Proceedings of SOAVE'2000, Ilmenau, Germany, VDI-verlag Nr. 643, pp. 86-95, 2000.

Farbbildbasierte Prozessführungen in der Kohlenstaubfeuerung mittels Reinforcement-Lernverfahren*

K. Debes, V. Stephan und H.-M. Groß
Technische Universität Ilmenau, Fachgebiet Neuroinformatik
D-98684 Ilmenau (Thür.), PF 100565, Tel. 03677-692858, Fax 691665
Klaus.Debes@informatik.tu-ilmenau.de

H. Wintrich, F. Wintrich
ORFEUS Combustion Engineering GmbH
Kleiststr. 10, 45128 Essen, Tel. 0201-8207230, Fax 0201-8207241
info@orfeus.de

Zusammenfassung

Mit dieser Arbeit soll gezeigt werden, dass die Nutzung von direkt am Verbrennungsprozess gewonnenen visuellen Informationen kombiniert mit einem adaptiven, modellfreien und selbstoptimierenden Reinforcement-Lernsystem eine effektive Prozessführung gestattet.

1 Problemstellung

Ziel der Kohleverbrennung ist im Grunde die maximale Energieausbeute unter Einhaltung definierter Randbedingungen (Emissionen). Deshalb ist es wünschenswert, den Prozess möglichst schnell in eine solche Situation zu führen, in der obige Forderungen optimal erfüllt werden. Gegenwärtige Steuer- und Regelstrategien beruhen auf stark totzeitbehafteten und nur punktförmig erfassten Messgrößen, brauchbare Modelle sind nicht verfügbar, so dass über den „Markov'schen" Charakter im regelungstechnischen Sinne keine Aussagen gemacht werden können. Für das nunmehr unmittelbar am Verbrennungsprozess zur Verfügung stehende Bild der Flamme kann jedoch keine Führungsgröße für einen Regler im klassischen Sinne definiert werden (man kennt kein „optimales" Flammenbild). Deshalb wird ein Prozessführungssystem gesucht, welches durch Exploration innerhalb eines vorgegebenen Arbeitsbereiches selbständig einen optimalen Arbeitspunkt findet und anfährt. Aufgrund dieser Anforderungen bietet sich das aus zahlreichen Veröffentlichungen der Robotik bekannte Reinforcement-Learning (RL) an.
Reinforcement-Lernverfahren sind in ihrer Vorgehensweise ähnlich dem in der Natur vorkommenden Lust- und Schmerz-Prinzip. Durch trial-and-error können mit einfachen, skalaren und evtl. zeitlich verzögerten Bewertungen komplexe Zusammenhänge gelernt werden. RL-Verfahren rangieren mit dieser Strategie zwischen supervised und unsupervised Verfahren. Als Grundlage dient dabei eine Menge von sensorischen Zuständen, in denen sich der Agent befinden kann. Zu jeder dieser Situationen existieren mehrere Aktionsmöglichkeiten, die während des Lernvorganges ausprobiert und entsprechend bewertet werden, so dass dann durch eine geeignete „Politik" ein maximales Bewertungssignal (*reward*) von der Umwelt erreicht werden

*gefördert durch das BMBF, Projekt-Nr.: 032 6843 B

kann. Verwendet werden RL-Algorithmen, wie SUTTON's $TD(\lambda)$ Algorithmus [16], oder WATKINS' Q-learning Algorithmus [17] zur Bewertung von Aktionen (*value function*) in einer Prozesssituation. In Systemen mit kontinuierlichen Zustands- und Aktionsräumen muss diese *value function* mit reell-wertigen Variablen arbeiten, die die Zustände und Aktionen repräsentieren. Typischerweise wird diese *value function* durch *neuronale Funktionsapproximatoren* realisiert, die endliche Ressourcen nutzen, um diese kontinuierlichen Zustands- Aktionspaare zu kodieren. Funktionsapproximatoren sind sehr nützlich, weil sie durch die Generalisierung einen Erwartungswert auch von bisher im Zustandsraum unbekannten Zustands-Aktionspaaren vorhersagen können. Aus der Vielzahl der zur Verfügung stehenden Funktionsapproximatoren muss der erwählt werden, der mit relativ geringem mathematischen Aufwand die bisher erlebten situationsspezifischen Aktionsbewertungen möglichst gut repräsentiert und dabei gleichzeitig noch nicht explorierte Stelleingriffe generalisiert. Im vorliegenden Fall soll Reinforcement-Learning im Sinne folgender Definition verwendet werden: Reinforcement-Learning ist eine Lernmethode, um mittels Versuch und Irrtum in einer Umgebung agieren zu können. Der Agent erhält zu seinen sensorischen Inputs zusätzlich ein numerisches, skalares Reinforcement-Signal, das ein Maß für den „Wert" eines Input-Status ist. Ziel ist das Erlernen einer optimalen Aktions-Auswahlstrategie zur Maximierung der Reinforcements über die Zeit (s. Abb. 1).

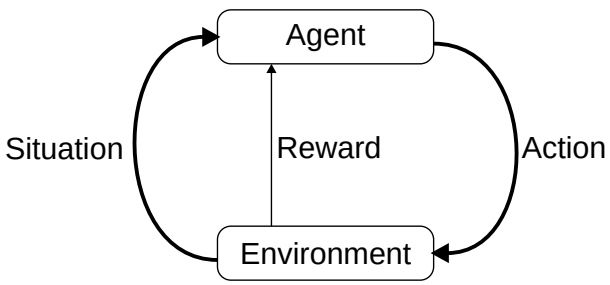

Abbildung 1: Blockschaltbild des Reinforcement-Lernsystems. Die Elemente bedeuten: Agent: Reinforcement-Prozessführung; Action: Stellgrößen; Environment: Verbrennungsprozess; Situation: Prozessgrößen; Reward: Reinforcement-Funktion

2 Stand der Forschung

Große Verbrennungsanlagen zur Wärme- bzw. Stromerzeugung werden mit einem Prozessleitsystem mit standardisierten Steuerelementen überwacht. Dieses System übernimmt u.a. die Visualisierung, Alarmindikation und klassische Prozesssteuerungs- und Regelungsaufgaben. Es gibt eine Reihe von Strategien zur Erzielung besserer Ergebnisse (bezüglich der Emissionen und des Wirkungsgrades), deren gemeinsames Hauptproblem aber die mangelnde Flexibilität unter verschiedenen Prozessbedingungen ist. Alternative Strategien erfordern sehr aufwendige mathematischer Prozessmodelle mit sehr großem rechentechnischen Aufwand, so dass dieser Weg nur bedingt praktikabel erscheint [1, 4, 13, 18]. Eine zweite Gruppe verwendet zur Lösung des Optimierungsproblems wissensbasierte Ansätze [12, 9, 6]. Hier liegt die Schwierigkeit in der sehr komplexen und schwierigen Erstellung der system-abhängigen Wissensbasis, die wiederum die Flexibilität und Portierbarkeit limitiert. Aus diesem Grund schlagen die Autoren ein Modell zur Prozessführung vor, das ohne a priori Wissen und ohne mathematisches Modell des Verbrennungsprozesses auskommt. Das vorgeschlagene RL-System [15] erlaubt eine autonome Exploration im Zustandsraum des Verbrennungsprozesses, bei dem durch den Betreiber vordefinierte Gütekriterien optimiert werden. RL wurde bereits in einer Reihe von real-world Problemen erfolgreich eingesetzt. So sind Anwendungen zur Verbesserung konventioneller Controller und Fuzzy-Controller, zur Fahrstuhlsteuerung und bei Routing-Problemen mit RL veröffentlicht

worden [7, 3, 2, 5]. Im vorliegenden Beitrag soll nun erstmalig ein RL-Einsatz für die Steuerung eines Verbrennungsprozesses in einem industriellen Kraftwerk vorgestellt werden.

3 Das Kraftwerk

Die Versuche wurden in einem Kraftwerk der „Hamburgischen Elektrizitätswerke“ (HEW) im Süden Hamburgs durchgeführt. Das kontrollierte Subsystem bestand aus einem Ofen mit 6 Brennern (je 2 Brenner auf den 3 Ebenen 10, 20 und 30) und einer Maximalleistung von 252 MW (s. Abb. 2). Die beiden Brenner einer Ebene werden durch eine Kohlemühle versorgt, die einen nur theoretisch gleichen Kohlestrom für beide Brenner liefert. Die Verschiebung des Gleichgewichtes und der tatsächliche Betrag des Kohlestromes ist nicht messbar und schließt damit klassische Ansätze wie Regelungen mit Störgrößenaufschaltung aus. Folgende Stellein-

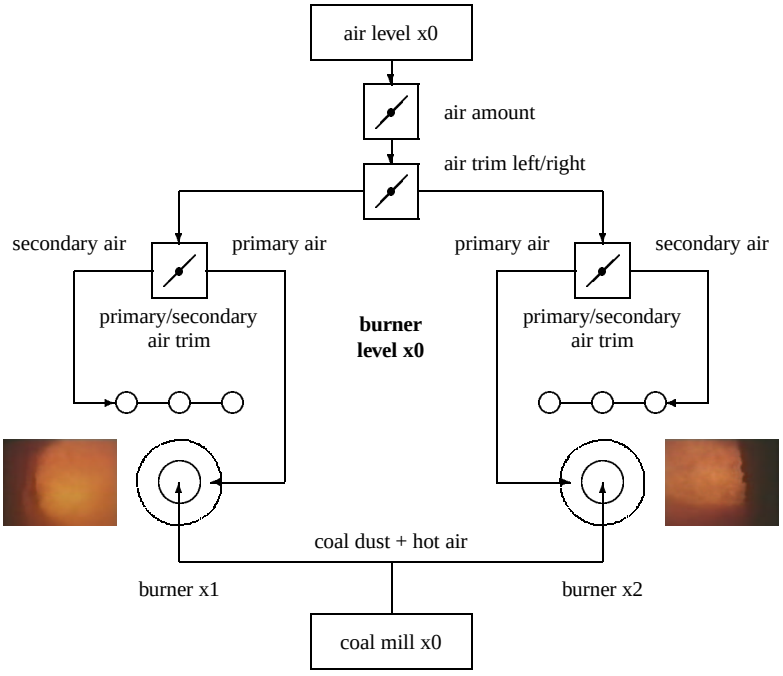

Abbildung 2: Schematische Ansicht der Verbrennungskammer mit Kohle- und Luftversorgung für eine der 3 Brenner-Ebenen.

griffe sind durch das vorgeschlagene System erlaubt:

Steuereingriff	Bedeutung
Primärluftvertrimmung in den Ebenen 10, 20, 30	Luftvertrimmung zwischen dem rechten und linken Brenner einer Ebene
Vertrimmung primär/sekundär Luft an den Brennern 11, 12, 21, 22, 31 and 32	Verteilung zwischen Primär- und Sekundärluft eines Brenners
Luftmenge in den Ebenen 10, 20, 30	Gesamtluftmenge der betrachteten Ebene

Es ist zu bedenken, dass diese 12 Steuereingriffe (s. a. Abb. 2) nur die Luftmenge und deren Verteilung auf die 6 Brenner berücksichtigen – nicht die Menge und Verteilung der eingebrachten Kohlemenge! Zur Reduktion dieses immensen Aktionsraumes nutzen wir relative statt absolute Stelleingriffe, d.h., es werden pro Steuereingriff 3 Aktionen definiert: $+1\%$, 0, -1% (Die Anwendung von absoluten Steuereingriffen mit nur 10 Abstufungen ergibt einen Aktionsraum von 10^{12} möglichen Aktionen. Selbst unter Verwendung der relativen Stelleingriffe bleibt ein

Aktionsraum von $3^{3+6+3} = 531.441$).
Nach Beschreibung der zur Verfügung stehenden Stelleingriffe sollen die zur Prozessführung verwendeten Informationen beschrieben werden. Normalerweise werden alle Prozessdaten außerhalb der Brennkammer gemessen (z.B. im Abgasteil der Anlage), so dass keine direkten Informationen über den Verbrennungsprozess selbst zur Verfügung stehen. Genau diese Informationen (Kohlemenge, Kohleverteilung, Flammenform, Temperatur) sind aber eigentlich erforderlich, um den Prozess *gezielt* zu beeinflussen. Deshalb werden alle 6 Brenner mit Spezial-Farbkameras überwacht, die von der Orfeus Combustion Engineering GmbH speziell für diesen Zweck entwickelt wurden. Die Abbildung 3 zeigt die Extraktion der Flammenmerkmale zur Beschreibung des Verbrennungsprozesses.

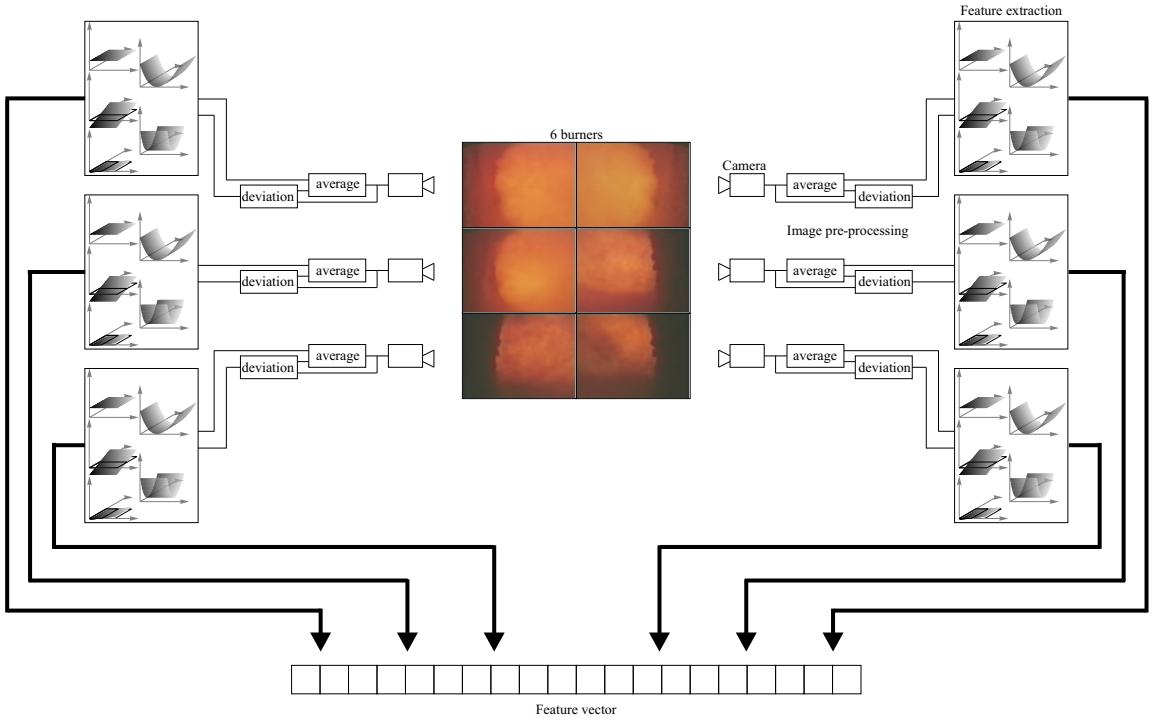

Abbildung 3: Schema der Extraktion visueller Merkmale aus dem Verbrennungsprozess. Aus den gewonnenen Kamerabildern werden Fitwerte berechnet, die die globale Flammenform beschreiben. Rechts bzw. links sind die Filtermasken angedeutet, mit denen jedes der 6 zuvor über einen definierten Zeitraum gemittelten Flammenbilder skalar verknüpft wird. Aus den so ermittelten Fitwerten entsteht ein Vektor, der die globale Form der jeweiligen Flamme beschreibt und Korrelationen mit Prozessdaten aufweist.

4 Architektur

Wie in Abschnitt 1 erwähnt, muss unsere Architektur für jede Prozesssituation eine Aktion auswählen, die den höchsten *reward* erwarten lässt. Das Schlüsselproblem dabei ist der ungeheure Aktionsraum in Kombination mit einem sehr breit gefächerten Eingangssignalraum. Ein üblicherweise verwendeter monolithischer Architekturansatz ist in diesem Fall nicht sehr sinnvoll, da die Explorationszeit, in der alle Aktionen für alle möglichen Situationen vom System erprobt werden können, für die vorliegende Prozessführung nicht praktikabel ist. Mit der oben genannten Zahl für die möglichen Stelleingriffe und einer (sehr sparsamen) Charakterisierung des Zustandsraumes mit 100 Knoten ergibt sich bei einer Versuchsdauer pro Stelleingriff von

10 min eine Explorationszeit von über 1000 Jahren! In diesem Zeitraum wäre dann jeder Prozesszustand mit jedem möglichen Stelleingriff genau einmal erprobt worden.

4.1 Problemdekomposition

Konsequenterweise schlagen wir aus diesem Grunde ein System mit mehreren Agenten vor, die jeder für sich einen relevanten Teil des Eingangssignalraumes kontrollieren. Die Abbildung 4 (links) zeigt die Dekomposition in 4 Agenten mit ihren Eingängen und korrespondierenden Prozesseingriffen. AGENTL10, AGENTL20, und AGENTL30, steuern die Luftverteilung auf der jeweiligen Brennerebene, während AGENTO2 die Gesamtluftmenge für jede Brennerebene kontrolliert. Dadurch reduzieren sich die Dimensionen der einzelnen Agenten wie folgt.

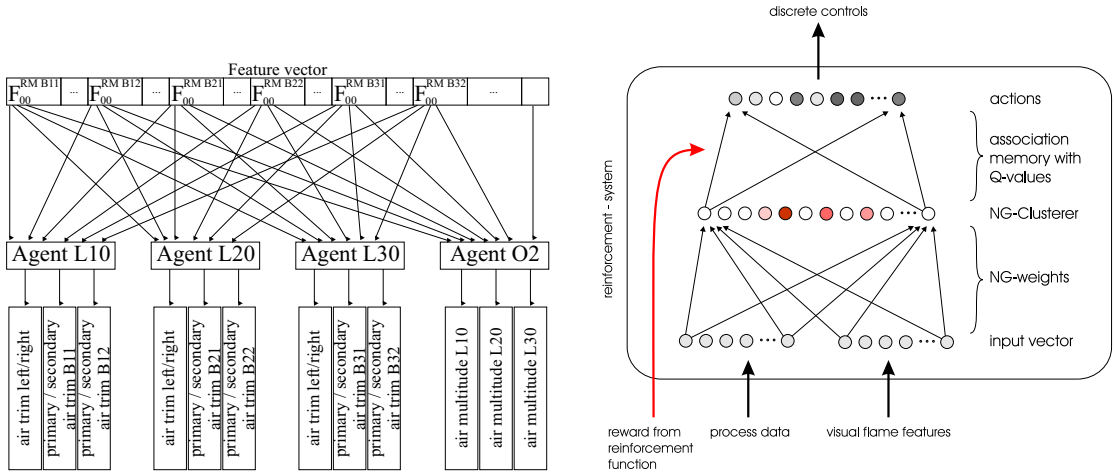

Abbildung 4: Links: Aufteilung der situationsbeschreibenden Größen und der Stellgrößen auf die 4 Agenten. Jeder Agent beobachtet einen relevanten Teil des Situationsraumes und hat Zugriff auf definierte Steuereingriffe. Rechts: Neuronale Steuerarchitektur für das RL-System eines Agenten. Der Input-Vektor besteht aus den Merkmalen, die die Flamme beschreiben; der Neuronale Clusterer bildet den kontinuierlichen und hochdimensionalen Eingangssignalraum auf einen diskreten Zustandsraum ab, mit dem die Q-Werte der ausgewählten Aktionen bestimmt werden.

	AgentL10	AgentL20	AgentL30	AgentO2
Eingänge	3	3	3	7
Anzahl Neuronen				
Neural-Gas-Klassifikator	20	20	20	50
Aktionen	27	27	27	27

Der Einsatz eines Multiagentensystems erfordert im Gegensatz zu einem monolithischen Ansatz die Definition eines Zeitregimes, welches festlegt, wann welcher Agent seinen Aktionsvorschlag real ausführen darf. Das ist notwendig, weil der Reinforcement-Lernvorgang die Auswirkungen der Aktionen zur Adaption seiner Aktionsauswahl (policy) nutzt. Da im vorliegenden Fall die Auswirkungen der Aktionen der 4 Agenten nicht voneinander trennbar sind (alle Stellgrößen beeinflussen die NO_x-Werte, welche Bestandteil der Reinforcement-Funktionen aller 4 Agenten sind), darf zu einem Zeitpunkt nur einer der 4 Agenten seine Aktion real ausführen. Auf diese Weise können alle Veränderungen am Prozess als Resultat dieses Stelleingriffes betrachtet und dem jeweiligen Agenten angerechnet werden. Für die Umsetzung im Versuchskraftwerk wurde empirisch unter Berücksichtigung von anlagenspezifischen Trägheiten des Verbrennungsprozesses ein Zeittakt für die Ausführung von Aktionen von 10 min gewählt.

4.2 Neuronaler Funktions-Approximator

Jeder der 4 Agenten enthält einen Neuronalen Funktions-Approximator, für den hier ein erster, einfacher Ansatz vorgestellt werden soll. Da die Q-Werte im R^n nicht linear von den Eingangsgrößen abhängig sind, scheiden Ansätze mit linearer Regression aus. Unter den Randbedingungen, dass der gewählte Approximator möglichst genau arbeiten soll, schnell belehrbar ist und die Möglichkeit des online-Lernens bietet, kombinierten wir einen neuronalen Vektor-Quantifizierer (Neural Gas [14]) für die optimale Clusterung des hochdimensionalen, kontinuierlichen Eingangsraumes [10] (s. Gleichung 1) mit einem Assoziativspeicher zur Bewertung der möglichen Aktionen (s. Abb. 4). Gleichung 1 zeigt die Berechnung der Wichtungen des Neural Gas - Netzwerkes $\underline{w}_k(t)$ für jedes Neuron k, wobei $\eta^{NG}(t)$ die Lernrate ist, $i(k)$ der Index des Neurons k sortiert nach dem Abstand zum Eingangsvektor $\underline{x}(t)$ und $h(t)$ der Lernradius.

$$\Delta\underline{w}_k(t+1) \;=\; \eta^{NG}(t) \cdot e^{-\frac{i(k)}{h(t)}} \cdot [\underline{x}(t) - \underline{w}_k(t)] \tag{1}$$

Später durchgeführte Untersuchungen bestätigten die Richtigkeit dieses Ansatzes. Für die action-value Approximation Q des Zustandes s^t und der Aktion a^t wird eine Q-learning Variante ([17]) des Reinforcement-Learning verwendet (Gleichung 2).

$$\Delta Q(s^t, a^t) \;=\; \eta\Big\{r^t + \gamma V(s^{t+1}) - Q(s^t, a^t)\Big\} \qquad \text{mit} \tag{2}$$

$$V(s^{t+1}) \;=\; \max_a Q(s^{t+1}, a^{t+1}) \tag{3}$$

Für unsere Experimente nutzten wir einen Discount-Faktor von $\gamma = 0.5$ und eine Q-Lernrate von $\eta = 0.2$. Das Reinforcement r ist das Ergebnis einer Agenten-spezifischen Reinforcement-Funktion: Die Agenten AGENTL10, AGENTL20 und AGENTL30 erhalten eine Belohnung, wenn die NO_x oder die O_2 Konzentration sinkt und eine Bestrafung, wenn diese Konzentrationen steigen (Gleichung 4). Das Reinforcement hängt auch von der O_2 Konzentration ab, weil diese Agenten durch eine bessere *Verteilung* der Luft eine vollständigere Verbrennung der Kohle erzielen können.
Agent AGENTO2 erhält ebenfalls eine Belohnung, wenn die NO_x Konzentration oder der totale Betrag der verbrauchten Luft sinkt (Gleichung 5).
Unabhängig von diesen Aktionsbewertungen wurde aus sicherheitstechnischen Gründen ein Arbeitsbereich für das Reinforcement-System durch Schwellwerte definiert. Bei Verlassen des Arbeitsbereiches erfährt das System ein stark negatives Feedback (Gleichungen 4 und 5). Die Terme K_{NO_x} und K_λ legen eine Balance zwischen der Bedeutung der NO_x Konzentration und des Wirkungsgrades der Anlage fest. In den Versuchen wurde $K_{NO_x} = K_\lambda = 0.5$ verwendet.

$$r_{AgentLXX} = \begin{cases} -10.0 & : \; beliebige\ Schwelle\ überschritten \\ K_{NO_x} \cdot \Delta NO_x + K_{O_2} \cdot \Delta O_2 & : \; sonst \end{cases} \tag{4}$$

$$r_{AgentO2} = \begin{cases} -10.0 & : \; beliebige\ Schwelle\ überschritten \\ K_{NO_x} \cdot \Delta NO_x + K_{air} \cdot \Delta Air & : \; sonst \end{cases} \tag{5}$$

5 Ergebnisse

Zur Reduzierung der Explorationszeit wurde der verwendete Multiagenten-Ansatz mit zuvor aufgezeichneten Prozessdaten vortrainiert. Die Abbildung 5 zeigt den Verlauf des Clusterfehlers (links) und des Reinforcement-Fehlers (rechts). Der sinkende Clusterfehler dokumentiert die Adaption des Neural Gas - Netzwerkes auf die Verteilung der Prozesssituationen im Eingangssignalraum, während der Q-Fehler die Konvergenz des Funktions-Approximators zur Minimierung des Vorhersagefehlers zeigt.

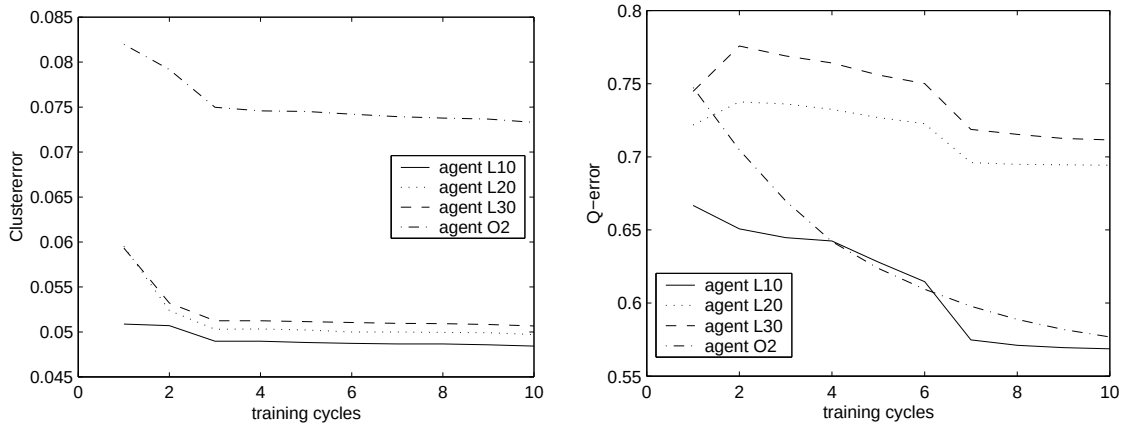

Abbildung 5: Entwicklung des Cluster-Errors (links) und des Reinforcement-Fehlers (rechts) über 10 Trainingszyklen aller 4 Agenten (s. Text).

Nach dem Vortraining wurde das Multiagenten-Reinforcement-System im Kraftwerk implementiert. Zur Sicherung des Explorationsverhaltens wurden die Q-Werte mit einem Rauschterm beaufschlagt, der im Verlauf der Zeit zu einem festen Wert größer Null abgesenkt wurde. Auf diese Weise wird eine zunehmende Exploitation des erlernten Verhaltenswissens realisiert, wobei gleichzeitig ein gewisses Explorationsverhalten die Anpassungsfähigkeit dieses Ansatzes an wechselnde Prozesseigenschaften (Kohlewechsel, Verschleiß) garantiert. Abbildung 6 zeigt die kumulativen Reinforcements aller 4 Agenten auf der Basis des vortrainierten Netzwerkes. Abbildung 7 (links) zeigt einen Vergleich der konventionellen Betriebsweise mit fester

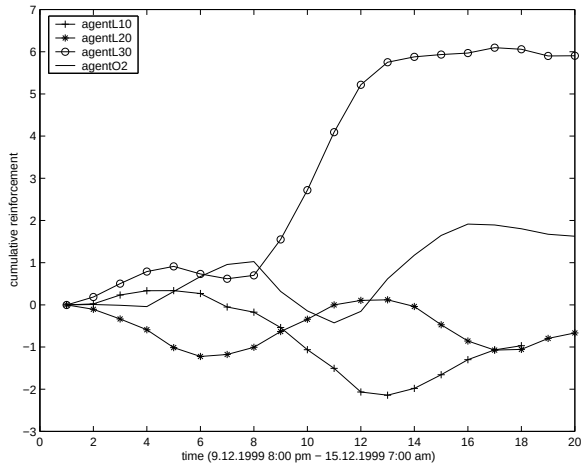

Abbildung 6: Entwicklung des kumulativen Reinforcements aller 4 Agenten im Verlauf von 2 Tagen nach einem Vortraining und relativ hohem Explorationsfaktor. Wie man sieht, erhalten besonders *AgentL30* und *AgentO2* trotz der permanent vorhandenen Exploration meistens positive Reinforcements. Dies ist plausibel, da die oberen Brennerebenen als Konsequenz auf die Strömungsverhältnisse im Verbrennungsofen den stärkeren Einfluss auf die Abgaskonzentrationen haben.

Luftverteilung und unserem Multiagenten-Reinforcement-System. Man kann sehen, dass der Betrag der eingesetzten Luftmenge mit dem RL-System signifikant geringer wird (links). Im Gegensatz dazu bleiben die NO_x und O_2 Abgaskonzentrationen bei diesen ersten Versuchen etwa gleich, wobei zu beachten ist, dass das Potenzial zur NO_x- und O_2-Senkung mit geringer werdendem Lastfaktor des Anlage steigt. Abbildung 7 (rechts) zeigt die relevanten Emissionsdaten mit sinkender NO_x Konzentration und sinkendem Lufteinsatz für die Betriebszeit mit dem RL-System.

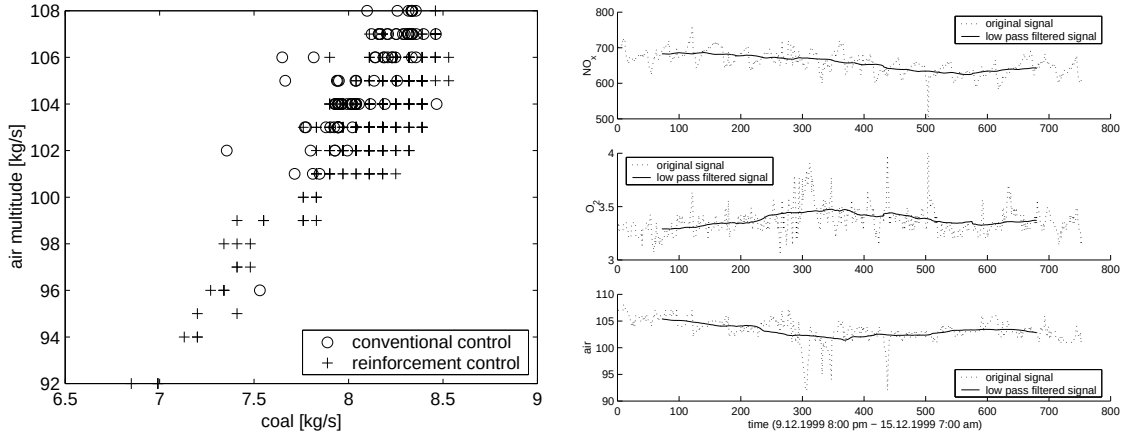

Abbildung 7: Vergleich von konventioneller und RL-basierter Prozessführung bezüglich verbrauchter Luft (links) über eine Testperiode von 6 Tagen. Der Betrag der eingesetzten Gesamtluftmenge konnte durch das RL-System signifikant gesenkt werden (= Wirkungsgraderhöhung). Über den gesamten Versuchszeitraum konnte das Reinforcement-System sowohl NO_x als auch Luftverbrauch kontinuierlich senken (rechts). Die Anlage lief mit 90% Last.

6 Zusammenfassung und Ausblick

Im vorliegenden Beitrag wird ein auf neuronalen Netzwerken basierendes Multiagenten-Reinforcement System für die Führung eines industriellen Verbrennungsprozesses eingesetzt. Zur Bewältigung des gewaltigen Aktions- und Zustandsraumes einer solchen Verbrennungsanlage wurde das komplexe System in mehrere Teile zerlegt. Das vorgeschlagene Multiagenten-Reinforcement-System besteht aus 4 Agenten, die durch relativ einfache neuronale Funktionsapproximatoren realisiert werden. Letztere sind sehr nützlich, da sie die erwartete Aktion der Zustands-Aktions-Paare eines Agenten sehr gut im Zustands-Aktions-Raum generalisieren können. Aus diesem Grund kann der Erfolg von Zustands-Aktions-Paaren bewertet werden, die vorher nie erprobt worden sind.
Zukünftige Arbeiten werden der Entwicklung verbesserter Funktionsapproximatoren dienen, da die verwendete Art nur die Wahrscheinlichkeitsdichteverteilung der Eingangsdaten im Merkmalsraum abbildet. Unter bestimmten Umständen ist diese rein datengetriebene Form des statischen Lernens nicht sinnvoll. Alternativen wären inkrementelle neuronale Netzwerke, wie z.B. das Growing Neural Gas nach Fritzke [8] oder das life-long learning nach Hamker [11].
In diesem Zusammenhang spielt das Stabilitäts-Plastizitäts-Dilemma eine wichtige Rolle, da die Änderung der Kohlequalität, Verschleißerscheinungen an der Anlage usw. den Prozess im Laufe der Zeit maßgeblich beeinflussen. Die ersten Resultate sind sehr vielversprechend – der weitere Einsatz der RL-Methoden ist eine anspruchsvolle Herausforderung für derartig komplexe und schwer exakt erfassbare Industrieprozesse.

Literatur

[1] 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 Conference´99, Karlsruhe, Germany*, page 585. VDI/VDE Gesellschaft Mess- und Automatisierungstechnik (GMA), 1999.

[2] H. Berenji, P. Chang, and S. Swanson. Refining the shuttle training aircraft controller. In *Proc. of the 6'th IEEE International Conference on Fuzzy Systems, Barcelona, Spain*. IEEE Press, 1997.

[3] H. Berenji and P. Khedkar. Learning and tuning fuzzy logic controllers through reinforcements. Technical report, NASA Ames Research Center, 1992.

[4] P. Chang and H. Hou. A fast neural network learning algorithm and its application . In *Proceedings of the 29th Southeastern Symposium on System Theory (SSST'97)*. 1997.

[5] R. Crites and A. Barto. Elevator Group Control using Multiple Reinforcement Learning Agents. *Machine Learning*, 33:235–262, 1998.

[6] P. Eklund and F. Klawonn. Neural Fuzzy Logic Programming. *IEEE Trans. on Neural Networks*, 3(5), 1992.

[7] J. Franklin, J. Sutton, and C. Anderson. Application of connectionist learning methods to manufacturing process monitoring . In *Proc. of IEEE International Symposium on Intelligent Control, Arlington, VA*, pages 709–712. IEEE Press, 1988.

[8] B. Fritzke. A Growing Neural Gas Network Learns Topologies. *Advances in Neural Information Processing Systems 7, MIT-Press, Cambridge MA*, 1995.

[9] S. Gehlen, M. Hormel, and J. Kopecz. Einsatz neuronaler Netze zur Kontrolle komplexer industrieller Prozesse. *Automatisierungstechnik*, 2, 1995.

[10] 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.

[11] F. Hamker and H.-M. Gross. A lifelong learning approach for incremental neural networks. In *Proc. of EMCSR'98, Vienna*, pages 599–604, 1998.

[12] D. Handelmann, S. Lane, and J. Gelfand. Integrating neural networks and knowledge-based systems for intelligent control. *IEEE Control Systems Magazine*, pages 77–86, 1990.

[13] H. Maier. *Experimentelle Untersuchungen der Kohlenstaubverbrennung unter Beruecksichtigung der Brennstoffaufbreitung*. PhD thesis, Universitaet Stuttgart, Fakultaet Energietechnik, 1997.

[14] 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.

[15] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[16] R.S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 1988.

[17] Ch. Watkins and P. Dayan. Q-learning. *Machine Learning 8, 1992*, pages 279–292, 1992.

[18] S. Wirtz. *Mathematische Modellierung der Kohlenstaubverbrennung*. PhD thesis, Ruhr-Universitaet Bochum, Fakultaet fuer Maschinenbau, 1989.