PID design with bio-inspired intelligent algorithms for high order systems

Ilir Juniku, Petraq Marango

***Abstract*—**In this paper we have proposed the application of PSO and BFO intelligent algorithms to find the optimal parameters of a PID controller, applied in the control of a high order process. The case of PID design for high order systems with time delay is also considered, since time delays introduce problems in the stability and performance of the control system. A fixed parallel structure of PID controller and four most common integral performance indexes are considered as optimization functions in this paper. The evaluation of the proposed methods is performed by using the classical performance measures in time domain.

***Keywords*—**BFO algorithm, Integral performance index (IAE, ISE, ITAE, ITSE), PID controller, PSO algorithm.

I. Introduction

MANY processes in industry are characterized by large time constants and time delays. These processes are usually represented in process control by high order mathematical models with or without time delays.

Delays are usually present in control systems as delays in computing processes or as delays in information transmission. They are very common in chemical, biological, information, and also measuring processes. Time delays introduce problems in process control due to decrease of robustness and performance deterioration, which affect the system's stability.

PID controllers, predictive control, and time-delay compensators are some of the several invented control methods, for control of high order time-delay processes [1]. Among them, PID controllers have found wider application. Their popularity is related to the fact that they are simple to understand and to operate by operators, and are effective and robust in control. Even though, approximately 95% of the control schemes in practice are based on PID controllers, finding the right parameters of these controllers to achieve maximum control performance, poses a challenge in itself. There exist many methods for the calculation of the PID optimal parameters that help to obtain a specific characteristic of process time response. To verify the effectiveness of various methods for PID controllers, the comparison is usually made by analyzing the transient characteristics of the system. The characteristics obtained by tuning the PID parameters, often do not meet the control performance criteria defined by the designer. This is mainly due to the fact that many processes in nature are of high order, and are characterized by time delays and nonlinearity [2]. For this reason, latest research is focused on optimization methods based on intelligent algorithms, which result very efficient in solving difficult optimization problems. Evidence of successful application of these algorithms can be found in [3]-[11]. Algorithms, inspired by characteristics and organized behaviors of organisms and microorganisms in nature, have recently achieved an increasing interest. Among these algorithms, we have chosen Particle Swarm Optimization (PSO) and Bacterial Foraging Optimization (BFO) algorithms as optimization methods. These two algorithms are the main focus of this work and our proposal is to apply them in finding the optimal PID parameters for a high order control system. The case related to the presence of time delays in high order systems will also be considered**.**

Previous research has been conducted on application of intelligent techniques such as Genetic Algorithm (GA) and Differential Evolution (DE) for the optimization of PID controller parameters for high order systems and systems with time delay [12]. PSO is proved to be more efficient than GA algorithm method and also has less computation codes. BFO algorithm is more complex and requires more computational capacity compared to PSO algorithm [3].

Four well-known integral performance criterions are proposed as optimization functions in our case. These performance indexes will be used to obtain the coefficients of PID controller, and then process transient responses will be analyzed and compared with the methods of obtaining PID coefficients from PSO and BFO algorithms.

The structure of the article is as follows. Section II presents the proposed control schemes for the high-order process with/without time delay, transient response measures and also the integral performance indexes used as optimization (cost) functions to find the coefficients of PID controllers. In section III, PSO and BFO algorithms, and their application in finding the coefficients of PID controllers are treated. The process models that will be considered for various simulations with classical methods (IAE, ISE, ITAE, ITSE) and intelligent methods (PSO and BFO) is presented in section IV. Conclusions obtained from simulations are presented in

I. Juniku is with the Department of Automatic Control, Faculty of Electrical Engineering, Polytechnic University of Tirana, Blvd. Deshmoret e Kombit, Tirana, Albania, (e-mail: ijuniku@hotmail.com).

P. Marango is with the Department of Automatic Control, Faculty of Electrical Engineering, Polytechnic University of Tirana, Blvd. Deshmoret e Kombit, Tirana, Albania, (e-mail: marango@fie.upt.al).

section V. Algorithms and computational simulations are performed in MATLAB environment.

II. Optimization Functions

A. Control Scheme

The proposed control scheme for finding the optimal coefficients of PID controller is illustrated in Fig. 1. Signals presented in the control scheme are: reference signal *R(s)*, that is a step unit function in our case, output signal of the system *Y(s)*, control signal *U(s)*, and error signal *E(s)* derived from *E(s) =R(s)-Y(s)*.

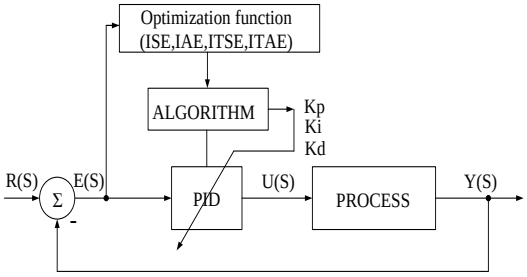


Fig. 1 Proposed control scheme for the process.

Proposed PID controller is in its parallel form, provided by the algorithm:

$$u(t)= K_p e(t)+ K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt} e(t) \quad (1)$$

where

$u(t)$-control signal in time domain

K_p -proportional coefficient, a tuning parameter

K_i -integral coefficient, a tuning parameter

K_d -derivative coefficient, a tuning parameter

$e(t)$ -error signal in time domain

In Fig. 2 is presented the typical PID controller structure in its parallel form. The process that will be studied is a single input-single output (SISO) high order system with or without time delay, which represents many processes in industry. The delay term represents the general effect of the plant inertia, and its high order and parameter distribution [13].

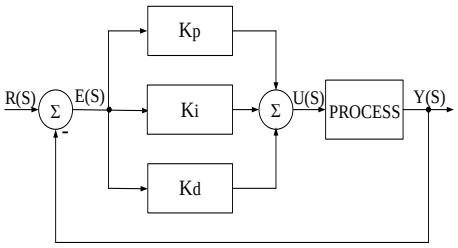


Fig. 2 Structure of PID controller

The mathematical model of delay term e^{-Ls} according to [14] can be approximated by a rational transfer function of the form:

$$G_i(s)= \frac{1}{\left(1+\frac{Ls}{i}\right)^i},\ i=1,2,\ldots \quad (2)$$

which is an *i*-th order truncation of expression

$$e^{-Ls} = \lim_{i\to\infty} \frac{1}{\left(1+Ls/i\right)^i}. \quad (3)$$

B. Transient Response Measures

Analysis for the process transient response in time domain is done through performance quantities [15] like:

-Rising time t_r: time required for the output of the system to reach 90% of its final value $h(\infty)$.

-Settling time t_s: time after which the output remains within ±2% of the final value $h(\infty)$.

-Peak value h_{max}: peak value of the transient response $h(t)$ of the process.

-Peak time t_{peak}: time required for the transient response $h(t)$ to reach the peak value h_{max}.

-Overshoot M_r *(%)*: output value exceeding final, steady-value of the process, expressed in percentage.

Rising and settling times are measures of response speed of the system. Overshoot, peak value, and peak time are related measures to the quality of response.

C. Integral Performance Criteria

In classical control methods, the performance of the entire control system can be estimated quantitatively using a single parameter that is the integral quality criterion *J*. This performance index is useful in treating optimization of parameters and obtaining optimal control designs. According to [16], a system is considered as an optimal control system when the system parameters are tuned to achieve an integral criterion ekstremum, which is usually a minimum value. Integral criterion should always be a positive number or equal to zero $J \geq 0$. The best achievable control system is the system that minimizes this criterion. The general form of the integral performance criterion is:

$$J = \int_0^T f(e(t),r(t),y(t),t)dt \quad (4)$$

where f is a function of error, input, ouput signals and time.

The most common integral criterions are:

- Integral of squared error $ISE = \int_0^T e^2(t)dt$ (5)

where T is a finite time, chosen arbitrarily, in order for the integral to approach the final stabilized value of the system. Generally T is chosen as equal to t_s-settling time.

-Integral of absolute error $IAE = \int_0^T |e(t)|dt$ (6)

-Integral of time multiplied absolute error $ITAE = \int_0^T t|e(t)|dt$ (7)

-Integral of time multiplied squared error $ITSE = \int_0^T te^2(t)dt$ (8)

III. Intelligent Algorithms

Nature is the perfect example of optimization, because many organisms and microorganisms in nature always interact or collaborate to find the optimal strategy for a specific natural process. The real interesting fact about nature inspired

algorithms is that they receive their inspiration from nature-related behaviours. These bio-inspired algorithms have found recent applications in computer networks, robotics, control systems, security systems, power systems and many more [17].

A. PSO Algorithm

The PSO is created by Eberhart and Kennedy in 1995 [18], [19]. The method is suitable for solving nonlinear problems and to give optimal solutions [20]. This algorithm is inspired by natural behavior of animals, such as organized behavior of birds in flock for finding food [21], [22]. PSO algorithm has simple computations because it does not require gradient calculations. Since it is a stochastic technique with a large number of candidate solutions, it is unlikely to be stuck at a local minimum [20]. Its suitability in distributed computing environments makes it a helpful tool for online controller tuning [3].

PSO algorithm operates using a population (called swarm) of a potential candidate solution (called particles). These particles move around a search space according to a specific routine (law). Movements of particles are guided by their best known position in the search space and also the best known position by the whole flock. When better positions are detected, these positions guide the further movement of the particles. The process is repeated until a satisfactory solution is reached. In this optimization method, a set of particles are placed in a *d*-dimensional space with a random specified speed and position. The initial position of the particle is taken as the best position in the beginning and then particle speed is reassessed based on the experience of other particles of the flock (population).

From [23], PSO algorithm elements are:

-the i-th particle in the population represented by:

$$x_i = (x_{i1}, x_{i2}, x_{i3}, \ldots x_{id}) \tag{9}$$

in the *d*-dimensional space.

-previous best positions of the *i*-th particle represented by:

$$P_{opt} = (P_{opt_{i,1}}, P_{opt_{i,2}}, P_{opt_{i,3}}, \ldots P_{opt_{i,d}}) \tag{10}$$

-the index of the best particle in the swarm $G_{opt,d}$

-the speed of the *i*-th particle represented by:

$$V_i = (v_{i1}, v_{i2}, v_{i3}, \ldots v_{id}) \tag{11}$$

-reassessed speed and distance from $P_{opti,d}$ in $G_{opti,d}$ given by the law:

$$V_{i,m}^{t+1} = W \cdot V_{i,m}^{t} + C_1 \cdot rand() \left(P_{opt_{i,m}} - X_{i,m}^{t}\right) + C_2 \cdot rand() \left(G_{opt_m} - X_{i,m}^{t}\right) - X_{i,m}^{(t+1)} = X_{i,m}^{(t)} + V_{i,m}^{(t+1)} \tag{12}$$

for *i=1,2,3,...n ; m=1,2,3,...d*

where *m* number of particles in swarm, *d* dimension index, *t* iteration index, $V_{i,m}^{(t)}$ the particle speed in iteration *i*, *W* weighting factor of inertia, C_1,C_2 acceleration constants, *rand()* random number between 0 and 1, $X_{i,m}^{(t)}$ actual position of the *i*-th particle in iteration.

Typically the number of particles in swarm is 20-40 to be able to solve most of the problems [24]. Constants C_1, C_2 represent the degree of influence of individual and group experience on the speed adjustment respectively. When C_1 is zero, the particle velocity is completely free from the impact of individual experience. In this case, it will soon converge, but it can easily fall into local minimum, and not into global optimum. When C_2 is zero, the particle velocity is completely free from the impact of group experience. This allows each particle to adjust its position according to their own experience, and in this case we don't receive an effective solution. Therefore, in order to ensure global convergence of particle swarm speed and fitness, we need appropriate C_1 and C_2 values [24]. Inertia weight *W* affects the capability of the global search. When *W* is large, the speed of the particles in the iterative process will be large, global search ability will be stronger. Conversely, when the *W* is small, the speed of particles in the iterative process will be small, local search capability enhanced, and global search capability weakened. Therefore, appropriate inertia weight can make PSO have better global search capability, while also having fast convergence rate. In the beginning of the algorithm, PSO has no inertia weight, meaning that the inertia weight is a fixed value of 1. However, in the process of the convergence of the particles, the inertia weight should be a dynamic value. So, at the beginning of the algorithm, the particle should have a large global search capability, to allow the rapid convergence of the particle swarm and to find the approximate location of the optimal solution. In the end of the algorithm, it's needed to increase the local search ability of particles, in order to achieve the optimal solution of the particles. Typically inertia weight of initial value is 0.9, the minimum value is 0.4. Random number is used to keep impressing particles with the randomness of the flight, to make sure particles can jump out of bondage, enlarging the searching scale [24].

The flowchart of PSO algorithm is illustrated in Fig. 3.

B. BFO Algorithm

BFO is based on research conducted by K.M. Pasino [25], [26], related to development and behavior of E.coli bacteria. This algorithm is gaining popularity in research community, due to its effectivenes in solving difficult real-world optimization problems. Some successful applications of this algorithm can be found in optimal control, transmission loss reduction, machine learning, etc. Usually BFO performance is poorer than PSO or GA algorithms due to it greater dimensions of searching space [27]. BFO due to its unique dispersal and elimination technique can find favourable regions when the population involved is small. The algorithm of this technique is composed of two steps: initialization and iterative algorithm for optimization [28].

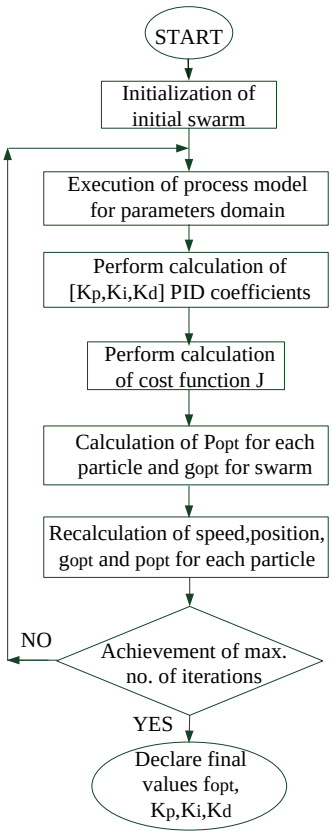


Fig. 3 Flowchart of PSO algorithm

In this optimization method there are four typical behaviors that imitate nature [29]:

Chemotaxis This process resembles the movement of a bacterium (E.coli) through swimming and displacement via flagella. Biologically an E.Coli can move in two different ways. It can swim for a period of time in one direction, or it can move alternatively between two modes of movements throughout lifetime. To represent a shift we use a casual direction with a given unit size by $\theta(j)$. This presentation is used to determine the movement direction after a displacement. In particular: $\theta^i(j+1,k,l)=\theta^i(j,k,l)+C(i)\cdot\theta(j)$ where $\theta^i(j,k,l)$ represents the i-th bacterium in the j-th chemotaxis, the k-th reproduction, the l-th elimination and dispersal step $C(i)$ is the size of the step taken in a random direction specified by the unit size $\theta(j)$. N_s represent the swimming length after which tumbling of bacteria will be undertaken in a chemotactic loop and N_c the number of iterations to be undertaken in a chemotactic loop ($N_c>N_s$).

Swarming E.coli bacteria organize themselves into well-structured colonies with high environmental adaptability using a complex communication mechanism. To create the colonies, bacteria produce signals which are attractive to each other. Analytical presentation of this process is:

$$J_{cc}=(\theta,P(j,k,l))=J_{cc}^{i}(\theta,\theta^{i}(j,k,l))=\sum[D_{attract}\cdot e^{(-W_{attract}\cdot\sum(\theta_m-\theta_m^i)^2}]+$$
$$+\sum[H_{repellant}\cdot e^{(-W_{repellant}\cdot\sum(\theta_m-\theta_m^i)^2}] \quad (13)$$

where $J_{cc}(\theta,P(j,k,l))$ is the value of the optimization function (to be minimized) to represent a cost function that depends on time. S is the total number of bacteria; P is the number of parameters to be optimized, which are present in each bacteria and $D_{attract}$, $W_{attract}$, $H_{repellant}$, $W_{repellant}$, are different coefficients that should be chosen carefully.

Reproduction Less healthy bacteria die and each healthier bacteria split into two daughter bacteria, each located in the same position. In this process N_{re} represents the maximum number of reproduction to be undertaken.

Elimination and Dispersal In the local environment, it is possible that the bacteria life of a population can change gradually (e.g. through the consumption of nutrients) or abruptly from other influences. It may happen that in a zone all the bacteria die, or a group is dispersed in a new environment. They can destroy the progress of chemotactic effect, but they can also help the effect, if dispersal occurs in areas with good food sources. From a broader perspective, elimination and dispersal are part of the motion behavior of the population for long distances. N_{ed} represents the maximum number of elimination and dispersal events to be imposed over the bacteria, P_{ed} probability with which the elimination and dispersal will continue.

In the first step are set the initial values of the algorithm parameters P, S, N_s, N_c, N_{re}, N_{ed}, P_{ed}, $C(i)$, $\theta(j)$, $D_{attract}$, $W_{attract}$, $H_{repellant}$, $W_{repellant}$. In the second step, the bacterial population (chemotaxis process), swarming, reproduction, elimination and dispersal are modeled. By updating the θ^i, automatically results in updating the P [28]. Flowchart of BFO algorithm [30] is illustrated in Fig. 4.

IV. Simulation Results

In this study we have taken a high (forth) order process, which has a characteristic with many oscillations. Two cases are considered: first case with no time delay (G_1) and second case with time delay (G_2), represented by the following models in Laplace domain.

$$G_1(s)=\frac{10}{s^4+10s^3+35s^2+50s+24}$$
$$G_2(s)=\frac{10}{s^4+10s^3+35s^2+50s+24}e^{-3s} \quad (14)$$

As shown in Section II.A, the delay in time will appear in a rational function form:

$$G_{delay,1}(s)=e^{-3s}=\frac{1}{3s+1} \quad (15)$$

Resulting from above, the transfer function G_2 to be considered during the simulations is:

$$G_2(s)=\frac{10}{3s^5+31s^4+115s^3+185s^2+122s+24} \quad (16)$$

Usually, to achieve an optimal PID controller, the four transient response measures, such as rise time t_r, steady state error E_{ss}, settling time t_s, and overshoot M_r are used. The optimization function in this case is:

$$J = (1 - e^{-\beta})(M_r + E_{ss}) + e^{-\beta}(t_s - t_r) \quad (17)$$

where β is a weighting factor ($\beta<0.7$ to decrease settling time t_s and rise time t_r and $\beta>0.7$ to decrease M_r and E_{ss}).

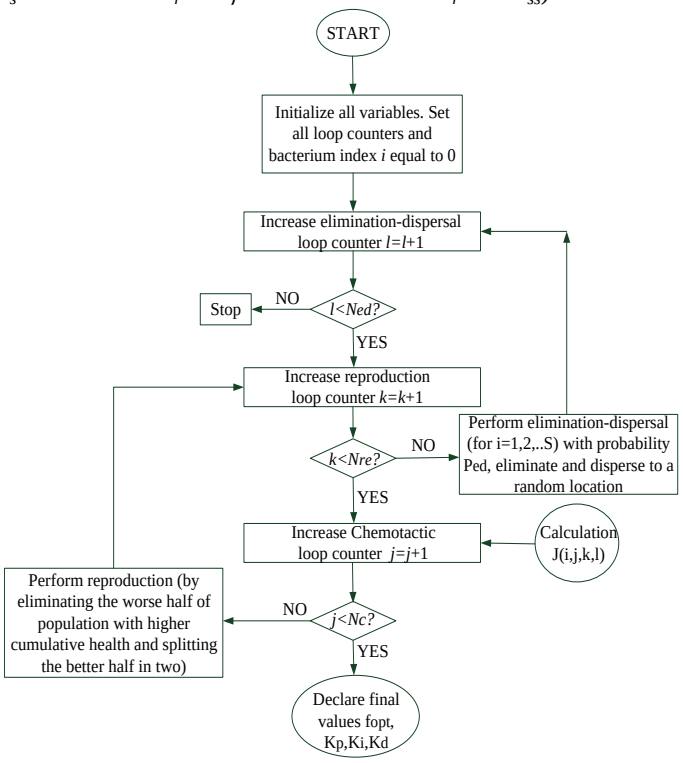


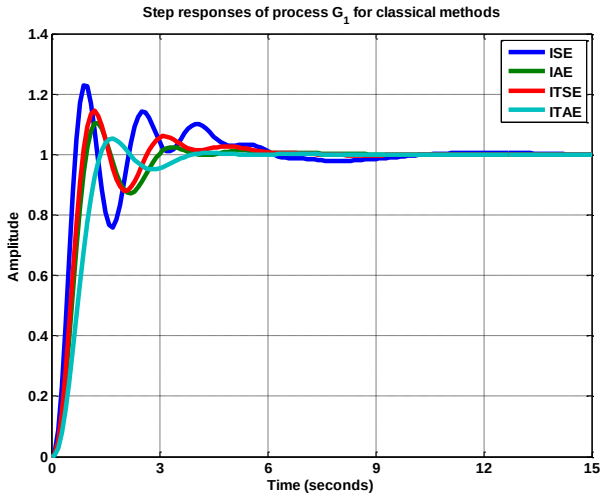
Fig. 4 Flowchart of BFO algorithm

This cost function was used in [20] in the case of PSO algorithm. The inverse of function (17) can be used as cost function in the case of BFO algorithm. In the proposed control scheme (Fig. 1), it is intended to tune the three coefficients of PID controller, in order to obtain the best (optimal) output results. Our proposal is to use integral criterions as cost functions, and to evaluate the performance of various combinations of PID coefficients in a 3-dimensional search domain. Each point in this 3-dimensional search domain for the proposed algorithms, represents a certain combination of [*Kp,Ki,Kd*] coefficients, for which a certain transient response of the system is achieved. In the proposed PSO and BFO algorithms, used to achieve the minimization of the integral performance indexes, the PID parameters are used as input values and as output is used the optimization value of the PID controller model (18).

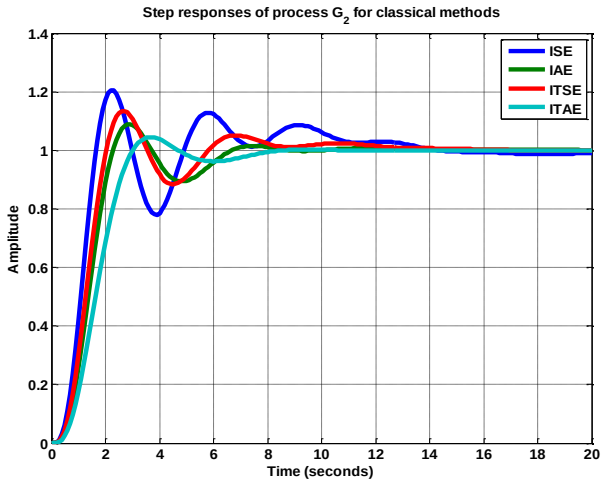
Function [J]=integral criteria (K_d,K_p,K_i) (18)

A. Classical Approach

Using IAE, ISE, ITAE, ITSE integral criterions, treated in Section II.C, in Fig. 5 are obtained the transient responses for processes G_1 and G_2. Executing the algorithms in [31], we obtain the corresponding PID controllers for our integral performance criterions.



a)



b)

Fig. 5 Step responses for classical algorithms: a) process G_1, b) process G_2.

PID controllers, obtained by the classical algorithms for processes G_1 and G_2 are respectively shown in Table I.

Table I. PID controllers obtained by classical algorithms for process G_1 and G_2.

Performance Index	Process	Coefficients of PID controllers		
		K_p	K_i	K_d
ISE	G_1	7.084	8.522	10.33
	G_2	7.616	3.112	18.49
IAE	G_1	7.262	4.091	5.353
	G_2	7.579	1.699	10.42
ITSE	G_1	7.492	5.322	6.186
	G_2	7.941	2.165	12.50
ITAE	G_1	5.833	3.180	3.100
	G_2	6.394	1.427	6.808

B. *Intelligent PSO and BFO algorithms approach*

In order to find the PID controllers coefficients by the intelligent algorithms PSO and BFO, in this study we have used the 3-dimensional search domain where the K_p, K_i, K_d values are the three dimensions of searching domain. In both algorithms, the four integral criterions of time domain, treated in Section II.C, were chosen as optimization functions. The algorithms were executed in Matlab environment where as cost functions were used:

- ISE: *J=e'*e*dt*
- IAE : *J=sum(abs(e)*dt)*
- ITSE: *J=(t.*e'*dt)*e;*
- ITAE: *J=sum(t'.*abs(e)*dt)*

In order to achieve the optimization goal, the PSO and BFO algorithm parameters should be selected properly [21]. The initial constants for the computing PSO algorithm were taken m=25, *W=0.7, C_1= C_2=1.5*. Upon execution of PSO algorithm, at Matlab prompt are displayed the *[Kp, Ki, Kd]* items of search domain, which is the final point of global optimum noted as g_{opt}, that corresponds to the minimum value of cost function, noted as f_{opt}.

Coefficients of PID controllers, obtained by intelligent PSO algorithm are shown in Table II. In Table II are also shown the best (minimum) values of cost functions (integral criterions). The constants used for the initialization of the computing program in BFO algorithm are chosen as $D_{attractant}=0.01$, $W_{attractant}=0.01$, $H_{repellant}=0.01$, $W_{repellant}=0.01$.

Table II. PID controllers obtained by PSO algorithms for processes G_1 and G_2.

Performance Index	Process	Coefficients of PID controllers K_p	K_i	K_d	f_{opt}
PSO-ISE	G_1	5.9979	4.5977	10.045	0.4167
	G_2	5.8358	2.4775	7.7808	1.3484
PSO-IAE	G_1	6.6516	3.0712	4.8638	0.8184
	G_2	5.9132	3.2248	8.4042	3.3861
PSO-ITSE	G_1	5.8190	4.5353	5.0056	0.1566
	G_2	6.3134	2.2741	8.7790	1.2883
PSO-ITAE	G_1	8.3095	4.2609	4.2611	0.7443
	G_2	4.4448	1.5272	5.5659	4.2088

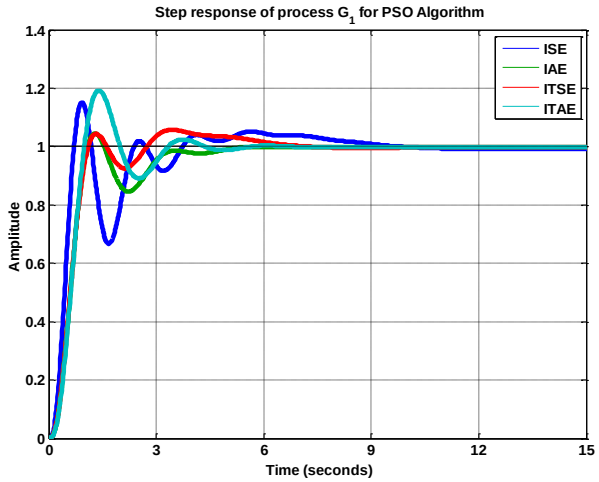
Table III presents the values of PID controller coefficients obtained by BFO algorithm, and the corresponding minimum values of cost functions for processes G_1 and G_2.

Table IV presents the transient response measures for processs G_1 and G_2 for the three investigated methods.

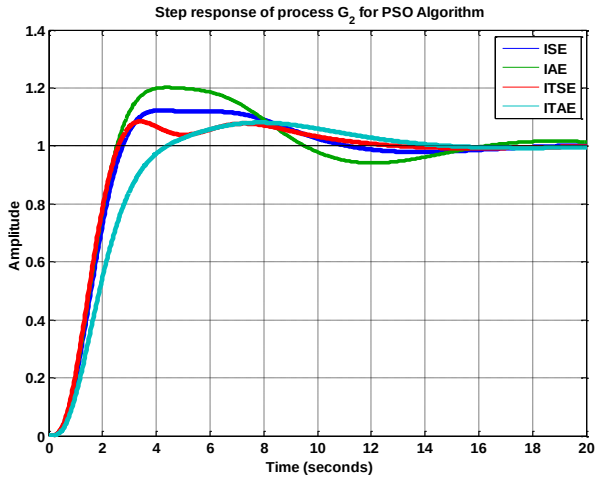
Table III. PID controllers obtained by BFO algorithms for process G_1 and G_2

Performance Index	Process	Coefficients of PID controllers K_p	K_i	K_d	f_{opt}
BFO-ISE	G_1	6.8883	4.4345	7.0298	0.4042
	G_2	4.4140	1.5289	3.3416	1.6023
BFO-IAE	G_1	7.1812	6.2277	6.6093	0.7874
	G_2	5.1516	1.6386	3.9190	2.1803
BFO-ITSE	G_1	7.1376	4.4838	4.7165	0.1403
	G_2	5.9585	2.8791	8.0138	0.8610
BFO-ITAE	G_1	6.4745	4.5093	3.4626	0.5943
	G_2	5.7687	2.8277	8.0427	0.9163

Fig. 6 presents the step responses for processes G_1 and G_2 obtained by PSO algorithm for various optimization functions.



a)



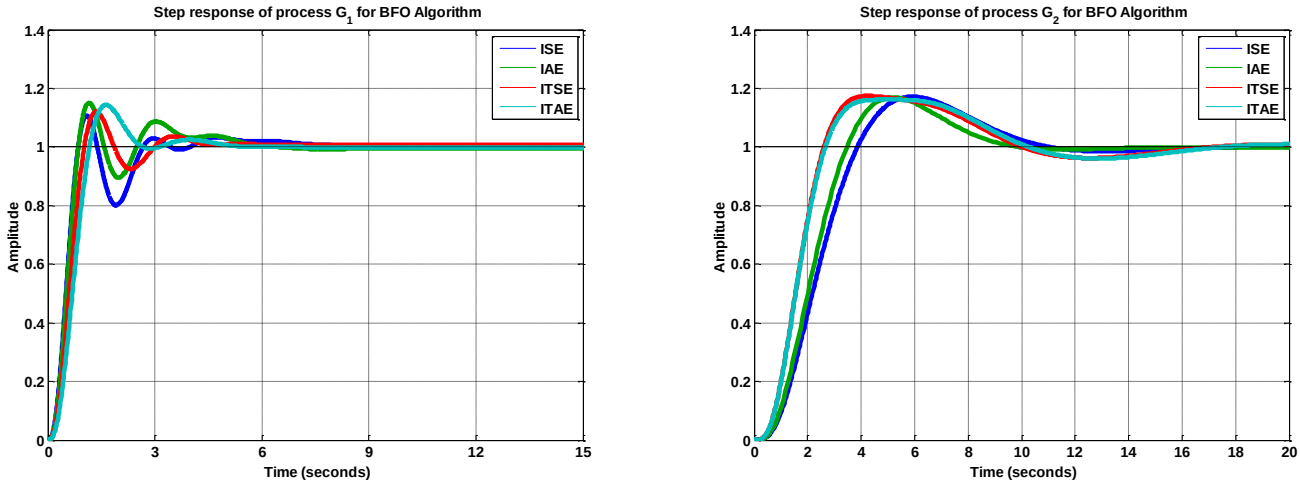
b)

Fig.6 Step responses for PSO algorithms: a) process G_1, b) process G_2.

Table IV. Transient response characteristics for processes G_1 and G_2

Characteristics	Method	Process G1				Process G2			
		ISE	IAE	ITSE	ITAE	ISE	IAE	ITSE	ITAE
Rising time t_r	Classical	4.073	5.996	5.451	8.335	9.398	13.222	11.734	17.489
	PSO	0.425	0.655	0.665	0.609	1.633	1.510	1.520	2.452
	BFO	0.520	0.523	0.625	0.723	2.389	2.118	1.557	1.587
Settling time t_s	Classical	84.948	36.467	54.598	35.841	131.74	64.118	115.39	73.285
	PSO	8.092	4.550	5.786	3.950	14.479	15.028	10.848	12.634
	BFO	5.564	5.326	4.059	4.371	10.126	8.866	15.057	15.472
Overshoot M_r (%)	Classical	22.903	10.348	14.461	5.208	20.356	8.940	13.421	4.515
	PSO	15.177	4.541	5.711	19.196	12.298	20.150	8.428	8.064
	BFO	10.663	15.052	12.066	14.342	17.141	16.917	17.408	16.118
Peak value h_{peak}	Classical	1.229	1.104	1.145	1.052	1.206	1.089	1.134	1.045
	PSO	1.152	1.045	1.057	1.192	1.123	1.202	1.084	1.081
	BFO	1.107	1.151	1.121	1.143	1.172	1.169	1.174	1.161
Peak time t_{peak}	Classical	10	14	13	18	23	30	28	38
	PSO	0.930	1.295	3.442	1.393	4.184	4.409	3.412	7.845
	BFO	1.096	1.162	1.362	1.628	5.920	5.204	4.237	4.902

Fig.7 presents the process transient responses obtained by BFO algorithm, for various optimization functions.



a) b)

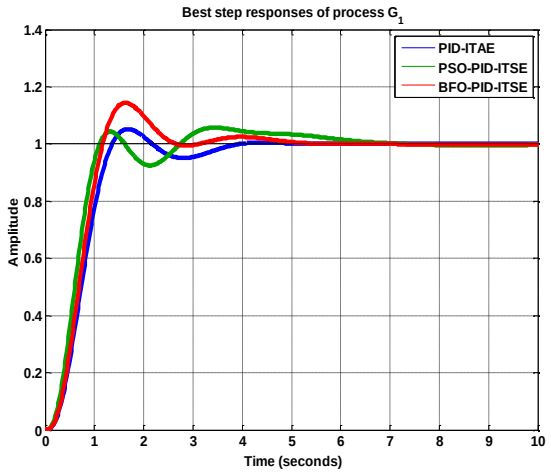
Fig. 7 Step responses for PSO algorithms: a) process G_1, b) process G_2.

Fig. 8 presents the best transient responses of process obtained by ITAE, PSO-ITSE, BFO-ITSE algorithms.

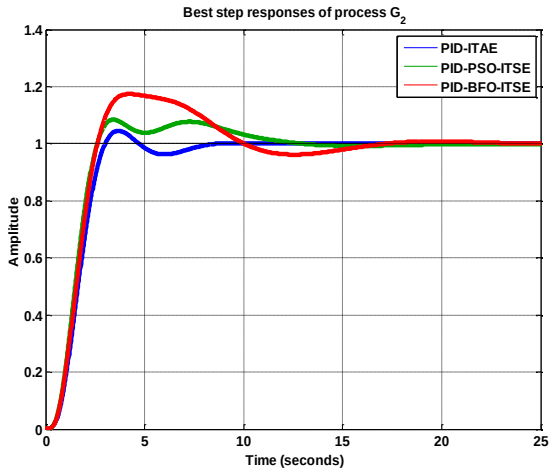
V. Conclusion

Based on the performed simulations, we arrive at the conclusion that methods based on PSO and BFO intelligent algorithms are quite efficient in achieving a very good control performance of high order processes and especially high order processes with time delay. Specifically, the resulting rise times t_r and settling times t_s are reduced considerably (respectively 6-12 times and 6-15 times), resulting in control systems that have a faster response to changes at the system's input. From Tables

II, III, IV, we conclude that ITSE and ITAE integral criterions, applied as cost functions to find the optimal values of K_p, K_i, K_d coefficients of PID controllers are the best functions that can be used for the methods discussed above. From the obtained results, the best integral criterion to be used in the cases of PSO and BFO algorithms is ITSE. Comparing the overall results from the application of the bio-inspired intelligent algorithms, we concluded that the PSO algorithm is more efficient and provides a better tuning of the process than BFO algorithm. It provides less overshoot in the step response of the system in the majority of the cases. Based on the simulations, computing time of PSO algorithm resulted smaller compared to BFO algorithm and this can be justified by the fact that PSO algorithm has simpler computing architecture than BFO algorithm.



a)



b)

Fig. 8 Best achieved step responses for process G_1 and G_2.

REFERENCES

[1] V. Bobal, M. Kubalcik, P. Dostal, and J. Novak, "Adaptive Predictive Control of Laboratory Heat Exchanger", *WSEAS Transactions on Systems*, E-ISSN:2224-2678, Volume 13, 2014.

[2] A. G. Abro and J. M. Saleh, "Enhanced Artificial Bee Colony Algorithm to Optimize PID-AVR Controller", *International Journal of Systems Applications, Engineering & Development*, Issue 5, Volume 7, 2013

[3] A. H. Halim and I. Ismail, "Bio-Inspired Optimization Method-A Review", *NNGT Journal: International Journal of Information Systems*, Vol.1, July 30, 2014

[4] G.D. Li, C.H. Wang, and S. Masuda, "The PID Prediction Control System using Particle Swarm Optimization and Genetic Algorithms", in IEEE International Conference of Grey Systems and Intelligent Services, November 10-12, 2009.

[5] W. Wang, Bin Wu, Y.Zhao and D. Feng, "Particle Swarm Optimization for Open Vehicle Routing Problem", in ICIC 2006, LNAI 4114, pp.999-1007, Springer-Verlag, 2006.

[6] Y.Zhang, F. Qiao, and J. Lu, "Performance Criteria Research on PSO-PID Control Systems", in IEEE International Conference on Intelligent Computing and Cognitive Informatics, 2010.

[7] H.M. Asifa and S.R.Vaishnav, "Particle Swarm Optimization Algorithm based PID Controller", in IEEE Third International Conference on Emerging Trends in Engineering and Technology, 2010.

[8] C.A. Coello Coello and M. S. Lechuga, "MOPSO: A proposal for Multiple Objective Particle Swarm Optimization", in *Proceedings of IEEE CEC '02 Congress on Evolutionary Computation*, ISBN:0-7803-7282-4, pp.1051-1056, 2002.

[9] N. Atiqah, A. Rahman, Z. Zakaria, T. Khawa and A. Rahman, "Active load and Loss Allocation in Transmission Line with Line Outage Condition via Cuckoo Search Optimization Technique", IEEE Conference (SCOReD), 2013

[10] Edite M. G.P. Fernandes, Tiago F. M. C. Martins, and Ana Maria A.C.Rocha, "Fish Swarm Intelligent for Bound Constrained Global Optimization", in *Proceedings of the International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2009*, 1–3 July 2009.

[11] X.S. Yang, "Firefly Algorithm, Levy Flights and Global Optimization", *Research and Development in Intelligent System XXVI*, Springer-Verlag, 2010.

[12] H. M. Yatim and I. Z. Mat Darus, "Self-tuning active vibration controller using particle swarm optimization for flexible manipulator system", *WSEAS Transactions on Systems and Control*, E-ISSN: 2224-2856, Volume 9, 2014.

[13] S. Yordanova, "Robustness of Systems with Various PI-like Fuzzy Controllers for Industrial Plants with Time Delay", *WSEAS Transactions on Circuits and Systems*, Issue 6, Vol.7, pp. 528-534, June 2008.

[14] J.E. Normey-Rico and E.F. Camacho, *Control of Dead Time Processes*, pp.22, Springer, New York, 2007.

[15] P.Marango, *Bazat e Automatikes* (in albanian), pp.128, SHBLU, Tirana 2011.

[16] R.C. Dorf and R.H. Bishop, *Modern Control Systems*, 12th ed., pp. 330-332, Prentice Hall, 2010.

[17] S. Binitha and S. S. Sathya, "A Survey of Bio inspired Optimization Algorithms", *International Journal of Soft Computing and Engineering (IJSCE)*, ISSN:2231-2307, Volume 2, Issue 2, May 2012.

[18] J. Kennedy and R. C. Eberhart, "Particle swarm optimization", in *Proc. IEEE International Conference on Neural Networks*, Vol. IV, pp. 1942-1948. IEEE service center, Piscataway, NJ, 1995.

[19] R. C. Eberhart and J. Kennedy, "A New Optimizer Using Particle Swarm Theory", in *Proc of 6th International Symposium on Micro Machine and Human Science*, pp.39-43, Nagoya, Japan, 1995.

[20] E.H.E. Bayoumi and Z. A. Salmeen, "Practical Swarm Intelligent Control Brushless DC Motor Drive System using GSM Technology", *WSEAS Transactions on Circuits and Systems*, Vol.13, pp. 188-201, 2014.

[21] R. C. Eberhart and Y.Shi, " Particle swarm optimization: developments, applications and resources", in *Proc.of Congress on Evolutionary Computation 2001*, pp.81-86, IEEE service center, Piscataway, NJ, Seoul, Korea, 2001.

[22] Q.Bai, "Analysis of particle swarm optimization algorithm", in *CIS Journal of Computer and Information Science*,Vol. 3, No 1, February 2010, pp.180-184, 2010.

[23] R.L.Haupt and S.E.Haupt, *Practical Genetic Algorithms*, 2nd ed., pp. 189-190, Wiley, 2004.

[24] Z. Dong, H. Wang, S. Wang, W. Hou, and Q. Zhao, "Intelligence Diagnosis Method Based on Particle Swarm Optimized Neural Network for Roller Bearings", *WSEAS Transactions on Systems*, E-ISSN:2224-2678, Issue 12, Volume 12, December 2013.

[25] Y.Liu and K.M. Passino, "Biomimicry of Social Foraging Bacteria for Distributed Optimization: Models, Principles, and Emergent Behaviors", in *Journal of Optimization Theory and Applications*, 2002.

[26] S.Das, A. Biswas, S. Dasgupta, and A. Abraham, "Bacterial Foraging Optimization Algorithm: Theoretical Foundations, Analysis, and Applications", in *Foundations of Computational Intelligence, Studies in Computational Intelligence*, vol. 3, Springer, Berlin, Heidelberg, pp. 23–55, 2009.

[27] A. Biswas, S. Dasgupta, S. Das, and A. Abraham , "Synergy of PSO and Bacterial Foraging Optimization-A Comparative Study on Numerical Benchmarks", in *Innovations in Hybrid Intelligent Systems*, ASC 44, pp.255-263, Springer-Verlag Berlin Heidelberg, 2007.

[28] A. S. Oshaba and E. S. Ali, "Bacteria Foraging: A new Technique for Speed Control of DC Series Motor Supplied By Photovoltaic System", *WSEAS Transactions on Power Systems*, E-ISSN: 2224-350X, Volume 9, 2014

[29] K.M. Passino, "Bacterial Foraging Optimization", in *International Journal of Swarm Intelligence Research*, 1(1), pp.1-16, January-March 2010.

[30] S. Dasgupta, S. Das, A. Abraham, "Adaptive Computational Chemotaxis in Bacterial Foraging Optimization: An Analysis", in *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, 2009.

[31] Y.Cao, "Learning PID Tuning III: Performance Index Optimization", Available: http://www.mathworks.com/matlabcentral/fileexchange/18674-learning-pid-tuning-iii-performance-index-optimization.