We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists



122,000





Our authors are among the

TOP 1%





WEB OF SCIENCE

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

### Interested in publishing with us? Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected. For more information visit www.intechopen.com



### Control of Robotic Systems with Flexible Components using Hermite Polynomial-Based Neural Networks

Gerasimos G. Rigatos Unit of Industrial Automation Industrial Systems Institute 26504, Rion Patras Greece

#### 1. Introduction

Flexible-link robots comprise an important class of systems that include lightweight arms for assembly, civil infrastructure, bridge/vehicle systems, military applications and largescale space structures. Modelling and vibration control of flexible systems have received a great deal of attention in recent years (Kanoh, Tzafestas, et. al., 1986), (Rigatos, 2009), (Rigatos, 2006), (Aoustin, Fliess, et al., 1997). Conventional approaches to design a control system for a flexible-link robot often involve the development of a mathematical model describing the robot dynamics, and the application of analytical techniques to this model to derive an appropriate control law (Cetinkunt & Yu, 1991), (De Luca & Siciliano, 1993), (Arteaga & Siciliano, 2000). Usually, such a mathematical model consists of nonlinear partial differential equations, most of which are obtained using some approximation or simplification (Kanoh, Tzafestas, et al., 1986), (Rigatos, 2009). The inverse dynamics modelbased control for flexible link robots is based on modal analysis, i.e. on the assumption that the deformation of the flexible link can be written as a finite series expansion containing the elementary vibration modes (Wang & Gao, 2004). However, this inverse-dynamics modelbased control may result into unsatisfactory performance when an accurate model is unavailable, due to parameters uncertainty or truncation of high order vibration modes (Lewis, Jagannathan & Yesildirek, 1999).

In parallel to model-based control for flexible-link robots, model-free control methods have been studied (Rigatos, 2009), (Benosman & LeVey 2004). A number of research papers employ model-free approaches for the control of flexible-link robots based on fuzzy logic and neural networks. In (Tian & Collins, 2005) control of a flexible manipulator with the use of a neuro-fuzzy method is described, where the weighting factor of the fuzzy logic controller is adjusted by a dynamic recurrent identification network. The controller works without any prior knowledge about the manipulator's dynamics. Control of the endeffector's position of a flexible-link manipulator with the use of neural and fuzzy controllers has been presented in (Wai & Lee, 2004), (Subudhi & Morris, 2009), (Talebi, Khorasani, et. al, 1998), (Lin & Lewis, 2002), (Guterrez, Lewis & Lowe, 1998). In (Wai & Lee, 2004) an

intelligent optimal control for a nonlinear flexible robot arm driven by a permanent-magnet synchronous servo motor has been designed using a fuzzy neural network control approach. This consists of an optimal controller which minimizes a quadratic performance index and a fuzzy neural-network controller that learns the uncertain dynamics of the flexible manipulator. In (Talebi, Khorasani, et. al, 1998) a fuzzy controller has been developed for a three-link robot with two rigid links and one flexible fore-arm. This controller design is based on fuzzy Lyapunov synthesis where a Lyapunov candidate function has been chosen to derive the fuzzy rules. In (Subudhi & Morris, 2003) a neurofuzzy scheme has been proposed for position control of the end effector of a single-link flexible robot manipulator. The scale factors of the neuro-fuzzy controller are adapted online using a neural network which is trained with an improved back-propagation algorithm. In (Caswara & Ubenhauen, 2002) two different neuro-fuzzy feed-forward controllers have been proposed to compensate for the nonlinearities of a flexible manipulator. In (Renno, 2007) the dynamics of a flexible link has been modeled using modal analysis and then an inverse dynamics fuzzy controller has been employed to obtain tracking and deflection control. In (Shi & Trabia, 2006) a fuzzy logic controller has been applied to a flexible-link manipulator. In this distributed fuzzy logic controller the two velocity variables which have higher importance have been grouped together as the inputs to a velocity fuzzy controller while the two displacement variables which have lower importance degrees have been used as inputs to a displacement fuzzy logic controller. In (Hui, Fuchun & Zenghi, 2002) adaptive control for a flexible-link manipulator has been achieved using a neuro-fuzzy time-delay controller. In (Nguyen & Morris, 2007) a genetic algorithm has been used to improve the performance of a fuzzy controller designed to compensate for the links' flexibility and the joints' flexibility of a robotic manipulator.

In this paper, a neural controller using orthogonal wavelet basis functions is first proposed for the control of the flexible-link robot. The neural controller operates in parallel to a PD controller the gains of which are calculated assuming rigid link dynamics. Neural networks with wavelet basis functions, also known as 'wavelet networks', were first introduced in (Zhang & Benveniste, 1993) aiming at giving to feed-forward neural networks multiresolution analysis features and at providing neural models with good approximation features while using a small number of tunable parameters. Wavelet neural networks can be classified into orthogonal and non-orthogonal. In orthogonal wavelet networks an orthonormal basis is generated, using the wavelet function. However, in order to create the orthonormal basis the wavelet function has to satisfy restrictions. The training of the orthonormal wavelet network is fast and its expansion is easy. On the other hand, the nonorthogonal wavelet network uses the so-called wavelet frame. The family of the wavelet functions that constitute a frame are such that the energy of the resulting wavelet coefficients lies within a certain bounded range of the energy of the original signal (Addison, 2002). Controllers based on Haar orthogonal wavelets have been used in vibration control problems (Karimi & Lohmann, 2006).

Next, a neural network with Gauss-Hermite polynomial basis functions is considered for the control of flexible-link manipulators. This neural model employs Gauss-Hermite basis functions which are localized both in space and frequency, as which, as wavelet basis functions, allow for better approximation of the multi-frequency characteristics of vibrating structures (Cannon & Slotine, 1995), (Krzyzak & Sasiadek, 1991), (Lin, 2006), (Sureshbabu & Farell, 1999). Gauss-Hermite basis functions have also some interesting properties

460

(Refregier, 2003), (Rigatos & Tzafestas, 2006): (i) they remain almost unchanged by the Fourier transform, which means that the weights of the associated neural network demonstrate the energy which is distributed to the various eigenmodes of the vibrating structure. This in turn enables to define thresholds for truncating the basis functions expansion and to design a neural controller with a small number of adaptable parameters, (ii) unlike wavelet basis functions the Gauss-Hermite basis functions have a clear physical meaning since they represent the solutions of differential equations describing stochastic oscillators and each neuron can be regarded as the frequency filter of the respective vibration eigenfrequency.

The structure of the chapter is as follows: In Section 2 the dynamic model of flexible-link robots is analyzed. In Section 3 a neural control scheme for flexible link robots is introduced. In Section 4 wavelet basis functions are proposed to implement the neural controller for the flexible-link manipulator. In Section 5 Hermite-polynomial basis functions are used to implement the neural controller which stabilizes the flexible-link robot dynamics. In Section 6 simulation experiments are presented. Finally in Section 7 concluding remarks are stated.

#### 2. Dynamic model of flexible-link robots

A common approach in modelling of flexible-link robots is based on the Euler-Bernoulli model (Talebi, Khorasani, et. al, 1998), (Wang & Gao, 2004). This model consists of nonlinear partial differential equations, which are obtained using some approximation or simplification. In case of a single-link flexible manipulator the basic variables of this model are w(x, t) which is the deformation of the flexible link, and  $\theta(t)$  which is the joint's angle.

$$E \cdot I \cdot w^{\text{im}}(x,t) + \rho w(x,t) + \rho x \theta(t) = 0$$
(1)

$$I_t \stackrel{\iota}{\theta(t)} + \rho \int_0^L x w(x,t) dx = T(t)$$
(2)

In Eq. (1) and (2)),  $w^{\text{init}}(x,t) = \frac{\partial^4 w(x,t)}{\partial x^4}$ ,  $w(x,t) = \frac{\partial^2 w(x,t)}{\partial t^2}$ , while  $I_t$  is the moment of inertia of a rigid link of length L,  $\rho$  denotes the uniform mass density and  $E \cdot I$  is the uniform flexural rigidity with units  $N \cdot m^2$ . To calculate w(x,t), instead of solving analytically the above partial differential equations, modal analysis can be used which assumes that w(x,t) can be approximated by a weighted sum of orthogonal basis functions

$$w(x,t) = \sum_{i=1}^{n_e} \phi_i(x) v_i(t)$$
(3)

where index  $i = [1, 2, ..., n_e]$  denotes the normal modes of vibration of the flexible link. Using modal analysis a dynamical model of finite-dimensions is derived for the flexible link robot. Without loss of generality assume a 2-link flexible robot (Fig. 1) and that only the first two

vibration modes of each link are significant ( $n_e = 2$ ).  $\Sigma_1$  is a point on the first link with reference to which the deformation vector is measured. Similarly,  $\Sigma_2$  is a point on the second link with reference to which the associated deformation vector is measured. In that case the dynamic model of the robot becomes (Wang & Gao, 2004), (Lewis, Jagannathan & Yelsidirek, 1999):

$$\begin{pmatrix} M_{11}(z) & M_{12}(z) \\ M_{21}(z) & M_{22}(z) \end{pmatrix} \cdot \begin{pmatrix} \vdots \\ \theta \\ v \end{pmatrix} + \begin{pmatrix} F_1(z,z) \\ F_2(z,z) \end{pmatrix} + \begin{pmatrix} 0_{2\times 2} & 0_{2\times 4} \\ 0_{4\times 2} & D(z) \end{pmatrix} \cdot \begin{pmatrix} \vdots \\ \theta \\ v \end{pmatrix} + \begin{pmatrix} 0_{2\times 2} & 0_{2\times 4} \\ 0_{4\times 2} & K(z) \end{pmatrix} \cdot \begin{pmatrix} \theta \\ v \end{pmatrix} = \begin{pmatrix} T(t) \\ 0_{4\times 1} \end{pmatrix}$$
(4)

where  $z = \begin{bmatrix} \theta & v \end{bmatrix}^T$ , with  $\theta = \begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix}^T$ ,  $v = \begin{bmatrix} v_{11} & v_{12} & v_{21} & v_{22} \end{bmatrix}^T$  (vector of the vibration modes for links 1 and 2), and  $\begin{bmatrix} F_1(z, z) & F_2(z, z) \end{bmatrix}^T$  (centrifugal and Coriolis forces). The elements of the inertia matrix are:  $M_{11} \in R^{2 \times 2}$ ,  $M_{12} \in R^{2 \times 4}$ ,  $M_{21} \in R^{4 \times 2}$ ,  $M_{22} \in R^{4 \times 2}$ . The damping and elasticity matrices of the aforementioned model are  $D \in R^{4 \times 4}$  and  $K \in R^{4 \times 4}$ . Moreover the vector of the control torques is  $T(t) = [T_1(t) & T_2(t)]^T$ .



Fig. 1. A 2-DOF flexible-link robot

#### 3. Neural control for flexible-link robots

#### 3.1 Neural network-based control of flexible manipulators

Adaptive neural network control of robotic manipulators has been extensively studied (Lewis, Jagannathan & Yelsidirek, 1999), (Ge, Lee & Harris, 1998). Following (Tian, Wang & Mao, 2004) a method of neural adaptive control for flexible-link robots will be proposed.

Eq. (4) represents the dynamics of the flexible-link manipulator. It actually refers to a nonlinear transformation (mapping) from inputs (torques T(t) generated by the motors) to outputs (motion of the joints). This nonlinear model can be written in the general form:

$$\hat{\theta} = G(\theta, \theta, v, v, T(t))$$
(5)

Consequently, the inverse dynamics of the flexible-link manipulator, is a relation that provides the torque that should be generated by the motors of the joints so as the joints angle, angular velocity and acceleration to take certain values. The inverse model of Eq. (5) is given by

$$T(t) = G^{-1}(\theta, \theta, \theta, v, v)$$
(6)

The dynamic model and its inverse are time dependent. If the inverse dynamic model of Eq. (6) can be explicitly calculated then a suitable control law for the flexible-link robot is available.

However, this model is not usually available and the system dynamics has to be adaptively identified. A neural network model can be used to effectively approximate the inverse dynamical model of Eq. (6). Variables  $\theta$ ,  $\dot{\theta}$ ,  $\ddot{\theta}$  can be measured while variables v,  $\dot{v}$  are non-measurable. Thus, the inverse dynamics of the manipulator can be decomposed into *n* sub-models given in the following form:

$$T(t) = G^{-1}(\theta, \dot{\theta}, \dot{\theta}) = \begin{pmatrix} g_1^{-1}(\theta, \dot{\theta}, \dot{\theta}) \\ g_2^{-1}(\theta, \dot{\theta}, \dot{\theta}) \\ \vdots \\ g_n^{-1}(\theta, \dot{\theta}, \dot{\theta}) \end{pmatrix}$$
(7)

where each  $g_i^{-1}$ , i = 1, 2, ..., n defines the inverse dynamics of the corresponding joint, while n is the number of joints of the manipulator.

A neural network can be employed to approximate each sub-model  $g_i^{-1}$  of the flexible robot's inverse dynamics. Therefore, the inverse dynamics of the overall system can be represented by a neural network  $N(\theta, \theta, \theta, w)$  (Tian, Wang & Mao, 2004)

(11)

$$T(t) \cong N(\theta, \dot{\theta}, \ddot{\theta}, w) = \begin{pmatrix} N_1(\theta, \dot{\theta}, \ddot{\theta}, w_1) \\ N_2(\theta, \dot{\theta}, \ddot{\theta}, w_2) \\ \dots \\ N_n(\theta, \dot{\theta}, \ddot{\theta}, w_n) \end{pmatrix}$$
(8)

where  $N_i(\theta, \theta, \theta, w_i)$ , i = 1, 2, ..., n is the *i*-th neural network that approximates the *i*-th sub-model of robot's inverse dynamics and  $w_i$  is the associated weights vector.

Using a NN to model the dynamics of the flexible-link robot provides a mapping of the joint angles vector to the motor torques vector. The torque appearing at the output of the neural network can be combined with a PD-feedback controller to generate the overall control signal that will finally drive the motors (Tzafestas & Rigatos, 1998). Therefore, the control scheme can be given as

$$T(t) = N(\theta, \theta, \theta, w) + K_p e + K_d e$$
<sup>(9)</sup>

where  $N(\theta, \theta, \theta, w)$  is the neural network approximation of the actual inverse dynamics of the manipulators, while  $K_p \in \mathbb{R}^{n \times m}$  and  $K_d \in \mathbb{R}^{n \times m}$  are the diagonal gain matrices with entries  $K_p$  and  $K_d$ , respectively, denoting a servo feedback that is introduced to stabilize

the system. In Eq. (9)  $e = \theta - \theta_d$ , and  $e = \theta - \theta_d$  denote the position and velocity error of the robot's joints, respectively, and  $\theta_d \in R^2$  is the vector of the position and velocity setpoints.

The architecture of NN-based control of flexible-link robots is depicted in Fig. 2. Making use of Eq. (7) and Eq. (8), the neural network controller that is described by Eq. (9) becomes

$$N(\theta, \theta, \theta, w) + K_p e + K_d e = G^{-1}(\theta, \theta, \theta)$$
(10)  
or equivalently,

$$K_{p}e + K_{d}e = G^{-1}(\theta, \theta, \theta) - N(\theta, \theta, \theta, w) = \overline{N}(\theta, \theta, \theta, w)$$

Eq. (11) represents a decoupled linear system, driven by the nonlinear vector function  $\overline{N}(\theta, \theta, \theta, \theta, w) \in \mathbb{R}^n$ . This function represents the error between the actual inverse dynamics  $G^{-1}(\theta, \theta, \theta, \theta)$  and its estimated model  $N(\theta, \theta, \theta, w)$  and can be written as

Control of Robotic Systems with Flexible Components using Hermite Polynomial-Based Neural Networks

$$\bar{N}(\theta, \dot{\theta}, \ddot{\theta}, w) = \begin{pmatrix} \bar{N}_{1}(\theta, \dot{\theta}, \ddot{\theta}, w) \\ ... \\ \bar{N}_{2}(\theta, \dot{\theta}, \ddot{\theta}, w) \end{pmatrix} = \begin{pmatrix} g_{1}^{-1}(\theta, \dot{\theta}, \ddot{\theta}) - N_{1}(\theta, \dot{\theta}, \ddot{\theta}, w) \\ ... \\ g_{n}^{-1}(\theta, \dot{\theta}, \ddot{\theta}) - N_{n}(\theta, \dot{\theta}, \ddot{\theta}, w) \end{pmatrix}$$
(12)

Fig. 2. NN-based control of the flexible-link robot

Instead of using one neural network to approximate the inverse dynamics of the flexiblelink robot one can use a separate network for each joint of the manipulator. In that case and using Eq. (12), the error equation for the i -th joint of the manipulator becomes

$$K_p^i e_i + K_d^i e_i = \overline{N}_i (\theta, \theta, \theta, w), \ i = 1, 2, \dots, n$$

$$(13)$$

where  $e_i$  and  $e_i$  denote the position and velocity errors of the *i*-th joint respectively,  $K_p^i$ and  $K_d^i$  are the proportional and derivative gains of the PD controller of the *i*-th joint, and  $\bar{N}_i(\theta, \dot{\theta}, \dot{\theta}, w)$  is the approximation error of the NN assigned to the *i*-th joint. The objective is to eliminate the approximation error, i.e. to succeed  $\lim_{t\to\infty} \bar{N}_i(\theta, \dot{\theta}, \ddot{\theta}, w) = 0$ . In that case, a suitable selection of the control gains  $K_p^i$  and  $K_d^i$  results into

$$K_p^i e_i + K_d^i \dot{e}_i = 0 \Longrightarrow \lim_{t \to \infty} e_i(t) = 0, \quad \lim_{t \to \infty} \dot{e}_i(t) = 0$$
(14)

Eq. (14) denotes that, the convergence condition for the closed-loop system is to make the error surface  $\varepsilon_i = K_p^i e_i + K_d^i \dot{e}_i$  approach zero. Then, a suitable selection of the control gains will result in the asymptotic convergence of e(t) to 0

$$K_{p}^{i}e_{i} + K_{d}^{i}\dot{e}_{i}(t) = 0 \Longrightarrow \dot{e}_{i}(t) = -\frac{K_{p}^{i}}{K_{d}^{i}}e_{i}(t) \Longrightarrow e_{i}(t) = e_{i}(0)e^{-\frac{K_{p}^{i}}{K_{d}^{i}}t}$$
(15)

Thus a measure of the output error can be considered to be

$$\varepsilon_i = \bar{N}_i(\theta, \theta, \theta, w) \tag{16}$$

which reflects the discrepancy between the actual inverse dynamics of the manipulator and its neural network approximation. To this end, the following cost function is defined for each joint

$$E_i(t) = \frac{1}{2}\varepsilon_i^2 \tag{17}$$

This cost function gives the squared distance of the error function  $\varepsilon_i = K_p^i e_i + K_d^i e_i$  from 0. The weights update algorithm is derived from the minimization of the cost function  $E_i(t)$  over the weight space of the corresponding NN model.

#### 3.2 Feed-forward neural networks for flexible-link robot control

The NN-based control for the flexible-link robots, which is depicted in Fig. 2, can be substantiated with the use of feed-forward neural networks (Rigatos, 2009). Feed-forward neural networks (FNN) serve as powerful computational tools, in a diversity of applications including function approximation, classification and pattern recognition. When equipped with procedures for learning from measurement data they can generate models of unknown systems. Feed-forward neural networks are the most popular neural architectures due to their structural flexibility, good representational capabilities, and availability of a large number of training algorithms.

The idea of function approximation with the use of feed-forward neural networks (FNN) comes from generalized Fourier series. It is known that any function  $\psi(x)$  in a  $L^2$  space can be expanded in a generalized Fourier series in a given orthonormal basis, i.e.

$$\psi(x) = \sum_{k=1}^{\infty} c_k \psi_k(x), \quad a \le x \le b$$
(18)

Truncation of the series yields in the sum

$$S_M(x) = \sum_{k=1}^M a_k \psi_k(x)$$
(19)

If the coefficients  $a_k$  are taken to be equal to the generalized Fourier coefficients, i.e. when

 $a_k = c_k = \int_a^b \psi(x)\psi_k(x)dx$ , then Eq. (19) is a mean square optimal approximation of  $\psi(x)$ . Unlike generalized Fourier series, in FNN the basis functions are not necessarily orthogonal.

The hidden units in a FNN usually have the same activation functions and are often selected as sigmoidal functions or gaussians. A typical feed-forward neural network consists of n inputs  $x_i$ , i = 1, 2, ..., n, a hidden layer of m neurons with activation function  $h : R \to R$  and a single output unit (see Fig. 3(a)). The FNN's output is given by





Fig. 3. (b) Neural network with Hermite basis functions

The root mean square error in the approximation of function  $\psi(x)$  by the FNN is given by

$$E_{RMS} = \sqrt{\frac{1}{N} \sum_{k=1}^{N} (\psi(x^k) - \psi(x^k))^2}$$
(21)

where  $x^k = [x_1^k, x_2^k, ..., x_n^k]$  is the *k*-th input vector of the neural network. The activation function is usually a sigmoidal function  $h(x) = \frac{1}{1+e^{-x}}$  while in the case of radial basis functions networks it is a Gaussian (Haykin, 1994). Several learning algorithms for neural networks have been studied. The objective of all these algorithms is to find numerical values for the network's weights so as to minimize the mean square error  $E_{RMS}$  of Eq. (21). The algorithms are usually based on first and second order gradient techniques. These algorithms belong to: i) batch-mode learning, where to perform parameters update the outputs of a large training set are accumulated and the mean square error is calculated (back-propagation algorithm, Gauss-Newton method, Levenberg-Marquardt method, etc.), ii) pattern-mode learning, in which training examples are run in cycles and the parameters update is carried out each time a new training pattern becomes available (Extended Kalman Filter algorithm).

Unlike conventional FNN with sigmoidal or Gaussian basis functions, Hermite polynomialbased FNN remain closer to Fourier series expansions by employing activation functions which satisfy the property of orthogonality (Zuo, Zhu and Cai, 2009). Other basis functions with the property of orthogonality are Hermite, Legendre, Chebyshev, and Volterra polynomials (Refregier, 2003), (Rigatos & Tzafestas, 2006), (Yang & Cheng, 1996).

#### 4. Neural control using wavelet basis functions

#### 4.1 Wavelet frames

First, as basis functions of the feed-forward neural network orthogonal wavelet functions will be considered. The continuous time wavelet is defined at scale a and b as

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}}\psi(\frac{x-b}{a})$$
(22)

It will be shown that a continuous time signal f(x) can be expressed as a series expansion of discrete wavelet basis functions. The discrete wavelet has the form (Addison, 2002)

$$\psi_{m,n}(x) = \frac{1}{\sqrt{a_0^m}} \psi(\frac{x - nb_0 a_0^m}{a_0^m})$$
(23)

The wavelet transform of a continuous signal f(x) using discrete wavelets of the form of Eq. (23) is given by

$$T_{m,n} = \int_{-\infty}^{+\infty} f(x) \frac{1}{\sqrt{a_0^m}} \psi(\frac{x - nb_0 a_0^m}{a_0^m}) dx$$
(24)

which can be also expressed as the inner product  $T_{m,n} = \langle f, \psi_{m,n} \rangle$ . For the discrete wavelet transform, the values  $T_{m,n}$  are known as wavelet coefficients. To determine how good the representation of a signal is in the wavelet space one can use the theory of wavelet frames. The family of wavelet functions that constitute a frame are such that the energy of the resulting wavelet coefficients lies within a certain bounded range of the energy of the original signal

$$AE \le \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} |T_{m,n}|^2 \le BE$$
(25)

where  $T_{m,n}$  are the discrete wavelet coefficients, A and B are the frame bounds, and E is the energy of the signal given by  $E = \int_{-\infty}^{+\infty} |f(x)|^2 dt = ||f(x)||^2$ . The values of the frame bounds depend on the parameters  $a_0$  and  $b_0$  chosen for the analysis and the wavelet

function used. If A = B the frame is known as tight and has a simple reconstruction formula given by the infinite series

$$f(x) = \frac{1}{A} \sum_{m = -\infty}^{+\infty} \sum_{n = -\infty}^{+\infty} T_{m,n} \psi_{m,n}(x)$$
(26)

A tight frame with A = B > 1 is redundant, with A being a measure of the redundancy. When A = B = 1 the wavelet family defined by the frame forms an orthonormal basis. Even if  $A \neq B$  a reconstruction formula of f(x) can be obtained in the form:

$$f'(x) = \frac{2}{A+B} \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} T_{m,n} \psi_{m,n}(x)$$
(27)

where f'(x) is the reconstruction which differs from the original signal f(x) by an error which depends on the values of the frame bounds. The error becomes acceptably small for practical purposes when the ratio B/A is near unity. The closer this ratio is to unity, the tighter the frame.

#### 4.2 Dyadic grid scaling and orthonormal wavelet transforms

The dyadic grid is the simplest and most efficient discretization for practical purposes and is used for the construction of an orthonormal wavelet basis. Substituting  $a_0 = 2$  and  $b_0 = 1$  into Eq. (23) the dyadic grid wavelet can be written as

$$\psi_{m,n} = \frac{1}{\sqrt{2^m}} \psi(\frac{x - n2^m}{2^m})$$
(28)

or more compactly

$$\psi_{m,n}(x) = 2^{-\frac{m}{2}} \psi(2^{-m}x - n)$$
<sup>(29)</sup>

Discrete dyadic grid wavelets are commonly chosen to be orthonormal. These wavelets are both orthogonal to each other and normalized to have unit energy. This is expressed as

$$\int_{-\infty}^{+\infty} \psi_{m,n}(x)\psi_{m',n'}(x)dx = \begin{cases} 1 & \text{if } m = m' & \text{and } n = n' \\ 0 & \text{otherwise} \end{cases}$$
(30)

Thus, the products of each wavelet with all others in the same dyadic system are zero. This also means that the information stored in a wavelet coefficient  $T_{m,n}$  is not repeated elsewhere and allows for the complete regeneration of the original signal without redundancy. In addition to being orthogonal, orthonormal wavelets are normalized to have unit energy. This can be seen from Eq. (30), as using m = m' and n = n' the integral gives

the energy of the wavelet function equal to unity. Orthonormal wavelets have frame bounds A = B = 1 and the corresponding wavelet family is an orthonormal basis. An orthonormal basis has components which, in addition to being able to completely define the signal, are perpendicular to each other.

Using the dyadic grid wavelet of Eq. (28) the discrete wavelet transform is defined as

$$T_{m,n} = \int_{-\infty}^{+\infty} f(x)\psi_{m,n}(x)dx$$
(31)

By choosing an orthonormal wavelet basis  $\psi_{m,n}(x)$  one can reconstruct the original signal f(x) in terms of the wavelet coefficients  $T_{mn}$  using the inverse discrete wavelet transform:

$$f(x) = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} T_{m,n} \psi_{m,n}(x)$$
(32)

Moreover, the energy of the signal can be expressed as

$$\int_{-\infty}^{+\infty} |f(x)|^2 dx = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} |T_{m,n}|^2$$
(33)

#### 4.3 The scaling function and the multi-resolution representation

Orthonormal dyadic discrete wavelets are associated with 'scaling functions' and their dilation equations (Addison, 2002), (Mallat, 1999). The scaling function is associated with the smoothing of the signal and has the same form as the wavelet

$$\phi_{m,n}(x) = 2^{-m/2} \phi(2^{-m/2} x - n) \tag{34}$$

The scaling functions have the property

$$\int_{-\infty}^{+\infty} \phi_{0,0}(x) dx = 1$$
(35)

where  $\phi_{0,0}(x) = \phi(x)$  is sometimes referred as the mother scaling function or mother wavelet. The scaling function is orthogonal to translations of itself, but not to dilations of itself. The scaling function can be convolved with the signal to produce approximation coefficients as follows:

$$S_{m,n} = \int_{-\infty}^{+\infty} f(x)\phi_{m,n}(x)dx$$
(36)

One can represent a signal f(x) using a combined series expansion using both the approximation coefficients and the wavelet (detail) coefficients as follows:

$$f(x) = \sum_{n = -\infty}^{+\infty} S_{m_0,n} \phi_{m_0,n} + \sum_{m = -\infty}^{+\infty} \sum_{n = -\infty}^{+\infty} T_{m,n} \psi_{m,n}(x)$$
(37)

It can be seen from this equation that the original continuous signal is expressed as a combination of an approximation of itself, at arbitrary scale index  $m_0$  added to a succession of signal details form scales  $m_0$  down to negative infinity. The signal detail at scale m is defined as

$$d_{m}(x) = \sum_{n=-\infty}^{+\infty} T_{m,n} \psi_{m,n}(x)$$
(38)

and hence one can write Eq. (37)

$$f(x) = f_{m_0}(x) + \sum_{n = -\infty}^{m_0} d_m(x)$$
(39)

From this equation it can be shown that

$$f_{m-1}(x) = f_m(x) + d_m(x)$$
(40)

which shows that if one adds the signal detail at an arbitrary scale (index m) to the approximation at that scale he gets the signal approximation at an increased resolution (at a smaller scale index m-1). This is the so-called multi-resolution representation.

#### 4.4 Examples of orthonormal wavelets

Wavelet functions can be further analyzed in terms of 'scaling functions'. The scaling equation (or dilation equation) describes the scaling function  $\phi(t)$  in terms of contracted and shifted versions of itself as follows (Addison, 2002), (Mallat, 1999):

$$\phi(x) = \sum_{k} c_k \phi(2x - k) \tag{41}$$

where  $\phi(2x-k)$  is a contracted version of  $\phi(x)$  shifted along the x-axis by an integer step k and factored by an associated scaling coefficient  $c_k$ . The coefficients of the scaling equation should satisfy the condition

$$\sum_{k} c_k = 2 \tag{42}$$

$$\sum_{k} c_k c_{k+2k'} = \begin{cases} 2 & \text{if } k' = 0\\ 0 & \text{otherwise} \end{cases}$$
(43)

This also shows that the sum of the squares of the scaling coefficients is equal to 2. The same coefficients are used in reverse with alternate signs to produce the associated wavelet equation

$$\psi(x) = \sum_{k} (-1)^{k} c_{1-k} \phi(2x-k)$$
(44)

This construction ensures that the wavelets and their corresponding scaling functions are orthogonal. For wavelets of compact support, which have a finite number of scaling coefficients  $N_k$  the following wavelet function is defined

$$\psi(x) = \sum_{k} (-1)^{k} c_{N_{k} - 1 - k} \phi(2x - k)$$
(45)

This ordering of scaling coefficients used in the wavelet equation allows for our wavelets and their corresponding scaling equations to have support over the same interval  $[0, N_{k-1}]$ . Often the reconfigured coefficients used for the wavelet function are written more compactly as

$$b_k = (-1)^k c_{N_k - 1 - k} \tag{46}$$

where the sum of all coefficients  $b_k$  is zero. Using this reordering of the coefficients Eq. (45) can be written as

$$\psi(x) = \sum_{k=0}^{N_k - 1} b_k \phi(2x - k) \tag{47}$$

From the previous equations and examining the wavelet at scale index m + 1 one can see that for arbitrary integer values of m the following holds

$$2^{-\frac{m+1}{2}}\varphi(\frac{x}{2^{m+1}}-n) = 2^{-m/2}2^{-1/2}\sum_{k}c_{k}\varphi(\frac{2x}{2\times 2^{m}}-2nk)$$
(48)

which may be written more compactly as

$$\phi_{m+1,n}(x) = \frac{1}{\sqrt{2}} \sum_{k} c_k \phi_{m,2n+k}(x)$$
(49)

That is the scaling function at an arbitrary scale is composed of a sequence of shifted functions at the next smaller scale each factored by their respective scaling coefficients. Similarly, for the wavelet function one obtains

$$\psi_{m+1,n}(x) = \frac{1}{\sqrt{2}} \sum_{k} b_k \phi_{m,2n+k}(x)$$
(50)

#### 4.5 The Haar wavelet

The Haar wavelet is the simplest example of an orthonormal wavelet (see Fig. 4 and Fig. 5). Its scaling equation contains only two nonzero scaling coefficients and is given by

$$\phi(x) = \phi(2x) + \phi(2x - 1)$$
(51)

that is, its scaling coefficients are  $c_0 = c_1 = 1$ . These values can be obtained from Eq. (42) and Eq. (43). The solution of the Haar scaling equation is the single block pulse defined as

$$\phi(x) = \begin{cases} 1 & \text{if } 0 \le x < 1\\ 0 & \text{otherwise} \end{cases}$$
(52)

Using this scaling function, the Haar wavelet equation is

$$\psi(x) = \phi(2x) - \phi(2x - 1) \tag{53}$$

The Haar wavelet is finally found to be (see Fig. 4 and Fig. 5)

$$\begin{aligned}
1 & \text{if } 0 \le x < 1/2 \\
\psi(x) = \begin{cases} -1 & \text{if } 1/2 \le x < 1 \\ 0 & \text{elsewhere} \end{cases}$$
(54)

The mother wavelet for the Haar wavelet system  $\psi(x) = \psi_{0,0}(x)$  is formed from two dilated unit block pulses sitting next to each other on the time axis, with one of them inverted. From the mother wavelet one can construct the Haar system of wavelets on a dyadic grid  $\psi_{m,n}(x)$ .

Control of Robotic Systems with Flexible Components using Hermite Polynomial-Based Neural Networks



Fig. 4 (i) The Haar scaling function in terms of shifted and dilated versions of itself, (ii) The Haar wavelet in terms of shifted and dilated versions of the scaling function.



Fig. 5. (i) Three consecutive scales shown from the Haar wavelet family specified on a dyadic grid, e.g. from the bottom  $\psi_{m,n}(x)$ ,  $\psi_{m+1,n}(x)$ ,  $\psi_{m+2,n}(x)$ , (ii) Three Haar wavelets at three

consecutive scales on a dyadic grid, iii) Three Haar wavelets at different scales. In this case, the Haar wavelets are not defined on a dyadic grid and are hence not orthogonal to each other.

#### 5. Neural Networks using Hermite activation functions

#### 5.1 The Gauss-Hermite series expansion

Next, as orthogonal basis functions of the feed-forward neural network Hermite polynomials are considered. These are the spatial components  $X_k(x)$  of the solution of Schrödinger's differential equation and describe a stochastic oscillation:

$$X_k(x) = H_k(x)e^{-\frac{x^2}{2}}, \ k = 0,1,2,...$$
 (55)

where  $H_k(x)$  are the Hermite orthogonal functions (see Fig. 6(a) and Fig. 6(b)). The Hermite functions  $H_k(x)$  are the eigenstates of the quantum harmonic oscillator. The general relation for the Hermite polynomials is

$$H_k(x) = (-1)^k e^{x^2} \frac{d^{(k)}}{dx^{(k)}} e^{-x^2}$$
(56)

According to Eq. (56) the first five Hermite polynomials are:  $H_0(x) = 1$ ,  $H_1(x) = 2x$ ,  $H_2(x) = 4x^2 - 2$ ,  $H_3(x) = 8x^3 - 12x$ ,  $H_4(x) = 16x^4 - 48x^2 + 12$ .

×





12(×)

Fig. 6(a). First five one-dimensional Hermite polynomial basis functions

Fig. 6(b). Analytical representaion of the 1D-Hermite polynomial basis function

Hermite polynomials are orthogonal, i.e. it holds

Control of Robotic Systems with Flexible Components using Hermite Polynomial-Based Neural Networks

$$\int_{-\infty}^{+\infty} e^{-x^2} H_m(x) H_k(x) dx = \begin{cases} 2^k \, k! \sqrt{\pi} & \text{if } m = k \\ 0 & \text{if } m \neq k \end{cases}$$
(57)

Using now, Eq. (57), the following basis functions can be defined (Refregier, 2003):

$$\psi_k(x) = [2^k \sqrt{\pi}k!]^{-\frac{1}{2}} H_k(x) e^{-\frac{x^2}{2}}$$
 (58)

where  $H_k(x)$  is the associated Hermite polynomial. From Eq. (57), the orthogonality of basis functions of Eq. (58) can be deduced, which means

$$\int_{-\infty}^{+\infty} \psi_m(x)\psi_k(x)dx = \begin{cases} 1 & \text{if } m = k \\ 0 & \text{if } m \neq k \end{cases}$$
(59)

Moreover, to succeed multi-resolution analysis Hermite basis functions of Eq. (58) are multiplied with the scale coefficient a. Thus the following basis functions are derived

$$\int_{-\infty}^{+\infty} \beta_m(x,a)\beta_k(x,a)dx = \begin{cases} 1 & \text{if } m = k \\ 0 & \text{if } m \neq k \end{cases}$$
(60)

which also satisfy orthogonality condition

$$\int_{-\infty}^{+\infty} \beta_m(x,a)\beta_k(x,a)dx = \begin{cases} 1 & \text{if } m = k \\ 0 & \text{if } m \neq k \end{cases}$$
(61)

Any function f(x)  $x \in R$  can be written as a weighted sum of the above orthogonal basis functions, i.e.

$$f(x) = \sum_{k=0}^{\infty} c_k \beta_k(x, a)$$
(62)

where coefficients  $c_k$  are calculated using the orthogonality condition

$$c_k = \int_{-\infty}^{+\infty} f(x)\beta_k(x,a)dx$$
(63)

Assuming now that instead of infinite terms in the expansion of Eq. (62), M terms are maintained, then an approximation of f(x) is succeeded. The expansion of f(x) using Eq. (62) is a Gauss-Hermite series. Eq. (62) is a form of Fourier expansion for f(x). Eq. (62) can be considered as the Fourier transform of f(x) subject only to a scale change. Indeed, the Fourier transform of f(x) is given by

$$F(s) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} f(x)e^{-jsx} dx \Longrightarrow f(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(s)e^{jsx} ds$$
(64)

The Fourier transform of the basis function  $\psi_k(x)$  of Eq. (58) satisfies (Refregier, 2003)

$$\Psi_k(s) = j^n \psi_k(s) \tag{65}$$

while for the basis functions  $b_k(x, a)$  using scale coefficient *a* it holds that

$$B_{k}(s,a) = j^{n} \beta_{k}(s,a^{-1})$$
(66)

Therefore, it holds

$$f(x) = \sum_{k=0}^{\infty} c_k \beta_k(x, a) \xrightarrow{F} F(s) = \sum_{k=0}^{\infty} c_k j^n \beta_k(s, a^{-1})$$
(67)

which means that the Fourier transform of Eq. (62) is the same as the initial function, subject only to a change of scale. The structure of a a feed-forward neural network with Hermite basis functions is depicted in Fig. 3(b).

#### 5.2 Neural Networks using 2D Hermite activation functions

Two-dimensional feedforward neural networks can be constructed by taking products of the one dimensional basis functions  $B_k(x, a)$  (Refregier, 2003). Thus, setting  $x = [x_1, x_2]^T$  one can define the following basis functions

$$B_k(x,a) = \frac{1}{a} B_{k_1}(x_1,a) B_{k_2}(x_2,a)$$
(68)

These two dimensional basis functions are again orthonormal, i.e. it holds

$$\int d^{2}x B_{n}(x,a) B_{m}(x,a) = \delta_{n_{1}m_{1}} \delta_{n_{2}m_{2}}$$
(69)

The basis functions  $B_k(x)$  are the eigenstates of the two dimensional harmonic oscillator and form a complete basis for integrable functions of two variables. A two dimensional function f(x) can thus be written in the series expansion:

$$f(x) = \sum_{k_1, k_2}^{\infty} c_k B_k(x, a)$$
(70)

The choice of an appropriate scale coefficient *a* and maximum order  $k_{max}$  is of practical interest. The coefficients  $c_k$  are given by

$$c_k = \int dx^2 f(x) B_k(x, a) \tag{71}$$

Indicative basis functions  $B_2(x, a)$ ,  $B_6(x, a)$ ,  $B_9(x, a)$ ,  $B_{11}(x, a)$  and  $B_{13}(x, a)$ ,  $B_{15}(x, a)$  of a 2D feed-forward neural network with Hermite basis functions are depicted in Fig. 7, Fig. 8 and Fig. 9.



Fig. 7(a). 2D Hermite polynomial basis functions: basis function  $B_2(x, a)$ 



Fig. 8(a). 2D Hermite polynomial basis function  $B_9(x, a)$ 



Fig. 7(b). 2D Hermite polynomial basis functions:  $B_6(x, a)$ 



Fig. 8(b). 2D Hermite polynomial basis functions:  $B_{11}(x, a)$ 



Fig. 9(a). 2D Hermite polynomial basis function  $B_{13}(x, a)$ 

Fig. 9(b). 2D Hermite polynomial basis function  $B_{15}(x, a)$ 

#### 6. Simulation tests

The 2-DOF flexible link robot of Fig. 1 is considered. The robot is planar and consists of two flexible links of length  $L_1 = 0.45m$  and  $L_2 = 0.45m$ , respectively. The dynamic model of the robot is given by Eq. (4). The elements of the inertia matrix M are:

$$M_{11} = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}, \quad M_{12} = M_{21}^T = \begin{pmatrix} 1 & 1 & 0.2 & 0.3 \\ 0.5 & 0.1 & 2 & 0.7 \end{pmatrix}, \quad M_{22} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$
(72)

The damping matrix was taken to be  $D = diag\{0.04, 0.08, 0.03, 0.06\}$  while the stiffness matrix was selected as  $K = \{0.02, 0.04, 0.03, 0.06\}$ . Initially only a PD controller is used. The selection of the gain matrices  $K_p$  and  $K_d$  determines the transient response of the closed loop system. The following controller gains have been considered:  $K_p = diag\{0.2, 0.2\}$  and  $K_d = diag\{0.1, 0.1\}$ . The desirable joints positions are  $\theta_1^d = 1rad$  and  $\theta_2^d = 1.4rad$ . Moreover, it is considered that an additive disturbance torque appears on each joint. The disturbance is given by  $d_i(t) = 0.3 \cos(t)$ . The simulation diagram of Fig. 9(a) shows the evolution in time of the angles of the robot's joints  $\theta_1$  and  $\theta_2$ , respectively, when only a PD controller is used in the loop and the flexibility of the link is not taken into account int the controller's design. In Fig. 9(b) the evolution in time of the vibration modes of the first link  $v_{11}$ ,  $v_{12}$  and of the second link  $v_{21}$  and  $v_{22}$ , respectively, is presented. It can be seen that vibrations around the desirable joint positions cannot be eliminated.

Control of Robotic Systems with Flexible Components using Hermite Polynomial-Based Neural Networks



Fig. 9(a). (a) Diagrams of the angles  $\theta_1$  and  $\theta_2$ , and of the angular velocities  $\omega_1$  and  $\omega_2$  of the joints of the flexible link manipulator when only a PD controller is used



Fig. 10(a). Diagrams of the angles  $\theta_1$  and  $\theta_2$ , and of the angular velocities  $\omega_1$  and  $\omega_2$  of the flexible-link robot when a wavelet-based NN controller is used to suppress vibrations.



Fig. 9(b). Diagrams of the vibration modes the two flexible links of the robotic manipulator when only a PD controller is used



Fig. 10(b). Diagrams of the robot's joint angles  $\theta_1$  and  $\theta_2$ , and of the angular velocities  $\omega_1$  and  $\omega_2$  of the flexible-link robot when a Hermite polynomial-based NN controller is used to suppress vibrations.

Next, a control loop with a NN which uses the Haar orthogonal wavelet functions of subsection 4.5 is considered. The neural controller is a single layer NN with wavelet basis functions, as shown in Fig. 3(b), and it is linear with respect to the output weights.

Fig. 10(a) presents the evolution in time of the joint angles of the robot when NN with wavelet basis functions are used for suppressing the vibrations of the flexible links. Fig. 10(b) shows the variation in time of the joint angles of the robot  $\theta_1$  and  $\theta_2$ , respectively, when the neural controller uses Gauss-Hermite basis functions.

Finally, simulation diagrams are presented showing how the proposed neural controllers succeed the suppression of the vibration modes of the flexible links. Fig. 11 (a) shows the evolution in time of the vibration modes  $v_{11}$  and  $v_{12}$  of the first link, as well as of the

vibration modes  $v_{21}$  and  $v_{22}$  of the second link, when the neural controller uses wavelet basis functions. Similarly, Fig. 11 (b) shows the variation in time of the vibration modes  $v_{11}$ ,  $v_{12}$  of the first link and  $v_{21}$ ,  $v_{22}$  of the second link, respectively, when the neural controller uses Gauss-Hermite basis functions.



Fig. 11(a). Diagrams of the vibration modes of the two flexilbe links of the robotic manipulator, when a wavelet-based NN controller is used to suppress vibrations



Fig. 11(b). Diagrams of the vibration modes of the two flexilbe links of the robotic manipulator, when a Hermite polynomialbased NN controller is used to suppress vibrations

From the simulation experiments it can be observed, that using a neural controller with basis functions which are localized both in space and frequency allows better approximation of the multi-frequency characteristics of the vibrating robot links. The angles of the robot's joints converge to the desirable set-points  $\theta_1^d$  and  $\theta_2^d$ , while fast and efficient suppression of the vibration modes  $v_{11}$ ,  $v_{12}$  and  $v_{21}$ ,  $v_{22}$  is also succeeded. Finally, in the above simulation experiments it was observed that the control signal (torque) generated by the neural controller with Hermite basis functions was smoother than the control signal of the neural controller which employs Haar wavelet basis functions.

Comparing the Gauss-Hermite basis functions to the Haar wavelet basis functions one can note the following: (i) both are basis functions which are local in space and spatial frequency. This allows better approximation of the multi-frequency characteristics of vibrating structures, such as the flexible-link robot, (ii) both satisfy the orthogonality property which helps to locally improve the accuracy of approximation of the unknown system dynamics. This means that the neural controller can be dynamically expanded by adding new basis functions which are orthogonal to the existing ones, while the coefficients of the new basis functions can be computed independently of the existing coefficients (iii) unlike wavelet basis functions, the Gauss-Hermite basis functions have a clear physical meaning, since they represent the solutions of differential equations of stochastic oscillators and each neuron can be regarded as the frequency filter of the respective vibration eigenfrequency, (iv) unlike the Haar wavelet basis functions, the Gauss-Hermite basis functions remain almost unchanged by the Fourier transform, which means that the weights of the associated neural network demonstrate the energy which is distributed in the various eigenmodes of the vibrating structure. This in turn allows to define thresholds for truncating the expansion and using neural controller with a small number of nodes and weight coefficients.

#### 7. Conclusions

Neural networks with Gauss-Hermite polynomial basis functions have been proposed for the control of flexible-link manipulators. The Gauss-Hermite basis functions are local in both space and spatial frequency. Locality in spatial frequency allows the representation to be adaptively tuned to the variations of the local bandwidth of the system dynamics. Moreover,adding new basis functions helps to locally improve the accuracy of approximation of the unknown system dynamics. Since the new basis functions are orthogonal to the existing ones, the new NN weights can be computed independently of the existing weights. Comparing to Haar wavelet basis functions which are also orthogonal, the Gauss-Hermite basis functions have some interesting properties: (i) they remain almost unchanged by the Fourier transform, which means that the weights of the associated neural network demonstrate the energy which is distributed to the various eigenmodes of the vibrating structure, (ii) unlike wavelet basis functions the Gauss-Hermite basis functions have a clear physical meaning since they represent the solutions of differential equations of stochastic oscillators and each neuron can be regarded as the frequency filter of the respective vibration eigenfrequency.

The proposed neural controller operates in parallel to a PD controller the gains of which have been selected assuming rigid-link robot dynamics. The performance of the control scheme has been tested through simulation experiments. Comparison to a neural controller with Haar orthogonal wavelet functions has been provided. The simulation tests showed that using a neural controller with basis functions which are localized both in space and frequency allows for better approximation of the multi-frequency characteristics of the vibrating robot links.

#### 8. References

Addison P.S (2002). *The illustrared wavelet transform handbook,* Institute of Physics Publishing. Aoustin Y.; Fliess M., Mounier H., Rouchon P., & Rudoplh J. (1997). Theory and practice in

- the motion planning control of a flexible robot arm using Mikusinki operators, *Proc.* of 4th Symposium on Robotics and Control, pp. 287-293 Nantes.
- Arteaga M.A. & Siciliano B. (2000). On Tracking Control of Flexible Robot Arms, *IEEE Transactions on Automatic Control*, Vol. 45, No. 3, pp. 520-527.
- Benosman A. & Le Vey G (2004). Control of flexible manipulators: a survey, *Robotica*, Cambridge University Press, Vol. 22, No 5, pp. 533-545.
- Bernard C.P. & Slotine J.J.E, (1997). Adaptive Control with Multiresolution *Bases, Proc. IEEE* CDC 97, 36th IEEE Intl. Conference on Decision and Control, San Diego, Califormia, USA.
- Cannon M. & Slotine J.J.E. (1995). Space-frequency localized basis function networks for nonlinear system estimation and control, *Neurocomputing*, Elsevier, Vol. 9, No 3, pp. 293-342.

- Caswara F.M. & Unbenhauen H. (2002). A neuro-fuzzy approach to the control of a flexiblelink manipulator, *IEEE Transactions on Robotics and Automation*, Vol.18, No.6, pp. 932-944.
- Cetinkunt S. & Yu W.L. (1991). Closed Loop Behavior of a Feedback Controlled Flexible Arm: A Comparative Study, *International Journal of Robotics Research*, Sage Publications, Vol. 10, No. 3, pp. 263-275.
- Ge S.S.; Lee T.H. & Harris C.J. (1998). Adaptive neural network control of robotic manipulators, World Scientific.
- Guterrez L.B.; Lewis F.L. & Lowe J.A. (1998). Implementation of neural network tracking for a single flexible link: comparison with PD and PID controllers, *IEEE Transactions on Industrial Electronics*, Vol.45, No.2, pp. 307-318.
- Haykin, S (1994). Neural Networks: A Comprehensive Foundation, McMillan.
- Hui D.; Fuchun S. & Zengqi S. (2002). Observer-based adaptive controller design of flexible manipulators using time-delay neuro-fuzzy networks, *Journal of Intelligent and Robotic Systems*, Springer, Vol. 34, No.4, pp. 453-466.
- Lewis F.L.; Jagannathan S. & Yesildirek A. (1999). Neural Network control of Robot Manipulators and Nonlinear Systems, Taylor and Francis.
- Lin C.K. (2006). Nonsingular terminal sliding mode control of robot manipulators using fuzzy wavelet networks, *IEEE Transactions on Fuzzy Systems*, Vol. 14, No 6, pp. 849-859.
- Lin J. & Lewis F.L. (2002). Fuzzy controller for flexible-link robot arm by reduced-order techniques, *IEE Proceedings on Control Theory and Applications*, Vol. 149, No.3, pp. 177-187.
- De Luca A. & Siciliano B. (1993). Regualation of flexible arms under gravity, *IEEE Transactions on Robotics and Automation*, Vol.9, No 4, pp. 463-467.
- Kanoh H.; Tzafestas S.G., Lee H.G. & Kalat J. (1986). Modelling and control of flexible robot arms, *Proc IEE CDC86, 25th Conference on Decision and Control,* pp. 1866-1870, Athens, Greece.
- Karimi H.R. & Lohmann B. (2006). A Computational Method to Robust Vibration Control of Vehicle Engine-Body System using Haar Wavelets, *Proc. of the 2006 IEEE International Conference on Control Applications*, pp. 169-174, Munich, Germany.
- Krzyzak A. & Sasiadek J. (1991). Flexible robot identification using nonparametric techniques. *Proc.IEEE CDC91, 30th IEEE Conference on Decision and Control,* Brighton UK.
- Mallat S (1999). A wavelet tour of signal processing, 2nd Edition, Acedemic Press.
- Nguyen V.B. & Morris A.S. (2007). Genetic algorithm tuned fuzzy logic controller for a robot arm with two-link flexibility and two-joint elasticity, *Journal of Intelligent and Robotic Systems*, Springer, Vol. 49, No.1, pp. 3-18.
- Refregier A. (2003). Shapelets I. A method for image analysis. *Mon. Not. R. Astron. Soc.*, Vol. 338, No 1, pp. 35-47.
- Renno J.M. (2007). Inverse dynamics-based tuning of a fuzzy logic controller for a single-link flexible manipulator, *Journal of Vibrations Control*, Sage Publications, Vol.13, No.12, pp. 1741-1759, 2007.
- Rigatos G.G. (2009). Model-based and model-free control of flexible-link robots: a comparison between representative methods, *Applied Mathematical Modelling*, Elsevier, Vol.33, No 10, pp. 3906-3925.

- Rigatos G.G. (2006). Model-free control of flexible-link robots, *Proc. IC-SCCE 2006, 2nd Intl. Conference "From Scientific Computing to Computational Engineering,* Athens, Greece.
- Rigatos G.G. & Tzafestas S.G. (2006). Feed-forward neural networks using Hermite polynomial activation functions. *Lecture Notes in Artificial Intelligence*, Springer, Vol. 3925, pp. 323-333.
- Shi L.Z. & Trabia M.B. (2006). Design and tuning of importance-based fuzzy logic controller for a flexible-link manipulator, *Journal of Intelligent and Fuzzy Systems*, IOS press, Vol. 17, No.3, pp. 313-323.
- Subudhi B. & Morris A.S. (2003). Fuzzy and neuro-fuzzy approaches to control a flexible single-link manipulator, *Proc. IMechE, Journal of Systems and Control Engineering*, Vol. 217, pp. 387-399.
- Subudhi B. & Morris A.S. (2009). Soft computing methods applied to the control of a flexible robot manipulator, *Applied Soft Computing*, Elsevier, Vol 9, No. 1, pp. 149-158.
- Sureshbabu N. & Farell J.A. (1999). Wavelet-based system identification for nonlinear control. *IEEE Transactions on Automatic Control*, Vol. 44, No, 2, pp. 412-417.
- Talebi H.A.; Khorasani K. & Patel R.V. (1998). Neural network based control schemes for flexible-link manipulators: simulations and experiments, *Neural Networks*, Elsevier, Vol.11, No. 7-8, pp. 1357-1377.
- Tian L. & Collins C. (2005). Adaptive neuro-fuzzy control of a flexible manipulator, *Journal of Mechatronics*, Elsevier, Vol. 15, No. 10, pp. 1305-1320.
- Tian L.; Wang J. & Mao Z. (2004). Constrained motion control of flexible robot manipulators based on recurrent neural networks, *IEEE Transactions on System, Man and Cybernetics Part B: Cybernetics*, Vol. 34, No. 3, pp. 1541-1552.
- Torrésani B (1995). Analyse continue par ondelettes, CNRS Editions.
- Tzafestas S.G. & Rigatos G.G. (1998). Neural and Neuro-fuzzy FELA Adaptive Robot Control Using Feedforward and Counterpropagation Networks, *Journal of Intelligent and Robotic Systems*, Springer, Vol. 23, No 2-4, pp. 291-330.
- Wai R.J. & Lee M.C. (2004). Intelligent optimal control of a single-link, flexible robot arm, *IEEE Transactions on Industrial Electronics*, Vol. 51, No.1, pp. 201-220.
- Wang F.Y. & Gao Y (2005). Advanced Studies of Flexible Robotic Manipulators, World Scientific.
- Yang S.S. & Cheng C.S. (1996). An orthogonal neural network for function approximation. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, Vol. 26 No. 5, 779-784.
- Zhang Q. & Benveniste A. (1993). Wavelet Networks. *IEEE Transactions on Neural Networks*, Vol. 3, No. 6, pp. 889-898.
- Zuo W. ; Zhu Y. & Cai L. (2009). Fourier-Neural-Network-Based Learning control for a class of nonlinear systems with flexible components. *IEEE Transactions on Neural Networks*, Vol. 20, No. 1, pp. 139-151.

# IntechOpen

## IntechOpen



**Robot Manipulators New Achievements** Edited by Aleksandar Lazinica and Hiroyuki Kawai

ISBN 978-953-307-090-2 Hard cover, 718 pages Publisher InTech Published online 01, April, 2010 Published in print edition April, 2010

Robot manipulators are developing more in the direction of industrial robots than of human workers. Recently, the applications of robot manipulators are spreading their focus, for example Da Vinci as a medical robot, ASIMO as a humanoid robot and so on. There are many research topics within the field of robot manipulators, e.g. motion planning, cooperation with a human, and fusion with external sensors like vision, haptic and force, etc. Moreover, these include both technical problems in the industry and theoretical problems in the academic fields. This book is a collection of papers presenting the latest research issues from around the world.

#### How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Gerasimos G. Rigatos (2010). Control of Robotic Systems with Flexible Components Using Hermite Polynomial-Based Neural Networks, Robot Manipulators New Achievements, Aleksandar Lazinica and Hiroyuki Kawai (Ed.), ISBN: 978-953-307-090-2, InTech, Available from: http://www.intechopen.com/books/robotmanipulators-new-achievements/control-of-robotic-systems-with-flexible-components-using-hermitepolynomial-based-neural-networks

### INTECH

open science | open minds

#### InTech Europe

University Campus STeP Ri Slavka Krautzeka 83/A 51000 Rijeka, Croatia Phone: +385 (51) 770 447 Fax: +385 (51) 686 166 www.intechopen.com

#### InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai No.65, Yan An Road (West), Shanghai, 200040, China 中国上海市延安西路65号上海国际贵都大饭店办公楼405单元 Phone: +86-21-62489820 Fax: +86-21-62489821 © 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the <u>Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License</u>, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.



# IntechOpen