PROCEEDINGS

53. IWK

Internationales Wissenschaftliches Kolloquium
International Scientific Colloquium

Faculty of
Mechanical Engineering

PROSPECTS IN MECHANICAL ENGINEERING

8 - 12 September 2008

www.tu-ilmenau.de

TECHNISCHE UNIVERSITÄT
ILMENAU

Home / Index:
http://www.db-thueringen.de/servlets/DocumentServlet?id=17534

Published by
Impressum

Publisher Herausgeber	Der Rektor der Technischen Universität Ilmenau Univ.-Prof. Dr. rer. nat. habil. Dr. h. c. Prof. h. c. Peter Scharff
Editor Redaktion	Referat Marketing und Studentische Angelegenheiten Andrea Schneider
	Fakultät für Maschinenbau Univ.-Prof. Dr.-Ing. habil. Peter Kurz, Univ.-Prof. Dr.-Ing. habil. Rainer Grünwald, Univ.-Prof. Dr.-Ing. habil. Prof. h. c. Dr. h. c. mult. Gerd Jäger, Dr.-Ing Beate Schlütter, Dipl.-Ing. Silke Stauche
Editorial Deadline Redaktionsschluss	17. August 2008
Publishing House Verlag	Verlag ISLE, Betriebsstätte des ISLE e.V. Werner-von-Siemens-Str. 16, 98693 Ilmenau

CD-ROM-Version:

Implementation Realisierung	Technische Universität Ilmenau Christian Weigel, Helge Drumm
Production Herstellung	CDA Datenträger Albrechts GmbH, 98529 Suhl/Albrechts

ISBN: 978-3-938843-40-6 (CD-ROM-Version)

Online-Version:

Implementation Realisierung	Universitätsbibliothek Ilmenau ilmedia Postfach 10 05 65 98684 Ilmenau

Home / Index:
http://www.db-thueringen.de/servlets/DocumentServlet?id=17534

53rd Internationales Wissenschaftliches Kolloquium
Technische Universität Ilmenau
08 – 12 September 2008

Alexey Bulgakov / Saad Zaghlul S. Al-Khayyt

ADAPTIVE NEURAL NETWORK CONTROLLER FOR FLEXIBLE-LINK ROBOT

Abstract: This paper is concerned with the design of active trajectory tracking control structure for flexible-link manipulator based on adaptive neural network. Timoshenko beam theory is used to model the elastic behavior of the robot's arm. Then the partial differential equation of motion corresponding to continuous model is transferred into a set of second-order differential equations using the finite element method.
A simple adaptive control structure is used to dampen the end-point vibration, which is consisted from PD controller to stabilize the system and radial base function neural network with feedback error on-line learning scheme to control the elastic deflection and compute the joints torque that position the end effecter of multi-link flexible robot along prescribed trajectory. The inputs to the control system are the link's tip normal deflection and the hub velocity.
The results are presented to illustrate the advantages of the proposed controller over the rigid body torque. Simulation's results show the effectiveness of the control structure and how it is succeeded in reducing the end-point vibration during tracking and after reaching the desired position.

Keywords: Flexible-link robot; neural network; adaptive control

1. INTRODUCTION

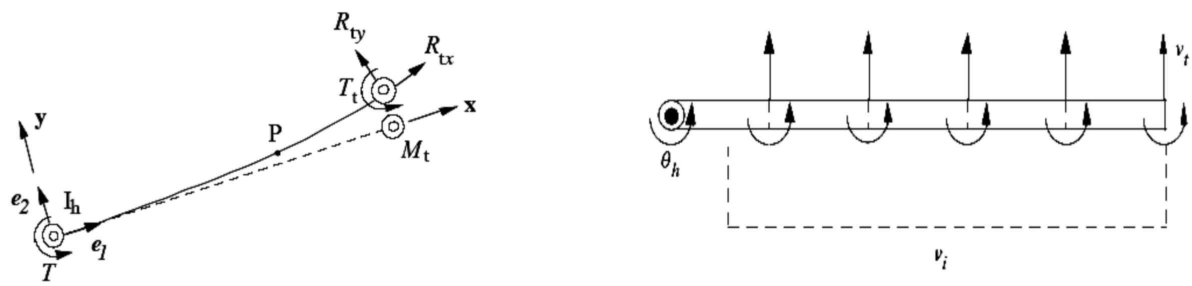
Flexible robot manipulators exhibit many advantages over rigid robots: they require less material, are lighter in weight, consume less power, require smaller actuators, are more maneuverable and transportable, have less overall cost and higher payload to robot weight ratio. Due to the flexible nature of the system, the dynamics are significantly more complex. Problems arise due to precise positioning requirements, system flexibility which leads to vibration, the difficulty in obtaining an accurate model of the system and non-minimum phase characteristics of the system [1]. If the advantages associated with lightness are not to be sacrificed, accurate models and efficient controllers have to be developed.
Various approaches have been developed previously for modelling of flexible manipulators. These can be divided into two main categories: the assumed modes method and the numerical analysis approach. The assumed modes method looks at obtaining approximate modes by solving the partial differential equation characterizing the dynamic behaviour of the system. Using this approach, the inverse kinematics problem was solved by special developed numerical technique in [3] so as to dampen the vibration of flexible manipulator. Also a control method at terminal stage of movement had developed and engaged with PID in [4].
Numerical analysis techniques include finite difference (FD) and finite element (FE) methods. Using the FE method for modelling the flexible links, a method for computing the feed foreword torque had presented in [4, 5]. The approach was based on solving the inverse dynamic problem for flexible manipulators in the frequency domain. The usage of the fast Fourier transform FFT and the necessity to iterate to obtain the required solution preclude this method from being used in real time control.
McGinley *et al.* [6] had developed a finite element model to simulate the performance of piezoelectric ceramic actuators, and was applied to a flexible two-arm manipulator system. Connected to a control voltage, the piezoelectric actuators produce control torques based on the optimal control theory. Also the finite element technique had been depended for investigating flexible link manipulator in combination with a perturbation method or fuzzy logic, respectively [7, 8]. These investigations had shown that the finite element method can be used to obtain a good representation of the system.

Simulation and experimental results of the response of flexible manipulator were presented in [9]. It had reported that using approximate modes method dose not always represent the fine details of the system; while in FM method, a single element is sufficient to describe the dynamic behaviour of a flexible manipulator reasonably well.

In this present work the highly nonlinear dynamic behavior of flexible robot manipulator is analyzed using the FM method. The formulations includes all the nonlinear Corioles and centrifugal effects. A control structure is proposed that uses PD controller combined with an observer for velocity estimation (position measurement only is required for feed back), while the adaptive neural-network is trained on-line to control the elastic deflection and compensate the system nonlinearities.

2. EQUATIONS OF MOTION OF AN INDIVIDUAL LINK [10]

The individual flexible link depicted in Figure 1.a forms part of a planar multilink manipulator and has a total length L, mass per unit length $\overline{m}$, moment of inertia I, area A, Young modulus E, shear modulus G, and shear coefficient ?. A tip mass of value M_t is attached at one end, and a hub of inertia I_h at the other end. The hub is attached to the actuator. T is the unknown torque to be applied at the hub. R_{ty}, R_{tx}, and T_t are the reaction forces and the torque at the tip that comes from the next link. The subscripts h and t indicate hub and tip, respectively. A point P at a distance x from the center of the hub has undergone elastic deflections of value u_x and u_y and rotation ?. These are defined with respect to a nominal position characterized by the moving frame ($\mathbf{e}_1$, $\mathbf{e}_2$) attached to the hub, that rotates at a specified (nominal) angular velocity and acceleration ω_h and $\dot{\omega}_h$, respectively.



a b

a- Nominal motion (dashed) and elastic deformation (solid) b- Finite element partition for flexible link.

Figure 1. Flexible link manipulator

As a consequence of the elastic deflections and rotating nominal motion, the point P is subjected to a total translational acceleration $\mathbf{a}_p$ and angular acceleration $\dot{\omega}_p$. Using the principles of relative motion, the acceleration of the point P can be set in terms of the translation and angular accelerations at the hub, $\mathbf{a}_h$ and $\dot{\omega}_h$, angular velocity ω_h at the hub, and the relative velocity $\mathbf{v}_{rel}$ and acceleration $\mathbf{a}_{rel}$ of point P. The latter are due to the elastic deflections u_x and u_y with respect to the moving frame. In vectorial notation:

$$\mathbf{a_p} = ?_{\mathbf{h}} \wedge (?_{\mathbf{h}} \wedge \bar{\mathbf{r}}_{\mathbf{p}}) + ?_{\mathbf{h}} \wedge \bar{\mathbf{r}}_{\mathbf{p}} + 2?_{\mathbf{h}} \wedge \mathbf{v_{rel}} + \mathbf{a_h} + \mathbf{a_{rel}}$$
$$?_{\mathbf{p}} = ?_{\mathbf{h}} + \ddot{?} \tag{1}$$

where $\bar{\mathbf{r}}_{\mathrm{p}} = (x + u_x)\,\mathbf{e}_1(t) + u_y \mathbf{e}_2(t)$ is the position of P after deformation, relative to the hub.

The components of the relative velocity are $\dot{u}_x$ and $\dot{u}_y$. Those of the relative acceleration are $\ddot{u}_x$ and $\ddot{u}_y$. Performing the vectorial operations involved in Eq.(1) the following components of the accelerations are obtained:

$$
\begin{aligned}
a_x &= -\omega_h^2\, u_x - \dot{\omega}_h\, u_y - 2\,\omega_h\, \dot{u}_y + \ddot{u}_x - \omega_h^2\, x + a_{hx} \\
a_y &= -\omega_h^2\, u_y - \dot{\omega}_h\, u_x + 2\,\omega_h\, \dot{u}_x + \ddot{u}_y + \omega_h^2\, x + a_{hy} \\
\dot{\omega}_p &= \dot{\omega}_h + \ddot{\theta}
\end{aligned}
\tag{2}
$$

Using the Timoshenko beam theory which includes the effects of shear deformation and rotatory inertia, the principle of virtual displacements can be used directly to set up the equations of motion. Then the displacement field can be discretized using the finite element under *pin-free* boundary conditions (Figure 1.b). A set of interpolation functions are defined within each body:

$$
u_x(x,t) = \sum_1^n H_i(x)\, u_x^i(t)\ , \quad u_y(x,t) = \sum_1^n H_i(x)\, u_y^i(t)\ , \quad \theta(x,t) = \sum_1^n H_i(x)\,\theta^i(t) \tag{3}
$$

Where H_i are the interpolation functions; u_x^i, x_y^i, θ^i indicate the nodal or generalized deflection, and n is the number of nodes.
Then following the standard procedures for the formation and assemblage of element matrices, the equations of motion of the link may be expressed by a set of time varying differential equations in the form:

$$
\mathbf{M}\,\ddot{\mathbf{v}} + [\,\mathbf{C} + \mathbf{C_c}(\boldsymbol{\omega}_\mathbf{h})]\,\dot{\mathbf{v}} + [\mathbf{K} + \mathbf{K_c}(\dot{\boldsymbol{\omega}}_\mathbf{h}, \boldsymbol{\omega}_\mathbf{h})]\,\mathbf{v} = \mathbf{T} - \mathbf{F}(\dot{\boldsymbol{\omega}}_\mathbf{h}, \boldsymbol{\omega}_\mathbf{h}) \tag{4}
$$

where **M** and **K** are the conventional finite element mass and stiffness matrices, respectively. $\mathbf{C_c}$ and $\mathbf{K_c}$ are the time varying Coriolis and centrifugal stiffness matrices that depend on the nominal angular velocity ω_h and acceleration $\dot{\omega}_h$ of the link. Matrix **C** has been added to represent the internal viscous damping of the material. Vector **T** contains one non-zero term only, and that is the unknown torque at the hub. Finally, **F** contains the reactions and the torque at the end of the link and the known forces produced by the rotating frame effect. Also in our presentation, we will refer to v_t as the elastic normal deflection at the tip, and $\mathbf{v_i}$ as all the other internal finite element elastic degrees of freedom of the manipulator.

3. CONTROL DESING

It is well known that most of the industrial robot manipulators are equipped with the simplest PD controller. The PD controller requires measurements of both link's positions and velocities. It is very important to realize the PD control scheme with only position measurement. One of the possible methods is to use a velocity observer. The most popular model-free observers are high-gain observers, which can estimate the derivative of the output [11].
If the link velocity (x_2) is not measurable and the dynamics of the system are unknown, a high gain observer may be used to estimate x_2. The high gain observer is of the form shown below [12]:

$$
\begin{aligned}
\dot{\hat{x}}_1 &= \hat{x}_2 + \frac{1}{e} L_1 (x_1 - \hat{x}_1) \\
\dot{\hat{x}}_2 &= \frac{1}{e^2} L_2 (x_1 - \hat{x}_1)
\end{aligned}
\tag{5}
$$

where

$x_1 \in \Re^n$ The vector of hub angular position.

$\hat{x}_1 \in \Re^n$ The vector of estimated hub angular position.

$\hat{x}_2 \in \Re^n$ The vector of estimated hub angular velocity.

e Small positive parameter.
L_1 & L_2 Positive definite matrices.

Feedback Error Neural Learning Technique (FENL)

An adaptive control can be constructed by using the feedback error neural learning (FENL) technique. In the FENL structure, the reference value (x_1^d) and the output of the system (x_1) adopted as the inputs of the neural network (NN) in order to learn the inverse dynamics of the control systems [13]. The objective of the control is to minimize the error (e) which has defined as the difference between the reference (desired) value (x_1^d) and the output of the system (x_1). In the FENL method, the output of the conventional feedback controller (u_c) is used for calculating the NN output error (Figure 2). When the output of the conventional controller (u_c) becomes zero, the error also becomes zero. Therefore, the objective of the control is satisfied. Also the total control signal (u) which will be applied to the system becomes the sum of the conventional output (u_c) and the neural network output (u_{NN}):
In [14] reported that the above control scheme had effectively compensated the effects of friction, gravity, and system nonlinearities for rigid robot-manipulator.

$$u = u_c + u_{NN} \tag{6}$$

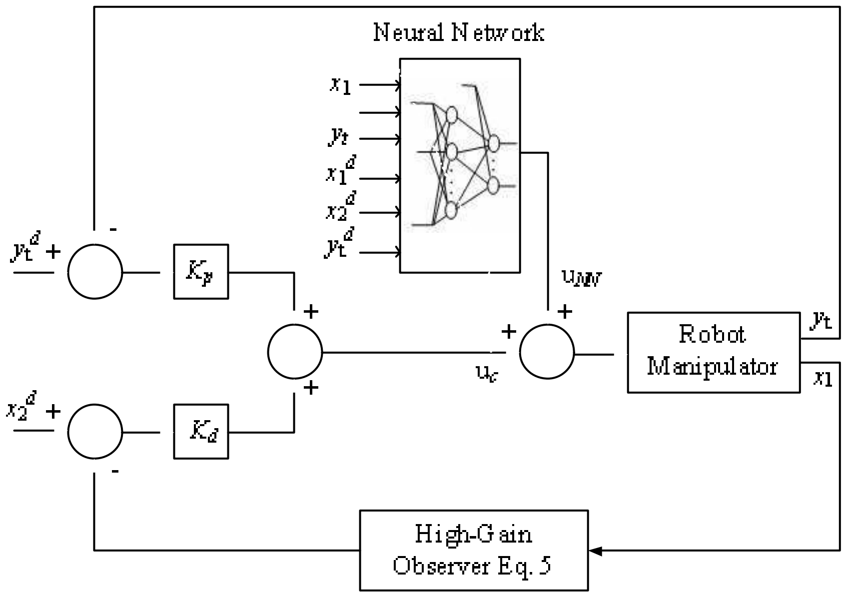


Figure 2 Control structure for flexible-link robot

Special Concerns and Techniques to Control Flexible Link

In general, the control of flexible manipulators to achieve and maintain accurate positioning can not be easily accomplished, and therefore many control techniques were presented in literatures. Tip deflection of flexible link is an important control parameter so as to achieve smooth trajectory tracking [3]. In literatures this parameter was used in different controlling schemes.
For example, this parameter was considered in [4, 5] by letting the link's tip normal elastic deflection equal to zero for calculating the torque in the frequency domain. Jnifene *et al.* [15], used strain gauges in order to measure the elastic deflection of the vibrating link. Then this deflection used as an input error to a fuzzy logic control system so as to reduce the tip vibration in the response to a step input. From the above we can conclude that the tip normal elastic deflection should be used in the feedback control.
In our work, we propose to use the tip normal elastic deflection in calculating and forming the proportional part of the PD controller, and this also logically will led to incorporate it in the

feedback error on-line learning scheme of the NN system. The derivative part of the PD controller is constructed on the error difference between the desired hub velocity and the estimated hub velocity. The estimated hub velocity as explained before is calculated by the high-gain observer (equation 5). The latter will maintain the stability of the proposed control system because it depends on the response of the hub at which the link is attached.
Considering linear displacements, the total deflection $y(x,t)$ at a distance x from the frame origin can be described as a function of both the rigid body motion $?_h(t)$ and elastic deflection $u_y(x,t)$ as[9]:

$$y(x,t) = ?_h(t) + u_y(x,t)$$

4. GROWING RADIAL BASIS FUNCTION NEURAL NETWORKS

Radial basis function with Gaussian functions have good local interpolation function and global generalization, thus they have extensively been used as the basis of NNs for nonlinear system identification and control [16]. In case of a Gaussian basis function, the output of the RBF-NN with N number of neurons is expressed as:

$$\widehat{y}(x,?) = w_? + \sum_{i=1}^{N} w_i \exp\left[\frac{-1}{s_i^2}\|x - \mu_i\|^2\right]$$

$$? = [w_? \ldots.. w_N, \mu_1 \ldots.. \mu_N, s_1 \ldots.. s_N]$$

where w, μ, s are the weights, centers and widths, respectively and ¦ ¦ denotes the Euclidean norm.
The input vector is x and $?$ comprises the set of parameters to be tuned by a learning algorithm. Figure 3 shows the structure of this neural network.

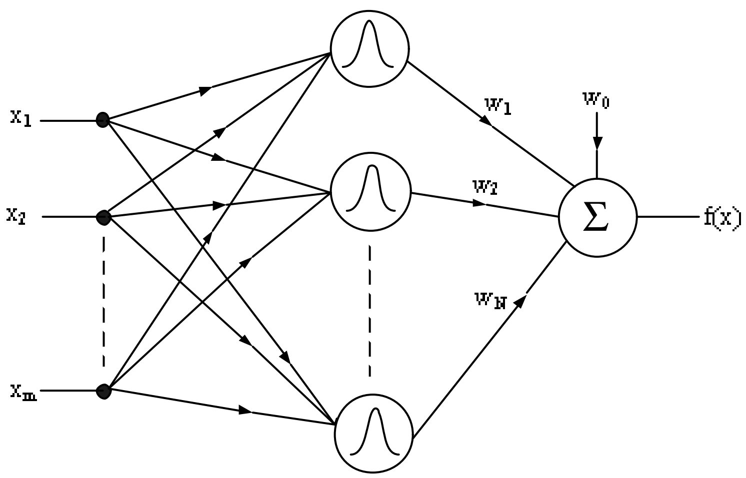


Figure 3 Radial basis function neural network

The main problem of RBF is that the total number of neurons tends to grow dramatically with the input dimension. This becomes particularly important when large dimension RBF-NNs are used in real-time problems. In order to avoid the dimensionality problems generated by standard RBF, Platt[17] proposed a sequential learning technique for RBF-NNs, where the emphasis was to learn quickly, generalize well and have a compact representation. The resulting architecture was called the Resource Allocating Network (RAN) and has proven to be suitable for online modeling of non-stationary processes. The RAN learning algorithm proceeds as follows:

- Current estimation error criteria, error must be bigger than a threshold:

$$e(k) = y(k) - \hat{y}(k) \geq E_1$$

where $y(k)$ is the function to be approximated by neural network at time k.

- Novelty criteria, the nearest center distance must be bigger than a threshold:

$$\inf_{i=1}^{N} \left\| x(k) - \mu_i(k) \right\| \geq E_2$$

- Windowed mean error criteria, windowed mean error must be bigger than a threshold:

$$\frac{1}{T} \sum_{i=0}^{T} [\, y(k-T+i) - \hat{y}(k-T+i)] \geq E_3$$

where T is the time of past data.

When all the above three criteria are satisfied, a new neuron (N+1) is added to the network; this new neuron is initialized with the following center, width and weight, respectively:

- $\mu_{N+1}(k) = x(k)$
- $s_{N+1} = ?\inf_{j=1}^{N} \left\| x(k) - \mu_j(k) \right\|$
- $w_{N+1}(k) = e(k) = y(k) - \hat{y}(k)$

Where ? is a constant called "overlapping factor".

When one (or more) of the criteria is not satisfied, the vector *?*(k) containing the tuning parameters of the RBF-NN is updated using the following relationship:

$$\theta(k+1) = \theta(k) - \eta \left. \frac{\partial \hat{y}(k)}{\partial \theta(k)} \right|_{(k)} \cdot e(k)$$

where $e(k)$ is the prediction error and *?* is the learning rate and *?*(k) is the vector of parameters to be updated.

A further improvement to the above algorithm was proposed by MRAN[18]. The growing and pruning mechanisms remains unchanged, while the parameters are updated following a 'winner takes it all' strategy. In practice only the parameters of the most activated neuron are updated, while all the other are unchanged. This strategy implies a significant reduction of the number of parameters to be updated online, and for this reason it is particularly suitable for online applications.

5. SIMULATION ANALYSIS

In order to illustrate the performance of the proposed control system we describe in this section some results. A robot manipulator is simulated consists of two flexible links and two revolute joints driven directly by servo motors. Each link in this simulation is modeled with 5 finite elements. The links are made out of aluminum and have the following characteristics [4]:

First link: L=0.66 m, A= 1.2097 x 10^{-4} m^2, I=2.2864x10^{-10} m^4, M_t =1.049 kg, I_h=0.0011823 kg.m^2

Second link: L=0.66 m, A= 0.5842 x 10^{-4} m^2, I=2.5753x10^{-11} m^4, M_t =0.0248 kg, I_h=0.00048 kg.m^2

While the material properties are the following: E=7.11 *10^{10} N/m^2 , mass density ?=2715 kg/m^3 , shear coefficient ?=5/6 and a damping ratio 0.002. The cross-section of the link is such that the arm is rigid in the vertical direction and flexible in the horizontal direction. The following PD controller gains are chosen to stabilize the system are K_p=[31 0; 0 45] and K_d=[60 0; 0 80]. The following observer design parameters are adopted [19]: L_1=[42.71 0; 0 42.71], L_2=[84.3242 0; 0 84.3242], and e=0.003. The NN parameters are shown in Table(1).
Figure 4 shows a comparison of the total tip deflection, resulting from our proposed control scheme (dotted curve) and the rigid body torque (dashed curve). The rigid body torque is calculated on the base of controlling the hub position and velocity only. While our proposed control scheme provides an excellent tracking of the tip trajectory, the rigid torque induces a large oscillation in the tip motion.

Another test for our proposed control structure is the response to sinusoidal desired trajectory. Figs. 5-7 show the hub angle ($?_h$) position, tip total deflection (y_t), and tip normal elastic deflection (u_{ty}) for both links, respectively. Vibration exists at the beginning of motion because the system is not trained yet. High vibration is recognized on the first link because of the actions from the first motor and the reaction from the second. Also high tracking error exists at the beginning of motion for link 2. Then after 3seconds excellent tracking is achieved.

Table 1: The Neural Network Updating Parameters

Parameter	Value	Description
No. of inputs :N_i	12	Friction Nonlinearity
No. of outputs: N_o	2	Approximated output
Max. no. neurons: N_{max}	5	Limits for the network growth
Tuning radius	5	Limits for the updated neurons
[$?_w$, $?_s$, $?_\mu$]	[0.05, 0.01, 0.01]	Learning Rates for the weights, widths & centers
[E_1, E_2, E_3]	[0. 1, 0.2, 0.1]	Thresholds for three condition of growing criteria
?	6.4	Activation Function Overlapping Factor

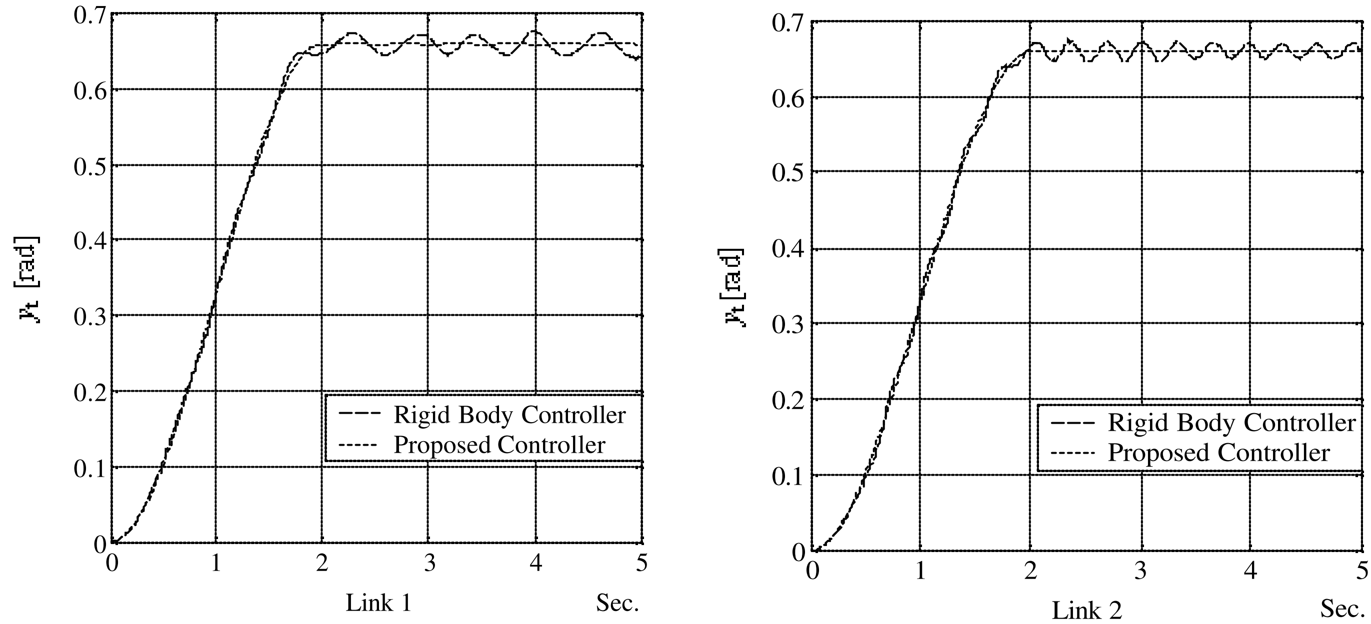


Figure 4 Comparison of the tip total deflection, our proposed control scheme (dotted curve) and the rigid body torque (dashed curve)

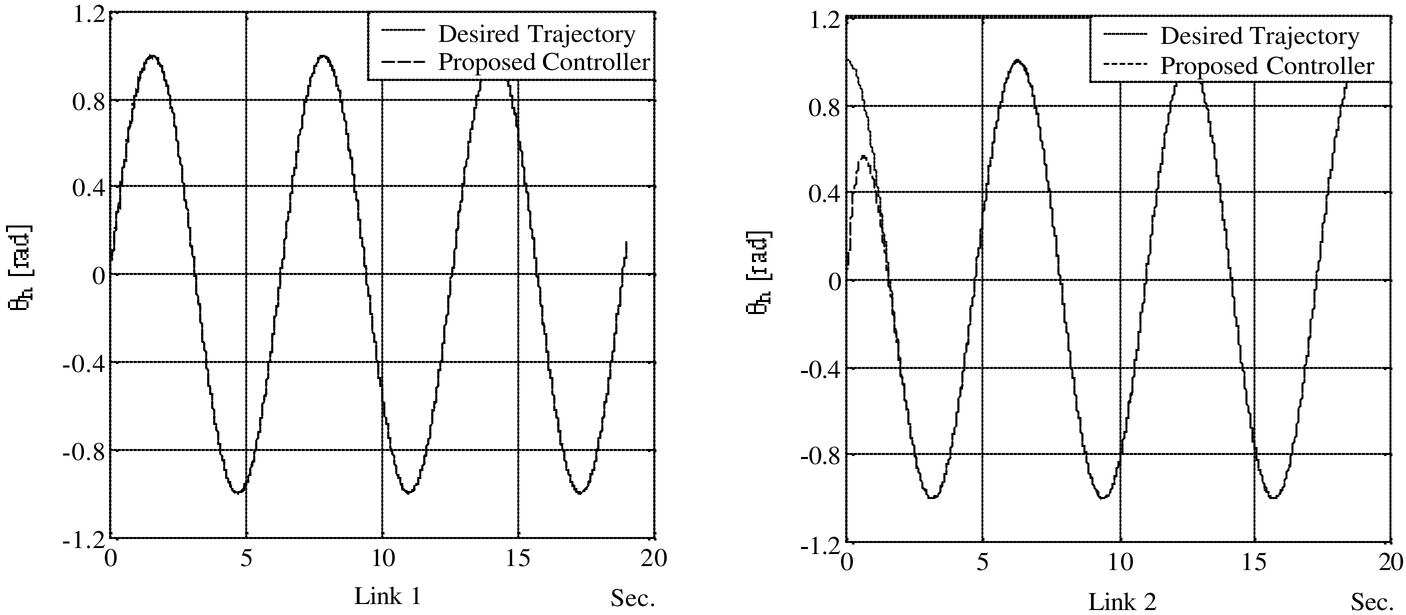


Figure 5 Hub angular position of two-link manipulator for sinusoidal input

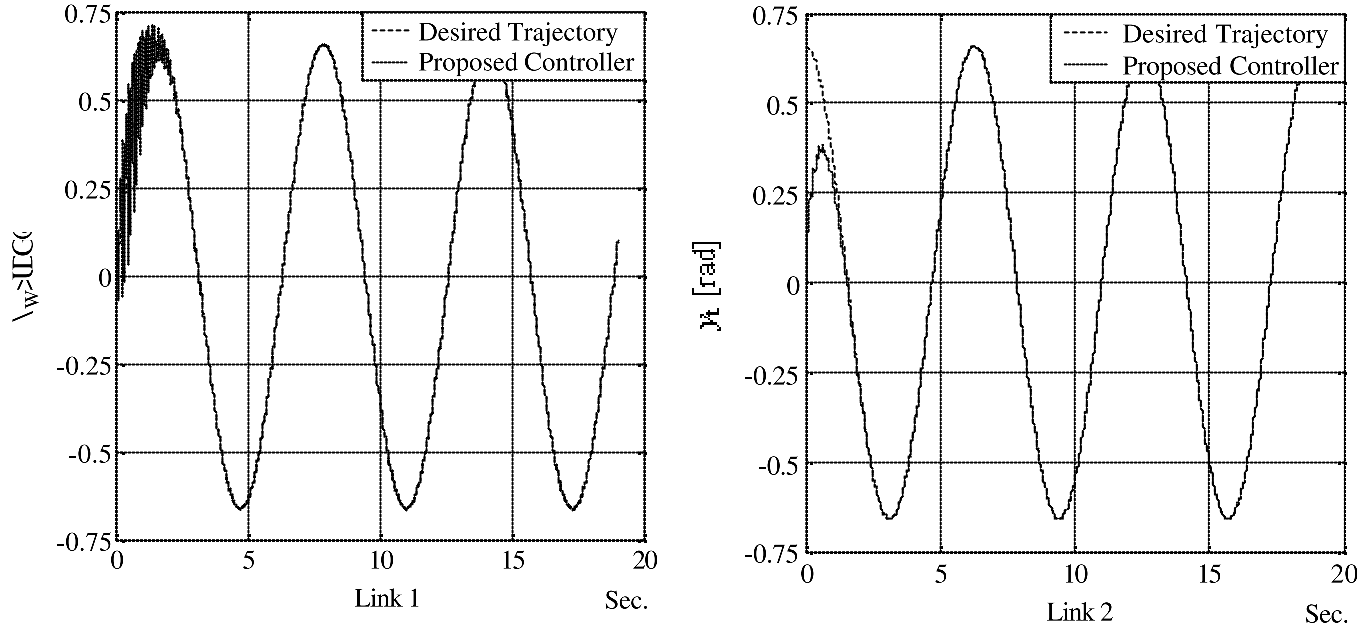


Figure 6 Tip total deflection of two-link manipulator for sinusoidal input

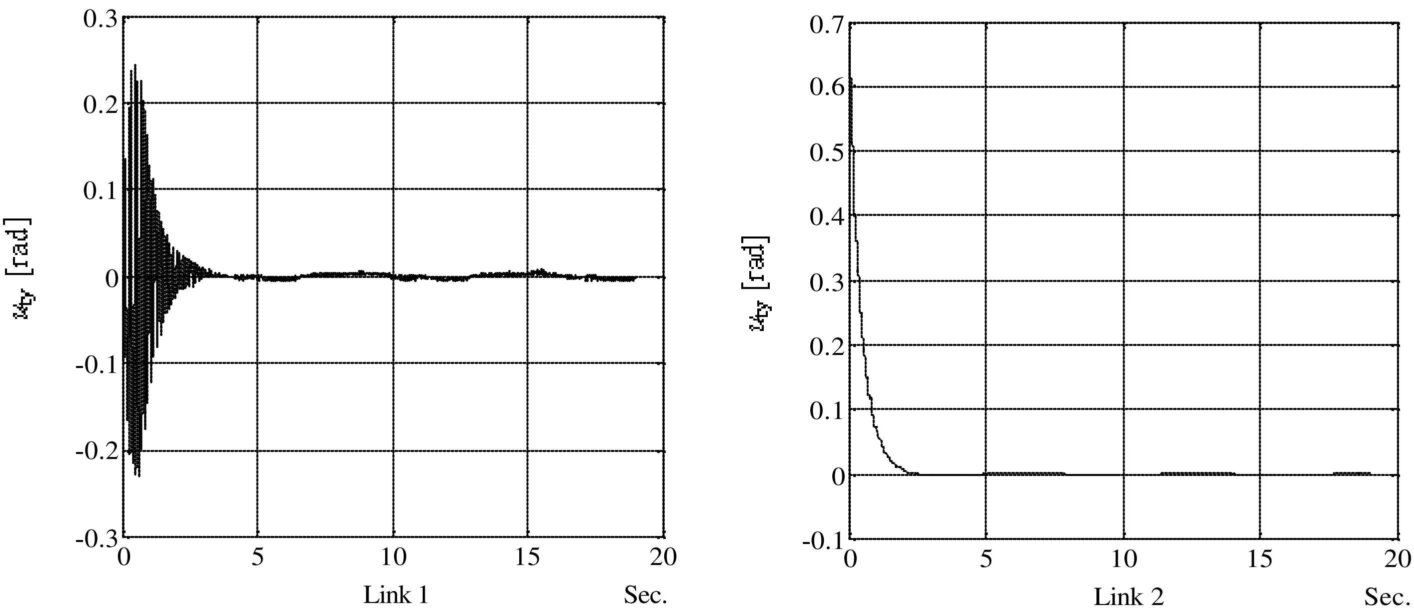


Figure 7 Tip normal elastic deflection of two-link manipulator for sinusoidal input

6. CONCLUSIONS

The Radial Basis Function neural network with the learning algorithm is a good universal approximators for any non-smooth nonlinear function. A PD controller alone can not cancel the effect of nonlinearity to achieve accurate tracking behavior for the system. The Radial Basis Function NN has a dynamic structure due to the pruning strategy, thus it has convergence to the desired target, which makes it suitable for real-time on-line learning.

Simulated case-studies for flexible arm had shown the effectiveness of the suggested control scheme. The NN with feedback error on-line learning scheme improves the system performance in terms of tracking accuracy. Using the link's tip normal deflection and the hub velocity as inputs to the neural network controller had resulted in a significant reduction in the end-point vibration of the flexible multi-link robot manipulator. Combining the PD controller with the NN was found to improve the trajectory tracking and controlling the end-point vibration of the flexible manipulator during the tracking process and after reaching the desired position.
A very important feature of the proposed control scheme is that not only the tip trajectory is tracked but also that the vibration is minimized; so that the actual motion of the whole system resembles that of a rigid system.

REFERENCES

1. J.-C Piedboeuf, M. Farooq, M.M. Bayoumi, G. Labinaz, and M.B. Argoun "Modelling and Control of Flexible Manipulators – Revisited" , Proc. 36th Midwest Symp. on Circuits and systems, Detroit, 1993, pp. 1480–1483
2. ? . Lukjanov „Kontrolle der Bewegungsbahn von flexiblen Manipulatoren mittels der Schwingungsdämpfung" // Mechatronik, Automatisierung, Steuerung. 2004. Nr. 11, S. 23
3. J. Sankin, V. Barachov „Mathematische Modellierung und Steuerung von Manipulatoren als Stabsysteme der variablen Konfiguration"// Nachrichten des Maschinenbaus. Uljanovsk: USTU, 2007. Nr. 3. S. 28-33
4. E. Bayo " Computed Torque for The Position control of Open-Chain Flexible Robots", IEEE Journal of Robotic Systems, 1988, pp. 316-321
5. M. A. Serna, E. Bayo "Trajectory Planning for Flexible Manipulators", IEEE Journal of Robotic Systems, 1990, pp. 910-915
6. W. Mark McGinley and Ji Y. Shen "Dynamic Analysis of a Two Member Manipulator Arm", NASA –CR-204996, Final Report, July 1997.
7. R.G.K.M. Aarts, J. B. Jonker and R. R. Waiboer "Modelling and Efficient Dynamic Simulation of Flexible Link Manipulators", Proc. Of WESIC 2001, University of Twente, The Netherlands, 27-29 June 2001, pp. 413-422
8. A. Green and J. Z. Sasiadek "Dynamic and Trajectory Tracking Control of a Two-Link Robot Maniplator", Journal of vibration and Control, 10: 1415-1440, 2004 .
9. J. M. Martins, Z. Mohamed, M.O. Tokhi, J. Sa'da Costa and M. A. Botto "Approaches for Dynamic Modelling of Flexible Manipulator Systems", IEE Proc.-Control Theory Appl., Vol. 150, No. 4, July 2003, pp. 401-411
10. J. García de Jalón and E. Bayo,: "Kinematic and Dynamic Simulation of Multibody Systems. The Real-Time Challenge", ISBN 0-387-94096-0, pp. 440 , Springer-Verlag, New-York, 1994
11. W. Yu and X. Li, "PD Control of Robot with Velocity Estimation and Uncertainties Compensation", IEEE Conference on Decision and Control, Florida USA, December(2001). pp. 1162-1167
12. W. Yu and A. Heredia, "PD Control of Robot with RBF Networks Compensation", IEEE International Conference on Decision and Control, Mexico, December(2000). pp. 329-334
13. H. Chen, K. Hirasawa and j. Hu, " Robust Feedback Error Learning Method for Controller Design of Nonlinear Systems", IEEE International Conference on Neural Networks, July (2004).
14. A. Bulgakov, Saad Zaghlul S. Al-Khayyt "Reibungskompensation in Kopplungen vom Manipulator mittels der Regelung auf Basis von adaptiven Neuro-Netzen" // Elektromechanik, 2008.

15. A. Jnifene and W. R. Andrews "Experimental Study on Active Vibration Control of a Single-Link Flexible Manipulator Using Tools of Fuzzy Logic and Neural Network" IEEE Tran. On Ins. & Measu., Vol. 54, No.3 , June 2005, pp. 1200-1208
16. N. Glebov, T. Kruglova „Neuro-Netzmethode der Voraussage des technischen Zustandes eines Berbaukomplexes"//Nachrichten der Hochschule. Nord-Kaujasische Region-Sonderausgabe «Probleme der Mechatronik 2006», 2007, S. .21-25.
17. J. C. Platt, "A Resource Allocation Network for Function Interpolation", Neural Computation 3(2), 1991, pp. 213-225
18. Y. Lu, N. Sundararajan, P. Saratchandran, "Analysis of Minimal Radial Basis Function Network Algorithm for Real-time identification of nonlinear dynamic systems", IEE Proceedings on Control Theory and Application 2000; Vol. 4, No. 147, pp. 476.
19. A. Heredia and Y. Wen, "A High-gain Observer Based PD Control for Robot Manipulator", American Control Conference, Chicago USA, (2000) , pp. 2518-2522

Autorenangabe:
Professor Alexey Bulgakov
Dipl.-Ing. Saad Zaghlul S. Al-Khayyt
132 Proswestschenija st.
346428 Nowotcherkassk / Russia
Tel: 007-86352 2 4568
Fax: 007-86352 2 4568
e-mail: a.bulgakow@gmx.de