Slovak University of Technology in Bratislava
Institute of Information Engineering, Automation, and Mathematics

PROCEEDINGS

17th International Conference on Process Control 2009

Hotel Baník, Štrbské Pleso, Slovakia, June 9 – 12, 2009

ISBN 978-80-227-3081-5

http://www.kirp.chtf.stuba.sk/pc09

Editors: M. Fikar and M. Kvasnica

Macek, K.: Reinforcement Learning Parameterization: Softmax Between Exploration and Exploation, Editors: Fikar, M., Kvasnica, M., In *Proceedings of the 17th International Conference on Process Control '09*, Štrbské Pleso, Slovakia, 438–442, 2009.

Full paper online: http://www.kirp.chtf.stuba.sk/pc09/data/abstracts/068.html

REINFORCEMENT LEARNING PARAMETERIZATION: SOFTMAX BETWEEN EXPLORATION AND EXPLOATION

K. MACEK*

** FNPE CTU, Department of Mathematics, Trojanova 1, Praha 2, CZ 120 00*

e-mail: macek@fjfi.cvut.cz

Abstract. Control in dynamic systems stands for a complex task with respect to changing conditions, nonlinear dependencies and time delays. One of tools of online optimization of control parameters is reinforcement learning. Present paper deals with its application in PID parameters optimization and examines the most appropriate parameterization of softmax selection mechanism.

Keywords: PID control, reinforcement learning, softmax, fluid dynamics.

1 INTRODUCTION

Large dynamic system control stands for a complex task. The behavior of the system is complicated by influences among its parts, positive and negative feedbacks, and time delays. Due the dynamic nature of control conditions (including failures and degradation of some functionalities), adaptive systems can be operate better than the once set up ones. Nevertheless, there are two different tendencies: on one hand, the system is expected to be controlled smoothly, on the other hand, operating modes are switched crisply and the control is done also at some more abstract and symbolic level.

Current research deals with online learning very intensively with respect to various application areas (Silva, Datta, & Bhattachaiyya, 2005). The present paper compares some of reinforcement learning parameterization with respect to a very simple mechanical system controlled by a PID controller. This topic was examined also in other works (Anderson, 1997), (Hafner, 2007). However, this paper however brings new results mainly in the parameterization of PID controller via a discrete set of values via reinforcement learning. $\dot{V}_{in}$

The motivation for reinforcement learning in the PID control obvious: some parameters are more suitable for one situation, while other situation is controlled by other parameters better. Proposed approach makes the control more robust and it is able to use this approach for cases where the parameters have to be determined online.

2 MODEL DESCRIPTION AND PROBLEM FORMULATION

A very simple system was addressed just for demonstrative purposes. The model consists of a bin with an inlet at bottom and a tap (Durst, 2008). The aim is to control the water flow so the water level reaches given setpoint. This setpoint may vary in time. Dynamics of the system can be described as follow:

$$h(t) = V(t)/S_B \tag{1}$$

$$v(t) = \sqrt{2h(t)g} \tag{2}$$

$$\dot{V}_{total} = \dot{V}_{in} - \dot{V}_{out} = \dot{V}_{in} - v \cdot S_E \tag{3}$$

In Equation (1), *h(t)* is fluid level height at time instant *t*, *V* is actual fluid volume, and S_B is bottom area surface. Equation (2) calculates efflux velocity with respect to Bernoulli's principle where g is the standard gravity. Finally, equation (3) provides the evolution of fluid volume in time determined by influx flow $\dot{V}_{in}$ and efflux flow given by the efflux velocity *v* and the exhaust area surface.

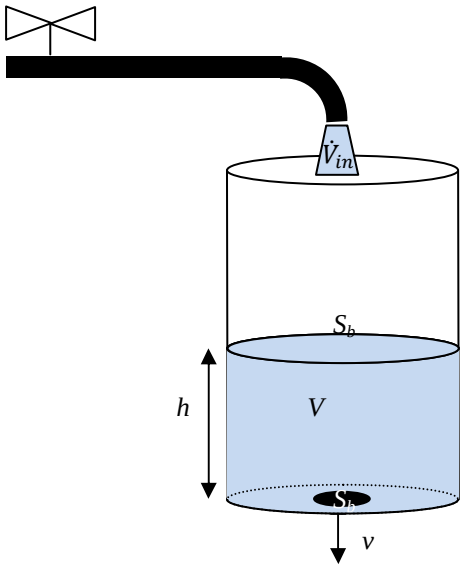

Figure 1: Simple control system. The aim is to reach setpoint h*by control of $\dot{V}_{in}$.

The tap is controlled by a PID controller with three parameters K_P, K_I, and K_D with respect to control error $e = h^* - h(t)$, namely

$$\dot{V}_{in} = K_P(e) + K_I \int_0^t e + K_D \frac{de}{dt} \quad (4)$$

For our purposes, we consider a set of combination of such parameters (triples). These combinations will be called actions be denoted a_1, a_2,…, a_{Na}, where N_a is count of actions. Parameters K_P, K_I, and K_D were sampled uniformly from [0 1000] for K_P and [0 1] for K_I, and K_D.

For the simulations, following parameters were fixed: S_B= 10 m^2, S_E = 1 m^2.

3 REINFORCEMENT LEARNING

Finding optimal PID controller setting has attracted many researchers for years and current methods provide sufficient results. The aim of the paper is not to offer new and better way to set up the PID parameters, but to analyze a particular aspect of reinforcement learning.

Reinforcement learning (Alpaydin, 2004) is a process when an agent takes actions in time, obtains reward from the system and with respect to it adapts and attempts to adopt such action selection mechanism that maximizes its objectives. Let the set of actions the agent can take is finite. Many reinforcement learning problems and methods work with this assumption, e.g. the well known *k*-armed bandit problem. In our case, the agent will decide which of N_a PID parameters triples will be chosen.

The dynamic optimization is performed in discrete time instants. Each T seconds [s], next action a is selected. The reinforcement learning requires some reward, i.e. penalty or payoff. In our case, the negative[1] aggregated control error for T time instants is calculated:

$$r_a(t) = -\int_{t-T}^{t} e(\tau)d\tau \approx -\sum_{\tau=t-T}^{t} e(\tau) \quad (5)$$

This aggregated control error expresses how good the action in given time instant t was. For each action, the quality Q is considered for all decisions the action was selected for. There are also other options how to calculate the quality; nevertheless, we will calculate it as follows:

$$Q_a(t) = \frac{\sum_{a(\tau)=a} r_a(\tau)}{|a(\tau)=a|+\epsilon} \quad (6)$$

where ϵ>0 is avoids division by zero and $|a(\tau) = a|$ is number of decisions when action a was selected. Dynamic decision making grapples with two usually conflicting requests: first, to decide for the best known action, i.e.

$$a(t) = \text{argmax}_{a=\{a_1,\ldots,a_{Na}\}} Q_a(t) \quad (7)$$

On the other hand, it is required to be sure with such decision. The information about the system may be maximized if the action is selected randomly. These two aspects call exploitation and exploration. This problem is discussed with respect to decision making in very various dynamic systems, including managerial sciences (Azoulay-Schwartz, Kraus, & Wilkenfeld, 2004), (Mom, Tom, Bosch, Frans, Volberda, & Henk, 2007), (Mom, Tom, Bosch, Frans, Volberda, & Henk, 2007).

4 PARAMETERIZED SOFTMAX ACTION SELECTION

One of the compromising methods is the softmax selection. First, the exponential is applied on actions' qualities. Afterwards, the action is chosen randomly with these values, i.e. the action a_i is selected with this probability:

$$p_i = \frac{\exp(Q_{a_i}(t))}{\sum_{j=1}^{Na} \exp(Q_{a_j}(t))} \quad (8)$$

The softmax action selection can be also parameterized, hence:

$$p_i = \frac{\exp(\alpha Q_{a_i}(t))}{\sum_{j=1}^{Na} \exp(\alpha Q_{a_j}(t))} \quad (9)$$

If the parameter α = 0, all actions have the same probability to be selected. If the parameter α is a big positive number, the best known action is selected and vice versa. The problem is how to set the parameter α so the system works optimally. Sometimes, in literature, parameter temperature $T = 1/\alpha$ is considered.

The algorithm can be summarized in following steps:

1. Initialization
2. Calculate qualities Q of actions
3. Calculate probabilities p with respect to qualities Q
4. Select action a with respect to the probabilities p
5. Measure the aggregated error r
6. Go to 2)

5 THE SIMULATION EXPERIMENTS

The model was implemented in MATLAB as a simple function with following inputs: actions (i.e. K_P, K_I, K_D triples), temperature, and setpoint evolution in

[1] The negation is applied so the reward is to be maximized as usual in reinforcement learning.

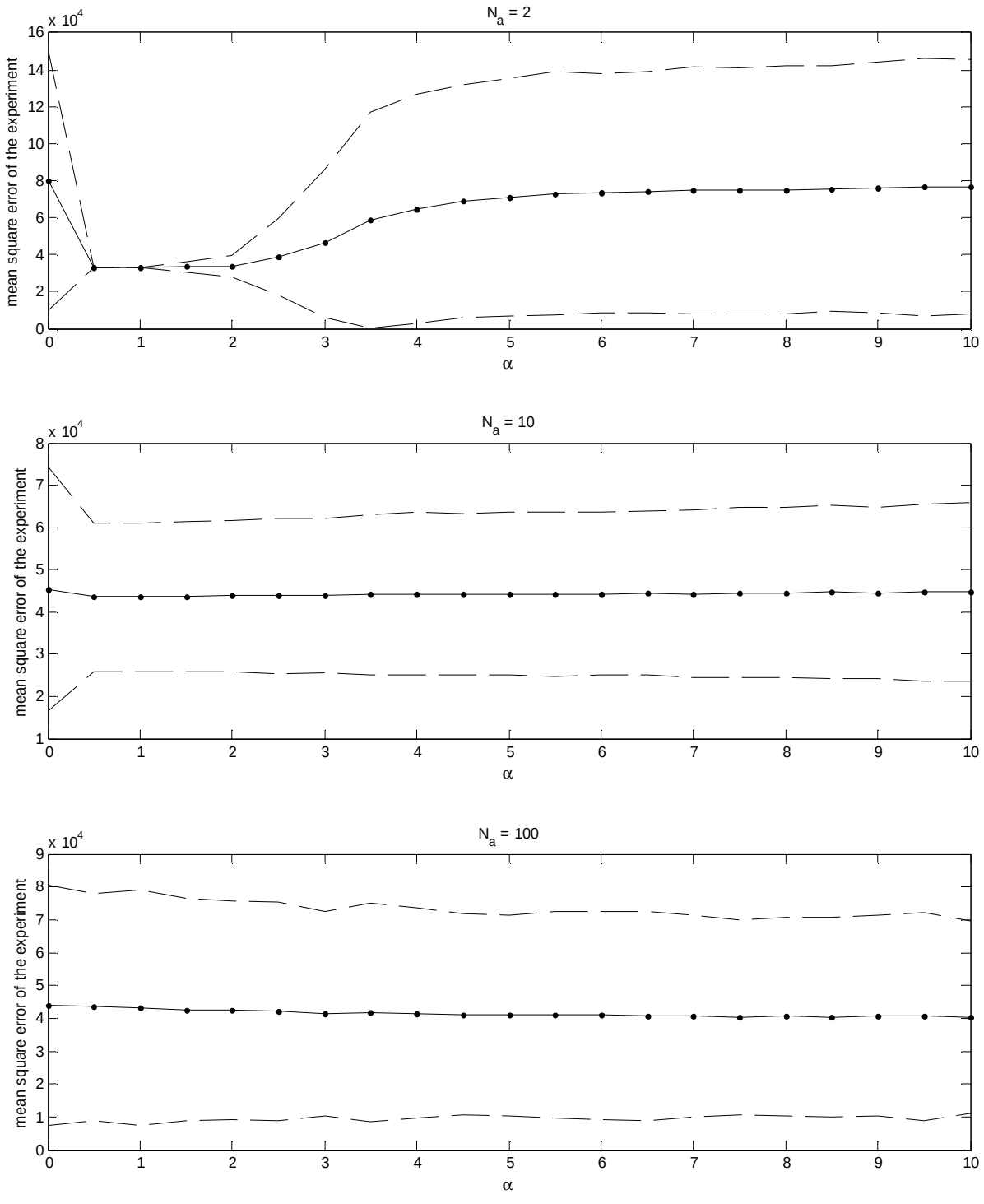

Figure 2: Results of experiments.

time. The experiment is repeated in 100 iterations so the results are statistically assessable. The function had two outputs, namely average of mean square control errors for all iteration and corresponding standard deviation.

The experiment was performed for different actions sets: N_a = 2, 10, and 100 and the coefficients were sampled randomly, K_P from [0 1000], K_I and K_D from [0 1]. For each of these sets, different values of α are considered, namely α = 0, 0.5, 1 …, 9.5, 10. For all experiments, only one setpoint evolution was considered. The time horizon is 1000 seconds. Within this period the setpoint changes 10 times and is sampled from [0 100] uniformly.

Figure 2 summarizes the results for N_a = 2, 10, and 100. Average experiment error (i.e. average of mean square of control errors during the experiment for all experiments) and 3σ tolerance interval are shown. On the average experiment error, following can be said with respect to statistical testing (each experiment was compared with other ones by *t*-tests).

- If there are only two PID parameter triples (Na = 2), the optimal value of α parameter is low, the figure shows that α = 0.5 is optimal. It is also interesting the variance is in this case extremely low. For all α>0, the average experiment error grows significantly and the variance as well.
- In case Na = 10, there is growing also tendency, but for values α = 7.5 becomes almost constant.
- In case Na = 100, a decreasing trend is determined that becomes for α = 6 constant.

Furthermore the influence of the α parameter is smooth.

Next, the standard deviation can be assessed as well. In case N_a = *2*, the deviation is growing. In other cases remains almost constant.

So, what are conclusions of performed experiments? When the system works best? Best and most reliable (with respect to variance) results are for N_a = *2* and α = *0.5* or *1*. However, till α = *2.5*, the results for N_a=*2* are better that for other *Na*. For N_a=*2*, α=*3* and N_a=*100*, α=*10* are the results similar.

What conclusions can be inferred from these facts? It is not surprising that smaller search space (N_a=*2*) operates better for very small values of parameter α, i.e. if the exploitation is high and vice versa. Hence, low $\alpha > 0$ stands for exploitation since values of *Q* are negative.

6 CONCLUSION AND FURTHER WORK

Present paper attempted to assess influence of softmax parameterization. It has been shown that optimal parameter depends also on the number of action and formulated problem: if more actions are considered, more exploration is needed. The testing of optimal setting of PID controller parameters will be estimated also in the future with more advanced tools like neural networks. Moreover, the softmax selection procedure for negative *Q* shall be examined with more alternatives where the *Q* is transformed to be positive before the softmax is applied. Proposed approach provides a robust control method that shall be compared with other PID tunings in the future and improved.

7 REFERENCES

Alpaydin, E. (2004). Introduction to Machine Learning (Adaptive Computation and Machine Learning). {The MIT Press}.

Anderson, C. W. (1997). Reinforcement learning, neural networks and PI control applied to a heating coil. Artificial Intelligence in Engineering , 11, 421-429.

Anderson, C. W., Hittle, D. C., Katz, A. D., & Kretchmar, R. M. (1997). Synthesis of reinforcement learning, neural networks and PI control applied to a simulated heating coil. AI in Engineering , 11, 421-429.

Azoulay-Schwartz, R., Kraus, S., & Wilkenfeld, J. (2004). Exploitation vs. exploration: choosing a supplier in an environment of incomplete information. Decis. Support Syst. , 38, 1-18.

Dayan, P., & Hinton, G. E. (1993). Feudal reinforcement learning. (pp. 271-278). Morgan Kaufmann.

Dimitrakakis, C. (2006). Nearly optimal exploration-exploitation decision thresholds. IDIAP-RR, IDIAP.

Durst, F. (2008). Fluid Mechanics:An Introduction to the Theory of Fluid Flows. Springer-Verlag Berlin Heidelberg.

Ellis, R., & Humphreys, G. (1999). Connectionist Psychology. A Text with Readings. Psychology Press.

Guestrin, C., Lagoudakis, M., & Parr, R. (2002). Coordinated Reinforcement Learning., (pp. 227-234).

Hafner, R. (2007). Neural Reinforcement Learning Controllers for a Real Robot Application.

Holmqvist, M. (2004). Experiential Learning Processes of Exploitation and Exploration Within and Between Organizations: An Empirical Study of Product Development. Organization Science , 15, 70-81.

Mom, Tom, J. M., Bosch, V. D., Frans, A. J., Volberda, & Henk, W. (2007). Investigating Managers' Exploration and Exploitation Activities: The Influence of Top-Down, Bottom-Up, and Horizontal Knowledge Inflows. Journal of Management Studies , 44, 910-931.

Moriarty, D. E., Schultz, A. C., & Grefenstette, J. J. (1999). Evolutionary Algorithms for Reinforcement Learning. Journal of Artificial Intelligence Research , 11, 241-276.

Poupart, P., Vlassis, N., Hoey, J., & Regan, K. (2006). An analytic solution to discrete bayesian reinforcement learning., (pp. 697-704).

Ramachandran, D. (2007). Bayesian inverse reinforcement learning.

Rivest, F., Bengio, Y., & Kalaska, J. (2004). Brain Inspired Reinforcement Learning.

Silva, G. J., Datta, A., & Bhattachaiyya, S. P. (2005). PID Controllers for Time-Delay Systems. Birkhäuser Boston.

Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2005). Robot Modeling and Control. John Wiley \& sons.

Torrey, L., Shavlik, J., Natarajan, S., Kuppili, P., & Walker, T. (2008). Transfer in Reinforcement Learning via Markov Logic Networks.

Zhang, X., Aberdeen, D., & N., S. V. (2007). Conditional random fields for multi-agent reinforcement learning. (pp. 1143-1150). ACM.