

Using Unconventional Methods to Control the Chaotic Behavior of Switched Time Systems: Application to a Stepper Motor

Y. Miladi*, M. Feki, and N. Derbel

Control and Energy Management Laboratory, National Engineering School of Sfax (ENIS), University of Sfax, Tunisia

Received 24 July 2013; Revised 30 September 2013; Accepted 5 October 2013

Abstract

In this paper we suggest a method to control the chaotic behavior of the stepper motor into a periodic one. In fact, using the supply frequency as a bifurcation parameter, we show that as the frequency is increased beyond a critical value the motor steps become irregular and even chaotic hence it becomes unpractical to be controlled in open loop mode. To circumvent the problem we propose a slight perturbation to the frequency in order to regularize the steps for high frequencies. The approach consists in using several heuristic methods such as Practical Swarm Optimization (PSO), Genetic Algorithms (GA) and Ant Colony Optimization (ACO) to obtain the optimal switching instances which define the change in the supply state (polarization). The numerical simulations performed on a stepper model show that regularization of the motor steps can be achieved for a large range of power supply frequencies ranging from quasi-periodic behavior to chaotic behavior.

Keywords: Practical Swarm Optimization, Ant Colony Optimization, Genetic Algorithms, Switched time systems, Chaos control.

1. Introduction

Switching systems are a special class of hybrid systems. A switching system is composed of several sub-systems, each subsystem is active according to a switching law. Generally, these systems are described by:

$$\dot{x} = f_{\alpha(t)}(x(t), u(t))$$

with x and u are respectively the state and input vectors and $\alpha(t)$ is the switching function that specifies the active subsystem. There are switched systems where the passage from a subsystem to another is done in a regular manner i.e. the switching cycle is repeated for each operating period, this type of switching is found in the periodic systems such as static converter. For the case where the transition from a subsystem to another is done in an arbitrary manner, the switching law evolves according to a predetermined function; we can cite for example chemical systems [1].

Control of switched systems with conventional optimization control and Lyapunov theory is the subject of several works [2, 3, 4, 5]. The synthesis of a control law in this case, requires a tedious calculation and delicate proofs. Usually, when it comes to determine simultaneously a sequence of optimal switching and optimal continuous

control, using a performance criterion, conventional optimization techniques cannot guarantee a global optimal solution.

Recently, several studies are oriented to optimization techniques based on artificial intelligence [6, 7] and heuristic search [8, 9], when it comes to the control of switching systems. In addition to their simplicity from the viewpoint of programming and implementation, these methods ensure convergence towards global solutions. Among these heuristic methods we can cite genetic algorithms, particle swarms optimization and ant colony optimization methods.

Genetic algorithms are optimization adaptive methods based on techniques derived from genetics and natural evolution. They are initially developed by John Holland in 1975 [10]. Having an initial population of individuals, these individuals undergo crossing and mutation operation, only the strongest persist in a population, thus after several generations (iterations) the population will converge to be composed of strong (best following a criterion) individuals only. Among the active domains of genetic algorithms, there is the system identification including non-linear systems, the design of fuzzy controllers and optimization of PID parameters [11, 12, 13].

PSO is an optimization algorithm invented by James Kennedy and Russell Eberhart in 1995 [14]. The origin of this method comes from observations done by simulations on schools of fish and grouped flights bird by Reynold [15,

* E-mail address: miladi.yosra@yahoo.fr

16], Heppner and Grenander [17]. It represents a method of iterative calculation. Having a population of candidate solutions, PSO algorithm searches for the optimal solution by moving the particles through their position and their velocity. The motion of each particle is conditioned by its local position and the best positions in the search space and is updated proportionally to the error with respect to global and local best positions. PSO algorithms are used in many optimization problems such as identification of nonlinear systems and training of neuronal networks [18, 19].

Optimization by ant colony is a biomimetic optimization technique originally proposed by Marco Dorigo and coworkers [20]. The principle of optimization is based on the behavior of ants looking for a food source near their colony. Initially, ants looking for food, once discovered, ants return to the nest by leaving on their way pheromone that attracts other ants. For the same source, the shortest path will be associated with more pheromone and will be covered by most of ants and the long way will disappear gradually. ACO algorithm is used in several optimization problems such as the traveling salesman problem solving [21].

In our work, we are interested in the control of the chaotic behavior of nonlinear systems. Indeed, power converters and a class of motors such as the stepper motor may exhibit irregular motion such as quasi-periodicity and chaos if the controlling signals are not well tuned [22, 23, 24]. In this paper we focus on the control of the stepper motor which has been shown to exhibit chaotic behavior if the driving signal frequency exceeds a certain threshold [25]. Therefore, the open loop driving becomes worthless and a feedback control scheme needs to be implemented. The main idea is to find the optimal switching instances to increment the motor angular position one step ahead. The optimization process will be implemented using GA, PSO and ACO and a comparison will be drawn.

This paper is organized as follows. In the second section, we present the dynamic model of the stepper motor, its chaotic behavior and we formulate the problem of control. In the third section, we will present the control algorithms (GA, PSO and ACO) that should be applied to the stepper motor. In the fourth section, we will show the effectiveness of each optimization algorithm through simulation results. The conclusion is presented in the fifth section.

2. Control Problem Formulation

The hybrid two-phased stepper motor is a common electromechanical converter widely used in robotic field and small devices positioning systems such as disk drives and X-ray scanning equipments. Originally, stepper motors were designed to provide precise discrete positioning in an open-loop control mode. However, it has been shown in [26] that using the stepper motor in an open-loop configuration gives poor performance if it is driven using higher stepping rates than advised by the constructor. Indeed, it was shown that quasi-periodic as well as chaotic behaviors appear as the power supply frequency is increased and this is due to incompatibility between the motor inertia and the driving speed. Therefore, controlling the chaotic behavior of the stepper motor becomes a worthwhile endeavor.

The stepping motor we are considering in this work is a commercial motor, the "Crouzet 82940 002". Dynamically, it is described by the following nonlinear system:

$$\begin{aligned} \frac{di_1}{dt} &= -\frac{R}{L}i_1 + \frac{K_\Phi}{L}\Omega \sin(p\theta) + \frac{1}{L}u_1(t) \\ \frac{di_2}{dt} &= -\frac{R}{L}i_2 - \frac{K_\Phi}{L}\Omega \cos(p\theta) + \frac{1}{L}u_2(t) \\ \frac{d\Omega}{dt} &= -\frac{1}{J}(C_r + f_v\Omega + K_\phi(i_1 \sin(p\theta) - i_2 \cos(p\theta)) + K_d \sin(4p\theta)) \\ \frac{d\theta}{dt} &= \Omega \end{aligned} \tag{1}$$

where i_1, i_2 are the currents of the two phases 1 and 2, Ω is the angular velocity, θ is the angular position, F_0 is the supply frequency of the two power signals u_1 and u_2 given by:

$$u_1(t) = E \text{sign}(\cos(2\pi F_0 t)) \tag{2}$$

$$u_2(t) = E \text{sign}(\sin(2\pi F_0 t)) \tag{3}$$

In the sequel, simulations are carried out using the following parameters: $R = 45 \Omega$, $L = 275 \text{ mH}$, $p = 12$, $J = 18 \times 10^{-6} \text{ kg.m}^2$, $f_v = 10^{-4} \text{ N.m.s}$, $K_\phi = 0.463$, $K_d = 16 \text{ mN.m}$ and $E = 24\text{V}$. Clearly, the motor performs 48 steps to realize a single complete turn. That is, each step corresponds to $\frac{2\pi}{48}$ rd or 7.5 degrees, and on each period

$T_0 = \frac{1}{F_0}$ the motor performs four steps or equivalently

$$\frac{2\pi}{48} = 0.5236 \text{ rd.}$$

For an operation period T_0 , u_1 and u_2 form, over time, four driving combinations: (E, E) ; $(E, -E)$; $(-E, -E)$; and $(-E, E)$ (see Fig. 1).

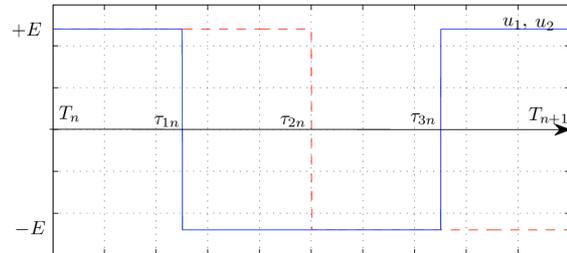


Fig. 1. Stepper motor driving signals u_1 (solid) and u_2 (dashed).

Therefore, during one driving signal period, we may consider the stepper is switching between four different non autonomous models with equal drift functions and different constant input signals. The system of equations (1) can be written as follows:

$$\dot{x} = f_\alpha(x(t))$$

with $\alpha \in \{1, 2, 3, 4\}$. By varying the frequency F_0 , we vary the toggling rate among different models.

Figure 2 depicts the bifurcation diagram of the steps of the stepper motor for a variable range of frequencies from 45Hz to 100Hz. The increase in supply frequency F_0 , beyond the critical frequency F_c leads to the appearance of

chaos in motor behavior. From Fig. 2, we notice that up to the critical frequency $F_c = 53\text{Hz}$ the motor steps are regular and four steps size is 0.5236 rd. We should mention here that according to the constructor data sheet, the maximum allowed driving frequency is 50 Hz, thus leaving a safety margin of 3 Hz.

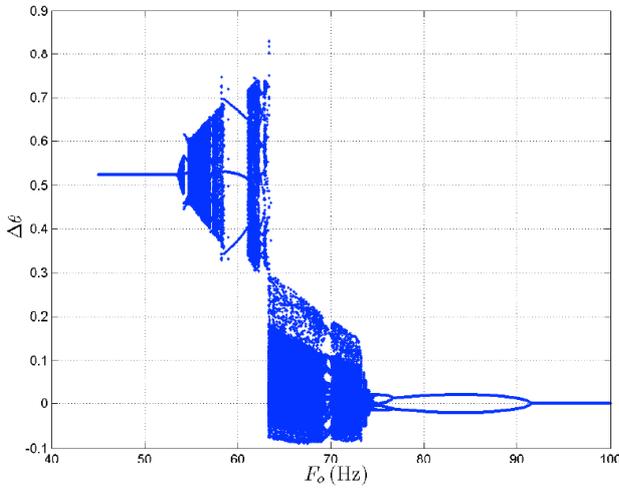


Fig. 2. Bifurcation of the step size when the frequency is varied beyond the critical value.

When we increase the frequency to 55 Hz, we obtain an irregular step size where the four steps size varies from 0.45rd to 0.61 rd. At this frequency the motor is behaving quasi-periodically which is confirmed by the closed curve attractor in the Poincaré map depicted in Fig. 3. Should we increase the frequency furthermore, to 67 Hz for instance, the motor behavior becomes chaotic with the attractor shown in Fig. 4.

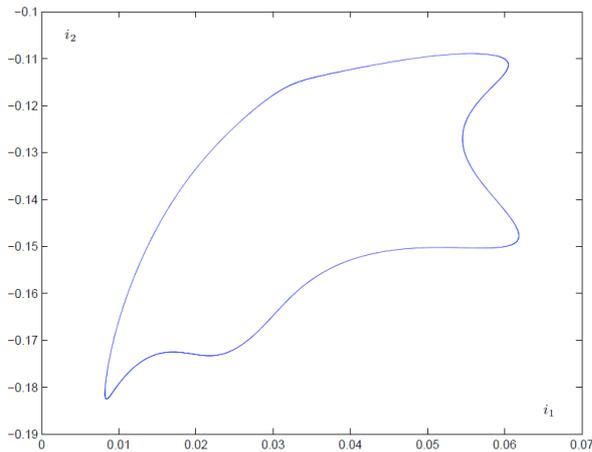


Fig. 3. Quasi-periodic Poincaré attractor of the stepper motor at 55Hz.

The aim of the control consists in slightly modifying the switching instants in order to have fixed and regular steps even for frequency range beyond the critical frequency.

Most of works which treat optimal control of switched systems, optimize all switching instants over a full period, based on a global performance criterion. In this work, we showed that it's more effective to optimize each switching time separately by considering a local performance criterion.

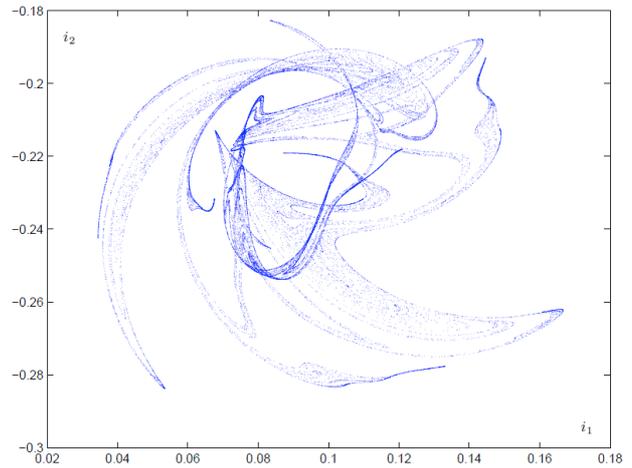


Fig. 4. Chaotic Poincaré attractor of the stepper motor at 67 Hz.

Let τ_{1n} , τ_{2n} and τ_{3n} be the switching instants at the n^{th} period $[T_n, T_{n+1}]$ of the open loop system, and let $\tilde{\tau}_{1n}$, $\tilde{\tau}_{2n}$ and $\tilde{\tau}_{3n}$ be their optimal values, then we may define the switching instants variations as:

$$\tilde{\Delta}\tau_{1n} = \tilde{\tau}_{1n} - \tau_{1n} \tag{4}$$

$$\tilde{\Delta}\tau_{2n} = \tilde{\tau}_{2n} - \tau_{2n} \tag{5}$$

$$\tilde{\Delta}\tau_{3n} = \tilde{\tau}_{3n} - \tau_{3n} \tag{6}$$

Let J_1 , J_2 and J_3 be the three local performance criterion given by:

$$J_1(\Delta\tau_{1n}) = \theta(\tau_{1n} + \Delta\tau_{1n}) - \theta(T_n) - \theta_{ref} \tag{7}$$

$$J_2(\Delta\tau_{2n}) = \theta(\tau_{2n} + \Delta\tau_{2n}) - \theta(\tau_{1n} + \tilde{\Delta}\tau_{1n}) - \theta_{ref} \tag{8}$$

$$J_3(\Delta\tau_{3n}) = \theta(\tau_{3n} + \Delta\tau_{3n}) - \theta(\tau_{2n} + \tilde{\Delta}\tau_{2n}) - \theta_{ref} \tag{9}$$

where $\theta_{ref} = \frac{2\pi}{48}$ rd. The main issue of the control is to determined for each quarter of the period and based on an optimization algorithm, the variations $\tilde{\Delta}\tau_{in}, n \in \{1, \dots, N\}, i \in \{1, 2, 3\}$, that lead to the optimum switching instant $\tilde{\tau}_{in}$ by minimizing for each operating area the performance criterion J_i . Once $\tilde{\Delta}\tau_{in}$ is obtained, it will define the switching instance for the next running subsystem and so on.

As a matter of fact, this control method requires knowledge of future values of the system state if $\Delta\tau_{in} > 0$, then a prediction step is necessary to apply unconventional optimization algorithms. Figure 5 shows the evolution curve of the stepper motor position during a period.

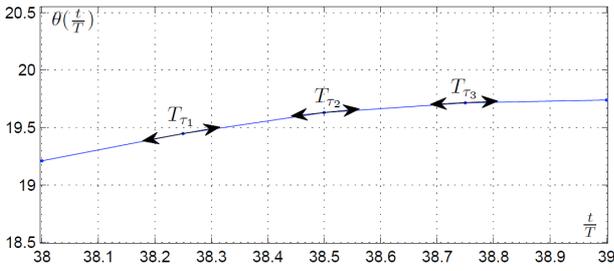


Fig. 5. Slowly varying aspect of the angular position evolution during one switching period.

Making profit of the slowly varying aspect of the angular position evolution shown on Fig. 5, we may use a linear prediction, that is the future values of positions at each instant $t = \tau_{in} + \Delta\tau_{in}$ are calculated from equation (10) (δT is an integration step size).

$$\hat{\theta}(\tau_{in} + \Delta\tau_{in}) = \theta(\tau_{in}) + \Delta\tau_{in} \frac{\theta(\tau_{in}) - \theta(\tau_{in} - \delta T)}{\delta T} \quad (10)$$

3. Switching Instances Optimization

The role of each optimization algorithm is then to find $\tilde{\Delta}\tau_{in}$. Every one has its specific org chart, but the control steps are the same.

3.1. Determining $\tilde{\Delta}\tau_{in}$ Using Genetic Algorithm

Genetic algorithms are optimization methods based on the genetic evolution process of biological organisms through the generations. According to Darwin's evolution theory, only strongest persist in a population and can reproduce and their descendants retain some of their characteristics, while the less adapted organisms disappear gradually.

Let $POP_{GA_n}, i \in \{1, 2, 3\}, n \in \{1, \dots, N\}$ be the three populations associated to the three switching times at the n^{th} period, each population consists of K individuals, each individual $I_{ink}, k \in \{1, \dots, K\}$ can be a solution of the optimization problem and let $J_{iGA}, i \in \{1, 2, 3\}$, be the three local performance criteria and $J_{iGA}(I_{ink})$ is the performance of the individual I_{ink} .

Each individual I_{ink} represents the variation $\Delta\tau_{ink}$ of the switching instants τ_{in} . The application of genetic algorithm in each area of operation will drive the initial population to converge to an optimal population that consists of K uniform individuals of value $\tilde{\Delta}\tau_{in}$.

Throughout generations, individuals are evaluated according to a fitness function. Less fit individuals will be eliminated and replaced by better fit ones according to their probability of occurrence. Next, to diversify the search within the population, offsprings are created using crossover and mutation operations.

The genetic optimization algorithm can proceed as follows:

1. Generate a random initial population of K individuals;
2. Evaluate the adaptation rate of each individual;
3. Select the set of parents considered to be best fit;

4. Cross parents with a probability P_c to generate children;
5. Mutate children with a probability P_m ;
6. Repeat steps 3, 4 and 5 until the new population contains K individuals;
7. Iterate from step 2 until the algorithm converges.

To control the stepper motor, we have applied the genetic algorithm as follows. We start by randomly creating an initial population of $K = 30$ individuals uniformly distributed over an interval of $\pm 20\%$ around the switching instant τ_{in} . Next, we evaluate the performance of each individual using:

$$J_{iGA} = J_i(\text{given in (7) - (9)}); i \in \{1, 2, 3\}$$

and we evaluate their fitness as:

$$Fitness(I_{ink}) = \frac{1}{J_{iGA}} \quad (11)$$

so that best fit individuals are selected for the next generation. The chosen selection is that of the lottery wheel, it consists in choosing individuals who are apostrophized by the best fitness. The probability of individuals' selection is proportional to their fitness and is given by:

$$P_r(I_{ink}) = \frac{Fitness(I_{ink})}{\sum_{s=1}^K Fitness(I_{ins})} \quad (12)$$

To improve the search process among the population, we make some convex crossover between several parents with a probability of $P_c = 0.8$. In addition, we also apply a mutation to a selection of children with a probability of $P_m = 0.01$. These operations are carried out over a number of generations equal to M after which we consider that the population has reached its steady status.

3.2. Determining $\tilde{\Delta}\tau_{in}$ Using Practical Swarm Optimization

PSO is an evolutionary optimization technique that found its origins from the social behavior such as the group of bees and birds in search of food. Having a population of particles dispersed in a random manner in the search space, the principle of the optimization algorithm is based on the analysis of the behavior of each particle relative thereto and to the other in order to find the optimal solution. Each particle must be able to capture the best solution visited and communicate with neighboring particles. From the information acquired, it must have the attitude to return to its best position or tend to the optimum position of other particles.

Let $POP_{PSO_n}, i \in \{1, 2, 3\}, n \in \{1, \dots, N\}$ be the three populations associated to the three switching times at the n^{th} period, each population consists of K particles, each particle $P_{ink}, k \in \{1, \dots, K\}$ is characterized by a position p_{ink} and a velocity v_{ink} , let $J_{iPSO}, i \in \{1, 2, 3\}$ be the three criteria of local performance as given by (7)-(9) and $J_{iPSO}(P_{ink})$ is the performance of particle P_{ink} . The position

of each particle p_{ink} represents a variation $\Delta\tau_{ink}$ of the switching instant τ_{in} .

During the flight of the particles, the velocity and the position of each particle is updated, by taking advantage of its local optima and other particles local optima according to the two motion equations (13) and (14).

$$v_{ink}^{j+1} = w^j v_{ink}^j + b_1(pbest_{ink}^j - p_{ink}^j) + b_2(pgbest_{ink}^j - p_{ink}^j) \quad (13)$$

$$p_{ink}^{j+1} = p_{ink}^j + v_{ink}^{j+1} \quad (14)$$

where v_{ink}^j and p_{ink}^j are respectively the velocity of the particle k and its position at the j^{th} iteration, b_1 , b_2 and w^j are weighting parameters, $pbest_{ink}^j$ is the best position reached by the k^{th} particle up to the iteration j , $pgbest_{ink}^j$ is the best position reached by the neighboring particles until the iteration j .

Then, the particle swarm will converge normally to the global optimal solution of K uniform particles value $\tilde{\Delta}\tau_{in}$.

The algorithm of particle swarm optimization can be implemented as follows:

1. Generate a random initial population of particles K with dispersed positions in the field of research;
2. Evaluate each particle according to the performance criterion J_{PSO} ;
3. Determine $pbest$ and $pgbest$;
4. Update $pbest$ and $pgbest$;
5. Update the position and the velocity of each particle using motion equations (14) and (13);
6. Iterate from step 2 until the stop condition is satisfied.

To control the stepper motor, the characteristics of the PSO algorithm are described as follows. We start by randomly creating an initial population of $K = 30$ individuals uniformly distributed over an interval of $\pm 20\%$ around the switching instant τ_{in} . Next, we evaluate the performance of each individual using:

$$J_{i_{PSO}} = J_i(\text{given in (7) - (9)}); i \in \{1, 2, 3\}$$

We update w^j using the following identity:

$$w^j = w_{\max} - \frac{w_{\max} - w_{\min}}{J_{\max}} j \quad (15)$$

with w_{\max} is the maximum value of w^j , w_{\min} is the minimum value of w^j and J_{\max} is the total number of iterations. The two weights b_1 and b_2 are random values in the intervals $(0, c_1)$ and $(0, c_2)$ respectively where c_1 and c_2 are predetermined as constants. We considered a maximum number of iteration of the PSO algorithm as stop condition.

3.3. Determining $\tilde{\Delta}\tau_{in}$ Using Ant Colony Optimization

The ant colony optimization algorithms are based on the behavior of ants when they are looking for food. Ants leave pheromone on paths of access to food, the optimization algorithm converges slowly by following the path that has

more pheromone. Indeed, an ant explores randomly the environment around its colony, once it finds a food source, it returns to the nest, leaving on its way a track of attractive pheromone, ants from nearby will tend to follow directly this way. Returning to the nest, these ants will strengthen the path; if two paths are possible to achieve the same food source, the shortest path will be traveled by more ants and so will be increasingly attractive while long path will disappear gradually. The pheromone trails evaporate at each iteration.

Let $POP_{ACO_n}, i \in \{1, 2, 3\}, n \in \{1, \dots, N\}$ be three populations of the n^{th} period, each population consists of K ants, each ant $A_{ink}, k \in \{1, K, K\}$ is characterized by a trajectory consisting of multiple nodes, each trajectory between two nodes n_p and n_q is characterized by a quantity of pheromone ph_{ink}^{pq} , let $J_{i_{ACO}}, i \in \{1, 2, 3\}$ be the three criteria of local performance.

The ant colony optimization method is usually used to solve the traveling salesman problem, thus our first concern is to model the problem of controlling the stepper motor to fit such resolution method. The stepper motor problem is constituted by three switching instants to be optimized for every period. Around each switching instant we define K nodes thus we may define K^3 paths to cross one operating period. Our aim is to find the optimum path that minimizes the given criteria. During the exploration of the environment by ants, each ant A_{ink} selects a path and leaves a pheromone proportional to the quality of the route. The K paths of the K ants will converge to the optimal path with more pheromone. The optimization algorithm by ant colony can proceed as follows:

1. Generate for each ant A a way constituted by three random nodes;
2. Evaluate the traversed path by each ant according to the performance criterion J_{ACO} ;
3. Determine the pheromone quantity for every path;
4. Update and evaporate the pheromone rate of each path;
5. Update the nodes of each path;
6. Iterate from step 2 until the algorithm converges.

For controlling the stepping motor, the characteristics of the ACO algorithm are as follows. The initial nodes are chosen in a random manner uniformly distributed over a range of $\pm 20\%$ around τ_{in} . The three performance criteria are given by:

$$J_{i_{ACO}} = J_i; i \in \{1, 2, 3\}$$

The quantity of pheromone laid by the ant A_{ink} on an arc connecting two nodes is calculated by:

$$ph_{ink}^{pq} = \frac{Q}{J_{i_{ACO}}} \quad (16)$$

where Q is an adjustment parameter. The evaporation of pheromone between two nodes nd_{in}^p and nd_{in}^q is determined by:

$$ph_{in}^{pq}(j+1) = (1 - \rho) ph_{in}^{pq}(j) \sum_{k=1}^K ph_{ink}^{pq} \quad (17)$$

where ρ is an adjustment parameter. We considered a maximum number of iteration of the ACO algorithm as stop condition.

4. Simulation Results

The three optimization algorithms were applied to the stepper motor in order to show the effectiveness of each method and its attitude to regulate the motor steps. We simulate the controlled motor along 50 periods.

The chosen control parameters are as follows:

- for Genetic Algorithm: $P_c = 0.8$, $P_m = 0.01$, $K = 30$ and iteration number $M = 50$,
- for Practical Swarm Optimization: $w_{\max} = 0.5$, $w_{\min} = 0.1$, $c_1 = 0.5$, $c_2 = 0.5$, $K = 30$ and maximum iteration number equals to $M = 50$,
- for Ant Colony Optimization: $Q = 10$, $\rho = 0.5$, $K = 30$ and maximum iteration number equals to $M = 50$.

Figures 6, 7 and 8 show the population evolutions of each optimization algorithm that specify the first, the second and the third switching instants during a given period. We note that the population of every algorithm converges to a uniform population constituted for the GA by K individuals having the same characteristics, for the PSO by K particles having the same position and for the ACO by K trajectories having the same nodes. It is worth to notice that the GA population is the slowest to converge, while the ACO is the fastest to converge. Nevertheless, ACO does not seem to converge to the optimal solution, indeed the instants variations shown on Fig. 9, 10 and 11 do not really settle at zero like for the GA and PSO methods. Finally, we should mention that after convergence, the switching instances converge to their regular positions that is:

$\tau_1 = \frac{T_0}{4}$, $\tau_2 = \frac{T_0}{2}$ and $\tau_3 = \frac{3T_0}{4}$, the shown values are normalized by the period length T_0 .

Figures 9, 10 and 11 show the optimal values evolutions of the switching instants variation for each optimization algorithm specific to the first, the second and the third switching instants along the simulated 50 periods.

Let J_{GA} , J_{PSO} and J_{ACO} be the three global criteria of the three optimization algorithms given by the three following equations:

$$J_{GA} = J_{1GA} + J_{2GA} + J_{3GA} \tag{18}$$

$$J_{PSO} = J_{1PSO} + J_{2PSO} + J_{3PSO} \tag{19}$$

$$J_{ACO} = J_{1ACO} + J_{2ACO} + J_{3ACO} \tag{20}$$

Figure 12 shows the criteria evolution of the three optimization algorithms when the frequency of the stepper motor is at $F_0 = 60 \text{ Hz}$ for one operating period. The evolution of the three performance criteria shows the effectiveness of every optimization algorithm. Eventually, to illustrate the effectiveness of the algorithms, we present the controlled motor steps and angular speed for three different frequencies. We recall that the uncontrolled motor behaves quasi-periodically at $F_0 = 55 \text{ Hz}$; on Fig. 13 and 14, we delineate respectively, the motor steps (4 steps of one driving input signal 0.5236 rd) and angular velocity

(controlled at 28.8 rd/s for 55 Hz) controlled using GA, PSO and ACO methods.

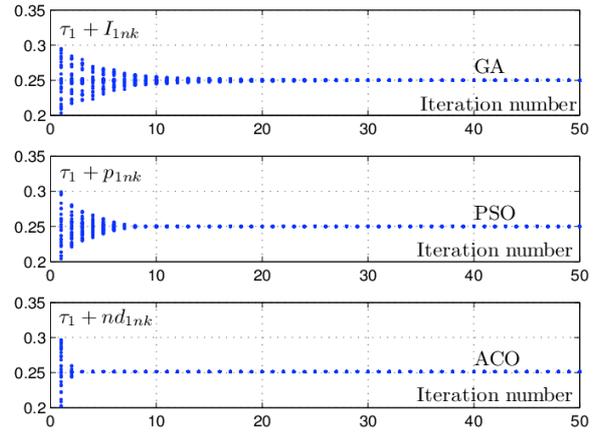


Fig. 6. Evolution of each optimization algorithm population specifying the first switching instant at the n^{th} period.

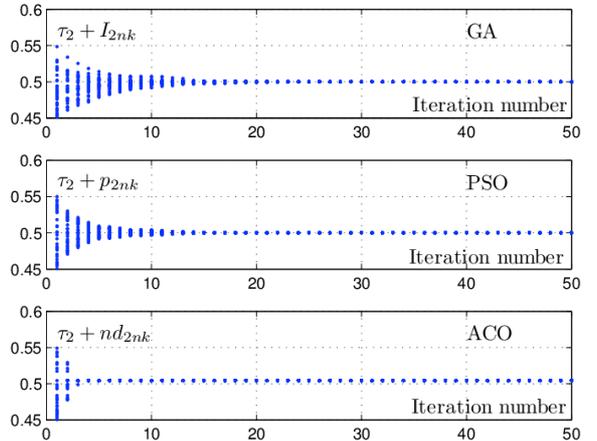


Fig. 7. Evolution of each optimization algorithm population specifying the second switching instant at the n^{th} period.

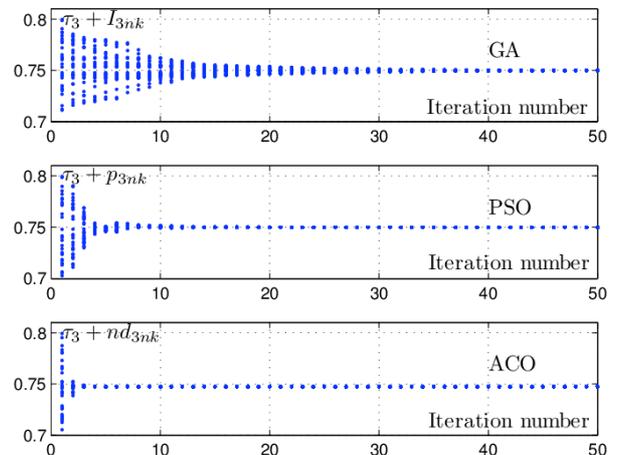


Fig. 8. Evolution of each optimization algorithm population specifying the third switching instant at the n^{th} period.

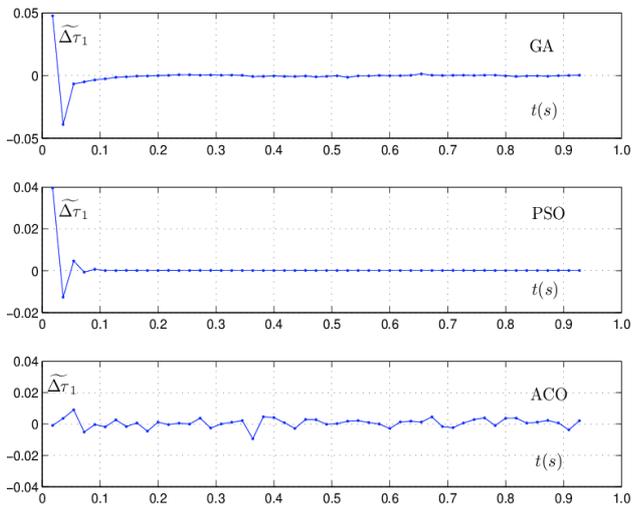


Fig. 9. The optimal values of the first switching instant along the simulated periods with $F_0 = 55\text{Hz}$.

Similarly, on Figs. 15 and 16, we delineate respectively, the motor steps (4 steps of one driving input signal 0.5236 rd) and angular velocity 31.4 rd/s with and without control and we recall that the uncontrolled motor behaves periodically with period $3T_0$ at 60 Hz. On Figs. 17 and 18, we delineate respectively, the motor steps (4 steps of one driving input signal 0.5236 rd) and angular velocity 35.1 rd/s with and without control and we recall that the uncontrolled motor behaves chaotically at 67 Hz.

Comparing all three methods reveals that the GA and PSO methods achieve a quite exact result in step size and in velocity. However, for the ACO method, although its convergence was the fastest, the achieved step size and velocity slightly wave around the expected value.

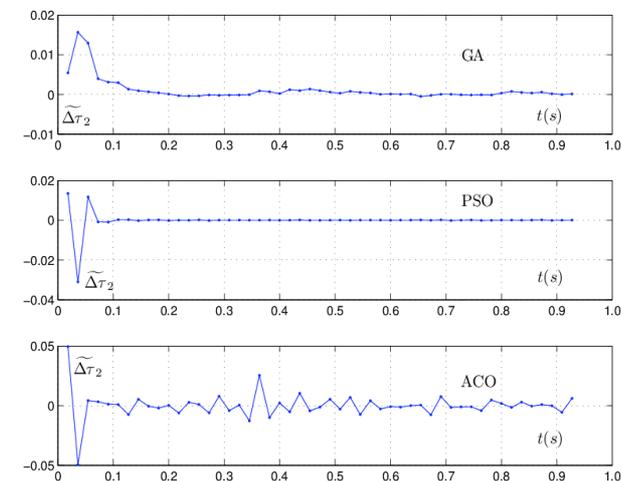


Fig. 10. The optimal values of the second switching instant along the simulated periods with $F_0 = 55\text{Hz}$.

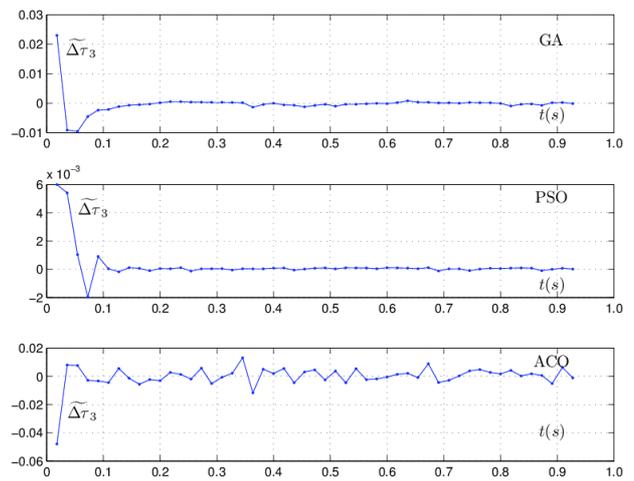


Fig. 11. The optimal values of the third switching instant along the simulated periods with $F_0 = 55\text{Hz}$.

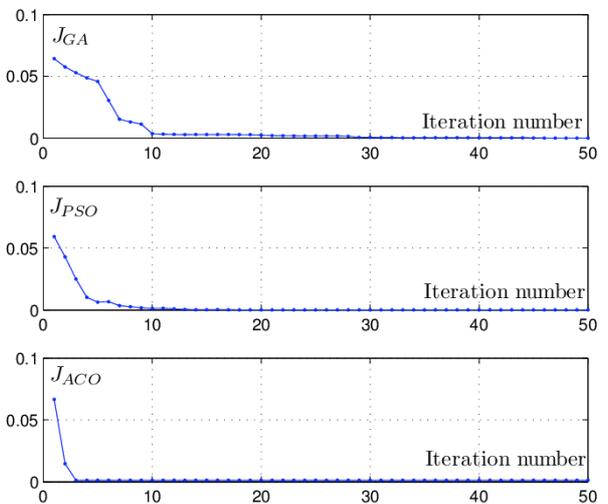


Fig. 12. Criteria evolution of the three optimization algorithms at $F_0 = 60\text{Hz}$ for one operating period.

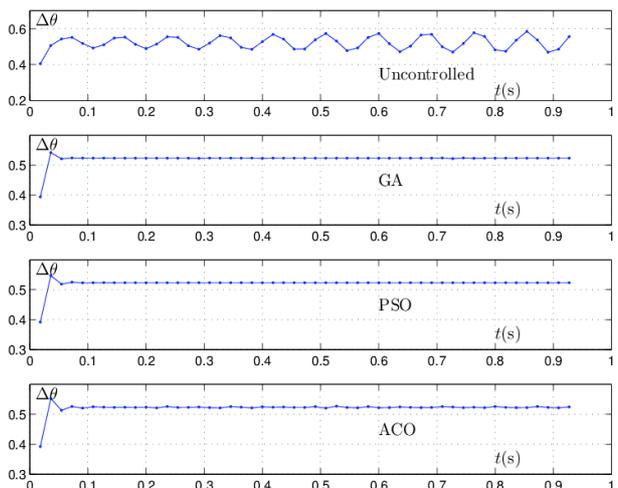


Fig. 13. Evolution of motor steps for a frequency $F_0 = 55\text{Hz}$ with and without control.

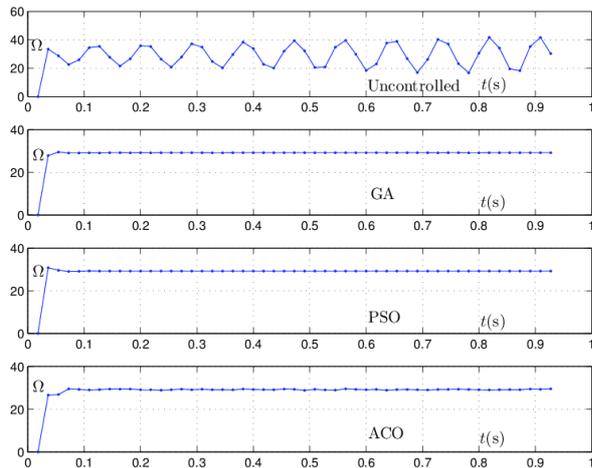


Fig. 14. Evolution of motor angular speed for a frequency $F_0 = 55$ Hz with and without control.

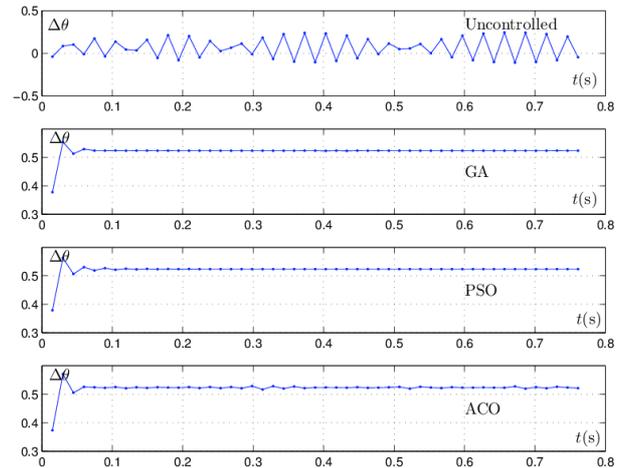


Fig. 17. Evolution of motor steps for a frequency $F_0 = 67$ Hz with and without control.

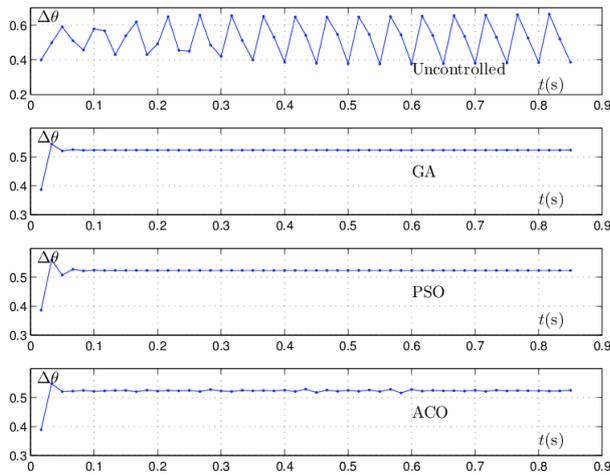


Fig. 15. Evolution of motor steps for a frequency $F_0 = 60$ Hz with and without control.

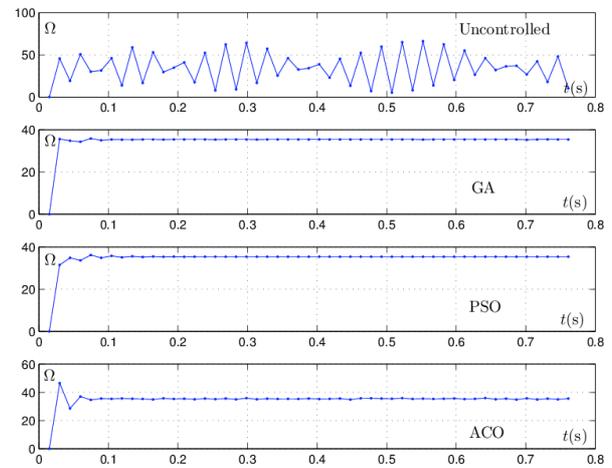


Fig. 18. Evolution of motor angular speed for a frequency $F_0 = 67$ Hz with and without control.

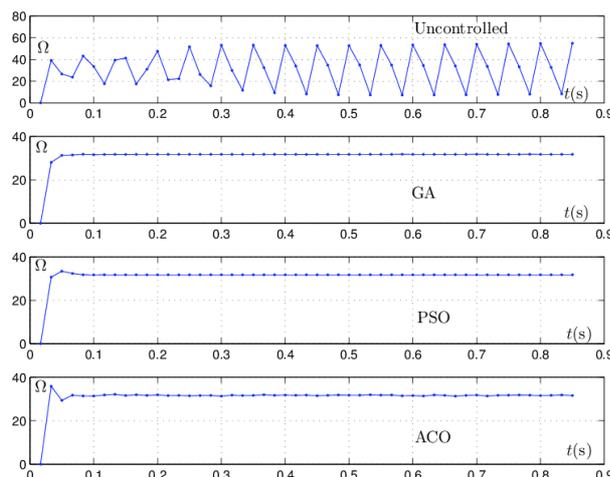


Fig. 16. Evolution of motor angular speed for a frequency $F_0 = 60$ Hz with and without control.

5. Conclusion

In this paper, three heuristic optimization algorithms GA, PSO and ACO were used to control the chaotic behavior of the stepper motor. The idea is based on slightly modifying the switching instants to maintain a constant step size and angular velocity despite the rapid rhythm imposed to the motor by the driving signal. The switching instants within an operating period, are modified each separately using an optimization method to minimize the variation of the step size. The presented simulation results showed that all three methods achieved the required results but with different performances. Indeed, the PSO method has shown the best result in terms of convergence speed and exactness. The genetic algorithm was relatively slow to converge but attained an exact result also. Finally, the ACO method which is less adapted to such problems, has reached a less exactness but in less iterations. The results were obtained on a large frequency range for which the uncontrolled stepper behaves quasi-periodically, periodically but with variable speed or even chaotically.

References

1. Z. Sun and S. S. Ge, *Stability Theory of Switched Dynamical Systems*, Springer, (2011).
2. J. Daafouz, P. Riedinger, and C. Iung, *IEEE Transactions on Automatic Control*, **47**(11), 1883 (2002).
3. C. Seatzu, D. Corona, A. Giua, and A. Bemporad, *IEEE Transactions on Automatic Control*, **51**(5), 726 (2006).
4. H. Lin and P.J. Antsaklis, *IEEE Transactions on Automatic Control*, **54**(2), 308 (2009).
5. G. Chen and Z. Xiang, *Advances in Difference Equations*, **2013**(1), 1, (2013).
6. R. Long, J. Fu, and L. Zhang, *Optimal control of switched system based on neural network optimization*. In D-S. Huang, I Wunsch, C. Donald., D. Levine, and K-H. Jo, editors, *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, **5227**, p. 799–806. Springer Berlin Heidelberg, (2008).
7. H.P. Singh and N. Sukavanam, *ISA Transactions*, **51**(1), 105 (2012).
8. N. Majdoub, A. Sakly, and M. Benrejeb, *Int. J. of Research and Reviews in Applied Sciences*, **10**(2), 200 (2012).
9. T.V. Mahendiran, K. Thanushkodi, and P. Thangam, *Journal of Computer Science*, **8**(9), 1473 (2012).
10. J. H. Holland, *Adaptation in natural and artificial systems*, The University of Michigan Press, (1975).
11. M. Gen and R. Cheng, *Genetic Algorithms and Engineering Optimization*, John Wiley & Sons, (2000).
12. Y. Miladi, M. Feki, and N. Derbel, *On the model identification of an incubator based on genetic algorithms*, In *IEEE Int. Multi-Conference on Systems, Signals and Devices (SSD'12)*, pp.1–6, Chemnitz-Germany, March (2012).
13. Y. Miladi, H. Medhaffar, M. Feki, and N. Derbel, *Controlling chaotic behavior of the stepper motor using genetic algorithms*, In *Int. Symposium on Chaos, Complexity and Leadership (ICCLS'12)*, Ankara-Turkey, Dec. (2012).
14. J. Kennedy and R. Eberhart., *Particle swarm optimization*, In *Proc IEEE Int. Conference on Neural Networks*, **4**, pp.1942–1948, (1995).
15. C. W. Reynolds, *Flocks, herds and schools: A distributed behavioral model*, In *Proc. of the 14th annual conference on Computer graphics and interactive techniques*, pp.25–34, New York, NY, USA, (1987.)
16. C. W. Reynolds., *Comput. Graph.*, **21**(4), 25 (1987).
17. F. Heppner and U. Grenander, *A stochastic nonlinear model for coordinated bird flocks*, In E. Krasner, editor, *The ubiquity of chaos*, pp.233–238. AAAS Publications, (1990).
18. V. Rashtchi, H. Shayeghi, M. Mahdavi, A. Kimiyaghalam, and E. Rahimpour, *Int. J. of Electrical and Electronics Engineering*, **2**(11), 666 (2008).
19. S.H. Ling, H.T. Nguyen, and K.Y. Chan, *A new particle swarm optimization algorithm for neural network optimization*, In *Third International Conference on Network and System Security, 2009. NSS '09.*, p. 516–521, (2009).
20. M. Dorigo and T. Stützle, *Ant Colony Optimization*, The MIT Press, (2004).
21. Z. C. Su, S. Hlaing, and M. A. Khine., *An ant colony optimization algorithm for solving traveling salesman problem*, In *Int. Conf. on Information Communication and Management*, pp.54–59, (2011).
22. M. Di Bernardo and C.K. Tse, *Chaos in power electronics: An overview*. In G. Chen and T. Ueta, editors, *Chaos in circuits and systems*, **11**, ch. 16, pp.317–340. World Scientific, New York, June (2002).
23. M. Feki, B.G.M. Robert, F. Alin, and C. Goedel, *Controlling chaotic behavior of the stepper motor.*, In *8th International Conference on Modeling and Simulation of Electric Machines, Converters and Systems, Hammamet-Tunisia, April (2005.)*
24. K. Kaoubaa, M. Feki, A. El-Aroudi, and B.G.M. Robert, *Coexistence of regular and chaotic behavior in the time-delayed feedback controlled two-cell dc/dc converter*, In *IEEE Int. Multi-Conference on Systems, Signals & Devices (SSD'09)*, pp.1–7, Jerba-Tunisia, March (2009).
25. B.G.M. Robert, M-C. Pera, and C. Goedel, *Internationale de Génie Electrique*, **3**, 375 (2000).
26. B. Robert, F. Alin, and C. Goedel. *Aperiodic and chaotic dynamics in hybrid step motor new experimental results*. In *Proc. IEEE International Symposium on Industrial Electronics.*, ISIE 2001, **3**, pp.2136–2141, (2001).