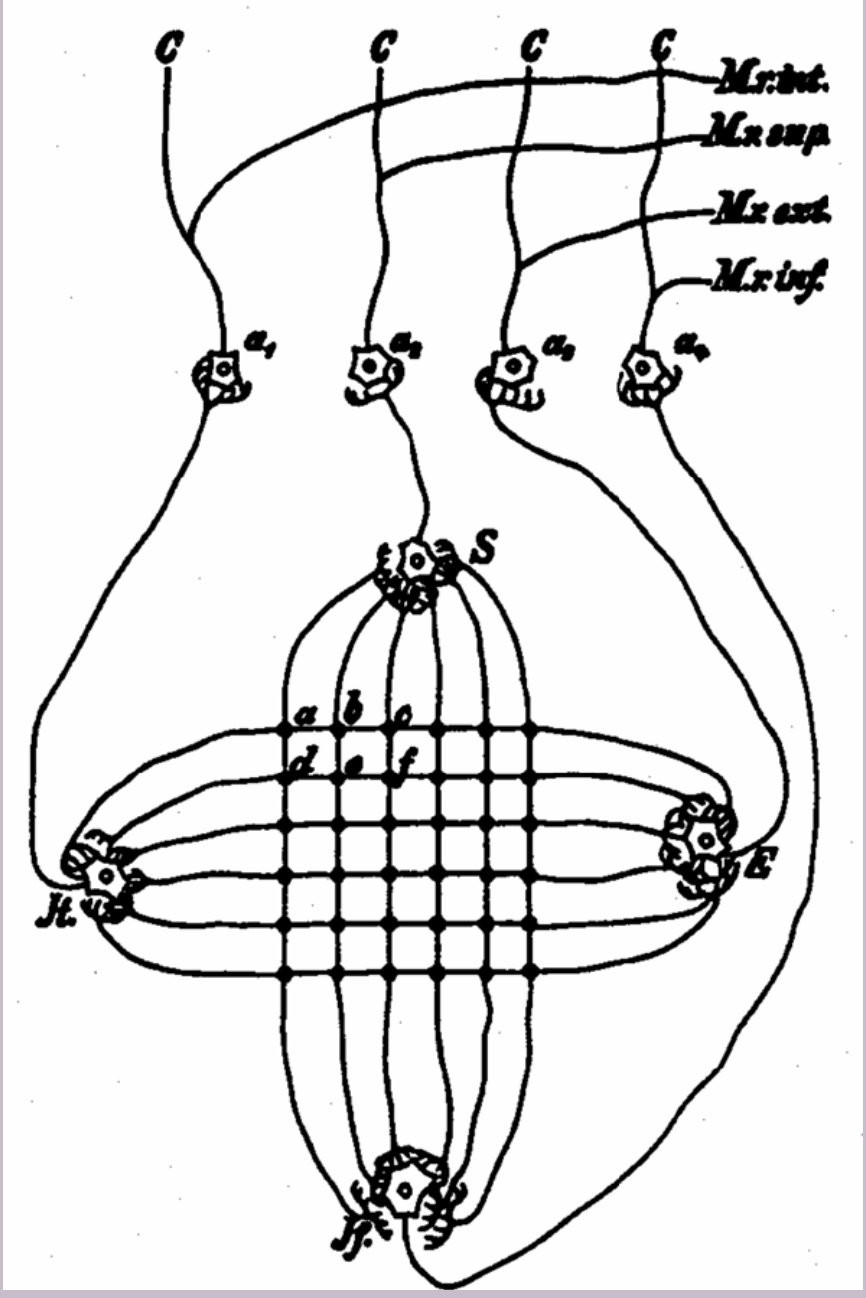
Neural Networks as Cybernetic Systems

3rd and revised edition

Holk Cruse



brains, minds media

Neural Networks as Cybernetic Systems – 3rd and revised edition

Holk Cruse, Dr.

Department of Biological Cybernetics and Theoretical Biology
Bielefeld University
Universitaetsstrasse 25
33615 Bielefeld
Germany

holk.cruse@uni-bielefeld.de

Electronic Version:
Cruse, Holk (2009). Neural Networks as Cybernetic Systems (3rd and revised edition). bmm1841. Brains, Minds and Media, Bielefeld, Germany. (urn:nbn:de:0009-3-18414), published April /July 2009.
http://www.brains-minds-media.org/archive/1841
This site refers to two parts:
Part I (Chapter 1-8): bmm1988 | urn:nbn:de:0009-3-19884, published July 2009
http://www.brains-minds-media.org/archive/bmm1988
Part II (Chapter 9-19): bmm1990 | urn:nbn:de:0009-3-19906, published July 2009
http://www.brains-minds-media.org/archive/bmm1990

Published by
Brains, Minds & Media
ISSN 1861-1680
www.brains-minds-media.org
info@brains-minds-media.org

Department of Neurobiology
Bielefeld University
Universitaetsstrasse 25
33615 Bielefeld
Germany

Second Edition published by Brains, Minds & Media, Bielefeld, 2006 (urn:nbn:de:0009-3-6153).
First edition published by Thieme, Stuttgart, 1996 (ISBN 3-13-102181-0) – out of print.

Table of Contents

Preface to the 1st edition 1

Preface to the 2nd edition 2

Preface to the 3rd edition 2

1 Introduction 3

1.1 Symbols 5

1.2 Linear Systems 6

1.3 The Tasks of Systems Theory Investigations 7

2 The Most Important Input Functions 8

2.1 The Step Function 8

2.2 The Impulse Function 9

2.3 The Sine Function 11

2.4 The Statistical Function 14

2.5 The Ramp Function 15

3 The Fourier Analysis 15

4 The Properties of the Most Important Filters 17

4.1 First Order Low-Pass Filter 18

4.2 First Order High-Pass Filter 21

4.3 N-th Order Low-Pass Filter 23

4.4 N-th Order High-Pass Filters 25

4.5 Band-Pass filter 26

4.6 Lead-Lag System 28

4.7 Pure Time Delay 31

Box 1 32

4.8 Integrator and Differentiator 33

4.9 Oscillating Systems 35

4.10 Combination of Different Filters 38

5 The Characteristics 39

5.1 General Remarks 39

5.2 The Static Characteristic 40

5.3 The Dynamic Characteristic 49

6 Combination of Linear and Nonlinear Elements 51

6.1 The Sequence Linear Element - Nonlinear Element 51

6.2 The Sequence Nonlinear Element - Linear Element 51

6.3 Unidirectional Rate Sensitive (URS) Element 52

7 The Study of Nonlinear Systems 54

Box 2 56

8 Feedback Systems 59

8.1 Open Loop Control - Closed Loop Control 59

8.2 The Most Important Terms 59

8.3 A Simple Feedback System 61

8.4 The Study of a Feedback System 65

8.5 Feedback Control Systems with Dynamic Properties 67

8.6 The Stability of Feedback Systems 71

Box 3 76

8.7 Further Types of Feedback Control Systems 77

8.8 An Example 79

8.9 Reafference Principle 81

8.10 The Dynamic Properties of a Simplified Neuron 83

9 Massively Parallel Systems 85

10 Feedforward Networks 89

10.1 Spatial Low-Pass Filter 89

10.2 Spatial High-Pass Filter 91

10.3 Spatial Band-Pass Filter 92

Box 4 93

10.4 A Winner-Take-All Network 95

10.5 Learning Matrix 96
10.6 Perceptron 98
10.7 Multilayered Feedforward Nets 99
11 Recurrent Networks 103
11.1 Pattern Generation 106
Box 5 108
11.2 Recurrent Winner-Take-All Network 110
11.3 Optimization 112
11.4 Associative Storage 113
11.5 Elman Networks and Jordan Networks 114
11.6 Nets showing chaotic behaviour 116
12 Learning 118
12.1 Hebb's Rule 118
12.2 The Delta Rule 120
12.3 Generalized Delta Rule 121
12.4 Teacher Forcing and Learning with Suppression Units 125
12.5 Reinforcement Learning 127
12.6 Expert Networks 132
Box 6 134
12.7 Genetic Algorithms 135
12.8 Competitive Learning 136
13 Feature Maps 140
Box 7 145
14 Some Special Recurrent Networks 146
14.1 Simple Oscillators 146
14.2 Echo State Networks 148
14.3 Linear differential equations and recurrent neural networks 149
14.4 MSBE Nets 152

14.5 MMC Nets 152

14.6 Forward Models and Inverse Models 156

15 Spatial Coding 161

16 Animats 166

17 Development of Complexity in Animats 176

Box 8 182

18 Emergent Properties 183

19 The Internal Aspect – Might Machines Have Feelings? 185

Appendix I – Laplace Transformation 191

Appendix II – Simulation of Dynamic Systems 195

Appendix III – Exercises 196

References 198

Preface to the 1st edition

Biological systems are usually much too complicated to be understood in their entirety. Scientific progress is therefore generally based on the fragmentation of the systems under investigation. This means the system is broken down into smaller parts or subsystems, which can then be more easily approached. However, such a reduction to a lower level - for instance - from behavior to reflexes, from reflexes to neuronal cells, or from cells to molecules - also has serious shortcomings. First, the overview of the whole system may be lost. Looking at the lower level, one may not see "the forest for the trees" because many system properties are only understandable when not only the individual parts of the system, but also the cooperation of these parts are taken into account. Second, this reduction may further increase the gap that already exists between biological research and the investigation of problems on more complex levels, such as those considered in psychology or even philosophy. To make this gap smaller "holistic" approaches are required.

One sensible way to oppose this reductionistic path is the use of simulation. The construction of quantitative models, usually in the form of computer simulation, is an important tool in biology. Such a simulation allows a step in the other, "antireductionistic" direction, namely to construct complex systems from smaller, simple elements. Through investigation of the simulated system, the properties of the whole system can be better understood.

The tool of simulation is particularly important for understanding the dynamic properties of systems. Such dynamic properties are often produced by feedback loops within the system. However, the human brain does not seem very well adapted to grasp such systems. Simulations could improve this situation. We might become more familiar with the properties of dynamic systems and thus train ourselves to understand such systems so that, even without an explicit computer simulation, some predictions could be made. Such dynamic systems occur in many fields, from genetics, metabolism, and ecology to, of course, neurobiology and ethology. Although this book will concentrate on the latter two, the tools provided can also be applied to the other fields. But these tools are also applicable to fields outside biology, e. g., psychology, and to even more distant areas, such as economics, physics, and electrotechnology (which in fact gave rise to many of these ideas).

Ethology, although an important field in biology, had been attracting less interest in recent decades mainly because of a lack of theoretical concepts. This has, however, dramatically changed in recent years because the emergence of the theory of artificial neural networks (ANN) and the field of artificial life, has led to the development of a great number of models and modeling tools that can now be fruitfully applied to ethology and neuroethology. Although the treatment of simple neuronal models was already an important subject of early biological cybernetics, the focus of interest later moved to "pure" systems theory. Only in the last decade did the field of ANN approach gain its enormous thrust. These two fields have not yet been brought into intensive contact with each other, but the consideration of dynamic properties so central to systems theory has the potential to make a great impact on ANN. This book attempts to combine both approaches which, as mentioned, stem from the same roots. It can be expected that both fields will profit from each other.

Usually textbooks on these fields are loaded down with a great deal of mathematics, which makes them somewhat indigestible for the typical student of biology. To minimize these problems, this book tries to avoid the use of mathematical formulas as far as possible. The text is based not on differential equations or on complex variables, but rather on illustrations. It nevertheless provides sufficient information to permit the reader to develop quantitative models. Technical aspects are relegated to the appendices.

The first part of this book is based on an earlier version of my book "Biologische Kybernetik", which

was inspired by the exciting lectures of D. Varju at the University of Tübingen. Writing this greatly expanded version of the book would not have been possible without the help of a number of colleagues who met in a research group funded by the Center of Interdisciplinary Research (ZiF) of the University of Bielefeld. I mainly want to thank to H. U. Bauer, Frankfurt; H. Braun, Karlsruhe; G. Hartmann, Paderborn; J. Dean, A. Dress, P Lanz, H. Ritter and J. Schmitz, all from Bielefeld; and H. Scharstein, Cologne, for a number of helpful comments. Furthermore, I would like to thank to A. Baker who helped with the English in an earlier version of the manuscript, A. Exter for the preparation of many figures, and P. Sirocka and Th. Kindermann for providing Figures B 5.3 and B 5.1, respectively. Furthermore, I owe a debt of gratitude to many students of my lectures who succeeded in finding errors and unclear formulations in the text. Of course, the responsibility for all remaining flaws is my own.

April 1996 Holk Cruse

Preface to the 2nd edition

Ten years ago, the first edition of the book ”Neural Networks as Cybernetics Systems“ has been published by Thieme. At this time there was still an ongoing debate whether the neural network approach comprises just a fashionable, but short living hype. Meanwhile this approach is well established. Understanding complex systems by means of simulation is more and more accepted, also within biology. The more it is important to provide students with a tool that helps to understand and also actively perform simulations. This is particularly important for students their primary education was not in mathematics or in computer science. It is the goal of this text to provide such a tool. In the first part, both linear and nonlinear aspects of systems theory, sometimes called filter theory or theory of dynamical systems, are treated in a way that mathematical terms are avoided as far as possible. In part II this is extended to the theory of massively parallel systems, or theory of neural networks. This qualitative approach is also suited as a first step for students that later plan to follow a more thorough quantitative understanding.

Appearing as an (open access) e-version, the handling of the text is easier compared to the earlier printed version, figures are coloured, errors are corrected, (hopefully new errors appear at a minimum), and some chapters, in particular those concerning the important field of recurrent networks, are added anew. The most helpful extension however concerns the software package that allows to perform exercises concerning simulations for part I. This package is written so that it can be used in an extremely simple way. Extensions for part II are in preparation.

January 2006 Holk Cruse

Preface to the 3rd edition

This third edition essentially compares with the 2nd one, but has been improved by correction of errors and by a rearrangement and minor expansion of the sections referring to recurrent networks. These changes hopefully allow for an easier comprehension of the essential aspects of this important domain that has received growing attention during the last years.

September 2008 Holk Cruse

1 Introduction

The task of a neuronal system is to guide an organism through a changing environment and to help it to survive under varying conditions. These neuronal systems are probably the most complicated systems developed by nature. They provide the basis for a broad range of issues, ranging from simple reflex actions to very complex, and so far unexplained, phenomena such as consciousness, for example. Although, during this century, a great amount of information on neural systems has been collected, a deeper understanding of the functioning of neural systems is still lacking. The situation appears to be so difficult that some skeptics speak of a crisis of the experimentalist. They argue that to continue to accumulate knowledge of further details will not help to understand the basic principles on which neural systems rely. The problem is that neuronal systems are too complicated to be understood by intuition when only the individual units, the neurons, are investigated. A better understanding could be gained, however, if the work of the experimentalist is paralleled and supported by theoretical approaches. An important theoretical tool is to develop quantitative models of systems that consist of a number of neurons. By means of proposing such models which, by definition, are simplified versions or representations of the real systems, comprehension of the properties of the more complicated, real systems might be facilitated.

The degree of simplification of such models or, in other words, the level of abstraction may, of course, be different. This text gives an introduction to two such approaches that will enable us to develop such models on two levels. A more traditional approach to developing models of such systems is provided by systems theory. Other terms of similar meaning are cybernetics or control theory. As the basic elements used for models in systems theory are known as filters (which will be described in detail later) the term filter theory is also used.

Systems theory was originally developed for electrical engineering. It allows us to consider a system on a very abstract level, to completely neglect the real physical construction, and to consider only the input-output properties of the system. Therefore, a system (in terms of systems theory) has to have input channels and output channels that receive and produce values. One important property is that these values can vary with time and can therefore be described by time functions. The main interest of the systems theory approach, therefore, concerns the observation of dynamic processes, i. e., the relation of time-dependent input and output functions. Examples of such systems could be an electrical system, consisting of resistors and capacitors as shown in Figure 1.**1**, or a mechanical system comprising a pendulum and a spring (Fig. 1.**2**). In the first case, input and output functions are the voltage values u_i and u_o. In the second case, the functions describe the position of the spring lever, x(t), and the pendulum, y(t), respectively. An example of a biological system would be a population, the size of which (output) depends on the amount of food provided (input). Examples of neuron-based systems are reflex reactions to varying sensory inputs. A classic example is the optomotor response that occurs in many animals and has been intensively investigated for insects by Hassenstein and Reichardt and co-workers in the 1950s and 60s (Hassenstein 1958 a, b, 1959: Hassenstein and Reichardt 1956; Reichardt 1957,1973).

Although the systems theory approach is a powerful tool that can also be applied to the investigation of individual neurons, usually it is used for model systems far removed from the neuronal level. However, earlier in the field of biological cybernetics, models were considered where the elements correspond to neuron-like ones, albeit extremely simplified. This approach has gained large momentum during the last 10 years, and is now known by terms such as artificial neural networks, massive parallel computation, or connectionist systems. Both fields, although sharing common interests and not included in the term cybernetics by accident, have developed separately, as

demonstrated by the fact that they are covered in separate textbooks. With respect to the investigation of brain function, both fields overlap strongly, and differences are of quantitative rather than qualitative nature. In this text, an attempt is made to combine the treatment of both fields with the aim of showing that results obtained in one field can be applied fruitfully to the other, forming a common approach that might be called neural cybernetics.

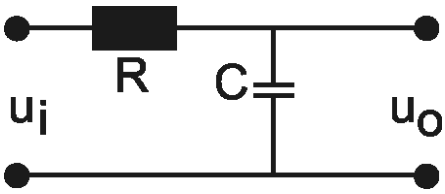


Fig. 1.**1 An example of an electric system containing an resistor R, and a capacitor C**. u_i input voltage, u_o output voltage

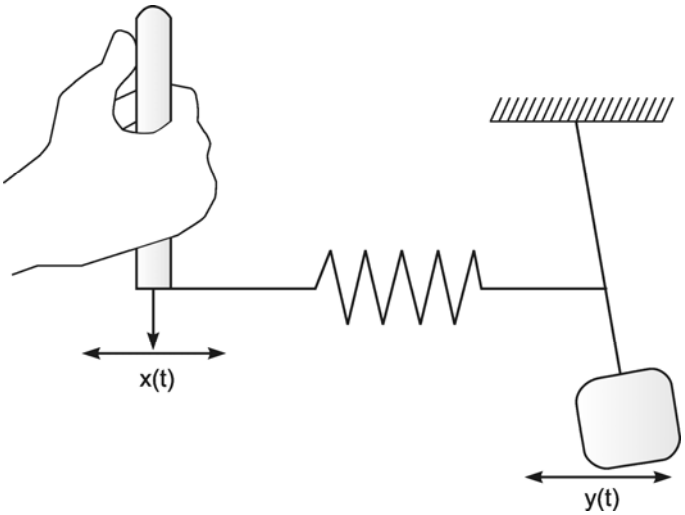


Fig.1.**2 An example of a mechanical system consisting of an inert pendulum and a spring.** Input function x(t) and output function y(t) describe the position of the lever and the pendulum, respectively

One difference between the models in each field is that in typical systems theory models the information flow is concentrated in a small number of channels (generally in the range between 1 and 5), whereas in a neural net this number is high (e. g., 10 - 10^5 or more). A second difference is that, with very few exceptions, the neural net approach has, up to now, considered only the static properties of such systems. Since in biological neural networks the dynamic properties seem to be crucial, the consideration of static systems is, of course, a good starting point: but unless dynamics are taken into account, such investigations miss a very important point. This is the main subject of systems theory, but neural nets with dynamic properties do exist in the form of recurrent nets.

To combine both approaches, we begin with classic systems theory, describing properties of basic linear filters, of simple nonlinear elements and of simple recurrent systems (systems with feedback loops). Then, the properties of simple, though massively parallel, feed-forward networks will be discussed, followed by the consideration of massively parallel recurrent networks. The next section will discuss different methods of influencing the properties of such massively parallel systems by "learning." Finally, some more-structured networks are discussed, which consists of different subnetworks ("agents"). The important question here is how the cooperation of such agents could be organized.

All of these models can be helpful to the experimentalist in two ways. One task of developing such a model-be it a systems theory (filter) model or a "neuronal" model-would be to provide a concise description of the results of an experimental investigation. The second task, at least as important as the first, is to form a quantitative hypothesis concerning the structure of the system. This hypothesis could provide predictions regarding the behavior of the biological system in new experiments and could therefore be of heuristic value.

Modeling is an important scientific tool, in particular for the investigation of such complex systems as biological neural networks (BNN). Two approaches will be considered: (i) the methods of system theory or cybernetics that treat systems as having a small number of information channels and therefore consider such systems on a more abstract level, and (ii) the artificial neural networks (ANN) approach which reflects the fact that a great number of parallel channels are typical for BNNs.

1.1 Symbols

A system, in terms of systems theory, can be symbolized quite simply by a "black box" (Fig. 1.**3a**) with input and output channels. The input value is described by an input function x(t), the output value by an output function y(t), both depending on time t. If one was able to look inside the black box, the system might consist of a number of subsystems which are connected in different ways (e. g., Fig. 1.**3b**). One aim of this approach is to enable conclusions to be reached concerning the internal structure of the system. In the symbols used throughout this text, an arrow shows the direction of signal flow in each channel. Each channel symbolizes a value which is transported along the line without delay and without being changed in any way. A box means that the incoming values are changed. In other words, calculations are done only within a box. Summation of two input values is often shown by the symbol given in Fig. 1.**4a** (right side) or, in particular if there are more than two input channels, by a circle containing the summation symbol Σ (Fig. 1.**4a,** left). Figure 1.**4b** shows two symbols for subtraction. Multiplication of two (or more) input values is shown by a circle containing a dot or the symbol Π (Fig. 1.**4c**). The simplest calculation is the multiplication with a constant w. This is often symbolized by writing w within the box (Fig. 1.**4d**, right). Another possibility, mostly used when plotting massively parallel systems, is to write the multiplication factor w beside the arrowhead (Fig. 1.**4d**, left).

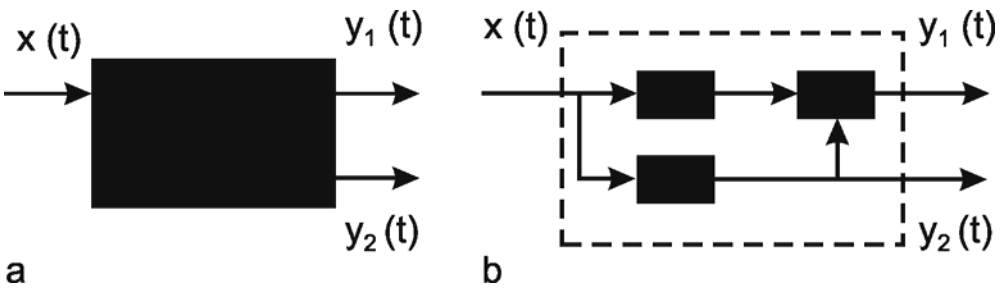


Fig. 1.**3 Schematic representation of a system.** (a) black box. (b) view into the black box. x(t) input function, $y_1(t)$,$y_2(t)$ output functions

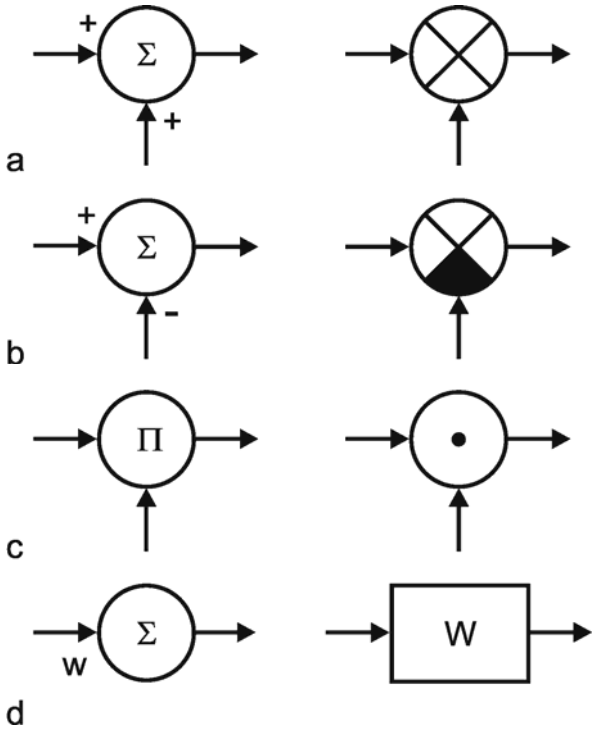


Fig. 1.**4 Symbols** used for (a) summation, (b) subtraction, and (c) multiplication of two (or more) input values and (d) for multiplication with a constant value

1.2 Linear Systems

We can distinguish between linear and nonlinear systems. A system is considered linear if the following input-output relations are valid. If y_1 is the output value belonging to the input value x_1, and y_2 that to the input value x_2, the input value $(x_1 + x_2)$ should produce the output value $(y_1 + y_2)$. In other words, this means that the output must be proportional to the input. If a system does not meet these conditions, we have a nonlinear system.

This will be illustrated in Figure 1.**5** by two selected examples. To begin with, Figure 1.**5a** shows the response to a step-like input function. If the input function rises sharply from the value zero to the value x, within this system the output function reaches the value y after a certain temporal delay. If the amplitude of the input function is doubled to 2 x, the values of the output function must be doubled, too, in case of a linear system (Fig. 1.**5b**, continuous line). If, however, the output function ($y^* \neq 2y$) shown in Figure 1.**5b** with a dashed line were achieved, it would be a nonlinear system.

Nonlinear properties occur, for example, if two channels are multiplied or if nonlinear characteristics occur within the system (see Chapter 5). Although the majority of all real systems (in particular, biological systems) are nonlinear in nature, we shall start with linear systems in Chapters 3 and 4. This is due to the fact that, so far, only linear systems can be described in a mathematically closed form. To what extent this theory of linear systems may be applied when investigating a real (i. e., usually nonlinear) system, must be decided for each individual case. Some relevant examples are discussed in Chapters 6 and 7. Special properties of nonlinear systems will be dealt with in Chapter 5.

Although biological systems are usually nonlinear in nature, linear systems will be considered first because they can be understood more easily.

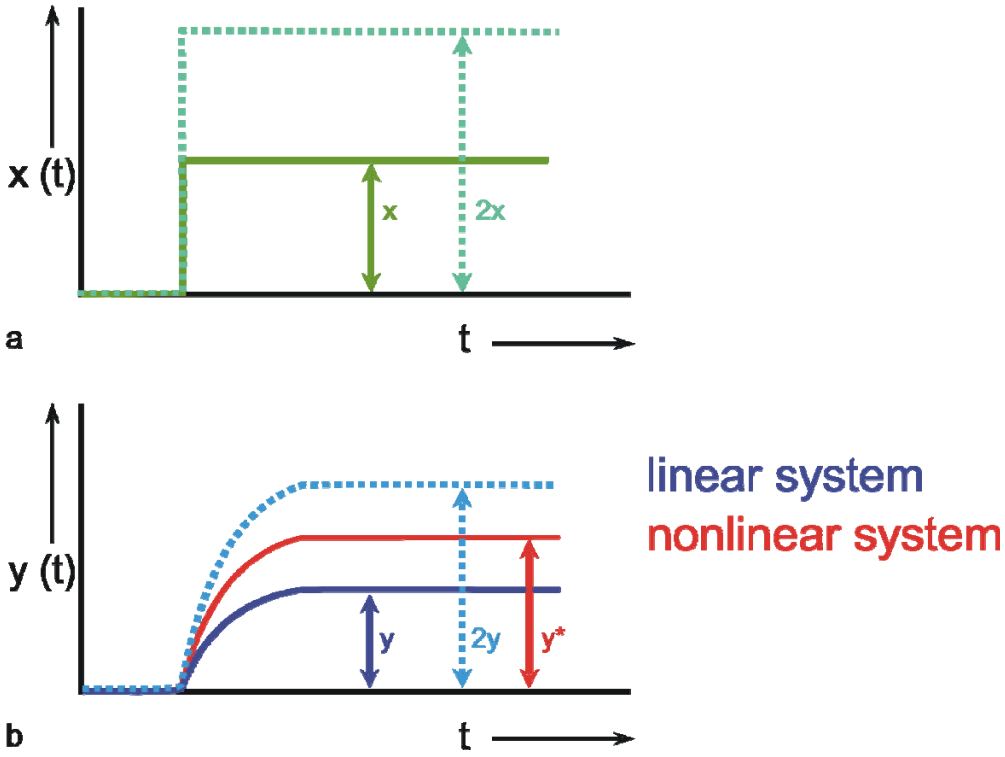


Fig. 1.**5 Examples of step responses of (a) a linear and (b) a nonlinear system.** x (t) input function, y (t) output function

1.3 The Tasks of Systems Theory Investigations

The first task of a systems theory investigation is to determine the input-output relations of a system, in other words, to investigate the overall properties of the box in Figure 1.**3 a**. For this so-called "black box" analysis (or input-output analysis, or system identification) certain input values are fed into the system and then the corresponding output values are measured. For the sake of easier mathematical treatment, only a limited number of simple input functions is used. These input functions are described in more detail in Chapter 2. In Figure 1.**3b** it is shown that the total system may be broken down into smaller subsystems, which are interconnected in a specific way. An investigation of the input-output relation of a system is thus followed immediately by the question as to the internal structure of the system. The second task of systems-theory research in the field of biology in thus to find out which subsystems form the total system and in what way these elements are interconnected. This corresponds to the step from Figure 1.**3a** to Figure 1.**3b**. In this way, we can try to reduce the system to a circuit which consists of a number of specifically connected basic elements. The most important of these basic elements will be explained in Chapter 4.

For the black box analysis, the system is considered to be an element that transforms a known input signal into an unknown output signal. Taking this input-output relation as a starting point, attempts are made to draw conclusions as to how the system is constructed, i. e., what elements it consists of and how these are connected. (When investigating technical systems, usually the opposite approach is required. Here, the structure of the system is known and it is intended to determine, without experiments, the output values which follow certain input values that have not yet been investigated. However, such a situation can also occur in biology. For example, a number of local mechanisms of an ecosystem might have been investigated. Then the question arises to what extent the properties of the complete system could be explained?)

If the total system has been described at this level, we have obtained a quantitative model of a real system and the black box analysis is concluded. The next step of the investigation (i. e., relating the individual elements of the model to individual parts of the real system) is no longer a task of systems theory and depends substantially on knowledge of the system concerned. However, the model developed is not only a description of the properties of the overall system, but also is appropriate to guide further investigation of the internal structure of the system.

The first task of the systems theory approach is to describe the input-output properties of the system, such that its response to any input function can be predicted. The next step is to provide conclusions with respect to the internal structure of the system.

2 The Most Important Input Functions

As was mentioned before, there are only a few functions usually used as input functions for investigating a system. Five such input functions will be discussed below. The investigation of linear systems by means of each of these input functions yields, in principle, identical information about the system. This means that we might just as well confine ourselves to studying only one of these functions. For practical reasons, however, one or the other of the input functions may be preferred in individual cases. Furthermore, and more importantly, this fundamental identity no longer applies to nonlinear systems. Therefore, the investigation of nonlinear systems usually requires the application of more than one input function.

2.1 The Step Function

Figure 2.**1 a** shows a step function. Here, the input value-starting from a certain value (usually zero)-is suddenly increased by a constant value at time $t = t_0$ and then kept there. In theoretical considerations, the step amplitude usually has the value 1; therefore this function is also described as unit step, abbreviated to x(t) = 1(t). The corresponding output function (the form of which depends on the type of system and which therefore represents a description of the properties of the system) is called accordingly step response h(t) or the transient function.

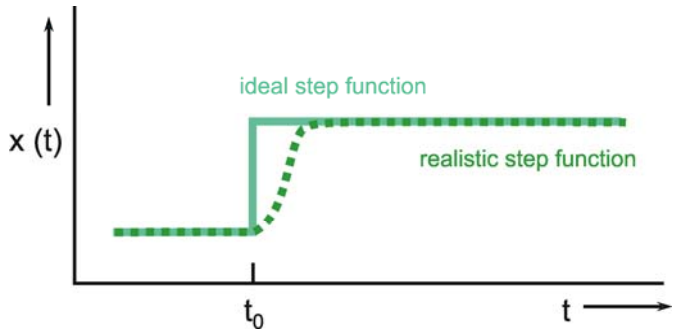


Fig. 2.**1 The step function.** Ideal (continuous line) and realistic version (dashed line)

The problem with using a step function is that the vertical slope can be determined only approximately in the case of a real system. In fact, it will only be possible to generate a "soft" trapezoidal input function, such as indicated in Figure 2.**1 b**. As a rule, the step is considered to be sufficiently steep if the rising movement of the step has finished before an effect of the input function can be observed at

the output. This step function is, for example, particularly suitable for those systems where input values are given through light stimuli, since light intensity can be varied very quickly. In other cases, a fast change in the input function may risk damaging the system. This could be the case where, for example, the input functions are to be given in the form of mechanical stimuli. In this situation, different input functions should be preferred.

The response to a step function is called step response or transient function h(t).

2.2 The Impulse Function

As an alternative to the step function, the so-called impulse function x(t) = δ(t) is also used as input function. The impulse function consists of a brief pulse, i. e., at the time t = t_0 the input value rises to a high value A for a short time (impulse duration Δt) and then immediately drops back to its original value. This function is also called Dirac, needle, δ, or pulse function. Mathematically, it can be defined as A Δt = 1, where Δt approximates zero, and thus A is growing infinitely. Figure 2.**2** gives two examples of possible forms of the impulse function.

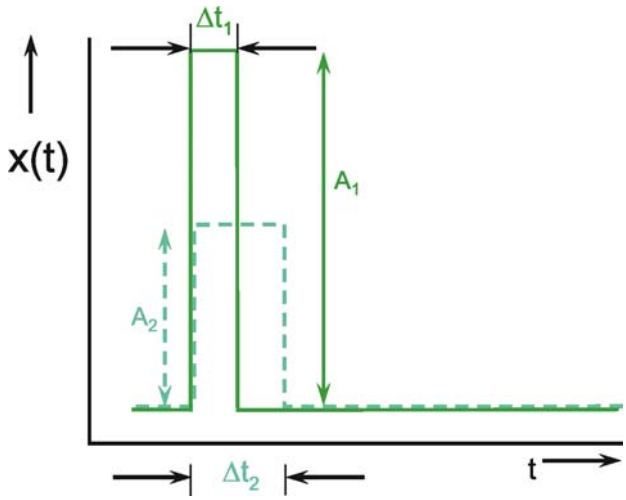


Fig. 2.**2 Two examples of approximations of the impulse** function. A impulse amplitude, Δt impulse duration

It is technically impossible to produce an exact impulse function. Even a suitable approximation would, in general, have the detrimental effect that, due to very high input values, the range of linearity of any system will eventually be exceeded. For this reason, usually only rough approximations to the exact impulse function are possible.

As far as practical application is concerned, properties of the impulse function correspond to those of the step function. Apart from the exceptions just mentioned, the impulse function can thus be used in all cases where the step function, too, could prove advantageous. This, again, applies particularly to photo reactions, since flashes can be generated very easily.

The response function to the impulse function is called impulse response or weighting function g(t). (In some publications, the symbols h(t) and g(t) are used in just the other way.) The importance of the weighting function in systems theory is that it is a means of describing a linear system completely and simply. If the weighting function g(t) is known, it will be possible to calculate the output function y(t) of he investigated system in relation to any input function x(t). This calculation is based on the following

formula: (This is not to be proved here; only arguments that can be understood easily are given.)

$$y(t) = {}_0\int^t g(t-t')\, x(t')\, dt'$$

This integral is termed a convolution integral, with the kernel (or the weighting function) g(t).

Figure 2.**3** illustrates the meaning of this convolution integral. First, Figure 2.**3a** shows the assumed weighting function g(t') of an arbitrary system. By means of this convolution integral the intent is to calculate the response y(t) of the system to an arbitrary input function x(t') (Fig. 2.**3b**). (t and t' have the same time axis, which is shown with different symbols in order to distinguish between time t and the integration variable t'.)

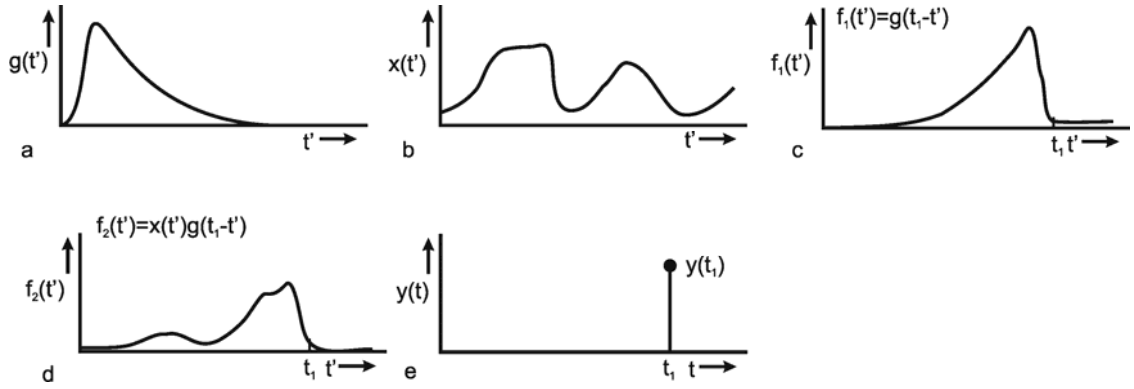


Fig. 2.**3 The calculation of the convolution integral.** (a) The weighting function g(t) of the system. (b) An arbitrary input function x(t'). (c) The signal of the weighting function g(t') is changed to g(-t') and it is shifted by t_1, leading to $f_1 = g(t_1 - t')$. (d) The product $f_2(t')$ of the input function x(t') and $g(t_1 - t')$. (e) The integral y(t) of this function $f_2(t')$ at time t_1

To begin with, just the value of the output function $y(t_1)$ at a specific point in time t_1 will be calculated. As the weighting function quickly approaches zero level, those parts of the input function that took place some time earlier exert only a minor influence; those parts of the input function that occurred immediately before the point in time t_1 still exert a stronger influence on the value of the output function. This fact is taken into account as follows: The weighting function g(t') is reflected at the vertical axis, g(-t'), and then shifted to the right by the value t_1, to obtain $g(t_1-t')$, which is shown in Figure 2.**3c**. This function is multiplied point for point by the input function x(t'). The result $f_2(t') = x(t')g(t_1 - t')$ is shown in Figure 2.**3d**. Thus, the effect of individual parts of the input function x(t') on the value of the output function at the time t, has been calculated. The total effect is obtained by summation of the individual effects. For this, the function $f_2(t')$ is integrated from 0 to t_1, i. e., the value of the area below the function of Figure 2.**3d** is calculated. This results in the value $y(t_1)$ of the output function at the point in time t_1 (Fig. 2.**3e**).

This process should be carried out for any t, in order to calculate point for point the output function y(t). If the convolution integral can be solved directly, however, the output function y(t) can be presented in a unified form.

The relation between weighting function (impulse response), g(t), and step response h(t) is as follows: $g(t) = dh(t)/dt$; $h(t) = {}_0\int^t g(t')\, dt'$. Thus, if h(t) is known, we can calculate g(t) and vice versa. As outlined above, in the case of a linear system we can obtain identical information by using either the impulse function or the step function.

The response to an impulse function is called impulse response or weighting function g(t). Convolution of g(t) with any input function allows the corresponding output function to be determined.

2.3 The Sine Function

In addition to the step and impulse functions, sine functions are quite frequently used as input functions. A sine function takes the form $x(t) = a \sin 2\pi\nu t$ (Fig. 2.**4**). a is the amplitude measured from the mean value of the function. ν (measured in Hertz = s^{-1}) is the frequency of the oscillation. The expression $2\pi\nu$ is usually summarized in the so-called angular frequency ω: $x(t) = a \sin \omega t$. The duration of one period is $1/\nu$.

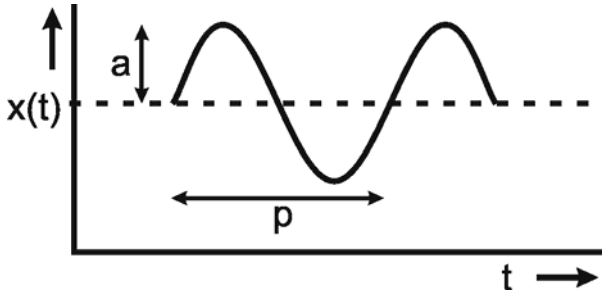


Fig. 2.**4 The sine function.** Amplitude a, period $p = 1/\nu$. The mean value is shown by a dashed line

Compared to step and impulse functions, the disadvantages of using the sine function are as follows: in order to obtain, by means of sine functions, the same information about the system as if we were using the step response or impulse response, we would have to carry out a number of experiments instead of (in principle) one single one; i. e., we would have to study, one after the other, the responses to sine functions of constant amplitude but, at least in theory, to all frequencies between arbitrary low and infinitely high frequencies. In practice, however, the investigation is restricted to a limited frequency range which is of particular importance to the system. Nevertheless, this approach requires considerably more measurements than if we were to use the step or impulse function. Moreover, we have to pay attention to the fact that, after the start of a sinusoidal oscillation at the input, the output function, in general, will build up. The respective output function can be analyzed only after this build-up has subsided. The advantage of using sine functions is that the slope of the input function, at least within the range of frequencies of interest, is lower than that of step functions, which allows us to treat sensitive systems with more care.

The response function to a sine function at the input is termed frequency response. In a linear system, frequency responses are sine functions as well. They possess the same frequency as the input function; generally, however, they differ in amplitude and phase value relative to the input function. In this case, the frequency responses thus take the form $y(t) = A \sin(\omega t + \varphi)$. The two values to be measured are therefore the phase shift φ between input and output function as well as change of the amplitude value. The latter is usually given as the ratio between the maximum value of the output amplitude A and that of the input amplitude a, i. e., the ratio A/a. Frequently, the value of the input amplitude a is standardized to 1, so that the value of the output amplitude A directly indicates the relation A/a. In the following this will be assumed in order to simplify the notation.

Both values, output amplitude A as well as phase shift φ, depend on the frequency. Therefore, both values are measured with a constant input amplitude (a = 1), but different angular frequencies ω. If the

output amplitudes are plotted against the angular frequency ω, the result will be the amplitude frequency plot A(ω) of the system (Fig. 2.**5a**). Usually, both amplitude and frequency are plotted on a logarithmic scale. As illustrated in Figure 2**.5a** by way of a second ordinate, the logarithms of the amplitude are sometimes given. The amplitude frequency plot shown in Figure 2.**5a** shows that the output amplitude with a medium frequency is almost as large as the input amplitude, whereas it gets considerably smaller for lower and higher frequencies. Very often the ordinate values are referred to as amplification factors since they indicate the factor by which the size of the respective amplitude is enlarged or reduced relative to the input amplitude. Thus, the amplification factor depends on the frequency. (It is called dynamic amplification factor, as is explained in detail in Chapter 5).

An amplification factor is usually defined as a dimensionless value and, as a consequence, can only be determined if input and output values can be measured on the same dimension. Since this is hardly ever the case with biological systems, we have sometimes to deviate from this definition. One possibility is to give the amplification factor in the relative unit such as decibels (dB). For this purpose, one chooses an optional amplitude value A_0 as a reference value (usually the maximum amplitude) and calculates the value 20 lg (A_1/A_0) for every amplitude A_1. A logarithmic unit in Fig. 2.**5a** thus corresponds to the value 20 dB. In Figure 2.**5a** (right ordinate) the amplitude A_0 = 1 takes the value 0 dB.

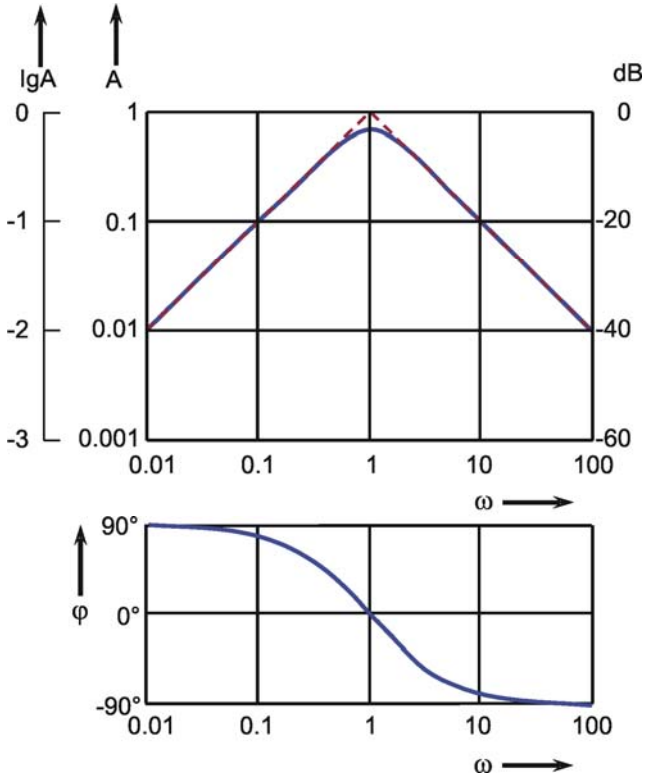


Fig. 2.**5 The Bode plot consisting of the amplitude frequency plot (a) and the phase frequency plot (b).** A amplitude, φ phase, ω = 2πν angular frequency. In (a) the ordinate is given as lg A (far left), as A (left), and as dB (decibel, right)

In order to obtain complete information about the system to be investigated, we also have to determine changes in the phase shift between the sine function at the input and that at the output. By plotting the phase angle in a linear way versus the logarithm of the angular frequency, we get the phase frequency plot φ(ω) (Fig. 2.**5b**). A negative phase shift means that the oscillation at the output lags behind that of the input. (Due to the periodicity, however, a lag in phase by 270° is identical with an advance in

phase by 90°). The phase frequency plot in Figure 2.**5b** thus means that in cases of low frequencies the output is in advance of the input by 90°, with medium frequencies it is more or less in phase, whereas at high frequencies it is lagging by about 90°.

Both descriptions, amplitude frequency plot and phase frequency plot, are subsumed under the term Bode plot. The significance of the Bode plot is dealt with in detail in Chapters 3 and 4. We shall now briefly touch on the relationship between Bode plot and weighting function of a system. By means of a known weighting function g(t) of a system, it is possible to calculate, by application of the convolution integral, the sine response using the method explained in Chapter 2.2:

$$y(t) = {}_{o}\int^{t} g(t-t') \sin \omega t' \, dt'.$$

The solution will be:

$$y(t) = A(\omega) \sin(\omega t + \varphi(\omega))$$

where $A(\omega)$ is the amplitude frequency plot and $\varphi(\omega)$ the phase frequency plot of the system (for proof, see e. g., Varju 1977). Thus, the sine function as an input function also gives us the same information about the system to be investigated as the step function or the impulse function.

As an alternative to the Bode plot, amplitude and phase frequency can be indicated in the form of the Nyquist or polar plot (Fig. 2.**6**). Here, the amplitude A is given by the length of the pointer, the phase angle is given by the angle between the x-axis and pointer. The angular frequency ω is the parameter, i.e., ω passes through all values from zero to infinity from the beginning to the end of the graph. In the example given in Figure 2.**6**, the amplitude decreases slightly at first in the case of lower frequencies; with higher frequencies, however, it decreases very rapidly. At the same time, we observe a minor lag in phase of output oscillation as compared to input oscillation if we deal with lower frequencies. This lag is increases to a limit of -180°, however, if we move to high frequencies.

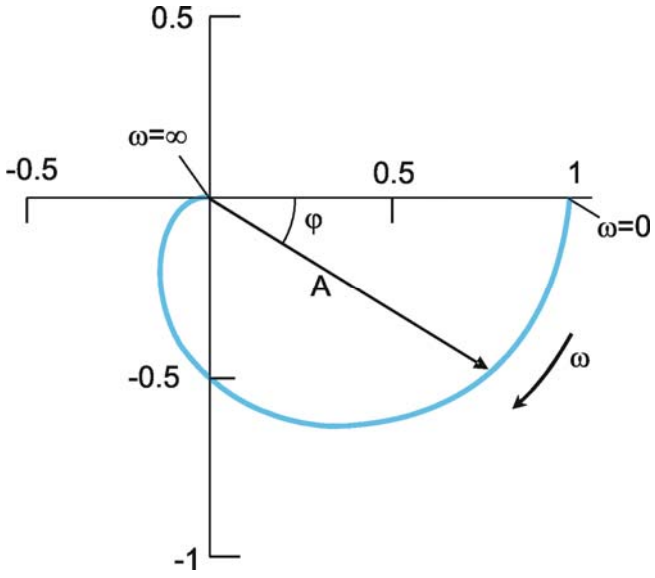


Fig. 2.**6 The polar plot.** A amplitude, φ phase, $\omega = 2\pi\nu$ angular frequency

The response to sine functions is called frequency response and is graphically illustrated in the form of a Bode plot or a polar plot.

2.4 The Statistical Function

An input function used less frequently is the statistical or noise function. It is characterized by the fact that individual function values follow one another statistically, i.e., they cannot be inferred from the previous course of the function. Figure 2.**7** shows a schematic example. Investigating a system by means of a noise function has the advantage over using sine functions insofar as in principle, again, just one experiment is required to characterize the system completely. On the other hand, this can only be done with some mathematical work and can hardly be managed without a computer. (To do this, we mainly have to calculate the cross-correlation between input and output function. This is similar to a convolution between input and output function and provides the weighting function of the system investigated). Another advantage of the noise function over the sine function is that the system under view cannot predict the form of the input function. This is of particular importance to the study of biological systems, since here, even in simple cases, we cannot exclude the possibility that some kind of learning occurs during the experiment. Learning would mean, however, that the system has changed, since different output functions are obtained before and after learning on the same input function. By using the noise function, learning activities might be prevented. Changes of the property of the system caused by fatigue (another possible problem when investigating the frequency response) generally do not occur, due to the short duration of the experiment. The statistical function is particularly suitable for the investigation of nonlinear systems because appropriate evaluation allows nonlinear properties of the system to be judged (Marmarelis and Marmarelis 1978).

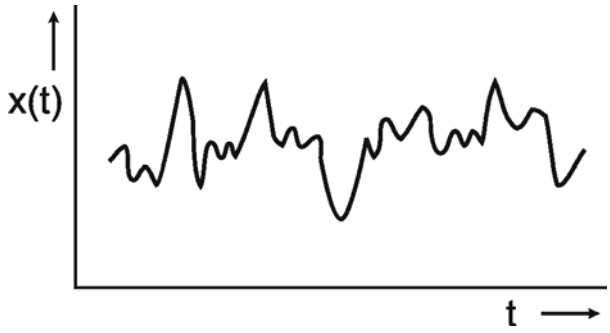


Fig. 2.**7 An example of the statistical function**

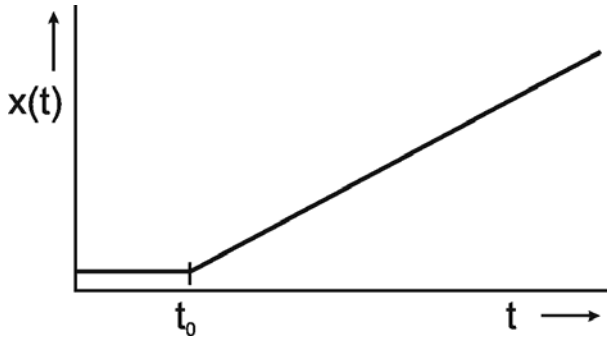


Fig. 2.**8 The ramp function** x(t) = kt (for $t > t_0$)

Application of the statistical function is technically more difficult, both in experimental and in theoretical terms, but permits treatment of some properties of nonlinear systems.

2.5 The Ramp Function

The last of the input functions described here is the ramp function x(t) = kt. The ramp function has a constant value, e. g., zero, for $t \leq -t_0$ and keeps rising with a constant slope k for $t > t_0$ (Fig. 2.**8**). Whereas the sine response requires us to wait for the building-up process to end before we can start the actual measurement, the very opposite is true here. As will be explained later (Chapter 4), interesting information can be gathered from the first part of the ramp response. This also holds for the step and impulse functions. However, a disadvantage of the ramp function is the following. Since the range of possible values of the input function is normally limited, the ramp function cannot increase arbitrarily. This means that the ramp function may have to be interrupted before the output function (the ramp response) has given us enough information. In this case, a ramp-and-hold function is used. This means that the ramp is stopped from rising any further after a certain input amplitude has been reached and this value is maintained.

The advantage of the ramp function lies in the fact that it is a means of studying, in particular, those systems that cannot withstand the strong increase of input values necessary for the step function and the impulse function. A further advantage is that it can be realized relatively easily from the technical point of view.

For the investigation of nonlinear systems, ramp function and step function (contrary to sine function and impulse function) have the advantage that the input value moves in one direction only. If we have a system in which the output value is controlled by two antagonists, we might be able to study both antagonistic branches separately by means of an increasing and (through a further experiment) a decreasing ramp (or step) function. In biology, systems composed of antagonistic subsystems occur quite frequently. Examples are the control of the position of a joint by means of two antagonistic muscles or the control of blood sugar level through the hormones adrenaline and insulin. At the input side, too, there may be antagonistic subsystems such as, for example, the warm receptors that measure skin temperature, and which cause the frequency of action potentials to increase with a rise in temperature, and the cold receptors, causing the frequency of action potentials to decrease with a lower outside temperature.

The ramp function is easy to produce but might have difficulties experimentally because of the limited range of input values.

3 The Fourier Analysis

We cannot deal with the characteristics of simple elements in the next chapter until we have explained the significance of the Bode plot in more detail. For this reason we introduce here the concept of Fourier analysis. Apart from a number of exceptions that are of interest to mathematicians only, any periodic function y = f(t) may be described through the sum of sine (or cosine) functions of different amplitude, frequency and phase:

$$y(t) = a_0 + a_1 \sin(\omega t + \varphi_1) + a_2 \sin(2\omega t + \varphi_2) + ... + a_k \sin(k\omega t + \varphi_k) + ..$$

$$= \sum_{k=0}^{\infty} a_k \sin(k\,\omega t + j_k)$$

This is called a frequency representation of the function y(t) or the decomposition of the function to individual frequencies. The amplitude factors of the different components (a_k) are called Fourier coefficients.

The concept of the Fourier decomposition can be demonstrated by the example of a square function (Fig. 3.**1,** y_S). The Fourier analysis of this square function yields:

$$y(t)=\frac{4}{\pi}\sin 2\pi t+\frac{4}{3\pi}\sin 6\pi t+\frac{4}{5\pi}\sin 10\pi t+..+\ =\sum_{k=0}^{\infty}\frac{4}{k\pi}\sin 2\,k\pi t \quad \text{for } k=1,3,5,...$$

Frequency components that have an even numbered index thus disappear in this particular case. The first three components and their sum (y') have also been included in Figure 3.**1**.

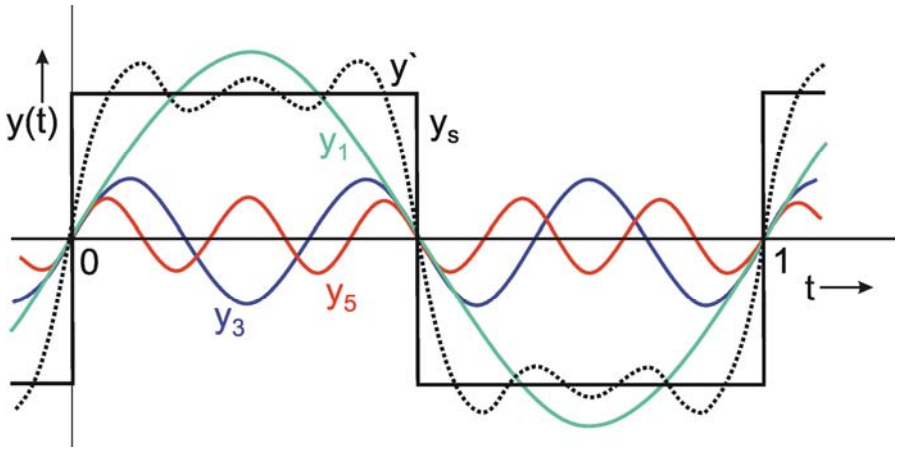


Fig.3.**1 Fourier decomposition of a square function ys.** The fundamental $y_1 = 3/\pi \sin 2\pi t$, and the two harmonics $y_3 = 4/(3\pi) \sin 6\pi t$ and $y_5 = 4/(5\pi) \sin 10\pi t$ are shown together with the sum $y' = y_1 + y_3 + y_5$ (dotted line).

As we can see, a rough approximation to the square function has already been achieved by y'. As this figure shows, in particular for a good approximation of the edges, parts of even higher frequency are required. The component with the lowest frequency (y_1 in Fig. 3.**1**) is called the fundamental, and any higher-frequency parts are referred to as harmonic components. In these terms we might say that the production of sharp edges requires high-frequency harmonic waves.

This concept permits an easy qualitative interpretation of a given amplitude frequency plot. As can be seen from its amplitude frequency plot shown in Figure 3.**2a**, a system which transmits low frequencies well and suppresses high frequency components affects the transmission of a square function insofar as, in qualitative terms, the sharp edges of this function are rounded off. As Figure 3.**1** shows, high-frequency components are required for a precise transmission of the edges. If the amplitude frequency plot of the system looks like that shown in Figure 3.**2b**, i. e., low frequencies are suppressed by the system whereas high frequency components are transmitted well, we can get a qualitative impression of the response to a square wave in the following way. Providing we subtract the fundamental wave y_1 from the square function y_s which would occur where all frequencies are transmitted equally well, it would turn out that there would be a good transmission within the range of the edges and a poor transmission within the range of the constant function values, if the lower frequencies were suppressed. As a result, the edges would be emphasized particularly strongly.

Due to the fact that, in most systems (see, however, Chapter 10), different frequency components undergo different phase shifts which we have not discussed so far, these considerations are only of a qualitative nature. The quantitative transfer properties of some systems will be dealt with in Chapter 4. It can already be seen, however, that the amplitude frequency plot allows us to make important qualitative statements concerning the properties of the system.

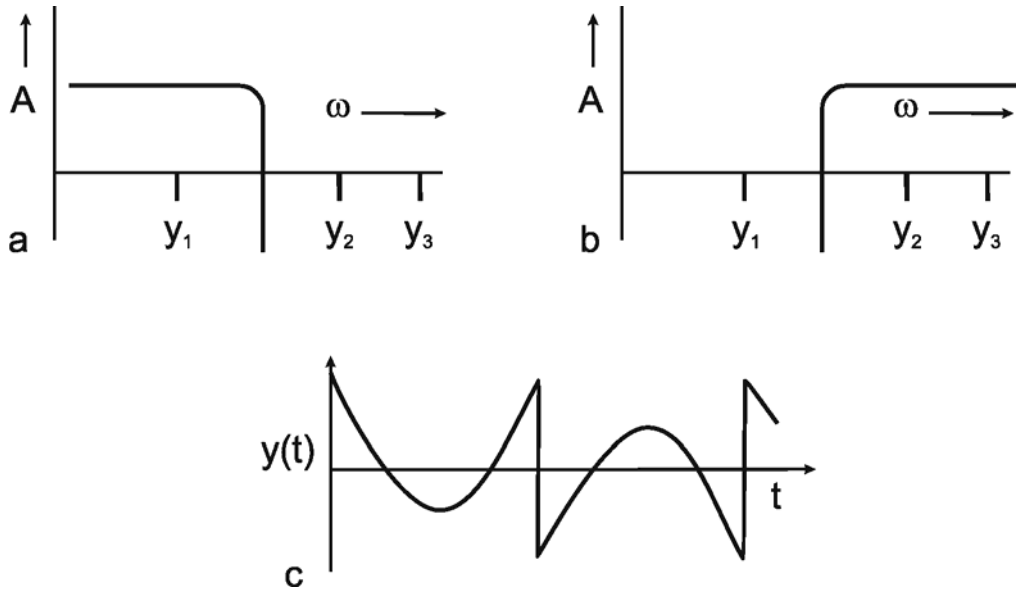


Fig. 3.**2 Assume a system having a negligible phase shift and showing an amplitude frequency plot which transmits low frequencies like that of the fundamental y1 in Fig. 3.1 but suppresses higher frequencies like y3 or y5.** A square wave as shown in Fig. 3.1 given at the input of this system would result in an output corresponding to the sine wave of y_1. When the system is assumed to have a mirror-image-like amplitude frequency plot, (b), the response to the square wave input would look like the square wave without its fundamental. This is shown in (c)

We are now able to realize more clearly why investigation of a linear system by means of step or impulse functions requires, in principle, only one experiment to enable us to describe the system, whereas in the case of an investigation by means of sine functions we need to carry out, theoretically, an infinite number of experiments. The reason is that the edges of step and impulse functions already contain all frequency components. These have to be studied separately, one after another, if we use sine functions.

The Fourier analysis shows that sharp edges of a function are represented by high frequency components and that low frequencies are necessary to represent the flat, horizontal parts of the function.

4 The Properties of the Most Important Filters

So far we have only considered how to obtain information about the transfer properties and thus to describe the system to be investigated; now we have a further task: how to determine the internal structure of the system from the form of the Bode plot or the weighting function. This is done in the following way: we try to break down the total system into elements, called filters. Below we describe the properties of the most important filters. Subsequently, using an example based on the transfer

properties of a real system, we will show how to obtain the structure of the system by suitably composing elementary filters.

4.1 First Order Low-Pass Filter

A low-pass filter allows only low frequencies to pass. As can be seen in the amplitude frequency plot of the Bode plot (Fig. 4.**1a**), low frequency sine waves are transmitted by the amplification factor 1; high frequency sine waves, however, are suppressed. The graph of the amplitude-frequency plot may be approximated by means of two asymptotes. One of those runs horizontally and in Figure 4.**1 a** corresponds to the abscissa. The second asymptote has a negative slope and is shown as a dashed line in Figure 4.**1 a.** If the coordinate scales are given in logarithmic units, this line has the slope -1. The intersection of both asymptotes marks the value of the *corner frequency* ω_0, also called the *characteristic frequency.* At this frequency, the output amplitude drops to about 71 % of the maximum value reached with low frequencies. In the description chosen here, ω_0 is set to 1. By multiplying the abscissa values with a constant factor, this Bode plot can be changed into that of a low-pass filter of any other corner frequency.

As the phase-frequency plot shows, there is practically no phase shift with very low frequencies; the phase-frequency plot approaches zero asymptotically. With very high frequencies, there is a lag in phase of the output oscillation, compared to the input oscillation, by a maximum of 90°. In the case of the corner frequency ω_0 (= $2\pi\nu_0$) the phase shift comes to just 45°.

Besides the sine function, the other important input function is given by the step function. The transient, or step response h(t) of this low-pass filter is shown in Figure 4.**1 b.** It starts at t = 0 with a slope of $1/\tau$ and asymptotically approaches horizontal. The amplitude value of this line indicates the static amplification factor of the system, which has been assumed to be 1 in this example. This static amplification factor corresponds to the amplification factor for low frequencies which can be read off from the amplitude-frequency plot. The *time constant* τ of this low-pass filter can be read from the step response to be the time at which the response has reached 63% of the ultimate value, measured from time t = 0.The transition phase of the function is also referred to as the dynamic part and the plateau area as the static part of the response (see also Chapter 5). For a decreasing step, we obtain an exponential function with negative slope.

Here, the time constant indicates the time within which the exponential function $e^{-1/\tau}$ from an amplitude value A has dropped to the amplitude value A/e, i. e., to about 37 % of the value A.

Corner frequency ω_0 and time constant τ of this low-pass filter (and of the corresponding high-pass filter, which will be described in Chapter 4.2) are related in the following way:

$$\omega_0 = \frac{1}{\tau}$$

Thus, the time constant of this low-pass filter can also be obtained from the Bode plot, to be τ = 1 s. (In the case of the Bode plot, and in all of the following figures in Chapter 4, the response functions of filters for other corner frequencies and time constants can be obtained by expansion of the abscissa by an appropriate factor). The formula for the impulse response shown in Figure 4.**1 b** is only applicable for the range t > 0. For the sake of simplicity, this restriction will not be mentioned further in the corresponding cases that follow.

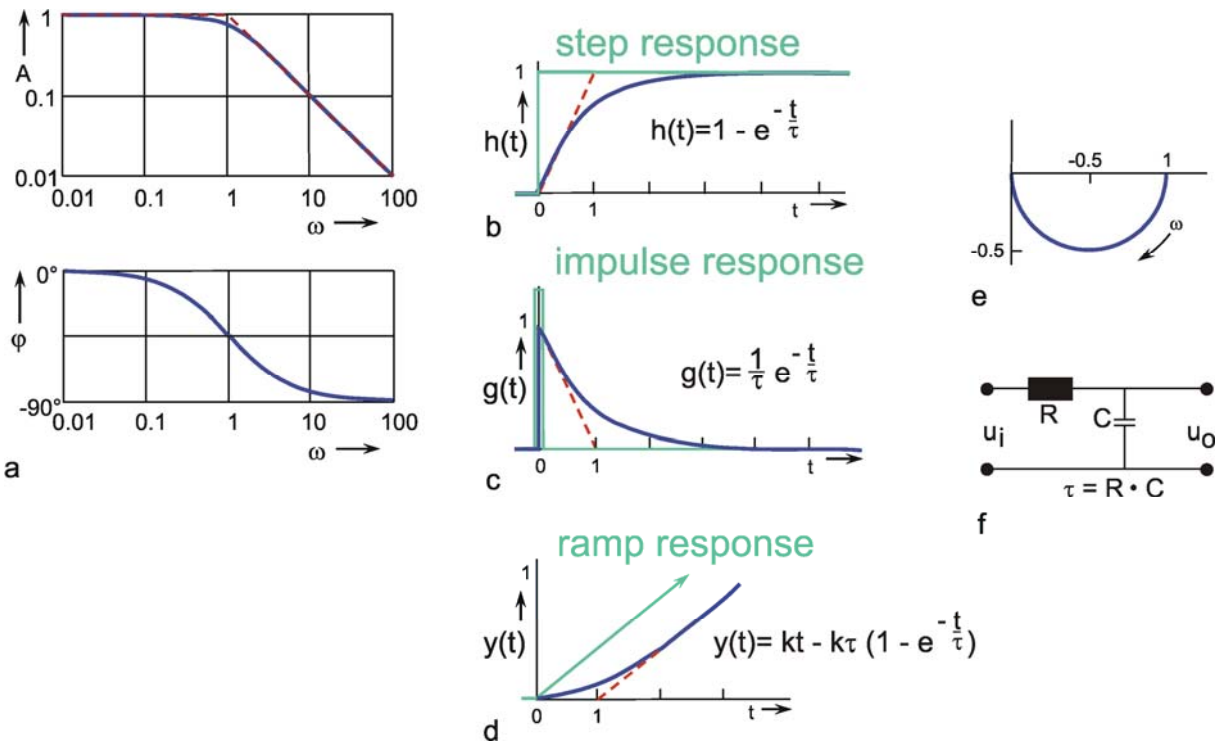


Fig. 4.**1 Low-pass-filter.** (a) Bode plot consisting of the amplitude frequency plot (above) and the phase frequency plot (below). Asymptotes are shown by red dashed lines. A amplitude, φ phase, $\omega = 2\pi\nu$ angular frequency, (b) step response, (c) impulse response, (d) ramp response. In (b)-(d) the input functions are shown by green lines. The formulas given hold for t > 0. k describes the slope of the ramp function x(t) = kt. (e) polar plot. (f) Electronic wiring diagram, R resistor, C capacitor, τ time constant, u_1 input voltage, u_0 output voltage

The easiest way to check whether a given function is an exponential one is to plot it on a semilogarithmic scale (with logarithmic ordinate and linear abscissa). This will result in a line with the slope $-1/\tau$. Quite often the half-time or half-life is used as a measure of duration of the dynamic part instead of the time constant τ. It indicates the time by which the function has reached half of the static value. Given a filter of the first order (also true for the corresponding high-pass filter, see Chapter 4.2) the relation between halftime (HT) and time constant is as follows:

$$HT = \tau \ln 2 = 0.7\,\tau.$$

Because of the step response, a low-pass filter may be understood intuitively to be an element which though it tries to follow the input value proportionally, succeeds only after a temporal lag due to some internal sluggishness. Figure 4.**2** shows an example of low-pass-like behavior. Here, the force produced by a muscle is shown when the frequency of action potentials within the relevant motoneuron has been changed.

The impulse response, i. e., the weighting function g(t) of this low-pass filter, is shown in Figure 4.**1 c**. If the input value zero has been constant for a long time, the output value is, at first, also zero. During the application of the impulse function at time t = 0 however, the output jumps to the value $1/\tau$ and then, starting with the slope $-1/\tau^2$ decreases exponentially, with the time constant τ, to zero. The maximum amplitude $1/\tau$ can be reached only if an ideal impulse function is used, however.

The weighting function of this lowpass filter is described by:

$$g(t) = \frac{1}{\tau}\, e^{-t/\tau}$$

Figure 4.**1 d** shows the ramp response of a low-pass filter. It starts with a horizontal slope at the beginning of the ramp and asymptotically approaches a straight line that is parallel to the input function x(t) = kt. This line intersects the time axis at the value τ. For the sake of completeness, the Nyquist plot has also been given in Figure 4.**1e**. However, this will not be described in more detail here or in further examples, since it contains the same information given in the Bode plot, as we have already mentioned above.

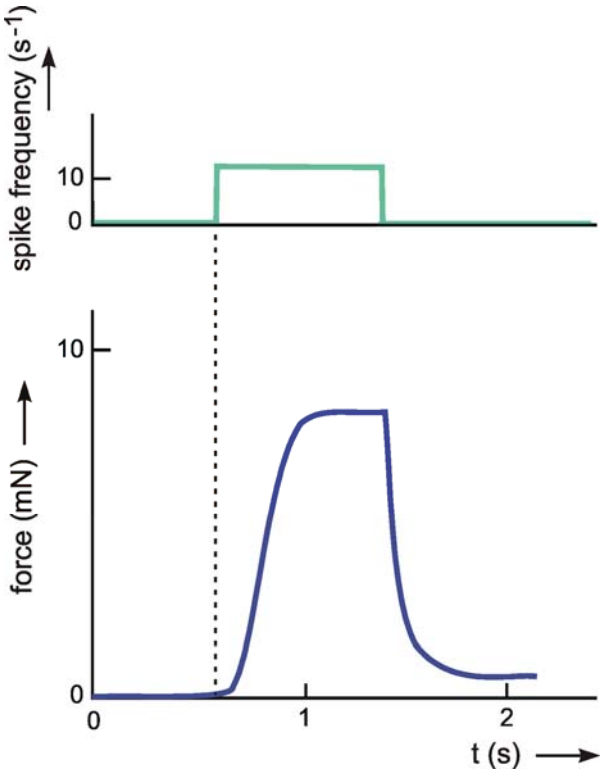


Fig. 4.**2 The response of an insect muscle to an increasing and a decreasing step function.** The input function (upper trace) is given by the experimentally induced spike frequency of the motorneuron. The output function (lower trace) shows the force developed by the muscle (after Iles and Pearson 1971)

Systems that possess low-pass characteristics can be found everywhere, since - for physical reasons - there is no system that is able to transmit frequencies with no upper limit. In order to investigate the properties of such a filter, it is most convenient to construct it of electronic components. The simplest electronic low-pass filter is shown in the wiring diagram in Figure 4.**1 f**. It consists of an ohmic resistor R and a capacitor C. The input function x(t), given as voltage, is clamped at u_i, the output function y(t) is measured as voltage at u_0. The time constant of this low-pass filter is calculated by τ = RC. Electronic circuits of this kind are also indicated accordingly for systems discussed later. Alternatively, a low-pass filter can be simulated on a digital computer as described in Appendix II.

A low-pass filter describes a system with sluggishness, i. e., one in which high frequency components are suppressed. Dynamic properties are described by the time constant τ or the corner frequency ω_0.

4.2 First Order High-Pass Filter

A high-pass filter permits only high frequencies to pass, while suppressing the low frequency parts. This is shown by the amplitude frequency, plot in Fig. 4.**3a**, which is the mirror image of tat of the low-pass filter (Fig. 4.**1**). When logarithmic coordinates are used, the asymptote along the descending branch of the amplitude frequency plot takes the slope + 1. As with the low pass filter, the corner frequency is indicated by the intersection of the two asymptotes. Similarly, the time constant is calculated according to $\tau = 1/\omega_0$. Here though, the corner frequency indicates that those oscillations are permitted to pass whose frequencies are higher than the corner frequency, while oscillations whose frequencies are markedly lower than the corner frequency, are suppressed.

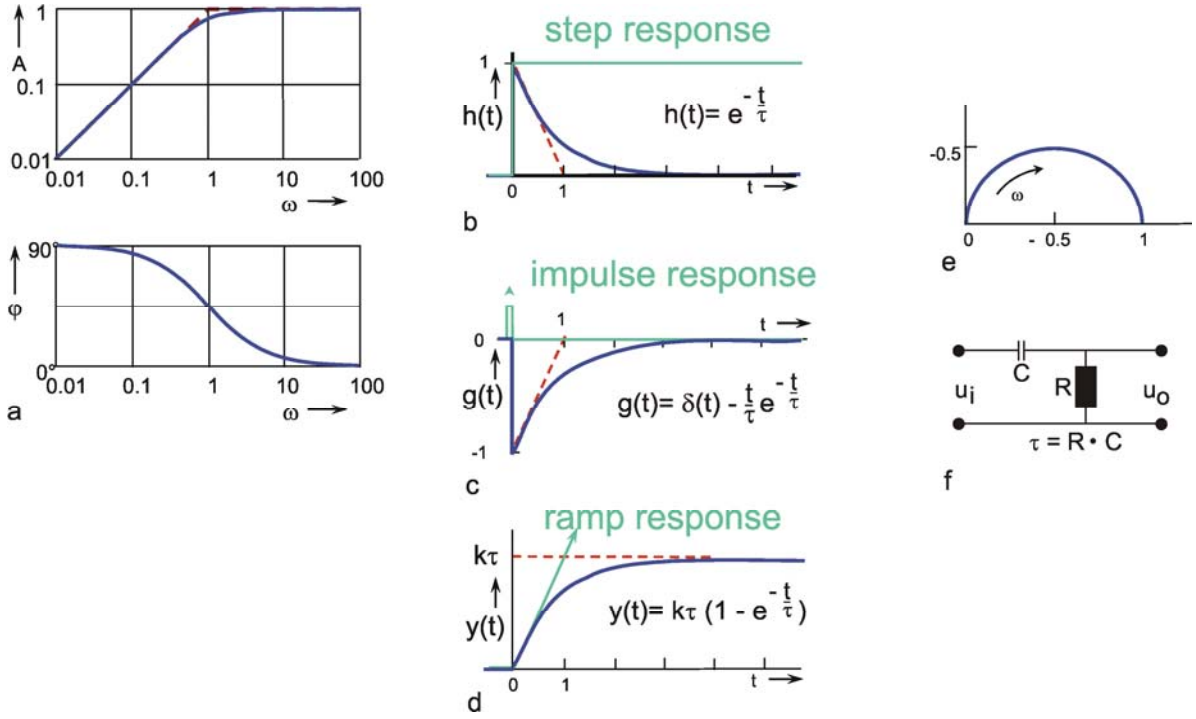


Fig. 4.**3 High-pass filter**. (a) Bode plot consisting of the amplitude frequency plot (above) and the phase frequency plot (below). Asymptotes are shown by red dashed. A amplitude, φ phase, $\omega = 2\pi\nu$ angular frequency. (b) step response, (c) impulse response, (d) ramp response. In (b) - (d) the input functions are shown by green lines. The formulas given hold for t > 0. k describes the slope of the ramp function x(t) = kt. (e) polar plot. (f) Electronic wiring diagram, R resistor, C capacitor, τ time constant, u_1 input voltage, u_o output voltage

As with the low-pass filter shown in Figure 4.**1**, the corner frequency is assumed to be $\omega_0 = 1\ s^{-1}$ and the time constant to be $\tau = 1$ s. When examining the phase-frequency plot, we notice a phase lead of output oscillation compared to input oscillation by maximally + 90° for low frequencies. For the corner frequency it amounts to +45°, asymptotically returning to zero value for high frequencies.

The step response (Fig. 4.**3b**) clearly illustrates the qualitative properties of a high-pass filter. With the beginning of the step, the step response instantly increases to the value 1, and then, with the initial slope -1 /τ and the time constant τ, decreases exponentially to zero, although the input function

continues to exhibit a constant value of 1. The maximum amplitude takes value 1 since, as can be gathered from the location of the horizontal asymptote of the amplitude-frequency plot, the amplification factor of this high-pass filter is assumed to be 1. The amplification factor in the static part of the step response (static amplification factor) is always zero. By contrast, the value to be taken from the horizontal asymptote of the amplitude-frequency plot is called maximum dynamic amplification factor k_d. (See also Chapter 5).

As in the case of the low-pass filter, the impulse response is represented in Figure 4.**3c** by assuming a mathematically ideal impulse as input. The response also consists of an impulse that does not return to value zero, however, but overshoots to $-1/\tau$, and then, with the time constant τ, exponentially returns to zero. The impulse response thus consists of the impulse function $\delta(t)$ and the exponential function $-(1/\tau)e^{-t/\tau}$. With an impulse function of finite duration, the duration of the impulse response and the amplitude of the negative part increase accordingly. (For details, see Varju 1977).

Whereas intuitively one could call a low-pass filter an inert element, a high-pass filter constitutes an element that merely responds to changes, but not to permanently constant values at the input. Such properties are encountered in sensory physiology, in the so-called tonic sensory cells, which respond in a low-pass manner, and in the so-called phasic sensory cells, which respond in a high-pass manner. The step response of such high-pass-like sensory cells is shown in Figure 4.**4**.

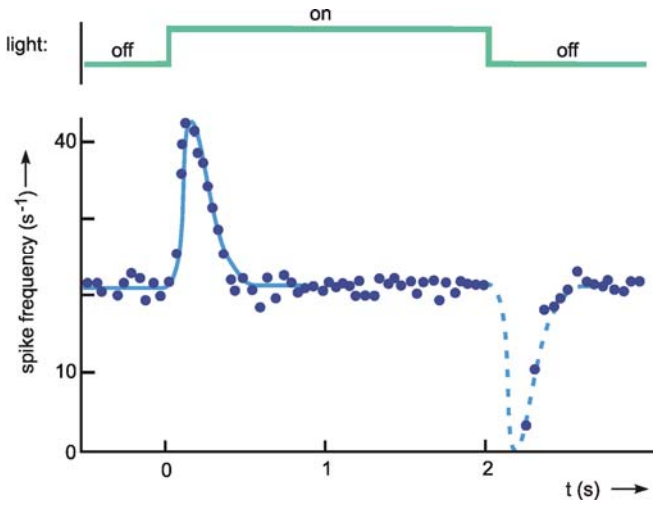


Fig. 4.**4 The response of a single fiber of the optical nerve *of* *Limulus.*** As input function light is switched of for 2 seconds (upper trace). The output function shows the instantaneous spike frequency (each dot is calculated as $1/\Delta t$, Δt being the time difference between two consecutive spikes). The solid blue line shows an approximation of the data. The dashed part (response to the decreasing step) represents the mirror image of the first part of the approximation (after Ratliff 1965). Note that negative spike frequencies are not possible

This behavior in biological systems is frequently called adaptation. In mathematical terms, one could speak of a differentiating behavior since, especially with very short time constants, the first derivative of the input function is formed approximately by a high-pass filter.

The ramp response of this high-pass filter is shown in Figure 4.**3 d.** It starts at t = 0 and a slope k, the slope of the input function, and then, with the time constant τ, increases to the plateau value $k\tau$. In order to obtain the time constant of the high-pass filter from the ramp response, the range of values

available for the input function has to be sufficiently great such that the plateau at the output can be reached while the ramp is increasing. Conversely, with the given input range A, the slope k of the ramp has to be correspondingly small. If it is assumed that the plateau is reached approximately after the triple time constant, then $k \leq A/3\tau$.

Like those for the low-pass filter, Figure 4.**3e** illustrates the polar plot of the high-pass filter and Figure 4.**3f** an electronic circuit. This circuit also consists of an ohmic resistor R and a capacitor C, but in reverse order this time. The time constant is also calculated as $\tau = R\ C$. For a digital simulation see Appendix II.

> A high-pass filter describes a system which is only sensitive to changes in the input function. It suppresses the low frequency components, which qualitatively corresponds to a mathematical differentiation. The dynamic properties are described by the time constant r or the corner frequency ω_0.

4.3 N-th Order Low-Pass Filter

If n low-pass filters are connected in series, a low-pass of the n-th order is obtained. If two systems with specific amplification factors are connected in series, we obtain the amplification factor of the total system by multiplying the two individual amplification factors. As mentioned in Chapter 3, the amplitude-frequency plot of a system reflects the amplification factors valid for the various frequencies. In order to obtain the amplitude-frequency plot of a 2nd order filter, it is thus necessary to multiply the amplitude values of the various frequencies (or, which is the same, add up their logarithms). This is why the logarithmic ordinate is used in the amplitude-frequency plot in the Bode plot: the amplitude frequency plot of two serially connected filters can be obtained simply by adding the amplitude frequency plots of the two individual filters, point by point if logarithms are used, as shown in the right hand ordinate of Figure 4.**5 a**. (See also Figure 2.**5**).

In addition to the amplitude frequency plot of a 1st order low-pass filter (already shown in Figure 4.**1**) Figure 4.**5a** shows the amplitude frequency plots of low-pass filters of the 2nd, 3rd, and 4th order, which are obtained by serially-connecting several identical 1st order low-pass filters. The corner frequency remains the same, but the absolute values of the slope of the asymptote at the descending part grows proportionately steeper with the order of the filter. This means that the higher the order of the low-pass filter, the more strongly high frequencies are suppressed. (Systems are conceivable in which the slope of the descending part of the amplitude frequency plot shows values that lie between the discrete values presented here. The system would then not be of an integer order. In view of the mathematical treatment of such systems, however, the reader is referred to the literature. (See e.g., Varju 1977).

If a sine wave oscillation undergoes a phase shift in a filter, and if several filters are connected in series, the total phase shift is obtained by linear addition of individual phase shifts. Since the value of the phase shift depends on the frequency, the phase frequency plot of several serially connected filters is obtained directly by a point-by-point addition of individual phase-frequency plots. For this reason, the phase frequency plot in the Bode plot is shown with a linear ordinate. Figure 4.**5a** shows that, at low frequencies, phase shifts are always near zero but for higher frequencies, they increase with the increasing order of the low-pass filter.

In the step responses (Fig. 4.**5b**), the initial slope decreases with an increasing order, although the maximum value remains constant (presupposing the same static amplitude factor, of course). In the

impulse responses shown in Figure 4.**5c**, both the initial slope, and the maximal amplitude of the impulse response, decrease with increasing order of the low-pass filter. Generation of the form of an impulse response or a step response of a filter of a higher (e. g., 2nd) order can be imagined as follows. The response of the first filter to the impulse (or step) function at the same time represents the input function for the second filter. This, in turn, qualitatively influences the new input function in a corresponding way. In the case of low-pass filters, output functions of the second low-pass filter will thus be much flatter.

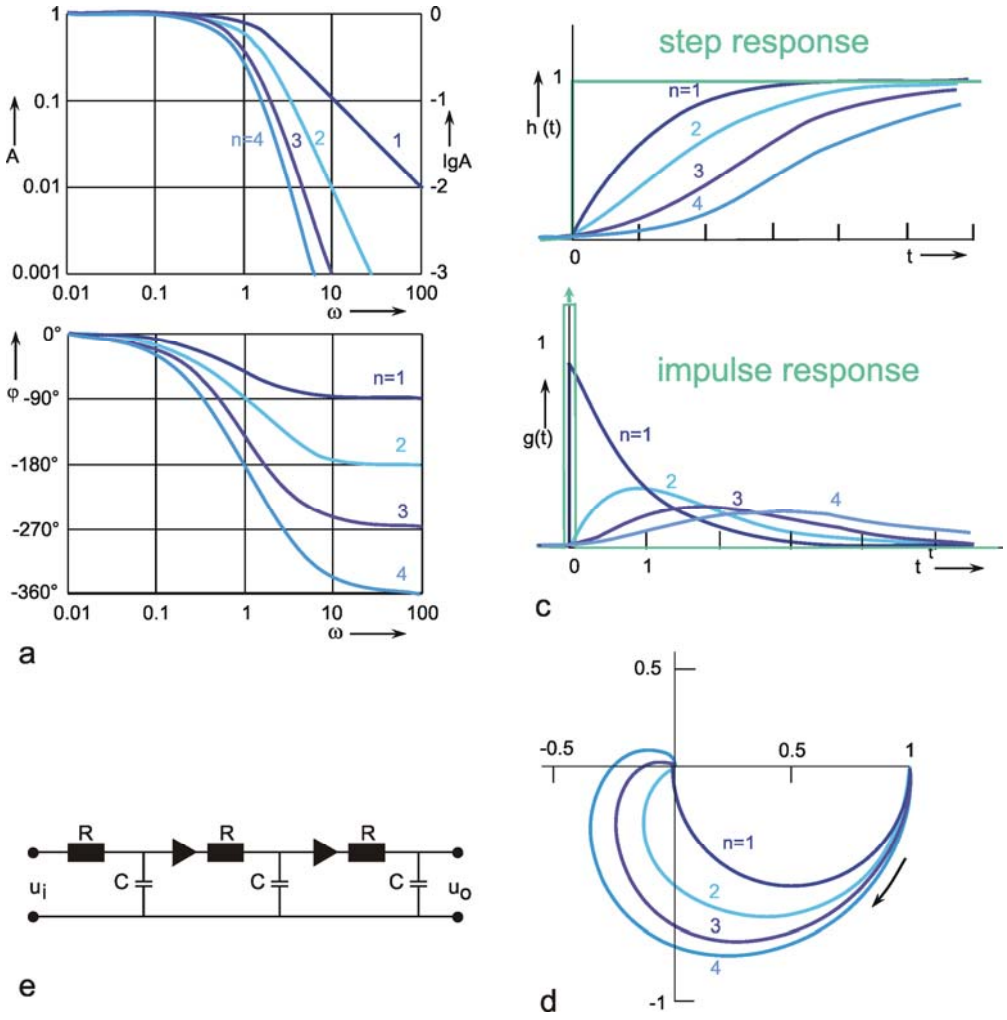


Fig. 4.**5 Low-pass filter of nth order.** (a) Bode plot, A amplitude, φ phase, ω angular frequency. (b) step response, (c) impulse response, (d) polar plot. n = 1 to 4 refers to the order of the filter. In (b) and (c) the input functions are shown by green lines. (e) Electronic wiring diagram for n = 3. The triangles represent amplifiers with gain 1, R resistor, C condensor, u_i input voltage, u_o output voltage

The formulas for impulse and step responses of low-pass filters of higher order are shown in the table in Appendix I. When a higher order filter is simulated by an electronic circuit, the individual filter circuits have to be connected via a high impedance amplifier to avoid recurrent effects. This is symbolized by a triangle in Figure 4.**5e.** Figure 4.**5d** contains the polar plots for low-pass filters of the 1st through the 4th order. The properties of low-pass filters which rely on recurrent effects are discussed in Chapter 4.9.

The Bode plot of two serially connected filters is obtained by adding the logarithmic ordinate values of the amplitude frequency plot and by linear summation of the phase frequency plot. Low-pass filters of higher order show stronger suppression and greater phase shifts of high frequency components.

4.4 N-th Order High-Pass Filters

If n high-pass filters are serially connected, the result is called a n-th order high-pass filter. The Bode plot (Fig. 4.**6a**) and polar plot (Fig. 4.**6c)** will be mirror images of the corresponding curves of low-pass filters shown in Figure 4.**5**. Here, too, the value of the corner frequency remains unchanged, whereas the slope of the asymptote approaching the descending part of the amplitude frequency plot increases proportionate to the order of the high-pass filter. For very high frequencies, the phase frequency plots asymptotically approach zero. By contrast, for low frequencies, the phase lead is the greater, the higher the order of the high-pass filter.

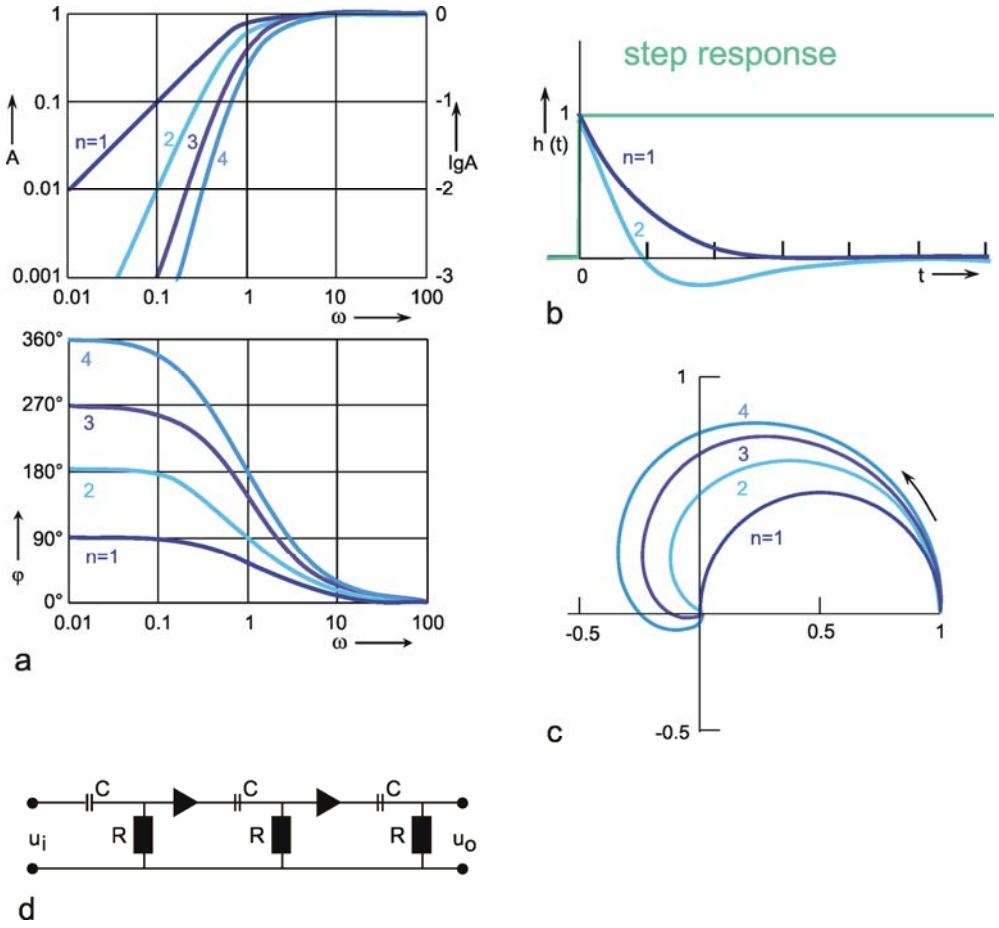


Fig. 4.**6 High-pass filter of nth order.** (a) Bode plot, A amplitude, φ phase, ω angular frequency. (b) step response, (c) polar plot. n = 1 to 4 refers to the order of the filter. In (b) the input function is shown by green lines. (b) Electronic wiring diagram for n = 3.The triangles represent amplifiers with gain 1, R resistor, C condensor, u_i input voltage, u_o output voltage

In addition to the step response of the 1st order high-pass filter (already shown in Figure 4.**3 c**) Figure 4.**6b** shows the step response of the 2nd order high-pass filter. After time T, this crosses the zero line and then asymptotically approaches the abscissa from below. The corresponding formula is given in

the table in Appendix I. The origin of this step response can again be made clear intuitively by taking the step response of the first 1st order high-pass filter as an input function of a further 1st order high-pass filter. The positive slope at the beginning of this new input function results in a positive value at the output. Since the input function subsequently exhibits a negative slope, the output function will, after a certain time (depending on the value of the time constant), take negative values, eventually reaching zero, as the input function gradually turns into a constant value.

In a similar way, it can be imagined that the step response of a 3rd order high-pass filter, which is not shown here, crosses the zero line a second time and then asymptotically approaches the abscissa from the positive side. Figure 4**.6d** shows an electronic circuit with the properties of 3rd order high-pass filter as an example.

> High-pass filters of higher order show stronger suppression and greater phase shifts of low-frequency components. Qualitatively, they can be regarded as to produce the nth derivative of the input function.

4.5 Band-Pass filter

In practice, a pure high-pass filter does not occur since no system can transmit arbitrary high frequencies. Therefore, every real system endowed with high-pass properties also has an upper corner frequency, which is given by the ever-present low-pass properties. A "frequency band," which is limited on both sides, is thus transmitted in this case. We, therefore, speak of a band-pass filter. A band-pass filter is the result of a series connection of high- and low-pass filters. In linear systems, the serial order of both filters is irrelevant. Amplitude and phase frequency plots of a band-pass filter can be obtained by logarithmic or linear addition of the corresponding values of individual filters, as described in Chapter 4.3.

The Bode plot of a symmetrical band-pass filter (i. e., where the high- and low-pass filter are of the same order) is shown in Figure 4.**7a**. As can be seen from the slope of the asymptotes approaching the ascending and the descending parts of the amplitude frequency plot, the low-pass filter and the high-pass filter are both of 1st order. The corner frequency of the high-pass filter is 1 Hz, that of the low-pass filter 100 Hz in this example. Figure 4.**7b, c** show the step response and the impulse response of this system.

The amplitude frequency plot does not necessarily indicate the time constants or corner frequencies of each filter. As described in Chapter 4.1 and 4.2, the values of the corner frequencies can be read from the intersection of the horizontal asymptote and the ascending asymptote (high-pass filter) or the descending asymptote (low-pass filter). A horizontal asymptote can only be plotted, however, if the corner frequency of the low-pass filter is sufficiently high compared to that of the high-pass filter, such that the amplitude frequency plot exhibits a sufficiently long, horizontal part. But if both corner frequencies are too similar, as in the example shown in Figure 2.**5**, or if the corner frequency of the low-pass filter is even lower than that of the high-pass filter, no statement is possible concerning the position of the horizontal asymptote on the basis of the amplitude frequency plot. (It should be stressed here that the amplification factor need not always be 1, although this has been assumed in examples used here, for the sake of clarity). In the band-pass filter shown in Figure 2.**5**, the upper and lower corner frequencies are identical in this particular case. As an illustration, the corresponding step response and the impulse response of this system are shown in Figure 4.**7d, e**.

Contrary to the Bode plot, the step and the impulse responses of two serially connected filters cannot be computed as easily from the step or impulse responses of each filter. The corresponding formulas for this case, as for the general case of different time constants, are given in Appendix I. For computation of the step and impulse responses of symmetrical or asymmetrical bandpass filters of a higher order, the reader is also referred to Appendix I. For the case of a symmetrical 1st order band-pass filter of different time constants, i. e., τ_1 (low-pass filter) ≠ τ_2 (high-pass filter), the time when the step response reaches its maximum is given by

$$t_{max} = \frac{\tau_1 \tau_2}{\tau_1 - \tau_2} \ln \frac{\tau_1}{\tau_2}.$$

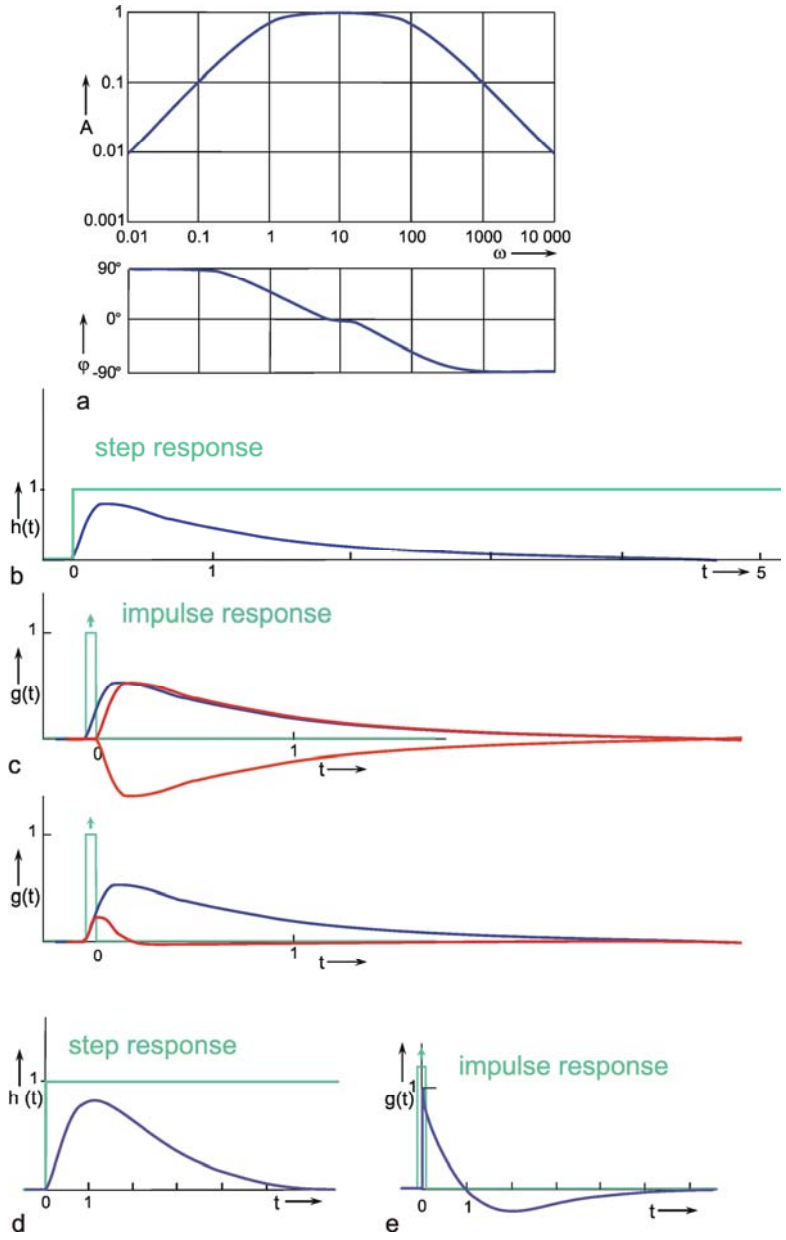


Fig. 4.**7 Band-pass filter consisting of serially connected 1st order low-pass and high-pass filters.** (a) Bode plot, A amplitude, φ phase, ω angular frequency, lower corner frequency (describing the high-pass properties): 1 Hz, upper corner frequency (describing the low-pass properties): 100 Hz. (b) Step response. (c) Impulse response. (d) and (e) Step and impulse responses of a band-pass filter whose upper and lower corner frequencies are identical (1 Hz). The Bode plot of this system is given in Fig. 2.**5**. In (b) to (e) the input functions are shown by green lines

A band-pass filter can be considered as serial connection of low-pass and high-pass filters. Its dynamics can be described by a lower and an upper corner frequency, marking a "frequency band."

4.6 Lead-Lag System

In the above section we have treated the basic types of filters and have shown how to connect them in series. Alternatively, filters might be connected in parallel. To discuss a simple example, we connect a low-pass filter and a high-pass filter with the same time constant in parallel. If the branch with the high-pass filter is endowed with an amplification k, and the outputs of both branches are added (Fig. 4.**8a),** we obtain different systems depending on the factor k. For the trivial case k = 0 the system corresponds to a simple low-pass filter.

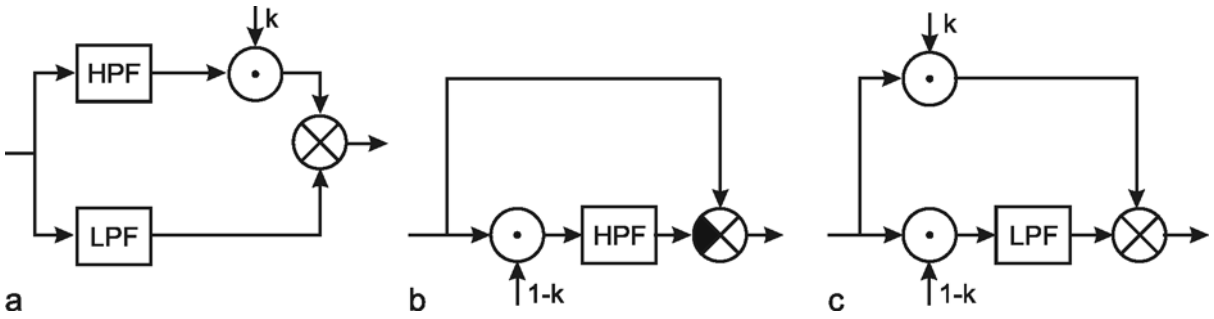


Fig. 4.**8 Three diagrams showing the same dynamic input-output behavior.** (a) A parallel connection of a high-pass filter (HPF) and a low-pass filter (LPF). The high-pass branch is endowed with a variable gain factor k. (b) The corresponding system containing only a high-pass filter, and (c) only a low-pass filter

If k takes a value between 0 and 1, we obtain a lag-lead system. As we have seen in the series connection of two filters, the Bode plot of the total system can be computed from the individual Bode plots by simple addition, while the computation of the step response is more difficult (see Chapter 4.3, 4.5). The opposite is true for parallel connection. Here, the step response of the total system can be computed by means of a point-by-point addition of the step responses of the two individual branches. Figure 4.**9 b** indicates the step responses of the two branches for the case k = 0.5 by dotted lines, and the sum of both functions, i. e., the step response of the total system, by a bold line. At the start of the input step, this function thus initially jumps to value k, and then, with the initial slope $(1 - k)/\tau$ and the time constant τ, asymptotically approaches the final value 1.

If the amplitude frequency plot of the total system is to be determined from that of each of the two branches, the correct result certainly will not be achieved by a point-by-point multiplication (logarithmic addition), as is possible in series connected filters. Nor does a linear addition produce the correct amplitude frequency plot of the total system. Since the oscillations at the output, responding to a particular sine function at the input, generally experience a different phase shift in each branch, (i. e., since their maxima are shifted relative to each other) a sine function is again obtained after addition. Because of this shift, the maximum amplitude of this function will normally be less than the sum of the maximum amplitudes of the two individual sine waves. In computation of the amplitude frequency plot of the total system, the corresponding phase shifts must, therefore, be taken into account.

For the same reasons, the phase frequency plot cannot be computed in a simple manner, either. However, a rough estimate is possible for specific ranges both of the amplitude and the phase

frequency plots (Fig. 4.**9a).** Thus, for very low frequencies, the contribution of the high-pass filter is negligible. The amplitude and phase frequency plots of the total system will therefore approximate those of the low-pass filter. This means that the horizontal asymptote of the amplitude frequency plot is about 1, and that, starting from zero, the phase frequency plot tends to take increasingly negative values.

Conversely, for high frequencies, it is the contribution of the low-pass filter that is negligible. The amplitude and phase frequency plots of the total system thus approximate those of the high-pass filter, and the amplitude frequency plot is endowed, in this range, with a horizontal asymptote at the amplification factor of k = 0.5. The phase frequency plot again approaches zero value from below.

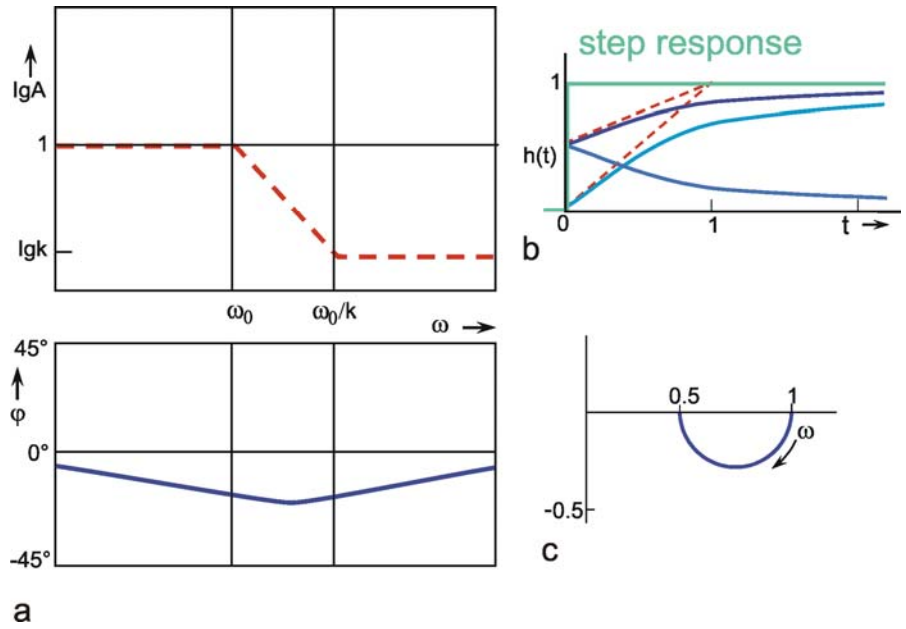


Fig. 4.**9 The lag-lead system, i. e., a parallel connection of a low-pass filter and a high-pass filter with k < 1** (see Fig. 4.**8a).** (a) The amplitude frequency plot is only shown by the three assymptotes. (b) The step responses of the two branches for the case k = 0.5 are shown by light blue lines, and the sum of both functions, i. e. the step response of the total system, by a dark blue line. The input function is shown by green lines. (c) polar plot for k = 0.5

In the middle range, the phase frequency plot is located between that of the high-pass filter and that of the low-pass filter. In Figure 4.**9a**, the amplitude frequency plot is indicated only by its asymptotes (a widely used representation in the literature). The asymptote in the middle range exhibits the slope -1 on a logarithmic scale. The intersections of this asymptote with the two others occur at the frequencies w_0 and w_0/k. Its Nyquist plot is shown in Figure 4.**9c.**

If the value of k is greater than 1, we speak of a lead-lag system. The step response is again obtained by a point-by-point addition of the low-pass response as well as of the correspondingly amplified high-pass response, shown in Figure 4.**10b** by dotted lines. As in the previous case, it immediately jumps to value k at the beginning of the step, and then, with the initial slope $(1-k)/\tau$ and the time constant τ, decreases to the final value 1.

The amplitude frequency plot of the lead-lag system is indicated in Figure 4.**10a** only by its three asymptotes, as in the case of the lag-lead system. Since the value of k is greater than 1, the horizontal asymptote in the range of high frequencies appears here above the corresponding asymptote in the range of low frequencies. The two intersections of the three asymptotes again lie at the frequencies ω_0

and ω_0/k, whereas the asymptote approaching the middle range exhibits the slope + 1 on a logarithmic scale.

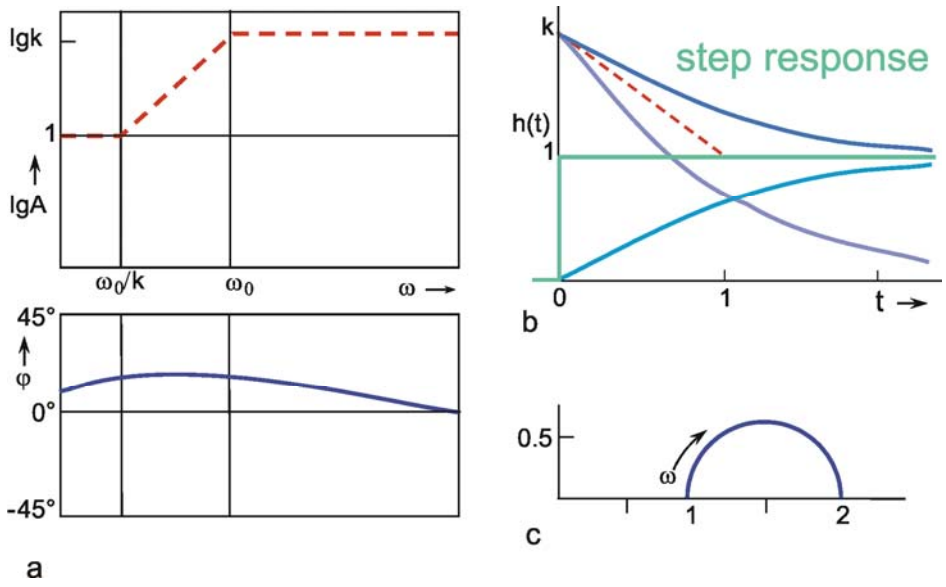


Fig. 4.**10 The lead-lag system, i. e., a parallel connection of a low-pass filter and a high-pass filter with k > 1** (see Fig. 4.**8 a).** (a) The amplitude frequency plot is only shown by the three asymptotes. (b) The step responses of the two branches for the case k = 2 is shown by light blue lines, and the sum of both functions, i. e., the step response of the total system, by a dark blue line. The input function is shown by green lines. (c) polar plot for k = *2*

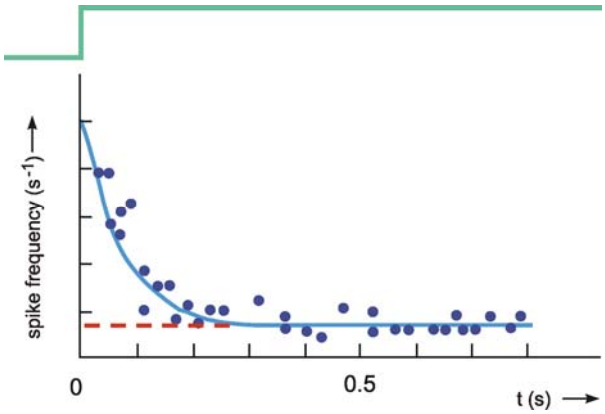


Fig. 4.**11 Step response of a mammalian muscle spindle.** When the muscle spindle is suddenly elongated (green line), the sensory fibers show a sharp increase in spike frequency which later drops to a new constant level which is, however, higher than it was before the step. Ordinate in relative units (after Milhorn 1966)

In Chapter 3, it was pointed out that an edge of a function is produced by high-frequency components. The large amplification factor in the high-frequency range that can be recognized in the amplitude frequency plot thus corresponds to the "over-representation" of the edge in the step response. On account of the larger amplification factor in the high-pass branch, the high-pass properties predominate to such an extent that the phase shift is positive for the entire frequency range. As can be expected in view of what has been discussed above, however, they asymptotically approach zero value, both for very high and very low frequencies. The corresponding Nyquist plot is shown in Figure 4.**10c**.

The behavior of a lead-lag system thus corresponds to that of the most widespread type of sensory cell, the so-called phasic-tonic sensory cell. Muscle spindles, for example, exhibit this type of step response (Fig. 4.**11**). (It should be noted, however, that there are also nonlinear elements which show similar behavior.)

If a lead-lag system is connected in series with a low-pass filter, the phasic part of the former can compensate for a considerable amount of the inert properties of the low-pass filter. This might be one reason for the abundancy of phasic-tonic sensory cells. (See also Chapter 8.6).

The system addressed here is described by the diagram shown in Figure 4.**8a,** but the same input-output relations are exhibited by the connection shown in Figure 4.**8b.** The origin of the responses of this system can be thought of as the high-pass response (which belongs to the input function and is multiplied by the factor 1-k), subtracted from the input function itself. The direct summation of the input onto the output dealt with here might be called "feed forward summation". The connection shown in Figure 4.**8c** also exhibits the same properties. Here, the input function, which is multiplied by the factor k, is added to the corresponding low-pass response, which is endowed with the factor (1-k).

To determine the transfer properties of filters connected in parallel, consideration of step responses is similar than for frequency responses. Lag-lead and lead-lag systems are simple examples of parallel connected high-pass and low-pass filters. The lead-lag system corresponds to the so-called phasic-tonic behavior of many neurons, in particular sensory cells.

4.7 Pure Time Delay

Especially in biological systems, a pure time delay is frequently observed, in which all frequencies are transmitted without a change of amplitude and with merely a certain time delay or dead time T. The form of the input function thus remains unchanged. The entire input function is shifted to the right by the amount T, however. Such pure time delays occur whenever data are transmitted with finite speed. This occurs, for example, in the transmission of signals along nerve tracts, or transport of hormones in the bloodstream.

Since the amplitudes are not influenced by a pure time delay, the amplitude frequency plot obtained is a horizontal line at the level of the amplification factor (Fig: 4.**12a).** The phase shift, on the other hand, takes increasingly negative values with increasing frequency such that $\varphi(\omega) = -\omega\ T$. The reason for this is that, compared to the oscillation period which decreases at higher frequencies, the constant dead time increases in relative terms. The value of dead time can be read from the phase frequency plot as follows. It corresponds to the oscillation period $\nu = \omega/2\pi$, in which a phase shift of exactly -2π or $-360°$ is present. The phase frequency plot of a pure time delay thus, unlike that of the other elements discussed so far, does not approximate a finite maximum value. Rather, the phase shifts increase infinitely for increasing frequencies. The corresponding Nyquist plot is shown in Figure 4.**12c**. (Confusion may be caused by the fact that, in some textbooks, the oscillation period $T = 1/\nu$, the dead time (in this text), as well as the time constant, may be referred to by the letter T.)

An electronic system endowed with the properties of a pure dead time requires quite a complicated circuit. However, it can be simulated very simply using a digital computer. (See Appendix II).

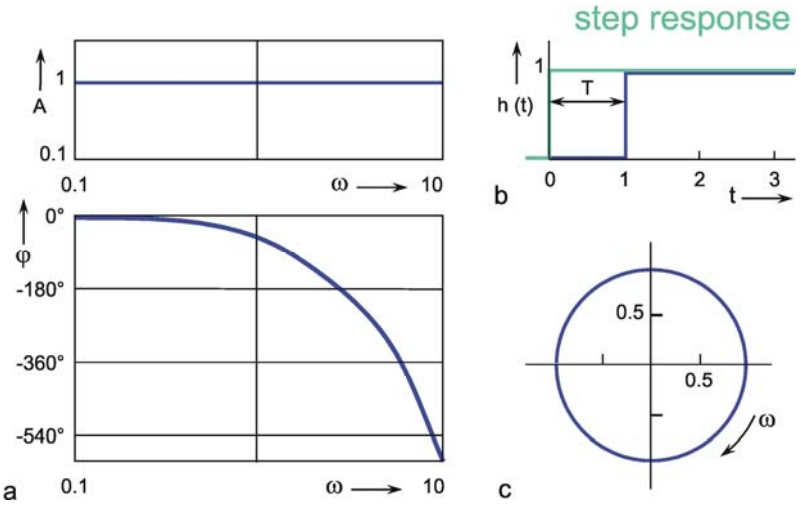


Fig. 4.**12** **Pure time delay**. (a) Bode plot, A amplitude, φ phase, ω = 2πν angular frequency. (b) step response, the input function is shown by green lines, T dead time. (c) polar plot

A pure time delay or dead time unit does not influence the amplitudes of the frequency components but changes the phase values the more the higher the frequency.

Box 1

An Asymmetrical Band-Pass Filter

In the following experiment a visual reflex of an insect is investigated. A locust is held by the thorax and a pattern, consisting of vertical black and white stripes, is moved sinusoidally in horizontal direction in front of it (Thorson 1966). According to the well-known optomotor response, the animal tries to follow the movement of the pattern by moving its head correspondingly. However, the head is fixed to a force transducer and the forces are measured by which it tries to move its head using its neck muscles. The amplitude and the phase shift of the sinusoidally varying forces are measured in dB (Fig. B1.**1**). How could this result be interpreted in terms of filter theory? Regarding the amplitude frequency plot, the slope of the asymptote for high frequency values (i. e., > 0.5 Hz) is approximately -1, indicating a 1st order low-pass filter. Note that 20 d8 correspond to one decade (Chapter 2.3). The asymptote to the low frequency values is approximately 0.5 and thus has to be interpreted as a high-pass filter of 0.5th order. Thus, as a first interpretation we could describe the system as a serial connection of a 1st order low-pass filter and a 0.5th order high-pass filter. Does the phase frequency plot correspond to this assumption? A high-pass filter of 0.5th order should produce a maximum phase shift of 45° for low frequencies, as was actually found. For high frequencies, a 1 st order low-pass filter should lead to a maximum phase shift of -90°. However, a somewhat larger phase shift is ob served. Thus, the results could only be described approximately by a band-pass filter which consists of a 1st order low-pass filter and a 0.5th order high-pass filter. The larger phase shifts could be caused by a pure time delay. The step response revealed a dead time of 40 ms. For the frequency of 1 Hz this leads to an additional phase shift of about 15° and for 2 Hz to a phase shift of about 30°. The phase shifts produced by such a pure time delay for the three highest frequencies used in the experiment are shown by open circles. These shifts correspond approximately to the differences between the experimental results and the values expected for the band-pass filter. Thus, the Bode plot of the system can reasonably be described by an asymmetrical band-pass filter and a pure time delay connected in series.

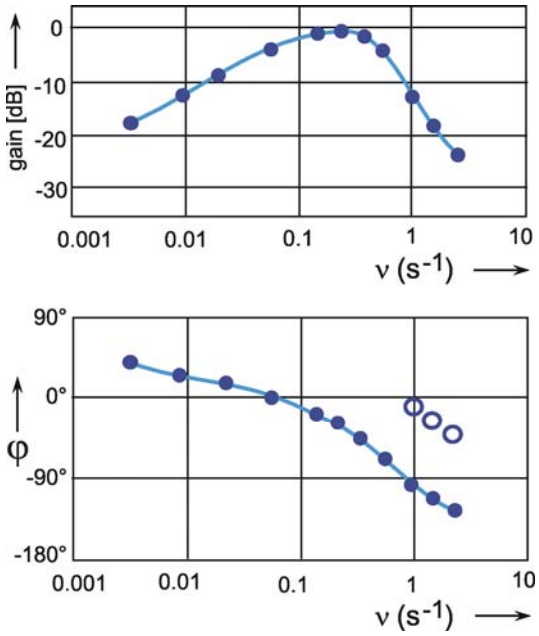


Fig. B 1.**1 Bode plot of the forces developed by the neck muscles of a locust.** Input is the position of a sinusoidally moved visual pattern. ν is stimulus frequency. The gain is measured in dB (after Thorson 1966)

4.8 Integrator and Differentiator

Whenever integration is carried out in a system (e. g., if, in a mechanical system, force (or acceleration) is to be translated into speed, or the latter is to be translated into position) such a process is described mathematically by an integrator. Within the nervous system, too, integration is performed. For example in the case of the semicircular canals, integrators have to ensure that the angular acceleration actually recorded by these sense organs, is transformed somewhere in the central nervous system into the angular velocity perceived by the subject.

The step response of an integrator is a line ascending along a slope τ_i from the beginning of the step (Fig. 4.**13b).** τ_i is called the integration time constant. It denotes the time that elapses before the response to a unit step reaches value 1. The "amplification factor" of an integrator is thus inversely proportional to τ_i ($k = 1/\tau_i$) and hence takes the dimension s^{-1}. In accordance with the properties of an integrator, the impulse response jumps to value 1 and remains there (Fig. 4.**13c**). The amplitude frequency plot of an integrator consists of a line descending along a slope -1 in the logarithmic plot. This means that the lower the input frequency, the more the amplitudes are increased, and that high frequency oscillations are considerably damped. The amplitude frequency plot crosses the line for the amplification value 1 at $\omega = 1/\tau_i$. For all frequencies, the phase shift is a constant -90°. Figure 4.**13d** shows the symbol normally used for an integrator.

An integrator might also be described by a system, the output of which does not solely depend on the actual input but also on its earlier state. This requires the capability of storing the earlier value. Therefore an integrator can also be represented by a unit with recurrent summation (Fig. 4.**13e).** This permits an easy way of implementing an integrator in a digital computer program (see Appendix II).

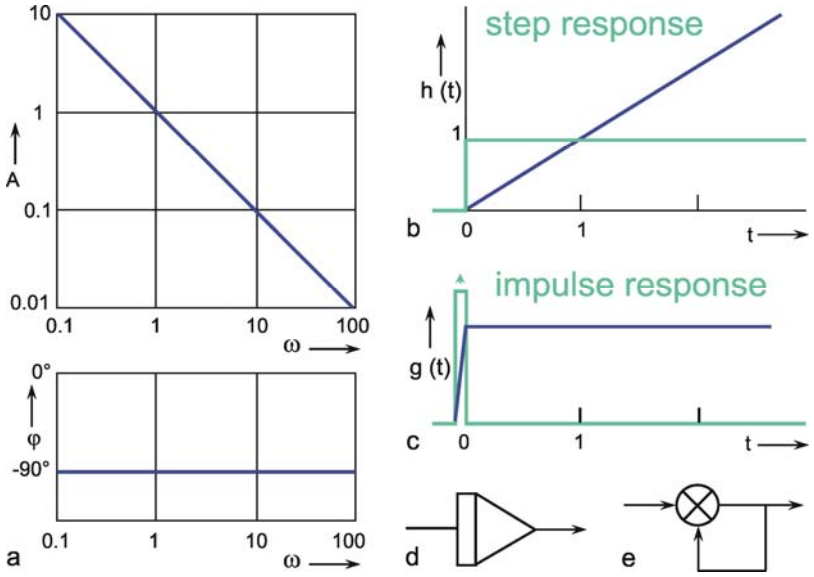


Fig. 4.**13 Integrator.** (a) Bode plot, A amplitude, φ phase, ω angular frequency. (b) step response. (c) Impulse response. In (b) and (c) the input functions are shown by green lines. (d) symbolic representation of an integrator. (e) An integrator can also be realized by a recurrent summation

A mathematically exact integrator can be interpreted as a limiting case of a low-pass filter with an infinitely small corner frequency and an infinitely high amplification. A real integrator could thus be approximated by a low-pass filter with large time constant and large amplification.

In Chapter 4.6 we considered the parallel connection of a low-pass filter and a high-pass filter. Occasionally, in biology we find systems having a step response which seems to consist of two (or more) superimposed exponential functions, i. e., it can be described by a parallel connection of two high-pass filters with different time constants. In a semi-logarithmic plot, this step response can be approximated by two lines with different slopes, corresponding to the two time constants. If the superimposition of more than two exponential functions is necessary for the approximation of the step response, a better interpretation might be obtained by using a power function y = const. t^{-n}. The power function is characterized by the fact that, plotted on a double logarithmic scale, it will show as a line with the slope -n. Mathematically, at the beginning of the step, for $t_0 = 0$ the power function takes an infinite amplitude which, of course, does not apply to realistic systems. If we use the maximum amplitude of the step response A_1 at time t_1 the power function can be written as $y(t) = A_1 (t/t_1)^{-n}$. The amplitude frequency plot of such a system is given by $A(\omega) = A_1\Gamma(1 - n) (t_1 \omega)^n$, whereby Γ stands for the so-called Gamma function (see e. g., Abramowitz and Stegun 1965). Thus, the amplitude frequency plot is given by a line with slope n. The phase frequency plot is given by $\varphi(\omega) = n\ 90°$, i. e., is constant for all frequencies; this holds for $0 < n < 1$. For the limiting case n = 1, this system could be interpreted as an ideal differentiator, i. e., as a system which forms the 1st derivative of the input function. The response to a ramp input (x(t) = k t) is given by $y(t) = A_1 k\ n/[t_1^{-n}\ \Gamma(2 - n)]\ t^{1-n}$.

Figure 4.**14** shows the step response of such an "approximate differentiator", namely the mechanoreceptive spine of a cockroach, in a plot with linear coordinates (Fig. 4.**14a**), and in a double logarithmic plot (Fig. 4.**14b**). In this case, A_1 and t_1 correspond to about 450 Hz and 0.02 s, respectively.

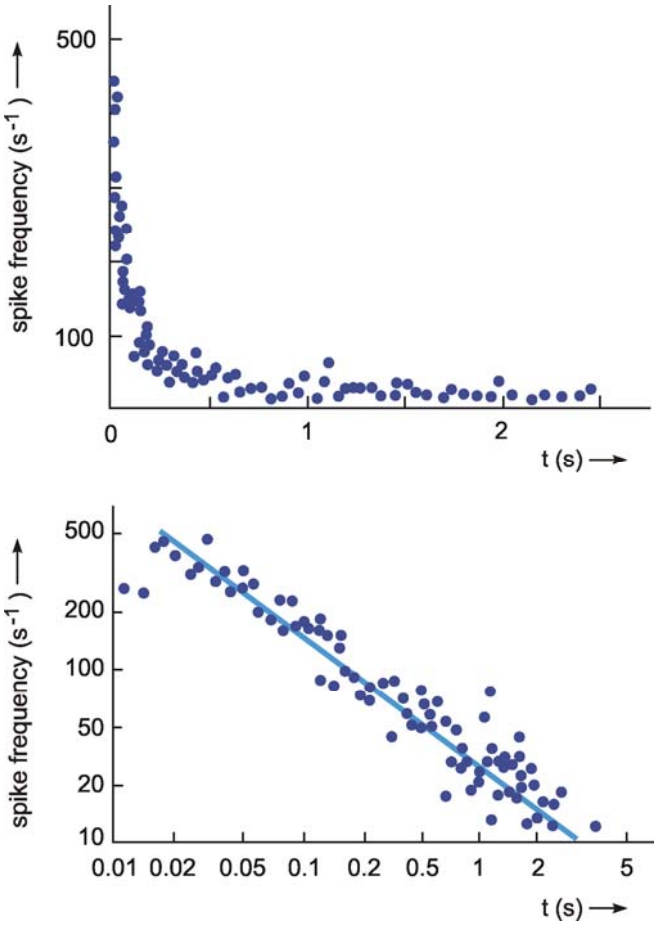


Fig. 4.**14 The step response of a cockroach mechanoreceptor,** a sensory spine. This response (a), can be approximated by a power function, $y(t) = A_1 (t/t_1)^{-n}$, which shows a straight line in a double logarithmic plot (b) (after Chapman and Smith 1963)

An integrator corresponds qualitatively to a low-pass filter with very low corner frequency and very high gain. A system showing a step response in a form of a power function can be interpreted as an approximate differentiator.

4.9 Oscillating Systems

Systems of the 1st order are characterized by the fact that they are endowed with only one energy store (capacitor, spring etc.). If a system has several different energy storage units (e. g., spring and inert mass, see Figure 1.**2**, or capacitor and inductor in an electrical system), the energy could, under some conditions, be transferred rhythmically from one store to another within the system. This makes the system oscillate. Imagine, for example, giving a short kick to the handle of the mechanical system shown in Figure 1.**2**. The number of the order of the system corresponds to its number of storage units. (This is also the case for the examples in Chapter 4.3 and 4.4.)

We will now consider an oscillating system of the 2nd order endowed with low-pass properties (Fig. 4.**15**). The transfer properties of this system are best described by the step response (Fig. 4.**15b**). Since the system's amplitude and the duration of oscillations are dependent on an additional parameter, the damping of the system, the step responses are given for different damping values ζ. (For how to determine ζ, see below). In the theoretical case of $\zeta = 0$ (i. e., if the system has no

damping) an unlimited continuous oscillation is noted in the frequency $\omega_s = \omega_0$ with a mean amplitude value of 1. ω_0 is known as the eigenfrequency or resonance frequency.

If $0 < \zeta < 1$, a "underdamped" oscillation of the frequency $\omega_s \ (= \omega_0 \sqrt{(1-\zeta^2)} \ )$ is obtained, which decreases to the final amplitude value 1.This would usually be expected to occur in the mechanical system of Figure 1.**2**.

The envelope of this damped oscillation is an exponential function with the time constant $1/(\zeta \ \omega_0)$. Thus, the smaller the damping, the longer will be the transient. For $\zeta = 1$ (the so-called critical damping), the step response approximates the final value 1 without overshoot. For higher damping values ($\zeta > 1$, the "overdamped" case) the curve is even more flattened (i. e., the final value is also obtained without overshoot) but this takes longer than in the case of $\zeta = 1$.This might, for example, happen when the pendulum of the mechanical system of Figure 1.**2** is submerged in an oil tank. The exact formulas for these step responses and the corresponding weighting functions are given in the table in Appendix I.

Although for this 2nd order filter, too, a time constant can be given, it cannot be read as easily as that of the step response of a 1st order filter. A practical measure used for the form of the increase of the step response is half-life (HL). This is the time needed by a response to increase to half of its final static value. For a 1st order filter, half-life can be computed simply from the time constant:

$$HL = \tau \ln 2 = 0.7 \tau \ .$$

For the 2nd order oscillatory system discussed here, the approximation formula is:

$$HL = \frac{1}{\omega_0}(1.047+0.37\zeta+0.2\zeta^2+0.067\zeta^3) \qquad \text{for } 0<\zeta<1.$$

A further useful approximation formula describes the strength of overshoot of the step response. If p denotes the difference between the amplitude value of the first maximum and the static final value,

$$p = e^{-\pi\zeta/\sqrt{1-\zeta^2}} \ .$$

Figure 4.**15 a** shows the course of the Bode plot for different damping values ζ. Apart from the formation of a maximum at the frequency ω_s, the amplitude frequency plot can be described, as in the case of the non-oscillating 2nd order low-pass filter (Fig. 4.**5**), by two asymptotes, one being horizontal and a second one exhibiting a slope of -2 on a logarithmic scale. Both asymptotes intersect each other at the eigenfrequency ω_0. For $\zeta = 1$, both systems become identical.

The occurrence of a maximum means that the frequencies near ω_s are particularly amplified by this system. The over-proportional amplification of this frequency also determines the form of the step response. If damping $\zeta = 0$, the infinitely high maximum occurs at frequency $\omega_s = \omega_0$. With increased damping, the amplitude of the maximum decreases, and its location shifts to lower frequencies ($\omega_s < \omega_0$). If the $\zeta \geq 1$, a maximum no longer exists. The amplification factor at the eigenfrequency becomes increasingly smaller, while the location of the two asymptotes remains unchanged. The general formula of the amplitude frequency plot is as follows:

$$A(\omega) = \frac{\omega_0\sqrt{1-\zeta^2}}{\sqrt{(\omega_0{}^2 - \omega^2)^2 + 4\zeta^2\omega_0{}^2\omega^2}}.$$

The phase frequency plot is also related to that of a nonoscillating 2nd order low-pass filter (see Fig. 4.**5**). The maximal phase shift amounts to -180°; at the eigenfrequency, ω_0, it takes the value -90°. The greater the damping, the smaller becomes the absolute value of the slope of the phase frequency plot.

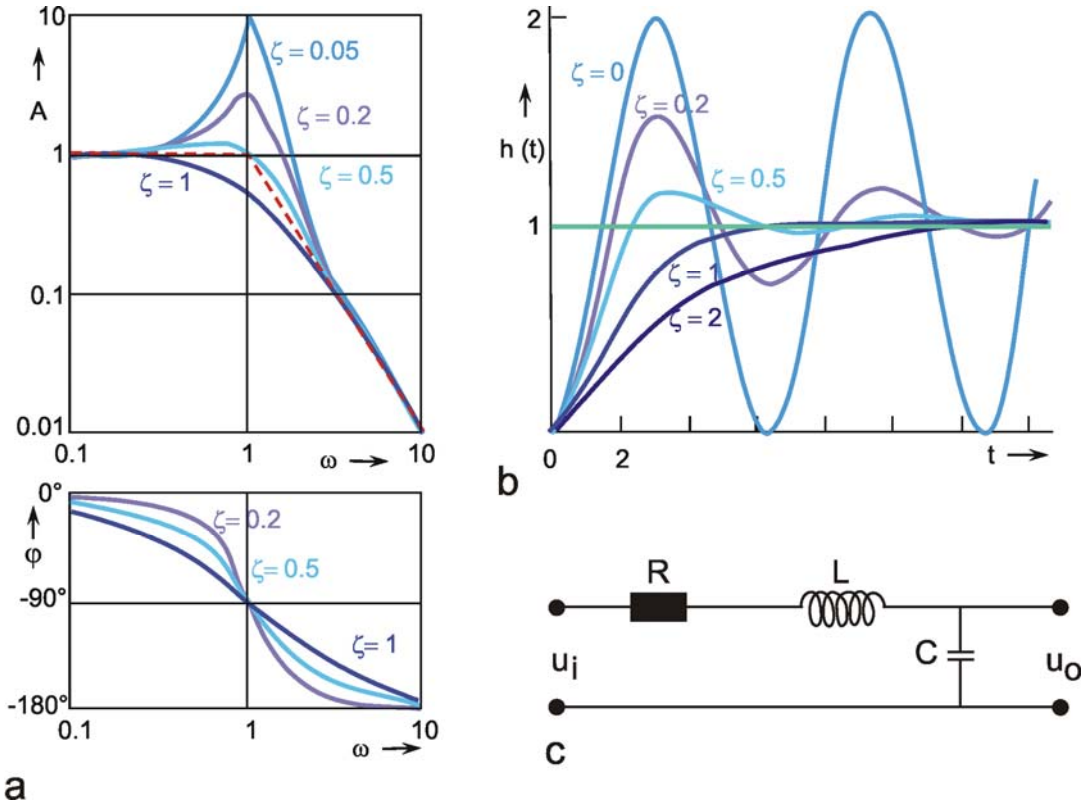


Fig. 4.**15 A 2nd order oscillatory low-pass filter.** (a) Bode plot, A amplitude, φ phase, $\omega = 2\pi\nu$ angular frequency, ζ, describes the damping factor, where ζ = 0 corresponds to no damping. (b) step response, the input function is shown by green lines.(c) Electrical circuit, R resistor, C capacitor, L inductor, u_i input voltage, u_o output voltage

Figure 4.**15c** represents an electronic circuit with the same properties as this system using an ohmic resistor R, an inductor L, and a capacitor C. L and C represent the two energy storages. For this system the following formulas hold:

$$\omega_S = \sqrt{1/LC - R^2/4L^2},$$

$$\omega_S = 1/\sqrt{LC}, \quad \zeta = \sqrt{R^2C/4L}\,.$$

A mechanical example of an oscillating system of the 2nd order is shown in Figure 1.**2**. If the parameters of a given 2nd order system are to be obtained quantitatively, the best way is to start by taking the eigenfrequency ω_0 from the Bode plot. Value ζ can be obtained by constructing an electronic system of eigenfrequency ω_0 and then changing the ohmic resistor until the step response or the Bode plot of the electronic system agree with that of the system under examination. ζ can then be computed from the values of R, C, and L. The simulation of such a filter is explained in Appendix II.

If, in the circuit diagram of Figure 4.**15c**, the ohmic resistor R and the capacitor C are interchanged, an oscillatory 2nd order system endowed with high-pass properties is obtained. This system will not be discussed further, however.

A system with two different energy storages may oscillate. In terms of low-pass filter properties this corresponds to that of a 2nd order low-pass filter that can show an output function with superimposed oscillation depending on a damping factor.

4.10 Combination of Different Filters

In the previous sections, a series of different simple systems was addressed from which arbitrarily complex systems could be constructed by combination. In the study of biological systems, however, the reverse problem is normally of interest. Assume that we have obtained the Bode plot of the system under view. The next task would be to describe the measured Bode plot by an appropriate combination of described filters. Once this has been done successfully, the question arises as to the extent to which this model agrees with the biological system in terms of the arrangement of individual elements. That is, to what extent statements can be made about its internal construction in this way.

A general impression of the limitations of this method will be illustrated with the help of very simple examples with known internal structure. Figure 4.**16** shows four combinations, each of a high-pass filter and a low-pass filter. Could these four systems be distinguished in terms of their input-output behavior? A comparison of the static part of the step responses of systems c and d shows that in case d static output value will be zero (since the high-pass filter takes the output value zero) while for case c, the output value zero of the high-pass filter is added to the output value of low-pass filter which differs from zero. Both systems can thus be distinguished from each other.

If we look at the static range, it is easy, for this simple case, to predict the step responses for systems a and b. In system a, the high-pass filter produces the static output value zero so that, after some time, the low-pass filter, too, takes the output value zero. In system b, the high-pass filter likewise adapts to zero, after the low-pass filter has reached a static output value (generally different to zero). Systems a, b, and d thus take value zero in the static part of the step response. In this way they can be distinguished from system c, but not from each other.

Systems a and b are linear, whereas system d is endowed with nonlinear properties due to built-in multiplication. If, for this reason, we use step functions of varying amplitudes, the step responses of systems a and b can be calculated by simple multiplication with an appropriate factor that corresponds to the ratio of the related input amplitudes. The responses of system d do not reflect this proportionality (Fig. 4.**16e).** Thus, while system d exhibits special properties, systems a and b, forming a band-pass filter, cannot be distinguished on the basis of their input-output behavior. Principally, this is not possible for linear systems as the sequence of series connection is irrelevant.

These examples show that, in the study of an unknown system, it is possible to rule out certain combinations (that is, certain hypotheses about the internal structure of a system), but that, on the other hand, a number of different combinations may exist, endowed with the same input-output behavior, which can not, therefore, be distinguished along these lines. A further example of a differently constructed system with the same input-output behavior has already been mentioned (Fig. 4.**8**).

Especially in the case of linear systems, a decision about the kinds of filters actually involved and about their arrangement can be taken only if a dissection of the total system is possible, and parts of the system can be examined in this way. (On this, see Hassenstein and Reichardt 1953). A decision is easier, however, if linear and nonlinear elements occur in a system. This is usually the case in biological systems. For this reason, further examples relating to the analysis of a system will only be dealt with after the discussion of nonlinear elements. Linear systems theory is not superfluous, thought

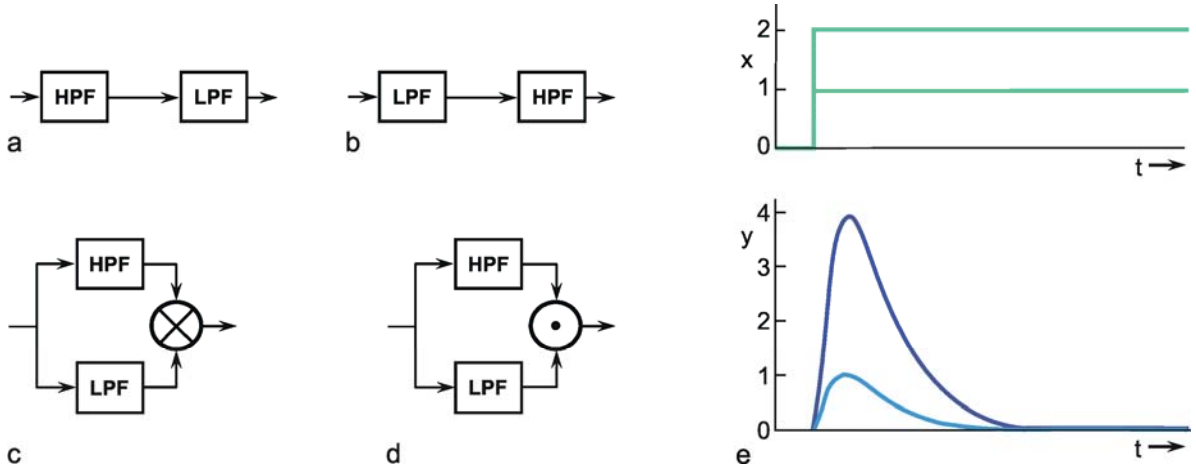


Fig. 4.**16 Four combinations of connecting a low-pass filter and a high-pass filter.** (a, b) Serial connection, (c, d) parallel connection with summation (c) and multiplication (d). The step responses of (a) and (b) are given in Fig. 4.**7**, that of system (c), for the case $\tau_{LPF} = \tau_{HPF}$ and different gain factors for the high-pass branch in Figs. 4.**9b** and 4.**10b**. (e) Two responses to steps of two different amplitudes (green lines) of the system shown in (d) illustrating the nonlinear property of this system (doubling the input amplitude does not lead to twice the output amplitude)

as might be concluded from the foregoing remarks. Rather, it is the prerequisite for understanding nonlinear systems.

The input-output behavior of a system does not, in many cases, permit a unique conclusion with respect to the internal structure. This is particularly so for linear systems.

5 The Characteristics

5.1 General Remarks

This chapter will address the essential properties of nonlinear systems. This requires the introduction of the term characteristic. The characteristic of a system is obtained by measuring the output values y belonging to the corresponding input values x, and then describing the relations between x and y in graphical form. The characteristic is not, therefore, a time function. As an example, the characteristic of a stretch receptor in the abdomen of the crayfish is shown in Figure 5.**1**. The input value is given by the expansion of the stretch receptor, which is measured in mm. The receptor potential, which is measured in mV, is used as output value.

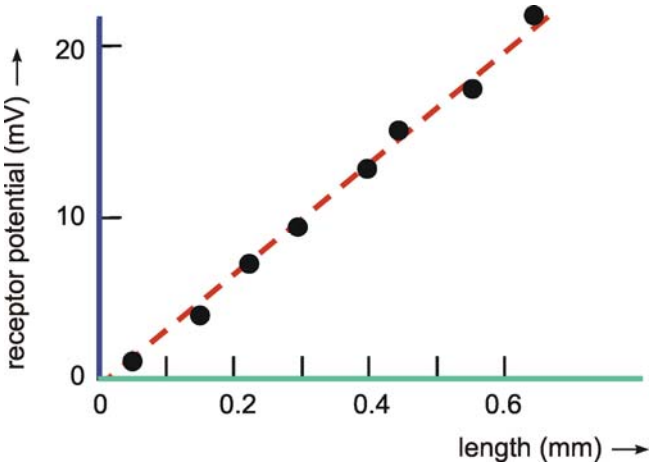


Fig. 5.**1 The characteristic describing the dependency of the receptor potential of the crayfish stretch receptor (ordinate: output) on the length of this sense organ (abscissa: input)** (after Schmidt 1972)

5.2 The Static Characteristic

The static characteristic constitutes the simplest case. A static characteristic can be used as a good representation of the properties of a system in those cases in which the output succeeds the input without measurable time delay (i. e., in which, for example, no measurable transients occur in the step response. Frequently, however, the output function requires a certain time until it has caught up with the input value. This applies to all systems that are endowed with an energy store (e. g., spring, inert mass, capacitor); that is, to all systems of the 1st and higher orders. (Systems without energy storage are therefore also known as systems of the zeroth order). In such cases dynamic characteristics may be obtained; they are discussed in Chapter 5.3.

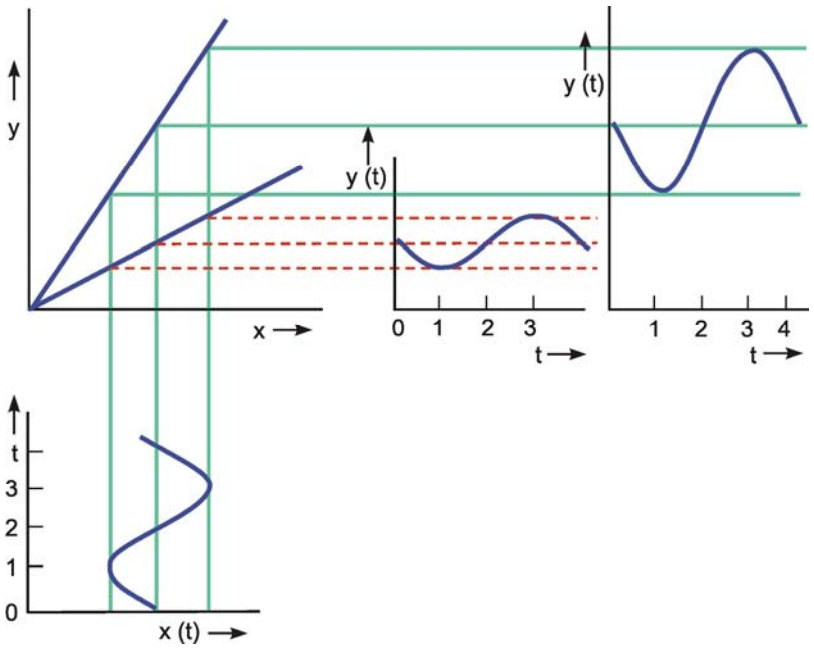


Fig. 5.**2 The construction of the output function y(t) for a given input function x(t), shown for two different linear characteristics**

If the static characteristic of a system of the zeroth order is known, the response of this system to an arbitrary input function can be obtained graphically by a "reflection along the characteristics." This is shown in Figure 5.**2** for two linear characteristics with different slopes. The input function x(t) is plotted relative to the characteristic such that the ordinate x(t) of the input function lies parallel to the abscissa x of the characteristic. The coordinate system for the output function y(t) is arranged such that its ordinate y(t) lies parallel to the ordinate y of the characteristic. The amplitude value of the output function can then be obtained for every moment of time by reflection of the amplitude value of the input function along the characteristic. The corresponding abscissa value t of the input function is again marked off the abscissa of the output function. (For the case of a curved, i. e., nonlinear, characteristic, this is shown for three selected points in Figure 5.**3**). For systems for the 1st and higher orders, this construction is possible only if transients are of such short duration that they are negligible in terms of the desired precision, or if only the static behavior of the system is of interest.

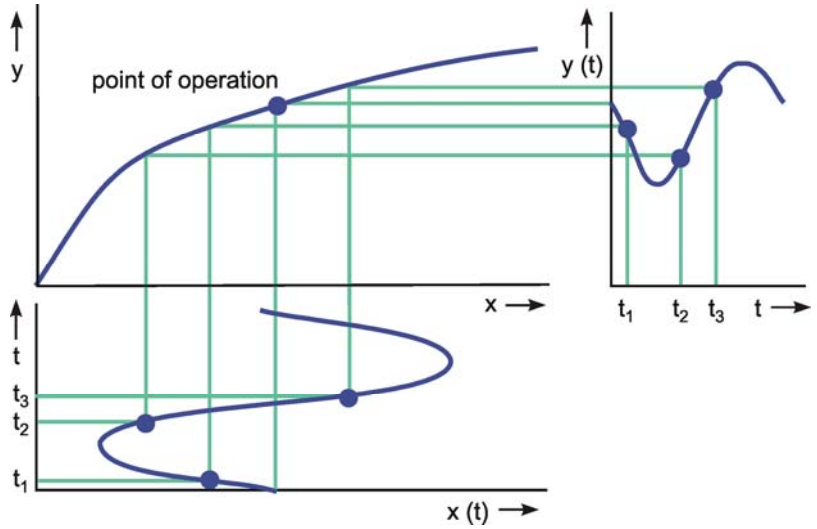


Fig. 5.**3 The construction of the output function y(t) for a given input function x(t), shown for a nonlinear (in this case logarithmic) characteristic.** As example, three time values t_1, t_2, and t_3 are marked. The point of operation is defined as the mean of the input function (after Varju 1977)

The comparison of the two characteristics presented in Figure 5.**2** shows that the steeper the slope of the characteristic, the greater is the amplification of the system. The slope of the characteristic thus reflects the amplification factor of the system. If the line does not cross the origin, or if the characteristic is nonlinear, the amplification A = y/x is not identical with the slope of the characteristic.

Rather, the relation of input and output amplitude (i. e., the amplification factor of the system) changes with the size of the input value. For the characteristic shown in Figure 5.**3**, for example, the amplification for small input amplitudes is large, whereas it continues to decrease with increasing input amplitudes. To enable us to describe the range of the characteristic in which the values of the input function are located *("range of operation*"), the *point of operation* is defined as that point of the characteristic which corresponds to the mean of the input function (Fig. 5.**3**). In addition, the term modulation is used here to characterize the transfer properties of a system. The modulation of a function means the ratio between amplitude and mean value of the function.

The great majority of characteristics occurring in biology are nonlinear; that is, they cannot be described by simple proportionality. In sensory physiology, for example, only a few receptors are known which are linear within a large range. One such case is shown in Figure 5.**1**. Many nonlinear characteristics can be approximated by a logarithmic or an exponential characteristic, at least within a given range.

The logarithmic characteristic y = const ln (x/x_0) constitutes the mathematical form of the so-called Weber-Fechner Law. This states that in a number of sense organs the response y is proportional to the logarithm of the stimulus intensity x. If $x \leq x_0$, the response takes value zero. X_0 is therefore known as threshold intensity. The simplest way to check whether a characteristic actually takes a logarithmic form, is to transfer it to a semilogarithmic scale, which should produce a straight line. This is shown in Figure 5.**4** by using as an example a sensory cell of a turtle's eye. A logarithmic relation between light intensity I and receptor potential A can only be found for a limited range of about two orders of magnitude. Above and below this range, there are deviations from the logarithmic course (i. e., in this representation, from the line).

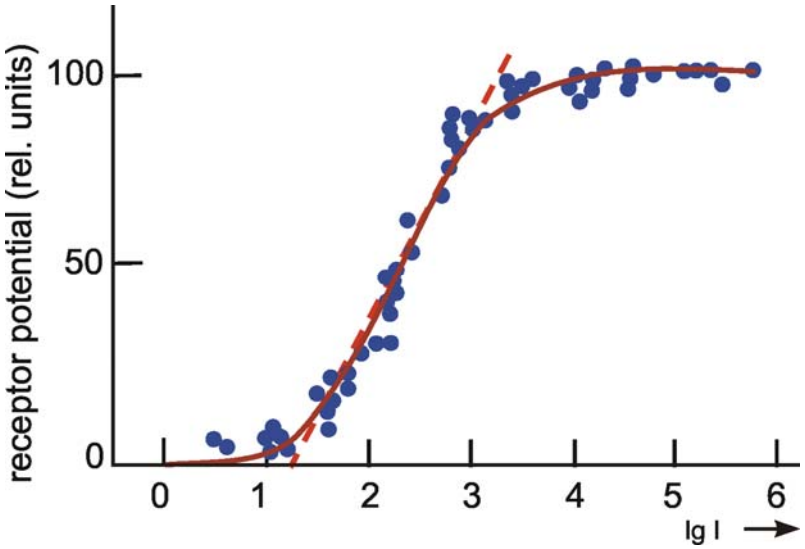


Fig. 5.**4 The dependency of the receptor potential of a retinal sensory cell of a turtle (ordinate) on the intensity of the light stimulus (abscissa, logarithmic scale).** The central part can be approximated by a straight line (dashed), indicating the logarithmic part of the characteristic (after Schmidt 1973)

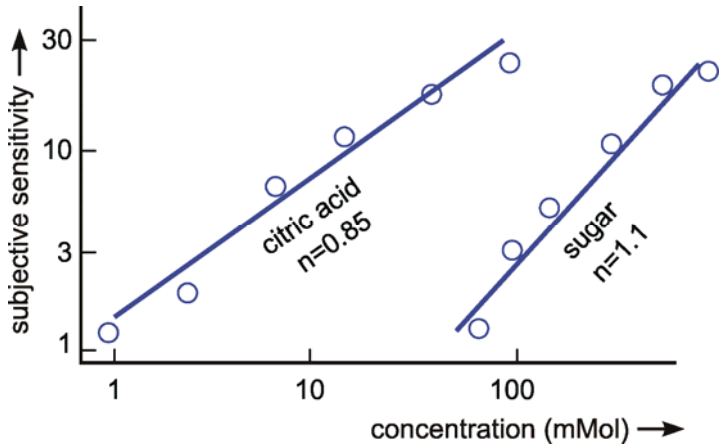


Fig. 5.**5 The subjective sensitivity of tasting the concentration of citric acid and sugar of human subjects (double logarithmic scales).** n gives the slopes of the lines which correspond to the exponents of the underlying power functions (after Schmidt 1973)

The power characteristic $y = \text{const } (x - x_0)^n$ is the mathematical form of the Power Law or Stevens' Law. x_0, in turn, denotes the threshold value, i. e., the value that has to be exceeded at the input in order to get a measurable response at the output. For the power characteristics measured in biology, the exponents occur between $0.3 < n < 3.5$. A power characteristic is present if this characteristic results in a line after double - logarithmic plotting. The slope of this line takes the value n. Two examples are shown in Figure 5.**5**. These describe the intensity of subjective sensations of taste in sampling various concentrations of citric acid and sugar solutions. Figure 5.**6** is a schematic illustration of the course of the power function on a linear scale with three different exponents. A linear characteristic is obtained if n = 1.

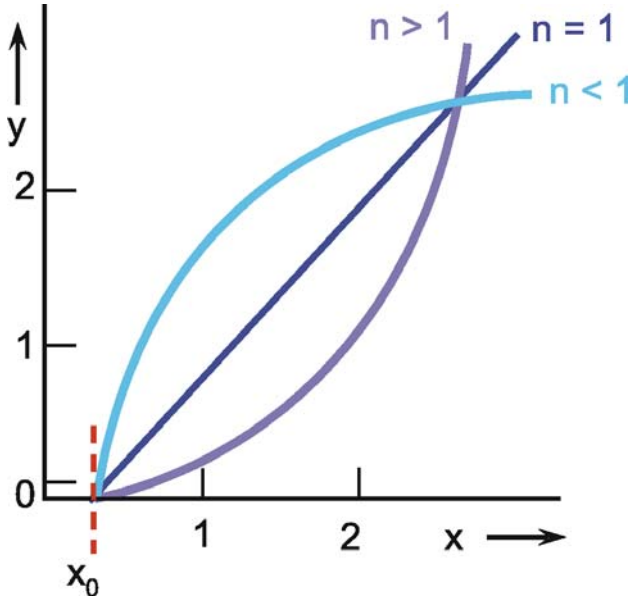


Fig. 5.**6 Three power functions plotted on linear scales.** x_0 describes a threshold value, n is the exponent of the power function. x input, y output

The example of the logarithmic characteristic, as shown in Figure 5.**7**, illustrates the most important transfer properties of nonlinear static characteristics. Figure 5.**7a** shows the sine responses obtained by "reflection" at the characteristic as already shown in Figure 5.**3**, but now with four different input amplitudes, and a fixed point of operation. The first property we notice is that the sine function is highly distorted. This shows that the output function is endowed with a number of harmonics in the sense of Fourier analysis, which do not occur in the input function, the latter being a pure sine function. This is also known as a change of the degree of harmonic distortion. This distortion is accompanied by a shift in the mean. Since the proportion of the area below the line, which runs at the level of the point of operation, becomes larger with increasing amplitudes, the mean of the output function decreases. Thus, as a function of the size of input amplitude, a shift in the mean ("dc-shift") is observed. On account of this dc-shift and the distortion of the amplitude, the degree of modulation also changes as a result of the transfer by nonlinear characteristics.

As shown in Figure 5.**7b**, these quantities change even if the amplitude of the input function, though not the point of operation, is kept constant. With respect to systems to be discussed later, it should be noted that with these nonlinear static characteristics, the response to a symmetrical input function (symmetrical in terms of a vertical axis through a minimum or maximum) is again symmetrical. This is not true for certain combinations of other systems, for example nonlinear static characteristics and linear filters addressed below.

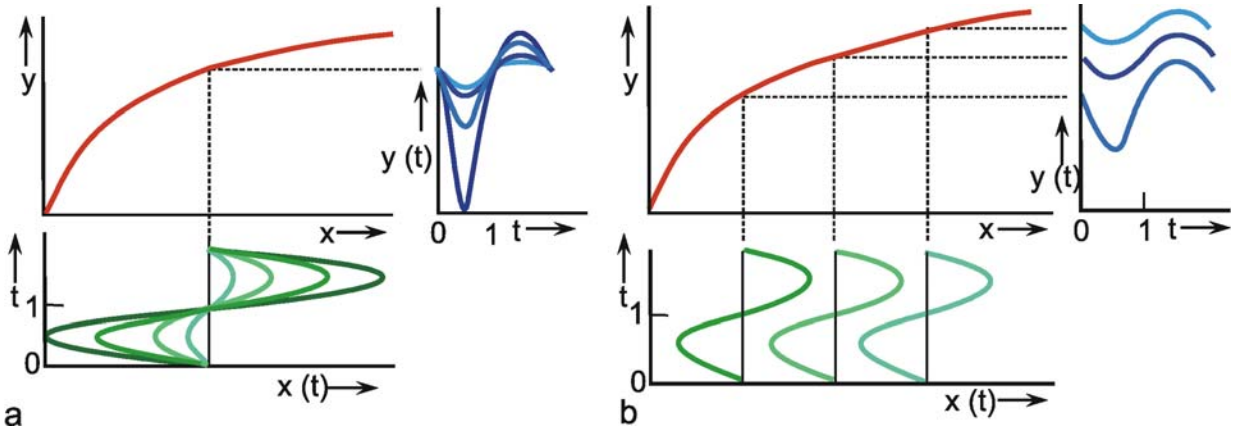


Fig. 5.**7 The effects of a nonlinear transformation.** (a) Effects of changing the input amplitude. (b) Effects of changing the point of operation. x(t) input function, y(t) output function (after Varju 1977)

If the form of a nonlinear characteristic shows no discontinuity in the slope, but only a smoothly changing slope, this may be called *soft* nonlinearity. This applies to the logarithmic and the exponential characteristic as long as the course of the input function stays above the threshold value. By contrast, those characteristics that show a discontinuous slope may be called hard nonlinearities. A hard characteristic in the strict sense does not occur in biology since the edge will, in reality, always tend to be more or less rounded down. The difference between a hard and soft characteristic is thus gradual. In the field of technology, such hard characteristics are more marked. Below, some of these hard characteristics will be described to the extent they are endowed with properties which may also play a role in biological systems. The typical effect of each characteristic will be illustrated in the way that the various response are drawn using an appropriately chosen sine function as input function.

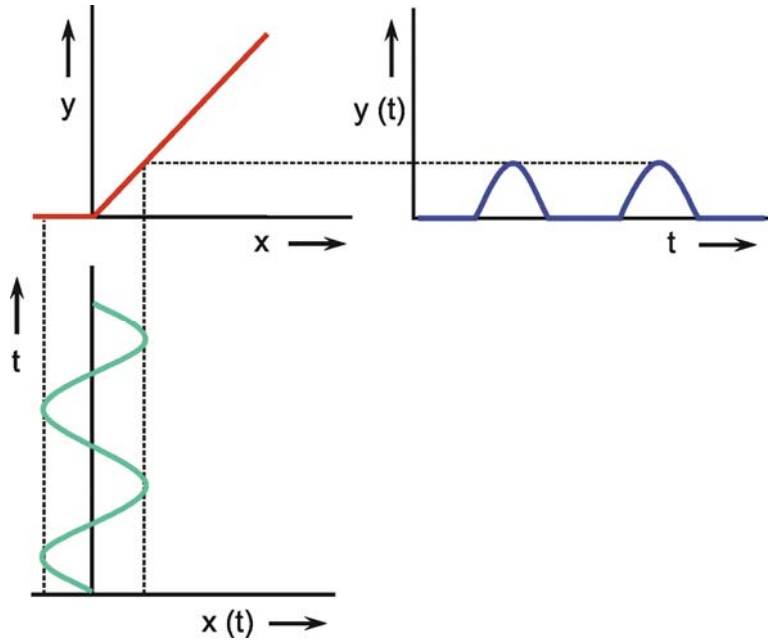


Fig. 5.**8 The characteristic of a rectifier.** Negative input values are made to zero. x(t) input function, y(t) output function

Figure 5.**8** shows the diode or rectifier characteristic. This has the effect that all negative input values take zero at the output. For a sine function with the mean value zero, all negative half-waves are thus cut off. The positive halfwaves are transmitted without any disturbance. Therefore it is also called the half-wave rectifier. If a range of operation is used that comprises only positive input values, the nonlinear property of this characteristic cannot be identified.

For a characteristic with saturation (Fig. 5.**9**), the output values remain constant, once the absolute value of the input has exceeded a certain value, the "saturation value" x_s. If the amplitude of a sine function at the input oscillation is thus sufficiently large, the extreme values of the sine curve are flattened due to this nonlinearity.

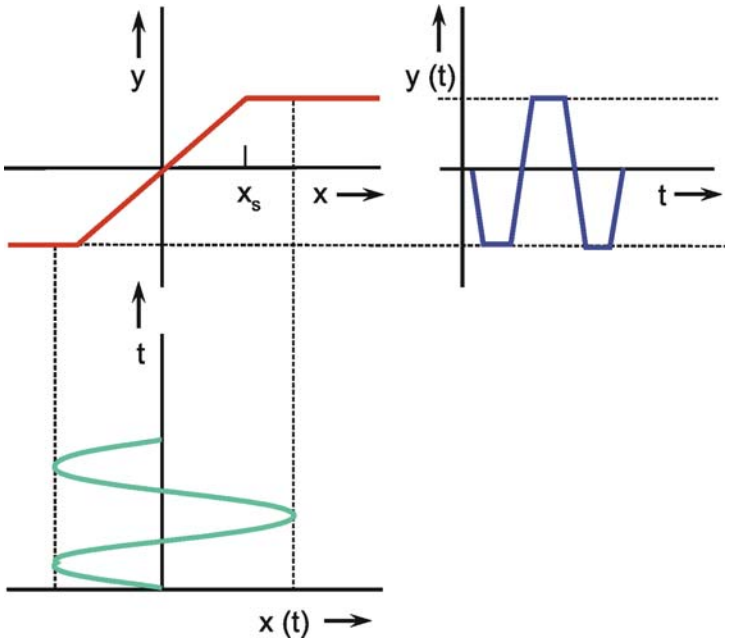


Fig. 5.**9 A characteristic with saturation**. When the input exceeds a given value x_s the output does not increase further. x(t) input function, y(t) output function.

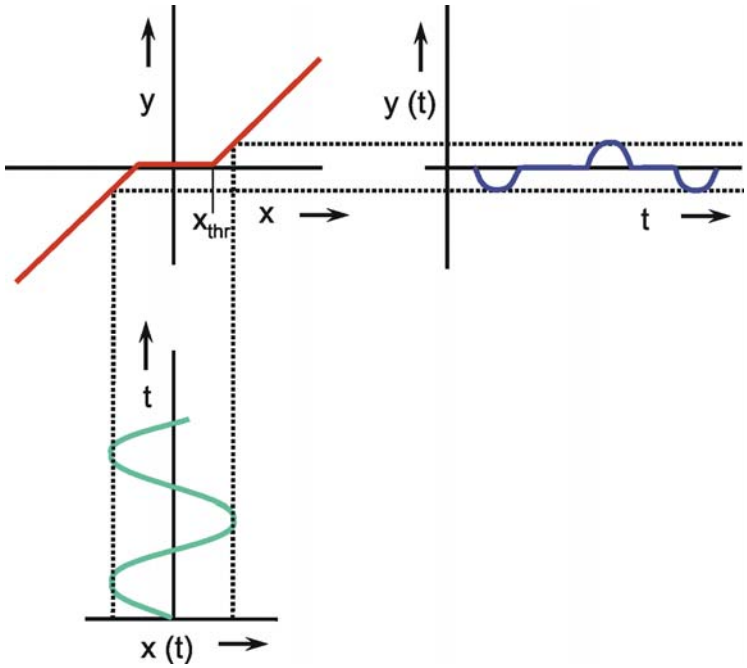


Fig. 5.**10 Characteristic with threshold.** The input value has to exceed a value x_{thr}, before the output value changes. x(t) input function, y(t) output function

If the characteristic is endowed with a threshold, usually known as a dead zone in the technical literature, the input quantity must first exceed a certain threshold value x_{thr} in order to show an output value that is different from zero (Fig. 5.**10**). In this case, only the values of the input function above the threshold are thus transmitted. As with the rectifier characteristic, those input functions whose values are all above the threshold are transferred without any disturbance.

In biology, a combination is frequently found of the properties of the characteristics shown in Figures 5.**8** to 5.**10**. For example, each sensory cell is endowed with a certain threshold as well as an upper limit. The subsequent synapses have a rectifying effect. The result is a characteristic whose general form can be approximated by three asymptotes, as shown in Figure 5.**11**. One example of such a characteristic is shown in Figure 5.**4**, although in this case, the ascending part follows a logarithmic rather than a linear course. It should also be mentioned that this characteristic is described by the function $y = A_0\ I/(I_W + I)$. This function is shown as a solid line in Figure 5.**4**, with I being plotted logarithmically. A_0 denotes the distance between the two horizontal asymptotes. The turning point is at $I_w/0.5\ A_0$. The asymptote, shown as a dashed line in Figure 5.**4**, is described by the formula $y = \text{const.} \log I/I_{thr}$. Here, I_{thr}, denotes the theoretical threshold value at which this asymptote cuts the zero line. As can be seen in Figure 5.**4**, the actual threshold value is lower, since this characteristic does not exhibit a discontinuous slope. Characteristics of this form are also known as sigmoid characteristics. Another often used form of a sigmoid characteristic is the "logistic" function $y = 1/(1 + e^{-ax})$. The parameter a describes the slope of the function (Fig. 5.**11 b).** This function uses the range between 0 and 1 on the ordinate. If the same function is spread to the range between -1 and 1, it is given by tanh ax.

The characteristic of a so-called ideal relay (Fig. 5.**12**) enables only two discrete output values, depending on whether the input value is located above or below a critical value (in this case zero). This characteristic thus describes an all-or-none effect. A function of this form is also called the Heaviside function.

The characteristic of a full-wave rectifier (Fig. 5.**13**) transmits the absolute value of the input function (i. e., the negative half-waves of the sine function are reflected along the abscissa). If the input function is placed symmetrically in relation to the zero point, the frequency of the fundamental of the output function is doubled by this characteristic, compared to the input function.

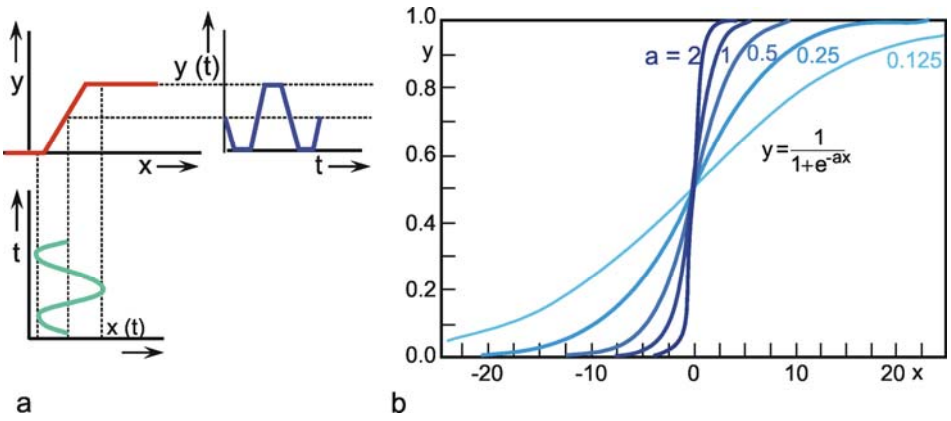


Fig. 5.**11 (a) A combination of a rectifier, a threshold, and a saturation, as it is often found in biological systems** (see Fig. 5.**4**). (b) A sigmoid characteristic corresponding to the logistic function, parameter a is a measure for the slope of the function. x(t) input function, y(t) output function

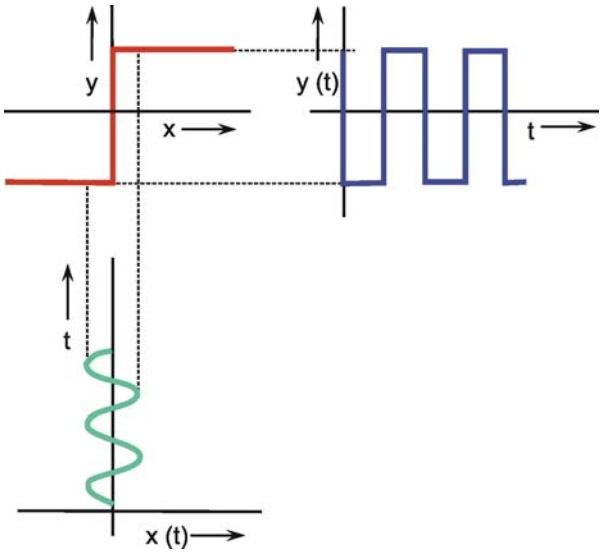


Fig. 5.**12 An all-or-none characteristic corresponding to the Heaviside function.** x(t) input function, y(t) output function

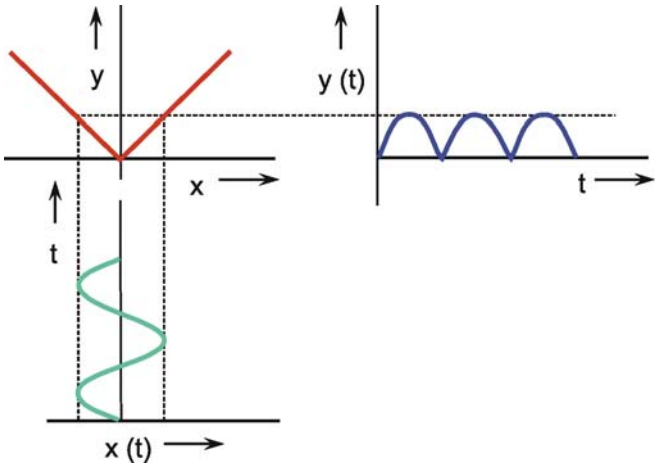


Fig. 5.**13 The characteristic of a full-wave rectifier.** Negative input values are made positive. x(t) input function, y(t) output function

With the exception of the full-wave rectifier, all the characteristics discussed so far are endowed with a monotonously ascending course; that is, their slope is always greater than, or equal to zero. A further example of a nonmonotonous characteristic is shown in Figure 5.**14**. As in the case of the two-way rectifier, an appropriate choice of range of operation will enable generation of a doubling of the fundamental frequency, or a phase shift of 180°. This occurs when the point of operation is changed from a to c in Figure 5.**14.** A doubling of frequency may occur when the point of operation lies at about position b. An example of a nonmonotonous characteristic of a biological system is shown in Figure 5.**15**.

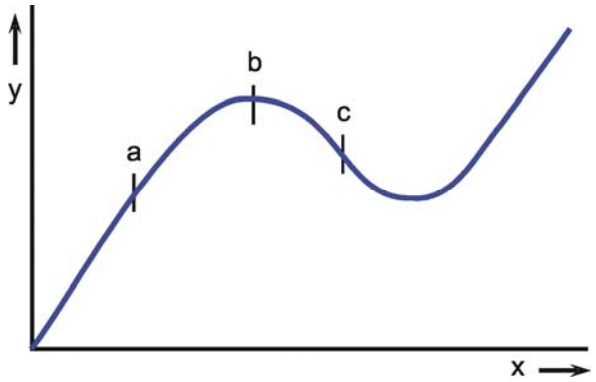


Fig. 5.**14 A nonmonotonuous characteristic.** a**,** b, and c mark three points of operation. x input, y output

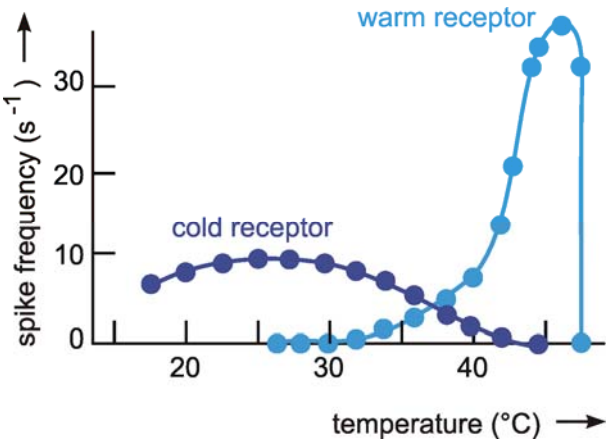


Fig. 5.**15 The discharge rate of cold receptors and warm receptors of the cat at different temperatures (tonic part of the responses).** In general, a given spike frequency can represent two different temperature values (after Schmidt 1973)

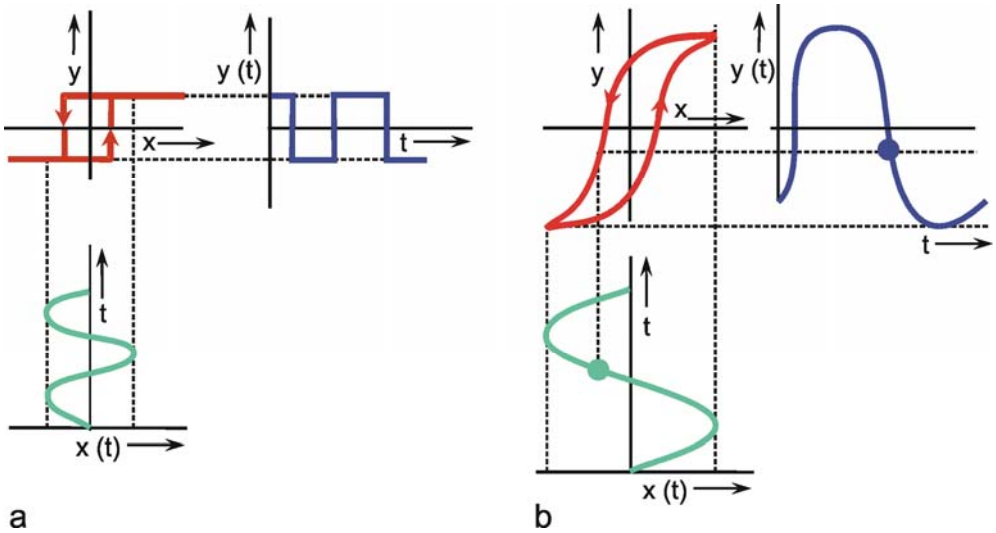


Fig. 5.**16 The characteristics of a realistic relay (a) and of a hysteresis (b).** In both cases, for a given input value the value of the output depends on the actual history, i. e., of how this input value was approached. This is symbolized by the arrows. x(t) input function, y(t) output function

In addition to exhibiting properties described as soft-hard, monotonous - nonmonotonous, static nonlinear characteristics can also be distinguished as univalued or multivalued. Unlike the univalued characteristics discussed so far, a multivalued characteristic is endowed with more than one output value corresponding to any one particular input value.

In a realistic relay, the value at which the switch is closed with increasing voltage usually differs from the value at which, during the drop in the voltage, the switch is opened again. The result is the multivalued characteristic of a realistic relay (Fig. 5.**16a).** This characteristic thus consists of two branches. Which of the two happens to apply at the time depends on the actual history; that is, on whether the input values happen to be increasing or decreasing. The temporal course is indicated by arrows. As with an ideal relay, only two discrete values are obtained at the output. Another characteristic of this type is the so-called hysteresis which matches the form of magnetization effects found in physics (Fig. 5.1**6b**).

> The transfer properties of a static characteristic can be illustrated by "reflecting" the input function at the characteristic to obtain the output function. Nonlinear characteristics distort the signal by increasing the content of higher harmonics. They can change the mean value (dc shift). Different types of static nonlinear characteristics may be grouped as soft - hard, monotonous - nonmonotonous, and univalued - multivalued.

5.3 The Dynamic Characteristic

Systems of the 1st and higher orders (i. e., systems with energy storage units) exhibit a typical transition behavior, for example in the case of the step response, until they reach a final output value. This transition during a constant input value (e. g., after a step or an impulse) is known as the dynamic part of the response or the transient. After the dynamic part fades, the stationary part of the response is obtained. This stationary part can be represented by a constant output value, as in the simple case of a first order filter. It is then also known as the static part. In undamped oscillatory systems of higher order (Fig. 4.**15b**) or in particular feedback systems (see Chapter 8.6), the stationary part may also be nonstatic. In these examples the stationary part of the response can be represented by infinitely continuing oscillations of a constant amplitude, i. e., it is periodic.

When applied to examples of tonic and phasic sensory cells, the dynamic part of the response of the phasic sensory cell to an ascending step thus exhibits a rapidly ascending increase initially, and then descends more slowly. In the case of a tonic sensory cell it consists of an ascending course. For the tonic sensory cell, the stationary part (in this case the static one, too) takes a higher value than the initial one, whereas for the phasic sensory cell, it remains at the level of the initial value.

Determination of the characteristic of a system endowed with dynamic properties poses a number of problems. This is demonstrated by the step response of a 1st order low-pass filter (solid lines) shown in Figure 5.**17a** together with the input step (dashed line). If one wants to read the output value corresponding to a particular input value x, a different value is obtained for each of the four selected time values t_1 to t_4. If the experiment is repeated with further step functions of varying amplitudes, a linear characteristic is obtained for each selected time t_i, but these characteristics differ from another (Fig. 5.**17b**). Only after the dynamic part is finished (t_∞) is a fixed characteristic obtained whose slope indicates the static amplification factor. During the dynamic part, the amplification factor changes continuously.

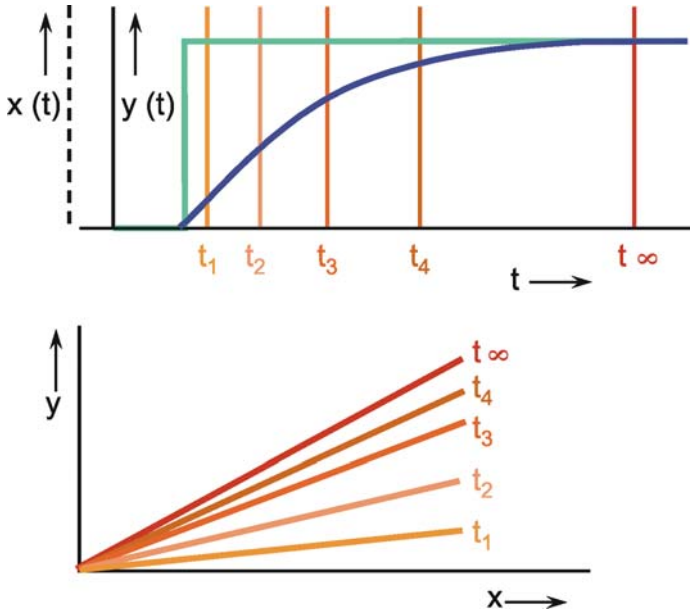


Fig. 5.**17 Example for a linear dynamic characteristic.** (a) Step input (green line) and step response (blue line) of a 1st order low-pass filter. Relating input and output of this filter yields different characteristics for each moment of time. These characteristics are shown for four selected time values t_1 – t_4, and for $t \infty$ in (b). x(t) input function, y(t) output function (after Varju 1977)

The dynamic characteristic of a system endowed with an energy store is thus not represented by a single curve, but rather a set with t as a parameter. Although the characteristic obtained for $t \infty$ is also known as the static characteristic of the dynamic system, construction of the output function by reflection at the characteristic is not possible in this case. The simplest way to obtain the response of a system endowed with a known dynamic characteristic is by simulation using electronic circuits or digital computers.

In Chapter 4 the amplification factors were mostly set at k = 1 for reasons of clarity. Since we have to distinguish between static and dynamic amplification factors, which was not explained in detail earlier, this can now be summarized as follows: the step response of the 1st order low-pass filter is $h(t) = k (1 - e^{-t/\tau})$. Here, k corresponds to the static amplification factor described above. It agrees with the ordinate value of the horizontal asymptote of the amplitude frequency plot. The step response of the 1st order high-pass filter is $h(t) = k_d e^{-t/\tau}$. The static amplification factor of a high-pass filter is always zero. k_d is the maximum dynamic amplification factor. It indicates the maximum amplitude of the step response, provided this is not influenced by low-pass properties of the system. k_d, too, agrees with the ordinate value of the horizontal asymptote of the amplitude frequency plot.

For a dynamic system, there is not just one characteristic but a set of characteristics, one for each time value. In nonlinear cases, the easiest way to illustrate the transfer properties of such a dynamic characteristic is to use a simulation.

6 Combination of Linear and Nonlinear Elements

So far only the properties of some nonlinear characteristics have been discussed. Since combinations of static nonlinear characteristics and linear filters are relatively easy to understand, some of them will be addressed below.

In a serial connection of a linear filter and a nonlinear system, the form of the output function depends essentially on the sequence of the two elements. For this reason, Chapter 6.1 will examine the sequence, linear element - nonlinear element, and Chapter 6.2 the converse sequence, nonlinear element - linear element.

6.1 The Sequence Linear Element - Nonlinear Element

To begin with, we will look at a system consisting of a low-pass filter and a nonlinear static characteristic connected in series, as shown schematically in Figure 6.**1.** Sine functions of low frequency at the input (x) of the low-pass filter result in high amplitudes at its output (z) and thus, at the same time, at the input of the nonlinear element. The higher the amplitude, the greater is the distortion by the nonlinear characteristic. This changes the mean of the output function and the amount of higher harmonics. High frequency sine functions at the input (x), result in low amplitudes at (z). Accordingly, the nonlinearity produces only slight distortions and slight shifts in the mean at (y).

In all cases, however, the sine responses constitute functions that are symmetrical in relation to a vertical axis through an extreme point of the function. If, accordingly, a high-pass filter is combined with a nonlinear element, the transfer properties of the total system are produced in an analogous way.

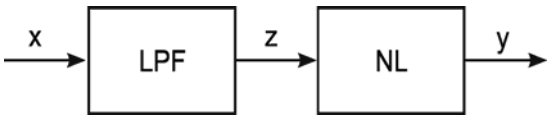


Fig. 6.**1 Serial connection of a linear filter (e. g., a low-pass filter) and a nonlinear characteristic**

6.2 The Sequence Nonlinear Element - Linear Element

If the linear filter is serially connected to the nonlinear one in the reverse order, different properties will result, since the harmonics, first produced by the nonlinear element, will undergo a variety of phase shifts, depending on a frequency of the subsequent linear filter. As a consequence, the output function can become asymmetrical in relation to a vertical axis through an extreme point. Figure 6.**2** and 6.**3** show the sine responses of such systems for different forms of characteristics and different nonlinear filters. In addition to the input function x(t) and the output function y(t), the output function of the nonlinear characteristic z(t) is also shown.

It should be mentioned, however, that one may not necessarily infer a series connection of a nonlinear characteristic and a linear filter from such an asymmetrical response to a sine function. Dynamic characteristics and multivalued nonlinear static characteristics can also produce such asymmetrical sine responses. (See the example shown in Figure 8.**16**). Those interested in a more detailed discussion of these systems are referred to the literature (e. g., Varju 1977).

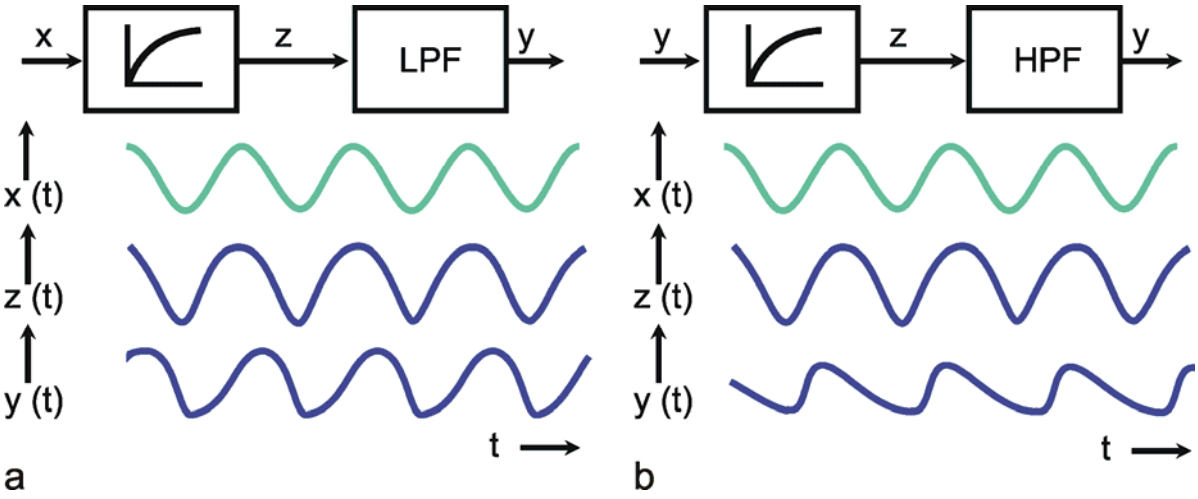


Fig. 6.**2 Serial connection of a nonlinear characteristic and a linear low-pass filter (a) or a high-pass filter (b).** The nonlinear characteristic is described by a function $z = \sqrt{x}$, x(t) input function, y(t) output function. z(t) describes the output of the first system

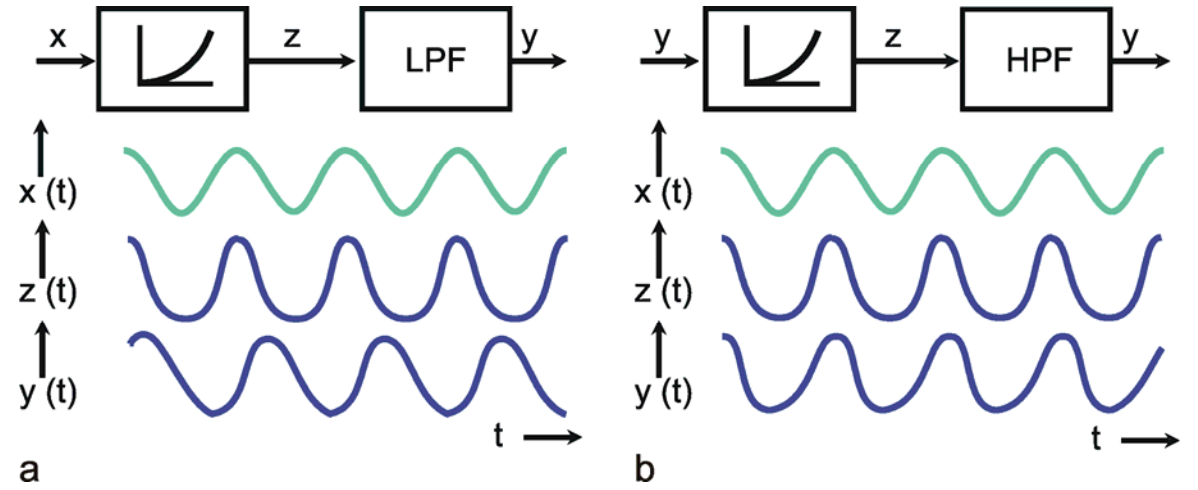


Fig. 6.**3 Serial connection of a nonlinear characteristic and a linear filter as in Fig. 6.2, but with a nonlinear characteristic following the function $z = x^2$.** x(t) input function, y(t) output function. z(t) describes the output of the first system

6.3 Unidirectional Rate Sensitive (URS) Element

A further example of the series connection of linear filters and nonlinearities is provided by a system that occurs quite frequently in biology. Some sensory cells, for example, respond only to an ascending, but not descending, stimulus. This means that the system responds only if the first derivative of the input function is positive. This system is called an "unidirectional sensitivity unit." Such a system is practical only if an adaptive system (that is, a high-pass filter) is serially connected to this unit. Otherwise, the output value, when responding to an alternately ascending and descending input function, would quickly approach an upper limit since only the ascending parts of the function exhibit a response.

Such a system may be present, for example, if a transfer of information is effected by the release of a chemical substance (synaptic transmitters, hormones). As the input value increases, there is a growing concentration of this chemical substance; but with a decreasing input value, the concentration

of the substance does not decrease accordingly, but may be reduced quite slowly (e. g., by diffusion or by a catabolic mechanism, producing an exponential decay). This corresponds to a serial connection of a unidirectional sensitivity element and a high-pass filter. This has become known as a "unidirectional rate sensitivity (URS)" (see Clynes 1968).

Figure 6.**4** shows the response of this system to a triangular function. Due to the positive slope, the ascending part of the triangular function is transmitted unchanged by the unidirectional sensitivity element. At the output, the following high-pass filter thus shows a typical ramp response (Fig. 4.**3**). For this reason, an exponential function is obtained which, if the ramp lasts long enough, ascends with the time constant τ of the high-pass filter to a plateau whose height is proportional to τ and the slope of the triangular function (ramp response, Fig. 4.**3d**). During the descent of the triangular function, a constant value is produced at the output of the unidirectional sensitivity element. Together with the time constant τ, the output value of the high-pass filter thus descends again to zero. If one shortens the ramp duration with the amplitude remaining unchanged, however, there is no longer sufficient time to attain the new plateau after ascending, or the zero line after descending. After a certain transient period, the ascending and descending parts are again antisymmetrical relative to each other. The mean of the output function shifts, however, as a function of the frequency (dc shift), as shown in Figure 6.**4**. This shows another qualitative property of a serial connection of a nonlinear element followed by a linear element. It is possible that there is a dc shift of the output signal, the size of which depends on the frequency of the input signal.

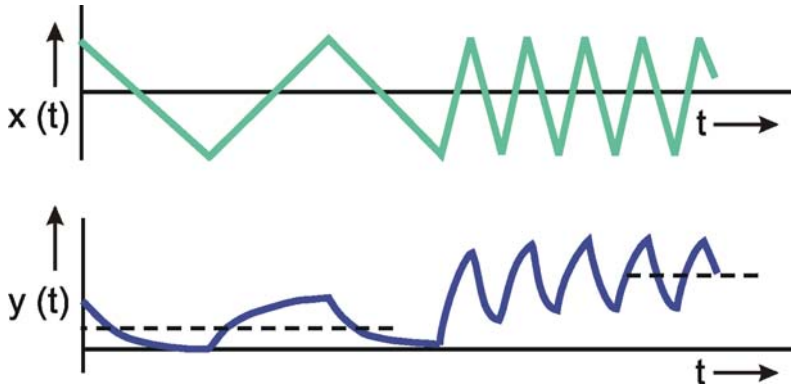


Fig. 6.**4 Ramp responses of an unidirectional rate sensitivity system.** This system responds only to those parts of the input function which have a positive slope. The subsequent high-pass filter determines the form of the response (ramp response, Fig. 4.**3**). When the frequency of the fundamental wave of the input function increases, the mean value of the output function (thin dashed lines) increases, too (after Milsum 1966)

When a system contains serially connected linear and nonlinear elements, then, depending on the order of the sequence, qualitative differences can be found with respect to its transfer properties. When the static nonlinear element is followed by a linear element, symmetrical input functions (e. g., sine functions) can be transformed to asymmetric forms; or, depending on the frequency of the input function, a dc shift of the output function could occur. This is not possible where the elements are in reverse order.

7 The Study of Nonlinear Systems

There are no general rules for study of nonlinear systems, since there is, as yet, no unified theory of nonlinear systems. Below, we will therefore describe several methods for the study of nonlinear systems, which should be considered no more than a collection of approaches.

First, it is important to check whether the system to be studied is linear or not. The simplest test is to examine the responses to steps of different amplitudes. Nonlinearity is present if a nonlinear relation between input and output is observed. However, the opposite is no proof of linearity of a system, though, as can be seen by examining the system shown in Figure 6.**4**. A better, though more difficult, test consists of studying the responses to sine functions of different frequencies and amplitudes. If the corresponding response functions turn out again to be sine waves in each case, the system is linear.

Once it is established that the system examined is of a nonlinear nature, one could try to simplify the problem by searching for a linear approximation to the system. If it is endowed with a nonlinear characteristic, one could try to undertake a "linearization" of the system by using only small input amplitudes, if possible. As Figure 5.**7** shows, an approximation to the linear case can always be obtained with soft characteristics. By changing the position of the point of operation, a nonlinearity can be divided into several small linear segments. As an example, Figure 7.**1** shows the result of an investigation of a position reflex. (A detailed description of this system is given in Chapter 8.4.) Whereas the response to the input function of a large amplitude differs clearly from the form of a sine function, the response to a small amplitude function is very similar to a sine function. When the system that is being studied is linear at least for selected ranges, a "linearization" can often be realized by first examining only the linear range.

In case of hard nonlinearities, a linearization can be obtained in a simple way only if an appropriate choice can be made regarding the point of operation. For example, the nonlinearity of the system shown in Figure 6.**4** cannot be linearized for sine function at the input. By contrast, a linearization for a step function at the input is possible, as this function contains no descending parts. (For other possibilities of linearization of such nonlinearities see Spekreijse and Ousting 1970).

If the system is endowed with a distinct dynamic characteristic, it is useful, first, to study the static behavior of the system (see Chapter 5.3). Once the static properties are known, the subsequent study of the dynamic properties will be less complex.

As we have seen in the previous sections, many nonlinear systems provide an asymmetrically distorted sine response. The Bode plot of such a system would be difficult to measure because the phase frequency plot cannot be determined exactly since a phase angle is defined only between two functions of the same type, e. g., two sine functions. If a reduction in amplitude does not produce a sufficient linearization (i. e., a sine function at the output) one option would be to study just the fundamental wave of the output function (see Chapter 3) and then to construct the Bode plot for this. Some mathematical operations are required, though, to determine the fundamental wave of the output function.

As mentioned earlier, for linear systems various input functions provide the same information about the system. Since this does not apply to nonlinear systems, it is possible that, with each of these input functions, different information will be obtained.

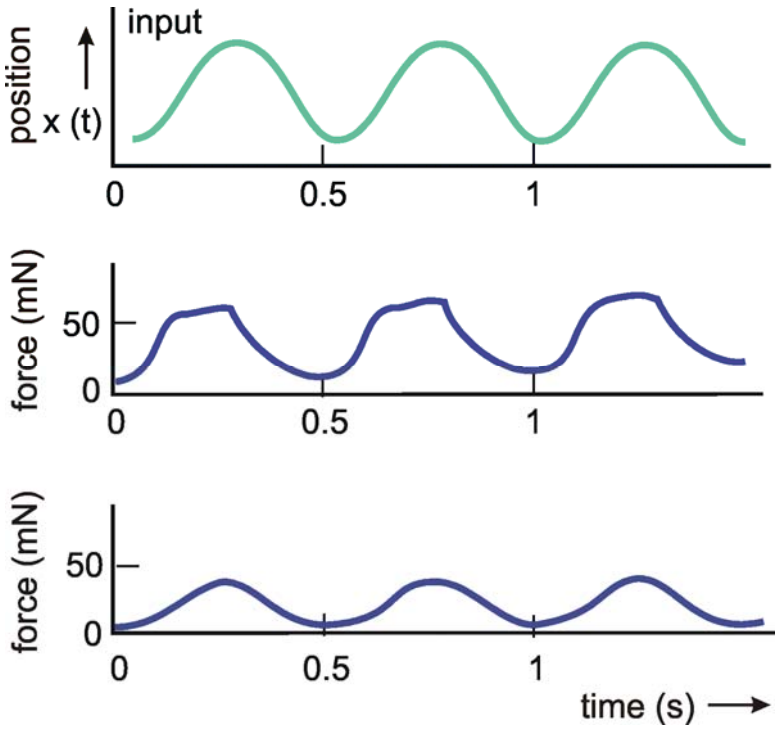


Fig. 7.**1 "Linearization" by means of small input amplitudes.** The example shows the investigation of a position reflex (see also Fig. 8.**6**). The upper trace shows the input function (position). The two lower traces show responses (force) for sine wave inputs of different amplitude, namely 0.3 mm, above, and 0.08 mm, below. The response resembles a sine function much better in the second case compared to the first.

As the example in Chapter 6.3 has already shown, it is quite conceivable that a specific property of the system can only be identified when using one particular input function, but not with another. For this reason, it is necessary in studying nonlinear systems that the responses to as many different input functions as possible should be considered. In order to characterize the system as completely as possible, the amplitude (in particular) in addition to the type of input function, should be varied within the widest possible range.

An input function that is sometimes used in the study of nonlinear systems, and which has not been mentioned so far, is the double impulse (i. e., two successive impulses with a short interval between). In a linear system the responses to the two parts of the function would add up linearly. From the type of additivity found in a nonlinear system, it is also possible to draw conclusions about the underlying elements. Analogously, the double step - two successive step functions - can be used. For a more detailed study of this method, the reader is referred to the literature (Varju 1977).

The more complicated, and thus harder to comprehend, the systems under view tend to be, the more it is necessary to accompany experimental investigation by a simulation of the system. This is especially true for the study of dynamic nonlinear characteristics. First, on the basis of initial experimental findings, a hypothesis is made about the structure of the system. This hypothetical system is then simulated by way of electronic circuits or digital computer programs. The hypothesis is tested, and modified if necessary, by comparing the responses of the real and the simulated systems. This procedure is repeated until the behavior of the model agrees with that of the biological system in terms of the desired degree of accuracy. Simulation is not an end in itself; rather, in its course, a number of insights, relevant to the further study of the system will result.

Experience has shown that, in attempts to formulate a merely qualitative hypothesis of a system's construction on the basis of its initial study, such a qualitative interpretation may easily contain unnoticed errors, e.g., by overlooking some inconsistencies. This is especially true if nonlinear properties are present, since their influences on the behavior of the total system are often hard to assess. By making a quantitative hypothesis, needed for the design of a simulation model, such errors are far more likely to be recognized. When realizing the simulation, at the least, one is obliged to formulate an elaborately developed hypothesis, which can then also be tested, quantitatively.

If agreement is found between model and system, it is relatively easy (on the basis of the known model, and by a systematic selection of new input functions) to discover new experiments suited to testing the hypothesis even more thoroughly, and thus making it more reliable. On the other hand, if inconsistencies between the real system and the model are found, which is usually the case it is now possible, since the discrepancies can be clearly defined, to modify the hypothesis, and thus the model, to ensure an improved description of the system. If, in the course of this procedure, two alternative hypotheses emerge, both of which agree with the results obtained so far, input functions can be selected, on the basis of the models, that will help to distinguish between the two hypotheses. In any case, the simulation is thus a heuristically valuable tool in the study of a real system.

Once a simulation has been achieved with the desired degree of accuracy, the model will provide a concise description both of the different experimental data and of the real system. Moreover, the model will come in useful when the response of the system is sought to input functions that have not previously been examined in detail, and whose investigation within the framework of the real system would be too complicated. The unknown response to the system can be obtained quickly and easily by using the model. One example of this is the study of processes covering an extremely long time period. Their simulation can be conducted more speedily.

If the simulation has been concluded successfully in the sense that the responses to a large number of greatly varied input functions can be simulated by means of a unified model, it may ultimately be assumed that the essential properties of the real system have been most probably identified and described.

There is no systematic procedure for investigating nonlinear systems. Rather, a number of rules have to be considered. Simulation is an important tool which can provide a concise description of the system and, more importantly can help to design new experiments.

Box 2

An Example: The Optomotor Response

A classic example of a nonlinear system is the optomotor response investigated by Hassenstein, Reichardt, and colleagues (Hassenstein 1958a, b, 1959, 1966; Hassenstein and Reichardt 1956; Reichardt 1957,1973). If a visually oriented animal is placed in the center of an optically structured cylinder, which, for example, contains vertical stripes, and this cylinder is rotated around its vertical axis, the animal tries to move its eyes with the moving environment. This can result in nystagmus-like eye movements (e. g., in humans) or in movements of the head or of the whole body (as in insects, for instance). The biological sense of this optomotor response is assumed to be an attempt keep the sensory environment constant.

In most experiments, the animal is held in a fixed position, and its turning tendency is measured in different ways, for example by recording the forces by which the animal tries to turn. When both eyes, or at least a large part of one eye, are regarded as one input channel, and the turning tendency is measured as output, the whole system can be considered as a band-pass filter (see Box 1).

The eyes of insects, however, consists of a large number of small units, the ommatidia. It is possible to stimulate neighboring ommatidia with independent step functions by changing the light intensities projected onto separate ommatidia. Using this technique, it was shown that, in order to detect the movement of the visual pattern, stimulation of two neighboring ommatidia of an eye is sufficient. Thus, for a more detailed analysis, a system can be considered that has two inputs (two ommatidia called O1 and O2) and one output channel (the turning tendency TT, Figure B2.**1**). For this system, where the first intensively studied object was the beetle *Chlorophanus,* the following results could be obtained.

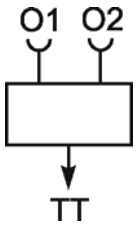


Fig. B 2.**1 The elementary movement detector**. Two ommatidia, O1 and O2, form the input; the turning tendency TT represents the output of the system

The two ommatidia were stimulated by optical step functions, i. e., by a stepwise increase or decrease in the light intensity. It could be shown that the absolute value of the light intensity of the step functions was not relevant over a wide range; only the size of the intensity change influenced the turning behavior of the animals. This indicates that only a high-pass filtered version of the input function is processed. This is shown by the two high-pass filters in Figure B2.**3**.

When two neighboring ommatidia O1 and O2 are stimulated with consecutive positive steps, i. e., by increasing the light intensity at O1 and, after a delay, at O2, a reaction following the apparent movement (O1 ⟶ O2) was observed. The same reaction was obtained when these ommatidia were stimulated by consecutive negative steps, i. e., with decreasing light intensity. This is shown in the Table B2.**2** where this movement direction is shown as positive. When, however, the first ommatidium received a positive step and, consecutively, the second ommatidium a negative step, the animal tried to move in the opposite direction. This means that the direction of the apparent movement was reversed (O2 ⟶ O1), which is shown by negative signs in Table B2.**2**. The same result was obtained when the first step was negative and the second positive. These results are summarized in Table B2.**2**. The comparison of the signs indicates that some kind of multiplication takes place between the two input channels (Fig. B2.**3**).

O1	O2	TT
+	-	+
-	-	+
+	-	-
-	+	-

Fig. B 2.**2 The table shows the behavior of the system in qualitative terms.** The stimulation of the ommatidia is marked by a positive sign, when the light intensity is increased, and by a negative sign, when it is decreased. In all cases, O2 is stimulated after O1. The turning tendency TT is positive when the movement follows the sequence of the stimulus, i. e., from O1 to O2, and negative, when the animal turns in the opposite direction

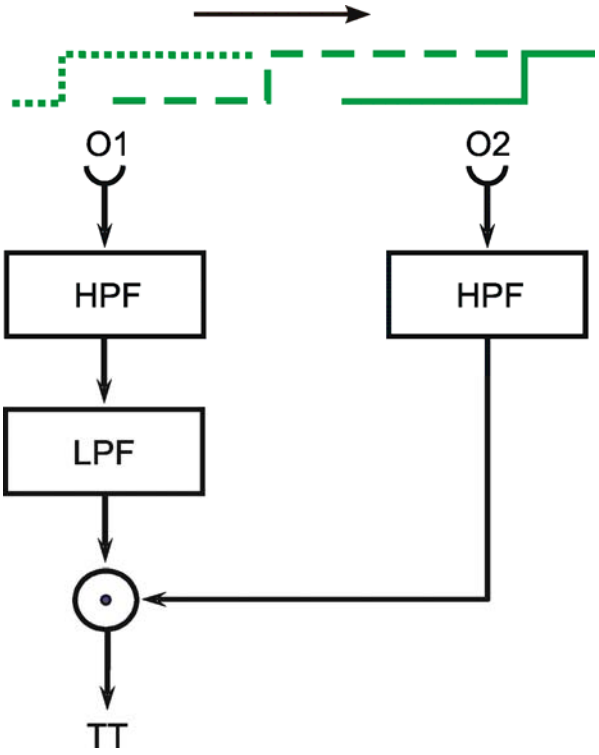


Fig. B 2.**3 A simple version of a movement detector** which responds to movements from left to right, i. e., from O1 to O2 (see arrow). LPF: low-pass filter, HPF: high-pass filter

However, a direct multiplication of the high-pass filter output signals does not agree with the following results. When the neighboring ommatidia O1 and O2 were stimulated with consecutive positive steps, but the delay between the two steps varied, the intensity of the reaction depended on the duration of the delay, being small for very short delays, strongest for a delay of about 0.25 s, and again decreasing for longer delays, but still detectable for a delay of up to 10 s. This means that the multiplication provides a zero result when the signal in the first input channel occurs at the same time as that of the second input channel. Multiplication provides the highest value when the first signal advances the second by about 0.25 s and the result steadily decreases if the delay increases further. This result is obtained when we assume that the signal from the first ommatidium, O1, having passed the high-pass filter, is transmitted by a low-pass filter. The resulting band-pass response is then multiplied by the signal of the second channel, O2 (Fig. B2.**3**). This hypothesis was supported by a number of control experiments (Hassenstein 1966). Thus, the elementary movement detector could be shown to consist of two parallel channels containing linear elements, and a nonlinear calculation, namely a multiplication, of the signals of the two channels as shown in Figure B2.**3**.

It is not shown in Figure B2.**3** that the real system, of course, consists of more than two such ommatidia, rather, the output of a large number of such elementary movement detectors is summed to produce the optomotor response. Furthermore, in addition to the movement detector monitoring the movement in the direction from O1 to O2, mirror image detectors monitor the movement in the reverse direction.

The properties of this movement detector could also be interpreted in the following way. Two input functions $x_1(t)$ and $x_2(t)$ are measured by the two ommatidia O1 and O2, respectively. For a relative movement between the eye and the environment, both ommatidia receive the same input function, except for a time delay Δt which depends on the relative speed. Thus, the input of the second ommatidium $x_2(t)$ equals $x_1(t+\Delta t)$. If, for the sake of simplicity, we neglect the transfer properties of the

filters before the multiplication, the output signal corresponds to the product $x_1(t)\ x_2(t) = x_1(t)\ x_1(t + \Delta t)$. This signal is not directly fed to the motor output, but is first given to a low-pass filter with a great time constant (not shown in Figure B2.**3**). As this low-pass filter can be approximated by an integrator (Chapter 4.8), the actual turning tendency might be described by $TT = \int x_1(t)\ x_1(t + \Delta t)\ dt$. This corresponds to the computation of the correlation between the two input functions. The correlation is higher, the more similar both functions are, which means the smaller Δt is, which, in turn, means the higher the speed of the movement. Thus, the movement detector, by determining the degree of correlation between the two input functions, provides a measure of speed.

8 Feedback Systems

8.1 Open Loop Control - Closed Loop Control

There are two ways of assigning a desired value to a variable, such as the position of a leg or the speed of a motor car-to give a biological and a technical example. One way of doing this is by the so-called open loop control. It requires knowledge as to which value of the controlling variable (input value) corresponds to desired value of the controlled variable. For the first example, this would be the connection between the spike frequency of the corresponding motor neurons (input) and the leg position (output), and for the second, technical example, it would be the connection between the position of the throttle and the driving speed. After adjusting the appropriate controlling variable, the correct final value is normally obtained.

If the value of the variable is influenced by some unpredictable disturbance, (e. g., by a force acting of the leg, or, in the case of the moving car, by a speed-reducing head wind) the effect of such a disturbance cannot be compensated for by such an open loop control system. Open loop control, therefore, presumably occurs in cases where no such disturbances are expected. For example the translation of a spike frequency into tension of a muscle, if we disregard effects like fatigue, or, in the case of a sensory cell, the transformation of the stimulus intensity into electrical potentials, thus correspond to an open loop control process.

If disturbances are to expected it is, however, sensible to control the value of the output variable via a closed loop control. This is done by using a feedback system. In this sense, a feedback control system is a system designed to keep a quantity as constant as possible in spite of the external influences (disturbances) that may affect it. It will be explained below how a feedback control system can eliminate, or at least reduce, the effect of such unpredictable disturbances.

In contrast to an open loop system, a closed loop system or feedback system may compensate effects of external disturbances.

8.2 The Most Important Terms

Before describing in detail the functioning of a feedback control system, the most important terms required for this will be presented schematically (Fig. 8.**1**), and using a highly simplified example, namely the pupil feedback system. (A detailed description of this system is found in Milsum (1966), for morphological and physiological details see Kandel et al. 1991.) The pupil feedback system controls the amount of light reaching the retina of the eye by reducing the diameter of the pupil, via the iris

muscles, as illumination of the retina increases. As illumination decreases, the size of the pupil increases accordingly. The controlled variable, whose value should be kept as constant as possible, despite the effect of external influences (here the illumination of the retina) is called output variable or control variable. This is the output variable of the total "feedback control system".

The value taken by the output variable, the so-called actual output (1) (that is, the value of the illumination on the retina in this example) is measured by the photoreceptors and transmitted to the brain in the form of spike frequencies. The element measuring the actual output (here the photoreceptors) is known as feedback transducer (2). This actual output measured (and therefore generally transformed) by the feedback transducer is compared to the reference input (4). This reference input may be considered an input variable of this system. It could be imagined to be generated by a higher level center and constitutes a measure of the desired value of the output variable. But, as will be shown, it need not be identical to the value of the output variable. (This follows already from a comparison of dimensions: in our example, the actual output variable is measured in units of illumination, the reference input in the form of a spike frequency.)

The actual output (which is encoded in spike frequencies) is compared to the reference input (probably also encoded in spike frequencies) by the comparator (3). As shown in Figure 8.**1**, this comparison is realized in such a way that the actual output measured by the feedback transducer is subtracted from the reference input. The difference between the reference input and the actual output, as measured by the feedback transducer, is known as an error signal (5). The error signal is transformed into a new signal, the control signal (7), via the controller (6). Again, in our example, the control signal is given in the form of spike frequencies. Via the corresponding motorneurons, action potentials influence the iris muscles, which constitute the actuator, or effector (8), of the feedback system. The actual output is influenced by the actuator, in the present example by dilatation or contraction of the pupil.

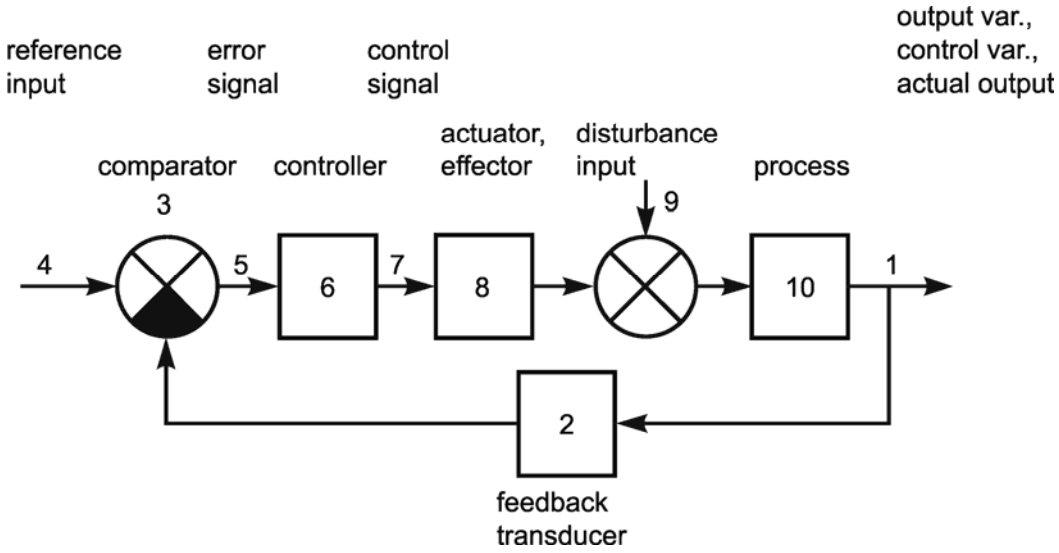


Fig. 8.**1 Schematic diagram of a negative feedback control system**

Apart from the actuator, an external disturbance, such as light from a suddenly switched on source, may also affect the output variable. The influence of this disturbance input (9) is assumed to be additive, as shown in Figure 8.**1**. (Disturbances may, of course, occur at any other point in the feedback system, although this will not be discussed here at this stage; but see Chapter 8.6.)

Actuator and disturbance input thus together influence the actual output variable. In our example this influence produces a change in the output value with practically no delay, because of the high speed

of light. In other systems, such as in the control of leg position, the output variable (because of the leg's inertia and frictional forces) does not change within the same course of time as the force generated by a muscle and a disturbance input. These properties of the system (here inertia and friction) are summarized in an element known as process (10). In the example of the pupil control system, the process thus corresponds to a constant (zeroth order system) and can therefore be neglected in this case.

Why is it that a feedback system can reduce the influence of a disturbance input, whereas this is not possible with an open loop control? The crucial difference lies in the feedback element of the former. Via the feedback loop, which contains the feedback transducer, information relating to changes in the actual output are registered. The error signal provided by the comparator allows determination of whether these changes are being caused by a change in the reference input or by the occurrence of a disturbance input. If the reference input has remained constant, the change has to be attributed to a disturbance input. In this case, an error signal different to zero is obtained. This constitutes a command to the controller, which in turn 'tells' the actuator to counteract the disturbance. An increase in the actual output by a disturbance has to be offset by a reduction from the actuator. If a disturbance input is introduced that reduces the value of the output quantity, a positive influence of the actuator on the output quantity is obtained. Because of this sign reversal in a feedback control system, one speaks of negative feedback. It should however be stressed that, as explained in the latter case, such a negative feedback can also produce positive effects, depending on the sign of the error signal.

If the reference input continues to exhibit the same fixed value, this is known as a homeostasis. If it is variable, however, this is called a servomechanism. According to the filter properties with which the controller (6) is endowed, we can distinguish between three different types, whose most important properties will be listed here, but explained later (Chapter 8.5). (This description is, however, only valid when the feedback loop contains a proportional term. In Chapter 8.5, additional possibilities are also discussed.) If (6) is a proportional term with or without delay, i. e., a low-pass filter or a constant, we speak of a proportional controller (P-controller). The most important property of a system containing a P-controller is the fact that, even in the stationary case, in reaction to a step-like disturbance input, an error signal unequal to zero will always persist and its value is proportional to that of the disturbance value. If (6) consists of an integrator, an integral controller (I-controller) is obtained. Unlike a P-controller, in a system with an I-controller the error will be zero in the stationary case. If (6) is a high-pass filter, we speak of a differential controller (D-controller). A feedback system with a D-controller shows a brief response to a step-like disturbance input, which however quickly descends to zero. The compensatory effect of this system is therefore only of short duration and the stationary error signal is equal to the disturbance input. The term adaptive controller is used if the properties of the controller are not fixed but can be influenced by external signals.

> Negative feedback is required to compensate for the effects of a disturbance input. Three types of controllers are distinguished: an I-controller fully compensates the effect of a constant disturbance input. By a P-controller, the effect of such disturbance input is compensated, but for a proportional value. A D-controller shows a compensatory effect only for a short time.

8.3 A Simple Feedback System

The basic quantitative properties of a feedback system will be explained by using the simplified system, shown in Figure 8.**2**. The various elements mentioned in Figure 8.**1** have been combined here to form just two elements of the zeroth order endowed with the amplification factors k_1 and k_2, i. e., two

proportional terms without delay. We are thus dealing with a proportional controller. With the aid of auxiliary variables h_0, h_1 and h_2, as shown in Figure 8.**2**, the following relations can be formulated for this system:

$$h_1 = k_1 h_0 \qquad (1)$$

$$h_2 = k_2 y \qquad (2)$$

$$h_0 = x_r - h_2 \qquad (3)$$

$$y = d + h_1 \qquad (4)$$

This results in

$$y = \frac{k_1}{1 + k_1 k_2} x_r + \frac{1}{1 + k_1 k_2} d \qquad (5)$$

With the abbreviations $C_1 = k_1/(1+k_1k_2)$ and $C_2 = 1/(1+k_1k_2)$ we obtain

$$y = C_1 x_r + C_2 d \qquad (6)$$

This equation illustrates how output quantity y depends on reference input x_r and disturbance input d. If the disturbance input d = 0, the equation can be simplified to $y = C_1 x_r$. The actual output is thus proportional (not equal) to the reference input x_r. Equation (5) also shows that, in the case of a P-controller, the actual output is proportional to the disturbance input. This accounts for the property, as mentioned, of a system with P-controller, namely, that the effect of a constant disturbance input can never be turned into zero, but, weighted with the factor C_2, will always contribute to the output value. With a reference input $x_r = 0$, and a disturbance input $d \neq 0$ the deviation from the originally desired value y = 0 no longer takes the value d, but still retains the value $C_2 d$.

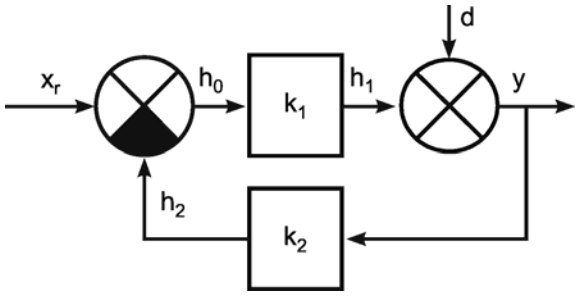


Fig. 8.**2 A simplified version of** a feedback controller with **static elements.** x_r reference input, d disturbance input, y control variable, h_0, h_1, h_2 auxiliary variables, k_1, k_2 constant amplification factors

The size of this deviation depends on the choice of constants k_1 and k_2. The effect of the feedback system is large when C_2 is small, that is, when k_1 and k_2 are large. In order not to make the amplification factor of the feedback system, C_1, too small, k_1 should be large, and k_2 small. Since both goals cannot be realized at the same time, a compromise is inevitable. As will be shown later, there are additional conditions that have to be fulfilled, since otherwise the system may become unstable (see Chapter 8.6). The following can be stated with respect to the possible values of constants C_1 and C_2: Both quantities are always larger than zero. Since k_1 and k_2, too, are always larger than zero, we

get $C_2 < 1$. Since $C_1 = k_1 C_2$, C_1 is smaller than 1 if $k_1 < 1$. If k_1 is sufficiently large, C_1 may also be > 1, but only if k_1 remains < 1. If $k_2 \geq 1$, C_1 will always be smaller than 1.

If one wants to investigate such a feedback system, in principle the values of the constants k_1 and k_2 are obtained in the following way. In the first experiment, the disturbance input d is kept constant, while the reference input x_r is varied. The slope of the line obtained when plotting y versus x_r provides the constant C_1. Conversely, if the reference input x_r is kept constant, and the disturbance input varied, the slope C_2 is obtained when plotting y versus d. Values k_1 and k_2 can be calculated from C_1 and C_2, after which the feedback system will have been completely described.

If the feedback control system is not endowed with simple proportional terms, but with linear filters exhibiting a dynamic behavior, the approach just outlined is still applicable for the static part of the response. k_1 and k_2 would then be the static amplification factors. If one is also interested in the dynamic part of the behavior of the feedback system, however, calculations will be more difficult because, in equations (1) and (2), convolution integrals will now occur with the weighting functions $g_1(t)$ and $g_2(t)$ (Fig. 8.**3**), as follows:

$$h_1(t) = \int g_1(t - t')h_0(t')dt' \qquad (1')$$

$$h_2(t) = \int g_2(t - t')y(t')dt' \qquad (2')$$

$$h_0(t) = x_r(t) - h_2(t) \qquad (3')$$

$$y(t) = d(t) + h_1(t) \qquad (4')$$

Equations (1')-(4') cannot be solved unless special mathematical methods are used. These are mentioned in Appendix I.

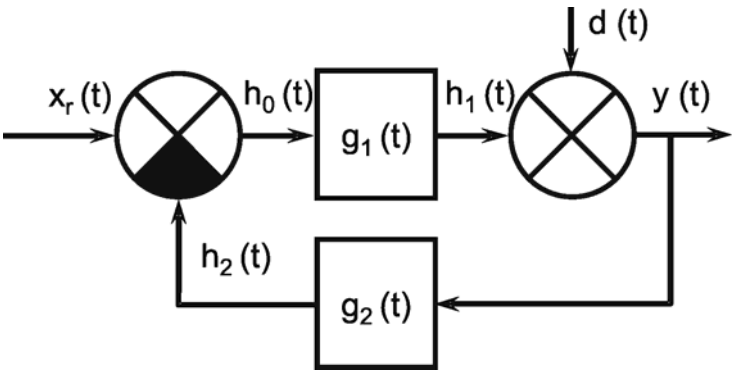


Fig. 8.**3 A simplified version of a feedback controller with dynamic elements.** $x_r(t)$ reference input, d(t) disturbance input, y(t) control variable, h_0 (t), $h_1(t)$, $h_2(t)$ auxiliary variables, $g_1(t)$, $g_2(t)$ weighting functions, describing the dynamic properties of the systems in the feedforward and the feedback channel, respectively

It will be briefly noted here how the input-output relations of these feedback systems can be determined for the static case, if nonlinearities occur. For this purpose, a static nonlinear characteristic will be assumed instead of the constant k_1. Equation (1) then changes from $h_1 = k_1 h_0$ to

$$h_1 = f(h_0) \qquad (1'')$$

Function $f(h_0)$ here defines the nonlinear characteristic. By combining equations (2), (3), and (4), we obtain a new set of equations:

$$h_1 = -\frac{h_0}{k_2} + \frac{x_r}{k_2} - d \qquad (7)$$

$$h_1 = f(h_0) \qquad (1'')$$

If it is possible to represent f(ho) as a unified formula, one can try to solve these equations arithmetically. Otherwise a graphic solution is possible. To this end, functions (1 ") and (7) are represented in a coordinate system using h_o as independent and h_1 as dependent variable (see Fig. 8.**4**). The intersections of the lines are the points whose coordinates fulfill the equations, that is, they are the solutions of the equation system. Figure 8.**4** shows this for the example of $f(h_0)$, being a nonlinear characteristic endowed with a threshold and saturation. The corresponding output value y can be obtained from the coordinates of the intersection, with reference input x_r and disturbance input d taking known values. If x_r or d are changed, a new value for y is obtained by a parallel shift of the line, accordingly. Depending on the form of the characteristic, several intersections are possible, and several output values y may result for the same input values x_r and d. This occurs, for instance, when the nonlinear characteristic is in the form of a hysteresis. Similarly, if, in the feedback system, a static nonlinear characteristic is present instead of constant k_2, equation (7) will become a nonlinear function. In this case, too, the solutions may be obtained by the graphic method as discussed.

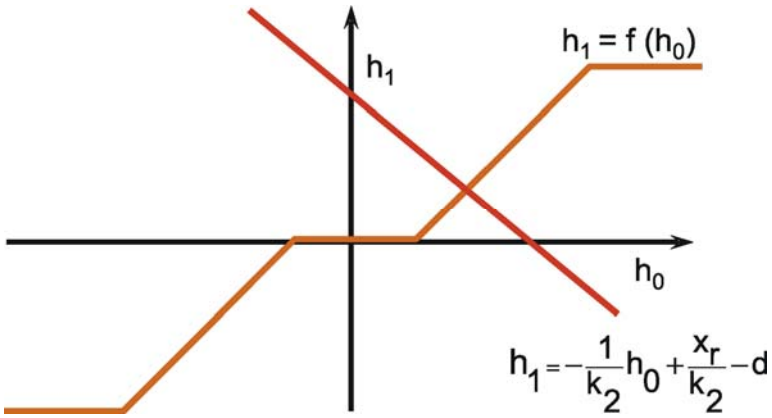


Fig. 8.**4 Graphical solution of input-output relation of a negative feedback system with a nonlinear characteristic $f(h_0)$ in the feedforward channel.** Equation (7) is graphically illustrated by the red line. Its position depends on the values of the parameters x_r and d. Changing these parameters results in a parallel shift of this line. As an example, the nonlinear characteristic $h_1 = f(h_0)$ is endowed with threshold and saturation (yellow lines). For given values x_r and d the intersection of both lines indicates the resulting variable h_1. Equation(4) ($y = h_1+d$) allows determination of the value of the actual output

When the static properties of a feedback system are considered, a simple calculation is possible (see e. g. equations (1) -(5). This can also be extended to systems with nonlinear static characteristics. For the dynamic case, more sophisticated methods (e. g., Laplace transformation) have to be used.

8.4 The Study of a Feedback System

A feedback system can be described as a system endowed with two input values, reference input x_r and disturbance inputs d, as well as an output, actual output y (Fig. 8.**5a**). If this "black box" is filled with a simple feedback system, as shown in Figure 8.**2**, the problem appears somewhat complicated at first (Fig. 8.**5b**). If, on the other hand, the equation (6) is taken into account, the feedback system can be simplified (Fig. 8.**5c**). As discussed, the values of C_1 and C_2 - and, therefore, the values of k_1 and k_2 - can be determined easily. Frequently, however, the problem in studying biological feedback systems is posed by the fact that reference input x_r is not accessible by experimentation. But in spite of this limitation, there are two possible ways of studying a feedback system, at least partially, in this case.

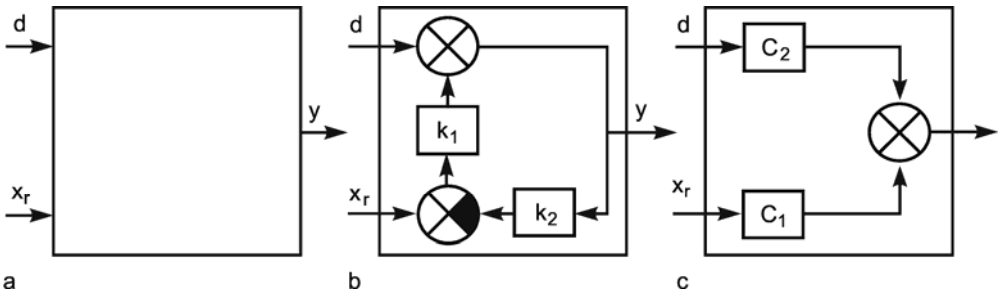


Fig. 8.**5 (a) The negative feedback system as a black box.** (b) A view inside the black box. (c) Rearrangement of the internal structure shows an equivalent feedforward system. x_r reference input, d disturbance input, y control variable, or actual output, C_1, C_2, k_1, k_2 constant amplification factors

One possibility is to examine the relation between disturbance input and actual output. In Figure 8.**5 c**, this corresponds to the determination of value C_2. The disturbance input is used as input, the actual output as output of the system. In these measurements, it has to be presupposed that the reference input remains constant during the investigation. The second way is a study of the open loop feedback system. A feedback system is opened, if it has been possible in some way to ensure that the value measured by the feedback transducer is no longer influenced by the actuator. This can be done by an interruption of the feedback loop between output variable and feedback transducer, as illustrated schematically in Figure 8.**7**. The input of the feedback transducer is then used as a new input x_{ol} of the open loop system. In this way, the problem has been reduced to the simpler task of the study of a system without feedback. In order to avoid confusion between the open loop feedback system and open loop control mentioned earlier, it is better to speak of the "opened" loop in the former case. However, as this is not usual in the literature, we also will speak of the open loop system where we mean the opened feedback system.

A feedback system can be opened in different ways. If it is possible, the simplest way is to open the feedback loop mechanically. This will be explained for the example already used in Figure 7.**1**. In the legs of many insects there is a feedback control system which controls the position of a leg joint, the joint connecting femur and tibia. The position of this joint is measured by a sense organ, the femoral chordotonal organ which is fixed at one end to the skeleton of the femur and at the other to the skeleton of the tibia by means of an apodeme, thus spanning the joint (Fig. 8.**6**). Because of this morphological arrangement, the more this sense organ is stretched, the more the femur-tibia joint is flexed. The reflex loop is closed such that by a dilatation of the sense organ-corresponding to a flexion of the joint-the extensor muscle is activated in order to oppose the flexion, whereas after a shortening of the sense organ the antagonistic flexor muscle becomes active. In this feedback system the loop can be opened mechanically by cutting the apodeme of the sense organ. The input function can then be applied by experimentally moving the apodeme and thereby stimulating the sense organ. The

activation of the muscles can no longer influence the value measured by the sense organ, thus the loop is opened. The two following examples show that eventually more elegant methods can also be used. For example, opening the pupil feedback system is very easy. The input of the feedback transducer is the illumination of the retina, subject to change by external illumination; the actual output is the pupil area. There is a difficulty when investigating the system in using a step function as input function, i. e., by suddenly switching on an additional light source. In this case the illumination of the retina is not changed in a step-like way. Due to the effect of the feedback system, the illumination is reduced again by the contraction of the pupil area. (By using an impulse function, this problem does not occur, provided the impulse is sufficiently short.) In the experiment, the problem can be avoided by using a light beam projected onto the retina which is narrower than the smallest possible pupil diameter (see Box 3). In this way, the experimentally generated illumination can no longer be influenced by the impact of the feedback system. The feedback system is thus opened.

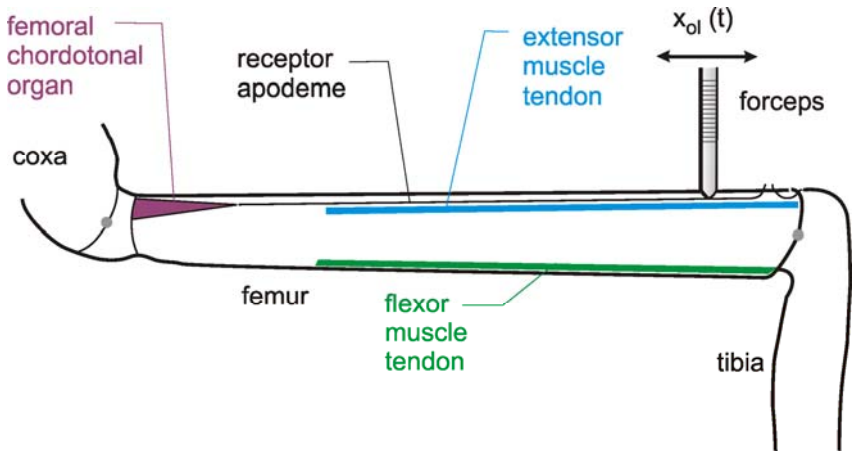


Fig. 8.**6 Morphological basis of a resistance reflex in the leg of an insect.** The angle between femur and tibia is monitored by the femoral chordotonal organ, which is mechanically connected to the tibia by a long receptor apodeme. By cutting this apodeme, the loop can be opened and the apodeme can be moved experimentally using a forceps and thus the x_{ol} input can be applied (Bässler 1983)

A further example of the opening of a feedback system is used in the study of system which controls the position of a leg joint (Bässler 1983). The joint is bent by the experimenter with great force, and the opposing force exerted by the system is measured at the same time. The bending force must be sufficiently great to prevent the counterforce developed by the feedback system from influencing the form of the bending to any measurable extent. In this example, too, the feedback system is thus opened. The advantage of both methods lies primarily in the fact that no surgical intervention with a potentially modifying effect are required, as is possible in the first example.

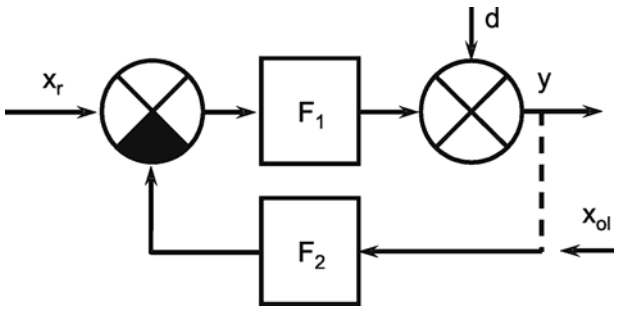


Fig. 8.**7 A negative feedback system as shown in Fig. 8.3, illustrating the input x_{ol} in the case of an opened loop.** x_r reference input, d disturbance input, y control variable. F_1 and F_2 represent the filter in the feedforward channel and the feedback channel, respectively

In the example shown in Figure 8.**7**, for the static case we obtain $y = - k_1k_2x_{ol}$, if k_1 and k_2 constitute the static amplification factors of the filters F_1 and F_2. This shows, though, that both methods, i. e., the study of the open feedback system (input x_{ol}, output y) and the study of the impact of the disturbance input on the closed loop (input d, output y), give information only on the product k_1k_2. The individual amplification factors of filters F_1 and F_2 can thus not be calculated separately. This means that even using both methods does not show the location where the reference input enters the system. Therefore, after these investigations, little or nothing can be said as to how the feedback system responds to a change in the reference input. This is only possible in such cases where the reference input can be manipulated experimentally. An example is pointing movements in humans. In this case, the subject can be shown where to point. If the subject is prepared to cooperate, arbitrary functions can be chosen for the reference input.

There are three possible ways to investigate a feedback controller, namely to experimentally change the reference input, the disturbance input, or, by opening the loop, by investigation of the opened loop response. Complete information about the system is only available if the first and at least one of the two latter methods can be applied.

8.5 Feedback Control Systems with Dynamic Properties

The problem mentioned in the last paragraph occurs independently of whether systems of the zeroth, first or higher orders are involved, that is, no further information can be obtained from the dynamic part of the responses. In the following, some examples will illustrate the behavior of all three controller types, mentioned above, namely the P-, D-, and 1-controllers. To this end, the x_r, d, and x_{ol} responses of the feedback system shown in Figure 8.**7** are compared, while different filters are substituted for F_1 and F_2.

According to equations (5) and (5'), we get the factors C_1 and C_2 according to $C_1 = k_1/(1 + k_1k_2)$ and $C_2 = 1 /(1 + k_1k_{2)}$. In the case of proportional elements, k_1 and k_2 denote the static amplification factors of filters F_1 and F_2. If a high-pass filter is present, the maximum dynamic amplification factor k_d (Chapter 5.3) has to replace the static amplification factor (which is always zero for a high-pass filter). The static amplification factor of a high-pass filter is used only in calculating the static opened loop response, i. e., using x_{ol} as input (Fig. 8.**7**). In the case of an integrator, the integration time constant τ_i has to be taken into account (see below). In the examples chosen here (Fig. 8.**8**), we assume $k_1 = 0.5$, and $k_2 = 1$.This means that $C_1 = 0.33$, and $C_2 = 0.66$. The integration time constant is assumed to be $\tau_i = 1$ for Figure 8.**8e** and $\tau_i = 0.5$ for Figure 8.**8f.** To show the dynamic properties of six different negative feedback system, the temporal change of the actual output y (see Figure 8.**7**) is presented, when a unit step function is given to the reference input x_r, the disturbance input d, or the input of the opened loop x_{ol},. The response to the disturbance input is shown by dotted lines.

a) Let filter F_1 be a 1st order low-pass filter with the time constant τ and the static amplification factor k_1, and filter F_2 be a constant (a zeroth order system) with amplification factor k_2. The step response of the open loop (input x_{ol}) is that of a low-pass filter endowed with the static amplification factor k_1k_2 and the time constant $\tau_{ol} = \tau$ (Fig. 8.**8a**, x_{ol}). If, in the case of a closed loop, the disturbance input d is used as input, a step response is obtained which first ascends to value 1, then, with the time constant $\tau_{cl} = C_2\tau$ (cl = closed loop), descends to the static value C_2 (Fig. 8.**8a**, dashed line). As C_2 is always smaller than 1 (see Chapter 8.3), the time constant of the closed loop is smaller than that of the open loop system. Here, the influence of the disturbance input is diminished from value 1 to value C_2 ($C_2 < 1$), but not to zero. It is typical of a P-controller (F_1 is a proportional term) that, in the case of a step-like disturbance input, a continuous error persists. The response of the closed loop system to a step-like

change in the reference input x_r corresponds to that of a low-pass filter endowed with the time constant $\tau_{cl} = C_2\,\tau$ and the static amplification factor C_1 (Fig. 8.**8a**, x_r). The time constant of the closed loop thus also depends on amplification factors k_1 and k_2.

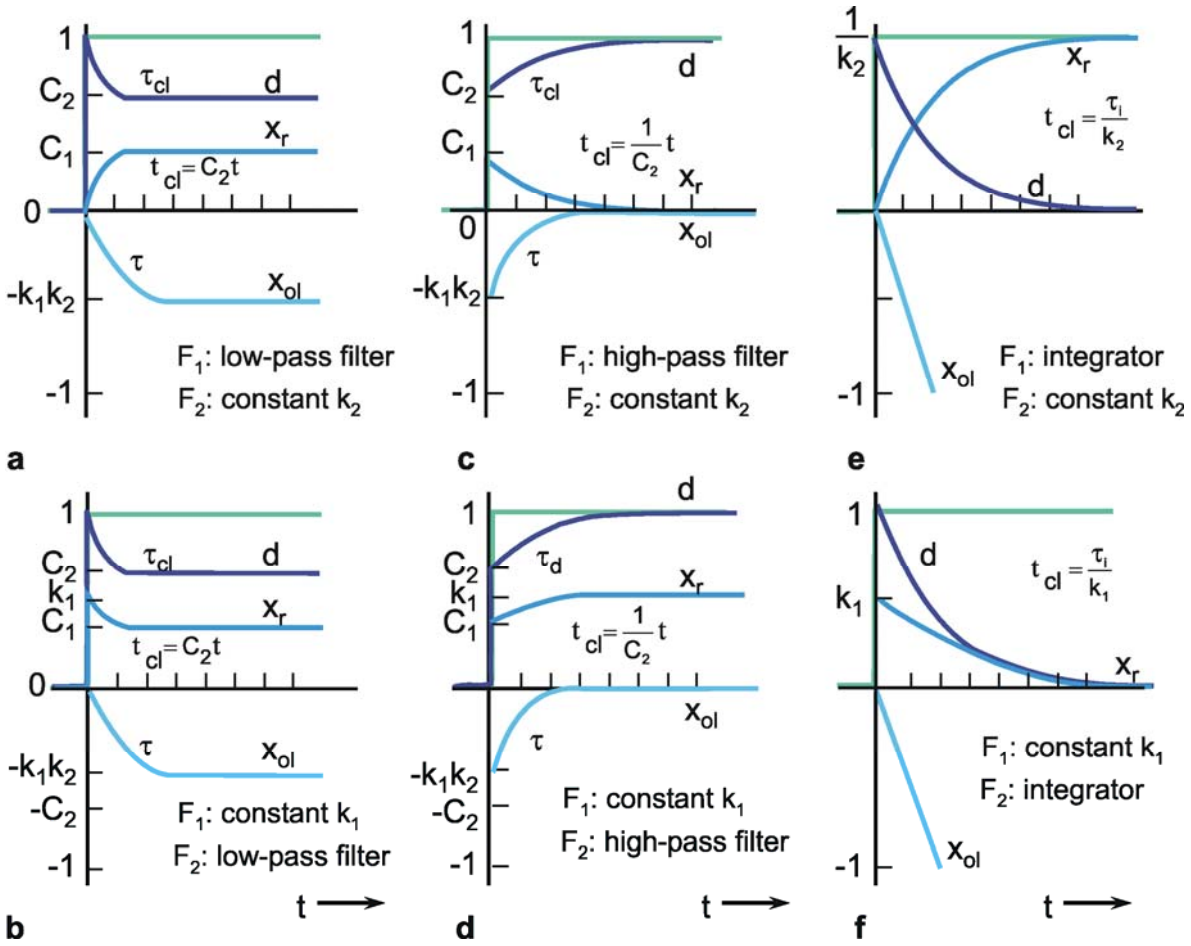


Fig. 8.**8 The step responses of the actual output y when a unit step (green lines) is given to the reference input x_r (blue lines), the disturbance input d (dark blue lines), or the input of the opened loop, x_{ol} (light blue lines).** Different combinations of filters F_1 and F_2 (see Fig. 8.7) are used (constants, low-pass filter, high-pass filter, and integrator, as mentioned). The meaning of the different parameters is explained in the text. The differences between the time constants of the open and the closed loop, τ and τ_{cl},, are graphically enlarged for the sake of clarity.

b) If the two filters are interchanged such that F_1 now represents a constant amplification factor k_1, and F_2 a low-pass filter endowed with the time constant τ and the static amplification factor k_2, we are still dealing with a P-controller. The response of the open loop system agrees with that of the previous example as, in linear systems, the sequence of the series connection is irrelevant (8.**8b,** x_{ol}). Also the response of the closed loop to a step-like disturbance input d exhibits the same course as in the first example (Fig. 8.**8b,** dotted line). Both feedback systems thus can not be distinguished in this respect. On the other hand, they respond differently to a step-wise change in the reference input x_r as the step response (Fig.8.**8b**, x_r) shows. This response initially jumps to value k_1 and then, with time constant $\tau_{cl} = C_2\,\tau$, descends to the value C_1. The occurrence of this response can be explained as follows: First, the step weighted with factor k_1 fully affects the output. Only then does an inhibiting influence occur via the feedback loop, whose time course is determined by the "inert" properties of the low-pass filter.

c) Figure 8.**8c** shows the case of a system with D-controller. Filter F_1 is a 1st order high-pass filter with time constant τ, the maximum dynamic amplification factor $k_d = k_1$, and the static amplification factor $k_s = 0$. Filter F_2 is formed by the constant k_2 (zeroth order system). The step response of the open loop is that of a high-pass filter of time constant τ and the maximum dynamic amplification factor $k_1\ k_2$ (Fig. 8.**8c,** x_{ol}). In studying the response to a step-like disturbance input in the case of a closed loop, we note that, immediately after the beginning of the step, the former decrease from value 1 to value C_2 (Fig. 8.**8c,** dotted line). In the course of time, though, the effect of the feedback diminishes. With the time constant $\tau_{cl} = \tau/C_2$ (i. e., $\tau_{cl} > 1$), the step response ascends to the final value 1. Is should be noted, however, that by using a "realistic high-pass filter," that is, a band-pass filter, the disturbance cannot be turned down immediately. Rather, the maximum feedback effect occurs only a little later, as it depends on the upper corner frequency of the band-pass filter. In principle, it can be said that the feedback effect of a system with D-controller occurs very quickly, but also that it descends to zero after some time, so that the disturbance input can then exert its influence without hindrance. A further property of the D-controller is mentioned in Chapter 8.6.

The response of the closed loop to a step-like change in the reference input x_r corresponds to that of a high-pass filter endowed with time constant $\tau_{cl} = \tau/C_2$ and the maximum dynamic amplification factor C_1 (Fig. 8.**8c**, x_r). With this feedback system, it is thus not possible to retain an actual output that is constant and derivates from zero over a prolonged period of time. If this process which takes place in the closed loop is simplified by mentally separating it into two successive simple processes, the result can be explained in qualitative terms as follows: The high-pass response to the input function is subtracted from the input function via the feedback loop. This new function, which comes close to the step response of a low-pass filter, now constitutes the new input function for high-pass filter F_1. On account of the prolonged ascent of this function, the response of the high-pass filter is also endowed with a greater time constant τ_{cl}.

d) If filters F_1 and F_2 are interchanged again such that F_1 be a constant k_1, F_2 a 1st order high-pass filter endowed with the time constant τ and the maximum dynamic amplification factor k_2, the same responses are obtained for the input quantities x_{ol} and d, as in example c) (Fig. 8.**8d**)**.** Again, after these investigations, it is not possible to make a statement concerning the behavior of the feedback system in response to changes in the reference input. In this case, the step response to a change in the reference input jumps initially to value C_1 and then, with the time constant $\tau_{cl} = \tau/C_2$, ascends to the final value k_1 (Fig. 8.**8d,** x_r).

e) In the next example, an integrator with the a time constant τ_i (see Chapter 4.8) is used as filter F_1, and a constant k_2 as filter F_2. The step response of the open loop is a line descending with the slope $-k_2/\tau_i$ (Fig. 8.**8e,** x_{ol}). The step response to a step-like disturbance input d initially jumps to 1, that is, the disturbance input fully affects the actual output. Subsequently, though, the latter decreases with the time constant $\tau_{cl} = \tau_i/k_2$ to zero, so that the effect of the disturbance input is fully compensated in the case of an I-controller (Fig. 8.**8e,** dotted line). This happens because the output value of the integrator increases as long as the error signal is different from zero. If the error signal, and hence the input value of the integrator, equals zero, the integrator retains the acquired output value. The response to a step-like change in the reference input corresponds to a step response of a low-pass filter endowed with time constant $\tau_{cl} = \tau_i/k_2$ and the static amplification factor $1/k_2$ (Fig. 8.**8e, x_r).**

f) In interchanging the two types of filters - F_1 is now a constant k_1 and F_2 an integrator of the integration time constant τ_i - one again obtains the same responses as in e) (Fig. 8. **8f**), apart from the difference between k_1 and k_2, for open loop systems and for the change in the disturbance input. If the reference input of the closed loop is changed, however, a completely different step response is

obtained. It corresponds to the step response of a high-pass filter of the time constant $\tau_{cl} = \tau_i/k_1$ and the maximum dynamic amplification factor k_1(Fig. 8.**8f**, x_r). Again, in this example as in that of c), no constant actual output values that are different from zero can be adopted either.

Examples e) and f) show an alternative way of constructing a low-pass and a high-pass filter, respectively, using a feedback system with integrator. These circuits provide the basis for the simulation of those filters shown in Appendix II.

In accordance with the definition given in Chapter 8.2, the feedback systems shown in Figure 8.**8d** and 8.**8f** are endowed with P-controllers, since in both cases filters F_1 are proportional terms. However, the rule that, with a P-controller at a constant disturbance input, a continuous error signal persists - which is smaller, though, than the effect of the disturbance input in itself - does not apply to these two cases. It only applies if no terms other than proportional ones occur in the feedback branch. For biological systems, this cannot be automatically assumed, however. The examples mentioned above should therefore also serve to illustrate other possibilities. A possible process (Fig. 8.**1**) was not considered here for reasons of clarity. An example taking the process into account will be described below.

It has already been pointed out that, in the case of a D-controller (Fig. 8.**8c**), the compensation of the effect of a disturbance input is much more rapid than with a P-controller (Fig. 8.**8 a, b).** The reason is that a P-controller is maximally effective only in the static range, whereas a D-controller operates maximally right from the start of the ascent of the disturbance input, since the value of the first derivation is highest at this stage. A major disadvantage of a D-controller is that the static effect is zero, i. e., that the disturbance input exerts its full influence in the static case. In technical systems, a parallel connection of P- and D-elements is frequently used to eliminate this disadvantage. The result is a P-D-controller. Compared to the P-, D-, and P-D-controllers, the I-controller is endowed with the property of changing its controlled variable until the error signal equals zero, that is, until the disturbance input is fully compensated. The disadvantage of the I-controller lies in the fact that it operates even more slowly than a P-controller. For this reason, P-I- or P-I-D-controllers, produced by the corresponding parallel connections, are used in technology, and probably in biology, too, thus combining the advantages of all types of controllers.

Although I-controllers appear more useful than P-controllers, since for the latter the disturbance input is only partially compensated, P-controllers are the rule in biological systems. How could this be explained? Actually, the use of a P-controller makes particular sense, if a specific variable is controlled by several, e. g., two, feedback systems rather than just one such system, as is frequently the case in biological systems. A familiar example is position control in fish. v. Holst showed that the position of the vertical body axis is controlled via visual pathways ("dorsal light response") as well as the otolith organs (v. Holst 1950 a, b). If the reference inputs of the two feedback systems do not absolutely agree, the control for outputs of both loops operate consistently against each other. When P-controllers are involved, both controller outputs take constant values which are proportional to the (usually small) deviation. In the case of I-controllers, however, the output values of the two controllers, which are operating against each other, would ascend to their maximally possible values. In this case, the use of I-controllers would obviously be uneconomical.

To summarize, we can say that, in principle, there is no problem in investigating the response of a biological feedback system to changes in the disturbance input. This is sufficient for a homeostatic feedback system. If a servomechanism is involved, however, the behavior of the feedback system in response to changes in the reference input would be of interest. As long as the reference input is not accessible, though, only very indirect conclusions can be drawn from these investigations or the study of the open loop feedback system.

When, however, we are interested in the transfer properties of the controller and of the process, for instance in the above mentioned example of bending the joint of a leg, the situation can be described by Figure 8.**9**. k, is the amplification factor of the controller and k_3 is that of the process, i. e., the elasticity of the muscles. k3 describes the transformation of force f produced by the muscles into position p, k, correspondingly the transformation from position (error signal) into force. Without loss of generality we can assume the reference input x, to be zero. Then, we have the equations $f = k_1k_2p_{ol}$ under open loop conditions and $p = k_3/(1 + k_1k_2k_3)\ d = k_{cl}\ d$ under closed loop conditions, i. e., when we load the leg and measure the position obtained under disturbances. Thus, k_1 and k_{cl} are measured and k_3 can then be calculated.

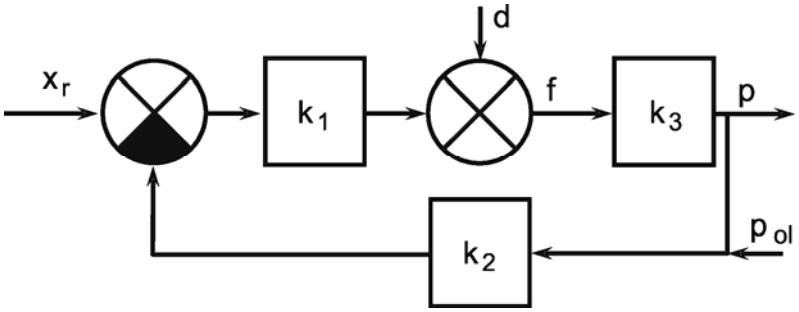


Fig. 8.**9 A negative feedback system for position control including a process (k_3).** k_1 and k_2 describe the gain of the controller and the feedback transducer, respectively, d disturbance input, f force developed by the controller (+ actuator which is not shown here) and the disturbance input, p position output, p_{ol} position input when the loop is experimentally opened

However, as a general comment it should be remembered that all these considerations, regarding comparison between open loop and closed loop investigation, hold only on condition that the system always remains the same, regardless of the kind of investigation. This is not always the case, as results of Schöner (1991), investigating the posture control in humans, and results of Heisenberg and Wolf (1988), considering the orientation response of *Drosophila,* have shown. The latter work shows an interesting, nonlinear solution of the control problem.

Several examples showing the dynamic properties of different feedback controllers are shown in Figure 8.**8**. In some cases, the behavior of the complete system corresponds to that of a low-pass filter or a high-pass filter. The properties of P-, I-, and D-controllers can be combined to obtain the advantages of the different elements. In biology I-controllers are less often used than P-controllers, because the former cannot be combined with other feedback system to control the same actual output value.

8.6 The Stability of Feedback Systems

Earlier we found that, on the basis of investigations of the opened feedback system, little can be said about the properties of the closed loop regarding its behavior in response to changes in the reference input. But there is one important property about which a statement can be made after the study of the open loop. This concerns the question of the stability of a feedback control system. Such a system is unstable, if the value of the output variable grows infinitely in response to an impulse- or step-like change in the input variable (reference input or disturbance input). Usually, the output variable of an unstable feedback system exhibits oscillations of increasing amplitude. In reality, however, the amplitude is always limited by an nonlinearity. In some cases, such oscillations may even be desirable (e.g., to generate a circadian rhythm). But in a feedback system constructed to keep a variable constant, such oscillations should be avoided.

The origin of these oscillations can be illustrated as follows: assume that any kind of input function is given to an input, for example the disturbance input. According to the Fourier decomposition, this input function can be considered to consist of the sum of several sine functions. When these sine functions pass the sign reversal at the comparator, this corresponds mathematically to a phase shift of 180°. Further phase shifts may be caused by the presence of a higher order low-pass filter or a pure time delay. If one takes account of that sine oscillation which, by passing the loop, receives a phase shift of 180° in this way, together with the effect of the sign reversal, an overall phase shift of 360° will result for this frequency. This means that the same sine function (except for an amplification factor) is added to the sine function of the input variable. Since this is repeated for each cycle, an oscillation results that keeps building up, and whose amplitude increases infinitely under certain conditions.

With the aid of the so-called Nyquist criterion, it is possible to decide, on the basis of the properties of the open loop system (i.e., the opened feedback system) whether the closed loop system is stable or unstable. The Nyquist criterion can be applied to the Bode plot of the open loop response as follows. (These data are applicable only if the impulse response of the open loop, that is, its weighting function, does not grow infinitely with increasing time. For the application of the Nyquist criterion in other cases, the reader is referred to the literature (DiStefano et al. 1967, Varju 1977)). First, the Bode plot of the open loop feedback system will be studied with a view to identifying critical situations. A total of eight different situations (S1-4, I1-4) are possible:

S1: The amplitude frequency plot crosses the amplification-1 -linie with negative slope.

The phase shift at this frequency comprises $0° > \varphi > -180°$.

S2: The amplitude frequency plot crosses the amplification- 1-line with positive slope. The phase shift at this frequency comprises $0° < \varphi < 180°$.

S3: The phase frequency plot crosses the -180°-line with positive slope. The amplification factor at this frequency is > 1.

S4: The phase frequency plot crosses the -180°-line with negative slope. The amplification factor at this frequency is < 1.

I1: The amplitude frequency plot crosses the amplification-1 -line with positive slope. The phase shift at this frequency comprises $0° \geq \varphi \geq -180°$.

I2: The amplitude frequency plot crosses the amplification- 1-line with negative slope. The phase shift at this frequency comprises $0° \leq \varphi \leq -180°$.

I3: The phase frequency plot crosses the -180°-line with negative slope. The amplification factor at this frequency is ≥ 1.

I4: The phase frequency plot crosses the -180°-line with positive slope. The amplification factor at this frequency is ≤ 1.

With the aid of these situations, the Bode plot of the open loop system is characterized in the following way. First, these critical situations have to be identified and noted in order of appearance beginning with the low frequencies. Figure 8.**10** shows the Bode plots of three different systems. All of them are endowed with the same amplitude frequency plot, i. e., they differ only in terms of their phase frequency plot. The following critical situations are obtained for these three Bode plots (1): S1, S4; (2): I3,I2; (3): I3, S3, S1, S4. If in this sequence, two situations with the same figure are placed next to

each other (as in (3) for I3, S3), both have to be canceled. If this result in two further situations with the same figure becoming neighbors, they, too, have to be eliminated. This procedure is continued until no further eliminations are possible, i. e., no situations with the same figure occur next to each other. Then only situations should occur belonging to either the S class or to the I class. In the first case (S), the closed system is stable, in the second case (I), it is unstable.

The application of the Nyquist criterion to the Bode plot of the open loop feedback system can be considerably simplified, if in the entire Bode plot no more than two of these critical situations are present, as is usually the case. (In Figure 8.**10,** this is true for systems (1) and (2).) For this purpose, we will introduce the concept of phase margin φ_m. If, at the frequency where the amplitude frequency plot crosses the amplification-1-line, the open loop system is endowed with the phase shift φ, then the phase margin can be calculated by $\varphi_m = 180° + \varphi$ (Fig. 8.**10** (1)). The phase margin thus represents the difference between -180° and the phase shift at the frequency considered. (φ_m is defined only between - 180° < φ_m < 180°.) With respect to this simple case, it could be said therefore that the closed system is stable, if $\varphi_m > 0$, and that it is unstable, if $\varphi_m < 0$.The frequency at which the open loop system shows a phase shift of -180° is known as critical frequency. If a feedback system is still in a stable range, but close to the limit of stability, this can be observed in the behavior of the system. The smaller the phase margin, the faster is the response to a step function, but the more prolonged, too, are the transient oscillations to the stationary value. The closer the system is to the limit of stability, the worse the "quality of stability" thus tends to become. In technical systems, the rule concerning a reasonable quality of stability states that the phase margin should be 30° to 60° at least. In the unstable case, the feedback system carries out oscillations with that frequency at which the phase frequency plot intersects with the - 180° line, i. e., the critical frequency.

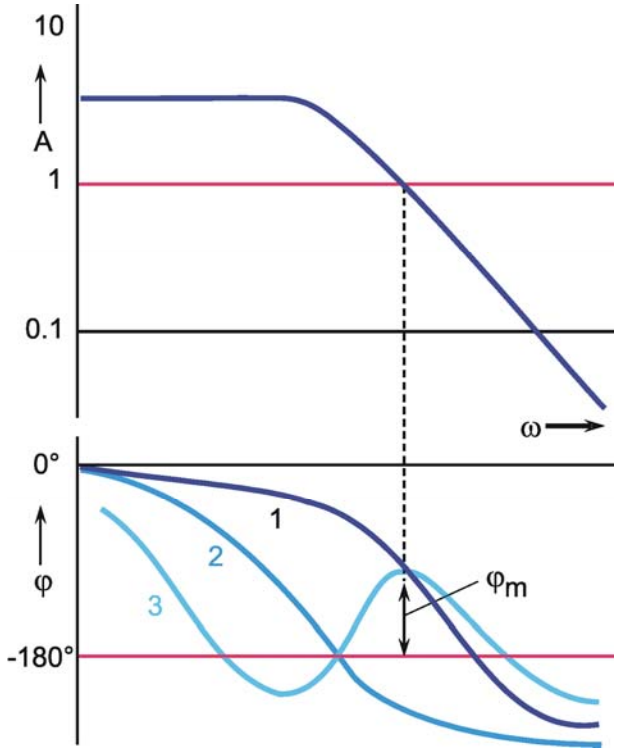


Fig. 8.**10 The Bode plots of the open loop system of three different feedback controllers.** The amplitude frequency plot is assumed to be the same in all three cases. The form of the different phase frequency plots (1-3) shows that the feedback controllers with the phase frequency plots (1, dark blue line) and (3, light blue line) are stable, whereas the plot no. (2, blue line) belongs to an unstable controller, because in this case the phase margin φ_m is negative

When we look at the open loop Bode plot of a stable system ((1) in Figure 8.**10**), we recognize that a stable system can be made unstable by increasing its amplification factors. Leaving the phase frequency plot unchanged, the intersection of the amplitude frequency plot with the amplification-1-line is shifted to higher frequencies. Another possibility would be to increase the phase shift, e. g., by increasing the dead time of the system, leaving the amplitude frequency plot unchanged. This phenomenon is observed in connection with the impact of alcohol on the central nervous system. Everyone is familiar with the resultant oscillations. Two methods for artificially raising the amplification factor for the pupil control system are described in Box 3. Thereby, the system can be stimulated to show continuous oscillations of the pupil diameter at a frequency of about 1-2 Hz.

It should be pointed out at this stage that any assumption concerning the inference of the closed loop from the open loop system is meaningful only, if the actual, (i. e., dimensionless) amplification factor of the open loop system can be given. For this, the input and output value has to be measured using the same dimension, or has at least to be converted into the same dimension. However, as was pointed out in Chapter 2.3, this is not always possible in the study of biological systems.

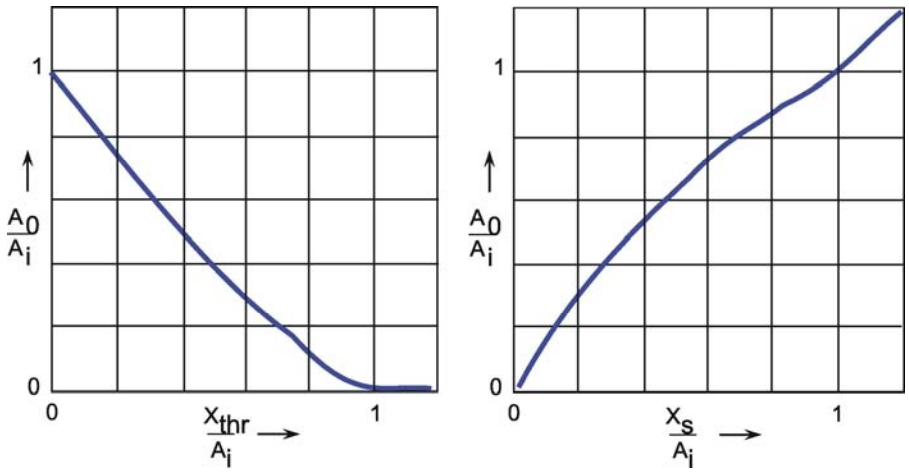


Fig. 8.**11 Gain factors for the fundamental wave for (a) a nonlinear characteristic with threshold value x_{thr} (see Fig. 5.10) and (b) a nonlinear characteristic with saturation value x_s (see Fig. 5.9).** A_1 input amplitude, A_o output amplitude (after Oppelt 1972)

Stability of nonlinear feedback systems. The Nyquist criterion is strictly applicable to linear systems only. The calculation of the static behavior of a feedback system endowed with nonlinear static characteristics has been dealt with in Chapter 8.3. But the question of whether such a feedback system is at all stable, that is, whether a static behavior exists at all, was not addressed. In the following, we will therefore answer the question of when such a nonlinear feedback system is unstable. An approximate solution to this problem is possible with the following assumption: The filters occurring in the total system, apart from the nonlinear static characteristic, are endowed with low-pass properties that have the effect that the harmonics generated in the sine response by the nonlinear characteristic are considerably weakened after having passed the closed loop. This leaves only the fundamental wave as an approximation to be considered. The relation of the maximum amplitude A_0 of the fundamental wave at the output to the maximum amplitude A_i of the sine oscillation at the input can be defined as the amplification factor of the nonlinear characteristic. This amplification factor depends both on the size of the input amplitude and on the position of the point of operation. In Figure 8.**11 a, b,** these amplification factors are given for the case of a characteristic with a threshold (Fig. 5.**10**) and a characteristic with saturation (Fig. 5.**9**) for different threshold *(x_{thr})* and saturation values (x_s) and various input amplitudes A_i. The slope of the linear parts of the characteristics is assumed to be 1, and the origin of coordinates always to be the point of operation.

If these amplification factors are introduced for the nonlinear static characteristic of the feedback system, the system can then be considered linear under the conditions mentioned earlier. The amplitude frequency plot of the opened loop is obtained by multiplying the amplitude frequency plot of the linear feedback system with this amplification factor. The Nyquist criterion may be applied to this. This shows that, with respect to a nonlinear feedback system, one can no longer speak of the system as such being stable or unstable. Rather, the stability of a nonlinear feedback system may depend on the range of operation involved. If nonlinear dynamic characteristics are present, the phase shifts produced by the nonlinearity may also be considered accordingly; but this will not be further discussed here (see e. g., Oppelt 1972).

Rather than oscillations of an infinitely large output amplitude, a feedback system that is unstable will, due to a characteristic with saturation, always produce only limited maximum amplitudes in a real situation. One example, discussed earlier, is the pupil control system. By an artificial increase in the amplification, this feedback system can be rendered unstable and it responds by continuous oscillations of a constant amplitude and frequency. Such oscillations are also known as *limit cycles*. Depending on the relative position of nonlinearity and low-pass elements in the loop, these limit cycles may be considerably distorted by the nonlinearity, as shown in Figure 5.**9**, or may produce nearly sine-shaped oscillations of small amplitude (see Box 3).

An example of a feedback system in which limit cycles almost always occur is the so-called two-point controller. This is a nonlinear controller which has the characteristic of a relay, i. e., there is an input value above which the control signal always takes position 1 (e. g., "on"), and below which it takes position 2 (e. g., "off") (Fig. 5.**12**). The temperature control of a refrigerator, for instance, operates according to this principle. If a critical temperature is exceeded, the cooling aggregate is switched on. If, after some time, the critical temperature falls below a certain threshold, the motor is switched off and the refrigerator warms up again. Its actual temperature thus oscillates regularly around a critical temperature. The relay characteristic may be considered formally as a characteristic with saturation and, on account of the vertical slope, with an infinitely high amplification. In view of the high amplification and the time delay present in the system, this is unstable. Due to the saturation, however, no infinitely increasing oscillations, but limit cycles of a constant amplitude are obtained. The presence of a hysteresis-like characteristic, e. g., a real relay (Fig. 5.**16**), influences the occurrence of limit cycles. The time required to get from one switch point of the relay to the next has an effect that is analogous to a dead time and thus to a time delay.

A system with positive feedback can be stable if the feedback channel contains a high-pass filter. In this case the whole system has the same properties as a system consisting of parallel connected high- and low-pass filters (Fig. 4.**8a**). In Appendix I such a system is used as an example to show how to calculate the dynamic properties.

A negative feedback system may become unstable under certain conditions. These are high gain factors and large phase shifts within the loop. A quantitative criterion to judge the stability of the system is given by the Nyquist criterion which is based on the Bode plot of the open loop system. It can also be applied to nonlinear systems in an approximation. Unstable nonlinear systems can show limit cycle oscillations.

Box 3

The Pupil Feedback System

It was already described in Chapter 8.4 that the feedback system controlling the pupil diameter can be opened experimentally. This is shown in Figure B3.**1** (Milsum 1966). The retina is illuminated by a light beam projected on to it that has a diameter smaller than the smallest pupil diameter. In this way change of pupil diameter does not influence the input function.

The responses the step and impulse input functions produced using this technique are shown in Figure B3.**1 a-d.** An increase in light intensity leads to a decrease of the pupil diameter. The two values of the pupil area A_1 and A_2 marked in the figures correspond to static response to low and high input intensity, respectively. The response to an increasing step (Fig. B3.**1 a**) corresponds qualitatively to that of the system shown in Figure 4.**7a**, namely a parallel connection of a low-pass filter and a band-pass filter with k > 1. However, reversal of the sign of the input functions does not produce a corresponding sign reversal of the output functions for both the step and the impulse response (Fig. B3.**1 c**). The comparison between the two impulse responses (Fig. B3.**1 b, d)** shows no corresponding sign reversal either. This means that the system contains nonlinear elements. How can these results be interpreted? The band-pass response is only found when the input function contains a part with increasing light intensity. Therefore, a simple hypothesis to explain these results is to assume a parallel connection of a low-pass filter and a band-pass filter as in Figure 4.**7a**, but with a rectifier behind the band-pass filter. Furthermore, there is a pure time delay which causes the dead time T mentioned in the figures.

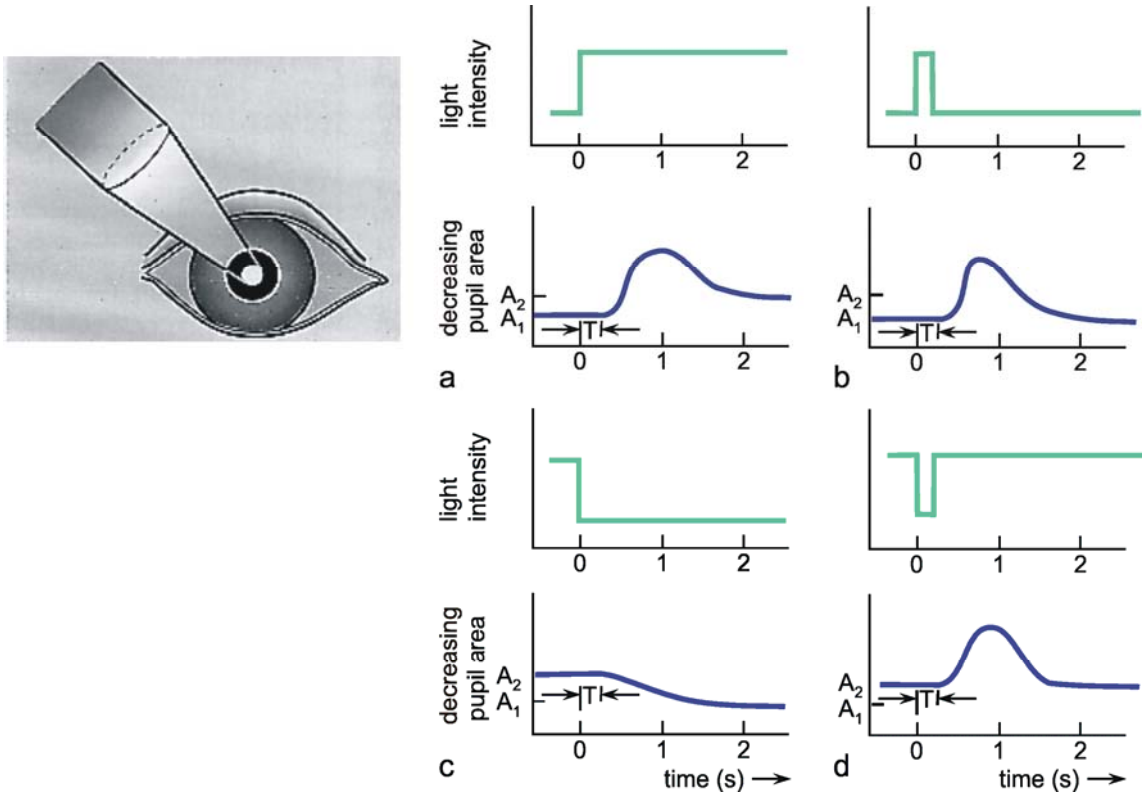


Fig. B 3.**1 Changes of the pupil diameter when the central part of the pupils is stimulated by a step function** (a), (c), or an impulse function (b), (d). T marks the dead time of the system

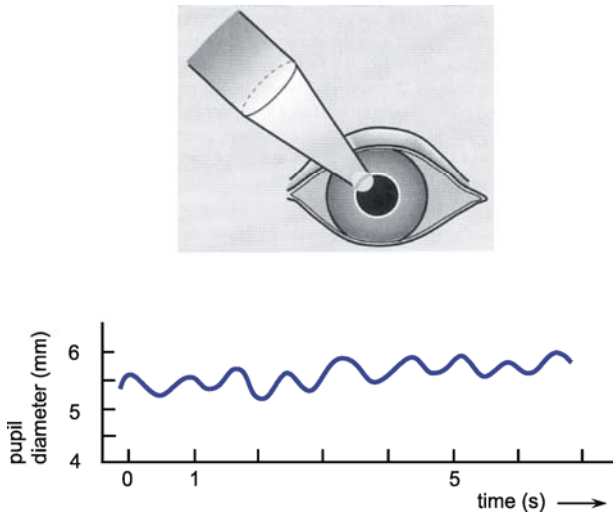


Fig. B3.**2 Limit cycle oscillations of the pupil diameter** after the gain of the control system has been increased artificially by illumination of the margin of the pupil (after Hassenstein 1966)

Figure B3.**2** shows a way of increasing the gain of the feedback loop (Hassenstein 1966). This can be done by projecting the light beam on the margin of the pupil such that the beam illuminates the retina with full intensity when the pupil is at its maximum diameter, and that the retina receives nearly to light when the pupil is at its smallest diameter. This leads to continuous oscillations at a frequency of about 1-2 Hz (Fig. B3.**2**). Another very simple and elegant method of raising the amplification factor has been proposed by Hassenstein (1971). The inner area of the pupil is obscured by a small piece of dark paper held in front of the eye. In this way, the change in the illumination of the retina accompanying a specific dilatation of the pupil diameter, and thus the amplification of the system, is increased. By using paper of an appropriate diameter and chosen according to given lighting conditions, oscillations can be elicited easily. As in humans both eyes are controlled by a common system, the oscillations can be observed by looking at the other, uncovered eye.

8.7 Further Types of Feedback Control Systems

In addition to the simple feedback system described above, there is a number of more complicated feedback systems, two examples of which will be discussed to conclude this topic. Figure 8.**12** shows a feedback system with feedforward connection of the reference input. The figure shows the two ways in which the reference input operates: it has a controlling effect on the output variable via the comparator, and it directly influences the output variable via a further channel in the form of an open loop (feedforward) control. If constants k_1 and k_2 are introduced for filters F_1 and F_2 to calculate the stationary case, the following equations will result, which is analogous to equation (5) in Chapter 8.3:

$$y = \frac{1+k_1}{1+k_1k_2}x_r + \frac{1}{1+k_1k_2}d$$

Whereas, in a simple feedback system, the reference input has to take the value

$$x_r = \frac{1+k_1k_2}{k_1}y$$

to obtain a particular value y as a output quantity, here is only needs to take the smaller value

$$x_r = \frac{1 + k_1 k_2}{1 + k_1} y$$

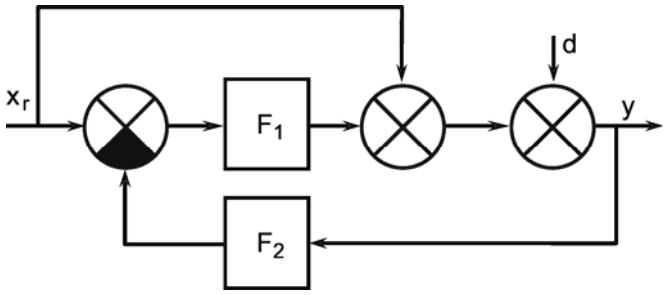


Fig. 8.**12 Control system with feedforward connection of reference input.** $x_r(t)$ reference input, d(t) disturbance input, y(t) control variable, F_1 and F_2 represent the filter in the feedforward channel and the feedback channel, respectively

The range of input values which the controller has to be able to process is thus smaller in a control system with feedforward of reference input, making it much simpler to construct the controller. This is an interesting aspect especially with regard to systems in neurophysiology, since here the range of operation of a neuron is limited to less than three orders of magnitude (about 1 Hz to 500 Hz of spike frequency). A further advantage of a feedback system with feed-forward of reference input lies in the fact that this can be constructed using a D-controller without the disadvantages described in Chapter 8.5. On account of the feedforward channel, the output variable receives a sensible input value, even if the output of the D-controller has fallen to zero. One advantage of this circuit lies in the response of the D-controller which is faster than that of the P-controller. Another aspect of this property of the D-controller, which was not discussed in Chapter 8.5, is that a system that is unstable (due the presence of a dead time, for example) can be made stable by the use of a P-D-controller instead of a simple P-controller. This can be explained by comparing this with Figure 8.**10** (2), namely that the phase frequency plot of the open loop is raised by the introduction of a high-pass filter (D-controller), especially at high frequencies, resulting in the fact that the phase frequency plot cuts the -180° line only at higher frequencies (see also Chapter 4.6). If the amplification of the high-pass filter is sufficiently small, the total amplification of the open loop may be smaller than 1 at this frequency. The system thus fulfills the conditions of stability.

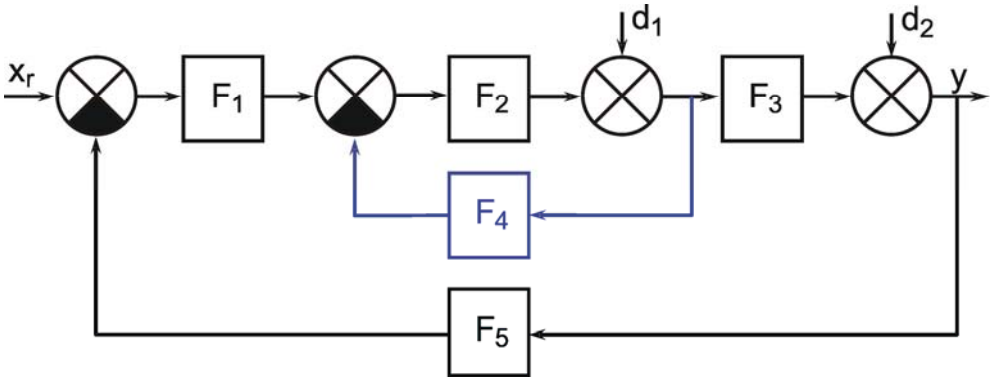


Fig. 8.**13 A cascaded feedback system.** $X_r(t)$ reference input, $d_1(t)$ and $d_2(t)$ disturbance inputs, y(t) control variable, F_1 to F_5 represent unspecified filters

If the structure of the controlled process is more complex, it is sometimes practical to control the value of a variable which represents a precursor of the actual output via a separate feedback system (Fig. 8.**13).** This is known as a cascaded feedback system. This circuit has the advantage that a disturbance at d, does not have to pass the entire loop (changing the actual output, passing the feedback loop), but is instead controlled beforehand so that the disturbance does not get through to the actual output (for a detailed study of cascaded feedback systems, see DiStefano et al. 1967). The cascaded system is a special case of the so-called mashed systems. We speak of a mashed system, if there are points in the circuit from which one can go through the circuit by at least two different routes and return to the same point.

A negative feedback loop with additional feedforward of reference input makes it possible to use a controller with a smaller gain factor. Cascaded feedback controllers can cope better with internal disturbances than can a simple uncascaded control system.

8.8 An Example

In order to illustrate the properties of feedback systems previously discussed, we will give a highly simplified description of the biological feedback system that controls the position of a joint, for example the elbow, in mammals. The description given will be on a qualitative level, to facilitate comprehension of the subject, and because there is still a number of unanswered questions. For a more detailed study of the underlying structures, the reader is referred to the literature (e. g., Kandel et al. 1991). To simplify matters, all antagonistically operating elements, such as muscles, nerves and the various receptors, are subsumed in one unit.

Several mechanisms are involved in the closed-loop control of the position of a joint, the most important being shown (in partly hypothetical form) in Figure 8.**14.** The anatomical relationships are represented in Figure 8.**14a** in highly schematic form. Figure 8.**14b** shows the same system in the form of a circuit diagram. The musculature (actuator) responsible for the position of a joint is innervated by the so-called α-motorneurons. The muscles controlled by the α-fibers produce a force which can shorten the muscle and, in this way, effect a change in the position of the joint. To this end, the mechanical properties (inertia of the joint moved, friction in the joint, elasticity and inertia of the muscles), which represent the process, have to be overcome. The muscle spindles lie parallel with this (the so-called extrafusal) musculature. They consist of a muscular part (the intrafusal musculature), whose state of contraction is controlled by the γ-fibers, and a central, elastic part which contains the muscle spindle receptors. These measure the tension within the muscle spindle. For a specific value of γ-excitation, this tension is proportional to the length of a given muscle spindle and thus to the length of the entire muscle. The response of the muscle spindles to a step-like expansion is of a phasic-tonic kind (Fig. 4.**11**), i. e., it might, as a first approximation, correspond to a lead-lag system, or, in other words, a parallel connection of a high-pass and a low-pass filter. If a disturbance is introduced into a muscle once its length is fixed (i. e., a particular position of the joint), as symbolized in Figure 8.**14a** by attaching a weight and in Figure 8.**14b** by the disturbance input d_1, the immediate result is a lengthening of the muscle. Due to the lengthening of the muscle spindle, the Ia-fiber stimulates the α-motorneuron, effecting a contraction of the muscle. The disturbance is counteracted so that we can speak of a feedback system.

Since we observe only stimulating synapses in this system, one obvious question concerns the point at which a reversal of signs occur, which is necessary in a feedback controller. The reversal is tied up with the mechanical arrangement of the muscle spindles, because shortening the muscle also shortens the spindle, which causes a decrease in activity of the sensory fibers. Therefore, the

frequency in the la-fiber is reduced when there is an increase in the frequency of the a-motorneuron, in spite of the fact that we have stimulating synapses only.

The γ-fibers also receive inputs from the brain. If the frequency of the γ-fibers is increased via these inputs, the intrafusal musculature of the muscle spindle is shortened. This causes an increase in the frequency of the la-fiber and, because of this, of the α-motorneuron, too. In this way, the length of the muscle (and thus the position of the joint) can be controlled via the γ-motorneuron. The γ-fibers, can therefore be regarded as the transmitters of the reference input. As early as around 1925 to 1927, Wagner interpreted this system as a servomechanism (Wagner 1960). It seems his work then fell into oblivion, and this interpretation surfaced again only in 1953 (Stein 1974).

For some time it has been known that α-motorneurons also receive inputs directly from the brain. Recent studies have shown that, during active voluntary movements, the fibers that activate the α and γ-motorneurons undergo parallel stimulation (Stein 1974) so that for the interrelated fibers, a joint input located in the brain must be postulated. In the physiological literature, a number of different terms have been put forward for this "α–γ-linkage". In terms of systems theory, this system could be described as a feedback system with feedforward connection of reference input (Chapter 8.7, Fig. 8.**12**). This assumption is supported by a comparison of movements actually carried out by test subjects with those simulated on the assumption of various hypotheses (Dijkstra and Denier van der Gon 1973).

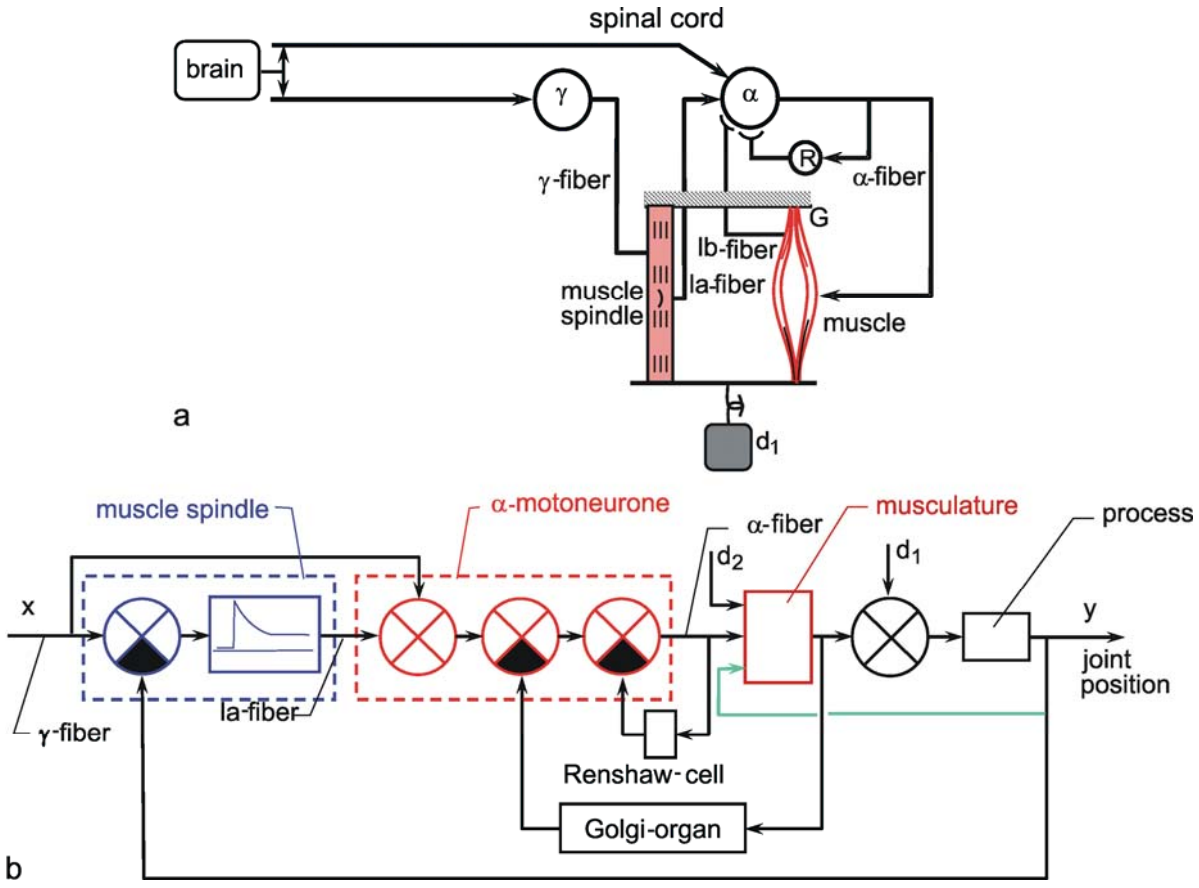


Fig. 8.**14 The mammalian system for the control of joint position.** (a) simplified illustration of the anatomical situation. (b) The corresponding model circuit. G Golgi organs, R Renshaw cells, x_r reference input, d_1 and d_2 disturbance inputs, y control variable (joint position)

In this feedback system, muscle spindles constitute feedback transducers, comparators, and controllers at the same time. As can be seen from the kind of step response (Fig. 4.**11**), we have a P-D controller. Its significance could lie in the fact that, in this way, the system can be prevented from becoming unstable, which might otherwise occur due to the mechanical properties of the process (the system is at least of 2nd order since elasticity and inertia are present) and additional delay times due to the limited speed of conduction of the nerves (see Chapter 8.6).

Additional sensors exist in this system in the form of the Golgi organs (Fig. 8.**14a,** G). They are located in the muscle tendons and measure the tension occurring there. We know that, when tension is increased via some interneurons, their effect on the α-motorneurons is inhibitory, but excitatory when the tension is lowered (by antagonistic action). It appears we are dealing with a feedback system designed to keep the tension of the muscles at a constant level. This could be interpreted as a second cascade which compensates disturbance inputs, such as fatigue of the musculature and other properties of the muscles, which affect the direct translation of the spike frequencies of α-fibers into the generation of force. Since, most probably, this influence cannot be described by simple addition, the disturbance input d_2 is shown as a second input of the system 'musculature' in Figure 8.**14b.** The third input in this system takes into account the fact that the force generated by the muscle also depends on its length.

A further cascade of this type is presented by the Renshaw cells (Fig. 8.**14a,** R) located in the spinal cord, which provide recurrent inhibition to an α-motorneuron. However, we do not want to go into detail concerning this and other elements involved in this feedback system, such as the position receptors in the joints or the tension receptors present in the skin, because little is known so far about the way in which they operate. (One possible explanation of the function of Renshaw cells is found in Adam et al. 1978).

The mammalian system to control joint position can be taken as an example that contains several interesting properties of feedback systems. It can be considered as a PD-controller with feedforward of reference input, and is built of several cascades.

8.9 Reafference Principle

Resistance reflexes as described above, where the reflex opposes the movement of a joint, were often discussed in terms excluding the possibility that active movements can be performed because the resistance reflex would immediately counteract such a movement. Thus, it was assumed that resistance reflexes have to be switched off in order to allow for an active movement. However, this is not the only possibility. We have seen that a feedback control system also solves this problem. It provides a resistance reflex when no active movement is intended, i. e., when the reference value is constant, and it performs active movements when the value of the reference input is changed. Another related problem occurs when a sensory input is influenced not only by changes in the outer world, but also by actions of the animal. How can the system decide on the real cause of the changes in the sensory input? One solution to the so-called reafference principle was proposed by v. Holst and Mittelstaedt 1950 (see also Sperry 1950). This will be explained using the detection of the direction of a visual object as an example. Figure 8.**15a** shows a top view of the human eye in a spatially fixed coordinate system. The object is symbolized by the cross. Thus, the absolute position of the cross is given by angle γ. The gaze direction of the eye is given by angle α. The position of the object on the retina, which is the system's only available information, is given by angle β. Angle β changes when the object moves, but also when only the eye is rotated. How can the real angle γ be detected? One

possibility is shown in Figure 8.**15b.** The position α of the eye is controlled by a feedback control system, as shown in the lower part of the figure and represented by the reference input x_α and the transfer elements k_1 and k_2. A second, vision system detects the angle β, the position of the object in a retina-fixed coordinate system. This is shown in the upper part of the figure. The angle β depends on α and γ, due to the geometrical situation according to $\gamma = \alpha + \beta$, or $\beta = \gamma - \alpha$ (Fig. 8.**15a).** This is shown by dashed lines in Figure 8.**15 b.** How can γ be obtained when the vision system has only β at its disposal? Possible solutions are to use the value α given by the sensory input in the form of $k_2\ \alpha$, or to use the efference value $k_3\ x_\alpha$ of the feedback loop. The first case, not shown in Figure 8.**15**, is simple but can be excluded by experimental evidence. As an alternative, v. Holst and Mittelstaedt proposed to use the copy of the efferent signal x_α the "efference copy" instead. According to equation (5) in Chapter 8.3, we obtain $\alpha = k_1 /(1 + k_1k_2)\ x_\alpha$. However, in general the vision system does not directly know β but $k_4\beta$, nor directly x_α but $k_3\ x_\alpha$. Thus, γ can be calculated by $\gamma = k_4\ \beta + k_3\ k_1 / (1 + k_1k_2)\ x_\alpha$ only if k_3 and k_4 fulfill particular conditions, a simple case being $k_4 = k_3 = 1$. (For investigation of the dynamics of this and alternative systems, see Varju 1990). Thus, with appropriate tuning of the transfer elements k_3 and k_4, the efference copy, or as it is often called in neurophysiology, the corollary discharge, can solve the problem of distinguishing between sensory signals elicited by the system's own actions and those produced by changes in the environment.

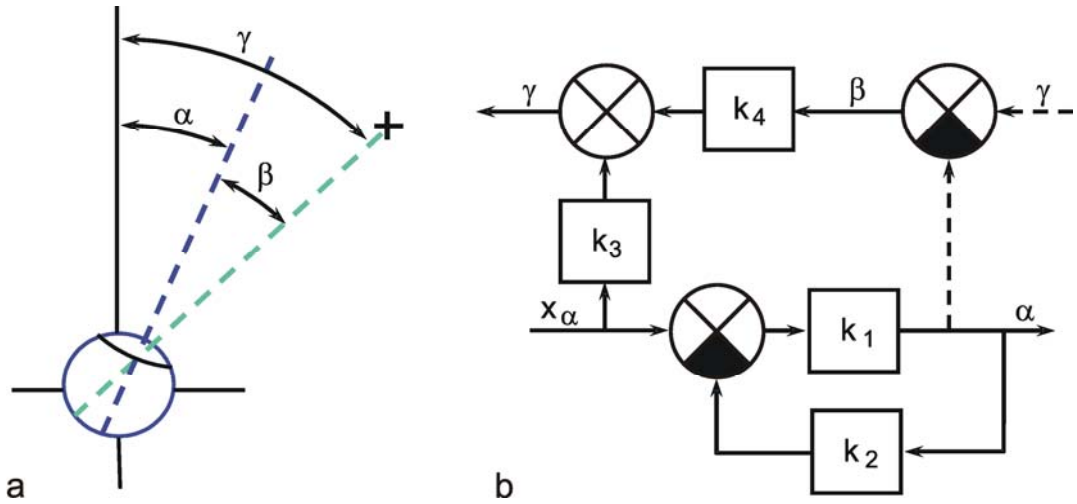


Fig. 8.**15 The reafference principle used to visually detect the position of an object, although the eye is movable.** (a) The position of the eye within a head-fixed coordinate system is given by α. The position of the object (cross) within an eye-fixed coordinate system is given by β, and within the head-fixed coordinate system by γ. (b) The lower part shows the negative feedback system for the control of eye position (x_a reference input, α control variable). The subtraction ($\gamma - \alpha$) shown in the right upper part represents the physical properties of the situation. The efference copy signal (passing k_3) and the measured value of β are used to calculate γ

A servocontroller can deliberately change the control variable and, at the same time, counteract disturbances. A system working according to the reafference principle, using the efference copy, can distinguish whether a change in sensory input was caused by an motor action of the system itself, or by a change in the environment.

8.10 The Dynamic Properties of a Simplified Neuron

The dynamic behavior of a neuron can be approximated by a relatively simple nonlinear system when the fact that the output of a typical neuron consists of a sequence of action potentials is ignored. If the frequency of these action potentials is not too low, the output value can be regarded as an analog variable. The behavior of such a simplified neuron can be approximated by the differential equation $dy/dt = x - f(y)$, where x is the input excitation and y is the output excitation (= spike frequency) (Kohonen 1988). f(y) is a static nonlinear characteristic the effect of which will be described below. A differential equation can easily be solved numerically by simulating the system on an analog or digital computer, when the equation is transformed into a circuit diagram. How can we find the circuit diagram that corresponds to this differential equation?

Since we have not considered in detail filters having the property of an exact mathematical differentiation, but have treated the much simpler integrators, it is better first to integrate this equation so as to obtain: $y = \int (x - f(y))\, dt$. We can now directly plot the corresponding circuit diagram. The characteristic f(y) obtains the output value y. The value f(y) is subtracted from the input x of the system. The resulting difference x - f(y) has to be inserted into an integrator, the output of which provides y. This is shown by the solid arrows in Figure 8.**16a.** As the output of the integrator and the input of the nonlinear characteristic f(y) comprise the same value, the circuit can be closed (dashed line in Figure 8.**16a**)**.** Thus, we obtain a system which formally corresponds to a feedback controller. The system might also be considered as a nonlinear "leaky integrator," with f(y) describing the loss of excitation in the system. f(y) should be chosen such that the loss increases overproportionately with the excitation y of the system. This has the effect that the output cannot exceed a definite value. Thus, the system has an implicit saturation characteristic. To make the simulation of a neuron more realistic, one should further introduce a rectifier behind the integrator (not shown in Figure 8.**16a**) because negative excitation is not possible. Figure 8.**16b** shows different step responses of this system. In this case, the time constant is the smaller, the greater the output amplitude. Figure 8.**16c** shows that the sine response of such a system can be asymmetric, as was already found in earlier examples.

As mentioned above, this simulation will produce a sensible replication of the behavior of a neuron only if input excitations above a given value are considered. To make the model also applicable to smaller input values, it could be improved by the introduction of a threshold characteristic at the input.

This model could be used as a simple description of the dynamic properties of a nonspiking neuron, but only as a very rough approximation to the properties of a spiking neuron. In a spiking neuron, information transfer is not continuous. Therefore, a circuit consisting of several spiking neurons can show a behavior which may be unexpected at first sight (Koch et al. 1989). For the investigation of the properties of such circuits, the simulation of a spiking neuron is of interest. Using the simulation procedure given in Appendix II, this can be done simply if it is sufficient to describe the spike by an impulse function of duration of 1 time unit (corresponding to about 1 ms in real time). Figure 8.**17** shows the diagram simulating a simple version of a spiking neuron. The first low-pass filter describes the time constant of a dendrite. The Heaviside function represents the threshold which has to be exceeded to produce a spike, and the low-pass filter in the feedback loop represents the refractory effect of a spike being elicited. Figure 8.**17b** shows an example of how this system reacts to different step-like input functions. As the system contains an implicit nonlinear property, an additional nonlinear characteristic at the input might be necessary to receive the behaviour of a specific neuron.

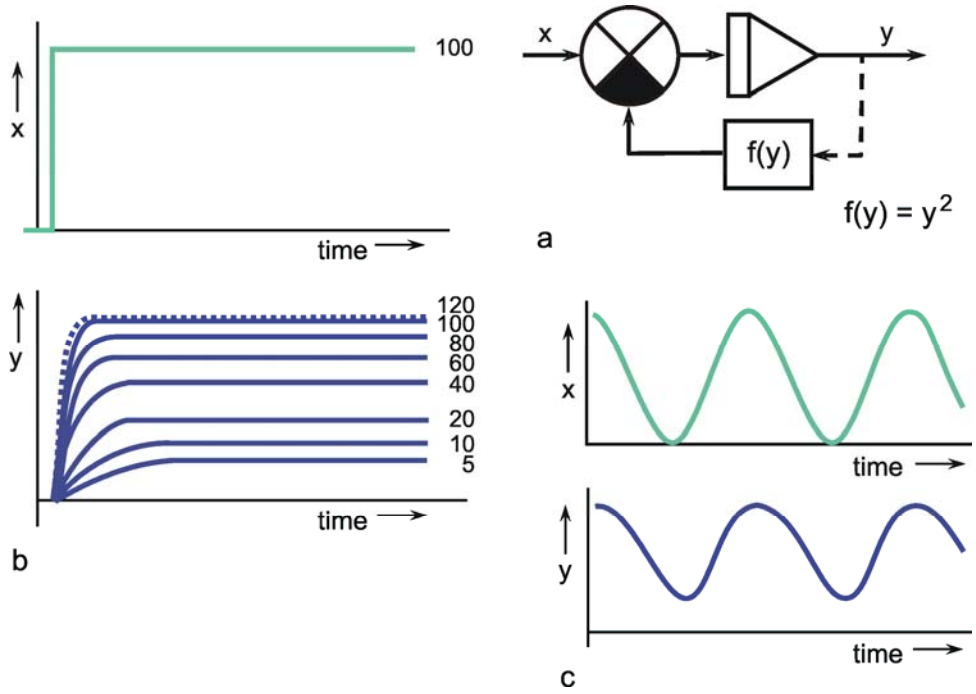


Fig. 8.**16 The approximation of the dynamic properties of a neuron.** (a) The circuit diagram representing the equation $y = \int (x - f(y))\, dt$. (b) Step responses of this system for input steps of different amplitudes. Only one input function (amplitude 100 units) is shown. The responses are marked by the amplitudes of their input functions. The system shows a saturation-like nonlinear behavior. (c) The response to a sine wave. Upper trace: input, lower trace: output. Note that the output function is asymmetrical with respect to a vertical axis

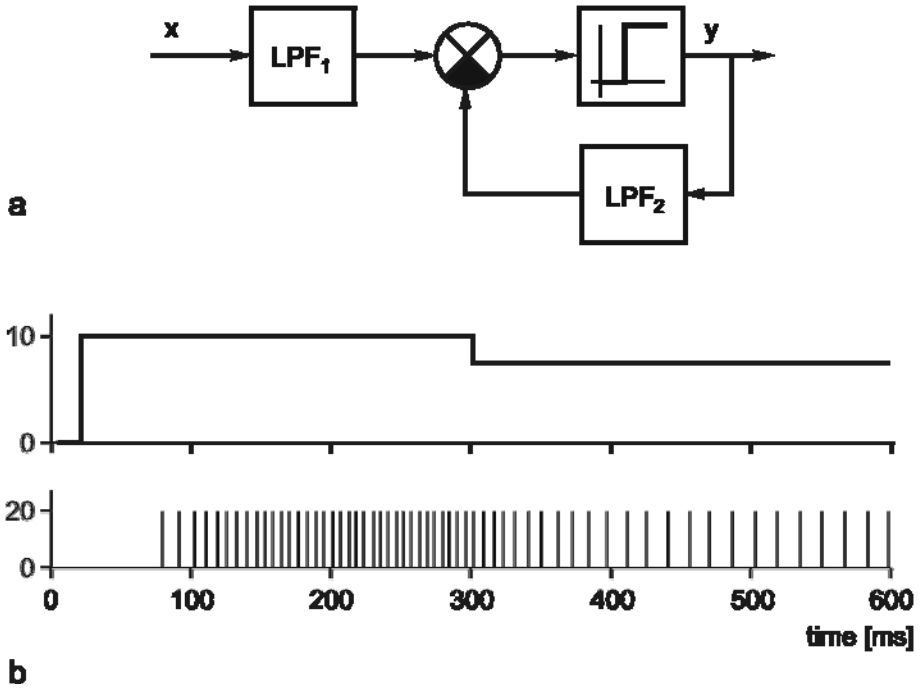


Fig. 8.**17 A simple system for the simulation of a spiking neuron.** (a) LPF_1, and LPF_2 represent two low-pass filters. As soon as the threshold of the Heaviside function is overcome, a spike is elicited. (b) Response of such a system to two step-like input functions (first step: from 0 to 10, second step: from 10 to 7.5). Heaviside function: 0 if input < 7, 20 else. Time constants: τ_1 = 50 ms, τ_2 = 10 ms

As an exercise for solving differential equations as explained above, the reader may like to try to solve the linear differential equations describing a 1st order low-pass filter [dy/dt = (x - y)/τ or y = 1 /τ ∫ (x - y) dt] or a 1st order high-pass filter [d(y-x)/dt = - y/τ or y = x - 1/τ ∫ y dt].

Linear and nonlinear differential equations can be solved by treating them as feedback systems. Two examples are shown which approximate the dynamic properties of nonspiking and of spiking neurons.

9 Massively Parallel Systems

Up to now, using the systems theory approach, we have dealt with systems where the number of parallel channels is very small. Most systems possess one or two parallel channels. This highest value, i. e., four, occurred in an example illustrated above (Fig. 8.**14**), if the feedback channels running in the opposite direction are included. Many systems, especially nervous systems, are endowed with structures containing an abundance of parallel channels. This is particularly clear for the above example where typically each muscle is excited by a large number of α- and of γ-motorneurones on the efferent side and by many sensory neurons (e. g., muscle spindles), on the afferent side. In previous considerations, models were constructed on the assumption that it is useful to simplify the systems by condensing a great number of channels to just a few. However, some characteristics of these systems become apparent only when the highly parallel or, as they are often called, massively parallel structure is actually considered. More recently such systems have been called neuronal nets or connectionist systems, but they had already been investigated in the early days of cybernetics (Reichardt and Mac Ginitie 1962, Rosenblatt 1958, Steinbuch 1961, Varju 1962,1965).

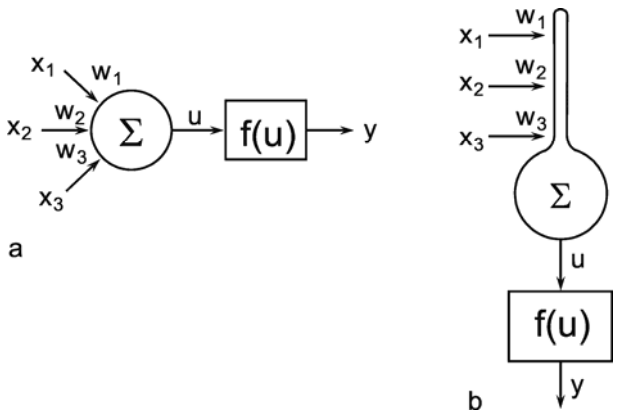


Fig. 9.**1 Two schematic presentations of a neuroid.** In this example the neuroid receives three input values x1, x2, x3. Each input is multiplied by its weight w1, w2, and w3, respectively. The linear sum of these products, u, is given to an, often nonlinear, activation function f(u) which provides the output value y. (a) resembles more the morphology of an biological neuron, (b) permits an clearer illustration of the connections and the weights

In order to make it possible to deal with a large number of parallel connections, we introduce a new basic element which, in the present context, we will call a neuroid, to show its difference from real neurons. The neuroid is a purely static element. Effects such as dynamic effects, for example low-pass properties (Chapter 6.4, 8.10) as well as more complicated properties such as spatial summation, or spatial or temporal potentiation, are not considered in this simple form of a neuroid nor are spiking

neurons considered (see however Chapter 8.10). Artificial neuronal nets consist of a combination of many of these basic elements. The neuroid displays the following basic structures (Fig. 9.**1 a, b).** It has an input section which receives signals from an unspecified number of input channels and summates them linearly. In principle, this input section may correspond to the dendritic section of a classic neuron, though it should be noted that there are important differences, as mentioned above. The input section of a neuroid acts as a simple summation element. It is represented in Figure 9.**1** in two different ways. In Figure 9.**1 a** it corresponds to the usual symbol for the summation. In Figure 9.**1 b** the input section is represented by a broad "column." This has advantages for subsequent figures, but should not encourage the (unintended) interpretation that this represents a real dendritic tree with the dynamic properties mentioned above.The inputs impact on the neuroid via "synapses." At these synapses the incoming signals may be subject to different amplification factors or weights which are referred to by the letter w. Generally, these weights may adopt arbitrary real values. Positive weights may formally correspond to an excitatory synapse, a negative weight to an inhibitory synapse. After multiplication with their corresponding weights w_1, w_2, and w_3, the incoming signals x_1, x_2, and x_3 are summated as follows:

$$u = \sum_{i=1}^{3} w_i x_i .$$

This value is eventually transformed by a static characteristic of a linear or, in most cases, nonlinear form which is often called the activation function. Two kinds of nonlinear characteristics are normally used in this context, namely the Heaviside (all-or-nothing) characteristic (Fig. 5.**12**) and the sigmoid characteristic. The logistic function already mentioned $1/(1 + e^{-ax})$ is frequently used as a sigmoid characteristic (Fig. 5.**11).** If the activation function is referred to by f(u), the resultant output value of the neuroid is $y = f(\Sigma\, x_i w_i)$.

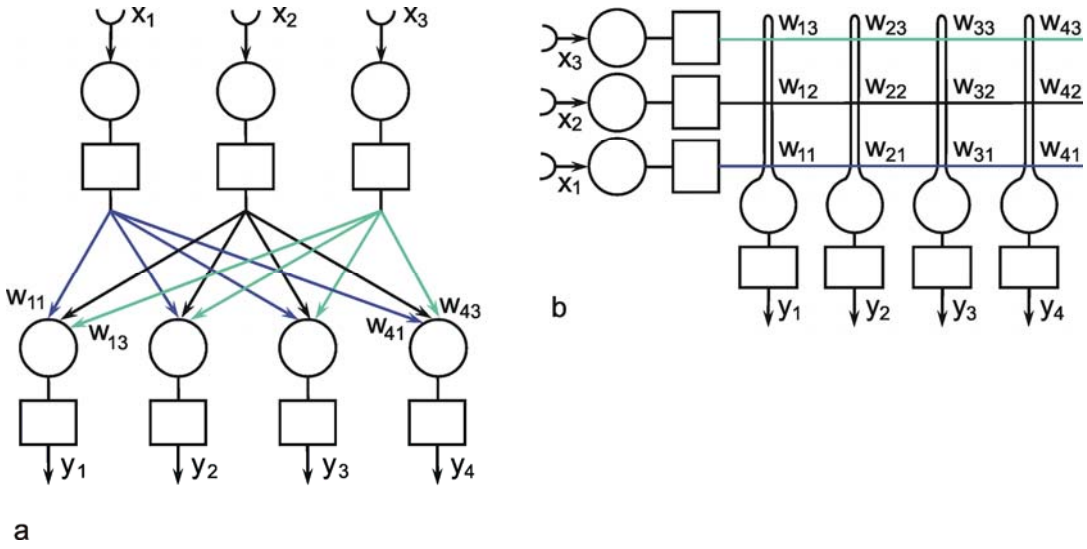


Fig. 9.**2 A two-layer feedforward network with n (= 3) units in the input layer and m (= 4) output units in the output layer.** x_i represent the input values, y_i the output values. The weights w_{ij} have two indices, the first corresponds to the output unit, the second to the input unit. As in Fig. 9.**1**, (a) more resembles the arrangement of a net of biological neurons, whereas the matrix arrangement of (b) permits a clearer illustration of the connections and the weights (only some weights are shown in (a)). The boxes represent unspecified activation functions

The neuroids can be linked together in different ways. The so-called feedforward connection is the one that can be most easily understood. The neuroids are grouped on two levels or layers, the input layer and the output layer (Fig. 9.**2a**). Each of the n neuroids of the input layer has only one input channel x_i which receives signals from the exterior world. The outputs of the neuroids of the first layer may be connected with the inputs of all m neuroids of the second layer. In this way, one obtains mn connections. Although the arrangement of Figure 9.**2a** does correspond more to biological structure, the matrix arrangement (Fig. 9.**2b**) is useful in providing an easily understood representation. Here the individual weights can be written as a matrix in a corresponding way. The connection between the j-th neuroid of the first layer and the i-th neuroid of the second is assigned the weight w_{ij}. The output of the i-th neuroid of layer two can thus be calculated as $y_i = f_i(_j\Sigma\ w_{ij}\ x_j)$. Using the feedforward connection, an unlimited number of additional layers may be added. Figure 9.**3 a, b** shows a case with three layers. Since the state of those neuroids which belong neither to the top nor the bottom layers is, in principle, not known to the observer, these layers are called hidden layers. It is also possible, of course, that direct links exist between nonadjacent layers.

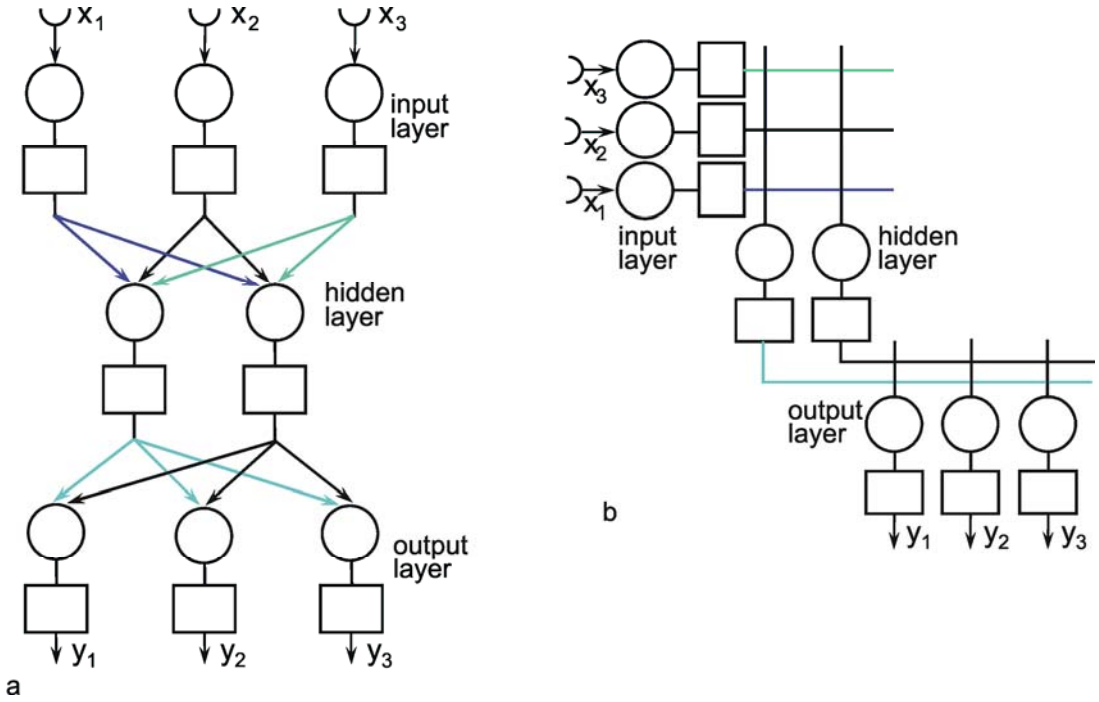


Fig. 9.**3 A three-layer feedforward network with three input units, two units in the intermediate, "hidden" layer, and three output units.** The boxes represent unspecified activation functions. Again, the illustration in (a) more resembles the biological situation, and the matrix arrangement provides an easier overview, particularly, if in addition direct connections between input and output layer are used (not shown)

Another form of connection occurs in the form of so-called recurrent nets. These contain feedback channels as described earlier for the special case of the feedback controller. In its simplest form, such a net consists of one layer. Each neuroid is provided with an input x_i from the exterior world and also emits an output signal y_i. At the same time, however, these output signals are relayed back to the input level, as shown in Figure 9.**4a.** Here, too, the matrix arrangement is helpful in illustrating this more clearly (Fig. 9.**4b**). The output of the j-th neuroid can, in principle, be calculated in the same way, namely as $y_i = f_i(_j\Sigma w_{ij}\ y_j)$, with neuroids i and j belonging the same layer, of course. From studies of the stability of feedback systems (Chapter 8.6) it follows that the behavior of recurrent nets is generally much more complicated than that of feedforward nets. It should, however, be emphasized that both

these types of nets are not alternatives in a logical sense. In fact, the recurrent net represents the general form since, by an elimination of specific links (i. e. introducing zero values for the corresponding weights), it can immediately be transformed into a feedforward net including feedforward nets of several layers (Fig. 9.**5**). For this purpose most of the weights (Fig. 9.**4**) have to become zero.

The properties of a net are determined by the types of existing connections as well as by the values taken by the individual weights. To begin with, the properties of nets with given weights will be discussed. Only later will we deal with the aspect to how to find weights, by searching for a net with desired transfer properties. These procedures are often called "learning."

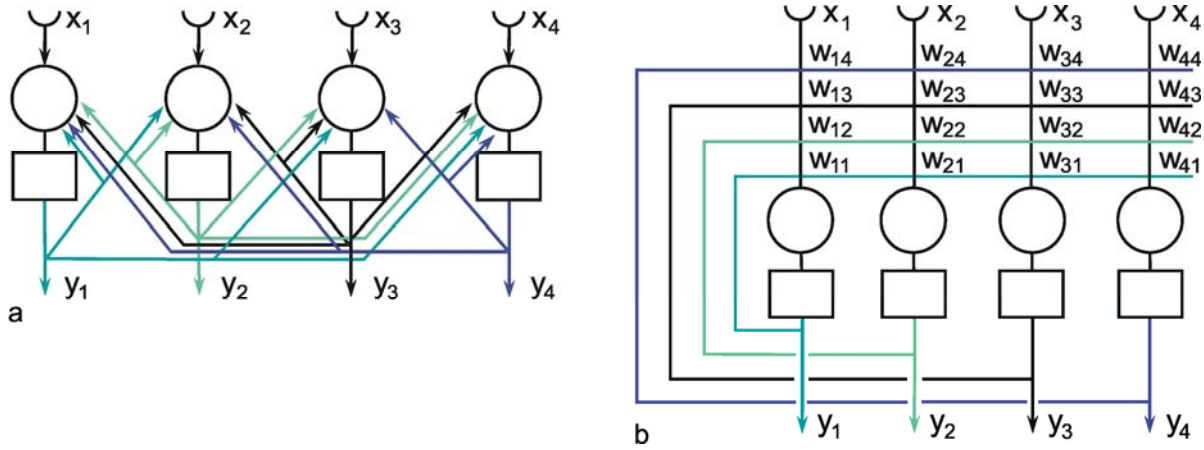


Fig. 9.**4 A recurrent network with four units.** In the case of the recurrent network the arrangement resembling biological morphology (a) is even less comfortable to read compared to the matrix arrangement (b). x_i represent the input values, y_i the output values. The boxes represent unspecified activation functions

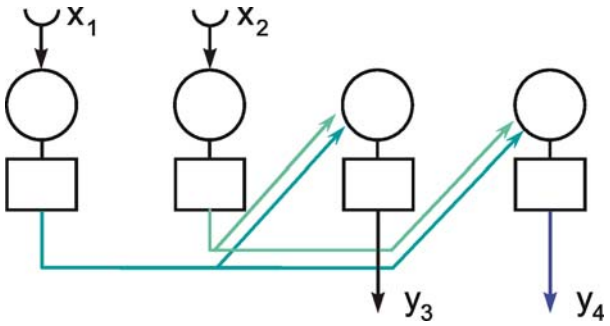


Fig. 9.**5 When some selected weights of the network shown in Fig. 9.4 are set to zero, a feedforward net, in this example with two layers, can be obtained.** To show this more clearly, the connections with zero weights are omitted in this figure. x_i represent the input values, y_i the output values

The designation of elements of these nets as neuroids suggests their interpretations as neurons. It should be mentioned, however, that other interpretations are conceivable as well, for example, in terms of cells which, during the development of an animal, mutually influence each other, e. g., by way of chemical messenger substances (see Chapter 11.1), and form specific patterns in the process. In a similar way, growing plants might influence each other and thus the whole plant might be represented by one "neuroid:" Another example are genes of a genom which may influence each other's activity by means of regulatory proteins.

Artificial neuroids have a number of inputs x_i and one output y. The input values are multiplied by weights w_i. The linear sum u might be subject to a (nonlinear) characteristic f(u). Dynamic properties like temporal low-pass filtering, or temporal and spatial summation are ignored. The neuroids might be arranged in different layers, an input layer, an output layer and several hidden layers. The neuroids might be connected in a pure feedforward manner or via recurrent connections.

10 Feedforward Networks

In the first four sections of this chapter, networks will be considered that are repetitively organized. This means that all neuroids have the same connections with their neighbors. In the subsequent three sections feedforward nets with irregular weight distribution, will be dealt with.

10.1 Spatial Low-Pass Filter

A simple structure, realized in many neuronal systems, is the so-called divergence or fan-out connection. Figure 10.**1** shows a simplified version. In this case, the two-layer feedforward net contains the same number of neuroids in each layer. Each neuroid of the second layer is assigned an input from the neuroid directly above it as well as inputs with smaller weights from the two neighboring neuroids. In a more elaborate version of this system, a second layer neuroid would be connected not only with its direct first layer neighbor but also with neuroids that are further removed; in this case, the value of the weights, i. e., the local amplification factors, are usually reduced as the distance increases. (By maintaining the sum of all the weights of a neuroid equal to one, the amplification of the total system also retains the factor one.)

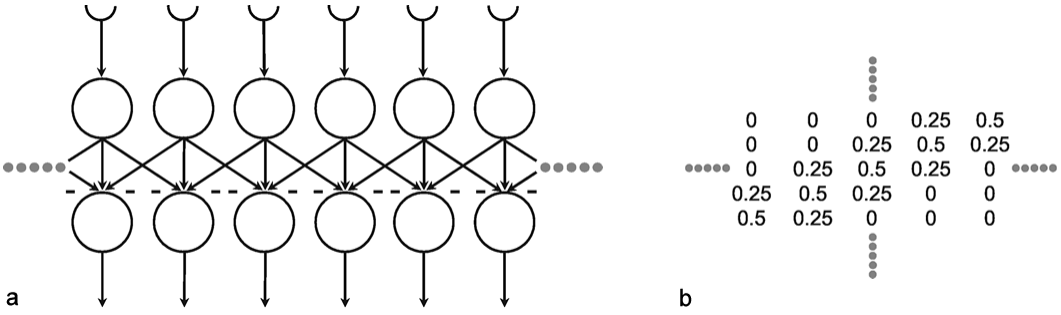


Fig. 10.**1 A two-layer feedforward net with the same number of input and output units.** Only a section of the net is illustrated. Directly corresponding neuroids in the two layers are connected by a weight of 0.5, the two next neighbors by weights of 0.25. This is indicated by large and small arrowheads, respectively. (b) shows the weights of the neuroids arranged in the matrix form described earlier

Figure 10.**2a** shows the response of such a system when one neuroid at the input is given the input value 1, while all others are given the value 0. For this input function, the output shows an excitation that corresponds directly to the distribution of individual weights. Since the input function can be considered an impulse function in spatial rather than in temporal domain, this response of the system can accordingly be defined as weighting function. This function directly reflects the spatial distribution of the weights. Figure 10.**2b** shows the response of the same system to a spatial step function. Here the neuroids 1 to i of the input layer are given the excitation 0, and the subsequent neuroids i + 1 to n, the excitation 1. In the step response the edges of the input function are rounded off. Qualitatively,

then, the system acts like a (spatial) low-pass filter. In contrast to the step response of a low-pass filter in the temporal domain, both edges of the step are rounded off in a symmetrical way. This property is already reflected in the symmetrical form of the weighting function (see difference to Figure 4.**1**). This property also affects the phase frequency plot, not shown here, which shows a phase shift of zero for all (spatial) frequencies. In a qualitatively similar way as described for the temporal low-pass filter, the corner frequency of the spatial low-pass filter is the lower the broader is the form of the weighting function. It should be mentioned that the Fourier analysis can, of course, also be applied to spatial functions. In Figure 3.**1** this could be done by interpreting the abscissa as a spatial axis instead of a temporal one. The rectangular function therefore corresponds to a regularly striped pattern. Examples of spatial sine functions will be shown in Box 4.

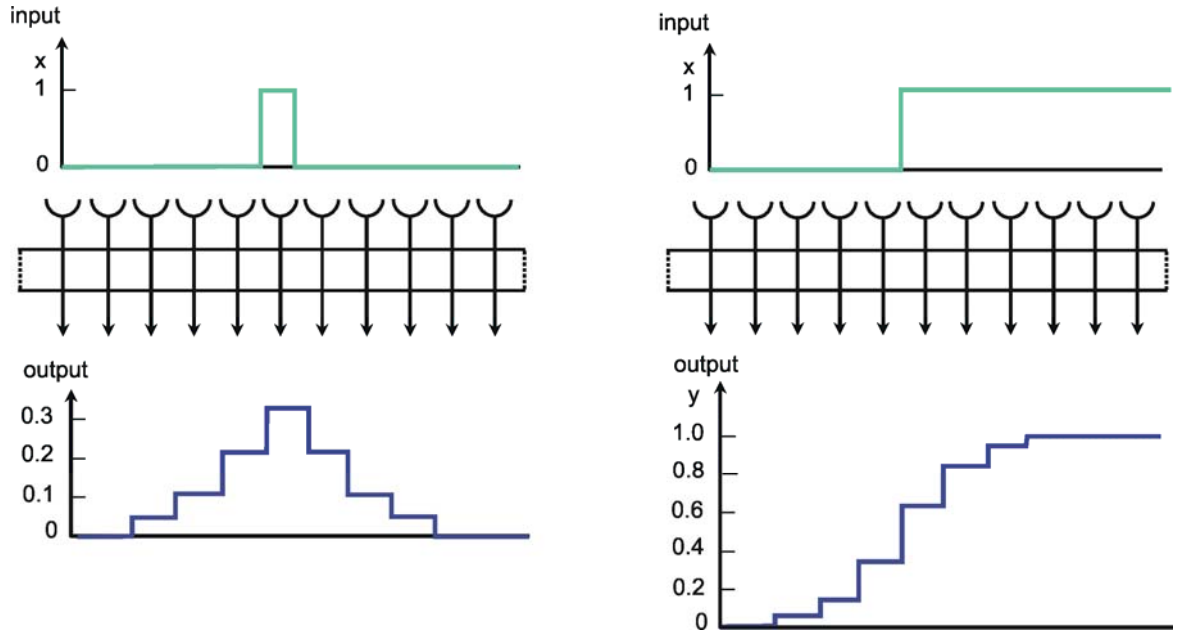


Fig. 10.**2 Impulse response (a) and step response (b) of a two-layer feedforward network (spatial low-pass filter) as shown in Fig. 10.1, but with lateral connections including three neighbors instead of only one.** The network is shown as a box. The central weight is 0.3. The weight values decrease symmetrically to right and left, showing values of 0.2, 0.1, and 0.05, respectively. These values can be directly read off the output, when a "spatial impulse" (a) is given as input, i. e., only one unit is stimulated by a unity excitation. The step response (b) is symmetrically, too, and represents the superposition of the direct and lateral influences. Input (x) and output (y) functions are shown graphically above and below the net, respectively

A spatial low-pass filter could be interpreted as a system with the property of "coarse coding." At first sight, the discrete nature of the neuronal arrangement might lead to the assumption that spatial resolution of information transfer is determined by the grid distance of the neurons. This seems to be even more the case when the neurons have broad and overlapping input fields. This overlap is particularly obvious for sensory neurons. For example, a single sensory cell in the cochlea is not only stimulated by a tone for a single frequency, but by a very broad range of neighboring frequencies. However, the sensitivity of the cell decreases with the distance between stimulating frequency and best frequency. In turn, this means, that a single frequency stimulates a considerable number of neighboring sensory cells. A corresponding situation occurs, if we consider several layers of neurons with fan-in fan-out connections as shown in Figure 9.**3**. The units in the second layer receive input also from neighboring units. In this way, as already mentioned, the exactness of the input signal degrades, and one might conclude that capability to determine the spatial position of a stimulus might grow worse as the grid size of the units increases. However, it could be shown (Baldi and Heiligenberg

1988, Rumelhart and McClelland 1986) that determination of the position of a single stimulus is better, the more the input ranges between neighboring units overlap. If, for instance, a single stimulus is positioned between two sensory units, the ratio of excitation of these two sensory units depends on the exact position of the stimulus relative to the two units. Thus, the information on the exact position is still contained in the system. An obvious example is represented by the human color recognition system. We have only three types of sensory units, but can distinguish a high number of different colors. These are represented by the ratios of excitation of these three units. Thus, although the system morphologically permits only "coarse coding," the information can be transmitted with a much finer grain by means of parallel distributed computation. Furthermore, such a spatial low-pass filter is more tolerant of failure of an individual unit, the more the input fields on the neighboring units in the net overlap.

Networks with excitatory fan-out connections show low-pass properties in the spatial domain. Symmetrical connections provide low-pass filters with zero phase shifts. In spite of coarse coding, in such a distributed network information can be transmitted with a much finer grain than that given by the spatial resolution of the grid.

10.2 Spatial High-Pass Filter

Figure 10.**3** shows a simple net with the same connection pattern as Figure 10.**1**, but a different distribution of weights. Each neuroid of the second layer receives an excitation, which has been given the weight 1, from the neuroid located directly above. From the neighboring neuroids of the first layer it additionally receives inputs of negative weights. As above, the figure shows only the linkages with the nearest neighbors; but linkages with neuroids further away, whose weights decrease correspondingly, are possible. In this case, the sum of all weights of a neuroid should be zero. Figure 10.**4a** shows the impulse response, i. e., the weighting function of the system when the connections with non-zero weights reach up to three neuroids in either direction. Correspondingly, Figure 10.**4b** shows the response to the step function. A comparison with the time filters shows that this system is endowed with high-pass properties. Here, too, the response, in contrast to the response of the time filter, exhibits symmetrical properties. These are reflected in the form of the step response and of the impulse response (compare to Figure 4.**3**).

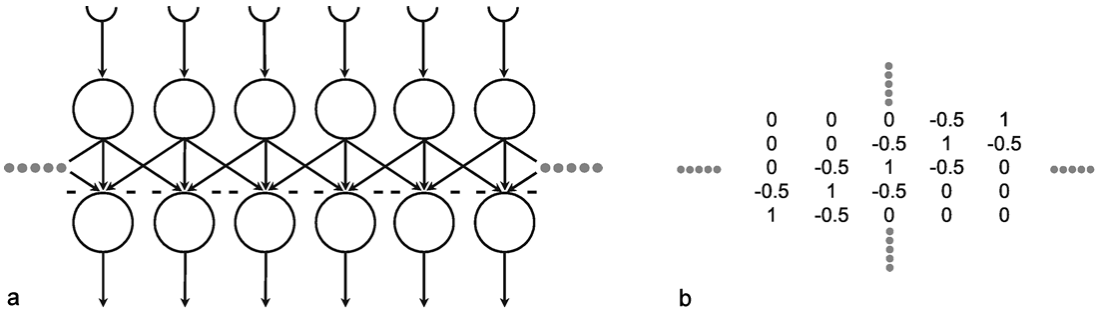


Fig. 10.**3 A section of a two-layer feedforward network as shown in Fig. 10.1, but with a different weight distribution.** Now, two directly corresponding neuroids in the two layers are connected by a weight of 1. The weights of the connections to the two neighboring units (thin arrows) are -0.5, forming the so-called lateral inhibition. (b) shows the corresponding matrix arrangement of the weights (see Fig. 9.**2b**)

The fact that in this net the lateral neighbors exhibit negative weights is called lateral inhibition in biological terms. This is a type of connection which can be found in many sensory areas. By means of

such a network, a low contrast in the input pattern can be increased. For example in the visual system such a low level contrast could result from using a low quality optical lens. An appropriate lateral inhibition could cancel this effect. A pure high-pass filter transmits only changes in the input pattern and thus produces an output which is independent of an absolute value of the input intensity. Thus, the output pattern is independent of the mean value of the input signals. There are also other forms of lateral inhibition which use recurrent connections (see Varju 1962, 1965).

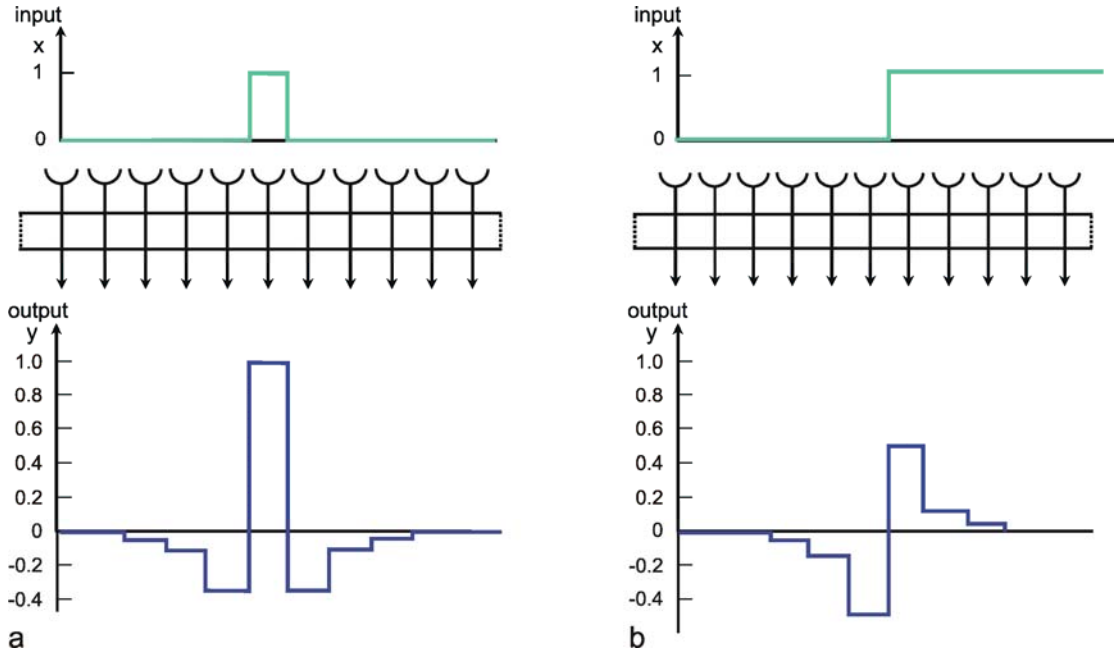


Fig. 10.**4 Impulse response (a) and step response (b) of a two-layer feedforward network as shown in Fig. 10.3, but with lateral connections including three neighbors instead of only one (spatial high-pass filter).** The network is shown as a box. The central weight is 1. The weight values decrease symmetrically to right and left, showing values of -0.35, -0.1, and -0.05, respectively. These values can be directly read off the output, when a "spatial impulse" (a) is given as input, i. e., only one unit is stimulated by a unity excitation. Input (x) and output (y) functions are shown graphically above and below the net, respectively

Networks showing connections with lateral inhibition can be described as a spatial high-pass filter.

10.3 Spatial Band-Pass Filter

In contrast to the time filters, a spatial band-pass filter can be obtained by connecting a spatial low-pass filter and a spatial high-pass filter in parallel. This occurs in many sensory areas, such as for example in the retina of mammals. As long as there are no nonlinearities, the weighting functions can be added up such that still only two layers are necessary. Figure 10.**5a** shows the weighting functions of a low-pass filter (E), a high-pass filter (I), and the sum of both (RF), plotted as a continuous function for the sake of simplicity. For the pure band-pass filter the sum of all weights acting on one neuron should accordingly be of value 1 (if this sum is larger than 1, we obtain a spatial lead-lag system, see Fig. 4.**10**). In this case the positive connection described for the high-pass filter is not more necessary, because this effect is provided by the positive low-pass connection. If we extend the one-dimensional arrangement of neuroids to a two-dimensional arrangement, making the corresponding connections in all directions, Figure 10.**5a** can be considered as the cross section of a rotation-symmetrical weighting function (Fig. 10.**5b**). When, as is the case in Figure 10.**5**, the weighting function of the high-pass filter

is wider, but has smaller weights than that of the low-pass filter, the result will be a weighting function with an excitatory center and inhibitory surrounding neighborhood. This connectivity arrangement is also known as "Mexican hat." This constitutes a representation of simple receptive fields as found in the visual system. When the weighting function of the low-pass filter is wider, but has smaller weights than the high-pass filter, we obtain a receptive field with a negative "inhibitory" center and a positive, excitatory surrounding.

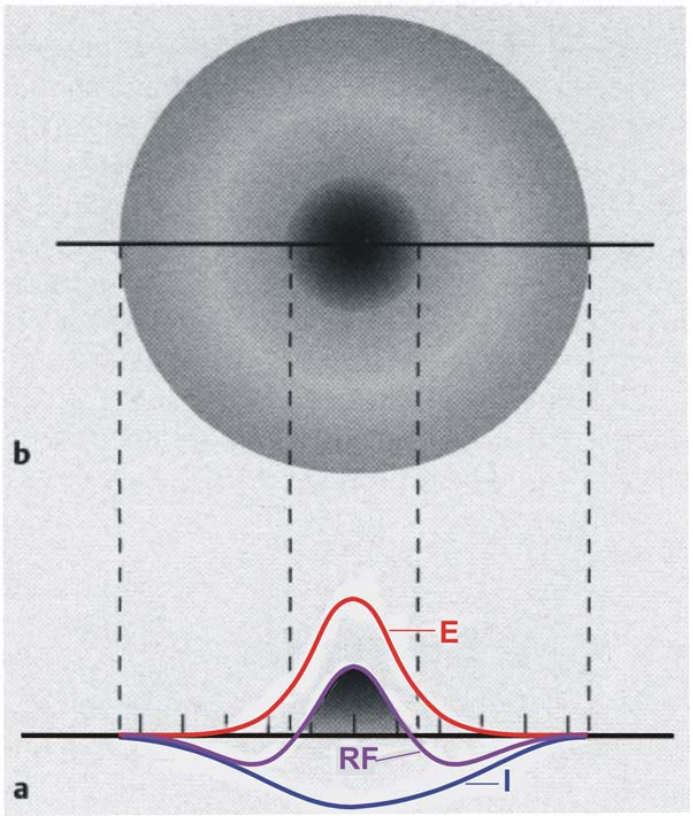


Fig. 10.**5 Distribution of the weights on a twodimensional, two-layer feedforward network.** (a) shows a cross section similar to those presented in Figs. 10.**2a** (E) and 10.**4a** (I), but for convenience the weighting functions are shown in a continuous form rather than in discrete steps. An excitatory center (dark shading) and an inhibitory surrounding is produced when both weighting functions are superimposed, forming a receptive field (RF). (b) shows this receptive field in a top view

It should be mentioned that in an actual neurophysiological experiment one usually investigates the weighting function, not by presenting an impulse-like input and measuring all the outputs values of the neurons in the second layer, but by recording from only one neuron of the second layer and moving the position of the impulse along the input layer.

Circular receptive fields with excitatory centers and inhibitory surroundings form the elements of a spatial band-pass filter.

Box 4

Spatial Filters in the Mammalian Visual System

A simple, qualitative experiment can illustrate that the human visual system shows band-pass properties. In Figure B4.**1** the abscissa shows spatial sinusoidal waves of increasing frequency. Along

the ordinate the amplitude of these sine waves decreases. In this way, the figure can be considered as illustrating the amplitude frequency plot. The sensitivity of the system, which in the usual amplitude frequency plot is shown by the gain of the system, is demonstrated here in an indirect way. Instead of using the gain, the sensitivity is shown by regarding the threshold at which the different sine waves can be resolved by the human eye. The line connecting these thresholds is illustrated in Figure B4.**1 b**. It qualitatively follows the course of the amplitude frequency plot of a band-pass filter.

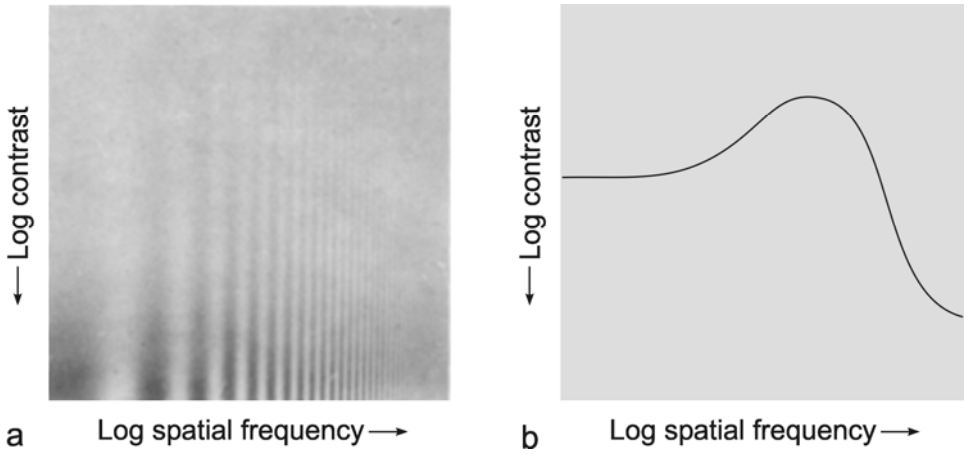


Fig. B **4.1 Qualitative demonstration of the low-pass properties of the human visual system.** As in the amplitude frequency plot, on the abscissa, sine waves with logarithmically increasing spatial frequency are plotted. The contrast intensity of each sine wave decreases along the ordinate. The subjective threshold, which is approximately delineated in (b), follows the form of the amplitude frequency plot of a low-pass filter (Ratliff 1965)

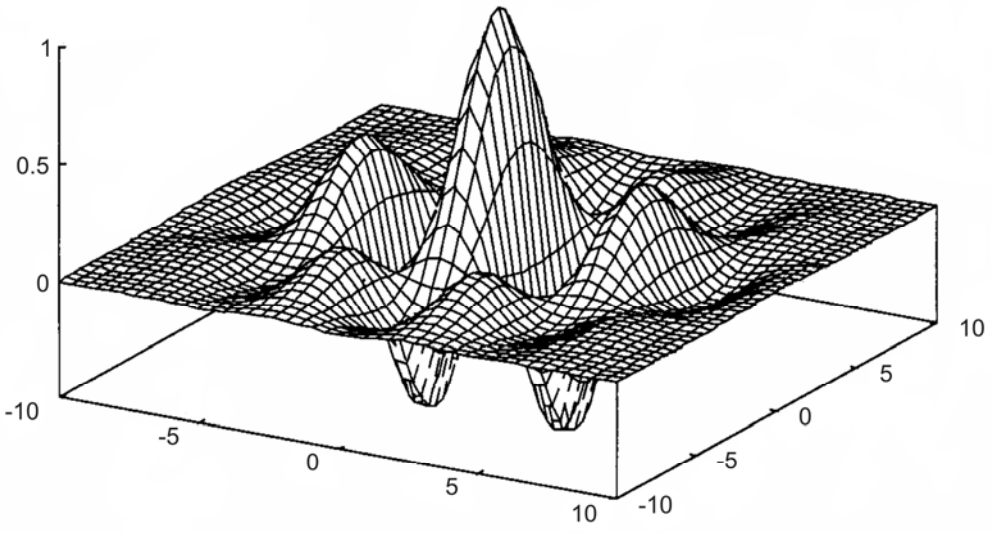


Fig. B 4.**2 A Gabor wavelet is constructed by first multiplying two cosine functions, one for each axis, which in general have different frequencies.** This product is then multiplied by a two-dimensional Gaussian function

Electrophysiological recording in the cortex of the cat indicate that the "Mexican hat" is only a first approximation for the weighting function of the spatial filter (Daugman 1980, 1988). The results could be better described by so-called Gabor wavelets. These are sine (or cosine) functions multiplied by a Gaussian function. In a two-dimensional case, the frequency of the sine wave might be different in the two directions. An example is shown in Figure B4.**2**. The application of Gabor wavelets to transmission

of spatially distributed information could be interpreted as a combination of a strictly pixel-based system and a system which performs a Fourier decomposition and transmits only the Fourier coefficients. The former system corresponds to a feedforward system with no lateral connections. It provides information about the exact position of a stimulus but no information concerning the global coherence between spatially distributed stimuli. Conversely, a Fourier decomposition provides only information describing the coherence within the complete stimulus pattern but no direct information about the exact position of the individual stimulus pixels. When a superposition of Gabor wavelets with different sine frequencies is used (for example a sum of wavelets, the frequency of which is doubled for each element added), they represent a kind of local Fourier decomposition, i. e., provide information on the coherence in a small neighborhood, but also contain spatial information. Thus, there is a trade-off between spatial information, i. e., local exactness, and information on global coherence.

A Gabor wavelet could be constructed by several ring-like zones of excitatory and inhibitory lateral connections. Thus, the "Mexican hat" can be considered as an approximation to a Gabor wavelet the sine frequency of which is chosen such that the second maximum of the sine wave, i. e., the first excitatory ring, is far enough distant from the center of the Gaussian function that its contribution can be neglected.

10.4 A Winner-Take-All Network

The following network has a similar structure to the inhibitory net described above. Each neuroid of the second layer obtains a positive input from the neuroid directly above it, which has been assigned the weight 1. All other neuroids obtain a negative, i. e., inhibitory, influence which, in this case however, does not depend on the positions of the neuroids in relation to each other. Rather, all negative weights have the same value. Since the sum of the weights at one neuroid will again be zero, in the case of n neuroids in each layer, the inhibitory synapses have to be assigned the value -1/(n - 1). This system, consisting of n parallel channels, has the following properties: if the inputs of the individual neuroids of the first layer are given excitations of different amplitude, these differences are amplified at the output, while the total excitation remains unchanged. This means that the difference between the channel with the highest amplitude at the input and all others at the output increases. Using a simple nonlinearity in the form of a Heaviside characteristic with appropriately chosen threshold value, the output values of all channels with a nonmaximum input signal become zero. A maximum detection is thus undertaken, or, in other words, the signal with the highest excitation chosen. For this reason, this system is also called a winner-take-all (WTA) net. One example, of its functioning is shown in Figure 10.**6**. For the purpose of clarity, the excitation values are given both before and after the nonlinear characteristics. The system is critical however with respect to the threshold position of the Heaviside characteristic. With an appropriately chosen threshold, the WTA effect could be obtained without using the net. The only advantage of the network lies in the fact that the difference is made larger and the position of the threshold is thus is less critical. The problem can be solved much better by winner-take-all systems based on recurrent nets which will be treated later (Chapter 11.2).

The winner-take-all network detects the maximally stimulated unit. This means that the unit with the highest input excitation obtains an output of 1 whereas all other units show a zero output.

10.5 Learning Matrix

The systems dealt with so far are characterized by the fact that the weights are prespecified and repeated for each neuroid in the same way. Below, the properties of those feedforward nets will be addressed in which the weights are distributed in an arbitrary way. This will be demonstrated using the simple example of a structure called "learning matrix" by Steinbuch (1961), but without actually dealing with the learning itself at this stage. A simple example is shown in Figure 10.**7**. The net contains six neuroids at the input and four neuroids at the output. Subsequently, these output signals are fed into a

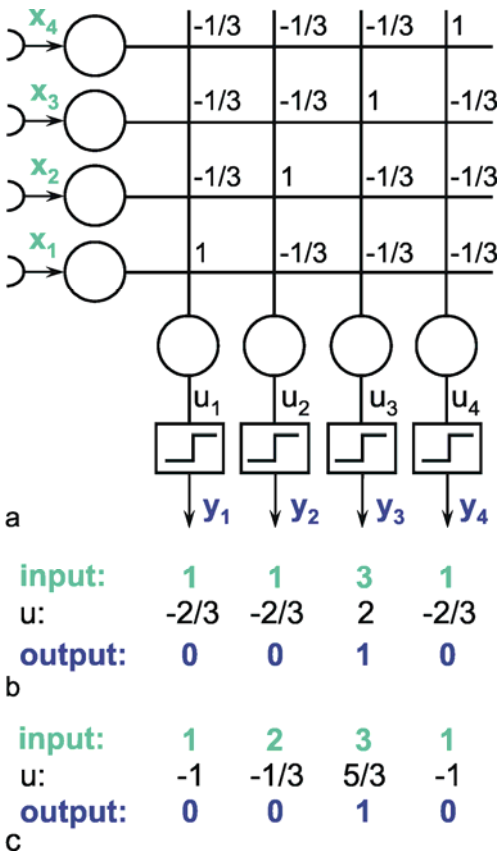
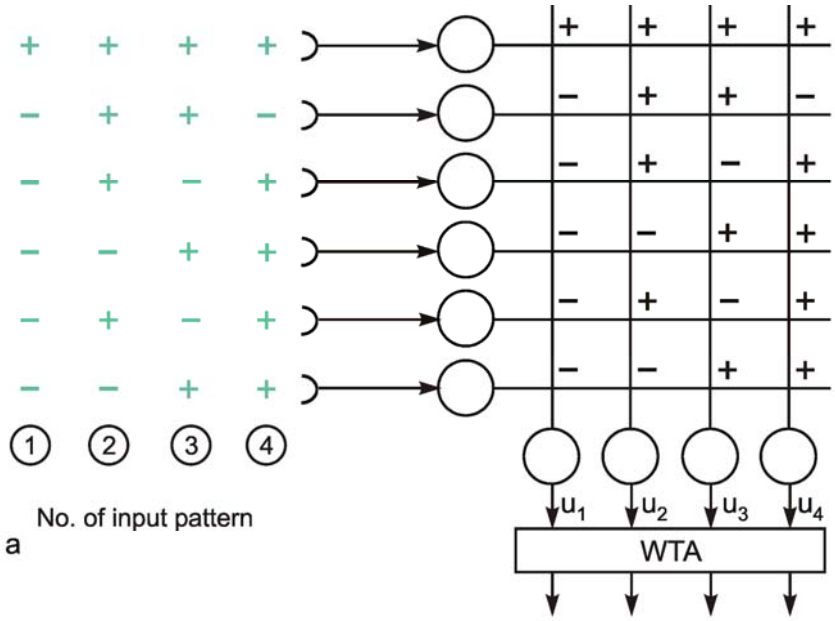


Fig. 10.**6 A feedforward winner-take-all network (a).** The weights at the diagonal are 1. Because the net consists of four neuroids in each layer, all other weights have a value of -1/3. (b) and (c) show two examples of input values (upper line), the corresponding summed excitations u_i (middle line) and the output values after having passed the Heaviside activation function (lower line). In the example of (b) the threshold of the activation function could be any value between -2/3 and 2. For (c) this threshold value has to lie between 1/3 and 5/3

The interesting property is that the net retains this ability even if, for some reason, one of the neuroids at the input ceases functioning. For instance, should the lowest input channel in Figure 10.**7a** fail, pattern 1 would, at the four outputs, exhibit the following values: 5, -1, 1, -1 and thus still be able to detect the right answer (Fig. 10.**7 c**). This property is sometimes referred to as graceful degradation. Moreover, the system responds tolerantly to inaccuracies in the input patterns. For example, if pattern 1 is changed in such a way that the sixth position shows +1 instead of -1, the outputs exhibit the following values: 4, - 2, 2, 0, and these still give the proper answer (Fig. 10.**7d**). Thus, the system does not only tolerate internal, but also external errors, i. e., "errors" in the input pattern. The latter may be also described as a capacity for "generalization," since different patterns, provided they are not too different, are treated in the same way. For this reason, such systems are also called classifiers. They attribute a number of input patterns to the same class. The properties of *error tolerance* and

generalization become even more obvious when the system is endowed with not just a few (as in our example for reasons of illustration) but many parallel channels. While the example chosen here, showing six input channels, will break down if three channels are destroyed, such an event will naturally have much less effect on a net with, say, a hundred parallel channels.



b

input pattern:	output: u_1	u_2	u_3	u_4
①	6	0	0	-2
②	0	6	-2	0
③	0	-2	6	0
④	-2	0	0	6

c

input pattern:	output: u_1	u_2	u_3	u_4
①	5	-1	+1	-1
②	-1	5	-1	+1
③	1	-1	5	-1
④	-1	+1	-1	5

d

1'	4	-2	+2	0

Fig. 10.**7 The "learning matrix," a two-layer feedforward net.** Input values and weights consist of either 1 or -1, but only the signs are shown. Four different input patterns are considered, numbered from 1 to 4 (left). (b) The summed values u; obtained for each input pattern. After passing the winner-take-all net, (Fig. 10.**6**, threshold between 0 and 6), only the unit with the highest excitation shows an output value different from zero. For each pattern, this is a different unit. (c) As (b), but the lowest input unit is assumed not to work. The winner-take-all net shows the same output as in (b), if the thresholds of the winner-take-all net units lie between 1 and 5. (d) The u; values obtained when a pattern 1' similar to pattern 1 (the sign of the lowest unit is changed) is used as input pattern. The output is the same as for pattern 1, if the thresholds lie between 2 and 4.

This system can also be regarded as information storage or memory. In a learning process, information on specific patterns, four in our example, has been stored in the form of a particular distribution of weights. This learning process will be described in Chapter 12 in detail, but it will not have escaped the reader that the weights correspond exactly to the patterns involved. These patterns are "recognized" after being put into the net. Here, the storage or input of data is not, as in a classical computer, realized via a storage address, which then requires the checking of all addresses in turn to locate the required data. Rather, it is the content of the pattern itself which constitutes the address. This type of memory is therefore known as *content addressable storage*. In addition to the properties already mentioned of error tolerance, this kind of storage has the advantage that, due to the parallel structure, access is much faster than in classical (serial) storage.

A distributed network can store information by appropriate choice of the weight values. In contrast to the traditional von Neumann computer, it is called a content addressable storage. The system can be considered to classify different input vectors and in this way shows the property of error tolerance with respect to exterior errors ("generalisation") and interior errors ("graceful degradation").

10.6 Perceptron

A network very similar to the learning matrix is the so-called perceptron (Rosenblatt 1958), which was developed independently at about the same time. The main difference is that in perceptron not only weights of the two values -1 and +1 are admissible, but also arbitrary real figures. This necessarily applies to the output values as well, which subsequently, however, at least in the classical form of the perceptron, are transformed by the Heaviside characteristic, thus eventually taking value 0 or 1. The threshold value of the Heaviside function is arbitrary but fixed.

The properties of such a network can be easily understood when some simple form of vector calculation is used. The values of the input signals $x_1 \ldots x_n$ can be conceived of as components of an input vector **x**, and similarly the weight values of the j-th neuron of the output layer $w_{1j} \ldots w_{nj}$, can be conceived of as components of a weight vector $\mathbf{w}_j$. The output value of the i-th neuron before entering the Heaviside function is calculated according to $u_i = {}_j\Sigma\, w_{ij}\, x_j$. u_i would then be the inner product (= scalar product, dot product) of the two vectors. The value of the scalar product is higher, the more similar the two vectors are. (It corresponds to the cosine of the angle between both vectors if both vectors have the same unit length.) The highest possible value is thus reached when both vectors agree in their directions. u_i can thus also be understood as a measure of the strength of the correlation between the components of both vectors. Such a feedforward net would thus "select," i. e., maximally stimulate, that neuroid of the output layer whose weight vector is the most similar to the vector used as input pattern. The numerical examples given in the table of Figure 10.**7** illustrate this. The function of the Heaviside characteristic is to ensure that only 0 or 1 values appear at the output. However, as mentioned earlier, the appropriate amount of the threshold value is critically dependent on the range of the input values. A better solution is given by using a recurrent winner-take-all net as described later.

A simple perceptron with only one single neuroid in the output layer would constitute a classifier that assigns all input patterns to one of two classes. Minsky and Papert (1969) were able to show that the capacity of this system for assigning a set of given patterns to two classes is limited. For example, it is not possible to assign all those patterns in which the number of input units showing +1 is even to one class, and all other patterns in which the number of input units showing +1 is odd to the other class (the so-called parity problem). On the other hand, many of the problems associated with the

perceptron can be solved, if the net is endowed with a further layer, i. e., has a minimum of three layers. This will be explained in more detail in Chapter 10.7.

Looking at a net endowed with m inputs and n outputs, the inputs, as mentioned above, can be interpreted as representations of different sensory signals, and the outputs as motor signals. The systems thus associates stimulus combinations to specific behavior responses. Since the output pattern and the input pattern, which are associated here, are usually different, we speak of a heteroassociator.

> The output of a unit of a two-layered feedforward network can be determined by the scalar product between the input vector and the weight vector of this unit. This product is greater the stronger the components of both vectors correlate. Different input vectors produce different output vectors. This is sometimes described as heteroassociation.

10.7 Multilayered Feedforward Nets

As already mentioned, multilayer nets, i. e., nets endowed with at least one "hidden" layer, exhibit some properties that are qualitatively new compared to two-layer nets. Two-layer nets can solve only problems which can be linearly separated. To show this we will use a simple example that is not linearly separable, namely the XOR problem. Solving this task requires a system with two input channels and one output channel, which should behave in the way shown in the table of Figure 10.**8a2**. The problem will be explained by representing the four input stimuli, i. e., the four two-dimensional vectors, in geometric form. According to the XOR task, these four vectors, which are shown in Figure 10.**8a3** geometrically in terms of their end points, are to be divided into two groups. This cannot be done linearly, i. e., by a line in this plane, since the two points which belong to the same class are located diagonally from each other. Other problems, e. g., the AND problem, are linearly separable and, for this reason, can be solved by way of a two-layer net, here consisting of a total of three neuroids (see Figure 10.**8b**). In this example, both weights are 1 and the threshold of the Heaviside characteristic is 1.5. The XOR problem can be solved only if a hidden layer is introduced, which, in this simple case, need only contain a single neuroid (Fig. 10.**8c**). The neuroid of the hidden layer is connected in such a way with the input units that it solves the AND problem. The output neuroid is then given three inputs which are shown in the table of Figure 10.**8c2.** The vector influencing the output neuroid is now three-dimensional, necessitating a spatial representation of the situation. As shown in Figure 10.**8c3**, the problem is now a linear one, i. e., separable by an oblique plane, and can be solved. By using the weights and threshold values shown in the network of Figure 10.**8c1**, the required associations can be obtained.

Thus, in some cases, hidden neuroids have to be introduced for logical reasons. It is also conceivable, though, that hidden neuroids are not required for the solution of a given problem, but that a net may become less complex, need fewer channels, and that individual neuroids could be interpreted as to have a symbolic significance. Figure 10.**9** provides a good, although somewhat artificial, example, namely a simulation of Grimm's Little Red Riding Hood (Jones and Hoskins 1987). In Figure 10.**9a** Riding Hood is simulated using a two-layer net. As we know, Riding Hood has to solve the problem of how to recognize the wolf, the grandmother, and the woodcutter and respond appropriately. To this end, she possesses six detectors which register the presence or absence (values of 0 or 1) of the following signals of a presented object: big ears, big eyes, big teeth, kindliness, wrinkles, handsomeness. If Riding Hood detects an object that is kindly, wrinkled, and has big eyes, which are supposed to be the features of grandmother, she responds by behaving in the following ways: approaching, kissing her cheeks, offering her food. With an object that has big ears, big eyes, and big

teeth, and is probably the wolf, Riding Hood is supposed to run away, scream, and look for the woodcutter. Finally, if the object is kindly, handsome, and has big ears, like the woodcutter, Riding Hood is supposed to approach, offer food, and flirt. The net shown in Figure 10.**9a** can do all this, if the existence or nonexistence of an input signal is symbolized by 1 or 0 and, correspondingly, the execution of an activity by 1 instead of 0 at the output.

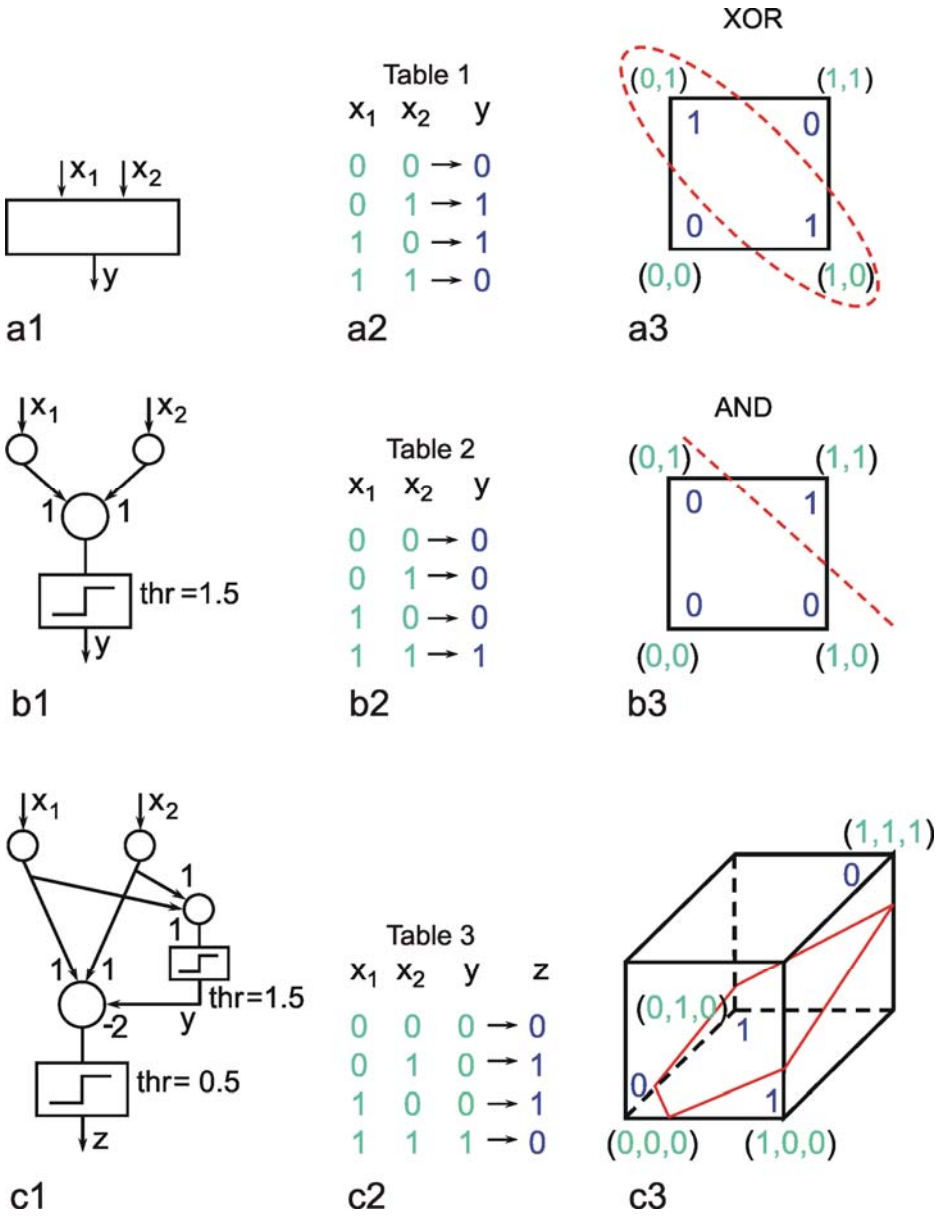


Fig. 10.**8 The XOR problem.** (a1) A system with two input units and one output unit should solve the four tasks presented in the Table (a2). These tasks are illustrated in geometrical form (a3) such that the four input vectors are plotted in a plane and the required output values are attached to their input vectors. The two input vectors which should produce an output value of one are circled. (b1) A two-layer system solving the AND problem. As in (a), the tasks to be solved are shown in Table (b2) and in geometric form in (b3). The network (b1) can solve the AND problem using the weights shown and a Heaviside activation function with a threshold of 1.5. (c) A three-layer system can solve the XOR problem, when, for example, the weights and threshold values shown in (c1) are applied. The output y of the hidden unit corresponds to that of the AND network (b1) (see Table c2). The input vector of the final neuroid, representing the third layer, is three-dimensional. Therefore, geometrically the input-output situation of this unit has to be illustrated using a three-dimensional representation. As above, the four input vectors are marked together with the expected output. The oblique plane separates the two vectors producing a zero output from the two others which produce unity output. Such a linear separation is not possible in (a3)

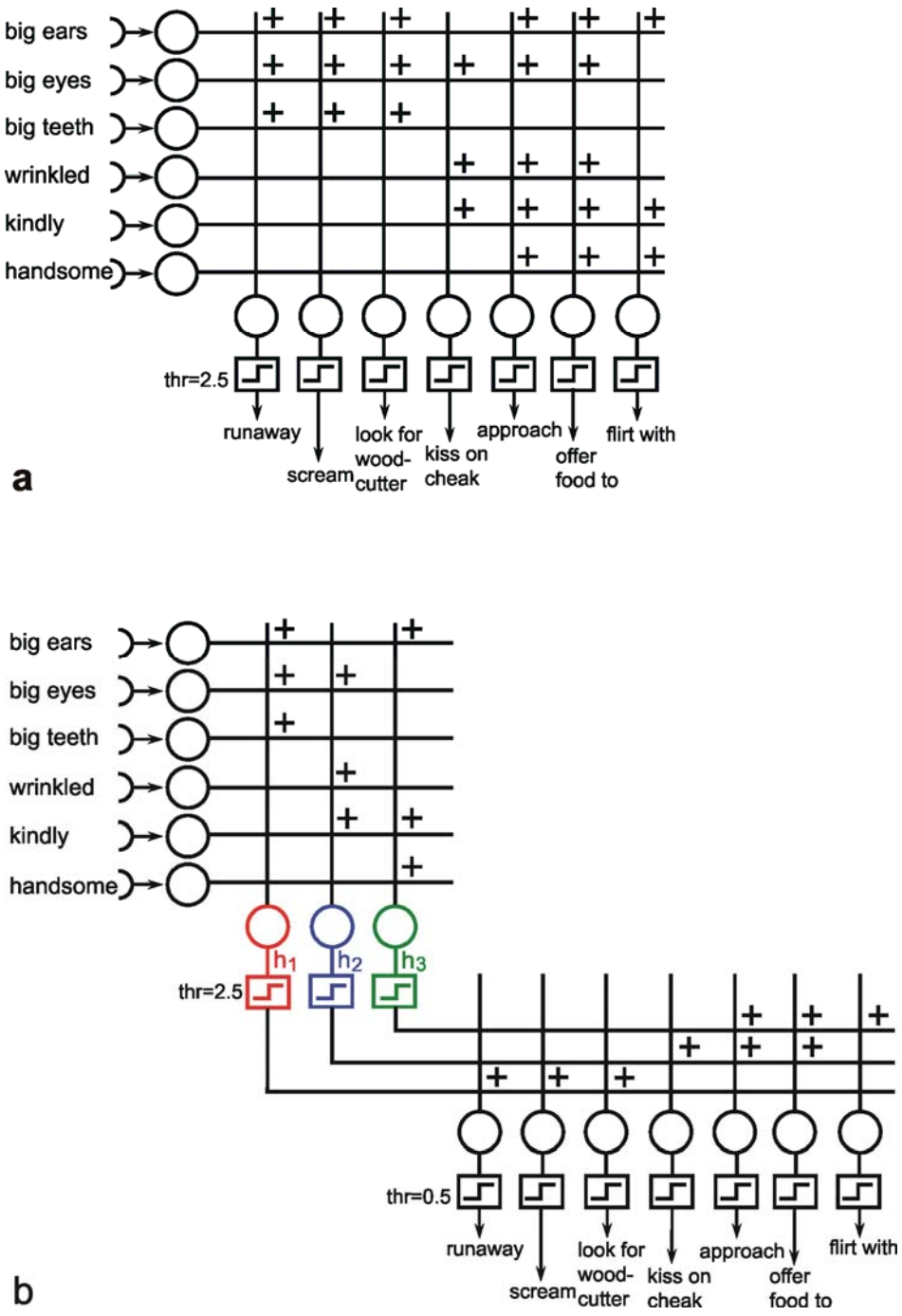


Fig. 10.**9 Two networks simulating the behavior of Little Red Riding Hood.** Input and output values are either zero or one. (a) In the two-layer net the thresholds of the Heaviside activation functions are 2.5. (b) For the three-layer network threshold values are 2.5 for the hidden units h1, h2, and h3 and 0.5 for the output units

Instead of a two-layer net, a three-layer net can also be used to simulate Riding Hood. This is shown in Figure 10.**9b**, with a middle (hidden) layer consisting of three neuroids. The question of how the values of the weights of both nets are obtained, i. e. the question of learning, will be discussed later. Both nets, the two-layered and the three-layered net, obey the same input-output properties but there are differences when the internal properties are considered. First, the three layered net requires fewer cross-sectional channels, i. e., number of weights per layer (a maximum of 21 in Figure 10.**9b**, compared to 42 in Figure 10.**9a**). Secondly, and this is especially interesting, a symbolic meaning can

be attributed to the three hidden neuroids. Neuroid h1 obtains a high excitation, if the features of the wolf are displayed at the input. The same applies to neuroid h2 in relation to Grandmother, and to neuroid h3 regarding the woodcutter. It is thus possible that intermediate neuroids can collect different, but qualitatively interrelated data, and can in this way be interpreted as symbols or concepts for the object concerned. If, as described later, these connections are learnt, one could argue that the system has developed an abstract concept of exterior world. It should, however, be mentioned here that the idea of the "grandmother cell" is intensively disputed in neurophysiology.

If, as an activation function, one includes not only the Heaviside function, but also linear or sigmoid ("semilinear") functions, both input and output values may take arbitrary real values. Generally, such a system exhibits the following property. It maps points from an n-dimensional (input) space onto points in an m-dimensional (output) space, thus representing a function. This could be a continuous function, whereas networks with Heaviside characteristics (see earlier examples in this chapter and also some recurrent networks) form functions with discrete output values. It can be shown that a net with at least one hidden layer can represent any function, provided it is integrable. A given mapping becomes more precise the greater the number of neuroids in the hidden layer. However, increasing the number of the hidden units can also provide problems as discussed in Chapter 12.

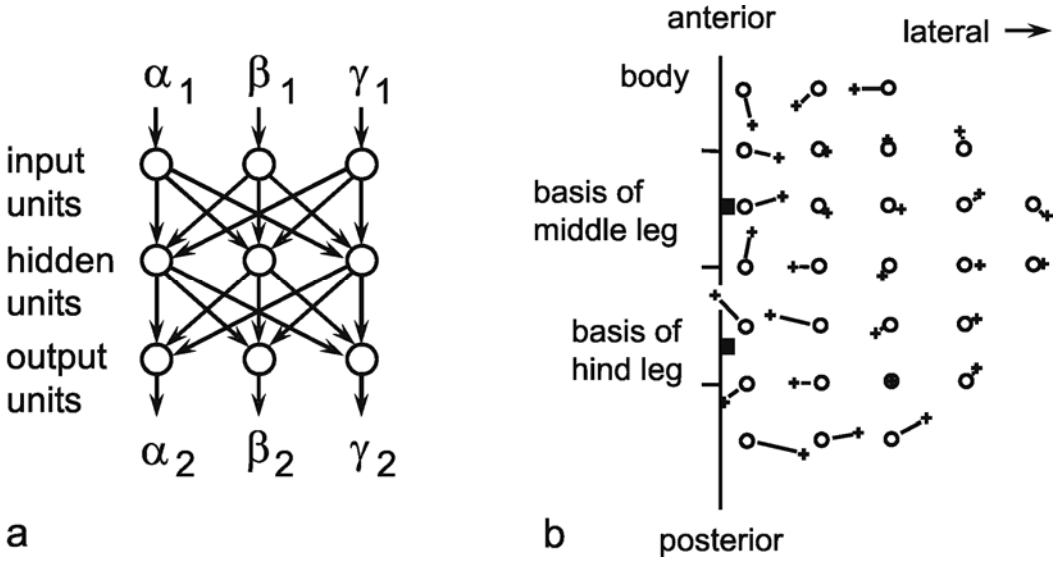


Fig. 10.**10 (a) A three-layer net which determines the position of an insect hind leg with three joints (given by the joint angles (α_2, β_2, γ_2), when the joint angles of the insect's middle leg (α_1, β_1, γ_1) are given.** The logistic activation functions are not shown. (b) Top view of the positions of the middle leg's foot (circles) and that of the hind leg (crosses). Corresponding values are connected by a line showing the size of the errors the network made for the different positions (after Dean 1990)

In the following, an example of such a mapping will be presented in which a two-dimensional subset of a three-dimensional space is mapped onto a two-dimensional subset of another three-dimensional space. During walking, some insects can be observed placing three legs of one side of their body in nearly the same spot. It can be shown that the reason for this is that the spatial position of the foot of one leg is signaled to the nearest posterior leg, which then makes a movement directed to this point. The foot's position is determined by three joints and hence three variables α, β, and γ. The system, which transmits the information from one leg to the next, thus, requires at least three neuroids in the input layer (Fig. 10.**10a**) (Dean 1990). These may correspond to the sense organs which measure the angles of the joints of the anterior leg. Furthermore, three neuroids are required in the output layer, serving to guide the three joints of the posterior leg. Since, in this model, only foot positions are

considered which lie in a horizontal plane, only two dimensional subsets of the three-dimensional spaces, which are generated by a leg's three joint angles, need to be studied. Figure 10.**10b** shows the behavior of the system in which three neuroids are given in the hidden layer. Circles show the given targets, crosses the responses of the net. The length of the connecting lines are thus a measure of the magnitude of the errors, which still occur when using only three hidden units.

This example illustrates the case of a continuous function. But it could be shown that such a net can also approximate mappings which, rather than having to be continuous, have only to be integrable, i. e., exhibit discontinuities.

Three (or more) layered feedforward networks can solve problems which are not linearly separable. Hidden units might represent complex (input) parameter combinations. A multilayered feedforward network constitutes a mapping from an m-dimensional sensor space to an n-dimensional action space (m, n: number of input and output units, respectively).

11 Recurrent Networks

As mentioned above, nets with feedback connections constitute the most general version because, by setting specifically chosen weights to zero, they can be transformed into any form of feedforward net. If feedback connections are present, the nets may have properties which do not exist in a simple feedforward net. This will be demonstrated using a fairly simple example, namely a net consisting of two neuroids. As activation functions saturation characteristics are used. These have the effect that the output values are restricted to the interval between +1 and -1 (Fig. 11.**1 a)**. All conceivable states (y_1, y_2) of this system can be represented in a two-dimensional state diagram by plotting the output values of the two neuroids along the coordinates (Fig. 11.**1 b**). Any point in the plane corresponds to a potential state of the system. What output values can be taken by this system? As a simple example, we use $w_{21} = w_{12} = -1$ and $w_{11} = w_{22} = 1$. Then the system is characterized by the fact that only two states, namely state (+1, -1) and state (-1, +1), are stable. All other states are unstable. This means that by an appropriate selection of the input values a specific state of the system can be brought about, but that, in the course of time, the system will by itself pass into one of the two stable states. The easiest way to understand this is to calculate this behavior by using concrete numerical examples. The state of the neuroids at time t + 1 can be determined according to the equations

$$y_1(t+1) = w_{12}\ y_2(t) + w_{11}y_1(t) + x_1(t)\ + \theta_1$$

$$y_2(t+1) = w_{21}\ y_1(t) + w_{22}y_2(t) + x_2(t)\ + \theta_2$$

with the offset values $\theta_1 = \theta_2 = 0$. This input to both neuroids could be represented by an additional “bias” unit showing a constant activation of 1 and being connected by weights θ_1 and θ_2, respectively. The input signal is turned off (set at zero) after the first cycle.

If one starts with state (+ 1, -1), this state is maintained for each subsequent cycle. If one starts, for example, with (0.5, 1), the resulting states are (-0.5, 0.5) (-1,1) (-1,1)... This means that the stable state is reached by the second time step. For other starting points the "relaxation" to a stable solution might take longer. Starting with (-0.2, 0.2), for example, yields (-0.4, 0.4), (-0.8, 0.8), (-1,1) as illustrated in Figure 11.**1 b**.

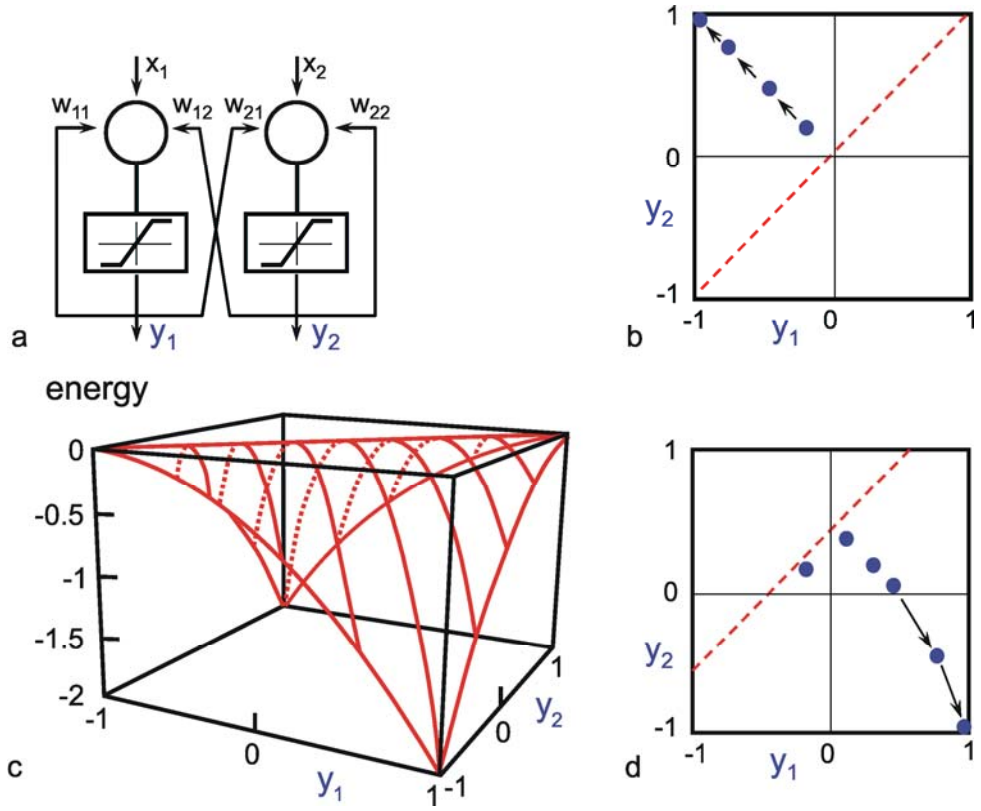


Fig. 11.**1 (a) A recurrent network consisting of two neuroids.** (b) The temporal behavior of the system shown in the state plane determined by the output values y_1 and y_2, starting with (-0.2/0.2). The dashed line separates the two attractor domains. (c) The energy function of the system. The energy (ordinate) is plotted over the state plane. (d) as (b) but using offset values $\theta_1 = 0.5$ and $\theta_2 = 0$. The (dashed) borderline between the two attractor domains is shifted and the system now finds another stable solution for the same input

The behavior of this system can be illustrated using an energy function introduced by Hopfield (1982). The "energy" of the total system is calculated as follows:

$$E = -\frac{1}{2}\sum_{ij} w_{ij} y_i y_j ,$$

or, if the neuroids contain an offset value θ_i,

$$E = -\frac{1}{2}\sum_{ij} w_{ij} y_i y_j + \sum_i \theta_i y_i .$$

(This offset value is sometimes misleadingly called threshold). In our case of two neuroids, this leads to $E = -1/2\ (w_{12}y_1y_2 + w_{21}y_2y_1 + w_{11}y_1y_1 + w_{22}y_2y_2)$.

For $w_{12} = w_{21} = -1$ and $w_{11} = w_{22} = 1$ we obtain $E = -\frac{1}{2}\ (y_1 - y_2)^2$. The area defined by this function is shown in Figure 11.**1c**. The system's behavior is characterized by the fact that it moves downward from any initial state and, in this way, reaches a state of rest at one of the two lowest points, i. e., (+1, -1) or (-1, +1). These states are therefore known as states with the lowest energy content, or as *attractors*. Such an attractor can be interpreted as representing a pattern stored in the net. Each attractor has a particular domain of attraction. Accordingly, the system becomes stabilized at that attractor in whose domain the initial state of the system was observed. As an attractor is defined a state that is not changed when being reached, the states $y_1 = y_2$ represent attractors, too. All states

characterized by $y_1 = y_2$ (e. g., 0.4, 0.4 or 0,0, see dashed line in Figure 11.**1 b**) belong to a special case. These attractors are instable ones because any small disturbance would drive the system away from such an attractor. Therefore instable attractors are less interesting.

Near the borderline between the domains of two stable attractors (dashed line in Figure 11.**1b**) the classical rule "similar inputs produce similar outputs" no longer holds. Small changes of the input signal may produce qualitative changes in the output. States where such “decisions” are possible are eventually termed bifurcation points. Such a network may also be regarded as a kind of distributed feedback control system because it shows an inherent drive trying to push the system in its attractor state. In this case the reference signal is not explicitly given but corresponds to the attractor of the domain of the actual input vector. The disturbance corresponds to the deviation of the input vector from the attractor. It should be noted at this point that, depending on the problem involved, other energy functions can also be defined (see Chapter 11.3 for an example). Other names for these energy functions are Lyapunov functions, Hamilton functions, cost functions or objective function.

The form of this energy plane can be altered by selecting different weight values, or by the fact that input values are not switched off, but remain permanently switched on. This corresponds to an alteration of the offset values of the neuroids. In this way, the size of the domain of attraction can be changed, and with it, of course, the behavior of the system. If, for example, $\theta_1 = 0.5$ and $\theta_2 = 0$, the range of the attractor 1, -1 becomes larger, and its minimum lower than those of the attractor -1, 1 (Fig. 11.**1d**). The borderline is shifted as indicated in Figure 11.**1d**. Starting at (-0.2, 0.2), the state will now change to (0.1, 0.4), (0.2, 0.3), (0.4, 0.1), (0.8, -0.3), and (1, -1), and therefore reach the other attractor.

The concept of an energy function can also be used if the system is endowed with more than two neuroids. This is generally known as a Hopfield net. In such a net, the activation function of neuroids may be linear in the middle range, or nonlinear, but has to show an upper and a lower limit. A further condition, and thus a considerable limitation of the structure of Hopfield nets, is that connections within the net must be symmetrical, i. e., $w_{ij} = w_{ji}$. This precondition is required so that the energy function introduced by Hopfield can be defined. Detailed rules describing the behavior of nonsymmetrically connected nets in general are as yet unknown.

On account of the symmetry condition, the number of potential weight vectors, compared to a net without this constraint, and correspondingly, the number of potential attractors or stable patterns (see Chapter 11.4), is lower than the maximum number of patterns that could be represented by a net of n neuroids. The symmetry condition follows from the application of Hebb's Rule in storing the patterns (see Chapter 12.1). Moreover, with this assumption, it is easy to define the energy function in question. Assuming that the patterns show a random distribution, i. e., that about half of the neuroids take value 1 and the other half value -1, one can store maximally n/(4 log n) patterns in a net consisting of n neuroids, if, after recalling the pattern, this pattern is to be generated without errors. If an error of 5% is tolerated, the maximum figure is 0.138 n. The reason for assuming randomness of distribution is that, in this case, no correlation is observed between the patterns. If correlation exists, cross talk is greater and the capacity of the net smaller, accordingly. (A special case of nonrandomness of distribution is present, if the patterns are orthogonal in relation to each other (see Chapter 12. At most n patterns can then be stored and recalled without error.)

If a net is endowed with n recurrently connected neurons, the energy function is defined within an n-dimensional space. It may then exhibit a higher number of minima, i. e., attractors and their domains of attraction. The depth of the minima and their domains of attraction may greatly vary. Undesirable local minima may also occur, known as spurious states. Usually, Hopfield nets are considered in which $w_{ii} = 0$, i.e. there are no connections of a neuroid onto itself. This is, however, not necessary. Positive

values of w_{ii} reduce the number of spurious states by slowing down changes of the activation values of the neuroids because thereby the dynamic of a temporal low-pass filter is introduced (see Chapter 11.5).

An important property usually applied in the Hopfield nets is that updating of the neuroids is not performed in a synchronous manner. Although synchronous updating was used in the above example (Fig. 11.**1**), asynchronous updating corresponds more to the situation of biological systems. A synchronous system is easier to implement on a digital computer but requires global information, whereas in biological systems only local information is available. Asynchronous updating can be obtained, for example, by randomly selecting the neuroid to be updated. However, the following examples will use synchronous updating which can be considered as a special case of asynchronous updating.

Recurrent networks constitute the general form of networks. However, only special versions can be understood at present. For symmetrical networks ($w_{ij} = w_{ji}$) an energy function can be formulated which illustrates the behavior of such a net. If the neuroids are endowed with bounded characteristics, the network belongs to the class of Hopfield nets. For any input vector, the net "relaxes" to one of several stable states ("attractors") during a series of iteration cycles. This attractor corresponds to a minimum of the energy function

11.1 Pattern Generation

As an example for Hopfield nets, first a recurrent net will be studied in which the neuroids are arranged topologically (e.g. along a line in the 1D case or in a plane in the 2D case) and the weights of each neuroid are arranged in the same manner. This means that, apart from the neuroids at the margin of the net, each neuroid is connected to its neighbors in the same way. The net's structure is thus repetitive. With adequate connections, the network is then capable of forming stable excitation patterns.

In biological systems, pattern generation is a frequently encountered phenomenon, e.g., the formation of stripe patterns in skin coloring, or the arrangement of leaves along the stem of a plant. Another problem concerning an organism's ontogenesis is posed by the generation of polarity without external determining influences being present. When the body cells *of Hydra* are experimentally disconnected and mixed up, an intact animal will be reorganized from this cell pile such that after some time one end of the heap contains only those cells that form the head of the animal and the other contains those that form its foot. How can the chaotic pile of cells of a *Hydra* become an ordered state? Assuming that the individual cells spontaneously choose one of the two alternatives, namely +1 and -1(i. e., "head" or "foot" cell), the formation of the desired polarity is possible as follows. Each "head cell" emits a signal, stimulating growth of the head, to the neighboring cells and another signal, inhibiting growth of the head, to the more distant cells. The corresponding applies to each "foot cell". These mechanisms can be simply demonstrated using a recurrent network. (Of course, the connections are not formed by axons and dendrites in this case. Rather, the cells send hormone-like substances and, by means of diffusion and possibly decomposition, the concentration decreases with distance from the sender.) One assumes that each cell is represented as a neuroid which is linked to its direct neighbors via positive weights, corresponding to excitatory substances, and to its more distant neighbors via negative weights, which correspond to inhibitory substances. The neuroids summate all influences and then "decide," with the aid of a Heaviside activation function, whether they are going to take value +1(for head) or -1 (for foot) at the output. Figure 11.**2** shows the behavior of the system, after an

appropriate choice of distributions of inhibitory and excitatory weights. After only a few time steps, a stable pattern develops from the initially random distribution of head and foot cells such that, at one end only head cells are to be found, whereas the other end shows only foot cells. Depending on the initial state, the final arrangement may also be exactly reversed. Thus the system has two attractors. If the weight distribution is chosen in such a way that coupling width is less than that for the system shown in Figure 11.**2**, periodic patterns may also result. One could say, therefore, that these systems are characterized by the fact that, by self-organization (i. e., on the basis of local rules) they are capable of generating a global pattern. It should be mentioned that such neuroids with Heaviside characteristics and connected by local rules can correspond to systems often described as cellular automata.

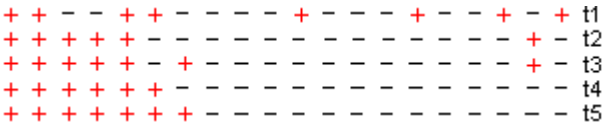


Fig. 11.**2 Development of an ordered state.** The recurrent net consists of 20 units which are arranged along a line. The units can take values of 1 or -1. Only the signs are shown. The first line shows an arbitrary starting state. During the four subsequent time steps (iteration cycles t2 - t5) the net finds a stable state. Positive output values are clustered at the left and negative values at the right. For another starting situation the final state may show the cluster of positive values on the right side

In the following we consider a similar network which represents a simple case of “neural fields” as studied by Amari (e.g. Amari 1977). Different to the net explained above as activation function a simple rectifier is used instead of the Heaviside function. Furthermore, the activation of the complete net is normalized to a given value, say 1. If a pulse-like input is given for one iteration at any neuroid, after relaxation the output looks similar to the positive part of the Mexican hat-like weighting function. The maximum of this hill-shaped activation distribution corresponds to the position of the neuroid that received the input. As the activation is stable over time, it might be considered as a memory storing the position of the input. If, instead, two pulses are given at different positions, the same hill-shaped activity distribution will appear, the maximum of which is now localized in between the positions of the two stimulated neuroids, thus forming some kind of spatial mean value. If these two input values have different amplitudes, the position of the final activation distribution is shifted to the more excited neuroid. Thus, the position corresponds to a weighted mean. Such a network could also be used to represent a “travelling wave”, if, after the hill-shaped activation has been stabilized, a small positive pulse is given at one, say the right side of the hill, which makes the hill moving in this direction (a small negative pulse being given to the left side produces the same result, if the input meets a neuroid with positive activation). Of course, these ideas can be applied not only to a linear arrangement of neuroids, but to a two-dimensional grid or to nets of higher dimensions.

With appropriately chosen weights, a recurrent network can show the formation of spatial patterns. This can be considered as an example of self-organization.

Box 5

Diffusion Algorithms

Diffusion is an important phenomenon because it forms the basis of many physical processes. Furthermore, the diffusion analogy can also be used to understand other processes as explained below. To describe those processes, the so-called reaction-diffusion differential equations often are used. However, diffusion can also be simulated using artificial neural nets.

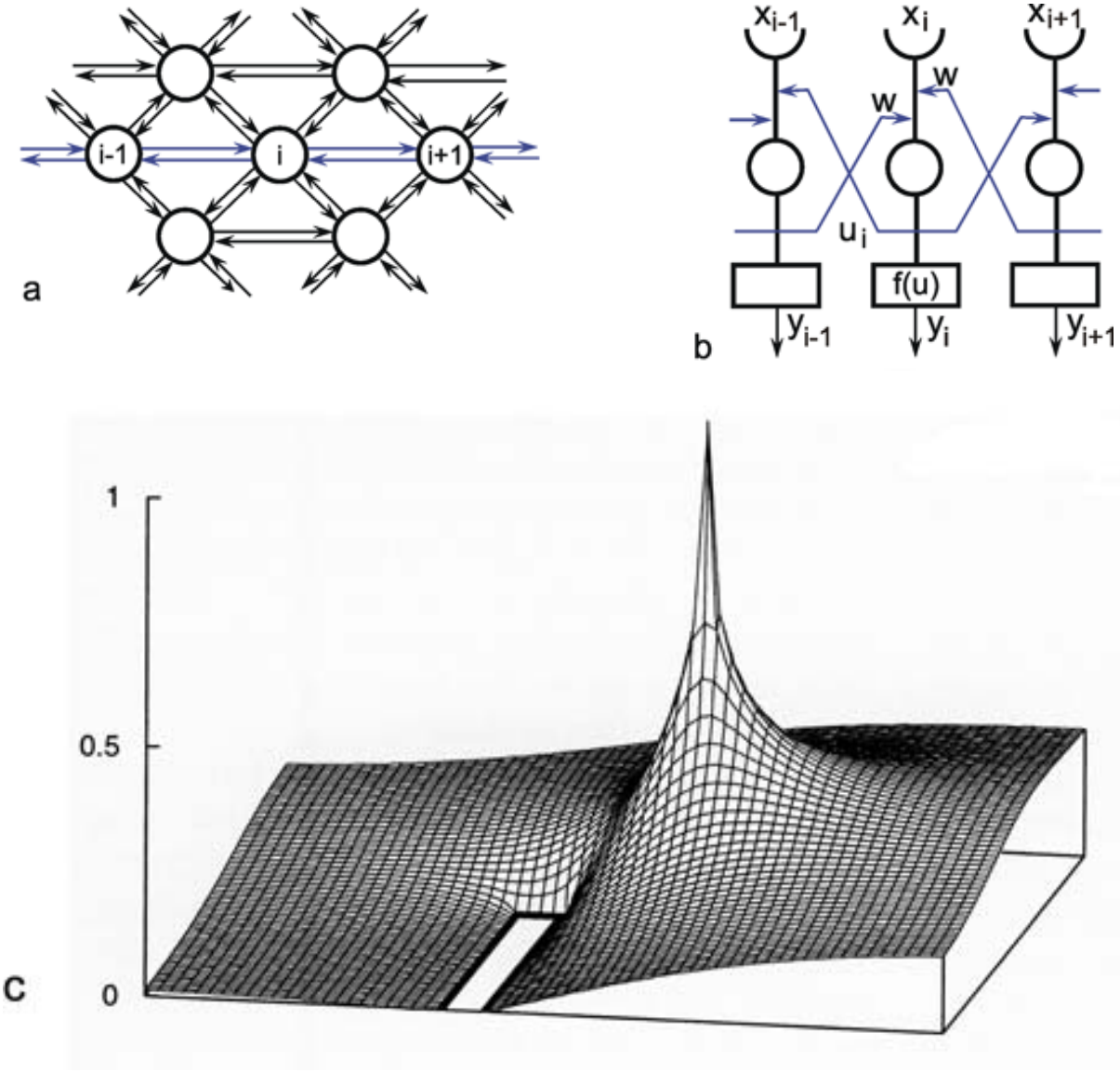


Fig. B 5.**1 Top view of a small section of the two-dimensional hexagonal grid.** Each unit is connected with its six direct neighboring units. (b) Cross section showing those three units and their recurrent connections which, in (a), are marked by bold lines. The weights are w = 1/6. (c) The excitation of the units after 5000 iteration cycles. The bold rectangle marks an obstacle (Kindermann et al. 1996)

As an example we will first consider the simulation of diffusion in a two-dimensional space. For this purpose, we represent this space by a two-dimensional grid of neuroids that obey recurrent connections with direct neighbors. This is shown in Figure B5.**1a, b**. In (**a**), the top view is shown, and

in (**b**), a cross section illustrating the connections between the neuroids i -1, i, and i + 1 is shown. The neuroids are arranged in the form of a hexagonal grid. Therefore, each neuroid has six neighbors. All weights are w = 1/6 except for those neuroids that coincide with the position of an obstacle (see below). To start the simulation, one neuroid is chosen to represent the source of a, say, gaseous substance. This neuroid receives a constant unit input every iteration cycle. As a result, a spreading of excitation along the grid can be observed (Fig. B5.**1 c**). This excitation, in the diffusion analogy, corresponds to the concentration of the diffusing substance, forming a gradient with exponential decay along the grid. (The exponential decay could be linearized, if this is preferred, by introduction of a logarithmic characteristic f(u) to compensate the exponential decay. In our example, however, f(u) is linear). Figure B5.**1 c** shows the result of such a diffusion which was, in this case, produced in a nonhomogeneous space: Some parts of the grid are considered as tight walls, which hinder the diffusion of the substance, i,e., connections across these walls are set to zero. The position of these walls is shown by bold lines. After 5000 iteration cycles, the excitations of the neuroids form a surface as shown.

Such a gradient could be applied, for example, to solve the problem of path planning in a plane cluttered with obstacles. In this case, the goal of the path is represented by the diffusion source. The walls represent the obstacles. The path can start at any other point in the grid. The path-searching algorithm simply has to step up the steepest gradient in this "frozen" diffusion landscape. If paths exist, this procedure finds a path to the goal, for any arrangement of the obstacles (Kindermann et al. 1996).

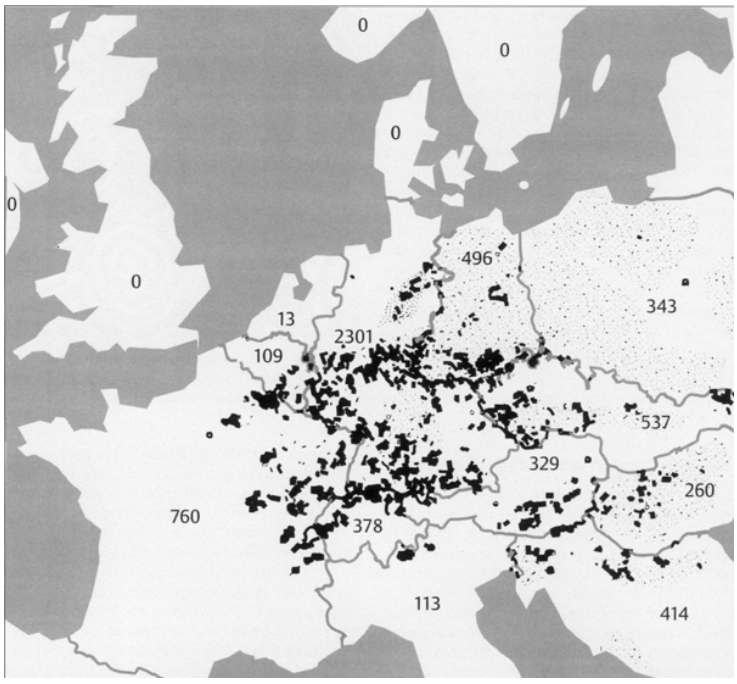


Fig. B 5.**2 Spotty distribution of rabies (dots) in central Europe in the year 1983.** In brackets: number of counts. The data from Poland and the former DDR are not comparable with the others because they were collected over larger spatial units (after Fenner et al. 1993; courtesy of the Center for Surveillance of Rabies, Tübingen, Germany)

As another example, the development of an epidemic disease, such as rabies, can be considered. Figure B5.**2** shows the spotty spatial distribution of rabies carried by foxes in Europe. This development is assumed to have begun in Poland in about 1939. Rabies also shows a temporal cyclic pattern with a new epidemic appearing every 3-5 years at a given local position. This can be understood by representing the area using a two-dimensional grid, the excitation of the neuroids

representing the intensity of infection of the corresponding local area. First of all, there is a diffusion process of spreading the infection as described above. However, a further mechanism is necessary. After infection in an area has reached a given threshold, self-infection leads to an almost complete infection of that area. This can be simulated by a recurrent excitation of the corresponding neuroid and an activation function with a saturation, describing the fully infected state. With these mechanisms, the infection would spread out over the grid like a wave front. There is, however, a third, important phenomenon. When an area is heavily infected, the foxes die from the disease and therefore the number of infected foxes in this area decreases to about zero. This "reaction" can be simulated by each neuroid being equipped with high-pass properties, or by application of an additional neuroid as shown in Figure 11.**8**. When the time constant of this high-pass is chosen appropriately, spatial and temporal rhythmical distributions can be found in this simulation of a reaction-diffusion system, which very much correspond to those found in nature. Exceptional cases of further examples can be found in Meinhardt (1995).

In the following example, other local rules have been applied. The cells are arranged in an orthogonal grid and each cell is connected with its eight neighbors. Each cell can adopt three different states of excitation (shown by white, gray, and black shading). The system is started with a random distribution of these states. The local rules are as follows: if a cell is in state 1 and three or more neighboring cells are in state 2, the cell adopts state 2 in the next iteration cycle. Correspondingly, if a cell is in state 2 (3) and three or more neighbors are in state 3 (1), the cell adopts state 3(1). After some time, a picture results similar to that shown in Figure B5.**3**. These spirals are not static, but change their shape with time, increase their diameters, seem to rotate, and merge with other spirals. The temporal behavior of this structure reminds one of the rotating spirals that can be observed during the growth of some fungi species. For the sake of simplicity, the local rules used here a formulated by logical expressions, but could be reformulated for an neuronal architecture.

Fig. B 5.**3 A snapshot of moving spirals** (Courtesy of A. Dress and P. Sirocka)

11.2 Recurrent Winner-Take-All Network

As mentioned above (Chapter 10.4) a winner-take-all network can also be constructed on the basis of a recurrent network. As shown in Figure 11.**3a**, this can be obtained when each neuroid influences itself by the weight 1 and all other neuroids by the weight -a. In general, if the system is linear, the

output values of this net will oscillate. If, however, half wave rectifiers are used as activation functions, the system shows better properties than the feedforward winner-take-all net. Even for small differences between the input values, the net always produces a unique maximum detection. The higher the value of a, the faster the winner appears, which means that all output values except for one are zero. If the value of a becomes too great, oscillations might occur. The optimal value is a = 1/(n-1) for a net consisting of n neuroids. Figure 11.**3b** shows a formally identical net in which a special neuroid is introduced to sum all output values. This value is then fed back.

Compared to the feedforward system, the recurrent winner-take-all net shows the same properties but behaves as if it could adjust its threshold values appropriately for the actual input values. The behavior of the recurrent system can be made clear by imaging instead a chain of consecutive feedforward winner-take-all nets. Each such feedforward net corresponds to a new iteration cycle of the recurrent system. Using this so-called "unfolding of time," any recurrent network can be changed into an equivalent feedforward network (s. Fig. 12.**4**).

A recurrent winner-take-all net can be used in two ways. First, the input vector can be clamped continuously to the system during the relaxation. Alternatively, the input vector can only be given into the system for the first iteration cycle and then switched off. In the second case, relaxation is faster and the value of the winner becomes stable. However, the value of the winner is small when the difference between the input values is small. In the first case, the value of the winner increases steadily with the iteration cycle. In both cases the problems can be solved when an additional neuroid is introduced to normalize the sum of all output values to 1. The output of the extra summation unit shown in Figure 11.**3b** can be used for this purpose. Examples of how WTA nets could be used for decision making are given in Chapter 16.

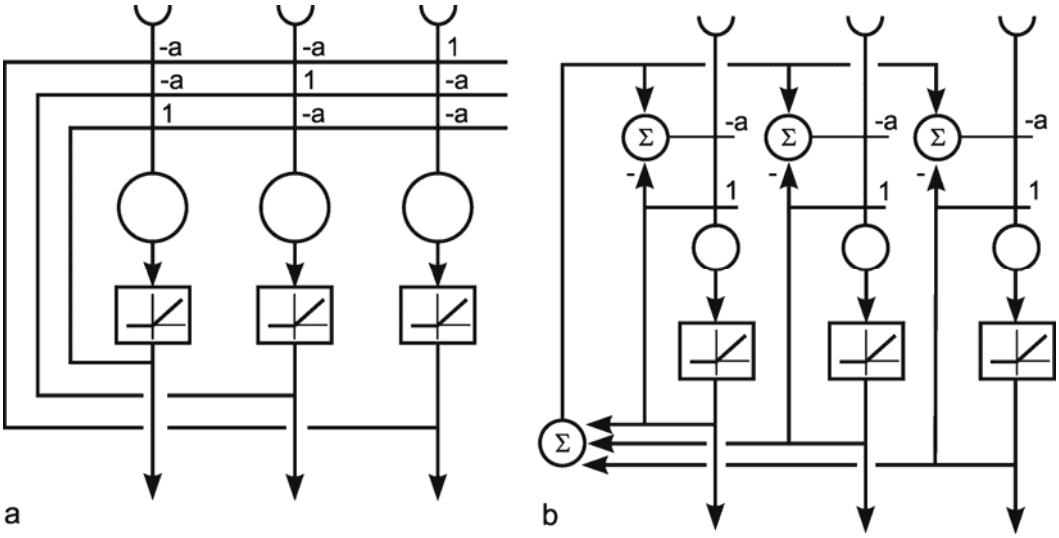


Fig. 11.**3 A recurrent winner-take-all network consisting of three units (a).** A formally identical system with four extra units (b). Rectifiers (Fig. 5.**8**) are used as activation functions

Using a recurrent structure, winner-take-all nets can be constructed having more convenient properties than those based on a feedforward structure.

11.3 Optimization

As we have seen, recurrent nets with a given weight distribution are endowed with the property of decreasing from an externally given initial state to a stable final state, i. e., "relaxing." This property could also be useful in solving optimization problems. This will be illustrated by the following example (Tank and Hopfield 1987). Let us assume that we have six subjects to solve six tasks. Each subject's ability differs in relation to the various tasks. To illustrate this, each line in the table of Figure 11.**4a** shows the qualities of one subject in relation to the six tasks, in numerical terms. The question arises as to which subject should be assigned which task to ensure that all the tasks will be optimally solved. There are 6! (= 720) possible solutions. The problem can be presented in the form of a net as follows: For each subject and each skill a neuroids is used with a bounded activation function (e.g., tanh ax, see Fig. 5.**11b**). The best way to arrange these 36 neuroids is in one plane as shown in Figure 11.**4a**, i.e., to arrange those neuroids representing the skills of one person in a line, and those representing the same skills of the various subjects in a column. Since each task is undertaken by only one subject, the neuroids in each column have to inhibit each other (this forms a recurrent winner-take-all subnet). Correspondingly, since each person can only do one task, the neuroids in each line have to inhibit each other. So far, each neuroid is linked with some but not all of the other neuroids in the net. A third type of linkage, however, has to be introduced which requires global connections. As for the final solution, exactly six units should be excited, the sum of all units is computed, and the value 6 is subtracted from this sum. This difference, which for the final solution ought to be zero, is then fed back with negative sign to all 36 units. These last connections can be interpreted as forming for each unit, a negative feedback system, the reference input of which is zero.

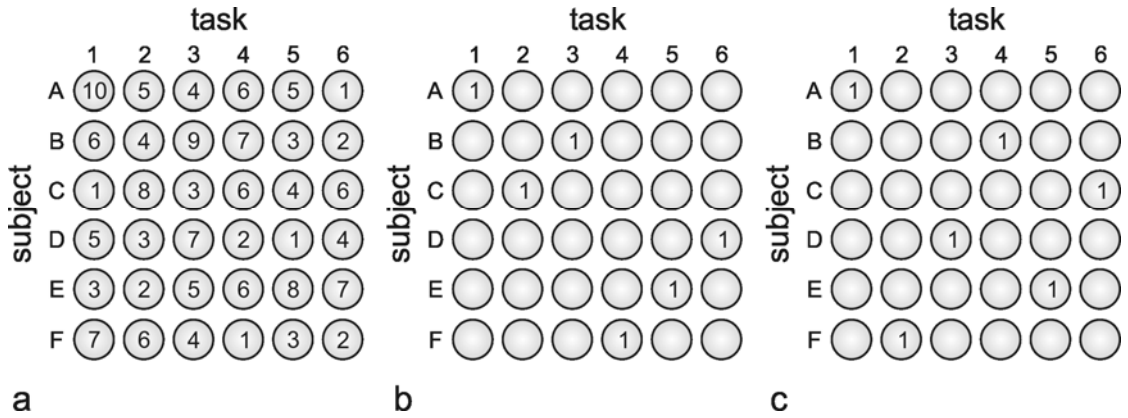


Fig. 11.**4 Six subjects A - F have to perform six tasks 1-6.** To solve the optimization problem, 6 x 6 units are arranged in six lines, one for each subject (a). The vertical columns correspond to the tasks to be performed. The number in each unit shows a measure for the qualification of each subject and each task. These numbers represent the input values. The units are connected such that the six units of a line as well as the six units of a column form winner-take-all nets. Furthermore, an additional unit computes the sum of the activations of all these 36 units (as shown in Fig. 11.**3b**) including an offset value. This sum is fed back to all 36 units. (b) An almost optimal solution found by this network. The sum of the qualification measures is 40 in this case. The optimal solution is shown in (c) with a total qualification of 44 (Tank and Hopfield 1987)

Each of the 720 possible solutions represents a minimum of the energy function of the net. This energy function is defined in a 36 dimensional space. (In the example of Figure 11.**1** we had an energy function defined in a two-dimensional space with two minima.) Taking into account the known qualities of each subject, the problem consists of finding the combination that ensures the highest overall quality, i.e. the largest possible sum of selected qualities. To achieve this, the individual quality values are given as an input to the 36 neuroids. This influences the energy plane of the net in such a

way that the lower the minimum of individual solutions, the better is the corresponding solution. The net then selects, either the best or one of the best solutions. Figure 11.**4b** shows an almost optimal, and Fig. 11.**4c** by comparison, the best, solution to this problem.

However, it should be mentioned that the Hopfield net is not generally appropriate for the solution of optimization problems when the number of neuroids increases, because the number of local minima increases exponentially. There are methods which ensure that the net is not so easily caught in a local minimum. In principle, this is achieved by introducing noise on the excitation of the individual neuroids. In this way, the system is also capable of moving to a higher energy level and thus of eventually escaping from a local minimum. Usually, at the beginning the noise amplitude is chosen to be large, but decreasing with time, thus corresponding to a decreasing temperature. Therefore, this procedure is eventually called “simulated annealing”. Noise can be introduced by simply adding a noisy signal on the neuroid before the activation function is applied. An alternative method is to use probabilistic output units. These can take either 0 or 1 as output value, and the input value determines the probability, whether the output will be 0 or 1. This probability is usually described by the logistic function (Fig. 5.**11b).** Such a probabilistic system is called a Boltzmann machine. This will not be discussed here (for more details see e. g., Rumelhart and McClelland 1986). Yet another possibility to introduce noise is to use asynchronous updating with randomly selecting the unit to be updated.

At least for networks with a sufficiently small number of units and appropriately chosen cost function (represented by the proper weight distribution), recurrent nets can be used to search for optimal solutions.

11.4 Associative Storage

Apart from pattern generation and optimization, the most frequently discussed application of recurrent nets is their use in information storage. According to the distribution of weights several patterns can be stored. As with feedforward nets, autoassociation, (i, e., where the input pattern also occurs at the output) as well as heteroassociation can be performed with recurrent nets. In the latter case, a pattern occurs at the output which differs from that at the input. The net can also be used as classifier, which can be regarded as a special case of heteroassociation. We use the term classification, if only one of the neuroids of the output pattern takes value 1 and the remaining (n - 1) ones take value zero.

Generally, the storage constructed on the basis of recurrent nets is endowed with the same properties described earlier for feedforward nets. They are error-tolerant, since a stable output pattern corresponds to an energy minimum. For this reason, the attractor is also reached, if the input pattern does not correspond exactly to the correct pattern. The domain of an attractor could thus also be defined as a domain of tolerance or of generalization. Error tolerance also includes internal errors, e.g. the failure of individual weights or of a whole neuroid. This storage is also, of course, content addressable. Figure 11.**5** provides an impressive representation of the properties of these nets. 20 patterns were stored in this Hopfield net, which consists of 400 neuroids with Heaviside activation functions. Figure 11.**5b** shows that after three time steps, one of these patterns is fully completed, even though only a relatively small part of this pattern was given as input. The same applies, when a noisy version of the original pattern is used as input pattern (Fig. 11.**5c**). Again, these examples demonstrate that, compared to a feedforward net, a recurrent net needs to perform a number of iterative steps, and thus a certain amount of time will elapse before it reaches its attractor. However, this apparent disadvantage can also be considered an advantage, as will be shown in the next section.

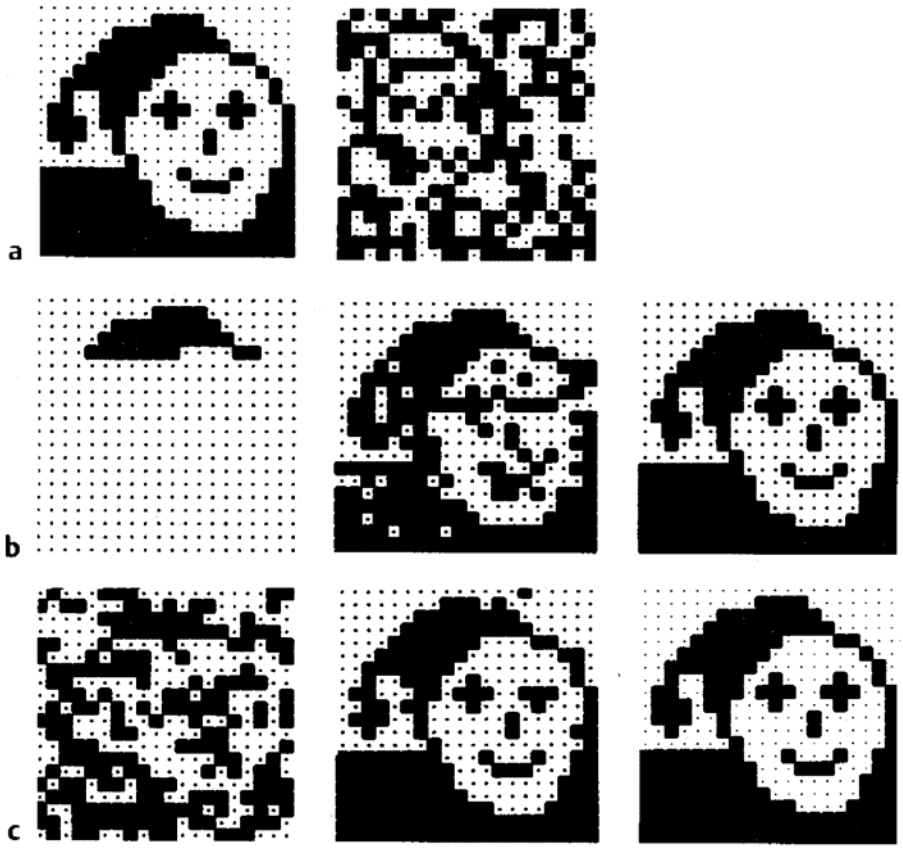


Fig. 11.**5 Pattern completion.** (a) The Hopfield net consists of 20 x 20 neuroids, which can show two states, illustrated by a dot or a black square, respectively. The weights are chosen to store 20 different patterns; one is represented by the face, the other 19 by different random dot patterns. (b) After providing only a part of the face pattern as input (left), in the next iteration cycle the essential elements of the final pattern can already be recognized (center), and the pattern is completed two cycles later (right). (c) In this example, the complete pattern was used as input, but was disturbed by noise beforehand (left). Again, after one iteration cycle the errors are nearly corrected (center), and the pattern is complete after the second iteration (right) (Ritter et al. 1992)

As mentioned, Hopfield nets are usually considered autoassociators. They could, however, also be used for heteroassociation. If one uses as input only a part of the complete pattern, for examle only the upper half of Figure 11.**5**, after relaxation the complete pattern will appear. Thus, the lower half of the pattern could be interpreted as the reponse to the (upper) input pattern.

Like feedforward networks, recurrent nets can be considered as content addressable storage, which has the properties of error tolerance (generalization, graceful degradation).

11.5 Elman Networks and Jordan Networks

A crucial disadvantage of feedforward nets has to do with their purely static character. A particular input pattern produces a particular output pattern, which does not change over time. Apart from the examples shown in Box 5, recurrent nets, too, have so far been considered in this way. Biological systems, however, are endowed with dynamic properties. The system responds to a particular input stimulus pattern with activities that change over time. As shown above, recurrent nets are in principle endowed with such properties. A particular input pattern is immediately followed by an output pattern,

which changes over time until, in the examples shown, a final, stable state is reached. This property, which at first seems to be an undesirable one, can be utilized to produce time functions at the output. In the following the two, next to the Hopfield net, most popular RNN, namely the Elman net and the Jordan net will briefly be described.

Let us begin with Elman nets as they comprise a simplified version of Jordan nets. In this case, most layers have only feedforward connections, and only one contains specified recurrent connections as proposed by Elman (1990) (Fig. 11.**6a**). The system has an input layer, a hidden layer, and an output layer all of which are connected in a feedforward manner. The hidden layer, however, is not only connected to the output layer but also, in a simple 1 : 1 connection, to a further layer called the context layer. To form recurrent connections, the output of this context layer is also inputted to the hidden layer. Except for these 1 : 1 connections from hidden to context layer, the weights of which are fixed to 1, all other layers may be fully connected and all weights may be modifiable. Therefore, all modifiable weights belong to the feedforward connections of the net. The recurrent connections of the context layer provide the system with a short-term memory; the hidden units do not only observe the actual input but, via the context layer, also obtain information on their own state at the last time step. Since, at a given time step, hidden units have already been influenced by inputs at the earlier time steps, this recurrency comprises a memory which depends on earlier states (though their influence decays with time).

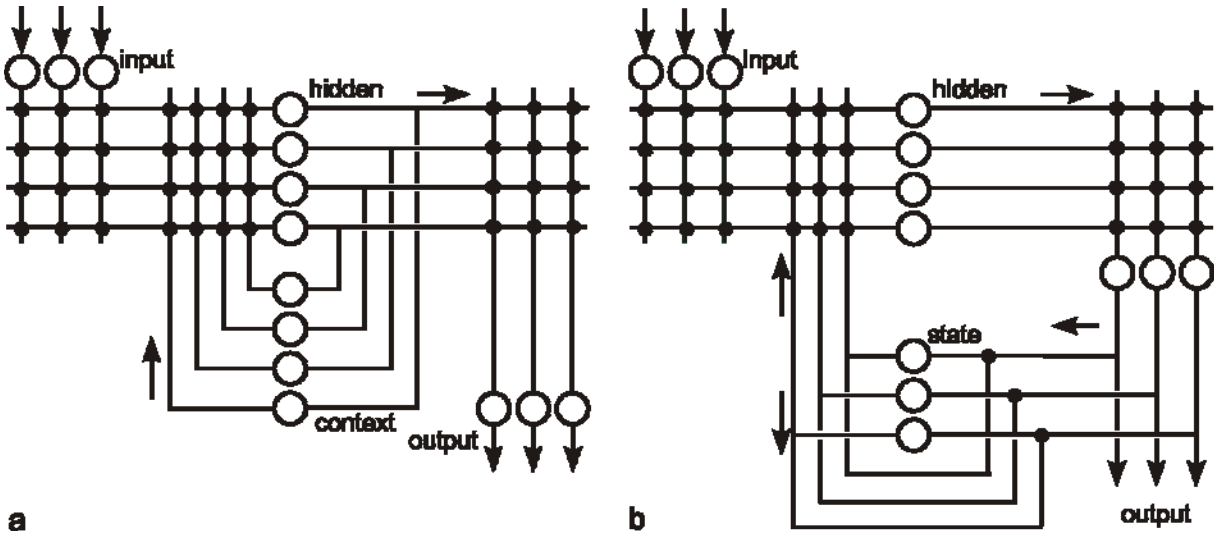


Fig. 11.**6 Two networks with partial recurrent connections.** (a) Elman net (b) Jordan net. Only the weights in the feedforward channels can be modified. The arrows show the direction of the information flow. These networks can be used to learn temporal sequences, i. e., to provide an output vector which depends on the temporal order of the input vectors. Conversely, for a given input vector, a series of output vectors can be produced

As an example, Elman provided such a network with a random sequence of three words of different lengths, namely two, three, or four letters. The net was trained to predict the next letter in the sequence (for explanation of the training procedures see Chapter 12.3). Of course, a high error was observed when the first letter of a new word appeared. However, all following letters were predicted with low error. As the letters included also an end-of-word symbol, the word lenght was also predicted. In a similar experiment, a net was used consisting of 31 units in the input and output layers, and 150 units in the hidden as well as in the context layer. Small sentences consisting of two, three, or four words, selected from a set of 29 words, were given to the net. The complete input series consisted of a sequence of 10000 sentences (27 534 words). The net was trained to predict the subsequent word. After training, the state of the corresponding hidden layer was scrutinized in the following way. All

hidden vectors for a given word were averaged, and, for each input word, these were then compared for similarity by means of a clustering technique. It turned out that different types of words appeared in separate clusters. For example, all verbs were grouped together, as were all nouns. The latter were again found in groups like animats and inanimats, the animats also containing different words describing humans as a subgroup. This means that the internal representation developed a sensible, implicit grouping of the words. This information is not explicitly given to the system, but is learnt only from the context in which the different words appear in the input stream (it should be mentioned that a similar grouping could be obtained using the feature map approach, Ritter and Kohonen 1989). Thus, sequences of letters (words) or sequences of words (sentences) are stored in this net in a distributed manner and can be recognized. Due to this distributed representation it is however not possible to address a specific word in order to be able to form new sentences.

As described, this network can recognize stored input sequences, another example being melodies, and as output can provide a vector, for example the title of the melody. However, such networks can also be used to produce a serially ordered output pattern (a sequence of actions) when a specific input vector is given. Therefore, such recurrent networks are particularly interesting in relation to motor control. For this purpose, Jordan (1986) proposed a very similar net (Fig. 11.**6b**). A difference is that the recurrent connections start from the output, rather than the hidden layer. Furthermore, the layer corresponding to the context, here called state layer, comprises a recurrent net itself with 1:1 connections and fixed weights. Another difference is that the network was used by Jordan so that a constant input vector is given to the net, and the output of the net performs a temporal sequence of vectors. The variation in time is produced by the two types of recurrent connections, namely those from the output layer to the state layer and those within the state layer. For each input vector, another temporal sequence can be produced. A more detailed example is given in Box 6.

As the Elman net and the Jordan net are quite similar, each can be used for both purposes. Both types of networks have the advantage that only weights in the forward connections are modifiable and therefore no special training methods for recurrent nets have to be introduced.

Examples of partially recurrent networks are described. Such networks can be used to learn temporal sequences such that they produce an output vector for a given sequence of input vectors, or, reversely, produce a series of output vectors for a given input vector.

11.6 Nets showing chaotic behaviour

To conclude this Chapter, a final remark should be made concerning the properties of systems with feedback. As already mentioned in Chapter 8, even systems with negative feedback can show instability. In Hopfield nets, unlimited growth is prevented by application of saturation characteristics. If no such limits are given, different behaviours could be observed. For illustration we will consider a simple example containing only one neuroid with a feedback connection (Fig. 11.**7**). If we consider a linear system with positive feedback and a nonzero input, its output will increase infinitely with time. Such positive feedback plays an important role in cooperative behavior, as is found, for example, in ant colonies (Deneubourg and Goss 1989, Beckers et al. 1992, Camazine et al 2003). To limit the growth of activation, instead of a saturation characteristic, the use of high-pass filtered positive feedback was proposed (Cruse et al., 1995). In this case the output values are not confined to a fixed range.

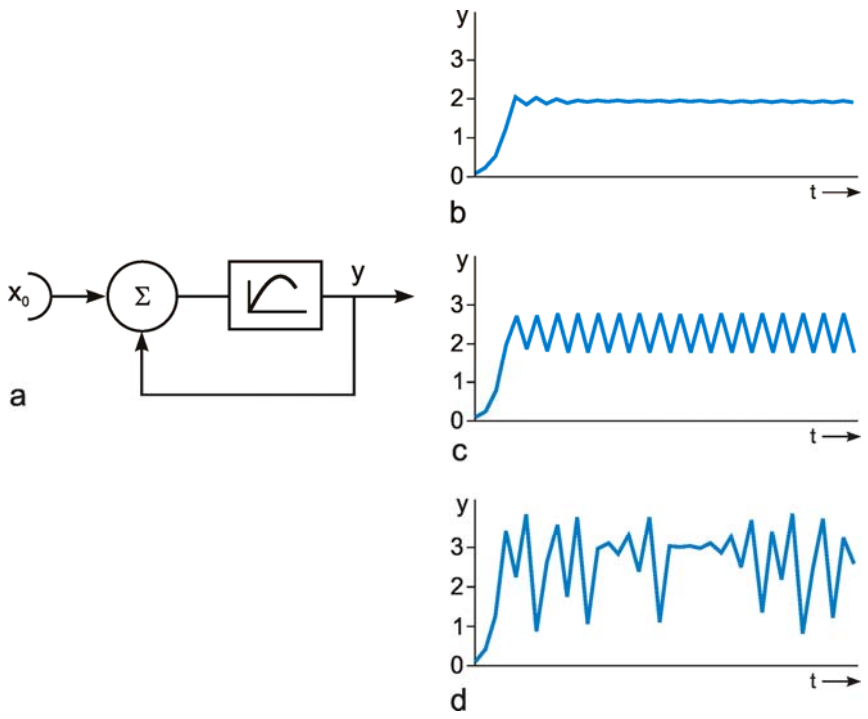


Fig. 11.**7 A recurrent system can show chaotic behavior.** (a) The system containing a nonlinear characteristic of the form $y = a_1x - a_2 x^2$.(b - d) Temporal behavior of the system for different parameters of the nonlinear characteristic. (b) $a_1 = 2.5$, (c) $a_1 = 3.2$, (d) $a_1 = 3.8$. a_2 is 0.2 and the starting value $x_0 = 0.1$ for all cases

A special property can come into play when nonmonotonous characteristics exist, as is shown in the following example. Let us assume that we have a simple system consisting of one neuroid with the nonlinear activation function described by $y = a_1 x - a_2 x^2$ ($a_1 > 1$, $a_2 > 0$) in the loop (Fig. 11.**7a**). This represents an inverted parabola and, therefore, a nonmonotonous function. Let us further assume that the input is nonzero only for a short period. In this case, the output variable approaches a stable value ($a_2/(a_1 - 1)$), if $a_1 < 3$ (Fig. 11.**7b).** The output oscillates between two values for $3 < a_1 < 3.44$ with a period of 2 time steps (Fig. 11.**7c**); it shows more complicated periodicity for $3.45 < a_1 < 3.57$ and exerts chaotic behavior for $a_1 > 3.57$ (Fig. 11.**7d).** Thus, even in this relatively simple case, a feedback system is shown to have highly undesirable properties, unless some kind of random generator is actually required. As this “chaotic” behavior can exactly be computed, it is usually called “deterministic chaos”. The behavior of such a system can be characterized by a “chaotic attractor”. The most interesting property such chaotic systems is the following. A small disturbance produces small immediate effects. However, in the long run the effect of even the smallest disturbance will be dramatic (this is eventually called “butterfly effect”). As small disturbances cannot be avoided in a realistic system, any physically realized chaotic system has nondeterministic properties. As long as the disturbances are not too large, the system remains within its chaotic attractor. Weather dynamics is a familiar example.

Dynamics of recurrent nets may be described by “point attractors”, as shown in the case of Hopfield nets, by “periodic attractors” (e.g. Fig. 11.7c) or by “chaotic attractors (Fig. 11.7d)

12 Learning

We have so far dealt with the properties of nets whose weights were assumed as given. One important question, of course, is how to find those weight values that ensure that the net is endowed with the desired properties. Below the six most important methods of how to set the weights are given.

12.1 Hebb's Rule

In the learning matrix (Fig. 10.**7**) it could be seen that the weights of each output neuroid correspond exactly to the excitation of their input patterns. How could this weight distribution have been obtained? This can be done as follows: We use the schema of the learning matrix without the subsequent winner-take-all net. We apply the input pattern **x** as usual, but give the desired output pattern **y** directly on the output units (training vector in Figure 12.**1a**). The weights can then be obtained according to the rule $w_{ij} = x_j\ y_i$. This means that the weight is a result of the product of the excitations of the two neuroids involved in the connection. This is in accordance with the learning rule, theoretically postulated by Hebb in 1949, in which the strength of a synapse is assumed to increase when the presynaptic and postsynaptic membranes are simultaneously highly excited. The strength of the synapse does not increase, however, if only one of the two neurons is excited. As a simple example, the Hebb rule might be the basis of classical conditioning or Pavlovian learning (Fig. 12.**1b**). Assume that a connection $w_1 > 0$ exists between the input (unconditioned stimulus, UCS) and the reaction R to form a reflex. The weight between a second input (conditioned stimulus, CS) and the output is assumed to be $w_2 = 0$ at the outset. If the two input signals UCS and CS occur simultaneously, according to the Hebb rule, w_2 increases such that the conditioned stimulus alone can elicit the response R.

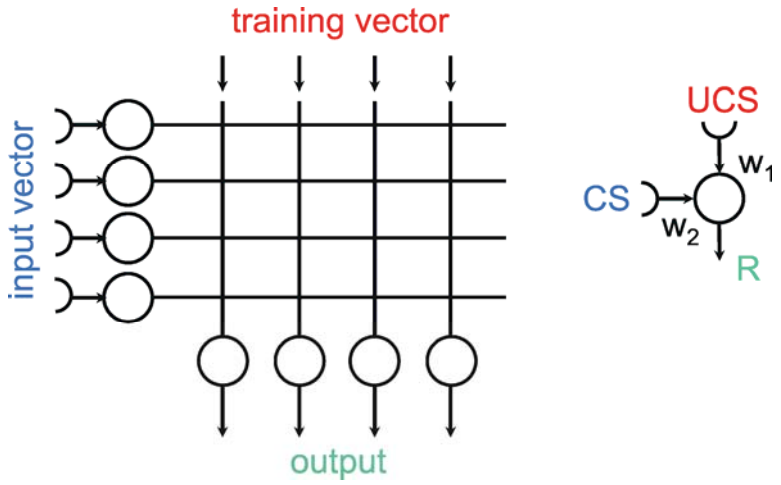


Fig. 12.**1 The weights of a network can be changed according to Hebb's rule, if input vector and training vector (the desired output) are given simultaneously to the net.** This corresponds to the Pavlovian type of learning (or classical conditioning) of a conditioned reflex (b). At the beginning, the weight $w_1 > 0$ connects the unconditioned stimulus (UCS) with a reflex response (R). When the conditioned stimulus (CS) is given simultaneously with UCS, according to Hebb's rule the weight w_2 being zero at the beginning increases and a conditioned reflex is activated

In most cases it is useful to change the weights not in one learning step, but in several smaller steps. Hebb's rule then runs as follows:

$$w_{ij}\ (t+1) = w_{ij}(t) + \varepsilon\ x_j\ y_i\ ,$$

in which the ε parameter, which is called learning rate and takes a value between 0 and 1, determines the step size of the "learning process". If several patterns are to be stored in the same net (Fig. 12.**1a**), the different patterns can be applied successively at the input, and the weights are changed according to Hebb's rule.

If all components of the training vector but one are zero, as is the case in the example of Figure 12.**2a**, each neuroid of the output layer "learns" the weights belonging to one input pattern (that one for which is shows a non zero output). If, in the general case, all components of the output vectors may be different from zero, this learning procedure produces some "crosstalk," i. e., the weights of each neuroid are influenced by all other training vectors. Under what conditions can we obtain a weight distribution which permits several input - output patterns to be stored, i. e., so that after the learning phase is finished, each of these input vectors produces its corresponding output vector without error? With the aid of Hebb's rule, the only patterns that can be stored without "crosstalk" are orthogonal to each other. This means that the scalar product of each pair of input vectors is zero. A simple case of orthogonality is given, for instance, when, in a net of n neuroids, each of the maximally n patterns is characterized by the fact that one neuroid takes value 1 and all others take value zero (see the example given in Figure 12.**2a**).

A disadvantage of Hebb's rule is that, during prolonged learning, the absolute values of the weights tend to increase, which in turn leads to increasing output values when tested. This can be avoided either by assuming that, according to a proposal by Grossberg (see Hertz et al. 1991), the sum of all weights of a neuroid must not exceed a maximum upper limit (which may be a plausible assumption in biological terms, after all) or that all weight changes are superimposed exponential decrease. This would mean that a weight that is rarely increased takes a lower final absolute value than a weight that is continuously growing during learning.

With the help of this method, Hebb's rule can be applied to feedforward nets to produce a heteroassociator. Hebb's rule can also be used for the training of a Hopfield net (Fig. 11.**5**) and, in this way, for producing an autoassociator. The Hopfield net is endowed with the properties of error correction, as described earlier, because all neuroids of a pattern which exhibits the same sign at the output are mutually linked with each other via positive weights. Those neuroids exhibiting different signs at the output are linked via negative weights. This has the effect that, when applying only a part of the complete input pattern, the remaining neuroids are excited via these weights in the same way, so that, after only a few iteration cycles, the entire pattern can be completed. How can the weights be adjusted so as to fulfill exactly the conditions mentioned? Here, too, these conditions are met, if the synaptic weight results from the product of the excitation of the two corresponding neuroids. This can be achieved in the following way: at the input of a Hopfield net that pattern is presented for which the weights are to be adjusted. If, starting with randomly distributed weight values, the weights are changed according to Hebb's rule, two neuroids with negative excitation as well as two neuroids that are positively stimulated, will be linked with positive weight values, while the neuroids that exhibit different signs are coupled with negative weights. In this case, too, the modification of weights is carried out not in one, but several small steps. The weight between neuroid i and neuroid j at the moment of t + 1 is thus calculated such that $w_{ij}(t + 1) = w_{ij}(t) + \varepsilon\, y_i\, y_j$. From this follows that symmetrical weights ($w_{ij} = w_{ji}$) result for the Hopfield net.

An attractive property of Hebb's rule is that it can be interpreted on a local level, because the weight changes due to the strength of the "presynaptic" excitation (Fig. 12.**1b**, neuron CS) and of the "postsynaptic" excitation (Fig. 12.**1b**, neuron R which is excited via neuron UCS). Actually there are biological mechanisms known that correspond to this principle (Kandel et al. 2000). However, there is also a problem, if one wants to apply this mechanisms directly to a classical conditioning procedure.

Assume that there was already some learning such that weight $w_2 > 0$. If now only the CS was presented several times, the weight w_2 would further increase because both neuron CS and neuron R are excited. This property has been termed “runaway effect” by Sejnowski and Tesauro (1989). This effect is, however, in contrast to what happens in the corresponding biological experiment. Presentation of only the CS without an UCS would lead to extinction, i.e. a decrease of the reaction R.

By means of the Hebb rule a two-layered feedforward network can be trained to store different pairs of input - output patterns. To avoid crosstalk between the different patterns, the input vectors have to be orthogonal. The Hebb rule can also be applied to recurrent networks to produce an autoassociator.

12.2 The Delta Rule

If Hebb's rule is applied to a Hopfield net, the mere input of the input vector is sufficient: if a feedforward net is to be trained, the output pattern needs to be given in addition. For this reason, this process is also called supervised learning. One alternative way of training the weights of a net (whether a feedforward net or, as described later, a recurrent net) to produce a given output pattern is to use the so-called delta rule or Widrow-Hoff rule. First, the input pattern **x** is presented. The corresponding output vector **y** is then compared to the desired output vector (or training vector) **t** by forming the difference $\delta_i = (t_i - y_i)$ for each component y_i. The weight between the i-th input neuroid and j-th output neuroid is then changed according to

$$w_{ij}(t+1) = w_{ij}(t) + \varepsilon\, x_j(t_i - y_i)\,.$$

The change of the weight is thus determined by the magnitude and sign of the "error" ($\delta_i = t_i - y_i$).The change also depends on the magnitude of the input signal x_j If the value of the input signal is near zero, the corresponding weight barely changes. In contrast to Hebb's rule, the change in weight ceases, once the value of the training vector has been attained.

Which properties can be imprinted upon a net with the aid of the delta rule? If only a neuroid is considered, whose activation function represents a Heaviside characteristic, the Perceptron-Convergence Theorem states the following (Minsky and Papert 1969): If a weight distribution can be found by which the total number of trained input patterns can be divided into two groups, so that the input patterns of one group produce a subthreshold output value, and those of the other group a suprathreshold output value at the neuroid considered, a corresponding weight distribution can actually be obtained with the delta rule in a finite number of learning steps. Such a division into two groups is possible if the two sets of patterns are linearly separable (see Chapter 10.**7**).

If, instead of a single output, the entire net with n input neuroids and m output neuroids is considered, the question arises as to what kinds of input patterns are needed to ensure that they can be stored in the net simultaneously without influencing each other. In contrast to Hebb's rule, which can be used only to store orthogonal patterns, the restrictions applying to the use of the delta rule are less rigorous; with the latter, patterns can be stored without overlap, if they are linearly independent. The maximum number of patterns to be stored does not increase, however. For n neuroids, there are maximally n linearly independent patterns. (If Heaviside characteristics are used, this figure is higher, i. e., maximally 2 n). Examples of orthogonal and linearly independent patterns are given in Figures 12.**2a, b.**

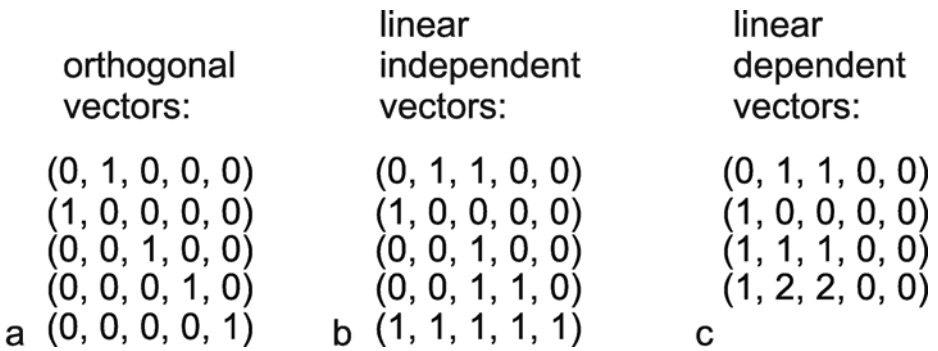


Fig. 12.**2 Examples of five orthogonal vectors (a), five linear independent vectors (b), and four linear dependent vectors (c)**

As shown with the simple example of the learning matrix, the adjustment of the offset or threshold values can be very important if a specific behavior of the net is to be obtained. The size of the offset of the neuroid can also be learned by using a simple trick with the help of the learning rules discussed here, although, strictly speaking, they can only change weights. For this purpose a so-called "bias neuroid" is introduced. It is not given an input and exhibits the constant excitation 1. It is linked to all neuroids of the net. The corresponding weights can, of course, also be adjusted by applying the learning rules. The bias neuroid has the effect that for each neuroid the value of the corresponding weight is added. This effect is the same as if the offset of the activation function was shifted. If this is a threshold, it would represent a shift in the threshold value.

The delta rule can also be used to train recurrent nets. For this, one iteration cycle is performed after presentation of the input pattern; subsequently, the results are compared with the training pattern and the delta rule is applied. The operation will be successful, if the desired mapping is "legal," i, e., if the condition of linear separability is given for each neuroid at the output. (If this condition is not met, the mapping can be legalized by an introduction of additional neuroids, as described above in relation to the introduction of hidden layers (Levin 1990)). In this way, a recurrent net can also be employed as a heteroassociator. The net shown in Figure 11.**9** was obtained in this way. This shows that, with the help of the delta rule, it is possible to obtain recurrent nets with asymmetrically distributed weights.

Using the delta rule, a two-layered feedforward network or a recurrent network can be trained. The growing of the weights is limited. The input vectors have to be linearly independent to avoid mutual influences. The introduction of a bias neuroid permits the training of threshold values.

12.3 Generalized Delta Rule

In feedforward nets, the application of the delta rule is restricted to two-layered nets. The study of highly parallel systems has been considerably accelerated by the development of methods that can be employed to train multilayered feedforward nets. This method, known as the generalized delta rule or error back propagation, merely presupposes that the activation function f(u) of the neuroids are differentiable. This excludes characteristics with discontinuous slopes like the Heaviside function. But the latter can, of course, be approximated by the frequently used sigmoid (or logistic, or Fermi-) function $f(u)= 1/(1 + e^{-ax})$. To begin with, as with the delta rule, the error $\delta_i = (t_i – y_i)$ is determined for each neuroid after the response to an input pattern has been determined. This error is then multiplied by the derivative of the value of the activation function $f'(u_i)$. Similarly to the delta rule, the weights of

the neuroids of the output layer change such that

$$w_{ij}(t + 1) = w_{ij}(t) + \varepsilon\, y_j\, \delta_i\, f'(u_i)\,.$$

For neuroids of the higher layers, the δ error is computed as the sum of all weighted errors of the following neuroids. With respect to the weight w_{42}, i.e., the weight between neuroid 2 and 4 in Figure 12.**3**, this means:

$$\delta_4 = (t_6 - y_6)\, w_{64}\, f'(u_6) + (t_7 - y_7)\, w_{74}\, f'(u_7)\,.$$

The error is thus computed back across the net (error back propagation). Therefore, the change of the weight w_{42} is determined by

$$\Delta w_{42} = \varepsilon\, y_2 \delta_4\, f'(u_4)\,.$$

The logistic function y = f(u) is usually used, because it is endowed with an easy-to-compute derivative: y' = y(1 - y). Although this was not specifically mentioned, the logistic function was applied in the network of Figure 10.**9b**. Unlike the simple delta rule, with the help of the generalized delta rule, even those patterns can be learned which are linearly dependent (see Figure 12.**2c** for an example of linearly dependent patterns).

The properties of the delta rule can be illustrated as follows. For a given pair of input and training patterns, the total error produced by a given weight distribution can be defined as

$$E = \sum_i (t_i - y_i)^2\,,$$

i. e., the sum of the squares of all errors produced by all output units. The aim of the learning procedure is to minimize this error value. As the error depends on the values of all k weights, the situation can be described as a function in a k-dimensional space. The goal is to find the minimum of the error function in this k-dimensional space. The delta rule is a gradient descent procedure. The gradient descent changes the weights in such away that the error is continuously reduced. The procedure starts at a position in the k-dimensional space given by randomly chosen initial values of the weights. During this gradient descent, there is, of course, a risk that the procedure gets stuck in a local minimum so that the optimum learning performance is not realized. This risk can be reduced by increasing the number of hidden units. This increases the dimension of the space and will in turn reduce the incidence of local minima because the introduction of an additional dimension may open a way out of the minimum. A further method consists in superimposing random values on the weights during the learning process. This can provide the possibility of moving against the gradient and thus to overcome local minima; during training the amplitude of this noise should continually decrease. This procedure is eventually called simulated annealing.

Another simple and often used method is the introduction of the momentum

$$\Delta w(t + 1) = \Delta w(t) + \eta \Delta w(t - 1)\,.$$

This means the direction of descent does not only depend on the actual gradient but also to some extent, given by the factor η, on the direction and size of the last step. Thus, some kind of low-pass filter-like inertia is introduced which might also help to avoid local minima.

The time to approach the minimum can be shortened by application of the RPROP algorithm (Riedmiller and Braun 1993): for each weight, the amount of the weight change increases by a given factor, if the gradient does not change its sign. If, however, the gradient has changed its sign, which means that the learning algorithm has just jumped over a minimum, the weight change is decreased by a given factor and its sign is changed, too.

To illustrate why the generalized delta rule requires a differentiable activation function, we have to consider the error function as a function of the weights. Considering weight w_{74} in Figure 12.**3** as a simple example, the error function is defined as $E(w_{74}) = (t_7 - y_7)^2 = (t_7 - f(u_7))^2 = (t_7 - f(y_4\ w_{74}))^2$. Calculation of the gradient means to determine the first derivative of E with respect to the weights. According to the chain rule, we obtain

$$dE/dw_{74} = -2\,(t_7 - f(y_4 w_{74}))\,f'(u_7) = -2\,(t_7 - f(y_4\ w_{74}))\,f'(y_4\ w_{74})\,y_4\ .$$

If the logistic or a similar function is used, the derivative has the property of being near zero for very small and very high input values, but high for values around the middle range. This could be interpreted as the weights do not change if the unit has "decided" to assume a very high or very low excitation, whereas learning is more probable as long as no such decision yet has been made.

Whereas the number of input and output units is given by the task for which the network is designed, it is an open question how many hidden units should be used. Generally, a given mapping can be approximated better, the more hidden units exist in the net. On the other hand, however, too many hidden units destroy the capability of generalization; the net learns the exact training patterns only and may not appropriately respond to other, though similar input patterns. This is called "overtraining, or overfitting." It is still an unsolved question what structure of a network (number of units in a layer, number of layers) is best suited for a given problem. One method to improve, i.e., simplify, the structure of a network, which has already been shown to solve the required task, is to look for those neuroids which have the smallest weight values, delete them and try to retrain this new network with the same task. By this "pruning," one may find a simpler version of the network. (For alternative methods, see Chapter 12.5).

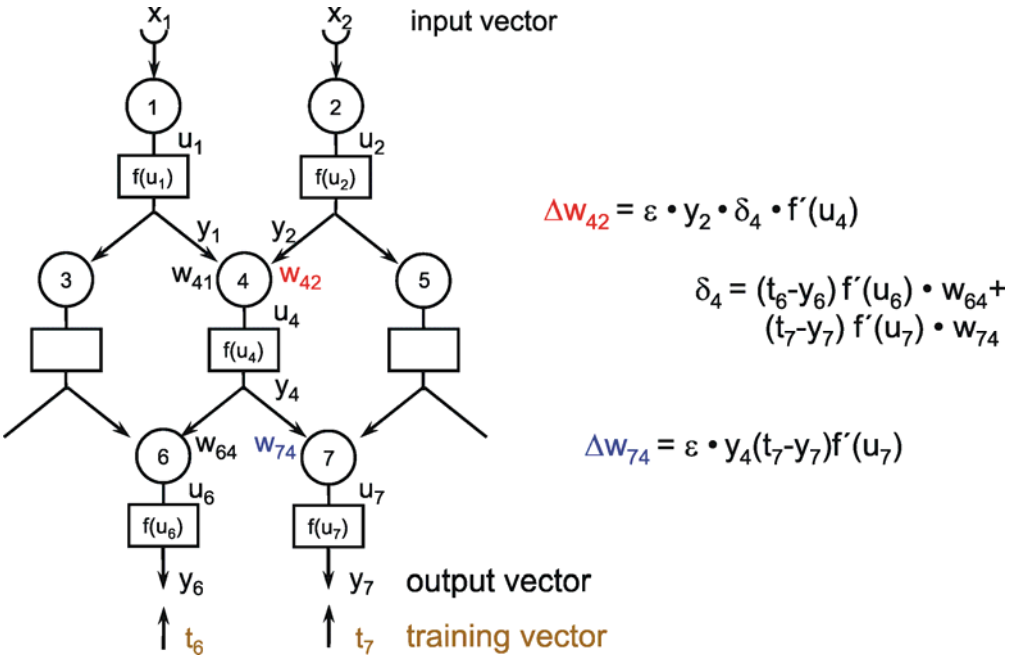


Fig. 12 **.3 A simple, three-layer network for the illustration of the generalized delta rule.** As examples, the formulae of how to change weights w_{42} and w_{74} during learning are given

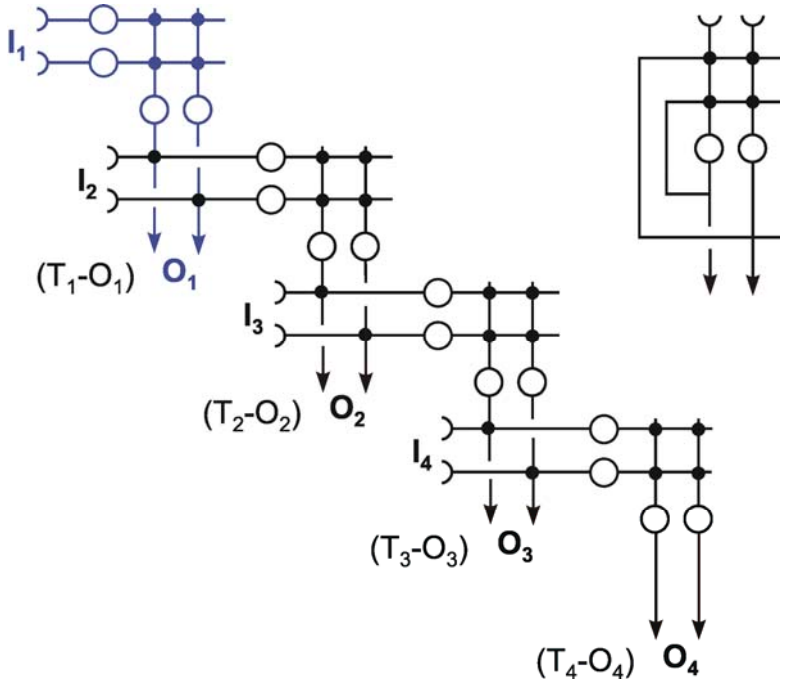


Fig. 12.4 Unfolding a recurrent network in time. To cope with the dynamics of a recurrent network, the net is copied several times, and these copies are connected to form a feedforward net. Each subnet represents another time step. Errors (T_i-O_i) have to be collected for each time step, i.e. for each subnet, and are then used to update the weights

Finally, it should be mentioned that the generalized delta rule can also be used for recurrent nets, as discussed above in connection with the simple delta rule. However, there is the problem that the dynamics of the recurrent net as such are superimposed to the dynamics due to the weight changes during learning. This problem is avoided by "unfolding" the recurrent network into several identical feedforward nets (Fig. 12.**4**). Each net in the spatial order corresponds to another iteration in time ("unfolding in time"). This means that the backpropagation method can be applied in such a way that, after having given the input to the uppermost net, the output of each net in the series has to be compared with the desired output expected in the corresponding time slot (backpropagation through time, BBTT, Rumelhart et al. 1986). All errors for a given weight are collected and used to train this weight (for details see Hertz et al. 1991). Different to the backpropagation method applied to feedforward networks, BBTT by itself may show complex dynamic behavior and does not necessarily lead to a minimum.

To train networks with more than two layers, the generalized delta rule or error back propagation can be used. This requires differentiable activation functions. Crosstalk can be avoided also for linear dependent input vectors. Several methods are available to avoid getting stuck in a local minimum during the learning process. How to choose the appropriate numbers of hidden units and layers is a still unresolved question.

12.4 Teacher Forcing and Learning with Suppression Units

A flaw of the generalized delta rule is that it is hard to believe that learning in biological systems is actually based on this principle. One problem being that additional networks are necessary to compute the error and transport this information to the corresponding synapses. The other problem concerns the determination of the error as such. How could the single error components be determined and provided to the backpropagation network? Furthermore, the training methods described above for recurrent nets are used off-line, i.e. the recurrent connections are switched off during learning to avoid the problems of superimposition of two dynamics, the dynamics of the recurrent net itself and the dynamics due the changing of the weights (Steil 1999). In the following a circuit is presented that copes with this problem. This circuit acts within a neuronal unit and incorporates a learning rule that formally corresponds to the delta rule, but does not require a separate network for backpropagating the error because the latter is computed locally within each single neuroid. The units, called suppression units (Su-units, Kühn et al. 2007, Makarov et al., 2008) contain a simple circuit that allows to disconnect the dynamics of the recurrent network from the dynamics due to the learning procedure. Such a neuroid is equipped with two types of inputs: One external input a_i and n internal, i.e., recurrent input connections x_j, the weights w_{ij} of which can be changed by learning. The overall output of one suppression neuroid is calculated in the following way:

$$x_i(t+1) = \begin{cases} s_i(t), & a_i = 0 \\ a_i, & a_i \neq 0 \end{cases} \qquad with\ s_i(t) = \sum_{j01}^{n} w_{ij}\ x_j(t) \quad \text{for i = 1 to n.} \qquad (1)$$

In other words, an external input a_i suppresses the recurrent connections if $a_i \neq 0$. This is a simple way to achieve that the recurrent connections are interrupted when an external signal is applied. (As in this implementation input of zero represents “no input”, zero input values are excluded. An easy way to solve this problem is to apply spatially coded networks, see Chapter 15). How such units can be used will be illustrated using a simple task first: The network should represent a fixed static pattern given at the input and produce sustained output activity even if the input pattern disappears. Specifically, the task is as follows: the net consists of n recurrently connected units. As example, a net of n = 3 units is depicted in Figure 12.**5a**. Any n-dimensional input vector **a** is provided to the net. The weights should be arranged in a way that all units adopt activations that correspond to the input and maintain their activation even after the external input is switched off. In other words, the task is to find a weight matrix **W** with **a**= **W a**. As we have n^2 weights, there is a manifold of matrices that fulfill this condition. n equations determine n degrees of freedom. Therefore (n^2 – n) of the n^2 weights can be chosen arbitrarily. For n = 3 as an example one possible solution is given by

$$1/(a_1 + a_2 + a_3) \begin{pmatrix} a_1 & a_1 & a_1 \\ a_2 & a_2 & a_2 \\ a_3 & a_3 & a_3 \end{pmatrix} \qquad (2)$$

Another solution is $w_{ij} = a_i a_j \big/ (a_1^2 + a_2^2 + a_3^2)$ (3)

Attractors of this network are all points of the line connecting (0,0,0) and (a_1, a_2, a_3), or $x_1 = \lambda\ a_1$, $x_2 = \lambda\ a_2$ and $x_3 = \lambda\ a_3$ for any real λ.

Another task is that more that one vector, for example two vectors a, b, are learnt by the net. In this case the net can be described as forming an attractor consisting of a two-dimensional subspace that is described by the plane $B_1\ x_1 + B_2\ x_2 + B_3\ x_3 = 0$, if a, b fulfill this so called basic equation. These nets

are only neutrally stable, which means that if any weight is changed arbitrarily, the activations of the units increase infinitely or decrease to zero. Therefore, a learning mechanism is necessary to stabilize the weights against disturbances.

In order to achieve this goal, the weights can be changed according to the following algorithm:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij} \qquad \text{with} \qquad \Delta w_{ij} = \varepsilon \cdot x_j(t) \cdot \delta_i(t) \tag{2}$$

with $\varepsilon > 0$ being the learning rate and $\delta_i(t)=a_i(t)-s_i(t)$. This corresponds to the delta rule which however is now applied within the neuroid. As the output of the neuroid is strictly determined by the external input, this is eventually called “teacher forcing” (Williams and Zipser 1989). During the training procedure, the weights stabilise at values that in the end we have $\delta_i(t)=0$ which means that the summed recurrent input s_i equals the external input a_i. After learning is completed, and the summed internal input equals the external input, the latter can be switched off without changing the activation of the net. However, learning has to be switched off, too, because otherwise the weights would then learn to represent the zero vector (0,0,0).

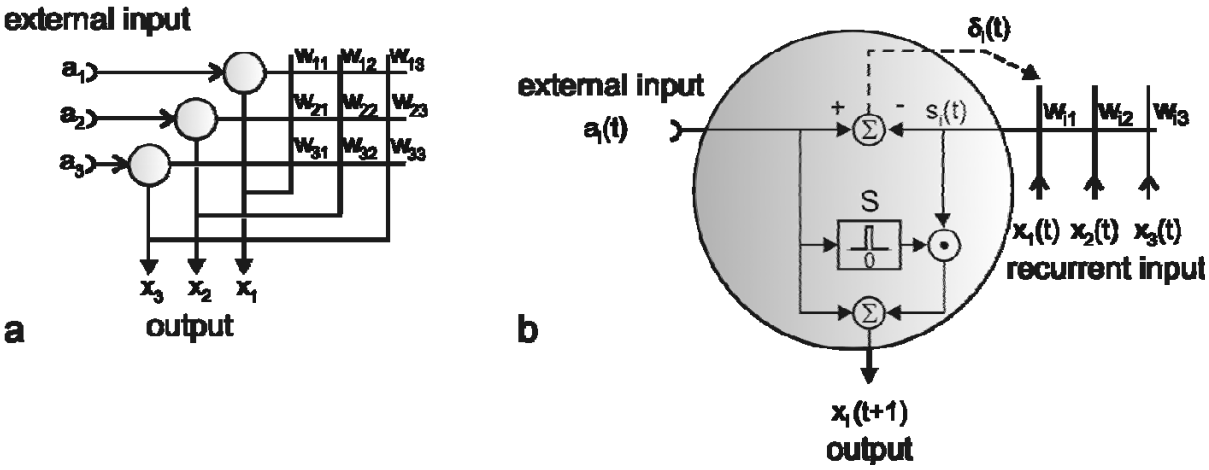


Fig. 12.**5 Recurrent network with Su units**. (a) A network containing 3 units. (b) The circuit characterizing a Suppression unit

For the first task mentioned, i.e. training with one vector a, the general solution for the weight matrix is given by $W = \frac{a\,a^T}{\|a\|^2}$, when starting learning with all weights being zero (Kühn et al. 2007, Makarov et al. 2008). For n = 3 (Fig. 12.**5a**) this corresponds to the matrix (3) shown above. General solutions for the cases that more than one vector is used for training are given by Makarov et al. (2008). Training vectors might be linearly dependent. The resulting weight matrix is independent of the choice of any linear combination of these vectors. As there are redundant degrees of freedom, some of the weights can be determined arbitrarily during the learning procedure. As an interesting case, the diagonal weights could be set to zero, for example. These diagonal weights influence the relaxation rate (Chapter 14.4), but not the final output. Generally, when testing the trained network with some input values being zero, the output evolves during time until the complete net fulfills the basic equation (pattern completion). Makarov et al. (2008) also investigated the conditions necessary for learning such networks with neuroids containing nonlinear characteristics.

Furthermore, application of this circuit allows to train any dynamic situation that can be described by linear differential equations (see Chapter 14.3) and for a broad class of nonlinear differential equations. To learn dynamic situations, the learning rule has to reflect the temporal sequence of the input vectors:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij} \quad \text{with} \quad \Delta w_{ij} = \varepsilon\, x_j(t-1)\, \delta_i(t) \qquad (4)$$

Two cases have to be distinguished. The length of the training period, i.e. the number of training vectors, p is smaller or equal to the number of neuroids ($p \leq n$) or $p > n$. In the former case, the net can store a temporal sequence of up to n vectors. The case $p > n$ is relevant for learning networks that represent differential equations (Chapter 14.3), for example (for details see Makarov et al. 2008).

Below, two further learning procedures will be presented, which also come closer to biological reality than the classical backpropagation method, namely the reinforcement method and the method of genetic algorithms. Both start from the assumption that feedback from the environment does not mean that the correct output pattern is given and then compared to the actual output; rather, what is implied is that the environment merely supplies a scalar quantity, e. g., a value between zero and one, comparable, for instance, to a quantitative judgment concerning the answer varying between “good” and “bad”.

> Application of suppression units allows online training of recurrent networks because they permit to separate the dynamics of the recurrent net from the dynamics due to the weight changes.

12.5 Reinforcement Learning

With the exception of the autoassociator trained by Hebb's rule or the IC algorithm, the net has so far always been informed of what the correct answer should be. In this context, learning is based on the evaluation of the difference between the response of the system, presented by the net to a specific input vector, and the actually correct answer, given by the trainer. Thus, an error signal can be determined for each component. These methods are described as supervised learning. Most often, however, an organism is not informed of the right answer in a specific case, and feedback merely consists of a scalar quantity (e. g., on a scale good-bad). If a ball is thrown at a target, for example, there is no feedback on the correct control of individual muscles, only on whether it hit or missed the target. This would be a case for reinforcement learning. Since feedback to the system consists only of information about whether the action was good or bad, but not about what the correct action would have been, we talk of a critic here, rather than of a trainer. In a simple case, the feedback is immediately provided by the environment. The more complicated case, when the feedback occurs only after several actions have been performed, will be treated later.

The input components x_i, describing the external situation, and the components of the output vector **y** may be positive real numbers (Fig. 12.**6a**), although in many examples, the neuroids of the output layer are endowed with a Heaviside activation function and can take the values 0 or 1. The system is presented with a number of input vectors. The task is to learn reactions to these inputs that will be rewarded as highly as possible. The net initially possesses a random distribution of the weights w_{ij}, and hence responds to an input excitation by reaction **y**, which is presumably not very sensible. The environment ("world") evaluates the system's reaction and responds by giving a reward r. The aim is that the weights w_{ij} within the systems should be changed in such a way that the reward r for the various input vectors is as high as possible (this corresponds to the situation in which the extreme value of a quantity has to be found.)

This task can be split into two problems, namely, first, to find an action **y** which provides a high reward, and, second, to find the weights for which, for a given input **x**, i.e. a given situation, the system produces the output **y**. The latter corresponds to the paradigm of supervised learning and, therefore, can be solved as described above by application of the delta rule or, if the activation functions are differentiable, e. g. logistic functions, by the generalized delta rule. The first task is the actual problem of reinforcement learning.

To begin with the simplest case, we assume that the world behaves deterministically. This means that a definite change of the action produces a predictable change in the reward. Let us further assume that the same input vector is presented several times in a row, i. e., that the environmental situation remains unchanged. The change in reward can then be compared for two successive actions. At moment t - 1, the system is informed of input **x**(t - 1), reacts by **y**(t - 1), and receives reward r(t - 1). A corresponding situation occurs at moment t. In order to search for a better action, $y_i(t)$ is changed compared to y_i (t - 1) by means of an addition of noise. The effect of this changed action can be quantified by multiplying it with the change in reward:

$$\Delta y_i = [r(t) - r(t-1)]\,[y_i(t) - y_i(t-1)]$$

This value Δy_i can then be used to solve the second task, namely to calculate the weight changes of the net. If this net contains several layers, the generalized delta rule can be applied. To simplify matters, it will be shown here for a two-layered net:

$$\Delta w_{ij} = \alpha \Delta y_i \; f'(u_i)\, x_j$$ (see Chapter 12.3).

Δy_i thus corresponds to the error signal of the delta rule. For a linear activation function, this means

$$\Delta w_{ij} = \alpha \Delta y_i x_i \; .$$

The net can then be trained by changing the weights in the following way (see Fig. 12.**6**) (Barto Sutton Brouwer 1981):

$$w_{ij}(t+1) = w_{ij}(t) + \alpha \;\; [r(t) - r(t-1)]\,[y_i(t) - y_i(t-1)]\, x_j(t) \; . \qquad (1)$$

On the basis of this learning rule, weight w_{ij} changes only if the activity of neuroid i changes, i. e., $\Delta y_i = y_i(t) - y_i(t-1) \neq 0$, and if the reward changes. The weight increases if the state of activity and the corresponding reward change in the same way. Otherwise the weight decreases. As a consequence, the rewarded behavior will follow this input vector with greater probability. If we introduce Heaviside activation functions, only the sign of the difference $y_i(t) - y_i(t-1)$ needs to be used in equation (1) in order to obtain a Δy_i of ± 1 or 0.

Stochastic properties might occur in the world such that the reward is noisy. In this case, a direct comparison with the action and the reward at time t - 1 may not be as good. Instead, the comparison with the mean values of earlier actions and rewards can lead to greater improvement of the weights.

In some particular cases, it may happen that r is scaled in such a way that a baseline or reference value r_o (e.g., = 0) can be given so that above r_o, r can be considered a positive reward and below it, a negative reward. The value $(r - r_o)$ then has the property of being a positive or a negative number. In this case, the comparison with the baseline value is possible, of course, and actually leads to faster learning (Williams 1992).

In principle, the reinforcement learning procedure corresponds to the operand conditioning (or instrumental learning) found in biological experiments. A rat in a Skinner box accidentally (see "noise" in Figure 12.**6a**) performs an act e. g., pushes a lever. If the lever is connected to a rewarding mechanism, there is greater probability that this behavior will be performed again.

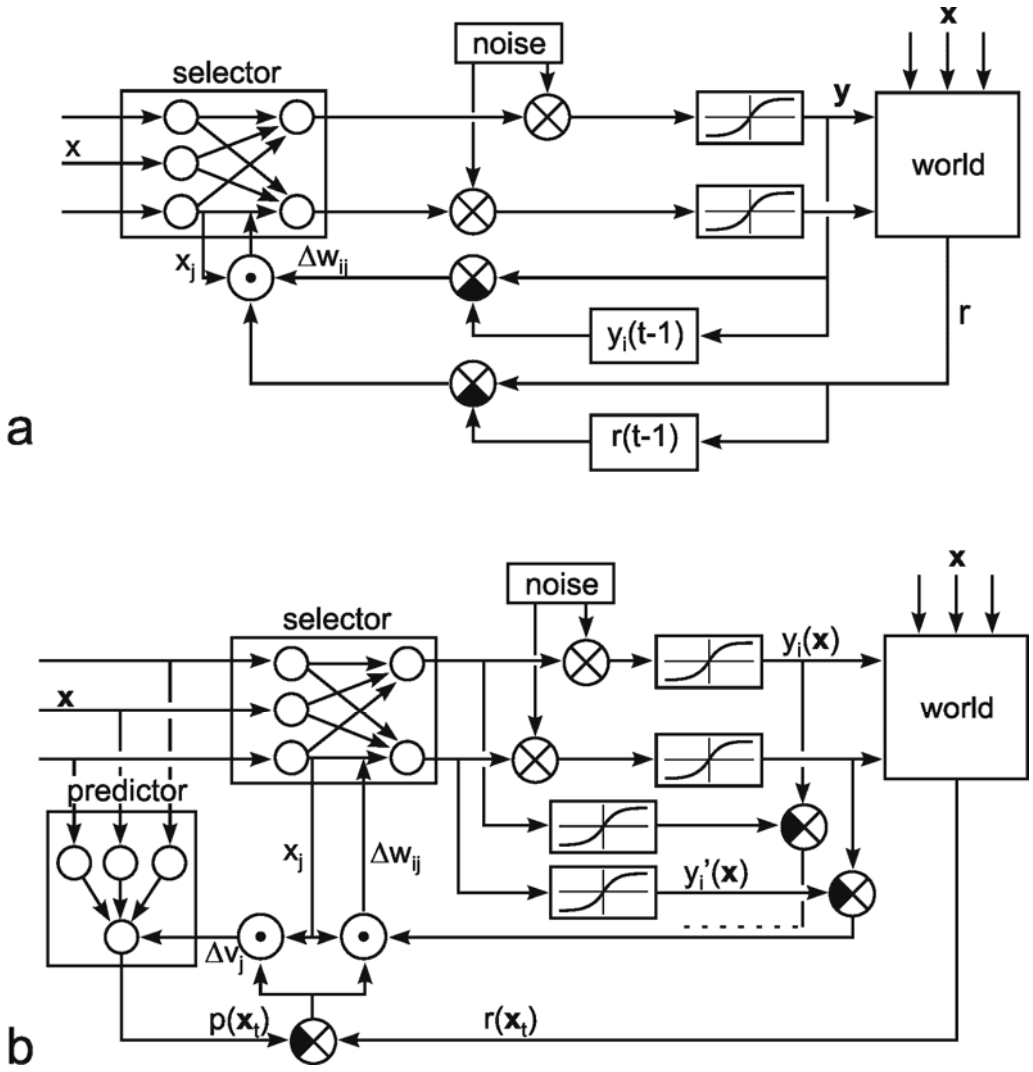


Fig. 12.**6 Reinforcement learning.** (a) The input vector **x** is given to the network, which, in this example, consists of two layers, but might be more complicated. The response of the net is somewhat changed by means of a noise produced by the system. The system contains a memory to store the components of the output vector applied during the previous time step [$y_i(t-1)$]. Furthermore, the system receives a scalar reward signal from the environment (world) and stores the previous reward value r(t-1). With these values, the appropriate change of the weights w_{ij} can be determined as shown in the lower part of the diagram. (b) If the input vector varies, the application of an additional "predictor net" can solve the problem. In this case, the actual reward is compared with the expected reward p provided by the predictor net. Here, the output vector is not compared with the previous one, but with the original vector that the network proposed before the vector was changed by means of the noise influence. These values can be used to change the weights of the selector net and of the predictor net, accordingly.

This disadvantage of the method described above is that it presupposes that the input vector (i.e., the environment) does not change or, if it does, perhaps just a little. In case of a sudden change, a comparison with the preceding situation would indeed not make much sense. In such a situation, a different reward is attributed to an action **y** depending on the input vector **x**. To solve this problem, an additional net is introduced which provides the reward to be expected for a given input according to

the earlier experience. This net is called predictor net because it provides the predicted reward p(**x**) for a given input **x** (Fig. 12.**6b**). If we had an ideal predictor, the value r($\mathbf{x}_t$) could be compared with the value p belonging to a given input vector. How can we obtain this net? The predictor net comprises a simple world model, so to speak, which computes the reward to be expected in connection with a specific input signal by $p(\mathbf{x}_t)=\Sigma\ v_j\ x_j(t)$, v_i being the weights of the net. This predictor net is simple to train since the reward p predicted by the predictor net can be compared to the actual reward r, thus enabling a direct application of the delta rule:

$$v_j(t+1) = v_j(t) + \alpha_1\ [r(\mathbf{x}_t) - p(\mathbf{x}_t)]\ x_j(t) \qquad (2a)$$

If a multilayered predictor net is used, the generalized delta rule could be applied accordingly.

Using the predictor net, the synapses of the net itself, sometimes called the selector net, can be trained, as above:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha_2\ [r(\mathbf{x}_t) - p(\mathbf{x}_t)]\ [y_i(\mathbf{x}_t) - y_i'(\mathbf{x}_t)]x_j(t) \qquad (2\ b)$$

y' is the action proposed by the selector. It is not possible to train only the predictor net because the reward depends on the actions y. Therefore, both nets have to be trained in parallel. The better the predictor net becomes in the course of learning, the smaller the error (r - p) will be, such that the predictor net, and thus the net itself, will move toward a stable final state.

In the simple version illustrated by equation (1) the system requires a storage for the earlier reward r(t - 1) and for the components of the earlier action y_i(t-1). Thus, if the output vector has n components, storage for n + 1 values is necessary, which stores the information from the last time step. By contrast, the system described by equation (2) and Figure 12.**6b** requires no such storage.

TD learning: Where the net acts in a deterministic world, i. e., a world that responds to a particular reaction **y** of the net with a particular environmental situation **x**, time series can also be learned in the same way. Pure reinforcement learning can, however, only function in a simple world. But a case may arise in which, in a specific situation **x**, a reaction **y** may be reinforced (and hence learned) even though the system, through action **x**, is confronted with a situation in which it may be difficult to perform new, reinforced actions. For example, if an animal eats up all available food, this may be desirable for the moment. But a decision to exercise temporary restraint may be preferable in the long term, since this might possibly prevent starvation. It would be better, therefore, instead of a momentary reinforcement r (t), to consider the reinforcement r_∞(t), which represents a combination of all further reinforcements to be obtained in this series of actions. In order to include to long-term effects of actions, TD (temporal difference) methods have been introduced by Barto et al. (1981) (Fig. 12.**7**). Assume that for each situation (input **x**) a number of actions are possible. The selector net chooses one of these actions. The function **y** = f(**x**), which describes which action is selected for a given situation, is called the policy or strategy of the system and is represented by the properties of the selector net. The reward r(t) depends on the situation $\mathbf{x}_t$ and the action $\mathbf{y}_t$ = f($\mathbf{x}_t$) chosen, and is therefore written as r($\mathbf{x}_t$, $\mathbf{y}_t$). As actions in the far future may be less important, for example, because they can be less exactly foreseen, they are given smaller weights. Barto et al. proposed to determine

$$r_\infty(t) = r(t) + \gamma r(t+1) + \gamma^2 r(t+2) + ... + \gamma^n r\ (t+n) + ... \qquad (3a)$$

with $\gamma < 1$ which produces smaller weights for more distant rewards. This can also be written as

$$r_\infty(t) = r(t) + \gamma\ r_\infty(t+1)\ . \qquad (3b)$$

Equations (3) hold for an unpredictable world, i. e., the reward may be influenced by noise and so may be the reaction of the world, namely the new input $\mathbf{x}_{t+1}$. In order to find a good strategy, in the TD reinforcement approach, the predictor net is replaced by an "evaluator net." For each action, this provides an estimate e(**x**) for the value r∞(**x**). The problem now is hot to learn a sensible evaluator net which can produce the value e(**x**), for a series of actions. Although the selector proposes action **y'**, a new action $\mathbf{y}(\mathbf{x}_t)$ will be tested by means of a new input. This leads to the reward $r(\mathbf{x}_t)$. If this reward is better than that expected for action **y'**, the evaluator net should be changed. According to the earlier case (reinforcement comparison, equation 2), this would be done by directly comparing the reward $r(\mathbf{x}_t)$ with the predicted reward $p(\mathbf{x}_t)$. As in TD learning future rewards should also be taken into account, so the expected value includes not only the actual reward $r(\mathbf{x}_t)$, but also the (weighted) rewards expected in future consecutive actions. As the expectation value e is an approximation to $r_\infty(t)$, equation (3) holds approximately also for e:

$$e(\mathbf{x}_t) \approx r(\mathbf{x}_t) + \gamma\, e(\mathbf{x}_{t+1}) .$$

Thus, $e(\mathbf{x}_t)$ should not be compared with the actual reward $r(\mathbf{x}_t)$, but with the complete exceptation $r(\mathbf{x}_t) + \gamma\, e(\mathbf{x}_{t+1})$. This leads to an "error" signal

$$\hat{r} = r(\mathbf{x}_t) + \gamma e(\mathbf{x}_t + 1) - e(\mathbf{x}_t).$$

The value $e(\mathbf{x}_{t+1})$ can be obtained by observing the state of the world after action $\mathbf{y}_t$ has been performed. This state is represented in the form of the subsequent input vector $\mathbf{x}_{t+1}$.Thus, the system needs to store the values $r(\mathbf{x}_t)$ and $e(\mathbf{x}_t)$ until $e(\mathbf{x}_{t+1})$ is computed. Using $\hat{r}$ the weights of the evaluator net can be changed according to

$$\Delta v_j = \beta\, [r(\mathbf{x}_t) + \gamma e(\mathbf{x}_{t+1}) - e(\mathbf{x}_t)]\, x_{j,t} = \beta\, \hat{r}\, x_{j,t} \qquad (4a)$$

In this way, the evaluator net is improved. As in the other cases of reinforcement learning (equation 2 b), the signal $\hat{r}$ can be used to train the weights of the selector net:

$$\Delta w_{ij} = \beta_2\, \hat{r}\, [y_i(\mathbf{x}_t) - y_i'(\mathbf{x}_t)]\, x_{j,t} \qquad (4b)$$

These procedures are schematically summarized in Figure 12.**7**.

If a network should be trained but detailed information concerning the output vector is not available, supervised learning methods cannot be applied. One possibility is to use reinforcement methods or learning with a critic. The critic provides only a scalar value (e.g., on a scale good-bad). Different reinforcement algorithms are used for such cases as learning a given task in a constant environment, for different tasks in a changing environment, and where long-term reward of a chain of behavior has to be evaluated.

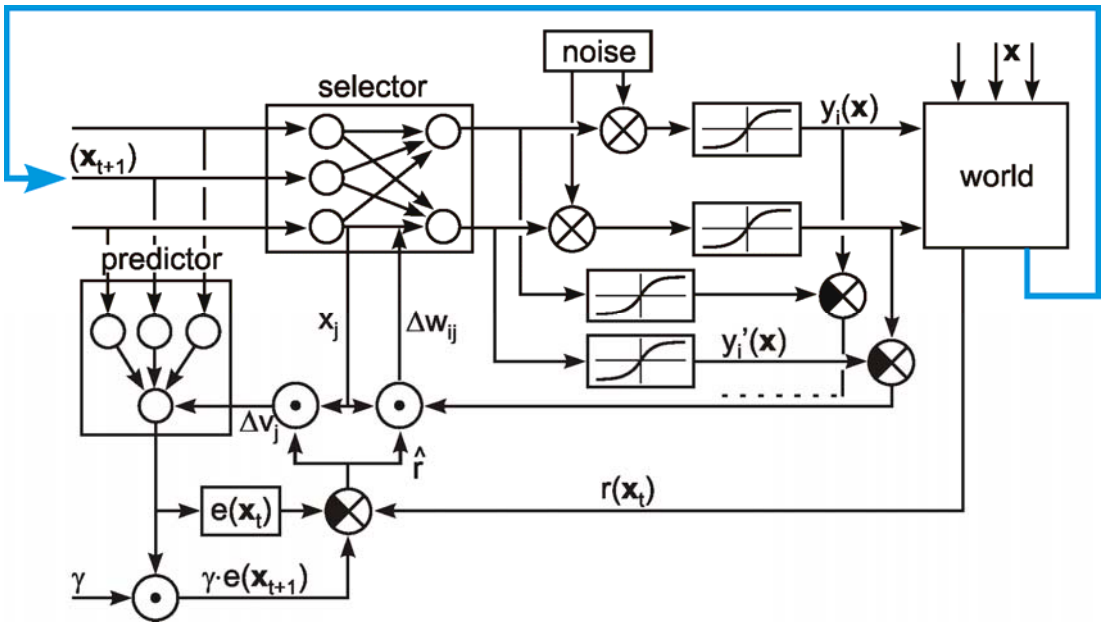


Fig. 12.**7 Reinforcement learning.** Temporal difference (TD) reinforcement permits finding a temporal chain of actions such that the total reward obtained during the complete chain is used as the criterion rather than the possibility high reward of an individual action. In this case, an evaluator net provides an estimation e(**x**) of the expected total reward. Using the actual reward r($\mathbf{x}_t$), the expected value e($\mathbf{x}_t$) and the expected value for the subsequent input vector e($\mathbf{x}_{t+1}$), the appropriate changes of the weights of the selector net and the evaluator net can be calculated. For this purpose, as in Fig. 12.6b, the actual noisy output values are compared with the original values provided by the selector net. The uppermost bold arrow indicates that because of some physical properties of the world, an action **y** in a given situation **x**(t) leads to a new situation **x**(t + 1) which represents the new input vector

12.6 Expert Networks

The types of networks treated above all consist of one, unified network. When such a ”monolithic” network, that has already learnt some tasks, is trained to learn a new task, in principle all weights will be changed and therefore it is possibly that already stored knowledge may be (partially) destroyed. This could be overcome only if, during the training of the new patterns, also the old patterns are trained again. This does however not correspond to the biological situation. The problem could be solved by a modular architecture consisting of a number of independent “expert nets”, if only one of them is learning a given task. For learning n tasks we require n expert nets. (Jacobs et al. 1991, Jordan and Jacobs 1992) proposed to add another net, termed gating network. All n+1 nets are multilayer perceptrons (MLP) and receive the same input **x** (Fig. 12.**8**). The gating net has n outputs g_i. The output vectors of the n expert nets are compared with the training vector. The overall error of each net is fed into a WTA network. Only that network with the smallest overall error, the winner, is allowed to change weights (i.e. there is a competition between the experts nets as is similarly found between single units in the case of competitive learning, see Chapter 12.8). This procedure guarantees that only that expert is learning that is already best suited for the actual task. In this way, the different nets specialize on different input situations. An expert for task A is not any more a good candidate to specialize on a different task B. Any one of the other nets might be better suited for the task B. The output of the WTA net is used as training signal for the gating network, too. The factors g_i, being 0 or 1, are not only used during the training phase, but are also used to control the output of the experts after training. Each expert's output is multiplied by its gating factor, so that only the actual expert controls the output.

Wolpert and Kawato (1998) introduced an expansion in the way that the gating network does not receive its input from the environment, but looks at the errors produced by each expert. From these error values err_i so called "responsibility" values g_i are calculated which are then used in the same way as the gating values, i.e. they are used for learning and for acting. A softmax function ($g_i = \exp(-err_i^2) / \Sigma \exp(-err_i^2)$; $\Sigma g_i = 1$) is used that allows for a blending between several experts. This architecture has the advantage that the gating network does not have to separate all possible environmental situations.

Tani and Nolfi (1999) applied these principles to recurrent nets of Jordan type (Chapter 14.1). Again, all n expert networks receive the same input. Their task is to predict the input during the next time step. All nets are trained using the prediction error (summed squared local errors) by Back Propagation through Time (BBTT; Chapter 12.3). Before this is done, there is a competition between the nets on the basis of the different prediction errors. Using the n prediction error values, a kind of soft winner take all circuit ("softmax function", see above) determines a "quality factor" g_i for each net. The BBTT procedure for each net is weighted using the factors g_i. This means that the best network learns stronger than the other ones, giving this network an advantage in the next learning step for this situation (similar to the feature maps, see Chapter 13, but instead of geometrical neighborhood we have here "performance neighborhood"). The quality or "gaiting factors" g_i are updated using a kind of momentum rule to avoid abrupt changes through time.

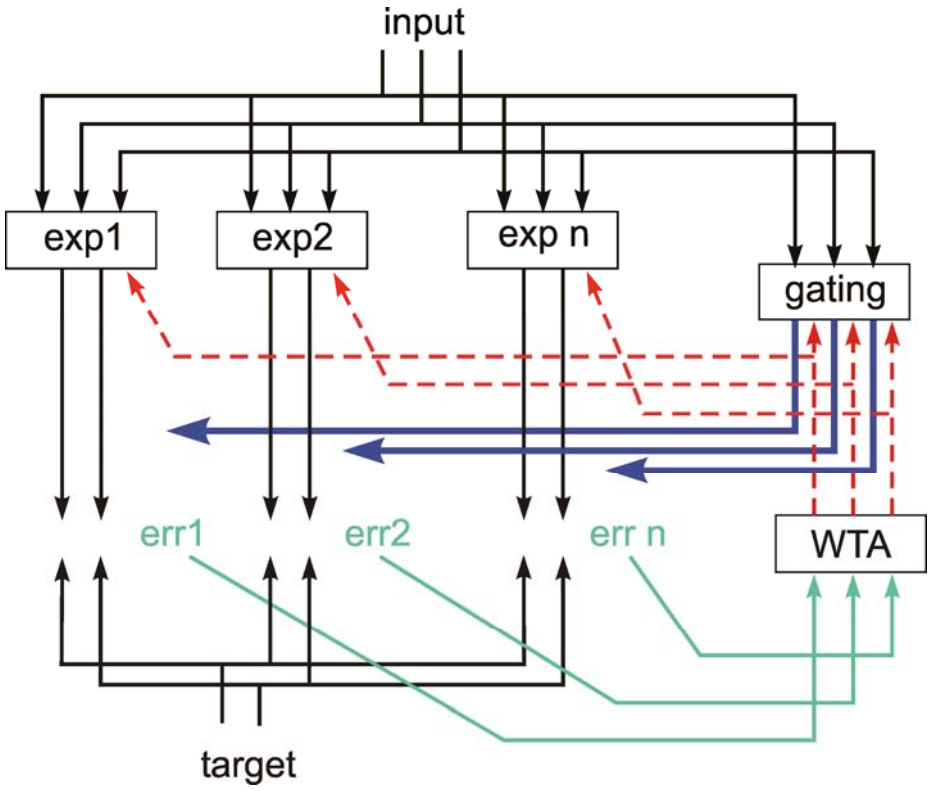


Fig. 12.8 **Expert networks.** N expert nets and one gating net receive the same input. The output of the gating net determines which expert has access to the overall output (blue arrows) and is used for training these experts (red arrows)

A modular architecture allows fast training of new tasks, and avoids the destruction of already acquired solutions.

Box 6

Recurrent networks with parametric biases

As mentioned, there are good reasons to apply a modular, or localistic, architecture. By avoiding cross talk during learning, learning can be sped up and different tasks can be coded more effectively. Furthermore, if a number of experts is already available, new tasks might be solved by a combination of these "vocabulary" and learning of completely new networks may not be necessary. However, there are also problems with this approach. Before learning can begin, the neuronal assemblies, that will form the modules, have to be specified in advance, or any self organizing rules had to be developed that allow an autonomous formation of such subnetworks. An elegant alternative solution is given by the so called Recurrent Neural Networks with Parametric Biases (RNNPB) proposed by Tani (2003). The architecture of these recurrent networks is based on that of Elman/Jordan nets (Chapter 11.5), which are expanded by units that represent the parametric biases (Fig. B6.**1**). These networks at the input receive sensory signals and an efference copy of motor signals at time t and, at the output, provide the corresponding sensor and motor signals at time t+1. Therefore, as in the case of the MMC net (Chapter 14.5), input vector and output vector have the same components. To learn a given sensory-motor task, the input vector for time t is given and the actual output vector is compared with the correct vector at time t+1 as defined by the task. The error is used to train the weights by application of BPTT (Chapter 12.3). After learning is finished, such a network can control the behaviour of a robot as it, for a given sensory input, is able to provide the adequate motor output in order to control the behaviour.

As the network not only provides the motor command for the next time step, but also the expected sensory input, this network could be used in an alternative mode, too. In this "predictive" mode the motor output is switched off, and the sensory output is given as new sensory input at the next time step (the corresponding circuit is not shown in Fig. B6.**1**), which could then be responded by the adequate motor response. If this loop is repeated, the network could perform "imagined behaviour" and thus be used to plan ahead as is possible with the MMC net (Chapter 14.5).

Neglecting the additional PB units, this Elman net can learn one attractor, but not multiple attractors. The situation changes dramatically, if the PB units are introduced which might be considered as a specific form of context units. For each individual PB vector, the network is capable of representing another attractor. Thus, one RNNPB net comprises a distributed memory for different sensory-motor mappings.

Whereas the weights of the net are trained using the BPTT algorithm as mentioned, the activation of the PB units can be considered as a short term memory that has to be learned, too. To this end, again the BPTT algorithm is used. However, in this case the sensory prediction error is used to calculate the activation of these units instead of weights (the PB vector is changed such that the (sensory) prediction error is minimized). Different to the learning of the weights, for learning the PB activation the errors are averaged over a larger time window. This approach supports the PB units to learn a vector that represents a complete behaviour. When different behaviours have been learned by the net, they all are represented by a common weight matrix, but show different PB vectors. After learning is finished, application of any of these PB vectors, that have to be stored separately, allows the net to perform the behaviour corresponding to this PB vector.

In order to control different behaviours, Tani (2003) applied a second, higher level RNNPB, that as input receives the PB vector of the lower level net and is able to control switching between different behaviours. Interestingly, this switching results from a cooperation /competition between the lower

level net that is influenced by the actual sensory input, that is variable due to changes of the actual situation and to noise, and, on the other hand, by the slowly changing state of the upper level network which is therefore able to stabilize the behaviour against such short term variations.

Of course, the disadvantages of a completely distributed memory as mentioned in Chapter 12.6 as well as the problems connected with the application of Backpropagation remain, letting Tani et al. (2004) argue that the localist approach as well as the distributed approach represent two extremes and future research might look for intermediate forms of representations that may combine the advantages of both solutions.

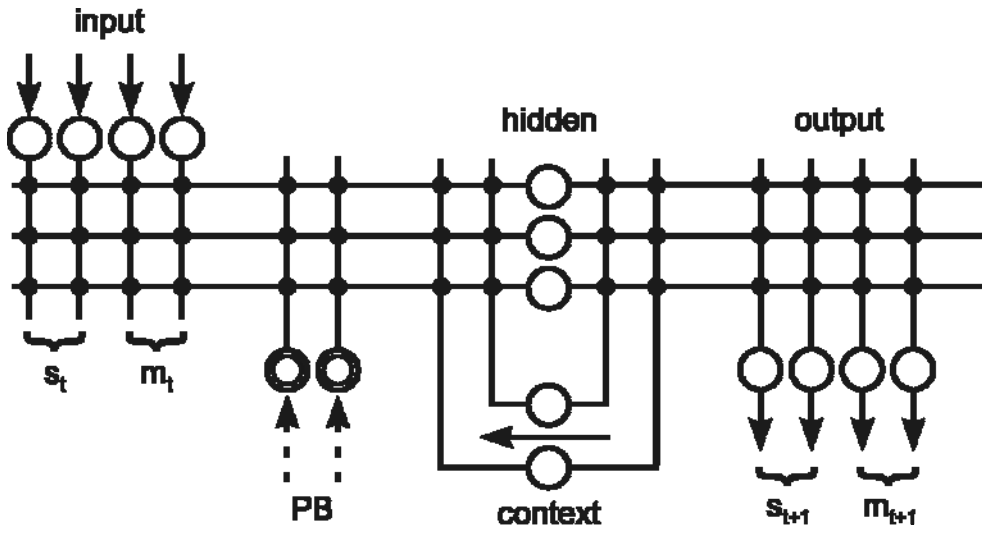


Fig. B6.**1. A recurrent neural net with parametric biases (RNNPB).** This network corresponds to an Elman net (Fig. 11.**6a**) that is expanded by additional input units, called parametric biases (PB).

12.7 Genetic Algorithms

While reinforcement learning corresponds to the situation that an individual net learns in the course of its "live," so to speak, the procedure of genetic algorithms or evolutionary strategies more resembles learning during the course of evolution. Each learning step corresponds to a transition from one generation of individuals to the next. The basic assumption is that there is a sizeable number of individuals (i.e., networks) that differ from each other with respect to their weight values. These nets have to fulfill the given tasks, and their quality is evaluated in terms of a one-dimensional, i, e., scalar, value ("fitness"). Subsequently, selection ensures that, in the next generation, those nets that performed relatively well during tests, occur more frequently, while the worst nets are eliminated. The nets of this new generation are now themselves subjected to mutations, i, e., weight values are varied by random influences, and this generation is then again tested. In this way, nets develop which, from one generation to the next, become increasingly well adapted to a set task. As a variation of this approach the size of random influences ("mutation rate") may decrease from one step to the next. The most important part of such a program is the choise of the fitness function or evaluation function, because the form of this function essentially determines the behavior of the algorithm. The evaluation function might not only refer to the behaviour of the net, but also to its structure. For example, a small number of weights being different from zero (“active” weights”) might be awarded. Thus, pruning could already occur during the learning procedure.

Apart from this so-called "asexual" reproduction, the possibility of "sexual" reproduction is also being studied. In this way, a new net can be generated by mixing the weight values of two (or more) parents. With reference to biology, this is occasionally referred to as "crossing over". These genetic algorithms may also be used to search for an optimal structure for the network. For this purpose, the activations of randomly selected neuroids are set to zero and the evaluation function includes a term which counts the total number of active neuroids such that a smaller number corresponds to a higher fitness value. By this way networks that solve the task but have fewer neuroids have a better chance of surviving the selection procedure.

The comparison of this stochastic method with the gradient descent methods (the delta rule procedures) described earlier reveals that the latter are computationally faster, but suffer from higher probability of sticking in a local minimum. This is much less the case for the evolutionary strategies because the latter search in a broad range of the search space simultaneously. For more detailed information concerning the properties of genetic algorithms the reader is referred to Koza (1992).

In a simplified version, one generation consists of only one single individual. In this case which is called random search, the fitness of the new individual is compared with that of the last one. If the fitness of the new individual is better, the new weight values are accepted. Otherwise the old weight values are maintained and another mutation is tried.

While the two learning methods described last, namely reinforcement learning and genetic algorithms, are much closer to biology than the generalized delta rule, for instance, they still have the disadvantage of requiring many more learning steps and, consequently, a computation that takes much more time than learning methods with superviser.

> Genetic algorithms and random search are learning methods which do not suffer from getting stuck in a local minimum as much as gradient descent methods. However, the former are slower in general. They do not require a supervisor but depend strongly on proper selection of the evaluation function.

12.8 Competitive Learning

We have so far discussed nets in which each neuroid of one level was usually connected to each neuroid of the next level (full connectivity). Apart from the number of levels and neuroids in each level, no further structures were given. Below, we will address some nets in which only selected connections are given. In addition, the weights are divided into two types, those whose values are fixed from the beginning, and those that may change their values during learning.

The task of the perceptron with one output neuroid, as mentioned in Chapter 10.6, consisted of dividing a large number of input patterns (input vectors) into two groups, such that a pattern of one group is responded to by 1, and a pattern of the other group by 0. This was learned with the help of the delta rule via a trainer. We will now consider a similar system which is modified in such away that two output elements are present, in which one neuroid (A) responds to the patterns of one group, with a value of +1 and the other neuroid (B) responds to the patterns of the other group by the output value +1. In this case, what should be learned is that either one or the other output element responds by 1.This can be achieved - even without a trainer - by connecting these two output neuroids by mutual inhibition, so that they represent a small recurrent winner-take-all net. This means that, starting with random input weights, one neuroid will "win," when a given input pattern is used. We can also apply a second input pattern which excites the second output neuroid more than the first. Using this pattern,

the second output neuroid will win. The excitation at the output, i. e., the excitation of the neuroids after passing through the winner-take-all net, is used to train the input connections according to Hebb's rule. Then the weights of the two output neuroids, if only these two patterns are used, will be changed in such a way that the weights of the two neuroids resemble the two input patterns, respectively (except for a factor). This means the system adapts by itself, i. e., without the trainer, to the two patterns. When the weight vectors correspond to the input vectors, as illustrated in the example of Figure 12.**9a** (the winner-take-all net is not shown), unit A responds strongest to input $\mathbf{i}_1$ and unit B to input $\mathbf{i}_2$. If the net is confronted with new patterns, these are divided into two categories, depending on their similarity to either of the two original training patterns. If learning is continued when presenting the two groups of patterns, the weights of the two output neuroids adapt to some kind of mean vector of both groups. The weight vectors find clusters in the input vectors. The vector which represents the center of such a cluster can be considered the representative or *prototype* of this group. In other words, the net develops a "concept" of the two types of the vectors obtained from the outer world. Thus, this is called competitive learning and is a type of unsupervised learning that has the advantage that the learning phase can continue during the active phase of the system. A separation between two states, learning and operation, is not necessary as it is in the case, for example, of the delta rule.

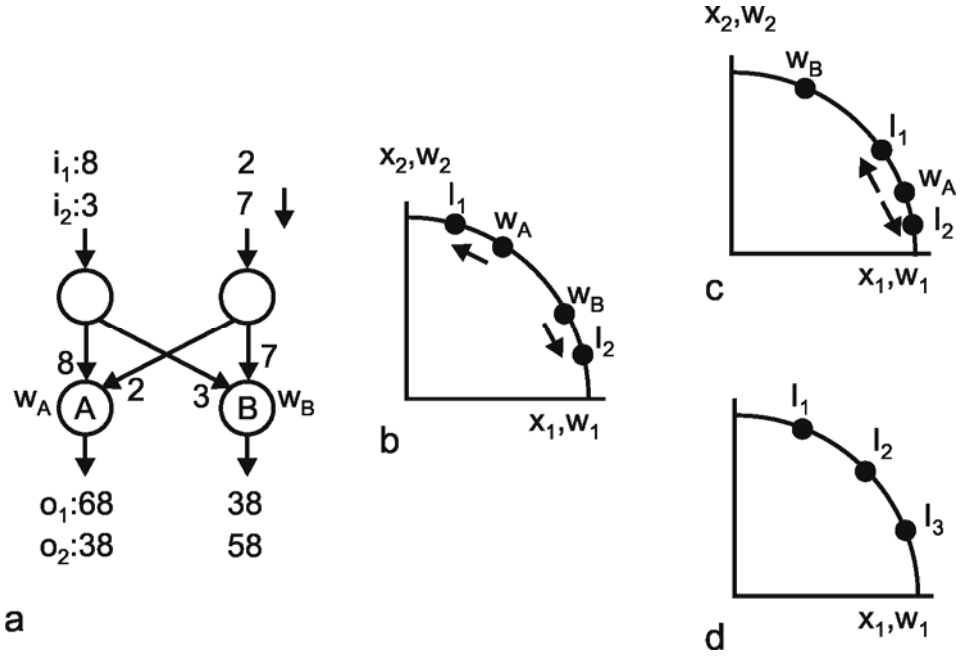


Fig. 12.**9 (a) Competitive learning.** A network consisting of two layers. For the given weights the input vector $\mathbf{i}_1$ yields the output vector $\mathbf{o}_1$ and $\mathbf{i}_2$ yields the output vector $\mathbf{o}_2$. The two neuroids A and B form a winner-take-all net the connections of which are not shown. Thus, neuroid A will win the competition after application of $\mathbf{i}_1$ whereas neuroid B will win when $\mathbf{i}_2$ presented. This means that, in the first case, only neuroid A will learn, i. e., change its weight, and in the second case only neuroid B will learn. (b) Graphical illustration of the two normalized input vectors $\mathbf{i}_1$ and $\mathbf{i}_2$. The two normalized weight vectors $\mathbf{w}_A$ and $\mathbf{w}_B$ can be plotted in the same coordinate system. The arrows indicate the change of the weight vectors during learning. (c) As (b), but showing an inappropriate arrangement of input and weight vectors. In this situation, neuroid A will win the competition for both input vectors $\mathbf{i}_1$ and input vector $\mathbf{i}_2$. Therefore, $\mathbf{w}_A$ will always move between $\mathbf{i}_1$ and $\mathbf{i}_2$ and neuroid B will never win. (d) Three input vectors can be separated only if the second layer of the network consists of the least three neuroids

In our example, we started with two input patterns which already can be distinguished by the net. The only improvement was that during training the system finds a better representation of the two groups of input vectors. The more interesting situation, however, is to start with two given sets of input patterns and to find a net which by itself separates these two groups. As will be shown in the following, this is possible even for the more general case, where the winner-take-all net consists of more than

two neuroids. The input patterns can then be divided into a correspondingly greater number of types of patterns. The net thus has the property of combining vectors into groups which are characterized by the excitation of different neuroids of the output layer. This is also known as vector quantization.

If we have n neuroids in the input layer and k neuroids in the output layer, the application of Hebb's rule ($\Delta w_{ij} = \varepsilon x_i y_j$; i = 1...n, j = 1 ... k) means that the weight vector of the winner becomes similar to the input vector **x**. The learning rule only works for the winner, since only here the output value y_j is 1, whereas for the other neuroids it is always 0. The value of ε (learning rate) may decrease during the course of learning. The disadvantage, that with Hebb's rule weights can grow without limit, can be counteracted if the weight vectors are normalized after each learning step. (Normalization usually means that all components of the vector are multipled by a common factor such that the vector length is 1). More important, the normalization avoids one weight vector becoming so large that it wins for every input vector so that other output neuroids have no chance to learn. Assume for example, that the weights of the neuroid A in Figure 12.**9a** had twice their values, then the output of A and B was 136 and 38, respectively, to input $\mathbf{i}_1$ and 76 and 58 to input $\mathbf{i}_2$. So in each case unit A will win and unit B will never have the chance to learn. On the other hand, the disadvantage of normalization is that the effective dimension of the input vector is n - 1, and that the input space is limited. For example, if the net contains two input neuroids, not the complete two-dimensional space is used as input but only those vectors which point on the unit circle. Thus, the effective input space is only one-dimensional and is of limited extension. For three-dimensional input vectors the effective range is the two-dimensional surface of the unit sphere.

Another method would be to use the learning rule $\Delta w_{ij} = \varepsilon (x_i - w_{ij}) y_j$, the so-called "standard competitive learning rule." This has the effect that input vector and weight vector gradually become similar to each other, but nevertheless the vectors have to be normalized to enable discrimination. Depending on the kind of normalization, this rule may be equivalent to the application of Hebb's rule. The situation is visualized in Figure 12.**9b** for the simple net already shown in Figure 12.**9a** where the input consists of only two neuroids. Then both the input vectors and the weight vectors can be plotted in the same plane. Because of normalization all vectors point on the unit circle. With the starting situation given in Figure 12.**9b**, the weight vectors $\mathbf{w}_A$ will be pulled to the input vectors $\mathbf{i}_1$ and $\mathbf{w}_B$ to $\mathbf{i}_2$. However, there is still the problem that success depends critically on the distribution of randomly chosen weights at the beginning. This can be shown in Figure 12.**9c** where both input vectors will attract only weight vector $\mathbf{w}_A$. This will move to and from between $\mathbf{i}_1$ and $\mathbf{i}_2$ whereas the second output neuroid (vector $\mathbf{w}_B$) will never win and therefore never change its weights. Thus, successful learning is not possible for any arbitrary configuration of weights at the beginning. A way to avoid this problem is to start with weights that are not randomly distributed but already equal some of the input vectors. This heuristic, of course, also decreases the time needed for learning. Another solution is "learning with conscience," where those units which often win are gradually excluded from the competition, so that other "dead" units also have the change to learn.

If we want to avoid the disadvantages of normalization and the reduction of the input dimension, the following net could solve the problem. One can use the standard competition learning rule. After learning is finished, the output value of the neuroids of the second layer is not determined by computing the scalar product $y = f(\Sigma(x_i w_i))$, but computing the Euclidean distance between vectors **x** and **w**, according to

$$y = f\left(\sqrt{(\Sigma(x_i - w_i)^2}\right) .$$

For f(u) the Gaussian function often is used. Then only a neuroid the weight vector of which equals the input vector **x** is excited by 1. The excitation of any other neuroid decreases exponentially with the Euclidean distance between the weight vectors of both neuroids. Neuroids of this form are called radial

neuroids. However, such radial neuroids can probably not be realized by a single biological neuron but require a microcircuit consisting of a number of biological neurons as will be explained in the next paragraph.

The following arrangement could ensure that, with a particular input excitation, only the topologically corresponding neuroid of the second layer will respond (Fig. 12.**10**). The network mentioned above, consisting of a small number (e. g., k = 2) of neuroids in the input layer and n topologically arranged neuroids in the second layer, has to be expanded in the following way (in the example in Figure 12.**10** we use k = 2 and n = 3). All neuroids are referred to by using their activation values as shown in Figure 12.**10**. The input neuroids are x_1 and x_2, respectively, in Figure 12.**10**. Each of the n neuroids of the second layer is replaced by k neuroids, here x_{i1} and x_{i2} (i = 1 to n). Each input layer neuroid x_i is connected to the n neuroids x_{ij} (i = 1 to n, j = 1 to k) of the second layer by a weight fixed to value 1. In addition, a bias neuroid, showing an excitation of - 1, is introduced which is linked to each neuroid of the second layer via variable weights w_{ij}. (The indices in this case are identical to those of the corresponding neuroid of the second layer).

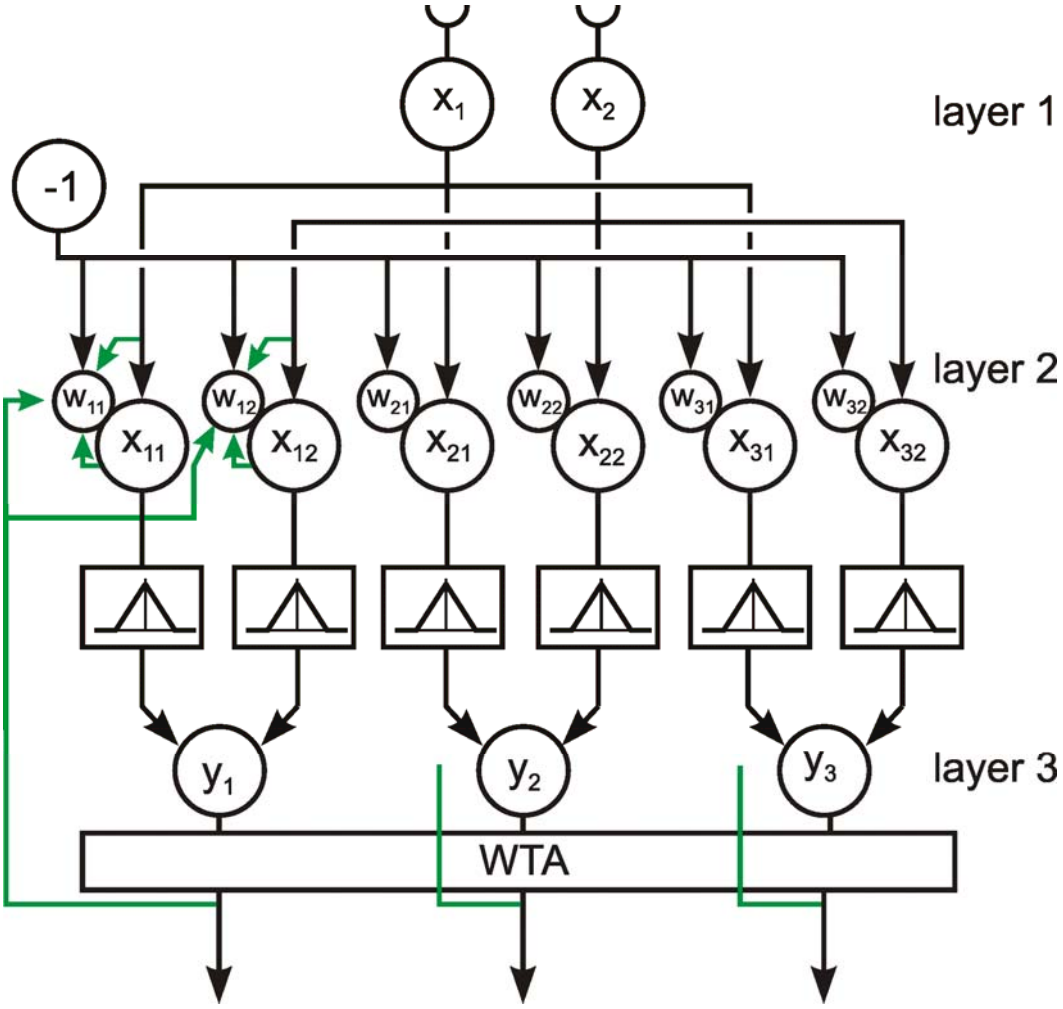


Fig. 12.**10 Competitive learning without normalization.** This network can learn to separate groups of vectors (in this example three) distributed over the whole two-dimensional space x_1, x_2. The changeable weights are shown by small circles

The output of each neuroid of layer 2 is endowed with a Λ-shaped activation function (e.g. an inverted full-wave rectifier or a Gaussian function). The outputs of the k related neuroids (x_{i1} to x_{ik}) are added to

a further neuroid y_i(i = 1 to n) of layer 3. The output of the neuroid y_i corresponds to that of a radial neuroid, as it reacts maximally if $(x_1 - w_{i1})^2 + (x_2 – w_{i2})^2$ is zero (or in other words, the input vector and the weight vector are identical). This means that in this way a radial neuroid is described by a small network consisting of k + 1 simple neuroids. A subsequent winner-take-all network decides which of the neuroids y_i is most excited. Learning is performed for neuroids x_{ij} which have the same first index i as that of the winning output neuroid y_i. Learning is not applied to the synapse between input and neuroid of the second layer, but to the weights w_{ij} (j = 1 to k) of the bias neuroid. Hebbian learning is applied according to the rule $w_{ij} = \varepsilon\ y_i\ x_{ij}\ x_j$ (j = 1 to k). This is shown by broken lines in Figure 12.**10**. In this way, after learning is completed, only that neuroid of layer 3 that corresponds to the input vector is given an output value of 1. All others are given a lower output excitation, which is smaller the greater the distance to the maximally stimulated neuroid.

As in earlier cases, the solution of problems which are not linearly separable (for example when input vector $\mathbf{i}_1$ and $\mathbf{i}_3$ in Figure 12.**9d** should be grouped together to excite output unit A and input vector $\mathbf{i}_2$ should excite output unit B) is only possible when several such networks are connected to form a system consisting of more than two layers.

One problem of competitive learning networks is that the number of clusters into which the input vectors (patterns) should be grouped, is predetermined by the number of units existing in the output layer. If the net contains more neuroids than the types of input patterns that occur, the categorisation might be too fine. This problem was solved by the so-called ART (adaptive resonance theory) network introduced by Carpenter and Grossberg (1987) in several versions. Only the qualitative properties of this network type will be described here. The network contains a large number of output units which, at a given time, can be distinguished between being "enabled" or "disabled." Each enabled unit is responsible for a cluster of input patterns; the disabled units did not respond to one of the input vectors presented up to this time. Only when an input vector occurs which cannot be attributed to one of the units already enabled, will one of the disabled units change state and become responsible for this new category. An input vector corresponds to a "new" category if the distance to each of the already existing prototype vectors is larger than an adjustable, prespecified parameter. If this parameter is high, the input space is divided into coarse intervals, if it is small, a large number of output units may be necessary to cope with the finer resolution required. Of course, the number of possible categories is not infinite but limited by the number of available output units.

> Competitive learning requires a prestructured network where only selected weights can be changed according to Hebb's rule. It provides a method whereby input vectors can be grouped without any trainer or critic, neither is the introduction of an evaluation function necessary. No switch from learning phase to action phase is necessary, rather learning can continue infinitely.

13 Feature Maps

Up to now there has been no special requirement as to how the neuroids in the second layer are arranged. A further extension of the principle of competitive learning can be made when these neuroids show a topological order, which means that there is a defined neighborhood for each neuroid. Such a topological arrangement is a frequently encountered feature of neuronal systems, as it is often found that neighboring neurons of one layer project to neighboring neurons of another layer. The topographic maps of the cortex, the sensorimotor homunculi, for example, are formed in this way. Similar projections are known for other sensory systems such as for instance the projection of the

retina onto the cortex. In such a connection, the question arises as to whether these precise projections between the cells of one level and another are genetically determined, or whether other mechanisms exist which can be considered responsible for the origin of such connections. An example of the latter will be given below.

When a neighborhood is defined in the second layer, the global winner-take-all network could be replaced by local "Mexican hat" connections in the sense that direct neighbors are excited and only the more distant neuroids are inhibited (von der Malsburg 1973, Fukushima 1975, 1980). In this way, a group of neighboring neuroids can learn to react to a given input. If another, similar input occurs later, neuroids in this region are better prepared to respond to this new input. Thus, a topological arrangement can be obtained in the sense that, after learning, a small change in the input causes the excitation of similar (i. e., neighboring) neuroids in the output layer. A somewhat different formulation of the same principle was later proposed by Kohonen (1982, 1989).

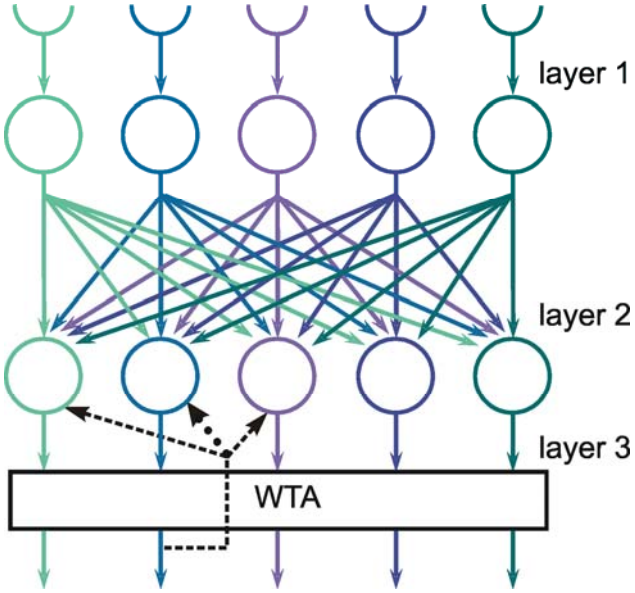


Fig. 13.**1 Feature map.** Competitive learning is applied to a network where, in layer 2, neuroids are arranged to form a neighborhood. For sake of clarity, the winner-take-all net is shown as a separate layer (layer 3). (Layer 2 and 3 could however also be taken together to form one layer as was the case in Fig. 12.**9**). As for normal competitive learning, the winner changes its weights (dashed lines) but, in addition, weights are also changed within a limited neighborhood. This change is smaller and is symbolized by the smaller arrowheads

To explain this principle, a two-layered net is assumed from which only a linear cross section is represented in Figure 13.**1**. Only connections in one direction are permissible so that we are dealing with a fully connected feedforward net. To begin with, the neuroids of one level are connected with those of the second level via randomly distributed weights. To make the situation more understandable, we introduce a third layer of neuroids, closely adjoining the second layer which only serves to influence the weights between the first and second layers, depending on the input signals. The third layer consists of identical neuroids which are excited each by their directly neighboring neuroids of the second layer by constant, equal weights. The neuroids of the third layer inhibit each other, and thus form a winner-take-all net. Each of these neuroids is endowed with excitatory connections to the neuroids of the second layer such that, instead of the neuroids being excited, their weights are being increased according to Hebb's rule or the standard competition learning rule, as mentioned above (dashed arrow in Fig. 13.**1**). As mentioned in Chapter 12.8, this means that the input vectors and weight vectors have to be normalized. The changes of the weights are large for directly

neighboring neuroids of the second layer and decrease with increasing distance, indicated by the smaller arrowheads in Figure 13.**1**.

What are the properties of this network? If a small group of neighboring neuroids of the input layer are stimulated, one or more neuroids of the second layer, and subsequently of the third layer, are excited, depending on the initially arbitrary arrangement of the weights. Then, the neuroid of the third layer neighboring the most excited neuroid of the second layer obtains the highest excitation in the third layer. Due to the winner-take-all properties of the third layer, this neuroid will be the only one that is excited. Thus, the weights of the corresponding neuroid of the second layer increase. Moreover, the weights of neighboring cells of the second layer also increase, although to a lesser extent. Consequently, in the case of a future excitation of a group of cells of the first layer that shows a certain overlap with the neuroid just mentioned, these neighboring cells of the second layer have a bigger chance of winning, and thus of increasing their weights. In this way, connections are developed in which neuroids, which are neighbors in the first layer, are directly connected to those that are also neighbors in the second layer. This means that a corresponding topology develops automatically in the second layer. It could well be, though, that the orientation of neighborhood in the first and second layers runs in opposite directions, so that connections occur corresponding to those known as chiasma in neuroanatomy.

In the network described above, the neuroids of the input layer usually contain a Heaviside activation function, i. e., are either on or off, depending on whether or not the stimulus lies within the "receptive field" of the neuroid. There is another paradigm often discussed. In this case, the input is not given in the form of an extended neuronal layer, but consists of only a small number of neuroids. Neighborhood is expressed as similarity of the input vectors by which the neuroids of the first layer are excited. An example is given in Figure 13.**2**. The net consists of two neuroids in the input layer and ten neuroids in the second layer arranged along a line (Fig. 13.**2a**). The ordinates of Figure 13.**2b** show the values x_1 and x_2 of the input neuroids, which can adopt values between 0 and 1. The input vectors are normalized to a length of 1, thus the endpoints are lying on the unit circle (Fig.13.**2b**). Because of the normalization, x_2 can be calculated when x_1 is given. Therefore, the net effectively has only one input unit. The ordinates of Figure 13.**2b** also designate the values of the two weights of each neuroid of the second layer. Figure 13.**2c** shows the development of the weight values during the learning process for the weights w_{i1} connecting the first input neuroid to the ten units of the second layer. The input values are taken from an equal distribution. As can be seen in Figure 13.**2c**, the weights, being randomly distributed in the range between 0.6 and 0.8 at the beginning, are now topologically ordered and cover the whole input range after 500 training trials. In this state, the network "transforms" a scalar input value into a spatial position. For example, an input $x_1 = 0.4$ will lead to an excitation of unit no. 2 and $x_1 = 0.7$ affects only unit no. 8. One interesting feature of these topology-preserving or feature maps, and also of the competitive learning net mentioned in Chapter 12.6, is that the arrangement of weights also depends on the frequency of the input patterns used. This will be demonstrated below in another example.

Feature maps are an interesting concept of how gradients could be trained that map the topology of the outside world. As mentioned earlier, the problems of normalization, which require a global knowledge, and the reduction of input space could be solved when applying the network shown in Figure 12.**6** to competitive learning. After a neighbourhood is introduced in layer 3, this network can also be used to train feature maps.

In the sense described so far, feature maps can be considered capable of producing an internal representation of the sensory environment. In order to use this approach to control motor output, the system was extended by Martinez et al. (1990) and Ritter et al. (1992) in the following way. In the example presented here, a three-joint robot arm should move its end effector to a given target. The

position of the target, corresponding to the desired position of the end effector in three-dimensional space, is given by two cameras that provide two coordinates each thus, taken together, a four-dimensional vector. Therefore, the desired position of the robot end effector can be represented by four input neuroids representing the x and y coordinates of the two camera frames. When only target positions on a horizontal table, i. e., in a two dimensional plane, are considered, a mapping from this four-dimensional input vector onto a two-dimensional layer of, in the example of Figure 13.**3**, 15 x 24 neuroids is possible using the procedure described above (Ritter et al. 1989). In this mapping, the position of each of these 15 x 24 neuroids within the layer corresponds to a position on the table. The development of this map is illustrated in Figure 13.**3a, b, c** in a similar way as was shown in Figure 13.**2c**, but now for a two-dimensional situation. The two weights of each unit are projected on the table position such that each unit (to be more exact, its weight vector) is represented by a point on the table. To show how the topological arrangement of the units in the 15 x 24 layer fit the position on the table, the weight vectors of neighboring units are connected by lines.

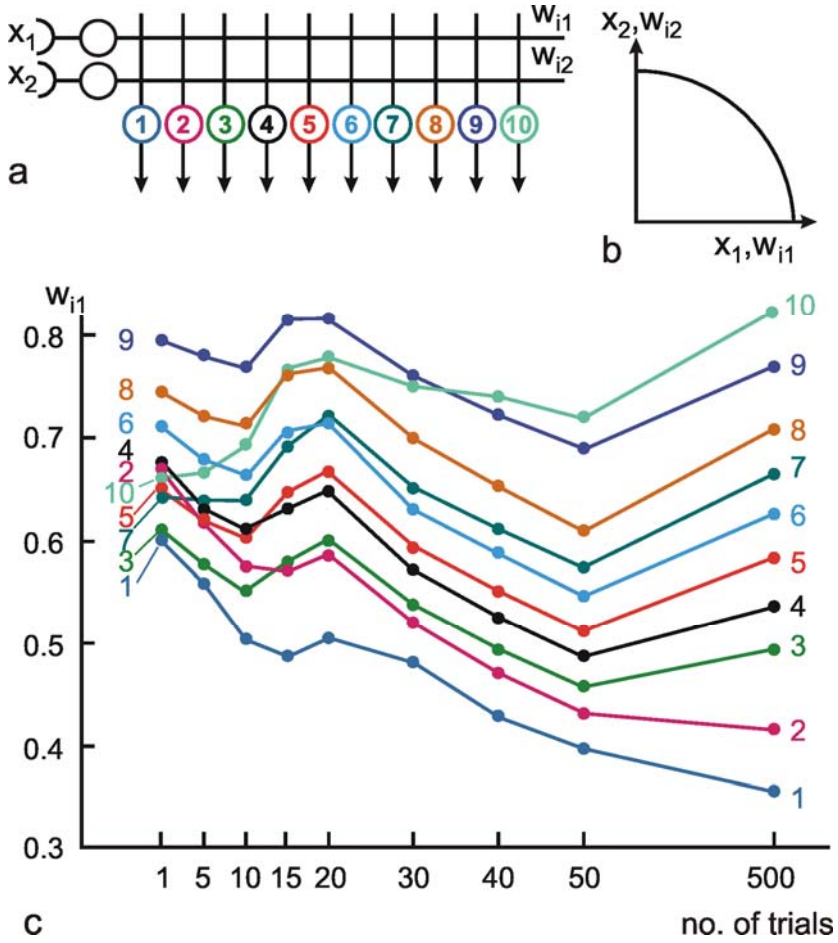


Fig. 13.**2 Development of the mapping of an activation value onto a spatially distributed layer.** (a) the net contains two input units and 10 topologically ordered output units, numbered from 1 to 10. (As the input vector is normalized to unit length, one scalar input, e. g., x_1, would suffice). (b) shows the superposition of two coordinate systems, one for the (normalized) input vectors and the other for the weight vectors. As the weight vector are normalized, too, all vectors point to the unit circle. (c) Development of the weights $w_{i1,}$ i. e., the weights connecting the first input unit with the output units. Weight values are plotted along the ordinate. The weights are identified by the number of the corresponding output unit. At the beginning, the weights show a random order in the range between 0.6 and 0.8. During the training procedure, after 40 trials, the weight values are properly ordered, i. e., reflect the spatial position of the neuroids. After 500 trials, they cover the range between 0.3 and 0.9, showing an about equal distribution. The winner-take-all connections in the output layer, as well as the circuits for normalization and for learning are not shown.

At the beginning, because of the randomly chosen weights, no order can be seen (Fig. 13.**3a**). During learning (Fig.13.**3b**, **c**), the initial random arrangement changes into a well-ordered map (b, 500 trials, c, 10000 trials). Thus, after learning, the 15 x 24 layer represents positions on the table for an arbitrary arrangement of the two cameras. Of course, a change of the position of a camera requires a new training session. However, small movements of the cameras need only minor changes of the weights, which means shorter learning sessions. Interestingly, when the central part of the workspace is more frequently used during the learning process, a non-uniform mapping results, showing a better spatial resolution in this central part (Fig. 13.**3d**).

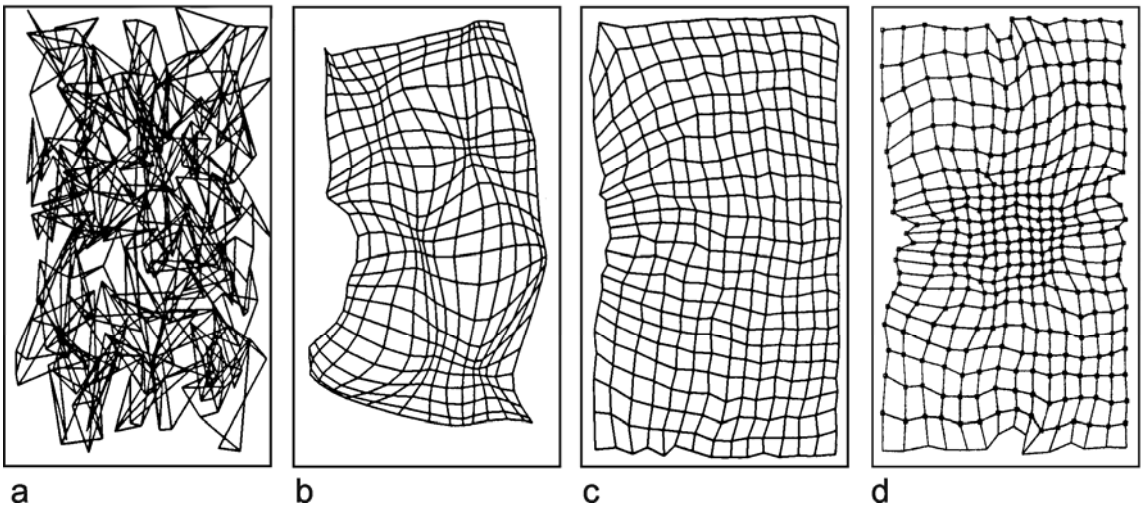


Fig. 13.**3 Development of a two-dimensional feature map.** Positions on a horizontal table are viewed by two cameras. Therefore, each point on the table produces a four-dimensional vector (two coordinate values for each camera frame). This vector is mapped on a two-dimensional grid consisting of 15 x 24 neuroids. The figure shows the values of two weights of each unit, corresponding to the input from one camera frame, projected onto the table. Two corresponding weights are shown by one point. Weight vectors of neighboring neuroids are connected by lines. The initial random distribution (a) becomes more ordered after 500 training procedures (b) and shows an about stable state after 10000 training sessions (c). When positions near the center of the table are presented more often, the mapping shows a higher spatial resolution in this central part (d) (Ritter et al. 1989)

In order to control the movement of the robot arm, Ritter, Martinez, and Schulten used a second mapping by which each unit of the 15 x 24 layer was associated to three output values corresponding to the positions of the three robot joints. Furthermore, Figure 13.**3** is based on a more complicated example, which also includes the dynamic properties of the robot arm. Both of these extensions lead to the concept of local linear map networks (LLM nets) in which each neuroid is assigned a local linear mapping that describes the transformation in a local region of the input space (for further details, see Ritter et al. 1992).

When the competitive learning procedure is applied to a neuronal layer with a defined neighborhood, topological mappings can be learned. Feature maps can also be used to transform scalar input values into an excitation of spatial positions. The mapping can be distorted to reflect the frequency distribution of the input vectors.

Box 7

Historical Note

An early sketch of a system which could be described as a distributed neuronal system was presented by Exner (1894). It represents a "center for motion perception" (see Franceschini et al. 1989). Sequential flashing received by the receptors of the human retina creates an impression of movement and the eye is moved in the direction of the apparent movement. Exner proposed a wiring diagram to explain this finding (Fig. B7.**1**). The nodes marked by letters a - f represent receptors of the retina. An excitation of one receptor proceeds in all four directions. These excitations are summed up at the four centers E, S, Jt and Jf. These, in turn, excite the motor centers a_1 - a_4 which are connected to four eye muscles (Mus. int., Mus. sup., Mus. ext., Mus. inf.) moving the eye. The delay required to detect the movement (see Box 2) was assumed to be due to the finite conduction velocity of the excitation along the nerve fibers. Connections C are assumed to transmit the signals to the cortex.

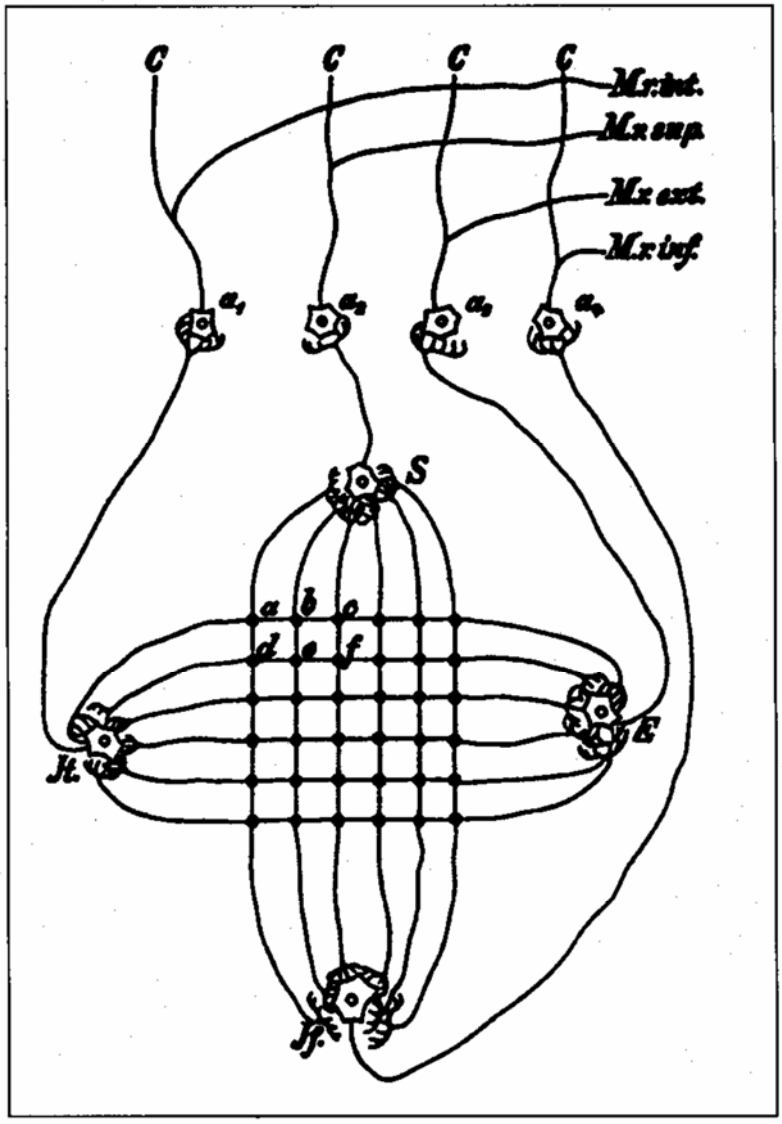


Fig. B 7.**1 Probably the first sketch of a neural network, as proposed by Exner (1894)**

First detailed theoretical investigations of the properties of neural networks were performed by the independent development of a two-layer feedforward system which was called Perceptron by Rosenblatt (1958) and Learning Matrix by Steinbuch (1961), and the investigations of spatial high-pass filters, namely lateral inhibition in the visual system (Reichardt and Mac Ginitie 1962, Varju 1962, 1965, Ratliff 1965).

Following the important, but partly misinterpreted book of Minsky and Papert (1969), most investigators lost interest in this field. About twenty years later, two different approaches gave an extraordinary thrust to the investigation of massively parallel systems. One approach was given by the seminal papers of Hopfield (1982, 1984) showing the interesting properties of recurrent networks. The other was the paper of Kohonen (1982) on the self organization of feature maps. The latter idea was to a large extent proposed 6 years earlier by Willshaw and von der Malsburg (1976), but this paper apparently was before its time. At that time, there were no significant groups of scientists interested in this research. The third strong push came in 1986 by, again, two independent developments. Both LeCun (1985) and Rumelhart, Hinton, and Williams (1986) showed that the back propagation algorithm is capable to train feedforward networks having more than two layers. These papers removed the ban under which feedforward systems were put unintentionally by Minsky and Papert. However, even here a paper had already appeared about 10 years earlier (Werbos 1974), and even in 1967 and 1968 Amari published results which, in principle, contain the necessary information, but neither were recognized as significant at the time.

Further important steps were the introduction of new training methods, in particular reinforcement learning (Barto et al. 1981), competitive learning (for an early collection of papers see Amari and Arbib 1982), and genetic algorithms which have been introduced independently of the development of artificial neural networks (Holland 1975, Rechenberg 1973). Promising developments concern structured networks, i.e., networks consisting of specialized modules ("expert nets," Jacobs et al. 1991) and cascaded networks (Fahlman and Lebiere 1990, Littmann and Ritter 1993). The probably most important fields for future research concern the properties of asymmetric recurrent networks and the question of "active learning", where training examples are not presented to the network in random order, but where the net actively selects interesting training examples. Furthermore, the organization and self-organized development of large memories as well as internal models for action planning have to be considered.

14 Some Special Recurrent Networks

14.1 Simple Oscillators

As was mentioned earlier, recurrent networks represent the most general format of a network. However, there is yet no general theoretical framework that describes the properties of recurrent nets. Therefore, in addition to Hopfield nets and Elman/Jordan nets described earlier, several networks will be discussed below which have a specifically defined structure. The net shown in Figure 14.**1** is endowed with an asymmetrical distribution of weights. Consequently, it no longer fulfills the conditions of a Hopfield net. This network shown generates a rhythmic output when started by an input vector of (1,0,0), for example. The output vector then regularly cycles from (1,0,0) to (0,1,0), (0,0,1), (1,0,0) ..., i.e. a cycle of period 3. For how to train such a network see Chapter 12.4.

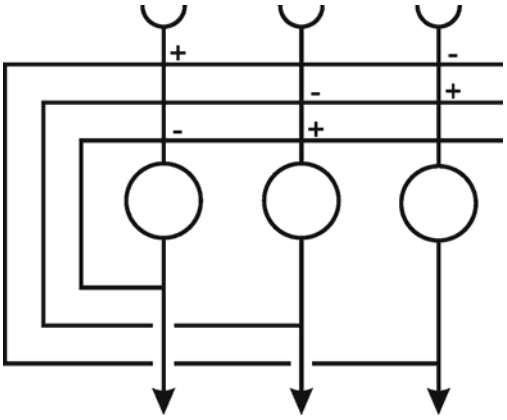


Fig. 14.**1 An asymmetrical recurrent network showing rhythmic behavior.** After the excitation of one unit, the three units are excited in cyclic order

In the following example, a further simple net with an asymmetrical distribution of weights will be used. Rhythmic movements, such as the movement of the wing of an insect in flight, or a fin during swimming, or a leg during walking, for example, are frequently not determined by an internal oscillator. Rather, as described in Figure 14.**1**, the reversal of the direction of a movement max be determined by a signal emitted by the sense organs in such a way that the registration of either of the two extreme positions triggers a reversal of the movement's direction.

This behavior is simulated by the recurrent net shown in Figure 14.**2**. The output of the first neuroid represents movement in one direction, i. e., in the case of a wing it would activate the levator muscles, which raise the wings (US, upstroke). The second neuroid would, in turn, activate the depressor muscles, which lower the wings (DS, downstroke). The activity of these muscles thus results in a change of wing position. In a simplified way, this can be simulated by giving the outputs of both neuroids to an integrator with opposite signs. The wing position determined in this way is measured by two sense organs, but an excitation is transmitted only if either the lower extreme position (LEP) is reached (Heaviside characteristic at the input of neuroid 1), or the upper extreme position (UEP) is exceeded (similarly at neuroid 4). The integrator thus represents a simulation of the mechanical part of the total system. On the basis of the given weights, the net is endowed with the following properties. If all inputs are zero, activity ceases. If an excitation, i. e., an input of 1, is given to neuroid 3 for a short period (i, e., the input vector has the components (0,0,1,0)), the depressor muscles are activated. By way of a positive feedback, the net remains in this state. By means of the integrator the position is moved downward until neuroid 1 obtains an above threshold excitation. The net thus obtains a new input vector (1,0,1,0). A simple calculation shows that the corresponding output vector is (1,0,0,0). In the subsequent cycle it will be (1,1,0,0). This vector is stable, meaning that the wing is now making an upward movement. When the threshold of neuroid 1 is reached again, the system obtains the input vector (0,1,0,0).This is also stable because the neuroid stimulates itself. The wing is thus being moved upward until, at the upper extreme position, the corresponding mechanism is triggered. Thus, with the mechanics and the sense organs, the total system performs rhythmic oscillations.

The weights shown in Figure 14.**2** can be obtained by applying the learning algorithm described in Chapter 12.2, last paragraph. To this end, the four input-output vector pairs have to be used for training: (0,1,0,0) : (0,1,0,0) to maintain upstroke, (0,1,0,1) : (0,0,1,0) to change from upstroke to downstroke, (0,0,1,0) : (0,0,1,0) to maintain downstroke, and (1,0,1,0) : (0,1,0,0) to change from downstoke to upstroke (Cruse et al. 1993).

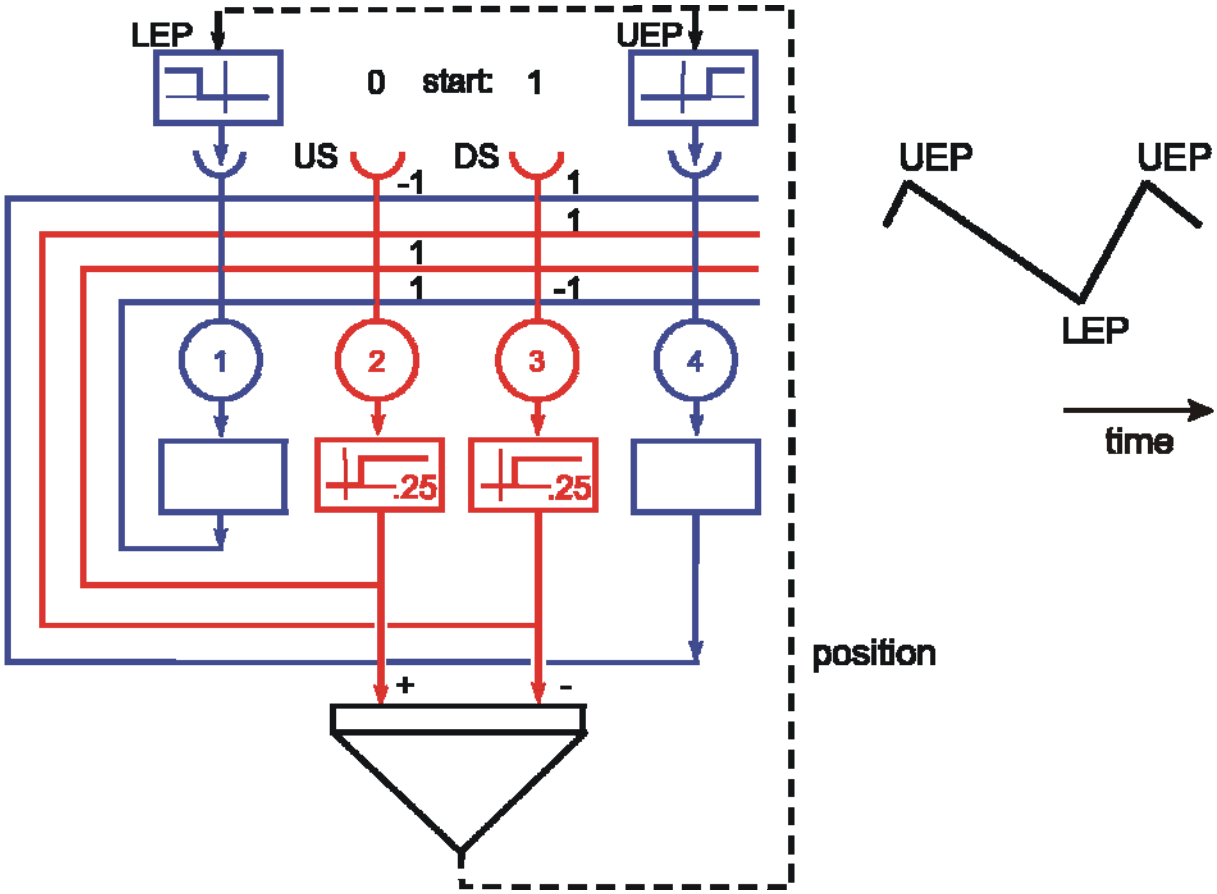


Fig. 14.**2 (a) A recurrent network for the control of rhythmic movement of, for example, a wing.** The net contains internal and external recurrent connections. The „external" connections shown by dashed lines represent the physical properties of the environment. Two motor units US (upstroke) and DS (downstroke) move the wing. The Heaviside characteristic has a threshold of 0.25 in both units. An integrator is used to represent the transformation from motor output to wing position. Two sensory units (blue) monitor whether the lower extreme position (LEP) or the upper extreme position (UEP) is reached. These can be represented by thresholds of -10 and 10, for example. The activation functions of neuroids 1 and 4 are linear. These sensory units have only feedforward connections with the motor units (red). The latter show positive feedback which stabilizes the ongoing activity.

14.2 Echo State Networks

When we are interested to construct a recurrent neural network that shows a complex, but given dynamic behavior, backpropagation through time can be applied as described above. A much simpler solution, however, is to use echo state networks (Jaeger and Haas 2004). An echo state network consists of two parts, a recurrent network with fixed weights, called dynamic reservoir, and output units that are connected to the neuroids of the dynamic reservoir. Only one output unit is depicted in Figure 14.**3** for simplicity. The dynamic reservoir consists of recurrently, usually sparsely connected units with logistic activation functions. The randomly selected strengths of the connections have to be small enough to avoid growing oscillations (this is guaranteed by using a weight matrix with the largest eigenvector smaller than 1). To test this property, the dynamic reservoir, after being excited by an impulse-like input to some of the units, may perform complex dynamics which however should decrease to zero with time. These dynamics are exploited by the output units. These output units are again randomly and recurrently connected to the units of the dynamic reservoir. Only those weights that determine the connections from the dynamic reservoir to the output units are learnt. All other weights are specified at the beginning and then held fixed.

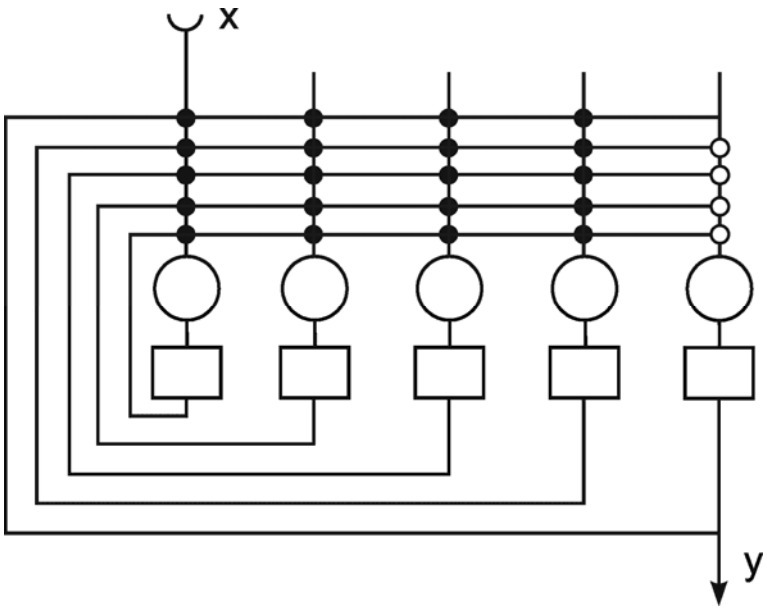


Fig. 14.3 Echo state network. The dynamic reservoir consists of recurrently connected units. The output units are recurrently connected to the units of the dynamic reservoir, but only the output connections (open circles) are due to learning. All the other weights (closed circles) are fixed before learning starts.

To learn a given dynamic behavior, the output units receive this dynamic function via an external input that is only used during the training procedure: by this way the output units are “forced” to show the appropriate behavior. Via the recurrent connections to the dynamic reservoir the latter is excited to show any probably complex dynamic behavior. This is again read by the output units the weights of which now have to be changed in a way that they exactly produce the forced activity of the output units. Learning of the weights can simply by done by application of the delta rule to the output units. If learning is finished, the complete system, dynamic reservoir plus output units, has developed an attractor that produces the desired dynamics which are stabilized against small disturbances. The units of the dynamic reservoir can also be driven by external input such that different dynamics could be produced depending on this input signal. A quite similar approach, called “liquid state machines” has been proposed by Maass et al. (2002), the main difference being that the latter is applied to spiking neurons.

Echo state nets consist of a dynamic reservoir, a recurrent net with fixed weights. The output units can be interpreted as to exploit the different local dynamics of the dynamic reservoir and superimpose them to achieve the desired dynamics in a way that reminds of the composition of Fourier components.

14.3 Linear differential equations and recurrent neural networks

With respect to their dynamical properties, recurrent neural networks may be described as showing fixed point attractors (e.g. Hopfield nets or MMC nets, Chapter 14.4), periodic attractors (e.g. Fig. 14.**1**), or chaotic attractors (Fig. 11.**7**). How is it possible to design a neural network with specific dynamics? One possible solution is given by the Echo State networks (Chapter 14.2). Dynamical systems are often described by differential equations. In such cases the construction of a recurrent network is easier: Any system described by a linear differential equation of order n can be transformed into a recurrent neural network containing n units (Nauck et al., 2003). To this end, the differential

equation has first to be transferred into a system of n coupled differential equations of the order one. This will be shown using the example of a damped oscillator, which is described by $\ddot{x} = -\omega^2 - r\,\dot{x}$, i.e. a second order differential equation with ω being the angular frequency and r the friction. Introduction of the auxiliary variable $\dot{x} = v$ gives the following two equations:

$$\dot{x} = v$$

$$\dot{v} - -\omega^2\, x - r\, v\,.$$

As explained in Chapter 8.10, the easiest way to transfer such differential equations into a neural system is to integrate the equations which leads to

$$x = \int v\, dt$$

$$v = \int (-\omega^2\ r - r\, v)\, dt$$

and in a discrete form to

$$x_{t+1} = x_t + v_t$$

$$v_{t+1} = -\omega^2\, x_t + (1 - r - \omega^2)\, v_t\,.$$

These equations describe a two unit recurrent network with the weight matrix

$$\begin{pmatrix} 1 & 1 \\ -\omega^2 & 1 - r - \omega^2 \end{pmatrix}$$

The net is depicted in Figure 14.**4**. This is a recurrent neural network with nonsymmetrical weights showing damped (r ≠ 0) or undamped (r = 0) periodicity. Such networks can easily be trained using the learning algorithm described in Chapter 12.4 (after Kühn et al 2006).

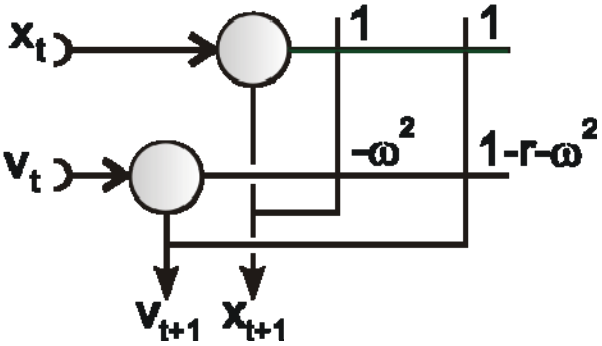


Fig. 14.**4 A recurrent net showing damped oscillations.** ω frequency, r friction

Another simple system is shown in Figure 14.**5a.** This network is endowed with the property of a temporal low-pass filter. This becomes clear when the circuit, as shown in Figure 14.**5b** is redrawn. Due to its direct positive feedback, the second neuroid is endowed with the property of an integrator. The entire system (Fig. 14.**5c**) thus formally corresponds to an I-controller which, as shown in Figure 8.**8e,** can be regarded as a low-pass filter. If the integrator is placed in the feedback loop, the result is

a temporal high-pass filter (Fig. 14.**6**, see also Fig. 8.**8f** and Appendix II). The circuit corresponds to that of Figure 14.**5**, although the output used here is that of neuroid 1 instead. In both cases the time constant is determined by the value of weight k ($\tau = 1/k$). For a derivation of these weights matrices fron the corresponding differential equations see Kühn and Cruse, 2007.

Networks representing differential equations can be trained by applying the learning algorithm described above in Chapter 12.4. If the network contains the necessary nonlinearities, aölso nonlinear systems as for example the van-der-Pol oscillator can be learnt.

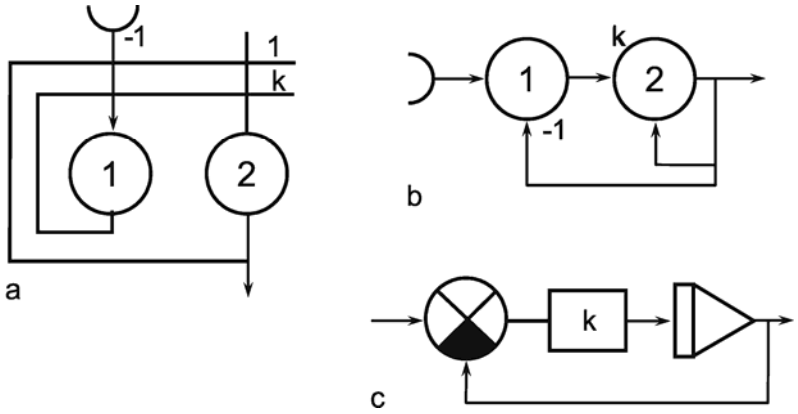


Fig. 14.**5 An asymmetrical recurrent network with the properties of a temporal low-pass filter (a).** (b) and (c) show the same system but are redrawn to allow comparison with figures of Chapter 8. For easier identification the two units are numbered in (a) and (b)

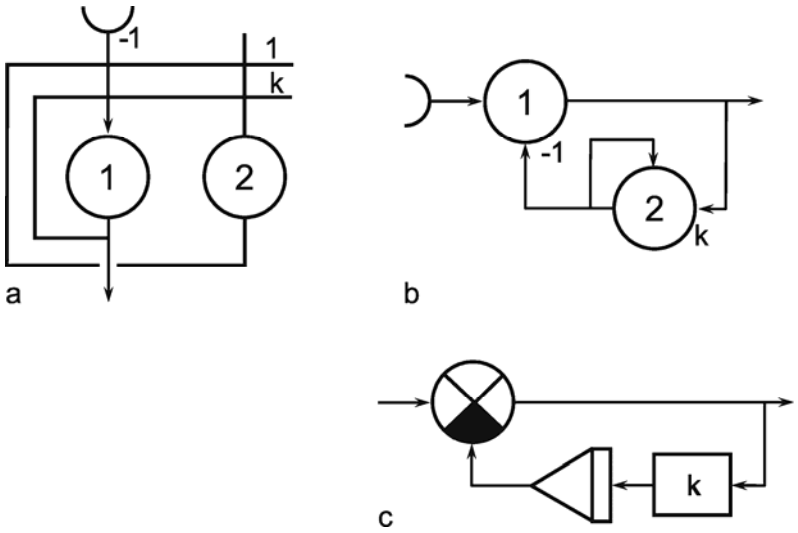


Fig. 14.**6 An asymmetrical recurrent network with the properties of a temporal high-pass filter (a).** (b) and (c) show the same system but are redrawn to allow comparison with figures of Chapter 8. For easier identification the two units are numbered in (a) and (b)

Any linear differential equation of order n can be transferred to a recurrent network containing n units which permits to construct networks of very different dynamics. In contrast to feedforward networks, recurrent nets show dynamic, i.e. time varying properties as do for example temporal filters. This is particularly true for systems with nonsymmetrical weight distributions

14.4 MSBE Nets

A specific type of recurrent neural networks that show fixed-point attractors and that are particularly suited to describe systems with redundant degrees of freedom are the so called MMC nets (see Chapter 14.5). The easiest way to construct such a net is to start with a simpler version. Given a linear equation with n variables

$$\sum_{i=1}^{n} a_i x_i = 0 \ ,$$

for example $a_1 x_1 + a_2 x_2 + a_3 x_3 = 0$. Solving this basis equation for each x_i gives a set of n equations as shown for n = 3:

$$x_{1,t+1} = 0\ x_{1,t} - a_2/a_1\ x_{2,t} - a_3/a_1\ x_{3,t}$$

$$x_{2,t+1} = - a_1/a_2\ x_{1,t} + 0\ x_{2,t} - a_3/a_2\ x_{3,t}$$

$$x_{3,t+1} = - a_1/a_3\ x_{1,t} - a_2/a_3\ x_{2,t} + 0\ x_{3,t}$$

Each of these n equations represents the computation performed by one neuroid. So the complete network represents Multiple Solutions of the Basis Equation, and is therefore termed MSBE net. Different to Hopfield nets, the weights are in general asymmetric (apart from the special case that all parameters a are identical, i.e. $a_1 = a_2 = a_3$), but follow the rule $w_{ij} = 1/w_{ji}$. The diagonal weights, by which each unit excites itself, could be zero, as in the example, or any positive value d_i, if all weights of this equation are further normalized by multiplication with $1/(d_i+1)$. Positive diagonal weights influence the dynamics to adopt low-pass filter-like properties, because the earlier state of this unit is transferred to the actual state with the factor d/(d+1). Indeed, the time constant τ can be determined by $\tau = d-1$ (Makarov et al. 2008). As d_i can arbitrarily be chosen ($d_i \geq 0$), the weights may then not follow the rule $w_{ij} = 1/w_{ji}$ anymore.

Starting this net with any vector **a**, the net will stabilize at a vector fulfilling the basic equation. This means that the attractor points form a smooth, in this example two-dimensional, space. This is another difference to Hopfield nets which show discrete attractor points. Furthermore, there are no nonlinear characteristics necessary.

As for the networks representing differential equations, learning of MSBE ntes is possible using the learning algorithm explained in Chapter 12.4.

14.5 MMC Nets

MSBE networks can be expanded to form an MMC net. MMC nets result from the combination of several MSBE nets (i.e. several basis equations) with shared units. As a simple example, consider the two basis equations (1) $x_1 + x_2 + x_3^{(1)} = 0$ and (2) $x_3^{(2)} + x_4 + x_5 = 0$. As described above, these basic equations lead to 2 times 3 equations sharing variable x_3.The two corresponding equations

$x_3^{(1)} = -x_1 - x_2 +$ and $x_3^{(2)} = - x_4 - x_5$

can be summed to

$x_3 = -½ x_1 - ½ x_2 - ½ x_4 - ½ x_5$.

Now we have five equations, where x_3 represents the mean of $x_3^{(1)}$ and $x_3^{(2)}$.

$x_1 = - x_2 - x_3$

$x_2 = - x_2 - x_3$

$x_3 = -½ x_1 - ½ x_2 - ½ x_4 - ½ x_5$

$x_4 = x_3 - x_5$

$x_5 = x_3 – x_4$

A more interesting example will be shown below. Such MMC nets can be used to describe landmark navigation in insects and as a model describing place cells found in rodents (Cruse 2002). However, the principle can also be used to represent the kinematics of a body with complex geometry including extra degrees of freedom (Cruse and Steinkühler 1993, Steinkühler and Cruse 1998). As an example, we will use a three-joint arm that moves in two-dimensional space, therefore having one extra degree of freedom (Fig. 14.**7a**).

The properties of this type of network can best be described by starting with a simple linear version (Fig. 14.**7b**). As we have a two-dimensional case, the complete net consists of two identical networks. The output values correspond to the Cartesian coordinates of the six vectors shown in Figure 14.**7a**, the x coordinates of the vectors given by the net shown with solid lines, the y coordinates by dashed lines. To obtain the weights, vector equations drawn from the geometrical arrangement shown in Figure 14.**7a** are used as basis equations. This means in this case there are several basis equations possible. For example, each three vectors forming a triangle can be used to provide a basis equation (e.g. $\mathbf{L}_1 + \mathbf{L}_2 - \mathbf{D}_1 = 0$). As a given variable (e.g. $\mathbf{L}_1$) occurs in different basis equations, there are several equations to determine this variable. The idea of the MMC net is to actually perform all these calculations in parallel, take the mean of each variable and use these mean values of each variable for the next iteration; therefore the name MMC: Mean of Multiple Computation, a principle exploiting the redundancy of the task.

When this network obtains as input the coordinates of vector **R** (which determines the position where the arm should point to), for example, after relaxation it describes a geometrically possible solution for the arm. In this way, the inverse kinematic problem (Chapter 14.5) is solved.

In this linear version, the lengths of the arm segments are not fixed, which corresponds to a robot with rotational and telescopic joints. To maintain constant segment lengths and thus constrain the arm to have only rotational joints, nonlinear elements have to be introduced. This is shown in Figure 14.**8** where the joints angles are calculated using the coordinate values (P). The corresponding inverse calculation is performed by (Q). Furthermore, rotations of the local coordinate systems are necessary (T). In this version, the input and the output vectors consists of five components, two describing the position of the end effector in Cartesian coordinates (R_x, R_y), and three describing the joint angles (α, β, and γ). All geometrically possible arm configurations form a three-dimensional subspace of this five-dimensional vector space. Each point of this three-dimensional subspace can serve as an attractor to which the net can relax. This shows again that the attractor points form a smooth space, in contrast to Hopfield nets where the attractors comprise discrete points. Such a relaxation occurs for each arbitrarily chosen combination of three or less input components. This means that even if only

one component is specified at the input, a five-dimensional vector appears at the output which, first, fulfils the input specification ("pattern completion") and, second, describes a geometrically possible arm configuration.

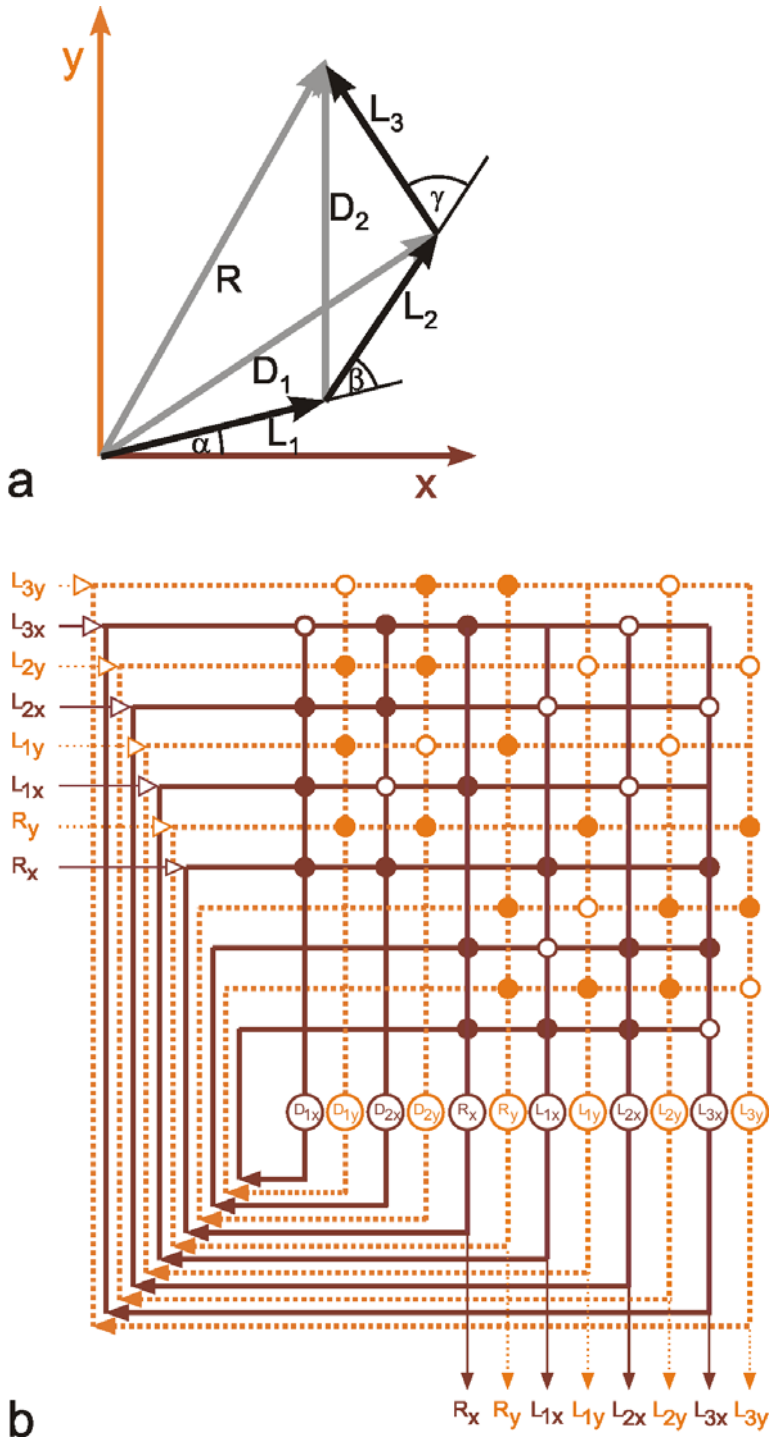


Fig. 14.7 **(a) An arm consisting of three segments described by the vectors L1 L2, and L3 which are connected by three planar joints.** The joint angles are α, β, and γ. The position of the endeffector is described by R. Furthermore, two additional vectors D_1 and D_2, describing the diagonals, are shown. (b) The linear version of the MMC net. The output values represent the components of the vectors R, L_1, L_2, and L_3 (lower line). Solid lines represent the subnet for the x components, dashed lines that for the y components. To calculate the inverse kinematics, the components of R have to be provided as input for the net. However, any other combination of input values could be chosen instead. The possible input values are shown on the vertical column at the left. For the chosen input, the recurrent line is suppressed, and only the input value is fed into the net, symbolized by the small arrowheads.

Thus, we can, for example, solve the inverse kinematic problem by giving R_x and R_y as input values. Or, we can solve a mixed problem by, for example, specifying R_x and α. Both tasks are underdetermined and the system by itself finds a solution. The actual solution depends on the starting configuration of the arm. If R_x, α, and β are given at the input, the situation is determined and the net provides the remaining values R_y and γ. Thus, the inverse, the direct, and any mixed kinematic problem can be solved. Furthermore, any special constraints of the variables (e. g., joint limits) can be introduced. In this way, the MMC-type network can be used as a kinematic inverse model (Chapter 14.6) which allows for very different solutions of the redundancy problem and which can easily cope with limitations of the joint space or the workspace. The net can even solve the task when R_x and R_y values are given that describe a position outside the workspace of the arm. In this case, the arm tries to reach this position as far as possible and points in the direction of the position determined by R_x and R_y. MMC nets can, therefore, be used to solve different tasks in motor control, but, comprising an internal model, also be used to perform “imagined” movements, a prerequisite for the ability to plan ahead. Furthrmore, such body models may form the basis of (visual) perception of movements of other bodies (Cruse 2003, Schilling and Cruse 2008).

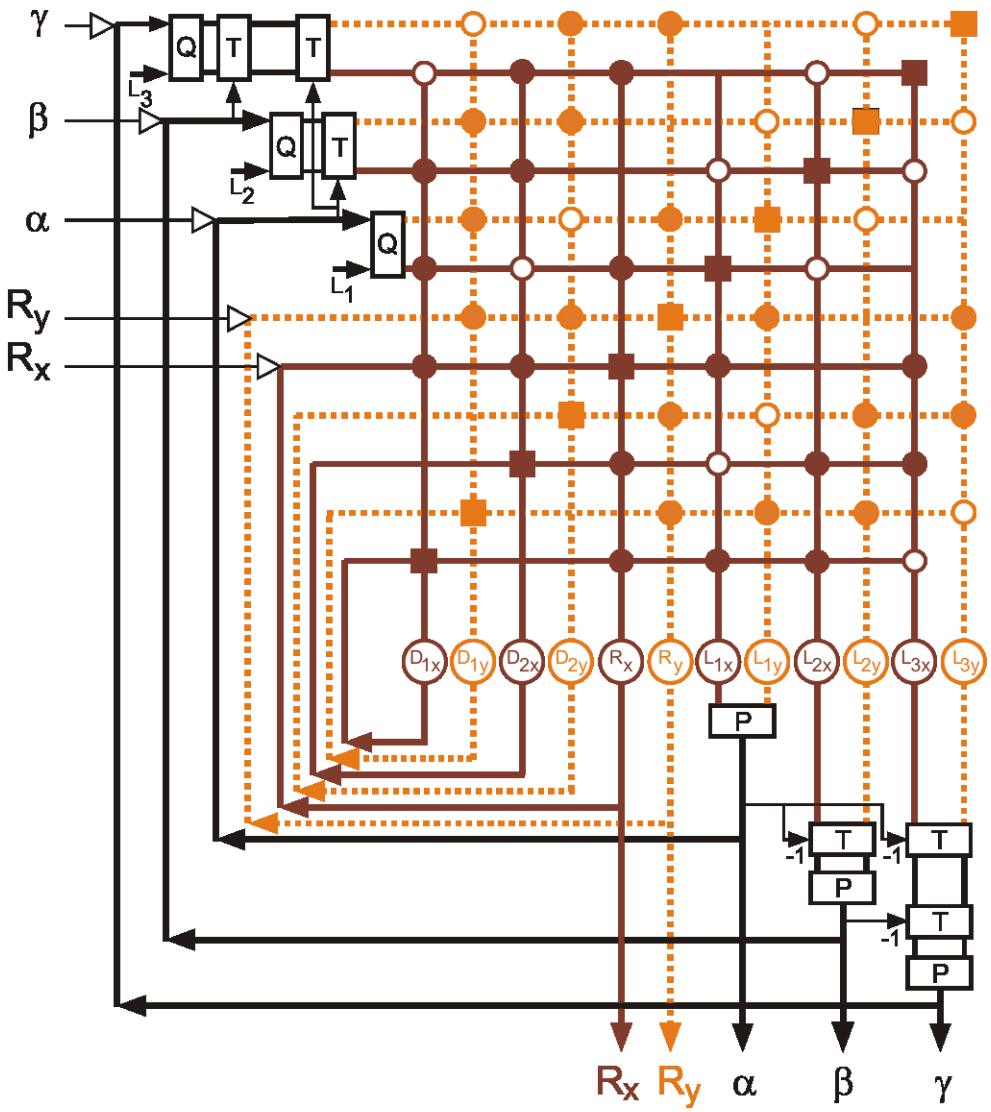


Fig. 14.8 **The MMC net as shown in Fig. 14.4b, but with nonlinear subsystems in the feedback loops.** In this case the input and output vectors contain the components R_x and R_y, describing the position of the endeffector, as well as the joint angles α, β, and γ. The nonlinear systems P, T, and Q are explained in the text.

An MMC net is shown that relaxes to a stable state corresponding to a geometrically correct solution even when the input does not fully constrain the solution. The underlying idea is that several geometrical relationships are used to calculate the same value (e. g., angle α) several times in parallel. The final output is then obtained by calculation of the Mean value of these Multiple Computations (MMC). These mean values are then fed back for the next iteration. Applied to a body model, this MMC net serves as an internal model to provide the inverse kinematics. It can exploit the full range of geometrically possible arm configurations and permits the introduction of variable constraints. Representing a holistic model, there is no separation possible between a motor system and a sensory system.

14.6 Forward Models and Inverse Models

Using recurrent neural networks like Elman/Jordan nets or MMC nets for the control of behavior, i.e. use their output for direct control of the actuators, would mean that no online feedback via the environment was applied (Chapter 8, open loop control). However, feedback control is helpful in a nonpredictable world. A complication can occur in an open loop or closed loop control system, (which was not addressed in Chapter 8) when the controlled variables and the desired variables are different in nature. This can be illusztrated for the case, for example, when the task is to control the movement of a two- (or three-) joint arm in a plane. Let us assume that the tip of the arm has to be moved along a trajectory given in x, y (e. g., Cartesian) coordinates of the plane (Fig. 14.**9**) and that, within the neuronal system, the task is defined using these x-y coordinates; the motor output of the system has, however, to be given in joint angles. Therefore, a coordinate transformation is necessary that translates the task from x-y coordinates into angle values. This transformation ($\mathbf{x} \rightarrow \alpha$) is called the inverse kinematics requiring an “inverse” model (IM) of the arm. This task is solvable if the number of degrees of freedom in workspace and jointspace is the same (e. g., two coordinates x, y; two joints α, β, see Fig. 14.**9a**), but there is no general solution for the so-called redundant case. This occurs when the jointspace has more degrees of freedom than the workspace (Fig. 14.**9b**). In this case, a given tip position could be reached by a number of different arm configurations (α, β, γ) and therefore no unique solution is possible. In this event additional constraints have to be introduced. Examples are minimization of energy consumption or of movement time. The transformations considered here could

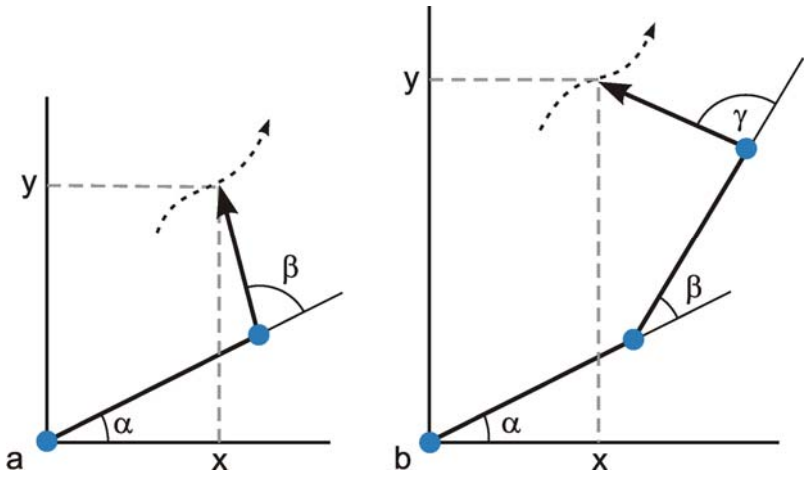


Fig. 14.**9 A two-joint arm (a) and a three-joint arm (b) moving in a two dimensional (x-y) plane.** When the tip of the arm has to follow a given trajectory (dotted arrow), in the redundant case (b) an infinite number of joint angle combinations can solve the problem for any given position of the tip.

be simple static transformations that concentrate only on the geometrical conditions and are then called kinematic or they might include the action of forces and are then called dynamic. In the following we will not regard this difference and therefore use the general term internal model. (Note that the term “dynamic” is used in different ways. In physics, as used in this section, a dynamic system deals with forces (referring to the litteral meaning of the word). Often, however, the term dynamic simply refers to the property of a system to depend on time as is the case for a low pass filter, for example).

Several architectures have been proposed to control the movement of such a redundant (or nonredundant) arm. In the simplest case, the inverse model translates all x-y coordinates into a set of fixed angular coordinates, for example using a multilayer perceptron (Fig. 14.**10a**). Thus the problem of controlling a system with redundant degrees of freedom is solved by disregarding the possible alternative solutions the system could use in principle. In motor control this is often formulated in a positive way such that the system exploits “synergies” between different joints, i.e. some joints cannot be moved independently, but only according to fixed rules (Bernstein 1967).

The output values of the net shown in Fig. 14.**10a** may directly drive the muscles, but if disturbances in the external world are to be expected, negative feedback loops have to be considered. The simplest solution is to apply a separate feedback controller for each joint (Fig. 14.**10b**) as described in Chapter 8 in more detail.

An alternative (or additional) way is to control the result procduced by the motor output at a higher level, often called the task level, i.e. on the level of the x-y coordinates. To this end, in the feedback loop, the angle values of the joints have to be transformed to position values of the tip. This transformation ($\alpha \rightarrow \mathbf{x}$) is called the direct kinematics or, more generally, the forward model of the body (FMb), and is solvable for an arbitrary number of joints. The x, y coordinates can then be compared with the desired coordinate values (Fig. 14.**11a**). The error signals given in Cartesian coordinates have then to be transformed into angle values (joint coordinates). In other words, an inverse model of the body (IMb) is required, too. This architecture is shown in Figure 14.**11a**, where FMb and IMb may represent feedforward (e. g., three-layer) networks, computing the direct and inverse kinematics solutions, respectively. (As mentioned, in the redundant case, IMb has to represent a particular solution).

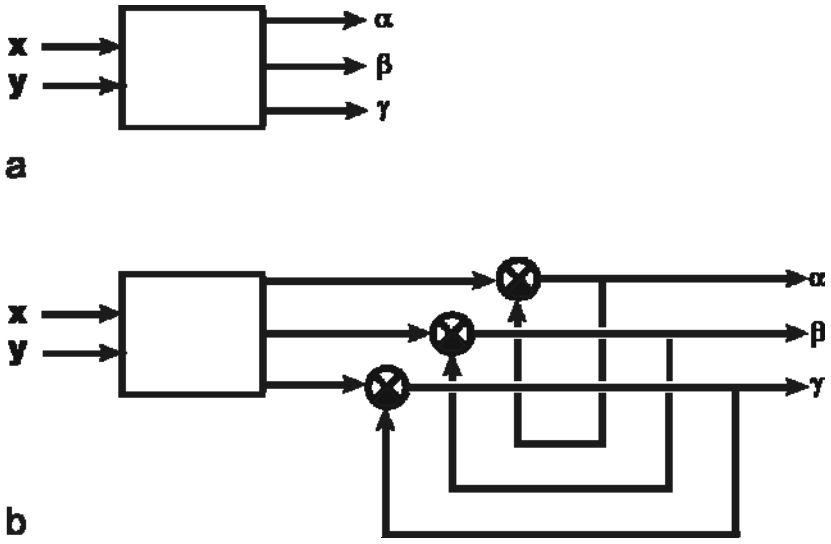


Fig. 14.**10**. **Coordinate transformation** required when the task (for example to move an arm, see Fig. 14.**9**) is given inCartesian coordinates, x,y, and the motor control system operates on joint angles (a). The output values might be used as reference values for separate feedback controllers (b)

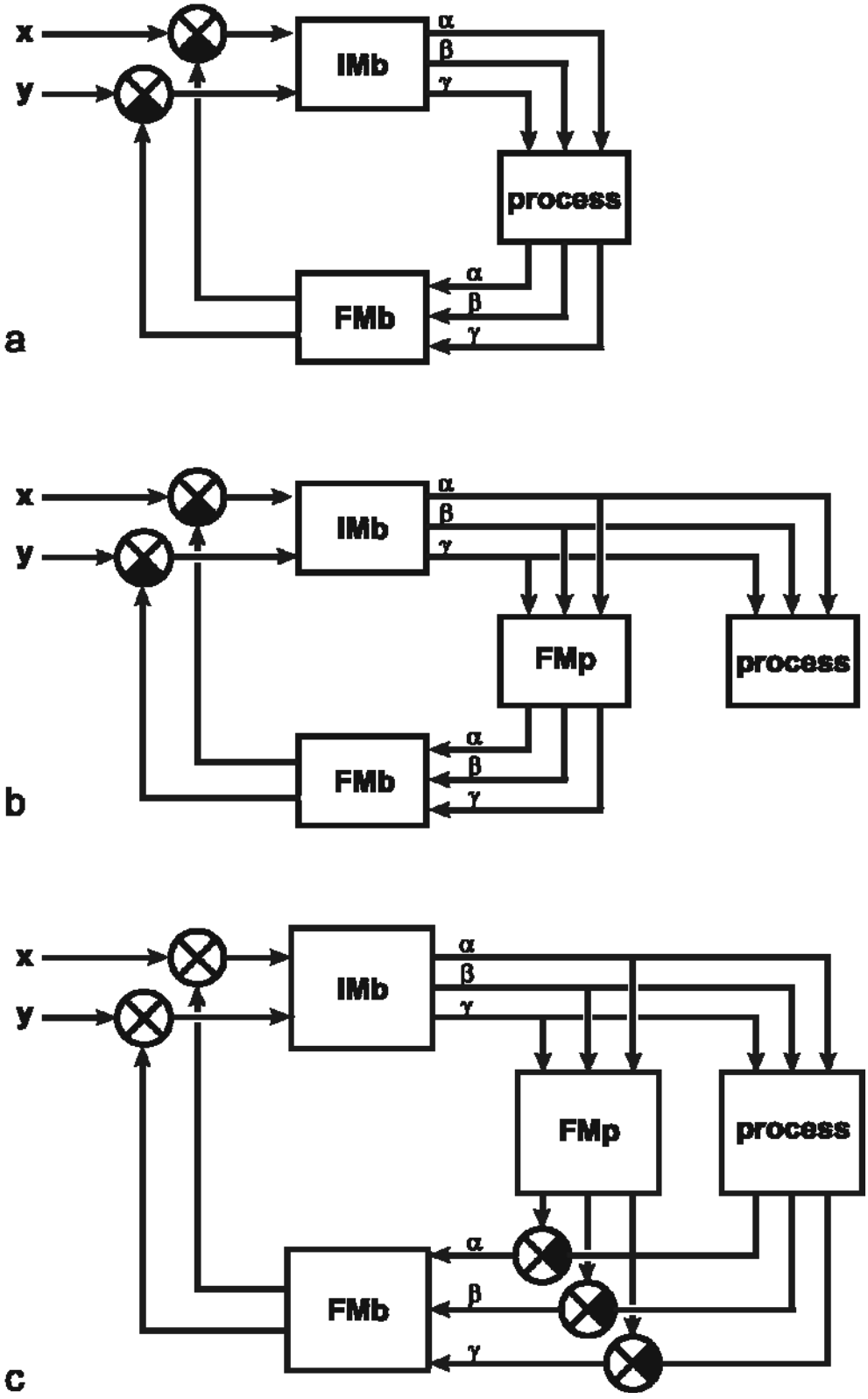


Fig. 14.**11 Three ways of controlling the movement of a redundant (or non redundant) arm as shown in Fig. 14.9.** FM, IM: networks solving the direct and inverse kinematics (or dynamics), respectively. FM: forward model of the process (FMp) or the body (FMb). IM: inverse model of the process (IMp) or the body (IMb). In (c) the FMp corresponds to the efference copy signals.

The process, not explicitly shown in Fig. 14.**10b**, represents properties of the environment, for example the inert mass of an object to be moved. In principle, the process can be controlled with the feedback network given in Fig. 14.**11a**. However, since in biological systems information transfer is slow, this solution is not appropriate for fast movements because long delays can lead to unstable behavior (Chapter 8). A simple solution of this problem would be to expand the system by introducing a (neuronal) forward model of the process (Fig. 14.**11b**, FMp). By this, the behavior of the process can be predicted in advance, e. g., inertia effects could be compensated for before they occur. In other words, the FM predicts the next state of the arm (e.g. position, velocity) if the actual state and the motor command is known. Usually, although depicted separately in Figure 14.**11b**, the FMb and the FMp together are considered the forward model. (The distinction might be sensible because the FMb can be learned once whereas the FMp depends on the actual situation and therefore changes between tasks). If the feedback loop has the same dynamics (e.g. temporal delay) as the process, this circuit would propose an ideal solution. If, however, the FMp is not accurate or if there are disturbances acting on the process (not depicted in Fig. 14.**11b**), the process output has also to be monitored via sensors. Different solutions are possible depending on whether the joint angles are monitored or the end position coordinates are registered directly, for example using vision. Therefore, the additional external feedback loop is omitted in Figure 14.**11b**. A specific solution refers to the proposal of v. Holst and Mittelstaedt (1950, see Chapter 8.9). In this case (Fig. 14.**11c**) the sensor signals are subtracted from the output of the FMp which represents an elaborated version of the efference copy channel (Fig. 8.**15**). The difference is then given to the FMb to determine the error signal in Cartesian coordinates which then is added to the reference input (Fig. 14.**12**). According to v. Holst and Mittelstaedt, it is the Cartesian version of the error signal that is subjectively experienced.

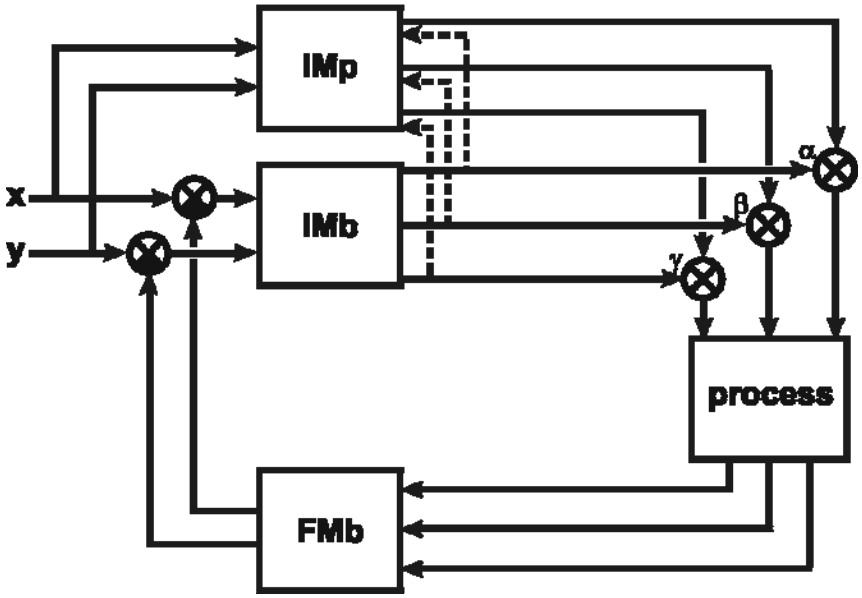


Fig 14.**12** **A way how to train an inverse model** of the process (IM), if a simple (kinematic) inverse model of the body (IMb) and a forward model of the body (FMb) is given. The dashed lines denote the error signals used to train the inverse model

According to Kawato and Gomi (1992) an even faster way to solve this problem would be to introduce an inverse model of the body and the process (IM), which corresponds to the inverse kinematics, but also incorporates the dynamic properties (Fig. 14.**12**). Given the desired new state of the arm, this model does provide the necessary motor command (**x** -> α) to reach this state. Using such an inverse model, the process could be controlled without the need of an internal feedback loop.

How can we obtain a network for the inverse model? Kawato and Gomi (1992) discuss several possibilities and proposes the following circuit. Starting with the simple network shown in Figure

14.**10a**, a naive net to learn the IM is connected parallel to the IMb model, which might represent a simple inverse kinematic solution (Fig. 14.**12**). At the beginning of learning, the output of the IM is weighed with zero. Thus, due to the slow feedback controller, only slow movements can be performed. However, during the movements, the IM net can be trained by using the output of the IMb module as error signals which can, for example, be backpropagated through the IM net, thereby improving its performance. Thus, with time, the output of IM could be attributed higher weights, and the IM module can exercise an increasing control over the process. (One measure to determine the relative strength of the IM influence might be something like the inverse of the averaged error). After learning, the feedback signals are no longer necessary for normal performance. However, if the properties of the system change, they are needed for new training. Of course, negative feedback is still necessary if there are unexpected disturbances. Note that, when the task is defined by a given starting configuration of the arm and a given endpoint to be reached, the trajectory has to be specified, too, for learning to be possible.

This arrangement has the advantage that an IM net can also be found for a redundant system, i.e. a system with extra degrees of freedom (e.g. Fig. 14.**9b**) provided that the IMb net is given. However, the latter net provides only one solution depending on its internal structure, i. e., the kind of constraints chosen, and only this solution can, of course, be learnt by the IM module.

Generally, forward models and inverse models are considered separate networks (e.g. Wolpert and Kawato 1998, Mohan and Morasso 2006). Instead of using these distinct networks, a holistic network as for example the MMC net (Fig. 14.**8**) could be applied for both cases. For the case of the body model this has been shown by Schilling and Cruse (2008). This network has the advantage to allow for the control of redundant systems while being able to exploit all geometrically possible solutions. Application of such a holistic body model in addition allows for a solution for the sensor fusion problem: In biological systems, but also in artificial systems, the same situation is often monitored by different sensors. In the above example the arm position is monitored via joint angle sensors, but could also be monitored by sensors measuring muscle length and/or by visual sensors. How to combine this information is termed the sensor fusion problem. A holistic network like the MMC net provides a natural solution, because, as shown in Fig. 14.**8**, it connects both angular data (Fig 14.**8,** α, β, γ) and data given by other (e.g. visual sensors (Fig 14.**8**, Rx, Ry). To include muscle lengths, this network had of course to be expanded by units representing these values.

Such holistic networks might, furthermore, form the basis of perception systems, as for example assumed to be represented by the so called mirror system (Rizzolatti 2005). Like mirror neurons, the units of the MMC net are active when the net is used for motor control or for perception (Schilling and Cruse 2008). This view is supported by a number of experimental results which indicate that the own body model is used to recognize the movements of others (Shiffar 2001).

When a FM of the body and of the process exists, in principle these models could also be used to predict the outcome of actions before they are performed. This ability has been termed imagined action or “probehandeln” (S. Freud), and allows for cognitive capabilities (McFarland and Bösser 1993). However, this capability requires to embed the forward model in a more complex circuit (Schilling and Cruse 2008).

Control of motor output can be improved by applying forward models and/or inverse models of the process. The former is used as predictor of the expected sensory feedback, the latter for the control of the process. Application of holistic networks can simplify the problem.

15 Spatial Coding

Most artificial networks considered usually code the signals in the form of varying excitation of the individual neuroids. This “intensity coding” is an appropriate general solution for artificial neuroids because they have a very good resolution compared to biological neurons. In biological neurons the amount of transferable information is much smaller. Therefore, in biological systems intensity coding can only be used in a limited way for both analog or spiking neurons. This limitation is compensated by application of even more extensively distributed architectures than are used in usual artificial neural networks. This can be done for example by application of range fractionation. In this case, neurons still use intensity coding, but via nonlinear characteristics each neuron covers only a limited section of the complete range to be transmitted. Another way is to apply spatial coding. Some examples will be explained below.

Spatial to spatial: The most obvious example of spatial coding is given in situations in which already the input information is represented in a spatial domain, as is the case for retinal images or for somatosensory input. In these cases we find spatial low pass filters and spatial high pass filters as described above (Chapter 10.1 – 10.3).

Coarse coding: One might assume that a spatial coding system using discrete units may have the disadvantage of limited spatial resolution. This is indeed the case if the units are not connected but form strictly separated channels. If there is an overlap between neighboring units, one might assume that the spatial resolution is even worse. However, the opposite is the case. A spatial low pass filter, providing such an overlap, allows to transmit also intermediate values. These are coded in the form of excitation ratio of neighboring units (see Chapter 10.1). Therefore, “coarse coding” (Rumelhart et al. 1986, Baldi and Heiligenberg 1988) is a way to increase the spatial resolution beyond the discrete nature of neuronal arrangements (a special example is color vision: we have only three color units but can see a huge number of intermediate colors). If the weight function of this low pass filter is chosen appropriately, in principle a completely smooth information transfer is possible.

Temporal differences to spatial difference: Another example is given by the system which has to detect the direction of a sound source. The information is given (in part) by the temporal difference between the signals received by both ears. The neuronal system transfers this temporal difference into a spatial difference (Fig. 15.**1**), using a simple feedforward network and coincidence detectors (e.g. multipliers). The system exploits the fact that in neuronal systems information transmission is relatively slow. This circuit allows very high temporal resolution (e.g. 0.03 ms in humans).

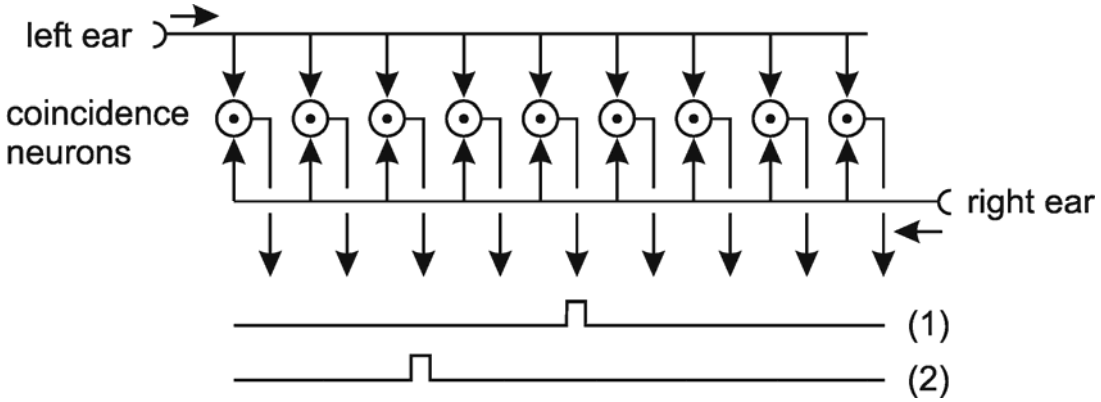


Fig. 15.**1 Calculation of temporal difference using spatial coding**. If the stimulus reaches right ear and the left ear at exactly the same time, the coincidence neurons will be excited that are positioned in the center of the chain (1). If the left ear, for example, is stimulated earlier, the position of the excited coincidence neurons will shift to the left side (2).

From intensity coding to spatial coding: A Kohonen network with a low number (e.g. one) of input channels and a high number (e.g. 10) of output channels can be considered as to transform an intensity coded signal into a topologically coded neural chain. The output units code the information in the spatial domain (see Fig. 13.**2**). The distribution of the output signal could of course be sharpened by application of a subsequent WTA net.

Calculation of mean values: The coarse coding principle mentioned above can also be exploited in another way. Assume that several (e.g. n) values are represented, each by its neuronal chain, with one unit of the chain excited. The task is to determine a mean value of these n values. (for example when the same value has been measured by different sense organs). For simplicity of explanation assume further that the chains all consist of k units.

To calculate the mean of n values represented spatially on n neuronal chains with k units, each of these n chains (Fig. 15.**2**, Spatial Input 1 and 2) has to be connected to a new chain with k units (Fig. 15.**2**, Spatial Output). However, the connections are not 1:1 but represent a spatial low pass filter with a broad weighting function covering the whole range of the chain such that e.g. excitation of the leftmost input unit gives a non zero output to the rightmost unit of the next layer. The form of the weight function must not be linear but has to decrease with a steeper slope near the center and a slope getting continuously smaller to the sides. In this case the superposition of all weighted inputs gives a function the maximum of which corresponds to some kind of mean value (it rather corresponds to the mode value than to the arithmetic mean). If this output function is given to a WTA net with k units, the maximum, i.e. the mean can be clearly detected. This net has a further advantage: Input values far away from the “center of gravity” do not contribute a lot, which means that this calculation is not very sensitive to outliers. In particular, this is true in the case when one value is not available at all, i.e. when all values of this chain have zero excitation. This means that no knowledge is necessary as to how many values are used to determine the mean value compared to a system using intensity coding where the sum has to be divided by n.

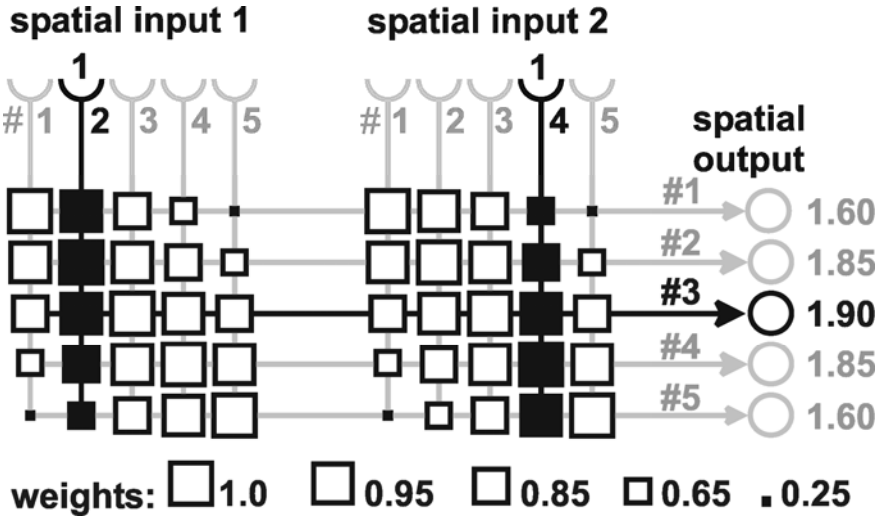


Fig. 15.**2 Calculating the mean value of two spatially encoded scalar**. Output neuroids are drawn as open circles, input neuroids as semicircles. Arrows represent exciting connections, the weights are represented as squares. Two spatially encoded scalars "2" and "4" are given by an excitation of 1.0 of the respective neuroids (black pathways, input neuroid #2 and #4, respectively). In that case, the highest activation can be found in output neuroid #3. If necessary, the mean value can be computed by a winner-take-all network (not shown) at the right side (after Linder 2005).

Deviation, confidence measure: This network can also be used to determine a measure for the deviation between the different input values. The more compact the distribution of the original input

values, the higher is the maximum obtained by this superposition. This means that the absolute value of the maximum excitation can also be used as a measure for the deviation of the input values from the mean. A good match between the input values gives a high maximum value. Therefore, the value of the maximum can be interpreted as a confidence measure for the mean value (see circular statistics). If in a subsequent calculation using this mean value is applied, the confidence measure could be used to weight the contribution of this mean value.

Multiplication: A simple solution to multiply spatially encoded input values with a constant factor is to arrange the connection matrix between input and output layer in a manner corresponding to the desired factor: for a factor of 0.5, every unit in the output layer is connected to two succeeding units in the input chain; for a factor of 2, only every second output unit gets connected to one of the input units. A proposal for multiplication of two variable inputs is given by Hartmann and Wehner (1995).

Temporal signal is integrated and transformed to a spatial output signal: The limited working range of real neurons becomes particularly obvious when a signal has to be integrated. A solution for the task of mathematically integrating an incoming signal has been shown in Chapter 4.8. However, if this task had to be solved by real neurons, because of their limited resolution (Chapter 8.7) only a limited accuracy is possible. Astonishingly, such integrations are performed, for example in navigation of ants, by an exactness of less than 1 %. How is this possible?

An elegant solution to cope with this problem is given by the Hartmann-Wehner integrator (Fig. 15.**3**). The distance moved by the animal (or to mention another example, the movement of a limb) is given by an intensity signal. The speed is registered by two directional sensitive sensors, one measuring movement in one direction (s^+), the other measuring the movement in the opposite direction (s^-). Integration of the sum of both signals provides the position of the limb. Hartmann and Wehner (1995) proposed to integrate these signals by using a spatially distributed system. In this architecture, the neuroids forming the integrator are ordered along a linear chain (Fig. 15.**3**). There is an input neuroid s^+ at one end of the chain and a second input neuroid s^- at the other end.

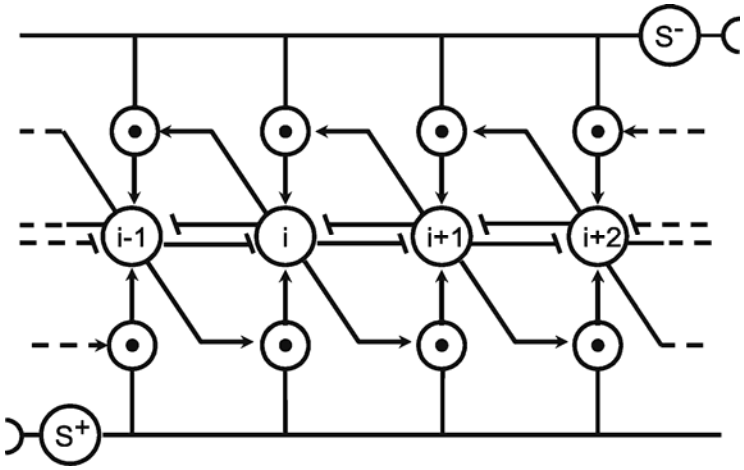


Fig. 15.**3 A simple network for spatial integration.** The input unit s^+ monitors the movement in one direction, the unit s^- that in the opposite direction. By excitation of s^+, the position of the excited neuroid of the chain is moved to the right, whereas it is shifted to the left, if s^- is excited. Therefore, the position of the excited neuroid represents the integral of the input signal s

The system is built of simple on-off units, i. e., neuroids with Heaviside activation functions. It is further assumed that the velocity signals are coded in the form of impulses similar to spiking neurons. Therefore, a high positive speed means that in many time units neuroid s^+ is excited, whereas, for a low positive speed, the probability of s^+ being excited is small. Each neuroid of the chain is self-

excitatory (not shown), i. e., if it was excited once and no further inhibitory influences occur, it will remain excited. What are the connections between these neuroids? Each neuroid of the chain inhibits both direct neighbors. The neuroids can be excited by both sensory units. These excitatory connections are stronger than the inhibitory ones, such that the inhibitory effect is overridden by an excitatory input signal. Finally, the excitatory input is gated by an input from the left neighbor for the s^+ neuroid and, correspondingly, from the right neighbor for the s^- unit (illustrated by the multiplication symbols).

How does this system work? Assume that, at the beginning, the i-th neuroid of the chain is already excited (this corresponds to setting the value of the integrator at time t_0). If an excitatory input arrives, say from the s^+ unit, the right neighbor, neuroid i + 1, will be excited, and the excitation of neuroid i will, as a consequence, be suppressed. If another impulse from s^+ arrives, again the next neighbor to the right, neuroid i + 2, will be excited and it will, in turn, inhibit neuroid i + 1, its left neighbor. Thus, every impulse from the s^+ unit shifts the activity peak of the chain by one unit further to the right. If, on the other hand, the limb is moved in the opposite direction, the s^- unit will be excited and, correspondingly, the excitation peak will be shifted to the left. Thus, the position of the peak on the chain represents the limb position and, therefore, the result of the integration of the velocity inputs.

The central units in Fig. 15.**3**, that represent the actual position comprise yes-or-no units a property which makes them relatively error tolerant. A particular advantage of this architecture using "moving activity patterns" is that it can also be easily used to represent cyclic variables as e. g. angular values. This can be done by simply connecting both ends of the chain to form a circle (Hartmann and Wehner 1995). In this case, the problem usually occurring when cyclic variables have to be treated by analog values is solved naturally. The precision of such a distributed integrator could be increased simply by increasing the number of units in the chain. Furthermore, application of the coarse coding principle (Chapter 10.1) in the form of the neural field approach (Chapter 11.1) may be another way of improving the resolution of this system.

Could you, as an exercise, design a similar network such that the value to be represented is not given by the position of the excited neuroid i+1, but by excitation of all neuroids 1, 2, ... , i ? Note that the net should be able to increase and to decrease the value being represented. An answer is hidden in Fig. 15.**5**.

Summation: Summation can be seen as the computation of a mean value (see Calculation of mean values), followed by a multiplication with a constant factor, which corresponds to the number of input values considered for determination of the mean value. A network for summation can thus be assembled from previous cases (i.e., Calculation of mean values and Multiplication).

Another possibility for the summation of two values is given by a combination of two of the aforementioned Hartmann-Wehner integrators, say I1 and I2 (Fig. 15.**4**) as shown by Linder (2005). They are connected in such a way that each unit of I2 which codes for a negative value is connected with the s- input neuron of I1 and the s+ input neuron of I2. Vice versa, each unit of I2 coding for a positive value is connected to the s+ input neuron of I1 and its own s- input neuron. Since the region at zero is not connected to any input neuron, the system stabilizes as soon as I2 represents a value of zero. At this time I1 contains the spatial representation of the sum of both values formerly represented in I1 and I2. This architecture can be nested for the summation of more than two values. In each case, the number of values to be summed has to be known in advance.

Subtraction: While subtraction can be done similar to addition by a combination of integrators, there are two alternatives (Linder 2005). In the examples described above, a neural chain is used to code a value in a way that only one unit of the chain is excites and the position of this unit within the chain represents the value. As one alternative, a complete set of neighboring units might be excited and the number of excited units represents the value. In this case, a subtraction of value a from a value b (i.e.:

a - b) can be performed when all excited units representing the value b inhibit in a 1:1 connection the units of the chain representing the value a. The number of the units of chain b which survive this inhibition correspond to the value of (a – b).

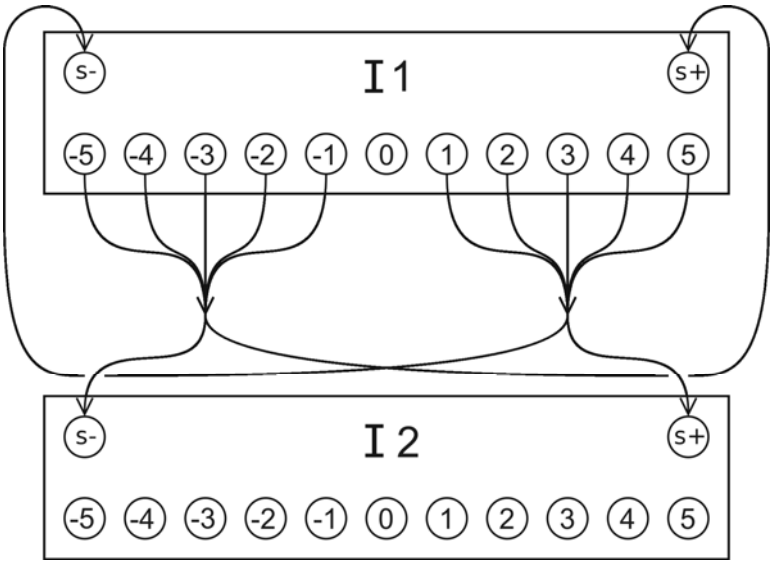


Fig. 15.**4 Circuit for the summation of two spatially encoded values**. Neurons are drawn as open circles, arrows represent exciting connections. Each box (I1 and I2) represents a Hartmann-Wehner Integrator (see Fig. 15.**3**). The left and right shift neurons are displayed as s- and s+, respectively. Two values are initially represented in I1 and I2. After stabilization, the sum of both values is represented in I2. See text for an explanation (after Linder 2005).

Another alternative is obtained if activation is assumed to spread in each neuronal chain unilaterally: We employ two chains of neuroids situated next to each other. One codes for the x value, the other for the y value (Fig. 15.**5**). We assume the neuronal chains to represent exactly the same input space, such that for every neuroid the neighboring neuroid in the opposite chain codes for the same range of data. Let us assume that neuroid D (an intensity coded unit) receives the positive part of the difference x - y, and the corresponding neuroid D' receives the negative part of the difference x - y. Furthermore, we assume each neuroid of one chain to inhibit (small filled circles in Fig. 15.**5**) all axonic branches of the neuroids in the other chain that are located towards one common direction (e.g., upwards in Fig. 15.**5**). Finally, we assume unidirectional exciting connections between directly neighboring units within each chain (Fig. 15.**5**, upward arrows).

What are the properties of this network? If directly opposing neuroids are activated in each chain (i.e., x = y), they reciprocally inhibit each others output and the upward spread of activation. Therefore, none of the output neuroids is excited. If, on the other hand, $x \neq y$, both chains are activated at distinct locations (in the example of Fig. 15.**5**: unit 2 and 5'). While the activated neuroid on one side (left side in Fig. 15.**5**) can promote its activation in upward direction, the spread of excitation along the neuronal chain is inhibited on the other, in our example the right side. This spread of activation is continued until the activation reaches the neuron next to the single activated neuroid in the other chain (unit 5 in Fig. 15.**5**), which inhibits further spread of activation. Thus, on the left side, neuroids 2, 3, 4 and 5 are activated (as the spreading of activation takes time, the properties of this network in the time domain qualitatively correspond to that of a low pass filter). As neuroids 5 in both chains suppress each others output, only units 2, 3 and 4 promote their activation to the output. The activation of neuroid D (3) is therefore proportional to the difference between x (unit #5) and y (unit #2). The output unit D' does not receive any input in this case. Since the architecture is symmetric, the outcome is reversed for reverse

input constellations: if x < y, the activation of the output unit D' is proportional to the difference, while D remains inactive. This architecture therefore performs the subtraction of spatially encoded values.

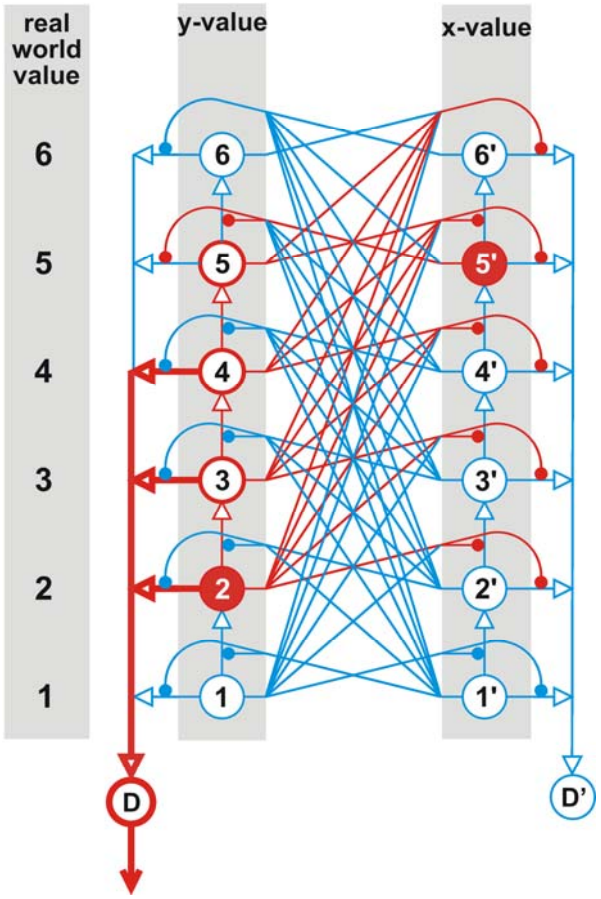


Fig. 15.**5 A neural network for the subtraction of two spatially encoded values x - y**. Neuroids are drawn as big circles, lines ending in little filled circles represent inhibiting connections. Exciting connections are drawn as arrows. Each unit represents an angular value as indicated on the left. On both sides, the position of the filled neuron represents the y value to be subtracted. Arrows indicate exciting synapses, circles inhibitory synapses. D and D' represent intensity coded output units, each receiving input from every neuroid in its chain. If one neuroid coding for the y value is excited (#2), activation is spread upward to the position opposite to the activated neuroid of the other chain (#5'). Excited elements are shown by red lines. Activation of output unit D is proportional to the absolute value of the difference between x value and y value (and activation of D' for the absolute value of the difference between y value and x value, respectively) (after Linder 2005).

In a biological neural network spatial coding allows better resolution compared to intensity coding, but requires more neurons. Ways are explained as to how specific calculations can be performed using spatially encoded networks.

16 Animats

As mentioned earlier, single neuroids or groups of neuroids may be considered as controlling separate behavioral units. These systems may be called experts or modules. Via the combination of such subsystems, models can be constructed which are capable of simulating the cooperation of different

behavioral units and thus are the origin of more complex behaviors. This is of special interest when we study (and simulate), not a single type of behavior, but the question of how the different modes of behavior an animal is capable of performing are stored in the central nervous system, and how one of these is selected and recalled ("action selection"). In the field of "Artificial Life," such systems are used to control the behavior of "autonomous" robots which have become called animats (Wilson 1987).

The classical view of how an "intelligent" system might be constructed could be described roughly as follows (Fig. 16.**1**): First, information from the environment is received by sensors. These data are then fed into an internal model. With the aid of this model and the goals inherent in the system, a decision is made in a planning unit as to which behavior will be executed. Following this, the corresponding behavior modul is selected which, in turn, executes the behavior via the effectors (shared by different behavior modules). In addition to this particular approach, which involves a central planning unit and follows the ideas of traditional Artificial Intelligence and earlier ethologists, other concepts are also discussed.

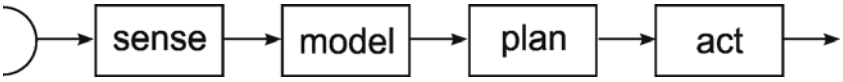


Fig. 16.**1 Schematic presentation showing how in an animal or an artifical system the sensory input may cause a behavior action**

A very simplified assumption is that single modes of behavior are realized in the form of input-output systems (modules) arranged in parallel, whose outputs run via a winner-take-all net. This net ensures that only one mode of behavior is executed, even though several behavior modules may have been activated by stimuli or internal motivation. According to this simplified view, the central nervous system would represent an accumulation of many stimulus-response (S-R) chains. Such a S-R chain is often regarded as a simple system. It should be noted, however, that this is not necessarily the case for a biological system. The identification and isolation of a stimulus signal against a noisy background may pose a formidable problem. Moreover, apart from a few extreme cases, the action following a stimulus requires a behavior sensibly adapted to the specific situation. For instance, grasping an object in itself presupposes a control mechanism of such complexity that so far, the problem has not been solved satisfactorily even for robots. Accordingly, when we speak here of a "simple" S-R chain, this may actually involve a highly complex system.

The parallel, equally weighted arrangement of individual behavior modules is contrasted with the observation that some modes of behavior seem to be "more equal" than others, suggesting that there may be hierarchies. A hidden hierarchical structure might be built into such a system by the way that the various modules arranged in parallel are endowed with different weights; therefore, they enter the decision-making winner-take-all net with varying degrees of intensity. The occurrence of displacement behavior, for example, can be simulated with the aid of a feedforward winner-take-all net (Fig. 16.**2**). Assume that two neuroids (e. g., "fighting," "fleeing") inhibit each other with equal intensity, both inhibiting a third neuroid ("sleeping") only slightly, and assume further that the neuroids "sleeping" is endowed with a constant excitation ("bias"). Normally, the most excited behavior wins (Fig. 16.**2b, c**). If fighting and fleeing are both excited equally, the neuroid "sleeping"- despite its own weak stimulation - may be the beneficiary when "fighting" and "fleeing" exhibit the same level of stimulation (Fig. 16.**2d**). The result becomes even more obvious if this system is followed by a second winner-take-all net.

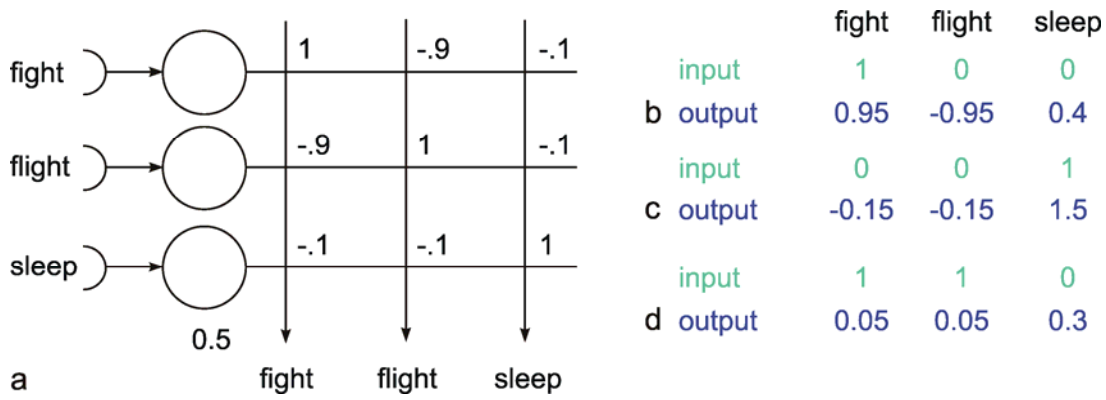


Fig. 16.**2 A winner-take-all all net with unequal inhibitory influences.** (a) The net has to decide which of three behavior modules (inputs fight, flight, and sleep) finally controls the activity, i. e., the output. The sleep modul is assumed to obtain a constant bias of 0.5. If only the fight module(b), the flight module (not shown) or only the sleep module (c) is activated, the output corresponds to the input situation. (d) If fight and flight are equally active, the sleep module will win. This simulates the occurrence of displacement behavior

The "subsumption architecture" proposed by Brooks (1986), is endowed with both a hierarchical and a parallel structure. The individual modules (Fig. 16.**3a**) are arranged in parallel, as in the model described above. This means each module is supported by its sensory apparatus, which initiates corresponding activations. However, the decision which of the different possible behaviors is to be executed is not determined via a winner-take-all net. Rather, there exists a hierarchically layered structure. A behavior module arranged above can, in a state of activation, suppress the module arranged below it, i. e., it can prevent the behavior controlled by the latter from being executed, and instead execute its own mode of behavior.

In Figure 16.**3**, this type of connection is shown by a circle containing the letter S. A simple example is shown in Figure 16.**3b**, which describes the control of a walking leg in, say, a six-legged machine. This leg is provided with two joints, the alpha joint for forward and rearward movement and the beta joint for lifting the leg up and down. During walking the "leg-down" module is continuously excited and influences the "beta position" module to adopt a value corresponding to a leg-down position (here assumed to be zero). Eventually the "leg-up trigger" module receives a signal from a central timer, which is effective when the beta position signal is zero. The trigger signal elicits a positive valued output, which holds for a given time period ("clock"). This signal suppresses the signal from the "leg-down" module. (Suppression in this special case corresponds to simple summation). The output of the "beta position" module drives the motor of the beta joint and excites the "alpha protraction" module. This suppresses the signal from the continuously excited "alpha retraction" module to increase the value of the "alpha position" module in order to move the leg in a forward direction. To obtain an increasing position signal, an integrator is used in the alpha position module. If the alpha protraction module provides a zero valued output, the alpha retraction module moves the position signal to lower values. Figure 16.**3b** shows the arrangement of the modules (which Brooks has described as "augmented finite state machines") in a simplified version.

This subsumption structure has an advantage, among others, that if individual modules were to fail, the system may still survive. Another interesting aspect is that an "evolutionary" development of such a system is easily conceivable. A new module could be introduced without reconstruction of the existing modules. The disadvantage is that this type of influence affecting individual levels has to be planned in great detail and is highly dependent on the internal structure of the levels as well as on the necessary interconnections between the behavioral modes.

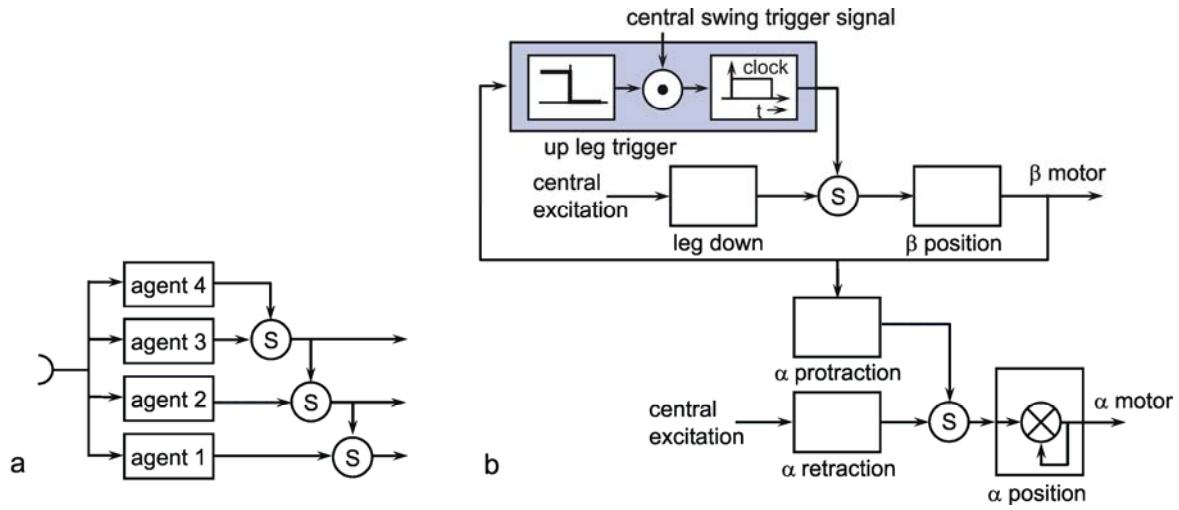


Fig. 16.**3 (a) Schematic illustration of Brook's subsumption architecture. The different modules are arranged in parallel, but determine the actual behavior in a hierarchical manner.** A superior module, if active, may suppress (S) the output of the inferior module and take over the control of the behavior. (b) A specific application of this principle for the control of the movement of a walking leg. The β motor output lifts the leg up or down, the α motor output moves the leg forward (protraction) of rearward (retraction). A central signal to trigger swing comes from a higher center which controls the coordination of the different legs. The swing movement continues for a prespecified time determined by a clock. During walking a continuous central signal excites the leg down module and the α retraction module. The α position module contains an integrator, implemented here using a recurrent positive feedback circuit (after Brooks 1986)

A basic problem concerns the action selection problem, i.e., the question as to how different behaviours can be selected. Within the subsumption architecture, this is done by a strict hierarchical structure. However, action selection might also be based on the intensity of the stimuli. A straight forward solution to select the appropriate action is to apply a recurrent WTA net (Chapter 12.2). As an example, this is illustrated in Figure 16.**5a** for three different behaviors, drinking, eating and courtship. These three behaviours are elicited by the three situations described by water, food and sexual partner, respectively. If only one of the three stimuli is apparent at a time, no connection between the three channels was necessary. If, however, the situation could be more complex and all three stimuli may be present simultaneously, with however slightly different intensity, the WTA net, after some iterations, leads to a clearcut decision (Fig. 16.**5a**). In the numerical example given this is the case after n iterations. The decision process takes the longer the smaller the difference between the input signals.

The models considered so far are greatly simplified in that modes of behavior have been assumed to depend exclusively on stimuli ("sensory-, or data-driven models"). It has been known for a long time, though, that the response of an animal to a specific stimulus may also depend on its internal state, known as motivation, in relation to a specific mode of behavior. We will not go into detail concerning the existing, and greatly varying, definitions of this concept, but mention only two simple versions. Motivation can be equated with the measure of intensity of a response to a specific stimulus. It may also be conceived of as an internal excitation that like an external stimulation, can influence the activation of a behavior module. (See, e.g., bias influence in Fig. 16.**2**). In the first case, the two quantities would be multiplied, and in the second, added together. By introducing a threshold for the initiation of the behavior, the difference between the two types of influence is blurred.

One example of the classic view is that of Lorenz' (1950) hydraulic model (Fig. 16.**4a**). The level of the water corresponds to the intensity of motivation, the weight to the intensity of the stimulus. The higher the motivation and the stronger the stimulus, the more intensive becomes the execution of behavior (the stronger the outflow of water). This corresponds to a multiplicative connection. Figure 16.**4b** shows a translation of this model into a circuit diagram (Hassenstein; in Eibl-Eibesfeldt 1980). If motivation is sufficiently high, though, a behavior can be initiated without a stimulus (vacuum activity). This corresponds to an additive connection. This is not shown in Figure 16.**4** (but see Fig. 16.**6**).

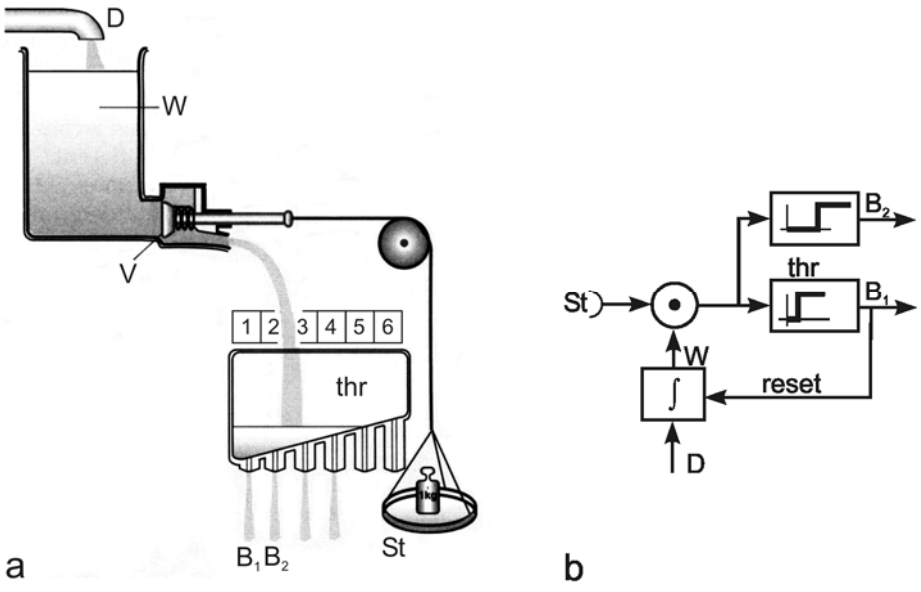


Fig. 16.**4 The „psychohydraulic" model of Lorenz illustrating the possible cooperation of stimulus (weight, St) and motivation (height of the water level, W) to elicit a behavioral response (B) by opening the valve (V).** Different behaviors may be elicited by reaching different threshold values (thr). The classical model further assumes a constant internal "drive" (D) to increase the motivation. Execution of the behavior diminishes the motivation level. (b) shows a simple circuit diagram with similar properties (after Lorenz 1950; Hassenstein, in Eibl-Eibesfeldt 1980)

An open question is how the intensity of motivation can be influenced. In the simplest case, it would always be constant. But it could also depend on internal states, e. g., on a circadian clock or on a physiological state, such as the motivation "hunger" on blood sugar level. The latter case also suggests that it may be influenced by the error signal of a feedback system.

Moreover, as assumed in the specific case of the Lorenz model, the intensity of motivation could also increase with the time that has elapsed since the behavior was last executed. It could decrease, again as assumed in the Lorenz model, once the behavior has been initiated (or even when only particular sense organs, such as taste papillae, have been stimulated).

An easy way to simulate action selection that is not only sensory driven, but depends on internal states, or motivations, is depicted in Figure 16.**5b**, which shows a WTA net as applied above (Fig. 16.**5a**), but is expanded by multiplicatively influencing the stimulus strengths by the corresponding motivations, following the proposal of Lorenz. In this way, also a weak stimulus can govern the behaviour if the corresponding motivation is strong enough.

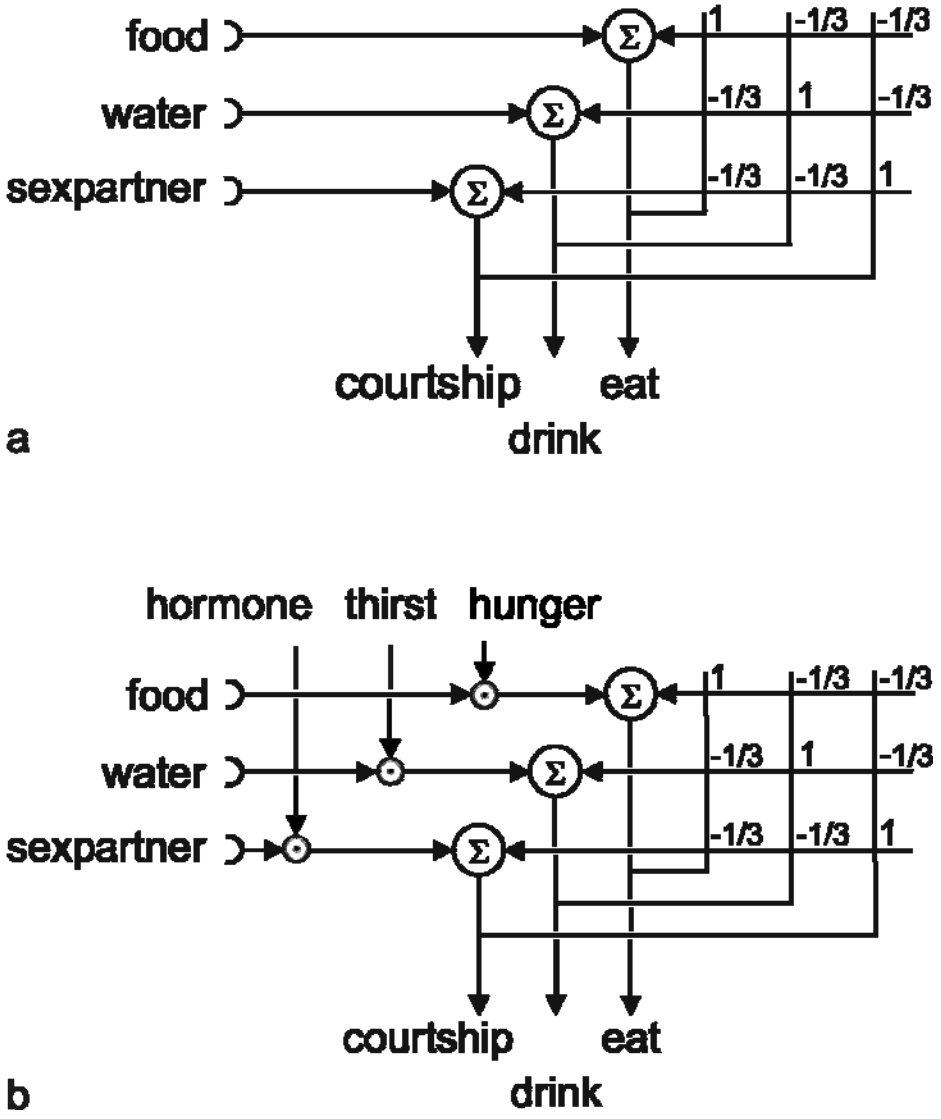


Fig. 16.**5**. **Two simple winner-take-all nets**. (a) The net allows to make a clear-cut decision for one behavior (output, bottom) even if the corresponding stimuli (input, left) are of similar intensity. (b) The decision might depend on internal states (“motivations”, upper input), too.

Before we go on to discuss models which combine "motivated behavior modules," a further principle of cooperation between different modes of behavior needs to be mentioned. Quite frequently, modes of behavior occur successively in temporal terms. These behavior chains cannot be described by a pure feedforward structure. A behavior chain, however, might simply develop because a behavior generates a situation in the environment which serves as an appropriate stimulus for a second behavior. Another possibility would be to assume that the individual modules are linked internally in such a way that, after completion of a behavior, the next module is activated (see Fig. 14.**1**). Between these two solutions there exists the possibility that by activation of one module the threshold for the activating signal of the second behavioral module is lowered, or the intensity of its motivation raised.

A combination of these possibilities is represented by the model of Maes (1991). This has a simple structure but some very impressive features and will be shown here in a somewhat simplified version. The model assumes that the individual behavior modules that are arranged in parallel are linked with each other on two levels. Before discussing this in detail, we will deal with the structure of an individual

module. The intensities of stimulus and motivations are linked additively. The sum is called internal behavior activation. If this is greater than a given threshold, the signal reaches a winner-take-all net which ensures that only one mode of behavior is executed (Fig. 16.**6a**). This is the lower level connection between the modules. The model incorporates ten modules whose concomitant behavioral response is given in Figure 16.**6b**. Modules 1 -7 denote consummatory behaviors, modules, A, B, C appetitive behaviors. The latter three have no motivation of their own, i. e., they respond exclusively to sensory input. The intensity of motivation is assumed to be constant for some modules (e. g., fear, readiness to fight). For other modules, motivation diminishes once a behavior is actually executed (dashed line in Figure 16.**6a**). In proportion to the intensity of the internal behavior activation of an module, the internal activations of other behavior modules can be raised or lowered. This is the upper level of cooperation. For this reason, linkages do not apply to all modules in the same way. Rather, the type of such a linkage depends on the type of modules, leading to the introduction of a hidden hierarchy. This is shown in Figure 16.**6b**. The rectangles represent the modules and the output corresponds to the internal behavior activations of the module. Some of them inhibit each other. For example, a higher internal activation of fighting behavior inhibits the internal activation of flight behavior. Some modules increase the internal activation of another module. These linkages are organized in such a way that they are capable of supporting temporal relations between two modules. For instance, if a specific behavior, such as "searching for food," is performed with greater intensity, the internal activation of the typically successive behavior, in this case feeding ("eat"), is also increased via a "successor" link (see solid arrows in Fig. 16.**6b**). At the same time, this raises the probability that, after the finding of food, the module "eat" prevails over the other modules in the winner-take-all net and its behavior is executed next. High internal activation, but in the absence of an activating stimulus, which indicates high motivation, activates the typical preceding behavior via "predecessor" links. The high internal activation of feeding thus activates the module "go to food," so that this behavior, which is appropriate in the circumstances, has a greater chance of being performed. In Figure 16.**6 b**, these linkages are represented by dashed arrows. Unless stimulated, internal activation decreases with a given time constant. In this way, this module is provided with low-pass properties, with the advantage that it does not respond to the high-frequency parts of the input signal as, for example, given by noise.

Though simple, the model contains a number of interesting analogies to the behavior of real animals. The behavior of the total system depends both on the internal state, the motivations, and the state of the exterior world. Depending on the internal state, it is thus capable of responding in different ways to the same stimulus situation. It can respond "opportunistically," displaying drinking behavior when water happens to be available, although hunger may actually be greater than thirst. A system driven purely by motivations ("goal driven") would just search for food in such conditions. The system may also show displacement behavior. When readiness to fight and fear are both high and are suppressed by mutual inhibition, a third behavior may win the decision of the winner-take-all net as described earlier (Fig. 16.**2**). Like animals, the system is endowed with the phenomenon of varying attention. This means that, if a strong motivation exists for one mode of behavior, other equally important modes of behavior may be suppressed. It is thus possible that, in case of very great hunger, the model might not allow the behavior "obstacle avoidance" to become operational, and that it runs up against the obstacle. Due to the stimulation influences of the successive behavior, the system is endowed with the advantageous property of anticipation. The running toward a conspecific individual already raises the readiness to fight, even before the activating stimulus is registered. Lorenz's view that motivation increases continuously, if not for all, then at least for some modes of behavior, could be simulated by the introduction of an integrator into the motivation channel (as used in Figure 16.**4b**). This would mean that the system is also capable of displaying a vacuum activity. It should also be mentioned that, rather than influencing just one mode of behavior as described here, a given motivation might possibly influence several modes of behavior.

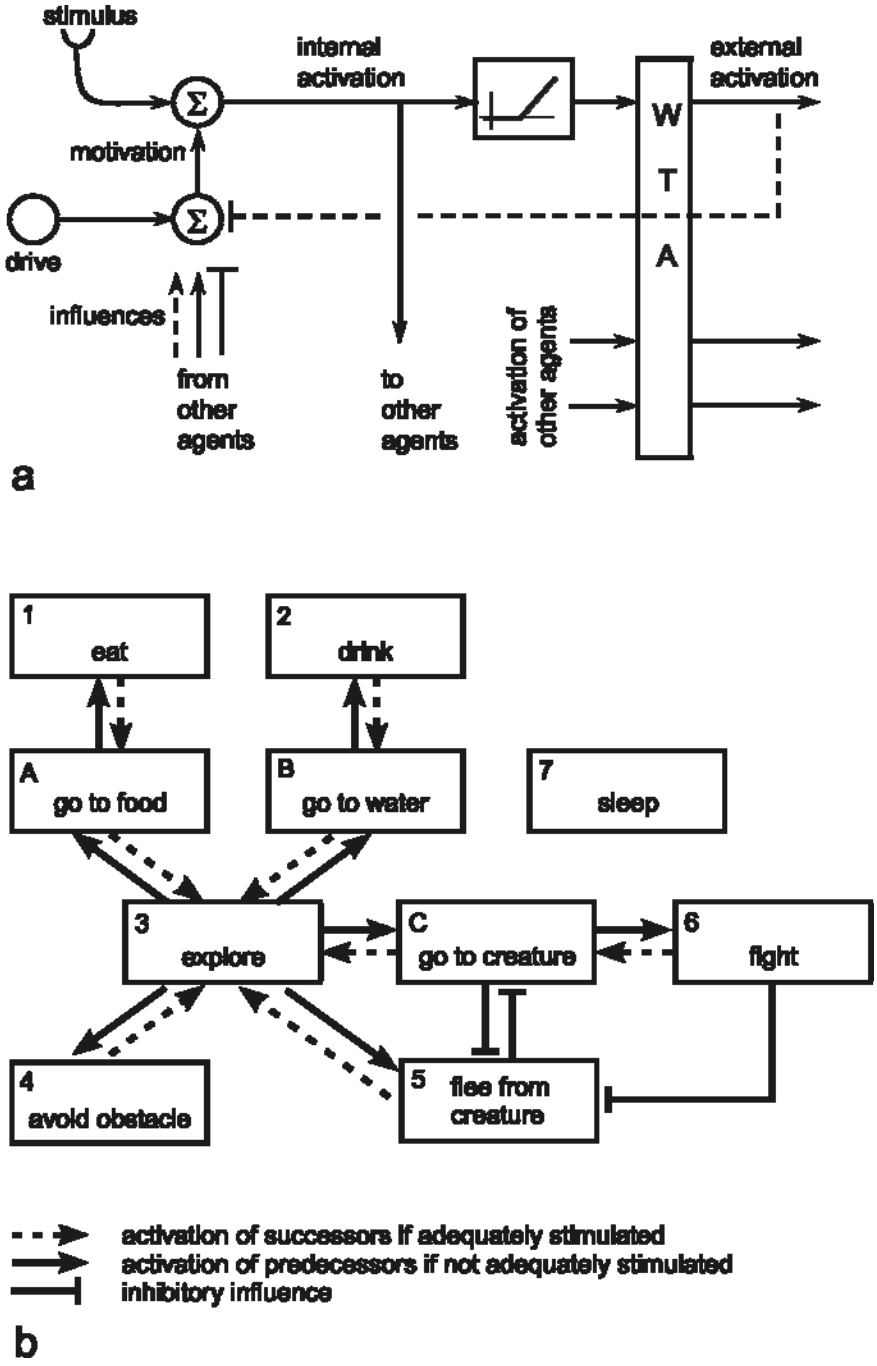


Fig. 16.**6 (a) A behavior module as used in the model of Maes.** The excitation produced by stimulus and motivation are added to form the „internal activation". If above threshold, this competes in a winner-take-all net (WTA) with the internal activations of other modules. If it succeeds, the external activation of this module elicits the behavior and inhibits the motivation. The motivation may also be influenced by an internal constant or varying drive (see Fig. 16.**4**) and by the internal activations of other modules. (b) The interior connections between the ten behavior modules used (the winner-take-all net is not shown). Solid arrows indicate an excitatory influence when the module sending the information receives an adequate stimulus. These arrows point to modules which usually follow the behavior of the sender (e. g., eat follows go-to-food). Broken arrows point to another module if the sender does not obtain an adequate stimulus. These arrows influence modules that are usually active before the sender is active (e. g. go-to-food is followed by eat) (after Maes 1991)

The problem with this "behavior-based" approach is that a considerable amount of intelligence is still necessary to design such a system. The question remains how animals solved this problem during evolution. How is it possible that a system can develop such structures without a specific designer? Tentative approaches have been made, for example, by Pfeifer and Verschure (1992) and Pfeifer (1995). With the aid of Hebb synapses, a certain flexibility can be achieved by formation of new linkages of stimulus inputs and responses in the sense of classical conditioning (Pavlovian conditioning). In this case, too, some structure has to be prespecified though. In the example mentioned, a small, wheeled robot was constructed which has two built-in, i. e., hardwired, reflexes (Fig. 16.**7a**). When a collision detector has registered an obstacle, the robot drives rearward for a short period and then turns. Two collision detectors are mounted such that one detects obstacles on the right side of the body, leading to a left turn, whereas the other detector mounted symmetrically on the other side, elicits a right turn. Furthermore, a target detector is stimulated by a certain target signal, for example, a light source. The built-in reflex elicits a turn to the direction of the target, if the sensors are stimulated accordingly. In this case, the whole 360° range is observed by four sensors. If none of these detectors is stimulated, the robot moves straight ahead. These reflexes can be implemented using simple feedforward networks which, for convenience, are not shown in detail in Figure 16.**7**, but are represented by the bold arrows which connect both the c-layer (consisting of two units, "avoid") and the t-layer (consisting of four units, "approach") with the m-layer. In case of a conflict between the two reflexes, "collision avoidance" and "target approach," an unidirectional inhibitory connection influence operates to suppress the latter. With this net, the robot can move in an obstacle-cluttered environment and, if a target is presented, find the target by trial and error.

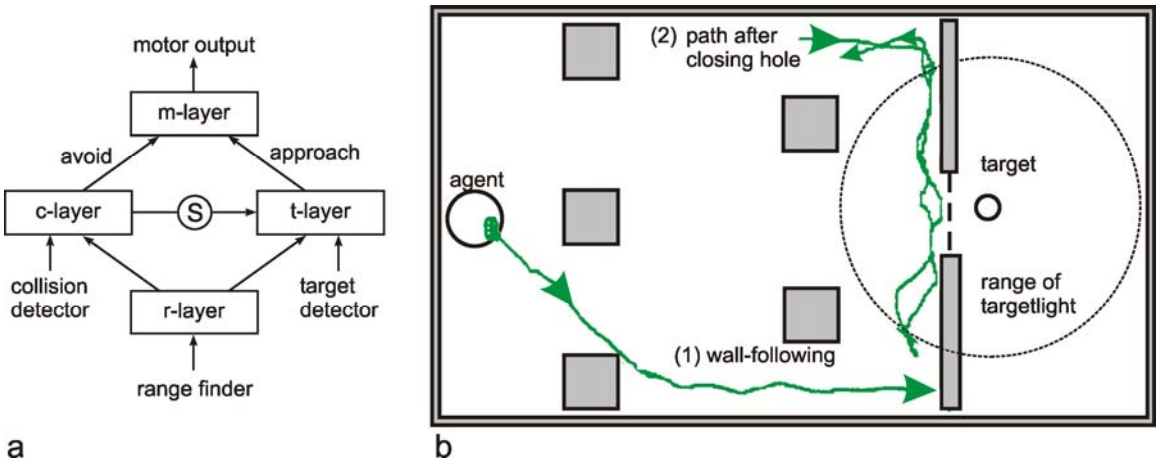


Fig. 16.**7 (a) The structure of a robot to learn a new reflex.** Two reflexes are built into the system: a connection between the collision detector and the motor output, producing an avoidance reflex, and a connection between target detector and motor output, producing a reaction to approach the target. In the case of a conflict the former reflex suppresses the latter (S). The connection between a third group of sensors, the range finder, and the other reflexes are learnt "automatically" using Hebbian learning. (b) shows some examples of how the robot behaves. It shows wall following (trace 1). Usually the robot finds the target after it has reached the range of target light source. If, in another experiment (trace 2), the hole in front of the target is closed, the robot "searches" in front of the closed door. As the collision avoidance net is stimulated quite often, the suppression of the approach net may become inhibited and the robot may move outside the range of the target light source (after Pfeifer and Verschure 1992, Pfeifer 1995)

Now a third, more complicated sensor is introduced. The sensor measures the distance of obstacles in a given angular range in front of the robot. The range sensor consists of a number of individual units (about 26, r-layer) each being responsible for a small angle range (3 degrees), covering a total range of about ± 40° around the forward direction. The output of these sensory units is now connected to all neuroids of the avoidance (c-layer) and approach subnets (t-layer).

The weights of these connections are zero at the beginning, but are changed according to Hebb's rule during the activity of the robot. Thus, when a collision detector is on, those units of the range sensor that are also stimulated will obtain stronger weights. Therefore, the weights develop such that, after learning, the range sensor signals achieve connections to the motor system which elicit similar behavior in similar situations as does the collision reflex. The same holds for the target approach reflex. Because the range sensor can detect objects from a greater distance than the collision detector, the number of collision decrease to zero during learning, and the robot shows much smoother trajectories.

In this way, different stimuli, which might occur at about the same time, can be learnt to replace each other, as is the case in classical conditioning. This is particularly interesting if, as in our example, the prewired reflex is a very simple one, and the more complicated connections have not to be hand-coded but are learnt automatically. Even better, it is learnt by using the perspective of the robot, and not that of an external designer, thereby taking the "situatedness" of the robot into account. The robot acquires its own knowledge, as it interacts with the environment. Because Hebb's rule is expanded by a decay term (see Chapter 12), the learning mode does not need to be switched off by an outside controller. Rather, the robot learns continuously.

The robot has developed a "wall following" behavior. (See trace 1 in Figure 16.**7b**). This continues even if the target is removed. An interesting behavior can be observed in another experiment. If the hole in the wall is closed, the robot moves along a trajectory similar to that shown in Figure 16.**7b**. If one describes the behavior of the robot as a psychologist would when observing a human subject, one might say the following: the animat follows the wall (trace 2 in Figure 16.**7b**) because it "expects" to find a target; it "knows" that the target is behind the wall, therefore, after "searching" along the wall for a given period or distance, it turns and tries again. However, the robot is not caught in this cycle. Rather, after some trials it "gives up," it "resigns." On the network level, the occurrence of this behavior can be explained in the following way: the "approach net" (Fig. 16.**7a**) is often inhibited. As the inhibitory connection has the property of a temporal low-pass filter, its output increases more and more, eventually inhibiting the approach net so strongly that the avoidance net completely governs the motor output for a prolonged time.

Of course, no "emotion units" for "expectation" or "resignation," or even special components for wall following are built into the system. The "psychological" interpretations are interesting, however, insofar as they show that, even if we have the strong impression of the existence of psychic states, they may not exist explicitly in the system in the form of special mechanisms. They might be just in the eye of the observer. This could even be the case for so-called higher systems. This question will be considered in Chapter 18.

The construction of animats or autonomous robots is a significant tool to understand and test hypotheses concerning the control of various types of behavior. Different architectures have been proposed. An important step beyond purely sensory driven systems is the introduction of such central aspects as motivation. Another important, but so far generally unsolved, problem concerns the question of how systems can by themselves develop new sensible behavior.

17 Development of Complexity in Animats

Although systems as outlined in the last chapter could serve to describe many phenomena found in animal behavior, the question remains whether all known modes of animal behavior are attributable to systems based on relatively simple rules, or whether other structures are required for higher, so-called intelligent processes, i. e., modes of behavior that do not rely on given rules, but on reflecting, or reasoning. This question will be approached by considering a hypothetical series of animats that show an increasing number of abilities making these animats more and more similar to those highly developed systems that we believe to be intelligent. This hypothetical series, shown below, will briefly repeat several examples described earlier. To some extent, this series might reflect the evolution of the capabilities of animal brains.

The most simple types of animats contain a simple feedforward net consisting of two or three layers (an associative memory such as shown in Figure 10.**9a**) representing a system with direct sensorimotor coupling. The system can react to a number of different stimuli with adequate responses. The "choice" between the different behavioral responses is completely sensory-driven or "reactive." As described above, such a system already shows the properties of generalization and fault tolerance. Coarse coding (see Chapter 10.1) is already an inherent property of discretely structured neuronal systems.

Such an animat could be improved on the sensory as well as on the motor side. Examples of special developments on the part of the input (sensory) layer could be the introduction of spatial high-pass filtering (lateral inhibition) to make the system independent of absolute stimulus intensity. Special systems may be introduced to allow for dimension reduction to simplify the multidimensional sensory input (see feature maps, Chapter 13). An often used principle in the animal kingdom, which is therefore also adopted by animat constructors, is to exploit the physical properties of the environment to obtain relevant information in a simple way. Examples are arrangement of polarization detectors in the ant eye that reflect the average arrangement of the celestial map, allowing the animal to detect the position of the sun with sufficient accuracy even when only a small spot of blue sky is visible (Wehner 1987). An exact mathematical solution for this problem is quite complicated. In motor systems potential energy can be stored during jumping by exploitation of the elasticity of muscles and tendons.

A further problem concerns phenomena of invariance. The size of a visual object on the retina, for example, depends highly on distance. Nevertheless, humans and many, though not all, animals treat such different appearances as the same object. Simple solutions providing size invariance in humans have been proposed by v. Holst (1957). The convergence angle between the two eyes and the degree of accommodation of the lenses are used by the brain to calculate the absolute size of an observed object. An invariance phenomenon can also be observed in color vision. Subjectively, we have the impression of the color of a given object being the same even when the actually reflected color changes dramatically, depending on varying influences from the environment.

Another problem on the sensory side is that of feature linking. Different aspects of a given perceptual unit (size and color of an object for example) may be recorded in different parts of the brain. In other words, different subsystems can exploit the same input in different ways. (See also the well-known ambiguous figures). The question is how the different aspects are combined in order to be exploited by further layers which control the actuators. Hypotheses of how this question might be solved have been put forward by Eckhorn, Singer, and von der Malsburg (e. g., Eckhorn et al. 1988, Engel et a1.1992, von der Malsburg and Buhmann 1992), but will not be discussed here in detail. This problem also involves the question of how information storage is organized in a neuronal system consisting of many modules, which will be considered below.

Further improvements discussed in the following require the existence of recurrent connections within the system. (Possibly, these are already necessary to solve the questions of feature linking mentioned above). It is clear that a strict separation between motor and sensory sides is no longer possible when recurrent connections are introduced. One may rather distinguish between "low-level" feedback belonging to the motor side and "high-level" feedback corresponding more to the sensory side. Simple improvements on the motor side imply the introduction of control systems with negative feedback (Chapter 8). Feedback can be used on several levels. Except for the classic form of continuous feedback, we find systems using intermittent feedback, i. e., the sensory signal is only used during defined time windows. Sensory input resulting from an action can, of course, also be used to trigger a new action (advance use), and thus as reference signal (Cruse, Dean, Heuer, Schmidt 1990). Whereas the sense of a simple feedback system is to maintain the output value near a desired value, the circuit of the reafference principle (Chapter 8.9) compares the input signal with a centrally produced "expected" value, ascertaining in this way whether a visually detected movement corresponds to that produced by its own motor system, thus, the reafference principle reflects some kind of "self-sense". Thereby the occurrence of external movements or a failure in the motor system could be detected.

An important capability of animats, and a necessity for evolved animals, is to cope with the problem of redundancy. This is due to both the sensory and the motor side. On the sensory side, most complex signals, e. g., in vision, can be interpreted in different ways, as mentioned below (e. g., figure-ground separation, ambiguous figures). On the motor side, a given goal can usually be achieved in different ways. On the lowest level, for instance, a observed object can be grasped by a manipulator, using the available actuators in different ways. For example, a human arm has more degrees of freedom than are necessary for most such tasks. This redundancy is even higher when the possibility of body locomotion is included. How does the system select between different possible behaviors, e. g., trajectories, and between different possible actuators, including their temporal order of activation? Simple solutions are known which imply fixed, preselected motor programs for given tasks, or fixed principles so as to minimize a given cost function (i. e., to find an energy optimal or time optimal solution). Recurrent networks are appropriate finding a unique solution from many possible ones when several limiting conditions have to be fulfilled. Therefore, recurrent nets may be used for this purpose (Chapter 11.3).

We have so far only considered systems which are purely sensory-driven. An important property probably of most animals is to contribute actively to the control of their behavior. A simple example is a central oscillator, as described in Chapter 14.1. Such central oscillators can be used to stabilize a behavior, making it independent of sensory influences, or as a kind of predictor system. The circadian clock of an animal, for example, is used to change internal parameters of the organism (e.g., body temperature) even before the external stimulus (e.g., light) occurs. More generally, animals and animats contain central pattern generators (CPG) that produce a special behavior (walking, flying). Simple examples are fixed action patterns where the activity of the CPG is only triggered by a sensory signal, but a CPG may also exploit sensory information during its activity. A simple example is given in Figure 14.**2**. In this case, the system consists of two fixed action patterns, the downstroke system and the upstroke system, but an oscillating system emerges when the environment is included as part of the system. A CPG may also use positive feedback, for instance to keep a behavioral state acting. Such central pattern generators may be used to control the input pattern in the sense of an active perception (see Box 8). By means of a central system only selected signals are further exploited by the system. The selection depends on the actual internal state of the system. This can already be observed in simple cases of "program dependent reflexes" where a stimulus is responded to in different ways depending on the internal state of the CPG. An example of this is the intermittent feedback mentioned above.

New problems arise when not one, but several CPGs are implemented in one system. These problems concern (a) structural arrangement and (b) how to select between these CPGs. On the structural side, one extreme case one could think of is a fully connected recurrent network, with the different functional units being more or less evenly distributed over the whole system, i. e., where all (or most of the) elements of the weight matrix show nonzero entries. Alternatively, one could think of separate subnets that may contain intensive recurrent connections within the subnet, but with only a small number of connections between different subnets. These subnets could be called modules, modules (Minsky 1985), or schemata (Arbib 1995). Because any number of intermodule connections could exist, it is not always possible clearly to define the modules. Apart from this problem, it should be noted that these modules might not only be arranged in parallel but also in some hierarchical order, and a module might therefore be regarded as consisting of several submodules.

Is there any advantage in modularity? If a system consists of subnets separated to some extent, the whole system is, intellectually speaking, easier to understand because these modules usually form smaller, and therefore more comprehensible, units. It may, then, be easier for an engineer, an animat and, for similar reasons, also evolution, to construct an animal from such modules. For example, if a new module is to be added to an already existing system, this is simpler if the modules are relatively separate. The old modules could be used nearly unchanged. For a completely interconnected system, however, most of the weights would have to be changed, if a new module is to be implemented (see expert nets, Chapter 12.6). On the other hand, modularity also has its problems. Apart from the problem already mentioned of how to select that module that will control the appropriate behavior, there are also structural problems. In the simplest case, these modules may be all arranged in parallel. Some hierarchy may be introduced according to the subsumption architecture (Chapter 16). Other, more implicit hierarchical structures can be used as indicated in the approach of Maes (Chapter 16). More explicit hierarchical arrangements may also be necessary. For example, one group of modules might form one CPG active in one situation, e. g., foraging, and another might exist for homing. Both groups, however, contain common submodules, e. g., to control leg movement.

The idea of a system consisting of modules, which are to some extent independent, also means that each module is active all the time (if not purposely switched off by another module), but only one or a few modules have control over the motor system (i.e., the actual behavior) at a given time. This requires selection between these modules. A simple way is to use a winner-take-all system (Figs. 11.**3**, 16.**2**), as is also applied in the Maes model. Other possibilities are given by hierarchies within the system (Brooks 1986, Maes 1991). As long as only one output actuator is concerned, a simple summation might be possible also (Steels 1994a). Selection of modules does not only refer to selection of actions, but also to selection of attention, i. e., to selection of special parts of the sensory layer. This leads to the active perception mentioned earlier. Mechanisms of selective attention are described in Figure 16.**5**, where variation of motivation leads to attributing more weight to some input signals than to others. The predecessor connections in the Maes model, for example, influence the system so that it, by changing the attention, even appears to have intentions. With such central circuits as a CPG, or the ability to select between different modules by means of influences of motivational elements and a winner-take-all net, the system is, within limits, independent of the actual sensory input. Such a system is much less directly sensory-driven than the strict feedforward systems mentioned earlier. In this sense, these systems can be called partially autonomous.

It should be mentioned here that the interaction of social animals, for example ants, termites, or bees, shows much similarity with interactions of modules within a brain. Here, too, we have partly independent units that provide each other with a limited amount of information on their own state and act together to achieve a higher goal without knowing it. The main difference is probably that, in social insects, the structure between the modules is simpler because many of the modules are of the same

type, and only a limited hierarchy (compared to the brain) can be found. But this simplicity makes social animals and social animats an excellent object for studying the possible results of cooperation between modules.

Up to now the animats and animals have been considered as having only an "innate" memory, implemented in the form of a defining of the connections and a fixing of the weights by the designer or by evolution, respectively. Of course, learning, i, e., the changing of weights and even the introduction of new connections, is an important capability. One could distinguish between adaptive behavioral changes - a change in behavior without a concomitant change in structure (e. g., weights) as, for example, achieved by the change of motivation described above - and actual learning, which implies a structural change. Different mechanisms of how learning could take place have been described earlier. Learning can occur on different levels. On the sensory side, learning of maps has been investigated in detail (Kohonen 1989). On the motor side, learning of fine adjustments of gain factors of actuators (to allow for drift, sensory differences, or damage in motors or muscles) has been shown to occur in biological systems (Möhl 1989, 1993), and may be transferred to technical solutions. A problem on the next higher level is the learning of conditioned reflexes, as shown above (Pfeiffer and Verschure 1992, Pfeifer 1995). The situation becomes more complicated, however, when animats consisting of different modules are considered. One can assume that each module has its private, local memory. Learning in a modular system is easier for the single module insofar as interference with information stored in other modules can be avoided, and as learning may be faster for smaller, as opposed to larger, subnets. However, there is also the problem of selecting the right module or, seen from the viewpoint of the module: How does the module decide when and what to learn? It is conceivable that learning is only "switched on" when the module is strongly excited by a motivational unit attached to it.

How might such a motivational unit be activated? One might assume that the modules do not only measure the physical ("objective") properties of the incoming signal, its mode (visual, acoustic ...), and its intensity, but also a third, "subjective" parameter, "pleasure" (Cabanac 1992), "utility" (McFarland and Bösser 1993), or the "reward" value in reinforcement learning systems (which evaluates each input signal by means of an innate (or learnt) evaluation scale). This requires a special "pleasure module" responsible for all incoming signals or distributed systems by which each module obtains its own pleasure signal or utility value. The selection between different activities could then occur, not by comparing the internal activation as in the Maes model (Fig. 16.**6**), but by comparing the utility values assigned to each activity (McFarland and Bösser 1993). Thus, the utility value might not only be used to select the behavior, but might also be used to trigger learning, as mentioned above.

An architecture using different modules might also require a mechanism for a system to invent new modules. However, in animals this is probably only possible in evolutionary time. Instead, new behaviors might be learnt not only by trial and error but by insight, i. e., by acquiring a new combination of already available modules. How to implement this capability in an animat is at present an open question, but some speculative considerations will be discussed below.

When the stored information is distributed over a number of modules, the question arises of how this distributed information is made available to the whole system. The simplest case discussed up to now is that every module uses only its local memory, which limits the overall capability of the system. A system might be much better if its modules (or some of them) could also use information stored elsewhere. This may correspond to the phenomenon of "insight" into a problem ("aha effect"); a coin is only stored so as to be used as money, but in a special situation one may suddenly have the idea of using it as a screwdriver. The phenomenon of insight raises the question of cognition. Cognition is often defined very broadly and includes aspects of introspection involving, for example, thinking, feeling, perceiving, and, as mentioned, insight. To obtain a definition that is not strictly tied to human

introspection, researchers have looked for definitions in terms of executable operations. As a prerequisite to explaining these, we need to make a distinction between procedural and declarative knowledge. Procedural knowledge is connected to a given procedure or a given action (for example, skilled movements such as piano playing or cycling). This type of knowledge can only be used by the system in the context of this procedure. Thus, it corresponds to the situation mentioned above when each module has its local memory which can only be used by this module, but not by others. Procedural knowledge is therefore also called "knowing how." By contrast, declarative knowledge is called "knowing what." In the human example, this type of knowledge can be represented by words. It is not easy nor helpful to explain in words how to ride a bicycle. Declarative knowledge concerns knowledge that is not tied to a particular process. It is usually defined as stored in "symbolic" form. However, what does it mean when a neuronal system stores symbols? An obvious case was explained in the example of Little Red Riding Hood (Fig. 10.**9b**). Individual neuroids can be regarded as "symbolizing" a given object. More complicated symbols might also be represented by groups of neuroids, the grouping of which may even dynamically depend on time (so-called cell assembly). However, as was already shown in the simple example of Fig. 10.**9a**, the same information can be stored in a fully distributed way.

At this point, it may be sensible to discuss briefly the meaning of information stored in symbolic form. The external observer cannot distinguish whether information is stored in a small number of units or in a broadly distributed way. The first system may, however, be more sensitive to destruction of individual units and require fewer connections. In the first system, the information is sometimes called encoded or explicitly represented, in the second system the encoding is implicit. However, this distinction may be only in the eye of the observer. For the system's functioning the information is available in the same way (apart from some technical differences, as mentioned). Thus, we want to look for a more appropriate and functional definition of these terms. We propose to call an encoding explicit, if an appropriate retrieval mechanism exists by which this particular information can be obtained. If the information is principally stored, but no such retrieval mechanism exists, we will call this implicit encoding. According to this definition, the information concerning the wolf is explicitly stored in both nets of Figure 10.**9**. If, in the above example, we present a naive human with a coin, the information that it can be used a screwdriver is only implicit so long as this human has not invented the appropriate extraction mechanism. Thus, the definition of implicit and explicit encoding relies critically on the retrieval mechanism. This agrees also with the provisional definition mentioned above as it includes the retrieval mechanism of the human observer; in Figure 10.**9b** we can directly "see" the encoding of the wolf, for example, but this is not so easily done in Figure 10.**9a**.

Therefore, the "symbol" seems to be in the eye of the observer, and the symbolic aspect seems not to be the critical property for stored knowledge to be "declarative." Rather, the important property seems to be that it can be recalled and manipulated by other modules not belonging to the process in the context of which this information was stored. According to the definition of McFarland and Bosser (1993), cognition is manipulation of declarative knowledge. We even tend to avoid the term "declarative" and use "manipulable" knowledge instead. Following the above arguments, this means that a system has cognitive abilities if it can deal with stored information independent of the context within which this information was acquired. This information does not need to be stored as a symbol in the sense of a morphologically simple structure, as, for example, a single neuron. However, such "simple" kind of storage might actually provide an easier recall of this information by other modules.

The important question for a cognitive system is how can this information stored in a module's local memory be retrieved in an economic way? An "intelligent" system will not work through all modules to find potentially interesting information. Rather, it should concentrate on promising regions of this distributed memory. The better it can do this, the higher is its intelligence. How to do this, i. e., how to

find related modules in a neuronal system, is an open question. In any case, one can assume that this search in the memory of other modules requires time. The procedure might correspond to "thinking," and is different to the application of procedural knowledge, which is less innovative but much faster. We mentioned earlier that an autonomous animat can act independently of direct sensory input. Correspondingly, during the "thinking" procedure, we have central activity within the system which is not directly coupled to the actuators, i. e., to motor output.

It is a moot question, whether it is important for the behavior of an animal, or an animat, that they contain modules representing an internal world model. A world model is an internal system which allows the animal or the animat to perform sensible actions because it "knows" some properties of the world. Of course, a world model does not mean that all properties of the world are represented. Every animat (and every module) already has a world model in the form of a weights connecting stimuli from the environment to appropriate reactions of the system. As long as these reactions are "appropriate," the weight distribution represents a, possibly simple, but sensible (implicit) world model. A simple central oscillator may, for example, be interpreted as a world model which is able to predict changes in the environment or to predict consequences of actions.

It is often postulated that brains of higher animals should have a world model, meaning that some neuronal structure represents properties of the world, and that these are used for planning actions in that model. In our definition this is an explicit world model. With the aid of such a model, modes of behavior and the corresponding results could be tested theoretically before actually being carried out. This means that cognitive mechanisms are necessary in order to extract the information. One may, therefore, also speak of cognitive maps. These ideas rest on several reasons. First, we know of the existence of a number of topological maps and these might be considered as a neuronal map that reflects geometric properties of the world. Second, we have the subjective impression of a continuous environment, although, on our retina, the image continuously changes. This could be explained by the assumption that we do not "look" at the retinal image, but at that of a "space model" within the brain. Hints of such continuous representation are even found in Drosophila, which follows an optical stimulus even if it is not visible for up to 3 s (Strauss and Pichler 1996).

Other arguments which are also proposed to support the assumption of such an explicit world model are that input from very different sensory organs, e. g., tactile and visual senses, have to be merged in the brain to form a complete representation of the object. However, we have already seen that a distributed representation is sufficient to control appropriate motor output even if the input comes from different sensors. Therefore, a number of authors suppose that it is sufficient for a system to have procedural knowledge and to be situated in the real physical world, and that no cognitive world models have to be introduced. Even the invention of human language is only a "trick" to invent new physical objects in the form of words which can then be treated as any other physical object (Bechtel and Abrahamsen 1990). These authors argue that it is easier (and more economical) to exploit the real physical world than to build explicit world models.

In any case, whether cognitive or noncognitive, higher developed systems possess world models which also contain information on the individual itself. This provides the basis for the fact that a human can react automatically to a stimulus, but can, at the same time, observe and record his or her reaction. This allows for two of the three levels of experience mentioned by Brentano: "hearing a tone," and "the subject which hears a tone." The third level, "my hearing of a tone," i. e., the introspective aspect, will be considered in Chapter 19.

This chapter reviews the important steps which illustrate the development of an animal or an animat to be able to behave autonomously in a realistic environment. These steps lead from feedforward, reactive structures to recurrent systems which may include central pattern generators and consist of more or less clearly separable modules or modules. Learning is important but how the appropriate module is selected for learning is an open question. Further, so far unsolved, problems address the question of application of symbolic information, of cognition and the need for internal world models.

Box 8

Active Perception

A fundamental problem is that the animat (or the animal) "lives" in a continuous stream of information ("situatedness"), and one important question is how to find discrete categories. This will be explained using an example, which is not yet fully understood, namely the question of figure-ground discrimination. How can the system decide which signals (e. g., a pixel in a visual image) belong to the

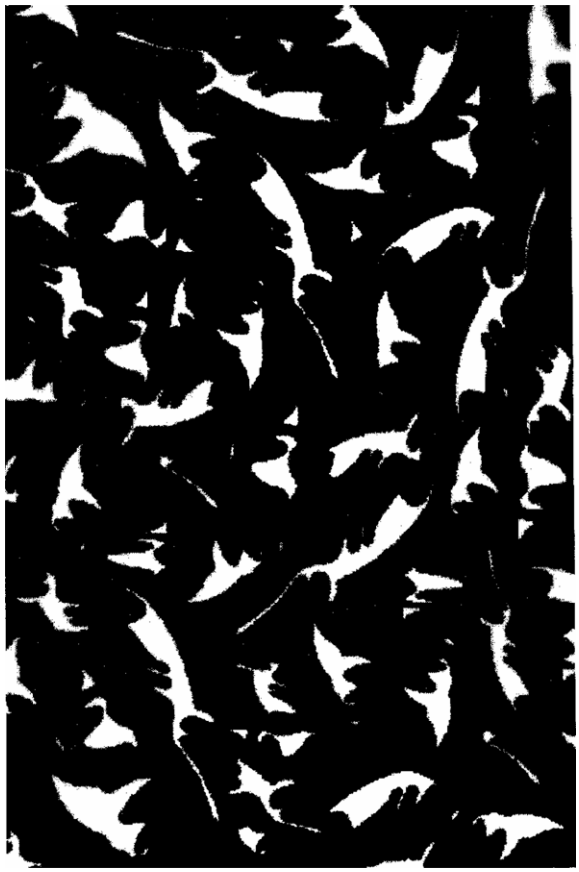


Fig. B 8.**1 The upper central part of the picture shows a human face.** The face can be recognized only with difficulty when looking at the picture for the first time (courtesy of H. Ritter)

figure and which to the ground. This is not a trivial problem as is made clear by for example regarding the well known ambiguous pictures. Part of the solution is probably that, in this case, we do not completely rely on a strict feedforward system, but that information from more central parts ("active perception") play a role. This can be illustrated by looking at Figure B8.**1**. On seeing this picture for the

first time, one may not immediately recognize the human face. However, if one has seen the face once, it is impossible to look at the picture subsequently and not to see the face. Therefore, central influences appear to be important for our perception. In other words, an item can be recognized because it is already expected. This problem does not only arise for signals distributed in space, but also for those distributed in time as, for example, in word recognition. This is often described as the cocktail party effect. Although at a party many different people may talk at the same time, concentration on one of the speakers permits separation of his or her phrases from the noisy background.

18 Emergent Properties

We mentioned above properties of systems that cannot directly be traced back on a particular mechanism or a particular structure. Such properties are sometimes called emergent properties. As these seem to be important abilities, the question of emergence should be discussed briefly. The term emergence is often used when a property of a system comes "out of the blue," i. e., if we cannot understand why or how the property occurs, although we have the impression of understanding the properties of the elements of the system. In order to apply the concept of emergence, a definition of the term "emergent property" might be useful. According to Steels (1994b), in an attempt to find an objective definition, a property is emergent if descriptive categories are needed to define it, which are not necessary to describe the behavior of the constituent components. A very simple example might be the occurrence of the resonance properties of a 2nd order system (Chapter 4.9). In this case, the term resonance frequency is not necessary (or useful) for describing the behavior of a 1st order electrical system consisting of an ohmic resistor and a capacitor, or a system consisting of an ohmic resistor and an inductivity. But the term resonance is very helpful to describe the behavior of a system which contains all three elements. There is no single module in the system that produces the effect of resonance. Rather, this is produced by the cooperation of these elements. Therefore, this property does not exist on the level of the elements but only on the level of the complete system. This means that there are phenomena where, to understand them, one might be inclined to look for a particular module responsible for a particular property, although no such module exists. It might be argued that this property exist only "in the eye of the observer," i. e., that it is produced by our brain, but this is an epistemological question which will not be treated here. Other examples, more sophisticated than the electric oscillator, are the rolling waves in the Benard experiment, or the states occurring in a gas laser (Haken 1973). In these cases, the occurrence of an emergent property is understood. A simple example in robotics might be the wall-following behavior of Pfeifer's robot (Fig. 16.**7**). A biological case is that some gait patterns in walking are not prespecified by a definite gait module, but "emerge" as a result of cooperation of leg controllers which cooperate via specific local rules (Cruse 1990). Thus, although a gait pattern like a gallop immediately takes our eyes, there might be no specific gallop module in the brain of such an animal. It is sometimes assumed that the property of intelligence, which some systems appear to have, is also an emergent property. Similarly, the impression of an human observer that an animal (or, even more obviously, an animat), appears to have emotional states may manifest emergent properties. In the later case, the epistemological question mentioned above is even more difficult to answer.

When discussing the question of representations, we earlier discussed the idea of storing symbolic information. However, as mentioned, such a symbol may also only exist as an emergent phenomenon, and the distinction between symbolic and nonsymbolic information may be irrelevant at the level of the constituent elements. (The individual bit in the computer does not know whether it belongs to a symbol or not). An animat may well be regarded a symbol user because it exploits entities of the physical

world (e.g., wall, corner) to which we would attach a symbol, although there is no corresponding symbolic representation in the "brain" of the animat.

In any case, these considerations should make clear that it is important to distinguish between the level of the observable behavior of an animal or an animat and the underlying mechanisms. There might even be more than two levels. This could be illustrated by slightly reformulating the classical question of Letvin and Maturana: what does the fly detector in the frog's retina tell the frog's brain? Does it say "there is a fly outside," or "start the catch reflex," or "there is an increased acetylcholine concentration at some synapses?" Of course, the sensors measure but do not interpret the world. However, the cooperation of the units of the whole system is such as to influence the human observer that he or she applies one of the "higher" interpretations.

Can we understand how the property of intelligence has evolved? There is an immense number of definitions of this term, and we do not intend to add another. It should, however, be clarified that there are two different ways of using this term, as an adverb and the nominal use. A system might be "intelligently designed," either by a constructor or by evolution, or it may itself have the capacity of intelligence. These two aspects might also be called species intelligence and individual intelligence. Sometimes, the first case is also called "hard-wired," although this term is misleading and will not be used here. Sometimes the term competence is proposed to distinguish species intelligence from nominal intelligence. Essential properties of a not just "competent," but "really" intelligent system might be the following. Such a system should sensibly react to novel situations, and find novel solutions to familiar situations. It should be capable of reasoning (i. e., finding a novel chain of inferences) and intelligent behavior should not be based on brute-force methods (i. e., exhaustive search) or on simple rules such as trial and error. In other words, cognition in the sense defined in Chapter 17 is a prerequisite for a system to be really intelligent.

Taking the discussion on emergent properties into account, one might assume that a module for intelligence does not necessarily exist, but that these properties may arise from the cooperation of simple elements. To approach this problem in more detail, the usually applied definitions, which tend to be subjective such as the Turing test (see Chapter 19), should be replaced by a more objective approach. In classic Artificial Intelligence a system was defined as intelligent, if it maximally applies the knowledge that it has (Newell: Steels 1994b). To apply this definition to an animal or an animat, first the "subjective environment" of the system must be defined, meaning the part which can be investigated by its sensors, and the part in which it is admissible for the system to use its actuators, because an action can only be judged intelligent (or not) for a given subjective environment. Furthermore, the cost function, which should be minimized, has to be defined. Taking into account the biological view of considering systems which have to survive, a system might be called intelligent if "it maximizes the chances of self-preservation in a particular environment even if the environment changes" (Steels 1994b). This definition could also be applied to animats and was actually invented for this purpose. It no longer distinguishes between intelligence by design and individual intelligence, but only concentrates on the ability of the system to survive. This means that no distinction is made in principle, as to whether the system has acquired its knowledge during evolution or during learning as an individual.

Thus, the evolutionary series described earlier illustrates a series of systems with an increasing capability to survive and therefore with an increasing intelligence. There is a continuum, and each observer may introduce a threshold above which the systems might be called "intelligent" depending on his or her personal intuition.

The evolved versions of animats described earlier already showed activities as acting partly independent of actual sensory input, keeping a "goal in mind" and following it, although the system has to deal with disturbances, and although even the direct sensory contact is missing at some time. Learning systems can develop new modules and use them later during their "life." All these properties might be subsumed under the term prerational intelligence (Cruse, Dean, Ritter, 1995 and Dean, Cruse, Ritter, 2000). Prerational intelligence is surely an emergent property, and it is an open question whether rational intelligence requires a special "rationality" module, or whether rationality, too, is an emergent property.

> A system property is called emergent if there is no special module that is responsible for the appearance of such a property. Symbolic information, and the phenomenon of intelligence, at least prerational intelligence, may be examples of emergent properties.

19 The Internal Aspect – Might Machines Have Feelings?

In the last two chapters, different properties of animats have been discussed. Sometimes, specifically at the end of Chapter 17, psychological terms have been used such as the system having emotions, expectations, and so on. These were, of course, as indicated by the inverted commas only used in a metaphorical sense for illustration, and not to mean that these animats actually have psychic states such as we experience by means of introspection, e. g., a sensation of pain, or being in an emotional state. The best one could say is that they behave as if they had such a state. However, this raises the question of whether such an artificial system could really have such states, and one could extend the question to phenomena such as our perception of the impression of consciousness or free will. An answer could be helpful in two aspects. If we could build such machines, it would not only help us to understand the important unsolved questions regarding what these phenomena are about but, with respect to application, if we assume that the occurrence of psychic states is coupled with higher mental capabilities, it might also help us to develop machines endowed with such capabilities.

As already described, machines can be built that give an observer the impression that they have internal states. Could an animat have something corresponding to a psychic state? Apart from asking more complicated questions, such as whether an animat could have consciousness or free will, we want to concentrate on the elementary, and more crucial question of whether, and under what conditions, an animat (or an animal) could have an internal impression (experience) at all, even of the simplest kind. In other words, is it possible that animals or animats experience the internal aspect as we do, or does the external aspect describe everything? For better clarification of the difference between internal and external aspects, we shall consider the phenomenon of pain. If a human subject touches a hot plate, he or she usually shows a fast withdrawal reflex and reports feeling pain. This is the external aspect of the observer (or third person's view). This observation could be described in more detail, if the neuronal activity of the subject could be investigated. But even if we assume that all neuronal details are known and a strict correlation (and a causal relation) between neuronal activities and the subject's report of feeling pain can be made, there is still a fundamental difference between this external aspect and the experience of "feeling pain." Even if the subject observes his or her own action potentials, this is not the pain felt. This internal aspect, based on the ability to feel, is also called the first person's view. Of course, a simple automaton could be constructed that shows a corresponding withdrawal reflex and expresses having felt pain: but we can be sure that there is no internal aspect in this case. That the latter is not necessary for the withdrawal reflex is already known for humans. The subjective impression of pain occurs some milliseconds after the motor reflex. When

thinking of the internal aspect, we are often using the metaphor of a homunculus who observes the brain activities. However, this is a misleading metaphor. It transforms the situation into a kind of daily experience of the outside world. However, the inner aspect is something completely different, and any outside-world metaphor would probably be wrong. How can we decide whether a system really has an internal aspect? Is it only a special property of highly developed systems such as primates or even just humans?

According to Descartes, an absolutely certain decision as to whether a sensation is experienced can only be made by the organism itself. Only the subject is capable of experience by introspection, using his or her internal view, of feelings such as pain or joy. Whether other subjects have the same, or corresponding, conscious experiences, can only concluded by analogy. "Based on my own experience in the past I judge that he behaves as I would behave if I had this or that particular conscious experience. Therefore, I assume that he has the same conscious experience." However, this is an indirect conclusion. One can never have direct access to the experience of another subject. If one had full knowledge of all neuronal activities in the other subject's brain, and one could understand the causal sequences of the events in that brain, one could probably predict the experiences which the subject reports to feel at a particular moment; but even if one had the most detailed knowledge, it would always reflect the "external aspect." The knowledge of the observer is never the same as the experience of the subject. In human beings, an observer can by means of empathy, of course develop a good imagination of what is felt by another human and thus, in this sense, have more or less the same knowledge. This is not what is meant here, though. The content of knowledge may be the same, but the ontological nature of knowledge is different. To illustrate this difference, two possible misinterpretations will be mentioned. It is conceivable that someone believes an animal or an animat to experience feelings although this is not justified for somebody who, in the case of a robot, knows the simple structure of the animat. Conversely, we may not be prepared to attribute to a fly the experience of an internal aspect, although it may possibly experience it.

In philosophy, this dualism between the external and the internal aspect has been discussed intensively. One question has been whether a causal relationship between both worlds exists. Some argue that this dualism does not really exist, but only seems real. They assume that, in time, we will be accustomed to these views and will consider both the same, nondifferent in the sense of the two sides of a coin (mind-brain identity). Others have said, and still say, "ignorabimus," i. e., we will never understand this problem. A way to overcome this dualistic view might be as follows: One might argue that all the entities we experience, be they feelings like fear, or "hard facts" like reading a figure from the scale of a voltmeter, basically belong to the internal world. Although usually only the former entities are described as belonging to the internal world, actually the latter do so as well. We experience the view of the figure too. Therefore, one might argue that both worlds are not distinct sets, but that the external, "objective" world is a subset of the internal, subjective world, and not at all its dualistic counterpart.

How could we decide whether an entity belongs to the "objective" subset? Two speculative possibilities will be proposed. All the entities of the subjective world might be ordered along a line describing whether, when being recognized, they elicit empathy or not. (There is no clear-cut borderline: a tree might be emotionally loaded when providing a shadow in the heat, and may be much less so in other situations). Only in the latter case is the entity considered as an objective fact. The entities of the subjective world might also be ordered along another line, which, however, may be correlated with the first. Some entities can more easily be distinguished from each other than others. Therefore, the former may be countable and can be more easily attributed to symbols. These are the ones we call objective whereas the other items are called subjective. Taken together, this view leads to the assumption that there is a quantitative, maybe a qualitative, but no ontological difference between both "worlds."

Nevertheless, the following can be stated: although the true epistemological borderline for realizing the existence of an internal view runs between the observer and all other organisms, including other humans, it is generally accepted that all human beings have an internal view. This can be supported by the argument of structural identity. As other humans have principally the same hardware, they function principally in the same way, and therefore also experience the internal aspect. With respect to animals, the less developed these are in phylogenetic terms, the less people are prepared to concede there is an internal aspect. However, the reason underlying this view is that, for example, a dog can much better provide us with information concerning its internal state than an insect can. This is, of course, not a conclusive argument. The structural argument does not help here either because we do not know whether any other structures, e. g., in insect brains, allow for the existence of the internal aspect.

Even if we had full knowledge of the state of a human or an animal brain, we would be able to parallel events in the outer world (the world of the observer) and the inner world only by means of correlation. Such a correlation becomes less possible, the more the observed brain differs from a human brain. Thus, the question of whether an insect "experiences," as opposed to being a simple "reflex machine," is (probably) impossible to answer. This, of course, becomes even more of a problem, if an artificial system is to be judged in this way. In some cases, a simple robot can give the impression of having experience but knowing its interior circuits, one would not be prepared to admit that it has experience. An attempt to solve this problem was Turing's test; an observer talks to a partner via a keybord and a monitor. The observer does not know whether the partner is a human subject or an artificial system. If the observer decides that the partner is human, although it is an artificial system, this system (according to Turing) may be called intelligent, which in our context, includes the property of having an internal aspect. But even if a complicated machine could be built that succeeded in all Turing tests, we would still not know whether it really has experience in the sense of the internal aspect or not. On the other hand, no dog would pass a Turing test although most dog owners would consider that their dogs can experience internal aspects.

Another approach to this problem is the biological view. The capability to have an experience in the sense of the first person's view might or might not affect the fitness of the animal, its capability to survive. If it touches a hot plate, the above mentioned withdrawal reflex could work, irregardless of whether the subject feels pain or not. Is there an evolutionary advantage to this experience, to the existence of the internal aspect? In what sense is a system fitter, if it has the ability to "feel?" Is the internal view an advantage or just an epiphenomenon?

To avoid possible misunderstanding, the following should be stated: One surely has to assume that in living systems, and also in artificial ones, states may exist that correspond to some motivational state (seen from the external view), and that therefore structures and mechanisms may exist which, for example, have the function of controlling attention shift (see Maes model). However, it is completely open whether the phenomenon of an internal aspect is necessarily connected to the appearance of such structures. There are two theoretical possibilities. (a) The existence of an internal aspect is necessary in the sense that a phylogenetic advantage is coupled with this phenomenon. This would mean that special structures have developed during evolution which are responsible for the occurrence of the internal aspect. This means that, in principle, they could be copied for an artificial system. This would be very interesting for any applications because it would enable us to build machines having these advantages whatever they are. (b) Alternatively, the existence of an internal aspect is a pure epiphenomenon in the sense that there is no advantage with respect to evolution. On this assumption, the conscious I would only be an observer. Two problems are involved in this possibility: if the conscious I is only a (passive) observer, it means that the impression of free will can only be a delusion. If epiphenomenon means that there is no effect which can be observed from

outside, it also means that this is no scientific hypothesis in the strict sense, because it cannot be tested.

Since, at this point, there seems to be no way of understanding the magic phenomenon of one's own internal aspect, we will, in the following, disregard it. Instead, we will concentrate on the external aspect only, i.e., we will ask under what conditions a subject may experience an internal aspect.

As mentioned earlier, one could imagine the brain as consisting of a number of modules which are responsible for specialized tasks. Many of these modules register inputs from the outer world and react to these signals. According to Minsky's ideas (Minsky 1985), there are modules working on a lower level which detect the details of the environment (e. g. form and color of an obstacle). There may also be modules which are specialized in representing particular aspects of the world, e. g., position of distant objects or of parts of the body. From the latter, an internal model of the world could be constructed. Some module (perhaps most of them) detect the states of other modules and react to their signals. Some of them might obtain input from these internal world representations. The task of these modules is to determine possible future actions on the basis of this information, for example controlling the movement of the parts of the subject's own body through an obstacle-cluttered environment.

Can these considerations also be applied to machines? How could such circuits work? In the past, it was even doubted whether a machine was capable of anything more than purely reflexive responses to sensory inputs, such as pursuing a goal. However, as shown in Chapter 8, a simple feedback system could do just this, holding a given value close to a goal (reference input) against various and unpredictable disturbances. As mentioned, animats already exist which have a simple explicit internal representation of the world and it is possible to build these animats with internal representations based on an modular architecture. Seen from the external aspect, those modules, which obtain input from the modules containing the world model, could be the locus where the whole system could "experience" the situation, in our example itself situated in a given environment. The system would therefore behave as if it really had an internal aspect. Whether it really does or not, cannot be answered. Nevertheless, we would like to speculate that the prerequisite for an "experiencing" system, i. e., a system with an internal aspect, is that modules exist which are able to view an inner representation of the world. One might further speculate that this is not only a necessary, but also a sufficient condition for a system to have an internal aspect.

Similarly, is it possible to think of a structure which shows the outer-aspect properties of a system having consciousness? The experience of feeling pain is a special and important event occurring in the consciousness. When we consider the phenomenon of consciousness generally, this is regarded as an experience to which several special properties can be attributed. Three important ones seem to mark this phenomenon. The first is the (i) "limitation" of consciousness: only very few items are conscious at a given time. It also seems (ii) that consciousness is occupied with the problem that is "most important" at the actual moment. This seems to help in enabling the whole system (iii) to concentrate on this important problem. Could these properties be implemented in an artificial system? These observations might be explained by the assumption that there is a module which detects problems, i. e., situations where two (or more) conflicting modules are highly excited. We can further assume that this problem-detecting module affects the system in such a way that the input to those conflicting modules remains high, so that any problem-solving module is constrained to deal with this problem. When another module, which surveys the entire situation, can only "see" the activities with the highest excitation, it "sees" only those types of information which seem to correspond to those which are in our consciousness; it refers to the most important problem of the moment; it contains only one problem, and the whole system concentrates on this problem as long as it is unsolved (and is still

the most important one). (See Cruse 1979 for a more detailed discussion). As it is generally not a problem to develop machines with internal models and modules which observe only the most important problem, one could assume that such a machine presents the external aspect of a system endowed with consciousness. However, as mentioned above, it is completely unclear, how one could test whether this machine really has an experience, or an internal view, other than by the very indirect tests like those proposed by Turing.

We believe that eventually machines will be built that - from the standpoint of the external view - will behave as if they had emotions, or motivations, and consciousness. Whether they will really have them in the sense of an internal aspect, strictly speaking, will (probably) never be known. Nevertheless, on the basis of an analog conclusion, we may ascribe to them an internal view; we do this now with respect to other humans, some of us to a dog, some even to a frog, and possibly a bee. Such a machine can take a position anywhere in this series, depending on its subjective similarity with the self-image of the observer.

One possible interpretation concerning the nature of the internal aspect is that there exist structures (the modules mentioned) which have an evolutionary advantage, but that the phenomenon of experience as such (e.g., of pain) can be considered an epiphenomenon which, by itself, has no particular advantage in an evolutionary sense. An internal view possibly appears whenever certain structural conditions are fulfilled. As mentioned above, the capability to experience an internal view may be based on the ability to view internal world models and may have developed quite early in evolution.

If the above mentioned speculations are correct, i. e., that a machine could have feelings and even consciousness, one may further ask whether a machine could also have free will. What does is mean for a human to have free will? We do not know what (free) will is, but we do experience it. If we have to make a decision, we do not have the impression that the result is already fixed beforehand, rather that we make it during the process, and that we are responsible for the result. This reflects the internal aspect. However, considering such a system from outside, the behavior seems to be determined by sensory input (plus internal systems including memory) and, possibly, random effects produced, for example, by chaotic properties. Insofar, there seems to be no basic difference to a machine. But even though one may accept this view, one still has the introspective impression of having a free will. Can a deterministic system have a free will and therefore be responsible for its actions? To some, this seems to be a contradiction, but reflects the different realities of the internal and the external aspect. This is similar to when we see the sun rise in the morning and move across the sky during the day, although we "know" that it is not the sun which moves. Similarly, we speak of psychosomatic effects, although many of us are sure that there is no literal influence of the "psychic" on the physical domain other than our ideas. We mix items of the external and the internal aspect, which is, strictly speaking, not permissible, but comfortable because it is complicated to always describe the "real" relationship. Thus, we will continue using the term free will not only because it corresponds to our internal experience, but also because it is a simple and important concept with respect to our whole cultural life. The loss of this concept could lead to misunderstandings which might produce a number of social problems. The concept that humans have a free will and are thus responsible for their actions and therefore the concept of moral depth, would lose its essential foundation. If this became accepted it would nevertheless be necessary to formulate social rules, such as laws and fines, as if free will existed, to avoid social disaster. Then, the mental modules would have to take into account the corresponding reactions of the world, and in this way, sensible social existence would still be possible. This view is actually already common, as it has long been in jurisdiction: fines represent a means of preventing crimes and not a means of atoning for moral debt. However, this line of argument is complicated and hence less comprehensible. Life would therefore not be easier, if this knowledge were to be accepted

by the human community. Another problem which might hamper the acceptance of these concepts is that they will be understood by many as questioning the idea of human self-understanding and of human dignity. We are, however, quite confident with respect to overcoming the latter problem because we easily survived three other attacks on these ideas, first, the Copernican revolution, second, the Darwinian revolution, and third, Freuds detection of the contribution of unconsciousness.

The elementary question discussed is whether animals or animals could have an internal experience as a human can experience this by observing him or herself. The difference between the internal and the external aspect can be made clear by considering the difference between feeling pain and looking at one's own neuronal activity correlated with this experience. Whether an insect (or an animal) "experiences" as opposed to being a simple "reflex machine" without an internal aspect, is (probably) impossible to answer. However, the argument of structural analogy, which can be used to accept the existence of an internal aspect in other humans except oneself, might, after further studies, in future also allow for conclusions with respect of other animals and possibly animats.

Appendix I – Laplace Transformation

As was shown in Chapter 8.3, difficulties may arise in the calculation of dynamic transfer properties, especially with feedback systems, since the convolution integrals cannot readily be solved. To overcome this problem, we use the Laplace transformation. This is a mathematical formula enabling a transformation of a time function f(t) into a function F(s):

$$L[f(t)]=F(s) \quad \text{and} \quad L^{-1}\,[F(s)] = f(t)$$

There are a number of simplifications for arithmetic operations with Laplace transformed functions F(s). Addition (and subtraction) are not affected by the transformation. But the latter offers the advantage that the convolution integral $y(t) = {}_0\int^t g(t - t') x(t')dt'$ becomes simplified into the product Y(s) = G(s) X(s). Y(s) and X(s) represent the Laplace transformations of y(t) and x(t), and G(s) that of g(t). G(s), the Laplace transformation of the weighting function g(t), is known as the transfer function. Transformations in both directions are achieved by means of tables. The most important transformations are given in the table below.

The following examples serve to illustrate how to use this table. We will start with the example of a simple feedback system (Fig. 8.**3**), whose filters F_1 and F_2 are described by the weighting functions $g_1(t)$ and $g_2(t)$. Since, after the Laplace transformation, only products occur instead of integrals, the equation system (1'), (2'), (3'), (4') described in Chapter 8.3.

$$h_1(t) = \int g_1(t-t')h_0(t')dt' \qquad (1')$$

$$h_2(t) = \int g_2(t-t')y(t')dt' \qquad (2')$$

$$h_0(t) = x_r(t) - h_2(t) \qquad (3')$$

$$y(t) = d(t) + h_1(t) \qquad (4')$$

is transformed to

$$H_1(s) = G_1(s)H_0(s) \qquad (1'')$$

$$H_2(s) = G_2(s)Y(s) \qquad (2'')$$

$$H_0(s) = X_r(s) - H_2(s) \qquad (3'')$$

$$Y(s) = D(s) + H_1(s) \qquad (4'')$$

Accordingly, the solution is the following equation:

$$Y(s) = \frac{G_1(s)}{1+G_1(s)G_2(s)} X_r(s) + \frac{1}{1+G_1(s)G_2(s)} D(s).$$

Thus, the transfer function of the closed loop (input: reference input x_r, output: controlled variable y) is as follows:

$$G(s) = \frac{G_1(s)}{1 + G_1(s)G_2(s)}.$$

This transfer function is formally obtained very easily by replacing the constants k_i in the formulas of Chapter 8.3 by the transfer functions G_i of the corresponding filters. When there is a series connection of several filters, the transfer function of the total system is obtained from the product of the transfer functions of individual filters. When making a parallel connection of two filters and adding their outputs, the transfer function of the total system is obtained by addition of individual transfer functions.

How do we obtain the corresponding weighting function from the transfer function of the closed loop? This will be demonstrated for a selected case (filter F_1 is a 1st order low-pass filter with the static amplification factor k_1 and the time constant τ, filter F_2 is a constant k_2). The weighting function of the low-pass filter is $g_1(t) = k_1\, e^{-t/\tau}$. The weighting function of a proportional term without delay, with the amplification factor k_2 is as follows: $g_2(t) = k_2$. According to the table, we obtain the two transfer functions $G_1(s)$ and $G_2(s)$ as $G_1(s) = k_1/(\tau\, s + 1)$ and $G_2(s) = k_2$. For the transfer function of the closed loop G(s), we then have

$$G(s) = \frac{k_1/(\tau s + 1)}{1 + k_2 k_1/(\tau s + 1)} = \frac{k_1}{\tau s + 1 + k_1 k_2}$$

This has to be reformulated in such a way that it takes a form which can be found in the table of the Laplace functions. This can be done as follows:

$$G(s) = \frac{k_1}{1 + k_1 k_2} \frac{1}{\tau s/(1 + k_1 k_2) + 1}$$

$$= \frac{k_1}{1 + k_1 k_2} \frac{1}{\tau_{cl} s + 1}$$

with

$$\tau_{cl} = \tau/(1 + k_1 k_2).$$

The corresponding weighting function can now be obtained by a transformation in the reserve direction by means of the table. A comparison of this transfer function with those given in the table immediately shows that this is the transfer function of a 1 st order low-pass filter with the time constant τ_{cl}, and the static amplification factor $k_1/(1 + k_1, k_2)$.

In the second example a system with positive feedback, a proportional element $G_1(s) = k_1$ and a high-pass filter in the feedback loop with $G_2(s) = k_2\,\tau s/(\tau s + 1)$ can be calculated as follows

$$G(s) = \frac{k_1}{1 + k_2 k_1 \tau s/(\tau s + 1)} = \frac{k_1 \tau s}{(1 - k_1 k_2)\tau s + 1} + \frac{k_1}{(1 - k_1 k_2)\tau s + 1}$$

$$= \frac{k_1}{1 - k_1 k_2} \frac{\tau_{cl} s}{\tau_{cl} s + 1} + \frac{k_1}{\tau_{cl} s + 1}$$

with

$$\tau_{cl} = \tau/(1 - k_1 k_2).$$

This shows that the system has the same properties as an parallel connection of a high-pass filter (first term) having a dynamic amplification factor of $k_1/(1 - k_1 k_2)$ (i. e. $k_1 k_2$ has to be smaller than 1) and a low-pass filter (second term). Both filters have the same time constant τ_{cl}. The output of these filters is added as shown in Figure 4.**8a**.

A detailed discussion of the preconditions to be fulfilled by a function to enable a Laplace transformation is not intended within the present framework. We are stating, though, that the transformation is permissible for all functions listed in the table. If, as in the first part of the table, general references to functions are made, such as y(t) or Y(s), all functions contained in the table may be used. See Abramowitz and Stegun (1965), for example, for more detailed tables, and DiStefano et al. (1967) for a detailed description of Laplace transformation.

	$x(t)$	$X(s)$
summation	$x_1(t) + x_2(t)$	$X_1(s) + X_2(s)$
constant value	a	a
constant factor	$ax(t)$	$aX(s)$
convolution	$\int_0^t g(t - t')x(t')dt'$	$G(s)X(s)$
integration	$\int_0^t x(t')dt'$	$\frac{X(s)}{s}$
dead time T	$x(t - T)$	$X(s)e^{-Ts}$
power function	t^{-n}	$\frac{\Gamma(1 - n)}{s^{-n}}$

Weighting functions:

1st order low-pass	$\frac{1}{\tau}e^{-t/\tau}$	$\frac{1}{\tau s+1}$
nth order low-pass	$\frac{1}{\tau(n-1)!}\left(\frac{t}{\tau}\right)^{n-1}e^{-t/\tau}$	$\frac{1}{(\tau s+1)^n}$
1st order high-pass	$\delta(t)-\frac{1}{\tau}e^{-t/\tau}$	$\frac{\tau s}{\tau s+1}$
2nd (g(t)) and nth (G(s) order high-pass	$\delta(t)+\left(\frac{t}{\tau^2}-\frac{2}{\tau}\right)e^{-t/\tau}$	$\frac{(\tau s)^n}{(\tau s+1)^n}$

1st order band-pass (1 = low-pass, 2 = high-pass filter)

$\tau_1 = \tau_2 = \tau$:	$\left(\frac{1}{\tau}-\frac{t}{\tau^2}\right)e^{-t/\tau}$	$\frac{\tau s}{(\tau s+1)^2}$
$\tau_1 \neq \tau_2$:	$\frac{1}{\tau_2^2-\tau_1\tau_2}\left(\tau_2 e^{-t/\tau 1}-\tau_1 e^{-t/\tau_2}\right)$	$\frac{\tau_2 s}{(\tau_1 s+1)(\tau_2 s+1)}$
lead-lag system	$k\delta(t)+\frac{1-k}{\tau}e^{-t/\tau}$	$\frac{k\tau s+1}{\tau s+1}$

2nd order oscillatory system $\omega_0 = 1/\tau$ and damping $\zeta = (1+\alpha^2)/2\alpha$:

$0<\zeta<1$:	$\frac{1}{\tau\sqrt{1-\zeta^2}}e^{-\zeta t/\tau}\sin\left(\frac{t}{\tau}\sqrt{1-\zeta^2}\right)$	$\frac{1}{\tau^2 s^2+2\zeta\tau s+1}$
$\zeta = 1$:	$\frac{t}{\tau^2}e^{-t/\tau}$	$\frac{1}{(\tau s+1)^2}$
$\zeta > 1$:	$\frac{\alpha}{\tau(\alpha^2-1)}(e^{-t/\alpha\tau}-e^{-\alpha t/\tau})$	$\frac{1}{(\alpha\tau s+1)(\tau s/\alpha+1)}$

Appendix II – Simulation of Dynamic Systems

Linear and nonlinear systems can be simulated in simple ways by means of digital programs. As a frame such a program requires a loop, with the loop variable t representing the time. By each iteration the time increases by one unit. Within the loop it is simple to simulate an integrator. This is done by the line $x_{out} = x_{out} + x_{in}$. As low-pass filters and high-pass filters can be constructed by feedback systems with integrators (Chapter 8.5), their simulation requires two or three program lines as shown in the example below. The output values can be subject to arbitrary continuous or non-continuous nonlinear characteristics.

By combining the output values in appropriate ways, arbitrary parallel and serial connections can be simulated. The following small C-program shows as an example the parallel connection of a low-pass and a high-pass filter. The summed output passes a rectifier, then a 2nd order oscillatory low-pass filter and finally a pure time delay. As input in this case a step function is chosen. Variables beginning with a y represent the output values of the system, variables starting with an aux represent the auxiliary variables. tau, omega, zeta represent time constant, eigenfrequency, and damping factor ζ of the filters, respectively, as described in Chapters 4.1, 4.3, and 4.9. The dead time is given by T.

```
main {

for (t = 1; t < 500; t++) {

/* input step function */
if (t < 20) input = 0; else input = 10.;

/* low-pass filter*/
aux_lpf = aux_lpf + input - y_lpf;
y_lpf = aux_lpf/tau_lpf;

/* high-pass filter*/
aux_hpf = aux_hpf + y_hpf;
y_hpf = input - aux_hpf/tau_hpf;

/* summation */
sum = y_hpf + y_lpf;

/* rectifier */
if (sum < 0.) sum = 0.;

/* 2nd order oscillatory low-pass filter */
input = sum;
aux0_osc = input1 + omega*aux3 osc + zeta* aux2_osc;
aux1_osc = aux1_osc + aux0_osc;
aux2_osc = - aux1_osc;
aux3_osc = aux3_osc + aux2_osc;
aux4_osc=- aux3_osc;
y_osc = aux4_osc;

/* pure time delay*/
input2 = y_osc;

y_ptd = buffer[T];
if (T > 0)
{for (i = T; i > 0; i--) buffer[i] = buffer [i - 1];}
buffer[0] = input2;

/* output */
output = y_ptd;}/* end of loop */

} /* end of main */
```

Appendix III – Exercises

Concepts are understood best, if the underlying ideas are not only verbally explained, but if, in addition, comprehension is supported by practical exercises. Whereas for the second part of the book, which deals with different types of neural networks, specifically designed software is provided, for the first part, i.e. Chapters 1 – 8, a general solution, the software package tkCybernetics is recommended. tkCybernetics, developed by Th. Roggendorf, allows on a graphical interface to easily and quickly construct different systems containing linear filters and nonlinear characteristics and the selection of different input functions.

The following exercises are designed to be performed with the simulation tool tkCybernetics.

A general advice before you start: Having a comfortable software package for the simulation of systems bears the following danger. One is inclined to first construct the system, then run it, look at the system's response, and only then try to understand what has happened. A much better way is to first construct the system, but ***before starting*** the input function, ***try to predict*** the probable result, for example by plotting a rough sketch of the expected response. Only after you have done this, start the input function and compare the result with your prediction. This is not as easy as the first approach, but helps dramatically to improve your understanding of the properties of dynamic systems. So, in parallel to your computer, always use paper and pencil when you are doing simulation exercises.

1) Investigate step response, ramp response, pulse response and frequency response (Bode plot) of:

a. Low pass filter (e.g. time constant = 30 units) (see Chapter 4.1)

b. High pass filter (see Chapter 4.2).

c. 2nd order low-pass filter (see Chapter 4.3).

d. 2nd order high-pass filter (see Chapter 4.4).

e. Integrator (see Chapter 4.8).

f. Serial combination of low-pass and high-pass filter (see Chapter 4.5). Try different time constants for both filters.

g. Parallel combination of low-pass and high-pass filter (see Chapter 4.6). Try different time constant for both filters and vary the gain in the branch of the high-pass filter, for instance.

2) Investigate the step responses of the circuits shown in Fig. 4.8.

3) Systems with feedback connections:

a. Investigate the step responses of a circuit as shown in Fig. 8.2, but without a disturbance input. Vary the values of the constants k_1 and k_2 between 0 and 5.

b. Replace one constant by a filter, e.g. a low-pass filter. Vary the values of the constants k_1 and k_2.

4) Repeat exercise 3) by applying a positive feedback connection instead of a negative one!

5) Negative feedback controller:

Test the six examples given in Fig. 8.8 by varying the values of the constants (note that some of these tasks have already been solved in the former exercises.) In particular challenge the stability of the system, by varying the constants (gain factors) and adding further low-pass filters of an order being high enough.

6) Nonlinear properties:

a. Apply a sine function to a concave or convex nonlinear characteristic (see Chapter 5).

b. Investigate the different serial connection of a linear filter and a nonlinear characteristic as shown in Figs. 6.2 and 6.3. Test step response and the response to a sinusoidal input.

c. Try a simulation of the system illustrated in Fig. 6.4.

7) Feedback systems with nonlinear properties.

a. Simulate the system illustrated in Fig. 8.16.

b. Simulate the system illustrated in Fig. 8.17.

References

Abramowitz, M., Stegun, I. (1965): Handbook of Mathematical Functions. Dover Publications, New York

Adam, D., Windhorst, U., Inbar, G.F. (1978): The effects of recurrent inhibition on the cross-correlated firing pattern of motoneurons (and their relation to signal transmission in the spinal cord-muscle channel). Biol. Cybernetics 29, 229-235

Amari, S. (1967): Theory of adaptive pattern classifiers. IEEE Trans., EC-16, No. 3, 299-307

Amari, S. (1968): Geometrical theory of information (in Japanese). Kyoritsu-Shuppan, Tokyo

Amari, S. (1977): Dynamics of pattern formation in lateral-inhibition type neural fields. Biol. Cybern. 27, 77-87

Amari, S., Arbib, M.A. (1982): Competition and Cooperation in Neural Nets. Springer Lecture Notes in Biomathematics Nr 45

Arbib, M.A. (1995): Schema theory. In: M.A. Arbib (ed.) The Handbook of Brain Theory and Neural Networks. MIT Press, Bradford Book, Cambridge, Mass, 1995

Baldi, P., Heiligenberg, J.M. (1988): How sensory maps could enhance resolution through ordered arrangements of broadly tuned receivers. Biol. Cybern. 59,313-318

Barto, A.G., Sutton, R.S., Brouwer, P.S. (1981): Associative search network: a reinforcement learning associative memory. Biol. Cybern. 40, 201 -211

Barto, A.G., Sutton, A.G., Watkins, C. (1990): Sequential decision problems and neural networks. In: M. Gabriel, J.W. Moore (eds.) Advances in neural information processing systems. Morgan Kaufmann 1990, 686-693

Bässler, U. (1983): Neural basis of elementary behavior in stick insects. Springer, Berlin, Heidelberg, New York

Bässler, U. (1993): The femur-tibia control system of stick insects - a model system for the study of the neural basis of joint control. Brain Res. Rev. 18, 207-226

Bechtel, W., Abrahamsen, A.A. (1990): Connectionism and the mind: an introduction to parallel processing in networks. Basil Blackwell, Oxford

Beckers, R., Deneubourg, J.L, Goss, S. (1992): Trails and U-turns in the selection of a path by the ant *Lasius niger*. J. of Theor. Biol. 159, 397-415

Bernstein, N. (1967): The co-ordination and regulation of movements. Pergamon Press Ltd.

Brooks, R.A. (1986): A robust layered control system for a mobile robot. J. Robotics and Automation 2, 14-23

Brooks, R.A. (1989): A robot that walks: emergent behaviours from a carefully evolved network. Neural Computation I, 253-262

Cabanac, M. (1992): Pleasure: the common currency. J. Theor. Biol. 155, 173-200

Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraulaz, G., Bonabeau, E. (2003) Self-organization in biological systems. Princeton Univ. Press, Princeton

Carpenter, G.A., Grossberg, S. (1987): A massively parallel architecture for a self-organizing neural pattern recognition machine. Computer Vision, Graphics, and Image Processing 37, 54-115

Chapman, K.M.. Smith, R.S. (1963): A linear transfer function underlying impulse frequency modulation in a cockroach mechanoreceptor. Nature 197, 699-700

Clynes, M. (1968): Biocybernetic principles and dynamic asymmetry: unidirectional rate sensitivity. In: H. Drischel (ed.) Biokybernetik. Vol. 1, Fischer, Jena 29-49

Cruse, H. (1979): Modellvorstellungen zu Bewußtseinsvorgängen. Naturw. Rdschau 32, 45-54

Cruse, H. (1981): Biologische Kybernetik. Verlag Chemie. Weinheim, Deerfield Beach, Basel

Cruse, H. (1990): What mechanisms coordinate leg movement in walking arthropods? Trends in Neurosciences 13, 1990, 15-21

Cruse, H. (2003): The evolution of cognition – a hypothesis. Cog. Science 27, 135-155

Cruse, H. (2002). Landmark-based navigation. Biol. Cybernetics 88, 425-437

Cruse, H., Bartling, Ch., Kindermann, Th. (1995): Highpass filtered positive feedback: decentralized control of cooperation. In: F. Moran, A. Moreno, J.J. Merelo, P. Chacon (eds.) Advances in Artificial Life. Springer, 668-678

Cruse, H., Dean, J., Heuer, H., Schmidt, R.A. (1990): Utilisation of sensory information for motor control. In: 0. Neumann, W. Prinz (eds.) Relationships between perception and action. Springer, Berlin, 43-79

Cruse, H., Dean, J., Ritter, H. (1995): Prärationale Intelligenz. Spektrum d. Wiss. 111-115

Cruse, H., Müller-Wilm, U., Dean, J. (1993): Artificial neural nets for controlling a 6-legged walking system. In: J.A. Meyer, H. Roitblat, S.Wilson (eds.) From animals to animats 2. 52-60. MIT Press, Cambridge MA

Cruse, H., Steinkühler, U. (1993): Solution of the direct and inverse kinematic problem by a unique algorithm using the mean of multiple computation method. Biol. Cybern. 69, 345-351

Daugman, J.G. (1980): Two-dimensional spectral analysis of cortical receptive field profiles. Vision Res. 20, 847, 856

Daugman, J.H. (1988): Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression. IEEE Transact. on acoustics, speech, and signal processing. 36, 1169-119

Dean, J. (1990): Coding proprioceptive information to control movement to a target: simulation with a simple neural network. Biol. Cybern. 63, 115-120

Dean, J., Cruse, H., and Ritter, H. (2000): Prerational Intelligence: Interdisciplinary perspectives on the behavior of natural and artificial systems. Kluwer Press, Dordrecht

Deneubourg, J.L, Goss, S. (1989): Collective patterns and decision making. Ethology, Ecology and Evolution 1, 295-311

Dijkstra, S., Denier van der Gon, J.J. (1973): An analog computer study of fast isolated movements. Kybernetik 12,102 -110

DiStefano, 111, Joseph J., Stubberud, Allen R., Williams, Ivan J. (1967): Theory and Problems of Feedback and Control Systems with Applications to the Engineering, Physical, and Life Sciences. McGraw Hill, New York, St. Louis, San Francisco, Toronto, Sydney

Eckhorn, R., Bauer, R., Jordan, W., Brosch, M., Kruse, W., Munk, M., Reitboeck, H.J. (1988): Coherent oscillations: a mechanism of feature linking in the visual cortex? Multiple electrode and correlation analysis in the cat. Biol. Cybernetics 60, 121 - 130

Eibl-Eibesfeldt, I. (1980): Jumping on the sociobiology bandwagon. The Behavioral and Brain Sciences 3, 631-636

Elman, J.L. (1990): Finding structure in time. Cognitive Science 14, 179-211

Engel, A.K., König, P., Kreiter. A.K., Schillen, TB., Singer, W. (1992): Temporal coding in the visual cortex: new vistas on integration in the nervous system. Trends in Neurosciences 15, 218-226

Exner, S. (1894): Entwurf einer physiologischen Erklärung der psychischen Erscheinungen. 1. Ted. Deuticke, Leipzig, 37-140

Fahlman, S.E., Lebiere, C. (1990): The cascade correlation learning architecture. In: D.S. Touretzky (ed.) Advances in neural information processing systems 2. Morgan Kaufman Pub. San Mateo, CA, 524-532

Fenner, F.J., Gibbs, E.P.J., Murphy, EA., Rott, R., Studdert, M.J., White, D.O. (1993): Veterinary Virology. Academic Press. San Diego, New York, Boston

Franceschini, N., Riehle, A., Le Nestour, A. (1989): Directionally selective motion detection by insect neurons. In: Stavenga, Hardie (eds.) Facets of Vision. Springer, Berlin, Heidelberg, 360-390

Fukushima, K. (1975): Cognitron: a self-organizing multilayered neural network. Biol. Cybern. 20, 121-136

Fukushima, K. (1980): Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cybern. 36,193-202

Haken, H. (ed.) (1973): Synergetik. Teubner, Stuttgart

Hartmann, G., Wehner, R. (1995): The ant's path integration system: a neural architecture. Biol. Cybern. 73, 483-497

Hassenstein, B. (1958 a): Über die Wahrnehmung der Bewegung von Figuren und unregelmäßigen Helligkeitsmustern am Rüsselkäfer *Chlorophanus viridis.* Zeitschrift für vergleichende Physiologie 40, 556-592

Hassenstein, B.(1958 b): Die Stärke von optokinetischen Reaktionen auf verschiedene Mustergeschwindigkeiten (bei *Chlorophanus viridis). Z.* Naturforschg. 12 b, 1-6

Hassenstein, B. (1959): Optokinetische Wirksamkeit bewegter periodischer Muster (nach Messungen am Rüsselkäfer *Chlorophanus viridis*). Z. Naturforschg. 14b, 659-674

Hassenstein, B. (1966): Kybernetik und biologische Forschung. Handbuch der Biologie, vol. 1/2, pp. 629-719. Akademische Verlagsgesellschaft Athenaion. Frankfurt/M.

Hassenstein, B. (1971): Information and control in the living organism. Chapman and Hall, London

Hassenstein, B., Reichardt, W. (1953): Der Schluß von den Reiz-Reaktions-Funktionen auf System-Strukturen. Z. Naturforsch. 86, 518-524

Hassenstein, B., Reichardt, W. (1956): Systemtheoretische Analyse der Zeit-, Reihenfolgen- und Vorzeichenauswertung bei der Bewegungsperzeption des Rüsselkäfers Chlorophanus. Z. Naturforschg. 11 b, 513-524

Hebb, D.O. (1949): The organization of behavior. Wiley, New York

Heisenberg, M., Wolf, R(1988): Reafferent control of optomotor yaw torque in Drosophilia melanogaster. J. Comp. Physiol. 163:373-388

Hertz, J., Krogh, A., Palmer, R.G. (1991): Introduction to the theory of neural computation. Addison-Wesley Pub., Redwood City

Hinton, G.E., McClelland, J.L., Rumelhart, D.E. (1986): Distributed representation. In: D.E. Rumelhart, J.L. McClelland (eds.) Parallel Distributed Processing, Vol. 1, MIT Press, Cambridge MA, 77-109

Holland, J.H. (1975): Adaptation in natural and artificial systems. Univ. of Michigan Press (2nd edition MIT Press, 1992)

Holst, E.v. (1950a): Die Arbeitsweise des Statolithenapparates bei Fischen. Zeitschr. Vergl. Physiol. 32, 60-120

Holst, E.v. (1950b): Die Tätigkeit des Statolithenapparats im Wirbeltierlabyrinth. Naturwissenschaften 37, 265-272

Holst, E.v. (1957): Aktive Leistungen der menschlichen Gesichtswahrnehmung. Studium Generale 10, 231-243

Holst, E.v., Mittelstaedt, H. (1950): Das Reafferenzprinzip: Wechselwirkungen zwischen Zentralnervensystem und Peripherie. Naturwissenschaften 37, 464-476

Hopfield, J.J. (1982): Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. 79, 2554-2558

Hopfield, J.J. (1984): Neurons with graded response have collective computational properties like those of two-state neurons. Proc. Natl. Acad. Sci. 81, 3088-3092

Hopfield, J.J., Tank, D.W. (1985): "Neural" computation of decisions in optimization problems. Biol. Cybern. 52, 141-152

Iles, J.F., Pearson, K.G. (1971): Coxal depressor muscles of the cockroach and the role of peripheral inhibition. J. Exp. Biol. 55,151 -164

Jacobs, R.A., Jordan, M.1., Nowlan, S.J., Hinton, G.E. (1991): Adaptive mixtures of local experts. Neural Computation 3, 79-87

Jaeger, H., Haas, H. (2004): Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. Science, April 2, 2004, 78-80. [Preprint including supplementary material: http://www.faculty.iu-bremen.de/hjaeger/pubs/ESNScience04.pdf]

Jordan, M.L (1986): Attractor dynamics and parallelism in a connectionist sequential machine. In: Proceedings of the eighth annual conference of the cognitive science society. (Amherst 1986) Hillsdale, Earlbaum, 531 - 546

Jordan, M.L (1990): Motor learning and the degrees of freedom problem. In: M. Jeannerod (ed.) Attention and Performance XIII. Hillsdale, NJ, Earlbaum, 796-836

Jordan, M.L, Jacobs, R.A. (1992): Hierarchies of adaptive experts. In: J. Moody, S. Hanson, & R. Lippmann, (eds.) Neural Information Systems, 4. Morgan Kaufmann, San Mateo, CA

Jones, W.P., Hoskins, J. (1987): Backpropagation: a generalized delta learning rule. Byte 155-162

Kandel, E.R.. Schwartz, J.H., Jessel, T.M. (2000): Principles of neural science. Elsevier, New York, Amsterdam, Oxford

Kawato, M., Gomi, H. (1992): The cerebellum and VOR/ OKR learning model. Trends in Neurosciences 15,455 - 453

Kindermann, Th., Cruse, H., Dautenhahn, K. (1996): A fast, three layered neural network for path finding. Network: Computation in Neural Systems 7, 423-436

Koch, U.T., Bässler, U., Brunner, M. (1989): Non-spiking neurons suppress fluctuations in small networks. Biol. Cybern. 62, 75-81

Kohonen, T. (1982): Self-organized formation of topologically correct feature maps. Biol. Cybern. 43, 59-69

Kohonen, T. (1988): An introduction to neural computing. Neural Networks 1, 3-16

Kohonen, T. (1989): Self-organization and associative memory. Springer Series in Information Sciences. Springer Verlag, 3rd edition

Koza. J.R. (1992): Genetic programming: A paradigm for genetically breeding computer population of computer programs to solve problems. MIT Press, Cambridge MA

Kühn, S., Beyn, W.-J., Cruse, H. (2007) Modeling memory function with recurrent neural networks consisting of Input Compensation units. I. Static Situations. Biol. Cybernetics 96, 455- 470

Kühn, S., Cruse, H. (2007) Modelling memory function with recurrent neural networks consisting of Input Compensation units. II. Dynamic Situations. Biol. Cybernetics 96, 471- 486

Le Cun, Y. (1985): Line procedure d'apprendissage pour reseau a seuil assymetrique. In: Cognitiva 85: A la frontiere de l'intelligence artificielle des sciences de la connaissance des neurosciences. Paris 1985, 599604. Paris CESTA

Levin, E. (1990): A recurrent network: limitations and training. Neural Networks 3, 641-650

Linder, C. (2005) Self-organization in a simple task of motor control based on spatial encoding. Adaptive Behavior 13, 189-209

Littmann, E., Ritter, H. (1993): Generalization abilities of cascade network architectures. In C.L. Giles, S.J. Hanson, J.D. Cowan (eds.) Advances in neural information processing systems 5. Morgan Kaufman Pub., San Mateo, CA, 188-195

Lorenz, K. (1950): The comparative method in studying innate behavior patterns. Symp. Soc. Exp. Biol. 221 - 268

Maass, W., Natschläger, T. and Markram. H. (2002): Real-time computing without stable states: A new framework for neural computation based on perturbations. Neural Computation, 14(11):2531-2560 http://www.lsm.tugraz.at/papers/lsm-nc-130.pdf

Maes, P. (1991): A bottom-up mechanism for behavior selection in an artificial creature. In: J.A. Meyer, S.W. Wilson (eds.) From animals to animats. Bradford Book, MIT Press, Cambridge Mass, London, 238-246

Makarov, V.A., Song, Y., Velarde, M.G., Hübner, D., Cruse, H. (2008): Elements for a general memory structure: Properties of recurrent neural networks used to form situation models. Biol. Cybern. 98, 371-395

von der Malsburg, Ch. (1973): Self-organization of oriented sensitive cells in the striate cortex. Kybernetik 14, 85-100

von der Malsburg, Ch. (1981): The correlation theory of brain function, Internal report 81-2, Göttingen, Germany: Max Planck Institut fur Biophysikalische Chemie. Reprinted in: Models of Neural Networks (K. Schulten, H.J. van Hemmen, eds.) Springer, 1994

von der Malsburg, Ch., Buhmann, J. (1992): Sensory segmentation with coupled neural oscillators. Biol. Cybern. 67, 233-242

Marmarelis, P.Z., Marmarelis, V.Z. (1978): Analysis of physiological systems: the white noise approach. Plenum Press, New York

Martinetz, T., Ritter, H., Schulten, K. (1990): Three-dimensional Neural Net for Learning Visuomotor-Coordination of a Robot Arm. IEEE-Transact. on Neural Networks 1,131-136

McFarland. D., Bösser, Th. (1993): Intelligent behavior in animals and robots. Bradford Book, MIT Press, Cambridge MA

Meinhardt, H. (1995): The algorithmic beauty of sea shells. Springer, Heidelberg

Milhorn, H.T. (1966): The application of control theory to physiological systems. Saunders, Philadelphia

Milsum, IT. (1966): Biological control systems analysis. McGraw Hill, New York

Minsky, M. (1985): The society of mind. Simon and Schuster, New York

Minsky, M.L., Papert, S.A. (1969): Perceptrons. MIT Press, Cambridge

Möhl, B. (1989): "Biological noise" and plasticity of sensorimotor pathways in the locust flight system. J. comp. Physiol. A 166, 75 -82

Möhl, B. (1993): The role on proprioception for motor learning in locust flight. J. Comp. Physiol. A 172, 325-332

Mohan, V., Morasso, P. (2006): A Forward / Inverse Motor Controller for Cognitive Robotics. ICANN (1) 2006, 602-611

Nauck D, Klawonn F, Borgelt C, Kruse R (2003) Neuronale Netze und Fuzzy Systeme. Braunschweig/ Wiesbaden: Vieweg-Verlag.

Oppelt, W. (1972): Kleines Handbuch technischer Regelvorgange. Verlag Chemie, Weinheim

Pfeifer, R. (1995): Cognition - perspectives from autonomous agents. Robotics and Autonomous Systems 15, 47-69

Pfeifer, R., and Verschure, P.F.M.J. (1992): Distributed adaptive control: a paradigm for designing autonomous agents. In: Towards a practice of autonomous systems: Proc. of the First European Conference on Artificial Life. MIT Press, Cambridge, Mass. 21-30

Pichler, J., Strauss, R. (1993): Altered spatio-temporal orientation in ellipsoid-body-open, a structural central complex mutant of Drosophila melanogaster. In: Elsner, N., Heisenberg, M. (eds.) Gene, Brain, Behavior. Proceedings of the 21st Göttingen Neurobiology Conference. p. 813. Stuttgart, Thieme

Ratliff, E (1965): Mach Bands: Quantitative Studies on Neural Networks in the Retina. Holden-Day, San Francisco, London, Amsterdam

Rechenberg, L (1973): Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Fromman-Holzboog. Stuttgart

Reichardt, W. (1957): Autokorrelationsauswertung als Funktionsprinzip des Zentralnervensystems. Z. Naturforsch. 12 b, 448 -457

Reichardt, W. (1973): Musterinduzierte Flugorientierung der Fliege Musca domestica. Naturwissenschaften 60, 122-138

Reichardt, W., MacGinitie, G. (1962): Zur Theorie der lateralen Inhibition. Kybernetik 1, 155-165

Riedmiller, M., Braun, H. (1993): A direct adaptive method for faster backpropagation learning: the RPROP Algorithm. Proceedings of the IEEE Int. Conf. on Neural Networks (IBNN), 586-591

Ritter, H., Kohonen, T. (1989): Self-Organizing Semantic Maps. Biol. Cybern. 61, 241-254

Ritter, H., Martinez, T., Schulten, K. (1989): Topology conserving maps for learning visuomotor coordination. Neural Networks 2, 159-168

Ritter, H., Martinetz, Th., Schulten, K. (1992): Neural Computation and Self-organizing Maps. Addison Wesley, 1 st revised english edition

Rizzolatti, G. (2005): The mirror neuron system and its function in humans. Anat. Embryol. 210(5–6) 419–421

Rosenblatt, E (1958): The perceptron: a probabilistic model for information storage and organization in the brain. Psychol. Rev. 65, 386-408

Rumelhart, D.E., Hinton, G.E., Williams, R.J. (1986): Learning internal representations by back-propagating errors. Nature 323, 533-536

Rumelhart, D.E., McClelland, J.L. (1986): Parallel Distributed Processing, Vol. 1, Bradford Book, MIT Press, Cambridge Mass, London

Schilling, M., Cruse, H. (2008): The evolution of cognition – from first order to second order embodiment. In: Modeling Communication with robots and virtual humans. I. Wachsmuth, G. Knoblich (eds.) Springer, Berlin, Lecture Notes in Artificial Intelligence, pp. 77-108

Schmidt, R. E. (1972): Grundriß der Neurophysiologie. Springer, Berlin, Heidelberg, New York

Schmidt, R. E. (1973): Grundriß der Sinnesphysiologie. Springer, Berlin, Heidelberg, New York

Schöner, G. (1991): Dynamic theory of action-perception patterns: the "moving room" paradigm. Biol. Cybern. 64, 455-462

Sejnowski, T. J., Tesauro, G. (1989): The Hebb rule for synaptic plasticity: algorithms and implementations. In: Byrne, J.H., Berry, W.O. (eds.) Neural models of plasticity: Experimental and theoretical approaches., Academic Press, pp. 94-103

Shiffrar, M. (2001): Movement and event perception. In: Goldstein, B. (ed.), The Blackwell Handbook of Perception. Blackwell Publishers, Oxford. pp. 237–272

Spekreijse, H., Oosting, H. (1970): Linearizing: a method for analysing and synthesizing nonlinear systems. Kybernetik 7, 22-31

Sperry, R. W. (1950): Neural basis of the spontaneous optokinetic response produced by visual inversion. J. Comp. Psychol. 43, 482-499

Steels, L. (1994a): Emergent functionality in robotic agents through on-line evolution. In: R. Brooks, P. Maes (eds.) Proceedings of the IV. MIT Press, Cambridge MA

Steels, L. (1994b): The artificial life roots of artificial intelligence. Artificial Life 1, 75-110

Steil, J.J. (1999) Input-Output Stability of Recurrent Neural Networks. Göttingen: Cuvillier Verlag.

Stein, R.B. (1974): Peripheral control of movement. Physiol. Rev. 54, 215-243

Steinbuch, K. (1961): Die Lernmatrix. Kybernetik 1, 36-45

Steinkühler, U., Cruse, H. (1998): A holistic model for an internal representation to control the movement of a manipulator with redundant degrees of freedom. Biol. Cybernetics 79, 457-466

Tani, J. (2003): Learning to generate articulated behaviour through the bottom-up and the top-down interaction process. Neural Networks 16, 11-23

Tani J, Nolfi S (1999) Learning to perceive the world articulated: an approach for hierarchical learning in sensory-motor systems. Neural Networks 12: 1131-1141.

Tani, J., Ito, M., Sugita, Y. (2004): Self-organization of distributedly represented multiple behaviour schemata in a mirror system: rebviews of robot experiments using RNNPB. Neural Networks 17, 1273-1289

Tank, D.W., Hopfield, J.J. (1987): Collective computation in neuron-like circuits. Scientific American 257, 104

Thorson, J. (1966): Small signal analysis of a visual reflex in the locust 11. Frequency dependence. Kybernetik 3, 53-66

Varju, D. (1962): Vergleich zweier Modelle für laterale Inhibition. Kybernetik 1, 200-208

Varju, D. (1965): On the theory of lateral Inhibition. Consiglio nazionale delle Ricerche, Rome, 1-26

Varju, D. (1977): Systemtheorie. Springer, Berlin, Heidelberg, New York

Varju, D. (1990): A note on the reafference principle. Biol. Cybern. 63, 315-323

Wagner, R. (1960): Über Regelung von Muskelkraft und Bewegungsgeschwindigkeit bei der Willkürbewegung. Z. Biol. 111, 449-478

Wehner, R. (1987): "Matched filters" - neural models of the external world. J. Comp. Physiol. A 161, 511-531

Werbos, P.J. (1974): Beyond regression: new tools for prediction and analysis in the behavioral sciences. Ph.D. Thesis, Harvard University

Williams, R.J. (1992): Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning 8, 229-256

Williams, R.J. and D. Zipser (1989): A learning algorithm for continually running fully recurrent neural networks. Neural Computation, 1, 270-280.

Willshaw, D.J., von der Malsburg, C. (1976): How patterned neural connections can be set up by self-organization. Proceedings of the Royal Society of London B 194, 431-445

Wilson, S.W. (1987): Classifier systems and the animat problem. Machine Learning 2,199 -228

Wolpert, D.M., Kawato, M. (1998): Multiple paired forward and inverse models for motor control. Neural Networks 11, 1317-1329