ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 237-242

Application of a Self-Learning Controller with Continuous Control Signals Based on the *DOE*-Approach

Martin Riedmiller
Institut für Logik, Komplexität und Deduktionssysteme
Universität Karlsruhe, D-76128 Karlsruhe, Germany
e-mail: riedml@ira.uka.de

Abstract.
The paper introduces the concept of ***dynamic output elements (DOE)***, a novel approach to generate continuous control signals within a self-learning neural control architecture. Using *DOEs* the control signal is not determined by the decision of the control unit directly, but rather dependent on the ***temporal sequence*** of these decisions. Thus the basic mechanisms of decision making and learning are preserved. The benefits of the *DOE*-architecture are shown on a challenging benchmark of learning to control a highly nonlinear chemical reactor.

1. Introduction

Dynamic programming techniques have proven to provide a powerful foundation for learning control of dynamic systems in a self learning fashion (Barto *et al.*, 1995). Given a discrete time dynamic system

$$x_{t+1} = f(x_t, u_t) \tag{1}$$

the idea is to formulate the control problem as the search for a policy π^* that minimizes the accumulated costs for a control trajectory:

$$V^*(x) = \min_{\pi} \sum_{t=0}^{\infty} r(x_t, \pi(x_t)), x_0 = x$$

Optimization problems of the above kind can be solved using dynamic programming techniques. One such method, called *value iteration*, is especially suited for the on-line solution of control problems, when only a minimum of training information is available (Barto & Crites, 1996), (Dietterich & Zhang, 1995), (Sutton, 1996). The main idea is to approximate the optimal value function by an iterative improvement of the estimation of $V^*(x)$. In our work, we use a multi-layer feed-forward neural network to learn to approximate the current value function (Riedmiller, 1996). Assuming that the optimal value function is finally approximated, then the optimal policy can be computed by stepping through a finite set of available actions, selecting the action that results in the minimum accumulated costs:

$$\pi^*(x_t) = a_t = \min_{a \in \mathcal{A}}\{r(x_t, a) + V^*(f(x_t, a))\} \tag{2}$$

Thus the control signal applied to the plant $u_t = a_t$ is one out of a *finite* set of available actions $\mathcal{A}$. From the viewpoint of the optimization process, the action set $\mathcal{A}$ should be kept as small as possible, in order to reduce the number of possible policies. Each additional action will lead to an exponential increase of search space - and thus will make the learning task considerably more difficult. On the other side, a small number of control signals will lead to coarse control of the plant which may not fulfill practical requirements. In the following we introduce a new concept called *dynamic output elements (DOE)*, that is able to deal with the above dilemma. Even continuous control signals can be generated, while the principal working scheme is preserved.

2. Dynamic Output Elements

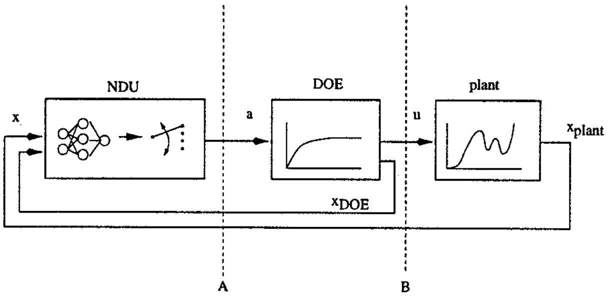

Figure 1: *DOE* approach: The controller consists of the combination of a neural decision unit (NDU) and a *dynamic output element (DOE)* (line **B**). The control signal u is determined by the output of the *DOE*. From the viewpoint of the neural controller (NDU), it has to control the combined system of *DOE* and plant (line **A**).

In the basic approach described in section 1. the control signal is *directly* determined by the action minimizing equation (2). This selection process based on the neural value function is performed by the neural decision unit (NDU). The idea of the concept of *dynamic output elements (DOE)* is to put an additional dynamic element between the output of the selection process and the input of the plant. The output of this *DOE* thus determines the control signal $u(t)$, while its input is the result of a discrete selection process (figure 1). In state space notation, the dynamic behavior of the *DOE* can be described by the following equations:

$$\xi_{t+1} = \varphi(\xi_t, a_t) \quad \text{and} \quad v_t = \psi(\xi_t) \tag{3}$$

where ξ_t denotes the current state vector of the *DOE* and v_t denotes its output. The input of the *DOE* is the action selected by equation (2), a_t. The output v_t determines the control signal u_t that is applied to the plant. Thus, u_t is a function of the action and the current state of the *DOE*, i.e. $u_t = \psi(\varphi(\xi_t, a_t))$. Putting this in the dynamic equation of the plant (1) we get:

$$x_{t+1} = f(x_t, \psi(\varphi(\xi_t, a_t))) = f'(x_t, \xi_t, a_t)$$

where f' denotes the function obtained by the concatenation of the functions f, ψ and φ. If we now regard the combination of *DOE* and plant as a new dynamic system, we can describe its behavior by the following equation:

$$\begin{pmatrix} x_{t+1} \\ \xi_{t+1} \end{pmatrix} = \begin{pmatrix} f'(x_t, \xi_t, a_t) \\ \varphi(\xi_t, a_t) \end{pmatrix}$$

This last equation expresses the crucial point of the concept of *dynamic output elements*: From the viewpoint of the neural decision unit, the combination of *DOE* and plant can be seen as another dynamic system with input a_t and state $\bar{x}_t = (x_t, \xi_t)$. Thus the task of the decision process basically remains the same: It now has to select an appropriate action to control the *combined* system. The state of this new system is the combination of the state of the *DOE* and the state of the plant. Thus the neural decision unit follows the same basic principles as described in section 1.. Its aim now is to learn an optimal control for the combined system of *DOE* and plant.

From the viewpoint of the plant (line B in figure 1), the control signal is computed by the output of the *DOE*. Since the *DOE* has dynamic behavior, its output is not determined by its current input directly, but it is rather dependent on the *temporal sequence* of control decisions made by the neural decision unit. In other words, the amplitude of the control signal is coded in the sequence of control decisions. The resulting control signals are not any longer restricted by the number of available decisions, but rather dependent on their adequate ordering in time. So the *DOE* approach is perfectly embedded in the underlying framework of sequential decision making.

By choosing the dynamic behavior of the *DOE*, we can obtain several effects on the resulting control signal. As an example, in the following a *DOE* with I-T$_1$behavior is described. However, the *DOE* approach is a very general concept, and various kinds of useful *DOE* behavior can be adequate depending on the respective control task.

2.1. The I-T$_1$-*DOE*

The range of the amplitude of an appropriate control signal may vary significantly according to the current situation. Intuitively, in a situation 'far away' from the target, a large control signal should be applied to quickly carry the system over to the target region, whereas within this region, a more cautious policy seems adequate. This is the motivation for a special *DOE* with so called I-T$_1$-behavior. The dynamics of this *DOE* are described by the following differential equation:

$$v(t) + T\,\ddot{v}(t) = a(t)$$

where $a(t)$ denotes the input and $v(t)$ denotes the current output of the *DOE*. The parameter T influences the 'quickness' with which the *DOE* reacts to a certain input signal. The working behavior of the I-T$_1$-*DOE* can be described as follows: The incoming signals are integrated over time. This is combined by an additional low-pass filter that smoothes the outgoing signal. A sequence of input signals with the same sign will lead to a smooth increase of the output signal, a sequence of signals with opposite sign will decrease the output. Thus a neural I-T$_1$-*DOE*-controller can generate continuous control signals with varying amplitudes.

3. Control of a chemical plant

The control of a chemical plant offers a challenging benchmark for nonlinear controller design in general. The task can be shortly described as follows (for a detailed description see (Föllinger, 1993)): In a reactor there is a chemical substance with concentration x_1. The substance chemically reacts with the fluid in the reactor in an energy-emitting decay process. This leads to a raise of the temperature x_2 in the reactor. On the other side, the temperature x_2 influences the rate of decay of the substance. Thus we got two highly interacting processes of concentration and temperature coupled via a nonlinear function $\gamma(x_1, x_2)$:

$$\begin{aligned} \dot{x_1} &= -a_1\,x_1 + \gamma(x_1, x_2) \\ \dot{x_2} &= -a_{21}\,x_2 + a_{22}\,\gamma(x_1, x_2) + b\,u \\ \gamma(x_1, x_2) &= (1 - x_1)\,k_0\,e^{-\frac{\epsilon}{1+x_2}} \end{aligned} \tag{4}$$

The aim of control is to keep the reactor producing a certain level of concentration by the control of external heating or cooling. In the following two analytical control approaches are compared to a self-learning neural controller using an I-T$_1$-*DOE*.

3.1. Analytical controller design

Two approaches of analytical controller design are considered for comparison. We just give the formulae here, for a detailed discussion the reader is referred to (Föllinger, 1993). The linear control law is given by:

$$u = -0.8\,y$$

The equation for the computation of the control signal in case of a nonlinear control law is much more complicated:

$$u = -k(y)\,y = -[C_0\,y + C_1\,y\,e^{-\frac{-\epsilon}{1+y}} + C_2\,y\,\frac{e^{-\frac{-\epsilon}{1+y}} - e^{-\frac{-\epsilon}{1+y_R}}}{y - y_R}]$$

One can immediately see two things from this expression: firstly, the control law is rather complicated and difficult to derive, and secondly much a priori information of both structure and parameters of the plant is integrated in the final control law.

3.2. Self-learning neural control

The self-learning neural controller has no knowledge about the nonlinear dynamic behavior - it just observes the result of its current policy with respect to success or failure in reaching the final concentration (for a more detailed description of the self-learning framework the reader is referred to (Riedmiller, 1996)). The following reports the results of applying the controller using an I-T_1-*DOE* in comparison to linear and nonlinear controller design. Starting at a concentration of $x_1 = 0.42$ the heating and cooling of the reactor has to be controlled until the new target concentration of $x_1 = 0.72$ has been reached. Figure 2(a) shows the temporal behavior of the concentration for the three different controllers. The linear controller behaves worst: It considerably overshoots the target and reaches the desired concentration after about 7 minutes. Applying the analytical nonlinear controller gives a better result: Overshooting of the target concentration is drastically reduced.

The performance of the neural controller overcomes both of the analytical approaches. It nearly avoids overshooting completely, reaching the final target concentration after less than 6 minutes. This is the more remarkable since in contrast to the analytical controllers, no knowledge about plant behavior is used and the control strategy is *learned.* The control signal is plotted in figure 2(b). By the adequate use of the features of the I-T_1-*DOE*, the controller has learned to generate a continuous control signal with varying amplitudes. This is of special importance for nonlinear plants, where the appropriate range of control signals may strongly depend on the situation.

4. Conclusion

The article introduces the idea of *dynamic output elements (DOE)* as a general concept to produce arbitrary control signals within the framework of a self learning neural controller. The general idea of learning to make a decision out of a finite set of available actions is preserved. Actually, this set can be kept small in order to achieve good conditions for the underlying learning process. The ability of producing a wide range of control signals arises from the *dynamic* behavior of the *DOE*: The control signal is determined by the *sequence* of decisions produced by the neural decision process. The application to a highly

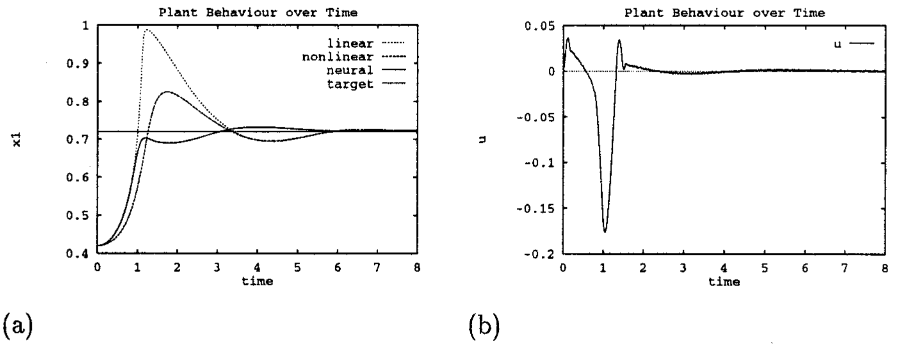

Figure 2: Control of a nonlinear chemical reactor - comparison of analytical linear and nonlinear controller design and self-learning neural controller. (a) Concentration of the substance (b) Control signal generated by the neural I-T_1-*DOE* controller

nonlinear chemical plant shows the favorable qualities of a self-learning neural controller with an I-T_1-*DOE* in comparison to analytically derived control laws.

References

Barto, A. G. and R. H. Crites (1996) Improving elevator performance using reinforcement learning. In D. S. Touretzky, M. C. Mozer, M. E. Hasselmo, editor, *Advances in Neural Information Processing Systems 8.* MIT Press.

Barto, A. G., S. J. Bradtke, and S. P. Singh (1995) Learning to act using real-time dynamic programming. *Artificial Intelligence*, (72):81–138.

Dietterich, T. and W. Zhang (1995) A reinforcement-learning approach to job-shop scheduling. In *Proceedings of the 14.th International Joint Coference on Artificial Intelligence.*

Föllinger, O. (1993) *Nichtlineare Regelungen.* Oldenbourg, 7. edition.

Riedmiller, M. (1996) Learning to control dynamic systems. In Trappl, Robert, editor, *Proceedings of the 13th. European Meeting on Cybernetics and Systems Research - 1996 (EMCSR '96)*, Vienna.

Sutton, R. S. (1996) Generalization in reinforcement learning: Successful examples using sparse coarse coding. In D. S. Touretzky, M. C. Mozer, M. E. Hasselmo, editor, *Advances in Neural Information Processing Systems 8.* MIT Press.