16

Model-Free Learning Control of Chemical Processes

S. Syafiie[1], F. Tadeo[1] and E. Martinez[2],
[1]*Department of Systems Engineering and Automatic Control, University of Valladolid,*
[2] *Consejo Nacional de Investigaciones Científicas y Técnicas,*
[1]*Spain,* [2]*Argentina.*

1. Introduction

Learning is the nature for human being. For example, a school-student learns a subject by doing exercise and home-work. Then, a school-teacher grades the school-student's works. From this student and teacher interaction, the ability of the student mastering the subject is a feedback that the previous teaching method is successful or failure. As a result, the teacher will change the teaching method to improve the student ability for mastering the subject. This is a picture that the reinforcement learning (RL) agent learns the environment.

Process control mainly focuses on controlling variable such as pressure, level, flow, temperature, pH, level in the process industries. However, the methodologies and principles are the same as in all control fields. The early successful application control strategy in process control is in evolution of the PID controller and Ziegler-Nichols tuning method (Ziegler and Nichols, 1942). Till nowadays, 95% of the controllers implemented in the process industries are PID-type (Chidambaram and See, 2002). However, as (i) the industrial demands (ii) the computational capabilities of controllers and (iii) complexity of systems under control increase, so the challenge is to implement advanced control algorithms.

There have been commercial successes of the intelligent control methods, but the dominating controller in process industries is still by far the PID-controller (Chidambaram and See, 2002). This stands to the fact that a simple and general purpose automatic controller (for example PID) is demanded in process industries. Therefore, designing advanced controllers are to address the industrial user demand. This is the reason that a learning method called model-free learning control (MFLC) is introduced. The MFLC algorithm is based on a well known Q-learning algorithm (Watkins, 1989).

Successful applications of RL are well documented in the recent literature, including learning to control mobile robots (Bucak and Zohdy, 2001), sustained inverted flight on an autonomous helicopter (Ng *et al.*, 2004), and learning to minimize average wait time in elevators (Crites and Barto, 1996). However, only few articles can be found regarding RL applications for process control: multi-step actions based on RL was fruitfully applied for thermostat control (Schoknecht and Riedmiller, 2003), and one of the authors successfully applied RL for modeling for optimization in bath reactors by making the most effective use of cumulative data and an approximate model (Martinez, 2000). The reason for the difference between robotics and process control is possibly the nature of the control task in

Source: Reinforcement Learning: Theory and Applications, Book edited by Cornelius Weber, Mark Elshaw and Norbert Michael Mayer
ISBN 978-3-902613-14-1, pp.424, January 2008, I-Tech Education and Publishing, Vienna, Austria

each field: typically in robotics the degrees of freedom for control are significantly high whereas in process plants are much more constrained. However with the shift from regulation to optimization and supervisory control the area is entering into a set of problems where RL can become the alternative choice.

This chapter discusses novel, yet simple to implement learning system in process control based on RL algorithms. As the ability to store and process large amounts of data in computer's memory and processor increases by time, this ability has made feasible the use of learning methods in systems for business, scientific and engineering, and medical decision-making. The proposed MFLC is mainly for nonlinear, complex, and time-varying chemical processes for which the development of a first-principles model is too costly in terms of time and money. The state-action space is defined using a symbolic representation and control incremental constraints. The state space is based on length errors of the system regarding a goal state. In this chapter, the MFLC approach is discussed for process control.

This proposed technique is then tested on two laboratory plants: pH control and oxidation plants. Industrial pH control has received considerable attentions in literature (see Kalafatis *et al.,* 2005 and references therein). However, as the inherent characteristics (time-varying, nonlinear and buffer capacity) of pH process dynamic are extremely difficult to model and predict in wastewater treatment plant, then a general purpose control strategy is a very challenging problem. As result most wastewater treatment plant uses on-off pH control.

The issues are more complicated when oxidation reduction potential (ORP) is used to guarantee on-specification discharge by regulating the residence time. The ORP sensor measures the presence oxidizer or reducer in the solution and not the concentration of a given chemical species (McPherson, 1993). Many researchers find some processes are near optimal in certain ORP values (Peng *et al.*, 2002; Baeza *et al.*, 2000; Kwan, 2005). Clearly, it is a challenge to use ORP sensor for controlling the load to a wastewater oxidation process.

This chapter is organized as follows: a MFLC algorithm for designing controller for chemical process control is given in section 2. In section 3, the application for a simulated buffer tank control is discussed. Laboratory online applications are discussed in section 4 for pH and ORP control processes.

2. MFLC algorithms

From the different proposed RL algorithms (Sutton and Barto, 1998), this paper proposes a Model-Free Learning Control (MFLC) where the basic Q-learning algorithm is combined with symbolic states which are frequently visited to address process control problems related to wastewater oxidation plants. The resulting value function, which is a mapping of history of visited states and executed actions to cumulative rewards, gives a clue for the learning controller to select an action in a given state. Through this function, the agent takes into account that taking an action in the current state will provide a given cumulative future reward derived from the control task at hand. This predicted value is used by the controller policy for selecting an action from those available in each visited symbolic state. This MFLC can be seen in Figure 1. The value of the reinforcement at each time reflects the control task objectives (Sutton and Barto, 1998), in process control problem it is proposed to involve control energy costs and error tolerance. The "situation" block is used to generate the symbolic state from plant readings and control task specification.

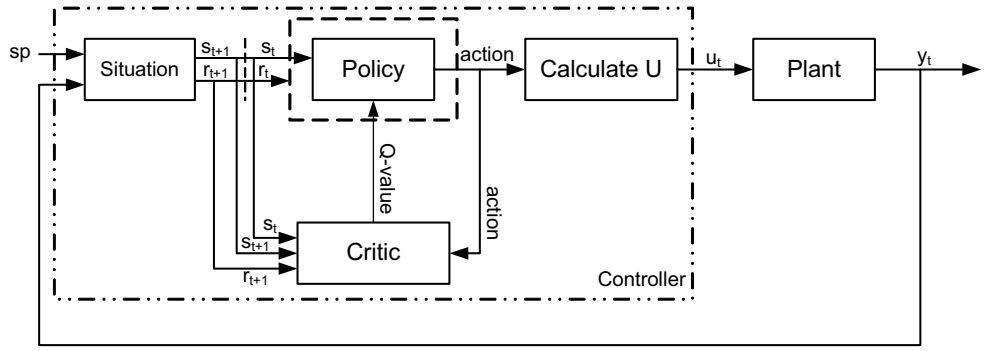


Fig. 1. MFLC architecture based on Q-Learning

A central part of RL algorithms is the estimation of the so-called *Q-function*, which gives the benefit of applying action a_t when the system is in state s_t. This function is denoted by $Q(s_t, a_t)$. To learn this Q-function it is necessary to take into account the benefit now and in the future: when action a_t has been selected and applied to the environment, the system moves to a new state, s_{t+1}, and receives a reinforcement signal, r_{t+1}; The value function for state-action pairs, $Q(s_t, a_t)$, is updated by the basic learning rule:

$$Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) + \alpha_t \left[r_{t+1} + \gamma \max_{b \in A_{s_{t+1}}} Q_\pi(s_{t+1}, b) - Q_\pi(s_t, a_t) \right] \tag{1}$$

where:

- $A_{s_{t+1}}$ is the set of possible actions in the next symbolic state.
- The *learning rate*, $0 \le \alpha \le 1$, is a tuning parameter, that can be used to optimize the speed of learning (Although too small learning rates might induce slow learning, while too large learning rates might induce oscillations).
- The *discount factor*, γ, is used to weight near term reinforcements more heavily than distant future reinforcements: If γ is small, the agent learns to behave only for short-term reward; the closer γ is to 1 the greater the weight assigned to long-term reinforcements.

2.1 MFLC state-action space

A central issue in Reinforcement Learning algorithms is the definition of the states. In MFLC the states are defined based on the control objective and control constraints, as follows:
In a SISO implementation of the MFLC approach, the control task is defined as the ability to achieve and maintain a given process variable inside a specification band *r-d* and *r+d*, as shown in Figure 2. The width of this band is defined based on the tolerance of the system (which depends on measurement noise, disturbances and systems specification) and referred to as the *goal band*, and corresponds to the *goal state*, where the learning control system operates (it is now assumed, without loss of generality, that it is exactly in the middle of the working range).

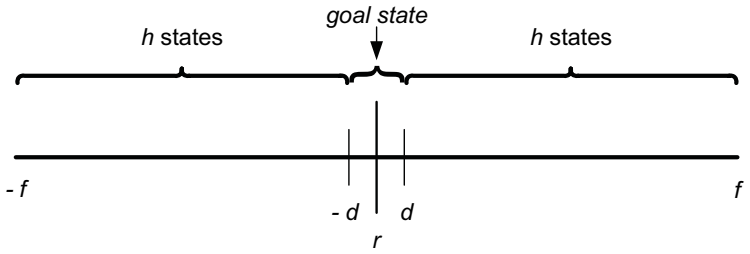


Fig. 2. Symbolic states definition in MFLC

To describe the rest of the symbolic states, it is considered that the process may be in h states from the goal state to the maximum positive or negative error of the system, f (Selecting h is a trade-off: this number must be large enough to describe all the different behaviours of the process, but small enough to reduce learning time and the size of the Q-value matrix).
If needed, the "length" of each state can be calculated as follows:

$$c = \frac{f - d}{h} \tag{2}$$

Thus, the positive bound parameter can be defined as:

$$\omega_i = d + (i-1)c, \quad i \in [1, \cdots, h] \tag{3}$$

(For negative errors, the bound parameter is trivial by changing signs).
Thus, the vector of symbolic states can be represented as follow:

$$g_j = \begin{cases} e - \omega_j & \text{if } e > \omega_j \\ \omega_j - e & \text{else} \end{cases}, \quad j \in [1, \cdots, 2h+1], \tag{4}$$

where e is the tracking error. The symbolic current state s_t, is just:

$$s_t = \arg\max(g_j) \tag{5}$$

In MFLC, the control signal u_t is calculated by varying the previous control signal in a magnitude calculated from the difference of the numerical values of the selected optimal action, a_t, with respect to the wait action, a_w (action corresponding to maintaining the previous control signal). That is,

$$u_t = u_{t-1} + k(a_w - a_t) \tag{6}$$

This gives a PI-like structure, which simplifies initialization and tuning for the end user (k is the tuning parameter defining the aggresiviness of the controller). At each state there is only a finite set of possible actions (see Figure 3). These actions are selected based on the systems description: in particular from the limitations on the minimum and maximum variations of the control signal, as follows:
Let the incremental control be bounded as:

$$\underline{\Delta u} \le \Delta u \le \overline{\Delta u}. \tag{7}$$

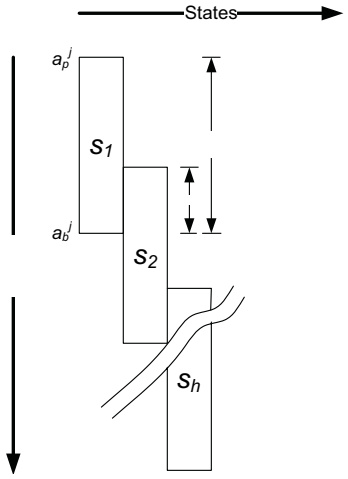


Fig. 3. Definition of the action set

The number of total actions needed to satisfy the input constraints can be calculated as follows:

$$N_a = round\left(\frac{\overline{\Delta u} - \underline{\Delta u}}{k}\right) + 1. \tag{8}$$

From (7) and (8), the value corresponding to the wait action a_w, can be calculated as follows:

$$a_w = 1 + round\left(\frac{\overline{\Delta u}}{k}\right). \tag{9}$$

As expected, not all the actions may be available at each state: only a constrained set of the actions is available depending of the symbolic state, e.g. if the error is very small, the only actions available are those that correct a small error.
The number of action in each state is

$$N_a^s = \left(\frac{N_a}{h}\right)\rho, \tag{10}$$

where ρ is a parameter that gives the degree of overlapping with neighboring states (selected such that N_a^s is integer. Then, the available actions for every state ranges from a_p^j to a_b^j (except in the goal state, where only the wait action can be selected). The idea is presented in Figure 3. Those available actions can be calculated as

$$\begin{aligned} a_p^j &= a_p + (j-1)v, \\ a_b^j &= a_p^j + N_a^s - 1, \end{aligned} \tag{11}$$

where

$$v = \frac{N_a^h h - N_a}{h - 1} - 1 \tag{12}$$

So far, the SISO implementation has been presented. For MIMO system the simplest methodology would be used several learning controllers that interact between them.
Next section discusses the application of the proposed MFLC for a buffer tank control. The application is to maintain smoothly the out flow of the tank and to keep the level of the tank to avoid overflow and empty.

3. Buffer tank control

Buffer tanks are very common in the process industry to alleviate the impact downstream of disturbances in temperature, concentration, and flow rate in important process streams (Faanes and Skogestad, 2003). In industry, buffer tanks are known under many different names, such as intermediate storage vessels, hold up tanks, surge drums, accumulators, inventories, mixing tanks, continuous stirred tank reactors (CSTR), and neutralization vessels. Typically, the buffer tank shown in Figure 4 is subject to significant and unsystematic variations in its inflow rate. For example, if the downstream is fed to a heater, fast changes in its feed give temperature variations which affect the rest of the process. Also, a buffer tank is often installed to avoid propagation of disturbances from batch operations to continuous processes. Furthermore, a buffer tank is also installed between operation units to allow a more flexible operation. Therefore, the task of controlling a buffer tank is such that the outflow rate must be changed smoothly despite significant variations in its incoming flow rate. To avoid overflow and empty, the level in the tank needs to be constantly varied within its operation minimum and maximum limits. However, the tank has a limited capacity that should be used appropriately. Thus, keeping the tank level in limitation is also an important component of the control task to be learnt.
These tanks are usually used as examples to check novel control algorithms, as they are simple to understand and easy to reproduce. For example, a neuro-fuzzy controller is proposed by Tani *et al.* (1996) for controlling a buffer tank using a predictive inductive model (neural network) and fuzzy decision rules.

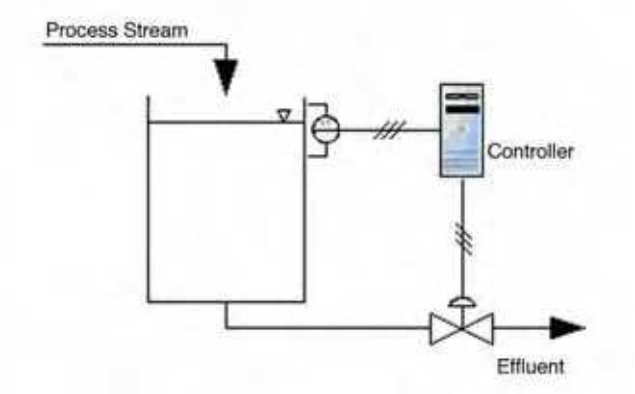


Fig. 4. Buffer tank

A very simple approach to control a buffer tanks using the proposed MFLC is now presented. For designing controller using MFLC, the designer should define how big the Q-

table and reward function can be. The learner will interact online with the environment and learn providing best actions to fulfil the requirement. Once the learner's parameters are defined, they can be used for other similar processes.

3.1 Problem definition

The tank has A=100 cm² and a level constrain $0<h<50$ cm. Clearly, the learning system is allowed to variate the level of the tank within its minimum and maximum capacity. Another limitation is that the controller can only manipulate the valve opening in the range $0\leq u\leq 100\%$. The learning controller must comply these limitations: The agent will be punished if it generates an action that causes the system to be outside this limitation.
The main objective of the proposed MFLC is to bring the outflow inside the goal band; the process responses are allowed to ocsillate within the band. Therefore, in the case of the buffer tank control, the goal band is selected to outflow within ± 2% error of the desired outflow (reference). On the other hand, the system allows the level to vary 60% from the head of the tank: The remaining 40% is for safety.

3.2 Design parameters

In this example, the goal band is defined as ± 1 l/m from the reference. Let reference, r see Figure 2, be 50 l/m and therefore, the parameter d is 1 and f is 5 l/m. The agent also has limitation $0.1\leq\Delta u\leq 0.3$ in the variation of the manipulated variable with regard to the previous control signal. The gain controller, k, is introduced to be 1×10^{-4}. By taking ρ = 1.5, therefore, there are 600 available actions in every symbolic state. For each available action, the controller will receive a positive reward (see equation 13) if the next response is inside the goal band. Meanwhile, if the next response is reaching the lower and upper bound output constrains, the selected action is punished. If the next state is not in the goal state, the selection of the action will also be negatively rewarded. That is :

$$r_{t+1}=\begin{cases}10 & \text{if } r-d<f_{out}{}_{t+1}<r+d,\\ -10 & \text{if } h_{t+1}\leq 0 \text{ or } h_{t+1}\geq 50,\\ -1 & \text{else.}\end{cases} \tag{13}$$

The discounted factor, γ, is set to 0.9 while the learning rate, α, is set to a value of 0.1. The policy for selecting an action is ε-greedy policy, with ε = 0.1. The probability for selecting an optimal action from those available in each state is 90%.

3.3 Simulation results and discussion

The inflow rate into a simulated buffer tank is introduced as in Figure 5 (a), which is a sinusoidal signal with amplitude 20, from an average value of 50 l/m. The level evolution can be seen in Figure 5(b). The control signal, which is the opening of the out-flow valve, is shown in Figure 5 (c). Clearly, the controller opens the valve widely when the system observes that the level of the tank is lower; otherwise, the opening of the valve is reduced when the level of the tank is high to maintain outflow as constant as possible. As a result, the outflow of the system remains in the defined goal band; as shown in Figure 5 (d). The noise observes in outflow is because the agent has finite-discrete action space. Thus, the

controller objectives are fulfilled: the controller is capable to learn to avoid abrupt changes in the out flow.
Next section discusses the online laboratory application of the proposed MFLC to control pH and oxidation processes.

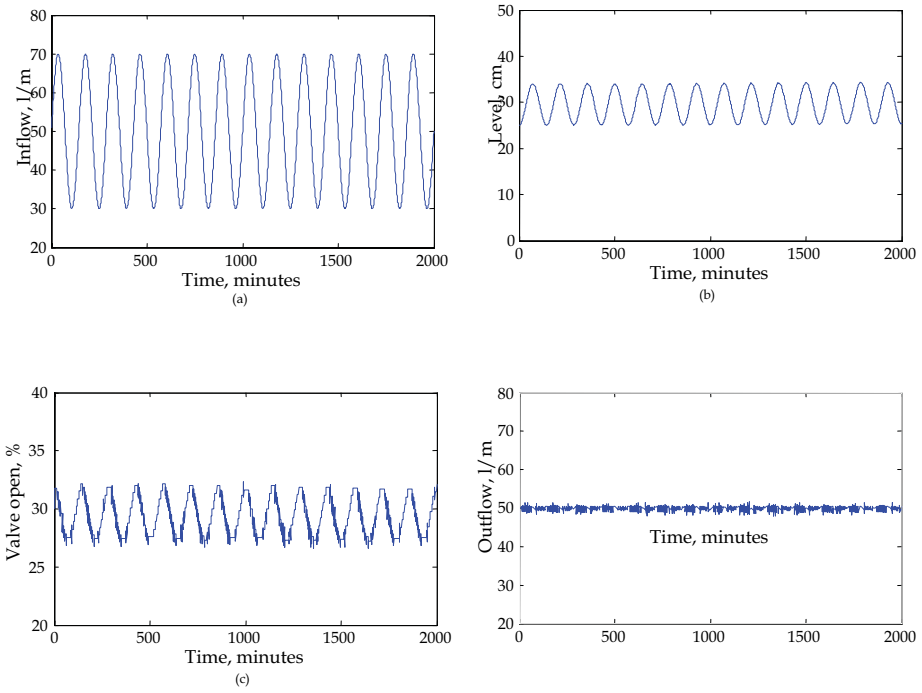


Fig. 5. Incoming and outcoming signal for learning to control buffer tank; (a) inflow signal, (b) liquid level in the tank, and (c) output manipulation flow.

4. Online laboratory assessment

The proposed algorithm has been tested for a view towards real-world applications in the laboratory plants. The first application is for controlling a pH process during wastewater treatment, which is known as a representative example of highly nonlinear, time-varying and difficult to model process plant, mainly resulting from interactions between many different chemical species. Thus, this pH process is very difficult to control using standard control techniques. Secondly, the MFLC algorithm is tested to control oxidation processes at certain ORP values corresponding to on-specification discharge.

4.1 pH control

pH control in neutralization process is a ubiquitous problem encountered many process control industry (see Kalafatis *et al.*, 2005 and references therein). For example, the pH value

is controlled in chemical processes such as fermentation, precipitation, oxidation, flotation and solvent extraction process. Also, control of pH in food and beverage production (such as in bread, liquor, beer, soy sauce, cheese, and milk production) is an important issue because the enzymatic reactions are affected by the pH value of the process and each has its an optimum pH which is critical to the yield. Other parameters involved in controlling pH process are chemical equilibrium, kinetic, thermodynamic and mixing problems. Considering all these influencing factors in controller design is an overwhelming task. On the other hand, the process buffer capacity varies with time, which is unknown and dramatically changes process gain. This can be understood as, for example, if either the concentration in the inlet flow or the composition of the feed changes, the shape of the titration curve will be drastically altered. This means that the process nonlinearity becomes time dependent and the system moves among several titration curves. Other characteristics include the dissociation of weak acids and bases or their salts involved in the solution determine the number of hydrogen ions. All weak species have the property, called buffering, to resist change in pH. A weak acid, for example, is not completely dissociated, so it can absorb hydrogen ion by converting them to undissociated acid molecules. Also, due to the nonlinear dependence of the pH value on the amount of titrated reactant the process will be inherently nonlinear. Therefore, it is difficult to develop a sound mathematical model of the pH process for designing a proper controller.

Many researchers proposed control strategies based on the titration curve (see Wright and Kravaris, 1991 and references therein). Wiener models are used for controller design by Kalafatis *et al.* (2005). These types of controllers are difficult to implement due to the complexity of the resulting control structures. Also, the designed controller is not a general purpose one, namely as the acid-base system change, the controller needs to be redesigned.

Intelligent controllers have been proposed by some researchers as alternative strategies, applying fuzzy control, neural networks or different combination of intelligent and model-based methods (Edgar and Postlethwaite, 2000; Krishnapura and Jutan, 2000; Mwembeshi *et al.*, 2004; Fuente *et al.*, 2006). As discussed in these cited references, tight and robust pH control are often difficult to achieve due to the inherent uncertain, nonlinear and time varying characteristics of pH neutralization processes. Also, the controller needs a huge number training examples in order to guarantee stability and performance.

In this section, the MFLC design strategy is assessed experimentally. The experimental setup consists of a CSTR (Figure 6) where a process stream (sodium acetate) is titrated with a solution of hydrochloric acid (HCl) to maintain at a certain pH value outflow stream. The solution of process stream is prepared for various concentration levels. However, the titrating stream is prepared using 1% concentration. To have the desirable outflow pH level, the controller manipulates titrating flow into the CSTR and it is assumed that the mixing in the tank is homogeneous; therefore, the concentration in the effluent stream is similar to the concentration in the reactor.

The control variable u_t is the flowrate of the titrating stream (normalized to the maximum value), which is applied using a peristaltic pump (ISMATEC MS-1 REGLO/6-160).

The output variable, y_t, is the logarithmic hydrogen ion concentration (pH) in the reactor. The pH value in the mixture is measured using an Ag-AgCl electrode (Crison 52-00) and transmitted using a pH-meter (Kent EIL9143). The electrode dynamic response presents appreciable and asymmetric inertia. The pH measured and the control signals are transmitted through an A/D interface (ComputerBoards CIO-AD16, 0-5V). The plant is

controlled and monitored with a standard PC, using Matlab and the Real-Time Toolbox for online control.

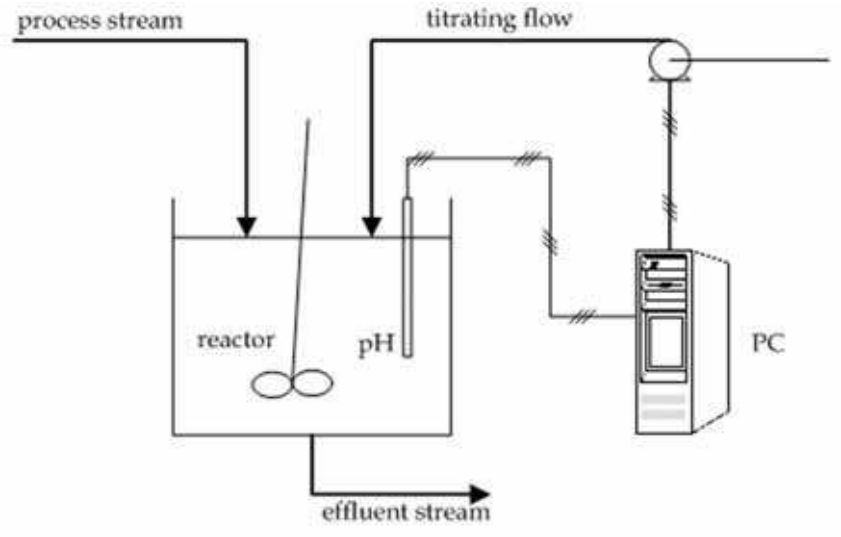


Fig. 6. Typical pH neutralization process plant.

The pH plant shown in Figure 6 is a typical laboratory set-up existing in the Department of Systems Engineering, University of Valladolid. The neutralization reactor is to overflow, hence the volume of liquid in the tank is constant (1 litter).

a) Parameters

The control objective is to bring the pH being inside a goal band with d=0.1, selected based on the level of measurement noise and the desired operating range of pH. The controller gain, k, is selected to be 2×10^{-5} and incremental control is defined as $-4.2\times10^{-4}<\Delta u<4.2\times10^{-4}$. There are 5 available states for negative or positive error. Therefore, there are 11 states. Every state has 5 available actions, except in goal state which only requires the wait action. Action 22 is the wait action.

In all the experiments the discounted factor, γ, is set to a value of 0.98 and the learning rate, α, is set 0.1. The Q-value matrix is initialized using zero entries. At every time step, the selected action is based on an ε-greedy policy, with ε=0.1, to leave enough room for the learning controller to explore state and actions. Rewards are defined using the simple assignment function

$$r_{t+1}=\begin{cases}1 & \text{if } r-d<pH<r+d\\ -1 & \text{else}\end{cases} \tag{14}$$

b) Online Experimental Results and Discussion

Many experiments were carried out in the laboratory plant with different conditions, and with small variations in the algorithms and tuning parameters. For most cases, the application of the proposed MFLC controller to the laboratory plant showed good

responses. Some responses of the plant for some changes in setpoint, compared with the responses for a PID in similar conditions can be seen in Figure 7 for the sodium acetate – hydrochloride acid system. The PID controller was tuned based on operating conditions at pH=5, where correction and proportional gains are chosen to be 0.01 and 0.001, respectively, whereas derivative and integral times are selected to be 1.
The comparison shows that the responses of the proposed MFLC algorithm settle in the reference faster than the PID controller, when a similar time is spent for the parameters: PID gives higher peaks and some oscillations due to variations from the nominal conditions.

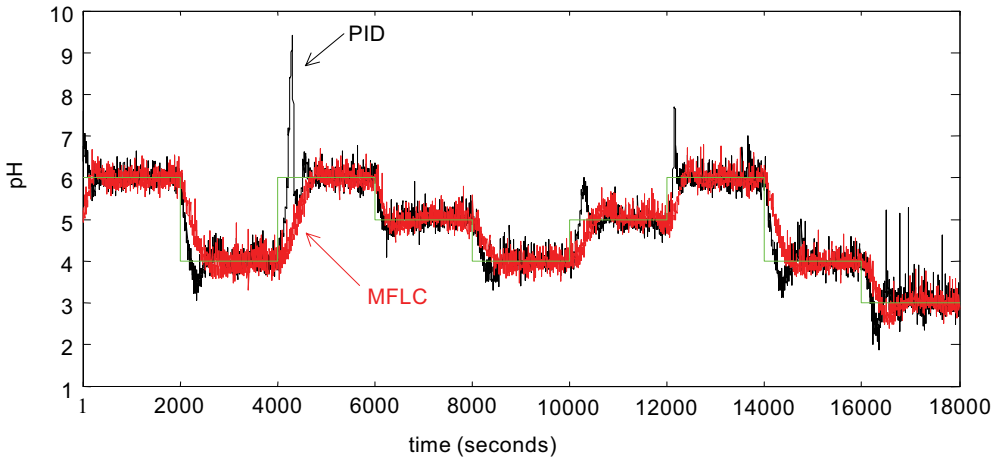


Fig. 7. Output responses of the plant for NaCH3COO-HCl.

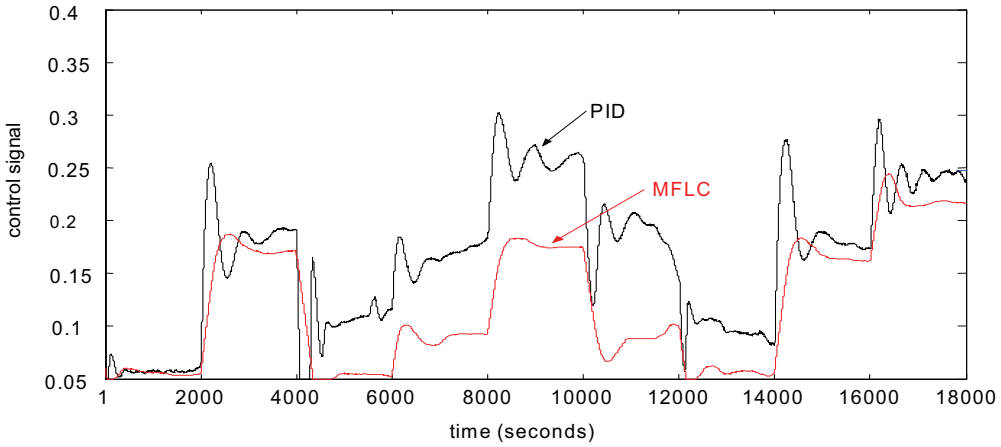


Fig. 8. Control signals for NaCH3COO-HCl.

Also the MFLC controller manipulates the actuator in a smoother than the one given by the PID controller (see Figure 8). Since MFLC allows a tolerance error of the process whenever the pH is within the control band, the control signal is very smooth when the pH is closer or within the control band, even if some exploration is carried out. The detailed discussion of the application MFLC to pH control is given in Syafiie *et al.* (2007a).

4.2 ORP control in Fenton's oxidation processes

Nowadays a central issue in the treatment of industrial wastewaters is the elimination of certain organic pollutants, which are very harmful to health even in small concentrations. Some of them are phenols, which are usually efficiently and economically eliminated through oxidation using Fenton's reagent. This Fenton's reagent refers to iron-mediated hydroxyl (●OH) production by hydrogen peroxide (H_2O_2).

The main issue is maintaining adequate values of hydroxil concentrations despite the huge number of chemical reactions involved (Laat and Gallard, 1999; Kwan, 2003). Unfortunately, it is very difficult to develop a sound mathematical model of the Fenton's oxidation processes for control purposes. Some reactions are slow rate and others are relative fast, but refractory intermediate act as a bottleneck for the complete oxidation. Also, as the process is used to decompose organic compounds, many parallel reactions are involved. For more detailed discussion, see Syafiie *et al.* (2007b).

Moreover, even if a detailed mathematical model were available (possible involving dozens of chemical reactions), it would be useless for real-time control, as it is not possible to measure in real-time the concentration of specific components ($OH^{\bullet}$, Fe^{3+}, Fe^{2+}, etc): the only available sensors are pH (to measure H^+ concentration) and ORP sensors to estimate the oxidizers activity (where ORP stands for oxidation-reduction potential). When using ORP for process control, it means that it is the present of the oxidizer or reducer that is being monitored, and not the chemical it is reacting with (McPherson, 1993). Thus, non-model based algorithms based on Reinforcement Learning ideas, such as the proposed MFLC algorithm would be very adequate to control this process.

A schematic of the experimental setup used to test the proposed algorithm is shown in Figure 9: For elimination of phenols, it is known that the oxidation reaction for phenol breakdown operates optimally on 550 to 600 mV of ORP value (Kwan, 2003), so the setpoint of the first MFLC agent is set to 570mV. It is also known that Fenton's reaction must be conducted on the range of temperatures between 80 to 90 ^{o}C, which is regulated using a simple thermostat, to represent industrial practice. Also, level in the buffer tank is not controlled to represent industrial practice, although there are detectors for low and high values. The reaction occurs on pH values between 3 and 5, so in the pretreatment the wastewater is titrated with hydrochloride acid.

The final part of the process is based on regulating pH to neutralize the drain (It would be dangerous for environment if the drain is released without neutralizing the pH). Therefore,. This neutralization is based on titrating the acidy stream (drain) with the base titrating flow (NaOH) to have pH around 7. Controlling pH of this strong acid-strong base system is known very difficult because the process is extremely nonlinear around the neutral pH, so it will be controlled using a second MFLC agent, designed following the methodology shown in previous section, coordinating with the first one.

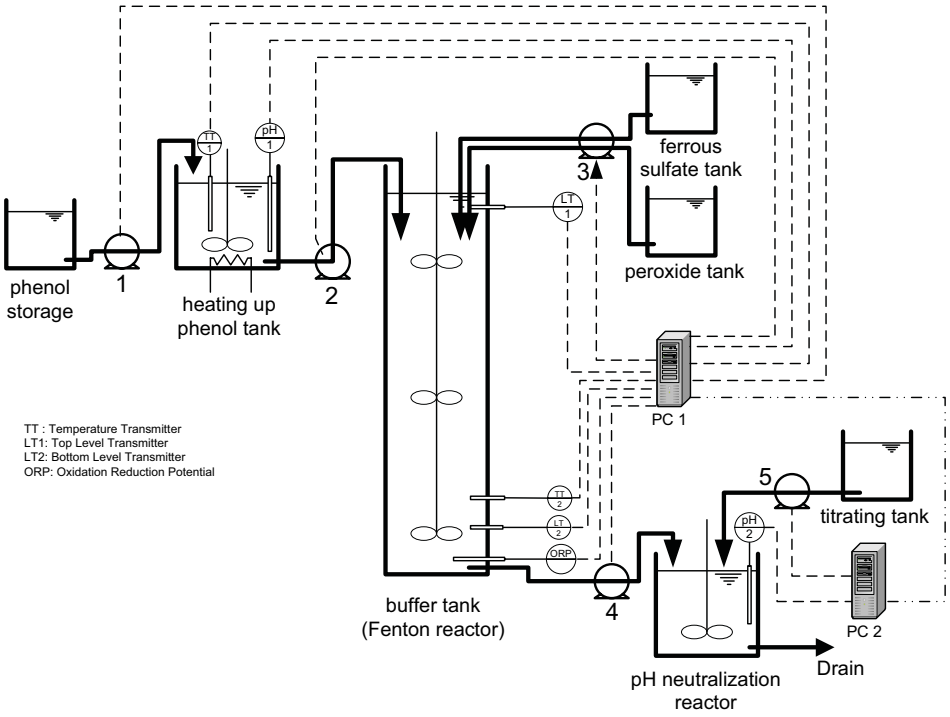


Fig. 9. Wastewater oxidation at a laboratory pilot plant

In summary, the first agent in order to handle the oxidation process receives reading of the process variables: apart from ORP value in buffer tank, temperature of inlet stream, and level and temperature in buffer tank. From this information, the agent learns to perform actions to control the oxidation process using the MFLC algorithm. For start-up of the process, first the wastewater is pretreated by heating and pH regulation. Once the temperature of the process stream reaches 80ºC, the first agent starts the process (turn on the pump 2) and starts manipulating the Fenton's reagent coming into the buffer tank (pump 3) to learn to handle the oxidation process.

a) Parameters

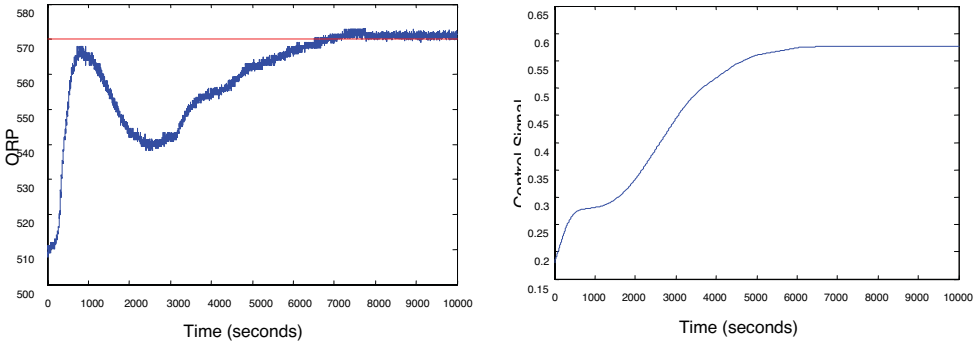
In this section, the selection of parameters for the MFLC agent that controls the oxidation process is discussed. The values of discounted factor γ=0.98 and learning rate a=0.1 from the previous study for pH control are maintain as the dynamics are similar. An ε-greedy policy is used, with parameter ε=0.1, to leave space for the agent to explore. To allow for sensor noise, the process goal is defined to be that the controller tolerates only d=5mV deviations from the setpoint r. In normal operation, states are defined for at most 100 mVolt for positive and negative error: thus, there are 41 states. The gain, k, is chosen to be 1×10^{-5}: Thus, every state has 20 available actions. The reinforcement signal is simple defined to be:

$$r_{t+1} = \begin{cases} 1 & \text{if } r-d < ORP < r+d \\ -1 & \text{else} \end{cases}, \tag{15}$$

The second agent, that controls the neutralization process, is the same as in the previous section, except that the controller gain is selected smaller, $k = 5\times10^{-7}$, because the process has higher gain.

b) Online Experimental Results and Discussion

Different experiments were carried out in the laboratory plant using the proposed MFLC agent. Some experiments are now shown for 1000 ppm phenol concentration, 10% $FeSO_4$, 1% NaOH, 1% HCl and 30% H_2O_2

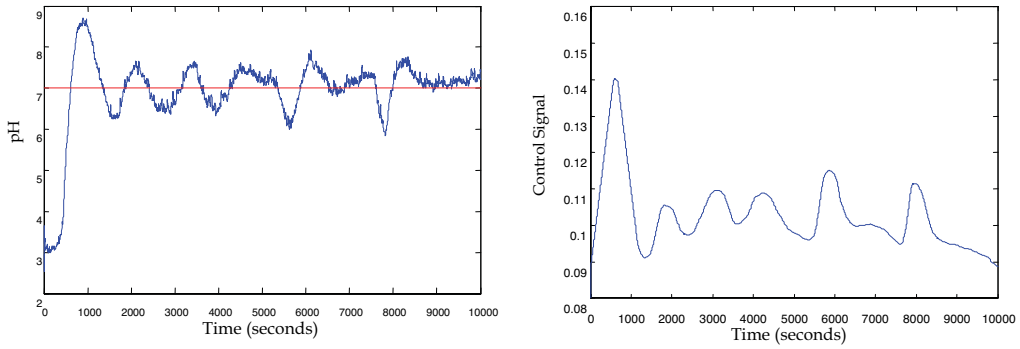


(a) (b)

Fig. 10. Oxidation control; (a) measured output, and (b) control signal

One typical response of the phenol decomposition process, controlled by the proposed MFLC agent, after learning, can be seen in Figure 10 (a): it can be seen that the MFLC controller maintains the oxidation process around the desired ORP level. This is carried out despite the complex dynamics of the system; During the first 1000 seconds, the process responses was reaching fast the reference, but then the process responses went down (until around 3000 seconds), because of the sequence of slow reactions that consumed both oxidizer and catalyst. After the balancing reaction are reached, then the responses of the process slowly returns to the goal band by increasing the control signal (see Figure 10, b). Thus, the responses of the process are most of time being on the optimal range of the reaction (550 to 600 mV), so phenols are correctly oxidized.

At the same time, the second agent manages the pH of the process on neutral range before it is discharged to environment. The responses of the neutralization process are plotted on Figure 11 (a). The second agents learns to manipulate the process to maintain it within the goal band, although there are some oscillations around the setpoint, as this is known to be a highly nonlinear process and the inlet composition changes with time, depending on the reactions in the buffer tank. The control signals (Figure 11, b) show that the agent actively manipulates the control signal when the process is outside the goal band.



(a) (b)

Fig. 11. pH process control; (a) measured output, and (b) control signal

5. Conclusion

This chapter has presented a proposal to apply RL algorithms for process control problems. This proposal (called MFLC algorithm) is based on the well known *Q*-learning algorithm, using an specific definition of symbolic states based on specifying tolerances on the outputs and constraints on the control and its variation. Also, the propose approach uses few and simple tuning parameters to simplify the presentation of these techniques to plant operators. The technique has been presented on a simple example (buffer tank) to present the ideas behind the algorithm (in particular, the parameter selection issue) and then some experimental results in wastewater control problems have been presented to show the applicability of the proposed ideas. It is shown that the control objectives are fulfilled by the proposed MFLC agents, with smooth manipulated variables. Thus, the proposed MFLC technique is promising for increasing the degree and type of automation that can be effectively used in process control.

6. References

Baeza J. A., E. C. Ferreira and J. Lafuente (2000), Knowledge-Based Supervision and Control of Wastewater Treatment Plant: a Real-Time Implementation, *Water Science and Technology*, Vol. 14, No. 12, pp. 129-137.

Bucak I. O. and M. A. Zohdy (2001), Reinforcement learning control of nonlinear multi-link system, *Journal of Engineering Application of Artificial Intelligence*, Vol. 14, pp. 563 – 575.

Chidambaram, M. and See, R. P. (2002), A simple method of tuning PID controllers for integrator/dead-time processes, *Computer & Chemical Engineering,* Vol. 27, pp. 211–215.

Crites R. H. And Barto A. G. (1996), Improving Elevator Performance Using Reinforcement Learning, *Advances in Neural Information Processing Systems 8*, MIT Press, Cambridge MA.

De Laat, J., and H. Gallard (1999), Catalytic decomposition of hydrogen peroxide by Fe(III) in homogeneous aqueous solution: mechanism and kinetic modelling, *Environmental Science Technology*, Vol. 33 No. 16, pp. 2726-2732.

Faanes, A. & Skogestad, S. (2003), Buffer Tank Design for Acceptable Control Performance, *Industrial Engineering Chemistry Research*, Vol. 42, pp. 2198-2208.

Fuente M. J., Robles C., Casado O., Syafiie S. & Tadeo F. (2006), Fuzzy Control of a Neutralization Process, Engineering Applications of Artificial Intelligence, Vol. 19, pp. 905-914.

Kalafatis A. D., L. Wang and W. R. Cluett (2005), Linearizing feedforward-feedback control of pH process based on Wiener model, *Journal of Process Control*, Vol. 15, pp. 10 3 – 112.

Krishnapura V. G. and Jutan. A. (2000), A neural adaptive controller, *Chemical Engineering Science*, Vol. 55, pp 3803 – 3812.

Kwan, W.P. (2003), *Decomposition of Hydrogen Peroxide and Organic Compounds in the Present of Iron and Iron Oxides*, PhD Dissertation, MIT.

Martinez E. C. (2000), Batch process modelling for optimization using reinforcement learning, *Computer and chemical engineering*, Vol. 24, pp. 1187 – 1193.

McPherson L. L. (1993), Understanding ORP's in the disinfection process. *Water Engineering Management*, Vol. 140, No. 11, pp.29-31.

Mwembeshi M. M., Kent C. A., and Salhi S. (2004), A genetic algorithm based approach to intelligent modelling and control of pH in reactor, *Computer and Chemical Engineering*, Vol. 28, No. 9, pp. 1743 – 1757.

Ng A. Y., Coates A., Diel M., Ganapathi V., Schulte J., Tse B., Berger E. and Liang E.(2004), Inverted autonomous helicopter flight via reinforcement learning, In *International Symposium on Experimental Robotics, 2004*.

Peng Y.Z. Gao J. F., Wang S. Y. and Sui M. H. (2002), Use pH and ORP as fuzzy control parameters of denitrification in SBR process, *Water Science Technology*, Vol. 46, No. 4-5, pp. 131 – 137.

Ramirez, N., and Jackson H. (1999), "Application of neural networks to chemical process control", *Computers and Chemical Engineering*, Vol. 37, pp. 387-390.

Schoknecht, R. and M. Riedmiller (2003), Learning to control a multiple time scales, *Proceeding of ICANN 2003*, Istanbul Turkey, June 26 – 29, pp 479 – 487.

Sutton, R. S. & Barto, A. G. (1998), *Reinforcement Learning: an Introduction*, the MIT Press, Cambridge, MA.

Syafiie S., F. Tadeo and E. Martinez (2007a), Model-Free Learning Control of Neutralization Process Using Reinforcement Learning, *Engineering Application Of Artificial Intelligence*, Vol. 20, No. 6, pp. 767-782.

Syafiie S., F. Tadeo, and E. Martinez (2007b), Coordinated Control of Wastewater Oxidation Processes under Constrained Incremental Control, proceeding of *European Control Conference 2007*, Kos, Greece 2 – 5 Juli, 2007

Tani T., Murakoshi S. and Umano M. (1996), Neuro-fuzzy hybrid control system of tank level in petroleum plant, IEEE transactions on Fuzzy Systems, Vol. 4, No. 3, pp. 360 – 368.

Watkins C. J. C. H. (1989), *Learning from delayed rewards*, a PhD thesis at King's College, Cambridge.

Wright, R. A., and Kravaris, C.(1991), "Nonlinear Control of pH Processes Using the Strong Acid Equivalent" *Industrial and Engineering Chemistry Research*, Vol. 30, No. 7, pp. 1561-1572.

Ziegler, J. G. and Nichols, N. B. (1942), Optimum settings for automatic controllers, *Trans. ASME*, Vol. 64, pp. 759 – 765.

Reinforcement Learning

Edited by Cornelius Weber, Mark Elshaw and Norbert Michael Mayer

ISBN 978-3-902613-14-1
Hard cover, 424 pages
Publisher I-Tech Education and Publishing
Published online 01, January, 2008
Published in print edition January, 2008

Brains rule the world, and brain-like computation is increasingly used in computers and electronic devices. Brain-like computation is about processing and interpreting data or directly putting forward and performing actions. Learning is a very important aspect. This book is on reinforcement learning which involves performing actions to achieve a goal. The first 11 chapters of this book describe and extend the scope of reinforcement learning. The remaining 11 chapters show that there is already wide usage in numerous fields. Reinforcement learning can tackle control tasks that are too complex for traditional, hand-designed, non-learning controllers. As learning computers can deal with technical complexities, the tasks of human operators remain to specify goals on increasingly higher levels. This book shows that reinforcement learning is a very dynamic area in terms of theory and applications and it shall stimulate and encourage new research in this field.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821