Identification of Jiles-Atherton model parameters using Particle Swarm Optimization

Romain Marion, Riccardo Scorretti, Nicolas Siauve, Marie-Ange Raulet, Laurent Krähenbühl

To cite this version:

Identification of Jiles-Atherton model parameters using Particle Swarm Optimization

Romain Marion*, Student Member, IEEE, Riccardo Scorretti*, Nicolas Siauve*, Marie-Ange Raulet*, and Laurent Krähenbühl‡

Abstract—This paper presents the use of the multi-objective Particle Swarm Optimization technique for the identification of Jiles-Atherton model parameters. This approach, implemented for the first time in order to solve this kind of problem, is tested for two magnetic materials: NO 3% SiFe and NiFe 20-80. The results are compared with those obtained with a Direct Search Method and a GA procedure. Experimental measures performed on both samples of materials allow us to complete and argue the validation for the PSO method.

Index Terms—Magnetic hysteresis, Optimization methods, Genetic algorithms, Magnetic materials, Modeling, Magnetic field measurement.

I. INTRODUCTION

The estimation of ferromagnetic losses in electromagnetic devices by field calculations requires accurate laws for the materials. These laws must consider the dynamic effects induced in the circuits (such as eddy-currents, wall motion or pinning effect) and hysteretic phenomenon of the materials behaviour. Generally, dynamic models of materials behaviour require a static hysteresis models. Thus it is crucial to dispose of a static hysteresis model.

The description of magnetization based on Jiles-Atherton (J-A) theory [1] is often used because it can be easily implemented. Moreover the J-A model requires few memory storage, as its status is completely described by only five parameters. However, convergence problems may be encountered in the identification of these parameters by using iterative procedure [2] [3].

Recently, based on theories and algorithms of optimization, many researchers have proposed new stochastic optimization methods and “intelligent” algorithm, such as the genetic algorithm (GA) [4] [5], artificial neural network [6], chaos optimization algorithm [7], ant colony algorithm [8], line up competition algorithm [9] and various hybrid method [10] [11] [12]. However, each method has its own applicability domain and constraints. Even worst, the problem of finding the global optimal of a non linear function may be NP-complete[13].

In the case of the optimization of J-A’s parameters, GA [14] and simulated annealing method [15] have been recently introduced. Like these evolutionary computation techniques, particle swarm optimization (PSO) is a population-based search algorithm.

After a reminder of the J-A model, this paper explains the idea and the procedure of the basic PSO. Then some improvements are described (multiobjective and constrained problem, swarm mutation). Finally, an experimental validation is led with comparison between PSO and Direct Search Method (DSM) or GA. An opening on an hybrid algorithm is also discussed.

II. J-A MODEL

Let us remind the J-A model. The following form of J-A equations are considered : [16]

\[
dM = \frac{(1-c)dM_{irr}}{dH} + c\frac{dM_{an}}{dH} - \alpha(1-c)\frac{dM_{irr}}{dH}
\]

(1)

where :

- \( M_{an} \) is the anhysteretic magnetization provided by the Langevin’s equation

\[
M_{an}(H_e) = M_s(cotanh\left(\frac{H_e}{a}\right) - \frac{a}{H_e})
\]

(2)

- \( H_e \) is the Weiss’ effective field : \( H_e = H + \alpha M \)

- \( M_{irr} \) is the irreversible magnetization component defined by :

\[
\frac{dM_{irr}}{dH} = \frac{M_{an} - M_{irr}}{k\delta} \quad \text{with} \quad \delta = sign\left(\frac{dH}{dt}\right)
\]

(3)

\( \alpha, a, c, k \) and \( M_s \) are the parameters of the model where \( a \) is a form factor, \( c \) the coefficient of reversibility of the movement of the walls, \( M_s \) the saturation magnetization, \( k \) and \( \alpha \) represent the hysteresis losses and the interaction between the domains respectively.

III. BASIC PSO

A. Idea

PSO is an evolutionary computation technique developed by Kennedy and Eberhart in 1995 [17] [18]. PSO is initialized with a population of random solution called particles. Each particle is also associated with a velocity. Particles fly through the search space with velocities which are dynamically adjusted in a collaborative way. Therefore, particles have a tendency to fly towards optimal solution(s).
B. PSO Process

Each particle $i$ of the swarm is defined as a potential solution of the identification problem in a five dimensional space. This particle $i$ is associated to a position $x_i = (a_1, a_2, c_1, k_1, M_{S1})$, and has its own speed (these values are randomized initially into a defined interval).

The fitness function for a particle $i$ is defined as the squared error between the measured values and the calculated ones (obtained by considering the associated position) of a static hysteresis major loop:

$$
\text{fitness}_1 = \frac{1}{N} \sum_{i=1}^{N} \left( \frac{(B_{\text{exp}}(i) - B_{\text{sim}}(i))}{\max(B_{\text{exp}})} \right)^2
$$

where $N$, $B_{\text{sim}}$ and $B_{\text{exp}}$ represents respectively the number of points of measurement, the calculated values and the measured values.

The position with the lowest fitness score in each iteration is defined to be the entire swarm’s global best ($g_{best}$) position. In addition, each particle keeps trace of its own best position that it has visited, known as the particle’s personal best ($p_{best}$).

The particle motions are governed by the following rules which update particle positions $x_i$ with variation’s step for each parameters $v_i = (v_{a1}, v_{a2}, v_{c1}, v_{k1}, v_{M_{S1}})$:

$$
v_i^t = \omega \times v_i^{t-1} + p_1 \times r_1 \times (g_{best} - x_i^t) + p_2 \times r_2 \times (g_{best} - x_i^t)
$$

where $x_i^t$ is the current position of particle $i$, $v_i$ is the velocity of the $i$-th particle, $\omega$ is an inertia weight, $r_1$ and $r_2$ are two random numbers between 0-1 and $t$ is the current iteration. In addition, the value of the inertia weight $\omega$ in the PSO is gradually decreased in order to improve the accuracy during the final steps of optimisation:

$$
\omega = \left( \omega_{\text{start}} - \omega_{\text{end}} \right) \frac{\text{Max}_{\text{iter}} - \text{Iter}}{\text{Max}_{\text{iter}}} + \omega_{\text{end}}
$$

where $\omega_{\text{start}}$ and $\omega_{\text{end}}$ are initial and final values for the random inertia weight.

In order to avoid convergence problem, velocity are restricted to a maximum value $V_{\text{max}}$. Then, we are ensured that a maximum scope of the searching space is covered.

IV. IMPROVEMENT (PSO+)

A. Multiobjective problem

It appears that the fitness explained previously is not a sufficient criterion for any magnetic material optimization. In order to improve the convergence, we introduced another fitness function (8) which represents the area error per cycle between measurement and simulation (that is, the discrepancy between the measured and computed losses during a single cycle):

$$
\text{fitness}_2 = \frac{|\text{Area}_{\text{simu}} - \text{Area}_{\text{meas}}|}{\text{Area}_{\text{meas}}}
$$

We can define a Pareto front with these two fitnesses. However, the apparition of this front means a disappearance of the global and personal best position concept: there is an impossibility to design an only leader for the entire swarm. Therefore, we had to revise the algorithm core.

To solve this difficulty we replaced the global best position $g_{best}$ (which in the former version was unique for the whole population) with the nearest non-dominated particle, by using the following norm in the space of the fitness values:

$$
\text{Norm} = \sqrt{\Delta \text{fitness}_1^2 + \Delta \text{fitness}_2^2}
$$

In this way each particle has its own $g_{best}$, which depends on its position into the space of fitness (Fig 1).

Fig. 1. Example of the use of the multi-objective criterion. Each particle of the front has a space dominance

B. Constrained problem and swarm’s modification

In order to make easier the convergence and to eliminate non-physical solutions, the search domain has been bounded (table 1).

<table>
<thead>
<tr>
<th>TABLE I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>$\alpha$</td>
</tr>
<tr>
<td>$a$</td>
</tr>
<tr>
<td>$c$</td>
</tr>
<tr>
<td>$k$</td>
</tr>
<tr>
<td>$M_s$</td>
</tr>
</tbody>
</table>

Moreover several sets of parameters don’t produce an hysteresis curve and their fitness values are huge. So we introduce a swarm modification by deleting these "crazy" particles (if their fitness is more than $10^6 * \text{fitness}_{g_{best}}$) and replacing them by a new randomly initialized.
V. VALIDATION – DISCUSSION

As a first step of validation, we fed basic and improved PSO with artificial data generated by the J-A model. The purpose of this step is to check the capability of our PSO algorithm to retrieve (known) J-A parameters in the ideal case where provided data are perfectly consistent with the model to fit. Two different materials have been used. As comparison, two other optimization methods (DSM and GA) have been used to solve this same problem. The foundations and implementations of the DSM and GA techniques are developed in several works [5][19]. The GA method has already been implemented for the J-A parameters identification [20]. The same parameters for this method (mutation, selection and crossover probability) as then ones specified are considered. In practice, the Matlab Optimization Toolbox[21] have been used.

The improved PSO and GA methods are carried out 50 times from different initial seeds of the random number generator to ensure the repetitiveness of convergence. It has been observed that the final solutions that we obtain with these two algorithms don’t differ much (standard deviation are less than 1% of the mean value). So presented parameters are the mean of the 50 parameters. The number of individuals was set to 50. The convergence criterion is reached if one of the following criteria (10) is satisfied.

$$\sqrt{\text{fitness}_1^2 + \text{fitness}_2^2} < 10^{-3}$$

or

$$\text{IterationNumber} > 250$$

The further step has been to test the PSO with true measurements. Again, we considered two materials.

A. NO 3% SiFe material

The material sample is built of a stack of rings made of NO 3% SiFe. The static first magnetization curve and the static major loop of the sample are measured at 1 Hz. The current excitation waveform is sinusoidal. The curve used during the different optimizations is a major loop with a saturation point $H_{\text{max}} = 1500$ A/m; $B_{\text{max}} = 1.37$ T, a coercive field $H_c = 42$ A/m and remanent induction $B_r = 0.86$ T. The table II compares the values of the different parameters obtained by using both PSO and PSO+, DSM and GA algorithm. Four methods lead to close solutions. The PSO and GA methods require a similar number of iterations to converge, conversely to the DSM which needs five times more iterations. Modifications performed on PSO technique allow obtain the convergence more quickly. The accuracy of optimized parameters remains correct.

With the aim of analysing and comparing the efficiency of each method, the discrepancy between the measured datas and the calculated ones by the J-A model by considering the four sets of parameters is computed. In the table III, the error is calculated in several characteristic points: $B_{\frac{1}{2}}$ (respectively $B_{\frac{1}{2}}$) is a point on the descending part of the B-H major loop, whose H-coordinate is equal to $0.5H_{\text{max}}$ (respectively $-0.5H_{\text{max}}$) and $B_{1M}$ is a point on the first magnetization B-H curve, whose H-coordinate is equal to $0.25H_{\text{max}}$.

### TABLE II

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>PSO</th>
<th>PSO+</th>
<th>Direct Search</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_c$</td>
<td>0.3%</td>
<td>0.3%</td>
<td>0.8%</td>
<td>0.3%</td>
</tr>
<tr>
<td>$B_r$</td>
<td>0.6%</td>
<td>0.6%</td>
<td>6.2%</td>
<td>0.7%</td>
</tr>
<tr>
<td>$B_{\frac{1}{2}}$</td>
<td>0.1%</td>
<td>0.1%</td>
<td>0.52%</td>
<td>0.1%</td>
</tr>
<tr>
<td>$B_{\frac{1}{2}}$</td>
<td>0.3%</td>
<td>0.2%</td>
<td>1.8%</td>
<td>0.2%</td>
</tr>
<tr>
<td>$B_{1M}$</td>
<td>4.1%</td>
<td>3.2%</td>
<td>42%</td>
<td>3.5%</td>
</tr>
</tbody>
</table>

### TABLE III

<table>
<thead>
<tr>
<th>Parameter</th>
<th>PSO</th>
<th>PSO+</th>
<th>Direct Search</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_c$</td>
<td>6.3542e-5</td>
<td>6.8493e-5</td>
<td>8.746e-5</td>
<td>9.2367e5</td>
</tr>
<tr>
<td>$B_{\frac{1}{2}}$</td>
<td>5.6289</td>
<td>5.8289</td>
<td>6.3452e-5</td>
<td>9.2689e5</td>
</tr>
<tr>
<td>$B_{1M}$</td>
<td>9.2367e5</td>
<td>9.2689e5</td>
<td>9.2689e5</td>
<td>9.2689e5</td>
</tr>
</tbody>
</table>

### TABLE IV

<table>
<thead>
<tr>
<th>Parameter</th>
<th>PSO+</th>
<th>GA</th>
<th>Point</th>
<th>PSO+</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_c$</td>
<td>25%</td>
<td>42%</td>
<td>$B_r$</td>
<td>0.3%</td>
<td>1.3%</td>
</tr>
<tr>
<td>$B_{\frac{1}{2}}$</td>
<td>0.9%</td>
<td>1.1%</td>
<td>$B_{\frac{1}{2}}$</td>
<td>0.8%</td>
<td>0.8%</td>
</tr>
<tr>
<td>$B_{1M}$</td>
<td>6.3%</td>
<td>12.7%</td>
<td>$B_{1M}$</td>
<td>6.3%</td>
<td>12.7%</td>
</tr>
</tbody>
</table>

The high relative error obtained for $H_c$ is not relevant because the material has a very small coercivity field (less than 1 A/m).

In order to obtain more insight about the performances of our optimization methods, the estimated parameters has been used to simulate a minor loop with the J-A model. The figure 2 shows the comparison between measurements and simulations by using the sets of parameters provided by PSO+ and GA.
During this study, we noticed that PSO+ and GA suit for this kind of identification; however, with our tuning, PSO+ is more faster than GA. In fact, tuning for GA (mutation, crossover and selection probabilities) is difficult to choose, whereas PSO algorithms are generally simpler to tune. Moreover this algorithm is more easy to implement than GA.

In future work it should be possible to create an hybrid PSO-GA algorithm which uses operations of GA into PSO system. The PSO method is being implemented to optimize other kind of applications of our laboratory.

### References


