Application of DSC in High Performance PMSM Control

Thit Waso Khine
Electronic Engineering Department, Technological University (Lashio)
thitwasokhine@gmail.com

Abstract

Digital signal controller DSC, a hybrid of microcontrollers and digital signal processors, is widely used in high performance industrial servo systems due to its rich of control oriented peripherals, its inherent computational power due to its single cycle multiply-accumulate MAC unit, to get very fast cycle times and wide closed-loop current control bandwidths. This paper discusses the use of DSC in high performance PMSM control systems from practical point of view. Firstly, sensing phase current of PMSM motor is filtered using infinite impulse response (IIR) filter. After that, all phase currents are transformed into two phase currents by Clarke and Park transform, in which the required sine and cosine values are calculated real time due to single cycle MAC. Finally, discrete proportional integral (PI) controller is implemented using DSP to generate control voltage. In this work, TMS320F28335 DSC is used to make performance comparison between DSC and general purpose controller.

Keywords: Digital Signal Controller, Digital Signal Processing, IIR filter, PI controller

1. Introduction

High performance motor control requires smooth rotation down to stall, fast accelerations and decelerations, full control of torque. All of these features in turn require a lot of computations in real time. System variables must be always monitored and control system must react accordingly in a possible shortest time. A cost-effective solution to industrial servo design is based on advances in digital signal processor (DSC). Powerful DSCs such as C2000 series from Texas Instrument and dsPIC series from Microchip, incorporate all the necessary circuitry required by power electronics applications such as: PWM channels, A/D converters, CAN interface, serial ports, event timer, and encoder interface. One of the most common high performance industrial servo motors is permanent magnet synchronous motor (PMSM). Control of PMSM is mainly controlling stator magnetic field. Overall control structure of PMSM using PI controller is shown in figure 1.

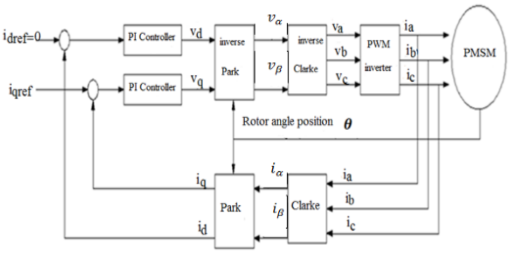

Figure 1. Control of PMSM

The required steps for motor control are the followings. First step is sensing the phase currents i_a, i_b, i_c. Second step is transforming these phase currents to torque generating portion (i_q) and magnetizing portion (i_d), using Clarke and Park Transform. Third step is to generate correcting voltage v_d and v_q, using PI controller and the fourth is transforming v_q and v_d to motor terminal voltage v_a, v_b, v_c. Phase currents i_a, i_b, i_c are 120° apart each other and vector sum of these currents, net current vector, has both magnitude and direction or angle. Under normal operation, the angle θ between net current vector and motor rotor position must be 90° to get the highest efficiency. The next sections describe the practical realization of all of above control steps.

2. Phase Currents Filtering

All control start from phase currents sensing. Because of high frequency switching, sensed current values required filtering. Hardware filtering circuit such as op-amp, resistors and capacitor combination circuit can be used, but it lacks flexibilities, fixed at only one cut-off frequency. Better approach is to use DSP filtering. Normally biquad filter, or IIR filter of second order is used. A standard biquad filter has the following transfer function in the Laplace-Domain which is described as in (1).

$$G(s) = \frac{b_{2s}s^2+b_{1s}s+b_{0s}}{a_{2s}s^2+a_{1s}s+a_{0s}} \tag{1}$$

Where b_{2s}, b_{1s} ,b_{0s} , a_{2s}, a_{1s} and a_{0s} are s-domain coefficients which can be easily adjusted according to the system requirement. Control unit must be able to do the calculation of this transfer function in real time. The transfer function needs to be transformed to time discrete domain by z-transformation and coefficients need to be normalized. This is done by the following equations.

$$b_{2z} = (b_{0s}.T^2 + 2b_{1s}.\mathrm{T} + 4b_{2s}) / (T^2 - 2a_{1s}.\mathrm{T} + 4a_{2s}) \tag{2}$$

$$b_{1z} = (2b_{0s}.T^2 - 8b_{2s})/(T^2 - 2a_{1s}.\mathrm{T} + 4a_{2s}) \tag{3}$$

b_{0z}= (b_{0s}. $T^2 - 2b_{1s}$.T + $4b_{2s}$)/ ($T^2 - 2a_{1s}$ .T + $4a_{2s}$) (4)

a_{2z}= (T^2+ $2a_{1s}$ .T +$4a_{2s}$))/ (T^2 −$2a_{1s}$ .T + $4a_{2s}$) (5)

a_{1z}=($2T^2$ −$8a_{2s}$) /($T^2 - 2a_{1s}$.T + $4a_{2s}$) (6)

Where b_{2z}, b_{1z}, b_{0z}, a_{2z}, a_{1z} are actual coefficients in z-domain and T is the current phase current sampling period. Actual filtering process is done in time domain in the following structure which is described as in (7).

Y (n) = X (n).b_{0z} + X (n−1).b_{1z} + X (n−2).b_{2z} + Y(n−1).a_{1z} + Y (n−2). a_{1z} (7)

Where X (n) is the actual current input sample, while Y(n-1) is the filter output of the last cycle.

For servo system that has outer control loops such as velocity and position control loop, DSC reads position or velocity from the sensors attached to rotor. The IIR filter must also be used for these velocity and position values, especially at very low speed to avoid quantization error, in which analogue filters cannot be used.

3. Current Transformation

The next step is to transform all phase current values measured to equivalent two phase system using Clarke transformation [1], as described in (8).

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \quad (8)$$

Transforming two phase system consisting of i_α and i_β to equivalent two phase system consisting of i_q and i_d is called Park transform and is described as in (9).

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \quad (9)$$

Where θ is the angle between net current vector and rotor. The reason for these transforms is to find the current components, one is perpendicular to rotor axis and one is parallel to rotor axis as shown in figure 2, where i_s is the vector sum of i_a, i_b, i_c.

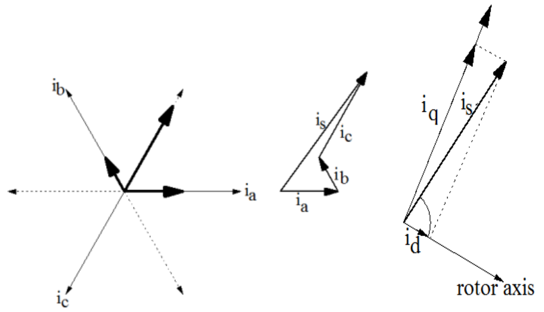

Figure 2. Motor current decomposed to perpendicular and parallel to rotor axis

For permanent magnet synchronous motor, i_d component must be zero and direction of i_q component must be perpendicular to PMSM motor rotor position to get the highest efficiency. The magnitude of i_q determines the amount of generated torque of the motor.

3.1. Calculation of Sine and Cosine in DSC

Park transform includes calculation of sine and cosine functions which requires high computing power. In normal microcontroller, look up table can be used. But for DSC, because of single cycle MAC, all trigonometric functions can be calculated in real time. First, The angle range θ = 0 to π/2 is normalized to the range θ' = − ½ to ½ via the transformation $\theta' = \frac{2}{\pi}\theta - \frac{1}{2}$ [2].

sin(θ') = a_0–b_1a+ a_2a^2 –b_3a^3+ a_4a^4–b_5a^5+ a_6a^6 (10)

cos(θ') = a_0+b_1a+ a_2a^2 +b_3a^3+ a_4a^4+b_5a^5+ a_6a^6 (11)

In practice, even and odd parts are calculated first described as in (12) and (13).

A = a_0+ a_2a^2 +a_4a^4+ a_6a^6 (12)

B = b_1a+b_3a^3+b_5a^5 (13)

From which both the sine and cosine values are trivially obtained as in (14) and (15).

sin(θ') = A – B (14)

cos(θ') = A + B (15)

Where a_0= 0.707106781187
a_2= −0.872348075361
a_4= 0.179251759526
a_6= −0.0142718282624
b_1= −1.11067032264
b_3= 0.4561589075945
b_5= −0.0539104694791

The absolute error of this calculation is smaller than 6.5 x 10^{-6}, and $\sin^2(x) + \cos^2(x)$ never exceed 1.

4. PI Controller using DSC

The next step is to generate the control voltage v_d and $v_q$using discrete PI controller. Motor voltage and current can be described as the following equations.

$i_d(R + DL) = v_d + \omega L i_q$ (16)

$i_q(R + DL) = v_q - \omega(L i_d + K_e)$ (17)

Generated torque when i_d= 0, $T = \frac{3}{2}p\Psi i_q$ (18)

Where R is the stator resistance, L is the stator inductance, D is the differential operator, ω is the electrical frequency, p is number of pole pair, ψ is flux linkage of permanent magnet and stator, K_e is the back EMF constant and T is the generated torque. All units are in SI units.

The two currents i_q and i_d are the ones to be controlled. From equations, it can be seen i_q and i_d are cross coupled and also dependent on motor rotation speed. If the current loop controller bandwidth is at least an order of magnitude higher than the speed ω, motor speed can be considered as constant and can be ignored for small signal analysis. So, PID controller is designed as if i_q and i_d are independent at first. The discrete PID controller for i_q has the form as described in (19) [4].

$$C(z) = V_q(z)/E_{iq}(z) = (b_0 + b_1 z^{-1} + b_2 z^{-2})/(a_0 + a_1 z^{-1} + a_2 z^{-2}) \qquad (19)$$

Where $V_q(z)$ = Controller Output v_q

$E_{iq}(z)$ = Controller Input, current error

$b_0 = K_p(1 + NT_s) + K_i T_s (1 + NT_s) + K_d N$

$b_1 = -(K_p(2 + NT_s) + K_i T_s + 2K_d N)$

$b_2 = K_p + K_d N$

$a_0 = (1 + NT_s)$

$a_1 = -(2 + NT_s)$

$a_2 = 1$

K_p = Proportional coefficient

K_i = Integral Coefficient

K_d = Differential Coefficient

N = LPF integer coefficient

Above discrete PID controller equation can be arranged as in (20).

$$U(z) = -a_1 z^{-1} U(z) - a_2 z^{-2} U(z) + b_0 E(z) + b_1 z^{-1} E(z) + b_2 z^{-2} E(z) \qquad (20)$$

For actual realization on DSC chip, U(z) in (20) is changed back to time domain difference equation as described in (21).

$$v_q[n] = -\frac{a_1}{a_0} v_q[n-1] - \frac{a_2}{a_0} v_q[n-2] + \frac{b_0}{a_0} e[n] + \frac{b_1}{a_0} e[n-1] + \frac{b_2}{a_0} e[n-2] \qquad (21)$$

PID controller for i_q is shown in figure 3.

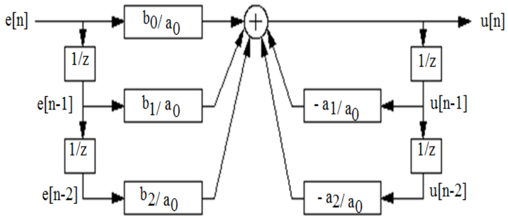

Figure 3. PID controller for i_q

In practice, differential term can be omitted and PI controller is enough. Controller for i_d is the same as that for i_q. To eliminate cross coupling between i_q and i_d, decoupling must be done. The complete control structure is shown in figure 4.

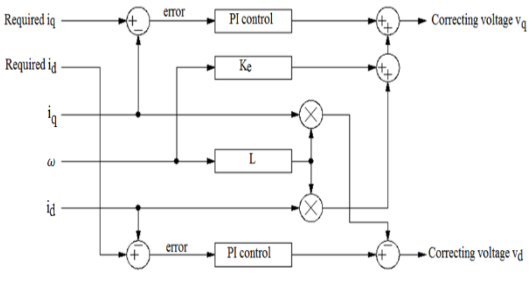

Figure 4. Complete structure of PI control

5. Transformation back to Motor Terminal Voltage

Output of PI controller voltage v_q and v_d must be transformed back to motor terminal voltages v_a, v_b, v_c. This is done by using inverse Park and inverse Clarke transform. Inverse Park transform is described as in (22).

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} v_d \\ v_q \end{bmatrix} \qquad (22)$$

Inverse Clarke transform is described as in (23).

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -1 & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} \qquad (23)$$

After v_a, v_b, v_c are obtained, control of one cycle completes. As the servo motor rotates, motor moves to new position and all of current sensing, transformation to 2 phase system, PI control and transform back to 3 phase system repeat and repeat again.

6. Software Structure

Two most important requirements in motor control software are timing and data integrity. In actual motor driver circuit, motor terminal voltages v_a, v_b, v_c are modulated from DC supply V_{DC} using power electronics devices such as MOSFET or IJBT as shown in figure 5. Switching frequency of this modulation, usually space vector pulse width modulation, is usually 20 kHz or higher.

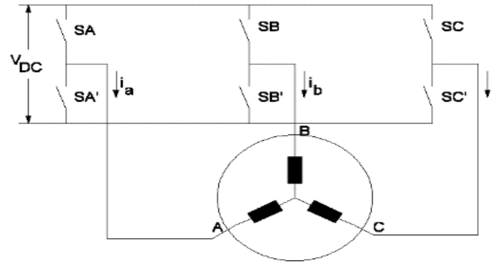

Figure 5. Typical configuration of PMSM

Since all control starts from phase current sensing, these current sensing must be synchronous to this switching i.e. all phase currents must be sampled when the respected power switches are actually turning on. In other words, phase current sampling frequency must be the same as switching speed. All computation of current filtering, transforming to two phase system, PI control, transforming back to three phase system must be finished within one PWM cycle. If the switching frequency is 20 kHz, all calculation must be finished within 50μsec. In actual system, DSC cannot be used only for these current control steps, it must also perform outer velocity and position control loop, and other tasks such as communication to host computer, receiving commands, sending status, checking limit switches, etc. Typical software structure is to use PWM interrupt and time critical controls are computed as interrupt service routine (ISR), and all other tasks are in main software loop. For data integrity, all variables used for specific iteration of the controller must be the same for entire execution cycle of the algorithm. Otherwise, control system will be operating on different sets of data. Typical software structure is shown in figure 6.

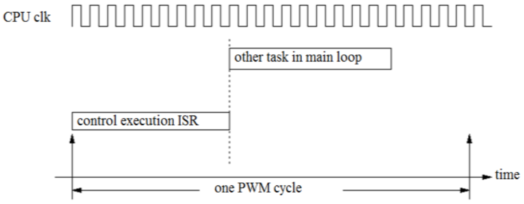

Figure 6.Typical software structure

7. Troubleshooting in Software

If motors behave somewhat unusual, even under normal condition, or do not perform very well, the first one to check is whether all control algorithms really finish within the defined amount of time. After one PWM finished, interrupt is triggered to start control loop. It is good idea to check previous control has actually finished, especially in software developing phase. Second one to check is data representation, especially in fixed-point DSC. In fixed-point DSC, floating points are represented in Qm.n format. The chosen format must cover the whole possible range and must be able to represent as precise as possible.

For example, for a system which has maximum current 10A, Q4.28 is suitable as integer 4 bits covers 16A. For sine and cosine values, Q2.30 is suitable, as integer 2 bit can represent the four quadrants and 30 bits decimal point can represent the sine and cosine value very precisely.

8. Performance Comparison

To test how well DSC performs in motor control application, exactly same motor control loop is implemented on TMS320F28335 DSC and normal general purpose microcontroller STM32F446RE and ATSAME0Q21. To make fair comparison, both DSC and normal controllers are set to run at same frequency of 120MHz. The control loop begins after the motor current measurement and ends after the space vector pulse width modulation. After current measurement, ADC finished interrupt is triggered to start timer on controller. Timer is counting up till control loop is finished. It turns out that timer value on STM32F446RE is 1860, that of ATSAME0Q21 is 1296 and 912 for TMS320F28335. At 120MHz, timer value 1860, 1296 and 912 are equivalents to 15.5µsec, 10.8µsec and 7.6µsec respectively. This shows DSC is $(10.8-7.6)/10.8 = 29.62\% \approx 30\%$ to $(15.5-7.6)/15.5 = 50.96\% \approx 51\%$ faster than normal general purpose controller for exactly same control algorithm.

9. Conclusion

This paper presents applying DSC for PMSM motor control system in which reasonably complicated digital control algorithms including vector control or field oriented control and current regulations are involved. Test result shows DSC is 30% to 51% faster than normal controller. Although these percentages do not represent for all DSCs and control algorithms, it actually reflects how well DSC outperforms general purpose controller in time critical control applications.

Acknowledgements

The author would like to thank her parents and husband for their encouragement. The author wishes to thank her friend, U Htet Phoe (ME.EC, MTU) for providing valuable suggestions for this work. The author also wishes to give her thanks to all teachers who teach her in her whole life.

References

[1] http://en.wikipedia.org/wiki/direct-quadatuer-zero_ transformation.
[2] http://en.m.wikipedia.org/wiki/CORDIC.
[3]http;//kappaiq.com/kappa-tech/articles/pi-controllers.
[4] http://www.scilab.org/discrete-time-pid-controller-implementation.
[5] R. Krishnam, Permanent Magnet Synchronous and Brushless DC Motor Drives. CRC Press, 2010.
[6] Digital Signal Processing in Power Electronics Control Circuits (2nd Edition) Springer.
[7] Advances Research on Computer Science and Information Engineering; International Conference, CSIE 2011; China, May 2011; Proceeding, Part 1.