**Fully Digital Permanent Magnet Synchronous Motor AC Servo System Implementation**

Hadi Hamedi
Young Researchers Club, Saveh Branch, Islamic Azad University, Saveh, Iran

**Abstract:** This paper proposed an advanced technology, which based on DSP and FPGA, applied in the fully digital and high-performance permanent magnet synchronous motor (PMSM) AC servo system. The SVPWM system is simulated and implemented. In the system, the DSP unit implemented the position loop control by combining PI control and feedforward control, while the FPGA unit implemented the speed loop and current loop control. Since FPGA provided a kind of hardware design scheme for the control system, it had the features of a special control IC. The functions of vector control algorithm, velocity and current sampling, space vector pulse width modulation and PI algorithm were included in the special control IC. In the meanwhile, it also provided host communication interfaces that could realize the real-time modification of the control parameters in the IC. Simulation results based on MATLAB 2011 presented. Finally, the experimental results indicate that this system has high-speed response and well static state control performance and have good agreement with simulation results.

**Key words:** PMSM, space vector PWM, MATLAB simulator, field programming gate array (FPGA).

**INTRODUCTION**

Permanent magnetic synchronous motor (PMSM) is widely used in CNC machine, position control, robot, and so on, because of its characters of higher running efficiency, higher torque density, little inertia, little torque ripple and high-speed running (Foo G. and M.F. Rahman, 2010; Kuang-Yao Cheng and Ying-Yu Tzou, 2004; Xianqing Cao, 2009). A high-performance servo system should have fast response and well control performance. Along with the development of the control techniques, excellent control algorithms are gradually introduced to servo system to improve the control performance (Lin, F.J., 2010; Considine, V., 1989). But the single control chip can’t respond the change of the system rapidly for the reasons of the increasing of the complexity of the control algorithm and the multi-functions of the system (Monmasson, E., 2011; Kantabutra, V., 1999). In this paper, putting the series of GK PMSM as the studied object, a high-performance position servo system based on DSP (Considine, V., 1989) and FPGA (Considine, V., 1989) was designed. The position loop control is achieved by DSP, and the speed loop and current loop controls are achieved by using FPGA, which acts as a control integration chip. Aiming at FPGA, the function of motion control chip is designed by adopting Verilog HDL language. The study concentrates on vector control algorithm, velocity and position sampling module, current sampling module, space vector pulse width modulation (SVPWM) module and digital architecture of the PI control algorithm.

2. **Vector Control:**

For PMSM, the typical vector control is the rotor flux-oriented vector control (FOC) algorithm. There are three transforms included in the algorithm. Reference frames for FOC is shown in Fig. 1.

![Reference Frames for FOC](image)

**Fig. 1:** Reference Frames for FOC.

2.1. **Clark Transformation:**

Clark transformation is written as:
\[
\begin{align*}
\begin{cases}
i_{sa} = i_d \\
i_{sb} = (i_d + 2i_B)/\sqrt{3}
\end{cases}
\end{align*}
\]  

(1)

2.2. Park Transformation:

Park transformation is written as:

\[
\begin{align*}
\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} &= \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} i_{sa} \\ i_{sb} \end{bmatrix}
\end{align*}
\]  

(2)

2.3. Inverse Park Transformation:

Inverse Park transformation is written as:

\[
\begin{align*}
\begin{bmatrix} i_{sa} \\ i_{sb} \end{bmatrix} &= \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix}
\end{align*}
\]  

(3)

For PMSM with surface-mounted magnets, there is \( L_d = L_q \). And the electromagnetic torque is expressed as follow:

\[ T_e = 1.5 p_m \psi_r i_{sq} \]  

(4)

Since \( \psi_r \) is a constant, it can be seen from (4) that the electromagnetic torque can be controlled independently by \( i_{sq} \). Then, the vector control strategy is realized.

3. FPGA Implementation:

Speed closed loop control and current closed loop control of the servo system are implemented by FPGA. In the implementation, modularizing design method is used for every function unit.

3.1. Current Closed Loop Control Hardware Implementation:

PI control arithmetic is used in the current closed loop control.

3.1.1. Vector Transformation Module:

Clark, Park and inverse Park transformations are included in FOC vector transformation. Clark transformation can be easily implemented with an adder and a multiplier, but it is difficult for Park and inverse transformation to be implemented because of the rotating transformation. Fortunately, there is an excellent algorithm called CORDIC (Coordinate Rotation Digital Computer) presented by Volder for the transformations. Now, the algorithm is widely used in digital computation of many functions because of its high speed and high resolution.

Basic form of coordinate transformations are followed:

\[
\begin{align*}
x' &= \cos \theta (x - y \tan \theta) \\
y' &= \cos \theta (y + x \tan \theta)
\end{align*}
\]  

(5)

2-radix CORDIC iterative equations are expressed as:

\[
\begin{align*}
x_{i+1} &= x_i - y_i d_i 2^{-i} \\
y_{i+1} &= y_i + x_i d_i 2^{-i} \\
z_{i+1} &= z_i - d_i \arctan(2^{-i})
\end{align*}
\]  

(6)

Where, \( d_i = \text{sgn}(z_i) \).
In practice, multiplication with the $K_i$ factor is omitted during the proceeding of iteration. Thereby, a gain of $1/K_i$ for iteration $i$ is caused. Where, $K_i = \cos(\arctan 2^{-i}) = 1/\sqrt{1+2^{-2i}}$. The total gain is defined as follow:

$$A_n = 1/K_n = \prod_{i=0}^{n-1} \sqrt{1+2^{-2i}} \approx 1.647$$

### 3.1.2. PI regulator:

PI arithmetic is widely used in the many fields of industry control. Improved increment mode PI arithmetic is used to current and speed regulators in this paper.

Discrete increment mode PI arithmetic is described as follow:

$$ \begin{align*}
\Delta u(k) &= k_p(e(k) - e(k-1)) + k_i e(k) \\
u(k) &= u(k-1) + \Delta u(k)
\end{align*} $$

(7)

The PI arithmetic of integral separation is selected in this paper. The implement of the arithmetic is described as follow:

1. A threshold value $\varepsilon$ is set to $\varepsilon > 0$ according to the fact.
2. When $|e(k)| > \varepsilon$, P control is used to avoid too big overshoot and improve system response; otherwise, PI control is used to ensure steady state precision.

The output of speed PI regulator defining the reference of the current closed loop is send to current PI regulator directly. And, the output of current PI regulator, a voltage quantity in the rotation frame, is transferred to SVPWM module after rotated to steady frame.

SVPWM module is also included in current closed loop. The design of SVPWM module is described in reference (Kantabutra, V., 1999) in detail, so it is not mentioned in this paper.

### 3.2. Speed Closed Loop Control:

Speed closed loop has the same control strategy with current closed loop. Speed feedback is acquired from the quadrature signal of the incremental encoder. In order to increase resolution at low and high speed, M/T algorithm is used in the speed feedback computation. Fig. 1 illustrates computation at each speed control loop update period. The algorithm utilizes both pulse count of encoder quadrature signal inputs (edge_count shown in Fig.1) and period measurement between incoming quadrature pulses ($\Delta T$ shown in Fig.2). Speed is calculated as following:

$$n = k \times \text{edge\_count} / \Delta T$$

(8)

Where, $k = 60 f_{clk} / 4PPR$, $f_{clk}$ is the frequency of the reference pulse, $PPR$ is the lines of the incremental encoder.

**Fig. 2:** M/T Algorithm at Low Speed Measurement.
It can be shown from Equation (8) that the speed feedback at low speed can be greatly increased so long as edge count is not zero within a speed control update period.

**Fig. 3:** Complete Control Block Diagram of FPGA.

Besides the above modules, there are other function modules, such as communication module, configuration and status registers, monitor and protection. All the modules constitute a complete FPGA controller, shown in Fig. 3.

All of the above modules are described using Verilog HDL, and are implemented on Spartan of the Xilinx Company. The system clock frequency of the controls can be up to 33.333MHz, and all of them are implemented in hardware mode. As a result, instructions executive period is decreased and system performance is improved.

**4. Position Loop Control:**

An incremental encoder is used as measurement component of position feedback. Position controller is a complex controller including PI regulator, velocity and acceleration feedforward and digital notch filter. Velocity feedforward is to decrease differential gain, and acceleration feedforward is to compensate track error origin brought by inertia, while digital notch filter is to eliminate the effect of resonance. The block of position controller is shown in Fig. 4.
Fig. 5: Prototype of Servo System.

5. System Hardware Design:
This servo system used the FPGA designed in this paper and TMS230LF2407 as its control chips, and IRAMS16UP60A, an intelligent power module (IPM) as its power components. Fig. 5 shows the prototype of the servo system designed in this paper. Fig. 6 shows the architecture of the servo system.

Fig. 6: Architecture of Servo System.

The functions of DSP in this system include: initializing FPGA, the algorithm implement of position loop control, keyboard and display. The functions of FPGA include: implementing of current and speed loop control, and then outputting SVPWM driving signals for the IPM.

The communications between DSP and FPGA is completed through data bus. The steps are that DSP selects FPGA via a decode circuit implemented by GAL, then uses OUT/IN instructions to access inner registers of FPGA to map the address of FPGA inner registers to DSP inner I/O space, and then completes the communications with FPGA.

6. Experimental Results:
The proposed system is simulated by Matlab simulator using field oriented control (FOC) PMSM model (Fig. 7). SVPWM control system is simulated as shown in Fig. 8.

Based on above simulation the responding curve of speed in the process of motor starting when speed reference is 500rpm is shown in Fig. 9. The responding curves of speed and position when position reference is 10000 pauses are respectively shown in Fig. 10 and Fig. 11.

The value of the shunt resister is 20mΩ. PWM is configured symmetrical center aligned mode. So, updating frequency of the current loop is 10kHz and updating frequency of the speed loop is 5kHz.

Experiment results shows that the highest efficiency is 90% at the rated speed of 2000 rpm with no load. The actual sinusoid current waveform of A and B phase, and FFT results are shown in Fig. 12.

Experimental results validate the reliability of the control system and correctness of loaded drive parameters in IRMCK201 and DSP’s program. The FFT results shown that the second harmonic and higher number of harmonics are not appear in this system and we can see just the first harmonic of frequency noise.

The elevator door opening/closing interval can be adjusted to be long enough by changing the delay time in drive parameters file. The reverse and forward running of motor in a sample delay time is shown in Fig. 13.
Fig. 7: Simulation of FOC using PMSM model (Matlab model).

Fig. 8: SVPWM PMSM system simulation in Matlab.

Fig. 9: Speed Curve of the Start Process in Matlab Simulation.
The responding curve of speed in the process of motor starting when speed reference is 500rpm is shown in Fig. 14. The responding curves of speed and position when position reference is 10000 pauses are respectively shown in Fig. 15 and Fig. 16.
Fig. 13: Current waveform of A phase during forward and reverse rotation of motor.

Fig. 14: Speed Curve of the Start Process in experimental results.

Fig. 15: Speed Response of the Closed Loop System in experimental results.

Fig. 16: Position response of the Closed Loop System.

According to Fig. 14 and Fig. 15, it can be seen that speed response is fast and position can track reference signals rapidly without any overshoot.
7. Conclusion:

This paper presents the successful design and implementation of a high performance and full digital AC servo system based on DSP and FPGA. In this system, position closed loop control, in which the PI algorithm and feedforward control are adapted integrative, is implemented by DSP; speed and current closed loops control are implemented by FPGA. Whole system is simulated by Matlab Simulator. The experimental and simulation results show that this system has high-speed response and well dynamic and static control performance. Simulation and experimental results have good agreement with each other as shown in Figures 14 and 9 for Speed Curve of the Start Process, Figures 10 and 15 for Speed Response of the Closed Loop System and in figures 11 and 16 for Position response of the Closed Loop System.

REFERENCES


