

## AC Induction Motor (ACIM) Control using a Digital Signal Controller (DSC)

<sup>1</sup>D. Diarisso, <sup>2</sup>I. Ly, <sup>3</sup>G. Sow, <sup>1</sup>O. Sow, <sup>1</sup>I. Gaye, <sup>1</sup>F.I. Barro and <sup>1</sup>G. Sissoko  
<sup>1</sup>Faculty of Science and Technology, University Cheikh Anta Diop, BP 5005,  
Dakar-Fann, Senegal

<sup>2</sup>Polytechnique School of Thies, EPT, Thies, Senegal

<sup>3</sup>Superior School of Dakar, ESP, Sénégal

**Abstract:** This study describes the use of a digital signal controller for an AC Induction Motor (ACIM) control through a three phase inverter. The main feature of the work is the simplicity of the hardware-only a 16 bit fixed point microcontroller with associated external conditioning integrated circuits. This lead to an extremely simple, low-cost and reliable system. A laboratory inverter prototype using a Microchip DSPIC33FJ128MC804 digital signal controller is developed and typical results are presented.

**Keywords:** ACIM, Digital Signal Controller (DSC), Pulse Width Modulation (PWM)

### INTRODUCTION

Pulse Width Modulation (PWM) techniques of various forms are available in the literature for the control of voltage, frequency and spectral response of three phase inverters (NEC, 2004; Zhou and Wang, 2002; Copeland *et al.*, 1999). Microprocessors ( $\mu$ P), DSPs and  $\mu$ controllers are used for implementation of PWM schemes efficiently (NEC, 2004; Copeland *et al.*, 1999; Salam and Salim, 2001). The PWM switching points are computed on-line and the PWM patterns are realized using the  $\mu$ P (or DSP), I/O interfaces and Programmable Timers (Microchip Technology Inc, 2009; Zambada and Microchip Technology Inc, 2005; Kumar *et al.*, 2010; Swamy and Kumar, 2008).

In this study, a Digital Signal Controller (DSC) is used to generate the Pulse Width Modulation (PWM) pulses for the control of an AC Induction Motor (ACIM) by a three-phase inverter (Kumar *et al.*, 2010; Swamy and Kumar, 2008; Islam *et al.*, 2004; Buso and Mattavelli, 2006). The used DSC will first be described, followed by the implementation of the PWM pulses; the resulting signals are then presented.

### MICROCHIP DSPIC33FJ128MC804

This study describes the use of Microchip DSPIC33FJ128MC804 to generate the Pulse Width Modulation (PWM) pulses for a three-phase inverter system.

There are numerous internal peripherals and the details on this DSC can be found in (Microchip

Technology Inc, 2009). The main peripheral that will be utilized in the waveform generation is the 6 channels PWM module also called Motor Control PWM module (MCPWM) which is briefly described. Note that there are two Motor Control PWM modules noted MCPWMx. In the following, we refer only to MCPWM1 (Fig. 1) since MCPWM2 has just one pair of PWM channels.

MCPWM1 can be divided into the following units (Zambada and Microchip Technology Inc, 2005): the 16 bit time base for all PWM channels, the duty generator and the dead time control for each PWM channel, the PWM override logic.

Duty cycle generators create pulses that contain the preprogrammed duty cycle information. Dead-time units offset the pulses to prevent shoot through in driving the inverter transistors. PWM override logic allows the output signals to be modified based on fault conditions and/or program instructions. For example, an output signal from this block can be inverted if negative polarity is selected or can be forced to a programmed value in the OVDCON register.

The DSPIC33FJ128MC804 allows the generation of six independent PWM signals; Fig. 2 illustrates the output of a centre aligned PWM. The width of the pulse is determined by the content of the pulse width register (PxDCy), where y denotes a particular PWM channel. In a centre-aligned PWM the value in the pulse width register effects both edges of the output signal symmetrically, whereas in the edge-aligned, one of the edges is fixed. The minimum pulse width depends on the resolution of the PWM period, PxTPER. For 40 MHz CPU clock and PxTPER value = 0x7FFE, the resolution

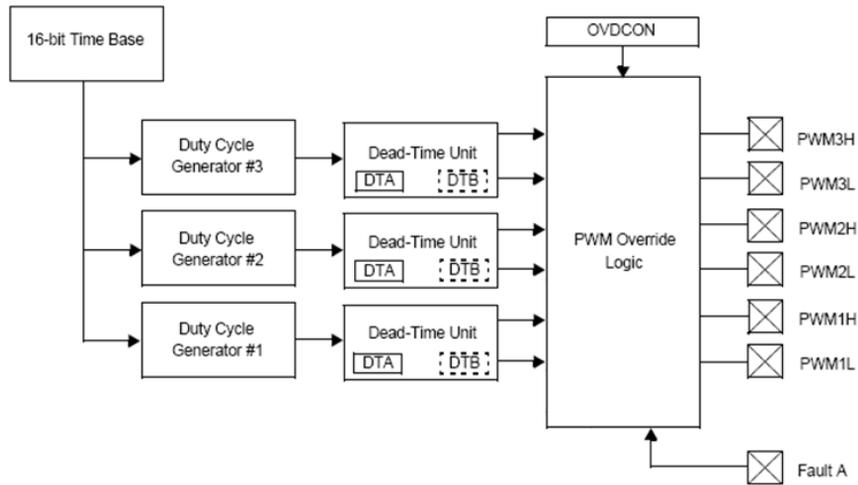


Fig. 1: MCPWM block diagram

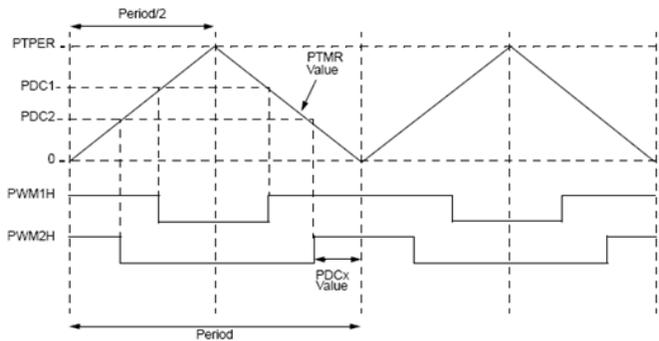


Fig. 2: Center-aligned PWM time diagram

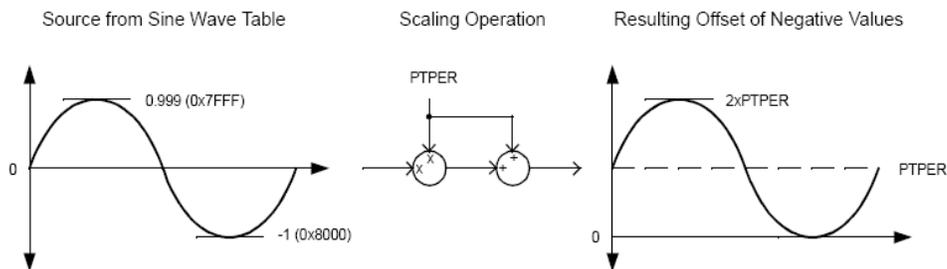


Fig. 3: Scaling of sine wave table values for the duty cycle registers

is 25 ns (Zambada and Microchip Technology Inc, 2005). The period of the pulse can be adjusted by changing the content of the pulse period register, P<sub>x</sub>TPER.

The value in the P<sub>x</sub>TPER is related to the PWM frequency as:

$$P_xTPER = (f_{cy}/f_{PWM}-1):2$$

$f_{PWM}$  is the PWM switching frequency and  $f_{CY}$  is the device operating frequency.

### IMPLEMENTATION OF THE PWM PULSES

To generate the three phase outputs, sinusoidal data for a complete electrical cycle is provided in a 64-word table. The data is in 16-bit signed fractional format normalized to the range-1 to 1. A variable called *Phase* is used as a 16-bit pointer to the table with 0x0000 representing 0° and 0xFFFF representing 359.99°. At each PWM period interrupt (50 μs), the *Delta\_Phase* variable is added to *Phase*. The value of *Delta\_Phase* determines

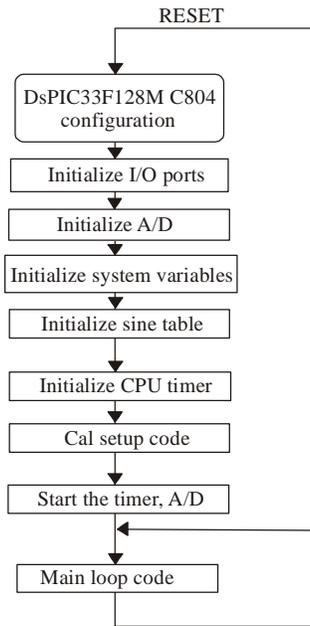


Fig. 4: Software flowchart

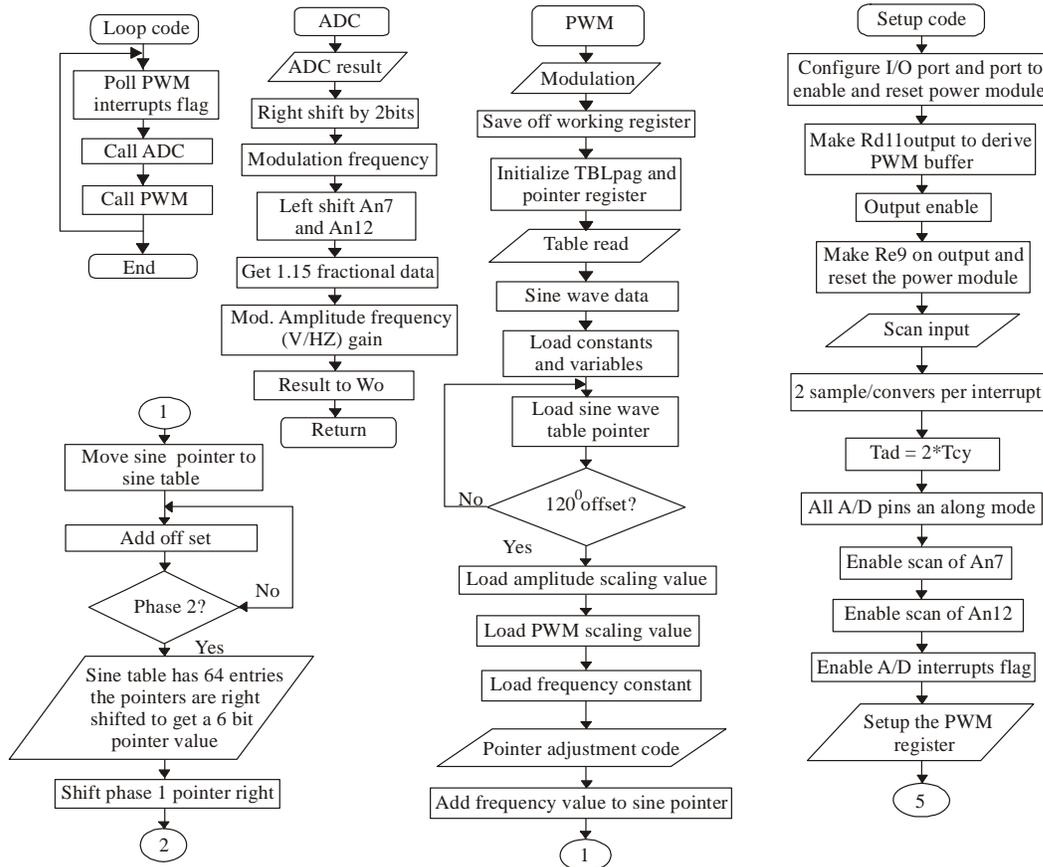


Fig. 5: Software flowchart (continued)

how fast the code moves through the sinusoidal data table and, as a result, sets the modulation frequency.

The *Delta\_Phase* variable is calculated as follows:

After the Phase variable has been adjusted by *Delta\_Phase*, two additional table pointers are calculated for the 2nd and 3rd motor phases by adding a constant offset to Phase. Assuming a 16-bit pointer, a value of 0x5555 provides a 120° offset and a value of 0xA555 gives a 240° offset.

Next, the three 16-bit pointers are right shifted by 10 to get the most significant 6 bits of information. Since we only have a 64-entry table, we only need a 6-bit pointer.

Finally, the three-phase pointers are added to the base address of the sine wave table stored in program memory and the sine values are retrieved. Now that we have the sine values, they need to be scaled for the desired modulation amplitude and PWM duty cycle range. First, the look-up values are multiplied by the value in the PTPER register to establish the amplitude. The PTPER value is then added to the amplitude to ensure that the resulting duty cycle value is positive. The Fig. 3 illustrates this scaling process. Since the duty cycle registers have twice the resolution compared to the PTPER register, a maximum value of 2xPTPER is required.

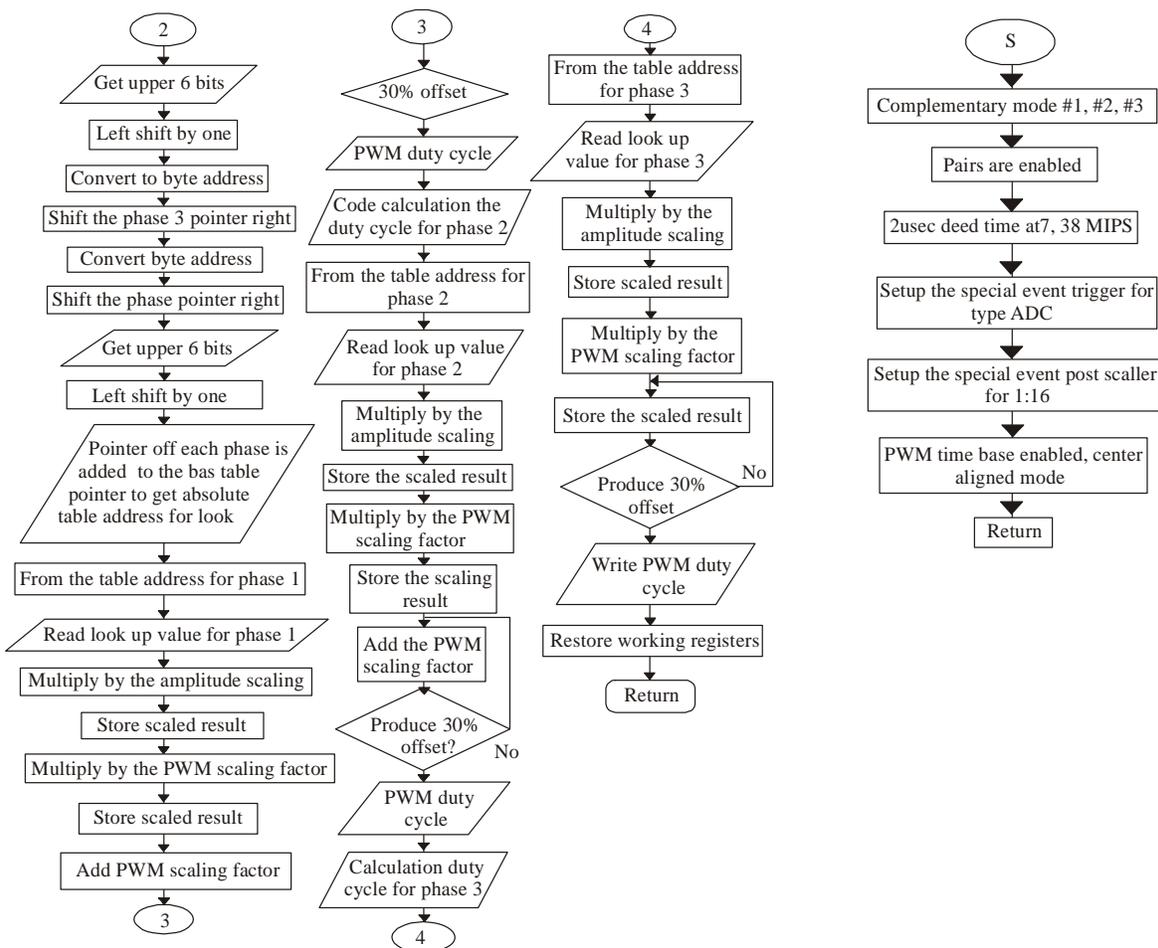


Fig. 6: Software flowchart (continued)

The ADC is enabled to scan the analog inputs AN7 and AN12, which are both connected to potentiometers on the experimental board; ADC conversions are triggered from the PWM module every 16 PWM cycles and the ADC interrupts the CPU after two sample/convert sequences.

With this configuration, the ADC effectively samples AN7 and AN12 at a frequency of 500 Hz. The *ReadADC* subroutine is called at each ADC interrupt to read the conversion values and calculate the Voltage-Frequency (VF) profile.

The potentiometer connected to AN7 sets the drive frequency. The 10-bit ADC result is right shifted by 2 and written to the *Frequency* variable. The frequency resolution is 0.244 Hz/bit, so the potentiometer can adjust the drive frequency up to 62 Hz with this scaling. The AN2 potentiometer is used to set the voltage-to-frequency ratio, which determines the slope of the VF profile. The VF profile has maximum slope with the potentiometer set to full value. The ADC results for AN7 and AN12 are left

shifted to convert them to fractional values, which simplifies the mathematics. In fractional mathematics,  $0x7FFF = 0.999$ . The two fractional values are multiplied using the fractional MPY instruction.

The result of the multiply is a fractional value that can be used to scale the modulation voltage between 0 and 100%. This value is stored in the *Amplitude* variable. Together, the *Frequency* and *Amplitude* values specify the input parameters for the sine wave generation. The value of *Amplitude* is limited, so that overmodulation will not occur in the *PWM modulation* routine.

In this application, the ADC and VF processing occurs at a 500-Hz rate to generate new voltage and frequency values.

The software process can be described by the following flowchart (Fig. 4) which is more detailed in Fig. 5 to 7.

The DSPIC33FJ128MC804 configuration bits are set before doing anything; the I/O ports, the system variables,

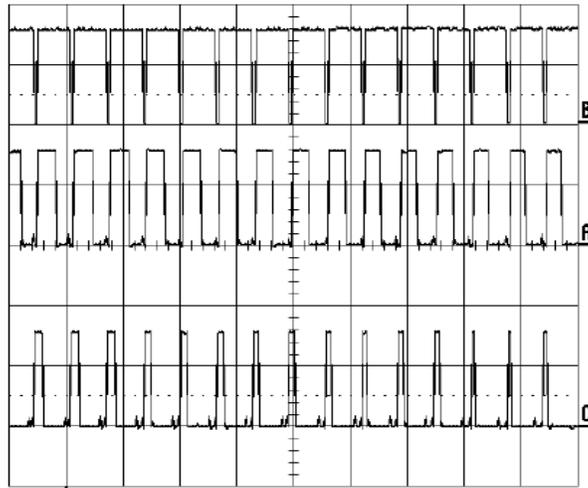


Fig. 7: Generated PWM signals

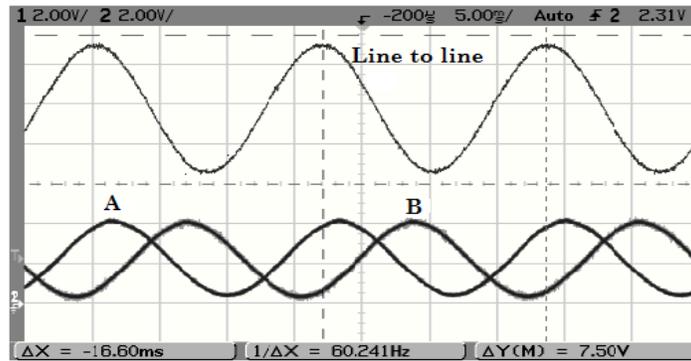


Fig. 8: Motor voltages

the sine table are initialized and then the ADC module and the PWM module are configured.

The ADC and PWM modules are started and the interruptions enabled before the main code begin.

Figure 5 present the main loop code and both PWM and ADC routines and setup code; PWM and ADC routines are continued in Fig. 6.

**Experimental results:** The obtained PWM signals are shown in Fig. 7 and the phase currents are presented in Fig. 8.

### CONCLUSION

This work presented an implementation of sinusoidal control of an ACIM through three phase inverter with dsPIC33FJ128MC804 Digital Signal Controller . The obtained line to line voltage and phase to neutral voltages are sinusoidal as expected with a 120° phase shift between phase voltages.

The next stage is the implementation of vector control of an ACIM using the same Digital Signal Controller. The choice of the dsPIC33F family of Digital

Signal Controllers is mainly guided by the fact that it is more powerful than simple  $\mu$ controllers due to its DSP unit and more cheaper than typical DSP.

### REFERENCES

- Buso, S. and P. Mattavelli, 2006. Digital Control in Power Electronics. Morgan and Claypool, San Rafal, Calif.
- Copeland, M., 1999. Infineon Technologies, AP0836 Application Note: Space Vector Modulation and Overmodulation with an 8-bit Microcontroller.
- Islam, A.E., K.M. Rahman, M.A. Choudhury, S.J. Al-Kadry, M. Islam, S.M. Rahman and A.M. Rizwan, 2004. Low-cost implementation of high-resolution PWM scheme for adjustable speed drives. 3rd International Conference on Electrical and Computer Engineering (ICECE), Dhaka, Bangladesh, pp: 335-338.
- Kumar, K.V., A.M. Prawin, J.P. John and S.S. Kumar, 2010. Simulation and comparison of SPWM and SVPWM control for three phase inverter. ARPN J. Eng. Appl. Sci., 5(7): 61-74.

- Microchip Technology Inc., 2009. DS70291D, High-Performance, 16-bit Digital Signal Controllers.
- NEC, 2004. 78K0 Application Note: An Introduction to Space Vector Modulation using NEC's 8-bit Motor Control Microcontrollers. NEC Corporation.
- Salam, Z. and K.M. Salim, 2001. Generation of Pulse Width Modulation (PWM) signals for three-phase inverter using a singlechip microcontroller. *J. Teknologi.*, 34(D): 1-12.
- Swamy, R.L. and P.S. Kumar, 2008. Speed control of space vector modulated Inverter driven induction motor. *Proceedings of the International Multi Conference of Engineers and Computer Scientists, II.*
- Zambada, J. and Microchip Technology Inc., 2005. DS93004A, Driving an ACIM with the dsPIC DSC MCPWM Module.
- Zhou, K. and D. Wang, 2002. Relationship between space vector modulation and three phase carrier-based PWM: A comprehensive analysis. *IEEE T. Indus. Electr.*, 49(1): 186-196.