Third International Conference on
Advances in Control and Optimization of Dynamical Systems
March 13-15, 2014. Kanpur, India

Learning Automata based Set-point weighted parameter for unstable systems

Lalit Musmade*, M. Chidambaram **

**Department of Chemical Engineering, Indian Institute of Technology Madras, Chennai, India (E-mail: lalitmusmade2@gmail.com).*
***Department of Chemical Engineering, Indian Institute of Technology Madras, Chennai, India (E-mail: chidam@iitm.ac.in).*

Abstract: A simple method using Learning Automata is given for selection of the set point weighting parameter (b) in PI controllers for unstable first-order plus time delay systems.The parameter (b) is selected, based on the state transition probabilities to reduce the over shoot for a servo problem. The performance evaluation of the control action is carried out by using the reinforcement scheme based on reward-penalty and accordingly state transition probabilities are updated. The learning automata consider the updated state transition probabilities as input and tuning parameter (b) as output. The responses of PI controllers without and with the set point weighting are compared.

Keywords: Learning Automata, Set point weighting, PI control, Adaptive control.

1. INTRODUCTION

Control of unstable systems is carried out conventionally by PI/PID controller because of its versatility, reliability and ease of operations. In industries, the skilled operators observe the control responses and controller parameters are adjusted based on the experiences of the operators. The controller settings can also be calculated based on the transfer function model of the system. However, if the process is not linear or if process gain and time constant are time-varying, the control performance will not be satisfactory. It is therefore highly desirable that the control system be able to learn to improve its performance on the basis of the response of the environment in which it operates. Learning automata can be used advantageously for problems where nonlinearities and uncertainties predominate (Narendra and Thathachar, 1989; Najim and Poznyak, 1994) and function optimization. (Verbeeck & Nowe, 2002; Wu & Liao, 2013).

Learning automata (Figure 1) comprises of two main building blocks, a *learning automaton* and l*earning algorithms*. The Learning automaton operates by selecting an action from a finite number of actions and interacts with random environment. Once a learning automaton has the response from environment, it uses learning algorithms to select next control action. By this process, the automaton learns asymptotically to select the optimal action. Learning automata(Mars et al. 1996) can be described by quintuple, $LA= \{\alpha, \beta, p, T, c\}$ where $\alpha(=\{\alpha_1,\alpha_2,\ldots,\alpha_r\})$ is set of outputs of automaton, $\beta(=\{\beta_1,\beta_2,\ldots,\beta_r\})$ is set of inputs to the automaton,$p(=\{p_1,p_2,\ldots,p_r\})$ is the probability vector, $T=p(n+1)=(T[\alpha(n),\beta(n),p(n)])$ is the learning algorithm or reinforcement scheme,$c(=\{c_1,c_2,\ldots,c_r\})$is set of penalty probabilities defining the environment. Based on the environment response and the action selected by the automaton at time t, reinforcement scheme generates $p_i(t+1)$ form $p_i(t)$ where $p_i(t+1)=$ probability of action i at iteration $(t+1)$ and $p_i(t)=$ probability of action i at iteration (t).

Howell and Best (2000) have used Continuous Action Reinforcement Learning Automata (CARLA) to select a single set of optimized PID settings to minimize the specified cost criterion. The PID controller parameters are initially set with wide search ranges, $\pm 200\%$ of the Zeigler-Nichols settings. Three separate learning automata are used (one for each controller parameter) to search the parameter space to minimize the cost criteria.These fixed set of PID parameters are used for simulation for a regulatory problem and the response is compared with that of the Zeigler-Nichols settings.The performance for the load disturbance is shown to be better for the proposed method. On-line learning and application to Ford Zetec 1.81 engine is also carried out real time. In the experimental study, the PID settings are selected by the learning automata at every sampling instant.

Recently, Lalit and Chidambaram (2013) presented, learning type PI control system using learning automata, which can adjust the control parameters for the regulation of a non-linear bioreactor. A tuning parameter α (appearing in the re-parameterization of Ziegler-Nichols tuning formulae) is selected based on the state transition probabilities. The value of α is used to tune the PI controller parameter K_C (process gain) and calculate the value of manipulated variable. The performance evaluation of the control action is carried out by using the reinforcement scheme based on reward-penalty and accordingly state transition probabilities are updated. Learning automata based self-tuning of a PI controller gives superior servo and regulatory performances compared to that of fixed parameter PI controller for nonlinear bioreactor model equations.

Many unstable systems are adequately represented by a first-order plus time delay (FOPTD) transfer function model for the purpose of controller design. Unstable systems exhibit multiple steady states due to inherent nonlinearity in the

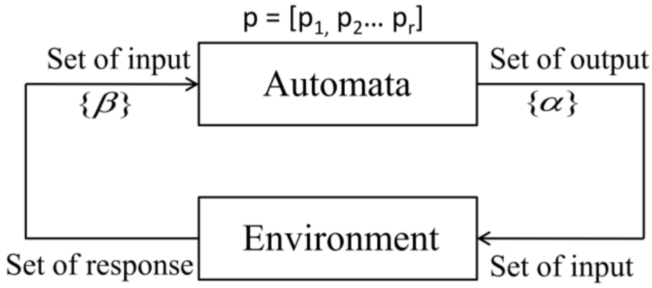

Figure 1 Learning Automata

systems. It may be desirable to operate a system at the unstable steady state for economic or safety reasons. Some of the reported methods for designing PI/PID controllers include two degrees of freedom method (Jacob, 1996; Park *et al*, 1998; Majhi & Atherton, 2000; Sul *et al*, 1999), pole placement method (Valentine and Chidambaram, 1997), optimization method (Visioli, 2001; Manoj & Chidambaram, 2001). However, all the methods give excessive overshoot, particularly for the servo problem. The use of a set point weighting parameter in PI/PID control law has been suggested (Astrom and Hagglund, 1995) for minimizing the overshoot in the case of stable systems. The modified form of PID control law is given as:

$$u(t) = k_C\left\{e_p + \frac{1}{\tau_I}\int_0^t e\,dt - \tau_D\left(\frac{dy}{dt}\right)\right\} \tag{1}$$

where,

$$e = y_r - y \tag{2}$$

$$e_p = by_r - y \tag{3}$$

Here y is the system output, y_r is the set point for y and b is the set point weighting parameter. The exact error is used for integral calculation in equation (1) ensures that the offset is zero. The value of the parameter b varies from 0 to1 and different methods are proposed to select this value. Since for unstable systems with time delay, the overshoot is larger than that of the stable systems, there is a need for a simple method to calculate the parameter b. For unstable systems several studies (Prashanti and Chidambaram, 2000; Padma Sree and Chidambaram, 2005; Chen, Huang, and Liaw, 2008; Rajinikanth and Latha, 2012) for set point weighting are reported. Usually values close to 0.3 are used.

The main focus of the present work is to apply the learning automata to select the set point weighting parameter (b) in PI controllers to reduce the overshoot for unstable first-order plus time delay systems. At each sampling instant based on the performance evaluation, the learning automata select the parameter b and is used to calculate value of e_p which in turn is used to calculate controller output u in (1). Section 2, describes use of learning automata to select set point weighting parameter (b). Section 3 gives performance comparisons of simulation study carried out using learning automata for set point weighting for PI controller and PI controller without set point weighting.

2. LEARNING AUTOMATA TUNER

General steps involved in implementation of learning automata (Najim and Poznyak, 1994) are as follows:

1. The set-point weighted parameter b is divided into a set of 10 discrete intervals (N = 10, $b_{\min} = 0.1$, $b_{\max} = 0.5$)

2. If no prior information is available, probability of selecting a particular b is taken equal (1/N).

3. One of the values of b is selected based on the generation of random number f ($f \in [0, 1]$). The algorithm chooses b such that the cumulative probability associated with each of the b equals to or greater than the generated random number that verifies the following constraint:

$$\sum_{j=1}^{i} p_j \ge f \tag{4}$$

4. For each selected value of b, value of e_p is calculated from equation (3) which in turn is used to calculate the value of controller output (u) by equation (1). The selected control action is applied and kept constant for a complete sampling period.

5. The performance of control action is measured by measuring the system output (y).The reinforcement scheme is used to update the action probabilities and steps from 1 to 4 are again repeated.

Following information is used to calculate reward or penalty:

$$\beta(t) = 0(\text{reward})\text{ if }\begin{cases} y(t) \le \text{desired value and} \\ \quad b(t) \le b(t-1) \\ \text{or} \\ y(t) \ge \text{desired value and} \\ \quad b(t) \ge b(t-1) \end{cases} \tag{5}$$

$$= 1(\text{penalty})\text{otherwise} \tag{6}$$

The reinforcement scheme used is as follows:

If action leads to reward (β =0):

$$p_i(t+1) = p_i(t) + \beta_0 p_i(t)[1 - p_i(t)] \tag{7}$$

$$p_j(t+1) = p_j(t) - \beta_0 p_i(t) p_j(t)$$

where $(j = 1,2,\ldots N;\ j \ne i)$

If action leads to penalty (β=1):

$$p_i(t+1) = p_i(t) - \beta_1 p_i(t)[1 - p_i(t)] \tag{8}$$

$$p_j(t+1) = p_j(t) + \beta_1 p_i(t) p_j(t)$$

where $(j = 1,2,\ldots N;\ j \ne i)$

The parameters β_0 and β_1 are varied in the range $0 < \beta_0 \le 1$ and $0 \le \beta_1 \le 1$.