1918
TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Neural Networks Based Identification and Control of Nonlinear Systems: ANARX Model Based Approach

EDUARD PETLENKOV

Tallinn 2007

Faculty of Information Technology
Department of Computer Control
TALLINN UNIVERSITY OF TECNOLOGY

Dissertation was accepted for the defence of the degree of Doctor of Philosophy in Engineering on October 17, 2007.

Supervisor: Ennu Rüstern, Prof., Ph.D., Department of Computer Control, Tallinn University of Technology

Opponents: Robert Tenno, Prof., D.Sc., Control Engineering Laboratory, Helsinki University of Technology

Tõnu Lehtla, Prof., Ph.D., Department of Electrical Drivers and Power Electronics, Tallinn University of Technology

Defence of the thesis: December 6, 2007

Declaration: Hereby I declare that this doctoral thesis, my original investigation and achievement, submitted for the doctoral degree at Tallinn University of Technology has not been submitted for any degree or examination.

Eduard Petlenkov

ISSN 1406-4731
ISBN 978-9985-59-734-7

Table of contents

Abstract

Neural Networks Based Identification and Control of Nonlinear Systems: ANARX Model Based Approach

During the last twenty years a lot of advances in control of complex nonlinear systems and process have been made. One of the most important reasons for that is development of various nonlinear system identification techniques based on input-output representation of the model such as training of artificial neural networks. While a lot of interesting results are achieved and practical applications are made in control system design using classical fully connected neural networks, structure of the model plays significant role in model based control of complex systems.

This thesis is devoted to neural networks based identification and model based control of nonlinear systems. Different structures of artificial neural networks are considered as approximation tools for identification of complex nonlinear systems and processes. It is shown in the research that proper application dependant architecture of the network can significantly improve quality of identification which has crucial importance for control system design. Moreover, specific structures of a neural network based model makes possible combination of classical control algorithms with neural network based identification and adaptation.

Additive Nonlinear AutoRegressive eXogenous (ANARX) model is considered as a reasonable choice for control-aimed modeling of a wide class of nonlinear systems because of its linearizability by dynamic feedback. Neural Network based ANARX (NN-ANARX) model is applied to control of nonlinear systems. An adaptive controller is designed by combining classical dynamic output feedback linearization with neural network based adaptation.

Multiply Input Multiply Output (MIMO) ANARX structure is proposed and applied to identification of nonlinear MIMO systems. NN-based Simplified ANARX (NN-SANARX) model class is introduced here as an alternative witch allows to simplify feedback computation bringing it to a solution of a system of linear equations. Linearization based control technique is applied to control of a large class of nonlinear MIMO systems by using NN-SANARX model.

NN-SANARX model based control imposes restrictions causing necessity to redesign adaptation technique based on training of the neural network representing the model. An adaptive controller for nonlinear MIMO systems identified by NN-SANARX structure is also designed in the thesis.

Kokkuvõte

Mittelineaarsete süsteemide identifitseerimine ja juhtimine tehisnärvivõrkudega: ANARX mudelil põhinev lähenemine

Viimaste aastakümnete vältel on tehisnärvivõrgud leidnud rakendust keerukate mittelineaarsete süsteemide juhtimisprobleemide lahendamisel. Üheks oluliseks juhtimisprobleemide lahendamise eelduseks on erinevate sisend-väljund mudelitel põhinevate mittelineaarsete süsteemide identifitseerimise meetodite väljatöötamine ja rakendamine. Vaatamata sellele, et on välja töötatud ja mittelineaarsete süsteemide juhtimisprobleemide lahendamisel rakendatud palju tehisnärvivõrkudel baseeruvaid meetodeid ei ole mudelite struktuurilisi võimalusi veel täiel määral kasutatud.

Käesolev väitekiri käsitleb tehisnärvivõrkudel põhinevat mittelineaarsete süsteemide identifitseerimist ja mudelil põhinevat juhtimist. Põhjalikult on analüüsitud erinevaid tehisnärvivõrkude arhitektuure keerukate mittelineaarsete süsteemide ja protsesside identifitseerimisprobleemide lahendamisel. Töös esitatud uuringutes on näidatud, et sobiva rakendusest sõltuva tehisnärvivõrgu arhitektuuri valikuga võib oluliselt tõsta identifitseerimise kvaliteeti, millel on kriitiline tähtsus süsteemide analüüsil ja juhtimissüsteemide projekteerimisel. Enamgi veel, spetsiifilised tehisnärvivõrkudel põhinevad mudeli struktuurid võimaldavad kombineerida klassikalisi juhtimisalgoritme tehisnärvivõrkudel põhinevate identifitseerimis- ja adapteerimisalgoritmidega.

Suurt tähelepanu on töös pööratud ANARX (Additive Nonlinear AutoRegressive eXogenous) mudelitele, mis on mõistlikuks valikuks suure mittelineaarsete süsteemide klassi identifitseerimiseks lahendamaks juhtimisülesandeid. Need mudelid on alati lineariseeritavad dünaamilise tagasiside abil. Tehisnärvivõrgudel põhinev ANARX (NN-ANARX) mudel on rakendatud mittelineaarsete süsteemide juhtimiseks. On väljatöötatud adaptiivne süsteem, mis kombineerib klassikalist lineariseerimist tehisnärvivõrkude treenimisel baseeruva adapteerimisega.

On väljatöötatud mitme sisendiga ja mitme väljundiga (MIMO) ANARX struktuur. See mudel on rakendatud mitmemõõtmeliste mittelineaarsete süsteemide identifitseerimisel. Samuti on väljatöötatud lihtsustatud NN-ANARX mudelite klass – NN-SANARX (NN-based Simplified ANARX), mis lihtsustab oluliselt lineaarse dünaamilise tagasiside arvutamist. Dünaamilisel lineariseerimisel baseeruv juhtimisalgoritm on rakendatud mittelineaarsete mitmemõõtmeliste süsteemide klassi juhtimiseks kasutades NN-SANARX mudelit. NN-

SANARX mudelil baseeruv juhtimine kehtestab lisatingimusi adapteerimis-algoritmile. Töös on väljatöötatud adaptiivsüsteem NN-SANARX mudeliga identifitseeritavate mitmemõõtmeliste mittelineaarsete süsteemide juhtimiseks.

Acknowledgements

Let me begin by expressing my sincere gratitude to my parents for their strong support, inspiration and belief in me. I am also very grateful to my wife and daughter for giving me energy and confidence in achieving my goals.

Very special thanks to my supervisor professor Ennu Rüstern who introduced me to the topics of automatic control. He provided me with strong support through my studies, necessary freedom for fulfillment of my scientific research interests and excellent working conditions.

I would like to thank Dr. Sven Nõmm for very fruitful scientific collaboration, interesting discussions and helpful advices. I am also grateful to him for proofreading of the draft of this thesis and pointing out some of my mistakes.

I am very grateful to my colleagues and co-authors from Institute of Cybernetics and Tokyo Denky University professor Ülle Kotta, professor Jüri Vain and professor Fujio Miyawaki for very useful discussions, advices and comments on my scientific research.

I would like to mention gratefully master student Juri Belikov who did a lot of work on the implementation of the algorithms discussed in the thesis and performed a huge number of experiments making possible conclusions presented here.

This work was partly supported by the Nations Support Program for the ICT in Higher Education "Tiger University", Estonian Doctoral School in Information and Communication Technologies and Estonian Science foundation under grants number 5170 and 6837.

Chapter 1

Introduction

This thesis summarizes research experience and the main results achieved by the author in the field of neural networks based system identification and model based control of nonlinear SISO and MIMO systems. The work is devoted to identification and control of nonlinear systems by means of different structures of neural networks based models. The main attention is paid to ANARX (Additive Nonlinear AutoRegressive eXogenous) structure of the model. Neural Network based Simplified ANARX (SANARX) model is introduced by the author for significant simplification and improvement of dynamic output feedback linearization based control of nonlinear SISO systems and application of this algorithm to control of a wide class of nonlinear MIMO systems.

The thesis considers

- analysis of three neuro models based control algorithms of different types, comparing them and defining their advantages and drawbacks;
- design and application of specific neural network structures improving the quality of model based neurocontrollers;
- application of dynamic output feedback linearization algorithm to control of nonlinear SISO and MIMO systems and design of adaptive controllers based on this algorithm.

The main original contributions of this thesis is in

- design of Neural Network based Hammerstein model and application of this model for significant improvement of model based predictive control of systems with static actuator nonlinearities;
- application of NN-ANARX model based dynamic output feedback linearization algorithm to control of nonlinear systems;
- definition of Neural Network based Simplified ANARX (NN-SANARX) model;

- development of a control technique based on dynamic output feedback linearization on NN-SANARX model and its application to control of nonlinear MIMO systems;
- definition of Neural Network based ANARX model for MIMO systems and its application to dynamic output feedback linearization based control of MIMO systems by using an additional neural network.
- design of adaptive controllers based on NN-ANARX and NN-SANARX models of controlled nonlinear SISO and MIMO systems.

Author’s contributions are discussed in more detail in the beginning of chapters 3, 4 and 5.

Significant attention is paid to identification of ANARX models of nonlinear systems and ANARX models based control because of their very important advantages over classical NARX models. ANARX is a sub-class of NARX models with separated time instances. Restrictions imposed by this sub-class guarantee linearizability by dynamic output feedback as well as state-space reprezentability of the model. These advantages are especially important for control applications. That is why this type of the model is a reasonable choice for control of a wide class of nonlinear systems.

The work is also inspired by neural networks ability to model the complex behavior of nonlinear systems from one side and their ability to reproduce an arbitrary structure of the system by choosing proper functions and connections between nodes from another side.

This thesis considers artificial neural network as an instrument for identification of nonlinear systems for model based control. Because of consisting of simple interconnected nodes (artificial neurons), neural network can represent any structure of the model corresponding to the requirements of the control algorithm. Proper structure of the model can be obtained by choosing connections between neurons (connecting them in the right way) and defining activation functions of each neuron in the network according to the needs of the control system.

1.1 State of the Art

Nonanalytical methods discussed in the thesis consider identified system as a “black box” and identify parameters of the model by using a set of data gathered from system’s input and output. When these methods are used, the structure of the model has to be chosen before starting the identification procedure or turned during it by a predefined algorithm. The structure of the model significantly depends on its application. For model based control considered in this work, it should satisfy the needs of the control algorithm.

The history of developing technical systems based on interconnection of nodes representing mathematical models of biological neurons takes its start from the year 1943 when McCulloch and Pitts proposed a mathematical model of the neurons [1]. This model is called an artificial neuron and is used in the most artificial neural networks based applications until nowadays. This model proposed almost 65 years ago is also a major basic element in systems discussed in this thesis.

Learning machine built by Edmonds and Minsky in 1951 can be considered as the first artificial neural network simulator. This neural network learning machine, called SNARC (Stochastic Neural-Analog Reinforcement Computer), was based on Hebb's ideas [48] replicating mathematically what happens when synaptic transmissions occur in the brain [2]. Nevertheless the real beginning of neural networks (NNs) and NN-based learning the invention of a simple neuron-like learning network by Rosenblatt [49] in 1962. This simplest layered fully connected neural network is called perceptron. Today multilayer perceptron is still the most popular and the most widespread neural network structure because of its very good and proofed [44] approximation capabilities.

It has to be mentioned that very little research was done in the area until about the 1980s mainly because of high computational complexity of training the networks that are capable of solving difficult problems. However, many of the artificial neural networks in use today are still based on the early advances of the McCulloch-Pitts neuron and the Rosenblatt perceptron. The majority of practical neural network based control applications utilize multilayer perceptron as the structure of the network. Numerous examples and research results can be found in literature demonstrating very good approximation, identification and adaptation abilities of this type of neural networks and their relevance to control systems design.

Majority of research is pointed to approximation capabilities of neural networks and application of this property in technical systems. At the same time significantly lower attention is paid to the structure of the neural network. During the last 20 years multilayer perceptron has shown its very good approximation capabilities and applicability for solving a lot of complex problems from very different fields and therefore it is too general to be the best in each particular application.

1.2 Author's point of view

While there are many ways to classify control algorithms in the framework of this thesis model based control algorithms can be divided into two main types: structure dependent and structure independent model based control algorithms.

Control algorithm is said to be structure independent if a controller is based on a model of the controlled system or process, but does not require any certain structure/architecture of the model. Any structure of the model can be utilized. The model used to estimate or/and predict the behavior of the controlled system or process. Model based predictive control algorithm is considered as an example of structure independent control algorithms. Inverse model based control also belongs to this class because the model (inverse model) is used to estimate the controlled process – to estimate the input which caused the known reaction.
Control algorithm is said to be structure dependent if a controller is based on a certain predefined structure/architecture of the model of the controlled system or process. Dynamic Output Feedback Linearization algorithm based on the model represented in the form of ANARX structure is considered as a structure dependent control algorithm.

Author's research has shown that properly chosen structure of the neural network based model can significantly improve the quality of control and makes possible combination of classical model based control approaches requiring certain representation of the model (for example, state-space representation) with NN-based identification and adaptation. Neural networks based architecture is adaptable in nature. Adaptation could be done by learning. Thus, robustness of classical model based control techniques can be significantly improved by introducing neural networks based models.

The quality of model based control algorithms significantly depends on the accuracy of the model. Besides ability to learn and adapt by learning neural networks based models makes possible to obtain a model of the desired structure.

Properly chosen neural network structure may increase identification quality and accuracy of the model thus increasing the quality of structure independent control algorithms. Varying the structure of the neural network it is possible to obtain application-specific models reproducing the structure of the controlled system for better identification and control quality as will be demonstrated in this thesis on example of model based predictive control of systems with actuator nonlinearities.

In case of structure dependent model based control algorithms using artificial neural networks for identification gives wonderful ability to choose the structure of the model and to obtain the model in the form relevant to the control algorithm very easily. By using this approach ANARX models of nonlinear systems are obtained by training the network of the corresponding structure and dynamic output feedback linearization algorithm based on this model is applied to control of nonlinear SISO and MIMO systems by the author of this thesis.

Neural networks based models of dynamic systems use external feedback from the output(s) of the model with delays or internal recurrent connections through delays and therefore these have discrete nature. All neural network based models of dynamic systems considered in this work are discrete-time models. Therefore only discrete-time control systems will be considered. The notations $x(t)$, $u(t)$ and $y(t)$ denote the value of a state, an input and an output at time step t correspondingly.

All the experiments discussed in the thesis are performed by simulating systems in MATLAB/SIMULINK environment.

1.3 Outline of the thesis

The thesis is organized as follows: chapter 2 gives an overview of mathematical tools used in the next parts of the thesis. It presents discrete-time models and control algorithms studied in the work, discusses basic principles and concepts of artificial neural networks. The main methods and theorems that are important for understanding the rest part of the thesis are also considered in the second chapter.

The third chapter is devoted to identification of nonlinear dynamic systems by artificial neural networks. Models based on recurrent and feedforward neural networks are considered. A novel NN-based Hammerstein model and its application to control of direct current servo motor with nonlinear driver is presented. NN-based ANARX model and its advantages over classical NARX model are also discussed and demonstrated on numerical examples. This chapter also presents NN-ANARX model for MIMO systems and discusses its application to identification of the surgeon's hand for the motion recognition.

The fourth chapter discusses control algorithms based on models obtained by training a neural network. Predictive control, inverse model based control and NN-ANARX model based control techniques are considered. They are analyzed and demonstrated on several numerical examples. Drawbacks and advantages of these methods are drawn in the end of the chapter.

The fifth chapter presents two novel methods for control of nonlinear MIMO systems. NN-based Simplified ANARX structure is introduced and a controller based on dynamic output feedback linearization of this model is designed. NN-ANARX model based control of nonlinear MIMO systems are demonstrated. This chapter also studies robustness of the proposed techniques and introduces NN-SANARX and NN-ANARX models based adaptive controllers capable of robust control of a wide class of nonlinear MIMO systems.

Conclusions summarizing the results of the thesis are drawn in the sixth chapter and subjects for further research and development are given in the last chapter.

Chapter 2

Overview of Mathematical Tools

An overview of basic instruments used in this thesis is given is this chapter. The main tools for nonlinear Single Input Single Output (SISO) systems modeling, analysis and control systems design are described here. The main attention is paid to artificial neural networks based models and methods as well as classical algorithms that can be combined with neural networks based approaches for achieving better quality of nonlinear control systems.

2.1 Nonlinear systems

The notion of a system can be defined as in [3].

A system is a combination of components that act together to perform a certain objective.

A system can be understood as an isolated part of the universe that is of interest to us and other parts of the universe that interact with the system compose the system environment , or neighboring system [3].

The majority of systems surrounding us are nonlinear. It means that the relations between these components have nonlinear nature. Using linearized models for control system design imposes strong restrictions to the range of set points that this system can reach and the range of reference signals that the control system can track. That is why control techniques based on nonlinear models that can better represent nonlinear relations between components are so important.

All existing systems change in time, and when the rates of change are significant, the systems are referred to as dynamic systems [3]. Practically it means that current behavior of a system depends on its previous behavior or in other words, it depends on the state of a system. The majority of systems are nonlinear dynamic systems.

2.2 Mathematical models of nonlinear dynamic systems

There exists the majority of techniques for mathematical modeling of nonlinear systems. The mathematical model has to describe the main features of the system. In case of models considered in the thesis this are features that are important for design of control systems.

Model of a system is a form of abstract descriptions of the relationships existing among system variables [3].

When these relationships are mathematically expressed by nonlinear functions the model is called nonlinear mathematical model.

There are continuous-time and discrete-time models exist. In case of continuous-time models output and inner states depend on time and can be calculated at each point of continuous time. In case of discrete-time models output and/or inner states can be calculated only in certain time instances. Period of time between these instances is called sample time. Mathematically,

Continuous-time models can be expressed as

$$\hat{y}(t) = f\left(x_1(t),\ldots,x_n(t),u(t)\right), \tag{2.1}$$

where
t is continuous time;
$\hat{y}(t)$ is the output of the model;
$u(t)$ is the input of the model;
$x_1(t),\ldots,x_n(t)$ are inner states of the model;
n is the order of the model and
f is a nonlinear function.

Unlike continuous signals, in discrete-time systems signals have discrete amplitude values at discrete times. The sampling is usually performed periodically with sampling time Λt [7]. Only discrete-time systems with constant sampling time are considered in this thesis.

Discrete-time model:

$$\hat{y}(t_{k+1}) = f\left(x_1(t_k),\ldots,x_n(t_k),u(t_k)\right),\ k = 0,1,\ldots. \tag{2.2}$$

Here
t_k is a time instant measured with sample time

$$\Delta t = t_{k+1} - t_k \text{ for } k = 0,1,\ldots. \tag{2.3}$$

For simplicity equation (2.2) can be rewritten as

$$\hat{y}(k+1) = f(x_1(k),\ldots,x_n(k),u(k)),\ k = 0,1,\ldots \tag{2.4}$$

or

$$\hat{y}(t+1) = f(x_1(t),\ldots,x_n(t),u(t)),\ t = 0,1,\ldots \tag{2.5}$$

Sometimes the notation (2.5) is also used in literature and will be used in this thesis when we talk only about discrete-time systems that has to be mentioned.

There exist several types of continuous- and discrete-time models. The choice of the model type depends on available data and information about the process or system that we have to model. It also depends very much on the application where this model will be used as it will be shown in this thesis later.

In the following systems that can be modeled by a finite number of differential or difference equations are considered. The most important types of nonlinear models used in control applications are input-output and state-space models.

For simplicity lets now consider Single Input Single Output (SISO) systems. Multiply Input Multiply Output systems will be considered in the thesis later.

2.2.1 Input-Output Models

In general, the relationships between the input u and the output y signals of a system can be represented by an n-th order differential equation

$$f\left(y,\frac{dy}{dt},\ldots,\frac{d^n y}{dt},u,\frac{du}{dt},\ldots,\frac{d^m u}{dt}\right) = 0 \tag{2.6}$$

where f is a nonlinear function and $m \le n$
or by an n-th order difference equation

$$f(y(k),\ldots,y(k-n),u(k),\ldots,u(k-m)) = 0 \tag{2.7}$$

where f is also a nonlinear function and $m \le n$.

Equation (2.6) is called an input-output model of a continuous-time nonlinear system and equation (2.7) is called an input-output model of a discrete-time nonlinear system [3]-[6].

2.2.2 State-Space Models

Another important class of models are state-space models. A lot of control techniques are based on this state-space representation that consists of state equations and output equations. [3]-[6]

In case of continuous-time nonlinear state-space model the dynamics of the system is modeled by a finite number of coupled first-order ordinary differential equations

$$\begin{aligned} \dot{x}_1 &= f_1(t, x_1, \ldots, x_n, u) \\ \dot{x}_2 &= f_2(t, x_1, \ldots, x_n, u) \\ &\vdots \\ \dot{x}_n &= f_n(t, x_1, \ldots, x_n, u) \end{aligned} \tag{2.8}$$

where $\dot{x}_i$ denotes the derivative of x_i with respect to the time variable t and u is the input of the system, $f_1, f_2, \ldots, f_n$ are nonlinear functions. Variables $x_1, x_2, \ldots, x_n$ are called state variables and n is the order of the model.

Output equation is a static nonlinear function that can be defined as follows

$$y = h(t, x_1, \ldots, x_n, u) \tag{2.9}$$

Equations (2.8) and (2.9) are called nonlinear continuous-time state-space model.

Discrete-time state-space model also consists of state equations and an output equation. Dynamics of the model is represented by a finite number of first-order difference equations

$$\begin{aligned} x_1(k+1) &= f_1\big(k, x_1(k), \ldots, x_n(k), u(k)\big) \\ x_2(k+1) &= f_2\big(k, x_1(k), \ldots, x_n(k), u(k)\big) \\ &\vdots \\ x_n(k+1) &= f_n\big(k, x_1(k), \ldots, x_n(k), u(k)\big) \end{aligned} \tag{2.10}$$

and output equation is

$$y(k) = h(k, x_1(k), \ldots, x_n(k), u(k)) \tag{2.11}$$

Equations (2.10) and (2.11) are called nonlinear discrete-time state-space model.

This thesis is devoted to development, analysis and application of neural networks based models and control algorithms based on these models. All models based on neural networks are discrete-time models. That is why only discrete-time models will be considered further.

2.2.3 NARX models

Nonlinear Autoregressive eXogenous (NARX) models is a sub-class of discrete-time input-output models (2.7). NARX model is represented by a high order difference equation

$$y(k) = f\,(y(k-1),\ldots,y(k-n),u(k-1),\ldots,u(k-n))\,, \tag{2.12}$$

where n is the order of the model.

This is a very widespread model that can model a very large class of nonlinear systems with high degree of accuracy, but it suffers from some serious drawbacks. Namely, it is not always realizable in classical state-space form or this representation is very complicated. It is also not always linearizable by using feedback. [8], [9], [10] The second property is very important for nonlinear control system design and will be discussed in more detail in section 2.4.3.

Additive NARX (ANARX) model was proposed in [11] to bridge the gap.

2.2.4 ANARX models

Unlike NARX models, ANARX models have all time instances separated [8], [11], [33]. The idea of separating time-instances was proposed in [12]. ANARX model is described by the following equation

$$\begin{aligned} y(t) &= f_1(y(t-1),u(t-1)) + \ldots + f_n(y(t-n),u(t-n)) = \\ &= \sum_{i=1}^{n} f_i(y(t-i),u(t-i)) \end{aligned} \tag{2.13}$$

Here $f_1,\ldots,f_n$ are nonlinear functions and n is the order of the nonlinear model (memory length).

One of the main benefits of this model is that it well suites to berepresented by artificial neural networks of the structure specified in [10] as it is shown in [13], [14]. Neural Networks based model will be described in detail in section 2.5.2 of this thesis.

Without regard to how ANARX model was obtained, by training a neural network or not, it is always directly rewritable in the form of state-space model by using equations (2.14)

$$\begin{aligned} x_1(t+1) &= x_2(t) + f_1(x_1(t), u(t)) \\ x_2(t+1) &= x_3(t) + f_2(x_1(t), u(t)) \\ &\vdots \\ x_{n-1}(t+1) &= x_n(t) + f_{n-1}(x_1(t), u(t)) \\ x_n(t+1) &= f_n(x_1(t), u(t)) \\ y(t) &= x_1(t) \end{aligned} \tag{2.14}$$

n states x_1 to x_n corresponding to the order of the model have to be introduced. State-space representation is very important for design and analysis of control systems. State-space concept is extremely useful not only for design of a specific optimal control system but also for improving the principle on which the system will operate.[15]

Another advantage of this model in means of control systems is that it is always linearizable by using dynamic output feedback linearization as it will be shown in section 2.4.3.

Model-based control algorithms suppose that the model obtained through the identification procedure are accurate. It means that the model describes the properties of the system that are important to control it with high enough degree of accuracy. The quality of a model-based control system depends very much on the quality of the model. The following approach can be used to evaluate the accuracy of a model.

2.3 Accuracy of a model

Model is a function that maps an input value *u* from the set of system inputs *U* into the set of model outputs $\hat{Y}$:

$$u \overset{\mathrm{mod}\,el}{\rightarrow} \hat{y}, \; u \in U, \; \hat{y} \in \hat{Y} \tag{2.15}$$

An output value $\hat{y}$ from the set $\hat{Y}$ is put into the correspondence to each input *u* from the set *U* by the model. In case of SISO systems $U, \hat{Y} \subset \Re$ and in case of MIMO systems $U \subset \Re^r$ and $\hat{Y} \subset \Re^m$ where *r* is the number or of model inputs and *m* is the number of model outputs.

At the same time each input *u* corresponds to a value *y* from the set of system outputs *Y*:

$$u \xrightarrow{system} y\,,\ u \in U\,,\ y \in Y \tag{2.16}$$

The relationship between the system and the model is schematically depicted in figure 2.1.

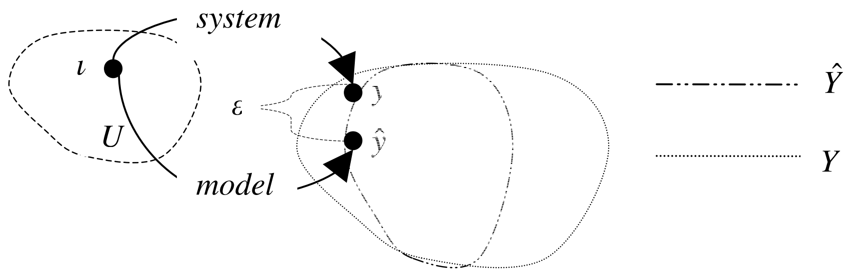

figure 2.1 Relationship between the system and the model

$y \neq \hat{y}$ because of inaccuracy of the model. The distance between the output of the system *y* and the output of the model $\hat{y}$ is inaccuracy of the model at point *u* and it is denoted as *ε* in figure 2.1.

$$\varepsilon = y - \hat{y} \tag{2.17}$$

Accuracy of the model can be measured on so-called validation set. Validation set is a set of known system input and system output pairs. Model outputs corresponding to each input value from the validation set are calculated and Mean Square Error (MSE) can now be used as the measure of the model accuracy.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2 \tag{2.18}$$

where *N* is the number of input-output pairs in the validation set.

For MIMO systems MSE can be calculated as

$$MSE = \frac{1}{N \cdot m}\sum_{i=1}^{N}\sum_{j=1}^{m}\left(y_{i,j} - \hat{y}_{i,j}\right)^2 \tag{2.19}$$

where m is the number of system outputs. Number of model outputs equals to the number of system outputs.

2.4 Model-based nonlinear system control techniques

After a nonlinear model of a nonlinear system is obtained, model-based control techniques can be applied to control the system. The most popular model-based algorithms for nonlinear system control are considered in this section.

2.4.1 Predictive control

Predictive control is one of the most widespread techniques for nonlinear systems control. The main advantage of this method is that it does not depend on the form of the model. The model of a nonlinear system can be obtained by using any type of artificial neural network, by using fuzzy systems, can be derived from physical laws and so on. No particular structure of the model is required.

Training an artificial neural network is one of nonlinear system identification techniques. Predictive control algorithm does not depend on the structure of neural network used as a model and can be considered as an algorithm that uses the model of the process under control as a “black box” with inputs and outputs.

The predictive control has been shown to exhibit robustness to model order uncertainly, variable time delay, and non minimum phase effects [16]. Predictive control schemes minimize future output deviations from set point while taking into account the control action needed to achieve the objective [17]. Such schemes have been proved to be very successful for linear systems [18]-[21] and have also been used with nonlinear models [17], [22]-[24]. Neural networks ability to learn and model wide range of nonlinearities makes them ideal candidates for use with a predictive control scheme. Neural Networks based predictive control technique will be discussed later in section 4.2.1.

The model is used to predict the behavior of the system under control. We suppose that the input of the system remains the same and calculate some steps of model outputs. These model outputs are used as predicted outputs of the system. The aim of the control system is to minimize the cost function $J(t)$.

$$J(t) = \sum_{j=N_1}^{N_2} (v(t+j) - \hat{y}(t+j))^2 + \sum_{j=1}^{N_u} \lambda_j (\hat{u}(t+j-1) - \hat{u}(t+j-2))^2 \qquad (2.20)$$

Here

$v(t)$ is the desire output of the system (reference signal) at time step t;

$\hat{y}(t)$ is the output of the model (predicted output) at time step t;

$\hat{u}(t)$ is the input of the system (control signal);
N_1 is the maximum output prediction horizon;
N_2 is the maximum output prediction horizon;
N_u is the control horizon
$N_u = N_2 - N_1 + 1$
and λ_j, $j = 1, \ldots, N_u$ is the control weighting sequence.

Minimum of the function (2.20) can be calculated by using gradient ∇J . Lets divide J into two parts

$$J = J_y + J_u \tag{2.21}$$

where

$$J_y = \sum_{j=N_1}^{N_2} (v(t+j) - \hat{y}(t+j))^2 \tag{2.22}$$

and

$$J_u = \sum_{j=1}^{N_u} \lambda_j (\hat{u}(t+j-1) - \hat{u}(t+j-2))^2 \quad . \tag{2.23}$$

Then

$$\nabla J = \begin{bmatrix} \dfrac{\partial J_y}{\partial \hat{u}(t)} \\ \vdots \\ \dfrac{\partial J_y}{\partial \hat{u}(t+N_u-1)} \end{bmatrix} + \begin{bmatrix} \dfrac{\partial J_u}{\partial \hat{u}(t)} \\ \vdots \\ \dfrac{\partial J_u}{\partial \hat{u}(t+N_u-1)} \end{bmatrix} \tag{2.24}$$

Equation (2.25) has to be solved to find the value of argument $u(t)$ at witch the value of the cost function (2.20) is minimal.

$$\nabla J = 0 \tag{2.25}$$

where $\nabla J \in \Re^{N_u}$ and 0 is a zero vector of the same dimension.

We assume that the reference signal $v(t)$ remains constant during the length of prediction.

$$v(t+N_1) = \ldots = v(t+N_2) = v(t) \tag{2.26}$$

It means that

$$\frac{\partial v(t)}{\partial \hat{u}(t)} = \ldots = \frac{\partial v(t)}{\partial \hat{u}(t+N_u-1)} = 0 \tag{2.27}$$

and consequently

$$\nabla J = -2 \cdot \begin{bmatrix} \frac{\partial \hat{y}(t+N_1)}{\partial \hat{u}(t)} & \cdots & \frac{\partial \hat{y}(t+N_2)}{\partial \hat{u}(t)} \\ \vdots & \ddots & \vdots \\ \frac{\partial \hat{y}(t+N_1)}{\partial \hat{u}(t+N_u-1)} & \cdots & \frac{\partial \hat{y}(t+N_2)}{\partial \hat{u}(t+N_u-1)} \end{bmatrix} \cdot \begin{bmatrix} v(t) - \hat{y}(t+N_1) \\ \vdots \\ v(t) - \hat{y}(t+N_2) \end{bmatrix} +$$

$$+ 2 \cdot \begin{bmatrix} \lambda_1 & -\lambda_2 & 0 & \cdots & 0 \\ 0 & \lambda_2 & -\lambda_3 & \cdots & 0 \\ 0 & 0 & \lambda_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_{N_u} \end{bmatrix} \cdot \begin{bmatrix} \hat{u}(t) - \hat{u}(t-1) \\ \vdots \\ \hat{u}(t+N_u-1) - \hat{u}(t+N_u-2) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{2.28}$$

$\frac{\partial \hat{y}(k_1)}{\partial \hat{u}(k_2)} = 0$ for $\forall k_2 > k_1 - N_1$ because predicted output $\hat{y}(k_2)$ does not depend on the input $\hat{u}(k_2)$ and changes in $\hat{u}(k_2)$ do not cause any changes in $\hat{y}(k_2)$. It means that

$$\begin{bmatrix} \lambda_1 & -\lambda_2 & 0 & \cdots & 0 \\ 0 & \lambda_2 & -\lambda_3 & \cdots & 0 \\ 0 & 0 & \lambda_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_{N_u} \end{bmatrix} \cdot \begin{bmatrix} \hat{u}(t) \\ \vdots \\ \hat{u}(t+N_u-1) \end{bmatrix} = \begin{bmatrix} \lambda_1 & -\lambda_2 & 0 & \cdots & 0 \\ 0 & \lambda_2 & -\lambda_3 & \cdots & 0 \\ 0 & 0 & \lambda_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_{N_u} \end{bmatrix} \cdot \begin{bmatrix} \hat{u}(t-1) \\ \vdots \\ \hat{u}(t+N_u-2) \end{bmatrix} +$$

$$+\begin{bmatrix} \frac{\partial \hat{y}(t+N_1)}{\partial \hat{u}(t)} & \cdots & \frac{\partial \hat{y}(t+N_2)}{\partial \hat{u}(t)} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\partial \hat{y}(t+N_2)}{\partial \hat{u}(t+N_u-1)} \end{bmatrix} \cdot \begin{bmatrix} v(t)-\hat{y}(t+N_1) \\ \vdots \\ v(t)-\hat{y}(t+N_2) \end{bmatrix} \quad (2.29)$$

Now lets multiply the left and the right parts of equation (2.29) by (2.30)

$$\begin{bmatrix} \lambda_1 & -\lambda_2 & 0 & \cdots & 0 \\ 0 & \lambda_2 & -\lambda_3 & \cdots & 0 \\ 0 & 0 & \lambda_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_{N_u} \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{\lambda_1} & \frac{1}{\lambda_1} & \cdots & \frac{1}{\lambda_1} \\ 0 & \frac{1}{\lambda_2} & \cdots & \frac{1}{\lambda_2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\lambda_{N_u}} \end{bmatrix} \quad (2.30)$$

Element $\hat{u}(t)$ from vector $\left[\hat{u}(t),\ldots,\hat{u}(t+N_u-1)\right]^T$ is applied as the control signal.

$$u(t)=\hat{u}(t) \quad (2.31)$$

$$u(t)=u(t-1)+\frac{1}{\lambda_1}\cdot\left[\frac{\partial \hat{y}(t+N_1)}{\partial \hat{u}(t)},\ \frac{\partial \hat{y}(t+N_1+1)}{\partial \hat{u}(t)}+\frac{\partial \hat{y}(t+N_1+1)}{\partial \hat{u}(t+1)},\ \ldots\right.$$

$$\left.\ldots,\ \frac{\partial \hat{y}(t+N_2)}{\partial \hat{u}(t)}+\cdots+\frac{\partial \hat{y}(t+N_2)}{\partial \hat{u}(t+N_u-1)}\right]\cdot\begin{bmatrix} v(t)-\hat{y}(t+N_1) \\ \vdots \\ v(t)-\hat{y}(t+N_2) \end{bmatrix}. \quad (2.32)$$

Equation (2.32) can be significantly simplified in case of non-adaptive predictive control algorithm. In this case vector of weight coefficients Q does not change.

$$Q=\left[q_1 \ \ldots \ q_{N_u}\right]=\frac{1}{\lambda_1}\cdot\left[\frac{\partial \hat{y}(t+N_1)}{\partial \hat{u}(t)},\ \frac{\partial \hat{y}(t+N_1+1)}{\partial \hat{u}(t)}+\right.$$

$$\left.+\frac{\partial \hat{y}(t+N_1+1)}{\partial \hat{u}(t+1)},\ldots,\ \frac{\partial \hat{y}(t+N_2)}{\partial \hat{u}(t)}+\cdots+\frac{\partial \hat{y}(t+N_2)}{\partial \hat{u}(t+N_u-1)}\right] \quad (2.32)$$

Elements of vector $Q \in \Re^{1 \times N_u}$ are parameters of predictive controller and the quality of control significantly depends on them. Control signal then can be calculated as

$$u(t) = u(t-1) + \begin{bmatrix} q_1 & \dots & q_{N_u} \end{bmatrix} \cdot \begin{bmatrix} v(t) - \hat{y}(t+N_1) \\ \vdots \\ v(t) - \hat{y}(t+N_2) \end{bmatrix} \tag{2.33}$$

When model is obtained in the form of artificial neural network, this algorithm can be improved to adaptive by training the network and adjusting parameters of the model as it will be shown later in section 4.2.1. General structure of the control system is shown in figure 2.2.

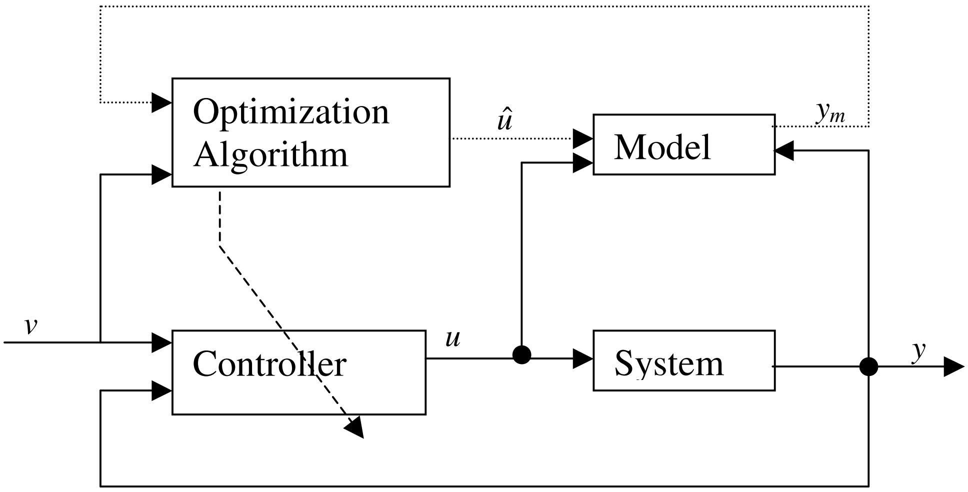

figure 2.2 Predictive Control technique

Another model based control technique that does not depend on the structure of the model and the method how it was obtained is inverse model based control.

2.4.2 Inverse model based control

The idea of inverse model based control technique is in implementing an inverse model of the system as the controller.

Consider an input-output discrete time model as defined in (2.7). The output of the model can be calculated as

$$y(t) = f(y(t-1), \dots, y(k-n), u(t), \dots, u(t-m)) \tag{2.34}$$

The aim of the inverse model is to calculate the input of the system *u(t)* that causes the known output *y(t)*. It can be calculated as

$$u(t) = f_{im}(y(t), y(t-1), \ldots, y(k-n), u(t-1), \ldots, u(t-m)) \qquad (2.35)$$

when a nonlinear function f_{im} is found.

After that this "known output" is taken as the reference signal *v(t)* and after being given to the input of an inverse model, the inverse model calculates control signal *u(t)* that has to be given to the input of the system in order to obtain $y(t) = v(t)$. Such a control technique is called *direct inverse control.*

Direct inverse control utilizes an inverse system model. The inverse model is simply cascaded with the controlled system in order that the composed system results in an identity mapping between desired response and the controlled system output [23].

Besides the fact that this control technique has been successfully applied in robotics [23], it suffers from some serious drawbacks. The main drawback is absence of any feedback from real output of the process that makes the control strategy very dependant on the quality of the inverse model. We fully rely on it. Serious questions arise regarding the robustness of direct inverse control [23]. This problem can be overcome to some extend by implementing a neural networks based inverse model and using an adaptation technique based on training of the network to minimize the difference between the reference signal and the output of the system. The parameters of the inverse model can be adjusted on-line. Such a technique is shown by the author in [26]. It introduces a feedback into the control system and significantly improves its quality and robustness.

When an inverse model is obtained in the form of a NARX structure (2.12) the feedback can be introduced by providing previous input and output values for the inverse model from the controlled process. The corresponding control scheme [25] is illustrated in figure 2.3

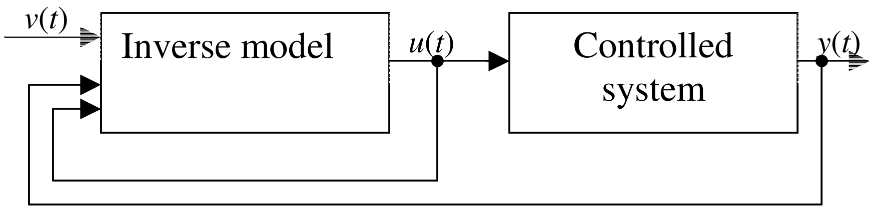

figure 2.3 Direct inverse control

For better robustness, the inverse model can also be used in combined with a forward model. In this strategy called *Internal Model Control* (IMC), the difference e between the output of the forward model $\hat{y}$ and the output of the plant y is used as feedback signal, while the inverse model is placed in the forward path.[25] This approach is illustrated in figure 2.4.

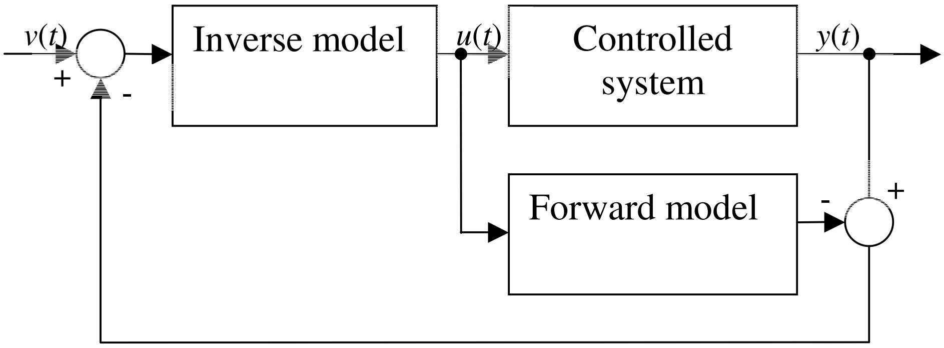

figure 2.4 Internal model control

If the forward model is perfect, the error signal will be zero and the control system will operate as if it was under direct inverse control.

Another important problem is defining the inverse function f_{im}. It has to be mentioned that not all continuous functions are invertible. This fact significantly restricts the class of systems to which this technique can be applied. If a continuous one-valued function defining an inverse model of a nonlinear system exists it can be modeled by a neural network because of neural networks ability to approximate any continuous nonlinear function. That is why neural networks inverse modelling based versions of this control algorithm became so popular and were successfully applied to control of nonlinear systems [26]-[31]. Neural networks based approach to inverse model based control will be considered in more detail in section 4.2.3.

The next approach to control of nonlinear systems is based on the specific of discrete-time model.

2.4.3 Dynamic Output Feedback Linearization

It is proofed in [32] that ANARX structure (2.13) described in the section 2.2.4 is always linearizable by Dynamic Output Feedback Linearization (2.36)-(2.38). Consider n-th order ANARX model. The dynamics of the linearization algorithm is defined by the following equations

$$
\begin{aligned}
&\eta_1(t+1) = \eta_2(t) - f_2\big(y(t),u(t)\big) \\
&\vdots \\
&\eta_{n-2}(t+1) = \eta_{n-1}(t) - f_{n-1}\big(y(t),u(t)\big) \\
&\eta_{n-1}(t+1) = v(t) - f_n\big(y(t),u(t)\big)
\end{aligned}
\tag{2.36}
$$

and control *u(t)* can be calculated as

$$u(t) = F^{-1}(y(t), \eta_1(t)) \tag{2.37}$$

where

$$F = f_1\big(y(t),u(t)\big) = \eta_1(t) \tag{2.38}$$

It can be seen from equations (2.36) and (2.38) that

$$
\begin{aligned}
&v(t-n) = f_1\big(y(t-1),u(t-1)\big) + f_2\big(y(t-2),u(t-2)\big) + \ldots \\
&\ldots + f_n\big(y(t-n),u(t-n)\big) = \sum_{i=1}^{n} f_i\big(y(t-i),u(t-i)\big) = y(t)
\end{aligned}
\tag{2.39}
$$

or

$$y(t) = v(t-n) \tag{2.40}$$

when dynamic output linearization algorithm (2.36)-(2-38) is applied to ANARX model (2.13).

In order to use this linearization algorithm as a nonlinear system control method. The system has to be modeled by the ANARX structure. Such a model can be obtained by training a specific neural network. Neural Networks based ARARX (NN-ANARX) structure will be introduced in section 3.6 and NN-ANARX structure based control technique will be shown in section 4.3.

The main problem in implementation of this technique to control of nonlinear systems is of control signal (2.37). Inverse of the function *F* (2.38) has to be calculated. The value of the function F^{-1} can be calculated numerically by using Newton method. [13], [14]

2.4.4 Newton's method

By using Newton's method [34] a nonlinear equation (2.41) can be solved.

$$f(x) = 0 \tag{2.41}$$

If x_0 is an initial point (initial approximation of the solution x^*) then the zero of the function (2.41) can be numerically calculated by using the following iterations

$$x_{k+1} = x_{k+1} - \frac{f(x_k)}{f'(x_k)}, \quad k = 0,1,2,\ldots$$
$$x_k \xrightarrow{k\to\infty} x^* \tag{2.42}$$

where $f'(x_k)$ is the derivative of the function $f(\cdot)$ at point x_k. Consequently this function has to have derivatives at all points x_k.

In case of nonlinear SISO systems modeled by ANARX structure, function (2.38) can be rewritten as

$$f_1(y(t),u(t)) - \eta_1(t) = 0 \tag{2.43}$$

Now it satisfies the condition (2.41) and the zero $u^*(t)$ can be calculated by using iterations (2.42) if function $f_1(\cdot)$ is differentiable. After applying p iterations approximation $u_p(t)$ is used as the control.

$$u(t) = u_p(t) \tag{2.44}$$

Previous control signal $u(t-1)$ can be used as an initial gas $u_0(t)$ for numerical calculation of the control $u(t)$ by Newton's method as it is usually reasonably close to the true zero $u^*(t)$.

$$u_0(t) = u(t-1) \tag{2.45}$$

When ANARX model is obtained in the form of artificial neural network as will be shown in section 3.6, function $f_1(\cdot)$ is differentiable at all points $u \in \Re$ and therefore Newton's method can be used for calculation of (2.37) as it is shown by the author of the thesis in [13] and [14].

This thesis is devoted to neural network based modeling and control. The idea of artificial neuron and artificial neural networks will be briefly described in the next section.

2.5 Artificial Neural Networks

Artificial Neural Networks consist of basic nodes – Artificial Neurons. To talk about Artificial Neural Networks, the notation of an Artificial Neuron has to be defined.

2.5.1 Artificial neurons

In 1909 Cajal [35] found that the brain consists of a large number of a large number of highly connected neurons. In 1943, McCulloch and Pitts proposed a mathematical model of the neurons and showed how neural-like networks could be computed.[24] The simplified mathematical model of the neuron by McCulloch and Pitts is usually called Artificial Neuron. This model is used in most neural networks based applications and will be used in this thesis. Artificial neuron is schematically depicted in figure 2.5.

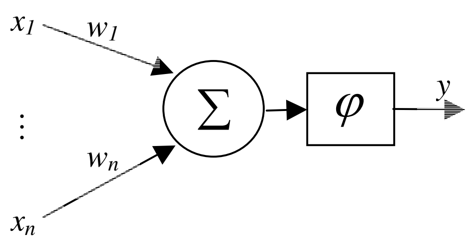

figure 2.5 Artificial Neuron

It can be seen from the picture that artificial neuron consists of two parts. The first part is a weighted sum of inputs and the second part is a nonlinear element called *activation function.*[36], [40], [24]. Choice of this function $\varphi(\cdot)$ significantly depends on application, learning strategy and algorithm [37].

Consider an artificial neuron having n inputs as shown in figure 2.5. Artificial neuron can mathematically be defined by the following function

$$y = \varphi\left(\sum_{i=1}^{n} w_i \cdot x_i\right) \tag{2.46}$$

or in a matrix form

$$y = \varphi\left(\begin{bmatrix} w_1 & \dots & w_n \end{bmatrix}\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}\right) = \varphi(W \cdot X) \tag{2.47}$$

where X is a vector of neuron's inputs and W is a vector of weight coefficients also called *synaptic weights*.

A linear neuron without activation function represented by the equation

$$y = W \cdot X \tag{2.48}$$

is called ADAptive LInear NEuron (here and after ADALINE).

One more parameter θ of an artificial neuron *called* bias or *threshold* that gives additional degree of freedom to the network can be introduced in some realizations of neural networks, but it is also an additional parameter to be adjusted during network's training. The function of a neuron with a bias takes the form (2.49)

$$y = \varphi\left(\sum_{i=1}^{n} w_i \cdot x_i + \theta\right) \tag{2.49}$$

2.5.2 Activation functions

The most common activation function is a continuous function that varies gradually between two asymptotic values, typically 0 and 1, or -1 and 1. These functions are called *sigmoidal functions* [40].

The definition of sigmoid functions can be given as follows

Def. 2.1
A C^k-sigmoid function $\sigma : \Re \to \Re$ is a nonconstant, bounded, and monotone increasing function of class C^k (continuous differentiable up to order k)

In other words, the sigmoid is a smooth nonlinearity with saturation. The most widely used sigmoid activation functions are logistic function and hyperbolic tangent represented by equations (2.50) and (2.51). The main advantage of using these functions is that they are always differentiable and it is very easy and fast (that is very important for control applications) to calculate the derivatives of these functions [37] as shown below.

Logistic function $\left(\varphi(I)\in[0;1]\right)$

$$\varphi(I)=\frac{1}{1+e^{-\alpha\cdot I}}\Rightarrow\varphi'(I)=\alpha\cdot\varphi(I)\cdot(1-\varphi(I)) \tag{2.50}$$

Hyperbolic tangent $\left(\varphi(I)\in[-1;1]\right)$

$$\varphi(I)=\frac{1-e^{-\alpha\cdot I}}{1+e^{-\alpha\cdot I}}\Rightarrow\varphi'(I)=\frac{\alpha}{2}\cdot\left(1-\varphi^{2}(I)\right) \tag{2.51}$$

It has been estimated that there are more than 100 billion (10^{11}) neurons in a human brain [41], [37]. Artificial neural networks have significantly less neurons. The largest artificial neural networks in control applications have no more than a few hundred neurons.

The neurons are organized as a natural network to receive information from the real-world environment, and then provide the corresponding response, e.g. decisions or actions. However, from an engineering point of view, a neural network can be viewed as a parallel information processing system with some human-like intelligent behavior.[37]

An artificial neural network can be defined as

A data processing system consisting of a large of simple, highly interconnected processing elements (artificial neurons) in an architecture inspired by the structure of the cerebral cortex of the brain [40].

These processing elements are usually organized into a sequence of layers. A typical neural network is “fully connected”, which means that there is a connection between each of the neurons in any given layer with each of the neurons in the next layer.[40]

2.5.3 Perceptron

In 1962 Rosenblatt invented a class of simple neuron-like learning networks which is called *perceptron* [42], [24]. The perceptron was actually an entire class of architecture which was composed of processing units that transmitted signals and adapted their connection weights. Rosenblat’s research was oriented toward modeling the brain in an attempt to understand memory, learning and cognitive process. Rosenblatt’s works were extended by many scientists and engineers and a number of machines were built based on perceptron architectures.[24]

Multi-layer perceptrons have very quickly become the most widely encountered artificial neural networks, particularly within the area of systems and control [23]. The structure of a two-layer perceptron is depicted in figure 2.6.

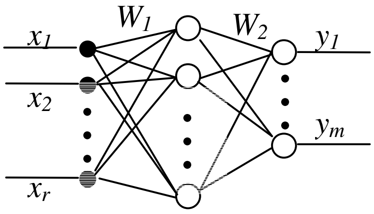

Figure 2.6 Two-layer perceptron

The first layer (from left to right) is called input layer. Te aim of elements of this layer is to receive signals from the environment and divide it between neurons of the layer neurons. All the layers that are connected to the network's environment are called hidden layers. The receive signals from previous layer neurons and produce signals for the next layer neurons. In the considered example the second layer is a hidden layer. Neurons of the last layer produce outputs of the network. The percepton shown in the picture is called two-layer perceptron (not three-layer), because there are no calculations in the first layer (input layer) and it is not taken into account when number of layers is calculated.

If r is the number of network inputs, m is the number of network outputs (number of output layer neurons) and l is the number of hidden layer neurons then $W_1 \in \Re^{l\times r}$ is a $l \times r$ matrix of synaptic weights between inputs and hidden layer neurons and $W_2 \in \Re^{m\times l}$ is a $m \times l$ matrix of synaptic weights between hidden layer and output layer. $\theta_1 \in \Re^{l\times 1}$ and $\theta_2 \in \Re^{m\times 1}$ are vectors of hidden an output layer neurons biases. Now a two-layer perceptron is reprezentable by the following mathematical function

$$Y = f_2\left(W_2 \cdot f_1\left(W_1 \cdot X + \theta_1\right) + \theta_2\right) \tag{2.52}$$

Where Y is the vector of network outputs, f_1 and f_2 are activation functions of hidden and output layer neurons correspondingly.

It has to be mentioned that not all neural networks (including perceptrons) have biases. They give additional parameters and sometimes make training faster, but neural networks can represent nonlinearities also without them. In this case the function reprezenting a perceptron is

$$Y = f_2\left(W_2 \cdot f_1\left(W_1 \cdot X\right)\right) \tag{2.53}$$

According to the Stone-Weierstrass theorem multilayer perceptrons are capable of approximating any continuous functions.

2.5.4 Stone-Weierstrass Theorem

The original theorem due to Weierstass shows that an arbitrary continuous function $f:[a,b]\rightarrow\Re$ can be uniformly approximated by a sequence of polynomial $\{p_n(x)\}$ to within a desired accuracy. This theorem was analyzed by Stone [44], who tried to find the general properties of approximating functions, not necessary polynomials [17]. Stone-Weierstrass theorem is relevant for approximation of continuous functions by artificial neural networks. To state the results of Stone some definitions have to be given.

Def. 2.2
A set A of functions from $K\subset\Re^r$ *to* $\Re$ *is called an algebra of functions iff* $\forall f,g\in\Re$ *and* $\forall\alpha\in\Re$ [17]

(I) $f+g\in A$
(II) $f\cdot g\in A$ (2.54)
(III) $\alpha\cdot f\in A$

Def. 2.3
Let B be the set of all functions which are limits of uniformly convergent sequences with terms in A, a set of functions from $K\subset\Re^r$ *to* $\Re$*. Then B is called a the uniform closure of A.*[17]

Def. 2.4
A set A of functions from $K\subset\Re^r$ *to* $\Re$ *is said to separate points on K iff* $\forall x_1,x_2\in K\ \ x_1\neq x_2\Rightarrow\exists f\in A,\ f(x_1)\neq f(x_2)$.[17]

Def. 2.5
Let A be a set of functions from $K\subset\Re^r$ *to* $\Re$*. We say that A vanishes at no point of K iff* $\forall x_1,x_2\in K\ \ \exists f\in A$*, such that* $f(x)\neq 0$.[17]

The main result of Stone [44] is the following

Theorem 2.1 (Stone-Weierstrass)
Let A be an algebra of some continuous functions from a compact $K\subset\Re^r$ *to* $\Re$*, such that A separates points on K and vanishes at no point of K. Then the uniform closure B of A consists of all continuous functions from K to* $\Re$.

The original formulation of the theorem is for $f:\Re^r\rightarrow\Re$ due to condition (II) in (2.54), but the result remains also valid for , because the codomain of a vector-valued functions is the cartesian product of its components.[17]

Two-layer perceptron is described by equation (2.52) or (2.53). To approximate nonlinear functions neurons of the hidden layer should have a nonlinear activation function. It means that at least function f_1 in (2.52) and (2.53) is a nonlinear function. Function f_2 is usually a linear function (in this case outputs of the network are not bounded). When f_1 is a sigmoid function and f_2 is an ADALINE as defined in (2.48), the function of the network is a linear combination of sigmoids.

The set of all linear combinations of sigmoids is a nonvanishing algebra separating point on a compact $K \subset \Re^r$ [17] and according to theorem 2.1 this type of neural networks is suitable for uniform approximation of an arbitrary continuous map. It means that any continuous nonlinear function can be approximated by a two-layer perceptron with sigmoid activation functions of its hidden layer and linear output layer neurons to within a desired accuracy.

There are no feedbacks in the structure of multi-layer perceptron. Such networks with no internal feedback between neurons and/or layers are called feedforward networks or static networks. Such networks can approximate only static functions. In order to model dynamic systems by this type of neural networks, an external feedback is needed (see section 3.3).

Recurrent (also called dynamic or feedback) neural networks is an alternative to static neural networks.

2.5.5 Recurrent Neural Networks

Neural networks of this type have internal feedbacks from outputs of neurons to the inputs of the same (or previous) layer neurons. These networks do not need any additional external feedbacks to model a dynamic processes. Number of network inputs equals to the number of the process's inputs. The Elman Network [45] will be considered as an example of recurrent networks. Capabilities of this network are studied in [46].

The Elman network commonly is a two-layer network with feedback from the first layer output to the first layer input. This recurrent connection allows the Elman network to detect and generate time-varying patterns. A two-layer Elman network is depicted in figure 2.7.

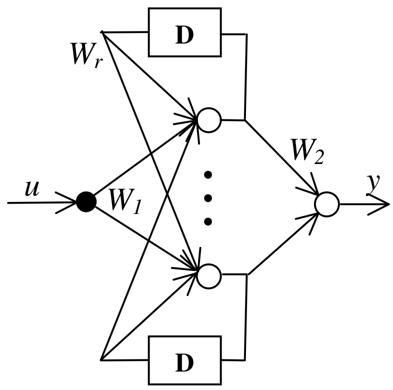

figure 2.7 Recurrent Elman network

The Elman network shown in the network has one hidden layer called *recurrent layer*. The function of the network with one input and one output satisfy the following equations

$$\begin{cases} X(t) = f(X(t-1), u(t)) \\ y(t) = h(X(t)) \end{cases} \tag{2.55}$$

where *u(t)* is the input of the network, *y(t)* is the output of the network and *X(t)* is the vector of recurrent layer outputs. It can be seen from equation (2.55) and figure 2.7 that the Elman network is a dynamic network. The output values of the hidden (recurrent) layer are stored in memory for one tact (delayed for one time step) and produce the vector of the network's states that influences output and state-vector of the network at the next time step.

Consider Elman network having *r* inputs, *m* outputs and *l* neurons in its recurrent layer. Then the network has the following parameters: $W_1 \in \Re^{l \times r}$ is the matrix of synaptic weights between network's inputs and recurrent layer neurons, $W_2 \in \Re^{m \times l}$ is the matrix of synaptic weights between network's hidden layer neurons and output layer neurons, $W_2 \in \Re^{n \times n}$ is the square matrix of synaptic weights between hidden layer outputs and inputs of the same layer (weight coefficients of the feedback), $\theta_1 \in \Re^{1 \times l}$ is the vector of hidden layer biases and $\theta_2 \in \Re^{1 \times m}$ is the vector of output layer biases. When recurrent layer neurons have nonlinear sigmoid activation function $f(\cdot) = \sigma(\cdot)$ and output layer neurons are linear ($h(\cdot)$ is a linear function), the function of the Elman network can be defined as (2.56) or (2.57) following from equation (2.55) by introducing parameters of the network.

$$\begin{cases} X(t) = f(W_1 \cdot U(t) + W_r \cdot X(t-1) + \theta_1) \\ y(t) = W_2 \cdot X(t) + \theta_2 \end{cases} \tag{2.56}$$

When there are no biases in the network (or they equals to zero) this equation takes the following form

$$\begin{cases} X(t) = f(W_1 \cdot U(t) + W_r \cdot X(t-1)) \\ y(t) = W_2 \cdot X(t) \end{cases} \tag{2.57}$$

The Elman network is capable of approximating any continuously differentiable dynamic functions as it was proofed by Sontag in [47].

2.5.6 Sontag Theorem

The main result of [47] is briefly stated here

Theorem 2.2 (Sontag)
Let a system

$$\begin{aligned} \dot{x} &= f(x,u), \qquad x(t_0) = x^0 \\ y &= h(x) \end{aligned} \tag{2.58}$$

be given with $x \in \Re^n$, $u \in \Re^p$, $y \in \Re^q$ *with f and h continuously differentiable, such that for all* $u : [0,T] \rightarrow \Re^p$ *the solution of (2.57) exists and is unique for all* $t \in [0,T]$ *and some compact sets* $K_1 \subset \Re^n$, $K_2 \subset \Re^p$, *while* $x \in K_1$, $u \in K_2$. *Then there exists a recurrent neural network of the form*

$$\begin{aligned} \dot{\chi} &= \sigma(A\chi + Bu), \\ y &= C\chi, \end{aligned} \tag{2.59}$$

where $\chi \in \Re^N$, $N \geq n$, $y \in \Re^q$, σ *is a vector of sigmoids and A, B are matrices such that on* $K_1 \times K_2$,

$$\forall \varepsilon > 0 \ \ \forall t \in [0,T] \ \ \|x(t) - M(\chi(t))\| < \varepsilon \ \text{and} \ \|h(x(t)) - C\chi(t)\| < \varepsilon$$

where M is a differentiable map.[17]

As equations (2.56) and (2.57) satisfy condition (2.59) it can be concluded that the Elman network with sigmoid functions of the recurrent layer and linear output layer neurons is capable of approximating continuously differentiable dynamic functions if the number of the recurrent layer neurons is larger than the order of the system that identifies the network.

Approximating of a function by a neural network (static or dynamic) means calculating (or adjusting) parameters (synaptic weights and biases if any) of the network. This process is called *network training (or learning).* There are supervised and unsupervised training algorithms. Supervised training techniques need a set of etalon maps to be presented to the network. In case of unsupervised learning, a network is capable of adjusting its parameters according to some criteria. Only supervised approach will be considered in this work, because it is more relevant to modeling and control tasks.

2.6 Training algorithms

The first set of ideas of learning in neural networks was contained in Hebb's book [48]. In 1951, Edmunds and Minsky built their learning machine using Hebb's idea. Although Minsky was he first person to propose a learning machine, the real beginning of a meaningful neuron-like network learning can be traced to the work of Rosenblatt [49] in 1962.[24]

Supervised learning considered in this section requires a "teacher" or other source of information which directly specifies the to the network the correct response to each defined stimulus it will encounter. For example, the teacher may be a human or a real process generating examples of the required network outputs corresponding to a defined set of inputs.[39]

Learning is the process of adapting the connection weights in an artificial neural network to produce the desired output vector in response to a stimulus vector presented to the input buffer.[40]

The training is based on a set of input-output data

$$Z_N = \{[u(t), y(t)], t = 1, \ldots, N\}, \tag{2.60}$$

Where N is the size of the trainind set.

Although there is a variety of different training techniques exists, two major training algorithms are used in this work. First of them is gradient descent error backpropagation (BP) training algorithm. Calculation of updates of a network parameters on each iteration is fast. It needs less computational resources than the second one and can be used for on-line adaptation of the model. Levenberg-Marquardt (LM) has better approximation capabilities, it needs less iterations to converge, but each iteration takes a much more time and computational resources. It can be used for off-line modeling.

In both cases the aim of the training is to minimize a cost function $F(\Theta, Z_N)$, where Θ is the vector of parameters of the network (synaptic weights and some applications also biases). Mean square error is usually used as the minimization criterion.

$$F(\Theta, Z_N) = \frac{1}{2N}\sum_{t=1}^{N}\left(y(t) - \hat{y}(t)\right)^2 = \frac{1}{2N}\sum_{t=1}^{N} e^2(t, \Theta). \tag{2.61}$$

Here $y(t)$ is known output value from the training set (2.60) corresponding to the input $u(t)$. It is used as the etalon value ("teacher"). $\hat{y}(t)$ is the output of the network (estimated output) that has to be as close to $y(t)$ as possible.

During the training procedures shown in the next two sections

$$F(\Theta, Z_N) \rightarrow 0 \tag{2.62}$$

2.6.1 Gradient Descent Error Backpropagation

This algorithm is a first-order method. The gradient is defined as

$$G(\Theta) = \frac{\partial F_n(\Theta, Z_n)}{\partial \Theta} = \frac{1}{N}\sum_{t=1}^{N}\frac{\partial e(t, \Theta)}{\partial \Theta} e(t, \Theta) \tag{2.63}$$

When BP algorithm [17] is applied, the vector of network's parameters is updated on each iteration as

$$\Theta(k+1) = \Theta(k) - \lambda(k) \cdot G(\Theta) \tag{2.64}$$

where $\lambda(k)$ is an adaptive adaptive coefficient called *learning coefficient* or *learning rate*.

One iteration of a training algorithm is also called *epoch* or *training epoch*.

2.6.2 Levenberg-Marquardt algorithm

LM algorithm [50], [51] is a blend of Gradient Descent and Gauss-Newton iteration. It also provides a solution for the nonlinear least squares minimization problem (2.61), (2.62). The Hessian is defined by

$$H(\Theta) = \frac{\partial^2 F(\Theta, Z_n)}{\partial\Theta\partial\Theta^T} = \\ = \underbrace{\frac{1}{N}\sum_{t=1}^{N}\frac{\partial e(t,\Theta)}{\partial\Theta}\left[\frac{\partial e(t,\Theta)}{\partial\Theta}\right]^T}_{R(\Theta)} - \frac{1}{N}\sum_{t=1}^{N}\frac{\partial^2 e(t,\Theta)}{\partial\Theta\partial\Theta^T}e(t,\Theta) \tag{2.65}$$

The update rule in LM method is

$$\Theta(k+1) = \Theta(k) + \Delta\Theta(k) \tag{2.66}$$

where

$$[R(\Theta(k)) + \lambda I]\Delta\Theta(k) = -G(\Theta(k)). \tag{2.67}$$

Here $G(\Theta(k))$ is the gradient defined by (2.63), I is the identity matrix and λ is the adaptive learning coefficient. This coefficient balances the behavior of the method between a second- and first-order one.[52], [53]

It was studied in detail in [54] that the training speed of this algorithm is much higher and a feedforward neural network trained with LM algorithm can better model nonlinearities. It is also found in [53] that this algorithm is much more efficient than either of the other techniques when the network contains no more than a few hundred weights.

Chapter 3

Nonlinear Dynamic Systems Identification with Artificial Neural Networks

From mathematical point of view, identification of nonlinear systems means approximation of a nonlinear map (nonlinear function or functions). It was proofed by Stone and Sontag (see sections 2.5.4 and 2.5.6) that artificial neural networks are capable of approximating any continuous nonlinearity. That is why neural networks have become an attractive tool in the construction of models of complex nonlinear processes and a large number of identification structures based on neural networks have been proposed.

Application of recurrent (dynamic) neural networks and feedforward neural networks based structures with external feedback for identification of nonlinear dynamic systems will be discussed in this section. Some application-specific feedforward network structures will also be shown. Identification of nonlinear SISO and MIMO systems will be considered.

3.1 Author's contribution

Author's contribution is in comparing different neural network structures, developing neural network structures for improvement of identification quality of some specific nonlinear systems (systems with static nonlinearities in actuators) and application of neural networks based ANARX (NN-ANARX) structure to identification of nonlinear systems.

- Inverse models of nonlinear dynamic systems based Dynamic neural networks with external and internal feedbacks are studied and compared in [26];
- A neural network structure for identification of systems with static nonlinearities in actuators (neural networks based Hammerstein type models) is proposed in [55];

- neural networks based ANARX (NN-ANARX) structure is applied to identification of nonlinear systems in [13], [14] and [76];
- MIMO NN-ANARX structure is proposed in [56] and applied to identification of nonlinear MIMO systems in [56], [57] and [73];
- MIMO NN-ANARX structure is successfully used for identification and prediction of the surgeon's hand motion in [73].

3.2 Recurrent Neural Networks Based Models

Majority of real world systems and processes have dynamic nature and that is why a dynamic (recurrent) neural network seems to be a reasonable choice for identification of nonlinear dynamic systems. The Elman network described in section 2.5.5 is one of the simplest and widespread recurrent fully connected network structures satisfying the conditions of Theorem 2.2 (See section 2.5.6 for Sontag Theorem).

The Elman network shown in figure 2.7 and defined by equations (2.56) and (2.57) can model can model a nonlinear system after being trained on a corresponding training set consisting of input-output pairs. The set of inputs given to the system is defined as

$$U_N = [U(t-N), U(t-N+1), \dots, U(t)], \tag{3.1}$$

where $U(t)$ is the vector of system's inputs. The number of the network's inputs equals to the number of inputs of the system *r*.

$$U(t) = \begin{bmatrix} u_1(t) \\ \vdots \\ u_r(t) \end{bmatrix}. \tag{3.2}$$

Inputs have to be measured continuously with a constant sample time as well as the corresponding outputs.

$$Y_N = [Y(t-N), Y(t-N+1), \dots, Y(t)], \tag{3.3}$$

where $Y(t)$ is the output vector at time step *t*

$$Y(t) = \begin{bmatrix} y_1(t) \\ \vdots \\ y_m(t) \end{bmatrix} \tag{3.4}$$

and m is the number of system outputs.

All the values from are necessary and have to be presented to the network in the right order during the training procedure.

First of all parameters of the network are given random values. Iterative training algorithm is applied and on each iteration k the output of the network (output of the model) $\hat{Y}(k)$ is calculated by (2.57) and compared to the etalon output Y_N. Mean square error is calculated by (2.61) and the process is repeated until it is smaller than a given number ε (desired accuracy of the model).

$$\frac{1}{2N \cdot m}\sum_{i=1}^{m}\sum_{t=1}^{N}\left(y_i(t) - \hat{y}_i(t)\right)^2 < \varepsilon \qquad (3.5)$$

According to Sontag theorem (section 2.5.6) neural network number of hidden layer neurons $n > r$ has to be chosen to obtain a model of a nonlinear dynamic system to within any desired accuracy. Practical problems that arise when applying this approach are described in [26] and [58].

Recurrent networks are not as reliable as feedforward networks, because training happens using an approximation of the error gradient. The contributions of weights and biases to errors via the delayed recurrent connection is ignored. In other case training of recurrent networks becomes extremely complicated.

Feedforward network with external feedbacks is s good alternative to recurrent networks.

3.3 Feedforward Neural Networks with External Feedback for Identification of Dynamic Systems. NN-NARX models

External feedback is sufficient to represent all dynamic systems.[38]

Discrete-time input-output model (2.7) can be realized in the form of a feedforward neural network with external feedback and delays. Static function $f(\cdot)$ can be approximated by a feedforward network to within any degree of accuracy according to Stone-Weierstrass theorem (theorem 2.1, section 2.5.4) and dynamic arguments of this function have to be given to the inputs of the network. Delayed input and output (from external feedback) values are given to the network's additional inputs. To realize model (2.7) a network has to have $n+m+1$ inputs. An example of a dynamic SISO model realized in a form of a feedforward neural network with external feedback is depicted in figure 3.1.

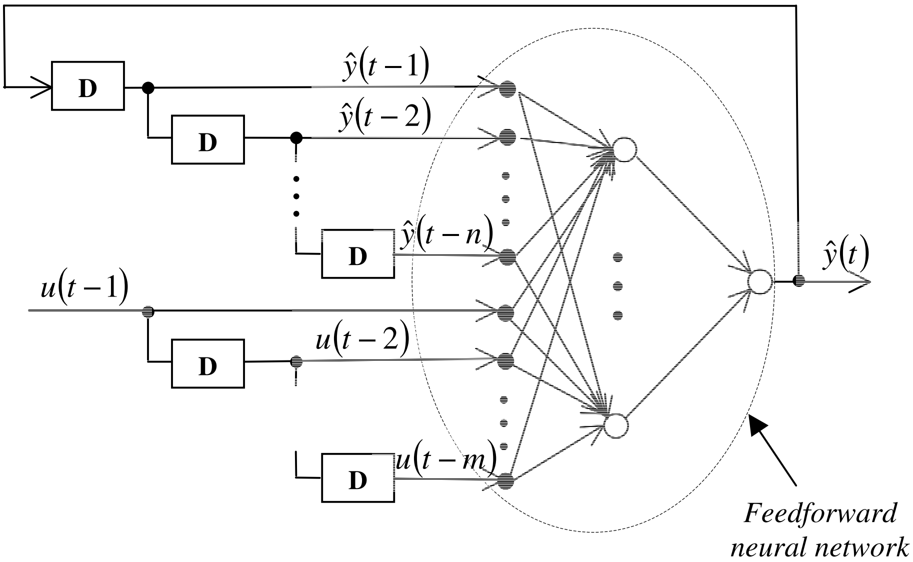

figure 3.1 Reprezentation of dynamic model by a feedforward neural network

Neural networks with a limited feedback which comes only from the output neuron rather from the hidden states are called Nonlinear AutoRegressive eXogenous networks or *NARX networks* [61], [62], [63]. They are formalized by

$$y(t) = \psi(y(t-1), \ldots, y(t-n), u(t-1), \ldots, u(t-m)) \tag{3.6}$$

where Ψ is the mapping performed by a feedforward neural network (for example, by a multilayer perceptron), n is the output order and m is the input order.

It is proofed in [64] that the NARX networks with finite number of parameters are computationally as strong as fully connected recurrent networks and thus Turing machines.

Models of nonlinear systems based on these networks are called *neural networks based NARX models* or *NN-NARX models*. It is also shown in [64] that in theory one can use the NARX models, rather than conventional recurrent networks without any computational loss even though their feedback is limited.

When a two-layer perceptron with nonlinear activation functions of the hidden layer neurons and llinrear output is used as a feedforward network in this model, it can be expressed by the following equation

$$\hat{y}(t) = C \cdot \varphi(W \cdot [\hat{y}(t-1), \ldots, \hat{y}(t-n), u(t-1), \ldots, u(t-m)]), \tag{3.7}$$

where $\hat{y}(t)$ is the output of the network (estimated output) calculated by the model at time step *t*; $\varphi(\cdot)$ is a nonlinear activation function of the hidden layer neurons; *C* is the vector of synaptic weights of the linear output neuron and *W* is the matrix of synaptic weights of the hidden layer neurons.

When the network has l neurons in the hidden layer, $W \in \Re^{l \times (n+m)}$ and $C \in \Re^{1 \times l}$. It means that $l \cdot (n+m) + l = l \cdot (n+m+1)$ parameters have to be adjusted by a training algorithm to model the dynamic system.

If a MIMO system with r inputs and k outputs is modeled by a two-layer perceptron with external feedback, then $W \in \Re^{l \times (k \cdot n + r \cdot m)}$, $C \in \Re^{k \times l}$ and already $l \cdot (k \cdot n + r \cdot m) + k \cdot l = l \cdot (k \cdot (n+1) + r \cdot m)$ synaptic weights have to be adjusted.

Unlike identification by recurrent neural networks discussed in the previous section, permutations in the training set are allowed during the training when a feedforward network is used. Permutations between input vectors (not between elements in each vector) are possible, because all the information about the dynamics of the process is presented in each input pattern.

By using this approach a neural network based model of a nonlinear dynamic system can very easily be obtained by training a feedforward network. Consider the following example.

3.3.1 Numerical example 3.1

A nonlinear system [59], [60] that is to be identified is represented by the following discrete-time input-output equation

$$\begin{aligned} y(t) = {} & \frac{1.5y(t-1)y(t-2)}{1 + y^2(t-1) + y^2(t-2)} + 0.3\cos(0.5(y(t-1) + y(t-2))) + \\ & + 1.2u(y-1) + e(t) \end{aligned} \tag{3.8}$$

where the system input $u(t)$ is a uniform distributed signal in the range $[-1,\ 1]$, and the noise sequence $e(t) \sim N(0,\ 0.1^2)$. A data sequence of 1000 input $u(t)$ and output $y(t)$ samples was generated and the training set (3.9), (3.10) consisting of input vectors U_t and corresponding outputs Y_t was produced.

$$U_t = \begin{bmatrix} y(999) & y(998) & & y(2) \\ y(998) & y(997) & \dots & y(1) \\ u(999) & u(998) & & u(2) \end{bmatrix}, \tag{3.9}$$

$$Y_t = [y(1000) \;\; y(999) \;\; \dots \;\; y(3)]. \tag{3.10}$$

This data set $\{U_t, Y_t\}$ was used to train the two-layer perceptron with three inputs and one output. The best accuracy of the model was achieved when the hidden layer consisted of 15 neurons. So, 60 synaptic weights and 16 biases (taken together 76 parameters) were adjusted by BP training algorithm (see section 2.6.1). After training the model is obtained in the following form

$$\hat{y}(t) = C \cdot \varphi(W \cdot [\hat{y}(t-1), \hat{y}(t-2), u(t-1)]) \tag{3.11}$$

where $\varphi(\cdot)$ is the hyperbolic tangent (2.51) activation function of the hidden layer neurons.

Another data set $\{U_v, Y_v\}$ was generated to validate the model. The network was simulated with inputs U_v and outputs calculated by the network $\hat{y}$ were compared with the corresponding outputs from the validation set Y_v. Sinusoidal input signal $u_v(t) = \sin(0.2 \cdot t)$ was used for model validation. The comparison of the system's output with the output of the model (neural network) is shown in figure 3.2.

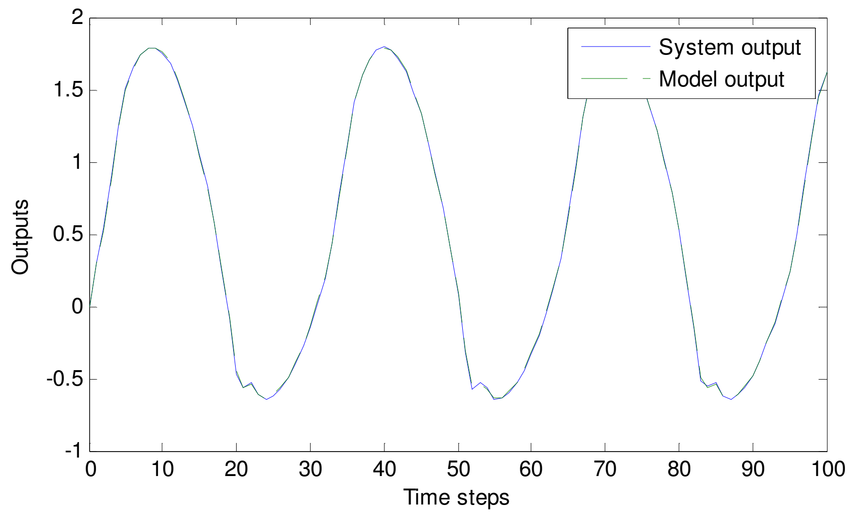

figure 3.2 System output vs. Model output

The system is simulated without noise in this experiment and it can be seen that the outputs coincide even when the model does not get any information from the real system. The mean square error of the model (2.18) is as low as $MSE \approx 1 \cdot 10^{-4}$. So, two-layer perceptron with external dynamics (feedback and delays) is capable of representing this nonlinear system with very high degree of accuracy. It is also capable of removing noise from the date set. That is why neural networks based models became so popular in nonlinear system identification and accurate enough to be used in model based control structures. Although NARX models based on multilayer perceptrons have very good approximation capabilities, practice shows that more specific structures depending on particular applications can also be of great value. When constructing a neural network one has freedom in choosing activation functions of neurons and in defining connections between neurons. For better identification quality of some specific nonlinear systems, a neural network with the structure that better represents the structure of the system can be used. Structure of a network can also depend on the needs of the control algorithm. Examples of such network structures will be considered in the following parts of this chapter.

A specific neural network structure for identification of nonlinear systems with actuator nonlinearities is proposed by the author of this thesis in [55].

3.4 Neural-network based Hammerstein model

One of the most common classes of models are Hammerstein models [65], [66] presented in figure 3.3.

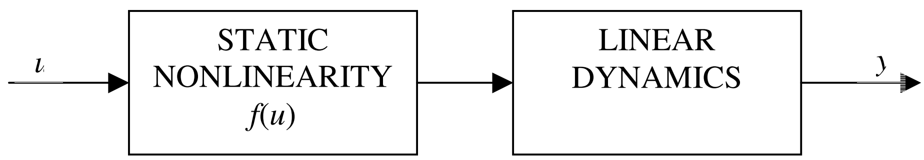

figure 3.3 Hammerstein model

Models of linear dynamic systems with static actuator nonlinearities belong to the class of Hammerstein models. Static actuator nonlinearities influence significantly the performance of adaptive and learning control systems. In many cases, systems under control can be approximated by linear dynamic functions with high precision and model-based control can be very easily implemented. Never the less, actuators make these systems nonlinear, because of inaccuracy of mechanical components and nature of physical laws. So, actuators introduce nonlinearity into system when it is considered as consisting of two parts: device that we have to control and an actuator. Static actuator nonlinearities such as saturation and rate limits degrade performance of control systems, especially model based control systems, when the control demand is high.[67]

While actuator nonlinearity is often present, most control design methods ignore them. This is particularly problematic in an adaptive control.[68] Actuator nonlinearities can not be canceled or compensated using feedback linearization techniques because they do not appear in the feedback path.[69] This is the main point why these nonlinearities can not be approximated using standard techniques. The accuracy of the models obtained by training standard single neural network is also poor. Dead zone and saturation are called *hard nonlinearities*.[69] This makes difficult to implement model-based control algorithms to systems with actuator nonlinearities. This problem was solved by the author of this thesis by introducing a special type of neural networks for identification of this type of nonlinearities in [55].

Systems having linear dynamic behavior with actuator devices having a static nonlinearities are considered. Another important fact that makes identification of such systems difficult is that the output of the actuator is not measurable. This is why the actuator and the system can not be identified separately.

In the above defined case, we have measurable actuator input and output of the system. The system between them consists of two parts: static nonlinear part and dynamic linear part. (see figure 3.4) Input nonlinearity is static and does not appear in the feedback. So, while the identification procedure it is important to separate these two parts.

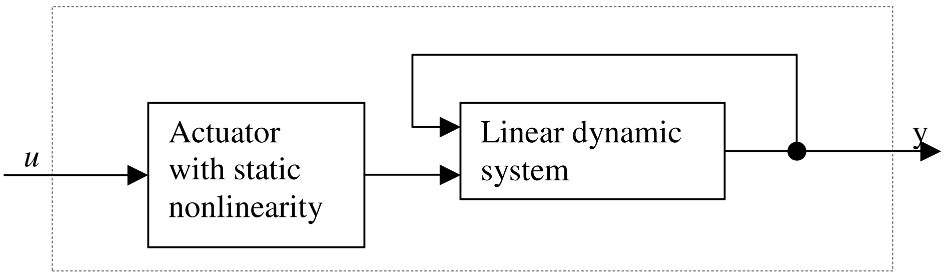

figure 3.4 System with static actuator nonlinearity

A neural network structure consisting of two sub-networks is proposed for identification of systems having this type of nonlinearities. The structure of the network is shown in figure 3.5. The network consists of two sub-networks. The first sub-network is a static feedforward network. It has one input and one output, but it has to approximate nonlinear function of actuator and therefore has nonlinear activation functions of the neurons in its hidden layer. According to Stone-Weirstrasse theorem (Theorem 2.1, section 2.5.4) it is capable of approximation any continuous nonlinearity of the actuator. The output of the first sub-network serves as the input of the second sub-network. The second sub-network is dynamic. It has external feedback from its output as well as past input and output values on its additional inputs as it is shown in figure 3.4. Number of additional inputs (memory length) depends on the order of the model. Second sub-network is used to identify linear dynamic part and therefore has linear activation functions of all of its neurons. It is capable of approximating any linear dynamic system. This network can be considered as the system consisting of two perceptrons that have to be trained together in order to separate linear dynamics and nonlinear static for better quality of identification of Hammerstein-type systems.

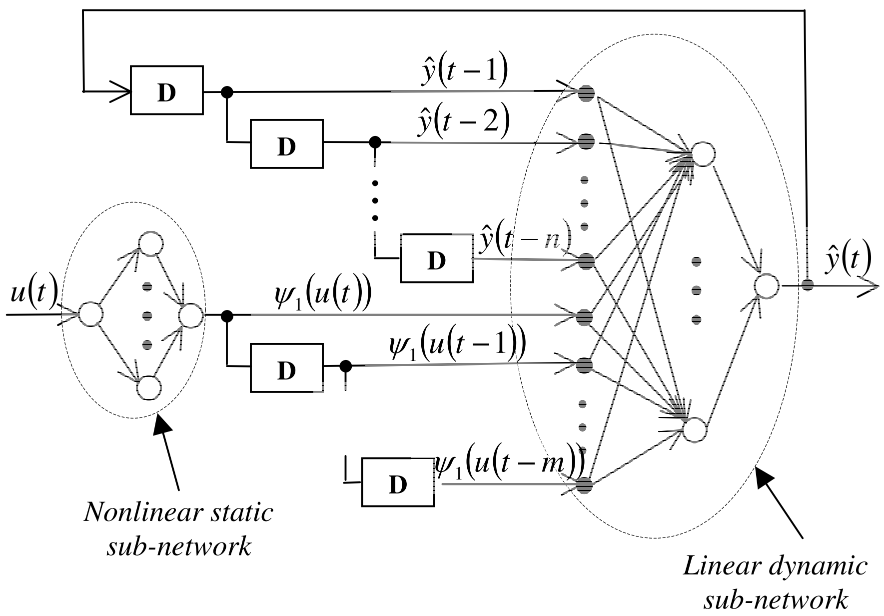

figure 3.5 Structure of neural network based Hammerstein model

The neural network based model shown in figure 3.5 belongs to the class of Hammerstein models. It can be formalized by

$$y(t) = \psi_2(y(t-1),\dots,y(t-n),\psi_1(u(t-1)),\dots,\psi_1(u(t-m))) \tag{3.12}$$

where Ψ_1 is the mapping performed by the nonlinear sub-network and Ψ_1 is the mapping performed by the linear sub-network. When two-layer perceptrons are used to perform these mappings, the model (3.12) is expressed by the following equation

$$\hat{y}(t) = C_2 \cdot W_2 \cdot [\hat{y}(t-1),\dots,\hat{y}(t-n),C_1\varphi(W_1 \cdot u(t-1)),\dots,C_1\varphi(W_1 \cdot u(t-m))]^T \tag{3.13}$$

where W_1 is the vector of synaptic weights of the hidden layer of the first (nonlinear) sub-network, C_1 is the vector of synaptic weights of the output layer of the first sub-network, W_2 is the matrix of synaptic weights of the hidden layer of the second (linear) sub-network, C_2 is the vector of synaptic weights of the output layer of the second sub-network and φ is a nonlinear activation function of the hidden layer neurons of the first sub-network.

Identification capabilities of this structure are demonstrated on the following examples.

3.4.1 Numerical example 3.2

Consider the following discrete-time linear plant [7]

$$H(z) = \frac{z+0.5}{z^2 - 1.5z + 0.7}. \tag{3.14}$$

A linear discrete-time system (3.14) with nonsymmetrical saturation and dead zone nonlinearities (3.15) on its input was used as a test system in [55].

Dead zone and saturation nonlinearities are defined by the following equations

$$s(u) = \begin{cases} s_{\max}, & if \quad K \cdot \left(u - u_{dz\_\max}\right) \geq s_{\max} \\ K \cdot \left(u - u_{dz\_\max}\right) & if \quad \left(K \cdot \left(u - u_{dz\_\max}\right) \leq s_{\max}\right) \& \left(u \geq u_{dz\_\max}\right) \\ K \cdot \left(u - u_{dz\_\min}\right) & if \quad \left(K \cdot \left(u - u_{dz\_\min}\right) \geq s_{\min}\right) \& \left(u \leq u_{dz\_\min}\right) \\ d & if \quad u_{dz\_\min} < u < u_{dz\_\max} \\ s_{\min}, & if \quad K \cdot \left(u - u_{dz\_\min}\right) \leq s_{\min} \end{cases}. \tag{3.15}$$

The parameters of the input nonlinearity (3.15) used in the experiment are $s_{\max} = 1.1$, $s_{\min} = -0.8$, $u_{dz\_\min} = -0.3$, $u_{dz\_\max} = 0.2$, $d = 0.1$ and $K = 1$.

This system consisting of linear plant and nonlinear actuator was simulated with a uniform distributed signal in the range $[-1.5,\ 1.5]$ and a data sequence of 1000 input-output samples $\{u(t), y(t)\}$ was generated to produce a training set.

First of all the model of the system (3.14), (3.15) was obtained by training a conventional two-layer perceptron based NARX network (3.11). The best accuracy of the model was obtained when the hidden layer consisted of 20 neurons with hyperbolic tangent (2.51) activation functions. Since the model is the second order SISO model (the order of the input is 2 and the order of the output is 2) NARX model has one external feedback from the output and the feedforward neural network has 4 inputs. It means that $4 \times 20 + 20 = 100$ synaptic weights had to be adjusted by a training algorithm. Because of big number of parameters to be turned, the network was trained by gradient descent error backpropagation algorithm (see section 2.6.1).

After training the network the model was tested on a data set consisting of 500 elements that was not used for training. Linearly growing in the range $[-3,\ 3]$ input signal $u_v(t) = -3 + 0.012t$, $t = 0,1,\ldots,500$ was used for generating the validation set $\{u_v(t), y_v(t)\}$. The result of model validation is shown in figure 3.6.

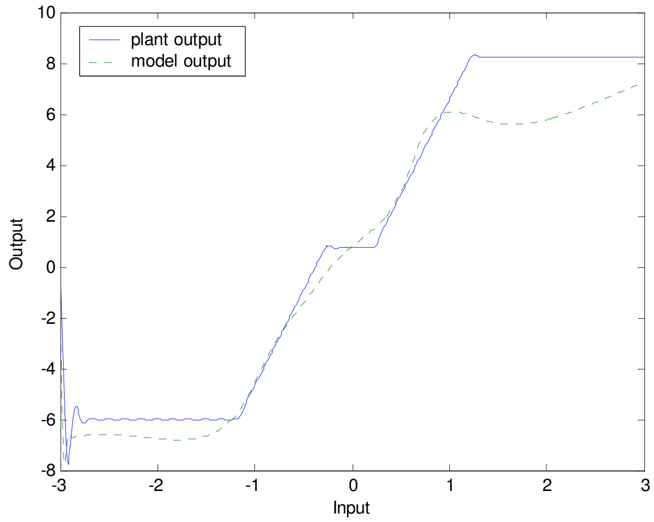

figure 3.6 Identification with single two-layer perceptron

Mean square error on the validation set in this experiment is $MSE \approx 1.749$.

After that the model was obtained by training the network (3.13) consisting of nonlinear static and linear dynamic sub-networks proposed in [55] and depicted in figure 3.5. The same training algorithm was applied.

The best accuracy was achieved by training the network with 6 nonlinear neurons in the hidden layer of the first sub-network and 3 neurons in the hidden layer of the second sub-network. Thus, the number of synaptic weights of the network is $6 + 6 + 4 \times 3 + 3 = 27$ that is almost 4 times less than in the previous experiment. The obtained model was validated on the same data set as the previous one. The validation result is depicted in figure 3.7. In case of using the proposed network structure for identification of this system $MSE \approx 0.013$.

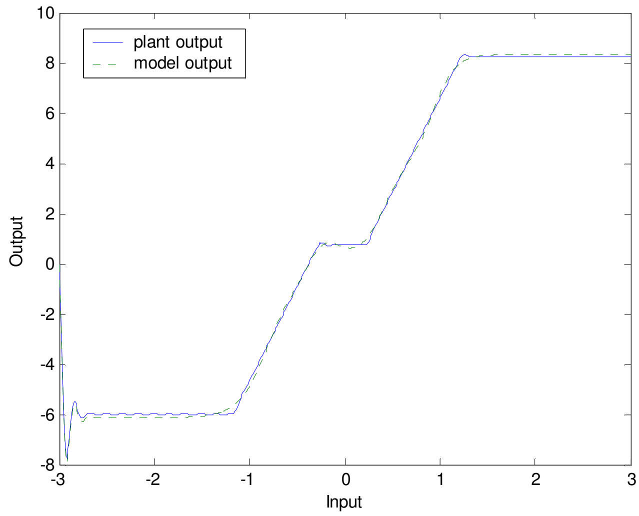

figure 3.7 Identification with the network separating nonlinear static and linear dynamic parts

Significant improvement in quality of identification can be seen by comparing mean square errors and figures 3.6 and 3.7. Much more precise model was obtained by training a neural network structure with almost 4 times smaller number of synaptic weights. This improvement can be explained by separating a complex model into two simple parts. So, there is no nonlinearities in the dynamic part and there is no dynamics in the nonlinear part. The proposed network structure can be of a great value for identification and model based control of systems with linear dynamic behavior and static nonlinearities in an actuator. In the next section implementation of this network structure for identification of a servo motor with nonlinear driver is shown.

3.4.2 Numerical example 3.3 – Application of neural-network based Hammerstein model to identification of direct current (DC) servo motor with nonlinear driver

The servo motor position control problem was formulated in [70]. The motor has bidirectional driver with nonlinear input/output characteristic having dead-zone and saturation

$$\begin{cases} if\ \left|V_i\right| \geq V_{dz},\ V_o = 0V \\ if\ \left(V_i > V_{dz}\right) \& \left(V_i < V_{\max}\right),\ V_o = (V_i - V_{dz}) \cdot gain \\ if\ \left(V_i < -V_{dz}\right) \& \left(V_i > -V_{\max}\right),\ V_o = (V_i + V_{dz}) \cdot gain \\ f\ V_i \geq V_{\max},\ V_o = V_{cc} \\ f\ V_i \leq -V_{\max},\ V_o = -V_{cc} \end{cases} . \qquad (3.16)$$

Here V_i is the input of the driver (control signal) and V_o is the output of the driver (input voltage of the motor). The driver has the following parameters: $V_{dz} = 0.25$, $V_{\max} = 0.5$, $gain = 40$, $V_{cc} = 10V$. DC motor has linear dynamics [71] and when the influence of the torque of the mechanical load is not considered it can be represented by the following second order state-space continuous-time model.

$$\frac{d}{dt}\begin{bmatrix} i_a \\ \omega_a \end{bmatrix} = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{k_v}{L_a} \\ \frac{k_t}{J} & -\frac{B}{J} \end{bmatrix} \begin{bmatrix} i_a \\ \omega_a \end{bmatrix} + \begin{bmatrix} -\frac{1}{L_a} \\ 0 \end{bmatrix} V_a$$
$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} i_a \\ \omega_a \end{bmatrix} + 0 \cdot V_a \tag{3.17}$$

Two inner states are the armature current i_a and the rotation velocity ω_a, one input is the voltage V_a and one output is the rotation velocity ω_a (second inner state). The parameters of this motor are as follows

$R_a = 1.75\Omega$ is the resistance of the motor armature;
$L_a = 2.83 \cdot 10^{-3} H$ is the inductance of the motor armature;
$k_v = 0.093 V\sec/rad$ is the velocity constant;
$k_t = 0.0924 Nm/A$ is the torque constant;
$J = 3 \cdot 10^{-5} kgm^2$ is the inertia seen by the motor. It includes the inertia of the load;
$B = 5 \cdot 10^{-3} Nms$ is the mechanical damping coefficient associated with rotation.

The model of the motor with nonlinear actuator (3.16), (3.17) was simulated. The work of this continuous-time system during 2 seconds was modelled. Since neural networks based models are discrete-time models, the data was sampled with 1*ms* intervals and the set of sampled input-output data was used for training the network of the proposed structure (3.13). As in the previous example the network was trained with 6 nonlinear neurons in the hidden layer of the first sub-network and 3 linear neurons in the hidden layer of the second sub-network. The obtained model was validated on the validation set consisting of input-output pairs $\{u_v(t), y_v(t)\}$, where $u_v(t)$ is a linearly growing in the range $[-1.5,\ 1.5]$ input signal $u_v(t) = -1.5 + 60t$, $t = 0,1,\ldots,50ms$ or in terms of discrete-time system with sample time $1ms$ $u_v(k) = -1.5 + 0.06k$, where $k = 0,1,\ldots,50$. The validation of the model is depicted in figure 3.8.

It can be seen from the figure that the output of the motor and the output of the proposed architecture based discrete-time model virtually coincide. The mean absolute error of this model is about 0.56*rad/sec* that is about 1% of the maximal velocity of the motor.

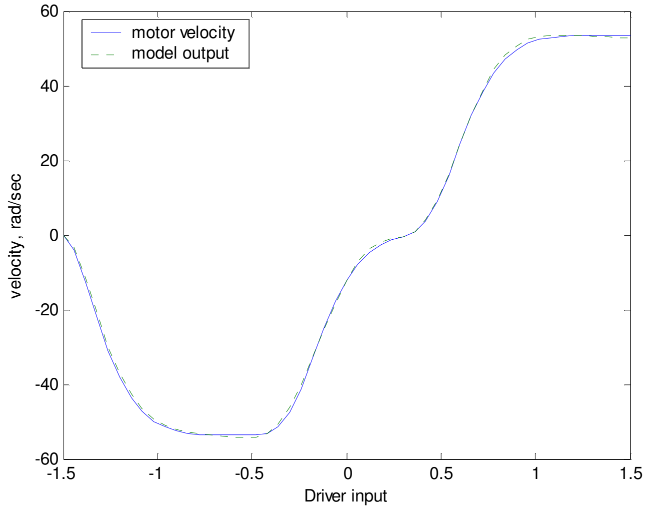

figure 3.8 Identification of the system consisting of the DC motor and nonlinear bidirectional driver by two sub-networks

Applying the proposed neural network structure instead of conventional multi-layer perceptron based NARX models to identification of systems with static input nonlinearities and linear dynamics can significantly improve the quality of model based control schemes. It is especially important when the control demand is high.

Here one of possibilities for improving the quality of models used in model based control is shown for one class of nonlinear systems. Implementation of these models in model based control schemes will be section 4.2.2 of this thesis. In the next section of this chapter a neural network structure constructed according to the requirements of the control algorithm will be shown.

3.5 Neural Networks based ANARX models

It is proofed in [32] that ANARX model (2.13) is linearizable by using dynamic output feedback linearization algorithm (2.36)-(2.38). It is also shown in [8], [9] that ANARX model is always realizable in a state-space form. A neural networks based structure for representing the ANARX model was shown in [8], [10], [11] and used for identification and control of nonlinear systems in [13] and [14]. This model is called *Neural Networks Based Additive NARX Model* (*or*

NN-ANARX). The structure of the network for ANARX model representation is depicted in figure 3.9.

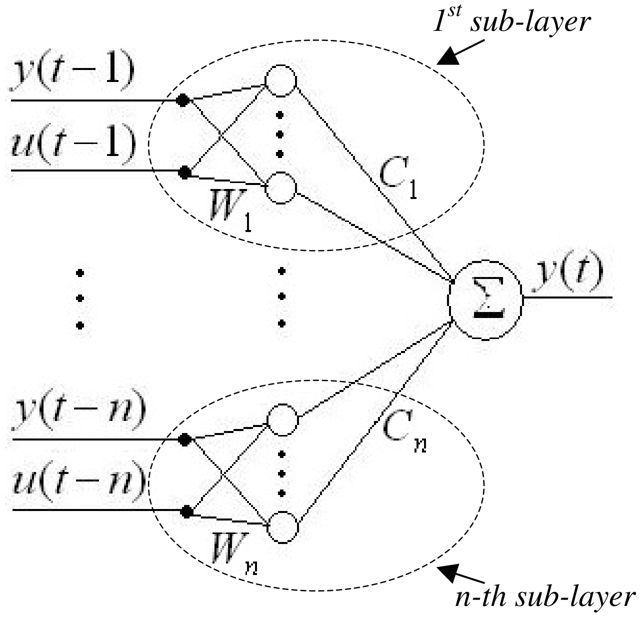

figure 3.9 Additive NARX model represented by a neural network

This network consists of output layer with ADALINE (Adaptive Linear Neuron) type neurons in it and a hidden layer consisting of n parallel sub-layers corresponding to the *n-th* order of the model. Each i-th sub-layer approximates function f_i from (2.13). This neural network based model can be formalized by using the following equation

$$y(t) = \sum_{i=1}^{n} C_i \varphi_i \left(W_i \cdot Z(t-i) \right) \qquad (3.18)$$

where $Z(t) = [y(t), u(t)]^T$, $\varphi_i(\cdot)$ is an activation function of i-th sub-layer neurons, C_i is the matrix of i-th sub-layer output synaptic weights, W_i is the matrix of *i-th* sub-layer input synaptic weights.

This neural network has restricted connectivity. Thus if *i-th* sub-layer of the hidden layer consists of l_i neurons then the number of parameters to be adjusted by a training algorithm is

$$size(W_1, \ldots, W_n, C_1, \ldots, C_n) = 3\sum_{i=1}^{n} l_i \, , \qquad (3.19)$$

that is much smaller than in case of a fully connected multilayer perceptron with $l_1 + \ldots + l_n$ neurons in the hidden layer. For example, if a two-layer perceptron with 9 neurons in the hidden layer is used for identification of a 3-rd order model ($n = 3$), it has 6 inputs $[y(t-1), u(t-1), y(t-2), u(t-2), y(t-3), u(t-3)]$ and thus $6 \times 9 + 9 = 60$ synaptic weights has to be adjusted. When NN-ANARX model (3.18) with 3 neurons in each of 3 sub-layers is used for identification of the same system then according to equation (3.19) only $3 \times 9 = 27$ synaptic weights has to be adjusted. This fact makes possible to use more precise second order algorithms for training of the network. For example, Levenberg-Marquardt training algorithm can be used instead of Gradient Descent Error Backpropagation as it was shown in [13], [14].

3.5.1 State-Space Realization by using NN-ANARX

ANARX structure is always realizable in the classical state-space form (2.14). When the model is obtained in the form of a neural network (3.18) by training the network of the structure shown in figure 3.9, the state-space realization can be formulated by using matrices of synaptic weights of the network [8] as

$$\begin{aligned}
&x_1(t+1) = x_2(t) + C_1\varphi_1\left(W_1\left(x_1(t), u(t)\right)^T\right)\\
&x_2(t+1) = x_3(t) + C_2\varphi_2\left(W_2\left(x_1(t), u(t)\right)^T\right)\\
&\vdots\\
&x_{n-1}(t+1) = x_n(t) + C_{n-1}\varphi_{n-1}\left(W_{n-1}\left(x_1(t), u(t)\right)^T\right)\\
&x_n(t+1) = C_n\varphi_n\left(W_n\left(x_1(t), u(t)\right)^T\right)
\end{aligned} \qquad (3.20)$$

where W_i and C_i are matrices of input and output synaptic weights of each sub-layer obtained by a training algorithm and $\varphi_i(\cdot)$ are nonlinear activation functions. State-space model of *n-th* is corresponding to each *n-th* order ANARX and NN-ANARX model.

In the following three numerical examples identification and state-space realization of nonlinear SISO systems by training neural networks of the corresponding structure will be shown.

3.5.2 Numerical example 3.4

The model of a liquid level system of interconnected tanks [72] is represented by the following input-output equation

$$
\begin{aligned}
y(t+3) &= 0.43y(t+2) + 0.681y(t+1) - 0.149y(t) + 0.396u(t+2) + \\
&+ 0.014u(t+1) - 0.071u(t) - 0.351y(t+2)u(t+2) - 0.03y^2(t+1) - \\
&- 0.135y(t+1)u(t+1) - 0.027y^3(t+1) - 0.108y^2(t+1)u(t+1) - \\
&- 0.099u^3(t+1)
\end{aligned} \tag{3.21}
$$

Identification of this system by training NN-ANARX structure is demonstrated by the author of this thesis in [14].

Equation (3.21) was used as an unknown nonlinear plant which is modeled by the NN-based ANARX structure

$$
y(t+3) = \sum_{i=1}^{3} C_i \varphi_i \left(W_i \cdot \left[y(t+i-1), u(t+i-1) \right]^T \right) . \tag{3.22}
$$

To obtain input-output data, the plant (3.21) was simulated with sinusoidal input signal $u(t) = \sin(0.05t) + 0.05e(t)$, where noise $e(t)$ is normally distributed with mean 0 and variance $\sigma^2 = 1$: $e(t) \sim N(0,\ 1^2)$. Neural network with three sub-layers (the order of the model $n = 3$) and with three neurons on each sub-layer of the hidden layer ($l_1 = l_2 = l_3 = 3$) with logistic sigmoid activation functions (2.50) was trained.

The LM training algorithm (see section 2.6.2) was chosen to perform off-line training since it is much more efficient compared to other techniques when the network contains no more than a few hundred weights [53], and in our case according to equation (3.19) the neural network has just 27 synaptic weights. Also the training speed of LM algorithm is much higher and the feed forward neural network trained with it can better model the nonlinearity [54]. Training took about 600 iterations (epochs) to converge. Identified parameters of the model (3.22) have the following values

$$
W_1 = \begin{bmatrix} 0.9132 & 0.2356 \\ 0.3809 & -1.3663 \\ 0.8472 & 0.0455 \end{bmatrix},
$$

$$
W_2 = \begin{bmatrix} -0.2946 & -4.4628 \\ 1.2427 & -0.5516 \\ 1.2093 & -0.6216 \end{bmatrix},
$$

$$
W_3 = \begin{bmatrix} 1.2788 & -3.0221 \\ 0.4240 & 0.4244 \\ 0.4339 & 0.5441 \end{bmatrix}, \tag{3.23}
$$

$C_1 = [-24.7605 \quad -4.5011 \quad 34.0981],$
$C_2 = [-0.2364 \quad -26.4052 \quad 27.2547]$
$C_3 = [1.6433 \quad -71.6573 \quad 64.4851]$

For validation of the model data set $\{u_v(t), y_v(t)\}$ was generated by using an input signal with two times higher frequency: $u_v(t) = \cos(0.1t) + 0.05e(t)$. The corresponding outputs of system (3.21) and model (3.22) with parameters (3.23) are depicted in figure 3.10.

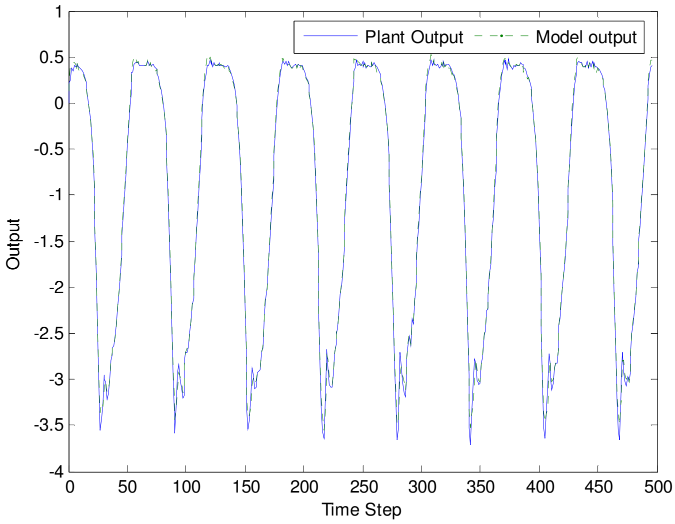

figure 3.10 Validation of the NN-ANARX structure based model of the system of interconnected tanks

Model validation shows nearly excellent overlap of the model and the plant outputs. Mean square error on the validation set was as low as $MSE \approx 4 \cdot 10^{-3}$. It can be seen that this system can be represented by NN-ANARX model with high degree of accuracy.

In the next example will be shown identification and state-space representation of the system which input-output model does not have an ANARX structure. It can be done by training a neural network having ANARX architecture shown in figure 3.9.

3.5.3 Numerical example 3.5

The second order discrete time model of a jacketed Continuous Stirred Tank Reactor (CSTR) [74], [75], [13], [14] is represented by the following input-output equation.

$$\begin{aligned} y(t+2) &= 0.7653y(t+1) - 0.231y(t) + 0.4801u(t+1) - 0.6407y^2(t+1) + \\ &+ 1.014y(t)y(t+1) - 0.3921y^2(t+1) + 0.592y(t+1)u(t+1) - \\ &- 0.5611y(t)u(t+1) \end{aligned} \tag{3.24}$$

Note that, because of the last term in (3.24), the model does not have an ANARX structure. To obtain input-output data from the "unknown" plant, equation (3.24) was simulated with sinusoidal input signal $u(t) = \sin(0.05t) + 0.05e(t)$, where $e(t) \sim N(0,\ 1^2)$. This input signal and corresponding output signal were used as a training set for identification by training a neural network based ANARX structure (3.18). This plant was modeled by NN-ANARX model with two sub-layers corresponding to the second order of the model (n = 2) and three neurons with logistic sigmoid activation functions (2.50) on each sub-layer of the hidden layer ($l_1 = l_2 = 3$). Thus 18 parameters were adjusted by LM training algorithm that took about 600 epochs to converge and the model having the following equation

$$y(t+2) = C_1\varphi_1\left(W_1 \cdot [y(t+1), u(t+1)]^T\right) + C_2\varphi_2\left(W_2 \cdot [y(t), u(t)]^T\right) \tag{3.25}$$

was obtained. The parameters of the model (3.25) have the following values

$$W_1 = \begin{bmatrix} 0.1793 & 0.2616 \\ -23.0273 & 22.2099 \\ -0.0785 & -0.4576 \end{bmatrix},\ W_2 = \begin{bmatrix} -23.4344 & 29.5285 \\ 1.5266 & -2.0768 \\ -1.0887 & 1.6082 \end{bmatrix} \tag{3.26}$$
$$C_1 = [31.9678\ \ -0.0166\ \ 13.8712],\ C_2 = [-0.0128\ \ -19.8288\ \ -25.9744]$$

For validation of the model data set $\{u_v(t), y_v(t)\}$ was generated by using an input signal with two times higher frequency: $u_v(t) = \cos(0.1t) + 0.05e(t)$. The corresponding outputs of system (3.24) and model (3.25) with parameters (3.26) are depicted in figure 3.11.

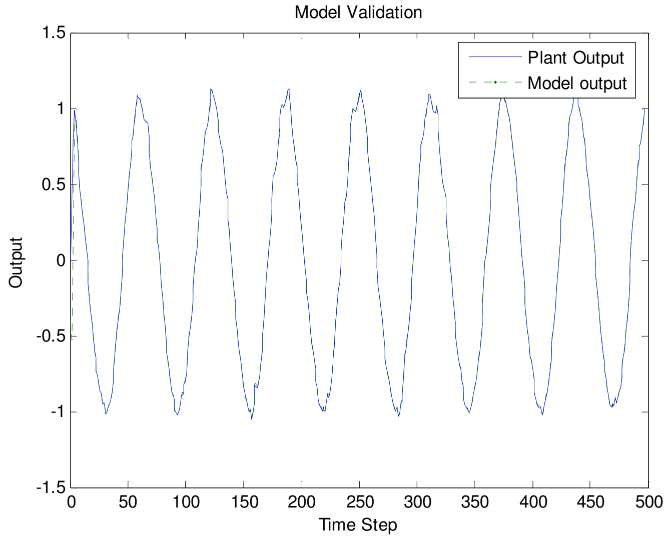

figure 3.11 Validation of the NN-ANARX structure based model of the jacketed CSTR

Model validation (see figure 3.11) shows that, in spite of the restrictions imposed by NN-ANARX structure, the identified model explains input-output data with high degree of accuracy. It can be seen from the figure that system and model outputs virtually coincide. The mean square error was as low as $MSE \approx 3.5 \cdot 10^{-3}$.

While the structural restriction imposed by NN-ANARX on connectivity matrix structure could seem too strong, training results [13], [14], [76] show that in many cases NN-based ANARX model is able to represent original model with high degree of accuracy and does not cause serious drawbacks in quality of identification.

According to (3.20), input-output model (3.24) can now be represented in the form of state-space model. The state-space model can be formally written down as

$$\begin{aligned} x_1(t+1) &= x_2(t) + C_1\varphi_1\left(W_1\left(x_1(t),u(t)\right)^T\right) \\ x_2(t+1) &= x_3(t) + C_2\varphi_2\left(W_2\left(x_1(t),u(t)\right)^T\right) \end{aligned} \tag{3.27}$$

where W_1, W_2, C_1 and C_2 are parameters (3.26) of the network (3.25) representing the system in the form of NN-ANARX model; φ_1 and φ_2 are logistic sigmoid activation functions (2.50) of the corresponding sub-layers of the hidden layer.

3.6 NN-ANARX based Hammerstein model

In section 3.4 of this thesis a neural network structure consisting of two fully connected multilayer perceptrons is proposed for representation of Hammerstein-type systems – systems having linear dynamics and nonlinear static input nonlinearity (see figure 3.3). The proposed network (3.12) depicted in figure 3.5 is capable of separating static nonlinearity and linear dynamics. The same approach can be used for developing a neural network based Hammerstein model that belongs to the class of ANARX models.

A neural network (3.18) of the structure shown in figure 3.9 with all linear neurons is used for identification of linear dynamic part of the model. ADALINEs has to be used instead of nonlinear neurons in the dynamic part. A static neural network with one input and one output and nonlinear activation functions of the hidden layer nonlinearity is used for identification of input nonlinearity. The output of this network is then passed to the sub-layers of the linear ANARX network with corresponding delays. The structure of the network is presented in figure 3.12.

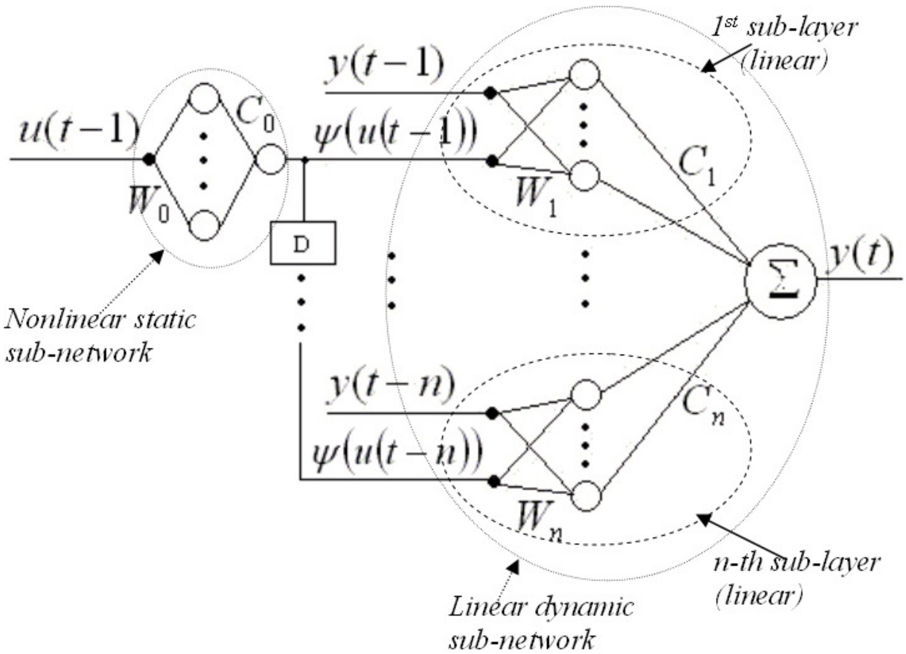

figure 3.12 NN-ANARX based Hammerstein model

The model proposed here can be formalized as

$$y(t) = \sum_{i=1}^{n} C_i \left(W_i \cdot \left[y(t-i), \psi(u(t-i)) \right]^T \right), \tag{3.28}$$

where $\psi(u(t))$ is the function of the first nonlinear static neural network

$$\psi(u(t)) = C_0\varphi(W_0 \cdot u(t)). \tag{3.29}$$

Here $\varphi(\cdot)$ is a nonlinear activation function of the hidden layer neurons of the static sub-network, $W_0 \in \Re^{l_0 \times 1}$ and $C_0 \in \Re^{1 \times l_0}$ are vectors of synaptic weights of the static sub-network, $W_i \in \Re^{l_i \times 2}$ and $C_i \in \Re^{1 \times l_i}$ where $i = 1, \ldots, n$ are matrices and vectors of the linear dynamic sub-network. l_i, $i = 0, \ldots, n$ are the numbers of neurons in the corresponding sub-layers. Thus from equations (3.28) and (3.29) we get

$$y(t) = \sum_{i=1}^{n} C_i \left(W_i \cdot \left[y(t-i), C_0\varphi(W_0 \cdot u(t-i))\right]^T\right). \tag{3.30}$$

As $C_i \in \Re^{1 \times l_i}$ and $W_i \in \Re^{l_i \times 2}$, vectors of two coefficients $G_i = [g_{i1} \quad g_{i2}]$ can be defined for each sub-layer of the linear dynamic ANARX sub-network as

$$G_i = [g_{1i} \quad g_{2i}] = C_i \cdot W_i, \ \forall i = 1, \ldots, n. \tag{3.31}$$

By using these coefficients NN-ANARX based Hammerstein model can now be formalized as

$$y(t) = \sum_{i=1}^{n} g_{1i} \cdot y(t-i) + g_{2i} \cdot C_0 \cdot \varphi(W_0 \cdot u(t-i)). \tag{3.32}$$

This model belongs to the both classes: to the class of neural networks based ANARX models (3.18) as it has all time instances separated and to the class of Hammerstein models (see figure 3.3) as it has linear dynamics and nonlinear static input. This model is capable of representing systems with static input nonlinearities (for example, with nonlinear actuators) with high degree of accuracy. Because of training neural networks based model it is capable of modeling systems with unknown static input nonlinearities by automatically separating linear dynamics and nonlinear static during the training procedure. Because of properties of ANARX models (2.14), the n-th order model (3.32) can be very easily represented in the classical state-space form suitable for the variety of control algorithms as

$$
\begin{aligned}
&x_1(t+1) = x_2(t+1) + g_{11} \cdot y(t) + g_{21} \cdot C_0 \cdot \varphi(W_0 \cdot u(t)) \\
&x_2(t+1) = x_3(t+1) + g_{12} \cdot y(t) + g_{22} \cdot C_0 \cdot \varphi(W_0 \cdot u(t)) \\
&\vdots \\
&x_n(t+1) = g_{1n} \cdot y(t) + g_{2n} \cdot C_0 \cdot \varphi(W_0 \cdot u(t)) \\
&y(t) = x_1(t)
\end{aligned} \tag{3.33}
$$

In order to compare different approaches of neural networks based modeling of Hammerstein-type systems consider the following example.

3.6.1 Numerical example 3.6

In [11] NN-ANARX based Hammerstein model (3.32) was used for identification of direct current (DC) servo motor with nonlinear driver (3.16)-(3.17) and was compared to the approach proposed in [51] which is based on training of two perceptrons (3.12)-(3.13) discussed in section 3.4 of this thesis.

A neural network with linear dynamic sub-network having two sub-layers of its hidden layer ($n = 2$, $l_1 = l_2 = 3$) corresponding to the second order of the model was trained. Six nonlinear hidden layer neurons were used in the static part of the model ($l_0 = 6$). Thus only $6 + 6 + 2 \cdot 2 \cdot 3 + 3 \cdot 2 = 30$ synaptic weights had to be adjusted by a training algorithm.

Both NN-NARX based Hammerstein model shown in numerical example 3.4 and NN-ANARX based model discussed in this section were trained on the same data set and validated on the input signal growing linearly from -1.5 to +1.5. The corresponding outputs of the system and corresponding ANARX model are depicted in figure 3.13.

The mean absolute error of this NN-ANARX based Hammerstein model of the DC motor and nonlinear bidirectional driver is about 0.8 *rad/sec* which is about less than 2% of the maximal velocity of the motor in the framework of present experiment. As quality of identification depends in a sense on the particular training experiment (depends on random initial values of the synaptic weights) it can be said that the accuracy of this model is about the same as in case of neural networks based NARX Hammerstein model discussed in section 3.4.

It can also be seen by comparing figures 3.13 and 3.8 that both models are capable of representing this system with nonlinear actuator with high degree of accuracy and restrictions imposed by the ANARX structure on the topology of the network do not cause serious decrease in the quality of identification.

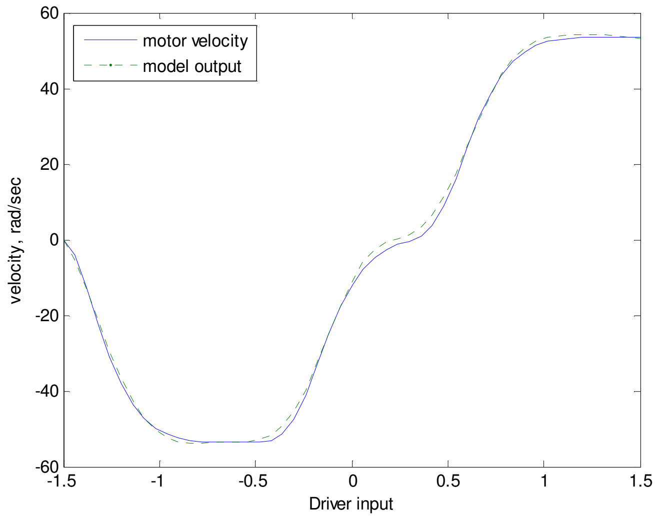

figure 3.13 Identification of the system consisting of the DC motor and nonlinear bidirectional driver by NN-ANARX model

At the same time when application specific neural network structures (3.12)-(3.13) or (3.16)-(3.17) are used for modeling of systems with static input nonlinearities, the quality of identification is much higher when compared with identification by training a single multilayer perceptron. So, in spite of the fact that according to the Stone-Weierstrass theorem (see section 2.5.4 of this thesis) two-layer perceptrons are capable of approximating any continuous nonlinearities and by adding an external feedback can be used for identification of nonlinear dynamic systems, practically implementation of application specific neural network structures proposed by the author of this thesis in [11] and [51] gives significant improvement in the quality of identification of dynamic systems with static input (actuator) nonlinearities.

Both NN-NARX and NN-ANARX Hammerstein models give about the same good quality of identification. The main advantage of using NN-NARX based Hammerstein models is in more simple structure of the model, but on the other hand NN-ANARX based Hammerstein models with the same number of neurons as NN-NARX network has smaller number of synaptic weights because of constrained connectivity. This fact makes training faster, allows using more complicated training algorithms and possible fast adapting of the model in real time adaptive schemes. One more advantage of NN-ANARX based Hammerstein models is that they are linearizable by dynamic output feedback (2.36)-(2.38) and are reprezentable in the classical state-space form (3.33) that makes them suitable for a wider range of model based control algorithms.

3.7 Models of Nonlinear MIMO Systems

Artificial neural networks is also a very convenient tool for identification of Multiply Input – Multiply Output (MIMO) systems. Both neural networks based NARX and ANARX models of nonlinear MIMO systems will be considered in this section.

3.7.1 MIMO NARX and MIMO NN-NARX models

It is known from literature (e.g. [77]) that a wide class of nonlinear MIMO systems can be represented by the nonlinear discrete model in input-output description known in literature as NARX models, as series-parallel-model, or as one-step ahead predictor.

Let $U(t)=[u_1(t),\ldots,u_r(t)]\in\Re^r$ is a vector of system inputs and $Y(t)=[y_1(t),\ldots,y_m(t)]\in\Re^m$ is a vector of system outputs at time step *t*, *r* is the number of inputs and *m* is the number of outputs of the model. Then MIMO NARX model can be defined as

$$Y(t)=f(Y(t-1),\ldots,Y(t-n_y),U(t-1),\ldots,U(t-n_u)),\qquad(3.34)$$

where n_u and n_y are the input and output order and $f(\cdot)$ is a nonlinear function of n_u+n_y arguments.

It is very convenient to obtain model (3.34) by training a multilayer perceptron approximating nonlinear function $f(\cdot)$. In order to obtain a of a dynamic MIMO system, external feedbacks from each output of the model $y_1(t),\ldots,y_m(t)$ are needed as well as delayed values of all inputs $u_1(t),\ldots,u_r(t)$. The structure of the corresponding neural networks based MIMO NARX model is depicted in figure 3.14.

According to Stone-Weierstrass theorem (theorem 2.1, section 2.5.4) any continuous nonlinear function $f(\cdot)$ can be approximated by a two-layer perceptron. So, a two-layer perceptron with external feedbacks and delays can also be used for identification of nonlinear MIMO systems. Models based on this structure are called neural network based NARX model or NN-NARX models.

MIMO NARX model obtained by training a two-layer perceptron has two matrixes of synaptic weights as its parameters. The first one is an input matrix of synaptic weights $W\in\Re^{(r\cdot n_u+m\cdot n_y)\times l}$, where l is the number of neurons in the hidden layer, and the second one is an output matrix $C\in\Re^{m\times l}$.

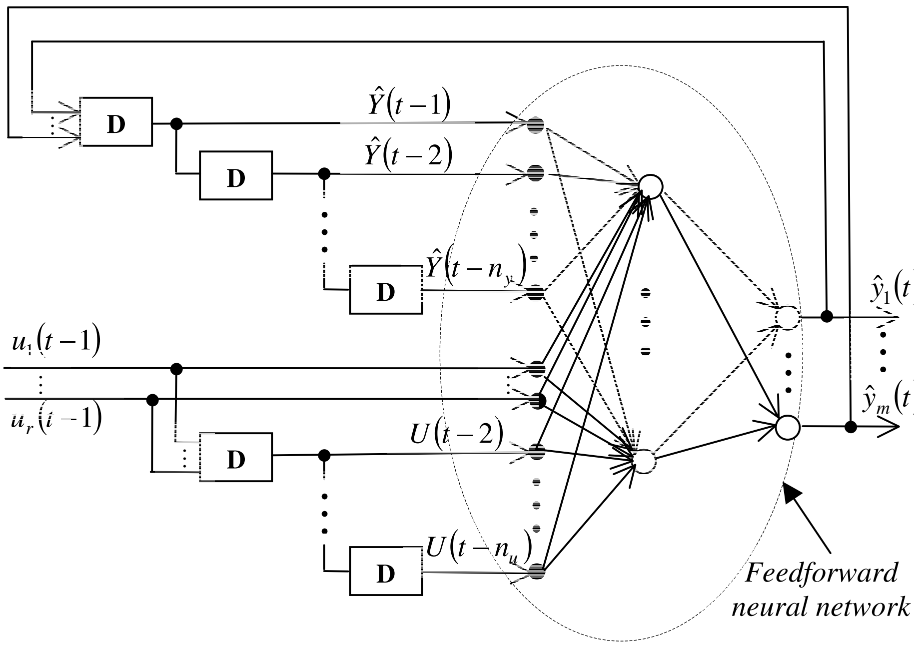

figure 3.14 Neural network based representation of MIMO NARX model

When the network is trained and optimal values of its synaptic weights are obtained, the vector of one-step ahead predictors $Y(t)$ can be calculated by using the following equation:

$$Y(t) = C \cdot \varphi\left(W \cdot \left[Y(t-1), \ldots, Y(t-n_y), U(t-1), \ldots, U(t-n_u)\right]^T\right) \tag{3.35}$$

In case of this model, the number of parameters that we have to calculate during the training of the network grows dramatically by increasing the number of outputs and/or inputs and/or the order of the model. It makes practical identification very slow or even impossible. Experiments show that in some cases training algorithms may not converge because of a huge number of adjustable parameters (synaptic weights). For example, when two-layer perceptron with 9 neurons in its hidden layer is used for obtaining a third order model of MIMO system having 2 inputs and 2 outputs, values of 126 synaptic weights have to be calculated by a selected training algorithm.

To bridge the gap, a MIMO NN-ANARX model was proposed by the author in [56].

3.7.2 MIMO ANARX and MIMO NN-ANARX models

ANARX model is a subclass of NARX models having all time instances separated. For MIMO systems ANARX model can be defined as

$$Y(t) = \sum_{i=1}^{n} f_i\left(Y(t-i), U(t-i)\right), \tag{3.36}$$

where n is the order of the model and $f_1(\cdot), \ldots, f_n(\cdot)$ are nonlinear functions.

Neural Networks based ANARX model for MIMO systems is defined in [56] as

$$Y(t) = \sum_{i=1}^{n} C_i \cdot \varphi_i\left(W_i \cdot \left[Y(t-i), U(t-i)\right]^T\right), \tag{3.37}$$

where $\varphi_1(\cdot), \ldots, \varphi_n(\cdot)$, $W_1, \ldots, W_n$, $C_1, \ldots, C_n$ are nonlinear activation functions and matrixes of synaptic weights of the sub-layers of the network structure shown in figure 3.15.

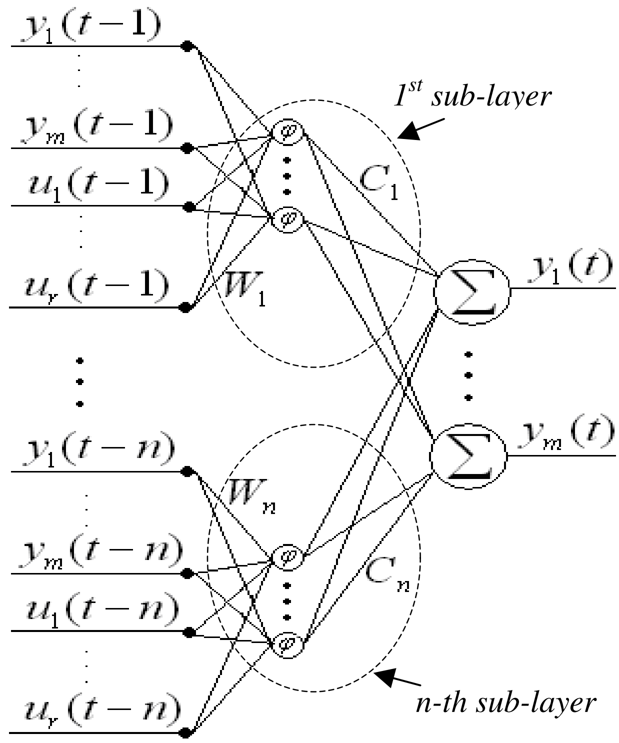

figure 3.15 Structure of neural network representing ANARX model for MIMO system

Matrixes of synaptic weights of this model have the following dimensions: $W_i \in \Re^{l_i \times (m+r)}$ and $C_i \in \Re^{m \times l_i}$, where l_i is the number of neurons in the i-th sub-layer ($i = 1, \ldots, n$).

To compare this structure with NN-NARX models discussed in the previous section, lets also consider a third order ($n = 3$) MIMO system having 2 inputs and two outputs ($m = r = 2$) obtained by training a neural network with 9 neurons in its hidden layer ($\sum_{i=1}^{3} l_i = 9$). In case of neural network based MIMO ANARX model (3.37), structure of the corresponding neural network shown in figure 3.15 should consist of 3 sub-layers. 3 neurons may be used in each sub-layer for 9 neurons in the hidden layer ($l_1 = l_2 = l_3 = 3$). So, by training this structure only 54 synaptic weights have to be calculated. It is more than 2 times less comparing to MIMO NN-NARX structure with the same number of inputs, outputs and neurons.

Experiments have shown that strong restrictions in the connectivity of the network's structure do not cause drawbacks in quality of identification. Even more, it gives some improvement, because the number of connections does not become too large. Smaller number of parameters makes training faster and allows implementing wider range of training algorithms. For example, second order training algorithms like LM-algorithm (see section 2.6.2).

Problems concerning MIMO NN-ANARX structure based control of nonlinear MIMO systems will be considered in detail in Chapter 5 of this thesis.

For better precision of the model in case of identification of complex multidimensional processes, a MIMO model can be replaced by a set of *r* MISO models, where *r* is the number of outputs of the process or by several MIMO systems where outputs are divided between several separate models. This approach was implemented by the author of the thesis for modeling of the surgeon's hand. This project will be discussed in the following example.

3.7.3 Numerical example 3.7 – NN-based ANARX model of the surgeon's hand for the motion recognition and movement prediction

The main goal of the project is executing the learning and adapting capabilities of the scrub nurse robot to be able to anticipate surgeon's requests during surgical operation and to react them with proper assisting action. One of very important parts of the robot's "brain" is the model of the surgeon's hand which is necessary for predicting coordinates of the surgeon's wrist and determining type of motion. A set of MIMO NN-ANARX structures was used to obtain the model. This results are published in [73]. The structure of the data flow of the scrub nurse robot is depicted in figure 3.16 and the project made by the author of this thesis is shown in the scheme by the dashed oval.

First of all, let me give a short overview of this project [73]. One of the recent trends in development of the medical robots can be seen as close interaction with humans. Such close interaction causes the necessity to detect human actions and

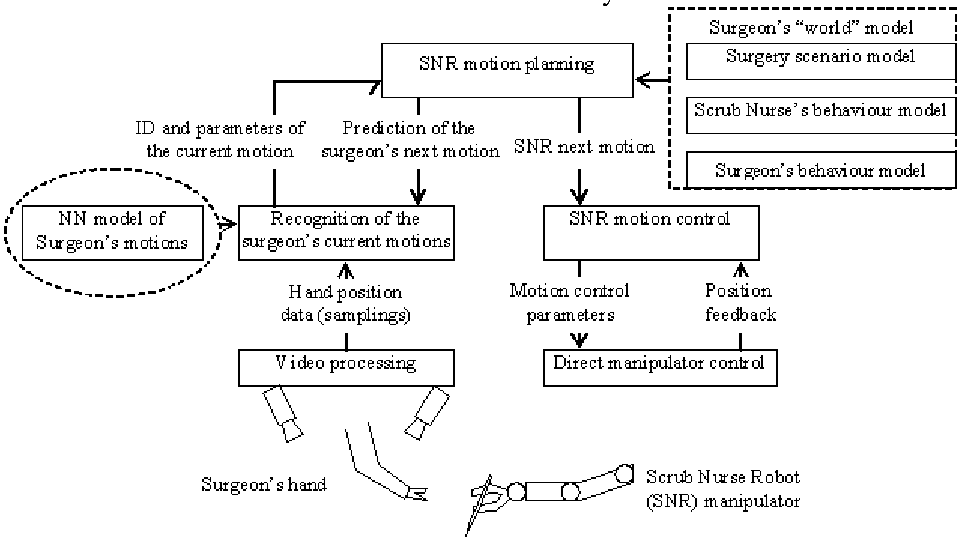

figure 3.16 Control data flow of the scrub nurse robot [73]

model human behavior.[78] In [79] main attention was made in building entire model of a scrub nurse which includes extended timed automata to model cooperative actions of all the surgical staff and dynamic models describing physical motion of the surgical staff.

The main aim of scrub nurse robot (SNR) is to recognize Surgeon's intentions and to provide the assistance he expects from SNR for that. The assistance may be just holding a surgical instrument but in more complicated cases it is a sequence of actions including, e.g., gasping an instrument from the tray, waiting, passing it to the Surgeon, receiving another instrument and putting it back on the tray, etc. Human adoption of the SNR means here not only right positioning of instruments and right timing of motions but also adjusting acceleration and trajectory of motions to the personal liking of a Surgeon. The only practical way of achieving human adoption in given sense is learning from a skilful scrub nurses and imitating their behavior in similar situations. Still, right behavior of a scrub nurse depends also on contextual knowledge not observable directly: how to behave in emergency situations, what is the agreed scenario of the surgical procedure, in what order the instruments are placed on the surgical tray etc. When the SNR is learning from the passive observation of surgical procedures some of the background knowledge should be already there to put the observation data into right context. Thus, a "model of world" should be a central knowledge unit of SNR. In fact, the "model of world" is a set of models partially hard-coded during design, partially created and instantiated by on-line learning. Currently we accumulate five types of knowledge in the robot world model: surgery scenario, scrub nurse's and surgeon's reactive behavior, as well as the (continuous) motion models of both. When the first three are encoded as extended timed automata models [80], the last two are neural network models discussed in more details below in this section of the thesis. For making control decisions the robot ”world model” is exploited together with on-line observation data. To implement the SNR features referred above, the robot control architecture with data flow depicted in Figure 3.16 is proposed. According to the given architecture the control loop includes video monitoring and signal processing, recognition of Surgeon's current motion, choosing the SNR's appropriate reaction, and forwarding the control command with necessary parameters to manipulator control unit.

Author's contribution is devoted to the modeling of the surgeon's hand movement by Neural Networks based ANARX model. Two main problems concerning the dynamics of the surgical staff movements that can be solved by using this model are

1. to detect current surgeon's motion (detect his current action)
2. to predict the coordinates of the point where instrument exchange should take place

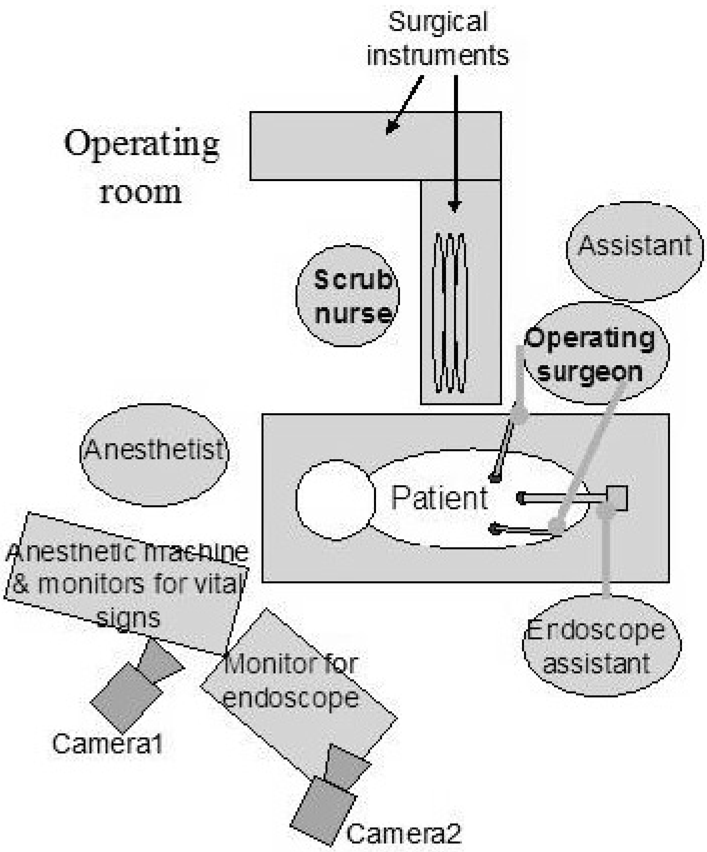

figure 3.17 Schematic diagram of the operating room [73]

Obtaining a model of the surgeon's movement means constructing an input-output model where the input is the vector of observed coordinates of the surgeon's chest, elbow and wrist and the output is the three dimensional vector which elements are prediction of the chest, elbow and wrist coordinates in certain number of time-steps.

In order to obtain experimental data from the real surgical operation the operating room was equipped with two cameras. Positions of medical staff during the surgical operation and placement of the cameras depicted in figure 3.17. In order to simplify monitoring process colour markers were placed on the surgeon's chest, elbow and wrist. See figure 3.18.

figure 3.18 Operating room [73]

The cameras were set to film the operation at 30 frames per second. Later on the basis of recorded operation information about the coordinates of chest, elbow and wrist was extracted. The sampling rate of gathered data is 0.033 sec. Measured data was then stored in MS Excel file for training the models. A short example of the data sequence (2 sec) used for training is shown in figure 3.19.

	A	B	C	D	E	F	G	H	I	J	K
1	Time [sec]	chest: X [cm]	right_elbow	right_wrist: X	chest: Y [cm]	right_elbow: Y	right_wrist: Y	chest: Z [cm]	right_elbow: Z	right_wrist: Z	motion name
2	0	26,032	52,352	58,253	44,762	32,819	24,629	16,948	−8,589	−17,068	working
3	0,0333	26,02	52,378	58,192	44,81	32,769	24,681	16,957	−8,616	−16,966	
4	0,0667	26,016	52,387	58,101	44,826	32,718	24,649	16,96	−8,625	−16,817	
5	0,1	26,016	52,387	57,909	44,826	32,698	24,723	16,96	−8,625	−16,434	
6	0,1333	26,016	52,388	57,637	44,826	32,697	24,921	16,96	−8,677	−15,85	
7	0,1667	26,016	52,483	57,514	44,826	32,653	25,01	16,96	−8,78	−15,622	
8	0,2	26,016	52,67	57,453	44,826	32,568	25,093	16,96	−8,781	−15,571	

figure 3.19 Measured training data

NN-ANARX structure based models of chest, elbow and wrist were trained. X, Y and Z coordinates of the surgeon's chest, elbow and wrist at 3 different time steps corresponding to the third order of the model were used as inputs of the models. 3 different neural networks of the structure shown in figure 3.15 having three output were trained to obtain models of movement of X, Y and Z coordinates of the surgeon's chest, elbow and wrist. Thus, each of 3 NN-ANARX MIMO models had 27 inputs (X, Y and Z coordinates of chest, elbow and wrist at 3 time instances).

Real measurements with sample time 0.033 sec were used for identification. Data set consisting of 1227 measurements of each coordinate was used as training set. MIMO NN-ANARX model with three sub-layers of the hidden layer corresponding to the third order of the model ($n=3$) were trained. The following three models were obtained as follows.

Model of the surgeon's chest:

$$[x_c(t), y_c(t), z_c(t)]^T = \sum_{i=1}^{3} C_c^i \cdot \varphi_c^i\left(W_c^i [X(t-i), Y(t-i), Z(t-i)]^T\right), \quad (3.38)$$

where

$x_c(t), y_c(t), z_c(t)$ are corresponding coordinates of the marker placed on the surgeon's chest;

C_c^i, W_c^i, $i=1,\ldots,3$ are matrixes of synaptic weight of MIMO NN-ANARX based model of the surgeon's chest;

$\varphi_c^i(\cdot)$, $i=1,\ldots,3$ are nonlinear activation functions of the corresponding sub-layers of the hidden layer neurons of this model;

$X(t), Y(t), Z(t)$ are the vectors of corresponding coordinates of chest, elbow and wrist:

$$X(t) = [x_c(t), x_e(t), x_w(t)], \quad (3.39)$$
$$Y(t) = [y_c(t), y_e(t), y_w(t)], \quad (3.40)$$
$$Z(t) = [z_c(t), z_e(t), z_w(t)]. \quad (3.41)$$

Analogously to the model of the surgeon's chest, the model of the surgeon's elbow is

$$[x_e(t), y_e(t), z_e(t)]^T = \sum_{i=1}^{3} C_e^i \cdot \varphi_e^i\left(W_e^i [X(t-i), Y(t-i), Z(t-i)]^T\right) \quad (3.42)$$

and the model of the surgeon's wrist is

$$[x_w(t), y_w(t), z_w(t)]^T = \sum_{i=1}^{3} C_w^i \cdot \varphi_w^i\left(W_w^i [X(t-i), Y(t-i), Z(t-i)]^T\right). \quad (3.43)$$

Note that input $[X(t-i), Y(t-i), Z(t-i)]^T$ is the same for all three models (3.38), (3.42) and (3.43).

Levenberg-Marquardt algorithm was chosen to perform training of the network. After training the network different data sets (obtained from another surgical operation) were used for validation of the model. Figures 3.20-3.22 represent the results of prediction simulation for 5 time steps.

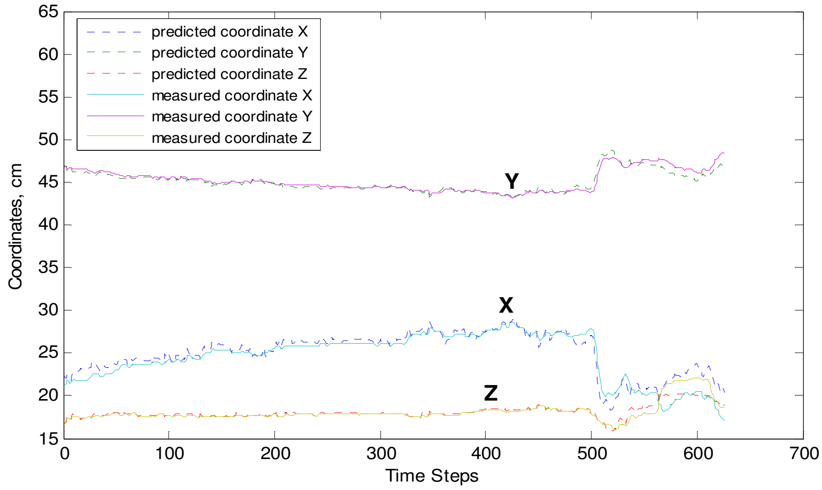

figure 3.20 Prediction of surgeon chest coordinates for 5 time-steps

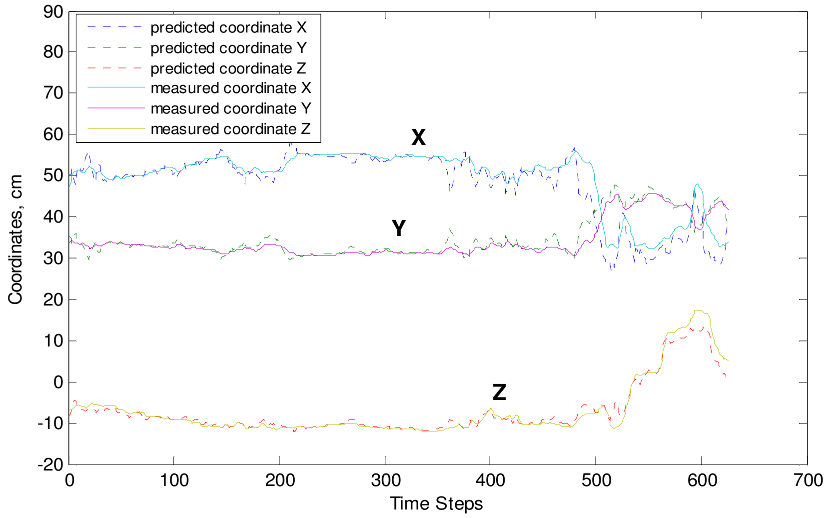

figure 3.21 Prediction of surgeon elbow coordinates for 5 time-steps

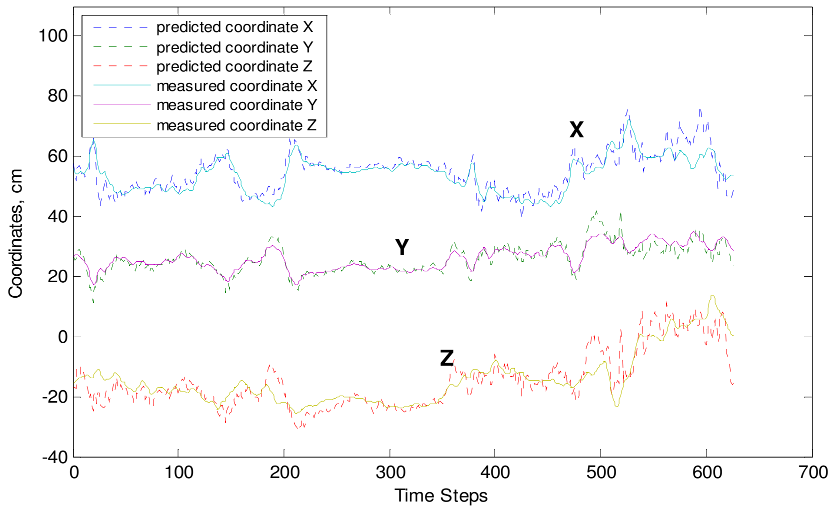

figure 3.22 Prediction of surgeon wrist coordinates for 5 time-steps

It can be seen from the figures that the model is capable of predicting chest, elbow and wrist coordinates up to 5 time-steps with high degree of accuracy. For the validation experiment shown in these figures, the mean absolute errors of prediction of the surgeon's chest X, Y and Z coordinates, elbow X, Y and Z coordinates and wrist X, Y and Z coordinates are 7.3mm, 2.6mm, 2.5mm, 1.68cm, 8.8mm, 8.4mm, 2.3cm,1.5cm, 3.4cm correspondingly. It is a very good accuracy, especially, if we take into account the fact that measurement errors of the image processing system persist in the validation set as well as in the training set. Coordinates were obtained by an image processing system tracking positions of 3 color markers that are about 3cm in diameter. Thus, prediction errors are not grater than diameters of the markers. It can also be mentioned that when training a neural network we filter random noise which present in the training set as it was demonstrated in numerical example 3.1 (see section 3.3.1). So, we assume that measurement errors are filtered during the identification procedure. Comparing model output with filtered measurement errors to a noisy validation data set also causes certain validation errors.

These predicted coordinates were successfully used for detecting current motion of the surgeon as described in [73]. 5 time-steps (0.166sec) prediction is enough for motion detection and image processing algorithms. It is also a very good prediction when we take into account that some actions during a real surgical operation, like passing or extracting an instrument take no more than 1 or at most 2 seconds.

The model was trained on the data obtained for a particular surgeon. The model may differ a little bit for another surgeon having another length of his/her hand and so on. So, it is very important that the model is obtained in the form of a neural network and therefore can be very easily adapted for another surgeon, a slight changes in positions of observing cameras and so on. It can be made during the working stage which lasts at least 10 or more seconds in the beginning of the operation.

This example demonstrates that NN-based MIMO ANARX structure having constrained connectivity because of separating different time instances can be very successfully used for obtaining quite complicated models, when using classical NN-based NARX structures is practically impossible because of great number of connections and therefore parameters that have to be calculated by a training algorithm. In some cases the need to separate time instances is also very obvious because of the nature of the modeled process.

3.8 Conclusions

Identification of nonlinear dynamic systems by recurrent and feedforward neural networks was discussed in this section and demonstrated on numerical examples. Two types of feedforward (static) neural networks based model of dynamic processes were considered. Both of them use external feedback and delays. First of them is classical NN-based NARX model and the second one NN-based Additive NARX model having constrained connectivity of its neurons.

Author's contribution is in implementing NN-based Additive NARX structure for identification of nonlinear dynamic systems [13], [14] and in developing NARX [55] and ANARX [13] based neural network structures representing Hammerstein models for identification of systems with static input nonlinearities. For example, systems with nonlinear actuators.

Identification of nonlinear multiply input – multiply output (MIMO) systems by ANARX structure is also shown in this section. NN-based MIMO ANARX model is proposed by the author [56] and successfully implemented for modeling of the surgeon's hand [73].

Feedforward neural networks based dynamic models have better identification capabilities than models based on neural networks with internal feedbacks, where the model becomes extremely complicated to be trained successfully.

It can be concluded by comparing two types of models based on feedforward neural networks with external feedbacks (NN-based NARX and NN-based ANARX models) that despite the fact that the first one has simpler structure, the second one has several significant advantages over classical NARX structure. Namely, it has much smaller number of parameters (synaptic weights) and as experiments have shown this fact does not cause any drawbacks in quality of identification but makes possible implementing wider range of training algorithms, for example second order training algorithms can be used instead of classical first order gradient descent error backpropagation. Levenberg-Marquardt algorithm with NN-ANARX can be used for obtaining even very complicated models like the model of the surgeon's hand movement. Especially in case of MIMO models, where number of connections and therefore parameters of the network grows dramatically, NN-ANARX models are preferable.

It can also be mentioned that NN-ANARX structure is always reprezentable in a classical state-space form and is always linearizable by dynamic output feedback. The last property is very essential for model based neurocontrol, which will be discussed in the next chapter.

Chapter 4

Model Based Neurocontrol

Advanced process control case studies by the Warren Center [81] provide strong evidence that economic benefits of up to 6% of plant operating cost can result from the application of advanced forms of process control. Examples of process features addressed by this level of control are nonlinearities, time-varying characteristics of parameters, multivariable inputs and outputs, unstable states and transport delays.[39] Artificial neural networks have shown grate capabilities of modeling these features. This stems from the theoretical ability of neural networks of various types to approximate arbitrary well continuous nonlinear mappings.[17]

Intelligent control and neural networks based control as a type of intelligent control is now a common tool in many engineering and industrial applications. [83], [84], [24] Intelligent control has the ability to comprehend and learn about plants, disturbances, environment, and operating conditions and artificial neural networks, with their self-organizing and learning conditions, are used as promising tool for such purposes.[85], [24]

By applying neural networks to control (neurocontrol) we treat them as a candidate for a genetic, parametric, nonlinear model of a broad class of nonlinear plants.[17], [82] This approach can be called Model Based Neurocontrol. Neurocontrol is a dynamic research field that has attracted considerable attention from the scientific and control engineering community. [38] Research on neural-network-based control systems has received a considerable attention over the past two decades.[87]

Because of neural network's ability to learn and approximate nonlinear functions arbitrarily well, a large number of identification and control structures based on neural networks have been proposed (see, for example, [11], [61], [62], [86], [88], [89], [90]). Numerous successful practical applications of different model based neurocontrol techniques have also been shown in literature in the last years (e.g. in [91], [92], [93], [94]), but there is also a lot of space for further research, development, analysis and improvements in this field.

Model based neurocontrol techniques can be divided into two types considered in this chapter. The first one let's call model structure independent control techniques, because the techniques belonging to this class of do not require any particular structure of the model. The model is considered as a "black box". Control quality depends on the quality of the model. The better model accuracy is, the better is control quality. Implementing better (more specific) model structures can improve the quality of identification of a nonlinear system and thus significantly increase control quality. The most popular model based predictive neurocontrol and inverse model based control will be considered as classical representatives of this type of model based neurocontrol techniques.

The second type of model based neurocontrol techniques can be characterized as model structure dependent control techniques, because control algorithms belonging to this class require particular structure (particular representation) of the model. NN-ANARX structure based dynamic output linearization will be considered as a control technique based on certain structure of the model. This algorithm requires ANARX-type model of the controlled system or process.

4.1 Author's contribution

Author's contribution is in comparing different nonlinear model based neurocontrol techniques and implementation of different neural network structures for model based control of nonlinear dynamic systems.

- Neural Networks based Hammerstein model of DC servo motor with nonlinear driver discussed in section 3.4.2 is applied to predictive control of the motor [55];
- An adaptive inverse model based control algorithm is proposed in [55];
- Dynamic output feedback linearization algorithm is applied to control of nonlinear systems by using neural networks based ANARX models of the systems [13], [14], [76];
- Adaptive control technique based on dynamic output feedback linearization of NN-ANARX model with History-Stack Adaptation is proposed in [14];
- Considered control techniques are compared. Their advantages and drawbacks are discussed.

4.2 Structure Independent Control Algorithms

Control algorithms based on forward and inverse models of controlled plant will be discussed in this section.. The techniques considered here are model based, but any structure of the model can be used. Even not necessarily neural networks based structure. Never the less, choosing proper structure of the model will increase the accuracy of the model and thus improve the quality of the control algorithms. Also, neural networks based models give additional advantages because of network's natural ability to learn and very good approximation capabilities demonstrated in the previous chapter of the thesis.

Nonlinear predictive control via neural network based models will be considered as classical and the simplest forward model based neurocontrol technique.

4.2.1 Neural network based predictive control

Neuro predictive control algorithm is one of the most popular (if not the most popular) neurocontrol techniques. The predictive control algorithm is simple and easy to implement.[90] Neural networks have very often been used in various predictive control algorithms that utilize nonlinear process models as a modeling tool.[90], [95], [96], [97] The resulting controller would prove to be more robust in practical situations where the nature of the non-linearity in the process is unknown.[17] Numerous practical modern time control applications use predictive controllers with neural network based predictive models (see, for example, [91], [98], [99]).

Among various modifications of the predictive control algorithm, simple technique defined by equation (2.33) and discussed in section 2.4.1 was chosen as a control algorithm for calculation of control signals. The corresponding structure of the closed loop system is depicted in figure 2.2.

Neural Networks can be used as a predictive model to calculate predictions $\hat{y}(t+N_1),\ldots,\hat{y}(t+N_2)$ used by equation (2.33). Quality of the control significantly depends on the quality (accuracy) of these predictions.[100] Choosing proper structure of the model can help significantly improve the accuracy of the predictive model.

It is shown in [16] that the determination of the structure is essential, because the model can easily become overparameterized by simple increasing the number of backward time shifts of the input and output signals or the degree of nonlinearity. In general, this procedure will result in an excessively complex model and possibly numerical ill-conditioning. However, determining the model structure of nonlinear systems is more complicated than determining the model structure of linear single-input single-output systems, where only the model order has to be detected.

Structural properties of a particular nonlinear modeled system or process can often be very easily taken into account when developing a neural network based model for being used by a predictive control algorithm. Because of consisting of single interconnected neurons, neural networks have natural ability to form almost any necessary structure. Application specific structures of the model can significantly improve the quality of the predictive control algorithm where they are used as predictive models.

Neural Networks based Hammerstein model proposed by the author in [55] was successfully implemented to identification of systems with nonlinear actuators as it was shown in section 3.4 of this thesis. Simulation of predictive control of DC servo motor with nonlinear driver based on this model will be demonstrated in the next example.

4.2.2 Numerical example 4.1 – predictive neurocontrol of DC servo motor with nonlinear driver

Identification of DC servo motor with nonlinear driver (3.16)-(3.17) by neural network based Hammerstein model (3.13) depicted in figure 3.5 was shown in section 3.4 of this thesis. Predictive control technique based on this model was proposed by the author in [55].

The aim of the control system is to reach the desired motor position (angle) with minimal possible regulation time and steady-state error. In [70] a multilayer perceptron was trained to perform as a nonlinear neurocontroller for this system. The control algorithm was tested with the set point 4*rad* and zero initial state. Regulation time was about 0.15*sec* and steady-state error was -0.06397*rad*.

Predictive control algorithm (2.33) corresponding to the structure of control system depicted in figure 2.2 was proposed by the author to control this system in [55]. DC motor has linear dynamics (or dynamics very close linear) and static nonlinear driver as an actuator. So, it is natural to separate linear dynamics and static input nonlinearity in the model. It was done automatically by training neural network based Hammerstein model (3.13). This model was used as to calculate predictions $\hat{y}(t+N_1),\ldots,\hat{y}(t+N_2)$ where prediction horizons were chosen as $N_1 = 1$ and $N_2 = 3$. Thus, the control is based on 3 time-steps prediction of the system's output $\hat{y}(t+1),\, \hat{y}(t+2),\, \hat{y}(t+3)$.

The output of the model is the predicted rotation velocity of the motor. As we have to control the position, it should be integrated. So, the model inside the controller consists of two sub-networks connected into the structure shown in figure 3.5 and one discrete time integrator. Controller has two set points: 0*rad/sec* for the rotation speed and 4*rad* for the position. The control simulation results are presented in figures 4.1 and 4.2. Figure 4.1 shows the velocity of the motor reaching the set point and figure 4.3 represents shaft position (angle) of the motor. It can be seen from the figures that the regulation time is less than 0.1 seconds when the technique proposed in [55] is used. Steady-state error is 0.00224*rad*. It is a very good result especially for an indirect control where the quality of the control system significantly depends on the accuracy of the model.

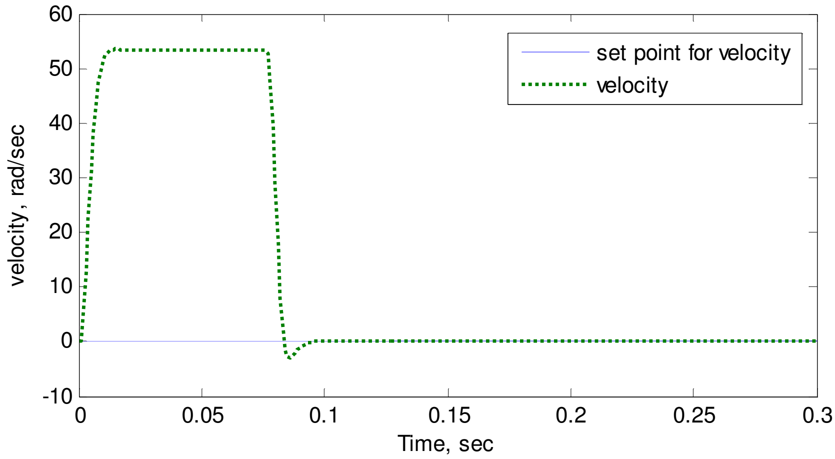

figure 4.1 DC motor velocity control

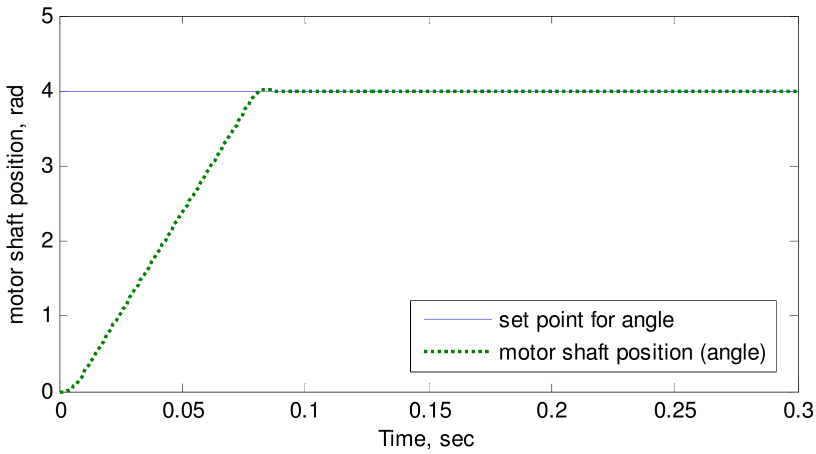

figure 4.2 Control of DC motor shaft position

It can be concluded that combining two sub-networks (one for static nonlinearities and second for linear dynamics) in one structure gives significant improvement in modeling of linear dynamic systems with static input nonlinearities. Because of high accuracy of the obtained models it is possible to implement indirect control algorithms to a class of dynamic systems with nonlinear actuators. Computer simulations show that by using the models proposed in [55] and discussed in sections 3.4 and 3.6 of this thesis can significantly improve the quality of predictive control of the systems with actuator nonlinearities, which are also called hard nonlinearities. Because of neural network's natural ability to learn and adjust their parameters, adaptive controllers based on this models will have strong robustness. Controllers can be turned to work in different environmental conditions. Control algorithms based on these models ban be especially useful in case of unknown or changing actuator nonlinearities.

4.2.3 Inverse modeling based control

This control technique is also based on modelling of the controlled system, but in this case an inverse model has to be obtained. Inverse model should be capable of calculating estimations of the system's input values $\hat{u}(t)$. This estimations are used as control signals. In general, inverse control techniques were discussed in section 2.4.2. Here the main attention is paid to inverse control based on neuro-models.

Any neural network structure can be used to obtain an inverse model, but it was studied by the author in [26] that feedforward neural networks with external feedback can better approximate inverse functions of nonlinear dynamic systems than recurrent neural networks with internal feedbacks. Direct inverse control relies heavily on the fidelity of the inverse model used as the controller, but serious questions arise regarding the robustness of direct inverse control, because of absence of feedback.[23] This problem can be overcome by using neural network based models capable of on-line learning [23] and/or by implementing inverse models belonging to the class of NARX models (see section 2.2.3) with the control scheme [25] depicted in figure 2.3. NARX models can be obtained by training a feedforward neural network as it was shown in section 3.3 of his thesis. When a two-layer perceptron is used, the inverse NARX model can be formalized by the following equation.

$$\hat{u}(t) = C \cdot \varphi(W \cdot [y(t), y(t-1), \ldots, y(k-n), u(t-1), \ldots, u(t-m)]^T), \qquad (4.1)$$

where C and W are matrixes of output and input weight coefficients, $\varphi(\cdot)$ is a nonlinear activation function of the hidden layer neurons, n and m are output and input order of the model.

This equation can be used for calculation of estimations of the controlled system's inputs. This inputs are used as control signals. It was demonstrated in [101] that it is very easy to apply inverse model based controllers to complicated systems. Neural network based inverse models have become a very popular tool in practical control applications (for example, see [102], [103], [104], [105], etc).

The parameters of the inverse model can be adjusted on-line by using on-line learning and thus the robustness of the inverse model based controller can be increased.[23] An adaptive inverse model based neurocontrol technique was used by the author in [26] to control of nonlinear system (3.8) considered in numerical example 3.1 (see section 3.3.1 of this thesis). Consider the following example.

4.2.4 Numerical example 4.2

Inverse identification and adaptive inverse neurocontrol of system (3.8) was shown by the author of this thesis in [26]. A two-layer perceptron was trained to perform as inverse model of nonlinear system (3.8) having NARX structure. System (3.8) is a second order system represented by a difference equation

$$y(t) = f\left(y(t-1), y(t-2), u(t-1)\right). \tag{4.2}$$

Inverse model of this system has to approximate the following function

$$\hat{u}(t-1) = f^{-1}\left(y(t), y(t-1), y(t-2)\right). \tag{4.3}$$

Estimation $\hat{u}(t-1)$ is then used as the control signal at time step t+1

$$u(t+1) = \hat{u}(t-1). \tag{4.4}$$

Two-layer perceptron was trained to approximate function (4.3). Matrixes of input and output synaptic weights W and C were obtained by BP training algorithm (see section 2.6.1). Inputs gathered from the input of the system were used as etalon values for outputs of the network representing the inverse model. After being trained on the training set consisting of 1000 input-output pairs the inverse model was tested on sinusoidal validation signal. Output of the system was given to the input of the inverse model and the output of the trained neural network was compared to the input of the system. Validation result of the two-layer perceptron based inverse model of system (3.8) is depicted in figure 4.3.

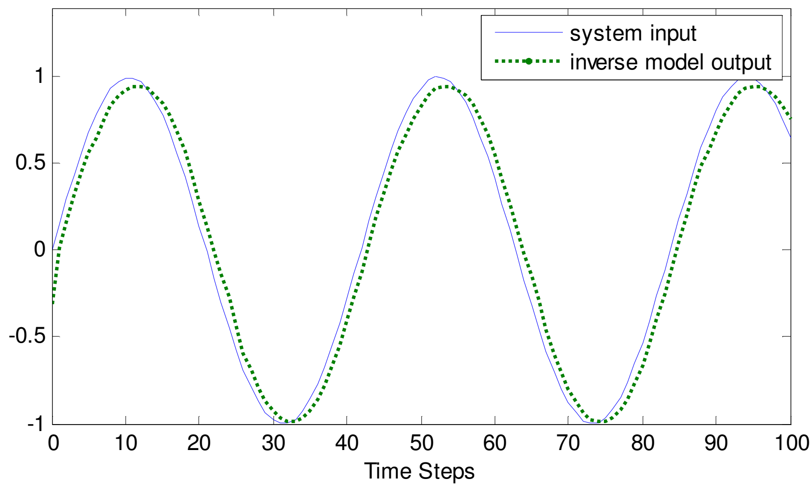

figure 4.3 NN-NARX structure based inverse model validation

It can be seen from the figure the output of the neural network based inverse model follows the input of the system with high degree of accuracy and exactly 1 time step delay. This is because of estimation of system's input $\hat{u}(t-1)$ (NB! not $\hat{u}(t)$) as it was shown in equation (4.3) following from equation (4.2). Mean Square Error on the validation set was as low as about 0.00010.

Control signals can now be calculated by this NN-NARX based inverse model as

$$u(t+1) = C \cdot \varphi\left(W \cdot [y(t), y(t-1), y(t-2)]^T\right), \qquad (4.5)$$

where sigmoid-type hyperbolic tangent function (2.51) was used as nonlinear activation function of 20 hidden layer neurons $\varphi(\cdot)$.

Schematic representation of the closed loop adaptive control system is depicted in figure 4.4.

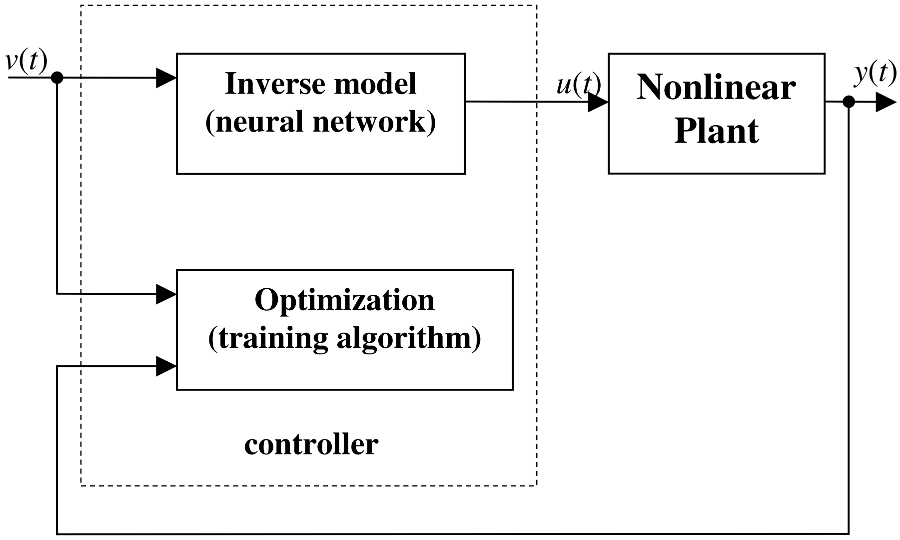

figure 4.4 Structure of inverse model based neurocontroller

The aim of the control system is to calculate such a control signal $u(t)$ the system's output $y(t)$ follows the desired reference signal $v(t)$. In other words, a neural network trained as an inverse model of the controlled system is used to calculate such an output (input of the plant) $u(t)$ that

$$y(t) \to v(t) \qquad (4.6)$$

Control error $e(t)$ can be defined as

$$e(t) = v(t) - y(t) \tag{4.7}$$

We assume that this error is caused only by inaccuracy of the inverse model and use it as a training signal (teacher) to adjust the model. Such an approach, where instead of receiving the correct responses from a teacher, learning information is derived from a scalar reinforcement or performance feedback signal indicating how well the network is performing its assigned task, is called Reinforcement Learning (RL).[39]

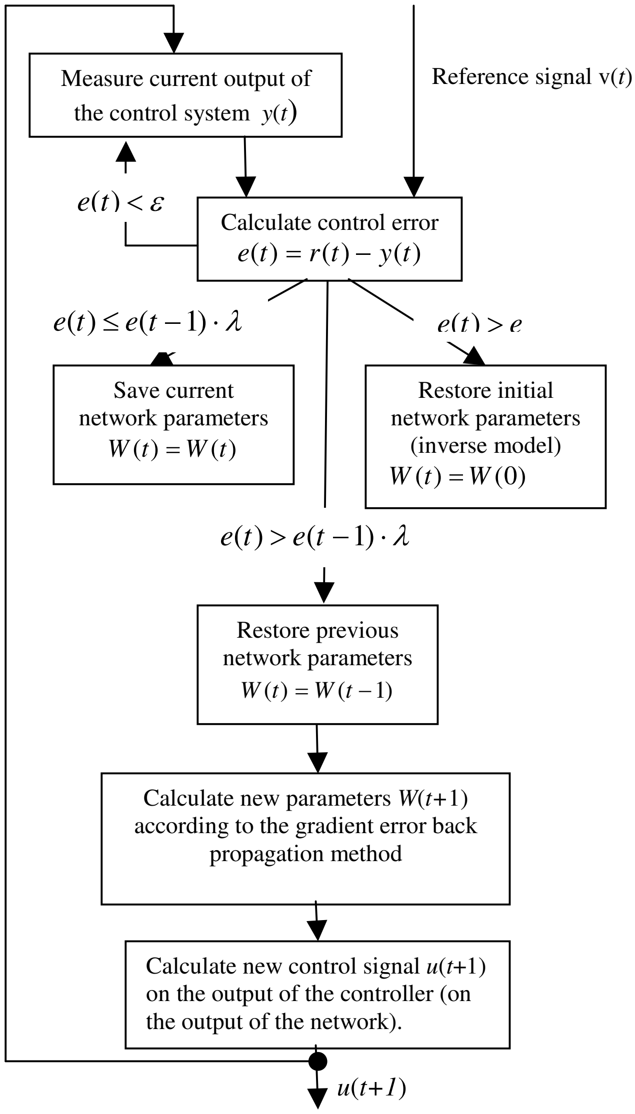

figure 4.5 Strategy for inverse model based neurocontrol

It has to be mentioned that by using RL with the structure depicted in figure 4.4, neural network can also be directly trained to perform as a controller without previous identification of the inverse model, but it was shown in [27] that the neural network controllers with model information give better results than without any model information. Another advantage of preliminary inverse modeling is in possibility to restore initial parameters of the controller when parameters begin significantly and unpredictably change during the adaptation (on-line training) in case of large short-time disturbances. This significantly increases the robustness of the control system. The following adaptation strategy for inverse model based neurocontrol was proposed by the author in [26].

If the error $e(t)$ becomes more than ε, a number defining the desired accuracy, neural network can be trained on-line to minimize it. Our experiments have shown that one training epoch of BP training algorithm before each computation of the control signal is enough to modify the controller's behavior in response to changes in the dynamics of the process and the character of the disturbances. Such adjustment of the network's parameters makes this controller adaptive [42]. The diagram representing the proposed algorithm is depicted in figure 4.5. Here $W(t)$ is the matrix of network parameters (synaptic weights) at time step t.

As an example, consider application of the considered technique for adaptive control of system (3.8). The results of the control system simulation are depicted in figure 4.6.

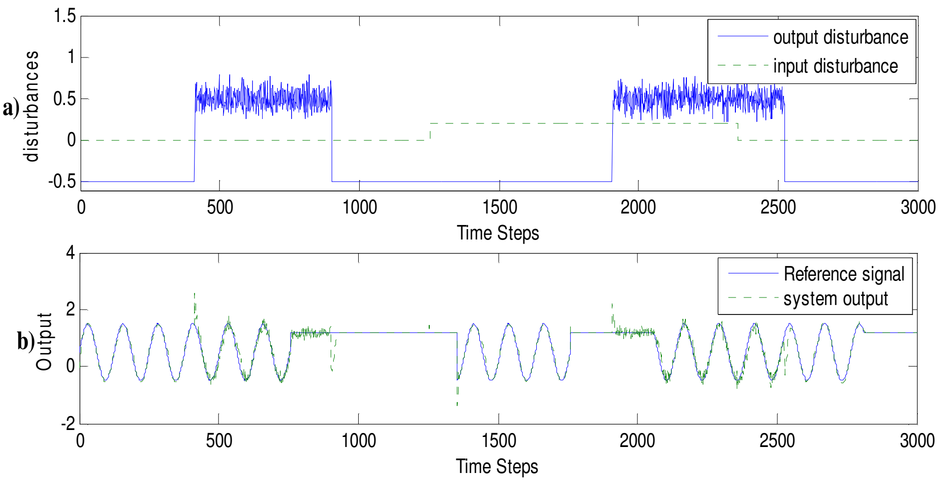

Figure 4.6 Simulation of adaptive inverse model based neurocontrol

Figure 4.6a shows system's input and output disturbances and figure 4.6b presents the output of the system with the reference signal (the desired output). It can be seen from the figures that this control system is capable of tracking the desired trajectory and compensating comparatively large input and output disturbances by adjusting the inverse model on-line.

Another approach for adjustment of the model is using measured in real time inputs and outputs of the controlled system as training data. In model based neurocontrol schemes, both reinforcement learning and training with teacher can be used for adaptation by adjusting inverse or/and forward neural networks based models. Such techniques have also been successfully implemented for neural networks based predictive control (e.g. see [106]), where forward model is adjusted online in response to disturbances, changing environmental conditions and so on. In case of inverse model based neurocontrol, RL provides additional negative feedback and thus increases the robustness of the control system. An alternative approach to adaptive model based neurocontrol will be discussed in section 4.4.

In the next section, a neurocontrol technique based on dynamic linearization of the model of the corresponding structure will be considered.

4.3 NN-ANARX structure based control

It was discussed in section 2.4.3 and proofed in [32] that ANARX structure (2.13) is always linearizable by Dynamic Output Feedback Linearization (2.36)-(2.38). Model of a nonlinear system can be obtained in the form of ANARX structure by training a neural network of the structure shown in figure 3.9. NN-ANARX model (3.18) belongs to the class of ANARX models (2.13) and also can always be linearized.[11] By using parameters of Neural Network based ANARX model (3.18), Dynamic Output Feedback Linearization algorithm (2.36)-(2.38) can be presented as follows.

$$F = C_1\varphi_1\left(W_1 \cdot [y(t), u(t)]^T\right) = \eta_1(t) \tag{4.8}$$

$$\begin{aligned}
\eta_1(t+1) &= \eta_2(t) - C_2\varphi_2\left(W_2 \cdot [y(t), u(t)]^T\right)\\
&\vdots\\
\eta_{n-2}(t+1) &= \eta_{n-1}(t) - C_{n-1}\varphi_{n-1}\left(W_{n-1} \cdot [y(t), u(t)]^T\right),\\
\eta_{n-1}(t+1) &= v(t) - C_n\varphi_n\left(W_n \cdot [y(t), u(t)]^T\right)
\end{aligned} \tag{4.9}$$

where n is the order of the model (number of sub-layers in the corresponding neural network);
C_i, W_i for $i = 1, \ldots, n$ are matrixes of synaptic weights of the network (3.18);
$\varphi_i(\cdot)$ for $i = 1, \ldots, n$ are nonlinear activation functions of neurons in the corresponding sub-layer of the hidden layer of the model (3.18) depicted in figure 3.9.

Control signal u has to be found by solving equation (4.8):

$$u(t) = F^{-1}\left(y(t), \eta_1(t)\right). \tag{4.10}$$

Numerical calculation can be used for calculation of (4.10). The structure of the corresponding control system is depicted in figure 4.7.

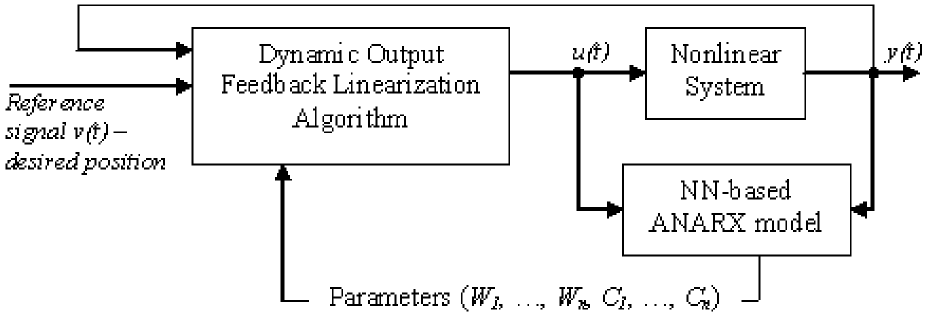

figure 4.7 Structure of neurocontrol system based on dynamic output feedback linearization of NN-ANARX model

The above described control technique was applied to control of nonlinear SISO and SIMO systems in [13], [14] and [76]. Numerical Newton's Method (see section 2.4.4) was used for solving equation (4.8) and finding control signals $u(t)$ on each time step. Our experiments [13], [14] have shown that 3 iterations will guarantee convergence of the Newton's Method with sufficient precision. Consider the following numerical examples.

4.3.1 Numerical example 4.3

This example is a continuation of numerical example 3.5 (see section 3.5.3).

The model of a jacketed Continuous Stirred Tank Reactor (CSTR) [74], [75], [13], [14] is represented by second order input-output equation (3.24). Identification of this system by neural network based ANARX model is shown in [13], [14] and discussed in detail in section 3.5.3 of this thesis (numerical example 3.5).

Because of the last term in (3.24), the model does not have an ANARX structure. Never the less, it was considered as a "black box" and neural network having ANARX structure (see figure 3.9) was trained. Validation results have shown that in spite of the restrictions imposed by NN-ANARX structure on connectivity matrixes, the identified model explains input-output data with high degree of accuracy. [13], [14]

NN-ANARX model based Dynamic Output Feedback Linearization algorithm (4.8)-(4.10) was used for control of jacketed CSTR system (3.24) by using parameters (3.26) of neural network representing ANARX model of this system.

For the second order model (3.25), control strategy NN-ANARX model based control strategy can be represented by the following equations.

$$\eta_1(t+1) = v(t) - C_2\varphi_2\left(W_2 \cdot [y(t), u(t)]^T\right) \tag{4.11}$$

$$F = C_1\varphi_1\left(W_1 \cdot [y(t), u(t)]^T\right) = \eta_1(t) \tag{4.12}$$

$$u(t) = F^{-1}\left(y(t), \eta_1(t)\right) \tag{4.13}$$

Thus, by using parameters (3.26) of the model, controls $u(t)$ for jacketed CSTR system (3.24) were calculated by numerical solving of equation

$$[31.9678 \quad -0.0166 \quad 13.8712] \cdot \varphi_1\left(\begin{bmatrix} 0.1793 & 0.2616 \\ -23.0273 & 22.2099 \\ -0.0785 & -0.4576 \end{bmatrix} \cdot \begin{bmatrix} y(t) \\ u(t) \end{bmatrix}\right) = \eta_1(t) \tag{4.14}$$

where

$$\eta_1(t) = v(t-1) - [-0.0128 \quad -19.8288 \quad -25.9744] \cdot \\ \cdot \varphi_2\left(\begin{bmatrix} -23.4344 & 29.5285 \\ 1.5266 & -2.0768 \\ -1.0887 & 1.6082 \end{bmatrix} \cdot \begin{bmatrix} y(t-1) \\ u(t-1) \end{bmatrix}\right). \tag{4.15}$$

Control strategy (4.14)-(4.15) was applied to control of system (3.24). The result of the control system simulation with the piece-constant reference signal $v(t)$ is depicted in figure 4.8.

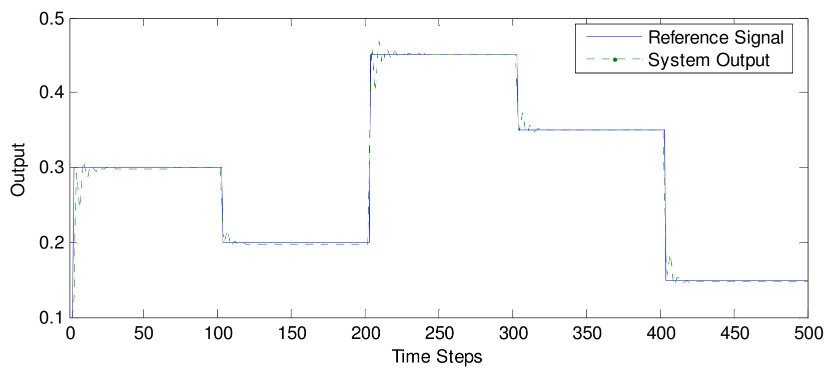

figure 4.8 Control of jacketed CSTR by NN-ANARX based dynamic output linearization

It can be easily seen from the figure that the output of the system closely follows the reference signal. Dynamic output feedback linearization can be successfully used for control of jacketed CSTR system by using synaptic weights of neural network trained off-line to perform as ANARX model of this system.

4.3.2 Numerical example 4.4 – Backing up control of a truck-trailer

Existing solutions of a truck-trailer backing up problem are mainly based on employing neural networks based, fuzzy or neuro-fuzzy controllers [107], [108], [109], [110] and practically no solutions available where more classical control methods are used. On one side such disproportion can be explained by the fact that fuzzy and neural controllers very well suited for such tasks. On the other side in many cases models of the truck-trailer are developed from the physical point of view and usually are unsuitable for application of such methods like feedback linearization. Of course one can linearize the model around a number of operating points, but as it was mentioned in [110] such approach can be computationally complicated and requires considerable design effort.

Neural networks based modeling and such a classical technique like dynamic output feedback linearization were combined in [76] with a purpose to backing up control of a truck-trailer. Backward motion of a truck-trailer is modeled by NN-based ANARX model and then linearization (4.8)-(4.10) is applied as control technique according to the structure of the control system depicted in figure 4.7.

Consider the problem of backing up control of a truck-trailer. The problem is to control steering angle in order to track the desired trajectory (y coordinate) of the truck-trailer from any initial position. Following equations were proposed by [107] to model the dynamics of the truck-trailer.

$$\begin{aligned}
x_1(t+1) &= x_1(t) + v \cdot \tau / \lambda \cdot \tan(u(t)) \\
x_2(t+1) &= x_2(t) + v \cdot \tau / L \cdot \sin(x_5(t)) \\
x_3(t+1) &= x_3(t) + v \cdot \tau \cdot \cos(x_5(t)) \cdot \cos(x_2(t+1) + x_2(t)/2) \\
x_4(t+1) &= x_4(t) + v \cdot \tau \cdot \cos(x_5(t)) \cdot \cos(x_2(t+1) + x_2(t)/2) \\
x_5(t) &= x_1(t) - x_2(t),
\end{aligned} \tag{4.16}$$

where $x_1(t)$ is the angle of truck, $x_2(t)$ is the angle of trailer, $(x_4(t), x_3(t))$ is horizontal and vertical positions of the rear end of trailer, respectively, $x_5(t)$ is the angle between truck and trailer, $u(t)$ is the steering angle, l is the length of the truck, $L$is the length of the trailer, τ is the sampling time and v is the constant speed of backing up. Figure 4.9 shows the model of the truck-trailer.

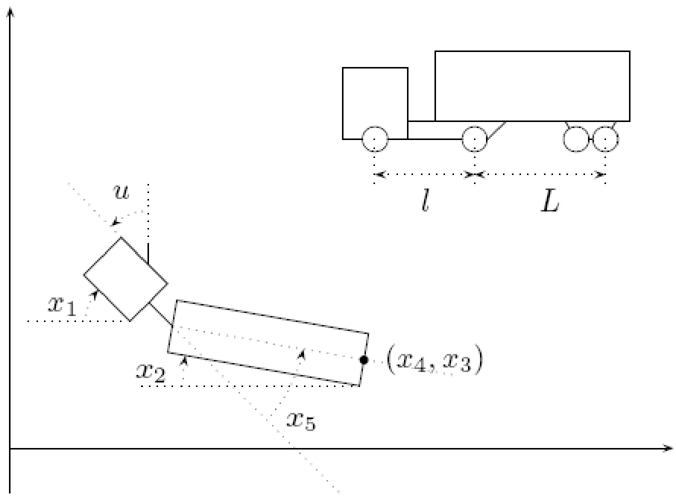

figure 4.9 Truck-trailer and main parameters of its model

The following parameters of the truck-trailer were used:

- $l = 2.8\,m$
- $\tau = 1.0\,s$
- $L = 5.5\,m$
- $v = -1.0\,m/s$

Note that allowed only backing up.

While system (4.16) describes the behavior of a truck trailer with high level of precision it is not linearizable by dynamic output feedback [111] and consequently could not be used directly for feedback design. Thus, necessity to use another model is obvious. NN-based ANARX structure was chosen to model the truck-trailer.

In order to obtain input-output data for identification, plant (4.16) was simulated with Uniform Random Number signal $u(t) \in [-\pi/4; \pi/4]$. Steering angle was used as the input of the model and vertical positions of the rear end of the trailer $x_3(t)$ was considered as the output of the model. After that NN-ANARX structure (see figure 3.9) with three sub-layers of the hidden layer corresponding to the third order of the model was trained. Three neurons were used in each sub-layer. After 5000 epochs of training by LM training algorithm (2.65)-(2.67) the Mean Square Error on the validation set (not used in the training set) was as low as about 0.001. The model was obtained in the following form

$$y(t) = \sum_{i=1}^{3} C_i \varphi_i \left(W_i \cdot [y(t), u(t)]^T \right), \tag{4.17}$$

where C_i and W_i are matrixes of synaptic weights (parameters of the model) and

$$y(t) = x_3(t). \tag{4.18}$$

After the system (4.16) was identified by NN-ANARX model (17) the linearization algorithm (4.8)-(4.10) was applied as

$$F = C_1\varphi_1\left(W_1 \cdot [y(t), u(t)]^T\right) = \eta_1(t) \tag{4.19}$$

$$\begin{aligned} \eta_1(t+1) &= \eta_2(t) - C_2\varphi_2\left(W_2 \cdot [y(t), u(t)]^T\right) \\ \eta_2(t+1) &= v(t) - C_3\varphi_3\left(W_3 \cdot [y(t), u(t)]^T\right) \end{aligned} \tag{4.20}$$

Newton's method (2.42)-(2.45) was used for numerical calculation of control signal $u(t)$ from (4.19).

A number of experiments of constructed system have been made to see the results of truck-trailer's behavior. Different initial $y(0) = x_3(0)$, $x_1(0)$, $x_2(0)$ and desired $v(t)$ positions of the truck-trailer have been chosen. Table 4.1 describes the experiments.

Table 4.1 Initial and desired positions of the truck-trailer

	Initial position			Desired position v(*t*) ($y(t) \to v(t)$)
	$x_1(0)$	$x_2(0)$	$y(0)=x_3(0)$	
Experiment 1	10°	-20°	10 m	0 m
Experiment 2	0°	0°	0 m	-10 m
Experiment 3	-20°	45°	0 m	5 m
Experiment 4	-180°	-180°	0 m	55 m
Experiment 5	0°	0°	0 m	$20\sin(0.03x)$ m

Experiments 1-4 show the movement of truck-trailer to desired line. Experiment 5 is harder, in this case truck-trailer's desired trajectory is described by sinusoidal signal. Next Figures 4.10-4.14 show simulation results for 5 experiments with different initial positions of the truck-trailer.

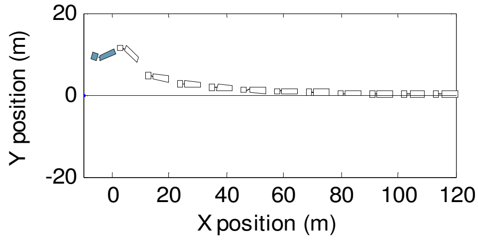

figure 4.10 Simulation of the control system: experiment 1

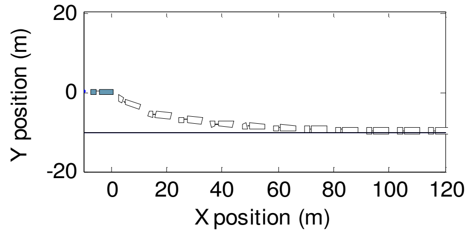

figure 4.11 Simulation of the control system: experiment 2

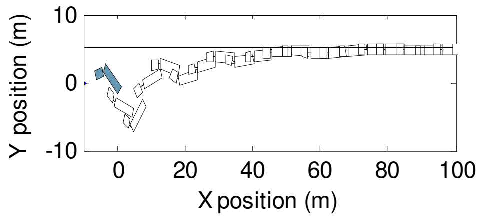

figure 4.12 Simulation of the control system: experiment 3

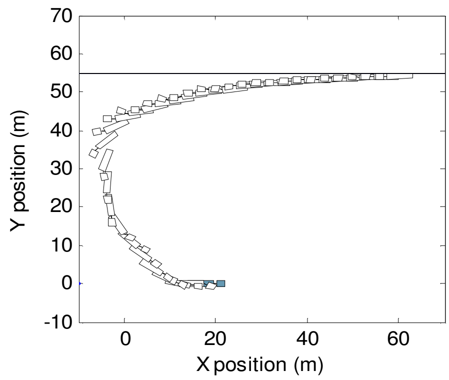

figure 4.13 Simulation of the control system: experiment 4

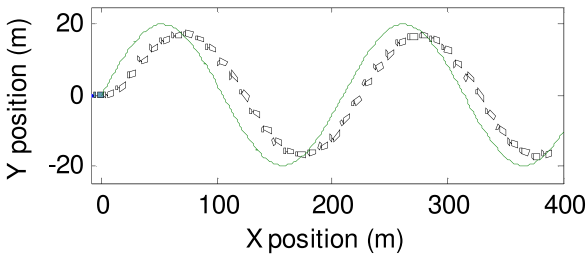

figure 4.14 Simulation of the control system: experiment 5

The presented approach is a combination of classical dynamic output feedback linearization with neural networks based modeling. This algorithm does not utilize the whole model, but uses parameters of the model for calculation of the dynamics of the controller providing high robustness and stability of the control system.

It is proofed in [32] that ANARX structure is always linearizable by dynamic output feedback linearization (2.36)-(2.38). NN-ANARX structure as a sub-class of ANARX structures is also always linearizable as shown by (4.8)-(4.10). Experimental results have shown that because of neural network's good approximating capabilities, NN-based ANARX structure is capable of modeling a wide class of nonlinear systems with high degree of accuracy. It means that

NN-ANARX model based dynamic output feedback linearization algorithm (4.8)-(4.10) can be for control of these systems.

Because of neural network's natural ability to learn and adjust it's parameters in response to unpredictable disturbances, changes in environmental conditions and/or changes in parameters of the controlled system, the dynamic output feedback linearization technique can be combined with neural networks based adaptation. An adaptive control algorithm based on output feedback linearization of NN-ANARX models was presented in [14] and will be discussed in the next section.

4.4 Adaptive NN-ANARX based control

The aim of the adaptive controller is to modify its behavior in response to changes in the dynamics of the process and disturbances. A real-time estimator is a central part in most adaptive controllers. Parameters of the controller has to be adjusted following the negative gradient of square error E^2 of the control system. There are many alternative ways to make real-time estimation [42], [43].

In case of model based neurocontrollers, parameters of neural network based model have to be adjusted. Since in case of neural network based dynamic output feedback linearization algorithm (4.8)-(4.10), the network was previously trained off-line and there is only a need to adjust the weights, it is suggested in [14] to implement in adaptive control scheme the gradient descent training algorithm (see section 2.6.1 of this thesis). The main advantage of latter is that it consists of a higher number of shorter iterations. On each iteration updated weights can be calculated very fast that is critical in some real-time applications[14]. Thus, classical dynamic output linearization can be combined with neural networks based adaptation by using neural networks for modeling of the controlled system.

The information contained in a single pattern can not be assimilated completely in a single presentation. So, more patterns representing the required mapping should be presented to the learning method at each time-step. The requirement to learn from incidents or samples from the environment has parallels with human learning and memory. There are two types of memories: short-term memory and long-term memory. The theory says that all new information to be memorized must first be processed through the short-term memory [39].

4.4.1 History-Stack Adaptation (HSA)

History-stack adaptation (HSA) algorithm [39] retains a short history of process patterns (short-time memory) that can represent an approximation to the nonlinear process dynamics. This information containing in the History Stack

(HS) is then transferred into the long-time memory by means of training algorithm.

The control system consists of unknown nonlinear plant preceded by the linearizing feedback. Unknown plant is modeled by the NN-ANARX structure (3.18). Dynamics of the feedback is defined by the NN-ANARX structure which is used to model the nonlinear plant and the desired linear dynamics. At each time-step the plant model is adjusted by HSA algorithm. The HS operates as a First-In-First-Out (FIFO) stack containing n_p patterns. At each time-step k HS accepts (memorizes) a net pattern $Z(t)=[y(t),u(t)]^T$ from the process and discards (forgets) the oldest pattern $Z(t-n_p)$. These elements $Z(t),Z(t-1),\ldots,Z(t-n_p+1)$ constitute the training set and are used in n_c cycles to update the weights [39], [112]. To achieve a good process performance the proper choice of the parameters n_p and n_c is essential [112]. The closed loop structure for the proposed control technique based on dynamic output feedback linearization of NN-ANARX model is depicted in figure 4.15.

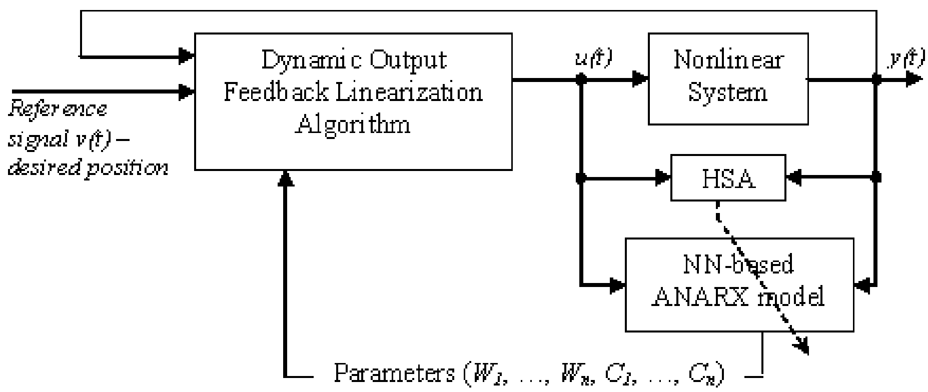

figure 4.15 Structure of adaptive control system based on dynamic output feedback linearization of NN-ANARX model with history-stack adaptation

This control system was successfully applied to adaptive control of nonlinear systems [14]. Consider the following numerical examples.

4.4.2 Numerical example 4.5

This example is a continuation of numerical example 3.4 (see section 3.5.2).

Identification of a liquid level system of interconnected tanks [72], [14] represented by input-output equation (3.21) by training a neural network based ANARX structure (3.22) was discussed in detail in section 3.5.2 (numerical example 3.4) of this thesis. After parameters (3.23) of NN-ANARX model of this system were obtained by off-line training of the corresponding neural network (the structure of this network is depicted in figure 3.9), dynamic feedback linearization algorithm (4.8)-(4.10) was used as the control technique for this system. This model was included into the adaptive closed-loop system depicted in figure 4.15.

History-Stack Adaptation (HSA) technique was used to adapt the control system by adjusting parameters (synaptic weights) of NN-based ANARX model of the controlled system. The stack length of HSA was set to 200 patterns (n_p = 200) and 20 cycles were used to update the weights on each time-step (n_c = 20). For the first eight hundred time-steps system was simulated without disturbances, starting at time-step 800 the output disturbance was added followed by the input disturbance starting at time-step 1600, repeating sequence of set points was used as reference signal $v(t)$. Simulation results of on-line training in closed-loop with dynamic output linearizing feedback are shown in Figure 4.16.

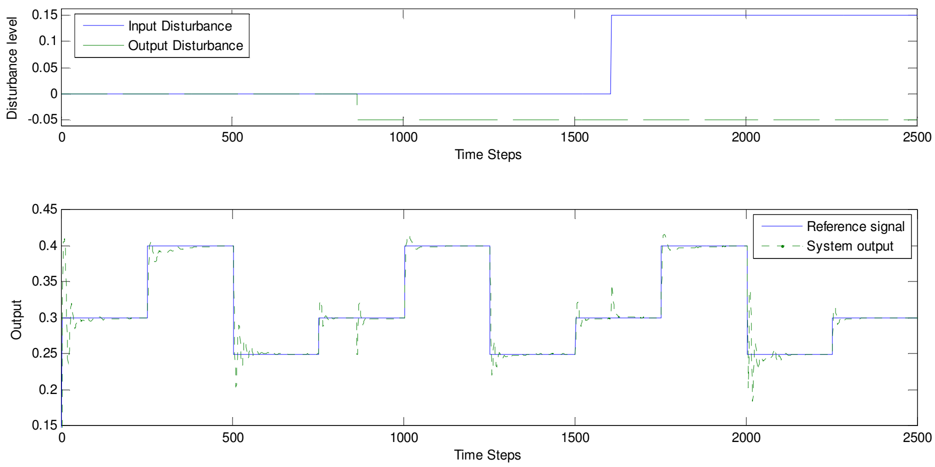

figure 4.16 Control of the liquid level system of interconnected tanks by dynamic output feedback linearization of NN-ANARX model with history-stack adaptation

It is easy to see that even with input and output disturbances NN-ANARX based adaptive control technique provides desired performance and the control system is capable of fast compensating these disturbances because of neural network's ability to learn.

4.4.3 Numerical example 4.6

This example is a continuation of numerical examples 3.5 section 3.5.3) and 4.3 (section 4.3.1).

We have also followed the same procedure as in the previous example for adaptive control of a jacketed Continuous Stirred Tank Reactor (CSTR) [74], [75], [13], [14] represented by equation input-output (3.24).

Identification of this system by second order NN-ANARX model (3.25) was shown in section 3.5.3 (numerical example 3.5) of this thesis. Parameters (3.26) of this model were used for nonadaptive control of this system by ANARX model based dynamic output feedback linearization as shown in section 4.3.1 (numerical example 4.3).

To make this system adaptive to changes in environmental conditions and/or parameters of the system itself, HAS was introduced into the closed loop as shown in figure 4.15 for on-line adjustment of parameters of the model. Parameters of the controller are also adjusted by the same HAS algorithm, because parameters of the model are then used as parameters of the controller.

The closed-loop system (see figure 4.15) was simulated for 1500 time steps with repeating sequence of set points as reference signal $v(t)$. During the first 350 time-steps there was no disturbance at all, then the output disturbance was added followed by the input disturbance at time-step 935. Simulation results of on-line training in closed- loop with dynamic output linearizing feedback are shown in Figure 4.17.

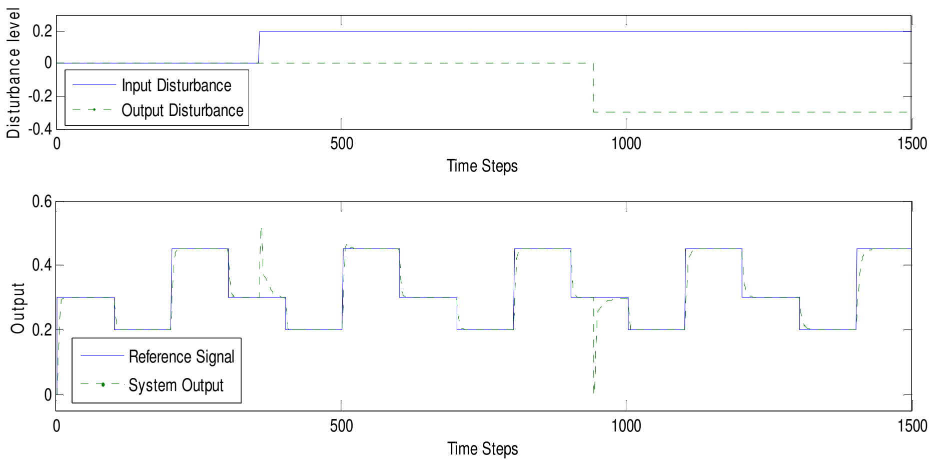

figure 4.17 Control of the jacketed CSTR by dynamic output feedback linearization of NN-ANARX model with history-stack adaptation

The figure shows that in case of adaptive feedback the system output closely follows the reference signal and the control system is capable of compensating the negative influence of input and output disturbances by including them into the neural network based model.

It can be concluded that the proposed adaptive feedback controller for nonlinear system modeled by NN-ANARX structure is capable of modifying its behavior in response to changes in the dynamics of the process and disturbances. Thus, the proposed control technique satisfies the requirements of adaptive controllers [42], [43].

4.5 Drawbacks and advantages of considered control techniques

The main advantages and disadvantages of the considered control techniques determined by the author will be given in the following to compare the methods. First of all lets consider predictive control algorithm described in 2.4.1 and 4.2.1.

The main advantage of predictive control technique is that it can use any type of direct (not inverse) model of the process. This algorithm is also very robust especially when a neural network based model is used. This algorithm cam very easily adapt by adjusting the model. For example, by training the neural network.

The main drawback of this algorithm is that it needs a very accurate model that is capable of working not only in a closed loop but also independently of the system to predict the behavior of the plant.

Consider an input-output discrete time model as defined in (2.7). Output of the model serves as predicted output of the system $\hat{y}(t+1)$ and can be calculated as

$$\hat{y}(t+1) = f_m(y(t), y(t-1)\ldots, y(t-n+1), u(t), \ldots, u(t-m+1)) \qquad (4.21)$$

where f_m is a nonlinear static function of $n+m$ arguments representing the model. $y(t), y(t-1), \ldots, y(t-n+1), u(t), \ldots, u(t-m+1)$ are known previous system output and input values. As we need more predictions, the next predicted output can be calculated as

$$\hat{y}(t+2) = f_m(\hat{y}(t+1), y(t)\ldots, y(t-n+2), u(t), u(t), \ldots, u(t-m+2)) \qquad (4.22)$$

and so on.

Output of the system $y(t+1)$ can not be used, because it is unknown. Previous prediction $\hat{y}(t+1)$ has to be used instead of it and so on N_2 times where N_2 is the prediction horizon as defined in (2.20). It was explained in section 2.3 that there is some small mistake (inaccuracy) in calculation each prediction. In case of some models it accumulates very fast. This problem is especially important with nonlinear systems. Some nonlinearities have such nature that previous state of the system has to be known.

We also suppose that control signal (input of the system) remains the same during the prediction $u(t)$ is used instead of $u(t+1)$ when $\hat{y}(t+2)$ is calculated. It is not so, because in the next time step this control signal will be recalculated. Change in $u(t)$ will not be very serious because minimizing function (2.3), but it can seriously influence the prediction in case of high nonlinearities.

All of the above mentioned reasons can result in a very inaccurate prediction $\hat{y}(t+N_2)$ in case of some nonlinear systems and it leads to possible deterioration of control.

Now let's consider model based control described in detail in sections 2.4.2 and 4.2.3.

The main advantage of this approach is simplicity of the control system. It is very easy to implement this technique to nonlinear system control and it is one of the most natural and obvious approaches from a human point of view. An adaptive control system can be very easily designed especially when a neural networks based model is used as it is shown in [26].

On the other hand, serious questions arise regarding the robustness and stability of direct inverse model based control [23]. This problem can be overcome to some extend by using inverse models belonging to the class of NARX models (2.12) with feedback from the input and output of the controlled system. When neural network is used as an inverse model, the robustness of the control system can also be improved by using an adaptation technique based on reinforcement learning minimizing the difference between the reference signal and the output of the system as it was shown by the author in [26] and section 4.2.4 of this thesis.

Despite the fact that this simple tool have become a very popular in practical control applications especially in robotics [23], the necessity to obtain this inverse of a nonlinear dynamic function is one of the main drawbacks of this technique. It is not always an easy task (if it is possible) to obtain an accurate enough inverse model of a nonlinear dynamic system. This fact significantly restricts the class of systems to which this control technique can be applied.

NN-ANARX model based Dynamic Output Feedback Linearization discussed in sections 4.3 and 4.4 is a control technique combining advantages of classical output linearization and neurocontrol. It is proofed in [32] that ANARX model (2.13) is always linearizable by dynamic output feedback (2.36)-(2.38) providing $y(t)=v(t-n)$, where $v(t)$ is a reference signal (desired output), $y(t)$ is the output of the controlled system and n is the order of ANARX model. Because of neural network's good approximation capabilities, ANARX can be very easily obtained by training a neural network (3.18) representing ANARX structure and depicted in figure 3.9. Experiments [13], [14], [57], [76] have shown that restricted connectivity of the neural network representing ANARX model does not cause serious drawbacks in quality of identification and precise NN-ANARX models can be obtained for a wide class of nonlinear systems. Thus, after the NN-ANARX model is obtained, its parameters are used to construct the dynamic output feedback linearization. This dynamic linearizing is used as a controller for the nonlinear system.

This approach allows to overcome some of disadvantages of predictive and inverse model based control techniques mentioned above. There is no necessity to obtain an inverse model of a dynamic system. The method is based on direct model of the controlled system. The model is used only in closed loop. It does not have to work separately of the modeled system as it is in case of predictive control (for calculation of predictions). Control algorithm (4.8)-(4.10) used in NN-ANARX model linearization based control system depicted in figure 4.7 utilizes only current inputs $u(t)$and outputs $y(t)$ from the controlled system. No external delays as in case of NN-NARX model based control is needed. All the necessary delays are provided by the dynamics of the controller.

Sub-layers of NN-ANARX model are used to provided the dynamics of the controller as shown by equation (4.9). Thus, one more advantage of the proposed technique is that the order of NN-ANARX model and consequently the order of the controller based on it can be very easily changed. When neural network based ANARX model (3.18) shown in figure 3.9 is used, additional sub-layer

$$f_{n+1}\left([y(t),u(t)]^T\right)=C_{n+1}\varphi_{n+1}\left(W_{n+1}\cdot[y(t),u(t)]^T\right) \tag{4.23}$$

Can be added to the network without need to make any changes in the structure of previous already trained network. It is also not necessary to retrain the already trained sub-layers. After adding a new sub-layer (4.23), the model of the order $n+1$ is expressed by the following equation

$$y(t)=\sum_{i=1}^{n+1}C_i\varphi_i\left(W_i\cdot[y(t),u(t)]^T\right). \tag{4.24}$$

Parameters $W_1,\ldots,W_n,C_1,\ldots,C_n$ remain the same and parameters W_{n+1} and C_{n+1} are added. After that the training may continue from the same point.

When the order of the model is changed, the order of the controller can also be changed very easily. In case of considered NN-ANARX model linearization based control, the order of the controller can be increased by simple changing the last term

$$\eta_{n-1}(t+1)=\eta_n(t)-C_n\varphi_n\left(W_n\cdot[y(t),u(t)]^T\right) \tag{4.25}$$

and adding the following terms

$$\eta_n(t+1)=v(t)-C_{n+1}\varphi_{n+1}\left(W_{n+1}\cdot[y(t),u(t)]^T\right) \tag{4.26}$$

to the system of equations (4.9) representing the dynamics of the controller. The first layer

$$C_1\varphi_1\left(W_1 \cdot [y(t), u(t)]^T\right) \tag{4.27}$$

used for calculation of control signals by (4.10) remains the same and synaptic weights of this sub-layer do not change abruptly even when online adaptation (training) is used and therefore the order of the model and control system can be adjusted even online giving additional possibilities for NN-ANARX model based adaptive control of nonlinear systems.

Thus, NN-ANARX model dynamic output feedback linearization based control technique can be a reasonable choice to control of a wide class of nonlinear systems [13], [14], [56], [57], [76]. The only problem is complexity of calculation of the control signal $u(t)$ by (4.8), (4.10) from the dynamics of the controller (4.9). Newton's method (see section 2.4.4) was used in [13], [14], [76] for numerical solving of equation (4.8) and calculation of control signal. The problem is that this method can be successfully applied to only nonlinear SISO systems and our experiments [13], [14] have shown that on each time step it needs about 3 iterations converge making the response of the controller slower (longer minimal allowed sample time). In case of MIMO systems numerical optimization becomes extremely complex. This problem was solved by the author of this thesis as will be shown in the next chapter. Two techniques making this making this control algorithm faster and applicable to a wide class of nonlinear systems are proposed in [56] and [57].

4.6 Conclusions

Three neurocontrol techniques are considered in this section. This techniques are based on models obtained by training a neural network or a system of neural networks.

Model based predictive neurocontrol and inverse model based neurocontrol are the most popular neural networks based control techniques. They have found their applications in variety of modern time practical control applications. References to the corresponding examples are provided in the beginning of this chapter.

These control techniques can utilize any structure of the model. It was shown that the proper choice of the structure can significantly improve the quality of the control system by better describing the controlled system. Neural networks give as the great possibility to choose the structure of the model corresponding to the structure of the system by varying the connections between neurons in the network. Neural Networks based Hammerstein model is applied to control of DC servo motor with nonlinear driver [13].

The quality of inverse model based control significantly depends on the quality of inverse model of controlled dynamic system. Neural network based inverse model can be obtained by training any structure of neural network which is capable of representing an inverse dynamics of the system. Robustness of inverse model based neurocontrol can be improved by introducing an adaptation based on on-line training of the neural network representing the inverse model of the controlled system and used as the controller for this system [26].

NN-ANARX model based Dynamic Output Feedback Linearization is a combination of classical output linearization and neural network based control. This neurocontrol algorithm is based on the model, which has to have ANARX structure. It means that this method requires particular structure of neural network based model depicted in figure 3.9.

Experimental results have shown that this control technique is less dependant on the quality of the model. Structural restrictions imposed by ANARX structure on connectivity matrixes of the network do not cause any drawbacks in quality of the control system [13], [14]. ANARX model is always reprezentable in classical state-space form which makes analysis of the control system much simpler compared to control systems based on other structures. The order of the model and the corresponding control system can be changed very easily. This control technique has strong robustness and can be applied to a wide class of nonlinear system [13], [14], [56], [57], [76].

Because of neural networks ability to learn, History-Stack Adaptation technique can be applied to adjust parameters of NN-ANARX model in response to disturbances, changing environmental conditions and/or changes in parameters of the controlled system. Thus, dynamic output feedback linearization technique can be combined with neural networks based adaptation [14]. Because of restricted connectivity, neural network based ANARX model has much smaller number of parameters (synaptic weights) than a conventional neural network based model and there fore better suites for adaptive control. Adaptation is faster because of smaller number of parameters to be adjusted on-line.

Control system based on dynamic output feedback linearization of neural network based ANARX model combines advantages of classical dynamic output linearization and neural network based approach. It can be a reasonable choice for control of a wide class of nonlinear systems. The only problem is complexity of calculation of control signals. Two methods for solving this problem and applying this technique to control of nonlinear MIMO systems are proposed by the author in [56] and [57] and will be discussed in detail in the next chapter of the thesis.

Chapter 5

NN-ANARX model based control of nonlinear MIMO systems

It was shown in the previous chapter that NN-ANARX model based dynamic output feedback linearization technique is a reasonable choice for control of a wide class of nonlinear systems. It has several advantages over others control techniques considered in the previous chapter, but at the same time it suffers from one serious drawback. Namely, it is not always an easy task to calculate exact values of control signals from the dynamics (4.9) of the controller. An inverse of a nonlinear function (4.8) has to be calculated.

Newton's method can be used for numerical calculation of control signal $u(t)$ by numerical solving of equation (4.8) in a current working point (for each current state of the control system). This approach proposed in [13] and [14]can be applied to only SISO models and on each time step it needs about 3 iterations to converge.

For modeling of nonlinear MIMO systems, MIMO NN-ANARX model (3.37) shown in figure 3.15 was proposed in [56] and [57]. This model belongs to the class of Additive NARX models, where inputs and outputs are vectors. By using these models, dynamic output feedback linearization technique can be applied to control of nonlinear MIMO systems, but practical application of this control method is complicated by calculation of a vector of control signals. In case of nonlinear MIMO systems numerical solving of equation (4.8) becomes extremely complex and practice has shown that classical numerical nonlinear optimization algorithms may not converge. Two alternative methods were proposed by the author to solve this problem and to apply dynamic output feedback linearization technique to control of nonlinear systems in [56] and [57].[58]

Neural network based Simplified ANARX (NN-SANARX) structure is proposed [56]. By using this sub-class of ANARX models, calculation of a control signal or a vector of control signals becomes very simple and come to solving of a linear equation or a system of linear equations. Linear independence of these equations can be controlled after training the network.

The second method proposed in [57] is based on introducing an additional static neural network for calculation of control signals by approximating function (4.10). In this case not simplified NN-ANARX structure can be used for identification of controlled system, which makes this control technique applicable to a wider range of nonlinear systems.

Both proposed methods will be discussed in detail in this section. The effectiveness of these approaches will be demonstrated on numerical examples.

5.1 Author's contribution

All the topics discussed in this chapter belong to the contribution of the author of this thesis.

- Simplified NN-based ANARX structure is proposed in [56];
- Capabilities of this structure for identification of nonlinear SISO and MIMO systems is demonstrated [56];
- By using NN-SANARX structure of the model, dynamic output feedback linearization algorithm is applied to control of nonlinear MIMO systems [55][56];
- Additional static neural network based approach is proposed in [57] for significant simplification of practical application of NN-ANARX model linearization based control technique;
- The proposed control scheme consisting of NN-ANARX model based linearizing output feedback and additional static neural network is applied to control of nonlinear MIMO systems [57];
- The effectiveness of the proposed techniques is demonstrated on examples.

5.2 Problem statement

The main problem regarding practical application of dynamic output feedback linearization algorithm (4.8)-(4.10) is in calculation of control signals $u(t)$ from equation (4.10) or by solving equation (4.8). As nonlinear sigmoid activation function $\varphi_1(\cdot)$ is used as activation function in the first sub-layer $C_1\varphi_1\left(W_1 \cdot [y(t), u(t)]^T\right)$ of neural network (3.18) representing ANARX model depicted in figure 3.8, equation (4.8) can not be solved analytically or the solution is dramatically complex. Thus, control signals have to be calculated numerically. Most numerical algorithms are iterative and need some time to converge. It delays the response of the controller and may significantly reduce the speed of the control system by increasing the minimal possible sample time of the discrete time control system.

It was shown in [13] and [14] that in case of nonlinear SISO systems, the value of the function F^{-1} in (4.10) can be numerically calculated by Newton's method and 3 iterations guarantee sufficient precision. When we have to calculate numerous controls this task becomes extremely complex. Inverse of the function of several arguments can not be calculated numerically fast enough to satisfy the needs of the control system. Practically it means that in many cases numerical nonlinear optimization algorithms do not converge or the result is not precise and can not be used as a control signal. Because of the problem described above the algorithm can not be applied to systems with multiply inputs.

Neural network based ANARX structure for modeling of nonlinear MIMO systems is proposed in [56]and [57]. It is depicted in figure 3.15 and can be formalized by the following equation

$$[y_1(t),\dots,y_m(t)]^T = \sum_{i=1}^{n} C_i \cdot \varphi_i\left(W_i \cdot [y_1(t-i),\dots,y_m(t-i),u_1(t-i),\dots,u_r(t-i)]^T\right), \tag{5.1}$$

where m and r is the number of outputs and inputs of the model, W_i and C_i are matrixes of synaptic weights, $\varphi_i(\cdot)$ is nonlinear activation of neurons in the corresponding sub-layer of the hidden layer and n is the order of the model.

When dynamic output feedback linearization algorithm (4.8)-(4.10) is applied to control of nonlinear MIMO systems by using MIMO NN-ANARX model (5.1), the vector of control signals has to be calculated by solving the following system of equations.

$$F = C_1\varphi_1\left(W_1 \cdot [y_1(t),\dots,y_m(t),u_1(t),\dots,u_r(t)]^T\right) = \eta_1(t) \tag{5.2}$$

where $\eta_i(t) = [\eta_{i,1}(t),\dots,\eta_{i,m}(t)]^T \quad i = 1,\dots,n-1$ are vectors of inner states of controller (4.9) at time step t. The size of this vector corresponds to the size of the output vector, because output feedback is used.

Thus,

$$[u_1(t),\dots,u_r(t)]^T = F^{-1}\left(y_1(t),\dots,y_m(t),\eta_{1,1}(t),\dots,\eta_{1,m}(t)\right) \tag{5.3}$$

The aim of the researches discussed in this chapter is to solve these problem by simplification of practical application of the control algorithm providing methods for fast calculation of controls (4.10), (5.3) and to apply ANARX structure based dynamic output feedback linearization algorithm to control of nonlinear MIMO systems.

5.3 NN-based Simplified ANARX structure

The problem of calculating the inverse of function F in (4.8) and (5.2) stated above can be solved by imposing one more restriction on NN-ANARX structure and introducing a new subclass of ANARX models, where the first sub-layer in neural network representing ANARX model (see figure 3.8) is linear. It means that function $\varphi_1(\cdot)$ in (3.18) and (5.1) is a linear function corresponding to ADALINE neurons of the first sub-layer of the corresponding ANARX-type neural network:

$$\varphi_1(W_1 \cdot Z(t-1)) = W_1 \cdot Z(t-1), \tag{5.4}$$

where

$$Z(t) = [y(t), u(t)]^T \tag{5.5}$$

in case of SISO models and

$$Z(t) = [y_1(t), \ldots, y_m(t), u_1(t), \ldots, u_r(t)]^T \tag{5.6}$$

for MIMO models having r inputs and m outputs.

This type of models (sub-class of NN-ANARX models) called Simplified NN-ANARX models (or NN-SANARX models) was proposed by the author in [56] for simplification of calculation of control signals by control algorithm (4.8)-(4.10) based on this model. If follows from (3.18 and (5.4) that NN-SANARX model can be formalized by the following equation

$$y(t) = C_1 \cdot W_1 \cdot Z(t-1) + \sum_{i=2}^{n} C_i \varphi_i (W_i \cdot Z(t-i)) \tag{5.7}$$

Such a restriction can guarantee that control $u(t)$ can be easily calculated from (4.8) or (5.2). It is obvious from (5.7) that we have to have at least second or higher order of the model for identification of nonlinear systems ($n \geq 2$). It was also shown in [55] that dividing linear and nonlinear parts of neural network based model does not cause drawbacks in the quality of identification and may give advantages in some control applications.

5.3.1 NN-SANARX structure for control of nonlinear systems

NN-SANARX models belong to the class of ANARX models. It means that ANARX based Dynamic Output Feedback Linearization (see section 2.4.3) can be applied to control of nonlinear systems identified by NN-SANARX structure.

As follows from equations (4.9) and (5.7), NN-SANARX model based Dynamic Output Feedback Linearization algorithm can be represented by the following equations

$$F = C_1 \cdot W_1 \cdot [y(t), u(t)]^T = \eta_1(t) \tag{5.8}$$

$$\begin{aligned} &\eta_1(t+1) = \eta_2(t) - C_2\varphi_2\left(W_2 \cdot [y(t), u(t)]^T\right) \\ &\vdots \\ &\eta_{n-2}(t+1) = \eta_{n-1}(t) - C_{n-1}\varphi_{n-1}\left(W_{n-1} \cdot [y(t), u(t)]^T\right), \\ &\eta_{n-1}(t+1) = v(t) - C_n\varphi_n\left(W_n \cdot [y(t), u(t)]^T\right) \end{aligned} \tag{5.9}$$

where n is the order of the model (number of sub-layers), number of nonlinear sub-layers is n-1, $v(t)$ is the reference (desired output).

When linearizing feedback (5.8)-(5.9) is applied to corresponding NN-SANARX model (5.7), $y(t) = v(t-n)$ if exact control $u(t)$ is calculated from (5.8). It is much easier than calculation of $u(t)$ from (4.8) as it is in case of systems identified by NN-ANARX models.

Let's now define matrix

$$T = C_1 \cdot W_1 \tag{5.10}$$

Matrix T can be divided into two parts $T = [T_1 \quad T_2]$ so that

$$C_1 \cdot W_1 \cdot Z(t-1) = T \cdot Z(t-1) = T_1 \cdot y(t-1) + T_2 \cdot u(t-1). \tag{5.11}$$

Function (5.8) can now be represented as

$$F = T_1 \cdot y(t) + T_2 \cdot u(t) = \eta_1(t) \tag{5.12}$$

and control signal $u(t)$ in (4.10) can be calculated by the following equation

$$u(t) = T_2^{-1}\left(\eta_1(t) - T_1 \cdot y(t)\right) \tag{5.13}$$

if T_2 is a nonsingular square matrix.

It has to be mentioned that in case of SISO systems $T \in \Re^2$ is a 2×1 vector and $T_2 \in \Re$ is a real number. So, the proposed control technique can be applied if after training the neural network $T_2 \neq 0$. It means that input $u(t-1)$ has nonzero influence on the output of NN-SANARX model $\hat{y}(t)$.

Before applying the proposed technique to control of nonlinear MIMO systems, consider the following numerical example of NN-SANARX model based identification and control of a nonlinear SISO system. To compare with the previous results.

5.3.2 Numerical example 5.1

The model of a liquid level system of interconnected tanks [72], [14] is represented by discrete time input-output equation (3.21). This system was identified by NN-ANARX structure (3.22) with 3 sub-layers of the neural network representing this structure (see figure 3.9) corresponding to the third order of the model. Identification of system of interconnected tanks (3.21) by Neural Network based ANARX model was discussed in section 3.5.2 of this thesis and in [14]. Control of this system by ANARX model based dynamic output feedback linearization was demonstrated in section 4.4.2 of this thesis and in [14]. Here, Neural Network based Simplified ANARX model (5.7) will be used for identification of system (3.21) and control of this system by dynamic output feedback linearization of obtained NN-SANARX model will be shown.

To obtain parameters of the model (5.7) the network was trained with Levenberg-Marquardt algorithm (see section 2.6.2) as neural networks representing both ANARX and Simplified ANARX models are restricted connectivity networks and LM training algorithm is much more efficient compared to other techniques when the network contains a small number of synaptic weights [53].

Neural network with three sub-layers corresponding to the third order model (*n=3*) and with tree neurons on each sub-layer of the hidden layer (*l=3*) was trained. Activation functions of the second and third sub-layer neurons ($\phi_2(\cdot)$ and $\phi_3(\cdot)$) were smooth logistic sigmoid functions (2.50). Thus, system (3.21) was identified by a third order Neural Network based Simplified ANARX model represented by the following equation

$$y(t) = C_1 \cdot W_1 \cdot \left[y(t-1), u(t-1)\right]^T + \sum_{i=2}^{3} C_i \varphi_i \left(W_i \cdot \left[y(t-i), u(t-i)\right]^T\right) \qquad (5.14)$$

Identified parameters of model (5.14) have the following values:

$$W_1 = \begin{bmatrix} 0.9195 & 0.2004 \\ 0.9025 & 0.2185 \\ -0.8592 & -0.7844 \end{bmatrix}, \; W_2 = \begin{bmatrix} 0.6558 & 0.2184 \\ 0.0371 & 0.1934 \\ 0.3225 & 0.0192 \end{bmatrix},$$

$$W_3 = \begin{bmatrix} 0.8405 & 1.0312 \\ 0.0298 & 0.0859 \\ 1.0043 & -7.0306 \end{bmatrix} \tag{5.15}$$

$$C_1 = [\, 0.6197 \quad 0.2042 \quad -0.7288 \,]$$
$$C_2 = [\, 19.4187 \quad -33.7707 \quad -28.8544 \,]$$
$$C_3 = [\, -5.6447 \quad 49.0729 \quad -0.2213 \,]$$

Model validation on the different data set (validation set) shows nearly excellent overlap of the model and plant outputs (see Figure 5.1).

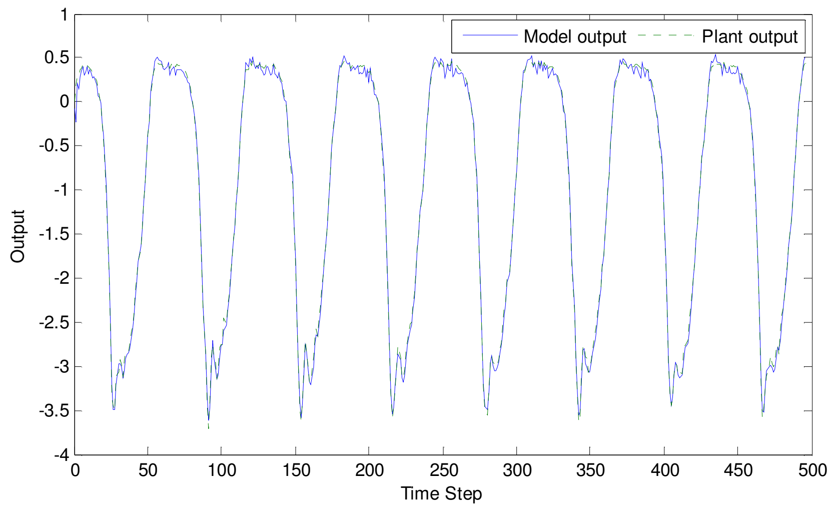

Figure 5.1 Identification results: output of plant (3.21) vs. output calculated by its NN-SANARX model

Mean Square Error on this validation set as low as about $3 \cdot 10^{-3}$.

It can be concluded that strong restrictions imposed by linearity of the first sub-layer in NN-SANARX structure (5.14) does not cause any drawbacks in quality of identification of a liquid level system of interconnected tanks (3.21). The same validation set was used for validation of the model of this system represented by NN-ANARX structure (3.22). See section 3.5.2, numerical example 3.4. It can be seen by comparing these results that MSE on the validation set is about the same and utilizing Simplified ANARX structure may even result in a little bit better quality of the model ($3\cdot10^{-3}$ vs. $4\cdot10^{-3}$), but, of course, the difference is very small and also depends on random choice of initial values of synaptic weights.

NN-SANARX model based Dynamic Output Feedback Linearization algorithm (5.8)-(5.9) was used for control of this system. It follows from (5.10) and (5.15) that

$$T=\begin{bmatrix}0.6197 & 0.2042 & -0.7288\end{bmatrix}\cdot\begin{bmatrix}0.9195 & 0.2004\\ 0.9025 & 0.2185\\ -0.8592 & -0.7844\end{bmatrix}=\begin{bmatrix}1.3803 & 0.7405\end{bmatrix} \qquad (5.16)$$

and according to equation (5.11)

$$T_1=1.3803,\ T_2=0.7405. \qquad (5.17)$$

$T_2\neq 0$ and thus, according to (5.13) control signal $u(t)$ can be calculated as

$$u(t)=1.3504\cdot\left(\eta_1(t)-1.3803_1\cdot y(t)\right), \qquad (5.18)$$

where $y(t)$ is the current output of the controlled system and $\eta_1(t)$ comes from the dynamics (5.9) of the controller represented for model (5.14)-(5.15) by the following equations

$$\eta_1(t+1)=\eta_2(t)-\begin{bmatrix}19.4187 & -33.7707 & -28.8544\end{bmatrix}\cdot\varphi_2\left(\begin{bmatrix}0.6558 & 0.2184\\ 0.0371 & 0.1934\\ 0.3225 & 0.0192\end{bmatrix}\cdot\begin{bmatrix}y(t)\\ u(t)\end{bmatrix}\right)$$

$$\eta_2(t+1)=v(t)-\begin{bmatrix}-5.6447 & 49.0729 & -0.2213\end{bmatrix}\cdot\varphi_3\left(\begin{bmatrix}0.8405 & 1.0312\\ 0.0298 & 0.0859\\ 1.0043 & -7.0306\end{bmatrix}\cdot\begin{bmatrix}y(t)\\ u(t)\end{bmatrix}\right)$$

(5.19)

Control algorithm (5.18)-(5.19) was applied to control of this plant by closed loop control system corresponding to the scheme depicted in figure 4.7. The control system was simulated with piece-constant reference signal $v(t)$. Simulation result is depicted in figure 5.2.

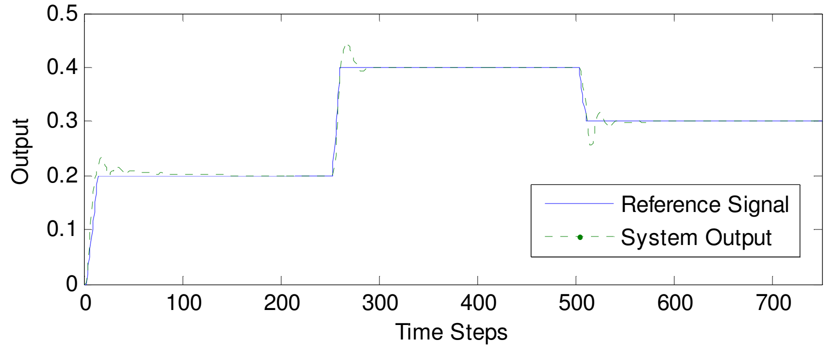

figure 5.2 Control of a liquid level system of interconnected tanks by NN-SANARX based dynamic output linearization

and the corresponding control signal calculated by (5.18) is presented in figure 5.3.

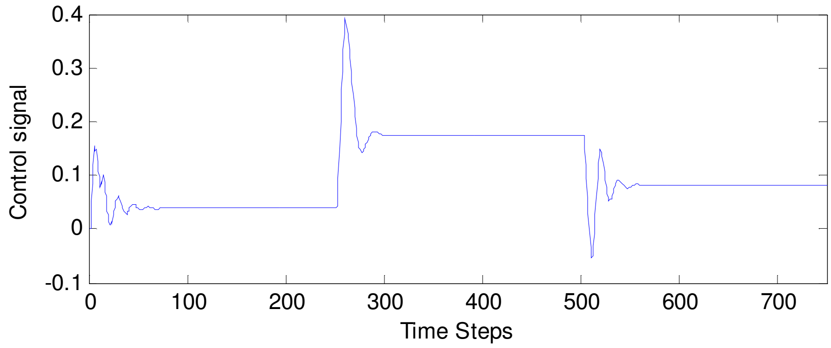

figure 5.3 Control of a liquid level system of interconnected tanks: control signal

It can be seen from the figure that the output of the system closely follows the reference signal. This control system shows the same good performance as NN-ANARX based control discussed in sections 4.3 and 4.4, but practical application is much simpler and more convenient when NN-SANARX model is used. Since inverse function can be computed analytically no numerical calculation is necessary. Control signal can be computed by solving linear equation (5.12). Computation of control signal is fast and does not require any time for convergence. Moreover, there is a certain criterion for defining applicability of each particular model for dynamic output feedback linearization: $T_2 \neq 0$ or T_2 is a nonsingular matrix.

In the next section an application of the discussed technique to control of nonlinear MIMO systems will be shown.

5.4 NN-SANARX structure based control of nonlinear MIMO systems

Structure of Neural Network representing Additive Nonlinear AutoRegressive eXogenous model for Multiply Input – Multiply Output systems having r inputs and m outputs was proposed in [56], [57] and presented in figure 3.15 (See section 3.7.2). MIMO NN-ANARX model is represented by expression (5.1).

Neural Network based Simplified ANARX structure proposed by the author in [56] has linear activation functions of neurons of the first sub-layer

$$\varphi_1\left(W_1 \cdot \left[y_1(t),\ldots,y_m(t),u_1(t),\ldots,u_r(t)\right]^T\right) = W_1 \cdot \left[y_1(t),\ldots,y_m(t),u_1(t),\ldots,u_r(t)\right]^T \tag{5.20}$$

and is a sub-class of ANARX models. MIMO NN-SANARX model can be formalized by the following equation

$$\begin{aligned}\left[y_1(t),\ldots,y_m(t)\right]^T &= C_1 \cdot W_1 \cdot \left[y_1(t-1),\ldots,y_m(t-1),u_1(t-1),\ldots,u_r(t-1)\right]^T + \\ &+ \sum_{i=2}^{n} C_i \cdot \varphi_i\left(W_i \cdot \left[y_1(t-i),\ldots,y_m(t-i),u_1(t-i),\ldots,u_r(t-i)\right]^T\right)\end{aligned} \tag{5.21}$$

After identification of parameters of model (5.21), control algorithm (5.8)-(5.9) with vectors

$$u(t) = \left[u_1(t),\ldots,u_r(t)\right] \tag{5.22}$$

$$y(t) = \left[y_1(t),\ldots,y_m(t)\right] \tag{5.23}$$

$$\eta_i(t) = \left[\eta_{i,1}(t),\ldots,\eta_{i,m}(t)\right], \text{ for } i = 1,\ldots,n-1 \tag{5.24}$$

$$v(t) = \left[v_1(t),\ldots,v_m(t)\right] \tag{5.25}$$

can be applied to control of nonlinear MIMO system identified by this model.

Matrixes of synaptic weights of model (5.21) have the following dimensions: $W_i \in \Re^{l_i \times (m+r)}$ and $C_i = \Re^{m \times l_i}$, where l_i is the number of neurons in the corresponding i-th sub-layer of the restricted connectivity network depicted in figure 3.15.

According to (5.10), $T \in \Re^{m \times (m+r)}$ and when divided into two parts as shown in (5.11), $T_1 \in \Re^{m \times m}$ is always a square matrix and $T_2 \in \Re^{m \times r}$. To calculate vector of control signals (5.22), system of linear equations (5.12) has to be solved. It can be represented as

$$T_2 \cdot [u_1(t),\ldots,u_r(t)]^T = [\eta_{1,1}(t),\ldots,\eta_{1,m}(t)]^T - T_1 \cdot [y_1(t),\ldots,y_m(t)]^T\,, \tag{5.26}$$

where $[\eta_{1,1}(t),\ldots,\eta_{1,m}(t)]^T$ is the vector of the controller's first inner state obtained from the dynamics of the controller (5.9) in each time step of the discrete time control system.

It can be seen that system of linear equations (5.26) has a solution if and only if $rank(T_2) \geq m$. It does not depend on the values of vector of the controller's inner states (5.24) changing in time and can be very easily checked after training the network and obtaining parameters of model (5.21). Matrix T_2 is a constant and depends only on parameters of the first sub-layer of the model. It means that after training the network, parameters of its first sub-layer should not be changed and can not be adjusted on-line during the control process. It imposes additional restrictions on NN-SANARX model based adaptive control.

In this research, systems with equal sizes of input and output vectors are considered. If $r = m$, T_2 is a square matrix $T_2 \in \Re^{m \times m}$ and the criteria for applicability of dynamic output feedback linearization to control of nonlinear MIMO system represented by NN-SANARX model is nonsingularity of matrix T_2. In this case vector of controls can always be calculated as

$$[u_1(t),\ldots,u_r(t)]^T = T_2^{-1}\left([\eta_{1,1}(t),\ldots,\eta_{1,m}(t)]^T - T_1 \cdot [y_1(t),\ldots,y_m(t)]^T\right) \tag{5.27}$$

and algorithm (5.9), (5.27) can be applied to control of nonlinear MIMO systems identified by Neural Network based Simplified ANARX model (5.21). The aim of this control system is to track the vector of reference signals (5.25). The structure of the corresponding closed loop discrete time control system is depicted in figure 5.4

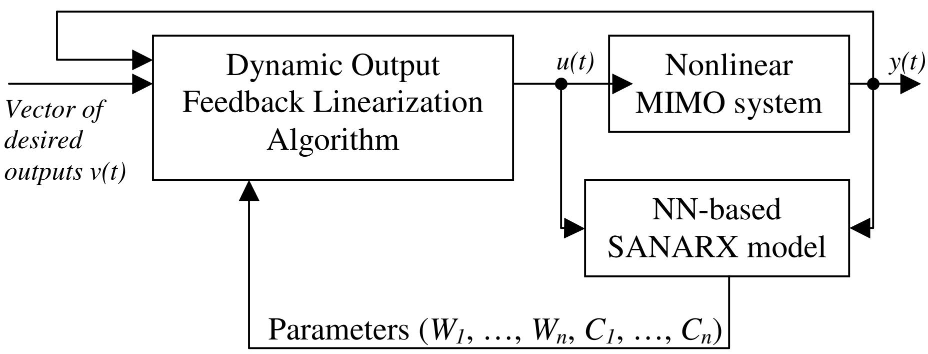

figure 5.4 NN-SANARX model based control system

Consider the following numerical examples of MIMO NN-SANARX structure based identification and control of three discrete and continuous time nonlinear MIMO systems. These experimental results were shown by the author in [56].

5.4.1 Numerical example 5.2

A nonlinear MIMO discrete-time system [113], [114], [56], [57] is represented by the following first order input-output equations

$$
\begin{aligned}
y_1(t+1) &= 0.4y_1(t) + \frac{u_1(t)}{1+u_1^2(t)} + 0.2u_1^3(t) + 0.5u_2(t) \\
y_2(t+1) &= 0.2y_2(t) + \frac{u_2(t)}{1+u_2^2(t)} + 0.4u_2^3(t) + 0.2u_1(t)
\end{aligned}
\tag{5.28}
$$

This system was simulated and a set of input-output data obtained. This training data set was used to train the following NN-SANARX structure

$$
\begin{aligned}
[y_1(t), y_2(t)]^T &= C_1 \cdot W_1 \cdot [y_1(t-1), y_2(t-1), u_1(t-1), u_2(t-1)]^T + \\
&+ C_2 \cdot \varphi_2\left(W_2 \cdot [y_1(t-2), y_2(t-2), u_1(t-2), u_2(t-2)]^T\right)
\end{aligned}
\tag{5.29}
$$

corresponding to the second order of the model ($n=2$). As it was mentioned in section 5.3, minimal order of NN-SANARX model representing a nonlinear dynamic system is 2, because of linearity of the first sub-layer.

The first layer consisted of 2 linear neurons ($l_1 = 2$) and the second layer consisted of 5 nonlinear neurons ($l_2 = 5$) with hyperbolic tangent sigmoid (2.51) activation functions $\varphi_2(\cdot)$. Identified parameters of model (5.29) have the following values

$$
\begin{aligned}
W_1 &= \begin{bmatrix} 0.9239 & -1.2909 & 0.3075 & -0.7308 \\ -1.2669 & -0.2253 & -0.7675 & -0.6805 \end{bmatrix}, \\
W_2 &= \begin{bmatrix} 0.0358 & -0.0214 & -0.0046 & -0.0424 \\ 0.1837 & -0.2258 & -0.0553 & -0.0473 \\ 0.0259 & -0.0167 & -0.0094 & -0.0373 \\ 0.2637 & 0.1549 & 0.0501 & 0.1054 \\ 0.2699 & 0.1615 & 0.0513 & 0.1096 \end{bmatrix}, \\
C_1 &= \begin{bmatrix} 0.1414 & -0.8987 \\ -0.7948 & -0.5827 \end{bmatrix}
\end{aligned}
\tag{5.30}
$$

$$C_2 = \begin{bmatrix} -84.6868 & -0.1466 & 107.9394 & -7.3793 & 6.8476 \\ 1.9867 & -1.6542 & 22.1403 & -27.8546 & 25.9052 \end{bmatrix}$$

Two sinusoidal input signals were used for generating validation set $\{u_v(t), y_v(t) | t = 0,1,\ldots,100\}$, where

$$u_v(t) = (u_{v1}(t), u_{v2}(t))^T, \tag{5.31}$$
$$u_{v1}(t) = 0.4\sin(0.2 \cdot t) + 0.1 \tag{5.32}$$
$$u_{v1}(t) = 0.8\sin(0.1 \cdot t) \tag{5.33}$$

and $y_v(t) = (y_{v1}(t), y_{v2}(t))^T$ is the matrix of corresponding outputs of system (5.28). Validation of model (5.29),(5.30) is shown in figure 5.5.

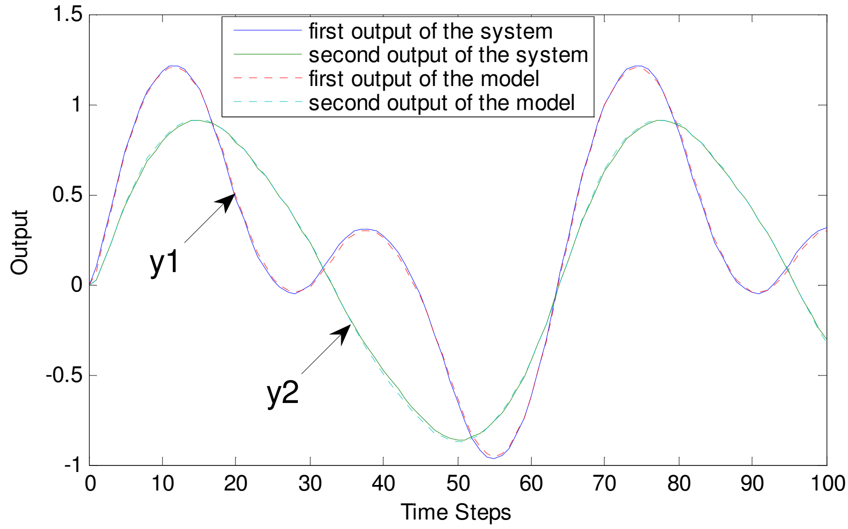

figure 5.5 Identification of nonlinear MIMO system (2.57) by second order NN-SANARX model (5.28),(5.29): model validation

Model validation shows nearly excellent overlap of the system's outputs $y_v(t)$ and outputs of the model $\hat{y}_v(t)$. Mean square error on this validation calculated by (2.19) was as low as about $1.36 \cdot 10^{-4}$. System (5.28) was represented by second order Neural Network based Simplified ANARX structure (5.29) with high accuracy and this model can be used for control system design if matrix T_2 is not singular.

According to (5.10) and (5.30)

$$T = \begin{bmatrix} 1.2720 & 0.0161 & 0.7341 & 0.5061 \\ 0.0039 & 1.1573 & 0.2028 & 0.9773 \end{bmatrix} \tag{5.34}$$

As the system has two inputs and two outputs, it follows from (5.11) that

$$T_1 = \begin{bmatrix} 1.2720 & 0.0161 \\ 0.0039 & 1.1573 \end{bmatrix} \text{ and } T_2 = \begin{bmatrix} 0.7341 & 0.5061 \\ 0.2028 & 0.9773 \end{bmatrix}. \tag{5.35}$$

T_2 is a nonsingular square matrix and

$$T_2^{-1} = \begin{bmatrix} 1.5896 & -0.8231 \\ -0.3299 & 1.1940 \end{bmatrix} \tag{5.36}$$

It means that NN-SANARX model based dynamic output feedback linearization algorithm (5.9),(5.27) can be applied to control of this system as

$$\begin{aligned} &[u_1(t), u_2(t)]^T = T_2^{-1}\left([\eta_{1,1}(t), \eta_{1,2}(t)]^T - T_1 \cdot [y_1(t), y_2(t)]^T\right) \\ &[\eta_{1,1}(t+1), \eta_{1,2}(t+1)]^T = [v_1(t), v_2(t)]^T - C_2\varphi_2\left(W_2 \cdot [y_1(t), y_2(t), u_1(t), u_2(t)]^T\right) \end{aligned} \tag{5.37}$$

Using parameters of the model (5.30) and calculated matrixes (5.35), (5.36), vector of control signals $[u_1(t), u_2(t)]$ for tracking the references $[v_1(t), v_2(t)]$ by nonlinear MIMO system (5.28) can be calculated from the following differential equation representing discrete time controller for this system.

$$\begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = T_2^{-1} \cdot \left(\begin{bmatrix} v_1(t-1) \\ v_2(t-1) \end{bmatrix} - C_2\varphi_2\left(W_2 \cdot \begin{bmatrix} y_1(t-1) \\ y_2(t-1) \\ u_1(t-1) \\ u_2(t-1) \end{bmatrix} \right) - T_1 \cdot \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} \right). \tag{5.38}$$

This control system corresponding to the structure depicted in figure 5.4 with dynamic output feedback linearizing controller represented by differential input-output equation (5.38) was simulated with sinusoidal tracking reference signals $v_1(t)$ and $v_2(t)$. Closed loop simulation results are presented in the next figure.

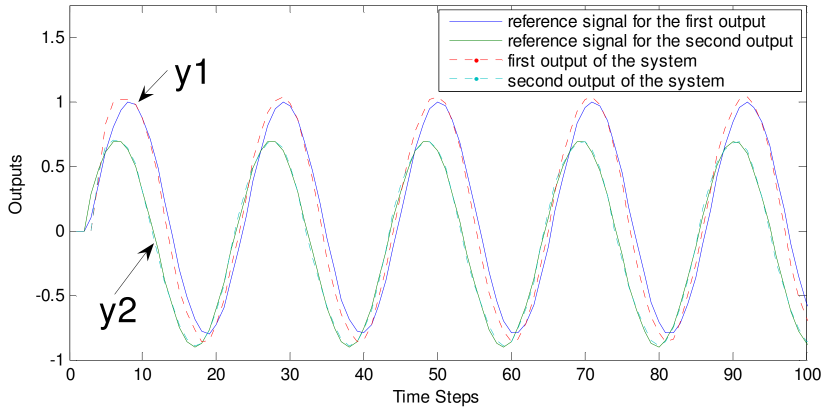

figure 5.6 Closed loop simulation with sinusoidal reference signals

The corresponding control signals $u_1(t)$ and $u_2(t)$ calculated by (5.28) are depicted in figure 5.7.

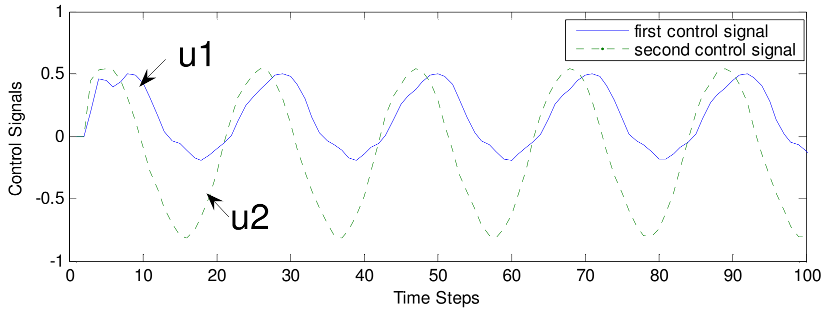

figure 5.7 Control signals for control shown in figure 5.6

The control system was also simulated with piece-constant reference signals. The result of this simulation is depicted in figure 5.8.

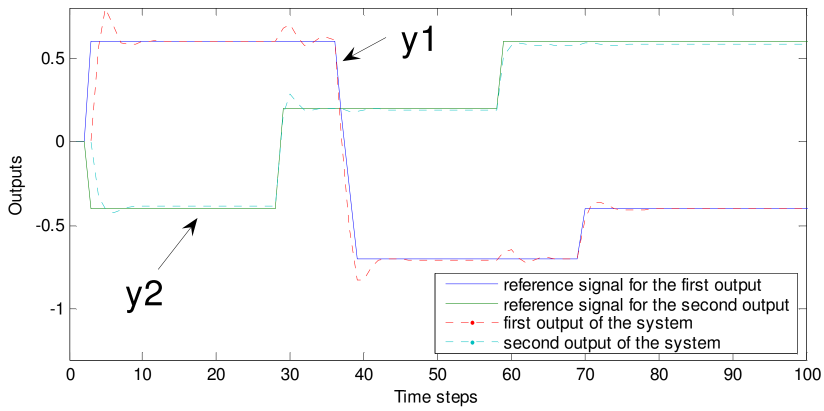

figure 5.8 Closed loop simulation with piece-constant reference signals

It can be seen from figures 5.6 and 5.8 that outputs of the control system $y_1(t)$ and $y_2(t)$ are capable of simultaneous tracking the desired reference signals $v_1(t)$ and $v_2(t)$ correspondingly and the proposed NN-SANARX model based control technique can be successfully applied to control of nonlinear MIMO system (5.28).

5.4.2 Numerical example 5.3

Nonlinear discrete-time system [115], [56] represented by the following input-output equations was also chosen to evaluate the effectiveness of the proposed control algorithm

$$
\begin{aligned}
y_1(t) &= \frac{0.7y_1(t-1)y_1(t-2)}{1+y_1^2(t-1)+y_2^2(t-2)} + 0.3u_1(t-2) + u_1(t-1) + 0.2u_2(t-2) \\
y_2(t) &= \frac{0.5y_2(t-1)\sin\left(y_2(t-2)\right)}{1+y_2^2(t-1)+y_1^2(t-2)} + 0.5u_2(t-2) + u_2(t-1) + 0.2u_1(t-2)
\end{aligned} \tag{5.39}
$$

Experiments have shown that the best accuracy of Neural Network based MIMO Simplified ANARX model (5.21) of system (5.39) can be obtained training network structure corresponding to the third order of the model. It means that neural network of the structure depicted in figure 3.15 has to be trained with 3 sub-layers (first of them is linear and the others are nonlinear).

Unlike system (5.28) shown in the previous example and system to be shown in the next example, this system (5.39) was identified by training third order MIMO NN-SANARX structure (n=3) represented by the following equation

$$
\begin{aligned}
[y_1(t), y_2(t)]^T &= C_1 \cdot W_1 \cdot [y_1(t-1), y_2(t-1), u_1(t-1), u_2(t-1)]^T + \\
&+ \sum_{i=2}^{3} C_i \cdot \varphi_i\left(W_i \cdot [y_1(t-i), y_2(t-i), u_1(t-i), u_2(t-i)]^T\right)
\end{aligned} \tag{5.40}
$$

Two neurons were used in the linear sub-layer ($l_1 = 2$) and 7 neurons having sigmoid Hyperbolic tangent activation function (2.51) in each nonlinear sub-layer ($l_2 = l_3 = 7$) of model (4.50). Matrixes of synaptic weights of the model $W_1 \in \Re^{2\times4}$, $W_2, W_3 \in \Re^{7\times4}$, $C_1 \in \Re^{2\times2}$, $C_2, C_3 \in \Re^{2\times7}$ were obtained by training the corresponding neural network representing MIMO ANARX Structure (see figure 3.15). Because of restricted connectivity of the network and comparatively small number of parameters to be adjusted, Levenberg-Marquardt training algorithm was used to the network. As the modeled dynamic system (5.39) and the corresponding model (5.40) have two inputs and two outputs, $T \in \Re^{2\times4}$ calculated by (5.10) can be divided into two 2×2 matrixes T_1 and T_2

as defined in (5.11), where T_2 is a nonsingular matrix. Thus, according to (5.9) and (5.27), the following second order discrete-time system (5.41)-(5.42) can be used to control nonlinear MIMO system (5.39) by means of parameters of third order MIMO NN-SANARX model (5.40).

$$[u_1(t), u_2(t)]^T = T_2^{-1}\left([\eta_{1,1}(t), \eta_{1,2}(t)]^T - T_1 \cdot [y_1(t), y_2(t)]^T\right) \quad (5.41)$$

$$\begin{aligned} [\eta_{1,1}(t+1), \eta_{1,2}(t+1)]^T &= [\eta_{2,1}(t), \eta_{2,2}(t)]^T - C_2\varphi_2\left(W_2 \cdot [y_1(t), y_2(t), u_1(t), u_2(t)]^T\right) \\ [\eta_{2,1}(t+1), \eta_{2,2}(t+1)]^T &= [v_1(t), v_2(t)]^T - C_3\varphi_3\left(W_3 \cdot [y_1(t), y_2(t), u_1(t), u_2(t)]^T\right) \end{aligned} \quad (5.42)$$

Control system corresponding to the structure of the closed loop system depicted in figure 5.4 was applied to control of system (5.39). The results of the control system simulation with sinusoidal reference signals are shown in the next figure.

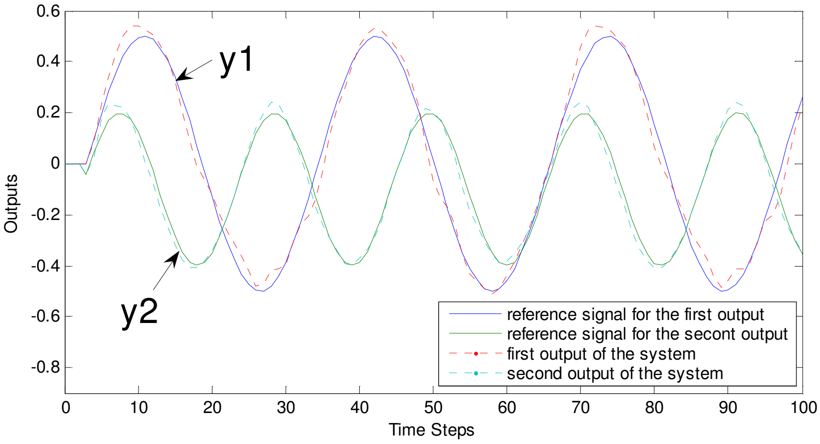

figure 5.9 Control result: output trajectories vs. reference signals

Simulation presented in the figure shows that by using the proposed control technique (5.41)-(5.42), both outputs of system (5.39) are capable of tracking the desired trajectories.

5.4.3 Numerical example 5.4

The third test system (5.43) is a nonlinear third order continuous time MIMO system, which was used as a test system for control algorithms in [116], [117], [118], [119], [56], [57]. The system is represented by the following system of differential equations.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_1 + x_2^2 + x_3 \\ x_1 + 2x_2 + 3x_3x_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 3u_1 + u_2 \\ u_1 + 2(2 + 0.5\sin(x_1))u_2 \end{bmatrix} \tag{5.43}$$

$$y_1 = x_1, \quad y_2 = x_3$$

The considered control system is a discrete-time system and is based on a neural network based discrete time model. This system was simulated and input-output data was sampled with sample time 0.05sec proposed in [119]. This data set was used to obtain a discrete-time MIMO NN-SANARX model (5.21). Neural network having structure shown in figure 3.15 with two sub-layers (n=2) of the hidden layer corresponding to the second order of discrete-time model (5.29) was trained by LM training algorithm. Parameters W_1, W_2, C_1 and C_2 of model (5.29) were obtained. Matrix T was calculated by (5.10). As system (5.43) has two inputs and two outputs $T_1, T_2 \in \Re^{2\times2}$. Nonsingularity of matrix T_2 was checked and control algorithm (5.38) was applied to the system. The structure of the corresponding control system is depicted in figure 5.4.

The following reference signals (5.44) were proposed in [117], [118] and [119] for testing of control methods applied to system (5.43).

$$\begin{aligned} v_1(t) &= 2\sin(0.5t + 0.5) \\ v_2(t) &= \sin(t) \end{aligned} \tag{5.44}$$

The same references are also used is this experiment and the result of closed loop system (see figure 5.4) simulation is depicted in the next figure.

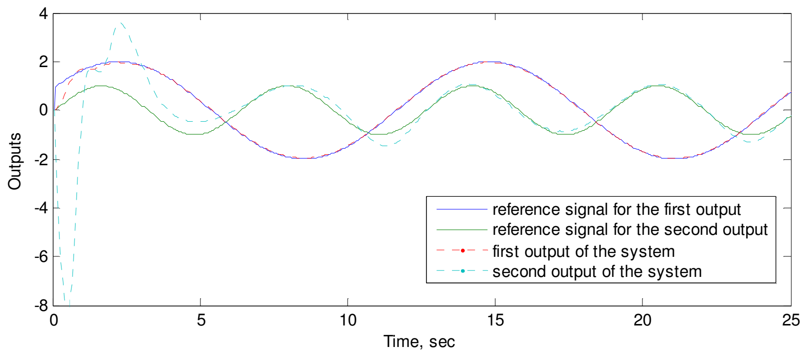

figure 5.10 Control of nonlinear MIMO continuous time system (5.43): output trajectories vs. reference signals

It can be seen from the figure that the control system is capable of tracking both reference signals $v_1(t)$ and $v_2(t)$ when the proposed control algorithm is applied.

Neural Network based Simplified ANARX structure (5.7) introduced here can help to overcome some limitations of ANARX structure based control. Namely, calculation of control signals by dynamic output feedback algorithm becomes much simpler and faster. It does not require any numerical calculation, which takes several iterations to converge and comes to solving a linear equation or system of linear equations (5.12) instead of numerical calculation of inverse function of nonlinear function (5.3). The condition for applicability of the considered control technique is nonlsingularity of matrix T_2 defined in (5.11) and computed from parameters W_1, C_1 of the first sub-layer of NN-SANARX model , which can be very easily checked after training the model.

By using NN-based Simplified ANARX model (5.21), dynamic output feedback linearization algorithm (2.36)-(2.38) was successfully applied to control of nonlinear MIMO systems.

The effectiveness of the proposed control technique was demonstrated on numerical examples 5.1-5.4.

The main drawback of the proposed NN-SANARX model based control technique is restriction imposed on the model by linearity of the first sub-layer of the corresponding neural network representing ANARX structure. In spite of the fact that this control method can be successfully applied to a wide class of nonlinear systems as demonstrated by numerical examples, restrictions imposed by simplification of ANARX model can lead to some constriction of the class of nonlinear system to which this control algorithm can be used.

An alternative approach for simplification of practical application of ANARX model based dynamic output feedback linearization to control of nonlinear systems was proposed by the author in [57]. The proposed technique also makes possible to apply ANARX model linearization algorithm to control of nonlinear MIMO systems without imposing additional restrictions on structure and/or functionality of NN-based ANARX model. It is based on introducing an additional static neural network into the control system for calculation of approximated control signals $u(t)$ from parameters of ANARX model and dynamics of the controller. This approach will be discussed in the next section.

5.5 Additional neural network based approach for practical application of ANARX model based Dynamic Output Feedback Linearization algorithm to control of nonlinear systems

In the previous two sections control technique based on dynamic output feedback linearization of Neural Network based Simplified ANARX model of the controlled system was considered. In this case exact values of control signals can be very easily analytically calculated by solving system of linear equations (5.12) if the corresponding condition ($rank(T_2) \geq m$) is satisfied, but this approach imposes additional restrictions on NN-ANARX model (linearity of the first sub-layer). This approach is based on dynamic output feedback linearization of ANARX model (5.1) without any additional restrictions. Additional static neural network has to be trained for calculation of control signals. Approximated values of vector of control signals $u(t)$ is used, but this technique can be applied to a wider class of nonlinear systems. It can be used for control of all nonlinear systems identified by ANARX models.

In order to use the dynamic output feedback linearization algorithm for control of nonlinear systems, function (2.37) has to be calculated to produce control signals. It means that inverse function of (2.38) has to be found. Function $f_1\left(y(t), u(t)\right)$ is the first element of sum (2.13) representing ANARX model of a nonlinear system. When ANARX model is represented in the form of neural network (3.18) shown in figure 3.9, function (2.38) takes the form (4.8) and is corresponding to the function of the first sub-layer of NN-ANARX model (3.18).

Because of well known neural networks approximation capabilities, a neural network can be trained to approximate function (2.37) or (4.10) and to find an inverse of function (2.38) or (4.8). These functions $\eta_1(t) = f_1\left(y(t), u(t)\right)$ and $\eta_1(t) = C_1\varphi_1\left(W_1\left[y(t), u(t)\right]^T\right)$ are static functions of arguments $u(t)$ and $y(t)$ producing an output value $\eta_1(t)$. When ANARX model is obtained, this function can be considered as a separate system and simulated with random inputs to produce a set of input-output data which can be used as a training set for approximation of function

$$u(t) = \psi(y(t), \eta_1(t)), \tag{5.45}$$

where $\psi(\cdot)$ is a nonlinear map performed by a feedforward neural network. According to Stone-Weirstrass theorem [44], [17], a two-layer perceptron with nonlinear sigmoid activation functions of its hidden layer neurons is capable of approximating any arbitrary continuous map to within a desired accuracy. It means that function (5.45) can be obtained by training a two-layer perceptron

$$u(t) = C_0\varphi_0\left(W_0\left[y(t), \eta_1(t)\right]^T\right), \tag{5.46}$$

where W_0 is the matrix of synaptic weights between input vector and the hidden layer neurons, C_0 is the vector of output layer synaptic weights and $\varphi_0(\cdot)$ is a nonlinear sigmoid-type activation function of the hidden layer neurons.

It was demonstrated in [13] and [14] that for a wide class of nonlinear systems, Neural Network based ANARX model is capable of representing original model with high degree of accuracy. Here ANARX models of nonlinear systems represented by a neural network of the corresponding structure will be considered. When NN-ANARX model is used to identify a nonlinear system, the first sub-layer

$$\eta_1(t) = C_1\varphi_1\left(W_1 \cdot [y(t), u(t)]^T\right) \tag{5.47}$$

can be considered as a system for generating a data set for training neural network (5.46).

NN-ANARX model based dynamic output feedback linearization control algorithm can now be represented as follows.

$$u(t) = C_0\varphi_0(W_0[y(t), \eta_1(t)]^T) \tag{5.48}$$

$$\begin{aligned} \eta_1(t+1) &= \eta_2(t) - C_2\varphi_2\left(W_2 \cdot [y(t), u(t)]^T\right) \\ &\vdots \\ \eta_{n-2}(t+1) &= \eta_{n-1}(t) - C_{n-1}\varphi_{n-1}\left(W_{n-1} \cdot [y(t), u(t)]^T\right) \\ \eta_{n-1}(t+1) &= v(t) - C_n\varphi_n\left(W_n \cdot [y(t), u(t)]^T\right) \end{aligned} \tag{5.49}$$

and the structure of the corresponding control system is presented in figure 5.11.

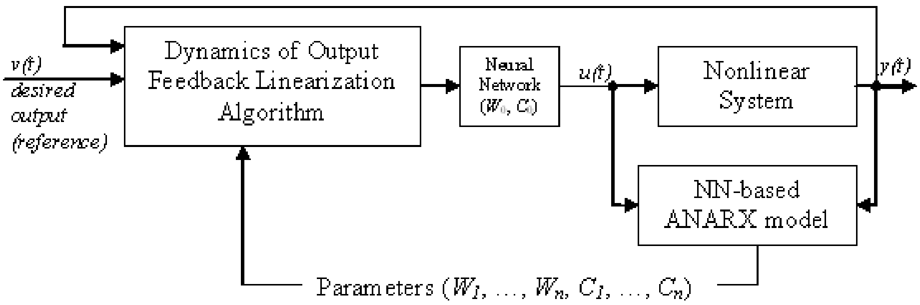

figure 5.11 NN-ANARX model based control with additional neural network

For convenience of comparing with the previously discussed technique, this control system was simulate on the same test systems. Consider the following example of a nonlinear SISO system control by dynamic output feedback linearization of NN-ANARX model with additional neural network.

5.5.1 Numerical example 5.5

The model of a liquid level system of interconnected tanks [72] is represented by the input-output equation (3.21). Identification of this system by NN-based ANARX model was demonstrated in [14] and discussed in detail in section 3.5.2 (numerical example 3.4) of this thesis. Third order NN-ANARX model (3.22) with parameters (3.23) was obtained. Identified parameters of the second and the third sub-layer of the network W_2, W_3, C_2, C_3 were used for constructing the dynamics of the second order controller according to (5.49) as

$$\begin{aligned} \eta_1(t+1) &= \eta_2(t) - C_2\varphi_2\left(W_2 \cdot [y(t), u(t)]^T\right) \\ \eta_2(t+1) &= v(t) - C_3\varphi_3\left(W_3 \cdot [y(t), u(t)]^T\right) \end{aligned} \qquad (5.50)$$

and by using parameters of the first sub-layer W_1, C_1, it was simulated as a separate system (5.47) to generate training data set for approximation of function (2.37) by (5.46). It was simulated with random *u*(*t*) and *y*(*t*) values. The sequence of corresponding values $\eta_1(t)$ was calculated. Vectors $\{y(t), \eta_1(t)\}$ were used as inputs and vector *u*(*t*) was used as the etalon values for training of network (5.46). Parameters W_0 and C_0 of this additional network were calculated by the LM training algorithm on the training set consisting of 1000 data patterns. logistic sigmoid activation function (2.50) was used as activation function $\varphi_0(\cdot)$ of neurons of the hidden layer of network (5.46). This network was validated on the different data set consisting of 500 random input and corresponding output values. The mean square error on the validation data set was as low as about $5.7 \cdot 10^{-5}$. This network was used with dynamic output feedback linearization (5.50) for calculation of control signals according to equation (5.46).

Algorithm (5.46), (5.50) was applied to control of nonlinear system (3.21). Figure 5.12 shows that the output of the system closely follows the reference signal.

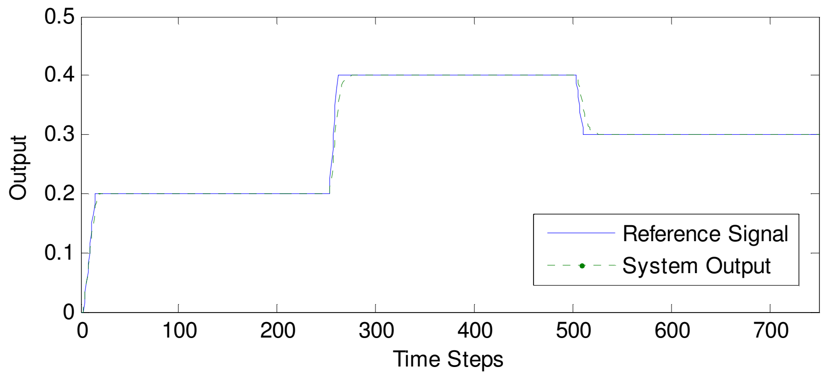

figure 5.12 Control of the liquid level system of interconnected tanks

The corresponding control signal calculated by (5.46) is depicted in figure 5.13.

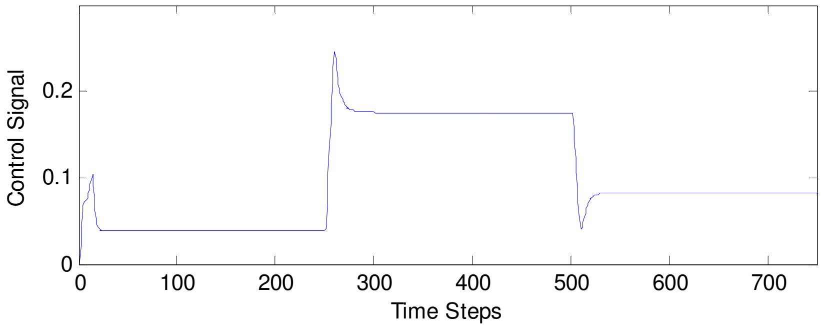

figure 5.12 Control signal for the liquid level system of interconnected tanks

The result can be compared to the technique based on calculation of control signal by the Newton's method proposed in [14] and discussed in section 4.4.2 (numerical example 4.5) of this thesis and also to the control based on NN-SANARX model presented in [56] and discussed in detail in sections 5.3.1-5.3.2 (numerical example 5.1) of this thesis. By comparing these approaches it can be concluded that the proposed control technique has better performance. Control of liquid level system of interconnected tanks (3.21) by the method based on additional neural network has no overshoots (see figure 5.11) and smaller regulation time.

There is no need to compute an inverse model of a dynamic system only inverse of static function (5.47) has to be approximated by training a static neural network (two-layer perceptron). After additional network is trained, control signals can be calculated very fast by function (5.48). No numerical computation requiring several iterations to converge is needed.

Additional network (5.48) can also be trained for MIMO ANARX model (5.11) shown in figure 3.15 and thus the considered control technique can also be applied to control of nonlinear MIMO systems as will be demonstrated in the next section.

5.6 NN-ANARX structure based control of nonlinear MIMO systems

In case of MIMO systems $u(t)$ and $y(t)$ are vectors of system's inputs and outputs. $u(t) \in \Re^{r \times 1}$ and $y(t) \in \Re^{m \times 1}$ as defined in (5.22) and (5.23). Neural network representing ANARX structure is depicted in figure 3.15 (see section 3.7.2) and can be formalized by difference equation (5.1). System of first order difference equations representing the dynamics for output linearization remains the same as in case of linearization of Simplified ANARX structure and is defined by (5.9). Because of nonlinearity of the first sub-layer, (5.2) has to be solved. It can be done by approximation of (5.3) by training an additional static neural network representing an inverse of (5.2).

First sub-layer of the network (5.1) representing ANARX structure for MIMO systems

$$\left[\eta_{1,1}(t),\ldots,\eta_{1,m}(t)\right]^T = C_1\varphi_1\left(W_1\cdot\left[y_1(t),\ldots,y_m(t),u_1(t),\ldots,u_r(t)\right]^T\right) \tag{5.51}$$

can be simulated as a separate system to generate training input-output data set for training neural network performing map (5.52)

$$\left[u_1(t),\ldots u_r(t)\right]^T = C_0\varphi_0(W_0\left[y_m(t),\ldots y_m(t),\eta_{11}(t),\ldots,\eta_{1m}(t)\right]^T) \, . \tag{5.52}$$

Here $W_0 \in \Re^{l\times 2m}$ and $C_0 \in \Re^{r\times l}$ are matrices of synaptic weights of the hidden and the output layers, $\varphi_0(\cdot)$ is an activation function of the neurons of the hidden layer and l is the number of hidden layer neurons. This network can be used for calculation of control signals by dynamic output feedback linearization algorithm (5.8)-(5.9) for control of nonlinear MIMO systems.

Consider the following examples demonstrating the effectiveness of the proposed approach. This control method was also simulated on the same nonlinear MIMO systems as the previous one for convenience of comparison between them.

5.6.1 Numerical example 5.6

A nonlinear MIMO discrete-time system [113], [114], [56], [57] with two inputs and two outputs represented by input-output equations (5.28) was simulated and the obtained set of input-output data was used for training of MIMO NN-based ANARX structure (5.1) depicted in figure 3.15 with two sub-layers corresponding to the second order of model (5.53).

$$\left[y_1(t), y_2(t)\right]^T = \sum_{i=1}^{2} C_i\cdot\varphi_i\left(W_i\cdot\left[y_1(t-i), y_2(t-i), u_1(t-i), u_2(t-i)\right]^T\right). \tag{5.53}$$

The first layer consisted of 2 neurons ($l_1 = 2$) and the second layer consisted of 5 neurons ($l_2 = 5$). Both sub-layers used hyperbolic tangent sigmoid (2.51) activation functions of their neurons. Identified parameters of trained model (5.53) have the following values:

$$W_1 = \begin{bmatrix} 0.1017 & -0.1164 & 0.0702 & 0.0900 \\ -0.1019 & 0.1150 & -0.0717 & -0.0963 \end{bmatrix},$$

$$W_2 = \begin{bmatrix} 0.0001 & 0.0007 & -0.0001 & -0.2396 \\ -0.0003 & 0.0105 & -0.0039 & -1.5285 \\ -0.2890 & 0.1291 & -0.6001 & -0.3454 \\ -0.0004 & 0.0001 & -0.0002 & 0.0936 \\ -0.1884 & 0.1765 & -0.2033 & 0.0062 \end{bmatrix}, \tag{5.54}$$

$$C_1 = \begin{bmatrix} 112.6799 & 100.0755 \\ -168.1208 & -167.6664 \end{bmatrix},$$

$$C_2 = \begin{bmatrix} 196.0781 & -1.4724 & -0.0253 & 488.6156 & 0.9161 \\ 99.8741 & -0.7769 & 0.2107 & 242.0401 & -0.7634 \end{bmatrix}.$$

Model (5.53) with identified parameters (5.54) was validated on sinusoidal input signals (5.32)-(5.33). The result of the model validation is depicted in the next figure.

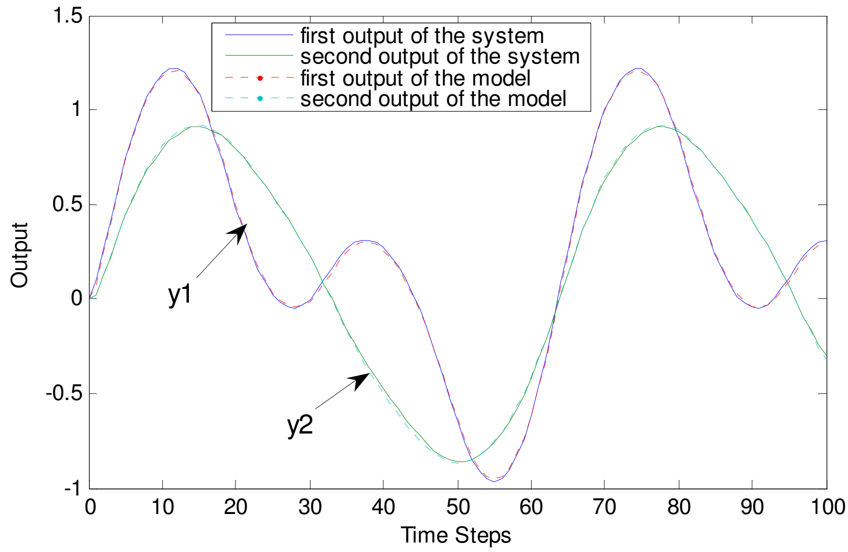

figure 5.13 Identification of nonlinear MIMO system (5.28) by second order NN-based ANARX model (5.53)-(5.54): model validation

It can be seen from the figure that this second order MIMO NN-based ANARX model represent nonlinear MIMO system (5.28) with high accuracy. Both outputs of the model closely repeats the outputs of the system. Mean Square Error on this validation set is about $1.32 \cdot 10^{-4}$, that is about the same good as in case of identifying this system by NN-based Simplified ANARX structure shown in numerical example 5.2 (see section 5.4.2).

After training the model, its first sub-layer was simulated as a separate system with four inputs $y_1(t), y_2(t), u_1(t), u_2(t)$ and two outputs η_{11}, η_{12} represented by function (5.55).

$$[\eta_{11}(t), \eta_{12}(t)]^T = C_1\varphi_1\Big(W_1 \cdot [y_1(t), y_2(t), u_1(t), u_2(t)]^T\Big). \tag{5.55}$$

Static function (5.55) was simulated with random inputs and the obtained data set was used to train a two-layer perceptron approximating an inverse of this function as

$$[u_1(t), u_2(t)]^T = C_0\varphi_0\big(W_0 \cdot [y_1(t), y_2(t), \eta_{11}(t), \eta_{12}(t)]^T\big), \tag{5.56}$$

where $\varphi_0(\cdot)$ is a nonlinear sigmoid activation function (2.51) of the hidden layer neurons. Experiments have shown that 4 neurons in the hidden layer is enough to approximate function (5.3) with accuracy satisfying the requirements of the control system. Identified parameters of additional neural network (5.56) are as follows:

$$W_0 = \begin{bmatrix} 0.0049 & -0.0082 & 0.0424 & 0.0713 \\ 0.0125 & -0.0197 & -0.0064 & 0.0473 \\ 0.0207 & -0.0271 & -0.0430 & 0.0181 \\ -0.0145 & 0.0201 & -0.0604 & -0.0137 \end{bmatrix},$$
$$C_0 = \begin{bmatrix} 55.3913 & -75.3111 & -32.4722 & 43.8034 \\ -57.9163 & 128.1436 & -51.7998 & -11.8438 \end{bmatrix} \tag{5.57}$$

After second order ANARX model (5.53) of system (5.53) is identified, first order dynamics of the controller can be obtained from the dynamic output feedback linearization algorithm (5.9) by using model's parameters (5.52) as

$$[\eta_{11}(t+1), \eta_{12}(t+1)]^T = [v_1(t), v_2(t)]^T - C_2\varphi_2\big(W_2 \cdot [y_1(t), y_2(t), u_1(t), u_2(t)]^T\big) \tag{5.58}$$

and the vector of control signals can be very easily and fast computed by static neural network (5.56). It follows from these equations that the controller for this nonlinear MIMO system can be represented by the following first order difference equation.

$$[u_1(t), u_2(t)]^T = C_0\varphi_0\Big(W_0 \cdot \Big[y_1(t), y_2(t), \big[v_1(t-1), v_2(t-1)\big] - \\ -\Big(C_2\varphi_2\Big(W_2 \cdot \big[y_1(t-1), y_2(t-1), u_1(t-1), u_2(t-1)\big]^T\Big)\Big)^T\Big]^T\Big) \quad (5.59)$$

System (5.59) with identified parameters W_0, W_2, C_0 and C_2 from (5.54), (5.57) was used to control of nonlinear MIMO system (5.28). The control system corresponding to the closed loop system depicted in figure 5.11 was simulated with sinusoidal reference signals $v_1(t)$ and $v_2(t)$ - see figure 5.14.

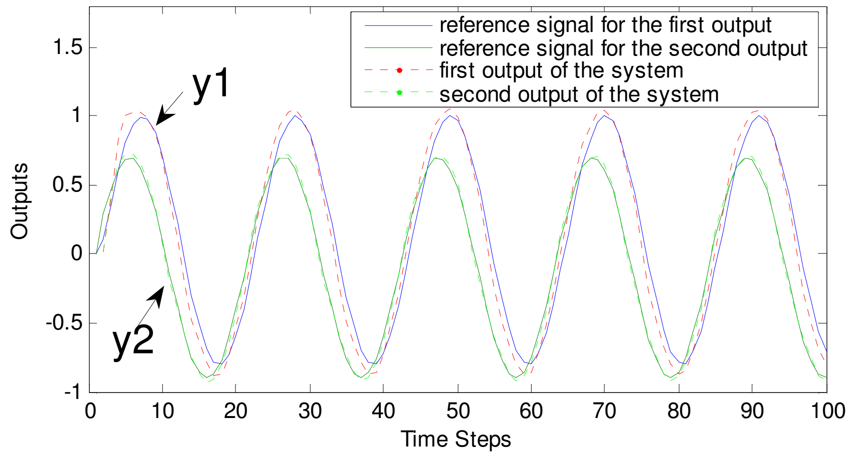

figure 5.14 MIMO NN-ANARX model based control: closed loop simulation with sinusoidal reference signals

The corresponding control signals $u_1(t)$ and $u_2(t)$ calculated by (5.59) are depicted in figure 5.15.

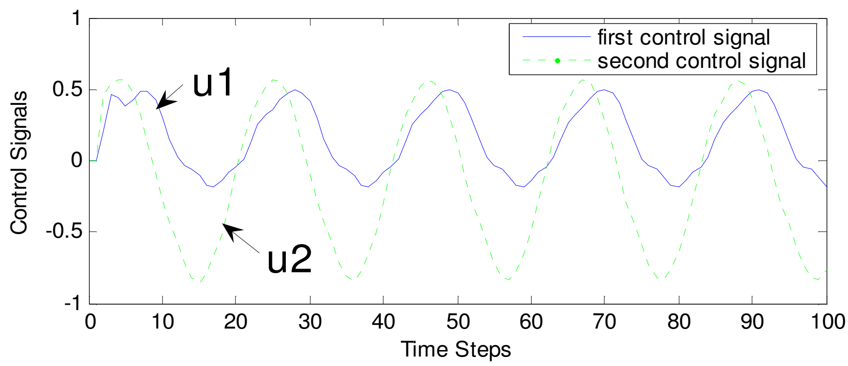

figure 5.15 MIMO NN-ANARX model based control: control signals

Control system based on dynamic output feedback linearization of ANARX model was also simulated with piece-constant reference signals. The results of this simulation are depicted in figure 5.16.

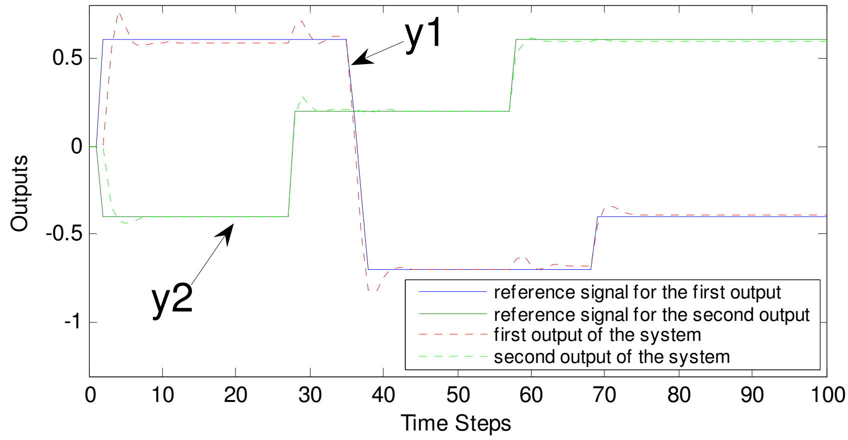

figure 5.16 MIMO NN-ANARX model based control: closed loop simulation with piece-constant reference signals

It can be seen from figures 5.14 and 5.16 that outputs of the control system $y_1(t)$ and $y_2(t)$ are capable of simultaneous tracking the desired reference signals $v_1(t)$ and $v_2(t)$ correspondingly when the proposed control technique is applied.

It can also be concluded by comparing these results with the results obtained by using NN-based Simplified ANARX model discussed in section 5.4.1 (numerical example 5.2), that both proposed technique lead to about the same good accuracy of tracking the reference signals by dynamic output feedback linearization algorithm.

5.6.2 Numerical example 5.7

Nonlinear discrete-time system [115], [56] represented by input-output equations (5.39) was also used to evaluate the effectiveness of the proposed control algorithm. This system was identified by MIMO NN-ANARX structure with two sub-layers (5.53) corresponding to the second order of the model.

Unlike in numerical example 5.3 (see section 5.4.2), when NN-based MIMO NN-ANARX structure with all nonlinear layers is used, this system can be successfully identified by the model of th second order. Smaller order of the model in this case decreases the order of the controller making control system simpler and faster without reducing the quality of the control.

After parameters of the model W_1, W_2, C_1 and C_2 were identified, the first sub-layer (5.55) was simulated with random inputs to produce a training data set for training neural network (5.56) approximating function 5.3.

Additional neural network (5.56) was trained and parameters W_0, C_0 of the inverse function of (5.55) were identified. After that, ANARX mode based dynamic output feedback linearization algorithm (4.8)-(4.9) can be applied to control of nonlinear MIMO system (5.39). The dynamic system (5.59) represents the controller for this nonlinear system. The control system was simulated with sinusoidal reference signals $v_1(t)$ and $v_2(t)$ as in example 5.3. The result of the control system simulation is presented in figure 5.17.

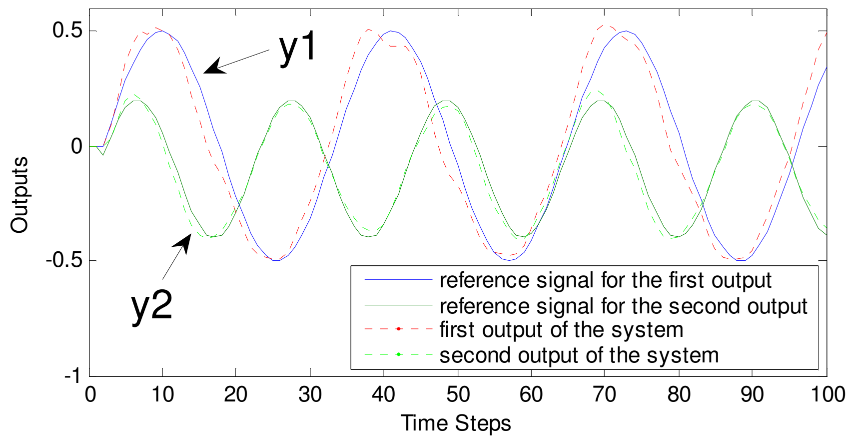

figure 5.17 Control system simulation: output trajectories vs. reference signals

It can be seen from the figure that by using additional neural network, dynamic output feedback linearization algorithm based on MIMO NN-ANARX model can be applied to control of nonlinear MIMO system (5.39). The control system is capable of simultaneous tracking of both reference signals. By comparing this result with dynamic output feedback linearization based on Simplified ANARX model presented in figure 5.9 (see section 5.4.2) it can be mention that by using NN-ANARX model with all nonlinear sub-layers, the order of the control system can be reduced without significant loss in quality of control. Second order MIMO NN-ANARX model can be used instead of third order MIMO NN-SANARX model for identification of second order nonlinear discrete time system (5.39).

The control technique based on dynamic output feedback linearization of ANARX model with additional neural network for calculation of control signals was also applied to control of a nonlinear MIMO continuous time system. Consider the following example.

5.6.3 Numerical example 5.8

Nonlinear continuous time MIMO system (5.43), which was used as a test system for control algorithms in [116], [117], [118], [119], [56], [57] was identified by discrete time second order MIMO NN-ANARX model (5.53) with sample time 0.05sec proposed in [119].

Control technique (5.59) based on calculation of inverse function (5.52) of the first sub-layer (5.51) of the network representing ANARX model (5.53) of this system was applied to control of this system. Closed loop system corresponding to the scheme shown in figure 5.11 was simulated with reference signals (5.44) were proposed in [117], [118] and [119]. These reference signals were also used in example 5.4 for evaluation of control system based on NN-SANARX model. The result of this is presented in the following figure.

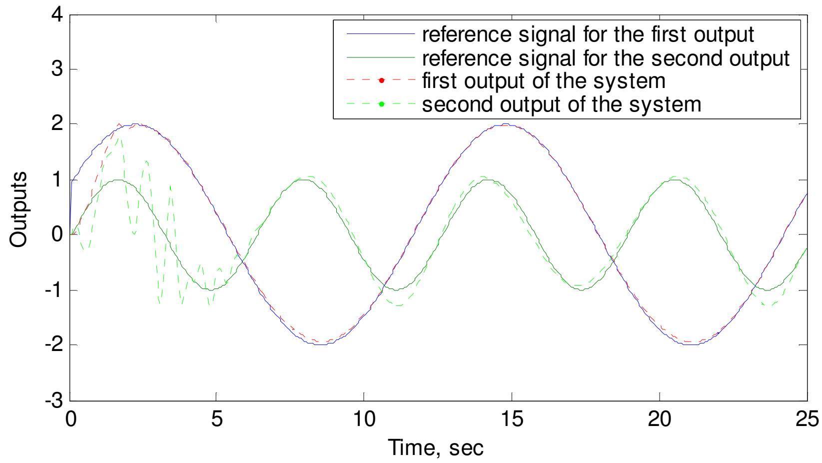

figure 5.18 Control of nonlinear continuous time MIMO system (5.43): output trajectories vs. reference signals

It can be seen from the figure that by using the proposed approach the system is capable of tracking the desired trajectories. By comparing this result with the results discussed in section 5.4.3 (numerical example 5.4) and shown in figure 5.10 it can be seen that using Simplified ANARX structure and analytical approach to calculation of controls from the dynamics of the controller results in nonoscillating output trajectories with higher overshooting when in case of nonanalytical approach with fully nonlinear NN-based ANARX model.

It can be seen from numerical examples 5.2-5.4 and 5.6-5.8 that dynamic output feedback linearization algorithm (4.8)-(4.9) can be successfully applied to control of a wide class of nonlinear MIMO systems. The problem of practical calculation of a vector of control signals from the dynamics of the output linearization based controller is solved by introducing two approaches. By introducing NN-based Simplified ANARX structure, the vector of controls can be analytically very easily computed by solving system of linear equations (5.12). When fully nonlinear NN-based ANARX structure has to be used to model a nonlinear MIMO system, nonanalytical additional neural network based approach is proposed for calculation of control signals. Results of simulations show that both techniques result in about the same good quality of control. It was also shown by simulations that these approaches give several advantages in control of nonlinear SISO systems. From the view point of these algorithms, SISO systems can be considered as a sub-class of MIMO systems.

Because of neural networks' well known ability to learn and to change their behavior in response to changes in environmental conditions, dynamics of the process and disturbances, an adaptive controller can be easily designed when an artificial neural network is used to model the process under control. Adaptive output feedback control based on NN-ANARX models with history-stack adaptation for nonlinear SISO systems was presented in [14] and discussed in detail in section 4.4 of this thesis. Experimental results shown in numerical examples 4.5 and 4.6 demonstrate the effectiveness of this approach.

Both of the techniques proposed in this chapter and applied to control of nonlinear MIMO systems by output feedback impose some restriction to the adaptation technique. That is why adaptive control of nonlinear MIMO systems by output linearization of NN-ANAARX and NN-SANARX models needs to be considered separately in the next section.

5.7 NN-ANARX and NN-SANARX model based adaptive control of nonlinear MIMO systems

Adaptive control of nonlinear SISO systems by dynamic output feedback linearization (4.8)-(4.9) with history-stack adaptation of NN-based ANARX model was demonstrated in [14] and discussed in detail in section 4.4 of this thesis. The same control algorithm was also applied to control of nonlinear MIMO systems [56], [57] by using two methods shown above in this chapter.

The first approach [56] (see sections 5.3-5.4) is based on introducing a neural network based Simplified ANARX structure (5.21) having linear first sub-layer. Parameters W_1 and C_1 of this sub-layer are used for analytical calculation of vector of control signals (5.22) linearizing NN-SANARX model by solving system of linear equations (5.26). The criteria of applicability of this technique is nonsingularity of matrix T_2 calculated. It can be seen from (5.10) and (5.11) that T_2 depends on parameters of the first sub-layer of the corresponding neural network representing SANARX structure. This criteria can be very easily checked after identifying the model by training the network.

When neural network is adjusted on-line by an adaptation algorithm, it can not be somehow guaranteed that matrix T_2 will not become singular or close to singular at any time instance. Thus, this sub-layer with synaptic weights W_1 and C_1 should not be changed by the adaptation algorithm.

The second approach [57] (see sections 5.5-5.6) is based on introducing an additional static neural network (5.52) into the control system (see figure 5.11) approximating function (5.3). This nonlinear function (5.52) represents an inverse of function (5.51) corresponding to the first sub-layer of neural network (5.1) representing fully nonlinear ANARX models of MIMO systems. This additional neural network has to be trained before running the control system. To train this network, the first sub-layer of NN-based ANARX model (5.1) has to be simulated with random inputs. This gathered training data set is used to identify parameters W_0 and C_0 of additional static network (5.52).

It is obvious that if an adaptation algorithm adjusting parameters of the model is applied, changing synaptic weights W_1 and C_1 of the first sub-layer of MIMO NN-ANARX model leads to necessity of adjusting parameters W_0 and C_0 of additional network as well by simulating the first sub-layer, gathering additional training set and adjusting parameters of this network after adjusting the model. Evidently, these actions can not be practically performed on-line when fast response from the controller is needed. It means that the first sub-layer of MIMO NN-ANARX model (5.1) should not be included in the adaptation procedure.

In case of both methods making possible practical application of NN-ANARX model based output feedback control technique to control of nonlinear MIMO systems, first sub-layer of the corresponding network representing the model in a suitable form, has to be excluded from the network during the adaptation procedure. The corresponding changes have been introduced into the model adaptation algorithm as will be shown below.

From the viewpoint of the control algorithm, both MIMO NN-ANARX (5.1) and MIMO NN-SANARX (5.21) models can be represented as follows

$$\begin{aligned}[y_1(t),\ldots,y_m(t)]^T &= [\eta_{1,1}(t-1),\ldots,\eta_{1,m}(t-1)]^T + \\ &+ \sum_{i=2}^{n} C_i \cdot \varphi_i\left(W_i \cdot [y_1(t-i),\ldots,y_m(t-i),u_1(t-i),\ldots,u_r(t-i)]^T\right),\end{aligned} \tag{5.60}$$

where from one side (from the model's viewpoint), vector $[\eta_{1,1}(t),\ldots,\eta_{1,m}(t)]^T$ is the vector outputs of the first sub-network.

Here a sub-network is a sub-layer multiplied by the corresponding output matrix of synaptic weights of neural network having structure depicted in figure 3.9 or 3.15.

For MIMO NN-SANARX models

$$[\eta_{1,1}(t),\ldots,\eta_{1,m}(t)]^T = C_1 \cdot W_1 \cdot [y_1(t-1),\ldots,y_m(t-1),u_1(t-1),\ldots,u_r(t-1)]^T \tag{5.61}$$

and for MIMO NN-ANARX models

$$\left[\eta_{1,1}(t),\ldots,\eta_{1,m}(t)\right]^T = C_1 \cdot \varphi_1\left(W_1 \cdot \left[y_1(t-1),\ldots,y_m(t-1),u_1(t-1),\ldots,u_r(t-1)\right]^T\right). \tag{5.62}$$

From another side, vector $\left[\eta_{1,1}(t),\ldots,\eta_{1,m}(t)\right]^T$ comes at each time step from the dynamics (5.9), (5.48) of the controller. Parameters of the next n-1 sub-networks are used to derive this dynamics and weighted sum of outputs of these sub-layers produce an adaptive neural network – adaptive part of the model. It can be represented by the following equation.

$$\left[y_1^a(t),\ldots,y_m^a(t)\right]^T = \sum_{i=2}^{n} C_i \cdot \varphi_i\left(W_i \cdot \left[y_1(t-i),\ldots,y_m(t-i),u_1(t-i),\ldots,u_r(t-i)\right]^T\right), \tag{5.63}$$

where, $\left[y_1^a(t),\ldots,y_m^a(t)\right]^T$ is the vector of outputs of neural network representing adaptive part of model (5.60). From the other side,

$$\left[y_1^a(t),\ldots,y_m^a(t)\right] = \left[y_1(t),\ldots,y_m(t)\right] - \left[\eta_{1,1}(t-1),\ldots,\eta_{1,m}(t-1)\right]. \tag{5.64}$$

Equation (5.64) can be used to produce a training data set for adaptation of the model when vector $\left[y_1(t),\ldots,y_m(t)\right]$ is obtained from the output of the controlled system and vector $\left[\eta_{1,1}(t),\ldots,\eta_{1,m}(t)\right]$ is computed by the dynamic feedback controller.

History-stack adaptation technique discussed in section 4.4.1 was used for adaptation of (5.63). Set of vectors $\left\{\left[y(t-i),u(t-i)\right]^T, y^a(t-i), i=1,\ldots,n_p\right\}$ is used as the training set for model adaptation by on-line training of (5.63). Here

$$\left[y(t),u(t)\right] = \left[y_1(t),\ldots,y_m(t),u_1(t),\ldots,u_r(t)\right], \tag{5.65}$$

vector $y^a(t)$ is defined by (5.64) and n_p is the number of patterns in the stack (size of the history-stack).

The gradient descent training algorithm (see section 2.6.1 of this thesis) was chosen to perform on-line adaptation because of shorter iterations and consequently faster calculation of changes $\Delta W_2,\ldots\Delta W_n,\Delta C_2,\ldots\Delta C_n$ of parameters of model (5.60) at each time step.

The structures of control schemes for adaptive control of nonlinear MIMO systems based on dynamic output feedback linearization of NN-ANARX and NN-SANARX models with history-stack adaptation are presented in figures 5.19 and 5.20. SISO systems can be considered as a special case of MIMO systems and the same approach to control system adaptation should be implemented when the system is being controlled by dynamic output feedback based on NN-SANARX model or NN-ANARX model with additional neural network.

When NN-based Simplified model (5.21) is used for identification of a nonlinear MIMO system under control, the corresponding adaptive control system has the structure depicted in the following figure.

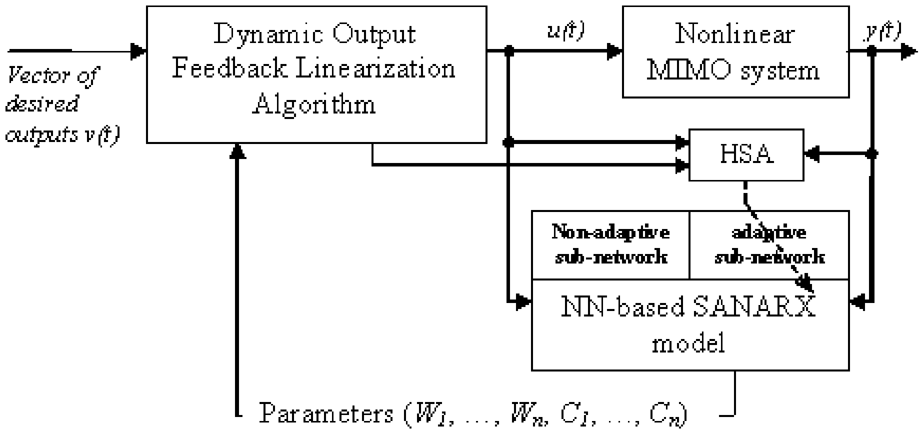

figure 5.19 Adaptive control of nonlinear MIMO systems: Structure of closed loop system with dynamic output feedback linearization of NN-SANARX model and history-stack adaptation

It can be seen from the scheme that the adaptation algorithm (HSA) utilizes current inputs and outputs from the output of the controlled system together with the vector of inner states of the controller. Thus, adaptation is based on the following vectors

$$HSA\begin{cases}[u_1(t),\ldots,u_r(t)]\\ [y_1(t),\ldots,y_m(t)]\\ [\eta_{1,1}(t),\ldots,\eta_{1,m}(t)]\end{cases} \tag{5.66}$$

and adaptation is applied to only a part of the model - to adaptive sub-networks (5.63).

When a nonlinear MIMO system is identified by a fully nonlinear NN-based ANARX structure (5.1), the corresponding adaptive control system can be represented by the structure depicted in figure 5.20.

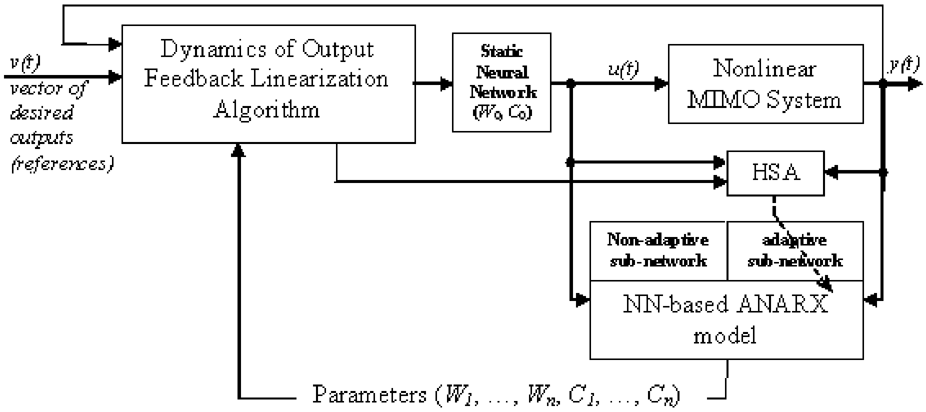

figure 5.20 Adaptive control of nonlinear MIMO systems: Structure of closed loop system with dynamic output feedback linearization of NN-ANARX model, additional neural network and history-stack adaptation

Adaptive control of a nonlinear MIMO system was simulated by using both of these approaches and is considered in the next section. By simulating both MIMO NN-ANARX and MIMO NN-SANARX models based control schemes (see figures 5.19 and 5.20) with input/output disturbances and flowing parameters, the robustness of both control systems is also studied and compared in the following numerical example.

5.7.1 Numerical example 5.9

A nonlinear MIMO discrete-time system [113], [114], [56], [57] represented by the input-output equations (5.28) was identified by NN-based Simplified ANARX model (5.29) with two sub-layers corresponding to the second order of NN-SNARX model as shown in section 5.4.1 (numerical example 5.2). Synaptic weights (5.30) were obtained by training the neural network representing the model. After that, algorithm (5.37)-(5.38) based on dynamic output feedback linearization of this model was successfully applied to control of system (5.28). The structure of the corresponding nonadaptive control system is presented in figure 5.4.

The same system (5.28) was also identified by second order MIMO NN-based ANARX model (5.53). Identification of this system by NN-ANARX model and control based on this model with additional neural network (5.56) was demonstrated in section 5.6.1 of this thesis (numerical example 5.6). Identified parameters (5.54) of the model and synaptic weights (5.57) of the additional neural network were used for control of this system by algorithm (5.59) based on dynamic output feedback linearization of NN-ANARX model according to closed loop control system depicted in figure 5.11.

Both control technique give about the same good result when the systems are simulated without disturbances and with constant parameters of the controlled system represented by equations (5.28). At the same time, the aim of the adaptive controller is to modify its behavior in response to changes in the dynamics of the process and disturbances [42], [43]. That is why both control systems were simulated with input and output disturbances as well as with time-varying parameters of (5.28).

First of all, both NN-SANARX (see figure 5.4) and NN-ANARX (see figure 5.11) models based control systems were simulated with input and output disturbances without adaptation. The following disturbances were added to inputs and outputs of system (5.28).

$$\begin{cases} d_1^i(t) = -0.06 \cdot \mathbf{1}(t-390) \\ d_2^i(t) = 0.15 \cdot \mathbf{1}(t-700) \\ d_1^o(t) = 0.2 \cdot \mathbf{1}(t-550) \\ d_2^o(t) = -0.05 \cdot \mathbf{1}(t-100) \end{cases}, \tag{5.67}$$

where $d_j^i(t)$ is the disturbance level added to the j-th input of the system at time step t; $d_j^o(t)$ is the disturbance added to the j-th output at time step t and $\mathbf{1}(n)$ is the Heaviside step function:

$$\mathbf{1}(n) = \begin{cases} 0, & n < 0 \\ 1, & n \geq 0 \end{cases}. \tag{5.68}$$

The results of simulation of both control systems (5.38) and (5.59) with input and output disturbances (5.67) are depicted in figures 5.21 and 5.22. Control was simulated for $t = 0,1,\ldots,1000$ and adaptation was not applied. It means that parameters (5.30) and (5.54) of the models and consequently parameters of the controllers remained the same during all the simulation time. In this case the influence of disturbances to the outputs of the closed loop control systems can be easily seen from the following figures and used for comparison of the robustness of the proposed approaches.

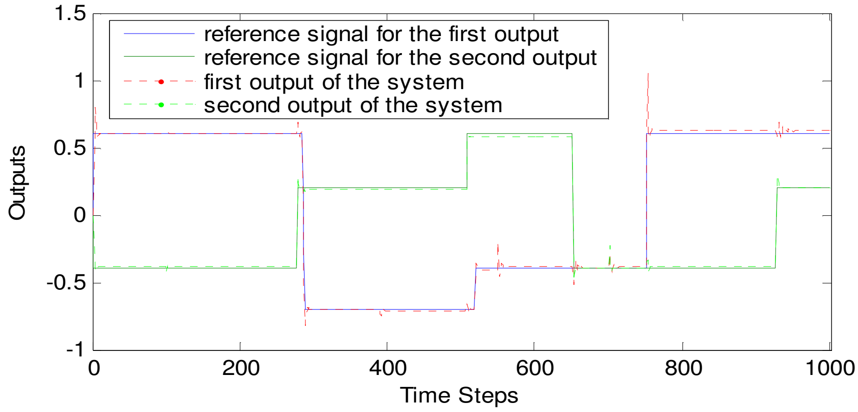

figure 5.21 Simulation of NN-based Simplified ANARX model based nonadaptive control system with input and output disturbances

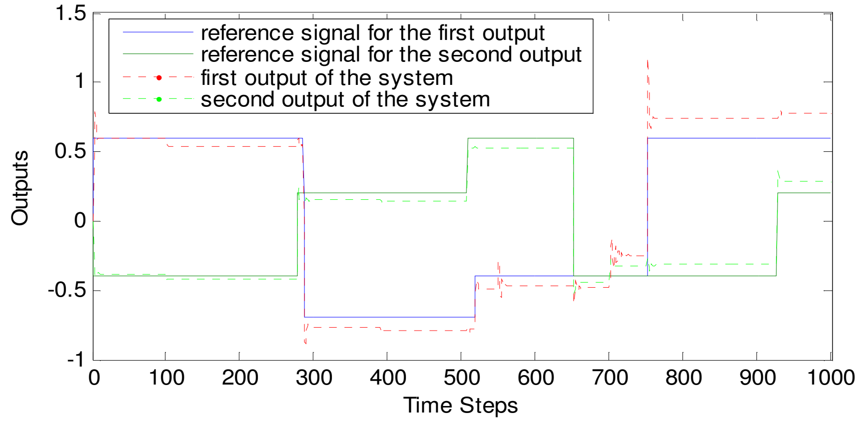

figure 5.22 Simulation of NN-based ANARX model based nonadaptive control system with input and output disturbances

Next, an influence of changes in the dynamics of the controlled process can be studied. In order to do it, four random parameters of equations (5.28) representing the dynamics of the controlled nonlinear MIMO system were chosen to change in time. In this case equations (5.28) take the following form.

$$
\begin{aligned}
y_1(t+1) &= (0.4+\Delta_1(t))y_1(t) + (1+\Delta_2(t))\frac{u_1(t)}{1+u_1^2(t)} + 0.2u_1^3(t) + 0.5u_2(t) \\
y_2(t+1) &= 0.2y_2(t) + (1+\Delta_3(t))\frac{u_2(t)}{1+u_2^2(t)} + (0.4+\Delta_4(t))u_2^3(t) + 0.2u_1(t)
\end{aligned}, \quad (5.69)
$$

where

$$\Delta_1(t)=\begin{cases}-0.15e^{5\cdot10^{-4}\cdot t}, & t<1400\\ 0.1\cos(0.01\cdot t), & t\geq 1400\end{cases}, \tag{5.70}$$

$$\Delta_2(t)=\begin{cases}6.5\cdot10^{-4}\cdot t, & t<550\\ -3.7\cdot10^{-4}\cdot t, & t\geq 550\end{cases}, \tag{5.71}$$

$$\Delta_3(t)=3.3\cdot10^{-4}\cdot t, \tag{5.72}$$

$$\Delta_4(t)=\begin{cases}0.1\cos\left(6.5\cdot10^{-3}\cdot t\right)-0.2, & t<1700\\ 0.15e^{5\cdot10^{-4}\cdot t}, & t\geq 1700\end{cases}. \tag{5.73}$$

For (5.70)-(5.73), $t\in[0;3000]$.

Changes of these parameters are also illustrated by figure 5.23 correspondingly to (5.70)-(5.73).

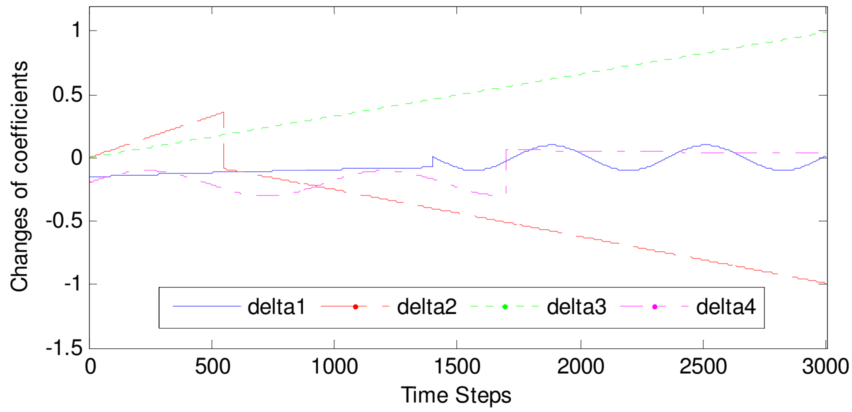

figure 5.23 changes of parameters of the controlled system in time

It can be seen from figure 5.23 and equations (5.69)-(5.73) that parameters of system (5.28) are changing during the control simulation up to 100% of their initial values.

To show the influence of the parameters' variations to outputs of the control systems, the same control systems as in numerical examples 5.2 and 5.6 were simulates with system (5.69) having time-varying parameters (5.70)-(5.70). The results of simulation of these control systems corresponding to two techniques for dynamic output feedback linearization based control of nonlinear MIMO systems proposed in the thesis are depicted in figures 5.24 and 5.25. In this experiment, the control systems were simulated without adaptation.

First, consider the control of system (5.69)-(5.73) by method (5.37)-(5.38) based on MIMO NN-SANARX model (5.29)-(5.30).

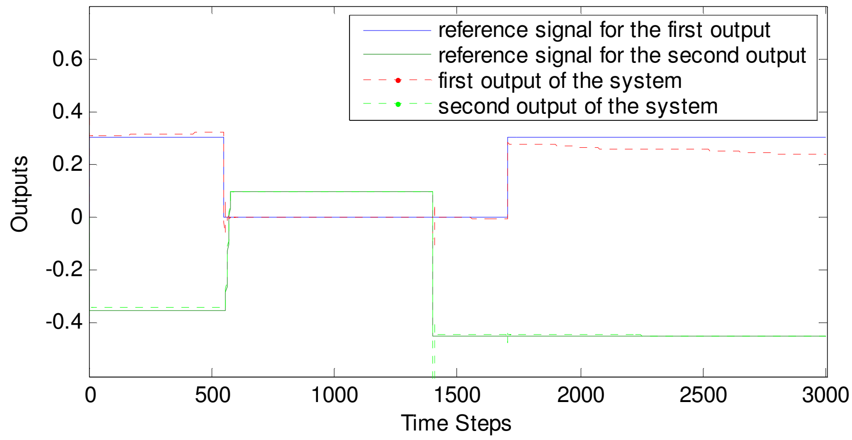

figure 5.24 Simulation of NN-SANARX model based nonadaptive control with flowing parameters of the controlled system

Now, consider the control of the same system by method (5.59) based on MIMO NN-ANARX model (5.53)-(5.54) and additional neural network (5.56)-(5.57).

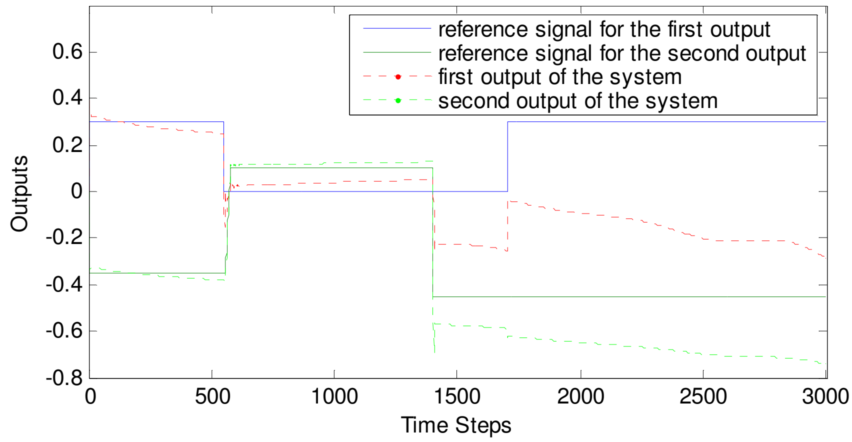

figure 5.25 Simulation of NN-ANARX model and additional neural network based nonadaptive control with flowing parameters of the controlled system

An important conclusion regarding robustness of the proposed algorithms can be made from these figures. It can be easily seen by comparing figure 5.21 with figure 5.22 and figure 5.24 with figure 5.25 that, if no model adaptation is used, control technique based on linearization of NN-SANARX model of a nonlinear MIMO system has much higher robustness than control technique based on linearization of fully nonlinear NN-ANARX model by using additional neural network.

To improve the robustness of these algorithms, especially the second one, an adaptation proposed in section 5.7 was added to both of the control techniques.

In both cases a part of the model (5.63) is adjusted on-line by using history-stack adaptation.

For second order NN-ANARX (5.29) and NN-SANARX (5.53) models of nonlinear system (5.28) having two inputs and two outputs, the adaptive part of the model corresponding to the second sub-layer of the corresponding neural network (the structure of the network is depicted in figure 3.15) is represented by the following equation

$$\left[y_1^a(t), y_2^a(t)\right]^T = C_2 \cdot \varphi_2\left(W_2 \cdot \left[y_1(t-2), y_2(t-2), u_1(t-2), u_2(t-2)\right]^T\right) \tag{5.74}$$

and the vectors of targets for training the model can be calculated according to (5.64) as

$$\left[y_1^t(t), y_2^t(t)\right] = \left[y_1(t) - \eta_{1,1}(t), y_2(t) - \eta_{1,2}(t)\right], \tag{5.75}$$

where for MIMO NN-based Simplified ANARX model

$$\left[\eta_{1,1}(t), \eta_{1,2}(t)\right]^T = C_1 \cdot W_1 \cdot \left[y_1(t-1), y_2(t-1), u_1(t-1), u_2(t-1)\right]^T \tag{5.76}$$

and for MIMO NN-based ANARX models

$$\left[\eta_{1,1}(t), \eta_{1,2}(t)\right]^T = C_1 \cdot \varphi_1\left(W_1 \cdot \left[y_1(t-1), y_2(t-1), u_1(t-1), u_2(t-1)\right]^T\right). \tag{5.77}$$

Thus, for each time step $t \geq n_p$ history-stack is formed as

$$HS = \left\{ \begin{bmatrix} u_1(t) & y_1(t) - \eta_{1,1}(t) \\ u_2(t) & y_2(t) - \eta_{1,2}(t) \end{bmatrix}, \ldots, \begin{bmatrix} u_1(t-n_c) & y_1(t-n_c) - \eta_{1,1}(t-n_c) \\ u_2(t-n_c) & y_2(t-n_c) - \eta_{1,2}(t-n_c) \end{bmatrix} \right\} \tag{5.78}$$

and for each time step $t < n_p$ the size of the stack equals t.

In the adaptive control experiments shown in this example below, the maximal size of the history-stack was chosen as $n_p = 200$.

Adaptive control of nonlinear MIMO system (5.28) with input and output disturbances (5.67) by dynamic output feedback linearization of NN-based Simplified ANARX model according to the structure of the control system depicted in figure 5.19 was simulated. The results of this simulation are shown in the following figure.

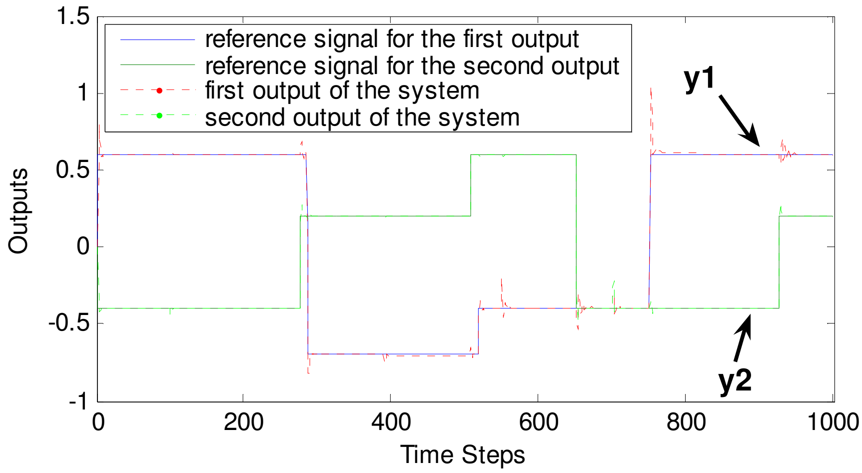

figure 5.26 Simulation of NN-SANARX model based adaptive control of nonlinear MIMO system with input and output disturbances

After that, the same adaptive control technique was also applied to the same system, but with flowing parameters (5.69)-(5.73). See figure 5.27.

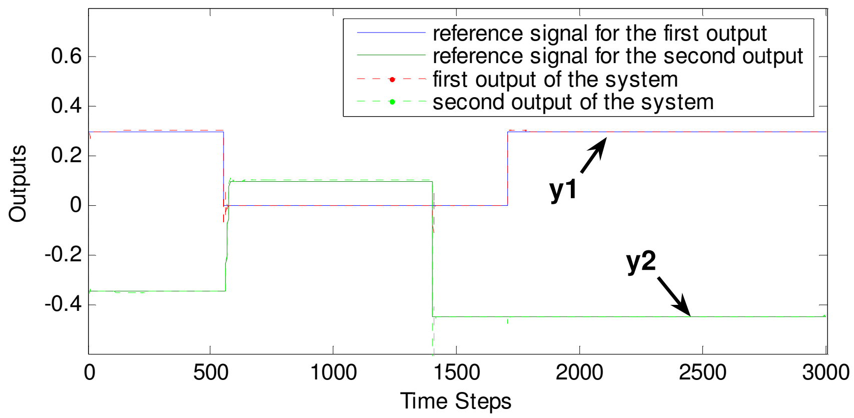

figure 5.27 Simulation of NN- SANARX model based adaptive control of nonlinear MIMO system with flowing parameters

It can be seen from the figures, that this adaptive control system presented in figure 5.19 is capable of tracking the desired reference signals compensating undesirable influence of disturbances and changes in the dynamics of the process. By comparing figure 5.26 with figure 5.21 and figure 5.27 with figure 5.24, a significant improvement in quality of control in case of disturbances and changing parameters can be seen when the proposed adaptation technique is used.

In the next figures the results of simulation of the adaptive control based on NN-ANARX model with additional neural network and history stack adaptation is presented. The corresponding structure of the closed loop control system is depicted in figure 5.20. This control strategy was also applied to nonlinear MIMO system (5.28) with input-output disturbances (5.67) and flowing parameters (5.69)-(5.73). Consider figures 5.28 and 5.29.

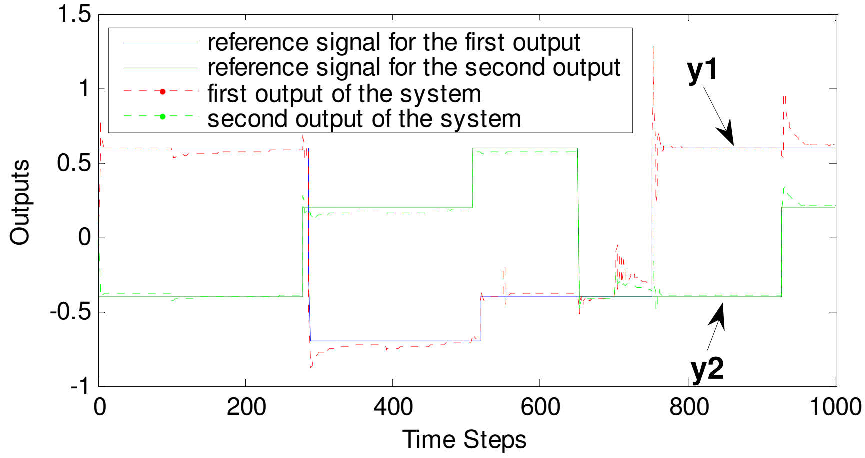

figure 5.28 Simulation of adaptive control of nonlinear MIMO system with input and output disturbances based on NN-ANARX model with additional neural network

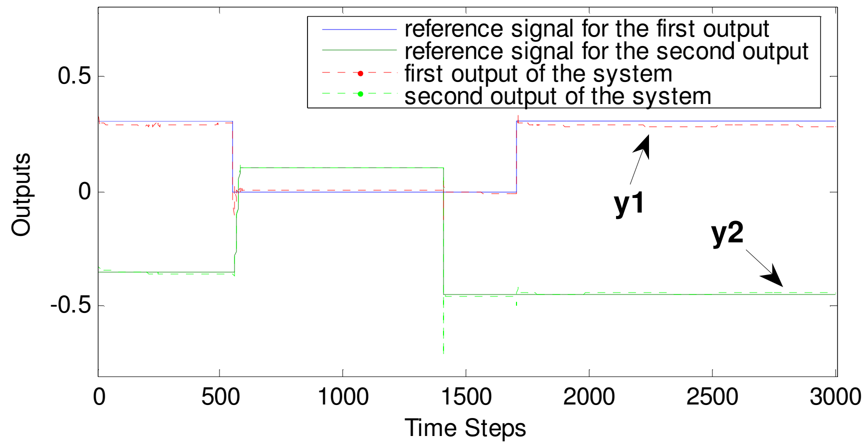

figure 5.29 Simulation of adaptive control of nonlinear MIMO system with flowing parameters based on NN-ANARX model with additional neural network

A dramatic improvement in quality of control can also be seen when the propose adaptation technique is used with NN-ANARX model based control by comparing figure 5.28 with figure 5.22 and figure 5.29 with figure 5.25. The proposed control system of the structure depicted in figure 5.20 is capable of tracking the desired reference signals with high accuracy compensating undesirable influence of disturbances and changes in the dynamics of the process.

It can be concluded by analyzing the results of the experiments discussed in this example that the adaptation technique proposed in section 5.7 significantly improves the robustness of both considered techniques proposed by the author for dynamic output feedback model linearization based control of nonlinear MIMO systems. Both adaptive controllers presented in section 5.7 (see figures 5.19 and 5.20) are capable of tracking the desired reference signals by modifying their behavior in response to changes in the dynamics of the process and disturbances and can be successfully applied to adaptive control of a wide class of nonlinear MIMO systems.

Nevertheless, the robustness of the controller based on dynamic output feedback linearization of NN-based Simplified ANARX structure and analytical calculation of control signals (presented in sections 5.3 and 5.4) is much higher than the robustness of the controller based on dynamic output feedback linearization of fully nonlinear NN-based ANARX structure and additional neural network for calculation of control signals (presented in sections 5.5 and 5.6). The robustness of the first one is higher even when the adaptation technique proposed in section 5.7 is added into both control systems. This adaptation technique especially dramatically increases the robustness of NN-ANARX model based control technique, but it still remains lower than the robustness of NN-SANARX model based control with adaptation.

Thus, NN-SANARX model based dynamic output linearization technique is more preferable for control of a nonlinear MIMO system if the controlled system can be identified SANARX model and functional restrictions (linearity of the first sub-layer) imposed by Simplified ANARX structure do not cause serious drawbacks in quality of identification. In case, when a system can not be identified by NN-based Simplified ANARX structure with the accuracy satisfying the needs of the model based control algorithm, a fully nonlinear NN-based ANARX structure can be used for identification of the system and control algorithm based on this model can be applied.

5.8 Conclusions

In this chapter dynamic output feedback linearization algorithm (2.36)-(2.38) is applied to control of nonlinear MIMO systems. This algorithm is based on the model of the controlled system. This model has to have an ANARX structure and can be easily obtained by training a neural network of the corresponding structure depicted in figure 3.15.

The main problem regarding practical application of this control algorithm proposed in [30] is in calculation of control signals by solving equation (2.38). As it was shown in the previous chapter, a solution can be found numerically for SISO systems, but for nonlinear MIMO systems this approach can not be used because the problem becomes much more complex. Two methods were proposed by the author to overcome this problem.

The first method is based on introducing a Simplified ANARX model. A neural network based model (5.21) having ANARX structure and linear activation function of neurons (ADALINEs) of the first sub-layer was proposed by the author and called NN-based Simplified ANARX model (or NN-SANARX). It was proofed in [30] that the dynamic output feedback linearization algorithm can be applied to models having ANARX structure. The propose NN-SANARX model is a sub-class of ANARX models and therefore can be linearized by the output feedback. This algorithm can be used for control of nonlinear MIMO systems identified by NN-SANARX model. In this case calculation of the vector of control signals comes to solving system of linear equations (5.26).

If functional restrictions imposed by Simplified ANARX structure on the model are too strong for some nonlinear system, it can be identified by a MIMO NN-ANARX structure (5.1) with all sub-layers having neurons with nonlinear activation functions. In this case, an alternative approach for application of the dynamic output feedback linearization algorithm to control of nonlinear MIMO systems is proposed by the author. This approach is based on simulation of the first sub-network of the neural network based ANARX model and training an additional simple static nonlinear neural network approximating function (2.37). This network is used to estimate the victor of controls.

The structure of the closed loop control systems corresponding to the proposed methods are presented in figures ..The effectiveness of the proposed techniques is demonstrated on numerical examples. In both cases control systems are capable of tracking the desire reference signals with high accuracy.

The robustness of the proposed approaches was also checked on a numerical example. Control of a nonlinear system was simulated by using both proposed approaches with input-output disturbances and flowing parameters of the controlled nonlinear system. The first control algorithm based on output linearization of NN-based Simplified ANARX model has shown much higher robustness than the second one based on fully nonlinear NN-ANARX model. This control system is capable of good compensation of undesirable influence of input-output disturbances and flowing parameters (up to 100%) of the controlled system even with no model adaptation.

Adaptive controllers based on both control techniques with history-stack adaptation of the model were also developed. Proposed control techniques impose restrictions on model adaptation. The first sub-layer of NN-ANARX and NN-SANARX can not by trained on-line. So, the model is divided into adaptive and non-adaptive parts and the corresponding corrections are introduced into the adaptation algorithm. Adaptation significantly increases the robustness of both control systems, especially the robustness of NN-ANARX model and additional neural network based control.

The proposed adaptive control systems are capable of tracking the desired reference signals with high accuracy modifying their behavior in response to changes in the dynamics of the process and disturbances by on-line training of the model. Controllers based on the proposed techniques can be successfully used for control of a wide class of nonlinear dynamic MIMO systems.

Chapter 6

Conclusions

Problems of nonlinear systems identification by artificial neural networks with different structures and applicability of these models for model based control algorithms are considered in this thesis. This chapter summarizes what has been studied and what are the main results achieved in this thesis.

While it is well known that double-layer perceptron is capable of approximating any continuous nonlinearity which means that most nonlinear dynamic systems can be identified by training a multilayer perceptron with external feedback on a set of input-output data, results of present research demonstrate importance of the structure choice. The structure of the neural network based model has to be chosen in accordance to the requirements of the particular control algorithm or/and corresponding to the structural features of the controlled system. Artificial neural network consisting of simple nodes (artificial neurons) interconnected in a way, which may be chosen according to our needs, is a very good if not the best tool for obtaining a model of a predefined structure.

First, choosing proper structure of the model repeating the structure of the physical system can significantly improve the quality of the model thus resulting in increased quality of a model based control algorithm. In this thesis it has been demonstrated on identification and control of Hammerstein-type nonlinear systems. The corresponding neural network structure representing Hammerstein model is proposed and applied for model based control of nonlinear systems belonging to this class. First of all these are systems with static actuator nonlinearities.

Second, variation of the structure of the network extends the number of algorithms that can be combined with neural networks based modeling and training based adaptation. In this thesis parameters of Additive NARX models for different nonlinear systems were identified by training a neural network of the corresponding structure (NN-ANARX model) making possible combination of neural network based identification and adaptation with dynamic output feedback linearization and application of this combined control technique to control of a wide class of nonlinear systems.

Model based control algorithms can be divided into algorithms not requiring any particular structure of the model (model independent algorithms) and algorithms requiring certain structure of the model (model dependant algorithms).

Quality of model independent algorithms can be improved by improving the accuracy of the model. It is shown in the thesis that proposed neural networks based Hammerstein model significantly quality of model based predictive control of systems with static actuator nonlinearities.

Predictive control, inverse model based control and NN-ANARX model based dynamic output feedback linearization control techniques are studied and compared in this thesis. Their advantages and drawbacks are derived by the author from the experimental results. Control technique based on dynamic output feedback linearization of NN-ANARX model may help to overcome several disadvantages of others techniques. This method is based only on direct dynamic model of the controlled system. The model has to work only in closed loop. No predictions calculated by stand alone model simulation are needed. So, model errors do not accumulate. In the same time, restrictions imposed by ANARX structure on the connectivity matrixes of neural network representing this model do not cause any serious drawback in quality of identification and gives significant advantages in control applications. That is why significant attention in this research is paid to identification of nonlinear systems by NN-based ANARX models and their control by dynamic output feedback linearization of this model.

The only problem regarding practical application of this technique is complexity of calculation of the control signal from the dynamics of the controller. This control algorithm was successfully applied to control of nonlinear SISO systems by using Newton's method for numerical calculation of the control signal, but this approach has two disadvantages. Firstly, numerical calculation takes several iterations to converge and, secondly, it can not be used for calculation of a vector of control signals.

NN-based ANARX structures for identification of MIMO systems was proposed. Two methods are also developed for NN-ANARX model based dynamic output feedback linearization of nonlinear MIMO systems.

NN-based Simplified ANARX (NN-SANARX) model was proposed by the author in for identification of nonlinear SISO and MIMO systems. This structure having one linear sub-layer of the hidden layer makes calculation of exact values of control signals (not estimated by a numerical method) for dynamic feedback linearization of this model much simpler. It comes to solution of a linear equation (in case of SISO systems) or a system of linear equations (in case of MIMO systems). NN-SANARX model was successfully applied to control of dynamic output feedback linearization based control of nonlinear MIMO systems with equal number of inputs and outputs. The criteria of applicability of a particular MIMO NN-SANARX model for dynamic linearization based control is also given.

An alternative approach which makes it possible to apply NN-ANARX model based dynamic output feedback linearization to control of nonlinear MIMO systems is presented and discussed in the thesis. This method utilizes additional static neural network for approximation of the function calculating control vectors.

The second approach is based on fully nonlinear not simplified (with no linear sub-layers) NN-ANARX model of a nonlinear MIMO system and that's why this method can be used for control of a wider class of nonlinear systems. At the same time it is demonstrated in the thesis that the first control technique has higher robustness. Both methods impose several restrictions on on-line model adaptation. Synaptic weights have to be fixed. The corresponding corrections have been introduced into history-stack adaptation and adaptive controllers for nonlinear MIMO systems were designed. It was shown that these controllers are capable of simultaneous tracking of several desired reference signals compensating undesirable influence of input and output disturbances as well as changes in the parameters of the controlled system.

The effectiveness of the proposed techniques is demonstrated on numerous numerical examples.

Chapter 7

Future work

Results achieved in the framework of present thesis leads variety of interesting research directions.

Science never solves a problem without creating ten more.
George Bernard Shaw

Some of these directions are

Application of NN-ANARX model based dynamic output feedback linearization to control of a wider class of nonlinear MIMO systems: systems with unequal numbers of inputs and outputs.

Development of an algebraic algorithm for calculation of control signals or vectors of control signals from NN-ANARX model without its simplification.

ANARX model has all time instances separated and is represented by a sum of nonlinear functions. Neural network based structure representing ANARX model consists of n sub-networks corresponding to the n-th order of the model. The order of the model can be very easily changed by adding or removing a corresponding sub-network without any need to retrain the remaining network. Based on this property development of ANARX model adaptation technique capable of adjusting the order of the model may make a subject of further research.

Identifiability of different classes of nonlinear systems by training NN-based ANARX and SANARX structures needs to be studied in detail.

Analogous restricted connectivity artificial neural network structures may also give advantages over classical structures in image recognition by separating different features of the image. This will also make a subject of further research.

Future research will also be pointed towards application of the techniques discussed in the thesis to modeling and contextual analysis of human motions.

These are only the main research directions. Undoubtedly, they are not limited by the above-mentioned.

References

[1] W. McCulloch and W. Pitts, “A Logical Calculus of Ideas Immanent in Nervous Activity,” *Bulletin of Mathematical Biophysics*, 1943, vol. 5, pp. 115-133.

[2] D. Cleveland, How Do We Know How the Brain Works, Rosen Publishing Group, 2005.

[3] J. L. Shearer, B. T. Kulakovski, J. F. Gardner, Dynamic Modeling and Control of Engineering Systems, Prentice Hall, Upper Saddle River, New Jersey, 1997.

[4] H. K. Khalil, Nonlinear Systems, Macmillan Publishing Company, New York, 1992.

[5] H. Nijmeier, A. J. van der Schaft, Nonlinear Dynamical Control Systems, Springer-Verlag, New-York, 1990.

[6] H. J. Marquez, Nonlinear Control Systems, John Wiley & Sons, New Jersy, 2003.

[7] R. Iserman, Digital Control Systems, Springer-Verlag Berlin, Heidelberg, 1981.

[8] Ü . Kotta, F. Chowdhury, and S Nõmm, “On realizability of neural networks-based input-output models in the classical state space form,” *Automatica*, 2006, vol. 42, no. 7, pp. 1211-1216.

[9] Ü. Kotta and N. Sadegh, “Two approaches for state space realization of NARMA models: bridging the gap,” *Mathematical and Computer Modelling of Dynamical Systems*, vol. 8, no. 1, pp. 21-32, 2002.

[10] F. N. Chowdhury, Ü. Kotta and S. Nõmm, “On realizability of neural-networks-based input-output models,” *Proc. of the 3rd Int. Conf. on Differential Equations and Applications, St. Petersburg, Russia*, vol 6., pp. 47-51, 2000.

[11] Ü. Kotta, S. Nõmm, and F. Chowdhury, “On a new type of neural network-based input-output model: The ANARMA structure,” *Proc. of the 5th IFAC Symposium on nonlinear control systems NOLCOS, St. Petersbourg, Russia*, July 2001.

[12] F. N. Chowdhury. “Input-output modeling of nonlinar systems with time-varying linear models. *IEEE Transactions on Automatic Control*, vol. 7, pp. 1355-1358, 2000.

[13] E. Petlenkov, S. Nõmm, and Ü. Kotta, “NN-based ANARX structure for identification and model-based control,” *Proc: 9th International Conference on Control Automation Robotics & Vision (ICARCV 2006),* Singapore, 2006, pp. 2284-2288.

[14] E. Petlenkov, S. Nõmm, and Ü. Kotta, “Adaptive Output Feedback Linearization for a Class of NN-based ANARX Models,” *Proc: 6th IEEE International Conference on Control and Automation,* Guangzhou, China, May 30 – June 1, 2007, pp. 3173-3178.

[15] K. Ogata, State Space Analysis of Control Systems, Prentece-Hall, Inc., Englewood Cliffs, N.J., 1967.

[16] M. Mrabet, F. Fnaiech, A. Chaari, and K. Al-Haddad, “Nonlinear Predictive Control Based on NARX Models with Structure Identification,” *in Proc. of the 28th Annual IEEE Conference of the Industrial Electronics Society (IECON02)*, vol. 3, Sevilla, Spain, November 2002, pp. 1757-1762.

[17] K. J. Hunt, G. R. Irwin and K. Warwick, Neural Network Engineering in Dynamic Control Systems: Advances in Industrial Control, Springer-Verlag, London, 1995.

[18] J. B. Rowlings, “Tutorial: Model Predictive Control Technology,” *in Proc. of the American Control Conference*, *1999*, Vol. 1, San Diego, California, USA, June 1999, pp 662-676.

[19] V. Peterka, “Predictor Based Self-Turning Control,” *Automatica*, vol. 20, no. 1, pp 39-50, 1984.

[20] D. W. Clarke, C. Mohtadi and P. S. Tuffs, “Generalized Predictive Control – Part 1. The basic algorithm,” *Automatica*, 1987.

[21] C. Kambhapati, K. Warwick and C. S. Berger, “A Comparative Study of Multilayered and Single Layered Neural Network Based Predictive Controllers,” *IEE International Conference on Intelligent Systems Engineering*, pp. 293-298, 1992.

[22] H. Chen and F. Allgöwer, “A quasi-infinit horizon nonlinear model predictive control scheme with garanteed stability,” Automatica, vol. 34, no. 10, pp. 1205-1217, 1998.

[23] G. W. Irwin., K. Warwick., K. J. Hunt., Neural network applications in control, The Institution of Electrical Engineers, 1995.

[24] S. Omatu, M. Khalid and R. Yusof, Neuro-Control and its Applications: Advances in Industrial Control, Springer-Verlag, London, 1996.

[25] F. Garces, V. M. Becerra, C. Kambhampai and K. Warwick, Strategies for Feedback linearization. A Dynamic Neural Network Approach, Springer-Verlag, London, 2003.

[26] E. Petlenkov and E. Rüstern, “Experiments in Neural Network Inverse Modelling Based Control for a Class of Nonlinear Systems,” *Proc. of the 9th Biennial Baltic Electronics Conference: BEC2004*, Tallinn, Estonia, October 3-6, 2004, pp. 145-148.

[27] M.V. Chen, M. S. Zalzala and N. E. Sharkey, “Towards a Comparative Study of Neural Networks in Inverse Model Learning and Compensation Applied to Dynamic Robot Control,” *Proc of the Fifth International Conference on Artificial Neural Networks (Conf. Publ. No. 440)* , 7-9 July 1997, pp. 146-151.

[28] E. Colina-Morles and N. Mort, “Inverse Model Neural Network-Based Control of Dynamic Systems,” *Control '94. International Conference*, vol. 2, 21-24 March 1994, pp.955-960.

[29] G. P. Plett, “Adaptive Inverse Control of Linear and Nonlinear Systems Using Dynamic Neural Networks,” *IEEE Trans. on Neural Networks*, vol.14, no.2, March 2003, pp.360-376.

[30] S. Ushida and H. Kimura, “FEL and JIT Approaches to Tracking Adaptive Control Based on the Internal Inverse Models,” *Proc. Of 42nd IEEE Conference on Decision and Control,* Vol. 6 , December 9-12, 2003, pp. 6363 – 6368.

[31] R. Boukezzoula, S. Galichet, and L. Foulloy, “Nonlinear Internal Model Control: Application of Inverse Model Based Fuzzy Control,” *IEEE Transactions on Fuzzy Systems*, Vol. 11, Issue 6, December, 2003, pp. 814-829.

[32] R. Pothin, Ü . Kotta, and C. Moog, “Output feedback linearization of nonlinear discrete-time systems,” *Proc. of the IFAC Conf. on Control system design*, Bratislava, 2000, pp. 174–179.

[33] S. Nõmm, Realization and Identification of Discrete-time Nonlinear Systems, *PhD thesis*, Tallinn University of Technology, Tallinn, 2004.

[34] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, Numerical Recipes in C: The Art of Scientific Computing, Cambridge University Press, 1992.

[35] S. R. Cajal, “Histologie du systeme nerveux de l-homme et des vertebres”, Maloine, Paris, 1909.

[36] S. Haykin, Neural Networks, Prentice-Hall international (UK) Limited, London, 1994.

[37] Y-Z. Lu, Industrial intelligent control. Fundamentals and applications. Chicher: Wiley, 1996.

[38] T. Hrycey, Neurocontrol. Towards an Industrial Control Methodology, John Wiley & Sons, Inc., Toronto, 1997.

[39] P. M. Mills, A. Z. Zomaya and M. O. Tade, Neuro-Adaptive Process Control, John Wiley & Sons, Inc., England, 1996.

[40] L. H. Tsoukalas and R. E. Uhrig, Fuzzy and neural approaches in engineering. Wiley-Interscience, New York, 1996.

[41] D. E. Rumelhart and J. L. McCLelland, Parallel Distributed Processing, vol. 1 Foundation, MIT Press,Cambridge, 1988.

[42] K. J. Åström and B. Wittenmark, "On Self-Turning Regulators," *Automatica*, 1973, vol. 9, pp. 185-199.

[43] K. J. Åström and B. Wittenmark. Adaptive control. Addison-Wesley Publishing Company, 1989.

[44] M. H. Stone, "The generalized Weierstrass approximation theorem," *Mathematics Magazine*, vol. 21, 1948, pp. 167-184, 237-254.

[45] J. L. Elman, "Finding Structure in Time," *Cognitive Science*, vol. 14, pp. 179-211, 1990.

[46] E. Haselsteiner, **"**What Elman networks cannot do,**"** Proc. Of The 1998 IEEE International Joint Conference on Neural Networks. IEEE World Congress on Computational Intelligence, vol. 2, pp. 1245-1249, Anchorage, Alaska, USA May 1998.

[47] E. Sontag, "Neural nets as systems models and controllers," *Proc. Of the Seventh Yale Workshop on Adaptive and Learning Systems*, pp. 73-79, Yale University, 1992.

[48] D. O. Hebb, The Organization of behavior, Wiley, New York, 1949

[49] F. Rosenblatt, "Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms, " Spartan Book, Washington, 1961.

[50] K. Levenberg, "A method for the solution of certain problems in least squares, *Quart. Appl. Math.*, 1944, Vol. 2, pp. 164–168.

[51] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM J. Appl. Math.*, 1963, Vol. 11, pp. 431–441.

[52] A. Toledo, M. Pinzolas, J. J. Ibarrola and G. Lera, "Improvement of the neighborhood based Levenberg-Marquardt algorithm by local adaptation of the learning coefficient," *Neural Networks, IEEE Transactions on*, Vol. 16, Issue 4, July 2005, pp. 988-992.

[53] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *Neural Networks, IEEE Transactions on*, Vol. 5, Issue 6, November 1994, pp. 989-993.

[54] F. Declercq and R. De Keyser, "Comparative study of neural predictors in model based predictive control" *Proc. of The International Workshop on Neural Networks for Identification, Control, Robotics,*

and Signal/Image Processing, Venice, Italy, 21-23 Aug. 1996, pp. 20-28.

[55] E. Petlenkov and E. Rüstern, "Linear Dynamic Systems with Static Actuator Nonlinearities Identification for Control," *Proceedings of the ICGST Automatic Control and System Engineering Conference: ACSE-05*, December 19-21, 2005, Cairo, Egypt, pp. 41-46.

[56] E. Petlenkov, „NN-ANARX Structure Based Dynamic Output Feedback Linearization for Control of Nonlinear MIMO Systems," *In Proc. Of The 15th Mediterranean Conference on Control and Automation, MED'07*, Athena, Greece, June 2007, pp. 1-6.

[57] E. Petlenkov, and J. Belikov, "NN-ANARX Structure for Dynamic Output Feedback Linearization of Nonlinear SISO and MIMO Systems: Neural Networks Based Approach," *In Proc. Of the 26th Chinese Control Conference*, Zhangjiajie, China , July 2007, vol. 4, pp. 138-145.

[58] H. Demuth, M. Beale, Neural network toolbox: for use with MATLAB, The MathWork, Inc., 1998.

[59] C. L. Harris, and X. Hong, "Neurofuzzy mixture of experts network parallel learning and model construction algorithms," *IEE Proc.-Control Theory Applications*, vol. 148, num. 6, November, 2001, pp. 456-466.

[60] F. C. Chen and H. K. Khalil, "Adaptive Control of a class of nonlinear discrete-time systems using neural networks," *IEEE transactions on Automatic Control,* 1995, vol. 40, num. 5, pp. 791-801.

[61] S. Chen, S. Billings, and P. Grant, "Non-linear system identification using neural networks," *International Journal on Control,* 1990, vol. 51, no. 6, pp. 1191–1214.

[62] K. Narendra, and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks,* March 1990, vol. 1, pp. 4–27.

[63] K. Narendra, and S. Mukhopadhyay, "Adaptive Control Using Neural Networks and Approximate Models," *IEEE Tranactionss on Neural Networks,* vol. 8, no. 3, pp. 475-485, 1997.

[64] H. T. Siegelmann, B. G. Horne and C. L. Giles, "Computational Capabilities of Recurrent NARX Neural Networks," *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 27, no. 2, April 1997, pp. 208-215.

[65] S. Billings, "Identification of Nonlinear Systems - A survey," *Proceedings of IEE, Part D,* 1980, pp. 127-272.

[66] G. A. Pajunen, "Application of a Model Reference Adaptive Technique to the Identification and Control of Wiener Type Nonlinear Processes," *Thesis for the degree of Doctor of Technology*, Helsinki University of Technology, Espo, Finland, 1984.

[67] B.-J. Yang, A. J. Calise, and N. Hovakimyan, "Augmenting adaptive output feedback control of uncertain nonlinear systems with actuator nonlinearities," *Proceedings of the 2004 American Control Conference*, Boston, Massachusetts, Vol. 5, June 30 - July 2, 2004, pp. 4675-4680.

[68] M. Polycarpou, J. Farrell and M. Sharma, "On-line Approximation Control of Uncertain Nonlinear Systems: Issues with Control Input Saturation," *Proceedings of the 2003 American Control Conference*, Denver, Colorado, Vol. 1 , 4-6 June 2003, pp 543-548.

[69] R. R. Selmic, V. V. Phoha, and F. L. Lewis, "Intelligent Compensation of Actuator Nonlinearities," *Proc. Of the 42nd IEEE Conference on Decision and Control*, Maui, Hawaii, Vol. 4 , 9-12 Dec, 2003, pp 4327-4332.

[70] C. K. Lee, T. H. M. Chow and D. K. W. Cheng, "DC Servo Motor Controllers Based on a Neural Network," *Proc. of the First International Conference on Intelligent Systems Engineering*, Conf. Publ. No. 360, 19-21 Aug 1992, pp. 287-292.

[71] Laboratoire de Meterologie Dynamique. Chemical and nuclear engineering. System dynamics. Mathematical Modeling of Engineering Systems, "Dynamic Model of a Permanent Magnet DC Motor," http://gershwin.ens.fr/vdaniel/Doc-Locale/Cours-Mirrored/Methodes-Maths/white/sdyn/s6/s6fmathm/s6fmathm.html [February, 2007].

[72] S. A. Billings and M. B. Fadzil, "The practical identification of systems with nonlinearities," *Proc. of the 7th IFAC/IFORS Symposium on Identification and System Parameter Estimation*, 1985, pp. 155–160.

[73] S. Nõmm, E. Petlenkov, J. Vain, K. Yoshimitsu, K. Ohnuma, T. Sadahiro and F. Miyawaki, „NN-based ANARX Model of the Surgeon's Hand for the Motion Recognition," *Proc. of the 4th COE Workshop on Human Adaptive Mechatronics (HAM),* Tokyo Denki University, Japan, March 2007, pp. 19-24.

[74] R. Pearson, Ü. Kotta, and S. Nõmm, "Systems with associative dynamics," *Kybernetika*, vol. 38, no. 5, pp. 585–600, 2002.

[75] T. Knapp and H. Budman, "Robust control design of non-linear processes using empirical state-affine models," *Int. J. Control*, no. 73 pp. 1525–1535, 2000.

[76] J. Belikov, E. Petlenkov and S. Nõmm, „Application of Neural Networks based ANARX structure to backing up control of a truck-trailer,“ *in Proc. of the 6th IFAC Symposium on Intelligent Autonomous Vehicles*, Tolouse, France, September 2007.

[77] T. F. Junge and H. Unbehauen, "Recursive Identification of a Turbo-Generator Plant Using Structurally Adaptive Neural Networks," *in Proc. of IEEE International Conference on Industrial Technology 2000*, Goa, India, vol. 1, pp. 572-577, January 2000.

[78] F. Miyawaki, K. Masamune, S. Suzuki, K. Yoshimitsu, and J. Vain, "Tscrub nurse robot system – intraoperative motion analysis of a scrub nurse and timed-automatabased model for surgery," *IEEE Transaction on Industrial Electronics*, vol. 5 no. 52, pp. 1227-1235, 2005.

[79] K. Ohnuma, K. Masamune, K. Yoshimitsu, F. Miyawaki, J. Vain, and Y. Fukui, "Analysis and recognition of a surgeon's motions in laparoscopic cholecystectomy giving a scrub nurse robot suitable timings for instrument exchange," *In Proceedings of the 3rd COE Workshop on Human Adaptive Mechatronics (HAM)*, Tokyo Denki University, Japan, pp. 1–5, March 2006.

[80] R. Alur, and D. L. D. Gerd, "Theory of timed automata," *TCS*, vol. 2, no. 126, pp. 183–235, 2004.

[81] T. E. Marlin, G. W. Barton, M. L. Brisk, and J. D. Perkins, "Advanced Process Control," *Project Report*, University of Sidney, 1987.

[82] K. J. Hunt, D. Sbarbano, R. Žbikowski, and P. J. Gawthorp, "Neural networks for control systems: A survey," *Automatica*, vol. 28, no. 6, pp. 1083-1112, 1992.

[83] D. A. White, and D. A. Sofge, "Handbook of Intelligent Control," Van Nostrand Reinhold, New York, 1992.

[84] W. T. Miller, R. S. Sutton, and P. J. Werbos, "Neural Networks for Control," MIT Press, Cambridge, 1990.

[85] J. E. Dayhoff, "Neural Network Architectures: An Introduction," Van Nostrand Reinhold, New York, 1990.

[86] S. A. Billings, and S. Chen, "Neural Networks and System Identification", In K. Warwick *et al.,* (eds.) *Neural networks for systems and control,* pp.181-205, 1992.

[87] K. S. Narendra, and F. L. Lewis, "Introduction to the special issue on neural network feedback control," *Automatica*, August 2001, vol. 37, no. 8, pp. 1147-1148.

[88] S. Fabri, and V. Kadirkamanathan, "Dynamic structure neural networks for stable adaptive control of nonlinear systems", *IEEE Transactions on Neural Networks,* vol. 7, iss. 5, Sept. 1996, pp. 1151-1167.

[89] G. P. Liu V. Kadirkamanathan and S. A. Billings, "Variable neural networks for adaptive control of nonlinear systems," *IEEE Transactions on System, Man and Cybernetics, Part C,* vol. 29, iss. 1, Feb. 1999, pp. 34-43.

[90] G. P. Liu V. Kadirkamanathan and S. A. Billings, "Nonlinear Predictive Control via Neural Networks," *In Proceedings of the UKCC International Conference on Control'96,* vol. 2, Sept. 1996, pp. 746-751.

[91] M. Alayon, D. Saez, and R. Veiga, "Comparative Analysis of Neural Predictive Controllers and Its Application to a Laboratory Tank System," *In Proceedings of 2004 IEEE International Joint Conference on Neural Networks*, Budapest, Hungary 25-29 July 2004, pp. 1249-1254.

[92] S. Kumarawadu, and T.-T. Lee, "A Model-based Neurocontrol Approach for Car-Following Collision Prevention," *In Proceedings of 2004 IEEE International Conference on Networking, Sensing and Control,* vol. 1, March 2004, pp. 152-157.

[93] M. A. Zayan, "Satellite Orbits Guidance Using State Space Neural Network," *2006 IEEE Aerospace Conference*, Montana, March 2006, pp. 1-16.

[94] J. F. Li, L. Gao, "Neural Network Control Approach for Improving Vehicle Stability," *In Proceedings of the 9th International Conference on Control, Automation, Robotics and Vision (ICARCV2006)*, Singapore, December 2006, pp. 2425-2428.

[95] K. J. Hunt, D. Sbararo, R. Zbikowski, and P. J. Gawthrop, "Neural networks for control systems - a survey," *Automatica,* Vol. 28, No. 6, pp. 1083-1112, 1992.

[96] M. J. Willis, G. A. Montague, C. Di. Massimo, M. T. Tham, and A. J. Morris, "Artificial neural networks in process estimation and control", *Automatica,* Vo1.28, No. 6, pp.1181-1187, 1992.

[97] G. A. Montague, M. J. Willis, and A. J. Morris, "Artificial Neural Network Model Based Control," *In Proc. Of the American Control Conference*, Vol. 2 , 29 June - 1 July 1994, pp 2134-2138.

[98] X. Wang, and Y. He, "Fuzzy Neural Network based Predictive Control for Active Power Filter," *in Proc. Of the International Conference on Power System Technology*, Chongquing, China, October 2006, pp. 1-5.

[99] C. Yang, and P. Wu, "Neural Networks Based Predictive Control for TRT," *in Proc. Of the International Conference on Neural Networks and Brain (ICNN&B'05)*, Beijing, China, October 2005, pp. 1041-1044.

[100] H. N. Hazem, and K. M.Passino, “Stable auto-tuning of hybrid adaptive fuzzy/neural controllers for nonlinear systems,” Engineering Applications of Artificial Intelligence. Vol. 18, No. 3, April 2005, pp. 317-335.

[101] S. Ushida, H. Kimura, “FEL and JIT Approaches to Tracking Adaptive Control Based on the Internal Inverse Models,” *In Proc. of the 42nd IEEE Conference on Decision and Control*, Vol. 6 , December 2003, pp. 6363 – 6368.

[102] G. C. M. De Abreu, R. L. Teixeira, J. F. Ribeiro, “A neural network-based direct inverse control for active control of vibrations of mechanical systems,” *In Proc. Of the 6th Brazilian Symposium on Neural Networks*, Nov. 2000, pp. 107-112.

[103] O. Aboulshamat, P. Sicard, “Position control of a flexible joint with friction using neural network feedforward inverse models,” *In Proc of the Canadian Conference on Electrical and Computer Engineering, 2001,* vol. 1, May 2001, pp. 283-288.

[104] C. Yu, J. Zhu, Z. Sun, “Nonlinear adaptive internal model control using neural networks for tilt rotor aircraft platform,” *In Proc. Of the 2005 IEEE Mid-Summer Workshop on Soft Computing and Industrial Applications*, June 2005, Helsinki University of Technology, Espoo, Finland, pp. 12-16.

[105] H. Chaoui, P. Sicard, A. Lakhsasi, “FPGA Implementation of Neural Network Based Adaptive Control of a Flexible Joint with Hard Nonlinearities,” *In Proc. Of the 2006 IEEE International Symposium on Industrial Electronics*, Montreal, Canada, vol. 4, July 2006, pp. 3118-3123.

[106] W. Wu, and W.-C. Hsu, “Adaptive neural-network predictive control for nonminimum-phase systems,” *in Proc. of the American Control Conference*, *2006*, June 2006, Minneapolis, Minnesota USA, pp. 2981-2986.

[107] H. Ichihashi, and M. Tokunaga. “Neuro-fuzzy optimal control of backing up a trailer truck,” *In Proc. Of the IEEE International Conference on Neural Networks*, vol. 1, San Francisco, California, March 28 – April 1, 1992, pp. 144-149.

[108] K. Tanaka, and M. Sano, “A robust stabilization problem of fuzzy control systems and its application to backing up control of a truck-trailer,” *IEEE Transactions on Fuzzy Systems*, vol. 2, no. 2, May 1994, pp. 119-134.

[109] S. Kong, and B. Kosko, “Adaptive fuzzy systems for backing up a truck-and-trailer,” *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 2, pp. 211-223.

[110]D. Nguyen, and B. Widrow, “The truck backer-upper: An example of self-learning in neural networks,” In Proc. Of the International Joint Conference on Neural Networks (IJCNN-89). Vol. 2, June, 1989, pp. 357-363.

[111]T. Mullari, Ü. Kotta, and M. Tõnso, “Equivalence of different static state feedback linearizability conditions of discrete time nonlinear control systems,” *In Proceedings of the European Control Conference*, Greece, 2007, Accepted.

[112]K. Vassiljeva, and E. Rüstern, “On-Line Identification and Adaptive Control of Nonlinear Systems using Neural Networks,” *In Proc. of the 9th Biennial Baltic Electronics Conference: BEC2004*, Tallinn, Estonia,October 3-6, 2004, pp. 149-152.

[113]X. Li, Y. Bai and L. Yang, “Neural Network Online Decoupling for a Class of Nonlinear System,” *In Proc. of the 6th World Congress on Intelligent Control and Automation*, Dalian, China, June 21-23, 2006, pp. 2920-2924.

[114]H. L. Shu, “PID neural network for analysis of the multivariable systems,” *Automation transaction*, vol.25, no.1, 1999, pp. 105-111.

[115]F. Song and P. Li, “MIMO Decoupling Control Based on Support Vector Machines αth-order Inversion,” *In Proc. of the 6th World Congress on Intelligent Control and Automation*, Dalian, China, June 21-23, 2006, pp. 1002-106.

[116]R. Ordones and K. M. Passino, “Stable multi-output adaptive fuzzy-neural control,” *IEEE Transactions on Fuzzy Systems*, vol. 7, 1999, pp. 345-353.

[117]S. Tong, Y. Shi, “Adaptive fuzzy output feedback control for nonlinear MIMO systems,” *In Proc. of IEEE International Conference on Fuzzy Systems*, Budapest, Hungary, 2004, vol. 3, pp. 1209-1213.

[118]T. Wang, “Indirect Adaptive Fuzzy Output Feedback Control for Nonlinear MIMO System,” *In Proc. of the 5th World Congress on Intelligent Control and Automation*, Hangzhou, China, June 2004, pp. 502-505.

[119]T. Yiquian, W. Jianhui, G. Shusheng and Q. Fengying, “Fuzzy Adaptive Output Feedback Control for Nonlinear MIMO Systems Based On Observer,” *Proc. of the 5th World Congress on Intelligent Control and Automation*, Hangzhou, China, June 2004, pp. 506-510.

List of publications

J. Belikov, E. Petlenkov, and S. Nõmm, „Application of Neural Networks based ANARX structure to backing up control of a truck-trailer,“ *Proc. of the 6th IFAC Symposium on Intelligent Autonomous Vehicles*, September 2007, Tolouse, France, pp. 1-5.

E. Petlenkov, and J. Belikov, "NN-ANARX Structure for Dynamic Output Feedback Linearization of Nonlinear SISO and MIMO Systems: Neural Networks Based Approach," *Proc. of the 26th Chinese Control Conference (CCC2007)*, Zhangjiajie, China, July 2007, vol. 4 pp. 138-145.

E. Petlenkov „NN-ANARX Structure Based Dynamic Output Feedback Linearization for Control of Nonlinear MIMO Systems,“ *Proc. of the 15th Mediterranean Conference on Control and Automation, MED'07*, June 2007, Athena, Greece, pp. 1-6.

E. Petlenkov, S. Nõmm, and Ü. Kotta, "Adaptive Output Feedback Linearization for a Class of NN-based ANARX Models," *Proc. of the 6th IEEE International Conference on Control and Automation,* Guangzhou, China, May 30 – June 1, 2007, pp. 3173-3178.

E. Petlenkov, "Neural Networks Based Simplified ANARX Structure for Control of Nonlinear MIMO Systems," *IKTDK teise aastakonverentsi artiklite kogumikus*, 11.-12. mai 2007, Viinistu, lk. 39-42.

S. Nõmm, E. Petlenkov, J. Vain, K. Yoshimitsu, K. Ohnuma, T. Sadahiro, and F. Miyawaki, „NN-based ANARX Model of the Surgeon's Hand for the Motion Recognition," *Proc. of the 4th COE Workshop on Human Adaptive Mechatronics (HAM),* Tokyo Denki University, Japan, March 2007, pp. 19-24.

E. Petlenkov, S. Nõmm, and Ü. Kotta, "Neural Networks based ANARX structure for identification and model-based control," *Proc. of the 9th International Conference on Control Automation Robotics & Vision (ICARCV 2006),* Singapore, 2006, pp. 2284-2288.

E. Petlenkov, "NN-ANARX Modeling Based Dynamic Feedback Linearization for a Class of Nonlinear Systems," *IKTDK esimese aastakonverentsi artiklite kogumikus*, 12.-13. mai 2006, Jäneda Mõis, lk. 99-102.

E. Petlenkov, and E. Rüstern, "Linear Dynamic Systems with Static Actuator Nonlinearities Identification for Control," *Proceedings of the ICGST Automatic Control and System Engineering Conference: ACSE-05*: December 19-21, 2005: Cairo, Egypt, pp. 41-46.

E. Petlenkov, and E. Rüstern, "Experiments in Neural Network Inverse Modelling Based Control for a Class of Nonlinear Systems," *Proc. of the 9th Biennial Baltic Electronics Conference: BEC2004*: October 3-6, 2004: Tallinn, Estonia, pp. 145-148.

List of abbreviations

ADALINE	ADAptive LINear Element
ANARX	Additive Nonlinear AutoRegressive eXogenous
BP	BackPropagation
CSTR	Continuous Stirred Tank Reactor
DC	Direct Current
FEL	Feedback Error Learning (in [101])
FPGA	Field Programmable Gate Array (in [105])
JIT	Just-In-Time (in [101])
HSA	History Stack Adaptation
LM	Levenberg-Marquardt
MIMO	Multiply Input Multiply Output
MSE	Mean Square Error
MISO	Multiply Input Single Output
NARX	Nonlinear AutoRegressive eXogenous
NN	Neural Network
NN-ANARX	Neural Network based Additive Nonlinear AutoRegressive eXogenous
NN-NARX	Neural Network based Nonlinear AutoRegressive eXogenous
NN-SANARX	Neural Network based Simplified Additive Nonlinear AutoRegressive eXogenous
RL	Reinforcement Learning
SIMO	Single Input Multiple Output
SISO	Single Input Single Output
SNR	Scrub Nurse Robot
TRT	Top Gas Pressure Recovery Turbine (in [99])

Elulookirjeldus

1. Isikuandmed
   Ees- ja perekonnanimi: Eduard Petlenkov
   Sünniaeg ja -koht: 28.06.1979, Tallinn, Eesti
   Kodakondsus: Eesti
   Perekonnaseis: abielus
   Lapsed: tütar Elisabet, sünniaasta 2006

2. Kontaktandmed
   Aadress: Võru tn. 8-58, 13612, Tallinn, Eesti
   Telefon: +372 56 622 694
   E-posti aadress: eduard.petlenkov@dcc.ttu.ee

3. Hariduskäik

Õppeasutus (nimetus lõpetamise ajal)	Lõpetamise aeg	Haridus (eriala/kraad)
Tallinna Tõnismäe Reaalkool	Juuni 1997	Gümnaasiumi haridus
Tallinn Tehnikaülikool, arvuti- ja süsteemitehnika teaduskond	Juuni 2001	Tehnikateaduste bakalaureus
Tallinn Tehnikaülikool, Infotehnoloogia teaduskond,	Juuni 2003	Tehnikateaduste magister

4. Keelteoskus (alg-, kesk- või kõrgtase)

Keel	Tase
Vene	kõrgtase
Eesti	kõrgtase
Inglise	kõrgtase
Saksa	algtase

5. Täiendõpe

Õppimise aeg	Õppeasutuse või muu organisatsiooni nimetus
10-22. Veebruar 2002	Poola-Jaapani Infotehnoloogia Instituut

6. Teenistuskäik

Töötamise aeg	Ülikooli, teadusasutuse või muu organisatsiooni nimetus	Ametikoht
september 2002 -	Tallinna Tehnikaülikool, Automaatikainstituut	Assistent
Jaanuar-august 2002	Tallinna Tehnikaülikool, Automaatikainstituut	Insener
Juuli-august 2001	Infineon Technologies, Corporate Research, Systems Technology, München, Saksamaa	Teadur
Juuli-august 2000	JOT Eesti	Elektrik

7. Teadustegevus

Osalus teadusprojektides:

SF projekt: T590 Arukad komponendid ja nende ühendamise probleemid (01.01.2003 – 31.12.2007), täitja

ETF grant: ETF6837 Keerukate süsteemide juhtimise robustsed meetodid: integreeritud lähenemine (01.01.2006 - 31.12.2009), täitja

ETF grant: ETF5170 Robustsed meetodid ja algoritmid süsteemide juhtimiseks, (01.01.2002 - 31.12.2005), täitja

ETF grant: ETF3423 Dünaamiliste süsteemide juhtimise ja modelleerimise meetodid (01.01.1998 - 31.12.2001), täitja

8. Kaitstud lõputööd

   Mittelineaarsete süsteemide identifitseerimine tehisnärvivõrkudega (tehnikateaduste bakalaureuse kraad, 2001)

   Tehisnärvivõrkudega identifitseerimisel baseeruv juhtimine ühe mittelineaarsete süsteemide klassi jaoks (tehnikateaduste magistri kraad, 2003)

9. Teadustöö põhisuunad

   mittelineaarsed süsteemid,
   mittelineaarsete süsteemide identifitseerimine ja juhtimine,
   tehisnärvivõrgud

10. Teised uurimisprojektid

    Meditsiinilise õe roboti väljatöötamine

Kuupäev: 07.07.2007

Curriculum Vitae

1. Personal information
   Name: Eduard Petlenkov
   Place and date of birth: 28.06.1979, Tallinn, Estonia
   Citizenship: Estonian
   Marital status: married
   Children: daughter Elisabet, 1 year

2. Contact information
   Address: Võru str. 8-58, 13612, Tallinn, Estonia
   Phone: +372 56 622 694
   E-mail: eduard.petlenkov@dcc.ttu.ee

3. Education

Institution	Graduation date	Education
Tallinn Tynismae Real School	June 1997	Secondary
Tallinn Technical University, Faculty of Computer and System Engineering	June 2001	B. Sc.
Tallinn University of Technology, Faculty of Information Technology	June 2003	M. Sc.

4. Languages

Language	Level
Russian	very good
Estonian	very good
Inglish	very good
German	beginer

5. Additional studies

Time of studies	Institution
February 10-22, 2002	Polish-Japanese Institute of Information Technology

6. Professional Employement

Date	Organisation	Position
September, 2002 -	Tallinn University of Technology, Department of Computer Control	Assistant
January-august, 2002	Tallinn University of Technology, Department of Computer Control	Ingeneer
July-august, 2001	Infineon Technologies, Corporate Research, Systems Technology, Munich, Germany	Researcher
July-august, 2000	JOT Eesti	Electrician

7. Scientific projects

SF project: T590 Intelligent components and their integration problems (01.01.2003 – 31.12.2007), performer

ETF grant: ETF6837 Robust methods for complex systems control: an integrated approach (01.01.2006 - 31.12.2009), performer

ETF grant: ETF5170 Robust methods for control of dynamic systems, (01.01.2002 - 31.12.2005), performer

ETF grant: ETF3423 Methods for Control and Modeling of Dynamic Systems (01.01.1998 - 31.12.2001), performer

8. Theseses

   Identification of nonlinear systems with artificial neural networks (B. Sc., 2001)

   Identification-based neural control for a class of nonlinear systems (M. Sc., 2003)

9. Main Areas of Scientific Work

   Nonlinear systems,
   identification and control of nonlinear systems,
   artificial neural networks

10. Other Research Projects

    Development of algorithms for scrub nurse robot

Date: 07.07.2007

DISSERTATIONS DEFENDED AT TALLINN UNIVERSITY OF TECHNOLOGY ON *INFORMATICS AND SYSTEM ENGINEERING*

1. **Lea Elmik**. Informational modelling of a communication office. 1992.
2. **Kalle Tammemäe**. Control intensive digital system synthesis. 1997.
3. **Eerik Lossmann**. Complex signal classification algorithms, based on the third-order statistical models. 1999.
4. **Kaido Kikkas**. Using the Internet in rehabilitation of people with mobility impairments – case studies and views from Estonia. 1999.
5. **Nazmun Nahar**. Global electronic commerce process: business-to-business. 1999.
6. **Jevgeni Riipulk**. Microwave radiometry for medical applications. 2000.
7. **Alar Kuusik**. Compact smart home systems: design and verification of cost effective hardware solutions. 2001.
8. **Jaan Raik**. Hierarchical test generation for digital circuits represented by decision diagrams. 2001.
9. **Andri Riid**. Transparent fuzzy systems: model and control. 2002.
10. **Marina Brik**. Investigation and development of test generation methods for control part of digital systems. 2002.
11. **Raul Land**. Synchronous approximation and processing of sampled data signals. 2002.
12. **Ants Ronk**. An extended block-adaptive Fourier analyser for analysis and reproduction of periodic components of band-limited discrete-time signals. 2002.
13. **Toivo Paavle**. System level modeling of the phase locked loops: behavioral analysis and parameterization. 2003.
14. **Irina Astrova**. On integration of object-oriented applications with relational databases. 2003.
15. **Kuldar Taveter**. A multi-perspective methodology for agent-oriented business modelling and simulation. 2004.
16. **Taivo Kangilaski**. Eesti Energia käiduhaldussüsteem. 2004.
17. **Artur Jutman**. Selected issues of modeling, verification and testing of digital systems. 2004.
18. **Ander Tenno**. Simulation and estimation of electro-chemical processes in maintenance-free batteries with fixed electrolyte. 2004.

19. **Oleg Korolkov**. Formation of diffusion welded Al contacts to semiconductor silicon. 2004.

20. **Risto Vaarandi**. Tools and techniques for event log analysis. 2005.

21. **Marko Koort**. Transmitter power control in wireless communication systems. 2005.

22. **Raul Savimaa**. Modelling emergent behaviour of organizations. Time-aware, UML and agent based approach. 2005.

23. **Raido Kurel**. Investigation of electrical characteristics of SiC based complementary JBS structures. 2005.

24. **Rainer Taniloo**. Ökonoomsete negatiivse diferentsiaaltakistusega astmete ja elementide disainimine ja optimeerimine. 2005.

25. **Pauli Lallo.** Adaptive secure data transmission method for OSI level I. 2005.

26. **Deniss Kumlander**. Some practical algorithms to solve the maximum clique problem. 2005.

27. **Tarmo Veskioja**. Stable marriage problem and college admission. 2005.

28. **Elena Fomina**. Low power finite state machine synthesis. 2005.

29. **Eero Ivask**. Digital test in WEB-based environment 2006.

30. **Виктор Войтович**. Разработка технологий выращивания из жидкой фазы эпитаксиальных структур арсенида галлия с высоковольтным p-n переходом и изготовления диодов на их основе. 2006.

31. **Tanel Alumäe**. Methods for Estonian large vocabulary speech recognition. 2006.

32. **Erki Eessaar**. Relational and object-relational database management systems as platforms for managing softwareengineering artefacts. 2006.

33. **Rauno Gordon**. Modelling of cardiac dynamics and intracardiac bio-impedance. 2007.

34. **Madis Listak**. A task-oriented design of a biologically inspired underwater robot. 2007.

35. **Elmet Orasson**. Hybrid built-in self-test. Methods and tools for analysis and optimization of BIST. 2007.