ROBUST ADAPTIVE NEUROCONTROL OF A CERTAIN CLASS OF CONTINUOUS-TIME PROCESSES

Jean-Michel Renders and Marco Saerens

Abstract — We introduce two different weight adaptation laws, based on Lyapunov stability theory, for the adaptive control of a certain class of MIMO continuous-time processes. The adaptation strategy is indirect in that it uses a sensitivity model of the plant (in our case, the sensitivity derivatives are estimated with a neural network that identifies the plant) in order to implement the adaptation law. The resulting adjustment rules are surprisingly simple in comparison with the gradient-based algorithm (also similar to the dynamic back-propagation, Williams & Zipser, 1989, introduced by Narendra & Parthasarathy (1990; 1992). However, the stability statements, based on a first-order expansion, are only valid if the initial weight values are not too far from their optimal values that allow perfect model matching (local stability). We therefore propose to initialize the weights with values that solve the linear problem. After the treatment of the ideal case, we derive a robust adjustment law that takes noise and modelling errors into account. Indeed, due to noise, disturbances and modelling errors, the perfect model matching condition cannot be verified, and it is not possible to ensure that the error converges to zero, but only that it remains bounded. Moreover, it has been shown that in certain cases, the simple adaptive scheme designed for the ideal case can be driven unstable in the presence of bounded disturbances and unmodelled dynamics. Finally, in practical situations, the sensitivity model needed for the adaptation strategy is always imperfect. We therefore use a modified adjustment law known as the "ci-modification" scheme, originally introduced by Narendra & Annaswamy (1987) for the adaptive control of linear systems, that does not require information about the bounds of the perturbation effects. This adaptive control scheme is proved to be robust in that all the signals in the adaptive loop are bounded for bounded disturbances.

I. INTRODUCTION

There has been a recent resurgence of interest for the old cybernetic project to exploit neural networks for control applications (Miller et al., 1990; Narendra and Parthasarathy, 1990; Hunt et al., 1992; White & Sofge, 1992; Rao Vemuri, 1992; Gupta & Sinha, 1994). Several approaches are possible when using neural nets for the control of plants (Psaltis, Sideris & Yamamura, 1987; Barto, 1990; Hunt et al., 1992). One possible strategy that fits in perfectly with the classical adaptive control attitude consists in using the difference between the actual output of the plant and the desired output in order to adapt the weights of connections. However, in order to implement this adjustment law, some prior knowledge on the way the plant reacts to slight input modifications, i.e. the Jacobian matrix of the plant, is needed. One possible, while inelegant, solution consists in approximating the partial derivatives by plotting the plant reactions to slight control modifications at the operating points (Psaltis, Sideris & Yamamura, 1987, 1988).

Another, more sophisticated, technique has been proposed (Jordan, 1989; Jordan & Rumelhart, 1991; Nguyen & Widrow, 1989, 1990), which incorporates the default prior knowledge directly in a network and links the neural controller to a neural model of the plant — a neural network that identifies the process. As pointed out by Narendra & Parthasarathy (Narendra & Parthasarathy, 1990), it corresponds to the classical indirect control approach: the parameters of the plant are estimated at any instant, and the parameters of the controller (in this case, the weights of the neural controller) are adjusted assuming that the estimated parameters of the plant represent the true values. However it needs, in compensation, either a preceding learning stage (the identification of the plant) or a "neurally expressed" prior knowledge of the dynamics of the plant. Narendra & Parthasarathy (1990, 1992) use a similar indirect technique to show that identification and adaptive control schemes are practically feasible.

Indeed, Narendra & Parthasarathy (1990) showed by simulations that a neural network can be used effectively for the identification and control of nonlinear dynamical processes. The results are based on the universal approximation properties of three-layer neural networks. As a matter of fact, it has been shown (Hornik, Stinchcombe & White, 1989 and 1990; Stinchcombe & White, 1989; for a review, see White, 1992) that a three-layer network (one hidden layer) with an arbitrarily large number of nodes in the hidden layer can approximate any continuous function \( f : \mathbb{R}^n \to \mathbb{R}^m \) over a compact subset of \( \mathbb{R}^n \). Other recent results indicate that, in some cases, two hidden layers are required for the approximation of inverses of continuous functions (Sontag, 1992). These are remarkable properties that make neural nets suitable for modelling nonlinear processes and designing nonlinear controllers. However, this is not enough. Indeed, as mentioned before, in the indirect approach, most of the control algorithms use a differentiable model of the process in order to compute adaptation laws. For instance, gradient descent algorithms involve the Jacobian matrix of the process that must be deduced from the model. In some cases, the model is itself a neural net which, for this reason, must also approximate the derivative. So, authors developed conditions in which a net will also approximate the derivative of the mapping (Cardaliaguet & Euvrard, 1992; Gallant & White, 1992).

These facts provide some motivations to study neural networks in the framework of nonlinear systems theory (see Narendra & Parthasarathy, 1990, for more details). However, the choice of identification and controller models for nonlinear plants is a formidable problem and successful identification and control has to depend upon

Corresponding author is Marco Saerens, Laboratoire IRIDIA – cp 194/6, Université Libre de Bruxelles, 50 av F. Roosevelt, 1050 Bruxelles, BELGIUM.
several strong assumptions regarding the input-output behavior of the plant. For instance, complete controllability and observability must be assumed (see Narendra & Parthasarathy, 1990, for a discussion).

Recently, researchers tried to design adaptation laws that ensure the stability of the closed-loop process (see Kawato, 1990; Renders, 1991; Gomi & Kawato, 1993 in the context of robotics applications; see Parthasarathy & Narendra, 1991; Sanner & Slotine, 1992; Narendra, 1992; Tzinkel-Hancock & Fallside, 1992; Renders, 1993; Jin, Pipe & Winfield, 1993; Chen & Liu, 1993; Jin, Nikiforuk & Gupta, 1993; Saerens, Renders & Bersini, 1994; Rovithakis & Christodoulou, 1994 in the context of adaptive control with neural nets), or to ensure the viability of the regulation law (Seube, 1990; Seube & Macia, 1991). Stability of the overall system is certainly a prime requirement that must be met before thinking of real-world applications. Our aim in this paper is first (section II) to present a weight adjustment law, based on Lyapunov stability theory, for the adaptive control of a certain class of MIMO continuous-time processes (Renders & Saerens, 1994). This extends our previous work (Renders, 1993; Saerens, Renders & Bersini, 1994), where only single-input single-output (SISO) systems were considered; moreover, we introduce an integral term in order to avoid the steady-state errors. As previously mentioned, the adjustment strategy is indirect in that it uses a sensitivity model of the plant (some sensitivity derivatives have to be known; in our case, we estimate them with a neural network that identifies the plant) in order to implement the adaptation law. The resulting adaptation law is surprisingly simple in comparison with the gradient-based algorithm (also known as the dynamic backpropagation algorithm used to train recurrent neural nets, Williams & Zipser, 1989) introduced by Narendra & Parthasarathy (1990; 1992). The stability statement, based on a first-order expansion, is, however, only valid if the initial weight values are not too far from their optimal values that allow perfect model matching (local stability). Therefore, following Kawato (1990), we propose to initialize the weights with values that solve the linear problem. Notice that the proposed adaptation law can be applied to more general processes than the algorithms of Parthasarathy & Narendra (1991), Narendra (1992), Sanner & Slotine (1992), Tzinkel-Hancock & Fallside (1992), Chen & Liu (1993), Jin, Nikiforuk & Gupta (1993), Johansen (1994), and Rovithakis & Christodoulou (1994), where affine processes were considered.

In the second part of the paper (section III), we propose a robust adjustment law that takes account of noise, disturbances, and modelling errors. Indeed, due to noise, disturbances, and modelling errors, the perfect model matching condition cannot be verified, and it is no longer possible to ensure that the error converges to zero, but only that it remains bounded. Moreover, it has been shown that in certain cases, the simple adaptive scheme proposed in section II, designed for the ideal case, can be driven unstable in the presence of bounded disturbances and unmodelled dynamics (see for instance Rohrs, Valavanii, Athans & Stein, 1985). The underlying instability mechanisms are basically the parameter drift phenomenon or the excitation of unmodelled high-frequency dynamics by the reference signal or by the high gain feedback generated by the adaptive control law (Ioannou & Datta, 1991). Finally, in practical situations, a perfect sensitivity model of the plant is not available, which results in disturbances in the adaptation law.

In order to counteract these instability phenomena, one possible choice is to adopt a "dead-zone" strategy (see Ortega & Tang for a review). However, the dead-zone approach requires some information about the bounds of the perturbations effects, which in general are not available. We therefore use a modified adjustment law known as the "e1-modification" scheme, originally introduced by Narendra & Annaswamy (1987) for the adaptive control of linear systems, that does not require this additional information. This adaptive control scheme is proved to be robust in that all the signals in the adaptive loop are bounded for bounded disturbances. Moreover, it has been shown that in the ideal case, that is when the disturbances are removed, the origin of the error equations is exponentially stable when the reference input is persistently exciting with a sufficiently large amplitude (Narendra & Annaswamy, 1987).

In the third part (section IV) of the paper, we propose an architecture that allows us to initialize the weights of the net with a linearized controller. This is important, since the local stability results are only valid if we are in the neighborhood of the optimal weight values.

II. ADAPTIVE NEUROCONTROL IN THE IDEAL CASE

II.1 Introduction

If \( y_i^{(d)} \) is the \( d \) order time derivative of the signal \( y(t) \), we assume the three processes can be described by

\[
y_i^{(d)} = f_i(x(t), u(t)); \quad i = 1, ..., n
\]

(1a)

where the state \( x(t) \) is supposed to be measurable and can be obtained from the \( n \) outputs \( y_i(t) \) and their time derivatives up to order \( d_{i-1} \). \( u(t) = [u_1, u_2, ..., u_n]^T \) and \( y(t) = [y_1, y_2, ..., y_n]^T \) are respectively the inputs and the outputs of the plant at time \( t \). \( d_i \) is the relative degree and, by definition of \( d_i \), \( \partial f_i / \partial u_j \neq 0 \) for at least one \( u_j \in \{u_1, u_2, ..., u_n\} \). We also require that the relationship between the \( n \) inputs \( u_i(t) \) and the \( n \) differentiated outputs \( y_i^{(d)} \) given by equation (1a) is a global diffeomorphism on the operating region. This means that, whatever the value of \( x(t) \) in the operating region, the set of equations \( y_i^{(d)} = f_i(x(t), u(t)) \) \( (i = 1, ..., n) \) can be solved uniquely in terms of \( u(t) \), for every set of values \( y_i^{(d)} \) \( (i = 1, ..., n) \) in the operating region. This implies that the Jacobian matrix is non-singular (of rank \( n \)) in the operating region: \( \det(\partial f_i / \partial u_i(t)) \neq 0 \). If we define the Jacobian matrix as \( [\xi_i(t)]_{ij} = \partial f_i / \partial u_j \), equation (1a) can be rewritten as

\[
\xi_i(t) y_i(t) = F(x(t), u(t))
\]

(1b)

where \( F = [f_1, f_2, ..., f_n]^T \). Note that (1b) involves a mixture of time-domain and frequency-domain notations (\( s \) is the Laplace variable); such hybrid notations are
common in the adaptive control literature and their interpretation is obvious.

We now consider that the output of a reference filter \( y_i^{(d)} \) (which is supposed to be both computable and achievable) is obtained by filtering a reference signal \( r(t) \), taking account of the error signal (see Figure 1):

\[
\begin{align*}
y_i^{(d)} - r_i^{(d)} &= \lambda_i^{d-1} (r_i^{(d-1)} - y_i^{(d-1)}) + \\
&+ \lambda_i^0 \int (r_i - y_i) \, dt \\
&= \lambda_i^{d-1} e_i^{(d-1)} + \\
&+ \lambda_i^0 \int e_i \, dt; \quad i = 1, \ldots, n \tag{2a}
\end{align*}
\]

where \( r(t) = [r_1, r_2, \ldots, r_n]^T \) is the reference signal, \( r_i^{(j)} \) the time derivative of order \( j \), and

\[
e_i^{(j)} = (r_i^{(j)} - y_i^{(j)}); \quad i = 1, \ldots, n \tag{3}
\]

In a more compact way, we have

\[
\xi(s) (\bar{y}(t) - r(t)) = H(s) \xi
\]

where \( H(s) \) is a diagonal polynomial matrix and \( \bar{y}(t) = [\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_n]^T \).

We observe that when perfect tracking is achieved, that is, by taking \( y_i^{(j)} \) instead of \( y_i^{(d)} \) in (2a), the equation allows us to design the transient of the initial error. Indeed, we have

\[
0 = e_i^{(d)} + \lambda_i^{d-1} e_i^{(d-1)} + \\
+ \lambda_i^0 \int e_i \, dt; \quad i = 1, \ldots, n \tag{4}
\]

and the coefficients \( \lambda_i^j \) will be chosen in order to obtain a desired transient behaviour.

If we make the assumption that the system is asymptotically minimum phase, then, there exists a stable control law, which allows perfect tracking of the signal \( \xi(s)\bar{y}(t) \):

\[
\mathbf{u} = \mathbb{Z}[\xi(s)\bar{y}(t), \mathbf{x}(t)] = \mathbb{Z}[\mathbf{p}(t), \mathbf{x}(t)] \tag{5}
\]

where we defined \( \mathbf{p}(t) = \xi(s)\bar{y}(t) \). We have

\[
\xi(s)\bar{y}(t) = \xi(s)\bar{y}(t) \quad \text{if} \quad \mathbf{u} = \mathbb{Z}[\mathbf{p}(t), \mathbf{x}(t)] \tag{6}
\]

We adopt a control law generated by a neural network:

\[
\mathbf{u}(t) = \mathbf{N}[\mathbf{p}(t), \mathbf{x}(t); \mathbf{w}(t)] \tag{7}
\]

where \( \mathbf{w} \) is the connection weight vector of the network. The network has inputs \( \mathbf{p}(t), \mathbf{x}(t) \) and output \( \mathbf{u}(t) \). We assume that the neural controller \( \mathbf{u}(t) = \mathbf{N}[\mathbf{p}(t), \mathbf{x}(t); \mathbf{w}(t)] \) can approximate the control law (5) to any degree of accuracy in the operating region. This is the standard perfect model matching condition.

### II.2 Adaptation Law

In Renders & Saerens (1994), we proved the stability of a simple adaptation law of the weight values, based on a first-order expansion around the perfectly tuned values. The statement is given without details, because of the lack of space. Further details can be found in Renders & Saerens (1994).

Let us define \( \delta(t) = [\delta_1, \delta_2, \ldots, \delta_n]^T \) and \( e_i = (r_i - y_i) \), where

\[
\begin{align*}
\delta_i &= e_i^{(d-1)} + \lambda_i^1 e_i^{(d-2)} + \\
&+ \lambda_i^0 \int e_i \, dt + \alpha_i^1 e_i^{(1)} + \alpha_i^d e_i + \alpha_i^{d+1} \int e_i \, dt \quad (8)
\end{align*}
\]

which is easily computable because directly related to the state. The coefficients \( \alpha_i \) must be chosen in such a way that the corresponding transfer function is Hurwitz; these coefficients are directly related to the \( \lambda_i \) (2a). In particular, this implies that if \( \delta_i \to 0 \), we have \( e_i \to 0 \).

**Convergence statement.** If \( \mathbf{v}(t) = \mathcal{F}[\mathbf{x}(t); \mathbf{u}(t)] \)

\[
\frac{\partial}{\partial \mathbf{w}} \mathcal{F}[\mathbf{p}(t), \mathbf{x}(t); \mathbf{w}(t)] \text{is globally bounded, the adaptation}
\]

\[
\dot{\mathbf{w}} = \eta \mathbf{v}^T \delta(t) \quad (\eta > 0)
\]

will drive the auxiliary error \( \delta(t) \) asymptotically to zero: \( \delta(t) \to 0 \); that is, 

\[
\delta_i = e_i^{(d-1)} + \lambda_i^1 e_i^{(d-2)} + \\
+ \lambda_i^0 \int e_i \, dt + \alpha_i^1 e_i^{(1)} + \alpha_i^d e_i + \alpha_i^{d+1} \int e_i \, dt \to 0, \text{and therefore} \ e_i \to 0.
\]

Notice that the term \( \mathcal{F}[\mathbf{p}(t), \mathbf{x}(t); \mathbf{w}(t)] \) appearing in (9) can be computed thanks to backpropagation algorithm, while \( \mathcal{F}[\mathbf{x}(t); \mathbf{u}(t)]/\partial \mathbf{w} \) is a matrix of sensitivity derivatives (the Jacobian matrix of the process defined by (1)). As proposed by Jordan (1989) and Narendra & Parthasarathy (1990), this Jacobian matrix can be estimated from a model of the process; possibly in the form of a neural identifier. As mentioned above, the adjustment strategy is indirect in that it uses a sensitivity model of the plant (in our case, we use a neural network that identifies the plant) in order to implement the adaptation law.

For the training of the neural identifier, in order to avoid the computation of the derivatives \( y_i^{(d)} \), we can apply the same trick as for the training of the neural controller, or adapt the linear adaptive identification algorithms (Narendra & Annaswamy, 1989; Rimon & Narendra, 1992).

Various simulations of a similar adaptation law have been carried out for the SISO case (Renders, 1993; Renders, Bersini & Saerens, 1993; Saerens, Renders & Bersini, 1994; Saerens et al., 1992).

### III. ADAPTIVE NEUROCONTROL IN THE PRESENCE OF DISTURBANCES

#### III.1 Introduction

Due to noise, disturbances, and modelling errors, the perfect model matching condition cannot be verified, and it will not be possible to ensure that \( e_i \to 0 \) asymptotically, but only that it will remain bounded. Moreover, in
practice, a perfect model of the process will certainly not be available, so that \(v(t)\) will also be subject to disturbances.

In certain cases, the simple adaptive scheme designed for the ideal case in section II can be driven unstable in the presence of bounded disturbances and unmodelled dynamics. The underlying instability mechanisms are basically the parameter drift phenomenon or the excitation of unmodelled high-frequency dynamics by the reference signal or by the high gain feedback generated by the adaptive control law (Ioannou & Datta, 1991).

We proposed (Renders & Saerens, 1994) to use a modified adjustment law known as the "\(\varepsilon_1\)-modification" scheme, originally introduced by Narendra & Annaswamy (1987) for the adaptive control of linear systems, that does not require any additional information.

**III.2 A modified adaptation law**

Let us first give the stability statement:

\[
\text{Stability statement. If } v(t) = \frac{\partial F}{\partial \varepsilon}[x(t); u(t)] \text{ and } \frac{\partial u(t)}{\partial \varepsilon} \text{ is globally bounded, the adaptation rule}
\]

\[
\dot{\varepsilon} = \eta v \varepsilon \delta(t) - \eta \gamma \| \delta \| w, \quad \text{with } \eta, \gamma > 0 \tag{10}
\]

will ensure that the weight vector \(w\) as well as the auxiliary error \(\delta(t)\) remain bounded, and therefore each \(\varepsilon_i\) remains bounded.

By defining robustness as boundedness of the signals in the adaptive loop for bounded disturbances, we have shown (Renders & Saerens, 1994) that the adaptive system with adjustment law (10) is robust. We considered two different cases, depending on the kind of disturbance considered: (i) bounded disturbances due to noise and neural network modelling errors for the neural controller, and (ii) bounded disturbances due to imperfect modelling of the process (which results in errors in the estimation of the sensitivity derivatives).

On the other hand, Narendra & Annaswamy (1987, 1989) showed that in the ideal case, that is, in the absence of disturbances and modelling errors, the error equations are exponentially stable when the reference input is persistently exciting with a sufficiently large amplitude.

**IV. WEIGHTS INITIALIZATION PROCEDURE**

Now, the result is valid only when the weights are not too far from their optimal values (local stability). We do not expect that this hypothesis could be easily removed because there is always a risk of falling into a local minimum when dealing with nonlinear systems such as, i.e., feedforward neural networks. This suggests a preliminary initialization of the weight values in order to be in the basin of attraction of the optimal values. One way to initialize the weights is to consider direct linear connections from the inputs to the outputs of the net. The weights of these connections can be trained by linear techniques, or initialized to values that solve the linear problem (Kawato, 1990; Gomi & Kawato, 1993; Scott, Shavitk & Ray, 1992). The other weight values should be initialized near zero in order to have negligible effect on the control law. Tuning of these weights can then be started, while maintaining the direct weights from input to output constant. In other words, we have

\[
u(t) = u_1(t) + u_2(t)
\]

with \(u_1(t)\) being the result of the linear transform from the input to the output, and \(u_2(t)\) being the output of the neural net, without the direct linear connections.

**REFERENCES**


---

**Figure 1. Overall architecture used for adaptive neurocontrol of a certain class of unknown continuous-time nonlinear processes.**

\[ x(t) \]

\[ r(t) \]

\[ p(t) \]

\[ \xi \]

\[ \text{Generalized PID} \]

\[ \text{NN} \]

\[ \text{Linear Controller} \]

\[ u(t) \]

\[ \text{PLANT} \]

\[ x(t) \]