SIEMENS

SIMATIC S7

Fuzzy Control

User Manual

C79000-G7076-C196-02

Safety Guidelines

This manual contains notices which you should observe to ensure your own personal safety, as well as to protect the product and connected equipment. These notices are highlighted in the manual by a warning triangle and are marked as follows according to the level of danger:

Danger

indicates that death, severe personal injury or substantial property damage will result if proper precautions are not taken.

Warning

indicates that death, severe personal injury or substantial property damage can result if proper precautions are not taken.

Caution

indicates that minor personal injury or property damage can result if proper precautions are not taken.

Note

draws your attention to particularly important information on the product, handling the product, or to a particular part of the documentation.

Qualified Personnel

The device/system may only be set up and operated in conjunction with this manual.

Only **qualified personnel** should be allowed to install and work on this equipment. Qualified persons are defined as persons who are authorized to commission, to ground, and to tag circuits, equipment, and systems in accordance with established safety practices and standards.

Correct Usage

Note the following:

Warning

This device and its components may only be used for the applications described in the catalog or the technical description, and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens.

This product can only function correctly and safely if it is transported, stored, set up, and installed correctly, and operated and maintained as recommended.

Trademarks

SIMATIC® and SINEC® are registered trademarks of SIEMENS AG.

Third parties using for their own purposes any other names in this document which refer to trademarks might infringe upon the rights of the trademark owners.

Siemens AG
Automation Group
Industrial Automation Systems
Postfach 4848, D-90327 Nürnberg

Disclaimer of Liability

We have checked the contents of this manual for agreement with the hardware and software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in this manual are reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvement are welcomed.

Technical data subject to change.

Siemens Aktiengesellschaft

Order No. 6ES7 831-0AA20-8BG0

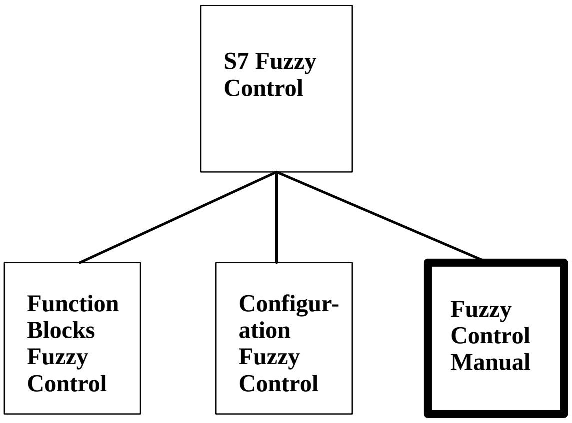
Preface

Purpose of this Manual

This manual helps you to select, configure, and assign parameters to an optimum Fuzzy Control block for your control task.

You will become familiar with the functionality of the Fuzzy Control block and with handling the configuration tool.

Where Does this Manual Fit in with the “S7 Fuzzy Control” Product Range?



The “S7 Fuzzy Control” software package consists of three individual products:

- The product “Fuzzy Control” mainly contains the control block (FB) and the data block (instance DB).
- The product “Configuration Fuzzy Control” mainly contains the tool for configuring the control block.
  This product will also be referred to as the “configuration tool” in the rest of the manual.
- This manual is a separate product and describes the products “Fuzzy Control” and “Configuration Fuzzy Control”.

The Software Package “S7 Fuzzy Control”

Using conventional control concepts to solve automation tasks only produces limited results or requires compromises with regard to the quality of the control. The software package “S7-300 Fuzzy Control” offers a complete concept for creating control functions tbe be used in solving automation tasks.

The control block (function block) is already prepared in its full range of performance and with all algorithms for configuration and assigning parameters.

A user-friendly tool is available for the configuration and parameter assignment of this function block.

Fuzzy controllers are easy to configure on the basis of Fuzzy Control because their functionality is limited to the definition and execution of core functions in fuzzy theory. Even a user with average closed-loop control knowledge is in a position to be able to create a high-quality control system in a short period of time.

Contents of the Manual

Chap. 1 gives you an overview of the structure of fuzzy systems and how they work

Chap. 2 gives you an overview of the function blocks for Fuzzy Control

Chap. 3 explains the functions of the Fuzzy Control function blocks

Chap. 4 gives you an overview of the configuration tool Fuzzy Control

Chap. 5 explains the functions of the configuration tool Fuzzy Control

Chap. 6 shows you how to work with the configuration tool Fuzzy Control

Audience

This manual is intended for use by:

- S7 programmers
- Closed-loop control programmers
- Operating and service personnel

Conventions Used in this Manual

We have used a number of conventions to make access to information in this manual easier:

- The titles and block labels in the left margin help you to access specific topics quickly and give you information on the contents of the manual.
- A topic block always answers a question on the functionality of the tool or gives information about required or recommended sequences.
- References to further information on a topic in other chapters are shown in brackets (see Section x.y). References to other documentation are shown in italics.
- Steps in a procedure are indicated by bullets.
- Steps in a procedure where the sequence is important are numbered.
- Alternatives in procedures or decisions are indicated by a dash.

Additional Assistance

If you have any questions on the use of any of the products described in this manual to which you cannot find an answer, please contact your local Siemens representative or sales office.
You will find the addresses for the main Siemens offices in a range of countries in the appendix “SIEMENS Worldwide” in the *S7-300 Programmable Controller Hardware and Installation Manual*, for example.

If you have any questions on the manual itself or would like to make remarks or suggestions, please fill out the remarks form at the end of the manual and return it to the address given. We would be grateful if you would include your own personal opinion on, and appraisal of, the manual on the remarks form.

To make it easier for you to learn about the SIMATIC S7 automation system we offer a number of courses. Please contact your regional training center or the central training center in Nuremberg, Germany, Tel. (+49) 911 985 3154.

Contents

Fuzzy Control Function Blocks

Fuzzy Control Configuration Tool

Figures

Tables

The Structure of Fuzzy Systems and How They Work

1

What Topics Are Covered in this Chapter?

This chapter contains information on:

- The relationships between process characteristics and fuzzy application
- Signal processing with Fuzzy Control
- The Fuzzy Programming Language (FPL)
- Background information on the methods of Fuzzy Control
- System reactions in marginal cases

1.1 Process Characteristics and Fuzzy Application

Process Characteristics and Fuzzy Control

The drafting and use of fuzzy systems requires no mathematically formulated descriptions of the process you want to automate. This is a great advantage because the calculation of closed process models often demands a very high level of description, particularly where complex systems with many influencing variables are concerned. Conventional methods require higher levels of theoretical knowledge than most practical users actually possess to accomplish these tasks.

Many processes still cannot be fully described even today despite advanced automation theory. The difficulties encountered in creating an optimum open-loop or closed-loop control strategy mean they are not being made adequate use of.

In the given applications the **S7 Fuzzy Control** package enables even processes that are theoretically very difficult to record to be controlled in a fully automated open-loop or closed-loop system with sufficient quality. To apply a fuzzy application you simply have to know how you influence the concrete process to provoke the required behavior. The plant operators and process operators have this knowledge as a result of their background of experience.

If this knowledge is “verbalized”, or expressed in words, it is not difficult to create a fuzzy system that produces results at least as good as those from a conventional system with the support of an experienced plant operator. The non-linear behavior of a fuzzy application represents a very effective copy of human expertise.

Process Analysis

The basis for the interpretation of automation systems is always a process analysis that records the physical behavior. Its data are then used in classic cases for the quantitative definition of the process model. This analysis is replaced in a fuzzy application by rules that describe the process behavior qualitatively.

Even the concrete process behavior which can be measured can be included. The configuration tool in the **S7 Fuzzy Control** package enables process variables to be monitored with the help of the “curve recorder” function.

Area of Application

The main aim of automation is automatic operation that is continuous and as optimal as possible (without influence from unavoidable process interference) in the whole working range of control and reference variables. In linear control systems of any level of dynamic response, this aim is reached by using closed-loop control with conventional P/PI/PID controllers and a steady or unsteady output signal.

Using fuzzy control is advantageous if conventional procedures require frequent corrective interventions from a plant operator or if the process can only be controlled manually. This is particularly the case where a number of highly fluctuating process parameters influence the results of automation. An example for this is the control of vehicles on tracks for which travel times, braking lengths, or accurate positioning etc. strongly depend on load, distance to travel, and state of the vehicle.

Technical processes which are conventionally difficult to control are generally characterized by multi-variable control or non-linear and time-variant process characteristics. They can only be described inadequately by mathematical models. Fuzzy applications may be employed either in addition to, or instead of, conventional control in these cases. The Fuzzy Control block with its defined input/output behavior can therefore be used on the same automation systems in the same way as the other S7 software components for process control, even in conjunction with the functions of other blocks.

Examples of typical successful fuzzy applications:

- Process control including the coordination of subordinate control activities
- Controlling non-linear single and multi-variable systems
- Quality control of a number of features of a product
- Time-variant control parameter assignment or correction of manipulated variables
- Logic structures in control procedures

Drafting a Fuzzy Project

The fuzzy application you are setting up is adapted to the process. Suitable “membership functions” and “rules” (see Section 1.4) have to be set up to perform this adaptation. The draft therefore changes emphasis to the interpretation of other variables with several degrees of truth. These degrees of truth and their use very quickly make the input/output cause-effect relationships more and more complex. We therefore recommend that only the really relevant process variables be included in the fuzzy application.

Advantages of Using Fuzzy Control

Fuzzy systems are particularly powerful because they enable normally vague human knowledge of processes in the environment to be converted into concrete operations. This knowledge, particularly about the static and dynamic behavior of processes and plants, is used to automate processes by employing fuzzy systems. Handling automation tasks requires empirical methods instead of rigid theories.

The creation of a fuzzy application can be reduced to two main tasks:

1. Characterizing the momentary values of relevant input and output variables using “vague” or “fuzzy” definitions and quantifying characteristics such as: *many, few, rather, warm, icecold, fast* etc.
2. Formalizing experience or knowledge of the relevant process or plant in colloquial rules.

   The often varied effects of the inputs on the outputs of a process can be described with the help of logical statements (If/Then rules). These can then be processed using algorithms.

Process and Characteristics of Fuzzy Control

The “membership functions” determine in sections the value patterns of the inputs and outputs which are then processed in the “set of rules”. “Membership functions” and “rules” supply the fuzzy application with all the information it needs to calculate the momentary output values. The output variables are calculated according to algorithms of the fuzzy theory. You will find a brief description of the calculation process in Section 1.4.

The following terms describe the internal subfunctions of a fuzzy system:

- Fuzzification Converting the momentary values of an input to degrees of truth
- Inference Processing the set of rules by linking the degrees of truth of the input with the help of the minimum operator (IF part) and calculating the truth value for the respective output (THEN part)
- Defuzzification Calculation of the numeric output values by weighting the resulting membership functions and forming the area center of gravity

Solutions for automation problems with fuzzy methods are characterized as follows:

- Experience-oriented
- Clear and vivid
- Relatively low in time and effort
- Suitable for practical use
- Generally robust

1.2 Signal Processing with Fuzzy Control

Fuzzy Logic

The basic idea used in the configuration of fuzzy systems is fuzzy logic. Fuzzy as a set of rules with fuzzy, or vague, logic has one distinct advantage over dual-value logic with crisp, clear expressions of membership (binary logic): it comes far closer to the human understanding of processes in plants and systems.

In daily life people describe things and processes with colloquial terms such as many, some, few, more, etc. without thinking about concrete numerical values. However, in binary logic, you are forced to define relatively arbitrary limits and remain within the confines of a strict Either-Or logic.

For example, if you have to describe a temperature in a process which is “hot”, then you can say that a temperature above 90°C is certainly in the “hot” category. The opinions of plant operators diverge at a temperature of 80°C but a temperature of 70°C is clearly classified as no longer “hot”. In fuzzy logic such vague statements can be described with a membership function as shown in Figure 1-1 on the left.

In binary logic, on the other hand, a limit value restricting the lower end of the class of the “hot” temperature must be defined. If, as in the example, you would fix the limit value of this class at 80°C, then a temperature of 79°C is already no longer in the “hot” class.

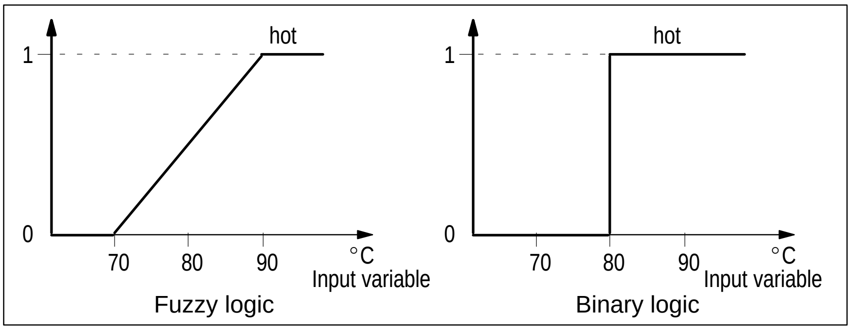


Figure 1-1 Membership Functions in Fuzzy and Binary Logic

The Membership Function

For each input and output variable of a fuzzy application you can set a value range within which this variable, for example a temperature, can vary. The **membership function** defines the relationship to a particular process status within the value range of the input or output variable. It is characterized by a slope-shaped description of the **truth values** of intermediate states.

The function can therefore have the shape of a triangle or a trapezium (see Figure 1-2). Rectangles are also possible and correspond to the effect of a binary status description. The membership graph is fixed by two base point coordinates and two top point coordinates. In the case of a triangular polygon, the top points coincide.

The degree of membership of a particular current process variable value to the various statements about this variable is represented in the form of truth values (between 0 and 1) (see Figure 1-2). The fuzzy system can always calculate the degree of truth accurately via the membership function (or via its definition coordinates).

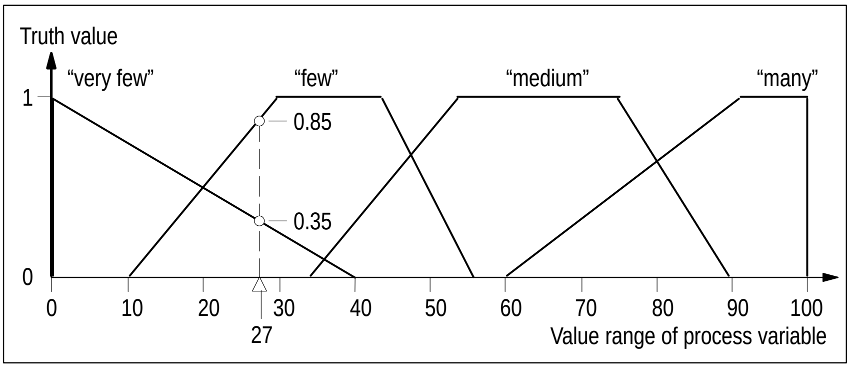


Figure 1-2 Description of a Process Variable by (Four) Membership Functions

Linguistic Values and Membership Functions

In fuzzy theory the concept of the linguistic value plays an important role. If we look at the vague areas of "cold", "warm", and "hot" temperature values within the whole value range of temperature variables, for example, we would describe "cold", "warm", and "hot" as **linguistic values** of the variable "temperature". Linguistic values are a means of addressing areas of a physical variable which are not clearly definable (fuzzy sets).

The classification of states (a possible maximum of seven in Fuzzy Control) within the numeric value range of an input or output variable requires these different states to be characterized. These states describe qualitatively the behavior of the relevant process variable. The description of the variable uses these so-called linguistic values which are all defined "fuzzily" by a membership function.

The membership function declares the degree of membership to the specific linguistic value for every numeric value of the process variable that occurs. (This process can be described as as quantifying the qualitative statement of a linguistic value.)

As the membership functions generally overlap, a number of membership functions of zero can provide different truth values for a particular process value. In Figure 1-2 the membership functions for the linguistic values "few" and "very few" define with which truth values the process value "27" is considered to be "few" and "very few".

Complement of a Linguistic Value

In addition to defining and quantifying linguistic values it is also possible to link the complement of a linguistic value into the IF part of a rule in the set of rules for Fuzzy Control. Here the complement C of a linguistic value or of a membership function A is described by the relation C = 1 - A (Figure 1-3).

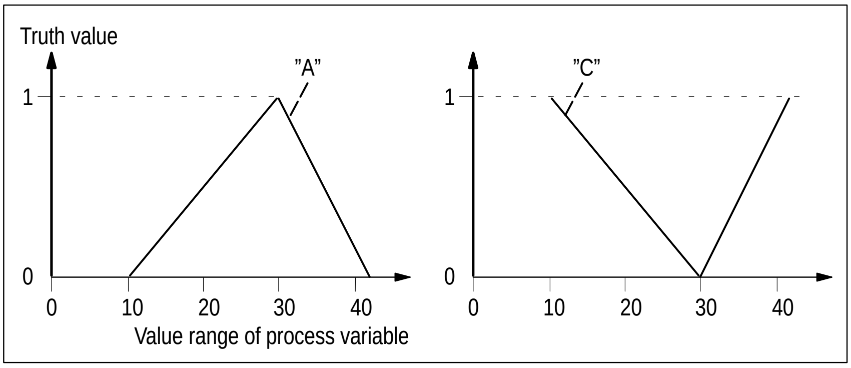


Figure 1-3 Forming the Complement of the Linguistic Value “A”

Set of Fuzzy Rules

Characteristic for fuzzy applications is the direct use of fuzzy empirical knowledge to create the optimum behavior of the output variables in all possible value constellations of the input variables. This knowledge of how the process reacts is summarized in the following rules using this method

IF <condition> ... THEN <conclusion>

These IF-THEN rules correspond to the simplest type of human decision-making. Condition and conclusion are vague statements such as:

IF “the pressure is high”,
THEN “open the valve a little”.

or linked statements such as:

IF “the pressure is high”
AND “the temperature rises steeply”,
THEN “open the valve a little”.

Each of these linguistic rules, also known as production rules, describes part of a strategy that determines the behavior of the fuzzy application.

To mathematically process these IF-THEN rules, an operation must be found to link the fuzzy values of the IF part and those of the THEN part (or to link their membership functions). In Fuzzy Control forming the minimum is used to do this.

This method (Figure 1-4, above) makes it clear that the conclusion part can at most only have the same degree of truth as the condition. The membership function of a rule results from limiting the membership function of the THEN part to the current truth value of the IF part.

The IF part can also be a network of logic operations as complex as you wish. The THEN part is generally a simple assignment of a linguistic value to an output variable. By formulating the rules accordingly it is possible to create a situation in which a maximum of one rule is responsible for each linguistic value of an output variable.

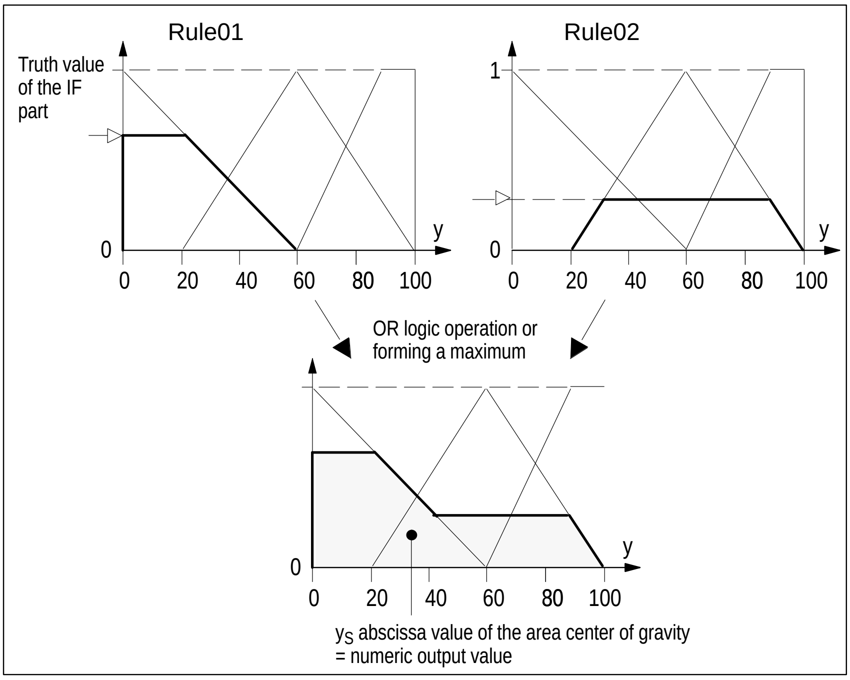


Figure 1-4 Calculating the Membership Functions for Two Rules

In Fuzzy Control algorithms have been implemented that work through these rules according to the laws of fuzzy logic. The sum of all rules, or the **set of rules**, forms the core of a fuzzy application. The overlapping of the rules produces a overall membership function (Figure 1-4, bottom) in the algorithmic system of Fuzzy Control. It characterizes the current effect of the set of rules.

From the form of the membership function a numerical value is generated by forming the area center of gravity (defuzzification). This operation completes the processing of the set of rules in the fuzzy system.

The methods used and the mathematical formulae are summarized in Section 1.4.

Process Handling with the Fuzzy Application

As fuzzy logic makes it possible to not only define two binary states but also intermediate values between 0 and 1, the description of the process is far better and more natural, coming much closer to the real-life situation. This extension of the concept of membership enables the linguistically formulated expertise to be calculated using the algorithms of the function block **Fuzzy Control**.

The fuzzy applications created with the standard function block are purely software solutions for the creation of digital sampling systems. Their input and output values are processed on a CPU (S7-300 or S7-400) with the help of digital algorithms. They can be time-driven by the operating system of the CPU or called cyclically for processing. In the user program they are called absolutely. The frequency with which a particular fuzzy application is processed is dependent on the program processing level.

The algorithm in the processor emulates the fuzzy application under real-time conditions. Between the exact points of the sampling times the system does not react to changes to the process variables; the output variables remain the same.

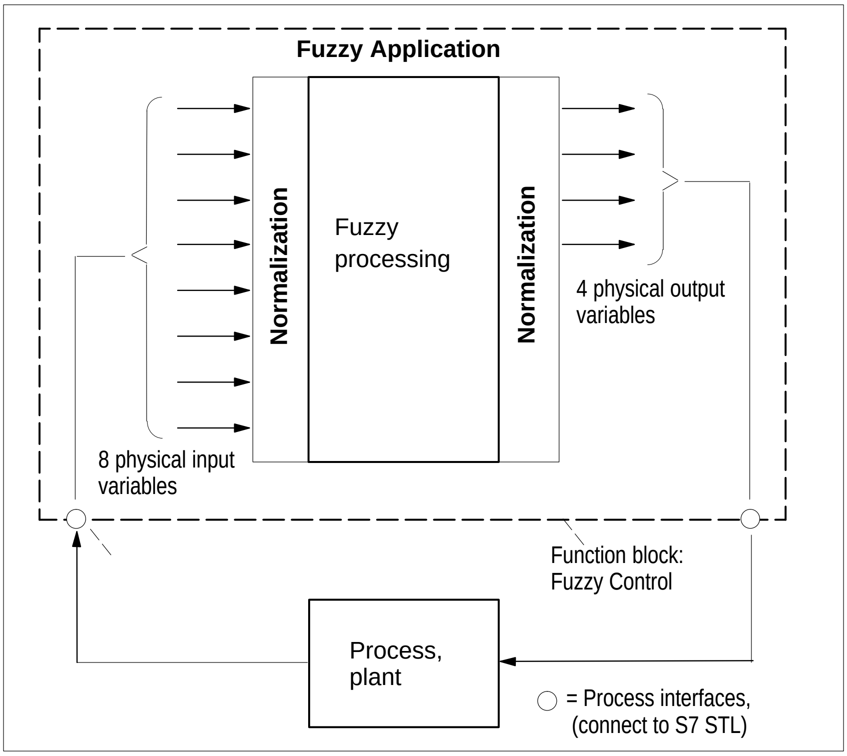

Fuzzy Application
Normalization
Fuzzy processing
Normalization
4 physical output variables
8 physical input variables
Function block: Fuzzy Control
Process, plant
○ = Process interfaces, (connect to S7 STL)

Figure 1-5 Fuzzy Application Working in Conjunction with the Process

In Figure 1-5 the use of a fuzzy application in a closed or open-loop control system is shown in schematic form. This structure with the maximum use made of the available inputs and outputs gives you an overview of the possible configurations.

Configuration and Structure of Fuzzy Control

Configuring a fuzzy application can be divided into three steps:

- Defining the inputs and outputs
- Determining the membership functions
- Setting up the fuzzy rules

All these steps are performed with the help of the “Configuration Fuzzy Control” tool.

By defining input and output variables you can create a multi-dimensional open-loop or closed-loop control system. The strategy of the fuzzy application represents a multi-dimensional, purely static non-linearity with a correspondingly non-linear family of graphs.

After configuration, it is recommended that you perform a few off-line tests in order to check that the fuzzy application is functioning and behaving correctly. For this purpose there is a curve recorder to which you can assign parameters which places triangular functions or constant numeric values at the required inputs.

Now you connect the configuration tool (installed on your programming device or PC) to the S7-CPU. The function block Fuzzy Control and the fuzzy application in the form of an instance data block are downloaded to the destination hardware via the multipoint interface (MPI) or via a bus link. The execution environment must already be installed on the destination system and linked in with the system environment. The inputs and outputs should already be connected in the system.

Only the data relevant for execution are downloaded to the CPU. The designations of the inputs and outputs are only stored in the off-line database on the programming device.
When reading back a fuzzy application from the CPU without the corresponding off-line database, the original formal designations are used.

With the monitoring functions in the configuration tool you can examine the behavior of the fuzzy application under real-time conditions. The values of the inputs and outputs are displayed in the form of time curves. Conclusions can be drawn on the effect of the fuzzy application by reference to the control effectiveness. The measurement values are stored so that you can also make a retrospective analysis.

These functions are arranged in Windows-compatible windows in “Configuration Fuzzy Control” and can be activated via menus (see Section 4.1).

1.3 The Fuzzy Programming Language

The FPL Standard

Fuzzy Control is not a method unique to Siemens. The data format used to store fuzzy applications is the Fuzzy Programming Language (FPL). The FPL standard was specified by Togai InfraLogic, Inc. and is used in a number of fuzzy tools. By using this standard it is possible to exchange fuzzy projects between the different development tools.

Limitations of the FPL Standard in Fuzzy Control

The processing tool “Configuration Fuzzy Control” uses a subset of the FPL standard for describing fuzzy applications. For example, all fuzzy projects can be processed with the TILShell. In reverse, projects of the TILShell can only be processed with “Configuration Fuzzy Control” if you keep to the limitations of the scope of the language:

- Only one set of rules is permitted (control word FUZZY).
- No modular fuzzy systems are possible in a Fuzzy application. The control word PACKAGE is not permissible.
- A maximum of 8 inputs, 4 outputs, and 200 rules are possible with 2 inputs and 1 output.
- Each input may contain a maximum of 7 membership functions.
- Each output may contain a maximum of 9 membership functions.
- 10 characters for inputs and outputs and 7 characters for the linguistic variables are significant for name identification in Fuzzy Control.
- Only AND operations may be used in the fuzzy rules. OR and NOT are not permissible.
- A membership function of an input may only have the form of a rectangle, triangle, or trapezium, in which case in an image with four points, the first and the fourth points must have a degree of truth of 0 and the second and third points a degree of truth of 1. With vertical lines specifying the upper point is sufficient. This means that a rectangular membership function can be described with only two points.
- Only singleton functions can be processed in Fuzzy Control as membership functions of the outputs. When data are entered in the TILShell narrow rectangular functions (width = $^{1}/_{200}$ of the total range) must be selected for the outputs in order to be able to process these functions in Fuzzy Control.
- The coordinates of the membership functions must be stored in a points list. Specifying equations for membership functions is not permissible when working with “Configuration Fuzzy Control”.

FPL Control Words

The following list gives the control words from the FPL syntax which are used by Fuzzy Control.

/***Comment***/	Comment within the character string /*. . . */
PROJECT	Start of a project with project designation
VAR	Start of the variable description (input/output)
TYPE	Type of the variable (in Fuzzy Control always FLOAT)
MIN, MAX	Limit values for a variable
MEMBER	Start of the definition of a membership function
POINTS	Points list of a membership function
FUZZY	Start of the set of rules
RULE	Definition of a rule
IF, IS, AND, THEN, =	Descriptive elements of a fuzzy rule
CONNECT	Determining the associations of input, output, set of rules
FROM, TO	Connection elements in CONNECT
END	End of an object

Tolerated FPL Elements

There are a number of language elements in FPL which are not used but are tolerated by Fuzzy Control. This means that these control words do not correspond to any function in Fuzzy Control. FPL elements which serve for incorporating program parts in C code are accepted without comments.

MAP	WARNING, no mapping possible
INITIALLY	WARNING, preallocation not permitted
DEFAULT	WARNING, value output with inactive rules not possible
OPTIONS	Additional information for objects
SOURCE	Incorporating parts in C code
FRAGMENT	Incorporating parts in C code
#CODE..#END_CODE	Incorporating C source code

Example of an FPL File

The following example shows a fuzzy application generated with “Configuration Fuzzy Control”. Two inputs and one output are used in this project. The first input is designated **Temperature**, the second input **Pressure**. The output has been designated **Valve**. Four fuzzy rules are defined.

```
STL                                                           Explanation
/*Copyright (c) Siemens AG 1995*/
/*FPL file generated by Fuzzy Control on 05.06.95*/

PROJECT FPLTEST
   VAR   Temp
         TYPE float
         MIN 0.00
         MAX 30.00

         MEMBER cold
            POINTS 0.00,1.0 3.00,1.0 27.00,0.0
         END

         MEMBER hot
            POINTS 3.00,0.0 27.00,1.0 30.00,1.0
         END
   END

   VAR   Pressure
         TYPE float
         MIN 50.00
         MAX 100.00

         MEMBER low
            POINTS 50.00,1.0 55.00,1.0 95.00,0.0
         END

         MEMBER high
            POINTS 55.00,0.0 95.00,1.0 100.00,1.0
         END
   END
   VAR Valve
         TYPE float
         MIN –101.0
         MAX 101.0

         MEMBER drainage
            POINTS –101.0,1.0 –99.0,1.0
         END

         MEMBER closed
            POINTS –1.0,1.0 1.0,1.0
         END

         MEMBER inlet
            POINTS 99.0,1.0, 101.0,1.0
         END
   END
```

```
STL                                                           Explanation
   FUZZY ProFuzzy
         RULE Rule_01
            IF (Temp is cold) AND (Pressure is low)
            THEN Valve = inlet
         END

         RULE Rule_02
            IF (Temp is cold) AND (Pressure is high)
            THEN Valve = closed
         END

         RULE Rule_03
            IF (Temp is hot) AND (Pressure is low)
            THEN Valve = closed
         END

         RULE Rule_04
            IF (Temp is hot) AND (Pressure is high)
            THEN Valve = drainage
         END
   END

   CONNECT
         FROM Temp
         TO ProFuzzy
   END

   CONNECT
         FROM Pressure
         TO ProFuzzy
   END

   CONNECT
         FROM FROM ProFuzzy
         TO Valve
   END
END
```

1.4 Methods and Algorithms in Fuzzy Control

Background Information

This section gives you information on:

- – The methods used by Fuzzy Control for processing signals
- – The algorithms in the fuzzy function block for calculating the output variables

If you are mainly interested in the practical handling and solving of automation tasks with fuzzy means you do not need to read this more theoretical section.

The calculation algorithm is explained first using the example of trapezoidal membership functions for the outputs. We will go into the special features of the calculation of output values with singleton functions later in this section.

Internal Algorithm

The fuzzy algorithm for calculating the numeric output values from a momentary constellation of numeric input values is performed internally in the following sub-operations (see Figure 1-6):

- Fuzzification (vague description of the behavior of input variable)
- Inference (IF...THEN rule processing)
- Defuzzification (calculation of the output variables by weighting the THEN parts of the set of rules and calculating the center of gravity)

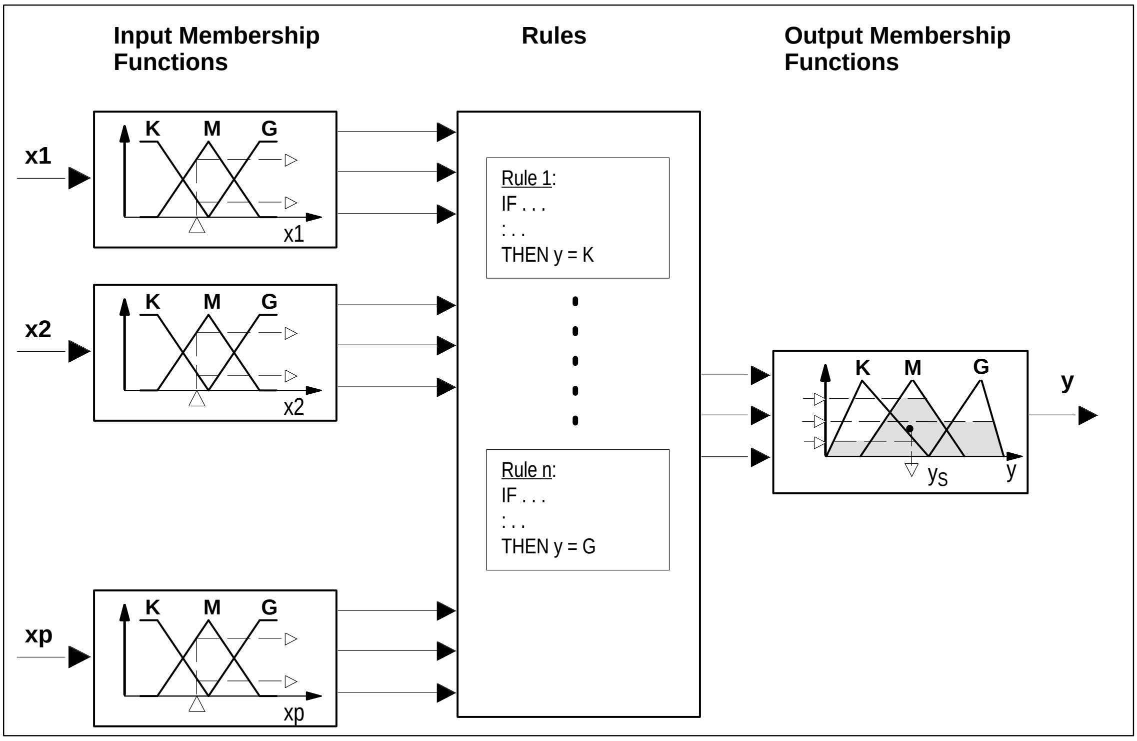


Figure 1-6 General Structure of a Fuzzy System with p Inputs, n Rules, and One Output

Fuzzification and Inference

In the first step, **fuzzification**, the degrees of truth for the linguistic variables are calculated proceeding from the current values of the input signals. In Figure 1-7, there is fuzzification for the input variables of temperature and pressure. For the temperature, the following degrees of truth result from the fuzzification:

$\vartheta_{cold} = 0$, $\vartheta_{warm} = 0.95$, and $\vartheta_{hot} = 0.35$

and for the pressure:

$p_{low} = 0.25$, $p_{medium} = 0.85$, and $p_{high} = 0$

The fuzzy rules are applied in the next step, **inference**. In the IF part of a rule the degrees of truth which were calculated in the fuzzification are linked by the fuzzy operator. There are a large number of operators which are used in fuzzy algorithms for calculating the AND and the OR logic operations. In most cases the **minimum** is used for calculating the AND and the **maximum** for the OR.

The following applies for given values of the input variables processed in the IF part: the membership function of the THEN part is restricted to the resulting value of the IF part. The membership function of a rule is calculated in Fuzzy Control with the forming of a minimum as an arithmetic standard.

In the example, for rule 01

ϑ_{warm} = 0.95 and p_{medium} = 0.85 linked by AND = 0.85

For rule 02:

ϑ_{hot} = 0.35 and p_{low} = 0.25 linked by AND = 0.25

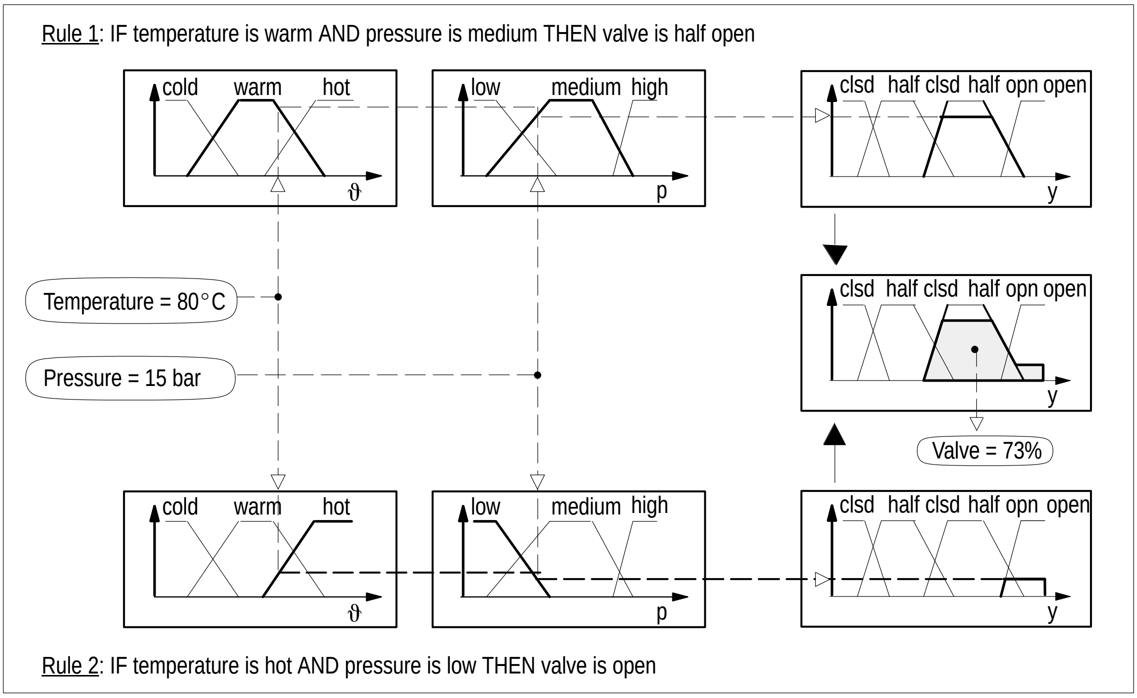

Rule 1: IF temperature is warm AND pressure is medium THEN valve is half open
cold
warm
hot
ϑ
low
medium
high
p
clsd
half clsd
half opn
open
y
Temperature = 80° C
Pressure = 15 bar
Valve = 73%
Rule 2: IF temperature is hot AND pressure is low THEN valve is open

Figure 1-7 Fuzzy Processing of the Input Variables ’Temperature’ and ’Pressure’ and the Output Variable ’Valve’

Linking together the membership functions for a number of rules is known as inference. An OR logic operation or the forming of a maximum is used here as the arithmetic rule. A composition operation is realized with the help of this common max-min inference method. This operation allows the effective parts of all rules to come into play. The overlapping of the cut-off membership functions of a number of rules is added together to form one resulting membership function (polygon, Figure 1-7 on the right).

Defuzzification

The result of the inference and composition methods of Fuzzy Control is a vague, or fuzzy output variable: a membership function is supplied as the output information for the set of rules for given momentary values of the input variables. In fuzzy systems this unclear information must be converted into a representative numeric value because a manipulated variable, for example, cannot process a membership function. This calculation of numeric values is known as **defuzzification**.

For the calculation of output values, the membership functions of the output are first weighted with the current degrees of truth.

This weighting can be done according to different methods:

– A simple cutoff of the corresponding membership functions according to the THEN part of the respective rule analysis (Figure 1-7).

– The multiplication of the membership functions with the degree of truth.

The advantages and disadvantages of both methods will be explained below.

In the third calculation step, defuzzification, the computed result of the IF part must be applied as the degree of truth for the THEN part of a rule. In other words, this means that the THEN part must be weighted with the results of the IF part.

In the example, the intended results are:

Rule 1: Valve = half open, with a degree of truth of 0.85
Rule 2: Valve = open, with a degree of truth of 0.25

A manipulated variable corresponding to the weighting of the degrees of truth must be calculated for the valve. It must lie between “half open” and “open” but a greater weight is attached to the linguistic variable “half open”.

The commonly used **center of gravity** method calculates the output value. It is calculated as the abscissa of the center of gravity of the area lying below the output membership function. A unification area is formed from the weighted area portions. The area center of gravity of this resulting area is the defuzzified result which is output to the process as the value of a manipulated variable. In the example (Figure 1-7) the center of gravity is represented by the valve being open by 73%.

Calculating the Center of Gravity

The general formula for calculating the coordinate x_S of the center of gravity $S(x_S, y_S)$ of an area segment that lies below a function y = f(x) between x = x_A and x = x_E is represented as follows in math:

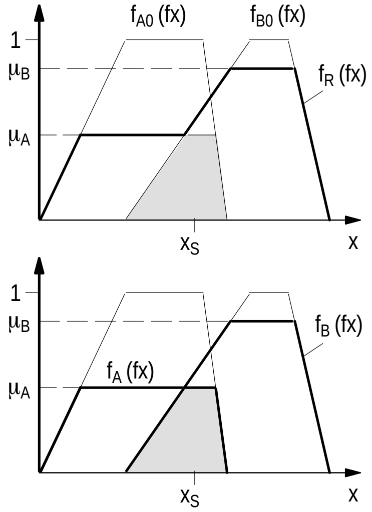
$$x_{S}{}^{\cdot} = \frac{\int_{x_A}^{x_E} x \ f_R(x) \ dx}{\int_{x_A}^{x_E} x \ f_R(x) \ dx}$$

Problems with Defuzzification

1. When the membership functions are cut off, the intersections of the output area with the horizontal straight lines (corresponding to the degree of truth) must be calculated.
2. When the unification area is formed the intersections of the output membership functions of each fuzzy rule (polylines) must be calculated.
3. A complex integration must be performed to calculate the center of gravity (see above).

For these reasons it is advisable to use methods which can be calculated as quickly as possible and with low memory requirements. The starting point of the calculation was cut-off area segments to form the unification area.

22



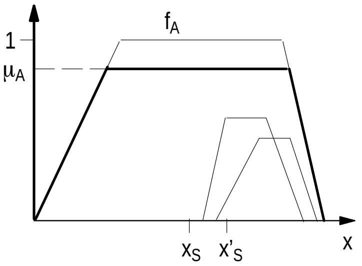
The two area segments of the output membership functions overlap in the shaded area. In the calculation of the unification area, this shaded area is taken into account only once.

If, on the other hand, the two part areas are considered separately, the calculation of the resulting function f_R (polygon) is not required. Accordingly, instead of an OR operation of the two areas, a summation is made. In this way, overlapping area segments are included several times in the calculation.

The center of gravity is calculated according to the following relation:

$$x_{S}' = \frac{\int x\ f_A(x)\ dx + \int x\ f_B(x)\ dx + ...}{\int f_A(x)\ dx + \int f_B(x)\ dx + ...}$$

Whether the overlapping portions are taken into account only once (OR operation) or several times (summation) in the calculation naturally influences the calculation result. It is simply a matter of opinion whether you prefer one method or the other.

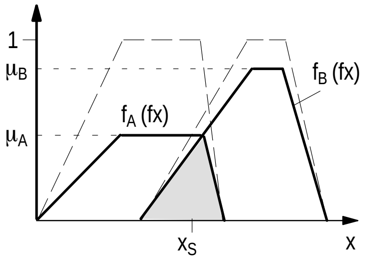


The adjacent figure shows an extreme case of overlapping in which both area segments of membership functions are located completely within a single membership function. If the overlap is not taken into account a center of gravity x_S for the area results.

If the internally located portions are taken into account additively, the center of gravity is obtained with the coordinate x'_S. If the function f_A is seen as dominant, x_S is the better solution. On the other hand, if you want to consider the influence of the fuzzy rules which generate the internal portions, then x'_S would give a better result.

Weighting Using Multiplication

If you calculate the area segments separately and handle the overlaps additively, this simplifies the calculation of the center of gravity. In this case multiplication is used for the weighting of the membership function with the degrees of truth.



The calculation of the weighted area segments f_A and f_B and of the center of gravity is as follows:

$$f_A(x) \;=\; \mu_A \cdot f_{A0}(x) \cdot$$

$$f_B(x) \;=\; \mu_B \cdot f_{B0}(x) \cdot$$

$$x_S \cdot = \frac{\mu_A \int x\ f_{A0}(x)\ dx + \mu_B \int x\ f_{B0}(x)\ dx + ...}{\mu_A \int f_{A0}(x)\ dx + \mu_B \int f_{B0}(x)\ dx + ...}$$

Within the integrals of the equation for the x_S calculation only the origin functions are still present. Since these functions no longer depend on the input variables of the process the integration can be performed off-line while the project is being generated. The integrals in the numerator of the fraction correspond to a moment M and the integrals in the denominator an area segment A. Under these assumptions the calculation is further simplified and the following applies:

$$x_S \cdot = \frac{\mu_A\ M_{A0} + \mu_B\ M_{B0} + ...}{\mu_A\ A_{A0} + \mu_B\ A_{B0} + ...}$$

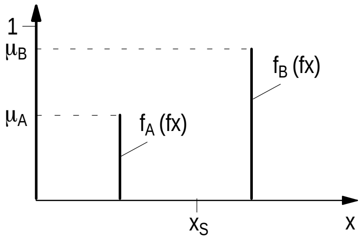
Only the moments M and the areas A of the membership functions of the outputs are included in the calculation in this equation. The form of the membership function no longer has any influence on the result of the calculation. Only the size of the area and the position on the x-axis are decisive.

Singleton Functions

Proceeding from the last equation, it requires only a small step to arrive at the calculation method with singleton functions:

- Select the membership functions f_{A0} and f_{B0} which are independent of x. (For example, rectangles with a function value of 1.)
- Now you can pull and shorten f_{A0} and f_{B0} in front of the integrals.
- Then let the width of the rectangles tend towards zero and cross the limit.

Only a dependency on the position x remains in the equation.



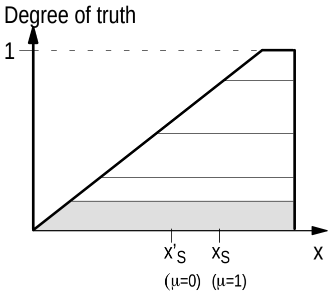
A relation which can best be interpreted visually with the geometry of “line functions” or **singletons** has arisen:

$$x_{S'} = \frac{\mu_A\ x_A + \mu_B\ x_B + ...}{\mu_A + \mu_B\ + ...}$$

This method of calculation is applied in Fuzzy Control for defuzzification. The calculation of output values with singleton functions offers a number of advantages for the fuzzy application:

1. It is simple, fast, and requires very little memory.
2. It is much easier for a user to define singleton functions for an output instead of areas. If singleton functions are used, the value of “valve = closed”, for example, is, in fact, zero.

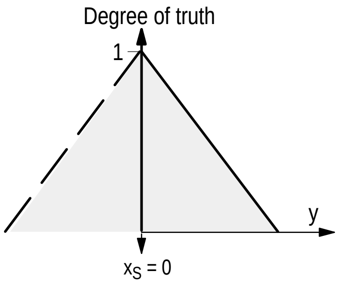
   Another side effect in the application of area membership functions has the following result:

   If unsymmetrical areas and the cut-off method are used, the center of gravity of the single area (**!**) depends on the degree of truth. If, for example, only a single rule acts, requiring “valve = closed”, the output value is dependent on the strength of the control activity.

3. When areas are used as membership functions for the outputs there is always the problem that the value range cannot be fully utilized because the center of gravity of an area does not lie at the edge.

   If you define a range from **minimum** to **maximum** and use area functions you will obtain an output value that reaches the range limits.



To solve this problem the membership functions are mirrored at the edge in the values of minimum and maximum. The center of gravity of the edge functions is thus placed artificially at minimum and maximum. This method changes your data entries.

4. The fact that the basic idea of processing vague information with fuzzy methods is not restricted by the use of singleton functions speaks as another argument for their use.

   The derivations for the algorithms show that the form of the defined membership functions only has a decisive influence on the inputs. Their form is of secondary importance at the outputs. The “soft” fuzzy transitions are achieved in defuzzification of the output information by weighting the membership functions and by calculating the center of gravity.

The example in Figure 1-7 takes on the following form (Figure 1-8) if it is realized using **Fuzzy Control**:

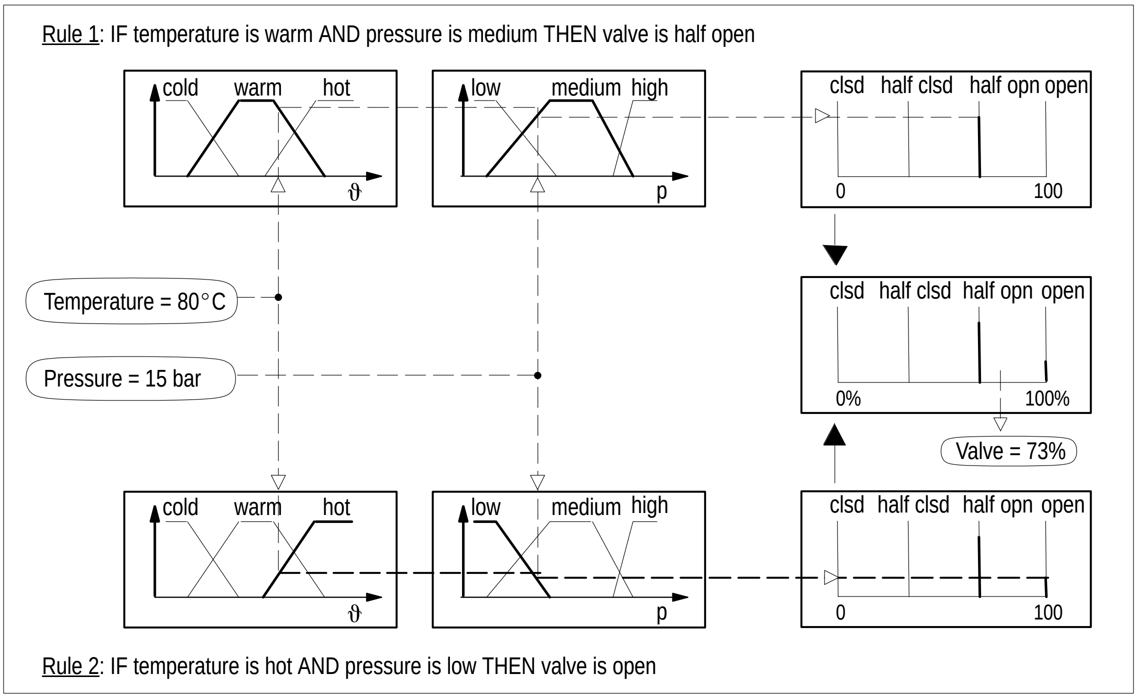


Figure 1-8 Fuzzy Control Processing of a Set of Rules with Two Input Variables and One Output Variable

1.5 System Reactions in Marginal Cases

Questions Which May Arise During Configuration

During the use of Fuzzy Control you will probably come across a number of questions, depending on your configuration, some of which are answered below.

What happens if only the IF part of a rule is present in a fuzzy system but no THEN part?

A rule without a THEN part has no influence on the output value but requires processing time on the destination system. Therefore you can delete the THEN part for test purposes and retain the IF part, but fragments of this type should be removed from the final fuzzy system.

What happens if only the THEN part of a rule is present in a fuzzy system but no IF part?

A rule without an IF part still has an influence on the output result and can also be used constructively in a fuzzy system. If there is no IF part, a truth degree of 1 always acts on the THEN part. This means that this rule always acts as if the conditions of the IF part were fully fulfilled.

What happens if no single rule is active for an output on processing?

If no single rule is active, then the value of the corresponding output is undefined. However, since “undefined” is not a numerical value, some substitute value must appear at the output. The two most popular methods of reacting in such a case are on one hand to output a default value and on the other hand to output the last defined value. The last defined value is output in Fuzzy Control.

What happens if no fuzzy rule was entered or all columns of the rule table are empty?

The fuzzy system is in the same state as in the last question. Since no rules were entered, no rule can make a contribution to the output signal. The outputs are undefined and the last defined state is retained. If there was no previously defined state, all outputs have the value zero.

What happens if signals have to be monitored in the PLC which are not inputs of the fuzzy system?

A total of eight inputs can be defined in Fuzzy Control. This maximum number is required in only a few cases. If you now want to monitor signals which are not inputs of the fuzzy system, you can switch these signals to free inputs and monitor them using “Configuration Fuzzy Control”. Only the limits of the value range and the names need to be specified for these “monitoring inputs”.

Membership functions are not defined. To retain optimum computing time in the CPU, these “monitoring inputs” should be placed at the end of the list of inputs.

What happens if an input signal is outside the defined limits?

If input signals are smaller than the defined Minimum or larger than the defined Maximum, Fuzzy Control fixes them at the value of the relevant limit. Thus a fuzzy system shows the same behavior at process data values outside the limits of the value range as at the limits.

Product Overview

2

What Topics Are Covered in this Chapter?

This chapter describes:

- The purpose and capabilities of the Fuzzy Control product
- Its structure
- Where it can be used

2.1 Purpose and Capabilities of Fuzzy Control

Purpose

Solving automation tasks with conventional control concepts only produces limited results or forces compromises to be made with respect to the quality of the control. The “S7-300 Fuzzy Control” software package with its multiple areas of application offers an extremely modern alternative for solving automation tasks. This applies particularly in the case of processes with non-linear behavior whose characteristics change considerably depending on the operating point. Fuzzy Control also promises particularly good results when an output variable is dependent on more than one process variable.

Capabilities

Fuzzy applications are automation functions that can be realized with Fuzzy Control. In their fully functional state they are programmed with all their algorithms and contained in an S7 standard function block. This function block can be used on all SIMATIC S7-300 and S7-400 programmable controllers. The processing of the block is either time-driven or triggered when it is called by the user program.

The user-friendly tool “Configuration Fuzzy Control” is supplied for you to configure your fuzzy application and adapt it to the concrete demands of your process. A small number of easy-to-handle editing menus in the dialog windows of the user interface lead you step-by-step through the creation of an application. All the configuration requires from you is for you to name the variables, define value ranges and function coordinates, and fill out rule tables.

With the software package “Fuzzy Control” a fuzzy application or a fuzzy controller can be configured for a specific automation task. The functional content of Fuzzy Control is determined freely by means of the set of rules for which up to 200 IF...THEN relationships can be formulated.

For the automation of technical processes the principal structure of a fuzzy application can look like Figure 2-1. From specially prepared input variables it creates output variables that influence the process in the required fashion according to the rules in the rule basis. The application is designed based solely on empirical knowledge of the process that can be formulated in colloquial terms. Theoretical knowledge of differential equations, transfer functions etc. is not required to understand a fuzzy system.

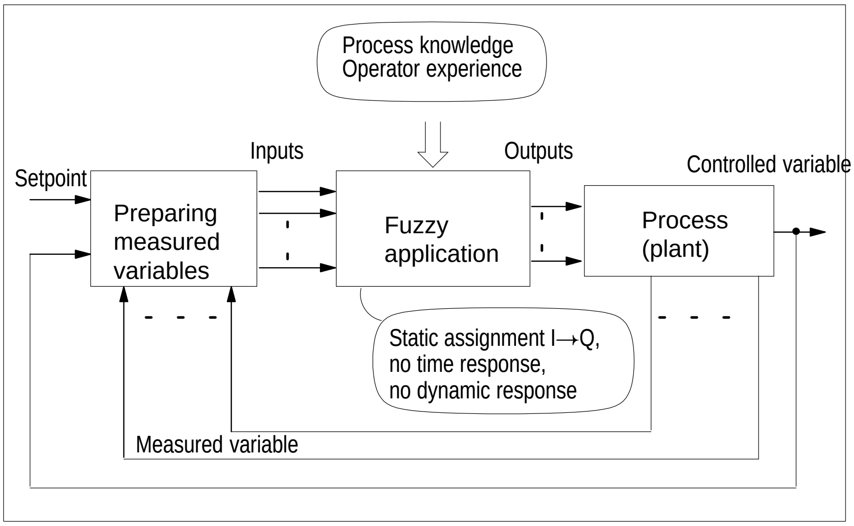


Figure 2-1 Schematic of a Fuzzy Control System

As well as being used as a controller, the fuzzy system can also be included in automation structures where it can contribute to forming controlled variables and to selectively correcting system parameters.

2.2 Structure of Fuzzy Control

Product Structure

The scope of supply of Fuzzy Control on diskette includes standard function blocks with a database, together with a pre-programmed standard example of a characteristic application.

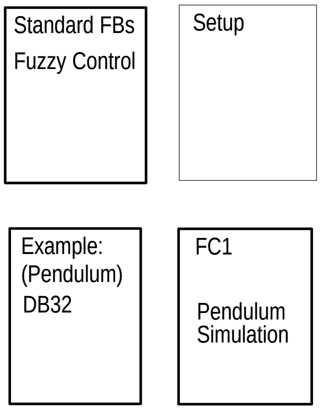


Figure 2-2 Contents of the Software Product “Fuzzy Control”

- The standard function block **Fuzzy Control** contains all the necessary algorithms for fuzzification, rule evaluation, inference, and defuzzification of a powerful fuzzy system.
- The instance data blocks with pre-programmed examples are ready to run, and to be loaded and called. The example “Pendulum” is supplemented by a software simulation of the pendulum (FC) so that a closed-loop control circuit can be set up.

The software package also contains a **Setup** program for installing “Fuzzy Control” on a programming device or PC.

Program and Data Structure

The software product “Fuzzy Control” consists mainly of a function block (FB) which contains the procedures and algorithms for shaping the behavior of fuzzy applications. This means it is an automation tool on the basis of a software program in which a standard function block embodies the whole functionality of a fuzzy application with several inputs and outputs.

The behavior of the actual application is determined by logical algorithms and numerical calculations of the function block. The data required for these cyclic calculations are stored in an application-specific data block. Only one function block is required to create a number of fuzzy applications.

Each fuzzy application is represented by an **instance data block** which has to be created for the specific application. When you use the configuration tool, this instance data block is created implicitly for you. This means you only have to define the inputs and outputs, the membership functions, and the set of rules in the editing windows of the user interface to complete the information for a specific application.

The algorithms for a particular fuzzy application are calculated in the processor of the S7 programmable controller (PLC) after an absolute call by the user program or cyclically in time-driven intervals. The results of the calculation (measured and manipulated variables) are stored in the respective instance data block or passed to the process I/O.

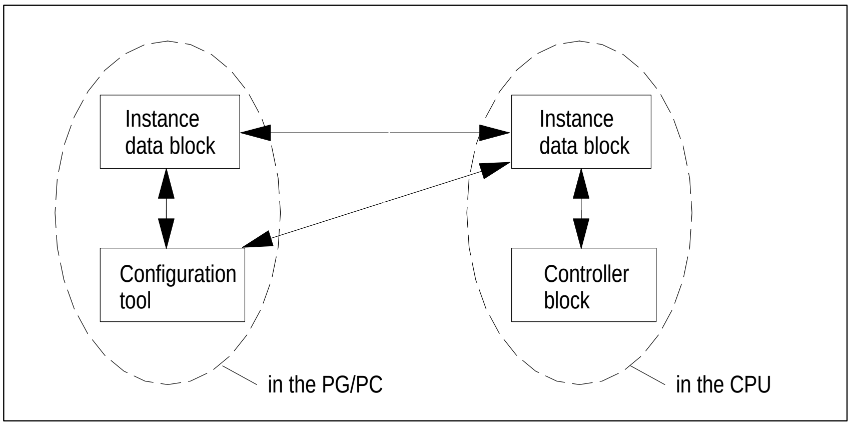


Figure 2-3 Data Structure

2.3 Working Environment

Hardware and Software Requirements

The software product “Fuzzy Control” can be used on all SIMATIC S7 programming devices on which the STEP 7 programming software is loaded.

Creating and starting up fuzzy applications is done with a programming device. This is connected to the CPU in the destination programmable controller either directly via a point-to-point link (MPI), or via the LAN bus (SINEC L2 or SINEC H1) and a communication module (CP). If the programming device does not have a multipoint interface, you will require a PC/MPI cable which you connect to the communication port on the programming device for a direct connection.

The fuzzy applications you create with the software package “Fuzzy Control” run on the CPUs (with real number and time-of-day interrupt capability) of the S7-300 and S7-400 family.

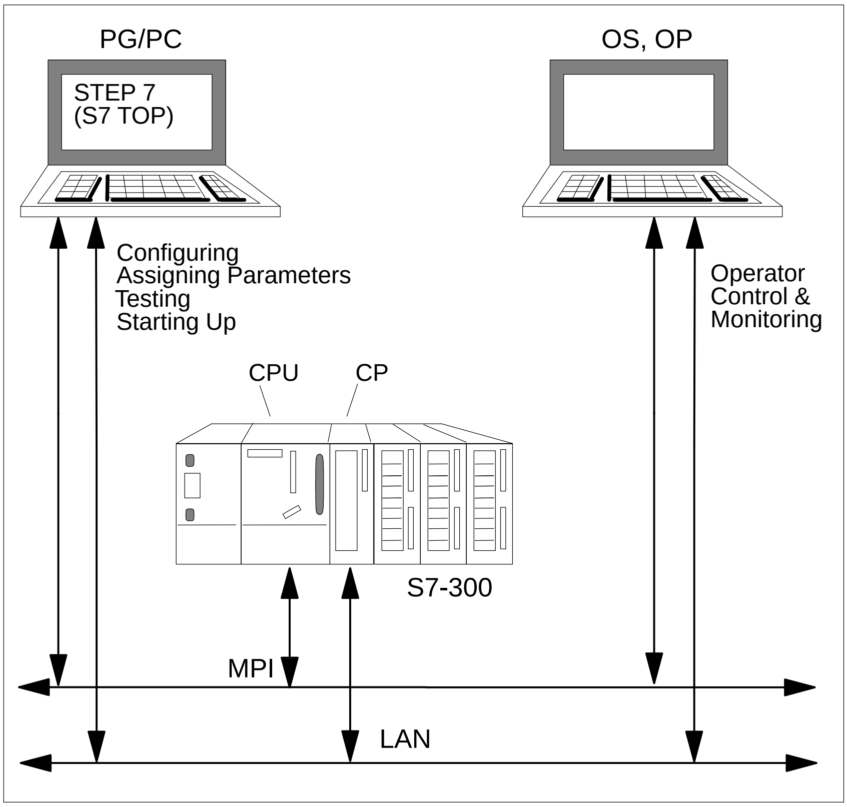


Figure 2-4 Working Environment for the Software Package “Fuzzy Control”

System Variables

Creating fuzzy functions on a software basis always involves calculation-intensive operations (word processing). This means it is important to check that the relevant CPU is capable of coping with the fuzzy function. Refer to the technical specifications in Section 3.4.

- The speed at which concrete fuzzy applications can be processed depends entirely on the performance of the CPU used.

  With a given CPU a compromise must always be made between the number of fuzzy applications and the frequency with which a single system is to be processed. The more often the output variables have to be calculated per unit of time, the smaller the number of fuzzy applications which can be installed.
- The required size of the user memory area and the number of fuzzy applications that can theoretically then be installed (with 50% utilization of the working memory by the automation tasks) can be calculated from the code length of the fuzzy function block and the length of the instance data block.

Environment for Creating the Control Block

The hardware and software requirements for creating and configuring the controller block are described in Section 4.3.

3 The Fuzzy Control Function Blocks

What Topics Are Covered in this Chapter?

This chapter describes:

- The Fuzzy Control function blocks
- The technical specifications of Fuzzy Control

3.1 The Function Block FUZZY

General Information on Functionality

In the standard function block FUZZY all the algorithms and procedures are implemented that are necessary to perform all the functions of a high-performance fuzzy application:

- Fuzzification of the inputs
- Editing the rules
- Defuzzification and value output at the outputs

You can edit the following in the set of rules or rule basis:

- A maximum of 8 inputs, each with up to 7 membership functions
- A maximum of 4 outputs, each with up to 9 membership functions
- A selectable number of rules (maximum of 200 rules with 2 inputs and 1 output)

Knowledge Required

The following tasks are part of creating a fuzzy system:

- Defining the membership functions
- Formulating the rules
- Calling the application at the correct time in the system program

You create a fuzzy system with a minimum of programming effort using a configuration tool. However, previous experience of working with the STEP 7 Program Editor is necessary.

The functionality of Fuzzy Control is carefully limited to the definition and execution of core functions of the fuzzy theory. This makes fuzzy applications on the basis of Fuzzy Control easy to configure. This only applies to the activities and procedures required to formally create a project.

This does not, however, apply to:

- The quantitative definition of the assignment of the values of the input variables to fuzzy sets
- The qualitative setting up of the rules governing how these sets are mapped to fuzzy sets for the output variables

An imperative requirement for coping successfully with this part of the project is empirical knowledge of the process behavior or experience as an operator of controlling the relevant plant manually.

Block Structure

An instance data block in the CPU of the programmable controller forms the interface between the function block, the configuration tool, and the user. You must enter the addresses of the inputs in this data block before the function block is called. When the function block has been executed, you can then read off the values stored at the outputs from the instance data block.

The membership functions and the rules are entered ’implicitly’ in the instance data block by the configuration tool. You can download a number of fuzzy applications to a CPU and run them there. Each application is stored in a separate data block; the number of the data block can be freely assigned.

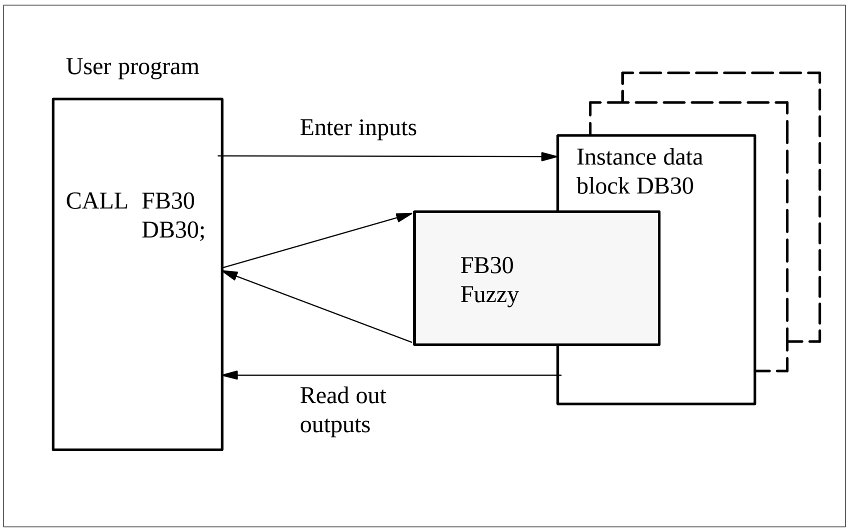


Figure 3-1 Structure of the Block Calls

Calling the Function Block FUZZY

1. The function block **FUZZY** must be called by the user. The call can be programmed in the cyclic or time-driven program processing level. The function block must be called absolutely.
2. When you call the function block you must specify the required instance data block (Fuzzy data block) that contains the application-specific fuzzy project created using the configuration tool.
3. You only need to specify the parameters required by the set of rules. Any inputs or outputs which are not assigned do not have to be connected.
4. Example:
   A minimal call with the parameter INFO would have the following listing:

```
STL                                                    Explanation
CALL FB30, DB30
(            INFO := MB30);
```

For a new fuzzy application, the function block must enter identifications in the data block. Once these entries have been made, the data block is recognized by the configuration tool as a fuzzy data block and processed.

Note

You can assign the number of the function block FUZZY freely, within the bounds of the CPU. The default is FB30. The block can only be renamed in the dialog box “Function Block Type Description” which appears when you select the menu command “Save As”.

The function block FUZZY must be called absolutely in the user program. It is controlled via the variable START_STOP in the instance data block.

External Supply to Inputs/Outputs

If a data block has been created as an instance data block and if a symbolic name has been assigned, external access to the fuzzy application in the user program is also possible.

Example:

```
 ..                                  Symbol for the DB
 ..
T      ”Pendulum”.INPUT1
 ..                                  Variable in the DB
L      ”Pendulum”.OUTPUT2
 ..
etc.
```

Controlling Processing

Processing of the fuzzy application must be controlled via the variable START_STOP in the data block. This variable can be modified or evaluated. The configuration tool can also modify the variable START_STOP.

You can influence processing of the fuzzy application directly via the value of the variable START_STOP:

START_STOP	Meaning
= W#16#0000	The fuzzy application is not processed
≠ W#16#0000	The fuzzy application is processed

Note

If a fuzzy application is not to be processed cyclically, processing can be controlled with the content of the variable START_STOP: for example, by calling the function block in OB1 and by creating a time slice in the time-driven organization block.

Influence from the Configuration Tool

Execution of the fuzzy application in the processor is also controlled by the configuration tool. Execution is stopped by the entry “W#16#0000” in the variable START_STOP before the application is transferred from the programming device/PC. Following the transfer the tool enters the value “W#16#FFFF” in the variable START_STOP and enables execution again. Example:

```
STL                                 Explanation
L    0
T    ”Pendulum”.START_STOP          Do not process set of rules

l    123
T    ”Pendulum”.START_STOP          Process set of rules

l    ”Pendulum”.START_STOP
l    W#16#FFFF
==I
=    M 10.0                         Modify the fuzzy application
                                    using the configuration tool
```

Note

Execution is always enabled by the configuration tool following a transfer, even if the content of the variable START_STOP = W#16#0000 was set by the user program before the transfer.

Evaluating “INFO”

The function block gives out information about the status of the fuzzy application via the parameter INFO. This information is divided into three categories: no error, warning, error.

No error
If the fuzzy application was executed without error, the content of the variable INFO is B#16#00.

Warning
If the content of the variable START_STOP is W#16#0000 (fuzzy application is not being processed), the content of the parameter INFO is B#16#01. This shows you that the contents of the outputs has not been recalculated but still consists of old values.

Error

– If an instance data block of the correct length is available in the CPU but a fuzzy application has not yet been downloaded from the configuration tool, the content of the INFO parameter is B#16#11.

– If the length of the specified instance data block is not sufficient, the content of the INFO parameter is B#16#21#.

The output INFO supplies information about the fuzzy application. It has the data type BYTE and has the default value B#16#00. The various meanings of the INFO parameter can be seen in the following table:

Contents (B#16#...)	Meaning
00	No error occurred during processing
01	Processing of the set of rules by the user program or by the configuration tool disabled
11	No valid set of fuzzy rules in the instance data block
21	Length of the data block not sufficient (not a fuzzy data block)

Note

If an error is found or a warning occurs, the outputs are not cleared by the function block. Having evaluated INFO, you must decide whether the output values (old values) are to be processed or whether a defined value is to be output.

Danger

If the function block finds a warning or an error, execution of the function block is stopped immediately.

Evaluating the Status Bit BR (Ladder Parameter ENO)

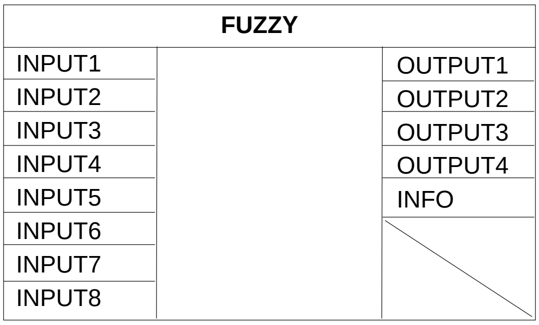
The function block influences the status bit BR. You can evaluate the status after the function block has been called:

BR 1 The function block has not found an error or a warning during execution. The content of the INFO parameter is B#16#00.

BR 0 The function block has found an error or a warning during execution. The content of the INFO parameter is not equal to B#16#00.

Block Diagram and Parameters of the Function Block FUZZY

The function block FUZZY has the following block diagram:



Input Parameters

The following table shows you the data type and the structure of the input parameters of the function block FUZZY.

Table 3-1 Input Parameters of the Function Block FUZZY (Fuzzy Data Block)

Byte	Parameter	Data Type	Explanation	Default
0	INPUT1	REAL	1st input of the fuzzy application	0.0
4	INPUT2	REAL	2nd input of the fuzzy application	0.0
8	INPUT3	REAL	3rd input of the fuzzy application	0.0
12	INPUT4	REAL	4th input of the fuzzy application	0.0
16	INPUT5	REAL	5th input of the fuzzy application	0.0
20	INPUT6	REAL	6th input of the fuzzy application	0.0
24	INPUT7	REAL	7th input of the fuzzy application	0.0
28	INPUT8	REAL	8th input of the fuzzy application	0.0

Output Parameters

The following table shows you the data type and the structure of the output parameters of the function block FUZZY.

Table 3-2 Output Parameters of the Function Block FUZZY (Fuzzy Data Block)

Byte	Parameter	Data Type	Explanation	Default
32	OUTPUT1	REAL	1st output of the fuzzy application	0.0
36	OUTPUT2	REAL	2nd output of the fuzzy application	0.0
40	OUTPUT3	REAL	3rd output of the fuzzy application	0.0
44	OUTPUT4	REAL	4th output of the fuzzy application	0.0
48	INFO	BYTE	Information about processing	B#16#0

Additional Parameters

In addition to the input and output parameters the function block has two other parameters:

Table 3-3 Control Parameters and Internal Area of the Function Block FUZZY (Fuzzy Data Block)

Byte	Parameter	Data Type	Explanation	Default
50	START_STOP	WORD	>< 0 Process fuzzy application == 0 Do not process fuzzy application	W#16#0
52 to 2126	FUZZY	BYTE	Internal area for the function block	B#16#0

3.2 Inputting an Analog Value (Function FUZZY_AI)

Application

The following condition must be fulfilled:

A sensor or an analog input module supplies an input value as a physical variable that has to be processed further.

Then:

The analog variable and its measurement range or the I/O value (number range –27648 to 27648) must be normalized to the real value required for further processing by Fuzzy Control.

The Function FUZZY_AI

The function (or FC) FUZZY_AI normalizes an analog input value to the lower and upper limit, i.e. the values MINIMUM and MAXIMUM with which the value range of the relevant input was defined. Then the normalized value is entered in the instance data block.

The block (FC) must be called in the user program for every input channel of the fuzzy application that is connected to an analog variable.

Note

You can assign the number of the function FUZZY_AI freely, within the bounds of the CPU. The default is FC30.

If the function is used you must connect all parameters when it is called.

Normalization and Signal Type

Using the input parameter THERM_EL you can determine what the normalization should use as a reference:

either:

- Unipolar and/or bipolar voltage or current signals

or:

- Thermocouples as sensors at the relevant analog output

With the aid of the parameter BIPOLAR you can specify whether a unipolar or bipolar voltage or current signal should be processed at the relevant input.

The valencies of the binary input parameters THERM_EL and BIPOLAR have the following effects:

THERM_EL	BIPOLAR	Meaning or Effect
FALSE (:= 2#0)	FALSE (:= 2#0)	Unipolar voltage or current signal at the input The function FUZZY_AI normalizes the signal.
FALSE (:= 2#0)	TRUE (:= 2#1)	Bipolar voltage or current signal at the input The function FUZZY_AI normalizes the signal.
TRUE (:= 2#1)	FALSE (:= 2#0)	Thermocouple at the input Normalization is performed in the analog input module. The function only performs an adaptation via the parameters FACTOR and OFFSET.

Voltage or Current Signal at the Analog Input

The function FUZZY_AI normalizes the analog input signal in this case to the value range of the relevant input channel of the fuzzy application. The limits MINIMUM and MAXIMUM that define the value range are read from the fuzzy data block. They are processed as limit values in the function. The normalized value is then entered in the fuzzy data block for the respective input.

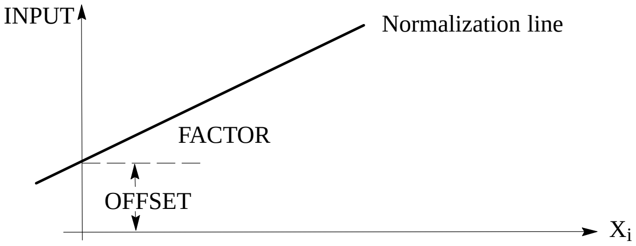
The normalizing value INPUT for **bipolar** signals is calculated from the respective input value X_i:

$$\text{INPUT} = \left(\frac{X_{min} * (27648 - X_i) + X_{max} * (X_i + 27648)}{2 * 27648}\right) * \text{FACTOR} + \text{OFFSET}$$

The corresponding relationship for **unipolar** signals is:

$$\text{INPUT} = \left(\frac{X_{min} * (27648 - X_i) + X_{max} * X_i}{27648}\right) * \text{FACTOR} + \text{OFFSET}$$

Where:		
	INPUT	Normalized value which is entered in the instance data block
	X_{max}	Upper limit of the input value range
	X_{min}	Lower limit of the input value range
	FACTOR	Factor (gradient) of the normalization lines
	OFFSET	Offset of the normalization lines at the zero point
	X_i	Value supplied by the analog module



The values for the parameters FACTOR and OFFSET are passed when the function is called.

Note

If analog signals with a bipolar range (e.g. ±10 V/ ±20 mA) are connected as an input signal, the MINIMUM and MAXIMUM values for the value range of the respective input must be given symmetrically (e.g. -100.0 and +100.0). If this is not the case, input values (INPUT) may be calculated incorrectly following the use of these values in the function FUZZY_AI.

Example of a call of the function FUZZY_AI for an input with a voltage or current signal:

```
STL                                                Explanation
call fc30 (
               AI_ADR   := 348,
               BIPOLAR  := TRUE,
               THERM_EL := FALSE,
               FACTOR   := 1.0,
               OFFSET   := 0.0,
               FUZZY_DB := db30,
               INPUT_NR := 1,
               INFO     := mb 30
               );
```

Thermocouple Signal at the Analog Input

Temperature signals are supplied from thermocouples and recorded via an analog module. When they are connected as an analog input, the normalization in °C is already performed in this module.

The function FUZZY_AI performs a conversion of the I/O value from WORD to REAL in this case, or from fixed-point numbers (–27648 to 27648) to real numbers. The temperature is adjusted via the parameters FACTOR and OFFSET.

Thermocouple	Measurement range
Pt 100	–200.0° C to 850.0° C
Ni 100	–60.0° C to 250.0° C
Type K (NiCr-Ni)	–270.0° C to 1372.0° C
Type N (NiCrSi-NiSi)	–270.0° C to 1300.0° C
Type J (Fe-CuNi IEC)	–210.0° C to 1200.0° C
Type E (NiCr-CuNi)	–210.0° C to 1000.0° C

The normalization value INPUT for thermocouples as temperature sensors is calculated from the respective input value X_i according to the following relationship:

$$\text{INPUT} = \frac{X_i}{10.0 * \text{FACTOR} + \text{OFFSET}}$$

Where:		
	INPUT	Normalized value which is entered in the instance data block
	FACTOR	Factor (gradient) of the normalization lines
	OFFSET	Offset of the normalization lines at the zero point
	X_i	Value supplied by the analog module

The normalization of the thermocouple in °C is performed in the analog input module and FUZZY_AI only makes an adjustment in this case. This means you can freely select the values MINIMUM and MAXIMUM of the relevant fuzzy input within the scope of the measurement range.

Example:

Measurement range Pt100: –200.0° C to 850.0° C
Technology range: 80° C to 100° C (boiling water)
Fuzzy INPUT MINIMUM = 75° C to MAXIMUM = 105° C

Example of a call of the function FUZZY_AI for an input with an analog signal from thermocouple sensors:

```
STL                                               Explanation
call fc30 (
               AI_ADR   := 348,
               BIPOLAR  := FALSE,
               THERM_EL := TRUE,
               FACTOR   := 1.0,
               OFFSET   := 0.0,
               FUZZY_DB := db30,
               INPUT_NR := 1,
               INFO     := mb 30
               );
```

Controlling Processing with the Function (FC)

Processing of the function FUZZY_AI must be controlled via the variable START_STOP in the fuzzy data block. This variable can be modified and evaluated by the user program. The configuration tool can also modify the variable START_STOP.

You can influence processing of the function FUZZY_AI directly via the value of the variable START_STOP:

START_STOP	Meaning
= W#16#0000	FUZZY_AI is not processed
≠ W#16#0000	FUZZY_AI is processed

Influence from the Configuration Tool

Execution of the function in the processor is also controlled by the configuration tool. Execution is stopped by the entry W#16#0000 in the variable START_STOP before the application is transferred from the programming device/PC. Following the transfer the tool enters the value W#16#FFFF in the variable START_STOP and enables execution again.

Execution is always enabled by the configuration tool following a transfer, even if the content of the variable START_STOP = W#16#0000 was set by the user program before the transfer.

Evaluating “INFO”

The function FUZZY_AI gives out information about the status of the fuzzy application via the parameter INFO. This information is divided into three categories: no error, warning, error.

No error
If the function was executed without error, the content of the variable INFO is B#16#00.

Warning
If the content of the variable START_STOP is W#16#0000 (fuzzy application is not being processed), the content of the parameter INFO is B#16#01. This shows you that the contents of the outputs has not been recalculated but still consists of old values.

Error

– If an instance data block of the correct length is available in the CPU but a fuzzy application has not yet been downloaded from the configuration tool, the content of the INFO parameter is B#16#11.
– If the length of the specified instance data block is not sufficient, the content of the INFO parameter is B#16#21#.
– If an input number (INPUT_NR) does not lie within the range 1 to 8, the content of the INFO parameter is set to B#16#31 by the function.

Meaning of the INFO parameter:

INFO	Meaning
00	No error occurred during processing
01	Processing of the fuzzy application by the user program or by the configuration tool disabled
11	No valid fuzzy application in the instance data block
21	Length of the data block not sufficient (not a fuzzy data block)
31	Parameter INPUT_NR not in the range 1 to 8

Note

If an error is found or a warning occurs, the input values are not entered by the function. Having evaluated INFO, you must decide whether the old values are to be processed or whether a defined value is to be used.

Danger

If the function finds a warning or an error, execution of the function is stopped immediately.

Evaluating the Status Bit BR (Ladder Parameter ENO)

The function influences the status bit BR. You can evaluate the status after the function has been called:

BR 1 An error or a warning has not been found during execution of the function. The content of the INFO parameter is B#16#00.

BR 0 The function block has found an error or a warning during execution. The content of the INFO parameter is not equal to B#16#00.

Block Diagram and Parameters of FUZZY_AI

The function FUZZY_AI has the following block diagram:

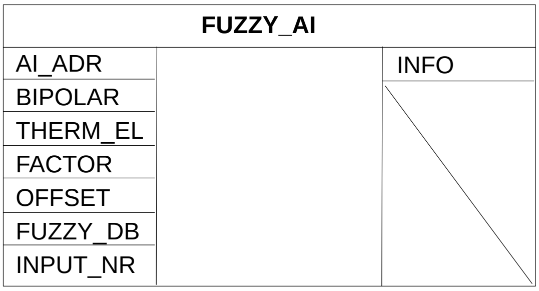


Table 3-4 Input Parameters of FUZZY_AI

Parameter	Data Type	Explanation	Permissible Value Range	Default
AI_ADR	INT	Address of the analog input		
BIPOLAR	BOOL	Measurement value bipolar or unipolar		FALSE
THERM_EL	BOOL	Analog input connected to one thermocouple measurement value		FALSE
FACTOR	REAL	Factor for adapting the measurement value	Whole value range (no dimensions)	1.0
OFFSET	REAL	Offset for adapting the measurement value	Whole value range (no dimensions)	0.0
FUZZY_DB	BLOCK_DB	Number of the fuzzy data block		DB30
INPUT_NR	INT	Number of the fuzzy input	1 to 8	1

Table 3-5 Output Parameters of FUZZY_AI

Parameter	Data Type	Explanation	Default
INFO	BYTE	Information about processing	

3.3 Outputting an Analog Value (Function FUZZY_AO)

Application

If an output value from an analog output module is to be processed further, the function FUZZY_AO can be used to normalize the fuzzy output variables in real number format to the I/O range.

The Function FUZZY_AO

The function FUZZY_AO normalizes an output value to the relevant analog output range. The lower and upper limits for the range are read from the fuzzy data block and processed in the function as limit values. The normalized value is then adjusted with the parameters FACTOR and OFFSET and written to the analog output.

Note

You can assign the number of the function FUZZY_AO freely, within the bounds of the CPU. The default is FC31.

If the function is used you must connect all parameters when it is called.

Normalization and Signal Type

With the aid of the parameter BIPOLAR you can specify whether a unipolar or bipolar voltage or current signal should be created at the relevant output (via an analog output module). The valency of the binary input parameter BIPOLAR has the following effects:

BIPOLAR	Meaning or Effect
FALSE (:= 2#0)	Unipolar signal at the output FUZZY_AO normalizes the signal.
TRUE (:= 2#1)	Bipolar signal at the output FUZZY_AO normalizes the signal.

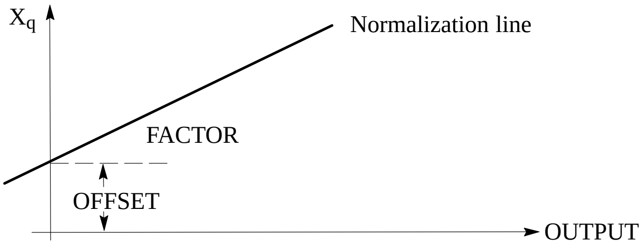
The function FUZZY_AO calculates the normalization value X_q for **bipolar** signals which is transferred to the analog output module. The calculation is performed according to the relationship shown here from the current value OUTPUT of the fuzzy output:

$$X_q = \left(\frac{27648 * (2 * \text{OUTPUT} - X_{max} - X_{min})}{X_{max} - X_{min}}\right) * \text{FACTOR} + \text{OFFSET}$$

The corresponding relationship for **unipolar** signals is:

$$X_q = \left(\frac{27648 * (\text{OUTPUT} - X_{min})}{X_{max} - X_{min}}\right) * \text{FACTOR} + \text{OFFSET}$$

Where:	OUTPUT	Output value of the fuzzy application
	X_{max}	Upper limit of the fuzzy output
	X_{min}	Lower limit of the fuzzy output
	FACTOR	Factor (gradient) of the normalization lines
	OFFSET	Offset of the normalization lines at the zero point
	X_q	Value written to the analog module



The values for the parameters FACTOR and OFFSET are passed when the function is called.

Note

If analog signals with a bipolar range (e.g. ±10 V/ ±20 mA) are used as a fuzzy output, the MINIMUM and MAXIMUM limits of the output must be given symmetrically (e.g. -100.0 and +100.0). If this is not the case, output values (X_q) may be calculated incorrectly following the use of these values in the function FUZZY_AO.

Example of a call of the function FUZZY_AO for an output with a bipolar signal:

```
STL                                              Explanation
call fc31 (
               AI_ADR    := 348,
               BIPOLAR   := TRUE,
               FACTOR    := 1.0,
               OFFSET    := 0.0,
               FUZZY_DB  := db30,
               OUTPUT_NR := 1,
               INFO      := mb 30
               );
```

Controlling Processing

Processing of the function FUZZY_AO must be controlled via the variable START_STOP in the fuzzy data block. This variable can be modified and evaluated by the user program. The configuration tool can also modify the variable START_STOP.

You can influence processing of the function FUZZY_AO directly via the value of the variable START_STOP:

START_STOP	Meaning
= W#16#0000	FUZZY_AO is not processed
≠ W#16#0000	FUZZY_AO is processed

Influence from the Configuration Tool

Execution of the function in the processor is also controlled by the configuration tool. Execution is stopped by the entry W#16#0000 in the variable START_STOP before the application is transferred from the programming device/PC. Following the transfer the tool enters the value W#16#FFFF in the variable START_STOP and enables execution again.

Execution is always enabled by the configuration tool following a transfer, even if the content of the variable START_STOP = W#16#0000 was set by the user program before the transfer.

Evaluating “INFO”

The function FUZZY_AO gives the user program information about the status of the fuzzy application via the parameter INFO. This information is divided into three categories: no error, warning, error.

No error
If the function was executed without error, the content of the variable INFO is B#16#00.

Warning
If the content of the variable START_STOP is W#16#0000 (fuzzy application is not being processed), the content of the parameter INFO is B#16#01. This shows you that the contents of the outputs has not been recalculated but still consists of old values.

Error

– If an instance data block of the correct length is available in the CPU but a fuzzy application has not yet been downloaded from the configuration tool, the content of the INFO parameter is B#16#11.
– If the length of the specified instance data block is not sufficient, the content of the INFO parameter is B#16#21#.
– If an output number (OUTPUT_NR) does not lie within the range 1 to 4, the content of the INFO parameter is set to B#16#31 by the function.

Meaning of the INFO parameter:

INFO (B#16#...)	Meaning
00	No error occurred during processing
01	Processing of the fuzzy application by the user program or by the configuration tool disabled
11	No valid fuzzy application in the instance data block
21	Length of the data block not sufficient (not a fuzzy data block)
31	Parameter OUTPUT_NR not in the range 1 to 4

Note

If an error is found or a warning occurs, the output values are not entered by the function. Having evaluated INFO, you must decide whether the old values are to be processed or whether a defined value is to be used.

Danger

If the function finds a warning or an error, execution of the function is stopped immediately.

Evaluating the Status Bit BR (Ladder Parameter ENO)

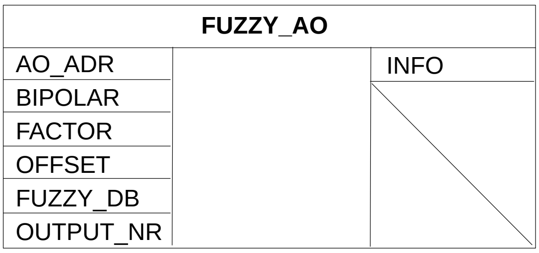
The function influences the status bit BR. You can evaluate the status after the function has been called:

BR 1 An error or a warning has not been found during execution of the function. The content of the INFO parameter is B#16#00.

BR 0 The function block has found an error or a warning during execution. The content of the INFO parameter is not equal to B#16#00.

Block Diagram and Parameters of FUZZY_AO

The function FUZZY_AO has the following block diagram:



Input Parameters

The following table shows you the data type and the structure of the input parameters of FUZZY_AO.

Table 3-6 Input Parameters of FUZZY_AO

Parameter	Data Type	Explanation	Permissible Value Range	Default
AO_ADR	INT	Address of the analog output		
BIPOLAR	BOOL	Output signal bipolar or unipolar		FALSE
FACTOR	REAL	Factor for adjusting the output signal		1.0
OFFSET	REAL	Offset for adjusting the output signal	Whole value range (no dimensions)	0.0
FUZZY_DB	BLOCK_DB	Number of the fuzzy data block	Whole value range (no dimensions)	0.0
OUTPUT_NR	INT	Number of the fuzzy output	1 to 4	1

Output Parameters

The following table shows you the data type and the structure of the output parameters of FUZZY_AO.

Table 3-7 Output Parameters of FUZZY_AO

Parameter	Data Type	Explanation	Default
INFO	BYTE	Information about processing	

3.4 Technical Specifications of the Function Block FUZZY

Technical Data

Block name	FUZZY FB 30	FUZZY_AI FC 30	FUZZY_AO FC 31
Family	FUZZYCON	FUZZYCON	FUZZYCON
Block length at runtime	1024 bytes	370 bytes	344 bytes
Block length in memory	1226 bytes	510 bytes	478 bytes
Call length	–		
Instance data block or shared data block	DB_FUZZY		
Family	FUZZYCON		
Block length at runtime	2126 bytes		
Block length in memory	2248 bytes		

Execution Times

The table below shows the typical execution times when the blocks are assigned parameters for 2 inputs, 1 output, and 25 rules.

Block name	FUZZY FB 30	FUZZY_AI FC 30	FUZZY_AO FC 31
Family	FUZZYCON	FUZZYCON	FUZZYCON
Execution time (in ms) CPU 314	13	3	3
Execution time (in ms) CPU 416	0.7	1.3	1.3

4

Product Overview

What Topics Are Covered in this Chapter?

This chapter describes:

- The purpose and the capabilities of the configuration tool
- Its structure
- Where it can be used

4.1 Purpose and Capabilities of the Configuration Tool

Purpose

The configuration tool allows you to create and start up fuzzy applications. With it you can define the inputs and outputs of the standard function block FUZZY and formulate and edit the set of rules or the rule basis for the corresponding fuzzy application.

To solve automation tasks using fuzzy concepts you will require a tool that supports you effectively in configuring and setting fuzzy applications. The functions of the configuration are divided up to make processing easier and it has a number of monitoring facilities. This makes it easier and safer to perform the steps required in setting up the applications.

Capabilities

The configuration and processing tool “Configuration Fuzzy Control” meets the highest demands from the user, but at the same time is also suitable for less experienced users who require a simple, user-friendly approach.

The tool was specifically designed to be easy to use, giving you the freedom to concentrate on the actual solving of your automation task.

The full range of capabilities of the configuration tool is split into sub-functions for processing the various tasks involved in creating fuzzy applications. Each of these functions runs in its own window:

Basic Functions

Opening a project or calling it for processing.

- Generating and naming a new fuzzy application or calling an existing fuzzy application to process it
- Configuring the structure of the fuzzy application

Editing Inputs

Defining the membership functions of the inputs.

- Naming the inputs
- Determining the value ranges of the inputs
- Defining the membership functions of the inputs

Editing Rules

Formulating or editing the set of rules.

- Formulating fuzzy rules by selecting linguistic values
- Editing the rules by adding, deleting, or modifying

Editing Outputs

Defining the membership functions of the outputs.

- Naming the outputs
- Determining the value ranges of the outputs
- Defining the membership functions of the outputs

3D Graphic Display

Static representation of the cause-effect relationship between several input variables and one output variable.

- Assigning the input/output variables to the display axes
- Setting family parameters
- Animation (rotation) of the spatial representation

Curve Recorder

Dynamic representation of up to five variables in a time curve and an output variable in a bar diagram.

- Time-dependent display of up to five process variables in a scaleable time curve
- Monitoring archive values by visualizing past events
- Bar diagrams showing singleton functions for a selected output

Off-Line Test

Defining triangular functions that can have parameters assigned in order to connect inputs in the off-line test of the fuzzy application.

- Defining a test function (triangular function) according to amplitude, form, and frequency, and passing it on to the relevant input

4.2 Structure of the Configuration Tool

Program Structure

The user interface corresponds to the user interface of STEP 7. All functions are activated by means of windows and menus.

The structure of the configuration tool showing its most important editing and monitoring windows is as follows:

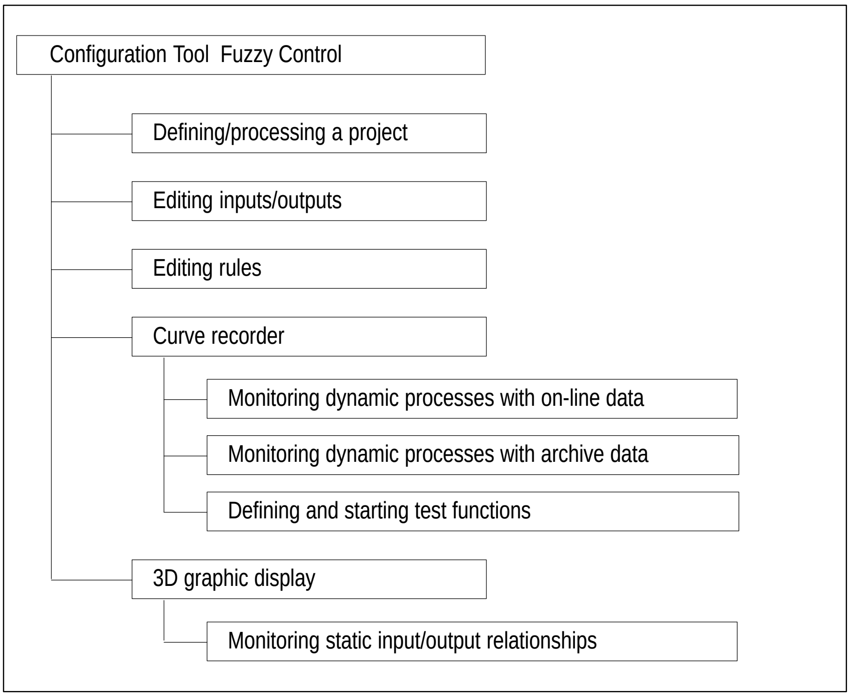

Configuration Tool Fuzzy Control
Defining/processing a project
Editing inputs/outputs
Editing rules
Curve recorder
Monitoring dynamic processes with on-line data
Monitoring dynamic processes with archive data
Defining and starting test functions
3D graphic display
Monitoring static input/output relationships

Figure 4-1 Structure of the Configuration Tool

Data Structure

All the data required for the function block FUZZY are stored in an instance data block which is assigned to the function block. The entries in this data block are made implicitly by the configuration tool, meaning you do not need to make any entries in the data block during configuration of fuzzy applications.

If the data block is stored in the programming device or PC, only off-line access to the respective fuzzy application is possible.

If the data block is downloaded to a CPU with the use of STEP 7 functions, on-line access is also possible.

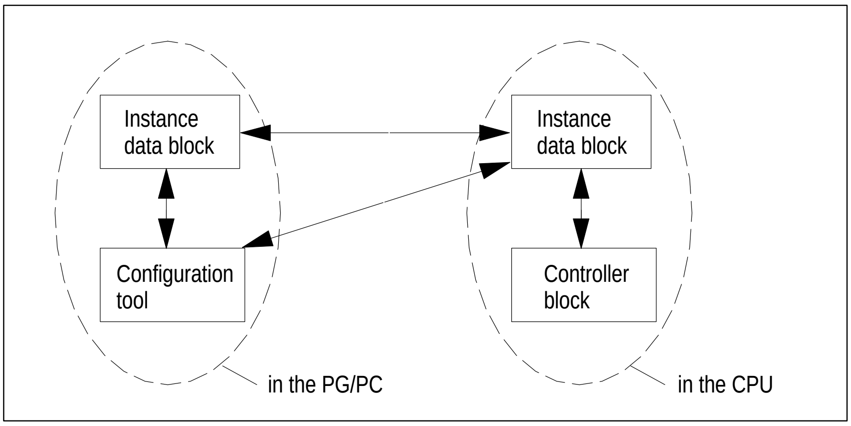


Figure 4-2 Data Structure

4.3 Working Environment

Software and Hardware Requirements

The configuration tool for fuzzy applications requires the same environment as STEP 7.

It can be installed locally on a PC or programming device or in a network on a central network drive.

5 The Fuzzy Control Configuration Tool

What Topics Are Covered in this Chapter?

This chapter describes:

- The tool “Configuration Fuzzy Control”

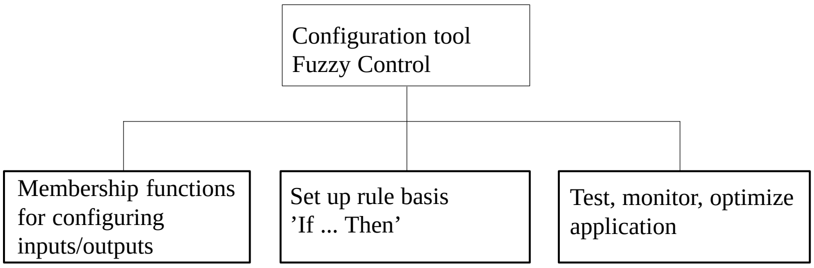
5.1 The Tool “Configuration Fuzzy Control”

Scope of Functions

The various functions of the configuration tool can be selected using the graphic user interface with ergonomically designed windows in which the parameters and rules of a fuzzy system can be defined. The most important steps in processing each fuzzy application are:

- Defining input and output variables
- Configuring fuzzy membership functions for inputs and outputs
- Setting up the fuzzy rule basis
- Performing an off-line analysis of the draft

The following diagram shows the sub-functions of the configuration tool, each of which has its own special editing windows:


Configuration tool Fuzzy Control
Membership functions for configuring inputs/outputs
Set up rule basis ’If ... Then’
Test, monitor, optimize application

Functional Structure and User Interface

Each subtask used to create a fuzzy application is processed in its own window. In the main window that appears after starting the tool you enter the input/output structure of the fuzzy application directly. You call the file functions and create the required entries for values of parameters and variables via the menus shown. You can also display representations of input/output relationships and time curves.

- **Configure membership functions for inputs/outputs**

The number of inputs and outputs to be generated is defined via the menu “Edit”. You can then define the membership functions for each input and output by entering parameters in a special window:

- Dialog window: “Edit Input”
- Dialog window: “Edit Output”

- **Set up rule basis ’If ... Then’**

The empirical relationships between inputs and outputs are collected in a table. If required, they can be modified once a field in the table is selected:

– Dialog window “Edit Rules”

- **Test, monitor, and optimize an application**

The window containing the 3D or 4D display can be called via the “View” menu. This visualizes the static relationships between inputs and a selected output as defined in the rule basis.

– Dialog window: “3D Graphic Display”

The menu commands to represent a curve enable static and time-dependent parameters to be selected and entered. These parameters serve to represent the cause-effect relationships between test functions and on-line process variables and the behavior of the respective output variables.

– Dialog window: “Curve Recorder”

Configuring and Starting Up Fuzzy Applications

6

What Topics Are Covered in this Chapter?

This chapter describes:

- Drafting and designing fuzzy applications
- Opening and calling a project
- Defining the application structure and the inputs and outputs
- Defining the membership functions
- Formulating the set of rules
- The off-line analysis of the fuzzy application: display and monitoring
- Downloading and starting up the fuzzy application

6.1 Drafting and Setting Out Fuzzy Applications

Specifying the Task

Before you create a fuzzy application with Fuzzy Control you should have clearly planned your process control or automation task, decided on the programmable controller you intend to use, and planned the operating and monitoring environment. To be able to specify your task, you will need:

1. An exact analysis of the automation task, meaning a calculation of the characteristics of the plant (behavior at different operating points, consideration of influencing variables depending on situation, changing strategy depending on status, etc.).
2. Knowledge of the behavior of the process variables relevant for solving the task, and a decision on the inputs and outputs to be used in the fuzzy system.
3. To decide on the CPU that the respective fuzzy application is to be installed and processed on.
4. If relevant, normalization or de-normalization of the I/O signals (inputs and outputs) with regard to the unified internal number range in REAL format.

Previous Knowledge and Requirements

With “Fuzzy Control” software application systems are generated on the basis of a standard function block (FB30 in this case) from the S7 block world. This means knowledge of working with S7 blocks and the structure of S7 user programs (for example, in the S7 programming language representation Statement List) is necessary.

The functionality of the fuzzy application you want to create is defined entirely by assigning parameters. Despite this, connecting the fuzzy block with the process I/O and linking it into the call hierarchy of the CPU require knowledge of procedures which are not described in this manual. You will need to have:

– Information on working with STEP 7
/User Manual: Standard Software for S7 and M7, STEP 7/;

– Basic information on programming with STEP 7
/Programming Manual: System Software for S7-300 and S7-400 Program Design/, /Manual: Statement List (STL) for S7-300 and S7-400, Programming/;

– Data about the programmable controller you intend to use

Planning Sequence

The systematic drafting and setting out of a fuzzy application require a procedure that can be performed in the following logical planning steps:

- Analyze the automation task to be solved:
  Closed-loop control - open-loop control - parameter adjustment - generating reference variables - strategy changes - feedforwarding manipulated variables - etc.
- Employ evaluation criteria for the use of a fuzzy application

- Define the process variables (naming, normalization, value range, linguistic values)
- Quantify the linguistic values by defining membership functions
- Formulate the rule basis
- Perform off-line analysis of the generated fuzzy application
- Download and start up the fuzzy application
- Correct and trim the parameters which determine the behavior of the "Fuzzy Control" system running on the programmable controller in on-line operation

Analysis of the Automation Task

Some of the main aims of fuzzy applications are the complete automation of technical processes (without the operator having to intervene in individual cases), the improvement of control quality, the optimum time control of movement and transport processes, the increase of productivity and product quality, and the sinking of energy costs.

These aims can be achieved using fuzzy applications for specific automation tasks, for example:

- Automation systems which depend on linking in many influencing variables to form an image of the process
- Systems with a number of degrees of freedom (strategies) which require several different process statuses to be satisfied in order to control them
- Time-variant or non-linear automation structures that cannot be fully automated because a mathematical model of the process does not exist or would be too complex to create
- Control loops that require partial manual intervention or in which manual intervention would have a negative effect on the product quality (for example, rejects following "poor" manual operation)
- Multi-variable systems that one plant operator cannot operate safely
- Complex start-up and shut-down of processes (for example, chemical reactors or autoclaves) and subplants that could previously only execute and be controlled using manual operation

The cause-effect relationships are easily assigned in fuzzy applications, giving you the opportunity to easily correct unwanted process behavior at different operating statuses. The system can react differently to particular statuses of your plant. Fuzzy applications are generally very easy to create and easy to understand. Even after a number of years you can still see which rules have which meaning.

Assessment Criteria for Fuzzy Application

For the use of a fuzzy application to be successful and produce optimum results, it is vital during the design phase to assess the automation task according to the following criteria:

- Have the solutions to the task been exhausted by conventional methods (for example, PID controllers, P/PI status controllers) and optimizing procedures (for example, setpoint control, feedforward control, parameter adaptation)?
- Does the plant operator have sufficient expertise to derive the rule basis for the fuzzy application?

  Note: It is important that he should be able to “verbalize” this knowledge; simply having “a feel for the job” but not being able to express it is not sufficient.
- Cost-benefit ratio: the benefits are supplied by the expected economic advantages that result from achieving the automation goals (savings in personnel and energy, better product quality etc.). The costs are caused by the increased engineering effort in comparison to conventional solutions which can reach a number of man months for complex fuzzy applications.
- To calculate the advantages of a fuzzy solution it is also necessary to measure and evaluate the current situation (manual operation, control quality). By logging the interventions of the plant operator you can also check whether his statements, which are being used for the rule basis, are plausible.

Defining the Process Variables

A fuzzy application is a multi-dimensional system of graphs each with a maximum of eight inputs and four outputs. To link a Fuzzy Control application (FB30) into the programmable logic control system it is necessary to calculate and specify the process variables to be assigned to its inputs and outputs.

If the required process variables are not already available in the programmable controller, the missing I/O variables must be fed to the programmable control system via additional sensors and analog input modules. Examples of this are:

- Laboratory analysis values of raw materials
- Energy consumption values for the concern and the plant available, for example, in host computers
- Environmental influences such as the effects of weather on fermentation processes

These variables must be defined using the STEP 7 software. Then they must be represented in the user memory (symbolically or absolutely) and connected to the input and output parameters of the fuzzy function block.

It is possible to link a fuzzy system directly into a programmable control system if the addresses of all relevant input and output variables are already present in the programmable controller. An example of drafting a fuzzy application using existing inputs and outputs is the conversion of the manual actions of a plant operator to equivalent fuzzy rules. In this case, two to three inputs and one output will be sufficient for the fuzzy controller.

Setting Up the Rule Basis

The core of a fuzzy application is the rule basis (set of all the fuzzy rules) that represent the empirical process knowledge (expertise) of an experienced plant operator, formulated ’fuzzily’. A fuzzy rule is an If ... Then relationship (see Chapter 2) and has the form:

IF ... <condition>   THEN ... <conclusion>

where up to 56 conditions can be logically linked in the If part of a set of rules (8 inputs each with 7 membership functions).

The aim of setting up the rule basis is the recording of as many operating points (combinations of different input variables) as possible that the operator has to consider in manual operation or which the process can take in during operation. A linguistic (fuzzy) value of the respective output variable is then assigned to each of these operating points.

Off-Line Analysis of the Fuzzy Application

You can simulate and test the operational behavior of a planned fuzzy application off-line (i.e. without linking up to the programmable controller) on the programming device with the help of the configuration tool. The following functions can be used for this:

- “3D Graphic Display” with which the effect of the fuzzy rules can be represented for freely selectable combinations of inputs and outputs in spatial coordinates
- “Curve Recorder” with which the effect of test functions (slope functions that can be dimensioned according to speed of change and amplitude) on selected inputs of the fuzzy system can be monitored

This makes it possible to check the reaction of the output variables to the changes in particular input variables in steps. Any errors or deviations from the given targets can be corrected in the configuration dialog by modifying the membership functions or rules in the rule basis.

Loading and Starting Up

Once planning is completed including the off-line analysis, the fuzzy application can be downloaded to the CPU in the S7 programmable controller. Using STEP 7 Statement List connect the fuzzy function block only to the inputs of the process I/O first. If the manipulated variables behave as intended (3D graphic display or curve recorder in the configuration tool) the outputs can then be connected to the actuators.

Correcting and Optimizing

A fuzzy application that does not display the intended behavior once it is started up must be corrected or optimized using the configuration tool. You can intervene by changing the membership functions or by modifying or extending the rule basis.

6.2 Opening a Project or Calling a Project to be Processed

Generating a New Project

When you call the configuration tool the main window appears. The figure shows the procedure for creating a new project.

1. Select the menu command **File** → **New**.
2. In the dialog box "Define Project" enter the name of your project (max. 8 characters). Delete the default name to do this.
3. Define the configuration of the fuzzy application for the project by entering the number of planned inputs (max. 8) and outputs (max. 4).

   You can correct these entries later using the "Edit" menu.
4. Once you click on **OK** the system generates a project file with the name you have entered and displays the configured fuzzy structure graphically. In addition to the inputs and outputs a block for the rule basis "If ... Then" is also displayed.
5. You can access further editing windows for more detailed configuration of the fuzzy application by clicking on an input or output field or the field of the rule basis.
6. When you select **File** → **Save** your project is saved in the selected project directory. If you want to save the file in another destination directory you can click on **Save As...** and enter a path name and a drive in the dialog box "Save File As".

Opening a Project for Editing

When you call the configuration tool the main window appears.When editing a project that already exists in the archive proceed as follows:

1. Select the menu command **File** → **Open**.
2. In the dialog box "Open File" select the directory and the file name of the project to be edited.
3. When the required file name is displayed in the name box, confirm that you want to load the file by clicking on **OK**.

   The system loads the selected project file and displays the corresponding fuzzy structure with its inputs and outputs and the rule basis graphically.
4. You can access further editing windows for more detailed configuration of the fuzzy application by clicking on an input or output field or the field of the rule basis (see Section 6.4).

5. When you select **File** → **Save** your fuzzy application with any changes made is saved in your project directory.

   If you want to save the file in another destination directory you can click on **Save As...** and enter a path name and a drive in the dialog box “Save File As”. In the dialog box “Function Block Type Interface” select the preset FB30 or copy it with a new FB number. Enter the DB number in the last dialog.

6. If you want to exit or cancel editing without saving the changes, click on the menu command **File** → **Exit** and the project file will remain saved in its original unchanged form and the configuration tool exited.

6.3 Defining the Structure and Editing the Inputs and Outputs

Modifying an Existing Fuzzy Structure

Once your project **demo.fpl** with two inputs and one output has been generated (as shown in Section 6.2) you can extend it by adding inputs and/or outputs within the framework of what is possible in Fuzzy Control using the “Edit” menu.

These are always inserted or deleted at the end of the structure.

Edit Window (Overview)

Once the structure of your fuzzy project has been defined you must now define the characteristics of the inputs and outputs and the relationships between these variables. The configuration tool provides you with the edit windows ”Edit Input”, ”Edit Output”, and ”Edit Rules” for this purpose.

6.4 Defining Membership Functions

Editing an Input

When you double-click on an input in the fuzzy structure a window appears in which you can make entries to define the value range and the membership functions for the input. When you set up your project, follow the steps in the procedure below.

1. **Naming the input**

   Give the input the name you defined in your project. To do this, delete the preset name and type the current name of the input. The name can have up to 10 characters. The first character may not be a number; special characters and accents are not permitted, only ”_” is allowed.

2. **Setting the value range**

   The default for the value range is –100 to 100. If, for example, a room temperature is to be recorded via the input, the value range for this input will be 0 to 40 [°C]. By entering the values in the fields for the range Minimum and Maximum the scaling of the membership function diagram is adapted accordingly.

3. **Creating and naming membership functions**

   Using the “Insert” button you create the membership functions that describe the behavior of the input variable:

   If you overwrite the default setting **5** with **4** (in the example the behavior of the input is defined by four membership functions) and confirm with **OK**, the system generates four membership graphs with triangular forms distributed evenly over the value range. In the display field for the membership functions (on the right) the preset name **n_big** for the first of the four graphs appears.

   Note: A maximum of seven membership functions is permitted. If you enter a value greater than 7, the system still only creates seven graphs.

   Every membership function corresponds to a “linguistic value”. This represents a fuzzy or vague description of the behavior of the relevant input variable. You should therefore replace the system defaults in the rule with the current linguistic values in your project. To do this, position the cursor in the display field and delete the default. Then type in the required name.

   Note: A maximum of seven characters is permitted.

   You can access the names of the other membership functions if you click on the arrow beside the display field.

4. **Inserting and deleting membership functions**

   If you want to add another membership function to the list, simply click on the “Insert” button. An additional graph appears and **MBF_x** is appended to the end of the description list. If you click on the “Delete” button you delete the membership function whose name is visible in the display field.

5. **Changing the value range**

   If, while editing the project, it becomes necessary to change the value range for an input, there are two distinct cases:

   – The value range is to be decreased;
   in this case the system displays the message: "*There are some points outside the selected range. Adjust all membership functions?*"
   By clicking on **OK** you accept the new value range; the alternative is to cancel the operation.

   – The value range is to be increased;
   in this case the system displays the message: "*Should the membership functions be adjusted to the new range of values?*"
   You can reply to this message with **Yes** or **No**. If you select **No** all point values are retained. Owing to the increased range the display is compressed horizontally.

6. **Entering the point coordinates of the membership functions**

   If you select a description in the list of membership functions (by clicking on the arrow) the system transfers this description name to the display field and displays the respective graph in a different color or highlighted by a circle.

   The first base point of the membership function whose description is displayed in the display field is highlighted by a circle. The coordinate values of the four interpolation points of the trapezium of a membership function are shown in the "Points" list. The value of the current point is highlighted by a circle. Points 1 and 4 are always base points while points 2 and 3 are top points which coincide in the case of triangular polygons.

   A coordinate point can be selected by:

   – Selecting it in the points list

   – Clicking on an interpolation point at one corner of the selected graph (graphics field).

   The coordinate value of a selected point can be changed:

   – Numerically by changing the numeric value in the points list (5 digits)

   – Graphically by clicking on the respective interpolation point and "dragging" it along the horizontal axis

   When two points coincide each successive mouse click selects the other point alternately. When you move an interpolation point the corresponding numeric value is shown simultaneously in the points list. This enables you to set the exact numeric value you require.

   Using the button "+" a zoom function with five steps can be switched on, meaning you can move a point to an accurate a setting as you need. Switch off the zoom function with "–" or "1:1".

Editing an Output

When you double-click on an output in the fuzzy structure a window appears in which you can make entries to define the value range and the membership functions for the output.

The outputs are edited in much the same way as the inputs. To set up your project, proceed step by step as described under “Editing an Input” (on page 6-9).

1. **Naming the output**

   Give the selected output the name you defined in your project (max. 10 characters).

2. **Setting the value range**

   The default for the value range is –100 to 100. If, for example, a heating element is to be controlled as an actuator via the output, the value range for this output will be 0 to 100 [%]. By entering the start and end values in the fields for the range Minimum and Maximum the scaling of the membership function picture is adapted accordingly.

3. **Creating and naming membership functions**

   Creating membership functions is done in the same way as for an input. Using the “Insert” button you create five singleton membership functions retaining the default (see Figure KEIN MERKER). The five “line functions” are distributed equally over the value range. In the display field for the membership functions the preset name **n_large** for the first graph appears.

   Note: A maximum of seven membership functions is permitted.

   Every membership function corresponds to a “linguistic value”. This represents a fuzzy or vague description of the behavior of the relevant output variable. You should therefore replace the system defaults in the rule with the current linguistic values in your project. To do this, position the cursor in the display field and type in the required name. A maximum of seven characters is permitted.

   You can access the names of the other membership functions if you click on the arrow beside the display field.

4. **Inserting and deleting membership functions**

   If you want to add another membership function to the list, simply click on the “Insert” button. An additional line graph appears and **MBF_x** is appended to the end of the description list. If you click on the “Delete” button you delete the membership function whose name is visible in the display field.

5. **Changing the value range**

   This operation is done in the same way and with the same messages as for the inputs.

6. **Entering the singleton coordinates of the membership functions**

   If you select a description in the list of membership functions (by clicking on the arrow) the system transfers this description name to the display field and displays the respective graph in a different color or highlighted by a circle. Because singletons only have one coordinate point you do not have to enter individual coordinate values. If you set the coordinates with the mouse, the whole line graph is moved horizontally. This setting can also be made numerically in the “Point Value” field.

6.5 Formulating the Rule Basis

The AND Logic Operation

The rules for the “If ... Then” part of your fuzzy structure are formulated and entered in a separate edit window of the configuration tool in table form. The sum of all the fuzzy rules together forms the rule basis of a fuzzy application.

Each column in the table corresponds to one fuzzy rule in which the linguistic values in the IF part (top) are linked by an AND logic operation to a linguistic value in the THEN part (bottom). These may also be linked by an AND logic operation.

For example, Rule 2:
IF insidetemp. is cool AND outsidetemp. is cold AND humidity is medium
THEN heater is full

or Rule 5:
IF insidetemp. is warm AND outsidetemp. is cool   THEN heater is low

Up to 200 of these rules are possible. However, to maintain an overview of the system and not to increase the reaction time of the system too greatly, we recommend using not more than 50 rules in a fuzzy application. The number of the fuzzy rule is shown in the respective column.

The OR Logic Operation

If you want to link the inputs with an OR logic operation you can do this by formulating a number of AND rules for the same membership function of an output. In the IF part of these rules enter only one of the linguistic values to be linked and leave the others blank:

The following two rules can also be seen as an OR logic operation:
IF insidetemp. is hot and outsidetemp. is cool   THEN heater is off
IF insidetemp. is hot and outsidetemp. is warm   THEN heater is off

These two rules together:
IF insidetemp. is hot AND (outsidetemp. is cool OR warm)
THEN heater is off.

Methods of Editing the Table

The rows in the table are assigned to the defined inputs and outputs. By clicking on the “Append” or “Insert” buttons an empty rule column can be added on the right or inserted in front of the selected column.

Nine rule columns can be displayed in the window. If more than nine rules are created, a scroll bar appears below the table with which the displayed section of the table can be moved horizontally in either direction by clicking on the arrow on the left or the right or by dragging the slider in the required direction.

If you want to remove a rule, simply select it and then click on the “Delete” button. With the “Compress” function you can remove empty table columns. Note that the rules may then be renumbered.

Entering Rules

When you click on the If ... Then block in the structure diagram of the main window the rule editing window appears. This is blank at first apart from the names of the inputs and outputs listed vertically on the left.

1. The table field in the top row is shown with a frame outlining it. Position the cursor in this field. A selection menu appears containing the linguistic values and membership functions that were defined for this input.
2. Select the required membership function for the rule you want to formulate. The system inserts its description in the selected table field.
3. Position the cursor in the next field down in the column. The membership functions for the respective input or output are also displayed and selected here.
4. By selecting the menu line “-blank-” you can delete the value entered in a selected table field.

   The selection menu disappears if you click with the mouse outside it. The selected (outlined) table field remains unchanged.
5. To enter the next rule click on “Append”, or on “Insert Blank” if you want the rule to be inserted before the last one.

   In general, a rule selected in the table is highlighted by the rule number being outlined and a selected field is highlighted by the field being outlined.

6.6 Off-Line Analysis of the Fuzzy Application, Displaying a Project

Displaying the Characteristics of a Fuzzy Structure

In the “View” menu there are a number of functions for monitoring the type of action of a fuzzy application. With the option “Display 3D Graphic” it is possible to display the characteristics of the effect of two selected inputs on an output spatially.

When you select the menu command **View → Display 3D Graphic** the window “3D Graphic Display” appears with the fields for setting the input variables you want to display on the X and Y axes and the output variable on the Z axis. The selection and assignment are done in a menu that appears when you click on the arrow (to the right of the name field).

These selection menus contain all input/output variables of the current fuzzy structure so that you can select any combinations and gradually build up a picture of the effects of your fuzzy application step by step.

Determining the Animation of the Display

Being able to recognize details in spatial display depends very much on the angle of viewing. Therefore you can rotate the 3D graphic any way you like about a vertical axis or change the angle of the top view. You can either use the sliders in the scroll bars at the bottom and on the right or you can use the animation function to change the display parameters automatically.

1. Dragging the horizontal slider rotates the three-dimensional display of the influence of two input variables on the output variable displayed in the Z direction about a vertical axis. Dragging the vertical slider, for example upwards, increases the angle of viewing on the display surface of the function Z = F (X, Y).
2. When you click on the “Animation” button the dialog box “Parameters for Animation” appears. Two display types are possible:
   - Continuous rotation by the system of the angle in steps that can be set.
   - An image of the 3D display at rest with an animated display of the influence of another input variable (4D display).

Rotating the 3D Display

As the step for creating a new image is constant, the set variable of the angle 5 to 45 ∡° immediately influences the speed of rotation of the coordinate system on the screen.

If you activate “Rotation” and set the required size of the angle (using the arrow on the list box and the selection menu) and confirm with “OK”, the display starts to rotate in steps. The label on the “Animation” button changes to “**Stop**”. You can then halt the rotating movement when you want by clicking on “Stop”. The button label then changes back to “**Animation**” which allows you to continue rotation again if you click on it.

In the stop state the graphic can also be set more accurately using the sliders.

4D Display

The visualizing possibilities of the configuration tool not only permit spatial display and system-controlled rotation to change the viewing perspective, but also the “animated” display of the influence of an additional input variable on the output in question. This has the effect of a step-by-step dynamic reforming of the display field of the output.

When you activate “4D Display” you can specify an additional input variable via the selection menu “Input”. Its effect on the spatial display is calculated and displayed in set percentages.

With a percentage of 2%, for example, the value of the viewed output variable for the value range of this input covered in 50 steps is recalculated cyclically 50 times. The value of the output variable here means the number of output values displayed as a surface for the value ranges of the other two input variables displayed on the X and Y axes.

When you have set the size of the required percentage step (using the arrow on the list box and the selection menu) and confirmed with “OK”, the display surface starts to change shape in steps. The label on the “Animation” button changes to “**Stop**”. You can then freeze the movement when you want by clicking on “Stop”. The button label then changes back to “**Animation**” which allows you to continue reshaping again if you click on it.

This type of animation can also be rotated using the sliders. You can set the most favorable viewing perspective this way.

Setting the “Family Parameters”

Many fuzzy applications will have more than three input variables. If you want to include their influence on a particular output variable in the display, this is possible by specifying values of interest in its value ranges in the form of parameter settings. When you click on the “Parameter” button a dialog box with an entry list for parameter values appears.

The parameter descriptions correspond to those of the possible eight input variables of a fuzzy structure. Apart from the two ’running’ parameters of the input values shown on the X and Y axes you can assign a specific value (7 digits) to every real input here. When you confirm with “OK” the system recalculates all values of the three-dimensionally displayed output variable for this new/changed input value.

In addition to the effect of the two inputs represented directly in the coordinate system and of a third automatically animated input variable, you can monitor the influence of five other inputs of a fuzzy structure on a particular actuator. As this is only possible one point at a time for discrete values, we recommend that only so-called critical values or values at the edges of the range are set.

6.7 Monitoring Fuzzy Applications

Curve Recorder

The “Curve Recorder” function enables the display of the time curves of up to five freely selectable input or output variables in ranges between 10 seconds and 9999 seconds. At the lower edge of the diagram window is a time measurement scale whose scale results from the length of the curve display you defined. 150 values can be displayed per curve, between them the curve is interpolated linearly.

The curve is recorded in the diagram window from left to right and proceeds in steps of the resulting sampling time. When the curve reaches the end of the diagram window, the whole diagram is moved to the left in steps of approximately $^{1}/_{4}$ of the diagram length. The curves are archived. The length of the archive can be a multiple of the length of the curve display. By dragging the slider along the scroll bar or by clicking on the right/left arrows you can - when the recorder is stopped - make the stored parts of a curve visible.

Displaying the Effectiveness of Rules

In the lower diagram window of the curve recorder the rule activities are visualized in bar diagram form and defuzzification using the center-of-gravity method is illustrated.

Every fuzzy rule is assigned a bar. The number of the respective fuzzy rule is shown inside the bar. The height of the bar corresponds to the effectiveness of the rule and its horizontal position to the position of the membership functions (singletons) of the corresponding output. The limits of the horizontal display range correspond to the “Minimum” and “Maximum” of the selected output.

Reading Line

When the recorder is in the stop state, a horizontally moveable vertical line makes it possible to display numerically the values of the process variables at a particular point in time and read them off in the corresponding value fields on the right. This “reading function” has the effect of simultaneously moving the point of rotation or the arrow for the display of the center of gravity in the bar diagram, and displaying the current position of the singleton bars for the value combination of the input variables on the reading line.

Curve Assignment to Input/Output Variables

Five variables can be displayed in a common time curve each as a curve trace in a different color. Assigning a process variable to a particular curve is done via the menu **Curve Selection** in the window for the curve recorder using the following procedure:

1. Select the required curve via the corresponding option buttons in the list of numeric values (to the right of the diagram).
2. Now, after clicking on the arrow (Curve Selection) in the list of inputs and outputs of the current fuzzy application, select the variable you want to display using the selected curve. This input or output is displayed in the field “Curve Selection”.
3. If necessary, activate another option button and assign another of the variables to the curve it describes using the same procedure.

   The system displays the current momentary value in 5 digits for each curve in the display field next to the option button.
4. Selecting one of the option buttons (marking it with a black circle) means the name of the variable assigned to this curve is displayed above the diagram field and in the field “Curve Selection”. At the same time the scaling of the diagram ordinates is switched over to the value range of the selected process variable.
5. In the lower diagram an output is assigned to its bar via the menu **Bar Selection**. This assignment is done in the same way as already described for the curve selection.

Specifying the Time Frame for the Display: “Set Sampling Times”

Once you have assigned the relevant inputs and outputs to particular curves in your fuzzy application, you still have to specify the time scale for the display by setting an appropriate scale for the time axis. Depending on the inertia of the process being monitored and depending on the time resolution required for the display of events, you can select either a small or larger “Length of Curve Display”.

“Length of Curve Display” is the timespan set for one full run through the time axis of the visible diagram. This parameter can be set between the limits of 10 seconds and 9999 seconds (= 166.6 minutes).

You select the value of the time axis parameter in the dialog box “Time Values for Curve Recorder” which you access via the menu command **View → Set Sampling Times**. Small parameter values permit the recording of fast processes given a high enough resolution. Slow processes require larger values.

In total 150 values for each of the displayed curves are recorded within the selected timespan. This means the sampling times that result for the display lie between 0.1 s (with a time axis of 20 s) and 66.6 s (with a time axis of 9999 s). The current value for the sampling time in seconds is also displayed in the dialog box.

The Archive Function

The measured values recorded when the “Curve Recorder” function is activated are temporarily stored in an archive that can be dimensioned and can store up to 500 data records. This archive function makes it possible for you to analyze the behavior of the fuzzy applications retrospectively. All input data are stored in the measured value archive, even those not currently displayed.

The total length of the archive, or the total of stored values, is at maximum approximately three times the displayed length of the curve. The parameter can also be set between 10 and 9999 seconds. The lower limit is the same as the length of the curve display, the upper limit is approximately three times the curve length.

If you want to analyze the archive you have to stop the curve recorder. This interrupts data transfer from the CPU, or, in off-line mode, from the curve generator. The recording stops at the right edge at the last values to be recorded. You can make hidden areas of the archive visible by dragging the slider to the left.

During the analysis of the archive information the display data are updated on the screen. The numeric values of the input and output signals, the bars for the rule effectiveness, and the triangle for the center-of-gravity display correspond to the measured values which are present at the current position of the reading line. During the analysis of the archive it is also possible to change the displayed inputs and outputs. If a new input or output signal is selected, the diagrams are redrawn and the numeric values updated.

When you click on “Start” the archive is deleted and the system starts receiving data and displaying it again.

Working with the Function “Curve Recorder”

In addition to being able to record and monitor selected process variables during on-line operation of the fuzzy application, the “Curve Recorder” function also includes an important aid to performing off-line tests: the ability to create test signals with the help of a curve generator. These test functions with a triangular profile have the effect of a continuous slope-shaped change of one or more input variables of the fuzzy application, depending on how they are assigned.

With the help of the recordings of the curve generator the reactions of the output variables on the traversing of the whole value ranges of the respective input variables can be observed simultaneously. We recommend you only change one input variable at a time and keep the others constant. This makes it easier to keep an overview. Switching over the test function of the curve recorder to one of the other inputs can be done easily in the dialog box “Curve Parameters”.

Setting Input Values

Using the menu command **View** → **Set Curve Parameters** you can access the dialog boxes for selecting and setting the required parameters for defining your off-line test conditions. In the “Curve Parameters” dialog box the inputs of the relevant fuzzy application are shown on the left and the outputs are shown on the right. By entering numeric values at the inputs you can control the values Fuzzy Control calculates at the outputs.

Set the values by clicking on the arrow and selecting the menu point **Constant**. After you have entered the numeric value and confirmed with “OK” in the dialog box “Constant” this value is stored and displayed. The system also calculates the new values of the outputs for the current combination of inputs and displays these. This permits an initial simple off-line test.

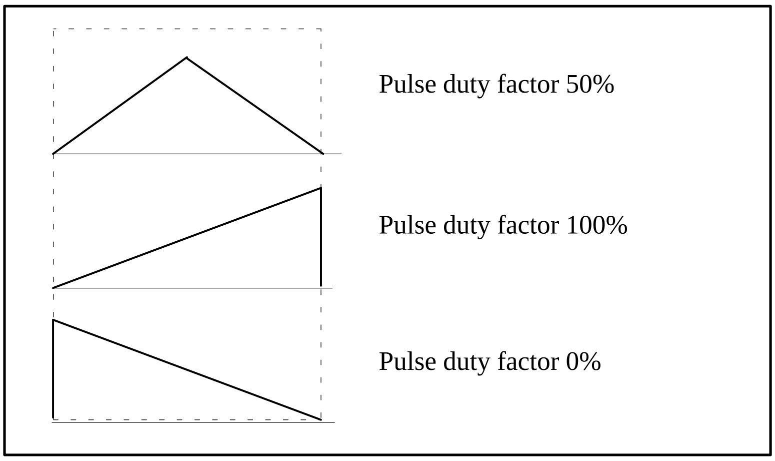
To judge the effects of input variables at their value range limits, the respective inputs can be set almost to their most extreme values in off-line test using the menu points **Maximum** and/or **Minimum**. Via the menu point **Zero** the respective input can be deactivated.

Setting Curve Parameters

If the static view of your fuzzy application with fixed input combinations is not sufficient, you can place a continuously changing slope-shaped test signal (triangular function) on one of the inputs whose time and amplitude characteristics can have parameters set. These independently defined triangular signals can also be assigned to a number of inputs.

Clicking on the menu point **Curve** the dialog box “Curve” allows you to set the characteristics of a test signal. Proceed as follows:

1. Set the amplitude of the test signal by entering the values for the “Maximum” and the “Minimum”. When these values agree with the range limits of the input variable, the slope function just touches the upper and lower horizontal limits of the viewing windows. Smaller or larger values are possible.
2. Now enter a value for the “Number of Periods”. This stands for the number of cycles of the periodic test function run through during the length of the curve display. The greater the value, the steeper the upward and downward slopes.
3. The value for the “Pulse Duty Factor” gives the relationship of the duration of the upward slope of the triangular signal to the total duration of the period as a percentage. The default is 50%.
   - Pulse duty factor = 50% means:
     The upward and downward slopes are split equally in a 1:1 relationship over the duration and their slopes are equally steep.
   - Pulse duty factor = 0% means:
     The upward slope has the gradient ∞ and the downward slope falls for the whole duration of the pulse (saw-tooth form).
   - Pulse duty factor = 100% means:
     The upward slope rises for the whole duration of the pulse, the gradient of the downward slope is ∞ (saw-tooth form).



4. The “Phase Angle” defines the starting point of the test function in relation to the curve display in the diagram. For example, a phase angle set to 25% moves the test signal by 25% of the diagram display time to the right. The default is 0%.

Entry is activated by clicking on any field in the dialog box with the mouse. Then the curve parameters are recalculated by the system and the schematic curve of the triangular signal is displayed in the graphic window of the dialog box “Curve”. When you have entered all curve parameters as required and confirmed with “OK” the values are saved and the dialog box is closed.

In the dialog box “Curve Parameters” the menu point **Curve** is now displayed instead of the numeric value for the respective input. The numeric values of the outputs are replaced by “???”. If at least one input is connected to a curve the values for the outputs can no longer be displayed as individual numeric values, therefore they are represented by the symbols “???”.

In this way all inputs can be assigned constants or curve forms in order to be able to examine the behavior of the fuzzy application off-line with the aid of the “Curve Recorder” function. It is also possible, as already mentioned, to assign curves to several inputs simultaneously. In most practical applications it is generally more useful to connect only one input to a test signal and assign constants to the remaining inputs. This makes it easier for you to analyze the behavior of the outputs in relation to this input.

Generally the curve recorder allows you to examine the input and output behavior of the fuzzy application you developed. You must not forget that you are still monitoring the fuzzy system independently from the process. When you connect it to the process on-line you may need to perform further tests in a closed function circuit.

6.8 Downloading and Starting Up the Fuzzy Application

Downloading the Fuzzy Application

Download the instance data block with the program for your fuzzy application together with the call OBs and the function block FUZZY into the CPU of your programmable controller. Use the SIMATIC manager application to do this.

Processing Call for Fuzzy Control

The function block FUZZY must be called by the user. The call can be programmed in the cyclic and/or in the time-driven program processing level. The block must be called absolutely. Processing is controlled via the variable START_STOP in the instance data block.

When the function block is called an instance data block must be specified. The set of rules for the fuzzy application is stored in this data block. A separate data block is required for each set of fuzzy rules. Only the input and output parameters required in the set of rules need be assigned. Any unused inputs or outputs do not need to be connected.

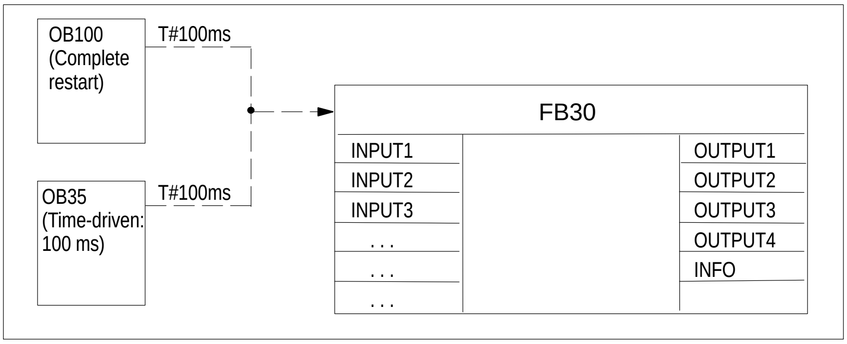


Figure 6-1 Connecting the Start-Up Blocks to the Fuzzy Function Block

Example: Minimal call only with the parameter INFO

```
STL                                         Explanation
CALL FB30, DB30 (
          INFO := MB30;
```

External Supply to Inputs/Outputs

If a data block has been created as an instance data block and if a symbolic name has been assigned, external access to the fuzzy application in the user program is also possible. Example:

```
STL                                         Explanation
..
T      ”Pendulum”.INPUT1                    = Symbol of the DB
..                                          Variable in the DB
L      ”Pendulum”.OUTPUT2
etc.
```

Glossary

Actual Value

The current value of the controlled variable

Actuator

A device used to influence the manipulated variable at the process input. Mostly consists of the combination of a positioner and a final controlling element.

Analog Value Input/Output

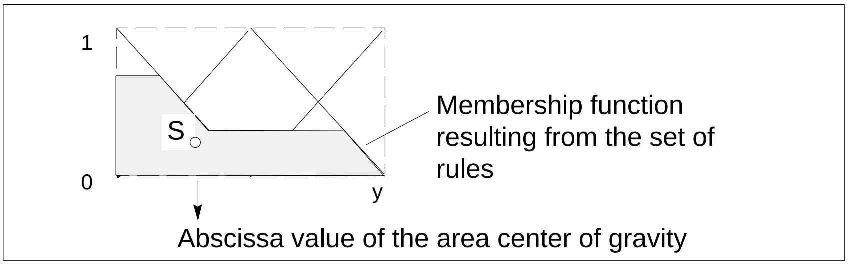
An algorithm (function) for converting an input value in I/O data format to a real number value and normalization to a percentage, and corresponding to converting an internal percentage value to the I/O format of the output variable.

Block Diagram

Represents the cause-effect relationships in a programmable control system or a process. The block diagram consists of function blocks which represent the transfer behavior of the real parts of the process and flow lines which represent the signal flow direction.

Center-of-Gravity Method

The output value of a set of rules is calculated as the abscissa value of the center of gravity S of the area lying below the output membership function.



Membership function resulting from the set of rules

Abscissa value of the area center of gravity

Complete Restart

When a complete restart is performed on a controller it can be started from a defined starting state. The output parameters and local, or static, data of the controller are preset in the complete restart routine. If the complete restart bit is set at the input COM_RST, the complete restart routine is run.

Configuration Tool

A software tool for creating and assigning parameters to a standard control system and for optimizing a controller with the help of data obtained by system analysis.

Controlled Variable

A process variable (output variable of the process) that is to be adapted to the momentary value of the reference variable. Its momentary value is known as the actual value.

Curve Recorder

Creates constant numeric values or time-dependent time functions at selected inputs of the fuzzy application for off-line test purposes.

Database

→ Instance Data Block

Defuzzification

The conversion of the fuzzy set expression of an output to a concrete numeric value, represented as a membership function. The influences of the rules on this output are also weighted and the area center of gravity of the resulting membership function is formed (→ Center-of-Gravity Method).

Disturbance

All variables which influence a controlled variable, with the exception of the manipulated variable, are described as disturbances or disturbance variables.

Fuzzification

The transfer of the momentary values of an input to degrees of truth of linguistic statements (→ Linguistic Value).

Fuzzy Application

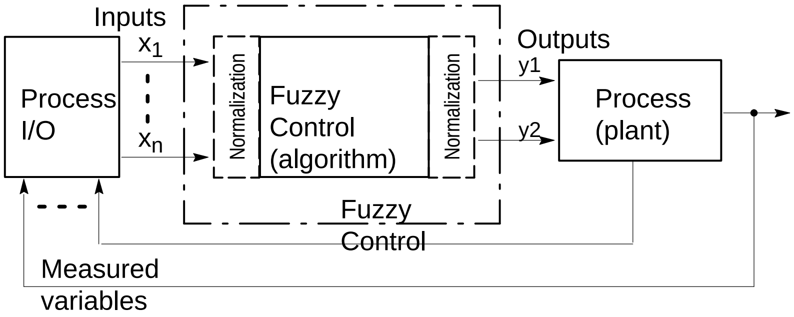
Configuration consisting of a defined number of inputs and outputs and the complete set of rules that determines the influence of the input variables on the output variables according to application and subdomain (→ Set of Rules). Each fuzzy application is contained in its own specific instance data block.

Fuzzy Application Restart

An “automatic restart” is started when power returns following power-down, “manual restart” is triggered by a switch setting or a communication command (→ Complete Restart, → Restart).

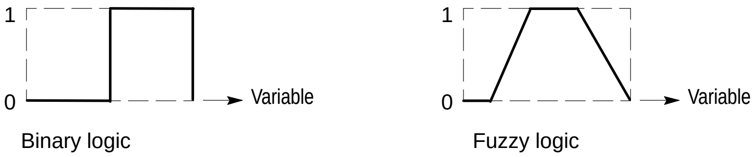
Fuzzy Control

An algorithm (function block) of the programmable controller (PLC) that records new values for the input variables in constant or program-dependent time intervals and then calculates new values for the output variables according to the empirical knowledge laid down in the set of rules.



Fuzzy Logic

Logic with fuzzy set membership of the elements (→ Truth Value).



Fuzzy Programming Language (FPL)

Fuzzy applications are saved in the data format of the Fuzzy Programming Language or FPL. The FPL standard was specified by Togai InfraLogic Inc.

Fuzzy Rule

Linguistic rule with the form IF A ... THEN B, also known as the production rule, describes a part of a strategy determining the behavior of a fuzzy application. The IF part (condition) can also be linked with a logic operation, the THEN part (conclusion) is a simple assignment of a linguistic value to an output variable.

Fuzzy Set

A fuzzy set is fully described by its → Membership Function. For each element in the basic set the membership function states the degree of membership to the relevant fuzzy set in the form of a real number between 0.0 and 1.0, also known as the → Truth Value.

Inference

Processing the set of rules by linking the degrees of truth of the inputs with the help of a minimum operator (IF part) and calculating the truth values for the respective output (THEN part) by overlaying the cut-off membership functions of several rules.

Instance Data Block

The data block in which the structure (inputs/outputs) and the behavior (set of rules) of a configured fuzzy application are stored.

Linear Scaling

A function for converting the values of an input variable into percentage values of the output variable before they are processed in internal comparison and control algorithms.

Algorithm: OUTV = INV * FACTOR + OFFSET

Linguistic Value

If you define a temperature range, for example, as being subdivided into the fuzzy sets of cold, warm, and hot temperature values, we describe ’cold’, ’warm’, and ’hot’ as *linguistic values* of the variable ’temperature’. Linguistic values are a means of addressing areas of a physical variable which are not clearly definable (fuzzy sets) qualitatively and not using concrete numeric values.

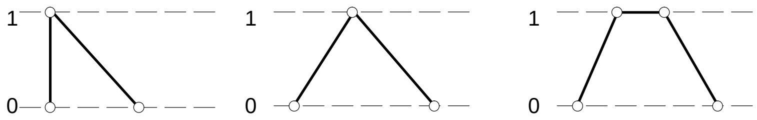
Manipulated Variable

Output variable of the programmable control system or input variable of the process. An actuating signal can create an image of the manipulating range as an analog percentage value or as a pulse width value or pulse duration. With an integrated actuator (e.g. a motor) it is sufficient to supply binary up/down and forwards/backwards switching signals.

Membership Function

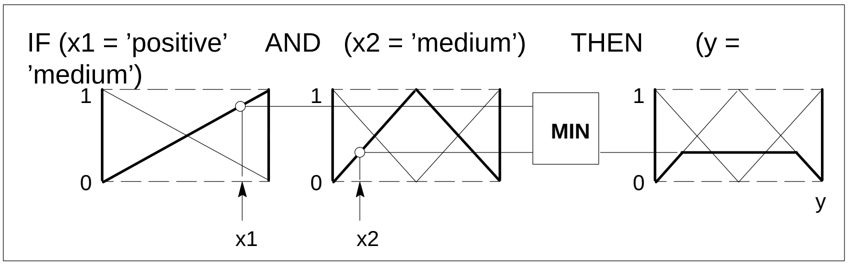
The membership function quantifies the qualitative expression of a → Linguistic Value in the manner that it gives the degree of membership (→ Truth Value) to the particular subset which embodies the respective linguistic value for each numeric value of the process variable.

Each membership function is defined by four interpolation points and can therefore take the form of a triangle or trapezium.



Minimum Operator

The selection of the ’minimum’ of the momentary truth values in the IF part of a rule creates an AND logic operation linking the fuzzy expressions.



Normalization

A procedure (algorithm) for converting the physical values of an input variable to the (internally processed) value range (X_{min} to X_{max}) and for conversion back again at the output. The normalization line is determined by the initial value (OFFSET) and the slope (FACTOR). Normalization is performed in blocks which have to be called separately (functions or FCs).

Numerical Notation

The values of analog I/O variables are realized in STEP 7 as fixed-point numbers (format: WORD) in the range –27648 to 27648.

Operating Point

Describes the value of the manipulated variable at which the deviation of the controlled variable from the setpoint value becomes zero. This value is significant in controllers without an I component in which a remaining system deviation is required in the steady state to uphold the required manipulated variable value. If this deviation disappears, the operating point parameter must be adjusted accordingly.

Process Simulation

A copy of a functional circuit with preset transfer fields to simulate a real process. Once they have been excited by changes in disturbance variables or changes to reference variables, the relevant process variables can be archived or made accessible for monitoring in the form of a curve.

Programmable Controller (PLC)

Programmable logic controller consisting of one or more CPUs, I/O devices with digital/analog inputs and/or outputs, devices for linking up to and communicating with other devices in the system, and a power supply if required.

Reference Variable

Specifies the required value or course of the relevant process variable. Its momentary value is known as the → Setpoint Value (SP).

Restart

When a fuzzy application is subject to a restart, it retains the operating mode and with the recorded results that prevailed at the point of interruption. This means the fuzzy system continues working with the values it had calculated at the time it was interrupted.

Scaling

Function for adapting (emulating) analog variables on the basis of setting the zero point and of the slope of a linear transfer function (→ Normalization).

Set of Rules

The sum of all IF/THEN rules of a fuzzy application (knowledge basis). Each of these linguistic rules describes a substrategy of the fuzzy application.

Setpoint Value

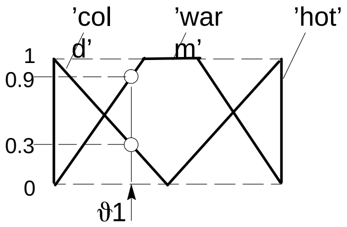
A value which the controlled variable should assume when subjected to the effects of an automation system.

Singleton Function

Membership functions of the outputs in the form of lines whose ordinate value gives the degree of membership to the respective linguistic value of the output variable. The influences on a particular linguistic value of the output variable resulting from various rules are added together.

Truth Value

The percentage to which a particular value of an input variable can be assigned to a linguistic value as “true”. In the example, the temperature $\vartheta 1$ is assigned to the linguistic value ’cold’ with a degree of truth of 30% and to the linguistic value ’warm’ with the degree of truth of 90%.



Value Range

Numeric range of a process variable (for example, –20°C to +80°C) whose value is processed internally as a real number in the range –100.0 to +100.0%). Physical values in the number range for STEP 7 can be entered at the inputs (→ Numerical Notation).

Index

Siemens AG
AUT E 146

Östliche Rheinbrückenstr. 50
D–76181 Karlsruhe
Federal Republic of Germany

From:
Your Name: _
Your Title: _
Company Name: _
Street: _
City, Zip Code_ _
Country: _
Phone: _

Please check any industry that applies to you:

❒ Automotive
❒ Chemical
❒ Electrical Machinery
❒ Food
❒ Instrument and Control
❒ Nonelectrical Machinery
❒ Petrochemical
❒ Pharmaceutical
❒ Plastic
❒ Pulp and Paper
❒ Textiles
❒ Transportation
❒ Other _ _ _ _ _ _ _ _ _ _ _

Remarks Form

Your comments and recommendations will help us to improve the quality and usefulness of our publications. Please take the first available opportunity to fill out this questionnaire and return it to Siemens.

Please give each of the following questions your own personal mark within the range from 1 (very good) to 5 (poor).

1. Do the contents meet your requirements? □
2. Is the information you need easy to find? □
3. Is the text easy to understand? □
4. Does the level of technical detail meet your requirements? □
5. Please rate the quality of the graphics/tables: □
6. □
7. □
8. □

Additional comments:

– –

– –

– –

– –

– –

– –

– –

– –

– –

– –

– –

– –