Stable Control of Nonlinear Systems Using Neural Networks

E. Tzirkel-Hancock & F. Fallside

CUED/F-INFENG/TR.81

Cambridge University Engineering Department
Trumpington Street
Cambridge CB2 1PZ
England

July 1991

email: et@eng.cam.ac.uk

The first author is supported by Trinity College Cambridge and the Committee of Vice-Chancellors and Principals of the UK.

Abstract

A neural network based direct control architecture is presented, that achieves output tracking for a class of continuous time nonlinear plants, for which the nonlinearities are unknown. The controller employs neural networks to perform approximate input/output plant linearization. The network parameters are adapted according to a stability principle. The architecture is based on a modification of a method previously proposed by the authors, where the modification comprises adding a sliding control term to the controller. This modification serves two purposes: first, as suggested in [15], sliding control compensates for plant uncertainties outside the state region where the networks are used, thus providing global stability; second, the sliding control compensates for inherent network approximation errors, hence improving tracking performance.

A complete stability and tracking error convergence proof is given and the setting of the controller parameters is discussed. It is demonstrated that as a result of using sliding control, better use of the network's approximation ability can be achieved, and the asymptotic tracking error can be made dependent only on inherent network approximation errors and the frequency range of unmodeled dynamical modes. Two simulations are provided to demonstrate the features of the control method.

1 Introduction

The potential benefits of using neural network architectures to solve tracking control tasks that involve nonlinear plants with unknown nonlinearities was demonstrated by many researchers, e.g. [5] [7] [10] [12]. Many of the recently proposed architectures, are based on the adjustment of the controller parameters according to a gradient of some measure of the tracking error, with respect to the controller parameters [5] [7] [10] [12]. A drawback of these methods is the difficulty encountered in assuring their closed loop stability. This drawback is more severe if the controller parameters are adapted in an on-line manner. A similar difficulty, associated with gradient methods for parameter estimation, was encountered by researchers in the field of adaptive control three decades ago. It led to the development of stability theory based parameter estimation and control architectures, i.e. modern adaptive control theory.

It is our belief that the development of neural control architectures based on stability theory will result in more reliable controllers and therefore widen the application range of neural control methodology. Such development will also enhance understanding of neural, on-line, parameter estimation, in the context of closed loop dynamical systems, by providing a link to adaptive control theory. It was this belief that motivated the development of the neural control architecture proposed in [20], which is presented and further modified in this paper.

The architecture is designed to solve the tracking control task for the nonlinear plant given by:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{a}(\mathbf{x}) + \mathbf{b}(\mathbf{x})u \\ y &= h(\mathbf{x}) \end{aligned} \tag{1}$$

where $\mathbf{a}(\mathbf{x})$, $\mathbf{b}(\mathbf{x}) : \Re^n \rightarrow \Re^n$ and $h(\mathbf{x}) : \Re^n \rightarrow \Re$ are smooth functions of the state vector $\mathbf{x}(t) : \Re^+ \rightarrow \Re^n$, and $u(t)$, $y(t) : \Re^+ \rightarrow \Re$ are the plant's input and output respectively. The functions $\mathbf{a}(\mathbf{x})$, $\mathbf{b}(\mathbf{x})$ and $h(\mathbf{x})$ are assumed unknown but satisfy some structural assumptions. This type of plant which is linear in the control variable is encountered in many applications, e.g. rigid link robot manipulators.

The advantage gained by restricting attention to this type of nonlinear system is that the control theory of these systems is well developed [1] [4]. In particular, when the plant's nonlinearities are known, the tracking control task can be solved with the technique of exact input/output linearization, and conditions under which stability is guaranteed are well understood. Adaptive tracking control methods for this type of plant also exist and are well analysed and understood. These may be applied when the plant nonlinearities can be expressed in terms of known functions linearly scaled by unknown parameters [16] [19].

The proposed control architecture is based on approximate input/output linearization of the plant dynamics with the aid of neural networks, along the lines suggested in [20]. The main modification to the formulation proposed in [20] is the inclusion of sliding control. This modification serves two purposes. Firstly, as proposed in [15], the sliding control compensates for uncertainties in plant nonlinearities outside the state region in which the network approximation is used. This serves to assure global stability. Secondly, the sliding control compensates for inherent network approximation errors within the state region in which the networks are used. This allows one to reduce the asymptotic tracking error to a neighbourhood of zero, determined solely by the inherent network approximation errors and the frequency range of high frequency unmodeled dynamics. It also leads to a better use of the networks' approximation ability by allowing one to set an adaptation deadzone smaller than the conservative deadzone required if sliding control is not used.

Section 2 presents the plant structural assumptions and the control task. In section 3 the networks' role is discussed. Section 4 presents the control architecture and design, and provides a complete stability and convergence proof. In section 5 considerations of setting the different controller parameters, notably the boundary layer thickness, are discussed in detail. Finally, section 6 provides two simulations with example plants to illustrate the features of the control architecture.

2 Task Definition and Plant Assumptions

This section defines the control task and details the underlying structural assumptions that are required in order to construct the controller. It is assumed that the reader is familiar with the basic theory of input/output linearization of affine systems [4] [18].

Assumption 1 *The plant has a known strong relative degree γ, i.e. $L_b h(\mathbf{x}) = L_b L_a h(\mathbf{x}) = \cdots = L_b L_a^{\gamma-2} h(\mathbf{x}) = 0$ for all $\mathbf{x}$ and $L_b L_a^{\gamma-1} h(\mathbf{x})$ is bounded away from zero for all $\mathbf{x}$, where $L_b h(\mathbf{x})$ denote the Lie derivative of h along the vector field $\mathbf{b}$ and $L_a^r h(\mathbf{x}) = L_a(L_a^{r-1} h(\mathbf{x}))$.*

With this assumption, one can prove the existence of a state diffeomorphism $\mathbf{z} = D(\mathbf{x})$ that when applied, transforms the plant (1) into its *normal form* given, in terms of the new state variable $\mathbf{z}$, by

$$\begin{aligned} \dot{z}_{11} &= z_{12} \\ &\vdots \\ \dot{z}_{1\gamma-1} &= z_{1\gamma} \\ \dot{z}_{1\gamma} &= f_1(\mathbf{z}) + g_1(\mathbf{z})u = f(\mathbf{x}) + g(\mathbf{x})u \\ \dot{\mathbf{z}}_2 &= \mathbf{I}(\mathbf{z}_1, \mathbf{z}_2) \\ y &= z_{11} \end{aligned} \tag{2}$$

where $\mathbf{z}^T = [\mathbf{z}_1^T \ \mathbf{z}_2^T]$ is separated to its observable part $\mathbf{z}_1(t) : \Re^+ \to \Re^\gamma$, and its unobservable part $\mathbf{z}_2(t) : \Re^+ \to \Re^{n-\gamma}$, and where $f(\mathbf{x}) = L_a^\gamma h(\mathbf{x}) : \Re^n \to \Re$ and $g(\mathbf{x}) = L_b L_a^{\gamma-1} h(\mathbf{x}) : \Re^n \to \Re$ are Lie derivatives of the system nonlinearities. The (vector) differential equation $\dot{\mathbf{z}}_2 = \mathbf{I}(\mathbf{z}_1, \mathbf{z}_2)$ represents the system's internal dynamics.

Task definition The controller's task is to force the plant output y and output derivatives up to order $\gamma - 1$ to track a given desired output y_d and the corresponding desired output derivatives. The desired output and its derivatives up to order γ are assumed bounded.

Assumption 2
1) Without loss of generality assume that $g(\mathbf{x})$ is positive and that $0 < g_l \leq g_l(\mathbf{x}) \leq g(\mathbf{x}) \leq g_h(\mathbf{x})$ for all $\mathbf{x}$, where the state dependent bounds $g_l(\mathbf{x})$ and $g_h(\mathbf{x})$ are known, and the constant lower bound g_l is known.
2) $|f(\mathbf{x})| \leq f_h(\mathbf{x})$ for all $\mathbf{x}$, where $f_h(\mathbf{x})$ is a known bound.
3) The zero dynamics $\dot{\mathbf{z}}_2 = \mathbf{I}(0, \mathbf{z}_2)$ are globally exponentially minimum phase [4].
4) The function $\mathbf{I}(\mathbf{z}_1, \mathbf{z}_2)$ has continuous and bounded partial derivatives in both $\mathbf{z}_1$ and $\mathbf{z}_2$.
5) The state $\mathbf{x}$, the plant output, and the output derivatives up to order $\gamma - 1$ are available for measurement, i.e. $\mathbf{x}$ and $\mathbf{z}_1$ are measurable.

The required availability of the plant output derivatives up to order $\gamma - 1$ is restrictive. However, in a large number of applications the derivatives are given as state variables, e.g. many electromechanical systems and specifically robot manipulators.

It is worthwhile noticing that in the case where the strong relative degree equals the plant order, i.e. $\gamma = n$, there are no internal dynamics and the plant output and output derivatives, which are available by assumption, may be used as the system states. This is advantageous since the plant in this case is given directly in normal form and the desired phase portrait is given. It will therefore be implicitly assumed in this paper that if $\gamma = n$, the output and its derivatives are used as state variables.

In the case where $\gamma < n$ the following proposition is useful for proving closed loop stability:

Proposition 1 *If assumption 2 (items 3 and 4) hold and if* $\mathbf{z}_1$ *is kept bounded by some means, then* $\mathbf{z}_2$ *is bounded and* $\mathbf{x}$ *is bounded.*

According to this proposition, it is sufficient to establish the boundedness of $\mathbf{z}_1$ in order to guarantee the boundedness of $\mathbf{z}$ and $\mathbf{x}$. The proof of proposition (1) can be found in [16].

3 The Network Approximation

The control architecture presented in this paper employs neural networks to approximate the Lie derivatives $f(\mathbf{x})$ and $g(\mathbf{x})$, defined in equation (2), in a state region denoted by $\mathcal{X}_n$. The general network structure is in the form $\sum_{i=1}^{p} \theta_i w_i(\mathbf{x})$ where $\boldsymbol{\theta}$ is a parameter vector and $\mathbf{w}(\mathbf{x})$ a vector of basis functions. Any type of basis functions may be used for the approximation, e.g. radial basis functions (RBF) [14], polynomial splines [6], volterra polynomials [8], etc. It is highly desirable to incorporate as much prior knowledge of the functions $f(\mathbf{x})$ and $g(\mathbf{x})$ into the network structure. For instance, the prior knowledge may be expressed in terms of the degree of smoothness of $f(\mathbf{x})$ and $g(\mathbf{x})$, or may include knowledge of basis functions that explicitly appear in the Lie derivatives, multiplied by unknown parameters.

The design of the networks involves the choice of basis function vectors such that the minimal network approximation errors in the state region $\mathcal{X}_n$ are uniformly bounded by specified bounds. The basis function vectors are designed to meet the specifications given in the next assumption. The assumption also defines the optimal (unknown) parameter vectors, that are to be estimated in the learning process.

Assumption 3 *Given the uniform approximation bounds* ϵ_f *and* ϵ_g*, and the prior estimates* $f_0(\mathbf{x})$, $g_0(\mathbf{x}) : \Re^n \to \Re$ *of* f *and* g *respectively, there exist basis function vectors* $\mathbf{w}_f(\mathbf{x})$, $\mathbf{w}_g(\mathbf{x}) : \Re^n \to \Re^p$ *and "optimal" parameter vectors* $\boldsymbol{\theta}_f \in \Re^p$ *and* $\boldsymbol{\theta}_g \in \Re^p$ *such that for all* $\mathbf{x} \in \mathcal{X}_n$ *the following inequalities are satisfied:*

$$\begin{aligned} \left| f(\mathbf{x}) - f_0(\mathbf{x}) - \boldsymbol{\theta}_f^T \mathbf{w}_f(\mathbf{x}) \right| &\leq \epsilon_f \\ \left| g(\mathbf{x}) - g_0(\mathbf{x}) - \boldsymbol{\theta}_g^T \mathbf{w}_g(\mathbf{x}) \right| &\leq \epsilon_g \end{aligned} \tag{3}$$

The ability of neural networks to uniformly approximate smooth functions is well known [2] [3]. The approximation theory for RBF networks in particular has been recently extensively investigated [9] [13] [14]. Practical design methods for radial basis function networks in which the basis functions are located on a regular grid have also been developed. One such method, based on multi dimensional sampling theory, was proposed in [15] and is particularly appealing from the engineering point of view. This method specifies a design of an RBF network with gaussian nodes, given the smoothness of the function to be approximated, expressed in terms of its spatial fourier transform, and a desired uniform approximation bound. In practice, when not much is known about the underlying nonlinearities, one must construct a network empirically, and assume certain, typically conservative, approximation error bounds.

In order to provide smooth function approximation over the entire state space, a state region $\mathcal{X}_n^- \subset \mathcal{X}_n$ is defined. This region is a subset of $\mathcal{X}_n$ and is slightly smaller than $\mathcal{X}_n$. The "optimal" approximations of the functions $f(\mathbf{x})$ and $g(\mathbf{x})$ are then defined by:

$$
\begin{aligned}
f(\mathbf{x}, \boldsymbol{\theta}_f) &= f_0(\mathbf{x}) + m(\mathbf{x})\boldsymbol{\theta}_f^T \mathbf{w}_f(\mathbf{x}) \\
g(\mathbf{x}, \boldsymbol{\theta}_g) &= g_0(\mathbf{x}) + m(\mathbf{x})\boldsymbol{\theta}_g^T \mathbf{w}_g(\mathbf{x})
\end{aligned}
\tag{4}
$$

where $m(\mathbf{x})$ is a state dependent modulation function that takes on the value $m(\mathbf{x}) = 1$ on $\mathbf{x} \in \mathcal{X}_n^-$ and the value $m(\mathbf{x}) = 0$ outside the state region $\mathcal{X}_n$. On the narrow transition region $\mathcal{X}_n - \mathcal{X}_n^-$ $m(\mathbf{x})$ satisfies $0 \le m(\mathbf{x}) \le 1$ and is decreasing from the value 1 to the value 0 as the state moves from the boundary of $\mathcal{X}_n^-$ to the boundary of $\mathcal{X}_n$ [15].

The *inherent approximation errors* of the functions $f(\mathbf{x})$ and $g(\mathbf{x})$ are defined by

$$
\begin{aligned}
\triangle f(\mathbf{x}) &= f(\mathbf{x}) - f(\mathbf{x}, \boldsymbol{\theta}_f) \\
\triangle g(\mathbf{x}) &= g(\mathbf{x}) - g(\mathbf{x}, \boldsymbol{\theta}_g)
\end{aligned}
\tag{5}
$$

From assumption 2 and the definition of the optimal network approximations (4), it follows that the inherent approximation errors $\triangle f(\mathbf{x})$ and $\triangle g(\mathbf{x})$ can be bounded by known functions of the state, that is:

$$
|\triangle f(\mathbf{x})| \le D_f(\mathbf{x}) \quad \text{and} \quad |\triangle g(\mathbf{x})| \le D_g(\mathbf{x}) \tag{6}
$$

It can be easily verified that the bounds $D_f(\mathbf{x})$ and $D_g(\mathbf{x})$ can be chosen such that

$$
\begin{aligned}
D_f(\mathbf{x}) &\le (1-m)\bar{f}_0(\mathbf{x}) + m\epsilon_f \\
D_g(\mathbf{x}) &\le (1-m)\bar{g}_0(\mathbf{x}) + m\epsilon_g
\end{aligned}
\tag{7}
$$

where $\bar{f}_0(\mathbf{x})$ and $\bar{g}_0(\mathbf{x})$ are bounds of the crude approximation errors $|f - f_0|$ and $|g - g_0|$ respectively. If the only information available about the network approximation are the uniform bounds ϵ_f and ϵ_g, then the right hand side of inequality (7) can be used to define the inherent approximation error bounds $D_f(\mathbf{x})$ and $D_g(\mathbf{x})$. However, if the networks are known to perform better in some state regions than others then $D_f(\mathbf{x})$ and $D_g(\mathbf{x})$ can be made state dependent within the region $\mathbf{x} \in \mathcal{X}_n^-$ in accordance with this knowledge, thus making a better use of available information.

The fixed optimal parameters $\boldsymbol{\theta}_f$ and $\boldsymbol{\theta}_g$ are estimated by the time varying parameters $\hat{\boldsymbol{\theta}}_f(t)$ and $\hat{\boldsymbol{\theta}}_g(t)$ that are adapted during learning. The network approximations associated with these parameters are denoted by $\hat{f} = f(\mathbf{x}, \hat{\boldsymbol{\theta}}_f)$ and $\hat{g} = g(\mathbf{x}, \hat{\boldsymbol{\theta}}_g)$ respectively. Defining the parameter errors $\tilde{\boldsymbol{\theta}}_f(t) = \hat{\boldsymbol{\theta}}_f(t) - \boldsymbol{\theta}_f$ and $\tilde{\boldsymbol{\theta}}_g(t) = \hat{\boldsymbol{\theta}}_g(t) - \boldsymbol{\theta}_g$, one obtains the following equation for the network approximation error:

$$
\begin{aligned}
\hat{f} - f &= m\tilde{\boldsymbol{\theta}}_f^T \mathbf{w}_f - \triangle f \\
\hat{g} - g &= m\tilde{\boldsymbol{\theta}}_g^T \mathbf{w}_g - \triangle g
\end{aligned}
\tag{8}
$$

Having defined the network approximation method, one needs to consider the choice of the approximation region $\mathcal{X}_n$. Consider first the simpler case where $\gamma = n$.

The desired state trajectory is then given by the specified desired output and desired output derivatives. Denote by $\mathcal{X}_d$ a state region in which the desired trajectory lies. A sensible choice for $\mathcal{X}_n^-$ is one that satisfies $\mathcal{X}_n^- \supset \mathcal{X}_d$. This choice ensures that if output tracking is achieved, the entire state trajectory is bounded in the set $\mathcal{X}_n^-$ on which the network approximations are used.

In the more general case where $\gamma < n$, the situation is complicated by the presence of internal dynamics. If an output tracking constraint is imposed, the state trajectory is not fully determined, but is guaranteed by proposition (1) to lie in some bounded state region. Denote this state region by $\mathcal{X}_d$. In order to design the networks, one has to use an estimate $\mathcal{X}_{de}$ of $\mathcal{X}_d$, which can probably be obtained by some physical insight, and then choose $\mathcal{X}_n^- \supset \mathcal{X}_{de}$. The region $\mathcal{X}_{de}$ can then be adjusted empirically by observing the actual system's trajectory. This may be followed by an adjustment of $\mathcal{X}_n$. In the next section it will be demonstrated that output tracking is achieved even if $\mathcal{X}_d$ is not a subset of $\mathcal{X}_n^-$ (by sliding control). This fact makes the empirical adjustment of $\mathcal{X}_{de}$ possible.

In some situations one may be satisfied with using the network approximations on a subset of $\mathcal{X}_d$ only, letting sliding control compensate for initial approximation errors on $\mathcal{X}_d - \mathcal{X}_n^-$. This is a valid mode of operation of the proposed controller.

4 The Controller Architecture and Design

This section introduces the controller architecture and design, and provides a stability and tracking error convergence proof. The controller is presented as a modification of the design introduced in [20].

From the normal form (2), the plant input/output relation may be written in the form

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

If the nonlinearities f and g are known, one may construct a control law of the form $u = (v - f)/g$ that yields a linear input/output relation $y^{(\gamma)} = v$ between the new control v and the output y. Standard linear controller design techniques may then be applied to achieve asymptotic output tracking.

Our approach is to use a similar control law with f and g replaced by their network approximations $\hat{f}$ and $\hat{g}$ respectively. The auxiliary input v is chosen as

$$v = y_d^{(\gamma)} - \alpha_\gamma e^{(\gamma-1)} - \cdots - \alpha_1 e \tag{10}$$

with $\Gamma(s) = s^\gamma + \alpha_\gamma s^{\gamma-1} + \cdots + \alpha_1$ a Hurwitz polynomial, i.e. , a polynomial that has all its roots in the open left half of the complex plane, and $e = y - y_d$ is the output tracking error. The variable s denotes the Laplace variable. This strategy leads to an error equation of the form

$$e = \Gamma(s)^{-1}\left(\tilde{\boldsymbol{\theta}}^T \mathbf{w}(t) + d(t)\right) \tag{11}$$

where $\tilde{\boldsymbol{\theta}}$ is a parameter error vector, $\mathbf{w}(t)$ a vector of basis functions and $d(t)$ a disturbance due to inherent network approximation errors.

In order to construct stable adaptation laws for the network parameters, the following error measure is introduced in [20],

$$e_s = \beta_\gamma e^{(\gamma-1)} + \cdots + \beta_1 e = \Psi(s)\, e \tag{12}$$

where $\Psi(s)$ is a Hurwitz polynomial. The error e_s is a compact measure of the output tracking error and the tracking error derivatives. In particular, as $\Psi^{-1}(s)$ is a proper, stable transfer function from e_s to e, the boundedness of e_s implies the asymptotic boundedness of the tracking error and its derivatives up to order $\gamma - 1$.

In terms of the error measure e_s the error equation (11) may be written as $e_s = \Psi(s)\Gamma^{-1}(s)(\tilde{\boldsymbol{\theta}}^T \mathbf{w}(t) + d(t))$. The polynomials $\Psi(s)$ and $\Gamma(s)$ are chosen in such a way that this error equation is strictly positive real (SPR) [11] [16]. This choice of the transfer function $\Psi(s)\Gamma^{-1}(s)$ enables the construction of stable parameter adaptation laws using the error measure e_s [20].

One difficulty with this architecture is that it is based on the assumption that the state $\mathbf{x}$ remains bounded within the state region in which the network approximation is used. Therefore, global stability is not guaranteed. This difficulty can be alleviated by incorporating a crude mechanism that forces output tracking outside the state region in which network approximation is used. The use of sliding control for this purpose was introduced in [15] and is adopted here. Another practical problem of the above architecture is that in order to ensure stability one must use deadzone adaptation in which the deadzone width is typically extremely conservative, thus making poor use of information that is available for network training. This problem can be alleviated by using sliding control to compensate for inherent network approximation errors.

The error measure e_s may conveniently be used to define a sliding surface. However, in order to incorporate sliding control into the architecture, the transfer function $\Psi(s)\Gamma^{-1}(s)$ must be made to have a unity order. That is, the zeroes of $\Psi(s)$ must all be cancelled by poles of $\Gamma(s)^{-1}$. Therefore, one must choose $\Gamma(s) = (s + k_d)\Psi(s)$ where $k_d > 0$ can be interpreted as a position derivative (PD) gain in the closed loop system expressed in terms of the sliding variable e_s. With this choice of $\Gamma(s)$, the auxiliary control variable v, defined in equation (10), may be expressed as $v(t) = -a_r(t) - k_d e_s$ where

$$a_r(t) = \dot{e}_s - y^{(\gamma)} \tag{13}$$

Note that the term $y^{(\gamma)}$ is cancelled out of $\dot{e}_s$ in the last expression, hence $a_r(t)$ is measurable.

The modified control law that includes a sliding term is given by

$$u = \frac{-\hat{f} + v + u_{sl}}{\hat{g}} = \frac{-\hat{f} - a_r - k_d e_s + u_{sl}}{\hat{g}} \tag{14}$$

where the sliding term is defined by

$$u_{sl} = -\bar{k}(\mathbf{x}, t)\, sat(\frac{e_s}{\phi}) \tag{15}$$

where ϕ is the boundary layer thickness, sat(r) the saturation function defined by

$$sat(r) = \begin{cases} 1 & r \geq 1 \\ r & -1 < r < 1 \\ -1 & r \leq -1 \end{cases} \tag{16}$$

and $\bar{k}(\mathbf{x}, t)$ a state dependent sliding gain yet to be defined [18]. The choice of a boundary layer thickness is discussed in the next section.

The control input (14) may be separated into three functionally different components. The first component

$$u_{inv} = \frac{-\hat{f} + y_d^{(\gamma)}}{\hat{g}} \tag{17}$$

represents the approximated inverse of the plant input/output dynamics, driven by the γ derivative of the desired output. This is the feedforward part of the controller. The second control component

$$u_{pd} = -\frac{\alpha_\gamma e^{(\gamma-1)} + \cdots + \alpha_1 e}{\hat{g}} \tag{18}$$

is a position derivative control. This is a feedback term whose purpose is to stabilize the closed loop system. The third control component

$$u'_{sl} = \frac{u_{sl}}{\hat{g}} \tag{19}$$

is a sliding feedback term added to compensate for inherent approximation errors.

Inserting the control law (14) into the plant input/output equation (9) one obtains, after some manipulations, a first order error equation in terms of the sliding variable e_s:

$$\dot{e_s} = -\bar{k}\ \text{sat}(\frac{e_s}{\phi})\frac{g}{\hat{g}} - k_d\,\frac{g}{\hat{g}}\,e_s - m\tilde{\boldsymbol{\theta}}_f^T\mathbf{w}_f + m\tilde{\boldsymbol{\theta}}_g^T\mathbf{w}_g\frac{a_r + \hat{f}}{\hat{g}} + d \tag{20}$$

where d(t) is a disturbance due to the inherent approximation errors, given by

$$d(t) = \triangle f(\mathbf{x}) - \triangle g(\mathbf{x})\frac{a_r(t) + \hat{f}}{\hat{g}} \tag{21}$$

The first term on the right hand side of equation (20) results from the sliding control. Its role is to compensate for the disturbance $d(t)$. The second term on the right of (20) is a stabilizing feedback term. The third and fourth terms on the right of equation (20) are driving terms, linear in the parameter error vectors, that vanish if the parameters are set to their "optimal" values.

The incorporation of sliding control to compensate for network inherent approximation errors inside the state region $\mathcal{X}_n$ enables one to choose an adaptation deadzone as small as the boundary layer thickness ϕ. This is an advantage over using a conservative adaptation deadzone, because it allows the networks to exploit their full approximation potential rather than their assumed, conservative, worst case potential. This point will be discussed further in the next section and demonstrated in simulations. Following this argument, an adaptation deadzone equal to ϕ is chosen and the following adaptation laws applied:

$$\begin{aligned} \dot{\hat{\boldsymbol{\theta}}}_f &= \eta_f\, m(\mathbf{x})\mathbf{w}_f(\mathbf{x})e_{s_\triangle} \\ \dot{\hat{\boldsymbol{\theta}}}_g &= \begin{cases} -\eta_g\, m(\mathbf{x})\mathbf{w}_g(\mathbf{x})e_{s_\triangle}\frac{a_r(t)+\hat{f}}{\hat{g}} & \hat{g} \geq g_l \\ (g_l-\hat{g})\frac{\mathbf{w}_g(\mathbf{x})}{\|\mathbf{w}_g(\mathbf{x})\|^2}\delta(t) & \hat{g} < g_l \end{cases} \end{aligned} \tag{22}$$

where η_f and η_g are positive adaptation rates, $\delta(t)$ the impulse function and where

$$e_{s_\triangle} = e_s - \phi\, \mathrm{sat}(\frac{e_s}{\phi}) \tag{23}$$

is used to implement smooth adaptation with a deadzone width ϕ [17]. Note that $e_{s_\triangle} = 0$ when $|e_s| < \phi$ and thus, parameter adaptation is stopped within the deadzone. The adaptation law for $\hat{\boldsymbol{\theta}}_g$ includes parameter resetting to ensure that $\hat{g}$ remains bounded below by g_l.

The closed loop stability of the overall system, defined by the closed loop dynamics (20) and the adaptation laws (22), is analysed with the aid of the Lyapunov function candidate

$$V = \frac{1}{2}e_{s_\triangle}^2 + \frac{1}{2\eta_f}\tilde{\boldsymbol{\theta}}_f^T\tilde{\boldsymbol{\theta}}_f + \frac{1}{2\eta_g}\tilde{\boldsymbol{\theta}}_g^T\tilde{\boldsymbol{\theta}}_g \tag{24}$$

Assume first that parameter resetting does not occur and consider the time derivative of V along the system's trajectory. From the definition of $e_{s_\triangle}$ and the parameter update law (22) it follows that $\dot{V} = 0$ for $|e_s| \leq \phi$. In the case $|e_s| > \phi$, using the closed loop dynamics (20), the derivative of V can be written as

$$\begin{aligned} \dot{V} &= \dot{V}_1 + \dot{V}_2 + \dot{V}_3 - k_d\,\tfrac{g}{\hat{g}}\,e_{s_\triangle}^2 \\ where: & \\ \dot{V}_1 &= \tilde{\boldsymbol{\theta}}_f^T\left(-me_{s_\triangle}\mathbf{w}_f + \tfrac{1}{\eta_f}\dot{\hat{\boldsymbol{\theta}}}_f\right) \\ \dot{V}_2 &= \tilde{\boldsymbol{\theta}}_g^T\left(me_{s_\triangle}\mathbf{w}_g\tfrac{a_r+\hat{f}}{\hat{g}} + \tfrac{1}{\eta_g}\dot{\hat{\boldsymbol{\theta}}}_g\right) \\ \dot{V}_3 &= d(t)e_{s_\triangle} - \left(\dot{\phi} + (\bar{k}+k_d\phi)\tfrac{g}{\hat{g}}\right)|e_{s_\triangle}| \end{aligned} \tag{25}$$

where the identities $e_{s_\triangle} = e_s - \phi\,\mathrm{sgn}(e_{s_\triangle})$ and $\dot{e}_{s_\triangle} = \dot{e}_s - \dot{\phi}\,\mathrm{sgn}(e_{s_\triangle})$, valid for $|e_s| > \phi$, are used in derivation and where ϕ is allowed to vary with time.

From the definition of the adaptation law (22) it follows directly that $\dot{V}_1 = 0$ and $\dot{V}_2 = 0$. The derivative $\dot{V}_3$ can be made non-positive by a proper choice of the sliding control gain $\bar{k}(\mathbf{x}, t)$. Indeed, this is the purpose of using sliding control. Let us define

$$k(\mathbf{x}, t) = \frac{1}{g_l(\mathbf{x})}\left(\hat{g}(\mathbf{x})(D_f(\mathbf{x})+\mu) + \left|a_r(t)+\hat{f}(\mathbf{x})\right| D_g(\mathbf{x})\right) \tag{26}$$

where $D_f(\mathbf{x})$ and $D_g(\mathbf{x})$ are the inherent approximation error bounds defined in (6) and μ is a small positive constant. From the disturbance $d(t)$ definition (21) and equation (6) it follows that $(g/\hat{g})k(\mathbf{x},t) \geq |d(t)| + \mu$. Therefore, assuming ϕ is constant, the choice

$$\bar{k}(\mathbf{x},t) = \max(0\,, k(\mathbf{x},t) - k_d\phi) \tag{27}$$

results in $\dot{V}_3 \leq -\mu|e_{s_\triangle}|$. From this analysis it follows that

$$\dot{V} \leq -\mu|e_{s_\triangle}| - k_d\,\frac{g}{\hat{g}}\,e_{s_\triangle}^2 \tag{28}$$

Consider next the behaviour of V when parameter resetting occurs. Assume that resetting occurs at time $t = t^*$. The function $V(t)$ is discontinuous at $t = t^*$. It can be easily established that with the parameter resetting defined in equation (22) one has $\tilde{\boldsymbol{\theta}}_g^T\tilde{\boldsymbol{\theta}}_g(t_+^*) \leq \tilde{\boldsymbol{\theta}}_g^T\tilde{\boldsymbol{\theta}}_g(t_-^*)$. Therefore one also has $V(t_+^*) \leq V(t_-^*)$. Hence, from this observation and the previous analysis it can be concluded that $V(t)$ is a non increasing function of time. Thus, from the definition of V, it follows that $e_{s_\triangle}$, $\tilde{\boldsymbol{\theta}}_f$ and $\tilde{\boldsymbol{\theta}}_g$ are bounded.

Consequently, the following can be established:

- From the boundedness of e_s, and since the polynomial $\Psi(s)$ is Hurwitz, it follows that the output tracking error and its derivatives up to order $\gamma - 1$ are bounded.

- The desired output and desired output derivatives up to order γ are assumed bounded. Therefore the boundedness of the output and its derivatives up to order $\gamma - 1$ and the boundedness of a_r follow from the boundedness of the tracking error and its derivatives.

- From proposition (1) and the boundedness of the output and its derivatives up to order $\gamma - 1$ follow the boundedness of the state $\mathbf{z}$ of the normal form and the plant state $\mathbf{x}$.

- Since the Lie derivatives $f(\mathbf{x})$ and $g(\mathbf{x})$ are bounded by functions of $\mathbf{x}$, and $\mathbf{x}$ is bounded, $f(\mathbf{x})$ and $g(\mathbf{x})$ are bounded.

- The boundedness of $\hat{f}(\mathbf{x})$ and $\hat{g}(\mathbf{x})$ result in from the boundedness of the basis vectors $\mathbf{w}_f(\mathbf{x})$ and $\mathbf{w}_g(\mathbf{x})$, and the boundedness of $\hat{\boldsymbol{\theta}}_f$ and $\hat{\boldsymbol{\theta}}_g$.

- The derivative $\dot{e}_s$ is bounded because all the terms in equation (20) are bounded.

- Since $\mathbf{I}(\mathbf{z}_1, \mathbf{z}_2)$ is Lipschitz in $\mathbf{z}_1$ and $\mathbf{z}_2$ by assumption, the boundedness of $\mathbf{I}(\mathbf{z}_1, \mathbf{z}_2)$ follows from the boundedness of $\mathbf{z}_1$ and $\mathbf{z}_2$, and therefore $\dot{\mathbf{z}}$ is bounded and $\dot{\mathbf{x}}$ is bounded. (The boundedness of $\dot{\mathbf{x}}$ may be established directly from the plant equation (1) and the assumed smoothness of $\mathbf{a}(\mathbf{x})$ and $\mathbf{b}(\mathbf{x})$).

The asymptotic convergence of the error e_s to the boundary layer may be established, similarly to [15], by using the non-negative function

$$V^\star(t) = V(t) - \int_0^t (\dot{V}(\tau) + k_d\frac{g_l}{\hat{g}_h}e_{s_\triangle}^2)d\tau$$

where $\hat{g}_h$ is a constant upper bound of $\hat{g}(\mathbf{x})$. By construction, the function $V^\star(t)$ satisfies $\dot{V}^\star(t) = -k_d e_{s_\triangle}^2 g_l/\hat{g}_h$. Therefore, since $\dot{e}_{s_\triangle}$ is bounded, $\dot{V}^\star(t)$ is uniformly continuous and one can apply Barbalat lemma [16] to conclude that $\dot{V}^\star(t) \to 0$ and hence $e_{s_\triangle} \to 0$. It is therefore established that e_s converges asymptotically to the boundary layer, i.e. $|e_s|$ is asymptotically bounded by ϕ.

The asymptotic bounds of the tracking error and its derivatives up to order $\gamma - 1$ can be computed using the properties of the the transfer function $\Psi(s)^{-1}$. In particular, if the common choice

$$\Psi(s) = (s + \lambda)^{\gamma - 1} \tag{29}$$

where $\lambda > 0$, is used to define e_s, then the tracking error and its derivatives are asymptotically bounded by [18]:

$$|e^{(i)}| \leq 2^i \lambda^{i-\gamma+1} \phi \qquad 0 \leq i \leq \gamma - 1 \tag{30}$$

Comments

The stability proof did not take into account the factor $\mu|e_{s_\triangle}|$ in inequality (28). Hence, μ may be taken as zero. On the other hand, the proof could have been based on the term $-\mu|e_{s_\triangle}|$, neglecting the term $-k_d e_{s_\triangle}^2 g/\hat{g}$ in (28). Thus one can also take $k_d = 0$ with $\mu > 0$. Hence, one has the flexibility of choosing the PD gain k_d and the sliding gain μ taking into account the desired rate of convergence, limitations on the control magnitude and the frequency range of unmodeled dynamics. For example, choosing a large PD gain results in fast convergence but large control input when the tracking error is initially large. Choosing a large sliding gain also increases the convergence rate but may result in the excitation of high frequency unmodeled dynamics.

The dual role of sliding control is implicitly contained in the definitions of the sliding gain $\bar{k}(\mathbf{x}, t)$, given in equations (26) (27), and the inherent approximation errors bounds $D_f(\mathbf{x})$ and $D_g(\mathbf{x})$ given by equations (5) (6). The sliding control takes large magnitudes outside the network approximation region $\mathcal{X}_n^-$ where it compensates for initial approximation errors, and small magnitudes within $\mathcal{X}_n^-$ where it compensates for inherent network approximation errors. It is desirable to the have $\mathcal{X}_n^- \supset \mathcal{X}_d$. Indeed, if this is the case then asymptotic convergence of the tracking error to the boundary layer implies that asymptotically the state trajectory remains bounded inside $\mathcal{X}_n^-$, thus avoiding the use of large sliding control activity. From the definition of $\bar{k}(\mathbf{x}, t)$ it is also apparent that if $k_d\phi$ is sufficiently large to compensate for the inherent network approximation errors, then the sliding controller is not activated at all in the state region $\mathcal{X}_n^-$. Thus, using a fixed, conservative, boundary layer thickness such that $\bar{k}(\mathbf{x}, t) = 0$ on $\mathcal{X}_n^-$ is a special case of controller implementation, without using sliding control on $\mathcal{X}_n^-$.

Neglecting the affect of high frequency unmodeled dynamics, the boundary layer thickness may be chosen arbitrarily small. This implies that the adaptation deadzone may be taken arbitrarily small. This result is contrasted with the fixed, conservative deadzone that must be used if sliding control is not incorporated on $\mathbf{x} \in \mathcal{X}_n^-$.

Since such a conservative deadzone is typically based on conservative bounds on the inherent network approximation errors and other terms that constitute the disturbance $d(t)$, incorporating sliding control on $\mathbf{x} \in \mathcal{X}_n^-$ enables the adaptation laws to take advantage of the actual approximation abilities of the networks, without impairing stability. This in turn results in improved network approximations and faster convergence rates.

One might question the benefit of reducing the adaptation deadzone when the sliding control compensates for the worst case network's approximation abilities anyway. This may be challenged by noticing that the convergence proof guarantees only asymptotic tracking error convergence to the boundary layer. The speed of convergence depends on the quality of parameter estimation. Better use of available information can result in faster convergence rates and better network approximation. An improved network approximation enables faster adaptation to changes in the nature of the desired trajectories. Additionally, reduced approximation errors result in improved performance within the boundary layer. To see this, consider the error equation $\dot{e} = -ae + d(t) - k\,\mathrm{sat}(e/\phi)$, where $|d(t)| \leq k$ is a bounded disturbance. Within the boundary layer the error equation may be written as $\dot{e} = -(a + k/\phi)e + d(t)$. Clearly, the behaviour of the error within the boundary layer is determined by the disturbance $d(t)$; for example, if $d(t) = 0$ then $e \to 0$. It is therefore beneficial to reduce the magnitude of the disturbance even though the sliding term guarantees asymptotic convergence to the boundary layer anyway.

In practice, it may be beneficial to explicitly keep $\hat{f}$ and $\hat{g}$ bounded by modifying the adaptation laws. This may be done in the same way that $\hat{g}$ is kept bounded below by the adaptation laws, without impairing stability. The advantage of this is that a bound on the sliding gain can be predetermined. This provides an easy way of setting the boundary layer thickness as will be seen in the next section.

Parameter convergence can be achieved if the vector

$$\mathbf{w}^T(t) = [-\mathbf{w}_f^T(\mathbf{x}) \;\; \mathbf{w}_g^T(\mathbf{x})\frac{a_r(t) + \hat{f}}{\hat{g}}]$$

is persistently excited [16] [20]. It is typically difficult to establish in advance the persistent excitation (PE) of this vector. Achieving persistent excitation in the case where RBF networks with local receptive fields are used for the approximation, is discussed in [20]. The main conclusion of that discussion is the desirability of inclusion of an initial identification phase in which the desired trajectory is designed to be sufficiently rich. This initial phase achieves good initial network approximations of the system's Lie derivatives, thereby improving performance, especially in the face of changes in the nature of the desired trajectories.

5 Practical Controller Design: Boundary Layer Thickness Setting

In many practical applications, the system contains high frequency unmodeled dynamics. These may arise from time delays neglected in the course of modelling, structural resonance modes or finite sampling rate effects [18]. The controller must be designed to avoid excitation of the unmodeled dynamics.

Denote by $\nu/(2\pi)$ the maximum frequency allowed to exist in the system without excitation of unmodeled dynamics. The following constraints should be applied in the controller design:

1. The boundary layer bandwidth, i.e. the cutoff frequency of the low-pass filter $\Psi^{-1}(s)$ should be set lower than ν. If equation (29) is used to define $\Psi(s)$ this implies $\lambda \leq \nu$.

2. The boundary layer thickness must be set such that the low-pass filter structure of the closed loop system, when e_s is inside the boundary layer, has a bandwidth smaller than ν [18].

3. The adaptation gains η_f and η_g must be chosen sufficiently small such that fast adaptation do not excite the unmodeled dynamics.

In order to consider the setting of the boundary layer thickness, one must first distinguish between two possible modes in which the control method may be applied. Recall that the set $\mathcal{X}_d$ was defined as the region in which the state trajectory lies when the output tracks the desired trajectory. One possible scenario is to use the network approximation only on a subset of $\mathcal{X}_d$, with the purpose of optimizing the control at least on this subset. It may be necessary to use this mode of control if the state has large variations or if the set $\mathcal{X}_d$ cannot be fully determined. This mode of operation does not violate any of the steps of the convergence proof and therefore is a valid operation mode. The designer must choose the boundary layer thickness according to a bound on the initial approximation errors *outside* the state region $\mathcal{X}_n$ and the frequency range of the unmodeled dynamics. A second scenario which is more desirable is to have $\mathcal{X}_n^- \supset \mathcal{X}_d$, in which case the state trajectory is asymptotically bounded in the region $\mathcal{X}_n^-$. In what follows, the appropriate choice of ϕ in this case is discussed.

Consider the dynamical behaviour of the sliding variable inside the boundary layer, i.e when $|e_s| < \phi$. From equations (20) and (27) it follows that

$$\dot{e}_s = -\frac{k_{\text{eff}}(\mathbf{x})}{\phi}\frac{g}{\hat{g}}e_s + \left(-m\tilde{\boldsymbol{\theta}}_f^T\mathbf{w}_f + m\tilde{\boldsymbol{\theta}}_g^T\mathbf{w}_g\frac{a_r + \hat{f}}{\hat{g}} + d(t)\right) \tag{31}$$

where $k_{\text{eff}}(\mathbf{x}, t) = \max\left(k(\mathbf{x}, t), k_d\phi\right)$ is the effective gain within the boundary layer. This equation represents a first order low-pass filter with a time varying gain and the total approximation error as an input. In order not to excite unmodeled dynamics, the gain in equation (31) must satisfy

$$\frac{k_{\text{eff}}(\mathbf{x}, t)}{\phi}\frac{g}{\hat{g}} < \nu \quad \text{for all } \mathbf{x} \in \mathcal{X}_d. \tag{32}$$

A conservative boundary layer thickness

A conservative boundary layer thickness may be chosen such that sliding control is not used at all in the state region $\mathcal{X}_n^-$. This is achieved if $k_d \phi > k(\mathbf{x}, t)$ for all $\mathbf{x} \in \mathcal{X}_n^-$, in which case $\bar{k}(\mathbf{x}, t) = 0$ on $\mathcal{X}_n^-$. Hence, a suitable conservative deadzone setting can be taken as

$$\phi_{con} = \frac{K_n}{k_d} \tag{33}$$

where K_n is a fixed upper bound of $k(\mathbf{x}, t)$ in the region $\mathbf{x} \in \mathcal{X}_n^-$.

Although the existence of K_n is guaranteed by the stability proof, it cannot be predetermined in the case $\gamma < n$. This is because of the presence of $a_r(t)$ in the expression (26) for $k(\mathbf{x}, t)$. Hence K_n must be determined experimentally in the case $\gamma < n$. From this fact, and since the inherent approximation error bounds are typically very conservative, it follows that K_n and therefore ϕ_{con} are likely to be extremely conservative. This may result in an unacceptably large tracking error.

Note that in order not to excite unmodeled dynamics, the PD gain must satisfy $k_d g/\hat{g} < \nu$. Combining this with the above choice of ϕ_{con}, one obtains

$$\phi_{con} > \frac{k(\mathbf{x}, t)}{k_d} > \frac{g}{\hat{g}} \frac{k(\mathbf{x}, t)}{\nu} \tag{34}$$

where the inequalities are likely to be strong.

A fixed, minimal, boundary layer thickness

If ϕ is allowed to be smaller than the conservative choice above, one has $k_d \phi < k(\mathbf{x}, t)$ for some values of $\mathbf{x} \in \mathcal{X}_n^-$ and the sliding controller is active. In order to comply with condition (32) ϕ must satisfy

$$\phi > \frac{g}{\hat{g}} \frac{k(\mathbf{x}, t)}{\nu} \quad \text{for all } \mathbf{x} \in \mathcal{X}_d. \tag{35}$$

The value of the right hand side of this inequality is equal to that of the right hand side of inequality (34). Therefore, the incorporation of sliding control allows one to set the minimum possible ϕ, regardless of the value of k_d. This is beneficial in many applications in which k_d cannot assume values as large as those allowed by inequality (34) because of limitations on the control magnitude. Further, since within the boundary layer the output error and its derivatives are small, one may assume $|a_r(t)| \approx |y_d^{(\gamma)}|$ when computing a bound on $k(\mathbf{x}, t)$. Therefore one normally has $\phi_{\min} << \phi_{con}$.

In order to choose a value for ϕ one can either find a suitable value experimentally, or compute a bound of the expression on the right hand side of inequality (35). A suitable choice of ϕ is given by

$$\phi_{\min} = \frac{g_d}{\nu \, g_l} \left(\epsilon_f + \mu + \epsilon_g \frac{\max |y_d^{(\gamma)}| + \hat{f}_d}{g_l} \right) \tag{36}$$

where g_d is an upper bound of $g_h(\mathbf{x})$ on $\mathbf{x} \in \mathcal{X}_d$, and $\hat{f}_d$ an upper bound of $|\hat{f}(\mathbf{x})|$ on $\mathbf{x} \in \mathcal{X}_d$. The bound $\hat{f}_d$ may be found experimentally or, preferably, $|\hat{f}(\mathbf{x})|$ may be kept bounded by the adaptation law as suggested in the previous section.

Time varying boundary layer thickness

In the case where the system's strong relative degree equals the system's order, i.e $\gamma = n$, the desired state is available. It is then possible to vary the boundary layer thickness with time so as to take maximum advantage of the available bandwidth [18]. In this case the sliding gain $\bar{k}$ must be modified in order to account for the presence of $\dot{\phi}$ in equation (25). In order to determine the sliding gain, one has to consider the cases $\dot{\phi} < 0$ and $\dot{\phi} > 0$ separately. It can be verified that an appropriate choice for $\bar{k}$ is

$$\bar{k}(\mathbf{x}, t) = \max\left(0\,, k_v(\mathbf{x}, t)\right) - k_d\phi) \tag{37}$$

where

$$k_v(\mathbf{x}, t) = k(\mathbf{x}, t) - \frac{\hat{g}(\mathbf{x})}{g_b(\mathbf{x})}\dot{\phi} \tag{38}$$

and

$$g_b(\mathbf{x}) = \begin{cases} g_h(\mathbf{x}) & \dot{\phi} > 0 \\ g_l(\mathbf{x}) & \dot{\phi} < 0 \end{cases} \tag{39}$$

Assuming continuity of f, g, $\hat{f}$ and $\hat{g}$, the dynamics of the variable e_s inside the boundary layer, may be written as

$$\dot{e_s} = -\frac{k_{\text{eff}}(\mathbf{x}_d)}{\phi}\frac{g(\mathbf{x}_d)}{\hat{g}(\mathbf{x}_d)}e_s + \left\{ f(\mathbf{x}_d) - \hat{f}(\mathbf{x}_d) + \left(\hat{g}(\mathbf{x}_d) - g(\mathbf{x}_d)\right)\frac{\hat{f}(\mathbf{x}_d) - y_d^{(\gamma)}}{\hat{g}(\mathbf{x}_d)} + O(\phi) \right\}$$

where

$$k_{\text{eff}}(\mathbf{x}_d) = \max\left(k_v(\mathbf{x}_d, t), k_d\phi\right)$$

The gain of this low-pass filter is expressed in terms of the desired state. Therefore, a time varying boundary layer that depends on the maximum inherent approximation error, evaluated at the desired state, can be used.

Consider the case in which sliding control is active. In order not to excite unmodeled dynamics ϕ must satisfy

$$\phi > \left.\frac{k_v}{\nu}\frac{g}{\hat{g}}\right|_d \tag{40}$$

where the subscript $|_d$ indicates the expression is evaluated at $\mathbf{x} = \mathbf{x}_d$. Using definition (26) of $k(\mathbf{x})$ one obtains

$$\left.\frac{g}{\hat{g}}k\right|_d = \left.\frac{g}{g_l}\left(D_f + \mu + \frac{|\hat{f} - y_d^{(\gamma)}|}{\hat{g}}D_g\right)\right|_d \leq \left.\frac{g_h}{g_l}k_1\right|_d$$

where

$$k_1(\mathbf{x}_d) = D_f(\mathbf{x}_d) + \mu + \frac{|\hat{f}(\mathbf{x}_d) - y_d^{(\gamma)}|}{\hat{g}(\mathbf{x}_d)}D_g(\mathbf{x}_d) \tag{41}$$

Using the last inequality one can establish that the following setting of ϕ satisfies inequality (40)

$$\phi\nu = \begin{cases} \beta_d k_1(\mathbf{x}_d) - \beta_d^{-1}\dot{\phi} & \dot{\phi} > 0 \\ \beta_d k_1(\mathbf{x}_d) - \beta_d \dot{\phi} & \dot{\phi} < 0 \end{cases}$$

where $\beta_d(\mathbf{x}_d) = g_h(\mathbf{x}_d)/g_l(\mathbf{x}_d)$. Hence, the following dynamics of the boundary layer thickness may be used to adapt ϕ:

$$\begin{cases} \dot{\phi} + \beta_d \nu \phi & = & \beta_d^2 k_1(\mathbf{x}_d) & \dot{\phi} > 0 \\ \dot{\phi} + \beta_d^{-1} \nu \phi & = & k_1(\mathbf{x}_d) & \dot{\phi} < 0 \end{cases} \tag{42}$$

The time varying bound $k_1(\mathbf{x}_d)$ is typically smaller than the parenthesised expression in the definition of $\phi_{\min}$ (36) that is used to set a fixed boundary layer thickness. Therefore, the time varying boundary layer is thinner than $\phi_{\min}$. In particular, if the network approximation ability is known to be better in some state space regions than in other regions, then the benefit of using the state dependent inherent approximation error bounds $D_f(\mathbf{x}_d)$ and $D_g(\mathbf{x}_d)$ rather than the uniform bounds ϵ_f and ϵ_g, to determine the boundary layer thickness is evident.

The boundary layer thickness can be varied according to equation (42) whether the sliding control is active or not. This will assure that ϕ is kept at a minimal value that reflects the history of the disturbance bound $k_1(\mathbf{x}_d)$. This formulation is valid regardless of the location of the state region $\mathcal{X}_n^-$ relative to $\mathcal{X}_d$. If the desired trajectory leaves $\mathcal{X}_n^-$, the boundary layer thickness is automatically increased to account for the large inherent approximation errors outside $\mathcal{X}_n^-$. Conversely, the boundary layer becomes thinner when the desired state enters the state region $\mathcal{X}_n^-$.

6 Simulation Results

Two simulations are used to demonstrate the features of the control method. In the first simulation the control architecture is applied to an example plant with no internal dynamics. In the second simulation an example plant with internal dynamics is used.

6.1 Simulation 1

The plant used in this simulation is identical to that used in [15]. The plant is given in normal form by

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

where

$$f(\mathbf{x}) = 16\frac{sin(4\pi x_1)}{4\pi x_1}\left(\frac{sin(\pi x_2)}{\pi x_2}\right)^2$$

and

$$g(\mathbf{x}) = 2 + sin\left(3\pi(x_1 - 0.5)\right)$$

The plant has a strong relative degree equal to the order: $\gamma = n = 2$. Hence, no internal dynamics exist. Assumptions (1) and (2) are satisfied and the required bounds may be taken as: $g_l = g_l(\mathbf{x}) = 1$, $g_h(\mathbf{x}) = 4$ and $f_h(\mathbf{x}) = 20$.

The plant output is required to track a desired trajectory given by y_d and $\dot{y}_d$. The signal y_d is the output of a low-pass filter with the transfer function $(1+s/10)^{-3}$, which is driven by a unity amplitude square wave input with frequency $0.4H_z$ and a time average of 0.5. It can be verified the desired trajectory lies in the state region $\mathcal{X}_d = \mathcal{X}_n^- = [0, 1] \times [-3, 3]$ and that $|\ddot{y}_d| \leq 25$.

The approximations $\hat{f}$ and $\hat{g}$ are given by equation (4) with $f_0(\mathbf{x}) = 0$ and $g_0(\mathbf{x}) = 1$. The network $\mathbf{w}_f(x_1, x_2)$ is constructed with gaussian radial basis functions with a standard deviation of 0.11, that are spread on a regular grid with a mesh size of 0.1. The grid covers the state region $[-0.5, 1.5] \times [-3.5, 3.5]$. The network $\mathbf{w}_g(x_1)$ is constructed in the same way but with a one dimensional grid on the interval $[-0.5, 1.5]$. The (conservative) uniform approximation bounds $\epsilon_f = \epsilon_g = 0.001$, can be computed using sampling theory [15]. Using these bounds, the inherent approximation error bounds in equation (6) may be taken as: $D_f(\mathbf{x}) = 0.001m + 20(1 - m)$ and $D_g(\mathbf{x}) = 0.001m + 3(1 - m)$, where $m(\mathbf{x})$ vanishes outside the state region $\mathcal{X}_n = [0, 1.025] \times [-3.075, 3.075]$.

The control law is defined by equation (14) with $k_d = 5$, $\lambda = 10$, $\mu = 0.05$ and $\bar{k}$ as defined by equations (27) and (26). The adaptation law is defined by equation (22) where the adaptation gains are $\eta_f = 500$ and $\eta_g = 4$. The adaptation law is modified to guarantee $|\hat{f}| \leq 20$ and $\hat{g} \leq 4$.

A conservative boundary layer thickness may be determined by equation (33). Using the inequality $|a_r| \leq 100$ which holds on $\mathbf{x} \in \mathcal{X}_d$, and the previously established bounds, one obtains $k(\mathbf{x}) \leq 0.125$ on $\mathbf{x} \in \mathcal{X}_d$. Thus, the choice $\phi = \phi_{con} = 0.025$ guarantees convergence of e_s into the boundary layer without using sliding control in the region $\mathcal{X}_d$.

The only source of high frequency unmodeled dynamics is assumed to be the finite sampling rate of $2000H_z$. Taking $\nu = 400$ (rad/sec) is therefore safe and guarantees that unmodeled dynamics are not excited [18]. Note that both k_d and λ are much smaller than ν. A minimum choice of a fixed boundary layer thickness can be made according to equation (36) which yields $\phi_{\min} \approx 0.001$. This is twenty five times smaller than the value of the conservative deadzone ϕ_{con}.

Figure (1) presents the output tracking error e during the first 50 seconds of operation. The top plot was obtained without adaptation at all, the middle plot with $\phi = \phi_{con} = 0.025$ and the bottom plot with $\phi = 0.005$. It can be seen that the tracking error in the second and third cases have converged to the boundary layer, whose width is given by ϕ/λ (see equation (30)).

Figure (2) depicts the feedforward control term u_{inv}, the PD control term u_{pd}, and the sliding control term u'_{sl}, that are given in equations (17) (18) and (19) respectively, during the first 50 seconds of operation. The boundary layer thickness was set to $\phi = 0.005$. It is seen that as the networks learn, the feedforward control term becomes increasingly dominant. The sliding control magnitude is seen to be much smaller than the both the forward and the PD control term magnitudes.

Figure (3) depicts the energy of the sliding variable e_s, during 1000 seconds of learning for the cases $\phi = 0.02$, $\phi = 0.01$ and $\phi = 0.005$. The sliding variable power was averaged over time intervals of 5 seconds to produce the plot. Performance is improved as ϕ is decreased. Note that while in the cases $\phi = 0.02$ and $\phi = 0.01$ the energy bottoms and remains flat after 100 seconds, in the case $\phi = 0.005$ it keeps on decreasing after that time. This observation suggests that even though e_s is small and "noisy", it still contains information that can be used to improve the network approximation. This conclusion is further supported by Figure (4) which depicts the approximation of $g(\mathbf{x})$ achieved after 1000 seconds of operation, with the three different settings of ϕ. The approximation improves as ϕ decreases. The benefit of using sliding control to reduce the adaptation deadzone width is evident. Note that in all cases the approximation is far from "optimal". This is because the desired trajectory does not provide persistent excitation.

6.2 Simulation 2

The plant used in this simulation is given in normal form by

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

where

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

The plant has a strong relative degree $\gamma = 1$, smaller than the plant order $n = 2$. The zero dynamics, given by $\dot{x}_2 = -x_2$ are globally exponentially stable and the internal dynamics $I(\mathbf{x}) = x_1 - x_2$ are Lipschitz in both x_1 and x_2. Therefore assumptions (1) and (2) hold and the control method can be applied. The bounds required by assumption (2) may be taken as $g_l = g_l(\mathbf{x}) = 1$, $g_h(\mathbf{x}) = 4$ and $f_h(\mathbf{x}) = 2$. Note that in this example both f and g depend on the entire state vector.

The plant output is required to track a desired trajectory given by y_d. The signal y_d is the output of a low-pass filter with the transfer function $(1 + s/10)^{-3}$, which is driven by a square wave input with amplitude 1.8, frequency $0.4H_z$ and a zero time average. From the definition of the desired trajectory, it can be concluded that if the tracking error is small, the internal state x_2 can be expected to satisfy $|x_2| \leq 1$. Therefore we set $\mathcal{X}_d = \mathcal{X}_n^- = [-1, 1] \times [-1, 1]$.

The two networks $\mathbf{w}_f$ and $\mathbf{w}_g$, that are taken as identical, consist of gaussian basis functions with standard deviation 0.03 that spread over a regular grid defined on the region $[-1.5, 1.5] \times [-1.5, 1.5]$, with a mesh size of 0.05. The (conservative) uniform approximation bounds are taken to be $\epsilon_f = \epsilon_g = 0.005$. Using these bounds, and $f_0(\mathbf{x}) = 0$, $g_0(\mathbf{x}) = 1$, the inherent approximation bounds in equation (6) may be given by $D_f(\mathbf{x}) = 0.005m + 3(1 - m)$ and $D_g(\mathbf{x}) = 0.005m + 2(1 - m)$.

The control law is defined by equation (14) with $k_d = 5$, $\lambda = 10$, $\mu = 0.01$ and $\bar{k}$ as defined by equation (27). The adaptation law is defined by equation (22) where the gains are $\eta_f = 25$ and $\eta_g = 25$. The adaptation law is modified to guarantee $|\hat{f}| \leq 2$ and $\hat{g} \leq 4$.

A conservative boundary layer thickness may be determined by equation (33). Using $|a_r| = |\dot{y}_d| \leq 3$, the previously established bounds, and $\mu = 0$, one obtains $k(\mathbf{x}, t) \leq 0.045$ on $\mathcal{X}_d$. Therefore $\phi_{con} = 0.012$. As in the previous simulation we have $\nu = 400$, and a minimum boundary layer thickness may be computed according to equation (36). This yields $\phi_{\min} \approx 0.0004$. This bound is thirty times smaller than the value of the conservative deadzone.

Figure (5) depicts the output tracking error e during the first 200 seconds of operation, with $\phi = 0.012$ and $\phi = 0.001$. Note the dramatically improved performance obtained for $\phi = 0.001$. Figure (6) depicts the energy of the sliding variable $e_s = e$, during 1000 seconds of operation, for the cases $\phi = 0.012$, $\phi = 0.005$ and $\phi = 0.001$. The results are similar to those in simulation 1.

7 Conclusions

A direct control method for the nonlinear plant (1) using neural networks was presented, analysed and proven stable. The method is based on neural network approximation of functions that define the plant input/output dynamics. The necessary structural assumptions of the plant, and the network approximation requirements were specified. The controller architecture and design, including practical design considerations, were detailed.

The control architecture was presented as a modification of the method introduced in [20], where sliding control is incorporated. The use of sliding control makes it possible to assure global stability of the closed loop system. Furthermore, by compensating for inherent network approximation errors, the sliding control enhances the controller's performance. Specifically, the asymptotic tracking error can be made as small as desired, limited only by the constraint imposed on the closed loop system bandwidth by the presence of high frequency unmodeled dynamics. Moreover, the networks' adaptation deadzone can be taken as small as the boundary layer thickness. This results in a better use of available information, and therefore enables improved network approximations and faster parameter convergence.

The setting of the different controller parameters was discussed in detail. It was pointed out that a desirable mode of operation is to have $\mathcal{X}_d \subset \mathcal{X}_n^-$. In this case, the state trajectory is asymptotically bounded in the region $\mathcal{X}_n^-$, thus avoiding the use of crude sliding control. Assuming $\mathcal{X}_d \subset X_n^-$, and given the frequency range of unmodeled dynamics, a lower bound on the thickness of the boundary layer was derived. A strategy for varying the boundary layer thickness with time, so as to make a better use of available bandwidth, was outlined for the case where no internal dynamics exist, i.e. $\gamma = n$.

It is generally desirable to include an initial identification phase with a persistently exciting desired trajectory signal, before engaging in actual control. This will provide good initial network approximations and improved performance especially if the desired trajectories are non-stationary.

Although knowledge of plant nonlinearities is not necessary, it is beneficial to include any such knowledge in the controller design. This knowledge may be expressed either in terms of the initial approximations $f_0(\mathbf{x})$ and $g_0(\mathbf{x})$, or by explicit inclusion of known nonlinear terms scaled by unknown parameters, in the networks' structure.

The control architecture described in this paper can be extended to deal with multiple input multiple output systems. Such an extension is described in [20]. In order to incorporate sliding control in the multiple input multiple output case, some further assumptions are required about the uncertainties of the plant dynamics.

8 Acknowledgements

I wish to thank my supervisor Prof. Frank Fallside for his support, to Prof. Keith Glover for his helpful comments and to Tim Jervis and Ian Braithwaite for their comments.

References

[1] D. Claude. Everything you always wanted to know about linearization. In M. Fliess and M. Hazewinkel, editors, *Algebraic and Geometric Methods in Nonlinear Control Theory*. D. Reidel, 1986.

[2] K. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2:183–192, 1989.

[3] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

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

[5] M. I. Jordan and D. E. Rumelhart. Forward models: Supervised learning with a distal teacher. Occasional Paper 40, Center for Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA, 1991.

[6] S. H. Lane, M. G. Flax, D. A. Handleman, and J. J. Gelfand. Function approximation using multi-layered neural networks with b-spline receptive field functions. Technical Report CSL Report 47, Cognitive Science Laboratory, Princeton University, Princeton, New Jersey 08544, 1991.

[7] W. Li and J. J. E. Slotine. Neural network control of unknown nonlinear systems. In *Proc. 1989 ACC*, pages 1136–1141, 1989.

[8] M. R. Lynch. *Adaptive techniques in Signal Processing and Connectionist Models*. PhD thesis, Cambridge University Engineering Department, Cambridge, England, 1990.

[9] W. T. Miller, F. H. Glanz, and L. G. Kraft. CMAC: an associative neural network alternative to backpropagation. *Proceeding of the IEEE*, 78:1561–1567, 1990.

[10] W. T. Miller, R. P. Hewes, F. H. Glanz, and L. G. Kraft. Real-time dynamic control of an industrial manipulator using a neural-network-based learning controller. *IEEE Transactions on Robotics and Automation*, 6(1), February 1990.

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

[12] K. S. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1(1):4–27, 1990.

[13] M. Niranjan and F. Fallside. Neural networks and radial basis functions in classifying static speech patterns. Technical Report CUED/F-INFENG/TR 22, Cambridge University Engineering Department, Cambridge, England, 1988.

[14] M. J. D. Powell. The theory of radial basis function approximation in 1990. Technical Report DAMTP/NA11, Department of Applied Mathematics and Theoretical Physics, Cambridge University, Cambridge, England, 1990.

[15] R. M. Sanner and J. J. E. Slotine. Gaussian networks for direct adaptive control. Technical Report NSL-910503, Nonlinear Systems Laboratory, Massachusetts Institute of Technology, Cambridge, MA, 1991.

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

[17] J. J. E. Slotine and J. A. Coetsee. Adaptive sliding controller synthesis for non-linear systems. *Intl. Journal of control*, 3:44–47, 1986.

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

[19] A. Teel, R. Kadiyala, P. Kokotovik, and S. Sastry. Indirect techniques for adaptive input-output linearization of non-linear systems. *Int. J. Control*, 53:193–222, 1991.

[20] E. Tzirkel-Hancock and F. Fallside. A direct control method for a class of nonlinear systems using neural networks. Technical Report CUED/F-INFENG/TR.65, Cambridge University Engineering Department, Cambridge, England, March 1991.

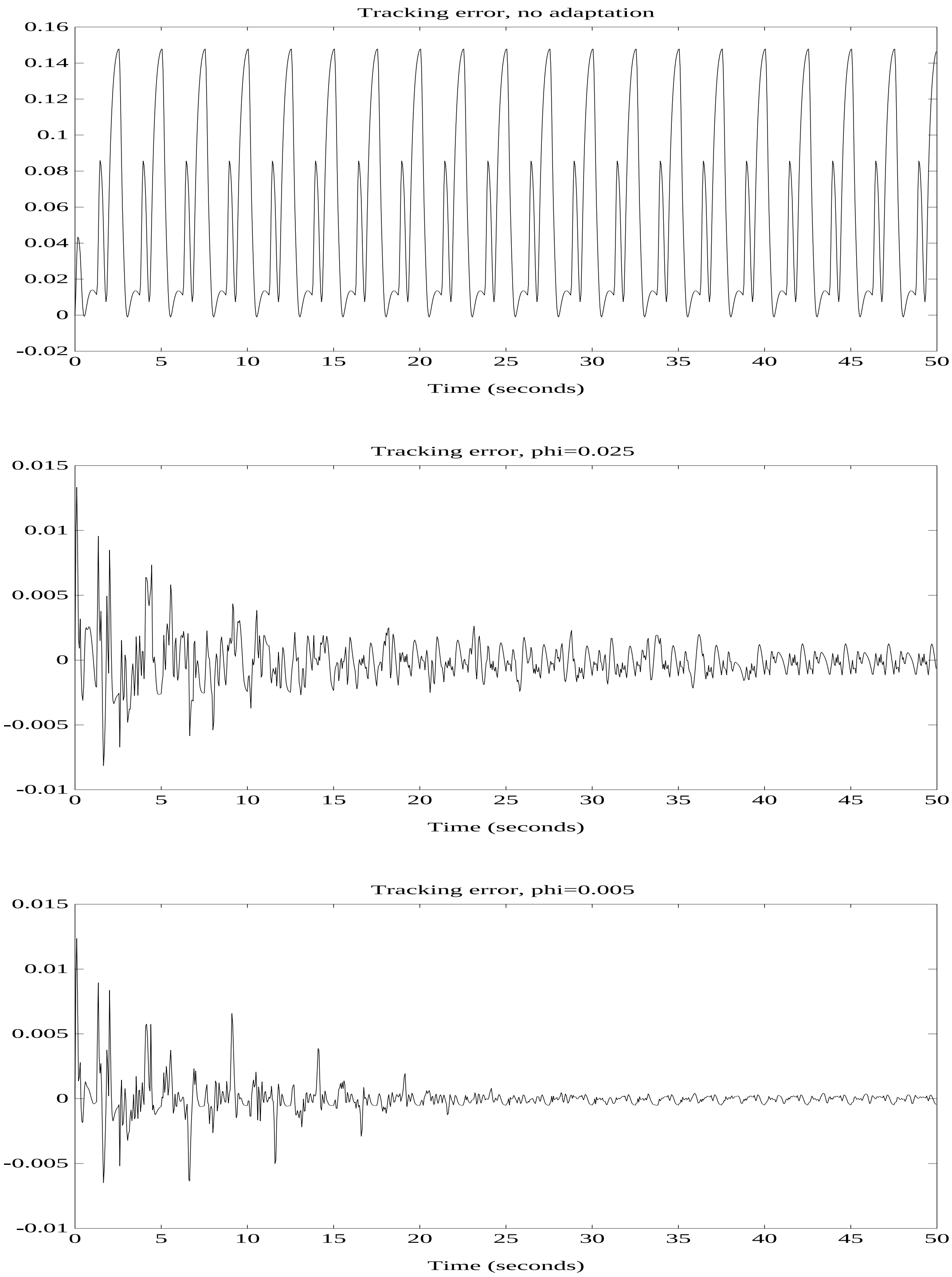


Figure 1: Output tracking error without adaptation (top), with $\phi = 0.025$ (middle) and with $\phi = 0.005$ (bottom).

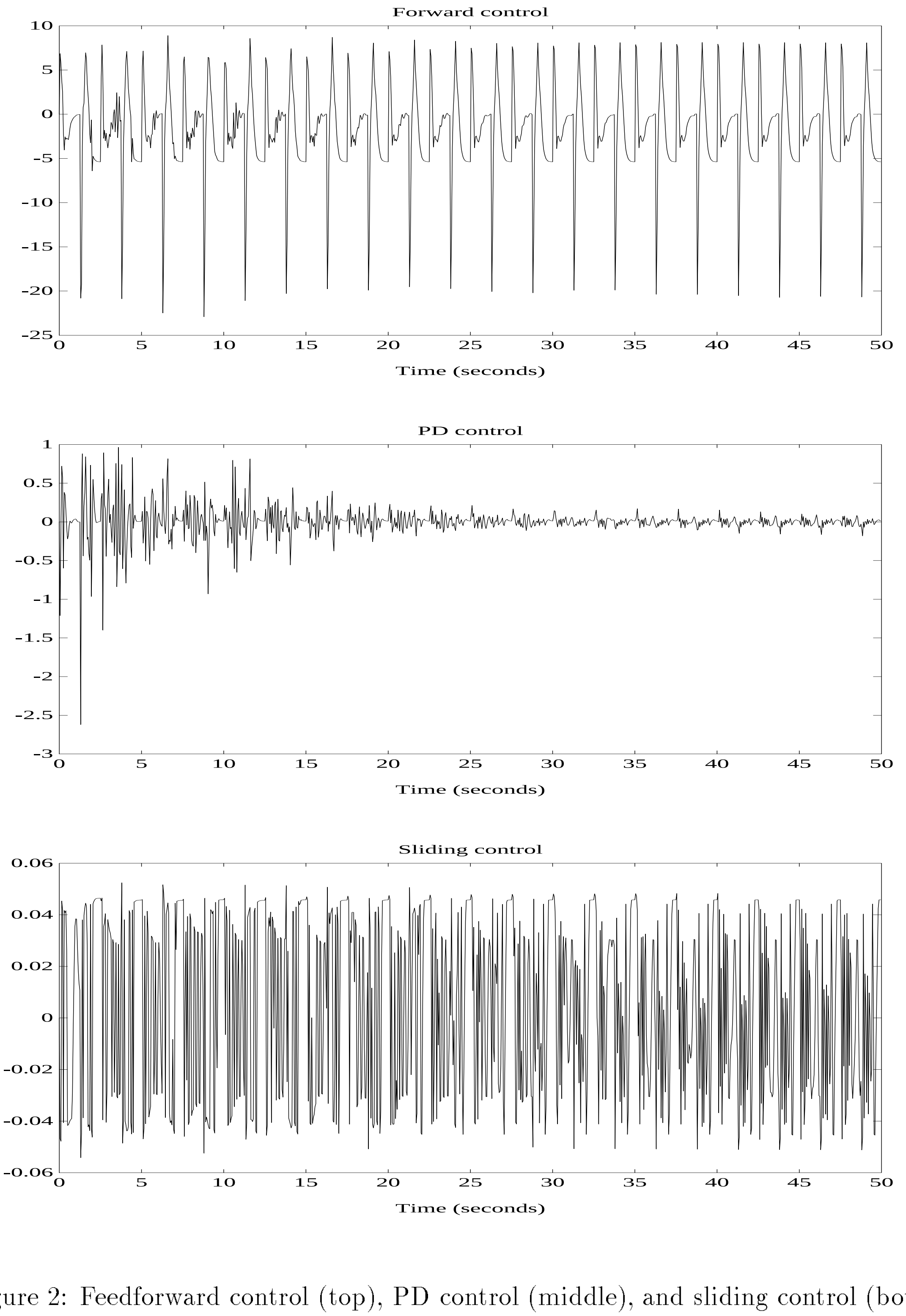


Figure 2: Feedforward control (top), PD control (middle), and sliding control (bottom) components, for $\phi = 0.005$.

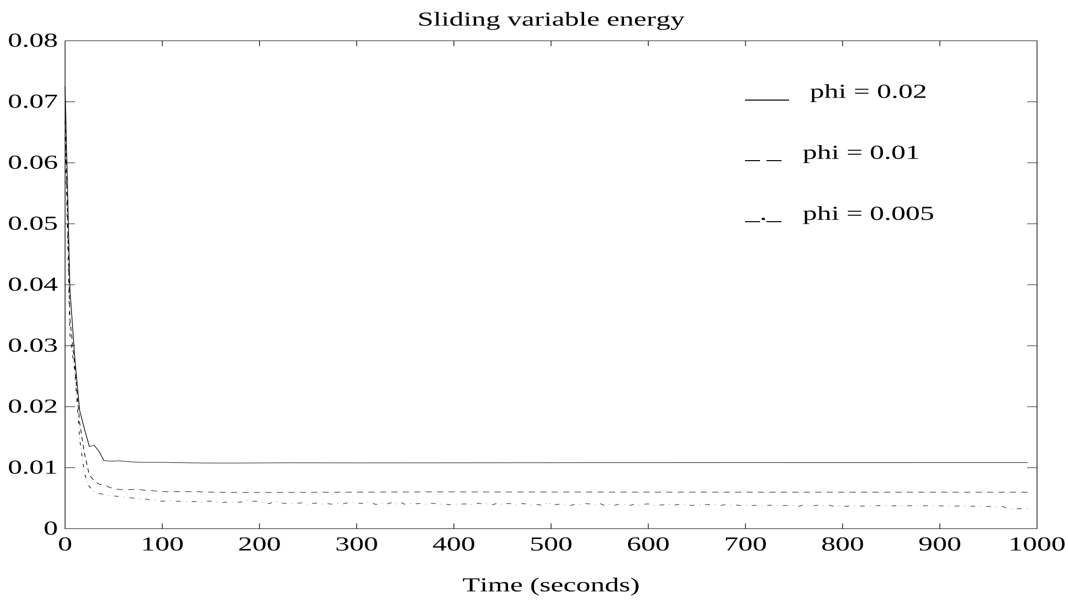


Figure 3: Sliding variable energy with different thickness settings of the boundary layer.

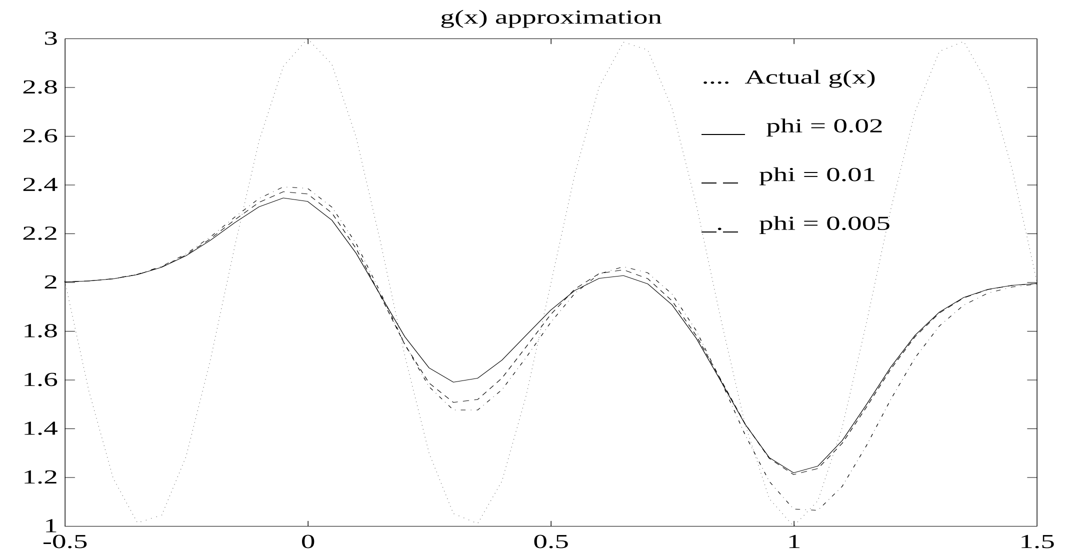
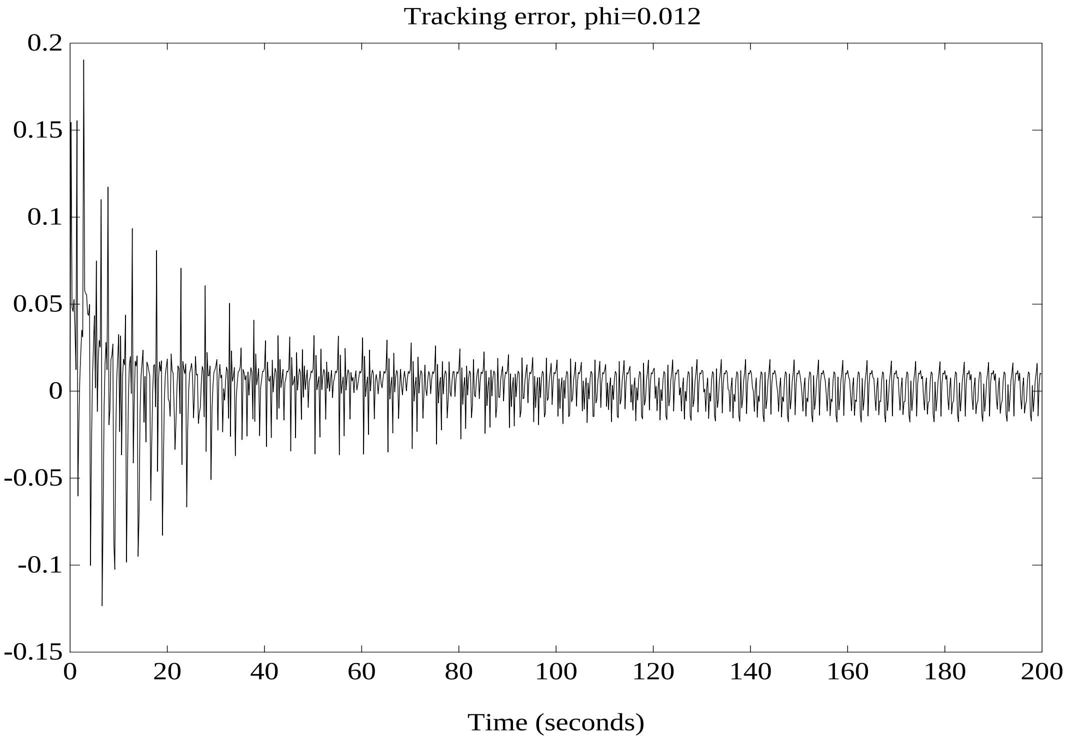


Figure 4: $g(\mathbf{x})$ approximation achieved with different thickness settings of the boundary layer.



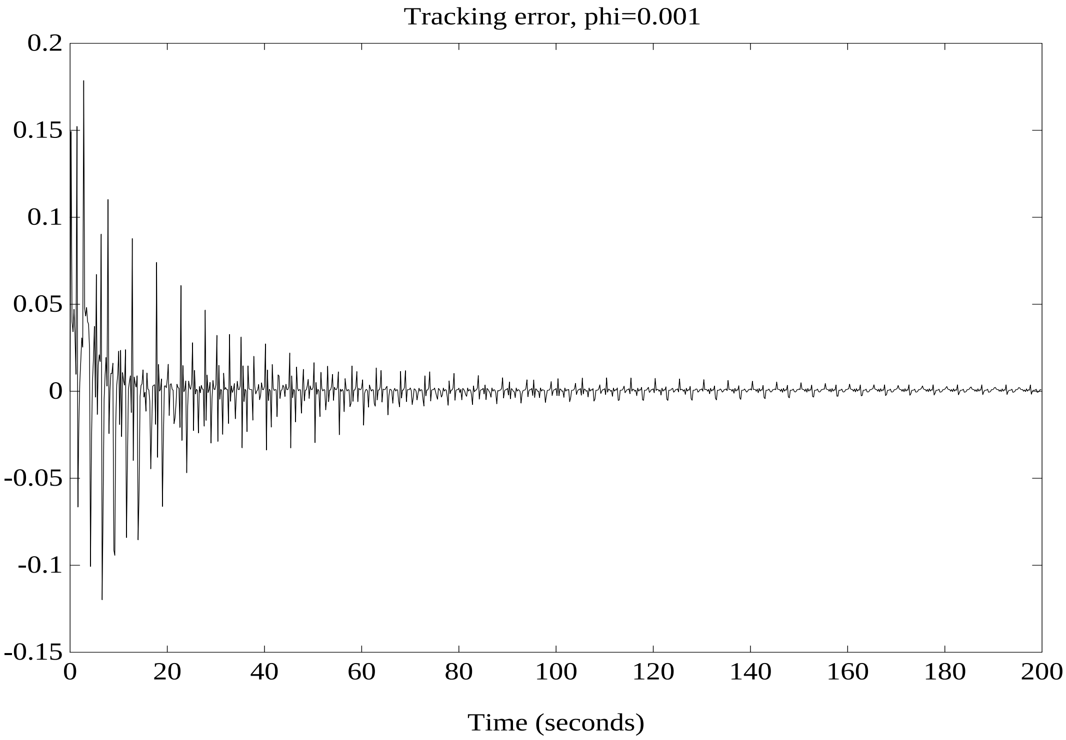


Figure 5: Tracking error with $\phi = 0.012$ (top) and $\phi = 0.001$ (bottom).

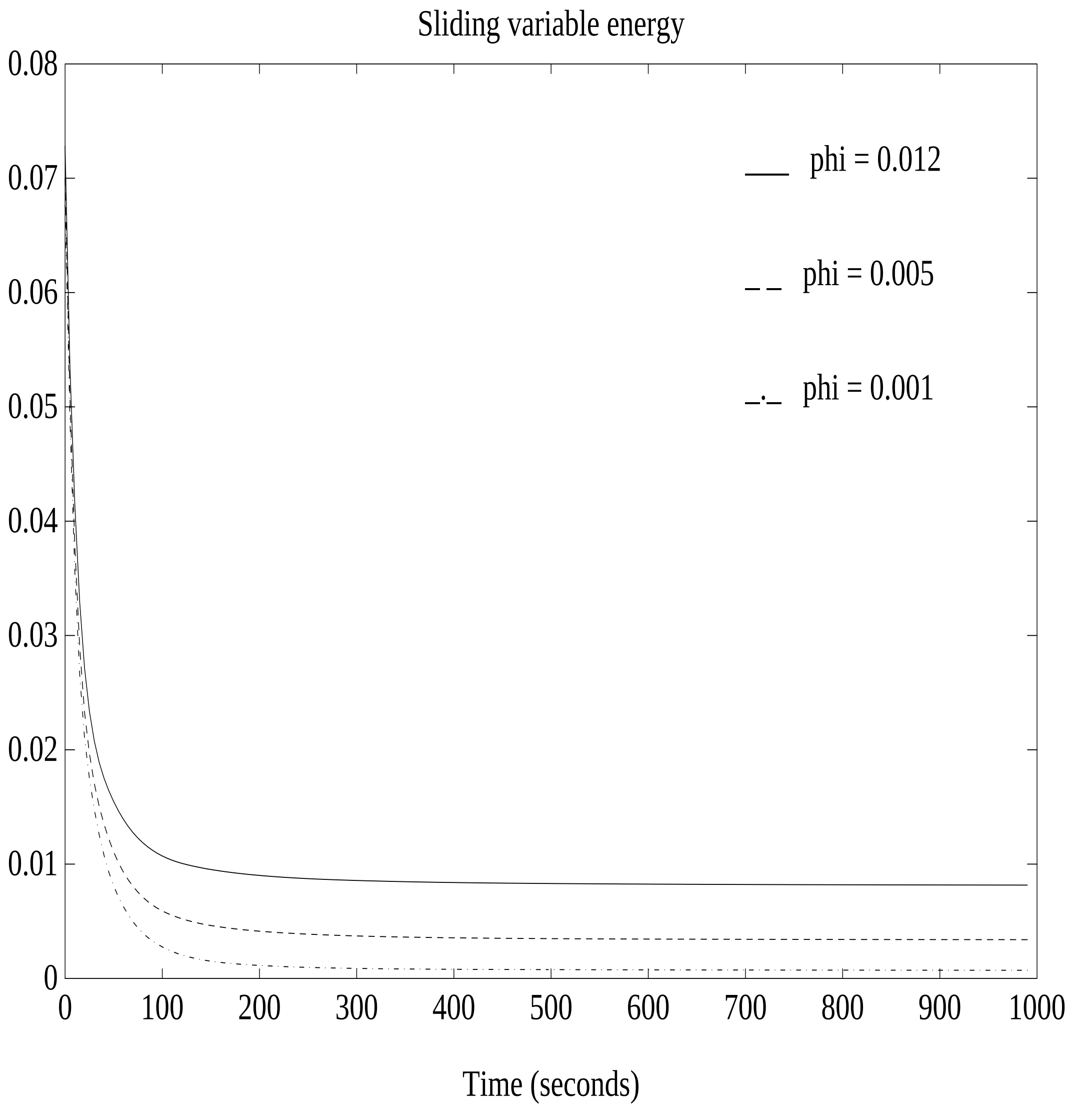


Figure 6: Sliding variable energy with different thickness settings of the boundary layer.