



INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

OPTIMAL DESIGN OF CONTINUOUS STIRRED TANK REACTOR CONTROL SYSTEM, USING EVOLUTIONARY OPTIMIZATION METHODS BASED ON TEACHING - LEARNING AND GENETICS

Alireza Baheri *, Gholamhossien Fereyduni, Milad Gorjian

*Department of Manufacturing, Collage Of Mechanics, Dezful Branch, Islamic Azad University, Dezful, Iran.

ABSTRACT

Continuous stirred tank reactor (CSTR) is an important issue in chemical processes and offers a diverse range of research on chemical control engineering and a variety of control methods in CSTR is used to control its parameters. CSTR various scientific control methods are used to control the parameters.

In this paper, two different control method is performed, an optimization algorithm based on training - learning and other genetic optimization algorithm. The goal is to control CSTR temperature in the presence of a set of point changes that factories are design its control and simulation it in MATLAB and also plans to design proposed control function based on the least square error (MSE) and standard deviation (SDC). The results clearly show that both control strategies provide acceptable performance due to functional changes. In other words it is tracking the robustness of the proposed methods in the face of uncertainty across achieving the reference signal in calculation. In addition, the algorithm based on training - learning has better dispersion than genetic algorithm. Genetics also has better rate than education – learning.

KEYWORDS: CSTR process, teaching-learning optimization algorithm, genetic optimization algorithm, temperature control

INTRODUCTION

CSTR control problem as a fascinating and controversial issue is similar to the non-linear dynamics, especially for control engineers. Many controllers are limited for invariant systems applications with linear time. However, in the real environment, the system nonlinear characteristics and fundamental change of parameters, which are required to wear and tear, cannot be ignored. In addition, the usage and handling systems with uncertainty in real applications, is another issue that must be addressed. In this way, the role of adaptive and intelligent control, by ability to overcome the above points is of considerable importance. One of the most popular controllers in both academic and industrial fields is PID. PID controller is used in feedback loop mechanism and has been widely employed in industrial process control since 1950. Easy implementation of PID controller is built in the more popular applications control systems that aim to correct the error between the measured output and the

desired output of this process, in order to improve the transient and steady state response as possible. The PID controller is emerging in many systems to have an acceptable performance. But sometimes, fundamental changes in the system parameters, is needed for an adaptive-based method or a method to find a closer adaptive answer. On the other hand, although the PID controller is used widely used in both academic and industrial control applications, its setting is still the controversial research domain. The limitations of traditional methods in dealing with the main reasons of restrictions, appears for powerful and flexible process. Smart bio Computing applied successfully become is inspired to solve complex problems in recent years. Evolutionary optimization algorithms, including genetic, particle swarm, imperial competition, based on the teaching - learning and to determine the controller parameters described above, are used. To set and adjust control parameters of the optimization algorithm based on

teaching - learning and genetics have been used in this article. The following will explain briefly about both algorithms.

CSTR MATEMATICAL MODELING[1],[2],[3]

Chemical reactions in a reactor, is exothermic or endothermic. Therefore the energy required or removed or added to the reactor to maintain a constant temperature. Figure 1 shows a schematic CSTR process irreversible exothermic reaction with taking place. Reaction heat in a cooler environment will be removed throughout the course of a coating around the reactor. Fluid flow is fed into the reactor. The catalyst is placed in the reactor. Fluid is perfectly mixed in the reactor to be sent out through the outlet valve. The considered cover is thoroughly mixed and at a lower temperature of the reactor, , mathematical models and the energy balance in the reactor is obtained by the mass balance equations.

$$\text{Component mass concentration} = (\text{Component mass})_{\text{Input}} + (\text{Component mass})_{\text{output}} + \text{production of component Mass} \quad (1)$$

$$(\text{component mass concentration } U+PE+KE) = (\text{H}+PE+KE)_{\text{Input}} - (\text{H}+PE+KE)_{\text{output}} + Q - W_s \quad (2)$$

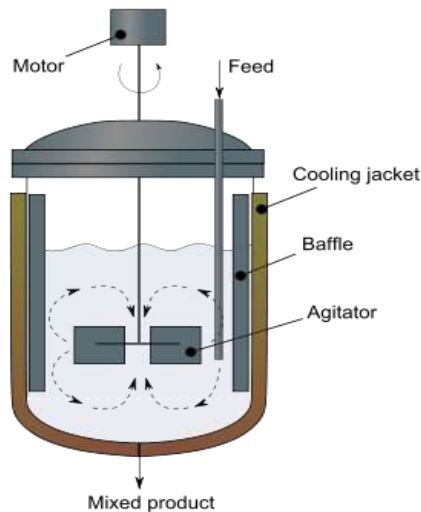


Figure 1. Schematic of a CSTR

$$\frac{dT}{dt} = \frac{F_i}{V} [\lambda T_i + (1 - \lambda)T(t - \tau(t)) - T(t)] + JK_0 e^{-E/RT} C_A - \frac{Q}{\rho c_p V} (T(t) - T_w) \quad (3)$$

$$\frac{dC_A}{dt} = \frac{F_i}{V} [\lambda C_{Ai} + (1 - \lambda)C_A(t - \tau(t)) - C_A(t)] - K_0 e^{-E/RT} V C_A(t) \quad (4)$$

Where Tw is the cover temperature as input while C_a,T are concentrations and temperature as output, respectively. It should be noted that the purpose is to control manual coating system temperature. Therefore keep it within the desired temperature Tw. So all parameters are shown in table (1).

Table 1. CSTR parameters

VALUE	SYMBOL	PARAMETER
0.5 mol/L	C _A	output current of molar concentration
1 mol/L	C _{Ai}	The inflow molar concentration
100 L/min	F	Volumetric flow rate output current
100L/min	F _i	Volumetric flow rate input current
300 K	T	reactor temperature
350 K	T _i	Input reactive material temperature
350 K	T _w	The average temperature of the cooler material
100 L	V	reactor volume
0.239 J/Kg	c _p	The heat capacity of liquid in the tank
1000 g/L	ρ	input and output density flow
5×10 ⁴ J/mol	(-ΔH _r)	reaction heat (positive)

5×10^4 J/mink	Q	Heat transfer between the coolant and Tank
8750 K	E/R	The activation energy rate to the ideal gas constant
7.2×10^{10} L/min	K_0	Kinetic coefficient

A BRIEF DESCRIPTION OF OPTIMIZATION ALGORITHMS BASED ON TRAINING - LEARNING AND GENETICS

Training - learning optimization algorithm[4],[5],[6]

TLBO is located in a class based on the training - learning process. The algorithm first introduced by Rao and his colleagues in 2011. TLBO algorithm employs learning abilities of students and teacher in the classroom to improve academic class level. Teachers and students are two main elements in TLBO. Accordingly, teachers phase and students phase are two important and fundamental part of the algorithm form. The output of the algorithm is students' grades and their knowledge level, and teacher quality and ability in this area is very effective. Therefore, teacher of each class selected by the best students of the class, to be able to help other students improve their scores. This process will be followed in teacher phase. Also, students also learn from each other to improve their scores, and will be followed in students phase.

TLBO is a modern optimization algorithm based on population, that the population is the class members. And the merit function of each, which is considered as the students score, is the objective function that should be minimal during the optimization process. Optimization procedure that performed on class population can be divided into two phases of teacher and student phase that each one are described sufficiently.

Teacher phase

This phase constitutes the first part of the algorithm that students are trying improve their level of knowledge and scores on the basis of teacher information and knowledge. During this process, the teacher is trying to make use of all its capabilities to increase the class's mean average Mean^k up to the

level of his/her knowledge Teacher^k. As a result, the difference between the class and the teacher's knowledge is formulated as follows:

$$Diference_Mean^k = Teacher^k - TF^k Mean^k \tag{5}$$

That TF^k is the learning coefficient, which controls the moving average toward the teacher and its amount it takes for 1 or 2 and follow the round(1+rand(.)) relation. Round is a function that used to round off numbers. Each of the students show their position based on the following equation.

$$X_{new1}^k = X_{old}^k + rand(.)Diference_Mean^k \tag{6}$$

If the new answer has better objective function from the standpoint of optimization problem, replace with previous solutions, otherwise the accepted answer has maintained. It is important to note that the population output of the first phase, i.e. the teacher population phase, is considered as input for the second phase, the student phase.

Student phase

This phase constitutes the second part of the TLBO optimization process, where the students increase the knowledge and information based on a compromise between their interactions. Each student randomly selected one of the students and changes their knowledge level by the following equation:

$$X_{new2} = \begin{cases} X_i + rand.(X_i - X_j) & \text{if } f(X_i) < f(X_j), i \neq j \\ X_i + rand.(X_j - X_i) & \text{else} \end{cases} \tag{7}$$

If the change is made the student's level of knowledge better, replace with previous solutions, otherwise the accepted answer has maintained. It is important to note that the population output of the first phase, i.e. the teacher population phase, is considered for the next repetition.

Implementing TLBO in optimization

Gradual implementation and enforcement strategy of TLBO is presented in this section.

The first step: optimization problem is defined and valuation parameters are optimized.

Population size (Pn), the number of generations (Gn), the number of design variables (Dn), and the limits of design variables (U_L, L_L) is determined.

Optimization problem is defined as minimizing function $f(x)$. Due to $x_i \in x_i = 1, 2, \dots, D_n$, When $f(x)$ is the objective function, X is a variable vector design as $L_{L,i} \leq x_i \leq U_{L,i}$.

Step Two: population is being evaluated.

A random population is produced according to population size and the number of design variables, and in TLBO, population are identified by size and the number of learners and design variables defined by the issues proposed (eg courses). This population can be expressed as follows:

$$\text{Population} = \begin{bmatrix} X_{1,1} & X_{1,2} & \dots & X_{1,D} \\ X_{2,1} & X_{2,2} & \dots & X_{2,D} \\ \vdots & \vdots & \dots & \vdots \\ X_{p_n,1} & X_{p_n,2} & \dots & X_{p_n,D} \end{bmatrix}$$

Step Three: Coach Stage

The population middle column is calculated, where the middle is offer on a particular topic as follows:

$$M_D = [m_1, m_2, \dots, m_D] \tag{8}$$

The best solution act as a coach for repeat

$$x_{\text{teacher}} = x_{f(x)=\min} \tag{9}$$

Coach attempts to change middle from M_D to the x_{teacher} , that define as a serious center for replication So,

$$M - \text{new}_D = x_{\text{teacher},D} \tag{10}$$

The difference between middles is as follows:

$$\text{Difference}_D = r(M - \text{new}_D - T_F M_D) \tag{11}$$

T_F , will be selected 1 or 2. The resulting difference is added to the current solution, which updates its values using the following formula.

$$x_{\text{new},D} = x_{\text{old},D} + \text{difference}_D \tag{12}$$

If the the better function was proposed, x_{new} accepted.

Forth step: learner step

As described above, learners raise their knowledge with the help of interaction between themselves. Mathematical expression described in section 3.1.2.

Step Five: The final scale

If the generation number is maximum, and the process stops, otherwise the third step is repeated.

From the above steps, no constraint is created in order to manage restrictions at issue. Many examples of constraint management techniques are available this season, such as continuity of statistical fine, dynamical fines, adaptive penalties, management method, Deb' aggressive and innovative methods, which used in proposed TLBO method.

This method uses a competitive selection operator in which two selected solution are compared with each other. Three ahead innovative laws in choices, are running.

- If a solution is practical and the other is impractical, so practical is preferred.
- If both solutions are practical, the solution is preferable that has objective function value.

If both solutions are practical, the solution is preferred that has the lowest limitation.

These laws are implemented at the end of step 2 and 3, so at the end of the coach and the learner step, instead of accepting the new solutions x_{new} , if much better function at the end of step 2 and 3 provided, Deb constraints management rules in x_{new} are use according to the innovative law.

Genetic Algorithm (GA) [7],[8]

Genetic algorithm is suitable for use in solving problems that are associated with search and optimization. Genetic algorithm uses A direct natural behavior simulation.

This algorithm works with population of "individual members", in which "fitness" is defined to each member.

It is clear that, members who have more fitness, have more opportunities for "birth" through "mixing" with the rest of the population throughout "mating". This leads to new members that inherit some of the characteristics of their parents. The lower of fitness for each members, is the lower his chances of being selected for reproduction. By choosing the best members of the current population and mating between them, a new set of members is created, that

the new population has higher demographic characteristics than the previous. By following this process, after several consecutive reproduce and create populations, member traits gradually released in populations and members are properly corrected, and thus if the algorithm is well designed, people will converge to an optimal solution of problem. So far genetic algorithms have been applied successfully in solving a wide range of issues. Of course, these algorithms do not guarantee finding the global optimum for each issue. But always act at an acceptable rate to finding solutions that are acceptable.

In general, two types of genetic algorithm are used to solve optimization problems:

1. Binary genetic algorithm
2. Continuous Genetic Algorithm

That which algorithm uses to solving an optimization problem depends on the type of problem, the optimization parameters, variation of parameters and quickly solves the problem. In the binary genetic algorithm, the coding process is used. In contrast, constant genetic algorithms is not used any coding and conversion process.

Continuous genetic algorithm components

GA, like any other optimization algorithm, started with defined optimization parameters and ends with the convergence condition examination. So in this algorithm, optimization parameters and cost function should be clearly defined.

Optimization parameters and cost function

In genetic algorithm culture, optimization of parameters set is called chromosomes. Thus, genetic algorithm optimization begins with the definition of a chromosome or set of parameters. . If the number of chromosome is Npar (optimization problem with Npar dimension), then the chromosome is shown as follow:

$$Chromosome = [P_1 \quad P_2 \dots P_{Npar}] \tag{13}$$

$P_1, P_2 \dots P_{Npar}$ are ptimization parameters that has actual numerical values in continuous genetic algorithm. In any optimization problem, by injection parameters to the cost function, the amount of the fee will be achieved for each parameter sets that will be the measure for compared to convergence algorithm. The relation shown as follow:

$$Cost = f(Chromosome) = F(P_1, P_2, \dots, P_{Npar}) \tag{14}$$

Thus, for each chromosome, there will be a cost and the this fee, is a criteria for the survival or non-survival of a chromosome, in order to create a generation and development of its properties. Usually in optimization problems, the objective of lowering the cost or minimizing the cost function. In most optimization problems , the cost function defined as the sum of squared error sum of squares SSE and relative sum of squared error $RSSE$. The functions are defined in the relations (15) and (16) :

$$SSE = \sum_{i=1}^n (F_{actual_i} - F_{estimated_i})^2 \tag{15}$$

$$\% RSSE = \frac{SSE}{\sum_{i=1}^n F_{actual_i}^2} \times 100 \tag{16}$$

F_{actual_i} is the actual signal value and $F_{estimated_i}$ is estimation of signal based on optimization parameters of specified Genetic Algorithm. Usually the sum of squared error sum less than 0.0001 and relative sum of squared error less than 0.01 defined as convergence criteria.

Initial population

Starting with the genetic algorithm, an initial population of chromosomes N_{ipop} is defined. This initial population is, in fact, is a matrix that each row of has $1 \times N_{par}$ and has N_{ipop} raw . Therefore, this matrix has $N_{ipop} \times N_{par}$ dimensions. On the other hand, by specifying ranges of values for each parameter, the matrix equation (17) can easily be created.

$$IPOP = (hi - low) \times random\{N_{ipop}, N_{par}\} + low \tag{17}$$

In the above equation:

$random\{N_{ipop}, N_{par}\}$: A function that produces a random number between zero and one

hi : The upper limit value of a parameter

low : The lower limit value of a parameter

The amount of the initial matrix value, is depending on the opinion of the investigator. GA will use a

method based on random search. But because genetic algorithm makes use of a method based on random search, the larger the initial population is likely to increase the convergence of the algorithm. Because the large initial population will diverse the range of parameters and algorithm will face with more choice. Sometimes consider the primary population very large in the range of 1,000 to 10,000 chromosomes but in the algorithm iteration process a small portion in the range of 50 to 100 chromosomes are used as the superior generation.

How to choose the superior race

In this section, how to select chromosomes that have suitable conditions for survival and the production of that are new chromosomes, is mentioned. In the first step should be calculate the cost function for each chromosome in the initial population, that chromosomes be sorted based on the lowest cost to highest. Thus, the number of N_{pop} chromosomes, which are the lowest cost for the chosen iterative algorithm process. Doing this, the size of the population in the iteration process will be $(N_{pop} \times N_{par})$. All selected chromosomes at this stage are not suitable for the production process. Usually from N_{pop} number of chromosomes in each population N_{good} that has the lowest cost is selected to create a new generation and N_{bad} number chromosomes that have higher costs are discarded (where $N_{good} + N_{bad} = N_{pop}$). Since in N_{good} chromosome, the chromosome are used pairwise to produce the new generation, it is necessary to be paired.

Mating

Among N_{good} selected chromosomes for mating, $\frac{N_{good}}{2}$ couple chosen as parents. Each pair selected, creates two chromosomes (two new parameter set) that inherit their parents to properties. The more similar the parents properties, the chromosomes from their parents in terms of characteristics, are more similar.

Generation

In this section, how to produce two new chromosomes from the two chromosomes are selected for the mating act is described. The most common way to do this, the process of cutting

between selected chromosomes. Many different methods have been cited for doing so in continuous genetic algorithm. In the simplest way, one or more locations on each chromosome are selected as cut-off points. In this method and similar methods, part of a chromosome is Transmitted to other chromosome without changing the cut point nature. This process works well in the case of binary genetic algorithms but in continuous genetic algorithms, performing only a cutting operation, leads to a limited set of numbers in different parts of the chromosome and this can lead to an optimal solution to a problem. So the proposed method for solving this problem, is a change in the parameters, especially cut-off points. So that by choosing a random point as the cut-off point in the chromosomes, the point parameter randomly changes at ranges of parameter.

For example, assume that selected couples are as follows:

$$\begin{cases} Parent1 = [P_{m1}, P_{m2}, \dots, P_{m\alpha}, \dots, P_{mN_{par}}] \\ Parent2 = [P_{d1}, P_{d2}, \dots, P_{d\alpha}, \dots, P_{dN_{par}}] \end{cases} \quad (18)$$

In this case, selecting a point quite randomly as cut-off point in the chromosome, that point is calculated from combination parameter (19):

$$P_{new} = \beta P_{m\alpha} + (1 - \beta) P_{d\alpha} \quad (19)$$

where in:

β : A random number between zero and one

$P_{m\alpha}$: α th Parameter in maternal chromosome

$P_{d\alpha}$: α th Parameter in paternal chromosome

Also the same parameter is calculated by substituting $(1 - \beta)$ instead β for second child. The α that is the cut-off point in the chromosomes, is calculated from (20):

$$\alpha = roundup \{ random \times N_{par} \} \quad (20)$$

In this case, specifying cut point, new chromosomes are calculated from (21):

$$\begin{cases} \text{offspring 1} = [P_{m1}, P_{m2}, \dots, P_{new1}, P_{d(\alpha+1)} \dots P_{dN_{par}}] \\ \text{offspring 2} = [P_{d1}, P_{d2}, \dots, P_{new2}, P_{m(\alpha+1)} \dots P_{mN_{par}}] \end{cases} \quad (21)$$

Mutation

Many optimization problems, has local extremum points, that genetic algorithm can quickly converge to these points, while the dynamic extremum point is beyond the these points. To achieve the extremum dynamic, it is necessary to fully integrate and maintain different paths algorithm, convergence of algorithm is tested. This can be done at any stage process by mutations in parameters value. In genetic algorithm this is done by convert zero to one and vice versa in a random point. Because randomly, the main parameters of the resulting amount will be diverted. A similar method in continuous genetic algorithm can be implemented on a continuous parameters. Binary genetic algorithm is works well with mutation rate between one to twenty percent for the process. The rate of more than twenty percent may be missing a lot of parameters and algorithm may not be converge. By multiplying the number of parameters in the mutation rate, the number of parameters that are changed by this action is determined. And in the next stage, the parameters with the new values and quite randomly, are replaced in their permitted area. Doing so, prevent convergence to static and extremum dynamic point there will be fined. Since the genetic algorithm using repetition, improve generation, it is better by increasing the frequency, the mutation process in reduced. This could be simulated in an exponential function in the form of equation (22).

$$\mu = \mu_o e^{-Counter} \quad (22)$$

Where

μ_o the rate of mutation

counter Represents the number of replications

THE SIMULATION EMPIRICAL STUDIES

In Sections 2 and 3 of the CSTR and optimization algorithms based on teaching-learning and genetics were introduced. In this section we compare the optimal design of control system process, using the two methods described above. For better comparison, the same circumstances considered. Least square error is considered as cost function. The parameters are as follows: population size = 10, max generation number = 10. The method was tested 10 times and

data we collected based on the best, middle, worst, answer variance and standard deviation. The best parameters obtained from the two methods are in Table 2.

Table 2. controller parameters

	Kp	Ki	Kd
TLBO	50	12.39	50
GA	45.4	20	47.7

Figure (2) and (3) shows reference input and model output using the two methods that both methods provide acceptable solutions

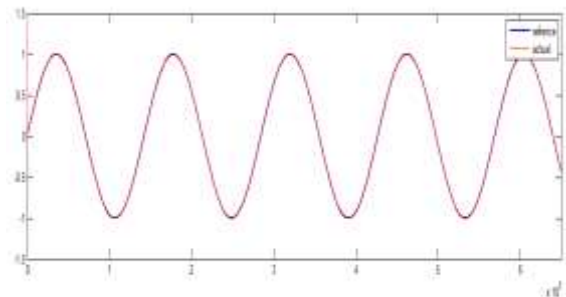


Figure 2. Reference input and model output TLBO

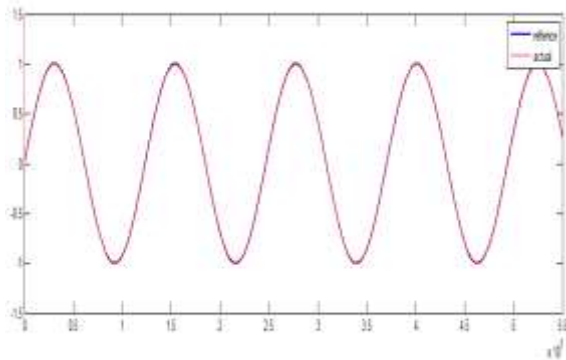


Figure 3. Reference input and model output of GA

As in Table 3, we see TLBO method is better in every way and the best answer is TLBO 19.17% better than GA. The dispersion of the GA answers is better than TLBO.

Unlike the GA, which requires the possibility of change rate intersection and changing the method of choice, TLBO requires no adapted parameter algorithm. This makes it easy to implement TLBO.

Table 3. Comparison of Two Methods

	Best	Intermediate	Worst	Variance	Deviation
TLBO	5.65 E-5	5.66E -5	5.73 E-5	6.22E-14	2.5E-14
GA	6.99 E-5	1.046 E-4	1.62 E-4	8.79E-10	2.98E-10

Another comparison is comparing the rate of answer convergence, that TLBO offers a better answer. Figure (4)

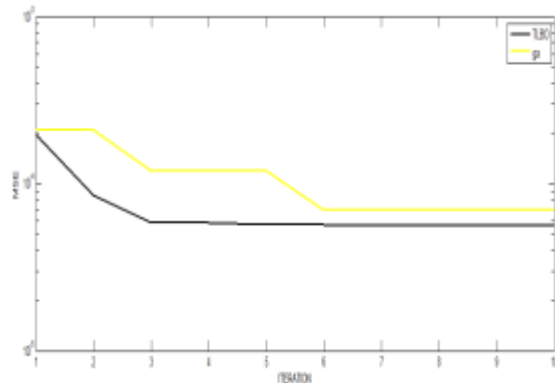


Figure 4. Compare the speed of convergence

CONCOLUSION

In this paper we tried to compare both TLBO and GA methods based on the best, middle, worst, answer variance and standard deviation and answer convergence rate. The results show that the TLBO method gives better answers because requires no adapted parameter algorithm. This makes it easy to implement TLBO.

REFERENCES

- [1] M.Moghadam Khaniki,(2006), "Fixative of Anticipator Controller For Discontinuous Polymerization Reactor" Master Of Engineering, Faculty Of Power Engineering And Robatic,Industrial University, shahrood, Iran.
- [2] S.M.Alavi,(2011), "Design Of Paralleled Distributed Resistant Phased Controller For a Continual Reactor Using Matrix Disparates" Master Of Engineering, Faculty Of Power Engineering And Robatic,Industrial University, shahrood, Iran.
- [3] P.Ngatchou, A.Zarei and M.A.El-Sharkawi, "Pareto Multi Objective Optimization",2005.
- [4] R.B.Gopaluni, I.Mizumoto, and S.L.Shah,"A Robust Nonlinear Adaptive Backstepping Controller for a CSTR", Ind. Eng. Chem. Res., vol 42, pp. 4628-4644,2003.
- [5] J.Gao and H. M.Budman, "Design of robust gain-scheduled PI controllers for nonlinear processes",Journal of Process Control, vol. 15 pp. 807–817,2005.
- [6] A.K.Chakrabarty and S.Bhattacharya,"Adaptive Control of a Plant with Time-varying Structured Uncertainties – A New Approach",IE (I) Journal-EL vol 90, pp.3-8,2010.
- [7] T.Zhang, M.Guay, "Adaptive control of uncertain continuously stirred tank reactors with unknown actuator nonlinearities", ISA Transactions vol. 44, pp. 55–68,2005.
- [8] D.L.Yua, T.K. Changa and D.W. Yub,"A stable self-learning PID control for multivariable time varying systems",Control Engineering Practice, vol. 15, pp. 1577–1587,2007.

AUTHOR BIBLIOGRAPHY

	<p>Alireza Baheri Department of Manufacturing, Collage Of Mechanics, Dezful Branch, Islamic Azad University, Dezful, Iran.</p>
	<p>Gholamhossien Fereyduni Department of Manufacturing, Collage Of Mechanics, Dezful Branch, Islamic Azad University, Dezful, Iran</p>
	<p>Milad Gorjian Department of Control, Collage Of Electronic, Najafabad Branch, Islamic Azad University, Najafabab, Iran</p>