Logic and Digital Systems

Jaroslav Šípal

Author: Jaroslav Šípal
Title: Logic and Digital Systems
Published by: Czech Technical University in Prague
Compiled by: Faculty of Electrical Engineering
Contact address: Technicka 2, Prague 6, Czech Republic
Phone Number: +420 2 2435 2084
Print: (only electronic form)
Number of pages: 211
Edition: 1st Edition

ISBN 978-80-01-05315-7

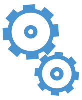
Reviewed by: Jaroslav Šípal

Innovative Methodology for Promising VET Areas
http://improvet.cvut.cz

Lifelong Learning Programme

This project has been funded with support from the European Commission.
This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Explanatory notes

	Definition
	Interesting
	Note
	Example
	Summary
+	Advantage
–	Disadvantage

ANNOTATION

Driving machines or machines can be assigned to one of three groups, which are named: controlling, regulation control and higher forms of control. Module describes essential information about controllers, signal management, filtering signals and fuzzy logic and neural nets. Each chapter is supplemented by examples and final questions.

OBJECTIVES

After studying this module, students obtain basic overview about digital/numerical control, computing signals, parameters and about two parts of artificial intelligence – fuzzy logic and neural networks. Modules also include basic information about digital filters and control algorithms like PID controllers.

LITERATURE

[1] BARTSCH Hans Matematické vzorce Mladá fronta Praha 2000; ISBN 80-204-0607-7

[2] Getting Started with MATLAB® 8; The MathWorks Inc. 2012

[3] HÄBERLE H. a kol. - Průmyslová elektronika a informační technologie; Europa Sobotáles Praha 2003; ISBN 80-86706-04-4

[4] REKTORYS Karel a kol. Přehled užité matematiky I+II vyd. Prometheus Praha: 2000; ISBN 80-7196-179-5

[5] SCHMID D. a kol. - Řízení a regulace pro strojírenství a mechatroniku; Europa Sobotáles Praha 2005; ISBN 80-86706-10-9

Index

1 Introduction and motivation

Modern man is today surrounded by many machines and devices that operate either on its instructions or are fully automated. Driving machines or machines can be assigned to one of three groups, which are named:

- control (so-called open - direct control without feedback),
- regulation control (closed feedback control),
- advanced control.

1.1 Control

Control is one of the simplest ways to control. This ensures startup of one or more devices. These devices can only be turned on or off, or perform a sequence of operations. Diagram of regulation control is shown in Figure 1.1

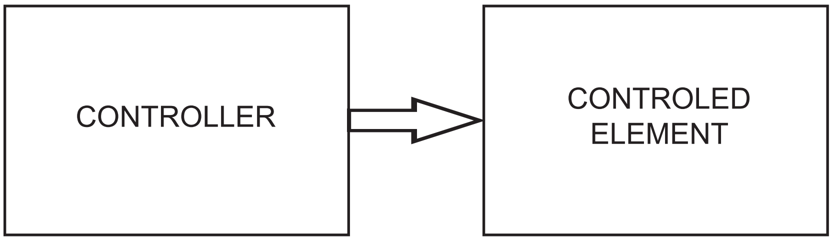


Fig. 1.1: Basic scheme for control

Controlled system is a machine which is controlled by the control system. For imagine is possible to show the following example. System control circular saw and actuator switch power. Power switch, an actuator, a circular saw, controlled system put into operation. Turn off the circuit breaker is made out of service.

Another example might be a set of belt conveyors that are logically connected. Activating this equipment, a gradual lowering of all conveyors. The first to put into operation the conveyor, which is in the direction of material flow last. When will be in normal operation (steady state) the previous conveyor is running. In this way, against the flow of conveyed material, there are all conveyors in operation. Shutdown may be gradual so that the first is off the conveyor, which was put into operation last. Other conveyors are being shut down at intervals. Length of the time interval between the impulse to turn off and the actual individual must ensure emptying conveyor belts from this material. Thus are gradually shut down all conveyors, gradually in the direction of material flow.

Control is possible according to type divided into several groups:

- Local
- Remote
- Automatic
- Software

Local control is used in simple machines and equipment. In particular, it is possible to see him in household appliances and power tools.

The remote control is very common in the manufacturing process, where there is more control of machines or equipment from a central control room operator or habitat. Very often it is a combination of local and remote machines using the switch. This method is applied especially in the partial repair of machinery.

Automatic control is such that when actuated member is featured / shut to / from the operation other than human impulses. As an example, the automatic pumping of water from the sump. Sensor measuring water level provides an impetus to start or stop the pump.

Program control is applied to the set of machines. For example, when you run big power units is necessary to provide lubrication and cooling machines triggered during service and sometime after weaning. Should this not happen, the machine would be damaged. So the command to start the machine are first put into operation the pump lubricating oil. After reaching the required oil pressure is put into the pump ensures coolant circulation. After meeting these two requirements, the main drive is put into operation. An example of a similar startup file device for household appliances can be automatic washing machine.

Control can show more detail, as shown in Figure 1.2 Controlled system as the pump that drives the electric motor (actuator). Connection to the power supply provides power switch, which is controlled locally or remotely (control).

Fig. 1.2: Detailed control scheme

1.2 Regulation control

Higher level for regulation is control (automatic control). This means that the system is controlled so that one or more physical variables are maintained at the prescribed parameters. An example of such a system may be gas furnace (controlled system), which performs preheating of material for reasons of a surface treatment (e.g., tempering). Material supplied inside the oven must be heated to a specified temperature and the temperature of the furnace, the regulatory circuits (control and regulation) to maintain a certain period of time (set point). This means that the temperature must be measured (measuring member) and its size is controlled by control valve (actuator) supply of fuel gas. Schematic representation of the process is shown in Figure 1.3.

The system is controlled by one or more of the parameters measured. These may be any physical quantity: temperature, pressure, speed, power, voltage, etc. Measuring unit processes the measured value to the appropriate signal and passes it to the controller. Since this is a transfer of information from the system, this branch is called feedback. Also entering into the controller set point (value). It is the size of the regulated parameter. The difference between value and reference signals feedback control deviation occurs. Control deviation signal enters the control block, the size of which creates the appropriate control input to the actuator. Their activities affect the actuator system and its parameters.

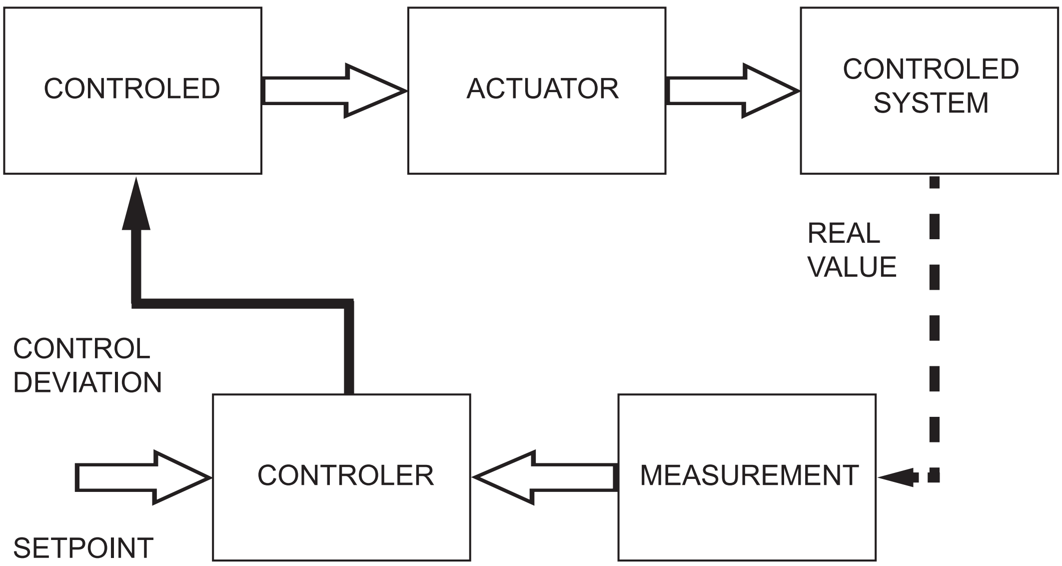


Fig. 1.3: Control regulation scheme

The above example can be applied to the schema. The technological process is needed certain temperature - Set point value. In the area of gas furnaces is measured temperature, which is lower than the set point. The controller evaluates the difference between the desired and the measured temperature and creates the appropriate error value. The control enters a block, it is interpreted so that it is necessary to increase the gas supply to the furnace. Size control deviation corresponds to the position of the actuator - the gas control valve. The valve release more gas into the furnace and burning to be released greater amounts of energy and thus a rise in temperature. The entire control process is repeated until the endpoint reaches the preset value.

1.3 Advanced control

In complex systems, the controller and the controller part of the control system. The system then uses a higher form of control. These higher forms of control include activities that support the resulting effect of the control process.

1.4 Repeating questions

1. Describe the methods of control.
2. What is the difference between the control and regulation?

2 Terminology

The theoretical basis for control is a math logic. This area is mainly represented by binary logic. Explained the logic functions, because the requirements for control of machines and their files are given verbally and is necessary for solving mathematical notation, which describes the technological requirements. Individual elements of math notation are technically feasible to implement. By simplifying the mathematical notation optimized number of physical elements needed to implement the control circuit.

2.1 System

The term system is necessary to understand certain set of elements, their relationships and properties schematic sketch is shown in Figure 2.1 Understand the definition of a complete set of "S", which consists of a set of elements and sets their mutual relations. The system "S" has a set of inputs and a set of faults. System "S" turn on the external environment has a set of outputs. If the system "S" has some prescribed behavior in him a set of desired variables and a set of regulatory differences.

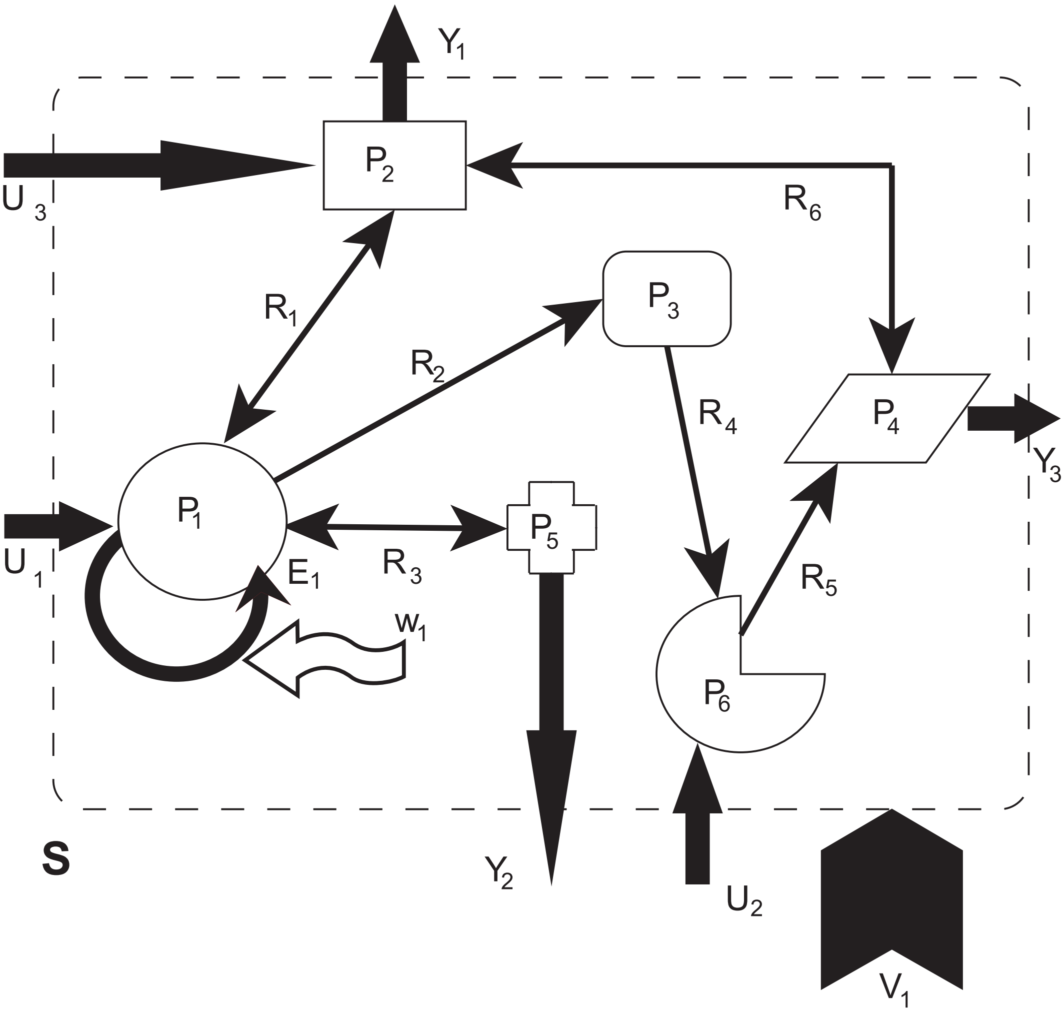


Fig. 2.1: System scheme

Image readers a general definition can be made on a technical example flying aircraft. The machine itself is composed of hundreds of parts, each with its own specific function. Furthermore, people in the aircraft, crew and passengers. Because the aircraft is moving in space and time, subject to external influences, such as temperature, wind direction and strength, etc. Other factors that can be described as failure can be sudden turbulence, lightning and aircraft equipment failure. Some of these defects may have a great influence on the main operation of the aircraft - the flight, such as engine failure. Others, such as failure toilets, strongly affect the mood among the passengers, but the main activity is not compromised. Between elements are created relationships, either on the basis of physical laws, social status, sympathy or antipathy. All these elements, states, relationships, values and external disturbances affect the outcome of a safe, happy years and achieving goals.

From the above example it is clear that the system may not only be a machine, but they can be distinguished systems of technical, physical, social, educational, mathematical, historical, social, and other livestock, as well as their combinations. This diversity systems shows that these systems are not easy to somehow split and described.

It may therefore be the mechanical systems, electronic systems, social systems, systems of botanical, zoological systems and of course, their combination. As an example, a technical system can be made car. If flying airliner described as a whole, can be seen as a technical-social system is a closed greenhouse botanical system, etc.

Systems can be classified according to the relationships between input and output variables to: static and dynamic, linear and non-linear, one-dimensional and multidimensional etc.

The numbers of input and output variables are divided systems:

- system with one input and one output signified SISO (single input, single output),
- system with several inputs and one output signified MISO (multi-input, single output),
- system with one input and several outputs signified SIMO (single input, multi-output),
- system with multiple inputs and multiple outputs signified MIMO (multi-input, multi-output).

Regarding the timing, and the sessions are divided systems:

- linear with time stable parameters - LTI (Linear Time Invariant System); example might be the relationship of voltage and current on the resistance,
- volatile linear with time parameters - LTNI (Non-Linear Time Invariant System); example might be the relationship of voltage and current on Photo-resistance,
- non-linear with time-stable parameters - NTI (Nonlinear Time Invariant System); example might be the relationship velocity discharge fluid through a hole at the bottom of an open tank,
- nonlinear unstable with time parameters - NTNI (Nonlinear Time-Invariant Non System); example might be the relationship between the number of births and marriages in society.

An example of a static system can be for example a voltage divider. The output voltage is defined by the size of resistors and input voltage. This relationship is described by the formula $u_2 = u_1 \times \frac{R_2}{(R_1 + R_2)}$. In this respect there is no time dependency and is described only dependent on the input voltage. In contrast, the electric current flowing through the photo resistor depends not only on the supply voltage, but also on the intensity of the incident light. This means that in the course of 24 hours as a session and the system is dynamic.

Linear systems are systems in which all members work with the linear relationship between the output of the input force such as a hydraulic press according to the movement of the control piston. Nonlinear system has at least one member with a non-linear dependence between input and output. An example of a non-linear relation is air resistance a moving car.

A system with one input and one output can be imagined as a refrigerator. The input value is a quantity that represents the loss of heat in the cooling space. The output variable is the actual temperature in the refrigerator compartment. In contrast, flying plane has several inputs: wind speed and direction, outside air temperature, load weight, etc. The outputs are: the angle of horizontal and vertical flaps, throttle, etc.

In Table 1 are used symbols set, their values are then labeled with a lowercase letter.

Table 1: Used symbols

S = {P;R;U;Y;V}	P – elements set	R – relations set	U – inputs set
Y – outputs set	W – set points set	E – control deviations set	V – errors set

Variables of the system are: input (action) variable "*u* (*t*)" output variable "*y* (*t*)" state variables "*x* (*t*)", the error "*e* (*t*)" fault variable "*v* (*t*) "and the set point value" *w* (*t*) ".

2.2 Continuity and discontinuity of physical quantities

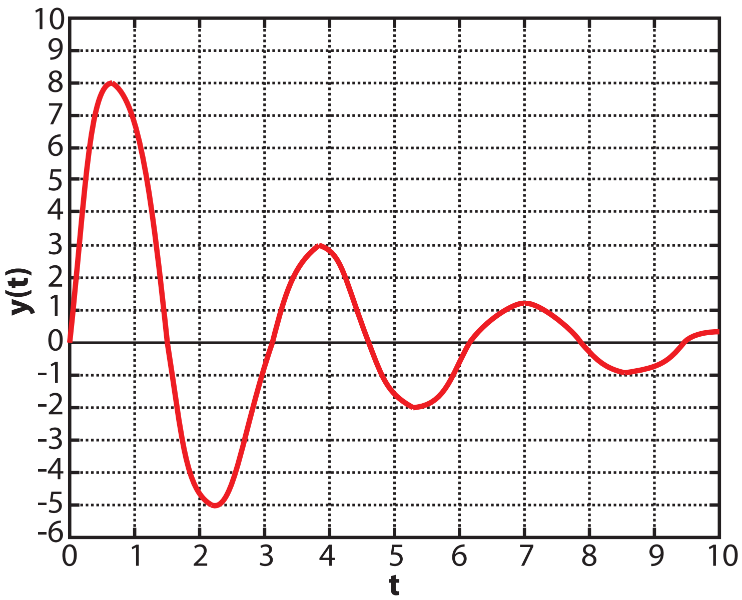
Physical phenomena in nature and run around in the great majority continuous manner. Values of physical quantities we describe the status of the action. The same applies to the controlled system (machine or group of machines). Information about the values of monitored variables as well as control commands is called signals.

2.3 Signals

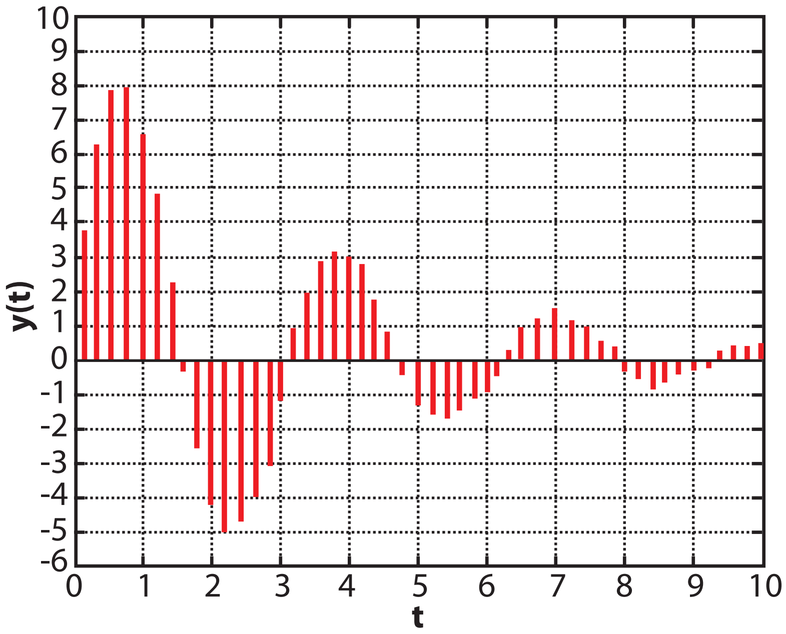
In control theory are important information. Bearer of information is the signal sources are transmitters, receivers provide reception. The signals are divided according to the relationship to self or dependent variable. Independent variable is time and the dependent variable is the amplitude. Distribution of signals is as follows:

- Signals are continuous in time (at each point in time is known amplitude value):
  - Analog signals, the example in Figure 2.3
  - Quantized signals, the example in Figure 2.3
- Signals discontinuous in time (between samples is not known amplitude value):
  - Sampled (discrete) signals, the example in Figure 2.3
  - Amplitude and time-discrete (digital) signals, the example in Figure 2.3

a)

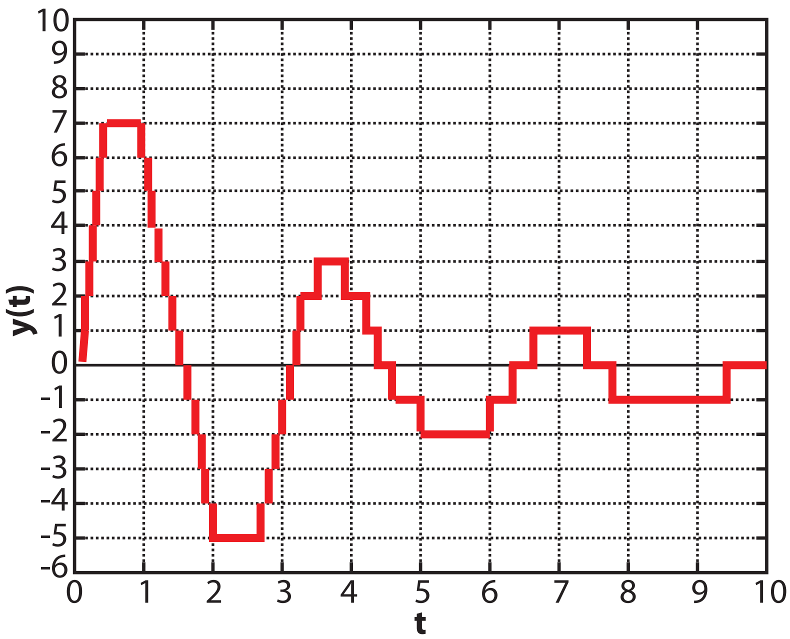


b)



10
9
8
7
6
5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
y(t)
0
1
2
3
4
5
6
7
8
9
10
t

c)



10
9
8
7
6
5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
y(t)
0
1
2
3
4
5
6
7
8
9
10
t

d)

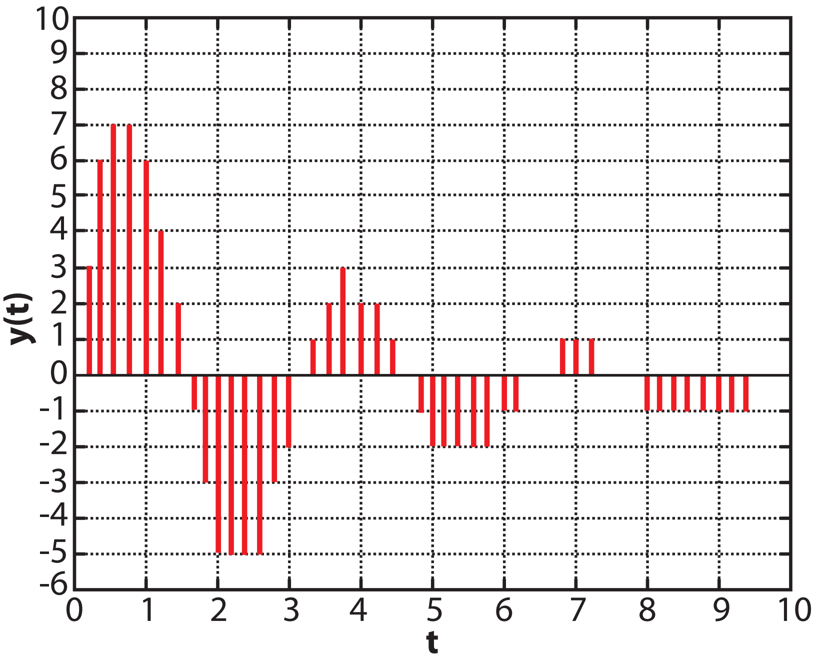


Fig. 2.3: Analog, quantized, sampled, amplitude nonsmooth signal

2.4 Used terms

In the area involved in the control of machines are used shortcuts that characterize the machine control. Abbreviations used are taken from the English terminology, these include:

- CNC
- PLC
- PAC
- HMI
- DCS

Numerical control - CNC (Computer Numerical Control)

Development of industrial production has brought with it the need for increased production of machine tools and forming machines. Machine control that was classic manual or cam was replaced by digital control. The first numerically controlled machines appeared in the 40th and 50 20th century. Servo control instructions for these machines were on punched tape (NC machines). With the development of computer technology, the machines were equipped with this information technology (CNC machines).

In modern CNC systems are highly automated component design using CAD programs (Computer-Aided Design) and CAM (Computer-Aided Manufacturing). These programs create a computer file that lists the sequence of commands needed to operate a particular machine, which can be equipped with various machine tools - machining centers.

PLC (Programmable Logical Controller)

Programmable logic controller (PLC) or programmable controller is a digital computer, which is used to control the operation of machinery and production processes. PLCs are used in many industrial sectors. Unlike general purpose computers, the PLC is designed for processing information from multiple inputs, is more resistant to temperature changes, to electrical interference and vibrations and shocks. Programs for the control of the device are typically stored in battery-backed or non-volatile memory. PLC operates in real-time system, because the output commands must respond to changing input conditions.

PAC (Programmable Automation Controller)

PLCs are gradually supplemented by new features that are no longer simple functions of logical type. They are complemented not only by regulatory roles, but a whole range of tasks in which there are also very demanding numerical algorithms and algorithms of artificial intelligence. For their designation is sometimes abbreviated PAC (Programmable Automation Controller), which can be translated as "programmable automation system."

HMI (Human Machine Interface)

Meaning this acronym describes the user interface. This is a place where there is an interaction between people and machines. The aim of the interaction between man and machine in the user interface is effective operation and control of the machine, and feedback from the machine. This is the information about the state of the machine or its components and the values of measured quantities ongoing technological process. This information helps the operator in taking operational decisions in the control of machines and processes.

The user interface is a system that includes hardware (physical) and software (logical) components. There are different systems of user interfaces from simple mechanical to computer systems. Currently, industrial process control use SCADA visualization programs (supervisory control and data acquisition), enabling the data to be displayed with animations.

DCS

These systems (Distributed Control System) is a large process control systems (PCS Process Control Systems - sometimes abbreviations are used as synonyms), which began to be used in the 60 years with the advent of the first control computers, which represented the numerical solution of the centralized control of large technological systems such as chemical processes, power plants, etc. They were built as suitable centralized solution. The 70 years there have been the first control minicomputers, which, while allowing partial decentralization. These solutions were used throughout the 70 summer and the beginning of the 80 years. These systems are generally divided into DCS: for power, for the nuclear program, for other technological processes, control systems for buildings.

Some DCS systems are specialized, some are, however applicable in more areas. Exceptions are control systems, where extremely high demands on safety and reliability of the control system. Highly secure and reliable systems are very expensive and therefore not deployed where it is not absolutely necessary.

DCS system is characterized by strict hierarchical construction with three levels of control that is bottom-up:

- sensors level - sensors, actuators,
- the level of the first control (technology control and regulation),
- operator level,
- Superior level.

In any case, however, still represent a large DCS control system with only a high degree of reliability in areas where it is necessary to treat a large number of inputs and outputs of various types and where reliability and security is absolutely categorical requirement. Their advantage is also the compactness of the system.

2.5 Repeating questions

1. Explain what is meant by the system and give an example.
2. Explain what is signal and how the signals section.
3. Explain the meaning of abbreviations: CNC, PLC, PAC, HMI, DCS.

3 Implementation of logic and digital systems

Binary systems can be implemented in different technologies, which are applied to logic functors. E.g. in control schemes, using transistors and even housing.

- Fixed logic
- FPGA
- program uC
- DSP
- PC
- Control schemes

Electrical control schemes are drawings in which is shown the control and power circuit. Elements of the main circuit are controlled by switches, power contacts of contactors or relays. The control circuit is composed of controls such as buttons, switches, timers and limit switches, coils, relays or contactors. Both elements can be replaced by weak-and power electronics.

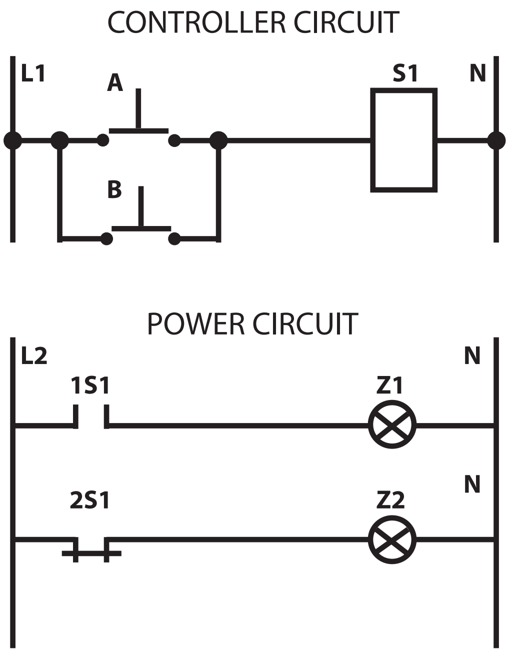


Fig. 3.1: Example for control scheme

The figure is drawn diagram of the control and power circuits with simple functions. Press A control voltage is applied to the contactor coil S1 is closed. Its normally open contact connected to the 1S1 voltage appliance (lamp) Z1, while NC 2S1 Z2 appliance disconnected from the power supply. The function of the circuit is such that pressing the A button will light lamp bulb goes Z1 and Z2. Release the button and the result will be the opposite.

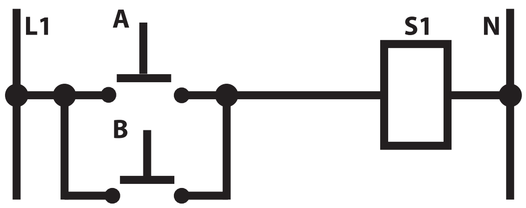
Independent variable, and realized two logical functions, equivalence and negation.

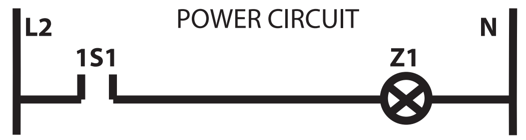
3.1 Implementation of basic logic functions of different technologies

Logical conjunction – AND

In the following examples, it is possible to study the technical implementation of the conjunction of two independent variables. The dependent variable takes value true only if at the same time both the independent variables are true. The control circuit is the solution to serial connection buttons. Implementation of transistors is solved using serial connection of transistors. The mechanical form is needed to operate both the mechanical linkage (a, b), in order to overcome the spring force.

CONTROLLER CIRCUIT





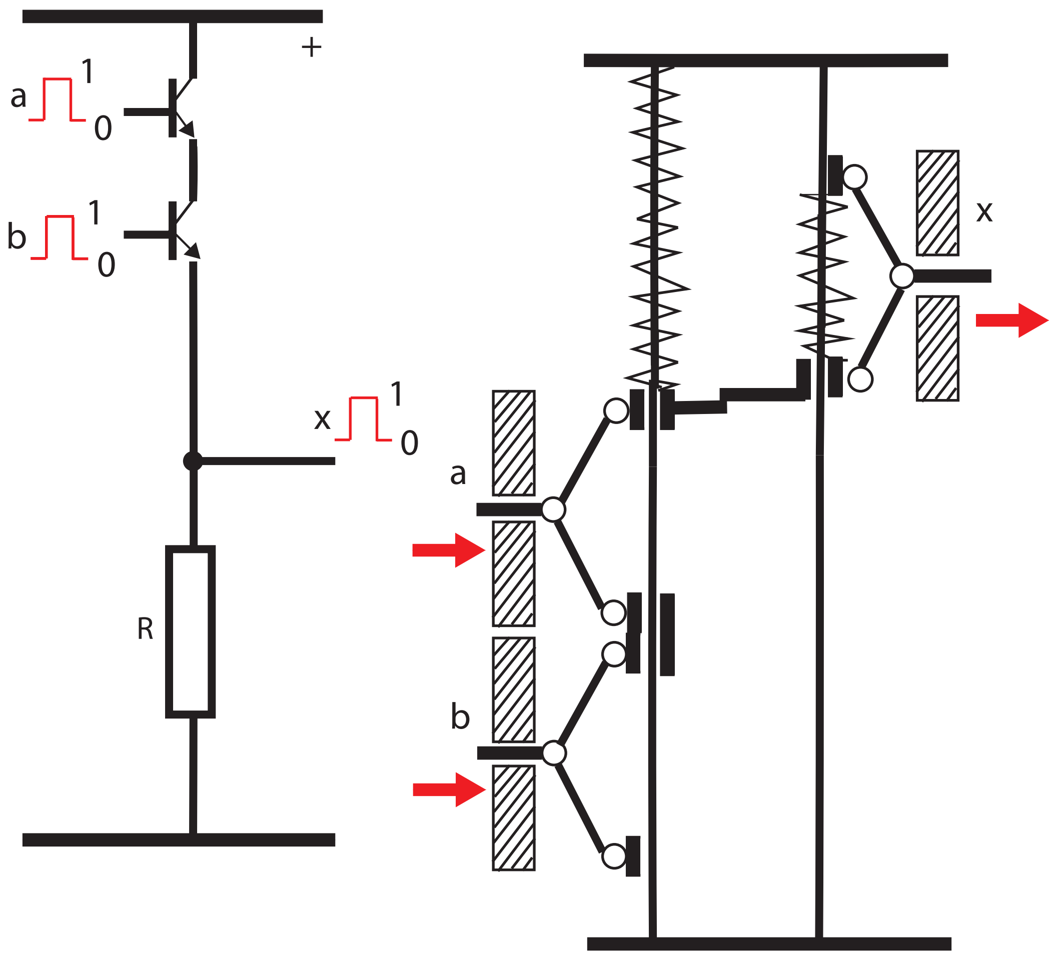


Fig. 3.2: Logical conjunction function – AND

Logical disjunction – OR

In the following examples, it is possible to study the technical implementation of the disjunction of two independent variables. The dependent variable takes a value of true if at least one independent variable is true. The control circuit is connected in parallel solution keys. Implementation using transistors is also dealt with parallel connection of transistors. The mechanical work is needed to form at least one of the mechanical linkages (a, b).

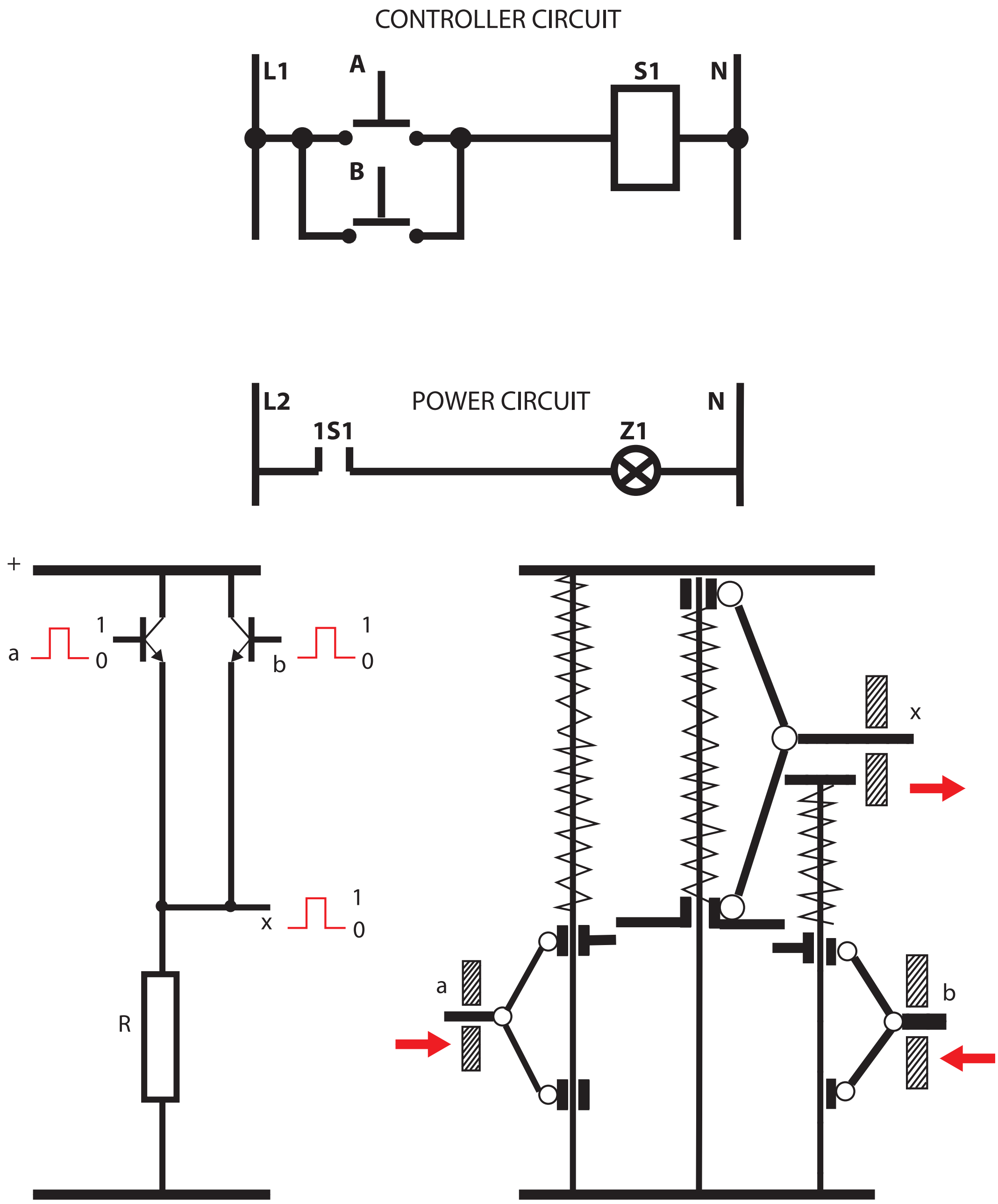


Fig. 3.3: Logical disjunction function – OR

Logical negation – NOT

In the following examples, it is possible to study the technical implementation of the negation of one independent variable. The dependent variable takes a value of true if the independent variable is false. The control circuit is the solution involved opening contacts button. Implementation is solved using a shorting transistor voltage transistor. The mechanical action of the form of mechanical linkage (a) is performed opposite reaction rod x.

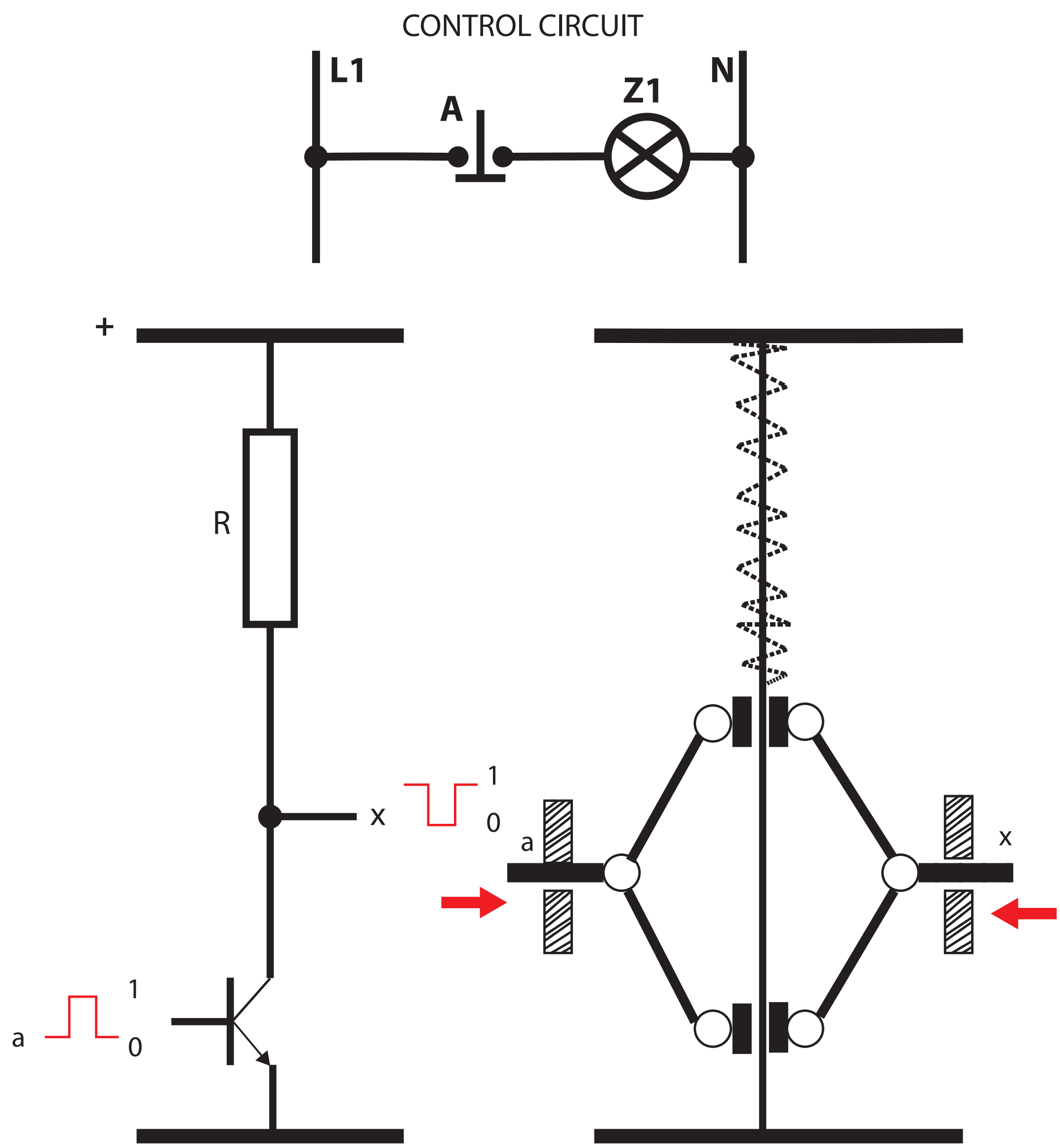


Fig. 3.4: Logical negation function – NOT

3.2 Schematic symbols for logic circuits

Given that the logical operations can be implemented using different technologies are generated logic diagrams. In logic circuit diagrams schematic symbols are used. Figure 3.5 Circuit symbol to represent logical sum, which has two inputs (left) and one output. Figure 3.6 is the schematic symbol for the logical product, which has two inputs and one output. Wheel in Figure 3.7 indicates negation. The diagrams are not drawn separately, but always in input or output tag for the product or sum.

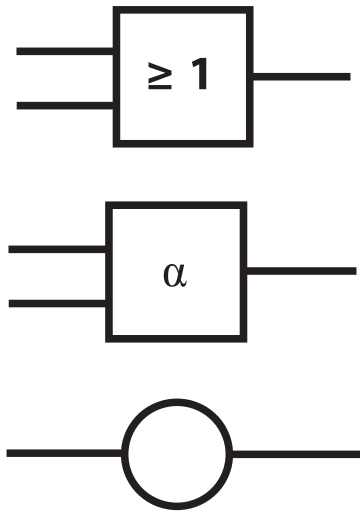


Fig. 3.5, 3.6, 3.7: Addition, product, negation

Note: In some, particularly older schemes may be signs of another shape, not a square or a rectangle. Mark for negation is the same.

Example 3.1

Draw the logic diagram of the circuit, which is expressed by the relation: $Y=\overline{A\times C+\overline{A}\times\overline{D}+B}$

Solution

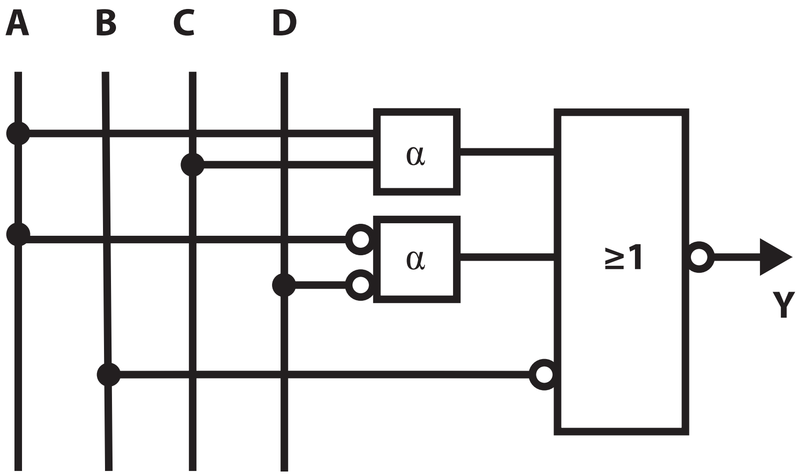


Fig. 3.8: Example 3.1

Example 3.2

Draw the logic diagram of the circuit, which is expressed by the relation:

$$Y = A \times B \times C + \overline{\overline{B} \times C} + D$$

Solution

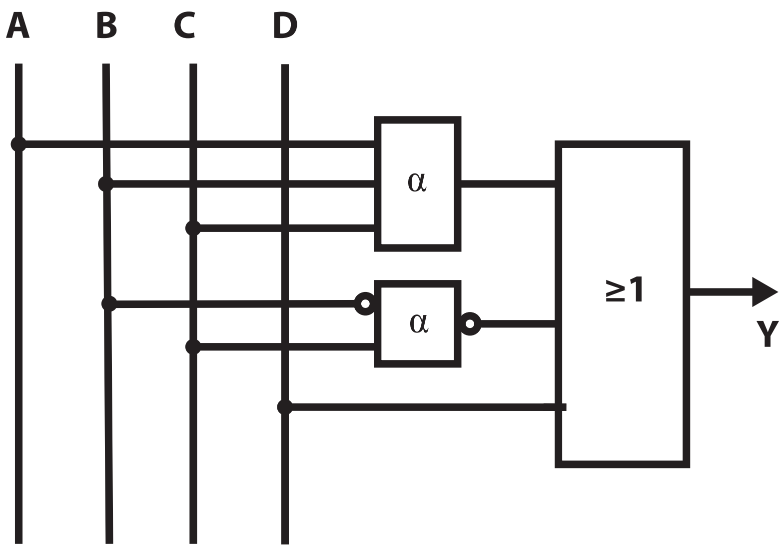


Fig. 3.9: Example 3.2

3.3 Editing logical expressions

From the preceding examples it is clear that each logic operation is represented by a single block in the logical schema that represents the relevant technology. The more logical blocks, the more components and the greater the probability of the failure of the control system. Reducing the number of used logic blocks is achieved by modifying and simplifying the logic functions. The simplification is performed on the basis of Boolean algebra rules, as stated in paragraph . Found source of reference. For clarity, the following example.

The function f (*A*, *B*, *C*), three logical variables is given by the following relation:

$$f_{(A,B,C)} = (A \times \overline{B} \times C + A \times B \times C) \times (A \times C + A \times \overline{C}) \times (\overline{C} + C\overline{A}) + \overline{A}$$

For its implementation need 14 logic blocks and the logic diagram is depicted in the following picture.

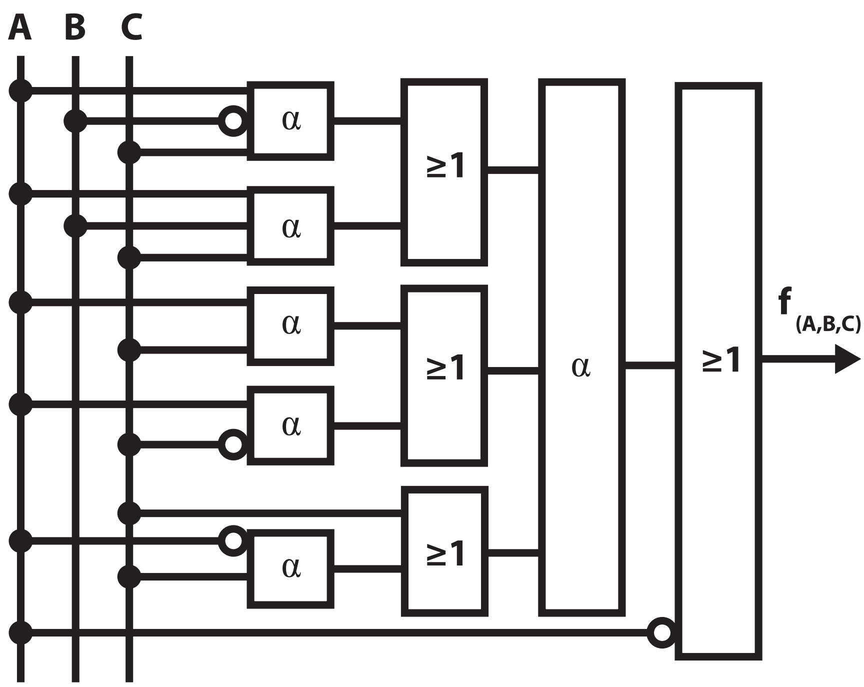


Fig. 3.10: Example 3.3

After making changes mathematical logic function is the following result:

$$\begin{aligned} f_{(A,B,C)} &= A \times C \times (\overline{B} + B) \times A \times (C + \overline{C}) \times (\overline{C} + \overline{A}) + \overline{A} \\ &= A \times C \times (\overline{C} + \overline{A}) + \overline{A} \\ &= \overline{A} \end{aligned}$$

This means that the result does not depend on the values of the logical variables *B* and *C*. The result is only negation of *A*. Implementation in the logical diagram in the following figure:

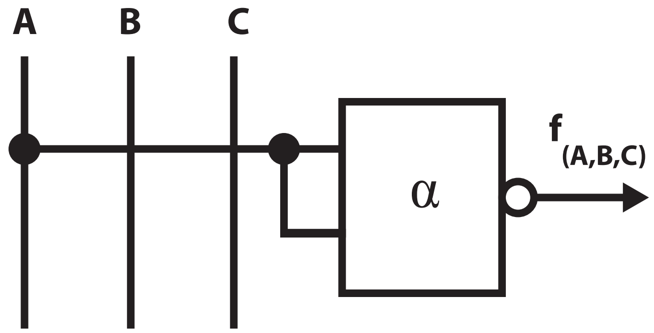


Fig. 3.11: Negation

Comparing the two schemes, it is clear that there is a significant saving of logic blocks and thus reduces the probability of failure control system.

3.4 Mathematical programs

For solving logic circuits is a large number of general and specialized programs. These include Visual Basic, Fortran, C + + and spreadsheets MS Excel and Open Office Calc. There are a number of specialized programs for drawing electrical and automation schemes, which can also design the circuits and to address their behavior. To address the study of systems and is very suitable programming environment MATLAB.

3.5 MATLAB

It is an integrated environment for scientific calculations, which contains a powerful graphics and computational tools. It is primarily intended for technical computing, data visualization and analysis, development and testing of algorithms, data acquisition and processing, signal and image processing, programming and creating custom applications. The advantage of this environment is that it contains a number of tools that allow researchers to focus on solving the problem, not the syntax.

For this purpose MATLAB contains tools:

- Computer tools and development environment,
- Library of mathematical functions,
- Language MATLAB,
- MATLAB Graphics,
- Eternal interface.

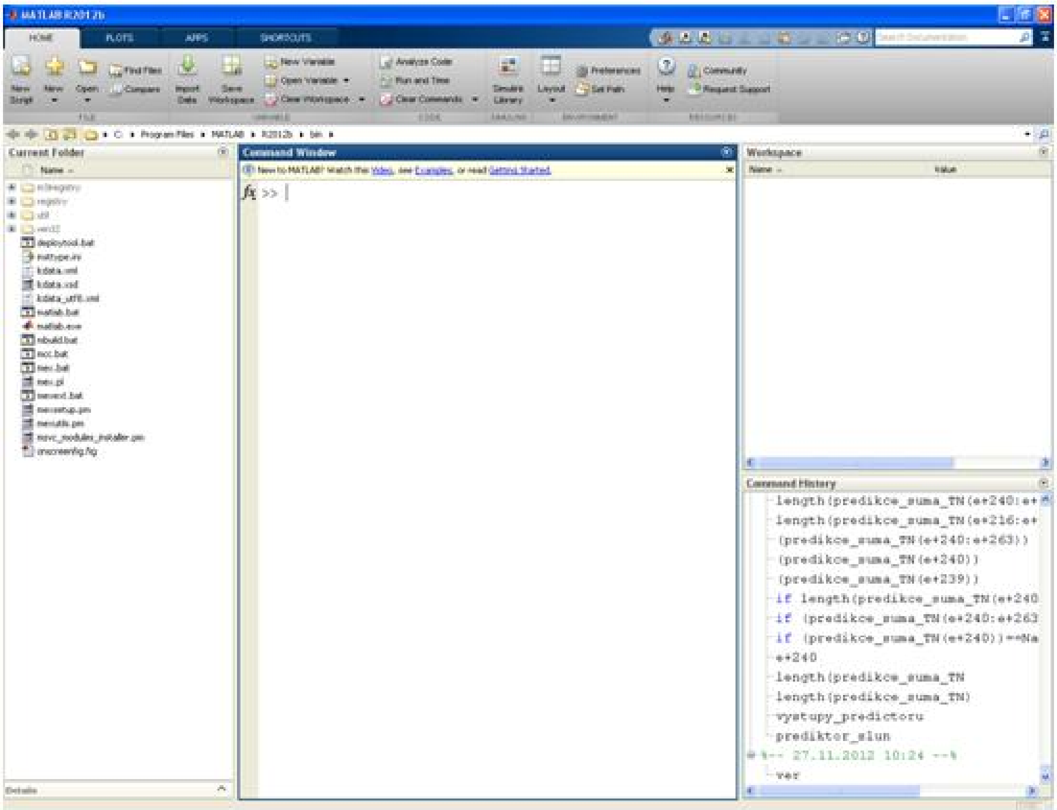


Fig. 3.12: Main MATLAB screen

The location in the directory and the current version

By default, after installing the program is located in the directory: "c: \ Program Files \ MATLAB \ R2012b \ *. *" For each version. The current version is version No. 8 Folder "R2012b" is a trading name of the installed update. Every year, two distributed updates, spring, marked with the letter "a", autumn, marked with the letter "b". Installing the new update will not overwrite the old one, but it creates a new folder:

```
c:\Program Files\MATLAB\R2012a\*.*
c:\Program Files\MATLAB\R2012b\*.*
```

The current version is found then sends a command to the command line:

```
>> ver
---------------------------------------------------------------------------
MATLAB Version: 8.0.0.783 (R2012b)
MATLAB License Number: 270637
Operating System: Microsoft Windows XP Version 5.1 (Build 2600: Service
Pack 3)
Java Version: Java 1.6.0_17-b04 with Sun Microsystems Inc. Java HotSpot(TM)
Client VM mixed mode
---------------------------------------------------------------------------
MATLAB                                          Version 8.0        (R2012b)
Simulink                                        Version 8.0        (R2012b)
Control System Toolbox                          Version 9.4        (R2012b)
MATLAB Compiler                                 Version 4.18       (R2012b)
Stateflow                                       Version 8.0        (R2012b)
Symbolic Math Toolbox                           Version 5.9        (R2012b)
System Identification Toolbox                   Version 8.1        (R2012b)
Symbolic Math Toolbox                           Version 5.1        (R2008b)
System Identification Toolbox                   Version 7.2.1      (R2008b)
>>
```

Fig. 3.13: Actual version detection

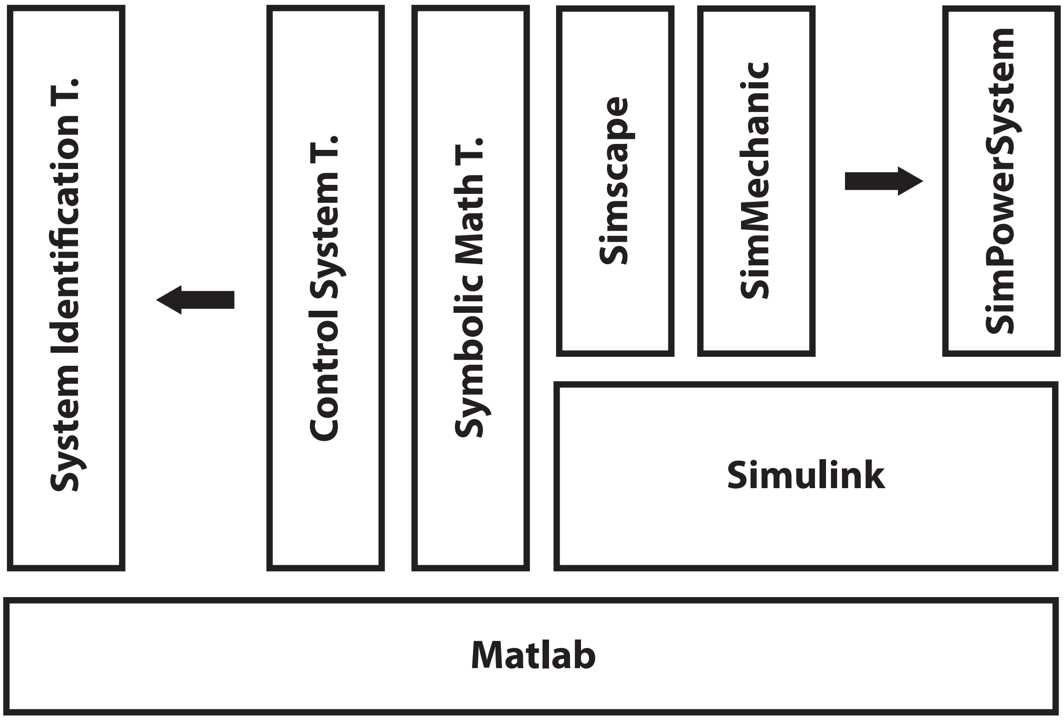


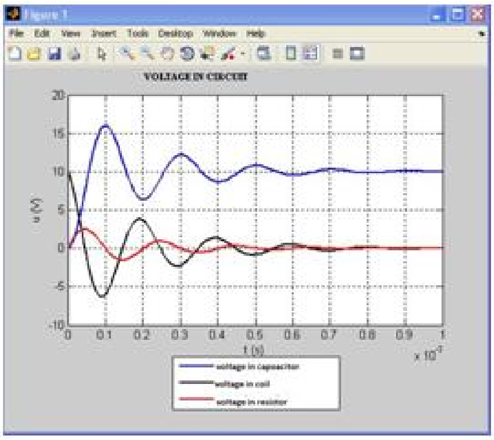
Fig. 3.14: Structure for MATLAB environment

MATLAB is modular and allows for expansion in various areas of computing. The base is expanded library of MATLAB tools labeled as "toolboxes". The oldest and most common toolbox is "Simulink", which allows the simulation of dynamic systems. Expanding Simulink is made of "Block set" libraries usable blocks. Diagram expansion MATLAB is shown in Figure 3.14.

MATLAB documentation is very extensive, and therefore occurs only in electronic form. Documentation contains a basic description of the function and syntax, examples, notes on new releases and printable documentation in "puff" format. The documentation is organized by category and alphabetically.

Example of solution

For example, solutions, and outcomes were selected sample solution voltage and current in a series RLC circuit. The calculation was performed numerical differentiation and integration, and the results shown in the charts below, Table 3.15.



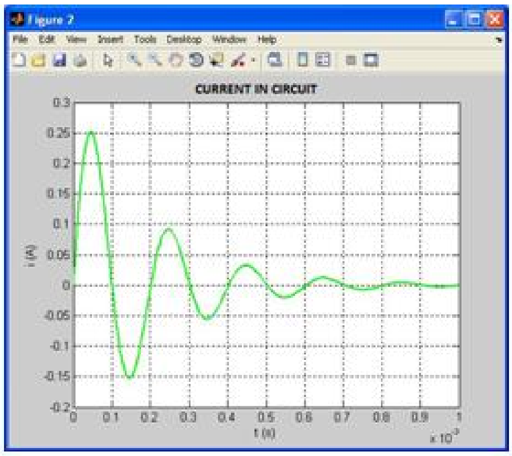


Fig. 3.15: Example for numerical solution results

SIMULINK

The purpose is to extend SIMULINK of MATLAB basic superstructure which will allow calculations of the dynamic behavior of systems environments using block diagrams. For a system described by differential or difference equations, a mathematical model is created using predefined blocks. These blocks are arranged in libraries. These libraries are divided into so-called Blockset:

- Simscape (basic library),
- SimDriveline,
- SimHydraulics - Blockset solution for hydraulic systems and mechanisms,
- SimMechanics,
- SimPowerSystems - Blockset solutions for high-voltage electrical systems.

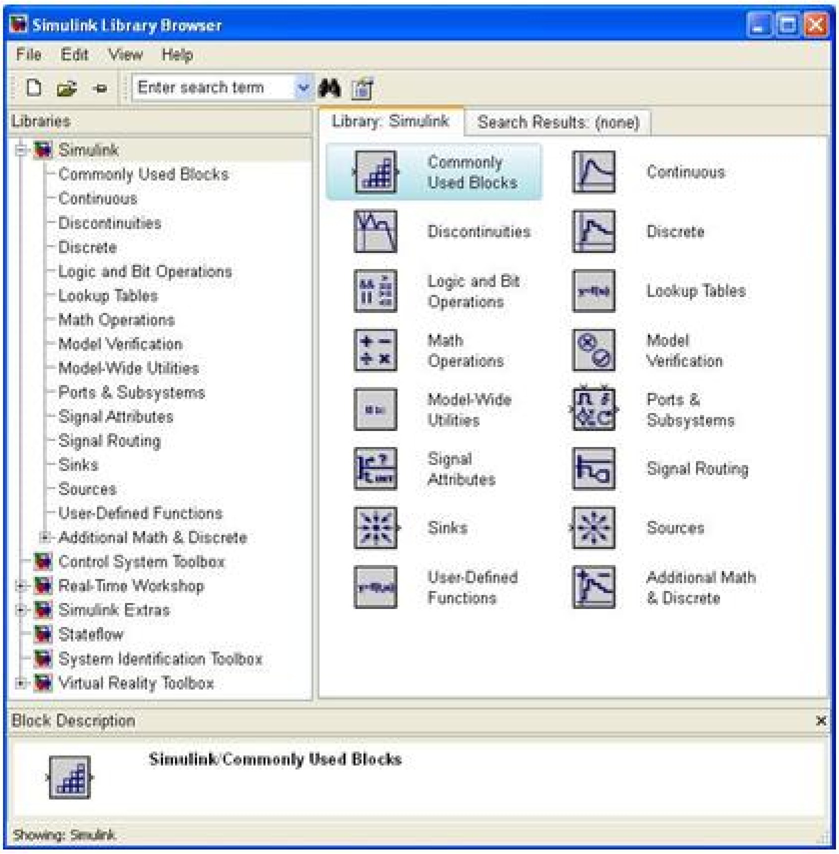


Fig. 3.16: Simulink libraries

Blockset "Commonly Used Blocks" offers the most frequently used blocks.

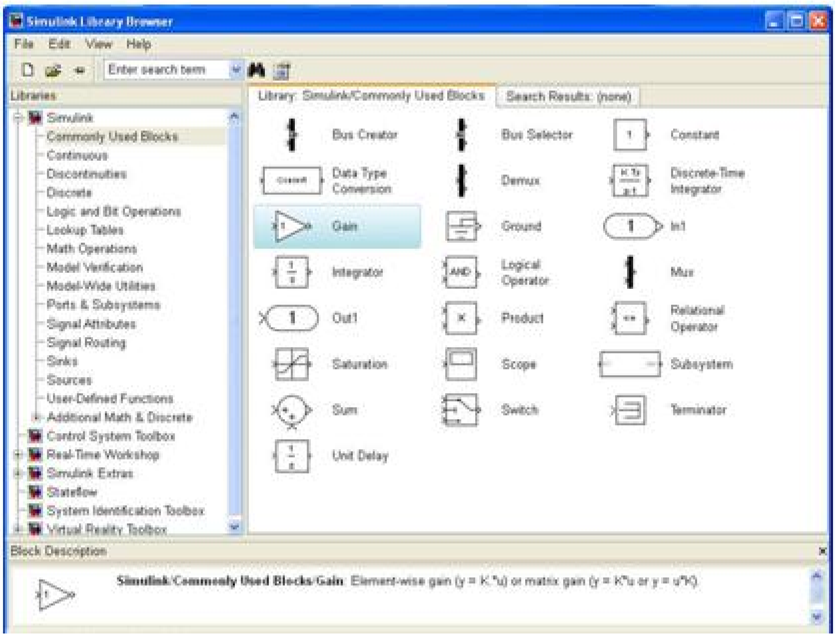


Fig. 3.17: Blockset example

For a comparable example was chosen sample solution dynamic behavior of RLC circuit as in the previous case. Given integro-differential equation was modeled using predefined blocks, the window on the right side of the picture and the course of the unit step response is displayed on the screen oscilloscope window on the left of picture 3.18.

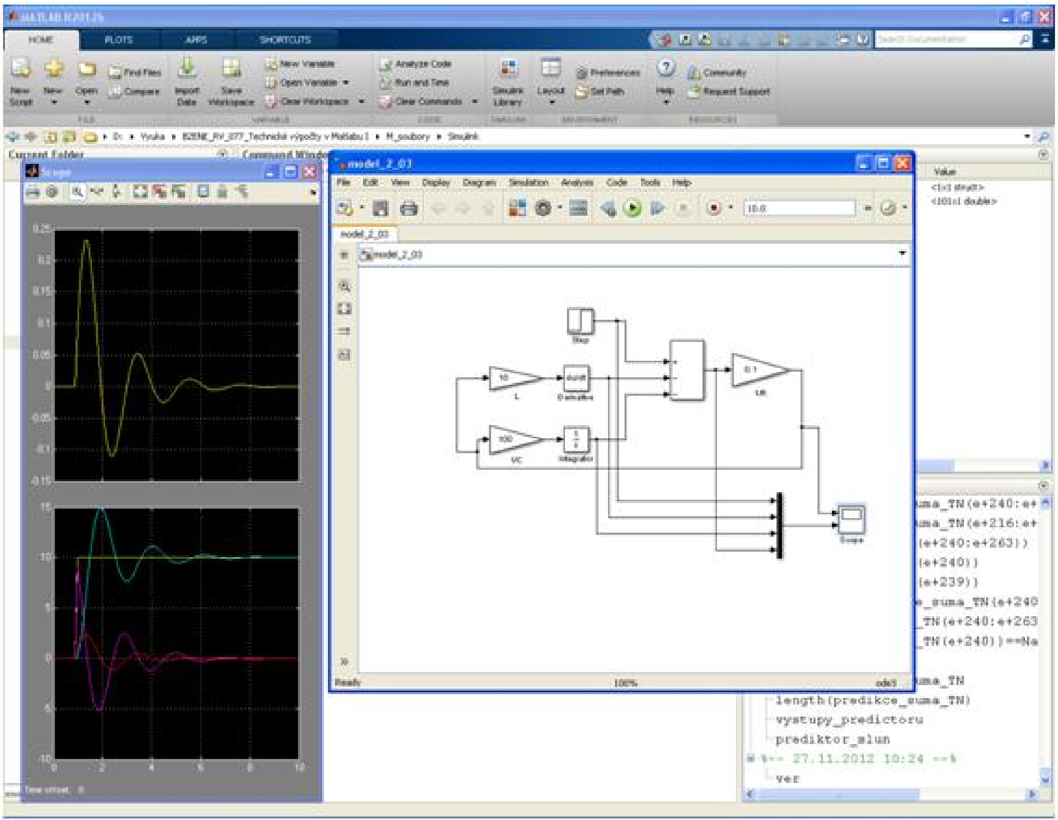


Fig.3.18: Example solution for system behaviour in time

3.6 Repeating questions

1. Draw the schematic symbols for logical operations.
2. Describe the ways in which it is possible to realize the logic control circuitry.
3. Draw implement logical sum of the control scheme.
4. Draw a realization of a logical product of the control scheme.
5. Draw a realization of negation with the control scheme.
6. Draw logical sum realization using transistors.
7. Draw logical product realization using transistors.
8. Draw a realization of negation using transistors.
9. Draw logical sum realization using mechanical linkage.
10. Draw a realization of a logical product with mechanical linkage.
11. Draw a realization of negation by mechanical linkage.

4 Combinational logic functions and Boolean algebra, logical tables, Karnaugh maps, minimization, solid execution logic and combinational logic functions

4.1 Logical Functions

As already mentioned, the theoretical basis for control and regulation of entire machines or technological systems is mathematical logic, and therefore this section focuses on the repetition of certain areas of mathematical logic and the propositional and Boolean algebra. Application of laws and rules, the reader is familiar with the simplification of logic expressions. To find the mathematical expression verbal expression is introduced by one of the possible ways of using Karnaugh map. In chapter 3 also introduced the implementation of the mathematical formulation of technical means.

4.2 Propositional algebra

Verdict

The term verdict is possible to imagine a statement of various people, various weather lore, definitions, declarations, etc. In mathematical logic, the term is restricted definition: "The award of each claim, which can be unambiguously assigned logical value at all times." This is specified, what is the verdict expected. Statement may be a proposition whose truth will be decided in the future.

Statement can have two values: true, false. To mark truthful statements can be used symbols: yes, true, 1, +, high; H. The false statements are used designation: no, false, 0, -, low; L.

For better understanding, are a few examples?

- "The voltage drop across the resistor is proportional to the electric current flowing through." This is a statement that is true, the value is true.
- "The inductance is equal to the product of current and voltage." This is a statement which is false, the value is false.
- "In the constellation of Orion are living sentient beings." This is a statement, and its value will be known in the future. The result will answer yes or no.
- "What's out there?" This is not a statement. Generally prefixed questions "how much" into group saying that no statement.
- "Go wash your hands!" This is not a statement, because it is an imperative sentence.

Propositional calculus

Number of propositional understand that part of mathematical logic that examines relationships between statements only with respect to their truth and falsity. Propositional calculus does not deal with the internal structure of atomic propositions and laws according to which forms.

Character or verbal expression with which to form new statements is called functors <predicament able> [logical connectives]. Propositional statement, atomic number is called a statement without functors.

The most important functors are:

- Functors negation - a sign above the bar statement $\overline{A}$ and phrase is: "no, not, not true that ...".
- Functors conjunctions – sign is $A \wedge B$ a logical product of the phrase is: "and, at the same time, and".

- Functors disjunction - sign is A∨B. The phrase logical sum is: ", or, or."
- Functors implications - sign is A→B. A logical implication is: "from ... that ..., if ..., then ...!.
- Equivalence functors - sign is A ↔ B. The phrase logical equivalence is "just as, if and only if ..."

Combinations of atomic propositions are formed propositional operations from the simplest to the complex. Their significance is that they can be easily interpreted technically.

Logic function

In classical algebra is known definition of a function: function display when one or more independent variables corresponding to one or more dependent variables. Thus, in mathematical logic to define the term logical function. Logic function is a relation between the dependent and independent variables logic. Logical variables are binary variables that take values 0 or 1

Function logic variables may be a function of one or more variables.

$y = f(x_1; x_2; x_3; ...; x_n)$

Each logical function can be expressed in three ways with equivalent results:

- Combining negation functors, disjunction, conjunction
- Pierce using multiple functions. Pierce function is called functor NOR. It is the negation of the disjunction of two statements.
- Sheffers using multiple functions. Sheffer function is called functor NAND. It is the negation of the conjunction of two statements. Its use is preferred for the reason that every operation in the logic function to be implemented by one technical element. Expression logical functions Sheffer function is advantageous, because it is at least technical elements.

Logic functions can solve logical table. This is a short statement of all combinations of independent variables, which may occur. Total number of options that can occur is calculated by the formula:

$k = 2^n$

where k is the number of all possible n is the number of variables.

Kind of solution is illustrated in the following example.

Example 4.1

Using a truth table for a logic function Y, which is a function of three variables and logic is expressed by the formula:

$$Y = (A + \overline{B}) \times (B + C) \times (C + \overline{A})$$

According to the formula is calculated as 8 choices for the independent variables.

A	B	C	Not A	Not B	A+not B	B+C	C+not A	Y
0	0	0	1	1	1	0	1	0
0	0	1	1	1	1	1	1	1
0	1	0	1	0	0	1	1	0
0	1	1	1	0	0	1	1	0
1	0	0	0	1	1	0	0	0
1	0	1	0	1	1	1	1	1
1	1	0	0	0	1	1	0	0
1	1	1	0	0	1	1	1	1

The first three columns are the independent variables (atomic statements) which is recommended to fill gradually from the last column (in this case C) alternating values 0 and 1 The next column (in this case B) alternating values 0 and 1 at double the number of lines before the last column (in this case C). The last column (in this case A), then alternating values 0 and 1 at double the number of lines than the previous column (in this case B). In this way it is possible to proceed in the case of multiple variables. Other columns are gradually solving logic functions. The last column is the result of a logical function. To search for a combination of independent variables, it is possible to find the appropriate row resulting value logic functions.

4.3 Boolean algebra

Technical elements used in the regulation and control of machines as buttons or switches. This is a two-valued element. For a mathematical description of these components was created two valued algebra, which, after its creator, British mathematician Boole, called Boolean algebra. These are the calculations with binary variables. For computations in Boolean algebra defined laws and rules as in other algebras. These are the laws and rules.

- Commutative law
- The associative law
- Distributive Law
- The rule of neutrality and aggressiveness 0 and 1
- Rule of Independence (Independence) elements
- The rule of the excluded middle
- The rule of double negation
- De Morgan's rules
- Rule of the uptake and absorption of negation

Legislation

Basic laws of commutative, associative and distributive, which are defined for any algebra, thus Boolean (Table 2), are expressed in two forms, disjunction and conjunction. In classical algebra is about addition and subtraction. In Boolean algebra is a logical sum and logical product.

Table 2: Main rules

Property	**Disjunction**	**Conjunction**
Commutative property	$A \vee B = B \vee A$	$A \wedge B = B \wedge A$
Associative property	$(A \vee B) \vee C = A \vee (B \vee C)$	$(A \wedge B)\ C = A \wedge (B \wedge C)$
Distributive property	$(A \vee B) \wedge C = A \wedge C \vee B \wedge C$	$(A \wedge B) \vee C = (A \vee C)\ (B \wedge C)$

In practice, the logical functors disjunction, conjunction and negation may use other methods of marking. Sample labeling the various ways in Table 3.

Table 3: Different type designation for logical operations

$A \vee B$	$A + B$	$A \cup B$	A or B
$A \wedge B$	$A \times B$	$A \cap B$	$A \alpha B$
$\overline{A}$	$\neg A$		not A

Rules

Boolean algebra is complemented by a set of rules that are used to simplify logic functions. List of all the rules in Table 4.

Table 4: Boolean algebra rules

Rules	**Addition**	**Multiplication**
Rule of neutrality 0 and 1	$A + 0 = A$	$A \times 1 = A$
Rule of aggressiveness 0 and 1	$A + 1 = 1$	$A \times 0 = 0$
Rule of independence elements	$A + A = A$	$A \times A = 0$
Rule of excluded middle	$A + \overline{A} = 1$	$A \times \overline{A} = 0$
De Morgan's rule	$\overline{A} + \overline{B} = \overline{A \times B}$	$\overline{A} \times \overline{B} = \overline{A+B}$
Rule of absorption	$A + A \times B = A$	$A + (A \times B) = A$
Rule of absorption negation	$A + \overline{A} \times B = A+B$	
Rule of double negation	$\overline{\overline{A}}=A$	

Example 4.2

Simplify the function of three variables:

$$f_{(A,B,C)} = (A \times \overline{B} \times C + A \times B \times C) \times (A \times C + A \times \overline{C}) \times (\overline{C} \times C\overline{A})$$

First, it is done before pointing out brackets and braces is a third application of the rule of absorption of negation.

$$f_{(A,B,C)} = A \times C \times (\overline{B} + B) \times A \times (C + \overline{C}) \times (\overline{C} \times \overline{A})$$

Furthermore, the rule is applied Independence elements is performed multiplication brackets and application rules exclude the third. The result is 0

$$f_{(A,B,C)} = A \times C \times (\overline{C} \times \overline{A}) = 0$$

This means that the function value is always zero, regardless of the combination of values of input variables, a function is called forgery.

4.4 Karnaugh map

In the previous section was solved when it is known mathematical expression of logic functions. In fact, there may be, and very often it is necessary to deal with cases where the outcome of logical functions and it is necessary to find a mathematical expression. For this purpose so-called graphical expression maps. One of the best known and most widely used is Karnaugh map. It's a different way of expressing the logical table, which is used to simplify logical expressions. For a better illustration of the problem is the following example. The logical table is a logical expression of results for all combinations of independent variables.

Example logical table for the three independent variables is shown in the given example 4.1. In the first three columns are expressed in values of independent variables. The last column (far right) shows the values of the dependent variable.

If Karnaugh maps the resulting column is displayed as a rectangular array. On each side of the rectangle are expressed in values of the independent variables. It does not matter who, regardless of which side variable is assigned. The values of the independent variables, it is appropriate to fill out so that those values in adjacent column/row have been changed by one bit. On the thus filled can be seen as are the cells that make up the array of values dependent variable. The cells of this array are filled with the values of the dependent variables based on combinations of independent variables. For a better understanding is given the following worked example with explanations.

Example 4.3

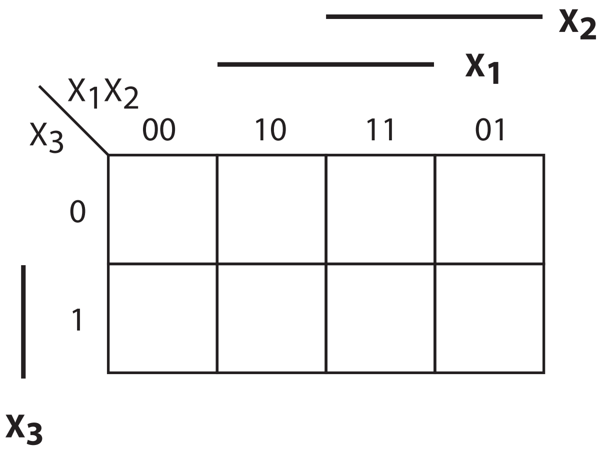
This award is given to the fire alarm system. In the area there are three independent fire sensors. When at least two sensors record the fire, it is necessary to declare a fire alarm. Expressed mathematically: For logical function of three variables Y, the value of the function Y takes values true when at least two input variables come true values.

Problem resolution:

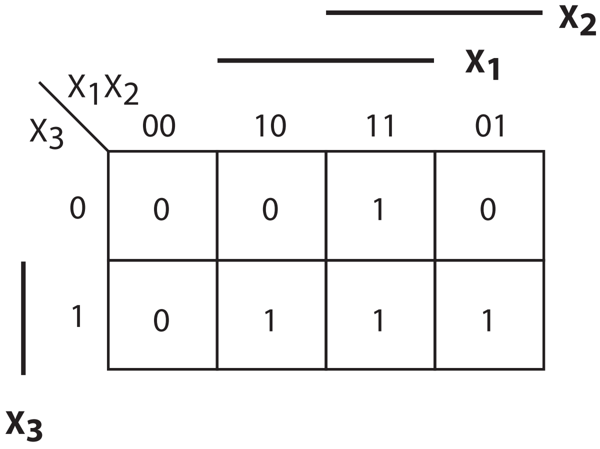
First, the embodiment shown in the logical table problem. This is a function of three independent variables "$Y = f(X_1, X_2, X_3)$." The number of all combinations of independent variables is calculated according to the formula No. 3.1 A is equal to the eighth This means that the logical table will have outside headers 8 lines. In accordance with the procedure referred to in the given example No. 4.1 is done filling in the values of independent variables. Column for the dependent variable "Y" is filled in based on the formulation of the assignment. The award is very important to pay attention to expression, whether it is "at least two" (our example) or "only two." In both cases the result is a different solution. Created a logical table for the specified example is as follows:

X_1	X_2	X_3	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Field values for the dependent variable "Y" can take various forms, such as "2 x 4" or "4 x 2". For arrays with eight cells are more options. In this case, the chosen shape Karnaugh maps "2 x 4". The horizontal side of the field were assigned to the independent variables "X_1" (top row headers) and "X_2" (bottom row headers). To the vertical side of the field was assigned to the remaining independent variable "X_3". Assignment of the independent variables to the parties only depends on the tastes of solvers and do not affect the outcome of the solution. In the next step of the process are filled with values of independent variables so that, when moving to an adjacent column/row is changed by one bit (Gray code). The result is shown in the following table:

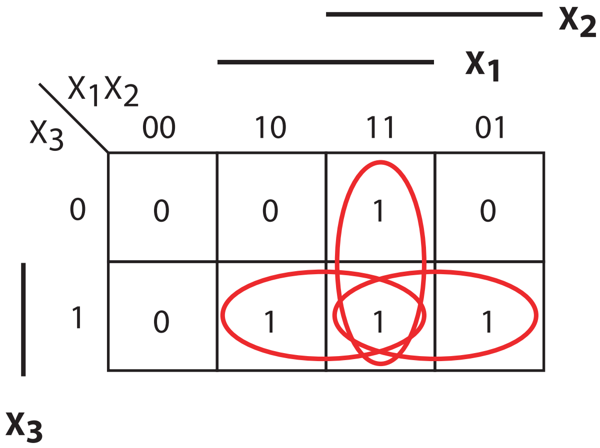


The cells are filled with values of field dependent variable to reflect the combined values of the independent variables:



The solution is carried out the following procedure. The map will combine field contains the value 1 in blocks (objects) according to the following rules:

- Blocks should be as large as possible to cover all the zeros in the blocks as possible.
- The number of related fields in a block must be equal to an integer power of 2 Thus, (1, 2, 4, 8, 16, etc.).
- Individual blocks can overlap.



Logical expression of one block is called term and consists of conjunctive input variables that do not alter its input value. Minimum logic function is created by these disjoint terms. The resulting mathematical expression logic function is:

$$Y_{(X_1;X_2;X_3)} = X_1 \times X_2 + X_1 \times X_3 + X_2 \times X_3$$

Karnaugh map can also solve the coverage of fields with zeros. The resulting logic function is the negation of a disjunction of terms. The previous example would be the following:

$$Y_{(X_1;X_2;X_3)} = \overline{\overline{\overline{X_1} \times \overline{X_2} + \overline{X_1} \times \overline{X_3} + \overline{X_2} \times \overline{X_3}}}$$

Form solution map is chosen depending on which fields are less.

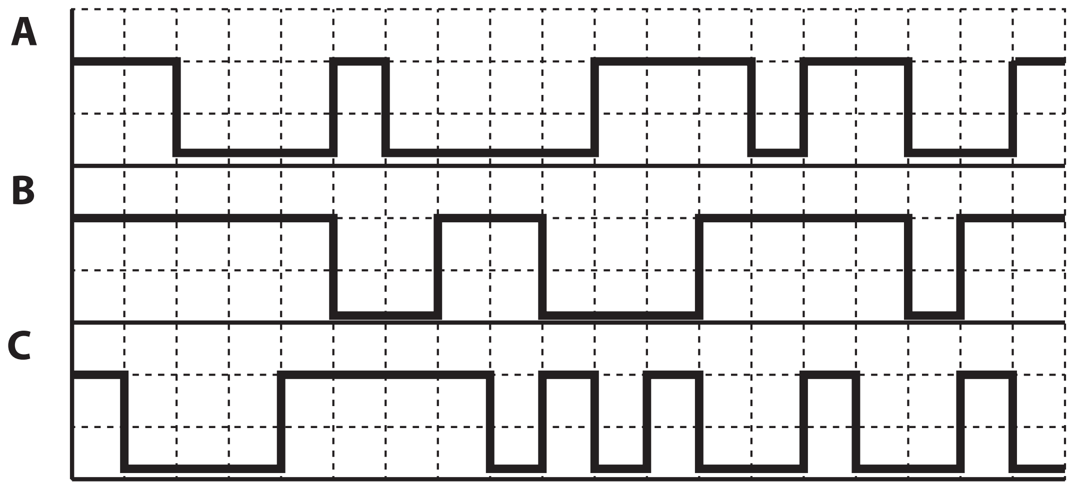
4.5 The time sequence of logic signals

In the previous section the logic circuits were designed as a static, solution was made without the influence of time that is the search for all combinations. In real life, it is necessary to keep in mind that depending on the time will change the values of the independent variables. It follows it may be the value of the dependent variable change with time as shown in the example. Simultaneously, over time, may but not occur all combinations of the independent variables.

Example 4.4

For the expression find time course of output logic values.

$$Y = A \times C + B \times C + A \times B$$

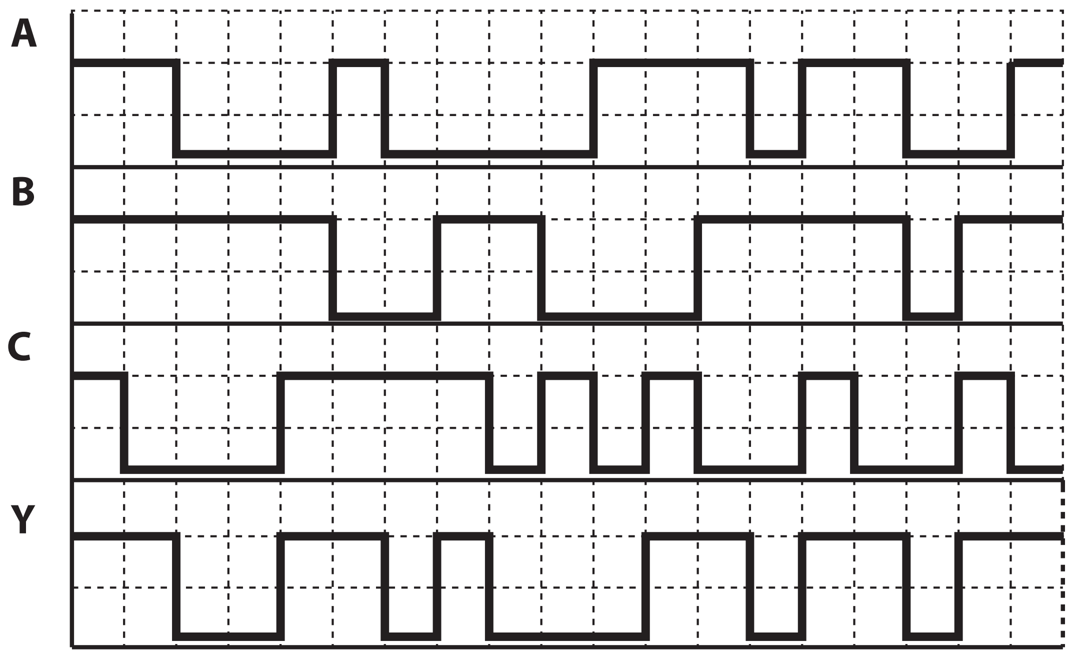


The time course of independent variables is shown in the graph. Examples of this type has two possible solutions.

a) Solution the first way consists in creating truth table for all combinations of independent variables. The table and the appropriate combination of independent variables in the interval search result value logic functions.

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

b) In the second method the solution you need to realize that the logical function is the disjunction of three conjunctions. In solving with the proceeds of conjunctions. The first $A \times C$ conjunction says that the result takes the value 1 if at the same time both the independent variable equal to 1. In the graph are searched those time intervals in which A and C are equal to 1 at the same time and in these intervals is Y = 1. This method is used for other conjunctions. In conclusion, some of the time intervals remain unfilled, those must be assigned a value of 0.



Implementation of logic function

In the field of automation technology the machine control or files subject to the gradual fulfillment of specified conditions. Any condition can be expressed as a logical variable, because it can also, as a logical variable, assigning two values – "passed" or "failed". Therefore simple feasibility logic variables and thus logic functions. The logic of this problem is the best approach for example.

Example 4.5

There is one logical independent variable "X". This variable can take two values 1 and 0. The second logical dependent variable "Y" is the result of a logical function:

$$Y = f(X)$$

All options, which can assume the dependent variable Y are described in the table.

X	Y_1	Y_2	Y_3	Y_4
1	0	0	1	1
0	0	1	0	1

It is clear that there are four options, that means four logical functions; forgery, negation, and equivalence verum. Their practical implementation will be presented in the following simple electrical circuit, composed of the power supply, buttons, bulbs and wires. Independent variable X is the mechanical force applied to the button, the dependent variable is the bulb glow. First logical function Y_3. When the button is pressed, light bulb, the value of Y = 1 if and only if X = 1.

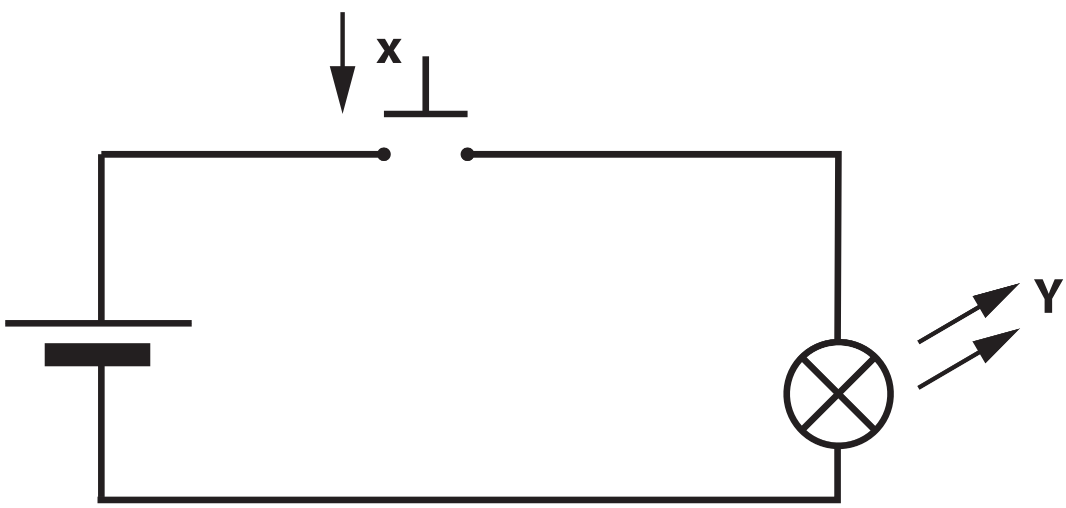


Fig. 4.1: Logical function Y_3

Logic function Y_2. When the button is pressed, the lamp is dark, the value of Y = 1 if and only if X = 0

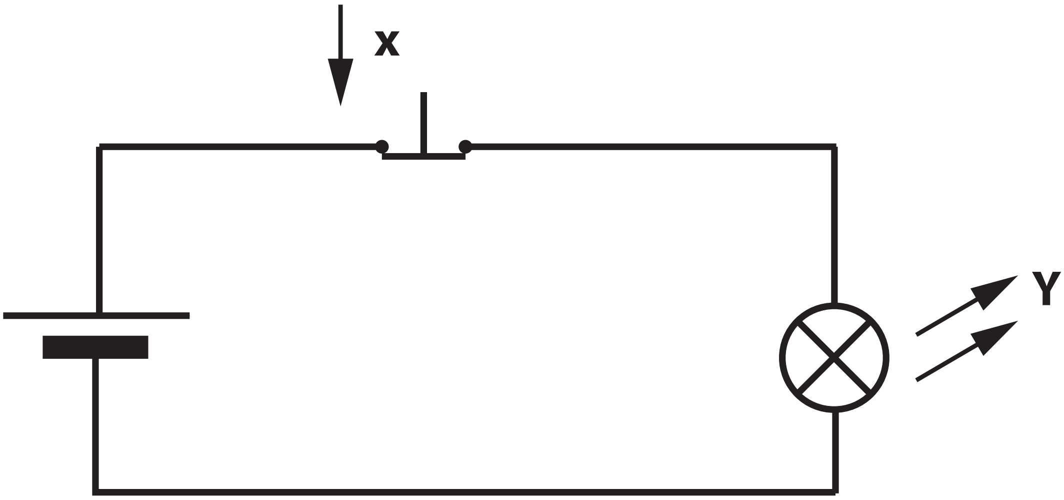


Fig. 4.2: Logical function Y_2

Logical Y_4. It does not matter if the button is pressed or not, the bulb lights up every value Y = 1 and is not dependent on X.

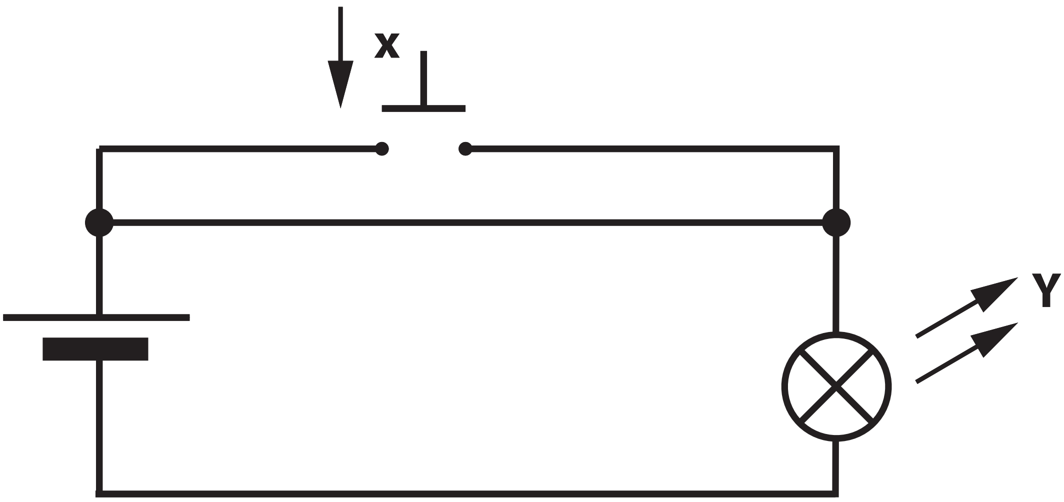


Fig. 4.3: Logical function Y_4

Logic function Y_1. It does not matter if the button is pressed or not, the bulb does not light, the value Y = 0 and does not depend on X.

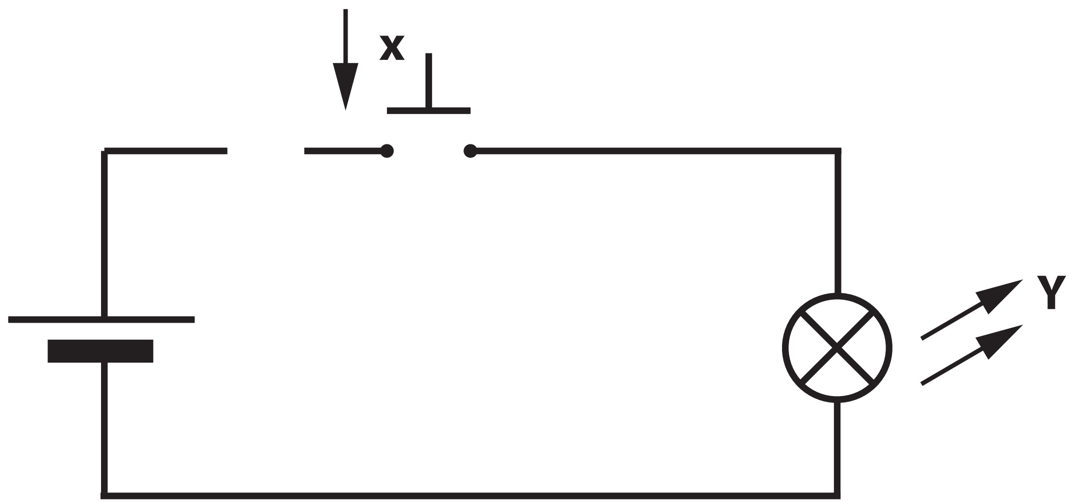


Fig. 4.4: Logical function Y_1

4.6 Repeating questions

1. What claims can and cannot be a statement and give examples.
2. What do you know propositional operations?
3. Show and explain an example of conjunctions statements.
4. Show and explain an example of disjunction statements.
5. Show and explain an example of negation statements.
6. Explain the concept of logical functions.
7. Describe the logical functions of one variable.
8. What determines the number of combinations of logical functions of several variables?
9. What you know the basic laws and rules of Boolean algebra?
10. What maps are and what they are for?
11. What do you understand by the term chronology signals?
12. What is the difference between a diagram, showing the chronology and logical table?

4.7 Examples for practice

Use logical tables verify the validity of the term:

1. $A + B \times B = A$
2. $A + \overline{A} \times B = A + B$
3. $\overline{A \times B} = \overline{A} + \overline{B}$
4. $\overline{A + B} = \overline{A} \times \overline{B}$
5. $A \times (A + B) = A$
6. $A \times B = \overline{\overline{A} + \overline{B}}$
7. $A \; xor \; B = \overline{A \times B + \overline{A} \times \overline{B}}$
8. $(A \; xor \; B) = \overline{A \times B + \overline{A} \times \overline{B}}$
9. $A \times \overline{B} + A \times C + B \times \overline{A} = \overline{A} \times B + \overline{B} \times A + B \times C$
10. $(A + \overline{B}) \times (B + \overline{C}) \times (C + \overline{A}) = (\overline{A} + B) \times (\overline{B} + C) \times (\overline{C + A})$

SOLUTION

1. $L = P$
2. $L = P$
3. $L = P$
4. $L = P$
5. $L = P$
6. $L = P$
7. $L = P$
8. $L \neq P$
9. $L = P$
10. $L \neq P$

Simplify and verify the veracity of using logical tables.

$$f_{(A,B,C)} = (A \times \overline{B} \times C + A \times B \times C) \times (A \times C + A \times \overline{C}) \times (\overline{C} + C\overline{A})$$

$$f_{(A,B,C)} = (A \times \overline{B} \times \overline{C} + A \times B \times C) \times (A \times C + A \times \overline{C}) \times (\overline{C} + C\overline{A})$$

$$f_{(A,B)} = A \times (\overline{A} \times B + B)$$

SOLUTION

$$f_{(A,B,C)} = 0$$

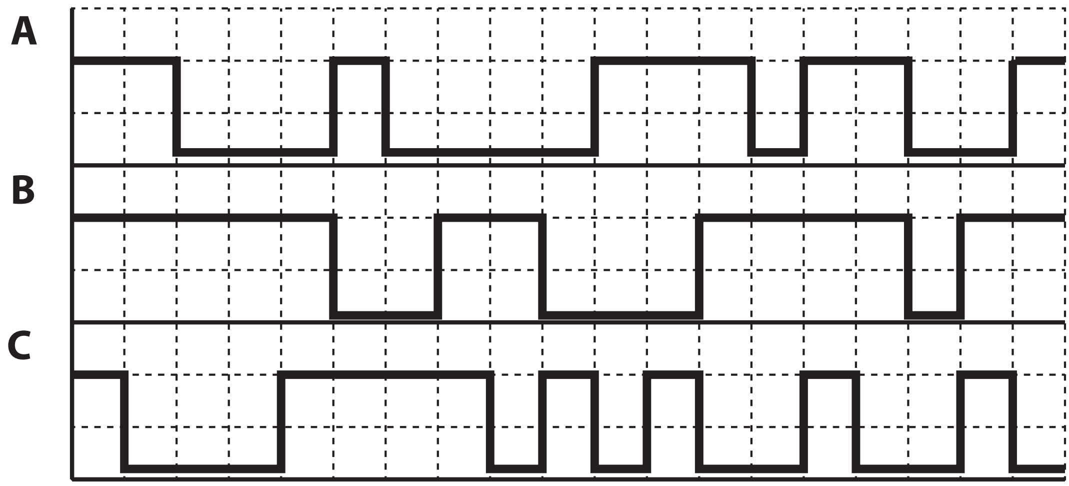
$$f_{(A,B,C)} = A \times \overline{B} \times \overline{C}$$

$$f_{(A,B)} = A \times B$$

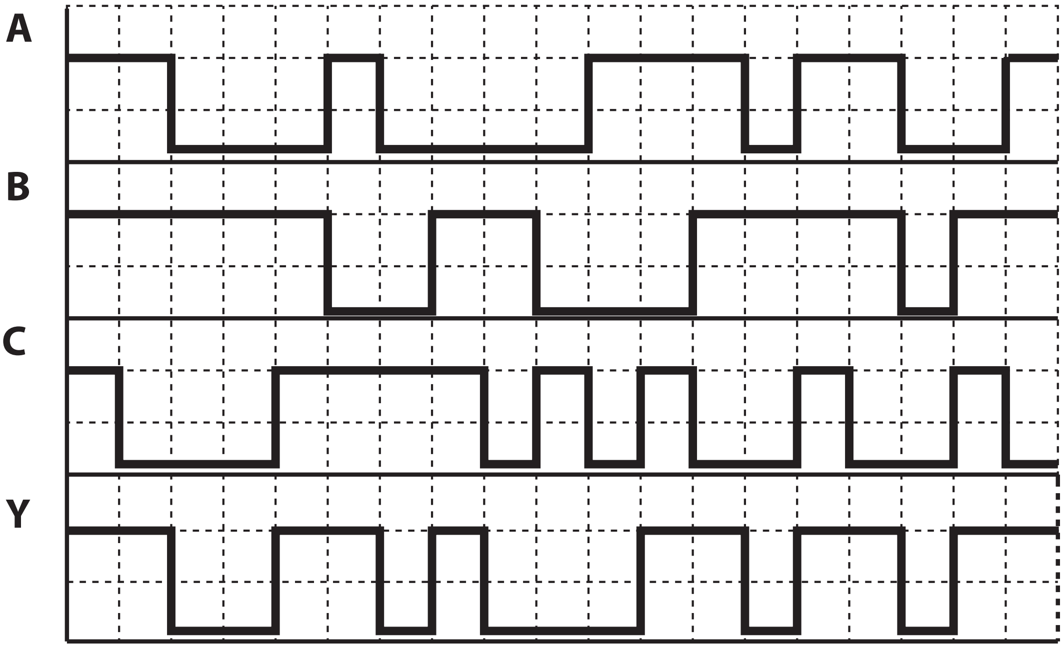
Example 4.5

Find the time course of the output signal of a logic function

$$Y = \overline{A} \times \overline{B} \times C + \overline{A} \times B \times C$$



Solution



Example 4.6

Write the logical table for the function of four variables, where Y = 1, where only two inputs is 0, find a mathematical expression using Karnaugh maps.

Solution

$$Y = C \times D \times \overline{A} \times \overline{B} + C \times \overline{D} \times A \times \overline{B} + C \times \overline{D} \times \overline{A} \times B +$$
$$+ \overline{C} \times \overline{D} \times A \times B + \overline{C} \times D \times A \times \overline{B} + \overline{C} \times D \times \overline{A} \times B$$

$$Y = \overline{A \times B \times C + A \times C \times D + B \times C \times D + A \times B \times D +}$$
$$\overline{+ \overline{B} \times \overline{C} \times \overline{D} + \overline{A} \times \overline{B} \times \overline{D} + \overline{A} \times \overline{C} \times \overline{D} + \overline{A} \times \overline{B} \times \overline{C}}$$

Example 4.7

Write a logical table for the function of the three variables, where Y = 1 if simultaneously on at least two inputs a value of 1, find the mathematical expression using Karnaugh map.

Solution

$$Y = A \times B + A \times C + B \times C$$

$$Y = \overline{\overline{A} \times \overline{B} + \overline{A} \times \overline{C} + \overline{B} \times \overline{C}}$$

Example 4.8

Write the logical table for the function of four variables, where Y = 1, if currently only three inputs a value of 1, find the mathematical expression using Karnaugh maps.

SOLUTION

$$Y = A \times B \times \overline{C} \times D + \overline{A} \times B \times C \times D +$$
$$+ A \times \overline{B} \times C \times D + A \times B \times C \times \overline{D}$$

$$Y = \overline{\overline{A} \times \overline{B} + \overline{C} \times \overline{D} + \overline{A} \times B \times \overline{C} + \overline{A} \times C\overline{D} +}$$
$$\overline{+ A \times \overline{B} \times \overline{C} + C \times \overline{D} \times \overline{A} + C \times \overline{D} \times \overline{B} + A \times B \times C \times D}$$

4.8 Systems distribution

Control systems are divided by function into three groups. These are the groups:

- logical systems,
- digital systems,
- hybrid systems.

Logical Systems

Logical systems are systems that interact with their environment using two-valued elements (yes, no, or 1.0, etc.) these systems process information according to the rules of Boolean algebra and are therefore also called Boolean or binary systems. Binary system can be implemented either a contacting or non-contacting switch.

Digital Systems

Digital systems are systems which work with arithmetic operations and are controled by algebraic rules. In particular, the basic algebraic operations of addition, subtraction, multiplication and division, but it is not excluded or the use of more complex operations or functions. Digital system is based on a microprocessor.

Hybrid systems

Combining logical and numerical operations in one system is a hybrid system.

Control systems are also divided according to the method of information processing in the two groups. These are the groups:

- combinational control
- sequential control

Combination control

When the combination reacts driven control system as a combinational logic circuit, i.e. the value of the output depends only on the combination of input values.

Sequence control

Sequence control unit controls the system depending on the time or the state of the controlled process. Alternatively, it is possible to combine both functions.

4.9 Repeating questions

1. What are the kinds of systems?
2. You know what types of proceedings?

4.10 Combinational logic function

All logic circuits can be divided according to the way the proceedings into two large groups. First group of logic circuits are called combinational logic circuits with them will be described in this chapter. Logic circuits of the second group are sequential logic circuits that are covered in chapter No. 5. The fundamental difference between the two groups is the ability of memory circuit status, and time control.

Solution of combinational logic circuits

Combinational logic circuits have no memory and their output depends only on the combination of inputs. These circuits can be created using a combination of:

- logic gates AND, OR, and inverter or:
- function NAND (Sheffer function),
- NOR function (Pierce functions).

To solve combinational logic circuits is possible to use two forms of solution: disjunctive normal form or conjunctive normal form. Both forms list all logical functions, which can be a solution to the task. Disjunctive normal form looks for solutions in which the output variable takes values 1 In contrast, normal form conjunctive searching solution when the output variable with values 0, which is a partial solution to the problem. The final negation partial solution is obtained by solving the task. The result of the solution does not depend on the form of a solution. Therefore, the form of solution collected by the difficulty of the procedure. In layman's terms, if it contains more zeros Karnaugh map is selected disjunctive form, otherwise conjunctive form.

Example: 4.9

Write Karnaugh map for the four variables, where Y = 1, where only two inputs a value of 0, find the mathematical expression.

Solution

CD \ AB	00	01	11	10
00	0	0	1	0
01	0	1	0	1
11	1	0	0	0
10	0	1	0	1

Disjunctive normal form:

$$Y = \overline{\overline{A} \times \overline{B} + \overline{C} \times \overline{D} + \overline{A} \times B \times \overline{C} + \overline{A} \times C\overline{D} +}$$
$$\overline{+A \times \overline{B} \times \overline{C} + C \times \overline{D} \times \overline{A} + C \times \overline{D} \times \overline{B} + A \times B \times C \times D}$$

Conjunctive normal form solution:

$$Y = \overline{A \times B \times C + A \times C \times D + B \times C \times D + A \times B \times D +}$$
$$\overline{+\overline{B} \times \overline{C} \times \overline{D} + \overline{A} \times \overline{B} \times \overline{D} + \overline{A} \times \overline{C} \times \overline{D} + \overline{A} \times \overline{B} \times \overline{C}}$$

4.11 Using combinational logic circuits

Given that the combinational logic circuits have no memory, the output information depends only on the combination of input variables. Description of their function, it is advantageous to solve using logical tables. Its use therefore occurs mainly as:

- multiplexers and demultiplexers,
- transmitters codes,
- safety circuits.

4.12 Multiplexers and Demultiplexers

Given that the transmission of information (particularly at greater distances) using a parallel bus requires a larger number of parallel conductors are used for serial information transmission lines. For example, for 16-bit parallel transfer of information over the serial line will suffice six channels of information, one locking, 4 and one addressing information.

Multiplexer (set of instructions) is a device that ensures the transfer of information transmitted in parallel on the bus to information transmitted via a serial line. Demultiplexer (selector instructions) on the other hand a device which divides the serial information to the individual bus wires.

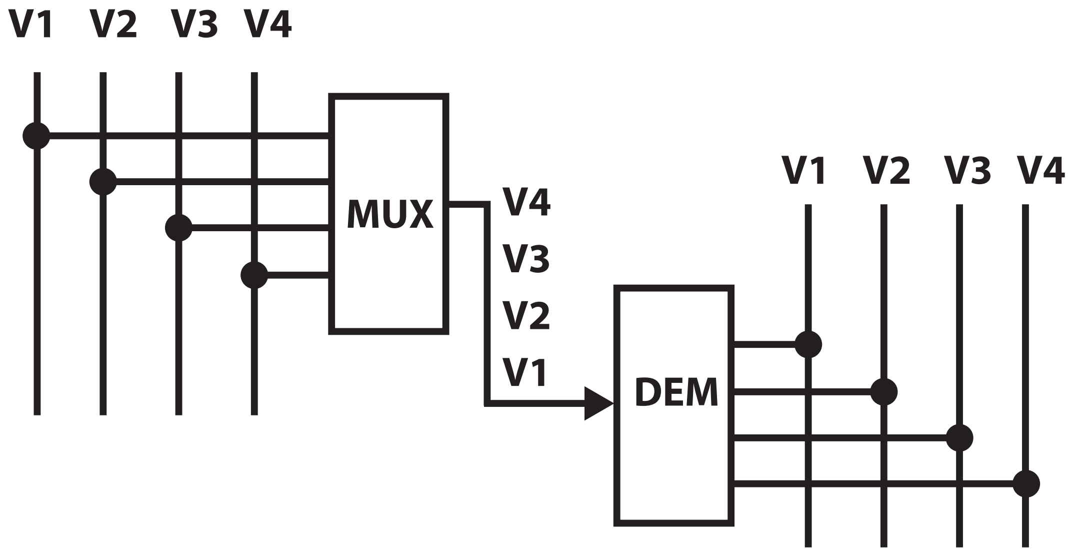


Fig. 4.5: Multiplexer and demultiplexer function

In the left part of Figure 4.6 shows the four bit bus for which information is transmitted. Wire bus has marked V1 to V4. Information is transmitted by a parallel combination of four bits. This information enters the multiplexer (MUX), which is converted to serial bits of information in the sequence V1 to V4. Thus structured information over the serial line progresses to the demultiplexer, which decomposes it back to the bus lines.

Figure 4.6 is plotted four-channel multiplexer logic diagram. V1 to V4 inputs are data inputs (input). Inputs A and B are the address inputs and EN is blocking input. Multiplexer function describes the following logical table.

Table 5: Logical table

EN	B	A	Q
0			0
1	0	0	V1
1	0	1	V2
1	1	0	V3
1	1	1	V4

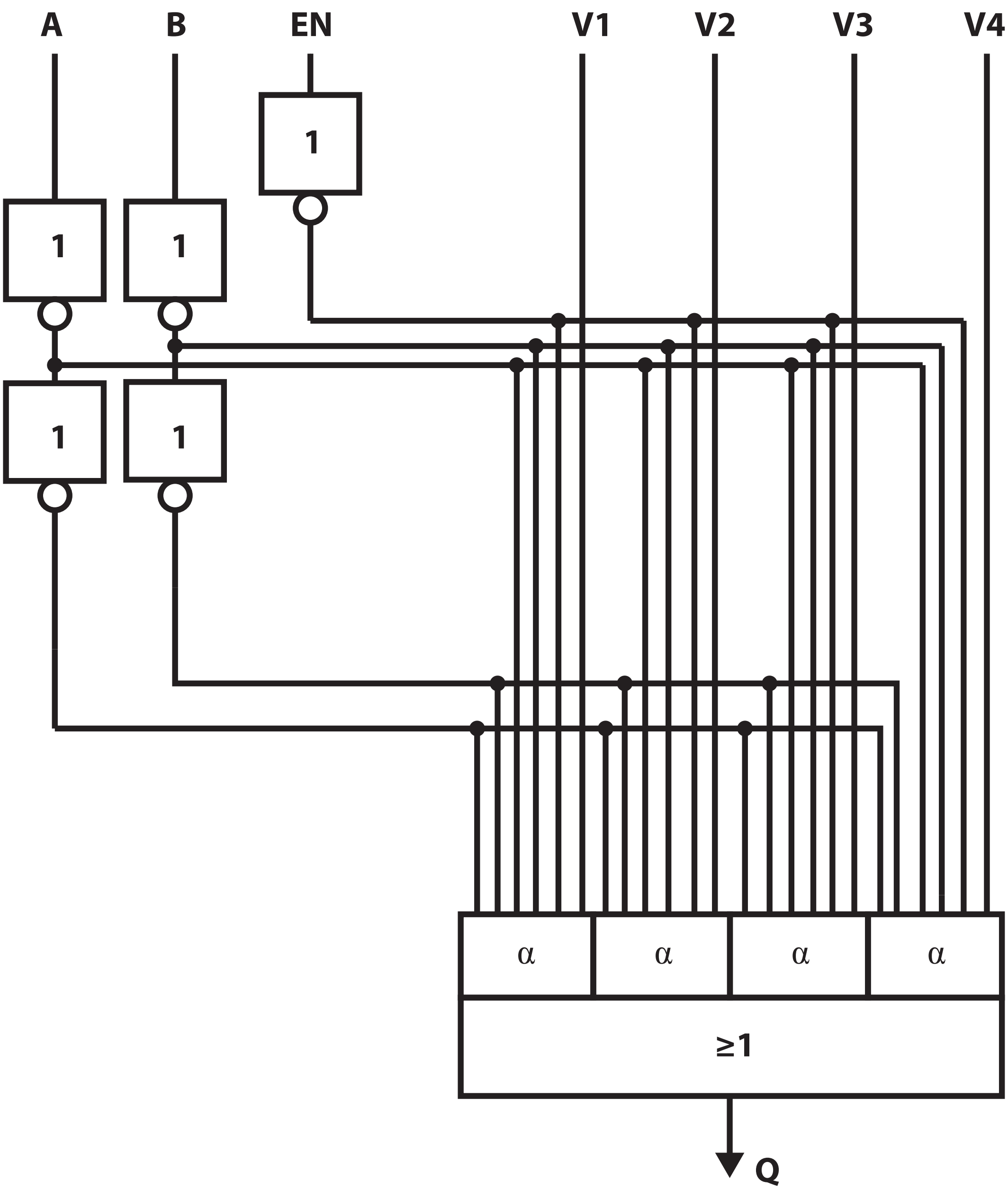
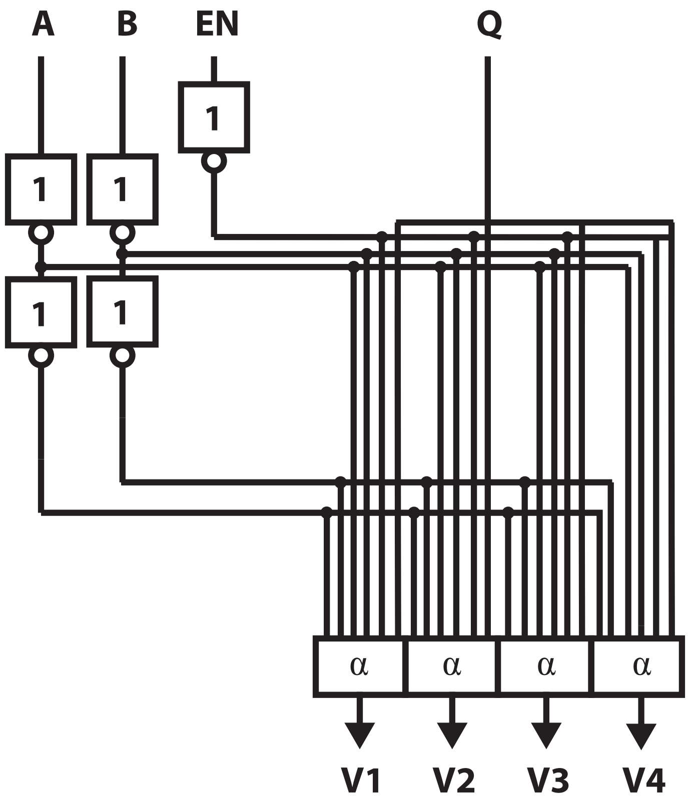


Fig. 4.6: Multiplexer



Pic 4.7: Demultiplexer

Figure 4.7 is plotted four-channel demultiplexer logic diagram. Q input is divided into outputs V1 to V4. Inputs A and B are the address inputs and EN is blocking input. The following describes the function of the demultiplexer logical table.

Table 6: Logical table

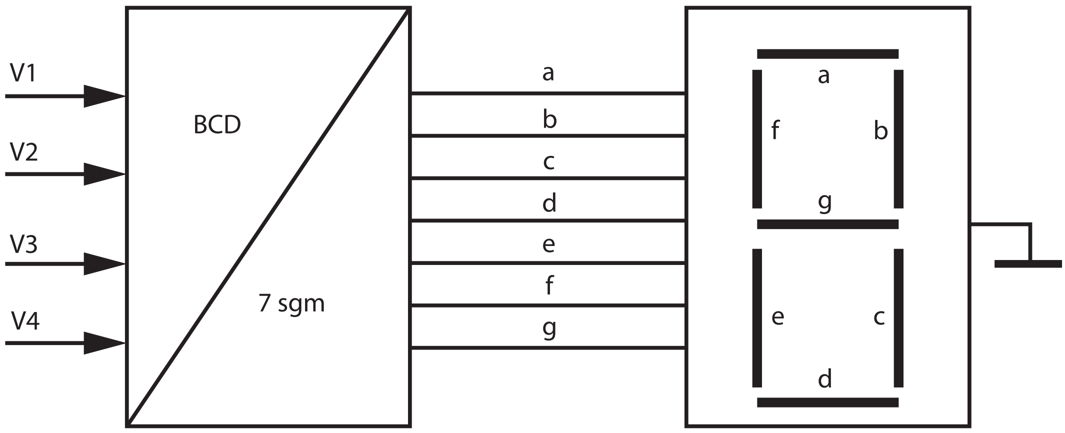
EN	B	A	výstup
0			0
1	0	0	V1
1	0	1	V2
1	1	0	V3
1	1	1	V4

4.13 Code converter

Logical systems work with binary quantities (1 and 0, H and L, true and false, etc.) there are many binary codes known as BCD code (Binary Coded Decimal).

Example 4.10

Prepare design for converter between binary code and the seven-segment display unit. Sketch input is shown below. Each segment is marked and lit closing an electrical circuit, i.e. by applying a voltage to the appropriate identically marked wire. Display unit draws 10 digits (0-9). This is ten combinations of output values. The number of output values depends on the number of inputs by Equation (3.1). For three inputs 8 outputs, this is less than required. For four inputs and outputs 16 it is more than the requirement. Consequently, it is necessary to have four input variables, which are identified in the 1 to V4.



Logical table for example is:

	V_4	V_3	V_2	V_1	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	0	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1
10	1	0	1	0	x	x	x	x	x	x	x
11	1	0	1	1	x	x	x	x	x	x	x
12	1	1	0	0	x	x	x	x	x	x	x
13	1	1	0	1	x	x	x	x	x	x	x
14	1	1	1	0	x	x	x	x	x	x	x
15	1	1	1	1	x	x	x	x	x	x	x

The first column is the number displayed. Inputs V 1 to V 4 are arranged as a conventional input variable logical table. The right side of this table is found the output values for individual segments. A value of 1 means on, 0 is off and will not affect the value of x solution is needed. In the next step, it is necessary to assemble Karnaugh maps for each segment.

For „a“ segment is :

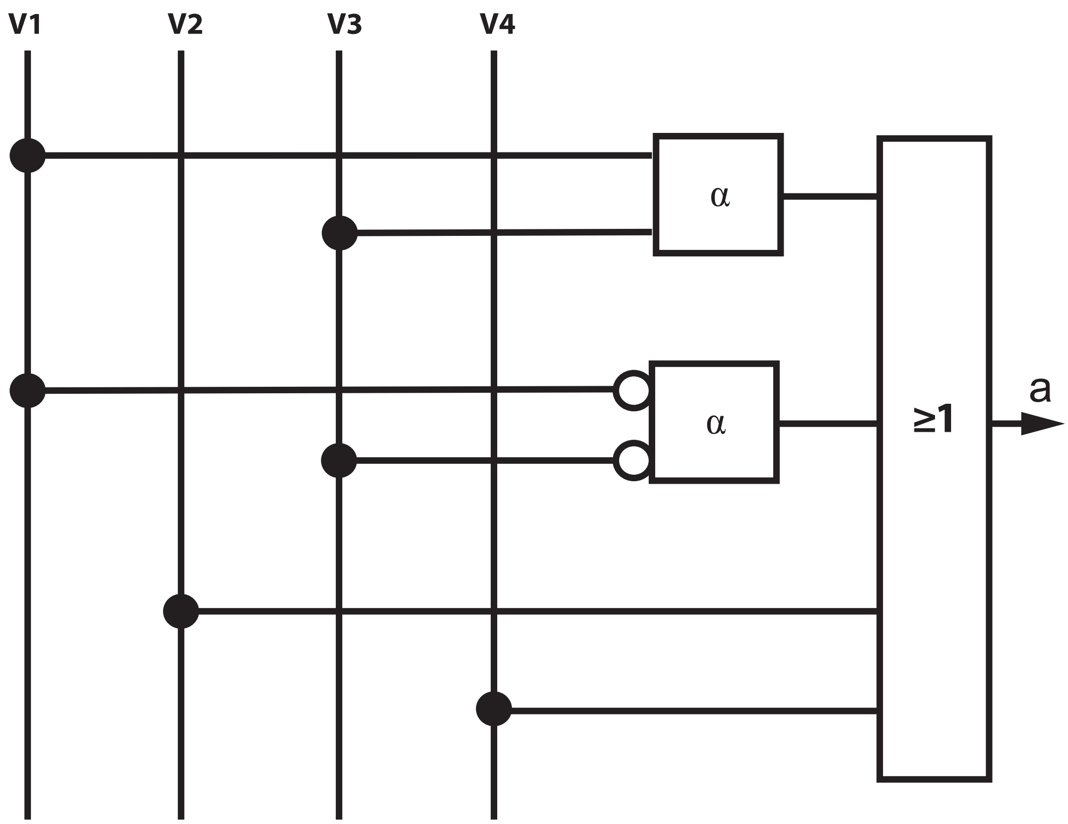
V3 V4 \ V1 V2	00	01	11	10
00	1	1	1	0
01	1	x	x	1
11	x	x	x	x
10	0	1	1	1

V1, V2, V3, V4

Expression with value x is equal to expression with value 1. Result function for segment is:

$$Y_a = V2+V4+V1\times V3+\overline{V1}\times\overline{V3}$$

For computed equation exists following scheme:



The same procedure will be created solutions for segments b to g. These blocks are connected to buses in 1 to V 4.

4.14 Safety circuits

These include electronic fire alarm circuits (EPS) circuits or electronic security system (ESS). These circuits operate on a logical principle at least two logical variable with a value of 1 in all variables.

5 Sequential logic functions, sequential nature of the behavior, feedback, sequential and temporal logic elements, synchronous and asynchronous execution

5.1 Sequential logic functions, Sequential logic circuits

Unlike combinational logic circuits depending on the state sequential logic circuit outputs only the combination of the input, but also on the internal state of the circuit, which contains an internal digital memory. Digital memory is realized by bistable flip-members.

Flip-members can be bistable or monostable. The bistable flip-member is two possible steady-state equilibrium states. In each member state of bistable flip-flop will remain until changed. To zoom is possible to imagine a two-position switch.

The opposite is monostable member who has only one equilibrium. After impulse, which monostable member deflects out of balance, the member returns to its original state. An example of this can be a key element.

Another way to flip the distribution of members is on asynchronous and synchronous. In this case, it is seen to change in terms of time. For asynchronous flip members change the output depends only on the change of inputs. In synchronous flip members change occurs not only on the change of inputs, but also passes through a synchronization (clock) pulse.

5.2 Binary memory

The need to preserve the value of signal is binary memory. Digital memory is schematically plotted in Figure 5.1. Inputs are identified by the letters "S" (as the English word set - set) and "R" (according reset - reset). Output values are complementary (opposite) and take values according to the logical table.

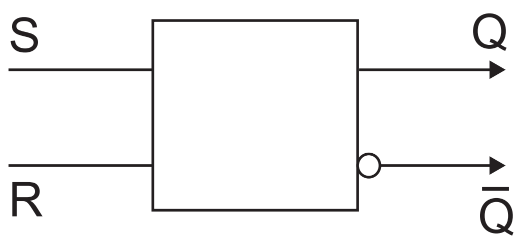


Fig. 5 .1: Pict. for binary memory

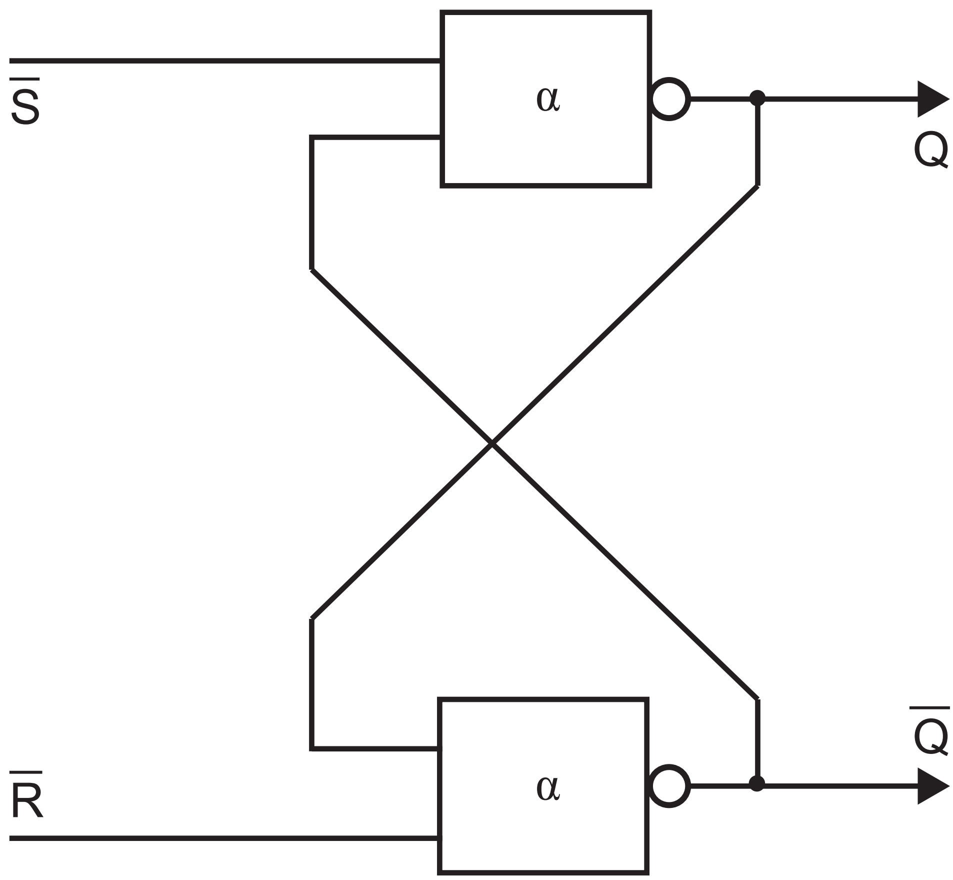
Table 7: Logical table for binary memory

S	R	Q	**$\bar{Q}$**
0	0	1)	1)
0	1	0	1
1	0	1	0
1	1	2)	2)
0→1	0→1	0 1 or 1 0	

1) the previous state

2) banned, indeterminate state; effect is such that when both inputs simultaneously change from 0 to 1, one of the outputs accidentally takes the value 0 and the second one, see the gray area of the table.

Logical function of binary memory can be implemented by connecting the two members of the "NAND" (negated logical product). The wiring is in Figure 5.2.



Pict. 5.2: Realization by using NAND

Technical realization of binary memory may be any according to the type of control variables, mechanical (Figure 5.4), electrical (Figure 5.3), hydraulic and electronic.

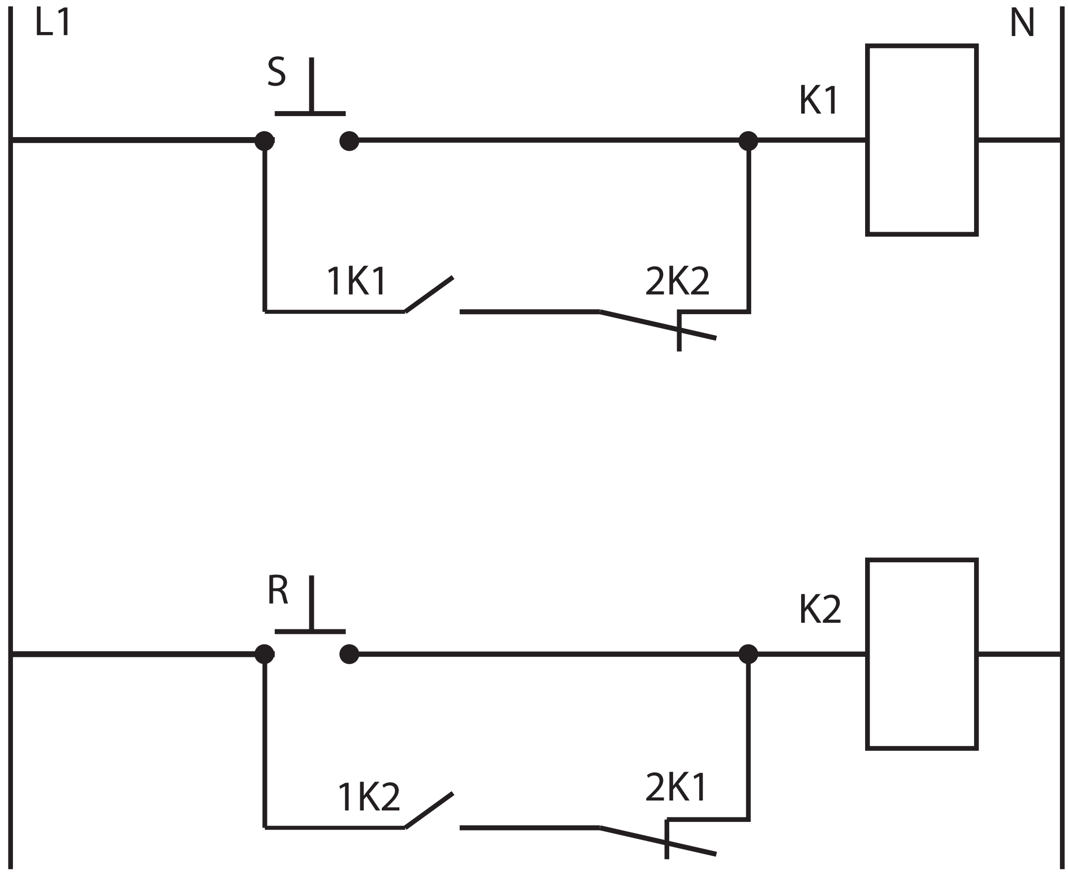


Fig. 5.3: Electrical realization for RS circuit

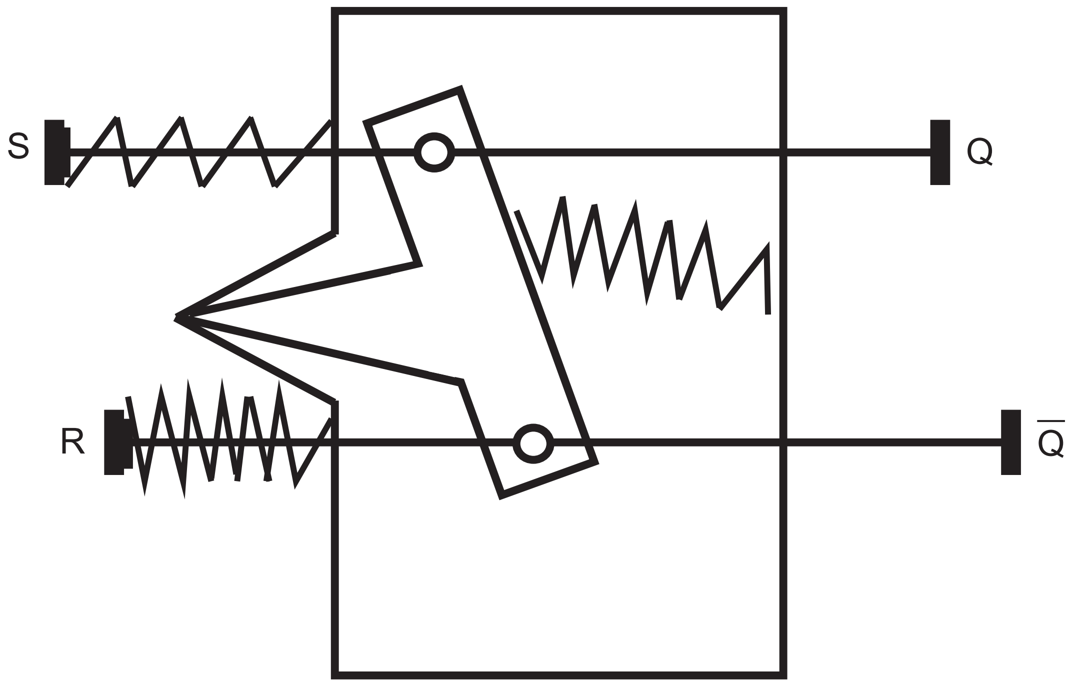


Fig. 5.4: Mechanical realization for RS circuit

5.3 Synchronous and asynchronous execution

Asynchronous flip-member

Asynchronous flip-flop is not controlled by any time signal and its status can be changed at any time by changing the input signal.

Basic asynchronous flip-flop circuit is a "RS" project gates "NAND" labeling scheme is shown in Figure 5.6 Logic is a wiring diagram in Figure 5.5.

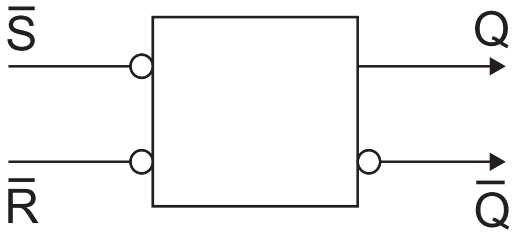


Fig. 5.5: Pictogram

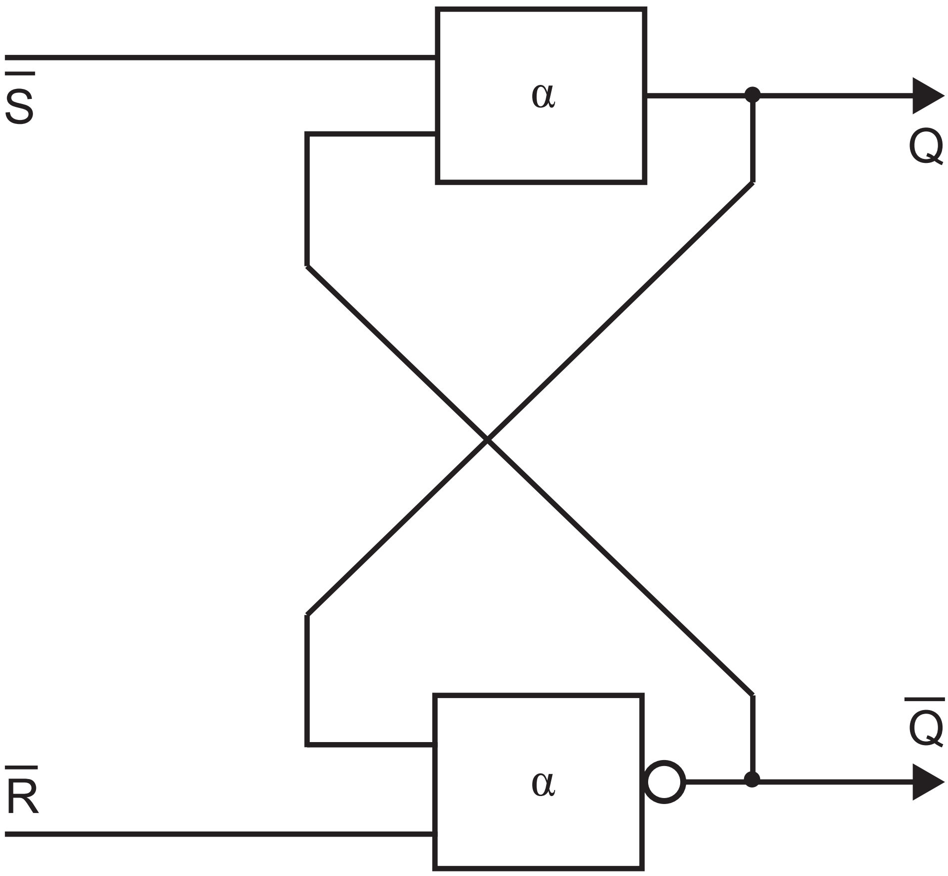


Fig. 5.6: Realization NAND gate

If the perimeter of the "RS" project gates "NOR" labeling scheme is shown in Figure 5.7. Logic is a wiring diagram in Figure 5.8. These circuits have one indeterminate state.

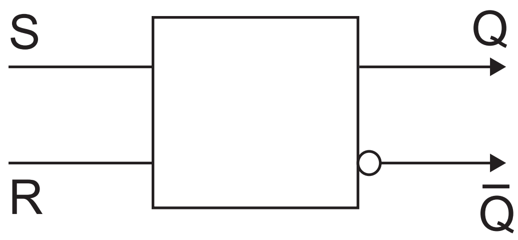


Fig. 5.7: Pictogram

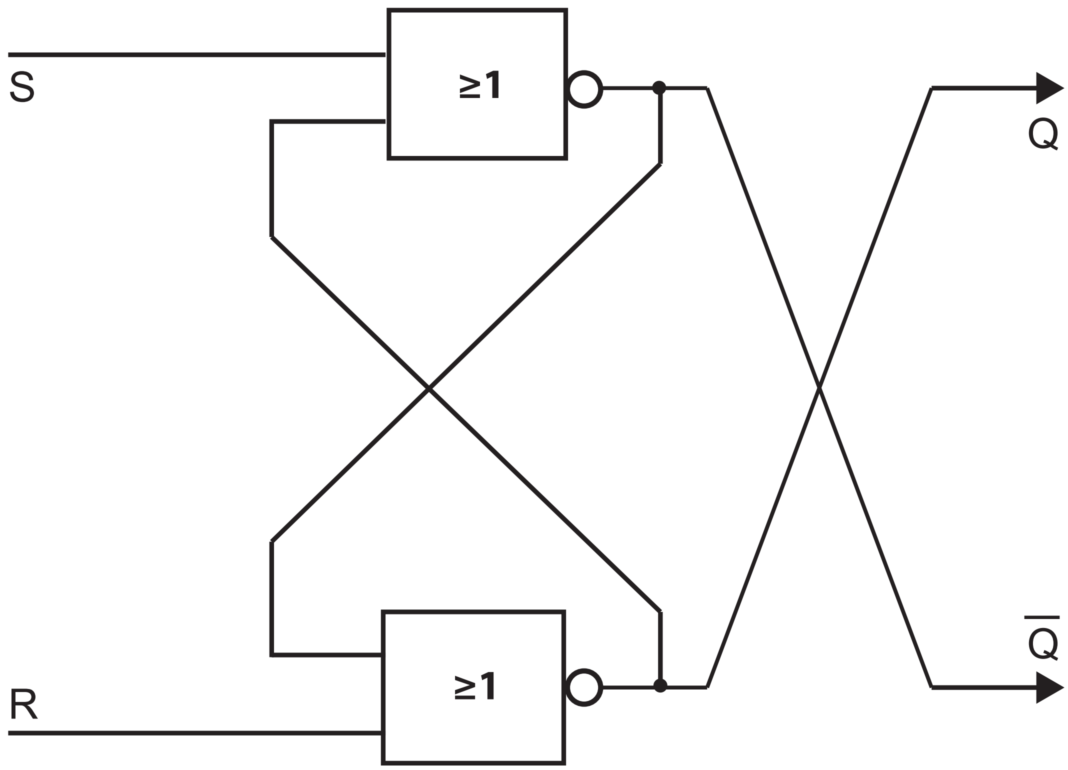
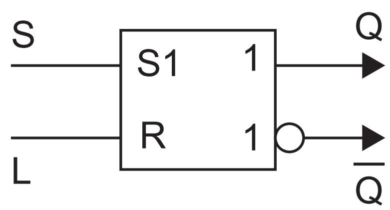


Fig. 5.8: Realization by NORs

Other flip-flops are circuits of type "SL" (set, latch - blocking) or "EL" (erase - delete, latch). These districts have banned or indeterminate states. Schematic symbol and logical connection SL-type flip-flop is shown in Figure 5.9. Schematic symbol and logical connection SL-type flip-flop is shown in Figure 5.10.



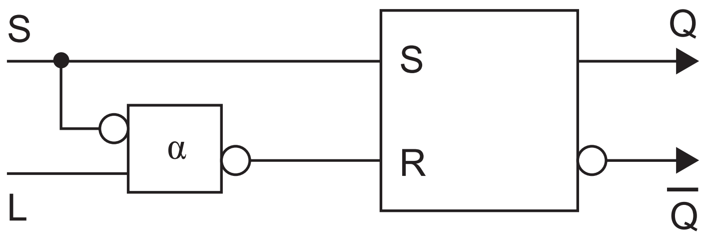


Fig. 5.9: Trigger circuit SL

Table 8: Logical table for trigger circuit SL

S	L	Q	$\bar{Q}$
0	0	PREVIOUS STATE	
0	1	0	1
1	0	1	0
1	1	1	0

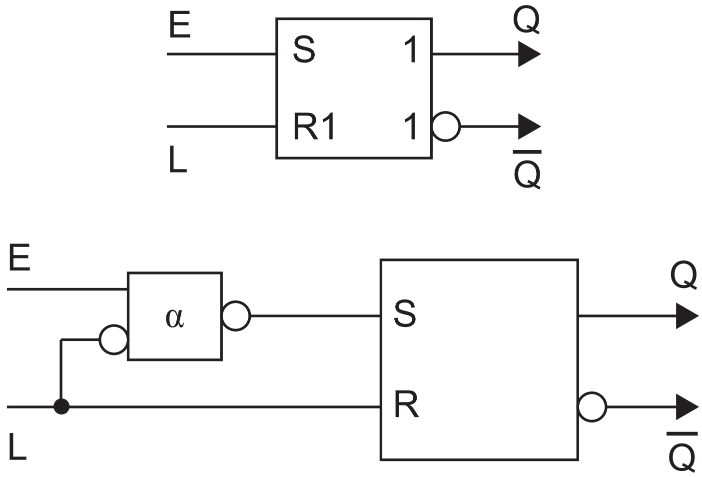


Fig. 5.10: Trigger circuit EL

Table 9: Logical table for trigger circuit EL

E	L	Q	$\bar{Q}$
0	0	PREVIOUS STATE	
0	1	0	1
1	0	1	0
1	1	0	1

Synchronous flip-member

In everyday life, it is possible to meet two kinds of sequence control. For example, the protection of man at the entrance to the working space robot and ensure safe entry into the road at a pedestrian light controlled crossing. In the first case there is an immediate cessation of movement of the robot. In the second case, after pressing the button, the first people to clear the road and then to indicate safe crossing for pedestrians. In the first case, it is necessary to use asynchronous flip-member, it means instant response to changing conditions. In the latter case, the use synchronous tilting member. It is a member that is activated in addition to changes in the input values have a time input. For time input is applied rectangular pulse. To change the output can then use rising or falling edge of the pulse.

Inputs to the synchronous flip-flop is three and labeled "J", "K" and "C" (clock - input clock pulse). The outputs are again coplanar (mutually opposite) labeled "Q" and "$\overline{Q}$". Schematic symbol and logical table corresponding to control rising edge of the clock input is given in Table 5.4. The time course of the output signal in response to changes in inputs in Figure 5.12 Figure 5.13. is plotted logic controlled synchronous flip-member rising edge.

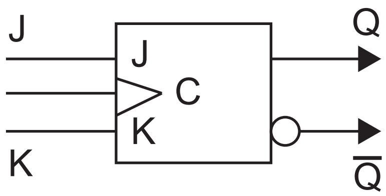


Fig. 5.11: Trigger circuit JK, output signal in time

Table 10: Logical table for trigger circuit JK

C	J	K	Q	**$\overline{Q}$**
	0	0	PREVIOUS STATE	
	0	1	0	1
	1	0	1	0
	1	1	NEG. PREVIOUS	

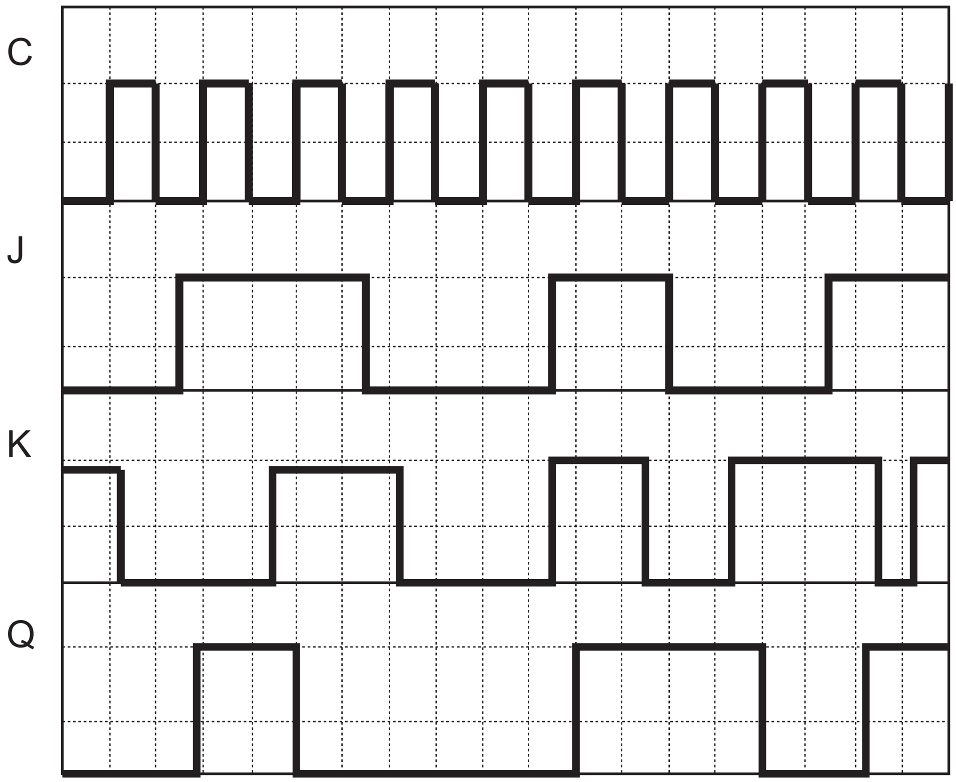


Fig. 5.12: Output signal in time

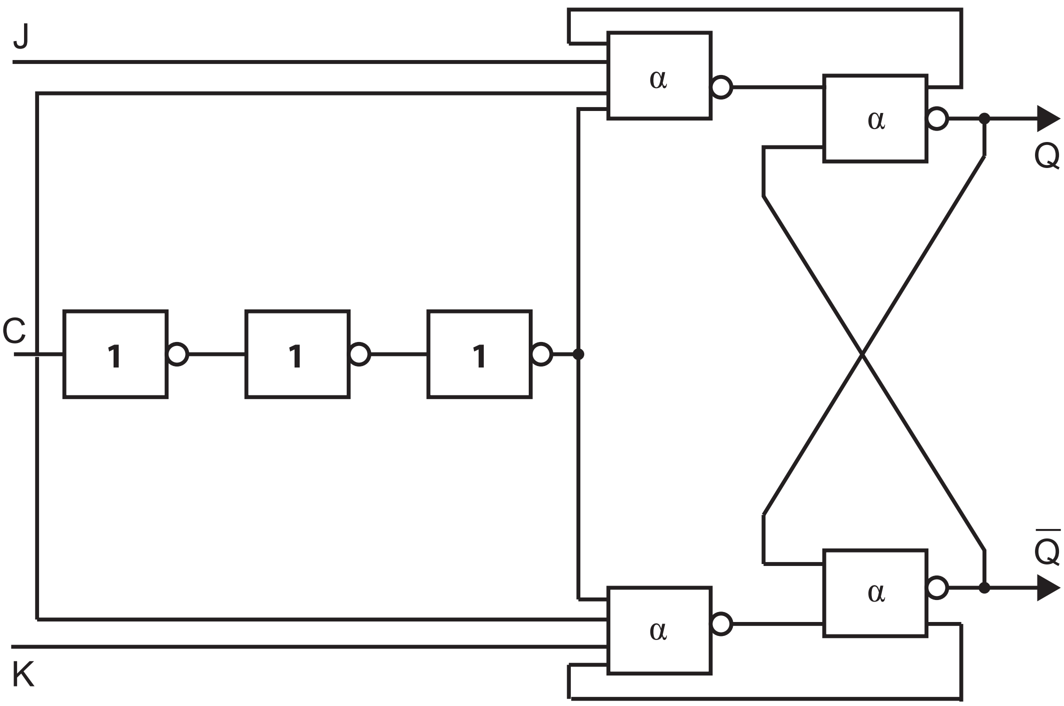


Fig. 5.13: Logical scheme for JK circuit

5.4 Repeating questions

1. Describe the function and application of sequential logic circuits.
2. Explain the concept of binary memory.
3. Explain the asynchronous trigger.
4. Explain the function of synchronous trigger.
5. Describe an example of using synchronous trigger.

6 Minimum about fuzzy logic

6.1 Introduction

Fuzzy logic is a part of artificial intelligence. Her mission is inflicts natural indeterminateness around us and indeterminateness of human thinking and locution. That way is possible to make accessible to realization by usual facilities for described phenomena and make possible their usage in technical practice.

There exist many fuzzy logic applications in various branches – technical and also untechnical from regulation, technical diagnostic to geology, linguistics, biology, social sciences etc.

We don't say „yes, yes, no, no“

„Let your language be yes, yes - no, no!", such should be the expressing style of a good man according to Rudyard Kipling. Usually we expect to hear such unambiguous language from statesmen and politicians; it is however only rarely encountered at our place (at most with populist promises). Our way of thinking is influenced by the Greek philosophy (particularly by the teaching of Aristotle and his logic) far more than we realize. Logical thinking facilitated incredible boom of science and technology since the Middle Ages up until now. All computers, control and communication systems, and other digital devices operate on the basis of two-valued logic.

The situation is, however, more complex. It is indeed fixed in our rational thinking that clear questions should get clear answers – either yes, or no. But our actual language and the way of thinking significantly differ from strictly unambiguous rules of mathematical logic. For example, try to analyze what you are saying in the statement nothing is impossible. Far too often do we use "fuzzy" and "uncertain" words about, I'm not sure, maybe, approximately, likely, probably, supposedly, presumably, in my opinion, more or less, almost, rather not, basically yes, at 70% yes, sometimes even the quaint almost exactly. Sometimes it is indolence in thinking and poor culture of expressing oneself. It might be even deliberate effort to "conceal" the fact, to avoid telling the whole truth and to complicate its understanding – this is particularly remarkable in the speech of lawyers, diplomats, politicians and their press secretaries.

False declaration subjective evaluation

In our vision, interpretation of the percepts, and subsequent assessment, we unknowingly use our own "filters and templates" which are individual and depend on the past experience and bearings. Consequently we also judge something uncommon with comments such as "my eyes are playing tricks", or "I must be dreaming" – but it is frequently only our misconception or misinterpretation of the sighted experience without deliberate comment. If our knowledge and opinions are biased with uncertainty, it is natural and correct to communicate them with "rating" of such uncertainty.

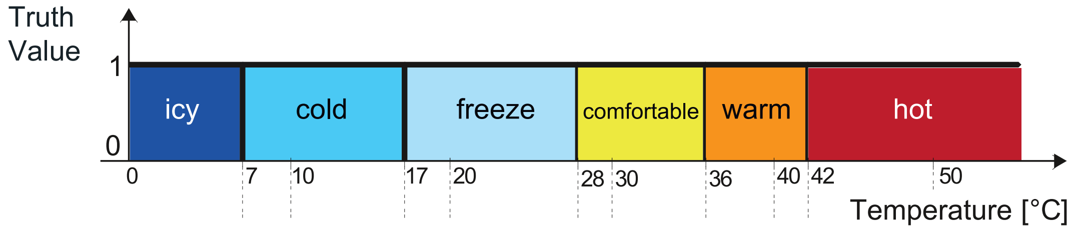


Fig. 6.1: Sharp distinction.

Another problem occurs when trying to assess exact and explicit data. The picture illustrates an attempt to distinguish data about water temperature in a subjective view of a person intending to take a bath. The boundary between zones assessed as pleasant and warm is shown here at 36 °C. Such sharp transition seems to be natural. We could possibly discuss the values for the individual zone boundaries because everyone judges temperature and its "congeniality" differently.

Fuzzy logic – mathematical branch

Consequently, an apparatus of fuzzy sets and fuzzy logic was created. The author is Lotfi Zadeh, who published his first article thereof in the fifties of the twentieth century. It raised hot interest, developed quickly and "cultivated" in a great thematic extent, especially in its application fields. There are comprehensive monographs and textbooks written on this topic. The general explanation is too complicated for the fuzzy logic to become a routine apparatus for the PLC programmers, who could use it and its advantages in their practice.

Logic is a science field concerned with the mechanisms of human thought and reasoning. Its roots stretch back to philosophers of the Ancient Greece. In this regard, the most popular is Aristotle who is considered the founder of two-valued logic and resulting way of thought, which has influenced the evolution of our entire Western (Euro-American) civilization. Logic became the basis of mathematics and mathematical thought in other technical fields; mathematical logic became independent (and relatively sophisticated) branch of mathematics. One part of mathematical logic is Boolean algebra, which has found its use in technical application.

The traditional logic is two-valued and deals with statements (logical variables) that have two values: true – false, logical 1 – logical 0. Composite statements (logical functions) are constructed by linking fundamental logical statements via logical conjunctions (e.g. and, or, either – or, neither – nor, it is not true that). Using certain rules of mathematical logic, it is possible to determine the value of a composite statement based on the truth or falsity (value) of the sub-statements.

6.2 Sources of fuzzy logic

Number sets

Not always have to be a typical cases of sets, e.g. the set of apples and pears. In technical applications, we're often front of sets that are relevant intervals on the axis. In the above examples, we can for example find several sets (intervals) to evaluate water temperature, bearing temperature or intensity of its vibrations. If the value of the input linguistic variables (eg temperature) belongs to one of the sets (eg increased), it is also true the phrase "temperature is increased." It is easier to talk about sets and membership in them, but directly to the veracity and truthfulness of the functions of input statements - entry terms.

Multi-valued logic and fuzzy logic

A natural generalization of two-valued logic represents the three-valued logic with values, for example, 0 (meaning false), 0.5 (partially true, maybe, unknown), and 1 (true); there exist also other logics with greater number of truth levels. Logical variable in fuzzy logic takes an infinite number of values from the closed interval [0, 1]; the number of values is limited during program implementation and depends on the method of numeric interpretation of the truth value.

Fuzzy sets

In fuzzy set theory, each element is assigned a degree of its membership in the fuzzy set (membership function) valued in the closed interval [0, 1]. This function is generally designated by the symbol μ, next to which the name of the set is written in subscript; for example μA represents membership of the element in fuzzy set A, μB represents membership of the element in fuzzy set B, μA represents membership in fuzzy set A, μincreased represents membership of the element in fuzzy set increased, etc.

Belonging to the fuzzy set

It is common that an element of fuzzy set "partially belongs to the set and partially does not belong to the set" (with membership between 0 and 1). The membership of the element in the set may be regarded as fuzzy. Boundary of a fuzzy set is fuzzy as well – meaning vague, hazy blurred. This is also the origin of the word fuzzy. As contrasted to classical sets, it is possible (and common) for fuzzy sets that one element belongs to two or more fuzzy sets with different membership degree at the same time. Thus, it is possible in fuzzy logic to peacefully reconcile conflicts such as "either I am right, or you are" by saying "we both are partially right". Similarly to classical sets, the system of set operations is defined also for fuzzy sets: among the fundamental ones are operations of fuzzy intersection, union, and complement, but there are other fuzzy set operations as well. Accordingly there exists a tight relationship between set operations and logical operations.

Composite systems and fuzzy logic

In technical applications of fuzzy systems we often (nearly always) encounter mixed systems, which have input variables in the form of numeric variables (language variables), and logical variables (input language terms) are defined above those variables. While in the two-valued logic transitions of the logical functions of the adjacent terms are sharp (steps), in fuzzy logic they can be gradual and the logical functions can overlap. For example, water temperature of 35 °C can be assessed as partially pleasant and partially already hot; similarly, water temperature of 37 °C can be assessed as partially hot and still partially pleasant.

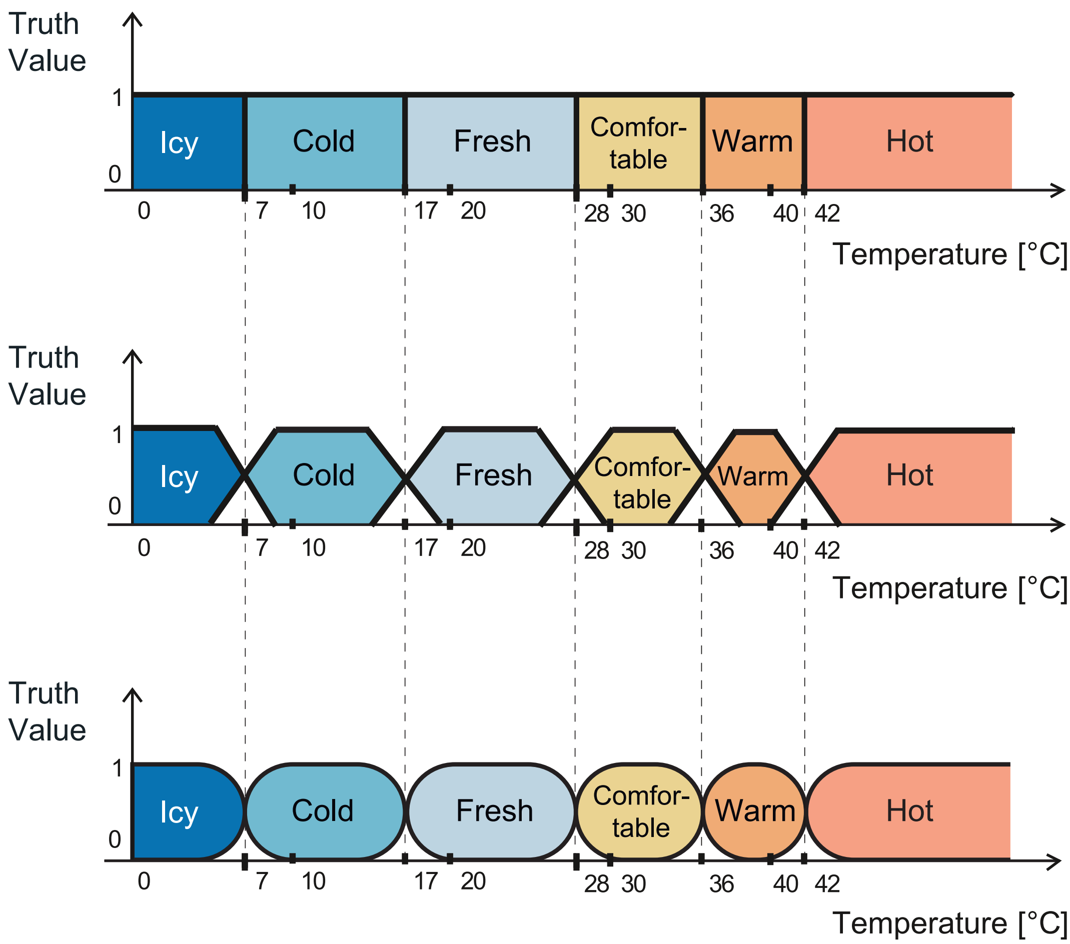


Fig. 6.2: Possible behavior of logical functions of the terms

Fuzzy system for bearing diagnostics

For example, when solving the problem of bearing diagnostics we can generalize the process described in the end of Chapter 5. Instead of two-valued terms we will now work with fuzzy terms that are fuzzy variables and take values from the interval [0, 1]. The logical functions for vibration intensity and temperature can have a shape of fractional function (trapezoids and ramps) and are overlapping for the adjacent terms.

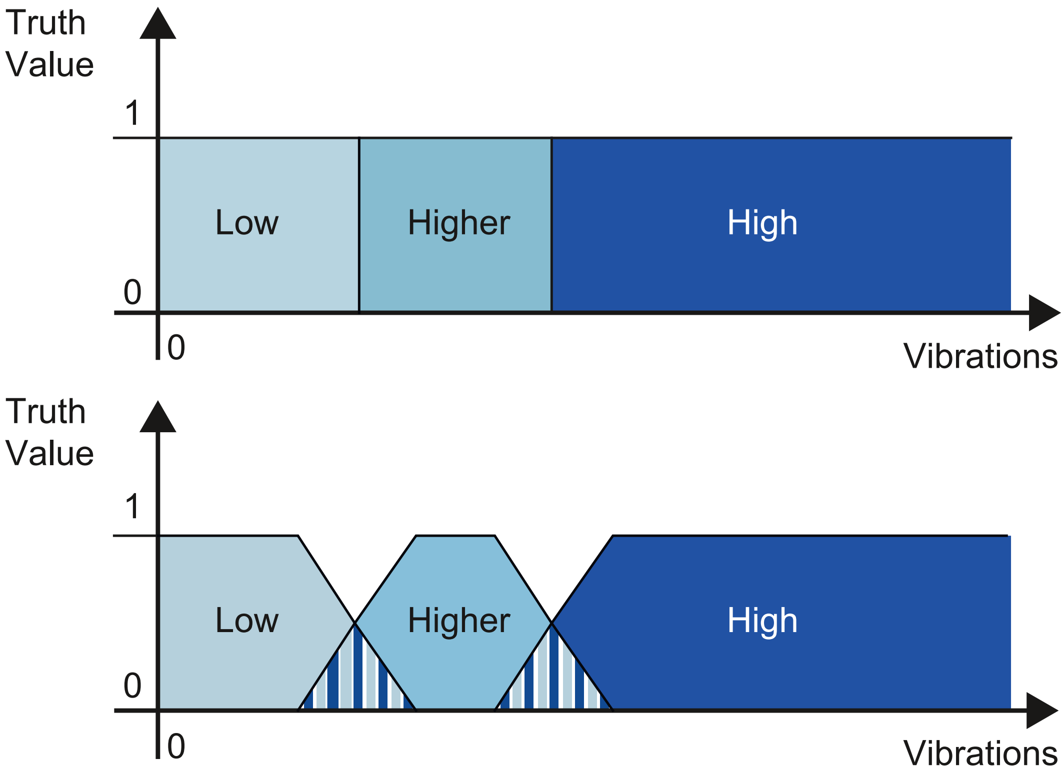


Fig. 6.3: Possible behaviour of logical functions of three terms.

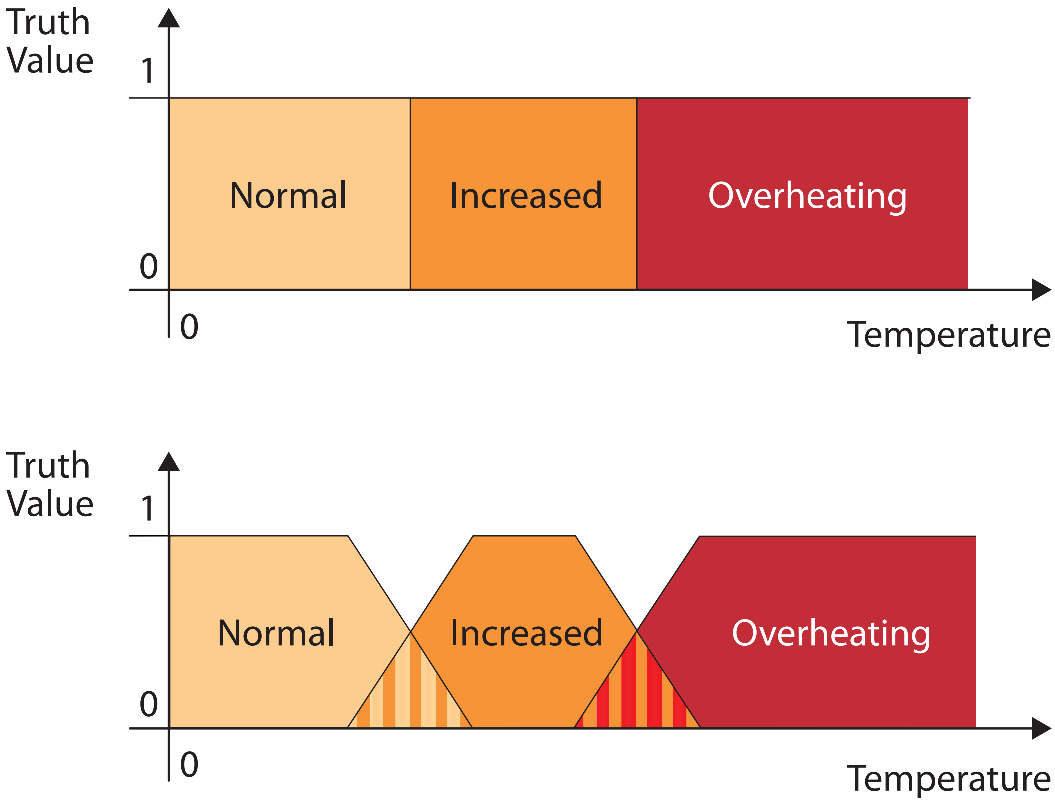


Fig. 6.4: Possible behaviour of logical functions of three terms.

Output fuzzy terms

The result of the evaluation is usually a group of fuzzy variables – output terms, for example with the meaning of OK, warning1, warning2, alarm, malfunction. However, we can desire the result as a value of a single, continuous (numeric) function – the output language function having the meaning of diagnosis of defective bearing. Its value can be determined from the logical values of the output terms. The logical functions of the said terms can have a shape of fractional function (trapezoids or triangles), possibly rectangles, or narrow pulses (singletons).

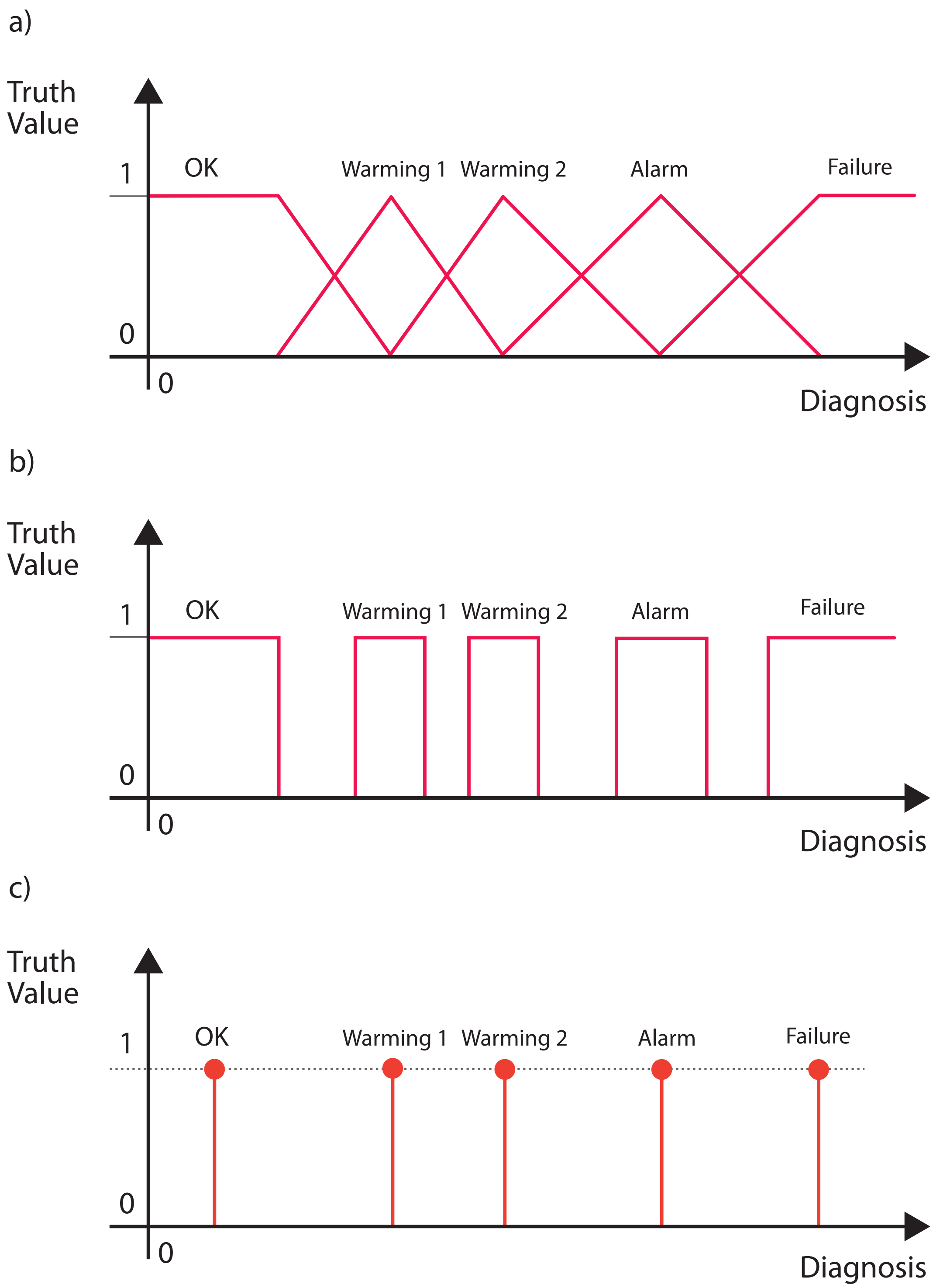


Fig. 6.5: Possible behaviour of logical functions of five terms.

6.3 Fuzzy logic as generalization of binary logic

Introduction

Substance of the simplification whose we are making in this module consists in consideration fuzzy logic for multivalued variables as binary logic generalization. We will illustrate the progress on the example of symmetric logical functions which were described for binary logic earlier.

6.4 Treshold and majority function in binary logic

In the schoolbooks of logical systems design, majority and threshold functions are being used as illustrative examples for showing the process of design and minimization of logical expressions. Their logical value depends on the number of true operands and it remains the same for the given number for all combinations of operands - consequently these are ranked among symmetric functions. Threshold functions ("at least k from n" – denoted as f_k_n) are true if at least k (k or more) of the n operands equals to one. The number k is called threshold. Majority functions are the special case of threshold functions. They are defined for odd number of operands and they are true if more than a half (majority) of the operands are true.

6.5 Logical terms

For the “3-minority” functions, it holds m3 = f_2_3; for the “5-majority” function, it holds m5 = f_3_5. It generally holds that the minimized logical expression for the k threshold contains the sum of product members with the length of k, in which all combinations of the n variables in the fundamental condition (no negations) are covered in a sequence. Thus, it holds for example:

f_3_3(a, b, c) = abc

f_2_3(a, b, c) = m3 (a, b, c) = ab + ac + bc

f_1_3(a, b, c) = a + b + c

f_4_4(a, b, c, d) = abcd

f_3_4(a, b, c, d) = abc + abd + acd + bcd

f_2_4(a, b, c, d) = ab + ac + ad + bc + bd + cd

f_1_4(a, b, c, d) = a + b + c + d

6.6 Connection with AND and OR

It can be seen that functions f_3_3(a, b, c) = abc and f_4_4(a, b, c, d) = abcd are equal to the logical product AND (it generally holds for f_n_n) and functions f_1_3(a, b, c) = a + b + c and f_1_4(a, b, c, d) = a + b + c + d are the logical sum OR (it generally holds for f_1_n).

6.7 Using in safety technique

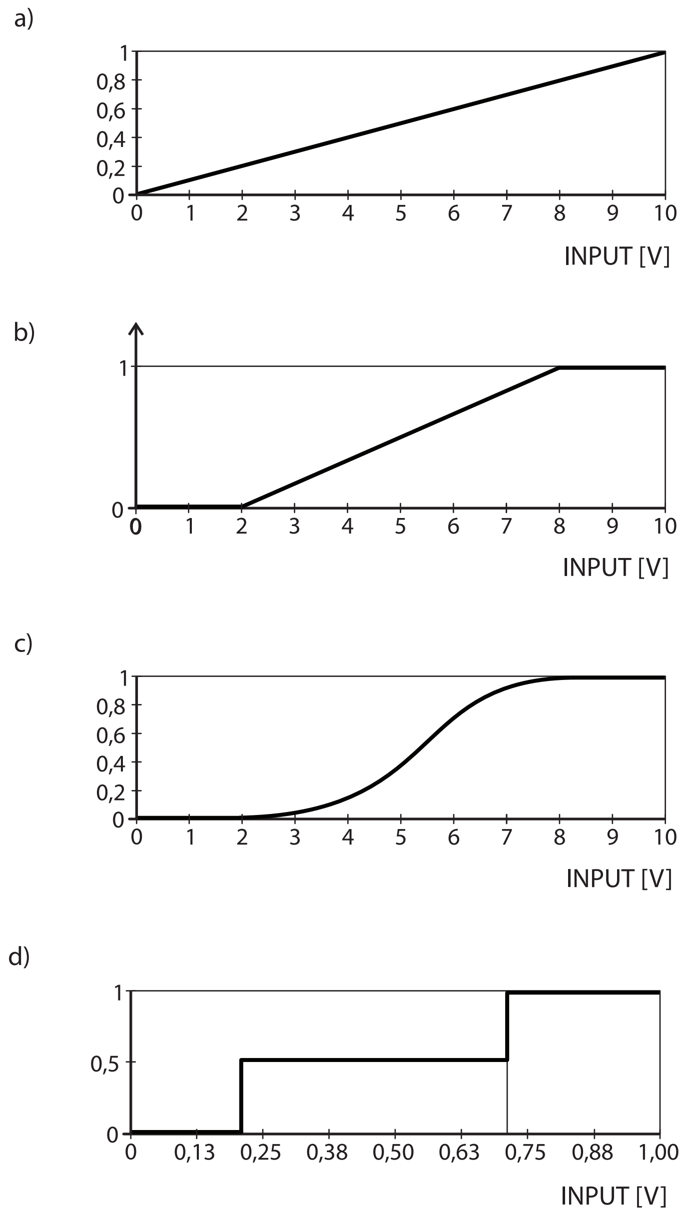
Threshold and majority functions are often applied in solutions of fire alarm systems when fire sensors (fire detectors) with binary output are being used. Those are used in redundant number and the resulting decisions about fire risk are implemented as threshold functions. Thus, the sensors are backed up to some extent and the solution is resistant to malfunction of any sensor. Occurrence of false alarms upon abolition or disoperation of any of the sensors is also reduced.

6.8 Reasons for fuzzy generalization

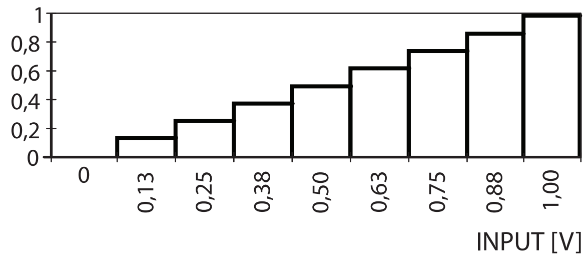
Foregoing examples of implementation of the majority function and threshold functions were introduced. Now suppose the goal is to secure the object in a similar way, but with the utilization of sensors, which provide continuous output, for example in the range from 0 V to 10 V, so that the zero value corresponds to a safe condition (risk of fire is zero) and the 10 V value corresponds to absolute certainty of fire (risk of fire is one).

6.9 Fuzzy generalization procedure

There is a possible solution by employing fuzzy logic. Binary logical variables will be replaced by fuzzy variables, which can take any values between zero and one, and the truth thereof represents the degree of fire risk. Furthermore, there will be introduced such logical operations that generalize the fundamental logical operators AND, OR, and NOT. Each sensor corresponds to one fuzzy variable. The assignment of logical value to the value of voltage signal received from the sensor (the simplest case of fuzzification) depends on our judgment – it could be a mere linear normalization of the range, or a linear dependence in the active area with limitation on both sides of the range, or a general non-linear dependence.



e)



f)

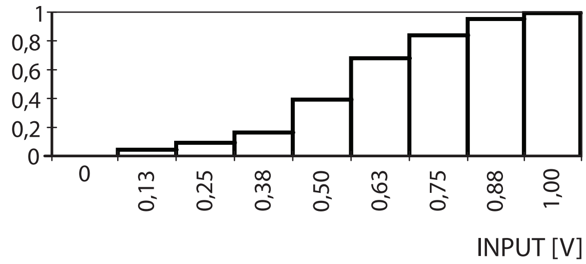


Fig. 6.6: Possible behaviors of the logical function for continuous fire sensor

6.10 Generalization for logical terms

For evaluation of alarm levels we can utilize the same logical expressions as in Chapter 5, just generalized for fuzzy logic. These are formally identical with only the operators of Boolean algebra being replaced by generalized operators for fuzzy logic.

6.11 Fuzzy generalization of AND, OR, NOT

There is infinite number of ways for generalization of the Boolean operators AND, OR, and NOT for multi-valued logic and fuzzy-logic (fuzzy generalization is suitable for multi-valued logics as well). Concerning combinations of the boundary values in the intervals (Boolean values 0 a 1), all generalized operators behave as Boolean operators AND, OR, and NOT. The logical values within the intervals [0, 1] are assigned “some” logical values again from within the interval [0, 1]. Simply said, the generalized operators perform interpolation between the Boolean values of 0 and 1. It may be discussed what operators are more suitable for the particular purpose and what provide better (smoother or less complicated) interpolation process. None of the operators described herein are unsuitable (so they can be suitable). Sometimes other symbols are used (e.g. ANF, ORF, and NOF) for distinguishing them from Boolean operators; however, traditional notation, which is used also in the standard IEC 61131-7 will be applied here and the type of the operator will be specified by an index.

6.12 Fuzzy generalization of negation – fuzzy NOT

As a generalization of fuzzy negation, the one's complement is most commonly used (in applications almost exclusively)

NOT a = 1 – a

Other works mention, for example, Sugeno class of complements

NOTλ a = (1 - a)/(1 - λa) for the parameter λ in the interval (-1, ∞)

or Yager class of negations

NOTw a = (1 - aw)1/w

6.13 Fuzzy generalization AND and OR

As a generalization of logical product AND and logical sum OR, the most commonly used are operations of minimum and maximum (Zadeh's or Gödel's operations).

$AND_m(a, b) = \min(a, b)$

$OR_m(a, b) = \max(a, b)$,

$AND_a(a, b) = ab$

$OR_a(a, b) = a + b - ab$

$AND_b(a, b) = \max(0, a + b - 1)$ ... the expression $a + b - 1$ bounded from below by 0

$OR_b(a, b) = \min(a + b, 1)$ ... the expression $a + b$ bounded from above by 1.

Behavior of the functions of generalized operators for two variables is generally visualized as a three-dimensional diagram. It can be understood as generalized logical maps (K-maps) depicted in space, where corners of the floor square represent the values assigned to the boundary (Boolean) values of operands 0 and 1 by the traditional Boolean operators.

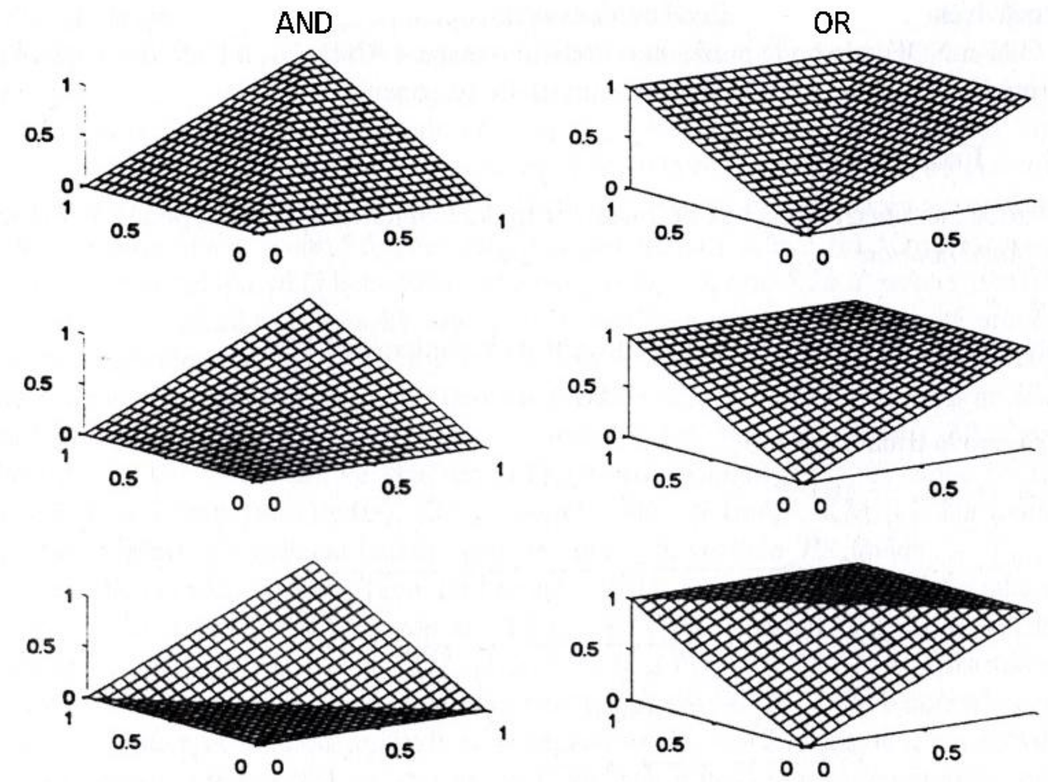


Fig. 6.7: Logical graphs of two operators

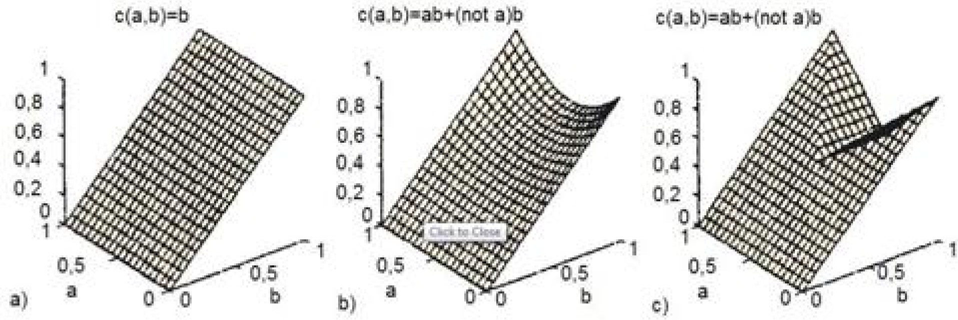


Fig. 6.8: Demonstration of dissimilarities between minimized term function

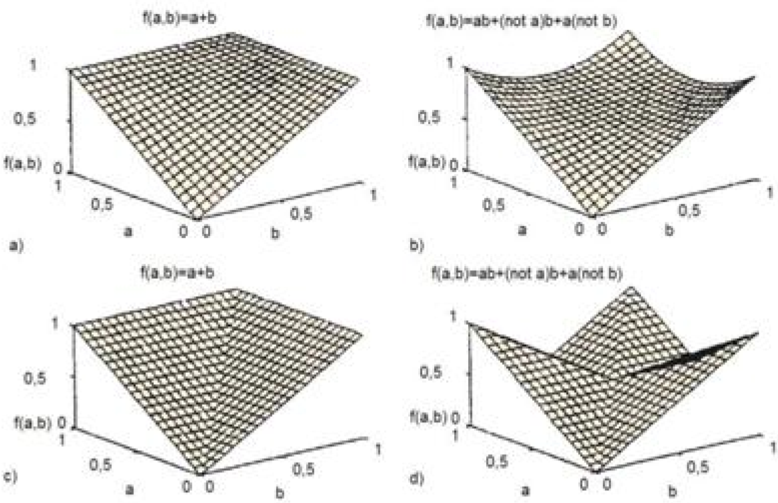


Fig. 6.9: Demonstration of dissimilarities between minimized term function a

Example 6.1

Let us give and illustrative example with logical values of the operands $a = 0.3$ and $b = 0.8$. It holds for the operations of maximum and minimum:

a AND$_m$ $b = 0.3$ (the lesser of the two), a OR$_m$ $b = 0.8$ (the greater of the two)

for probability operations:

a AND$_a$ $b = 0.3 * 0.8 = 0.24$ a OR$_a$ $b = 0.3 + 0.8 – 0.24 = 0.86$

and for Łukasiewicz's operations:

a AND$_b$ b = max (0; 0.3 + 0.8 – 1) = the greater of (0 and 0.1) = 0.1

a OR$_b$ b = min (0.3 + 0.8; 1) = the lesser of (1.1 and 1) = 1

It can be seen that the following inequalities hold for the results of generalized logical operators (not only for the abovementioned examples, but generally):

$\text{AND}_b < \text{AND}_a < \text{AND}_m$

$\text{OR}_m < \text{OR}_a < \text{OR}_b$

6.14 Satisfy minimum a maximum

The most frequently used are presumably the operations of minimum and maximum. Their computation is not demanding and they have additional advantages. The most of the rules of Boolean algebra hold for them (all of the rules but the rule of the excluded third) and usually they confirm with our intuitive understanding of logical issues.

6.15 Fuzzy diagnostic system - example

Suppose that it is required to implement the program for diagnosis of bearings by evaluating measured data about temperature and vibrations. Consider the same task and the same rules (logical expressions) as the example for binary logic, just with generalization for fuzzy logic.

6.16 Determination of input terms verity - fuzzification

We will consider the logical functions of the input terms for vibration intensity and for temperature have the form of fractional function (trapezoids and ramps). Determination of logical values of the input terms based on the values of vibration intensity and temperature is called fuzzification.

6.17 Verity numeration - inference

For the example, we have again chosen five diagnostic levels, each of which is represented by a separate fuzzy logical variable – output language terms: *OK*, *warning1*, *warning2*, *alarm*, *malfunction*. We can evaluate their logical values using a group of logical expressions (assignment commands) generalized for fuzzy logic. In order to use conditional commands IF – THEN, it would be necessary to deal with generalization of implication function and deriving rules, so these are not used herein.

OK := v_low & normal;

warning1 := v_low & increased OR v_increased & normal;

warning2 := v_low & overheating OR v_increased & increased OR v_high & normal;

alarm := v_increased & overheating OR v_high & increased;

malfunction := v_high & overheating;

We can arrange the rules in a matrix form (map) for better clarity.

Temperature \ Vibrations	Low	Higher	High
Normal	OK	Warming 1	Warming 2
Increased	Warming 1	Warming 2	Alarm
Overheating	Warming 2	Alarm	Failure

Fig. 6.10: Matrix arrangement of the rules.

6.18 Single valued result numeration - defuzzification

Upon evaluation of the expressions, a situation may occur when several output terms (up to four) are nonzero at the same time, so the result is rather confusing and multivalued. Therefore, it is practical to convert the combination of the term values to a single variable comprising the overall information about the resulting diagnosis, for example with the meaning risk_of_malfunction or diagnosis. This operation is an opposite problem to fuzzification, thus it is called defuzzification. The five output terms can be thereby interpreted as a group of logical functions above the output language variable diagnosis.

6.19 Deffuzification

By set of rules enumeration we generally obtain several output logical variables (output terms) whose verity is nonzero. Each logical conclusion has different verity as is usual in fuzzy logic, but the same situation is in conclusions from several experts whose advised the same problem. With a set of multivalent (sometimes also with contradictory) results is difficult to work. That is the reason why is necessary to obtain unique result – one concrete value for each output language variables, e.g. "diagnosis" about bearign situation ("risk of retention by bearing is 38") or value of action hit by fuzzy regulator ("turn on the lid to 27%", "voltage for gearing = 5,6 V", "valve slightly open by impulse in duration 3,8 s"). Procedure how to make unique (sharp) result from the set of multivalent fuzzy logical conclusions is called defuzzification.

6.20 Output terms

Defuzzification can be also considers to be the opposite of fuzzification. We define course of logical functions for individual terms above the output language variables (e.g. in the meaning of bearing failure risk, action hit voltage, regulation gage or valve displacement, time of action hit duration). These terms could have similar shape and dislocation as with input terms definition – usual triangle, trapeze or curves with smooth course.

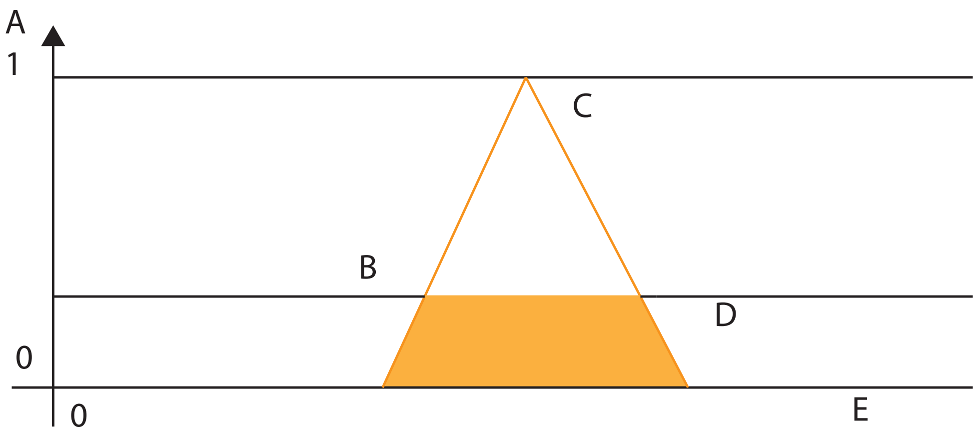
6.21 Defuzzification process without implication

In literature is defuzzification process usual identify with implication functions (some of their many possibilities of fuzzy generalization). Interpretation is complicated and in practice are realized just roughly simplified variants. We will describe in this text just using process without theoretical reasoning.

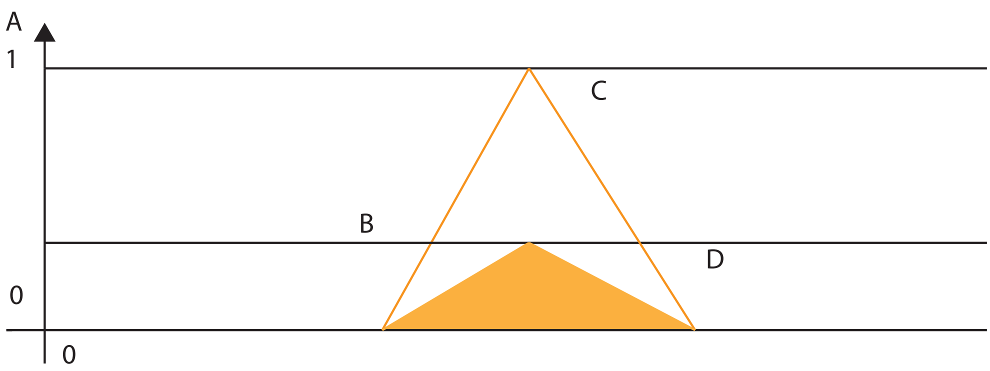
6.22 OR intermediate data – united area

We will make described operation for each term (for terms with non-zero actual verity will be enough) and sub results for each term logically add up. Logical sum OR is interpretate as maximum function. This process is possible to interpretate graphically as figure unification under „deformed“ term logical functions („cropping“ or „compressed“).

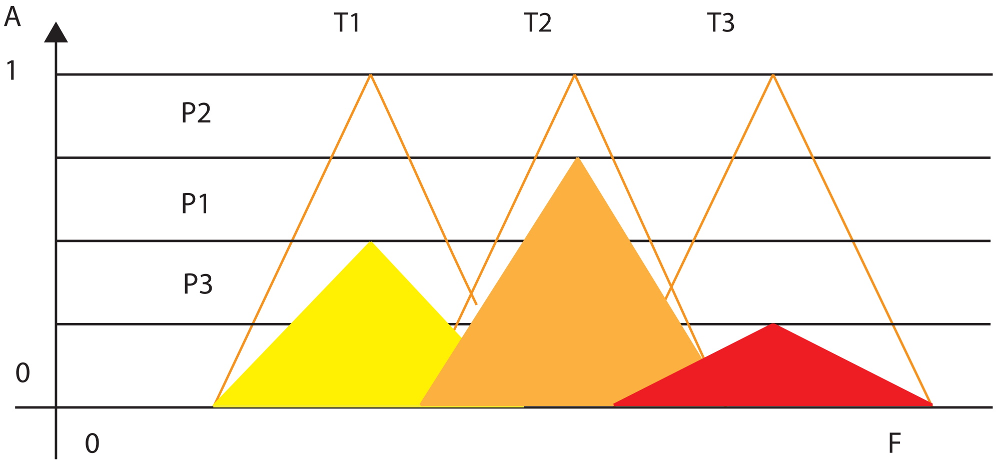
a)



b)



c)



d)

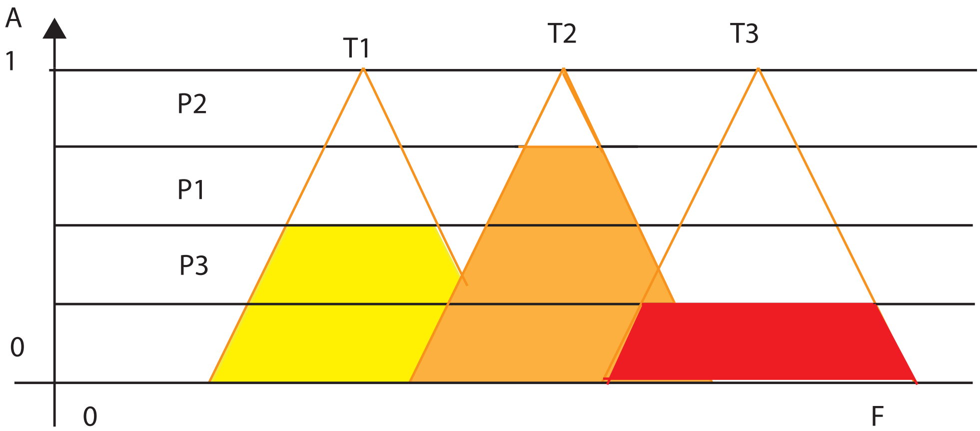


Fig. 6.11: Logical verity product AND of term course. A – verity; B – condition verity; C – output term; T1 – term 1; T2 – term 2; T3 – term 3; D – fuzzy result series; E – output language variable; F – resulting set; P1 – conditon 1; P2 – condition 2; P3 – conditon 3

6.23 Centroid method – CoG

For final evaluation is possible to use several methods. If we use centroid method CoG (Centre of Gravity) we compute centroid of united area. The result is his axis in horizontal direction.

6.24 Bisection method – CoA

If we use bisection method CoA (Centre of Area) the result is axis of vertical section which divided final areal into two equally large parts.

6.25 Maximum method – LM, RM

In use left maximum method LM (Left Most Maximum) is situation of the largest from the left maximum as a result. In use right maximum method RM (Right Most Maximum) is situation of the largest from the right maximum as a result. If the maximum isn´t „sharp top", but „plate" (horizontal) then is possible to use middle maximum method where the result is midpoint of a line. Somewhat danger would be determine result as average value from the situation of two or more consistent tops. We can also use another method.

6.26 Singleton centroid method – CoGS

Singletons centroid method CoGS (Centre of Gravity for Singletons) is often use for its small computation needs. As a result is horizontal axis of singletons centroid. Singletons are limited in verity values which correspond to output terms (each „tangible stakes“ are shorten) and horizontal component centroid axis is computed. For this axis is applied simple relation which has them analogy in mechanics in the balance on two-armed work computation:

$$x_{tez} = \frac{\sum x_i y_i}{\sum y_i}$$

Where xtez is final value – singletons centroid axis in the direction of horizontal axis (along the arm), x_i are situations of each singletons and y_i are values corresponding to output terms verities („stakes high“).

6.27 Weighted average analogy

From another aspect we can comprehend singletons centroid computation as standardized weighted average where the situations of singletons values are summation (x_i) and then they are multiplicand by corresponding probabilities (y_i) as weighted coefficient and the result is normed by weighted coefficient sum $\sum y_i$ – i.e. Sum of all output term verities (combined weight of all „stakes“). Implicit premise is request to verity sum is not-zero. Methods centroid and bisection united area are weighted average generalization.

6.28 Mamdani fuzzy system

The two examples mentioned heretofore are not alike very much, yet they have common features and illustrate common process during program implementation of different types of fuzzy systems. Two stages can be distinguished therein:

- fuzzification - in this stage, numeric value of the input language variable is converted to logical values of one or more input fuzzy variables – input fuzzy terms,
- evaluation of logical values, inference (logical) core – group of rules or logical expressions that describe logical relations among the input and output terms,
- defuzzification – process roughly opposite to fuzzification – the logical values of the group of output terms defined above output language variable are assigned a single (sharp) value of the output language variable.

Similar process is repeated regardless of the fact, whether the fuzzy system implements a control, decision, or optimizing algorithm, a regulator, a regulator adjuster, a diagnostic or control system, a model, an expert system, or other function. The described system is sometimes referred to as Mamdani system or Mandani's automaton.

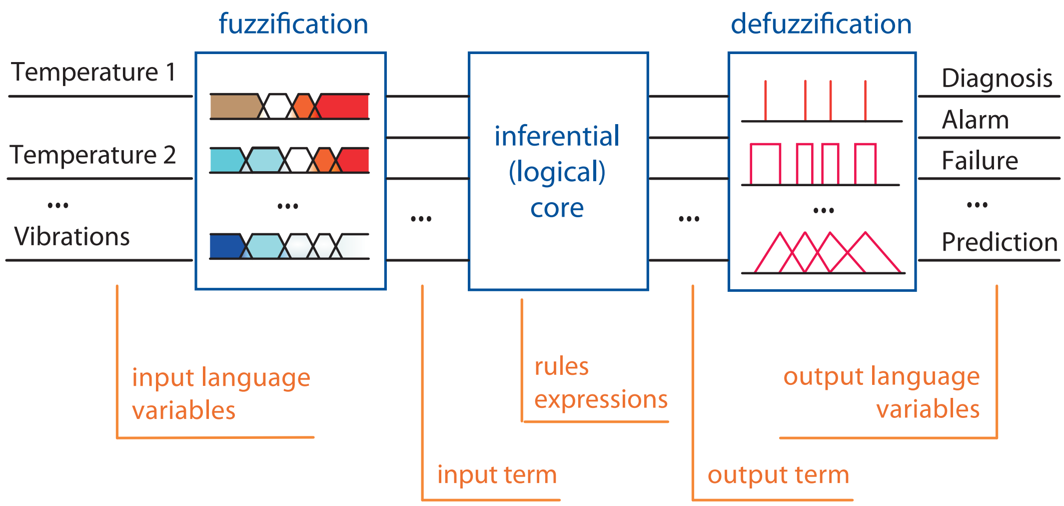


Fig. 6.12: Diagram of a typical process during implementation of Mandani system.

6.29 Fuzzy system tuning

Regarding to the fuzzy system complexity is almost exempt to fuzzy system will practices his functions by required way promptly after his programmed. It is necessary tuning phase. It is not expediency to change all parameters which can influenced behavior system and also shapes of his transfer characteristic. Usual are tuning „just“ shapes and positions logical function input and output terms other parameters are fixed (if they are not demonstrably wrong select).

Problem is exacting and lengthy. To be just not accidental „wandering in darkness“ is useful to come to realize how is influence the shape of transfer characteristics by partial hits.

6.30 System description

We will illustrate the process on the simple examples of fuzzy system Mamdani type with one and two input language variables and with one output language variable. Above each language variable is defined 7 terms. Five input term have shape of triangle, two outer have the same shape descending and entering ramp. Output terms have shape of narrow rectangles which imitate singletons. Logical system core is formed by simple set of rules which relay out input terms verity to output. If is use minimum function for the computation logical produce AND for logical sum OR is use maximum function. Defuzzification is provides by centroid singletons method. For the realization is use computing system MATLAB.

6.31 Uniform placement of all terms

At first we will show the example when input and also output terms are uniform placements. For one-dimension system is evident that transfer characteristic is in the middle linear with almost rectilinear course which is making round on the both ends and crossing into the horizontal section of saturation owing to edge effect. Fuzzy system is possible to consider as linear with nonlinear saturation – what is usual for most of real systems.

Shape of convert characteristics for two-dimensional system in middle part is near to inclined plane, only at boundaries are evident nonlinearities like saturation.

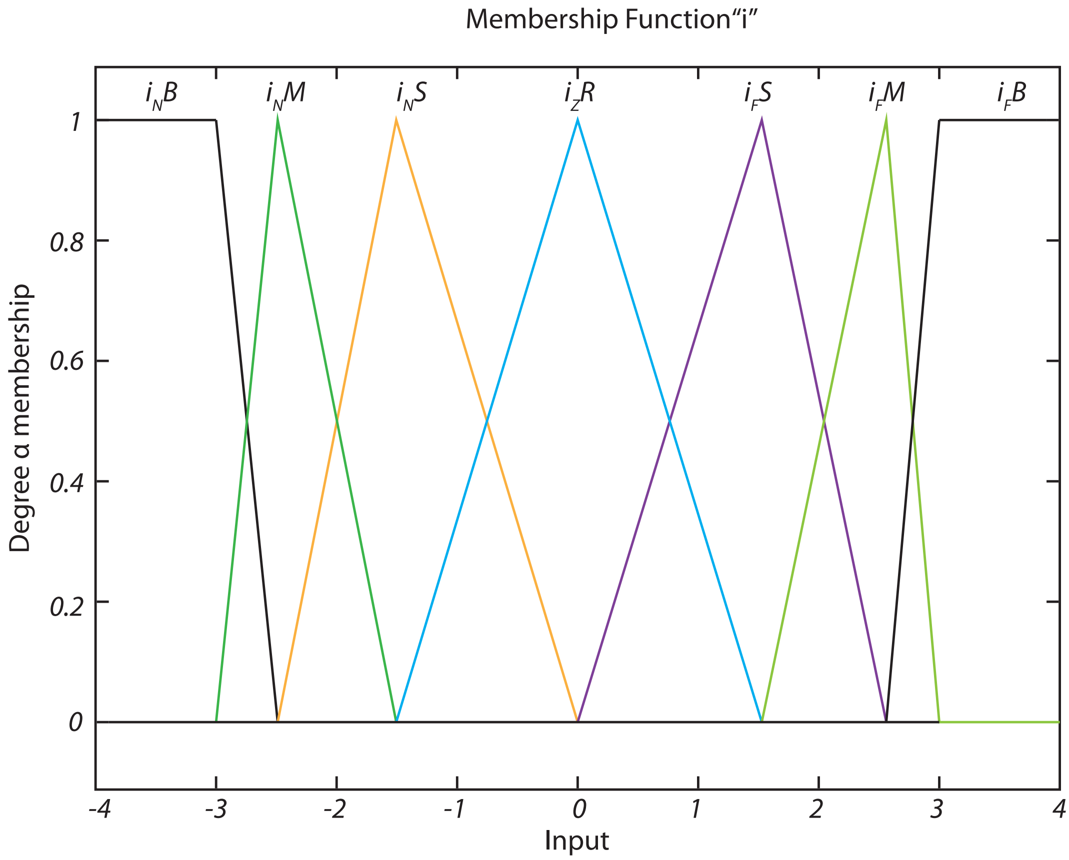


Fig. 6.13: Possible behav. of the logical fun. for cont. fire sensor

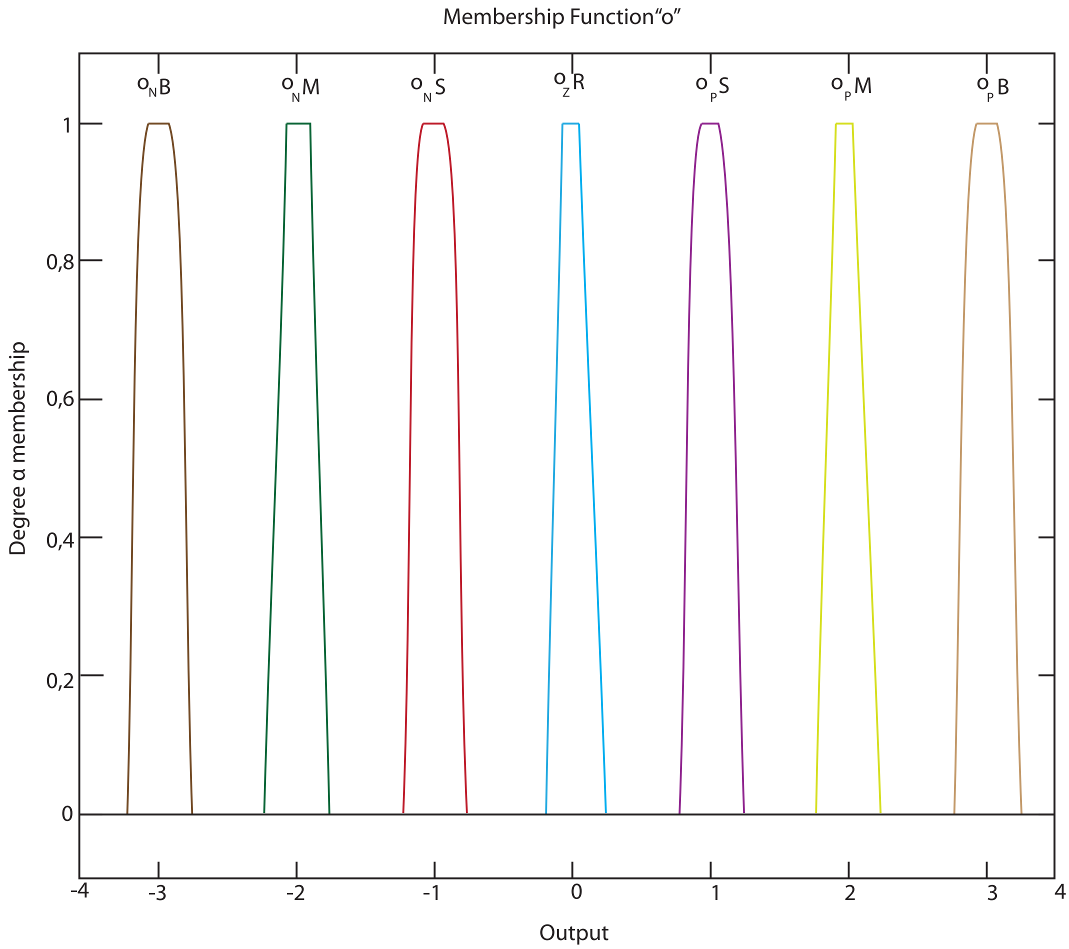


Fig. 6.14: Equally distributed output terms

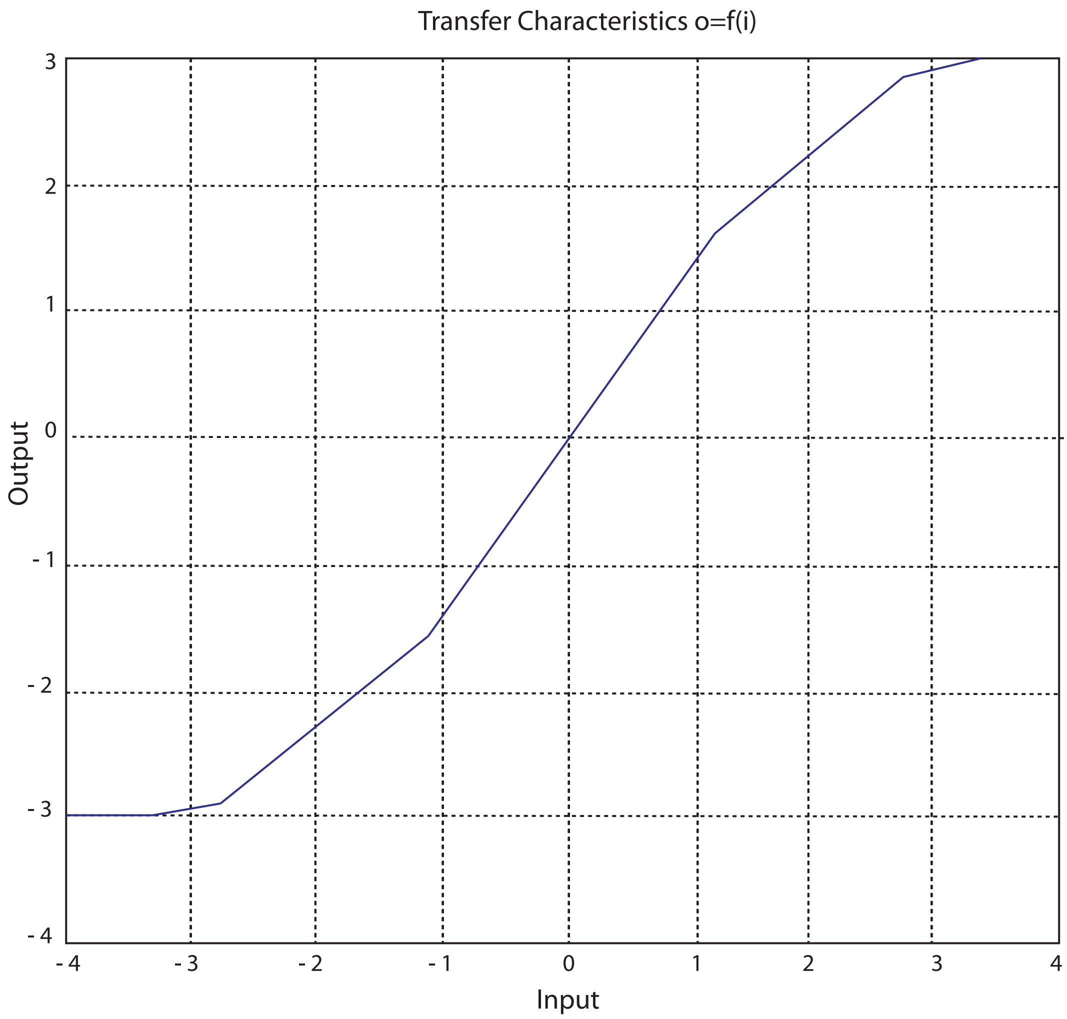


Fig. 6.15: Convert characteristic for one dimensional system

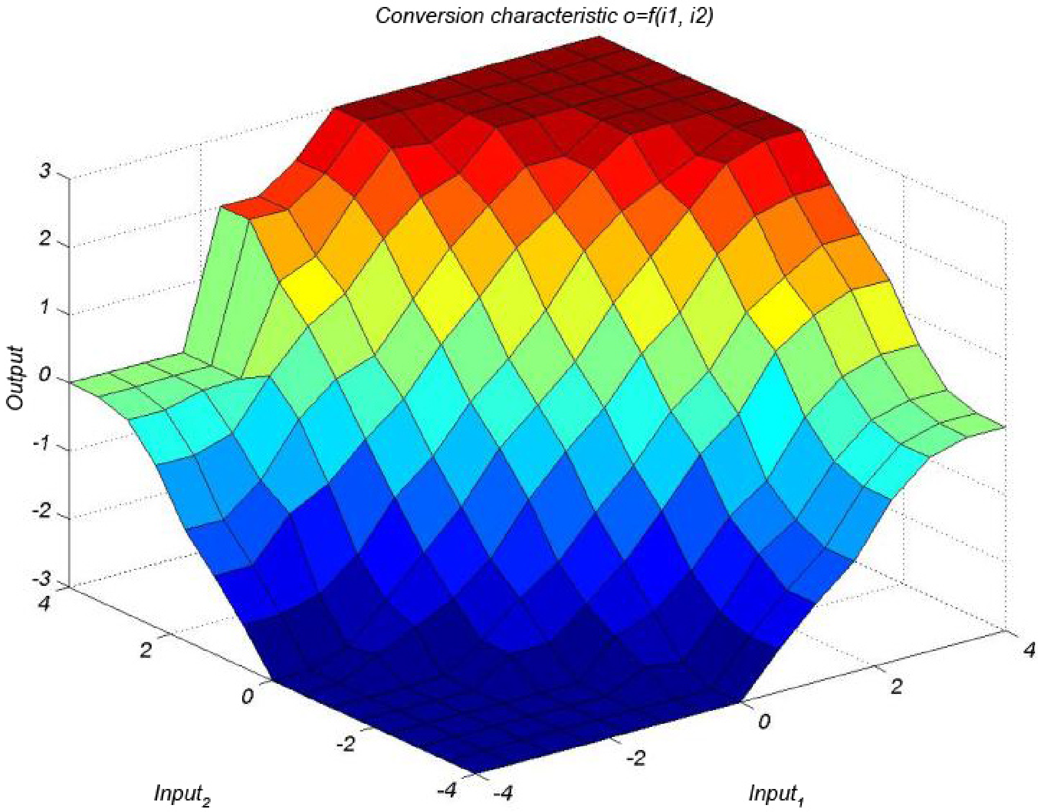


Fig. 6.16: Convert characteristic for two dimensional system

6.32 Consolidated input terms

As next example we will show fuzzy system which output terms are closely distributed in the middle than in the boundaries. Output terms staying equally distributed. Convert characteristic has a sharp of letter „S“ and in the middle embodies markedly bigger steepness (thickness) than in the boundaries. Likewise as for two-dimensional system is the area of convert characteristic in the middle markedly steepness than at boundaries.

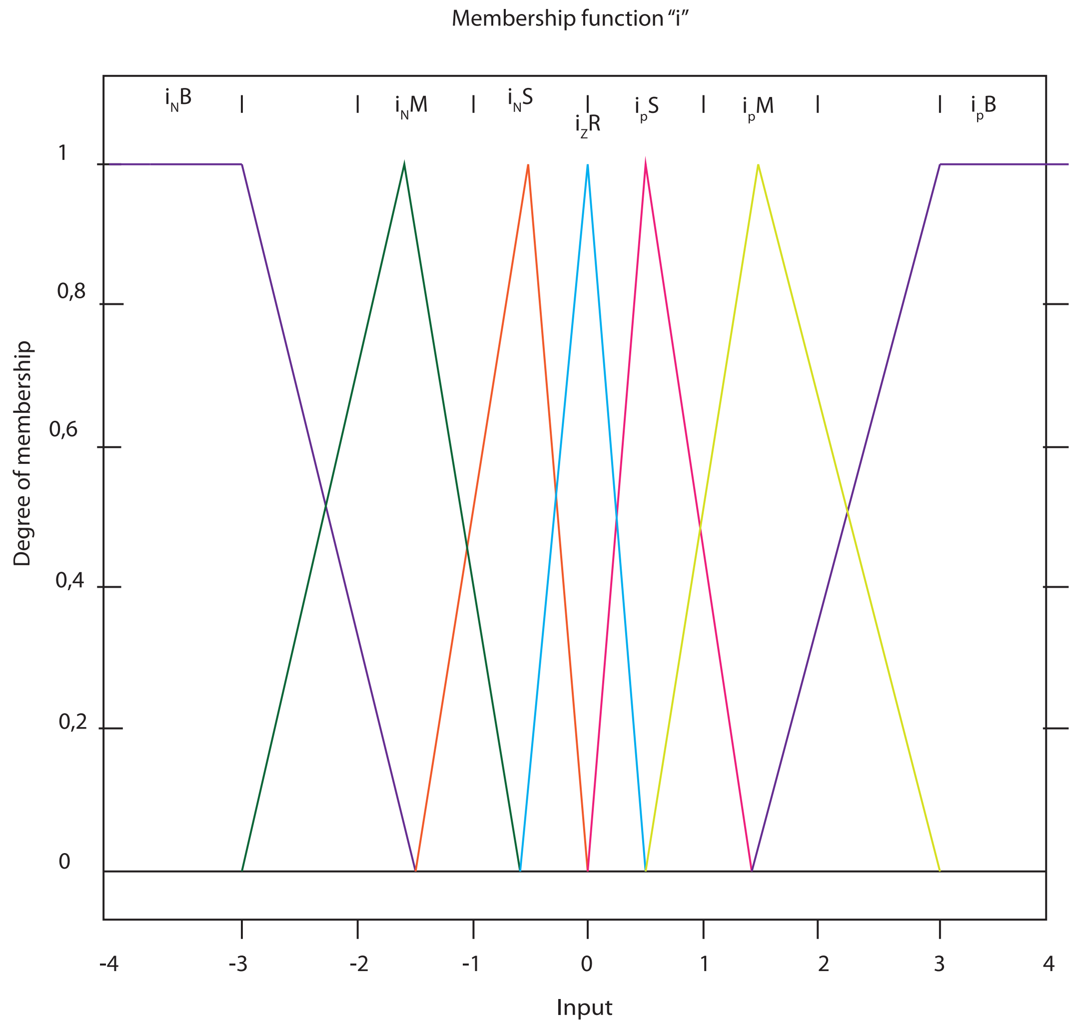


Fig. 6.17: In middle condensed input terms

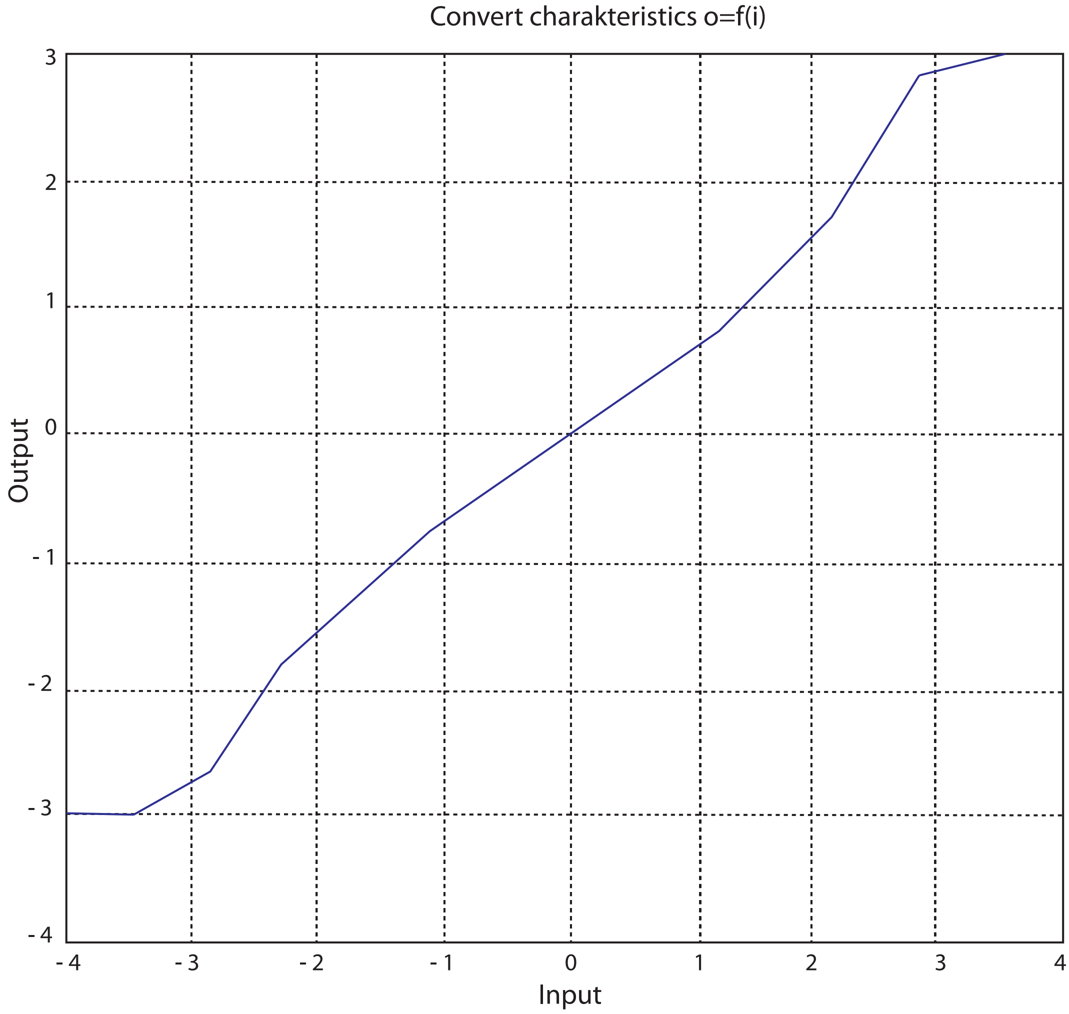


Fig. 6.18: Convert characteristic of one-dimensional system

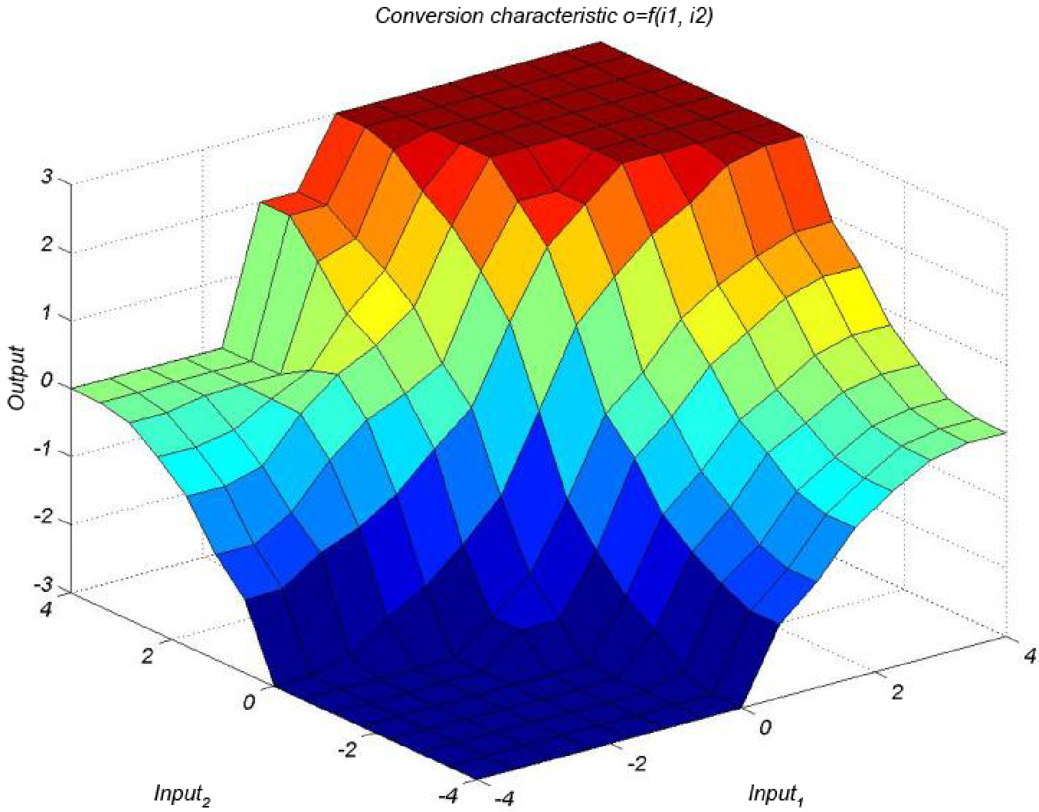


Fig. 6.19: Convert characteristic of two-dimensional system

6.33 Dilute input terms

As next example we will show fuzzy systems which input terms are thinner distributed in the middle than at the boundaries (condensing towards to the suburbs). Output terms staying equally distributed. Convert characteristic in the middle embodies markedly smaller steepness (thickness) than in the parts which are farther from the middle. Near suburbs there is another curvature in consequence edge effect. Likewise as for two-dimensional system is the area of transfer characteristic less steepness in the middle than at boundaries.

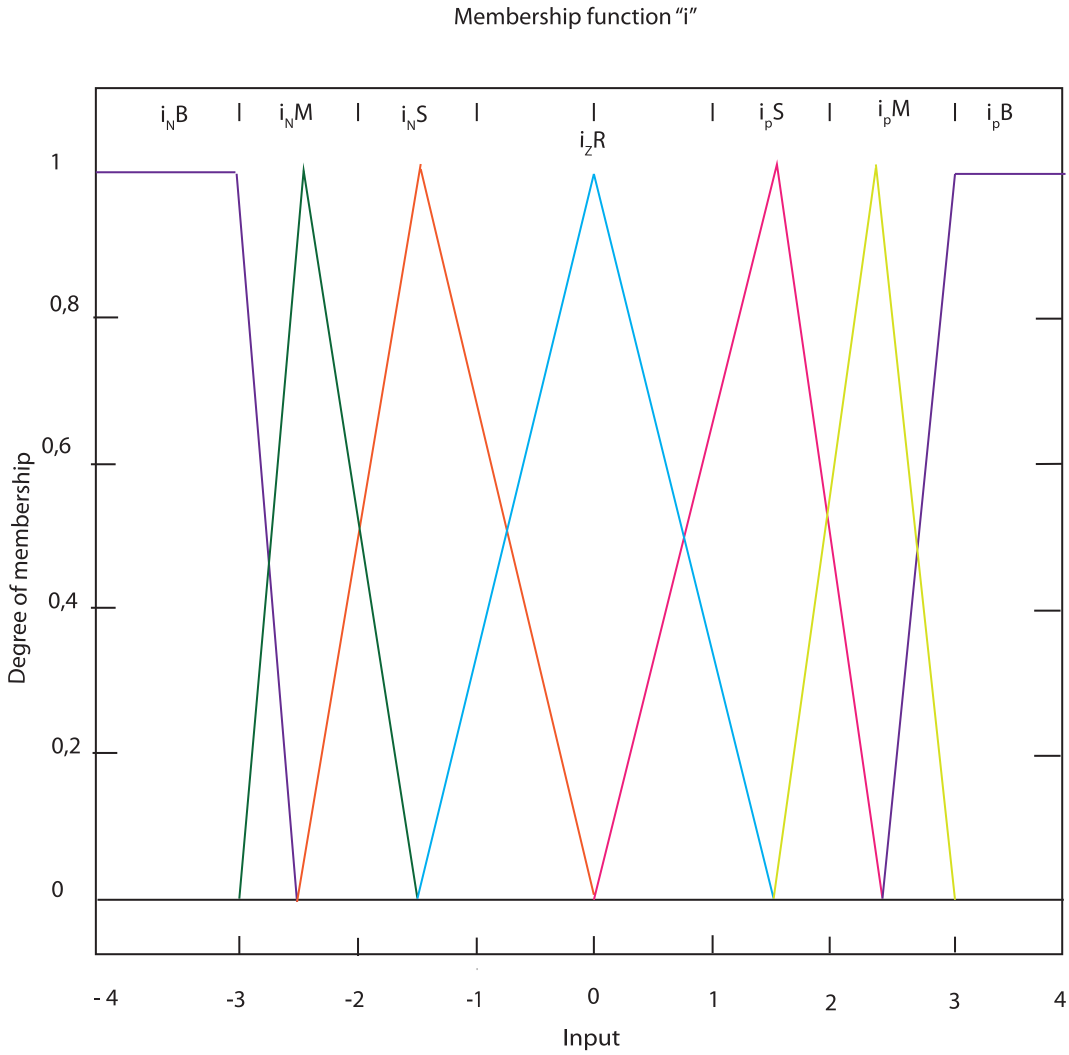


Fig. 6.20: Dilute input terms in the middle

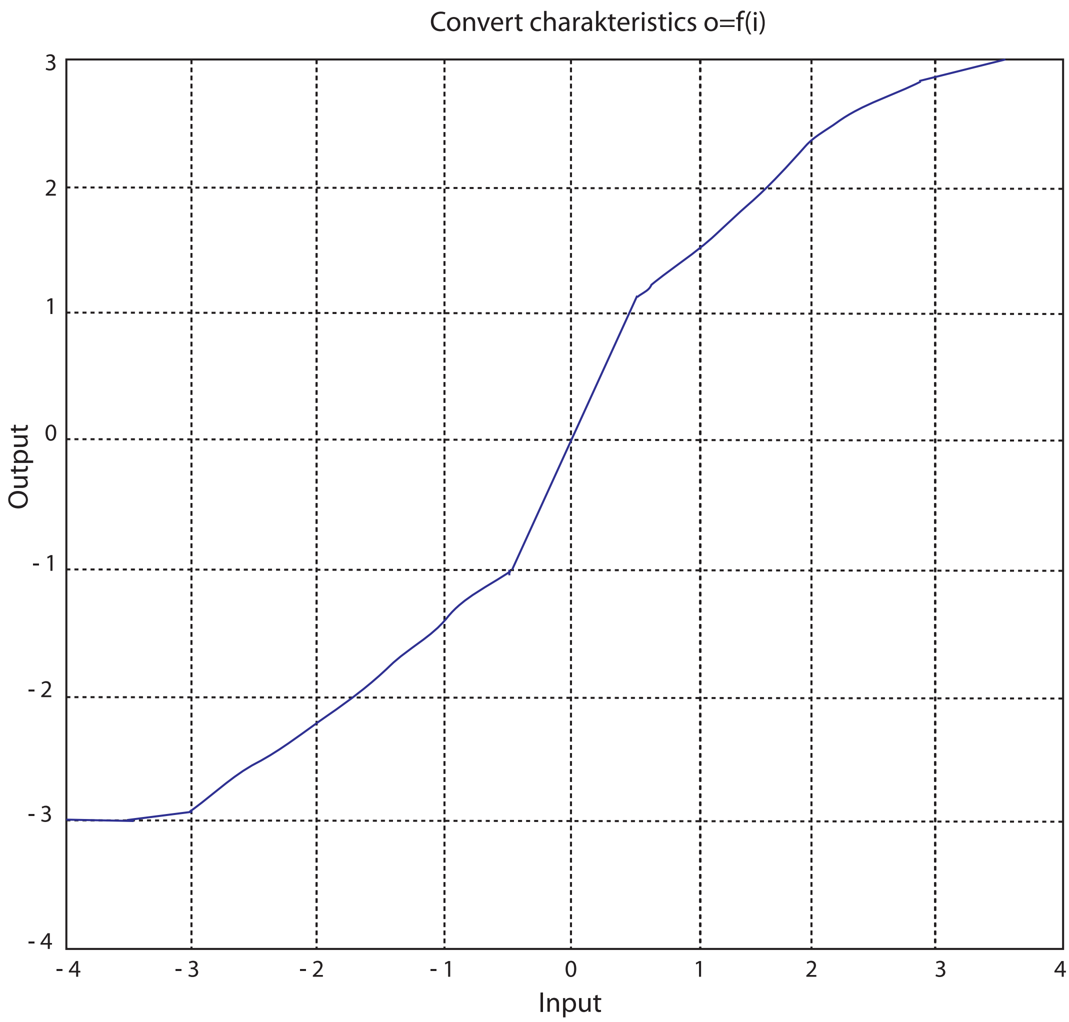


Fig. 6.21: Convert characteristic of one-dimensional system

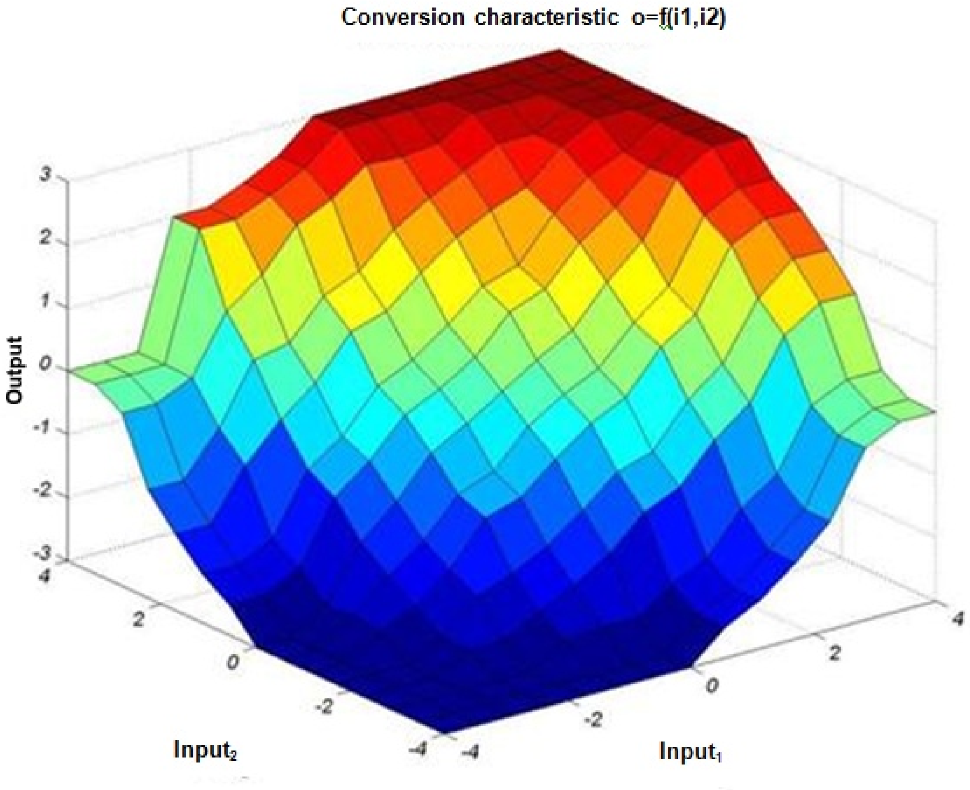


Fig. 6.22: Convert characteristic of two-dimensional system

6.34 Consolidated output terms

Manipulation with output terms (singletons) has counteraction to the shape of transfer characteristic than manipulation with input terms. Compacting output terms toward to the middle results in flattening middle part of transfer characteristic while dilution output terms in the middle of range increase steepness of transfer characteristic in middle part. We will illustrate this reality only on examples of one-dimensional systems.

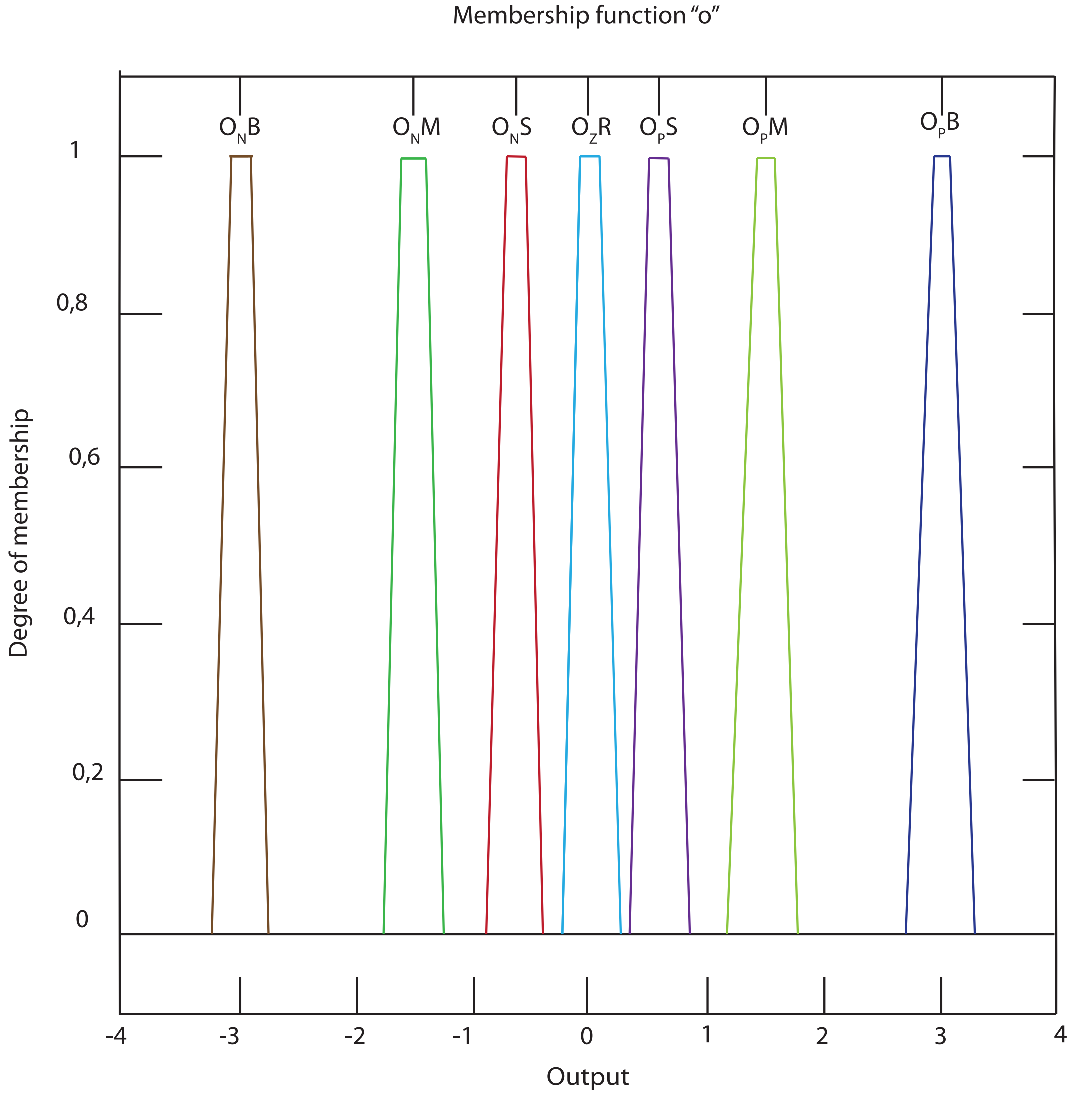


Fig. 6.23: Compressed output terms in the middle

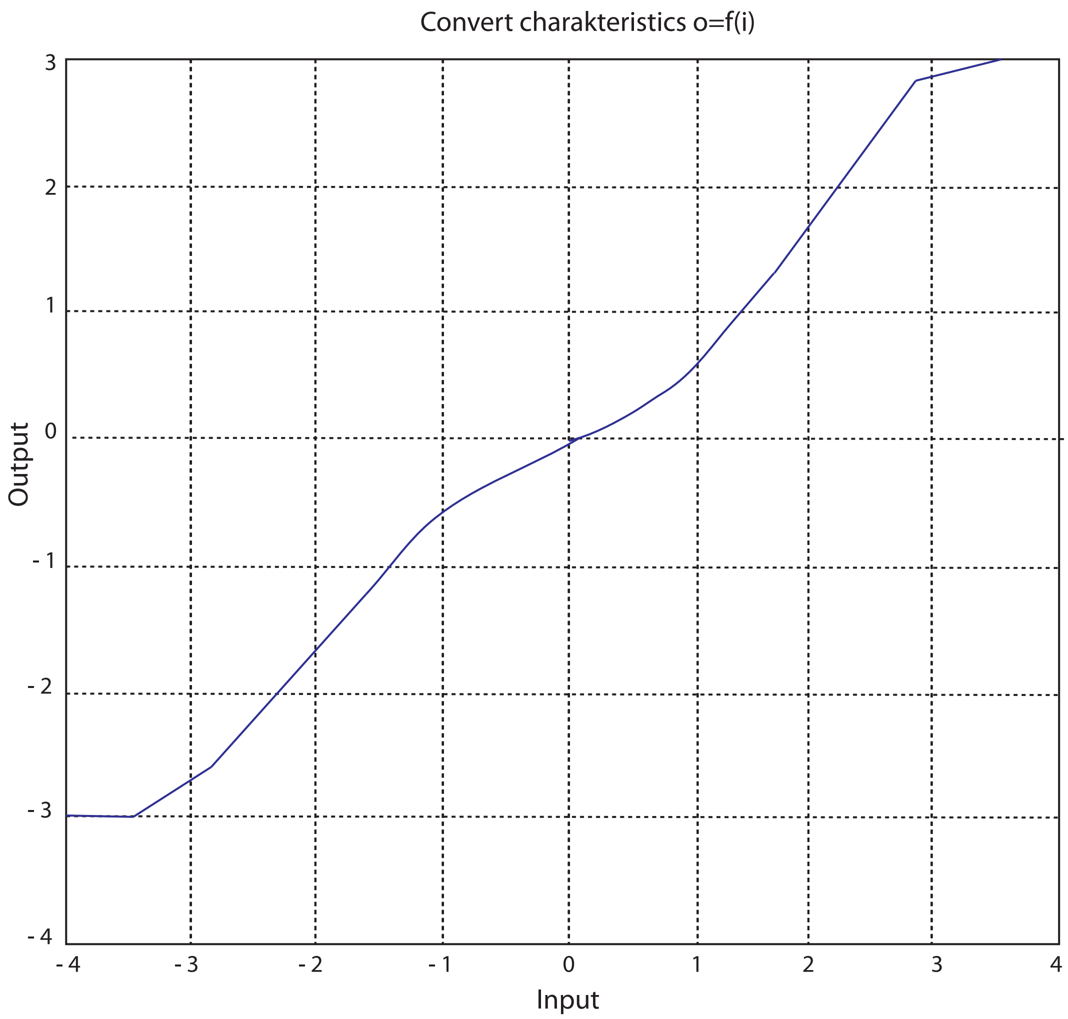


Fig. 6.24: Convert characteristic of one-dimensional system

6.35 Dilute output terms

The shape of transfer characteristic (especially steepness in middle part) is possible to influence by change in distribution of input or output terms. In efficient examples is possible to manipulate along with input terms. But purposeful is just thickening input terms along with dilution output terms by which increase steepness of transfer characteristic in the middle or on the contrary dilution output terms along with thickening output terms in the middle of range in which are lessen steepness of transfer characteristic in the middle part. Compressing or dilution terms in the middle part of range at the same time are senseless, because effects with one another are cancelling by manipulation.

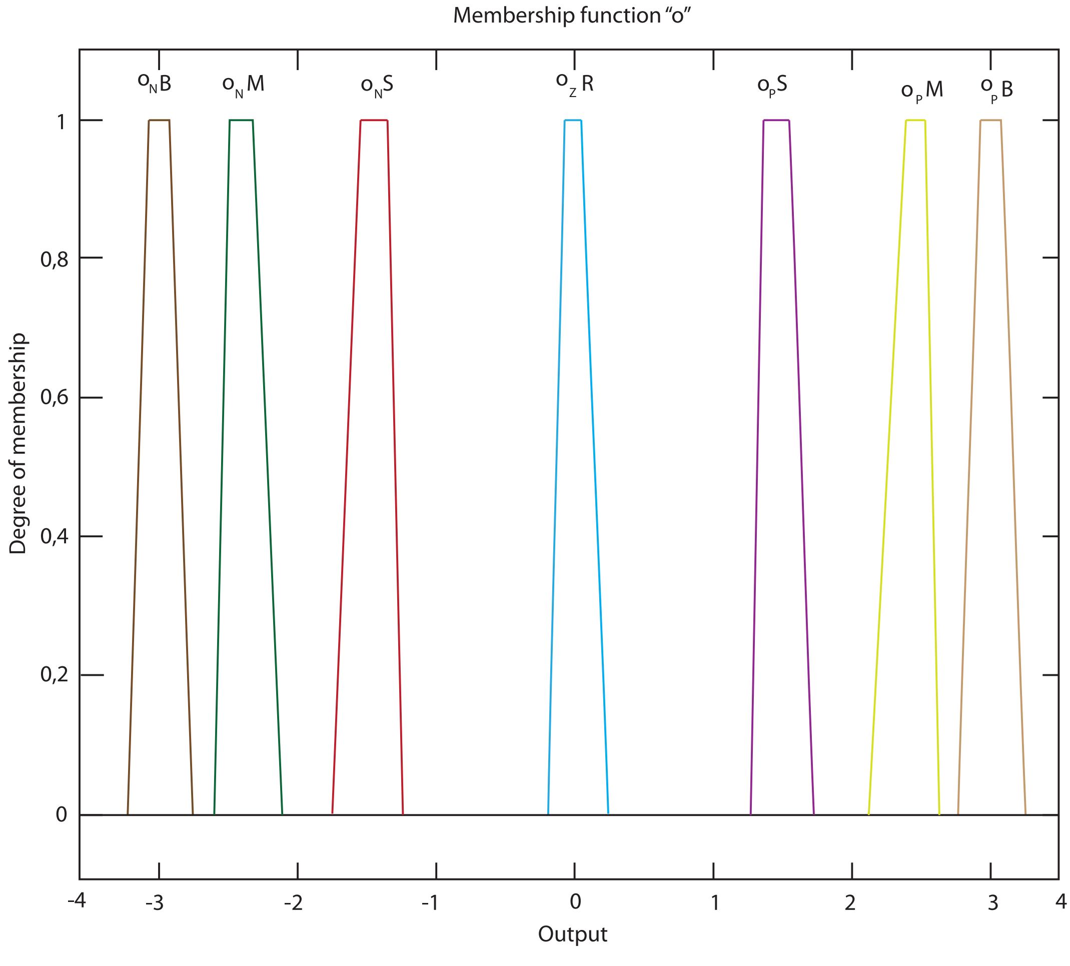


Fig. 6.25: Diluted output terms in the middle

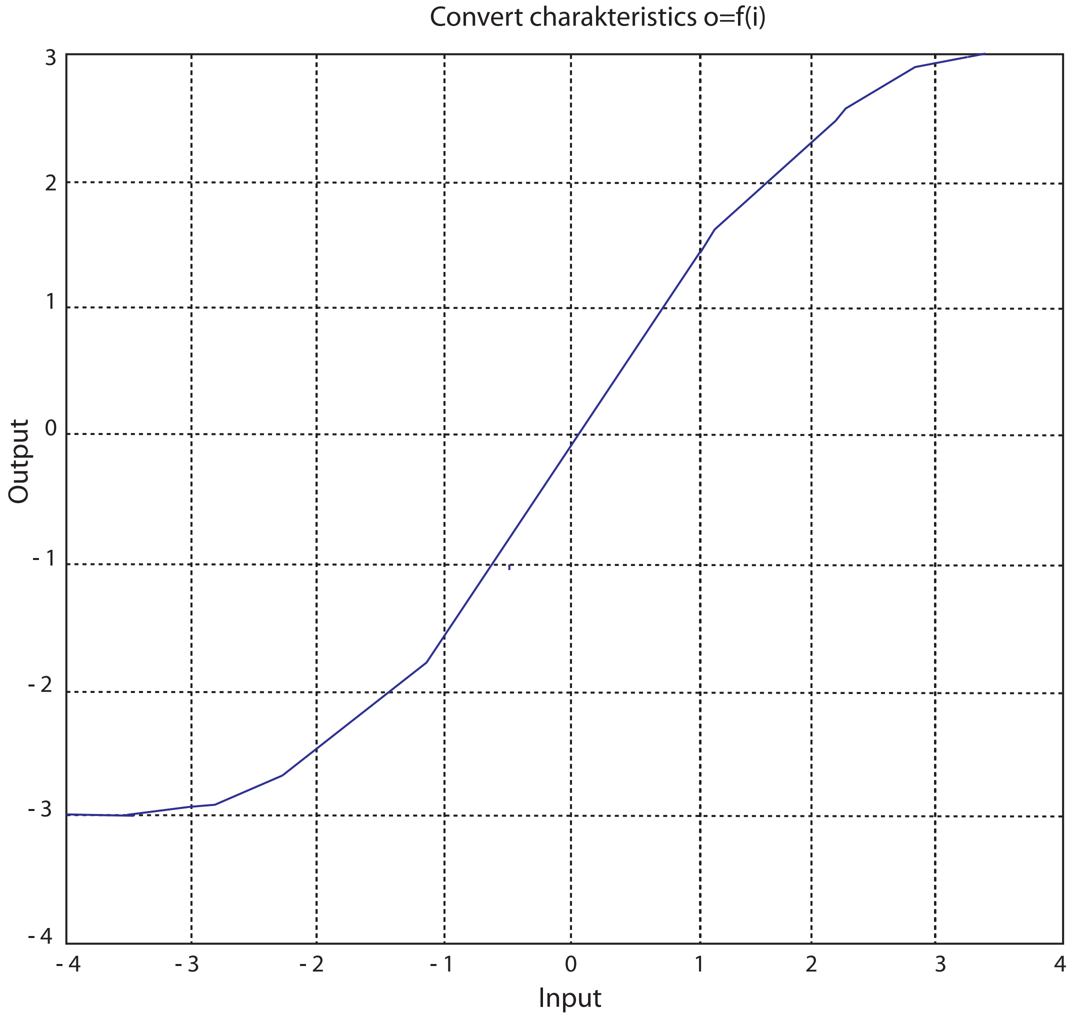


Fig. 6.26: Convert characteristic of one-dimensional system

6.36 Chapter's summary

These chapters are relatively extensive. At their beginning is mentioned AND, OR and NOT generalization. Then is established fuzzy diagnostic system from the fuzzification over establishing inference up to defuzzification. There are demonstrated individual defuzzification methods and enumerated their advantages and disadvantages. Defuzzification methods are also demonstrated on concrete examples in MATLAB. In other parts is shortly describe MATLAB´s fuzzy toolbox. And there is also describe Mamdani fuzzy system and Sugeno-Takagi system. Very important is part about fuzzy system tuning, because it is one of the most demanding operations in practice. At the close are defined inputs and outputs terms in all forms. Chapters are close by enumeration of usability fuzzy algorithms in practice.

7 Neural nets

7.1 Introduction

Neurons are basic elements of nervous system. They are life cells that are specialized in collecting, preserving, processing and transmitting information. There is a wide range of neuron types. In all types, the neuron consists of a body (soma), which gets information through input branches (dendrites) – a single neuron has about ten thousand dendrites – and which transmits the information through a single output (axon), which has a large number of branches at its end. The output signal of an axon corresponds to inputs that are processed inside the neuron. Therefore, the output of the neuron depends on inputs. The diameter of neuron's body varies. It may range from several μm to several tens of μm. The length of dendrites is two to three mm in maximum. On the other hand, the axon may be over one meter long.

In order to perform their function, neurons must be interconnected into neural networks inside the brain. Such interconnection is accomplished by dendrites that – by means of special projections (synapses) - connect to dendrites of other neurons or directly to bodies of other neurons, or even directly to axons in special cases. Intelligent and powerful behavior of the brain (neural networks in general) is given by perfect interconnection. Neurons of neural networks of the brain are arranged such that failure of one neuron cannot cause danger for the whole (due to combination of topological features and parallel processing).

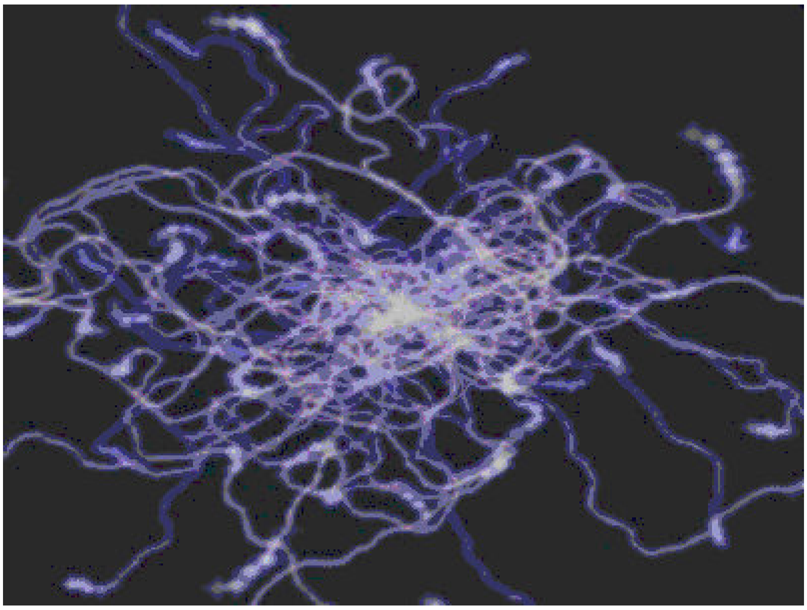


Fig. 7.1: Live neuron

7.2 Principles

Neuron is the basic element of every neural networks; it processes input data by a chosen method.

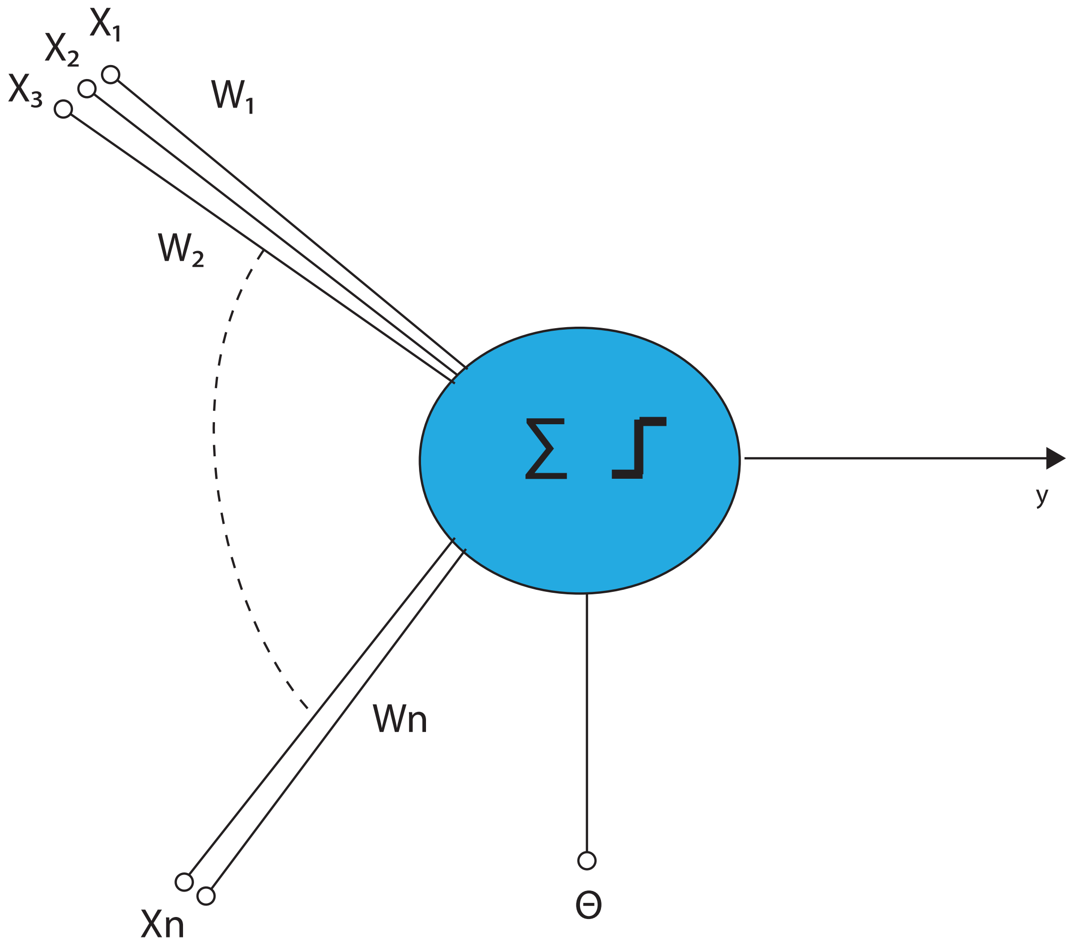


Fig. 7.2: Neuron

According to the type of input data, neurons are further classified as binary and continuous, i.e. processing multi-value information. The term "continuous" points out the philosophy of signal processing, where in the interpretation is discretized in absolute majority of cases as neuron and neural networks are simulated on a computer. From original HW implementations, only the class of optical implementation remains important.

An important term in the field of neural networks is a transfer function, sometimes referred to as activation function. This function converts the internal potential of the neuron to the range of output values. The most important functions are shown in the figures.

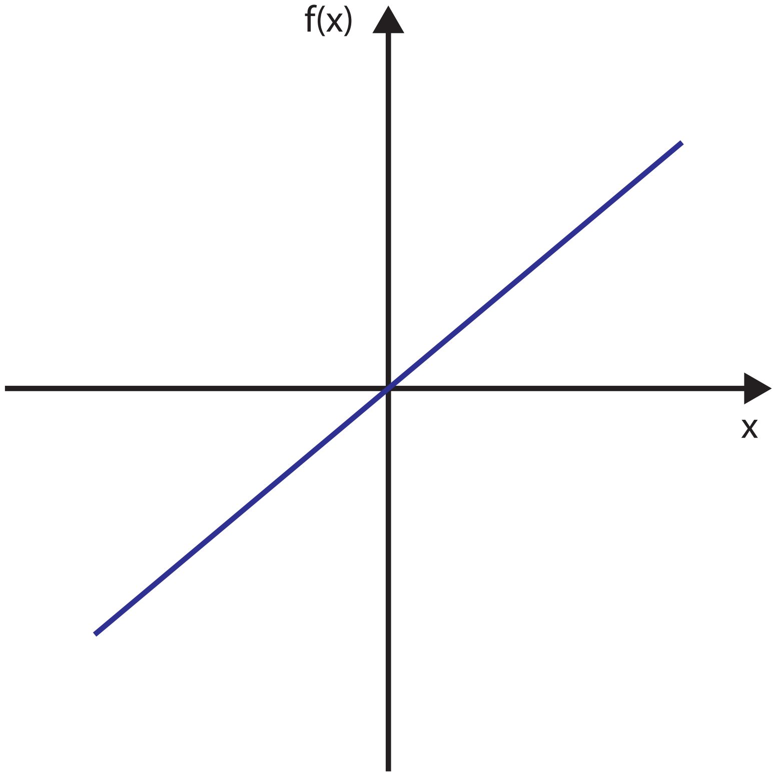


Fig. 7.3: Linear transfer function

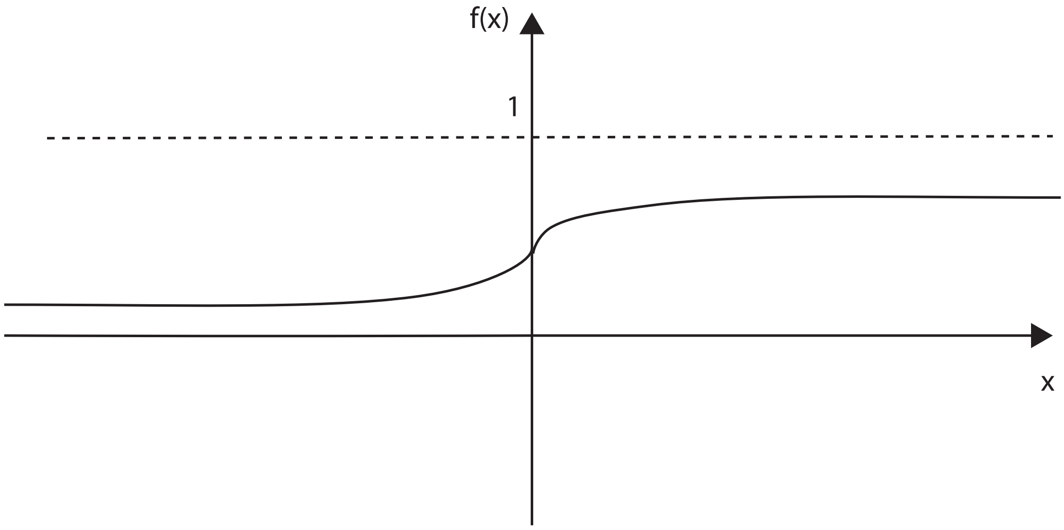


Fig. 7.4: Sigmoid transfer function

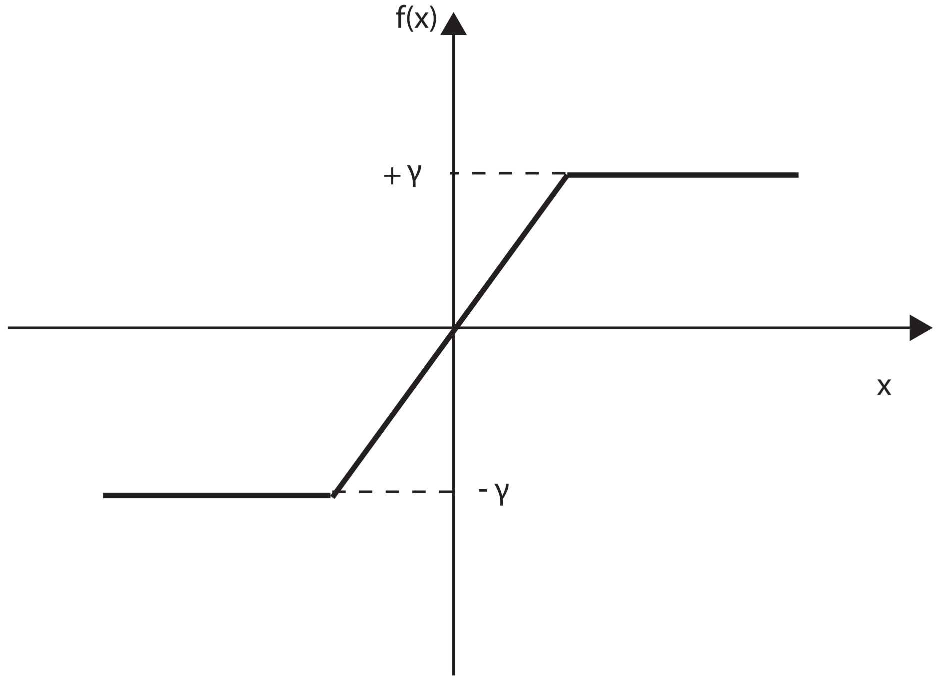


Fig. 7.5: Bounded function

Neural networks work basically in two phases – learning phase (adaptation) and active phase (relaxation).

During learning, NN are change in such a way that the network is being adapted for solution of given problem. The learning is realized by setting weights between nodes. In practice, this is achieved by assignment of initial values, either random, or chosen upon experience, or according to some similar problem previously solved. This is followed by introduction of a training input.

The learning is further divided into supervised learning and unsupervised learning.

During supervised learning, NN learns by comparing actual input with the required output such that the weights are adapted towards the best match. Decreasing the difference is controlled by a learning algorithm.

On the other hand, unsupervised learning does not have any specific validity criterion. The learning proceeds such that the algorithm searches in the input data for samples with similar properties. Such learning is sometimes called self-arrangement.

During active phase (relaxation), a state outside equilibrium appears in the output layer based on data inputs. The values stored within neurons begin to change by effect of other neurons, until a stable equilibrium is established.

7.3 Artificial Neural Networks

Even a single neuron may implement rather complex (non-linear in general) functions. It is usually called perceptron and it is able to e.g. process system inputs and recognize fault symptoms, single characters in a text, simple elements of an image, etc. More complicated functions may be achieved by neural networks. They are usually connected into layers – input layer, output layer and one or more hidden layers. In feed forward networks, the signal flow proceeds from input layer to output layer. However, there are networks (recurrent, Hopfield network), where the signal can proceed in an opposite direction, similarly to sequence logical functions.

Neural networks are often used for classification of events and their sorting into groups, image processing and recognition, other cognition processes (e.g. diagnostics), creation of models, prediction or estimation of progress. In general, neural networks are suitable for solution of problems whose principles are unknown, not known enough or that we cannot describe well enough. Neural networks may be used as controller or models that are adapted by learning mechanism of the network. Neural network may be used for pattern learning (to observe actions of a skilled operation in various situations). In most cases, the learned (trained) network works in an appropriate way, but it cannot reveal the principles of the problem. It may happen that it suddenly fails in certain situations.

The learning process of an artificial neural network is rather complicated. Other methods are used as well, such as genetic algorithms.

The implementation of a program for a neural network is not difficult – it is substantially a sum of products and a rather simple non-linear function. It is possible to implement it even on a PLC. However, learning of the network is much more difficult – not only from the point of view of numerical complexity of the learning algorithm, but also from the point of view of competence and experience of the solver. Therefore, pre-programmed and verified programs are often used, e.g. specialized tools of general computing systems (MATLAB, Mathematica) or specialized tools for implementation and learning of neural networks.

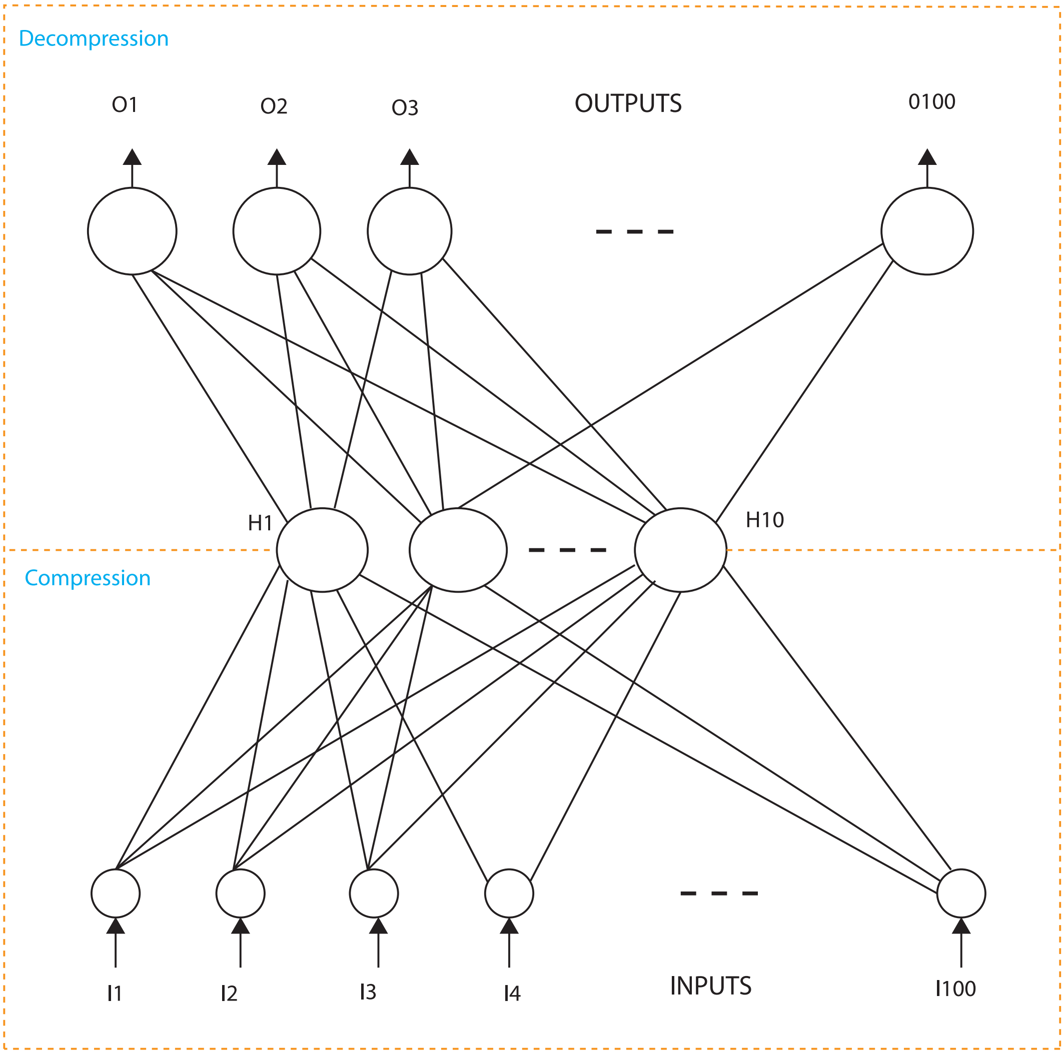


Fig. 7.6: Neural net for data compression

7.4 Perceptron

First of all, we will introduce the simplest model consisting of a single neuron, usually called perceptron. There are n connections leading to the neuron that represent either outputs of other neurons or stimuli from the exterior. Each of the input carries an information xi in the form of a real number every moment. The numbers represents values of some attributes; if we consider input information from the surrounding world – input space – the entire vector $\mathbf{x} = [x_1, ..., x_n]$ will characterize certain object of study. The attributes may be represented by data of temperature, pressure, color, Boolean, etc.

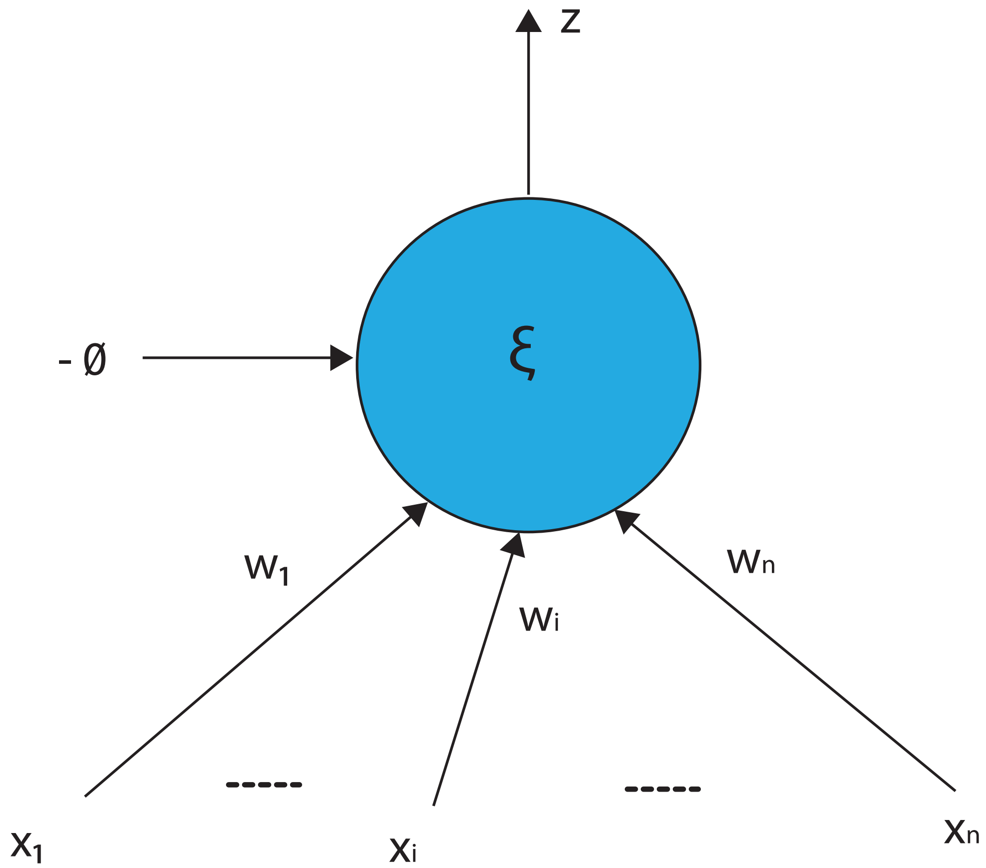


Fig. 7.7: Scheme of neuron with weights of respective connections

Every connection leading to the neuron is characterized by another real number wi which represents a so called synaptic weight (importance) of a connection, and every neuron is characterized by a threshold θ. A weighted sum $\xi = \sum w_i x_i - \upsilon$ (i= 1, ..., n) represents the overall stimulus, so called potential of a neuron. The neuron reacts on this potential by an output response z = S(ξ), wherein S is a predetermined non-linear transfer function, usually in the form of a sigmoid (monotonously rising between two asymptotic values, e.g. 0 and 1, with largest derivative in the point 0).

Networks with multiple neurons may be classified according to various aspects; we will concern on two of them. The first one is given by network topology, the other one by mode of operation. According to the first aspect, we may introduce recurrent networks (their graph contains cycles, i.e. outputs of some neurons are introduced back to the network as stimuli) and other networks, wherein multilayer networks are most important; they have neurons divided into layers, where the outputs of neurons of one layer are inputs of all neurons proximate to the "higher" layer; there is no other connection between neurons of such network.

Perceptron paradigm

Perceptron was invented by F. Rosenblatt in 1958. He was inspired by the model of human eye and tried to develop its model. He proceeded from the fact that retina comprises light sensitive sensors arranged into a matrix. Their outputs lead to specialized cells, so called daemons, genetically predetermined to recognize specific types of patterns. Outputs of daemons are further processed in cells with threshold behavior, so their output is active from a certain level of input stimulus.

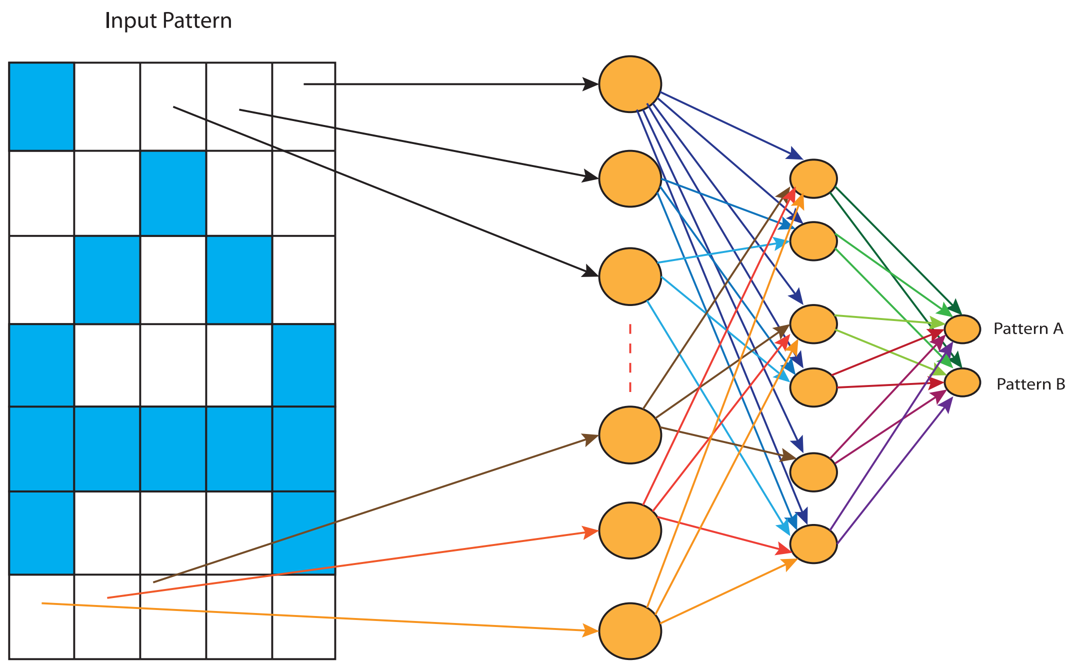


Fig. 7.8: Recognition with perceptron

Perceptron network is also three-layered, according to its physiological model. The input layer works as a compensating or branching layer. Its objective is to map a two-dimensional field of sensors onto a one-dimensional vector of processor elements. This second layer is formed by stimuli detectors. Each of them is randomly connected to elements of the input layer. The weights assigned to inputs are constant. The objective of the daemons is to detect specific stimuli. The last, third layer contains pattern recognizers (or perceptron). While weights of the first and second layer are fixed, the weights of the inputs of the output layers may be set up during training. For learning of a perceptron network, Rosenblatt suggested a so called perceptron learning algorithm.

The elements differ according to layers. The neurons have one input (threshold) fixed to a constant of 1. Other inputs are randomly connected to daemon outputs of the middle layer and their weights may be set up. The transfer characteristic of a processor element of a perceptron layer is as follows: the output is zero if the weighted sum of all its inputs is zero or negative. If not, the output is one. Other non-linear transfer function is sometimes used. In such a case, if the weighted sum of all its inputs is zero or negative, the output is -1, otherwise +1.

The weights are set up randomly. If the output is correct, the weights do not change. If the output has to be 1, but it is 0, weights on all active inputs are incremented. If the output has to be 0, but it is 1, weights on all active inputs are decremented.

The inputs are active if their value is > 0. The value of weight change depends on chosen option.

- Fixed increments or decrements are always applied.
- The increments (decrements) vary according to the value of error. It is advantageous to have larger increments in the case of large errors and vice versa. However, such acceleration of convergence may result in learning instability.
- Fixed and variable increments (decrements) are combined with respect to the value of error.

Besides the classic perceptron that has just been introduced, there are other types of perceptron – Minsky and Papert (MP-perceptron) and Adaline (or Madaline) by Widrow.

7.5 Back-propagation algorithm

One of the most used algorithms for neural network learning is the algorithm of back propagation. After twenty years of relative stagnation of neural networks, this algorithm started a new wave of research in this scientific field.

This algorithm is suitable for learning of multi-layer networks with feed forward propagation. The learning is of a supervised type. It is an iteration algorithm according to the method of computation. The energetic function is minimized based on gradient; therefore it is sometimes classified as a gradient learning method.

The principle of back propagation algorithm is similar to common learning method used in education. If a student finds deficiency in the schoolwork, he has to study more. This is similar in back propagation. Neural network can be considered as a student, testing may be considered as a mechanism for testing, whether the neural network answers to the input vectors according to a training set. Training set is a schoolwork that the student has to study. If we find out that the network does not respond correctly, we must change its weight coefficients, until it starts to react correctly.

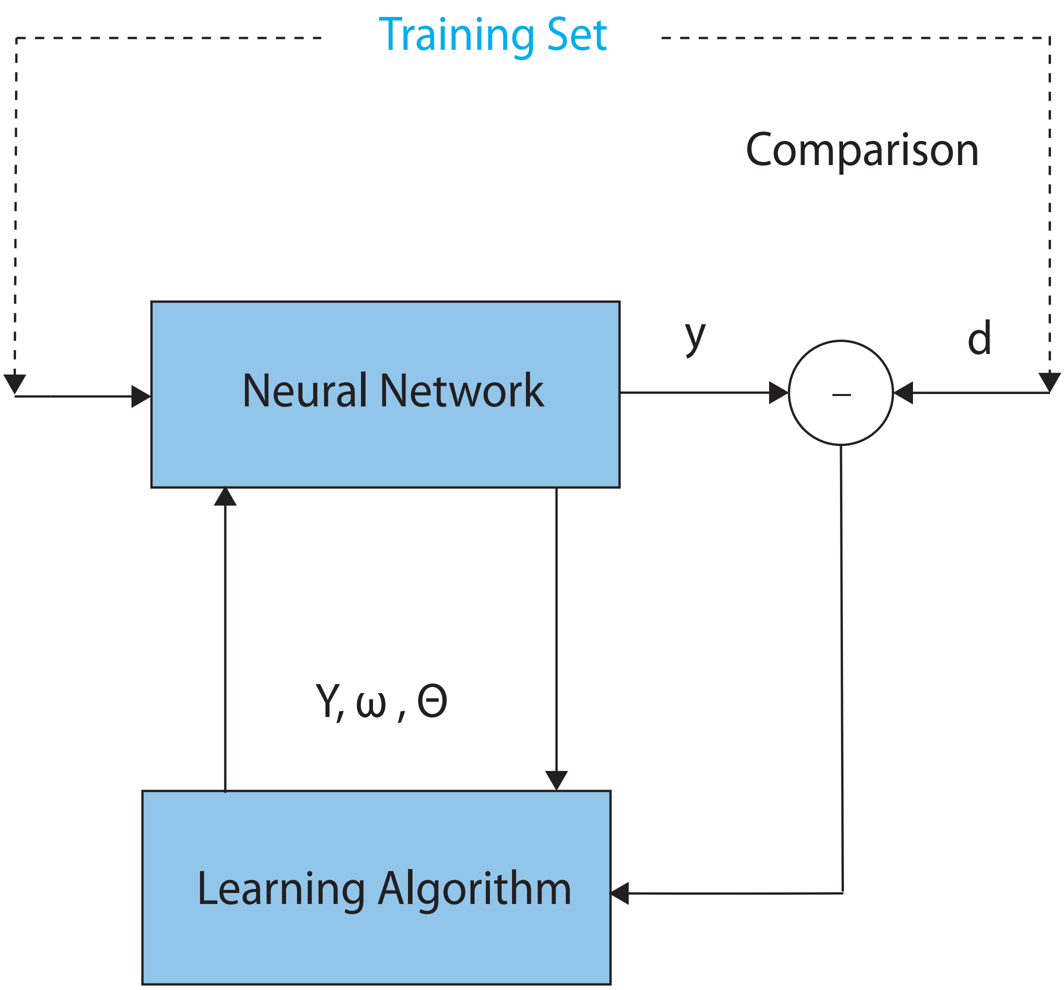


Fig. 7.9: Graphical representation of neural network learning

Back propagation is an iteration process, wherein we proceed from an initial state to the state of full knowledge.

The choice of weights from the training set does not need to be sequential, very often we use e.g. random order. The choice of strategy is not given strictly, it rather belongs to a category "art of learning neural networks". In the course of learning, we may make some patterns more frequent, some less etc. A suitable strategy may significantly influence the speed and success of learning. A global error in the algorithm can be represented by e.g. mean square error calculated over all patterns from the training set. This error represents the extent of knowledge of the network. The criterion is a limit, where the learning process is stopped. The algorithm of back propagation is based on minimization of energy of a neural network. Energy represents the extent of knowledge, i.e. deviation between real and required values of outputs of a neural network for given training set. It substantially the same as the global error, but for calculation of the global error, we use rather statistic criteria.

For a neural network with back propagation, the energy function is defined as follows:

$$E = \frac{1}{2}\sum_{i=1}^{n}(y_i - d_i)^2$$

Where in d is the number of inputs, y_i is i-th output and d_i is i-th required output. This function is nothing but a sum of squares of deviations. The back propagation learning algorithm has to be considered as an optimization method, which is able to find correct weights and thresholds for a given neural network.

7.6 Hopfield Network

In the beginning of eighties, J. Hopfield developed a new model of neural network inspired by study by auto-associative networks. During the research, he developed an energy function that has major impact on correct function of the network – rules for learning and relaxation are derived from said function. Hopfield illustrated the application of this network on several physical models. There are several modifications of this network today – Hopfield network mey be used as associative memory, classifier or optimization problem solver. The behavior of the network may be well illustrated on image patterns, as binary values can be easily assigned to image pixels. Hopfield network is not suitable for continuous inputs, as the conversion of continuous signals onto binary values presents a major problem.

7.7 ART network

One of the problems that limit the use of neural networks is the problem called "problem of variable stability" – the network is not able to learn new information without damaging already stored information. This effect is apparent for multi-layer perceptron network. When training network for new pattern, the entire network can be broken down, i.e. all information that has already been stored is lost. This effect is caused by change of weights of the network. To finally train the network for required new information, we are often forced to start over again. This may present a considerable time delay, hours or even days. The network that we are going to describe deals with the problem of variable stability quite well. This network was developed by a mathematician and biologist S. Grossberg. Adaptive Resonance Theory (ART) was developed for modeling of large parallel architecture for self-learning network for pattern recognition. A property of ART network is that it can switch between vigilant and stable mode without damaging already stored information. The vigilant mode is a learning mode, wherein initial parameters may be modified. Stable mode is a mode, wherein the network is fixed and behaves like a finished classifier.

Just note that ART network exists in three basic modifications (ART-l, ART-2 and ART-3). The basic modification, which will be described here, is ART-1; ART-2 does not work with binary values, but with real ones, ART-3 uses similar structure to ART-2, but the model of this network is described by equations, which express the dynamics of chemical carriers of information. This modification assumes that the inputs to this network come continuously and change continuously.

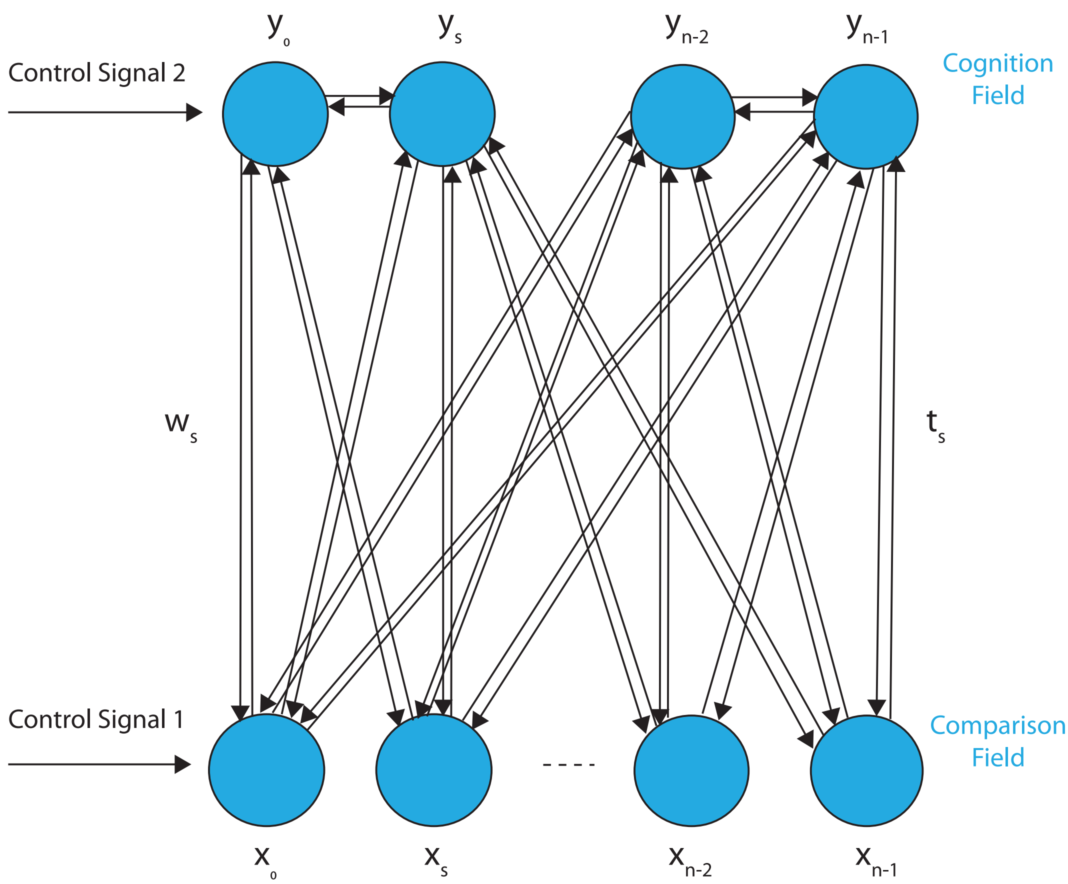


Fig. 7.10: Architecture_ART_net

7.8 Applications of Neural Networks

Neural networks are applied mainly where classic computers fail. The problems for neural networks do not have known algorithm, or analytical description it too complicated for computer processing. Their main application field is for problems with large example data that cover sufficient area of the problem. The basic application fields of neural networks are described in following: Economic information systems, Technology and manufacture, Health care, Meteorology.

Artificial neural networks have some advantageous properties. They can implement an arbitrary transformation above input date, so they are universal. A neuro-computer based on neural network does not need to be programmed, as its required behavior can be achieved by training with appropriate examples. Neural networks are, thanks to a large number of neurons and connections (synapses) and thanks to the fact that the information is distributed over the entire network, very robust. Failures of neurons lead only to a slow degradation of the network. They have an ability of generalization, abstraction, i.e. to react equally on a certain set of input data, not only to certain element from this set.

7.9 Repeating questions

1. Try to define using fuzzy logic in application praxis.
2. Make description of fuzzy system working procedure.
3. Try find example from praxis for combination of fuzzy system with classical control approach by PLC (mixed control).
4. Describe design of fuzzy system from definition to function results.

8 Digital systems

8.1 Introduction

Application of digital data processing systems is facilitated by greatly developed computer technology, especially by use of microprocessors, microcontrollers, microprocessor based personal computers and other programmable systems, especially programmable logic controllers (PLC). In practice, the digital systems have almost completely superseded their analogue counterparts, realized either by passive or active (e.g. operational amplifier) component circuits.

Equivalents of all analogue systems (e.g. controllers, filters, models) can be realized by digital systems. In addition, there is a kind of behavior, found in digital systems, which is unavailable to analogue systems: finite impulse response (FIR) systems, which do not have their equivalent in analogue domain. There are also other functions, inconvenient in analogue implementation (principally impossible or too complicated), e.g. nonlinear filters, statistical operations, identification or optimization tasks.

Digital systems are applied in numerous fields for many different functions. However, their structure and implementation are always very similar. The difference is in the design methodology and structural and parametric requirements. For example, the requirements for controller are stability and desired control performance. On the other hand, requirement for digital model is a maximum match in behavior to the original.

8.2 Digital filters

Digital filters have to modify frequency content of input signal in a particular way, e.g. to act as a low pass, passing low frequencies from specified band through, and attenuating higher frequencies. It is possible to realize high pass or band pass filter as well, using the very same structure, only using different parameters (coefficients in FIR or IIR [infinite impulse response] system interconnections).In certain applications, the requirements for digital filter are not imposed on frequency filtration, but rather on shaping of time behavior of processed value, e.g. to supress a noise or short time disturbance impulses. In digital image processing, a two dimensional filters are used. They are used for noise suppression, contrast manipulation, contour enhancement etc. Design of digital filter is with DFT and spectral analysis base for numerical control in signals. Describe the transfer, frequency response, impulse response and differential equation. These are algorithms or circuits, changing spectrum of discrete input signal. In real time the filter between the two samples have to calculate the convolution (FIR filters). Digital filters follow the passive and active analog filters and can be designed either directly (FIR), or by converting the analog prototype (IIR).

Filters are divided by impulse response:

- Filters with finite impulse response - FIR
- Filters with infinite impulse response – IIR

And by structure of scheme:

- Nonrecursive filters NRDF (haven't feedback)
- Recursive filters (feedback) mostly IIR filters

LTI

Linear time invariant systems.

If $x(t)$ is input signal and $y(t)$ output signal, then output signal lis given by transformation ofinput signal, thus $y(t) = \mathrm{T}\{x(t)\}$.

Time invariance mean, that system responding for given input signal $x(t)$ by the same output signal $y(t)$. If system is excited by signal $x(t)$ shifted in time, $x(t-t_0)$, then system responding $y(t)$ with the same time shift, $y(t-t_0)$.

Linear system is system, where for multiply input signal $x(t)$ by k, responding k-multiplication output signal $y(t)$ and for sum of input signals $\sum_i k_i \cdot x_i(t)$ responding by sum of responses $\sum_i k_i \cdot y_i(t)$.

These behavior are very important, because tend to simplify some math operations, and system understanding. „Digital variant"for LTI are named DLTI (discrete LTI). In digital systems instead of analogue signals we work with numerical sequentions. Input signal is done by sequentions $\{x\,(n\,T), n$ belongs Z – discrete state space$\}$, output signal (response) numerical sequentions $\{y\,(n\,T), n$ belongs Z$\}$. Note $x\,(n\,T)$ mean, that this number can be size of signal in time $n\,T$, where T is period of samples. If we consider (input) signal as numerical sequentions without relation to time, we can samples write only with index n, $x(n)$. For many cases of use numerical computation is time variable important (e.g. for signal filtration), and we'll keep note $x\,(n\,T)$.

Discrete time invariant system convert input signal (sequentions) $\{x\,(n\,t)\}$ to output sequentions (signal) $\{y\,(n\,t)\}$, thus $\{y\,(n\,\mathrm{t})\} = \mathrm{T}\{x\,(n\,T)\}$. Impulse response of digital system is own response to one input sample applied in time t = 0. Input signal (sequention) is $\{x\,(n\,T) = 1$ for $n = 0$, $x\,(n\,T) = 0$ for $\mathrm{n} \neq 0\}$. In determination of impulse response digital system we assume, that before application input impulse is system relaxed.

For linear time invariant systems is done superposition rule – input signal we decompose into suitable parts, find responses for these parts a responses compose to output. Response for input signal will be given. Input signal $\{x\,(n\,T)\}$ we can decompose to system of samples supposed as impulses x size $(i\,T)$ placed in time $i \cdot T$. Response for this impulses is $x\,(i\,T)\{h\,(n\,T - i\,T)\}$. Whole output sequention as response for input signal $\{x\,(n\,T)\}$ is sum of all responses, responses for all i.

8.3 FIR

Filters with finite response. Not containing feedback and this is nonrecursive (convolution) filters. Transfer of causal (only positive values *n*) filtering by:

$$H(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots\dots\dots + b_M z^{-M} = \sum_{n=0}^{M} b_n z^{-n}$$

and differential equation is:

$$y(n) = b_0 x(n) + b_1 x(n-1) + \dots\dots\dots b_M x(n-M)$$

FIR filter are always stable.

Mostly used structure is transversal filter. It's delaying line with switches to multipliers. Filter computing weighted moving average from $M+1$ last value. Coeficients are values from impulse response – $b_n = h(n)$.

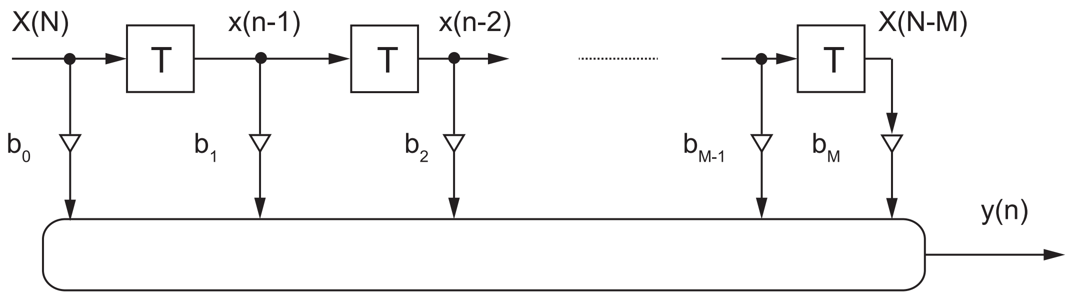


Fig. 8.1: Scheme for transversal FIR filter

Filter delaying signal for $\frac{M+1}{2}$ beats and stabilize after $M + 1$ beats.

8.4 IIR

Minimal one feedback is needed, and is recursive filters. Transfer is done by quotient of polynoms. Zeros realize nonrecursive part, poles recursive part. Order of filter is done by higher of polynoms order. Order is much lower than FIR filters, so response is faster. Example of simplified IIR filter is in follow picture.

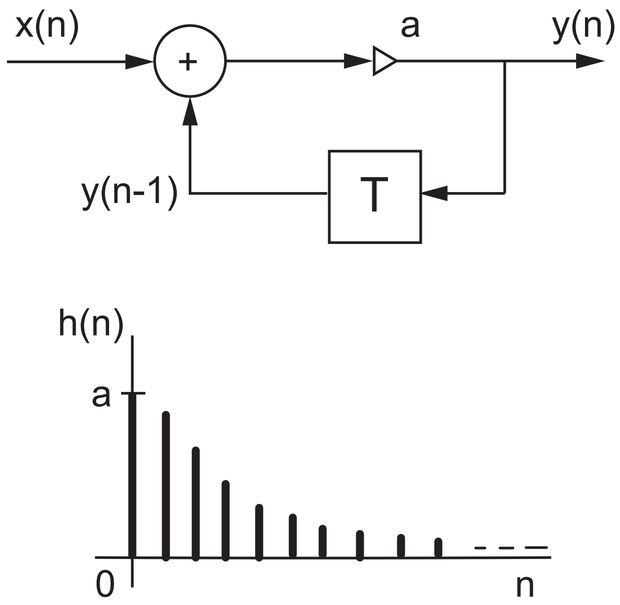


Fig. 8.2: Simple IIR filter

Filters are also divided by purposes:

1. Fervency selective digital filters

This is Lowpass (DP) , highpass (HP) pass filter (PP) and

Band-stop filter (PZ).

Realized as FIR or IIR filters with linear phase and as FIR filters.

2. Discrete integrator

Integral is computed by numerical procedures like e.g. $y(n) = y(n\text{-}1) + x(n)$

$$y(n) = y(n-1) + \frac{1}{2}\left[x(n) + x(n-1)\right]$$

3. Discrete differentiator

Differentiate input signal, important for acceleration detection or edge detections.

4. Moving average MA

Computing average from sample and M previous. Can weighted with same coefficients

$b_i = \frac{1}{M+1}$,(no weighted), or with exponential forgetting as recursive filter with weights of coefficients a-ci, where c is real number.

8.5 Application fields

Each application field of digital systems has its own tradition, terminology, notation, developed theory and design methodology, as well as experts in research and practice. Communication between these branches is only occasional, e.g. in case of internal model controller design or in case of control, where a massive filtration or prediction of controlled value behaviour is used. It is not possible or useful, to explain details of all fields of digital system theory and application. We will focus on control problems, discussing other fields only marginally.

We will not concern even in a control theory. It is well described in specialized monographs, very extensive in general. Following explanation will be based on insight into physical principles, engineer's feel and intuition. This approach is reasonable, among others because of that some theoretical methods are hard to be used in practice.

8.6 Numerical control

Let us describe the process of a feedback control (see Fig. 5.2d). The controller R evaluates the controlled actual output value (*y*), measured at the controlled plant (*S*), and compares it with the desired input value (*w*). Resulting difference is an error signal (*e*) (equal to subtraction $e = w - y$), which is converted to a control variable (*u*) by the controller. Controller actuates the plant by the control variable to eliminate, or at least to minimize the error. The controlled value (*y*) can be e.g. a temperature (in a room or inside a process), position or velocity (of moving mechanical parts, lifts, and vehicles), liquid level, pressure, flux, humidity etc.

The objective of feedback control is to regulate or track the desired value behavior in time, with minimal costs and risk. The control can be regarded as a continuous process of error compensation. The error can be caused by desired input value change (e.g. a change of desired temperature in a room by resident's wish, by given time schedule or by detection of person presence).

8.7 Disturbances and control performance

The main source of error is a presence of various disturbances, acting on a plant. They often appears as random variables, e.g. a sudden weather change, room insolation change, window opening, draught, switching of electrical equipment, arrival or departure of several persons, noise in a temperature measurement or random turbulence in its neighbourhood. Also some regular effects are regarded as being random, because they can hardly be predicted by the controller (e.g. a temperature or flux variance of heating or cooling medium).

There are several criteria to evaluate a control performance. Beside the error in steady state, the behavior of controlled value in time is also important. Each error signal change causes a transient response of the output value, which gets settled to the steady state (theoretically) for an infinitely long period. The control performance is evaluated by the transient response as well. In some application, the shortest settling time is desired, even if causing a large overshoot. Otherwise, no overshoot is required, even if the settling will last longer.

8.8 Stability and transient response analysis

The obvious requirement is a stability of control system. Roughly speaking, the controlled output will settle at desired value, in its close neighborhood (tolerance limits), or it will oscillate or fluctuate randomly within tolerance limits. Unstable system exhibits unlimited growth of the error or oscillations with increasing amplitude.

The source of transient response in the control system is a control system dynamics. It manifests itself as inertia, delay, oscillation. It is implied by system ability to accumulate energy or mass (e.g. heat, water, electric charge) or to perform energy transformation from one form to another one (e.g. to exchange static and kinetic energy of a pendulum, electric and magnetic field in circuits with inductors and capacitors).

System dynamics also include a transport delay – e.g. during transport of mass (coal on a belt, water or air in a pipe) or a delay during data transmission and processing (e.g. sound propagation, slow data link communication, computation or slow sampling).

8.9 Static systems

Some systems do not apparently exhibit any intrinsic dynamics, we call them static systems or systems without dynamics. For example, an amplifier or transducer. In closer view, we can see, that these systems do have dynamics as well, and value changes contain transient responses. These transients, however, can be inappreciable to us (e.g. lighting a bulb, switching a contact or thermostat, dial pointer displacement, electric or pneumatic motor run or stop).

Sometimes the transient is apparent, but for particular purpose its dynamics is not significant, and it can be neglected. For example, the process of opening or closing of heater valve or settling of thermometer value is not significant in temperature control, which is a substantially slower process. For a logical control of manipulator, we are concerned in a trajectory, velocity and end points of motion, but the acceleration and deceleration dynamics lies outside of our interest. Similarly, in programming of numerically controlled machining, we are concerned in a trajectory and velocity along the path and we hope that servo motors of slide rests perform well enough. On the other hand, there is a technician, who has to tune up the dynamics of servo motors. As elevator passengers, we feel clearly the system dynamics, but as logical controller programmers, we do not take it into account – we let it up to elevator constructors and servicemen.

8.10 System description

Let us consider a system, consisting of a cylindrical tank, without any outlet, only with a spillover at the top. It is supplied by a water pipe from a source (e.g. water conduit), drawing a constant flux, independent neither on time or tank level. The incoming flux cannot be adjusted, only fully opened or closed.

After opening the input pipe (step change of control variable from zero to a constant value), the water level in the tank will continuously grow, i.e. the growth velocity is constant, independent on time nor current level. The level is determined by flux and bottom plate surface. This behavior continues, until the level reaches the spillover. After that, all superfluous water flows over and the level settles at a constant value (assuming, that all overflowing water is properly drained away). The system thus contains a nonlinearity, consisting in the level limitation (saturation).

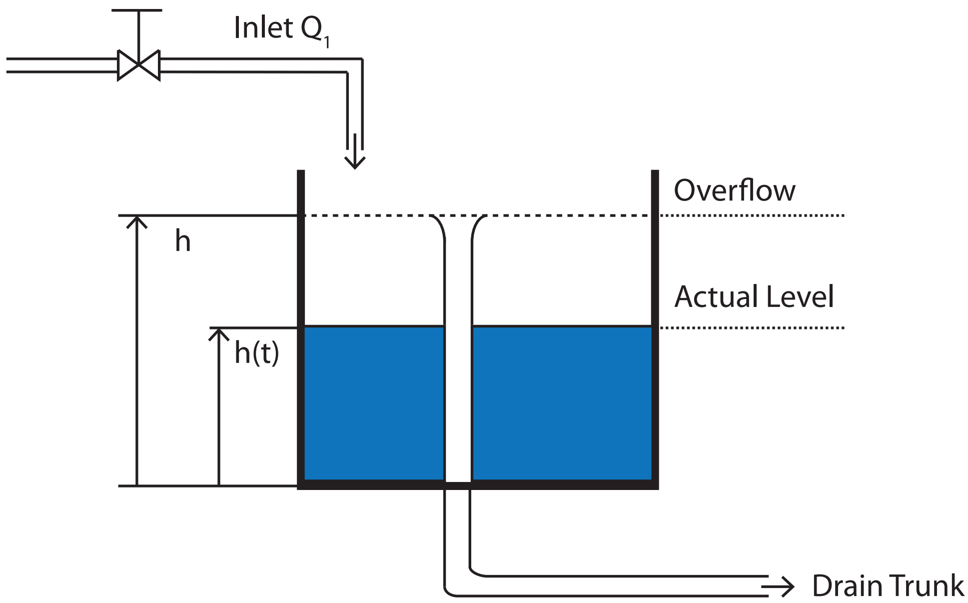
8.11 System characteristics

If the influx is interrupted (pipe closed), the level stops rising and settle at the reached point. After reopening the pipe, the filling will continue the same way.

The level is proportional to filled volume, which is an integral of flux value at the input over time. The system has an integral character; it can be called as a first order integrator with saturation. In control engineering, such system is often called as a system with first order astatism. It means that with constant input value (flux), the output (level) will not settle at any value – there is no steady state value, the system is astatic. This is true, while operating in linear operation range, before tank overflow.

On the other hand, the static system will always have a certain output value, where the system settles, if driven by constant input variable.

Notice: The term "static system" is a slightly ambiguous in the context of control engineering. The first meaning stands for a system without intrinsic dynamics, whose output immediately (without delay) reacts to the input variable changes. The other meaning stands for a dynamical system, where the output settles at a constant value after step change of input variable – such system is a contraposition to astatic system.



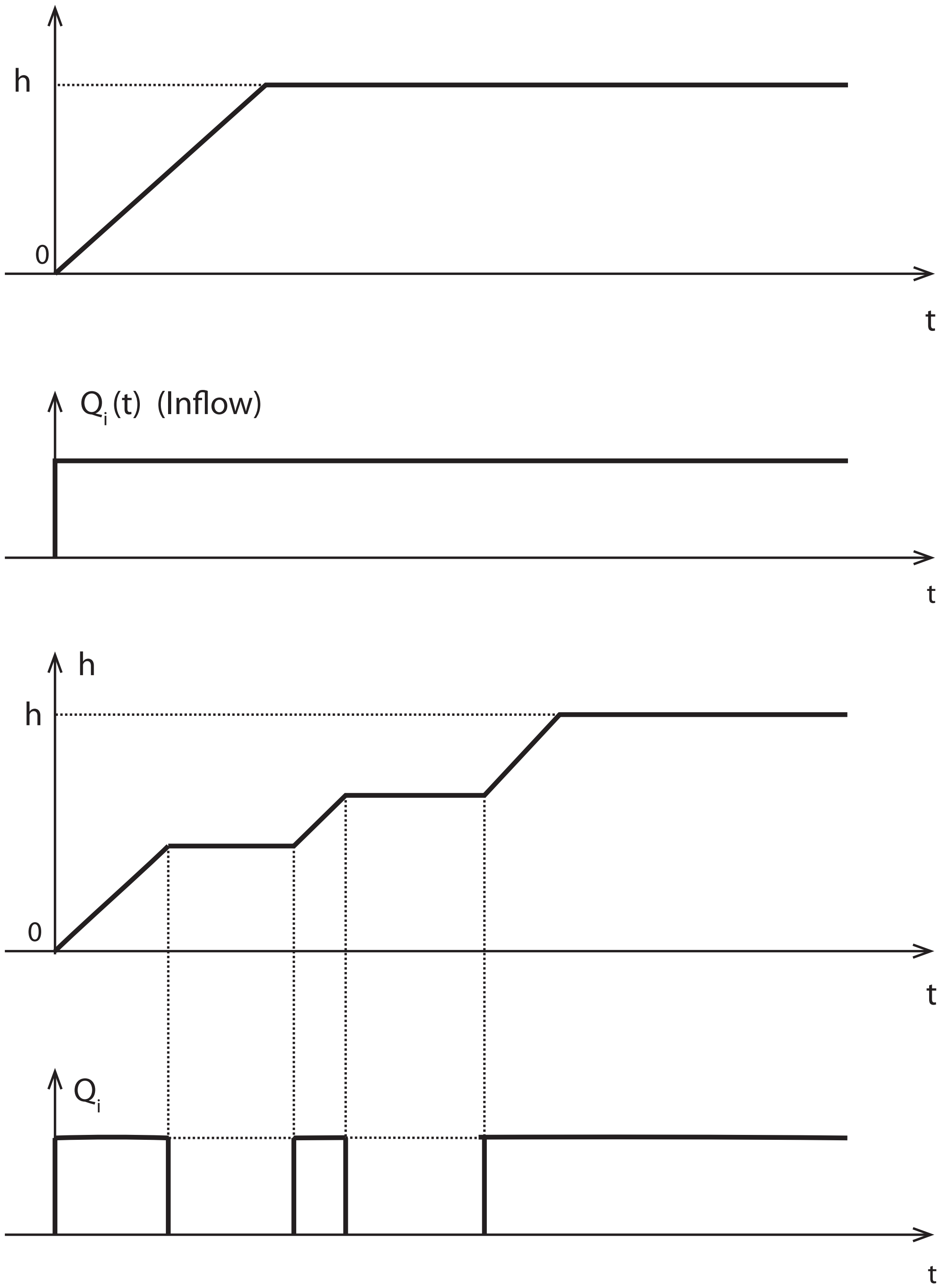
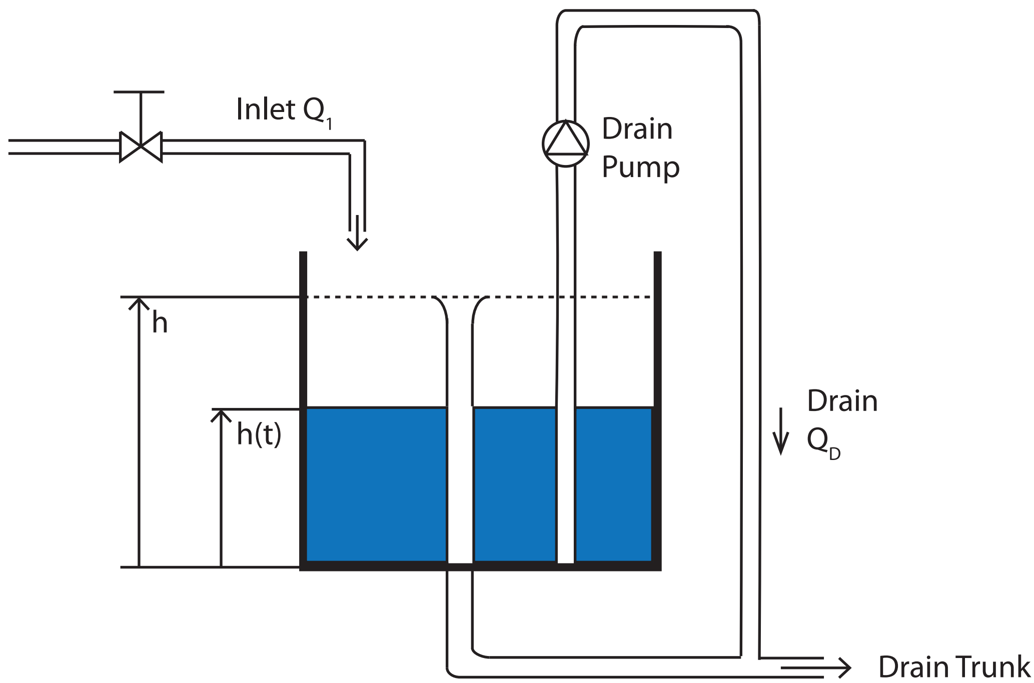


Fig. 8.3: Hydraulic system with integration behavior

The term "static system" is in control technology somewhat confusing because it is used in two senses. The first corresponds to the system without its own dynamics, the output immediately (without delay) responds to changes in input variables. In the second sense, corresponds to dynamic systems, for which the step change of the input variable output also stabilizes at a constant value, but only in the aftermath of transient - so named system is the antithesis of the term astatic system.

8.12 System modification

Disadvantage of presented example is, that there is no way to lower reached level (if evaporization and leakage is neglected) of water in the tank – we cannot set negative flux as a control variable. It disables a control of the level in downward direction.



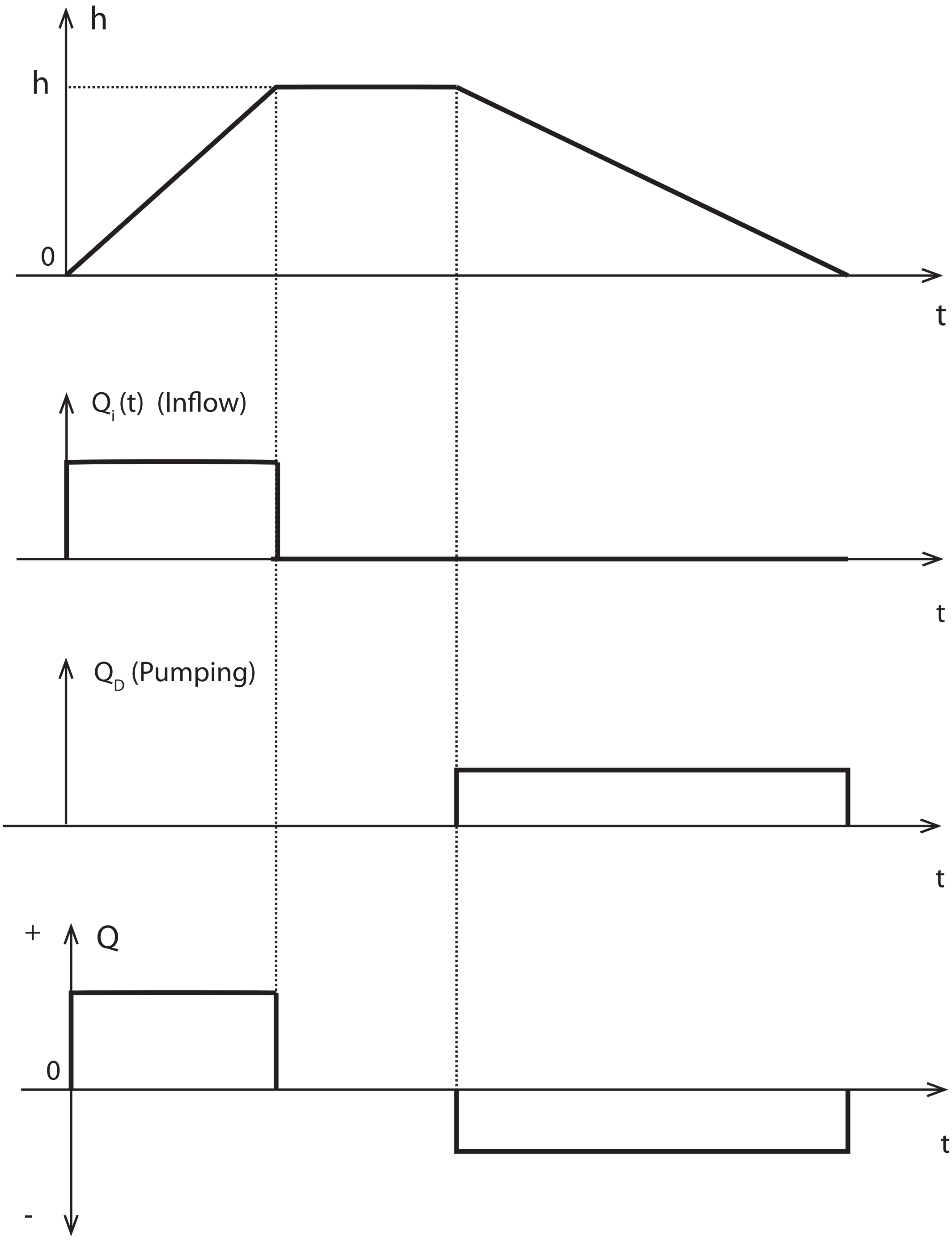


Fig. 8.4: Hydraulic system of integral type

Let us consider an extension of present example by a pump, able to drain water away from the tank. Its suction pipe is connected at the bottom plate. Let us assume constant flux incoming by pipe, as well as a flux drawn by the pump, both switched between zero and constant value. Dynamics of the pump is neglected.

During drainage, a similar behavior to filling is exhibited, only the level is falling down continuously by a constant velocity, until it reaches level of suction pipe, where it stops. It is an integral phenomenon as well, only the input flux (control variable) is a negative value. If the flux of filling and drainage is equal, the system is linear in the operation range, given by limits (saturation) on upper and bottom ends. If there are different values of flux during these two operations, the system becomes nonlinear – it is an asymmetric system with discontinuous variable parameter – the integration constant (different for drainage than for the filling).

8.13 Electric systems

Astatic systems can be found in many domains, e.g. in electrical engineering. On the Figure is shown, where the integration occurs by charging and discharging of capacitor, supplied by a constant current source. During the charging, an electric charge is continuously fed to the capacitor. The charging velocity (current) does not depend on time nor actual capacitor voltage. The capacitor voltage u_2 increases proportionally to the total amount of stored charge. Discharging is an opposite process of decreasing capacitor charge. If the absolute value of charging and discharging current is equal, the system is linear, if these values differ; the system is asymmetric as above, i.e. nonlinear. Theoretically, the capacitor voltage is unlimited up to infinity. In practice, the saturation will apply as well due to voltage limitations of the supply.

Notice: In the circuit theory point of view, this circuit is linear; the only nonlinearity is due to current source limitation, which is, in our particular case, able to supply only two values of output current. It is similar to hydraulic system mentioned above, where the nonlinearity consists of input valves and constant flux. The system properties, like the (non)linearity, can change, if we regard the system only as some "internal subsystem", or if we include actuators and sensors as well, causing additional nonlinearity or dynamics.

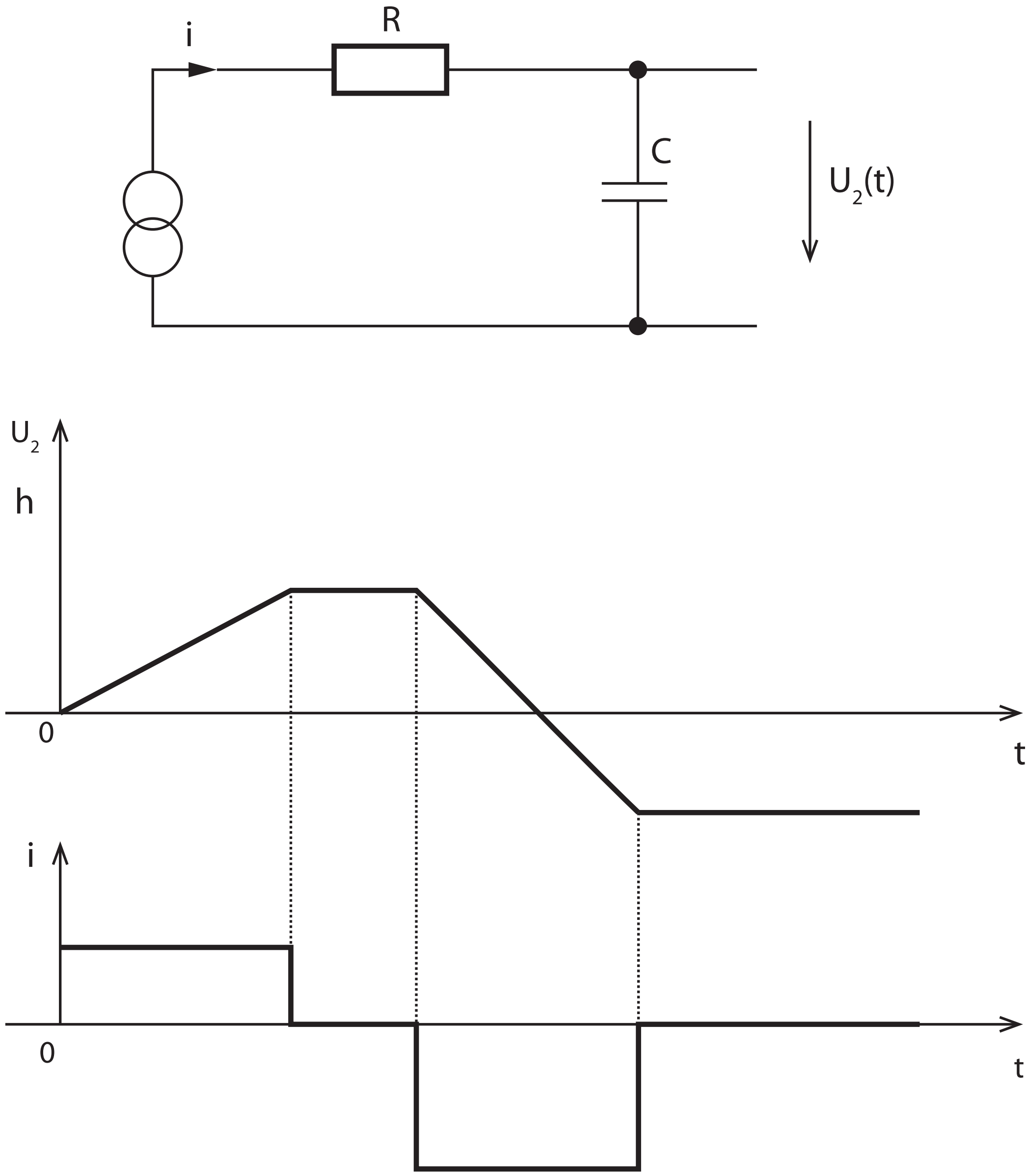


Fig. 8.5: Example of astatic system

8.14 Thermal and mechanical systems

There are thermal systems, which are approximately astatic. They can be approached by a space, insulated thermally as well as possible, with minimum heat losses (polystyrene box, vacuum bottle, experimental chamber with ideal insulation) and with a hard source of heating, independent on temperature difference (e.g. a light bulb, heater, heating without thermal protection).

In mechanics, there are a lot of systems of purely astatic character. All systems, having a linear or angular velocity as an input and position or angle as an output, are astatic, e.g. vehicles, trains, slide rests and turntables of machine tools, movable parts of manipulators, elevators and cranes (heading angle, trolley position or length of wound rope).

The astatism is contained also in systems, where the input is acceleration and output value is a velocity. But if a position is regarded as an output value, while having the acceleration as the input, the system would have second order astatism (consisting in double integration). Free fall is not a common phenomenon in mechanical designs, but systems with constant acceleration and deceleration are often implemented. They can be realized e.g. in elevators, because this kind of take off is comfortable for passengers. At some level of simplification, the takeoff can be regarded as having constant acceleration at some aircraft motors, vehicles, trains or rockets.

For demonstration clarity, only the step changes at the input (valve opening and closing, switching a pump on or off, constant current of directions, constant acceleration or deceleration) are considered. Step changes at the input are also often used in practice in system (plant) identification and model creation, i.e. determination of its structure and parameters from measured response. System properties (e.g. that it is a first order astatic), however, do not depend on an input value time behavior. If we can (and want to) actuate continuous values at the input and measure continuous values at the output, we would observe the very same properties of the system, as if we actuate it by step input values – only the behaviors would be more complicated and less comprehensible. For example, the water level will be still an integral of flux over time, even if the flux will be varied by arbitrary continuous opening and closing of the valve.

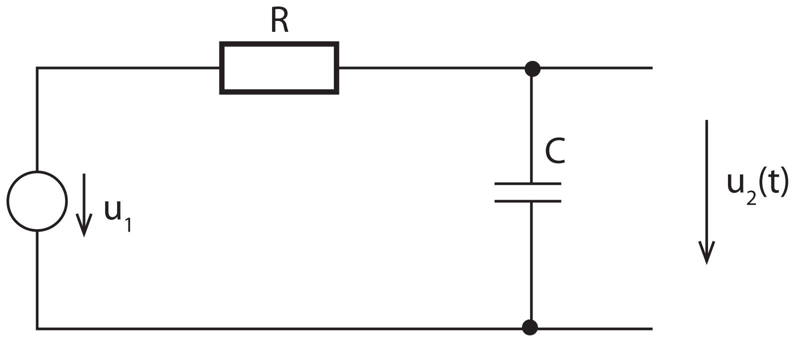
8.15 Capacitor charging

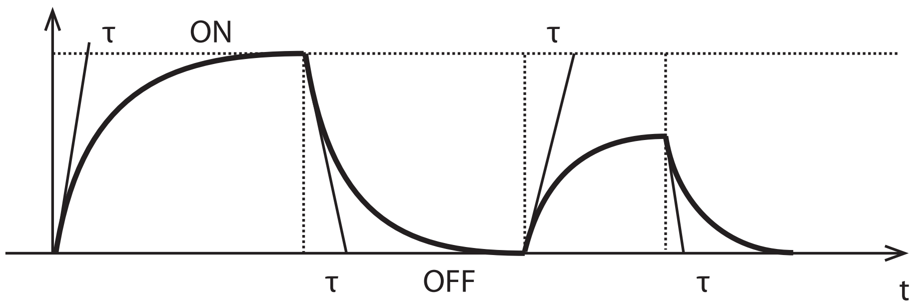
Let us consider modified circuit as shown in picture. Capacitor in series with a resistor is being supplied from a constant voltage source. The response of this system to two input pulses of magnitude u_{1m} and two different lengths (longer and shorter pulse) is plotted on the figure. The output value is a capacitor voltage u_c.

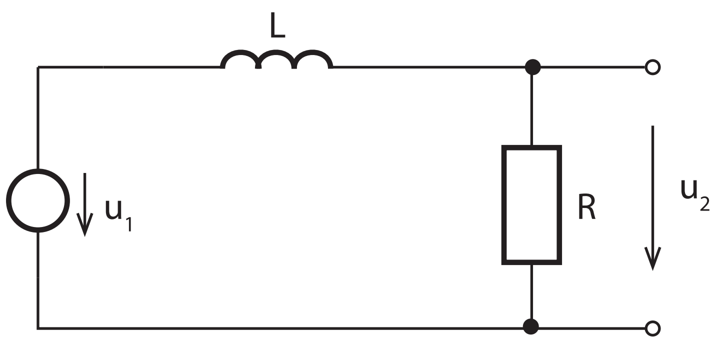
After step of input voltage u_{1m}, the capacitor is being charged (accumulates electric charge) and a its voltage follows an exponential function

$$u_c = u_{1m}(1 - e^{-t/\tau})$$

At the beginning, the capacitor voltage is zero and charging current is maximal (limited by a resistance of resistor R). As the capacitor charges, its voltage increases and difference between it and a voltage at the input ($\underline{u}_{1m} - u_c$), leading to decrease of charging current, and, as a consequence, the increase of capacitor voltage slows down. In steady state, the capacitor voltage (u_c) is equal to the input voltage (u_{1m}) and no current flows through the resistor.







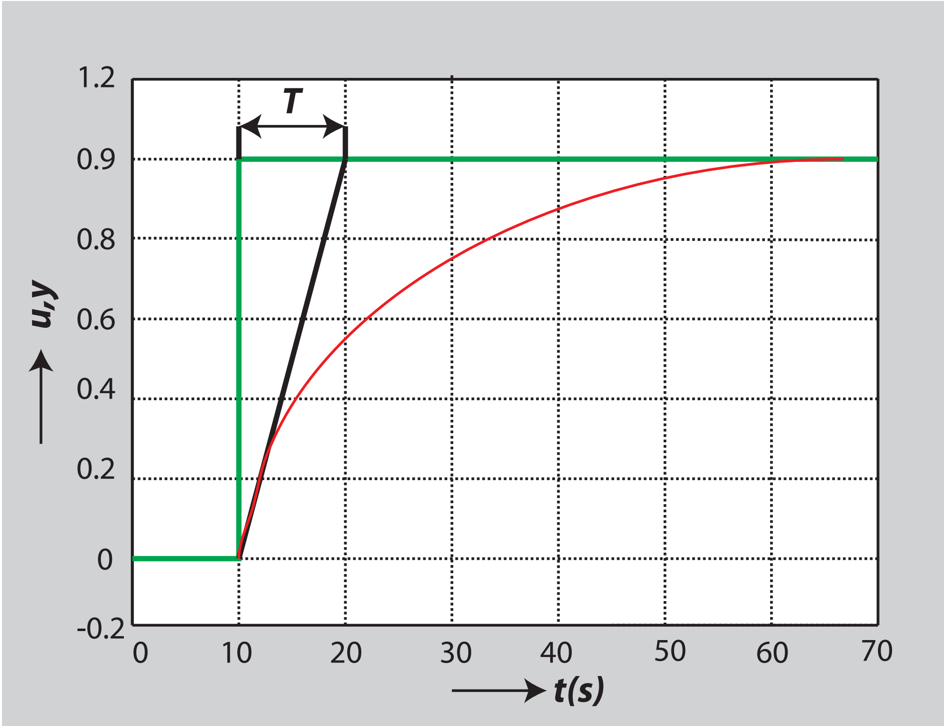


Fig. 8.6: Simple electric circuit

8.16 Capacitor discharging

Similar process comes on during capacitor discharging, i.e. after zero voltage at the input. The behavior in time follows an exponential as well, this time the function is:

$$u_c = u_{1m}(e^{-t/\tau})$$

If a short pulse is actuated at the input, the transient does not finish and the capacitor is partly charged, only. The discharge then commences from the last reached value.

Exponential behavior is parameterized by a time constant τ. A line, tangential to exponential curve, intersecting it at zero time, crosses a time axis at the time τ. For the RC circuit, the time constant is equal to product of capacity and resistance, $\tau = RC$.

The system described above is a first order static system, also called a one capacitor system. Circuit with an inductor and resistor (RL) behaves a similar way – the current passed through the inductor follows the exponential function and the output voltage on the resistor varies proportionally to it.

We assumed, that the input is actuated by a hard voltage source, i.e. there is a voltage equal to u_{1m} or 0 for discharge. The capacitor is discharged by negative current, flowing back to the voltage source. However, in many situations (e.g. in case of tank filling), the input is being disconnected, rather than connected to zero value. In this case, no discharging current is present and a capacitor remains charged forever (if parasitic self-discharging is neglected).

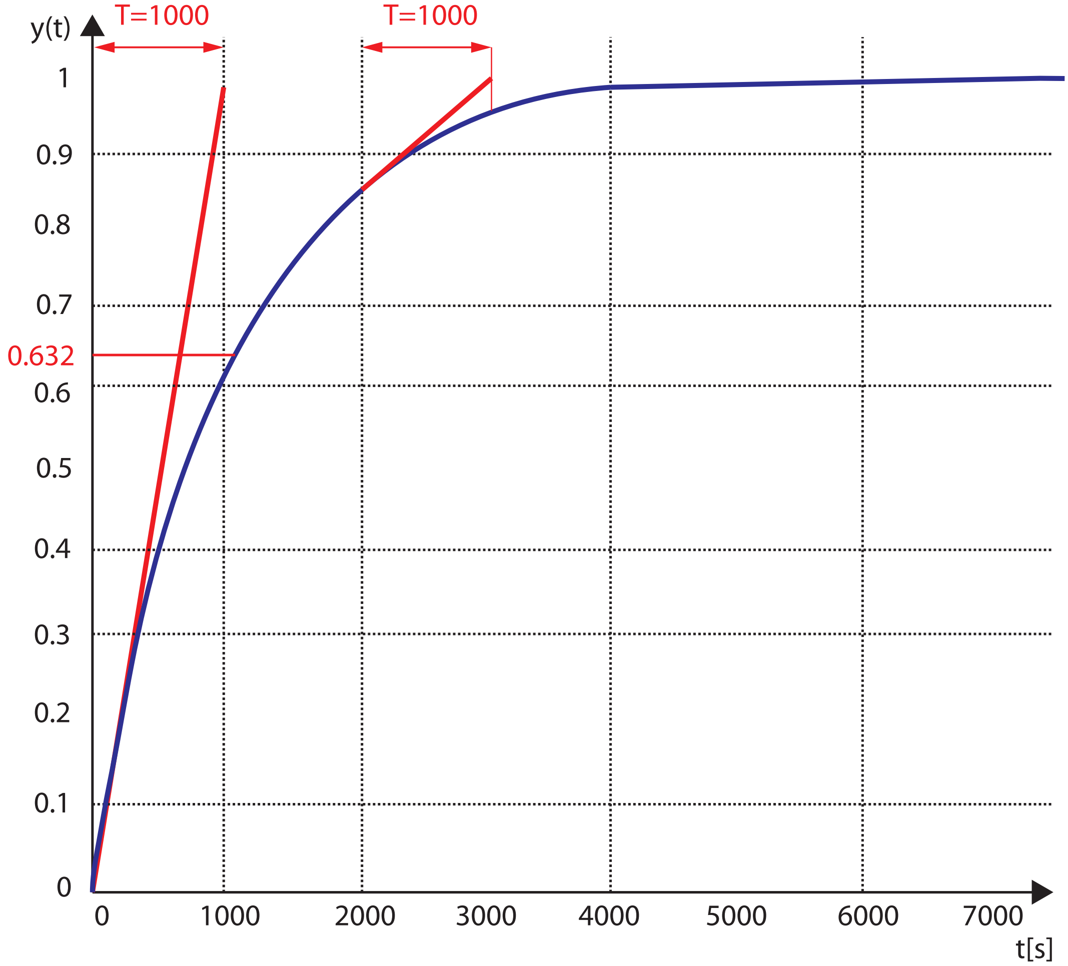


Fig. 8.7: Response static system 1. order to step input and relation to time constant

8.17 Thermal and electric systems relationship

Apparently, there is an analogy between electric circuit described above, and a thermal system, where a room with heat losses is heated, or water boiling in an electric kettle. The difference is, that in electric analogy, a discharging is done by zero voltage at the input, and the capacitor is discharged partly via shunt R_2 and partly via input resistor R_1 back to the source – effectively, it discharges via parallel combination of both resistors,

$R_1 \parallel R_2 = \dfrac{R_1 R_2}{R_1 + R_2}$ to the zero voltage. The system is symmetric, thus linear.

On the other hand, thermal systems are generally significantly asymmetric, thus nonlinear. Interruption of heating represents disconnection of the source, i.e. disconnection of thermal flow between source and system; the system then cooles down only by heating loss. More realistic analogy to the thermal system is thus a tank with simultaneous filling and drainage or an electric circuit with disconnection of supply source, as shown in following Figure. At the end of charging, the current is cut off and the capacitor discharges only through resistor R_2.

This example matches much better the behavior of heating system; however, it is not clear, what the zero voltage (temperature) means. More realistic model is shown on following Figure, which takes arbitrary ambient temperature into account, represented here as second voltage source u_2.

If the voltage of the second source u_2 is lower than capacitor voltage, the current flows back to the source, resembling a heating loss. In contrary, if the ambient temperature is higher than a room temperature, heat flows into the room, what corresponds to capacitor charging from source u_2.

8.18 Relations for thermal, hydraulic and electrical systems

The presented model of thermal system is very simplistic, compared to reality (e.g. a heated room). Real models have more complicated structure and they are usually of higher order. In our model, the resistor R_1 stands for a heating, a thermal resistance of heater, and the resistor R_2 corresponds to heating losses - lower the resistance, greater the loss. Well insulated building typically does have a large resistance values. Capacitance of C represents thermal capacity, i.e. amount of heat, accumulated at certain temperature. It is given by volume of air, partly a walls and heating elements as well.

Electric circuit with two voltage sources corresponds to a hydraulic system with two large basins with different level and it is an analogy of thermal system (we must not forget the square root characteristic of flux on level difference in case of hydraulic system).

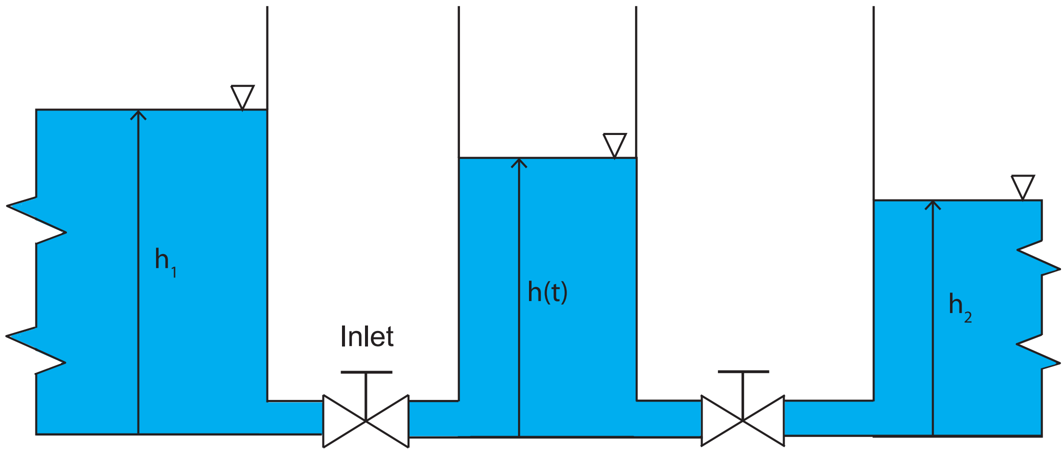


Fig. 8.8: Hydraulic system with two large

8.19 Power sources (current and voltage) for different physical processes

In circuit theory, it is common to work with ideal (i.e. constant) voltage and current sources. It is less common in case of thermal or hydraulic systems, so it is important to understand the real structure while modeling.

The ideal (constant) voltage source gives exactly the same voltage at its terminals, regardless to the load (a current, being drawn) – theoretically, it has a zero internal resistance. Most of common voltage sources in electrical practice approaches the ideal voltage source. Some of them are very close to the ideal (e.g. mains supply, lead accumulators), others are a bit far from the ideal operation (e.g. primary batteries, small transformers).

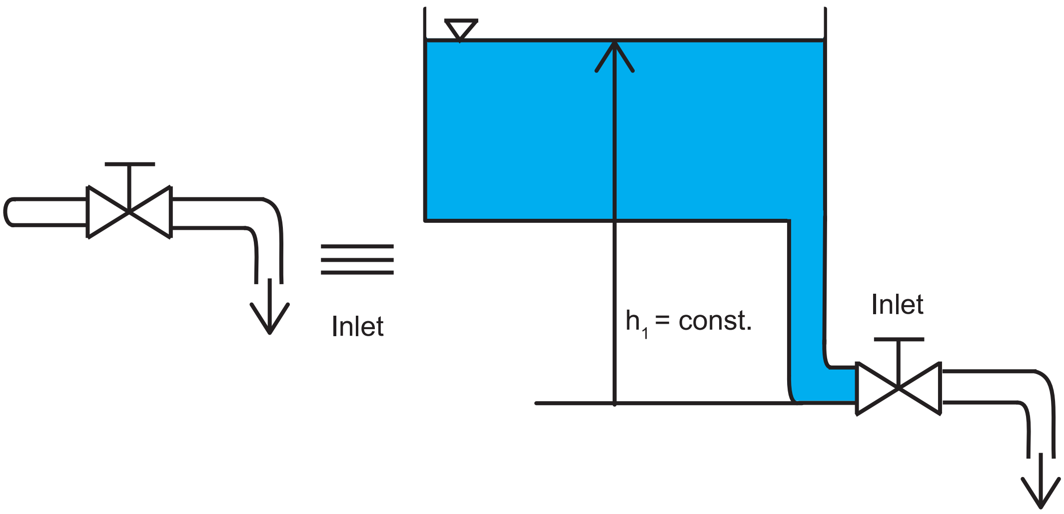


Fig. 8.9: Hydraulic analogies of voltage source

In hydraulic systems, the counterpart of electric voltage is a pressure, or a pressure difference. The voltage source is then represented by a constant pressure source, which is not apparently influenced by load (consumption). It is often realized as a large source (e.g. a basin, reservoir, lake or the ocean or a water conduit).

The voltage is analogous to a temperature or a temperature gradient, in thermal domain. The ideal voltage source corresponds to a massive heating or cooling source, e.g. a huge source of water of certain temperature, boiler, massive wall, ground, river or a well regulated source with sufficient rating.

The ideal (constant) current source is able to drive a circuit through its terminals by constant current, regardless to the load (resistance, causing a voltage drop). It can be approximated by a source of sufficiently high voltage with a large resistance in series.

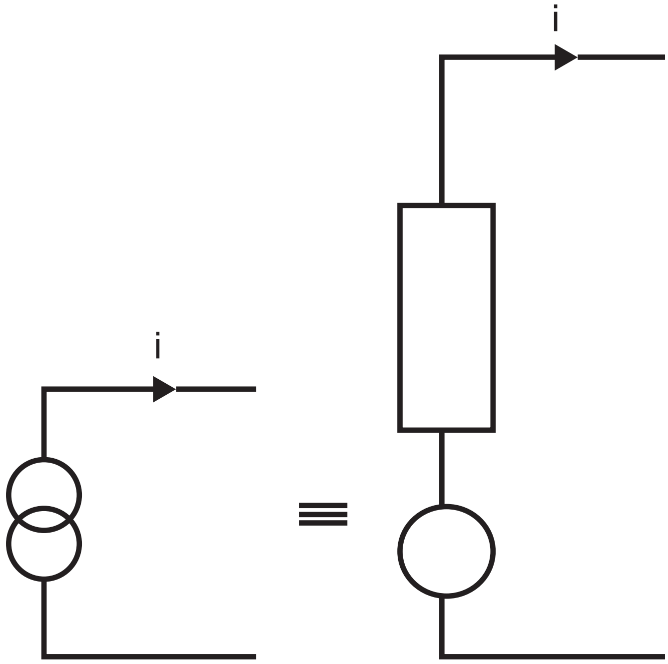


Fig. 8.10: Approximation of constant current source

In hydraulic systems, the current source is analogous to a constant flux source, which is not influenced by consumption, e.g. a water conduit or reservoir with sufficient pressure or a filling without interaction (feedback) of filled liquid (eg. if poured from the top).

The analogy of electric current in thermal systems is a heat flux. The current source corresponds to a source of constant heat flux, independent on a temperature of heated object, e.g. a heater without thermal limitation (protection). Such source (e.g. a light bulb, heating, furnace), if not cooled, can reach a very high temperature, often causing a disaster (firemen can share their experience).

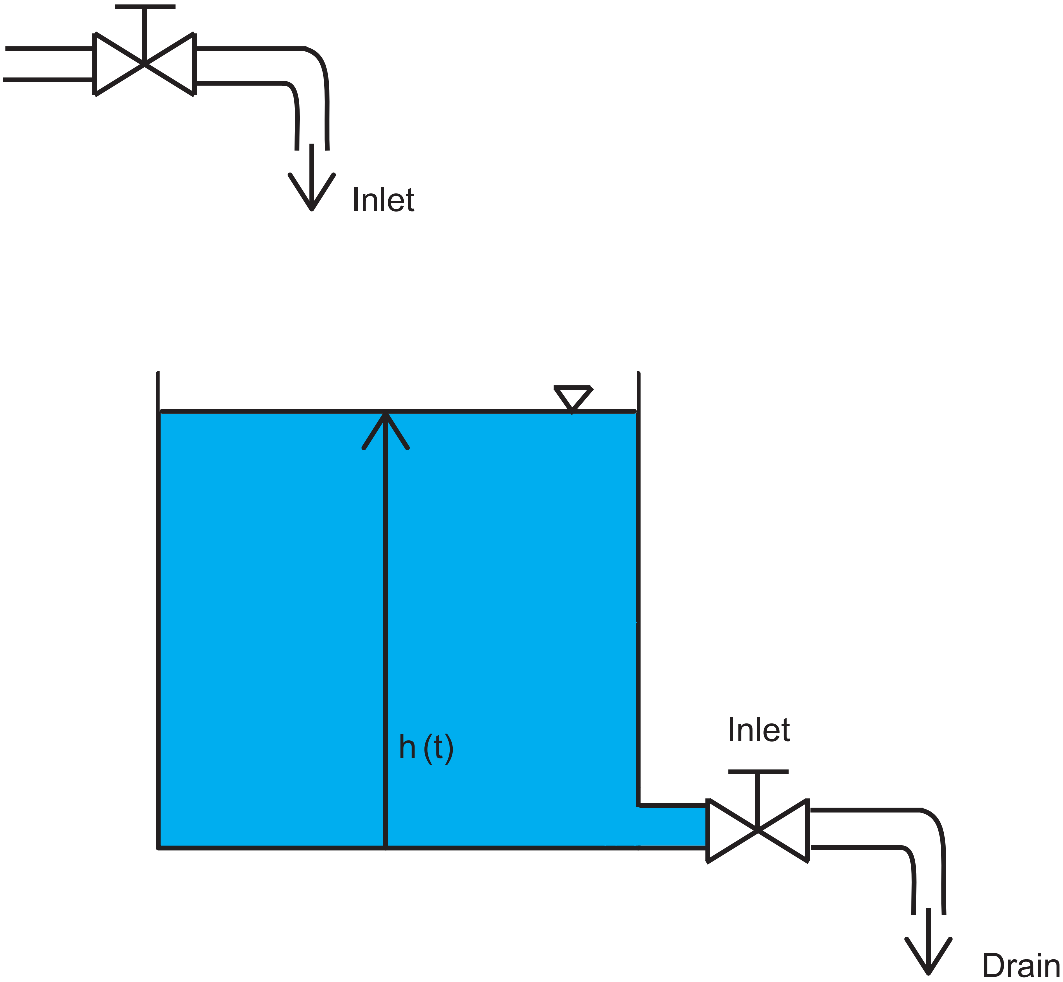


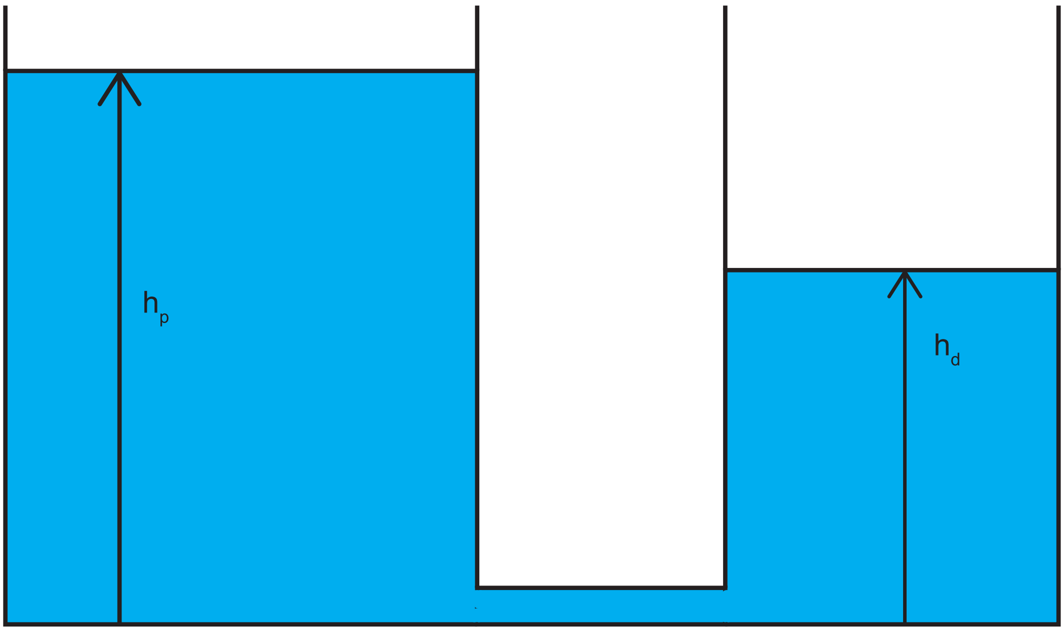
Fig. 8.11: Hydraulic one capacitor system with drainage bore

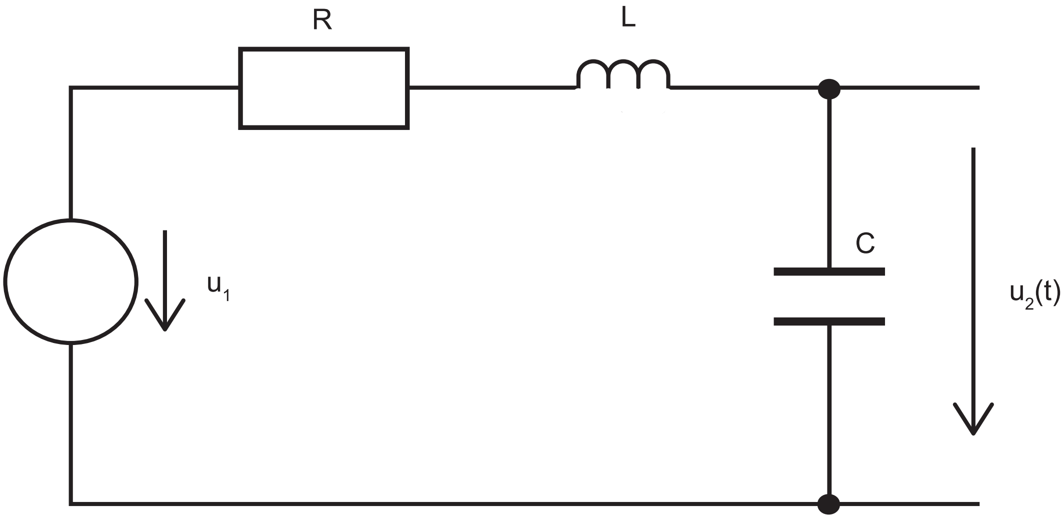
Most of discussed systems (electric and hydraulic), except the astatic one, were supplied from voltage source or its hydraulic counterpart (source of constant pressure). Let us show several modifications of their structure by use of the current source or its hydraulic counterpart (source of constant flux).

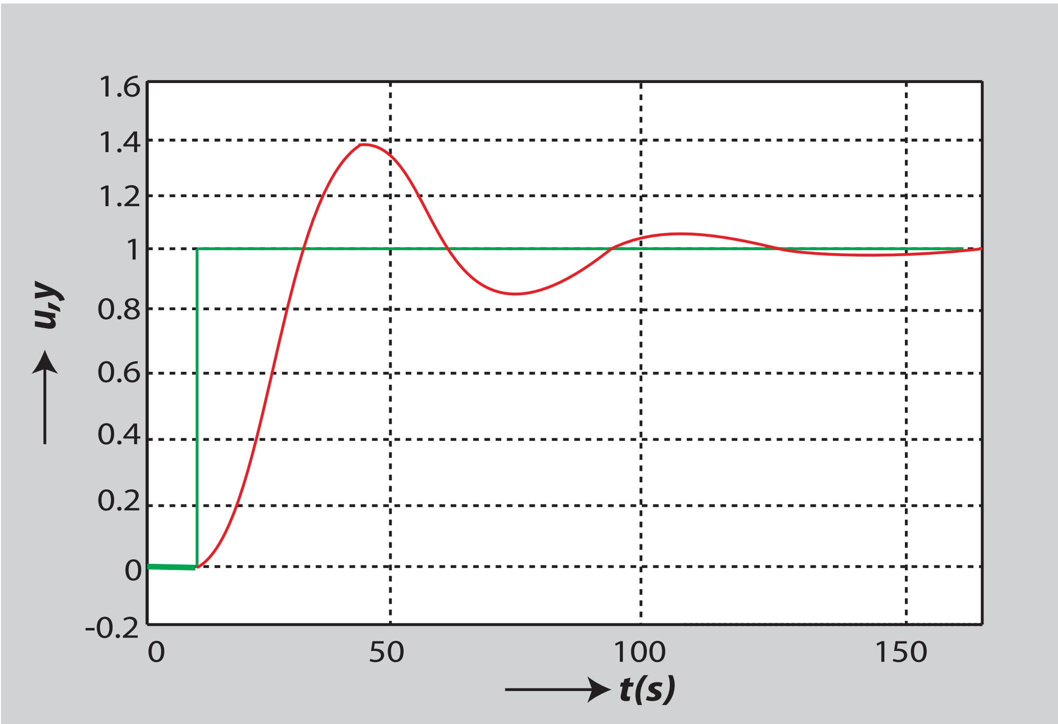
8.20 Oscillatory systems

Higher order systems can have also a non-monotonic transient response. Some kind of systems has typical oscillatory response. The essence of this behavior lies in mutual interchange of energy between different forms (accumulators). E.g. in case of suspended pendulum, a potential energy is transformed to a kinetic energy, and vice versa; in case of torsion pendulum, energy of elastic deformation is transformed to a kinetic energy, and vice versa. The oscillatory characteristic can be found in mechanic systems, where an inertia (mass or momentum), springiness (spring or other reversible deformation) and a damping. Commonly, the goal of control system is to increase the damping (e.g. active damping of carriage or driver's seat, crane motion optimization with respect to cancel oscillations). Electric circuits, composed of inductors, capacitors and resistors are also susceptible to oscillation. Large water areas are usually also oscillatory. Oscillation, even unstable, is often caused by a feedback in complex systems.

Series connection of subsystems can be composed not only of static systems, but of the astatic or a combination of static and astatic as well.




R
L
C
u1
u2(t)


1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
u,y
0
50
100
150
t(s)

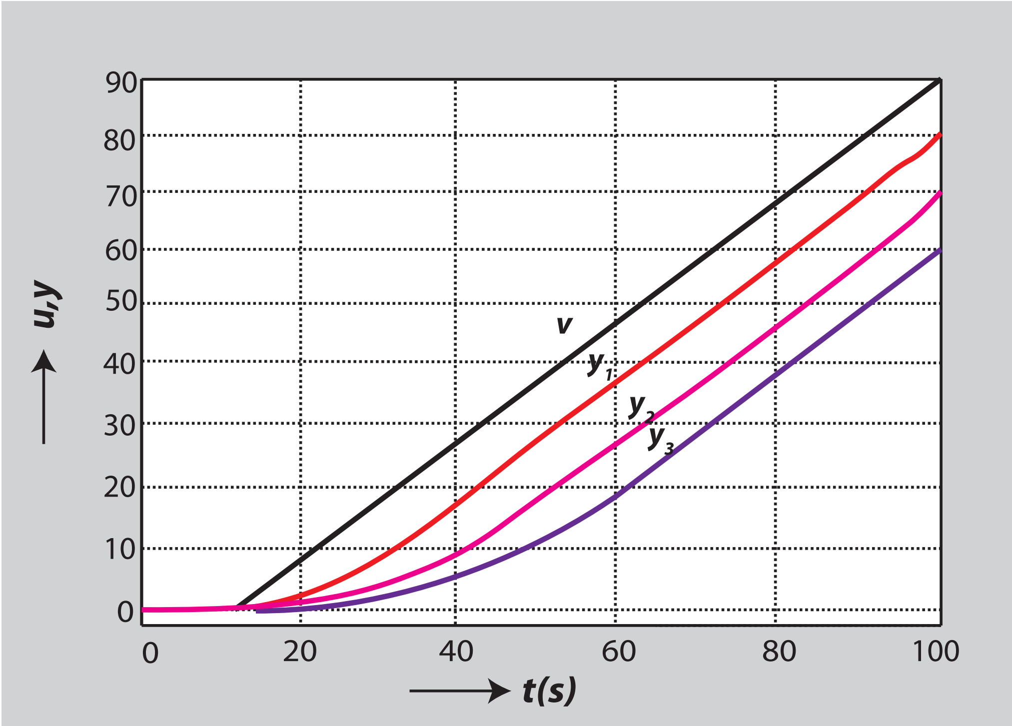
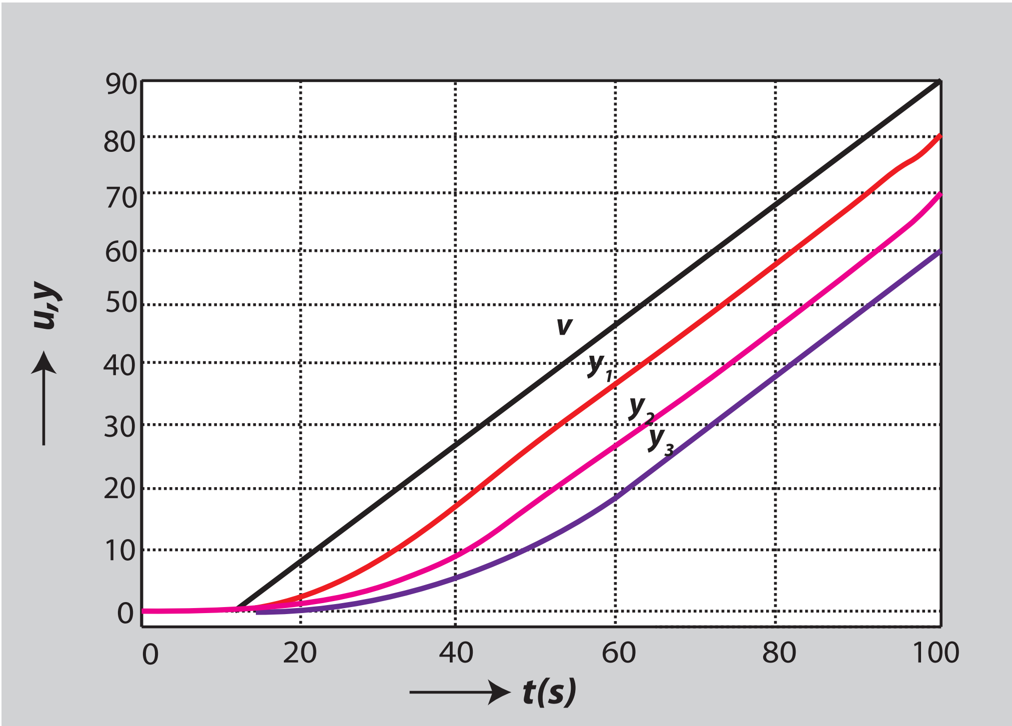
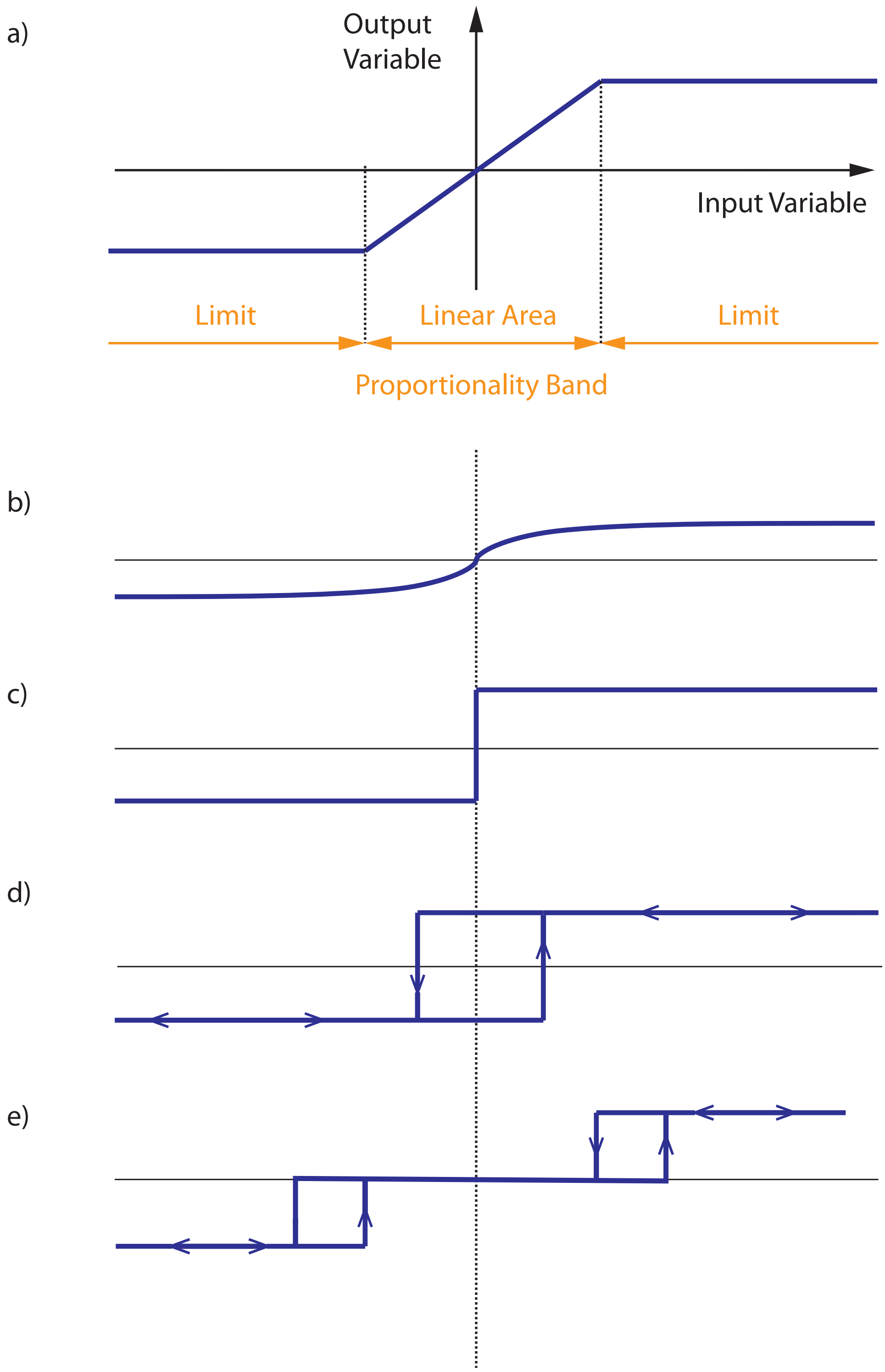


Fig. 8.12: Electric circuit of RLC type

8.21 Nonlinearity

Systems can contain significant nonlinearities. Some of them were presented in previous text, such as limitation (saturation), system asymmetry or a general function dependency (e.g. quadratic, sigmoidal or square root). Other examples are a relay characteristic, hysteresis, multiple hysteresis, dead zone, Coulomb friction, etc. The characteristic of saturation is very common at actuators (controller output) – for example, a controller computes desired heating water temperature to 250 °C, but physical limit is up to 100 °C, e.g. 80 °C. In that case, an output value is clamped to the limit. The system can contain multiple nonlinearities, e.g. the asymetry together with square root dependency in case of hydraulic system. Another nonlinearities in hydraulic systems can be caused e.g. by varying tank shape.




a)
Output
Variable
Input Variable
Limit
Linear Area
Limit
Proportionality Band
b)
c)
d)
e)

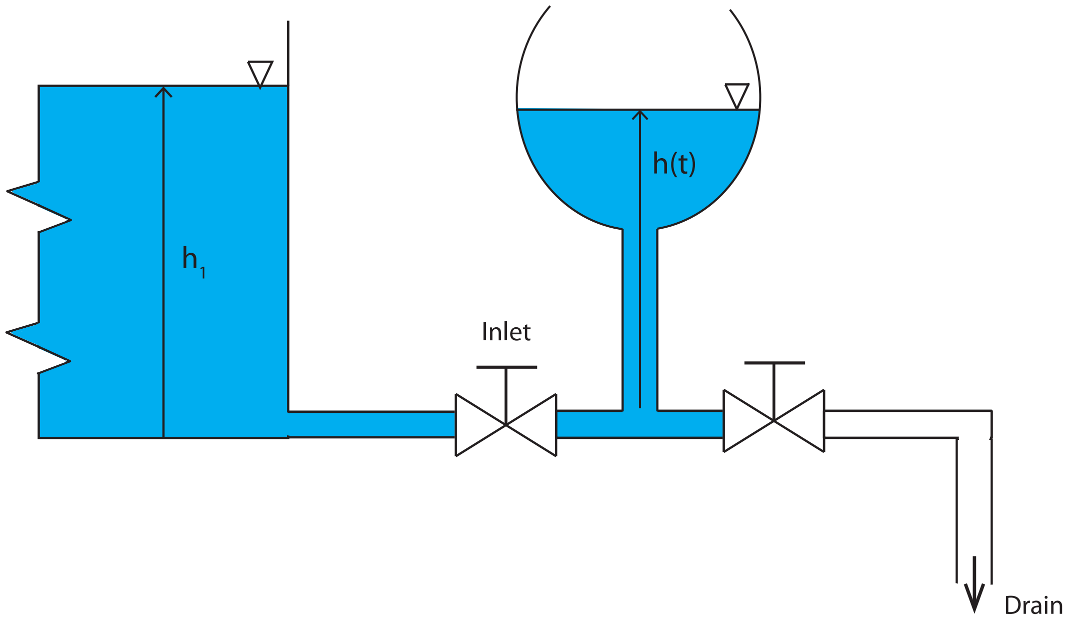


Fig. 8.13: Characterics for typical

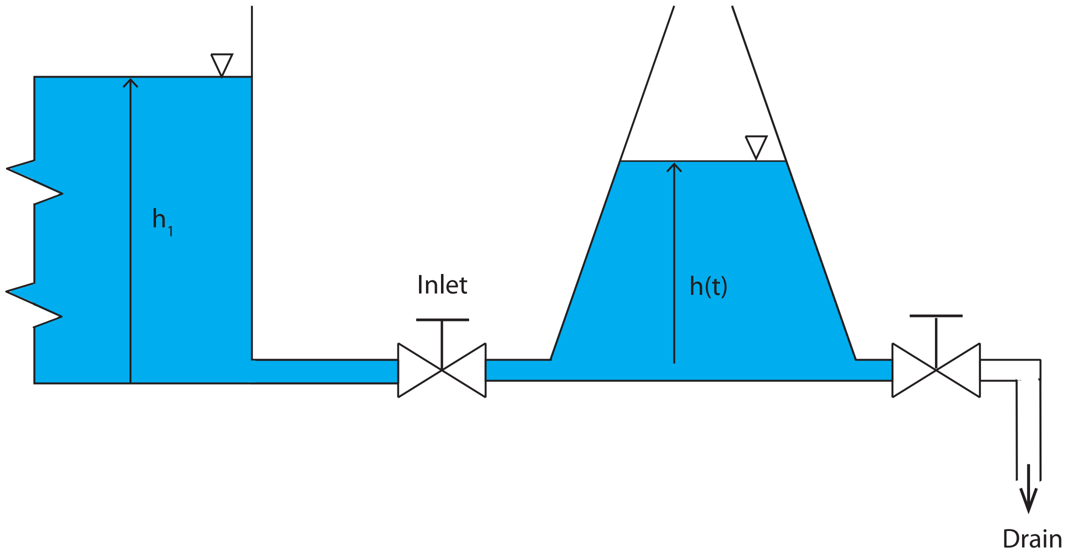


Fig. 8.14: Hydraulic nonlinear one capacitor system

8.22 System identification

System identification is usually done by actuation of the system by shaped input signal, and evaluation of its response at the output. Then, a structure (type, order) of the system and parameters (coefficients) are determined. Often, a step input is considered, the output is then a transient response, discussed earlier. Sometimes, a very short pulse is introduced, then the response is called impulse response. Both signals (and responses) mentioned have a significant theoretical meaning. However, such sudden changes can be impossible or dangerous to use in certain systems, in practice (e.g. the sudden opening or closing of a valve at a long pipe can cause dangerous blasts, also a very fast acceleration of vehicle, elevator or a cableway can be unwanted). Sudden changes are often impossible, due to slow action, e.g. of motor, valve or a heating. So called ramp or trapezoidal signals are then used instead of step and impulse at the input. For their theoretical significance, also a sinusoidal or random (continuous or pulse) signals are used.

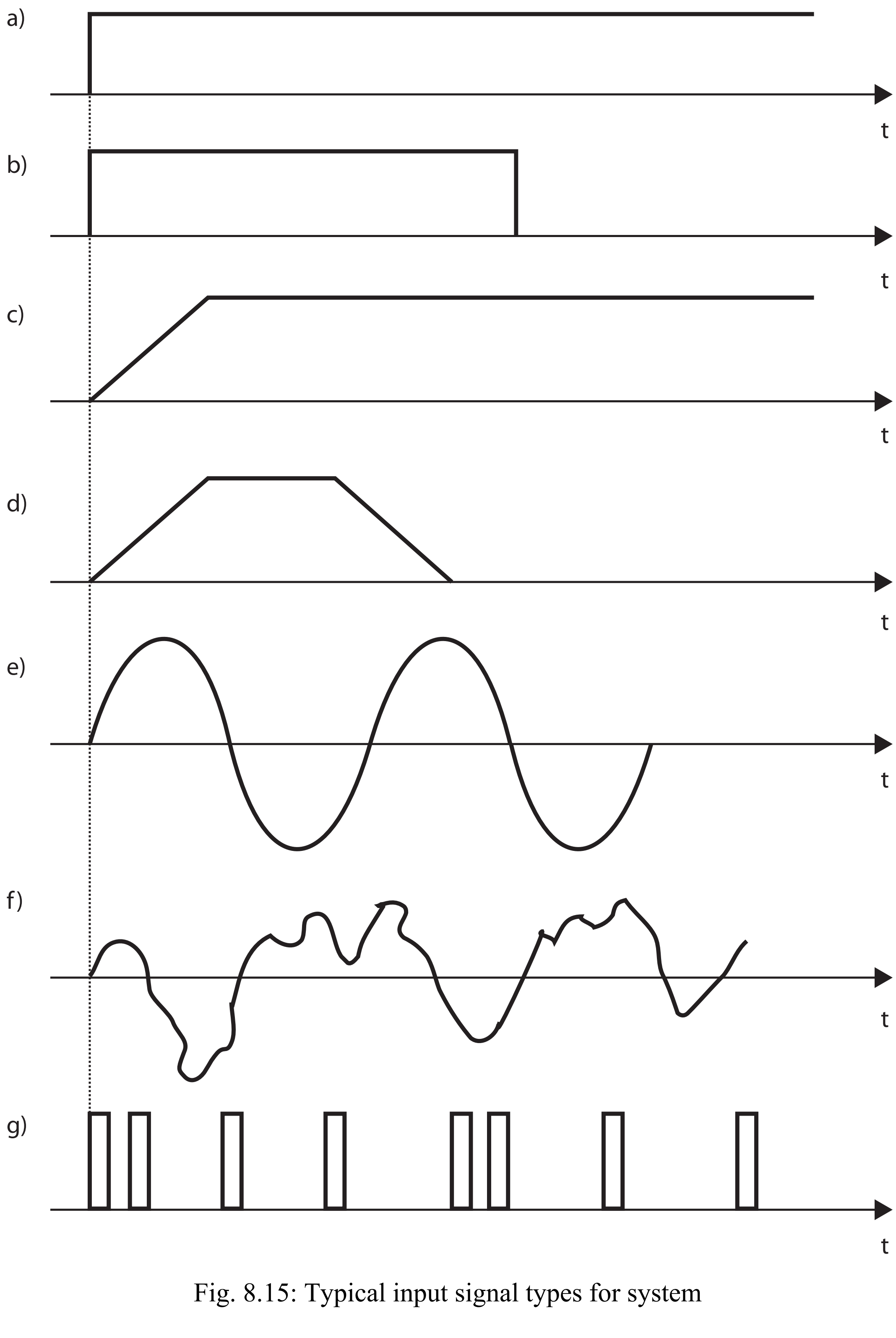


Fig. 8.15: Typical input signal types for system

8.23 Control

Control is a process, using a feedback to reach desired goal (feedback control). The goal of the control is to reach and ensure desired value of controlled (output) value (e.g. room temperature, tank level) or the desired time behavior (e.g. temperature behavior according to weekly plan or a temperature in chemical reactor according to specifications). The desired value must be assured not only after desired value change, but under disturbances, acting to a system, as well. The disturbances typically do have an unpredictable characteristic, e.g. thermal loss or increase in heated room (outdoor temperature change, window opening, draught, wall and room insolation, presence of persons or powered electric equipment).

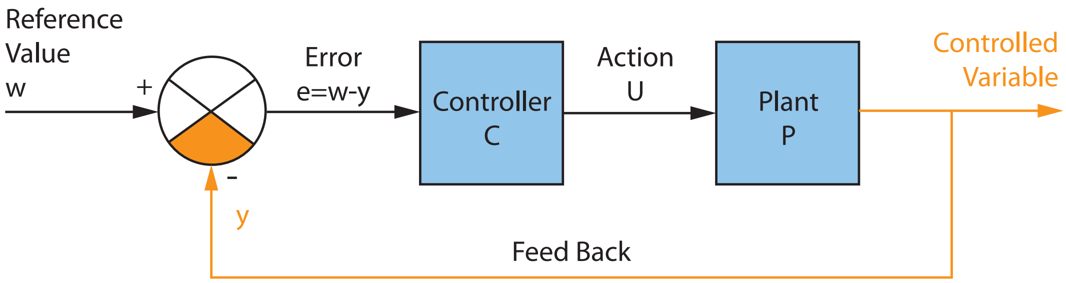


Fig. 8.16: Control system setup

8.24 Feedback

Principal schematic of feedback control system is shown on the Figure above. The input of the whole system is a desired value (*w*) and an output is the actual value (*y*). The subtraction element evaluates an error $e = w - y$, what is input to a controller R. The controller processes the error and outputs a control variable (*u*), which acts via actuators to the controlled system (plant), S. The controller tries to minimize the error, for the actual value y to approach the desired *w*.

In practice, often a discontinuous controllers are used, e.g. a two step (thermostat), three or more step, various variants of PID controllers (PI, P, PII2). There are also several types of much more complex controllers.

9 PID controllers

The common property of ordinary P, PI and PID controllers is linearity. In case of proportional controller (P), the control variable u is directly proportional to the error e.

The control variable of proportional-integral controller (PI) is a sum of two components – the proportional one (which is, as well as in case of pure P controller, directly proportional to the error), and an integral one, which is proportional to accumulated value of the error, i.e. to its integral. Integral action is able to reach zero error in some cases, where it is impossible with pure proportional controller, e.g. if Couloumb friction is present. It is of inertial character, sometimes it can cause unstable or oscillatory response of the control system.

The output (control variable) of proportional-integral-derivative controller (PID) contains an additional derivative action. It has "anticipating" behavior and brings faster response to sudden changes. Its disadvantage is that it amplifies high frequency noise, present in measurement, what may cause random, erratic operation of the system.

9.1 Implementation

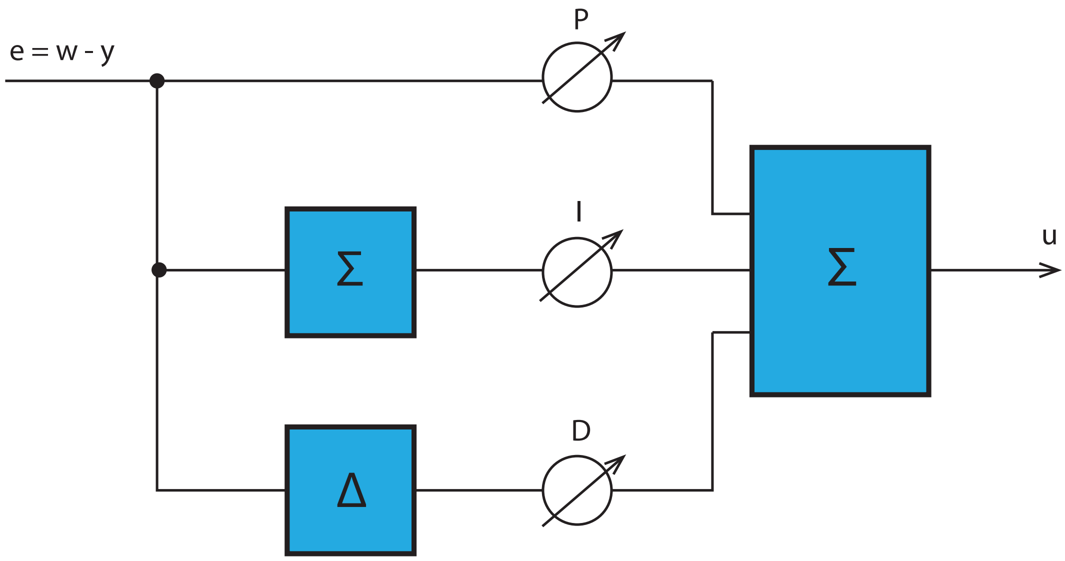
Until recently, PID controllers were implemented as analogue circuits, usually based on operational amplifiers.

Presently, the controllers are usually implemented in software. The software can run on a microcontroller, digital signal processor or a PLC in case of industrial application, or an ordinary personal computer.

Evaluates mathematical expression:

$$u_k = \mathrm{P} \cdot e_k + \mathrm{I} \cdot \sum e_k + \mathrm{D} \cdot \Delta e_k$$

Integral of the error is replaced by sequential sum of individual error value samples at each step ($\sum e_k = \sum e_{k-1} + e_k$). Derivation is replaced by a backward difference, i.e. the difference between actual and previous error sample $\Delta e_k = e_k - e_{k-1}$.



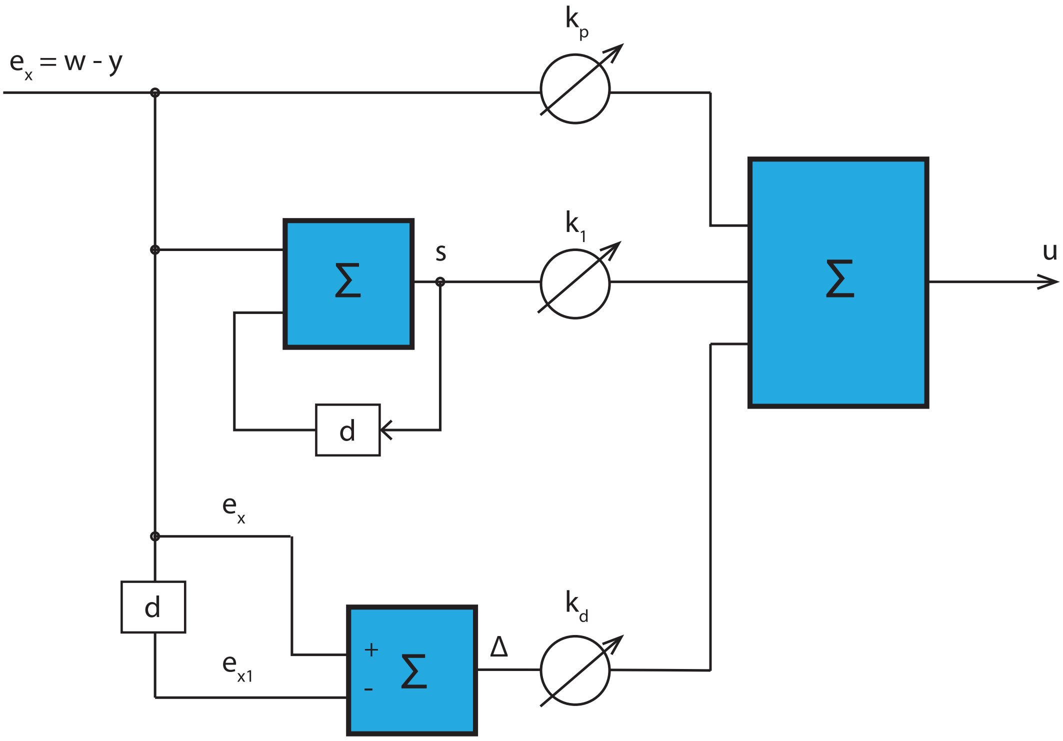


Fig. 9.17: Principial schematics of positional (absolute) PID

9.2 Variants

Although the later controller is not an exact equivalent of analogue PID controller, it is able to replace or even supersede it in most applications. It is sometimes called a PSD controller (proportional-summation-difference). In this text, we will call the digital implementation a PID as well. Output of the program is the control variable u_k (e.g. a position of actuated mechanic part, a valve displacement). Corresponding algorithm is called positional or absolute. In theory, various coefficients are used to define controller adjustment, here we will use simply P, I, D standing for gains of particular components.

The program, implementing this algorithm, computes control variable u at each step of execution. The term positional or absolute is used to distinguish it from another PID algorithm derivative, which computes increments of control variable Δu, instead. The later one is called a velocity or incremental algorithm. It can be regarded as the absolute algorithm followed by a differentiator. The actual value of u can be summed inherently by the actuator, e.g. in case of stepper motor, fed by number of steps Δu.

9.3 Incremental algorithm

The incremental algorithm is given by formal modification of the absolute one, according to following expression:

$\Delta u_k = \mathrm{P} \cdot \Delta e_k + \mathrm{I} \cdot e_k + \mathrm{D} \cdot \Delta 2 e_k$

Output of the algorithm is a control variable increment, Δu_k. Integral component is proportional to current error e_k , proportional component is proportional to its first difference $\Delta e_k = ek - e_{k-1}$ and a derivative component is proportional to a second difference, $\Delta 2e_k = \Delta e_k - \Delta e_{k-1} = e_k - 2e_{k-1} + e_{k-2}$. The increment Δu_k represents a velocity of control variable u_k change. Thus, the algorithm is often called a *velocity* or *incremental*. Absolute value of u_k can be computed by cumulative sum of the increments, $u_k = u_{k-1} + \Delta u_k$). Lettres P, I, D represents particular gains (amplifications) of respective components.

After formal modification, the incremental algorithm can be simplified to:

$\Delta u_k = q_0 \cdot e_k + q_1 \cdot e_{k-1} + q_2 \cdot e_{k-2}$

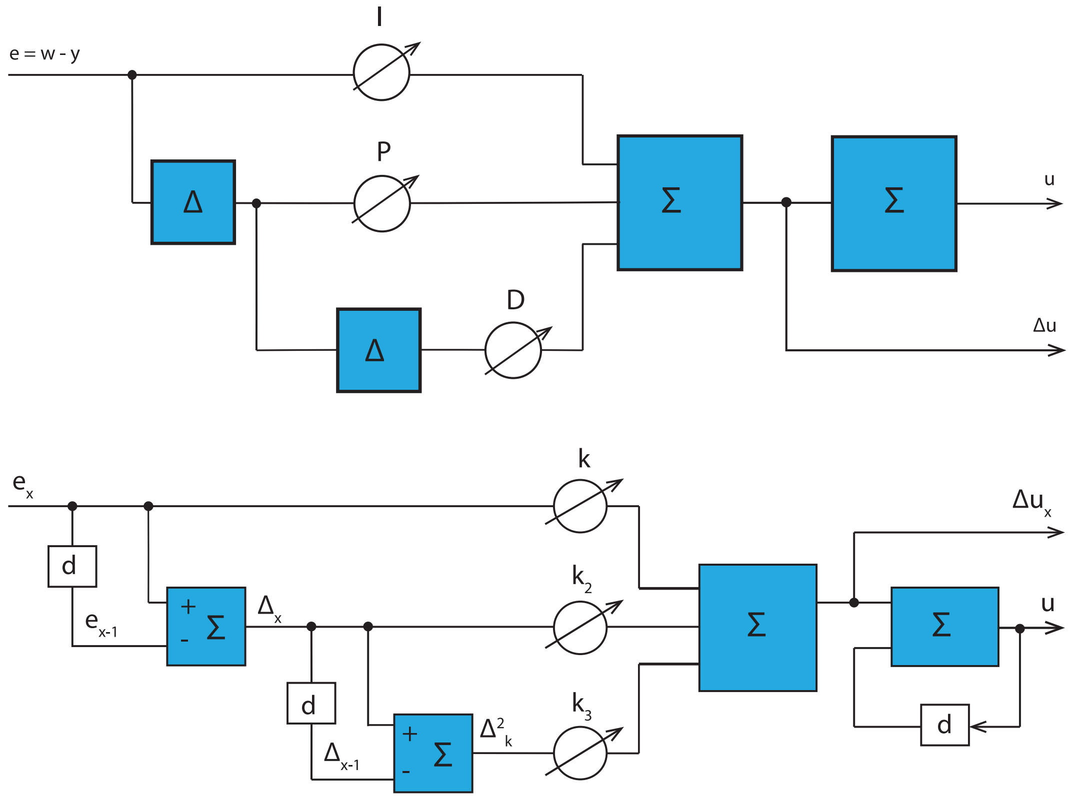


Fig. 8.18: Principial schematics of velocity (incremental) PID algorithm

9.4 FIR system relationship

Control variable increment is a linear combination of three samples of the error – current ek, previous e_{k-1} and even one before e_{k-2}. The increment is a linear combination of values in three element delay line. Such system is an example of a FIR system. The algorithm contains only three multiplications and two additions, coefficients q_0, q_1, q_2 can be precomputed once from given coefficients P, I, D (gains).

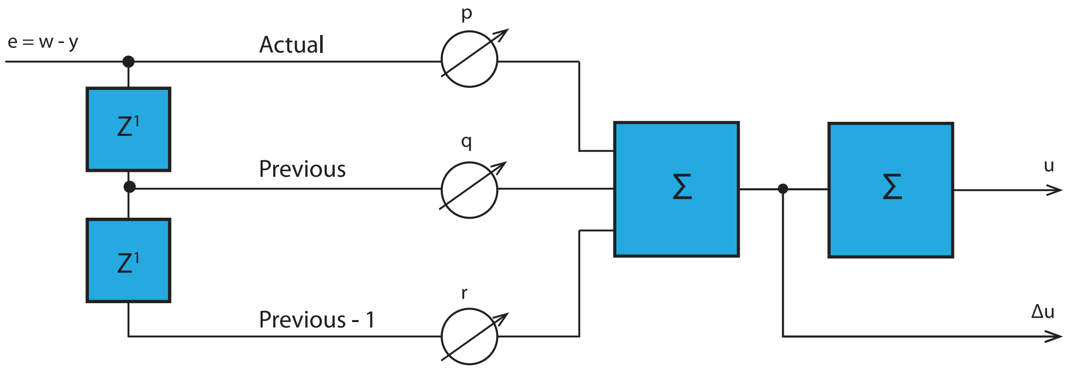


Fig. 8.19: Incremental PID controller as a FIR system

9.5 PI, PII variants

In some applications, two PI controllers cascaded (output of first controller is connected to input of the second), so the resulting system performs double integration. The controller with double integration is usually called a PII or PI2.

As shown above, the actual software implementation of PID controller is not difficult. Of course, in real life application, much more complicated structures of PID control are used. Many PLC offer dedicated instructions of functional blocks for PID control.

The main challenge is to adjust controller parameters (coefficients), to obtain control performance as good as possible (stable, fast, but without unwanted overshoot, with minimal energy consumption and cautious about controlled process and actuators).

The control design is determined mainly by properties of controlled plant and its identification is a basic requirement to follow design methodology, often described in literature. However, this is often difficult to do. The other theoretical requirements of linear control design are that the plant should be a linear and time invariant system. Due to limited scope of this text, we will only show some illustrative examples.

9.6 P variant

The principle of plain P controller is simple: "greater the actual error *e* (greater the difference between actual value *y* and desired value *w*), greater the actuated control variable *u*". Transient responses (responses to a step of desired value) of a system, controlled by P controller, are shown on the Figure.

As shown on the behavior, the output value y never settles exactly at the value of w (the error *e* remains nonzero – so called steady state error). Greater the gain, smaller the error in steady state, but at the cost of augmented overshoot and oscillations. For higher values of proportional gain, the control system can even become unstable.

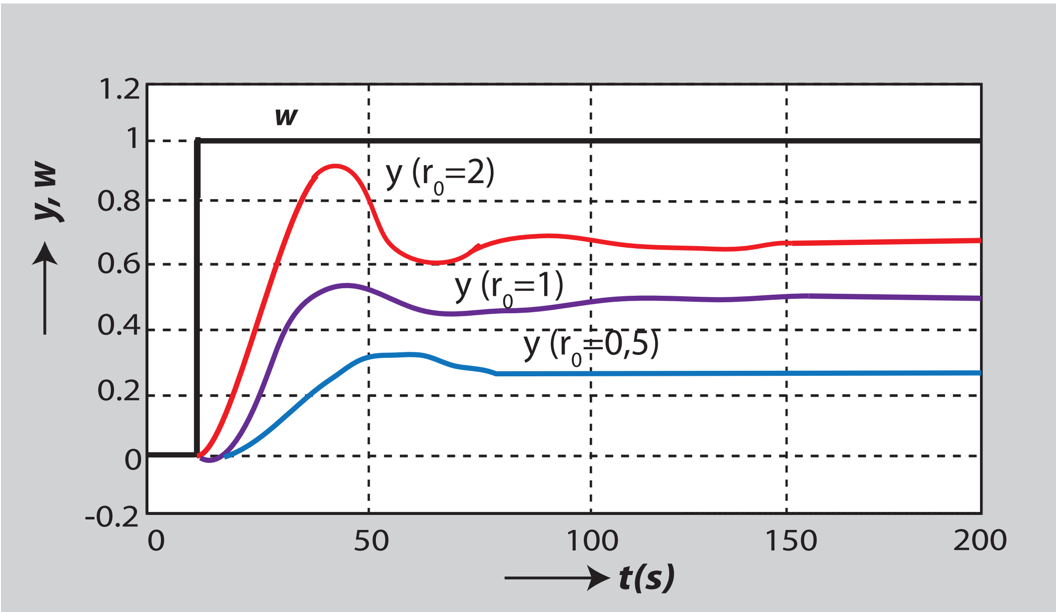


Fig. 8.20: Transient response of system with proportional controller with different gains (r0)

9.7 PI variant

The disadvantage of nonzero steady state error can be eliminated by use of proportional-integral controller (PI). Transient response of system containing the PI controller is shown on the Figure, along with behavior of control variable *u*.

The response of the control system is different for different controller parameters (coefficients), as well as for different plant properties (structure, parameters). Often, plants, whose parameters vary a much, must be considered. One way to solve this problem is to use an adaptive controller, or a controller, which is able to perform some subset of system identification and to adjust its own coefficients (auto tuning). The other possibility is to use a robust control design methodology, which is able to assure stability for some range of varying plant parameters.

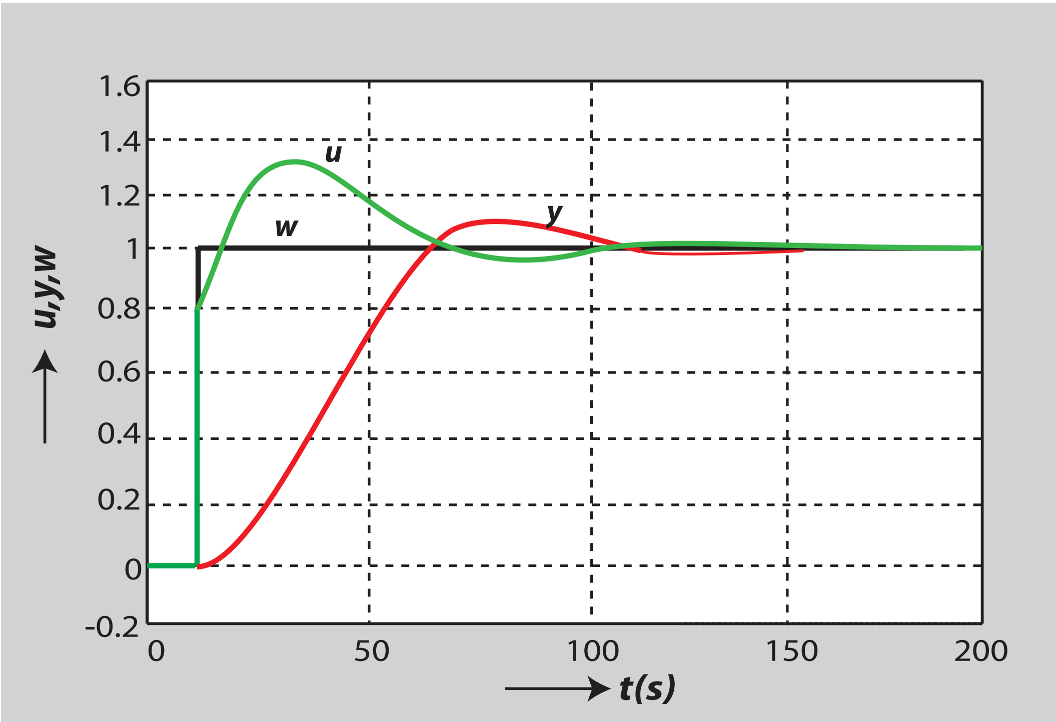


Fig. 8.21: Transient response for 3 different plants

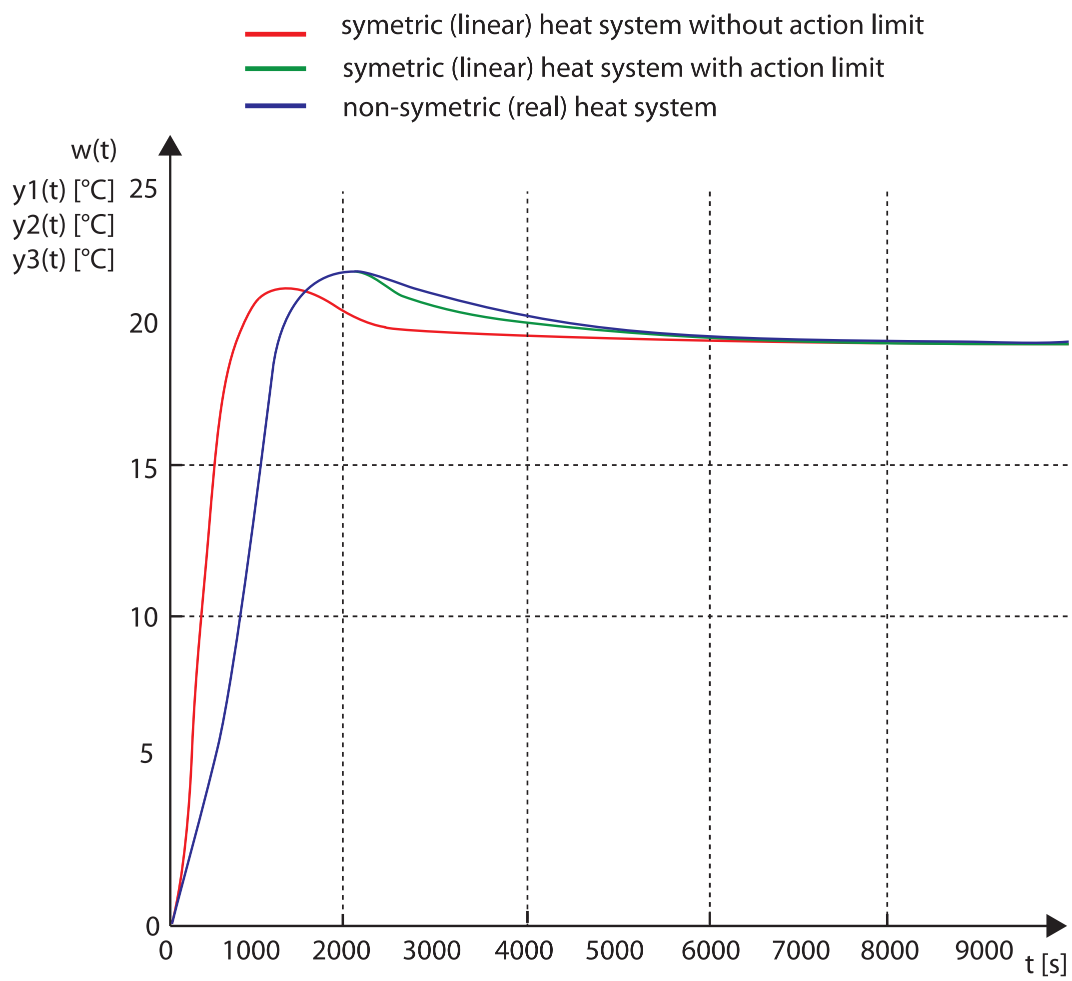


Fig. 8.22: Trensient response - PID controler

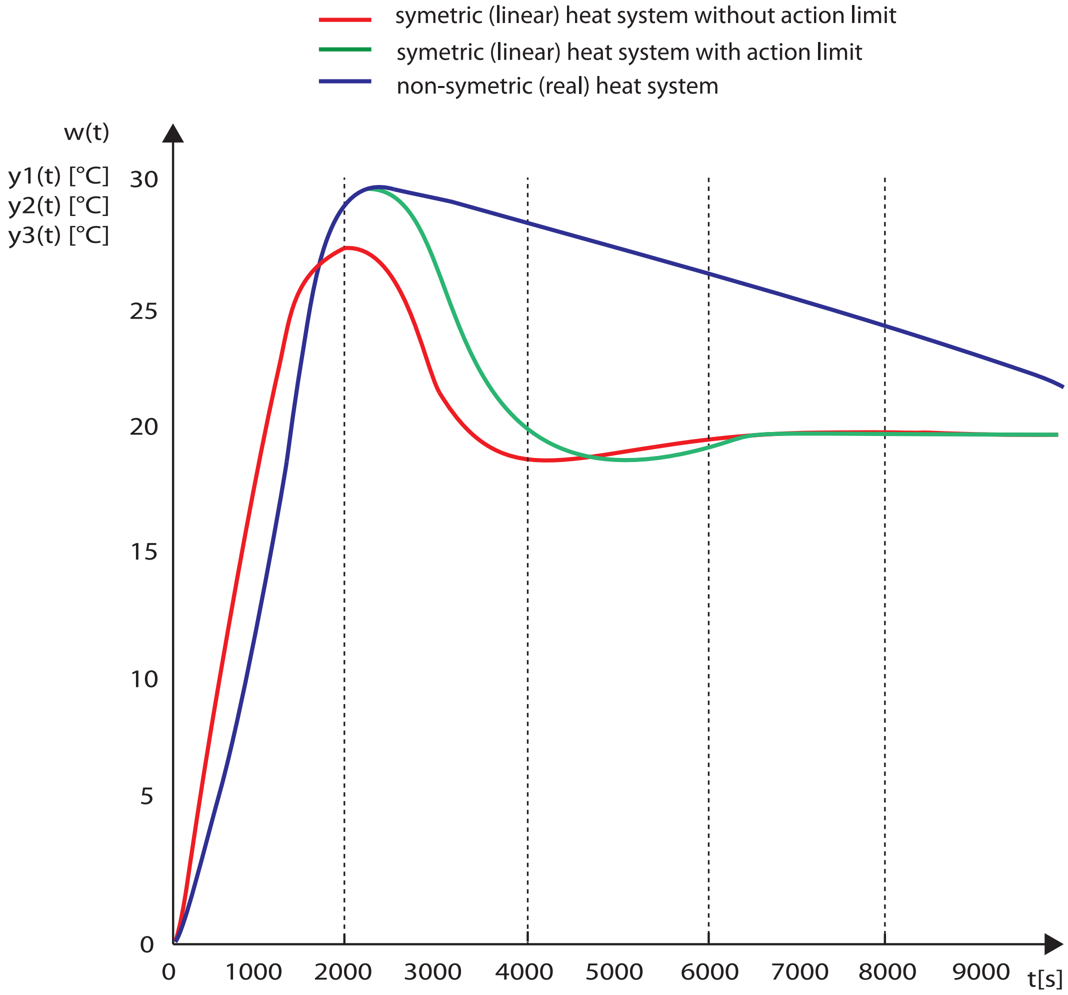


Fig. 8.23: Tansient response for 3 different plants symetric

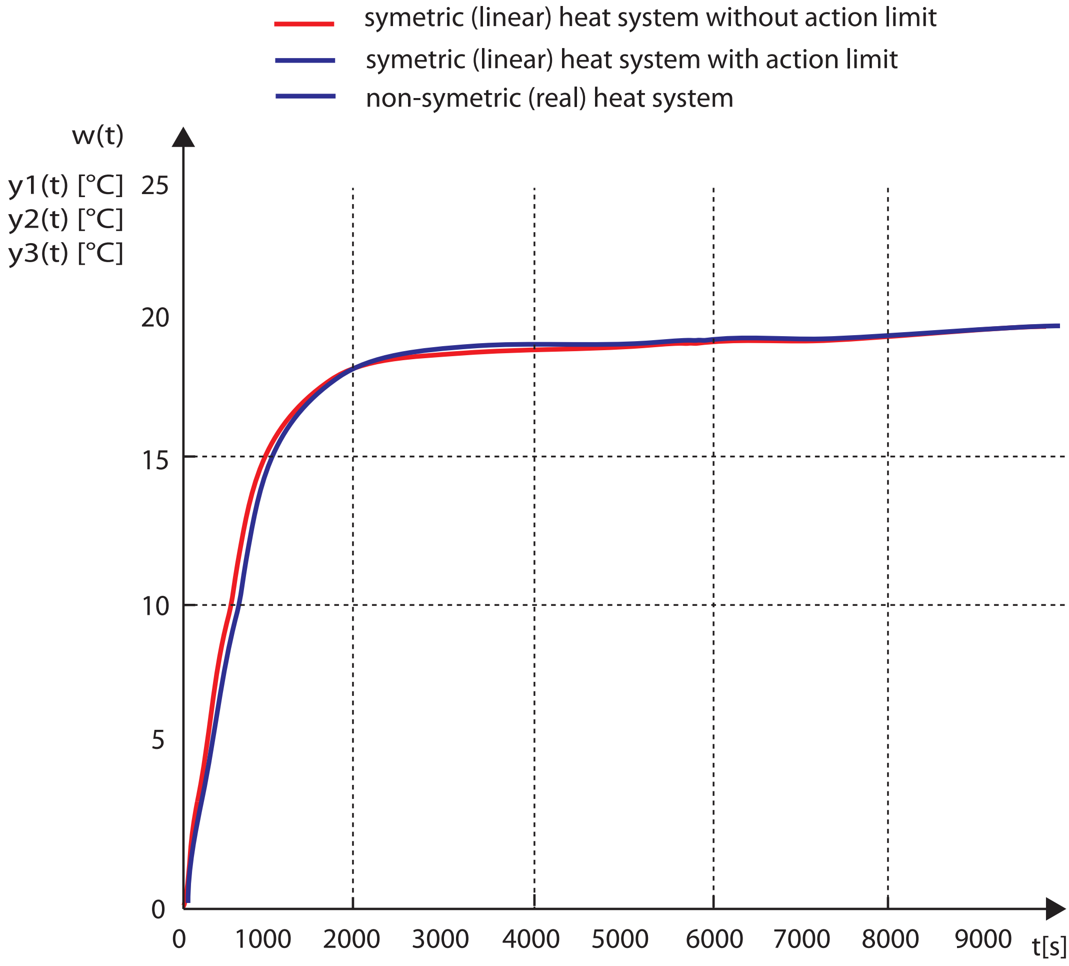


Fig. 8.24: Tansient response for 3 different plants symetric

9.8 Digital controller implementation

Digital controller implementation in PLC requires use of analogue outputs. Such configuration is usually more expensive, than basic one with binary only outputs. Much more expensive and difficult is to amplify analogue control variable to appropriate power level, needed to drive actuators or plant, respectively. It is the reason, why discontinuous controllers are often applied in practice. The parameter adjustment is simplified, and binary only outputs can be used. The actuators are switched using relays, solenoids, electro-pneumatic transducers, nonlinear elements etc.

The performance by means of output value behavior is usually a little bit worse for discontinuous controllers, than for continuous ones, but it is usually satisfactory for most applications. Unwanted effect of discontinuous control can be the oscillatory response and frequent actuator changes. The union of both principles, a multi-step discontinuous and digital PID controller, is a digital PID controller, where the computed control variable is led through a multi-step element. The multi-step output value is encoded as a combination on binary output lines. Continuous behavior can be approximated by use of fast switching of actuated value with variable duty cycle, i.e. pulse width modulation, or by pulses of constant width and variable spacing, i.e. Frequency modulation.