Open access • Proceedings Article • DOI:10.1049/CP.2014.0298

Induction machine Winding Faults Identification using Bacterial Foraging Optimization technique — Source link

Salaheddine Ethni, Shady Gadoue, Bashar Zahawi

Institutions: Newcastle University

Published on: 08 Apr 2014

Topics: Fault (power engineering), Wound rotor motor, Rotor (electric) and Condition monitoring

Induction Machine Winding Faults Identification using Bacterial Foraging Optimization Technique

S. A. Ethni, S. M. Gadoue and B. Zahawi

School of Electrical and Electronic Engineering, Newcastle University, Newcastle upon Tyne NE1 7RU, UK

Email: salaheddine.ethni@ncl.ac.uk

Abstract-**The performance of a stochastic search algorithm, Bacterial Foraging Optimization (BFO), when used for fault identification of induction machine stator and rotor winding faults, is investigated in this paper. The proposed condition monitoring technique uses time domain terminal data in conjunction with the optimization algorithm and an induction machine model to indicate the presence of a fault and provide information about its nature and location. The proposed technique is evaluated using experimental data obtained from a 1.5 kW wound rotor three-phase induction machine. BFO is shown to be effective in identifying the type and location of the fault without the need for prior knowledge of various fault signatures.**

Index Terms-- **Induction machine, bacterial foraging algorithm, condition monitoring.**

I. INTRODUCTION

Induction motors are used in a wide variety of domestic and industrial applications due to their simple construction, ruggedness, low price and efficiency. The monitoring the condition of the motor is essential to detect any developing fault at an early stage, reducing the risks of severe motor faults. Faults can then be treated before completely damaging the motor, thus decreasing the maintenance cost and shutdown time. Consequently, there is an increasing need for a simple and reliable technique to detect incipient motor faults. Traditional induction machine condition monitoring techniques [1] usually involve the use of sensors embedded in the machine to measure, for example, temperature or vibration [2]. There has also been considerable interest in detecting windings and other machine faults by examination of terminal current waveforms [3] using data gathered under steady-state operating condition. This may involve the calculation of quantities such as input power [4] or negative sequence components [5]. Recent trends in condition monitoring include the detection of machine faults using data acquired during speed transients [6] and the estimation of machine parameters [7-11].

A new fault identification technique using machine terminal data and rotor position information has been recently proposed by the authors [8-10]. In this method, a stochastic search is carried out to estimate the values of machine parameters which give the best possible match between the performance of the faulty experimental machine and its mathematical model, thus identifying both the location and nature of the winding fault. Figure 1 shows a schematic diagram of the fault identification technique. Stator currents are calculated from an induction motor dynamic model and compared to the actual measured currents to produce a set of current errors that are integrated then summed to give an overall error function. When the machine is in its healthy state, there is a high correlation between its effective parameters and the model parameters resulting in a small calculation error. If a fault develops in the machine, its electrical parameters are of course modified and when the measured currents are compared with calculated currents there will be a large calculation error giving a fast indication that a fault of some type is present. Fault identification is carried out by adjusting the model parameters, using a stochastic search method to minimize the error. The new set of model parameters then defines the nature and location of the fault. Unlike many other methods, it should be noted here that the new stochastic search based approach does not require any expert prior knowledge of the type of fault or its location; both are identified as an integral part of the optimisation process.

The Fault identification technique proposed in this paper is based on Bacterial Foraging Optimization (BFO). This stochastic algorithm continuously adjusts the induction machine model parameters off-line to achieve the minimum error between the measured and calculated stator currents. The new set of model parameters defines the nature and location of the fault.

Experimental tests based on a 1.5 kW wound rotor three phase induction machine have been carried out to validate the proposed fault identification algorithm with stator and rotor faults considered. Results confirm the capability of BFO to identify and locate the fault without the need for a previous knowledge of different fault current signatures.

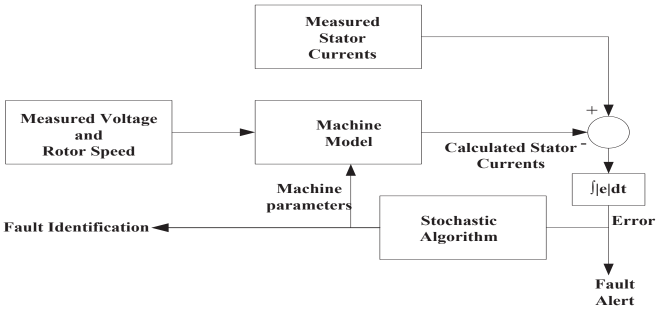

Fig. 1 Block diagram of the stochastic search based fault identification technique

II. Induction Machine Mathematical Model

The mathematical ABCabc model of an induction motor is developed using Simulink software and used with BFO to identify different machine winding faults. This ABCabc model is obtained from the standard machine voltage equations and represented by (1):

$$\begin{bmatrix} V_{sA} \\ V_{sB} \\ V_{sC} \\ V_{ra} \\ V_{rb} \\ V_{rc} \end{bmatrix} = \begin{bmatrix} R_{sA}+pL_{ss} & pM_{ss} & pM_{ss} & pM_{sr}\cos\theta_r & pM_{sr}\cos\theta_{r1} & pM_{sr}\cos\theta_{r2} \\ pM_{ss} & R_{sB}+pL_{ss} & pM_{ss} & pM_{sr}\cos\theta_{r2} & pM_{sr}\cos\theta_r & pM_{sr}\cos\theta_{r1} \\ pM_{ss} & pM_{ss} & R_{sC}+pL_{ss} & pM_{sr}\cos\theta_{r1} & pM_{sr}\cos\theta_{r2} & pM_{sr}\cos\theta_r \\ pM_{sr}\cos\theta_r & pM_{sr}\cos\theta_{r2} & pM_{sr}\cos\theta_{r1} & R_{ra}+pL_{rr} & pM_{rr} & pM_{rr} \\ pM_{sr}\cos\theta_{r1} & pM_{sr}\cos\theta_r & pM_{sr}\cos\theta_{r2} & pM_{rr} & R_{rb}+pL_{rr} & pM_{rr} \\ pM_{sr}\cos\theta_{r2} & pM_{sr}\cos\theta_{r1} & pM_{sr}\cos\theta_r & pM_{rr} & pM_{rr} & R_{rc}+pL_{rr} \end{bmatrix} \cdot \begin{bmatrix} I_{sA} \\ I_{sB} \\ I_{sC} \\ I_{ra} \\ I_{rb} \\ I_{rc} \end{bmatrix} \quad (1)$$

where (V_{sA}, V_{sB}, V_{sC}) ,(I_{sA}, I_{sB}, I_{sC}) are the stator winding voltages and currents, (V_{ra}, V_{rb}, V_{rc}), (I_{ra}, I_{rb}, I_{rc}) are the rotor winding voltages and currents, (R_{sA}, R_{sB}, R_{sC}), (R_{ra}, R_{rb}, R_{rc}) are the stator and rotor winding resistances respectively, L_{ss} and L_{rr} are the stator and rotor winding self-inductances respectively, M_{ss} and M_{rr} are the mutual inductance between pairs of stator and rotor windings respectively, M_{sr} is the peak value of the rotor position dependent mutual inductance between stator and rotor winding pairs, θ_r is the rotor position angle, $\theta_{r1}=\theta_r+2\pi/3$, $\theta_{r2}=\theta_r+4\pi/3$ and p is the differential operator.

III. Bacterial Foraging Optimization

Bacterial Foraging Optimization (BFO) was introduced in 2002 by Passino [12]. The BFO is a stochastic search and optimization technique based on the foraging behaviour of Escherichia coli (E. coli) bacteria which takes advantage of a variety of bacterial swarming and social foraging behaviours. Unlike Particle Swarm Optimization (PSO) [13] the bacterial foraging is based on the idea of the survival of the fittest. In contrast, PSO is a collective method in which members of the population cooperate to find a global optimum in a partially random way and without any selection. Members of the population with the lower fitness functions are not discarded but do survive and can potentially be the future successful members of the swarm.

The bacterial foraging system consists of four principal mechanisms, namely chemotaxis, swarming, reproduction, and elimination dispersal.

A. chemotaxis

A chemotactic step can be described as a tumble followed by another tumble or a tumble followed by a swim. The chemotactic process of E. coli is modelled within the BFO algorithm according to the possible mediums the bacteria discovers and its reaction within such mediums.
This can be simply described as follow:
If the matrix $\mathbf{X}^i(j, k, l)$ represents the current position of the i^{th} bacterium at the j^{th} chemotactic, k^{th} reproductive and l^{th} elimination-dispersal step the new position $\mathbf{X}^i(j+1, k, l)$ is then determined by:

$$\mathbf{X}^i(j+1,k,l) = \mathbf{X}^i(j,k,l) + C\boldsymbol{\lambda}(j) \quad i = 1,2,\ldots S$$

where S is the number of bacteria to be used in the search, C is the maximum step size taken to the next possible position, and the elements of the matrix $\boldsymbol{\lambda}(j)$ are random functions used to define the size of the step and the direction of movement given by:

$$\boldsymbol{\lambda}(j) = \frac{\boldsymbol{\Delta}(i)}{\sqrt{\boldsymbol{\Delta}^T(i)\boldsymbol{\Delta}(i)}}$$

$\Delta^T(i)$ is the transpose of $\Delta(i)$ and $\Delta(i) = [r_{i1},\ldots,r_{im}]$, r_i is a random number between [-1, 1] and $m = 1, 2,\ldots, p$, where p is the number of dimensions of the search space (i.e. the number of variables).

B. Swarming

Swarming is a technique used in some versions of the algorithm to smooth the progress of the convergence of cells of bacteria to form groups around areas in the solution with high nutrient concentration, thereby improving the efficiency of the search and foraging process. Swarming was not implemented in the simple form of the algorithm used in this study.

C. Reproduction

After N_c chemotactic steps, a reproduction step takes place. All bacteria are arranged in order according to their fitness, only the first half of the population survives and each surviving bacterium splits into two new bacteria, located at the same position.

D. Elimination-dispersion

The chemotaxis gives a basis for a local search, while the reproduction process speeds the convergence of the algorithm. However, chemotaxis and reproduction are not enough for global optima searching since bacteria may get stuck around the initial positions or local optima. In the BFO, a dispersion event takes place after a certain number of reproduction processes. In each elimination-dispersal step, all members are subjected to elimination-dispersal with a probability of P_{ed}. For each bacterium, if P_{ed} is greater than a random number in the interval [0–1] the bacterium is eliminated and replaced by another bacterium dispersed to a new, random location within the search space.

At the beginning of the algorithm, the E. coli are randomly distributed in the solution space, which has different concentrations of nutrients and noxious substances (different function values). The fitness function (function value or nutrient concentration value J) for each randomly distributed bacterium is then calculated at its initial location. A tumble then takes place in a random direction and the fitness value J_i corresponding to the new position is calculated. This value of $J(i, j, k, l)$ is then compared with the previously calculated value and if the new value of J is better, a swim in the same direction as the previous tumble follows. If the fitness value is less at the new position, a second tumble takes place to a new random position, and so on. The maximum number of successive steps in any one swim sequence is limited to N_s steps. The cumulative fitness function of each bacterium is calculated after N_c steps as the sum of the nutrient concentration value $\sum_{j=1}^{Nc} J(i,j,k,l)$ obtained during its life time, i.e. the previous N_c chemotactic steps. The bacteria are then arranged in order according to their fitness values. The healthier half of the population survives and the less healthy half dies out. Each surviving bacterium split into two new bacteria, located at the same position and begins the exploration of the search space from a healthier starting position than the previous generation.

Step 1: Initialize, the BFO algorithm parameters p, S, N_c, N_s, N_{re}, N_{ed}, P_{ed}, C(i), i = 1,…,S. Distribute the initial population ($X^i_m(j,k,l)|i$=1,2,…,S) randomly within the solution space.

Step 2: Elimination-dispersal loop: $l = l + 1$

Step 3: Reproduction loop: $k = k+1$,

Step 4: Chemotaxis loop: $j = j+1$,

substep a: for i = 1,2,…S, take a chemotactic step for bacterium i

substep b: calculate the fitness function, $J(i,j,k,l)$.

substep c: Let $J_{\text{last}} = J(i,j,k,l)$ to save this value since the algorithm may find a better cost via a run.

substep d: Tumble: generate a random vector $\Delta(i) \in \Re^p$ with each element $\Delta_m(i)$, m= 1,2,. . . ,p, a random number on [-1, 1].

substep e: Move: Let

$$X^i(j+1,k,l) = X^i(j,k,l) + C\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

Use the new value of $X^i(j+1,k,l)$ to calculate the concentration function of bacterium i.

substep f: compute $J(i,j+1,k,l)$, of the two feasible solutions (J^j and J^{j+1}), the one with the lowest value of J is selected.

substep g: Swim

i) Let m = 0 (counter for swim length).

ii) While m < Ns

Let m=m+1.

If $J(i,j+1,k,l) < J_{\text{last}}$, let $J_{\text{last}} = J(i,j+1,k,l)$ and let $X^i(j+1,k,l) = X^i(j,k,l) + C\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$ use the value of $X^i(j+1,k,l)$ to calculate the new $J(i,j+1,k,l)$ as in (***substep f***)

Else let m = Ns. (end while statement).

Substep h: Go to next bacterium (i + 1) if $i \neq S$ (go to ***substep b***)

Step 5: If j < Nc, go to ***step 4***. In this case, continue chemotaxis, since the life of the bacteria is not over.

Step 6: Reproduction

For the given k and l, and for each i = 1,2,…,S, let $J^i_{health} = \sum_{j=1}^{N_C+1} J(i,j,k,l)$ be the health of bacterium i (a measure of how many nutrients it got over its lifetime and how successful it was at avoiding noxious substances). Sort bacteria and chemotactic parameters C in order of ascending cost J_{health}, higher cost means lower health.

The S_r (S/2) bacteria with the highest J_{health} values die and the other S_r bacteria with the best values split (and the copies that are made are placed at the same location as their parent).

Step 7: If k < Nre, go to ***step 3***. In this case, we have not reached the number of specified reproduction steps, so we start the next generation in the chemotactic loop.

Step 8: Elimination-dispersal: For i = 1,2,…,S, with probability P_{ed}, eliminate and disperse each bacterium which keeps the swarm size constant. When eliminate a bacterium, simply disperse one to a random location within the search space.

Step 9: If l < N_{ed}, then go to ***step 2***; if not end.

The BFO parameters necessary for its implementation are first specified including the number of bacteria within the population S = 8, P_{ed} = 0.25, and C = 0.1, the initial position of each bacterium within the solution space, the number of chemotactic steps N_c = 10 taken during each bacterium lifetime, the maximum number of successive steps in any one swim sequence N_s = 4 steps and the number of reproduction N_{re} = 4 and elimination/ dispersal events N_{ed} = 2 that would occur during the BFO implementation.

IV. Experimental Results

The experiment work was conducted on a 1.5kW, 50 Hz, 240V, 2-pole wound rotor induction machine coupled to a 3kW DC machine used as a generator to provide the necessary load torque. The induction motor has a star connected stator windings and a short circuited delta connected rotor winding. Standard tests (dc resistance, no-load and locked rotor tests) [14] were carried out to determine the nominal values of the machine parameters, giving the following results in Table 1.

Tests are carried out emulating stator and rotor open-circuit winding fault conditions. In all tests, the measured waveforms are the three terminal voltages, three stator currents and rotor speed. Voltage differential probes, current probe amplifier and a digital tachometer are used to measure these signals. Data are collected over a time window of 0.2 sec, with a sampling interval of 1ms, as the machine was operating at steady state with no load. The acquired data were then processed off-line using the BFO algorithm to determine the effective resistances of the six windings. The position of each bacterium within the solution space $\mathbf{X}_i$ = (R_{sA}, R_{sB} ,R_{sC},R_{ra},R_{rb},R_{rc}) is a potential solution which can be applied to the induction motor model to evaluate a set of stator currents. Each parameter value must lie within a pre-defined search space and the overall calculation error; the Integral Absolute Error (*IAE*) as defined in (2). This error function is the cost function to be minimized by BFO.

$$IAE=\sum\left(\left|i_{sAm}-i_{sAc}\right|+\left|i_{sBm}-i_{sBc}\right|+\left|i_{sCm}-i_{sCc}\right|\right)\Delta T \qquad (2)$$

where $(i_{sAm}, i_{sBm}, i_{sCm})$ are the measured currents, $(i_{sAc}, i_{sBc}, i_{sCc})$ are the calculated currents and ΔT is the sampling period.

TABLE 1
INDUCTION MOTOR PARAMETERS

INDUCTION MOTOR PARAMETERS	Values
Stator resistances	$R_s = 5.88\ \Omega$
Rotor resistances	$R_r = 6.83\ \Omega$
Stator self-inductances	$L_{ss} = 0.729\ H$
Rotor self-inductances	$L_{rr} = 0.578\ H$
Mutual inductances between the stator windings	$M_{ss} = 0.25\ H$
Mutual inductances between the rotor windings	$M_{rr} = 0.7\ H$
Mutual inductance between stator and rotor winding pairs	$M_{sr} = 0.769\ H$ $M_{rs} = M_{sr}$

A. Stator winding open-circuit fault

A developing stator open-circuit winding fault is emulated by connecting a 7Ω resistor in series with a stator phase winding (winding B) as shown in Fig. 2.

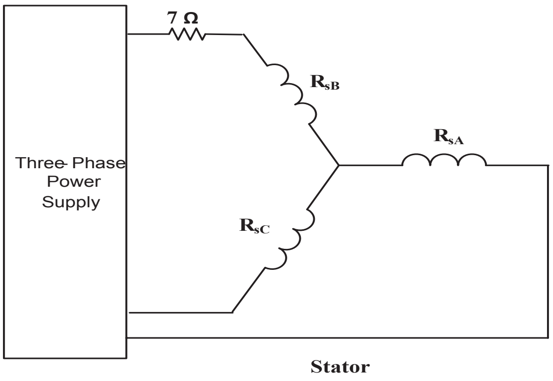

Fig. 2 Developing stator winding open-circuit fault test circuit

Results of the identification algorithm are shown in Figs. 3-5. The BFO algorithm successfully identifies the presence of the stator winding fault as indicated by the high values of R_{sB} compared with R_{sA} and R_{sC}. The number of investigations required to obtain convergence is 1844 where the calculation error falls from a maximum value of 0.068 A.s to 0.022112 A.s. Figs. 3-4 show the estimated stator and rotor resistances, respectively. The error function corresponding to the existing best solution is shown in Fig. 5. The final estimated values of the stator and rotor resistances are given in Table 2.

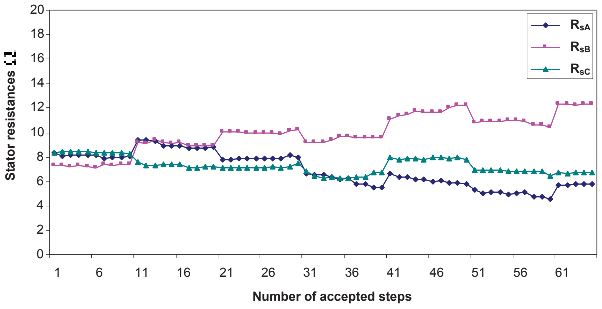

Fig.3 Stator resistance estimation using BFO for operation with stator winding fault

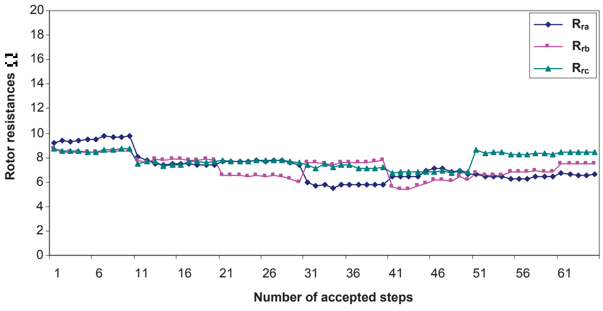

Fig. 4 Rotor resistance estimation using BFO for operation with stator winding fault

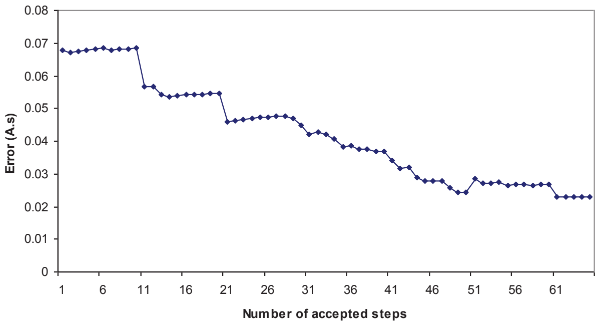

Fig. 5 Current estimation error using BFO for operation with stator winding fault

TABLE 2

FINAL VALUES OF WINDING RESISTANCES OBTAINED USING BFO WITH STATOR OPEN-CIRCUIT FAULT

R_{sA} (Ω)	R_{sB} (Ω)	R_{sC} (Ω)	R_{ra} (Ω)	R_{rb} (Ω)	R_{rc} (Ω)
5.75	12.31	6.7	6.6	7.5	8.47

B. Rotor open-circuit fault

A developing open-circuit rotor winding fault is emulated by connecting a 7 Ω resistor in series with the line connected to the two ends of the b-c rotor delta windings as shown in Fig. 6. This arrangement was used because it was not possible to gain access to the three separate delta connected windings.

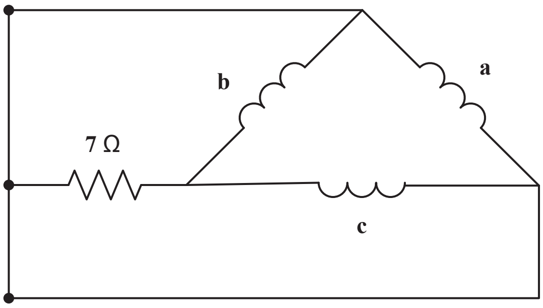

Fig. 6 Developing rotor winding open-circuit fault test circuit

The BFO algorithm is implemented to identify the presence of a developing rotor winding open-circuit fault based on the experimental measurements. In this test, the six winding resistances (R_{sA}, R_{sB}, R_{sC}, R_{ra}, R_{rb}, R_{rc}) are again the parameters to be optimized in order to minimize the *IAE* (2). Figs. 7 and 8 show the estimated stator and rotor resistances, respectively obtained by the BFO algorithm. The error function corresponding to the existing best solution is shown in Fig. 9. The number of steps or investigations required to obtain convergence of the two data sets was 1882. The calculation error falls from a maximum value of 0.068 A.s, before reducing to 0.02 A.s. Because of the simplicity of the machine model used in the investigation, it would be unrealistic to expect this error to reduce to zero, even with a much larger number of iterations. Clearly, the algorithm successfully detects the presence of the rotor winding fault as indicated by the high values of R_{rb} and R_{rc} in Fig. 8. The final estimated values of the stator and rotor resistances are given in Table 3. The final values of stator resistances are higher than the nominal values identified in Table 1 due to the fact that the algorithm is limited to changes in resistance values alone and has to find a way to compensate for the effect of the fault on other machine parameters.

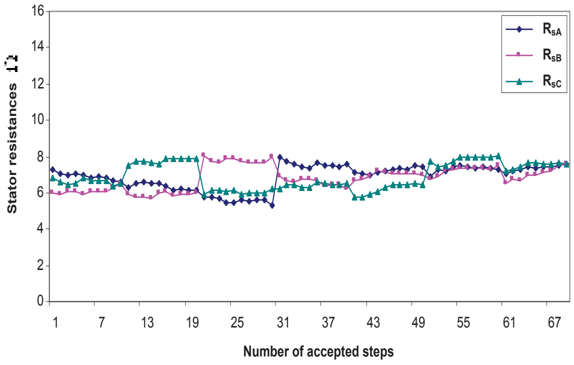

Fig. 7 Stator resistance estimation using BFO for operation with rotor winding fault

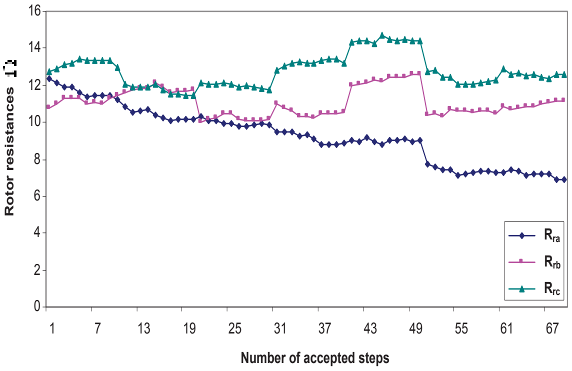

Fig. 8 Rotor resistance estimation using BFO for operation with rotor winding fault

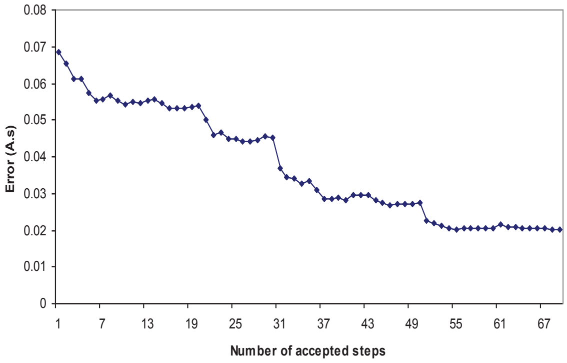

Fig. 9 Current estimation error using BFO for operation with rotor winding fault

TABLE 3
FINAL VALUES OF WINDING RESISTANCES OBTAINED USING BFO WITH ROTOR OPEN-CIRCUIT FAULT

R_{sA} (Ω)	R_{sB} (Ω)	R_{sC} (Ω)	R_{ra} (Ω)	R_{rb} (Ω)	R_{rc} (Ω)
7.58	7.624	7.61	6.88	11.1	12.534

C. Comparison with PSO algorithm

Table 4 shows a comparison of the BFO results with those obtained using the PSO algorithm [8, 9]. The BFO algorithm had a success rate of about 75% when used with the no-load measured current data compared with a success rate of about 85% for the PSO algorithm. PSO was also substantially faster than BFO which requires a much larger number of investigations to produce consistent values for the estimated rotor and stator resistances (the number of investigations when conducting a BFO search being noticeably larger than the number of accepted solutions). This demonstrates the robust nature of the PSO process and its suitability to this type of nonlinear multivariable optimization problem. Both algorithms showed estimated stator and rotor resistances to converge to similar values, confirming that there is fault in the machine's stator and rotor windings.

TABLE 4
ALGORITHM COMPARISIONS; STATOR AND ROTOR OPEN-CIRCUIT WINDING FAULTS

Stator fault				
Algorithm	Computational time (sec)	Current error (A)	Number of evaluations	Success Rate
PSO	44.8	0.01700	320	85% (120 trials)
BFO	258.16	0.022112	1844	75% (120 trials)
Rotor fault				
Algorithm	Computational time (sec)	Current error (A)	Number of evaluations	Success Rate
PSO	53.76	0.016713	384	80%(120 trials)
BFO	263.48	0.020247	1882	70%(120 trials)

V. CONCLUSION

The use of the BFO search algorithm to detect a developing induction motor winding fault has been presented in this paper. The condition monitoring method is based on the comparison of measured machine stator currents with those obtained from a machine mathematical model, and then using the stochastic search algorithm to minimise the resulting error function. BFO has been shown to be effective in determining the winding fault type and location. However results show that the PSO algorithm is better suited for this type of application, achieving a success rate of about 85% compared with 75% for BFO algorithm with noticeably improved execution times because of the smaller number of function evaluations needed for convergence.

REFERENCES

[1] P. J. Tavner, and J. Penman, “Condition monitoring of electrical machines,” Research Studies Press, 1987, Letchworth, England, 1987.

[2] P. Zhang, Y. Du, T. G. Habetler, and B. Lu, “A Survey of Condition Monitoring and Protection Methods for Medium-Voltage Induction Motors,” *IEEE Transactions on Industry Applications*, vol. 47, no. 1, pp. 34-46, Jan-Feb 2011.

[3] M. E. H. Benbouzid. “A review of induction motors signature analysis as a medium for faults detection”, *IEEE Transactions on Industrial Electronics*, vol. 47, no.5, pp. 984-992, 2000.

[4] A. J. Trzynadlowski, and E. Ritchie. “Comparative investigation of diagnostic media for induction motors: a case of rotor cage faults”, *IEEE Transactions on Industrial Electronics*, vol. 47, no.5, pp. 1092-1099, 2000.

[5] F. C. Trutt, J. Sottile, and J. L. Kohler. “Online condition monitoring of induction motors”, *IEEE Transactions on Industry Applications*, vol. 38, no.6, pp. 1627-1632, 2002.

[6] H. Douglas, P. Pillay, A. K. Ziarani. “Broken rotor bar detection in induction machines with transient operating speeds”, *IEEE Transactions on Energy Conversion*, vol. 20, no.1, pp. 135-141, 2005.

[7] M. S. N. Said, M. E. H. Benbouzid, and A. Benchaib. “Detection of broken bars in induction motors using an extended Kalman filter for rotor resistance sensorless estimation”, *IEEE Transactions on Energy Conversion*, vol. 15, no.1, pp. 66-70, 2000.

[8] S. Ethny, P. P. Acarnley, B. Zahawi and D. Giaouris, “Induction Machine Fault Identification Using Particle Swarm Algorithms,” *IEEE International Conference on Power Electronics, Drives and Energy Systems for Industrial Growth* (*PEDES 2006)*, New Delhi, India, December 2006.

[9] S. A. Ethni, B. Zahawi, D. Giaouris and P. P. Acarnley. “Comparison of Particle Swarm and Simulated Annealing Algorithms for Induction Motor Fault Identification”, *7th IEEE International Conference on Industrial Informatics (INDIN 2009),* pp. 470-474, Cardiff, UK, June 2009.

[10] M. Alamyal, S. M. Gadoue and B. Zahawi, "Detection of induction machine winding faults using genetic algorithm," *9th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives (SDEMPED 2013)*, pp.157-161, Valencia, Spain, August 2013.

[11] C. Picardi, and N. Rogano, “Parameter Identification of Induction Motor Based on Particle Swarm Optimization”, in *Proc. 2006 International Symposium on Power Electronics, Electrical Drives, Automation and Motion*, Italy, pp. 32-37, 2006.

[12] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 52-67, 2002.

[13] M. Clerc, “Particle Swarm Optimization,” ISTE, USA, 2006.

[14] *IEEE* Standard Test Procedure for polyphase Induction Motors and Generators. New York, USA, 2004.