Induction motor control by iMOTION™

About this document

Scope and purpose

This document is intended for customers who would like to use the iMOTIONTM system applied to an induction motor for the variable speed application.

Intended audience

This document is intended for customers who would like to use the iMOTION™ system applied to an induction motor for the variable speed application.

Table of contents

1 Induction motor V/Hz control overview

1.1 Introduction

The latest software release of iMOTION™ Motion Control Engine (MCE) includes a script engine together with the necessary hook-up for the open loop V/Hz control for induction motors. The open loop V/Hz control is based on the AC machine model so-called "per-phase equivalent circuit" shown in Figure 1. The main impedance of stator winding is an inductance, Lm, which typically ranges between tens and hundreds of milli-Henry while series inductance, Ls, in a range of hundreds micro-Henries for the motor power range of 5 HP or less.

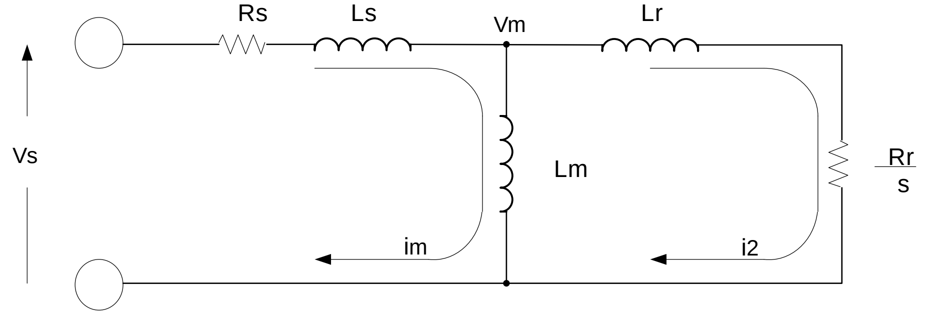


Figure 1 Induction motor equivalent circuit

Where:

Rs: stator leakage inductance

Ls: stator leakage inductance

Lm: stator magnetizing inductance

Lr: rotor leakage inductance

Rr: rotor resistance

s: slip frequency

Based on the per-phase equivalent circuit above, the torque can be expressed by the following equation.

$$T = k \times \frac{R_r}{s} \times {i_2}^2$$

Where:

k: constant including the rated flux

i2: rotor circuit current

Induction motor V/Hz control overview

Torque is uncontrolled all the time. The AC machine itself takes care of torque generation according to the shaft load. If the load increases, then the slip frequency increase thereby the rotor current increase resulting in a torque increase. Therefore it is important to magnetize the motor at full flux for a full torque generation.

The key concept of the V/Hz control is to maintain the field flux constant so that a motor is uniformly magnetized at all speed ranges up to the base speed. In order to keep the same amount of magnetization of flux, the applied voltage, Vs (more accurately Vm, magnetizing voltage) must be proportional to the electric angular speed, ω. In this way, by keeping Vs/ω constant the resulting magnetizing current, i_m, also kept constant. Thereby the magnetizing flux is also maintained constant regardless of the rotating speed. Vs is almost the same as Vm except in the low speed region where the dominant stator impedance is driven by Rs, stator resistance, therefore additional voltage needs to be applied to Vs in order to make the magnetizing voltage, Vm to be unchanged. This additional voltage in the low speed range is so-called "torque boost" voltage. Vs needs to be increased as speed increases, thereby the excitation frequency also increases linearly. When the excitation frequency reaches the base frequency, which is typically a 60 Hz, then the applied voltage reaches its limit which is often AC230 V or 460 V. If the demanded speed exceeds the base frequency, the operation becomes field weakening. In this field weakening operation the torque is inversely reduced thus the resulting power, which is a product of torque x speed, is maintained constant. Therefore this region is also called "constant power operation range". Figure 2 illustrates these operation in terms of speed (Hz) and voltage.

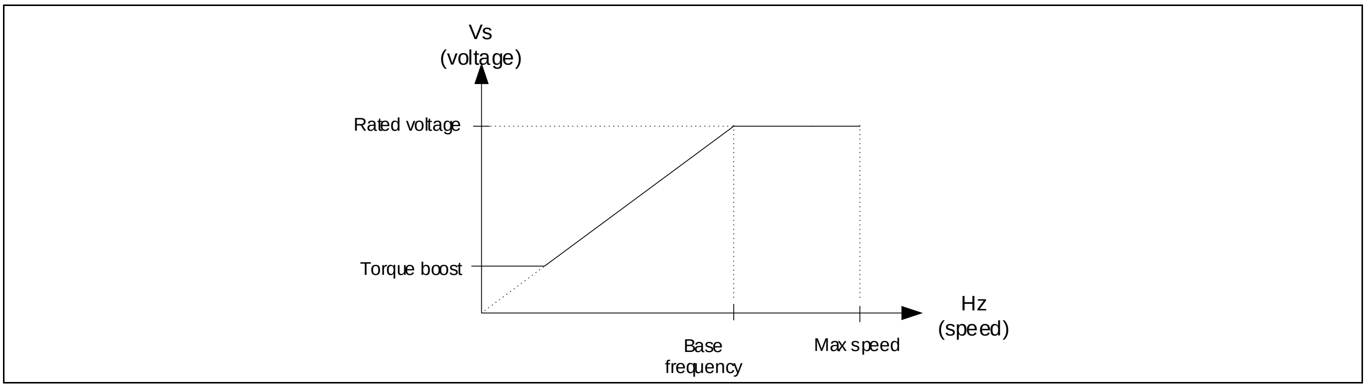
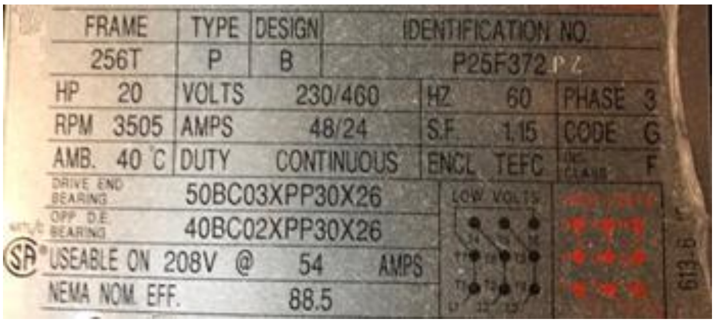


Figure 2 Voltage vs. speed

In this operation, all necessary application information needs to be entered into the configuration tool, MCEWizard. Fundamentally all necessary information can be obtained from the motor nameplate which is very much standardized in case of induction motors. Figure 3 shows an example of nameplate data of a typical induction motor. The required information from the nameplate are:



1) VOLTS: rated voltage
2) HZ: Base frequency at the rated voltage
3) RPM: rated speed at rated voltage and base frequency at full load

Figure 3 Motor nameplate example

From these information, user needs to derive the pole number. The pole number is calculated by:

$$p = \frac{120 \times Hz}{RPM}$$

Where: Hz = base frequency in herz and RPM = RPM in nameplate data

P = pole number (must be even number) and rounded to the closest integer.

In Figure 3 example, p = 2 which is truncated from 2.05 = 120*60/3505

These figure will be used to configure the desired V/Hz control system.

1.2 Script development workflow

The typical workflow of script program development starts from using MCEWizard (or any other text editors) to write script code and save as script input file with '.mcs' suffix. MCEWizard is used to configure available Analog-to-Digital Converter (ADC) or General-Purpose-Input-Output (GPIO) pins if needed, and MCEWizard is also used to compile the script code to generate a script object file with '.ldf' suffix. The ldf file contains information about the total number of script instructions for Task 0 and Task 1, as well as a list of global variables defined in the script code. Then MCEDesigner [3] is used to download the ldf file to the target MCE, and it also supports monitoring the values of global variables used in the script program. More details about the script language and its development can be found in [2].

2 V/Hz control structure

The V/Hz control is implemented and divided into two parts. One is the iMOTION™ provided MCE (Motion Control Engine) and the other is supplemented by the Script language which is open to user for own modification in its programming and needs to be downloaded into the target system. Two variables have to be controlled. One is the rotating frequency which is proportional to the commanded speed input and generated by the provided MCE hardware thereby there is no need for programming. The other is the voltage applied to the motor which is also proportional to the commanded voltage and the speed command is translated to the applied voltage command, namely Vq, by the Script language. The MCE contains all elements of motor control by a ready made firmware and user only needs to configure the control for V/Hz by properly selecting the software switches and multiplexer by the MCEWizard and the Script language programming. The Script language is the “C” like language being able to describe the desrired user function by programming, and in this case, the V/Hz control front-end function (circled by the dashed line in Figure 4) is implemented by the Script language.

Overall functional block is shown in Figure 4.

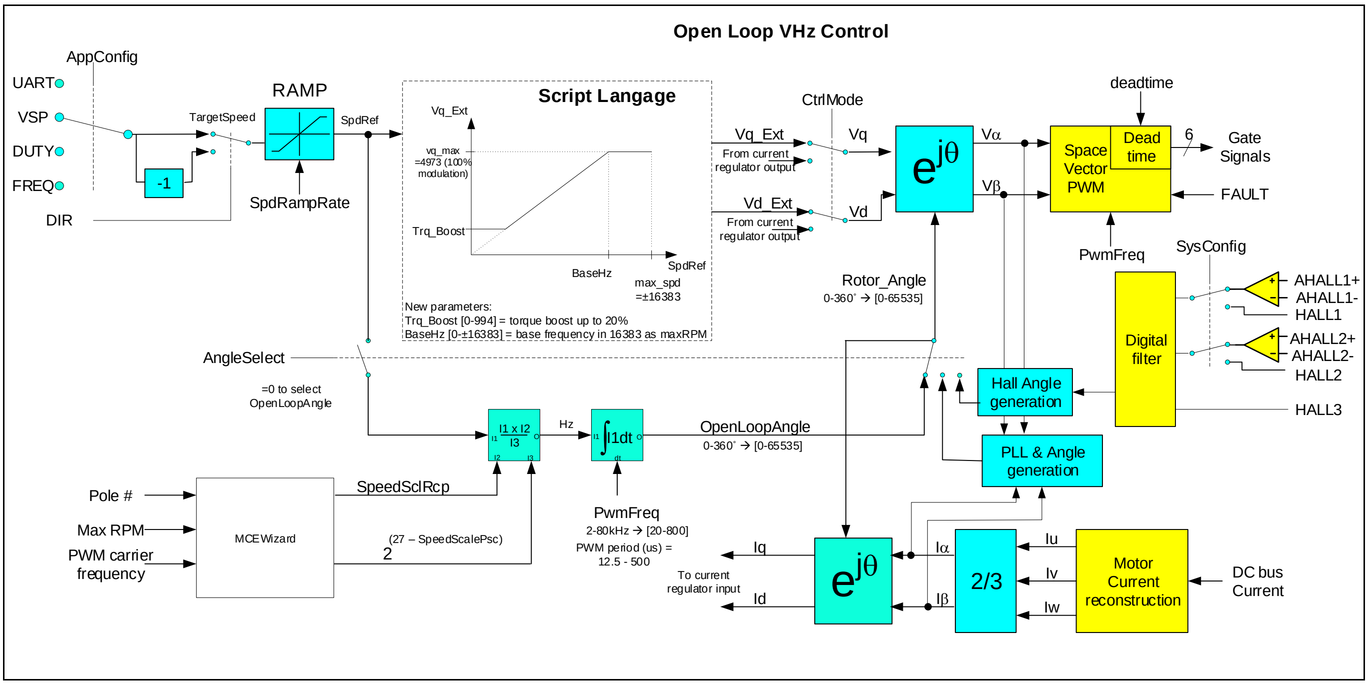


Figure 4 Functional block diagram

2.1 MCEWizard input entry requirement

The following inputs are required in the MCEWizard in order to perform the induction motor open loop V/HZ control.

- Motor Control Mode Input: Voltage OpenLoop Control
- Motor Angle Select: Open Loop Angle
- Script Function Options: Enabled
- Motor 1 PWM Frequency
- Motor Poles
- Motor Max RPM

First, in Welcome page, two boxes need to be checked (Figure 5). One is "I have modified the circuit board" and the other is "Enable advanced questions" This is required in order to make the the following selections available.

Motor 1 PWM Frequency, Motor Poles, and Motor Max RPM are application design and motor dependent, and need to be entered accordingly. "Motor Control Mode Input" always has to be Voltage OpenLoop Control and "Motor Angle Select" always has to be Open Loop Angle regardless of the application or motor selection.

These parameters entry can be found in "Base Configuration Options" page shown in Figure 6 and "Advanced Mode" page shown in Figure 7.

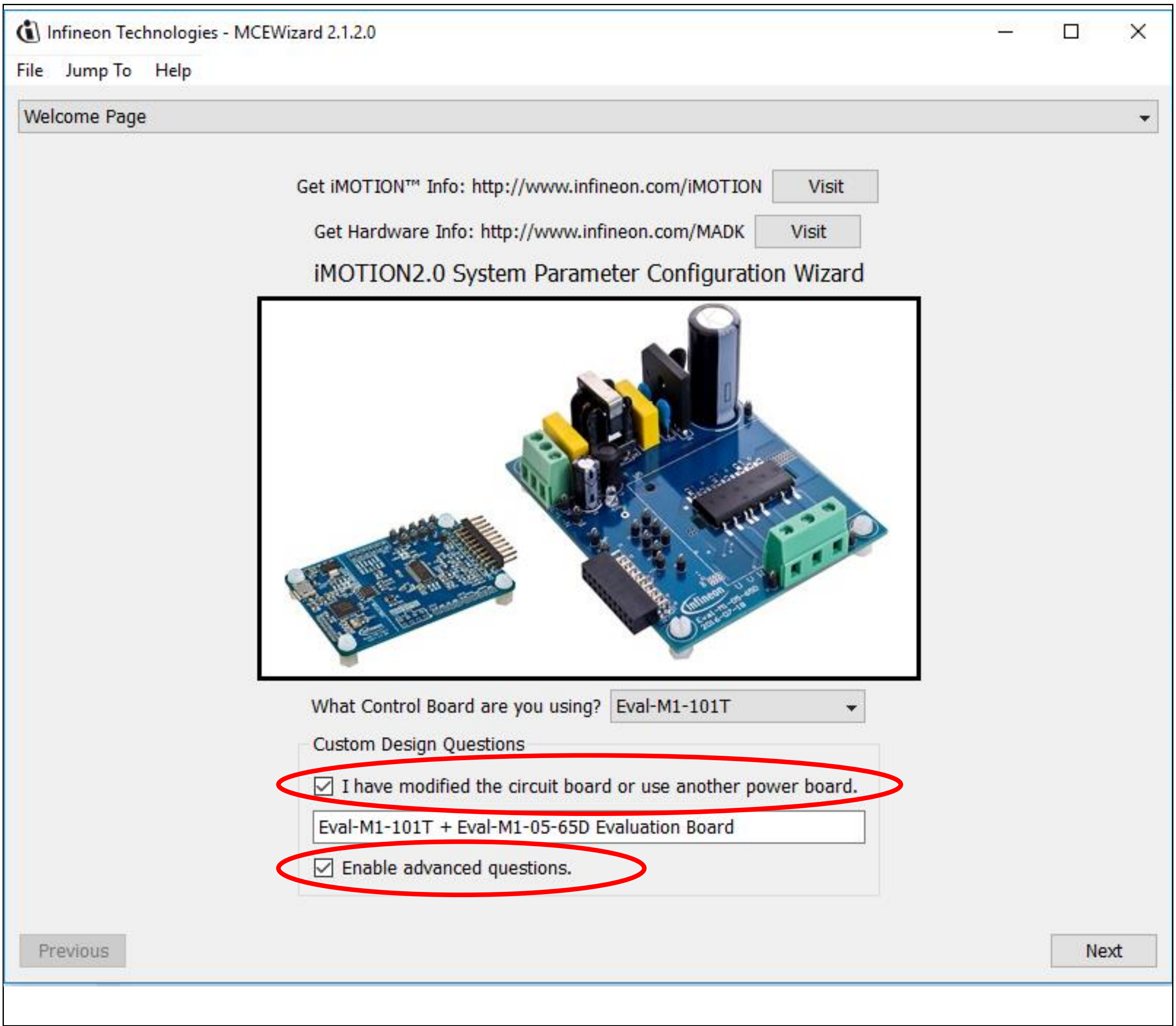


Figure 5 MCEWizard – Welcom Page

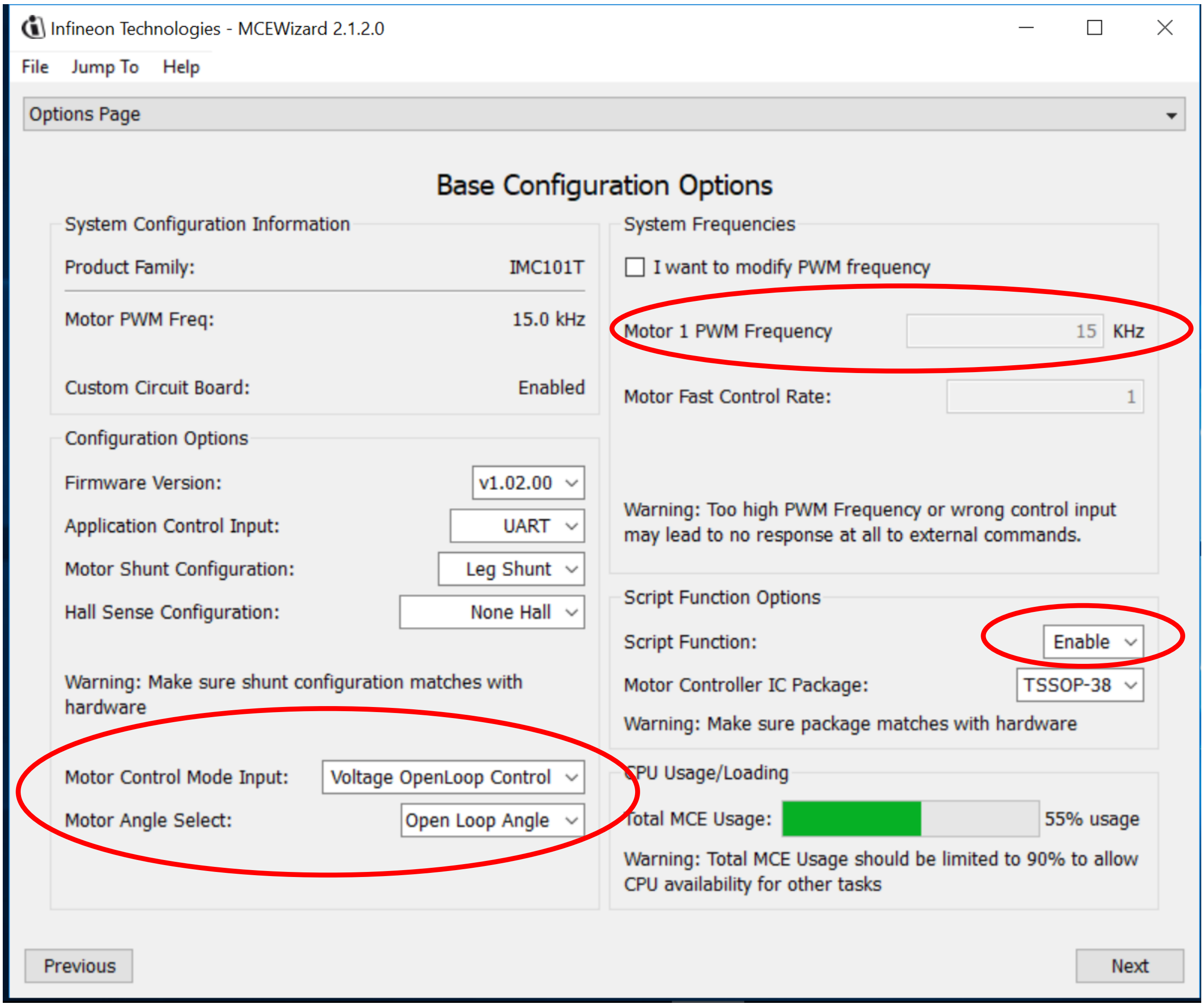


Figure 6 MCEWizard - Base configuration options

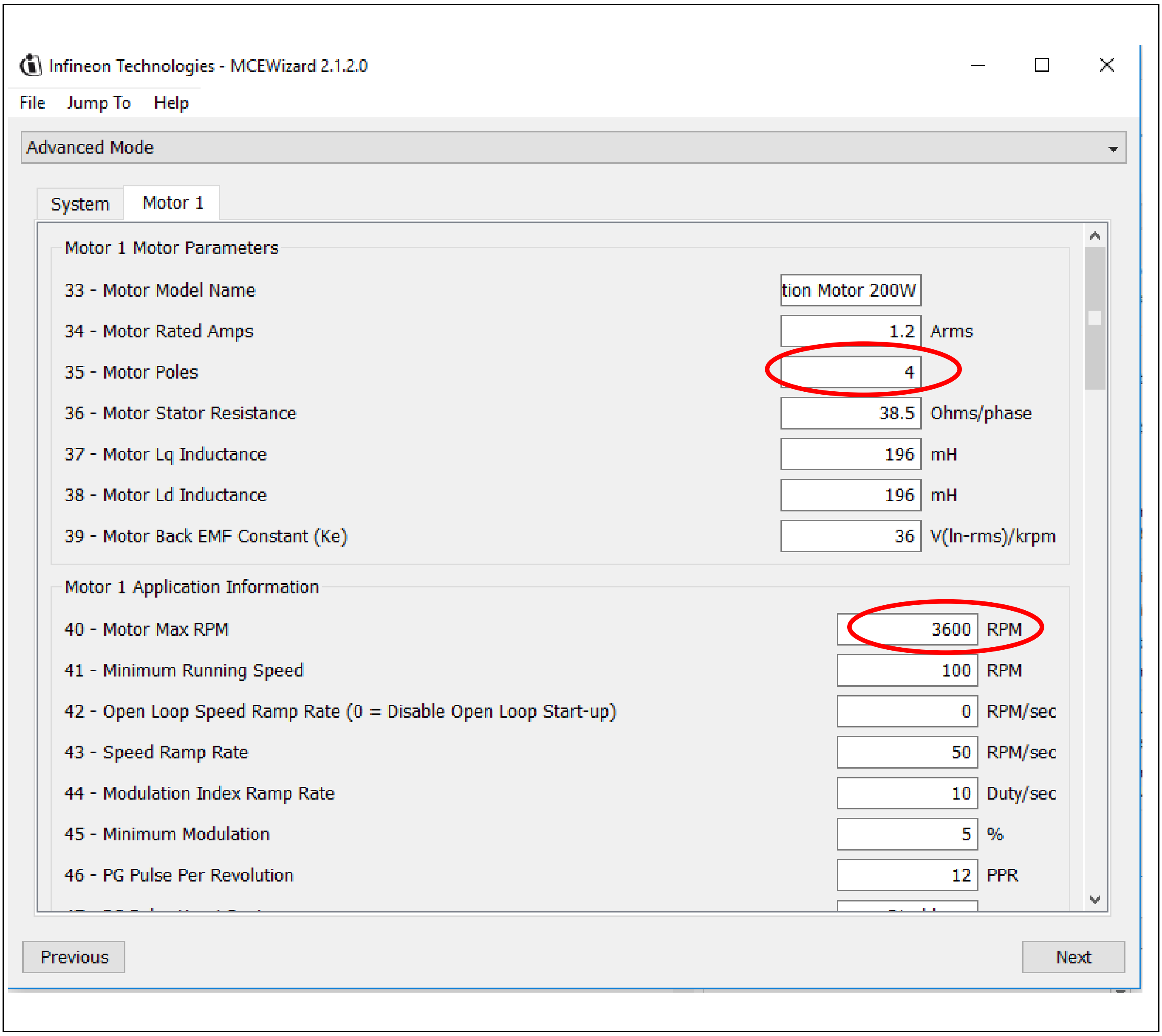


Figure 7 **MCEWizard – Advanced mode**

2.2 V/Hz supporting MCE functions

The control elements excluding the Script language (enclosed by dashed line shown in Figure 4) are provided by the MCE functions. All colored blocks are already in place in the released MCE function. User only needs to configure the signal path switches and the associated parameters mostly in the MCEWizard configuration tool. The MCE functions associated with V/Hz control are divided into the following sections.

- Reference generator with ramp block
- Frequency generator/accumulator
- PWM generation

2.2.1 Reference generator with ramp block

The reference generator with ramp block has four input selection depending on the user interface.

Except for UART communication option, VSP or DUTY or FREQ inputs have a pre-mapped scaling into the motor stop point, the motor start point and the motor max speed. Please refer to MCE Reference Manual, section 2.1.8 Control Input. The selection is made by the parameter, AppConfig (Index 71) in the bit position of [0:2] (0=UART, 1=VSP, 2=FREQ, 3=DUTY). When VSP is chosen, make sure that VSP pin is connected to provide the speed reference by analog voltage while FREQUDUTY pin is connected to provide the speed reference by external digital interface when chosen. Internal speed reference is then go through DIR switch polarity insertion for forward/reverse rotation. After DIR switch, the signal variable, TargetSpeed is signed and going into the ramp block.

2.2.2 Frequency to Angle Accumulator with ramp block

This block integrates the motor electrical frequency to generate the electrical angle that passes to the vector rotator and the inverse vector rotator which perform the coordinate transformation of the voltage and current. The fundamental frequency applied to the motor is determined by this control element.

The basic mechanism of generating the electrical angle variable, "OpenLoopAngle" shown in Figure 4, at a given instance is implemented by the frequency accumulator as an integrator block. At each PWM carrier frequency update, it adds the right amount to the frequency accumulator of which applies to the electrical angle. The right amount of frequency increments, namely "Hz" in Figure 4, are derived from the MCEWizard input, motor pole number, PWM carrier frequency and maximum speed in rpm. The command speed needs to be re-scaled according to the relation between the engineering unit in RPM and the internal number as in max speed= 16383.

The motor pole number and the maximum rpm speed generates the electrical maximum frequency applied to the motor stator. The PWM carrier frequency and the electrical maximum frequency determines the incremental adding amount to the frequency accumulator which generates the electrical angle.

These three MCEWizard parameters must be synchronized with the Script language task parameter, BaseHz, as derived from the same input source.

The basic angle generation mechanism is simple and shown below with numerial data range/resolution. Parenthesis number indicates the data range.

Frequency Generator:

OpenLoopAngle [0 – 65535] = Hz + previous_angle

Previous_angle = OpenLoopAngle

For example, if the pole number is 4 and the maximum rpm speed is 3600 rpm, then the electrical maximum frequency is 120 Hz at the maximum rpm speed and if the PWM update rate is 12 kHz, then it requires 100 times of PWM updates to complete one electrical cycle to achieve 120 Hz. Therefore Hz = 65535/100 = 655 in this example. These calculations are done automatically within the MCEWizard and the firmware will generate the required angle generation automatically.

2.2.3 PWM generation/deadtime/VDC bus voltage compensation

The PWM modulator block shown in yellow color in Figure 4 generates the necessary gate signal to the external six power switching devices in the form of pulse width modulated digital signals.

Type of PWM can be selected by choosing the configuration of HwConfig defined in the static parameter. The bit position[4:3] defines: “0”-3 phase PWM, and “3”-2 Phase Type 3 PWM.

The deadtime can be adjusted by PWMDeadtimeR and PWMDeadtimeF and is depending upon the gate driver and power devices used. In general it is recommended to choose a value between 500 nanoseconds and 1 microsecond.

The DC bus compensation is enabled by setting the bit position [0] of SysConfig register to ‘1’. The DC bus compensation automatically adjust the instantaneous PWM pulse width accoding to the DC bus voltage fluctuation. If the DC bus is 320 V norminal and fluctuate to 400 V, for example, then the corresponding PWM pulse width be also adjusted to reduce to 320/400 = 80 % of its nominal value.

2.3 Script language portion of V/Hz control

The Script language implementation, shown in the chapter 3, has two parts. One is the initialization task which execute only once at the beginning and the other is a periodically executed scanned task at a rate of every 50 msec. the initialization task initializes the following parameters.

- VQ_MAX: maximum voltage command internal unit, 4973 is the max voltage. Local variable
- TrqBoost: torque boost voltage scaled to VQ_MAX. If 10 % torque boost, it is 497, global variable
- BaseHz: base frequency equivalent speed internal unit, mapped to max_rpm=16383.

$$BaseHz = \frac{Base\ speed\ in\ frequency}{Max\ rpm\ in\ frequency} \times 16383$$

  For example, If max_rpm is mapped to 120 Hz in the MCEWizard, and want BaseHz to be 60 Hz, then BaseHz = 8191. Then the voltage command and frequency is proportional up to BaseHz. Beyond BaseHz, field weakening operation starts from 60 Hz to 120 Hz.
- AngleSelect: set to 0 to choose open loop control
- CtrlModeSelect: set 0 to choose voltage mode cntrol

In the initialization task, two configuration parameters, namely AngleSelect and CtrlModeSelect, need a coherent update since we need to avoid a glitch at the moment where one of them are updated while the other still holds the previous status which could lead to possible malfunction.
The main body task is relatively simple that the required voltage from the ramp block output, SpdRef, be generated and passed to the output, Vq_Ext. the voltage commend is clamp to VQ_MAX which is currently mapped 4973 as a maximum voltage limit corresponding to a 100 % PWM modulation. If the base frequency, BaseHz, is specified to be less than that of maximum speed, then the frequency generator continues to generate PWM output beyond BaseHz resulting infield weakening operation with the clamped voltage.

2.3.1 Torque boost

Torque boost function can be usuful when the AC machine operation requires low speed range control. Without torque boost function, the induction motor becomes sluggish or exhibits cogging or sometimes results in stall condition depending on the load condition. All these negative phenomenon are stemmed from the lack of torque caused by not enough voltage at low speed operating range.
When the induction motor being operated at low speed, the majority of motor impedance is the stator resistance, Rs, due to the fact that inductance portion of impedance becomes negligible by near DC equivalent frequency. Therefore the magnetizing current, Im, will be significantly reduced without torque boost function and the resulting magnetizing flux decreases significantly.
Ideally if the correct voltage be applied at low speed in order to maintain the magnetizing flux, then the minimum torque boosting voltage, Trq_Boost, should be expressed theoretically by:

$$Trq\_Boost[\%] = \frac{Im \times Rs}{Rated\ Voltage} \times 100$$

In reality, the torque boosting voltage should be somewhat bigger than that of above equation in order to preserve the room for inductance portion. Figure 8 illustrates a linear and gradual increase of voltage as a frequency (speed) increases at torque boosting area. Therefore it should be adjusted to a slightly bigger voltage than the theoretical value to minimize the gap.

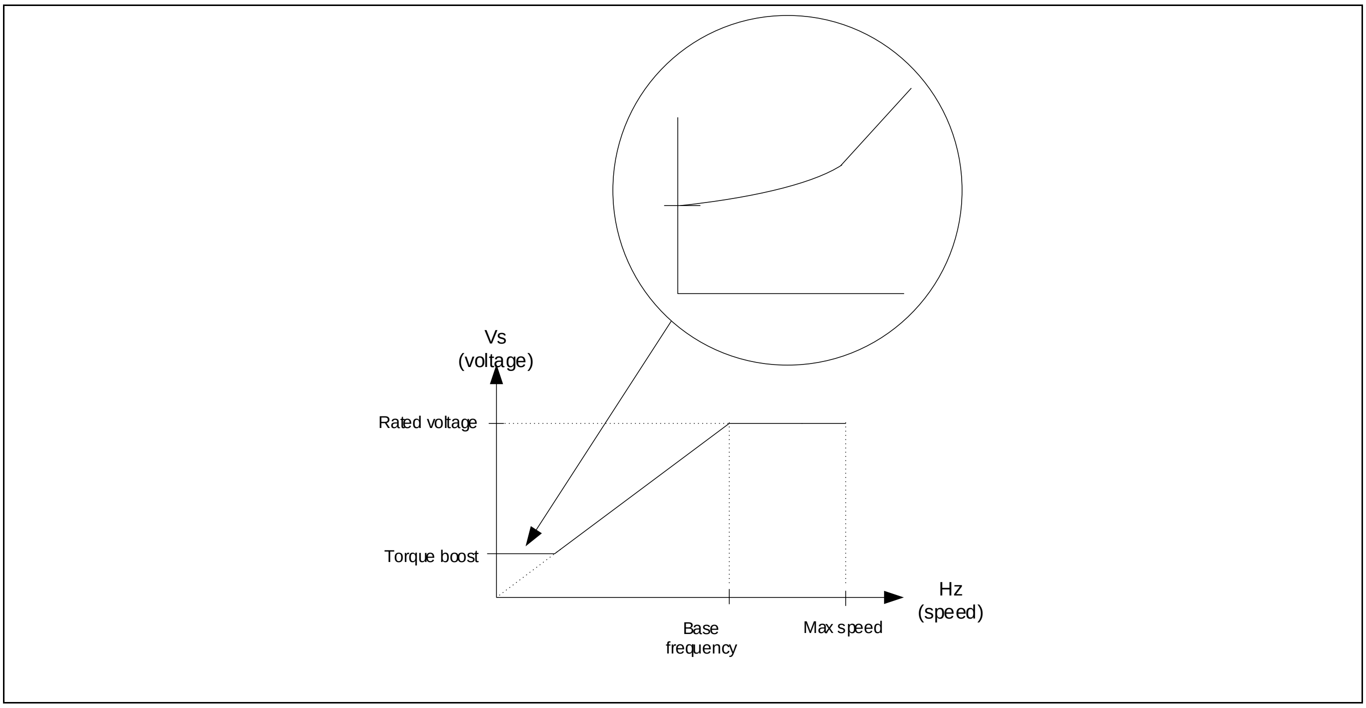


Figure 8 Torque boost function

2.3.2 Field weakening

Field weakening can be achieved by properly configuring the base frequency and the maximum speed in combination of the voltage command. It is an open loop control by impressing the voltage and frequency to a motor by predetermined values.
In order to achieve the field weakening, the user has to the power constant control where high speed beyond the base speed to be only achieved while sacrificing the torque. In the other word, the shaft output power of the motor cannot exceed the rated power output specified by the motor size.

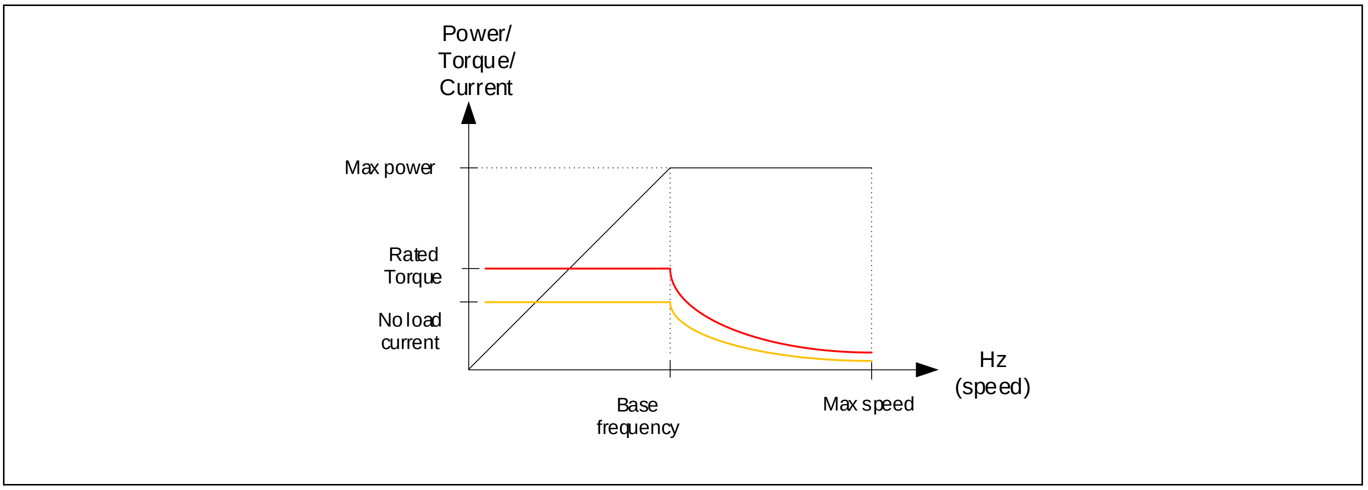


Figure 9 Field weakening

As the speed go beyond the base frequency, torque should be reduced by inverse proportional to speed increase so that “power = torque x speed” needs to be maintained to be maximum power. The motor current will be also reduced accordingly by inverse proportional to speed.
The V/Hz control will provide this field weaknening operation by the Script language

3 V/Hz control Script language example

```
#SET SCRIPT_USER_VERSION (1.00)  /*Script version value should be 255.255*/
#SET SCRIPT_TASK0_EXECUTION_PERIOD    (50)  /*Script execution time for Task0 in mS, maximum value 65535*/
#SET SCRIPT_TASK1_EXECUTION_PERIOD    (1)  /*Script execution time for Task1 in 10mS, maximum value 65535*/
#SET SCRIPT_START_COMMAND (0x3)  /* Start command, Task0 : Bit0, Task1 : Bit1; if bit is set, script executes after init */
#SET SCRIPT_TASK0_EXECUTION_STEP (10)  /* Script Task0  step, This defines number of lines to be executed every
50mS*/
#SET SCRIPT_TASK1_EXECUTION_STEP (1) /* Script Task1  step, This defines number of lines to be executed every 1mS*/
/* motor example, Rs = 6ohm, Ls = 169mH, 4p, base freq=60Hz */
/*****************************************************************************************************************/
int TrqBoost,BaseHz;        /* Global variable definition */
/*****************************************************************************************************************/
Script_Task0_init()        /*Task0 initialization - only once at start*/
{
 int volt, VQ_MAX;      /*Local variable definition */
 VQ_MAX = 4973; /* 100% duty command */
 TrqBoost =129;  /* 129 = 2.6%, 4973*2.6%, torque boost = VQ_MAX*% of troque boost (%) ~ Im x Rs /rated voltage */
 BaseHz =8191;              /* 8191 = 60Hz = 1800rpm for Max_RPM= 3600rpm =16383*/
 Vd_Ext = 0;         /* d-axis voltage to be zero */
 EnableCoherentUpdate();
 AngleSelect =0;         /*Set to open loop mode*/
 CtrlModeSelect =0;       /*voltage control mode */
 DoCoherentUpdate();
}
/*****************************************************************************************************************/
Script_Task0()           /*Task0, 50msec update */
{
  volt = SpdRef * VQ_MAX / BaseHz;  /* voltage command scaled to VQ_MAX */
  if (volt > VQ_MAX)
  {
    volt = VQ_MAX;       /* feild weakening */
  }
  if (volt < TrqBoost)
  {
    volt = TrqBoost;      /* clamp to min voltage */
  }
  Vq_Ext = volt;          /* update the voltage command */
}
```

V/Hz control Script language example

Figure 10 shows the motor current waveform of the Script language example listed above. The motor specification in this example area:

- Rated motor current =1.2 Arms
- Rated voltage = 230 V
- Pole number = 4
- Max speed RPM = 3600
- Base Frequency = 60 Hz.

It accelerates from 0 to 88 Hz and decelerates from 88 Hz to reverse direction speed. During acceleration, the motor current reaches 870 mA and settled at 500mA at 88 Hz steady state.

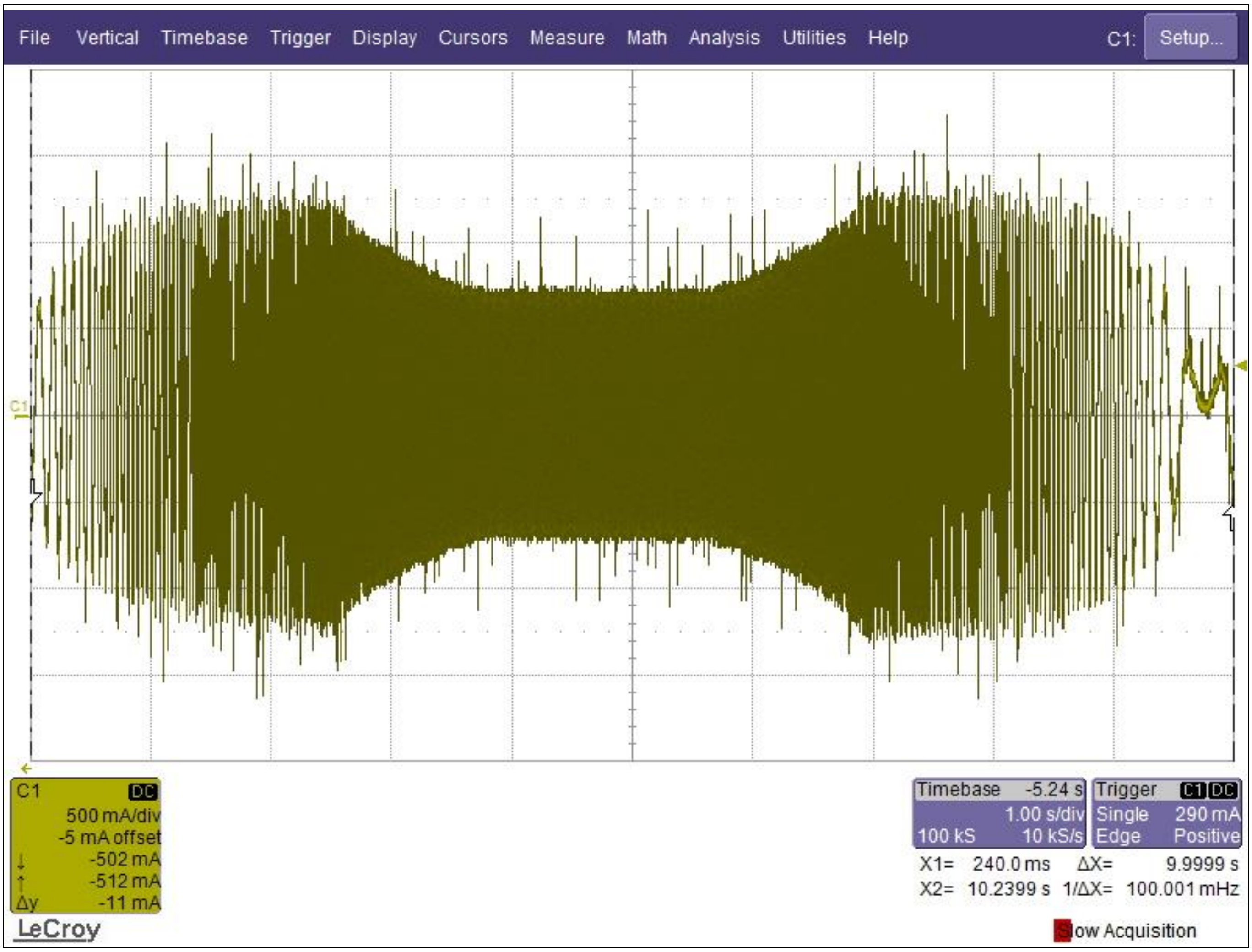
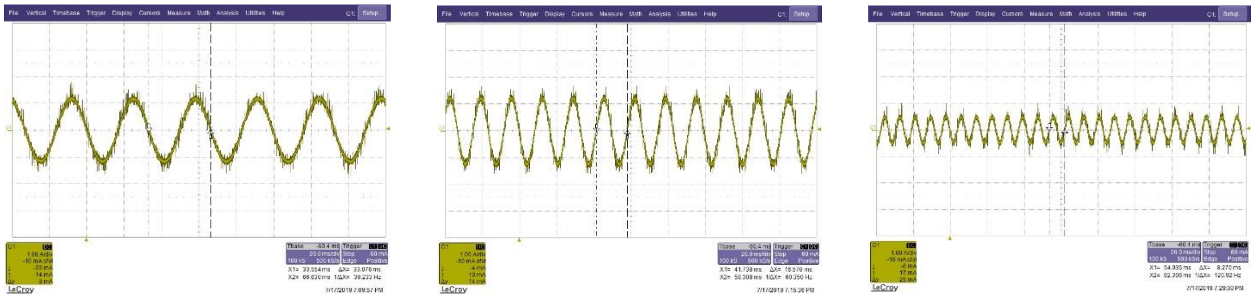


Figure 10 Induction motor current waveform

V/Hz control Script language example

Figure 11 shows the no load motor current at 30/60/120 Hz where the current amplitude stays same up to 60 Hz (Base frequency) and will be inversely reduced as speed go beyong the base frequency in field weakening range (120 Hz).



30 Hz	60 Hz	120 Hz

Figure 11 Motor current at 30/60/120 Hz

Figure 12 shows the MCEDesigner scope picture at starting. It helps examine the details of starting flux as to whether or not it is enough torque at staring due to lack of magnetizing current. In this example, it shows the motor phase current, IU, and the motor speed command, SpdRef, after the ramp block. It was triggered at starting when the ramped speed command reaches at 500 count. It exhbits a lower-than-expected magnetizing current thereby causing a lack of torque to start. The current should be constant magnitude at beginning. It requires a larger value of TrqBoost parameter which was originally set to 129.

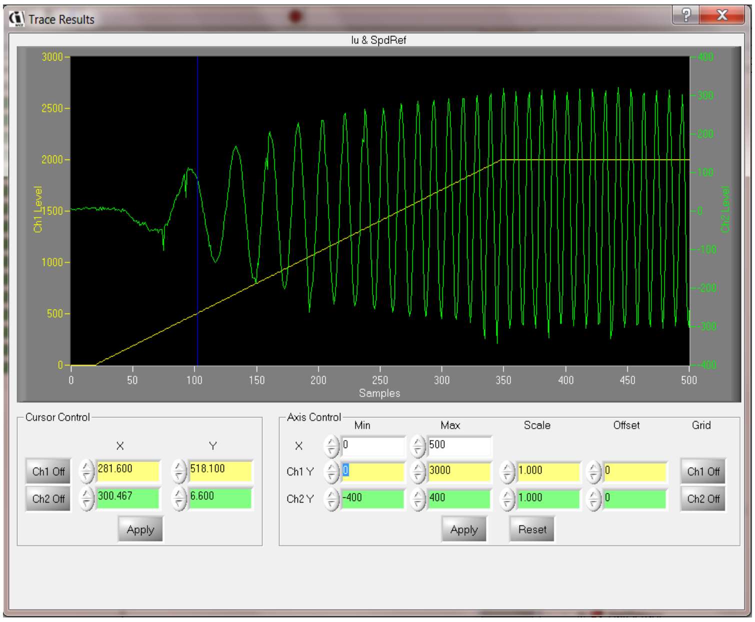


Figure 12 MCEDesigner scope picture at starting

4 References

[1] iMOTION™ IMC100 High Performance Motor Control IC Series Datasheet (REV 1.4).

[2] iMOTION™ Motor Control Engine Software Reference Manual (REV 1.2).

[3] MCEDesigner User Guide (REV 2.0.1.0).

[4] Electric Drives – An Integrative Approach, Ned Mohan, 2000

[5] Power Electronics and AC Drives, B.K.Bose, 1986

Revision history

Document version	Date of release	Description of changes
0.1	7/15/2019	Initial draft.
1.0	7/23/2019	Initial release

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

Edition 2019-07-29

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Email: erratum@infineon.com

Document reference
AN2019-17