MODEL-FREE INTELLIGENT CONTROL USING REINFORCEMENT LEARNING AND TEMPORAL ABSTRACTION-APPLIED TO pH CONTROL

S. Syafiie*[#], F. Tadeo*, and E. Martinez**

**Department of Systems Engineering and Automatic Control*
Science Faculty, University of Valladolid
Prado de la Magdalena s/n., 47011 Valladolid. Spain
Email: {syam,fernando}@autom.uva.es
***Consejo Nacional de Investigaciones Científicas y Técnicas*
Avellaneda 3657 3000, Santa Fe, Argentina
Email: ecmarti@ceride.gov.ar

Abstract: This article presents a solution to pH control based on model-free intelligent control (MFIC) using reinforcement learning. This control technique is proposed because the algorithm gives a general solution for acid-base system, yet simple enough for its implementation in existing control hardware. In standard reinforcement learning, the interaction between an agent and the environment is based on a fixed time scale: during learning, the agent can select several primitive actions depending on the system state. A novel solution is presented, using multi-step actions (MSA): actions on multiple time scales consist of several identical primitive actions. This solves the problem of determining a suitable fixed time scale to select control actions so as to trade off accuracy in control against learning complexity. The application of multi-step actions on a simulated pH process shows that the proposed MFIC learns to control adequately the neutralization process.

Keywords: learning control, intelligent control, online learning, agents, process control, neutralization process, pH control, temporal abstraction, model free

1. INTRODUCTION

The control of pH in neutralization processes is a ubiquitous problem encountered in chemical and biotechnological industries. For example, the effluent of streams from wastewater treatment plants must be maintained within stringent environment limits, using neutralization to eliminate undesirable compounds (Shinskey, 1973).

In most neutralization processes the control of pH is not only a control problem but also the chemical equilibrium, kinetic and thermodynamic problems must also be considered (Gustafsson *et. al.*, 1995). This characteristic makes difficult to control pH process. Another problem is the process buffer capacity, which is unknown and dramatically changes process gain.

Also, due to the nonlinear dependence of the pH value on the amount of titrated agent the process will be inherently nonlinear. Moreover, variations of the buffering effects could make the process time-varying. Therefore, it is difficult to develop an appropriate mathematical model of the pH process for controller design. All of this makes the process difficult to control with classical process control techniques (Loh *et. al.*, 1995).

Several control strategies have been previously applied for neutralization processes; for instance, Fuzzy Control (Fuente *et al.*, 2003), Fuzzy Internal Model Control (Edgar and Postlethwaite, 2000), Fuzzy Predictive Control (Biasizzo *et al.*, 1997), and Neural Networks (Loh *et al.*, 1995). Unfortunately, in

[#] corresponding author

these approaches there are some weaknesses, such as:

- complexity of the control structures (which could be difficult to implement on existing control systems),
- conservativeness (the controllers take a long time to reject disturbances and to reach the desired setpoint),
- difficulty of tuning, which makes it a time-consuming task (these controllers have many tuning parameters, or require many experiments before its application to a real industrial process).

This paper presents an alternative solution to pH control based on Model-Free Intelligent Control (MFIC). This control technique is proposed because the algorithm gives a general solution for acid-base system, simple to implement in existing control hardware and easy to handle. This alternative approach is used to solve the pH control process problem, which is based on applying *reinforcement learning* algorithms.

2. REINFORCEMENT LEARNING

Reinforcement Learning can be defined as '*learning what to do by doing'*, i.e. how to map perceptions of process states to control actions, so as to maximize an externally provided scalar reward signal. These algorithms are based on online learning directly from the closed-loop behaviour of the plant. Compared to other control techniques based on learning, the reinforcement learning approach to model-free control design has some clear advantages:

- It is possible to put in the design of the controller previous knowledge of the system
- The control algorithm is quite simple from a computational point of view, so it is feasible to implement using low-cost hardware.
- It is possible to derive and obtain a feedback control law from an optimal control point of view based on actual experience rather than a process model, which makes it attractive for Control and Plant Engineers

In standard reinforcement learning algorithms, like *Q*-learning, there is a difficulty for Process Control implementations: these algorithms scale very badly with increasing problem size, granularity of states or control actions. Among others, one intuitive reason for this is that the number of decisions from the start state to the goal state increase exponentially.

According to the problem size, to keep tractable the number of decision to be taken to reach the goal state hierarchical approaches based on *temporal abstraction* have been proposed. Temporal abstraction can be defined as an explicit representation of *extended actions*, as policies together with a termination condition (Precup, 2000). The original one-step action is called *primitive action*. Semi Markov Decision Processes (SMDPs) is the theory used to deal with temporal abstraction as a minimal extension of reinforcement learning framework. SMDPs is a Markov Decision Processes (MDP) appropriate for modeling continuous-time discrete-event systems.

Several reinforcement learning algorithms resorting to hierarchical temporal abstraction approaches have been proposed: Options (Sutton *et al.*,1999); Hierarchy of Abstract Machine (HAM) (Parr, 1998); MaxQ (Dietterich, 1997) and Multi-step actions (MSA) (Riedmiller, 1998). The first three methods are based on the notion that the whole task is decomposed into subtasks each of which corresponds to a subgoal.

3. MULTI-STEP ACTIONS

In this paper the concept of MSA (Schoknecht and Riedmiller, 2003) is applied to pH control because it is suited for systems where no decomposition in subproblems is known in advance. As in the general framework defined by Sutton *et al.* (1999), MSA is a special type of semi-Markov option. A Markov option would require a state-dependent termination condition. In the MSA algorithms, the termination condition is applied after executing a sequence of *n* primitive actions.

The MSA method is a method enabling an intelligent control to learn a control policy by using multiple time scales simultaneously. The MSA consists of several identical actions on the primitive time scale. This algorithm is possible to increase responsiveness and add flexibility to the controller behavior. Also, giving a learning controller the possibility of using MSA to reach the goal can improve the speed of learning and reduce control efforts. This approach have been successfully applied in a simple thermostat control (Riedmiller, 1998; Schoknecht and Riedmiller, 2003). Thus, we think that the algorithm can be extended to complex and highly nonlinear problem, such as pH control problem.

The MFIC based on MSA can address many of weaknesses inherent in traditional PID or other advanced control methods. For example:

- PID is basically linear and time-invariant and cannot effectively control complex processes that are nonlinear, time variant, coupled, and have large time delays, major disturbance and uncertainties.
- Model based adaptive control methods have some problems such as requirement of off line training, excitation of signal for correct identification and the model convergence and the system stability issues in real application.
- Model predictive control is designed using empirical model of the plant, so this controller cannot be used for general acid-base systems

because each model only represents the specific process.

4. *Q*-LEARNING WITH MULTI-STEP ACTIONS

This paper proposes the application of the MSA algorithm for pH control. This proposed solution can be used for general-purpose of acid-base system control, providing the user-friendly and smart control system that users demand. The advantages of MFIC based on MSA are:

- no precise quantitative knowledge of the process is available,
- no process identification mechanism (identifier is include in the system, which is an online learning), no controlled design for a specific process is needed,
- simple manual tuning of controller parameter is required and stability analysis criteria are available to guarantee the closed-loop system stability.

The idea of MSA is based on a set of all multiple step actions of degree *m,* defined as $A^{(m)} = \left\{a^m \mid a \in A^{(1)}\right\}$, where a^m denotes the MSA that arises if action a is executed in m consecutive time steps (Schoknecht and Riedmiller, 2003). The next action will be executed after the whole MSA has been applied. Thus, the MSA has a time-dependent termination condition after m primitive time steps.

The concept of MSA can be integrated in learning algorithms, such as *Q*-learning. For example, when the agent executes action a^m of degree m in state s the environment makes transition to state s_n after m time steps. The state-action value can be updated as follows:

$$Q_{sa^m}^{t+1} \leftarrow Q_{sa^m}^{t} + \alpha\left[r_{sa^m} + \gamma^m \max_{a_n \in A} Q_{s_n a_n}^{t} - Q_{sa^m}^{t}\right] \quad (1.)$$

where

$$r_{sa^m} = \sum_{\tau=i}^{i+m-1} \gamma^{\tau-i} r_{s_\tau a}$$

where Q_{sa}^{t} is *Q*-value for state-action in time *t,* and $Q_{s_n a_n}^{t}$ is *Q*-value for next state, α is learning rate, and γ is discount factor. When action a^m with degree m is selected in s_i, the environment makes transition to s_{i+m} with reward $r_{s_i a^m}$. When executing a^m, all actions a_i, $i = 1, 2, \ldots, m-1$ are executed implicitly. The transition from s_i to state s_{i+m} contains all information necessary to update the *Q*-values for those lower-level actions at all intermediate states.

Compared to standard reinforcement learning, this *Q*-learning-modification algorithms is proposed for pH process because it can extract more training examples from the same experiences. The agent executes a primitive action and applies it for m time steps.

Reinforcement learning based on MSA seems to be a promising approach to overcome the pH problem as mentioned above because the control law can easily adapt to varying scenarios by online learning. By experiencing a sequence identical action applying for pH process, the agent can speed up learning and planning to maintain the process in the desired pH value. In order to explore the set of possible actions and acquire experience through the reinforcement signals, the actions are selected using an exploration/exploitation policy. In this study *ε-greedy policy* is applied to select one of the available actions in visited state and experience it for a multiple time steps of the plant. The ε-greedy policy has been selected because it gives better performance for pH process than *softmax* policy (Syafiie *et al.*, 2004).

5. APPLICATION TO A NEUTRALIZATION PROCESS

This section describes the implementation of the MSA approach on a pH neutralization process.

5.1 States and Reward

The control objective is to maintain the pH inside a band of ±δ around the desired setpoint (the width of this band is defined by measurement noise in the process and allowed tolerance). This band is defined as the goal state. Another states are defined corresponding to values of the pH outside this band, as depicted in Figure 1.

To classify the reading of pH and to select an action available in each visited state, this study uses 11 symbolic states, where the goal state is 6 (corresponds to the desired pH band). Each state has 5 possible actions, except in the goal state that has only 1 action, which is called *wait action*. This is selected by experience.

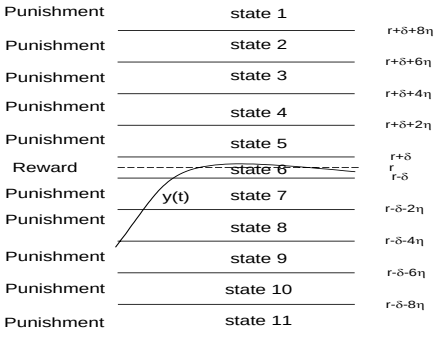

Figure 1. Control objective and definition of states.

The probability that the system moves to a new state from the current state depends on the system behavior following the execution of the chosen action. For instance, if the process is in state 1, and the controller chooses action 1, the process may move either to state 2 or to another state or stay in state 1.

These states and reward are defined by a parameter that refers to the setpoint, r, as a desired output. The goal state is restricted by boundary values: upper, $r+\delta$, and lower, $r-\delta$ as shown in Equation 2. The maximum reward is introduced in the goal state. When the system is outside the desired band, the controller is punished by negative rewards. This reward function is applied in each state as a single number:

$$reward = \begin{cases} 1 & if\ r-\delta < pH < r+\delta, \\ -1 & else \end{cases} \quad (2.)$$

5.2 Control Actions

In reinforcement learning, the agent selects an action and executes it in current time and receives next reward. From the chosen action, the control signal, u_t, is calculated in MFIC as follows:

$$u_t = u_{t-1} + k(a_w - a_t) \quad (3.)$$

where a_t is the optimal action chosen by the agent from those available actions in every visited state, and a_w is *wait action* where there is no variation of the previous control signal. For example, if the system has three actions, where action one is to increase the previous control signal, action two is to maintain the previous control signal and action three is to decrease it. Therefore, action two is called wait action. The controller gain, k, is a tuning parameter that can be selected to weight how much to increase or decrease previous control signal over action chosen.

MFIC uses zero initial condition of Q-function. This value is updated by time over taking action and the process behavior. When the environment changes, for example, the setpoint changes, the action-value is immediately reset to initial condition. Resetting Q-value to the initial condition makes possible for the agent to learn new environment without any influence from the past learning of the past environment.

5.3 Control Algorithm

The MSA algorithm developed for the learning system is as follows:

1. Read the state s_t
2. Select an action a_t, this action is chosen from state s_t using ε-greedy policy
3. Apply the selected action a_t for n time steps
4. Do until terminating condition $m=n$
    1. Read the resulting state s_{t+m}
    2. Update Q-value using Equation (1)

6. EXPERIMENTAL RESULTS AND DISCUSSION

The experimental results section describes and discusses the application of MFIC based on reinforcement learning to a pH process control on the laboratory plant.

6.1 Description of the Experimental Setup

The experimental setup, shown in Figure 2, consists of a continuous stirred tank reactor (CSTR) where a process stream (sodium acetate) to be maintained at certain pH value is titrated with a solution of hydrochloric acid (HCl). The solution of sodium acetate (CH_3COONa) is prepared and stored in a storage tank. Concentration and pH value of sodium acetate can be achieved by adding varying amounts into the storage tank. This solution is fed from storage tank using a pump. The reaction occurs in the CSTR which has overflows (outlet not shown); therefore the volume of liquid in the tank (1 liter) can be considered constant.

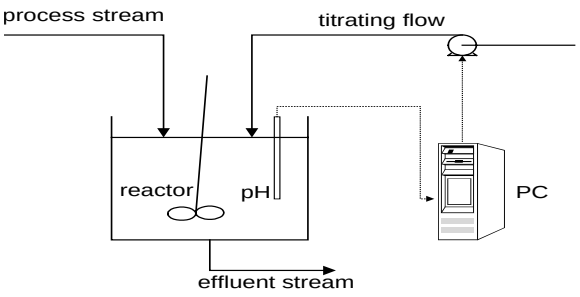

Figure 2. pH neutralization process plant. It is to control pH value of the process stream (alkaline) manipulating titrating flow (acid).

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. It is assumed that the mixing is homogeneous, therefore the concentration in the effluent stream is similar to the concentration 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 from a personal computer, using Matlab and the Real-Time Toolbox for online control.

6.2 Parameters Selection

For the experimental process, the zero initialized *Q*-value is used. The value of the *meta – parameter* for the agent are selected to be: discount factor, γ=0.98 and learning rate, α=0.1, which were determined to be a good values from previous work (Syafiie *et al.*, 2004).

As mentioned above, the defined system has eleven states. In this implementation, the parameters δ and η are selected to be $\delta=0.1$ and $\eta=0.1$, based on the level of measurement noise and the desired operating range of pH. From the parameters δ and η, it can be defined that state 1 is when the measured pH is higher than $r+\delta+8\eta$. It means that the agent is in state 1 if the pH is higher than $r+\delta+8\eta$. State 2 is defined when the pH is lower than $r+\delta+8\eta$ and higher than $r+\delta+6\eta$. The rest of the states are defined following Figure 1.

In MFIC, the ε-greedy policy, is applied for choosing an action in every visited state of the pH process. Parameter ε used in the ε-greedy policy is selected to be ε=0.1, to leave space for the agent to explore the available actions. This means that exploration (choosing an action that does not have maximum action-value) 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 (less experience would be necessary if the plant were more linear and the concentration less uncertain).

The PID controller was tuned based on operating condition at pH = 5, where correction gain and proportional gain are chosen 0.01 and 0.001 respectively. Derivative time and integral time are selected to be 1. Whereas for MFIC, the wait action is chosen to be 22, which is no manipulation of previous control signal. The gain of MFIC is chosen $2\text{x}10^{-5}$, and 3 identical primitive actions are executed in every multiple time scale.

6.3 Experimental Results and Discussion

Application of the proposed MFIC of MSA controller to the laboratory plant shows good result. The responses of the plant for some changes in setpoint and comparison to a PID and standard *Q*-learning controller can be seen in Figure 3 for the sodium acetate – hydrochloride acid system.

The comparison shows that the responses of the proposed MFIC of MSA algorithm settle in reference faster than the PID and standard *Q*-learning controller. The responses of the plant show that MSA controller based on reinforcement learning algorithms are lay closer to the references, whereas PID and standard *Q*-learning controllers have higher peak of oscillations. It can be seen in Figure 4.

The control signal (Figure 5) shows that MSA and standard *Q*-learning controllers manipulate the actuator smoother than PID controller: MSA and original *Q*-learning controllers have smoother control signals than PID. Since MFIC allows a tolerance error of the process whenever the pH is within the control band, the control signal is smoother when the process is closer or within the pH band.

7. CONCLUSIONS

The Model-Free Intelligent Control (MFIC) algorithm based on multi-step actions (MSA) has been extended to process control problems and applied on a pH control. The optimal control actions are selected using the ε-greedy policy. It has been shown that the behavior of the pH control over the application of multi-step actions gives good performance. It is noteworthy the smoothness of the resulting control signal. Thus, the proposed technique is promising for pH process.

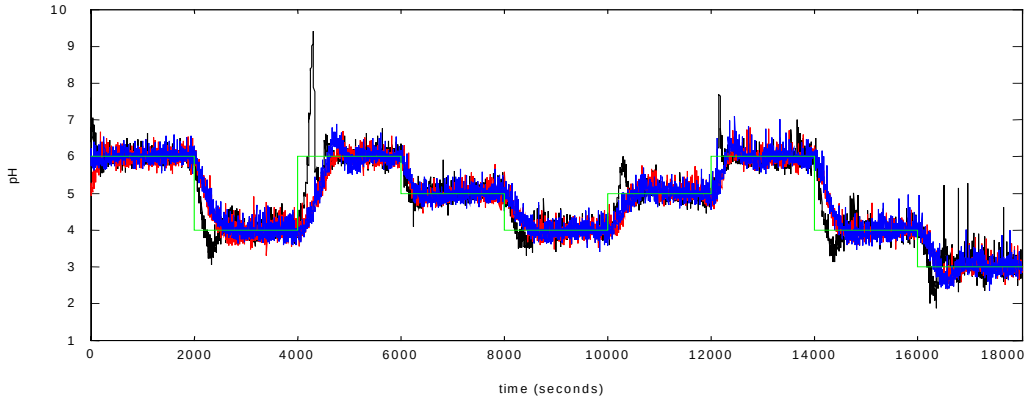

Figure 3. Output Responses of the plant for NaCH3COO-HCl systems.

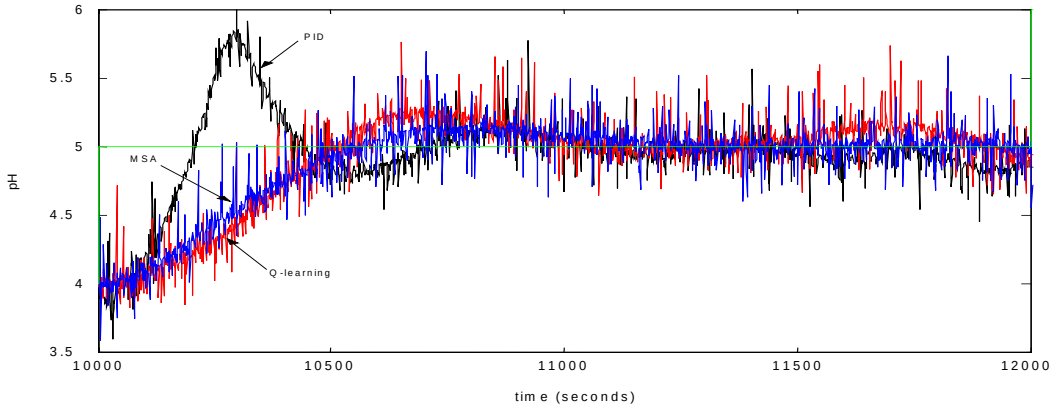

Figure 4. Output Responses of the plant for NaCH3COO-HCl systems for 10000 to 12000 seconds.

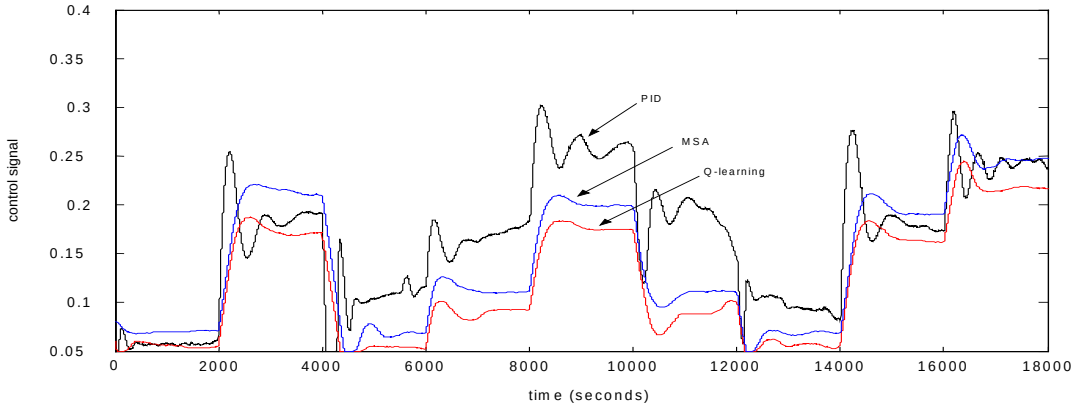

Figure 5. Control signal of the plant for NaCH3COO-HCl systems.

Acknowledgment

This work was funded by MCYT-CICYT (DPI2004-07444-C04-02) and the first author would like to thank the MAE-AECI for financial support.

REFERENCES

Biasizzo, K. K., I. Skrjanc, and D. Matko (1997), Fuzzy predictive control of highly nonlinearity pH process, *Computers and Chemical Engineering*, **Vol. 21,** pp. s613 – s618

Edgar, C. R., and B. E. Postlethwaite (2000), MIMO fuzzy internal model control, *Automatica*, **Vol. 34,** pp. 867 – 877

Fuente, M. J., C. Robles, O. Casado, S. Syafiie, and F. Tadeo (2002), Fuzzy control of neutralization process, IEEE CCA 2002, Glasgow.

Gustafsson, T.K., Skrifvars B. O., Sandström K.V., Waller K. V. (1995), Modeling of pH for Control, *Industrial Engineering Chemical Research*, **Vol. 34**, pp. 820-827

Loh, A. P., K. O. Looi, and K. F. Fong (1995), Neural network modeling and control strategies for a pH process, *Journal of Process Control*, **Vol. 6**, pp. 355 – 362

Parr, R. (1998), *Hierarchical Control and learning for Markov decision processes*, PhD thesis, University of California at Berkeley

Precup D. (2000), *Temporal Abstraction in Reinforcement Learning*, Ph.D. Dissertation, Department of Computer Science, University of Massachusetts, Amherst.

Riedmiller M. (1998), High quality thermostat control by reinforcement learning-A case study, in *Proceedings of the Conald Workshop* 1998, Carnegie Mellon University.

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

Shinskey, F. G. (1973), *pH and pIon Control in Process and Waste Stream*, Wiley, New York.

Sutton, R.S., Precup, D., Singh, S. (1999). Between MDPs and semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning. *Artificial Intelligence,* **Vol. 112**, No.1 –2, pp.181-211

Syafiie S., F. Tadeo, and E. Martinez (2004c), Softmax and ε-Greedy Policies Applied To Process Control, *IFAC Workshop on Adaptive and Learning Systems (ALCOSP04)*, Yokohama, Japan, August 2004, pp.729 – 734.