ISSN No:-2456 –2165

SVPWM based Closed Loop Speed Control of Induction Motor with PSO and SMC

Beryl.J[1], Rajan.V.R[2], Dr.K.Selvi[3]
Department of EEE[1,2,3]
V V College of Engineering [1,2],
Thiagarajar College of Engineering[3]
India
Jkumarberyl234@Gmail.Com

Abstract:-**The sliding mode based particle swarm optimization control scheme is designed for the closed loop control of induction motor drive. The MATLAB simulation is formed with particle swarm optimization (PSO) and sliding mode control (SMC) using PWM techniques. In this study PSO is considered as major controller and stability can be indirectly cinched by the concept of SMC without strict captivity and precise system knowledge. Space vector pulse width modulation are used for the speed control of IM and their performance is analysed using MATLAB. The main objective of using PWM technique is to control the speed, current, and torque. The space vector pulse width modulation technique provides better utilization of DC and harmonics got reduced in this perusal. The entire intend is tested in MATLAB. Numerical simulations and experimental results are taken to verify the effectiveness of the proposed control system. Total harmonic distortion (THD) is also observed.**

Keywords:- Induction Motor, Particle Swarm Optimization, Sliding Mode Control, Space Vector Pulse Width Modulation.

I. INTRODUCTION

Almost 70% of the machines used in industries are 3 phase Induction motors (IM). It works on the principle of induction. When rotating magnetic field of the stator cuts the stationary rotor conductors, electro-magnetic field (emf) is induced in to the rotor conductors. As the ac power is used by industries for generation, transmission and distribution, induction motors are occupied noteworthy place in industrial drive applications. Squirrel cage induction motors are widely used in motor and drive application. Squirrel cage Induction motors have soft controllability and snaggy constructions. So induction motors are now playing obligation role in industries for high speed operation. Also induction motors are self starting and cheaper in cost due to the absence of brushes, commentators, and slip rings. Particle swarm optimization (PSO) technique is rapidly used for solving optimization problems. Particle swarm optimization is Swarm intelligence based non-deterministic optimization technique. Stability is achieved by using sliding mode control (SMC) and then incorporated into particle swarm optimization[1].

The main aim of this study is to design a sliding mode control based particle swarm optimization controller (TSPSOC) to verify the dynamic performance parameters of Induction motors.

II. SVPWM

Most commonly Induction motors are controlled by PWM based drives. As compared with fixed frequency drives, these PWM drives controls the both magnitude of voltage and frequency of the current as well as voltage applied to the induction motor. The amount of power delivered by these drives is also varied by changing the PWM signals applied to the power switch gates so that the three phase induction motor speed control is achieved.

A number of Pulse width modulation (PWM) techniques are used for controlling three phase motor drives. But most widely Sine PWM (SPWM) and space vector PWM (SVPWM) are used for controlling. The space vector PWM is an advanced computation technique that has been playing a pivotal and viable role in power conversion.

The main objective of space vector PWM technique is to approximate the reference voltage vector V_{ref} using the eight switching patterns. It also minimizes the THD as well as the switching losses. Here the direct controlled variable is motor voltage and motor frequency. By controlling the amplitude and frequency, the motor voltage and frequency can be controlled. The output voltages are produced either by $V_{dc/2}$ or $-V_{dc/2}$. When switches S_1, S_6 and S_2 are closed, the corresponding output voltage are $V_{ao}=V_{dc/2}$, $V_{bo}= -V_{dc/2}$ and $V_{co}= -V_{dc/2}$. This state is denoted as (1, 0, 0) so that the space vector [2]

$$V(t) = \frac{2}{3}[V_{dc}e^{jo}] \quad (1)$$

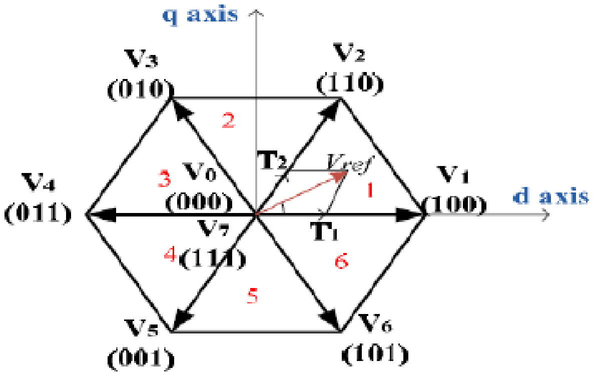

Fig. 1.Trajectory of Reference Vector and Phase Voltage Vector.

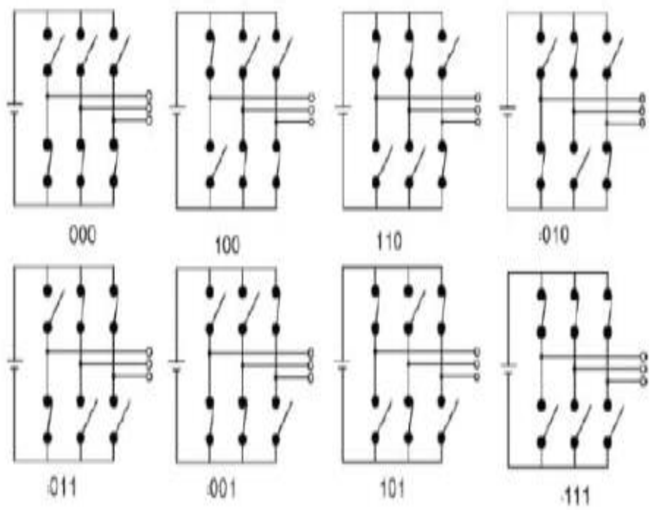

Fig. 2.Eight Switching Configuration of Three Phase Inverter[2]

The remaining active and non active state can be insistent by repetition of the same procedure, shown in Table I. The eight inverter states can be transformed into eight related space vectors. The space vector and related switching states relationships are shown in Table II. The reference vector ,V_{ref} rotates in space at an angular velocity ω=2πf ,where f - denotes the fundamental frequency of the inverter voltage. The diverse set of switches which are mentioned in table I will be turned ON or OFF, when the V_{ref} vector passes through each sector. The frequency of the inverter coincides with the rotating speed of the V_{ref}. Zero vectors (V_0 and V_7) and active vectors (V_1-V_6) do not move in space [3] [4]. The derived output line to line and phase voltage in terms of the DC supply are,

$$\begin{bmatrix} V_{ab} \\ V_{bc} \\ V_{ca} \end{bmatrix} = V_{dc} \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} V_{an} \\ V_{bn} \\ V_{cn} \end{bmatrix} = \frac{V_{dc}}{3} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix} \quad (3)$$

The output line to neutral voltage and the output line to line voltage in terms of DC link V_{dc} are given in TableII The reference voltage V_{ref} and angle α of corresponding sector can be determined as

$$V_{ref} = \sqrt{V_d^2 + V_q^2} \quad (4)$$

$$\propto = \tan^{-1}\left(\frac{V_q}{V_d}\right) \quad (5)$$

Duration of switching time at any sector is given by,

$$T_1 = \frac{\sqrt{3}T_z|V_{ref}|}{V_{dc}}\left(\sin\left(\frac{N\pi}{3}\right)\cos\alpha - \cos\left(\frac{N\pi}{3}\right)\sin\propto\right) \quad (6)$$

$$T_2 = \frac{\sqrt{3}T_z|V_{ref}|}{V_{dc}}\left(-\cos\propto\sin\left(\frac{(N-1)\pi}{3}\right)\sin\propto\cos\left(\frac{(N-1)\pi}{3}\right)\right) \quad (7)$$

$$T_0 = T_z{\_}T_1 - T_2 \quad (8)$$

Space vector	**Switching stage**	**On state switch**	**Vector definition**
$\vec{V}_0$	[000]	S_4,S_6,S_2	$\vec{V}_0 = 0$
$\vec{V}_1$	[100]	S_1,S_6,S_2	$\vec{V}_0 = \frac{2}{3}V_{dc}e^{j0}$
$\vec{V}_2$	[110]	S_1,S_3,S_2	$\vec{V}_0 = \frac{2}{3}V_{dc}e^{j\frac{\pi}{3}}$
$\vec{V}_3$	[010]	S_4,S_3,S_2	$\vec{V}_0 = \frac{2}{3}V_{dc}e^{j\frac{2\pi}{3}}$
$\vec{V}_4$	[011]	S_4,S_3,S_5	$\vec{V}_0 = \frac{2}{3}V_{dc}e^{j\frac{3\pi}{3}}$
$\vec{V}_5$	[001]	S_4,S_6,S_5	$\vec{V}_0 = \frac{2}{3}V_{dc}e^{j\frac{4\pi}{3}}$
$\vec{V}_6$	[101]	S_1,S_6,S_5	$\vec{V}_0 = \frac{2}{3}V_{dc}e^{j\frac{5\pi}{3}}$
$\vec{V}_7$	[111]	S_1,S_3,S_5	$\vec{V}_0 = \frac{2}{3}V_{dc}e^{j0}$

Table 1: Space Vectors, Switching Stage, on State Switch

ISSN No:-2456 –2165

Voltage vector	Switching vector			Line to Neutral voltages			Line-line voltages		
	a	***b***	***c***	V_{an}	V_{bn}	V_{cn}	V_{ab}	V_{bc}	V_{ca}
$\vec{V}_0$	0	0	0	0	0	0	0	0	0
$\vec{V}_1$	1	0	0	$2\frac{V_{dc}}{3}$	$-\frac{V_{dc}}{3}$	$-\frac{V_{dc}}{3}$	V_{dc}	0	$-V_{dc}$
$\vec{V}_2$	1	1	0	$\frac{V_{dc}}{3}$	$\frac{V_{dc}}{3}$	$-2\frac{V_{dc}}{3}$	0	V_{dc}	$-V_{dc}$
$\vec{V}_3$	0	1	0	$-\frac{V_{dc}}{3}$	$2\frac{V_{dc}}{3}$	$-\frac{V_{dc}}{3}$	$-V_{dc}$	V_{dc}	0
$\vec{V}_4$	0	1	1	$-2\frac{V_{dc}}{3}$	$\frac{V_{dc}}{3}$	$\frac{V_{dc}}{3}$	$-V_{dc}$	0	V_{dc}
$\vec{V}_5$	0	0	1	$-\frac{V_{dc}}{3}$	$-\frac{V_{dc}}{3}$	$2\frac{V_{dc}}{3}$	0	$-V_{dc}$	V_{dc}
$\vec{V}_6$	1	0	1	$\frac{V_{dc}}{3}$	$-2\frac{V_{dc}}{3}$	$\frac{V_{dc}}{3}$	V_{dc}	$-V_{dc}$	0
$\vec{V}_7$	1	1	1	0	0	0	0	0	0

Table 2: Voltage Vectors, Switching Vector, Phase Voltage, Line-Line Voltage As A Function of DC Bus Voltage

Sector	Upper switches	Lower switches
1	$S_1=T_1+T_2+T_0/2$ $S_3=T_2+T_0/2$ $S_5=T_0/2$	$S_4=T_0/2$ $S_6=T_1+T_0/2$ $S_2=T_1+T_2+T_0/2$
2	$S_1=T_1+T_0/2$ $S_3=T_1+T_2+T_0/2$ $S_5=T_0/2$	$S_4=T_2+T_0/2$ $S_6=T_0/2$ $S_2=T_1+T_2+T_0/2$
3	$S_1=T_0/2$ $S_3=T_1+T_2+T_0/2$ $S_5=T_2+T_0/2$	$S_4=T_1+T_2+T_0/2$ $S_6=T_0/2$ $S_2=T_1+T_0/2$
4	$S_1=T_0/2$ $S_3=T_1+T_0/2$ $S_5=T_1+T_2+T_0/2$	$S_4=T_1+T_2+T_0/2$ $S_6=T_2+T_0/2$ $S_2=T_1+T_0/2$
5	$S_1=T_2+T_0/2$ $S_3=T_0/2$ $S_5=T_1+T_2+T_0/2$	$S_4=T_1+T_0/2$ $S_6=T_1+T_2+T_0/2$ $S_2=T_0/2$
6	$S_1=T_1+T_2+T_0/2$ $S_3=T_0/2$ $S_5=T_1+T_0/2$	$S_4=T_0/2$ $S_6=T_1+T_2+T_0/2$ $S_2=T_2+T_0/2$

Table 3: Switching Time Calculation At Each Sector.

III. SLIDING MODE CONTROL (SMC)

Sliding-mode control (SMC) is one of the competent nonlinear control approaches which provide system dynamics that are controlled in the sliding mode. The first step of SMC design is to select a sliding surface that models the desired closed-loop control performance in state variable space as shown in [1]. Then design control is made such that the system state trajectories are forced close to the sliding surface and stay on it [5]. The system trajectory which slides along the sliding surface to the origin is said to be sliding mode [6]. Once the sliding mode occurs, the system is insensitive and is helpful for defining the corresponding fitness function and particle velocity to become a sliding mode based PSO algorithm. Inertial weight, w is introduced in this control system to decide the applicable evolutionary length for accelerating the searching speed. Define a tracking error, e=d*-d, in which d* represents a specific position reference trajectory. The control nominal system dynamic is given by

$$\ddot{e} + k_1\dot{e} + k_2 e$$

where k1 and k2 are positive constants. Properly choosing the values of k1 and k2, the desired system dynamic such as rise time, overshoot, and settling time can be easily designed. The baseline model dynamic can be rewritten in the state variable form as

$$\frac{d}{dt}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k_2 & -k_1 \end{bmatrix}\begin{bmatrix} e \\ \dot{e} \end{bmatrix}$$

Or $\dot{\boldsymbol{e}}$ = where e $=\begin{bmatrix} e \\ \dot{e} \end{bmatrix}$ and A$=\begin{bmatrix} 0 & 1 \\ -k_2 & -k_1 \end{bmatrix}$

Now consider the sliding surface as

$$\mathrm{S}(t) = C(\boldsymbol{e}) - C(\boldsymbol{e}_0) - \int_0^t \frac{\partial C}{\partial \boldsymbol{e}^T}\mathrm{A}\boldsymbol{e}d\tau$$

Where $C(\boldsymbol{e})$ is a scalar variable and $\boldsymbol{e}_0$ is the initial state.To maintain the state on the surface S(t)=0 for all time, one only needs to show that

$$S(t)\dot{S}(t) < 0,\text{if } S(t) \neq 0$$

When boundary layer is introduced around the sliding surface, the integral terms were used to eliminate steady-state error that results from continuous approximation of switching control. The special integral term is designed to eliminate the reaching phase and not to reduce steady-state error due to continuous control. This feature makes it easy to select the system performance based on overshoot, rise time and settling time specifications. Selection of proper boundary values makes the system dynamics to be stable. The Significant effect on the control performance is the selection of the upper bound. If the bound is selected too large, the controller will result in serious chattering phenomena which might excite unstable system dynamics.

IV. PSO (PARTICLE SWARM OPTIMIZATION)

Particle swarm optimization (PSO) is an progressive computation technique developed by Eberhart and Kennedy in 1995.It was inspired by social behaviour of bird flocking which has recently received much interest for achieving high efficiency and searching global optimal solution in problem spaces [7-17]. It is used for searching optima by updating the generations. The role of PSO is used as a minor compensatory tuner for controlling the speed and the stability of PSO control scheme cannot be guaranteed. The procedure for implementing the global version of PSO is given by the following steps [18][19][20].

- Initialize a population (array) of particles with random positions and velocities in the *n*-dimensional problem space using a uniform probability distribution function.
- Evaluate the fitness value of each particle. According to the spirit of SMC, the ultimate evolutionary target of PSO can be regarded as S(t)=0 and S'(t)=0. Therefore, a fitness function is defined as

$$FIT(S) = \exp\left[-\eta \ X\left(S^2 + \dot{S}^2\right)\right] \epsilon [0,1] \quad (1)$$

Where exp[.] is the exponential function, η is a positive constant, and S is a sliding surface.

- Compare each particle's fitness with the particle's l_{best} . If the current value is better than the l_{best}, then set the l_{best} value equal to the current value and the l_{best} location is equal to the current location. If the fitness value of present particle position is lower than the fitness values of local best particle position, the local best particle position should be modified by

$$l_{best}(n+1) = l_{best}(n) + \delta \quad (2)$$

where n indicates the current state, n+1 means the next state, and δ is designed as a small positive constant to prevent the local optimum problem.

4) Compare the fitness with the population's overall previous best. If the current value is better than g_{best} , then reset g_{best} to the current particle's array index and value.

5) Change the velocity and position of the particle according to [21] respectively. The sliding surface and its derivative are introduced into the particle velocity operation. The control effort is made to stay in sliding surface, will be changed fewer or even unchanged. According to this principle, the new particle velocity can be generated by

$$p_v(n+1) = p_v(n) \text{ x w} + c_1 \text{ x rand(.) x } [l_{best}(n) - u(n)] \text{ x } \dot{S} + c_2 \text{ x rand(.)x} [g_{best}(n) - u(n)] \text{ x S} \quad (3)$$

where p_v represents the particle velocity, rand(.) is a random number between 0 and 1, c1 and c2 are the acceleration constants with positive values, w is an inertial weight and is defined as[8]

$$w = a + [b \times \exp(S^2)]/[1 + \exp(S^2)] \quad (4)$$

where a and b are positive constants. The parameter w can be adjusted to reduce w as the system dynamic approaches the sliding surface. Due to particle's new velocity, the new particle position [22] will be calculated at the next time step according to the following equation:

$$u(n+1) = u(n) + p_v(n+1) \quad (5)$$

However, the new particle position will be set as u_{min} or u_{max} if the new particle position flies beyond the boundary [u_{min}, u_{max}].

A. Procedure Involved in PSO

In order to state the procedures involved in the design of a PSO-based control scheme, the flowchart of the proposed TSPSOC system is depicted. For easy to understand the TSPSOC operation process, it supposes that four particle positions (u1, u2, u3, u4) and the corresponding particle velocities (pv1, pv2, pv3, pv4) in the initial population (N=4) are randomly chosen from the reasonable region [u_{min}, u_{max}].

Process 1

In steps 1 and step 2, the corresponding fitness values (FIT1, FIT2, FIT3, FIT4) are evaluated via immediate tracking responses in Step 3.

Process 2

The first fitness value is regarded as the global best particle position by choosing the particle positions with the first and second fitness values from these four particles, and the second fitness value is utilized to be the local best particle position in Step 4.

Process 3

u3 is chosen as the global best particle position (g_{best} (0)),and u4 is selected to be the local best particle position (l_{best}(0)).

Process 4

The particle velocity and position is updated via Eqs. (3) and (5) in Step 10.

Process 5

If the new particle position flies beyond the boundary [u_{min}, u_{max}] as discussed in step 11, the new particle position will be set as u_{min} or u_{max} (Step 12), else it directly outputs the results of Eq. (5) as the new particle position. Then the new particle position is regarded as the new control effort to export as shown in Step 13.

Process 6

After the first generation this new control effort will be treated as the current particle position of the next generation because it is the most suitable particle for the present circumstance.

Process 7

If the fitness value of current particle position lies in between the fitness values of the global and local particle positions (Step 5), the current particle position will become the new local best particle position.

ISSN No:-2456 –2165

Process 8

If the fitness value of current particle position is higher than the global best fitness value, the current particle position will become the new global best particle position and the old global best particle position will replace the local best particle position (Step 8).

Process 9

If the fitness value of current particle position is lower than the local best fitness value (Step 7), the local best particle position should be modified by Eq. (2) (Step 9).

* ***Flowchart***

The operation process will repeat continuously until this program completes, so that adaptive control effort will be produced persistently .Consequently the objective of PSO control can be achieved and the stability of the proposed TSPSOC strategy can be indirectly ensured by means of the spirit of SMC. The process involved by PSO for finding the optimum solution is shown by using the flowchart

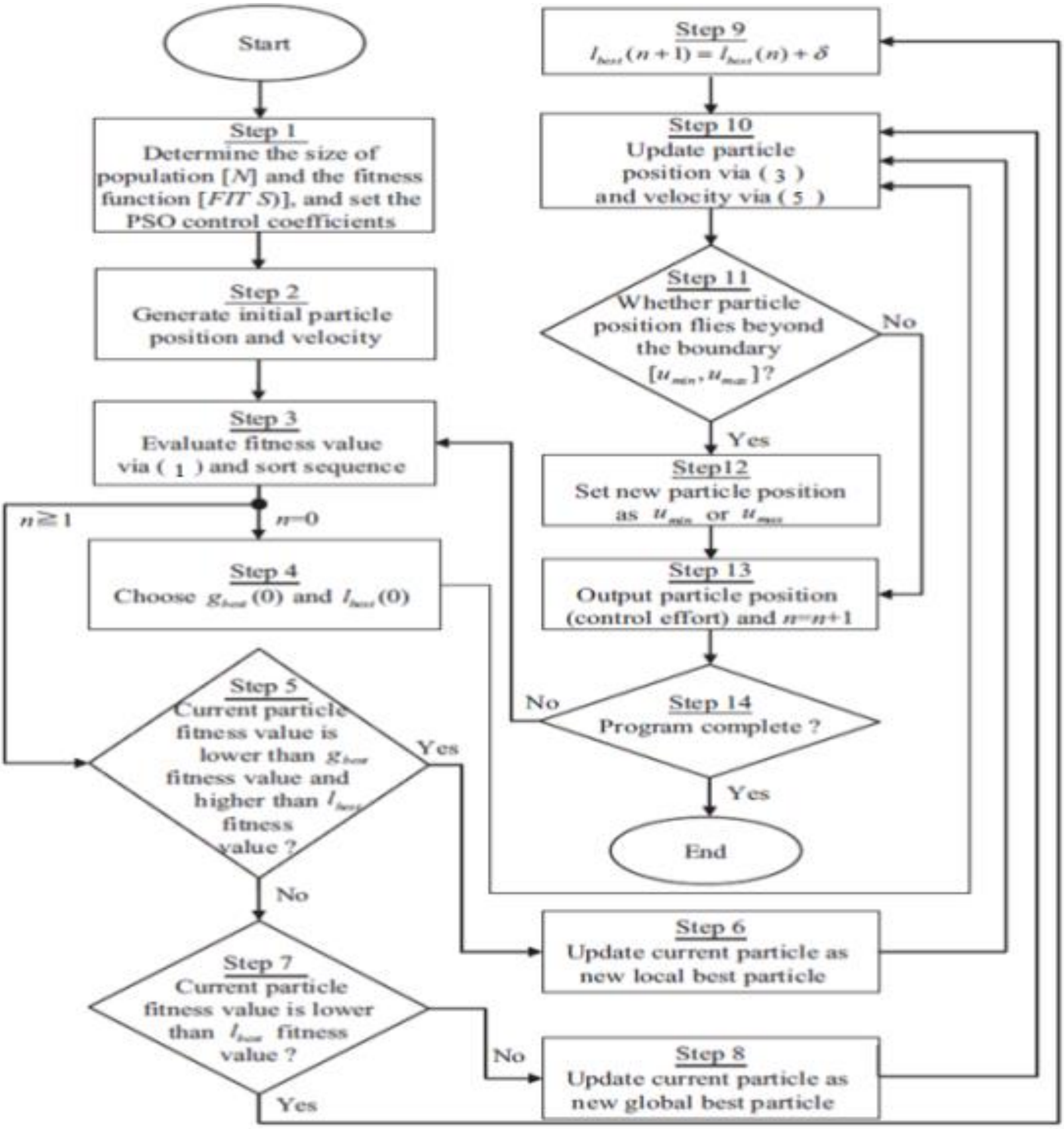

Fig.3. The Process Involved By PSO Using Flowchart

V. SIMULATION RESULTS

Simulation of induction motor with PSO and SMC is done by MATLAB/SIMULINK. . The error and derivative of error are used as the input of SMC block. The tracking error is reduced by using the sliding mode controllers and the boundary layer was introduced around the sliding surface, in order to eliminate the steady-state error which results in a continuous approximation of switching control. This property is helpful for defining the corresponding fitness function and particle velocity to become a direction-based PSO algorithm.

The PSO control scheme is utilized to be the major controller, and the stability can be indirectly incorporatd by the concept of SMC without strict constraint and detailed system knowledge. The output of PSO is given to given to the space vector pulse width modulation. Parameters of induction motor which has tested with SMC and PSO is shown in table IV.

Parameter	**Values**
Power	5.4 hp (4 KW)
Rated Speed	1430 rpm
Type	Squirrel cage
Voltage	400 V
Dc Link voltage (Vdc)	600 V
Stator Resistance (Rs)	1.405 Ω
Stator Inductance (Ls)	0.005839 H
Rotor Resistance (Rr)	1.395Ω
Rotor Inductance (Lr)	0.005839 H
Mutual Inductance(M)	0.1722 H
Number of Poles (P)	4
Inertia (J)	0.0131
Friction (B)	0.002985

Table 4: Parameters of Induction Motor Which Has Tested With PSO And SMC

The actual speed of Induction motor is compared with reference speed and produced the speed error as output. That steady state error gets reduced by continuous switching of sliding mode control. The particle swarm optimization finds the optimum error and that decides the width of pulse for PWM inverter. The V/F control method is done using Space vector pulse width modulation. Here frequency is set constant as 50 Hz and the magnitude gets changed to generate desired pulse for inverter. The inverter converts dc into ac supply and fed to induction motor. The simulation model using space vector pulse width modulation with sliding mode control and particle swarm optimization for induction motor is shown in figure 4. Most commonly Induction motors are controlled by PWM based drives. The main advantage is that the harmonics as well as switching losses gets reduced using space vector pulse width modulation.

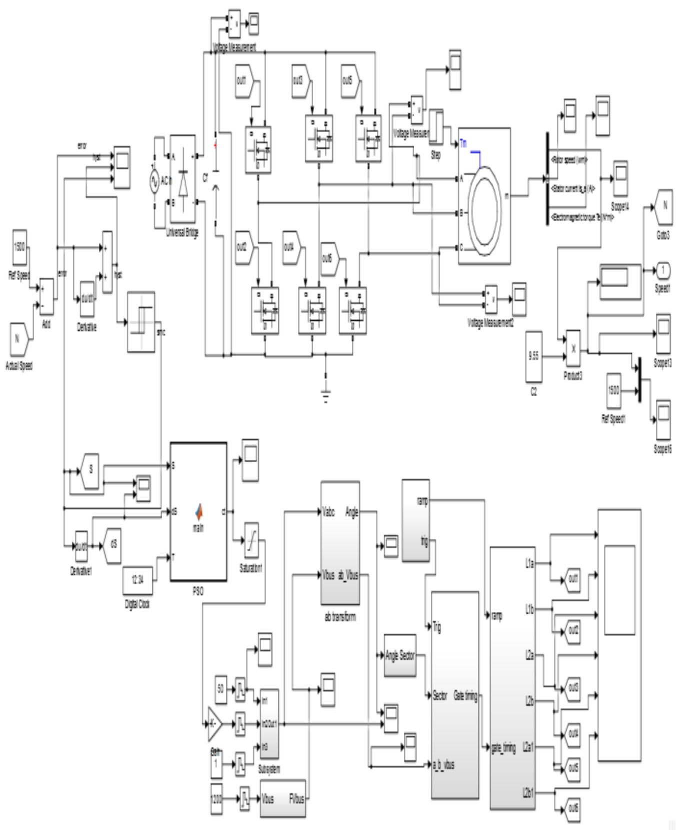

Fig. 4. Simulink Model of Induction Motor With PSO and SMC

Thus the speed gets controlled using space vector pulse width modulation with sliding mode control and particle swarm optimization. The switching frequency can be calculated by the ramp wave frequency. The switching frequency is 50Kz. The design of SMC and PSO controller for Space Vector Pulse Width Modulation inverter for induction motor V/f speed control modelling.The inverter's output is controlled by V/F technique and the output is given to the three phase induction motor. The speed has been set as set point while the actual speed is measured as feedback signal. The error and its derivative are used as the input of SMC block.

The output of SMC is the required stabilized error which will be integrated continuously. And the optimum error is decided by PSO block. In this case, the control signal which represents the voltage is sent to V/Hz block in order to maintain the ratio of voltage and frequency as constant to keep the torque as constant while the speed changes.

The output of V/Hz is used to generate SVPWM. The SVPWM pulse is given to the MOSFET as triggering pulse. The output voltage from the inverter is used to drive the motor in required speed which is in fundamental frequency. So the closed loop control of the SMC and PSO system works to achieve the required motor speed continuously and also keeps the system stable at any variations in load.

A. *Motor Speed*

Initially the reference speed is set at 1500 rpm. Figure 5 shows the starting characteristics of SMC and PSO controller using space vector pulse width modulation with no load. When the motor runs at no-load the peak overshoot is lesser and the value is 0.53. The settling time is 0.18 second.

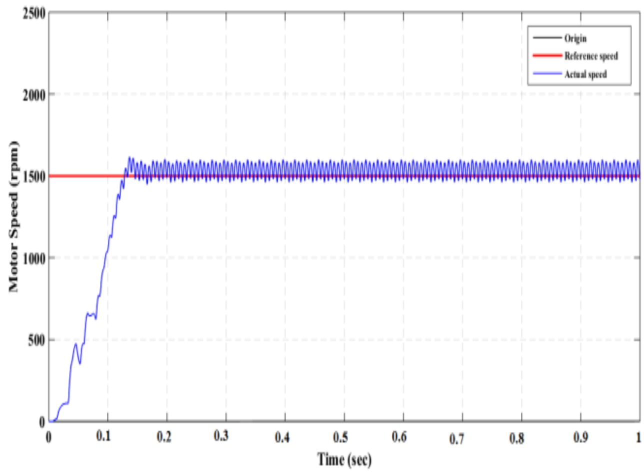

Figure 5 Starting Characteristics of IM With No Load-Motor Speed

B. *Stator Current*

At starting no load condition using SVPWM with PSO and SMC controller the time length of inrush current is up to 0.12 seconds and then it starts to settle at 0.18 second.

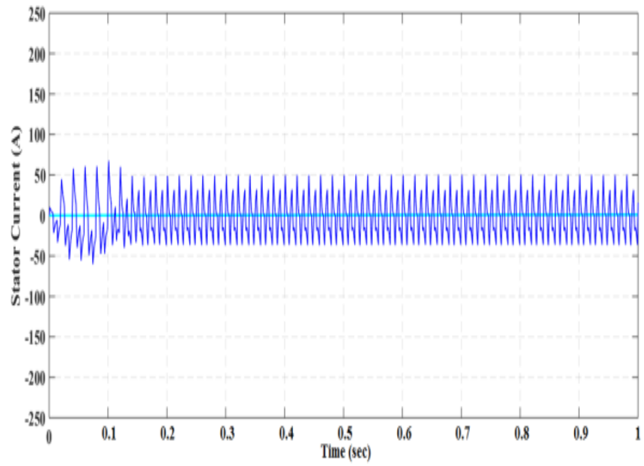

Figure 6 Starting Characteristics of IM With No Load-Stator Current

C. *Electromagnetic Torque*

At initial condition with no load, the Torque starts to reach the steady state at 0.11 second with lesser ripples and finally reaches the steady state at 0.18 sec with ripples.

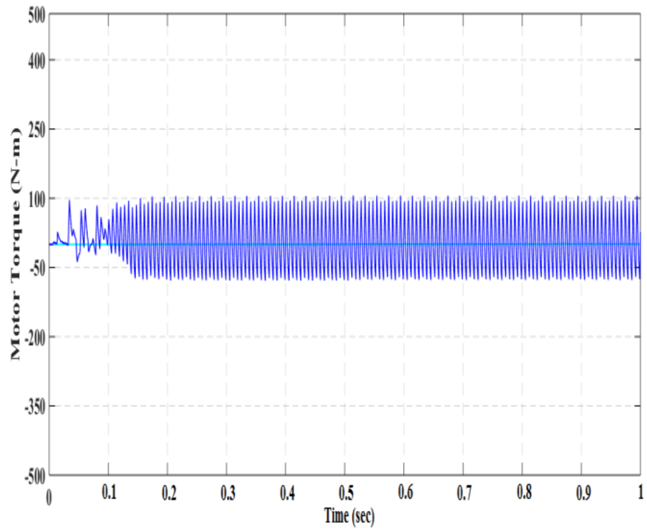

Figure 7 Starting Characteristics of IM With No Load-Starting Torque

Figure 8 shows the starting characteristics of SMC and PSO controller using space vector pulse width modulation with slight load of 2 N-m and the reference speed is set at 1400 rpm.

The speed has been settled at 0.18 second. At start the time length of inrush current in SMC and PSO is up to 0.08 second. The Torque reaches the steady state at 0.08 second in SMC and PSO using single pulse width modulation. Thus the Starting characteristics of IM with SMC & PSO at starting with slight load of 2 N-m is shown in figure 8.

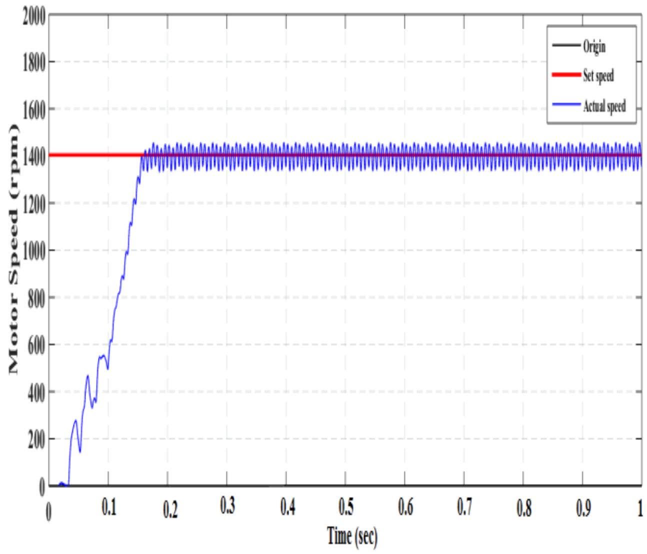

Figure 8 Starting Characteristics of IM With Slight Load of 2 Nm - Motor Speed

The starting characteristics of speed control of Induction motor with SMC and PSO using different PWM techniques namely single pulse width modulation and space vector pulse width modulation is observed.

S.No	Load Condition	Peak Overshoot %	Rise time (sec)	Peak time (sec)	Settling time (sec)
1	No load	0.07	0.08	0.14	0.18
2	Slight load (2 Nm)	0.01	0.03	0.04	0.18

Table 5: Performance In Time Domain Analysis of IM Using SVPWM

D. Total Harmonic Distortion

The Total Harmonic Distortion (THD) spectrum of stator current at starting time 0.1 second with PSO and SMC using single PWM is analysed for fundamental frequency 50 Hz and the THD observed is 12.27 % as shown in figure 9.So SVPWM has the better performance compared to single pulse width modulation.

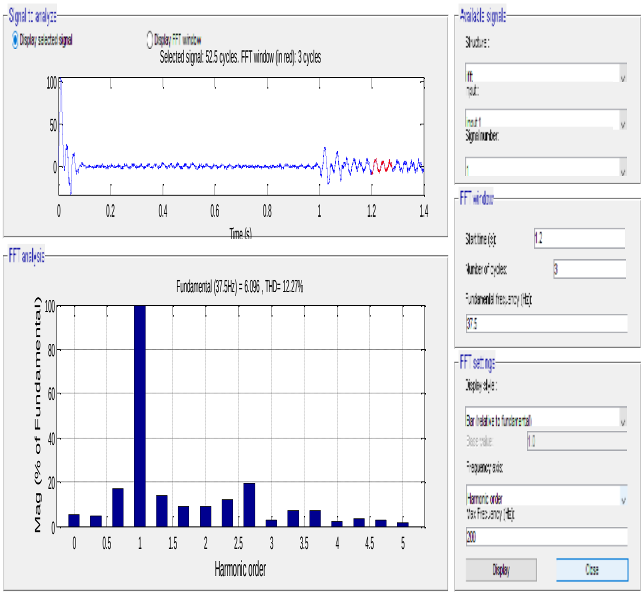

Figure 9 T.H.D of Stator Current At T = 0.1 Second

VI. CONCLUSION

This proposal of speed control of an AC drive using sliding mode control and particle swarm optimization controller with SVPWM shows swift, downy, and high dynamic response. 80 % of speed is achieved by tuning PSO parameters using space vector pulse width modulation. The stability of system dynamics is achieved by sliding mode control which reduces the steady state error by continuous switching. The MATLAB software is tested for the entire closed loop control of Induction motor using PWM techniques with particle swarm optimization (PSO) and sliding mode control. The performance of a system with variable load and the speed is good. The controller gives uniform torque and the THD value for stator current is 12.27%.

REFERENCES

[1]. Rong-Jong Wain, Yeou-Fu Lin, Kun-Lun Chuang, “Total sliding-mode-based particle swarm optimization control for linear induction motor”, www.elsevier.com/locate/jfranklin, Journal of the franklin institute 351 (2014) 2755–2780.

[2]. Divya Rai, Swati Sharma, Vijay Bhuria, “Fuzzy speed controller design of three phase induction motor”,International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, Volume 2, Issue 5, May 2012.

[3]. V.Vengatesan, M.Chindamani, “Speed Control of Three Phase Induction Motor using Fuzzy Logic Controller by Space Vector Modulation Technique”, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, ISSN 2320-3765 Vol. 3, Issue 12, December 2014.

[4]. B. K. Bose, Modern Power Electronics and AC Drive, Prentice Hall, 2002, PP.63-70.

[5]. RIOS-BOLIVAR, M., and ZINOBER, A.S.I.: 'Dynamical adaptive sliding mode control of observable minimum-phase urlcerlain non- linear systems'. in 'Variable Structure Systems, Sliding Mode and Nonlinear Control' (Springer-Veilag. Londan. UK, 1999),pp. 211-235.

[6]. RARTOLINI. G.. FERRARA. A,, GIACOMINI. L.. and USAI, E.: 'Properties of a combined adeptivc/smond-order sliding modc control algorithm for some classes of uncertain nonlinear systems', IEEE Trun.. Auro. Conrrol, 2000, 45, (7). pp. 1334-1341

[7]. R.C. Eberhart, J. Kennedy, A new optimizer using particles swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995, pp. 39–43.

[8]. [8] R.C. Eberhart, J. Kennedy, Particle swarm optimization, in: Proceedings of the IEEE Conference on Neural Networks, 1995, pp. 1942–1948.

[9]. [9] Y. Shi, R.C. Eberhart, A modified particle swarm optimizer, in: Proceedings of the IEEE Conference on Computation Intelligence, 1998, pp. 69–73.

[10]. [10] S.H. Ling, H.H.C. Iu, F.H.F. Leung, K.Y. Chan, Improved hybrid particle swarm optimized wavelet neural network for modeling the development of fluid dispensing for electronic packaging, IEEE Trans. Ind. Electron. 55 (9) (2008) 3447–3460.

[11]. [11] L.S. Coelho, B.M. Herrera, Fuzzy identification based on a chaotic particle swarm optimization approach applied toa nonlinear yo-yo motion system, IEEE Trans. Ind. Electron. 54 (6) (2007) 3234–3245.

ISSN No:-2456 –2165

[12]. [12] H.M. Feng, Self-generation RBFNs using evolutional PSO learning, Neurocomputing 70 (2006) 241–251.

[13]. [13] A.Chatterjee, K. Pulasinghe, K. Watanabe, K. Izumi, A particle-swarm-optimized fuzzy-neural network for voice controlled robot systems, IEEE Trans. Ind. Electron. 52 (6) (2005) 1478–1489.

[14]. [14] S. Panda, B.K. Sahu, P.K. Mohanty, Design and performance analysis of PID controller for an automatic voltage regulator system using simplified particle swarm optimization, J. Frankl. Inst. 349 (2012) 2609–2625.

[15]. [15] F.J. Lin, L.T. Teng, H. Chu, Modified Elman neural network controller with improved particle swarm optimisation for linear synchronous motor drive, IET Electr. Power Appl. 2 (3) (2008) 201–214.

[16]. [16] R.-J. Wai et al. / Journal of the Franklin Institute 351 (2014) 2755–2780 2779

[17]. [17] K.W. Yu, S.C. Hu, An application of AC servo motor by using particle swarm optimization based sliding mode controller, in: Proceedings of the IEEE Conference on Systems, Man, and Cybernetics, 2006, pp. 4146–4150.

[18]. [18] R. A. Krohling and L. S. Coelho, “Coevolutionary particle swarm optimization using Gaussian distribution for solving constrained optimization problems,” IEEE Trans. Syst., Man Cybern. B, Cybern., vol. 36, no. 6,pp. 1407–1416, Dec. 2006.

[19]. [19] A.Karimi, A. Feliachi, Decentralized adaptive backstepping control of electric power system, Electr. Power Syst.Res. 78 (3) (2008) 484–493.

[20]. [20] R. A. Krohling, F. Hoffmann, and L. S. Coelho, “Co-evolutionary particle swarm optimization for min–max problems using Gaussian distribution,”in Proc. Congr. Evol. Comput., Portland, OR, 2004, pp. 959–964.

[21]. [21] Y. Shi and R. C. Eberhart, “A modified particle swarm optimiser,” in Proc.IEEE World Congr. Comput. Intel., Evol. Comput., Anchorage, AL, 1998,pp. 69–73.

[22]. [22] L. S. Coelho and R. A. Krohling, “Predictive controller tuning using modified particle swarm optimization based on Cauchy and Gaussian distribution,” in Soft Computing: Methodologies and Applications, Advances in Soft Computing.