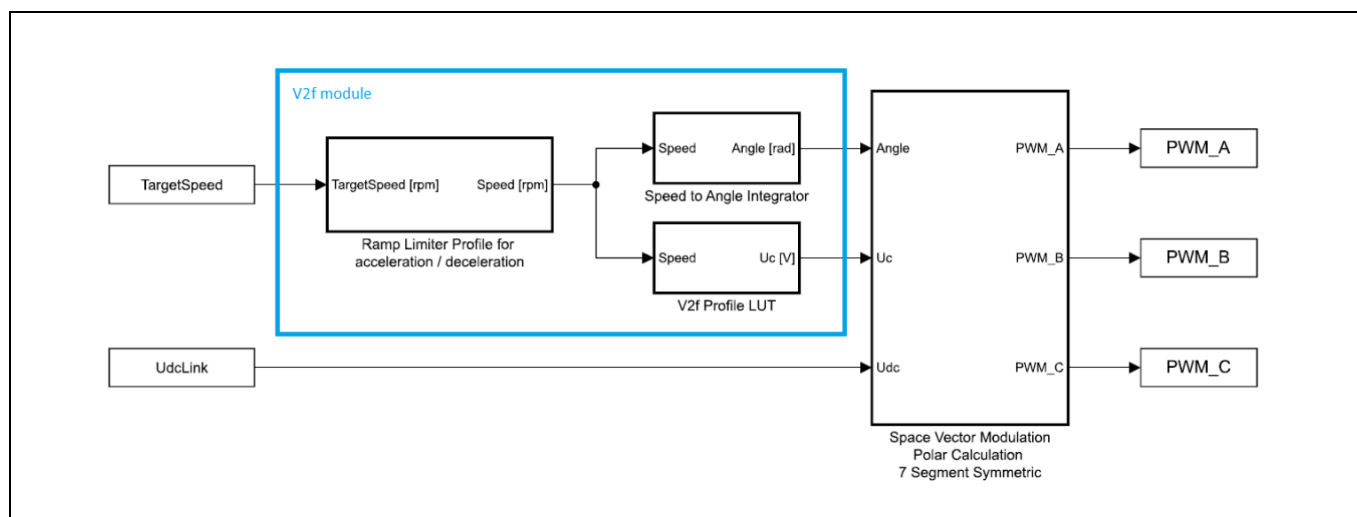


Figure 18 **Block diagram of VToF open loop control (simplified)**

3 FOC implementation in the BASIC SW

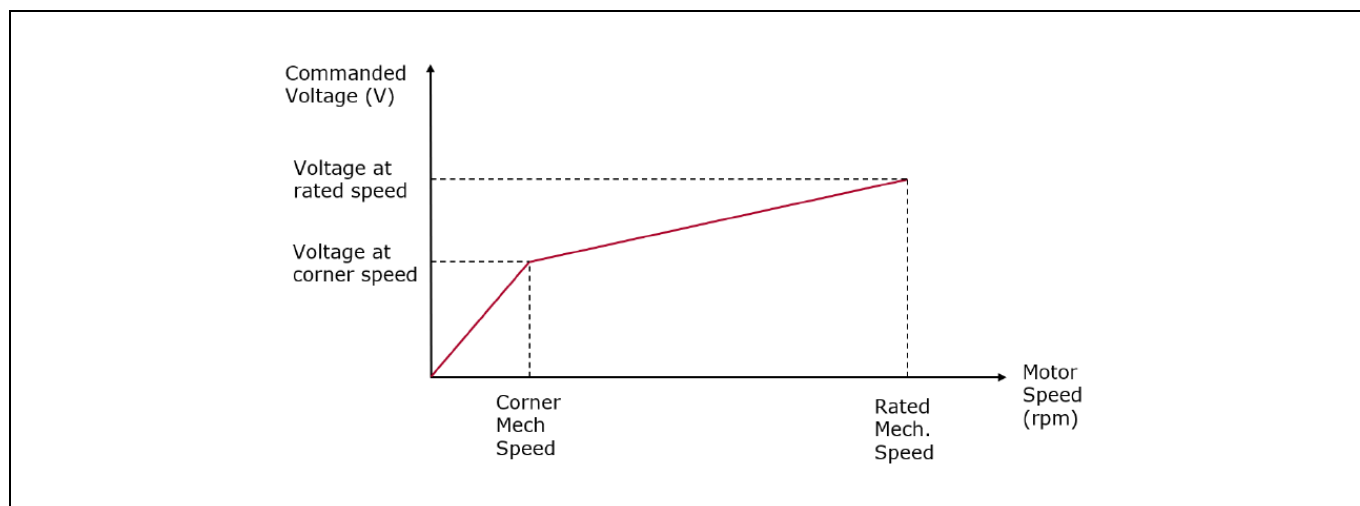


Figure 19 VToF configuration example

Such open loop control structure is only recommended in applications where the load characteristics are stable, such as for fans and pumps. In applications where the load torque at constant speed changes dynamically, this control structure could become unstable.

3.6 TLE987x features used in the BASIC SW

This section describes more in detail some of the TLE987x features used in the BASIC SW.

3.6.1 PWM generation

The CCU6 timer unit of the TLE987x has been designed to generate the PWM signal by using the timers T12 and T13.

T12 is used to generate the signals for the space vector modulation (SVM), which is why it has three different compare values set on each single timer period. The timer period is setup according to the PWM frequency, whereas the three compare values define the on-time for the three phases.

Additionally, the CCU6 can automatically add a dead time. Depending on the MOSFET type used in the half-bridges, the switch-on and switch-off times will change. In the worst case, the switch-on time is smaller than the time for switch-off and a short circuit in the inverter leg might occur. In order to avoid such cross conduction in the motor bridge, the capture/compare unit contains a programmable dead-time generation block, which can automatically add a delay between the switch-off on one MOSFET and the switch-on of its complementary one.

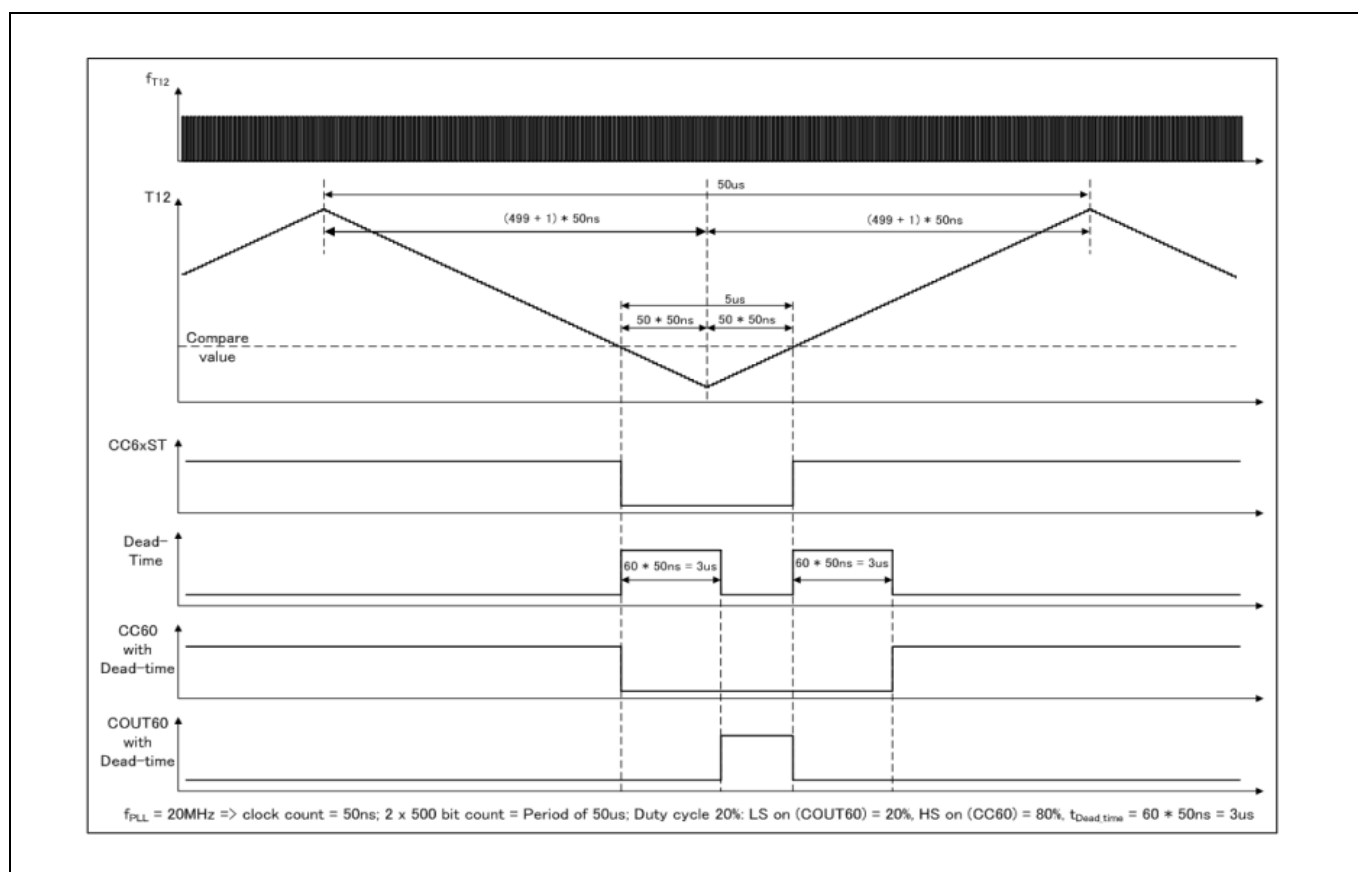


Figure 20 Example of PWM generation in center aligned mode

3.6.2 ADC

Single shunt current measurement is used to measure the currents of the motor phases. The ADC module is used to convert the voltage measurement from the shunt resistance to a digital value that can be used by the software.

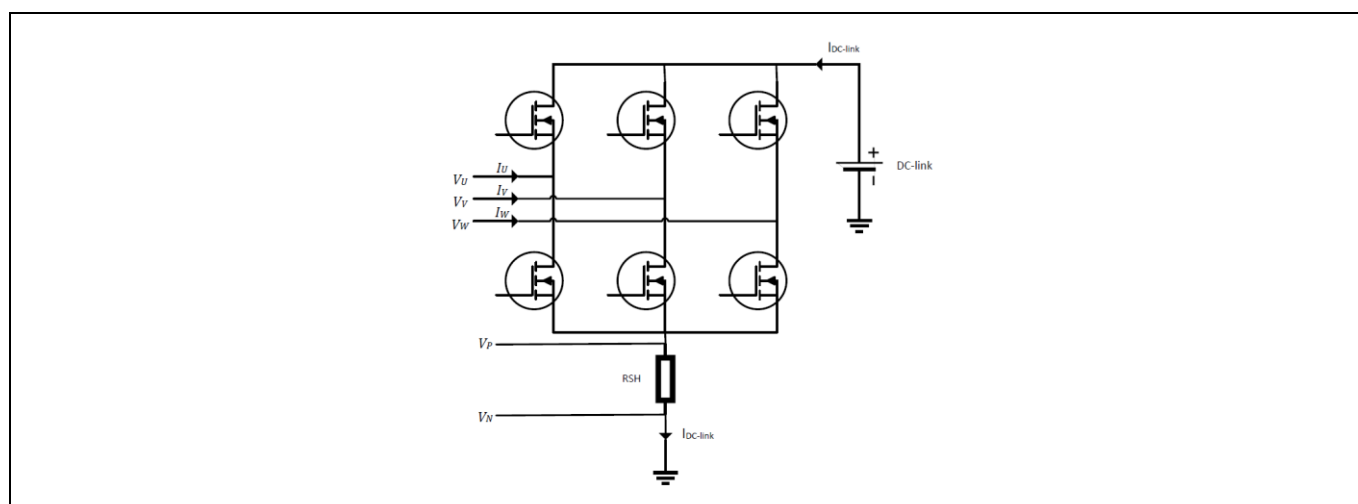


Figure 21 Single shunt current measurement

Using the single shunt approach, the measurement is performed twice during each PWM cycle, at points where different phases are active, relying on the SW knowledge of the implemented phase modulation pattern. These

3 FOC implementation in the BASIC SW

two measurements allow the calculation of the current of two phases, where the third one can be derived by the equation $i_U + i_V + i_W = 0$.

The measured currents are used in the regulation according to the current loop period defined in the Config Wizard.

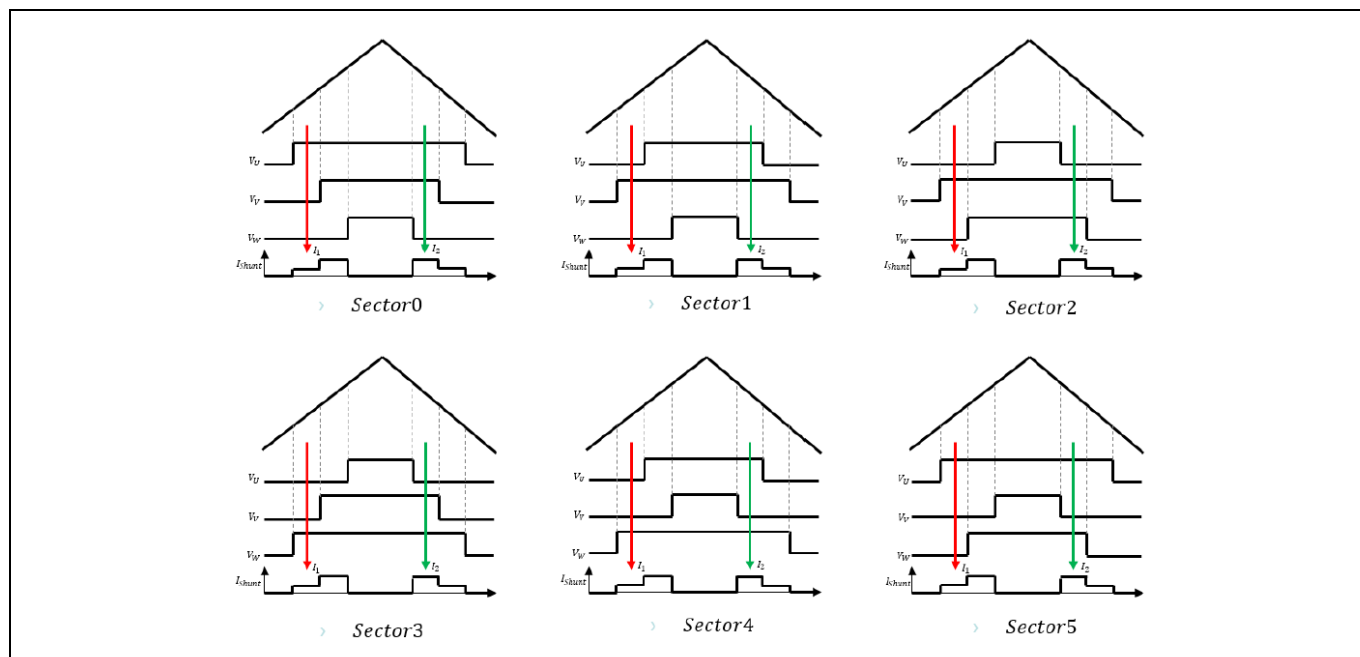


Figure 22 Current sampling position according to sector number

3.6.2.1 ADC triggering with PWM

Timer3 can function as a timer or a counter. When working as a timer, Timer3 is incremented in periods based on the MI_CLK or LP_CLK clocks (as a reference: [1]). When working as a counter, Timer3 is incremented in response to a 1-to-0 transition (falling edge) at its configured input. Timer3 can be configured in four different operating modes for a variety of applications. Out of that one mode is used for triggering the measuring unit upon PWM/CCU6 unit. During up counting, Timer3 starts at zero match of Timer12, used as a trigger 1 to ADC (EIM).

T13 is used in combination with T12 timing to generate the trigger events for the ADC. T13 runs in single shot mode and starts automatically on one defined compare match of one of the 3 compare channels. The generated trigger starts the conversion of the ADC1 channel which generates an interrupt after the channel is converted. During down counting, Timer13 starts at period match of Timer12, used as a trigger 2 to ADC (ESM, as a reference: [1]).

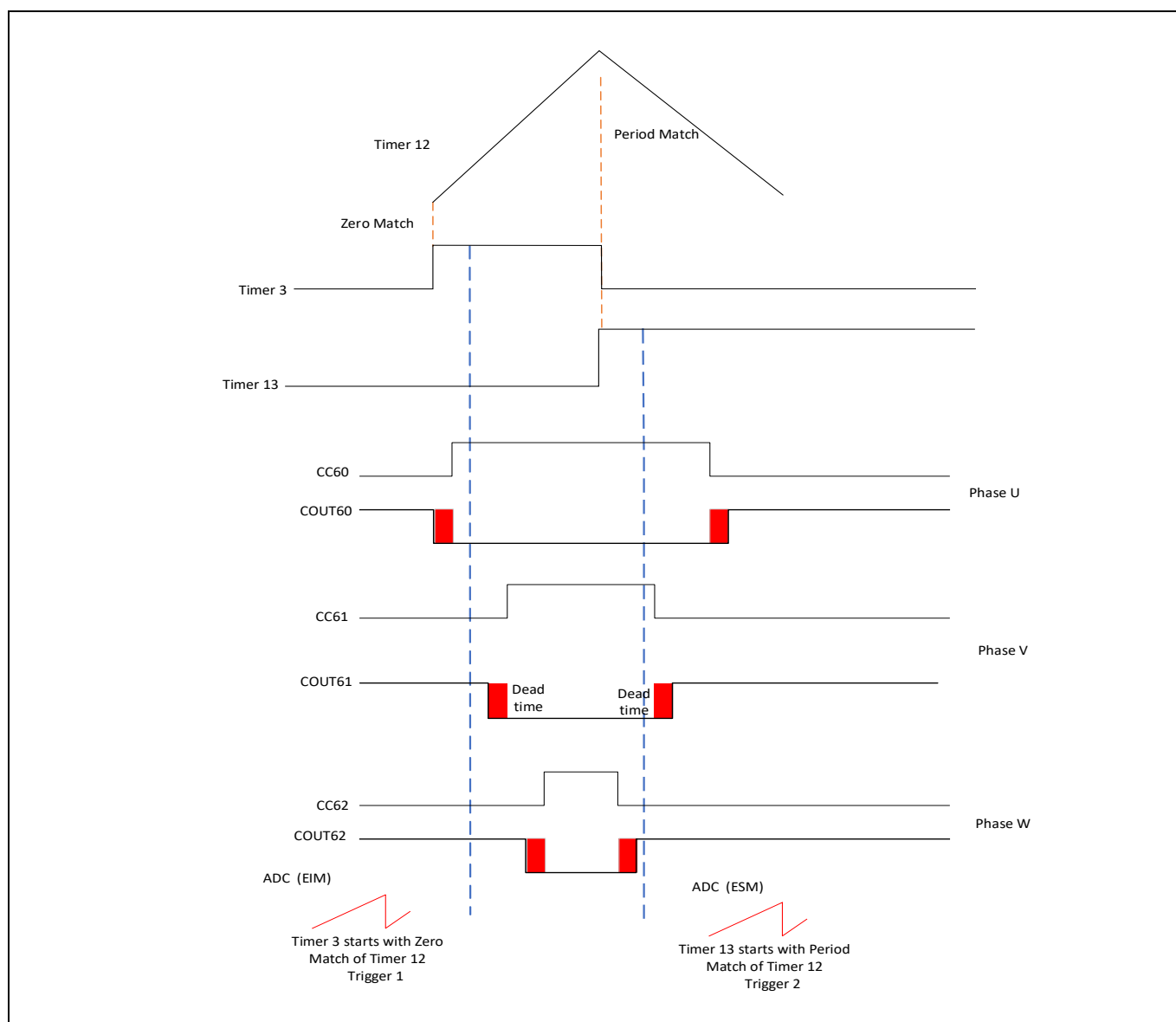


Figure 23 ADC triggering with PWM

Situation 1

- Single phase is active at trigger 1 (for example here: U)
- Both phases are active at trigger 2, one phase is the same phase of trigger 1 (for example here: U and V)

Note: In corner cases, there is not enough time to perform the ADC measurements (minimum sense time) between the pulses, then it may lead to shift the pulse and trigger.

Situation 2

- Single phase is active at trigger 1
- Single phase is active at trigger 2, which is a different compared to the phase of trigger 1

3.6.2.2 ADC read with PWM

Two interrupts of T12 are used to execute the function loops to read the currents from ADC (Figure 24). The two function names are placed in the interrupt call back section of T12 period-match and T12 one-match (Figure 25).

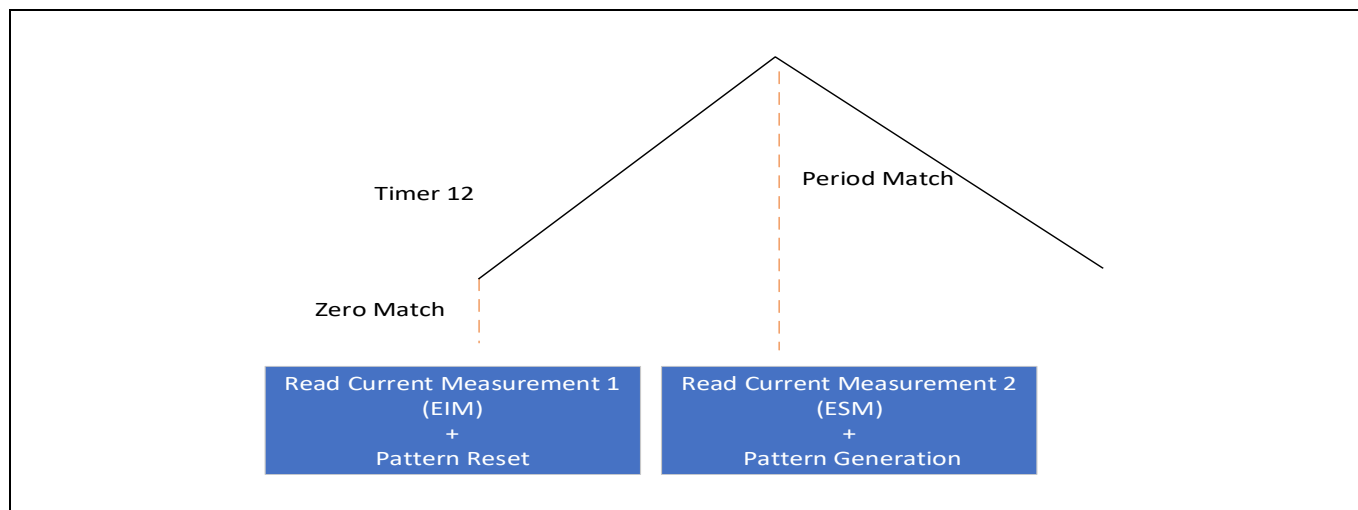


Figure 24 ADC read interrupt from PWM

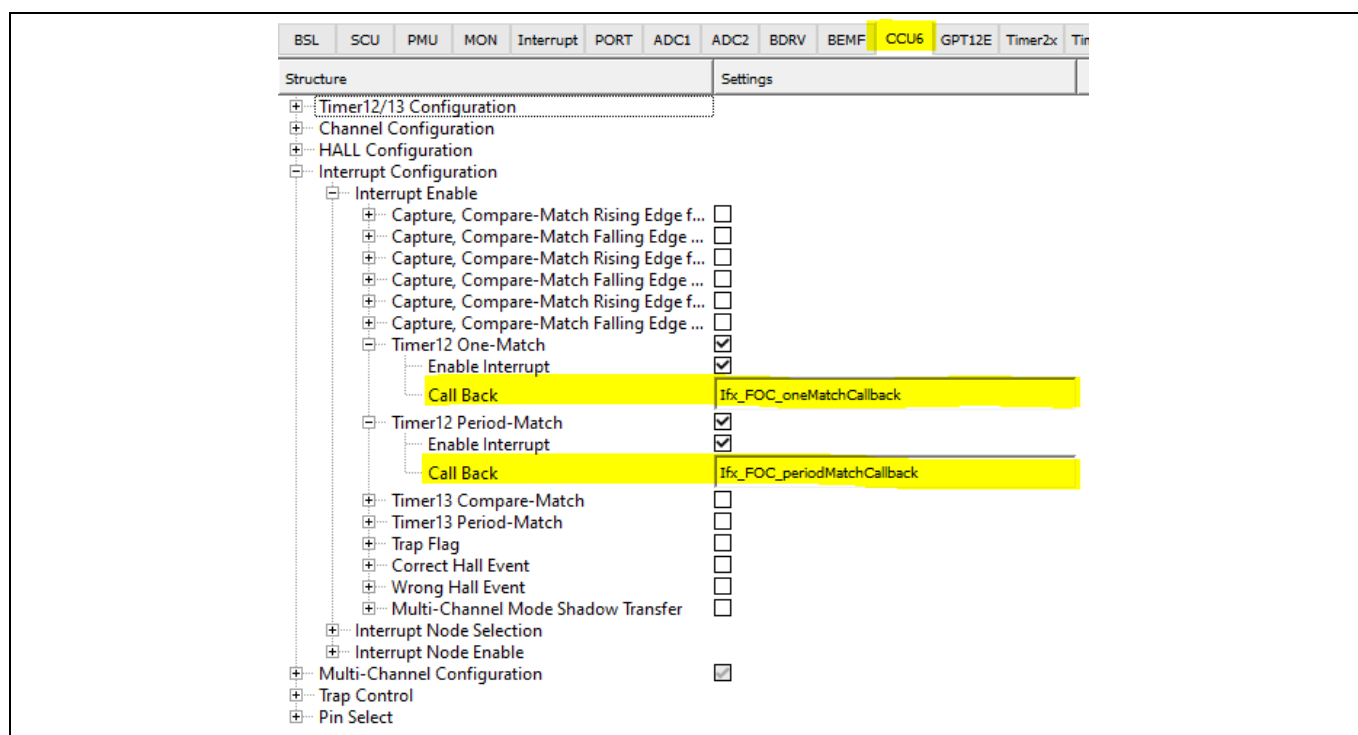


Figure 25 ADC read interrupt from PWM in the Config Wizard

3.6.3 Bridge Driver

The MOSFET driver is designed to drive N-MOSFETs. The Bridge driver typically has to operate at 12V, however the voltage range at the supply can vary greatly. The use of the charge pump enables to drive N-MOSFETS with a duty cycle range from 0% to 100%. The regulated output voltage is typically $V_s + 14\text{ V}$ (V_s is the DC link

3 FOC implementation in the BASIC SW

voltage), but it can be set to $V_s + 9\text{ V}$ if necessary. The charge pump output VCP is monitored by undervoltage comparator. If undervoltage is detected, the pumping is stopped and the drivers will be switched off.

The configuration of the bridge driver HW modules is done in Config Wizard.

3.6.4 Shunt resistor and CSA

The TLE987x is meant to be used typically with single shunt resistor, enabling a cost-optimized solution compared to multi-shunt approaches. The resulting voltage of the CSA is then measured by ADC1 via an internal current sense amplifier, as shown in Figure 26.

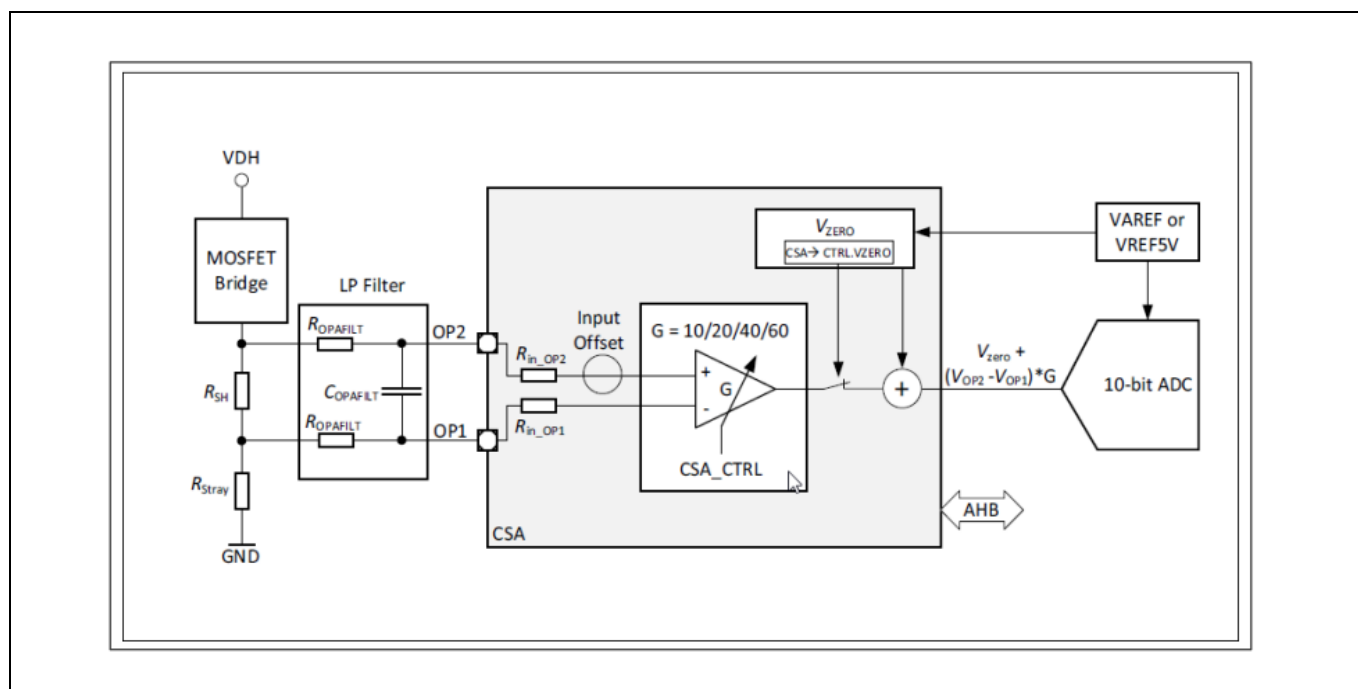


Figure 26 CSA in TLE987x

The user can refer to the “[TLE987x/6x Hardware Design Application Note](#)” for details and recommendations for the CSA configuration and the external components.

4 Structure of the BASIC SW

The BASIC FOC motor control SW is developed based on libraries structured on a well-defined layered approach. The architecture is designed in such a way to separate the libraries into specific groups. This allows different modules in a given layer to be replaced without affecting the performance in other modules, as well as the structure of the complete system.

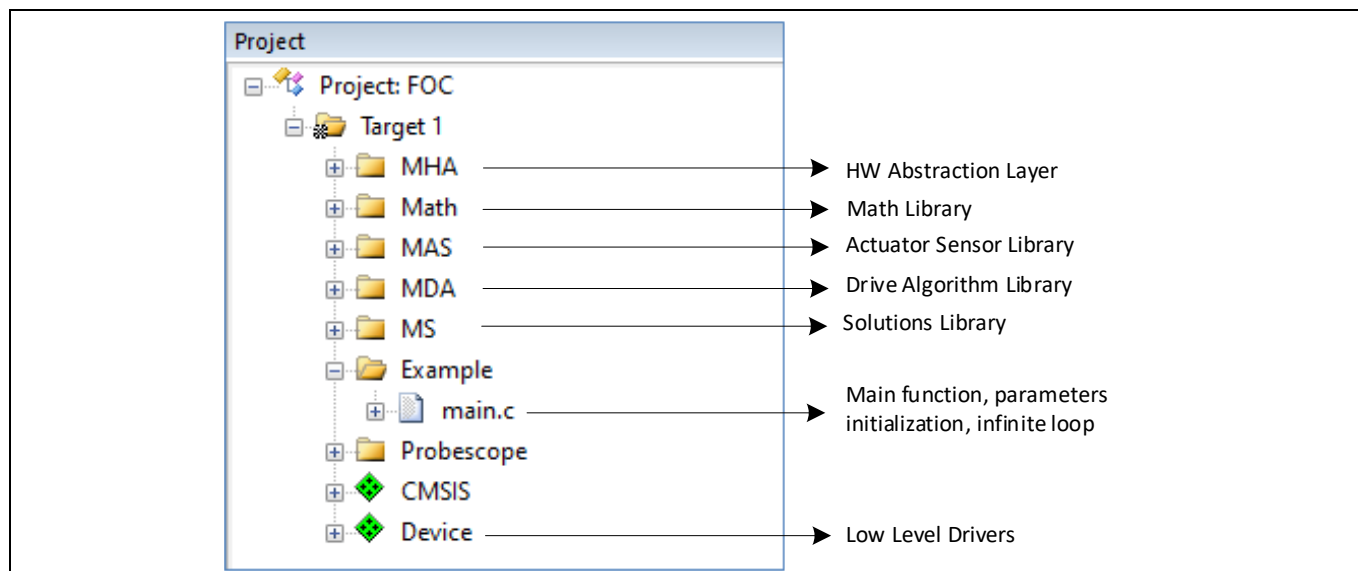


Figure 27 BASIC SW framework in KEIL

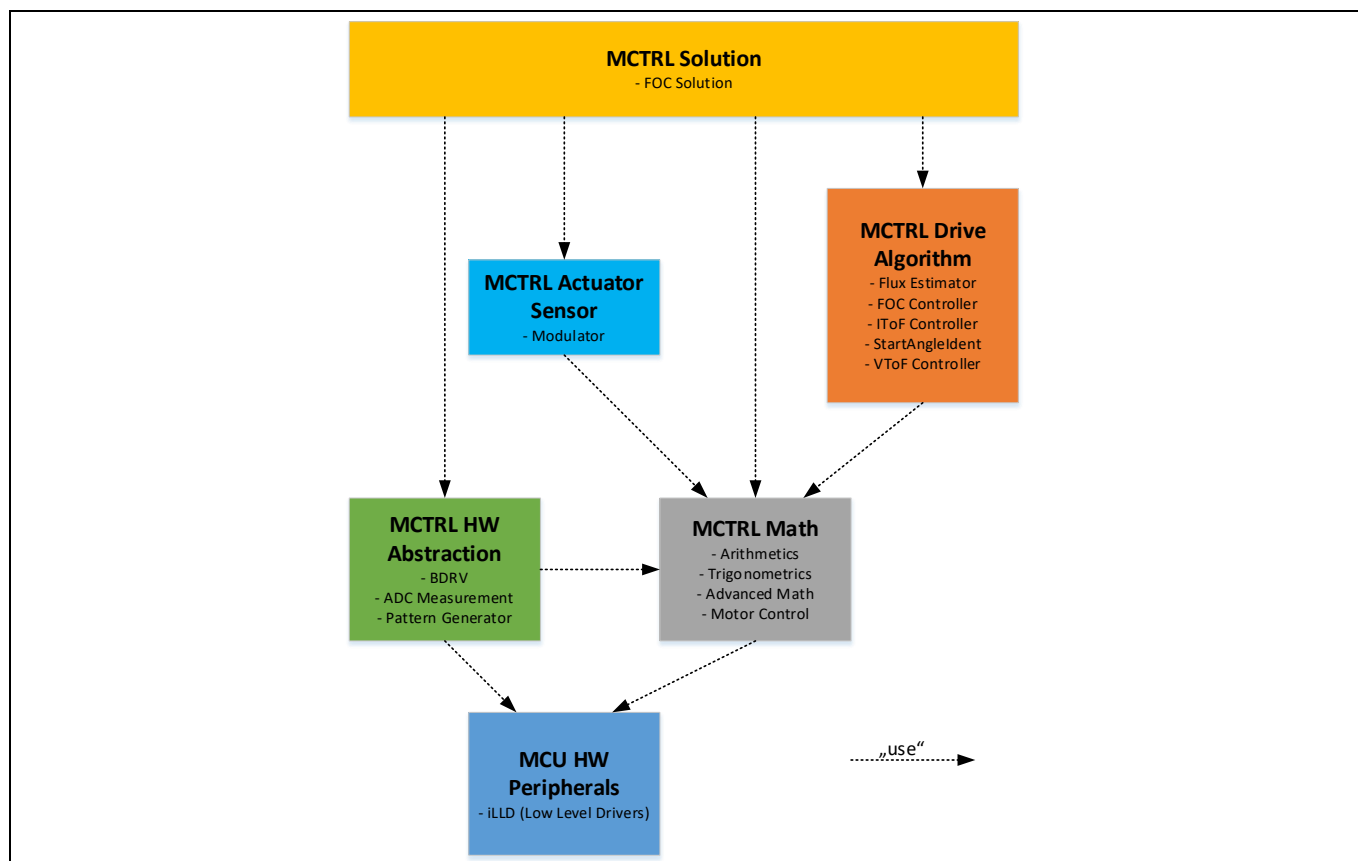


Figure 28 BASIC SW architecture overview

4 Structure of the BASIC SW

A detailed description of the libraries, as well as additional documentation, is included in the SW licensed version. A first overview is provided in the following sections.

4.1 MCTRL Solution (“FOC Solution”)

MCTRL Solution (MS-BAS) contains the main “**FOC solution**” state machines and the speed controller. This library utilizes all the modules available in the BASIC SW, providing two control modes for driving a motor: field oriented control (FOC mode) and voltage to frequency control (VToF control mode).

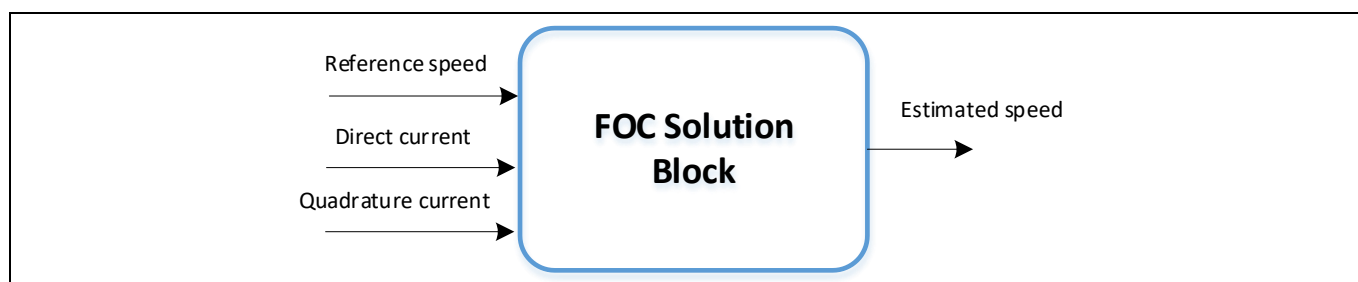


Figure 29 FOC solution block (simplified)

4.2 MCTRL Drive Algorithm

The MCTRL Drive Algorithm (MDA-BAS) module estimates the electrical **position** and electrical **frequency** of the rotor flux, based on stator voltage and current. It contains multiple blocks described in following section.

- **Flux estimator:** this software component estimates the electrical position and electrical frequency of the rotor flux based on stator voltage and stator current in stator reference frame (alpha-beta). The angle of the rotor flux is estimated based on a mathematical model of the motor and processed with a phase locked loop. The angular speed (electrical angular velocity) is calculated from the angle by using a differentiation with averaging and a low pass filter.

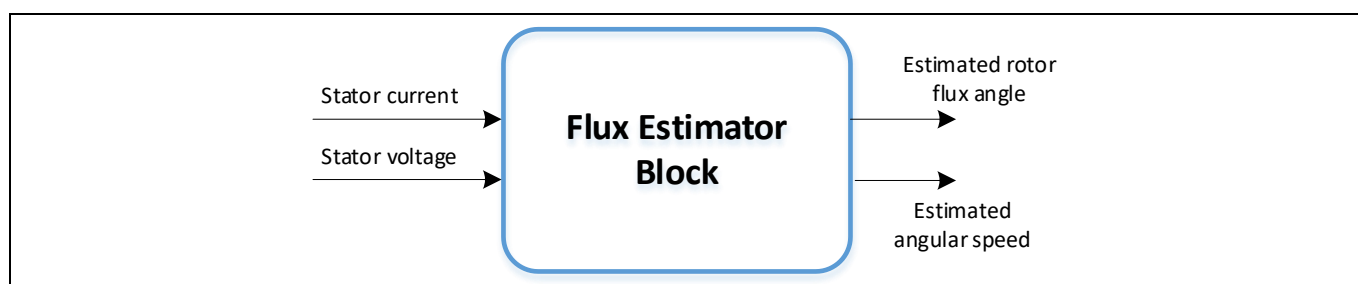


Figure 30 Flux estimator block (simplified)

- **FOC controller:** it takes as input the currents in alpha-beta format, the direct and quadrature reference currents and the rotor flux angle. Then it performs the current control using two PI controllers, and outputs the voltage command in polar format.

4 Structure of the BASIC SW

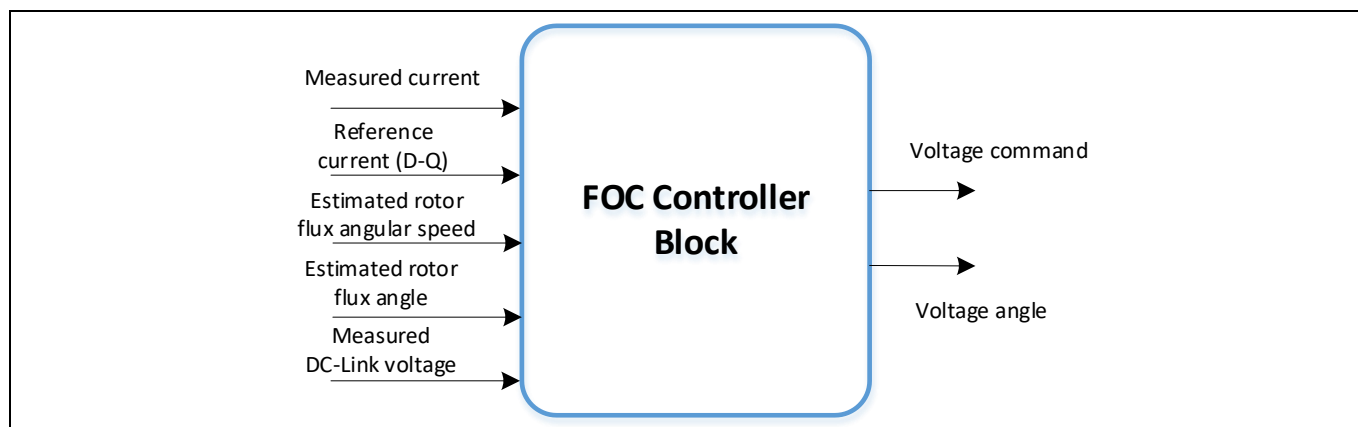


Figure 31 FOC controller block (simplified)

- **Current to frequency (IToF):** this software component generates the reference direct and quadrature currents as well as the angle of a rotating voltage vector, in order to enable a current to frequency control of a rotating field machine. The IToF block is used during the start-up phase, when the control method is open loop.

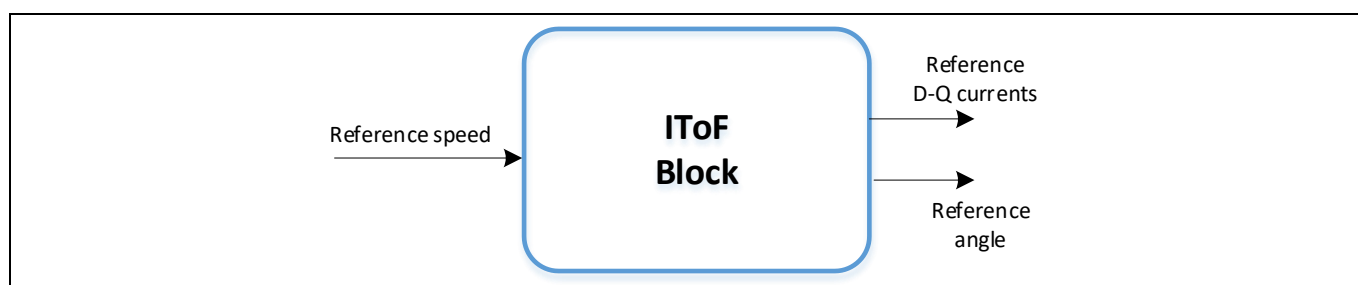


Figure 32 Current to frequency block (simplified)

- **Control voltage to frequency (VToF):** it generates a rotational voltage command in alpha- beta coordinate system based on the input reference speed. The VToF control mode is an open loop control structure, hence there is no feedback from the motor such as motor position or motor current.

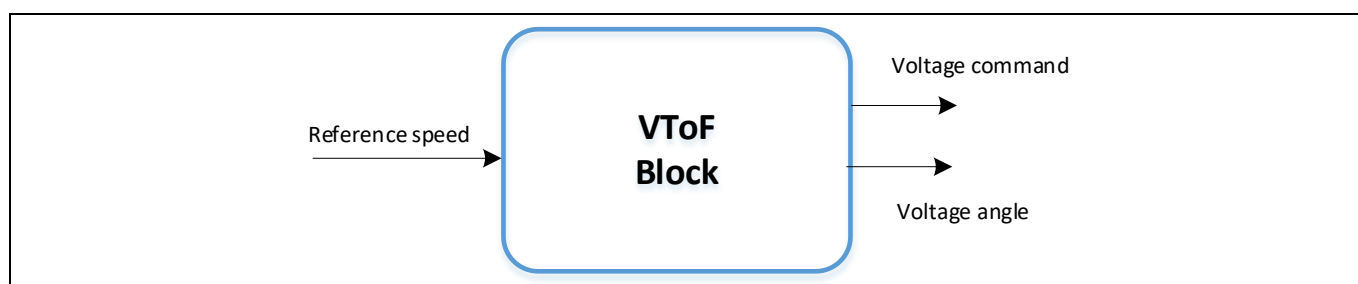


Figure 33 Voltage to frequency block (simplified)

- **Control angle identification (Start Angle Ident):** it is used to estimate the electrical rotor angle at standstill with an error of 60 degrees.

4 Structure of the BASIC SW

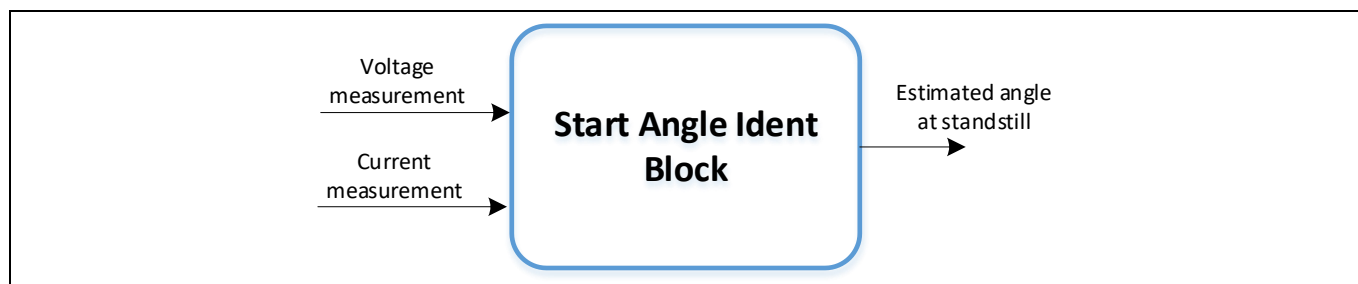


Figure 34 Control angle identification block (simplified)

4.3 MCTRL Actuator Sensor

The MCTRL Actuator Sensor (MAS-BAS) contains the **modulator block** and its different techniques, which provide the signals to the pattern generator (for PWM generation) and to the current reconstruction block. Space vector modulation (SVM) is a fundamental basic function for any kind of BLDC motor control application. The modulator contains math algorithms and calculations, executed in the current control loop.

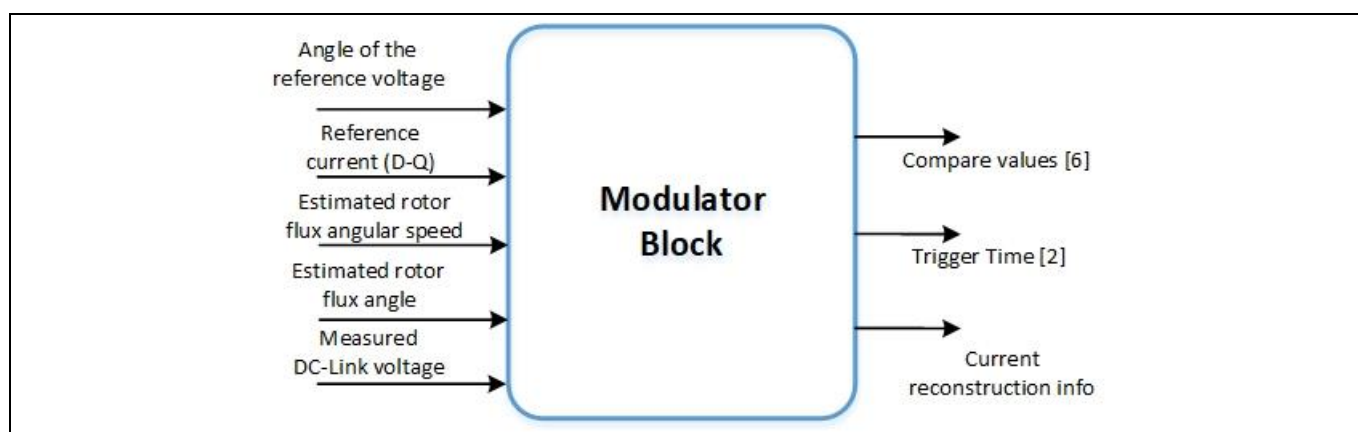


Figure 35 Modulator block (simplified)

4.4 MCTRL Hardware Abstraction Layer

The MCTRL Hardware Abstraction (MHA-BAS) consists of SW components that manage the TLE987x HW blocks, such as the bridge driver and the ADC measurement. This layer allows to read current from the ADC, to generate a PWM and to detect a fault from the bridge driver.

- **Pattern Generator:** it sets the timer compare values needed by the hardware module that drives the switches of the bridge driver, as well as the timer compare values needed by the hardware module to trigger a current measurement. The timer compare values are calculated and provided by the modulator. The Pattern Generator can be configured to set the PWM frequency, to set the dead-time of the system, and it can be configured to react on a trap fault.

4 Structure of the BASIC SW

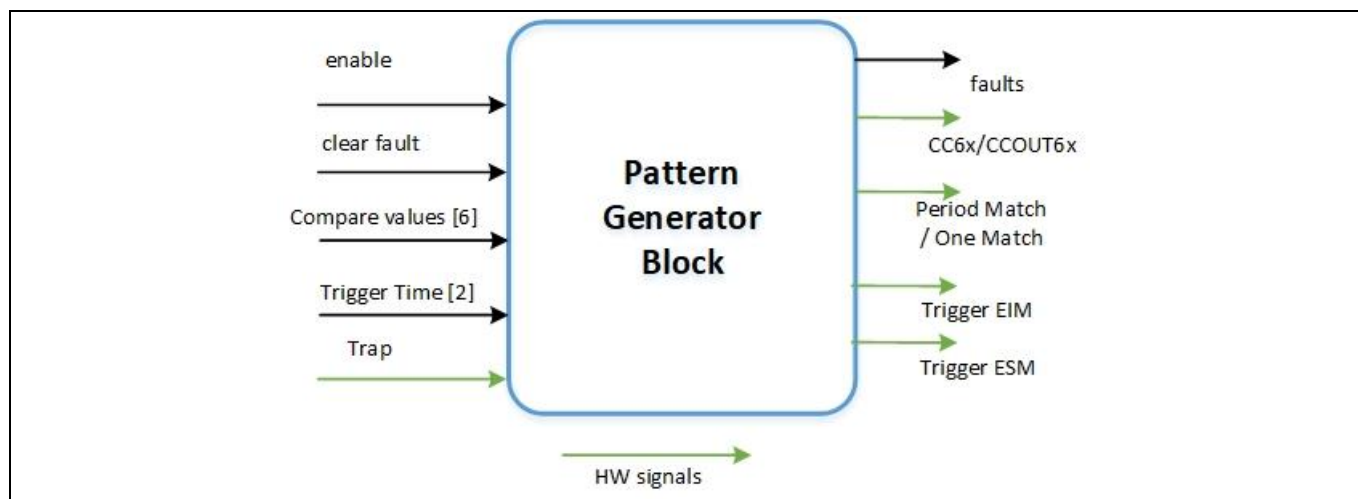


Figure 36 Pattern Generator block (simplified)

- **Bridge Driver:** it provides the management of the hardware faults detected by the BDRV HW module.

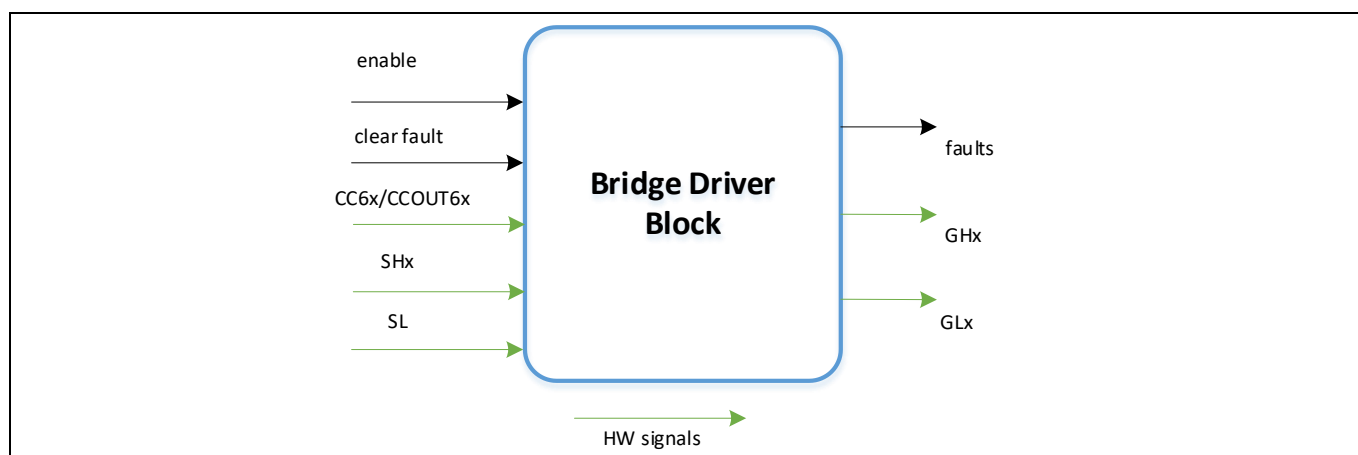


Figure 37 Bridge driver block (simplified)

- **Measurement ADC:** it reads the DC link voltage and the phase current from the hardware measurement module (current sense amplifier), removes the current measurement offset, and scales the values by the respective base value. In addition to that, it can be used to calibrate the current offset during the initialization phase.

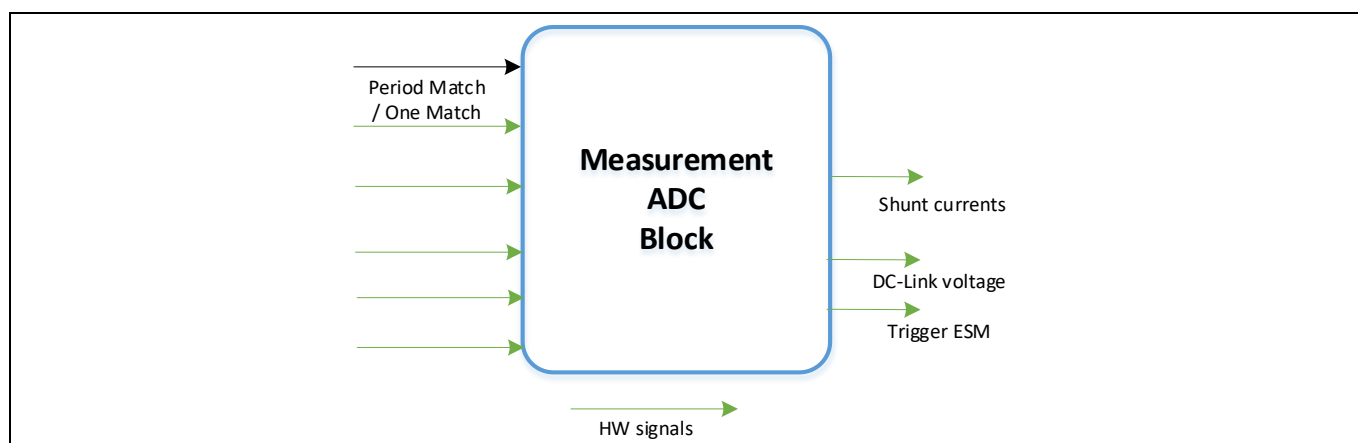


Figure 38 Measurement ADC block (simplified)

4.5 MCTRL Math

The MCTRL Math (MAS-BAS) provides optimized mathematical functions with multiple header and source files. Each header file contains a specific function, like for example Clarke transformation, or a group of similar operations, for instance multiplication of 16 bits and of 32 bits variables with or without saturation.

The static configuration of the functions is done in the Config Wizard tool.

The **arithmetic functions** provide simple arithmetic operations like addition, subtraction, multiplication, with or without saturation.

The **trigonometric functions** included in the BASIC SW are operations like:

- **Atan**: arctangent of a fractional input value
- **Atan2**: function to calculate the four-quadrant arctangent of x (abscissa) and y (ordinate)
- **Sin**: sine of an angle
- **Cos**: cosine of an angle
- **SinCos**: the sine and cosine value of an angle

Look-up tables have been implemented in the SW (for example like Lut_Atan, Lut_SinCos) to store the values for the specific functions. The size of the look-up tables can be set in the Config Wizard (8 bits, 10 bits or 12 bits).

The **math functions**, used specifically in motor control and provided within the BASIC SW, are the following:

- **CartToPolar**: function to calculate the Cartesian (α , β) to polar (amplitude, angle) transformation
- **PolarToCart**: function to perform the polar to Cartesian transformation, necessary to calculate the coordinates from polar (amplitude, angle) system to Cartesian (α , β) system
- **Clarke**: function to calculate the Clarke transformation of the three phase input
- **InvClarke**: function to calculate the inverse Clarke transformation of the two-axis orthogonal input
- **Park**: function to calculate the Park transformation
- **InvPark**: function to calculate the inverse Park transformation
- **Current reconstruction**: function to calculate the three phase currents based on two single shunt measured currents and the sector number
- **DqDecoupling**: to implement the decoupling of the DQ currents in a field oriented controller
- **SpeedPreControl**: function to implement speed pre-control (also known as feed-forward speed control)

Moreover, the MCTRL Math consists also of software components, which provide advanced mathematics operations like linear interpolation, elements of control algorithms like PI controller, low-pass filter, PLL, rate limiter.

- **PI controller**: PI controller functions with anti-windup feature and different discretization methods
- **PLL**: Phase locked loop (PLL) function which takes an angle as input and calculates the speed as well as filters the input angle
- **AccelerationLimit**: function to limit the acceleration and deceleration of an input speed signal

4 Structure of the BASIC SW

- **Interp1DLUT**: function that calculates the intermediate values between known points (provided within a 1-D Lookup Table) by linear interpolation
- **Limit**: function to limit an input variable between the set maximum and minimum values.
- **RateLimiter**: function to limit the rate of change of a signal
- **LowPass1st**: function to filter an input signal by a 1st order type low pass filter

4.6 MCU HW Peripherals

The MCU HW Peripherals (or iLLD – Low Level Drivers) are provided in the SDK of the TLE987x and they consist of all the hardware register settings of the TLE987x peripherals, such as ADC, CCU6, UART, bridge driver, watchdog etc. The configuration is done in the Config Wizard tool (Figure 39).

Application	MCTRL Solution	MCTRL Drive Algorithm	MCTRL Actuator Sensor	MCTRL Hardware Abstraction	Math Library	TLE987x Peripherals												
BSL	SCU	PMU	MON	Interrupt	PORT	ADC1	ADC2	BDRV	BEMF	CCU6	GPT12E	Timer2x	Timer3	UART	SSC	LIN	CSA	DMA

Structure	Settings
<ul style="list-style-type: none"> [-] fSYS <ul style="list-style-type: none"> [-] Clock Configuration <ul style="list-style-type: none"> [-] Automatic Clock Configuration <ul style="list-style-type: none"> fSYS: 40 MHz [-] Manual Clock Configuration <ul style="list-style-type: none"> <input type="radio"/> [-] Clock out enable <ul style="list-style-type: none"> <input type="checkbox"/> [-] WDT1 (mandatory watchdog) <ul style="list-style-type: none"> Watchdog Period: 1008 [ms] Watchdog Period effective: 1008ms [-] WDT enable (optional watchdog) <ul style="list-style-type: none"> <input type="checkbox"/> [-] NVM Protection 	

Figure 39 TLE987x Peripherals in Config Wizard

5 Configuration of the static parameters

The static parameters are constants that cannot be changed during the program execution. These parameters can be configured only before compiling the SW, in the Config Wizard or directly in the program configuration files.

In this section of the document, an overview of the SW parameters is given, as implemented in the Config Wizard tool.

Note: if not reported in the tables, the default values and the range of the parameters are easily visible in Config Wizard, in a pop-up window appearing moving the mouse cursor over the value of each parameter.

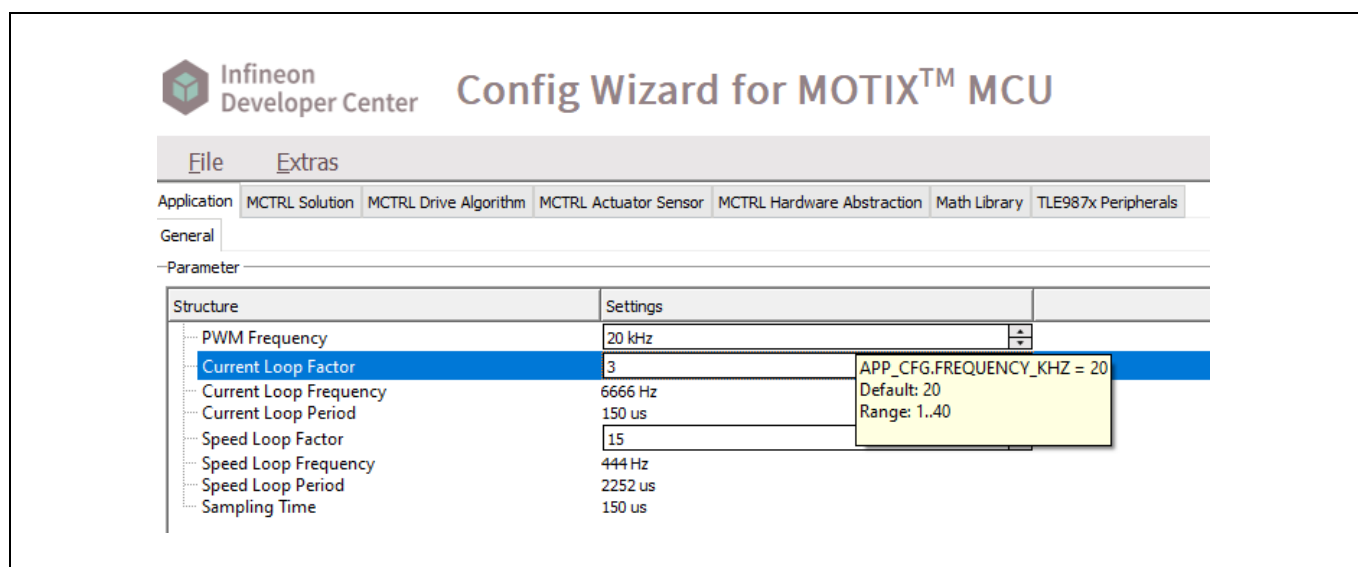


Figure 40 Software layers in the Config Wizard

5.1 Application layer

In the application layer, the PWM switching frequency, the current loop factor and the speed loop factor are configured.

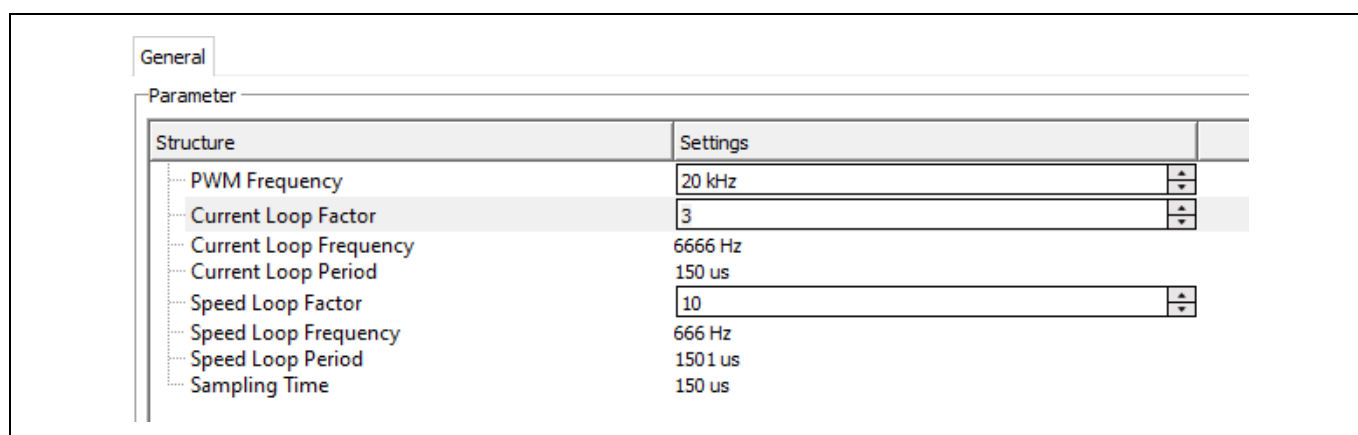


Figure 41 Application layer in Config Wizard

5 Configuration of the static parameters

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
PWM frequency	PWM switching frequency of the bridge driver	1	20	40	kHz
Current Loop Factor	Used to set the time to execute of the current control loop (fast loop) function	1	3	25	1
Current Loop Frequency	Calculated as current loop / PWM frequency		6666		Hz
Current Loop Period	Reciprocal of current loop frequency		150		μs
Speed Loop Factor	Used to set the time to execute of the speed control loop (slow loop) function	1	10	25	1
Speed Loop Frequency	Calculated as current loop frequency / speed loop factor		666		Hz
Speed Loop Period	Reciprocal of speed loop frequency		1501		μs
Sampling time	It configures the period of time in which the FOC Controller is executed, equal to the current loop period		150		μs

Note: the program execution time starts from 35 us and varies upon the user configuration in Config Wizard. If the PWM is set to 20 kHz, then one period is equal to 50 μs. If the program execution time is lower than 50 μs, then it is included within the period of the PWM, so the user can also configure a current loop factor equal to 1.

5.2 MCTRL solution

The MCTRL solution is the library that implements the FOC solution, which is the top layer in the MOTIX™ BASIC SW. The static parameters belonging to this library are classified in 4 categories:

1. Base values
2. System parameters
3. PI controllers
4. General operation settings

5.2.1 Base values

The base values are the maximum values that the SW can accept, so the inputs should be the maximum values that can be given to the motor.

Important: the base values are used for scaling, they are not the nominal parameters of the motor.

5 Configuration of the static parameters

Base values		
Structure	Settings	
System base voltage	24.00000 V	▲ ▼
System base current	5.00000 A	▲ ▼
System base mechanical speed	4000 rpm	▲ ▼
System base resistance	4.8 Ohm	
System base inductance	2.87 mH	
System base electrical speed	1675 rad/s	
System base torque	0.29 Nm	

Figure 42 MCTRL solution - base values in Config Wizard

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
System base voltage	Maximum voltage that can be represented in the SW, used for scaling		24		V
System base current	Maximum current that can be represented in the SW, used for scaling		5		A
System base mechanical speed	Maximum motor rotation speed that can be represented in the software for scaling		4000		rpm
System base resistance	Calculated from base voltage and current		4.8		Ohm
System base inductance	Calculated from system base speed, voltage and current		2.87		mH
System base electrical speed	Calculated from base mechanical speed and number of pole pairs		1675		rad/s
System base torque	Calculated from number of pole pairs, base power and base time		0.29		Nm

The equations used to calculate the base parameters are reported here below.

$$\text{Base power (W)} = \text{System base voltage} \times \text{System base current}$$

$$\text{Base time (s)} = \frac{1}{\text{Electrical base speed}}$$

$$\text{System base resistance (Ohm)} = \frac{\text{System base voltage}}{\text{System base current}}$$

$$\text{System base inductance (H)} = \frac{\text{System base power} \times \text{Base time}}{\text{System base current}^2}$$

$$\text{Electrical base speed} \left(\frac{\text{rad}}{\text{s}} \right) = \text{System base mechanical speed} \times \text{Number of pole pairs}$$

$$\text{Mechanical base Torque (Nm)} = \text{Number of pole pairs} \times \text{Base power} \times \text{Base time}$$

5 Configuration of the static parameters

5.2.2 System parameters

These parameters are the actual parameters of the motor, which can be found in the datasheet or which can be calculated manually. Moreover, in this section the parameters for start-up current of the motor are also included.

System Parameters	
Structure	Settings
Phase resistance	0.75000 Ohm
Direct phase inductance	1.05000 mH
Quadrature phase inductance	1.05000 mH
Average inductance (Ls)	1.05 mH
Pole pairs	4
Rotor inertia	2.4 mgm ²
Torque constant	0.035 Nm/A
Viscous friction constant	0.00000 mNms
Start-up current	1.00000 A
Initial Start-up current	0.00000 A
Start-up current ramp up rate	4.00000 A/s
Effective startup current ramp up rate	3.96 A/s
PWM frequency	20 kHz
Current loop factor	3
Current loop frequency	6666 Hz
Current loop period	150 us
Speed loop factor	10
Speed loop frequency	666 Hz
Speed loop period	1501 us
Sampling time	150 us

Figure 43 MCTRL solution – System Parameters in Config Wizard

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
Phase resistance	Resistance of the motor phase		0.75		Ohm
Direct phase inductance	Direct inductance of the motor phase		2.1		mH
Quadrature phase inductance	Quadrature inductance of the motor phase		2.1		mH
Average inductance	Calculated as average between direct and quadrature inductance		2.1		mH
Pole pairs	Number of pole pairs of the motor		4		1
Rotor Inertia	Moment of inertia of the rotor. If not available, keep the default value		2.4		mgm ²
Torque constant	Torque constant of the motor. If not available, keep the default value		0.035		Nm/A

Sensorless FOC motor control using Infineon MOTIX™ TLE987x

5 Configuration of the static parameters

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
Viscous friction constant	Viscous friction constant of the motor. If not available, keep the default value		0		mNm/s

Rotor inertia, torque constant and viscous friction constant are used in the **speed pre-control** function. This technique is used to improve the dynamics of speed control especially at higher speeds and if a rapid change in speed is needed. These parameters should be available in the data sheet of the motor.

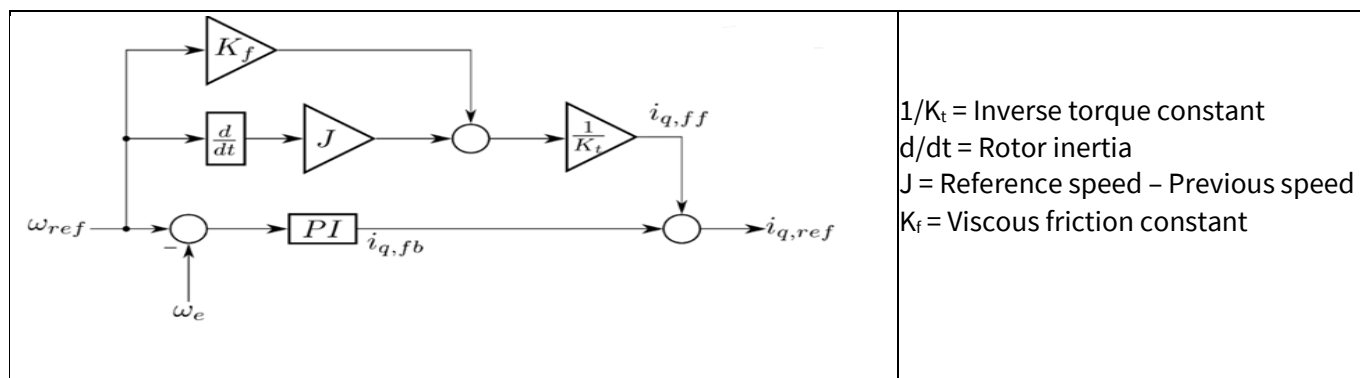


Figure 44 Speed pre-control

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
Start-up current	Constant current required to turn the motor initially at IToF open loop. This current should be big enough to start the motor, so it shall be adjusted according to the torque needed for load		1		A
Initial start-up current	Initial value of the start-up current, useful to set a level where the measurement has sufficient accuracy		0		A
Start-up current ramp up rate	Used for smooth acceleration by limiting ramp up rates of start-up current within the specified amplitude. Recommended to have		4		A/s

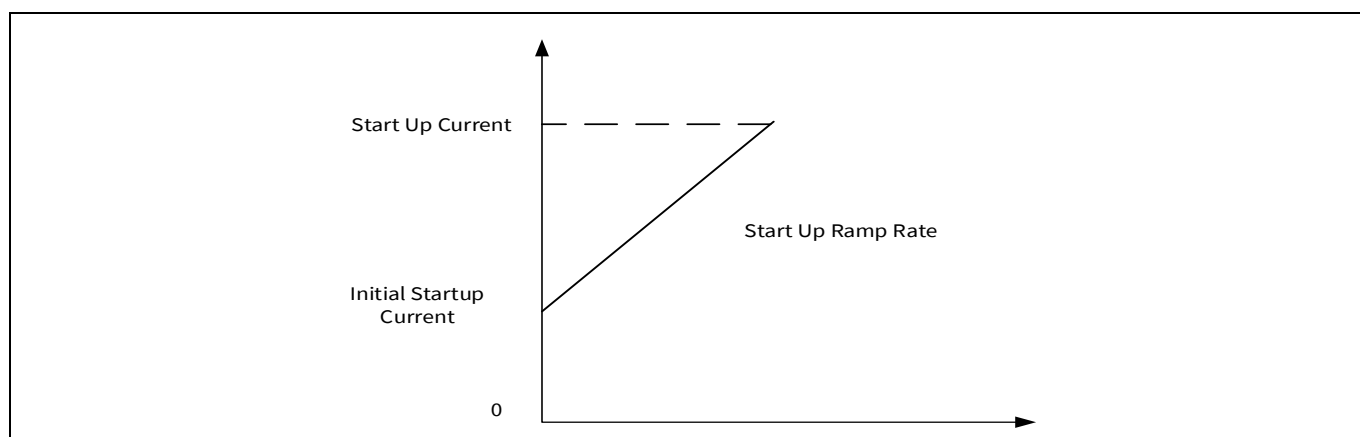


Figure 45 Start-up current with initial start-up ramp

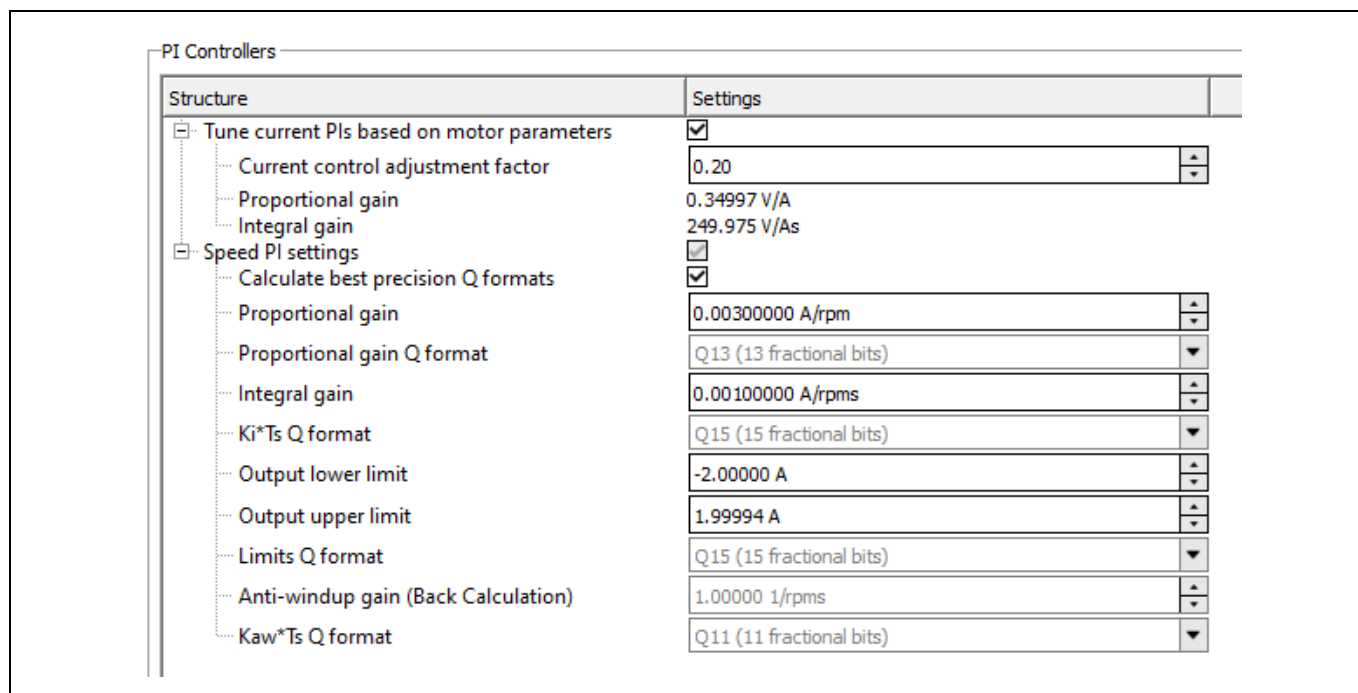
Sensorless FOC motor control using Infineon MOTIX™ TLE987x

5 Configuration of the static parameters

The remaining parameters of this layer are already described in the “application layer” section.

5.2.3 PI controllers

These parameters are the used to configure the current PI (ID and IQ) regulator, as well as for the speed PI regulator.



Structure	Settings
<input checked="" type="checkbox"/> Tune current PIs based on motor parameters	<input checked="" type="checkbox"/>
Current control adjustment factor	0.20
Proportional gain	0.34997 V/A
Integral gain	249.975 V/As
<input checked="" type="checkbox"/> Speed PI settings	<input checked="" type="checkbox"/>
Calculate best precision Q formats	<input checked="" type="checkbox"/>
Proportional gain	0.00300000 A/rpm
Proportional gain Q format	Q13 (13 fractional bits)
Integral gain	0.00100000 A/rpms
Ki*Ts Q format	Q15 (15 fractional bits)
Output lower limit	-2.00000 A
Output upper limit	1.99994 A
Limits Q format	Q15 (15 fractional bits)
Anti-windup gain (Back Calculation)	1.00000 1/rpms
Kaw*Ts Q format	Q11 (11 fractional bits)

Figure 46 MCTRL solution – PI Controllers parameters in Config Wizard

Tune current PIs based on motor parameters (checkbox): if this checkbox is enabled, the current PI regulators tuning depends upon the parameters of motor. If this checkbox is disabled, then the user can adjust the proportional and integral gains manually in the FOC controller section of the MCTRL Drive Algorithm layer.

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
Current control adjustment factor	Used for automatic calculation of the proportional and integral gains		0.2		1
Proportional gain	Calculated from the current control adjustment factor, used to improve stability and high frequency response of the current loop		0.34997		V/A
Integral gain	Calculated from the current control adjustment factor, used to reach current average error equal to zero		249.975		V/As

The calculation of the current gains is reported in the equations here below.

Current proportional gain

$$= \frac{\text{Current control adjustment factor} \times \text{Average inductance} \times \left(\frac{\text{Sampling frequency}}{1000} \right)}{4}$$

5 Configuration of the static parameters

$$\text{Current integral gain} = \frac{\text{Current control adjustment factor} \times \text{Phase resistance} \times \text{Sampling frequency}}{4}$$

Speed PI settings: they are basically the currents used by the current PI regulators (Id and Iq) in the speed loop control block, in order to achieve the target speed set by the user.

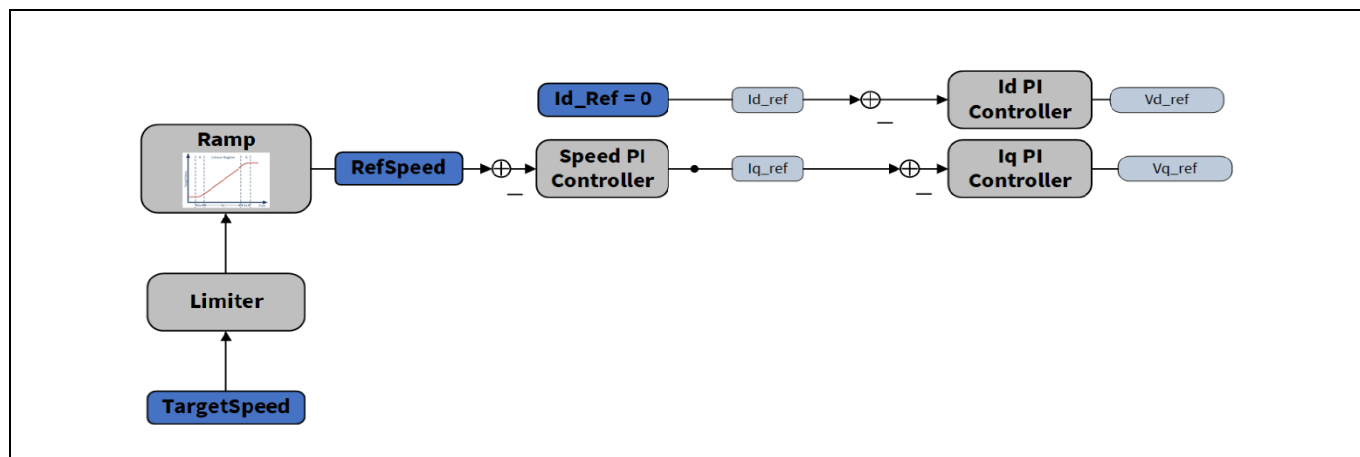


Figure 47 Speed loop control block

Calculate best precision Q formats (checkbox): in the BASIC SW libraries, the format of the parameters is called Q format (abbreviated as Qy, where the “y” represents the number of fractional bits of the parameter). If this checkbox is enabled, the best Q format is automatically calculated. If this checkbox is disabled, then the user can adjust the Q formats for each parameter, from 0 to 15 fractional bits.

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
Proportional gain	Proportional gain of the speed controller, used to improve stability and high frequency response of the speed loop		0.003		A/rpm
Proportional gain Q format	Q format of the proportional gain of the speed controller		Q13		
Integral gain	Integral gain of the speed controller, used to reach speed average error equal to zero		0.001		A/rpms
Ki*Ts Q format	Q format of the integral gain of the speed controller		Q15		
Output lower limit	Lower limit of the current of the speed controller. It can be adjusted according to the current at the target speed		-2		A
Output upper limit	Upper limit of the current of the speed controller. It can be adjusted according to the current at the target speed		1.99994		A
Limits Q format	Q format of the output limits		Q15		
Anti-windup gain (Back Calculation)	Anti-windup gain of the speed controller		1		1/rpms
Kaw*Ts Q format	Q format of the anti-windup gain		Q12		

5 Configuration of the static parameters

Anti-windup gain

The PI integral part of the speed controller acts in situations where there is a change in the system set point, such as when the user enters a speed target value different than the motor actual speed.

After setting the new speed, a constant error will then appear in the PI controller, so that the integrator error output will rise (Figure 48). The speed might then reach a saturation limit and it could remain at that constant value for some time. If the integration error keeps increasing above the saturation region (“windup”), this may lead to bad dynamic behavior of the motor. The speed can be controllable only if the integral error part is reduced.

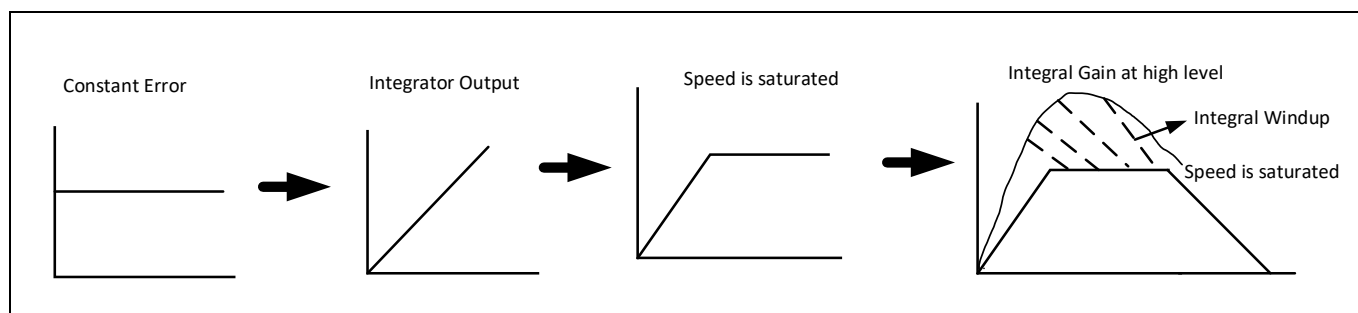


Figure 48 Anti-windup control

Clamping and back calculation techniques are used to avoid the overshoot (extra windup) of the PI integral values. These techniques are used to limit the PI integral values.

The anti-windup control status can be used in both clamp and back calculation modes. The **anti-windup gain** is especially used in back calculation method. A comparison between clamping and back calculation techniques is available in the literature publicly available.

The anti-windup feature can be configured in the Math Library layer.

5.2.4 General operation settings

The parameters in this section are related to the motor speed and the transition settings.

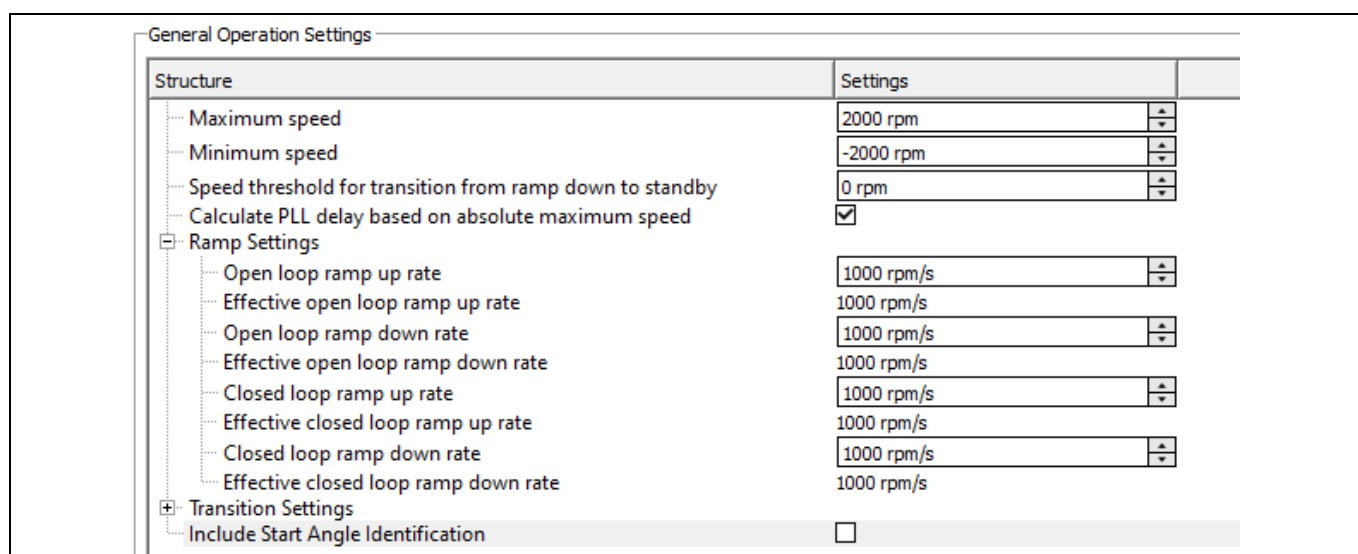


Figure 49 MCTRL solution - general operation settings

Sensorless FOC motor control using Infineon MOTIX™ TLE987x

5 Configuration of the static parameters

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
Maximum speed	Maximum motor speed supported by the SW		2000		rpm
Minimum speed	Minimum motor speed supported by the SW		-2000		rpm
Speed threshold for transition from ramp down to standby	Minimum speed to trigger the transition from ramp down to standby state. This variable can be used to set a minimum speed after which the motor goes into freewheeling		0		rpm

Calculate PLL delay based on absolute maximum speed (checkbox): the PLL is used to calculate the speed of the motor, measuring the rotor angle. The delay works basically as a filter. This option enables or disables the automatic calculation of the PLL delay based on the absolute maximum speed of the motor. The PLL delay is used in the flux estimator. It calculates the average rate of change of the input angle. This checkbox is linked to the **Math Library -> PLL delay length**.

$$PLL\ delay: 2^{\left(\left\lfloor \frac{\log_{10}\left(60 \times \frac{1000000}{\left(\frac{Sampling\ time}{Maximum\ speed \times Pole\ pairs \times 4}\right)}\right)}{\log_{10}(2)}\right)\right)}$$

Ramp settings

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
Open loop ramp up rate	Ramp up acceleration in open loop. With high loads, a low value is recommended		1000		rpm/s
Open loop ramp down rate	Ramp down deceleration in open loop. With high loads, a low value is recommended		1000		rpm/s
Closed loop ramp up rate	Ramp up acceleration in closed loop		1000		rpm/s
Closed loop ramp down rate	Ramp down deceleration in closed loop		1000		rpm/s

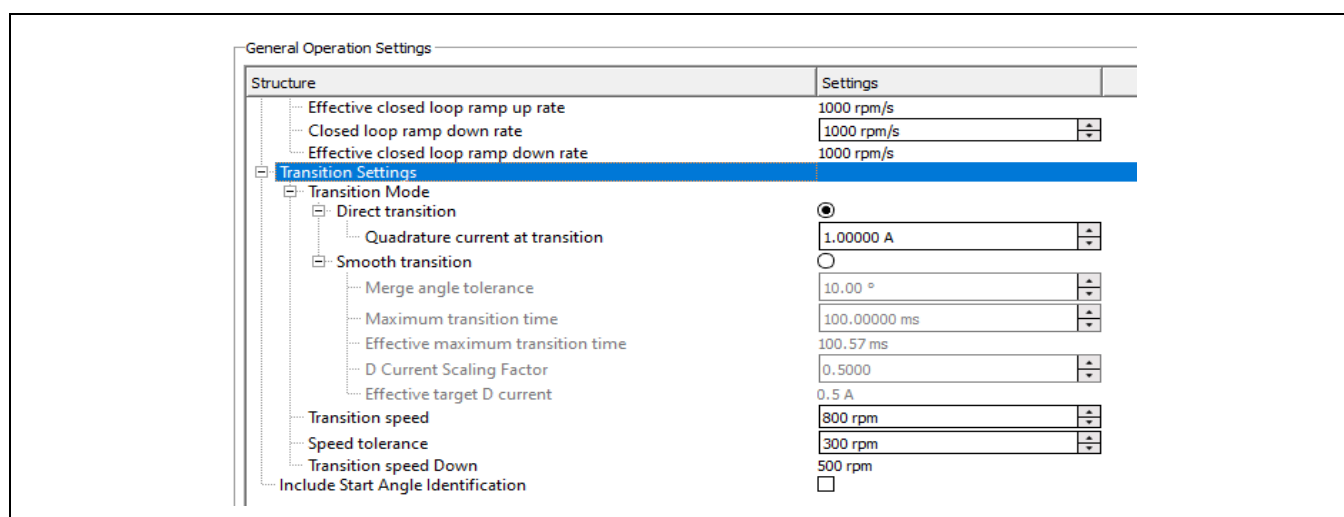


Figure 50 MCTRL solution - general operation settings – transition mode

Sensorless FOC motor control using Infineon MOTIX™ TLE987x

5 Configuration of the static parameters

In the transition settings section, the user can configure the mode of the transition from open loop control to closed loop control and the other way around.

Direct transition: if enabled, the transition is done instantly from open loop to closed loop or from closed loop to open loop.

Smooth transition: if enabled, the transition is done following a ramp during a configurable period of time. The smooth transition is used to reduce the angle error between IToF and FOC loop.

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
Quadrature current at transition	Speed PI integral value during transition from open to close or from close to open loop		1		A
Merge angle tolerance	Angle error between IToF angle and flux estimator angle, when the FOC solution switches from open-loop control to closed-loop control. Also known as transition angle tolerance		10		Degrees
Maximum transition time	Time limit for the smooth transition. If the angle error is not below the merge angle tolerance in the defined transition time, then FOC will go from open loop to closed loop		100		ms
D current scaling factor	Ratio between the target D current for smooth transition and the start-up current		0.5		
Transition speed	Maximum speed limit for the open loop, after that it will switch to closed loop		800		rpm
Speed tolerance	Difference between transition speed up and transition speed down.		300		rpm
Transition speed down	Minimum speed limit for the closed loop, after that it will switch to open loop. Equal to: Transition speed - Speed tolerance		500		rpm

Include start angle identification: if enabled, it enables the start angle identification feature in the FOC.

5.3 MCTRL drive algorithm

In the MCTRL drive algorithm library, the following software components can be configured:

- **Flux estimator:** it estimates the electrical position and electrical frequency of the rotor flux based on stator voltage and stator current in stator reference frame (alpha-beta). The angle of the rotor flux is estimated based on a mathematical model of the motor and processed with a phase locked loop. The angular speed (electrical angular velocity) is calculated from the angle by using a differentiation with averaging and a low pass filter
- **FOC controller:** it takes as input the currents in alpha-beta format, the direct and quadrature reference currents and the rotor flux angle. Then it performs the current control using two PI controllers, and it outputs the voltage command in polar format
- **I/f:** known as current to frequency scalar control module (IToF), it generates the reference direct and quadrature currents as well as the angle of a rotating voltage vector to enable a current-to-frequency control of a rotating field machine
- **V/f:** known as voltage to frequency scalar control module (VToF)
- **Start angle identification:** it estimates the electrical rotor angle at standstill with an error of 60 degrees.

5.3.1 Flux estimator

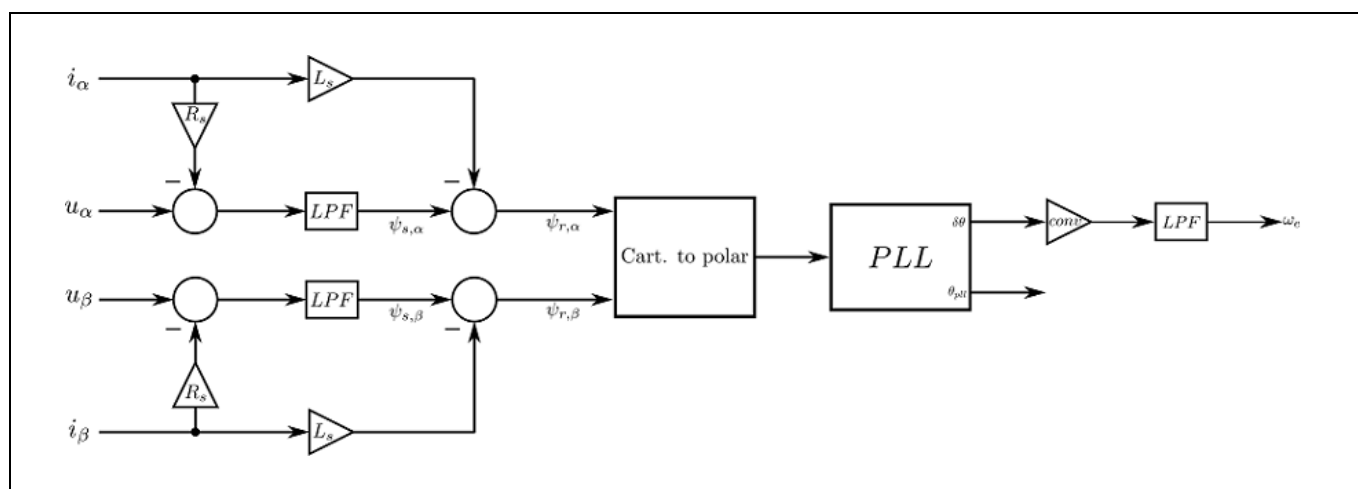
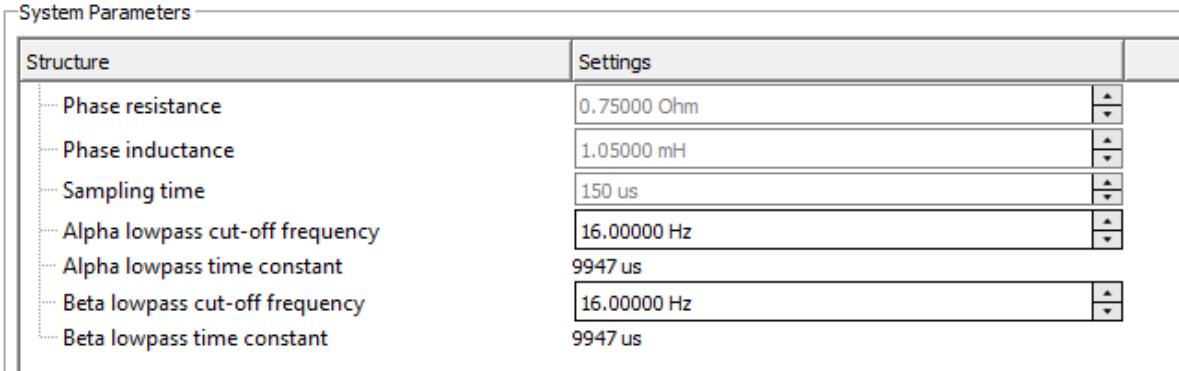


Figure 51 Flux estimator - Block diagram

5 Configuration of the static parameters



Structure	Settings
Phase resistance	0.75000 Ohm
Phase inductance	1.05000 mH
Sampling time	150 us
Alpha lowpass cut-off frequency	16.00000 Hz
Alpha lowpass time constant	9947 us
Beta lowpass cut-off frequency	16.00000 Hz
Beta lowpass time constant	9947 us

Figure 52 Flux estimator – System Parameters

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
Alpha low pass cut-off frequency	Cut-off frequency of the low pass filter of the stator alpha and beta components. The low pass filter is equivalent to a high pass filter in series with an integrator. This configuration corresponds to the cut-off frequency of the high pass filter		16		Hz
Beta low pass cut-off frequency			16		Hz

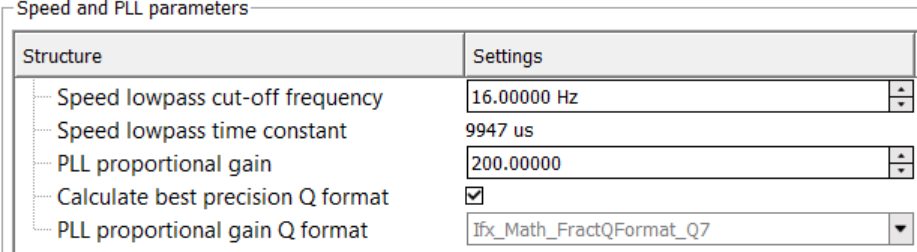
The flux estimator calculates angle and speed correctly when there is a stable phase current. The user can choose the speed from which the flux estimator can execute the calculation.

For example: flux estimator calculation from 240 rpm with 4 pole pairs

$$\text{Stator frequency} = \frac{\text{Speed in RPM} \times \text{Pole pairs}}{60}$$

$$\text{Stator frequency} = 240 \times \frac{4}{60} = 16 \text{ Hz}$$

The cut-off frequency shall be then at least 16 Hz. If the cut-off frequency is set too low, then this may lead to noise at lower speeds, so that the flux estimator calculates the angle and the speed incorrectly.



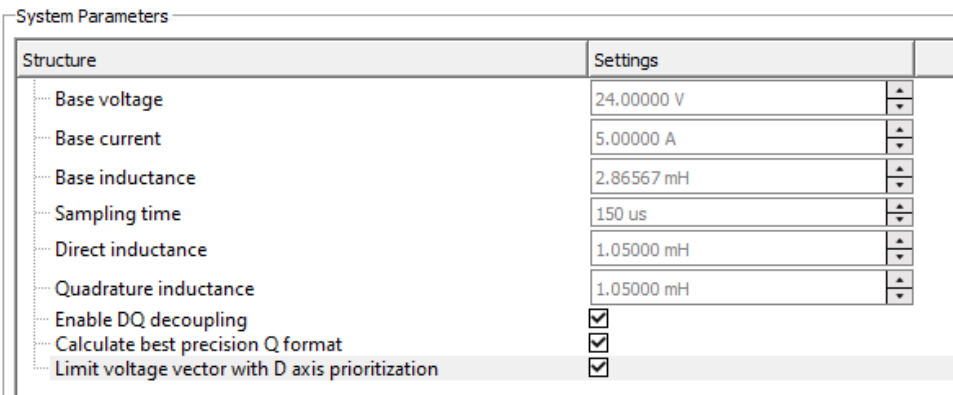
Structure	Settings
Speed lowpass cut-off frequency	16.00000 Hz
Speed lowpass time constant	9947 us
PLL proportional gain	200.00000
Calculate best precision Q format	<input checked="" type="checkbox"/>
PLL proportional gain Q format	Ifx_Math_FractQFormat_Q7

Figure 53 Flux estimator – Speed and PLL parameters

5 Configuration of the static parameters

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
Speed lowpass cut-off frequency	Cut-off frequency of the speed low pass filter		16		Hz
PLL proportional gain	Proportional gain of the phase locked loop filter, used in the flux estimator		200		
PLL proportional gain Q format	Q format selection of the PLL proportional gain		Q7		

5.3.2 FOC controller



Structure	Settings
Base voltage	24.00000 V
Base current	5.00000 A
Base inductance	2.86567 mH
Sampling time	150 us
Direct inductance	1.05000 mH
Quadrature inductance	1.05000 mH
Enable DQ decoupling	<input checked="" type="checkbox"/>
Calculate best precision Q format	<input checked="" type="checkbox"/>
Limit voltage vector with D axis prioritization	<input checked="" type="checkbox"/>

Figure 54 FOC controller– System Parameters

Enable DQ decoupling: if enabled, this feature increases the dynamics of the current control. It calculates the voltage that compensates the cross-coupling effect between the direct and quadrature axis of a permanent magnet synchronous motor.

A permanent magnet synchronous motor can be modelled in the dq rotor reference frame with the following equations:

$$\begin{aligned}
 u_d &= R_s \times i_d + L_d \times \frac{di_d}{dt} - \omega_e \times L_q \times i_q \\
 u_q &= R_s \times i_q + L_q \times \frac{di_q}{dt} + \omega_e \times L_d \times i_d + \omega_e \times \psi_{pm}
 \end{aligned}$$

As can be seen from the model, the speed and current dependent cross-coupling voltages ($-\omega_e \times L_q \times i_q$) and ($\omega_e \times L_d \times i_d$) appear in the d and q axis respectively. This implies that a change in one axis will have an effect on the other axis. This can be seen as a disturbance to the FOC scheme, where the aim of the controller is to control the two axes independently.

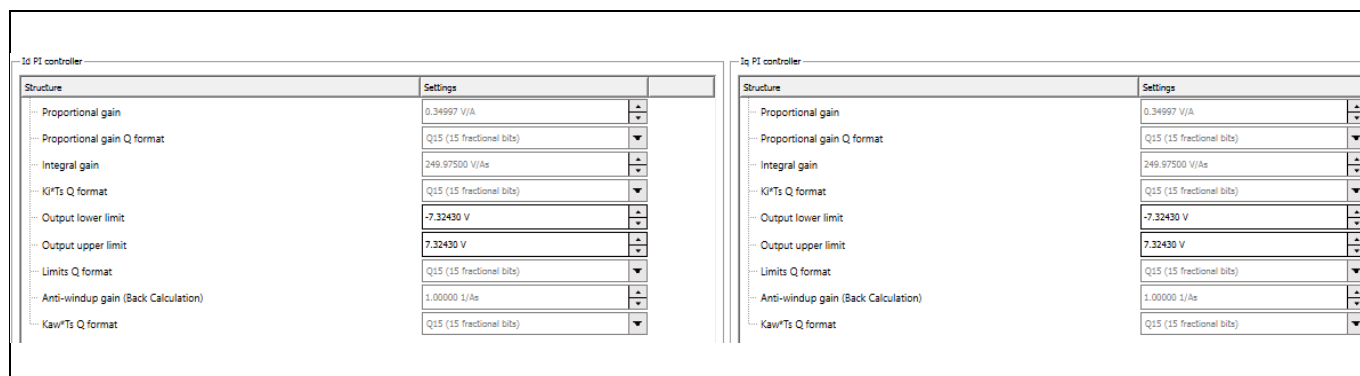
Sensorless FOC motor control using Infineon MOTIX™ TLE987x

5 Configuration of the static parameters

Calculate best precision Q format: if enabled, the best precision Q format for the parameters of the Id and Iq PI controller are calculated.

Limit voltage vector with D axis prioritization: it enables or disables the limitation of the amplitude of the output voltage vector, according to the available DC link voltage with a D component prioritization.

The maximum allowable voltage amplitude is the DC link voltage divided by square root of 3. First the D voltage component is limited to the maximum allowed voltage amplitude. Then the Q voltage component is limited so that the amplitude of the resulting calculated voltage vector does not exceed the maximum allowable voltage.



The screenshot displays two side-by-side configuration panels for the Id and Iq PI controllers. Each panel has a 'Structure' column with a tree view and a 'Settings' column with input fields and dropdown menus. The parameters and their values are as follows:

Parameter	Id PI controller Value	Iq PI controller Value
Proportional gain	0.34997 V/A	0.34997 V/A
Proportional gain Q format	Q15 (15 fractional bits)	Q15 (15 fractional bits)
Integral gain	249.97500 V/As	249.97500 V/As
Integral gain Q format	Q15 (15 fractional bits)	Q15 (15 fractional bits)
Output lower limit	-7.32430 V	-7.32430 V
Output upper limit	7.32430 V	7.32430 V
Limits Q format	Q15 (15 fractional bits)	Q15 (15 fractional bits)
Anti-windup gain (Back Calculation)	1.00000 1/As	1.00000 1/As
Kaw*Ts Q format	Q15 (15 fractional bits)	Q15 (15 fractional bits)

Figure 55 FOC controller– Id and Iq PI controller parameters

The tuning of the PI controller gain parameters is possible if in the MCTRL Solution the checkbox “Tune current PIs based on motor parameters” is disabled.

For simplicity, the Id and Iq parameters are described in the same table.

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
Proportional gain	Proportional gain of the PI controller		0.34997		V/A
Integral gain	Integral gain of the PI controller		249.975		V/As
Output lower limit	Minimum value of the output voltage vector of the PI controller, as defined by the user. The absolute minimum value is: $Vd_{ref\ minimum} = - \frac{DClink\ Voltage}{\sqrt{3}}$		-7.3243		V
Output upper limit	Maximum value of the output voltage vector of the PI controller, as defined by the user. The absolute maximum value is: $Vd_{ref\ maximum} = \frac{DClink\ Voltage}{\sqrt{3}}$		7.3243		V

5 Configuration of the static parameters

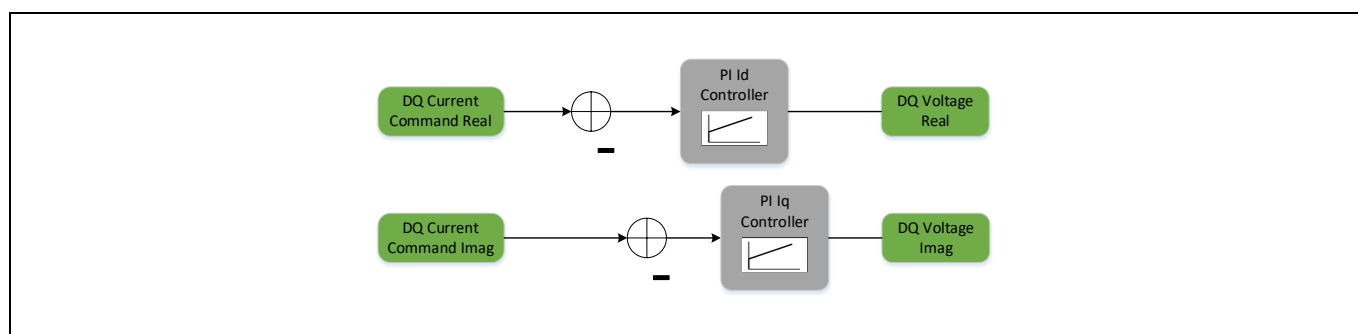


Figure 56 Current PI regulators

5.3.3 I/f software component

Also known as current to frequency block, it generates the reference direct and quadrature currents as well as the angle of a rotating voltage vector, in order to enable a current to frequency control of a rotating field machine in open loop. The parameters are equal to the ones configured in the FOC solution.

Flux Estimator	FOC Controller	I/f	V/f	Start Angle Identification
Structure		Settings		
Sampling time		150 us		
Base electrical speed		1675 rad/s		
Base current		5.00000 A		
Real Current		0.00000 A		
Imaginary Current		1.00000 A		

Figure 57 FOC controller– I/f parameters

5.3.4 V/f software component

Also known as voltage to frequency (VToF), it generates a rotational voltage command in alpha beta coordinate system based on the input reference speed. The VToF controller is an open loop structure. It is only recommended in applications where the load characteristics are stable, such as for fans and pumps.

Application		MCTRL Solution	MCTRL Drive Algorithm	MCTRL Actuator Sensor	MCTRL Hardware Abstraction	Math Library	TLE987x Peripherals
Flux Estimator		FOC Controller	I/f	V/f	Start Angle Identification		
Structure				Settings			
Base voltage				24.00000 V		▲▼	
Base mechanical speed				4000 rpm		▲▼	
Base electrical speed				1675 rad/s			
Sampling time				150 us		▲▼	
Pole pairs				4		▲▼	
Rated mechanical speed				1999 rpm		▲▼	
Rated electrical speed				837 rad/s			
Corner mechanical speed				250 rpm		▲▼	
Corner electrical speed				104 rad/s			
Voltage at rated speed				12.00000 V		▲▼	
Voltage at corner speed				0.75000 V		▲▼	
Voltage at minimum speed				0.75000 V		▲▼	

Figure 58 FOC controller– V/f parameters

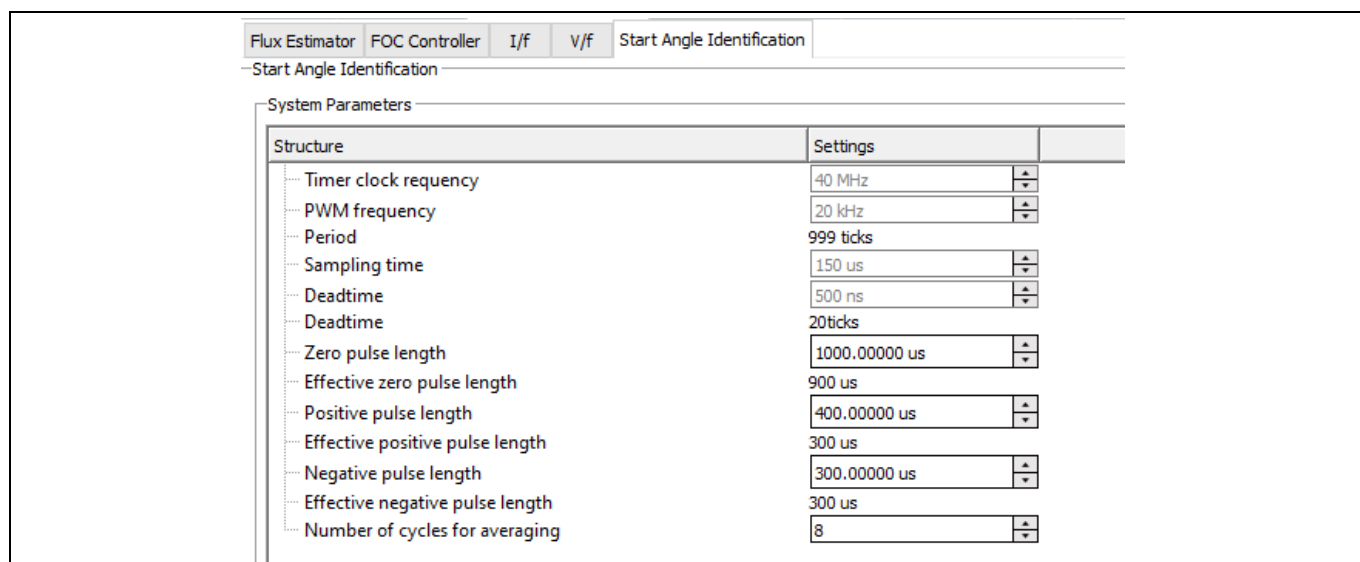
Sensorless FOC motor control using Infineon MOTIX™ TLE987x

5 Configuration of the static parameters

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
Rated mechanical speed	Rated mechanical speed of the motor		1999		rpm
Corner mechanical speed	Corner mechanical speed of the motor		250		rpm
Voltage at rated speed	Rated voltage of the motor, reached at motor rated speed		12		V
Voltage at corner speed	Voltage at the corner mechanical speed of the motor		0.75		V
Voltage at minimum speed	Voltage at the minimum mechanical speed of the motor		0.75		V

5.3.5 Start Angle Identification

The Start Angle Identification is used to estimate the electrical rotor angle at standstill, with an error of 60 degrees.



Structure	Settings
Timer clock frequency	40 MHz
PWM frequency	20 kHz
Period	999 ticks
Sampling time	150 us
Deadtime	500 ns
Deadtime	20 ticks
Zero pulse length	1000.00000 us
Effective zero pulse length	900 us
Positive pulse length	400.00000 us
Effective positive pulse length	300 us
Negative pulse length	300.00000 us
Effective negative pulse length	300 us
Number of cycles for averaging	8

Figure 59 MCTRL Drive Algorithm - Start Angle Identification system parameters

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
Timer clock frequency	Frequency of the system tick timer, as configured in the TLE987x device		40		MHz
PWM frequency	Frequency of the PWM, as configured in the Application layer		20		kHz
Deadtime	Dead time to prevent cross-conduction of the BDRV, as configured in the TLE987x device		500		ns
Zero pulse length	Length of the zero pulse, which is the first pulse applied, neither positive or negative		1000		μs
Positive pulse length	Length of the positive pulse		400		μs

Sensorless FOC motor control using Infineon MOTIX™ TLE987x

5 Configuration of the static parameters

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
Negative pulse length	Length of the negative pulse		300		μs
Number of cycles for averaging	Number of cycles to average the measurements of the current		8		

The parameters of the “Start Angle Identification – measurement trigger” are equal to the ones configured in the “MCTRL Actuator Sensor – Modulator measurement trigger”.

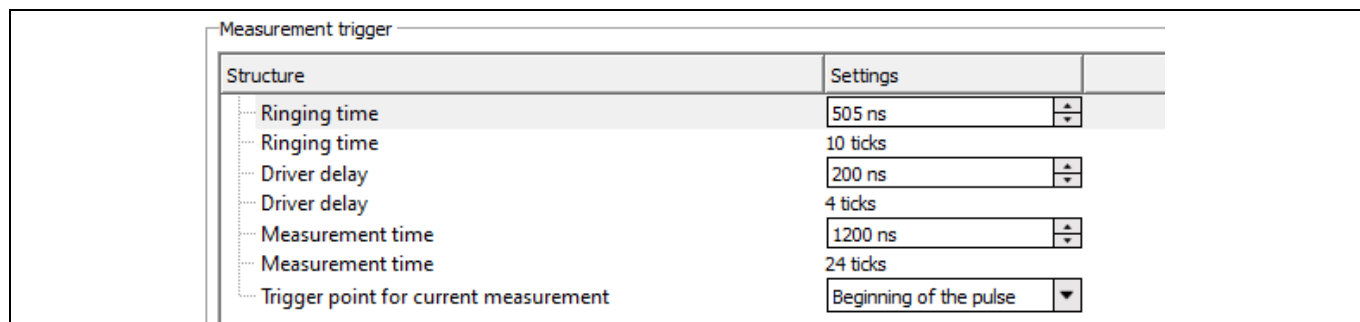


Figure 60 MCTRL Drive Algorithm - Start Angle Identification measurement trigger

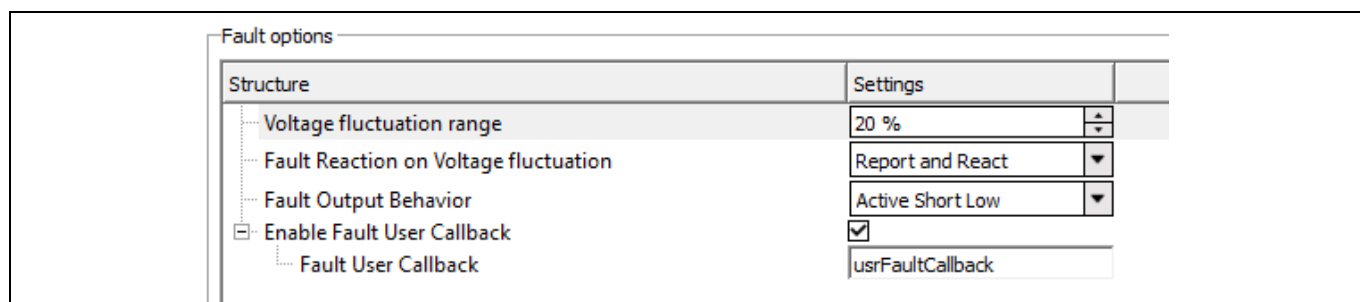


Figure 61 MCTRL Drive Algorithm - Start Angle Identification fault options

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
Voltage fluctuation range	Allowable voltage fluctuation range before an error is raised		20		%
Fault Reaction on Voltage fluctuation	Fault reaction if a "Vdc variation/fluctuation" fault is detected, which can be configured to one of the options listed		Report and React		
Fault output behavior	Fault output behavior if fault reaction is configured to report and react occurs.		Active Short Low		
Fault User Callback	If enabled, subroutine to be called in case any of the above listed faults occurs and is configured correspondingly.		usrFault Callback		

5 Configuration of the static parameters

5.4 MCTRL actuator sensor

This library contains the modulator software component. The space vector modulation (SVM) is a fundamental basic function for any kind of BLDC motor control application. The modulator contains complex math algorithms and time-consuming calculations, which must be executed in the current control loop.

5.4.1 System parameters

Modulator	
System Parameters	
Structure	Settings
Timer clock frequency	40 MHz
Base Voltage	24.00000 V
PWM frequency	20 kHz
Period	2000 ticks
Deadtime	500 ns
Deadtime	20 ticks
Enable deadtime compensation	<input type="checkbox"/>
Minimum ON time	0 ns
Minimum ON time	0 ticks
Transition from bi-directional two phase to bi-directional th...	0.100
Transition from bi-directional three phase to bi-directional t...	0.080

Figure 62 MCTRL actuator sensor – Modulator system parameters

Enable dead time compensation: if enabled, the dead time compensation is activated. In order to avoid a cross current through a phase, a time delay (dead time) is usually added to the turn on time of the switch. This leads to a reduction of the rms (root mean square) output voltage of the inverter as well as a phase shift as shown in the figure below:

- the blue curve is the reference voltage
- the black dotted curve is the output voltage of the inverter, distorted by the deadtime
- the red curve is the phase current, shifted by the inductive load behind the voltage

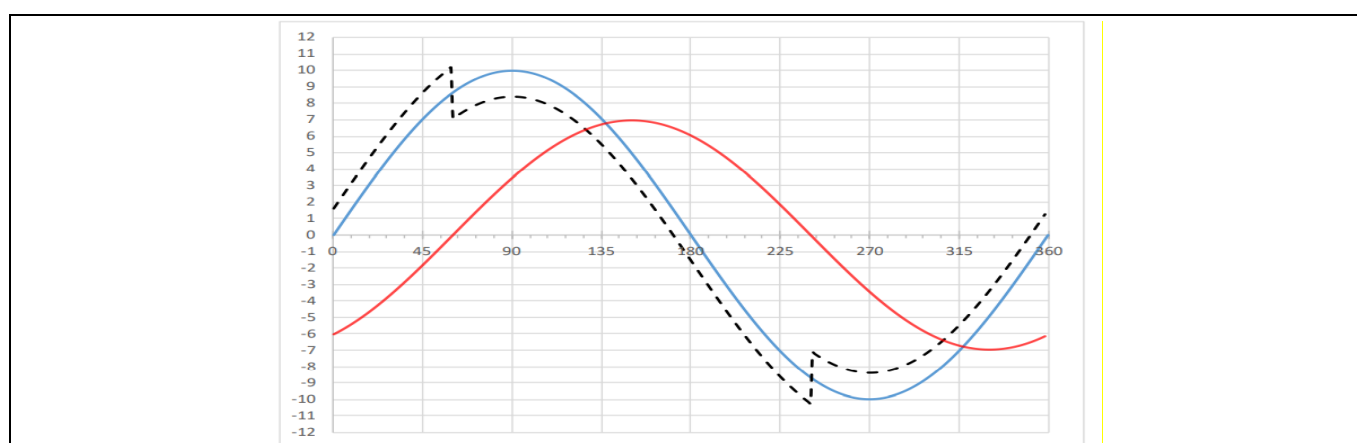


Figure 63 Dead time distortion

For a single leg, this is done by increasing the turn on time of the upper switch (decreasing the turn on time of the lower switch) when the current is positive and decreasing the turn on time of the upper switch (increasing the turn on time of the lower switch) when the current is negative, according to the figure above. In the simplest case, the time to increase the turn on time is the “duty cycle” of the dead time, defined as compensation time:

5 Configuration of the static parameters

$$T_{comp} = \frac{T_{dead_time}}{T_{sw}}$$

Where T_{comp} is the compensation value for the duty cycle, T_{dead_time} is the dead time value, and T_s is the sampling time. If the gate drivers have a turn on delay and a turn off delay this can be considered, then

$$T_{dead_time} = \text{Dead time} + T_{on} - T_{off}$$

Example:

- With a supply voltage of 12 V and 50% duty cycle, the expected output voltage is 6 V.
- 6 V cannot be achieved due to the dead time distortion.
- In order to achieve the expected voltage, it is needed to use dead time compensation technique, which increases the duty cycle of the PWM compared to the normal duty cycle and commands extra voltage such as 6.5 V instead of 6 V.

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
Minimum ON time	Minimum allowed on time of the gate signal of each switch (a non-zero dead time will lead to a shorter on time of the gate signal). This feature is used to prevent short pulses, where the switches cannot be effectively switched on due to dead time and turn-on time of the switches		0	ns	
Transition from bi-directional two phase to bi-directional three phase shifting	Ratio of commanded voltage and DC link voltage representing the hysteresis band upper limit, which is used to switch between bi-directional two phase and bi-directional three phase shifting		0.1		
Transition from bi-directional three phase to bi-directional two phase shifting	Ratio of commanded voltage and DC link voltage representing the hysteresis band lower limit, which is used to switch between bi-directional two phase and bi-directional three phase shifting		0.08		

The transition from 3-phase to 2-phase shifting takes place if there is an under modulation (small amplitude):

$$M = \frac{U_{ref}}{\frac{2}{\pi} U_{dc}}$$

Over modulation: the actual amplitude of the modulation index M is > 0.907 .

Normal modulation: the actual amplitude of the modulation index M is ≤ 0.907 . The circle represents the maximum amplitude. It means the circle should be inside the hexagon. If the modulation index > 0.907 , then it leads to over modulation, which is reported as fault.

5 Configuration of the static parameters

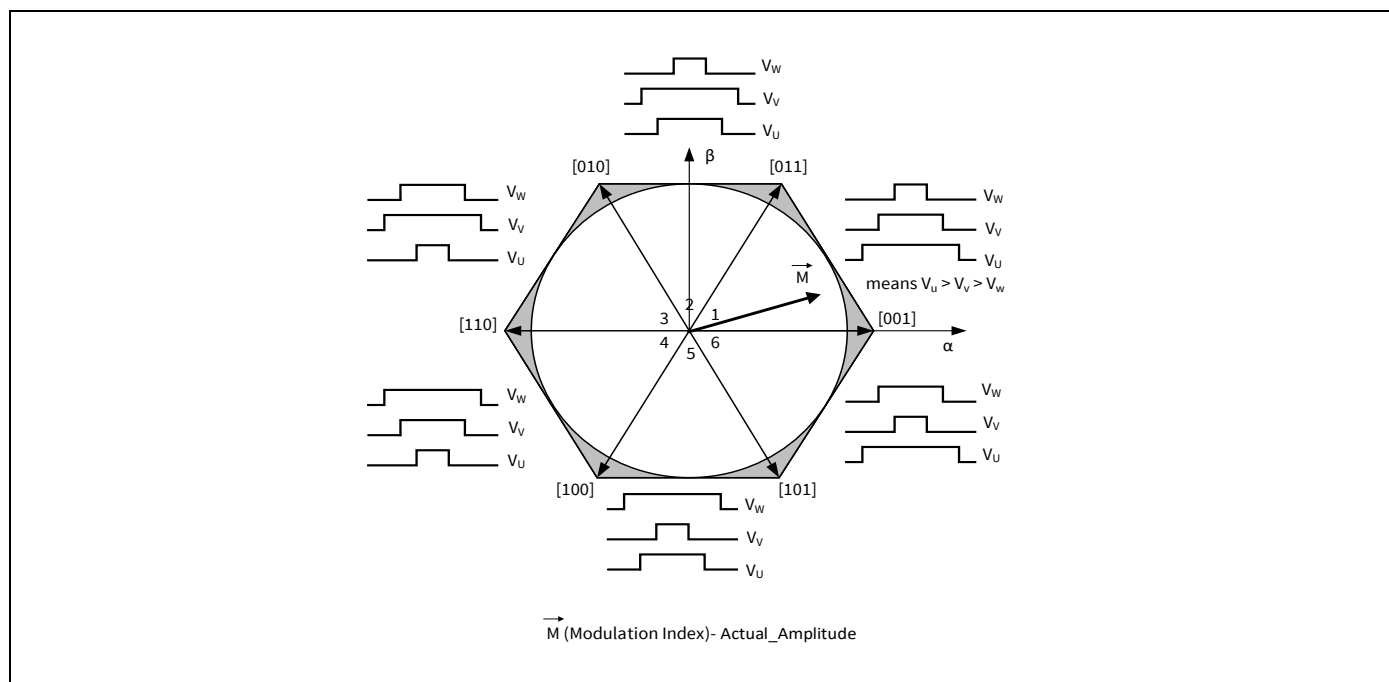


Figure 64 Normal modulation - modulation index within the expected range

Small amplitude (small gap between the pulses): the actual amplitude of the modulation index is too low, all the PWM duty cycles are around 50% duty cycle. In this situation it is difficult to trigger and measure the currents. The **asymmetric switching** helps to shift the pulses for the minimum measurement time, then it triggers the ADC.

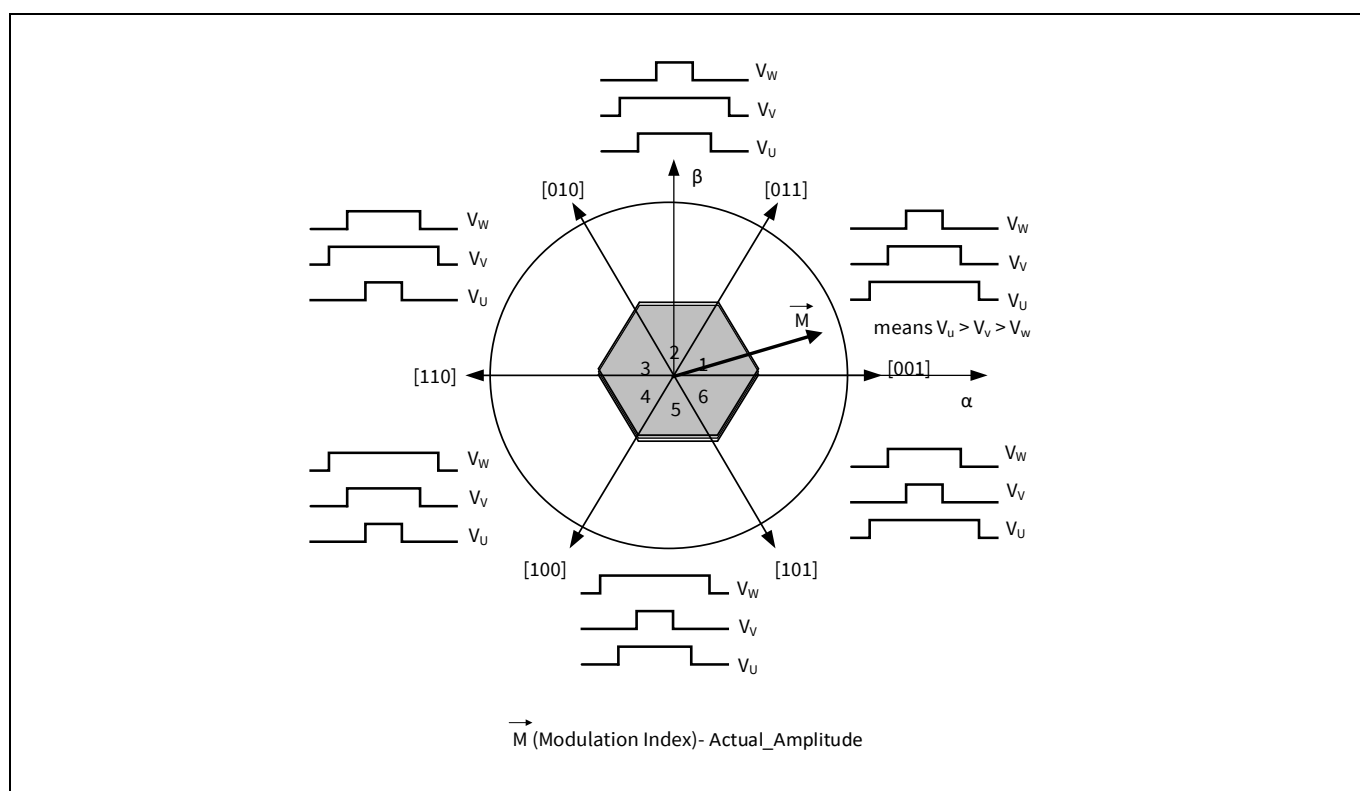


Figure 65 Modulation index with smaller amplitude

5 Configuration of the static parameters

The asymmetrical PWM switching strategy consists in shifting the PWM pulses to allow the current measurement, keeping the overall duty cycle. In case if the PWM pulse is shifted, the up-counting part and the down-counting are not equal and are calculated separately. For this reason, six timer compare values are necessary instead of only three.

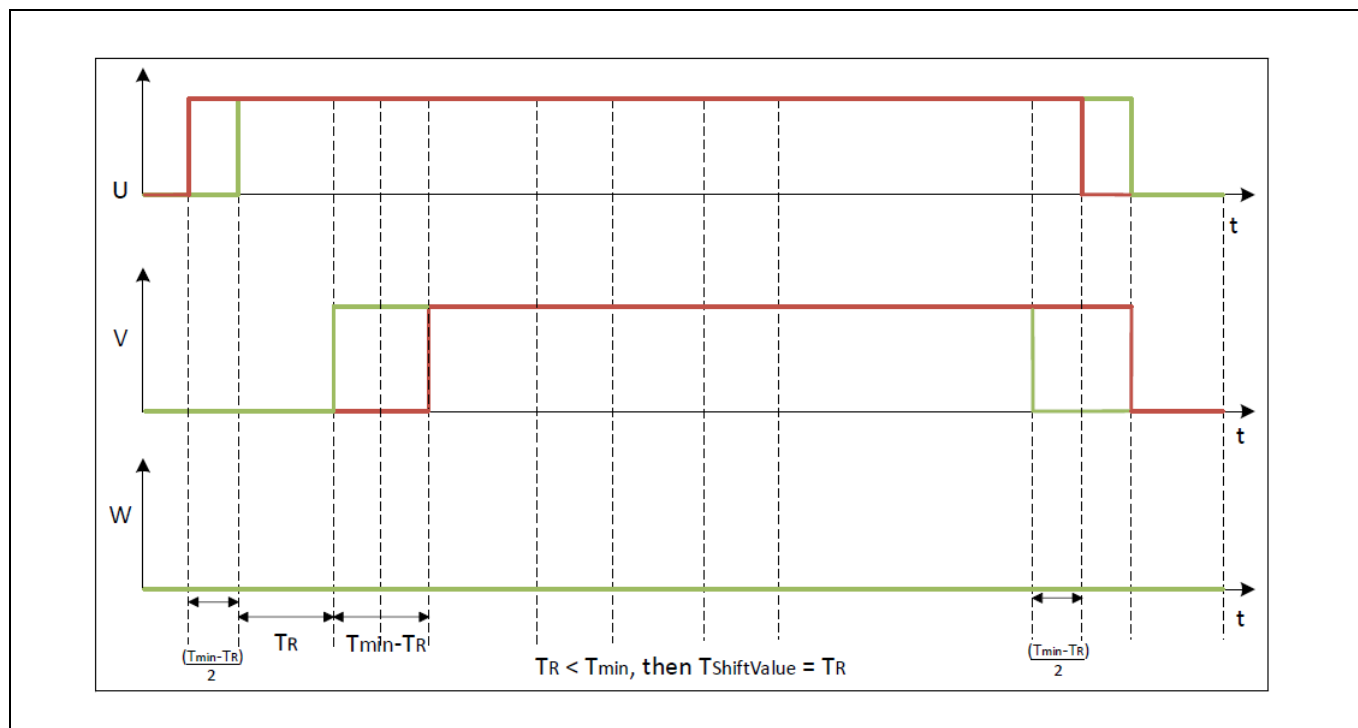


Figure 66 Example: asymmetric switching with pulse elimination for sector 0

5.4.2 Measurement trigger

Measurement trigger	
Structure	Settings
... Ringing time	500 ns
... Ringing time	20 ticks
... Driver delay	200 ns
... Driver delay	8 ticks
... Measurement time	1200 ns
... Measurement time	48 ticks
... Trigger point for current measurement	Beginning of the pulse

Figure 67 MCTRL actuator sensor – Modulator measurement trigger

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
Ringing time	Period of ringing time after switching a MOSFET, before starting the current measurement with the ADC		500		ns
Driver delay	Time delay between the switching command and the reaction of the gate driver		200		ns

Sensorless FOC motor control using Infineon MOTIX™ TLE987x

5 Configuration of the static parameters

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
Measurement time	Measurement time of the ADC current measurement		1200		ns
Trigger point for current measurement	Instant at which the current measurement is triggered. It is either at the start or at the end of the pulse. The earliest position in the pulse is calculated by considering the driver delay, ringing time and dead-time configuration. The latest position in the pulse is calculated by considering the driver delay and measurement time configuration		Beginning of the pulse		

The typical waveforms of the switching currents on the LSS and on the HSS are depicted in the timing diagram below (Figure 68).

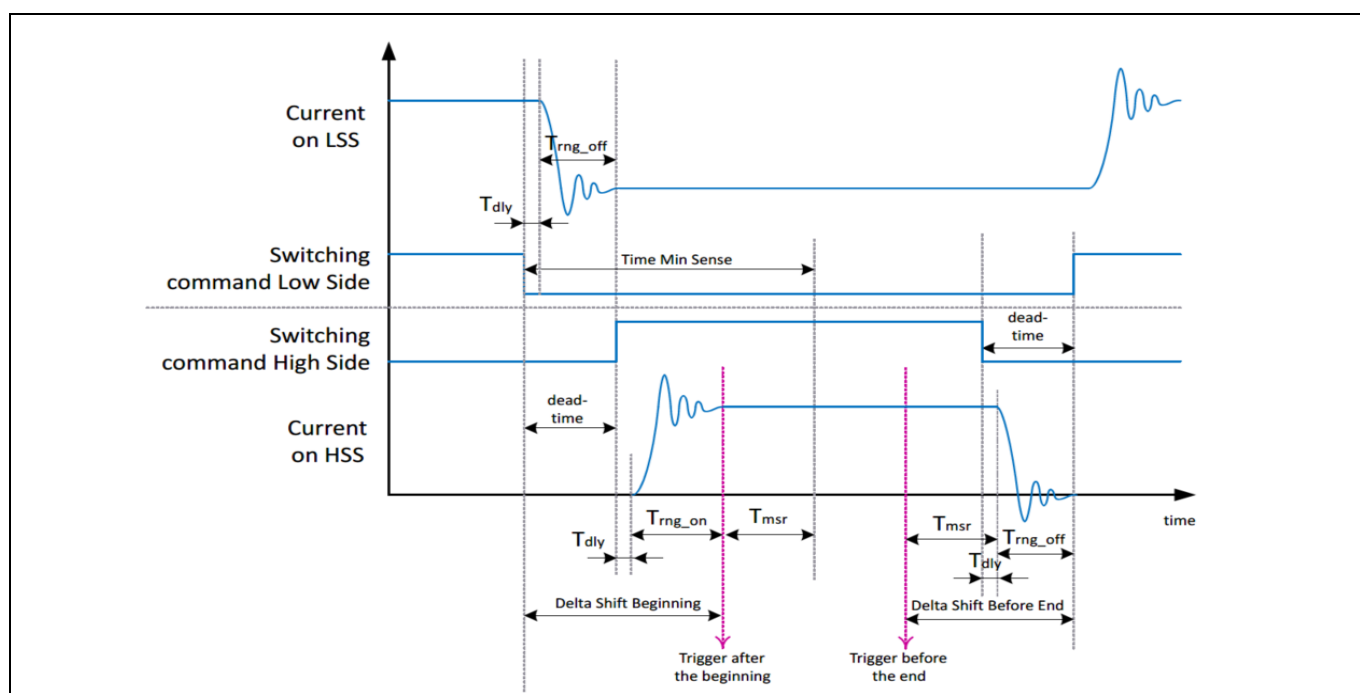


Figure 68 Switching time

5.4.3 Fault options

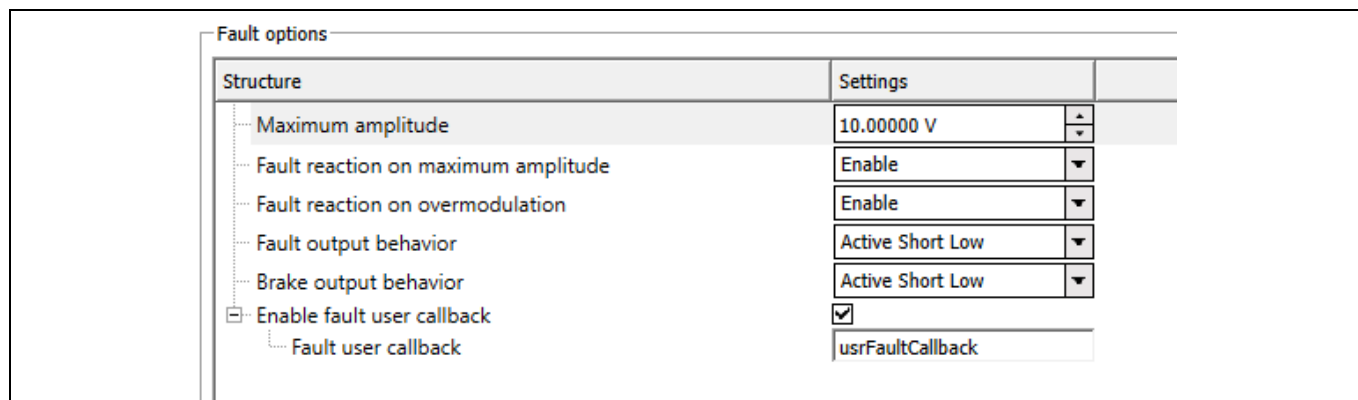


Figure 69 MCTRL actuator sensor – Modulator fault options

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
Maximum amplitude	Absolute value of the maximum permitted reference voltage amplitude		10		V
Fault reaction on maximum amplitude	Fault reaction mode in case that the reference voltage amplitude have exceeded the configured maximum amplitude. The options are: enable, disable, report only, report and react		Enable		
Fault reaction on over-modulation	Fault reaction mode in case that the modulator goes into over modulation mode. The options are: enable, disable, report only, report and react		Enable		
Fault output behavior	Behavior in case that any fault (maximum amplitude exceeded or over modulation entered) have occurred and the fault that occur is configured to Report and react		Active Short Low		
Brake output behavior	Behavior when a brake is requested.		Active Short Low		
Fault user callback	If enabled, name of the user-defined callback in which a custom-defined reaction can be programmed in case of modulator error		usrFaultCallback		

The fault and brake output behaviors are described in the table below.

Name	Description
Active short low	Closing the LSS of all phases with 100% duty cycle (the HSS of all phases will be opened)
Active short high low	Closing the LSS of all phases with 50% duty cycle, this means the HSS are open and closed with 50% duty cycle

5 Configuration of the static parameters

Active short high	Closing the HSS of all phases with 100% duty cycle (the LSS of all phases will be opened)
-------------------	---

5.5 MCTRL Hardware Abstraction

The MCTRL Hardware Abstraction Layer Library contains the following software components:

- Measurement ADC
- Bridge Driver
- Pattern Generator

5.5.1 Measurement ADC

This software component reads the DC link voltage and the phase current from the hardware measurement module (current sense amplifier), removes the current measurement offset, and scales the values by the respective base value. In addition to that, it can be used to calibrate the current offset during the initialization phase.

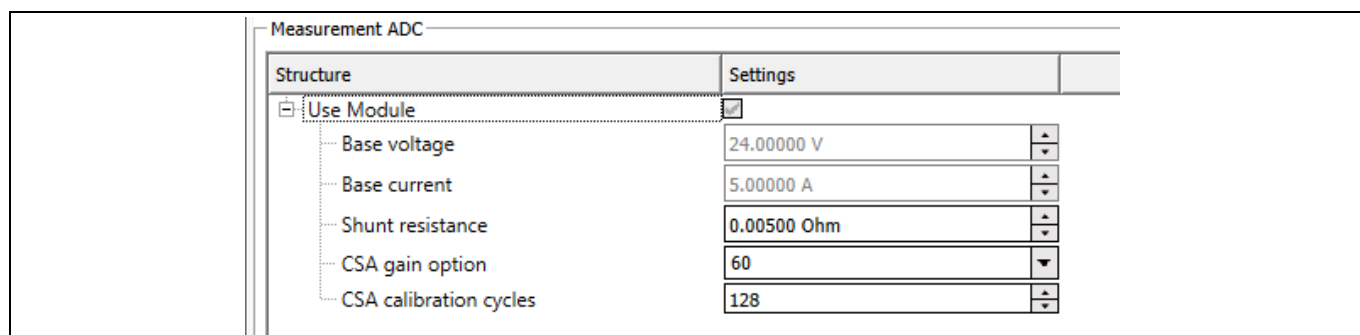


Figure 70 MCTRL Hardware Abstraction – Measurement ADC

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
Shunt resistance	Value of the shunt resistor used to measure the current, in FOC single-shunt sensorless configuration		0.005		Ohm
CSA gain option	Value of the CSA gain, it depends upon the shunt resistance and the current consumption		60		
CSA calibration cycles	Value of the calibration cycles, which is the number of times a function loop can be executed. This loop is used to calculate the ADC offset		128		

The operating differential input voltage range of the TLE987x CSA is $\pm 1.5 \text{ V/G}$, where G is the CSA gain. The value of the shunt resistor as well as of the CSA gain must be adequately selected. For further information, the user can refer to the [TLE987x User Manual](#) and the [TLE987x/6x Hardware Design Application Note](#).

5 Configuration of the static parameters

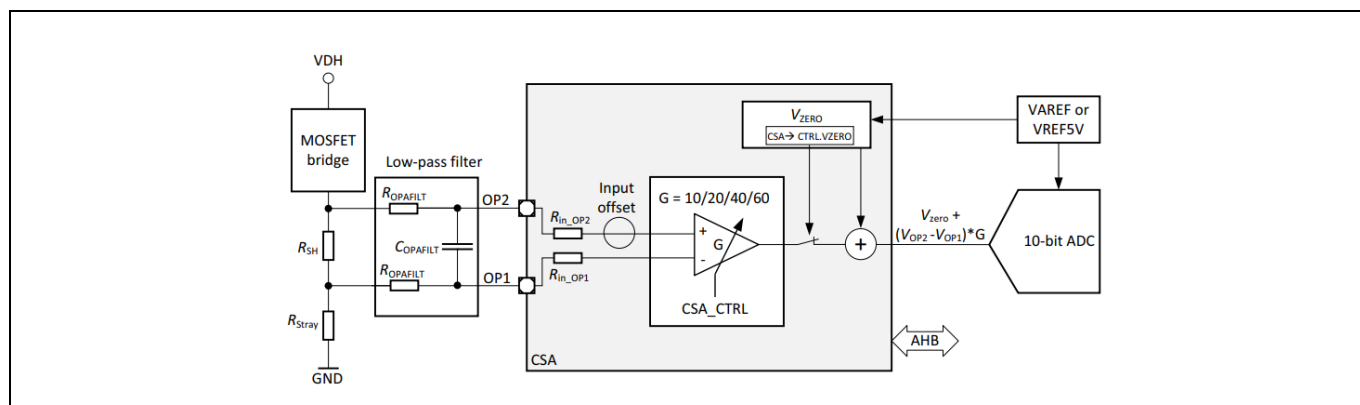


Figure 71 TLE987x current sense amplifier block diagram

5.5.2 Pattern generator

This software component sets the timer compare values needed by the hardware module that drives the switches of the bridge driver, as well as the timer compare values needed by the hardware module to trigger a current measurement. The timer compare values are calculated and provided by the modulator. The Pattern Generator can be configured to set the PWM frequency, the dead-time of the system and to react on a trap fault.

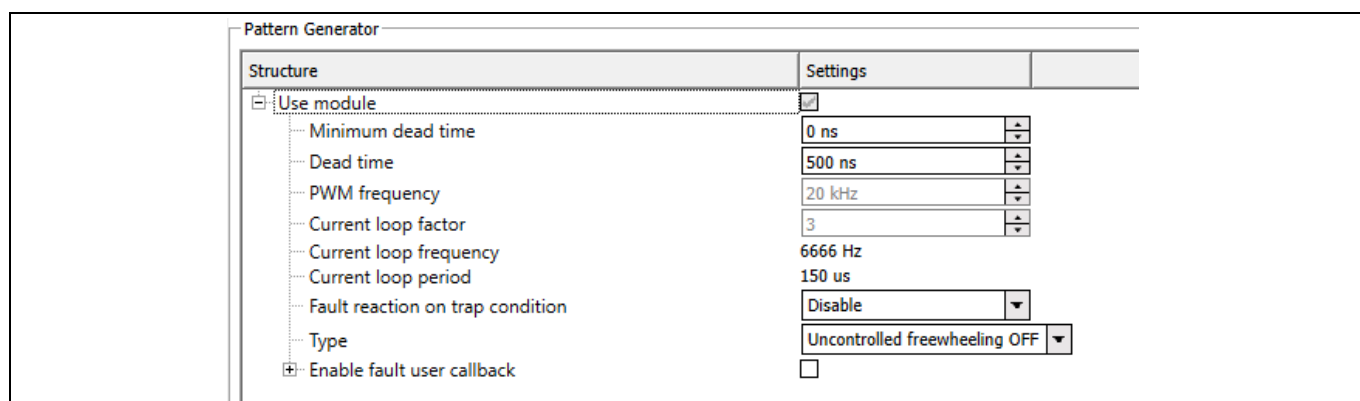


Figure 72 MCTRL Hardware Abstraction – pattern generator

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
Minimum dead time	Lower limit of the dead-time of the switching mosfets		0		ns
Deadtime	Initialization value for the dynamic configuration Dead-time.		500		ns
Fault reaction on trap condition	Fast emergency stop without CPU load via external signal (CTRAP). It can be configured to one of the options: Disabled, Enabled, Report only, Report and react		Disabled		
Type	If the fault reaction on trap condition is configured to Report and react, then this is immediate hardware reaction (if any of the faults occurred)		Uncontrolled freewheeling OFF		

5 Configuration of the static parameters

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
Fault user callback	If enabled, name of the user-defined callback in which a custom-defined reaction can be programmed in case of modulator error		usrFaultCallback		

5.5.3 Bridge driver

This software component provides the management of the hardware faults detected by the BDRV HW module. Fault reaction on overcurrent, overvoltage, undervoltage are defined in [Chapter 8.1.2](#).

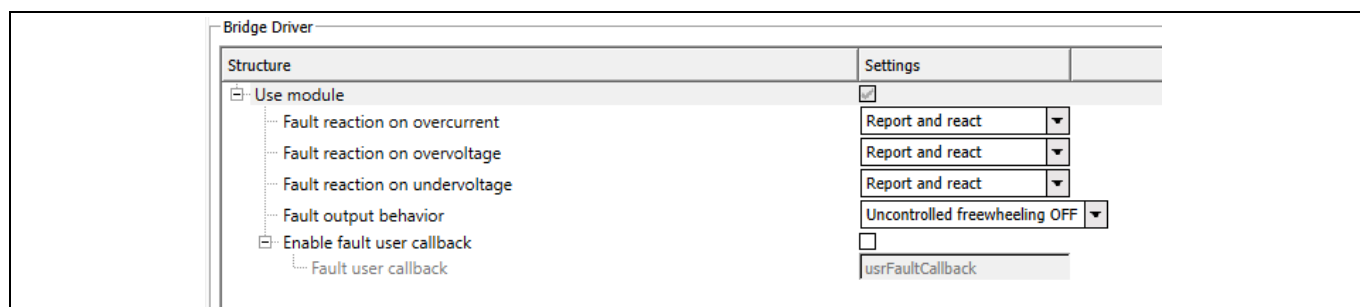


Figure 73 MCTRL Hardware Abstraction – bridge driver

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
Fault reaction on overcurrent	Fault reaction if the maximum current of a switch is exceeded		Report and react		
Fault reaction on overvoltage	Fault reaction in case of overvoltage		Report and react		
Fault reaction on undervoltage	Fault reaction in case of undervoltage		Report and react		
Fault user callback	If enabled, name of the user-defined callback in which a custom-defined reaction can be programmed in case of modulator error		usrFaultCallback		

5.6 Math library

The Math Library provides optimized mathematical functions using the Q number format described in Q-format concept, like for example Clarke and Park transformation, or a group of similar operations, for instance multiplication of 16 bits and of 32 bits variables with or without saturation.

Math Library Configurations		
Structure	Settings	
PI controllers discretization method	Trapezoidal	▼
PI controllers anti-windup method	Clamp	▼
PLL delay length	8 cycles	▼
Cartesian to Polar table size	10 bits	▼
Sine table size	10 bits	▼
Atan table size	10 bits	▼

Figure 74 Math Library Configurations – structure

Parameter name	Description	Values			Unit
		Min.	Default	Max.	
PI controllers discretization method	The user can select one of the three methods of discretization implemented in software (trapezoidal, backward Euler, forward Euler). Details about discretization methods can be found in the MathWorks® website		Trapezoidal		
PI controllers anti-windup method	The anti-windup method is used to reduce the windup (increasing) of error in the PI integral. Two methods are implemented in software to reduce the integral error windup (clamp and back calculation).		Clamp		
PLL delay length	Number of execution steps (cycles) between two angle measurements used to calculate the speed (higher implies more delay with more filtering)		8		cycles
Cartesian to Polar table size	Size of the look-up table used by cartToPolar function		10		Bits
Sine table size	Size of the look-up table used by sine function		10		Bits
Atan table size	Size of the look-up table used by atan function		10		Bits

5.7 TLE987x peripherals

Details about the configuration of the TLE987x can be found in the “[TLE987x User Manual](#)” and “[TLE987x/6x Hardware Design Application Note](#)”

5 Configuration of the static parameters

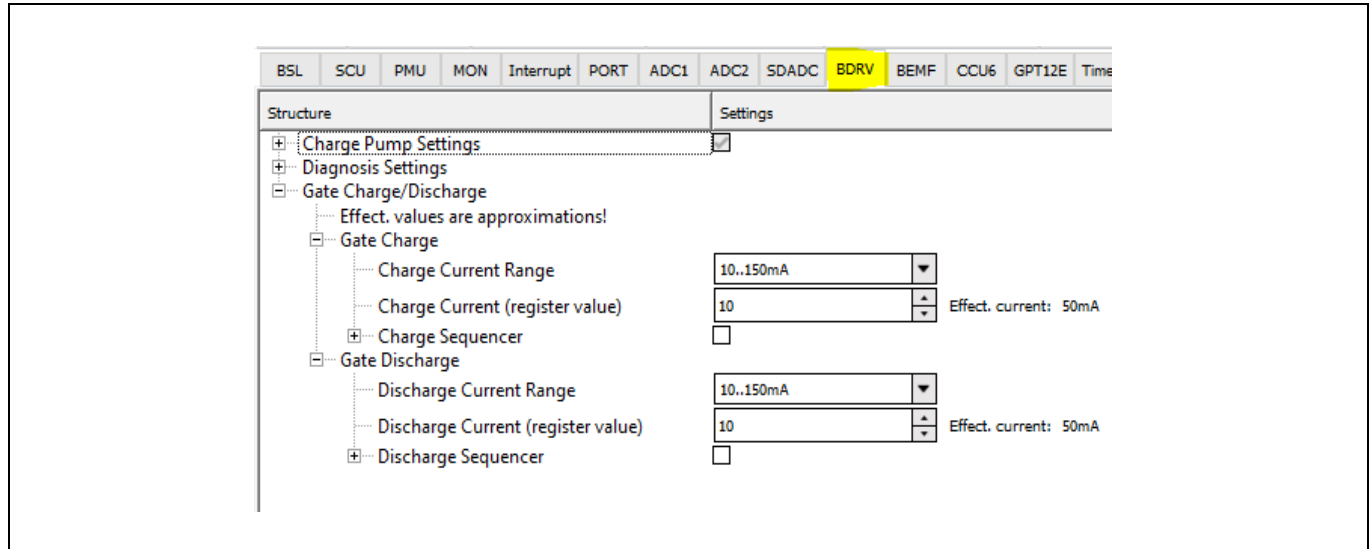
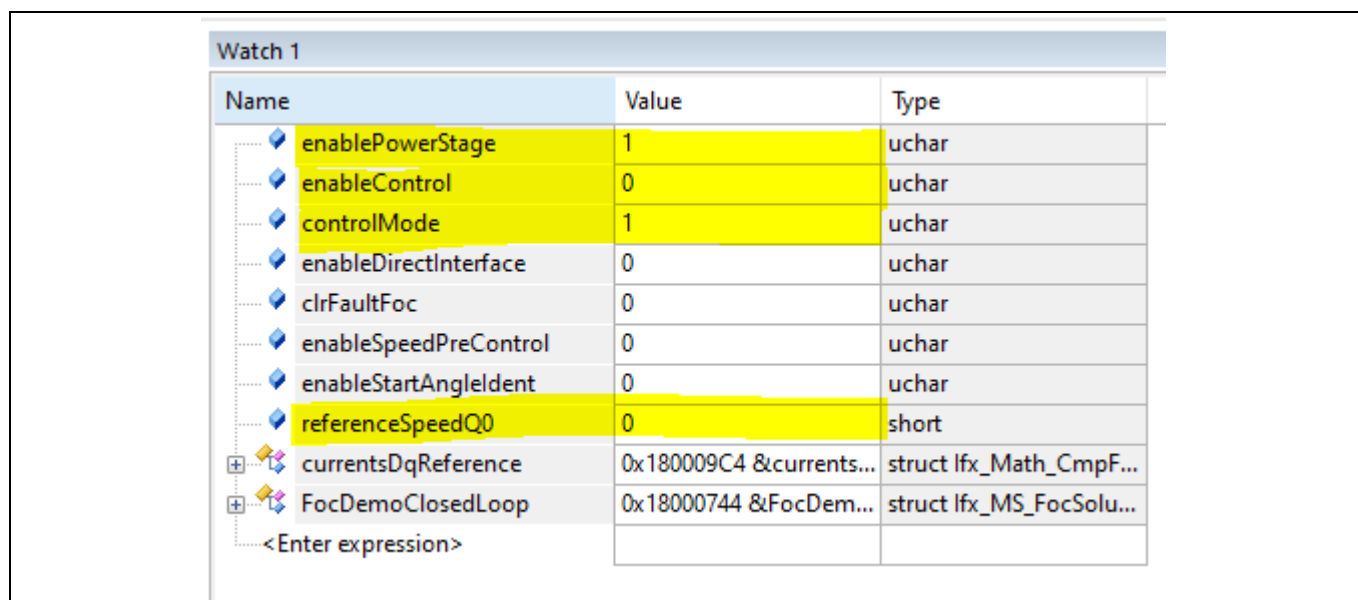


Figure 75 Static parameters of TLE987x - BDRV

6 Dynamic parameters for the user

Dynamic parameters are parameters that can be changed while executing the program. These parameters can be configured using a debugger or using tools like the MOTIX™ Micro Inspector Pro and MOTIX™ Solution Designer, available in the Infineon Developer Center Launcher. The Figure 76 shows an example of the available dynamic parameters, such as the reference speed, the enable of the power stage and so on.



Name	Value	Type
enablePowerStage	1	uchar
enableControl	0	uchar
controlMode	1	uchar
enableDirectInterface	0	uchar
clrFaultFoc	0	uchar
enableSpeedPreControl	0	uchar
enableStartAngleIdent	0	uchar
referenceSpeedQ0	0	short
currentsDqReference	0x180009C4 ¤ts...	struct Ifx_Math_CmpF...
FocDemoClosedLoop	0x18000744 &FocDem...	struct Ifx_MS_FocSolu...
<Enter expression>		

Figure 76 Dynamic parameters in Keil debugger

6.1 Functional flow chart

The following dynamic parameters are influencing the state machine of the FOC Solution (Figure 2).

- **Enable power stage: 1**
If power stage is enabled, the state machine goes into standby mode, where modulator is set to zero and resets all the previous values.
- **Enable control: 1**
If the control is enabled, the state machine enters into run mode, where FOC or VToF will be executed.
- **Control mode: 1**
If control mode is 1, the state machine executes FOC in run mode. If the control mode is 0, it executes VToF in run mode.

7 Actions needed to run the new motor with MOTIX software

This chapter shows the actions that are needed to run the motor with the FOC software. The tool installations and details of the Keil Project can be found in the Getting Started Guide [2].

The software is developed with Keil compiler V6.13. By default, if user downloads Keil from the webpage, the user gets the latest Keil compiler version. It is recommended to use the compiler version V6.13.

If user is using the old compiler version (for example, compiler 5.06), then user needs to change the compiler to V6.13 or the latest versions. How to change from one compiler to another can be found in [Keil](#).

The Micro Inspector Pro (MIP) tool support for MOTIX IC's is stopped from Infineon. If the user still wants to use MIP tool then

- the user with Keil compiler version 5 can use MIP tool with out any issues. User can use both debugger and oscilloscope feature.
- the user with Keil compiler version 6 can enter **-gdwarf-3** in the compiler (Misc Controls). User can use debugger but Oscilloscope feature will be disabled.

*Note: For the better performance user can use **-O3 optimization** and enable the **Link – Time Optimization (LTO)** as shown in Figure 77. The users with Keil license can enable LTO.*

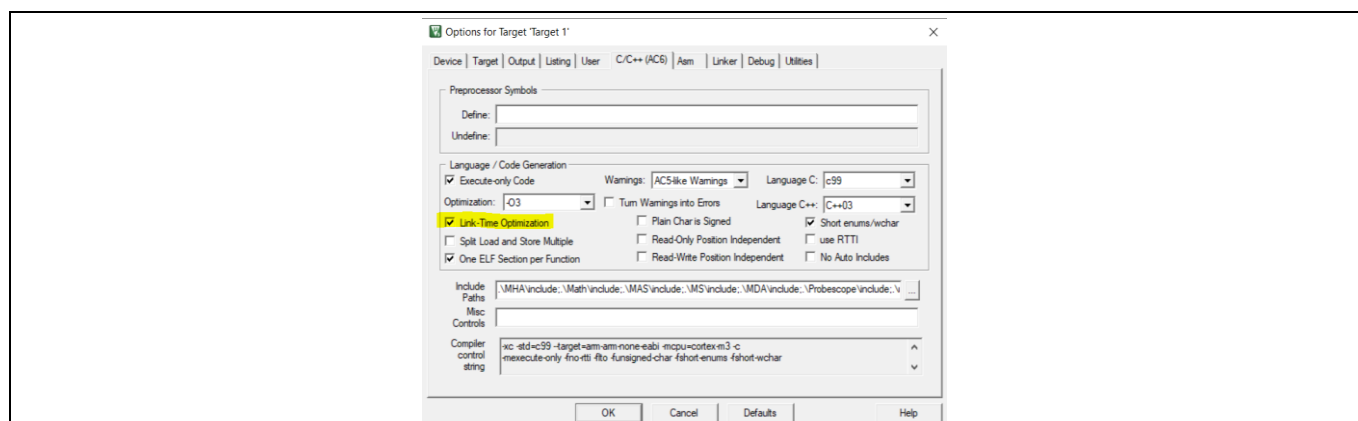


Figure 77 Keil Compiler option

Comparison of execution time between different Keil compiler versions

X axis - Different parameters enabled and disabled in config Wizard,

Y axis - Execution time in microseconds (us)

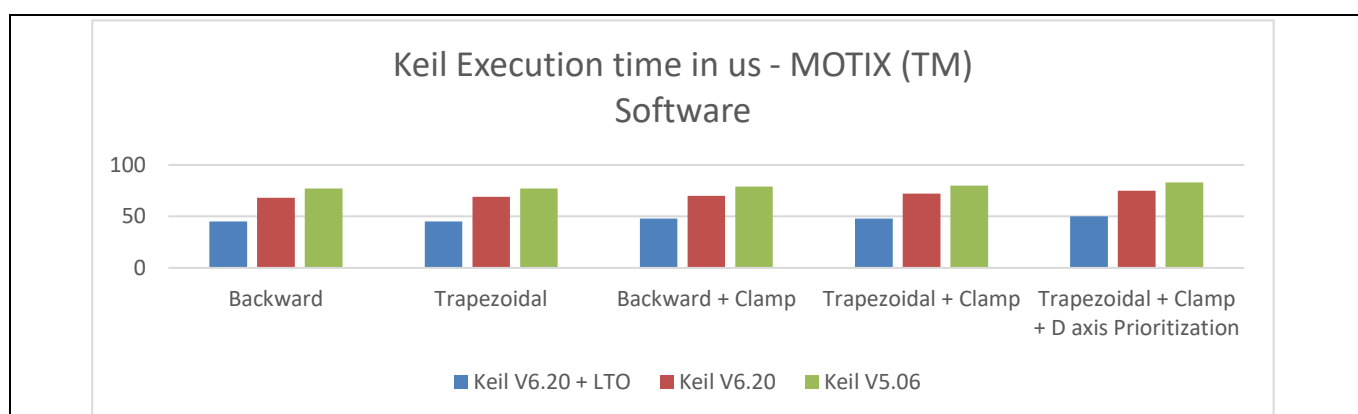


Figure 78 Execution time comparison

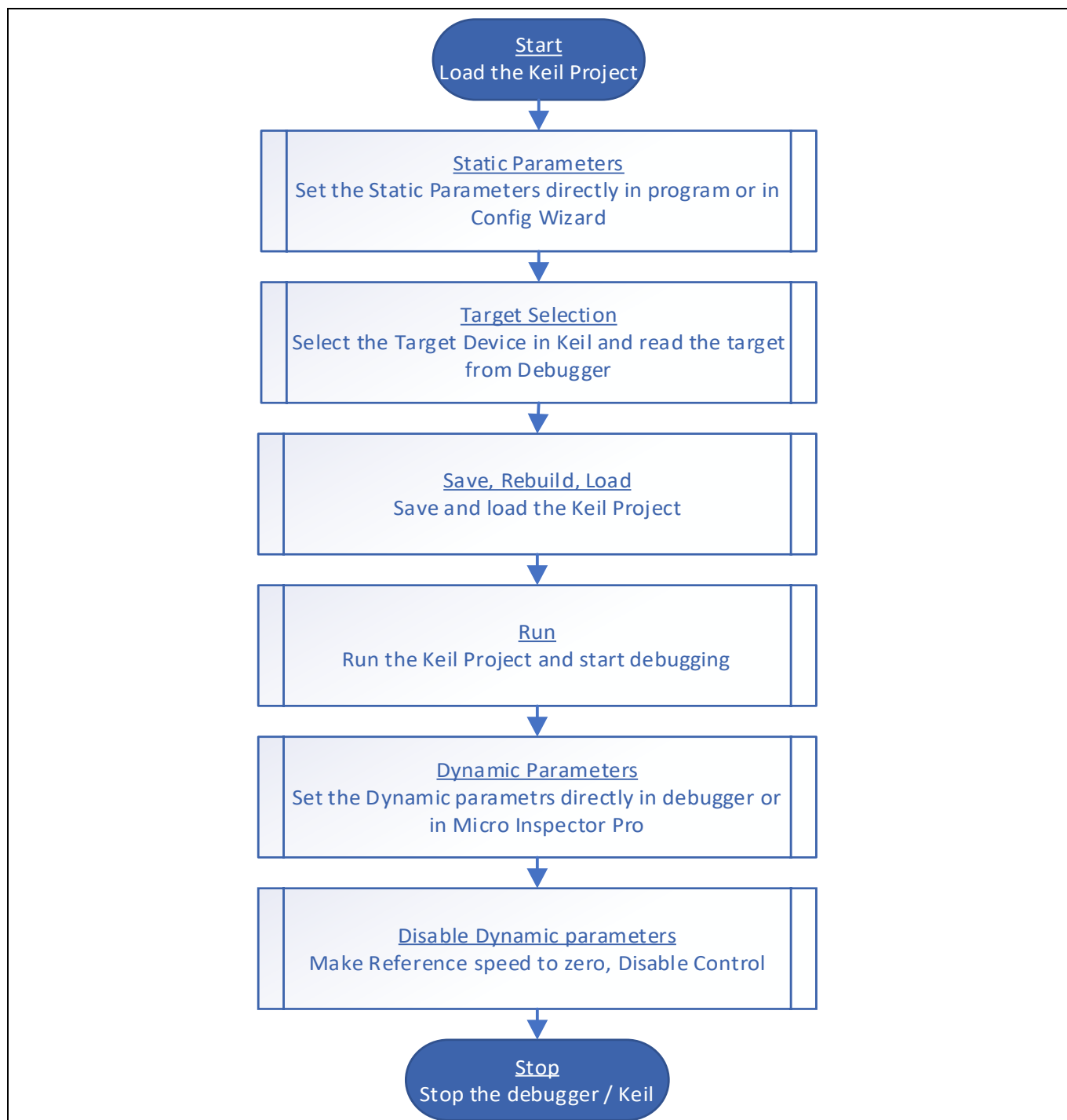


Figure 79 Steps to run the motor

7.1 Minimum parameters configuration to spin a motor

The following static and dynamic parameters are the mandatory ones to configure, in order make a motor spin.

- **Static Parameters:**
 - System base voltage
 - System base current
 - System base mechanical speed
 - Phase resistance

Sensorless FOC motor control using Infineon MOTIX™ TLE987x

7 Actions needed to run the new motor with MOTIX software

- Phase inductance
- Number of pole pairs
- Startup current
- Shunt resistance
- CSA gain option
- **Dynamic Parameters:**
 - Enable power stage
 - Enable control
 - Reference speed
 - Control mode

7.2 Example of a Nanotec motor with no load

The motor used in this example is the Nanotec DB42S03. The parameters have been configured in the Config Wizard to run the motor, in no load condition.

The tool installation and details of the Keil project can be found in the Getting Started Guide [2].

The details of the Nanotec motor can be found in the [Nanotec DB42S03 datasheet](#).

SPECIFICATION	CONNECTION	DELTA
NO. OF POL./PHASE		8/3
VOLTAGE RATED (VDC)		24
CURRENT NO LOAD/RATED/PEAK (AMP)		0.2/1.79/5.4
RESISTANCE/PHASE TO PHASE (Ohms) @25°C		1.5±15%
INDUCTANCE/PHASE TO PHASE (mH) @1KHz		2.1±20%
TORQUE RATED/PEAK (Nm) [lb-in]		0.0625/0.19 [0.553/1.68]
TORQUE CONSTANT (Nm/A)		0.035 ⚠ ⚠
POWER RATED (W)		26
SPEED RATED/NO LOAD (RPM) ⚠		4000/6200
ROTOR INERTIA (Kg-m²) [lb-in²]		2.4x10 ⁻⁶ [8.2x10 ⁻³]
WEIGHT (Kg) [lb]		0.3 [0.661]
TEMPERATURE RISE: MAX.80°C		
AMBIENT TEMPERATURE -10°~ 50°C [14°F ~ 122°F]		
INSULATION RESISTANCE 100 MOhm (UNDER NORMAL TEMPERATURE AND HUMIDITY)		
INSULATION CLASS B 130° [266°F]		
DIELECTRIC STRENGTH 500VAC FOR 1 MIN. (BETWEEN THE MOTOR COILS AND THE MOTOR CASE)		
AMBIENT HUMIDITY MAX. 85% (NO CONDENSATION)		

Figure 80 Nanotec DB42S03 datasheet

Mandatory static parameters:

- **System base voltage:** 24 V
The maximum voltage for scaling, higher than the supply voltage.
- **System base current:** 5 A
The maximum current for scaling, higher than the rated current.
- **System base mechanical speed:** 4000 rpm
- **Phase resistance:** $0.75000 \Omega \left(\frac{1.5 \Omega}{2} \right)$
- **Phase inductance:** $1.05000 \text{ mH} \left(\frac{2.1 \text{ mH}}{2} \right)$
- **Number of pole pairs:** 4

Sensorless FOC motor control using Infineon MOTIX™ TLE987x

7 Actions needed to run the new motor with MOTIX software

- **Startup current:** 1 A

The start-up current amplitude should be big enough to start the motor, it can be adjusted depending on the load. It has to be checked at which current the motor starts rotating in IToF open loop. At no load, 1A is good enough as a reference current for this motor.

- **Startup current ramp up:** 4 A/s (default)
- **Shunt resistance:** 0.005 Ω
- **CSA gain option:** 60

Additional static parameters:

- **Rotor inertia:** 2.4 mgm²
(Here rotor means “everything that rotates”, not only the rotor of the motor, but also the load, e.g. fan, needs to be considered)
- **Torque constant:** 0.035 Nm/A

Mandatory dynamic parameters:

- **Enable power stage:** 1
- **Enable control:** 1
- **Reference speed:** 0 to 2000 rpm
- **Control mode:** 1

Note: the static and dynamic parameters shall be adapted for other motor types

Result

The motor has been tested successfully up 2000 rpm with no load.

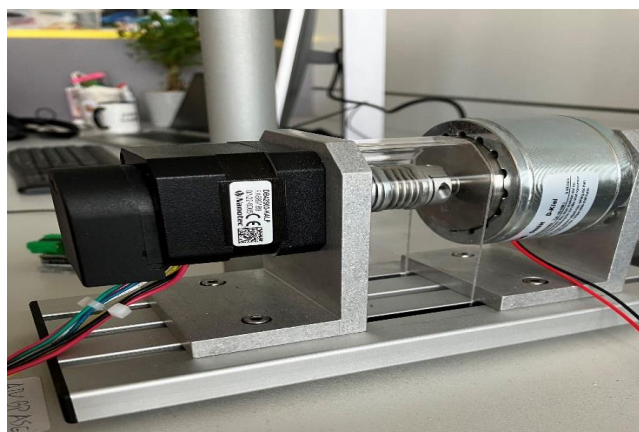


Figure 81 DB42S03 motor

7.3 Example with MOTIX™ 1 kW Engine Cooling Fan reference design

This reference design is an automotive 3-phase motor drive for an [engine cooling fan](#) application, using MOSFETs with gate charge > 100 nC.

Mandatory static parameters:

- **Current loop factor:** 3 to 2

Sensorless FOC motor control using Infineon MOTIX™ TLE987x

7 Actions needed to run the new motor with MOTIX software

Decreasing the current loop factor, i.e. running the FOC solution more often, leads to a better performance. Decreasing the current loop factor is application dependent.

- **System base voltage:** 24 V

The maximum voltage for scaling, higher than the supply voltage.

- **System base current:** 50 A to 80 A

Considering 1 kW at 12 V supply, the maximum current should be 80 A, so the base current should be higher than that.

- **System base mechanical speed:** 7000

- **Phase resistance:** 0.01000 Ω

- **Phase inductance:** 0.03000 mH

- **Startup current:** 15 A

The start-up current amplitude should be big enough to start the motor, it can be adjusted depending on the load. It has to be checked at which current the motor starts rotating in IToF open loop.

- **Startup current ramp up:** 60 A/s (default)

The start-up ramp rate is used for smooth acceleration. By ramping up the start-up current slowly a steady behavior can be seen.

- **Current control adjustment factor:** 0.40

This parameter is used to control the response of the controller. Sometimes for higher load, the adjustment factor must be high.

- **Speed PI, proportional gain:** 0.012 (changed from 0.003)

- **Speed PI, integral gain:** 0.004 (changed from 0.001)

The proportional and integral gains shall be adjusted to achieve the ramp up time of the speed.

- **Speed PI, output lower limit:** -70 A

- **Speed PI, output upper limit:** 70 A

- **Maximum speed:** 6000 rpm

- **Minimum speed:** -6000 rpm

- **Open loop ramp up rate:** 1000 to 100 rpm/s

- **Open loop ramp down rate:** 1000 to 100 rpm/s

Due to the high load, it is difficult to ramp up and ramp down with high values. The open loop ramp rate shall be tuned to smaller values.

- **Limit voltage vector with D axis prioritization (Checkbox):** enabled

Anti-windup status takes DC link voltage into account and checks the PI values to be in specified limits.

- **FOC controller (Id, Iq), output lower limit:** -10

- **FOC controller (Id, Iq), output upper limit:** 10

- **Shunt resistance:** 0.00076 Ω

- **CSA gain option:** 20

- **PI controllers discretization method:** backward Euler

- **PI controllers anti-windup method:** Clamp

- **Cartesian to polar table size:** 10 to 12 bits

- **Sine table size:** 10 to 12 bits

- **Atan table size:** 10 to 12 bits

Changed from 10 to 12 bits for the better results, supported only by the licensed version of Keil

- **Gate charge, charge current range:** 20 mA to 300 mA

Sensorless FOC motor control using Infineon MOTIX™ TLE987x

7 Actions needed to run the new motor with MOTIX software

- **Gate charge, charge current (register value):** 22
- **Gate discharge, discharge current range:** 20 mA to 300 mA
- **Gate discharge, discharge current (register value):** 22

Mandatory dynamic parameters:

- **Enable power stage:** 1
- **Enable control:** 1
- **Reference speed:** 0 to 4000
- **Control mode:** 1

Note: These static and dynamic parameters might be changed for other motors and with the other loads

Result

The reference design has been tested successfully till 800 W at 16 V with 50 A motor current, reaching a speed of around 3800 rpm.

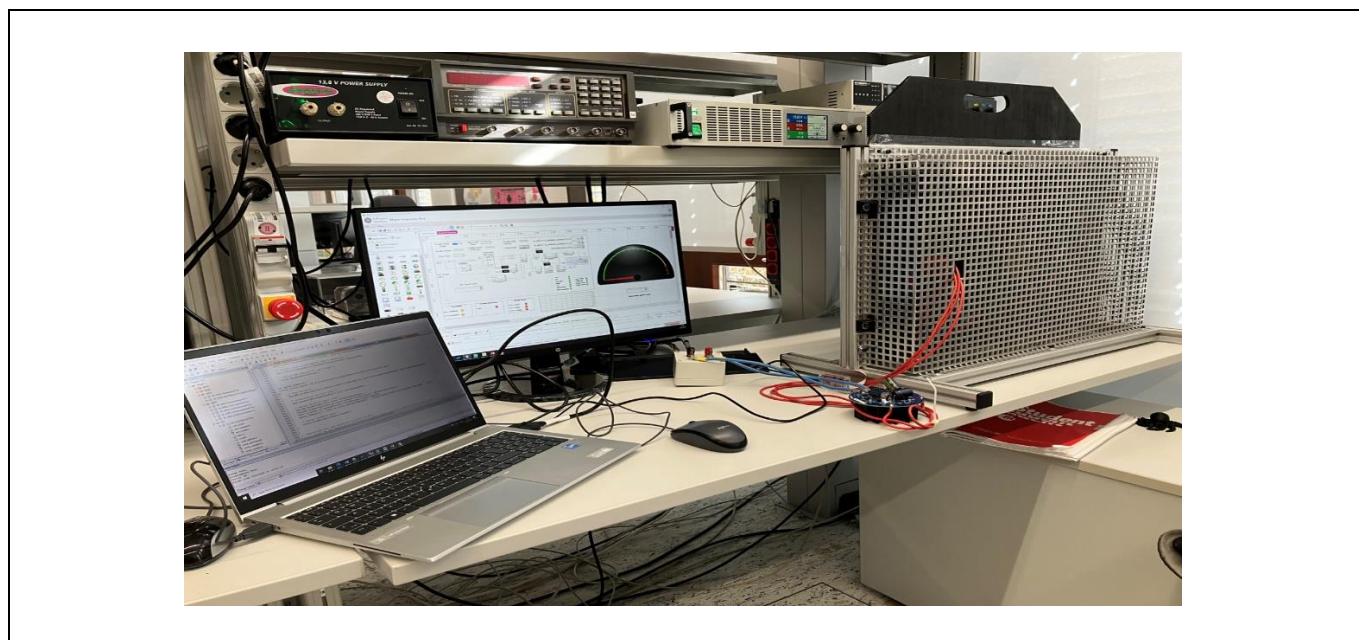


Figure 82 Engine cooling fan with reference design

7.3.1 Challenges when turning a motor for the first time

In case the motor does not start or stops suddenly, it may be due to multiple reasons, like for example:

- Software execution, so the user needs to check whether the program enters into FOC execution by checking the status flag or by putting a break point in the program and check the execution
- Insufficient current supply, due to the limitation of external power supply
- Low start-up current configured in the Config Wizard
- Low current PI adjustment factor
- High ramp up and ramp down values instead of low values

Sensorless FOC motor control using Infineon MOTIX™ TLE987x

7 Actions needed to run the new motor with MOTIX software

For example: a power supply limitation can be monitored by comparing the dqCommand_Imag with the currentDQ_Imag.

dqCommand_Imag: it is the reference current to the PI regulator. Depending on the user reference speed, the software calculates the reference current.

currentDQ_Imag: it is the real current measured from the ADC.

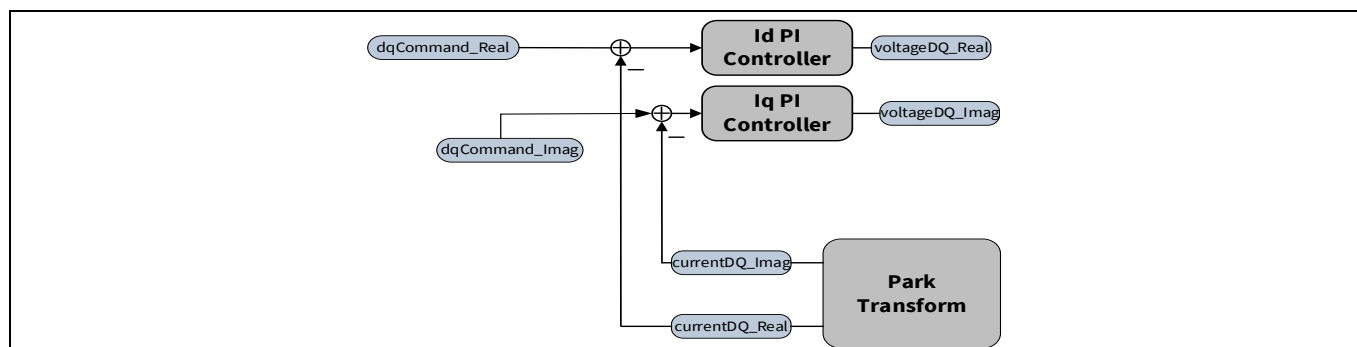


Figure 83 Reference q current

As an example, in the debugger window:

Name	Value	Type
FocDemoClosedLoop	0x18000744 &FocDem...	struct Ifx_MS_FocSolu...
focController	0x18000744 &FocDem...	struct Ifx_MDA_FocCo...
currentDPi	0x18000744 &FocDem...	struct Ifx_Math_PiF16
currentDQ	0x18000760	struct Ifx_Math_CmpF...
real	0	short
imag	0	short
currentQPi	0x18000764	struct Ifx_Math_PiF16
dqDecoupling	0x18000780	struct Ifx_Math_DqDe...
p_output	0x18000784	struct Ifx_MDA_FocCo...
voltageDQ	0x1800078C	struct Ifx_Math_CmpF...
p_dAntiwindupCtrl	0	char
p_qAntiwindupCtrl	0	char
modulator	0x18000794	struct Ifx_MAS_Modul...
patternGenTLE987	0x180007D4	struct Ifx_MHA_Patter...
bridgeDrvTLE987	0x18000800	struct Ifx_MHA_Bridge...
measurementADCTLE987	0x18000808	struct Ifx_MHA_Meas...
fluxEstimator	0x1800082C	struct Ifx_MDA_FluxEs...
iToF	0x1800089C	struct Ifx_MDA_IToFC...
speedPreControl	0x180008AC	struct Ifx_Math_Speed...
p_refCurrentLimit	0x180008B8	struct Ifx_Math_LimitF...
p_status	0x180008BC	struct Ifx_MS_FocSolu...
speedLimit	0x180008C0	struct Ifx_Math_LimitF...
accelerationLimit	0x180008C4	struct Ifx_Math_Accel...
speedPi	0x180008D0	struct Ifx_Math_PiF16
startAngleIdent	0x180008EC	struct Ifx_MDA_StartA...
vToF	0x18000948	struct Ifx_MDA_VToFC...
p_startCurrentRateLimit	0x18000964	struct Ifx_Math_RateLi...
currentsUVW	0x1800096A	struct Ifx_Math_3Phas...
currentsAlphaBeta	0x18000970	struct Ifx_Math_CmpF...
voltageAlphaBeta	0x18000974	struct Ifx_Math_CmpF...
previousVoltageAlphaB...	0x18000978	struct Ifx_Math_CmpF...
dqCommand	0x1800097C	struct Ifx_Math_CmpF...
real	0	short
imag	0	short
angle	0	uint
p_transitionAngleTolera...	1820	int
p_speedRampUpRateQ...	402921	int

Figure 84 q current comparison in Keil debugger window

If the SW is not able to start in FOC mode, then the user could change the control mode to VToF mode and check whether the motor rotates.

Watch 1		
Name	Value	Type
enablePowerStage	1	uchar
enableControl	1	uchar
controlMode	1	uchar
enableDirectInterface	0	uchar
clrFaultFoc	0	uchar
enableSpeedPreControl	0	uchar
enableStartAngleIdent	0	uchar
referenceSpeedQD	1000	short
currentsDqReference	0x18000A30 & currentsDqReference	struct Ifx_Math_CmpF...
FocDemoClosedLoop	0x1800078C &FocDemoClosedLoop	struct Ifx_MS_FocSolu...
focController	0x1800078C &FocDemoClosedLoop	struct Ifx_MDA_FocCo...
modulator	0x180007DC	struct Ifx_MAS_Modul...
patternGenTLE987	0x1800081C	struct Ifx_MHA_Patter...
bridgeDrvTLE987	0x18000848	struct Ifx_MHA_Bridge...
measurementADCTLE987	0x18000850	struct Ifx_MHA_Meas...
fluxEstimator	0x18000874	struct Ifx_MDA_FluxEs...
iToF	0x18000904	struct Ifx_MDA_IToF...
speedPreControl	0x18000914	struct Ifx_Math_Speed...
p_refCurrentLimit	0x18000920	struct Ifx_Math_LimitF...
p_status	0x18000924	struct Ifx_MS_FocSolu...
state	4 Ifx_MS_FocSolutionF16_State_run	enum (uchar)
subState	2 Ifx_MS_FocSolutionF16_SubState_closedLoop	enum (uchar)
actualControlMode	1 Ifx_MS_FocSolutionF16_ControlMode_foc	enum (uchar)

7.4 Methods to determine the motor parameters

The parameters of a motor can be generally found in the datasheet. If they are not included in the documentation, then there are some other methods to measure the parameters.

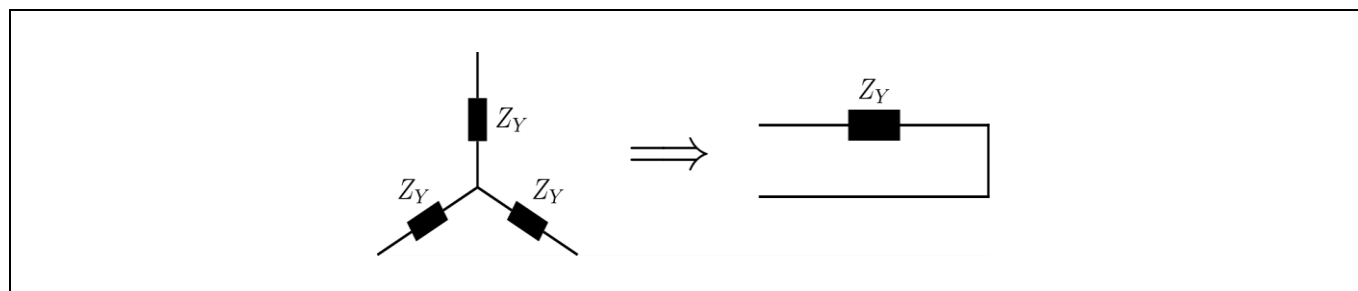


Figure 85 Motor windings

Depending on the known parameter, the parameters needed by FOC can be calculated as follows.

(1) If the phase impedance is known

- If a motor is Y-connected, the phase impedance Z_Y can be directly used in FOC
- If a motor is delta-connected, the phase impedance is Z_{delta} needs to be converted into an equivalent Y-connection (see [Y-Δ transform in Wikipedia](#))

$$Z_Y^{equivalent} = \frac{1}{3} \times Z_{delta}$$

(2) If the phase impedance is unknown, the parameter needed by FOC can be calculated by measuring line-to-line impedance Z_{LL}

In case of Y-connection, the phase impedance is calculated as:

$$Z_Y = \frac{1}{2} \times Z_{LL}$$

In case of delta-connection, the phase impedance is calculated as:

$$Z_{delta} = \frac{3}{2} \times Z_{LL}$$

According to the conversion mentioned in 2:

7 Actions needed to run the new motor with MOTIX software

$$Z_Y^{equivalent} = \frac{1}{3} \times \frac{3}{2} Z_{LL} = \frac{1}{2} \times Z_{LL}$$

Which is equal to the case of Y-connection.

Independently if a motor is delta connected or Y-connected, half of the line-to-line value can be used for FOC.

7.4.1 Phase resistance

Use a multimeter to measure the winding resistance. Measure the resistance between the two phases of a DC motor.

For example: between phase U and phase V or phase V and phase W or phase W and phase U (all must be the same). The resistance is the line-to-line (l-l) resistance, if the BLDC motor is WYE connected. In order to obtain the phase resistance, divide the resistance (l-l) by 2.

7.4.2 Phase inductance

Two methods can be used to calculate the phase inductance.

- **First method:** use a low voltage AC source connected to the motor winding to measure the inductance of the motor. Apply the ac voltage to the armature winding of a dc motor. Apply the ac voltage to one pair of the three wires of a BLDC motor. Measure the voltage and current in both scenarios. Keep in mind that the BLDC motor is often wired in a WYE. Line-to-line readings will be the result. Divide the voltage by two if you want the phase values for the voltage. AC frequency can be around 60 Hz.

$$\text{BLDC motor winding impedance [ohms]} = \frac{\text{Phase voltage [volts]}}{\text{Line current [amps]}}$$

$$\text{Reactance [vars]} = \sqrt{\text{Impedance}^2 - \text{Phase resistance}^2}$$

$$\text{Inductance [henry]} = \frac{\text{Reactance [vars]}}{2\pi \times \text{frequency [Hz]}}$$

- **Second method:** use an LCR meter to measure the inductance between two phases. User can check the inductance of two phases and change to the other two phases. For example, UV, VW, WU and take average of them as the inductance.

7.4.3 Pole pairs

There are some methods to calculate the pole pairs of a BLDC motor.

- **First method:** the easiest way is to count the magnets manually. This is only possible if the user can see into the motor or if the outer casing can be removed. The poles pairs will be the number of magnets divided by 2.
- **Second method:** the user should connect the power supply to two phases of a motor and turn on the power supply, limiting the supply current. The motor will generate electromagnetic force in the stator, and rotor will try to touch it. It is also important to choose the proper current, so that the user should feel only the torque between the positions. A stable position is reached when the rotor is fixed and aligned with the stator field, so it does not move anymore. Moving manually the rotor, the user can count how many stable positions (pole pairs) can be reached for one complete mechanical rotation.

7 Actions needed to run the new motor with MOTIX software

- **Third method:** frequency and speed also can be useful to find out the pole pairs, due to the following equation:

$$Pole\ pairs = \frac{Stator\ frequency \times 60}{Speed\ in\ rpm}$$

8 Fault types, configuration and indication

This chapter provides the information about the fault's types and the different methods of configuration of the faults.

8.1 Types of fault

There are two types of faults:

- Software fault
- Hardware fault

8.1.1 Software fault

The software faults can occur due to the configuration of SW static parameters, either directly by writing macros or by the Config Wizard. The main software faults are:

- Maximum amplitude
- Over modulation
- Anti-windup saturation status

8.1.2 Hardware fault

A hardware fault can occur due to external factors, the bridge driver sends the signals to ADC2. ADC2 in TLE987x is used to check the fault detection of the bridge driver when there is:

- **Overcurrent (OC)**
This configuration defines the fault reaction if the maximum current of a switch is exceeded
- **Overvoltage (OV)**
This configuration defines the fault reaction if any of the following measured voltage level exceeded the configured upper threshold:
 - ADC2 channel 2 "VSD upper threshold", configurable in the range from 3 V to 30 V
 - ADC2 channel 3 "VCP upper threshold", configurable in the range from 5 V to 53 V
- **Undervoltage (UV)**
This configuration defines the fault reaction if any of the following measured voltage levels drops below the configured lower threshold:
 - ADC2 channel 2 "VSD lower threshold", configurable in the range from 3 V to 30 V
 - ADC2 channel 3 "VCP lower threshold", configurable in the range from 5 V to 53 V
 - BDRV charge pump diagnosis VCP low voltage threshold, configurable as one of the following:
 - 7.325 V
 - 7.645 V
 - 7.982 V
 - 8.309 V
 - 8.638 V
 - 8.966 V
 - 9.293 V
 - 9.620 V

8 Fault types, configuration and indication

The critical levels of OC, OV, UV used in software are defined in [1]. For example: if OC, OV, UV occurs, then the software disables all the PWM channels, but the bridge driver is still active.

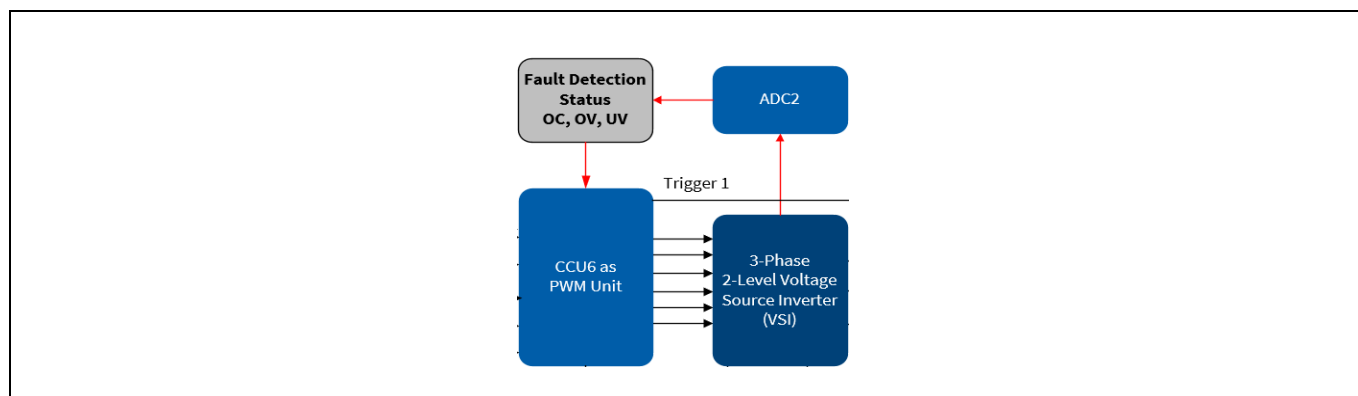


Figure 86 Block diagram for bridge driver fault detection

TRAP condition: the trap can be configured in the CTRAP register. It indicates fast emergency stops without CPU load via external signal (CTRAP). The TRAP pin of TLE987x is used as an input pin where the user can enable the pin to deactivate the PWM outputs. Further details can be found in the TLE987x user manual [1].

8.2 Fault configuration

In the software, the fault check status function is called continuously. Fault detection options can be configured from the Config Wizard. The following options are:

Table 2 Fault configuration options

Name in the configuration tool	If the fault occurs
Disable	<ul style="list-style-type: none"> It will not be detected It will not be stored It will not be reported It will not trigger any reaction
Enable	<ul style="list-style-type: none"> It will be detected It will be stored internally to a flag in the “status” output It will not be reported It will not trigger any reaction
Report only	<ul style="list-style-type: none"> It will be detected It will be stored internally to a flag in the “status” output It will trigger the execution of a function callback where user code can be inserted It will not trigger any reaction
Report and react	<ul style="list-style-type: none"> It will be detected It will be stored internally to a flag in the “status” output It will trigger the execution of a function callback where user code can be inserted It will trigger the designed reaction

8 Fault types, configuration and indication

8.3 Indication of faults

The faults can be seen by the user easily either by checking debug window in Keil or also in the Micro Inspector Pro.

8.3.1 Indication for faults in the debug watch window

There are fault status variables, which the user can monitor from the watch window. If the fault can be ignored, then the clear fault option can be selected.

Name	Value	Type
referenceSpeedQ0	0	short
currentsDqReference	0x180009C4 ¤tsDqReference	struct Ifx_Math_CmpF...
FocDemoClosedLoop	0x18000744 &FocDemoClosedLoop	struct Ifx_MS_FocSolu...
focController	0x18000744 &FocDemoClosedLoop	struct Ifx_MDA_FocCo...
currentDPi	0x18000744 &FocDemoClosedLoop	struct Ifx_Math_PiF16
currentDQ	0x18000760	struct Ifx_Math_CmpF...
currentQPi	0x18000764	struct Ifx_Math_PiF16
dqDecoupling	0x18000780	struct Ifx_Math_DqDe...
p_output	0x18000784	struct Ifx_MDA_FocCo...
voltageDQ	0x1800078C	struct Ifx_Math_CmpF...
p_dAntiwindupCtrl	0	char
p_qAntiwindupCtrl	0	char
modulator	0x18000794	struct Ifx_MAS_Modul...
p_clearFault	0 false	enum (uchar)
p_currentMeasurement	0x18000796	struct Ifx_MAS_Modul...
p_deadTime_tick	20	short
p_deadTimeCompensation	10	short
p_enable	1 true	enum (uchar)
p_enableBrake	0 false	enum (uchar)
p_maxAmplitudeQ15	13653	short
p_minSenseTime_tick	88	short
p_minTurnOnTimeHalf_ti...	10	short
p_output	0x180007B0	struct Ifx_MAS_Modul...
p_period_tick	2000	short
p_status	0x180007CE	struct Ifx_MAS_Modul...
state	2 Ifx_MAS_ModulatorF16_State_on	enum (uchar)
subState	0 Ifx_MAS_ModulatorF16_SubState_bidi...	enum (uchar)
maxAmplitudeFlag	0 false	enum (uchar)
overmodulationFlag	0 false	enum (uchar)
patternGenTLE987	0x180007D4	struct Ifx_MHA_Patter...
bridgeDrvTLE987	0x18000800	struct Ifx_MHA_Bridge...
_Super_Ifx_MHA_BridgeDrv	0x18000800	struct Ifx_MHA_Bridge...
p_status	0x18000802	struct Ifx_MHA_Bridge...
state	2 Ifx_MHA_BridgeDrv_TLE987_State_on	enum (uchar)
overcurrent	0 false	enum (uchar)
overvoltage	0 false	enum (uchar)
undervoltage	0 false	enum (uchar)

Figure 87 Fault check status in Keil debugger window

8.3.2 Indication for fault in the Micro Inspector Pro

There are status LEDs for each fault (software or hardware) in the Micro Inspector Pro. If the fault can be ignored, then the clear fault option can be selected.

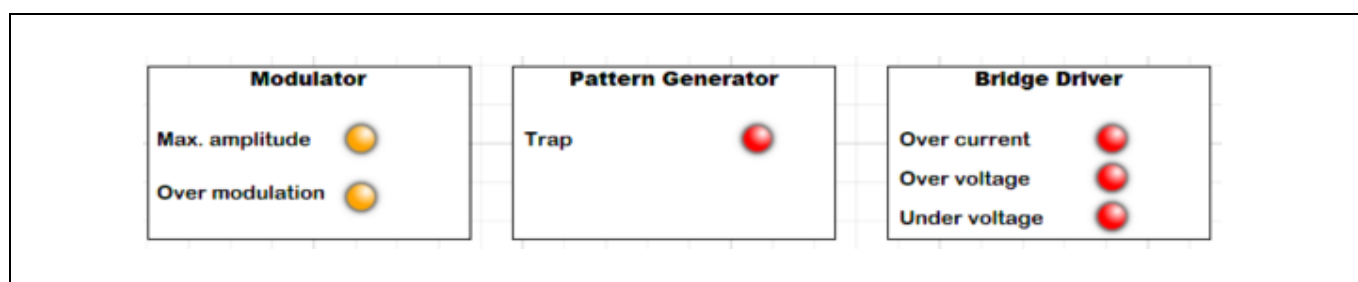


Figure 88 Fault check status in the Micro Inspector Pro

9 Execution time and memory consumption

This chapter shows the execution time of every software function block and also the memory consumption of software in high and low RAM condition. The best performance with TLE987x is achieved moving code to RAM, in particular the SW of the current loop. Of course, the performance depends strongly upon also the options selected in the compiler.

9.1 Execution time of the motor control software

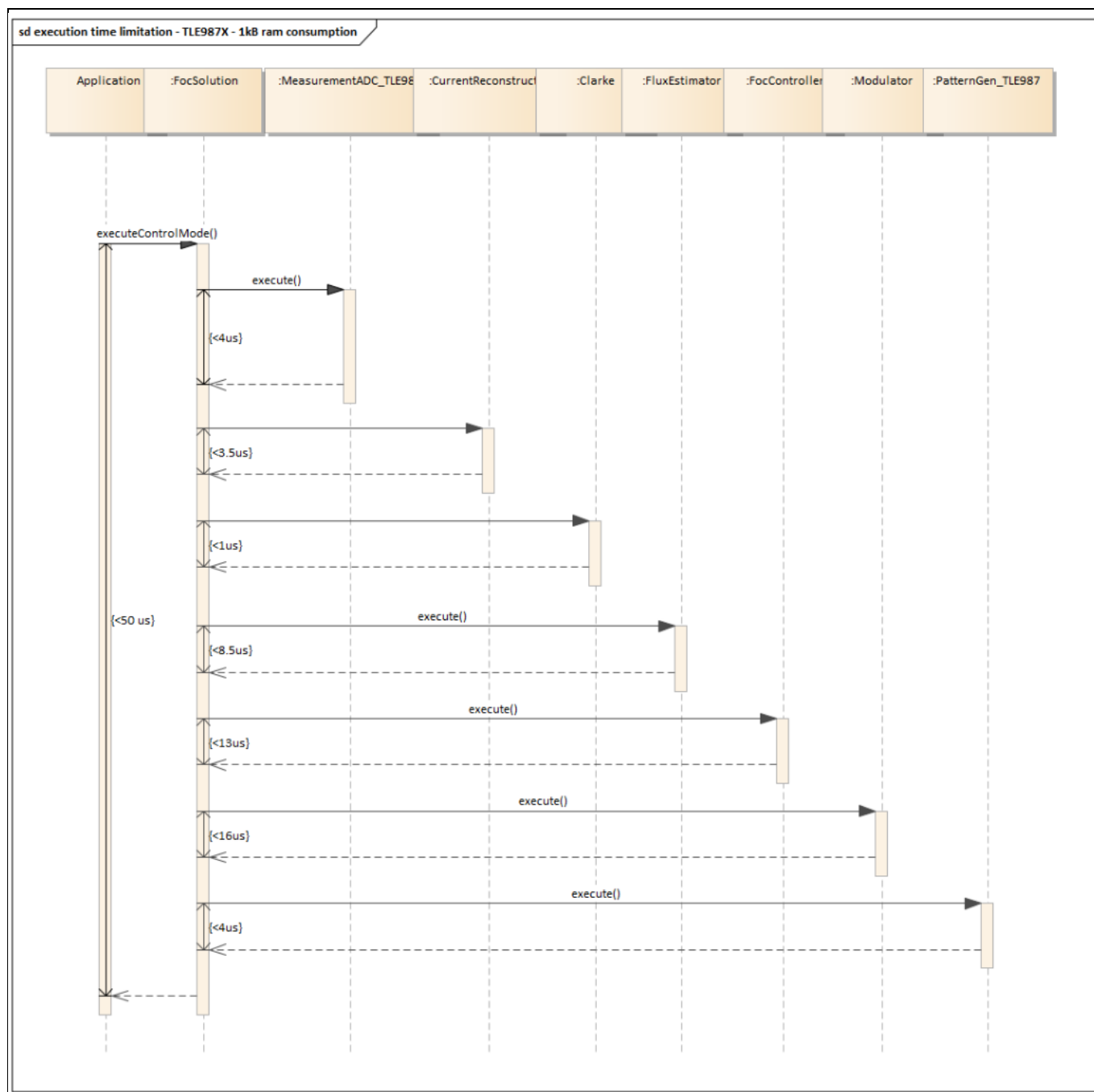


Figure 89 Execution time under low RAM condition

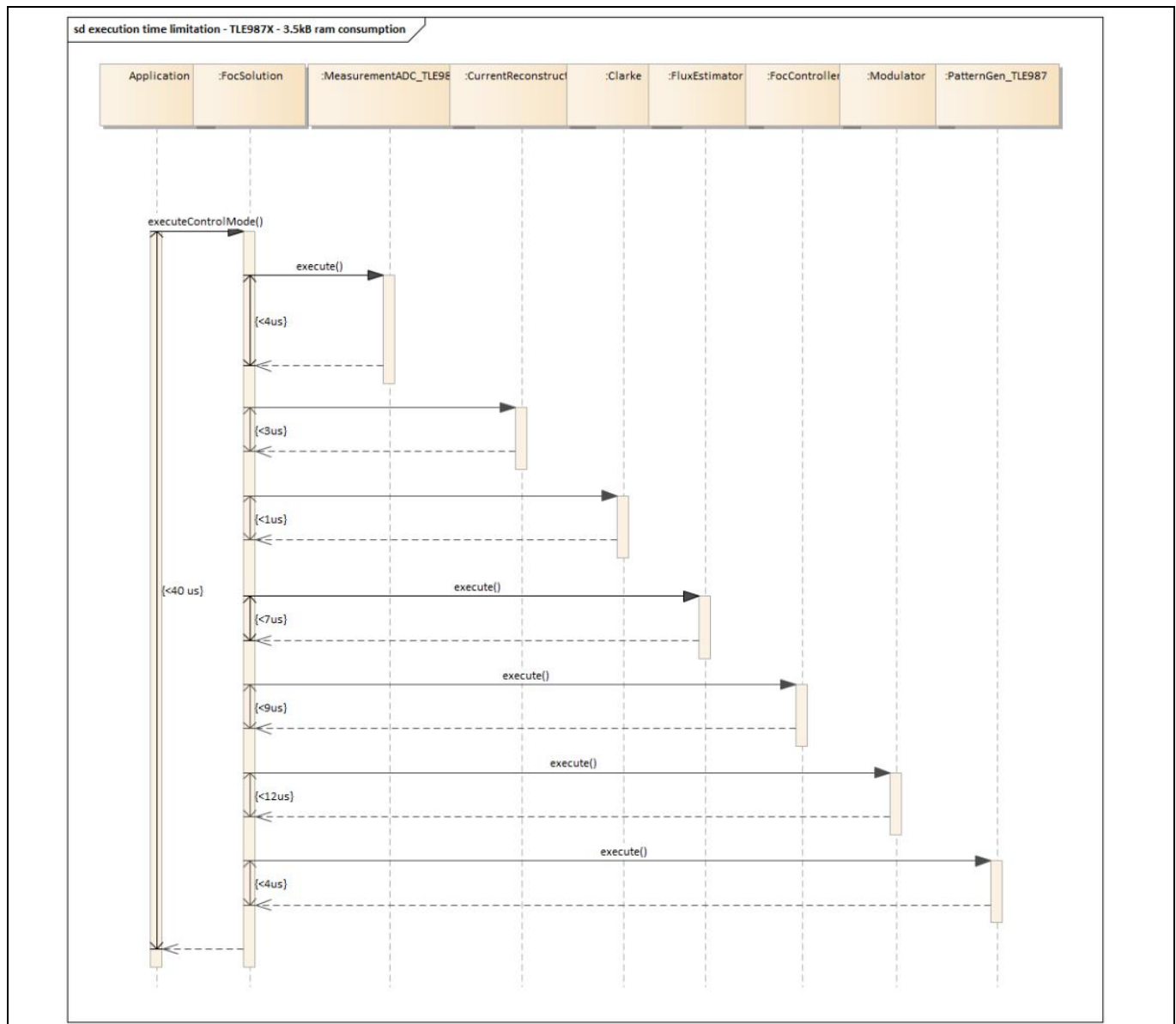


Figure 90 Execution time under high RAM condition

9.2 Memory consumption of the motor control software

The MOTIX™ [Memory Analyzer](#), included in the Infineon Developer Center Launcher, is a tool that enables the analysis of the memory consumption of a device (in this case, the TLE987x) when using a specific SW. In Figure 91, the tools show with pie graphs the memory usage in flash and in RAM, using the BASIC SW with current loop routines executed from RAM.

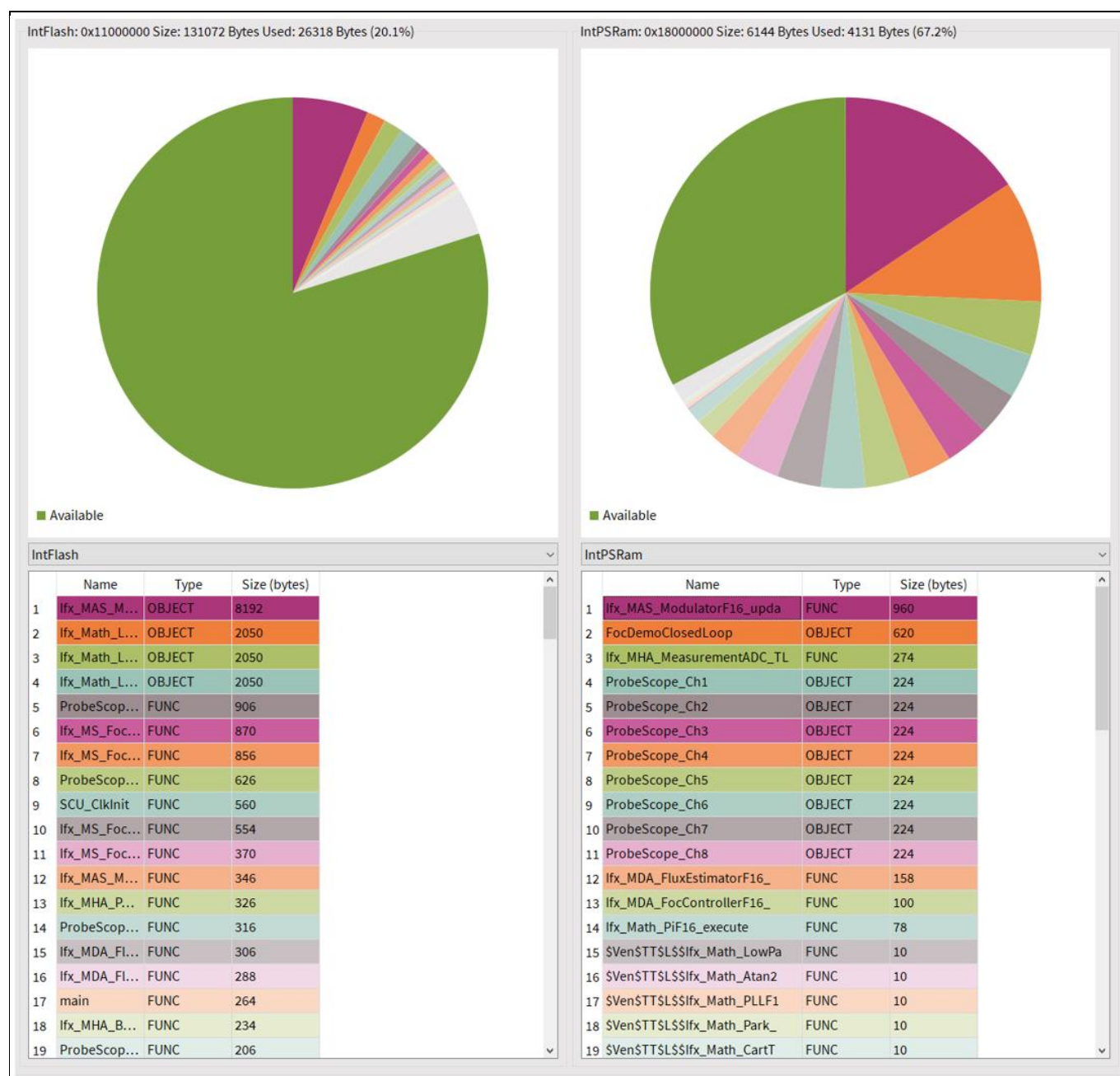


Figure 91 Memory consumption of the motor control software of the TLE987

References

References

- [1] Infineon Technologies AG: TLE987x User Manual, Microcontroller with LIN and BLDC MOSFET Driver for Automotive Applications
- [2] Infineon Technologies AG: Getting started, Motor Control Software Libraries BASIC for TLE987x, 2022-09-30
- [3] Infineon Technologies AG: TLE986x/TLE987x BF-/A-/UH-Step BootROM User Manual
- [4] Infineon Technologies AG: TLE987x/6x Hardware Design Guideline Application Note
- [5] Infineon Technologies AG: Sensorless Field Oriented Control with TLE987x



Revision history

Revision history

Document revision	Date	Description of changes
Rev. 1.0	2023-05-19	Initial creation

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2022-11-07

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2023 Infineon Technologies AG.

All Rights Reserved.

Do you have a question about this document?

Email: erratum@infineon.com

Document reference

AppNote Number

IMPORTANT NOTICE

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.