Reinforcement Learning based Neurocontrollers

Erik Cuevas[1,2], Daniel Zaldivar[1,2], Marco Perez[2] and Raúl Rojas[1]

[1] Freie Universität Berlin, Takustr. 9
14195 Berlin, Germany
{cuevas, zaldivar, rojas}@inf.fu-berlin.de
http://www.inf.fu-berlin.de
[2] Universidad de Guadalajara, CUCEI, Av. Revolución 1500,
44430 Guadalajara, Jalisco, Mexico
marcop@cucei.udg.mx

(Paper received on June 22, 2007, accepted on September 1, 2007)

Abstract. An important characteristic of a controller is the adaptation to dynamic changes in the system to be controlled; however most of the algorithms used consist on complex and computationally expensive structures. This paper presents a controller based on radial neural networks which is able to perform parameter adaptation as a response to changes on the system dynamics using a simple reinforcement learning rule. The approach has been tested on a typical non-linear benchmark plant: the steering ship control.

1 Introduction

Conventional controllers are only efficient where the system to be controlled (the plant) or rather the model of that system represented within the controller is characterized by constant parameters applicable at all operating points. And yet, most complex systems are characterized by parameters that vary with the system operating point, thus failing to meet the basic assumption just stated.

Neural networks have been successfully applied on identification and control of dynamical systems. They are also regarded as good approximation elements for modelling non-linear systems, with remarkable results on designing neurocontrollers [1]. Advantages seem evident as they do not require the plant's model as a proper selection on data from system's input and output might suffice.

There exist several neural architectures for controlling dynamic systems [2]. The design procedure includes two steps, first the identification of the plant dynamics and second the controller generation from the inverse approximation of operational data. Following this assumption, the new controller would have the ability to control the plant by constantly keeping the same dynamic characteristics, which is not sometimes the case.

Radial function neural networks may require more neurons than a classical feed-forward network. However, the weights on the overall structure hold a semantic meaning which in turn allows a customized update rule and therefore approaches each neu-

Special Issue in Neural Networks and Associative Memories
Research in Computing Science 28, 2007, pp. 173-182

ron accordingly to its overall contribution [3], and therefore have been considered as an attractive proposal for online adaptation.

Considering the use of reinforcement learning in control engineering, the controller learns how to lead the plant through direct interaction by generating signals to the plant and evaluating their impact. The controller would therefore modify its parameters according to the relative success or failure of its actions [4]. Applying reinforcement learning to control implies that the controller learns how to stabilize the plant through a direct interaction. It applies input signals to the plant measuring the output to evaluate overall impact of the control signal. Hence the controller updates its parameters by evaluating the relative success of its actions [4].

This work presents a controller based on radial neural networks which is able to perform parameter adaptation as a response to changes on the system´s dynamics by using a simple reinforcement learning rule based on a reference model of the plant response. The approach has been tested on the steering ship control.

This work is organized as follows: section 2 present a brief description of the neural network theory, in particular to radial base networks. Section 3 describes main features on reinforcement learning while section 4 shows the ship steering problem which is used in demonstrations. Section 5 explains the reinforcement signal used in updating the parameters while Section 6 describes the radial-based neuro-controller architecture which lies on the foundations of this work. Section 7 presents the adaptation law applied to the fixed controller while Section 8 discusses some conclusions and future work.

2 Radial Base Neural Networks

Radial base neural networks (RBNNs) use functions whose output depends on the distance between the input and a value w considered as reference. Figure 1 shows a representation of such arrangement. RBNNs differ from other feedforward networks on the fact that the activation function receives as input the difference ($|\text{dist}|$) between the input vector and the weights instead of its arithmetic product.

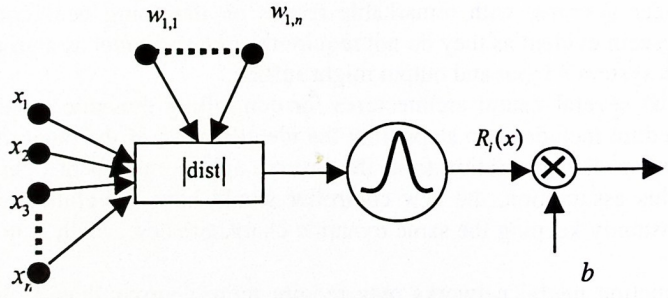

Fig. 1. Schematic representation of one radial base neuron.

The activation function normally employed is a Gaussian [6] as represented by the following equation:

$$R_i(x) = \exp\left(\frac{|x-w|}{(\sigma^i)^2}\right) \tag{1}$$

The network's output y is computed by the product $R_i(x)$ times b. The last parameter becomes important given that it can modify the sensitivity of function $R_i(x)$ in the network input. Hence, as long as the input vector **x** goes further away from the centre on $R_i(x)$ as represented by vector **w**, the value of $R_i(x)$ decreases.

3 Reinforcement Learning

In control engineering, appropriate actions required by the plant to keep the requirements are commonly unknown. In the case of non-linear systems, neural networks have shown good skills to be applied for identification by using the backpropagation learning rule. Unfortunately it also shows some drawbacks. It requires the network outputs in advance to training. Reinforcement Learning (RL) method holds a more convenient feature for the systems control problem. Instead of requiring an appropriate control action to learn from, it accepts a reward index to score its own actions, commonly known as the critic. Such element is able to define an acceptable or disappointing performance on following the required control strategy. The overall method resides in the middle of supervised and unsupervised algorithms.

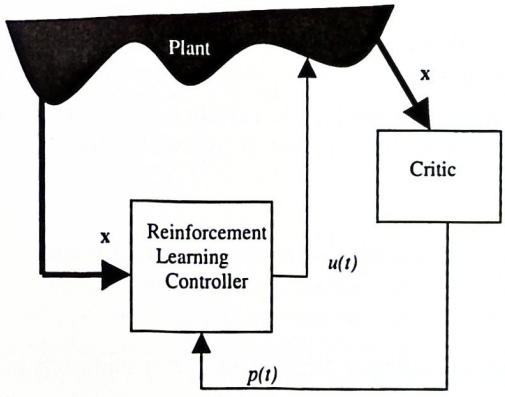

Fig. 2. Reinforcement Learning control scheme.

After receiving the system state **x** (Fig. 2), the learner receives reinforcement $p(t)$ from the environment notifying about the usefulness of its output $u(t)$. The main objec-

tive is therefore to maximize this reward signal over the time [7]. This can be achieved by trial and error training until the learner is able to discover those outputs with a maximum reward.

The main component within an RL based control scheme is therefore the critic signal and how it is processed to adjust the controller parameters. The solution proposed in this paper employs radial base neural networks and a critic signal named as J_R. In order to provide a reliable critic signal to represent how the plant must behave on real time, our implementation uses a reference model based approach.

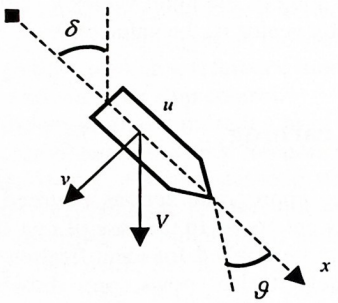

Fig. 3. The ship´s model.

4 The Plant Model

The testbed in this paper is the ship´s steering wheel control problem [8] as explained in Fig. 3. The ship moves forward in the x direction with a speed u, while ϑ refers the steering angle which in turn depends on steering wheel angle δ. Hence ϑ_r is the target direction angle as defined by the captain o desired trajectory. The objective is to develop a control strategy to assure that ϑ follows ϑ_r. The ship´s movements can be

$$\dddot{\vartheta}(t)+\left(\frac{1}{\tau_1}+\frac{1}{\tau_2}\right)\ddot{\vartheta}(t)+\left(\frac{1}{\tau_1\tau_2}\right)H(\dot{\vartheta}(t))=\frac{K}{\tau_1\tau_2}(\tau_3\dot{\delta}(t)+\delta(t)) \tag{2}$$

with $H(\dot{\vartheta}(t))$ being a non-linear function on the relationship between δ and ϑ on steady state. From a well-known test called the "spiral" [1], the function can be approximated by

$$H(\dot{\vartheta}(t))=\overline{a}\dot{\vartheta}^3(t)+\overline{b}\dot{\vartheta}(t) \tag{3}$$

with $\overline{a}$ and $\overline{b}$ being real valued and $\overline{a}$ always positive. This paper considers $\overline{a}=\overline{b}=1$ while δ is limited to ± 80 degrees as in [8]. The value of K and constants τ_i depends on the ship's speed u.

5 Reinforcement Signal based on the Reference Model

Commonly a reference model is used to score the desired performance on closed-loop. A fixed trajectory is generated from the reference model as to define how the plant must behave on each time instant. In such model, all performance indexes must therefore be clearly defined. If the reference model holds a very strict behaviour model in terms of performance, then the controller would never reach a satisfactory adaptation to it.

In general, the reference model may be continuous or discrete, linear or non-linear, time invariant or not. In this paper, the reference model which represents a correct response in time is a continuous expression as follows:

$$G(s)=\frac{1}{s+1}=\frac{Y_m(s)}{R(s)} \tag{4}$$

The reference model is discretized using as sample time T=0.1 seconds and its bilinear transform is defined as follows:

$$y_m(kT)=\frac{19}{21}y_m(kT-1)+\frac{1}{21}r(kT)+\frac{1}{21}r(kT-1) \tag{5}$$

With $r(kT)$ is the reference signal. The reinforcement signal employed in this work by the parameter update rule is given by:

$$J_R=y_m(kT)-y(kT) \tag{6}$$

with $y(kT)$ being the plant's output. It is important to provide one more feature in case the plant's output approaches the desired behaviour. It due case the reinforcement learning should be zero, i.e. no parameter adjustment is done. In real-time operation, if small values on the reinforcement signal results on several changes on the controller parameter set, instability may occur as a result of unnecessary storage of such updating orders. In order to avoid such problem in this work, we employed a threshold function that allows changes if only the reinforcement signal may be consider as admissible. Such function is defined as follows (considering α =0.005).

$$J_R = \begin{cases} J_R \text{ if } |J_R| \geq \alpha \\ 0 \quad \text{if } |J_R| < \alpha \end{cases} \tag{7}$$

6 The Neurocontroller

This section describes the most important features in the neurocontroller with respect to its own ability to modify its behaviour as a response to changes on the plant's dynamics. The radial base neural network controller follows the work of Passino in [5].

The controller seeks to regulate the ship's direction by using 2 inputs: steering error and its derivative. The network architecture is shown in Fig. 4.

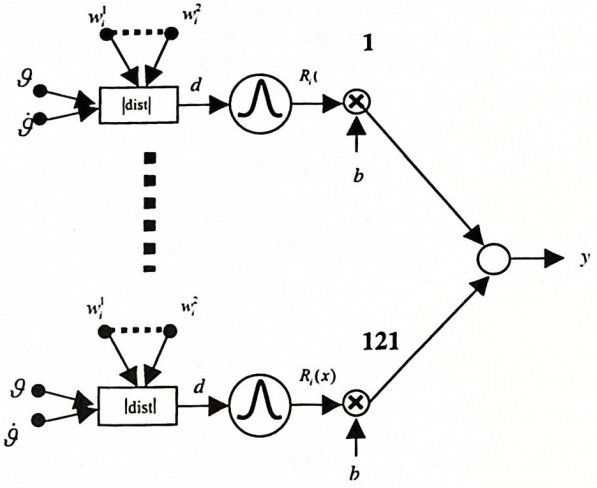

Fig. 4. The neural network architecture used as controller.

The network has 121 radial base neurons distributed among all input span for ϑ $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$ (also known as **e** in this work), and for $\dot{\vartheta}$ (-0.01, 0.01) (also named as **c**). The radial base function parameters (Gaussians) are defined by the variants over the ϑ axis in both ways yielding

$$\sigma_\vartheta = 0.7\frac{\pi}{\sqrt{n_R}} \tag{8}$$

As for $\dot{\vartheta}$

$$\sigma_{\vartheta} = 0.7\frac{0.02}{\sqrt{n_R}} \tag{9}$$

In the expression above, n_R represents the linear neuron distribution. In this work such value is 11 with a total receptive field number of n_R^2=121. Using this data set, the overall input space is fully covered while allowing a smooth transition between neurons.

The values on vector $\mathbf{b} = [b_1 \quad b_2 \quad \cdots \quad b_{121}]$ are computed following the Passino's method on [5]. The training data set is obtained from simulation on the ship's dynamical model. Under this assumptions, the system was able to control the non-linear ship model (section 4), considering a speed of u=5 m/s. However, if a change on the plant's dynamics occurs, such as varying the ship's speed then the control rule is lost.

7 Reinforcement Learning based Controller Adaptation

There are several options to adjust the network weights using the reinforcement signal. This paper considers the update of vector $\mathbf{b} = [b_1 \quad b_2 \quad \cdots \quad b_{121}]$ which multiplies all 121 functions on the radial base (receptive fields).

The adaptation law on these parameters is defined by the following equation:

$$b_i(kT) = b_i(kT-1) + J_R(kT)R_i(\vartheta, \dot{\vartheta}) \tag{10}$$

The parameter b on neuron i is computed considering its previous value plus the result of the product between the reinforcement signal J_R and the radial base value R_i which triggered such neuron.

In order to test the performance of the adaptation law and the quality on the non-linear control law applied to the ship, the dynamics model is changed from u=5 m/s to 10 m/s. The fixed radial base neurocontroller (explained in section 6) is taken as reference and the vector **b** is changed according to the adaptation law presented in (10). The results are shown in Fig. 5.

It can be seen that the adaptation time is very small allowing testing the plant against to one dynamics change. In order to test the robustness on the controller adaptation, the effect of wind perturbations on the ship are considered. Assuming the wind is flowing on time intervals (as normally happens in the sea), the overall effect may be simulated by adding a sinusoidal signal to the steer angle (input to the plant) as follows:

$$2\left(\frac{\pi}{180}\right)\sin(2\pi(0.001)*t) \tag{11}$$

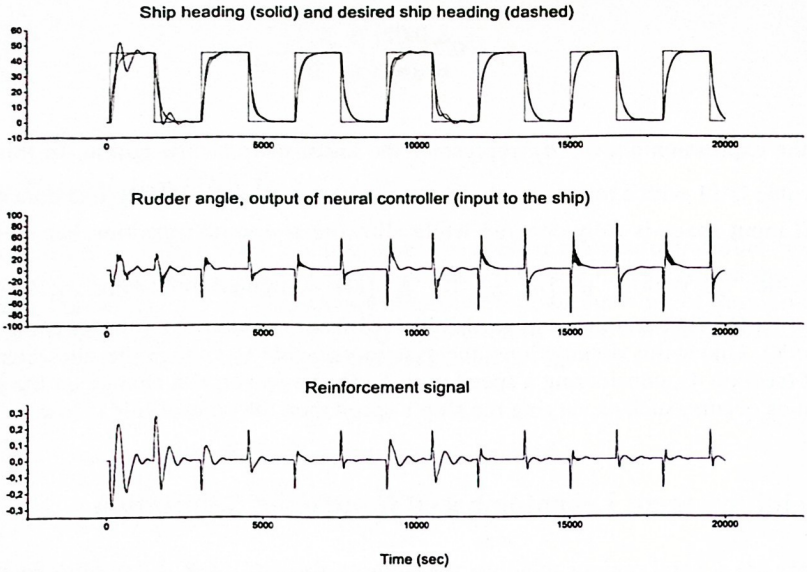

Fig. 5. Controller response using the adaptative controller and changing the dynamics on the model to u=10 m/s.

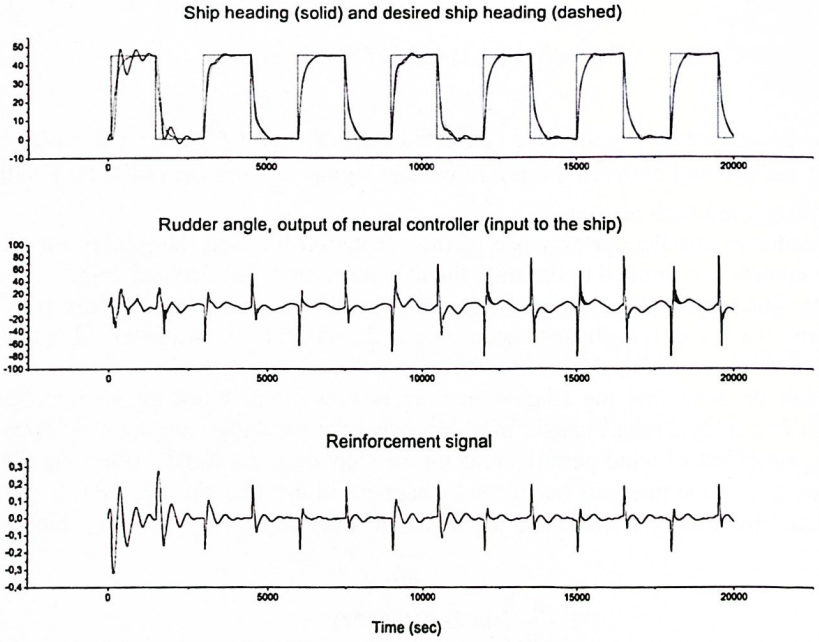

Fig. 6. Controller´s response one wind perturbation as defined in Eq. 11.

Figure 6 shows the controller's response to the perturbation. It can be seen how the controller is able to perform the required modifications to dismay the perturbation's effect which normally tend to generate instability. It can also be observed within the steady state analysis, how the wind effect can be modelled as a sinusoidal disturbance on the steer angle. Another important feature to consider in the controller adaptation comes from the fact of adding noise to the plant's output despite still keeping control of the plant. This test adds uniformly distributed and random noise to the output as shown in Fig. 7.

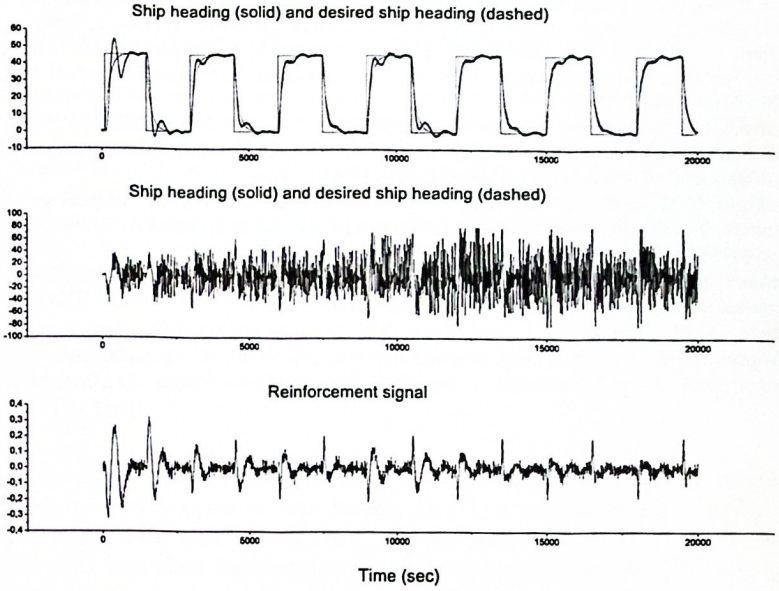

Fig. 7. Controller's response considering noise added to the output.

7 Conclusions

This paper presents a neural network based controller and a reinforcement learning update algorithm. The overall algorithm is able to exert control over a non-linear problem despite applying changes on the plant's dynamics. The reinforcement signal is computed from measurements on the plant's performance and its comparison to a simple reference model.

Despite it appropriately controls the plant, the neurocontroller initially shows some overshooting as a result of the parameter adaptation (Fig. 5). It takes about 1.2 seconds to define that no more changes are required in the controller's structure.

The updating algorithm based on reinforcement has shown an acceptable robustness, as it keeps the ship steering control despite perturbations shown by Fig. 6 or the noisy signals presented by Fig. 7.

References

1. Miller, W.T., Sutton, R.S. and Werbos, P.J., editors. Neural Networks for Control. The MIT Press, Cambridge, MA, 1991.
2. Narendra, K.S. and Parthasarathy, K., Identification and Control of Dynamical Systems using Neural Networks. IEEE Transactions on Neural Networks 1(1):4-27, 1990.
3. Chen, S., Billings, S.A., and Grant, P.M., Recursive hybrid algorithm for nonlinear system identification using radial basis function networks. International Journal of Control, 55(5):1051-1070, 1992.
4. Farrell, J.A., and Baker, W., Learning Control Systems. In P.J. Antsaklis and K.M Passino, editors, An Introduction to intelligent and Autonomous Control systems, pages 273-262. Kluwer academic publishers, Norwell, MA, 1993.
5. Passino, K.M., and Antsaklis, P.J., A System and Control Theoretic Perspective on Artificial Intelligence Planning Systems. International Journal of Applied Artificial Intelligence, 3:1-32,1989.
6. Sanner, R. M and Slotine, J. J. E. Gaussian Networks for Direct Adaptive Control. IEEE Transactions on Neural Networks, 3(6), 837-863.
7. Barto, A. G., Bradtke, S. J., and Singh, S.P. Learning to Act using Real-Time Dynamic Programming. Department of Computer Science, University of Massachusetts.
8. Amström K. J. and Wittenmark. Adaptive Control. Addison-Wesley, Reading, MA. 1995.