THESIS ON INFORMATICS AND SYSTEM ENGINEERING C74

Restricted Connectivity Neural Networks based Identification for Control

KRISTINA VASSILJEVA

TUT PRESS

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

Dissertation was accepted for the defence of the degree of Doctor of Philosophy in computer and system engineering on May 8, 2012

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

Opponents: Yevgeniy Bodyanskiy, Prof., D.Sc., Head of Control System Research Laboratory, Kharkiv National University of Radioelectronics

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

Defence of the thesis: June 21, 2012

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 academic degree.

/Kristina Vassiljeva/

European Union
European Social Fund

Investing in your future

ISSN 1406-4731
ISBN 978-9949-23-294-9 (publ)
ISBN 978-9949-23-295-6 (PDF)

INFORMAATIKA JA SÜSTEEMITEHNIKA C74

Piiratud ühenduvusega tehisnärvivõrkudel põhinev identifitseerimine juhtimiseks

KRISTINA VASSILJEVA

TTÜ
KIRJASTUS

Contents

List of Abbreviations

ANARX	Additive NARX
ARMAX	AutoRegressive Moving Average with eXogenous inputs
CSTR	Continuously Stirred Tank Reactor
FFNN	FeedForward Neural Network
GA	Genetic Algorithm
LLS	Liquid Level System
LM	Levenberg-Marquardt algorithm
MIMO	Multi-Input Multi-Output
MSE	Mean Square Error
NARX	Nonlinear AutoRegressive with eXogenous inputs
NN	Neural Network
NN-ANARX	Neural Networks based Additive NARX
NN-NARX	Neural Network based NARX
NN-SANARX	Neural Network based Simplified ANARX
ODACF	OmniDirectional AutoCorrelation Function
ODCCF	OmniDirectional Cross-Correlation Function
SISO	Single-Input Single-Output
TITO	Two-Input Two-Output

Chapter 1

Introduction

> The beginning is half of every action.
>
> Greek

1.1 General overview

In a field of system engineering process control is a traditional area which is of great practical importance. Control usually involves methods from various fields: dynamic modeling, identification, etc. Thus, to analyze and design nonlinear and dynamical systems we need to absorb and digest a wide range of nonlinear analysis tools.

Usually when we deal with control of strong nonlinear systems a model-based controllers are applied, where a detailed dynamic process model is used in an optimization framework. To develop a nonlinear controller with reasonable complexity a usage of heuristic black-box type control approaches like neural networks, genetic algorithms, fuzzy logic or a combination thereof can significantly help. Therefore, any work in the area of nonlinear process control should be based on an interdisciplinary approach that integrates the results and techniques of process systems engineering with nonlinear systems and control theory [25], [31].

Fundamental control tasks, which include above mentioned techniques, are all functional approximation tasks. Thus, as neural networks have a theoretical capability to approximate any continuous nonlinear function and ability to process many inputs and outputs, then they are applicable for identification and control. So, neurocontrol has a capacity to optimally solve nonlinear control problems directly from sampled data [26].

On the other hand, majority of the techniques for analysis, modeling and control design are based on a classical state-space representation. Therefore, one of the problems discussed in this thesis is to establish a relationship between neurocontrol approaches and the classical control theory. A good example of this kind is an identification of nonlinear SISO and MIMO systems with the neural network based ANARX (Additive Nonlinear AutoRegressive eXogenious) structure. Using that structure allows one to obtain the minimum state-space representation directly in case of SISO systems. On the other hand, in case of the MIMO systems it is not so trivial. Some changes should be applied to the architecture of the neural network based ANARX structure. One of the possibilities to reach the goal is presented in the current thesis.

Working with neural networks one of the main problems one can face the necessity for the training of networks with different number of the parameters. This trial-and-error process should be done due to the need of finding the optimal neural network. Author's research has shown that properly chosen structure of the neural network can significantly improve quality of the control. Thus, for the both structural and parametrical identification of the neural network with restricted connectivity genetic algorithms can be used.

Combinations of genetic algorithms and neural networks have been supportive and collaborative. Supportive combinations typically using one of these techniques to prepare data for consumption by other. Collaborative combinations involve using genetic algorithms to determine neural networks topologies or weights or both at the same time [60, 77].

One particularly significant problem when using genetic algorithms for neural networks is called competing conventions or the permutation problem. Genetic algorithm operates on chromosomes or genotypes. To evaluate obtained genotype one should map into solution of the task, called phenotype. If this mapping is many-to-one then different genotypes map into the same phenotypes even if their genotypes are quite different. In other words, this is where a system of encoding may provide several different ways of encoding networks that exhibit identical functionality. This result is undesirable, as offspring produced by the crossover between these two genotypes will lose functionality of their parents and only show the performance degradation. Consequently, it is a problem researchers do their best to avoid when designing an encoding scheme [6, 24, 56, 77]. Modified structure of the neural network based on ANARX model very clearly defines connections between the different neurons of the network. In addition, proposed by the author chromosome encoding method together with the features of ANARX structure provide one-to-one mapping between obtained chromosome and the corresponding phenotype.

1.2 Author's Contribution

This thesis summaries research experience and the main results achieved by the author in the framework of neural networks based system identification for control. The main attention is paid to ANARX structure of the model.

Set of test objects is presented as SISO and MIMO models with different types of nonlinearities, levels of complexity and orders. Thus, the chosen systems are different models of real plants from the field of process control, as well as academic examples used in the theory of control systems. A more detailed description and references can be found later in this thesis.

The thesis considers

- analysis and design of specific neural network structures to apply them to classical control techniques
- application of genetic algorithms for finding the reduced models of the SISO and MIMO systems based on the neural networks with restricted connectivity
- application of state feedback linearization algorithm to control of nonlinear SISO and MIMO systems

The main original contributions of this thesis are

- design minimal state-space representation of MIMO system based on the parameters of the Neural Network of specific structure (see chapter 5)
- application of NN-ANARX model based dynamic state feedback linearization algorithm to control of nonlinear systems (see chapter 4)
- design of the specific topology of the Neural Networks that easily can applicable with genetic algorithm (see section 6.3)
- development of a technique for obtaining the optimal model for control of nonlinear SISO and MIMO systems (see section 6.6 and Chapter 7)

Author's contributions are discussed in more detail at the beginning of the chapters 4, 5 and 6.

1.3 Outline

The thesis is organized as follows. Part I gives a short overview of the mathematical preliminaries and tools considered later. It discusses motivation and problems, presents different models and basic principles and concepts of neural networks, genetic algorithms and feedback linearization.

Part II is devoted to state-space control identified by ANARX or NN-ANARX structure. These chapters also problems arisen in obtaining the minimal state-space representation of the MIMO system based on neural networks and their possible solutions. Results of this part were presented [72] and [71].

Part III discusses problems of structural identification. The main contribution of this part is in the proposed neural network structure modification and its applicability with genetic algorithms. These chapters also present genetic algorithms with different evaluation functions to obtain reduced model for control of nonlinear systems. Results obtained in this part were published in [49, 68, 69, 70].

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

1.4 Acknowledgements

First I express my gratitude to my supervisor professor Ennu Rüstern who introduced me to the topic of automatic control and guarded me though all the difficulties of my doctoral studies. Without his strong support and freedom for fulfillment of my scientific research interests and excellent working conditions this work would not have been possible.

Very special thanks to my parents and relatives for their strong support and belief in me. I am also very grateful to my daughter for inspiration and giving me energy to pursue my goals.

I would like to thank Dr. Eduard Petlenkov and my colleague Juri Belikov for very fruitful scientific collaborations and discussions.

I thank Dr. Sven Nõmm for very useful discussions and comments.

This work was partially supported by the Estonian Science Foundation Grant no. 8738 and the Estonian Doctoral School in Information and Communication Technology.

Part I

Overview of Mathematical Tools

Chapter 2

System Identification

2.1 Introduction

Constructing models from observed data is a fundamental element in science. In the control area the techniques are known under the term System Identification.

Definition 1 *The determination on the basis of input and output, of a system (model) within a specified class systems (models), to which the system under test is equivalent (in terms of a criterion)[82].*

The significance and difficulty of estimating nonlinear systems is widely recognized. System Identification theory was developed around 1960 based on induction of the state-space representation by Kalman and Bartram for model-based control. Further Åström and Bohlin introduced the AutoRegressive Moving Average with eXogenous inputs (*here and after ARMAX*) model. That led to the predominance of using the method based on prediction error identification.

When as such identification is not the goal, so it considered as a design problem such, that the estimated model is used for specific purpose. For example, one of the main motivations for model building are control applications based on model-based control design operating in closed loop [73].

2.2 Identification for Control

From the control point of view system identification is an exercise in estimating the best possible approximate model within the set of competing representations, rather than a search for the "true" model. Exact model of the system is optimal for all kind of applications, but on the other hand in most real life cases we deal with approximation of the "true" system, thus the quality of the model should depend on the destined application. The reasons for the development of the goal-oriented identification are: high performance control can often be achieved by simple models in condition if basic dynamical features are reflected accurately; identification for control led to iterative model and controller tuning tools that were intuitive and easy implement.

If design of a controller is based on the identified model, then what really matters is the performance achieved by this model-based controller on the "true" system, rather than the distinctive quality of the model. Thus, the fit of the model for controller design depends both on controller which will be used in current application, and on the plant/model mismatch. In practice, the true system is unknown, the model is unknown at the identification design stage, and the controller that will be implemented is unknown as it depends on that model. What is commonly known during identification for control is the control performance goal, in addition some prior knowledge about the process/plant may be available. Thus, in automation community came to the conclusion that the identification criterion should be a function of the control performance criterion. This led to the point, that for most control performance objectives, identification should be performed in closed loop [16, 17, 73].

Thus, the control-oriented identification problem can be formulated as follows

Based on a given control performance goal, design the identification in such way that the performance achieved by the model-based controller on the process is as high as possible [17].

The system identification problem can be divided into number of subproblems: input data selection, (possible) feedback configuration, data length, model structure selection, identification criterion, validation criterion. Once experiment design issues are settled, arises the next problem - choice of the model structure. This is a crucial step in the identification process, therefore this step must be done with care [45]. One of the possibilities are shown in chapter 6.

2.3 Plant Identification by Neural Networks

Sometimes the use of analytical approach is complicated or impossible in principle, then the numerical methods may solve the task. Identification by neural networks has many similarities with classical identification. Frequently, model structures are directly inspired by the classical algorithms. On the other hand, generality of the structure is very important in case of identification based on neural networks. The main difference lies on the fact that the results of this method is almost entirely derived from the data, rather than the knowledge needed to design structure for a specific problem. As data for identification is collected with the limited accuracy (from the available sensors), the restrictions imposed to the structure are not known. Thus, a critical part of the identification is to find which terms should be included in the model. In general, engineers need to find as simple as possible model with good performance that fits the control criteria, see [26], [29],[44] and [45].

There are two types of control: direct and indirect control. In the case of direct control the neural network controller is implemented in control loop in such a way, that tries identify mappings between the reference signal and the output of the system. In other words parameters of the controller are directly adjusted to minimize the error of the control. in case of indirect control first of all parameters of the model of the plant are estimated, further the parameters of the controller are estimated based on both the model of the plant and the actual plant output[44] and [78]. In this thesis the second approach is considered.

Chapter 3

Mathematical models

Further our work will be based upon mathematical models of a process. These models can be constructed from a physical and chemical nature processes or can be abstract. The study of dynamical properties of processes as well as whole control systems give rise to need to look for effective means of differential and difference equation solutions [41].

3.1 State-space models

Introduction

Investigation of processes as dynamical systems is based on theoretical state-space balance equations. If a model of the process is described by the state-space equations, we speak about state-space representation [41]. Many books largely devoted to the subject and the continuing emphasis the evidence of the importance of the state-space approach. The key factor is that in spite of the transfer functions describing the relation between the inputs and outputs of the system, such models are not always suitable for some dynamic analysis and control applications. On the other hand, state models are often the basis of feedback design and stability analysis because their implementation for many systems arises more naturally from governing laws and reveals the behavior inside the system: describes dynamic numerical relations between system inputs, outputs and state variables in the time domain. Of cause, generally state-space variables can also be abstract. Also transfer functions do not lay bare the behavior inside the system: unobservable unstable modes. In addition to this the state-space formulation seems to be the most elegant way of dealing with generalizations like nonlinear Multi-Input Multi-Output (*here and after MIMO*) systems. Also this formulation can easily be extended to the time-varying case. As a result, the general methodologies of system analysis and design using state-space models can be applied to a wide variety of the problems

[14, 58, 67].

SISO State-space systems

The nonlinear Single-Input Single-Output (*here and after SISO*) state-space models can be modeled by a finite number of coupled first-order ordinary differential equations

$$\begin{aligned} \dot{x_1}(t) &= f_1(x_1(t), \ldots, x_n(t), u(t)) \\ \dot{x_2}(t) &= f_2(x_1(t), \ldots, x_n(t), u(t)) \\ &\vdots \\ \dot{x_n}(t) &= f_n(x_1(t), \ldots, x_n(t), u(t)) \\ y(t) &= h(x_1(t), \ldots, x_n(t), u(t)) \end{aligned} \tag{3.1.1}$$

in the continuous-time case, or by a finite number of first-order difference equations

$$\begin{aligned} x_1(k+1) &= f_1(x_1(k), \ldots, x_n(k), u(k)) \\ x_2(k+1) &= f_2(x_1(k), \ldots, x_n(k), u(k)) \\ &\vdots \\ x_n(k+1) &= f_n(x_1(k), \ldots, x_n(k), u(k)) \\ y(k) &= h(x_1(k), \ldots, x_n(k), u(k)) \end{aligned} \tag{3.1.2}$$

in the discrete-time case. In both cases $u \in I\!R$ is the scalar input variable, $x \in I\!R^n$ is the n-dimensional state vector, $y \in Y$ is the scalar output variable, n is a nonnegative integer, f and h are real analytic functions defined on $I\!R^n \times I\!R$ and $I\!R^n$ respectively.

Throughout the work we consider analysis and development of the control applications based on neural network models. Since all models based on neural networks are discrete-time models, then only discrete-time models will be considered further.

MIMO State-Space systems

A nonlinear MIMO control system can be described by the sate equations

$$\begin{aligned} x_1(k+1) &= f_1(x(k), u(k)) \\ x_2(k+1) &= f_2(x(k), u(k)) \\ &\vdots \\ x_n(k+1) &= f_n(x(k), u(k)) \\ y(k) &= h(x(k), u(k)) \end{aligned} \tag{3.1.3}$$

where $x(k) \in X \subset I\!R^n$, $u(k) \in U \subset I\!R^m$, $y(k) \in Y \subset I\!R^m$ and the maps f and h are analytic functions of their arguments.

3.2 Input-Output models

MIMO systems in the input-output form are described by the set of higher order difference equations.

NARX models

All input-output models considered here belong to the class of Nonlinear AutoRegressive with eXogenous inputs (*here and after NARX*) type systems. This structure models the input-output relationship as nonlinear difference equation of the following form

$$y_i(k+n_i) = f_i(y_\alpha(k), \dots, y_\alpha(k+n_{i\alpha}-1), u_\beta(k), \dots, u_\beta(k+\theta_{i\beta}), \\ \alpha, \beta = 1, \dots, m), \qquad i = 1, \dots, m \quad (3.2.4)$$

where $u = (u_1, \dots, u_m) \in U \subset I\!R^m$ is an input variable, $y = (y_1, \dots, y_m) \in Y \subset I\!R^m$ ia an output variable and f_i are real analytic functions. We assume that the system is strictly proper, i.e. $\theta_{i\beta} < n_i$, for $i, \beta = 1, \dots, m$, $n_{i\alpha} < \min(n_i, n_\alpha)$ and $n := n_1 + \dots + n_m$ is the order of the system.

Such class of models is suitable for modeling both the stochastic and deterministic components of a system and capable of describing a wide variety of nonlinear systems [8, 59, 38]. Unfortunately, despite the high accuracy of such kind of models, this class has some drawbacks. First of all, it is not always realizable in classical state-space form, secondly, it is not always linearizable by feedback. Thus, to obtain the above mentioned properties we could use the Additive NARX (*here and after ANARX*) models proposed in [11] and [37].

ANARX models

The main structural property of the ANARX model is that it has all time instances separated.

$$y(k) = f_1(y(k-1), u(k-1)) + \dots + f_n(y(k-n), u(k-n)) = \\ = \sum_{i=1}^{n} f_i(y(k-i), u(k-i)). \quad (3.2.5)$$

That gives the main advantage over the original NARX model: the ANARX structure is always realizable in the classical state-space form without any additional calculations [37]

$$\begin{array}{rcl} x_1(k+1) & = & x_2(k) + f_1(x_1(k), u(k)) \\ x_2(k+1) & = & x_3(k) + f_2(x_1(k), u(k)) \\ & \vdots & \\ x_{n-1}(k+1) & = & x_n(k) + f_{n-1}(x_1(k), u(k)) \\ x_n(k+1) & = & f_n(x_1(k), u(k)) \\ y(k) & = & x_1(k) \end{array} \tag{3.2.6}$$

where n corresponds to the order of the system. The main feature of this representation that it is given in the minimal form: it is accessible and observable. All that provide possibility to use ANARX structure for state-space control of a wide class of nonlinear systems.

3.3 Linearization via State Feedback

Feedback

Feedback is a key notion in the control theory. Most of the controllers (but not all of them) use state or output feedback to calculate a control signal. Feedback is usually used to reduce some uncertainties of the control system and it is the only tool to stabilize an unstable systems. There are different kinds of feedback. For example, static feedbacks make instant relations between the output (or state) and input variables, while dynamic feedbacks bring additional dynamics into the control loop. Output feedbacks use only output information to generate input, while state feedbacks process the whole state vector [25].

Linearization

The aim of the linearization is to apply suitable nonlinear static state feedback to a nonlinear system in order to obtain a linear one in the new coordinates and between the original output and the newly introduced input [25].

Thus, after linearization any controller design method, applicable for linear systems, can be used for control. This technique is a basic for nonlinear control but it is limited due to restrictions imposed by relative degree.

Definition 2 *System (3.1.3) is linearizable by a state coordinate change, if there exists a smooth diffeomorphism $T : X \to X$ which transforms given system to a reachable linear system, in the variable $\zeta = T(x)$:*

$$\zeta(k+1) = A\zeta(k) + Bu(k), \qquad \zeta \in X.$$

Definition 3 *System (3.1.3) is a static-feedback linearizable, if there exists a smooth map $\gamma : X \times U \to U$ such that feedback $u = \gamma(x, v)$ results in a closed-loop system*

$$x(k+1) = f(x(k), \gamma(x(k), v(k))), \qquad x(k) \in X, v(k) \in U$$

which is linearizable by a state coordinate change.

Dynamic state feedback amounts to the use of a controller with dynamics

$$z(k+1) = g(x(k), z(k), v(k)), \qquad z \in X_c \subset I\!R^n, v \in U \tag{3.3.7}$$

and smooth map $h : X \times X_c \times U \to U$, which is combined with system (3.1.3) yields the closed-loop system with extended state-space $X \times X_c$

$$\begin{array}{rcl} x(k+1) & = & f(x(k), h(x(k), z(k), v(k))) \\ z(k+1) & = & g(x(k), z(k), v(k)). \end{array} \tag{3.3.8}$$

Definition 4 *System (3.1.3) is dynamic-feedback linearizable, if there exists a smooth dynamic feedback (3.3.7) which yields a closed-loop system (3.3.8) that is linearizable by a state coordinate change.*

3.4 Artificial Neural Networks

The contributions of present work are made in the domain of control and identification of nonlinear systems. The reason to use the neural networks in principle makes it unnecessary to spend much effort on system modeling in cases where such modeling is difficult. In neurocontrol the unknown nonlinear system dynamics are approximated by linearly or nonlinearly parameterized multilayer neural networks [73].

Introduction

An artificial neural network, usually called Neural Network (*here and after NN*), is a mathematical model for information processing based on a connectionist approach to computation which is generally has the following features:

- a set of processing units (neurons) where each has a certain activation level, which is equivalent to the output of the unit;
- weighted interconnections between various processing units which determine how the activation of one unit leads to input for another unit;
- as activation rule which acts on the set of input signals at a unit to produce a new output signal or activation;
- optionally, a learning rule that specifies how to adjust the weights for a given input/output pair.

There are different types of activation functions: threshold, piecewise-linear, sigmoid. Since in this thesis the sigmoid activation functions are used, then we consider it in detail.

Definition 5 *A C^k-sigmoid function $\sigma : I\!R \to I\!R$ is a nonconstant, bounded, and monotone increasing function of class C^k (continuously differentiable up to order k) [27].*

In other words it is a smooth nonlinearity with saturation range from 0 to $+1$ or from -1 to $+1$.

Two or more neurons can be combined in a layer, and a particular network could contain one or more such layers. The layers of multilayer network play different roles: input, output and all others layers called hidden layers.

The above mentioned structure can be organized differently from the signal flow point of view. If input and intermediate signals are always propagated forward the system is called static or *feedforward network.* Feedforward networks are widely used in pattern recognition and approximation applications. In dynamic or *recurrent networks*, the output depends not only on the current input of the network, but also on the current or previous inputs, outputs or states of the network. The signals are reused, thus recurrent networks can use their internal memory to process arbitrary sequences of inputs [27, 29].

Approximation with Feedforward Networks

The problem of feedforward neural modelling can be formulated as follows: find a representation of continuous mapping $f : \mathbf{K} \to I\!R^k$ by means of known functions and finite number of real parameters, such that the representation yields the uniform approximation of f over K. Where K is uncountable compact subset of $I\!R^{kn+rm}$, k and r are the number of outputs and inputs correspondingly modeled by NN, m and n are input and output orders correspondingly. Thus, mathematically, this is an approximation problem: a possibility to present f by some standard functions with an arbitrary accuracy. It should be mentioned, that in our case f is given in the next form (U_k, Y_k); hence we only have to interpolate the continuum $f(K)$ from the samples (U_k, Y_k) [29].

Thus, according to the Stone-Weierstrass theorem sigmoid functions are suitable for uniform approximation of an arbitrary continuous mapping. In order to present this theorem we should set some definitions before.

Definition 6 *A set A of functions from $K \subset I\!R^{kn+rm}$ to $I\!R$ is called an algebra of functions iff $\forall f, g \in A$ and $\forall \gamma \in I\!R$*

1. $f + g \in A$;
2. $fg \in A$;
3. $\gamma f \in A$.

Definition 7 *Let B be the set of all functions which are limits of uniformly convergent sequences with terms in A, a set of functions from $K \subset I\!R^{kn+rm}$ to $I\!R$. Then B is called the uniform closure of A.*

Definition 8 *A set A of functions from $K \subset I\!R^{kn+rm}$ to $I\!R$ is said to separate points on K iff $\forall x_1, x_2 \in K$ $x_1 \neq x_2 \Rightarrow \exists f \in A, f(x_1) \neq f(x_2)$.*

Definition 9 *Let A be a set of functions from $K \subset I\!R^{kn+rm}$ to $I\!R$. We say that A vanishes at no point of K iff $\forall x \in K \exists f \in A$, such that $f(x) \neq 0$.*

Theorem 1 (Stone-Weierstrass) *Let A be an algebra of some continuous functions from a compact $K \subset I\!R^{kn+rm}$ to $I\!R$, such that A separates points on K and vanishes at no point of K. Then uniform closure B of A consists of all continuous functions from K to $I\!R$.*

Therefore the theorem is a criterion which given functions have to satisfy in order to demonstrate approximation capabilities.

Theorem 2 (G. Cybenko) *Let ϕ be any continuous sigmoidal function. Then finite sums of the form*

$$G(x) = \sum_{j=1}^{N} \alpha_j \phi(w_j^T x + \theta_j)$$

are dense in $C(I_n)$. In other words, given any $f \in C(I_n)$ and $\varepsilon > 0$, there is a sum, $G(x)$, of the above form, for which

$$|G(x) - f(x)| < \varepsilon \qquad for\ all \quad x \in I_n.$$

Where I_n denotes the n-dimentional unit cube, $[0, 1]^n$, the space of continuous functions on I_n is denoted by $C(I_n)$. Theorems 1 and 2 show that neural networks with one hidden layer and an arbitrary continuous sigmoidal function can approximate continuous functions with arbitrary accuracy. In [12] it is proven that no constraints are placed on the number of neurons or the size of the weights.

Feedforward Neural Networks with External Feedback

Control is concerned with dynamic systems. Thus, besides inputs and outputs dynamic system needs some states to characterize the behavior of the controlled system. This leads to the idea of using a feedback on the structure of the NN. In this work a standard FeedForward Neural Network (*here and after FFNN*) is taken and external feedback is implemented to it.

External feedback is sufficient to represent all dynamical systems [26].

The next structure of neural network is obtained as represented in figure 3.1.

Models of nonlinear systems based on that structure are called Neural Network based NARX models (*here and after NN-NARX*). Typical SISO NN-NARX model is given by

$$y(k+n) = \sum_{i=1}^{l} c_i \phi(w_{i,1} y(k) + \cdots + w_{i,n} y(k+n-1) + \\ + w_{i,n+1} u(k) + \cdots + w_{i,n+m} u(k+m-1)), \quad (3.4.9)$$

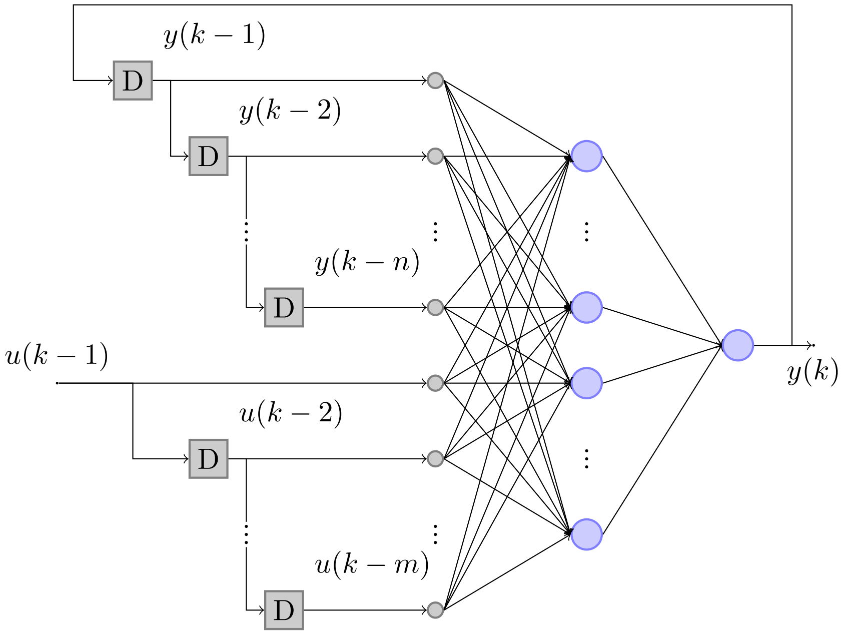

Figure 3.1: Representation of a dynamic model by FFNN.

where $u \in I\!R$ is a real-valued scalar input, $y \in I\!R$ is a real-valued scalar output, $\phi(\cdot)$ is a saturation-type smooth nonlinear function, l is the number of hidden neurons and c_i, w_i are synaptic weights.

3.5 Genetic Algorithms

Introduction

Part of this work is devoted to the system identification problem from the standpoint of control system design. The neural networks of specific structure are used to obtain a suitable model for future controller design. Correct choice of the NN-model improves the control quality of the nonlinear processes. Hence, for the best model identification a set of neural networks must be trained. Moreover, the main problem to obtain a good model using neural networks is to find its optimal structure. These two problems can be solved simultaneously using Genetic Algorithm (*here and after GA*). This leads to the point, that structure of the neural network could be defined by the genetic algorithm, which finds optimal NN parameters and dependencies between the inputs of dynamic model and outputs of the controlled system.

Basic steps of Genetic Algorithm

GAs are numerical optimization algorithms inspired by evolution. GAs are modeled on the principles of natural genetic systems, where genetic information or potential solution is encoded in structures called chromosomes. An implementation of a genetic algorithm begins with a population of chromosomes. Each individual or chromosome has an associated fitness value, which indicates its degree of goodness with respect to the solution it represents. GAs search from a set of points, called a population. Various biologically inspired operators like selection, crossover and mutation are applied on the chromosomes in the population to yield potentially better solutions. And only those individuals in a population who are better suited to the environment can survive and generate offspring [1, 62, 75].

The studies made in present work are based on the Canonical Genetic Algorithm. The first step of the genetic algorithm implementation is a generation of an initial population, the size of which may be constant or may vary from generation to another. In the canonical GA a binary string of the finite length, which refers to a coded possible solution, is a member of population called chromosome. The advantages of the binary representation lie in its simplicity and generality. It is straightforward to apply classical crossover and mutation to binary strings. On the next step each string is evaluated and a fitness value assigned to it. The frequently used operators are selection, crossover and mutation. Selection is applied to the *current population* in order to create the *intermediate* one. To create the next population several operations should be applied. First of all, the recombination of parts of the selected chromosomes (called crossover) is needed to derive the offspring from the intermediate population. After that a mutation takes place. The process of going from the current population to the next one forms one generation. The schematic diagram of the GA is shown in figure 3.2.

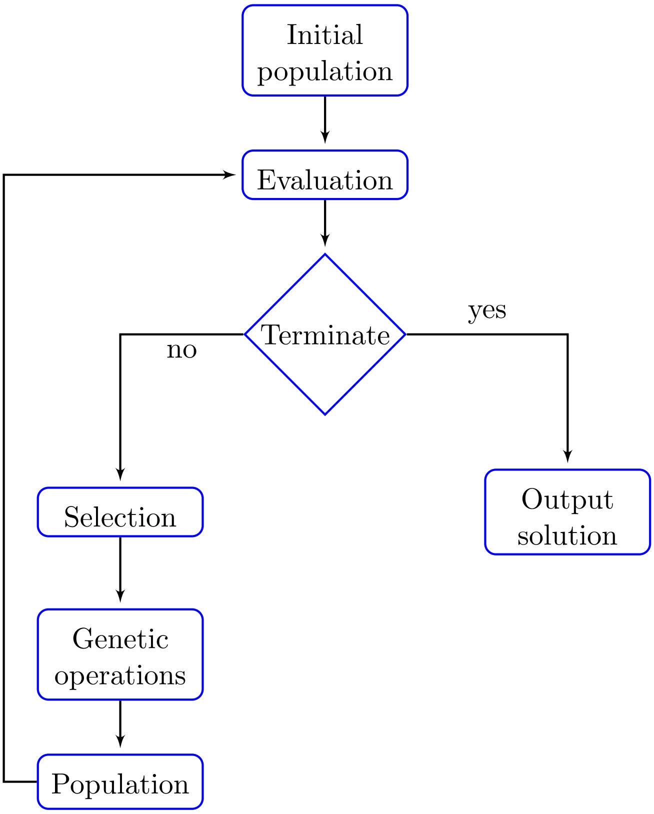

Figure 3.2: Basic steps of GA

There are several parameters that should be tuned in GA. These are the population size, the length of a chromosome, the probabilities of performing crossover and mutation, the termination criteria, and the population replacement strategy. These parameters are problem dependent and no guidelines for their choice could be made.

Part II

State-Space Representation and Feedback Control

Chapter 4

State-space control of Nonlinear Systems Identified by ANARX and Neural Network based SANARX Models

Control is concerned with dynamical systems, which means that states of an identified model depend on both the inputs and the outputs of the system. For the modeling the most commonly used structure is NARX, which is based on determination of the regressors (see Chapter 3.2 and 3.4). Also during the resent years subclass of NARX architectures NN-based ANARX type models have shown their applicability to a wide range of problems, such as rear-motion modeling of truck-trailer [3], electric generator [50], heater, some chemical processes [54] and modeling of surgeon hand movements during surgery [48].

From the theoretical point of view NARX models may better represent real world systems. However, experience has shown that in many cases the performance of NN-based ANARX sructure models is close enough to NN-based NARX model of the same order. A more detailed comparison of the performance of these two models can be found in [47] and [54].

In conclusion we can say that the majority of the general class of mechanical systems described by the following equation (4.0.1) can be approximated by NN-based ANARX structure to obtain the state-space representation [79].

$$M(q,\sigma)\ddot{q} + C(q,\dot{q},\sigma)\dot{q} + G(q,\dot{q},\sigma) = u, \tag{4.0.1}$$

where $q(t) \in I\!R^m$ is the vector of generalized coordinates, $M(q, \sigma)$ is inertia matrix, the term $G(q, \dot{q}, \sigma)$ represents all external generalized forces and term $C(q, \dot{q}, \sigma)$ depends on inertia matrix $M(q, \sigma)$.

4.1 Author's contribution

The main contribution of this chapter is devoted to the implementation of the state-space based algorithm for control of nonlinear SISO and MIMO systems.

- Capabilities of the ANARX and NN based ANARX structure for identification of the nonlinear systems to obtain a state-space representation are demonstrated;

- All the necessary constraints on the model for implementation of the proposed control technique are listed;

- A state-feedback controller is presented for a class of nonlinear systems identified by ANARX or NN-SANARX models;

- The effectiveness on the proposed method is shown on examples;

- Considered technique is compared with other works.

4.2 ANARX

For the most real-world nonlinear systems it is very difficult to find the state-space representation directly from the identification procedures. According to this, most often high order systems written by nonlinear input-output difference equations, obtained on the basis of sampled experimental data. In many cases arbitrary structured NARX (see section 3.2) model does not necessarily have a state-space realization. Nevertheless, practically all existing control theory for nonlinear systems are based on a state-space description. Thus, in [37] and [55] a class of NARX models was presented that always admit classical state-space realization. Hereinafter, speaking about realizability, we mean finding the minimal: accessible and observable realization. Remarkable property of ANARX model that it is always linearizable by dynamic output feedback.

In many situations the ANARX model is obtained from experimental data using the identification procedures or neural networks. Let the ANARX model of the controlled system be given by the state-space representation (3.2.6). If dynamics of the system on the previous time steps are unknown the initial states of the model can be found as follows

$$\begin{cases} x_n(0) = f_n(y_0, u_0) \\ x_i(0) = x_{i+1}(0) + f_i(y_0, u_0), \quad 1 < i < n \\ x_1(0) = y_0, \end{cases} \tag{4.2.2}$$

then assuming that all the states in the previous time steps are zeros. We suppose $y(0) = y_0$, $u(0) = u_0$ and the mentioned below assumptions are imposed on the system:

Assumption 1 Relative degree of the system $r = 1$

Assumption 2 Order of the controlled system $n \geq 2$.

Relative degree of the discrete-time system is often called the time delay or delay between input and output of the system. In other words it is the number of time steps needed to calculate output of the system from its input.

It comes from the (3.2.6) that $x_1(k) = y(k)$. We formulate a control task as $y(k+1) = \nu(k)$. Thus, it can be linearized by the following state feedback algorithm

$$\begin{cases} x_1(k) = y(k) \\ y(k+1) = \nu(k) \end{cases} \Rightarrow \quad x_1(k+1) = \nu(k), \tag{4.2.3}$$

where $\nu(k)$ is the desired output of the system (reference signal). Control signal can be found by solving the following equations. First, according to (3.2.6) state x_1 can be found as shown further

$$x_1(k+1) = x_2(k) + f_1(x_1(k), u(k)). \tag{4.2.4}$$

Substituting (4.2.3) into the previous equation, we set

$$\nu(k) = x_2(k) + f_1(x_1(k), u(k)) \tag{4.2.5}$$

or

$$\nu(k) - x_2(k) = f_1(x_1(k), u(k)). \tag{4.2.6}$$

From the (4.2.6) it follows that the control signal can be calculated as

$$u(k) = F(y(k), \nu(k), x_2(k)). \tag{4.2.7}$$

Thus, dynamic feedback controller consists of dynamic state-space model (state-space representation of ANARX class input-output model) and static state linearization. The structure of the control system is shown in figure 4.1. In that case assuming that the state-space model is perfect, we can say that according to (4.2.3), the closed loop system is a first order system. The above described control technique was applied to control of nonlinear SISO systems in [72]. Consider the following numerical example.

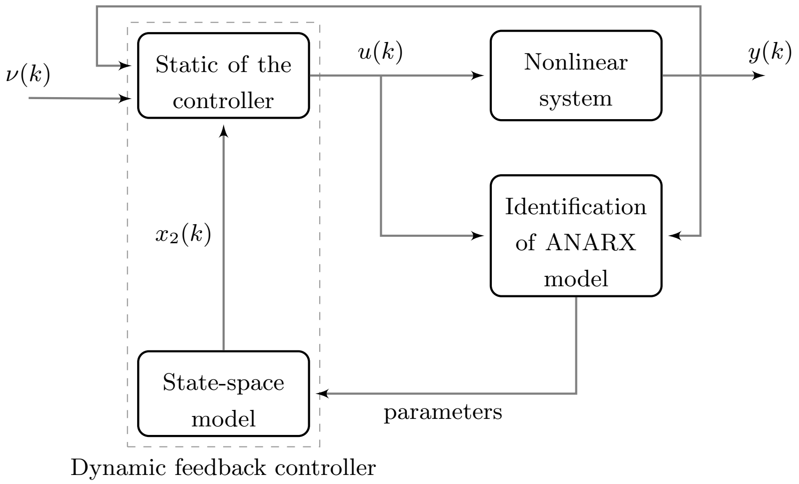

Figure 4.1: Structure of the control system

Numerical example 1

A model of the liquid level system of interconnected tanks [5] is presented by the following input-output equation

$$\begin{aligned} y(k+3) \;=\; & 0.43y(k+2) + 0.681y(k+1) - 0.149y(k) + \\ & +0.396u(k+2) + 0.014u(k+1) - 0.071u(k) - \\ & -0.351y(k+2)u(k+2) - 0.03y^2(k+1) - \\ & -0.135y(k+1)u(k+1) - 0.027y^3(k+1) - \\ & -0.108y^2(k+1)u(k+1) - 0.099u^3(k+1). \end{aligned} \qquad (4.2.8)$$

This third order system is given in the form defined by ANARX structure, so according to [34] and [36] it can be directly realized in the classical state-space form

$$\begin{aligned} x_1(k+1) &= x_2(k) + 0.43x_1(k) + 0.396u(k) - 0.351x_1(k)u(k) \\ x_2(k+1) &= x_3(k) + 0.681x_1(k) + 0.014u(k) - 0.03x_1^2(k) - \\ & \quad -0.135x_1(k)u(k) - 0.27x_1^3(k) - 0.108x_1^2(k)u(k) - \\ & \quad -0.099u^3(k) \\ x_3(k+1) &= -0.149x_1(k) - 0.071u(k) \\ y(k) &= x_1(k) \end{aligned} \qquad (4.2.9)$$

By using state-space representation (4.2.9), ANARX model based dynamic feedback controller can be represented by the following equations

$$u(k) = \frac{\nu(k) - x_2(k) - 0.43y(k)}{0.396 - 0.351y(k)}$$

$$
\begin{aligned}
x_2(k+1) &= x_3(k) + 0.681x_1(k) + 0.014u(k) - 0.03x_1^2(k) - \\
&\quad -0.135x_1(k)u(k) - 0.27x_1^3(k) - \\
&\quad -0.108x_1^2(k)u(k) - 0.099u^3(k) \\
x_3(k+1) &= -0.149x_1(k) - 0.071u(k)
\end{aligned}
$$

According to the initialization algorithm (4.2.2) initial states were obtained as $x(0) = [0 \quad 0 \quad 0]^T$. One of the major advantages of the state-space control is its speed. As you can see in figure 4.2, response of the state-space controller is much faster than response based on the dynamic output feedback algorithm of the same model.

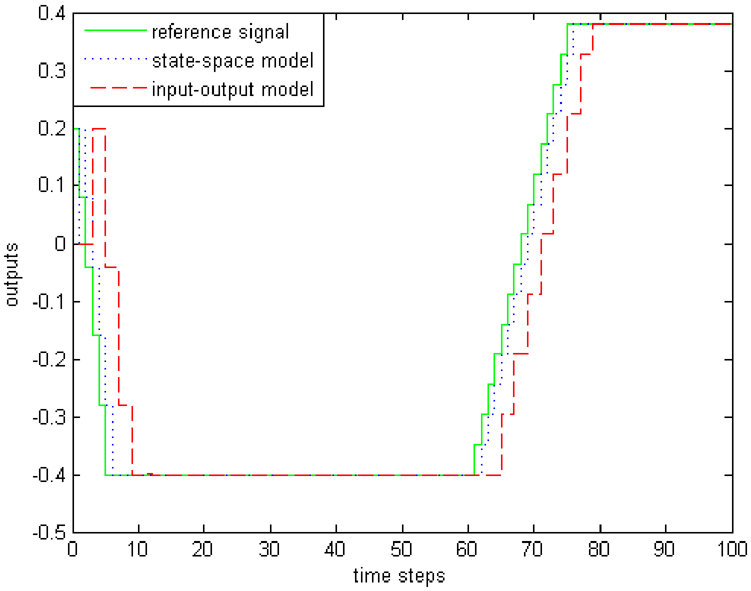

Figure 4.2: Comparison of the two approaches

Closed loop control system was simulated with piece-constant reference signal. The results of this simulation are depicted in figure 4.3. It is easy to see that the output of the model perfectly follows the reference signal.

4.3 NN-ANARX

The realizability problems of ANARX models are caused by the fact that this is very restricted class. In the most cases using classical identification

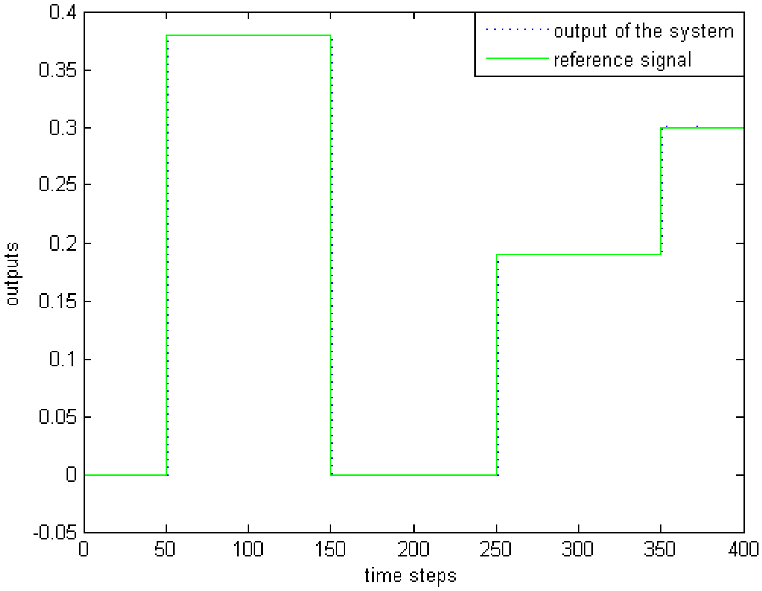

Figure 4.3: Control of the liquid level system interconnected tanks

procedures there is no way to separate (decompose) the different time steps. One of the possible solutions is to use a new subclass of NN-Based models.

This model called Neural Networks based Additive NARX Model (*here and after NN-ANARX*) was shown in [10, 34, 35]. This type neural network has a restricted connectivity: a hidden layer consists of n parallel sub-layers corresponding to the n-th order of the model. Each i-th sub-layer approximates the function f_i from (3.2.5). Thus, model based on neural networks can be formalized in the following form

$$y(k) = \sum_{i=1}^{n} C_i \phi_i(W_i \cdot z(k-i)), \tag{4.3.10}$$

where $\phi_i(\cdot)$ is an activation function of neurons of the corresponding sub-layer, W_i and C_i are matrices of synaptic weights of inputs and outputs of i-th sub-layer and for SISO systems $z(k) = [y(k), u(k)]^T$, see figure 4.4.

From (3.2.5) and (3.2.6) it follows that (4.3.10) also can be written in

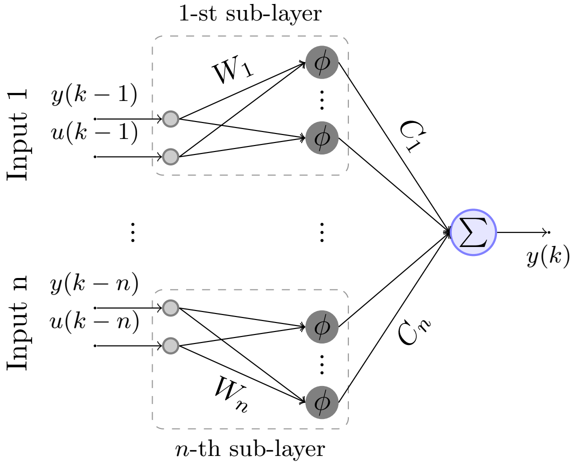

Figure 4.4: Representation of NN-ANARX structure

the state-space form as

$$\begin{array}{rcl} x_1(k+1) & = & x_2(k) + C_1 \cdot \phi_1(W_1 \cdot [x_1(k), u(k)]^T) \\ x_2(k+1) & = & x_3(k) + C_2 \cdot \phi_2(W_2 \cdot [x_1(k), u(k)]^T) \\ & \vdots & \\ x_{n-1}(k+1) & = & x_n(k) + C_{n-1} \cdot \phi_{n-1}(W_{n-1} \cdot [x_1(k), u(k)]^T) \\ x_n(k+1) & = & C_n \cdot \phi_n(W_n \cdot [x_1(k), u(k)]^T) \\ y(k) & = & x_1(k) \end{array} \quad (4.3.11)$$

On the basis of the above mentioned if ANARX model is given in the form of neural network (4.3), equations (4.2.4)-(4.2.6) can be rewritten by using parameters of the neural network

$$x_1(k+1) = x_2(k) + C_1 \cdot \phi_1(W_1 \cdot [x_1(k), u(k)]^T), \quad (4.3.12)$$

$$\nu_1(k) = x_2(k) + C_1 \cdot \phi_1(W_1 \cdot [x_1(k), u(k)]^T), \quad (4.3.13)$$

$$\nu_1(k) - x_2(k) = C_1 \cdot \phi_1(W_1 \cdot [x_1(k), u(k)]^T). \quad (4.3.14)$$

Calculation of the inverse function ϕ_1 of the first sub-layer (4.3.14) is not an easy task. Several techniques for doing this were suggested. First of all, Newton's method (or Newton-Raphson method) can be used for solving the equation or system of equations (4.3.14) with respect to variable(s) $u(k)$ by the methods of calculus, see [3] and [54]. Additionally to previous technique an alternative method was proposed in [52]. Control problem can be solved by imposing one more restriction on NN-ANARX structure, namely the first sub-layer is linear or in other words ϕ_1 is a linear transfer function. Another possibility is to train an additional simple static nonlinear neural network approximating function, which is used to estimate the vectors of controls, see [53] for details.

The simplest and the fastest of them is to use Neural Network based Simplified NN-ANARX structure (*here and after NN-SANARX*). In that subclass of NN-ANARX model [52] first layer is linear, thus function (4.3.10) can be rewritten in the following form

$$y(k) = C_1 \cdot W_1 \cdot z(k-1) + \sum_{i=2}^{n} C_i \cdot \phi_i(W_i \cdot z(k-i)). \tag{4.3.15}$$

Such a restriction guarantees that control $u(k)$ can be easily calculated from the system of linear equations. On the basis of (4.3.15), the need to use a model of the second order or higher is apparent.

Let us now define the following matrix

$$D := C_1 \cdot W_1. \tag{4.3.16}$$

Since $Z(k) = [y(k), u(k)]^T$, then matrix D can be divided into 2 parts $D = [D_1 \quad D_2]$ so that

$$C_1 \cdot W_1 \cdot z_1(k) = D \cdot z(k) = D_1 \cdot y(k) + D_2 \cdot u(k). \tag{4.3.17}$$

From (4.3.17) it follows that control signal can be calculated as

$$u(k) = D_2^{-1}(\nu(k) - x_2(k) - D_1 \cdot y(k)). \tag{4.3.18}$$

Notice, in the case of SISO systems $D \in I\!R^2$ is a 2×1 vector and, as result, $d_1, d_2 \in I\!R$ are real numbers. It is obvious, that described technique can be applied, only if after training of the neural network $d_2 \neq 0$.

Before applying the proposed technique to control of nonlinear MIMO systems, consider the next numerical example.

Numerical example 2

The case under study is the control of reactant concentration in an exothermic Continuously Stirred Tank Reactor (*here and after CSTR*). [32], [51] provide an example of CSTR model given by input-output equation

$$\begin{aligned} y(k+2) \;\; = \;\; & 0.7653y(k+1) - 0.231y(k) + 0.4801u(k+1)- \\ & -0.6407y^2(k+1) + 1.014y(k)y(k+1)- \\ & -0.3921y^2(k+1) + 0.592y(k+1)u(k+1)- \\ & -0.5611y(k)u(k+1). \end{aligned} \tag{4.3.19}$$

As it can be seen, because of the several terms (including the last one) in (4.3.19), the model does not belong to the class of ANARX models. It means that we cannot use the state feedback algorithm (4.2.7) directly.

Consider proposed technique, we treat this as our unknown dynamic system which can be approximated by the NN-SANARX model. To obtain such a model defined by equation (4.3.15), system (4.3.19) was simulated with uniformly distributed random signal. We trained neural network with two sub-layers, corresponding to the second order model ($n = 2$) and with three neurons on sub-layers of the hidden layer. Levenberg-Marquardt (*here and after LM*) algorithm was used to perform the training. The linear activation function was chosen on the first sub-layer, with respect to the NN-SANARX structure, and the hyperbolic tangent sigmoid activation function on the second sub-layer.

Remark 1 *In the application to simplify the calculations biases are proposed to be equal to "0". But this algorithm can be always extended to the case based on neural network with nonzero bias values.*

Identified parameters of the model (4.3.15) have the following values

$$W_1 = \begin{bmatrix} -0.7761 & 0.5588 \\ 0.2931 & 0.4308 \\ 0.2088 & 0.7890 \end{bmatrix},$$

$$W_2 = \begin{bmatrix} 0.0193 & -0.8089 \\ 0.0205 & -0.8086 \\ 0.0406 & 0.6729 \end{bmatrix},$$

$$C_1 = \begin{bmatrix} -0.2261 & 0.0339 & 0.7777 \end{bmatrix},$$
$$C_2 = \begin{bmatrix} 1.4446 & 1.5765 & 3.5949 \end{bmatrix}.$$

According to equation (4.3) now we can write the state-space representation of the model (4.3.19) as follows

$$\begin{aligned} x_1(k+1) &= x_2(k) + C_1 \cdot W_1 \cdot [y(k), u(k)]^T \\ x_2(k+1) &= C_2 \cdot \phi_2(W_2 \cdot [y(k), u(k)]^T) \\ y(k) &= x_1(k). \end{aligned}$$

Using algorithm (4.2.2) for calculation of the initial states, we obtain

$$x(0) = \begin{bmatrix} 0 \\ C_2 \cdot \phi_2(W_2 \cdot [0 \quad 0]^T) \end{bmatrix}.$$

Control signal for nonlinear discrete-time system was calculated by equation (4.3.18) using obtained parameters of the neural network

$$u(k) = \frac{\nu(k) - x_2(k) - 0.3478y(k)}{0.5019}.$$

Closed loop control system was simulated with piece-constant reference signal. The result of the simulation is shown in figure 4.5.

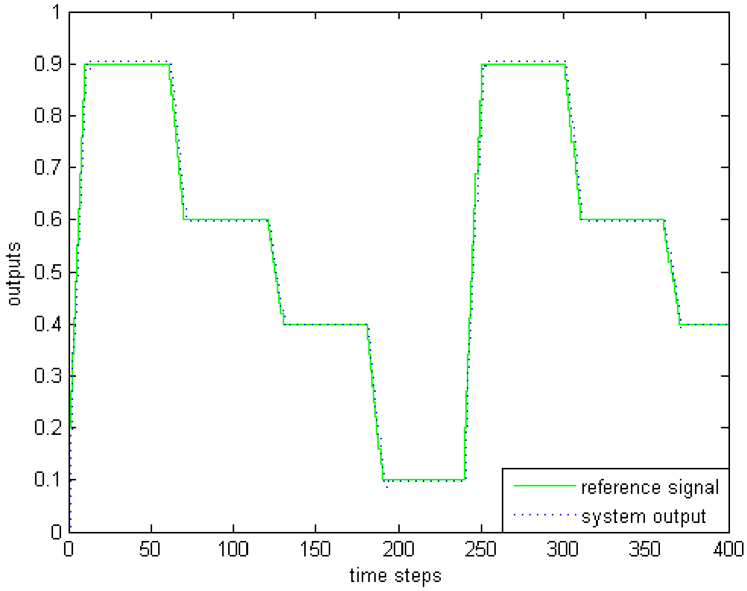

Figure 4.5: Control of the of reactant concentration in an exothermic CSTR

It can be seen that strong restrictions imposed by NN-SANARX structure with one linear sub-layer do not cause drawbacks in quality of identification and control.

Finally, the results in this work were compared to other technique's results. First of all it can be seen that using our algorithm even if concentration value is 0.9, simulations seemed to indicate that the system is stable and has a fast response to a set point change. Unlike the previous works [32] and [51] where concentration value 0.8 was set as a point of boundary of the robust stability region. Also it should be mentioned that good performance of other techniques were obtained for a set point change 0.2 of reactant exit concentration. As it is shown in figure 4.5 even if reference signal changes drastically from 0.1 up to 0.9, controlled system behavior is in stability limits and has a good performance.

4.4 MIMO systems

Above described technique can be applied to the more general class of nonlinear MIMO systems. Thereby in that case $y(k), u(k)$ and $z(k)$ in equation (4.3.15) are the following vectors:

$$u(k) = [u_1(k), \ldots, u_m(k)], \tag{4.4.20}$$

$$y(k) = [y_1(k), \ldots, y_m(k)], \tag{4.4.21}$$

$$z(k) = [y_1(k), \ldots, y_m(k), u_1(k), \ldots, u_m(k)]. \tag{4.4.22}$$

The structure of the corresponding neural network based model is depicted in figure 4.6.

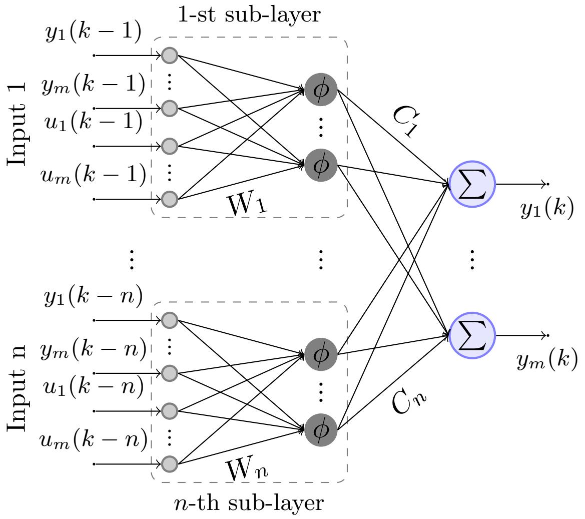

Figure 4.6: Representation of NN-ANARX structure for MIMO systems

$W_i \in I\!R^{l_i \times (m+m)}$ and $C_i \in I\!R^{m \times l_i}$ are input and output matrices of synaptic weights. Assume that number of inputs equals to the number of outputs.

In case of MIMO system the same problem raises during the application of the NN-based ANARX, namely complexity of the calculation of the control signal from the dynamics of the controller.

First of all, Newton's method [3] in case of MIMO systems has several drawbacks: high complexity of applying control and slow convergence speed.

First submethod of Taylor series based approach, see [4], can be used to control very restricted class where a number of inputs, first hidden layer and outputs has to be the same. Also process of finding solutions of each polynomial equation with arbitrary degree of accuracy in some situations may become costly. Second method has also its disadvantages: in case if approximation order of the hidden layer activation function greater than one - solution of the system of polynomial equations becomes an extremely difficult task.

In analytical approach, see [2] the number of inputs, neurons of first hidden layer and outputs also has to be equal, in addition hidden layer activation function has to be invertible.

More preferable techniques are additional static neural network based approach and the NN-based Simplified ANARX method. The main disadvantage of the first technique is an additional time required to train another static neural network. In the second case additional restrictions are imposed: linearity of the fist sub-layer and system should be with equal number of inputs and outputs.

Based on the foregoing, if imposed by the NN-SANARX approach constraints are feasible, then this technique should be used as the most easiest one.

Therefore for this class of nonlinear MIMO systems identified by NN-SANARX model, it follows from definition (4.3.16) that D can be divided into two matrices $D_1 \in I\!R^{m\times m}$ and $D_2 \in I\!R^{m\times m}$. Above mentioned criterion guarantees that D_2 is a square matrix. If matrix D_2 is a nonsingular, system has a unique solution, which can be found as

$$\begin{aligned}[u_1(k),\ldots,u_m(k)]^T = D_2^{-1}([\nu_1(k),\ldots,\nu_m(k)]^T - \\ - C_2\cdot\phi_2(W_2\cdot z(k)) - D_1\cdot[y_1(k),\ldots,y_m(k)])\end{aligned} \tag{4.4.23}$$

Consider the following numerical example of MIMO NN-SANARX structure based identification and control.

Numerical example 3

A nonlinear MIMO discrete-time system [39] and [61] is presented by the following input-output equation

$$\begin{aligned}y_1(k+1) &= 0.4y_1(k) + \frac{u_1(k)}{1+u_1^2(k)} + 0.2u_1^3(k) + 0.5u_2(k)\\ y_2(k+1) &= 0.2y_2(k) + \frac{u_2(k)}{1-u_2^2(k)} + 0.4u_2^3(k) + 0.2u_1(k).\end{aligned} \tag{4.4.24}$$

During the system simulation a set of input-output data was obtained. On the basis of the data a neural network of the following NN-SANARX structure was trained with LM algorithm

$$\begin{aligned} [y_1(k), y_2(k)]^T = C_1 \cdot W_1 \cdot [y_1(k-1), y_2(k-1), u_1(k-1), u_2(k-1)]^T + \\ + C_2 \cdot \phi_2(W_2 \cdot [y_1(k-2), y_2(k-2), u_1(k-2), u_2(k-2)]^T) \end{aligned} \tag{4.4.25}$$

The neural network has two sub-layers corresponding to the second order of the model ($n = 2$). Thus the necessary condition for usage of the NN-SANARX model and design of the controller, as it was mentioned in section 4.2, is satisfied. Activation function ϕ_2 of the second sub-layer with 5 neurons ($l_2 = 5$) was chosen as hyperbolic tangent. The corresponding matrices of synaptic weights were obtained

$$W_1 = \begin{bmatrix} -0.6759 & 2.7452 & 0.1107 & 2.1674 \\ 1.5531 & -0.3622 & 0.9593 & 0.3234 \end{bmatrix},$$

$$W_2 = \begin{bmatrix} 6.8147 & -5.8647 & -4.5813 & 0.6941 \\ 0.0579 & -0.1924 & -0.0200 & -0.0218 \\ -0.2389 & 0.1834 & -0.5898 & -0.0125 \\ -0.0023 & 0.0032 & -0.0038 & -0.0188 \\ 0.2430 & -0.1850 & 0.6676 & 0.0117 \end{bmatrix},$$

$$C_1 = \begin{bmatrix} 0.1150 & 0.7792 \\ 0.3836 & 0.1638 \end{bmatrix},$$

$$C_2 = \begin{bmatrix} -0.0085 & 1.8092 & 7.8487 & 17.7184 & 6.2815 \\ -0.0002 & 1.4234 & 0.0607 & 35.5487 & 0.0668 \end{bmatrix},$$

Now according to the proposed technique state-space representation o f the model should be written down

$$\begin{aligned} x_{11}(k+1) &= x_{21}(k) + C_{11} \cdot W_1 \cdot [y_1(k), y_2(k), u_1(k), u_2(k)]^T \\ x_{12}(k+1) &= x_{22}(k) + C_{12} \cdot W_1 \cdot [y_1(k), y_2(k), u_1(k), u_2(k)]^T \\ x_{21}(k+1) &= C_{21} \cdot \phi_2(W_2 \cdot [y_1(k), y_2(k), u_1(k), u_2(k)]^T) \\ x_{22}(k+1) &= C_{22} \cdot \phi_2(W_2 \cdot [y_1(k), y_2(k), u_1(k), u_2(k)]^T) \\ y_1(k) &= x_{11}(k) \\ y_2(k) &= x_{12}(k) \end{aligned} \tag{4.4.26}$$

Foregoing system of equations (4.4.26) was obtained in such form which is easy to use in control application. Further initial states calculations were made using algorithm (4.2.2)

$$x(0) = \begin{bmatrix} 0 \\ 0 \\ C_{21} \cdot \phi_2(W_2 \cdot [0 \quad 0 \quad 0 \quad 0]^T) \\ C_{22} \cdot \phi_2(W_2 \cdot [0 \quad 0 \quad 0 \quad 0]^T) \end{bmatrix}.$$

By using these parameters and equation (4.4.23) the following controller, based on state-space feedback algorithm, was designed

$$D = \begin{bmatrix} 1.1324 & 0.0334 & 0.7602 & 0.5012 \\ -0.0049 & 0.9938 & 0.1996 & 0.8845 \end{bmatrix},$$

As the deal with Two Input Two Output (*here and after TITO*) system, it follows from (4.3.17) that

$$D_1 = \begin{bmatrix} 1.1324 & 0.0334 \\ -0.0049 & 0.9938 \end{bmatrix}, \qquad D_2 = \begin{bmatrix} 0.7602 & 0.5012 \\ 0.1996 & 0.8845 \end{bmatrix}.$$

D_2 is nonsingular square matrix, hence it can by applied to control signals calculations

$$\begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} = D_2^{-1} \cdot \left(\begin{bmatrix} \nu_1(k) \\ \nu_2(k) \end{bmatrix} - \begin{bmatrix} x_{21}(k) \\ x_{22}(k) \end{bmatrix} - D_1 \cdot \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} \right).$$

This control system was simulated with piece-constant and sinusoidal tracking reference signals $\nu_1(k)$ and $\nu_2(k)$. Closed loop simulation results are presented in the next figures.

It can be see from figures 4.7 and 4.8 that controls $u_1(k)$ and $u_2(k)$ are capable of simultaneous tracking of the desired reference signals $\nu_1(k)$ and $\nu_2(k)$ respectively. Proposed technique can be successfully applied to control nonlinear MIMO systems.

Numerical example 4

Nonlinear discrete-time MIMO system [64] was also chosen to evaluate the effectiveness of the proposed control algorithm. It distinguishes from the previous example (4.4.24) by the second order of the both difference equations.

$$\begin{aligned} y_1(k) = & \frac{0.7y_1(k-1)y_2(k-2)}{1+y_1^2(k-1)+y_2^2(k-2)} + \\ & + 0.3u_1(k-2) + u_1(k-1) + 0.2u_2(k-2) \\ y_2(k) = & \frac{0.5y_2(k-1)\sin(y_2(k-2))}{1+y_2^2(k-1)+y_1^2(k-2)} + \\ & + 0.5u_2(k-2) + u_2(k-1) + 0.2u_1(k-2) \end{aligned} \qquad (4.4.27)$$

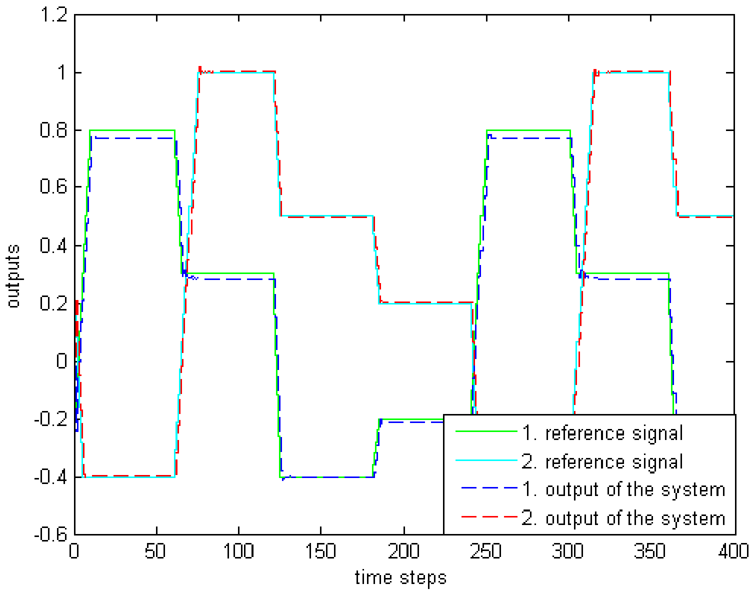

Figure 4.7: Control of the of MIMO system

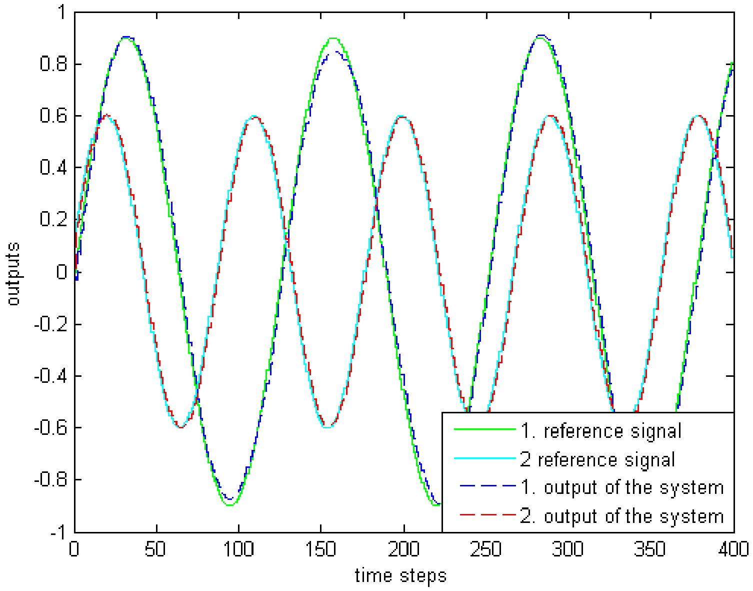

Figure 4.8: Control of the of MIMO system

Neural network having structure shown in figure 4.6 with two sub-layers corresponding to the order of each equation was trained by LM training algorithm. Levenberg-Marquardt algorithm can be used because of the comparatively small number of the parameters caused by the restricted connectivity of the neural network. Minimal state-space form of the model with four states was obtained based on the NN-SANARX structure parameters. Results of the simulation are shown in figure 4.9.

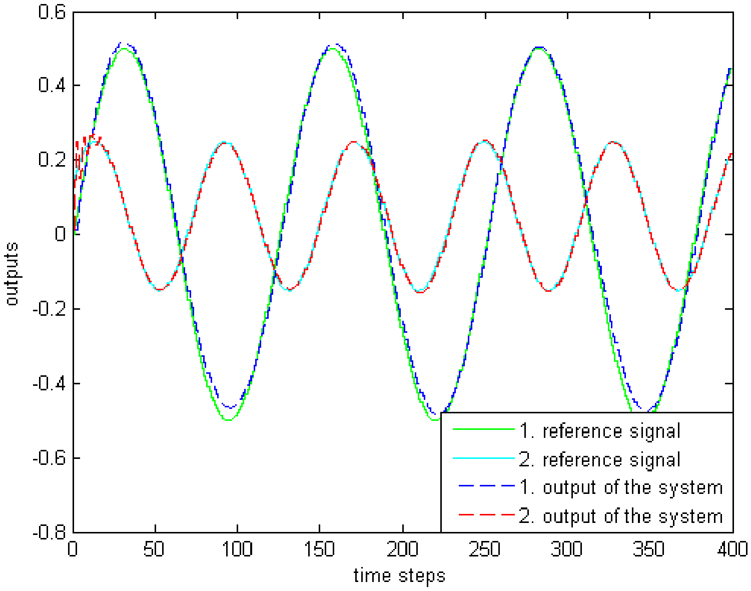

Figure 4.9: Control of the of MIMO system

4.5 Conclusions

In this chapter we have studied applicability of the ANARX or NN-ANARX structure for control by the state feedback linearization technique of nonlinear SISO and MIMO systems. This algorithm is based on the model of the controlled system with relative degree $r = 1$ and order of the model $n \geq 2$. The technique proposed in the chapter can be considered as a combination of a classical state feedback linearization with neural networks based approach.

In general calculations of the control signal for state feedback controller can be done by the following algorithm:

Step 1: If nonlinear system is not given in the form defined by ANARX structure, then approximate it by the NN-based SANARX model, otherwise proceed directly to Step 2.

Step 2: Write state-space representation of the model using (3.2.6) or (4.3) respectively.

Step 3: Calculate initial states using equations (4.2.2).

Step 4: Calculate control signal $u(k)$

- for ANARX structure using equation (4.2.7);
- for NN-SANARX structure using equations (4.3.14) or (4.4.23) in MIMO case.

The control method was demonstrated by means of numerical examples (1)-(3). Our simulations have shown the effectiveness of the proposed technique. First of all, proposed state-based algorithm for control of nonlinear systems works faster than dynamic output feedback control method used in [54], [53], because on each time step we know all the states representing n-th order dynamics of the controlled system and closed-loop system is a first order system. In all cases control systems are capable of tracking the desired reference signals with high accuracy.

Restriction $n \geq 2$ imposed on the model by this method rises the issue of finding the minimal state-space representation especially in the case of MIMO systems. As you can see in the numerical example 3 (4.4.24), the order of each subsystem equals to 1. To comply with the given assumption, system was identified by the second order NN-SANARX structure. This leads to the point that according to the order of the original system itself (orders of the equations of the given system); the obtained state-space representation is not in the minimal form. On the contrary, as each difference equation of the last MIMO system (4.4.27) has the second order, derived sate-space representation appears in the minimal form.

Chapter 5

Neural Network Based Minimal State-Space Representation of Nonlinear MIMO Systems

For the first time problem of finding minimal state-space realization can be dated back to the early 1960s. The minimal state-space realization problem for linear systems was started by Gilbert [18] and by Kalman [30]. The approach of Gilbert was based on partial-fraction expansions and worked under the assumption that each entry of the transfer function matrix has distinct poles. Kalmans algorithm was based on the theory of controllability and observability and reduction of a non-minimal state space realization until it becomes minimal.

Task of finding the minimal state-space representation has attracted much attention of the scientific community and led to the development of the large number of algorithms to solve that problem.

The problem of determining the minimal state-space representation is a fundamental problem for control systems. In order to analyze the system it is advantageous to have its compact description. Hereinafter, speaking about compact form we mean finding a state-space model of minimal size of the given system. It connects to many other topics in realization theory, like controllability and observability properties, similarity invariants, balanced realizations and model reduction.

As the minimal realization of the system is both controllable and observable, then it is a good basis for designing a state feedback controller.

Usually there are two main groups of minimal state-space realization methods:

- methods that starts with non-minimal realization of the system. Further it can be reduced to get the desired representation
- complicated methods that start with impulse response of the system and obtain the minimal realization directly by suitable transformations.

In case of linear systems the theory is quite complete [14]. Recently increased interest in studying of multi-dimensional systems because of various applications in control of multi-pass processes, image processing, etc. Model reduction plays an important role in the analysis and design of multidimensional systems because of large amount of data involved in multidimensional signal processing. However, the general problem of minimal state-space realization of multidimensional systems has not been solved. Obtained methods exist only for some special cases. For example, in [9] the recursive recurrent neural network learning algorithm based on the ordered derivatives has been developed for the parameter learning and minimal model determination. Proposed approach was used for linear discrete-time dynamic system identification.

In analogy with linear systems, some authors define a minimal realization of a nonlinear system as a nonlinear system which is observable, weakly controllable and either analytic or symmetric [14, 58, 66]. The system is weakly controllable if every state in state-space can be reached arbitrarily closely at time t by applying an appropriate control over $[0, t]$; the system is strongly controllable if every state in state-space can be reached exactly at time t. Clearly, strong controllability implies weak controllability.

This chapter considers the minimal representation of the nonlinear MIMO systems.

5.1 Author's contribution

The contribution of this chapter is devoted to the minimal NN-Based state-space representation of the system and its implementation for control of nonlinear MIMO systems.

- Development of the algorithm for elimination of the redundant connections in a neural network between hidden and output layers in order to obtain the minimal state-space representation;
  - finding the regressors of the black-box system using MATLAB System Identification Toolbox

- derivation of the minimal state-space representaion

- Application of the obtained structure to the control algorithm;

- Effectiveness of the algorithm is demonstrated on numerical example.

5.2 Problem statement

The basis of a good identification is finding a proper order of the system. System performance degrades if identified order is less than the true order of the system. On the contrary, if the identified system has the redundant order, that increases the complexity of network computation and some times even the system performance.

As it was pointed out in [72] and section 4.5 the easiest way of finding the minimal representation is that case, when controlled system has the same order of the subsystems. Moreover, that number should be a multiple to the quantity of the system outputs, otherwise solution of the problem is not so trivial [71]. Detailed explanation of this statement is given below.

As illustrative example let be this system

$$y_p(k+n_p) = f_p(\cdot), \tag{5.2.1}$$

where n_p is an order of the p-th subsystem.

Thus the order of the whole MIMO system is

$$n := \sum_{p=1}^{m} n_p, \tag{5.2.2}$$

where m is the number of system output. On the other hand, the structure of an identified process by NN-SANARX implies that the order of the state-space representation is

$$n' = m \cdot l, \tag{5.2.3}$$

where l is the number of sub-layers of the neural network with matrices of synaptic weights of inputs and outputs $W_i \in I\!R^{l_i \times (m+m)}$ and $C_i \in I\!R^{m \times l_i}$ of i-th sub-layer.

Consequently, we obtain that if order of all subsystems is the same and $n \geq 2$ then state-space representation is the minimal. Otherwise it can be chosen as multiple of m that leads to $n' \neq n$.

The aim of this research finding the minimal state-space representation of MIMO systems based on neural network ANARX structure for different orders of subsystems, where $n' = n$.

5.3 Minimal state-space representation

Suppose that we have a discrete time NARX model represented by general equation (5.2.1), where n_p is an order of the p-th subsystem.

If a given system has the same order of the subsystems then the number of sub-layers should be chosen equal to them. Further the minimal state-space representation can be found directly from NN-ANARX model using

$$\begin{aligned}
x_{1j}(k+1) &= x_{2j}(k) + C_{1j}\cdot\phi_1(W_1\cdot \\
&\quad\cdot[x_{11}(k),\ldots,x_{1m}(k),\quad u_1(k),\ldots,x_m(k)]^T) \\
x_{2j}(k+1) &= x_{3j}(k) + C_{2j}\cdot\phi_2(W_2\cdot \\
&\quad\cdot[x_{11}(k),\ldots,x_{1m}(k),\quad u_1(k),\ldots,x_m(k)]^T) \\
&\vdots \\
x_{(n-1)j}(k+1) &= x_{nj}(k) + C_{(n-1)j}\cdot\phi_{(n-1)}(W_{(n-1)}\cdot \\
&\quad\cdot[x_{11}(k),\ldots,x_{1m}(k),\quad u_1(k),\ldots,x_m(k)]^T) \\
x_{nj}(k+1) &= C_{nj}\cdot\phi_n(W_n\cdot \\
&\quad\cdot[x_{11}(k),\ldots,x_{1m}(k),\quad u_1(k),\ldots,x_m(k)]^T) \\
[y_1(k),\ldots,y_n(k)]^T &= [x_{11}(k),\ldots,x_{1m}(k)]^T\,,
\end{aligned} \tag{5.3.4}$$

where $j = 1,\ldots,m$.

Otherwise, the number of the sub-layers of the NN-ANARX structure should be chosen as the maximum order of one of the subsystems. Thus we set maximum limit to the possible order of the system, considering the standard structure of NN-ANARX model, since in that case order of the state-space representation is $n' = m\cdot l$. So (4.3.10) can be rewritten as follows

$$y(k) = \sum_{i=1}^{max(n_p)} C_i\cdot\phi_i(W_i\cdot z(k-i)). \tag{5.3.5}$$

Then the neural network structure must undergo further transformations.

If system under control is given as black-box, model estimation of the regressors should be done. Otherwise, regressors can be found directly from the NARX model of the process. Rely on that information all redundant interconnections between hidden layers and outputs should be eliminated. Thus equation (5.3.5) converts to the next form

$$y_i = \sum_{i=1}^{max(n_p)} C_{is}\cdot\phi_i\left(W_{is}\cdot\left[\left\{\delta^{-i}y_{d_{yi}}\right\}_{d_{yi}\in D_{yi}},\left\{\delta^{-i}u_{d_{ui}}\right\}_{d_{ui}\in D_{ui}}\right]^T\right), \tag{5.3.6}$$

if $s = i'$ and $s \in R_i$, then corresponding terms are taken, otherwise they are excluded. Where

$max(n_p)$ - maximal order among all subsystems;

s - number of the output of the NN, with which current sub-layer is connected;

d_{yi} - index of the previous output y on the i-th layer;

D_{yi} - set of indexes d_{yi};

d_{ui} - index of the previous input u on the i-th layer;

D_{ui} - set of indexes d_{ui};

$i' = 1, \ldots, m$;

δ - time shift;

R_i - number of output for which sum is taken.

In other words we obtain a neural network, where each output depends on the specific number of time instances, which corresponds to the order of the subsystem. Using this representation we can easily get the minimal state-space realization.

Consider the following numerical example of MIMO NN-SANARX minimized structure based control using dynamic feedback controller.

Numerical example 5

The model to be estimated is given as discrete-time black-box MIMO model. Estimation process can be done using MATLAB System Identification Toolbox with nonlinear ARX black-box models. The input vector $u(k)$ is composed of 2 variables with 800 data samples.

First of all, for model identification and to obtain the minimal state-space representation, it is necessary to estimate the regressors of the given TITO system. Thus, each output of the model can be taken as a function of regressors, which are transformations of past inputs and past outputs. Typical regressors are simply delayed input or output variables, which are functions of measured input-output data.

On the first step of regressors defining, orders of the sub-models should be set. After the choice of model order, we should choose the nonlinearity estimator to be used. To obtain state-space representation of the model a NN-SANARX structure should be employed for the model identification.

System identification is an iterative process, where we should identify models with different structures from the obtained data and compare model performance. First of all we start by estimating the parameters of simple model structures and gradually increase the complexity of the model structure, if its performance is poor. That gives opportunity to choose the simplest model that best describes the dynamics of a given system.

For the better understanding of the estimation process let rewrite NARX equation (3.2.4), as it is used in System Identification Toolbox. Thus we obtain the following structure

$$y_i(t) = f_i(y_\alpha(k-1), \dots, y_\alpha(k-na_i), u_\beta(k-nk_i), \dots, u_\beta(k-nk_i-nb_i+1), \\ \alpha, \beta = 1, \dots, m), \quad i = 1, \dots, m \quad (5.3.7)$$

Where the function f_i depends on a finite number of the previous inputs u and outputs y. na and nb are the numbers of past output and input terms used to predict the current output, respectively. nk is a delay from the input to output, specified as a number of samples, or *relative degree*. The nonlinear function of the NARX model is a flexible nonlinearity estimator with parameters that do not need to have physical significance [28]. System Identification Toolbox provides several nonlinearity estimators $F(x)$ for NARX models, where x is a vector of regressors. As neural network activation functions are sigmoid, so it was decided to use *sigmoidnet* nonlinear estimators. Over 30 estimation models with different orders and correlations between regressors were studied. Due to fulfill control condition, that $r = 1$, matrix nk does not change.

The most suitable of them are displayed in the figure 5.1.

During examination of the results the outputs simulated with the estimated models and the outputs in the measured data were compared. The best estimated model m_y3322_u3322 with $97,4\%$ and $97,9\%$ fit to the first and second outputs was chosen on the basis of these data (see figure 5.1). This model fits better both subsystems simultaneously. And the parameters of the model are

$$na = \begin{bmatrix} 3 & 3 \\ 2 & 2 \end{bmatrix}, \qquad nb = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}, \qquad nk = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}.$$

Thus, obtained regressors are:

- for the first output $y1 : y_1(k-3), y_1(k-2), y_1(k-1), y_2(k-3), y_2(k-2), y_2(k-1)$,
  $u_1(k-3), u_1(k-2), u_1(k-1), u_2(k-2)$
- for the second output $y2 : y_1(k-2), y_1(k-1), y_2(k-2), y_2(k-1)$,
  $u_1(k-2), u_2(k-2), u_2(k-1)$

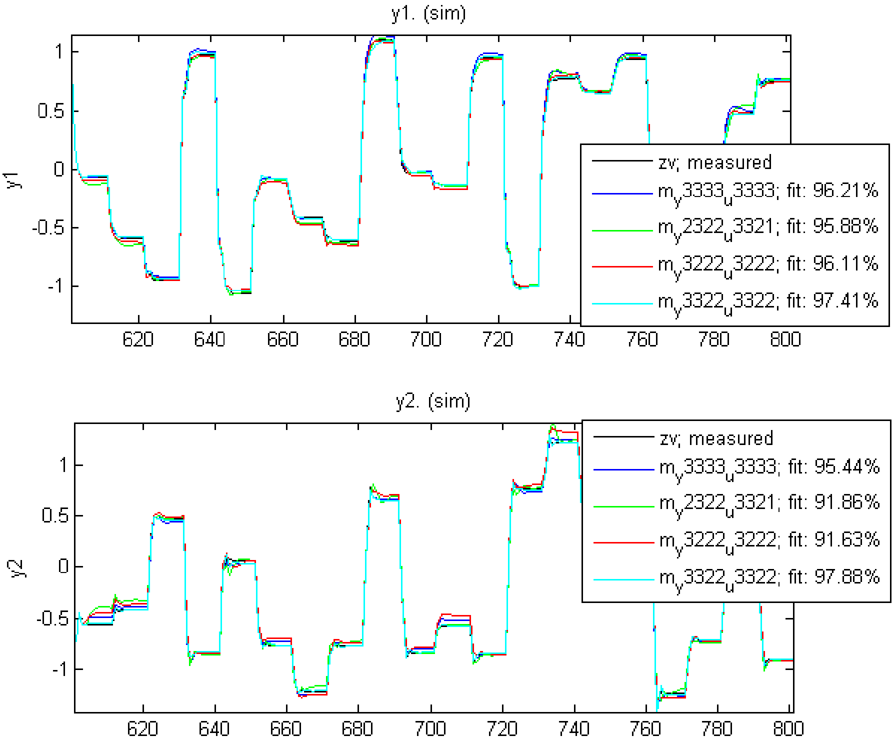

Figure 5.1: Comparison of the different estimated models

The predicted outputs $\hat{y}(k)$ of the nonlinear model at the time-step k are given by the following general equation

$$\begin{cases} \hat{y}_1(k) = F_1(y_1(k-3), y_1(k-2), y_1(k-1), y_2(k-3), y_2(k-2), y_2(k-1), \\ \qquad\qquad u_1(k-3), u_1(k-2), u_1(k-1), u_2(k-2)) \\ \hat{y}_2(k) = F_2(y_1(k-2), y_1(k-1), y_2(k-2), y_2(k-1), \\ \qquad\qquad u_1(k-2), u_2(k-2), u_2(k-1)) \end{cases} \tag{5.3.8}$$

Thereby overall order of the system under the study is 5, as it can be calculated using equation (5.2.2). As the maximum order of the first subsystem is 3, the neural network should be trained with three sub-layers on the hidden layer. On the other hand, the structure of ordinary NN-ANARX implies that order of its state-space representation, according to equation (5.2.3), should be 6. Thus, the minimal state-space representation can be obtained only using custom architecture of the NN-ANARX structure as it shown in figure 5.2.

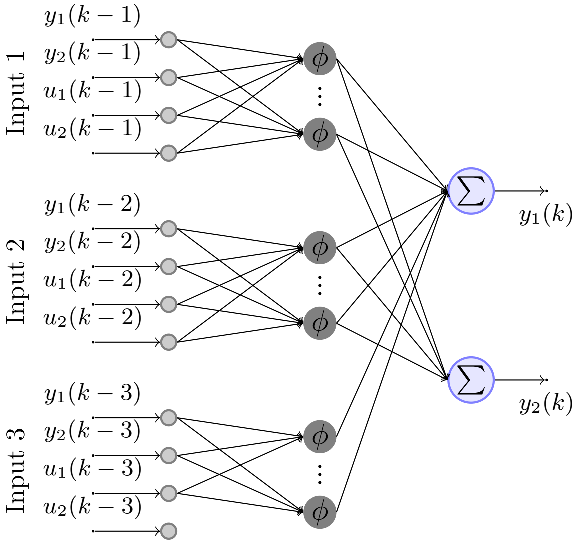

Figure 5.2: NN-ANARX structure of the model(5.3.8)

This scheme clearly shows that all redundant connections between hidden layers and outputs are eliminated. On the third sub-layer unused connections between some inputs and hidden layer are also excluded. As you can see in (5.3.8), $\hat{y}_1(k)$ does not depend on $u_2(k-1)$, but this input should be included because another output depends on it. Unfortunately, with such kind structure of the neural network elimination of the redundant connections between inputs and hidden layer only possible if and only if both outputs do not depend on that data, like in case of input $u_2(k-3)$.

As in case of ordinary NN-ANARX/SANARX structure this neural network also was trained with Levenberg-Marquardt algorithm. LM method is much more efficient than either of other techniques when the network contains no more than few hundred weights [23]. So as ANARX ordinary structure and its custom representation can reduce the number of the used parameters, that leads to increased neural network training efficiency.

The linear activation function was chosen on the first sub-layer with 3 neurons with respect to NN-SANARX structure and the hyperbolic tangent sigmoid activation function on the other hidden sub-layers with 7 and 5 neurons correspondingly.

Thus the minimal state-space representation of the given model can be written as follows according to equation (3.2.6).

$$
\begin{aligned}
x_{11}(k+1) &= x_{21}(k) + C_{11} \cdot W_1 \cdot [y_1(k), y_2(k), u_1(k), u_2(k)]^T \\
x_{12}(k+1) &= x_{22}(k) + C_{12} \cdot W_1 \cdot [y_1(k), y_2(k), u_1(k), u_2(k)]^T \\
x_{21}(k+1) &= x_{31}(k) + C_{21} \cdot \phi_2(W_2 \cdot [y_1(k), y_2(k), u_1(k), u_2(k)]^T) \\
x_{22}(k+1) &= C_{22} \cdot \phi_2(W_2 \cdot [y_1(k), y_2(k), u_1(k), u_2(k)]^T) \\
x_{31}(k+1) &= C_{31} \cdot \phi_3(W_3 \cdot [y_1(k), y_2(k), u_1(k), u_2(k)]^T) \\
y_1(k) &= x_{11}(k) \\
y_2(k) &= x_{12}(k)
\end{aligned}
$$

Using algorithm (4.2.2) we obtain initial states

$$
\begin{bmatrix}
0 \\
0 \\
C_{21} \cdot \phi_2(W_2 \cdot [0 \quad 0 \quad 0 \quad 0]^T) + C_{31} \cdot \phi_3(W_3 \cdot [0 \quad 0 \quad 0 \quad 0]^T) \\
C_{22} \cdot \phi_2(W_2 \cdot [0 \quad 0 \quad 0 \quad 0]^T) \\
C_{31} \cdot \phi_3(W_3 \cdot [0 \quad 0 \quad 0 \quad 0]^T)
\end{bmatrix}.
$$

By using these parameters and equation (4.4.23) the following controller based on state-feedback algorithm, was obtained

$$
D = \begin{bmatrix} 0.1493 & 0.1493 & 1.0768 & 0.0024 \\ -0.5560 & 0.7970 & 0.0173 & 1.0011 \end{bmatrix}.
$$

$$
\begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} = D_2^{-1} \cdot \left(\begin{bmatrix} \nu_1(k) \\ \nu_2(k) \end{bmatrix} - \begin{bmatrix} x_{21}(k) \\ x_{22}(k) \end{bmatrix} - D_1 \cdot \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} \right).
$$

Closed loop control system was simulated with piece-constant and sinusoidal reference signals. The corresponding control signals are shown in figure 5.3. The results of this simulation are depicted in figure 5.4. It can be seen that controls $u_1(k)$ and $u_2(k)$ are capable of simultaneous tracking of the desired reference signals $\nu_1(k)$ and $\nu_2(k)$, respectively.

5.4 Conclusions

Problem of finding the minimal state-space representation is discussed in this chapter. A novel algorithm for neural network based minimal state-space representation of a wide class of nonlinear MIMO systems is proposed as well as its application to the state feedback linearization based control. Thus to obtain a more accurate model with less used parameters the structure of the neural network simplified ANARX model should be simplified. The technique suggested in this work is capable of closely capturing the dynamical behavior of the unknown system with satisfactory performance

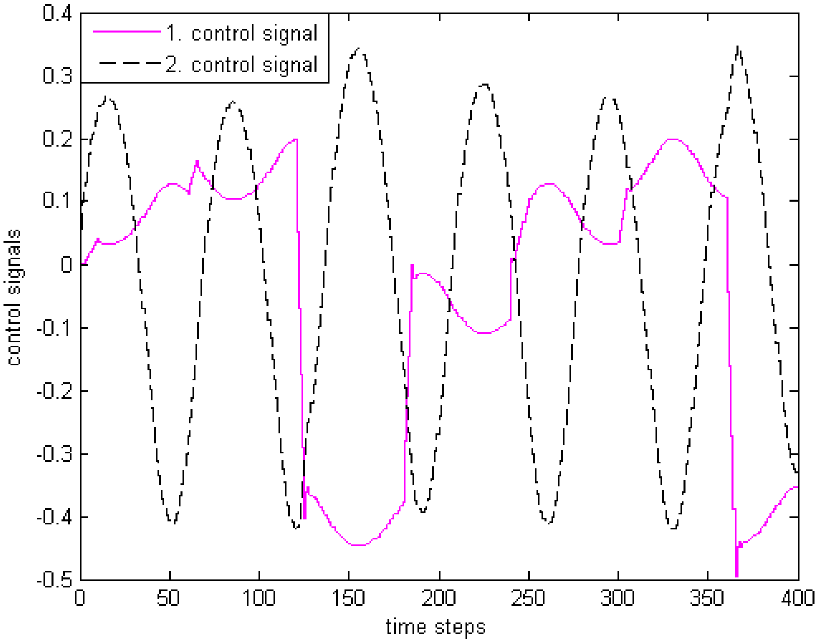

Figure 5.3: Control signals

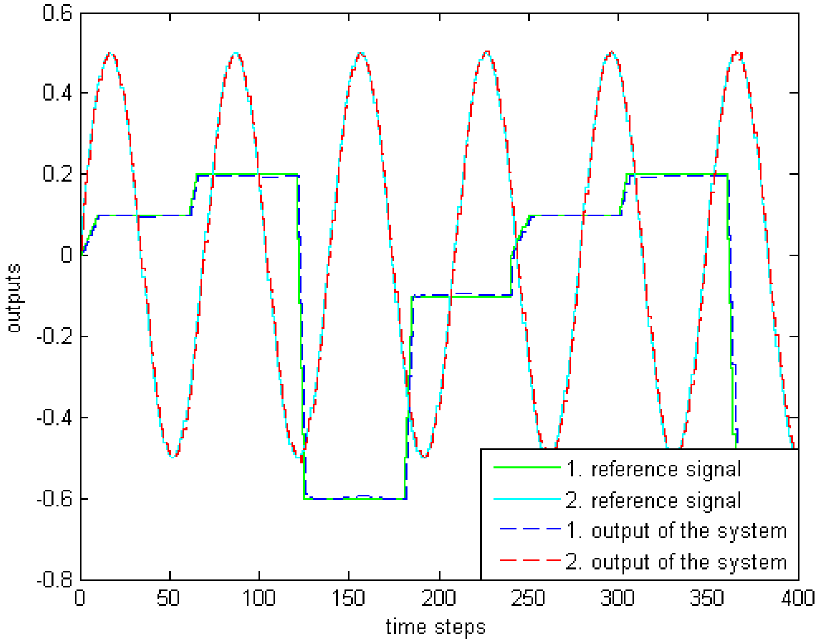

Figure 5.4: Control of the MIMO system using estimated model

of control as it was shown in a numerical example.

Based on the obtained regressors from the black-box controlled system, structure of the neural network provides a minimal state-space representation of the model. However, such structure itself does not guarantee using the minimal number of the parameters of the neural network. Regressors' estimation analysis, made in Matlab System Identification Toolbox, showed that subsystem outputs do not depend on some previous inputs. Despite this, restrictions imposed by the structure of the current neural network do not allow to exclude all unused connections. Thus, connections between inputs and hidden sub-layer can be eliminated only if both outputs do not depend on the same regressors. Hence finding the architecture of the neural network, which fully reflects the dependencies of the system outputs from its regressors and does not have any redundant interconnections and parameters, as represents the minimal state-space representation could be the subject of one of the further chapters.

Another issue raised in this chapter is a usage of the Levenberg-Marquardt training algorithm. This method is very effective, but has a drawback: quantity of the parameters used is limited. Thus, for a complex systems with large number of the parameters using custom representation of ANARX structure is justified by the fact that if allows to increase neural network training efficiency.

Part III

Structural Identification and Model Validation

Chapter 6

Structural Identification

> Everything should be made as simple as possible, but not simpler.
>
> Albert Einstein

In the identification of multivariable processes from input-output sequences a very important role plays selection of the suitable structure of the model. From computational point of view, structural identification presents a challenging problem, especially when we deal with complex system with large number of the parameters. The performance of the model based control algorithms considerably depends on the model accuracy of the controlled process. However, the model is not usually well-defined, because of existing uncertainties and non-modeled dynamics, among other causes. Thus, the key steps are to identify structure and the parameters of the system based on the available data, which requires a good understanding of the system.

Recently research interest has increased in that area due to rapid significant enhancement in computer power and development of new methods.

Structural identification methods can be classified into various categories, such as parametric and non parametric models, deterministic and stochastic approaches, etc.

Origins of the classical methods lie in the late 1970s and early 1980s. They are derived from mathematical theories and generally calculus-based. On the other hand, non-classical methods typically depend on computing power due to extensive search. Analysis used for the ”fitness” evaluation of the test parameters and subsequent selection/guess is based on some heuristic rules as evolutionary approaches or neural networks. These methods based on neural networks and evolutionary algorithms can be considered as general tools for searching and optimization [21, 33, 43].

Genetic algorithms imitate evolution of living things by natural selection. These methods are base on the principal survival of the fittest and developed for various optimization problems. Its advantage in a remarkable balance between exploitation of the good candidates and exploration by random chances. GA has been shown several advantages over classical methods in context of structural identification. Namely more rapid global convergence by conducting population-to-population search, random initial parameters and no requirement of gradient information, as relative ease of implementation [33].

The purpose of structure selection is to find the structure or the order of the model what is the most accurate for its use in control [82]. This method is important from the control point of view, because it gives possibility during identification process obtain a representation of the system with a reduced or minimal number of the parameters.

Combining the neural networks and genetic algorithms gives us possibility to deal with complex nonlinear systems. The advantage of that kind of identification methods generally lies on a fact that the techniques do not require *a priori* knowledge or assumption on the system structure. And only a set of the input-output data is necessary for the identification [7, 19].

Choosing the topology of a neural network for performing given tasks usually requires some prior knowledge of the problem's complexity and usually requires a lot of trial-and-error. Basically architecture of the NN affects two main factors of the training: generalization and training time [42]. In general case it could be said that larger networks with redundant connections and large number of neurons tend to overfit the training data which leads to poor generalization. In addition bigger networks are more computational resource-intensive. If NN is to small, it cannot learn dependencies in training samples. On the other hand, it is easier to understand and extract useful information from a smaller model. For better understanding of the proposed approach it is useful to discuss some of the most common methods for customization or selecting the neural network structure [13, 15, 19, 77].

- The most common approach is the trial-and-error method. The method consists in the training of NNs of different sizes. Network with the smallest number of the parameters that learns the samples is considered as the optimal one. This technique requires some experience in training on the particular problem in order to select the optimal structure

- Another approach is to use a natural method of selection such as genetic algorithm that selects the best network from the population of neural networks and removes unnecessary connections and neurons.
- Destructive method. At the beginning a fairy large neural network is chosen. Then some of the unimportant connections or neurons are removed. This technique trying to achieve increased generalization capability of the network.
- Finally, constructive method. Unlike the previous method, in this case one should start with a small neural network which grows then needed. This method is good because if requires less computational cost than the previous one. However, it is important to regulate the ability of network grow, as the process can lead to the oversized network.

6.1 Author's contribution

The author contribution is devoted to application of genetic algorithms to identification (simplification) of NN-ANARX structure and its implementation of the state-space based algorithm for control of nonlinear MIMO systems.

- Developing neural network structure that gives possibility to use it with GA in structural and parametrical identification;
- Developing the GA for the optimal structure finding with different fitness criteria;
- Formalization of the proposed algorithm using the programming language of MATLAB environment;
- The effectiveness of the proposed techniques is demonstrated on examples.

6.2 Problem statement

Model structure belongs to a set of the most important prerequisites for obtaining an accurate model. Choosing the model structure we should be guided not only by the factors affecting the quality of identification but also take into account conditions imposed by the further usage of identified model. Those conditions are required from the necessity to apply certain methods for model analysis and control syntheses or to limit computational complexity.

There are three main problems in identification for control that can be solved if optimal model is chosen. First, flexibility of describing different systems; second, algorithm complexity to describe different systems, and finally, the reduction of number of the model parameters.

Assuming that only the external description of black-box system is available our objective is to obtain the optimal neural network of specific structure which is capable of capturing the dynamics of the process and has the compact representation. In [74] a recurrent network with the fully automated construction algorithm was proposed. Thus to achieve such a goal we have to develop a fully automated construction algorithm that performs final system order determination and parameters initialization.

For the best model identification a set of neural networks must be trained. Correct choice of the model improves the control quality of the nonlinear processes. The main problem obtaining a good model using neural networks is to find its optimal structure. The problem was firstly described in Kolmogorovs theorem. Further, Kolmogorovs representation theorem was improved by several authors. So in terms of neural networks Sprecher proposed his own version of the representation theorem.

Theorem 3 ([27]) *Any continuous function defined on the n-dimensional cube E^n can be implemented exactly by a three-layered network having $2n+1$ units in the hidden layer with transfer functions $\alpha^{ij}\phi_j$ from the input to the hidden layer and χ from all of the hidden units to the output layer.*

Where α^{ij} are constants, ϕ_j are monotonic increasing functions and χ_i are real continuous functions of one variable.

Thus, there are no exact solution how to choose structure of the neural network. And for the most works in this area estimation of the number of parameters and topology of the NN turns into a trial-and-error process for a specific problem. Besides, quality of the model very much depends on the choice of initial values of the parameters. To solve both problems simultaneously we use genetic algorithm which finds optimal NN parameters and dependencies between the inputs of dynamic model and outputs of the controlled system.

Typically success of the model is evaluated using a performance criterion; in our case it is an efficiency of the controller designed using obtained model. The result of the control simulation can be used as an evaluation function for further calculation of the fitness function of a chromosome in current population. Since we want to obtain a compact state-space representation, a model with the minimal states among the models with the same performance should get more points during the fit function calculation.

Summarize above mentioned, the objective of this work is focused on solving simultaneously the problem of lack of the generic structure and the problem of the minimal realization of the structures.

6.3 Architecture of the Neural Network

Usually most genetic algorithms have only two main components that are problem dependent: the problem of encoding and the evaluation function. According to [75] and [63] the first step is encoding a neural network into the binary strings called chromosomes that is determined by the structure of the neural network.

Consider a discrete-time MIMO NARX, described by equation (3.2.4). This model can be easily obtained by using classical fully connected neural network and covers a wide class of of nonlinear systems. However, classical representation of that model does not assure the minimal/optimal number of the parameter being used in neural network.

The minor corrections in the structure were made earlier, see section 5.3. This approach allows to remove some connections between the hidden sub-layer and output layer. However, this architecture does not make it possible to exclude those connections between inputs and hidden layers if some output depend on one of the inputs. It is only possible if all outputs do not depend on the same input.

Thus, for a more flexible structure, which allows to exclude all redundant interconnections and parameters of the neural network, it is necessary to modify its architecture, as shown below.

Above all, each sub-layer should be divided into groups of neurons each of which is responsible for the interconnections of inputs to each specific output. Thus connections between inputs and hidden sub-layer can be eliminated [68]. For better understanding let consider the following illustrative example.

Illustrative example

Assume that black-box model has two inputs and two outputs. Examine in more detail the obtained structure of the model, which maximal order of a subsystem was found using some regression analysis as three. Thus, architecture of the fully connected NN-NARX model is defined as follows: the maximal order of the subsystem, number of system inputs and outputs, determine the number of inputs of the neural network. This parameter differs from the number of inputs of the identified system. Since we are dealing with feedforward neural network with external feedback, consequently number of the NN inputs is determined by the quantity of the regressors on which it depends. As number of outputs is two, each sub-layer should be divided into two groups L_{11} and L_{12} like it can be seen in figure 6.1. Since ANARX is a subclass of NARX models with some good properties for using it in control, structure of the neural network based on ANARX can be changed in a similar was as shown in figure 6.2.

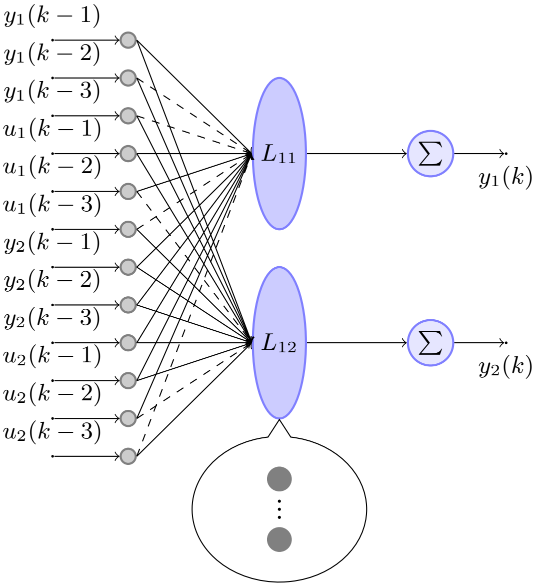

Figure 6.1: Structure of the custom NN-NARX model

This structure is very flexible and allows us to describe nonlinear MIMO system with any interconnections between inputs and outputs of the dynamical model using NN-NARX/ANARX architecture. It makes this architecture available to use in the genetic algorithms for finding the optimal structure of the NN in black-box model identification.

Since we consider the identification process as a part of the control task

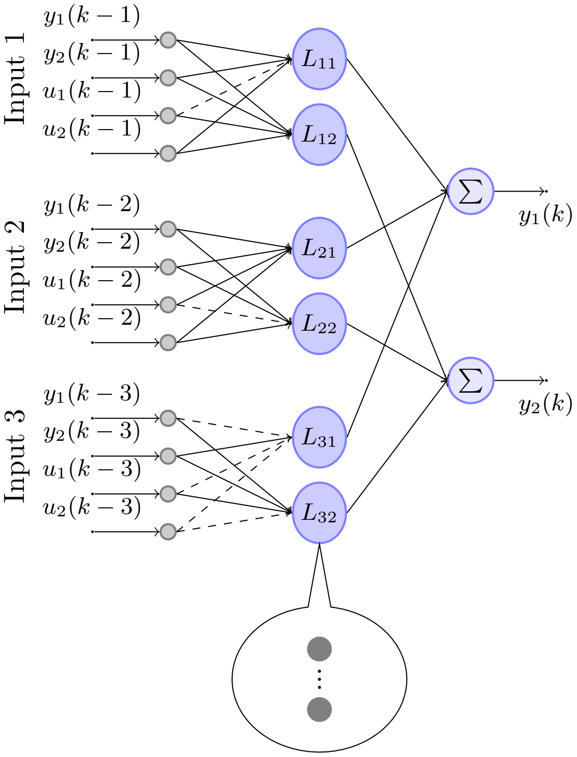

Figure 6.2: Structure of the custom NN-ANARX model

it is logical to use the neural network based additive NARX models. Thus the classical representation of NN-ANARX models (see equation (4.3.10)) can be converted to the same form (5.3.5) as was used to find the minimal state-space representation in section 5.3. Which means that custom NN-ANARX architecture with any interconnections present can be described as follows

$$y_{i'} = \sum_{i=1}^{max(n_p)} \sum_{j=1}^{q_i} \sum_{s \in R_{ij}} c_{ijs} \cdot$$
$$\cdot\, \phi_{ijs}\Big(w_{ijs} \cdot \Big[\Big\{\delta^{i-1} y_{d_{yij}}\Big\}_{d_{yij} \in D_{yij}}, \Big\{\delta^{i-1} u_{d_{uij}}\Big\}_{d_{uij} \in D_{uij}}\Big]^T\Big), \quad (6.3.1)$$

if $s = i'$ and $s \in R_{ij}$, then corresponding terms are taken, otherwise they are excluded. Where
$max(n_p)$ - maximal order among all subsystems;
qi - maximal number of the decomposed sub-layers on the i-th layer;
s - number of the output of the NN, with which current sub-layer is connected;

R_{ij} - set of all connections between i-th hidden layer and output of the j-th subsystem;
d_{yij} - index of the output y on the j-th decomposed sub-layer on the i-th layer;
D_{yi} - set of indexes d_{yij};
d_{uij} - index of the input u on the j-th decomposed sub-layer on the i-th layer;
D_{uij} - set of indexes d_{uij};
δ - time shift.

Consider that dashed lines show the absent connections between layers/neurons and solid - the present ones. Thus, neural network depicted above (see figure 6.2) represents the following system given by general equation

$$\begin{aligned} y_1(k) &= f_1(y_1(k-1), y_2(k-1), u_2(k-1), \\ &\quad y_1(k-2), y_2(k-2), u_1(k-2), u_2(k-2), \\ &\quad y_2(k-3)) \\ y_2(k) &= f_2(y_1(k-1), y_2(k-1), u_1(k-1), u_2(k-1), \\ &\quad y_1(k-2), y_2(k-2), u_2(k-2)), \\ &\quad y_1(k-3), y_2(k-3), u_1(k-3)). \end{aligned} \tag{6.3.2}$$

This scheme is very flexible and clearly shows that all redundant interconnections can be eliminated. Number of groups of neurons on the hidden layer defines the number of the states. Therefore, if the overall order of the system had ones determined, then network size is obtained. It can be seen that in this particular case order of the model equals to 6. As NN-ANARX with custom structure is a representative of feedforward networks with extended feedback, it can be trained with the classical Levenberg-Marquardt algorithm. Linear or sigmoid type functions can be used as activation functions.

6.4 Encoding and optimization

First, determine the length of the chromosome, which describes the neural network in figure 6.2. Parameters that determine its length are:

$$l = o \cdot (n + m) \cdot m, \tag{6.4.3}$$

where l is a length of the gene, o is a maximal order among subsystems of the controlled MIMO system, n and m are numbers of inputs and outputs of the system, respectively. In case of this model $l = 3 \cdot (2 + 2) \cdot 2 = 24$. Thus, for a fully connected NN-NARX the gene will be as follows

$$gene = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \quad 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \quad 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1],$$

where the first bit of the chromosome shows the connection between input $y_1(k-1)$ of the first sub-layer L_{11} and output $y_1(k)$, the second bit - connection between input $y_1(k-1)$ of the first sub-layer L_{12} and output $y_2(k)$, third bit - connection between input $y_1(k-2)$ of the first sub-layer L_{11} and output $y_1(k)$, etc.

If any connection between an input and any output does not exist it can be represented as 0 in chromosome encoding. let existing connections are presented by the solid lines and the absent connections by the dashed lines. Thus, the structure demonstrated in 6.2 can be described using the following gene

$$gene = [1\ 1\ 1\ 1\ 0\ 1\ 1\ 1 \quad 1\ 1\ 1\ 1\ 1\ 0\ 1\ 1 \quad 0\ 1\ 1\ 1\ 0\ 1\ 0\ 0].$$

In order to minimize the number of parameters used, in case when all inputs of one sub-layer are interconnected with all outputs of the neural network, there is no need to divide this sub-layer into groups of neurons. Thus, if NN-NARX (NN-ANARX) model is fully connected, we obtain the classical representation of that structure.

6.5 Evaluation and fitness functions

After the initial population is formed, it is necessary to define the criteria by which it could be determined which of the genes are the best suited. It means that the neural network with such structure would provide the best solution to a given problem. This requires a function that processes the genes and returns a single value indicating the measure of the suitability of this solution. The main focus of this work is directed to the identification of nonlinear processes. Hence, the mean square error (MSE) of the process can be used as objective function, so-called *evaluation function* [40, 65, 75, 76]. Here we should consider two cases: when the ultimate goal is to obtain an accurately identified model or the main goal is to use identified model to control the process. In the first case as MSE an identification error should be taken in the second, we will have a more comprehensive solution.

In applications we consider the set-point tracking problem. The primary aim of out work is to find a suitable controller for the nonlinear process, so fit of the model should be evaluated using one of the control criteria. The most common structure is feedback (see figure 4.1), as this structure can monitor variations in the process and successfully compensate for the unwanted executions in a manner consistent with the performance objectives. Usually a criterion based on the shape of the complete closed-loop response or so-called error-based criterion is chosen.

Further, the quality criterion is calculated on the basis of obtained MSE of the control [68].

$$quality = e^{-k \cdot MSE}, \tag{6.5.4}$$

where k is a proportional coefficient.

Often in the literature evaluation function and fitness function are used interchangeably. However, we distinguish that notions [75]. Evaluation function is objective function that provides a measure of performance, it is independent of evaluation of any other chromosomes. But the *fitness function* is defined as $f_i/\bar{f}$, where f_i is the evaluation associated with $gene_i$ and $\bar{f}$ is an average evaluation of all chromosomes in the population. Thus, fitness function is always defined with respect to other individuals of the current population.

If small population of genes is used in genetic algorithm it is very important to regulate the number of copies. It is common at the beginning of the GA work to have some extraordinary individuals in a population of ordinary members. If evaluation function is left in its original form, then a situation can emerge, where the best individuals would take over a significant proportion of the finite population in the current generation. That would lead to the premature convergence. On the other hand, during the next generations it may happen that despite the variety of genes, their fitness functions differ slightly. That means that population average fitness may be close to the best fitness. So, we will go over the mediocre representatives instead of marking out the successful ones.

In both cases, fitness scaling can help. It can be done like linear scaling, proposed in [20] or by (6.5.4). In this case, even slight decrease of MSE small values will have an effect in the case of forming the new offspring.

6.6 Genetic algorithm description

As it was said earlier in section 3.5 the canonical genetic algorithm is used in our work. Consider it in detail in the context of determining the optimal structure of neural network for model identification.

First, the number of chromosomes is specified. This number remains constant for all future generations. Each gene corresponds to a specific NN-ANARX structure with unspecified weights values. That means that several neural networks with different weights can correspond to the same structure. These weights are determined during the neural network training phase. Further all genes are estimated with the aid of properly selected evaluation function. Depending on the ultimate goal of an identification one of the main criteria for the goodness estimation can be MSE of the identification or MSE of the control. In the second case the controller based on the parameters of neural network is automatically designed, see section 4.3, to obtain necessary signals. On the next step fitness function is calculated. Intermediate generation filled by the chromosomes of the current generation using stochastic sampling with replacement. Crossover with the single recombination point is applied to randomly paired stings. After recombination a mutation operator is used. Also, according to the elitism operator the best chromosomes of the current generation is copied to the next one. Elitism is the last step of our search.

After evaluation each gene is ordered according to its fitness value. Thus, the new population is established and a new generation begins. The algorithm terminates whenever identification or control error is less than an a priori defined threshold or a pre-specified number of generations has been reached.

Proposed algorithm has been formalized using the programming language of MATLAB environment. Further, a separate application package was written with the number of functions. Efficiency of the algorithm was tasted on a number of academic examples [49, 68, 69], the results of which lead to the conclusion about the viability of this technique. This makes sense to its further improvement and development.

Numerical example 6

As it was already mentioned, a developed algorithm is required to obtain the optimal structure of the neural network where all redundant parameters are eliminated. To verify the correctness of the proposed method the following model was used: the model to be estimated is given as a nonlinear discrete-time gray-box model. The input vector $u(k)$ is composed of 2 variables and the output vector $y(k)$ contains 2 variables with 600 data samples.

First, for the model identification and in order to to obtain a state-space representation, it is necessary to estimate the regressors of the given MIMO system. Estimation process can be done using MATLAB System Identification Toolbox with NARX type black-box models. Thus, each output can be taken as a function of regressors which are transformations of the past outputs and inputs. This carry out procedure is similar to finding the minimal state-space model done in example 5. Briefly, first of all, to define the regressors, orders of the models should be set. After the choice of a model order, we should choose the nonlinearity estimator to be used. To obtain state-space representation of the model a NN-SANARX structure

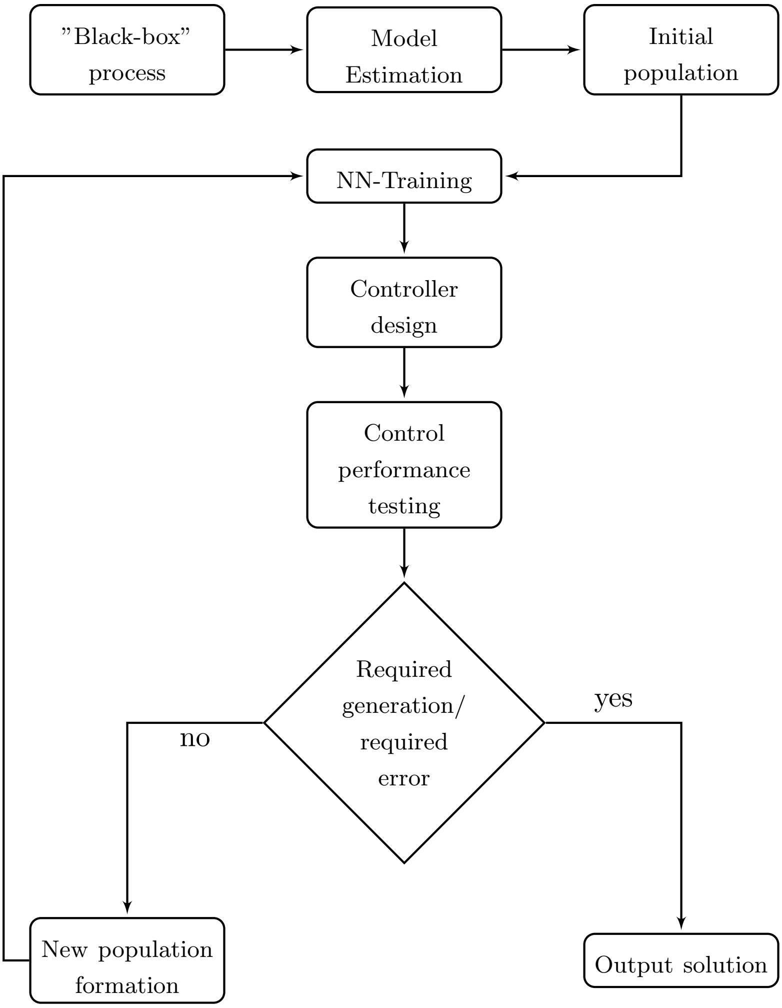

Figure 6.3: Reduced model finding algorithm

(see [52, 71, 72]) could be employed for the model identification. As neural network identification functions were chosen as sigmoid, so it was decided to use sigmoid net nonlinear estimators.

Estimation showed that the maximal order of the subsystems is 3, therefore the neural network should be trained with three sub-layers on the hidden layer. On the other hand, the given system has two outputs; it means that each sub-layer should be divided into two groups of neurons.

According to the proposed algorithm, see figure 6.3 the next step is a creation of an initial population. First of all, the length of the chromosome should be defined using (6.4.3)

$$l = 3 \cdot (2 + 2) \cdot 2 = 24.$$

As for the calculation of the control signals $u_i(k)$ (see 4.4) we need to know

1. the influence of the control signal on the previous time step to the both outputs, so restriction should be imposed on the values of 5-8 bits of the genes: $gene_i(5:8) = 1$;
2. second state of the model, it means that we need to have it explicitly or be able to calculate it using subsequent states. That leads us to the point, that obtained chromosome should be tested for the presence of that interconnections.

On the basis of the obtained genes neural network of the specific structure was generated. Levenberg-Marquardt algorithm was used to perform the training. At this stage number of the training epochs was taken small. In the future, if structure will be suitable for the process, we can train our network additionally to obtain more precise control. The linear activation functions was chosen on the first sub-layer with 2 neurons, with respect to NN-SANARX structure. And the hyperbolic tangent sigmoid activation function was on the other hidden sub-layers with 4 nodes for each group of neurons. On the next step neural network parameters were used to design a controller, based on the state-space algorithm, see section 4.4. Closed loop control system was simulated with piece-constant and sinusoidal reference signals, then quality criterion was calculated.

Described earlier GA with different initial population size from 50 to 200 individuals was applied. Crossover rate varied from 90% to 98% and mutation probability was around 1%. Depending on the size of the population percentage of the parents in the next generation ranged from 1% to 2%.

Distinctive feature of the applying GA to the neural networks is that the same gene could correspond to different neural networks. Although the neural networks have the same structure, due to the different initial parameters their final weights are different. On this basis two approaches were used: in the first case several neural networks were trained for one chromosome and the best of them was chosen for the representation of the gene. It is necessary not to lose a good structure if initial parameters for the training the NN has led to poor results. In the second case only one neural network for the gene was trained. Further experiments showed that total time needed for optimal NN search was almost the same. The only difference was that in the second approach first took place the structural identification and only then for the best structure the optimal neural network was chosen. In the first case structural and parametrical identification took place for the same time. That naturally increased time for the formation of a new generation.

If obtained structure and designed with its aid controller gave unstable behavior of the system, then MSE equate to 50. Such big value of the MSE is needed to obtain fitness function close to 0. In the most cases such situation emerge due to the fact that during the training matrices of synaptic weights were obtained close to the singular, see equation (4.4.23).

Another significant remark should be done. Several experiments were conducted with initial population containing a few genes which describe a fully connected neural network of ANARX type. All results have shown that for this kind gray box model the classical structure what was used earlier is not the optimal one.

Depending on how good was the initial selection, fitness function in the first phase was from 6 to 3.5 for the best representative of the first generation. But just in a couple of generations, this value could drop up to 1.5. In general, the final stage of the value function dropped somewhere before 1.3. However, if a good neural network was present at the beginning of training, the final stages of the fitness function dropped to 1.017 for the best individual. This means that almost all genes have the same opportunities for reproduction.

Most experiments came to the same structure, described by the next gene

$$gene = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \quad 0\ 1\ 0\ 1\ 1\ 1\ 1\ 1 \quad 0\ 0\ 1\ 0\ 0\ 0\ 1\ 1].$$

On this basis the custom architecture of the NN-SANARX structure could be as depicted in figure 6.4.

Some experiments with population size more than 150 individuals had two groups of genes: one of them is mentioned before and another is

$$gene = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \quad 0\ 0\ 1\ 0\ 1\ 1\ 1\ 1 \quad 0\ 1\ 1\ 1\ 0\ 0\ 1\ 1].$$

So in all cases the order of the obtained models equals to 6. Closed loop control system was simulated with piece-constant and sinusoidal reference signals. The results of this simulation are depicted in figure 6.5. It can be seen that controls $u_1(k)$ and $u_2(k)$ are capable of the simultaneous tracking of the desired reference signals $v_1(k)$ and $v_2(k)$, respectively.

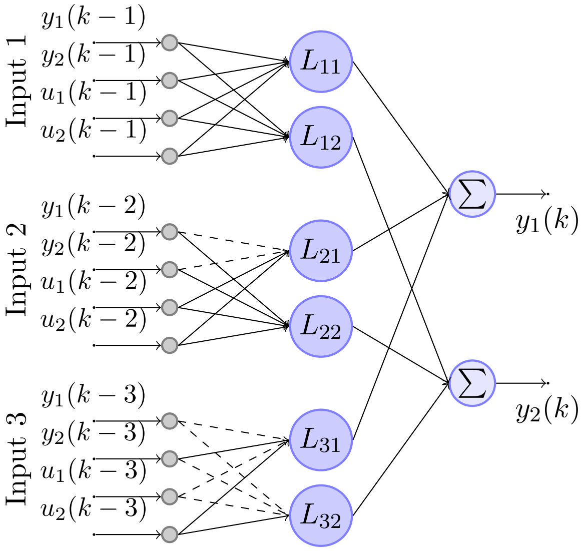

Figure 6.4: neural network custom structure of the controlled model

In the example above, the choice of the model is made only by one criterion. Thus, one can get a result where several models with different orders will have almost the same value of the control error. In this case a Occam's blade rule should be used, which states that among the models of the same quality the simplest one should be chosen. In other words, the preference should be given to the model with the lowest order. To achieve a balance between goodness of the fit and the complexity of the model is possible if the evaluation function is calculated on the basis of several criteria. One of the possible realizations is proposed in the next example.

Numerical example 7

This example is a logical continuation of the Numerical example 6.

As it was said earlier the objective of our work is to find a suitable controller of compact form for the nonlinear MIMO process. Now, in case that the model with a lower order takes precedence of the more complex one, fitness function has been taken as following

$$quality = e^{-k \cdot MSE} + e^{-o}, \tag{6.6.5}$$

where k is a proportional coefficient and o is gradually depends on the number of the states of the current neural network. The less states neural network has the less the o. Choice of the k and o grades should be done carefully to avoid such situations, where the model with small number of states and poor performance is preferable than the model with more states

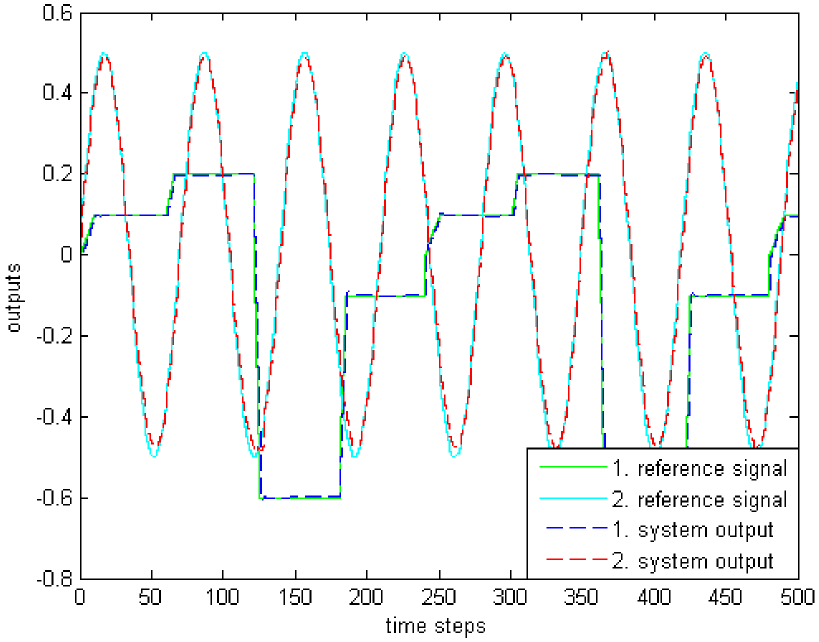

Figure 6.5: Control signals

and good performance.

In general case usage of expositional function is justified by the fact that during the fit function calculation the poor control performance could overweight and distort the overall picture. So we get that big difference in large values of the errors will have less impact than minor errors of control in positive direction. It should be mentioned, that such method of evaluation function calculation allows to limit the growth of the model order during searching for the optimal structure, but does not guarantee to find a model with the minimal number of states.

As the gray-box model a 5-th order MIMO model was taken. Thus, a neural network with 3 sub-layers on the hidden layer should be trained, as it was also done in the previous example. Since by the 19 generation more than 40 genes out of 50 possible converged to the below mentioned structure, the process of optimal structure search has been stopped. Obtained gene

$$gene = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \quad 0\ 1\ 1\ 1\ 1\ 0\ 1\ 0 \quad 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1]$$

can be represented graphically as follows.

Clearly, we are dealing with 5-th order model, which is equivalent to the

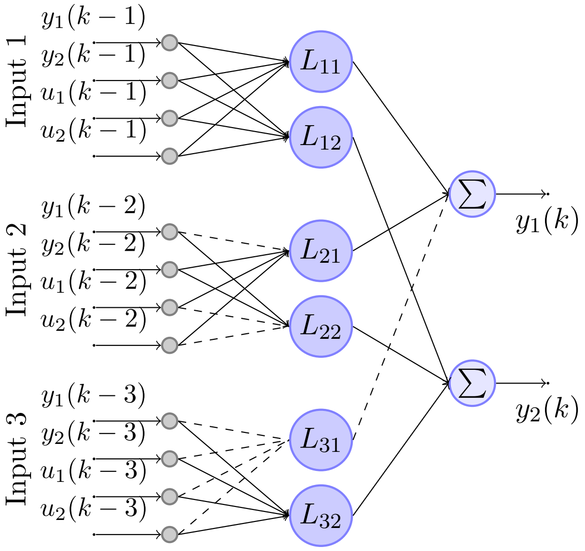

Figure 6.6: Custom structure of the NN model

initial model. Hence introducing an additional criterium, we obtained a model that more accurately reflects the actual model. Closed loop control system was simulated with piece-constant and sinusoidal reference signals. The results of this simulation are depicted in figure 6.7. It can be seen that controls $u_1(k)$ and $u_2(k)$ are capable of the simultaneous tracking of the desired reference signals $v_1(k)$ and $v_2(k)$, respectively.

6.7 Conclusions

In this chapter problems of system identification of dynamic systems are considered. Proposed algorithms for solving such problems are based on the feed-forward neural networks of specific structure. First of all, we have focused on the issue of identification for control of nonlinear MIMO systems. Thus, the main contribution is devoted to the novel representation of the ANARX or more general case, NARX structure. That leads to the possibility of using such a structure for the formation of any genes, which in turn allows the use of the genetic algorithms for more flexible model order identification and parameter optimization.

The possibility of obtaining optimal structure which is not necessary fully connected enables to reduce the number of the parameters used in NN. It is a significant factor especially in adaptive control, when improvement of the weights of the neural network occurs in on-line mode.

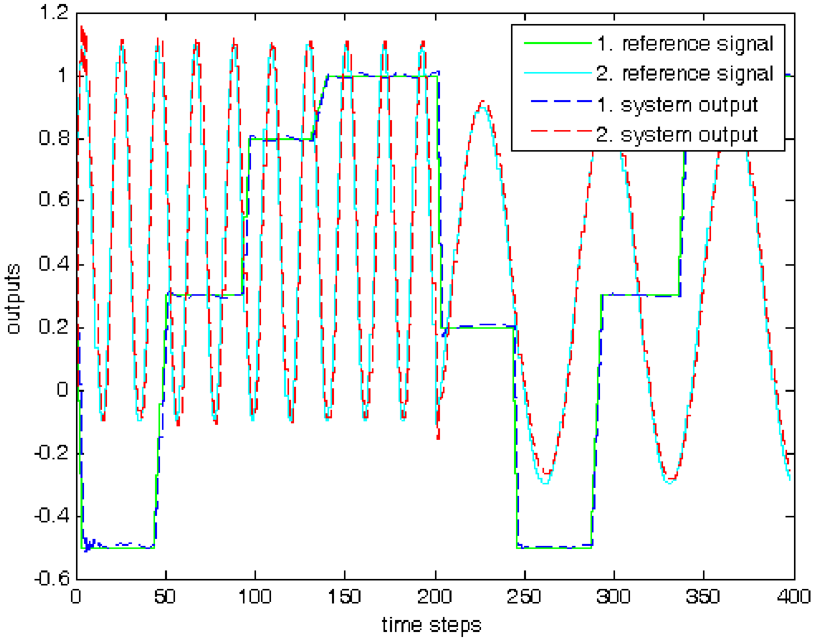

Figure 6.7: Control signals based on two quality criteria

Usually, since a black-box model is used, the minimal order of the system is not known and data received from the regression analysis is not necessarily the right one. Having in mind that direct value of the minimal order of the system is absent, so for our practical needs, we choose the best model from the GA point of view being in the interval $min(N) \leq NN(N) \leq max(N)$, where N is the order of the system and $NN(N)$ is the order of model obtained using neural networks.

Chapter 7

Model Validation

Validation procedure is almost as old as the identification procedure. Model validation step allows to provide certification that model can be used for particular application. It guarantees that model is able to achieve purpose for which it is used. In addition model validation determines whether the obtained model is flexible enough to describe the system.

7.1 Author's contribution

Present chapter focuses its attention on the application of genetic algorithm to adjust the NN-ANARX type structure improving performance of the identified model. Constructive procedure is proposed to choose parameters of the multi-criteria fitness function, whereas main goal of present research is to find optimal linear combination of those parameters that are commonly used to evaluate model performance and validity

- application of the correlation-test-based approaches as one of the validation criterion;
- find optimal linear combination of three qualitative parameters: OD-CCF based criteria, mean square error and model order.

The result of the derived evaluation function is obtaining the minimal structure of the neural network in terms of goal-oriented validation.

7.2 Application of the correlation-test-based approaches

Quite often it is considered that mean square error provides sufficient information about quality of identified model. Procedure to validate identified neural network model on the basis of correlation test was described in

[81] and later adopted to compare quality of NN-ANARX and NN-NARX models on the same system in [47]. Present contribution is devoted to the application of genetic algorithm to determine the structure of the model whereas fitness function will depend both on MSE and results of correlation based test. Such process can be seen as trade off between finding the order of the system and elimination of certain redundant interconnections and weights.

In system identification procedure validation is the final step to check the performance of the obtained model. Generally the correctness of the identified model is checked on the basis of residuals: their mean, variance or standard deviation. However, the low residual values are not always clearly and directly point to correctly identified model based on neural networks, especially in unknown and noisy environment. The main idea is if one has a proper model its residuals reduce to white noise and uncorrelated to the delayed system inputs and outputs. To properly validate nonlinear models several techniques based on correlation tests have been developed that use correlation-test-based approaches to detect the nonlinear correlations between residuals and delayed residuals, inputs and outputs. Two methods, namely combined omnidirectional autocorrelation function (ODACF) and combined omnidirectional cross-correlation function (ODCCF) , were used to construct a set of nonlinear model validity tests in [80]. To overcome problems with non-detection of all possible omitted regressors in residuals, the combined ODACF- and ODCCF- based model validation technique were developed and applied to check the quality of identified model based on neural networks [81] and [47].

Unlike the linear case there are four types of nonlinear associations.

- **Type 1:** The amplitude of the dependent variable varies as the amplitude of the independent variable varies.

- **Type 2:** The amplitude and the sign of the dependent variable varies as the amplitude of the independent variable varies.

- **Type 3:** The amplitude and the sign of the dependent variable varies as the amplitude and the sign of the independent variable varies.

- **Type 4:** The amplitude of the dependent variable varies as the amplitude and the sign of the independent variable varies.

Let $\{\gamma(t), t = 1, \dots, N\}$ and $\{\eta(t), t = 1, \dots, N\}$ denote two arbitrary data sequences (usually $\eta(t)$ represent independent variable and $\gamma(t)$ - dependent variable). The normalized sequences $\{\gamma'(t)\}$ and $\{\eta'(t)\}$ with removed mean level are defined as follows:

$$\gamma'(t) = \gamma(t) - \frac{1}{N}\sum_{t=1}^{N}\gamma(t), \tag{7.2.1}$$

$$\eta'(t) = \eta(t) - \frac{1}{N}\sum_{t=1}^{N}\eta(t). \tag{7.2.2}$$

Denote by $\alpha(t)$ and $\beta(t)$ sequences which elements are absolute values of the sequences $\{\gamma'(t)\}$ and $\{\eta'(t)\}$ correspondingly.

$$\alpha(t) = \Big|\eta'(t)\Big|,$$

$$\beta(t) = \Big|\gamma'(t)\Big|.$$

Finally, normalize those sequences again by removing mean level

$$\alpha'(t) = \alpha(t) - \frac{1}{N}\sum_{t=1}^{N}\alpha(t), \tag{7.2.3}$$

$$\beta'(t) = \beta(t) - \frac{1}{N}\sum_{t=1}^{N}\beta(t). \tag{7.2.4}$$

The validation procedure is based on the analysis of the set of first order omnidirectional cross-correlation functions which is able to identify four types of above-mentioned nonlinearities.

- **Type 1:**

$$r_{\beta'\alpha'}(\tau) = \frac{\sum\limits_{t=\tau+1}^{N}\alpha'(t)\beta'(t-\tau)}{\Big[\Big(\sum\limits_{t=1}^{N}\alpha'^2(t)\Big)\Big(\sum\limits_{t=1}^{N}\beta'^2(t)\Big)\Big]^{1/2}}, \tag{7.2.5}$$

- **Type 2:**

$$r_{\beta'\gamma'}(\tau) = \frac{\sum\limits_{t=\tau+1}^{N}\gamma'(t)\beta'(t-\tau)}{\Big[\Big(\sum\limits_{t=1}^{N}\gamma'^2(t)\Big)\Big(\sum\limits_{t=1}^{N}\beta'^2(t)\Big)\Big]^{1/2}}, \tag{7.2.6}$$

- **Type 3:**

$$r_{\eta'\gamma'}(\tau) = \frac{\sum\limits_{t=\tau+1}^{N} \gamma'(t)\eta'(t-\tau)}{\Big[\Big(\sum\limits_{t=1}^{N} \gamma'^2(t)\Big)\Big(\sum\limits_{t=1}^{N} \eta'^2(t)\Big)\Big]^{1/2}}, \qquad (7.2.7)$$

- **Type 4:**

$$r_{\eta'\alpha'}(\tau) = \frac{\sum\limits_{t=\tau+1}^{N} \alpha'(t)\eta'(t-\tau)}{\Big[\Big(\sum\limits_{t=1}^{N} \alpha'^2(t)\Big)\Big(\sum\limits_{t=1}^{N} \eta'^2(t)\Big)\Big]^{1/2}}. \qquad (7.2.8)$$

In (7.2.5) - (7.2.8), τ denotes the time delay.

Combined ODCCF's are defined as follows:
If

$$\Big|\max\big(r_{\beta'\alpha'}(\tau), r_{\beta'\gamma'}(\tau), r_{\eta'\alpha'}(\tau), r_{\eta'\alpha'}(\tau)\big)\Big| > \Big|\min\big(r_{\beta'\alpha'}(\tau), r_{\beta'\gamma'}(\tau), r_{\eta'\alpha'}(\tau), r_{\eta'\alpha'}(\tau)\big)\Big|,$$

then

$$\begin{aligned} \rho_{\gamma\eta}(\tau) &= \max\big(r_{\beta'\alpha'}(\tau), r_{\beta'\gamma'}(\tau), r_{\eta'\alpha'}(\tau), r_{\eta'\alpha'}(\tau)\big) \\ &\text{else} \\ \rho_{\gamma\eta}(\tau) &= \min\big(r_{\beta'\alpha'}(\tau), r_{\beta'\gamma'}(\tau), r_{\eta'\alpha'}(\tau), r_{\eta'\alpha'}(\tau)\big). \end{aligned} \qquad (7.2.9)$$

The equation above allows to show more clearly detected correlations. The values of combined ODCCF's are found for the following pairs

- residuals and delayed outputs - $\rho_{\epsilon\epsilon}(\tau)$,
- residuals and delayed inputs - $\rho_{\epsilon u}(\tau)$,
- residuals and delayed residuals - $\rho_{\epsilon y}(\tau)$.

Since the values of $\rho_{\epsilon u}(\tau)$, $\rho_{\epsilon y}(\tau)$ and $\rho_{\epsilon\epsilon}(\tau)$ belong to the confidence interval, correlations between residuals and delayed residuals, delayed outputs and delayed inputs are insignificant and therefore identified NN is valid.

In [81] the values of cross-correlation functions were computed for time delays $\tau = 1, 2, \ldots, 20$. For the 95% confidence level the confidence interval is $\pm 1.96/(\sqrt{N})$, where N is the number of data samples.

Definition 10 ([57]) *A confidence interval is an interval in which a measurement or trials falls corresponding to a given probability.*

In other words, the width of the confidence interval gives us some idea about how certain one about the unknown parameter, where confidence level is the probability value associated with confidence interval [46].

In order to use results of correlation based test to compute the value of evaluation function one should summarize the values of combined ODCCF's computed for all time delays $\tau = 1, \ldots, 20$ and for all associations. In [47] qualitative parameter computed as the mean of the means of the means of cross correlation coefficients was proposed for this purpose. In the framework of current work similar idea was employed [49].

On the basis of the values of combined ODCCF's one may define the qualitative parameter as it is shown in (7.2.10).

$$Q_{st} = \frac{1}{3}\left(\left(\frac{-p_1}{\tau-1}\sum_{\imath=2}^{\tau}\rho_{\epsilon\epsilon}(i)+p_2\right)+\left(\frac{-p_1}{\tau-1}\sum_{\imath=2}^{\tau}\rho_{\epsilon y}(i)+p_2\right)\right.$$
$$\left.+\left(\frac{-p_1}{\tau-1}\sum_{\imath=2}^{\tau}\rho_{\epsilon\epsilon}(i)+p_2\right)\right), \quad (7.2.10)$$

here parameters p_1 and p_2 defined in the following way: if all the values of combined ODCCF's fall into the confidence interval then $p_1 = 1/(2p_c)$ (p_c is the critical value) and $p_2 = 1$. If at least one value of values of combined ODCCF's falls out of confidence interval then $p_1 = p_2 = 0.5$ Such procedure assures, that valid model (all the values of combined ODCCF's fall into confidence interval) always has higher value of qualitative parameter than the model which is not valid (at least one value of combined ODCCF's does not belong to the confidence interval).

The following examples are based on the results presented in [49].

Numerical example 8

To validate proposed technique models of real-life systems known in the literature will be used. Of cause if we deal with unknown plants and processes order is usually unknown. Such approach allows to see wherever evaluation converge to the model of the same order as "unknown" system or not.

In this work canonical genetic algorithm is used. Each gene corresponds to a specific NN-ANARX structure with unspecified weights values. These weights are determined during the neural network training phase. Next model validation and correlation based test are made. Further on the basis of the obtained evaluation parameters fitness function is calculated.

As there are three criteria for the best model selection, we need to integrate a set of different criteria in order to determine the most suitable model for all of them. The best model could be obtained in the following steps, using technique proposed in [70]:

- **Step 1.** Sort all models in ascending/descending order by one of the criteria. That task should be done for statistical and MSE criteria. As a result we obtain a table with two columns, elements of which are the corresponding index numbers p^{MSE} and p^{st}, respectively.

- **Step 2.** For each of parallel models a quality coefficient q has to be found. This coefficient is a sum of the position number for each of the criterion multiplied by weighting factor. This factor indicates the importance of the corresponding criterion during the model selection.

- **Step 3.** Model with the smallest weighting factor is the winner or so-called the best model.

Next, we normalize quality coefficients such that $p^{MSE}, p^{st}, n \in [0 \dots 1]$. So after these transformations, we obtain the following evaluation function

$$q_i = k_1 \cdot p_i^{st} + k_2 \cdot p_i^{MSE} + k_3 \cdot n_i, \tag{7.2.11}$$

where $i = 1, \dots, N_m$ and N_m is a number of candidate models of the one specific system. Moreover, we impose the following restriction on the coefficients of equation (7.2.11)

$$0 \leq k_1 + k_2 + k_3 \leq 1.$$

Intermediate generation filled by the chromosomes of the current generation using stochastic sampling with replacement. Crossover with a single recombination point is applied to randomly paired strings. After recombination, a mutation operator is used and according to the elitism operator the best chromosome of the current generation is copied to the next generation. Elitism is the last step of our search.

A. *Model of electrical generator*

In order to illustrate proposed technique, let us consider the model of electrical generator, describing the relationship between the varying part of the current and the frequency of the generated voltage [22]

$$\begin{aligned} y(t+4) \;&= -0.00113 - 0.0628u(t+2) - 0.0675u(t+1) \\ &+0.84y(t+3) - 0.0526u(t+1)y(t+2) \\ &-0.053u(t+2)y(t+3) + 0.0613y^2(t+3) \\ &-0.0071u(t+2)u(t+1) - 0.0234u^2(t+2)y(t+3) \\ &-0.044u^2(t+1)y(t+3) + 0.0573u(t+2)y^2(t+3) \\ &-0.02y^2(t+1). \end{aligned} \tag{7.2.12}$$

We treat this the 4-th order system as our unknown plan. Note, that (7.2.12) does not have ANARX structure. To obtain input-output data, system (7.2.12) was simulated with uniformed random signal. Levenberg-Marquardt algorithm was used to perform the training. The hyperbolic tangent sigmoid activation function was used on the hidden sub-slayers. Initial population consisted of 50 models whereas maximal model order was 7. According to equation (7.2.11) weighting factors were chosen as follows $k_1 = 0.3, k_2 = 0.2, k_3 = 0.5$. Evolution of the GA tooks place for 20 generations. On the basis of quality parameters 5 models were selected (all the values of combined ODCCF's were within the confidence interval). Their order varied between 3 and 4 and mean square errors were less than 3×10^{-2}, which is within acceptable limits for majority of the applications. Table 7.1 shows the best models for different confidence intervals for the first and the last generations. The fact that order of the final models were 3 and 4 allows to conclude that proposed technique converges to the original system order. During different experiments convergence rates of quality parameters differs a lot, which leads necessity to study separately influence of different initial parameters on convergence rate.

Table 7.1: Experimental results

confidence interval	criterion	generation	
		1	20
95%	MSE	0.02829	0.02783
	ODCCF's	0.4686	0.4710
	Order	3	4
90%	MSE	0.02801	0.02788
	ODCCF's	0.4672	0.4711
	Order	3	4

B. *Liquid level system of interconnected tanks*

Let us now consider the model of a liquid level system of interconnected tanks (LLS) [5]

$$\begin{aligned} y(t+3) &= 0.43y(t+2) + 0.681y(t+1) - 0.149y(t) \\ &\quad +0.396u(t+2) + 0.014u(t+1) \\ &\quad -0.071u(t) - 0.351y(t+2)u(t+2) \\ &\quad -0.03y^2(t+1) - 0.135y(t+1)u(t+1) \\ &\quad -0.027y^3(t+1) \\ &\quad -0.108y^2(t+1)u(t+1) \\ &\quad -0.099u^3(t+1) \end{aligned} \tag{7.2.13}$$

Unlike model of electrical generator (7.2.12), model of LLS belong to ANARX model class. To obtain input-output data, system (7.2.12) was simulated sinusoidal input signal whereas small amount of white noise was added. All the other parameters describing initial population and NN-models are the same as in previous experiment described above. After 20 generations algorithm has converged to the population of models where majority were the NN-ANARX models of order 4 also number of lower and higher order models, with certain connections eliminated, were present. Again the algorithm converged to the models of the same order as original "unknown" plant.

7.3 Conclusions

To obtain an order derived from the calculations that would be close to the minimal order of the system, it was suggested in GA the use of the evaluation functions that based not only on the mean square error but on several other criteria. In case of the SISO systems another criterion could be based on the values of combined omnidirectional cross-correlation functions.

Examples considered in the framework of present research has clearly demonstrated that proposed methods converge to NN-ANARX structure of the same order as original system or to the NN-ANARX structure of slightly higher order whereas certain connections are eliminated.

Conclusions

Problems of the identification based on artificial neural networks for control of nonlinear SISO and MIMO systems and control design were considered in the present work. This chapter summarizes what have been done and what goals were achieved in this work.

Concluding remarks

Choosing a proper structure of the model can significantly improve the quality of a model based control and reduce computational costs. Architecture of the neural network based model has to be selected in accordance with requirements of control application to be used. Moreover, alternation of the structure of the neural network extends the number of the control algorithms that could be combined with neural network based modeling. In this thesis the use of the Neural Network based Additive NARX structure making possible combination of neural network based identification with classical control algorithms. Application of the state feedback linearization is designed in Matlab/Simulink environment.

The most significant result of this work is the modification of the neural network architecture that gives possibility to eliminate any redundant connections between system inputs and outputs. In case of MIMO systems, that approach allows us to obtain minimal state-space representation based on the parameters of the neural network, if order of the model is already known. Overwise, topology of the neural network (including the order of the system) should be selected manually.

Slight modification of NN-NARX as well as NN-ANARX architecture allows to describe it by a binary strings called chromosomes. Thus, the tedious try-and-error process can be automated by genetic algorithm where some control criteria is taken as evolution function. First of all, such techniques allow to find the optimal model not so much in terms of accuracy of identification itself, but in terms of process control. Another important factor is that in case of using genetic algorithm to find the NN-ANARX structure, we do not face the problem of so-called competing conventions. As encoded chromosome specifies a relationship between specific input and output, as well additive subclass of NARX models has all time instances separated, so different genotypes cannot map into the same or equivalent phenotypes. In other words, neural networks described by the very different binary strings will have completely different structures.

Research have shown that genetic algorithms are effective search techniques, but they have their own limitations. Such techniques are known to be sensitive to control parameters, these are: population size, rates of mutation and recombination, selection methods. Adjusting these parameters in different ways one can get a variety of algorithms that are inherently slow or fast. Fast algorithm rapidly exploits best genes in the early populations, but it leads to the loose of genetic diversity and eventual stagnation. On the other hand, progress of the slow algorithms is less rapid on the early stages, but having more genetic diversity, that kind of algorithms may eventually surpass a faster methods by finding even better combinations of genes. In short, the choice of a finite algorithm depends on the task and the complexity of the problem.

Use of different criteria as the fitness function gives possibility to find a good controller based on the parameters of obtained neural network with reduced number of parameters. This approach allows us to solve the problem of the model overparametrization. A very high degree of correlation among parameters may provide a useful indicator of overparameterization which leads to slow convergence and computational costs. Thus, elimination of the redundant parameters is especially critical when the on-line adaptation is used.

The effectiveness of the proposed techniques is demonstrated on numerous numerical examples.

Contents of publications

The present thesis is based on 6 academic papers listed in List of publications.

P[1] Describes how using ANARX or NN-ANARX structure one can obtain state-space representation of the model. Further, based on this data classical control algorithms can be used. The main contribution of this paper is in application of the state feedback linearization algorithm to control nonlinear SISO and MIMO systems.

P[2] Describes a technique that allows fully automated model selection for control based on feedback linearization. Further, software written in this work were implemented to evaluation function calculation in genetic algorithms.

P[3] The main contribution of this paper is in proposed novel algorithm for neural network based minimal state-space representation of a wild class of nonlinear MIMO systems and its application to the state feedback linearization based control.

P[4] This paper represents feed-forward neural network with external feedback of specific structure for dealing with dynamic problems. We have focused on the issue of identification for control of nonlinear dynamic MIMO systems in the state-space form. The proposed algorithm consists of two mechanisms: a minimal state-space realization technique for model with some a priori knowledge or reduced model representation if a priori information about the system is absent.

P[5] The main contribution of this paper is in proposed novel representation of the NN-NARX (NN-ANARX) structure which easily can be encoded by binary strings. Thus, topology of the network is chosen by means of canonical genetic algorithm. The possibility of obtaining optimal neural network structure which is not necessarily fully connected enables to reduce the number of the parameters used in controller.

P[6] Application of genetic algorithms for identification of NN-ANARX structure is presented in this paper. Main distinctive property of proposed approach is that evaluation function is based not only on means of square error but also on the values of combined omnidirectional cross-correlation functions and model order. Examples considered in the framework of present research has clearly demonstrated that proposed method converges either to NN-ANARX structure of the same order as original system or to the NN-ANARX structure of slightly higher order whereas certain connections are eliminated.

Author's contribution to the publications

In P[1] the general form of the dynamical state controller for SISO systems was designed during discussions with Eduard Petlenkov and Juri Belikov. The author extended it to the more general case - control of MIMO systems. All applications of the proposed technique were contributed by the author.

In P[2]-P[5] work of the co-authors was of consultative nature.

In P[6] Sven Nõmm worked on a part that relates to the use of statistical methods. The author carried out a part that was related to genetic algorithm and contributed especially the experimental part.

Future work

Results achieved in the framework of this thesis give impetus to several research directions.

Development of algorithm for control signal calculation for MIMO nonlinear systems, where simplification of the model is not used.

More work should be done on examination of impact of each of the criteria for calculating the evaluation function if the model fitness calculation is base on the several criteria. While in this work choice of the weighted coefficients still very much depends on the expert experience, there may be an opportunity to give a general recommendation on the choice of the parameters or how it depends on the class of the studied system.

As current work considers a specific class of models with ANARX structure, so the applicability of neural networks with restricted connectivity based approach for identification of associative models could be the subject of further research.

References

[1] S. Bandyopadhyay and S. K. Pal. *Classification and Learning Using Genetic Algorithms.* Springer, Berlin, Heidelberg, 2007.

[2] J. Belikov and E. Petlenkov. Calculation of the control signal in MIMO NN-based ANARX models: Analytical approach. In *The 10th International Conference on Control Automation Robotics and Vision*, pages 2196–2201, Hanoi, Vietnam, December 2008.

[3] J. Belikov, E. Petlenkov, and S. Nõmm. Application of neural networks based ANARX structure to backing up control of a truck-trailer. In *6th IFAC Symposium on intelligent Autonomous Vechicles*, pages 1–5, Toulouse, France, September 2007.

[4] J. Belikov, K. Vassiljeva, E. Petlenkov, and S. Nõmm. A novel Taylor series based approach for control computation in NN-ANARX structure based control of nonlinear systems. In *The 27th Chinese Control Conference (CCC2008)*, pages 474–478, Kunming, China, July 2008.

[5] S. A. Billings and M. B. Fadzil. The practical identification of systems with nonlinearities. In *The 7th IFAC/IFORS Symp. Identification Syst. Parameter Estimation*, pages 155–160, 1985.

[6] M. Buckland. *AI Techniques for Game Programming.* Premier Press, Cincinnati, Ohio, USA, 2002.

[7] D. Chen, Lee C. Giles, G. Z. Sun, H. H. Chen, Y. C. Lee, and M. W. Goudreau. Constructive learning of recurrent neural networks. In T. Petsche, S. Judd, and S. Hanson, editors, *Computational Learning Theory and Natural Learning Systems III.* Cambridge, MAMIT Press, 1993.

[8] S. Chen and S. A. Billings. Representations of nonlinear systems: The narmax model. *International Journal of Control*, 49(3):1013–1032, 1989.

[9] Y. P. Chen and J. S.Wang. A novel recurrent neural network with minimal representation for dynamic system identification. In *IEEE Int. Joint Conf. Neural Networks*, pages 849–854, Budapest, July 2004.

[10] F. Chowdhury, Ü. Kotta, and S. Nõmm. On realizability of neural networks-based input-output models. *Differential Equations and Applications*, 6:47–51, 2000.

[11] F. N. Chowdhury. Input-output modeling of nonlinear systems with time-varying linear models. *IEEE Transactions on Automatic Control*, 7:1355–1358, 2000.

[12] G. Cybenko. Approximations by super-positions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, (2):303–313, 1989.

[13] Dipankar Dasgupta and Douglas R. Mcgregor. Designing application-specific neural networks using the structured genetic algorithm. In *In Proceedings of the International Conference on Combinations of Genetic Algorithms and Neural Networks*, pages 87–96. IEEE Computer Society Press, 1992.

[14] B. De Schutter. Minimal state-space realization in linear system theory: An overview. *Special Issue on Numerical Analysis in the 20th Century vol.I: Approximation Theory*, 121(1-2):331–354, september 2000.

[15] M. Delgado and M. C. Pegalajar. A multiobjective genetic algorithm for obtaining the optimal size of a recurrent neural network for grammatical inference. *Pattern Recognition*, pages 1444–1456, 2005.

[16] U. Forssell and L. Ljung. Closed-loop identification revisited. *Automatica*, 35:1215–1244, 1999.

[17] M. Gevers. Identification for control: From the early achievements to the revival of experiment design. *European Journal of Control*, 11:1–18, 2005.

[18] E. G. Gilbert. Controllability and observability in multy-variable control systems. *SIAM Journal on Control*, 1(2):128–151, 1963.

[19] C. L. Giles, D. Chen, G. Z. Sun, H. H. Chen, Y. C. Lee, and M. W. Goudreau. Constructive learning of recurrent neural networks: limitations of recurrent cascade correlation and a simple solution. *IEEE Transactions on Neural Networks*, 6(4):829–836, 1995.

[20] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley Professional, New York,USA, 1989.

[21] R.P. Guidorzi, M. P. Losito, and T. Muratory. The range error test in the structural identification of linear multivariable systems. *IEEE Transactions on Automatic Control*, 27(5):1044–1054, October 1982.

[22] R. Haber and H. Unbehauen. Structure identification of nonlinear dynamic systems - a survay of input/output approaches. *Automatica*, 26:651–677, 1990.

[23] M. T. Hagan and M. B. Menhaj. Training feedforward networks with the Marquadt algorithm. *IEEE Transactions on Neural Networks*, 5(6):989–993, November 1994.

[24] P. J. B. Hancock. Genetic algorithms and permutation problems: A comparison of recombination operators for neural net structure specification. In *Int. Workshop Combinations Genetic Algorithms Neural Networks*, pages 108–122, Los Alamitos, CA, June 1992.

[25] K.M. Hangos, J. Bokor, and G. Szederkenyi. *Analysis and Control of Nonlinear Process Systems.* Springer-Verlag, London, 2004.

[26] T. Hrycej. *Neurocontrol: Towards an Industrial Control Methodology.* John Wiley, New York, NY, 1997.

[27] K. J. Hunt, G. R. Irwin, and K. Warwick, editors. *Neural network engineering in dynamic control systems.* Springer, Great Britain, 1995.

[28] The MathWorks Inc. Documentation: System identification toolbox. `http://www.mathworks.com/help/toolbox/ident/ug/bq5o_xw-1.html`, July 2011.

[29] G. R. Irwin, K. Warwick, and K. J. Hunt, editors. *Neural network applications in control.* Institution of Electrical Engineers, London, 1995.

[30] R. E. Kalman. Mathematical description of linear dynamical systems. *SIAM Journal on Control*, 1(2):152–192, 1963.

[31] H. K. Khalil. *Nonlinear Systems.* Prentice Hall, Englewood Clis, NJ, 2nd edition, 1995.

[32] T. Knapp and H. Budman. Robust control design of non-linear processes using empirical state-affine models. *International Journal of Control*, (73):1525–1535, 2000.

[33] C. G. Koh, B. Hong, and C. Y. Liaw. Substructural and progressive structural identification methods. *Engineering Structures*, 25:1551–1563, 2003.

[34] Ü. Kotta, F. Chowdhury, and S. Nõmm. On realizability of neural networks-based input-output models in the classical state space form. *Automatica*, 42(7):1211–1216, 2006.

[35] Ü. Kotta, S. Nõmm, and F. Chowdhury. On a new type of neural network-based input-output model: The ANARMA structure. In *The 5th IFAC Symposium on nonlinear control systems NOLCOS*, pages 1535–1538, St. Petersbourg, Russia, July 2001.

[36] Ü. Kotta, S. Nõmm, and A.S.I. Zinober. Classical state space realizability of input-output bilinear models. *International Journal on Control*, 76(12):1224–1232, 2003.

[37] Ü. Kotta and N. Sadegh. Two approaches for state space realization of narma models: bridging the gap. *Mathematical and Computer Modelling of Dynamical Systems*, 8(1):21–32, 2002.

[38] I. J. Leontaritis and S. A. Billings. Inputoutput parametric models for nonlinear systems part i: Deterministic nonlinear systems. *International Journal of Control*, 41(2):303–328, 1985.

[39] X. Li, Y. Bai, and L. Yang. Neural network online decoupling for a class of nonlinear system. In *The 6th World Congress on Intelligent Control and Automation*, pages 2920–2924, Dalian, China, June 2006.

[40] B. T. Luke. Genetic algorithms and beyond. In R. Leardi, editor, *Data Handling in Science and Technology: Nature-inspired Methods in Chemometrics: Genetic Algorithms and Artificial Neural Networks*, pages 3–54. Elsevier, 2003.

[41] J. Mikleš and M. Fikar. *Process Modeling, Identification, and Control.* Springer, Berlin, Heidelberg, 2007.

[42] C. Miller and C. Giles. Experimental comparison of the effect of order in recurrent neural networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(4):849–872, 1993.

[43] K. Murase, S. Usui, and N. Toda. Structural identification algorithms of nonlinear systems using volterra functional series. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 2711–2714, Tokyo, Japan, April 1986.

[44] K. S. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Trans. on Neural Networks*, 1(1):4–27, march 1990.

[45] O. Nelles. *Nonlinear System Identification.* Springer, Berlin, Heidelberg, 2001.

[46] W.B. Nelson. *Accelerated Testing: Statistical Models, Test Plans, and Data Analysis.* John Wiley & Sons, Inc, Hoboken, New Jersey, USA, second edition, 2004.

[47] S. Nõmm and Ü. Kotta. Comparison of neural networks-based ANARX and NARX models by application of correlation tests. In *2011 Internationa Joint Conference on Neural Networks*, pages 2113–2118, San Jose, California, July 31 - August 5 2011.

[48] S. Nõmm, E. Petlenkov, J. Vain, K. Yoshimitsu, K. Ohnuma, T. Sadahiro, and F. Miyawaki. NN-based ANARX model of the surgeons hand for the motion recognition. In *The 4th COE Workshop on Human Adaptive Mechatronics (HAM)*, pages 19–24, Tokyo Denki University, Tokyo, Japan, March 2007.

[49] S. Nõmm, K. Vassiljeva, J. Belikov, and E. Petlenkov. Structure identification of NN-ANARX model by genetic algorithm with combined cross-correlation-test based evaluation function. In *Proceedings of the 9th IEEE International Conference on Control and Automation*, pages 65–70, Santiago, Chile, December 2011.

[50] S. Nõmm, K. Vassiljeva, and E. Petlenkov. Evaluation function optimization for the genetic algorithm based tuning of NN-ANARX model structure. In *WCCI 2012 IEEE World Congress on Computational Intelligence: IJCNN*, Brisbane, Australia, June 2012. Accepted.

[51] R. K. Pearson and Ü. Kotta. Nonlinear discrete-time models: state-space vs. i/o representations. *Journal of Process Control*, (14):533–538, 2004.

[52] E. Petlenkov. NN-ANARX structure based dynamic output feedback linearization for control of nonlinear MIMO systems. In *The 15th Mediterranean Conference on Control and Automation*, pages 1–6, Athena, Greece, June 2007.

[53] E. Petlenkov and J. Belikov. NN-ANARX structure for dynamic output feedback linearization of nonlinear SISO and MIMO systems: Neural networks based approach. In *The 26th Chinese Control Conference (CCC2007)*, pages 1806–1811, Zhangjiajie, China, July 2007.

[54] E. Petlenkov, S. Nõmm, and Ü. Kotta. NN-based ANARX structure for identification and model-based control. In *The 9th International Conference on Control Automation Robotics and Vision*, pages 2284–2288, Singapore, December 2006.

[55] R. Pothin, Ü. Kotta, and C. Moog. Output feedback linearization of nonlinear discrete-time systems. In *Proc. of the IFAC Conf. on Control system design*, pages 174–179. Bratislava, 2000.

[56] N. J. Radcliffe. Genetic set recombination and its application to neural network topology optimization. Technical Report EPCC-TR 21, University of Edinburg, Edinburg, Scotland, UK, 1991.

[57] Wolfram Research. Wolfram mathworld. `http://mathworld.wolfram.com/ConfidenceInterval.html`, October 2011.

[58] N. Sadegh. Minimal realization of nonlinear systems described by inputoutput difference equations. *IEEE Transactions on Automatic Control*, 46(5):698–710, may 2001.

[59] R. S. Sánchez-Peña, J. Q. Casin, and V. P. Cayuela, editors. *Identification and Control : The Gap Between Theory and Practice*. Springer, London, 2007.

[60] J. D. Shaffer, D. Whitley, and L. J. Eshelman. Combinations of genetic algorithms and neural networks: A survey of the state of the art. In *Int. Workshop Combinations Genetic Algorithms Neural Networks*, pages 1–37, Los Alamitos, CA, June 1992.

[61] H. L. Shu. Pid neural network for analysis of the multivariable systems. *Automation transaction*, 25(1):105–111, 1999.

[62] S. N. Sivanandam and S.N. Deepa. *Introduction to Genetic Algorithms*. Springer, Berlin, Heidelberg, 2010.

[63] J.S. Son, D.M. Lee, I.S. Kim, and S.K. Choi. A study on genetic algorithm to select architecture of a optimal neural network in the hot rolling process. *Journal of Materials Processing Technology*, 153-154:643–648, 2004.

[64] F. Song and P. Li. MIMO decoupling control based on support vector machines αth-order inversion. In *The 6th World Congress on Intelligent Control and Automation*, pages 1002–106, Dalian, China, June 2006.

[65] K. Stanley and R. Miikkulainen. Efficient reinforcement learning through evolving neural network topologies. In *Genetic and Evolutionary Computation Conference (GECCO-2002)*, San Francisco, CA, USA, 2002.

[66] H. J. Sussman. Existence and uniqueness of minimal realizations of nonlinear systems. *Math. Syst. Theory*, 10:263–284, 1977.

[67] J. Van de Vegte. *Feedback Control Systems.* Prentice Hall, Englewood Cliffs, NJ, 3rd edition, 1994.

[68] K. Vassiljeva, J. Belikov, and E. Petlenkov. Genetic algorithm based structure identification for feedback control of nonlinear MIMO systems. In Abdelhamid Bouchachia, editor, *Adaptive and Intelligent Systems : Second International Conference, ICAIS 2011, Klagenfurt, Austria, September 6-8, 2011, Proceedings*, pages 215–226. Springer, Lecture Notes in Artificial Intelligence, 2011.

[69] K. Vassiljeva, J. Belikov, and E. Petlenkov. Neural network based minimal or reduced model representation for control of nonlinear MIMO systems. In *Proceedings of International Joint Conference on Neural Networks*, pages 1706–1713, San Jose, CA, USA, July 31 - August 5 2011.

[70] K. Vassiljeva, E. Petlenkov, and J. Belikov. Automated neural network model selection algorithm for feedback linearization based control. In *2010 12th Biennial Baltic Electronics Conference (BEC 2010)*, pages 235–238, Tallinn, October 2010.

[71] K. Vassiljeva, E. Petlenkov, and J. Belikov. Neural network based minimal state-space representation of nonlinear MIMO systems for feedback control. In *2010 the 11th International Conference on Control Automation, Robotics and Vision*, pages 2191–2196, Singapore, 7-10 December 2010.

[72] K. Vassiljeva, E. Petlenkov, and J. Belikov. State-space control of nonlinear systems identified by ANARX and neural network based SANARX models. In *WCCI 2010 IEEE World Congress on Computational Intelligence: IJCNN*, pages 3816–3823, Barcelona, Spain, July 2010.

[73] C. Wang and D.J. Hill. *Deterministic Learning Theory for Identification, Recognition, and Control.* CRC Press, Boca Raton, FL, 1st edition, 2010.

[74] J. S. Wang and Y. P. Chen. A fully automated recurrent neural network for unknown dynamic system identification and control. *IEEE Transactions on Circuits and Systems-I*, 56(6):1363–1372, 2006.

[75] D. Whitley. Genetic algorithms and neural networks. In *Genetic Algorithms in Engineering and Computer Science*, pages 191–201. John Wiley, 1995.

[76] K. Worden and G. R. Tomlinson. *Nonlinearity in structural dynamics: detection, identification, and modelling.* Taylor and Francis, The Institute of Physics, London, 1 edition edition, 2001.

[77] Xin Yao. Evolving artificial neural networks. *Proceedidngs of the IEEE*, 87(9):1423–1447, September 1999.

[78] W. Yu and A.S. Poznyak. Indirect adaptive control via parallel dynamic neural networks. In *IEE Proc-Control Theory Appl*, volume 146, pages 25–30, january 1999.

[79] S. Zenieh and A. L. Elshafei. Robust control of uncertain nonlinear mechanical systems using a high gain observer. In *American Control Conference*, pages 3620–3625, Chicago, IL,USA, June 2000.

[80] L. F. Zhang, Q. M. Zhu, and A. Longden. Development of omnidirectional correlation functions for nonlinear model validation. *Automatica*, 43:1519–1531, 2007.

[81] L. F. Zhang, Q. M. Zhu, and A. Longden. A correlation-tast-based validation procedure for identified neural networks. *IEEE Transactns on Neural Networks*, 20(1):1–13, 2009.

[82] Y. Zhu. *Multivariable System Identification For Process Control.* Elsevier Science, Oxford, UK, 1st edition, 2001.

List of Publications

P[1] K. Vassiljeva, E. Petlenkov, and J. Belikov. State-space control of nonlinear systems identified by ANARX and neural network based SANARX models. In *WCCI 2010 IEEE World Congress on Computational Intelligence: IJCNN*, pages 3816–3823, Barcelona, Spain, July 2010.

P[2] K. Vassiljeva, E. Petlenkov, and J. Belikov. Automated neural network model selection algorithm for feedback linearization based control. In *2010 12th Biennial Baltic Electronics Conference (BEC 2010)*, pages 235–238, Tallinn, October 2010.

P[3] K. Vassiljeva, E. Petlenkov, and J. Belikov. Neural network based minimal state-space representation of nonlinear MIMO systems for feedback control. In *2010 the 11th International Conference on Control Automation, Robotics and Vision*, pages 2191–2196, Singapore, 7-10 December 2010.

P[4] K. Vassiljeva, J. Belikov, and E. Petlenkov. Neural network based minimal or reduced model representation for control of nonlinear MIMO systems. In *Proceedings of International Joint Conference on Neural Networks*, pages 1706–1713, San Jose, CA, USA, July 31 - August 5 2011.

P[5] K. Vassiljeva, J. Belikov, and E. Petlenkov. Genetic algorithm based structure identification for feedback control of nonlinear MIMO systems. In Abdelhamid Bouchachia, editor, *Adaptive and Intelligent Systems : Second International Conference, ICAIS 2011, Klagenfurt, Austria, September 6-8, 2011, Proceedings*, pages 215–226. Springer, Lecture Notes in Artificial Intelligence, 2011.

P[6] S. Nõmm, K. Vassiljeva, J. Belikov, and E. Petlenkov. Structure identification of NN-ANARX model by genetic algorithm with combined cross-correlation-test based evaluation function. In *Proceedings of the 9th IEEE International Conference on Control and Automation*, pages 65–70, Santiago, Chile, December 2011.

Kokkuvõtte

Käesoleval ajal tootmisprotsesside automatiseerimisel kasutatavate kontrollerite võimalused (eriti arvutusvõimsused) on oluliselt kasvanud. Automaatikavahendeid tootvad firmad on laiendanud kontrollereid funktsionaalsete plokkidega, luues eeldused mudelipõhiste juhtimismeetodite laialdaseks kasutamiseks tootmisprotsesside automatiseerimisel.

Paljud juhtimissüsteemide modelleerimisel, identifitseerimisel ja analüüsil kasutatavad meetodid põhinevad olekumudelil. Mittelineaarsete süsteemide identifitseerimisel kasutatakse tavaliselt sisend - väljund mudelitel põhinevaid meetodeid. Üheks võimaluseks klassikaliste ja kaasaegsete meetodite integreerimiseks on kasutada eristruktuuriga tehisnärvivõrke.

Mittelineaarsete süsteemide identifitseerimisel on otstarbekas kasutada tehisnärvivõrkudel põhinevat ANARX mudelit, kuna täielikult ühendatud närvivõrgud ei ole alati esitatavad olekumudelina. ANARX mudel võimaldab leida olekumudeli ilma täiendavate kitsendusteta. Käesolevas väitekirjas on uuritud mittelineaarsete süsteemide identifitseerimist juhtimiseks tehisnärvivõrkude abil.

Mitmemõõtmeliste süsteemide minimaalse olekumudeli leidmiseks tuleb teha muudatusi tehisnärvivõrgu arhitektuuris. Antud töös on uuritud kahte inseneripraktika vaatepunktist olulist identifitseerimise juhtumit:

1. Halli kasti (gray-box) lähenemine (eeldame, et identifitseeritava süsteemi järk on teada);

2. Musta kasti (black-box) lähenemine.

Halli kasti lähenemise puhul saame täpsustada vajalikke tehisnärvivõrgu struktuuri muudatusi, kuid tulemus sõltub olulisel määral eksperdi kogemustest.

Musta kasti lähenemisel tuleb erinevatest piiratud ühenduvusega tehisnärvivõrkude arhitektuuridest leida ANARX mudeli alaliik, mis kindlustab närvivõrgu parameetrite põhjal loodud regulaatorile nõutud juhtimise kvaliteedi. Töös väljatöötatud meetod võimaldab leida tehisnärvivõrgu optimaalse arhitektuuri mittelineaarse süsteemi identifitseerimiseks. Meetod põhineb mitmekriteriaalse hindamisfunktsiooni ja geneetilise algoritmi kasutamisel. On oluline märkida, et meetod lahendab ka mudeli üleparametriseerimise probleemi.

Töös väljatöötatud meetodeid on testitud inseneripraktikas oluliste objektide mudelitel.

Abstract

Nowadays, the performance capabilities of controllers used in industry has increased significantly. Many manufacturers include in their production series Function Blocks, which allow to use model-based control techniques.

A lot of different methods for the analysis, modeling and control design based on the classical state-space model. However, for the identification of nonlinear systems techniques based on input-output models are usually used. One possibility to combine classical and modern approaches is the use of neural network with specific structure.

Fully connected neural networks are not always directly realizable in the state-space representation. In this case NN-based Additive NARX (Nonlinear AutoRegressive eXogenious) model is considered as a reasonable choice for control-aimed identification of a wide class of nonlinear systems. It allows to obtain state-space representation without additional assumptions on the structure of the identified model. The identification process for control using neural networks for further application in state-space controller is studied in this thesis.

However, in the case of MIMO systems, finding the minimal state-space representation requires some changes in the neural network architecture. The major contributions of the present work are made for two cases: when we deal with a gray-box system (order of the identified system is known a priori) or with a black-box system. In the first case, we can specify the structural changes that have to be done, so the result strongly depends on the expert experience. In the second case, from the different restricted connectivity neural networks architectures it is necessary to find ANARX type subclass, such as the controller based on the parameters of a chosen neural network has a good performance.

A method for deriving the optimal structure of a neural network using genetic algorithm with multicriteria evaluation function is developed. As a result of which a neural network, where all redundant interconnections

are eliminated, is obtained. That approach allows us to solve the model over-parameterization problem. On the basis of neural network parameters a dynamic state feedback controller is designed.

Elulookirjeldus

1. Isikuandmed

Ees- ja perekonnanimi	Kristina Vassiljeva
Sünniaeg ja -koht	03.01.1979, Tallinn, Eesti
Kodakondsus	Eesti

2. Kontaktandmed

Aadress	Ehitajate tee 5, A-II-320, Tallinn, 19086
Telefon	+372 620 2116
E-posti aadress	kristina.vassiljeva@dcc.ttu.ee

3. Hariduskäik

Õppeasutus (nimetus lõpetamise ajal)	Lõpetamise aeg	Haridus (eriala/kraad)
Tallinna Tehnikaülikool	2001	arvuti- ja süsteemitehnika, B.Sc.
Tallinna Tehnikaülikool	2003	arvuti- ja süsteemitehnika, M.Sc.

4. Keelteoskus

Keel	Tase
Eesti	kõrgtase
Inglise	kõrgtase
Saksa	algtase
Vene	emakeel

5. Täiendusõpe

Õppimise aeg	Õppeasutus või muu organisatsiooni nimetus
2012	”Introduction to Nonlinear Control”, European Embedded Control Institute, Supelec, Pariis, Prantsusmaa
2011	”Metso DNA basic and engineering course ”How to teach technical engineering””, Metso Automation, Tampere, Soome
2011	”Normal Forms for Nonlinear Control Systems”, European Embedded Control Institute, Supelec, Pariis, Prantsusmaa
2005	”Kõrgkoolididaktika”, Tallina Tehnikaülikool, Tallinn, Eesti
2003	”Academic Scientific Writing”, Tallina Tehnikaülikool, Tallinn, Eesti
2003	”Education of Highly Qualified Information Technology Specialists for Central and Eastern European Countries”, Japan International Cooperation Agency, Polish-Japanese Institute of Information Technology, Varssava, Poola

6. Teenistuskäik

Töötamise aeg	Tööandja nimetus	Ametikoht
2002 – . . .	Automaatikainstituut, TTÜ	assistent
01.2002 – 08.2002	Automaatikainstituut, TTÜ	insener

7. Teadustegevus

   Ajakirja- ja konverentisartiklite loetelu on toodud ingliskeelse CV juures.

8. Kaitstud lõputööd

   Tehisnärvivõrgupõhise ennustava regulaatori uurimine, B.Sc., Tallinna Tehnikaülikool, 2001.

   Mittelineaarsete süsteemide adaptiivjuhtimine tehisnärvivõrkudega, M.Sc., Tallinna Tehnikaülikool, 2003.

9. Teadustöö põhisuunad

   Mittelineaarsete juhtimissüsteemide teooria, tehisnärvivõrgud, geneetilised algoritmid

Curriculum vitae

1. Personal data

Name	Kristina Vassiljeva
Date and place of birth	03.01.1979, Tallinn, Estonia

Address	Ehitajate tee 5, A-II-320, Tallinn, 19086
Phone	+372 620 2116
E-mail	kristina.vassiljeva@dcc.ttu.ee

2. Education

Educational institution	Graduation year	Education (field of study/degree)
Tallinn University of Technology	...	Information Technology PhD student
Tallinn University of Technology	2003	Computer and Systems Engineering, M.Sc.
Tallinn Technical University	2001	Computer and Systems Engineering, B.Sc.

3. Language competence/skills

Language	Level
Estonian	fluent
English	fluent
German	poor
Russian	native

4. Special Courses

2012	"Introduction to Nonlinear Control", European Embedded Control Institute, Supelec, Paris, France
2011	"Metso DNA basic and engineering course "How to teach technical engineering"", Metso Automation, Tampere, Finland
2011	"Normal Forms for Nonlinear Control Systems", European Embedded Control Institute, Supelec, Paris, France
2005	"Didactics in Higher Engineering Education", Tallinn University of Technology, Tallinn, Estonia
2003	"Academic Scientific Writing", Tallinn University of Technology, Tallinn, Estonia
2003	"Education of Highly Qualified Information Technology Specialists for Central and Eastern European Countries", Japan International Cooperation Agency, Polish-Japanese Institute of Information Technology, Warsaw, Poland

5. Professional Employment

Period	Organization	Position
01.2002 – 08.2002	Department of Computer Control, TUT	engineer
2002 – ...	Department of Computer Control, TUT	teaching assistant

6. Scientific work

    1. V. Vansovits, and K. Vassiljeva E. Petlenkov, and A. Guljajev. Identification of industrial water boiler for model predictive control of district heat plant. In *2012 13th Biennial Baltic Electronics Conference (BEC 2012)*, Tallinn, October 2012. Accepted.

    2. S. Nõmm, K. Vassiljeva, and E. Petlenkov. Evaluation function optimization for the genetic algorithm based tuning of NN-ANARX model structure. In *WCCI 2012 IEEE World Congress on Computational Intelligence: IJCNN*, Brisbane, Australia, June 2012. Accepted.

    3. K. Vassiljeva, E. Petlenkov, and J. Belikov. GA based optimization of NN-SANARX model for adaptive control of nonlinear systems. In *WCCI 2012 IEEE World Congress on Computational Intelligence: IJCNN*, Brisbane, Australia, June 2012. Accepted.

4. S. Nõmm, K. Vassiljeva, J. Belikov, and E. Petlenkov. Structure identification of NN-ANARX model by genetic algorithm with combined cross-correlation-test based evaluation function. In *Proceedings of the 9th IEEE International Conference on Control and Automation*, pages 65–70, Santiago, Chile, December 2011.

5. K. Vassiljeva, E. Petlenkov, and J. Belikov. Neural networks based identification for control. In *Info- ja kommunikatsioonitehnoloogia doktorikooli IKTDK viienda aastakonverentsi artiklite kogumik : 25.-26.novembril 2011, Nelijärve*, pages 157–160, Tallinn, November 2011.

6. K. Vassiljeva, J. Belikov, and E. Petlenkov. Genetic algorithm based structure identification for feedback control of nonlinear MIMO systems. In Abdelhamid Bouchachia, editor, *Adaptive and Intelligent Systems : Second International Conference, ICAIS 2011, Klagenfurt, Austria, September 6-8, 2011, Proceedings*, pages 215–226. Springer, Lecture Notes in Artificial Intelligence, 2011.

7. K. Vassiljeva, J. Belikov, and E. Petlenkov. Neural network based minimal or reduced model representation for control of nonlinear MIMO systems. In *Proceedings of International Joint Conference on Neural Networks*, pages 1706–1713, San Jose, CA, USA, July 31 - August 5 2011.

8. K. Vassiljeva, E. Petlenkov, and J. Belikov. Neural network based minimal state-space representation of nonlinear MIMO systems for feedback control. In *2010 the 11th International Conference on Control Automation, Robotics and Vision*, pages 2191–2196, Singapore, 7-10 December 2010.

9. K. Vassiljeva, E. Petlenkov, and J. Belikov. Towards genetic algorithm based structure identification for minimal state-space representation and feedback control of nonlinear MIMO systems. In *Info- ja kommunikatsioonitehnoloogia doktorikooli IKTDK neljanda aastakonverentsi artiklite kogumik : 26.-27.novembril 2010, Essu Mõis*, pages 11–14, Tallinn, November 2010.

10. K. Vassiljeva, E. Petlenkov, and J. Belikov. Automated neural network model selection algorithm for feedback linearization based control. In *2010 12th Biennial Baltic Electronics Conference (BEC 2010)*, pages 235–238, Tallinn, October 2010.

11. K. Vassiljeva, E. Petlenkov, and J. Belikov. State-space control of nonlinear systems identified by ANARX and neural network

based SANARX models. In *WCCI 2010 IEEE World Congress on Computational Intelligence: IJCNN*, pages 3816–3823, Barcelona, Spain, July 2010.

12. J. Belikov, K. Vassiljeva, E. Petlenkov, and S. Nõmm. A novel taylor series based approach for control computation in NN-ANARX structure based control of nonlinear systems. In *The 27th Chinese Control Conference (CCC2008)*, pages 474–478, Kunming, China, July 2008.

13. K. Vassiljeva and E. Rüstern. On-line identification and adaptive control of nonlinear systems using neural networks. In *2004 The 9th Biennial Baltic Electronics Conference (BEC 2004)*, pages 149–152, Tallinn, October 2004.

7. Defended theses

   Research of the predictive controller based on artificial neural networks, B.Sc., Tallinn University of Technology, 2001.

   Adaptive control of nonlinear systems with Artificial Neural networks, M.Sc., Tallinn University of Technology, 2003.

8. Main areas of scientific work/Current research topics

   Nonlinear control systems, neural networks, genetic algorithms

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.

36. **Eduard Petlenkov**. Neural Networks Based Identification and Control of Nonlinear Systems: ANARX Model Based Approach. 2007.

37. **Toomas Kirt**. Concept Formation in Exploratory Data Analysis: Case Studies of Linguistic and Banking Data. 2007.

38. **Juhan-Peep Ernits**. Two State Space Reduction Techniques for Explicit State Model Checking. 2007.

39. **Innar Liiv**. Pattern Discovery Using Seriation and Matrix Reordering: A Unified View, Extensions and an Application to Inventory Management. 2008.

40. **Andrei Pokatilov**. Development of National Standard for Voltage Unit Based on Solid-State References. 2008.

41. **Karin Lindroos**. Mapping Social Structures by Formal Non-Linear Information Processing Methods: Case Studies of Estonian Islands Environments. 2008.

42. **Maksim Jenihhin**. Simulation-Based Hardware Verification with High-Level Decision Diagrams. 2008.

43. **Ando Saabas**. Logics for Low-Level Code and Proof-Preserving Program Transformations. 2008.

44. **Ilja Tšahhirov**. Security Protocols Analysis in the Computational Model - Dependency Flow Graphs-Based Approach. 2008.

45. **Toomas Ruuben**. Wideband Digital Beamforming in Sonar Systems. 2009.

46. **Sergei Devadze**. Fault Simulation of Digital Systems. 2009.

47. **Andrei Krivošei**. Model Based Method for Adaptive Decomposition of the Thoracic Bio-Impedance Variations into Cardiac and Respiratory Components. 2009.

48. **Vineeth Govind**. DfT-Based External Test and Diagnosis of Mesh-like Networks on Chips. 2009.

49. **Andres Kull**. Model-Based Testing of Reactive Systems. 2009.

50. **Ants Torim**. Formal Concepts in the Theory of Monotone Systems. 2009.

51. **Erika Matsak**. Discovering Logical Constructs from Estonian Children Language. 2009.

52. **Paul Annus**. Multichannel Bioimpedance Spectroscopy: Instrumentation Methods and Design Principles. 2009.

53. **Maris Tõnso**. Computer Algebra Tools for Modelling, Analysis and Synthesis for Nonlinear Control Systems. 2010.

54. **Aivo Jürgenson**. Efficient Semantics of Parallel and Serial Models of Attack Trees. 2010.

55. **Erkki Joasoon**. The Tactile Feedback Device for Multi-Touch User Interfaces. 2010.

56. **Jürgo-Sören Preden**. Enhancing Situation - Awareness Cognition and Reasoning of Ad-Hoc Network Agents. 2010.

57. **Pavel Grigorenko**. Higher-Order Attribute Semantics of Flat Languages. 2010.

58. **Anna Rannaste**. Hierarcical Test Pattern Generation and Untestability Identification Techniques for Synchronous Sequential Circuits. 2010.

59. **Sergei Strik**. Battery Charging and Full-Featured Battery Charger Integrated Circuit for Portable Applications. 2011.

60. **Rain Ottis**. A Systematic Approach to Offensive Volunteer Cyber Militia. 2011.

61. **Natalja Sleptšuk**. Investigation of the Intermediate Layer in the Metal-Silicon Carbide Contact Obtained by Diffusion Welding. 2011.

62. **Martin Jaanus**. The Interactive Learning Environment for Mobile Laboratories. 2011.

63. **Argo Kasemaa**. Analog Front End Components for Bio-Impedance Measurement: Current Source Design and Implementation. 2011.

64. **Kenneth Geers**. Strategic Cyber Security: Evaluating Nation-State Cyber Attack Mitigation Strategies. 2011.

65. **Riina Maigre**. Composition of Web Services on Large Service Models. 2011.

66. **Helena Kruus**. Optimization of Built-in Self-Test in Digital Systems. 2011.

67. **Gunnar Piho**. Archetypes Based Techniques for Development of Domains, Requirements and Sofware. 2011.

68. **Juri Gavšin**. Intrinsic Robot Safety Through Reversibility of Actions. 2011.

69. **Dmitri Mihhailov**. Hardware Implementation of Recursive Sorting Algorithms Using Tree-like Structures and HFSM Models. 2012.

70. **Anton Tšertov**. System Modeling for Processor-Centric Test Automation. 2012.

71. **Sergei Kostin**. Self-Diagnosis in Digital Systems. 2012.

72. **Mihkel Tagel**. System-Level Design of Timing-Sensitive Network-on-Chip Based Dependable Systems. 2012.

73. **Juri Belikov**. Polynomial Methods for Nonlinear Control Systems. 2012.