IMPROVED ROBUST CONTROL OF NONLINEAR STOCHASTIC SYSTEMS USING UNCERTAIN MODELS

Randa Herzallah and David Lowe
email: herzarom@aston.ac.uk

NCRG, Aston University, UK

Abstract: We introduce a technique for quantifying and then exploiting uncertainty in nonlinear stochastic control systems. The approach is suboptimal though robust and relies upon the approximation of the forward and inverse plant models by neural networks, which also estimate the intrinsic uncertainty. Sampling from the resulting Gaussian distributions of the inversion based neurocontroller allows us to introduce a control law which is demonstrably more robust than traditional adaptive controllers.

Keywords: Uncertainty. Neural networks. Stochastic systems. Error bar. Distribution modelling.

1. INTRODUCTION

In modern control theory nonlinear stochastic optimal control is of great importance, due to its immediate application to real world problems, where disturbances play an important part in the performance of control processes. In such situations, once the objective functional is defined we would ideally seek a dynamic programming solution. This however, is practically unfeasible, not least because of the unbounded search space which we need to try and maintain possible solution trajectories. The method of approximation we choose is to try and model the conditional uncertainty of the control signal by modelling its statistical properties, expressed in terms of a conditional distribution function, which would be generated by the real stochastic system under ideal circumstances. Since we are interested in nonlinear stochastic systems, a nonlinear controller is required to satisfactorily control such plants. The nonlinear controller can be viewed as a nonlinear approximation problem.

Recently, neural network models have evolved into favourite candidates in the field of nonlinear system identification and control, due to their ability to approximate multi-variable nonlinear mappings. Besides having nonlinear features, dynamic systems may have noise events affecting their inputs and outputs, and usually they are time-variant. Because artificial neural networks can be adapted on line [1–3], usually they are capable of good performances in such situations. However for most real control problems where disturbances play an important part and where a relatively big sampling interval is used, the predicted output of the neural network is inherently uncertain.

Many researchers have considered the use of the uncertainty measure to build a more robust controller. Some of the recent works on the use of uncertainty have been introduced in [4] where a systematic procedure that accounts for the structured uncertainty when a neural network model is integrated in an approximate feedback linearisation control scheme has been developed. The use of an adaptive critic controller when there is input uncertainty has been discussed in [5]. The application of recently developed minimal resource allocating network $(MRAN)$ in a robust manner under faulty conditions has been demonstrated in [6]. A robust adaptive nonlinear control method for controlling a class of nonlinear systems in the present of both unknown nonlinearities and unmodelled dynamics has been illustrated in [7].

However neural networks now have the ability to model general distributions rather than just producing point estimates, and in particular can produce an estimate of the uncertainty involved in its own predictions [8–10]. But none of the recent works have considered the possibility of using the neural network's own estimate for error bars. In this paper we address the use of this extra knowledge to approximate the conditional distribution of the control signal for stochastic systems of the form

$$\hat{y}(t) = f(y(t-1), ..., y(t-n), u(t-d) \\ , \ ..., u(t-m), \bar{v}(t)) \quad (1)$$

where $y(t)$ is the measured plant output, $u(t)$is the measured plant input, $\bar{v}(t)$ is the noise affecting the plant output, n is maximum delay of the output, m is the maximum delay of the input, d is the relative degree of the plant.

After modelling the conditional distribution of the control signal we search for the optimal control law from the estimated control signal distribution, almost in the same way as dynamic programming, where the optimal control law is chosen such that to minimise a certain utility function

$$u_{opt} = \underset{u \in U}{Min} \underset{\bar{v}}{E} J(U) \quad (2)$$

but with the advantage of avoiding the computation requirements for the dynamic programming. Furthermore, a stability analysis for the updating rule of the control signal will be studied.

This paper begins with presenting the principle of system model and error bar estimation. Next, we develop a nonlinear controller architecture based on approximate dynamic inversion and the use of error bar knowledge. After that we present the stability analysis for the proposed control algorithm. This development is then employed to control a nonlinear uncertain simulated system.

2. ADAPTIVE INVERSE CONTROL

The classical inverse adaptive control technique is shown in figure 1. Basically the neural network is learning to recreate the input that created the current output of the plant. So in this case the desired response for the adaptive controller is the plant input. The inverse controller contains adjustable parameters that control its impulse response. An adaptive algorithm is usually used to automatically adjust the controller parameters to minimise some function of the error (usually mean square error, other error functions can also be used). The error is defined as the difference between the input of the plant and the actual output of the controller. Many such algorithms are described in the reports and textbooks by Narendra and Parthasarathy [3] and by White and Sofge [1].

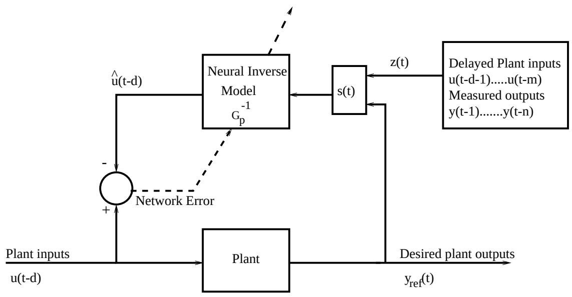

Fig. 1. Training of an inverse controller.

3. DISTRIBUTION MODELLING OF CONTROL SIGNAL

In classical inverse control the challenge is to build a neural network that will take past values of the input and output of the plant $z(t) = [y(t-1), ..., y(t-n), u(t-2), ..., u(t-m)]$ and the desired output value $y_r(t)$ as an input, and outputs the control signals $u(t-d)$ (assuming d relative degree), which will move the plant output to the desired value. For dynamical systems it is reasonable to assume that the output of the system $y(t)$ is a function f of its input $u(t-d)$ and the delayed vector $z(t)$. Furthermore in the case of a one-to-one mapping, and only in this case, the inverse of the function denoted by f^{-1} can be used to tell us how to choose the control signal value $u(t-d)$ to give the desired output value $y_r(t)$. In this case a feedforward neural network trained using the sum of the square error function (between the input of the system and the actual output of the controller) can perform well. In this work our basic goal is to model the conditional uncertainty of the control signal, by modelling its statistical properties, expressed in terms of the conditional distribution function $p(u(t-d)|s(t))$. Here $s(t) = [z(t), y_r(t)]$ is the input vector to the neural inverse model. Different methods for estimating the uncertainty around the predicted output of a neural network have been presented in [8–10]. In this work the predictive error bar method will be used. It is reported in [9]. This approach is based on the important result that for a network trained on minimum square error the optimum network output approximates the conditional mean of the target data, or $f_{opt}^{-1}(s(t)) =< u(t-d)|s(t) >$, and that the local variance of the target data can be estimated as $\|u(t-d) - f_{opt}^{-1}(s(t))\|^2$. If this variance is used as a target value for another neural network, then the optimum output of this second network is again the conditional mean of that variance. As reported in [9], in the implementation of predictive error bars two correlated neural neural networks are used. Each network shares the same input and hidden nodes, but has different final layer links which are estimated to give the approximated conditional mean of the target data in the first network, and the approximated conditional mean of the variance in the second network. Thus the second network predicts the noise variance of the predicted mean by the first network. This architecture is shown in figure 2. Optimisation of the weights is a two stage process: The first stage determines the weights w_1 conditioning the regression on the mapping surface. Once these weights have been determined, the network approximations to the target values are known, and hence so are the conditional error values on the training examples. In the second stage the inputs to the networks remain exactly as before, but now the target outputs of the network are the error values. This second pass determines the weights w_2 which condition the second set of output noise to the squared error values $\sigma^2(s(t))$. The distribution assumption is then Gaussian with input dependent mean given by the predicted value of the control signal from the first network, and input dependent variance given by the predicted value of the noise from the second network. However, if the inverse of the function f can not be defined uniquely direct inverse mapping f^{-1} found by using the sum of the square error function between the input of the system and the actual output of the controller can not be used to tell us how to choose the control signal $u(t-d)$ so as to reach the desired response $y_r(t)$. Therefore, assuming a Gaussian distribution can lead to a very poor representation of the control signal. In this case a more general framework for modelling conditional probability distributions is required. This general framework (is not going to be discussed in this paper), is based on the use of the mixture density network.

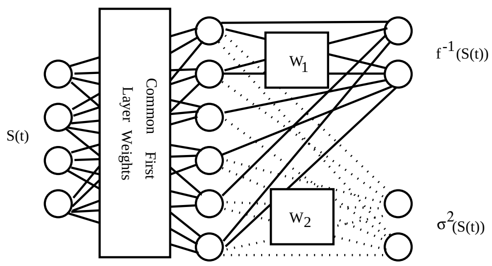

Fig. 2. The architecture of the predictive error bar network.

4. PROBLEM FORMULATION AND SOLUTION DEVELOPMENT

Dynamic programming is a powerful tool in stochastic control problems [11,12]. However, it performs poorly when the order of the system increases. The algorithm proposed here is based on incorporating the uncertainty knowledge from the neural network to avoid the computational requirements for the dynamic programming solution for stochastic control problems. We search for an algorithmic approach yielding numerical solutions to the minimisation problem. The proposed method is equivalent to sampling values from the distribution of u and using the function value alone to determine a reasonable minimisation of the objective, $J(t)$. Using the gradient information of $J(t)$, although it would be more efficient, is not exploitable here due to the random sampling nature of the algorithm and the potential stochastic nature of the plant.

4.1 *Neural Network Development for Incorporating Uncertainty*

Once properly trained, the inverse model can be used to control the plant since it can create the necessary control signals to create the desired system output. Despite the fact that neural networks have been accepted as a suitable model for capturing the behaviour of nonlinear dynamical systems, it is also accepted that such models should not be considered exact. The algorithm proposed here circumvents the dynamic programming scaling problem by using the predicted neural network error bars to limit the possible control solutions to consider. Accepting the inaccuracy of neural networks and assuming the output of the inverse control network can be approximated by a Gaussian distribution of control signals, the mean and variances can be obtained as discussed previously. Using just the mean estimate of the control is typically suboptimal in nonlinear systems. Even though the Gaussian assumption used here is an approximation, using the on-line variance estimate of the neural network determines a region around the predicted mean value where sampling can be used to obtain a better estimate of the control signal than the mean. The distribution assumption is Gaussian but the predicted mean and variance are nonlinear functions of previous states, thus allowing for good models of forward and inverse plant behaviour provided the inverse plant is a function. If this is not the case a similar approach, but using Gaussian mixtures (a mixture density network) could be employed. Based on estimates of the mean and variance of the distribution of control signal values, we can construct the following algorithm incorporating the uncertainty directly:

(1) Based on the pre-collected input-output data, an accurate model of the process is constructed and trained off line. It is assumed to be described by the following neural network model:

$$\hat{y}(t) = f(y(t-1), ..., y(t-n) \\ , \; u(t-d), ..., u(t-m)) \quad (3)$$

(2) An accurate inverse model of the plant should also be constructed, and trained off line to approximate the conditional mean of the control vector and the conditional variance. It is assumed to be described by the following neural network

$$x(t) = f^{-1}(y(t), y(t-1), ..., y(t-n) \\ , \; u(t-d-1), ..., u(t-m)) \quad (4)$$

$$\hat{u}(t-d) = x(t)w_1 \quad (5)$$

$$var_{u(t-d)} = x(t)w_2 \quad (6)$$

where
$x(t)$: is the predicted hidden variable from the neural network at each instant of time t.
w_1: is the weight of the linear layer estimated to predict the conditioned mean of the control

signal.

w_2: is the weight of the linear layer estimated to predict the variance of the predicted control signal.

(3) Define the desired response of the plant by defining a suitable reference model, which should be chosen to have the same relative degree as that of the plant.
(4) At each instant of time t the desired output is calculated from the reference model output.
(5) Bring the control network on line and at each time t estimate the appropriate control signal from the controller and the variance of that control signal. The control signal distribution is then assumed to be Gaussian and given by

$$p(u(t-d) \mid s(t)) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp(-\frac{(u(t-d) - \hat{u}(t-d))^2}{2\sigma^2}) \quad (7)$$

where

σ^2: is the variance of the control signal $var_{u(t-d)}$

$s(t) : [y_r(t), y(t-1), ..., y(t-n)$
$, u(t-d-1), ..., u(t-n)]$

(6) Generate a vector of samples from the control signal distribution. That vector of samples is considered as the admissible control values at each instant of time. The number of samples is chosen based on the value of the predicted variance of the control signal $number\, of\, samples = K \times var_{u(t-d)}$. This equation determines the number of samples based on the confidence of the controller about the predicted mean value of the control signal. So more samples are generated for larger variance.
(7) Feed these samples to the system model and calculate the output from each sample.
(8) Based on the effect of each sample on the output of the model, the most likely control value is taken, which is assumed to be the value that minimises the following cost function.

$$J(t) = \underset{u \in U}{Min} \underset{\bar{v}}{E}[(\hat{y}(t) - y_r(t))^2] \quad (8)$$

where U is a vector containing the sampled values from the control signal distribution, E is the expected value of the cost function over the random noise variable $\bar{v}$. Because we are using a neural network to model the system, and because the neural network predicts the mean value for the output of the model averaged over the noise on the data, the above function can be optimised directly.
(9) Go to step 4.

4.2 *Stability Analysis*

The development of the previous section was based on updating the control signal from it's estimated distribution. Here we use the Lyapunov method to determine stability and convergence of the output error for the proposed updating rule of the control signal. The Lyapunov function V_k is defined as

$$V_k = e_k^2 \quad (9)$$

If V_k can be shown to be positive and decreasing when updating the control signal from its distribution, then we conclude that the output error converges and the updating process is stable. By definition, V_k is positive. We now proceed to find whether V_k is decreasing or not. The variation of Lyapunov function from iteration k to iteration $k+1$ is:

$$\Delta V_k = V_{k+1} - V_k = [e_{k+1}^2 - e_k^2] \quad (10)$$

Assuming that the input, z and the desired output, y_r remain the same from iteration k to iteration $k+1$, then

$$e_{k+1} = y_{r_k} - f(z_k, u_{\text{new}}) \quad (11)$$

$$e_k = y_{r_k} - f(z_k, u_{\text{old}}) \quad (12)$$

where u_{old} is initially set to be equal to the predicted mean value of the control signal $\hat{u}(t-d)$, $u_{new} = U(k)$, and where $U = [u_1, u_2, ..., u_N]$ is the admissible control signal values sampled from the control signal distribution with N equals to the number of the generated samples.

In order to find the minimum of the performance measure criterion using the value of the performance criterion function only, the updating rule is:

$$\begin{array}{lll} \text{If} & e_{k+1}^2 \leq e_k^2 & u_{\text{old}} = u_{\text{new}} \\ \text{Otherwise} & & u_{\text{old}} = u_{\text{old}} \end{array} \quad (13)$$

Since $e_{k+1} \leq e_k$, ΔV_k is always negative and consequently the error is decreasing by updating the control signal in that way.

5. SIMULATION STUDY

5.1 *Introduction*

In order to illustrate the validity of the theoretical developments, we consider the liquid-level system described by the following second order equation

$$\begin{aligned} y(t) = {} & 0.9722y(t-1) + 0.3578u(t-1) \\ & - 0.1295u(t-2) - 0.3103y(t-1)u(t-1) \\ & - 0.04228y^2(t-2) + 0.1663y(t-2)u(t-2) \\ & - 0.3513y^2(t-1)u(t-2) \\ & + 0.3084y(t-1)y(t-2)u(t-2) \\ & + 0.1087y(t-2)u(t-1)u(t-2) \\ & - \bar{v}y^2(t-1)y(t-2) \end{aligned} \quad (14)$$

This model has been used in [13] to illustrate theoretical development for the direct adaptive controller. Because disturbances play an important part in real world processes, a stochastic component, $\bar{v}$, has been added to this model. This component is considered to be a Gaussian random variable with a mean of 0.03259 and a 0.2 variance. The plant has been considered to be described by equation (14). In order to identify the plant, an input-output model described by the following equation was chosen:

$$\hat{y}(t) = f(y(t-1), y(t-2), u(t-1), u(t-2))$$

where f is a Gaussian radial basis function network. This neural network model was trained using the scaled conjugate gradient optimisation algorithm, based on noisy input output data measurements taken from the plant with sampling time of $1s$. The input to the plant and the model was a sinc function followed by a sine wave in the interval $[-1, 1]$ with additive Gaussian noise of zero mean and 0.3 variance. The single optimal structure for the neural network found by applying the cross validation method consisted of 6 Gaussian basis functions. Similarly an input output model described by the following equation was chosen to find the inverse model of the plant:

$$\hat{u}(t-1) = f^{-1}(y(t-1), y(t-2), y(t), u(t-2))$$

where f^{-1} is a Gaussian radial basis function network. The training data has been the same as in the forward model. A neural network with 6 Gaussian basis functions was found to be the best model.

5.2 *Classical Inverse Control Approach*

After training the inverse controller off line, the control network is brought on line and the control signal is calculated at each instant of time from the control neural network and by setting the output value $y(t)$ at time t equal to the desired value $y_r(t)$

$$u(t-1) = f^{-1}(y(t-1), y(t-2), y_r(t), u(t-2))$$

where $y_r(t) = 0.2 * r(t-1) + 0.8 * y_r(t-1)$ and r is the set point. The predicted mean value from the neural network was forwarded to the plant. After running the process for about 600 time steps the output of the system went unstable, and the classical inverse controller was unable to force the plant output to follow the reference output.

5.3 *Proposed Control Approach*

In our new approach, both the mean and the variance of the control signal were estimated. Following the procedure presented earlier, the best control signal was found and forwarded to the plant. This control signal was obtained from a small number of samples from the Gaussian distribution, typically maximum of 27 samples. The overall performance of the plant under the proposed method is shown in figure 3, where it is evident that the system outputs remain stable across the whole region, and that the proposed sampling approach managed to stabilise the plant. The control signal is shown in figure 4, and the variance of this control law is shown in figure 5. The error from the absolute difference between the plant output and the desired output of the classical inverse controller and the proposed sampling approach is shown in figure 6. More specifically figure 6 is the plot of $\text{error} = |y - y_r|_{\text{sampling}} - |y - y_r|_{\text{classical inverse}}$ against the time. y is the actual plant output. From this figure we can see that the sampling approach is no worse than taking the mean in the inverse control, and in addition, the sampling method remains stable in regions where the classical approach diverges.

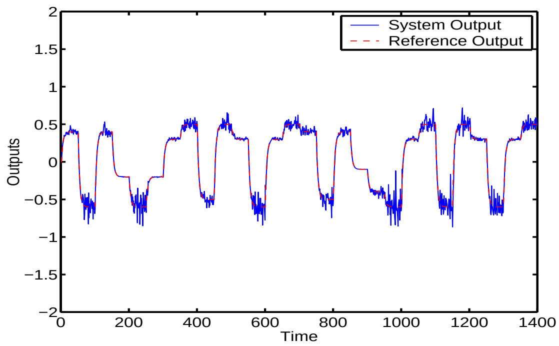

Fig. 3. The desired and actual output values.

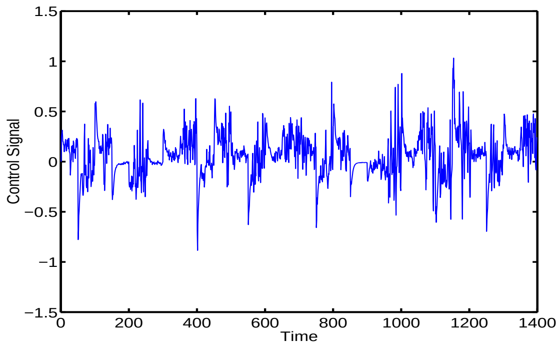

Fig. 4. The Control Signal.

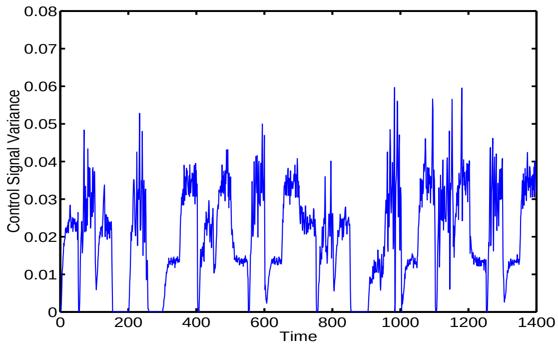

Fig. 5. The Control Signal Variance.

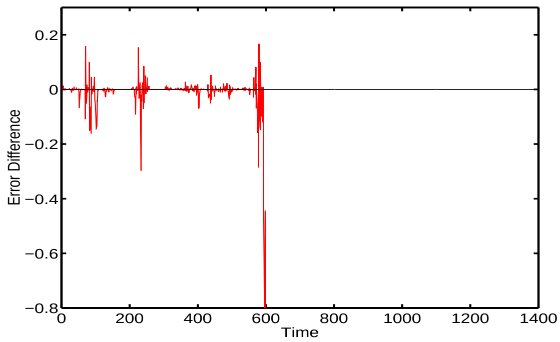

Fig. 6. The Error Difference.

6. CONCLUSIONS

This work focused on modelling conditional distribution of the control signal, by modelling its conditional distribution function. The estimation of the distribution function is based on the use of neural network as an approximation method to approximate the inverse model of the plant.

Since the use of the predicted mean value of the control signal may not be optimal for any given performance function, a new method that uses uncertainty measure around the predicted mean value of the control signal was proposed. The new proposed method allows for the control signal to be adapted from its distribution, to obtain a better estimate of the control signal than the mean. Convergence of the output error from updating the control signal was verified by using a proper Lyapunov function. The proposed control strategy in this work choose the optimal control value almost in the same way as in the dynamic programming as mentioned in the introduction. However, the proposed method is computationally more efficient and dose not base on the use of the recurrence relation as in the dynamic programming. This is because of the fact that the estimated mean and variance being predicted from a neural network which is supposed to be optimised on the input output data. By predicting the mean and the variance of the control signal from the neural network, searching for a better value of the control signal than the mean can be performed only in this region in which the optimal trajectory is expected to lie.

Simulation experiments demonstrated the successful application of the proposed strategy to improve the controller performance and to stablize a class of nonlinear control system with large uncertainties. Since we are sampling our control signal from the estimated distribution and choosing one which better fits the model, the predicted mean value of the control signal in the next time step should be more accurate. By feeding back a better value of the control signal, another benefit is that there should be no need to change the controller parameters as long as we are dealing with stationary processes.

The example given in this paper is perhaps the simplest representative of a whole class of density-estimating neural networks (such as the Mixture Density Network) and also points out a fruitful direction for control research: that of sampling control signals from estimated distribution functions which can incorporate even more information on the full distribution such as higher order moments beyond just the first two, representing the mean and the uncertainty around the mean. This more general approach is not constrained by assumptions of invertibility and can deal with hysteretic processes as well.

7. REFERENCES

[1] D.A. White and D. Sofge, *Handbook of Intelligent Control* (Van Nostrand Reinhold, 1992).

[2] R. Herzallah, *Process Identification and Control Using Neural Networks*, (MSc thesis, University of Jordan, 1996).

[3] K.S. Narendra and K. Parthasarathy, *Neural Networks in Control Systems*.

[4] M.A. Botto, B. Wams, T. Boom, and J. Costa, Robust Stability of Feedback Linearised Systems Modelled with Neural Networks: Dealing with Uncertainty, *Engineering Applications of Artificial Intelligence* **13** (2000) 659–670.

[5] Z. Huang and S.N. Balakrishnan, Robust Adaptive Critic Based Neurocontrollers for Systems with Input Uncertainties, *IEEE* (2000) 67–72.

[6] Y. Li, N. Sundararajan and P. Saratchandran, Neuro-Flight Controllers for Aircraft Using Minimal Resource Allocating Networks (MRAN), *Neural Computing and Applications* (2001) 172–183.

[7] M.B. McFarland and A.J. Calise, Robust Adaptive Control of Uncertain Nonlinear Systems Using Neural Networks, *IEEE Transactions on Automatic Control* (1997) 1–5.

[8] C.M. Bishop, *Neural Networks for Pattern Recognition* (Oxford University Press, 1995).

[9] D. Lowe and C. Zapart, Point wise Confidence Interval Estimation by Neural Networks: A Comparative study Based on Automative Engine Calibration, *Neural Computation and Application* **8** (1999) 77–85.

[10] W.A. Wright, G. Ramage, D. Cornford and I.T. Nabney, Neural Network modelling with Input Uncertainty: Theory and Application, *Journal of VLSI Signal Processing* **26** (2000) 169-188.

[11] R.E. Larson, *State Increment Dynamic Programming* (American Elsevier Publishing Company, INC. New York, 1968).

[12] L. Moreno, L. Acosta, J.A. Mendez, A. Hamilton, G.N. Marichal, J.D. Pineiro and J.L. Sanchez, Stochastic Optimal Controllers for a DC Servo Motor: Applicability and Performance, *Control Eng. Practice* **4** (1996) 757–764.

[13] M.S. Ahmed, Neural-Net-Based Direct Adaptive Control for a Class of Nonlinear Plants, *IEEE Transactions on Automatic Control* **45** (2000) 119–124.