Neural Adaptive Control

Master's Thesis

A.J.N. van Breemen

January 28, 1997

Supervisors:	Ir. L.P.J. Veelenturf Prof. Dr. -Ing. O.E. Herrmann Dr. Ir. T.J.A. De Vries

Report no: BSC003N97
Period: Februari 1996 - Januari 1997

University of Twente (The Netherlands)
Research Group BSC
Laboratory of Network Theory
Department of Electrical Engineering

Abstract

Most controller design methods are *model-based*. This means that a model of the process which has to be controlled is needed. It is not always easy and/or possible to obtain a model, especially when:

- the process is too complex
- the process contains uncertainties
- there is no time for building a model
- there is no knowledge for building a model

Therefore another controller design method is needed which does not need a model of the process. Such a design method is called a *model-free controller design method.*

During the early 1990's it has been recognized that neural networks can serve as a basis for a model-free controller design method. A lot of simulation results were published using neural networks for control, which showed that neural networks are indeed capable of controlling complex processes without using a model of the process. These, so called, *neural controllers* were lacking systematics; it was not always motivated why a particular neural controller structure was used.

The basis of a good control design method is systematics and a mathematical foundation. In this report an attempt has been made to systematize *neural adaptive controllers*, which are a subset of the neural controllers. The basis of this systematization is the point of view that a neural adaptive controller consists of two parts: a controller and an adaptation mechanism. In general, both parts can be designed independently. Furthermore a contribution has been made to the mathematical foundation of neural adaptive controllers. This contribution reaches a climax with the proposal of two stability proofs of different types of neural adaptive controllers.

Beside theoretical issues of neural adaptive controllers, this report also discusses practical implementation issues. Two neural networks commonly used are the *Multi Layer Perceptron* (MLP) and the *Radial Basis Function Network* (RBFN). It will be shown that by making use of some properties of the RBFN, this network can be very efficiently implemented. This efficiency is obtained by recognizing the fact that under certain conditions the number of calculations to compute the response of the RBFN is independent of the number of basis functions used.

Contents

Abstract iii

Preface xiii

Abbreviations and Symbols xv

1 Introduction 1

1.1 Neural Adaptive Control . . . 1
1.2 The scope of this report . . . 2

2 Developments in Automatic Control 5

2.1 A Brief History of Automatic Control . . . 5
2.1.1 Early Control: to 1900 . . . 5
2.1.2 The Pre-Classical Period: 1900-1940 . . . 7
2.1.3 The Classical Period: 1935-1950 . . . 8
2.1.4 Modern Control . . . 9
2.2 The Classical Controller Design Methodology . . . 10
2.2.1 Systems . . . 10
2.2.2 Signals . . . 12
2.2.3 Controller Design . . . 13
2.3 Intelligent control . . . 16

3 Artificial Neural Networks 19

3.1 Neuron models . . . 19
3.2 Architectures . . . 22
3.3 Multilayer Perceptron . . . 26
3.3.1 Backpropagation . . . 27
3.4 Radial Basis Function Networks . . . 32
3.4.1 Implementing RBFNs - some practical tips . . . 34
3.4.2 RBFN for control . . . 36

4 Neural Adaptive Control 39

4.1 Neural Control . . . 40
4.2 Introduction to Adaptive Control Methodologies . . . 42
4.2.1 Model Reference Adaptive Control . . . 44
4.2.2 Self-tuning regulator . . . 46
4.2.3 Gain scheduling . . . 46
4.3 NAC Design Methodology . . . 47
4.4 System Descriptions . . . 47
4.4.1 Continuous time I/O model . . . 48
4.4.2 Discrete time I/O models . . . 49

4.4.3 Continuous time state space model . . . 49
4.4.4 Discrete time state space model . . . 50
4.4.5 From continuous to discrete time state space model . . . 50
4.4.6 From discrete time state space to discrete time I/O model . . . 51
4.5 Controller structure . . . 54
4.5.1 Series inverse control . . . 55
4.5.2 Parallel inverse control . . . 59
4.5.3 Parallel corrective control . . . 60
4.5.4 Feedback linearization . . . 60
4.6 Adaptation rules . . . 62
4.6.1 Direct adaptation . . . 63
4.6.2 Direct sensitivity adaptation . . . 64
4.6.3 Indirect inverse adaptation . . . 65
4.6.4 Indirect sensitivity adaptation . . . 65
4.6.5 Indirect update . . . 66
4.7 Stability . . . 68
4.7.1 Stability proof of series inverse controller for continuous time nonlinear SISO systems . . . 68
4.7.2 Stability proof of Neural Adaptive Feedback Linearization Controller for continuous time nonlinear SISO systems . . . 71
4.8 Parameter Convergence . . . 74

5 Experiments **77**
5.1 NAC of continuous time SISO system . . . 77
5.1.1 Linear state feedback . . . 78
5.1.2 Series inverse control . . . 79
5.1.3 Parallel corrective control . . . 79
5.1.4 PD control . . . 79
5.1.5 Parallel inverse control. . . . 84
5.1.6 Discussion . . . 84
5.2 NAFL control of inverted pendulum . . . 87
5.2.1 Derivation of the nonlinear difference equations of the inverted pendulum problem . . . 87
5.2.2 Neural adaptive feedback linearization control . . . 88
5.2.3 Discussion . . . 90
5.3 NAC of discrete I/O system using update adaptation . . . 90
5.3.1 Linear state feedback of system Σ_1 . . . 90
5.3.2 series inverse control of system Σ_1 . . . 91
5.3.3 Parallel corrective control of system Σ_1 . . . 91
5.3.4 Parallel inverse control of system Σ_1 . . . 93
5.3.5 Parallel corrective control of system Σ_2 . . . 93
5.3.6 Discussion . . . 97
5.4 NAC using indirect inverse adaptation . . . 99
5.4.1 Parallel corrective control of system Σ_1 . . . 99
5.4.2 Discussion . . . 99
5.5 NAC using MLP . . . 99
5.5.1 Serie inverse control of system Σ_1 using the MLP . . . 101
5.5.2 Parallel corrective control of system Σ_1 using the MLP . . . 101
5.5.3 Discussion . . . 101

6 Conclusions, discussion & recommendations **105**
6.1 Conclusions . . . 105
6.2 Has NAC something to offer? . . . 106
6.3 When to use NAC? . . . 106
6.4 Future developments . . . 107
6.5 Future research . . . 108

A Mathematics **109**
A.1 Sets . . . 109
A.1.1 Operations . . . 110
A.1.2 Properties . . . 110
A.1.3 Relations . . . 111
A.1.4 Intervals . . . 112
A.1.5 Maximum and minimum . . . 113
A.1.6 Majorant, minorant . . . 113
A.1.7 Supremum, infimum . . . 113
A.2 Functions . . . 113
A.3 Linear vector space . . . 115
A.4 Matrices and vectors . . . 120
A.5 Differential equations . . . 122
A.5.1 Solution, existence, uniqueness . . . 123
A.5.2 Linear, homogene, autonome . . . 124
A.5.3 State space, equilibrium and stability . . . 124

B Matlab/Simulink sources **129**
B.1 rbfn11.m . . . 129
B.2 rbfn21.m . . . 129
B.3 rbfn31.m . . . 130

C C/C++ sources **131**
C.1 rbfn.hpp . . . 131
C.2 rbfn.cpp . . . 133
C.3 expm3x.cpp . . . 141
C.4 expm4x.cpp . . . 143
C.5 expm5x.cpp . . . 144
C.6 mlp.hpp . . . 145
C.7 mlp.cpp . . . 146
C.8 expm6x.cpp . . . 153
C.9 expm7x.cpp . . . 154

Bibliography **157**

Index **160**

List of Figures

0.1 Block diagram symbols: (a) $y = Su$, (b) $\epsilon_2 = \frac{\partial y}{\partial u}\epsilon_1$, (c) $S = S(.|\mathbf{w}), \dot{\mathbf{w}} = \Gamma(\epsilon)$, (d) $\epsilon_2 = S\epsilon_1$, (e) $\epsilon_2 = S^{-1}\epsilon_1$, (f) $S_1 = S(.|\mathbf{w}_1), S_2 = S(.|\mathbf{w}_2), \mathbf{w}_1 = \mathbf{w}_2$. xvi

2.1 Hero's device for opening temple doors. 6
2.2 Watt's steam engine governor. 7
2.3 Antiaircraft radar-tracking control system. 9
2.4 The system-view. 11
2.5 Solar-heated house. 11
2.6 System-view solar-heated house. 12
2.7 General controller diagram 14
2.8 Diagram of development of system complexity, control objective and the mathematical complexity of the controller design. 17

3.1 General schematic diagram of a neuron. 20
3.2 Activation functions: (a) threshold, (b) signum, (c) piecewise linear, (d) unipolar sigmoid, (e) bipolar sigmoid, (f) gaussian. 22
3.3 Examples of neuron operation functions: (a) linear integration and unipolar sigmoid activation function ($w_1 = 1.0, w_2 = 3.0, \theta = 3.0$), (b) spherical integration and unipolar sigmoid activation function ($w_1 = 1.0, w_2 = 0.0, \sigma^{-2} = 0.4, \theta = -3$), (c) elliptical integration and unipolar activation function ($w_1 = 0, w_2 = 0.5, \theta = 2, \mathbf{C} = (0.7, 0.42; 0.42, 0.7)$), (d) quadratic integration and exponential activation function ($w_1 = 0.3, w_2 = 0.3, \theta = 0.0$). 23
3.4 Example of neuron as match filter: (a) Schematic diagram of neuron as match filter, notice the use of a tapped delay line, (b) puls to be detected, (c) 15 weights values of the neuron, (d) signal containing puls (a) at $t = 50$ and normal distributed noise, (e) output neuron, notice maximum at $t = 50$, puls has been located. 24
3.5 Four architectures of ANN: (a) perceptron, (b) single layer feedforward network, (c) multi layer feedforward network, (d) recurrent network 25
3.6 Multilayer perceptron. 26
3.7 Summary: Backpropagation (pattern mode training) 31
3.8 Radial Basis Function Network. 32
3.9 1-7-1 RBF. 33
3.10 Basis function placement to efficiently calculate the response of the RBFN and the weight adaptation. 35
3.11 Summary: Fast RBFN implementation. 37

4.1 Reinforcement learning control scheme. The plant is directly controlled without having a model of the plant 41
4.2 Internal model control. 41
4.3 Model predictive control. 42
4.4 Model Reference Adaptive System 44

4.5 Self-tuning regulator 46
4.6 Gain scheduling 46
4.7 The five stage during NAC design. 48
4.8 Summation: Transformation from discrete time SISO state space model to discrete time SISO I/O model 53
4.9 Series inverse control. 55
4.10 Summary: series inverse control for SISO nonlinear discrete time I/O systems. 57
4.11 Summary: series inverse control for MIMO nonlinear discrete time I/O systems. 58
4.12 Parallel inverse control. 60
4.13 Parallel corrective control. 61
4.14 Neural feedback linearization controller. 62
4.15 Direct adaptation. 63
4.16 Direct sensitivity adaptation. 65
4.17 Indirect inverse adaptation. 65
4.18 Indirect sensitivity adaptation. 66
4.19 Indirect inverse update. 67
4.20 Indirect corrective update. 67

5.1 Response of the nonlinear system given by Eqn. (5.1) and the reference model given by Eqn. (5.2). It can be seen that the system is hard to control without a controller. 78
5.2 The system given by Eqn. (5.1) is controlled by a linear state feedback controller. 80
5.3 series inverse control of system Eqn. (5.1), using direct adaptation mechanism. 81
5.4 Parallel corrective control of system Eqn. (5.1), using direct adaptation mechanism. 82
5.5 PD control of system Eqn. (5.1). 83
5.6 Parallel inverse control of system Eqn. (5.1), using direct adaptation mechanism. Notice that the adaptation rule uses the output error $e = y_m - y$. . 84
5.7 Parallel inverse control of system Eqn. (5.1), using direct adaptation mechanism. Notice that the adaptation rule uses the output of the PD controller, because this signal can also be seen as a measure for the parameter error of the neural weights. 85
5.8 Inverted pendulum problem. 87
5.9 Simulation of inverted pendulum using a neural adaptive feedback linearization controller. 89
5.10 Linear state feedback control of system Σ_1. 91
5.11 series inverse control of system Σ_1, using indirect update mechanism. 92
5.12 Parallel corrective control of system Σ_1, using indirect update mechanism. Linear state feedback controller: $u_{lfb}[k] = 0.3x[k] + r[k+1]$. 94
5.13 Parallel corrective control of system Σ_1, using indirect update mechanism. Linear state feedback controller: $u_{lfb}[k] = 0.3x[k]+0.1r[k+1]$. If the system isn't excited enough therefore the control objective will not be achieved. . . 95
5.14 Parallel inverse control of system Σ_1, using indirect update mechanism. . . 96
5.15 Linear state feedback control of system Σ_2. 97
5.16 Parallel corrective control of system Σ_2, using indirect update mechanism. . 98
5.17 Parallel corrective control of system Σ_1 using indirect inverse adaptation mechanism. 101
5.18 Serie inverse control of system Σ_1 using direct adaptation mechanism. The controller consist of a MLP. 103

5.19 Parallel corrective control of system Σ_1 using an indirect update adaptation mechanism. The controller consist of a MLP. 104

6.1 Practical guideline to decide when NAC should be applied. 107

A.1 Venndiagrams of: (a) subset $A \subset B$, (b) relative complement $A - B$, (c) complement $-A$, (d) intersection $A \cap B$, (e) union $A \cup B$, (f) disjunction $A \cap B = \emptyset$ 111
A.2 Illustration of: (a) Cartesian produkt $A \times B$, (b) binary relation $R \subset A \times B$ 112
A.3 Illustration of: (a) injective f, (b) surjective f, (c) bijective f 114
A.4 Trajectorie of Lorentz attractor 125
A.5 Trajectory around equilibrium point $\mathbf{x} = \mathbf{0}$:(a) stable, (b) asymptotically stable, (c) unstable 126

Preface

Personal motivation

During my practical assignment at the University of Amsterdam I came for the first time in contact with the field of neural control. The original assignment I had was the implementation of some control algorithm for a robot system. There were however some technical problems with this system and therefore I had to abort this assignment. I switched to another assignment to end my practical assignment properly. The goal of this new assignment was to control water vessels by using neural networks and it resulted into the report *Water vessels under control.* It was during this period that I became interested in neural control and I wanted to continue this research after my practical assignment.

I started my graduation assignment at the end of February 1996 and it had a subject in neural control. A precise assignment description could only be given after August 1996 because the subject was too unknown to define a proper subject of research. Finally the research subject was chosen to be *neural adaptive control.* During the period from February to August I did some side-projects, such as the publication of an article ("Neural Adaptive Feedback Linearization Control", *Journal A*. Vol. 37, no. 3, page 65-71), the implementation of a neural controller for a real world inverted pendulum system, demonstrating this system at the re-opening of the Neuro-Fuzzy Centrum in Germany and at the seventh lustral feast of the University of Twente.

Neural networks are magical ...

"What is it that makes one intelligent?" This question is probably asked often by researchers. Not only for curiosity but also with the hope to abstract the mechanism of intelligent behaviour into a mathematical description. If such a description would be available then there would be mechanical, or better, computational intelligence. But what is so special about (computational) intelligence?

Human behaviour has some remarkable properties not seen yet in any artificial machine built by human hands. Humans can learn, exchange knowledge, remember, associate, communicate and adapt themselves. They have also a good physical structure which makes it possible for them to walk, jump or grasp a pencil. Even some low live forms have some of these properties. It has been found that trying to mimic these properties by mechanical means is an almost impossible task.

The standard scientific approach to investigate some phenomena is to break it apart into small pieces. By trying to understand the properties of these pieces and their interaction, the phenomena is explained. This has also been done for human intelligent behaviour. It is found that the human body consists of some different systems. One of these systems is the nerve system and it controls part of the human body (together with the hormones systems). The nerve system is responsible for the fact that we can understand, think, grasp pencils, etc. Scientific investigation of the nerve system has shown that it consists of an interconnection of small units, called neurons. A neuron is connected to many other neurons and it can receive signals from and send signals to other neurons. The signal sent away to other neurons is some sort of transformation of the incoming signals. It is unbelievable that such a simple construction is capable of such a complex behaviour.

Neurons have been abstracted into a mathematical description which has been used to build artificial neural networks. An artificial neural network is an interconnection of artificial neurons and it tries to mimic some part of the human brain. It is hoped for, and sometimes it is claimed, that these artificial neural networks result into the same behaviour as the human nerve system does, namely intelligent behaviour. This has excited many researchers and artificial neural networks have been used to solve complex problems, such as natural speech recognition, vision, robot control, classification and control problems. A lot of experiments have been done, some with success and some without, but intelligent behaviour has not been achieved, at least, not comparable to human intelligent behaviour.

Philosophy of this report

This report has been written with one thought in my mind: neural (adaptive) controllers are no magical controllers. Artificial neural networks are not used because of the fact that they might result into intelligent behaviour. They are used because of the proven fact that they are capable of approximating any smooth function as closely as desired. Although there are other mathematical structures with the property of being a universal function approximator, the focus in this report is at using artificial neural networks. This is motivated by the fact that artificial neural networks are more easy to implement and are more robust. *By fitting artificial neural networks into the current nonlinear control theory, a practically valuable control methodology is obtained.*

Hard- and software usage

This report is made by using LaTeX2e. All the plots are made using *gnuplot 3.5*. Figures are drawn using *IntelliDraw 2.0* and are converted to postscript format and edited using *Ghostview 2.0 / Ghostscript 4.01*. Simulations are done using *Matlab 4.2c.1/Simulink* and *Turbo c++ 3.0 (Borland)*. These programs ran on Intel based computers (486/pentium) with Windows 95 as operating system.

Acknowledgments

I want to thank Leo Veelenturf for being my supervisor and giving me all the freedom I needed. I also want to thank my parents, Netty and Albert van Breemen Sr., because they have made it possible for me to study and they have always supported me. I also want to thank Niels Moseley for the many technical and more or less philosophical discussions we had. Andre Beltman I want to thank for help with software problems. Paul Schijfsma (Brammetje) I want to thank for his help with the mechanics of the inverted pendulum system, such that this system was ready for a demonstration for TV-recordings of "TV Oost". Finally I want to thank Hans Schurer for his help with DSP's and nonlinear control theory and Anneke for the many sociable chats we had.

Enschede, The Netherlands
January 1997 *A.J.N. van Breemen*

Abbreviations and Symbols

Abbreviations

ANN	Artificial Neural Network
DSP	Digital Signal Processor
I/O	Input / Output
MIMO	Multiple Input Multiple Output
MLP	Multi Layer Perceptron
MRAS	Model Reference Adaptive System
NN	Neural Network
NAC	Neural Adaptive control
NACler	Neural Adaptive controller
NAFLC	Neural Adaptive Feedback Linearization Controller
RBFN	Radial Basis Function Network
SISO	Single Input Single Output
STR	Self Tuning Regulator

Symbols

C	controller mapping
C_{lin}	linear controller mapping
C_{NAC}	neural adaptive controller mapping, often $C_{NAC} = C_{NN} + C_{lin}$
C_{NN}	neural controller mapping
e	output error $e = y_m - y$
$\hat{e}$	identification error $\hat{e} = y - \hat{y}$
$e_{\hat{u}}$	input error $e_{\hat{u}} = u - \hat{u}$
e_{u_m}	reference input error $e_{u_m} = u_m - u$
$\hat{e}_{u_m}$	estimated reference input error $\hat{e}_{u_m} = \hat{u}_m - u$
M	identification model mapping
M^{-1}	inverse identification model mapping
M_{NN}	neural identification model mapping
M_{NN}^{-1}	inverse neural identification model
r	reference input signal
S	process mapping
S_m	reference model mapping
u	process input, controller output $u = C(.)$
$\hat{u}$	estimated process input
v	unmeasurable disturbance
w	measurable distrubance
w_i	scalar weight (scalar parameter)

$\mathbf{w}$	weight vector (parameter vector)
$\mathbf{W}$	weight matrix (parameter matrix)
y	process output $y = Su$
$\hat{y}$	estimated process output $\hat{y} = M(.)$
y_m	reference output $y_m = S_m r$

Block diagram symbols

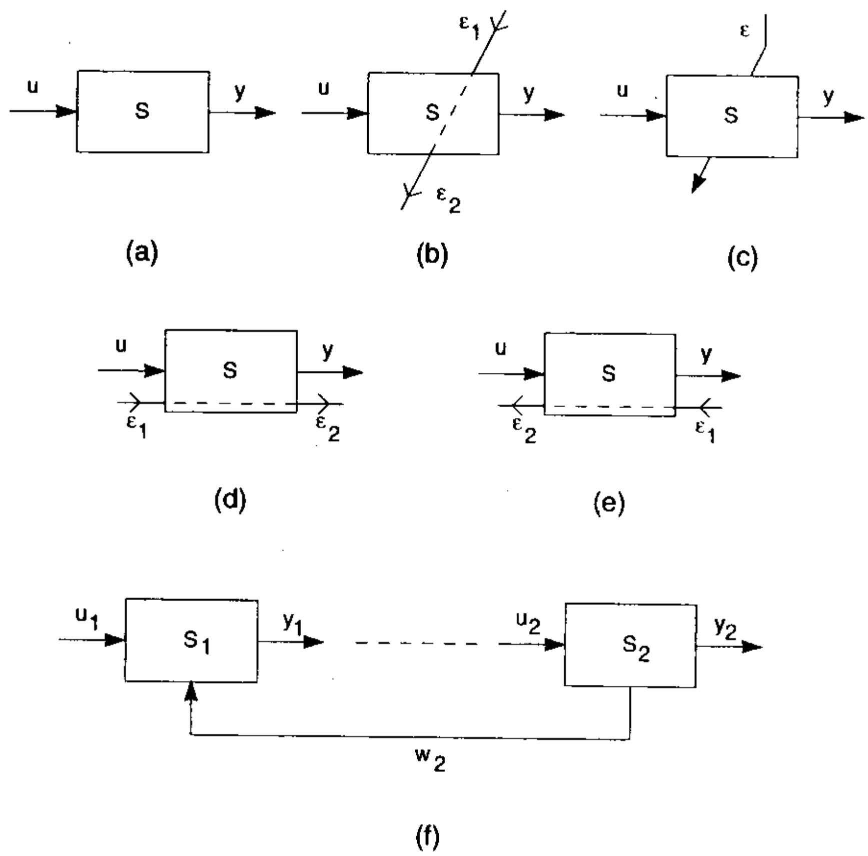

Figure 0.1: Block diagram symbols: (a) $y = Su$, (b) $\epsilon_2 = \frac{\partial y}{\partial u}\epsilon_1$, (c) $S = S(.|\mathbf{w}), \dot{\mathbf{w}} = \Gamma(\epsilon)$, (d) $\epsilon_2 = S\epsilon_1$, (e) $\epsilon_2 = S^{-1}\epsilon_1$, (f) $S_1 = S(.|\mathbf{w}_1), S_2 = S(.|\mathbf{w}_2), \mathbf{w}_1 = \mathbf{w}_2$.

Chapter 1

Introduction

1.1 Neural Adaptive Control

Designing a controller for a process is often model-based. This means that a model of the process is needed. Sometimes a model is not available because

- the process is too complex
- the process contains uncertainties
- there is no time to obtain a model
- there is no knowledge to obtain a model

Therefore a model-free controller design method is needed. A controller which is universal enough to control a wide range of processes and which has the ability to adapt itself during control would be a good basis for a model-free controller design method.

A neural network is an adaptive information processing system which is capable of realizing any desired input/output mapping to any desired degree of accuracy. Originally neural networks were built from a biological background to mimic some parts of the human brain. It was hoped for that these 'models of the brain' would have the same property as the brain itself, namely that they would result into intelligent behaviour. Currently neural networks are seen as 'just' a parameterized class of nonlinear maps. Because a neural network is adaptive and capable of approximating any input/output mapping to any degree of accuracy, it is a good candidate for serving as a basis for a model-free controller design method. A controller which uses neural networks and adapts these networks during control is called a neural adaptive controller.

To gain a deeper understanding in the working of neural adaptive controllers they have to be systematized. In this report a systematization is proposed which is based on the point of view that a neural adaptive controller consists of two parts, namely:

- a controller
- an adaptation mechanism

The controller, denoted by the signal operator C, is parameterized, which means that the mapping of the controller depends on a set of parameters $\mathbf{w}$: $C = C(.|\mathbf{w})$. The controller structure has to be chosen such that there is at least one set of parameters $\mathbf{w}^*$ for which the control objective can be reached. In this report the control objective will always be reference signal following. The adaptation mechanism is used to adapt the parameters $\mathbf{w}$ of the controller during control, such that $\lim_{t\to\infty} \mathbf{w}(t) = \mathbf{w}^*$.

Both the controller and the adaptation mechanism can be categorized. In this report it is proposed to categorize the controller into the following five classes:

- conventional controller
- series inverse controller
- parallel inverse controller
- parallel corrective controller
- feedback linearization controller

This categorization is based on the functionality of the neural network within the controller context. The adaptation mechanism has been categorized into the following five classes:

- direct adaptation
- direct sensitivity adaptation
- indirect inverse adaptation
- indirect sensitivity adaptation
- indirect update

This classification is based on the way a correction signal is calculated to adapt the controller parameters.

An overview of the theoretical status of the field of neural adaptive control in 1994 is given in the report *ESPRIT III PROJECT 8039: NACT - Neural Adaptive Control Technology* [46]. In this report it was discussed that in order to built an consistent methodology for neural adaptive control at least the following steps have to be made:

- a classification should be made of nonlinear plants to control.
- directly linked to this classification, a classification should be made of suitable neural controller structures,
- and directly linked to the neural controller structure, a classification should be made of suitable parameter adaptation rules.

The systematization of neural adaptive controllers in this report is thus a continuation of the ESPRIT III report [46].

1.2 The scope of this report

Chapter 2: Developments in Automatic Control

This chapter gives an overview of developments in automatic control. This historical overview serves as foundation for a better understanding and appreciation of neural (adaptive) control. It is discussed that intelligent control, of which neural control is a part of, is an alternative to overcome the growing complexity of control problems. Furthermore it is discussed that although the success of a control strategy depends on its mathematical foundation, the choice of the actual controller used to control the system depends also on more subjective factors, such as experience, tradition and available knowledge.

Chapter 3: Artificial Neural Networks

In this chapter neural networks are being discussed from an engineering point of view. Instead of explaining neural networks from a biological background, this chapter starts from a signal processing view. A more in-depth treatment is given to the multi layer perceptron and the radial basis function network. A learning rule for the MLP, called the back propagation rule, is derived. A method to implement a RBFN very efficiently, using a standard DSP, is also discussed. At the end of this chapter a discussion is given why the RBFN is more suitable for control purpose then the MLP.

Chapter 4: Neural Adaptive Control

This chapter and the implementation of RBFNs forms the core of this assigment. In this chapter a systematization is presented of neural adaptive control. The general point of view is that neural adaptive controllers consist of two parts: a controller and an adaptation mechanism. Both the controller and the adaptation mechanism are categorized in order to gain deeper under standing in the working of neural adaptive controllers and to gain a systematic design method. Besides a systematization of neural adaptive controllers, also two Lyapunov based stability proofs for different neural adaptive controllers are presented. These proofs shows the theoretical limitations of the two neural adaptive controllers when they are implemented in practice.

Chapter 5: Experiments

To demonstrate the working of neural adaptive controllers extensive experiments have been done. These experiments show that different combinations of controller structures and adaptation mechanisms can be used to control the same process successful.

Chapter 6: Conclusions, discussion & recommendations

The results of this report are summarized in this chapter. Furthermore recommendations are made for further research and the practical issue *when to apply neural adaptive control* is discussed.

Chapter 2

Developments in Automatic Control

Looking back in wonder!: in the past decades so much has changed and still so much is changing rapidly. The impact of control concepts and tools has increased dramatically. Even without a crystal ball we may predict this increase to continue ... Yes indeed, we are living in an interesting period of time!

P. Eykhoff

Goal: *This chapter gives an overview of several developments in automatic control. This historical overview will be the foundation for a better understanding and appreciation of neural control and gives an opportunity to discuss future neural control developments at the end of this report.*

To use a machine effectively we must have some influence on the machine's behaviour. It is often a difficult task to decide how to influence the machine, such that a desired behaviour of the machine is obtained. A control system is a device which generates a set of control signals automatically, given some user specifications. The design of such a control system has proven to be a difficult task and many approaches have been proposed. We can see an ever changing shift in controller design methodologies. In this chapter a brief historical overview is given of automatic control systems. The controller design methods and the controller specifications have always been influenced by politics, economics, available theoretical knowledge and available implementation methods. It's not the intention of this chapter to provide a fully historical overview of this subject, this chapter serves only to provide a rough sketch of developments in automatic control. Section 1 is based largely on an article called *A Brief History of Automatic Control* by Stuart Bennett [4].

2.1 A Brief History of Automatic Control

2.1.1 Early Control: to 1900

When man discovered the regularities of nature, summarized in the laws of nature, he was enabled to use these in a directed way to design machines with some predefined tasks. Sometimes a machine was nothing more then a toy to demonstrate some principle of nature. Hero (1st century BC) was one of the earliest technician who built automatic machines and one of his best known machines is his device to open temple doors. This device is pictured in Fig. 2.1 and is what we now call an open-loop system. The input was the fire on the altar. Due to the heat, the air in the container expanded and drove

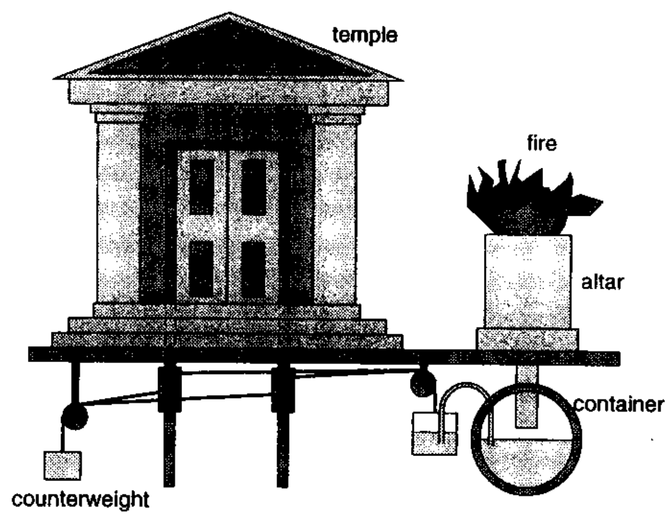

Figure 2.1: Hero's device for opening temple doors.

the water into the bucket which became heavier. If the bucket was weighty enough, it descended and the door spindles by means of the ropes. This caused the counterweight to rise. When the fire was put out, the air in the container shrank and water was flowing from the bucket into the container. As the bucked became lighter the counterweight descended and the doors closed.

In the eighteenth and nineteenth century the development of control devices and control theory was stimulated by the industrial revolution in England. Many applications were concerned with trying to keep some system variables at a given desired value. This problem is called the regulation problem and regulators were designed to regulate temperature, pressure and velocity. The physicist René-Antione de Réamur (1683-1757) designed a temperature regulator based on a U-tube filled with mercury. This U-tube was connected to a vessel filled with liquid and if the temperature of the liquid increased there was a float in the mercury which operated an arm which was connected to a furnace. Based on this idea, the chicken-farmer Bonnemain (circa 1743-1828) designed a temperature regulator based on the differential expansion of different metals. Characteristic for these two regulators is that both use power from the measured signals to actuate an actuator.

Another very important regulator of this period is the centrifugal regulator or the steam engine governor (see Fig. 2.2). Although James Watt (1736-1819) has not invented this device, he was the first person who used it to control a steam engine automatically. This governor is what we now call a closed loop or feedback controller and consisted of a proportional feedback of the measured signal. Many improvements have been made to Watt's original governor among others by William Siemens (1846 and 1853), Charles T. Porter (1858), Thomas Pickering (1862) and William Hartnell (1872). There were however some stability problems with these governors and this motivated the search for theoretical means of predicting stable operations regions. The first papers on this topic were published by J.V. Poncelet (1788-1867) and G.B. Airy (1801-1892), but they didn't solve the stability problem. The stability problem was solved by James Clerk Maxwell (1831-1879) in 1868 and published in a paper called *On Governors*. He derived linear differential equations of the governor and used the already know fact, that the stability of a dynamic system is determined by the roots of its characteristic equation. A more generalized stability criteria was introduced by Edward J. Routh (1831-1907) in his work

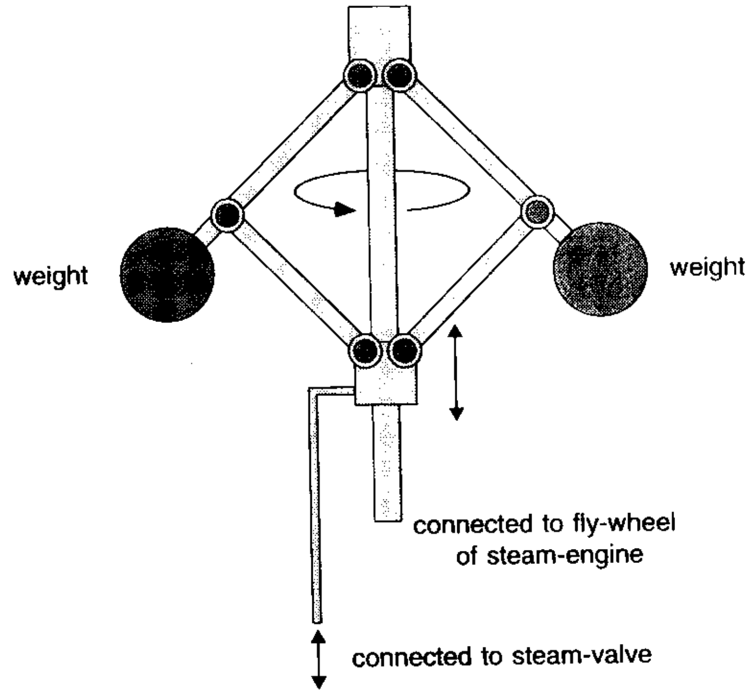

Figure 2.2: Watt's steam engine governor.

Stability of Motion and is now known under the name Routh-Hurwitz stability criteria, because Adolf Hurwitz (1859-1919) derived the same criteria independently.

With the growing sizes of the machines there rose the desire for having powered actuator devices. These devices found their applications in the steering mechanisms of large ships. Frederick Sickels designed the first powered open-loop steering engine and it was patented in 1853. The first closed-loop steering engine was patented in 1866 and was designed by J. McFarlane Gray. Around the same time Jean Jospeh Farcot designed a range of closed-loop positioning control systems which he called "servo-moteur", what we now call "servomechanisms". The development in electrotechnical devices, to measure, transmit, manipulate signals and for actuating motivated further control system research.

2.1.2 The Pre-Classical Period: 1900-1940

During the period 1900-1940 most developments in automatic control systems were motivated by industrial problems. Control devices were built in different areas of engineering: for ships, airplanes, automobiles and flow control in process industries. Most designs were not based on a theoretical foundation but rather on practical experience. In this period one recognized two major problems [4]: "(1) there was a lack of theoretical understanding and no common language in which to discuss problems, and (2) there were no simple, easily applied analysis and design methods." Only differential equations and the Routh-Hurwitz stability criteria were known, although this last criteria wasn't generally known. There was a desire for a more comprehensive understanding of control systems, because often a controller designed for a particular application didn't work for another. Furthermore, if the operation conditions changed the controlled system could become unstable.

The lack of having theoretical tools wasn't the only problem, there were also no suitable linear, stable amplification devices. When Nicholas Minorsky (1885-1970) presented an analysis of a PID controller for positioning control, it couldn't be implemented very well because of this lack of amplification devices. Amplification devices were also a problem in long-distance telephony. Signals were distorted which resulted in a limited number of amplifiers that could be used in series. In the early 1920s Harold Stephen Black (1898-1983) experienced with feeding back part of the output signal to the input of the amplifier, which resulted in a reduction of noise and component drift. This techniques was used within AT&T in 1931. Black's assistant, Harry Nyquist (1889-1979), wrote a paper in 1932 called *Regeneration Theory* which led to a deeper understanding of feedback devices. It also resulted in a practical, rather then an analytical, method for designing and analyzing control systems.

Another significant contribution to the theory of feedback control systems is the paper of Harold Lock Hazen (1901-1980) called *Theory of Servomechanisms* which was published in 1934. This work was the result of a study and design of high-performance servomechanisms by using analog calculating machines.

2.1.3 The Classical Period: 1935-1950

At the end of the classical period (1935-1950) the, what we now call, classical control techniques were established. These were design techniques for control systems for linear single input single output (SISO) systems with constant coefficients. Among these techniques were the frequency response techniques based on the work of Nyquist, Bode and Nichols. Systems could be described in terms of frequency bandwidth, resonance and gain and phase margins. By using differential equations and the Laplace transform to solve these equations, the system could be characterized in terms of rise time, percentage overshoot, steady-state error and damping.

The development of these techniques was the result of different groups of researchers. At AT&T research was concentrated on finding possibilities to enlarge the bandwidth of communication systems. Hendrik Bode contributed a lot to the solution of this problem. His ideas were published in 1945 in his book *Network Analysis and Feedback Amplifier Design*. A group of mechanical engineers and physicists working in the process industry in the U.S., together with Ed S. Smith of the Builders Iron Foundry Company, developed a systematic and universal approach for control system design. At the Electrical Engineering Department of MIT a group led by Harold. L. Hazen and Gordon S. Brown developed time-domain techniques, using the (signal) operators point of view, and developed the block diagram method.

During WOII research was mainly concentrated at some particular problems. A problem which did get a lot of attention was the antiaircraft radar-tracking control system (see Fig. 2.3). This system consisted of two parts: a radar and an antiaircraft gun. The radar was used to locate the position of the airplane. Next, the position of the plane had to be predicted for the moment at impact. Finally the gun had to be positioned. The first systems used human operators to position the gun, but it was soon found that this method took a long time and wasn't good enough anymore for the faster becoming airplanes. So, the radar had to be connected directly to the gun. Gordon S. Brown and his students used the block diagram approach to solve this problem. It was Albert C. Hall who showed that each block (electrical and mechanical) could be seen as a transfer function, and therefore Nyquist stability test could be used. At the Radiation Laboratory of MIT a group of researchers designed the SCR-584 radar system. This system was connected to the M9 director, built by a group led by Bode and including Blackman, C.A. Lovelland Claude Shannon, working at the Bell Telephone Laboratory, and proved to be very effective against V1 rockets.

From the antiaircraft control system research some new problems arose. When integrating the different parts of the system, which were built by different groups, the performance depended on how these parts worked together, and did not depend on the performance of the individual parts. Research to this problem led to a deeper understand-

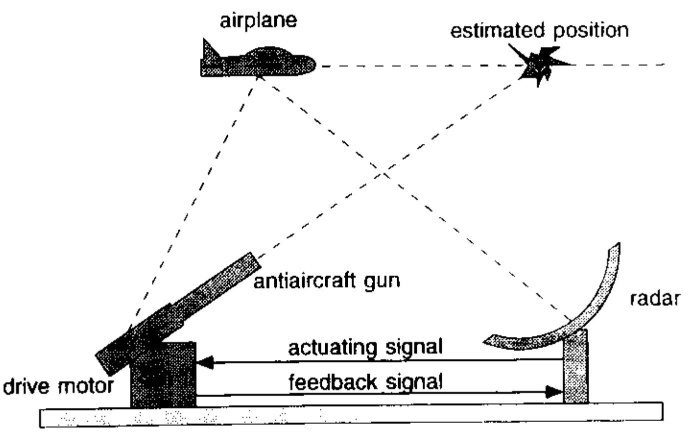

Figure 2.3: Antiaircraft radar-tracking control system.

ing of bandwidth, noise and non-linearities.

Another person who has contributed a lot to the theory of control systems is Norbert Wiener (1894-1964). He was also tackled to the antiaircraft radar-tracking control problem. He made a study of stochastic systems and his work resulted in the report *The Extrapolation, Interpolation and Smoothing of Stationary Time Series with Engineering Applications* which was written in 1942, but published in 1949. Nowadays one often use the terms prediction and filtering instead of extrapolation and interpolations when considering signal filtering problems.

After the war new problems gained the attention of many researchers. One was becoming aware of the fact that real systems weren't linear and that measurement signals were corrupted with errors and noise. Also stability wasn't the only most important guideline for the design of the control system, the controller had to give "optimal" performance. New techniques had to be developed to solve these problems.

2.1.4 Modern Control

The origin of the modern control is placed in 1957 by Micheal Athans. In this years an international conference was held on automatic control to form an international organization. On September 11 and 12 1957 this organization was officially formed at Paris and was called the International Federation of Automatic Control (IFAC). The establishment of this organization shows the changing relation of the East and the West. The first congress of the IFAC was held during the summer of 1960 in Moscow. The research in the West was mainly concentrated on the development of frequency methods and input-output models, while research in the East was concentrated at the state space method. During the next two decades these two approaches were combined. Nowadays we still recognize these two different point of view in controller design methods, as can be clearly seen from the construction of the book *Computer Controlled Systems* by Karl J. Åström and Björn Wittenmark [1].

During this period research was mainly motivated by two factors [4]: "first, the problem that governments saw as important, the launching, maneuvering, guidance, and tracking of missiles and space vehicles, and second, by the advent of the digital computer." The space program introduced new theoretical and practical problems. Beside positional accuracy and stability new problems such as "minimum time" and "minimum fuel" consumption had to be solved. This led to the development of optimal control. Richard Bellman formulated in this context the "principle of optimality" and dynamic programming. In