View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Elsevier - Publisher Connector

Applied Mathematical Modelling 33 (2009) 3392–3401

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

Real-coded genetic algorithm for system identification and controller tuning

K. Valarmathi [a,*], D. Devaraj [a], T.K. Radhakrishnan [b]

[a] Arulmigu Kalasalingam College of Engineering, Krishnankoil, Virudhunagar District, Tamil Nadu 626 190, India
[b] Department of Chemical Engineering, National Institute of Technology, Trichy, Tamilnadu 620 015, India

ARTICLE INFO

Article history:
Received 15 August 2008
Received in revised form 29 October 2008
Accepted 3 November 2008
Available online 14 November 2008

Keywords:
System identification
PID controller
Real-coded genetic algorithm
pH process

ABSTRACT

This paper presents an application of real-coded genetic algorithm (RGA) for system identification and controller tuning in process plants. The genetic algorithm is applied sequentially for system identification and controller tuning. First GA is applied to identify the changes in system parameters. Once the process parameters are identified, the optimal controller parameters are identified using GA. In the proposed genetic algorithm, the optimization variables are represented as floating point numbers. Also, cross over and mutation operators that can directly deal with the floating point numbers are used. The proposed approach has been applied for system identification and controller tuning in nonlinear pH process. The simulation results show that the GA based approach is effective in identifying the parameters of the system and the nonlinearity at various operating points in the nonlinear system.

© 2008 Elsevier Inc. All rights reserved.

1. Introduction

The pH control finds wide applications in process industries. The pH process is a nonlinear dynamic system and an extremely complex and challenging control problem in process industries [1]. The extensive applications of the pH process in industry merit the study of control of these processes. Proportional integral (PI) controller has been widely used in pH process for many years. A PID controller improves the transient response of a system by reducing the overshoot, and by shortening the settling time of a system [2]. The nonlinearity of the pH process presents the need to tune the parameters of this controller. Also, the process nonlinearities and time dependent characteristics cause a significant change in the dynamic parameters of the process. In general, plant parameters change due to ageing of the plant or changes in the load [3]. The transient response will be worse if the plant dynamics change, which necessitates identification of the process model at different operating conditions so that controller design can be effected [4]. To effect this plant model is identified periodically and the changes in its dynamic characteristics are observed. This offers a great advantage over the conventional controller tuning methods, which uses the plant model at the nominal operating conditions.

In conventional identification methods, a model structure is selected and the parameters of the model are calculated by optimizing an objective function using an optimization technique. The selection of model structure is a compromise between model accuracy and simplicity. Auto regressive with exogenous inputs (ARX) is one of the simplest structures for system modeling. Fassois and Florakis [5] demonstrated that an auto regressive with moving average exogenous inputs (ARMAX) structure is a better choice than the ARX structure. But compared to ARX, ARMAX is more complicated.

* Corresponding author.
E-mail addresses: krvalarmathi@yahoo.co.in (K. Valarmathi), deva230@yahoo.com (D. Devaraj), radha@nitt.edu (T.K. Radhakrishnan).

0307-904X/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.apm.2008.11.006

The nonlinear models available for system identification include Hammerstein, Weiner model and nonlinear auto regressive with moving average with exogenous inputs (NARMAX). The conventional parameter identification methods namely least squares [6] and maximum likelihood method [7] often fail in the search for global optimum in the search space. Further, they require large set of input/output data from the system.

Traditionally, the PID controller parameters are evaluated using Ziegler–Nichols (ZN) [2] and Cohen Coon (CC) [8] methods. In both these methods, the parameters of the controller are obtained for an operating point where the model can be considered linear. The dynamic characteristics of most of the industrial processes exhibit nonlinear behavior and vary with time. This implies that there is sub-optimal tuning when a process operates outside the validity zone of the model. Internal model control (IMC) [9] overcomes the above said problem but its design calculations could be complicated for higher order process.

Heuristic search techniques like genetic algorithm overcome the difficulties and limitations encountered by the conventional approaches for system identification and controller tuning. Genetic algorithm [10,11] is a general-purpose optimization algorithm based on the mechanics of natural selection and genetics. Kristinsson and Dumont [12] proposed GA to identify plants with either minimum phase or non-minimum phase characteristic and un-modeled dynamics. Zibo and Naghdt [13] applied genetic algorithm to identify the parameters of the multi input and multi output (MIMO) system that is assumed to have an auto regressive with moving average exogenous (ARMAX) structure. Lu and Basar [14] presented the standard GA-based estimation scheme in a neural network framework, which ensure a good approximation for the system nonlinearity. Dangprasert and Avatchanakorn [15] employed GA for on-line parameter identification and controller tuning in load frequency control of a power system. In [16], the authors have proposed a GA-based design strategy for offline PI controller tuning in linear systems. Mwembeshi et al. [17] proposed GA-based internal model control (IMC) strategy for pH process.

In the traditional GAs, all the variables are encoded as binary digits forming a string. Then the genetic operators are applied to generate a new population. Such procedures are repeated until the optimal solution is reached. The binary-coded GA has Hamming cliff problems [18], which sometimes may cause difficulties in the case of coding continuous variables. To overcome the above difficulty this paper proposes a real-parameter genetic algorithm in which the optimization variables are represented as floating point numbers.

Further for effective genetic operation, crossover and mutation operators, which can directly deal with real variables, are used. The proposed approach has been applied to estimate the changes in the parameters of the system and to identify the optimal PID controller parameters in pH process.

2. pH process

The pH process is very important in many industrial applications. As shown in Fig. 1, acetic acid is fed to the reactor with a constant flow rate and sodium hydroxide is introduced to the reactor.

The mathematical model of the pH neutralization process proposed by McAvoy et al. [19] and reproduced below is used in this work to simulate the pH process.

$$V\frac{dx_a}{dt} = F_aC_a - (F_a + F_b)x_a \quad (1)$$

$$V\frac{dx_b}{dt} = F_bC_b - (F_a + F_b)x_b \quad (2)$$

$$[H^+]^3 + [H^+]^2\{K_a + x_b\} + [H^+]\{K_a(x_b - x_a) - K_w\} - K_wK_a = 0 \quad (3)$$

$$pH = -\log_{10}[H^+] \quad (4)$$

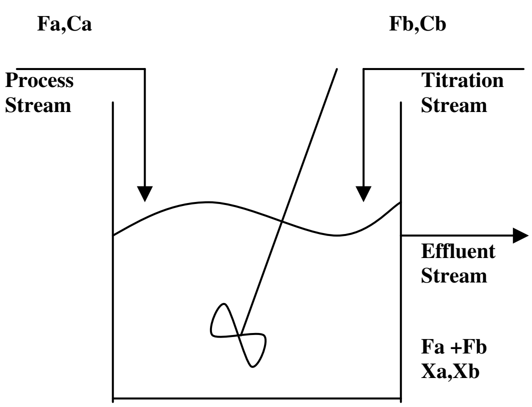

Fig. 1. Block diagram representation of pH process.

The pH variables used in this study are volume of the continuous stirred tank reactor (V), base flow rate of pH process (F_b) and concentration of acid (C_a) and concentration of base (C_b).

3. Parameter estimation and controller tuning

System identification and controller tuning are important issues in pH process. For system identification, the structure of the system is specified first and then the parameter of the model are estimated.

A weak-acid strong-base pH process may be approximated as

$$V\frac{dx_a}{dt} \approx F_a(C_a - x_a) \tag{5}$$

$$V\frac{dx_b}{dt} \approx F_b(C_b - x_b) \tag{6}$$

The acidic ionic concentration (x_a) will be approximately constant when the flow rate of the process stream (F_a) is a constant value. Hence, the titration curve is essentially stationary and the dynamics of the CSTR may be modeled as a linear system. F_b is the manipulated input variable and x_b is the controlled variable. The pH process can therefore be approximated by the Wiener-type nonlinear model, which is a linear system, followed by a static nonlinearity and is shown in Fig. 2. Although the manipulated variable F_b has minimal effect on the state x_a; the titration relation is still dependent on the acid concentration term, and so changes in the acid flow rate and acid concentration will act as load disturbances.

The Wiener structure is relatively easy to eliminate the static nonlinearity in such a model. The linearized process can then be placed under the control of linear controllers. Fig. 3 shows the architecture that may be used to control plants, which are described by Wiener-type nonlinear models. A linear system can be obtained by cascading the inverse of pH–x_b titration relationship. By rearranging Eq. (3), it may be deduced that the inverse titration curve has the following general structure.

$$x_b^* = \frac{1}{[H+]^2 + (d_1)[H^+]}\{-[H+]^3 - (d_1)[H+]^2 + (d_1x_a + d_2)[H+] + (d_1d_2)\} \tag{7}$$

Once the output pH value has been transformed into an estimate of the basic ion concentration (x_b), pH control can be achieved by employing a simple linear controller to regulate the state, x_b and to reject disturbances caused by changes in the acid flow rate and concentration. The variables $h(\cdot)$ and $h^{-1}(\cdot)$ are static nonlinearity and the inverse model of the nonlinearity, respectively.

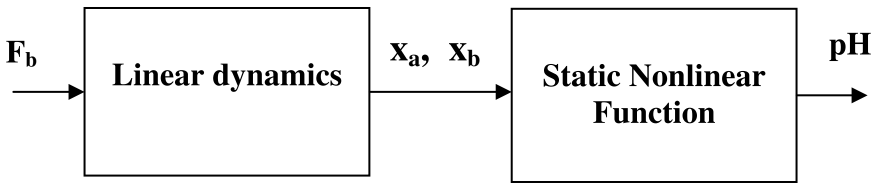

Fig. 2. Wiener-type nonlinear model.

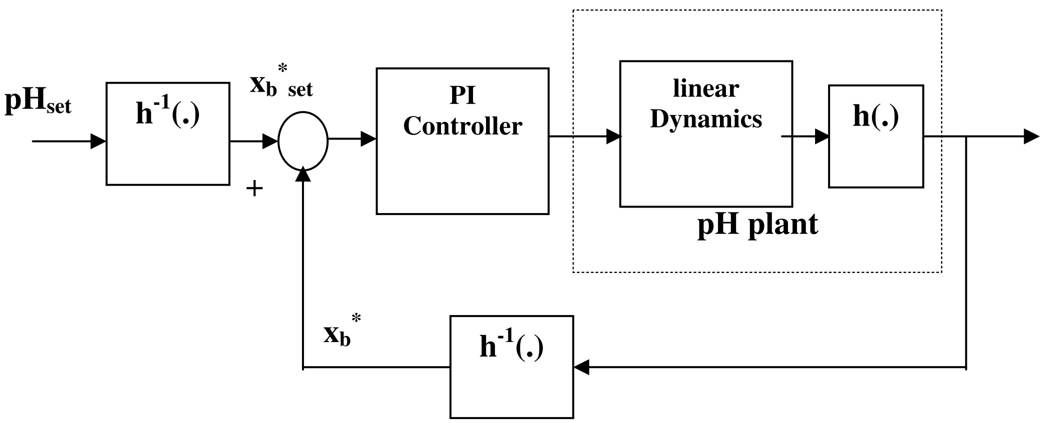

Fig. 3. Wiener-model control strategy.

The next step after system identification is the tuning of the PI controller whose transfer function is given by

$$G_c(s) = K_p + \frac{K_i}{s} \tag{8}$$

where K_p is the proportional gain and K_i is the integral gain. The performance of the closed loop system is defined by the performance criteria of integral square error for controller (ISE(cont)), over shoot (OS) and settling time (ST) of the transient response.

The integral square error squares the magnitude of error with respect to time. The overshoot is the difference between the maximum value of the output response and the steady state value and the settling time is the time for the response to stay within the specified percentage of its final value. In this work the problem of controller tuning is formulated as an optimization problem. The objective function of the controller is to minimize the integral square error, peak overshoot, rise time and settling time of the transient response.

$$F = w_1 F_{ISE_{(cont)}} + w_2 F_{OS} + w_3 F_{ST} \tag{9}$$

where w_1, w_2 and w_3 are the weight factors for the Integral square error, overshoot and settling time, respectively. The weight factors are varied in between 0 and 1 to get the optimal system response. Gradient-based conventional methods are not good enough to solve this problem and a global optimization technique like genetic algorithm is well suited for this kind of problems.

4. Proposed genetic algorithm

Genetic algorithm [10] is a general-purpose optimization algorithm based on the mechanics of natural selection and genetics. Unlike traditional hill-climbing methods involving iterative changes to a single solution, genetic algorithms work with a population of solutions. A fitness value, derived from the problem's objective function is assigned to each member of the population. Individuals that represent better solutions are awarded higher fitness values, thus enabling them to survive more generations. Starting with an initial random population, successive generations of populations are created by the genetic operators reproduction, crossover and mutation to yield better solutions, which approach the optimal solution to the problem. The GA repeats the above steps until the predetermined criteria are met.

Conventionally, binary strings are used to represent the decision variables of the optimization problem in the genetic population, irrespective of the nature of the decision variables. The binary-coded GA has number of difficulties in dealing with continuous search spaces. To overcome the above difficulty this paper proposes a real-parameter genetic algorithm in which the optimization variables are represented as floating point numbers.

The use of floating point numbers in the GA representation has a number of advantages over binary coding. The efficiency of the GA is increased as there is no need to convert the solution variables to the binary type, less memory is required, there is no loss in precision by discretization to binary or other values, and there is greater freedom to use different genetic operators.

With floating point representation, the evaluation procedure and reproduction operator remain the same as that in binary-coded GA, but crossover operation is done variable by variable. Also, the real parameter mutation operator, "uniform mutation", is used. These details are presented in the following subsections.

4.1. Reproduction

Reproduction is a method that stochastically selects the individuals from the population according to their fitness; the higher the fitness, the more chance an individual has to be selected for the next generation. There are three main types of selection methods: fitness proportionate selection, ranking method and tournament selection. Tournament selection [20–22] is used in this work. In tournament selection, '*n*' individuals are selected randomly from the population, and the best of the '*n*' is inserted into the new population for further genetic processing. This procedure is repeated until the mating pool is filled. Tournaments are often held between pairs of individuals, although larger tournaments can be used.

4.2. Crossover operation

The crossover operator is mainly responsible for the global search property of the GA. Crossover basically combines substructures of two parent chromosomes to produce new structures, with the selected probability typically in the range of 0.6–1.0. The Blend crossover operator (BLX-α) [18] is employed in this study.

Fig. 4 illustrates the BLX-α crossover operation for the one-dimensional case. In the BLX-α crossover the off spring y is sampled from the space $[e_1,e_2]$ as follows:

$$y = \begin{cases} e_1 + r \times (e_2 - e_1) : \text{if } u^{\min} \leqslant y \leqslant u^{\max} \\ \text{repeat sampling} : \ \text{otherwise} \end{cases} \tag{10}$$

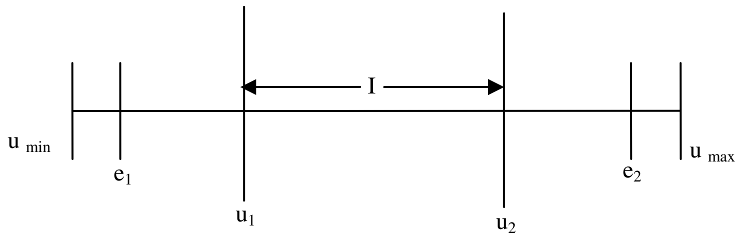

Fig. 4. Schematic representation of BLX-α crossover.

where

$$e_1 = u_1 - \alpha(u_2 - u_1) \quad (11)$$

$$e_2 = u_2 + \alpha \times (u_2 - u_1) \quad (12)$$

$$r: \text{ Uniform random number} \in [0, 1]$$

It is to be noted that e_1 and e_2 will lie between $u_{\min}$ and $u_{\max}$, the variable's lower and upper bound, respectively. In a number of test problems, it was observed that $\alpha = 0.5$ provides good results. One interesting feature of this type of crossover operator is that the created point depends on the location of both parents. If both parents are close to each other, the new point will also be close to the parents. On the other hand, if parents are far from each other, the search is more like a random search.

4.3. Mutation operation

After crossover is performed, mutation takes place. The mutation operator is used to inject new genetic material into the population. Mutation randomly alters a variable with a small probability. ''uniform mutation'' operator is applied in this work. In uniform mutation, the variable is set to a uniform random number between the variable's lower and upper limit.

5. Results and discussion

This section presents the details of the simulation carried out to estimate the system parameters and to tune the PI controller in pH process. The pH process was simulated in MATLAB Simulink. The software for the genetic algorithm was written in MATLAB and executed on a PC with 2.4 MHz and 256 MB RAM. The description of the simulated pH process is given in Table 1. Fig. 5 shows the titration curve of the experimental data obtained from experimental setup and simulated pH process using MATLAB Simulink.

5.1. Experimental setup

In the experimental setup, acetic acid is fed to the reactor with constant flow rate and sodium hydroxide is introduced to the reactor through the pump. Fig. 6 shows the pH process experimental setup. In the real time implementation, the dSPACE processor can be easily interfaced with Simulink and automatically convert the Simulink model into a targeted C code and download it to the designated hardware (dSPACE DS1102) via RTI. To read or write the internal variables of the control system, dSPACE Control Desk provides a user-friendly graphic user interface (GUI) environment that enables the user to observe vital data in the system.

In this paper, GA is applied to identify the parameters of the inverse titration equation and the linear controller in the Wiener-model control architecture. The inverse titration curve variables d_1 is vary between 1 and 14 and d_2 is between 0 and 1.

The optimal values of model parameters obtained by the RGA based algorithm for the Weiner model are given in Table 2. For comparison, GA based Weiner model parameters is also given in Table 2. From this table, it can be seen that RGA has resulted in more accurate estimation of model parameters and MSE is minimum than the GA based Weiner model. The

Table 1
Description of the pH process.

Description	Symbol	Value
Volume of the continuous stirred tank reactor	V	7.4 l
Flow rate of the influent stream	F_a	0.24 l min^{-1}
Flow rate of the titrating stream	F_b	0–0.80 l min^{-1}
Concentration of the influent stream	C_a	0.2 g mol l^{-1}
Concentration of the titrating stream	C_b	0.1 g mol 1^{-1}

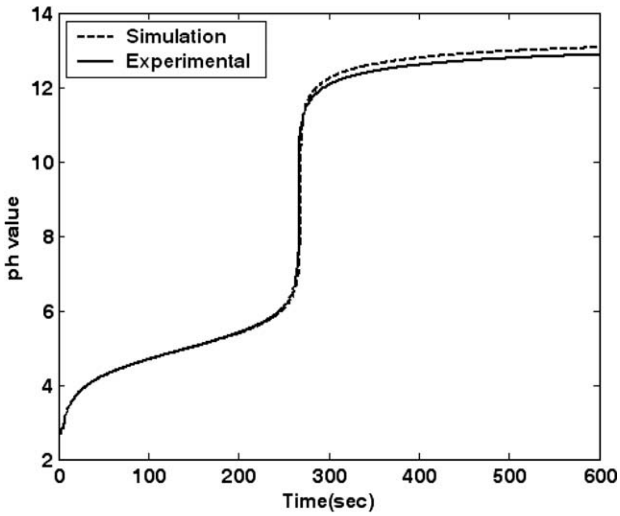

Fig. 5. Titration curve of pH process.

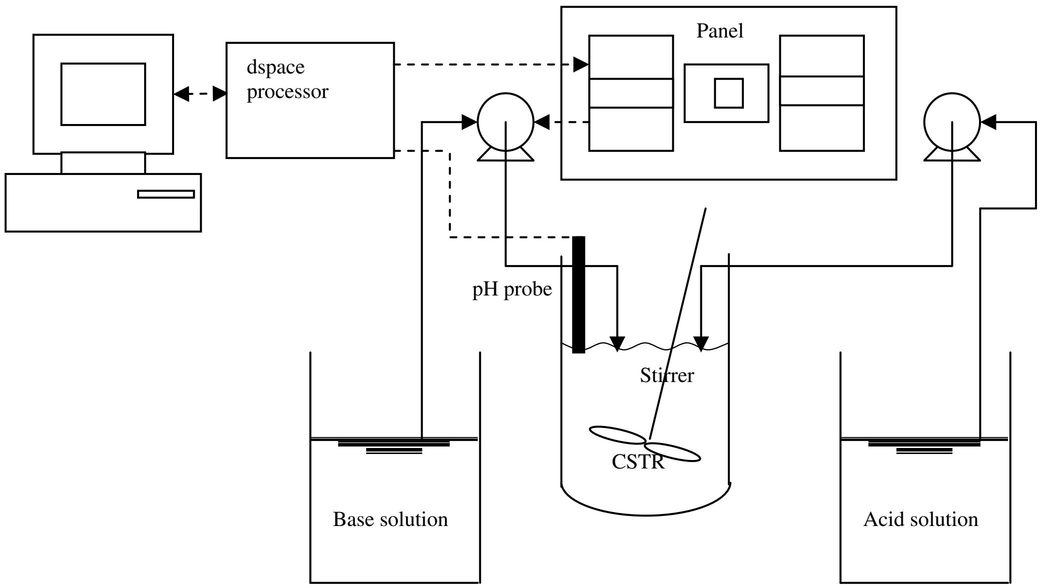

Fig. 6. Experimental setup.

Table 2
Estimated parameters for the pH process.

Type	d_1	d_2	MSE	CT
Binary-coded GA	6.8641	0.225	$2.868e^{-1}$	$7.1506e^{2}$
Real-coded GA	6.505	0.206	$2.844e^{-4}$	$3.4967e^{2}$

inverse titration curve estimated by the GA is shown in Fig. 7. From this figure it is found that the real-coded GA is able to find a solution even though the relationship between the input and output data is highly nonlinear.

Next the closed loop PI controller which is embedded in the Wiener-model control architecture is tuned for the optimal values of K_p and K_i using proposed GA algorithm. The boundaries of the optimization variables are taken as $0.1 < K_p < 50$; $0.1 < K_i < 10$. Initially boundaries are randomly selected. The best results of the real-coded GA are obtained with the following control parameters.

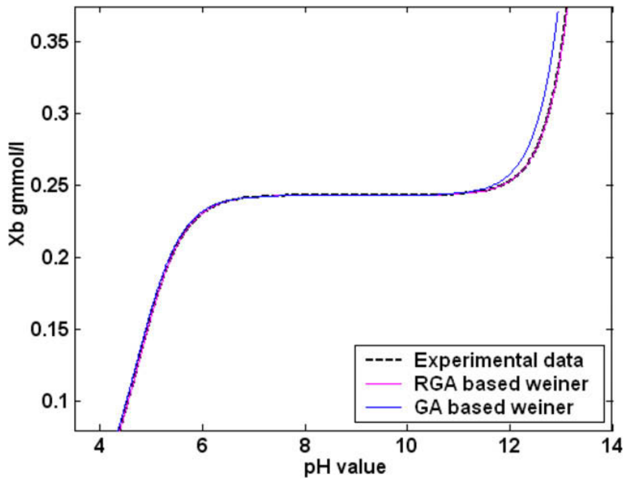

Fig. 7. Inverse titration curve estimated by GA.

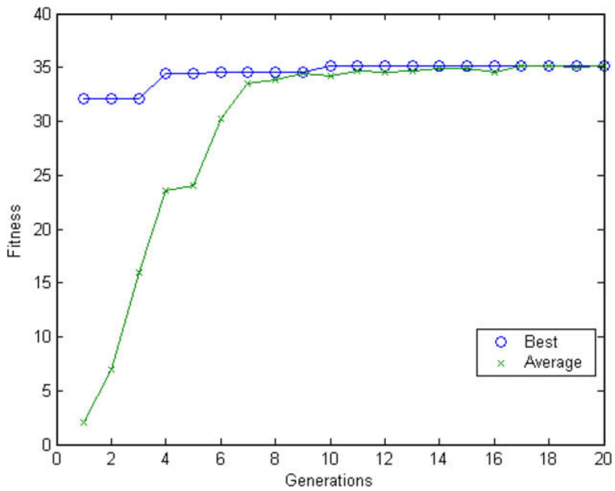

Fig. 8. Convergence of proposed GA.

Table 3
Comparison of performance analysis.

SP	Tuning	K_p	K_i	ISE	OS	ST (s)	PT (s)	RT (s)	CT (s)
5	ZN	1.2	1.149	0.1996	3.97	25	1	1	11
	IMC	0.071	0.124	0.1510	4.16	20	1	1	10
	GA-Weiner	92.64	40	0.0139	3.85	4	1	1	32
	RGA-Weiner	84.31	5	0.0518	0.326	3	1	1	28
7	ZN	0.854	0.581	2.987	4.13	189	15	1	10
	IMC	0.071	0.124	0.4985	14	68	14	1	12
	GA-Weiner	62.64	35	0.067	1.81	35	1	1	42
	RGA-Weiner	79.01	40	0.0422	0.1967	25	1	1	36
9	ZN	0.111	0.983	14.47	3.44	232	19	33	16
	IMC	0.071	0.124	0.2914	4.5	55	4	33	14
	GA-Weiner	65.78	4.5	0.08	0.32	40	4	15	53
	RGA-Weiner	80.1	39.7	0.0376	0.2693	35	3	13	42
11	ZN	0.143	0.163	48.62	15.6	210	92	86	12
	IMC	0.071	0.124	3.2960	10.256	70	25	20	15
	GA-Weiner	92.6	40	0.04502	0.220	25	3	18	62
	RGA-Weiner	92.647	40.12	0.0442	0.3373	23	3	16	50

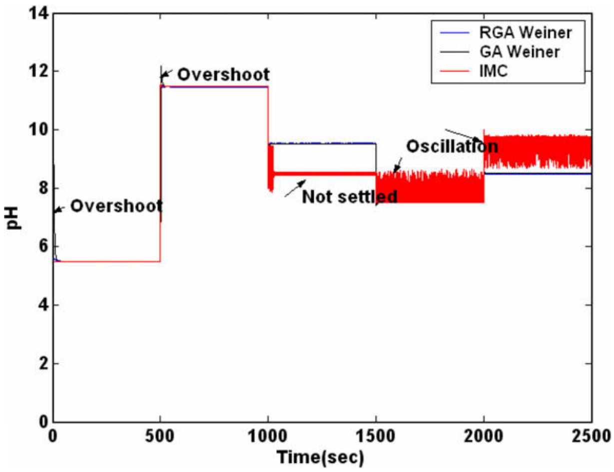

Fig. 9. pH value for various set-point tracking.

Number of generations: 20.
Population size: 20.
Crossover probability: 0.8.
Mutation probability: 0.08.

Fig. 8 shows the convergence characteristics of proposed GA algorithm. It is observed that the variation of the fitness during the GA run for the best case and shows the generation of optimal variables. It can be seen that the fitness value increases rapidly in the first three generations of the GA. During this stage, the GA concentrates mainly on finding feasible solutions to the problem. Then the value increases slowly, and settles down near the optimum value with most of the individuals in the population reaching that point.

The optimal control gains obtained by the proposed algorithm along with the ISE and the system performance indices are given in Table 3. It also gives the values and the performance indices obtained using the ZN, IMC and GA. The performance of the system is found to be satisfactory with the control gains obtained using the proposed GA. From the Table, it is found that the proposed GA in minimum ISE, minimum peak overshoot and minimum settling time. Also, the computation time requirement is less in proposed GA.

Fig. 9 shows that the Wiener-model controller, whose parameters are identified using proposed GA, is able to control the pH level over a wide range of set-point tracking. The overshoot following a set-point change is almost zero which indicates

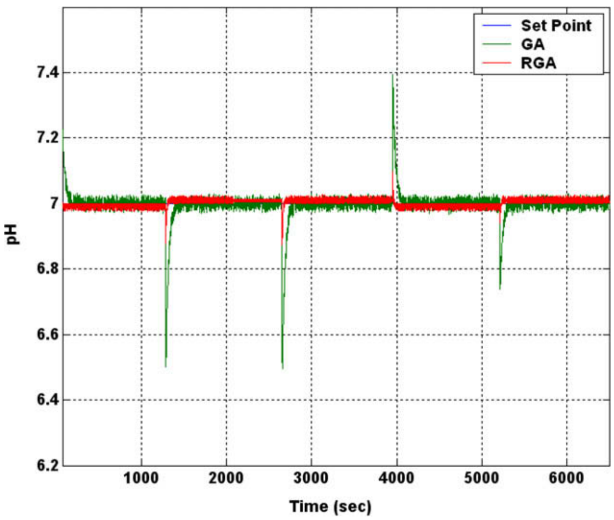

Fig. 10. Tracking of pH value at seven in the presence of acid flow rate disturbance.

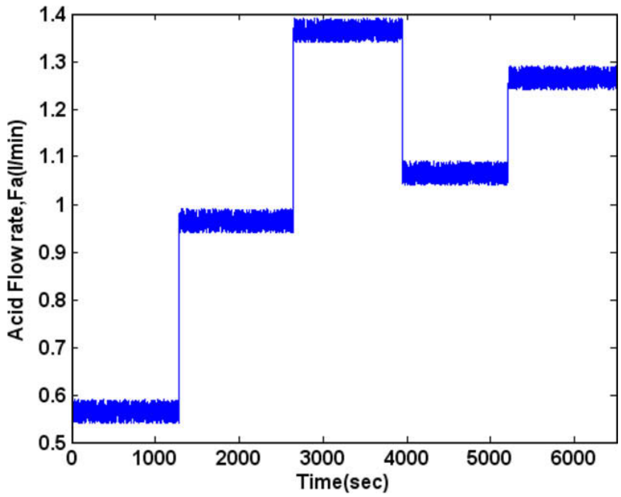

Fig. 11. Variation in acid flow rate.

that the RGA has successfully minimized the peak overshoots. Also, the proposed GA has no oscillations and minimum settling time in the nonlinear region 7–9.

Proposed GA based Weiner controller is able to reject the load disturbance taking place at feed flow rate. Fig. 10 demonstrates that the PI controller evolved by the RGA is able to reject the load disturbances. Fig. 11 shows the load disturbances in acid flow rate is changed from the nominal value to 0.6,1, 1.4, 1.1 and 1.3 l/min, respectively at 20, 40, 60 and 85 min. From the figures, it is found that the controller is able to maintain the pH value of the effluent stream at the neutral value of set-point seven in the presence of load disturbances. This characteristic is important in waste-water treatment, where disturbances should not cause the pH value of the effluent stream to deviate too much from the set-point.

6. Conclusion

In this paper, real-coded genetic algorithm has been applied for identifying the parameters and obtaining the optimal PID controller variables in the process plants. In the proposed approach, the optimization variables are represented as floating point numbers in the genetic population, and the crossover and mutation operators which can directly deal with floating point numbers are used. The proposed GA is applied to identifying the parameters of the Wiener-model in pH process. Also computing the optimal values of PID controller parameters has been analyzed. The simulation result shows that the GA is able to tune the PID controller satisfactorily and able to regulate the set-point tracking with minimal overshoot and fast rise time in all the cases. Load disturbances in acid flow rate in pH process also rejected by the proposed control methodology. Further the proposed algorithm takes less time for convergence compared to the conventional binary-coded GA.

References

[1] B.D. Kulkarni, P.B. Deshpande, Nonlinear pH control, Chem. Eng. Sci. 46 (1991) 995–1003.
[2] K.J. Astrom, T. Hagglund, The future of PID control, Control Eng. Pract. 9 (11) (2001) 1163–1175.
[3] T.S. Schei, Automatic tuning of PID controllers based on transfer function estimation, Automatica 30 (12) (1994) 1983–1989.
[4] A. Florakis, S.D. Fassosis, MIMO LMS–ARMAX identification of vibrating structures. Part II: A critical assessment, Mech. Syst. Signal Process. 15 (4) (2001) 737–758.
[5] S. Dionisio, O.P. Pinto, Genetic algorithm based system identification and PID tuning for optimum adaptive control, in: International Conference on Advanced Intelligent Mechatronics, California, 2005, pp. 801–806.
[6] B. Rad, W.L. Lo, Self-tuning PID controller using Newton–Raphson search method, Trans. Ind. Electron. 44 (5) (1997) 717–725.
[7] L. Ljung, System Identification – Theory for the User, second ed., PTR Prentice Hall, Upper Saddle River, NJ, 1999.
[8] F.G. Shinskey, Process Control System: Application, fourth ed., Design and Tuning, McGraw-Hill, 1996.
[9] M. Morari, E. Zufiriou, Robust Process Control, Prentice-Hall Inc., Englewood cliffs, NJ, 1987.
[10] D.E. Goldberg, Genetic Algorithms in Search Optimization and Machine Learning, Addison Wesley, 1989.
[11] K.M. Passino, Towards bridging the perceived gap between conventional and intelligent control, in: Intelligent Control: Theory and applications, IEEE Press, Piscataway, NJ, 1996, pp. 1–27.
[12] K. Kristinsson, G.A. Dumont, System identification and control using genetic algorithms, IEEE Trans. Syst. Man Cyb. 22 (5) (1992) 1033–1046.
[13] Z. Zibo, F. Naghdt, Application of genetic algorithms to system identification, IEEE Int. Conf. Evolut. Comput. 2 (1995) 777–782.
[14] S. Lu, T. Basar, Genetic algorithms-based identification, IEEE Inter. Conf. Syst. Man Cyb. 1 (1995) 22–25.
[15] P. Dangprasert, V. Avatchanakorn, Genetic algorithms based self-tuning regulator, IEEE Int. Conf. Evolut. Comput. 1 (1995) 444–449.
[16] C. Vlachos, D. Williams, J.B. Gomm, Genetic approach to decentralized PI controller for multivariable processes, Proc. IEE Control Theory Appl. 146 (1) (1999) 58–64.
[17] M.M. Mwembeshi, C.A. Kent, A genetic algorithm based approach to intelligent modelling and control of pH in reactors, Comput. Chem. Eng. 28 (9) (2004) 1743–1757.

[18] D. Devaraj, B. Yegnanarayana, Genetic algorithm-based optimal power flow for security enhancement, IEE Proc. Gener. Trans. Distrib. 152 (6) (2005) 899–905.
[19] T.J. McAvoy, E. Hsu, Dynamics of pH in controlled stirred tank reactor', Ind. Eng. Chem. Process Des. Dev. 68 (1972) 114–120.
[20] F. Herrera, M. Lozano, A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study, Int. J. Intell. Syst. 18 (3) (2003) 309–338.
[21] L.J. Eschelman, J.D. Schaffer, Real-coded Genetic Algorithms and Interval-Schemata Foundations of Genetic Algorithms 2, Morgan Kaufman Publishers, San Mateo, 1993.
[22] W. Chang, Nonlinear system identification and control using a real-coded genetic algorithm', Appl. Math. Model. 31 (2007) 541–550.