BULETINUL Universităţii Petrol – Gaze din Ploieşti	Vol. LXI No. 3/2009	193 - 198	Seria Tehnică

Using Reinforcement Learning in the Adaptive Fuzzy Controller

Gabriela Moise, Florentina Alina Chircu

Petroleum – Gas University of Ploieşti, 39 Bucureşti Blvd., Ploieşti, ROMÂNIA
e-mail: gmoise@upg-ploiesti.ro, chircu_florentina@yahoo.com

Abstract

In this paper, the authors present their researches on the learning mechanism of an adaptive fuzzy controller. The proposed method consists in using reinforcement learning techniques in the learning mechanism so that the rules base of the controller's knowledge base is updated to maintain the system's performance. The algorithm is a general algorithm, and therefore can be applied in a large variety of problems. Experimental results are provided using MATLAB® software for a study case.

Key words: *adaptive fuzzy control, Q-learning.*

Introduction

An adaptive controller is a controller with adjustable parameters and an automatic mechanism to adjust these parameters [1].

"An adaptive controller is therefore intuitively a controller that can modify its behavior after changes in the controlled plant or in the environment. Some objectives for using adaptive control are the character of the plant varies, or the character of the disturbances varies" [2].

There are two general structures for adaptive controller. A former type of structure contains an adaptation mechanism, which observes the evolution of the reference and changes the parameters of the controller to maintain the system's performance in the conditions of acting perturbation over process. The controller acts in the sense of obtaining a reference model, in the conditions of perturbations' actions. The model is called direct adaptive regulator.

The latter structure consists in estimations of the process' parameters and a design module of the controller to specify the future parameters of the process. This model is known as indirect adaptive controller.

According to the components of a fuzzy controller (the fuzzifier and defuzzifier block, fuzzy inference block and rules base block), this has the following elements, which can be altered to obtain different behaviors of the controller [6]: the scaling factors for inputs and outputs variables, the membership functions of linguistic values: type of functions, the rules base.

Detailing, an adaptive fuzzy regulator contains the following changeable elements: the fuzzy partition of inputs and outputs variables, the type and parameters of membership functions, the type of fuzzy implication, the aggregation rules of the rules base, the degree of contribution of a particular fuzzy rule in the fuzzy controller, the consequence of each rule within the rules base, the parameters used in the pre-processing block and post-processing block.

The FMRLC Model

The Fuzzy Model Reference Learning Controller (FMRLC) is a direct adaptive learning controller that learns to control the process, namely to solve new situations and memorize how it did it. The general schema of FMRL is presented in figure no. 1.

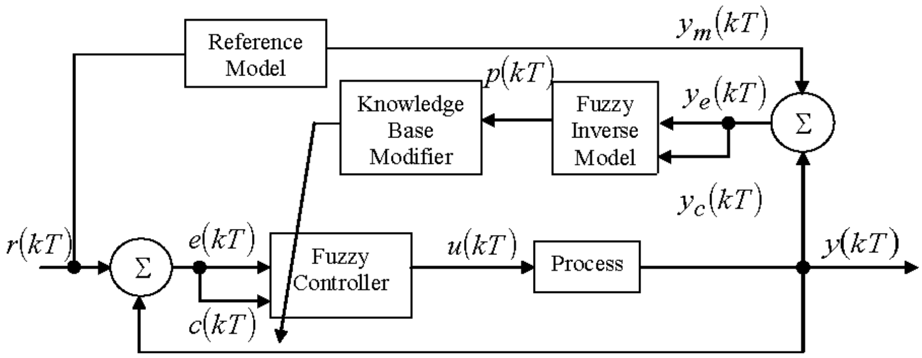

Fig. 1. General Schema of FMRLC.

The functional components of FMRLC are as follows:

1. Fuzzy controller

The inputs to the fuzzy controller are error and change of error (1).

$$e(kT) = r(kT) - y(kT)\ ,\ c(kT) = \frac{e(kT) - e(k(T-1))}{T}\ , \tag{1}$$

where $r(kT)$ is the reference input and $y(kT)$ is the process output.

The fuzzy controller contains a pre-processor block, a post-processor, a knowledge base and a mechanism of inference.

2. The reference model

The reference model provides the performance specifications. The reference model characterizes design criteria [4]: stability, rise time, overshoot, settling time. The input of the reference model is the reference $r(kT)$ and the output is the desired performance $y_m(kT)$ of the controlled process.

3. Fuzzy inverse model and knowledge base modifier

The fuzzy inverse model and knowledge define the learning mechanism. The learning mechanism tunes the rules base of the direct fuzzy controller, so that the closed loop system behaves like the reference model. Fuzzy inverse model maps the $y_e(kT)$ and $y_c(kT)$ to $p(kT)$. The variables $y_e(kT)$, $y_c(kT)$ and $p(kT)$ are computed as follows (2, 3):

$$y_e(kT) = y_m(kT) - y(kT), \tag{2}$$

$$y_c(kT) = \frac{y_e(kT) - y_e(k(T-1))}{T}. \tag{3}$$

The variable $p(kT)$ represents the changes needed in the process input, so that $y_e(kT)$ be zero or smaller (we consider $y_e(kT) < \varepsilon$, where ε is a predefined small value).

The knowledge base modifier consists in a technique of modifying fuzzy controller's rules base in order to achieve a better performance. The necessary changes in the input ($p(kT)$) modify the control action (4):

$$u(kT-T) \text{ will be replaced by } u(kT-T)+p(kT). \tag{4}$$

A more detailed description of FMRLC can be found in [4].

In this paper, we propose a knowledge base modification procedure based on reinforcement learning techniques.

A reinforcement learning technique: Q-learning

The origins of the reinforcement learning techniques are to be traced in the early 20th century. In 1910, Edward Thorndike developed a learning theory, which represents the base of the Stimulus -Answer Theory. He stated three primary laws of: effect, readiness and exercise.

Another theory that has influenced the reinforcement learning was the optimal control theory. The objective of the optimal control theory is to find a control which minimizes some measures of the system's behavior. The most known methods to solve reinforcement learning problems are Q-learning (Watkins, 1989 [9]) and SARSA-learning [7]. The mathematical model of the reinforcement learning method is the Markov decisional process model.

The Q-learning algorithm (Watkins, 1989) is detailed in [8] and has the formula 5.

$$Q(s,a) := Q(s,a) + \alpha(R(s,a) + \gamma \max_{a'} Q(s',a') - Q(s,a)), \tag{5}$$

where $\alpha \in (0,1)$ is the learning rate, $\gamma \in (0,1)$ is the discount factor and s' is the achieved state starting from state s and applying the action a.

Q-learning begins with an initial estimation $Q(s,a)$ for each state-action pair. The system receives the reward $R(s,a)$, when action a is chosen in the state s and the next state is observed, i.e. s'.

The selecting modes for the parameters learning rate and discount factors are presented in paper [5]: a null value for learning rate means that the system will never learn. A high value of the learning rate means that the system learns quickly. The immediate reward is more important than the past reward. The discount factor has the value in the range $(0,1)$. A value close to 1 means that the future reward is more important than the immediate reward.

The pseudo-code of the Q-learning algorithm is:

initialization $Q(s,a)$ *arbitrarily*
repeat (for each scenario)
    initialization s
    repeat (for each step of scenario)
        choose a *using the derived policy from* Q *procedure Q-learning*
        observes s*, executes* a*, observes reward* R*, observes* s'
        $Q(s,a) \leftarrow Q(s,a) + \alpha(R(s,a) + \gamma \max_{a'} Q(s',a') - Q(s,a))$
        $s \leftarrow s'$

The Q-learning mechanism of FMRLC and experimental results

Our contribution in this paper consists in applying Q-learning algorithm in the learning mechanism of the FMRLC. The Q-learning mechanism will establish the rules base for fuzzy inverse model, which will become the rules base for fuzzy controller. The algorithm associates a value (Q variable) for each possible rules base, and finally it selects the rules base with the highest value Q. The Reward is defined as the difference between the obtained performance and desired performance.

The pseudo-code of the algorithm is:

Procedure QLF

e=Defuzzy(ye)

c=Defuzzy(yc)

choose arbitrarily a value Q for each possible Rules Base (Q=0)

repeat

execute the simulation for each rules base (MATLAB module)

compute reward R

$Q(BRi) \leftarrow Q(BRi)+\alpha\ (R(e, c, p)+\gamma\ maxiQ(BRi)- Q(BRi))$

Memorize Q in KBM

where $|-R| < \varepsilon$ *(*ε *is a predefined small value)*

choose the rules base that has maximum Q from KBM

update rules base to FC

In our study case, we consider the discourse universe normalized to the range $[-1,1]$.

We compared two rules bases (RB1 and RB2) to learn the fuzzy inverse module to obtain the performance presented in table no. 1.

Table 1. Desired Performance

y_{error}	$y_{change\ of\ error}$	desired performance
-1.00	-1.00	-1.00
-0.50	-1.00	-0.80
0.00	-1.00	-0.50
0.50	-1.00	-0.20
1.00	-1.00	0.00
-1.00	-0.50	-0.80
-0.50	-0.50	-0.61
0.00	-0.50	-0.31
0.50	-0.50	0.00
1.00	-0.50	0.20
-1.00	0.00	-0.50
-0.50	0.00	-0.31

y_{error}	$y_{change\ of\ error}$	desired performance
0.00	0.00	0.00
0.50	0.00	0.31
1.00	0.00	0.50
-1.00	0.50	-0.20
-0.50	0.50	0.00
0.00	0.50	0.31
0.50	0.50	0.61
1.00	0.50	0.80
-1.00	1.00	0.00
-0.50	1.00	0.20
0.00	1.00	0.50
0.50	1.00	0.80
1.00	1.00	1.00

The rules bases are presented in table no. 2.

Table 2. Analyzed Rules Bases

RB1			RB2		
y_{error}	$y_{change\ of\ error}$	desired performance	y_{error}	$y_{change\ of\ error}$	desired performance
neg	Pos	Ze	neg	Pos	ze
neg	Ze	Ze	neg	Ze	neg
neg	Neg	Ze	neg	Neg	neg
ze	Pos	Ze	ze	Pos	ze
ze	Ze	Ze	ze	Ze	ze
ze	Neg	Ze	ze	Neg	ze
pos	Pos	Ze	pos	Pos	ze
pos	Ze	Ze	pos	Ze	ze
pos	Neg	Ze	pos	Neg	ze

We studied two cases ($\alpha = 0.99$ and $\gamma = 0.5$; $\alpha = 0.5$ and $\gamma = 0.25$). The results obtained for Q values are presented in table no. 3.

Table 3. Results for Q values

Q	case 1	case 2	Q	case 1	case 2
RB1	0.000000	0	RB1	0.284113	0.225758
RB2	0.000000	0	RB2	0.282297	0.085102
RB1	0.990000	0.5	RB1	-0.163424	-0.0139
RB2	0.257400	0.13	RB2	-0.163443	-0.08423
RB1	1.291950	0.7125	RB1	-0.577527	-0.25869
RB2	1.110384	0.4395	RB2	-0.577527	-0.29385
RB1	1.147435	0.695313	RB1	-0.093653	-0.06168
RB2	1.145619	0.558813	RB2	-0.269873	-0.16826
RB1	0.777452	0.53457	RB1	-0.047293	-0.03855
RB2	0.777434	0.46632	RB2	-0.225275	-0.18084
RB1	0.392612	0.334106	RB1	-0.330781	-0.17909
RB2	0.392612	0.299981	RB2	-0.332561	-0.25024
RB1	0.990268	0.608817	RB1	-0.770944	-0.41693
RB2	0.323008	0.254754	RB2	-0.770962	-0.45251
RB1	1.103986	0.685511	RB1	-1.181325	-0.66058
RB2	0.921094	0.41948	RB2	-1.181325	-0.67837
RB1	0.864415	0.583444	RB1	-0.596567	-0.41286
RB2	0.862586	0.450429	RB2	-0.596567	-0.42176
RB1	0.436527	0.364652	RB1	-0.499268	-0.35804
RB2	0.436509	0.298144	RB2	-0.499268	-0.36249
RB1	0.022448	0.127907	RB1	-0.747131	-0.47377
RB2	0.022447	0.094653	RB2	-0.747131	-0.476
RB1	0.506337	0.329942	RB1	-1.169301	-0.69611
RB2	-0.226263	-0.05668	RB2	-1.169301	-0.69722
RB1	0.562602	0.361214	RB1	-1.580497	-0.93507
RB2	0.381036	0.079901	RB2	-1.580497	-0.93562

The two analyzed rules bases have an almost identical Q final value. At this point, the better rules base proves to be RB1.

Conclusions

The proposed algorithm dynamically updates the rules base of FMRLC. The researches presented in this paper highlight a problem of the FMRLC: defining more precise and faster algorithms for the learning mechanism of a FMRLC. A possible solution is to use learning techniques from artificial intelligence as reinforcement learning. In this paper, authors present some promising and actual results of implementing a Q-learning algorithm in the learning mechanism of FMRLC. Our future research direction is to develop a software that may implement the Q-learning into FMRLC, in order to establish ways to set the rules bases of FMRLC.

References

1. Aström, K.J., Wittenmark, B. – *Adaptive Control*. 2nd ed. Addison-Wesley, 1995.
2. Jantzen, J. – *A Tutorial on Adaptive Fuzzy Control*. Technical University of Denmark, 2002.
3. Jantzen, J. – *Design of Fuzzy Controllers*. Technical University of Denmark, Department of Automation, Bldg 326, DK-2800 Lyngby, DENMARK. Tech. report no 98-E 864 (design), 19 Aug 1998.
4. Layne J. R., Passino, K. M. – *Fuzzy Model Reference Learning Control*, http://www.ece.osu.edu/~passino/
5. Leon, F., Şova, I., Gâlea, D. – *Reinforcement Learning Strategies for Intelligent Agents in Knowledge-Based Information Systems*. In: Proceedings of the 8th International Symposium on Automatic Control and Computer Science, Iaşi, 2004.
6. Petrov, M. G., Topalov, A. V. – *On Adaptive Fuzzy Control Using Neural Network*. IFAC Workshop on New Trends in Design on Control System, Smolenice, Slovak Republik, 1994.
7. Rummery, A., Niranjan, M. – *On-line Q-learning Using Connectionist Systems*. Technical Report CUED/F-INFENG/TR 166, Engineering Department, Cambridge University, 1994.
8. Sutton, R. S., Barto, A. G. – *Reinforcement Learning: An Introduction*. A Bradford Book, The MIT Press Cambridge, 1998.
9. Watkins, C. – *Learning from Delayed Rewards*, Thesis, University of Cambridge, England, 1989.

Utilizarea tehnicilor de învăţare prin recompensă pentru un controller fuzzy adaptiv

Rezumat

În această lucrare, autorii prezintă cercetările lor asupra mecanismelor de învăţare aplicate în controller-ele fuzzy adaptive. Metoda propusă constă în utilizarea tehnicilor de învăţare prin recompensă pentru actualizarea bazei de cunoştinţe a controller-ului în scopul menţinerii performanţei sistemului. Algoritmul este un algoritm general care poate fi aplicat într-o varietatea mare de probleme. Este prezentat un studiu de caz utilizând mediul software MATLAB®.