Speed Control of SRM Fed by Photovoltaic System Using Ant Colony Optimization Algorithm

A. S. Oshaba¹, E. S. Ali² and S. M. Abd Elazim³

¹Research Institute, Power Electronics and Energy Conversions, NRC Blg., El-Tahrir St., Dokki, 12311-Giza, Egypt, Email: oshaba68@hotmail.com
²Electric Power and Machine Department, Faculty of Engineering, Zagazig University, Zagazig, Egypt, Email: chabalimalisalama@yahoo.com
³Electric Power and Machine Department, Faculty of Engineering, Zagazig University, Zagazig, Egypt, Email: saharedeeph@yahoo.com

Abstract: This paper proposes a speed control of Switched Reluctance Motor (SRM) supplied by Photovoltaic (PV) system. The proposed design of speed controller is formulated as an optimization problem. Ant Colony Optimization (ACO) algorithm is employed to search for optimal Proportional Integral (PI) parameter of speed controller by minimizing the time domain objective function. The behavior of the proposed ACO has been estimated with the behavior of Genetic Algorithm (GA) in order to prove the superior efficiency of the proposed ACO in tuning PI controller over GA. Also, the behavior of the proposed controller has been estimated with respect to the change of load torque, variable reference speed, ambient temperature, and radiation. Simulation results confirm the better behavior of the optimized PI controller based on ACO compared with optimized PI controller based on GA over a wide range of operating conditions. Simulation results have shown the validity of the proposed technique in controlling the speed of SRM.

Keywords: Ant Colony Optimization; Genetic Algorithm; High Speed SRM; Speed Control; PI Controller; Photovoltaic System.

1. Introduction
Over the past decades, the switched reluctance motors (SRMs) have been the focus of several researches [1–2]. The SRM has a simple, rugged, and low-cost structure. It has no Permanent Magnet (PM) or winding on the rotor. This structure not only reduces the cost of the SRM but also offers high speed operation capability for this motor. Unlike the induction and PM machines, the SRM is capable of high speed operation without the concern of mechanical failures that result from the high level centrifugal force. In addition, the inverter of the SRM drive has a reliable topology. The stator windings are connected in series with the upper and lower switches of the inverter. This topology can prevent the shoot through fault that exists in the induction and permanent motor drive inverter. Moreover, high efficiency over wide speed range and control simplicity is known merits of the SRM drive [3–4].

Several Artificial Intelligence (AI) techniques have been addressed in literatures to solve problems related to the speed control of SRM. In last few years, Fuzzy Logic Control (FLC) has received much attention in the control applications. In contrast with the conventional techniques, FLC formulates the control action of a plant in terms of linguistic rules drawn from the behavior of a human operator rather than in terms of an algorithm synthesized from a model of the plant [5–13]. It offers the following advantages: they do not require an accurate model of the plant, they can be designed on the basis of linguistic information obtained from the previous knowledge of the control system and give better performance results than the conventional controllers. However, a hard work is inevitable to get the effective signals when designing FLC. Also, it requires more fine tuning and simulation before operational. Another AI approach likes Artificial Neural Network (ANN) for designing adaptive speed control of SRM is presented in [14–15]. The ANN approach has its own advantages and disadvantages. The performance of the system is improved by ANN based controller but, the main problem of this controller is the long training time, the selecting number of layers and the number of neurons in each layer.

H∞ optimization techniques have been applied to robust speed control problem [16–17]. However, the importance and difficulties in the selection of weighting functions of the H∞ optimization problem have been reported. Also, the additive and/or multiplicative uncertainty representation cannot treat situations where a nominal stable system becomes unstable after being perturbed. Moreover, the pole-zero cancellation phenomenon associated with this approach produces closed loop poles whose damping is directly dependent on the open loop system. On the other hand, the order of the H∞ based controller is as high as that of the plant. This gives rise to complex structure of such controllers and reduces their applicability.

Recently, global optimization techniques have attracted the attention in the field of controller parameter optimization. Genetic Algorithm (GA) is illustrated in [18] for optimal design of speed control of SRM. Despite this optimization technique requires a very long run time that may be several minutes or even several hours depending on the size of the system under study. Swarming strategies in fish schooling and bird flocking are used in the Particle Swarm Optimization (PSO) and presented in [19] for optimal design of speed control of different motors [20–22]. However, PSO suffers from the partial optimism, which causes the less exact at the regulation of its speed and the direction. In addition, the algorithm cannot work out the problems of scattering and optimization [23, 24]. Also, the algorithm pains from slow convergence in refined search stage, weak local search ability.
and algorithm may lead to possible entrapment in local minimum solutions. A relatively newer evolutionary computation algorithm, called Bacteria Foraging (BF) scheme has been presented by [25–27] and further established recently by [28–34]. The BF algorithm depends on random search directions which may lead to delay in reaching the global solution.

In order to solve the above mentioned problems and drawbacks, this paper proposes the use of a new evolutionary algorithm known as Ant Colony Optimization (ACO) algorithm to design a robust speed controller for SRM. ACO is multi-agent system in which the behavior of each single agent, called artificial ant or ant is inspired by the behavior of real ants [35]. ACO has been successfully employed to optimization problems in power system such as power quality enhancement [36], optimal reactive power dispatch [37]. The feature of technique presentation is different from other method since it can be implemented easily; flexible for many problem formulations and finally its capability in avoiding the occurrences of local optima for a given problem [38].

This paper proposes a new optimization algorithm known as ACO for controlling high speed SRM supplied by PV system. ACO is used for tuning the PI controller parameters to control the duty cycle of DC/DC converter and therefore speed control of SRM. The design problem of the proposed controller is formulated as an optimization problem and ACO is employed to search for optimal controller parameters. By minimizing the time domain objective function representing the error between reference speed and actual one, the system performance is improved. Simulation results assure the effectiveness of the proposed controller in providing good speed tracking system over a wide range of load torque, ambient temperature and radiation with minimum overshoot/undershoot and minimal settling time. Also, these results assure the superiority of the proposed ACO method in tuning controller compared with GA.

2. System under Study

The system under study consists of PV system acts as a voltage source for a connected SRM. The speed control loop is designed using ACO. The speed error signal is obtained by comparing between the reference speed and the actual speed. The output of the ACO controller is denoted as duty cycle. The schematic block diagram is shown in Fig. 1.

2.1 Construction of SRM

The construction of a 8/6 (8 stator poles, 6 rotor poles) poles SRM has doubly salient construction [39]. Usually, the number of stator and rotor poles is even, and the construction is well explained as in Fig 2. The windings of the SRM are simpler than those of other types of motors, and winding exists only on stator poles, and is simply wound on it with no winding on the rotor poles. The winding of opposite poles is connected in series or in parallel forming a number of phases, and exactly half the number of stator poles, and the excitation of a single phase excites two stator poles. The rotor has a simple laminated salient pole structure without winding. SRMs have the advantage of reducing copper losses while its rotor is winding. Its windings are made preferably of silicon steel, especially in higher efficiency applications. For aerospace application the rotor operates at very high speeds, requiring the use of cobalt, iron and other variants. The air gap is kept as minimum as possible, and the rotor and stator pole arc should be kept the similar. It is advantageous if the rotor pole arc is larger than the stator pole arc [40–41]. The construction of an 8/6 SRM is shown in Fig. 2.

Torque is developed in SRMs due to the tendency of the magnetic circuit to adopt the configuration of minimum reluctance i.e. the rotor moves in line with the stator pole thus maximizing the inductance of the excited coil. The magnetic behavior of the SRM is highly nonlinear. The static torque produced by one phase at any rotor position is calculated using the following equations [40–41].

\[ \text{Co energy} = W' = \int \psi (\theta, i) di \]  

(1)

\[ \text{Static torque} = T_{\text{static}} = dW' / d\theta \]  

(2)

From equations (1) and (2) a similar static torque matrix can be estimated where current will give the row index and \( \theta \) will give the column index as in [40–41]. The value of developed torque can be calculated from the static torque look up table by using second order interpolation method by used them the current value and \( \theta \).

The value of actual speed can be calculated from the following mechanical equations:

\[ da / dt = (T(\theta, i) - T_{\text{mech}}) / J \]  

(3)

where, the speed error is obtained from the difference between the rotor speed and its reference. The value of rotor
angular displacement $\theta$ can be calculated from the following equation:
$$d\theta/dt = \omega$$
(4)
where $\delta$ is the angle corresponding to the displacement of phase A in relation to another phase is given by:
$$\delta = 2\pi \left(\frac{1}{N_r} - \frac{1}{N_s}\right)$$
(5)
where $N_r$ and $N_s$ are the number of rotor and stator poles respectively. Also, the positive period of phase is determined by the following equation:
$$duty \ period = 2\pi \left(\frac{1}{qN_r}\right)C_r$$
(6)
where $q$ is number of phases and $C_r$ is the commutation ratio.

$C_r$ can be calculated by the following equation.
$$C_r = 2\pi \left(\frac{1}{\beta_r} - \frac{1}{\beta_s}\right)$$
(7)
Where $\beta_r$, $\beta_s$ are the stator and rotor pole respectively.

Duration of negative current pulses is depended on the stored energy in phase winding. On running, the algorithm is corrected by PI controller. This method is suitable with special range for turn on angle. The parameters of SRM are shown in appendix.

### 2.2 Photovoltaic System
Solar cell mathematical modeling is an important step in the analysis and design of PV control systems. The PV mathematical model can be obtained by applying the fundamental physical laws governing the nature of the components making the system [42].

To overcome the variations of illumination, temperature, and load resistance, voltage controller is required to track the new modified reference voltage whenever load resistance, illumination and temperature variation occurs. I-V characteristics of solar cell are given by the following equations [43-44]:

$$I_c = I_{ph} - I_o \left(\frac{q_o}{AKT} \left(V_c + I_c R_s\right) - 1\right)$$
(8)
$$V_c = AKT \left(\frac{I_{ph} + I_o - I_c}{I_o}\right)e^{\frac{-q_o}{AKT} \left(V_c + I_c R_s\right)} - I_c R_s$$
(9)
$$I = I_{ph} - I_o \left(\frac{q_o}{n_s AKT} \left(V + n_s I R_s\right)\right)$$
(10)
$$V = n_s AKT \left(\frac{I_{ph} + I_o - I}{I_o}\right)e^{\frac{-q_o}{AKT} \left(V + n_s I R_s\right)} - n_s I R_s$$
(11)

where;
$$I_{ph} = \frac{G}{1000} \left[I_{sc} + k_i \left(T - T_p\right)\right]$$
(12)
$$I_o = I_{or} \left(\frac{T}{T_p}\right)^{\frac{3}{4}} e^{\frac{q_o E g}{AKT} \left(1 - \frac{1}{T} - \frac{1}{T_p}\right)}$$
(13)

The module output power can be determined simply from
$$P = V I$$
(14)

where;
$I$ and $V$: Module output current and voltage;
$I_c$ and $V_c$: Cell output current and voltage;
$I_{ph}$ and $V_{ph}$: The light generation current and voltage,
$I_s$: Cell reverse saturation current,
$I_{sc}$: The short circuit current,
$I_o$: The reverse saturation current,
$R_s$: The module series resistance,
$T$: Cell temperature,
$K$: Boltzmann’s constant,
$q_o$: Electronic charge,
$K_T$: (0.0017 A/°C) short circuit current temperature coefficient.
$G$: Solar illumination in W/m²,
$E_g$: Band gap energy for silicon,
$A$: Ideality factor,
$T_r$: Reference temperature,
$I_{or}$: Cell rating saturation current at $T_r$,
$n_s$: Series connected solar cells,
$k_i$: Cell temperature coefficient.

Thus, if the module parameters such as module series resistance ($R_s$), reverse saturation current ($I_o$), and ideality factor ($A$) are known, the I-V characteristics of the PV module can be simulated by using equations (12 and 13). PV system is used in this paper to power SRM. The parameters of PV system are given in appendix.

### 2.3 DC-DC Converter
The choice DC-DC converter technology has a significant impact on both efficiency and effectiveness. Many converters have been used and tested; buck converter is a step down converter, while boost converter is a step up converter [45-46]. In this paper, a hybrid (buck and boost) DC/DC converter is used. The equations for this converter type in continuous conduction mode are:

$$V_B = -\frac{k}{1-k} V_{ph}$$
(15)
$$I_B = \frac{k-1}{k} I_{ph}$$
(16)
where $k$ is the duty cycle of the Pulse Width Modulation (PWM) switching signal. $V_B$ and $I_B$ are the output converter voltage and current respectively. The Matlab/Simulink of PV system can be simulated as shown in Fig. 3.

![Matlab/Simulink for PV system](image)

**Fig. 3. Matlab/Simulink for PV system.**

### 3. Objective Function

A performance index can be defined by the Integral of Time multiply Absolute Error (ITAE). Accordingly, the objective function $J_t$ is set to be:

$$J_t = \left[ \int_{t_{ref}}^{t_{act}} (w_{ref} - w_{act}) dt \right]$$  \hspace{1cm} (17)

Where $w_{ref}$ is the reference and $w_{act}$ is the actual.

Based on this objective function $J_t$, optimization problem can be stated as: Minimize $J_t$, subjected to:

$$\min \{ K_p, K_i \} \leq \max \{ K_p, K_i \}$$  \hspace{1cm} (18)

This paper focuses on optimal tuning of PI controller for speed tracking of SRM using ACO algorithm. The aim of the optimization is to search for the optimum controller parameters setting that minimize the difference between reference speed and actual one. On the other hand, in this paper the goal is speed control of SRM and finally designing a low order controller for easy implementation.

### 4. Overview of ACO and GA Optimization Technique

#### 4.1 Ant Colony Optimization

The first ACO algorithm was introduced by Marco Dorigo [35]. The development of this algorithm was inspired by the observation of ant colonies. The behavior that provided the inspiration for ACO is the ants’ foraging behavior, and in particular, how ants can find shortest paths between food sources and their nest. When searching for food, ants initially explore the area surrounding their nest in a random manner. While moving, ants leave a chemical pheromone trail on the ground. The pheromone quantity depends on the length of the path and the quality of the discovered food source [47]. An ant chooses an exact path in connection with the intensity of the pheromone. The pheromone trail evaporates over time if no more pheromone is laid down. Other ants are attracted to follow the pheromone trail. Therefore, the path will be marked again and it will attract more ants to use the same path. The pheromone trail on paths leading to rich food sources close to the nest will be more frequented and will therefore grow faster. In this way, the best solution has more intensive pheromone and higher probability to be chosen. The described behavior of real ant colonies can be used to solve optimization problems in which artificial ants search the solution space by transiting from nodes to nodes. The artificial ants movement usually associated with their previous action that stored in the memory with a specific data structure [48]. The pheromone consistencies of all paths are updated only after the ant finished its tour from the first node to the last node. Every artificial ant has a constant amount of pheromone stored in it when the ant proceeds from the first node. The pheromone that has been stored will be evenly distributed on the path after artificial ants finished its tour. The amount of pheromone will be high if artificial ants finished its tour with a good path and vice versa. The pheromone of the routes progressively decreases by evaporation in order to avoid artificial ants stuck in local optima solution [48-49]. The ACO algorithm can be divided into the following steps:

**Step 1: Initialization**

In this step, the following parameters $(n, m, \max_d, \max_k, \beta, \alpha, q_a, \tau_o)$ of ACO algorithm are initialized.

Where

- $n$: Number of nodes,
- $m$: Number of ants,
- $\max_t$: Maximum iteration,
- $\max_d$: Maximum distance for each ant’s tour,
- $\beta$: Parameter determines the relative importance of pheromone versus distance ($\beta > 0$),
- $\rho$: Heuristically defined coefficient ($0 < \rho < 1$),
- $\alpha$: Pheromone decay parameter ($0 < \alpha < 1$),
- $q_a$: Parameter of the algorithm ($0 < q_a < 1$),
- $\tau_o$: Initial pheromone level,

The maximum distance for every ant’s tour $d_{max}$ can be calculated using the following equation:

$$d_{max} = \max \left[ \frac{n-1}{\sum_{i=1}^{n} d_i} \right]$$  \hspace{1cm} (19)

$$d_i = \left| r - \max(a) \right|$$  \hspace{1cm} (20)

**Step 2: Provide first position**

Generate first position randomly; the first node will be selected by generating a random number according to a uniform distribution, ranging from 1 to $n$. 

Step 3: Transition rule
The probability for an ant $k$ at node $i$ to choose next node $j$ can be expressed as:

$$p_{ij}^k(t) = \frac{\tau_{ij}^k(t)^{\alpha} \eta_{ij}^k(t)^{\beta}}{\sum_{j' \in T^k} \tau_{ij'}^k(t)^{\alpha} \eta_{ij'}^k(t)^{\beta}} , \quad i, j \in T^k$$

(21)

Where:

- $\tau_{ij}^k(t)$: The pheromone trial deposited between node $i$ and $j$ by ant $k$.
- $\eta_{ij}^k(t)$: The visibility and equal to the inverse of the distance ($\eta_{ij} = 1/d_{ij}$).
- $\tau_k^k$: The path effectuated by the ant $k$ at a given time.

Step 4: Local pheromone updating
Local updating pheromone is different from ant to other because each ant takes a different route. The initial pheromone of each ant is locally updated as shown below.

$$\tau_{ij}^k(t+1) = (1-\rho)\tau_{ij}^k(t) + \rho\tau_0$$

(22)

Step 5: Fitness function
After all ants attractive to the shortest path that having a strongest pheromone, the best solution of the objective function is obtained.

Step 6: Global pheromone updating
Amount of pheromone on the best tour becomes the strongest due to attractive of ants for this path. Moreover, the pheromone on the other paths is evaporated in time. The pheromone level is updated by applying the following equation:

$$\tau_{ij}^k(t+1) = (1-\alpha)\tau_{ij}^k(t) + \alpha\Delta \tau_{ij}^k(t)$$

(23)

Step 7: Program termination
The program will be terminated when the maximum iteration is reached or the best solution is obtained without the ants stagnations. The proposed procedure steps are shown in Fig. 4. The parameters of ACO are shown in appendix.

4.2. Genetic Algorithm (GA)
In the animal kingdom, animals evolve and generate according to the role of "survival of the fittest". In nature, animals fight constantly for food, shelter and mates. Thus, only the fittest will survive and the weak will perish. This mechanism of weeding out the useless has worked perfectly for centuries and it is a good method for optimization. GA is such an optimization method. It is based on the mechanics of natural selection and natural genetics. The search process is very similar to the natural evolution of biological creature in which successive generations of organisms are given birth and raised until they are able to breed. Just like in animal kingdom, only the fittest will survive to produce while the weakest will be eliminated [50].

Four main parameters affect the performance of GAs: population size, number of generations, crossover rate, and mutation rate. Larger population size and large number of generations increase the likelihood of obtaining a near-global optimum solution, but substantially increase processing time. Crossover among parent chromosomes (solution vectors) is a common natural process and traditionally is given a rate that ranges from 0.6 to 1.0. In crossover, the exchange of parents’ information produces an offspring. As opposed to crossover, mutation is a rare process that resembles a sudden change to an offspring. This can be done by randomly selecting one chromosome from the population and then arbitrarily changing some of its information. The benefit of mutation is that it randomly introduces new genetic material to the evolutionary process, perhaps thereby avoiding stagnation around local minima. A small mutation rate less than 0.1 is usually used [51]. A flowchart for the GA algorithm is shown in Fig. 5. The parameters of GA are shown in appendix.

Step 1: Initialization; Input parameters, control limits and initial population
Step 2: Generate the initial population
Step 3: Calculate the fitness of each individual
Step 4: Apply state transition rule
Step 5: Apply local pheromone updating rule
Step 6: Fitness function evaluation
Step 7: Apply global pheromone updating rule
Step 8: Max iteration is reached

Yes

Output of the best individual

Yes

Create new Generation: Selection, Crossover, and Mutation

No

Maximal number of generation?

Start

End
5. Results and Simulations

In this section, the superiority of the proposed ACO algorithm over GA in designing PI controller for speed control of SRM is illustrated. Fig. 6. shows the variations of objective function with two optimization techniques. The objective functions decrease monotonically over generations of ACO and GA. Moreover, ACO converges at a faster rate (35 generations) compared with GA (50 generations). Moreover, computational time (CPU) of both algorithms is compared based on the average CPU time taken to converge the solution. The average CPU for ACO is 32.1 s while it is 43.9 s for GA. The proposed ACO methodology and GA are programmed in MATLAB 7.1 and run on an Intel(R) Core(TM) i5 CPU 2.53 GHz and 4.00 GB of RAM. The mentioned CPU time is the average of 10 executions of the computer code. Table 1. shows the parameters of PI controller, average settling time, and average percentage overshoot based on two optimization techniques. It can be seen that the parameters for ACO are smaller than GA. Hence, compared to GA, ACO greatly enhances the time domain characteristics for SRM.

5.1 Response under step change in load torque

Fig. 7 shows the step change in load torque of SRM. The speed response and control signal for this case are shown in Figs. 8-9 respectively. These Figures indicate the capability of the ACO in reducing the settling time and system oscillations over GA. Moreover, the actual speed tracks the reference speed rapidly. The settling time is approximately 0.06, and 0.064 second for ACO and GA respectively. Hence, the proposed ACO is capable of providing sufficient speed tracking compared with GA.

5.2 Response under variable speed and load torque

In this case, the system responses under variation of reference speed and load torque are obtained. Fig. 10. shows the variation of the load torque as an input disturbance while the...
parameters of PV system are constant. Moreover, the system responses for different controllers are shown in Figs. 11 and 12. It is clear from these Figs.; the proposed ACO algorithm outperforms and outlasts GA in controlling the speed of SRM and reducing settling time effectively. Therefore, compared with GA based controller, ACO based controller greatly enhances the system performance.

5.3 Response under variable load torque, reference speed and PV parameters

In this case, variations of load torque, reference speed, and PV parameters are applied. Fig. 13 shows the change of load torque, radiation and temperature respectively. Moreover, the system responses for both controllers are shown in Figs. 14 and 15. It is clear from these Figs, that the proposed ACO is more efficient in improving speed control of SRM compared with GA. Also, the proposed controller has a smaller settling time and system response is quickly driven with the reference speed. Thus, the potential and superiority of the proposed ACO over GA is demonstrated.
6. Conclusions
In this paper, a new method for speed control of SRM (8/6 poles) is proposed via ACO. The design problem of the proposed controllers is formulated as an optimization problem and ACO is employed to search for optimal parameters of PI controller. By minimizing the time domain objective function, in which the difference between the reference and actual speed are involved; speed control of SRM is improved. Simulation results emphasis that the designed ACO based PI controller is robust in its operation and gives a superb performance over GA for the change in load torque, reference speed, radiation, and temperature. Besides the simple architecture of the proposed controller, it has the potentiality of implementation in real time environment.

Appendix
The optimization parameters are as shown below:

a) Genetic parameters: Max generation=100; Population size=50; Crossover probabilities=0.75; Mutation probabilities =0.1.

b) ACO parameters: n=10, m=5, t_max=5, d_max =49, β=2, ρ =0.6, α =1, q_z =0.6, τ =0.1.

c) SRM parameters: N_s =8, N_p =6, Rating speed =13700 r.p.m, C_r =0.8, q =4, Phase resistance of stator=17 ohm, Phase inductance of aligned position=0.605 H, Phase inductance of unaligned position=0.1555 H, Step angle=15°.

d) PV parameters: A = 1.2153; E_g = 1.11; I_{so} = 2.35e-8; I_{sc} =4.8; T_r =300; K= 1.38e-23; n_s =36; q_o =1.6e-19; k_l =0.0021.

References

Table 2. Values of performance indices.

<table>
<thead>
<tr>
<th>Performance index</th>
<th>IAE</th>
<th>ISE</th>
<th>ITSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACO</td>
<td>12.0937</td>
<td>19.1421</td>
<td>72.1439</td>
</tr>
<tr>
<td>GA</td>
<td>13.3529</td>
<td>20.9312</td>
<td>76.3588</td>
</tr>
</tbody>
</table>

Fig. 15. Change in control signal for different controllers.

5.4 Robustness and performance indices
To demonstrate the robustness of the proposed controller, three different performance indices are used. These indices are: The Integral Absolute value of the Error (IAE), the Integral of the Square value of the Error (ISE), and the Integral of the Time multiplied Square value of the Error in time domain characteristic [52]. Numerical results of performance robustness for variations of load torque, reference speed, and PV parameters are listed in Table 2. It can be seen that the values of these indices corresponding to ACO are smaller compared to those of GA. This demonstrates that the overshoot, undershoot, and settling time are reduced by applying the proposed ACO based controller.


