Neural Adaptive Assist-As-Needed Control for Rehabilitation Robots

Hamed N. Rahimi, Ian Howard, Lei Cui

Department of Mechanical Engineering, Curtin University, Perth, Western Australia 6102, Australia

hamed.rahiminohooji@postgrad.curtin.edu.au, {I.Howard, Lei.Cui}@curtin.edu.au

Abstract

Robot-assisted therapy can improve motor function in patients recovering from stroke. Assist-as-needed algorithms provide only minimal robotic assistance in the therapy, thus requiring significant effort from the impaired subject. This paper presents an adaptive neural assist-as-needed controller for rehabilitative robots. The controller combines the Lyapunov direct method with the computed torque control and neural networks. Robot assistance is limited to only as needed by adding the force reducing term into the adaptive control law. This paper shows that by the presented method the tracking error converges to a small value around zero while the neural network weights and system uncertainties remain bounded. Simulation on a robot manipulator model is presented to demonstrate the effectiveness of the proposed method.

1 Introduction

Strokes are one of the significant causes of disability in Australia. According to the National Stroke Foundation [National Stroke Foundation, 2016], in 2015 the number of new and recurrent strokes was more than 50,000, and it is predicted to increase to 130,000 by 2050. Furthermore, the number of New Zealanders suffering new strokes annually is around 9000 [Facts and fallacies, 2016]. Stroke survivors usually suffer a loss of control of the arm and hand, mainly through a loss of hand dexterity and motor impairments on their upper-limb movements [Richards *et al*., 2015]. To improve muscle strength and movement coordination for such patients, long duration rehabilitation with repetitive motions is required [Riener *et al*., 2005]. Presently, rehabilitation robots are accepted as satisfactory platforms for recovery of the brain motor function in patients with neurological injuries [Chase, 2014]. They can offer consistent repetitive therapy with slight supervision. They can provide the possibility to measure the improvement in skills very accurately, as well. Robotic rehabilitation can be considered as a potential solution for the problem of "*movement training therapist shortages*" in the near future.

Over the past two decades, various end-effector based [Rosati *et al*., 2007; Schoone *et al*., 2007; Spencer *et al*., 2008] or exoskeletal based [Nef *et al*., 2009; Perry *et al*., 2007; Sanchez Jr *et al*., 2005] robotic devices were designed for upper-extremity rehabilitative movement training. The review on robotic system for upper limb rehabilitation can be found in [Brackenridge *et al*., 2016; Brewer *et al*., 2007; Maciejasz *et al*., 2014]. However, although different robots were designed for rehabilitation, a paramount aspect for robots potentiality lays on a control side [Proietti *et al*., 2016]. In fact, control strategies addressing neurorehabilitation can dictate the human-robot interactions. The desirable controller for robot-aided movement training following stroke has the ability to assist patients in completing desired movements, and the ability to provide only the minimum necessary assistance [Wolbrecht *et al*., 2008].

Reviewing the literature on rehabilitative robot control shows that assist-as-needed (AAN) algorithms exhibit great progress in recent rehabilitation robotic control [Pehlivan *et al*., 2016]. In these control strategies, robots assist the patient to perform the training movement only as needed. As a result, the patient is encouraged to provide significant effort which leads to an increase in the patient engagement in therapy. This helps in inducing neural plasticity to facilitate recovery [Blank *et al*., 2014]. Emken *et al*. [Emken *et al*., 2005] derived an adaptive AAN controller using an established model of human motor adaptation. Wolbrecht *et al*. [Wolbrecht *et al*., 2008] introduced the force decreasing term to the adaption law to refine the control for the AAN purpose. They showed that adding forgetting terms in the adaptive law resulted in higher levels of patient involvement in rehabilitation [Wolbrecht *et al*., 2007]. Another important modification of this research was using the Gaussian radial basis functions (RBFs) neural networks for the estimation purpose. Rosati *et al*. [Rosati *et al*., 2008] improved controller performance of [Wolbrecht *et al*., 2008] through AAN compliant control by splitting up the target motion into multiple parts and considering a separate parameter estimator for each segment. In [Guidali *et al*., 2011] and [Bower *et al*., 2013] the estimation abilities of [Wolbrecht *et al*., 2008] were improved through directionally dependent RBFs.

This paper is motivated by the concept of "*assistive robot for upper-limb rehabilitation in human friendly environment*", an area where the AAN control is highly applicable. A new AAN controller was developed, borrowing the idea of the force decreasing term from [Wolbrecht *et al*., 2008]. The proposed scheme takes advantage of the computed torque control based on a known nominal robot dynamic model, and further utilizes RBFs neural networks to compensate the uncertain parts of the computed torque as well as the unknown interaction force. Lyapunov theory is employed to stability analysis for training the neural networks. The assist-as-needed strategy is conducted by adding the forgetting term into the adaptive law. The proposed control scheme shows that the error signals converge to a small neighborhood of zero. Compared with [Wolbrecht *et al*., 2008], in the presented control design, boundedness of the neural network weights are achieved. Using this property and utilizing a useful Lemma, the boundedness of the uncertain parts is proven and it is further shown that all closed-loop signals remain bounded. Finally, a simulation study is performed on a robot manipulator model to demonstrate the effectiveness of the proposed method.

2 Preliminaries and Problem Formulation

2.1 Preliminaries

Neural Network

Due to its approximation property we employ the following RBFs neural networks to estimate the system uncertainties [Broomhead and Lowe, 1988; Liu, 2013; Yu *et al*., 2011]. Consider a continuous function $\boldsymbol{f}(\boldsymbol{Z}):\mathbb{R}^s \to \mathbb{R}$, where using RBFs neural networks it can be approximated by

$$\boldsymbol{f} = \boldsymbol{\omega}^T \boldsymbol{h}(\boldsymbol{Z}), \qquad (1)$$

where $\boldsymbol{Z} \in \mathbb{R}^s$ is the neural network input vector ($\boldsymbol{s}$ being the neural network input dimension), $\boldsymbol{\omega} \in \mathbb{R}^{\boldsymbol{b}}$ is the weight vector ($\boldsymbol{b} > 1$ is the neural network node number), $\boldsymbol{h}(\boldsymbol{Z}) \in \mathbb{R}^{\boldsymbol{b}}$ is a basis function vector with Gaussian functions $\boldsymbol{h}_i(\boldsymbol{Z})$ express as $\boldsymbol{h}_i(\boldsymbol{Z}) = \exp\left(-\left(\boldsymbol{Z}-\boldsymbol{\lambda}_i\right)^T \left(\boldsymbol{Z}-\boldsymbol{\lambda}_i\right)/\delta^2\right)$, for $\boldsymbol{i} = 1,\ldots,\boldsymbol{b}$, with $\boldsymbol{\lambda}_i = \left[\lambda_{i1}, \lambda_{i2}, \ldots, \lambda_{is}\right]^T$ the center of the $\boldsymbol{i}^{th}$ input element of the neural networks, and $\boldsymbol{\delta}$ being the width of the Gaussian function.

In general, if $\boldsymbol{b}$ is sufficiently large, the RBFs neural network given by (1) can estimate any continuous function in the form of,

$$\boldsymbol{f}(\boldsymbol{Z}) = \boldsymbol{\omega}^{*T} \boldsymbol{h}(\boldsymbol{Z}) + \boldsymbol{\varepsilon}(\boldsymbol{Z}),\ \forall\, \boldsymbol{Z} \in \mathbb{R}^s, \qquad (2)$$

where, $\boldsymbol{\omega}^*$ is an unknown ideal constant weight vector, and $\boldsymbol{\varepsilon} \in \mathbb{R}$ is the approximation error which is bounded, i.e., $|\boldsymbol{\varepsilon}| \le \boldsymbol{\varepsilon}^*$, with $\boldsymbol{\varepsilon}^*$ being an unknown positive constant.

Lemma 1 [Kurdila *et al*., 1995; Wang *et al*., 2006]:

Consider the Gaussian RBFs neural networks (1) and let $\boldsymbol{s}$ be the dimension of neural input $\boldsymbol{Z}$, and $\boldsymbol{\delta}$ be the width of Gaussian function; further let $\boldsymbol{\eta} = (1/2)\min_{i\ne j}\left\|\boldsymbol{\lambda}_i - \boldsymbol{\lambda}_j\right\|$, then an upper bound of $\boldsymbol{h}(\boldsymbol{Z})$ is taken as

$$\boldsymbol{h}(\boldsymbol{Z}) \le \sum_{\boldsymbol{r}=0}^{\infty} 3\boldsymbol{s}(\boldsymbol{r}+2)^{\boldsymbol{s}-1} \boldsymbol{e}^{-2\boldsymbol{\eta}^2 \boldsymbol{r}^2/\delta^2}. \qquad (3)$$

2. 2 Problem Formulation

Consider an $\boldsymbol{n}$ degree of freedom robotic system as,

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{V}(\boldsymbol{q}) = \boldsymbol{T} + \boldsymbol{F}_h, \qquad (4)$$

where $\boldsymbol{q} \in \mathbb{R}^{\boldsymbol{n}}$ is the robot generalised coordinate vectors, $\boldsymbol{H}(\boldsymbol{q}) \in \mathbb{R}^{\boldsymbol{n}\times\boldsymbol{n}}$ denotes the inertia matrix, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{\boldsymbol{n}\times\boldsymbol{n}}$ represents the centrifugal and Coriolis forces matrix, $\boldsymbol{G}(\boldsymbol{q}) \in \mathbb{R}^{\boldsymbol{n}}$ is the gravitational force/torque vector; $\boldsymbol{F}_h \in \mathbb{R}^{\boldsymbol{n}}$ is the effect of the robot-patient's interaction force on each joint, and $\boldsymbol{\tau} \in \mathbb{R}^{\boldsymbol{n}}$ denotes the external force/torque vector.

Note that in this study, the AAN controller is designed for the system model presented by (4), which describes the general dynamic model of the robotic system. Accordingly, the presented control can be applied on various robotic models having numbers of both the revolute joint and/or the prismatic joint.

In reality, due to its complex structure, the perfect dynamic model of the robot is very difficult to obtain. Thus, the dynamic equation governed by (4) may not cover all the robot's accessories and small parts perfectly. To solve this difficulty, in the presented study, we use the nominal model of the robot denoted by $\boldsymbol{H}_0(\boldsymbol{q})$, $\boldsymbol{C}_0(\boldsymbol{q},\dot{\boldsymbol{q}})$ and $\boldsymbol{V}_0(\boldsymbol{q})$ to design the controller.

Property 1: Nominal matrixes $\boldsymbol{C}_0(\boldsymbol{q},\dot{\boldsymbol{q}})$ and $\boldsymbol{V}_0(\boldsymbol{q})$ are assumed to be bounded. Also, $\boldsymbol{H}_0(\boldsymbol{q})$ is a positive definite symmetric matrix and bounded by

$$\boldsymbol{H}_0(\boldsymbol{q}) \le \boldsymbol{h}_0 \boldsymbol{I},$$

where $\boldsymbol{I}$ is the $\boldsymbol{n}\times\boldsymbol{n}$ identity matrix and $\boldsymbol{h}_0$ is a known positive constant.

Defining the uncertain parts of the robot by $\Delta\boldsymbol{H} = \boldsymbol{H}_0 - \boldsymbol{H}$, $\Delta\boldsymbol{C} = \boldsymbol{C}_0 - \boldsymbol{C}$, and $\Delta\boldsymbol{V} = \boldsymbol{V}_0 - \boldsymbol{V}$; then (4) can be rewritten as

$$\boldsymbol{H}_0(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}_0(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{V}_0(\boldsymbol{q}) + \boldsymbol{F}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}}) = \boldsymbol{T} + \boldsymbol{F}_h, \qquad (5)$$

where,

$$\boldsymbol{F}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}}) = -\left(\Delta\boldsymbol{H}\ddot{\boldsymbol{q}} + \Delta\boldsymbol{C}\dot{\boldsymbol{q}} + \Delta\boldsymbol{V}\right). \qquad (6)$$

The goal of this study is to design the stable AAN controller for the nominal model given by (5) with both known and unknown system dynamic models and the interaction force. To control the robotic system with guarantee of tracking performance, an adaptive controller based on the computed torque method is developed and RBFs neural networks are employed for handling the uncertainties. Then, we improve the controller to support robotic rehabilitation by adding the AAN force decaying term.

For simplifying notation, from this point onwards, whenever no confusion would arise we omit the time and state dependence of the system.

3 Controller Design for the Known System

The position, velocity and acceleration tracking errors can be defined by $\boldsymbol{e}=\boldsymbol{q}-\boldsymbol{q}_d$, $\dot{\boldsymbol{e}}=\dot{\boldsymbol{q}}-\dot{\boldsymbol{q}}_d$ and $\ddot{\boldsymbol{e}}=\ddot{\boldsymbol{q}}-\ddot{\boldsymbol{q}}_d$, respectively where $\boldsymbol{q}_d$, $\dot{\boldsymbol{q}}_d$ and $\ddot{\boldsymbol{q}}_d$ stand for bounded vectors of the desired joint position, velocity and acceleration, respectively. Choose the control law for the known system dynamics, $\boldsymbol{F}$, and known interaction force as

$$\boldsymbol{T}=\boldsymbol{H}_0\left(\ddot{\boldsymbol{q}}_d-\boldsymbol{k}_v\dot{\boldsymbol{e}}-\boldsymbol{k}_p\boldsymbol{e}\right)+\boldsymbol{C}_0\dot{\boldsymbol{q}}+\boldsymbol{V}_0+\boldsymbol{F}(.)-\boldsymbol{F}_h, \quad (7)$$

where $\boldsymbol{k}_p$ and $\boldsymbol{k}_v$ are the proportional and derivative gain matrices, respectively.

By substituting (7) into (5) the closed loop system is obtained as

$$\ddot{\boldsymbol{e}}+\boldsymbol{k}_v\dot{\boldsymbol{e}}+\boldsymbol{k}_p\boldsymbol{e}=0\,. \quad (8)$$

To guarantee that the tracking performance of (8) is asymptotically tending to the desired trajectory, one can easily chose $\boldsymbol{k}_p$ and $\boldsymbol{k}_v$, so that the polynomial

$$\boldsymbol{S}^2+\boldsymbol{k}_v\boldsymbol{S}+\boldsymbol{k}_p=0 \quad (9)$$

is a Hurwitz polynomial, where $\boldsymbol{S}$ is the Laplace operator.

4 Controller Design with Handling Uncertainties

Due to the existence of several small and geometrically complex parts, deriving the accurate dynamic behaviour of the robotic model is practically impossible. In addition, in many cases measuring the exact value of the patient-robot interaction force is impossible or very hard to obtain. Therefore a strategy to handle the system uncertainties must be considered. We employed RBFs neural networks to cope with the un-modelled dynamics of the robotic system in addition to unknown patient contributed forces. The details of the RBFs neural networks developed for this study is available at the Section 2. We define the function $\boldsymbol{f}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})$ to include all uncertainties of the system as:

$$\begin{aligned}\boldsymbol{f}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})&=\boldsymbol{H}_0^{-1}\left(\boldsymbol{F}-\boldsymbol{F}_h\right)\\&=-\boldsymbol{H}_0^{-1}\left(\Delta\boldsymbol{H}\ddot{\boldsymbol{q}}+\Delta\boldsymbol{C}\dot{\boldsymbol{q}}+\Delta\boldsymbol{V}+\boldsymbol{F}_h\right).\end{aligned} \quad (10)$$

Estimation of $\boldsymbol{f}$ using RBFs neural networks can be given by

$$\hat{\boldsymbol{f}}=\hat{\boldsymbol{\omega}}^T\boldsymbol{h}, \quad (11)$$

where, $\hat{\bullet}$ represents the estimation value of $\bullet$.

Choose the desired control in the uncertain case as

$$\boldsymbol{T}=\boldsymbol{H}_0\left(\ddot{\boldsymbol{q}}_d-\boldsymbol{k}_v\dot{\boldsymbol{e}}-\boldsymbol{k}_p\boldsymbol{e}+\hat{\boldsymbol{f}}\right)+\boldsymbol{C}_0\dot{\boldsymbol{q}}+\boldsymbol{V}_0, \quad (12)$$

and let $\boldsymbol{E}=\left[\boldsymbol{e}\quad\dot{\boldsymbol{e}}\right]^T$; then, substituting (12) into (5) one can obtain

$$\dot{\boldsymbol{E}}=\Pi\boldsymbol{E}+\Gamma\left(\boldsymbol{\varepsilon}-\tilde{\boldsymbol{\omega}}^T\boldsymbol{h}\right), \quad (13)$$

where $\Pi=\begin{bmatrix}0 & \boldsymbol{I}\\ -\boldsymbol{k}_p & -\boldsymbol{k}_v\end{bmatrix}$, and $\Gamma=\begin{bmatrix}0\\ -\boldsymbol{H}_0^{-1}\end{bmatrix}$. The modelling error $\boldsymbol{\varepsilon}$ can be defined as $\boldsymbol{\varepsilon}=\boldsymbol{f}-\hat{\boldsymbol{f}}$ and will be bounded by the precision parameter $\boldsymbol{\varepsilon}^*=\sup\left\|\boldsymbol{f}-\hat{\boldsymbol{f}}\right\|$. Further the parameter estimation error $\tilde{\boldsymbol{\omega}}$ can be defined as $\tilde{\boldsymbol{\omega}}=\hat{\boldsymbol{\omega}}-\boldsymbol{\omega}^*$, where the ideal weight vector, $\boldsymbol{\omega}^*$, can be expressed as $\boldsymbol{\omega}^*=\arg\min_{\boldsymbol{\omega}\in\mathbb{R}^b}\left\{\sup\left|\hat{\boldsymbol{f}}-\boldsymbol{f}\right|\right\}$.

Choose the Lyapunov function candidate as

$$\boldsymbol{V}=\frac{1}{2}\boldsymbol{E}^T\boldsymbol{P}\boldsymbol{E}+\frac{1}{2\gamma}\operatorname{tr}\left[\tilde{\boldsymbol{\omega}}^T\tilde{\boldsymbol{\omega}}\right], \quad (14)$$

with the adaption law

$$\dot{\hat{\boldsymbol{\omega}}}=\gamma\boldsymbol{h}\boldsymbol{E}^T\boldsymbol{P}\Gamma, \quad (15)$$

where $\gamma>0$, $\boldsymbol{P}=\boldsymbol{P}^T>0$, and satisfying $\boldsymbol{P}\Pi+\Pi^T\boldsymbol{P}=-\boldsymbol{Q}$, where $\boldsymbol{Q}\geq 0$.

By differentiating the Lyapunov function (14) with respect to time we have

$$\begin{aligned}\dot{\boldsymbol{V}}&=\frac{1}{2}\left(\boldsymbol{E}^T\boldsymbol{P}\dot{\boldsymbol{E}}+\dot{\boldsymbol{E}}^T\boldsymbol{P}\boldsymbol{E}\right)+\frac{1}{\gamma}\operatorname{tr}\left[\dot{\tilde{\boldsymbol{\omega}}}^T\tilde{\boldsymbol{\omega}}\right]\\&=\frac{1}{2}\begin{pmatrix}\boldsymbol{E}^T\boldsymbol{P}\left(\Pi\boldsymbol{E}+\Gamma\left(\boldsymbol{\varepsilon}-\tilde{\boldsymbol{\omega}}^T\boldsymbol{h}\right)\right)\\+\left(\boldsymbol{E}^T\Pi^T+\left(\boldsymbol{\varepsilon}^T-\boldsymbol{h}^T\tilde{\boldsymbol{\omega}}\right)\Gamma^T\right)\boldsymbol{P}\boldsymbol{E}\end{pmatrix}+\frac{1}{\gamma}\operatorname{tr}\left[\dot{\tilde{\boldsymbol{\omega}}}^T\tilde{\boldsymbol{\omega}}\right]\\&=\frac{1}{2}\begin{pmatrix}\boldsymbol{E}^T\left(\boldsymbol{P}\Pi+\Pi^T\boldsymbol{P}\right)\boldsymbol{E}\\+\left(\boldsymbol{E}^T\boldsymbol{P}\Gamma\boldsymbol{\varepsilon}+\boldsymbol{\varepsilon}^T\Gamma^T\boldsymbol{P}\boldsymbol{E}\right)\\-\left(\boldsymbol{E}^T\boldsymbol{P}\Gamma\tilde{\boldsymbol{\omega}}^T\boldsymbol{h}+\boldsymbol{h}^T\tilde{\boldsymbol{\omega}}\Gamma^T\boldsymbol{P}\boldsymbol{E}\right)\end{pmatrix}+\frac{1}{\gamma}\operatorname{tr}\left[\dot{\tilde{\boldsymbol{\omega}}}^T\tilde{\boldsymbol{\omega}}\right]\\&=-\frac{1}{2}\boldsymbol{E}^T\boldsymbol{Q}\boldsymbol{E}+\boldsymbol{E}^T\boldsymbol{P}\Gamma\boldsymbol{\varepsilon}-\boldsymbol{h}^T\tilde{\boldsymbol{\omega}}\Gamma^T\boldsymbol{P}\boldsymbol{E}+\frac{1}{\gamma}\operatorname{tr}\left[\dot{\tilde{\boldsymbol{\omega}}}^T\tilde{\boldsymbol{\omega}}\right].\end{aligned} \quad (16)$$

Using (16) and noting that $\boldsymbol{h}^T\tilde{\boldsymbol{\omega}}\Gamma^T\boldsymbol{P}\boldsymbol{E}=\operatorname{tr}\left(\Gamma^T\boldsymbol{P}\boldsymbol{E}\boldsymbol{h}^T\tilde{\boldsymbol{\omega}}\right)$, we can obtain

$$\begin{aligned}\dot{\boldsymbol{V}}=&-\frac{1}{2}\boldsymbol{E}^T\boldsymbol{Q}\boldsymbol{E}+\boldsymbol{E}^T\boldsymbol{P}\Gamma\boldsymbol{\varepsilon}\\&+\frac{1}{\gamma}\operatorname{tr}\left[\dot{\tilde{\boldsymbol{\omega}}}^T\tilde{\boldsymbol{\omega}}-\gamma\,\Gamma^T\boldsymbol{P}\boldsymbol{E}\boldsymbol{h}^T\tilde{\boldsymbol{\omega}}\right].\end{aligned} \quad (17)$$

Noting that the ideal weight vector, $\boldsymbol{\omega}^*$, is assumed to be constant, thus $\dot{\tilde{\boldsymbol{\omega}}}=\dot{\hat{\boldsymbol{\omega}}}$. Then, it can be verified easily from the adaption law given by (15), that

$$\dot{V} = -\frac{1}{2}\boldsymbol{E}^T\boldsymbol{Q}\boldsymbol{E} + \boldsymbol{E}^T\boldsymbol{P}\Gamma\boldsymbol{\varepsilon}\,. \tag{18}$$

From the property 1, one can obtain $\left\|\boldsymbol{H}_0^{-1}\right\| \le \frac{1}{\boldsymbol{h}_0}$, and noting that $\left\|\boldsymbol{\varepsilon}\right\| \le \left\|\boldsymbol{\varepsilon}^*\right\|$, then, the following inequality holds:

$$\begin{aligned}\dot{V} &\le -\frac{1}{2}\lambda_{\min}\left(\boldsymbol{Q}\right)\left\|\boldsymbol{E}\right\|^2 + \frac{1}{\boldsymbol{h}_0}\left\|\boldsymbol{E}\right\|\lambda_{\max}\left(\boldsymbol{P}\right)\left\|\boldsymbol{\varepsilon}^*\right\| \\ &\le -\frac{1}{2}\left\|\boldsymbol{E}\right\|\left(\lambda_{\min}\left(\boldsymbol{Q}\right)\left\|\boldsymbol{E}\right\| - 2\frac{1}{\boldsymbol{h}_0}\lambda_{\max}\left(\boldsymbol{P}\right)\left\|\boldsymbol{\varepsilon}^*\right\|\right).\end{aligned} \tag{19}$$

To guarantee $\dot{V} \le 0$, then $\lambda_{\min}\left(\boldsymbol{Q}\right)\left\|\boldsymbol{E}\right\| \ge 2\frac{1}{\boldsymbol{h}_0}\lambda_{\max}\left(\boldsymbol{P}\right)\left\|\boldsymbol{\varepsilon}^*\right\|$ that is $\left\|\boldsymbol{E}\right\| \ge 2\frac{\lambda_{\max}\left(\boldsymbol{P}\right)}{\lambda_{\min}\left(\boldsymbol{Q}\right)}\frac{\left\|\boldsymbol{\varepsilon}^*\right\|}{\boldsymbol{h}_0}$. Thus, using the method presented in this paper, the asymptotically stability of the system cannot be guaranteed. However, it is shown that the system is stable in the sense of uniform ultimate boundedness with the convergence boundary of $\boldsymbol{E}_c = 2\frac{\lambda_{\max}\left(\boldsymbol{P}\right)}{\lambda_{\min}\left(\boldsymbol{Q}\right)}\frac{\left\|\boldsymbol{\varepsilon}^*\right\|}{\boldsymbol{h}_0}$. Note that $\boldsymbol{E}_c$ is the maximum error for $\dot{V}\left(t\right) > 0$, thus larger magnitudes of $\boldsymbol{E}_c$ will lead to $\dot{V}\left(t\right) < 0$, and the closed-loop system will then converge to this boundary.

Theorem 1: *Consider property 1 and let the desired joint trajectories* $\boldsymbol{q}_d$, $\dot{\boldsymbol{q}}_d$ and $\ddot{\boldsymbol{q}}_d$ *be bounded and the neural network modelling error bound,* $\boldsymbol{\varepsilon}^*$, *will be constant. For the system given by (5) with the control (12), consider the Lyapunov function (14) with the adaption law (15), then,*

(i) The tracking error $\boldsymbol{e}\left(t\right)$ *belongs to a residue of radius* $\boldsymbol{r} = \varsigma\left\|\boldsymbol{\varepsilon}^*\right\|$, *where* $\varsigma = 2\lambda_{\max}\left(\boldsymbol{P}\right)/\boldsymbol{h}_0\lambda_{\min}\left(\boldsymbol{Q}\right)$, *and* $\lambda_{\max}\left(\bullet\right)$, *and* $\lambda_{\min}\left(\bullet\right)$ *denotes the maximum and the minimum eigenvalues of the matrix* $\bullet$, *respectively.*

(ii) The control ($\boldsymbol{T}$*) is smooth.*

Proof:

Part (*i*) follows from the application of Lyapunov's direct method from (14) to (19).

Part (*ii*) follows directly from the construction of the Lyapunov function in (14) and (19), and the control, $\boldsymbol{T}$, in (12) and the corresponding equations. □

5 Controller Design with Assist-As-Needed Modification

Human-robot interaction, in the sense of AAN control, is considered by modifying the conventional adaptive law (15). The modified AAN adaption law is formed as

$$\dot{\hat{\boldsymbol{\omega}}} = \gamma\boldsymbol{h}\boldsymbol{E}^T\boldsymbol{P}\Gamma + \frac{1}{\tau}\gamma\left\|\boldsymbol{E}\right\|\Omega\hat{\boldsymbol{\omega}}\,, \tag{20}$$

where τ is the time constant. In the adaptive law constructed by (20), the first term on the right side reduces the tracking error while the second term is the AAN term, designed to reduce the robot-patient force. The forgetting rate, $\frac{1}{\tau}$, is designed to weight the balance between the error and assistance provided by the rehabilitative robot. In this paper, inspired by [Wolbrecht *et al*., 2008] the matrix Ω is chosen as $\Omega = \boldsymbol{h}\left(\boldsymbol{h}^T\boldsymbol{h}\right)^{-1}\boldsymbol{h}^T$. As discussed in [Wolbrecht *et al*., 2008], Ω in the AAN term in (20), limits the change in parameter estimates $\hat{\boldsymbol{\omega}}$ to those with the largest current influence on the output force to keep the parameter decay local with respect to the state of $\boldsymbol{h}$. Accordingly, it causes the force decay to affect the parameter estimates associated with the RBFs when the patient does the rehabilitation therapy well. On the other hand, the parameter decay decreases as the patient trajectory and the associated RBFs are increased.

Theorem 2: *For the robot system defined by (5), under the hypotheses of Theorem 1, let the control given by (12), and the weight tuning given by (20). Then, the tracking error* $\boldsymbol{e}\left(t\right)$ *and NN weight* $\tilde{\boldsymbol{\omega}}$ *are bounded with the practical bounds given by the right-hand side of (25) and (26), respectively. Also, all closed-loop signals remained bounded.*

Proof:
Substitute (20) into (17), and noting $\boldsymbol{h}^T\tilde{\boldsymbol{\omega}}\Gamma^T\boldsymbol{P}\boldsymbol{E} = \mathrm{tr}\left(\Gamma^T\boldsymbol{P}\boldsymbol{E}\boldsymbol{h}^T\tilde{\boldsymbol{\omega}}\right)$ gives,

$$\begin{aligned}\dot{V} &= -\frac{1}{2}\boldsymbol{E}^T\boldsymbol{Q}\boldsymbol{E} + \boldsymbol{E}^T\boldsymbol{P}\Gamma\boldsymbol{\varepsilon} \\ &\quad + \mathrm{tr}\left[\frac{1}{\tau}\left\|\boldsymbol{E}\right\|\hat{\boldsymbol{\omega}}^T\Omega\tilde{\boldsymbol{\omega}}\right].\end{aligned} \tag{21}$$

We further have $\mathrm{tr}\left[\hat{\boldsymbol{\omega}}^T\Omega\tilde{\boldsymbol{\omega}}\right] = \mathrm{tr}\left[\hat{\boldsymbol{\omega}}^T\tilde{\boldsymbol{\omega}}\right]$, and $\mathrm{tr}\left[\hat{\boldsymbol{\omega}}^T\tilde{\boldsymbol{\omega}}\right] = \mathrm{tr}\left[\tilde{\boldsymbol{\omega}}^T\hat{\boldsymbol{\omega}}\right] = \mathrm{tr}\left[\tilde{\boldsymbol{\omega}}^T\left(\boldsymbol{\omega}^* + \tilde{\boldsymbol{\omega}}\right)\right] \le \left\|\tilde{\boldsymbol{\omega}}\right\|\left\|\boldsymbol{\omega}^*\right\| - \left\|\tilde{\boldsymbol{\omega}}\right\|^2$.

Letting $\boldsymbol{\varpi} = \left\|\boldsymbol{\omega}^*\right\|$ and using the definition given by Proof 1, we can reform (21) as

$$\begin{aligned}\dot{V} &\le -\frac{1}{2}\lambda_{\min}\left(\boldsymbol{Q}\right)\left\|\boldsymbol{E}\right\|^2 + \frac{1}{\boldsymbol{h}_0}\lambda_{\max}\left(\boldsymbol{P}\right)\left\|\boldsymbol{\varepsilon}^*\right\|\left\|\boldsymbol{E}\right\| \\ &\quad + \frac{1}{\tau}\left\|\boldsymbol{E}\right\|\left(\left\|\tilde{\boldsymbol{\omega}}\right\|\boldsymbol{\varpi} - \left\|\tilde{\boldsymbol{\omega}}\right\|^2\right) \\ &\le -\left\|\boldsymbol{E}\right\|\begin{pmatrix}\frac{1}{2}\lambda_{\min}\left(\boldsymbol{Q}\right)\left\|\boldsymbol{E}\right\| - \frac{1}{\boldsymbol{h}_0}\lambda_{\max}\left(\boldsymbol{P}\right)\left\|\boldsymbol{\varepsilon}^*\right\| \\ + \frac{1}{\tau}\left(\left\|\tilde{\boldsymbol{\omega}}\right\| - \frac{\boldsymbol{\varpi}}{2}\right)^2 - \frac{1}{\tau}\frac{\boldsymbol{\varpi}^2}{4}\end{pmatrix}.\end{aligned} \tag{22}$$

Thus, $\dot{V} \le 0$ is guaranteed as long as either

$$\frac{1}{2}\lambda_{\min}(\boldsymbol{Q})\|\boldsymbol{E}\| \ge \frac{\lambda_{\max}(\boldsymbol{P})\|\boldsymbol{\varepsilon}^*\|}{\boldsymbol{h}_0} + \frac{1}{\tau}\frac{\varpi^2}{4}, \tag{23}$$

or

$$\frac{1}{\tau}\left(\|\tilde{\boldsymbol{\omega}}\| - \frac{\varpi}{2}\right)^2 \ge \frac{\lambda_{\max}(\boldsymbol{P})\|\boldsymbol{\varepsilon}^*\|}{\boldsymbol{h}_0} + \frac{1}{\tau}\frac{\varpi^2}{4}. \tag{24}$$

Then, to get the boundedness for the tracking error and neural network weights, (23) and (24) can be reformed as

$$\|\boldsymbol{E}\| \ge 2\frac{\lambda_{\max}(\boldsymbol{P})}{\lambda_{\min}(\boldsymbol{Q})}\frac{\|\boldsymbol{\varepsilon}^*\|}{\boldsymbol{h}_0} + \frac{1}{2\tau}\frac{\varpi^2}{\lambda_{\min}(\boldsymbol{Q})}, \tag{25}$$

$$\|\tilde{\boldsymbol{\omega}}\| \ge \frac{\varpi}{2} + \sqrt{\frac{\tau\lambda_{\max}(\boldsymbol{P})\|\boldsymbol{\varepsilon}^*\|}{\boldsymbol{h}_0} + \frac{\varpi^2}{4}}. \tag{26}$$

Thus, both $\|\boldsymbol{E}\|$ and $\|\tilde{\boldsymbol{\omega}}\|$ are uniformly ultimately bounded.

Since $\tilde{\boldsymbol{\omega}}$ is bounded and with the use of Lemma 1, $\boldsymbol{h}$ can easily prove to be bounded, then $\hat{\boldsymbol{f}}$ is also bounded. Then, since $\boldsymbol{E} = [\boldsymbol{e}, \dot{\boldsymbol{e}}]^T$ is bounded, the control $\boldsymbol{T}$ is bounded. Also, since the desired signals $\boldsymbol{q}_d$, and $\dot{\boldsymbol{q}}_d$ are bounded, then, $\boldsymbol{q}$, and $\dot{\boldsymbol{q}}$ are bounded. Furthermore, by bounding $\dot{\boldsymbol{V}}$ as in (22), it is obvious that the Lyapunov function (14) is bounded. Therefore, boundedness of all closed-loop signals are obtained. □

6 An example of simulation

A simulation study is performed to demonstrate the performance of the presented method. A simple 2 DOF robot manipulator with two revolute joints in the vertical plane was used in the simulation. This robot is considered as a simple robot which can contribute in the upper-limb rehabilitation to verify the presented control numerically.

The neural controller with AAN modification terms as presented in Section 5 is considered in this simulation. Physical robot parameters were taken from [Korayem and Nikoobin, 2008] as: length of link 1 $\boldsymbol{l}_1 = 1\,\mathrm{m}$, length of link 2 $\boldsymbol{l}_2 = 0.5\,\mathrm{m}$, mass of link 1 $\boldsymbol{m}_1 = 2$ kg, mass of link 2 $\boldsymbol{m}_2 = 2$ kg, inertia of link 1 $\boldsymbol{I}_1 = 0.166\ \mathrm{kgm}^2$, and inertia of link 2 $\boldsymbol{I}_2 = 0.166\ \mathrm{kgm}^2$. The desired trajectories are given as $\boldsymbol{q}_d = [\sin(\boldsymbol{t}), \sin(\boldsymbol{t})]^T$, where $\boldsymbol{t} \in [0, \boldsymbol{t}_f]$, and $\boldsymbol{t}_f = 20$ s. It assumed that $\Delta\boldsymbol{H} = 0.2\boldsymbol{H}_0$, $\Delta\boldsymbol{C} = 0.2\boldsymbol{C}_0$, $\Delta\boldsymbol{V} = 0.2\boldsymbol{V}_0$; also, we consider the robot under interaction force $\boldsymbol{F}_h = 2 + 4\|\boldsymbol{e}\| + 3\|\dot{\boldsymbol{e}}\|$. To satisfy the condition that the polynomial (9) is Hurwitz, $\boldsymbol{k}_p$ and $\boldsymbol{k}_v$ are chosen as $\boldsymbol{k}_p = \boldsymbol{k}^2\boldsymbol{I}_2$ and $\boldsymbol{k}_v = 2\boldsymbol{k}\boldsymbol{I}$, where $\boldsymbol{k}$ is a small positive constant, and $\boldsymbol{I}_2$ is the 2×2 identity matrix. In addition, in this simulation a 30 layer RBFs with the input chosen by $\boldsymbol{Z} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \dot{\boldsymbol{e}}_1, \dot{\boldsymbol{e}}_2, \boldsymbol{F}_h]^T$ was employed. The initial conditions were given as $\boldsymbol{q}(0) = [0.2, 0.1]^T$, and $\dot{\boldsymbol{q}}(0) = [0.8, 0.6]^T$; other simulation parameters were chosen as $\boldsymbol{k} = 3$, $\gamma = 10$, $\tau = 100$, and $\boldsymbol{Q} = \mathrm{diag}(50)$. The results of the simulation are shown in Figs 1-6.

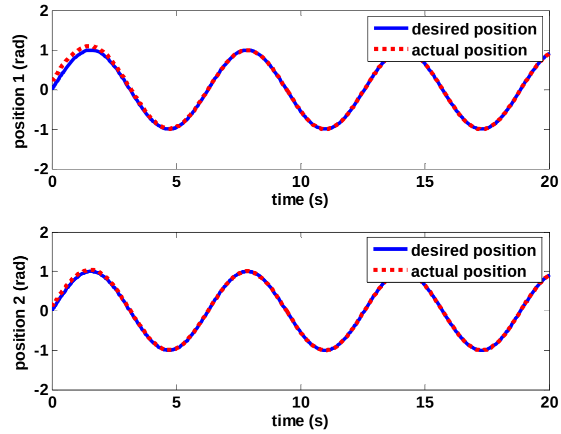

Figure 1: Desired and real position signals

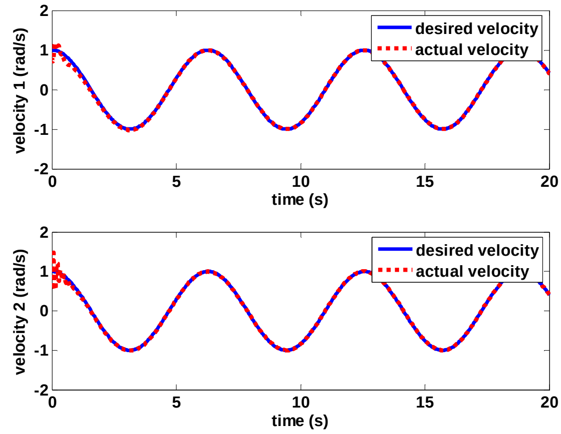

Figure 2: Desired and real velocity signals

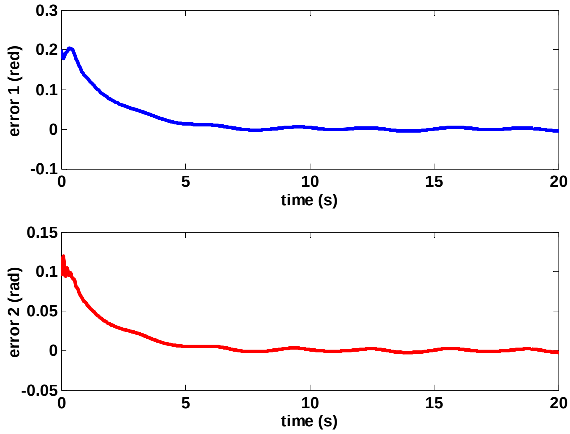

Figure 3: Error in position tracking

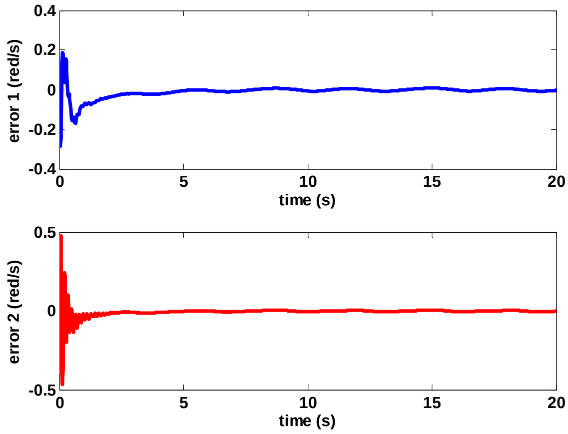

Figure 4: Error in velocity tracking

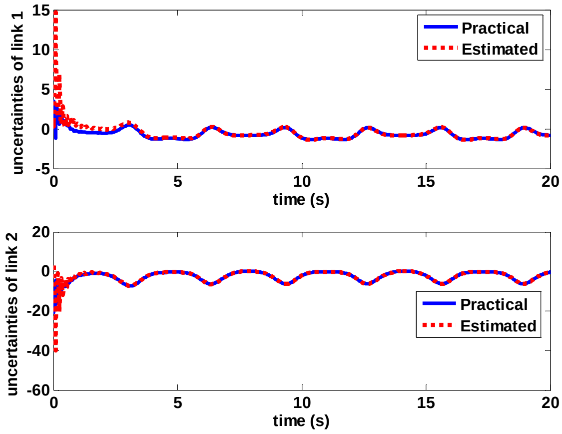

Figure 5: Practical and estimated uncertainties

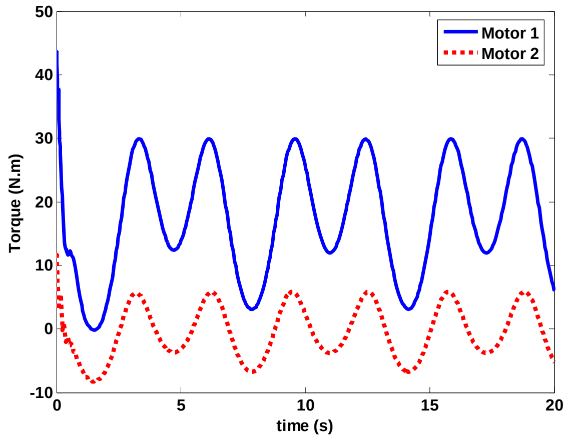

Figure 6: Control input

Figs. 1 and 2 show the tracking of positions and velocities of joints, respectively. The tracking errors for positions and velocities are shown in Figs 3, and 4, respectively. It is clear from these figures that all signals track the desired values successfully. To show the ability of the controller to estimate uncertainties, the practical and estimated system uncertainties are shown in Fig. 5. Figure 6 shows the control input signals and it is obvious from the figure that the control inputs are bounded. As it is shown in the simulation results, good tracking performances are achieved and all the closed-loop signals are bounded.

7 Conclusion

A new adaptive neural control has been presented in this work, to provide an assist-as-needed strategy for after stroke patients. The proposed scheme can effectively deal with known and unknown dynamic models of the robot and the interaction force. Using the presented control, the neural network weights are bounded, which further leads to the bounding of the system un-modelled parts and uncertainties. We showed that under the proposed control scheme, the tracking error converges to a small set around zero; while uniformly ultimately boundedness of the closed-loop system is guaranteed. Simulation results on a simple robot verified the effectiveness of the method. The presented method can be employed to control various robots for upper-limb, finger, and wrist or lower-limb rehabilitation.

References

[Blank *et al.*, 2014] Blank A A, French J A, Pehlivan A U, O'Malley M K, "Current trends in robot-assisted upper-limb stroke rehabilitation: promoting patient engagement in therapy" *Current Physical Medicine and Rehabilitation Reports* **2** 184-195, 2014.

[Bower *et al.*, 2013] Bower C, Taheri H, Wolbrecht E, "Adaptive control with state-dependent modeling of patient impairment for robotic movement therapy", in *IEEE International Conference on Rehabilitation Robotics* pp 1-6, 2013.

[Brackenridge *et al.*, 2016] Brackenridge J, V Bradnam L, Lennon S, J Costi J, A Hobbs D, "A Review of Rehabilitation Devices to Promote Upper Limb Function Following Stroke" *Neuroscience and Biomedical Engineering* **4** 25-42, 2016.

[Brewer *et al.*, 2007] Brewer B R, McDowell S K, Worthen-Chaudhari L C, "Poststroke upper extremity rehabilitation: a review of robotic systems and clinical results" *Topics in stroke rehabilitation* **14** 22-44, 2007.

[Broomhead and Lowe, 1988] Broomhead D S, Lowe D, "Radial basis functions, multi-variable functional interpolation and adaptive networks", (DTIC Document) , 1988.

[Chase, 2014] Chase A, "Neural Repair and Rehabilitation: New assistive devices for stroke rehabilitation" *Nature Reviews Neurology*, **10** 59, 2014.

[Emken *et al.*, 2005] Emken J L, Bobrow J E, Reinkensmeyer D J, "Robotic movement training as an optimization problem: designing a controller that assists only as needed", in *9th IEEE International Conference on Rehabilitation Robotics*, pp 307-312, 2005.

[Facts and fallacies, 2016] Facts and fallacies. "http://www.stroke.org.nz/", 2016. [On-line; accessed 15-November-2016].

[Guidali *et al.*, 2011] Guidali M, Schlink P, Duschau-Wicke A, Riener R, "Online learning and adaptation of patient support during ADL training", in *12th IEEE International Conference on Rehabilitation Robotics*, pp 1-6, 2011.

[Korayem and Nikoobin, 2008] Korayem M, Nikoobin A, "Maximum payload for flexible joint manipulators in point-to-point task using optimal control approach" *The International Journal of Advanced Manufacturing Technology* **38** 1045-1060, 2008.

[Kurdila *et al.*, 1995] Kurdila A, Narcowich F J, Ward J D, "Persistency of excitation in identification using radial basis function approximants" *SIAM Journal on Control and Optimization* **33** 625-642, 1995.

[Liu, 2013] Liu J "Radial Basis Function (RBF) neural network control for mechanical systems: design, analysis and Matlab simulation" *Springer Science & Business Media*, 2013.

[Maciejasz *et al.*, 2014] Maciejasz P, Eschweiler J, Gerlach-Hahn K, Jansen-Troy A, Leonhardt S, "A survey on robotic devices for upper limb rehabilitation" *Journal of neuroengineering and rehabilitation* **11** 3, 2014.

[National Stroke Foundation, 2016] National Stroke. "Foundation https://strokefoundation.org.au/", 2016. [On-line; accessed 15-November-2016].

[Nef *et al.*, 2009] Nef T, Guidali M, Riener R, "ARMin III–arm therapy exoskeleton with an ergonomic shoulder actuation" *Applied Bionics and Biomechanics* **6** 127-142, 2009.

[Pehlivan *et al.*, 2016] Pehlivan A U, Losey D P, O'Malley M K, "Minimal assist-as-needed controller for upper limb robotic rehabilitation" *IEEE Transactions on Robotics* **32** 113-124, 2016.

[Perry *et al.*, 2007] Perry J C, Rosen J, Burns S, "Upper-limb powered exoskeleton design" *Mechatronics, IEEE/ASME Transactions on* **12** 408-417, 2007.

[Proietti *et al.*, 2016] Proietti T, Crocher V, Roby-Brami A, Jarrasse N, "Upper-limb robotic exoskeletons for neurorehabilitation: a review on control strategies" *IEEE Reviews in Biomedical Engineering* **9** 4-11, 2016.

[Richards *et al.*, 2015] Richards C L, Malouin F, Nadeau S, "Stroke rehabilitation: clinical picture, assessment, and therapeutic challenge" *Progress in brain research* **218** 253-280, 2015.

[Riener *et al.*, 2005] Riener R, Nef T, Colombo G, "Robot-aided neurorehabilitation of the upper extremities" *Medical and Biological Engineering and Computing* **43** 2-10, 2005.

[Rosati *et al.*, 2008] Rosati G, Bobrow J E, Reinkensmeyer D J, "Compliant control of post-stroke rehabilitation robots: using movement-specific models to improve controller performance", in *ASME 2008 International Mechanical Engineering Congress and Exposition*, American Society of Mechanical Engineers pp 167-174, 2008.

[Rosati *et al.*, 2007] Rosati G, Gallina P, Masiero S, "Design, implementation and clinical tests of a wire-based robot for neurorehabilitation" *Neural Systems and Rehabilitation Engineering, IEEE Transactions on* **15** 560-569, 2007.

[Sanchez *et al.*, 2005] Sanchez Jr R, Wolbrecht E, Smith R, Liu J, Rao S, Cramer S, Rahman T, Bobrow J, Reinkensmeyer D, "A pneumatic robot for re-training arm movement after stroke: Rationale and mechanical design", in *9th IEEE International Conference on Rehabilitation Robotics,* pp 500-504, 2005.

[Schoone *et al.*, 2007] Schoone M, Van Os P, Campagne A, "Robot-mediated Active Rehabilitation (ACRE) A user trial", in *10th IEEE International Conference on Rehabilitation Robotics,* pp 477-481, 2007.

[Spencer *et al.*, 2008] Spencer S, Klein J, Minakata K, Le V, Bobrow J, Reinkensmeyer D, "A low cost parallel robot and trajectory optimization method for wrist and forearm rehabilitation using the Wii", in *2nd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics,* pp 869-874, 2008.

[Wang *et al.*, 2006] Wang C, Hill D J, Ge S S, Chen G, "An ISS-modular approach for adaptive neural control of pure-feedback systems" *Automatica* **42** 723-731, 2006.

[Wolbrecht *et al.*, 2007] Wolbrecht E T, Chan V, Le V, Cramer S C, Reinkensmeyer D J, Bobrow J E, "Real-time computer modeling of weakness following stroke optimizes robotic assistance for movement therapy", in *3rd International IEEE/EMBS Conference on Neural Engineering,* pp 152-158, 2007.

[Wolbrecht *et al.*, 2008] Wolbrecht E T, Chan V, Reinkensmeyer D J, Bobrow J E, "Optimizing compliant, model-based robotic assistance to promote neurorehabilitation" *Neural Systems and Rehabilitation Engineering, IEEE Transactions on* **16** 286-297, 2008.

[Yu *et al.*, 2011] Yu H, Xie T, Paszczynski S, Wilamowski B M, "Advantages of radial basis function networks for dynamic system design" *Industrial Electronics, IEEE Transactions on* **58** 5438-5450, 2011.