Monograph:

Fabri, S. and Kadirkamanathan, V. (1996) Nonlinear Adaptive Control Using Gaussian Networks with Composite Adaptation for Improved Convergence. Research Report. ACSE Research Report 4653 . Department of Automatic Control and Systems Engineering

NONLINEAR ADAPTIVE CONTROL USING GAUSSIAN NETWORKS WITH COMPOSITE ADAPTATION FOR IMPROVED CONVERGENCE

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. **653**

Department of Automatic Control & Systems Engineering
The University of Sheffield

RERUM COGNOSCERE CAUSAS

October 1996

Nonlinear Adaptive Control Using Gaussian Networks with Composite Adaptation for Improved Convergence

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

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, combining techniques used in direct and indirect adaptive control. This is motivated by the need to improve the speed of convergence of the unknown parameters, hence resulting in a better system performance. The inherent approximation error of the neural networks might lead to instability because of parameter drift. This is compensated for by augmenting the control law with a low gain sliding mode component and using deadzone adaptation 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 and comparison with a direct adaptive control scheme.

1 INTRODUCTION

Neural networks have emerged as an important methodology for controlling nonlinear systems whose dynamics are either unknown or not accurately known [1, 11, 13, 14, 18, 19, 21, 25, 29, 30, 35]. The networks used are the multilayer perceptron using sigmoidal units [4, 27, 28] and the radial basis function network using Gaussian units [3, 24].

The presence of neural networks entails the use of a training mechanism by which the network learns some unknown functional related to the system, thus enabling the generation of appropriate control signals. Training involves the adjustment of the network parameters according to some signals in the system, preferably in real-time. Conventional adaptive control theory [2, 7, 8, 20] has provided a good framework based on the indirect or direct adaptive control methodology that has led to development of on-line training methods when neural networks are used for control. The indirect method [2] is based on system identification and uses the difference between some system variable and an approximation to it generated from an identification model. The difference, referred to as the *estimation error*, is used to vary the parameters of the neural network. The network approximation is then used in the control law on a certainty equivalence basis [8]. On the other hand, direct adaptive control [12] relies on the use of some norm of the difference between the system controlled output and the desired output, referred to as the tracking error, to adjust the parameters of the neural network directly in such a way as to ensure stability or optimize some cost function. Hence, direct adaptive schemes do not require an identification model.

As noted in [20] and [33] both the tracking and estimation errors contain relevant information about the system. Hence, in adaptive control, it makes sense to combine the direct and indirect approaches into one control law, resulting in *composite adaptive* control. This approach has been investigated in [5] and [34] for adaptive control systems that are linearly parameterized with unknown parameters and it has resulted in improved performance over conventional direct adaptive control schemes.

In this paper we shall consider adaptive control of a particular class of nonlinear systems when their dynamics are unknown. Radial basis function neural networks shall be used to learn the unknown system functions. Indirect adaptive control approaches for such systems result in network parameter adjustment rules that are closely related to neural network identification techniques [16, 17], since parameter adaptation is driven by a prediction or an estimation error. On the other hand, direct adaptive control methods for similar systems [6, 10, 15, 25, 30, 35] have network parameter adjustment rules derived in a manner similar to Lyapunov-based model reference adaptive control (MRAC), so as to ensure convergence of the tracking error. This work shall extend the direct adaptive control law for these systems to a composite law where *both* tracking and estimation errors shall be used for network parameter adaptation, within the framework of Lyapunov stability. The method is based on the composite adaptive schemes described in [34] for robot manipulators. However, while the latter did not require the use of neural networks because the unknowns consisted of system *param-*

eters, we shall consider a more general class of nonlinear systems that needs to be controlled adaptively by neural networks because the *functions* representing the nonlinear dynamics are unknown. This way we would be able to compare the performance of the system utilising a composite law for neural network parameter adjustment with that resulting from the use of a direct adaptive law.

2 STATEMENT OF THE PROBLEM

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:

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

where $\mathbf{x} \in \Re^n$ 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 [9],[31],[33].

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 [9], [31], [32], 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 $y_d, y_d^{(1)}, \ldots, y_d^{(r)}$ are bounded, where $y_d^{(i)}$ represents the ith derivative with respect to time of y_d.
- the control law generating $u(t)$ must ensure that the output and its $(r-1)$ derivatives $y, y^{(1)}, \ldots, y^{(r-1)}$ 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 $u(t) = \frac{-\hat{f}(\mathbf{x})+v(t)}{\hat{g}(\mathbf{x})} + u_{sl}(t)$, as in references [30], [35]. 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 [24] are used to approximate the nonlinear functions $f(\mathbf{x})$, $g(\mathbf{x})$ within a compact set $\chi_n \subset \Re^n$, where the state vector $\mathbf{x}(t)$ is known to be contained. χ_n thus represents the network approximation region. The output of the neural networks is given by

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

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

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

where $\mathbf{m}_{f_i}$, $\mathbf{m}_{g_i}$ are the coordinates of the centre of the *ith* 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 [30, 35] 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 χ_n because the network is not able to approximate correctly outside the approximation region. Within χ_n^-, a slightly smaller subset of the network approximation region χ_n, $\kappa(\mathbf{x}) = 1$ so that the full network approximation is utilised. In the boundary between χ_n^- and χ_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 χ_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 [20], [33]. However, Gaussian radial basis function networks satisfy the Universal Approximation property [22, 26], stating that given any uniform bounds ϵ_f, ϵ_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 $\overline{f}_o$ and $\overline{g}_o$ respectively.

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

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

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 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. Reference [30] provides methods for determining χ_n, the mesh spacing and the variance that will satisfy any desired ϵ_f, ϵ_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 Δ_f and Δ_g we obtain that $\forall \mathbf{x}$:

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

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 [6, 10], [35] namely

$$u(t) = u_{al}(t) + u_{sl}(t) \tag{5}$$

where

- $u_{al}(t) = \frac{-\hat{f}(\mathbf{x}) + v(t)}{\hat{g}(\mathbf{x})}$ is inspired from feedback linearization control laws [9], [33], with $v(t) = y_d^{(r)} - \alpha_r e^{(r-1)} - \ldots - \alpha_1 e$ representing an auxiliary input whose coefficients α_i are chosen such as to form a Hurwitz polynomial $\Gamma(s) = s^r + \alpha_r s^{r-1} + \ldots + \alpha_1$ in terms of the Laplace variable s.

- $u_{sl}(t)$ represents a sliding mode component introduced as in [30], [35] so as to ensure global stability if and when the state moves outside χ_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:

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

where $d(t) = -\Delta_f - \Delta_g u_{al}$. Note that the non-zero network approximation errors Δ_f and Δ_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 [30], [35] the filtered tracking error $e_1 = \Psi e$ defines a suitable sliding surface [37, 38], obtained from equation (6) as

$$\dot{e}_1 = -k_d e_1 - \kappa \tilde{\mathbf{w}}_f^T \mathbf{\Phi}_f - \kappa \tilde{\mathbf{w}}_g^T \mathbf{\Phi}_g u_{al} + g u_{sl} + d(t) \tag{7}$$

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 [33], whereby one seeks a linear relation between the unknown network parameters and some measureable signal. For this case, a linearly parameterized relation is developed as follows:

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

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

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 $A_o(s) = a_o + a_1 s + \ldots + a_{r-1}s^{r-1} + s^r$ is a Hurwitz and monic polynomial, reflecting stable filtration. This filtration results in the following dynamics

$$\begin{aligned} y(\frac{s^r}{A_o}) - \frac{f_o}{A_o} - \frac{g_o u(t)}{A_o} \quad + \quad \frac{\Delta_f}{A_o} + \frac{\Delta_g u}{A_o} = \\ \frac{\mathbf{\Phi}_f^T(\mathbf{x})\mathbf{w}_f^*}{A_o} \quad + \quad \frac{\mathbf{\Phi}_g^T(\mathbf{x})\mathbf{w}_g^* u}{A_o} \end{aligned}$$

which can be re-written in linear form as

$$Y = \mathbf{p}^T \mathbf{w}^* \tag{8}$$

where

$$\mathbf{p}^T = \left| \begin{array}{cc} \frac{\mathbf{\Phi}_f^T(\mathbf{x})}{A_o} & \frac{\mathbf{\Phi}_g^T(\mathbf{x})u}{A_o} \end{array} \right|, \ \mathbf{w}^* = \left| \begin{array}{c} \mathbf{w}_f^* \\ \mathbf{w}_g^* \end{array} \right|,$$
$$Y = \ y(\tfrac{s^r}{A_o}) - \tfrac{f_o}{A_o} - \tfrac{g_o u}{A_o} + \tfrac{\Delta_f}{A_o} + \tfrac{\Delta_g u}{A_o}.$$

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

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

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:

$$\hat{Y} = \mathbf{p}^T \hat{\mathbf{w}} \tag{10}$$

where

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

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

$$\epsilon = \mathbf{p}^T \tilde{\mathbf{w}} - d_f \tag{11}$$

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 ϵ is generated from available signals $\hat{y}^{(r)}$ and y by stable filtration and without using differentiators, as depicted in figure 1

3.4 The Adaptation Law

The composite adaptation law proposed has the form

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

where

- $\epsilon_\Delta = \epsilon$ if $|\epsilon| > \overline{d_f}$ and zero otherwise, represents a deadzone function on $\epsilon(t)$.

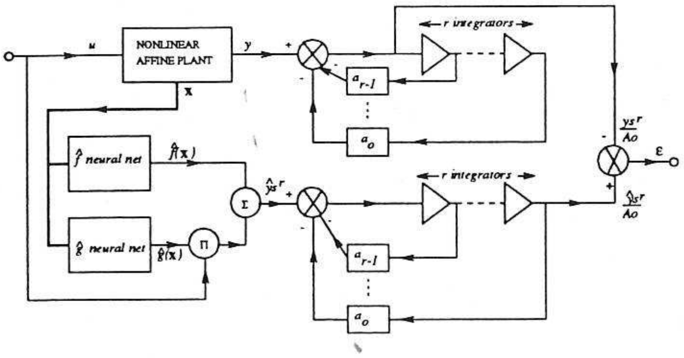

Figure 1: Generation of the estimation error.

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

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 [23] 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 $0 < g_l(\mathbf{x}) \leq g_o(\mathbf{x}) \; \forall \mathbf{x} \in \chi_n^-$. Methods of implementing this can be found in [35], [36].

Figure 2 shows the complete representation of the system in block diagram form.

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 [34].

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 [2, 20]. Various

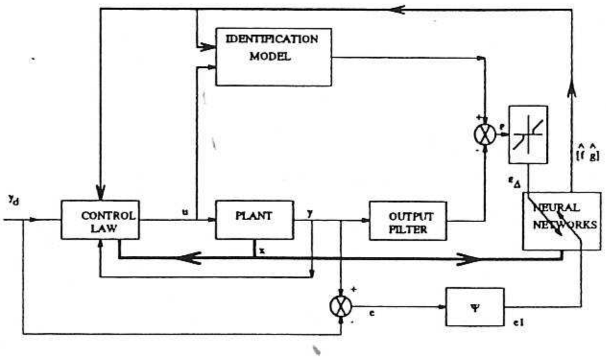

Figure 2: Composite adaptive control using neural networks.

parameter estimation techniques that involve time-varying gains have been developed [20, 33, 34]. We have utilised the standard least-squares algorithm, resulting in the following gain update law:

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

where $k_\Delta = k_e$ if $|\epsilon| > \overline{d_f}$ 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

$$V = \frac{1}{2}(e_1^2 + \tilde{\mathbf{w}}^T\mathbf{P}^{-1}(t)\tilde{\mathbf{w}}) \tag{14}$$

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

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

Choosing the sliding mode component of the control law as

$$u_{sl}(t) = -\frac{\overline{d}(\mathbf{x})}{g_l(\mathbf{x})} sign(e_1) \tag{16}$$

where $\overline{d}(\mathbf{x})$ is a known bound on the disturbance $d(t)$ (*i.e.* $|d(t)| \leq \overline{d}(\mathbf{x})$) ensures that the term $e_1(gu_{sl} + d)$ 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 $|\epsilon| \leq \overline{d}_f$ or

$k_e\mathbf{p}^T\tilde{\mathbf{w}}(\mathbf{p}^T\tilde{\mathbf{w}} - d_f)$ when $|\epsilon| > \overline{d}_f$. Since $\overline{d}_f$ represents a bound on d_f, the latter term is semi-positive definite [23], so that in general, $-k_e\mathbf{p}^T\tilde{\mathbf{w}}\epsilon_\Delta \le 0$. Hence, $\dot{V} \le -k_d e_1^2 \le 0$.

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

$$\dot{V} - = k_d e_1^2 + e_1(gu_{sl} + d) - k_e\mathbf{p}^T\tilde{\mathbf{w}}(\frac{\mathbf{p}^T\tilde{\mathbf{w}}}{2} - d_f) \qquad (17)$$

Since $k_e\mathbf{p}^T\tilde{\mathbf{w}}(\frac{\mathbf{p}^T\tilde{\mathbf{w}}}{2} - d_f) \ge 0$ for $|\epsilon| > \overline{d}_f$, then choosing u_{sl} as shown before in equation (16) ensures that $\forall\epsilon$, $\dot{V} \le -k_d e_1^2 \le 0$.

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 [9], 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_{al} is bounded, from which follows boundedness of $d(t)$. These imply that $u_{sl}(t)$ 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 $\dot{V} \le -k_d e_1^2$ and defining a function $V_1(t) = V(t) - \int_o^t \dot{V}(\tau) + k_d e_1^2 \, d\tau$, it follows that $V_1(t) \ge 0$ and $\dot{V}_1(t) = -k_d e_1^2$, the latter implying that $\dot{V}_1$ is semi negative definite and uniformly continuous. Hence Barbalat's Lemma [33] implies that $\dot{V}_1(t) \to 0$ as $t \to \infty$, 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 [34] as desired.

5 DETERMINATION OF THE DISTURBANCE BOUNDS

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

From equation (3) and the definition of $d(t)$ it follows that $\forall\mathbf{x}$

$$|d(\mathbf{x})| \le \kappa(\epsilon_f + \epsilon_g|u_{al}|) + (1-\kappa)(\overline{f_o} + \overline{g_o}|u_{al}|) \qquad (18)$$

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 $\frac{-\overline{d}(\mathbf{x})}{g_l(\mathbf{x})}$, 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 χ_n, taking over control so as to pull $\mathbf{x}$ inside χ_n if and when the state goes outside the network approximation region, as in references [30], [35].

From the definition of d_f, it follows that

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

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

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

so that

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

$\overline{d}_1$ thus represents a bound on the term $(d(t) - \Delta_g u_{sl})$. But equation (19) shows that $d_f(t)$ is a stable filtration of $(d(t) - \Delta_g u_{sl})$ 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 [33], $|d_f^{(i)}| \leq 2^i \lambda^{i-r} \overline{d}_1$ for $0 \leq i \leq r$, so that

$$\overline{d}_f = \lambda^{-r} \overline{d}_1 \tag{21}$$

6 SIMULATION RESULTS

Two simulations were carried out to check the performance of the composite adaptive system.

6.1 Simulation 1

The first simulation was done on the nonlinear affine plant used in reference [36]:

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

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

The system is of order $n = 1$ and degree $r = 1$. Its internal dynamics satisfy the appropriate Lipschitz conditions and the zero dynamics are globally exponentially stable. 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 $f_o = 0$. As shown in [36], 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 χ_n. Assuming that it is known that $f(x)$ 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 $\lambda = 5$.

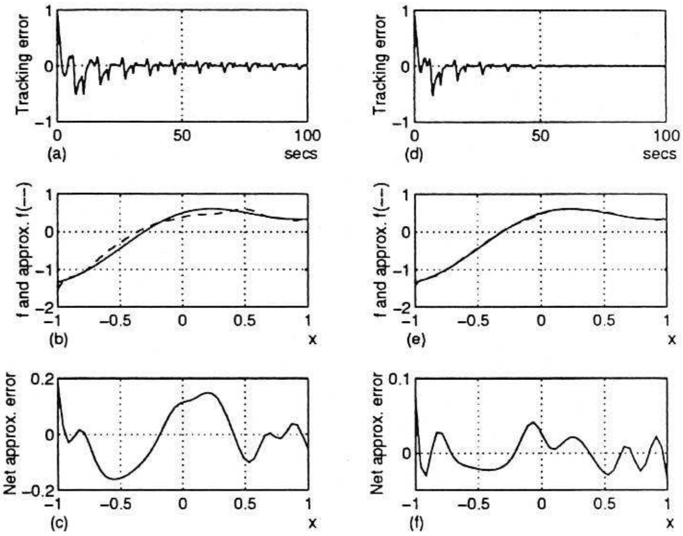

Figure 3: Simulation 1:direct and composite adaptation

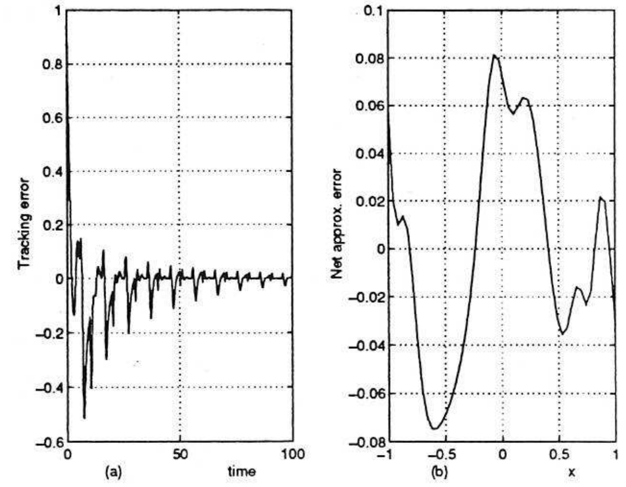

Figure 4: Simulation 1:composite adaptation using least-squares

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 3 a-c. Figure 3 a shows the first 100 seconds of simulation. The system is stable with the tracking error asymptotically converging to zero. Figure 3 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 3 c that the network approximation error $(f(\mathbf{x}) - \hat{f}(\mathbf{x}))$ lies within the range ±0.18 for the values of $\mathbf{x}$ inside χ_n^-. The results of the composite adaptive controller are shown in figures 3 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 ±0.0058 compared with ±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 3 e and 3 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 ±0.08 for most values of $\mathbf{x}$ inside χ_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 4 a, 4 b.Stability and asymptotic convergence of the tracking error are clearly seen in figure 4 a. The error converges to within ±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 4 b.

6.2 Simulation2

The second example is taken from [35]. The plant of this example is given by the dynamics:

$$\begin{aligned}\dot{x}_1 &= f(\mathbf{x}) + g(\mathbf{x})u \\ \dot{x}_2 &= x_1 - x_2 \\ y &= x_1\end{aligned}$$

where

$$\begin{aligned}\mathbf{x} &= [x_1\ x_2]^T \\ f(\mathbf{x}) &= cos(7(x_1^2 + x_2^2))\exp(-(x_1^2 + x_2^2)) \\ g(\mathbf{x}) &= 2 + cos(7x_1x_2)\end{aligned}$$

This system is of order $n = 2$ and has degree $r = 1$. Its internal dynamics satisfy the appropriate Lipschitz conditions and the zero dynamics are globally exponentially stable.

The desired output, y_d, is obtained by filtering a zero-average, 0.9 amplitude, 0.4 Hz square wave by a filter $1/(1+0.1s)^3$, so that the derivative of the filter output is bounded. As shown in [35], for the given plant and reference input, the desired state is bounded well within the interval $[-1,\ 1] \times [-1,\ 1]$ (along x_1, x_2 respectively). To cater for the fact that during the transient period the actual state may overshoot these bounds, the network approximation region is taken to be larger, namely $\chi_n = [-1.5,\ 1.5] \times [-1.5,\ 1.5]$. This ensures that the state is bounded within the approximation region, thus avoiding the use of crude, high-gain sliding control. χ_n^-, by definition, must be a subset of χ_n so that it is set to $\chi_n^- = [-1,\ 1] \times [-1,\ 1]$. It is assumed that the known prior estimates to the functions being approximated are $f_o = 0$, $g_o = 2$ and as shown in [35], full network inherent approximation error bounds $\epsilon_f = \epsilon_g = 0.005$ would be obtained via basis functions having a standard deviation σ_f, σ_g of 0.03 located on a mesh of spacing $\mu = 0.05$ within χ_n. Bounds $\overline{f}_o$, $\overline{g}_o$ are set to 2 and $g_l(\mathbf{x})$ to 0.895. The control law is given by $v(t) = \dot{y}_d - 5(y - y_d)$ and $e_1(t) = e(t)$. The gain matrix was chosen as $\mathbf{P}_o = 25\mathbf{I}$ and maintained constant, corresponding to gradient descent estimation, and $A_o = (s+5)$.

Two simulations were performed: one with $k_e = 0$, corresponding to direct adaptation, and another with $k_e = 15$, corresponding to composite adaptation. The results are shown in figures 5 a and 5 b respectively. Once again, note that the system is stable and that the tracking error converges asymptotically to zero. However the composite law gives better performance than the direct law because the error converges much more rapidly. After 100 seconds, the tracking error for the composite law was well within ±0.0035, a factor of four better than that for the direct law.

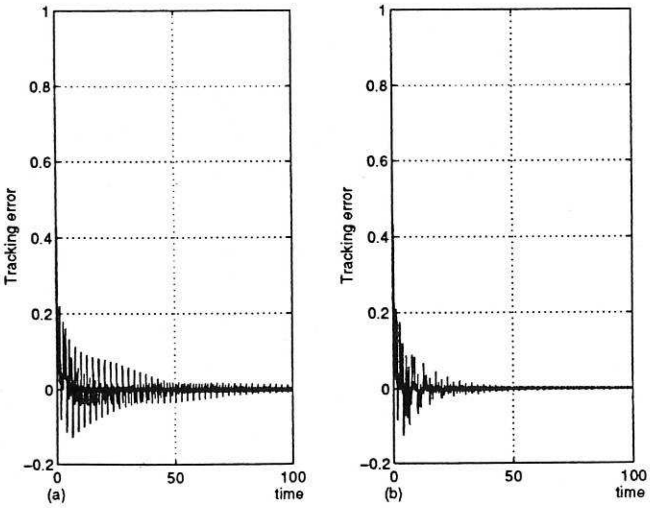

Figure 5: Results of simulation 2: (a) Direct adaptation (b) Composite adaptation

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 [30], [36] and deadzone adaptation as in [23] 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 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] P. J. Antsaklis (Ed), Special Issue on Neural Networks in Control Systems, *IEEE Control Systems Magazine*, Vol.10, No.3, April 1990.

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

[3] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks", *Complex Systems*, Vol.2, pp.321-355, 1988.

[4] G. Cybenko, "Approximations by superpositions of a sigmoidal function", *Mathematics of Control, Signals and Systems*, Vol.2, pp.303-314, 1989.

[5] M. A. Duarte and K. S. Narendra, "Combined direct and indirect adaptive control of plants with relative degree greater than one", *Automatica*, Vol.25, No.3, 1989.

[6] 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, 1996

[7] G. C. Goodwin and K. S. Sin, *Adaptive Filtering, Prediction and Control*, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

[8] I. Isermann, K. Lachmann and D. Matko, *Adaptive Control Systems*. Prentice-Hall International, Englewood Cliffs, New Jersey, 1991.

[9] A. Isidori. *Nonlinear Control Systems: An Introduction*. Springer-Verlag, Berlin, 1989.

[10] V. Kadirkamanathan and S. Fabri. "Stable nonlinear adaptive control with growing radial basis function networks" In *Preprints of the IFAC Symposium for Control and Signal Processing*, Budapest, Hungary, pp.231-236, 1995.

[11] A. Karakasoglu, S. L. Sudharsanan and M. K. Sundareshan, "Identification and decentralized adaptive control using neural networks with application to robotic manipulators", *IEEE Transactions on Neural Networks*, Vol.4, No.6, pp.919-930, 1993.

[12] Y. D. Landau. *Adaptive Control: The Model Reference Approach.* Marcel Dekker Inc., New York, 1979.

[13] A. Levin and K. S. Narendra, "Control of nonlinear dynamical systems using neural networks: Controllability and stabilization", *IEEE Transactions on Neural Networks*, Vol.4, No.2, pp.192-206, 1993.

[14] G. Lightbody and G. W. Irwin, "Direct neural model reference adaptive control", *IEE Proceedings-Control Theory Applications*, Vol.142, No.1, pp.31-43, 1995.

[15] G. P. Liu, V. Kadirkamanathan and S. A. Billings, "Variable neural networks for nonlinear adaptive control" in *Preprints of the 13th World Congress of the International Federation of Automatic Control*, San Francisco, California, USA, Vol.F, pp.181-186, 1996.

[16] G. P. Liu, V. Kadirkamanathan and S. A. Billings, "Neural network based predictive control for nonlinear systems", Research Report No.619, Department of Automatic Control & Systems Engineering, The University of Sheffield, Sheffield, UK, March 1996.

[17] G. P. Liu, V. Kadirkamanathan and S. A. Billings, "Stable sequential identification of continuous nonlinear dynamical systems by growing radial basis function networks", *International Journal of Control*, Vol.65, No.1,pp.53-69, 1996.

[18] W. T. Miller, R. S. Sutton and P. J. Werbos, (Eds), *Neural Networks for Control*, MIT Press, Cambridge, Massachusetts, 1990.

[19] S. Mukhopadhyay and K. S. Narendra, "Disturbance Rejection in nonlinear systems using neural networks", *IEEE Transactions on Neural Networks*, Vol.4, No.1, pp.63-72, 1993.

[20] K. S. Narendra and A. M. Annaswamy. *Stable Adaptive Systems.* Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[21] 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.

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

[23] B. B. Peterson and K. S. Narendra. "Bounded Error Adaptive Control", *IEEE Transactions on Automatic Control*, Vol. AC-27, No.6, December 1982.

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

[25] M. Polycarpou and P. Ioannou, "Identification and Control of Nonlinear Systems Using Neural Network Models:Design and Stability Analysis", Technical Report 91-09-01, Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, USA, September 1991.

[26] M. J. D. Powell, "Radial basis functions for multivariable interpolation: A review", In J. C. Mason and M. G. Cox (Eds.), *Algorithms for Approximation*, Oxford University Press, Oxford, pp.143-167, 1987.

[27] D. Psaltis, A. Sideris and A. A. Yamamura, "A multilayered neural network controller", *IEEE Control Systems Magazine*, Vol.8, pp.17-21, 1988.

[28] 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.

[29] N. Sadegh, "A perceptron network for functional identification and control of nonlinear systems", *IEEE Transactions on Neural Networks*, Vol.4, No.6, pp.982-988, 1993.

[30] 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.

[31] S. Sastry and M. Bodson. *Adaptive Control:Stability, Convergence and Robustness.* Prentice-Hall International, Englewood Cliffs, New Jersey 1989.

[32] S. Sastry and A. Isidori. "Adaptive Control of Linearizable Systems", *IEEE Transactions on Automatic Control*, Vol.34, No.11, November 1989.

[33] J. J. E. Slotine and W. Li. *Applied Nonlinear Control.* Prentice-Hall International, Englewood Cliffs, New Jersey 1991.

[34] J. J. E. Slotine and W. Li. "Composite Adaptive Control of Robot Manipulators", *Automatica*, Vol.25,No.4,pp. 508-519, 1989.

[35] 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, May 1992.

[36] E. Tzirkel-Hancock. *Stable Control of Nonlinear Systems using Neural Networks.* PhD Thesis, Cambridge University Engineering Department, Cambridge, England, June 1992.

[37] V. I. Utkin. *Sliding Modes and their Application in Variable Structure Systems.* MIR Publishers, Moscow, 1978.

[38] V. I. Utkin. "Sliding mode and its applications to variable structure systems", *IEEE Transactions on Automatic Control*, Vol.22, No.2, pp.212-222, April 1977.