Monograph:
Fabri, S. and Kadirkamanathan, V. (1997) Dual and Adaptive Control of Nonlinear Stochastic Systems Using Neural Networks. Research Report. ACSE Research Report 660 . Department of Automatic Control and Systems Engineering

---

DUAL ADAPTIVE CONTROL OF NONLINEAR STOCHASTIC SYSTEMS USING NEURAL NETWORKS

Simon Fabri and **Visakan Kadirkamanathan**

Department of Automatic Control & Systems Engineering
The University of Sheffield
PO Box 600, Mappin Street, Sheffield S1 3JD, UK
Email: fabri/visakan@acse.sheffield.ac.uk

Research Report No. **660**

Department of Automatic Control & Systems Engineering
The University of Sheffield

January 1997

Dual Adaptive Control of Nonlinear Stochastic Systems using Neural Networks

Simon Fabri and **Visakan Kadirkamanathan**
Department of Automatic Control & Systems Engineering
The University of Sheffield, UK

Abstract

A suboptimal dual adaptive system is developed for control of stochastic, nonlinear, discrete-time plants that are affine in the control input. The nonlinear functions are assumed to be unknown and neural networks are used to approximate them. Both Gaussian radial basis function and sigmoidal multilayer perceptron neural networks are considered and parameter adjustment is based on Kalman filtering techniques. The result is a control law that takes into consideration the uncertainty of the parameter estimates, thereby eliminating the need of performing prior open-loop plant identification. The performance of the system is analyzed by simulation and Monte Carlo analysis and the advantages of the scheme are clearly outlined.

1 INTRODUCTION

The use of neural networks for adaptive control of the affine class of nonlinear systems in discrete time has been recently investigated [9, 10, 11, 21, 22, 27]. The neural networks used are normally the multilayer perceptron (MLP) with sigmoidal units [31] or the Gaussian radial basis function (RBF) type [29]. They are included for modelling the system functionals which are assumed to be unknown. Adaptive laws are used to adjust the network parameters so as to obtain good control performance.

The approaches taken so far in neural adaptive control typically adopt a heuristic certainty equivalence procedure. This implies that the network approximations are used in a control law as if they were the true system functions, completely ignoring their uncertainty. When the uncertainty is large, for example during start-up, this can lead to an inadequate transient response. To take the uncertainty of the unknowns into consideration, a stochastic adaptive approach can be taken [5, 33]. This leads to the so-called *dual control* principle [5, 14, 40, 41] introduced by Fel'dbaum in the 1960's [12, 13]. Dual adaptive control has been analyzed mainly for adaptive control of linear systems with unknown parameters [4, 8, 15, 24, 30, 39] or for nonlinear systems having known functionals but whose state must be estimated [34, 36]. Because of the advantages associated with it there has been a recent resurgence of research on dual control [1, 15, 16, 30, 41]. However none of these addresses the problem when the system is nonlinear and the functions are unknown.

Hence in this work we shall investigate the use of dual adaptive control for the affine class of nonlinear, discrete-time systems when the nonlinear functions are unknown and a stochastic additive disturbance is present at the output. Neural networks shall be used for modelling the unknown functions.

The fundamental idea behind dual control is that it treats the unknown parameters as random variables and the situation is cast as a stochastic adaptive optimal control problem; the control law ensuring minimization of a cost function which *explicitly* includes information about parameter uncertainty in addition to the usual information on tracking or regulation of the system output and state variables. This concept is particularly appealing to adaptive control because the estimates to the unknown parameters are uncertain due to both the measurement noise corrupting the signal that drives the parameter estimation algorithm and also to the incomplete (a priori) information on the value of the parameters. Because of the way that the cost function is formulated, dual control leads to a closed-loop system having the *dual* features of (i) taking the system state optimally along a desired trajectory, with due consideration given to the uncertainty of the parameter estimates and (ii) eliciting further information so as to reduce future parameter uncertainty, thereby improving the estimation process. Effect (i) is called *caution* because, in providing the tracking function, the controller does not use the estimated parameters blindly as if they were true. Effect (ii) is called *probing* because the controller generates signals that encourage faster parameter convergence. Such a controller is said to be actively adaptive. These two features are generally conflicting because efficient parameter estimation demands a rich control input whilst, simultaneously, the control input must be in a way so as to maintain optimal state tracking; the latter input not necessarily

being rich. The balance between these two conflicting demands is determined via the cost function.

The dual cost function is based on optimizing the N-stage criterion

$$J_{dual} = E\left\{\sum_{t=0}^{N-1}[y(t+1) - y_r(t+1)]^2 \mid Y^t\right\} \tag{1}$$

where $y_r(t)$ is the system reference input, $y(t)$ is the controlled output, $E\{\cdot\}$ denotes mathematical expectation taken over all random variables, including the parameters, and Y^t denotes the information state at time t, defined as $Y^t := \{y(t)\ldots y(0) \quad u(t-1)\ldots u(0)\}$. It turns out that the control input which minimizes J_{dual} must satisfy the so-called *Bellman equation*:

$$V(\xi(t), t) = \min_{u(t)} E\left\{(y(t) - y_r(t))^2 + V(\xi(t+1), t+1) \mid Y^{t-1}\right\}$$

where $\xi(t)$, known as the hyperstate, is the full conditional probability density function of the unknown parameters given Y^{t-1}.

Although, in principle, $u(t)$ which optimizes the Bellman equation can be found by dynamic programming techniques, in most practical situations this is impossible to implement because it involves operations that are highly computationally and memory intensive [5]. For this reason most practical adaptive controllers disregard completely the dual features proposed by Fel'dbaum and they are referred to as *non dual controllers*. Two such examples are the Heuristic Certainty Equivalence (HCE) and the Cautious controllers [7]. The HCE controller lacks caution because the parameter estimates are simply "plugged" in a control law that was determined on the basis that the true value of the parameters is known. In most cases, failure to take into consideration the uncertainty of the parameter estimates often results in a suboptimal solution. Systems that maintain optimality even when the uncertainty of the estimates is neglected are said to be *certainty equivalent*. However such systems are rare, a well known example being the linear quadratic Gaussian (LQG) problem which is certainty equivalent with respect to the state estimates. Controllers based on a law that was derived under the assumption of certainty equivalence, when in fact it does not hold, are called HCE controllers.

Note that most neural network adaptive control schemes adopt a HCE approach, ignoring completely the initial parameter uncertainty arising due to lack of prior knowledge of the parameters. To avoid the serious problems in control caused by this, intensive open-loop off-line training is normally performed first to identify the plant and reduce the prior uncertainty of the parameters. Then the control phase is started with the neural network parameters set to these pre-trained values, which are substantially close to the actual values. In our case this pre-training phase is avoided and the initial (large) parameter uncertainty is taken into consideration by compensating for it in the control law derived from dual adaptive principles. This is more efficient and economical in practical applications because the off-line training scheme can be time consuming and hence expensive.

The Cautious controller is based on optimization of the one-stage criterion

$$J_{cautious} = E\left\{(y(t+1) - y_r(t+1))^2 \mid Y^t\right\}$$

This results in a control law that does take into consideration parameter uncertainty but which, on the other hand, lacks probing. The reason is that, being a one stage criterion, the control does not influence the future parameter estimates. Hence it typically results in very small control signals when the uncertainty is large and consequently tracking is slow and the parameters take a longer time to converge. This often leads to a temporary switching-off of the control signal and an increase in parameter uncertainty; a phenomenon called *turn-off* [5].

Another class of controllers attempts to solve the problem of complexity of Bellman's equation by deriving control laws that are implementable but which, to a certain extent, retain the desireable properties of the ideal dual controller, *i.e.* caution and probing. This class of controllers, called *suboptimal dual*, can be broadly divided into two types:

1. the *implicit* type which attempts to find approximations to the exact solution of the Bellman equation [30, 34, 35]. In general this approach still results in relatively computationally complex solutions.

2. The *explicit* type which generally starts off from a cost function which explicitly includes information regarding parameter estimation [8, 15, 25].

This investigation shall use an explicit type suboptimal dual approach based on that described in references [8, 25] for suboptimal control of linear stochastic systems, but extended to the case of stochastic discrete-time affine nonlinear systems. Gaussian radial basis function and sigmoidal multilayer perceptron neural networks shall both be considered for modelling the unknown nonlinear system functions.

2 CONTROLLER DESIGN

2.1 The control objective

The objective is to control the stochastic single-input single-output affine nonlinear system of the general form:

$$y(t) = f[\mathbf{x}(t-1)] + g[\mathbf{x}(t-1)]u(t-1) + e(t) \tag{2}$$

where $y(t)$ is the output, $u(t)$ the control input, $\mathbf{x}(t-1) = [y(t-n)\ldots y(t-1)\quad u(t-1-m)\ldots u(t-2)]^T$ is the system state vector, $m \leq n$ are known system parameters, $f[\mathbf{x}(t-1)]$ and $g[\mathbf{x}(t-1)]$ are the unknown nonlinear functions of the state vector, $e(t)$ is independent and zero-mean Gaussian noise with known variance σ^2. As explained in Chen and Khalil [11], for stability of the internal dynamics, we assume that the system is minimum phase and also that $g[\mathbf{x}(t-1)]$ is bounded away from zero.

If the nonlinear functions $f[\mathbf{x}(t-1)]$,$g[\mathbf{x}(t-1)]$ were known, the control law

$$u(t) = \frac{y_r(t+1) - f[\mathbf{x}(t)]}{g[\mathbf{x}(t)]} \tag{3}$$

results in $y(t+1) - y_r(t+1) = e(t+1)$, which minimizes J_{dual} because the term in the summation of cost function (1) will then be $e^2(t+1)$ which by assumption is independent of $u(t)$ and Y^t [5].

2.2 The Gaussian RBF neural network controller

We shall first develop the design of the suboptimal dual controller implemented via Gaussian radial basis function neural networks.

2.2.1 Radial basis function networks

Two Gaussian radial basis function neural networks [29] are used to approximate the nonlinear functions $f[\mathbf{x}(t-1)]$, $g[\mathbf{x}(t-1)]$ within a compact set $\chi_n \subset \Re^n$, where the state vector $\mathbf{x}(t-1)$ is known to be contained. χ_n thus represents the network approximation region. The output of the neural networks is given by

$$\begin{aligned} \hat{f}_r[\mathbf{x}(t-1), \hat{\mathbf{w}}_\mathbf{f}(t)] &= \hat{\mathbf{w}}_\mathbf{f}{}^T(t)\Phi_\mathbf{f}[\mathbf{x}(t-1)] \\ \hat{g}_r[\mathbf{x}(t-1), \hat{\mathbf{w}}_\mathbf{g}(t)] &= \hat{\mathbf{w}}_\mathbf{g}{}^T(t)\Phi_\mathbf{g}[\mathbf{x}(t-1)] \end{aligned} \tag{4}$$

where $\hat{\mathbf{w}}_\mathbf{f}$, $\hat{\mathbf{w}}_\mathbf{g}$ are vectors containing the linear coefficients (parameters) of the neural network and $\Phi_\mathbf{f}[\mathbf{x}(t-1)]$, $\Phi_\mathbf{g}[\mathbf{x}(t-1)]$ are the Gaussian basis function vectors, whose *ith* element is given by,

$$\begin{aligned} \Phi_{\mathbf{f}i} &= \exp\left\{\frac{-||\mathbf{x} - \mathbf{m}_{\mathbf{f}_i}||^2}{2\sigma_f^2}\right\} \\ \Phi_{\mathbf{g}_i} &= \exp\left\{\frac{-||\mathbf{x} - \mathbf{m}_{\mathbf{g}_i}||^2}{2\sigma_g^2}\right\} \end{aligned}$$

where $\mathbf{m}_{\mathbf{f}_i}$, $\mathbf{m}_{\mathbf{g}_i}$ are the coordinates of the centre of the *ith* basis function and σ_f^2, σ_g^2 are the variances for the networks approximating $f[\mathbf{x}(t-1)]$ and $g[\mathbf{x}(t-1)]$ respectively.

Gaussian RBF networks satisfy the Universal Approximation Property [28] stating that one could always find a number of basis functions k^*, variances and optimal parameter vectors $\mathbf{w}_\mathbf{f}^*$, $\mathbf{w}_\mathbf{g}^*$ such that within a closed and bounded region of the network input space, the function approximation error could be reduced arbitrarily. In our case, the basis functions shall be centred on points of a regular square sampling mesh inside χ_n so that the mesh spacing, the variance of the basis functions and the network parameters directly affect the optimal approximation accuracy of the neural networks inside χ_n. Sanner and Slotine [32] describe methods for determining χ_n, the mesh spacing and the variance that will satisfy any desired accuracy assuming that bounds on the smoothness and the magnitude of the spectrum of $f[\mathbf{x}(t-1)]$, $g[\mathbf{x}(t-1)]$ are known. The optimal network parameters $\mathbf{w}_\mathbf{f}^*$ and $\mathbf{w}_\mathbf{g}^*$ are, however, unknown and so the actual parameter vectors $\hat{\mathbf{w}}_\mathbf{f}$, $\hat{\mathbf{w}}_\mathbf{g}$ are adjusted recursively by estimation techniques.

2.2.2 Parameter estimation

Assuming that inside χ_n the network approximation errors are negligibly small when the optimal parameter vectors $\mathbf{w}_\mathbf{f}^*$, $\mathbf{w}_\mathbf{g}^*$ are used, we could say that

$$\begin{aligned} f[\mathbf{x}(t-1)] &\approx \mathbf{w}_\mathbf{f}^{*\,T}\Phi_\mathbf{f}[\mathbf{x}(t-1)] \\ g[\mathbf{x}(t-1)] &\approx \mathbf{w}_\mathbf{g}^{*\,T}\Phi_\mathbf{g}[\mathbf{x}(t-1)] \end{aligned}$$

Realizing that the optimal parameter vectors $\mathbf{w}_\mathbf{f}^*$, $\mathbf{w}_\mathbf{g}^*$ are constant and substituting the above approximations in equation (2), we get:

$$\begin{aligned} \mathbf{w}^*(t+1) &= \mathbf{w}^*(t) \\ y(t) &= \mathbf{w}^{*\,T}\Phi[\mathbf{x}(t-1)] + e(t) \end{aligned} \tag{5}$$

where $\mathbf{w}^* = [\mathbf{w}_\mathbf{f}^{*\,T} \vdots \mathbf{w}_\mathbf{g}^{*\,T}]^T$, $\Phi[\mathbf{x}(t-1)] = [\Phi_\mathbf{f}{}^T[\mathbf{x}(t-1)] \vdots \Phi_\mathbf{g}{}^T[\mathbf{x}(t-1)]u(t-1)]^T$.

Equation (5) shows the advantage of using Gaussian RBF networks because the parameters to be estimated appear linearly in the output equation. Hence the well established estimation techniques based on Kalman filtering [3, 18] can be used if we assume that the initial optimal parameter vector $\mathbf{w}^*(0)$ has a Gaussian distribution with mean m and covariance R_o. Note that in practice R_o can be used to reflect the extent of prior knowledge of the parameters; larger values indicating great uncertainty, and hence less confidence, in the initial parameter estimate $\mathbf{w}^*(0)$ [23, 38].

Using Kalman filter theory [3, 5] we obtain the following recursive parameter adjustment rules

$$\begin{aligned} \mathbf{K}(t) &= \mathbf{P}(t)\Phi[\mathbf{x}(t-1)]/(\sigma^2 + \Phi^T[\mathbf{x}(t-1)]\mathbf{P}(t)\Phi[\mathbf{x}(t-1)]) \\ \hat{\mathbf{w}}(t+1) &= \hat{\mathbf{w}}(t) + \mathbf{K}(t)(y(t) - \hat{\mathbf{w}}^T(t)\Phi[\mathbf{x}(t-1)])\ ,\ \hat{\mathbf{w}}(0) = m \\ \mathbf{P}(t+1) &= \left\{\mathbf{I} - \mathbf{K}(t)\Phi^T[\mathbf{x}(t-1)]\right\}\mathbf{P}(t)\ ,\ \mathbf{P}(0) = R_o \end{aligned} \tag{6}$$

From the properties of Kalman filtering it follows that the conditional distribution of $y(t+1)$ given Y^t is Gaussian with mean $\hat{\mathbf{w}}^T(t+1)\Phi[\mathbf{x}(t)]$ and variance $\Phi^T[\mathbf{x}(t)]\mathbf{P}(t+1)\Phi[\mathbf{x}(t)] + \sigma^2$.

2.2.3 The control law

We shall consider an explicit type cost function similar to the innovations dual controller developed by Milito et al. for linear systems [25]. This cost function explicitly includes a term concerning the prediction error (or innovations sequence) $\epsilon(t+1) := y(t+1) - \hat{\mathbf{w}}^T(t+1)\Phi[\mathbf{x}(t)]$. The idea is to reward performance that encourages $\epsilon^2(t+1)$ to remain high, so that parameter updating in equation (6) is driven by richer information. Hence the cost function has the form:

$$J_{inn} = E\left\{[y(t+1) - y_r(t+1)]^2 + qu^2(t) + r\epsilon^2(t+1) \mid Y^t\right\} \tag{7}$$

where $E\{\cdot \mid Y^t\}$ denotes mathematical expectation given information Y^t, $y_r(t)$ is the reference input, $q \geq 0$ and $-1 \leq r \leq 0$ are designer-chosen scalar weighting factors. Note that a higher q induces larger penalty on high control energy and smaller r rewards higher variance of the innovations.

Since the reference and control inputs are deterministic and $y(t+1) \mid Y^t$ is Gaussian distributed, then from the previously-mentioned properties of the Kalman filter it can be shown that:

$$J_{inn} = (r+1)(\Phi^T[\mathbf{x}(t)]\mathbf{P}(t+1)\Phi[\mathbf{x}(t)] + \sigma^2) + qu^2(t) + (\hat{\mathbf{w}}^T(t+1)\Phi[\mathbf{x}(t)] - y_r(t+1))^2$$

It is now relatively easy to optimize this cost function with respect to $u(t)$ by differentiation and equating to zero, avoiding the need to resort to a complex dynamic programming algorithm that would have resulted from solution of the ideal optimal dual control problem. This results in the optimal control law:

$$u^*(t) = \frac{(y_r(t+1) - \hat{f}_r[\mathbf{x}(t), \hat{\mathbf{w}}_\mathbf{f}(t+1)])\hat{g}_r[\mathbf{x}(t), \hat{\mathbf{w}}_\mathbf{f}(t+1)] - (1+r)v_{gf}}{\hat{g}_r^2[\mathbf{x}(t), \hat{\mathbf{w}}_\mathbf{g}(t+1)] + q + (1+r)v_{gg}} \tag{8}$$

where $v_{gf} := \Phi_\mathbf{g}{}^T[\mathbf{x}(t)]\mathbf{P}_\mathbf{gf}(t+1)\Phi_\mathbf{f}[\mathbf{x}(t)]$ and $v_{gg} := \Phi_\mathbf{g}{}^T[\mathbf{x}(t)]\mathbf{P}_\mathbf{gg}(t+1)\Phi_\mathbf{g}[\mathbf{x}(t)]$ and matrix $\mathbf{P}(t+1)$ has been repartitioned as:

$$\left|\begin{array}{ccc} \mathbf{P}_\mathbf{ff}(t+1) & \vdots & \mathbf{P}_\mathbf{gf}^T(t+1) \\ \cdots & \cdots & \cdots \\ \mathbf{P}_\mathbf{gf}(t+1) & \vdots & \mathbf{P}_\mathbf{gg}(t+1) \end{array}\right|$$

where $\mathbf{P}_\mathbf{ff}$, $\mathbf{P}_\mathbf{gg}$ are square $(n_{rf} \times n_{rf})$, $(n_{rg} \times n_{rg})$ sub-matrices; n_{rf}, n_{rg} denoting the number of basis functions in the $\hat{f}_r$, $\hat{g}_r$ networks respectively.

2.2.4 Analysis of the control law

From equation (8) it is clear that the controller can take into consideration the uncertainty of the parameter estimates via inclusion of the terms v_{gf} and v_{gg} in the control law. Parameter r is acting as a weighting factor where, at one extreme, the controller can completely ignore these parameter uncertainty terms when $r = -1$ or, at the other extreme, it gives maximum attention to parameter uncertainty when $r = 0$. For intermediate settings, $-1 < r < 0$, one obtains a balance between these two extremes.

The case $r = 0$ is equivalent to optimization of $J_{cautious}$ with the disadvantages normally associated with this kind of suboptimal controller; namely turn-off and slowness of response. This follows because strong emphasis is given to the uncertainty of the parameter estimates and the controller is very cautious on using them. In fact very small control signals are applied when terms v_{gf} and v_{gg} are large.

The case $r = -1$, on the other hand, corresponds to a controller designed on a Heuristic Certainty Equivalence basis because it is effectively using the parameter estimates $\hat{\mathbf{w}}(t)$ as if they were the optimal parameters $\mathbf{w}^*$. This is equivalent to replacing the actual nonlinear system functions in control law (3) with the network approximations, completely disregarding the approximation uncertainty. Hence certainty equivalence is being assumed to hold when in fact it is not true, because neither $J_{cautious}$ and less so J_{dual} is optimized. This often results in excessively high peak overshoot during the transient part of the response because no consideration is given to the fact that the parameters have not yet achieved their optimal values and so quite large control signals are applied, as evident by the absence of the uncertainty terms in the control law (8).

The case $-1 < r < 0$ provides the best compromise between these two extremes, being neither too cautious (and hence sluggish) nor too bold (and hence crude). This is the motivation behind the design of Milito's innovation suboptimal dual controller, where the level of caution can be varied between zero and a value which optimizes the one-stage criterion $J_{cautious}$.

2.3 The sigmoidal MLP neural network controller

Another neural network that is more widely used than the radial basis function type is the sigmoidal multilayer perceptron network. Unfortunately this neural network does not preserve the advantage of linearity in the unknown parameters and so its parameter adjustment rules tend to be more complex than for the RBF case. However, because the support of its basis functions is not localized, one typically requires a relatively smaller number of neurons to achieve similar function approximation accuracy. In fact it has been shown that given particular conditions on the spectrum of the function to be approximated, the approximation error of a multilayer neural network is independent of the dimensionality of the input space [6]. This consideration is more so important for cases of high dimensional input, considering that RBF networks suffer from the *curse of dimensionality* where the number of units increases exponentially with state dimension. Hence we shall now adapt the design of the suboptimal dual controller to the case of implementation via sigmoidal multilayer networks.

2.3.1 Sigmoidal MLP networks

Two sigmoidal MLP networks shall be used, each having one hidden layer and one summing output node to approximate the unknown nonlinear functions $f[\mathbf{x}(t-1)]$, $g[\mathbf{x}(t-1)]$ respectively. The outputs of the two neural networks are respectively given by

$$\begin{aligned} \hat{f}_s[\mathbf{x}(t-1), \hat{\mathbf{c}_\mathbf{f}}(t)] &= \hat{\mathbf{c}_\mathbf{f}}^T(t)\mathbf{\Phi_f}[\mathbf{x}(t-1)] \\ \hat{g}_s[\mathbf{x}(t-1), \hat{\mathbf{c}_\mathbf{g}}(t)] &= \hat{\mathbf{c}_\mathbf{g}}^T(t)\mathbf{\Phi_g}[\mathbf{x}(t-1)] \end{aligned} \tag{9}$$

where $\hat{\mathbf{c}_\mathbf{f}}$,$\hat{\mathbf{c}_\mathbf{g}}$ are vectors containing the parameters (weights) of the output layer neuron and $\mathbf{\Phi_f}[\mathbf{x}(t-1)]$,$\mathbf{\Phi_g}[\mathbf{x}(t-1)]$ are the sigmoidal activation function vectors, representing the output of the nodes

in the hidden layer, whose *ith* element is given by:

$$\begin{aligned}\Phi_{\mathbf{f}_i} &= 1/(1+\exp(-\hat{\mathbf{w}}_{\mathbf{f_i}}^T \mathbf{x_a}(t-1))) \\ \Phi_{\mathbf{g}_i} &= 1/(1+\exp(-\hat{\mathbf{w}}_{\mathbf{g_i}}^T \mathbf{x_a}(t-1)))\end{aligned}$$

where $\hat{\mathbf{w}}_{\mathbf{f_i}}^T$, $\hat{\mathbf{w}}_{\mathbf{g_i}}^T$ are the parameter vectors of the *ith* neuron in the hidden layer and $\mathbf{x_a}(t-1) := [\mathbf{x}(t-1)^T \vdots 1]^T$ denotes the system state vector augmented by an additional constant input serving as a neuron bias input. The number of units in the hidden layers of the $\hat{f}_s$ and $\hat{g}_s$ networks is denoted by n_{sf} and n_{sg} respectively.

Note that the (optimal) values of the network parameters required to ensure some desired approximation accuracy are unknown and hence require estimation. For convenience these parameters shall be grouped in a single vector $\hat{\mathbf{w}} := [\hat{\mathbf{c}_f}^T \ \hat{\mathbf{w}}_{\mathbf{f1}}^T \ \ldots \ \hat{\mathbf{w}}_{\mathbf{fi}}^T \ \ldots \ \hat{\mathbf{c}_g}^T \ \hat{\mathbf{w}}_{\mathbf{g1}}^T \ \ldots \ \hat{\mathbf{w}}_{\mathbf{gi}}^T \ \ldots]^T$. In contrast to the RBF network, not all of these parameters appear linearly in the network equations (9) because of the parameters of the hidden layer neurons.

2.3.2 Parameter estimation

As in the previous case we shall assume that there exist some optimal values of the parameter vector $\hat{\mathbf{w}}$, denoted by $\mathbf{w}^* := [\mathbf{c_f^*}^T \ \mathbf{w_{f1}^*}^T \ \ldots \ \mathbf{w_{fi}^*}^T \ \ldots \ \mathbf{c_g^*}^T \ \mathbf{w_{g1}^*}^T \ \ldots \ \mathbf{w_{gi}^*}^T \ \ldots]^T$, such that the network approximation errors are arbitrarily small in the space of interest. Hence from equations (9) and (2), the plant can be modelled by the equations

$$\begin{aligned}\mathbf{w}^*(t+1) &= \mathbf{w}^*(t) \\ y(t) &= h(t, \mathbf{w}^*, \mathbf{x}(t-1), u(t-1)) + e(t)\end{aligned} \tag{10}$$

where $h(t, \mathbf{w}^*, \mathbf{x}(t-1), u(t-1)) := \mathbf{c_f^*}^T \Phi_{\mathbf{f}}[\mathbf{w}_{\mathbf{f_i}}^*, \ \mathbf{x}(t-1)] + \mathbf{c_g^*}^T \Phi_{\mathbf{g}}[\mathbf{w}_{\mathbf{g_i}}^*, \ \mathbf{x}(t-1)]u(t-1)$ is a nonlinear function of the unknown optimal parameters $\mathbf{w}^*$. Since the parameters to be estimated do not appear linearly in the system model, nonlinear estimation techniques have to be used. The Extended Kalman Filter (EKF) [2, 18, 26] is the most widely used nonlinear estimator and for our case it also represents a natural progression from the (linear) Kalman filter used in the RBF network case. The EKF has also been applied in system identification using MLP networks [20, 37] and shown to give better results than the back-propagation training algorithm [31] and also for function estimation using Gaussian RBF networks [19].

The extended Kalman filter applied to the model of equation (10) gives

$$\begin{aligned}\mathbf{K}(t) &= \mathbf{P}(t)\nabla_{\mathbf{h}}{}^T(t)/(\sigma^2 + \nabla_{\mathbf{h}}(t)\mathbf{P}(t)\nabla_{\mathbf{h}}{}^T(t)) \\ \hat{\mathbf{w}}(t+1) &= \hat{\mathbf{w}}(t) + \mathbf{K}(t)(y(t) - h(t, \hat{\mathbf{w}}(t), \mathbf{x}(t-1), u(t-1)) \ , \ \hat{\mathbf{w}}(0) = m \\ \mathbf{P}(t+1) &= \{\mathbf{I} - \mathbf{K}(t)\nabla_{\mathbf{h}}(t)\}\,\mathbf{P}(t) \ , \ \mathbf{P}(0) = R_o\end{aligned} \tag{11}$$

where $\nabla_{\mathbf{h}}(t)$ denotes the transpose of the gradient vector of $h(t)$ with respect to $\mathbf{w}^*$ evaluated at $\mathbf{w}^* = \hat{\mathbf{w}}(t)$, for which a closed form expression can be found directly by differentiation of $h(t)$

resulting in

$$\nabla_{\mathbf{h}}(t) = [\nabla_{\mathbf{h_f}}(t) \vdots \nabla_{\mathbf{hg}}(t) u(t-1)] \tag{12}$$

where

$$\nabla_{\mathbf{h_f}}(t) = \left[\frac{1}{1+\exp(-\hat{\mathbf{w}}_{\mathbf{fi}}^T \mathbf{x_a})} \cdots \left(\frac{\hat{c}_{f_i} \exp(-\hat{\mathbf{w}}_{\mathbf{fi}}^T \mathbf{x_a})}{\left(1+\exp(-\hat{\mathbf{w}}_{\mathbf{fi}}^T \mathbf{x_a})\right)^2} \right) \mathbf{x_a}^T \cdots \right], \; i = 1 \ldots n_{sf}$$

$$\nabla_{\mathbf{h_g}}(t) = \left[\frac{1}{1+\exp(-\hat{\mathbf{w}}_{\mathbf{gi}}^T \mathbf{x_a})} \cdots \left(\frac{\hat{c}_{g_i} \exp(-\hat{\mathbf{w}}_{\mathbf{gi}}^T \mathbf{x_a})}{\left(1+\exp(-\hat{\mathbf{w}}_{\mathbf{gi}}^T \mathbf{x_a})\right)^2} \right) \mathbf{x_a}^T \cdots \right], \; i = 1 \ldots n_{sg}$$

To be able to proceed in a similar manner to the Gaussian RBF case, we shall assume that the initial optimal parameter vector $\mathbf{w}^*(0)$ has a Gaussian distribution with mean m and covariance R_o and that the conditional distribution of $\mathbf{w}^*(t+1)$ given Y^t is approximately Gaussian with mean $\hat{\mathbf{w}}(t+1)$ and covariance matrix $\mathbf{P}(t+1)$ as given by the EKF equations (11) [17]. It should be emphasized that the latter is only an approximation and does not follow naturally as in the linear Kalman filter. Even so, this assumption still does not lead to straight forward conclusions regarding the conditional distribution of $y(t+1)$ given Y^t, because of the nonlinear relationship between y and $\mathbf{w}^*$ as seen in equation (10). Hence we shall linearize $y(t+1)$ about $\hat{\mathbf{w}}(t+1)$ by a first order Taylor series expansion to get

$$\begin{aligned} y(t+1) &\approx h(t+1, \hat{\mathbf{w}}(t+1), \mathbf{x}(t), u(t)) + \nabla_{\mathbf{h}}(t+1)(\mathbf{w}^*(t+1) - \hat{\mathbf{w}}(t+1)) + e(t+1) \\ &= \nabla_{\mathbf{h}}(t+1)\mathbf{w}^*(t+1) + h(t+1, \hat{\mathbf{w}}(t+1), \mathbf{x}(t), u(t)) \\ &\quad -\nabla_{\mathbf{h}}(t+1)\hat{\mathbf{w}}(t+1) + e(t+1) \end{aligned} \tag{13}$$

where $\nabla_{\mathbf{h}}(t+1)$ is the same as $\nabla_{\mathbf{h}}(t)$ but evaluated at $\hat{\mathbf{w}}(t+1)$, $u(t)$ and $\mathbf{x}(t)$.

Hence from the assumed Gaussian conditional distribution of $\mathbf{w}^*(t+1)$, mentioned before, and equation(13) it follows that the conditional distrubution of $y(t+1)$ given Y^t is also approximately Gaussian with mean $h(t+1, \hat{\mathbf{w}}(t+1), \mathbf{x}(t), u(t))$ and variance $\nabla_{\mathbf{h}}(t+1)\mathbf{P}(t+1)\nabla_{\mathbf{h}}{}^T(t+1) + \sigma^2$.

2.3.3 The control law

We shall consider the same cost function as before, based on Milito's innovations dual controller given by equation (7), where in this case the innovations sequence $\epsilon(t+1)$ is defined as $\epsilon(t+1) := y(t+1) - h(t+1, \hat{\mathbf{w}}(t+1), \mathbf{x}(t), u(t))$.

Proceeding exactly as for the RBF case and using the approximations on the conditional distribution of $y(t+1)$ outlined in the previous section, the optimal control law is obtained as:

$$u^*(t) = \frac{(y_r(t+1) - \hat{f}_s[\mathbf{x}(t), \hat{\mathbf{w}}_{\mathbf{f}}(t+1)])\hat{g}_s[\mathbf{x}(t), \hat{\mathbf{w}}_{\mathbf{f}}(t+1)] - (1+r)\mu_{gf}}{\hat{g}_s^2[\mathbf{x}(t), \hat{\mathbf{w}}_{\mathbf{g}}(t+1)] + q + (1+r)\mu_{gg}} \tag{14}$$

where $\mu_{gf} := \nabla_{\mathbf{hg}}(t+1)\mathbf{P_{gf}}(t+1)\nabla_{\mathbf{h_f}}{}^T(t+1)$ and $\mu_{gg} := \nabla_{\mathbf{hg}}(t+1)\mathbf{P_{gg}}(t+1)\nabla_{\mathbf{hg}}{}^T(t+1)$ represent the uncertainty terms for this case and $\mathbf{P}(t+1)$ has been partitioned as before, but in this case $\mathbf{P_{ff}}$, $\mathbf{P_{gg}}$ are square $(n_{sf}(n+m+2) \times n_{sf}(n+m+2))$, $(n_{sg}(n+m+2) \times n_{sg}(n+m+2))$ sub-matrices respectively.

Note that the result is a control law which is identical in structure to that of the RBF network controller, except that the Gaussian basis function vectors appearing in the uncertainty terms are replaced by the corresponding gradient vector of h evaluated at $\hat{\mathbf{w}}$. This reflects the principle behind the extended Kalman filter which was used for estimation; namely that it linearizes a nonlinear system about the most recent parameter estimate and consequently the statistical properties of its estimates are calculated on this basis. Hence, as far as controller performance is concerned, similar comments to the RBF dual adaptive controller apply in the case of the MLP controller.

3 SIMULATION RESULTS

The performance of the system was tested by simulation of two example plants. Note that the plants were not subjected to an initial open loop system identification phase and closed loop control was activated immediately, with the initial parameter estimates selected at random and not having been pre-trained.

3.1 Simulation 1

The plant of the first simulation is given by:

$$y(t+1) = sin(x(t)) + cos(3x(t)) + (2 + cos(x(t)))u(t) + e(t+1) \tag{15}$$

where state $x(t) = y(t)$ and the noise variance $\sigma^2 = 0.001$. $f(x) = sin(x(t)) + cos(3x(t))$ and $g(x) = 2 + cos(x(t))$ represent the unknown nonlinear dynamics. The reference input is obtained by sampling a unit amplitude, 0.1 Hz square wave filtered by a network of transfer function $1/(s+1)$. A Gaussian RBF controller is implemented in this system. The network approximation region is chosen as $\chi_n = [-2, 2]$ and two RBF networks are used to approximate $f(x)$ and $g(x)$ respectively. The $\hat{f}$ network is chosen to have Gaussian basis functions of variance 1 placed on a mesh of spacing 0.5, whilst the $\hat{g}$ network basis functions have variance of 3.6 and a mesh spacing of 2. The Kalman filter initial parameter covariance was set to $\mathbf{P}(0) = 1000\mathbf{I}$.

Trials were conducted with three different controllers corresponding to the heuristic certainty equivalence (HCE) ($r = -1$), cautious ($r = 0$) and innovation dual ($-1 < r < 0$) controller. The same noise sequence, initial conditions and reference input were used in each case. A typical output is shown in figure 1. Note that, as expected, the HCE controller initially responds violently because it is not taking into consideration the inaccuracy of the parameter estimates. Only after the initial period, when the parameters converge, does the control assume good tracking. On the contrary

the cautious controller is slow to respond during the initial period, knowing that the parameter estimates are still inaccurate. Hence although no violent response is exhibited, the controller is practically turned off during the initial part of the response. The innovation dual controller reaches a compromise between these two extremes, clearly showing no particularly unacceptable peak overshoot whilst tracking the reference input earlier than the cautious contoller. Hence, even qualitatively, it is clear that the performance of the innovation dual controller is the better one. To quantify the performance a Monte Carlo analysis involving 500 trials was performed. A fresh realization of the noise sequence was generated at each trial and the accumulated cost $V(T) = \sum_{t=0}^{T}(y_r(t) - y(t))^2$ was calculated over the whole simulation interval time T at each trial. The results are shown in figure 1.

The average of the accumulated cost over 500 trials is shown in table 1 where it is seen that the innovations dual controller shows the best performance; although the improvement over the cautious controller is marginal. This is due to the fact that the nonlinear functions considered in this example are time invariant and hence the optimal parameters of the network are constant. In case of time-varying nonlinearities, the improvement of the dual controller over the cautious is expected to be greater because the uncertainty of the estimates would not be pronounced only during the initial period of the response but also when the parameters change appreciably. Hence, under time-variant conditions, the cautious controller is bound to turn off more frequently during operation.

In order to reduce the overshoot of the HCE controller it is tempting to increase the cost function weight q associated with $u(t)$. Although this does reduce overshoot, in some cases it can cause a general deterioration of the tracking capabilities as shown in figure 2, where q was set to 1 for the HCE controller and 0.0001 for the other two. Although the HCE accumulated cost is reduced drastically to around 15, it is still higher than 6, the order of magnitude of the cautious and dual controllers. The reason is that q tends to limit the amplitude of the control at all times and not only during those periods when parameter uncertainty is large.

3.2 Simulation 2

The plant of the second simulation is similar to that used in reference [11], namely:

$$y(t+1) = \frac{1.5y(t)y(t-1)}{1+y^2(t)+y^2(t-1)} + 0.35sin(y(t)+y(t-1)) + 1.2u(t) + e(t+1) \qquad (16)$$

where $\mathbf{x}(t) = [y(t-1) \;\; y(t)]^T$, $g(\mathbf{x}) = 1.2$ and $f(\mathbf{x}) = (1.5y(t)y(t-1)/(1+y^2(t)+y^2(t-1))) + 0.35sin(y(t)+y(t-1))$ represent the unknown nonlinear dynamics and the noise $e(t)$ has variance $\sigma^2 = 0.01$. The reference input is obtained by sampling a unit amplitude, 0.1 Hz square wave filtered by $1/(s+1)$. Both a RBF and a MLP controller shall be tested on this plant. For the RBF network, the approximation region is chosen as $\chi_n = [-2,2] \times [-2,2]$ for the $\hat{f}$ network and $[-2,2]$ for the $\hat{g}$ network. The $\hat{f}$ network has Gaussian basis functions of variance 0.64 placed on a mesh of spacing 0.5, whilst the $\hat{g}$ network has a variance of 9 and a mesh spacing of 1.5. The Kalman

filter initial parameter covariance was set to $\mathbf{P}(0) = \mathbf{I}$. As before, trials were conducted using the three different controllers with q set to 0.0001 in all cases. A typical output is shown in figure 3. Note that the same comments as before apply in this case, with the innovations dual controller performing better. Figure 3 also shows the results of the accumulated cost following Monte Carlo analysis. The average of the accumulated cost over 100 trials can also be seen in table 1 where it is clear that the innovations dual controller shows best performance.

3.3 Simulation 3

The same plant of simulation 2 is used for testing the MLP controller. The noise $e(t)$ had a variance of 0.05 and the $\hat{f}$ and $\hat{g}$ networks were structured with 10 and 5 hidden unit neurons respectively. The initial parameter estimates were chosen at random and the initial covariance matrix $\mathbf{P}(0)$ had a diagonal structure with the terms corresponding to $\hat{f}$ and $\hat{g}$ set to 50 and 10 respectively. Figure 4 shows a typical output for the three different controllers. Once again it is noted that, compared with the innovations dual controller, the HCE controller exhibits higher peak overshoot whilst the cautious controller is slower to respond. The results of the Monte Carlo analysis are shown in figure 4 and table 1. Note that the same comments as for the RBF controller apply, with the innovations dual controller having best performance. An important observation is that the MLP controller has resulted in a similar performance as the RBF controller but with an appreciably smaller number of neurons: 10 as opposed to 64 for the $\hat{f}$ network. This follows because the $\hat{f}$ network has a two dimensional input space and so the number of nodes required in the RBF network starts to increase reasonably with dimensionality.

Simulation No.	HCE	CAUTIOUS	DUAL
1	1434	6.7	5.7
2	273	7.72	5.96
3	500	48	42

Table 1: Mean accumulated cost

4 CONCLUSION

A neural network adaptive dual control approach has been investigated for the affine class of nonlinear systems under stochastic conditions when the nonlinearities are unknown. The controller developed is based on the explicit type suboptimal dual criterion as originally proposed by Milito et al. [25] for linear systems. It was shown that the advantages of Milito s design, namely superior performance over non-dual controllers and a relatively simple control law, also hold for the more complex case of adaptive control of stochastic affine nonlinear systems. Moreover, because the

suboptimal dual controller takes into consideration parameter uncertainty by introducing caution-like effects, control and estimation are performed simultaneously and it is not necessary to preceed the control phase with an off-line, open-loop system identification phase, as is typically the case with adaptive HCE neural network control schemes.

Both Gaussian RBF and sigmoidal MLP neural networks have been tested for learning the unknown nonlinearities. The network training algorithms were derived using Kalman filters for the RBF case and extended Kalman filter theory for the MLP case. Simulation results have been presented and the advantage of utilising a suboptimal dual approach has been confirmed by Monte Carlo analysis. Evaluation of the conditions under which the assumptions taken in the derivation of the MLP control law are justified is open to further reseach; together with analysis of the stability properties of the closed loop system.

References

[1] B. J. Allsion, J. E. Ciarniello, J. C. Tessier and G. A. Dumont, "Dual adaptive control of chip refiner motor load", *Automatica*, Vol. 31, No. 8, pp.1169-1184, 1995.

[2] B. O. Anderson and J. B. Moore, *Optimal Filtering*, Prentice-Hall Inc., U.S.A, 1979.

[3] K. J. Astrom, *Introduction to Stochastic Control Theory*, Academic Press, New York, 1970.

[4] K. J. Astrom and A. Helmersson, "Dual control of an integrator with unknown gain", *Comp. & Maths. with Applications*, Vol. 12A, No. 6, p.653-662, 1986.

[5] K. J. Astrom and B. Wittenmark, *Adaptive Control*, Addison-Wesley, Reading, MA 1989.

[6] A. R. Barron, "Universal approximation bounds for superpositions of a sigmoidal function", *IEEE Transactions on Information Theory*, Vol. 39, No.3, pp.930-945, May 1993.

[7] Y. Bar-Shalom and E. Tse, "Dual effect, certainty equivalence and separation in stochastic control", *IEEE Transactions on Automatic Control*, Vol. AC-19, pp.494-500, October 1974.

[8] S. S. Chan and M. B. Zarrop, "A suboptimal dual controller for stochastic systems with unknown parameters", *International Journal of Control*, Vol.41, No.2, pp.507-524, 1985.

[9] F. C. Chen, "Back-propagation neural networks for nonlinear self-tuning adaptive control", *IEEE Control Systems Magazine, Special issue on neural networks for control systems*, April 1990.

[10] F. C. Chen and H. K. Khalil, "Adaptive control of nonlinear systems using neural networks", *International Journal of Control*, Vol.55, pp.1299-1317, 1992.

[11] F. C. Chen and H. K. Khalil, "Adaptive control of a class of nonlinear discrete-time systems using neural networks", *IEEE Transactions on Automatic Control*, Vol.40, No.5, May 1995.

[12] A. A. Fel'dbaum, "Dual control theory I-II", *Automation and Remote Control*, Vol.21, pp.874-880, 1033-1039, 1960.

[13] A. A. Fel'dbaum, "Dual control theory III-IV", *Automation and Remote Control*, Vol.22, pp.1-12, 109-121, 1961.

[14] A. A. Fel'dbaum, *Optimal Control Systems*, Academic Press, New York, 1965.

[15] N. M. Filatov, H. Unbehauen and U. Keuchel, "Dual version of direct adaptive pole placement controller", *Preprints of the 5th IFAC Symposium on Adaptive Systems in Control and Signal Processing*, pp. 449-454, Hungary, June 1995.

[16] M. Gevers, "Identification for control", *Preprints of the 5th IFAC Symposium on Adaptive Systems in Control and Signal Processing*, pp.1-12, Hungary, June 1995.

[17] O. L. R. Jacobs, *Introduction to Control Theory*, Oxford University Press, U.K., 1974.

[18] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, Academic Press, New York, 1970.

[19] V. Kadirkamanathan and M. Niranjan "A function estimation approach to sequential learning with neural networks", *Neural Computation 5*, Vol.6, pp.954-975, 1993.

[20] A. Kimura, I. Arizono and H. Ohta, "An improvement of a back propagation algorithm by the extended Kalman filter and demand forecasting by layered neural networks", *Interntional Journal of Systems Science*, Vol.27, No.5, pp. 473-482, 1996.

[21] G. P. Liu, V. Kadirkamanathan and S. A. Billings, "Nonlinear predictive control via neural networks", *Proceedings of the IEE International Conference on Control '96*, pp.746, U.K., September 1996.

[22] G. P. Liu, V. Kadirkamanathan and S. A. Billings, "Neural network based variable structure control for nonlinear discrete systems ", Research Report No.626, Department of Automatic Control & Systems Engineering, The University of Sheffield, Sheffield, UK, May 1996.

[23] L. Ljung, "Asymptotic behaviour of the extended Kalman filter as a parameter estimator for linear systems", *IEEE Transactions on Automatic Control*, Vol. AC-24, pp.36-50, February 1979.

[24] A. L. Maitelli and T. Yoneyama, "A two-stage suboptimal controller for stochastic systems using approximate moments", *Automatica*, Vol. 30, No. 12, pp.1949-1954, 1994.

[25] R. Milito, C. S. Padilla, R. A. Padilla and D. Cadorin, "An innovations approach to dual control", *IEEE Transactions on Automatic Control*, Vol. AC-27, No.1,pp.133-137,February 1982.

[26] N. E. Nahi, *Estimation theory and applications*, John Wiley & Sons, U.S.A., 1975.

[27] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks", *IEEE Transactions on Neural Networks*, Vol.1, No.1, pp.4-27, 1990.

[28] J. Park and I. W. Sandberg, "Universal approximation using radial basis function networks", *Neural Computation*, Vol.3, pp.246-257, 1990.

[29] T. Poggio and F. Girosi, "Networks for Approximation and Learning", *Proceedings of the IEEE*, Vol.78, No.9, September 1990.

[30] L. Pronzato, C. Kulcsar, E. Walter, "An actively adaptive control policy for linear models", *IEEE Transactions on Automatic Control*, Vol.41, No.6, June 1996.

[31] D. E. Rumelhart, G. E. Hinton and Williams, "Learning internal representations by error propagation", In J. L. McClelland and D. E. Rumelhart, (Eds). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Bradford Books/MIT Press, Cambridge, MA, Vol.1, pp.318-361, 1986.

[32] R. M. Sanner and J. J. E. Slotine, "Gaussian Networks for Direct Adaptive Control", *IEEE Transactions on Neural Networks*, Vol.3,No.6, November 1992.

[33] T. Soderstrom, *Discrete-time stochastic systems: Estimation and control*, Prentice-Hall International, U.K., 1994.

[34] E. Tse and Y. Bar-Shalom, "Wide-semse adaptive dual control for nonlinear stochastic systems", *IEEE Transactions on Automatic Control*, Vol. AC-18, No.2, pp. 98-108, April 1973.

[35] E. Tse and Y. Bar-Shalom, "An actively adaptive control for linear systems with random parameters via the dual control approach", *IEEE Transactions on Automatic Control*, Vol. AC-18, No.2, pp. 109-117, April 1973.

[36] E. Tse and Y. Bar-Shalom, "Actively adaptive control for nonlinear stochastic systems", *Proceedings of the IEEE*, Vol.64, No.8, pp. 1172-1181, August 1976.

[37] K. Watanabe, T. Fukuda and S. Tzafestas, "Learning algorithms of layered neural networks via extended Kalman filters", *Interntional Journal of Systems Science*, Vol.22, No.4, pp. 753-768, 1991.

[38] P. E. Wellstead and M. B. Zarrop, *Self-Tuning Systems:Control and Signal Processing*, John Wiley & Sons, U.K., 1991.

[39] C. J. Wenk and Y. Bar-Shalom, "A multilple model adaptive dual control algorithm for stochastic systems with unknown parameters", *IEEE Transactions on Automatic Control*, Vol. AC-25, No.4, pp.703-710, 1980.

[40] B .Wittenmark, "Stochastic adaptive control methods: a survey" *International Journal of Control*, Vol. 21, pp. 705-730, 1975.

[41] B .Wittenmark, "Adaptive dual control methods: An overview." *Preprints of the 5th IFAC Symposium on Adaptive Systems in Control and Signal Processing*, pp. 67-72, Hungary, June 1995.

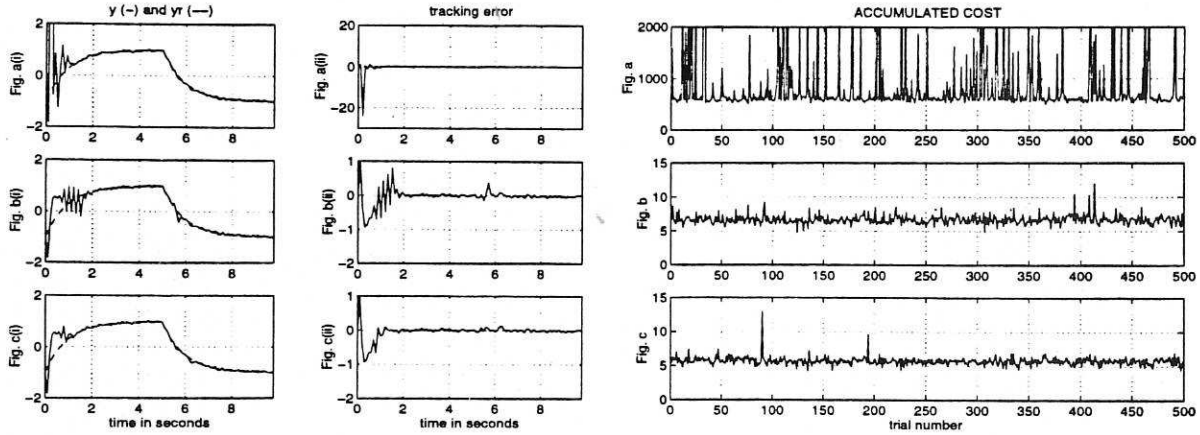

Figure 1: Tracking error & Accumulated Cost: (a) HCE (b) Cautious (c) Dual

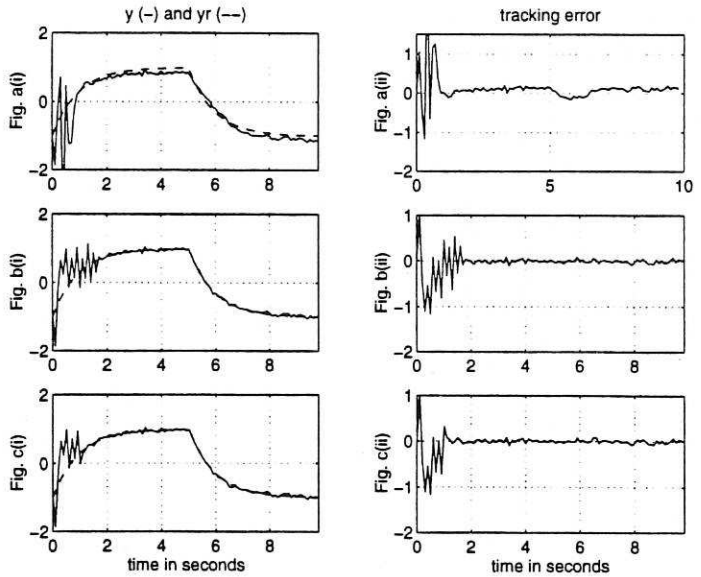

Figure 2: Effect of q: (a) HCE ($q = 1$) (b) Cautious ($q = 0.0001$) (c) Dual ($q = 0.0001$)

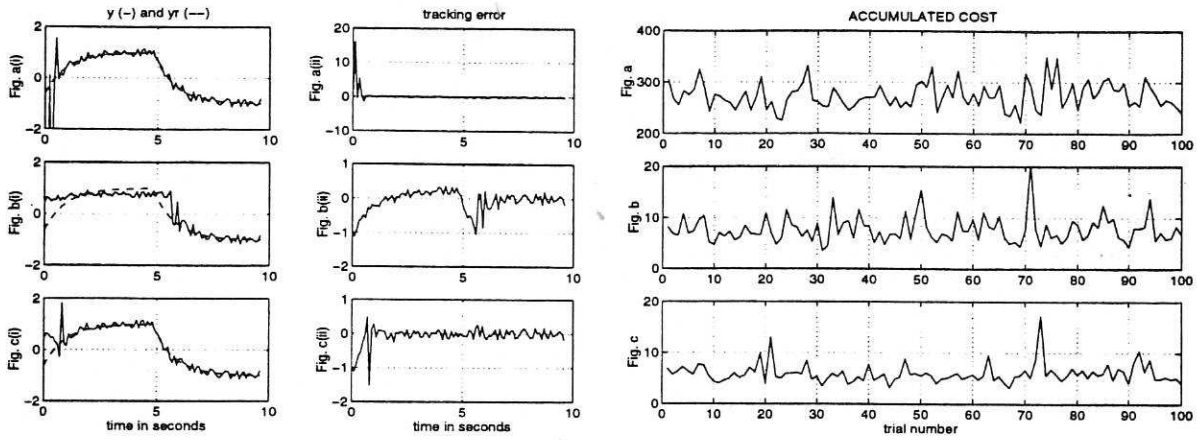

Figure 3: Tracking Error & Accumulated Cost: (a) HCE (b) Cautious (c) Dual

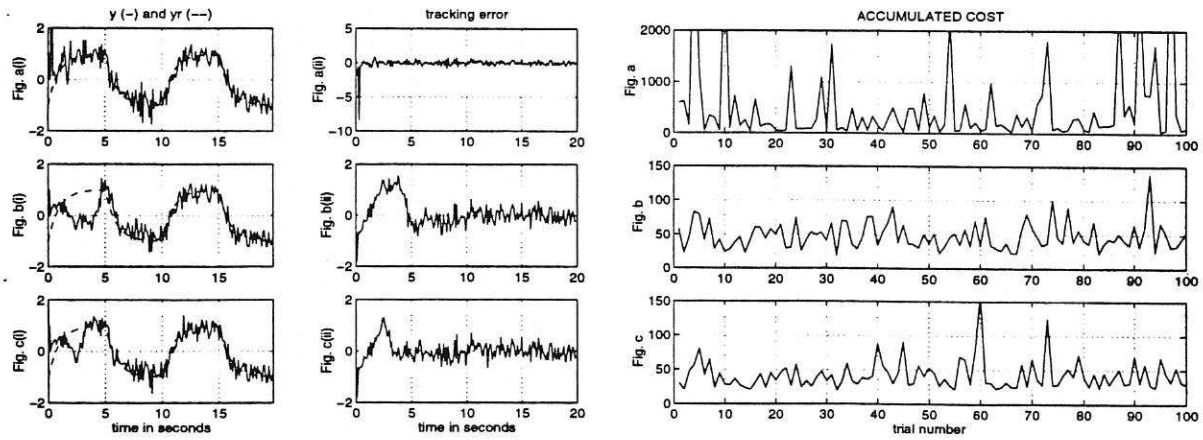

Figure 4: Tracking Error & Accumulated Cost: (a) HCE (b) Cautious (c) Dual