Journal of Applied Science and Agriculture, 8(5) October 2013, Pages: 490-499

AENSI Journals

Journal of Applied Science and Agriculture

Journal home page: www.aensiweb.com/jasa/index.html

Designing an Optimal PID Controller based on ICA-NM Hybrid Algorithm

[1]Mehrdad Beykverdi, [2]Azar Fakharian, [3]Ahmad Ashouri

[1]*Department of Electrical Engineering, Islamshahr Branch, Islamic Azad University, Islamshahr, Tehran, Iran*
[2]*Department of Electrical Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran*
[3]*Department of Electrical Engineering, Khodabandeh Branch, Islamic Azad University, Khodabandeh, Zanjan, Iran*

ARTICLE INFO

Article history:
Received 14October 2013
Received in revised form 17 November 2013
Accepted 23 November 2013
Available online 20 December 2013

Key words:
ICA-NM Hybrid Algorithm, PID Controller, Parameter Optimization.

ABSTRACT

PID parameter optimization is an important problem in control field. In this paper, we propose a new powerful hybrid algorithm called "ICA-NM Hybrid Algorithm" to design an optimal PID controller for a sample system. The PID controller is designed in such way, minimized the sum of settling time, rise time, maximum overshoot and integral absolute error. This new algorithm is simulated with MATLAB programming. A comparison among ICA-NM Hybrid Algorithm and Imperialistic Competitive Algorithm and Genetic Algorithm is made through designing controller. The simulation result shows that the PID controller with ICA-NM Hybrid Algorithm has a fast convergence rate and better dynamic performance.

To Cite This Article: Mehrdad Beykverdi, Azar Fakharian, Ahmad Ashouri., Designing an Optimal PID Controller based on ICA-NM Hybrid Algorithm. ***J. Appl. Sci. & Agric.,*** *8(5): 490-499, 2013*

INTRODUCTION

PID controller is one of the earliest control technique that is still used widely in industrial because of their easy implementation, robust performance and being simple of physical principle of parameters. For achieving appropriate closed-loop performance, three parameters of the PID controller must be tuned (Bianchi, F.D. *et al*., 2008; Farahani M., *et al*., 2012; Padula F., A. Visioli, 2012). Tuning methods of PID parameters are classified as traditional and intelligent methods. Conventional methods such as Ziegler-Nichols and simplex methods are hard to determine optimal PID parameters and usually are not caused optimal tuning, i.e. it produces surge and big overshoot (Ziegler, J.G., 1942; Lee, K.C. *et al*., 2004). Recently, intelligent approaches such as genetic algorithm, Particle Swarm Optimization and Ant Colony optimization have been proposed for PID optimization (Shen, J.C., 2002; Herreros, A., *et al*., 2002; R. Bandyopadhyay, *et al*., 2001; H. Han, A. Luo, 2005; Y. Yang, 2006).

These algorithms such as Imperialistic Competitive Algorithm are powerful tools for determining the optimal global solution of objective function (Kim, dT.H. *et al*., 2008). Nelder-Mead simplex search and Honey Bee Algorithm are fast and optimized tools for finding the local solution of the problem (Krohling, RR.A., Rey, J.P., 2001). In the paper we propose a new hybrid algorithm called ICA-NM Hybrid Algorithm that it can determine the global and local solution of the optimizing problem simultaneously. This algorithm is a powerful tool for optimizing problems that developed for the first time.

This paper is organized as follows: Design of PID Controller in sec. 2. Fitness function is described in sec. 3. ICA algorithm and Nelder-Mead simplex search and ICA-NM Hybrid Algorithm are explained in sec. 4, 5 and 6 respectively. Sec. 7 described the simulation results and finally conclusion is presented in sec. 7.

Design of PID Controller:

One of the most common controlling devices in the market is the PID controller. The order of types of controller positioning is depicted in Fig. 1.

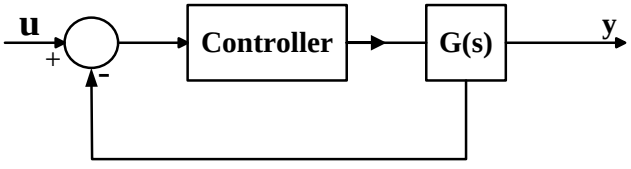

Fig. 1: PID Controller positioning in a system

Corresponding Author: Mehrdad Beykverdi, Department of Electrical Engineering, Islamshahr Branch, Islamic Azad University, Islamshahr, Tehran, Iran
E-mail: Mehrdad_b2003@yahoo.com; Tel: +98-912-4570855

Journal of Applied Science and Agriculture, 8(5) October 2013, Pages: 490-499

There are different processes for different composition of proportional, integral and differential. The duty of control engineering is to adjust the coefficient of gain to attain the error reduction and fast dynamic responses simultaneously. The transfer function of PID controlling is defined as follows:

$$G_{PID}(s)=Kp+\frac{Ki}{s}+KD.S=\frac{KD.S^2+Kp.S+Ki}{S} \quad (1)$$

PID controller is a linear control methodology with a very simple control structure. PID controllers operate directly on the error signal, which is the difference between the desired output and the actual output, and generates the actuation signal that drives the plant. These types of controllers have three basic terms: proportional action, in which the actuation signal is proportional to the error signal, integral action, where the actuation signal is proportional to the time integral of the error signal, and derivative action, where the actuation signal is proportional to the time derivative of the error signal. In PID controller design, k_p , k_I and k_D related to the closed loop feedback system within the least time is determined and requires a long range of trial and error.

To design a particular control loop, the values of three constant (k_p , k_I and k_D) have to be adjusted so that the control input provides acceptable performance from the plant. In order to get a first approach to an acceptable solution, there are several controllers design method that can be applied. For example classical control methods in the frequency domain or automatic methods like Ziegler-Nichols, which the most is well known of all tuning PID methodologies. Although these methods provide a first approximation, the response produced usually needs further manual retuning by the designer before implementation.

Rise Time:

The rise time is defines as the time required for the step response to rise from 10 to 90 percent of its final value.

Settling Time:

The settling time is defined as the time required for the step response to stay within 2 percentage of its final value.

Max Overshoot:

If y_{max} denotes the maximum value of y and y_{ss} represent the steady state value of it, the maximum overshoot will be defined as:

$$MaxOvershoot = ymax - yss \quad (2)$$

Integral Absolute Error:

The integral absolute error (IAE) is defined as:

$$IAE = \int_0^\infty e(t)dt \quad (3)$$

In our implementation we calculate the integral up to 3×settling time, which is an acceptable approximation of the real value of IAE.

To design an optimal PID controller and making a comparison we use the total cost function:

$$f_{Total} = f_{MO} + f_{RT} + f_{ST} + f_{IAE} \quad (4)$$

f_{MO}, f_{RT}, f_{ST} and f_{IAE} are the maximum overshoot, rise time, settling time and integral absolute error respectively.

Fitness Function:

The fitness function is important to be properly defined. As usual, control tuning has to achieve different types of specification, such as:

1) Obtaining dynamic performances evaluated with:

a) Minimization of a performances index such as IAE (integral of absolute value error).

b) Adjusting time specifications.

2) Obtaining robustness properties:

a) Model error robustness.

b) Input noise robustness.

Integral error is usually use as performance index of PID system parameter tuning, while ITAE is often used in optimal analysis and design. In this study fitness function is defined as follows:

Journal of Applied Science and Agriculture, 8(5) October 2013, Pages: 490-499

$$F = \int_0^{\infty} t|e(t)|dt \quad (5)$$

Where F and e(t) are fitness function and error respectively.

Imperialistic Competitive Algorithm:

Atashpaz Gargari and Lucas introduced Imperialistic Competitive Algorithm that is proposed the evolutionary process to optimize which is illustrated by imperialistic competition. Optimization is a process of making something better. We want to find the argoman *x* in the way its analogous cost be optimum, which having function in optimization.

In this algorithm all the countries are divided into two groups: imperialist and colonies. Imperialistic competition is the main part of this algorithm and causes the colonies to converge to the global minimum of the cost function. In continue it will be described how the imperialistic competition is modeled and implemented among empires.

Imperialist Competition Strategy:

Like other evolutionary algorithms, ICA starts with an initial population. Some of the best countries in the population are selected to be the imperialist and rests are colonies. All the colonies of initial population are divided among the mentioned imperialists based on their power. After dividing all colonies among imperialists, these colonies start moving toward their relevant imperialist countries. The total power of an empire depends on both the power of the imperialist country and the power of its colonies. We will model this fact by defining the total power of an empire by the power of imperialist country plus a percentage of mean power of its colonies.

The imperialistic competition begins between all the empires. Any empire that is not able to succeed in the competition and can't increase its power (or at least prevent decreasing its power) will be eliminated from the competition. The imperialistic competition will gradually result in an increase in the power of powerful empires and a lost in their power and ultimately they will collapse. The movement of colonies to their relevant imperialists along with competition among empires and also the collapse mechanism will hopefully cause all the countries to converge to a state in which there exist just one empire in the world and all the other countries are colonies of that empire. In this ideal new world colonies, have the same position and power as the imperialist.

Generating Initial Empires:

The goal of optimization is to find an optimal solution in terms of the variables of the problems. We form an array of variable values to be optimized. In GA terminology, this array is called "Chromosome" and in PSO, it's called "Particle", but here the term "Country" is used for this array. In an N_{var} dimension optimization problem, a country is a $1*N_{var}$ array. This array is defined by:

$$Country = [p1, p2, p3, \ldots., pNvar] \quad (6)$$

The variable values in the country are represented as floating point numbers. The cost of a country is found by evaluating the cost function *f* at the variables (p_1,p_2,p_3,….,p_{Nvar}). Then:

$$Cost = f\,(country) = f\,(p1, p2, p3, \ldots., pNvar) \quad (7)$$

At the first of optimization, we generate the initial population of size N_{pop}. We select N_{imp} of the most powerful countries to form the empires. The remaining N_{col} of the population will be the colonies each of which belongs to an empire. Then we have two types of countries; imperialist and colony.

To create the initial empires, we divide the colonies between imperialists based on their power. To divide the colonies among imperialists proportionally, we define the normalized cost of an imperialist by:

$$Cn = cn - maxi\{ci\} \quad (8)$$

Where c_n is the cost of nth imperialist and C_n is its normalized cost. Having the normalized cost of all imperialists, the normalized power of each imperialist is defined by:

$$p_n = \left| \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \right| \quad (9)$$

Journal of Applied Science and Agriculture, 8(5) October 2013, Pages: 490-499

The initial number of colonies of an empire will be:

$$N.C.n = round\{pn.Ncol\} \quad (10)$$

Where N. C.$_n$ is the initial number of colonies of nth empire and N_{col} is the number of all colonies. To divide the colonies for each imperialist we randomly choose N. C.$_n$ of the colonies and give them to it. These colonies along with the imperialist will form nth empire. Figure 2 shows the initial population of each empire. As shown in this figure, bigger empires have greater number of colonies while weaker ones have less. In this figure imperialist 1 has formed the most powerful empire and has the greatest number of colonies.

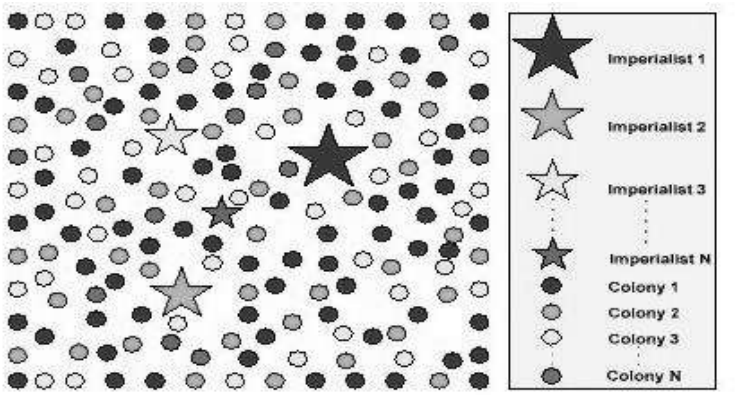

Fig. 2: Initialize the empires: The more colonies an imperialist possess, the bigger one is its relevant Star mark.

Moving the Colonies of an empire toward the Imperialist:

Imperialist countries started to improve their colonies. We have modeled this fact by moving all the colonies toward the imperialist. This movement is shown in figure 3 which the colony moves toward the imperialist by *x* units. The new position of colony is shown in a darker color. The direction of the movement is the vector from colony to imperialist. In this figure *x* is a random variable with uniform (or any proper) distribution. Then for *x* we have:

$$x \sim U\ (0\ ,\beta \times d\ ) \quad (11)$$

Where β is a number greater than 1 and *d* is the distance between colony and imperialist. A β›1 causes the colonies to get closer to the imperialist state from both sides.

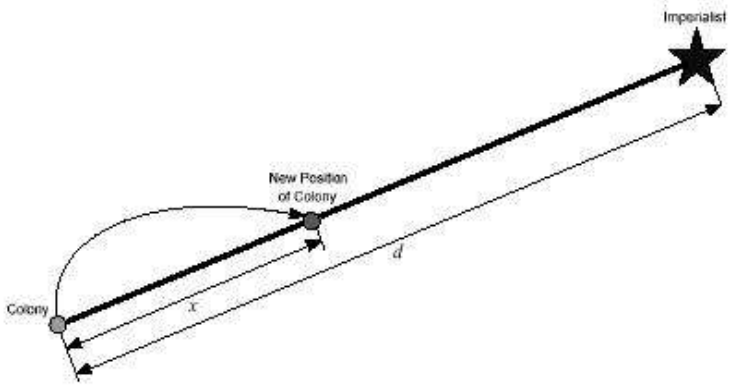

Fig. 3: Moving colonies toward their relevant imperialist

To search different points around the imperialist we add a random amount of deviation to the direction of the movement. Figure 4 shows the new direction. In this figure θ is a random number with uniform (or any proper) distribution. Then:

$$\theta \sim U\ (-\gamma\ ,\gamma) \quad (12)$$

γ is a parameter that adjusts deviation from the original direction.

Journal of Applied Science and Agriculture, 8(5) October 2013, Pages: 490-499

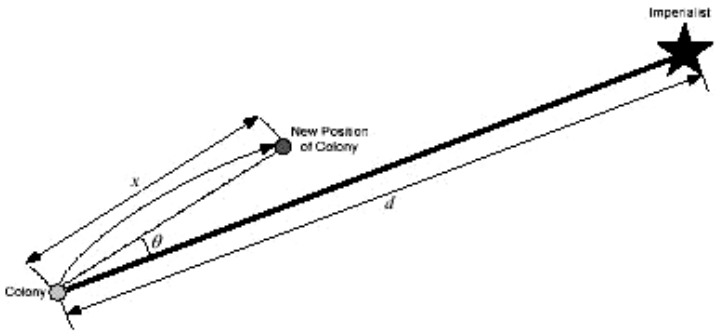

Fig. 4: Moving colonies toward their relevant imperialist in a randomly deviated direction

Exchanging positions of the Imperialist and Colony:

While moving toward the imperialist, a colony may reach to a position with lower cost than that of imperialist. In such a case, the imperialist moves to the position of that colony and vice versa. Then algorithm will continue by the imperialist in a new position and then colonies start moving toward this position. Figure 5a illustrates the position exchange between a colony and the imperialist. In this figure the best colony of the empire is shown in a darker color. This colony has a lower cost than that of imperialist. Figure 5b shows the whole empire after exchanging the position of the imperialist and that colony.

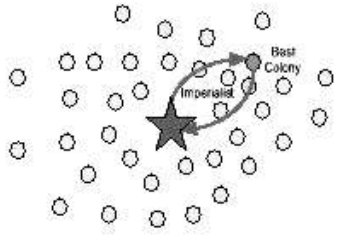

Fig. 5a: Exchanging the positions of a colony and the imperialist

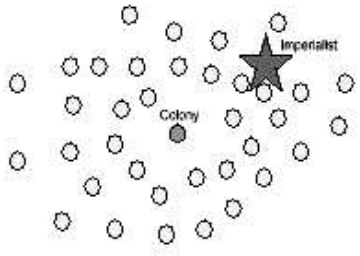

Fig. 5b: Main imperialist after position exchange

Total Power an Empire:

Total power of an empire is mainly affected by the power of imperialist country. But the power of the colonies of an empire has an effect, albeit negligible, on the power of that empire. We have modeled this fact by defining the total cost by:

$$\mathrm{T.C.n} = \mathrm{cost(imperialist)} + \zeta\,\mathrm{mean\{Cost(colonies\ of\ empiren)\}} \quad (13)$$

Where $T.C._n$ is the total cost of the nth empire and ζ is a positive number which is considered to be less than 1. A little value for ζ causes the total power of the empire to be determined by just the imperialist and increasing it will increase the role of the colonies in determining the total power of an empire.

Imperialist competition:

This competition is modeled by picking some of the weakest colonies of the weakest empires and making a competition between all empires to possess these colonies. Figure 6 shows the modeled imperialist competition each of empires will have a likelihood of taking possession of the mentioned colonies. In other words these colonies will not be possessed by the most powerful empires, but these empires will be more likely to possess them.

Journal of Applied Science and Agriculture, 8(5) October 2013, Pages: 490-499

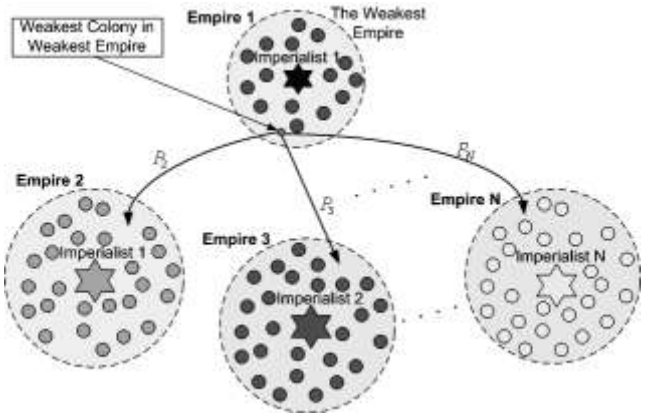

Fig. 6: Imperialist Competition

To start the competition, first we find the possession probability of each empire based on its total power. The normalized total cost is simply obtained by:

$$N.T.Cn. = T.Cn. - maxi\{T.Ci\} \tag{14}$$

Where $T.C_n.$ and $N.T.C_n.$ are total cost and normalized total cost of nth empire respectively. Having the normalized total cost, the possession probability of each empire is given by:

$$p_{p_n} = \left| \frac{N.T.C._n}{\sum_{j=1}^{N_{imp}} N.T.C._j} \right| \tag{15}$$

To divide the mentioned colonies among the empires based on the possession probability of them, we form the vector P as:

$$P = [pp1, pp2, pp3, \ldots., pNimp] \tag{16}$$

Then we create a vector with the same size as P whose elements are uniformly distributed random numbers.

$$R=[r_1,r_2,r_3,\ldots.,r_{Nimp}];\ r_1,r_2,r_3,\ldots.,r_{Nimp} \sim U(0,1) \tag{17}$$

Then we form the vector D by simply subtracting R from P:

$$D = P - R = [D1, D2, D3, \ldots.., DNimp] = [pp1 - r1, pp2 - r2, \ldots.., ppNimp - rNimp] \tag{18}$$

Referring to vector D we will hand the mentioned colonies to an empire whose relevant index in D is maximum.

Ignoring the Weak Empires:

Weak empires will eliminate in the imperialistic competition and their colonies will be divided between other empires. In this algorithm, it's assumed that an empire will be removed it when it loses all of its colonies.

Convergence:

ICA will run until all the empires except the most powerful one will destroy and all the colonies will be under the rule of this unique empire. In this ideal new world all the colonies will have the same position and same costs and they will be controlled by an empire with the same position and cost as themselves (Atashpaz-Gargari, A.E., C. Lucas, 2007).

Nelder-Mead Simplex Search:

Nelder-Mead simplex algorithm is a classical powerful local search, making no use of the objective function derivatives. A simplex is a geometrical figure consisting, in *n* dimensions, of (n+1) points $s_0, \ldots, s_n$. If any point of a simplex is taken as the origin, the *n* other points define vector directions that span the n-

Journal of Applied Science and Agriculture, 8(5) October 2013, Pages: 490-499

dimension vector space. If we randomly draw as initial starting point s_0, then we generate the other *n* points s_i according to the below relation:

$$Si = s0 + \lambda ej \tag{19}$$

Where the e_j are n unit vectors and λ is a constant which is typically equal to one.

Through a sequence of elementary geometric transformations (reflection, contraction, expansion and multi-contraction), the initial simplex moves, expands or contracts. To select the suitable transformation, the method only uses the values of the function to be optimized at the vertices of the simplex considered. After each transformation the current worst vertex is replaced by a better one. Trial moves shown in figure 7 are generated according to the following basic operations:

$$\bar{s} = (1/n) \sum_{i=1}^{n} si \tag{20}$$

Reflection: $$sr = (1 + \alpha)\, \bar{s} - \alpha Sn + 1 \tag{21}$$

Expansion: $$sc = \gamma\, sr + (1 - \gamma)\, \bar{s} \tag{22}$$

Contraction: $$sc = \beta sn + 1 + (1 - \beta)\, \bar{s} \tag{23}$$

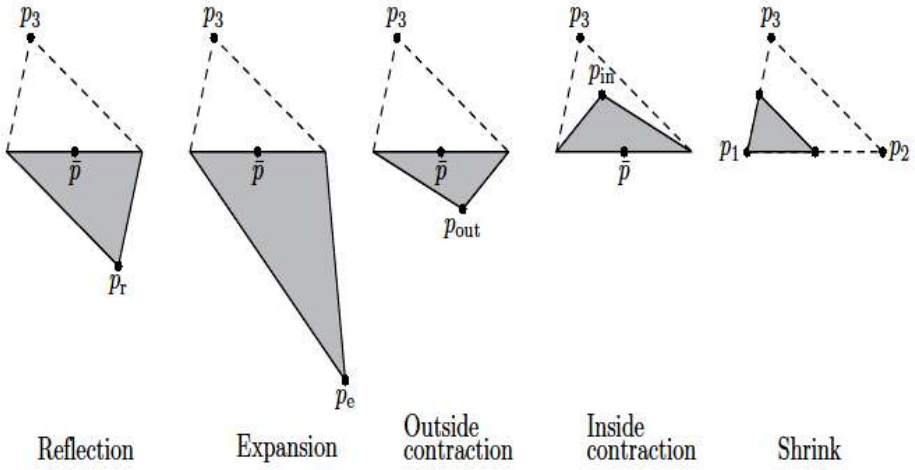

Fig. 7 Available moves in the Nelder-Mead simplex method

In initial of the algorithm, one moves only the point of the simplex, where the objective function is worst, and one generates another point image of the worst point. This operation is the reflection. If the reflected point is better than all other points, the method expands the simplex in this direction, otherwise, if it is at least better than the worst, the algorithms performs again the reflection with the new worst point. The contraction step is performed when the worst point is at least as good as reflected point, in such a way that the simplex adapts itself to the function landscape and finally surrounds the optimum. If the worst point is better than the contracted point, the multi-contraction is performed. At each step we check that the generated point is not outside the allowed reduced solution space (Nelder, J.A., R. Mead, 1965).

ICA-NM Hybrid Algorithm:

This paper proposes a new method to determine the optimal parameters of PID controllers by ICA-NM hybrid algorithm. The goal of integrating Nelder-Mead simplex search method and Imperialistic Competitive Algorithm is to combine their advantages and avoid disadvantages. The NM method is a powerful local search, while ICA is a strong global search. To use the advantages of these algorithms and ignoring disadvantages, we hybridize them. Figure 8 shows the flowchart ICA-NM hybrid algorithm.

Journal of Applied Science and Agriculture, 8(5) October 2013, Pages: 490-499

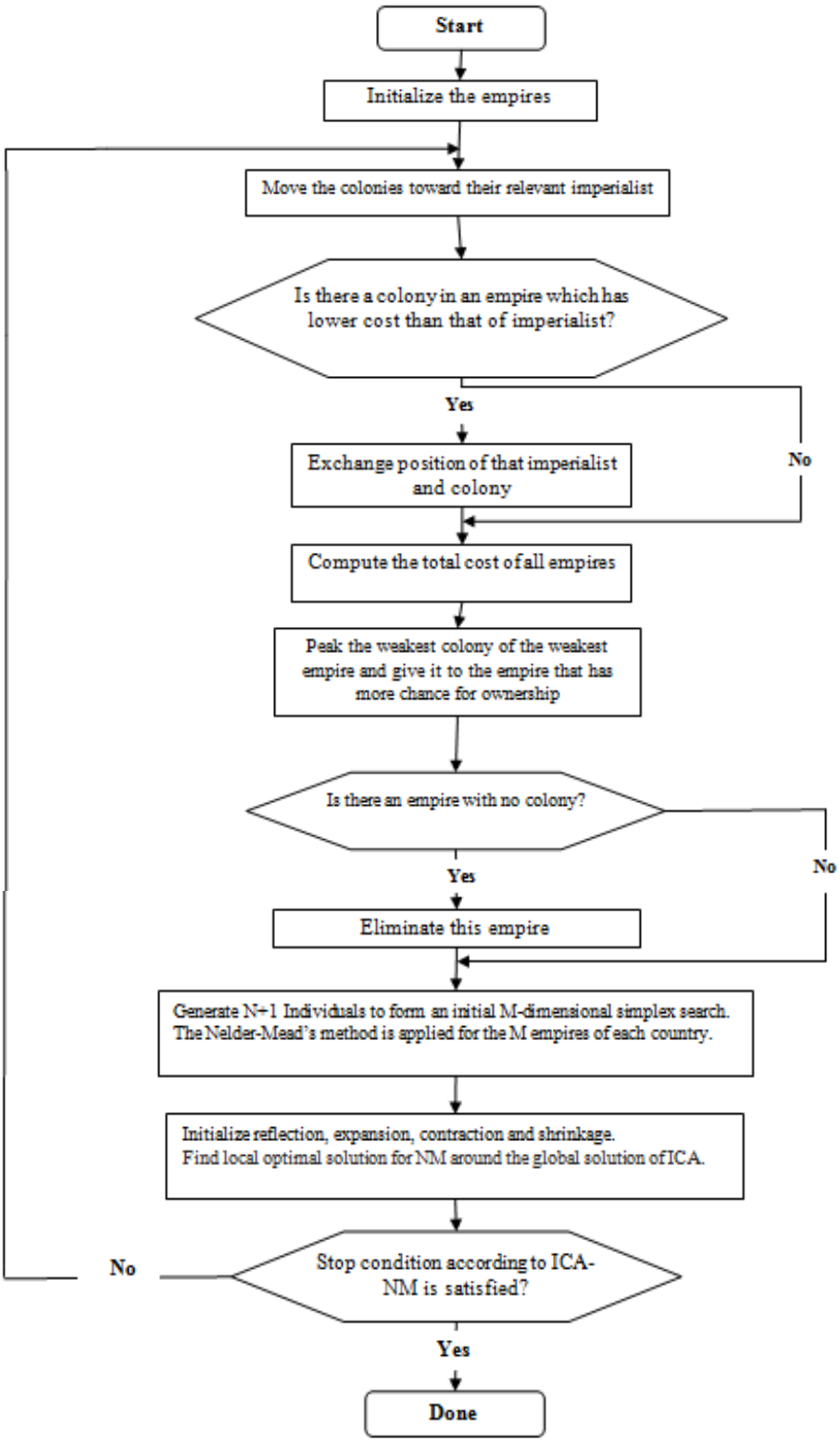

Fig. 8: Flowchart of ICA-NM hybrid algorithm

Simulation Results:

In this paper three proposed algorithms is applied to single loop PID optimization system and obtained results compare with together. With the respect to optimize of PID parameters (k_p, k_I and k_D), the proposed country is determined as follows:

Journal of Applied Science and Agriculture, 8(5) October 2013, Pages: 490-499

Table 1: Proposed country

k_P	k_I	k_D

The transfer function of the object control model is defined as:

$$G(s) = \frac{20}{1.5S^2+4.5S+1} \quad (24)$$

The single loop PID parameter tuning for the case study system is accomplished by ICA-NM Hybrid Algorithm, ICA, Genetic Algorithm, ACO and Ziegler-Nichols. The desired parameters are obtained according to Table 2.

Table 2: Parameters of the controllers and their related characteristics of step response

Method and Controller	f_{MO}	f_{RT}	f_{ST}	f_{IAE}	f_{Total}
ICA-NM Kp=3.571 K_I=3.192 K_D=4.308	0.1906	0.33	2.64	0.8597	4.0203
ICA Kp=3.445 K_I=2.166 K_D=4.233	0.1818	0.3	2.85	0.8614	4.1932
GA Kp=3.369 K_I=2.125 K_D=4.204	0.1762	0.31	2.86	0.8669	4.2131
ACO Kp=2.517 K_I=2.219 K_D=1.151	0.156	0.62	4.99	0.8861	6.6525
Ziegler-Nichols Kp=2.19 K_I=2.126 K_D=0.565	0.165	0.73	5.37	0.9595	7.2242

The convergence curve and the step response of ICA-NM Hybrid Algorithm is shown in figure 9 and 10 respectively.

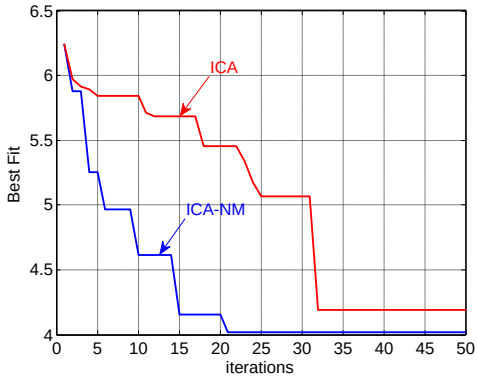

Fig. 9: Convergence curve of ICA and ICA-NM Hybrid Algorithm

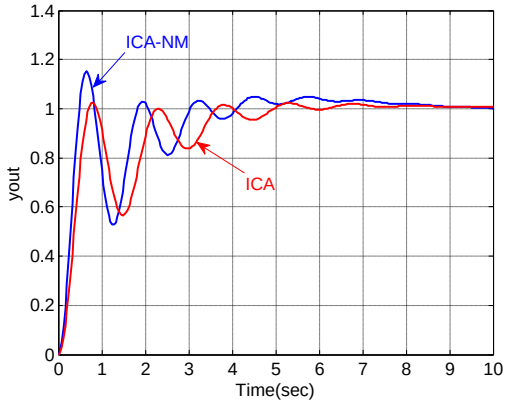

Fig. 10: Comparison of step response for ICA and ICA-NM Hybrid Algorithm

Journal of Applied Science and Agriculture, 8(5) October 2013, Pages: 490-499

Conclusion:

In this paper the PID controller parameters have been tuned by ICA-NM Hybrid Algorithm comparing with ICA, GA, ACO and Ziegler-Nichols. The simulation results reveal that ICA-NM Hybrid Algorithm is powerful optimization tool for adjusting PID controller parameters. The PID controller is designed in such a way, minimizes the sum of settling time, rise time, maximum overshoot and integral absolute error. Comparison among ICA-NM Hybrid Algorithm and the others shows better results for designing PID controller.

REFERENCES

Atashpaz-Gargari, A.E., C. Lucas, 2007. "Imperialist Competitive Algorithm: An Algorithm for Optimization inspired by Imperialistic Competition", *IEEE Congress on Evolutionary Computation*, pp: 4661-4667.

Bandyopadhyay, R., U.K. Chakraborty, D. Patranabis, 2001. "Autotuning a PID controller: A fuzzy-genetic approach", Journal of system architecture., 47: 663-673.

Bianchi, F.D., R.J. Mantz, C.F. Christiansen, 2008. "Multivariable PID control with set-point weighting via BMI optimization", *Automatica*, 44: 472-478.

Farahani, M., S. Ganjefar, M. Alizadeh, 2012. "PID controller adjustment using chaotic optimization algorithm for multi-area load frequency control", *IET journal*, 6(13): 1984-1992.

Han, H., A. Luo, 2005. "Nonlinear PID controller based on genetic tuning algorithm", *Control and Decision*, 20(4): 448-450.

Herreros, A., E. Baeyens, J.R. Pera'n, 2002. "Design of PID-type controller using multiobjective genetic algorithms", ISA transaction., 41: 457-472.

Kim, dT.H., I. Maruta, T. Sugie, 2008. "Robust PID controller tuning based on the constrained particle swarm optimization", *Automatica*, 44(4): 1104-1110.

Krohling, RR.A., J.P. Rey, 2001. "Design of optimal disturbance rejection PID controllers using genetic algorithm", *IEEE Trans. Evol Comput*., 5(1): 78-82.

Lee, K.C., S. Lee, H.H. Lee, 2004. "Implementation and PID tuning of network-based control systems via Profibus polling network", Computer standards & Interfaces, 26: 229-240.

Nelder, J.A., R. Mead, 1965. A simplex for function minimization, The computer journal, 7: 308-313.

Padula, F., A. Visioli, 2012. "On the stabilizing PID controller for integral processes", *IEEE Transaction*, 57(2): 494-499.

Shen, J.C., 2002. "New tuning method for PID controller", *ISA Transaction.*, 41: 473-484.

Yang, Y., 2006. "PID control for a binary distillation column using a genetic searching algorithm", *WSEAS Trans. Systems*, 5(4): 720-726.

Ziegler, J.G., 1942. "Nichols N B optimization setting for automatic control", *Trans. ASME*, 64(11): 756-769.