

# NEURAL CONTROL OF NONLINEAR SYSTEMS WITH COMPOSITE ADAPTATION FOR IMPROVED CONVERGENCE OF GAUSSIAN NETWORKS

S. Fabri, V. Kadiramanathan

*Department of Automatic Control & Systems Engineering  
The University of Sheffield, Mappin Street, Sheffield S1 3JD, U.K.  
Fax : +44 114 273 1729 and e-mail : COP95SF/visakan@sheffield.ac.uk*

**Keywords** : adaptive, neural nets, nonlinear control.

## Abstract

The use of composite adaptive laws for control of the affine class of nonlinear systems having unknown dynamics is proposed. These dynamics are approximated by Gaussian radial basis function neural networks whose parameters are updated by a composite law that is driven by both tracking and estimation errors. This is motivated by the need to improve the speed of convergence of the unknown parameters, hence resulting in better system performance. To ensure global stability despite the inevitable network approximation errors, the control law is augmented with a low gain sliding mode component and deadzone adaptation is used for the indirect part of the composite law. The stability of the system is analyzed and the effectiveness of the method is demonstrated by simulation.

## 1 Introduction

The use of control schemes based on composite adaptive laws for systems having unknown or uncertain parameters has resulted in improved performance over direct adaptive control laws [2, 13]. These laws are composite in the sense that they combine features of both direct and indirect adaptive methods. The direct method relies on the use of the tracking error (i.e. the difference between the controlled output and the reference input) for direct adjustment of controller parameters in such a way as to force the tracking error to zero, a task generally ensured by Lyapunov stability considerations. The indirect method relies on the prediction error (i.e. the difference between some output and an approximation to it) to obtain estimates of the system parameters for use in a certainty equivalence control law.

In situations whereby the unknown system dynamics constitute nonlinear functionals, rather than simply parameters, neural networks can be used for function ap-

proximation and their parameters are adjusted via on-line adaptive laws. In this respect, for an affine class of nonlinear systems, methods using direct adaptive laws and Gaussian radial basis function neural networks are well documented [3, 10, 14]. In this paper we shall implement a composite adaptive law for neural control of the same class of nonlinear systems, so as to improve parameter convergence and hence overall system performance. The approach is based on the composite adaptive schemes described in [13] for robot manipulators where the unknowns consisted of system parameters, such as mass. This work shall thus extend composite adaptation to a more general class of affine nonlinear plants whose unknowns are nonlinear functionals.

## 2 Problem statement

The objective is to design a controller that results in good tracking performance for the class of nonlinear, single-input single-output plants that could be expressed in the affine form:

$$y^{(r)}(t) = f(\mathbf{x}) + g(\mathbf{x})u(t) \quad (1)$$

where  $\mathbf{x} \in \mathcal{R}^n$  is the system state vector,  $u(t)$  is the system input,  $y(t)$  is the system output,  $f(\mathbf{x})$  and  $g(\mathbf{x})$  are nonlinear functions of the system state (where  $g(\mathbf{x}) \neq 0 \forall \mathbf{x}$ ) and  $r$  is the relative degree of the system [4, 12].

The control task is for the output  $y(t)$  to track a desired output  $y_d(t)$  while the state  $\mathbf{x}(t)$  is to remain bounded. As shown in [4, 11], the conditions for this to occur are:

- the zero dynamics of system (1) must be globally exponentially stable.
- the internal dynamics of system (1) must satisfy a Lipschitz condition in terms of the system's normal states.
- the desired output  $y_d(t)$  must satisfy the condition that  $y_d, y_d^{(1)}, \dots, y_d^{(r)}$  are bounded, where  $y_d^{(i)}$  represents the  $i$ th derivative with respect to time of  $y_d$ .

- the control law generating  $u(t)$  must ensure that the output and its  $(r-1)$  derivatives  $y, y^{(1)}, \dots, y^{(r-1)}$  are bounded and that  $y(t)$  asymptotically tracks  $y_d(t)$ .

Adaptive control shall be used because it is assumed that the nonlinear system functions  $f(\mathbf{x}), g(\mathbf{x})$  are unknown, and so neural networks are used to generate approximations  $\hat{f}(\mathbf{x}), \hat{g}(\mathbf{x})$  to be used in the control law  $u(t) = \frac{-\hat{f}(\mathbf{x}) + \dot{y}_d(t)}{\hat{g}(\mathbf{x})} + u_{si}(t)$ , as in references [10, 14]. Adaptivity takes place through the process of on-line adjustment of the parameters of the neural networks and it is this aspect that shall be covered in more detail in this paper, by investigating the use of composite adaptive schemes.

### 3 Controller design

#### 3.1 The Neural Networks

Two Gaussian radial basis function neural networks [9] are used to approximate the nonlinear functions  $f(\mathbf{x}), g(\mathbf{x})$  within a compact set  $\chi_n \subset \mathbb{R}^n$ , where the state vector  $\mathbf{x}(t)$  is known to be contained.  $\chi_n$  thus represents the network approximation region. The output of the neural networks is given by

$$\begin{aligned}\hat{f}(\mathbf{x}, \hat{\mathbf{w}}_f) &= \kappa(\mathbf{x}) \hat{\mathbf{w}}_f^T \Phi_f(\mathbf{x}) + f_o(\mathbf{x}) \\ \hat{g}(\mathbf{x}, \hat{\mathbf{w}}_g) &= \kappa(\mathbf{x}) \hat{\mathbf{w}}_g^T \Phi_g(\mathbf{x}) + g_o(\mathbf{x})\end{aligned}\quad (2)$$

where  $\hat{\mathbf{w}}_f, \hat{\mathbf{w}}_g$  are vectors containing the linear coefficients (parameters) of the neural network and  $\Phi_f(\mathbf{x}), \Phi_g(\mathbf{x})$  are the Gaussian basis function vectors, whose  $i$ th element is given by,

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

where  $\mathbf{m}_{f_i}, \mathbf{m}_{g_i}$  are the coordinates of the centre of the  $i$ th basis function and  $\sigma_{f_i}^2, \sigma_{g_i}^2$  are the variances for the networks approximating  $f(\mathbf{x})$  and  $g(\mathbf{x})$  respectively.  $f_o(\mathbf{x}), g_o(\mathbf{x})$  are known prior estimates to  $f(\mathbf{x}), g(\mathbf{x})$ .

$\kappa(\mathbf{x})$  is included as in [10, 14] so that the contribution of the networks is limited only to  $f_o(\mathbf{x})$  and  $g_o(\mathbf{x})$  when  $\mathbf{x}$  is outside  $\chi_n$  because the network is not able to approximate correctly outside the approximation region. Within  $\chi_n^-$ , a slightly smaller subset of the network approximation region  $\chi_n$ ,  $\kappa(\mathbf{x}) = 1$  so that the full network approximation is utilised. In the boundary between  $\chi_n^-$  and  $\chi_n$ ,  $\kappa(\mathbf{x})$  is reduced gradually so as to suppress the neural network output smoothly in those areas that lie close to the boundary of the network approximation region, thus avoiding sudden switching when  $\mathbf{x}$  goes outside  $\chi_n$ . Hence,  $\kappa(\mathbf{x})$  is defined as being equal to one if  $\mathbf{x} \in \chi_n^-$ , 0 if  $\mathbf{x} \notin \chi_n$  and  $0 \leq \kappa(\mathbf{x}) \leq 1$  otherwise.

The neural networks can never approximate the actual functions perfectly, so that there will always be the

presence of approximation errors. This introduces disturbances in the system that could lead to parameter drift [6, 12]. However, Gaussian radial basis function networks satisfy the Universal Approximation property [7], stating that given any uniform bounds  $\epsilon_f, \epsilon_g$  one could always find an optimal number of basis functions  $k^*$ , variances and optimal parameter vectors  $\mathbf{w}_f^*, \mathbf{w}_g^*$  such that  $\forall \mathbf{x} \in \chi_n$  the corresponding optimal network approximation errors, denoted respectively by  $\Delta_f := f^*(\mathbf{x}) - f(\mathbf{x})$  and  $\Delta_g := g^*(\mathbf{x}) - g(\mathbf{x})$ , satisfy:

$$\begin{aligned}|\Delta_f| &= |f^*(\mathbf{x}) - f(\mathbf{x})| \leq \epsilon_f \\ |\Delta_g| &= |g^*(\mathbf{x}) - g(\mathbf{x})| \leq \epsilon_g\end{aligned}$$

where  $f^*(\mathbf{x}) = \hat{f}(\mathbf{x}, \mathbf{w}_f^*)$  and  $g^*(\mathbf{x}) = \hat{g}(\mathbf{x}, \mathbf{w}_g^*)$ .

When  $\mathbf{x} \notin \chi_n$ , the optimal approximation errors are given by  $\Delta_f = (f_o(\mathbf{x}) - f(\mathbf{x}))$  and  $\Delta_g = (g_o(\mathbf{x}) - g(\mathbf{x}))$ , assumed to be bounded by known bounds  $\bar{f}_o$  and  $\bar{g}_o$  respectively.

Hence one could say that  $\forall \mathbf{x}$

$$\begin{aligned}|\Delta_f| &\leq \kappa(\mathbf{x})\epsilon_f + (1 - \kappa(\mathbf{x}))\bar{f}_o \\ |\Delta_g| &\leq \kappa(\mathbf{x})\epsilon_g + (1 - \kappa(\mathbf{x}))\bar{g}_o\end{aligned}\quad (3)$$

Knowledge of these bounds is crucial in overcoming the problem of parameter drift.

The basis functions shall be centred on points of a regular square sampling mesh inside  $\chi_n$ , 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  $\chi_n$ . Reference [10] provides methods for determining  $\chi_n$ , the mesh spacing and the variance that will satisfy any desired  $\epsilon_f, \epsilon_g$  assuming that bounds on the smoothness and the magnitude of the spectrum of  $f(\mathbf{x}), g(\mathbf{x})$  are known. Note however that the optimal network parameters  $\mathbf{w}_f^*$  and  $\mathbf{w}_g^*$  are unknown, so that the actual parameter vectors  $\hat{\mathbf{w}}_f, \hat{\mathbf{w}}_g$  are adjusted recursively via the adaptation laws, to ensure system stability and good tracking performance. In composite adaptation, the adjustment of  $\hat{\mathbf{w}}_f, \hat{\mathbf{w}}_g$  shall depend upon *both* the tracking error  $e = (y - y_d)$  and an estimation error, *i.e.* a measure that reflects the error between  $y$  and its estimate as predicted by an identification model based on the neural networks.

Using equations (2) and the definitions of the optimal network approximation errors  $\Delta_f$  and  $\Delta_g$  we obtain that  $\forall \mathbf{x}$ :

$$\begin{aligned}\hat{f}(\mathbf{x}) - f(\mathbf{x}) &= \kappa \tilde{\mathbf{w}}_f^T \Phi_f + \Delta_f \\ \hat{g}(\mathbf{x}) - g(\mathbf{x}) &= \kappa \tilde{\mathbf{w}}_g^T \Phi_g + \Delta_g\end{aligned}\quad (4)$$

where  $\tilde{\mathbf{w}}_f = (\hat{\mathbf{w}}_f - \mathbf{w}_f^*)$  and  $\tilde{\mathbf{w}}_g = (\hat{\mathbf{w}}_g - \mathbf{w}_g^*)$  represent the parameter errors.

#### 3.2 The Control Law

The control law to be used is similar to that in references [3, 5, 14] namely

$$u(t) = u_{ai}(t) + u_{si}(t)\quad (5)$$

where

- $u_{al}(t) = \frac{-\hat{f}(\mathbf{x})+v(t)}{\hat{g}(\mathbf{x})}$  is inspired from feedback linearization control laws [4, 12], with  $v(t) = y_d^{(r)} - \alpha_r e^{(r-1)} - \dots - \alpha_1 e$  representing an auxiliary input whose coefficients  $\alpha_i$  are chosen such as to form a Hurwitz polynomial  $\Gamma(s) = s^r + \alpha_r s^{r-1} + \dots + \alpha_1$  in terms of the Laplace variable  $s$ .
- $u_{sl}(t)$  represents a sliding mode component introduced as in [10], [14] so as to ensure global stability if and when the state moves outside  $\chi_n$  and to ensure robustness to parameter drift whilst  $\mathbf{x} \in \chi_n$ .

Using control law (5) in system (1) and equations (4) we obtain:

$$e = \Gamma^{-1}[-\kappa \tilde{\mathbf{w}}_f^T \Phi_f - \kappa \tilde{\mathbf{w}}_g^T \Phi_g u_{al} + g u_{sl} + d(t)] \quad (6)$$

where  $d(t) = -\Delta_f - \Delta_g u_{al}$ . Note that the non-zero network approximation errors  $\Delta_f$  and  $\Delta_g$  give rise to a disturbance term  $d(t)$  that affects the error dynamics represented by equation (6).

If the tracking error  $e(t)$  is filtered by a Hurwitz polynomial  $\Psi(s)$  chosen such that  $\Psi(s)\Gamma^{-1}(s)$  is a first order lag transfer function  $\Psi\Gamma^{-1} = \frac{1}{s+k_d}$  where  $k_d > 0$ , then as in [10, 14] the filtered tracking error  $e_1 = \Psi e$  defines a suitable sliding surface [16], obtained from equation (6) as

$$e_1 = -k_d e_1 - \kappa \tilde{\mathbf{w}}_f^T \Phi_f - \kappa \tilde{\mathbf{w}}_g^T \Phi_g u_{al} + g u_{sl} + d(t) \quad (7)$$

### 3.3 Linear Parameterization

The indirect part of the composite adaptation law relies on the use of an estimation error to drive the adaptation, obtained by using the neural network outputs in an identification model. This way, the weights of the neural networks are adjusted using parameter estimation techniques. Linear parameterization provides a general model for parameter estimation methods [12], whereby one seeks a linear relation between the unknown network parameters and some measurable signal. For this case, a linearly parameterized relation is developed as follows:

From equations (1), (2) and (4)

$$y^{(r)} = \mathbf{w}_f^*{}^T \Phi_f + f_o + (\mathbf{w}_g^*{}^T \Phi_g + g_o)u(t) - \Delta_f - \Delta_g u(t)$$

This equation could be re-written in a linear form in terms of the parameter vectors  $\mathbf{w}_f^*$  and  $\mathbf{w}_g^*$ , excluding the purely derivative term  $y^{(r)}$ , by filtering with a network of transfer function  $A_o^{-1}(s)$ , where  $A_o(s) = a_o + a_1 s + \dots + a_{r-1} s^{r-1} + s^r$  is a Hurwitz and monic polynomial, reflecting stable filtration. This filtration results in the following dynamics

$$y\left(\frac{s^r}{A_o}\right) - \frac{f_o}{A_o} - \frac{g_o u(t)}{A_o} + \frac{\Delta_f}{A_o} + \frac{\Delta_g u}{A_o} = \frac{\Phi_f^T(\mathbf{x})\mathbf{w}_f^*}{A_o} + \frac{\Phi_g^T(\mathbf{x})\mathbf{w}_g^* u}{A_o}$$

which can be re-written in linear form as

$$Y = \mathbf{p}^T \mathbf{w}^* \quad (8)$$

where

$$\mathbf{p}^T = \left| \begin{array}{cc} \frac{\Phi_f^T(\mathbf{x})}{A_o} & \frac{\Phi_g^T(\mathbf{x})u}{A_o} \end{array} \right|, \quad \mathbf{w}^* = \left| \begin{array}{c} \mathbf{w}_f^* \\ \mathbf{w}_g^* \end{array} \right|$$

$$Y = y\left(\frac{s^r}{A_o}\right) - \frac{f_o}{A_o} - \frac{g_o u}{A_o} + \frac{\Delta_f}{A_o} + \frac{\Delta_g u}{A_o}.$$

Defining  $\hat{y}$  as an approximation to the actual plant output  $y$ , obtained from considering the neural network approximations in the model for the plant, one could say after comparison with the dynamics of equation (1), that

$$\hat{y}^{(r)} = \hat{f}(\mathbf{x}) + \hat{g}(\mathbf{x})u(t) \quad (9)$$

Filtration of  $\hat{y}^{(r)}$  by  $A_o^{-1}(s)$  results in a suitable identification model. In fact, after substitution of equations (2) and filtering by  $A_o^{-1}(s)$  we obtain:

$$\hat{Y} = \mathbf{p}^T \hat{\mathbf{w}} \quad (10)$$

where

$$\hat{\mathbf{w}} = \left| \begin{array}{c} \hat{\mathbf{w}}_f \\ \hat{\mathbf{w}}_g \end{array} \right|, \quad \hat{Y} = \hat{y}\left(\frac{s^r}{A_o}\right) - \frac{f_o}{A_o} - \frac{g_o u}{A_o}.$$

Subtracting (8) from (10) results in the linear relation:

$$\epsilon = \mathbf{p}^T \tilde{\mathbf{w}} - d_f \quad (11)$$

where  $\epsilon = (\hat{y} - y)\frac{s^r}{A_o}$  constitutes the estimation error,  $\tilde{\mathbf{w}} = \hat{\mathbf{w}} - \mathbf{w}^*$  is the error between the actual and the optimal network parameters and  $d_f = -\frac{\Delta_f}{A_o} - \frac{\Delta_g u}{A_o}$  reflects the disturbance due to the network inherent approximation errors, arising from the fact that the optimal parameters  $\mathbf{w}^*$  do not ensure zero approximation errors. Note that  $\epsilon$  can be generated from available signals  $\hat{y}^{(r)}$  and  $y$  by stable filtration and without using differentiators.

### 3.4 The Adaptation Law

The composite adaptation law proposed has the form

$$\dot{\hat{\mathbf{w}}} = \mathbf{P}(t)(\Phi e_1 - k_e \mathbf{p} \epsilon_\Delta) \quad (12)$$

where

- $\epsilon_\Delta = \epsilon$  if  $|\epsilon| > \overline{d}_f$  and zero otherwise, represents a deadzone function on  $\epsilon(t)$ .
- $\overline{d}_f$  represents a bound on  $d_f$  such that  $|d_f| \leq \overline{d}_f$ .
- $k_e$ , a design parameter, is a positive constant that determines the extent to which the indirect component shall be used in the composite adaptation law.
- $\mathbf{P}(t)$  is a positive definite, symmetric gain matrix that could, in general, be either constant or time varying.
- $\Phi = [\Phi_f^T(\mathbf{x}) \quad \Phi_g^T(\mathbf{x})u_{al}(t)]^T$ .

Note the composite nature of adaptation law (12), being driven by signals derived from *both* the tracking error  $e(t)$  and the estimation error  $\epsilon(t)$ . Note also the use of deadzone adaptation as used in reference [8] for the estimation part of the adaptation law, included to ensure boundedness of signals despite the presence of the disturbance  $d_f(t)$  in the estimation error equation (11).

Furthermore it is assumed that for the  $\hat{g}(\mathbf{x})$  network, a parameter resetting mechanism is included so as to keep  $\hat{g}(\mathbf{x})$  bounded below by  $g_l(\mathbf{x})$ , the latter being a known lower bound on  $g(\mathbf{x})$  satisfying  $0 < g_l(\mathbf{x}) \leq g_o(\mathbf{x}) \forall \mathbf{x} \in \mathcal{X}_n^-$ . Methods of implementing this can be found in [14], [15].

### 3.5 The Gain Update Law

If the gain matrix  $\mathbf{P}(t)$  is maintained constant, say  $\mathbf{P}(t) = \mathbf{P}_o$ , then the indirect part of the composite adaptation law would correspond to gradient descent estimation techniques [13].

On the other hand, one could opt for a time varying gain matrix which would result in better parameter convergence in the absence of persistently exciting signals [1, 6]. Various parameter estimation techniques that involve time-varying gains have been developed [6, 12, 13]. We have utilised the standard least-squares algorithm, resulting in the following gain update law:

$$\dot{\mathbf{P}}(t) = -k_\Delta \mathbf{P} \mathbf{p} \mathbf{p}^T \mathbf{P} \quad (13)$$

where  $k_\Delta = k_e$  if  $|\epsilon| > \bar{d}_f$  and zero otherwise,  $\mathbf{P}(t_o)$ , the initial gain matrix, is symmetric and positive-definite.

## 4 Stability analysis

Stability and boundedness of the system variables is obtained by using the Lyapunov function candidate

$$V = \frac{1}{2}(\epsilon_1^2 + \tilde{\mathbf{w}}^T \mathbf{P}^{-1}(t) \tilde{\mathbf{w}}) \quad (14)$$

Differentiating (14) with respect to time, substituting equation (7), using parameter update law (12) and gain update law (13) together with the identity  $\dot{\mathbf{P}}^{-1} = -\mathbf{P}^{-1} \dot{\mathbf{P}} \mathbf{P}^{-1}$  results in

$$\dot{V} = -k_d \epsilon_1^2 + e_1(gu_{s1} + d) - k_e \mathbf{p}^T \tilde{\mathbf{w}} \epsilon_\Delta + k_\Delta \frac{\tilde{\mathbf{w}}^T \mathbf{P} \mathbf{p} \mathbf{p}^T \tilde{\mathbf{w}}}{2} \quad (15)$$

Choosing the sliding mode component of the control law as

$$u_{s1}(t) = -\frac{\bar{d}(\mathbf{x})}{g_l(\mathbf{x})} \text{sign}(e_1) \quad (16)$$

where  $\bar{d}(\mathbf{x})$  is a known bound on the disturbance  $d(t)$  (*i.e.*  $|d(t)| \leq \bar{d}(\mathbf{x})$ ) ensures that the term  $e_1(gu_{s1} + d)$  appearing in equation (15) is semi negative definite.

The gradient descent case involves that  $\mathbf{P}(t)$  remains constant, *i.e.*  $\dot{\mathbf{P}}(t) = 0$ . This is equivalent to letting  $k_\Delta =$

0 in equation (13). Hence, the last term of equation (15) simply vanishes. On the other hand, the term  $k_e \mathbf{p}^T \tilde{\mathbf{w}} \epsilon_\Delta$  is either equal to zero when  $|\epsilon| \leq \bar{d}_f$  or  $k_e \mathbf{p}^T \tilde{\mathbf{w}} (\mathbf{p}^T \tilde{\mathbf{w}} - d_f)$  when  $|\epsilon| > \bar{d}_f$ . Since  $\bar{d}_f$  represents a bound on  $d_f$ , the latter term is semi-positive definite [8], so that in general,  $-k_e \mathbf{p}^T \tilde{\mathbf{w}} \epsilon_\Delta \leq 0$ . Hence,  $\dot{V} \leq -k_d \epsilon_1^2 \leq 0$ .

If a least squares algorithm via gain update law (13) is employed, then for  $|\epsilon| \leq \bar{d}_f$  the last two terms of equation (15) simply vanish, whilst for  $|\epsilon| > \bar{d}_f$  equation (15) becomes

$$\dot{V} = -k_d \epsilon_1^2 + e_1(gu_{s1} + d) - k_e \mathbf{p}^T \tilde{\mathbf{w}} \left( \frac{\mathbf{p}^T \tilde{\mathbf{w}}}{2} - d_f \right) \quad (17)$$

Since  $k_e \mathbf{p}^T \tilde{\mathbf{w}} \left( \frac{\mathbf{p}^T \tilde{\mathbf{w}}}{2} - d_f \right) \geq 0$  for  $|\epsilon| > \bar{d}_f$ , then choosing  $u_{s1}$  as shown before in equation (16) ensures that  $\forall \epsilon, \dot{V} \leq -k_d \epsilon_1^2 \leq 0$ .

Hence in *both* cases of gradient descent and least squares estimation, adaptation law (12) and gain update law (13) ensure that  $\dot{V}$  is semi negative definite, from which it follows that  $e_1(t)$  and  $\tilde{\mathbf{w}}$  are bounded. Boundedness of  $\tilde{\mathbf{w}}$  implies that  $\hat{\mathbf{w}}_f, \hat{\mathbf{w}}_g$  are bounded, keeping  $\hat{f}(\mathbf{x}), \hat{g}(\mathbf{x})$  bounded. Also  $e(t)$  and its  $(r-1)$  derivatives are bounded since  $e_1(t)$  is bounded and  $\Psi(s)$  is Hurwitz. Assuming that the desired output and its  $r$  derivatives are bounded, then the latter implies that the output and its  $(r-1)$  derivatives are bounded as well. Hence, as shown in section 2, if we additionally assume the zero dynamics of the system to be globally exponentially stable and the internal dynamics to satisfy the appropriate Lipschitz condition in terms of the system normal states [4], then  $\mathbf{x}(t)$  is bounded. Also, boundedness of  $e(t)$ ,  $y_d(t)$  and their derivatives results in boundedness of  $v(t)$ .

In addition,  $\hat{g}(\mathbf{x})$  is bounded away from zero via parameter resetting so that  $u_{d1}$  is bounded, from which follows boundedness of  $d(t)$ . These imply that  $u_{s1}(t)$  is bounded and so from equation (7),  $\dot{e}_1(t)$  is bounded. Hence,  $e_1(t)$  is not only bounded but also uniformly continuous.

Using the fact that  $\dot{V} \leq -k_d \epsilon_1^2$  and defining a function  $V_1(t) = V(t) - \int_0^t \dot{V}(\tau) + k_d \epsilon_1^2 d\tau$ , it follows that  $V_1(t) \geq 0$  and  $\dot{V}_1(t) = -k_d \epsilon_1^2$ , the latter implying that  $V_1$  is semi negative definite and uniformly continuous. Hence Barbalat's Lemma [12] implies that  $V_1(t) \rightarrow 0$  as  $t \rightarrow \infty$ , and so also  $e_1(t)$ . Since  $e(t)$  can be considered as a filtration of  $e_1(t)$  via the stable and strictly proper transfer function  $\Psi^{-1}(s)$ , then the tracking error  $e(t)$  also converges asymptotically to zero as desired.

## 5 The disturbance bounds

We shall now consider evaluation of the disturbance bounds  $\bar{d}(\mathbf{x})$  and  $\bar{d}_f(\mathbf{x})$ , which are required for use in the sliding mode component of the control law  $u_{s1}(t)$  and the deadzone of the adaptation law, respectively.

From equation (3) and the definition of  $d(t)$  it follows

that  $\forall \mathbf{x}$

$$|d(\mathbf{x})| \leq \kappa(\epsilon_f + \epsilon_g |u_{ai}|) + (1 - \kappa)(\overline{f}_o + \overline{g}_o |u_{ai}|) \quad (18)$$

Hence  $\overline{d}(\mathbf{x})$  is given by the right hand side of (18). Note that this affects the gain of the sliding mode component  $\frac{-\overline{d}(\mathbf{x})}{g_i(\mathbf{x})}$ , which is small whilst  $\mathbf{x} \in \chi_n^-$ , just enough to overcome the effect of the disturbance  $d(t)$  on signal boundedness in the direct part of the control law. On the other hand, the sliding gain increases appreciably as  $\mathbf{x}$  ventures outside the network approximation region  $\chi_n$ , taking over control so as to pull  $\mathbf{x}$  inside  $\chi_n$  if and when the state goes outside the network approximation region, as in references [10, 14].

From the definition of  $d_f$ , it follows that

$$d_f(t) = \frac{d(t) - \Delta_g u_{si}}{A_o} \quad (19)$$

From equation (3) and the definition of  $u_{si}(t)$  it follows that

$$|\Delta_g u_{si}| \leq \frac{\overline{d}}{g_i} [\kappa(\mathbf{x})\epsilon_g + (1 - \kappa(\mathbf{x}))\overline{g}_o] \quad (20)$$

so that

$$\begin{aligned} |d(t) - \Delta_g u_{si}| &\leq |d| + |\Delta_g u_{si}| \leq \\ \overline{d} + \frac{\overline{d}}{g_i} [\kappa(\mathbf{x})\epsilon_g + (1 - \kappa(\mathbf{x}))\overline{g}_o] &:= \overline{d}_1 \end{aligned}$$

$\overline{d}_1$  thus represents a bound on the term  $(d(t) - \Delta_g u_{si})$ . But equation (19) shows that  $d_f(t)$  is a stable filtration of  $(d(t) - \Delta_g u_{si})$  via  $A_o^{-1}(s)$ , and if  $A_o(s)$  is chosen to have the form  $A_o(s) = (s + \lambda)^r$  where  $\lambda > 0$ , then as shown in [12],  $|d_f^{(i)}| \leq 2^i \lambda^{i-r} \overline{d}_1$  for  $0 \leq i \leq r$ , so that

$$\overline{d}_f = \lambda^{-r} \overline{d}_1 \quad (21)$$

## 6 Simulation results

The system was tested via simulation of the nonlinear affine plant used in reference [15]:

$$\begin{aligned} \dot{x} &= \sin(x) + 0.5\cos(3x) + u \\ y &= x \end{aligned}$$

where  $g(\mathbf{x}) = 1$  is assumed known and  $f(\mathbf{x}) = \sin(x) + 0.5\cos(3x)$  represents the unknown dynamics.

The system is of order  $n = 1$  and degree  $r = 1$ . The desired output  $y_d$  is a unity amplitude, 0.1 Hz square wave filtered by  $1/(s + 1)$ . The network approximation region is chosen as  $\chi_n = [-1.7, 1.7]$  and  $\chi_n^- = [-1.1, 1.1]$ . It is assumed that no prior estimate to  $f(\mathbf{x})$  is known so that  $f_o = 0$ . For this case, an optimal network approximation error bound  $\epsilon_f = 0.02$  is obtained with radial basis functions having  $\sigma = 0.06$  and a mesh of spacing 0.05 inside  $\chi_n$  [15]. Assuming that it is known that  $f(\mathbf{x})$  is bounded by 1.5, then since  $f_o = 0$  it follows that  $\overline{f}_o = 1.5$ . The

choice of  $\Gamma(s) = (s + 1)$  results in  $v(t) = \dot{y}(t) - (y - y_d)$ . Also,  $\Psi(s)$  could be set to unity because this way  $\Psi\Gamma^{-1}$  is a first order lag transfer function having  $k_d = 1$ . The filter characteristic equation was chosen as  $A_o = (s + 5)$ , so that  $\lambda = 5$ .

Three trials were performed for comparison purposes. In the first two, the gain matrix  $\mathbf{P}_o$  was kept constant and equal to the identity matrix. Initially  $k_e$  was set to zero so as to utilize only the direct component of the adaptation law and then  $k_e$  was set to 10 so as to utilize the combined adaptation law. The results of the direct adaptive controller are shown in figures 1 a-c. Figure 1 a shows the first 100 seconds of simulation. The system is stable with the tracking error asymptotically converging to zero. Figure 1 b shows the actual function  $f(\mathbf{x})$  plotted together with the network approximation  $\hat{f}(\mathbf{x})$  using the parameters obtained after 100 seconds of adaptation. Note from figure 1 c that the network approximation error  $(f(\mathbf{x}) - \hat{f}(\mathbf{x}))$  lies within the range  $\pm 0.18$  for the values of  $\mathbf{x}$  inside  $\chi_n^-$ . The results of the composite adaptive controller are shown in figures 1 d-f. Note that the system is also stable, with the tracking error converging to zero much faster than for the direct controller. In fact, after 100 seconds, the composite law resulted in the error converging to within  $\pm 0.0058$  compared with  $\pm 0.06$  for the direct law. This improved transient performance is attributed to the fact that the composite law provides a better approximation to the unknown functions in a shorter time, because more information is used for parameter adjustment. In fact as can be seen in figures 1 e and 1 f, the network approximation utilising the parameters obtained after 100 seconds of composite adaptation is superior to the direct adaptive controller, the approximation error being well within the range  $\pm 0.08$  for most values of  $\mathbf{x}$  inside  $\chi_n^-$ .

Finally a third trial was performed using a time-varying gain matrix  $\mathbf{P}$ , corresponding to least-squares estimation. The results are shown in figures 2 a, 2 b. Stability and asymptotic convergence of the tracking error are clearly seen in figure 2 a. The error converges to within  $\pm 0.028$  after 100 seconds, so that although the convergence rate is slower than for the gradient descent case, it is better than for the direct adaptive law. The network approximation error is of the same order as that for the gradient descent case as seen in figure 2 b.

## 7 Conclusions

A stable adaptive control scheme based on neural networks and using composite adaptation laws has been presented for the affine class of nonlinear systems. Sliding control as in [10, 15] and deadzone adaptation as in [8] were included to ensure global stability and robustness to the presence of the disturbance term arising from the network approximation errors. Two approaches were considered for the estimation part of the control law: gradient descent and least-squares estimation. In all cases the system exhibited

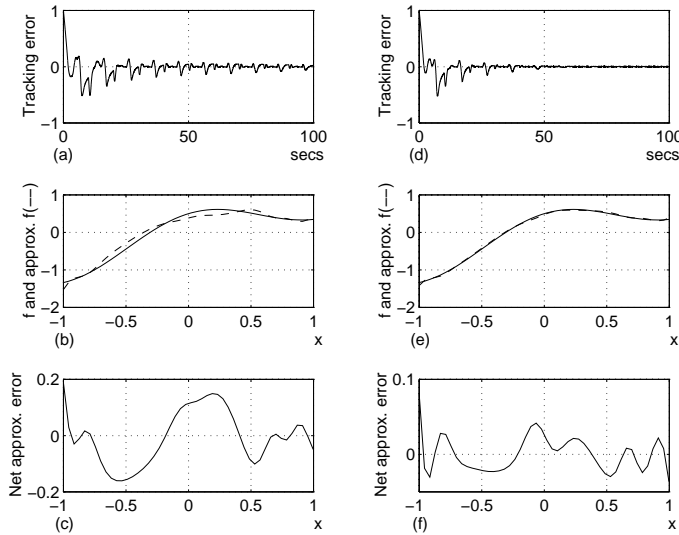


Figure 1: Direct and composite adaptation

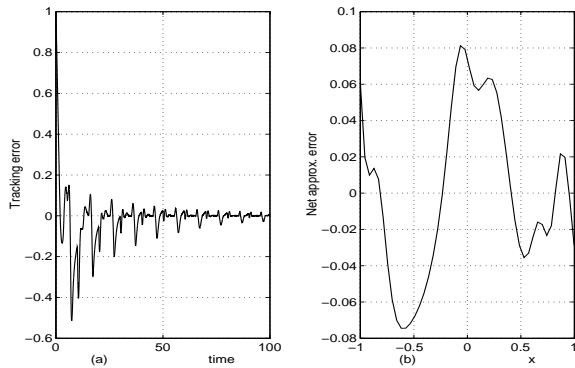


Figure 2: Composite adaptation using least-squares

improved performance over the direct adaptive controller, because the tracking error converges to zero much faster. The network approximation to the unknown system dynamics was also faster and of a better quality.

## References

- [1] K. J. Astrom and B. Wittenmark, *Adaptive Control*, Addison-Wesley, Reading MA, (1989).
- [2] M. A. Duarte and K. S. Narendra, "Combined direct and indirect approach to adaptive control", *IEEE Transactions on Automatic Control*, **Vol.34**, **No.10**, pp.1071-1075, (1989).
- [3] S. Fabri and V. Kadirkamanathan, "Dynamic structure neural networks for stable adaptive control of nonlinear systems", *IEEE Transactions on Neural Networks*, **Vol.7**, **No.5**, pp.1151-1167, (1996).
- [4] A. Isidori. *Nonlinear Control Systems: An Introduction*. Springer-Verlag, Berlin, (1989).
- [5] V. Kadirkamanathan and S. Fabri. "Stable nonlinear adaptive control with growing radial basis function networks" *Preprints of the IFAC Symposium for Control and Signal Processing*, Hungary, pp.231-236, (1995).
- [6] K. S. Narendra and A. M. Annaswamy. *Stable Adaptive Systems*. Prentice-Hall, Englewood Cliffs, New Jersey, (1989).
- [7] J. Park and I. W. Sandberg. "Universal approximation using radial basis function networks", *Neural Computation*, **Vol.3**, pp.246-257, (1990).
- [8] B. B. Peterson and K. S. Narendra. "Bounded Error Adaptive Control", *IEEE Transactions on Automatic Control*, **Vol.AC-27**, **No.6**, pp.1161-1168, (1982).
- [9] T. Poggio and F. Girosi. "Networks for Approximation and Learning", *Proceedings of the IEEE*, **Vol.78**, **No.9**, pp.1481-1497, (1990).
- [10] R. M. Sanner and J. J. E. Slotine. "Gaussian Networks for Direct Adaptive Control", *IEEE Transactions on Neural Networks*, **Vol.3**, **No.6**, pp.837-863, (1992).
- [11] S. Sastry and A. Isidori. "Adaptive Control of Linearizable Systems", *IEEE Transactions on Automatic Control*, **Vol.34**, **No.11**, pp.1123-1131, (1989).
- [12] J. J. E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice-Hall International, Englewood Cliffs, New Jersey, (1991).
- [13] J. J. E. Slotine and W. Li. "Composite Adaptive Control of Robot Manipulators", *Automatica*, **Vol.25**, **No.4**, pp.508-519, (1989).
- [14] E. Tzirkel-Hancock and F. Fallside. "Stable Control of Nonlinear Systems using Neural Networks", *International Journal of Robust and Nonlinear Control*, **Vol.2**, pp.63-86, (1992).
- [15] E. Tzirkel-Hancock. *Stable Control of Nonlinear Systems using Neural Networks*. PhD Thesis, Cambridge University Engineering Department, Cambridge, England, (1992).
- [16] V. I. Utkin. "Sliding mode and its applications to variable structure systems", *IEEE Transactions on Automatic Control*, **Vol.22**, **No.2**, pp.212-222, (1977).