Manufacturing Systems Modeling Using Neural Networks

George A. Rovithakis, Vassilis I. Gaganis, Stelios E. Perrakis and Manolis A. Christodoulou

Department of Electrical & Computer Engineering

Technical University of Crete

73100 Chania, Crete

GREECE

Abstract: In this paper a neural network approach to the factory dynamics modelling problem is discussed. A recurrent high-order neural network structure (RHONN) is employed to identify the manufacturing cell dynamics, which is supposed to be unknown. The model is constructed in such a way that enables the design of a controller which will force the model and thus the original cell to display the required behaviour. Buffer states as well as control input signals are transformed into continuous ones so as to be conformant with the RHONN assumptions. A case study demonstrates the approximation capabilities of the proposed architecture.

George A. Rovithakis was born in Chania, Crete, Greece in 1967. He received the diploma in Electrical Engineering from the Aristotelian University of Thessaloniki, Thessaloniki, Greece in 1990 and the MSc. and Ph.D degrees in Electronic and Computer Engineering from the Technical University of Crete, in 1994 and 1995 respectively.

He served as a Teaching and Research Assistant at the Laboratory of Automation, Department of Electronic and Computer Engineering, Technical University of Crete. Currently he is there as a visiting Assistant Professor.

His research interests include Factory Automation, Nonlinear Systems, Robust Adaptive Control, Intelligent Control and Control of Unknown Systems using Neural Networks, where he has authored or co-authored over 40 publications in scientific journals, refereed conference proceedings and book chapters.

He is a member of the IEEE and of the Technical Chamber of Greece.

Vassilis I. Gaganis was born in Chania, Greece, in 1972. He received his diploma in Mechanical Engineering from the University of Patras, Greece, in 1995. Currently he works on obtaining the MSc. degree in Electronics and Computer Science at the Technical University of Crete, Greece.

His current research interests are in dynamic neural networks, adaptive non-linear control and real-time manufacturing systems scheduling.

Stelios E. Perrakis was born in Chania, Greece, in 1972. He received his BSc. degree in Computer Science from the University of Crete, Greece, in 1995. Currently he works on obtaining the M.Sc. degree in Electronics and Computer Science at the Technical University of Crete.

His current research interests are in dynamic neural networks and computer implementations for real-time manufacturing systems scheduling.

Manolis A. Christodoulou was born in Kifissia, Greece in 1955. He received the diploma from National Technical University of Athens, Greece, the MSc. degree from University of Maryland, the Degree of Engineer from USC and the Ph. D from Democritus University of Thrace, Greece.

After being at University of Patras, he joined the Technical University of Crete, Greece in 1988, where he is currently a Professor of Control and Director of the Laboratory of Automation. He spent two sabbatical years (1987-1989) at Syracuse University, Syracuse, NY.

Professor Christodoulou has authored and co-authored over 120 journal articles, book chapters and conference publications in the areas of control theory and applications in Robotics. Factory Automation and Computer Integrated Manufacturing in Engineering and recently in neural networks for dynamic system identification and control. He is the organiser of various conferences and sessions of IEEE and IFAC and guest editor of various special issues of international journals. Professor Christodoulou is managing and co-operating on various research projects in Greece, within the European Union and in collaboration with the USA. He bears membership to IEEE and the Technical Chamber of Greece. He has been active in the IEEE Control Systems Society as the Chairman of the IEEE CS Greek Chapter and as the founder of the IEEE Mediterranean Symposium on new directions in Control & Automation.

1 Introduction

The manufacturing cell dynamics modeling problem can be stated as follows :

Factory Dynamics Modeling Problem: Find a mathematical model that will describe the unknown factory dynamics with a prescribed degree of accuracy; in other words the error between the actual dynamics and that of the proposed model should lie within a "small" neighborhood of zero. The dimension of the above mentioned neighborhood is a design requirement which will strongly affect the behavior of a future developed control law.

Models for factory dynamics usually fall under one of the categories below

- Discrete Event Dynamical Systems (DEDS)
- Discrete Time Dynamical Systems (DTDS)

The basic property of the DEDS model is that the events occur at discrete but unknown times. The above which is very useful in describing some processes, cannot take into consideration the so called "fast dynamics", which is the system dynamics when viewed microscopically, and cannot be excluded since it affects directly the quality of the output product or sometimes leads to non-optimal energy consumption, just to name a few. On the other hand, DTDS [1] evolve at fixed discrete times with sampling period T. The "fast dynamics" can be more easily introduced, and the connection to the existing control theory becomes more obvious. However, such approaches suffer from the "dimensionality" drawback, that is a huge number of state variables is required even for a small manufacturing cell.

From the above discussion it becomes apparent that a more generalized model (GM) is needed in order to better analyze different kinds of processes that exist in the real world. On line learning is the main property these GM's should exhibit in order to become more efficient and to provide high degree of autonomy, where what we mean by autonomy is "the power and ability for self governance in the performance of control functions" [2]. The above considerations lead to more complicated models which are more difficult to analyze, but they are neat and uniform and possibly describe the factory dynamics with minimal error.

Since we are concerned with the problem of regulating manufacturing cells e.g forcing the output of such a dynamical system to reach some desired constant value [3], and assume that we have no *a priori* information about the dynamics governing the cell, all existing techniques do not apply. Hence, the objective will be to approximate the unknown nonlinear dynamical system by neural networks [4][5][6][7] and once a model is obtained, to use it, to develop adaptive laws for on line adjustment of the weights of these networks, such that the stability of the overall system is guaranteed and furthermore, the control objective is asymptotically achieved. Therefore, in this way the problem is transformed into a nonlinear adaptive control problem, where the uncertainty in the system is due to some unknown parameters and to the existence of a modeling error term, which always appears in such identification procedures.

In order that a neural network architecture is able to approximate in some sense the behavior of a dynamical system such as a manufacturing cell, it is clear that it should contain some form of dynamics, or put differently, feedback connections [8]. In the neural network literature, such networks are known as recurrent neural networks [9], originally designed for pattern recognition applications. A special category of recurrent neural networks, namely Recurrent High Order Neural Networks (RHONNs), possess a linear in the weights property, thus making the issues of proving stability and convergence feasible and their incorporation into a control loop promising. A mathematical analysis of the approximation capabilities of a generalized type of RHONNs is presented in [10].

This paper focusses on the state-space approach to continuous-time recurrent (dynamic) neural networks for the purpose of nonlinear control and identification. The inherent dynamics of recurrent networks seems perfectly suited for control problems, although involving fairly complex analysis [11].

The remaining of the paper is organized as follows: continuous state and control input signals are defined in Section 2, while the proposed model is presented in Section 3. In Section 4 adaptive laws are derived, and stability and convergence properties are explored and established. Finally, in Section 5 an implementation example is given and results are discussed.

2 Continuous Signals Definitions

RHONNs structure requires that time evolves continuously. Additionally, state and control input signals are continuous functions of time. However manufacturing cell dynamics is commonly described by models which are based on the assumption that time evolves in steps (DTDS or DEDS), causing any event such as a machine start or an object production be brought at integer numbers of time steps. Machine commands are usually encoded as series of ones and zeros (where one may be interpreted as a command to start operation), and moreover, buffer states are a subset of the integers.

In order to have a successful relation between these two different strategies, we assume that an equivalent machine-operation frequency is used as control input to the system. This equivalent frequency is defined as the inverse of the time between two successive machine-starts. As shown in Figure 1, we can translate the absolute times in a piece-wise constant function of the frequency, and obtain an u(t) diagram.

Obviously, using this definition the frequency ranges between zero and a fixed finite value. The lower bound is equal to zero, which corresponds to an infinite period of time e.g the time until the machine operates again is infinite and thus it will never work again. By forcing the controller to send a zero-value frequency input to a machine, we can model a machine breakdown, or express the fact that a machine has reached the production requirements, and thus completed the prespecified work.

The upper bound corresponds to a minimum period of time. This can be measured by assuming that the machine works continuously. In such a case the period of time is equal to the machine production time, and the frequency is equal to its inverse. However, since there is always some idle time between two consecutive part productions, the frequency input is never expected to reach this upper bound.

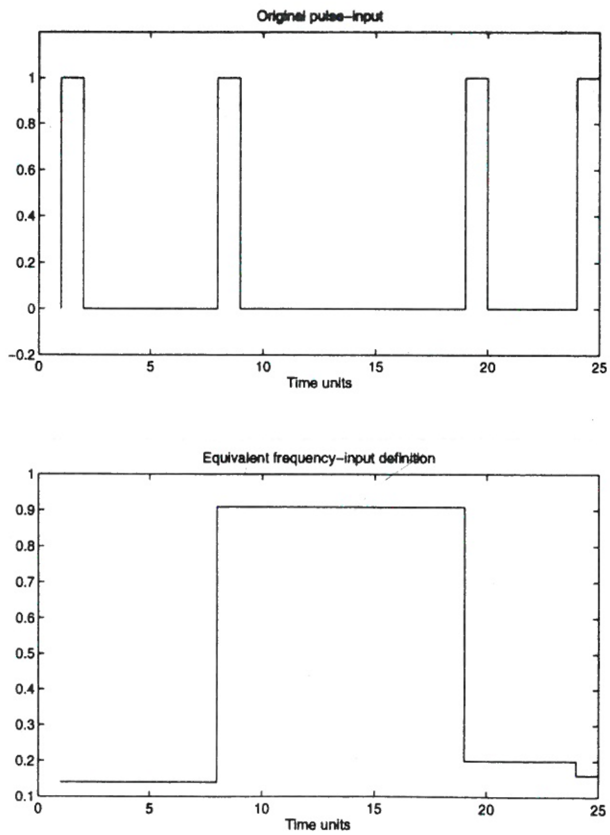

Figure 1: Equivalent Frequency Control Input Definition

If we define the states to represent buffer levels, they will be discontinuous functions expected to vary by one unit each time a new product is accumulated or taken out respectively, from the corresponding buffer. However, if we consider any part as a fluid which is to be processed by a machine at a constant rate, then the corresponding input (output) buffers will decrease (increase) linearly, as shown in Figure 2.

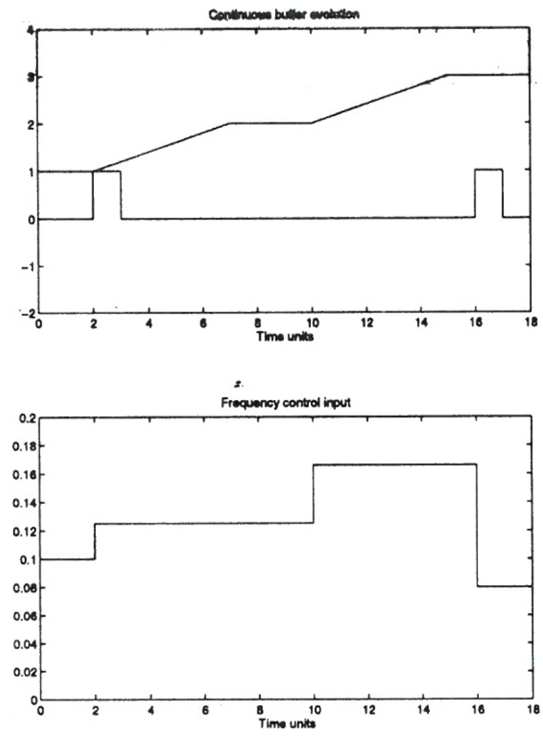

Figure 2: Continuous Buffer State Definition

The slope of this decrease is such that the buffer reaches its next integer level at a time interval equal to the one required by the machine to complete the specific operation. Such an assumption is implementable, since machine-starts can be on line detected and operation times are supposed to be known.

Although, using the above definitions time discontinuities seem to smooth out, control input is still a discontinuous function of time, since frequencies are piecewise constant. It is clear that such discontinuities render both input and state signals highly nonlinear, and thus very difficult for the neural network to learn. This problem can be overcome by applying some implementable smoothing functions, provided that there is an 1-1 mapping between the original signal and the smoothed one.

Since the control signal is piecewise continuous, changing from value x_a to a value x_b, a smart way to smooth values is to use two consecutive parabolas, one ranging between x_a and $\frac{x_a+a_b}{2}$, while the other ranges between $\frac{x_a+a_b}{2}$ and x_b. Parabolas derivatives should be zero at x_a and x_b respectively, while at their intersection they share a common derivative value. The unknown coefficients can easily be calculated and produce the required shape . Although such a smoothing function produces a delay so as to be implementable (the smoothed value is based on current or previous values only), it can be adjusted to be "narrow" enough so that the deviation from the original signal is neglectable. A sigmoid function ranging between x_a and x_b, with a steep slope could be used instead. However, such a technique suffers from derivative discontinuities at common points between the original signal and the smoothing sigmoidal function. Both situations are shown in Figure 3.

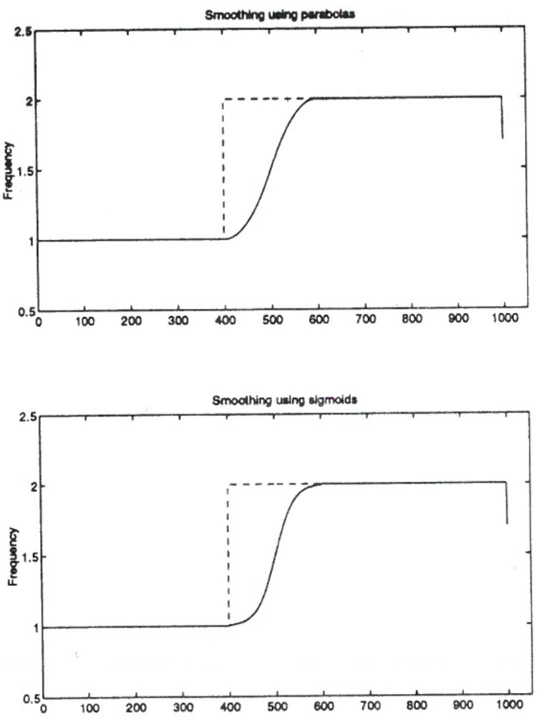

Figure 3: Frequency Control Input Smoothing

3 The Manufacturing Cell Dynamic Model

Consider the sample configuration shown in Figure 4, and assume that only machine A is working.

Since the buffer level increase rate is proportional to the input machine frequency, any change due to work done by the preceding machine A is governed by the equation

$$x_3 = f(.)u_A \tag{3.1}$$

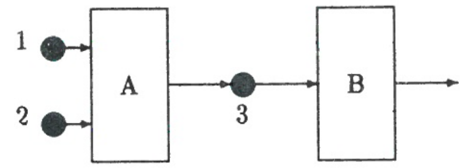

Figure 4: Sample Configuration

The unknown $f(.)$ function depends on buffer x_3. However, to relax a future control law we can add some restrictions to the state equations. Such a restriction is that buffer 3 should remain unchanged in case either buffer 1 or buffer 2 are empty. This can be accomplished by assuming that $f(.)$ is written as the product

$$f(.) = f_1(x_1)f_2(x_2)f_3(x_3) \tag{3.2}$$

where $f_1(.)$ and $f_2(.)$ are functions which cross the origin, that is $f_1(0) = 0$ and $f_2(0) = 0$. Obviously if either of x_1 or x_2 are equal to zero, the whole derivative term also becomes equal to zero . A possible selection for the $f_1(.)$ and $f_2(.)$ functions could be the sigmoids which cross zero, defined by

$$s_c(x) = \frac{k}{1+e^{-lx}} + \lambda$$

where

$$\lambda = -\frac{k}{2}$$

Due to the well known approximation capabilities of static High Order Neural Networks (HONNs) [12] we can assume that the unknown function $f_3(x_3)$ can be approximated to any desired degree of accuracy as the weighted sum of some high order sigmoids of x_i, that is

$$\forall \epsilon > 0 \;\; \exists k, \mathcal{L} = \{d_1, d_2, ..., d_k\} :$$

$$\forall t \left| f_3(x_3(t)) - \sum_{n=1}^{k} w_n s^{d_n}(x_3(t)) \right| \leq \epsilon \tag{3.3}$$

where $d_i \in \mathcal{N}$, $s(x)$ the sigmoid function and w_i are the unknown adjustable weights to be determined. Under this assumption the unknown function equivalent form becomes

$$f_3(x_3) = \sum_{i=1}^{k} w_i s^{d_i}(x_3) = W_1^T \bar{S}_1(x_3) \tag{3.4}$$

There is no way to determine either integer k or the set $\mathcal{L}$. However practical applications have shown that the larger the number of high order terms, the smallest the approximation error obtained.

Now assume that machine B is the only to work. The decrease of buffer x_3 is due to control input u_B, and thus can be described by

$$x_3 = f(u_B) \qquad (3.5)$$

Once more, HONN approximation capabilities are employed and the unknown function is approximated by

$$\dot{x}_3 = W_2^T S_2(u_B) \qquad (3.6)$$

where vector W_2 also contains unknown adjustable weights.

Consider the general case where a buffer x_i is connected to machines u_{in} and u_{out}, and assume that machine u_{in} operation depends on the preceding buffers $x_{i_1}, x_{i_2}, ..., x_{i_p}$. Based on the above observations the generalized dynamic equation becomes

$$\dot{x}_i = W_1^{T^*}\left(\prod_{j=1}^{p} s_c(x_{i_j})\right)\bar{S}_1(x_i)u_{in} + W_2^{T^*}S_2(u_{out})$$

or in a more compact form

$$\dot{x}_i = W_1^{T^*}S_1(\bar{x})u_{in} + W_2^{T^*}S_2(u_{out}) \qquad (3.7)$$

Notice that the dynamic equation of a buffer should be designed in such a way that in case no commands are sent, the $\dot{x}$ term should vanish, that is the buffer level should remain unchanged. This includes the possibility of a machine breakdown. Such a property is inherent in the above model, since no machine-start commands denote a zero-frequency, and obviously the $\dot{x}_i$ term vanishes.

The most interesting and promising part of this approach is that it leads to the construction of an adaptive system, that is self adjusted in any change to the cell desired performance. Moreover, it can easily take into account the fact that there is possibly a machine breakdown. It is obvious that a controller based on the above model needs no redesign whenever system behaviour changes, and thus can be implemented in real time situations.

The proposed model not only satisfies the above requirements, but also can be extended to including as many high order terms as required for approximating the manufacturing cell dynamics to any degree of accuracy.

4 Identifier Structure and Adaptive Laws

In order to identify the above model, an identifier of a similar structure is required. The identifier selected is governed by the equation

$$\dot{\hat{x}} = \hat{W}_1^T S_1(\bar{x})u_{in} + \hat{W}_2^T S_2(u_{out}) + a_m\hat{x} - a_m x \qquad (4.1)$$

where a_m is a strictly negative design constant, which directly affects the convergence speed as to be shown. $\hat{W}_1$ and $\hat{W}_2$ are the estimates of weight vectors W_1^* and W_2^* respectively.

Observe that the sigmoid functions vector $S_1(x)$ contains the unknown plant states instead of those of the identifier, thus rendering the identifier model a series-parallel one. If some stable adaptive laws can be derived so as to force identification error $e = \hat{x} - x$ go to zero, the last two terms of the identifier equation vanish, thus making the identifier equivalent to the original manufacturing cell model. The fact that a sufficient set of high-order connections has been found, is assumed, so as to consider the zero modelling error case.

To derive weight adjusting adaptive laws the Lyapunov synthesis method is employed [13]. Define the identification error as

$$e = \hat{x} - x \qquad (4.2)$$

and subtract equations () and () to obtain the error equation

$$\dot{e} = a_m e + \tilde{W}_1^T S_1(\bar{x})u_{in} + \tilde{W}_2^T S_2(u_{out}) \qquad (4.3)$$

where

$$\tilde{W} = \hat{W} - W^*$$

and subscript i has been omitted for the sake of simplicity.

Next, consider the Lyapunov function candidate

$$V = \frac{1}{2}e^2 + \frac{1}{2\gamma_1}tr\{\tilde{W}_1\tilde{W}_1^T\} + \frac{1}{2\gamma_2}tr\{\tilde{W}_2\tilde{W}_2^T\} \qquad (4.4)$$

where γ_1 and γ_2 are positive gain constants.

By differentiating with respect to time, along the trajectories of the error equation, we obtain the Lyapunov derivative function

$$\dot{V} = e\dot{e} + \frac{1}{\gamma_1}tr\{\dot{\tilde{W}}_1\tilde{W}_1^T\} + \frac{1}{\gamma_2}tr\{\dot{\tilde{W}}_2\tilde{W}_2^T\}$$

which, after substituting the error derivative can be written as

$$\dot{V} = a_m e^2 + e\tilde{W}_1^T S_1(\bar{x})u_{in} + e\tilde{W}_2^T S_2(u_{out})$$

$$+ \frac{1}{\gamma_1} tr\{\dot{\tilde{W}}_1 \tilde{W}_1^T\} + \frac{1}{\gamma_2} tr\{\dot{\tilde{W}}_2 \tilde{W}_2^T\} \qquad (4.5)$$

By splitting products of vectors into elements the derivative of the Lyapunov candidate can be written in the following form

$$\dot{V} = a_m e^2 + eu_{in} \sum_j s_{1j}(\bar{x})\tilde{w}_{1j} + e \sum_j s_{2j}(u_{out})\tilde{w}_{2j}$$

$$+ \sum_j \dot{\tilde{w}}_{1_j} \tilde{w}_{1_j} + \sum_j \dot{\tilde{w}}_{2_j} \tilde{w}_{2_j} \qquad (4.6)$$

Obviously, by selecting the update laws to be

$$\dot{\tilde{w}}_{1_j} = \dot{w}_{1_j} = -\gamma_1 e s_{1_j}(\bar{x}) u_{in} \qquad (4.7)$$

and

$$\dot{\tilde{w}}_{2_j} = \dot{w}_{2_j} = -\gamma_2 e s_{2_j}(u_{out}) \qquad (4.8)$$

the derivative of the Lyapunov function becomes

$$\dot{V} = a_m e^2 \leq 0$$

which holds since a_m is strictly negative. Moreover, the Lyapunov function is chosen to be a quadratic one, and thus positive definite and descrescent. Hence, according to Lyapunov's theorem, $e = 0, \tilde{W}_1 = 0, \tilde{W}_2 = 0$ is an equilibrium point.

To proceed on the Lyapunov function derivative is integrated over the time domain

$$\int_0^\infty \dot{V} dt = V(\infty) - V(0) = a_m \int_0^\infty e^2 dt \quad (4.9)$$

Since the Lyapunov function is non-negative and decreasing, it is also bounded, e.g.

$$V(0) \geq V(t) \geq 0 \Rightarrow V(0) - V(\infty) < \infty$$

$$\Rightarrow V(\infty) - V(0) \in \mathcal{L}_\infty$$

The above result combined with (.) leads to $e \in \mathcal{L}_2$. Moreover, since the Frobenius forms $tr\{\tilde{W}_1 \tilde{W}_1^T\}$ and $tr\{\tilde{W}_2 \tilde{W}_2^T\}$ are non-negative

$$V \in \mathcal{L}_\infty \Rightarrow e^2 \in \mathcal{L}_\infty \Rightarrow e \in \mathcal{L}_\infty$$

and consequently, using the identification error definition (.) we conclude that $\hat{x} \in \mathcal{L}_\infty$, since buffer levels x_i are supposed to be bounded. Finally, from the error derivative equation (.) we obtain $\dot{e} \in \mathcal{L}_\infty$, since all quantities in the right-hand are also bounded. Now since $e, \dot{e} \in \mathcal{L}_\infty$ and $e \in \mathcal{L}_2$ applying Barbalat's Lemma [14] we obtain that

$$\lim_{t \to \infty} e(t) = 0 \qquad (4.10)$$

The boundness of the weight estimates can easily be proved since

$$V \in \mathcal{L}_\infty \Rightarrow tr\{\tilde{W}\tilde{W}^T\} \in \mathcal{L}_\infty \Rightarrow W \in \mathcal{L}_\infty$$

However, although the control input is bounded and $e \to 0$ which means that

$$\dot{\tilde{w}}_{1j} = \dot{\hat{w}}_{1j} = -\gamma e s(\bar{x}) u \Rightarrow \dot{\hat{w}}_{1j} \in \mathcal{L}_\infty$$

$$\dot{\tilde{w}}_{2j} = \dot{\hat{w}}_{2j} = -\gamma e s(u) \Rightarrow \dot{\hat{w}}_{2j} \in \mathcal{L}_\infty$$

we cannot guarantee that $\hat{w}$ also converges to some value.

The above results hold under the assumption that the appropriate integer k and a set $\mathcal{L}$ have been found. However since such information cannot be known *a priori* and some other used instead, the error is expected to approach zero as close as possible. The error magnitude upper bound depends on the similarity of the used parameters k and $\mathcal{L}$ to the original ones. Similarly weight estimates are expected to vary within a region instead of converging to specific values. However, since the control of the manufacturing cell is the main objective, there is no need for the network weights to stabilize at certain constant values, even some different from the optimal ones, which are unknown. Instead, the error minimization is required. That means, that the identification phase is used for two main purposes. The first one concerns whether there are enough neuron connections to allow the error to be minimized within a desired degree of accuracy, that is, check if the network structure renders it capable of approximating the manufacturing cell. The second purpose is to find, if possible, some good initial weight values which, although not optimal, will be used through the control phase.

5 Implementation

5.1 Cell Configuration

The above developed model has been implemented in a case study, shown in Figure 5. In this certain example the candidate manufacturing cell consists of five machines denoted by letters A, B, C, D and E. 10 different products are used in this example, namely

- four raw materials denoted by r_1 up to r_4
- five semi-finished products numbered from 1 to 5
- one finished product, denoted as buffer 6.

In the following Table, the operation time required by each machine is presented. Observe

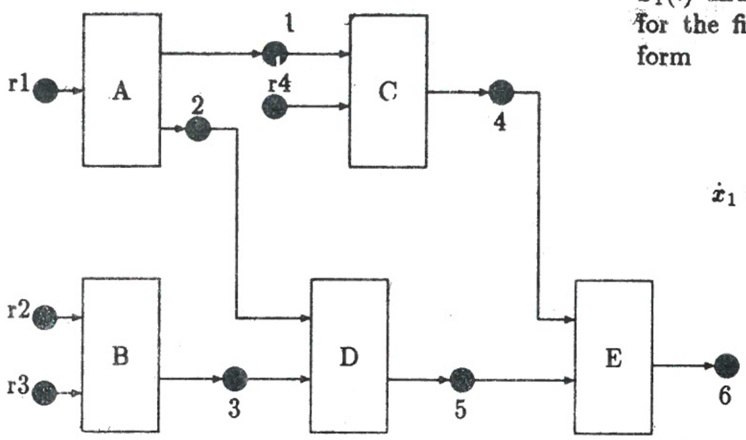

Figure 5: The Case Study

that the selected times do not correspond to a real plant, since they lead to product accumulation in some semi-finished product buffers. However, such a behaviour is recommended, since it forces the neural network identifier to learn a behaviour, more complex than a simple signal which toggles between one and zero.

Table 1: Production Times

Machine	Time required per part production
1	5
2	6
3	5
4	4
5	3

Each product corresponds to a specific buffer. Under this assumption there are 10 buffers. Note that the raw-material buffers are assumed to be of infinite capacity, and thus their dynamics is ignored, and there are no dynamical equations for them.

5.2 Dynamical Equations Construction

In the following table the manufacturing interconnection is presented. Based on that information the dynamic equations are constructed.

Table 2: Manufacturing Cell Interconnection Data

Buffer No.	Preceding machine	Succeeding machine	Feeding buffers
1	A	C	r1
2	A	D	r1
3	B	D	r2, r3
4	C	E	r4, 1
5	D	E	2, 3
6	E		4, 5

If four high order terms are to be included in $S_1(.)$ and $S_2(.)$ vectors, the dynamic equation for the first buffer is written in the following form

$$\dot{x}_1 = \begin{Bmatrix} w_{1_1}^* \\ w_{1_2}^* \\ w_{1_3}^* \\ w_{1_4}^* \end{Bmatrix}^T \begin{Bmatrix} s(x_1) \\ s^2(x_1) \\ s^3(x_1) \\ s^4(x_1) \end{Bmatrix} u_A + \begin{Bmatrix} w_{1_5}^* \\ w_{1_6}^* \\ w_{1_7}^* \\ w_{1_8}^* \end{Bmatrix}^T \begin{Bmatrix} s_c(u_C) \\ s_c^2(u_C) \\ s_c^3(u_C) \\ s_c^4(u_C) \end{Bmatrix} \quad (5.1)$$

where the c subscript denotes the use of centered sigmoid functions.

The identifier equation for the same buffer is

$$\dot{\hat{x}}_1 = \begin{Bmatrix} \hat{w}_{11} \\ \hat{w}_{12} \\ \hat{w}_{13} \\ \hat{w}_{14} \end{Bmatrix}^T \begin{Bmatrix} s(x_1) \\ s^2(x_1) \\ s^3(x_1) \\ s^4(x_1) \end{Bmatrix} u_A + \begin{Bmatrix} \hat{w}_{15} \\ \hat{w}_{16} \\ \hat{w}_{17} \\ \hat{w}_{18} \end{Bmatrix}^T \begin{Bmatrix} s_c(u_C) \\ s_c^2(u_C) \\ s_c^3(u_C) \\ s_c^4(u_C) \end{Bmatrix} + a_m \hat{x}_1 - a_m x_1 \quad (5.2)$$

Dynamic equations for the rest buffers are constructed in a similar way. All non-centered sigmoids share the same parameters $k = 3, l = 1, \lambda = 0$, while the centered sigmoids parameters are set to $k = 5, l = 1, \lambda = -2.5$. The a_m constants have been set equal to -50, and the adaptive laws gains have been set to the value of 25.

5.3 Results

In the following Figure, the feasible but randomly selected machine-command schedule is presented. Each line represents the schedule for the corresponding machine, where the raised values denote a machine-start command. The corresponding frequencies have been calculated and presented in Figure 7. Observe the last frequency values, which are not equal to zero, since the present schedule is only a small part of the original one used for the network training.

Buffers states evolution as well as the identifier states with the error term multiplied by a factor of 500, are presented in Figure 8. These states were calculated by our own simulator, and correspond to the control input schedule defined above. Observe that products are accumulated in buffers 3 and 6, thus making the signal to be learned more complex.

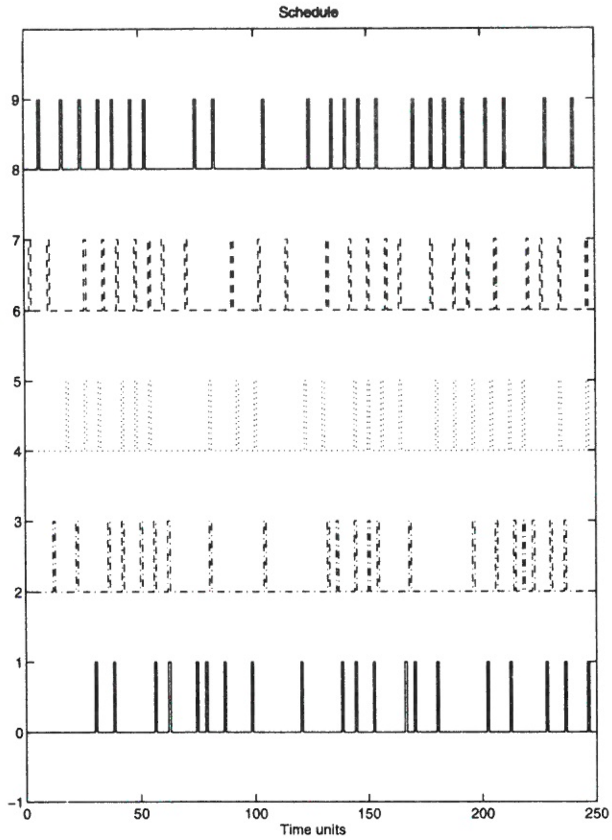

Figure 6: Machine-Start Control Input Schedule

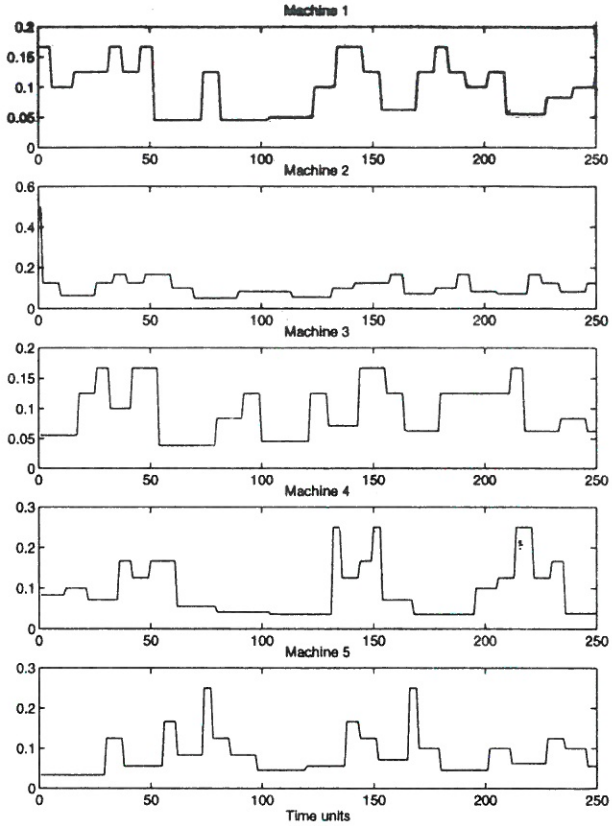

Figure 7: Frequency Control Input Schedule

It is obvious that buffers state identification error is bounded by a constant value envelope, from both top and below. This value in any case did not exceed **0.0005**, although only 8 weights per equation were used.

The error magnitude is not expected to reach zero, even if a large-time schedule is used as training function. This is due to the capabilities of this certain high order connection selected. Using the derived adaptive laws, the error reaches the minimum value it can get. In order to increase the accuracy of the model, higher order terms should be introduced in the dynamical equations. However, such an extension is not always desired, since it always leads to an error reduction, but also excites an oscillatory behaviour when used at a very large scale.

6 Conclusions

A neural network identifier structure for identifying manufacturing cell dynamics is presented. The proposed equations are non-linear, but still possess a linear-in-the-weights property. Hence stable adaptive laws can be derived, which guarantee convergence and stability properties. The emerging non-linearities do not seem to be a problem since the controller, which is to be designed later, does not require that the model is linear. Instead, a non-linear controller is expected to be more flexible and capable of absorbing any possible disturbances.

The results in Figures 8 and 9 show that the proposed model seems to be satisfactory enough to describe a complex dynamic system, such as manufacturing cells. Finally, the current model presents two important properties. First, it can always be improved by adding new high order terms, in order to increase accuracy. Second, it is self adjusted in environmental changes, that is, not only does it need no training whenever a specific machine is no more capable of producing new parts, but also in case the production requirements are altered. These two properties are of great importance, since the whole idea is based on the assumption that the control policy will be decided on-line.

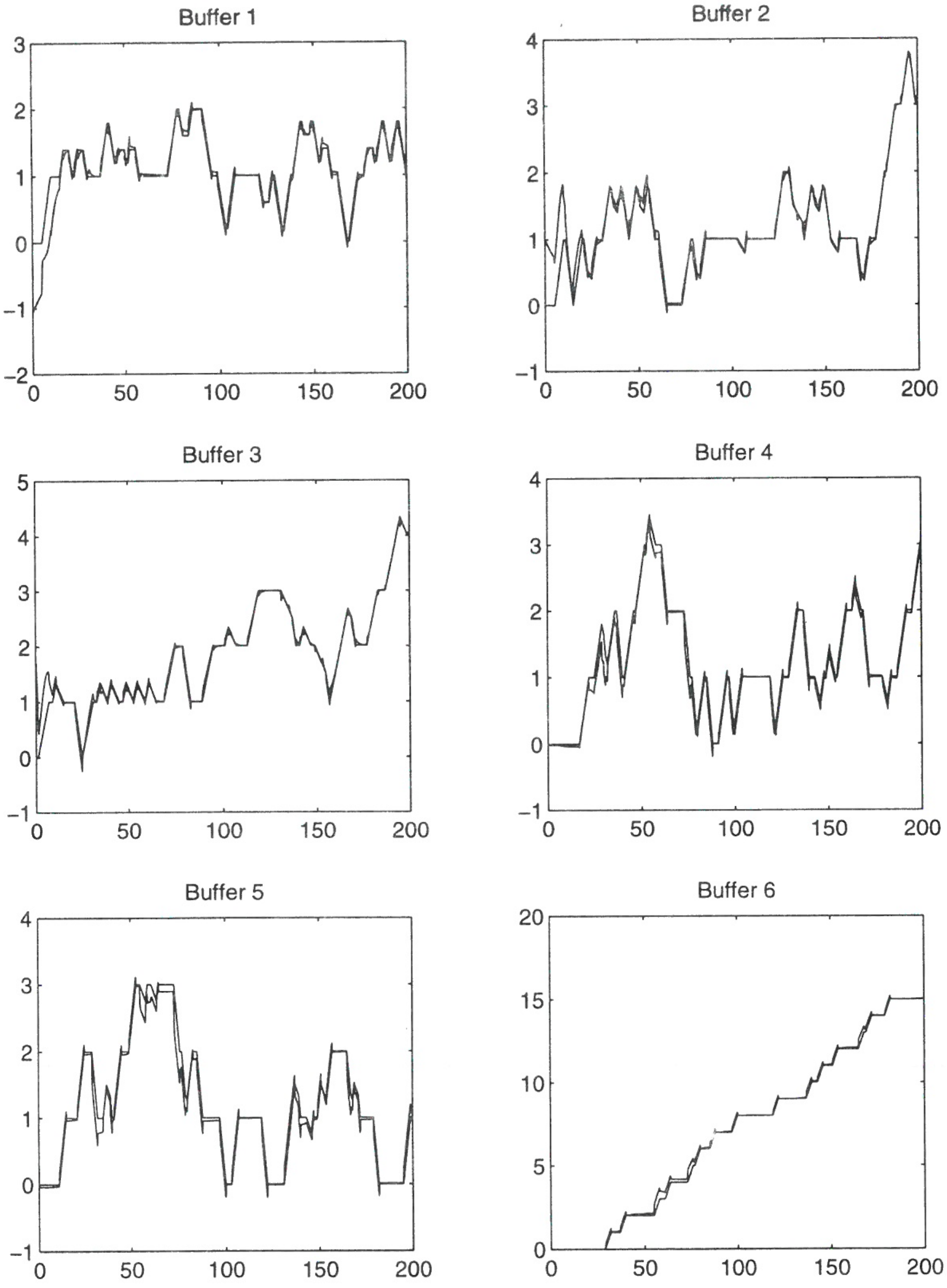

Figure 8: Original and Identified Signals: Errors Multiplied By 500

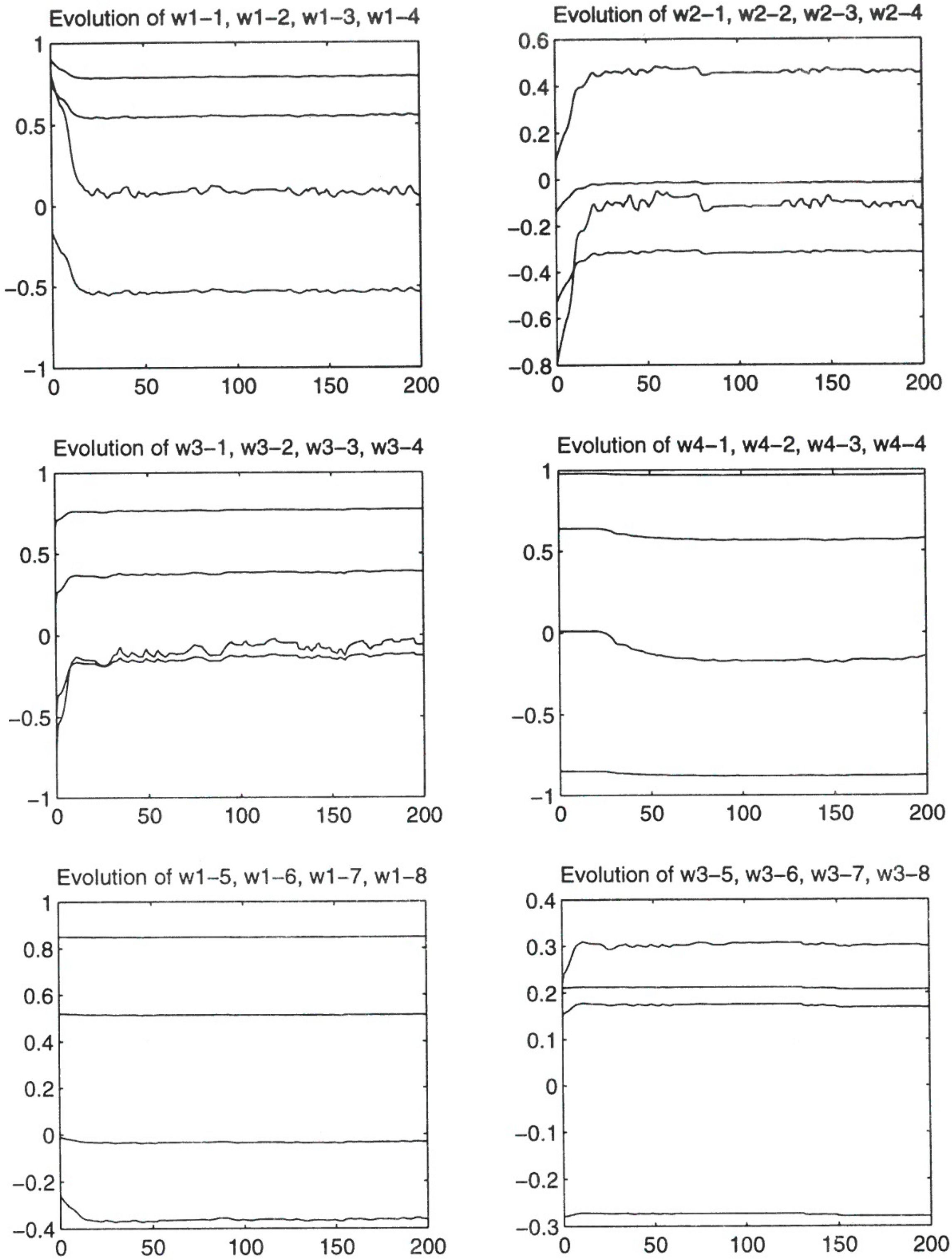

Figure 9: Weights Evolution

REFERENCES

[1] KOSMATOPOULOS, E.B. and CHRISTODOULOU, M.A., **A New Model Used for Suboptimal Solutions in Control of Manufacturing Processes**, supported by the HIMAC-ESPRIT project 8141.

[2] ANTSAKLIS, P.J., PASSINO, K.M. and WANG, S.J., **An Introduction To Autonomous Control Systems**, IEEE CONTROL SYSTEMS MAGAZINE, Vol. 11, No. 4, 1991, pp. 5-13.

[3] TAYLOR, D.G., KOKOTOVIC, P.V., MARINO, R. and KANELLAKOPOULOS, I., **Adaptive Regulation of Nonlinear Systems With Unmodeled Dynamics**, IEEE TRANS. AUTOMAT. CONTR., Vol. 34, No. 4, 1989, pp. 405-412.

[4] NARENDRA, K.S.,**Stable Identification Schemes in System Identification : Advances and Case Studies**, ACADEMIC, New York, 1976.

[5] IOANNOU, P.A. and KOKOTOVIC, P.V., **An Asymptotic Error Analysis of Identifiers and Adaptive Observers in the Presence of Parasitics**, IEEE TRANS. AUTOMAT. CONTR., Vol. 27, No. 4, 1982, pp. 921-927.

[6] CHEN, S., BILLINGS, S.A. and GRANT, P.M., **Non-linear System Identification Using Neural Networks**, INTERNATIONAL JOURNAL OF CONTROL, Vol. 51, No. 6, 1990, pp. 1191-1214.

[7] POLYCARPOU, M.M. and IOANNOU, P.A., **Identification and Control of Nonlinear Systems Using Neural Network Models: Design and Stability Analysis**, Tech. Rep. 91-09-01, Dept. Elec. Eng.-Systems, University of Southern California, Los Angeles, 1991.

[8] HERTZ, J., KROGH, A. and PALMER, R.G., **Introduction To the Theory of Neural Computation**, ADDISON-WESLEY PUBL. CO., 1991.

[9] HOPFIELD, J.J., **Neurons With Graded Response Have Collective Computational Properties Like Those of Two State Neurons**, Proc. Natl. Acad. Sci., Vol. 81, 1984, pp. 3088-3092.

[10] KOSMATOPOULOS, E.B., **Learning Via Recurrent High Order Neural Networks**, MSc. Dissertation, Electronic and Computer Engineering, Technical University of Crete, Greece, 1992.

[11] ROVITHAKIS, G. and CHRISTODOULOU, M., **Adaptive Control of Unknown Plants Using Dynamical Neural Networks**, IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS, Vol. 24, No. 3, March 1994.

[12] KOSMATOPOULOS, E., **Recurrent High Order Neural Networks for Learning in Dynamical Environments**, Ph.D Thesis, Department of Electronic and Computer Engineering, Technical University of Crete, 1995.

[13] IOANNOU, P. A. and SUN, J., **Robust Adaptive Control**, PRENTICE HALL, Upper Saddle River, NJ, 1996.

[14] ROUCHE, N., HABETS, P. and LALOY, M., **Stability Theory by Lyapunov Direct Method**, SPRINGER-VERLAG, New York, 1977.