Design of Speed and Current Controllers Using Online PSO for IPMSM

ZHENGQIANG SONG[1]; RENPU LI[2]; HUILING YANG[1];DAN XIAO[3;]

1: Yangzhou Polytechnic College, Wenchang West-road, Yangzhou, CHINA;
2: Jiangsu University of Technology, Zhongwu road 1801, Changzhou, CHINA;
3:University of New South Wales, Sydney NSW 2052, AUSTRALIA ;
(This work was supported in part by Jiangsu Natural Science Foundation: BK20161199)

Abstract: - A novel online particle swarm optimization method is proposed to design speed and current controllers of vector controlled interior permanent magnet synchronous motor drives considering stator resistance variation. In the proposed drive system, the space vector modulation technique is employed to generate the switching signals for a two-level voltage-source inverter. The nonlinearity of the inverter is also taken into account due to the dead-time, threshold and voltage drop of the switching devices in order to simulate the system in the practical condition. Speed and PI current controller gains are optimized with PSO online, and the fitness function is changed according to the system dynamic and steady states. The proposed optimization algorithm is compared with conventional PI control method in the condition of step speed change and stator resistance variation, showing that the proposed online optimization method has better robustness and dynamic characteristics compared with conventional PI controller design.

Key-Words: PMSM, Online PSO, Parameters Variation

1 Introduction

Proportional-integral (PI) control technique has been widely used in high performance field orientation controlled IPMSM drives.

There are numerous researches on the applications of computational intelligence techniques to controller parameters design for PMSM [1-3]. Among these, Because of its simplicity and computational efficiency[4-5], PSO has been widely used to solve a broad range of optimization problems, such as adaptive tuning of controller gains and parameters identification. However, there still exist some problems/limitations with this method on the optimization of controller gains. Firstly, the PSO optimization applications in designing controller gains for PMSM rely on offline precise calculations of responses of PMSM using mathematical model [3]. It makes optimization effectiveness rely on the fixed PMSM model excessively. Secondly, the online PSO for PMSM controller optimization, however only applied to the speed controller gains without considering current controller optimization. It is difficult to achieve high dynamics and robustness for PMSM drive system. Thirdly, much effort has been made on real-time PSO application in parameters identification, and results show effectiveness because parameters of PMSM are changed slowly [3]. This should be incorporated with the optimization process online.

In this paper, a discrete simulation model of IPMSM is set up firstly considering dead-time of inverter to present a real-time simulation condition with a step speed reference signal., while the stator resistance is varying. Simulation results show that this online optimization method is model free and has better robustness and faster optimization ability.

2 Mathematical Model of IPMSM Drives

The IPMSM model in the rotor reference frame is

$$\begin{bmatrix} u_{d(t)} \\ u_{q(t)} \end{bmatrix} = \begin{bmatrix} R_s & 0 \\ 0 & R_s \end{bmatrix} \begin{bmatrix} i_{d(t)} \\ i_{q(t)} \end{bmatrix} + \begin{bmatrix} p & -\omega_{re(t)} \\ \omega_{re(t)} & p \end{bmatrix} \begin{bmatrix} \phi_{d(t)} \\ \phi_{q(t)} \end{bmatrix} \quad (1)$$

$$\phi_{d(t)} = L_d i_{d(t)} + \phi_f \quad (2)$$

$$\phi_{q(t)} = L_q i_{q(t)} \quad (3)$$

$$T_{em(t)} = \frac{3}{2} P_n \cdot (\phi_{d(t)} \cdot i_{q(t)} - \phi_{q(t)} \cdot i_{d(t)}) \quad (4)$$

$$T_{em(t)} - T_{L(t)} = \frac{J}{P_n} \frac{d\omega_{re(t)}}{dt} \quad (5)$$

where $u_{d(t)}$, $u_{q(t)}$, $i_{d(t)}$, $i_{q(t)}$, L_d, L_q, $\phi_{d(t)}$ and $\phi_{q(t)}$ are the stator voltages, stator currents, inductances and flux linkages on *dq*-axis respectively; ϕ_f is permanent flux; $p = \frac{d}{dt}$ is the differentiation operator; P_n is the number of pole pair; $\omega_{re(t)}$ is rotor velocity electrical; $T_{em(t)}$ is generated torque; $T_{L(t)}$ is load torque.

The $d-$ and $q-$ axis current controllers are PI whose transfer functions are $k_{id}(1+\tau_{id}s)/s$ and $k_{iq}(1+\tau_{iq}s)/s$, respectively. The reference of $q-$ axis current i_q^{ref} is provided by the speed controller ($k_{s\_p} + k_{s\_i} \cdot \frac{1}{s}$). A low-pass filter is applied to the measured speed and a rate limitation is applied to the speed reference signal. They are used to reduce the overshoot and settling associated with step commands, while retaining a fast disturbance rejection. The online PSO scheme is implemented with a S-Function. The input variables of the S-Function are references and feedback values of q axis current and rotor speed respectively, the output variables are gains of the q axis current and speed PI controllers respectively.

3 Online PSO Parameter Tuning

3.1 Basic principle of the PSO algorithm

Particle swarm optimization algorithm is an evolutionary computation technique developed by Kennedy and Eberhart in 1995. It finds global optimum solution in search space through the interactions of individuals in a swarm of particles. Similar to other evolutionary algorithms, the PSO algorithm firstly produces initial swarm of particles in search space. Each particle represents a candidate solution to the problem and it has its own position *X* and velocity *V*. Each row in the position matrix *X* shows each particle's position, through which we can acquire the evaluation value of the particle. At each iteration, each particle memorizes and follows the tracks of its personal best (*Pbest*) and the global best position (*Gbest*) vectors to update the velocity matrix *V*.
Known these two best positions, particles can change velocities and positions using the following rules:

$$v_{j,g}^{(t+1)} = w \cdot v_{j,g}^{(t)} + c_1 \cdot r_1 \cdot (pbest_{j,g}^{(t)} - x_{j,g}^{(t)}) + c_2 \cdot r_2 \cdot (gbest_g^{(t)} - x_{j,g}^{(t)}) \quad (6)$$

$$X^{(t+1)} = X^{(t)} + V^{(t)} \quad (7)$$

where *j*=1, 2, … *m*; *g*=1, 2,… *n*. The superscripts *t* and *t+1* denote the time index of the current and the next iterations respectively. The parameters c_1 and c_2 are called acceleration constant which adjust the maximum step of the particle flight towards *Pbest* and *Gbest* position.

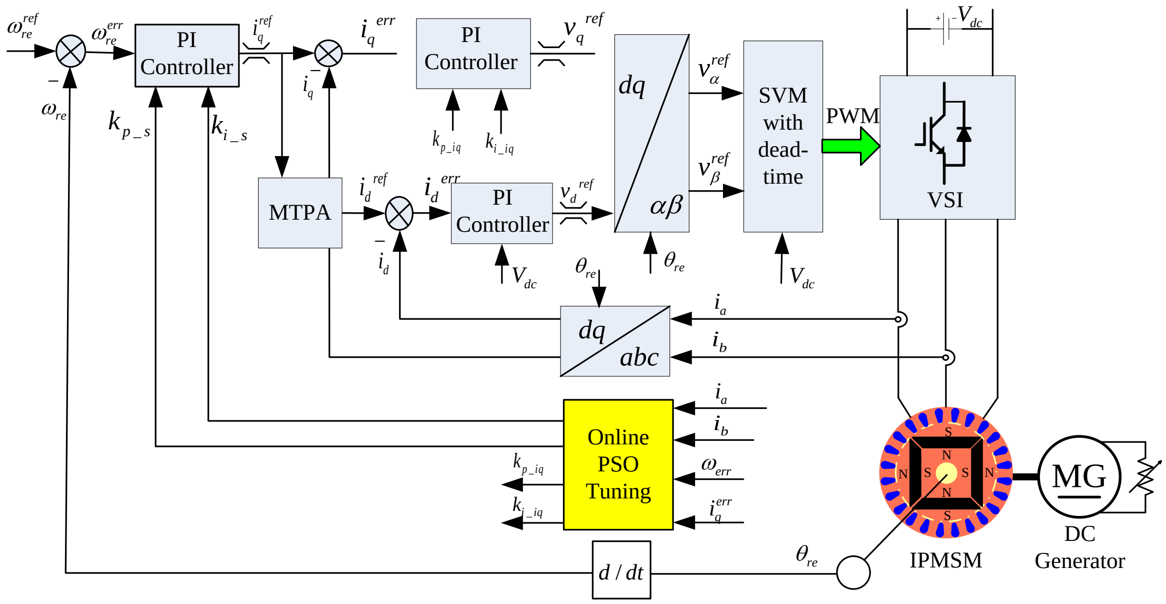


Fig.1. Block diagram of IPMSM drives based on PSO controller

Usually, parameters c_1 and c_2 are equal to 2, r_1 *and* r_2 are uniformly distributed random numbers in the interval (0, 1). Parameter w is inertia weight factor that usually decreases linearly from 0.9 to 0.4 in according to (8) over the course of the run. In this way, the algorithm can easily escape from local optimal solution in the early iteration stage as well as speed the convergence in the later iteration stage, and increase the reliability of finding the global optimal solution.

$$w^{(t)} = w_{\max} - t \cdot (w_{\max} - w_{\min})/iter_{\max} \tag{8}$$

where $w_{\max}$ and $w_{\min}$ are the maximum and minimum values of w , and $iter_{\max}$ is the maximum iteration times.In order to reduce the likelihood of particles leaving the search space, the value of each dimension of the velocity $v_{j,g}^{(t)}$ is clamped to the interval $[-v_g^{\max}, v_g^{\max}]$. The value of $v_g^{\max}$ is usually chosen to be

$$v_g^{\max} = k \cdot x_g^{\max}, \ 0.1 \le k \le 0.5 \tag{9}$$

where $x_g^{\max}$ is the upper bound of search region in the g-th dimension.

3.2 Implementation of online PSO

During the optimization process, to evaluate a candidate solution, such as PI gains, they were kept constant under the whole system simulation, at the same time output errors such as speed errors were added up to evaluate the candidate solution. Usually, PSO algorithm requires a number of iterations to obtain a satisfactory solution. For iteration, the system model has to be simulated once. Then, the model needs to be simulated a number of times to search the best solution. In fact, we can not make IPMSM drive system repeat starting continually. That means such simulation results can not be acquired in real system. However, it is difficult for each particle of candidate PI parameters to be evaluated within one sampling time, which is usually from 10-100 μsec. So, in this paper, we adopted a new method for online PSO update calculations of particle positions *X*, velocity *V*, *Pbest* and *Gbest* of the PSO algorithm. We measured speed and current as sampling values, while updating present particle information instead of updating whole swarm information. For example, there is a particle swarm with 30 particles for optimization. That means if the sampling of IPMSM drive system is 10μsec, each particle’s information such as position and velocity updates once within 100 μs, however, *Pbest* and *Gbest* would be updated completely within 30*10 μsec.

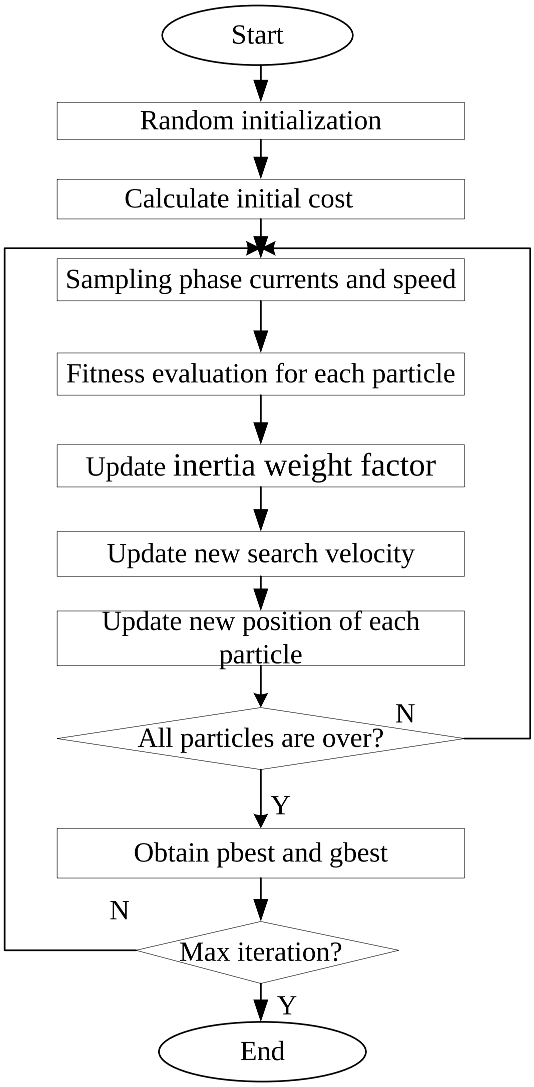


Fig.2 Online PSO flow frame within one sampling period

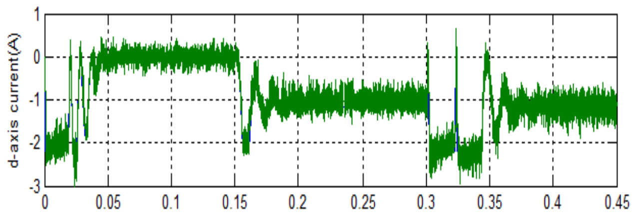
4. Simulation Results

The block diagram of the drive system used for the simulation is shown in Fig.1 The flow chart of online PSO of controllers is shown in Fig. 2. The simulation model includes the encoder, sensor and control input nonlinearities. The controllers are in discrete time with the Euler method, to make the simulation to mimic the real experiment closely. The controller sampling time is chosen as 100 μsec, which is the same as the controller sampling time of real experiment.

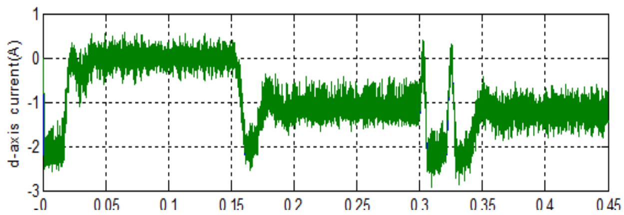
IPMSM parameters are presented in TABLE I, and resistance can be changed from 5.8 to 7.8 for testing the robustness of proposed method. Initial swarm size of particles is 30, and each particle has four variables ($k_{p\_iq}, k_{i\_iq}, k_{p\_s}, k_{i\_s}$) representing its position vectors in search space. After 30 iterations, the results of *Gbest* would be updated as optimized outputs. In order to test the proposed method in this paper, two different controll strategies IPMSM drive, and comparison results between conventional PI controllers and online PSO-PI controllers for IPMSM drives are presented in Fig.4 and Fig.5. In this IPMSM control system, a MTPA method is adopted, and among 20ms to 50ms of the transient state, motor begins speed up and reach 1000rpm; when time is about 150ms, extra load will be added to the system, and which is in steady state again at about 200ms; when time is 300ms, motor will rotate in inverse direction suddenly and reaches -1000rpm at about 350 ms. As is shown in Fig.3 where(a) is i_d current response curve with fixed PI parameters and (b) is i_d current response curve with Online-PSO PI parameters, obviously, when time is about 20ms and speed is changed. Suddenly, four times spike peak appear in i_d of (a) but not in i_d of (b); at the same time here , these esults clearly show the effectiveness of the PSO method proposed in this paper.

TABLE I: PARAMETERS of IPMSM

Phase resistance, R_s	5.8-7.8 Ω
d-axis inductance L_d	0.0448H
q-axis inductance L_q	0.1027H
Rated voltage	230V
Moment of inertia	0.001329 $kg \cdot m^2$
Rated speed	4000 rpm
Permanent magnet flux	0.533 wb
Pole pair	2

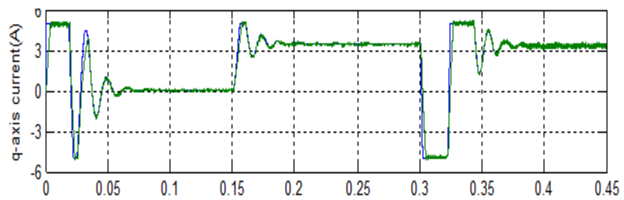


(a) d-axis current with Fixed-PID

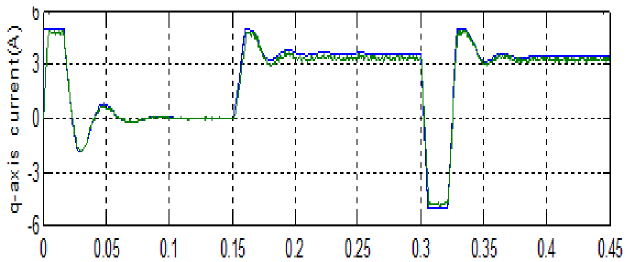


(b) d-axis current with On-line PSO-PID

Fig.3 Comparative results of d-axes current between conventional PI and On-line PSO-PI method

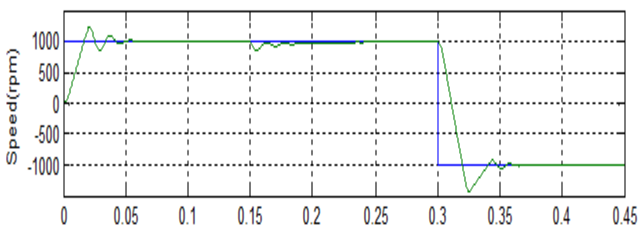


(a) q-axis current with Fixed-PID



(b) d-axis current with On-line PSO-PID

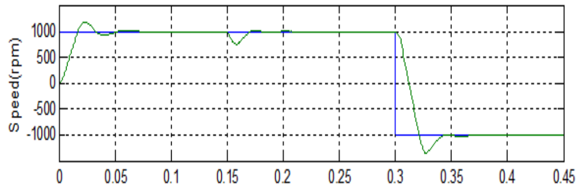
Fig.4 Comparative results of q-axes current between conventional PI and On-line PSO-PI method



(a) Rotor speed curve with Fixed-PID

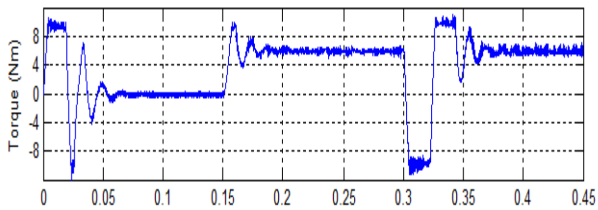
(b)

(c)

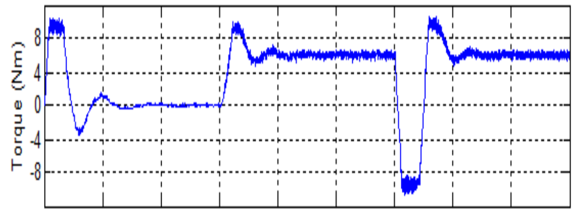


(b) Rotor speed curve with On-line PSO-PID

Fig.5 Comparative results of rotor speed between conventional PI and On-line PSO-PI method



(a) Rotor torque curve with Fixed-PID



(b) Rotor torque curve with On-line PSO-PID

Fig.6 Comparative results of rotor torque between conventional PI and On-line PSO-PI method

5. Experiment Results

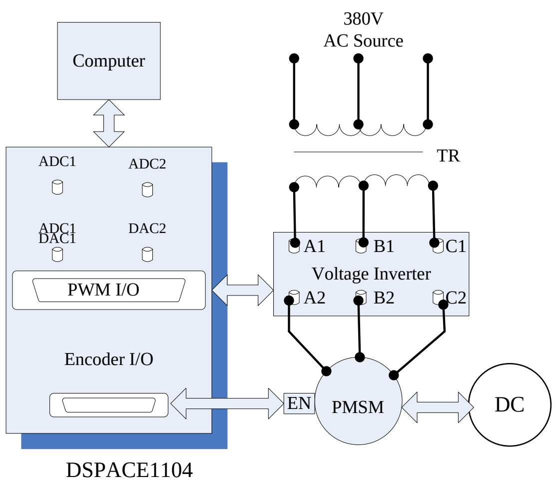


Fig 7. Speed Control Development System of PMSM

In order to test the excellent functions of On-line PSO raised in this paper, PMSM Development System based on DSPACE 1104 Controller equipments with CPU TI-F240 is set up.

As shown in Figure 1, in the permanent magnet synchronous motor AC speed control system development, the main circuit is composed of a three-phase transformer with 0-380 V power supply; and the three-phase voltage inverter can be converted into frequency in 0-1000 Hz and 0-415 V in the voltage change of the power supply, the main switch device using IGBT, control signal by DSPACE PWM I/O; variable frequency power supply by A2, B2, C2 connected to the salient pole permanent magnet synchronous motor IPMSM motor, with parameters of table I, EN encoding for rotor position detection of rotor, each turn output 4000 Hz high level pulse, and connected to the DSPACE encoding plate interface; DC motor load controller used as a test; using DSP as the core technology of DSPACE1104, and a desktop computer connected to complete the visual software programming and IPMSM closed-loop control; phase current, after testing through the coaxial cable is connected to the DSPACE ADC1 and ADC2 port, to complete conversion from analog signal to digital signal. Detection tools need to be prepared: DC constant voltage constant current source, multi-meter, oscilloscope, tachometer and wire number.

51 Initial installation angle between rotor and incremental encoder

Unlike DC motor and asynchronous motor, before the start we need to know the accurate rotor initial position of permanent magnet synchronous motor (PMSM) , to complete the coordinate transformation realization of vector control strategies. We usually use the incremental encoder to obtain the rotor position, and incremental encoder generally provide 3 group of pulses, A, B, and Z. the A and B combined pulse rotor speed and positive & negative information can be obtained, and also when the rotor to rotate a circle with A\ Z pulse. After the encoder is actually installed to the rotor shaft, there is a certain angle between the initial angle of the rotor and the mechanical position of the Z pulse, which is called the initial installation angle. Only when the initial installation angle is accurate, can the position of the rotor be acquired in the condition of standstill or rotation.

Through the DC to the DC motor rotation, and drag the PMSM, through the oscilloscope or DSPACE 1104 control window to observe the counter electromotive force waveform and Z pulse waveform. As shown in Figure 2, adjust the two waveform, so that the Z pulse down along with the anti electromotive force waveform zero crossing alignment, can be observed in the Z pulse ahead of the rotor N pole (the highest point of the back EMF) point of view. Due to the PMSM of the rotor pole number is 2, so the angle between the two Z pulses is 360*2 electric angle.

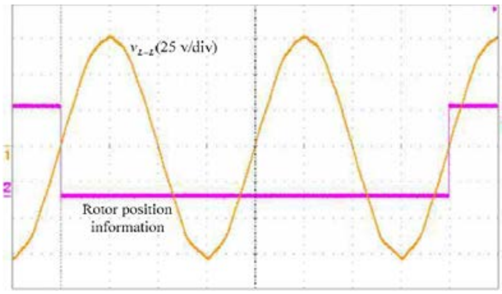


Fig.8 Z Pulse of rotor position and Back-EMF curve

5.2 Detection and elimination of the error of phase current detection

Because the current sensor has a certain nonlinear, the output voltage of the inverter is zero, the current signal is not zero, and sometimes even to achieve the 0.5A error, as shown in figure 3. And the initial error of the current is too large, which has great influence on the flatness and smoothness of the PMSM three-phase current, which leads to the precision control of the performance and even the failure. And different current sensor board, the detection error is not the same size, so developers usually use the software method to eliminate the initial error of current detection. Since the load is generally three-phase symmetrical load, so we can measure currents i_a , i_b, and get the i_c , $i_c = -i_a - i_b$. In Figure 3, i_c is about to 0.45 amperes or so, we can subtract that value in the software in advance, for the same, you can make the initial value of three-phase current is close to zero, error is generally in the range of 0.05A, the system control performance had little effect.

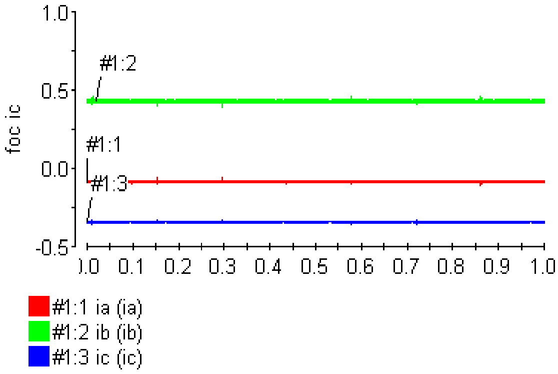


Fig. 9 initial detection error of three phase current

5.3 Phase current and voltage scaling factor

Since the voltage and current sensor signal is generally small voltage signals, and in the PMSM control software programming, in order to make the program voltage and current variables value and the actual measured values are equal, we must to measure voltage, current sensor of the linear range and calculate the corresponding proportion coefficient. For example, table 2 shows, using a wire along the current sensor in the direction of the arrow, the DC current source 1A, 3A, 5A were input to a fixed resistors, and get the value of the corresponding current sensor, which can calculate the current sensor in the actual circuit in the proportion coefficient is 10, pay attention to the positive and negative are key here. And the use of a universal table for simple voltage measurement can get the voltage scaling factor.

5.4 Test for inverter operating frequency and voltage

Voltage inverter in high precision speed control of PMSM control system plays an important role. Therefore, it is necessary to tests were carried out under different conditions on the inverter, especially of high frequency and high voltage stage, current waveform can maintain good sine degree directly affects the weak magnetic speed control strategy. In order to prevent the high frequency, the inverter main switch device in the course of a short circuit, the inverter protection, control system can not work properly.

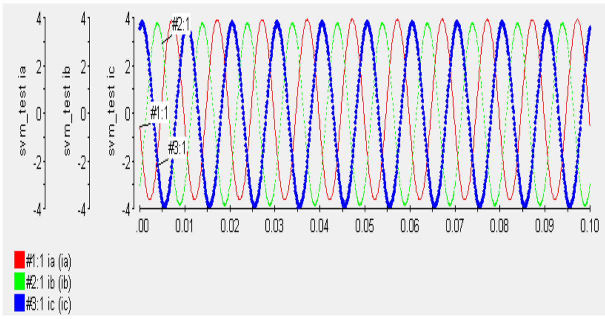


Fig.10 Load current curves when input voltage is DC 100V and output frequency is 100Hz

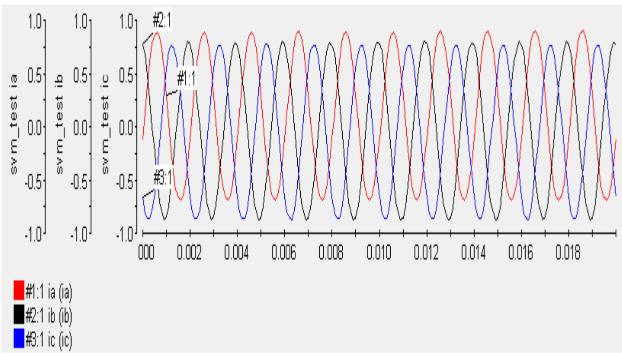
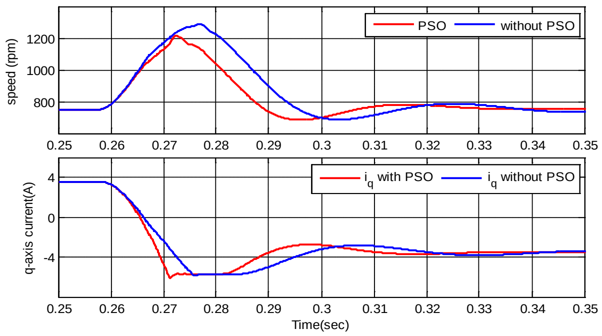


Fig.11 Load current curves when input voltage is DC 100Vand output frequency is 500Hz

Therefore, the dead time must be set according to the performance of the main switch device, which is generally 2-4us. Figure 4 shows the voltage inverter in the input DC voltage is 100V, AC output voltage and frequency is 100Hz load current waveform curve; Figure 5 for the inverter output frequency is 500Hz load current waveform. Compared to the two can be seen, the inverter in frequency 500Hz began to appear current waveform distortion.



5 Conclusion

In this paper, a novel online particle swarm optimization method is proposed to design speed and current PI controllers of vector controlled interior permanent magnet synchronous motor drives, taking into account stator resistance variation. The speed and current controllers are optimized online as PSO update is carried out in each sampling cycle for the position and velocity of particle which represent the performance of the controllers. Furthermore, optimization process does not rely on the preciseness of IPMSM model excessively, because when stator resistance changes within 15%, the dynamic and steady-state performance of the drive system has been improved by optimization of speed and current controller at the same time.

References:

[1] Shin-Hung Chang, Shin-Hung Chang, “Self-tuning gains of PI controllers for current control i n PMSM,” *IEEE 5th Conference on Industrial Electronics and Applications*, pp.1282-1286, 2010.

[2] Muhammed Fazlur Rahman, Md. Enamul Haque, “Problems associated with the direct torque control of an Interior permanent-magnet synchronous motor drive and their remedies,” *IEEE Trans. on industrial electronics*, vol.51, no. 4, pp. 799-808, 2004.

[3] Wenxin Liu, Li Liu, Il-Yop Chung, David A. Cartes, “Real-time particle swarm optimization based parameter identification applied to permanent magnet synchronous machine,” *Elsevier*, *Applied Soft Computing*, pp. 2556–2564, 2011.

[4] Kennedy J, Eberhart RC. ‘‘Particle swarm optimization,’’ in *Proceedings of International Conference on Neural Network (ICNN)*, Vol. 4, pp. 1942–1948 Perth, Australia 1995.

[5] Shuai Yan, Dianguo Xu, Xianguo Gui, Ming Yang and Bingbing Li, “Online Particle Swarm Optimization Design of Speed Controller considering Anti-windup for PMSM Drive System”, *IEEE Annual Power Electronics Specialists Conference*, pp.2273-2277, 2007.