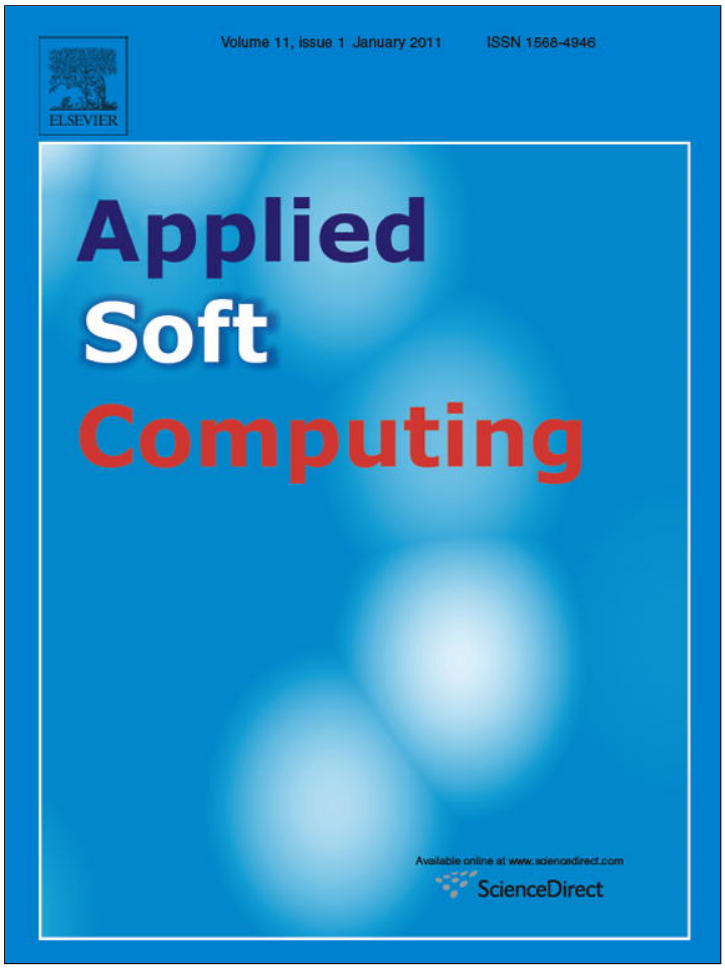

Applied Soft Computing 11 (2011) 73–82

Contents lists available at ScienceDirect

Applied Soft Computing

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

# Model-free control based on reinforcement learning for a wastewater treatment problem

S. Syafiie[a], F. Tadeo[b], E. Martinez[c], T. Alvarez[b,*]

[a] Department of Chemical and Environmental Engineering, Faculty of Engineering, University Putra Malaysia, 43400 Serdang, Selangor D.E., Malaysia
[b] Department of Systems Engineering and Automatic Control, Faculty of Sciences, University of Valladolid, Prado de la Magdalena s/n, 47011 Valladolid, Spain
[c] Consejo Nacional de Investigaciones Científicas y Técnicas, Avellaneda 3657, 3000 Santa Fe, Argentina

ARTICLE INFO

*Article history:*
Received 10 April 2008
Received in revised form
25 September 2009
Accepted 18 October 2009
Available online 25 October 2009

*Keywords:*
Wastewater treatment plants
Intelligent control
Oxidation–reduction potential

ABSTRACT

This article presents a proposal, based on the model-free learning control (MFLC) approach, for the control of the advanced oxidation process in wastewater plants. This is prompted by the fact that many organic pollutants in industrial wastewaters are resistant to conventional biological treatments, and the fact that advanced oxidation processes, controlled with learning controllers measuring the oxidation–reduction potential (ORP), give a cost-effective solution. The proposed automation strategy denoted MFLC-MSA is based on the integration of reinforcement learning with multiple step actions. This enables the most adequate control strategy to be learned directly from the process response to selected control inputs. Thus, the proposed methodology is satisfactory for oxidation processes of wastewater treatment plants, where the development of an adequate model for control design is usually too costly. The algorithm proposed has been tested in a lab pilot plant, where phenolic wastewater is oxidized to carboxylic acids and carbon dioxide. The obtained experimental results show that the proposed MFLC-MSA strategy can achieve good performance to guarantee on-specification discharge at maximum degradation rate using readily available measurements such as pH and ORP, inferential measurements of oxidation kinetics and peroxide consumption, respectively.



## 1. Introduction

In many industrial wastewaters there are important organic pollutants (such as phenols) which are resistant to biodegradation and other conventional methods. Increasing concern about environmental and health risks demands a more rigorous control of wastewaters, promoting the development of new treatment technologies that are capable of dealing with toxic organic pollutants. These technologies depend on a rigorous control of the operating conditions to guarantee on-specifications discharge, so automation of wastewater treatment plants is an important issue when introducing these advanced treatment technologies.

This paper focuses on model-free control of advanced oxidation processes (AOPs), specifically the so-called Fenton process [1], as they set challenging problems that need to be carefully addressed from a control point of view in order to adequately eliminate a variety of biocide and refractory organic pollutants. In particular, AOPs are an interesting treatment option for phenolic wastewaters because of their potential to oxidize recalcitrant organic molecules. These processes are based on the in situ generation of a powerful oxidant: hydroxyl radicals ($OH^{\bullet}$). Although there are many efficient methods to generate this radical: Fenton, photo-Fenton, $O_3$/UV, $O_3/H_2O_2$, etc. [29], in this study, Fenton's reagent is used to generate the hydroxyl radicals as it is the most cost-effective approach. In this process, hydroxyls are generated from the decomposition of hydrogen peroxide ($H_2O_2$) using iron as the catalyst [19]; these hydroxyl radicals then attack the phenol molecule to form catechol and hydroquinone.

Fenton's set of reaction steps is quite complex, involving crossed reactions between hydrogen peroxide, hydroxyl radicals, iron complexes, etc. In fact, ignoring the effect of organic materials completely, there are at least a dozen chemical reactions involved, with intermediate products impossible to measure in practice [4], so using a model for the development of controllers is not feasible. For example, although the main reaction, from the hydroxyl production point of view, is the decomposition of $H_2O_2$ using $Fe^{2+}$:

$$Fe^{2+} + H_2O_2 \rightarrow Fe^{3+} + OH^- + OH^{\bullet} \quad (k = 76\,M^{-1}\,s^{-1}), \tag{1}$$

in the presence of $Fe^{3+}$, the hydroxide peroxide decomposes: this reaction must be inhibited by limiting the amount of $H_2O_2$ in the reaction medium, by introducing it as it is consumed, through careful control of the amount of hydroxide peroxide added [8]. Moreover, the pH of the reaction mixture must always be kept

* Corresponding author. Tel.: +34 983423276.
*E-mail addresses:* syafiie@eng.upm.edu.my (S. Syafiie), fernando@autom.uva.es (F. Tadeo), ecmarti@ceride.gov.ar (E. Martinez), tere@autom.uva.es (T. Alvarez).

1568-4946/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2009.10.018

between 2 and 6 to avoid the formation of iron complexes that decrease the efficiency of the catalyst, and to favor the formation of a stable, electrophilic structure via the solvation of a proton by the $H_2O_2$ molecule [4] (this pH range depends on the refractory nature of intermediate oxidation compounds such as carboxylic acids, so usually a tighter bound is used for control). Moreover, when organic compounds are introduced (in this study phenol [16]), other reactions will be involved, which makes it extremely difficult to develop a precise mathematical model of Fenton's oxidation kinetics for control purposes. The consideration of each organic component greatly increases the number of reactions and compounds: for example, it has been reported by [6] that adding monochlorophenol gives at least 28 additional reactions. Thus, it is impossible in practice to use model-based approaches for Fenton processes, and the system is too complex and subject to many uncertainties and variations to include expert knowledge in the controller.

Moreover, the level of automation of this kind of process in wastewater plants remains low, as sensors for specific chemical species are extremely expensive and inadequate for wastewater problems [3,12,15]. In this paper, ORP sensors are used: an ORP sensor is nearly identical to a pH sensor, estimating the oxidation–reduction potential (ORP), which is related to the concentration, activity and strength of oxidizers and reducers in a solution [5]. Thus, it provides an indication of the solution's ability to oxidize or reduce another material: the addition of oxidizers raises the ORP value, while the addition of reducers lowers the ORP value. Unfortunately, from a control point of view, it means that the oxidation capacity of the reactor content is being monitored, rather than the concentration of a given chemical species [11].

The literature on advanced control applications of ORP control in the Fenton process is rather scarce: some researchers used ORP readings as a key measurement to monitor redox reactions during process operation [3] and to manage the aeration in nitrification and denitrification of aerobic–anoxic activated sludge [12]. The sharp drop in the ORP curve in the denitrification reactor before the reactor reaches the nitrate "knee" point was maintained for phosphorus removal by [15]. Finally, [13,30] proposed a fuzzy logic controller based on ORP to determine the time of denitrification in a sequencing batch reactor (SBR).

To deal with the issue of modeling difficulties and costs, a control approach based on learning would be more adequate: as mentioned above, the problem at hand is not suitable for model-based approaches. Moreover, in wastewater processes, the characteristics change with time, depending on the chemical compounds, their concentration, temperature, pH, etc, of the inlet flow to be treated (see, for example, [28]). Thus, a learning approach is very adequate for this kind of problems, which require some kind of continuous on-line learning.

Different model-free learning control approaches adequate for process control problems have been proposed in the literature. Of these we would like to mention the approximate dynamic programming approach proposed by Lee and Lee [9,10], which uses a priori input–output data to develop a controller. Instead of the above, reinforcement learning [21] has been selected, as it provides a rigorous methodology for designing learning control systems without resorting to a priori modeling while using a simple control logic suitable for industrial implementation using conventional hardware.

An alternative methodology is based on replacing the table with a function approximation [9,10]. This methodology has been quite successful in many practical cases, and does not need to use explicitly the table of $Q$-values. However, it has been discarded in the approach presented in this paper, as the convergence guarantees for $Q$-learning algorithms do not generally hold when using a function approximator [2,27], which might make it difficult to promote its extended application in the area of process control, where only well-tested approaches are incorporated. This comes from the fact that, as the agent iteratively approximates the value function, each successive approximation depends on the previous one. Thus, a slight approximation error can quickly be incorporated into the "correct" model. As time goes on, the error can become large enough to make the learned approximation worthless. Another reason for not using function approximation is that the state-action space does not often have a straight forward distance metric that make sense in process control problems [20].

In particular, the model-free learning control (MFLC) approach, previously proposed by some of the authors [24,25], will be used. These MFLC controllers are based on reinforcement learning algorithms, so the control objective is the optimization of a desired performance index by learning to create appropriate control actions through continuous interaction with the plant. Thus, learning is performed without requiring an explicit model of the plant: instead, the system's dynamics is implicitly learned and represented in the control feedback laws and value function.

This article is organized as follows: first, a short presentation of MFLC is given in Section 2. The proposed technique to control the oxidation process by using MFLC is given in Section 3. Experimental results obtained from the application of the proposed technique to a laboratory plant are given in Section 4. Finally, some conclusions are given.

## 2. MFLC-MSA for process control

### 2.1. MFLC

Model-free learning control is a control design methodology, based on reinforcement learning [17,21], that was proposed [23] because of the difficulty of applying reinforcement learning directly to practical process control problems: even if most RL algorithms are memory-linear on the number of actions $n$, polynomial in the feature space and not worse than $O(n \log n)$ in time [7], the space of actions and states would be huge and the transition probabilities are unknown in advance. On the other hand, designing a controller should take into account the fact that the system usually has input and output constraints. The proposed MFLC satisfies this issue by bounding the incremental control signal, and including hierarchical input and output limitations. It was presented in detail in [24,25], so only a brief presentation is given here.

#### 2.1.1. Motivation for MFLC

MFLC is a promising control approach for wastewater treatment processes: MFLC-based controllers can be considered a "smart" control, as this kind of controller fulfills the usual requirements in this kind of process with little input from the designer or operator.

A summary of the characteristics of MFLC is as follows:

- General purpose approach.
- Adaptive capability to handle environmental uncertainty.
- Modeling is not required.
- Easy integration of a priori knowledge.
- Few and easy-to-understand tuning parameters.
- Convergence of the algorithm is ensured.

#### 2.1.2. MFLC architecture

The generic architecture for model-free learning control is depicted in Fig. 1; it is a modular strategy, based on a simple selection of states, actions and control signals, with the objective of being easily understood by the final user. At each sample time, the controller resorts to the "Policy" to select one action from those available in the current plant state. Then, the chosen action is converted to an analogical control signal in the "Calculation U" block.

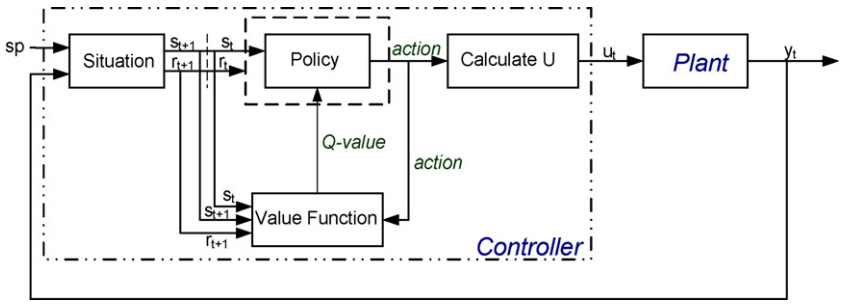


**Fig. 1.** MFLC architecture based on Q-learning.

Based on the measured output, the "Situation" block categorizes the resulting state and the corresponding reward. From this reward the so-called Q-value table is updated in the "Value Function" block to reflect the adequacy of the selected action.

In the basic formulation of MFLC, at each time step, actions are selected by the controller and learning is carried out by externally criticizing them as "good" or "bad" depending on the resulting state. Every action that drives the system into the goal state is considered a good action and receives a positive reward. However, actions that do not drive the system into the goal state are not rewarded. Prior knowledge can used to define purposeful courses of actions (macro-actions) to achieve the goal state of the system along with stopping conditions to switch to a different behavior when the expected results are not obtained after a number of time steps.

A central part of the learning algorithm is the discovery of the Q-value function, which measures the cumulative benefit of applying the action $a_t$ when the system is in state $s_t$. This function is stored in a matrix $Q(s, a)$. To incrementally learn this Q-function, it is necessary to take into account the current and future benefits: when action $a_t$ has been selected and applied to the plant, the system moves to a new state $s_{t+1}$ and the learning controller 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:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left( r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1}, a) - Q(s_t, a_t) \right), \quad (2)$$

where:

- $A$ is the set of possible actions in the new state.
- The learning rate $\alpha$ is a tuning parameter that can be used to optimize the speed of learning (large learning rates make learning faster, but might induce oscillations). In fact, in practice the learning rate decreases with time, to represent the fact that the system is better known as times goes by.
- The discount factor $\gamma$ is used as a forgetting factor to weight near-term reinforcements more heavily than distant future reinforcements: If $\gamma$ is small, the controller learns to act only for short-term reward; the closer $\gamma$ is to 1 the greater the weight assigned to long-term reinforcements.

### 2.2. MFLC-MSA control algorithm

In the original MFLC formulation, the one-step learning strategy [21] was used: a sequence of primitive actions is chosen following a given policy by the agent in the MFLC and policy improvement is made on-line using rewards given at each time step. However, oxidation processes have slowly unfolding dynamics due to a number of oxidation reactions, giving rise to intermediate species which are eventually mineralized to carbon dioxide. Accordingly, we propose an approach based on temporal abstraction [14,21,22]. From a control point of view this means that the controller executes the selected primitive action over several time steps until a suitable termination condition is reached. Undoubtedly, when the controller design is based on temporal abstraction, the policy will hold until a given termination condition applies. As soon as the termination condition is enforced, the control policy is used to explore and select another primitive or extended action from those available. Clearly, through temporal abstraction, the learning controller reduces the amount of non-relevant exploration and also the calculation time for defining the action to be implemented.

In particular, from the alternative approaches to temporal abstraction, this paper proposes the use of the concept of multiple step actions [18]. This makes it possible to reduce the computational cost, as the MSA method consists of several identical actions on the primitive time scale. Roughly speaking, the idea is to select one action (called "primitive action" in the MSA literature) that will be applied during a given period of time. This set of successive actions is called the "extended action", which is applied completely unless the system reaches the objective (in the MFLC algorithm, when the estimated symbolic state is the goal state) or clearly diverges from the desired objective.

Applying the same primitive action for several time steps makes it possible to reduce control efforts and speed up learning. This faster learning is obtained because using temporally extended actions forces the controller to take larger actions (the effects of primitive actions are usually small, so they have to be repeated many times to find the effect). Moreover it increases the responsiveness, as the agent exploits a given action and knows the effects clearly. This technique is suitable for process control because it is suited to systems where no decomposition in sub problems is known in advance [18].

Although the original MSA approach is not adequate for unstable, fast or very nonlinear systems, as it is assumed that the system does not change significantly while the extended action is applied, special measures are taken in MFLC-MSA if the system changes significantly during the application of the extended action: the extended action is interrupted if the system does not respond as expected. The idea is that if the responses of the process drive it far from the desired state, the extended action is interrupted and the agent re-explores the best action in the current state [25].

Thus, the basic idea in MSA algorithms is as follows:

(1) The learning controller selects a primitive action $a_t$ which is applied during $m$ time steps, unless the system reaches the objective or diverges from it.
(2) Rewards are assigned not only to the implemented MSA but to all parts of the MSA, which are actually also MSA with fewer time steps.

It must be pointed out that while the agent is applying a given primitive action, the agent stops exploring the other primitive actions for $m$ time steps. However, the agent keeps criticizing those MSAs which are sub-sequences of the selected MSA. Although temporal abstraction is used to define macro-actions of variable

lengths, exploration is required to discover when is more productive to interrupt a given macro-action and switch to a different macro-action. Clearly, the advantage is that the agent reduces the amount of exploration for learning.

Thus, the proposed MFLC-MSA algorithm is the following:

1. Initialize $m = 0$ and estimate the state $s_t$.
2. Select an action $a_t$, using $\varepsilon$-greedy policy from the actions available in state $s_t$.
3. (MSA) Apply $a_t$ during $n$ times steps.
4. Repeat for these $n$ time steps:

(a) Set $m = m + 1$ and estimate the new state $s_{t+m}$.
(b) If the evolution of the states deviates from the goal state, interrupt the MSA and return to Step 1.
(c) Update the $Q$-value using

$$Q^m(s,a) \leftarrow Q^m(s,a) + \alpha\left(r_{t+1}^m + \gamma^m \max_{a\in A} Q(s,a) - Q^m(s,a)\right) \quad (3)$$

5. Return to Step 1.

In reinforcement learning literature, the *optimal action* denotes the action that has maximum $Q$-value at a given time. In order to avoid confusion with the usual terminology of optimality in control literature, this action is now called *best action*. The greedy policy for selecting the best action is

$$\pi(s) = arg\max_{a\in A} Q, \quad (4)$$

However, as the agent's goal is to maximize future reward, it should then, naively, always choose the one action which leads to maximal immediate reward. This, however, prevents exploration. A suboptimal action performed now will perhaps lead to a much higher cumulative reinforcement later. In other words, it can be described as follows: the agent knows exactly neither the optimal value function nor the correct estimation of the plant dynamics. If the agent knows this value correctly, the policy can select a greedy action that maximizes the value function at each state. If this estimation and prediction are good enough, therefore, a good policy would simply be to choose the greedy action; this is called exploitation. However, when the learning agent does not know the correct optimal value function, in order to discover it, the agent should execute trial actions, i.e. actions that are not optimal with respect to the current value function; this is called exploration.

The so-called $\varepsilon$-greedy strategy may be used to explore apparently non-optimal actions by giving a space, $\varepsilon$, for the agent to search and discover better actions, that is

$$\pi(s) = \begin{cases} a^* \equiv arg\max_{a\in A} Q, & \text{with probability } 1-\varepsilon \\ a \in A,\ a \neq arg\max_{a\in A} Q, & \text{with probability } \varepsilon \end{cases} \quad (5)$$

Thus, the action that has maximum value function will be selected with $1-\varepsilon$ probability, whereas any action that does not have maximum value function will be selected with probability $\varepsilon$.

## 3. MFLC for control of oxidation of wastewater

### 3.1. State description

A central issue in reinforcement learning algorithms is the definition of the system states: so far, RL has been successfully used mostly for simple problems characterized by discrete state and action spaces. However, in process control, both state and action spaces are a continuum, so a symbolic state characterization is now proposed to help address the dimensionality problem with little loss in performance. Moreover, the state definition rationale must be simple enough for operators to understand and must also be based on insights concerning the process. Thus, in MFLC, the states are defined based on the control objective and input constraints, trying to follow simple ideas borrowed from control techniques frequent in process control.

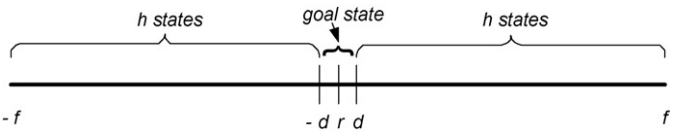


**Fig. 2.** State definition from tracking error.

In a SISO implementation of the MFLC framework, the control task is defined so as to maintain the desired output (e.g., ORP or pH) inside the band $r-d$ and $r+d$, as shown in Fig. 2. The width of this band is defined based on the tolerance of the control task (which depends on measurement noise, disturbances and system specifications). This band is defined as the *goal band*, and corresponds to the *goal state*, which the learning controller should achieve and maintain (it is now assumed, without loss of generality, that this is exactly in the middle of the working range). To ensure offset-free control, it is important that $d$ corresponds to the amplitude of the expected measurement noise (which is assumed to have a mean value of 0).

To describe the rest of the symbolic states, an approach similar to that of the gain-scheduling technique frequently used in the process control problem is used: it is considered that the agent has $h$ 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 process behaviors, but small enough to reduce computational time and the size of the $Q$-value function matrix).

If needed, the "length" of each state can be calculated as follows:

$$c = \frac{f-d}{h}. \quad (6)$$

Thus, the positive bound parameter can be presented as:

$$\omega_i = d + (i-1)c, \quad i \in [1,\ldots,h]. \quad (7)$$

(For negative errors, the bound parameter is trivial by changing signs.)

Thus, the vector of symbolic states can be presented as follows:

$$g_j = \begin{cases} e - \omega_j & \text{if } e \le \omega_j \\ \omega_j - e & \text{otherwise} \end{cases} \quad (8)$$

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

$$s_t = arg\max g_j. \quad (9)$$

### 3.2. Action description

In the single-input single-output version of MFLC, the control signal $u_t \in \mathbb{R}$ is calculated by varying the previous control signal in a magnitude calculated from the difference of the numerical values of the selected best action, $a_t \in \mathbb{N}$, with respect to the *wait action*, $a_w$ (action corresponding to maintaining the previous control signal). That is,

$$u_t = u_{t-1} + k(a_w - a_t). \quad (10)$$

This gives a PI-like structure, which simplifies initialization and tuning for the end user ($k$ is the tuning parameter). The gain $k$ must be positive if we want the control signal to increase when the output is below the reference. This is because, for states below the goal state, the only available actions are actions with numbers smaller than the wait action (see Fig. 3). These actions increase the control signal if $k$ is positive.

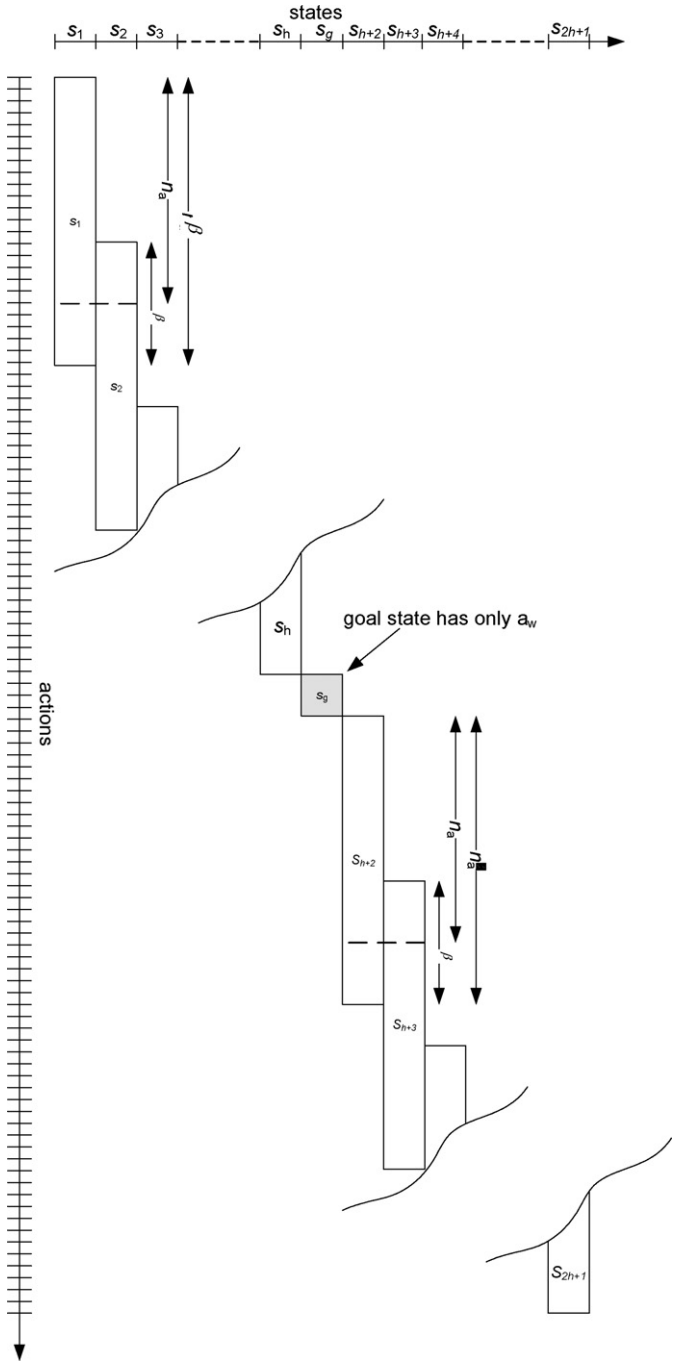


**Fig. 3.** Definition of the set of actions at each state.

At each state, there is only a finite set of possible actions (see Fig. 3). These actions are selected based on the limitations on the minimum and maximum variations of the control signal (which appear in any practical control problem, and are known from basic knowledge of the process), as follows: Let the incremental control be bounded as:

$$\underline{\Delta u} \leq \Delta u \leq \overline{\Delta u}. \tag{11}$$

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

$$N_{\mathrm{a}} = 2h\left(round\left(\frac{\overline{\Delta u} - \underline{\Delta u}}{kh}\right)\right) + 1. \tag{12}$$

To satisfy the upper bound (12), the round up function is used. From (13), the value corresponding to the wait action $a_{\mathrm{w}}$, can be calculated as follows:

$$a_{\mathrm{w}} = \frac{N_{\mathrm{a}} + 1}{2}. \tag{13}$$

If there is no overlapping, the number of actions in each state can be calculated as:

$$n_{\mathrm{a}} = \frac{N_{\mathrm{a}} - 1}{2h}. \tag{14}$$

However, to increase the number of available actions and represent nonlinear action-to-space relations (important in process control), a degree of overlapping is included (see Fig. 3). As is logical, at each state, not all the control actions are available, but only a subset. For example, if the state were a temperature, if this temperature is very low, the only logical actions are those that increase the temperature. Thus, the number of actions in each state is

$$n_a^{\beta} = n_a(1 + \beta), \tag{15}$$

where $\beta$ is a parameter that gives the degree of overlapping of the actions with neighboring states (always selected such that $n_a^{\beta}$ is integer): this overlapping represents the fact that some actions are possible in similar states.

Then, the available actions for every state go from $a_p^j$ to $a_b^j$ (except in the goal state, where there is only one action, namely the wait action). The idea is presented in Fig. 3. Those available actions can be calculated as:

$$\begin{aligned} a_p^j &= a_p^{j-1} + (j-1)v, \\ a_b^j &= a_p^j + n_a^{\beta} - 1, \end{aligned} \tag{16}$$

where $v = \beta(n_a^{\beta}/h)$ and $a_p^{j-1}$ is the first action in the state $j$ calculated as

$$a_p^{j-1} = \begin{cases} 1 & \text{if } j = 1 \\ 2a_{\mathrm{w}} - a_b^{j-2} & \text{if } j = h+2 \end{cases} \tag{17}$$

## 4. Application to oxidation in wastewater plants

The oxidation of organic pollutants in industrial wastewater is usually carried out in three phases: pre-treatment (preheating, pH correction), oxidation in a reactor and post-treatment (pH neutralization). This paper concentrates on controlling the ORP in the oxidation reactor, as it is the most challenging problem, that will be solved using the MFLC-MSA technique described in this paper: temperature control in pre-treatment can be controlled using simple control techniques (a thermostat is frequent in practice), whereas control of the pH and neutralization process can be done using independent MFLC-based controllers, as presented previously [24,25]: a short summary will be given later.

It must be noted that, as has been mentioned before, the internal variables of this system cannot be measured on-line, as they correspond to concentrations of complex chemical compounds, for which standard sensors cannot be used, so all the information regarding the symbolic states must be extracted from the measured output: the ORP includes the information from all these chemical compounds in a single number. This contrasts with the systems usually solved using reinforcement learning approaches, where measurements give a precise characterization of the state.

The application of the proposed learning control methodology to the oxidation of phenols using Fenton's reagent in a laboratory plant is now presented in detail.

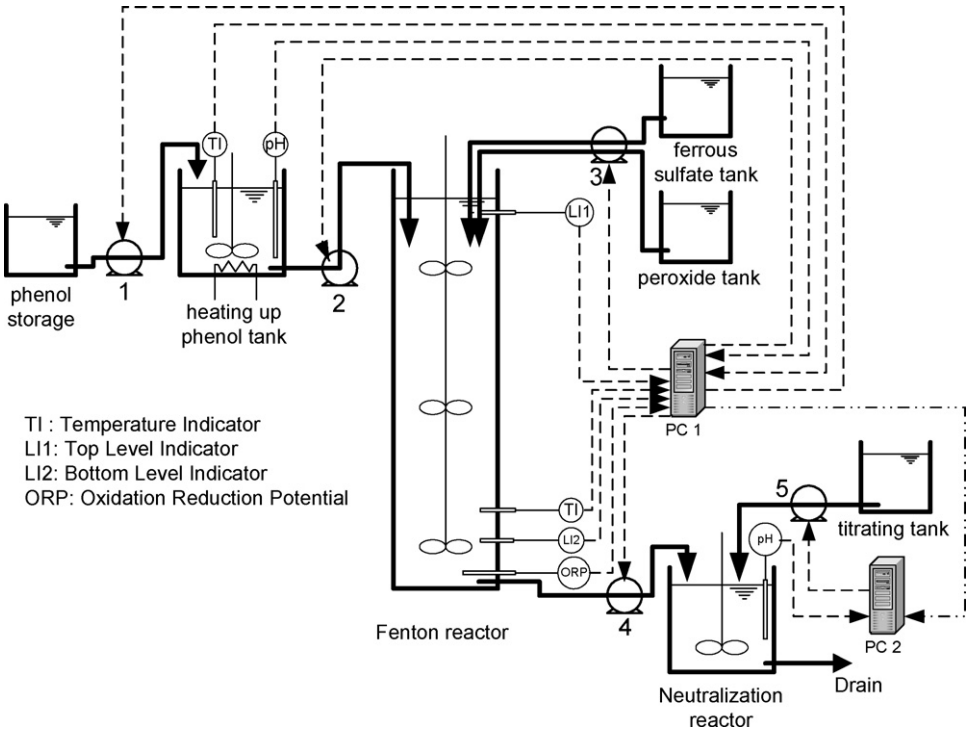


**Fig. 4.** Phenol decomposition laboratory pilot plant.

### 4.1. Plant description

The experimental setup used to test the proposed algorithms, shown in Fig. 4, consists of three processes, carried out in continuous stirred tank reactors (CSTRs):

- The *heating up tank* is used to maintain the temperature and pH of the phenol solution (process stream).
- The *Fenton tank* is used to control the phenol decomposition (oxidation) process by manipulating the hydrogen peroxide (oxidizer) and ferrous sulphate (catalyst) flow into the reactor (pump 3 in Fig. 4), at specified ORP levels (mV): this is the central control problem that will be solved using MFLC-MSA.
- The *pH neutralization reactor* is used to neutralize the pH of the yield from the second reactor (Fenton reactor) by manipulating the titrating stream (pump 5 in Fig. 4), which is an alkaline solution.

The objective of the experimental setup is to represent the control and instrumentation available in most industrial wastewater installations. Thus, simple instrumentation is used and independent controllers are used for each section of the process, as this is usually considered to give a more flexible, robust and cost-effective approach:

- A simple thermostat is used for control the temperature in the "heating up" tank (the process reacts optimally in the range of 80–90 °C, so tight temperature control in the "heating up" tank is seldom used).
- The proposed MFLC-MSA approach is used to control the main process (oxidation in the Fenton reactor) by using the local variables of the Fenton reactor and acting on the peristaltic pump 3.
- The original MFLC algorithm is applied to the neutralization process in the neutralization reactor (following [24,25]), by acting on the peristaltic pump 5, using the local variables of the neutralization reactor (and a binary variable that simply informs of the state of pump 4).

The MFLC-MSA and MFLC controllers were implemented in Matlab, acting on the system using the real-time toolbox for control.

The phenol synthetic wastewater, known as the process stream, which is about 1000 ppm, is prepared with variable concentrations. To have pH levels of the process stream in the desired range, the phenol solution is titrated with 1% hydrochloric acid (HCl). The catalytic solution, $FeSO_4$, is prepared for about 1% concentration. The oxidant reagent, hydrogen peroxide ($H_2O_2$), is used directly from the commercial product concentration, 30% $H_2O_2$ (w/w), without diluting it to a lower concentration. The phenol wastewater is exposed to hydrogen peroxide dosage in the presence of the ferrous catalyst at a given temperature and pH. As partial oxidation tends to lower the pH further, draining the treated stream to the natural environment without any neutralization procedure is dangerous and neither advisable nor prudent. So, following standard industrial practice, the pH is adjusted to near 7 with 1% sodium hydroxide (NaOH).

Clearly, this last operation is a pH control problem for strong acid–strong base systems, which is known to be extremely nonlinear. In this paper, the pH neutralization process is controlled using the proposed MFLC controller [23].

To guarantee phenol destruction through oxidation processes using Fenton's reagent, it is necessary to control the temperature, the pH and proper mix of the chemical species involved. Fenton's set of reactions must be conducted between 80 and 90 °C; therefore, the temperature of the solution is maintained using a thermostat. As mentioned in the introduction, the reaction is only efficient in a certain pH range: in this application the objective is to keep the pH between 2 and 4, which is achieved by titration with hydrochloric acid. Finally, treated wastewater on-specification discharge corresponds to a value of the oxidation–reduction potential of between 550 and 600 mV. This ORP goal band is standard in practice, because

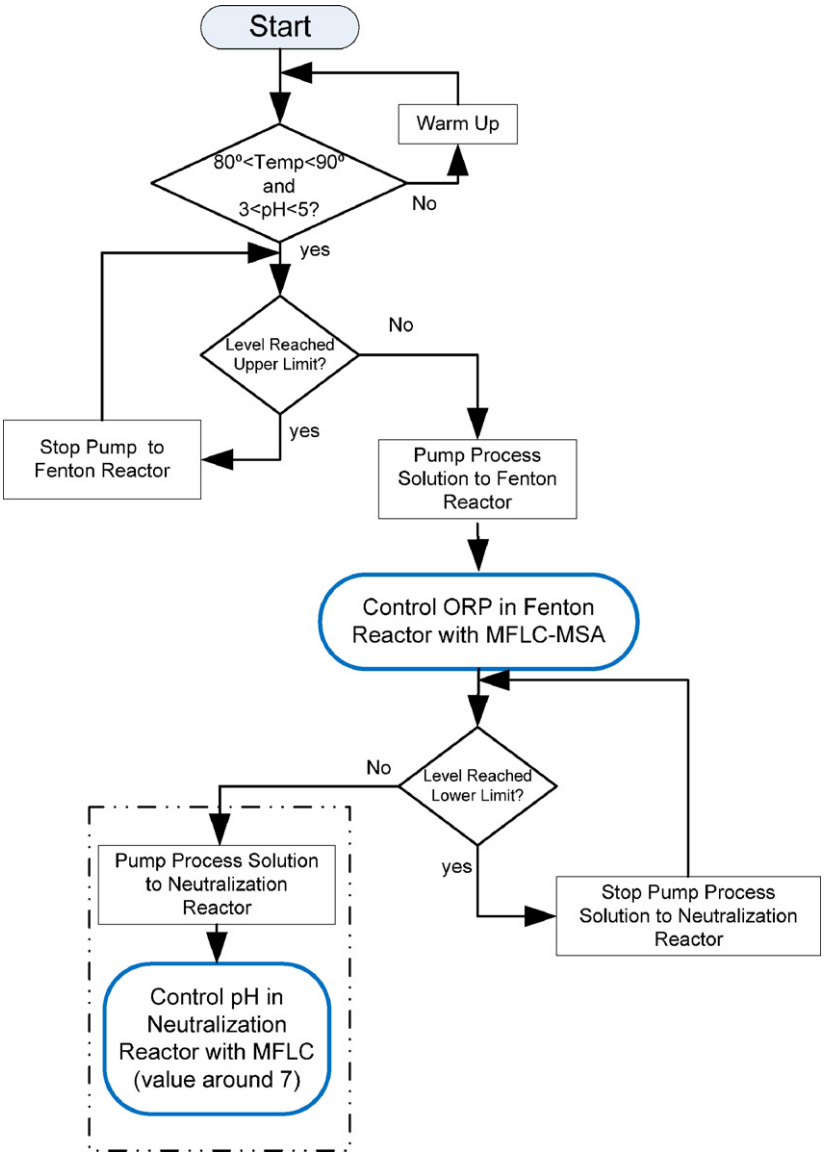


**Fig. 5.** Logic flow of oxidation process control.

it corresponds to a typical breakpoint in most oxidation processes [28].

### 4.2. Controller design

Fig. 5 presents the control logic of the proposed learning controller, showing that the first part of the control task is to regulate the temperature and the pH of the phenol solution before it reacts with Fenton's reagent. In this study, the temperature is controlled using a thermostat controller, because the range of valid temperatures of the process is wide, so it is not necessary to have a tight control on this task. Meanwhile, the pH value is maintained using the MFLC algorithms as discussed previously.

As mentioned above, apart from these main control objectives, control logic is needed to start and stop the process pumps when the maximum or minimum level limits are reached. The process pump (pump 2 in Fig. 4) is working if the liquid level of the process system has not reached the top level of the buffer tank, whereas, when the liquid level reaches the bottom level, the process pump (pump 4 in Fig. 4) stops.

To apply MFLC, it is assumed that the ranges of control and output signals are known from basic knowledge of the process. For example, the bounds on the increments on the control signal can be obtained experimentally by studying the characteristics of the pump, whereas the bounds on the output signal are known from the usual range of operation of the system. If they are not perfectly known, the best strategy is to use a worst-case approach, increasing the number of actions and states, and as a consequence the dimension of the $Q$-value. This increases the learning time and the memory requirements, so a trade-off might be reached.

#### 4.2.1. State selection

The main control objective is to drive and maintain ORP inside the goal band of 550 mV < ORP < 560 mV (see Fig. 2). Thus, following Section 3.1, the parameter $d$ is 5 mV. To define $f$ for this problem, it is assumed that the controller only learns when the output is within the band 455 mV < ORP < 655 mV (outside this band the system responses change significantly due to the different chemical reactions that are active). Thus, it is immediate to obtain that

$f = 100$ mV. The number of states over the goal was selected to be $N_a = 19$, so the width of each state is 5 mV.

#### 4.2.2. Control actions and reward function

To define the number of states $h$ is a trade-off: based on previous experiences on similar process control problems, the number of states from the goal band to the maximum ORP has been selected at $h = 20$, or equivalently, following (7), each state comprises $c = 5$ mV of the measured ORP. Selecting a smaller $h$ reduces the size of the matrix of $Q$-values, which gives smaller memory and computational requirements; however, this gives less precise learning, as part of the information from the measurement is lost.

The parameter $k$ is left to the control engineer as a tuning parameter (from (11) it can be seen that it is roughly equivalent to an integral gain of a nonlinear controller). Based on previous experiences with similar systems, $k = 0.0001$ was selected after a couple of trials. Fixing $k$ gives the number of total actions directly from (13), as it is known that in this process, the control signal should vary between samples in the range $0.0015 \leq \Delta u \leq 0.02$: thus the number of total positive actions is $N_a = 400$, so the dimension of the $Q$-table is 39 states by 801 actions. Each state has $n_a = 10$ possible actions. To increase the possible nonlinearity in the action-state mapping, the degree of overlapping was selected at $\beta = 100\%$ (lower values are known to be adequate for linear processes and higher values for highly nonlinear processes).

The goal of the control task is to maintain the process ORP in its goal state, or to return it to the goal state if there has been any disturbance or change to ORP reference values. To achieve this, maximum reward is given to actions causing the process to achieve or remain in the goal state. Therefore, the reward is given as: 1 for actions causing the next state to be in the goal state and $-1$ for actions not achieving the goal state in the resulting state following a control action:

$$reward = \begin{cases} 1 \text{ if the actions move the systems towards the goal state} \\ -1 \text{ otherwise} \end{cases} \quad (18)$$

Of course, more complex reward functions could be selected, but this particular reward function has been selected following the ideas given by [20], which recommends not to give the agent a detailed path to achieve the goal, but only the goal, because the path suggested as the most adequate might not really be the best (learning takes care of finding the most adequate approach).

The $\varepsilon$-greedy policy is used in this application for choosing an action at each visited state. Parameter $\varepsilon$, used in the $\varepsilon$-greedy policy, is selected to be 0.1, to leave space for the agent to explore other non-optimal available actions; so choosing a non-optimal action will be selected with a probability of 1 out of 10, which represents a good compromise for the plant, given its time-varying and nonlinear characteristic. Of course, this parameter might be selected to decrease when the system remains in the desired steady-state, but some space for the agent to explore must always be available.

Based on previous experience [23,24], the values of the *meta-parameters* for the MFLC-MSA and MFLC($Q$) agents are selected to be constant: discounted factor $\gamma = 0.98$ and learning rate $\alpha = 0.1$.

### 4.3. Experimental results and discussion

To reproduce realistic conditions, a phenol solution is prepared with variable concentrations (at around 2000 ppm). The process stream (phenol solution), in the pH range of 2.0–4.0, is titrated with 10% hydrochloric acid (HCl). The catalytic solution, $FeSO_4$, is prepared for about 2% (w/w) concentration. The $H_2O_2$ is used directly from the commercial product (30%, v/v $H_2O_2$). To neutralize the treated stream, it is titrated with an aqueous solution of 2% (w/w) pf sodium hydroxide (NaOH). As no simulation of the process is available, the learning control strategy was carried out directly from on-line interaction with the plant.

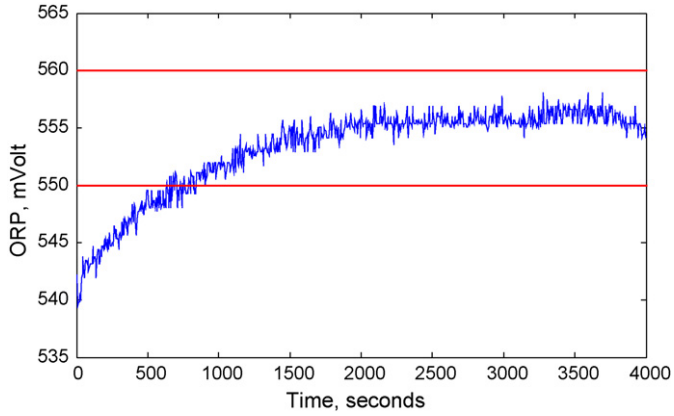


**Fig. 6.** Measured ORP using MFLC(MSA), with the objective of 550 <ORP < 560, starting at ORP = 540.

#### 4.3.1. ORP regulation

The responses of the application of the proposed MFLC-MSA to the phenol decomposition process can be seen in Fig. 6, which clearly shows that the proposed MFLC-MSA controller learns to maintain the oxidation process within the desired ORP control band (550–560 mV). The first 600 s correspond to the training phase, where the agent learns the environment in order to keep it within the goal band. The initial phase reaction also occurs during this training phase, when the sequence reactions consume both oxidizer and catalyst. After the balancing reactions are reached, then the responses of the process reach and stay in the goal band. Clearly, the controller regulates the system by increasing the control signal and maintaining the previous control signal when the process is in the goal band, as can be seen in Fig. 7.

The reason for selecting the set point for the oxidation process at a 555 mV ORP reading is that the optimal decomposition of the phenol contaminant using Fenton's reagent is in the range 550–600 mV ORP. Therefore, if this range is taken as a reference, the optimal reaction is in this range, and the responses of the process are most of the time within the optimal range of the reaction. Clearly, the process is oxidized in the optimal region of the decomposition reaction.

The control signal (Fig. 7) shows that when the process is outside the goal band, the agent provides a control signal by exploring the available actions (Fig. 8) in every visited state. The agents are allowed to provide the action that satisfies the incremental constraints. Due to the incremental constraints, the agent cannot quickly provide sufficient actions to move the process quickly to the goal band.

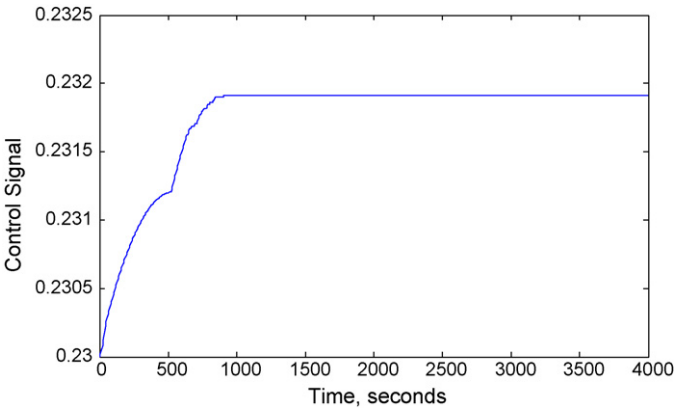


**Fig. 7.** Control signal for MFLC(MSA): hydrogen peroxide added, normalized from 0 to 1.

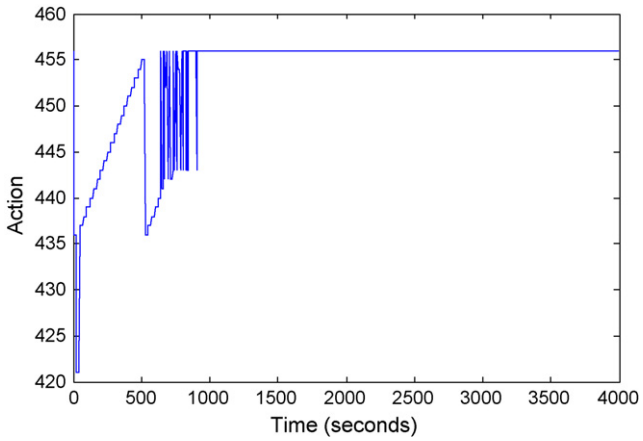


**Fig. 8.** Index of selected actions, corresponding to Fig. 7.

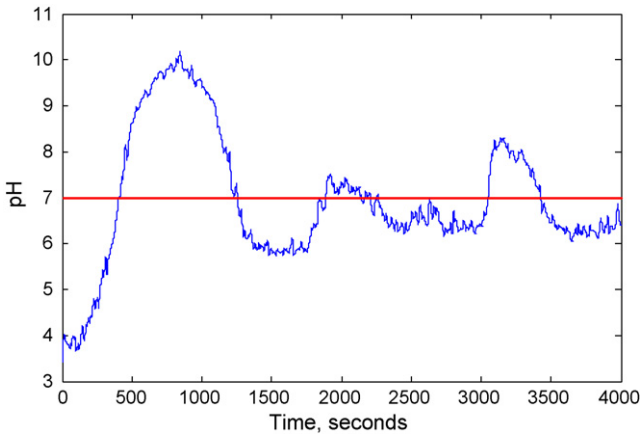


**Fig. 9.** pH responses of the yield reaction neutralization.

*4.3.2. pH regulation*

The second agent, based on a standard MFLC algorithm, controls the second part of the process, which is to neutralize the pH of the effluent from the buffer tank, following the ideas first proposed by [23]. This agent regulates the titrating flow to the neutralization reactor and also interacts with the first agent to control the oxidation process, as mentioned before. In this application, the second agent has 21 symbolic states and 205 actions.

The effluent of Fenton's reaction is known as the acid flow. As Fenton's reaction is optimally decomposed in an acid solution, the phenol solution is titrated with strong acid for 3–5 pH value. Therefore, the effluent is a strong acid solution and is titrated with strong base solution (NaOH). Thus, the process is a strong acid and strong base system, which is known as a highly nonlinear process. Even though the system is highly nonlinear, the MFLC can regulate the process to keep it within the goal band, as shown in Figs. 9 and 10: the responses to applying the second agent for the neutralization of the effluent from the buffer tank are plotted in Fig. 9: the process is maintained within the goal band. This MFLC controller regulates the control signal by increasing or reducing the titrating flow (NaOH), shown in Fig. 10.

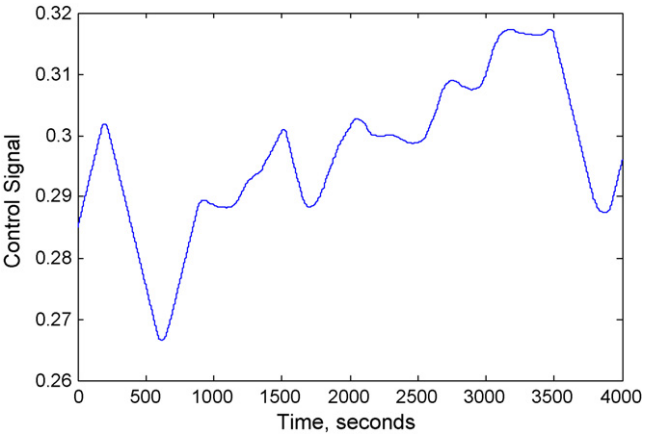


**Fig. 10.** pH control signal of the yield reaction neutralization.

## 5. Conclusions

A solution to the phenol destruction problem in industrial wastewater has been presented, based on the regulation of the ORP in a reactor, using the proposed MFLC-MSA algorithm and the Fenton process, which is followed by pH neutralization in a separate buffer tank, also controlled by the proposed algorithm.

The MFLC-MSA algorithm presented is based on a *Q*-learning look-up table and multi-step actions. This automation strategy is appealing for this kind of process, as they are time-varying, nonlinear and it is not practical to derive a precise model. The use of multiple step actions makes it possible to eliminate unnecessary calculation and makes learning faster.

The proposed algorithm has been applied to a real process at laboratory scale: we have shown how the controller parameters can be easily obtained from control requirements. The experimental results obtained show that the combination of the two controllers makes it possible to maintain the treated stream within specifications, allowing its safe discharge to the environment.

It must be pointed out that the proposed approach is quite general, so it is very adequate to solve problems in process control when it is not feasible to use a process model (for example, it has already been applied by the authors to thermal food processing [26]). In particular, it is especially adequate for other wastewater treatment processes, as the variations in concentration and type of products to be treated makes deriving models unfeasible.

## References

[1] R.J. Bigda, Consider Fenton's chemistry for wastewater treatment, Chem. Eng. Prog. 91 (1995) 62–68.
[2] J.A. Boyan, A.W. Moore, Generalization in reinforcement learning: safely approximating the value function, Adv. Neural Inf. Process. Syst. 7 (1995) 369–376.
[3] K.C. Chen, C.Y. Chen, J.W. Peng, J.Y. Houng, Real time control of an immobilized-cell reactor for wastewater treatment using ORP, Water Res. 36 (2002) 230–238.
[4] J. De Laat, H. Gallard, Catalytic decomposition of hydrogen peroxide by Fe(III) in homogeneous aqueous solution: mechanism and kinetic modelling, Environ. Sci. Technol. 33 (1999) 2726–2732.
[5] B. Li, P.L. Bishop, The application of ORP in activated sludge wastewater treatment processes, Environ. Eng. Sci. 18 (2001) 309–321.
[6] N. Kang, D.S. Lee, J. Yoon, Kinetic modeling of Fenton oxidation of phenol and monochlorophenols, Chemosphere 47 (2002) 915–924.
[7] M. Kearns, S. Singh, Finite-sample convergence rates for Q-learning and indirect algorithms, Adv. Neural Inf. Process. Syst. 11 (1999) 996–1002.
[8] W.P. Kwan, Decomposition of hydrogen peroxide and organic compounds in the present of iron and iron oxides, PhD thesis, MIT, 2003.
[9] J.H. Lee, J.M. Lee, Approximate dynamic programming based approach to process control and scheduling, Comput. Chem. Eng. 30 (2006) 1603–1618.
[10] J.M. Lee, J.H. Lee, Approximate dynamic programming based approaches for input–output data-driven control of nonlinear processes, Automatica 41 (2005) 1281–1288.
[11] L.L. McPherson, Understanding ORP's in the disinfection process, Water Eng. Manage. 140 (1993) 29–31.
[12] E. Paul, S. Plisson-Saune, M. Mauret, J. Cantet, Process state evaluation of alternating oxic-anoxic activated sludge using ORP, pH and DO, Water Sci. Technol. 38 (1998) 299–306.
[13] Y.Z. Peng, J.F. Gao, S.Y. Wang, M.H. Sui, Use pH and ORP as fuzzy control parameters of denitrification in SBR process, Water Sci. Technol. 46 (2002) 131–137.
[14] D. Precup, Temporal abstraction in reinforcement learning, PhD thesis, University of Massachusetts, 2000.

[15] C.S. Ra, K.V. Lo, D.S. Mavinic, Real-time control of two-stage sequencing batch reactor system for the treatment of animal wastewater, Environ. Technol. 19 (1998) 343–356.
[16] Z. Rappoport, The Chemistry of Phenols, Wiley, New York, 2003.
[17] M. Riedmiller, High quality thermostat control by reinforcement learning—a case study, in: Conald Workshop, Carnegie Mellon University, USA, 1998.
[18] R. Schoknecht, M. Riedmiller, Learning to control at multiple time scales, in: ICANN 2003, Istanbul, Turkey, 2003, pp. 479–487.
[19] W.C. Schumb, C.N. Satterfield, R.L. Wentworth, Hydrogen peroxide, in: ACS Monograph Series, Reinhold, New York, 1955.
[20] W.D. Smart, Making reinforcement learning work on real robots, PhD thesis, Brown University, 2002.
[21] R.S. Sutton, A.G. Barto, Reinforcement Learning: an introduction, The MIT Press, Cambridge, MA, 1998.
[22] R.S. Sutton, D. Precup, S. Singh, Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning, Artif. Intell. 112 (1999) 181–211.
[23] S. Syafiie, F. Tadeo, E. Martinez, Softmax and $\varepsilon$-greedy policies applied to process control, in: IFAC Workshop on Adaptive and Learning Systems (ALCOSP04), Yokohama, Japan, 2004, pp. 729–734.
[24] S. Syafiie, F. Tadeo, E. Martinez, Learning to control pH processes at multiple time scales: performance assessment in a laboratory plant, Chem. Prod. Process Model. 2 (2007).
[25] S. Syafiie, F. Tadeo, E. Martinez, Model-free learning control of neutralization process using reinforcement learning, Eng. Appl. Artif. Intell. 20 (2007) 767–782.
[26] S. Syafiie, C. Vilas, M.R. Garcia, A.A. Alonso, E. Martinez, F. Tadeo, Intelligent control based on reinforcement learning for batch thermal sterilization of canned foods, in: IFAC World Congress, Seoul, Korea, 2008.
[27] C.J. Watkins, C.H. Watkins, P. Dayan, Q-learning, Mach. Learn. 8 (1992) 279–292.
[28] R.-F. Yu, Feed-forward dose control of wastewater chlorination using on-line pH and ORP titration, Chemosphere 56 (2004) 973–980.
[29] J.A. Zazo, J.A. Casas, A.F. Mohedano, M.A. Gillarranz, J.J. Rodríguez, Chemical pathway and kinetics of phenol oxidation by Fenton's reagent, Environ. Sci. Technol. 39 (2005) 9295–9302.
[30] M. Fiter, D. Güell, J. Comas, J. Colprim, M. Poch, I. Rodríguez-Roda, Fuzzy logic control for saving energy in a biological nitrogen removal process, in: J. Vitra, et al. (Eds.), Recent Advances in Artificial Intelligence Research and Development, IOS Press, 2004.