PSO BASED TUNING OF PI CONTROLLER FOR A DUAL STAR INDUCTION MACHINE FED BY A FIVE-LEVEL INVERTER

Dr. Bouziane. MELIANI
University Centre of Relizane, Algeria
Thiagarajar College, Madurai, India

Dr. Abdelkader. MEROUFEL
Djillali Liabès University of Sidi Bel Abbès (DLUSBA)
Intelligent Control and Electrical Power Systems Laboratory

ABSTRACT

Tuning methods for PID controllers are very important for the process industries. Traditional methods such as Ziegler-Nichols, pole placement, metho²xds often do not provide adequate tuning. In this connection, our proposed method is using Particle Swarm Optimization (PSO) to optimize PI-controller parameters; the PSO algorithm is used off-line to update the PI parameters for controlling the speed of the double star induction machine, the machine is fed by a five-level inverter. Analytical solutions of Pulse Width Modulation (PWM) strategies for multilevel Neutral Point Clamped (NPC) are presented. Digital simulation results demonstrate that the designed PSO-PI speed controller realize a good dynamic of the DSIM, a perfect speed tracking with no overshoot, give better performance and high robustness.

Key words: Dual star, Particle Swarm, Neutral Point Clamped, Multilevel Inverter.

Cite this Article: Dr. Bouziane. MELIANI and Dr. Abdelkader. MEROUFEL, PSO Based Tuning of PI Controller for a Dual Star Induction Machine Fed by a Five-Level Inverter. International Journal of Electrical Engineering & Technology, 8(4), 2017, pp. 26–35.
http://www.iaeme.com/IJEET/issues.asp?JType=IJEET&VType=8&IType=4

1. INTRODUCTION

The induction motor takes a large place in industry and we can find it in electrical ships. However, when high reliability is required, the control using the field-oriented method is often necessary and on-line diagnosis or off-line diagnoses have to be considered. The use of a double star induction motor (DSIM) or six-phase induction motor is slowly increasing and we find it in the high power process. Its main advantage lies in most reliability in case of a normal operating system [1]. Double star machine supplied with non sinusoidal waveforms causes perturbations in the torque (harmonics in torque). Recently for high performance power application multi level converters are widely used such as static vary compensators,
drives and active power filters. The advantage of multilevel inverter is good power quality, low switching losses and high voltage capability. Increasing the number of voltage levels in the inverter without requiring high ratings on individual devices can increases the power rating, and can gives sinusoidal waveforms, add at this multilevel structures which reduce harmonics in output voltages [3][4].

The DSIM it is desirable to control the flux and torque separately in order to have the same performances as those of DC motors. One way of doing this is by using the field oriented control. This method assures the decoupling of flux and torque [2]. The vector-controlled DSIM with a conventional PI speed controller is used extensively in industry, because has easily implemented. Alongside this success, the problem of tuning PI-controllers has remained an active research area. In recent years, many intelligence algorithms are proposed to tuning the PID parameters. Tuning PID parameters by the optimal algorithms such as the Simulated Annealing, Genetic Algorithm, and Particle Swarm Optimization (PSO) [10], this paper presents the design of PI controller based on PSO for the speed of a DSIM. The control system is modeled in Simulink and the PSO algorithm is implemented in MATLAB. The ultimate goal of this optimization algorithm is to find a global solution from a group of local solutions. These optimization algorithms are applicable to functions that are multi-modal, non-differentiable and discontinuous.

2. INDIRECT FIELD-ORIENTED CONTROL OF THE DOUBLE STAR INDUCTION MACHINE

The model of the machine that is established is described by a set of differential equations linking statoric and rotoric magnitudes. The dynamic behavior of this machine supplied by two current inverter is described by a model with rotor flux as state variable [2] [3].

\[ \frac{d\varphi_d^r}{dt} = -\frac{1}{T_r} \varphi_d^r + \omega_g \varphi_q^r + \frac{M^rf}{T_r} (i_{1d} + i_{2d}) \]  
(1)

\[ \frac{d\varphi_q^r}{dt} = -\frac{1}{T_r} \varphi_q^r + \omega_g \varphi_d^r + \frac{M^rf}{T_r} (i_{1q} + i_{2q}) \]  
(2)

\[ \omega_g: \text{being the slip pulsation the electromagnetique torque is described by the next equation} \]

\[ T_e = \frac{3}{2} p_1 \left( \frac{M^rf}{L_c^*} \right) \left[ \varphi_d^r (i_{1q} + i_{2q}) - \varphi_q^r (i_{1d} + i_{2d}) \right] \]  
(3)

P1: number of the pairs vectorial control by orientation of the flux necessitates a choice of judicious referential. The choice of a particular referential allows transforming the expression of the electromagnetic torque so that the induction machine can be compared to D.C machine at least by torque expression. In our study, we have chosen the case of an indirect control by orientation of the rotor flux, whose advantages resides in the simplification and the decoupling of equations of the system. In the model of the double-star asynchronous machine represented by equations (1 to 3), we choose a referential linked to the rotating field in order that the direct axis D coincides with the desired direction of the rotor flux. This choice of referential imposes \( \varphi^r(c) = \varphi^r_q(c) = 0 \) and \( \varphi^r_q(c) = 0 \). Thus equations (1 to 3) can be written as:

\[ T^r \frac{d\varphi_d^r}{dt} + \varphi_d^r = M^rf_c (i_{1d} + i_{2d}) \]  
(4)

\[ \omega_g = \frac{M^rf_c (i_{1q} + i_{2d})}{\varphi_d^r} \]  
(5)

\[ T_e = \frac{3}{2} p_1 \left( \frac{M^rf_c}{L_c^*} \right) \varphi_d^r (i_{1q} + i_{2q}) = \frac{3}{2} p_1 \frac{M^rf^2_c}{R_f} \omega_g \]  
(6)
The mechanical equation is given by:

\[ j \frac{d}{dt} \Omega + f \Omega = T_e - T_c \]  

(7)

With:

- \( T_e \): torque of load
- \( j \): moment of inertia
- \( f \): damping constant of the machine

The model that is established constitutes the main idea of the control of the double star asynchronous machine. We notice that components \( i_{1q}^s \) and \( i_{2q}^s \). If the rotor flux, while the torque depends only of quadrature components of the stator currents \( i_{1q}^s \) and \( i_{2q}^s \).

If the rotor flux \( \varphi^r \) is maintained constant. For the double star asynchronous machine supplied by current inverter. The stator currents \( (i_{1d}^s, i_{2d}^s, i_{1q}^s, i_{2q}^s) \) and the slip pulsation are considered as variables of control.

The dynamic model of speed induction motor drive is significantly simplified, and can be reasonably represented by the bloc diagram shown in Fig. 1.

**Figure 1** Block diagram of DSIM speed control with PI controller

The classic numerical PI (Proportional and Integral) regulator is well suited to regulating the torque, to the desired values as it is able to reach constant reference, by correctly both the \( P \) (Kp) and \( I \) (Ki) winches are respectively responsible for error sensibility and for the steady state error, the transfer function is as following (8):

\[ G(s) = \frac{K_p s + K_i}{j s^2 + (f + K_p) s + K_i} \]  

(8)

Where:

\[ P(s) = s^2 + \frac{f + K_p}{j} s + \frac{K_i}{j} = 0 \]  

(9)

The expressions for Kp and Ki of the regulator is calculated by imposition of poles complexes combined with real part negative s = (-1± j) \( \rho \)

\[ \begin{align*}
K_p &= 2p j - f \\
K_i &= 2j p^2
\end{align*} \]  

(10)

Where \( \rho \) It is a positive constant

3. FIVE-LEVEL INVERTER WITH NPC STRUCTURE

The NPC inverter uses a series string of capacitors to subdivide a single high voltage DC bus into the required number of voltage levels, and each phase leg output can be switched to any one of these levels. [4]. Two common multilevel inverter structures are the Neutral Point Clamped (NPC) inverter, and the Cascaded inverter. The NPC inverter uses one DC bus subdivided into a number of voltage levels by a series string of capacitors. The Cascaded
The inverter is made up from series connected single phase full bridge inverters, each with their own isolated DC bus [5].

For the NPC inverter, most carrier based PWM schemes that have been investigated derive from the carrier disposition strategies originally proposed by Carrara et al [6], i.e. for an NPC inverter with five levels, four triangular carriers with the same frequency and amplitude are arranged so that they fully occupy contiguous bands in the range of $+V_{DC}$ and $-V_{DC}$. A sinusoidal reference centered in the middle of the carrier set is then compared with each carrier to determine the voltage level that the converter should switch to. The inverter produces output voltage in five levels: 0, $+1/2V_{dc}$, $V_{dc}$, $-1/2V_{dc}$, and $-V_{dc}$ assuming that $V_{dc}$ is the supply voltage.

The carrying equations are given as follows:

\[
U_{p1}(t) = \begin{cases} 
U_{pm} \left( \frac{2t}{T_p} - 1 \right) & 0 \leq t \leq \frac{T_p}{2} \\
U_{pm} \left( \frac{2t}{T_p} - 1 \right) & \frac{T_p}{2} \leq t \leq T_p 
\end{cases}
\]

\[
U_{p2}(t) = \begin{cases} 
U_{pm} \left( \frac{2t}{T_p} - \frac{1}{2} \right) & 0 \leq t \leq \frac{T_p}{4} \\
U_{pm} \left( \frac{2t}{T_p} - \frac{3}{2} \right) & \frac{T_p}{4} \leq t \leq \frac{3T_p}{4} \\
U_{pm} \left( \frac{2t}{T_p} - \frac{5}{2} \right) & \frac{3T_p}{4} \leq t \leq T_p 
\end{cases}
\]

Figure 3 shows the reference and carrier waveform arrangements necessary to achieve PWM for a five level converter.

**Figure 2** Five-level tension inverter with NPC structure
4. TUNING OF PI CONTROLLER USING PARTICLE SWARM OPTIMIZATION APPROACH

4.1. Particle Swarm Optimization, PSO Overview

PSO is a population based optimization technique based on intelligent scheme developed by Kennedy and Eberhart (1995) [7]. PSO optimizes a problem having a population of the solutions, here dubbed particles, and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity. Each particle's movement is influenced by its local best known position and it is also guided towards the best known positions in the search-space, which are updated as better positions are found by other particles [8]. The fitness function evaluates the performance of particles to determine whether the best fitting solution is achieved. During the run, the fitness of the best individual improves over time and typically tends to stagnate towards the end of the run. Ideally, the stagnation of the process coincides with the successful discovery of the global optimum trained by updating generations. In an n-dimensional search space, the position and velocity of each particle is given by $x_i = [x_{i1}, x_{i2}, ..., x_{in}]^T$ and $v_i = [v_{i1}, v_{i2}, ..., v_{in}]^T$, respectively. The position of particle indicates the possible solution in the n-dimensional search space, whereas its velocity indicates the amount of change between the current and next positions. Corresponding to the personal best solution obtained so far at time $t$, each particle has its own best position, $p_i = [p_{i1}, p_{i2}, ..., p_{in}]^T$. The global best particle, $p_g$, represents the best particle found so far at time $t$ in the entire swarm [9]. The new velocity of each particle is calculated by (11)

$$v_{ij}(t + 1) = wv_{ij}(t) + c_1r_1(p_{ij} - x_{ij}(t)) + c_2r_2(p_{gi} - x_{ij}(t)) \quad (11)$$

where $w$ is the inertia weight and $c_1$ and $c_2$ are called acceleration coefficient. $r_1$ and $r_2$ are two independent random numbers Figure 4 shows the concept of modification of searching points highlighted by indications in the Equation (11).

$$x_{ij}(t + 1) = x_{ij}(t) + v_{ij}(t + 1), j = 1,2, ..., n \quad (12)$$

Where $w$ is the inertia weight and $c_1$ and $c_2$ are called acceleration coefficient. $r_1$ and $r_2$ are two independent random numbers Figure 4 shows the concept of modification of searching points highlighted by indications in the Equation (11).
The main steps in the particle swarm optimization and selection process are described as follows [8]:

Step 1: Initialize a population of particles with random positions and velocities in m-dimensions of the problem space and fly them.

Step 2: Evaluate the fitness of each particle in the swarm.

Step 3: For every iteration, compare each particle’s fitness with its previous best fitness \((P_{best})\) obtained. If the current value is better than \((P_{best})\). Then set \((P_{best})\) equal to the current value and the \((P_{best})\) location equal to the current location in the n-dimensional space.

Step 4: Compare \((P_{best})\) of particles with each other and update the swarm global best location with the greatest fitness \((g_{best})\).

Step 5: Change the velocity and position of the particle according to equations (9) and (10) respectively.

Step 6: Repeat steps (1) to (5) until convergence is reached based on some desired single or multiple criteria.

4.2. Design of PSO-PI Controller

The structure diagram of PI control system based on PSO is shown in Figure 4.

![Figure 4: Modification of a searching point by PSO](image)

Figure 4 Modification of a searching point by PSO

For a PID-controlled system, there are often four indices to depict the system performance: ISE, IAE, ITAE and ITSE. Therefore, for the PSO-based PID tuning, the ITAE performance index given in equation (3) will be used as the objective function. In other word,
the objective in the PSO-based optimization is to seek a set of PID parameters such that the feedback control system has minimum performance index [10] [11].

\[ ITAE = \int_0^\infty t|e(t)| \, dt \]  

(13)

PSO algorithm is employed to tune PID gains/parameters \((K_p, K_i)\) using the model in Equation (11). PSO algorithm firstly produces initial swarm of particles in search space represented by matrix. Each particle represents a candidate solution for PI parameters where their values are set in the range of 0 to 50. For this 2-dimensional problem, position and velocity are represented by matrices with dimension of 2xSwarm size. The swarm size is the number of particle where 5 are considered a lot enough.

A good set of PI controller parameters can yield a good system response and result in minimization of performance index in Equation (13).

The PSO algorithm method has been implemented as M file which interconnected to simulink model. At first, the PSO program module passes the initial value of the particles to the simulink module then, the simulink module calculates the fitness value that based on the performance criteria ITAE and outputs it to the PSO program module. So the cycle to repeat, until the maximum iteration number comes to the end or the performance criteria is satisfactory.

The optimal controlling parameters of the classical and proposed optimal PI controller are shown in Table 1 at the end of the test period.

<table>
<thead>
<tr>
<th>Table 1 Selection parameters of PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population Size</td>
</tr>
<tr>
<td>Number of Iterations</td>
</tr>
<tr>
<td>Velocity Constant, (c_1)</td>
</tr>
<tr>
<td>Velocity Constant, (c_2)</td>
</tr>
</tbody>
</table>

The PI parameters are obtained for 15 iterations:

<table>
<thead>
<tr>
<th>Table 2 Optimized PI parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tuning Method</strong></td>
</tr>
<tr>
<td>pole placement</td>
</tr>
<tr>
<td>PSO-PI(ITAE)</td>
</tr>
</tbody>
</table>

5. RESULTS AND DISCUSSION

The SIMULINK model for indirect FOC of the 4.5 Kw cage rotor DSIM associated with PI controller is shown in Fig.6. The machine is fed by five-level tension inverter with NPC structure. The parameters of the induction motor are summarized in Appendix.

The first test concerns a no-load starting of the motor with a reference speed \(\omega_{ref} = 288\) rad/s. and a nominal load disturbance torque (14N.m) is suddenly applied at 1.5s, followed by a consign inversion (-288rad/s) at 2s, this test has for object the study of controller behaviors in pursuit and in regulation.

The Fig.7 and Fig.8 shows dynamic performance of speed and torque respectively. From both the figure, this is clear that the speed and torque ripples are less in PSO-PI mode as compared to pole placement strategy. The comparison shows that the motor speed reached to 0.8 s compared with pole placement technique that reached to 1.4 s.
Figure 6 Simulink diagram for DSIM control systems

Figure 7 Speed response of PSO-PI controller comparing with the speed response of PID-conventional controller

Figure 8 Torque response of PSO-PI controller comparing with the speed response of PI-conventional controller
6. CONCLUSIONS

In this paper, the speed of indirect method of Vector Control of a Double Star induction Machine is controlled by the PSO-PI algorithms, the machine is fed by a five-level inverter. According to the results of the MATLAB simulation, the proposed controller using PSO algorithm is proved to be better than the controller tuned by pole placement method. From the results, the designed PI controller using PSO algorithm shows superior performance over the traditional method of pole placement, in terms of the system overshoot, settling time and rise time. However, the traditional method provides us with the initial PID gain values for optimal tuning. Therefore the benefit of using a modern artificial intelligence optimization approach is observed as a complement solution to improve the performance of the PI controller designed by conventional method. Of course there are many techniques can be used as the optimization tools and PSO is one of the recent and efficient optimization tools. The PSO method is an excellent optimization methodology and a promising approach for solving the optimal PI controller parameters problem.

APPENDIX

<table>
<thead>
<tr>
<th>Table 3 DSIM PARAMATERS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>N nominal values</strong></td>
</tr>
<tr>
<td>Power</td>
</tr>
<tr>
<td>Frequency</td>
</tr>
<tr>
<td>Voltage (∆/Y)</td>
</tr>
<tr>
<td>Current (∆/Y)</td>
</tr>
<tr>
<td>Speed</td>
</tr>
<tr>
<td><strong>Constant</strong></td>
</tr>
<tr>
<td>Poles of pair</td>
</tr>
<tr>
<td>Rs₁= Rs₂</td>
</tr>
<tr>
<td>Rr</td>
</tr>
<tr>
<td>Ls₁= Ls₂</td>
</tr>
<tr>
<td>Lr</td>
</tr>
<tr>
<td>Lₘ</td>
</tr>
<tr>
<td>J</td>
</tr>
<tr>
<td>Kᵣ</td>
</tr>
</tbody>
</table>

REFERENCES


