Freescale Semiconductor
White Paper

MC3PHACWP
Rev. 0, 10/2005

Ready-to-Use AC Induction Motor Controller IC for Low-Cost Variable Speed Applications

by: Dave Wilson and Bill Lucas
Freescale Semiconductor, Inc.

1 Abstract

Recent advancements in PWM generation and control techniques have been combined with state-of-the-art control algorithms in a pre-programmed solution referred to as the MC3PHAC, which is designed to dramatically minimize up front development costs and time to market in variable speed AC motor control applications. Even though the device requires no programming, flexibility has been maintained, allowing the user to customize the device to a particular application. Particular attention has been given to the safety and fault processing features, including "dead crystal" shutdown, hardware fault shutdown, and DC bus monitoring/protection.

Table of Contents

2 Introduction

Use of variable speed control of AC induction motors has increased sharply over the last decade as the promise of energy savings and more elegant control techniques are being realized. This is particularly true in applications that require full speed operation for only a small percentage of the time, such as certain fan and pump loads. Since line-connected AC motors have intractable speed characteristics, such load demand variations have historically been handled by throttling between the motor and load, a technique which has been compared to driving a car with the accelerator pedal to the floor, and controlling speed with the brakes. However, by controlling the operating speed of the motor directly, energy savings of up to 75% have been cited for certain applications as compared to direct line-connected motor operation.

Several techniques ranging in sophistication have been utilized to accomplish variable speed AC motor control. If high bandwidth torque control is required across a wide operating speed, field oriented techniques utilizing a rotor speed sensor, or using the motor itself as the feedback sensor may be employed. Since AC induction motors are asynchronous in nature, the calculations needed to accomplish this often require a high performance controller, such as a DSP. However, many (if not most, by volume) variable speed AC motor control applications only require moderate torque control performance down to a frequency of about 5 Hz. In these cases, simply controlling the waveform voltage and frequency to the motor (volts-per-hertz control) is the most economical approach.

Regardless of the control topology selected, it is undeniable that larger portions of the task of developing a variable speed drive are being devoted to the software effort, with its associated tools investment. Since the MC3PHAC requires no programming, it eliminates this investment requirement, which consistently reduces overall project development and debug time. Being a “fixed” solution, it is unavoidable that the trade off for these advantages will come at the cost of some reduced flexibility. However, great care has been taken to insure that most of the critical system parameters common to high performance AC drives are dynamically configurable, allowing the MC3PHAC to work in a multitude of variable speed configurations. Also, the MC3PHAC utilizes a serial interface which implements a special communication protocol, that allows a PC or microcontroller to configure the operating characteristics and control the motor in real time as a host. For example, through the host software a computer can exercise complete control over the volts-per-hertz relationship, allowing the MC3PHAC to work in variable torque as well as constant torque variable speed applications.

3 Advanced Motor Control PWMs

At the core of the MC3PHAC's capabilities is an advanced PWM module designed specifically to meet the demanding requirements of high performance AC drives. The module is clocked at 8 MHz (125 ns between timer tics), and generates six center-aligned PWMs in three groups of complimentary PWMs. This allows the MC3PHAC to connect directly to inverters that are indigenous to almost all three-phase AC motor drives, as shown in Figure 1. The polarity of the high-side PWM signals can be specified independently from the low-side PWM polarities. Dead-time is inserted between the on-times of each complimentary signal pair and may be adjusted to any value between 0 and 32 µs, in 125 ns increments.

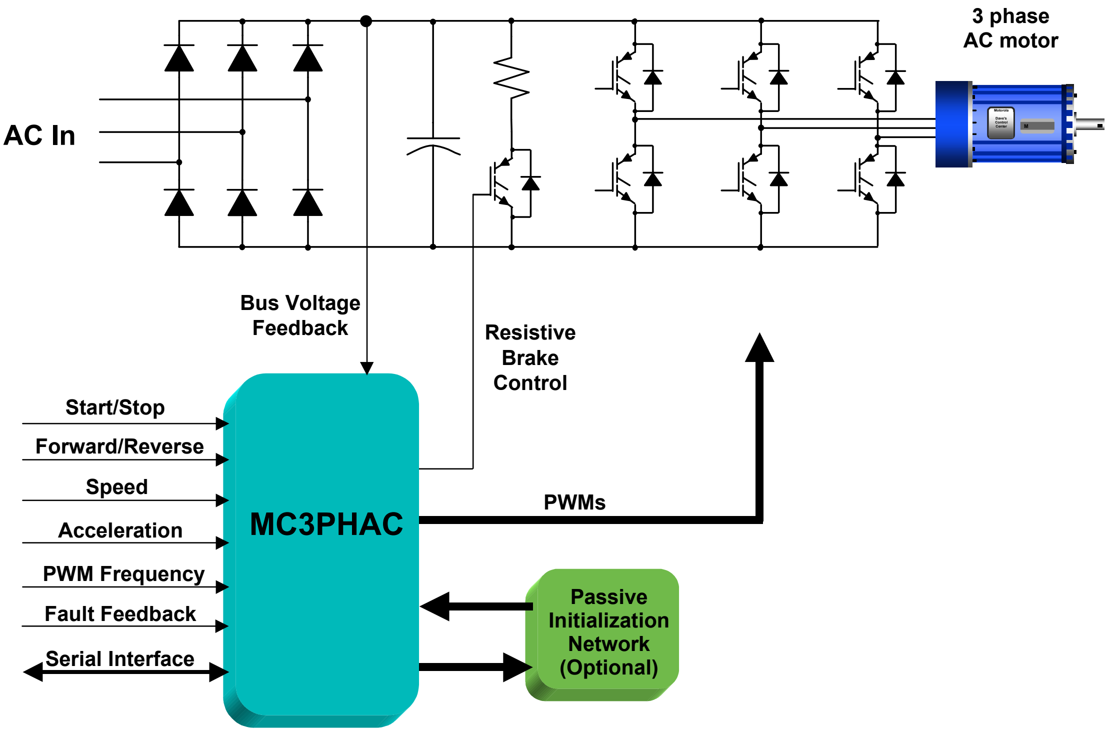


Figure 1. Typical 3-Phase AC Motor Drive Using the MC3PHAC

The PWM frequency can be specified as one of four values, as illustrated in Table 1, along with the effective PWM resolution for each frequency. Each PWM output is synthesized from a 512-entry table, consisting of 8-bit values, as shown in Figure 2. While this limits the peak-to-peak resolution of the output waveform to eight bits, it does not necessarily mean that the PWM resolution itself is limited to eight bits. This is particularly true for smaller modulation indices. The PWM resolution defines how many distinct values can exist over the full modulation range (0% to 100%), which is different from the peak-to-peak resolution of the output waveform.

Table 1. PWM Frequencies and Corresponding Resolutions

PWM Frequency	PWM Resolution
5.291 kHz	9.6 bits
10.582 kHz	8.6 bits
15.873 kHz	8 bits
21.164 kHz	7.6 bits

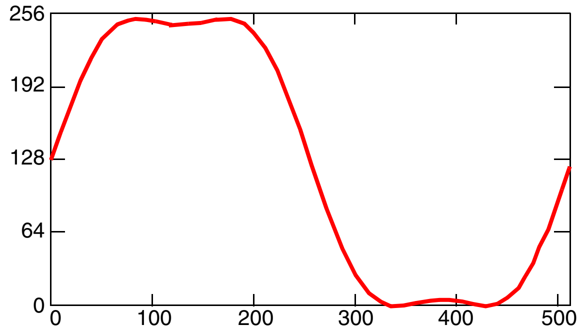


Figure 2. Table Used for Waveform Synthesis in the MC3PHAC

Another factor that can have a much greater impact on waveform distortion than the resolution of the wave table entries, is the sampling frequency by which the motor waveforms are updated. Since the PWM module acts as a sample and hold function, the waveform will be distorted in two ways.

1. Sample and hold functions generate phase lag, which increases as the sampling frequency decreases. This is generally not a problem for open-loop waveform generation. However, it must be considered when performing any closed-loop functions such as bus-ripple compensation, which will be discussed later.
2. Since the PWM value is held constant until the next update, it results in a “stair-stepped” waveform, which introduces amplitude distortion when compared to a reference sine wave. This distortion is proportional to the first derivative of the waveform, which means that an output waveform synthesized from Figure 2 will experience more distortion when the waveform is changing rapidly near the zero crossings.

Since the distortion is related to the phase uncertainty for all non-zero derivative functions, then the sampling frequency and the output motor waveform frequency also affects it. For all carrier frequencies except 15.9 kHz, the MC3PHAC PWMs are updated at a sampling frequency of 5.3 kHz, which results in a timing jitter of ±95 µs. For a 15.9-kHz carrier, the PWMs are updated at a 4-kHz rate, with a timing jitter of ±126 µs. This results in a phase uncertainty as a function of the motor waveform frequency, which is illustrated in Figure 3. As the motor waveform frequency decreases below about 10 Hz, improvements to the phase jitter are not observed since the phase resolution of the 512-point waveform table is reached. For both update rates, the result is motor waveforms with greater accuracy than can be achieved with designs that utilize higher waveform resolution but a lower waveform update frequency.

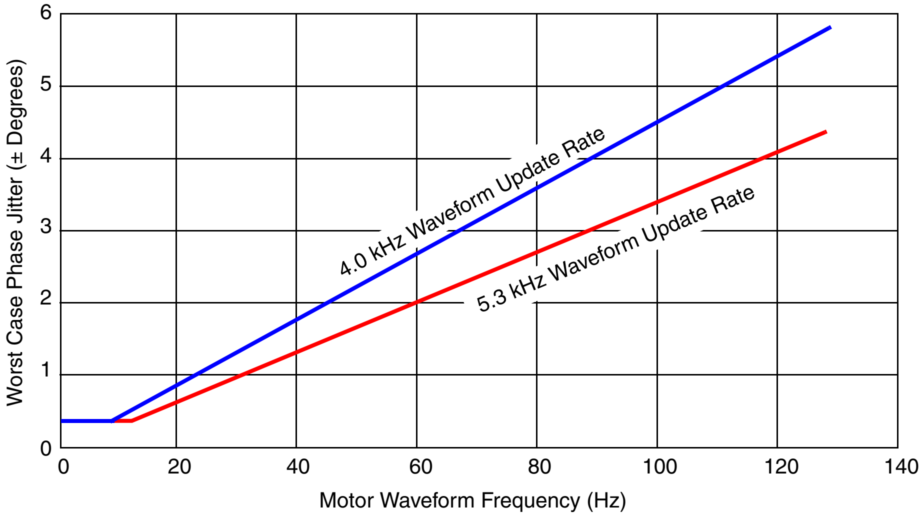


Figure 3. MC3PHAC Phase Uncertainty as a Function of Motor Waveform Frequency

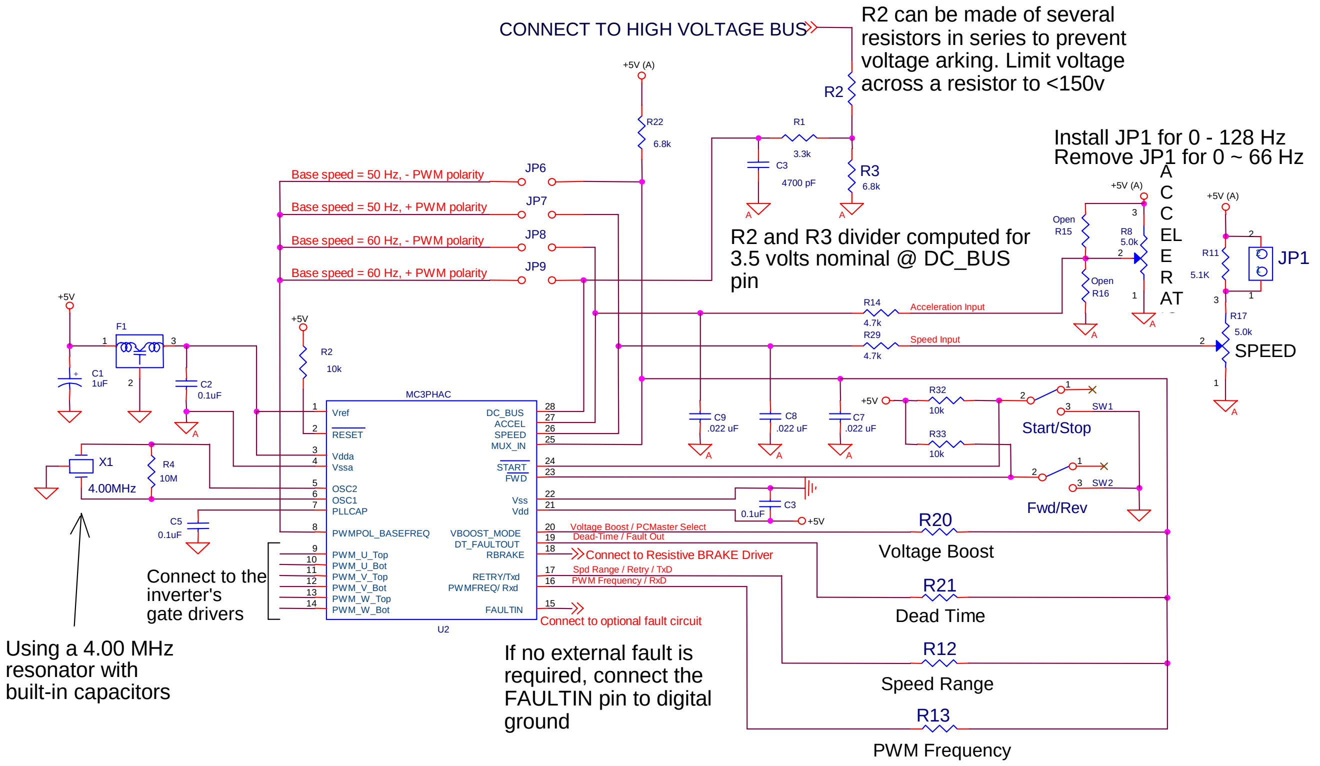
From Figure 2, it can be seen that the waveform contains a third harmonic component added to the sine wave, which results in 15% greater phase-to-phase amplitudes compared to traditional sine modulation. However, this restricts the MC3PHAC usage to three-phase loads that have a floating neutral, since a common-mode third harmonic frequency component results from this modulation technique. It also places limitations on the synthesis technique of the three-phase outputs, since the sum of the output voltage waveforms no longer equals zero.

4 Modes of Operation

The MC3PHAC will operate in either of two modes: Standalone or Host. Mode selection occurs at power-up as specified by the state of pin 20. Both modes are described in further detail below.

4.1 Standalone Mode

In this mode, the MC3PHAC operating parameters are configured during power-up via passive components connected to the device. Once the MC3PHAC determines that there is no external host (pin 20 is high), it begins interrogating the externally connected resistor network to obtain operating parameters such as Speed Range, Dead-Time, and Voltage Boost. Other parameters continue to be input in real time as the system operates, such as Start/Stop, Forward/Reverse, Motor Speed, PWM Frequency, Bus Voltage, and Acceleration. Standalone mode is the most economical mode from an overall system cost point of view, as no host controller is needed for MC3PHAC operation. Figure 4 shows an example circuit using the MC3PHAC in standalone mode.



R2 can be made of several resistors in series to prevent voltage arking. Limit voltage across a resistor to <150v

Install JP1 for 0 - 128 Hz
Remove JP1 for 0 ~ 66 Hz

R2 and R3 divider computed for 3.5 volts nominal @ DC_BUS pin

Connect to the inverter's gate drivers

Using a 4.00 MHz resonator with built-in capacitors

If no external fault is required, connect the FAULTIN pin to digital ground

Figure 4. Schematic of the MC3PHAC in Standalone Mode Operation

4.2 Host Mode

The second mode of operation is called Host Mode, which utilizes a PC running host software (available from Freescale), or a microcontroller emulating the host software commands. Instead of using discrete components to specify operating parameters, they are controlled in software, directly from the host.

Immediately following power-up, the MC3PHAC determines that an external host is present by reading pin 20 as a logic low level. After continuing to initialize to an inert, safe condition, it remains dormant, waiting for commands over its serial interface to specify operating parameters. The MC3PHAC will not allow the motor to be activated until certain critical parameters have been communicated, such as PWM Polarity and Dead-Time information. Host Mode allows an external controller to monitor and control all aspects of the MC3PHAC's operation, and permits much wider control over the system's operating environment as compared to Standalone Mode, which is illustrated by Table 2.

Table 2. Comparison Between Standalone and Hosted Operating Modes

Name	Standalone Mode	Host Mode	Description
Commanded PWM polarity	Top and Bottom All Positive or All Negative at 50 Hz or 60 Hz	Bottom Positive, Top Positive Bottom Positive, Top Negative Bottom Negative, Top Positive Bottom Negative, Top Negative At 50 Hz or 60 Hz	Specifies the polarity of the MC3PHAC PWM outputs.
Dead-time	5 to 6 µs	0 to 32 µs	Specifies the dead-time used by the PWM generator.
Fault timeout	1 second to –53 seconds	25 seconds to 4.55 hours	Specifies the delay time after a fault condition before re-enabling the motor.
Voltage boost	0% to 35%	0% to 100%	Zero hertz voltage
Maximum voltage	Fixed at 100%	0% to 100%	Maximum allowable modulation index value
Vbus decel value	Fixed at 110% of nominal Vbus	0% to 143% of nominal Vbus	Vbus readings above this value result in reduced deceleration.
Vbus Rbrake value	Fixed at 110% of nominal Vbus	0% to 143% of nominal Vbus	Vbus readings above this value result in the R brake pin being asserted.
Vbus brownout value	Fixed at 50% of nominal Vbus	0% to 143% of nominal Vbus	Vbus readings below this value result in an undervoltage fault.
Vbus overvoltage value	Fixed at 125% of nominal Vbus	0% to 143% of nominal Vbus	Vbus readings above this value result in an overvoltage fault.

In Host Mode, remote control over the Internet is even possible. By running a separate server application (also available from Freescale) connected to the MC3PHAC, a remote computer running the aforementioned host software can control a motor at one location in the world from another location in the world.

An example circuit utilizing the MC3PHAC in Host Mode is illustrated in Figure 5.

Figure 6 shows a GUI utilizing Freescale's interface host software to control the MC3PHAC.

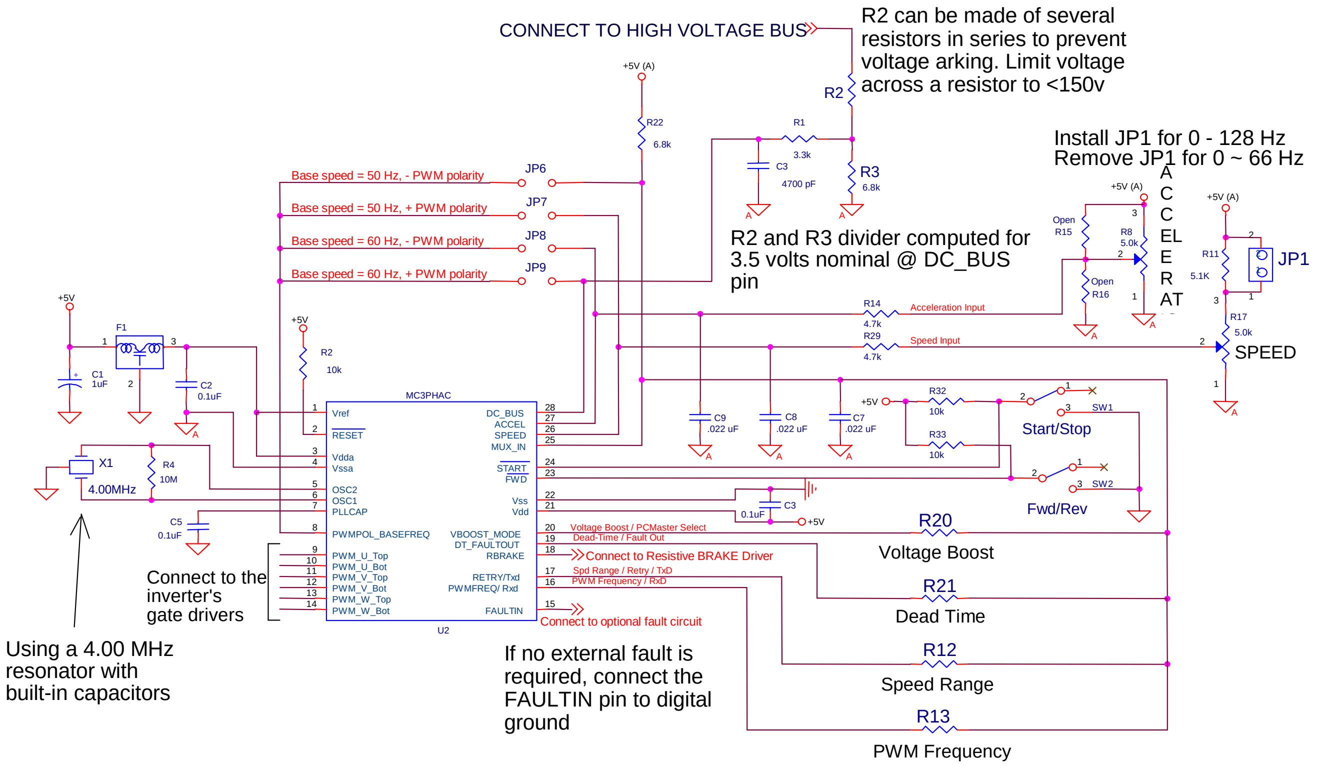
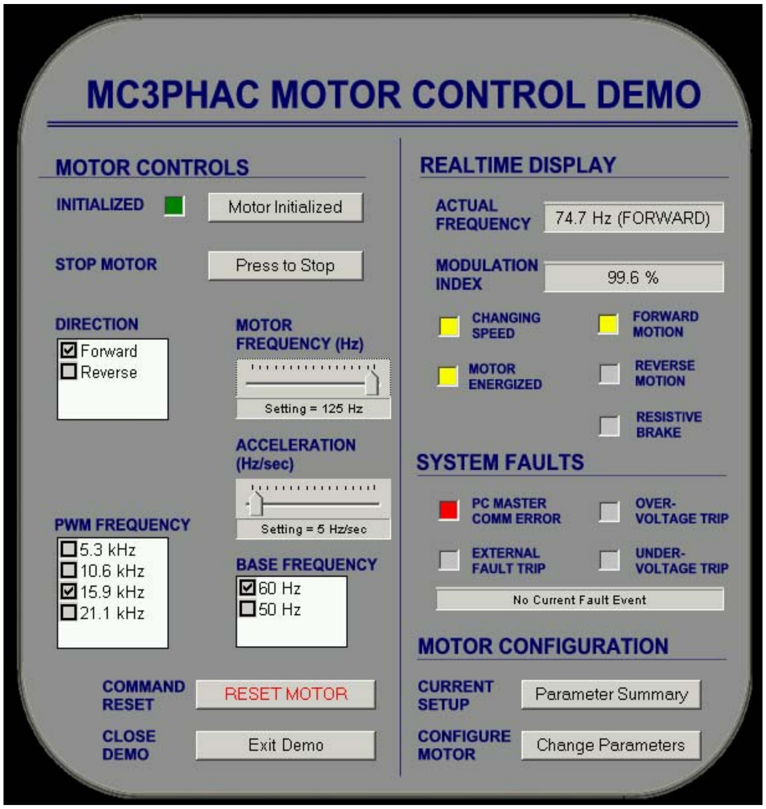


Figure 5. Schematic of the MC3PHAC in Host Mode


MC3PHAC MOTOR CONTROL DEMO

MOTOR CONTROLS

INITIALIZED 🟩 Motor Initialized

STOP MOTOR: Press to Stop

DIRECTION
☑ Forward
☐ Reverse

MOTOR FREQUENCY (Hz): Setting = 125 Hz

ACCELERATION (Hz/sec): Setting = 5 Hz/sec

PWM FREQUENCY
☐ 5.3 kHz
☐ 10.6 kHz
☑ 15.9 kHz
☐ 21.1 kHz

BASE FREQUENCY
☑ 60 Hz
☐ 50 Hz

COMMAND RESET: RESET MOTOR

CLOSE DEMO: Exit Demo

REALTIME DISPLAY

ACTUAL FREQUENCY: 74.7 Hz (FORWARD)

MODULATION INDEX: 99.6 %

🟨 CHANGING SPEED
🟨 FORWARD MOTION
🟨 MOTOR ENERGIZED
☐ REVERSE MOTION
☐ RESISTIVE BRAKE

SYSTEM FAULTS

🟥 PC MASTER COMM ERROR
☐ OVER-VOLTAGE TRIP
☐ EXTERNAL FAULT TRIP
☐ UNDER-VOLTAGE TRIP

No Current Fault Event

MOTOR CONFIGURATION

CURRENT SETUP: Parameter Summary

CONFIGURE MOTOR: Change Parameters

Figure 6. Host Software GUI Interface used with MC3PHAC

5 Bus Ripple Cancellation

In many AC drives, the inverter is powered from a DC bus with a large capacitor connected in parallel acting as an energy reservoir. To prevent fluctuations on the bus from disturbing the motor waveforms, this capacitor is often oversized, especially if a standard rectifier converter powers the bus. These fluctuations may be the result of voltage surges on the AC mains, regeneration resulting from fast deceleration of the motor, or even higher frequency ripple resulting from rectification of the AC line. Due to the high bus feedback sampling frequency with the MC3PHAC, all of these distortions can be compensated for. Every 189 or 252 μs, depending on the PWM frequency, the DC_BUS input pin is sampled, and the reading is used to compensate the modulation index in real time to regulate the motor current. While many AC drives implement a similar function, they can only compensate for lower frequency distortions since they sample the bus voltage too infrequently to permit real-time ripple rejection.

Referring to Figure 7, assuming that the transistors are driven in a complimentary fashion with zero dead-time, the equation defining the average voltage of the output waveform is given by:

$$\overline{v_o(t)} = \frac{t_h(t)}{T} V_{bus}(t) \qquad \textit{Equation 1}$$

$$\begin{aligned} where: \overline{v_o(t)} &\ is\ the\ average\ output\ voltage \\ t_h(t) &\ is\ the\ high\ time\ of\ the\ PWM\ waveform \\ T &\ is\ the\ PWM\ period \\ V_{bus}(t) &\ is\ the\ voltage\ of\ the\ DC\ bus \end{aligned}$$

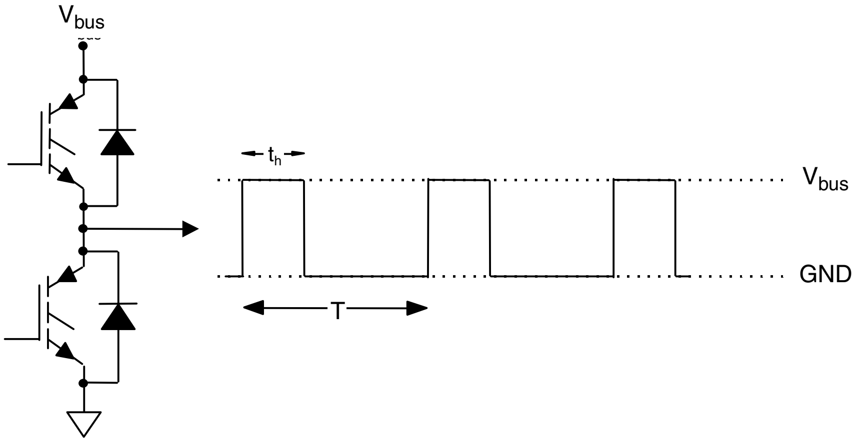


Figure 7. PWM Waveforms Generated from a Half-Bridge

Notice that equation 1 does not assume that V_{bus} is a constant, but rather a function of time t. However, let's assume that there exists an optimum value of V_{bus} that we will call V_{norm} such that when $V_{bus}(t)$ equals V_{norm}, then $\overline{v_o(t)}$ will equal the desired value $\overline{v_o(t)}_{norm}$ based on the specified PWM high time and period. However, when V_{bus} differs from V_{norm}, it is still possible to drive $\overline{v_o(t)}$ to equal $\overline{v_o(t)}_{norm}$ by applying a correction factor to the modulation term $t_h(t)/T$ in equation 1, as indicated in equation 2.

$$\overline{v_o(t)}_{norm} = \frac{t_h(t)\left[\dfrac{V_{norm}}{V_{bus}(t)}\right]}{T} V_{bus}(t) \qquad \textit{Equation 2}$$

where: V_{norm} *is the optimum or reference value for* $V_{bus}(t)$

[ ] *term is the correction factor*

Since the $V_{bus}(t)$ terms cancel out, we see that any perturbations in the bus voltage do not affect the output voltage. Also, since the ratio $t_h(t)/T$ will always be a positive fractional value, we must make sure that whatever waveform is desired on the output will be properly scaled and biased to reflect this. For example, if sinusoidal modulation is desired, then the sine wave amplitude should be scaled so as to not exceed a peak-to-peak value of 1, and the waveform should be biased at ½ in order to achieve full utilization of the dynamic range. If we modify equation 2 to reflect this, and accounting for all three phases of the output, we obtain:

$$\overline{v_o(t,x)}_{norm} = \left(\frac{1}{2} + \frac{M}{2}\sin\left(\omega_o t + \frac{2\pi(x-1)}{3}\right)\right)\left[\frac{V_{norm}}{V_{bus}(t)}\right] V_{bus}(t) \qquad \textit{Equation 3}$$

where: x *is the output phase number* (1, 2, 3)

ω_o *is the frequency of the output waveforms*

M *is the modulation index* (*0 through 1*)

Equation 3 results in total bus ripple cancellation of the output waveforms. However, this is not the optimal situation because the output waveforms are biased around a fixed voltage of ½ V_{norm}, NOT ½ $V_{bus}(t)$, as they should be. Note that the modulation waveform consists of two terms; a DC term of ½, and an AC sine term. In equation 3, the correction is being applied to BOTH terms, when in fact, it should only be applied to the AC term. If we decouple the correction from the DC term, it will allow the AC waveform to "auto-center" itself in the dynamic range represented between ground and $V_{bus}(t)$. Rewriting equation 3 to achieve this decoupling, we obtain:

$$\overline{v_o(t,x)}_{norm} = \left(\frac{1}{2} + \left[\frac{V_{norm}}{V_{bus}(t)}\right]\frac{M}{2}\sin\left(\omega_o t + \frac{2\pi(x-1)}{3}\right)\right) V_{bus}(t) \qquad \textit{Equation 4}$$

This is the technique used on the MC3PHAC. $V_{bus}(t)$ is sampled at every PWM update interval (189 or 252 µs), and is divided into a number which represents a V_{norm} value of 3.5 volts. The resulting correction factor is then applied only to the modulation index (M) to correct any distortions in the output waveforms resulting from perturbations in $V_{bus}(t)$. Because the correction factor is not applied to the DC term, a noise artifact will appear on the bias. However, since this noise is a common mode signal to all three output waveforms, it will be rejected by the motor, assuming its neutral node is floating.

Figure 8 shows actual current waveforms taken from a ½ hp motor driven from a power stage supplied by a single-phase 115 VAC input at 60 Hz with excessive bus ripple present. In the first waveform, the MC3PHAC bus ripple cancellation feature was disabled, as evident by the distortion in the waveform. The second waveform shows the exact same conditions, but this time with the bus ripple cancellation feature enabled.

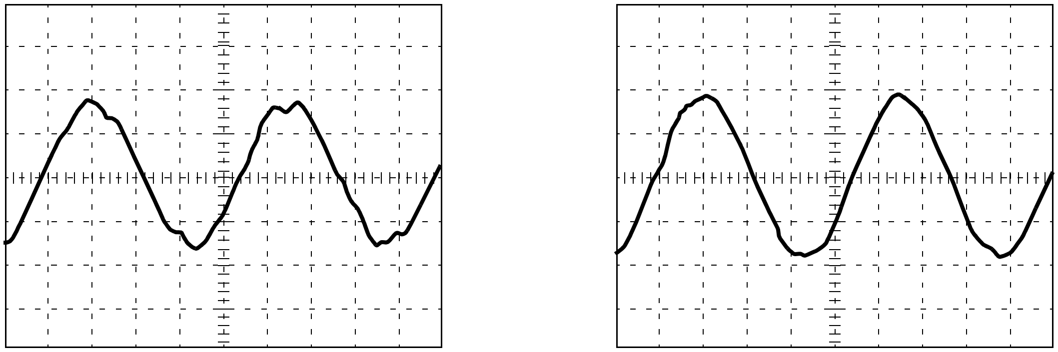


Figure 8. Effect of Bus Ripple Cancellation with the MC3PHAC

6 Velocity Pipelining and Interpolation

Assuming that the voltage on the ACCEL pin is held constant, or the Acceleration value is set to a constant via host software, the MC3PHAC will generate a linear velocity profile. To obtain other types of profiles, the acceleration value must be changed dynamically while the velocity is ramping. For example, to obtain a parabolic velocity profile, the acceleration must be changed in a linear profile while the velocity is ramping.

The velocity profiler in the MC3PHAC is not only responsible for controlling the speed of the motor, but the motor voltage as well, since the two are related in a volts-per-hertz controller. To implement all of the features incorporated in the MC3PHAC velocity profiler requires many calculations per second, which impacts how frequently the velocity can be updated. If the calculations are performed too infrequently, a stair-stepped velocity profile is created, which can result in torque perturbations and vibration during ramping.

The MC3PHAC employs two techniques that work in tandem to eliminate this problem. The first is velocity piplining, which is illustrated in Figure 9.

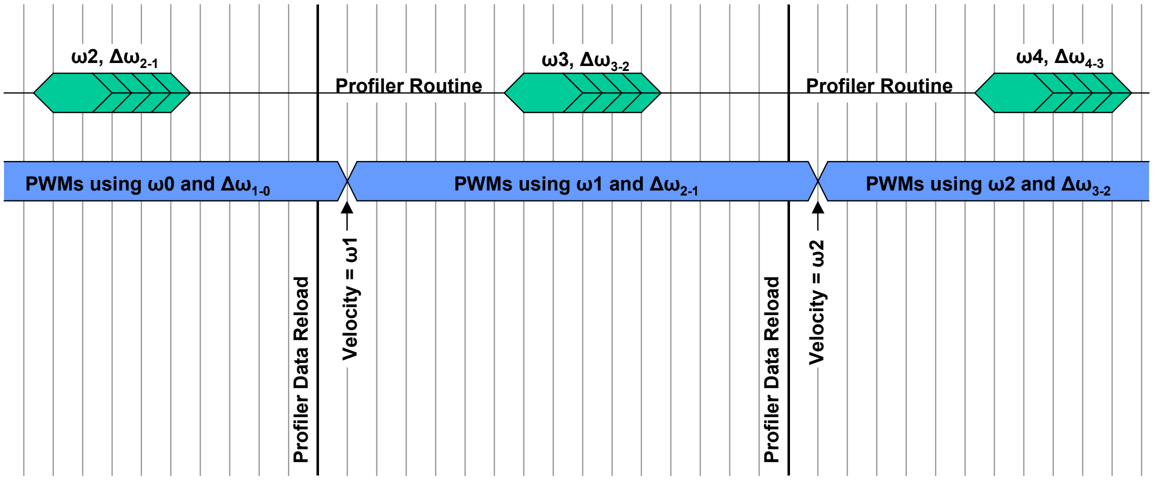


Figure 9. Velocity Pipelining

Each vertical partition represents a PWM update interval. As can be seen, at every 16th update interval, the profiler is triggered in order to synthesize new velocity and voltage information. In this particular illustration, the first triggering of the profiler generates a velocity labeled w_2. However, notice that the PWM waveforms applied to the motor will not reflect this velocity until much later. This “pipelined” effect allows PWM waveforms to be constructed based on older velocity data, while the profiler is generating new velocity data at the same time. Since the absolute time reference of the profiler velocity output waveform is arbitrary (with the exception of deceleration moderation in the presence of excessive regeneration), the phase delay represented by this pipelining will not have an adverse affect on system performance.

To further improve the resolution of the velocity waveform, the MC3PHAC employs an interpolation technique that allows the velocity waveform to be updated with a new value at each PWM update interval. Figure 10 shows an example velocity ramp waveform, where each discrete velocity output value from the profiler is represented by a circular point. These updates occur every 3 or 4 ms (depending on the PWM frequency), and can result in less than optimum motor performance during acceleration and deceleration.

Each time the profiler is triggered, three values are supplied as outputs; the old velocity, the “delta” velocity, and the modulation index (not shown in Figure 10). During each PWM update interval, the delta velocity value is divided into finer velocity increments. By adding these increments to the old velocity value each time the PWMs are updated, a new velocity curve with 16x more resolution is generated, as shown in Figure 10. As a result, the motor transitions from one speed to the next with extremely smooth velocity performance.

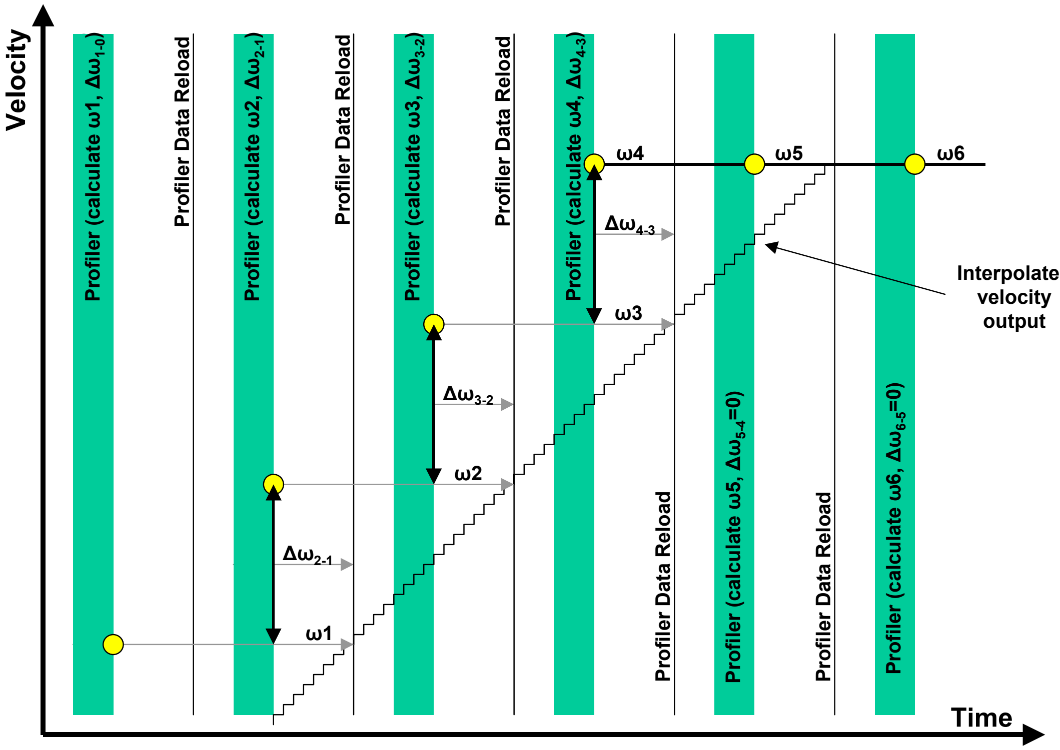


Figure 10. Velocity Interpolation

7 System Monitoring and Protection

Since the MC3PHAC is designed to control power inverters that handle dangerously high voltages and currents, it incorporates an extensive array of features for system monitoring and protection. In some cases when a problem is detected, the MC3PHAC responds immediately in an attempt to mitigate the situation. In other scenarios, the PWMs are immediately shut down until the offending problem is removed, and a timeout has completed indicating it is safe to restart. For two of the failure modes (low V_{DD} and lost crystal detection), the MC3PHAC will reset, resulting in the PWMs immediately going to a high-impedance state, and forcing a reset of all external hardware connected to the reset pin. Each protection feature is discussed in further detail below, in order of severity of the problem.

- **High Bus Voltage** — Since the type of PWMs supplied by the MC3PHAC to an AC drive results in full 4 quadrant operation of the inverter, it is possible for energy to be transferred from the motor back to the DC bus. However, in many cases, this energy is prevented from returning back to the AC mains, and is stored in the bus capacitor as $½CV^2$. In most cases, this scenario is the result of aggressive deceleration of the motor. If the bus voltage exceeds the “V_{bus} Decel Value” limit described in Table 2, the MC3PHAC will reduce the deceleration in an attempt to regulate the regeneration process. Also, if the bus voltage exceeds the “V_{bus} Rbrake Value” limit described in Table 2, the MC3PHAC will activate the RBRAKE pin, which is intended to turn on a resistive load across the capacitor to dissipate the regenerated energy instead of storing it. Figure 11 shows waveforms acquired via the PC host software of an acceleration and deceleration cycle of a ½ hp motor with its corresponding effect on the bus voltage. In this case, only resistive braking was used to limit the bus voltage, with the “V_{bus} Rbrake Value” set to its default value of 110%.

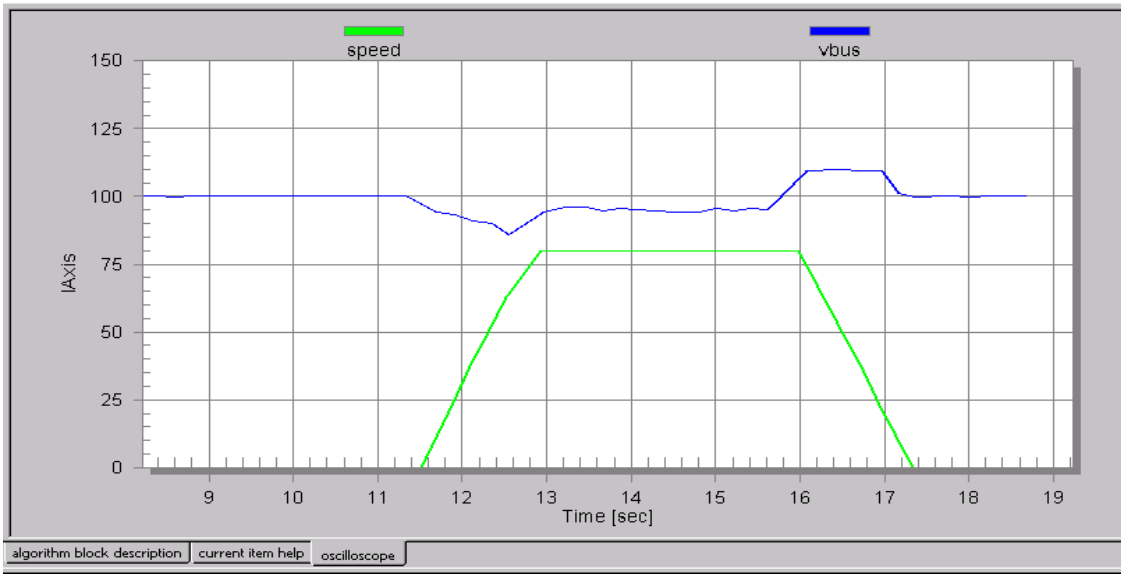


Figure 11. DC Bus Voltage During Acceleration and Deceleration with RBRAKE Clamping

- **Excessively High Bus Voltage** — If the techniques described above are not successful in limiting the voltage buildup on the DC bus capacitor, and the bus voltage exceeds the “V_{bus} Overvoltage Value” limit described in Table 2, the PWM outputs are immediately driven off. They will remain off until the bus voltage drops to within safe limits, AND a specified timeout has occurred indicating it is safe to re-energize the inverter.
- **Low Bus Voltage** — If the bus voltage drops too low (such as during a brownout condition), certain systems which are powered from the bus may function erratically, causing other system problems. If the bus voltage falls below the “V_{bus} Undervoltage Value” limit described in Table 2, the PWM outputs will be disabled and re-enabled as described above for an overvoltage condition.
- **External Fault Condition** — The MC3PHAC incorporates a special input called the FAULT IN pin to allow processing of other system faults. It is up to the user to determine what system parameter(s) should be monitored by this pin. Unlike the fault modes discussed previously which are based on sampling the DC BUS IN pin at the PWM update sampling frequency, this is a digital input which immediately drives the PWMs off when asserted. Once the input is negated, the PWMs will be re-enabled after a specified timeout has occurred indicating it is safe to re-energize the inverter.
- **Lost Clock Detection** — Loosing the input clock to the MC3PHAC (or any standard micro or DSP for that matter) can represent a potentially dangerous condition in a motor control system. In fact, some regulatory agencies are now mandating a “dead-crystal” test for certain appliance applications to verify that nothing is left energized which could cause a safety hazard. In the case of an AC motor control system, the most likely failure mechanism is that the PWM signals would freeze in their present condition, leaving certain transistors in the inverter turned on. Such a condition could easily destroy the motor, the inverter, or both. With the MC3PHAC, this problem is eliminated since the device will reset the system and disable the PWM outputs immediately following the loss of the input clock.

- **Low V_{DD} Protection** — As with the loss of the input clock, a low value of V_{DD} could result in a dangerous system failure since the MC3PHAC and other circuitry driven off of V_{DD} could malfunction. The MC3PHAC contains an on-board voltage monitor which will reset the system in the event that V_{DD} falls below 4 V. This permits 5-V power supplies to be used, regardless of whether their outputs are regulated to a 5 or 10% tolerance.

8 Conclusion

A device for controlling the speed of AC induction motors has been presented which can be adapted to most open-loop, volts-per-hertz applications with no programming required. This can significantly reduce the up-front development effort and cost, while maintaining flexibility to fit a myriad of variable-speed applications.

The MC3PHAC is offered in several standard package types. There are two 28-pin packages: 28-pin, .6″ wide, plastic DIP and a 28-pin plastic SOIC. A 32-pin (QFP) quad plastic flat pack is also available. All packages are characterized for operation from –40° to 105° Centigrade.

Pricing for the MC3PHAC in all packages is <50k = $5.50, >50k = $4.25, 1 million = $1.85

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.
© Freescale Semiconductor, Inc. 2005. All rights reserved.