Sixth-Generation Computer Technology Series

Branko Souček, Editor
University of Zagreb

Neural and Massively Parallel Computers: The Sixth Generation
Branko Souček and Marina Souček

Neural and Concurrent Real-Time Systems: The Sixth Generation
Branko Souček

Neural and Intelligent Systems Integration: Fifth and Sixth Generation Integrated Reasoning Information Systems
Branko Souček and the IRIS Group

Fast Learning and Invariant Object Recognition: The Sixth-Generation Breakthrough
Branko Souček and the IRIS Group

Dynamic, Genetic, and Chaotic Programming: The Sixth Generation
Branko Souček and the IRIS Group

Fuzzy, Holographic, and Parallel Intelligence: The Sixth-Generation Breakthrough
Branko Souček and the IRIS Group

Fuzzy, Holographic, and Parallel Intelligence

The Sixth-Generation Breakthrough

BRANKO SOUČEK
and
The IRIS GROUP

A Wiley-Interscience Publication
JOHN WILEY & SONS, INC.
New York-Chichester-Brisbane-Toronto-Singapore

E. Prugovecki, *Quantum Mechanics in Hilbert Space*, Academic Press, New York, 1981.
J. Sutherland, *Holographic Model of Memory, Learning and Expression*, Internat. J. Neur. Syst., 1 (3), 256–267, (1990).
J. Sutherland, *A Transputer Based Implementation of Holographic Neural Technology*, *Transputing '91*, 2, 657–675, (1991).
L. Yarolavski, *Methods of Digital Holography*, Plenum Press, New York, 1980.

CHAPTER 3

Fuzzy Reasoning in Planning, Decision Making, and Control: Intelligent Robots, Vision, Natural Language

DARKO STIPANIČEV
JANET EFSTATHIOU

3.1 INTRODUCTION

A quarter of century ago Lotfi Zadeh introduced a mathematical theory known as fuzzy set theory suitable for dealing with problems and situations that are in their nature ambiguous and not precisely defined. On the other side, in the past couple of decades robotics has been established and developed as a well defined and widely practically applied field of mechanical and electrical science. At first sight it seems that there are no connecting points between terms as ambiguous, imprecise, vague, or fuzzy and the mathematically well defined field of robotics. This chapter will try to emphasize the possible connections between them.

Let us start with a few notations concerning robustness and its importance, particularly in the field of robotics. In this context robustness means the ability to respond without program modification to slightly perturbed or to somewhat inexactly specified situations [1]. The expression "slightly perturbed situation" needs a more detailed explanation. For example, in an assembly line equipped with industrial manipulators it would be highly desirable for each step to perform its function in spite of inevitable inaccuracies of the positions of objects coming down the line. Standard programs for assembly line robots, like, "rotate 14.3 degrees clockwise, raise 10.3 cm," certainly do not possess this property. But an example of desirable robust robot behavior would be the situation in which an industrial manipulator carries loads whose weights alter from sample to sample. Even more complex situations, where it is possible to find more ambiguity and fuzziness, are connected with robot action in an unknown environment. A typical example would be the activity of a mobile, unmanned robot in unknown space. Such a robot is usually a sensor-guided robot, the machine that exhibits a connection of perception and action, and perception is usually connected with a considerable degree of inprecision.

A lot of vagueness, ambiguity, and fuzziness is present in all of these examples, so that all methods capable of dealing with and solving problems in such situations can find a lot of applications. Of course, the fuzzy set theory is not an exception. This chapter will be a kind of survey, the review of applications of fuzzy set theory in different subfields of robotics.

Four main areas of application of fuzzy set theory in robotics could be distinguished:

1. Control of robot dynamics
2. Interpretation of information received from a robot's sensors
3. Robot motion planning and control
4. Modes of communication with robots using natural language

Before proceeding with particular applications, a short introduction to fuzzy set theory will be presented. Fuzzy set theory is nothing but a generalization of classical set theory. In classical set theory, an object (element) from the universal set may or may not be a member of a particular set of object classes, because the characteristic function of an element in a set in classical set theory is binary: either 0 or 1. This law of excluded middle (*tertius non datur*) limits the applicability of classical set theory. In many practical applications the membership of an object in a class is not binary: Classes of fat men, pretty persons, or elongate objects are some classes whose membership cannot be represented satisfactorily by only 0 and 1.

Zadeh observed that allowing the characteristic function of elements of a set to take a value in the interval [0, 1] will dramatically extend the applicability of the set theory. He introduced concept of fuzzy set (A) as a subject of universal set (Ω), such that the characteristic function $f_A(x)$, $x \in \Omega$ takes any value in the interval [0, 1].

The union, intersection, and complementation of fuzzy sets A and B are defined as fuzzy sets, too, and their membership functions are calculated by formulas:

$$f_{A \cup B}(x) = \max\ (f_A(x), f_B(x)) \tag{3.1}$$

$$f_{A \cap B}(x) = \min\ (f_A(x), f_B(x)) \tag{3.2}$$

$$f_{\bar{A}}(x) = 1 - f_A(x) \tag{3.3}$$

Since its introduction, fuzzy set theory has been studied and applied in diverse fields. In this report its applications in robotics are considered. Let us start with its first application area, the control of robot dynamics.

3.2 CONTROL OF ROBOT DYNAMICS

Fuzzy dynamic controllers have been used for the management of various systems, from large scale industrial processes, such as cement kilns, water purification and treatment, heat exchangers, and sinter plants, to smaller scale laboratory and experimental systems, such as the famous Mamdani's steam engine, marine auto pilots, diesel engines, processes of pH-neutralization, and water level control [2]. Recently a whole range of consumer goods run by fuzzy controllers, such as washing machines, vacuum cleaners, air conditioners, TV sets, video cameras, dishwashers, and microwave ovens, have been introduced on the market, announced as intelligent household appliances [3]. The results of these applications have shown that fuzzy controllers perform either better or at least as well as optimally tuned PID controllers. The main advantages of the fuzzy approach to the control are as follows:

It does not require a detailed mathematical model of the controlled process to formulate the control algorithm

It has more robust and adaptive capability

It is capable to operate for a large range of inputs

It is cheaper

The properties of fuzzy controllers, robustness and adaptivity, are particularly interesting in their application in the field of robotics. Let us use as an example a robot system with revolute and prismatic joints [4].

Such robots are widely used for numerous tasks in industry because they are fast acting and they approach the flexibility of use normally ascribed to the human arm. Industry has successfully mastered the techniques of manufacturing such robot arms, but it is now at the stage where their dynamic performances are often called into question.

The problem is that the moment of inertia of the arm, links about their main control axes (rotation, shoulder and arm) can exhibit pronounced changes with the change in disposition of the robot, as well as the load carried at its tip. This results in highly complex dynamics [5].

Model-reference adaptive control (MRAC) is the usual approach to the control of such systems. The disadvantage of MRAC is that it generally requires a detailed knowledge of the required system dynamics and assumes that

1. These are linear
2. The dynamics of the actual system differs from the model only with respect to values of its coefficients

Contrary to this approach, the fuzzy control approach is based either on only appropriate knowledge of the system behavior, which need not be linear (in the case of a nonlearning rule based fuzzy controller), or on very simple incremental system model (in the case of a learning, self-organizing, rule based fuzzy controller). The joint feature of both fuzzy control approaches to the control of robot dynamics is the existence of a knowledge base or, more precisely, the existence of a rule base where rules about control procedures are stored. Because of that, a fuzzy controller is a special case of *production system*, where fuzzy set theory has been used for representation of knowledge (control rules) and for doing inferences with the knowledge. Figure 3.1 shows situation schematically.

The main idea of fuzzy controller is quite simple: Knowledge about the control procedure is stored in a rule base as a collection of "if . . . then" rules. A typical example of such rules is

If error is *small positive* and change-in-error is *small positive* or *zero*, then control most be *small negative*

Error, change-in-error, and control normally take values in real universes. Fuzzy set theory connects these real universes (subsets of real numbers) and linguistic terms such as small positive, zero, and small negative.

For each robot movement a controller considers at each sampling instant the error between required and actual angles, as well as the change-in-error. The con-

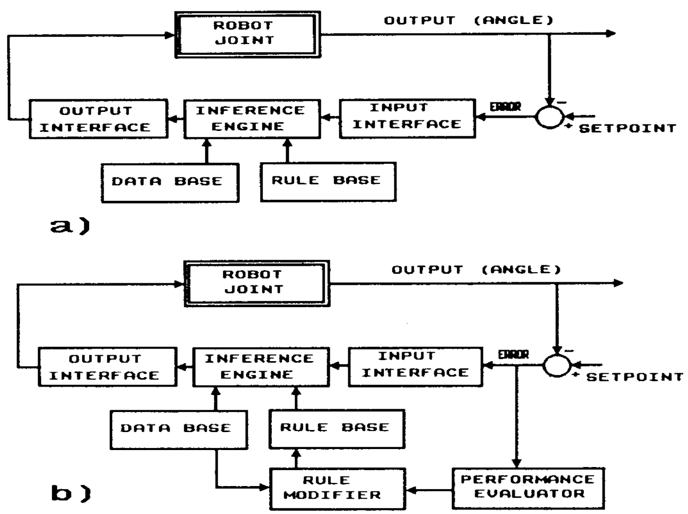

Figure 3.1 *Block diagrams of rule based fuzzy controllers:*
a. Simple nonlearning rule based fuzzy controller.
b. Self-organizing rule based fuzzy controller.

troller feeds these two quantities into a rule base through the inference engine and decides, by use of fuzzy logic, which of several rules may be used to provide a contribution to a controller output and what this output (control action) should be. This control action is then applied through output interface. The main difference between a nonlearning and a learning fuzzy controller is that for the first one all control rules must be known and defined at the beginning of the control session, whereas for the other one, a self-learning fuzzy controller is capable of creating its own rule base during on-line operation. Also, a self-learning fuzzy controller has the property to adapt itself and tune its rule base if the system is changed. To do this self-organizing, a fuzzy controller needs quite simple performance evaluators, which is another collection of "if . . . then . . ." rules created using a simple incremental model of the controlled system. A typical example of such rules is

If error is x and change-in-error is y, then control should be reinforced by z

where values x, y, and z could be also expressed linguistically.

It is interesting that such a simple idea applied in the field of robotics in a number of cases gave better results than mathematically well defined and *precisely* described controllers.

Both nonlearning and learning fuzzy controllers have been applied to control the robot dynamics. The first example that will be presented here is the fuzzy control of one link manipulator servo system [6]. The nonlearning fuzzy controller was applied and the control procedure was defined by six general proposed rules:

1. If e is *LP* and $\dot{e}$ is any, then u is *LP*
2. If e is *SP* and $\dot{e}$ is *SP* or *ZE*, then u is *SP*
3. If e is *ZE* and $\dot{e}$ is *SP*, then u is *ZE*
4. If e is *ZE* and $\dot{e}$ is *SN*, then u is *SN*
5. If e is *SN* and $\dot{e}$ is *SN*, then u is *SN*
6. If e is *LN* and $\dot{e}$ is *any* then u is *LN*

where e is error (output from shaft encoder − set point), $\dot{e}$ is change in error ($\dot{e} = e(nT) - e[(n-1)T]$), u is output from servo drive unit and *LP*, *SP*, *ZE*, *SN*, and *LN* mean large positive, small positive, zero, small negative, and large negative. Two quantizations of variables were used: one for coarse control when the output is distant from the set point and one for fine control when the output is close to the set point. For example, for an incremental type of shaft encoder having a resolution of 1000 for coarse control, the real universe of error from −1000 to 1000 was quantized into eleven levels (−5, −4, −3, −2, −1, 0, 1, 2, 3, 4, 5). The breaking point between coarse and fine control was an error value less than 100, and then error values from −100 to 100 were quantized into nine levels (−4, −3, −2, −1, 0, 1, 3, 4). The same was done for change-in-error and control. Figure 3.2 shows a block diagram of the system layout.

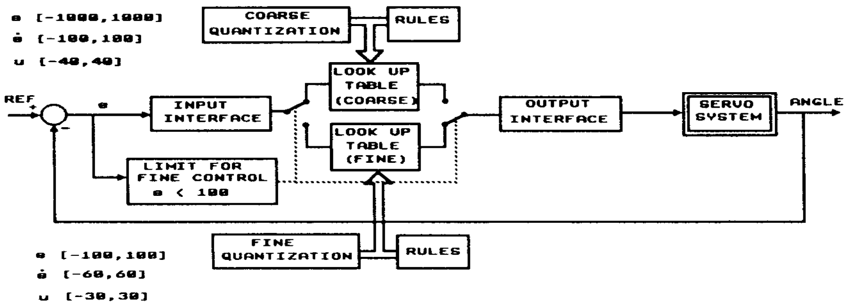

Figure 3.2 A block diagram of a servo rule based fuzzy controller according to [6].

Linguistic terms *LP*, *SP*, *ZE*, *SN*, and *LN* were defined by appropriate fuzzy sets. Figure 3.3 shows fuzzy set membership functions for fine quantization.

Control procedure was the classical fuzzy control graphically represented in Figure 3.4. The real error and change-in-error were quantized and applied to each rule. The control contribution of each rule was calculated using a min method, and overall control was evaluated by a max procedure. The real-time quantized control was calculated by a center of gravity method. For values marked on Figure 3.4 ($e_0 = \dot{e}_0 = 3$), $u_0 = (0.6 \cdot 2 + 1.0 \cdot 3 + 0.6 \cdot 4)/(0.6 + 1.0 + 0.6) = 3$. The next procedure was inverse mapping from quantized u_0 to real control value from interval $[u_{min}, u_{max}]$.

For comparison, two other control algorithms were implemented, the conventional digital PI controller and the model-reference adaptive controller (MRAC), and then applied for servomotor control. The fuzzy controller gave better results than the PI controller. In was comparable to MRAC from the point of view of its sensitivity to disturbance (changing of mechanical time constant); yet its dynamic response was better (less overshoot and shorter settling time).

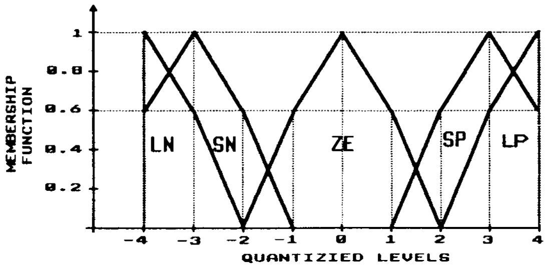

Figure 3.3 Graphic representation of fuzzy sets membership functions.

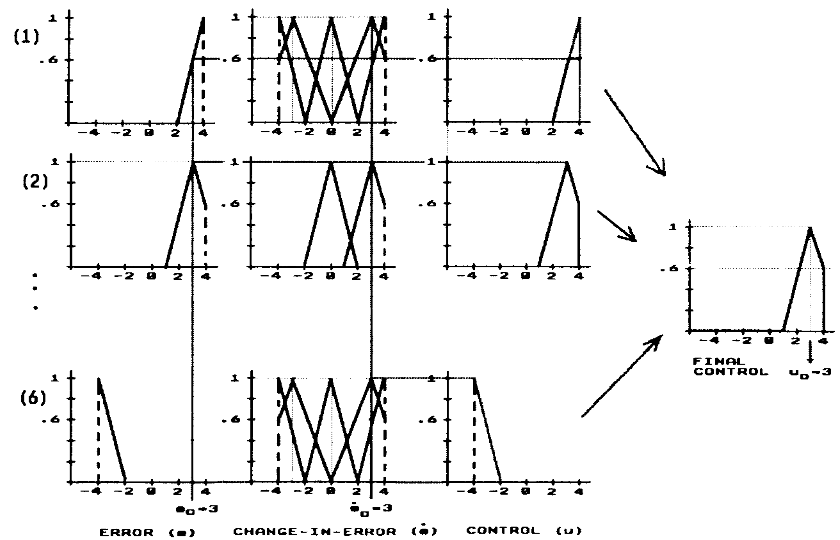

Figure 3.4 Graphic representation of fuzzy rule based control procedures.

Another group of researchers [7] made a modification on a fuzzy controller layout and put the fuzzy controller in parallel with a feedforward gain passage, so mixed fuzzy control was obtained. Figure 3.5 shows the system layout.

The fuzzy controller was responsible for only dynamic regulation. Static per-

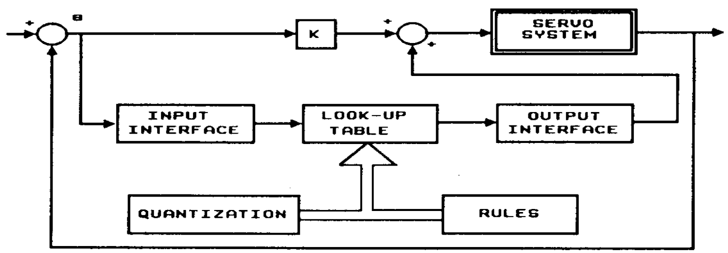

Figure 3.5 A mixed fuzzy controller according to [7].

formances were satisfied by position feedback. Because of that, the problems of oscillation and static error that could occur in conventional fuzzy controllers were overcome. Simulation results have shown better dynamic response performances and static behavior than that of optimally tuned PID and common fuzzy controllers.

Another specific use of fuzzy set theory in control of robot dynamics is the tuning of conventional robot servo systems [8]. The position and velocity gains K_p and K_v are tuned by fuzzy interface autotuning system using a rule based, common fuzzy controller with two sets of rules that correspond to human tuning procedures.

In the following, the application of a fuzzy controller in controlling robot dynamics differs from previous approaches in the sense that a learning, self-organizing type of fuzzy controller was used and real life experiments were performed [4, 9].

The principles of a self-organizing fuzzy controller (SOC) applied in this case could be summarized as follows: In the main control loop there is a common fuzzy controller. For a specific error value (any change-in-error value) one of few control rules is chosen, and subsequently the specific control action is evaluated. The learning part monitors the controller's performances with reference to an incremental performance criterion and decides if it is necessary to reinforce some of the existing rules or even create a new rule in the rule base. The performance table is derived from linguistic rules of the form:

If error E is x, and change-of-error ΔE is y, then process output U should be really different by p units

where quantity p could be a positive or negative value. Now a simple incremental model is assumed, consisting of a gain (which is usually unity for linear processes) and a process time-delay of m samples.

In each sampling instant, the reinforcement p is taken from the performance table, and the control rule (or rules) responsible for control action m sampling instants in the past are changed from $(\Delta E_m, E_m, U_m)$ in $(\Delta E_m, E_m, U_m + P)$. The robot used in these experiments with SOC is shown schematically in Figure 3.6. The robot has the 1m reach and a lifting capacity of 1.5 kg. An SOC algorithm was applied in a Pascal-like process control language of a DEC computer.

Each robot joint (rotation, shoulder, arm) was controlled with a separate self-organizing fuzzy controller. Nonlinear gains for input and output variables were used. The system has shown better performances than an optimally tuned PID controller, particularly in the tracking procedure. The maximum deviation from the nominal squared trajectory was twice as big for the PID control than for the SOC. The final conclusions were that a self-organizing fuzzy controller

a. Could cope with the transient aspect of the robot response
b. Could deal with changing system properties (unlike the PID controller)

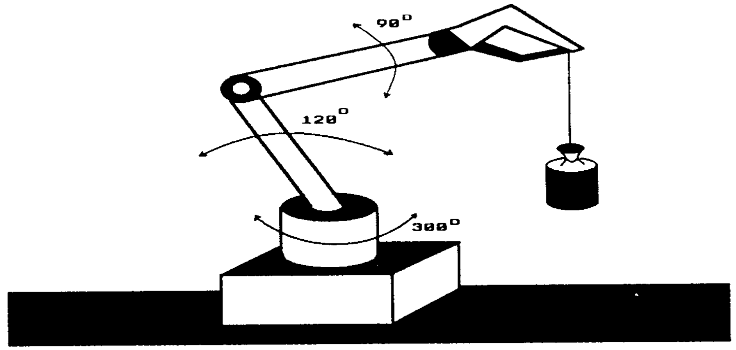

Figure 3.6 A schematic drawing of a revolute joint robot arm used in experiments with SOC, according to [4, 9].

c. Could learn fast so that the controller could adapt itself to changing process characteristics, such as robot loading
d. Could work in cooperation with another separate SOC in the face of cross-coupling effects

We also did a lot of simulation trials of robot dynamic control with different forms of self-organizing controllers and found that, in the synthesis of self-organizing controllers, one of the most difficult parts was the tuning of controller parameters [10]. SOC has many parameters that have to be defined and tuned in order to obtain satisfactory system response and convergence of self-organization of control rules. The most important parameters of SOC are

- Input and output scaling factors
- Delay in rule modification m
- Performance index (PI) table

but these parameters

- Definition of fuzzy values in control rules
- Definition of inference procedure
- Definition of defuzzyfication method

also have an influence on closed loop behavior.

A lot of simulations and experimental trials have been made to compare different inference and defuzzification methods and to find the influence of fuzzy values definition on controller performances, but a general procedure has not been found. In order to overcome the problem with the tuning of scaling factors, we have proposed a self-tuning procedure [10]. Our idea was to try to summarize the influence of different tuning parameters on a system response in the form of tuning rules, and then to use this approximate knowledge in the tuning procedure.

Three main difficulties were met during the simulations of a self-tuning, self-organizing fuzzy procedure for the control of robot dynamics with a nonlinear model of a PUMA 560 manipulator.

The first problem was due to the unstable characteristic of the system because the final position was in an unstable space sector. The use of high values for the gains allowed the process stabilization but the convergence was very hard. With low values of gains, several learning experiences were needed to obtain stability. Afterward, convergence was possible. This problem could have been solved by including an "a priori" knowledge in the control rule base to avoid instability, instead of starting the learning phase without rules, as we did. That starting rule base could approximate the behavior of a conventional controller. Equations $\Delta U \cong E + \Delta E$ or $\Delta U \cong E + \Delta E + \Delta E^2$ could be used in the case of the three term controller.

The second difficulty was that several steps in the same direction of change (increase or decrease) were needed to observe the effects of these changes in gains. In fact, the relationships linking the ratios and the gain have several local minima due to the nonlinearities of the controller and the process.

The third problem was that it was necessary to start with scaling factors, which at least would have led to a stable closed loop and to the convergence of the control strategy. Our self-tuning strategy was good for the improvement of response characteristics but not for finding the starting combination of scaling factors.

Although our study was primarily related to the project aiming at the use of fuzzy models of human behavior and their transpose to robotics applications, we believe that such control principles could be quite useful also for the real control of robot dynamics. Experimental results previously reported have demonstrated the advantages of self-organizing fuzzy control for robot dynamics control. We believe that in the future more applications of fuzzy controllers and particularly self-organizing fuzzy controllers to control robot dynamics will be encountered.

Another group of fuzzy controller applications to control robot dynamics is in the field of *sensor-guided robot manipulators*. The sensor-guided robot requires, according to the nonlinear characteristics of a certain interaction process between the robot effector and the object, a robust control algorithm for the sensory control loop. Here we will suppose first that the values obtained from the robot sensors are precisely known and that they could be unambiguously interpreted. Fuzzy set theory is used only in the control loop. Palm [11] has classified the total task in sensor-guided robots in two parts: the motion task and the reaction task, schematically shown in Figure 3.7.

The motion task is controlled on the servo level, whereas the relation task has to be controlled on the sensor level. The values obtained from sensors are, together with values described from the reaction task, considered as inputs to a controller.

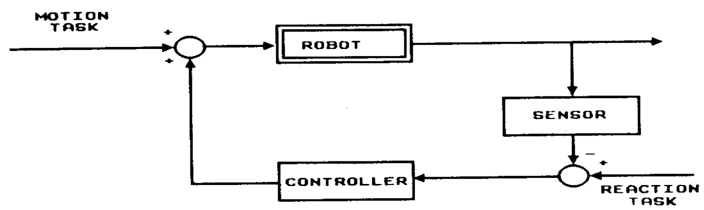

Figure 3.7 *A block diagram of sensor-guided robot control.*

The controller produces a control law. That control law is responsible for the correction of the planned trajectory obtained in the pure execution of the motion task. The controller in the external feedback loop requires a quite high robustness because of nonlinear effects, like the loss of contact between the effector and the object, or dry friction. Fuzzy control is an example of such a control procedure.

Palm has used the ordinary, common fuzzy controller. Control rules were derived classifying the phase plane of error and change-in-error into six ambiguous regions, as shown in Figure 3.8. Following the principle of suboptimal control, the line $e + \dot{e} = 0$ serves as the switching line of the control variable sign.

The linguistic interpretation of fuzzy sets for error, change-in-error, and control were positive big (*PB*), positive small (*PS*), negative big (*NB*), and negative small (*NS*). Each fuzzy set was linearly quantized into 22 levels. Shapes of membership functions are shown in Figure 3.9.

The general control procedure, obtained heuristically, could be summarized as follows:

1. If region F then $u = NB$
2. If region A or D then $u = NS$
3. If region B or C then $u = PS$
4. If region E then $u = PB$

where regions $A, B, \ldots, f$ are defined on Figure 3.8.

The inference mechanism was quite simple, a conventional max–min procedure and the center of gravity method for defuzzification. Experiments were performed with a tactile sensor, with the controller programmed in Prolog and implemented on a PDP-11-like machine. The sensor, mounted at the robot's hand, measured the reactions between the robot effector and the object. The task was to keep a constant distance from the object during the tracking procedure.

The advantages of the fuzzy controller, in contrast to a conventional PD-controller, were better tracking capabilities and less error, and the main disadvantage was a relatively fast alternation of control output.

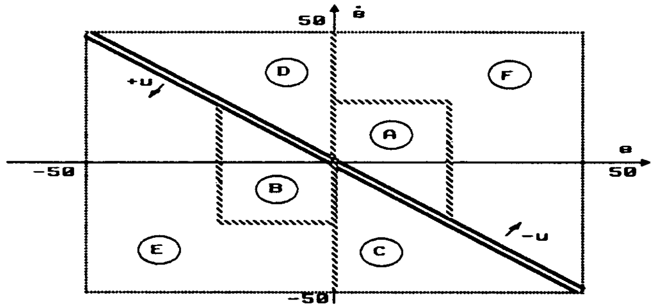

Figure 3.8 A phase plain of sensor-guided fuzzy robot control, according to [11].

In this section we have shown how fuzzy set theory was used and how it could be used for the control of robot dynamics. Although all of the described applications were laboratory experiments, we suppose that the newly introduced fuzzy hardware (fuzzy integrated controllers and fuzzy microprocessors) will give additional spur to the practical application of fuzzy control principles. The conventional control of robot dynamics will be no exception, but from our point of view, fuzzy control principles will find a better field of application in sensor-guided robots. This kind of robot control has been partly discussed in the present section, but only for the case in which the sensory information is precisely known and when it is not necessary to analyze or interpret it. The following section deals with the cases when additional analysis of sensory data are required, showing how fuzzy set theory could be employed for such tasks.

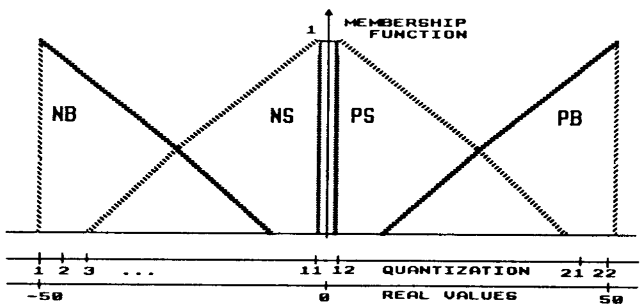

Figure 3.9 A definition of fuzzy sets in sensor-guided fuzzy robot control.

3.3 ANALYSIS AND INTERPRETATION OF ROBOT SENSOR INFORMATION

According to R. Palm [11], the new generation of sensor-guided robots (robots of the third generation) must have "intelligent connection of perception and action." An academic discussion could be opened in connection with the meaning of the word "intelligent."

A unique approach does not exist and lot of definitions of intelligence have been given, but it is possible the exact answer could never be made because the intelligence itself is a fuzzy feature. Here we are dealing with robots, and in this context an intelligent robot is treated as a machine capable of solving problems. The problem is generally defined as a situation for which a person or machine does not have a ready answer, a ready response. Problem solving involves

The sensing and identification of the problem
The formulation of the problem
The utilization of relevant information
The generation and evaluation of solutions (hypotheses) [12]

In robotics, problems are always connected with some kind of *motion*: motion of a robot hand to reach an object, motion of a robot grasper to take an object, or motion of a robot itself if it is a mobile robot. The functional decomposition of intelligent robot tasks could be done according to Figure 3.10.

Fuzzy set theory has already found a lot of applications in almost all parts of this functional decomposition. New trends in sensor technology and particularly the development of fuzzy sensors and fuzzy transducers [13, 14] will probably find a lot of applications in robotics, but let us here discuss techniques already applied.

Gupta et al. [15] have emphasized that in systems of natural perception, where natural sensors are used for perception of images, temperature, sounds, and fragrance, the process of "feeling" or "recognizing objects" entirely depends upon the perception of certain attributes, rather than the measurement of their physical characteristics in absolute terms. In natural systems "perception" plays the same role as "measurement" in artificial systems. In the development of an intelligent system it is proposed, therefore, to make use of "perceptions" by employing some sort of *perceptor* rather than taking absolute quantitative measurements by means of precise and conventional measuring devices. Fuzzy reasoning could easily find application in such systems.

As a typical example let us use the perception of distance for an adaptive arc welding industrial robot [16]. An industrial robot with nine servo controlled axes is equipped with a noncontact distance sensor, which scans seam motion in the plane perpendicular to the trajectory to be welded. The sensor consists of an infrared pulse laser mounted on scanning platform together with a CCD (charge coupled device) line camera. The laser sensor measures distance between the CCD

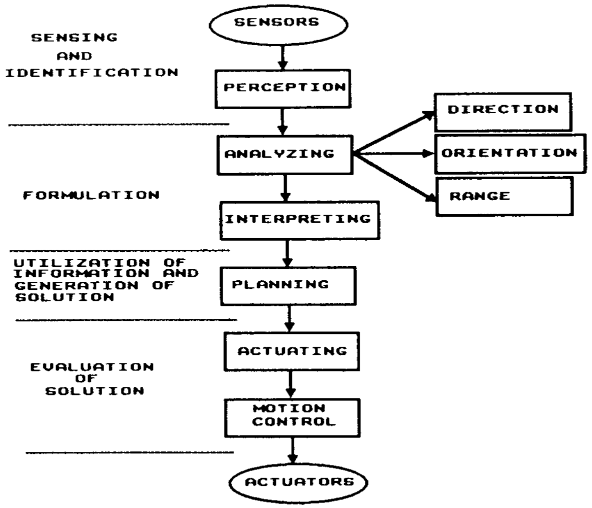

Figure 3.10 Functional decomposition of intelligent robot tasks.

receiver and the surface to be welded, as Figure 3.11 shows. For every scanning motion a snap of the cross section is registered.

If such distance sensor is to be used, it is necessary to find

1. How to estimate measured distances
2. How to estimate actual trajectory deviation from the pretaught one on the bases of registrations in 1

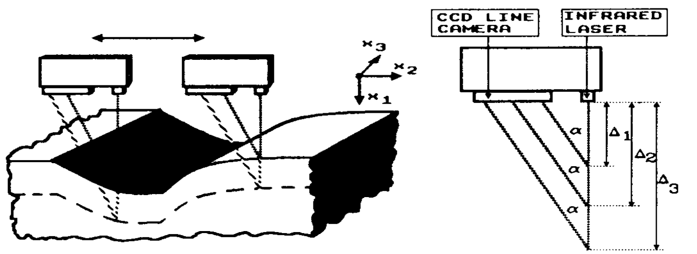

Figure 3.11 Laser distance sensor.

Every laser emission is registered via its reflective image on the line CCD camera. By a definite constant angle between the falling and reflecting laser beam (α), the distance Δ from receiver to measured surface is given as a serial number of the CCD camera lines. Unfortunately the image is rather a spot than a point, and consequently the registered value is not a single value but a set or values. This unsharp measuring calls for a fuzzy estimate.

The idea of ith level fuzzy set was used; ith level fuzzy set was defined as

$$F^{(i)}(X_i) = \{\mu^{(i)} | \mu^{(i)}: X_i \to L_i\} \tag{3.4}$$

where L_i is an argument of the previous level. For example, for a vertical distance X_1, $L_1 = [0, 1]$, for a cross-deviation X_2 of welding seam $L_2 = X_1$, and for a longitudinal deviation X_3 of the welding seam, $L_3 = X_2$ were used. As an information carrier of fuzzy set its power was taken. Based on the relative power of the longitudinal deviation for fixed prediction interval length, the proximity was determined and expressed in linguistic terms: very close, medium, far, and very far, and for every proximity estimate, the corresponding fuzzy action was assigned to eliminate deviation. Figure 3.12 shows fuzzy sets for one step of this procedure.

Another quite important problem in modern sensor-guided robotics is a problem of robot vision, as a special case of the more general problem of computer vision. Computer vision has a wide range of potential applications, so that the past two decades have seen a growing interest in building systems able not only to "see" using a camera as an eye, but also to understand the image. In the next section, a fuzzy vision transducer will be introduced, designed for tasks of robot motion planning and control. Here the process of image understanding or scene analysis

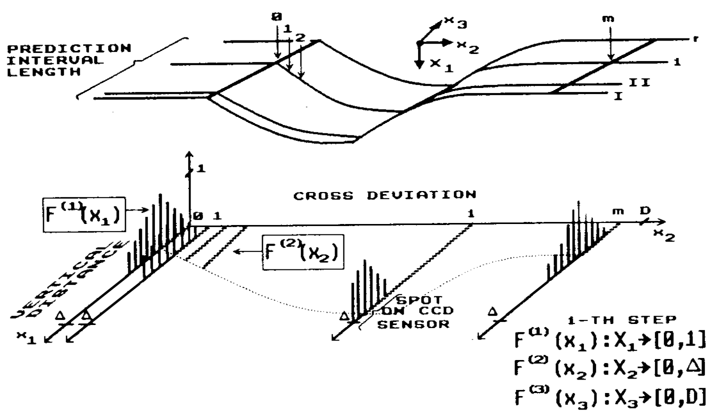

Figure 3.12 One step of the proximity estimation procedure.

based on fuzzy set theory will be described. This process involves the extraction of the description of the scene in terms of the objects presenting the image and the spatial relationships among them. The goal is to obtain the three-dimensional information about the scene from its two-dimensional projection, the image.

There are lots of difficulties caused by complexity and imprecision, which makes the process of scene analysis quite difficult [17]. A scene analysis procedure usually has two distinct phases:

1. segmentation or feature selection
2. interpretation or classification

In segmentation, the image array is partitioned into disjoint regions, each region satisfying a predicate based on some property of intensity value. During the interpretation, the domain knowledge is applied to extract some specific information from the segmentated image.

For a variety of inspections and manipulation tasks in robotics, the recognition of objects by means of computer vision is quite important. A typical example is the safety inspection in nuclear power plants by robot vehicles. Such a system [18] has to

- Verify the position of values and dampers
- Measure oil and liquid levels in sight meters
- Read instruments and gauges
- Detect and locate steam/water leaks
- Verify the integrity of security locks, and so forth

As an example of fuzzy set theory application in this domain, the edge perception will be described.

Perception of edges is an example of segmentation processes suitable for object recognition. This could be done by transforming the gray-level image from the absolute intensity domain to the perception domain using fuzzy set theory, since it deals with a set of phenomena that may ambiguously belong to a set. That is also a feature or natural human perception. Figure 3.13 gives the functional block diagram of such a system [15].

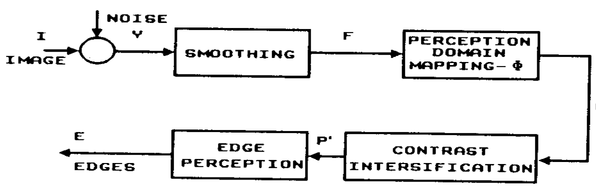

Figure 3.13 *A functional block diagram of an edge perception system.*

I is the original, gray-level digital image in a spatial-intensity domain, which could be mathematically represented by a matrix $\mathbf{I} = [i_{m,n}]$. The image Y, corrupted by noise, still in a spatial-intensity domain, is a matrix $\mathbf{Y} = [y_{m,n}]$. If we take the histogram of this noisy image, the distinct regions of the various intensity levels may not be readily identifiable. In order to reduce the effect of noise, a simple averaging scheme is taken over a window W of size $q \times q$. Thus, the smoothed pixel in the image can be defined as an average pixel intensity over the window W. The smoothed image is $\mathbf{F} = [f_{m,n}]$, where $f_{m,n}$ is the intensity value of the pixel (m, n). The next procedure is domain mapping from a spatial-intensity domain to a spatial-perception domain using the mapping function ϕ. The transformed image is $\mathbf{p} = [p_{m,n}]$, where $p_{m,n} \in [0, 1]$. The mapping function decimates the entire intensity region into a number of distinct regions assigning alternatively the low $\in [0, 0.5)$ and high $\in [0.5, 1]$ perception values. Thus, mapping from $f_{m,n}$ to $p_{m,n}$ may be considered as an aggregate phenomenon of perceiving the intensity levels of the gray digital image. This aggregation or granularity is one of the important attributes found in human perception, thinking, and decision making. The calculation procedure is quite simple. First the histogram for intensity levels from an image is determined. Generally an image is composed of k intensity levels $L_1, L_2, \ldots, L_k$ in intensity range $[x_m, x_M]$ as Figure 3.14 shows.

These intensity levels are distinctly identified at their peaks for values $x_1, \ldots x_k$, and maximum uncertainty is at valleys $x_{c1}, \ldots, x_{ck}$. The reason is that, for example, the intensity value x_{c1} may belong to either intensity level L_1 or intensity level L_2, so the uncertainty is maximal. The multiregion mapping $\phi(x_1)$ is defined so that the whole range of intensity values $[x_n, x_M]$ is mapped in two distinct intensity levels: low and high. Crossover points from low to high and high to low are intensity values with maximum uncertainty $x_{c1}, x_{c2}, \ldots, x_{ck}$. Now to each

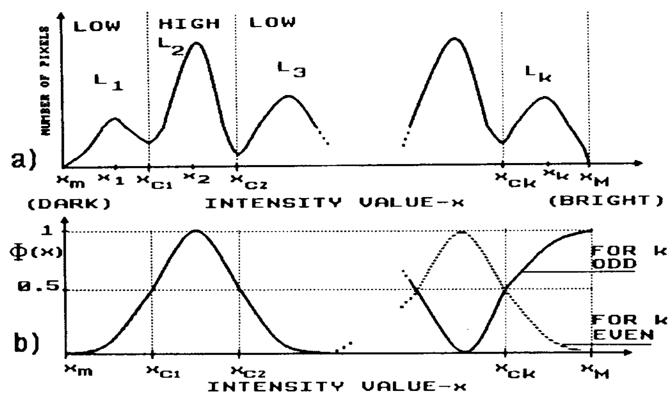

Figure 3.14 *A multiregion mapping function, according to [15].*

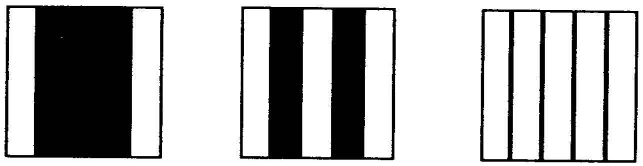

Figure 3.15 *A simple example of edge perception with three distinct intensity levels.*

pixel from smoothed image $f_{m,n}$ which has some intensity value x, a number $\phi_{m,n} = \phi(f_{m,n})$ from interval [0, 1] is assigned. The transformed image $\mathbf{P} = [\phi_{m,n}]$ is nothing but a fuzzy matrix. Value 1 corresponds to completely bright and 0 to completely dark. The next step would be the *contrast intensification*, an operation well known in fuzzy set theory. Its task is to reduce the entropy and to make some kind of focusing and reduction of the fuzziness. As a result of this operation bright becomes very bright and dark becomes very dark.

The resulting image $\mathbf{P}' = [p'_{m,n}]$ is composed of regions with two distinct brightnesses. A simple example of this is shown in Figure 3.15.

The next step would be edge perception. Most of the pixels in the image $\mathbf{P}'$ are either in the low or high perception range, but at the edge of this region are pixels with perception levels near to 0.5. In order to detect these boundaries, a max–min operation is performed over a window W of size $q \times q$ using the operation known as EDG[·]. An example of this operation is

$$E = [e_{m,n}] = \text{EDG}[\mathbf{P}'] = |\{p'_{m,n} - \min_{(i,j)\in W} p'_{i,j}\}| \qquad (3.5)$$

where for $(i, j) \in W$, $(i, j) \neq (m, n)$.

Finally the edge locus could be defined as all pixels (m, n) for which $e_{m,n}$ belongs to a certain interval $[\alpha_1, \alpha_2]$, where $\alpha_i \in [0.1]$ are the values that bound the degree of perception. For example, primary edges could be defined as $e_{m,n}$ from interval (0.75, 1], secondary edges for $e_{m,n} \in [0.5, 0.75]$ and no edge for $e_{m,n} \in [0, 0.25]$. The size of the window W determines the width of the edges in perception domain.

When the edges are determined and located (position is determined with pixel position (m, n)), the recognition process could be started using specific knowledge. An example will be the knowledge about spectral, spatial, and relational properties of objects. In [18] a frame based system was used for the presentation of such knowledge. The knowledge for slider recognition could be

\# 1 NAME: SLIDER No. of subobjects = 2
  Name of subobject: HANDLE
  Name of subobject: SLOT

  Intrarelational property: Slot is right-of-handle
  Spatial property: undefined
  Spectral property: undefined
\# 2 NAME: HANDLE No. of subobjects = 0
  Intrarelational property: no
  Spatial property: min relative size: 20%
    max relative size: 60%
    shape: rectangular
  Spectral property: The object is darker than background

Although in [18] the nonfuzzy techniques were used for object recognition, the fuzzy set theory could make the system even more versatile by expressing object properties with linguistic values modeled by fuzzy sets. For example, the linguistic expression of the size could be small, quite small, very big, and so forth, and to each of them an appropriate fuzzy set could be assigned according to Figure 3.16.

The other applications of interpretation of information obtained from robot sensors, also based on fuzzy set theory, are as follows:

- filtering and editing sensor data in space flights for space shuttle rendezvous problem [19]
- moving mark recognition [20]
- information fusion in computer vision [21], and so forth

Some examples from the next section deal with these tasks, as well. In addition, a good review with extensive bibliography covering publications concerning fuzzy set theory application in more general problems of pattern recognition could be found in [22].

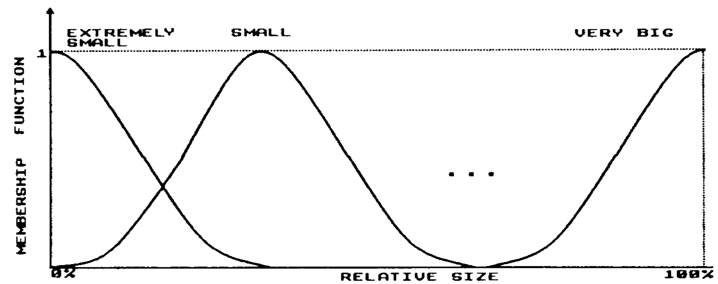

Figure 3.16 *Definition of fuzzy sets for object size descriptions.*

3.4 ROBOT MOTION PLANNING AND CONTROL

After the object has been perceived and recognized, the motion of the robot has to be planned and executed, that is, the motion of a robot hand toward the object to be reached or picked or the motion of the whole mobile robot avoiding obstacles. In both cases, the planning procedure is usually based on the construction of an internal robot model of the real world. The problem is that the complexity of the environment in which the robot has to act cannot be fully represented in the model. A lot of uncertainty and fuzziness is present in such robotic systems either because of inadequacy of robot receptors and effectors, or because of impossibility to represent objects, locate objects, or perform actions on objects with sufficient accuracy. Because of that it is not possible to construct a precise functional mapping between the state space of the model and the state space of the external world. Fuzzy set theory gives the methodology to handle such kinds of problem. Our approach based on fuzzy relations [23] will be discussed here.

The starting point was the work of Averkin and Dulin [24]. They used fuzzy mapping as a bridge between the state space of the internal model and the state space of the external world, so the first fuzzy mapping and fuzzy relations will be introduced shortly.

The binary fuzzy relation R^* is the relation that may hold between the elements of any two crisp sets X and Y to any degree between 0 and 1. Formally, it is a set of ordered pairs $R^* = ((x, y), R(x, y))$, where (x, y) is an element of the Cartesian product $X \times Y$ while $R(x, y)$ is its characteristic function. Transition from a binary fuzzy relation to an n-ary fuzzy relation is straight.

If X and Y are discrete sets, $X = \{x_i \mid i \in I\}$ and $Y = \{y_j \mid j \in J\}$, where I and J are index sets, then R^* is a discrete fuzzy relation and it can be completely given by its fuzzy matrix $\mathbf{R}^*$ with components

$$r_{ij} = R(x_i, y_j), \qquad i \in I, \quad j \in J \tag{3.6}$$

This fuzzy matrix may be concrete if X and Y are finite sets; otherwise it would be only conceptual.

Let us suppose that we have a fuzzy relation R^* from X to Y and fuzzy relation P^* from Y to Z. The composition o of fuzzy relations R^* and P^* is also a fuzzy relation S^*, but from X to Z whose membership function, for each pair (x, z) could be obtained by equation

$$S(x, z) = \sup_{y \in Y} \min (R(x, y), P(y, z)) \tag{3.7}$$

$S(x, z)$ could be seen as the strength of a set of chains linking x and z. The strength of such a chain is that of the weakest link, and because of that the operation min is used. But between x and z there are more chains through different y, consequently the strength of the relation between x and z is that of the strongest one (operation supremum over all y from Y).

The composition of finite fuzzy relations can be viewed as a matrix product. With $\mathbf{R}^* = [r_{ij}]$, $\mathbf{P}^* = [p_{jk}]$, $\mathbf{S}^* = [s_{ik}]$, and $\mathbf{S}^* = \mathbf{R}^* \circ \mathbf{P}^*$ we have

$$s_{ik} = \sum_j r_{ij} \otimes p_{jk} \tag{3.8}$$

where Σ is in fact operation max and $\otimes$ operation min. The composition (3.3) or (3.4) is usually called sup-min composition.

Let A^* and B^* be fuzzy sets defined on X and Y, respectively. A^* implies B^* ($A^* \rightarrow B^*$) or, expressed in words, "If A^* then B^*" is a fuzzy conditional proposition. A mathematical operation for translating this proposition into a fuzzy relation R^* in $X \times Y$ is called the fuzzy implication operator. There are many possible definitions of this operator, but in control applications Mamdani min definition is most frequent:

$$R(x, y) = \min (A(x), B(y)), \qquad x \in X, \quad y \in Y \tag{3.9}$$

Fuzzy relational models are the appropriate way to represent the uncertainty of the external world. They can be used in cases when it is not possible to construct a precise functional mapping between the state space of the internal model and the state space of the external world. The values of membership functions of fuzzy modeling relations may be seen as degrees of similarity between the world and the model, or as degrees of precision of the real world description. Fuzzy modeling relation R^* is a binary fuzzy relation between the world state space W and the model state space M. For example, R^* could be seen as a fuzzy matrix whose columns correspond to the robot's world state space, let us say, to the discrete values of the passageway width through which the mobile robot must pass. The rows of fuzzy matrix R^* correspond to the robot's internal model state space as, for example, elements symbolically expressed in words: "very wide" (VW), "wide" (W), "narrow" (N), and so forth. Table 3.1 is a typical simple example. The elements in this table express degrees to which the elements of the world state space W belong, to the elements of the model state space M, and vice versa.

Each row of Table 3.1 defines a membership function of fuzzy set m_i^* from the model state space whose support set is the real world state space W. The situation is more-or-less identical for each column in Table 3.1, which defines a member-

TABLE 3.1 Fuzzy Modeling Relation RI^* for Passageway Width (values of W are in meters)

PASSAGEWAY WIDTH		WORLD (WI)			
		0.5	1	1.5	2
MODEL (MI)	VERY WIDTH (VW)	0	0.1	0.25	1
	WIDE (W)	0	0.3	0.5	1
	NARROW (N)	1	0.7	0	0

ship function of the fuzzy set w_j^* from the world state space whose support set is the state space of the model *M*. A typical example is the fuzzy set "wide" whose membership function is fuzzy vector [0 0.3 0.5 1]. This fuzzy vector says that, for example, the real value 1.5 belongs to the concept expressed with "wide" degree 0.5. Also, the real value 1.5 m could be seen as fuzzy set "1.5 m" defined on the model state space with fuzzy vector [0.25 0.5 0], which means that, for example, the symbolic value "very wide" belongs to the fuzzy set "1.5 m" with degree the 0.25. The elements of the real world are not fuzzy in the real world state space, but they become fuzzy in the model world state space, and vice versa.

It is important that using this approach we can construct the internal model representation of the real world with various levels of abstraction. The level of abstraction is directly connected with the cardinality of the model state space *M*. At the lowest level of abstraction, the state space of the model is the same as the state space of the world ($M = W$). A case apart represents the nonfuzzy, functional model when modeling relation is Boolean and one-to-one. The real abstraction begins when state spaces *M* and *W* are no more sets with the same elements. The special case with Boolean but not one-to-one mapping is the quantization of real line. For example, the element symbolically expressed with "0.5" or *I* or *#*1 may stay for all real values between 0. and 0.5.

If we introduce more elements in the model state space *M*, for example, the terms "very very wide" or "not so wide," the level of abstraction diminishes, but if we reduce the cardinality of the set *M* the level of abstraction increases. The model state space with only two elements, "wide" and "narrow," is more abstract than the one given in Table 3.1.

It is important to notice that, in the case when model state space *M* has more elements than world state space *W*, we have a situation contrary to abstraction; we have some kind of interpolation.

Let us now suppose that we have a fuzzy modeling relation for both inputs and outputs of the robot control system. Table 3.1 could serve as an example of relational models of input information "passageway width" and Table 3.2 for output information "robot velocity."

The task of the control system is to plan the robot velocity according to the specific input information about passageway width. Let us suppose that the velocity could be adjusted only in discrete steps from Table 3.2 and that information about passageway width is also discrete shown in Table 3.1.

TABLE 3.2 Fuzzy Modeling Relation *RO for Robot Velocity (values of *W* are in meters per second)**

	ROBOT VELOCITY	WORLD (WO)				
		0.2	0.5	0.8	1	1.5
MODEL (MO)	HIGH (H)	0	0	0.2	0.8	1
	MEDIUM (M)	0.2	0.7	1	0.8	0.1
	LOW (L)	1	0.9	0.2	0	0

The final result of the planning procedure, which is the input information to low level controller, must be one and only one element coming from the robot velocity world state space. This means that finally each controller that acts in the real world must be a deterministic controller. But on the model level it is not necessary to have a deterministic procedure. Moreover, a nondeterministic, fuzzy procedure is closer to the description of the control procedure that humans use during control. Let us suppose that we are using human knowledge for passageway width—a robot velocity planning task. The planning procedure could be expressed with production rules like these:

If passageway width is very wide, than velocity could be high

or

If passageway width is wide, then velocity could be high, too

or

If passageway width is narrow, then velocity could be low

Conventional procedure is to express these rules with fuzzy condition propositions connected with union (**VW*** → **H*** ∪ **W*** → **H*** ∪ **N*** → **L***) where **VW***, **H***, **W***, **N*** and **L*** are fuzzy vectors obtained as rows of Table 3.1 and 3.2. To transform these rules in fuzzy relation between the real world state space of passageway width and real world state space of robot velocity, a fuzzy implications operator must be used. For example, if we define a fuzzy implication with Eq. (3.10) and union with max, the result is Table 3.3. The final step is interpretation, and that means changing Table 3.3. into a Boolean table, where in each row one and only one nonzero element exactly equal to 1.

Using the theory of fuzzy relations, let us now show how this procedure could be interpreted. The production rules are relations between the model state space of passageway width and the model state space of the robot velocity. For the production rules this relation is a Boolean one, given in Table 3.4.

As elements of fuzzy modeling, relations of control algorithms could be seen as degrees of strength between input and output world state space; the natural way of obtaining this relation is by composing fuzzy relations *RI**, *RP**, and *RO**.

TABLE 3.3 Fuzzy Relation of Control Algorithm

CONTROL ALGORITHM		WO				
		0.2	0.5	0.8	1	1.5
WI	0.5	1	0.9	0.2	0	0
	1	0.7	0.7	0.2	0.3	0.3
	1.5	0	0	0.2	0.5	0.5
	2	0	0	0.2	0.8	1

TABLE 3.4 Production Rules as Relation RP^*

PLANNING RULES		MO		
		HIGH	MEDIUM	LOW
MI	VERY WIDTH (VW)	1	0	0
	WIDE (W)	1	0	0
	NARROW (N)	0	0	1

Boolean relation P^* from Table 3.4 is also only a special case of fuzzy relations:

$$RC^* = RI^* \circ RP^* \circ RO^* \tag{3.10}$$

It's interesting that the results obtained by Eq. (3.10), where composition $\circ$ is given by Eq. (3.7), or more preciously by Eq. (3.8), completely coincide with Table 3.3, which is obtained by fuzzy implication operator, Eq. (3.9), and the definition of union with max. This equality could be proved.

This approach, which uses the composition of relations instead of fuzzy implication, has one additional degree of freedom. That degree of freedom is that, in production rules, instead of a Boolean relation there is a real fuzzy relation. Examples would be as follows:

If passageway width is very wide, then velocity could be high with degree 1, medium with degree 0.8, or low with degree 0.2

or

The velocity could be high if passageway width is very wide with degree 1, wide with degree 0.9, or narrow with degree 0.1

The existing fuzzy control algorithms are usually based on a fuzzy implication, which leads to Boolean relations. The proposed fuzzy approach is a novelty and it is suitable for the cases when it is not possible to obtain consensus about control actions.

Practical applications of these ideas concerning fuzzy modeling relations could be motion planning and control of a mobile robot. An example would be an unmanned submersible, which must act in relatively unknown underwater surroundings.

The typical decomposition of a robot's tasks are planning, navigation, and piloting [25]. For each of these tasks, fuzzy modeling relations with different levels of abstraction could be used. At the highest but least precise and detailed level is the *planner*, which operates on incomplete information. The level of abstraction is here the highest one. Real world is described just roughly, with an approximate model. The next level is *navigator*, which utilizes a more detailed map to evaluate an obstacle-free local path satisfying some performance criteria. With the path description the pilot provides motion control avoiding obstacles. It requires the most precise information and the least possible abstract model state space. This operation of decreasing the level of abstraction of internal model state space could be called *zooming of abstraction*. It could be defined as a procedure that increases the cardinality of the model state space. Complementary to this procedure could be *zooming of fuzziness* when values of fuzzy relation membership are changed so that fuzziness diminishes (values are clear to 1 and 0). The third procedure could be *zooming of precision*, defined as a procedure that increases the cardinality of the world state space. For example, the world state space of the passageway width is more precise then the one shown in Table 3.1 if it has more then five elements for widths between 0 and 2 m.

These three procedures could be used together to obtain a more-or-less precise picture of the external, real world for different tasks of mobile robot motion planning and control.

Let us explain how fuzzy set theory could be used in mobile robot motion, planing, and control *on the pilot level*, when the robot is equipped with the vision sensor. The image represents a limited range of space in front of the mobile robot. For free obstacle motion control, it is necessary to evaluate the boundaries dividing the floor and the objects. For this task a simple differentiation of camera signal could be used [25]. Distances and positions of objects and walls are defined according to their x, y position in the image, taking into account the camera height and the tilt angle. Figure 3.17 shows a simplified image together with definition of most important quantities. W is the width and L is the distance between the robot and the obstacles.

The main planning objective on the pilot level is to find a permissible route without obstacles. The widest permissible path is determined by W and P, where $W = \max_i (W_i)$, and P is the coordinate that represents the direction of the robot motion. For the case shown in Figure 3.17 the widest path is W_2. If the robot is too close to the object or the robot diameter is bigger than W_2, the pilot will conclude that there is no passageway in front of the robot. Consequently the robot will start turning around to find a passage in a new direction.

When there is an acceptable passageway, a minimum distance between the ro-

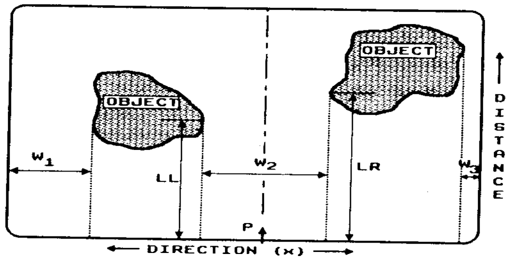

Figure 3.17 *The simplified image perceived by the pilot camera.*

bot and the obstacles is calculated, taking into account the robot width WR. For this calculation equation $L = \min(LL, LR)$ is used. After that a simple fuzzy algorithm could be applied for direction and velocity control. In [25] direction control was obtained by changing the speed of the left and the right wheel according to these equations:

$$U_L(k) = U_L(k-1) + G\,\Delta U + aT \tag{3.11}$$

$$U_R(k) = U_R(k-1) + G\,\Delta U + aT \tag{3.12}$$

U_L and U_R are left and right reference wheel speed commands, G is the direction control gain, ΔU is the direction control rate, a is the acceleration control input, and T is the sampling period. Conventional feedback loop was used to stabilize real speeds V_L and V_R to their reference values U_L and U_R. The control variables determined by fuzzy controller were g, ΔU, and a. Three types of control rules were defined:

$$R_1\text{: } (\Delta V = \Delta V_i \text{ and } P = P_i) \rightarrow \Delta U = \Delta U_i \tag{3.13}$$

$$R_2\text{: } (\text{sgn}(\Delta V)) = S_i \text{ and } L = L_i) \rightarrow G = G_i \tag{3.14}$$

$$R_3\text{: } (V = V_i \text{ and } W = W_i) \rightarrow U = U_i \tag{3.15}$$

where ΔU and G were defined previously; ΔV is the speed difference between the left and right wheels at the sampling instant $k-1$; p is the permissible passageway direction; sgn(ΔV) is the sign of ΔV; L is the distance from the robot to the nearest obstacle; V is the average speed command of the left and right wheels at sampling instant $k-1$; W is the permissible passageway width; and U is the reference speed. ΔV_i, P_i, ΔU_i, S_i, L_i, G_i, V_i, W_i and U_i are linguistic values of variables mathematically defined by fuzzy sets; 49 rules of type R_1, eight rules of type R_2, and 16 rules of type R_3 were defined. Real worlds of ΔV, P, and ΔU were quantized into 13 levels $(-6, -5, \cdots, 6)$ and their corresponding model world into seven levels (negative big (NB), negative medium (NM), negative small (NS), zero (ZO), and the same for positive). The real world of G was quantizied into seven levels $(1, 2, \cdots, 7)$ and its model world into four levels (zero (GZ), small (GS), medium (GM), bit (GB)), and the same was done for the model worlds of LL, LR, W, and U (Z, S, M, B), but their real worlds were quantized into nine levels $(0, 1, 2, \cdots, 8)$. Rules are given in Table 3.5.

Max-min composition was used, the implication was defined by min operator, and the center of gravity method was applied for the interpretation of the resulting fuzzy sets. G and ΔU from Eqs. (3.11) and (3.12) were directly derived from the rules. Acceleration control a was calculated by using this simple equation:

$$a = (U - V)/(m \cdot T) \tag{3.16}$$

where constant $m \in [1, 5]$ was chosen on the basis of the surrounding situations and acceleration characteristics of the robot.

TABLE 3.5 Fuzzy Control Rules for Mobile Robot Steering and Speed Control According to [25]

		ΔV: NB	NM	NS	ZO	PS	PM	PB
P	NB	ZO	NS	NS	NM	NM	NB	NB
	NM	PS	ZO	NS	NS	NM	NM	NB
	NS	PS	PS	ZO	NS	NS	NM	NM
	ZO	PM	PS	PS	ZO	NS	NS	NM
	PS	PM	PM	PS	PS	ZO	NS	NS
	PM	PB	PM	PM	PS	PS	ZO	NS
	PB	PB	PB	PM	PM	PS	PS	ZO
		ΔU						

		LL				LR			
		Z	S	M	B	Z	S	M	B
sgn (ΔV)	PO	GZ	GZ	GS	GS	GB	GB	GM	GM
	NE	GB	GB	GM	GS	GZ	GZ	GS	GS
		G							

		V: Z	S	M	B
W	Z	Z	Z	Z	Z
	S	S	S	S	S
	M	M	M	M	B
	B	M	M	M	B
		U			

The reported simulation and experiment results were quite successful, and the only problem was not connected with the fuzzy control, as one might expect, but with the vision system. The reflected glares on the floor due to the lamps on the ceiling were occasionally mistaken for obstacles, and sometimes strong surrounding reflections influenced the detection of boundaries.

In [26, 27] the same problem of motion planning and control on pilot level was solved in a different way. The rules were defined as a prescription of motion execution command to the actuator controller, taking into account that the present state would be changed to the desired one in a specified time. It was necessary to construct the robot environment representation, and robot self-representation was constructed.

The pilot used a "self-concentrated" reference frame for environment representation, where the location of each object in relation to the robot was defined by two polar coordinates: distance and angle. The robot self-representation contained two locomotion parameters, velocity and acceleration, and one steering parameter, angle of steering.

The control rules were defined as order triplets of present state, final state, and acceleration command. Rules were obtained from verbal description of analytical systems models and constraints derived from an optimal criterion of minimum time control. That means "bang-bang" change of acceleration. Two types of rules were used: acceleration–deceleration (braking) rules and steering rules. Acceleration–deceleration rules were of the form:

$$\begin{pmatrix}\text{present velocity}\\ \text{final velocity}\\ \text{goal distance}\end{pmatrix} \rightarrow \text{acceleration or deceleration}$$

Similarly, steering rules were of the form:

$$\begin{pmatrix}\text{present velocity}\\ \text{goal distance}\\ \text{goal angle}\end{pmatrix} \rightarrow \text{steering angle}$$

A typical example of acceleration–deceleration rules for static objects when the final robot velocity is zero, because the object velocity is zero, is this:

If robot velocity is *medium fast*, and object (goal) distance is *close* then acceleration–deceleration is *hard brake*

If several rules are triggered, a strategy of minimum time locomotion would be used. For example, for acceleration control that means to choose the highest value of acceleration. To illustrate this procedure, let us suppose that the obtained acceleration command is $ACC = \{0.8/HB, 0.8/B, 0.6/SU\}$, where HB is hard brake, B is brake, and SU is speed up. First a certainty level α is defined and than maximum acceleration is chosen, whose membership value is higher than that certainty level α. For higher certainty level, $\alpha = 0.7$, it is "brake," and for lower, $\alpha = 0.5$, it is "speed up." The certainty level could be treated as a kind of hazard measure. If it is low, the behavior is more hazardous, which means that the obstacle will be reached in shorter time, but in that case it might happen that the robot doesn't stop in time. The authors have concluded that the main disadvantage of a knowledge-based approach to robot motion planning and control is the difficulty in analyzing the controllability of the system and the stability of the control loop. Also, a large number of max–min operations must be computed and this is not possible in real time without special fuzzy hardware.

Let us now explain how a fuzzy approach could be used in motion planning and control of the robot hand in the case of "eye-hand coordination." The main idea is to control the motion of the robot hand using information obtained from vision sensor, the robot eye. The control task stated as "perceive and follow point light source in two-dimensional space" will be considered, but first let us describe principles of the fuzzy eye [14].

As a rule the process of vision has three stages:

1. the optical stage when an image of the outside world is projected on the retina
2. the transduction stage when the light-sensitive visual cells absorb photons and respond by generating electrical signals
3. the physiological stage when these primary signals are analyzed [28]

The idea of the fuzzy vision is primarily connected with the second and the third stage. Fuzzy vision sensor (fuzzy eye) is conceived as an array of light-sensitive elements (photo detectors) arranged in a way that a quite simple analysis of electric signals generated by them is sufficient to detect the position of the light source. This vision signal is used as input information to the controller, whose task is to control the actuator. Generally, the geometry of an eye is quite different for the eye that requires an image of a point source of light and the eye of the extended bright field that normally is encountered by diurnal animals. The idea of the fuzzy vision based control is here introduced and explained in the first case where the main control task is to see and then to follow a point source of light. A typical example is a lamp or a small luminescent object on a dark background. To simplify the presentation, a one-dimensional case will be considered. The control task is to see and to point by an actuator at the light source positioned on the circle. Figure 3.18 shows situation schematically. In the center of the circle is the vision sensor and actuator, the pointing device. The final goal is to position the pointing device in the direction of the light source.

Since nature is an unlimited source of ideas, the inspiration for the fuzzy eye comes from nature, too. In the animal kingdom, two types of eye are encountered. The first one is a well known lens eye characteristic of vertebrate animals, and the second one is a compound eye characteristic of invertebrate animals or, more precisely, insects. A man-made copy of a lens eye is a conventional video camera. Here we are proposing a fuzzy eye as a man-made copy of a compound eye or, to be more precise, of an apposition compound eye [28]. Figure 3.19 shows the cross section of the apposition compound eye and the fuzzy eye.

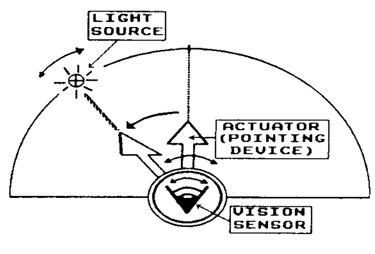

Figure 3.18 *A light source following based on the principle of eye-hand coordination.*

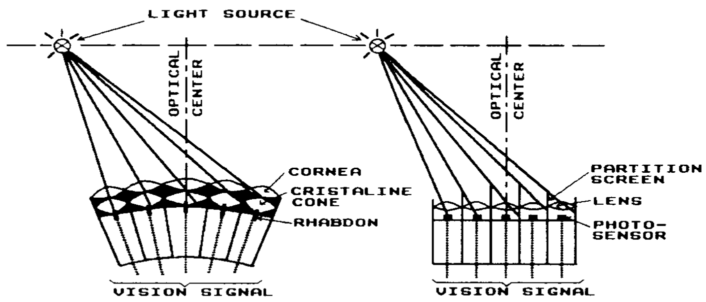

Figure 3.19 *Cross sections of the apposition compound eye (left) and the fuzzy eye (right).*

The apposition compound eye consists of sections called ommatidea. In each ommatidea light passes through the cornea, which acts as a lens, and the crystalline cone, exciting a light-sensitive sensor called rhabdon. In the rhabdon, light is converted into a vision signal. The amount of light that the rhabdon receives is biggest when the light passes through the optical axis of the rhabdon. As the light source moves away from the optical axis, the amount of light the rhabdon received decreases rapidly. Ommatidea are arranged so that each one "covers" one part of the space. When the light source is positioned in its part of the space, the vision signal of that ommatidea will be biggest, but other ommatidea in its vicinity will be excited, too. As a result the vision signal received from a set of ommatidea carries information about the position of the light source. A real compound eye is an array of ommatidea, so it can detect a light source in two dimensions, and binocular vision with two compound eyes could be used to detect the light in three-dimensional space.

The organization of the fuzzy eye is quite similar, as Figure 3.19 shows, but the fuzzy eye has an additional feature, which we describe below.

Light sensors are photo detectors constructed in such a way that lenses are incapsulated together with the sensitive element. The sensitivity of such sensors also decreases with angular distance from the photo detector center line (optical axis), similar to the sensitivity of the rhabdon. Even the shapes of sensitivity curves of both natural and man-made light sensors are qualitatively quite similar. They are both "bell-like" curves shown on Figure 3.20. The angular distance between the light source and the sensor optical center is plotted on x-axis and the relative sensor sensitivity on y-axis.

The fuzzy eye has one additional feature as a result of introducing the partition screens between photo sensors. These screens have a strong influence on the vision signal. Changing the screen height, we can manipulate the shape of the obtained vision signal and assign more-or-less importance to certain photo sensors.

The vision signal carries information about a light source position. Usually this position is determined according to the optical center of the eye. For the case

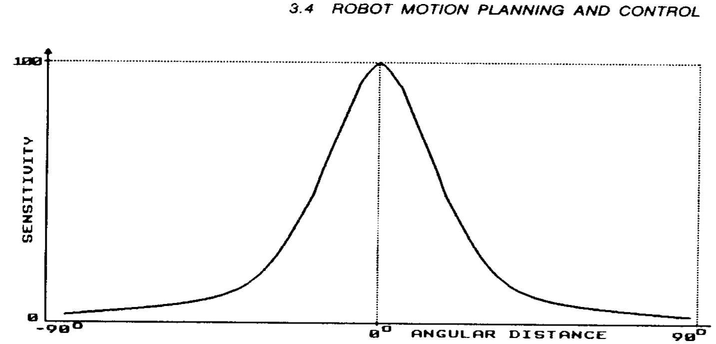

Figure 3.20 *Sensitivity of photo sensors as a function of angular distance from the sensor's optical center.*

shown in Figure 3.19, the intensity of the signal produced by the sensor in the left corner is the biggest, and the last three sensors on the right are not affected by the light. After we normalize, the vision signal could be expressed by a vector, for example [1 0.5 0 0 0]. Numbers represent the relative intensity of each sensor signal normalized according to the highest value. This vector carries information about the light source position, and it could be seen and analyzed as a fuzzy vector. That is the reason why we have used the expression "fuzzy eye." Fuzzy vectors could be used as an input to the controller that controls the velocity and the position of the actuator.

Figure 3.21 shows two typical examples. The first one shows vision signal of

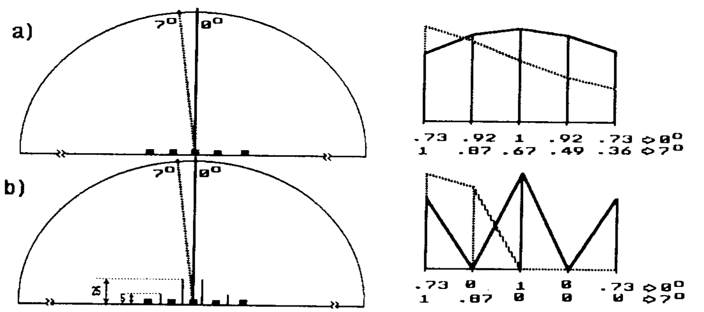

Figure 3.21 *Fuzzy eye geometry and the fuzzy vision signal for a light source position at the optical center and seven degrees left:*
a) Without partition screens.
b) For partition screens 0, 5, 25, 5, 0 mm high, looking from left to right.

a fuzzy eye without partition screen and the second one a vision signal of a fuzzy eye for certain combinations of screen height. Light was positioned at the optical center and seven degrees left of the optical center.

According to Figure 3.18, our final goal is to position the actuator (pointing device) in the direction of the light source. The actuator and the fuzzy eye have independent motors, so each one is able to rotate independently. When the light turns on, the fuzzy eye produces the vision signal in the form of a fuzzy vector. From the shape of this vector it is possible to determine the approximate position of the light. This information is used as an input to the rule based fuzzy control algorithm, which gives the direction and velocity of fuzzy eye rotation. The eye will stop rotating when its optical center line intersects the light. The pointing device starts to rotate at the same time as the fuzzy eye, but the control algorithm takes care that the eye is positioned first. Then final precise adjustment of the pointing device position is made using the precise information of the fuzzy eye location.

In this section few examples of robot motion planning and control using fuzzy set theory were described. Planning procedure is usually based on internal models representative of the robot surroundings, and fuzzy models are quite appropriate for internal model construction if the robot surroundings are not precisely known. Our next task is to show how the fuzzy technique could be used in communication with robots by means of words and sentences of natural languages.

3.5 COMMUNICATION WITH ROBOTS USING NATURAL LANGUAGE

Giving instructions to robots using natural language is one of the challenging tasks of modern robotics. The procedure consists of three main parts: sensing, recognizing, and understanding natural language. In this chapter our interest focuses on the third and the most complicated part: making the robot understand commands given in natural language and having it act according to the instructions given therein. We will suppose that words are received and correctly recognized, but that their meaning is not yet understood. Fifteen years ago Gougen [1] wrote a visionary article about fuzzy robot planning and control using natural language. He emphasized that rigid syntactic foundation is not necessary for natural language understanding. That is a rather robust affair because the words of most natural languages have inherent fuzziness. No rigid boundaries could be drawn for their use. He gave a simple example of colors: There is a linear continuum of hues between red and yellow, and at no point there is a clear separation. Even the intermediate value "orange" doesn't help, because the boundaries of orange are equally unclear.

In connection with natural language communication with robots by means of instructions about robot motion, Gougen has proposed to map the words of a natural language by a semantic interpreter into an "intermediate representation language" (IRL). In its simplest form, an IRL consists of a sequence of fuzzy vectors, each vector containing *fuzzy length*, a fuzzy set defined on a interval of path lengths [0, L], and *fuzzy direction*, a fuzzy set defined on circle unit, represented with degrees from 0° to 360°. A subset of natural language words is used to express the appropriate kind of hints (H) which are in IRL a selection of fuzzy vectors. Gougen claims that such hints, and even a very simple one, can be sufficient for robot guidance through a very complex maze, but no simulation or experiment results are presented.

A couple of years latter Zadeh [29] introduced and more correctly defined *meaning representation language for natural language* (*PRUF*). His main assumption was that impression intrinsic in natural language is, for the most part, "possibilistic" in nature rather than probabilistic. So each proposition translates into a procedure P, which returns a possibility distribution Π. P is the meaning of the proposition, and Π is the information conveyed by proposition.

Let us use, for example, a simple proposition with an object description:

The blue box which is quite small and very close.

In PRUF the representation of this proposition is this:

(Object (category = box)
(n (color) = blue)
(n (size) = quite small)
(n (distance) = very close)).

where n is a fuzzy descriptor. Thus n(color) is a fuzzy set characterized by possibility distribution over perceived colors of box, n(size) descriptor contains the linguistic modifier "quite," which modifies the possibility distribution associated with the fuzzy linguistic value "small" and similar to that n(distance) with a modifier "very."

Although there is a close connection between the concept of a possibility distribution and that of fuzzy set, there is also a difference between them, especially in interpretation. The reader can find complete discussion in [29, 30].

A simple proposition is defined as p: X is F, where X is a variable taking values in a universe U, and F is a fuzzy set defined in U, which induces a possibility distribution Π_x numerically equal to F. Linguistically F is expressed by words of a natural language. For example, a proposition in linguistic form could be: "*Relative size is small*," where *relative size* is a variable defined in an interval [0%, 100%] and *small* is a fuzzy set whose possibility distribution could be defined by Figure 3.16.

In PRUF there are also four types of *translation rules:*

(I) Modification Rules

$$p : x \text{ is } mF$$

where m is the modifier (not, very, quite, extremely, etc.) that modifies possibility distributions, for example, not corresponds to the complement $(1 - \Pi)$, or very to the second power (Π^2).

Example: Relative size is *very small.*

(II) Translation Rules

$$p = q * r$$

where g, q, r are propositions and $*$ denotes the operation composition, for example, conjunction (and), disjunction (or), implication (if · · · then · · ·), and so forth

Example: q Object is close
r Speed is very small
p If object is close then speed is very small

(III) Quantifier Rule

$$p : QN \text{ is } F$$

where Q is a fuzzy quantifier (most, many, few, some, almost all, etc.)

Example: Almost all boxes are black.

(IV) Qualification Rules

$$P : p \text{ is } \gamma$$

where γ might be a truth value $- \tau$ (true, not true, very true, more or less true, not very true, etc.), a probability value $- \lambda$ (probable, not very probable or likely, not very likely, etc.) or a possibility value $- \omega$ (quite possible, almost impossible, etc.)

Example: Object is *very close* is *not very true.*
Object is *very close* is *quite likely.*
Object is *very close* is *very possible.*

In conclusion, we quote Zadeh, "PRUF bears the same relation to fuzzy logic that predicate calculus does to two-valued logic. Thus it serves to translate a set of premises expressed in natural language into expressions in PRUF to which the rules of inferences may be applied, yielding other expressions in PRUF, which upon retranslation become the conclusion inferred from the original premises" [30, p. 4].

As an example of giving instructions to robots in terms of natural language using fuzzy set theory, a work of Uragami et al. [31] will be presented. A three legged "Inchworm robot" moved in real space under control of a program expressed in terms of natural language. Four types of fuzzy-linguistic instructions were used:

1. Go about n steps
2. Go to x
3. Go about n steps to x
4. Turn to the right (left)

with two supplementary instructions, which give initial values to the system

5. Head east/west/north/south.
6. Start from x.

References of the real world were given in virtual space by a data structure map, which contains the robot's actual position and gives information of the robot's immediate surrounding. An example of a town map and a data structure map is given in Figure 3.22.

Fuzzy linguistic instructions were transformed into machine instructions using the max method. For example, the linguistic value "about n steps" is defined as "normalize a fuzzy set having the membership value equal to 1 for steps whose value equals n." Figure 3.23 shows a fuzzy set "about 15 steps."

If robot is at position x with "head east" and has to execute instructions, "Go about 15 steps," it will stop near the church because for the church (C) the value $\mu_{\text{about15}}(X_c)$ is maximum. Machine instructions for degrees below a certain threshold are not chosen. For example, for the threshold shown in Figure 3.23 intersection 1 is abandoned. If there is no solution, back tracking to the previous instruction occurs and the machine instruction with the second highest degree is executed. For example, if we go back to the instruction "go about 15 steps" for the second time the robot will stop near the restaurant $R1$.

The second type of instructions is "go to x." If x is not specified, the nearest x is chosen. For example, if the robot starts from x and has to execute "go to

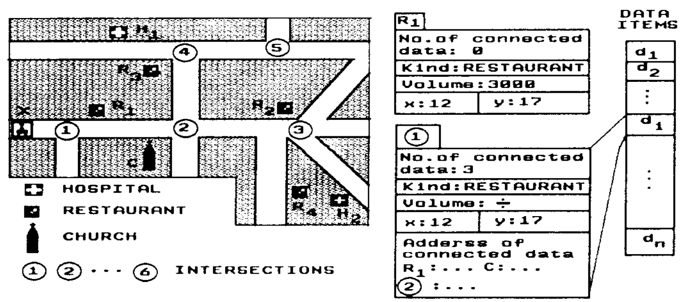

Figure 3.22 Town map and data map (robot can stop only at data items).

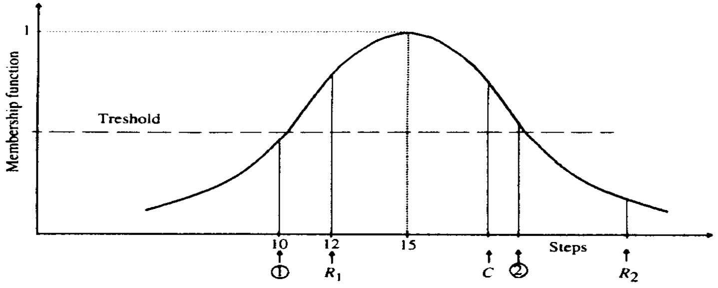

Figure 3.23 Membership function of the fuzzy set "about 15 steps."

restaurant," it will stop at R_1, but if specific instruction is given "go to R_2," it will go to R_2.

The third type of instructions is "go about n steps to x." Now again a figure like Figure 3.23 is used. For example, for the instruction "go about 15 steps to restaurant," the robot will stop near the restaurant R_1.

In object size description, three linguistic values were used: "large," "middle," and "small," defined with adequate fuzzy sets. An example is shown in Figure 3.24. If x is again the starting point and the instruction is "go to small restaurant," the robot will stop near R_2 because the membership value of fuzzy set "small" is maximum at point R_2.

The last type of instruction is "turn to the right (left)." The membership function of direction is shown in Figure 3.25. Again the maximum principle is used, so it the robot is on intersection 3 and instruction is "turn to the right," it will go

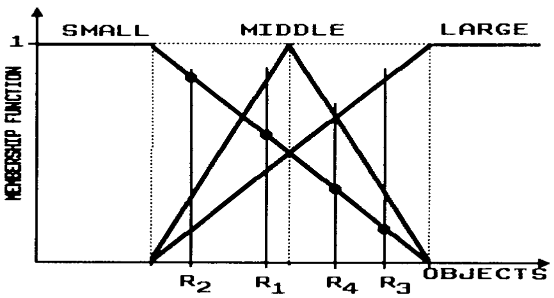

Figure 3.24 Membership functions for object size values definition.

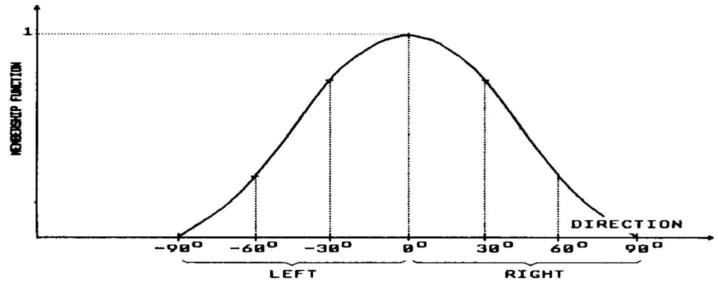

Figure 3.25 Membership function of a fuzzy set for direction.

to house H_2, and not to restaurant R_4, because for the street leading to H_2 the direction angle is smaller and degree from Figure 3.25 is higher.

Uragami and his collaborators reported a number of successful simulation trials. Their conclusion was that by using fuzzy linguistic instructions as commands, people will be able to handle robots more easily, especially in situations where it is necessary to handle a robot from a remote place. Typical examples are explorations under water and in outer space.

Another interesting example of communication between a person and robot by using natural language words is the control of a robot gripping force reported in [32]. The commercial fuzzy inference controller was used and a force feedback loop was arranged according to Figure 3.26. When gripping an object, the experimenter made a judgment of the attributes of the object. Then using natural language expressions he communicated this judgment to a fuzzy inference controller. Words as "heavy," "fragile," and "slippery" were used. Then by a conventional fuzzy control algorithm, based on control rules, the appropriate gripping force was determined. An example of a gripping force rule is

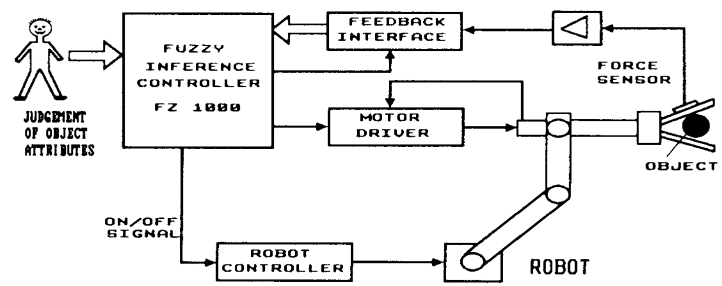

Figure 3.26 Configuration of gripping system according to [32].

If the object is *heavy*, *slippery*, and *nonfragile* then the desired level of gripping force is *high*.

The fuzzy controller was also used to keep the desired gripping force. For this task, the feedback signal from strain gauges was employed together with control rules of the form:

If error between desired force and real force is *positive small* and change-of-error is *positive big*, then open the hand *slightly slower*.

The third task of the fuzzy controller was to send start and stop signals to the robot controller, and to issue the instruction to open the hand at the end of the operation. Robot dynamics was controlled by a conventional controller.

This last application can be treated as a particular case of the idea of fuzzy transducer [13] where human sensor could be used to determine variable values. Results of human observations expressed in natural language, together with signals measured conventionally, form a basis for an intelligent measuring system and could be used for control purposes, too.

We believe that in the future more applications like this will be registered, because fuzzy reasoning is particularly suitable for the representation of meanings in natural languages.

3.6 CONCLUSION

In this chapter is was shown that fuzzy reasoning has already produced some interesting applications in the field of robotics. The control of robot dynamics, analysis, and interpretation of information obtained from robot sensors, particularly visual information, planning, and control of robot motion and communication with robots using natural languages are the areas where most theoretical and experimental work has been done, but the range of applications in other fields is growing steadily.

Fuzzy set theory gave to robotics more flexibility, robustness, and adaptability, and these are characteristics of human problem solving. Fuzzy set theory gave simple but powerful practical methodology for transfer of human knowledge, especially heuristic knowledge based on intuition and experience, to artificial robot systems, providing them with at least some primitive features of human intelligence.

REFERENCES

[1] J. A. Gougen, "On Fuzzy Robot Planning," in C. A. Zadeh, et al. Eds., *Fuzzy Sets and Their Application to Cognitive and Decision Processes*, Academic Press, New York, 1974.

[2] J. Mayers and Y. S. Sherif, Application of fuzzy set theory, *IEEE Trans. SMC*, 15(1), 175–189 (1985).

[3] K. Self, Designing with fuzzy logic, *IEEE Spectrum*, 42–44 (November 1990).

[4] H. Scharf, N. Mandič, and E. H. Mamdani, A self-organizing algorithm for the control of a robot arm, *Internat. J. Robotics and Automation*, 1(1), 33–41 (1986).

[5] G. Lee, Robot arm kinematics, dynamics and control, *IEEE Computer*, 15(part 12), 62–80 (1982).

[6] Y. F. Li and C. C. Lav, Development of fuzzy algorithms for servo systems, *IEEE Control Syst. Mag.*, 4, 65–72 (1989).

[7] C. Ken, L. Jin n-Ya, and Z. Y. Xiang, Fuzzy control of robot manipulator, *Proc. 1988 IEEE Internat. Conf. on Systems, Man, and Cybernetics*, Vol. 2, 1210–1212 (1988).

[8] M. Arfo, Auto-tuning system for robots, *Proc. Internat. Workshop on Fuzzy System Applications, IIZUCA-88*, 263 (1988).

[9] R. Tanscheit and E. M. Scharf, Experiments with the use of a rule-based self-organizing controller for robotics applications, *Fuzzy Sets and Systems*, 26, 195–214 (1988).

[10] D. Stipaničev, M. De Neyer and R. Gorez, Self-tuning self-organising fuzzy robot control, *Proc. IFAC Symp. on Robot Control SYROCO '91*, Vienna (September 1991).

[11] R. Palm, Fuzzy controller for a sensor guided robot manipulator, *Fuzzy Sets and Systems*, 31, 133–149 (1989).

[12] R. P. Sobek and R. G. Chatila, Integrated planning and execution control for an autonomous mobile robot, *Artificial Intelligence in Eng.*, 3(2), 103–113 (1988).

[13] J. Božičević and D. Stipaničev, Development of fuzzy transducer, *Proc. IMECO 88 Word Conference*, Austin, Tex. (1988).

[14] D. Stipaničev, Fuzzy vision and fuzzy control, *Proc. 13th IMACS World Congress*, (July 22-26 1991).

[15] M. M. Gupta, G. I. Knopf, and P. N. Nikiforuk, "Edge Perception Using Fuzzy Logic," in M. M. Gupta and T. Yamakawa, Eds., *Fuzzy Computing: Theory, Hardware and Applications*, North Holland, Amsterdam, 1988, pp. 35–51.

[16] D. Lakov, Adaptive robot under fuzzy control, *Fuzzy Sets and Systems*, 17, 1–8 (1985).

[17] D. Jain and S. Haynes, "Imprecision in Computer Vision," in P. A. Wang, Ed., *Advances in Fuzzy Sets, Possibility Theory and Applications*, Plenum Press, New York, 1983, pp. 217–236.

[18] M. M. Trivedi, C. X. Chen, and S. B. Marapane, A vision system for robot inspection and manipulation, *IEEE Computer*, 22, 91–96 (June 1989).

[19] R. N. Lea and L. B. Johnson, Fuzzy sets and autonomous navigation, *Proc. SPIE Conf. on Application of Artificial Intelligence*, 786, 448–452 (1987).

[20] K. Hirota, Y. Arai, and S. Hachisu, Moving mark recognition and moving object manipulation in fuzzy controlled robot, *Control Theory and Advanced Technol.*, 2(3), 399–418 (1986).

[21] H. Tahani and J. Keller, Information fusion in computer vision, *IEEE Trans. SMC*, 20(3), 733–741 (1990).

[22] W. Pedrycz, Fuzzy set theory in pattern recognition: methodology and methods, *Internat. Workshop of Fuzzy System Applications, IIZUKA-88 (tutorials)*, 51–64 (1988).

[23] D. Stipaničev, Fuzzy relational models for intelligent control, *Proc. IMACS Annals on Computing and Applied Mathematics MIM-S^2, 90*, IV.B.2.1–2.5 (1990).

[24] A. N. Averkin and S. K. Dulin, Fuzzy modelling relations for robots, *Proc. IFAC Symp. on Artificial Intelligence*, 331-336 (1983).

[25] T. Takeuchi, Y. Nagai, and N. Enomoto, Fuzzy control of a mobile robot for obstacle avoidance, *Information Sci.*, 45, 231-248 (1988).

[26] J. C. Isik, and A. M. Meystel, Pilot level of a hierarchical controller for an unmanned mobile robot, *IEEE J. Robotics and Automation*, 4(3), 241-255 (1988).

[27] C. Isik, Inference engine for fuzzy rule based control. *Internat. J. Approximate Reasoning*, 2, 177-187 (1988).

[28] J. N. Lythgoe, *The Ecology of Vision*, Clarendon Press, Oxford, 1979.

[29] L. A. Zadeh, "PRUF—A Meaning Representation Language for Natural Language," *Intern. J. Man-Machine Studies* 10, 395-460 (1978).

[30] L. A. Zadeh, Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets and Systems*, 1, 3-28 (1978).

[31] M. Uragami, M. Mizumoto, and K. Tanaka, Fuzzy robot control, *J. Cybernetics*, 6, 39-64 (1976).

[32] T. Ishida, Sensory information control system, *Proc. Int. Workshop of Fuzzy System Applications*, IIZUKA-88, 261-262 (1988).

CHAPTER 4

A Fuzzy Cognitive Structure: Foundations, Applications, and VLSI Implementation

W. PEDRYCZ
J. DIAMOND
R.D. MCLEOD

4.1 INTRODUCTION

This chapter deals with the VLSI implementation of a fuzzy cognitive system. Aspects of fuzzy set theory and neural networks are combined to produce a robust system for representing and handling uncertainty. A novel architecture for digital neural networks is introduced and analyzed. The system concept is implemented in a three-chip set which incorporates fault tolerance and a cellular automata based built-in self-test. VLSI design trade-offs are explored and details of the implementation are presented.

Specifically, an artificial neural network that exploits pre- and postprocessing stages, based upon fuzzy set theory, will be discussed. The structure resembles that of a referential cognitive system. This will be followed by a brief introduction to VLSI implementation strategies with the focus being primarily upon digital realizations. This section will discuss some of the problems common to implementing VLSI with particular attention to testing and reliability issues particular to artificial neural and fuzzy logic types of systems. The final section will overview the three-chip set prototype designed and fabricated during this project. The chapter is self-contained: The reader will be familiarized with essentials of fuzzy set theory, fundamentals of VLSI design, and basic ideas of test and fault tolerance.

Recent progress in the development of fuzzy set theory has been centered around areas such as fuzzy controllers, expert systems, digital signal and image processing systems, and robotics [1, 2]. With this progress has come an increase in system complexity, meaning that some software driven systems are very slow.

Fuzzy, Holographic, and Parallel Intelligence, By Branko Souček and the IRIS Group.
ISBN 0-471-54772-7