Journal of Fundamental and Applied Sciences

ISSN 1112-9867

Available online at http://www.jfas.info

Research Article

A COMBINATION OF SCALABLE ALGORITHMS FOR OPTIMISING PI CONTROLLER

K. Laroussi*[1], A. Iratni[2]

[1]Faculté des Sciences et Technologie département des Sciences et Technologie
Laboratoire d'Automatique Appliquée & Diagnostic Industriel, Université de Djelfa, Algeria

[2]Universitaire de Bordj Bou Arréridj, El-Anasser, 34265, Algeria

Received: 04 May 2017 / Accepted: 29 August 2017 / Published online: 01 September 2017

ABSTRACT

In several works using a single approach for optimization the parametres of PI controller confirms that the use of a single approach does not necessarily produce optimal results. In this paper, we propose to optimize the performance of the parametres controller by combining two scalable algorithms, genetic algorithms GA and particle Swarm PS, in order to optimize the parameters of the PI controller and to minimize the. By refining the parameters of controleur that monitor performance. Using a search engine that compares the error values of the different approaches and scenarios and, in each scenario, selects the results with the minimum error value. This method has been applied to control the speed of the induction machine. The results obtained by simulation show the high performance and robustness of this technique.

Keywords: fuzzy logic, Genetic algorithm, Induction motor, optimization, particle Swarm.

Author Correspondence, e-mail: kouider.laroussi@hotmail.com

doi: http://dx.doi.org/10.4314/jfas.v9i3.27

1. INTRODUCTION

The most commonly used regulators in industrial applications are PID type correctors because they have simple structures and good performances under certain conditions. [3] In the literature PID correctors can be divided into two parts, in the first part the parameters are chosen optimally by the known methods such as the imposition of poles, Zeigler and Nichols. The correctors are simple but their disadvantage is that they are linear and can not control systems with changes of Parameters and a large non-linearity. In the second part, the controllers have a structure identical to the PID correctors with a real-time adaptation of their parameters. These regulators are dynamic PID adaptive [4]. Today, the adjustment by the fuzzy logic with its nonlinear structure a deployment of good performance and robustness in the control of MAS., This is a new technique dealing with numerical control of process and decision price. Fuzzy logic is based on the fuzzy set theory developed by Lotfi Zadah. Besides a well-developed mathematical formalism [5], the interest of fuzzy logic control stems from the fact that the theory of fuzzy sets makes it possible to process and reason with variables that integrate the notion of imprecision, Uncertainty of subjective assessments or linguistic quantifications, which allows the controller to be designed to replace an experienced human operator [6]. The fuzzy controllers may be different as nonlinear PID or their parameters are determined in real time based on the error and its derivative, the disadvantage of FLC controllers is that they need a lot of information to compensate for nonlinearity. When changing the parameters, more and the number of FLC entries increase the size of the rule base increase [7]. The problems of improving the capabilities of the conventional PID or fuzzy corrector to best adjust a disturbed system in a disturbed universe and a very fine adjustment of the parameters of the regulator to achieve the optimal objectives are thus posed.

In order to ensure optimum regulation, meeting the requirements of the user, even in difficult and variable environments, it is necessary to develop a mechanism for adapting IP gains to incorporate a certain degree of intelligence in the regulatory strategy [7].

So, the problem of choosing a type of optimization of the gains of the PI now arises. In order to obtain responses of the acceptable output value to the desired response, the PIDs must be set and optimized. To achieve this, scalable algorithms have been widely used by several researchers to optimize and optimize systems. In most studies, the researchers used a single scalable algorithm in PID tuning and optimization. However, it is unlikely that a single scalable algorithm will find the optimal solution for all the scenarios encountered, and may even choose a locally optimal solution rather than an overall optimal solution. What makes us notice that a crucial lack of search engines to compare the results of different scalable algorithms to ensure that PID parameters are, in fact, optimal. This study attempts to minimize the uncertainty level of PID optimality by defining a search engine that includes two popular scalable $\overrightarrow{v_i} = (v_{i1}, v_{i2}, v_{i3}, , , v_{iD})$ methods. Using and comparing multiple scalable algorithms, this approach increases the likelihood of identifying truly optimal conditions and reduces the risk of selecting optimal conditions locally rather than globally optimal conditions. This dynamic tuning system simultaneously uses two evolutionary algorithms to find the optimized parameters of the PID. In addition, this study was well tested by simulation to control the speed of the induction machine under different conditions in order to test its robustness.

2. DEVELOPMENT IF THE STRATEGY

In the literature, there are several methods for optimizing controller parameters such as Ant Colony (AC), Colony of Bees (BC), Imperial Competitive Algorithm (ICA), Genetic Algorithms (GA), PSO, ANFIS, it has been demonstrated in several articles that the use of a single optimization ap-proach to the parameters does not necessarily produce op-timal results. Therefore, in this article the optimized parame-ters were found by two popular methods: PS, GA, the opti-mization learning process for PS, GA is based on the mini-mization of the local error which is controlled by the param-eters of PID.

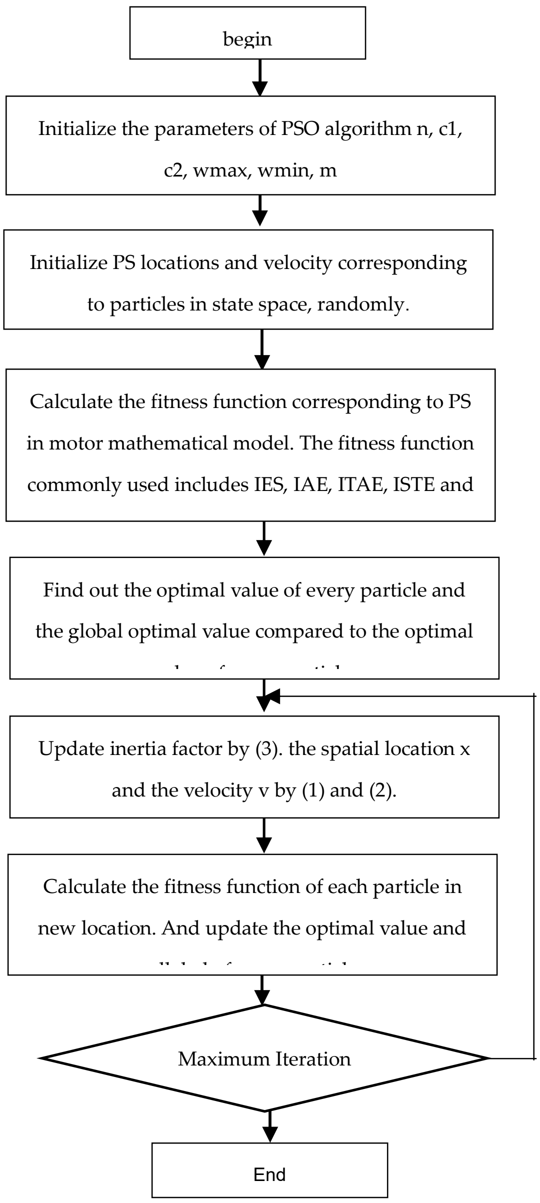


Fig.1. Organigramme de l'algorithme PSO pour le réglage du gain K p et Ki du régulateur PI

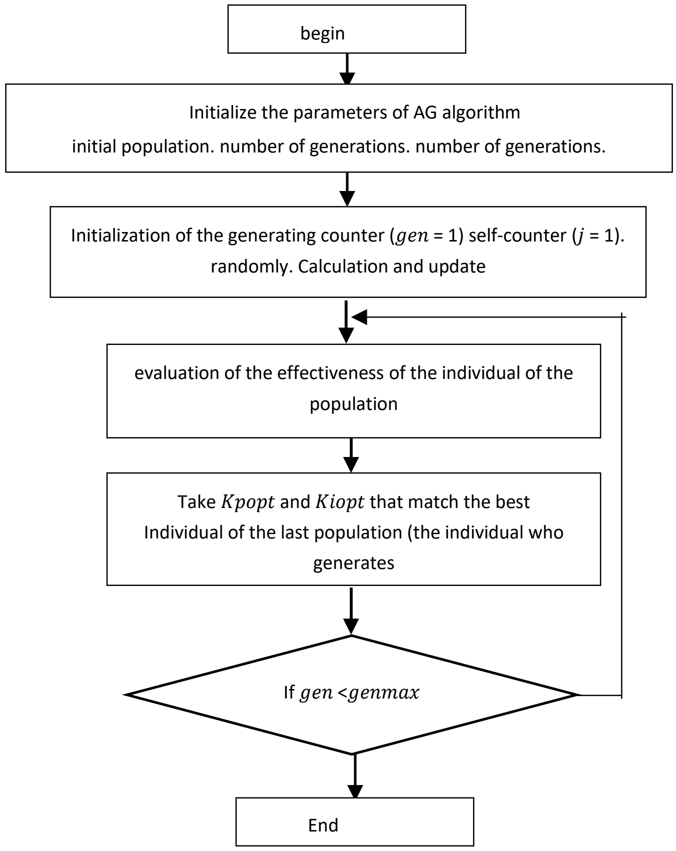


Fig.2. Organigramme de l'algorithme GA pour le réglage du gain K p et Ki du régulateur PI

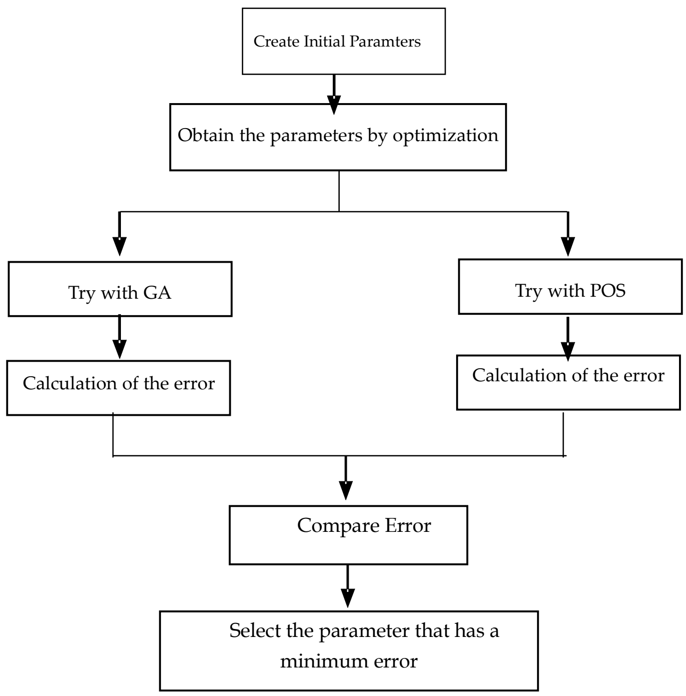


Fig.3. Schematic of the procedure for selecting the best optimization algorithm

In this study, and in order to guarantee the optimality of the parameters, we defined a unit in the PID to define the otimal parameters involved. This unit acquires the initial controller parameters which are calculated by conventional methods and in an online function these parameters are op-timized by different evolutionary methods and the optimal set of parameters of these methods has been compared and the optimal parameters of PID will be selected according to the error minimization figure 3.

2.1. Particle Swarm Optimization (PSO)

This method is inspired by the social behavior of swarming animals. The most commonly used example is the behavior of clouds of birds and schools of fish. The principle of the algorithm consists in moving these particles in the search space in order to find the optimal solution [Rio 07], Knowing that in a search space of dimension D, the particle i of the swarm is modeled by its vector position $\vec{x_i} = (x_{i1}, x_{i2}, x_{i3}, ,, x_{iD})$ and by its speed vector $\vec{v_i} = (v_{i1}, v_{i2}, v_{i3}, ,, v_{iD})$ this particle keeps in memory the best position by which it has already passed, which is noted bay $\vec{P_i} = (P_{i1best}, P_{i2best}, P_{i3best}, ,, P_{iD_{best}})$

The best position reached by all the particles of the swarm is noted $\vec{G_i} = (G_{i1best}, G_{i2best}, G_{i3best}, ,, G_{iD_{best}})$

At time t, the velocity vector is calculated from equation

$$v_{ij}(t) = w.v_{ij}(t-1) + c_1.r_1\left(Pij_{best}(t-1) - x_{ij}(t-1) + c_2.r_2\left(Gij_{best}(t-1)\right)\right) \quad (1)$$

$$j \in \{1, , D\}$$

The position at time t of the particle is then defined by equation

$$x_{ij}(t) = x_{ij}(t-1) + v_{ij}(t), \quad j \in \{1, , D\} \quad (2)$$

w is in general a constant called the coefficient of inertia, c1 and c2 are two constants called acceleration coefficients, r1 and r2 are two random numbers drawn uniformly in [0,1] at each iteration and for each dimension. $w.v_{ij}(t-1)$ Corresponds to the physical component of the displacement. The parameter w controls the influence of the direction of travel on future movement. It should be noted that, in some applications, the parameter w may be variable,

$c_1.r_1(Pij_{best}(t-1))$corresponds to the cognitive component of the displacement where c1 controls the cognitive behavior of the particle, $c_2.r_2(Gij_{best}(t-1))$ corresponds to the social component of the displacement, where c2 controls the social fitness of the particle.

The coefficient of inertia is given by [Pad 10]: $w = w_{max} - \left(\frac{w_{max} - w_{min}}{k_{max}}\right).k$

Where k_{max}, k are respectively the maximum number of iterations and the current iteration number. Wmin and wmax are respectively the minimum and maximum coefficients of inertia. Where k_{max}, k are respectively the maximum number of iterations and the current iteration number. w_{min} and w_{max} are respectively the minimum and maximum coefficients of inertia

2.2. Optimization of the PI Controller Using the PSO Method

The PI regulator is a first-order linear system with an input and an output, whose transfer function in the Laplace domain is given by equation $c(p) = k_p + \frac{k_i}{p}$ with $k_{p:}$ proportional gain, and ki: integral gain. $IAe = \int_0^{\infty} |e(t)| dt$ is the integration absolute error and the integral of the product of the error by time $ItAe = \int_0^{\infty} t.|e(t)| dt$ and the integrated of squared error is given by $Ise = \int_0^{\infty} e(t)^2 dt$ where e (t) is the error between the setpoint (desired value) and the measured value $e(t) = w^* - w$. The goal of the PSO algorithm is to minimize the objective function, which is the integral of the product of the error by time OF=$\sum ItAe$The PSO algorithm is used to determine the parameters of the PI controller (kp and ki) (based on the error of the speed where all the particles in this algorithm are decoded in two dimensions for kp and ki. For the computerization of the PSO algorithm, we have exploited the equations 1 and 2 for the development of a program under the Matlab software that can be schematized under the flowchart presented in figure 1.

2.3. Optimization of the PI Controller Using the GA Method

The second method used in this work is the genetic algorithms, these are adaptive heuristic search algorithms based on the evolutionary ideas of natural selection and genetics. Thus, they represent an intelligent exploitation of a random search used to solve optimization problems. The evolution from one generation to the next is based on the use of the three operators; Selection, crossing and mutation that are applied to all elements of the population [16]. There are three parts of this process that are randomly generated, namely the initial population, the crossing and the mutation. In the first step of the process, the initial population in the genetic algorithm generates random solutions. In the second step, the random value of the aid for crossing a new progeny and in the third step, some individuals of the generations are changed via a random mutation value [17].

The most crucial step in applying the genetic algorithm is to choose the objective functions that are used to evaluate the ability of each chromosome. Some works, such as [18] and [19], use performance indices as objective functions. Both authors use the time integral multiplied by the absolute error (ITAE), the absolute magnitude integral of the error (IAE) and Square Error Integral (ISE). Here, we use all three performance indices indicated above and integral time multiplied by the quadratic error (ITSE) in order to minimize the error signal e (t) and compare them to find the most appropriate one, where e (t) is the error signal in the time domain. If we want to minimize the setting energy, the criterion ITAE and the IAE are considered. In the event that we prefer the climb time, we take the criterion of ITSE. The steps of calculating the control law are summarized in the algorithm. Figure 2.

3. MODEL OF INDUCTION MACHINE

In this work, the system to be modeled consists of a squirrel-cage induction machine and a frequency converter, the machine is assumed to be non-saturated and the windings arranged on the machine frames create Purely sinusoidal magnetic force distributions so, we can write the equations of the machine in a reference frame related to the rotating field by

$$\frac{dX}{dt} = AX + BU \tag{3}$$

$$A=\begin{bmatrix} -(\frac{1}{T_s s}+\frac{1}{T_r}.\frac{1-s}{s}) & 0 & \frac{1-s}{s}.\frac{1}{L_m T_r} & \frac{1-s}{s}.\frac{1}{L_m}w_r \\ 0 & -(\frac{1}{T_s s}+\frac{1}{T_r}.\frac{1-s}{s}) & -\frac{1-s}{s}.\frac{1}{L_m}w_r & \frac{1-s}{s}.\frac{1}{L_m T_r} \\ \frac{L_m}{T_r} & 0 & -\frac{1}{T_r} & -w_r \\ 0 & \frac{L_m}{T_r} & w_r & -\frac{1}{T_r} \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad X = \begin{bmatrix} i_{ds} \\ i_{qs} \\ \phi_{dr} \\ \phi_{qr} \end{bmatrix} \quad U = \begin{bmatrix} v_{ds} \\ v_{qs} \end{bmatrix}$$

With $\sigma = 1 - \frac{L_m^{\ 2}}{L_s.L_r}$; T_s, T_r : Stator and rotor time constants respectively;

L_s, L_r : Stator and rotor cyclic inductances respectively; L_m : Mutual inductance

3.1 Structure of the control

Given that the inverter is controlled by current, in order to obtain the control laws, it is necessary to model the machine as if it were powered by a current, by applying the transformation of the park in a field rotation frame, the currents of the phases are known and the currents Ids and Iqs are also [3, 4, 5] the model (3) is reduced to two fields of equations (4) and the equation of motion

$$\frac{dX}{dt} = \begin{bmatrix} \frac{-1}{T_r} & w_{sl} \\ w_{sl} & \frac{-1}{T_r} \end{bmatrix} X - \frac{L_m}{T_r} \begin{bmatrix} i_{ds} \\ i_{qs} \end{bmatrix} \tag{4}$$

$$X = \begin{bmatrix} \phi_{dr} & \phi_{qr} \end{bmatrix}^t$$

$$w_{sl} = w_s - w_r$$

$$T_e = \frac{3}{2}\frac{pL_m}{L_r}\left(\phi_{dr}.i_{qs} - \phi_{qr}.i_{ds}\right)$$

$$j\frac{dw_r}{dt} = p\left(T_e - \frac{ff}{p}w_r - T_{cb}\right)$$

And by applying the principle of rotor field flux orientation to obtain decoupling between the two control variables $\varphi_{qr} = 0$, the equations of the machine (4) and (5) can be written as follows:

$$\begin{aligned} &T_r\frac{d\phi_r}{dt} + \phi_r = L_m.i_{ds} \\ &w_{sl} = \frac{L_m}{T_r}.\frac{i_{qs}}{\phi_r} = w_{sl} - p\Omega_r \end{aligned} \tag{6}$$

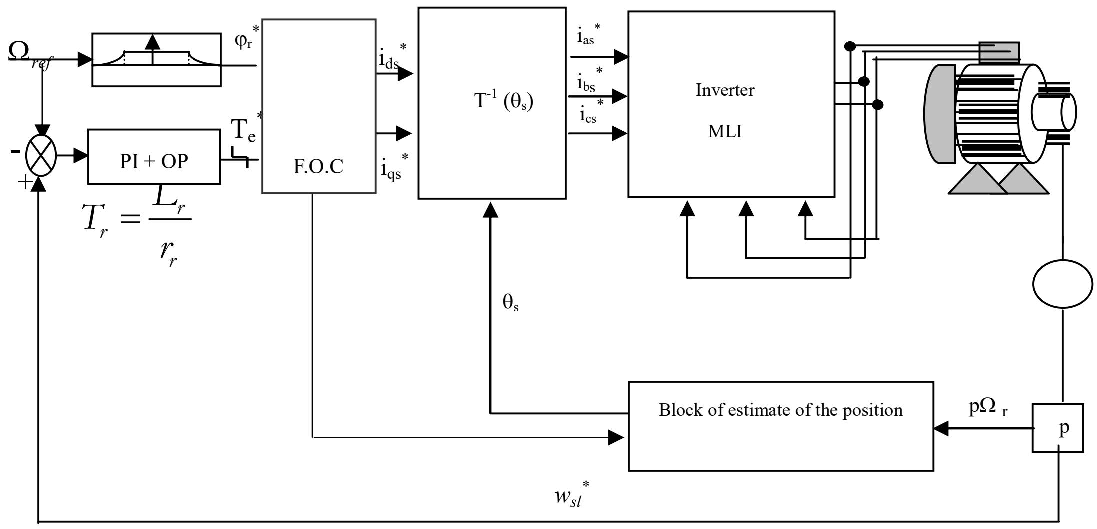


Fig.4. block diagram of indirect space control with rotor field oriented

$$j\frac{d\Omega_r}{dt} + ff\Omega_r = T_e - T_c$$

$$T_e = \frac{3}{2}\frac{pL_m}{L_r}\left(\phi_{dr}.i_{qs}\right)$$

With: rotor time constant

If we consider the couple T_e^* and the flux φ_r^* as control references, and we invert the system of equations (6) and (7), we obtain the following control equations [9]

$$w_{sl}^{\bullet} = \left(\frac{L_m}{T_r}\right).\left(\frac{i_{qs}^{\bullet}}{\phi_r^{\bullet}}\right)$$

$$i_{qs}^{\bullet} = \left(\frac{2}{3}\frac{L_r}{p.L_m}\right).\left(\frac{T_e^{\bullet}}{\phi_r^{\bullet}}\right)$$

$$i_{ds}^{\bullet} = \left(\frac{1+sT_r}{L_m}\right).\phi_r^{\bullet}$$

From these equations, we obtain the general structure of the block F.O.C (Field Orientation Control). figure 4 shows the simplified block diagram of indirect space control with rotor field oriented on the basis of equations (3). i_{ds} and i_{qs} have control of the rotor and torque field. The torque image is generated by the cruise control; A controller with a proportional integral action combined with the optimizers (PI + AO) [8, 10]. In Fig. 1, the position estimation block is used to control the orientation of the field and the block $T^{-1}(\theta_s)$ has the inverse Parck transformation.

4. OPERATION OF OPTIMIZATION OF ALGORITHMS

The figure 4 illustrates the diagram of the dynamic model of the speed control of an induction motor. The model was developed using the Simulink toolbox at Matlab. The PI parameters defined in the dynamic model were optimized by the two optimization strategies. Figure 4 illustrates the general scheme of the optimization process using the artificial intelligence methods used in this study.

The parameters optimized for the controller have been inserted into the online PI, the inputs of the optimizers are the output error and its variation. The adjustments made to the gains of the IP are intended to correct progressively the evolution of the system the regulation law; The error at a time serves to act on the adjustment of the regulator at the sampling instant according to the algorithm of the figure 3.

5. SIMULATION AND INTERPRETATION OF RESULTS

In this section, we will detail the results in simulation, the model used is that of the induction machine controlled by the rotor flux orientation described in figure 4, where the parameters of

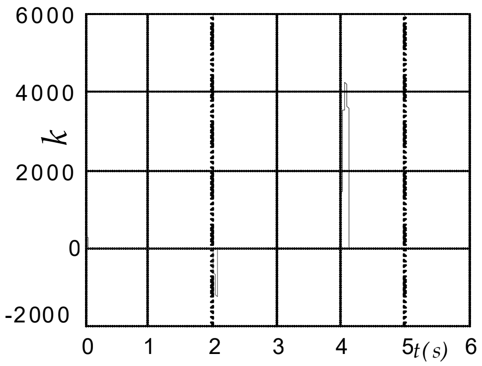


Fig.6. Adaptation of the K_i

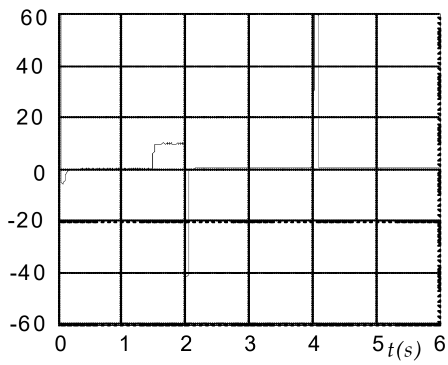


Fig.5. adaptation of the K_p

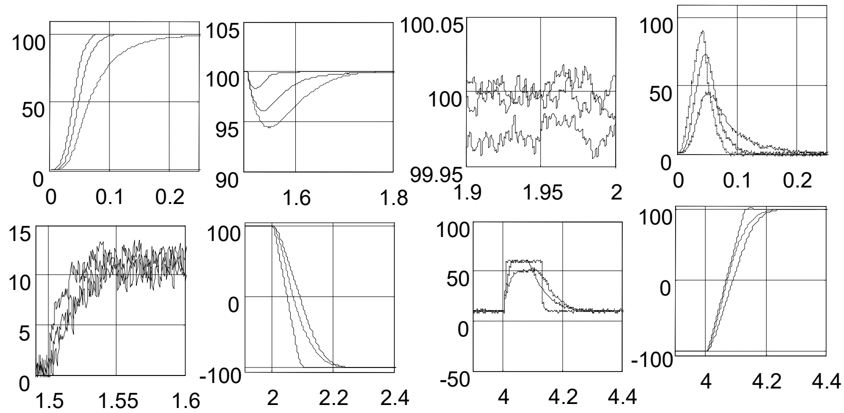
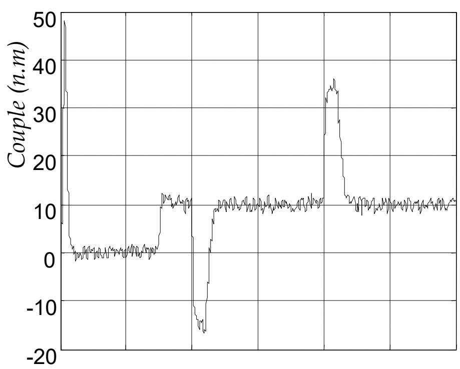
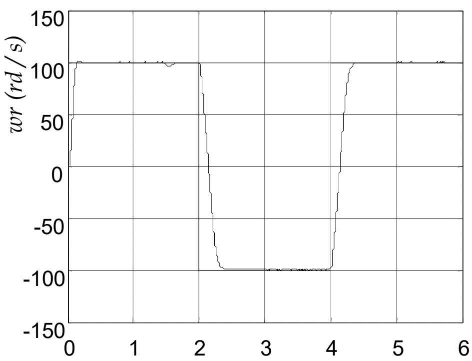


Fig.7. comparison between the different regulators

the PI regulator are optimized by the two strategists and the strategy which has a minimum error will be selected. The graphs of figure. 7, a, b, c, d show the responses of the velocity to a setpoint variation and when the load is applied, the rise and disturbance rejection time is decreased. The figure 7 clearly shows the superiority of the combination with respect to the PI optimizing by a single technique with a very fast rise time an excellent disturbance rejection and in the face of the setpoint variations, and is explained by the good optimization of the gains of the regulator PI. The figure 9 shows that the torque reaches 100 N / m at start-up and exceeds 60 N / m when the direction of rotation is reversed so that it must at all costs be limited by limiting the reverse current ιqs. Another result deserved well the shape of the variation of the adaptation coefficient and the detection of the non-linearity of the system (figure 5, 6). The combination clearly identifies areas where it is necessary to adapt the gains and the adaptation is done well as expected to increase the Ki if one is laws of the reference to increase the response time and decrease it if one is near the reference to minimize the static error, to increase the Kp if we are away from the reference to increase the response time and decrease it if we are near the reference to decrease the oscillation.





6. CONCLUSION

Compared to previous studies, the approach of this study of using the search engine to ensure the optimality of the parameters of the developed PIs makes it a very powerful system control tool whose approach can be extended to others Dynamic systems and real-time projects. The use of the search engine allowed us to have significantly accurate performance by targeting the expected acting points, making them to control the speed of the induction machine. From a broader theoretical point of view, the use of a search engine unit containing several optimization approaches has proved beneficial. No single optimization approach produced the best outputs under all conditions, so the ability to select the best outputs from several approaches improved PI performance.

7. REFERENCES

[1] J.S. Bridle, "Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition," Neurocomputing—Algorithms, Architectures and Applications, F. Fogelman-Soulie and J. Herault, eds., NATO ASI Series F68, Berlin: Springer-Verlag, pp. 227-236, 1989. (Book style with paper title and editor)

[2] Ahlawat, A. S., & Ramaswamy, A. (2004). "Multiobjective optimal fuzzy logic control system for response control of wind-excited tall buildings ". Journal of Engineering Mechanics, 130(4), 524–530.

[3] Amrani. A.N., & All Optimisation des performances d'un système de pompage photovoltaïque par neuro floue et le contrôle direct du couple Rev. Roum. Sci. Techn. – Électrotechn. et Énerg., 59, 3, p. 279–289, Bucarest, 2014

[4] Bingül, Z., & Karahan, O. (2011). A fuzzy logic controller tuned with PSO for 2 DOF robot trajectory control. Expert Systems with Applications, 38(1), 1017–1031.

[5] Bouchiba B, & All (20171). Artificial neural network sliding control for multi-machine web winding system. Rev. Roum. Sci. Techn.– Électrotechn. et Énerg. Vol. 62, 1, pp. 109–113, Bucarest, 2017

[6] Chen, J., & Rine, D. C. (2003). Training fuzzy logic controller software components by combining adaptation algorithms. Advances in Engineering Software, 34(3), 125– 137.

[7] Khan, M. W., Choudhry, M. A., Zeeshan, M., & Ali, A. (2015). Adaptive fuzzy multivariable controller design based on genetic algorithm for an air handling unit. Energy, 81, 477–488.

[8] Indranil. P et Saptarshi D. nTuning of an optimal fuzzy PID controller with stochastic algorithms for networked control systems with random time delay. ISA Transactions Volume 50, Issue 1, January 2011, Pages 28–36

[9] Laroussi. K. & Zelmat. M, Raouff. M. Implementation of a fuzzy logic system to tune a PI

controller applied to an induction motor. In advanced electrical and computer engineering volume 9 number 3-2009.

[10] Lee, C. C. (1990). Fuzzy logic in control systems: Fuzzy logic controller. IEEE Transactions on Systems, Man and Cybernetics, 20(2), 419–435.

[11] Martinez-Soto, R., Castillo, O., Aguilar, L. T., & Melin, P. (2010). Fuzzy logic controller's optimization using genetic algorithms and particle swarm optimization. In Ad- vances in soft computing (pp. 475–486). Berlin, Heidelberg: Springer.

[12] Muthukaruppan, S., & Er, M. J. (2012). A hybrid particle swarm optimization based fuzzy expert system for the diagnosis of coronary artery disease. Expert Systems with Ap- plications, 39(14), 11657–11665.

[13] Navale, R. L., & Nelson, R. M. (2012). Use of genetic algorithms and evolutionary strate- gies to develop an adaptive fuzzy logic controller for a cooling coil—Comparison of the AFLC with a standard PID controller. Energy and Buildings, 45, 169–180.

[14] Yang, Y., & Soh, C. K. (2000). Fuzzy logic integrated genetic programming for optimiza- tion and design. Journal of Computing in Civil Engineering, 14(4), 249–254.

How to cite this article:
Laroussi K, Iratni A. A combination of scalable algorithms for optimising pi controller. J. Fundam. Appl. Sci., 2017, *9(3), 1672-1684.*