Geometric Identification and Control of Nonlinear Dynamic Systems Based on Floating Basis Vector Representation

József K. Tar, Imre J. Rudas

Institute of Intelligent Engineering Systems,
John von Neumann Faculty of Informatics
Budapest Tech
Bécsi út 96/B
H-1034 Budapest
Hungary
tar.jozsef@nik.bmf.hu, rudas@bmf.hu

Miklós Rontó (student)

Faculty of Science,
Eötvös Loránd University
Pázmány Péter sétány 1/A
H-1117 Budapest
Hungary
ronto@t-online.hu

Abstract: In this paper a simple adaptive controller is outlined that creates only temporal and situation-dependent system model. It may be a plausible alternative of the more sophisticated soft computing approaches that aim the identification of permanent and complete models. The temporal model can be built up and maintained step-by-step on the basis of slow elimination of fading information by the use of simple updating rules consisting of finite algebraic steps of lucid geometric interpretation. It may be used for filling in the "lookup tables" or rule bases of the more sophisticated representations experimentally. The method applies simple elimination of the the casual algebraic singularities the occurrence of which cannot be evaded in the practice. The operation of the method is illustrated by the control of a 2 Degrees Of Freedom dynamic system as a typical paradigm via simulation.

Keywords: Adaptive Control; Floating Basis Vector Representation; Nonlinear Control.

1 Introduction

Though strictly stable controller designs already have been proposed on the basis of infinite order models [1, 2], too, the necessary mathematical deductions are very complicated and their complexity strongly increases with the increase in dimensionality. A few research efforts have also been reported which solved this problem by energy based controller design methods that could provide a relatively simple control in spite of considering infinite-dimensional model. Such controllers are normally designed by the use of certain Lyapunov function, and ensure Lyapunov stability [3, 4]. However, the control of infinite-order physical systems are commonly based on finite order approximations in which the infinite modes are neglected for ease of design as e.g. in [5, 6]. These finite order models lead to handling discrete time-series only.

Another important class of physical systems' control is the set of non-stationary stochastic processes in which some deterministic response to an external input and a stationary stochastic process are superimposed. This is relevant, for instance, when the external input cannot be effectively described by some probabilistic distribution. A discrete time model can be formulated in the form of a difference equation with an external input $\{u_k\}$ that is usually considered to be known (Autoregressive Moving Average Model with external input - ARMAX) [7]:

$$y_{k+1} = \sum_{s=0}^{N} a_s y_{k-s} + \sum_{w=0}^{M} b_w u_{k-w} \tag{1}$$

In the so-called Takagi-Sugeno fuzzy models the consequent parts are expressed by analytical expressions similar to (1). The TS fuzzy controllers use some linear combinations of the (1)-type rules in which the coefficients depend on the antecedents. With the help of such Takagi-Sugeno fuzzy IF-THEN rules sufficient conditions to check the stability of fuzzy control systems are now available. These schemes are based on the stability theory of interval matrices and those of the Lyapunov approach [8]. It was already observed that the fuzzy controller stability conditions can be rewritten in form of Linear Matrix Inequalities (LMIs) [9, 10]. LMIs can be efficiently solved numerically by solving very complex Riccati equations for a positive definite solution [11].

Neural Networks in general are useful means of developing nonlinear models. A particular case of such applications is when the model itself consists of certain nonlinear mapping, for instance in the linearization of the nonlinear characteristics of various sensors [12]. Neuro-fuzzy systems provide the fuzzy systems with automatic tuning systems using Neural Network (NN) as a tool. The Adaptive Neuro-Fuzzy Inference System (ANFIS) is a cross between an artificial neural network and a Fuzzy Inference System (FIS) [8, 13, 14, 15]. The adaptive network can be a multi-layer feed-forward network in which each node (neuron) performs a particular function on incoming signals. Based on the ability of an ANFIS to learn from training data, it is possible to create an ANFIS structure from an extremely limited mathematical representation of the system. The ANFIS system generated by the fuzzy toolbox available in MATLAB allows the generation of a standard Sugeno style fuzzy inference system or a fuzzy inference system based on sub-clustering of the data [16]. Radial Basis Function Networks (RBFNs) provide an attractive alternative to the standard Feedforward Networks using backpropagation learning technique [17]. The linear weights associated with the output layer can be treated separately from the hidden layer neurons. As the hidden layer weights are adjusted through a nonlinear optimization, output layer weights are adjusted through linear optimization [8]. In fact the nodes of a RBFN represent "fuzzified" or "blurred" regions which correspond to the well defined antecedent sets of a fuzzy controller. The neuron's firing achieves its maximum at the centre of the region while its strength decreases with the distance from the center according to some Gaussian function (various distance measures can also be used). In many cases development of the whole model is a complicated task especially when the "antecedent" part is strongly nonlinear multivariable function of the input. Evolutionary methods as e.g. the Particle Swarm Optimization Method that realizes stochastic random search in a multi-dimensional optimization space [18, 19] therefore may also be combined with them. In the case of certain problem classes similarity relations can also be observed and utilized to simplify the design process [20].

A significant common feature of the above approaches is that they try to develop a "complete" soft computing based model of the system to be controlled. This naturally makes the question arise whether it is always reasonable to try to identify a "complete" model. As a plausible alternative simple adaptive controllers can be imagined that do not wish to create a complete model. Instead of that on the basis of slowly fading recent information a more or less temporal model can be constructed and updated step by step by the use of simple updating rules consisting of finite algebraic steps of lucid geometric interpretation. This method may be used for filling in lookup tables of the above representations experimentally. In the sequel this simple approach is detailed and illustrated via simulation results.

2 Geometric Approach for Dynamic Systems

Consider a simple nonlinear causal Multiple Input – Multiple Output (MIMO) system described by the equation:

$$\mathbf{y}^{(n)}(t) = \mathbf{F}\left(\mathbf{y}^{(n-1)}(t), \mathbf{y}^{(n-2)}(t), \ldots, \mathbf{y}^{(0)}(t), \mathbf{f}(t)\right) \tag{2}$$

in which $\mathbf{f}(t)$ represents the external driving forces to be utilized for controlling purposes. Let us suppose that the time-derivatives can be approached by certain finite element approach using time-resolution δt. To numerically estimate the n^{th} order time-derivatives at least (n+1) discrete values has to be taken into account via considering their linear combination as

$$\mathbf{y}^{(n)}(t) \cong \sum_{s=0}^{n} \mathbf{c}_s(\delta t)\mathbf{y}(t - s\delta t) \tag{3}$$

in which the $\mathbf{c}_n$ coefficients depend on δt and can be chosen in various manners. We also note that the number of the coefficients may be somewhat greater that (n+1), e.g. in the case of computing the central first derivatives we may use 3 points, too. Via rearranging (2) and using (3) the following ambiguous representation can be obtained:

$$\mathbf{y}(t) \cong \mathbf{\Phi}(\mathbf{y}(t-\delta t), \mathbf{y}(t-2\delta t), \ldots, \mathbf{y}(t-n\delta t), \mathbf{f}(t-\delta t)) \tag{4}$$

in which the actually used values are concentrated in the vicinity of the values of time t. Supposing that the array of the values $\mathbf{Y}_f$:=[$\mathbf{y}(t\text{-}\delta t)$,…, $\mathbf{y}(t\text{-}n\delta t)$, $\mathbf{f}(t\text{-}\delta t)]^T \neq \mathbf{0}$ (4) can be replaced by a scalar product in ambiguous manner by an array $\mathbf{G}$ as

$$\mathbf{y}(t) = \mathbf{G}^T(t)\mathbf{Y}_f(t) \tag{5}$$

in which both the angle between $\mathbf{g}$ and $\mathbf{y}$ and the absolute value of $\mathbf{g}$ are not well defined. If the n^{th} derivative of $\mathbf{y}(t)$ is directly measurable similar ambiguous approximation can be constructed for $\mathbf{y}^{(n)}(t)$ as

$$\mathbf{y}^{(n)}(t) = \mathbf{g}^T(t)\mathbf{Y}_f(t)\,. \tag{6}$$

Let us suppose that on the basis of some rough initial or preliminary model we can compute the appropriate control action $\mathbf{f}(t)$ and can store the $\mathbf{y}(t)$ values, too. It is evident that in the case of a time-invariant linear system $\mathbf{g}$ does not depend on t, therefore collecting sufficient information coded in the form of (6) leads to the system of linear equations that belong to the constant array $\mathbf{g}$ as

$$\begin{bmatrix} \mathbf{y}^{(n)}(t-\delta t) \\ \ldots \\ \mathbf{y}^{(n)}(t-M\delta t) \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_f^T(t-\delta t) \\ \ldots \\ \mathbf{Y}_f^T(t-M\delta t) \end{bmatrix} \mathbf{g}\,. \tag{7}$$

Equation (7) has very simple and lucid geometric interpretation: the constant vector $\mathbf{g}$ is represented by time-varying or "floating" system of basis vectors $\mathbf{Y}_f(t\text{-}n\delta t)$ (n=1,…,M). If this set is linearly independent $\mathbf{g}$ can be reproduced as the linear combination of these vectors as

$$\mathbf{g} = \sum_{s=1}^{M} \mu_s(t)\mathbf{Y}_f^T(t - s\delta t) \tag{8}$$

In (8) it is naturally supposed that for a constant $\mathbf{g}$ for a floating system of basis vectors a floating or time-varying system of the $\mu_s(t)$ coefficients belongs in a special manner that they together can provide a constant vector. No let us suppose that we have two vectors **a** and **b** having known dot product with **g**. Let us find the component of **b** in the orthogonal subspace of **a** in the form of $\mathbf{b}_\perp=\mathbf{b}+\lambda\mathbf{a}$:

$$0=\mathbf{a}^T\mathbf{b}_\perp=\mathbf{a}^T\mathbf{b}+\lambda\mathbf{a}^T\mathbf{a}\Rightarrow\lambda=\frac{-\mathbf{a}^T\mathbf{b}}{\mathbf{a}^T\mathbf{a}}\,. \tag{9}$$

Due to the linear property of the dot or "scalar" product the dot product of $\mathbf{b}_\perp$ with **g** can also be computed as

$$\mathbf{g}^T\mathbf{b}_\perp=\mathbf{g}^T\mathbf{b}+\lambda\mathbf{g}^T\mathbf{a}\,. \tag{10}$$

Now let us apply the following algorithm that is similar to the Gram-Schmidt orthogonalization with the exception of normalizing the vectors: remove the components in the direction of $\mathbf{Y}_f(t\text{-}\delta t)$ from $\mathbf{Y}_f(t\text{-}2\delta t)$,…, $\mathbf{Y}_f(t\text{-}M\delta t)$ with the method given in (9). Then the new set indexed with 2,3,…M-1 will be in the orthogonal subspace of $\mathbf{Y}_f(t\text{-}\delta t)$. Then the 2nd vector of the remaining set and subtract the components of the remaining ones in its direction, etc. while tracing the variation of the dot products according to (10). (To avoid numerical difficulties the components in the direction of very small vectors need no to be subtracted.) Furthermore, since in the case of linear systems it is just enough to obtain sufficient information on the *independent directions* only, the approximately same direction of vectors **a** and **b** can be stated if

$$\left|\cos\varphi(\mathbf{a},\mathbf{b})\right|\cong\frac{\left|\mathbf{a}^T\mathbf{b}\right|}{|\mathbf{a}|\times|\mathbf{b}|+\varepsilon_1}\geq 1-\varepsilon_2\,. \tag{11}$$

in which ε_1 and ε_2 are small positive number. Otherwise these vectors have *essentially different directions*. Now let use suppose that we continue the systematic observation and obtain further information on **g** in the form of (6) as

$$\mathbf{y}^{(n)}(t+\delta t)=\mathbf{g}^T(t)\mathbf{Y}_f(t+\delta t)\,. \tag{12}$$

Together with the information coded in (7) (12) is *redundant but free of contradiction* if **g** is exactly constant. In this case either (12) or one of the vectors in (7) can be dropped, replaced with the 1st vector in the set in (7), and the orthogonalization algorithm can be repeated. As a result the same constant **g** must be obtained by the use of this new set of basis vectors.

Now let us suppose that our system is *linear* but *not time-invariant*! In this case (7) and (12) are rather controversial than redundant because these vectors do not belong exactly to the same **g** since they were obtained from measurements taken in different time instances. A plausible and lucid method of contradiction resolution may be finding the vector in the closest direction of the last one in the sense of (11) since the remaining vectors convey less relevant information on the system's behavior in this direction. This vector can be omitted in the system in (7) and it can be replaced by the new information conveyed by (12). Then by executing the orthogonalization algorithm on the remaining set the "*obsolete information regarding the new direction*" can be removed and replaced by the fresh information. [Since the addition in (8) is commutative, in practice the first column of the original set can be put in the place of the dropped vector, and the new one can be placed into the 1st place.]

Finally let us suppose that our system is neither time-invariant nor linear! In this case not only the direction but the absolute values of the vectors also influence the behavior of the system. In this case the old vector closest to the new one in the sense of a norm can be dropped and replaced by the new one because the information mainly conveyed by it is "refreshed".

In the possession of some prescribed control strategy formulating the desired trajectory tracking with asymptotic convergence continuous tracking error is expected and the array **g** in (8) can be used for calculating the necessary control action instead of the rough initial model as

$$\mathbf{f}(t)=\mathbf{g}_M^{-1}\left\{\mathbf{y}^{(n)Desired}(t)-\begin{bmatrix}\mathbf{g}_1 & \dots & \mathbf{g}_{M-1}\end{bmatrix}\begin{bmatrix}\mathbf{Y}_1\\ \dots \\ \mathbf{Y}_{M-1}\end{bmatrix}\right\} \tag{13}$$

in which the quadratic matrix $\mathbf{g}_M$ corresponds to those part of **g** in which the coefficients of $\mathbf{f}(t)$ are placed. In (13) it is supposed that $\mathbf{g}_M$ is invertible. However, since **g** is obtained via observaton of often noisy signal this supposition is no always substantiated. To obtain a useful control signal even in the case of singularity two plausile methods are applied in this paper. The first one is based on the algebraic adjoint of $\mathbf{g}_M$. Since for an invertible quadratic matrix $\mathbf{M}^{-1}$=Adj($\mathbf{M}$)/det($\mathbf{M}$), and in Adj($\mathbf{M}$) only the linear combinations of the products of the matrix elements of **M** occur, instead of $\mathbf{M}^{-1}$ the following matrix containing a small positive constant ε can be used:

$$\tilde{\mathbf{M}}=Adj(\mathbf{M})\frac{sign(\det(\mathbf{M}))}{|\det(\mathbf{M})|+\varepsilon} \tag{14}$$

If $|\det(\mathbf{M})|>>\varepsilon$ this matrix well approximates $\mathbf{M}^{-1}$, otherwise (14) yields an unique value the geometric interpretation of which is not very lucid. An alternative approach utilizes the linearity of the scalar product and the matrix product. If **M** has an inverse, and **a**, **b**, **c,** and **w** are vectors, λ is a scalar, if $\mathbf{c}=\mathbf{a}+\lambda\mathbf{b}$, then $\mathbf{M}^{-1}\mathbf{c}=\mathbf{M}^{-1}\mathbf{a}+\lambda\mathbf{M}^{-1}\mathbf{b}$, and $\mathbf{w}^T\mathbf{c}=\mathbf{w}^T\mathbf{a}+\lambda\mathbf{w}^T\mathbf{b}$. Therefore if the effect of $\mathbf{M}^{-1}$ is known on a set of linearly independent vectors, its effect also is known on an arbitrary vector if this vector is expressed as the linear combination of these vectors. If **I** is the unit matrix, $\mathbf{I}=\mathbf{M}^{-1}\mathbf{M}$, therefore effect of $\mathbf{M}^{-1}$ is well known on the columns of **M**: they are the columns of **I**. (If **M** is invertible, its columns serve as a full set of basis vectors.) These vectors can be orthogonalized by the Gram-Schmidt algorithm without normalizing the vectors during this process. The vectors the norm of which is under a limit are replaced by zero vectors. In the case of using an orthogonal set if $\mathbf{w} = \sum_{i=1}^{n} \nu_i \mathbf{u}^{(i)} \Rightarrow \nu_i = \frac{\mathbf{w}^T \mathbf{u}^{(i)}}{\mathbf{u}^{(i)T} \mathbf{u}^{(i)}}$. At the beginning of the algorithm the scalar product of vector **w** is computed with the columns of **M**. Following the Gram-Schmidt algoritmh the scalar product of **w** with the new vectors is computed according to the above observation. In the possession of the scalar products and the norms the ν_i coefficients can simply be computed to find $\mathbf{M}^{-1}\mathbf{w}$. If the columns of **M** are linearly dependent, and there is a solution to the problem, a well defined element of the ambiguous solutions is obtained. If there is no solution, the part of **w** in the orthogonal subspace of the columns of **M** is omitted, and a well defined elelement of the ambiguous approximations of **w** is obtained.

3 The Model of the Cart and Pendulum System

The Euler-Lagrange Equations of motion of the system is

$$\begin{bmatrix} M+m & mL\cos\varphi \\ mL\cos\varphi & I+mL^2 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{\varphi} \end{bmatrix} = \begin{bmatrix} -b\dot{x} - mL\sin\varphi\dot{\varphi}^2 + Q_1 \\ -f\dot{\varphi} + mgL\sin\varphi + Q_2 \end{bmatrix} \tag{15}$$

in which some “realistic” data were used, i.e. M=1.096 *kg* and m=0.109 *kg* denote the mass of the cart and the pendulum, L= 0.25 *m* and φ [*rad*] is the length and the rotational angle of the pendulum with respect to the upper vertical direction (clockwisely), x [*m*] denotes the horizontal translation of the cart+pendulum system in the right direction, b=0.1 *N/(m/s)* and f=0.00218 $kg\times m^2/s$ are viscous friction coefficients, I=0.0034 $kg\times m^2$ denotes the momentum of the arm of the pendulum, and Q_1 [*N*] denotes the force horizontally accelerating the whole system. In the forthcoming simulations this system was “identified”.

4 Simulation Results

In the simulations the evenly distributed joint coordinate acceleration measuremement noise was supposed. In both cases ε=10^{-3} was chosen. In Fig. 1 the appropriate trajectory tracking and in a finer resolution the tracking error vs. time is described. In both cases one can realize the initial “learning phase” on the graphs as well as the session of adaptation.

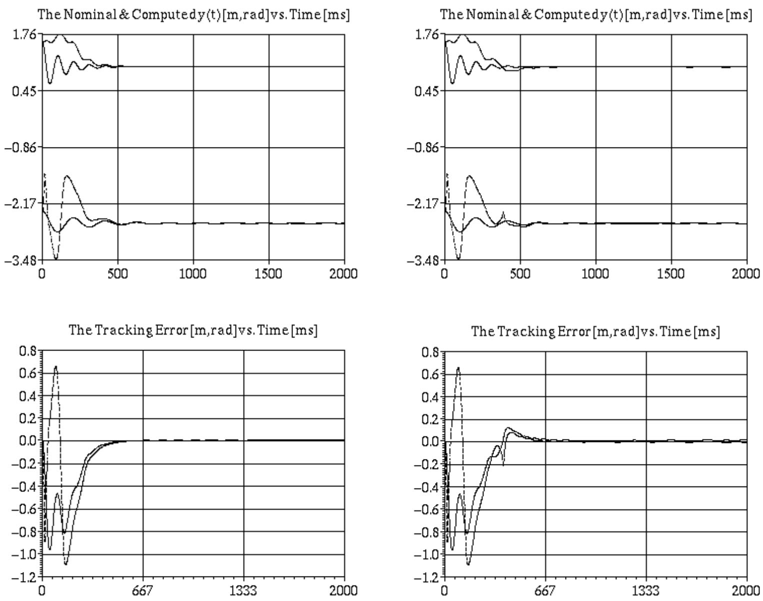

Figure 1

Trajectory tracking (1st row) and tracking error (2nd row) for the geometrically interpreted singularity avoidance (left column), and the algebraic adjoint based method (right column)

The geometrically interpreted singularity avoidance and inversion method seems to yield more stable results.

The same information is communicated by the figures containing the identified components of the $\mathbf{g}_M$ matrix (Fig. 2). The presence of the random noise can better be revealed in the case of the geometrically interpreted solution, especially on the last two graphs.

Figure 3 reveals that normally the system has a “dominant” basis vector that – due to the operation of the algorithm applied – normally stands in the 1st place, and it also has small components (the remaining basis vectors) that seem to be responsible for minor corrections in the prediction. As it was expected their little norm does not cause numerical problems in the calculations.

It also reveals the necessity of gradually letting the “obsolete” information fade.

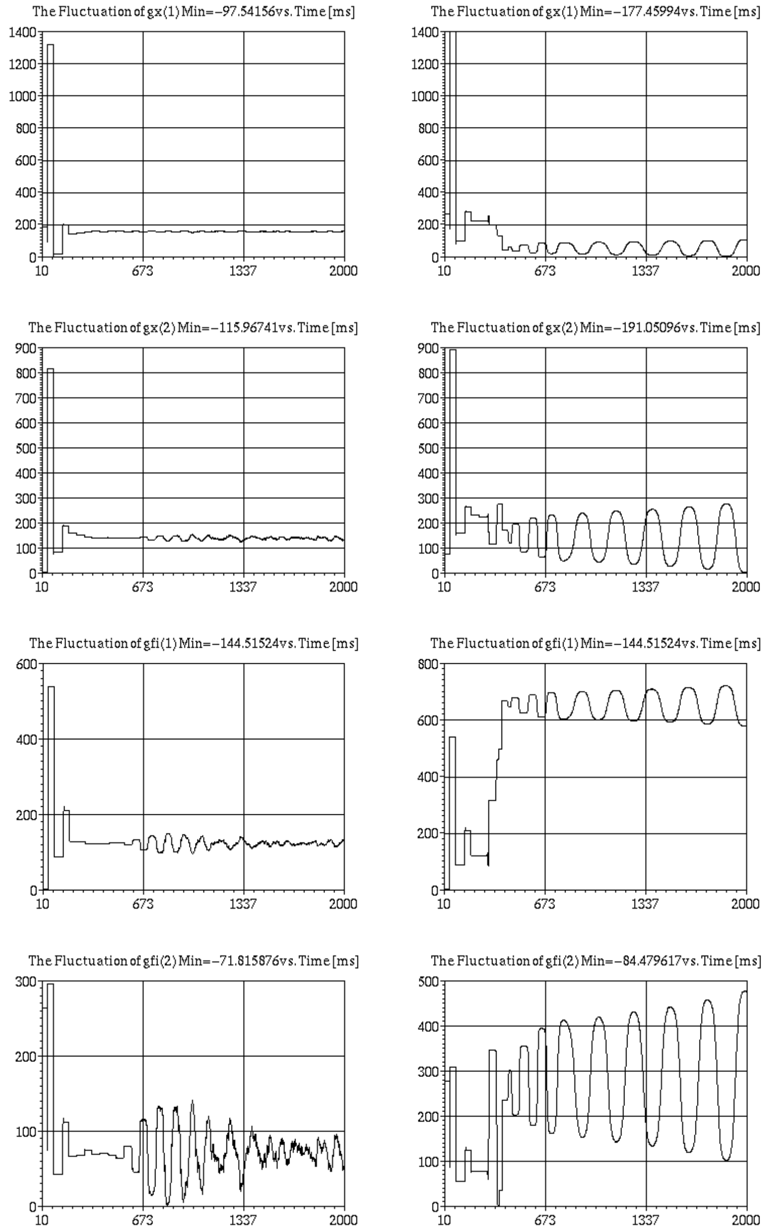

Figure 2

The identified components for the geometrically interpreted singularity avoidance (left column), and the algebraic adjoint based method (right column)

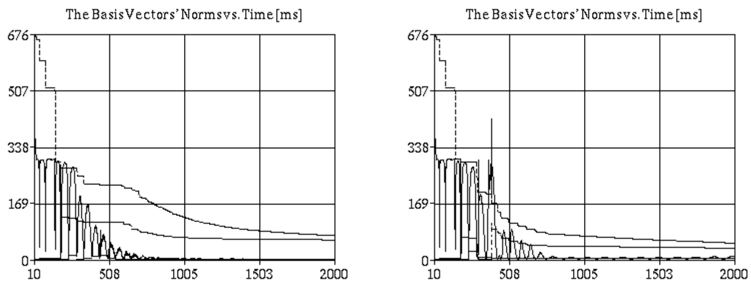

Figure 3

The norms of the floating basis vector system for the geometrically interpreted singularity avoidance (left column), and the algebraic adjoint based method (right column)

Conclusions

In this paper, as a plausible alternative of certain sophisticated soft computing approaches trying to identify "complete" system models, a simple adaptive controller dealing with continuously updated temporal model was investigated via simulation in the case of a 2 Degree Of Freedom nonlinear system.

This model utilizes the slowly fading information via applying finite algebraic steps of lucid geometric interpretation based on the Gram-Schmidt orthogonalization algorithm. The simulation investigations indicated that this approach can be useful. Its great advantage is simplicity, limited number of algebraic operations and lucid interpretation. In contrast to the mathematically far more intricate solutions based on the Lyapunov technique normally guaranteeing Lyapunov stability without making it possible to prescribe dynamic details of trajectory tracking this simple approach makes it possible to prescribe arbitrary error relaxation by the use of simple kinematic terms. Neither complicated evolutionary computation or LMIs based optimizations seems to be necessary for its use. The method may be used for filling in the "lookup tables" or rule bases of the above representations experimentally. Further investigations concerning the operation of this approach in the cases of e.g. fractional order linear or nonlinear systems seem to be expedient in the future.

References

[1] Ö. Morgül, "Stabilization and disturbance rejection for the beam equation," IEEE Trans. Automatic Control, vol. 46, no. 2, pp. 1913-1918, 2001

[2] B. Guo and Z. Luo, "Initial-boundary value problem and exponential decay for a flexible beam vibration with gain adaptive direct strain feedback control," Nonlinear Analysis, Theory, Methods and Applications, vol. 27, no. 3, pp. 353-365, 1996

[3] S. S. Ge, T. H. Lee, and G. Zhu, "Energy-based robust controller design for multi-link flexible robots," Mechatronics, vol. 6, no. 7, pp. 779-798, 1996

[4] X. Zhang, W. Xu, S. S. Nair, and V. Chellaboina, "PDE modeling and control of a flexible twolink manipulator," IEEE Tran. Control System Technology, vol. 13, no. 2, pp. 301-312, March, 2005

[5] P. V. Kokotovic, H. K. Khalil, and J. O'Reilly, Singular Perturbation Methods in Control: Analysis and Design. New York: Academic Press, 1986

[6] J. Lin and F. L. Lewis, "Two-time scale fuzzy logic controller of flexible link robot arm," Fuzzy Sets and Systems, vol. 139, pp. 125-149, 2003

[7] The Mechatronics Handbook. Editor-in-Chief: Robert H. Bishop, common issue by ISA – The Instrumentation, Systems, and Automation Society and CRC Press, Boca Raton London New York Washington, D.C., ISBN: 0-8493-0066-5, (2002)

[8] Intelligent Control Systems Using Soft Computing Methodologies. Eds. Ali Zilouchian & Mo Jamshidi, CRC Press, Boca Raton London New York Washington, D.C., ISBN: 0-8493-1875-0, (2001)

[9] H. O. Wang, K. Tanaka, and M. F. Griffin, "Parallel distributed compensation of nonlinear systems by Takagi-Sugeno fuzzy model," in Proceedings of the FUZZ-IEEE/IFES'95, 1995, pp. 531-538

[10] H. O. Wang, K. Tanaka, and M. F. Griffin, "An analytical framework of fuzzy modeling and control of nonlinear systems: stability and design issues," in Proceedings of the 1995 American Control Conference, Seattle, 1995, pp. 2272-2276

[11] K. Tanaka, T. Ikeda, and H. O. Wang, "Robust stabilization of a class of uncertain nonlinear systems via fuzzy control: quadratic stabilizability, $H\infty$ control theory and linear matrix inequalities," IEEE Tran. Fuzzy Systems, vol. 4, no. 1, pp. 1-13, February, 1996

[12] I. Kováčová, L. Madarász, D. Kováč, J. Vojtko: "Neural Network Linearization of Pressure Force Sensor Transfer Characteristics". Proc. of the 8th International Conference on Intelligent Engineering Systems 2004 (INES'04), September 19-21, 2004, Technical University of Cluj-Napoca Romania, pp. 79-82, ISBN 973-662-120-0 (2004)

[13] Jang, J. S., Self-Learning Fuzzy Controllers Based on Temporal Back Propagation, IEEE Trans. on Neural Networks, Vol. 3, No. 5, (1992)

[14] Jang, J. S. and Sun, C., Neuro-Fuzzy Modeling and Control, Proc. of IEEE, Vol. 83, No. 3, 378–406, (1995)

[15] Jang, J. S. and Sun, C., Neuro-Fuzzy Modeling and Control, Proc. of IEEE, Vol. 83, No. 3, 378–406, (1995)

[16] Fuzzy Logic Toolbox User's Guide, MathWorks, Inc., (1995)

[17] Moody, J. and Darken, C., Fast Learning in Networks of Locally-Tuned Processing Units, Neural Computation, Vol. 1, 281−294, 1989

[18] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability and convergence in a multidimensional complex space," IEEE Tran. Evolutionary Computation, vol. 6, no. 1, pp. 58-73, February 2002

[19] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in Proceedings of the 1999 Congr. Evolutionary Computation, IEEE Service Center, Piscataway, NJ, 1999, pp. 1945-1950

[20] J. Vaščák, L. Madarász: "Similarity Relations in Diagnosis Fuzzy Systems", Journal of Advanced Computational Intelligence, Vol. 4, Fuji Press, Japan, ISSN 1343-0130, pp. 246-250, (2000)