Fuzzy Logic Controllers. Advantages and Drawbacks.

Pedro Albertos and **Antonio Sala**
Dept. de Ing. de Sistemas y Automática.
Universidad Politécnica de Valencia
P.O. Box 22012, E-46071 Valencia
E-mail: pedro@aii.upv.es, asala@isa.upv.es

September 14, 1998

Abstract

In this paper, an analysis of the role of fuzzy logic controllers is carried out. Its interpretation and the conditions for successful implementation in several control structures, jointly with their advantages and drawbacks with relation to other advanced control approaches are discussed.

Keywords: Fuzzy control, fuzzy logic, advanced process control.

1 Introduction

Efficient control is tightly related to improvements in the quality of industrial production processes. In complex plants, a choice has to be made between the various available strategies (conventional, fuzzy and neural) developed in the last decade. In the industry, neural network-based controllers with learning capabilities are in a very initial stage. On the other hand, users accept with ease and interest a broad range of applications based in the fuzzy logic paradigm, due to the parallelism with the reasoning that operators do apply in some of their decisions. The ability fuzzy systems possess to explain their conclusion is an inherent user-interaction advantage with respect to the rest of the approaches. In the sequel, a detailed presentation of those advantages and drawbacks will be one of the main objectives of this work.

Granulation is one of the key aspects in the advantageous success of fuzzy logic: the treatment of information is divided in partially overlapping clusters whose linguistic labels stand as symbols. The creation of *rules* and *exceptions* produces a reduced complexity description of a system. The *local model* approaches to control [28] are somehow related to fuzzy logic in that sense. *Interpolation* complements granulation: smooth and simple system descriptions are created.

The basic structure of any control system is as depicted in figure 1, where two layers can be distinguished. At the local layer, the control function is implemented. Usually this is an analogue (or discretised) control action. The typical local controllers are PID like controllers with a feedback, feedforward or cascade structure. On the other hand, at the upper layer, a decision is made about either the operating conditions, the controller parameters, the proper structure of the local control or even the control goals. This layer is a logic or reasoning one and it is typically implemented by means of a simple device (relay, selector), a PLC, a piece of software code or a human action. One of the advantages of the fuzzy logic controllers (**FLC**), that is, controllers using the fuzzy logic concepts to compute the control action, is the ability to combine both layers activity: actions in both layers are described in the same language.

Other than the basic control actions above mentioned, advanced control algorithms imply a more complex control structure. Some of those with an impact in practical applications [34] are based on:

- the internal structure of the process to be controlled: State feedback, Decoupling, ...
- the knowledge of the model of the process: In-

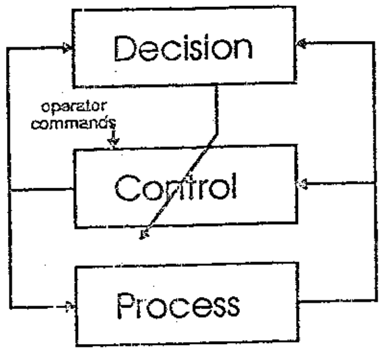


Figure 1: Control structure.

ternal Model Control, Generalised Predictive Control, ...

- the controlled system uncertainties: Stochastic, Adaptive and/or Robust control
- the nonlinearity of the process: Gain scheduling, nonlinear controllers, feedback linearisation, ...

The common characteristic of these approaches is that they use crisp (or stochastic) information for the description of both, the process model and the design criteria. In this sense, the user interface can not be very friendly because the user must know about the exact meaning of the parameters to be tuned. This is one of the reasons why basic controllers (like PID ones) with easy to follow rules of thumb for parameter tuning are so popular. But, in many industrial applications, the knowledge about the goals, the process, or both, is just approximated, and the interaction with the end-user should be based on this kind of knowledge. FLC are also able to provide an user-friendly interface to some of these advanced control structures. In a conventional control scheme, fuzzy systems can also carry out supervision tasks.

The paper is organised as follows. The basic concepts on fuzzy logic which are used in control are reviewed in the next section. The FLC's structure and operation are analysed in section 3. The different approaches to design a FLC are outlined in section 4. Then, some tools to develop and implement a FLC, as well as some applications are discused in section 5. Finally, a non-exhaustive summary of advantages and drawbacks of FLC is presented.

2 Fuzzy Logic Systems

Since ancient Greek times (binary) logic has been used to formalise reasoning, and a wide range of applications are present in most industrial processes either as PLC's or as expert systems or decision trees based on propositional calculus or first-order predicate logic axioms. Notwithstanding, an important aspect in industrial process control is uncertainty modelling, and the treatment of qualitative magnitudes not properly described by binary concepts. In the particular area of logic, several approaches have been devised to deal with the referred uncertainty and ambiguousness: evidential reasoning [35], bayesian inference [36], certainty factors [7], rough sets [25], and fuzzy logic [45]. In some cases, uncertainty arises due to assumptions that hold in most cases except a handful of exceptions. Non-monotonic logic [16] deals with those situations.

In the field of control design, the fuzzy logic alternative is the one that has gained most popularity. Fuzzy logic belongs to the class of multivalued logics [30, 40].

2.1 Fuzzy sets

Fuzzy logic was originated as a logic of inexact or ambiguous concepts. They are described by ***fuzzy sets*** over a universe U that attain a membership value $\mu : U \to [0, 1]$. Thus, they are generalised ordinary sets, which are described by a characteristic function $\xi : U \to \{0, 1\}$.

The meaning of that membership value (fuzziness) can pose interpretation problems, and in particular, its relationship to the well-established concept of probability. Some authors think of it as a particular *a priori* bayesian probability assignment [20]. In the particular case of industrial practice, the possible interpretation as the knowledge the control engineer has in mind, can lead to different membership assignments to an element of the universe set, and from the formal point of view, some operations can be allowed or forbidden depending on that interpretation.

The interpretation of fuzziness can fall into the following categories:

Probabilistic. The meaning of "the Temperature is 0.7 High" could be that 70% of engineers would say that the temperature being considered (for example, 80°C) is high, or that 70% of them would propose an interval of temperatures considered high that would include the referred value.

Metric. Fuzziness is considered as a kind of *conceptual distance* that can be compared, added, scaled, such that a distance of 1 means *contrary* concepts.

Possibilistic. A fuzzy set is a distribution of the "possibility" that a variable attains a certain numeric value given a purely linguistic description. This is the interpretation of the fuzzy modal logic approaches such as possibility theory [12, 13].

Others. Such as utility (optimisation of an implicit cost function in the mind of the designer) or ordinality (only comparisons between logic values are allowed).

The reader is referred to [14], for example, for a more detailed discussion of these issues. The key idea here is that membership assignment and its interpretation can be one of the drawbacks of the fuzzy approach.

Fuzzy connectives. Conjunction, disjunction and negation of fuzzy sets are described by triangular norms (T), conorms (S) and the 1-complement, respectively:

$$\begin{aligned} \mu_{A\cup B}(x) &= T(\mu_A(x), \mu_B(x)) \\ \mu_{A\cap B}(x) &= S(\mu_A(x), \mu_B(x)) \\ \mu_{\neg A}(x) &= 1 - \mu_A(x) \end{aligned} \tag{1}$$

The relevant properties and a great set of those norms and conorms can be found in [18, 42]. The most prevalent choices are minimum and product (as conjunctions) and the maximum (as disjunction). If other choices are used, they usually verify the De Morgan duality law: $T(a, b) = \neg(S(\neg a, \neg b))$. More than a dozen of such pairings (and some of them including infinite choices depending on some parameters) can be found in the cited references.

2.2 Rule-based fuzzy systems

In the context of industrial control and expert systems if fuzzy propositional calculus is applied, the set of propositions to encode the knowledge is divided into premises and rules. Premises are obtained from either the user or a sensor input. Conclusions are obtained so they can be used as new premises to other rules. Rules are expressions usually in *implication* form:

> If x is A_i (antecedent), Then y is B_i (consequent)

Antecedent and consequent can be atomic proposition, as in the example or complex ones (only in antecedent in most applications) formed by conjunction and disjunction of atomic ones.

Generalisation of the binary logic equivalences for IMPLICATION ($\Rightarrow$) and DOUBLE IMPLICATION ($\Leftrightarrow$), for example, $\neg A + B$ and $\neg(A \oplus B)$ respectively, to the fuzzy case is not obvious, giving also rise to various interpretations [19].

Errors in the definition of rules can originate anomalies [29] such as redundancy, inconsistency, incompleteness, etc. Formal validation methods should be set up in a general case to check for these properties [31].

2.2.1 Evaluation of fuzzy systems

The evaluation process consists on extracting a numeric conclusion from a set of data via a rulebase. It is based on several steps.

Premise information is provided at the first step.

In *ordinary inference*, premises are propositions with known logic value (in most cases, obtained by *fuzzification* of sensor measurements). For example, if N linguistic concepts (fuzzy sets) are defined for a variable in a universe U, the fuzzification operation transforms a crisp value u into an N-length vector of *fuzzy coordinates*: $FZ(u) = (\mu_1(u), \ldots, \mu_N(u))$.

The second step is rule evaluation (inference). Several options are discussed in literature, such as obtaining consequent fuzzy coordinates via a matrix rule operator $\mathbf{b} = \bigvee_{j=1}^{m} t_{ij} \wedge \mu_j(u)$, or obtaining individual consequent fuzzy sets $\mu_{C_i}(y) = \mathcal{I}(\mu_{B_i}(y), \mu_{A_i}(u))$.

The third step is the combination of conclusions from individual rules, usually by means of a T-conorm.

The last step is *defuzzification*, i.e., the conversion of fuzzy conclusion information to a crisp output, needed to actually operate upon the process being controlled. This operation can be carried out with averaging formulae on the fuzzy coordinates b (thus including in the same expression the previous combination step) or otherwise operating on an aggregated conclusion set obtained from individual C_i in the previous step.

For further information on fuzzification-inference-defuzzification schemes the reader is referred to [18, 41, 42, 31].

Generalised inference deals with obtaining conclusions from linguistic or ambiguous sensor input expressed as fuzzy sets (interpreted as possibility distributions). It is usually carried out by means of *compositional* rules of inference [46].

One of the main concerns for real time operation of rule based systems is the problem of dimensionality. If the number of rules, facts and variables is high, the rule base should be very well structured and debugged to avoid scheduling problems. That claims for efficient ways of rule base inference and validation.

Different choices in the selection of membership functions, their assignment and interpretation, the selection of fuzzy connectives, the generalisation of logic equivalencies as well as the many options in the FLS evaluation introduce a great variety of possibilities but, in some sense, it results in a drawback for an easy use.

3 Fuzzy Logic Controllers

FLC can be considered as approximated version of already defined controllers or as a combination of a number of operator strategies or controllers. In the first case, the function approximation property of the fuzzy systems is used. In the second case, the combination of reasoning and function approximation is exploited.

3.1 Function approximators

In intelligent control applications [1, 4, 6], rule-based fuzzy systems are in most cases used as function approximators to emulate either a controller or a plant model (for which some other fuzzy or non-fuzzy control strategies may be devised).

The kind of models to be discussed in this paper are those that can abide to the notion of *locality* or *granularity* so that given an overall complex system, its input space is decomposed into "granules" with similar behaviour, being that behaviour given by a "local description". The local description may be given by means of traditional functional expressions or by alternatives such as the fuzzy logic models here discussed.

The core of the model definitions will be a static model of a target function $f : X \to Y$. This static model will be defined via "granules" and "local description" as just mentioned, being the domain (input universe) the plant state (or a reduced subset of it) and/or the input variables. The output universe is either the control action (fuzzy regulator) or the state space (fuzzy model). Hence, dynamic system models are obtained with the static function f acting as part of a bigger structure such as:

$$\dot{x} = f(x) \qquad x_{k+1} = f(x)$$

for continuous and discrete models, respectively [1]. Delay lines are a common way of generating a state vector for discrete-time systems (extending FIR, ARMAX or BJ settings) and filter banks (i.e., noise-filtered measurements and trends -derivatives-) for continuous-time ones.

Given a static function $f : X \to Y$, an order-0 fuzzy model of that function is defined as a set of rules:

$$\text{If } x \text{ is } A_i, \text{ then } y \text{ is } B_i \quad i = 1, \ldots, n_r$$

where $x \in X, y \in Y$, and A_i, B_i are fuzzy subsets of X and Y respectively, characterised by membership functions $\mu_{A_i} : X \to [0,1]$ and $\mu_{B_i} : Y \to [0,1]$. If formal validation is used for knowledge acquisition supervision, the shape of the consequents has to abide to certain requirements implied by the fuzzy extension principle, even in the case that only centroid and area information may be used for defuzzification [32].

In order to interprete a fuzzy system using the notion of "locality", the fuzzy sets A_i must verify certain restrictions:

- Distinguishability: different fuzzy sets should correspond to different regions (granules) of the input space. This can be expressed by:

$$\sum_{i=1}^{n_r} \mu_{A_i}(x)^p \leq 1$$

where $p \geq 1$ is a parameter for the restriction strength.

- Convexity: $\forall\, x_1, x_2 \in X$

$$\mu_{A_i}(\alpha x_1 + (1-\alpha)x_2) \geq \min(\mu_{A_i}(x_1), \mu_{A_i}(x_2))$$

- Coverage:

$$\sum_{i=1}^{n_r} \mu_{A_i}(x)^p \geq 1$$

being in this case $p \leq 1$.

Add-1 *partitions* are a characteristic example of locally-interpretable fuzzy set distributions. Multidimensional partitions are usually constructed via cartesian product of one-dimensional (triangular or trapezoidal) add-1 partitions. The *curse of dimensionality* is present in many multivariable cases (with 3 or more input variables). Gaussian coverings are also often used, mainly in the neurofuzzy context. Clustering techniques may avoid the exponential increase in the number of rules.

A key result is the ability of fuzzy systems to be universal function approximators under mild conditions (Stone-Weierstrass theorem [41]), i.e., any countinuous function in a compact set can be approximated within any desired error bound with a certain number of rules (local models).

3.1.1 Tagaki-sugeno models

If the rules that model the function take the form:

If u is A_i, then y is $f_i(u)$ $\quad i = 1, \ldots, n_r$

where $f_i : U \to Y$ are component-models of the function, then the fuzzy model is called Tagaki-Sugeno model. If sets A_i are locally interpretable, then f_i are named "local models".

The model output is usually obtained as a weighted sum of the local-models output,

$$y = \frac{\sum_{i=1}^{n_m} \mu_{A_i}(u) f_i(u)}{\sum_{i=1}^{n_m} \mu_{A_i}(u)} \qquad (2)$$

If $f_i(u)$ are constant, those models reduce to the previously outlined ordinary inference averaging defuzzifier expressions. In that case, and also if f_i are linear in parameters, the whole fuzzy system belongs to the class of ***linear in parameter approximators***.

This kind of typology is very similar to some widely referenced ***functional-link*** or ***neuro-fuzzy*** systems [24, 17, 42].

3.2 FLC Structure

The simplest structure of a FLC is a direct translation of a fuzzy system used in real time. It requires a couple of input/output devices (the fuzzifier and defuzzifier) to convert crisp data to fuzzy variables and viceversa. As previously mentioned, the number of input (output) variables should be low (generally no more than two) to avoid an explosion in the number of rules. Otherwise the controller resolution, related to the number of linguistic variables and its user-interpretability, or the computing time are questionable. The real-time operation is a strong constraint for FLC. It does not allow for the possibility of reasoning loops and, in general, only one level of reasoning is used.

The structure of a fuzzy controller is depicted in figure 2. Different controller structures can be set up depending on the preprocessing and postprocessing operations, and the inference block must meet the real-time constraint above mentioned. For example, a PID-like FLC using a bidimensional input space is depicted in figure 3. Several fuzzy controllers can be combined in decentralized control

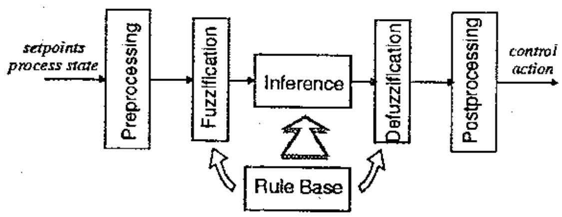


Figure 2: Fuzzy regulator structure.

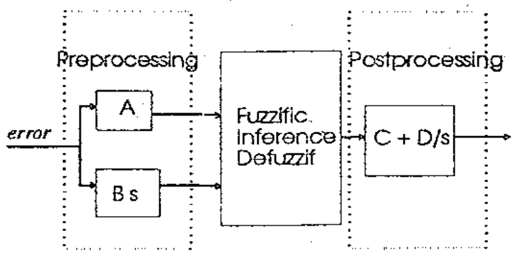


Figure 3: PID-like fuzzy regulator.

structures. Other variables can be present in the rulebase acting, for example, as feedforward terms.

If the number of rules is very large, either because the number of physical variables (state feedback control, different modes of operation) or that of the fuzzy variables attached to each physical one is enlarged, the rule base should be structured. There are three basic options:

- *Hierarchy.* This is the case if the reasoning follows different levels of refining. For instance, if the number of linguistic variables is very high they could be accessed in two steps. First it is determined if the physical variable is positive/zero/negative and then if it is large/medium/small. A similar treatment can be planned for the case of many input variables, in which successive inputs refine the conclusions of previous rulebases, so approximating a complex function via an expression such as:

$$F(x_1, x_2, x_3) = f_1(x_1, f_2(x_2, x_3)) \qquad (3)$$

  in which f_1, f_2 are constructed by means of a fuzzy system.

- *Optional* In this case, only a reduced number of rules is applied, based on a previous level of decision and selection. This is the typical case for different and independent modes of operation. If the rule base is split into a number of partial rulebases, one or two of them are fired. Their evaluation can be also combined to get the final result. This approach can be used in, for instance, in piecewise linearised control of nonlinear systems. An example of the described setting could be:

  IF x_1 is *True* Then $y = f_1(x_2, x_3)$
  IF x_1 is *False* Then $y = f_2(x_2, x_3)$

  where f_1 and f_2 are implemented as a reduced rule-set fuzzy controller.

- *Multifunction.* In large-scale nonlinear applications, the rulebases can be divided into different groups according to their goals, being evaluated in a structured sequence. For instance, there may be two groups: ***filtering*** or ***state detection*** rules and ***action*** rules. Some of the state detection rules may also act as mode of operation switches, as previously mentioned.

  State detection:
  IF T_1 is *High* Then *Yield is Low*
  IF T_1 is *Low* Then *Yield is Intermediate*
  ...
  Action Rules:
  IF *Yield is Low* Then *Increase* Q_1
  ...

4 Fuzzy Controller Design

As mentioned in the introduction, FLC is one of the strategies devised to deal with control of systems subject to uncertainty, based on symbolic (linguistic, qualitative) information management for complexity reduction.

The use of fuzzy control at the decision or supervision level (fig. 1) is rather natural, as a way to express reasoning based on qualitative information. PID tuning [5] and adaptive control supervision [22] are well referenced applications.

Direct fuzzy control is suitable for application to:

- processes with modelling difficulties, either because it is unknown or it has a lot of adjustable parameters,
- processes with ambiguous specifications,
- processes already controlled by human operators to mimic them,
- processes with unreliable or imprecise sensor measurements.

In structures where a fuzzy model of the process is used, *identification* can be carried out by gradient backpropagation, Levenberg-Marquardt and other algorithms [9]. In linear in parameter-systems, least-squares algorithms and dead-zone gradient methods can ensure a quick convergence [27]. In recurrent structures (where some input components are previous *model* outputs), dynamic backpropagation has to be used, and backpropagation through time should be used if target values are delayed [44], as it is the case if the final positions are specified.

The plant *controller* design can be based on several strategies:

- Mimicking an expert operator knowledge: Examples of those are the applications to cement kilns [8] or waste water plants [39].
- Intuitive design of PID-like fuzzy controllers: If processes are relatively fast in human perception time-scale terms, operators have only qualitative control information very similar to the basic control actions. Many fuzzy systems are set up according to the structure depicted in figure 3. Depending on the values of the scaling coefficients A,B,C and D different strategies (fuzzy-P,PI,PD,PID) can be set up. The rulebases are very similar in most cases, and many optimisation schemes adapt only the scaling coefficients. Only if the rulebases are used to fine-tune asymmetric and nonlinear characteristics (for example, including in the rulebase the setpoint level) the performance of these schemes can be substantially better than that of conventional controllers [2].
- Porting any conventional controller into rulebase form, due to the fact of the universal approximation property. Typical constructs in this direction are fuzzy dead-zone-interpolated sliding mode controllers [42]. Other references [18] report pole assignment, state feedback, etc.
- Fuzzy combination of locally designed controllers (local model approach). If local linear controllers are devised for a plant described by a model in the form (2), then linear matrix inequality techniques can be used to test global stability by searching for a common quadratic Lyapunov function [37]. This is the interpolated generalisation of crisp *switching* controllers for setpoint changes from a PLC.
- Direct and indirect adaptive controllers. Model reference adaptive controllers for affine in control plants have been proposed, as well as indirect approaches such as identification and subsequent feedback linearisation. The reader is referred to [41, 27] for details. Other model predictive schemes can be adapted to previously identified fuzzy plant models.
- Direct learning and causal inversion of fuzzy models via transforming the rules in a set of equations [32]. Similarly, a neurofuzzy system can directly learn plant inverses by switching the role of input and output in the training process.
- Other neural-network related schemes based in neurofuzzy equivalence (reinforcement learning, backpropagation through time) [17, 44, 23].

Global stability analysis may be carried out by means of Lyapunov-La Salle theorems, passivity, small gain theorem, phase plane analysis, etc. [3]. In this context, fuzzy systems are considered as ordinary nonlinear functions.

5 Controller implementation

The actual implementation of fuzzy logic controllers has to be done via a piece of software on a microprocessor-based system. Depending on the level of interactivity with the end user, in the sense of his ability and ease to modify controller parameters either in on-line or off-line operation (simulation), several solutions may be possible:

- *Fuzzy logic in a chip*. Specially designed DSP chips, reading the controller parameters from a tiny sized RAM or EPROM, can quickly carry out simple fuzzy inferences [38, 43]. They may implement certain fuzzy inference algorithms or just perform a precompiled lookup table interpolation.
- *Intelligent PID regulators*. Some companies offer PID regulators with autotuning capabilities implemented via a set of rules describing a relation between overshoot, rise time, settling time and controller parameter increments [5].
- *Simple FLC routines*. Non-generalised fuzzy inference with centroidal defuzzification can be

easily written into a short piece of code. This code can be part of a wider PLC-like control structure, as already used in the microcontroller operations for photographic cameras or washing machines. Its complexity is not much grater than that of a conventional antiwindup-PID. The execution time depends on the number of rules.

- *Design-tools.* A lot of software packages have been developed for design, implementation and testing of FLC prototypes. Some of them are libraries of general-purpose calculus packages (fuzzy logic toolbox for *M*atlab), and some others are standalone applications (FULDEK [10], UNFUZZY [11], ...).

- *(Real-time) Fuzzy expert systems.* In large-scale applications with various control and supervision tasks, real-time expert systems have been developed in which real-time specifications are combined with fuzzy and nonfuzzy inference over large sets of rules (structured in different layers of rulebases). Examples of those are commercial packages (COGSYS, G2) and others developed in academia (RIGAS [8], CONEX [33]). Their flexibility is high but they imply a high computing overhead and they do not usually incorporate simulation and design tools.

5.1 Applications

The field of applications has been growing in the last decade. Other than the initial fuzzy control of a cement kiln, one of the pioner application proposed by Mandami [21], fuzzy control has been applied to a variety of processes. From appliances and domestic devices (dishwashers, cameras, ...) to really complex processes, such as wast-water treatment [39]. The presence of biological and sometimes not well understood processes call for approximate reasoning and computation.

The main point is the cooperative use of fuzzy logic techniques together with other well-stablished control techniques.

The hierarchical and integrated control structure is illustrated by the cement kiln control system RIGAS [8]. A number of controllers are provided for each one of the manipulated variables. According to the available computing time, always limited in a realtime multipurpose control system, the most suitable controller is selected and the best output is computed and forwarded to the local actuators.

As previously mentioned, some PID controller manufacturers provide some kind of autotuning involving the use of reasoning and fuzzy logic implementation [5].

Mobile robots trajectory planning and supervision is another field of application [15]. The unpredictable and changing environment is a suitable framework to use FLC.

6 Advantages and drawbacks

6.1 Advantages

Fuzzy logic is not the only way to reason with ambiguous concepts but it seems to be the most apt to function approximation in control engineering. From the previous considerations, some of the most important advantages the use of fuzzy logic can entail to control system design are here detailed:

- Flexible, intuitive knowledge base design. Control and supervision speak the same language.

- Convenient user interface. Easier end-user interpretation when the final user is not a control engineer.

- Easy computation. Widely available toolboxes and dedicated integrated circuits.

- Learning. Linear in parameter systems (in most cases) make possible least squares, dead-zone learning algorithms and other results from adaptive control.

- Validation. Consistency, redundancy and completeness can be checked in rulebases (knowledge acquisition supervision). That could speed up automated learning and improve user interpretability.

- Ambiguousness. Fuzzy logic is a "natural" way of expressing uncertain information. Research must be done in reasoning with incompleteness, i.e., concluding different actions depending on the ***possibility*** or ***necessity*** of certain plant situations. Some tools for it are already available [31].

- Combine regulation algorithms and logic reasoning, allowing for integrated control schemas.
- Its conceptual model (granulation, soft commutation [26]) has been used in many other paradigms (RBFN, local models).
- FLC can incorporate a conventional design (PID, state feedback) and fine-tune it to certain plant nonlinearities due to universal approximation capabilities.

6.2 Drawbacks

The previous discussion also points out that fuzzy logic is not the perfect solution for all cases. Its most relevant drawbacks may be summarised as follows:

- Experimental. Manual tuning in large-scale industrial applications. Time-consuming retuning even if applied to a similar plant in other location.
- Intuitive fuzzy PID-like design does not clearly outperform well-tuned conventional controllers.
- The performance-robustness tradeoff is not usually taken into account in FLC tuning. Robustness is often assumed in FLC as a fundamental property but, in principle, that is a false myth: mediocrely hand-tuned FLC happen to be robust because they do not exploit to a full extent the nonlinear plant characteristics so its performance is far from being stringent and time-optimal. A non-aggressive hand-tuned PID will share the same characteristic: robustness is not a "natural" property of fuzzy regulators.
- Many (localised) parameters. Manual local tuning is tedious but it can be an advantage for automated learning algorithms. A much simpler global description may exist, being hidden by too detailed local descriptions. As in most modelling situations, the compromise between accuracy and readability may be present.
- Many (unclear) options: thousands of different fuzzy system configurations may arise depending on conjunction, disjunction, implication and defuzzification choices. Some of the alternatives even stem from the root interpretation of the meaning of fuzziness.
- Many actual implementations are just equivalent to lookup-table interpolation schemes.
- Commercial packages use non-standard file formats, and have a huge software overhead for simple applications.
- Dimensionality. Cartesian product of partitions is the most used way of setting up antecedents in multi-dimensional models. That is a very inefficient and memory-intensive setting for most functions (even linear ones). Other approaches, such as multilayer neural networks, can learn and generalise in a much better way with a much more reduced set of parameters (number of rules or neurones), at the cost of a perhaps slower training progress and less understandability. If input data are not evenly distributed on a highly dimensional space, clustering prior to learning may be advisable.

  The ***curse of dimensionality*** makes nearly impossible in practice to set up a rulebase with more than three inputs (assuming from three to nine sets over each of them) while preserving end-user interpretability. If many inputs ar present, such as in control of complex large-scale industrial processes, the fuzzy system structure is set up as a collection of chained rulebases, for example as a hierarchy of fuzzy observers (filtering and estimation of unmeasured "state" variables), controllers (action rules) and supervisors (mode changes, anomalies detection) as previously discussed in the FLC structure section. Decentralized fuzzy controllers are also a way of dealing with a great number of inputs and outputs. Clustering techniques over experimental data can help to reduce the number of rules, but at the risk of losing the interpretability of the automatically obtained classes.

7 Conclusions

The paper has presented an outline of fuzzy logic systems and its use in control, oriented to raise the discussion of advantages and drawbacks when compared to other alternatives. The main conclusion is that advanced nonfuzzy strategies can outperform FLC in processes with a detailed mathematical model to work with (for example, robotics). However, in many industry situations, where models are either inaccurate or totally absent, FLC can integrate control and supervision tasks in a user-friendly way, even accommodating PID strategies. For stringent specifications, manual or automated learning can outperform conventional controllers but application of learning controllers is still restricted to academia prototypes.

References

[1] P. Albertos and J. Aracil. Analysis of fuzzy models and stability issues of fuzzy controlled systems. In C. Bonivento, C. Fantuzzi, and R. Rovatti, editors, *Fuzzy Logic Control: Advances in methodology*, pages 35–74. World Scientific, 1998.

[2] P. Albertos, M. Martinez, J.L. Navarro, and F. Morant. Fuzzy controller design: a methodology. In *Proc. IEEE Conference on Control Applications*, Vancouver, 1993.

[3] J. Aracil, F. Gordillo, and T. Alamo. Global stability analysis of second-order fuzzy control systems. In R. Palm, D. Driankov, and H. Hellendorn, editors, *Advances in Fuzzy Control*. Springer-Verlag, 1997.

[4] K.J. Åström, Anton J., and K.-E. Årzén. Expert control. *Automatica*, 22(3):277–286, 1986.

[5] K.J. Åström and T.J. McAvoy. Intelligent control: An overview and evaluation. In D.A. White and D.A. Sofge, editors, *Handbook of Intelligent Control. Neural, Fuzzy and Adaptive Approaches*, pages 3–34. Van Nostrand Reinhold, New York, 1992.

[6] L. Boullart, A. Krijgsman, and R.A. Vingerhoeds, editors. *Application of Artificial Intelligence in Process Control*. Pergamon Press, 1992.

[7] B.G. Buchanan and E.H. Shortliffe. *Rule-Based Expert Systems*. Addison–Wesley, Readings (MA), USA, 1984.

[8] C. Carrión, A. Crespo, J.L. Navarro, M. Martínez, and R. Vivó. RIGAS. Sistema inteligente para el control de procesos en la industria del cemento (in Spanish). *Cemento-Hormigón*, (733):645–665, June 1994.

[9] M. di Martino, S. Fanelli, and M. Protasi. Exploring and comparing the best direct methods for the efficient training of MLP-networks. *IEEE Trans. on Neural Networks*, 7:1497–1502, 1996.

[10] M.E. Dreier and G.A Cunningham. The fuzzy logic development kit for IBM PCs. In P.W. Ross, editor, *Handbook of Software for Engineers and Scientists*. CRC Publishers, 1996.

[11] O. Duarte. UNFUZZY: Software para el diseño, análisis, simulación e implementación de sistemas de lógica difusa(in Spanish). In *III Jornadas Iberoamericanas de Automática Industrial*, Cartagena (Colombia), 1997.

[12] D. Dubois and H. Prade. Necessity measure and the resolution principle. *IEEE Trans. on Systems, Man & Cybernetics*, 17:474–478, 1987.

[13] D. Dubois and H. Prade. *Possibility Theory–An Approach to Computerized Processing of Uncertainty*. Plenum, New York, 1988.

[14] D. Dubois and H. Prade. The three semantics of fuzzy sets. *Fuzzy Sets and Systems*, 90(2):142–150, 1997.

[15] A. García-Cerezo, A. Ollero, and Martínez. Design of a robust high performance fuzzy path tracker for autonomous vehicles. In *Proc. of the IAV'95*, Helsinki (Finland), 1995.

[16] S. Hanks and D. McDermott. Default reasoning, nonmonotonic logics and the frame problem. In A. Gupta and B.E. Prasad, editors,

Principles of Expert Systems, pages 203–208. IEEE Press, New York, 1988.

[17] C.J. Harris, C.G. Moore, and M. Brown. *Intelligent Control. Aspects of Fuzzy Logic and Neural Nets.* World Scientific, Singapore, 1993.

[18] R. Jager. *Fuzzy Logic in Control.* PhD thesis, Delft University of Technology, Delft (Netherlands), 1995.

[19] E. Kerre. A comparative study of the behaviour of some popular fuzzy implication operators on the generalized modus ponens. In L.A. Zadeh and J. Kacprzyk, editors, *Fuzzy Logic for the Management of Uncertainty*, pages 281–295. John Wiley & Sons, New York, 1992.

[20] D.V. Lindley. The probability approach to the treatment of uncertainty in artificial intelligence and expert systems. *Statistical Science*, 2(1):17–24, 1987.

[21] E.H. Mamdani. Application of fuzzy algorithms for control of simple dynamic plant. *Proc. IEE*, 121(12), December 1975.

[22] M. Martínez. *Indicadores para la Supervisión del Control Adaptativo. Implementación mediante la metodología de Sistema Experto (in Spanish).* PhD thesis, E.T.S.I.I., Universitat Politècnica de València, Feb. 1991.

[23] W. III Miller, R. Sutton, and P. Werbos, editors. *Neural Networks for Control.* The MIT Press, Cambridge, MA, 1990.

[24] Y.-H. Pao. The functional link net approach to the learning of real-time optimal control. In *Int. Symp. on Artificial Intelligence in Real-Time Control (AIRTC-92)*, pages 35–41, València (Spain), June 1992.

[25] Z. Pawlak. Rough sets. *Int. J. Inform. Comput. Science*, 11:341–356, 1982.

[26] W. Pedrycz and M. Reformat. Rule-based modeling of nonlinear relationships. *IEEE Trans. on Fuzzy Systems*, 5(2):256–269, May 1997.

[27] J. Picó. *Identificación y Control Adaptativo Neuroborroso de Sistemas Discretos no Lineales (in Spanish).* PhD thesis, Depto. Ingeniería de Sistemas, Comp. y Automática, Univ. Politécnica Valencia, 1996.

[28] J. Picó and A. Sala. Local models, learning and validation. In *Proc. of Portuguese Control Conference (Controlo'98)*, Coimbra (PT), September 1998.

[29] A. Preece and R. Shingal. Foundation and application of knowledge base verification. *Internationla Journal of Intelligent Systems*, 9:683–701, 1994.

[30] N. Rescher. *Many-valued logics.* McGraw-Hill, New York, 1969.

[31] A. Sala. *Validación y aproximación funcional en sistemas de control basados en lógica borrosa (in Spanish).* PhD thesis, DISA, Univ. Politécnica Valencia, 1998.

[32] A. Sala and P. Albertos. Fuzzy systems evaluation: The inference error approach. *IEEE Trans. on Syst. Man & Cybernetics*, 28B(2):268–275, 1998.

[33] R. Sanz, A. Jiménez, R. Galán, F. Matía, and E.A. Puente. Intelligent process control: the CONEX architecture. Technical report, DISAM, Univ. Politécnica de Madrid, 1991.

[34] D.E. Seborg. A perspective on advanced strategies for process control. *Modelling, Identification and Control*, 15(3):179–189, 1994.

[35] G. Shafer. *A mathematical theory of evidence.* Princeton University Press, Princeton (USA), 1976.

[36] G. Shafer and J. Pearl, editors. *Readings in uncertain reasoning.* Morgan Kauffman, San Mateo (CA), USA, 1990.

[37] K. Tanaka, T. Ikeda, and H. Wang. Robust stabilisation of a class of uncertain nonlinear systems via fuzzy control. *IEEE Trans. on Fuzzy Systems*, 4(1):1–13, 1996.

[38] M. Togai and H. Watanabe. Expert system on a chip: and engine for realtime approximate reasoning. *IEEE Expert*, 1(3), 1986.

[39] R.M. Tong, Beck M.B., and A. Latten. Fuzzy control of the activated sludege wastewater treatment process. *Automatica*, 16(6):695–701, 1980.

[40] E.A. Walker. Stone algebras, conditional events and three valued logics. *IEEE Trans. on Systems, Man & Cybernetics*, 24(12):1699–1707, 1994.

[41] L.-X. Wang. *Adaptive Fuzzy Systems and Control*. Prentice-Hall, Englewood Cliffs, NJ, 1994.

[42] L.X. Wang. *A Course in Fuzzy Systems and Control*. Prentice-Hall International, Inc., 1997.

[43] H. Watanabe. RISC approach to design of fuzzy processor architecture. In *IEEE Int. Conf. Fuzzy Systems*, pages 431–441, San Diego, CA, 1992.

[44] P. Werbos. Backpropagation through time: What it does and how to do it. *Proc. of IEEE*, 78(10):1550–1560, 1990.

[45] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

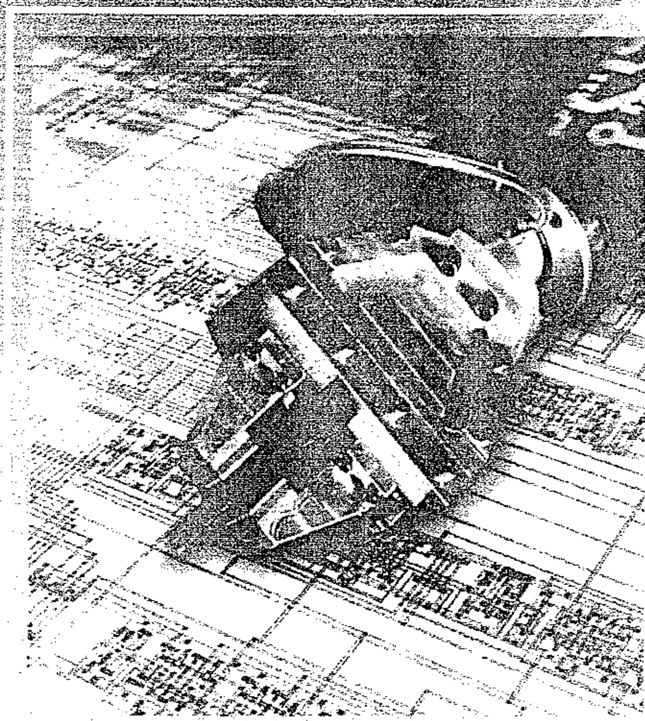
[46] L.A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. on Systems, Man & Cybernetics*, 3:28–44, 1973.

VIII Congreso Latinoamericano de Control Automático
XIII Congreso de la Asociación Chilena de Control Automático

ANALES

Volumen III

Editores:
Ricardo Pérez
Manuel Duarte
Gastón Lefranc



Organizado por la Asociación Chilena de Control Automático
Miembro de la International Federation of Automatic Control IFAC

ORGANIZACIÓN / ORGANIZATION

COMITÉ ORGANIZADOR / STEERING COMMITTEE

Gastón Lefranc, Presidente / Chairman
Manuel Duarte
Ricardo Pérez
Iván Solar
Fernando de Mayo
Ingeborg Mahla
Gonzalo Acuña
Felisa Córdova
Juan Hernández
Juan Herrera
Eligio Amthauer

Comité Internacional de Programa /

International Program Committee

Ricardo Pérez, Presidente / Chairman	(Chile)
Mario Fernández, Secretario / Secretary	(Chile)
Benjamín Kuchen	(Argentina)
Jonas Paiuk	(Argentina)
Freddy Naranjo	(Colombia)
Carlos Vélez	(Colombia)
Rafael Castro	(México)
Jaime Alvarez	(México)
Ricardo Tanscheit	(Brasil)
Oswaldo Luiz de Valle	(Brasil)

Comité Latinoamericano de Automática /

Latinoamerican Committee on Automatica

Manuel Duarte, Presidente / Chairman	(ACCA, Chile)
Jonas Paiuk	(AADECA; Argentina)
Marcio Rillo	(SBA, Brasil)
Freddy Naranjo	(ACA, Colombia)
Juan M. Ibarra	(AMCA, México)

INDICE / INDEX