Reinforcement Learning Based Neural Controllers for Dynamic Processes without Exploration

Frank-Florian Steege[1,2], André Hartmann[2],
Erik Schaffernicht[1], and Horst-Michael Gross[1]

[1] Ilmenau Technical University, Neuroinformatics and Cognitive Robotics Lab,
P.O. Box 100565, D-98684 Ilmenau, Germany
[2] Powitec Intelligent Technologies GmbH, 45219 Essen-Kettwig, Germany
frank-florian.steege@stud.tu-ilmenau.de

Abstract. In this paper we present a Reinforcement Learning (RL) approach with the capability to train neural adaptive controllers for complex control problems without expensive online exploration. The basis of the neural controller is a Neural fitted Q-Iteration (NFQ). This network is trained with data from the example set enriched with artificial data. With this training scheme, unlike most other existing approaches, the controller is able to learn offline on observed training data of an already closed-loop controlled process with often sparse and uninformative training samples. The suggested neural controller is evaluated on a modified and advanced cartpole simulator and a combustion control of a real waste-incineration plant and can successfully demonstrate its superiority.

Keywords: Neural Control, Adaptive Control, Exploration-Exploitation.

1 Introduction

In the area of industrial process control most problems are still solved via conventional solutions from the field of control engineering. The problem with such conventional systems is that they are not able to adapt to changes of the systems to be controlled. In case of a change, the expert who designed the controller has to adapt the parameters of the controller again. For the described problem, it is desirable to use a self-learning controller which is able to adapt to changing dynamics.

Several approaches with learning controllers for unknown processes have been published in recent years. Examples are Reinforcement Learning (RL) Systems such as Q-Learning ([1]), Neural-fitted Q-Iteration (NFQ, [2], [3]) or Bayesian RL ([4], [5]) . Unfortunately, most of the RL-approaches rely on the assumption that it is possible to learn the optimal policy online and/or to explore different strategies for industrial control problems, such as the control of a waste incineration plant, this assumption is not realistic. An online learning phase of an agent

K. Diamantaras, W. Duch, L.S. Iliadis (Eds.): ICANN 2010, Part II, LNCS 6353, pp. 222–227, 2010.

with an inefficient or too explorative strategy at the beginning could commit serious damage to the plant. That means the learning process has to be done completly offline based on observed data. To complicate matters further, the observed data is taken from a closed-loop process where the acting controller is a conventional system which reacts with exactly the same action every time it observes the same state (see Fig. 1). This results in training data less informative than from real exploration periods and causes serious problems for the training of self-learning function approximators. To the best knowledge of the authors, no RL-approach has been published so far which is able to control the key elements of an industrial combustion process due to the charges and restrictions concerning exploration and training data mentioned above.

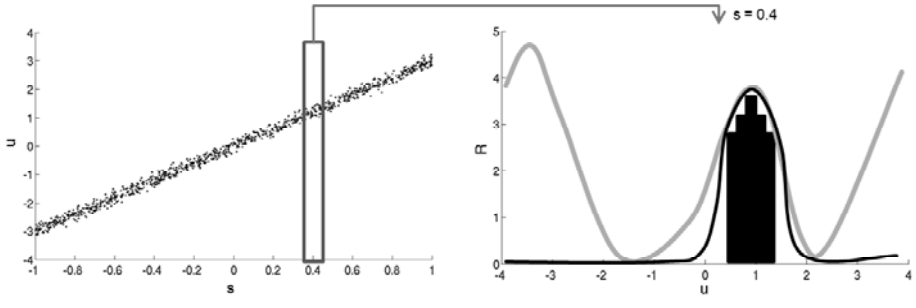

Fig. 1. *Left:* Training data set with the state s and control force u from a process controlled by a PI-Controller. *Right:* Action histogram at the state $s = 0.4$ (black boxes) and two possible reward estimations (black and lightgrey lines). Due to the unbalanced distribution of examples in the action space, neural networks can approximate very different reward functions for non-observed actions. Both reward functions would show the same approximation error but cause completly different agent policies.

In this paper, we present an RL-system which meets the demands of the described combustion control systems. The basis of our approach is a NFQ network as presented in [2]. We use the capability of the NFQ to add artificial data-points to the training set to ensure a correct learning of a good policy despite of the less informative nature of the observed data.

2 Problem Description and Experimental Setup

2.1 Application Domain

In a waste incineration plant the system dynamics are very complex and only partially known. Due to the changing process dynamics most existing control systems are set to cope with all appearing dynamics in general. This solution passes up chances to optimize the combustion for each single process dynamic. Therefore, an adaptive self-learning controller could significantly improve the combustion control.

We applied our controller to a plant with a forward-acting reciprocating grate (see Fig. 2). The stirring of the firebed by the movement of the grates is the main factor for the intensity of the combustion process and is the actuating variable of our controller.

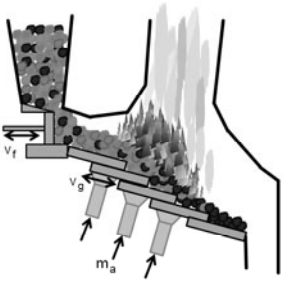

Fig. 2. Waste incineration plant. The feeder plunger brings the waste upon the combustion grate with the speed v_f. Air with the mass m_a is blown into the combustion zone. The grate elements are locomotive with the speed v_g.

2.2 Cartpole-Simulator

The cartpole problem is frequently used as a benchmark for RL solutions [3]. In order to take characteristic demands of a combustion process, like aforementioned changes of the system dynamics into consideration the simulator split the simulation into three sequent phases. In every phase, the parameters of the cart and so the dynamics of the system are slightly different. Only the mass of the cart ($5.0kg$) and the gravity ($9.81\frac{m}{s^2}$) are unchanged. The following table lists the parameters of the cart simulator for every phase:

Parameter	**Phase 1**	**Phase 2**	**Phase 3**
Mass Pole (m_p)	$2.0kg$	$2.0kg$	$2.5kg$
Pole Length (l_p)	$1.0m$	$1.0m$	$0.75m$
max. Random Force (F_r)	$1.0N$	$1.0N$	$1.0N$

Since the focus of this paper lies on training a neural controller which could observe only data from an already controlled process, our simulator also utilizes a conventional PID-controller for controlling the actions of the cart. The maximum Control Force was limited to $5.0N$ for the NFQ and the PID-Controller.

3 Algorithm

Our approach aims at training an adaptive controller for an already controlled closed loop process which is

1. not worse than the existing controller,
2. shows a behaviour similar to the old controller
3. but is able to adapt to changes of the process dynamics.

The problem with data observed from a process which is controlled by a conventional deterministic controller is shown in Fig. 1. To each state s the controller only chooses very few different actions from the possible pool of actions a. A function approximator, which approximates the reward for the whole action space can calculate very different values for the reward of actions not observed without an increase of the training error (see Fig. 1).The policy choosen by such a net would be very different to the policy of the old controller which is very critical for a serious industrial application.

So, the basic idea of our approach is to label those actions which have not yet been observed with a low reward without worsening the approximation of observed actions and the generalisation-capabilities of the network for not observed states.

NFQ as presented in [2] requires only very few parameters which can influence the training process and have to be optimised, and offers the possibility to insert artificial data into the training set in a very elegant and simple way. An artifical point is a tuple (s, a, f, Q) consisting the state s, action a and Q-Value Q. A flag f is used to signal the artifical state. We identify non-observed actions by histogramms build over clusters in the state space of the observed dataset (see Fig. 1). The insertion of the artificial data and the training of the NFQ is done as follows:

1. Input = exampleset X where $x_i = (s, a, s', Q)$, $x_i \in X$, $i = 1..n$
2. Cluster X into m cluster C depending on the state s, $C_i = (S_{C_i}, A_{C_i}, S'_{C_i}, Q_{C_i})$, $C_i \in C$, where S_{C_i}, A_{C_i}, S'_{C_i} and Q_{C_i} are sets of all s, a, s' and Q in the cluster C_i
3. For every C_i build a histogram with k bins of actions A_{C_i} appearing
4. If there are action bins with no examples, insert new examples x_{new} into the exampleset X where $x_{new} = (s_{new}, a_{new}, f, Q_{new})$
   $s_{new} = \frac{1}{l} \sum_{i=1}^{l} s_i, s_i \in S_C, l = |S_C|$
   a_{new} = action value of the empty histogram bin
   f = binary flag to label as *artificial data*
   $Q_{new} = min(Q_{C_i}) - Q_{offset}$; Q_{offset} = Reward-Offset to penalize non-observed actions
5. Train NFQ as described in [2] with the new exampleset X_{new}

By this algorithm, artificial examples with lower reward are created for all non-observed actions in observed states. This prevents that a function approximator estimates high rewards for such actions, and the policy of the trained agent does not choose them. The number of the artificial data points inserted at a certain state depends on the number of already observed data points in that state. This is important because the lower reward of the artificial points is used to decrease the value of unobserved actions, but should not change the value of a state compared to the value of other states. If a sufficient number of new samples is collected, the agent is retrained with the new data. So the adaptive nature of the controller to changes of the process dynamics is realised.

4 Experimental Evaluation of the Approach

The experiments on a real combustion process were executed on a waste incineration plant in Germany with a steam production of 30 t/h. The new NFQ-controller was trained with seven days of data observed from the conventional controllers. After training, the NFQ was tested on the plant. The test phase covered eight days which were split between the NFQ and the PID-controllers. Both controllers acted with a clock of five seconds which results in 17,280 actions per day. The achieved experimental results are as follows:

Controller	**ØCD Steam**	**Max(CD Steam)**	**ØCO**	**ØNOx**
PID-Controller	1.61%	22.2%	12.77	87.06
NFQ with art.data	1.45%	16.1%	11.21	86.95
NFQ without art.data	canceled (>10%)	canceled (>25%)	canceled	canceled

CD is the abbreviation for control deviation and is specified in % of the total amount of steam production. The emissions of carbon monoxide (CO) and nitrogen oxide (NOx) were measured in mg/Nm^3. The NFQ which was trained with additional data achieved a better control deviation and reduced noxious gases better than the classical PID-controller. A comparison with a NFQ-Controller without insertion of artificial data had to be canceled after a while because the policy of the NFQ without artificial data was not similar to the policy of the old controller and caused a continuous control deviation of more than 15%. This dangerous policy was the result of a wrong reward approximation as it was shown in Fig. 1.

The cartpole simulator was configured as described in section 2.2. At first the cartpole was controlled by a PID-Controller. 3,000 samples of this experiment were recorded as training samples for the improved NFQ training described in section 3. We inserted 4 virtual points for non-observed actions per real data point. After training, we created two instances of the cartpole simulator. Both instances received the same sequence of random forces affecting the pole. One of the instances was controlled by the PID the other one was controlled by the improved NFQ. Both instances were simulated for 15,000 steps, and each experiment was repeated 10 times with different random sequences:

Controller	**Balancing steps**	**ØControl error**	**SD Control error**
PID-Controller	15,000	0.0739	0.0015
NFQ without art.data	5	1.5707	0.0001
NFQ with art.data	15,000	0.0575	0.0018

The NFQ without artificial samples in the training set was not able to balance the pole for more than 5 steps. Contrary, the NFQ with artificial samples was able to balance the pole and had a lower control error than the conventional PID-controller.

As we explained in section 1, the main advantage of a self-learning conroller is its ability to adapt itself to the process. For the next experiment we collected

a data set of 3,000 samples from the runtime of the PID-controller and 3,000 samples of the runtime of the NFQ-controller described above. With this data we trained a new NFQ. The results are as follows:

Controller	Balancing steps	ØControl error	SD Control error
PID-Controller	15,000	0.0735	0.0014
NFQ with art.data	15,000	0.0469	0.0010

While the results of the PID-controller are the same as in the first experiment, the NFQ-controller was able to improve its result from the first test by 19%. It should be explicitely mentioned, that all results were achieved without explicit exploration phases, all controllers were run in exploitation mode the whole time.

5 Conclusion and Outlook

The paper presents a new approach to train neural controllers with data from closed loop processes without exploration. Through the insertion of virtual points with state-depending Q-Values, neural controllers were able to control processes. The same controllers failed if no virtual points were inserted. The new approach offers the possibility to replace conventional controllers through neural adaptive controllers without expensive exploration phases.

Our further research is supposed to concentrate on increasing the applicability of the controller. The possibility to insert artificial data to the training set could allow us to influence the controller in many ways. Expert knowledge about very rare special states and the right control-action could be integrated into the controllers. Such knowledge is extremly valuable because rare special states might not be observed in the example set and the desired policy for these states may differ from the normal policy observed in common states.

References

1. Watkins, C.: Learning from delayed rewards, PhD Thesis, University of Cambridge, England (1989)
2. Riedmiller, M.: Neural fitted Q-Iteration - First Experiences with a Data Efficient Neural Reinforcement Learning Method. In: Gama, J., Camacho, R., Brazdil, P.B., Jorge, A.M., Torgo, L. (eds.) ECML 2005. LNCS (LNAI), vol. 3720, pp. 317–328. Springer, Heidelberg (2005)
3. Riedmiller, M.: Neural Reinforcement Learning to Swing-Up and Balance a Real Pole. In: Proc. Int. Conf. SMC, vol. 4, pp. 3191–3196 (2005)
4. Price, B., Boutilier, C.: A Bayesian Approach to Imitation in Reinforcement Learning. In: Proc. IJCAI, pp. 712–720 (2003)
5. Schaffernicht, E., Stephan, V., Debes, K., Gross, H.-M.: Machine Learning Techniques for Selforganizing Combustion Control. In: Mertsching, B., Hund, M., Aziz, Z. (eds.) KI 2009. LNCS, vol. 5803, pp. 395–402. Springer, Heidelberg (2009)