Vol.1, No. 2, pp. 121-136, 2001

International Journal of Computational Intelligence and Applications

A NEW CONTROL SCHEME FOR COMBUSTION PROCESSES USING REINFORCEMENT LEARNING BASED ON NEURAL NETWORKS [*]

V. STEPHAN, K. DEBES, H.-M. GROSS
Department of Neuroinformatics, Ilmenau Technical University
98684 Ilmenau, Germany, P.O.B. 100565
volker.stephan@informatik.tu-ilmenau.de

F. WINTRICH, H. WINTRICH
ORFEUS Combustion Engineering GmbH
45128 Essen, Germany
info@orfeus.gelsen-net.de

Received (received date)
Revised (revised date)

We present a new control scheme for an industrial hard-coal combustion process in a power plant based on reinforcement-learning in combination with neural networks. To comply with the great requirements for environmental protection, the plant operator is interested in a minimization of the nitrogen oxides emission and a maximization of the efficiency factor, while other process parameters have to be kept within predefined limits. To cope with both the tremendous action and state space of the power plant, we present a multiagent-reinforcement-system consisting of 4 agents, which are realized by relatively simple neural function approximators. We demonstrate, that our multiagent-system was able to significantly reduce the overall air consumption of the real combustion process of the power plant.

Keywords: Reinforcement-Learning, neural networks, combustion process

1. Introduction

Since the immediate objective of a power plant is the production of energy, the plant operator is trying to maximize the efficiency factor. Simultaneously, both the system-constraints and great requirements for environmental protection limit the workspace. Because of time varying plant properties caused by pollution, fair wear and tear, changing coal qualities, etc., a control system is sought, which autonomously tries to minimize a predefined cost function.

Reinforcement learning (RL) can be used to solve such problems. The main idea of RL consists in using experiences obtained through interaction with the en-

[*] This work is supported by the German BMBF (grant number 032 6843 B) and the RWE AG

vironment (in this case the combustion process) to progressively learn an optimal value function. This value function predicts the best long-term outcome an agent can receive from a given state when it applies a specific action and follows the optimal policy thereafter.[1] The agent can use a RL-algorithm such as SUTTON's $TD(\lambda)$ algorithm,[1] or WATKINS' Q-learning algorithm [2] to improve the long-term estimate of the value function associated with the current state and the selected action. However, in systems with continuous state and action spaces, the value function must operate with real-valued variables representing states and actions. Therefore, the value functions are typically represented by function approximators, which use finite resources to represent the value of continuous state-action pairs. *Neural Function approximators* are useful because they can generalize the expected return of state-action pairs the agent actually experiences to other regions of the state-action-space. Thus, the agent can estimate the expected return of state-action pairs that it has never experienced before. Many classes of function approximators have been presented, each with advantages and disadvantages. The choice of a function approximator depends mainly on how accurate it is in generalizing the values for unexplored state-action pairs, whether it is able to realize online learning and how expensive it is to store in memory.

2. State of the Art

Large-scale combustion power plants are monitored by and operated via process control systems, which solve problems of visualization, alarm indications and the application of low-level control-components. It is commonly known, that the performance of such complex processes can be significantly improved through a higher control level realized by manual control actions of an experienced operator. Although automated solutions for this high level control are very complicated, there exist a lot of intelligent control solutions.[3,4] Most of the existing intelligent control schemes intensively use models of the process to be controlled[5,6,7,8] or build knowledge bases derived from experienced experts[9,10,11]. The decision to use these so called expert systems for intelligent process control mainly depends on the availability and quality of the knowledge base. Likewise, the application of model based control strategies crucially depends on the quality of the process model.

Due to the absence of mathematical models and ingenious knowledge bases in our special case of controlling a combustion process, these approaches for intelligent process control did not seem practicable. For this reason, the authors investigated a control scheme, that requires no a priori knowledge and no mathematical model of the combustion process. The proposed Reinforcement-learning (RL)[12] based control scheme allows an autonomous exploration of the state-action space of the combustion process, while predefined quality factors established by the plant operator have to be optimized.

In summary, the approaches based on mathematical models, ingenious knowledge bases or reinforcement learning probably would have a similar performance to solve the task given. The differences between these approaches can be characterized as follows. The main disadvantage of the model or knowledge based systems is the compliance with the requirements of the very expensive a priori knowledge. Another problem is the resulting limited portability to other plants, caused by slightly different process parameters. Furthermore, also time variant processes would crucially require an expensive redesign of the process model or knowledge base from time to time. In contrast, the reinforcement-approach does not require any a priori knowledge about the process, because it explores the process-features and properties through interaction with the process itself. Hence, RL is much more flexible, portable and can also adapt to changing plant properties. This main advantage of RL implies on the other hand a minor drawback, because during the necessary exploration phase the system has to perform many different control actions, of course also suboptimal ones!

RL has been successfully applied to solve real world problems in many cases. RL is applied to refine the performance of conventional controllers[13] or to tune fuzzy-controllers.[14,15] Further real-world applications of RL concern dynamic channel allocation in telephone systems,[16] optimization of an elevator system[17] or packet routing.[18] To the best of our knowledge, this paper presents for the first time a RL based approach to control the combustion process of an industrial power plant.

3. The Plant

The electrical power plant we used for our experiments is owned by the "Hamburgische Elektrizitaetswerke" (HEW) and is situated in the south of Hamburg (Germany) and has a maximal output of 252 MW.

Figure 1 gives an overview of the plant. The coal mills produce coal dust, which together with fresh air is fed into the combustion chamber via several valves. The emerging warmth is used to heat steam, which propels a steam turbine to produce electricity. The emerging exhaust gas is used to preheat the incoming cold fresh air and is cleaned in subsequent parts of the plant.

The subsystem we have to control consists of 6 burners aligned in 2 columns at 3 levels (10, 20, 30) in one combustion chamber (Fig. 1).

As introduced earlier, the main objective of the plant operator are the increase of the efficiency factor and a reduction of NO_x-emissions. Both variables crucially depend on several properties of the underlying combustion process. Unfortunately, there is only unspecific and global information available, like total amounts of inlet air and coal, NO_x or O_2 emissions. But, more detailed information for instance about the very important ratio of air and coal for each burner was not available.

This fact becomes more important, since the exact amount of inlet coal can not be measured for each burner separately. That is because the burners at each level are supplied with coal by one coal mill. Although the distribution of inlet coal should be equal for each of the two supplied burners, due to varying flow dynamics or pollution, this equilibrium may be shifted to benefit one burner.

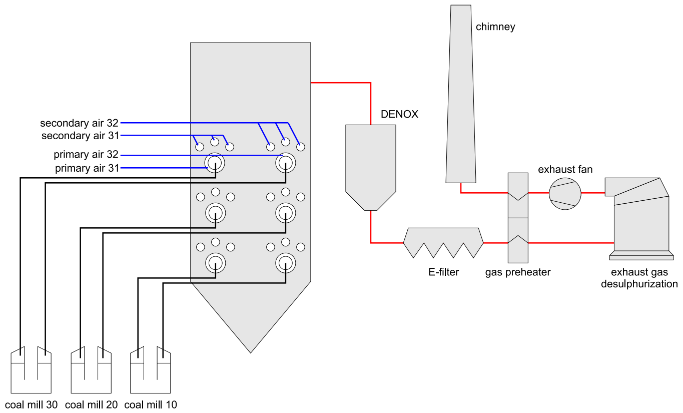

Fig. 1. Schematic view of the power plant. The combustion chamber features three burner-levels with 2 burners each and is supplied with coal by three coal mills.

To get more burner specific information about the distribution of coal and air inside the combustion chamber or about the flames, we observe each of the 6 flames by a special color camera system, provided by the ORFEUS Combustion Engineering GmbH, and use these data to control the process. Figure 2 depicts the extraction of visual features describing the combustion process. First, we filter the incoming videostream from each camera in the time domain and afterwards we calculate fitvalues describing intensity, form and position of the flames. In order to reduce the large set of flame-describing features, we analyzed the correlation of the visual features and several important process data, for instance the NO$_x$ and O$_2$ emissions and the waste gas temperature. We found, that already the mean intensities in the R-Band of the RGB-images of the flames ($F_{00}^{RMB11} \dots F_{00}^{RMB32}$) entail very detailed information about the distribution of coal and temperature inside the combustion chamber.

The plant operator defined the goal of the controller as follows. First, we have to satisfy several limitations at all times to guarantee the safety precautions:

- steam temperature $> 540°$ C
- waste gas temperature $> 340°$ C
- NO$_x$ concentration in waste gas $< 1200 mg/m^3$
- unused carbon $< 5\%$
- O$_2$ $> 3\%$

In addition, the control system has to minimize both the NO$_x$ emissions and the air consumption in order to increase the efficiency factor.

To fulfil the defined goals the plant operator gave us direct access to the controls depicted in table 1 (see also Fig. 3). Please bear in mind, that these 12 controls

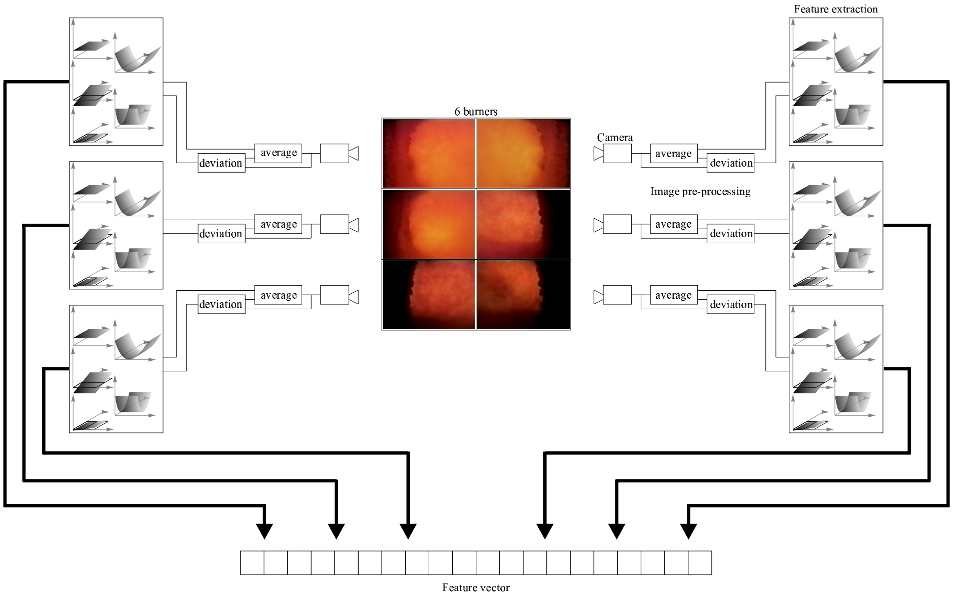

Fig. 2. Extraction of visual features describing the combustion process very closely.

Table 1. Control variables.

control variable	meaning
primary air trim at levels 10, 20, 30	air-distribution between the left and right burner on the specified level
primary/secondary air trim for all burners	distribution between primary and secondary air at the specified burner
air amount at levels 10, 20, 30	overall air amount on the specified level

(see Fig. 3) only influence the air amount and the distribution of air between these 6 burners, but neither amount nor distribution of inlet hard-coal! To reduce this immense action space, we use relative instead of absolute controls. That means, we define only three actions for each control: increase by 1%, remain unchanged or decrease by 1% (the use of absolute controls with only 10 quantization steps for each control would lead to an overall action space of 10^{12} different actions!!) Despite the usage of relative controls, our system would have to cope with an enormous action space of $3^{12} = 531441$ actions.

4. Architecture

As motivated in sections 1 and 2, we investigated the application of an RL based control scheme to control the combustion process. The main feature of RL is its self exploration of the outcomes of control actions with respect to a predefined goal. Based on a sensory description of the current process situation, the RL-system selects an appropriate control action, performs it, observes its consequences

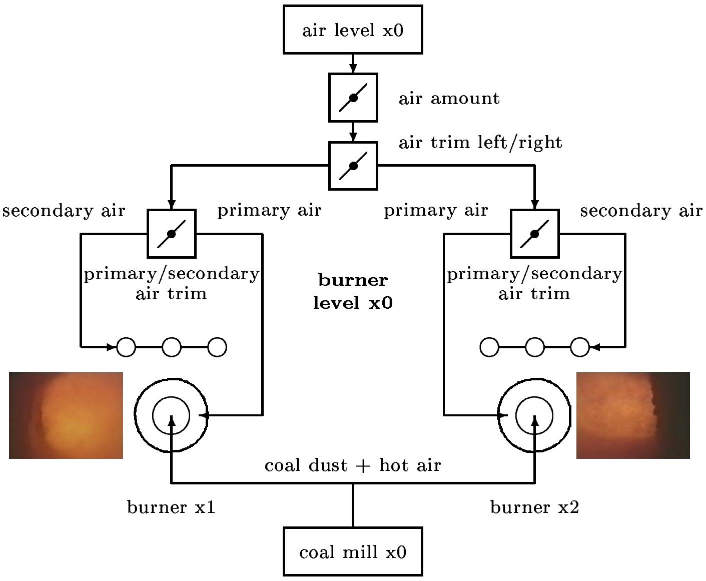

Fig. 3. Schematic view of the combustion chamber with coal and air supply for one of three burner-levels.

and acquires a reward (see Fig. 4). The task of the architecture is to obtain a utility-value for all experienced state-action-pairs, which is defined by the reward and the discounted value of the new state. We decided to use the so called classical Q-learning because of very promising experiences on robot-navigation tasks.[19,20] The applied neural function approximator, which approximates the utilities of all actions for a given process situation, utilizes a Neural-Gas-Clusterer,[21] providing a topologically organized and continuous process situation code in combination with a subsequent output layer providing the utility-values (Q-values) of the state-action-pairs (see Fig. 7 right).

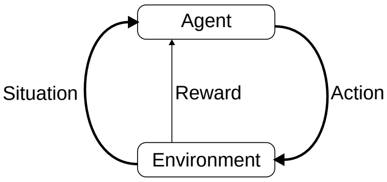

Fig. 4. Closed interaction cycle between RL-system and the environment (combustion process).

During the exploration phase the RL-system has to perform all control actions in all process situations. However, due to the tremendous action space (12 independent control variables) in combination with the very large situation space of the process a full exploration of all state-action-pairs would last a very long time and is therefore not applicable to solve our control task.

4.1. *Problem decomposition*

Consequently, we designed several agents, each observing only a relevant subset of the situation space and using only a subset of the available controls. Figure 5 (left) depicts the decomposition into 4 agents with their inputs and their corresponding controls.

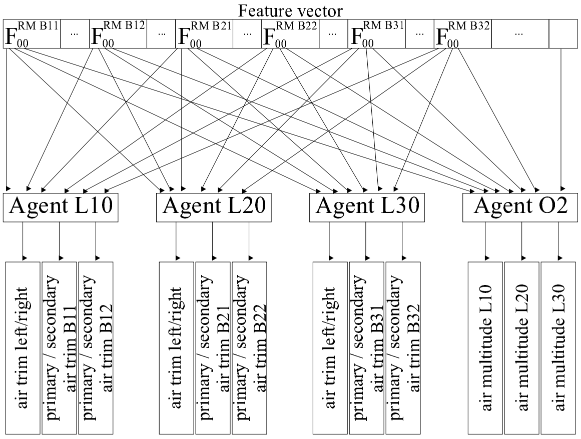

Fig. 5. Decomposition of the control task into 4 agents with their inputs and their corresponding controls. Each agent observes the relevant part of the situation space and has access to an assigned subset of controls.

Thus, according to table 2, AGENTL10, AGENTL20, and AGENTL30, observe only the intensity-ratios of the left and right flame on all levels and control the air distribution at their corresponding burner level (3 control variables each). AGENTO2 observes the intensities of all flames and the global ratio of inlet air and coal (λ). This Agent controls the total amount of air consumption for each burner level (3 control variables). The introduction of a scheduling of the 4 agents at this point is very important, since the reinforcement-approach assumes, that each agent is able to observe directly the consequences of its own actions. If two agents would perform their actions together, the consequences of their actions (e.g. NO_x concentration) would interfere and the resulting cross-talk between the agents would prevent a correct acquisition of the real outcomes of the respective actions. Hence, in this first approach, we defined, that all 4 agents operate sequentially in time intervals of 10 minutes (see Fig. 6).

Table 2. Inputs, states (NGU) and outputs of all agents.

Agent	input	NGU	output
AgentL10	3	20	$3 \cdot 3 \cdot 3 = 27$
AgentL20	3	20	$3 \cdot 3 \cdot 3 = 27$
AgentL30	3	20	$3 \cdot 3 \cdot 3 = 27$
AgentO2	7	50	$3 \cdot 3 \cdot 3 = 27$

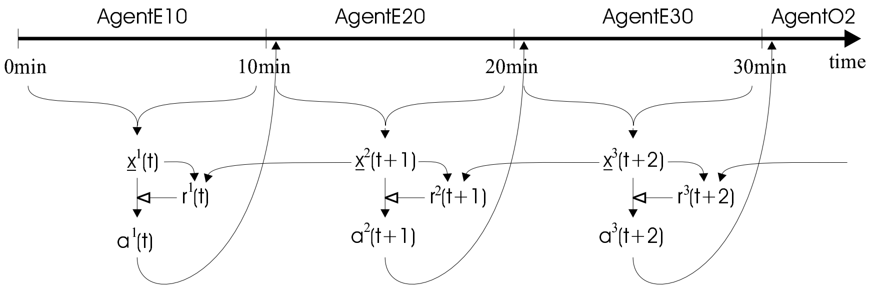

Fig. 6. Applied time scheduling of our 4 agents. First, AgentE10 observes the process for about 10 minutes, generates its action $a^1(t)$ and performs it at time $10min$. Now, AgentE20 observes the process for about 10 minutes, generates its action $a^2(t)$ and performs it at time $20min$ and so on. The difference between the observations $\underline{x}^1(t)$ and $\underline{x}^2(t+1)$ yields to the reinforcement $r^1(t)$, which is used to adapt the policy of AgentE10.

4.2. *Neural function approximator*

As described earlier, each of these 4 agents is realized by a neural function approximator. In this paper, we present a very first and simple approach to this state-action function approximator that combines a neural vector quantization technique (Neural Gas[21]) for optimal clustering of the high-dimensional, continuous input space[20] (Eq. 4.1) with a subsequent associative memory, to estimate the values of the assigned actions (see Fig. 7).

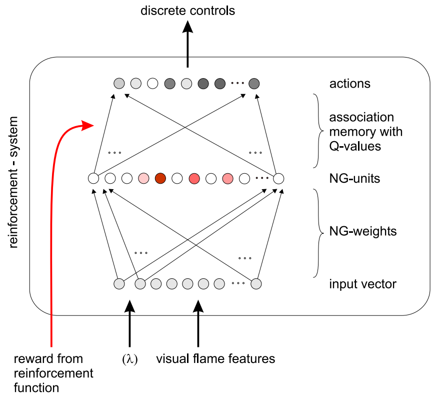

Fig. 7. Neural control architecture for reinforcement-learning of a single agent. The input-vector consists of visual flame-describing features obtained from the camera systems observing the 6 flames of the combustion process. Agent O2 additionally receives the ratio of inlet air and coal (λ). The neural clusterer maps the continuous and high dimensional input-space onto a discrete state-space, whereupon the Q-values of all assigned actions are estimated.

Equation 4.1 shows the neural-gas weight $\underline{w}_k(t)$ update rule for the neuron k, where $\eta^{NG}(t)$ is a learning rate, $i(k)$ is the index of neuron k in the list sorted by distance to the input $\underline{x}(t)$ and $h(t)$ is the learning radius. Thus, the real-valued process describing input data are mapped onto a low dimensional representation s^t.

$$\Delta \underline{w}_k(t) \quad = \quad \eta^{NG}(t) \cdot e^{-\frac{i(k)}{h(t)}} \cdot \left[\underline{x}(t) - \underline{w}_k(t)\right] \tag{4.1}$$

For action-value approximation Q for state s^t and control action a^t, we utilize the Q-learning [2] variant of reinforcement-learning (Eq. 4.2). The usage of expected future returns discounted by γ in addition to the current reward ensures a policy maximizing not only the immediate, but also of long term rewards. Thus, this algorithm is searching a global maximum of the defined reinforcement function.

$$\Delta Q(s^t, a^t) \quad = \quad \eta\Big\{r^t + \gamma V(s^{t+1}) - Q(s^t, a^t)\Big\} \qquad \text{with} \tag{4.2}$$

$$V(s^{t+1}) \quad = \quad \max_a Q(s^{t+1}, a^{t+1}) \tag{4.3}$$

For our experiments, we use a discount factor for the value of the subsequent state of $\gamma = 0.5$, and a Q-learning-rate of $\eta = 0.2$. The reinforcement r is the result of an agent-specific reinforcement function, which strongly corresponds to the plant operator objectives described in section 3. Agents AgentL10, AgentL20 and AgentL30 are rewarded, if the NO_x or the O_2 concentrations decrease and punished, if these concentrations increase (Eq. 4.4). The reinforcement depends on the O_2 concentration, since these agents can only change the distribution of the air, and a reduction of unused oxygen implies, that this redistribution caused a more complete combustion of the coal. Agent AgentO2 is also rewarded, if the NO_x concentration or the total amount of used air decreases (Eq. 4.5). Any violation of thresholds for process data, that are defined by safety precautions of the plant, results in a very strong punishment (Eq. 4.4 and 4.5). The terms K_{NO_x}, K_{O_2} and K_λ allow to balance the importance of the NO$_x$ concentration and the efficiency value. We used for our experiments $K_{NO_x} = K_{O_2} = K_\lambda = 10.0$.

$$r^{AgentLXX} = \begin{cases} -10.0 & : \textit{any threshold violated} \\ K_{NO_x} \cdot \Delta NO_x + K_{O_2} \cdot \Delta O_2 & : \textit{else} \end{cases} \tag{4.4}$$

$$r^{AgentO2} = \begin{cases} -10.0 & : \textit{any threshold violated} \\ K_{NO_x} \cdot \Delta NO_x + K_{air} \cdot \Delta Air & : \textit{else} \end{cases} \tag{4.5}$$

5. Results

To reduce the exploration time for the plant, we pre-trained our multiagent-approach on past process data. This is a kind of supervised reinforcement learning. Figure 8 shows the development of the cluster-errors (left) and of the Q-prediction-error (right). The decreasing cluster error documents the adaptation of the neural gas towards the distribution of process situations in the input space. The Q-error shows the convergence of our function approximator minimizing the Q-prediction error.

After pre-training, we installed the neural multiagent-reinforcement-system on the plant. The exploration behavior of our system was not frozen at this time,

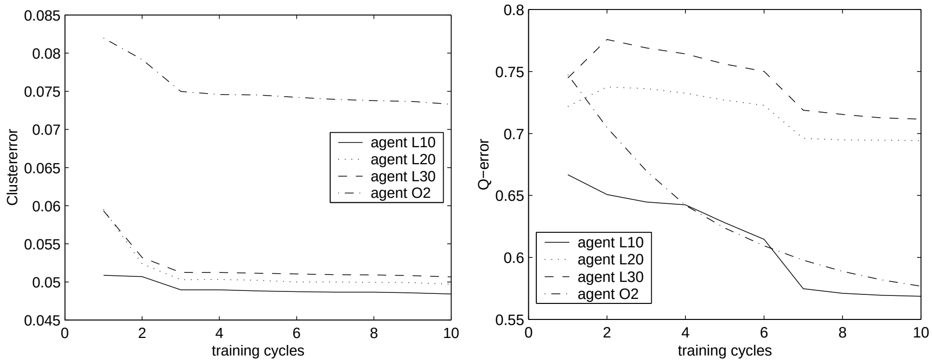

Fig. 8. Development of the cluster errors (left) and the Q–prediction-errors (right) over 10 training cycles on passed process data of the 4 agents. For details see text.

instead we introduced a noise term on all estimated Q-values, which decreases over time to a fixed minimum greater than zero. Thus, we guarantee a fading exploration behavior in favor of the exploitational behavior, whereby always a certain exploration performance remains. Figure 9 depicts the evolving cumulative reinforcements of the 4 agents after the pre-training of the networks.

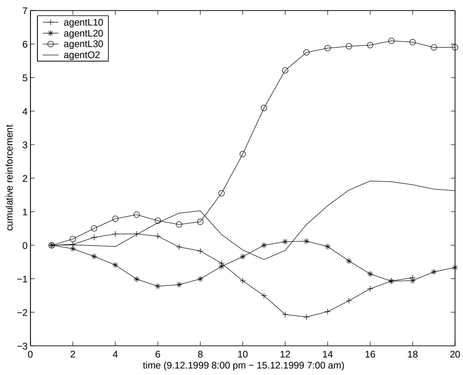

Fig. 9. Development of the cumulative online-reinforcements of the 4 agents during 2 days after pre-training with a relatively high exploration performance. As can be seen, despite the still present exploration performance, especially *AgentL30* and *AgentO2* mostly receive positive reinforcements. This is plausible, since the upper burner level has the strongest influence on the waste gas concentrations, as a consequence of the flow characteristics inside the combustion chamber. *AgentL10* and *AgentL20* performed not so well, probably due to the still ongoing exploration behavior.

In Figure 10 a comparison of the conventional control scheme with fixed air distributions used up to now and our multiagent-reinforcement-system is given. As can be seen, the amount of used air could be reduced significantly by the reinforcement-system (top left). In contrast, both the NO_x and O_2 waste-gas concentrations remained at the same level in these first investigations, whereby we have to remark, that the potential for NO_x and O_2 reduction vanishes with increasing load factors

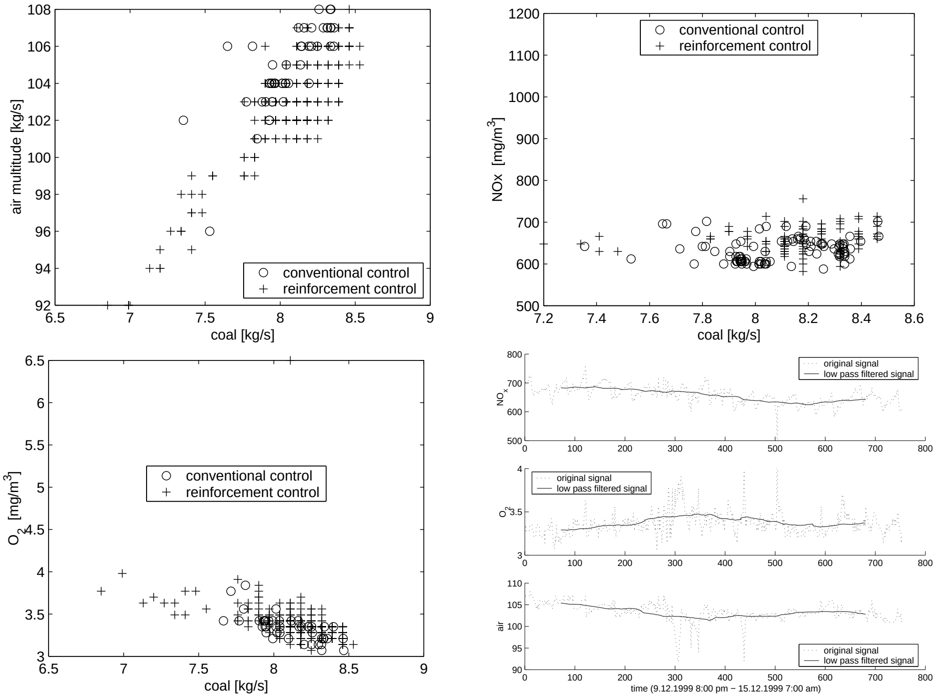

Fig. 10. Comparison of the conventional and our reinforcement-based control schemes by means of consumed air (top left), NO_x (top right) and O_2 concentrations (bottom left) in the waste gas for a time period of about 6 days. Both, the mean NO_x and O_2 waste-gas concentrations remained at the same level, whereas the amount of used air could be reduced significantly by the reinforcement-system. Thus, the efficiency factor could be increased. Also, over the time of its operation, the reinforcement-system was able to decrease the NO_x level (bottom right). The dotted curves represent the original data, whereas the solid line shows the low-pass filtered signal to clarify the claimed changes. During this investigation period, the power plant worked with a load of about 90%.

of the power plant. Figure 10 (bottom right) shows the relevant emission data for the time, during which our reinforcement system was operating. As can be seen, the mean NO_x concentration decreased over time, where both the O_2 concentration and the total air consumption remained almost unchanged.

Of course, this comparison has to be statistically analyzed over many trials, but for industrial processes it is very hard to benchmark different control approaches, where all other process parameters have to be kept constant.

6. Conclusions and Outlook

In this paper we presented a reinforcement-based multiagent approach based on neural networks to control a complex industrial combustion process. To cope with both the tremendous action and situation space of the power plant, we decomposed the complex system into several agents. The proposed multiagent-reinforcement-system consists of 4 agents, which are realized by relatively simple neural function approximators. Neural function approximators are very useful, because they can generalize the expected return of state-action pairs the agent actually experiences to other regions of the state-action-space. Thus, the agent can estimate the expected return of state-action pairs that it has never experienced before.

Future work should address the development of more powerful function approximators than our very simple approach utilizes, because this kind of network tries to approximate the probability density distribution of the input data in the feature space. This data driven statistical learning seems to be insufficient under certain circumstances. For this reason, incremental neural networks are very promising alternatives, for example, the Growing Neural Gas Approach of Fritzke[22] or the life-long learning approach of Hamker.[23] For a faster learning, we also plan to investigate function approximators on the basis of the Adaptive Resonance Theory, e.g., the Fuzzy-Art approach.[24,25] In this context, the stability-plasticity-dilemma has to be addressed, since changing coal qualities, wear and tear, etc. result in time varying process properties and a powerful system has to consider and solve these problems for the use in an industrial process.

Nevertheless, the first results are very promising, but the application of RL-methods to this pretentious control problem is a great challenge.

7. References

1. R.S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 1988.
2. Ch. Watkins and P. Dayan. Q-learning. *Machine Learning 8, 1992*, pages 279–292, 1992.
3. F.O. Karray and C.W. de Silva. Learning and Adaptation in Complex Dynamic Systems. In L.C. Jain and C.W. de Silva, editors, *Intelligent Adaptive Control*, pages 25–40. CRC Press, Boca Raton, London, New York, Washington D.C., 1999.
4. Q.M.J. Wu, K. Stanley, and C.W. de Silva. Neural Control Systems and Applications. In L.C. Jain and C.W. de Silva, editors, *Intelligent Adaptive Control*, pages 63–104. CRC Press, Boca Raton, London, New York, Washington D.C., 1999.
5. G. Baldini, S. Bittanti, A.De Marco, F. Longhi, G. Poncia, W. Prandoni, and D. Vettorelo. A Dynamic Model of Moving Flames for the Analysis and Control of Combustion Instabilities. In *Proceedings of European Control Conference99, Karlsruhe, Germany*, page 585. VDI/VDE Gesellschaft Mess- und Automatisierungstechnik (GMA), 1999.
6. P.S. Chang and H.S. Hou. A fast neural network learning algorithm and its application. In *Proceedings of the 29th Southeastern Symposium on System Theory (SSST'97)*, 1997.
7. H. Maier. *Experimentelle Untersuchungen der Kohlenstaubverbrennung unter Beruecksichtigung der Brennstoffaufbreitung.* PhD thesis, Universitaet Stuttgart, Fakultaet Energietechnik, 1997.
8. S. Wirtz. *Mathematische Modellierung der Kohlenstaubverbrennung.* PhD thesis, Ruhr-Universitaet Bochum, Fakultaet fuer Maschinenbau, 1989.
9. D.A. Handelmann, S.H. Lane, and J.J. Gelfand. Integrating neural networks and knowledge-based systems for intelligent control. *IEEE Control Systems Magazine*, pages 77–86, 1990.
10. S. Gehlen, M. Hormel, and J. Kopecz. Einsatz neuronaler Netze zur Kontrolle kom-

plexer industrieller Prozesse. *Automatisierungstechnik*, 2, 1995.

11. P. Eklund and F. Klawonn. Neural Fuzzy Logic Programming. *IEEE Trans. on Neural Networks*, 3(5), 1992.
12. R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, 1998.
13. J.A. Franklin, R.S. Sutton, and C.W. Anderson. Application of connectionist learning methods to manufacturing process monitoring. In *Proc. of IEEE International Symposium on Intelligent Control 1988, Arlington, VA*, pages 709–712, 1989.
14. H.R. Berenji and P. Khedkar. Learning and tuning fuzzy logic controllers through reinforcements. Technical report, NASA Ames Research Center, 1992.
15. H.R. Berenji, P.W. Chang, and S.R. Swanson. Refining the shuttle training aircraft controller. In *Proceedings of the 6'th IEEE International Conference on Fuzzy Systems, Barcelona, Spain*, pages 677–681, 1997.
16. S. Singh and D. Bertsekas. Reinforcement learning for dynamic channel allocation in cellular telephone systems. In *Advances in Neural Information Processing Systems 10.* 1997.
17. R. Crites and A. Barto. Elevator Group Control using Multiple Reinforcement Learning Agents. *Machine Learning*, 33:235–262, 1998.
18. J.A. Boyan and M.L. Littmann. Packet routing in dynamically changing networks: A reinforcement learning approach. In J.D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 671–678. San Francisco, CA: Morgan Kaufmann, 1993.
19. H.-M. Gross, H.-J. Boehme, and V. Stephan. Sensory-based Robot Navigation using Self-organizing Networks and Q-learning. *World Congress on Neural Networks 96, San Diego USA*, pages 94–99, 1996.
20. H.-M. Gross, V. Stephan, and M. Krabbes. A Neural Field Approach to Topological Reinforcement Learning in Continuous Action Spaces. In *Proc. of WCCI-IJCNN'98, Anchorage*, pages 1992–1997. IEEE Press, 1998.
21. T.M. Martinetz and K. Schulten. A "neural gas" network learns topologies. In T. Kohonen, Mäkisara, K., O. Simula, and J. Kangas, editors, *Artificial Neural Networks*, pages 397–402. Elsevier Amsterdam, 1991.
22. B. Fritzke. A Growing Neural Gas Network Learns Topologies. *Advances in Neural Information Processing Systems 7, MIT-Press, Cambridge MA*, 1995.
23. F. Hamker and H.-M. Gross. A lifelong learning approach for incremental neural networks. In *Proc. of EMCSR'98, Vienna*, pages 599–604, 1998.
24. G.A. Carpenter, S. Grossberg, and D.B. Rosen. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4:759–771, 1991.
25. J.R. Williamson. Gaussian ARTMAP: A Neural Network for Fast Incremental Learning of Noisy Multidimensional Maps. *Neural Networks*, 9(5):881–897, 1996.