Fuzzy Logic Control — A Taxonomy of Demonstrated Benefits

DAVID E. THOMAS AND BRIAN ARMSTRONG-HÉLOUVRY

The motivations for fuzzy logic control (FLC) are illuminated by exploring the benefits obtained by application designers through its use. A context for this exploration is set with a discussion of the characteristics of control policies and of the general attributes of FLC. Each benefit is described by reference to reported FLC implementations in which the benefit is demonstrated. Based on common features of the example applications, application preconditions for obtaining each benefit are stated.

I. Introduction

During its early development, control theory enjoyed "applications pull," in the sense that progress in applications led progress in theoretical development [4]. In a similar manner, FLC enjoys "applications pull" today; much of the interest in the field is stirred by the reported demonstrations of FLC. Correspondingly, a large portion of the FLC literature is addressed to the "how to" issues of specific implementations.

In this work it is not questions of "how to" which are addressed. The fundamentals of fuzzy set theory and FLC have been ably described elsewhere (e.g., [12], [41]). Rather, in this paper questions of "why to" and "when to" are addressed. Specifically:

- What benefits can a control system designer obtain through the use of FLC?
- What aspects must the application possess for the designer to obtain these benefits?

The primary research tool of this work has been close reading of FLC applications found in the control systems literature. By examining reported applications, one can gain insight into the benefits obtained by the designers. The method of close textual analysis, which is more commonly used in disciplines of the letters, is applied to clarify the "why" of FLC design [42]. We seek to extend the state of the art in FLC by offering several items which will support design decisions:

Manuscript received August 8, 1994; revised November 1, 1994.

The authors are with the Department of Electrical Engineering and Computer Science, The University of Wisconsin–Milwaukee, Milwaukee, WI 53201 USA.

IEEE Log Number 9407857.

1) A characterization of control policies.
2) A taxonomy of benefits demonstrated to be obtained through the use of FLC.
3) An enumeration of application preconditions required to obtain each benefit.

Before proceeding with the taxonomy of demonstrated benefits, groundwork must be laid in the form of definitions, presented in Section II, a scheme for characterizing control policies, presented in Section III, and a discussion of attributes of FLC, presented in Section IV. In Section V the taxonomy of demonstrated benefits is presented, along with a discussion of application preconditions; and finally, conclusions appear in Section VI.

II. Definitions

For convenience, definitions and acronyms for terms used in the paper are collected in this section. For readability, acronyms are introduced for many terms. Where these terms are spelled out in the text, they are capitalized, to show their relation to the acronym.

The benefits of FLC presented here have been identified by examining many articles and conference papers reporting applications of FLC. A primary question posed for each application was simply "What did the designer gain by the use of FLC?" In practice, this was a sifting process. As more applications were examined, common threads or ideas appeared. Some threads resolved into specific application benefits derived from the use of FLC. Common characteristics among applications sharing a demonstrated benefit reveal aspects of the application required to obtain the benefit. These are the "application preconditions." Common aspects in the way fuzzy logic delivers the benefits show important attributes of FLC; and commonalities among designer goals leads to a characterization of control policies, the subject of the next section.

III. A Characterization of Control Policies

To understand the benefits of FLC, it is important to consider the goals of the control system designer. To refer to the design goals implemented in a controller, we

0018–9219/95$04.00 © 1995 IEEE

Table 1 A Characterization of Control Policies Along Three Dimensions

Antecedent Space	the input space of the fuzzy inference engine. Sometimes referred to as the state space [19], but distinct from the state space of the system under control.
Locality of Control (LOC)	an attribute of FLC which permits the designer to specify control in local antecedent-space regions.
Interpolation Among Rules (IAR)	an attribute of FLC which permits the designer to control the rate of change of control effort as rule firing changes.
Information Equity (IE)	an attribute of FLC which permits the designer to apply linguistic or qualitative information which is often unused in traditional controller design.
Capture of Operator or Artisan Knowledge (COAK)	an FLC benefit derived from the use of human experience in controller design.
Local Adaptation (LA)	an FLC benefit derived from adaptive modification of control effort made on a local, antecedent-space basis.
Exception Handling (EH)	an FLC benefit descended from the use of context sensitive control policies.
Generalized Damping (GD)	an FLC benefit derived from the nonlinear use of velocity or other energy storage variables to modulate control effort.
Generalized Constraint Enforcement (GCE)	an FLC benefit obtained by the use of a time invariant control policy whose control effort increases quickly as the constraint is approached.
Input/Output maps via Interpolation (IOI)	a benefit stemming from the ability of an FLC to approximate a nonlinear transfer element using interpolation among rules.

introduce the term "control policy." A control policy (e.g. regulation or constraint enforcement) is distinct from a design methodology (e.g., root locus or μ synthesis). The control policy is the statement of objectives, the design methodology is the tool used to achieve those objectives. There is no generally accepted characterization of control policies. The characterization offered in Table 1 evolved during the consideration of many papers from the controls literature. In the proposed framework, control policies are characterized by three attributes: type, composition and temporal behavior.

I) Type:
- A. Regulation / Tracking
- B. Maximization / Minimization
- C. Constraint Enforcement

II) Composition:
- A. Simple
- B. Compound

III) Temporal Behavior:
- A. Static
- B. Dynamic

A. Type: Regulation, Maximization or Constraint Enforcement

There are three distinct types of control policy: regulator, maximizer and constraint enforcer. Regulators include controllers which determine control action as a function of output or state error and are designed to bring error to zero. Both tracking and fixed reference applications are included under regulation/tracking, and termed regulators.

In a maximization policy, control is selected to maximize a performance measure. Generalized predictive control can be constructed in this way [8], [9]. In economics and decision theory, many control policies are maximizers. In the control of dynamic systems, maximization policies often coexist with regulator policies, particularly when the plant has redundant degrees of freedom. A well developed example is the control of redundant manipulators [25]. An interesting source of redundancy for optimization is redundancy in the trajectory specification. This is exhibited in some of the examples of maximizer policies described below (e.g., [46] and [50]).

Examples from the theory of optimal control require special consideration for characterization. If the optimization is performed as part of the controller *design*, the controller is a regulator. That is to say that the design procedure used to produce the control law does not influence its characterization type. This characterization is supported by the fact that many frequency-domain designs can also be obtained as the result of a corresponding optimization problem. If the optimization is performed during *execution*, the controller is a maximizer. With the rising availability of computing power, real time optimization is becoming more common (e.g., [18]).

A constraint enforcement policy is used when violation of a condition exacts a high or catastrophic penalty. This may reflect either physical limits of the plant or sharp bounds on acceptable performance. Torque limits on high performance aircraft wings and emission restrictions applying to waste water treatment are two examples of applications which call for constraint enforcement control policies.

A constraint condition alone is generally insufficient to generate a unique control sequence. Rather, a statement of the desired regulatory or maximization behavior is needed, with constraints placing limits on behavior. For example, consider an automobile negotiating a curve. While knowledge of the maximum permissible tire sidewall stress may be useful in establishing a constraint, this information alone is insufficient to generate a sequence of steering angles which successfully guides the vehicle through the curve.

The three distinct types of policies are illustrated by a hypothetical automotive engine controller. Tracking behavior is apparent in following the sequence of power commands from the operator. A maximization policy uses additional degrees of freedom in the control variables of throttle angle, fuel injection and ignition timing to achieve good fuel economy with low exhaust emissions, while the importance of avoiding predetonation places a constraint on ignition timing. This example illustrates how multiple control policies could be called for simultaneously.

B. Composition: Simple or Compound Control Policies

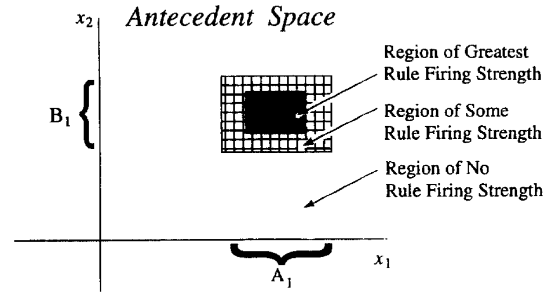
Control policies expressing just one goal are simple, and those reflecting multiple goals are compound. The property of being simple or compound is referred to as the "composition" of the policy.

The statement of a simple control policy has a single focus, such as "drive error to zero." Statements of compound control policies often contain subordinate or qualifying clauses, such as "drive error to zero without saturating the actuator" or "track the reference trajectory while maintaining rider comfort." Compound control policies are composed of simple regulatory, maximizing, and constraining policies with an explicit or implicit relative priority.

C. Temporal Behavior–Static or Dynamic Control Polices

Compound control policies may be either static or dynamic. In the static case, the set of active simple control policies does not change with time. An example might be the hypothetical automotive engine controller above. In the dynamic case, the set of active control policies changes over time, often in response to plant state and environmental conditions. Examples of dynamic control policies can be found in batch-type industrial processes, vehicle navigation, and automation.

When implemented in crisp control, dynamic control policies are often reflected by controllers which implement an abrupt switch between control laws, or multimode controllers. An example might be an environmental control system which implements one set of control policies, including energy conservation, during normal operation; and a completely different set of control policies in response to a fire. As with composition, the temporal behavior of a controller is a property of the control policy, which is a reflection of the goals of the control system designer.



Rule Antecedent: **IF** (x_1 is A_1) **AND** (x_2 is B_1)

Fig. 1. Local nature of a single FLC rule.

D. Characterizing Controllers

Using the characterization of control policies, control implementations may be classified along the three dimensions of type, composition and temporal behavior. For example, the Aircraft Roll Controller [6] discussed below seeks to drive the roll rate error to zero without exceeding torque limits. The controller is a regulator with constraints and thus compound. The goals expressed by the controller do not change with time, thus the policy is static, resulting in the description "compound static regulator with constraints." The hypothetical environmental control system above, with the addition of temperature setpoint tracking and constraints on ventilation, would have the most complex characterization: "a compound dynamic regulator/maximizer with constraints." In this characterization of control policies, a large percentage of classical and modern control theory is concerned with the implementation of simple, static regulators/trackers.

IV. Attributes of Fuzzy Logic Control

Whereas the structural components of crisp controllers are quantitative relationships, the structural components of an FLC are input and output linguistic variables, and linguistic rules. Three attributes which are important for the benefits discussed below arise from the interplay of these structural components:

1) Locality of Control (LOC)
2) Interpolation Among Rules (IAR)
3) Information Equity (IE).

A. Locality of Control (LOC)

The antecedent of an FLC rule, together with the input membership functions, specifies a bounded region, or a "patch," in the antecedent space of the FLC controller. Via output membership functions, the rule consequents specify the control action in that region. This is locality of control (LOC). Locality of Control is illustrated in Fig. 1, where

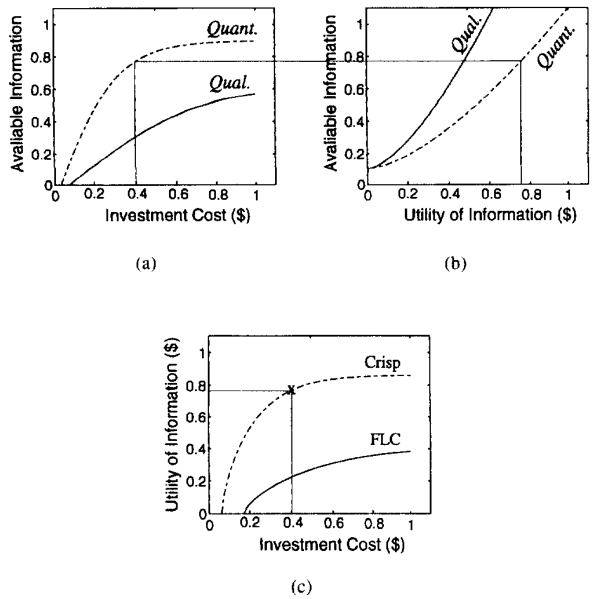


Fig. 2. Situation characterized by *quantitative* information, such as an inverted pendulum.

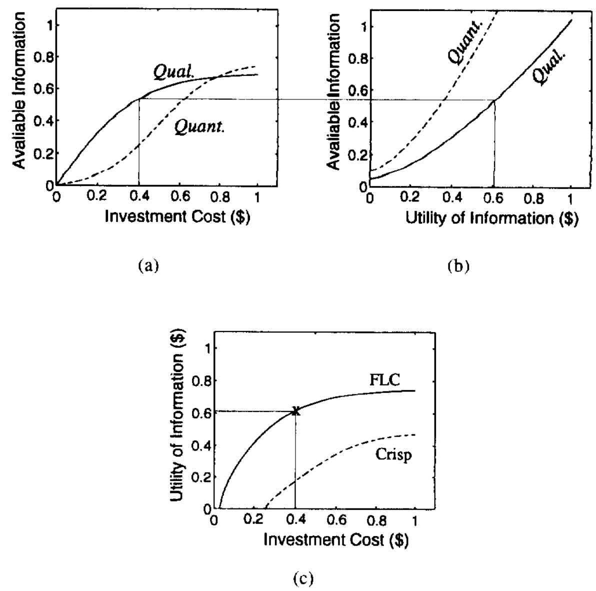


Fig. 3. Situation characterized by *qualitative* information, such as a cement kiln.

a rule contributes to the specification of control action in a bounded region of a two dimensional antecedent space. The LOC attribute allows the designer to tailor or tune the control response in a single region without changing the control action taken in other regions.

The local nature of control in FLC may be compared and contrasted with other control implementation techniques. Both neural networks and learning control exhibit a local nature. Given the appropriate training data, a neural network will reflect local features of a control surface or function [14]. Learning control [43] will reduce the error vector of a repetitive state trajectory given a plant model and a sufficient number of trials over the same trajectory. However, in learning control, corrections are made upon a finely grained sequence of errors uniformly spaced in time, rather than on identified regions or patches of controller input space. The technique of gain scheduling [2], that is, constructing linear or nonlinear controllers for given controller input-space regions and switching among these controllers, also has the flavor of LOC, embodying the notion of local or regional control. By contrast, the traditional PD or PID controller casts a control law in a global form. Any adjustment to the PD controller has a global effect on the control surface.

B. Interpolation Among Rules (IAR)

The FLC attribute of Interpolation Among Rules (IAR) combines the contributions of active rules, allowing the designer to bound the rate of change of control with respect to state. This interpolative attribute of FLC makes possible a graded transition from one control action to the next as the plant traverses state-space regions.

The graded transition of output stands in direct contrast to the "Boolean if" statement. The "switch-on/switch-off" nature of the "Boolean if" statement introduces a discontinuity in control action. The relay is a physical counterpart of the "Boolean if." [13] has demonstrated the undesirable behaviors of chattering and limit cycling brought on by the use of bivalent control. By interpolating, an appropriately constructed FLC can implement the rule based character of an expert system, while avoiding the discontinuity of the "Boolean if."

C. Information Equity (IE)

Unlike software, telecommunications or construction engineering, the field of controls has no systematic methodology for cost estimation. But as with any field, cost plays an important role in the selection of tools and the successful outcome of a controls engineering project.

One view of FLC is as an implementation tool suited to constructing a controller based on qualitative or linguistic information [1]. In this view, it is important that all information has cost and all information has value. The cost of needed information can very widely. Information that has been previously developed represents the least direct cost to a project; whereas new information that must be developed—by analysis or experiment—may come at considerable direct cost.

This discussion of the cost of information and the curves of Figs. 2 and 3 are necessarily subjective; there is no quantitative metric of controls-application information. The subjective notion of *"quantity of information"* is introduced to be able to discuss the availability and cost of different kinds of information, and the impact of available information on controlled-system performance.

Controls-application information might include system models, model parameters, details of nonlinearities, information about possible process upsets, information concerning the thresholds and costs of violating process constraints, knowledge of how to respond to unusual plant conditions, etc. Information Equity (IE) is the difference between the value of information, that is, its utility in the controls application, and the cost of that information. Information Equity represents the net value of the information.

For the purposes of this discussion, information is taken to come in two broad categories: quantitative and qualitative; and control is implemented in one of two broad ways: Crisp and FLC. For some control applications, such as an inverted pendulum, an analytic model is readily available, and thus considerable quantitative information may be had at low cost. This situation is reflected in Fig. 2. In plot 2(A), the quantity of available information is plotted as a function of invested resources. The quantitative curve rises above the qualitative curve in this case. The presence of the qualitative curve indicates that it would be possible to collect qualitative information; in the example of the inverted pendulum, one might interrogate a skilled juggler. But the relative position of the curves in plot 2(A) indicates that systems characterized by quantitative information yield less qualitative information for a given investment.

Plot 2(B) shows the value or use of information, also for a situation characterized by quantitative information. The amount of information required to achieve a certain level of performance may be less for the Crisp controller than for the FLC. The inverted pendulum is again an example, where two numbers describing the plant linearization about the equilibrium may be sufficient for control.

Fig. 3 presents a situation characterized by qualitative information. The cement kiln application of Holmblad and Ostergaard [17] is one example. In addition to the challenges of obtaining a quantitative model for this and many industrial processes, considerable qualitative knowledge may be readily available in the form of designer knowledge, operating manuals or experienced operators. Each of these resources may represent many man years of investment; but the cost of developing manuals or experience is external to the control design project. So in such a case, the control design project related investment per unit available information can be less for qualitative than quantitative information, as indicated in plot 3(A). Abramovitch [1] describes several control applications characterized by qualitative information.

Plot 3(B) again shows the value or use of information, this time for a situation characterized by qualitative information. In this case the amount of information required to achieve a certain level of performance may be less for the FLC than for the crisp controller. This may be the case when, for example, velocity-dependent damping is desired for a low order but not easily modeled system. Available designer knowledge may be adequate to achieve the desired level of performance (see, for example, [6]).

In plots 2(C) and 3(C), the investment cost and utility of information are brought together in one plot. This is done by projecting the Available Information for a certain investment onto the Utility of Information for that method, as shown for one point in each of Figs. 2 and 3. The resultant point is indicated by an "x" in Figs. 2(c) and 3(c). The height of the curve in plot (C) indicates the Information Equity: the value of the information in excess of what it cost to acquire. Information Equity can be improved either by increasing the Utility of Information or by reducing its Investment Cost.

There is no *a priori* guarantee that a situation with high availability of qualitative information, e.g., plot 3(A), will correspond to a situation in which qualitative information yields more utility, e.g., plot 3(B). Plots 2(A) and 2(B) could, for example, join to describe a particular application. It is the plots of Utility versus Investment, 2(C) and 3(C), which indicated whether Crisp or FLC will achieve greater Information Equity in a given situation.

A collection of examples that highlight the effective use of qualitative and quantitative information (maximizing information equity) employ the fuzzy partitioned linear control of Sugeno and his coworkers [36], [38], [39]. For systems for which neither linguistically expressed nor functionally expressed control strategies existed, the authors demonstrated control by first partitioning the input space of the system into operating regions, collecting input-output data from each partition and performing linear system identification. With the identified model, linear optimal control was designed for each partition. The final controller combines functionally expressed linear control laws for each partition with fuzzy inference, which determines the current operating partition or partitions and blends the outputs of the individual linear controllers. Functionally expressing the linear control laws maximizes the value of the quantitative input-output data; while linguistically expressing the partitioning maximizes the value of available designer knowledge regarding the operating regimes of the system.

At this time, plots of available information and utility of information are necessarily subjective. Figs. 2 and 3 are meant to illustrate the point that the role played by information in determining performance versus cost is a consequence of two processes: the cost of acquiring the information, and the ability to use the information. The importance of Information Equity for FLC lies in the fact that FLC is an effective method for achieving utility from qualitative information. In the next section, it is seen that several of the demonstrated benefits depend on increasing the equity of qualitative information.

V. Demonstrated Benefits And Application Preconditions.

The taxonomy includes six benefits of FLC:

1) Capture Operator or Artisan Knowledge (COAK)
2) Exception Handling (EH)
3) Generalized Damping (GD)

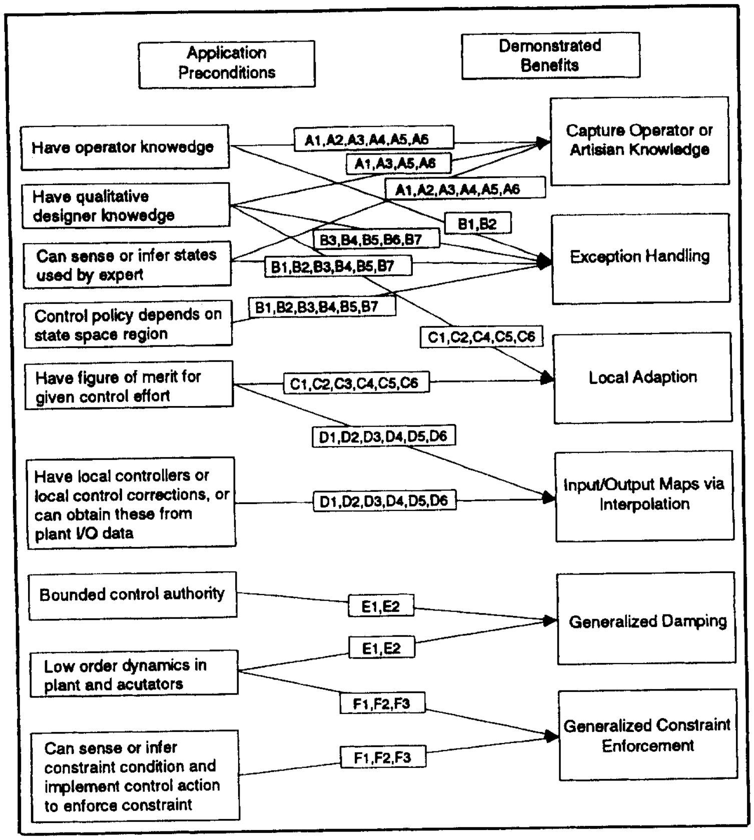


Fig. 4. Control application preconditions mapped to FLC benefits.

4) Local Adaptation (LA)
5) Generalized Constraint Enforcement (GCE)
6) Input/Output Maps via Interpolation (IOI).

Each benefit and the associated application preconditions will be described by reference to papers in which they are demonstrated. The cited papers are the foundation upon which the taxonomy is built: they demonstrate the benefit; and from them the case is made that the associated application preconditions are important for the benefit to be obtained. Every effort has been made to select papers which demonstrate each benefit to a significant degree and which make comparisons that will be credible to the controls engineer.

The taxonomy is neither complete nor exclusive. Completeness, an assurance that there are no unlisted benefits, would require a complete list of controls applications. The characterization of control policies described in Section III is a step in this direction, but it is unlikely that a complete list will ever exist. As with a taxonomical classification in the natural sciences, this taxonomy will grow and evolve as new ideas and applications emerge.

For the benefits to be exclusive to FLC would mean that they could not be obtained in any other way, and this is not the case. Indeed, if gain scheduling is defined sufficiently broadly and some mild smoothness conditions are imposed, any device that maps sensed signals and internal state to actuation is a gain scheduler. Control design methodologies are often compared on the basis of the range of systems for which the design method is assured to yield a stabilizing controller. On this basis, FLC does not show a substantial benefit relative to traditional control methods. But implementation is another issue. In the absence of any costing methodology for controls applications, a rigorous case for an implementation method is difficult to make. A circumstantial case, however, can be made by examples. The examples cited below present implementations of control policies that are not customarily implemented in other ways. Several of the examples include comparisons with production controllers that are crisp and are the result of significant design efforts (e.g., [6], [10], [21], [35]).

In Fig. 4, links are shown between demonstrated benefits and associated application preconditions. The identifiers which appear on each link refer to the application papers listed in Table 2. Fig. 4 has value for design: knowing the characteristics of a proposed application, the control designer may use the figure to establish which of the preconditions describe the problem at hand, and thus which of the benefits can likely be obtained.

Table 2 Paper from the Application Literature Cited in Fig. 4

Link ID	Application	Reference
A1	5MW Nuclear Reactor	[5]
A2, B1	Cement Kiln	[17]
A3	Sake Brewing	[27]
A4, B2	Activated Sludge Wastewater Plant	[44]
A5, B3	Model Car in Extreme Situations	[46]
A6	Water Purification Plant	[48]
B4	Automotive Automatic Transmission	[31]
B5, F1	Mobile Robot	[30]
B6	Aircraft Carrier Landing System	[35]
B7	Container Crane	[49]
C1	Cargo Ship Steering	[20]
C2	Automotive Anti-Skid Braking	[21]
C3	Attitude Control of a Flexible Satellite	[10]
C4	Autonomous Vehicle Controller	[16]
C5	Robot Arm Controller	[32]
C6	Nonlinear Channel Equalization	[47]
D1	Multilayer Waste Incinerator	[36]
D2	Nuclear Reactor	[29]
D3	Model Car Parking	[37]
D4	Model Car Controller	[38]
D5	Automotive Engine Idle Control	[45]
D6	Environmental Emissions	[40]
E1, F2	Flexible Wing Roll Control	[6]
E2	Pitch Control of an Interceptor Missile	[7]
F3	Automatic Train Operation	[50]

A. Capture Operator or Artisan Knowledge (COAK)

In process, vehicle and other control applications, a skilled operator may be available who is able to perform adequate or superlative control. If these operators are able to articulate their expertise, FLC is a convenient method for instantiating that knowledge. Capturing this information and directly using it in an FLC yields the COAK benefit, and is demonstrated in [17], [27], [44], and [5].

Likewise, control system designers climb learning curves in attempting to control plants. This viewpoint combines both "designer as engineer," attempting to use first principle, derived knowledge and crisp quantitative information in plant control, and "designer as artisan," using qualitative information and accumulated experience. Among others, [46], [48], and [27] demonstrate this mode of the COAK benefit.

The operator knowledge approach is demonstrated in [17], in which a cement kiln is automated through application of rules found in kiln operator training manuals. The Cement Kiln Fuzzy Logic Controller acts as a compound dynamic regulator. In normal operation, the plant is regulated about an operating point, while provision is made to return the plant to normal operation if the drive torque of the kiln rotation motor displays large oscillations.

Bernard [5] used extensive operator surveys and interviews to obtain the operating rules for a 5MW nuclear power plant. The resulting FLC implemented a static compound regulator, in which power level was driven towards a target, subject to conditions imposed on the use of control rods.

Tong *et al.* [44] and Oishi *et al.* [27] employ rule-bases derived from expert knowledge to control a Sludge Wastewater Treatment plant and Sake Brewing respectively. The Sludge Plant implements a dynamic compound regulator/maximizer with constraints, seeking inner loop setpoints for aeration, inflow and recycle flows to hold effluent oxygen demand and suspended solids within acceptable limits in the face of process upsets. The Sake Brewing control incorporates knowledge of the specific, time-varying needs of the sake brewing process.

These applications—the Sludge Plant, the Sake Brewing operation, the 5MW Nuclear Reactor, and the Cement Kiln—exhibit nonlinear, time varying behavior which had made effective automatic control difficult. All these plants were well controlled by the associated FLC. Of the four, the Sludge Plant is the only pure simulation, the rest have been physically realized.

The designer knowledge approach is typified in [46], in which a model car is controlled on an obstacle course, and to a lesser extent in [48], in which the feed rate of precipitating agents in a waste water treatment plant is regulated. The Model Car controller is a compound static maximizer with constraints, implementing a control policy which requires the car to go as fast as possible without striking an obstacle or entering skidding or sliding. Using everyday driving knowledge of the form "turn hard left (right) if too close to the right (left) wall and increase speed if the forward direction is clear," the Model Car control system designers were able to construct a working controller in short order. Here the designer is seen as artisan, applying his own experience to the control problem.

In the waste water plant [48], the designers obtained a statistical regression relationship between rainfall and water turbidity, and used the regression to chose a candidate feed rate of precipitate agent. An FLC constructed with operator knowledge about measured and observed plant states fine tunes the feed rate to reduce water turbidity. This controller implements a simple static maximizer with constraints to control the precipitate feed rate in response to measurements of water alkalinity and water temperature. The operator's observations about the turbidity condition of inlet water, the precipitated solids within the plant, and the condition of outlet water are also used as inputs to the FLC. The controller chooses precipitate feed rates which are high enough to satisfy the constraint of low turbidity at the outlet without excessive use of expensive precipitating agents, thus optimizing the feed rate.

The sake brewing application described by Oishi [27] is an interesting blend of operator and designer knowledge. The process begins with a prepared rice mixture in a large (3000 l) vat; the resulting rice liquor is brewed over a 15 day period. The vat temperature is well regulated by a classical PID controller. The supervisory control problem is to chose an appropriate state versus time trajectory in the form of a temperature setpoint sequence for a plant with irreproducible initial conditions. The designers developed a crisp, quantitative kinetic model for the specific gravity and alcohol content of the liquor in terms of various enzyme activation energies and vat temperature differences. Prior attempts at optimal control using precomputed temperature trajectories resulted in low quality product.

The designers realized the COAK benefit by using the experience of several sake brewmasters or *toji*. This experience had the simple form of "avoid cooling the mixture when alcohol content is low, and avoid heating the mixture when alcohol content is high." By expressing this principle as a graded transition among rulebases characterized by *low, medium and high* alcohol content, and by using the quantitative models to compute error and rate-of-change of error between the measured and predicted specific gravity, the designers realized a FLC controller which closely mimicked the control of an expert *toji*. The FLC component of this application was a simple static regulator, using measured specific gravity and alcohol to generate a temperature setpoint trajectory which reflected the judgment of the *toji*. The designers' success in controlling the sake brewing process is due both to their knowledge of sake vat chemistry and to the use of a control technique which accommodated the qualitative knowledge of experienced *toji*.

Not every application which demonstrates COAK is employed in a supervisory mode. Operator experience is used to generate direct control action in the Model Car [46]. In this case, linguistic rules relating plant states to control actions are directly represented in the FLC membership functions and rulebases.

Note that each of the applications demonstrating COAK implements a control strategy derived from articulated human experience. In each case, the FLC instantiated control rules first expressed in linguistic form; no plant identification or on-line adaptation was used. The states used by experts to derive control effort were either inputs to the FLC or inferred by the FLC in forming a control effort. Thus two preconditions for obtaining the COAK benefit may be stated:

1) The control system designer must have access to expert, articulated human knowledge, either that of an operator or that of a designer.
2) The plant states used by the expert in forming a control judgment must be directly sensed or indirectly inferred.

A counter-example to the COAK benefit in FLC is the "depth of anesthesia" metric used by anesthesiologists during surgical procedures. Depth of anesthesia is a subjective, state-dependent observation made by the attending physician, not a measurement. There is little consensus in the expert community for a clearly reliable indication of this patient condition. There are several clinical indications (blood pressure, CO2 concentration, heart rate, pupil dilation, electrocardiogram (EKG), electroencephalogram (EEG) and electromyographic (EMG) measurements), but all vary with patient condition and the anesthetic agent used. Thus in the recirculating anesthetic agent FLC reported by [24], the plant state used by the expert is not available to the FLC. Rather, the controller strives to maintain the patient's mean arterial blood pressure at a setpoint. While the controller has performed well in several clinical trials, the gamut of the attending physician's experience is not captured in the controller's design.

In providing the COAK benefit, FLC allows the designer to exploit qualitative information which may be difficult to use in a traditional control design. While the Locality of Control and the Interpolation Among Rules attributes contribute to the construction of the FLC, it is the act of raising the value of artisan information, of increasing the Information Equity of the application, which delivers the COAK benefit.

B. Exception Handling (EH)

An exception is a change in plant operation or environment which requires a change in control policy. Exception Handling is thus associated with dynamic control policies. The Cement Kiln Controller of Holmblad and Ostergaard [17] provides an example. Ordinarily, material in the rotating kiln is processed in an approximately steady flow. But under some conditions, the material aggregates into cakes, which ride up the side of the kiln and break off in large pieces. This condition is detected as large oscillations in motor torque and can be corrected by modifying the coal feed rate. In the FLC of Holmblad and Ostergaard [17], two distinct control policies are present, but not simultaneously active: a normal operating conditions policy, and a policy designed to return to normal operations when the torque oscillation is detected.

The Activated Sludge FLC reported in [44] uses two exception handling policies to account for process upsets. These upsets, called rising sludge and bulking sludge, interfere with the goal of maintaining effluent conditions within acceptable limits. This FLC generates supervisory control, deciding what incremental change to apply to the setpoints of local-loop PID controllers to return the plant to normal operating conditions. An comparable application described by Terano *et al.* [41] uses an exception handling control policy to implement a rainwater pump management system that maximizes pump service life during normal operation and transitions to a "safety first" policy during heavy rains.

While EH may at first seem limited to supervisory control, several applications – the Model Car [46], the Mobile Robot [30] and the Automatic Carrier Landing System [35] – use Exception Handling to generate control signals directly. In these cases, the control policy is tailored to the state-space region which is characterized by the exception, advancing a regional goal rather than simply attempting to return to normal operation.

The Model Car application [46] well demonstrates the Exception Handling benefit. The car uses infrared wheel rotation sensors and distinct rulebases to infer sliding and skidding conditions, together with position information from ultrasonic sensors. The FLC uses a second set of rulebases to generate steering angle, motor current and brake force commands. The Model Car FLC implements a control policy which is predicated on inferred sliding and skidding conditions. The dynamic aspect of the control policy is seen in the use of hierarchical rulebases. When a slide or skid arises, control emphasis is shifted from achieving high forward speed to recovery from the exception condition.

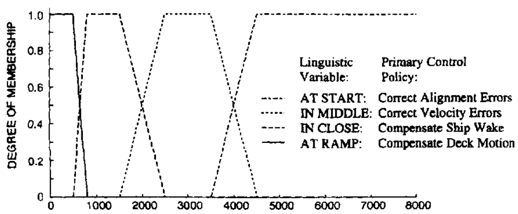


Fig. 5. Control policies expressed by graded transitions among rule bases. (Adapted from [35]).

Likewise, the Mobile Robot application [30], employs a strategic "reach the designated target position" control policy, and switches to a tactical "avoid collisions" policy when a previously undetected or randomly placed barrier is encountered. The pattern of rule firing in the FLC rulebases changes sharply as avoiding the collision assumes paramount importance. After the barrier is successfully navigated, the strategic policy reasserts its dominance and the robot continues to seek the target position.

The Carrier Landing application simulated by Stienberg [35] uses Exception Handling to perform a graded transition between rulebases as an aircraft approaches for carrier landing. The FLC is a compound dynamic regulator. As illustrated in Fig. 5, the control policy changes as the aircraft approaches the deck.

Six rulebases are implemented in the Carrier Landing application; all use distance to carrier deck in rule antecedents. The roll, sink rate and throttle rulebases grade to zero as the range to carrier decreases. The lineup and glide slope rulebases are concerned with correcting large errors quickly when range is large or "at start." When the aircraft is halfway through the approach, trends in drift rate are more important than small errors. Thus when the range is at a linguistic value of "at middle," the lineup and glide slope rulebases attempt to cancel drift in aircraft lineup. When the range is small, static errors become important, and the lineup and glide slope rulebases apply corrective action, attempting to level the aircraft and avoid striking the carrier deck. The sixth rulebase is intended to deal with the air disturbance over the carrier wake and with carrier deck motion.

Thus the control policy for the Carrier Landing application changes as the range to the deck declines. This is very much like human pilot behavior, responding to changes in the aircraft's state in a context sensitive manner. Unlike the "return to normal" character of exception control policies in the Cement Kiln or navigation applications, in a given landing attempt, the aircraft does not revisit a range region.

The Automobile Automatic Transmission application reported in [31] demonstrates Exception Handling based on driver's intent inferred via a fuzzy rulebase using recent and current brake and throttle positions. With an ordinary automatic transmission controller, frequent shifting can arise when the system is operating just at the boundary between one gear and the next. In the FLC, the inferred driver's intent is used to select between shift policies suitable for sustained acceleration or sustained deceleration. In this way frequent shifting is avoided.

A Container Crane application [50] also uses several domains with distinct rulebases to control the wire rope and trolley motion of an overhead crane moving cargo from ship to wharf. The transition between adjacent domains is explicitly dependent on the carrying time of the load. The domains themselves characterize different regions in the trajectory of the cargo, corresponding to starting, accelerating, traversing with constant speed and stopping. Within a domain, different rules for trolley speed commands and rope length changes are used. This FLC, like the Carrier Landing System and Automobile Transmission, implements a sheaf of control policies, forming a compound dynamic regulator.

These examples demonstrate the EH benefit in two distinct forms. The first has a strategic control policy, which is dominant in a state-space region of normal operation. When the system goes out of this region, FLC allows the designer to implement a second policy, which is designed to return the system to normal operation. The Sludge Process, Model Car, and Mobile Robot demonstrate behavior of this sort. By contrast, the Carrier Landing, Container Crane, and Automatic Transmission applications traverse several different regions characterized by a plant state. Within each region, a different control policy dominates. The FLC controls the transition between adjacent regions, strengthening one rulebase at the expense of another.

The common thread between these two modes of the Exception Handling benefit is the temporal nature of the control policy. In implementing a dynamic control policy, both modes withdraw an inapplicable control policy and introduce a policy more applicable to the current context of the plant. In the Exception Handling benefit, the Locality of Control and Interpolation Among Rules attributes allow the designer to engineer a graded transition among rulebases. To obtain the this benefit, several prerequisites must be met:

1) The FLC must be able to detect or infer the exception condition.
2) The FLC must implement a control policy to cope with each exception.
3) The transition between adjacent control policy regions must be engineered by the designer.

The last point may pose the greatest challenge to realizing the EH benefit. The transition between control policies may introduce an abrupt change in control effort, which is effectively a high-gain nonlinearity. Depending on system order and transport lag, the high-gain nonlinearity may lead to limit cycling comparable to that possible with on/off control. As seen in the examples, Exception Handling is most easily implemented in first order systems. With respect to this issue, the "return to normal operation" Exception Handler faces a greater challenge than a "region traversal" EH, because the former can limit cycle across the boundary of the exception condition, whereas the later makes a one-way progress from the region of one control policy to the region of the next.

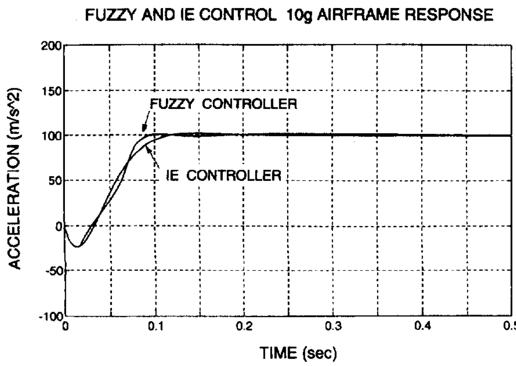


Fig. 6. Step response of the Hypervelocity Interceptor Pitch Controller (Adapted from [7]).

C. Generalized Damping (GD)

The generalized damping (GD) benefit of FLC arises from a desire for fast system response when the setpoint error is large. The damping effect of error derivative control is blocked while the system error is large, and a progressively greater damping effect is introduced as the system approaches a zero error condition. Such an FLC behaves in a manner similar to time optimal "bang-bang" control when the error is large and the damping effect small, using the full control authority to quickly drive the system towards the goal of zero error. This may be contrasted with the rise time/overshoot/control effort trade-offs seen in linear control, which is known to be far from time optimal when control authority is bounded [22].

While bang-bang control may be time optimal, its use in practice is limited. The principle barrier is the requirement of a precise model to construct the exact switching surfaces. If the applied control does not achieve precisely the modeled acceleration, chattering, overshoot and undershoot may occur [15].

The construction of an FLC enjoying the Generalized Damping benefit answers some of the objections to bang-bang control. By aggressively attacking a large error with large control commands and moderating these control commands by increasing the weight of derivative control as the error becomes small, the FLC behaves like the modified bang-bang controllers described by Gibson [15] and Sharuz *et al.* [33]. The interpolative nature of FLC permits a smooth transition between the bang-bang and PD regions.

Two applications which demonstrate the GD benefit are the Flexible Wing Roll Controller [6] and a Hypervelocity Interceptor Pitch Controller [7]. Both applications employ a static regulator implementing a nonlinear control policy to reduce system rise time in response to setpoint error. In the Flexible Wing Roll Controller, role rate is controlled by two rulebases, one contributing control predominately determined by error and the other predominately determined by error rate. The GD benefit is derived by the use in the error-rate rulebase of rules predicated on a near-zero-error condition. The error-rate rulebase applies damping when error is near zero, and serves to avoid overshoot. The airframe model used in simulation is first order, relating roll rate and a generalized control surface deflection to the angular acceleration.

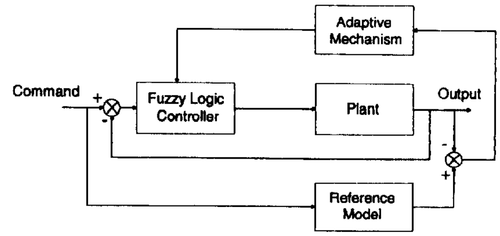


Fig. 7. Direct adaptive FLC block diagram.

The interceptor pitch plane control application found in [7] uses body acceleration error and pitch rate as inputs to an FLC generating a thruster control signal. The primary FLC rulebase approximates a crisp Sliding Mode Controller [34]; a second set of FLC rules with a near-zero-error predicate operates in parallel to moderate the control effort as the plant approaches the desired acceleration. The model used in simulation was second order, coupling angle of attack and pitch rate in two equations expressing the time rates of change of these plant states. The thruster was modeled as having fast first order dynamics. The step responses of the Hypervelocity Interceptor with "Integral Error" control and the proposed FLC are shown in Fig. 7. The two responses are comparable in the first portion of the move, but as the exponential decay of the linear controller becomes evident, the fuzzy controller converges more rapidly to the goal.

Both applications show a connection between FLC's enjoying the GD benefit and time optimal bang-bang control. It is important that in both cases the system being controlled was either first order or dominated by a first order mode, so overshoot was not substantial. If the plant or actuators had had higher order dynamics or significant time lag, the GD benefit might have been much harder to achieve. Two preconditions for the GD benefit may be stated:

1) Control authority is bounded.
2) The system is dominated by a first order response and transport lags are negligible.

The Locality of Control and Interpolation Among Rules attributes of FLC as applied in these applications delivers the Generalized Damping benefit.

D. Local Adaptation (LA)

Local Adaptation (LA) is an FLC benefit which exploits the Locality of Control attribute to adjust control action in local regions of antecedent space. Adaptive control is a rich area of study, reflecting the difficulty in constructing controllers which tolerate uncertainty. While it is difficult to generalize about a diverse field, some useful observations are available. Narendra and Annaswamy [26] point out that adaptive control is concerned with two fundamental aspects of control in the face of incomplete plant characterization. The first is that of identification, in which we seek to determine the characteristics and state of the plant under control. The second aspect is the challenge of determining control.

The issues of identification and control appear in an adaptive FLC as the distinct but related problems of credit assessment and credit assignment. The credit assessment problem is that of determining how much controller adjustment is needed. The credit assignment problem is that of determining which controller component or components should be adjusted. Several adaptive schemes are used in the FLC applications described here. Consequent membership functions are adjusted in a Cargo Ship Steering application [20], in a Robot Arm controller [32], an Anti-Skid Braking application [21], in an Autonomous Vehicle Controller [16], and in an adaptive filtering application [47]. A Satellite Attitude Controller [10] implements an adjustment scheme through the addition and deletion of rules, following the general outline of the Self Organizing Controller (SOC) [28].

Since the control policy expressed by these controllers is concerned with driving system error to zero, these FLC's may be viewed as simple static regulators. The antecedent of an FLC rule can be viewed as defining a patch or region of antecedent space over which the rule applies. Adapting the size, shape and location of the region, or the control effort specified in the region effects control in the region defined by the rule antecedent, but not in other regions.

The Local Adaptation benefit of FLC may be directly contrasted with the global nature of adaptation in a traditional adaptive controller, for example, the MRAC [2]. In an MRAC construction, a vector of parameters in the controller is adjusted via an adaptive law using the error between the plant and a reference model. If there exists a set of parameter values which brings the closed loop system into agreement with the reference model, the MRAC will, under suitable conditions, converge to these desired parameter values. But the existence of such a set of parameter values requires that the system be linear and perhaps of known order [2]. If the system is nonlinear, the optimal tuning in one region may be different from the optimal tuning in another region, and the adaptive action of tuning the controller for the current state may detune the controller for states previously encountered. This is in sharp contrast to Local Adaptation in a FLC, wherein the adjusted rulebase outputs act as a kind of memory, retaining the adaptation the controller achieved during past traversals of a given antecedent-space region.

The adaptive FLC block diagram in Fig. 7 illustrates the construction used by the majority of the applications discussed here. This is a direct adaptive control construction, akin to the crisp MRAC construction. The credit assessment problem is solved by comparing the output of the combined controller/plant system to a reference model. The error and rate-of-change of error derived from this comparison is evaluated, a controller correction is generated, and this correction is applied to the controller. The method of generating and applying this correction, that is, the adaptive mechanism, is the chief distinction among the Adaptive FLC's discussed here.

The Cargo Ship Steering [20] and Anti-Skid Braking [21] applications introduce an adaptive controller design technique called fuzzy model reference learning control (FMRLC). A fuzzy inverse model maps model reference error into changes in plant inputs, which are used to generate adjustments to the rulebase consequents. Only the active rules participate in the adjustment mechanism. By independently adjusting rule consequents, only local regions of the FLC input/output map are adjusted.

Wang and Mendel [47] present a recursive least squares (RLS) based algorithm for locally adapting a fuzzy inference engine to the filtering problem of nonlinear channel equalization. By constructing the fuzzy inference such that the output is linear in the consequent membership functions, they are able to employ a standard RLS algorithm. Although the application is outside the field of controls, it is included here because of the rigorous justification presented by the authors and the exploitation of Locality of Control.

The Robot Arm application [32] and the Autonomous Vehicle controller [16] do not employ an explicit reference model. These applications use a performance evaluation component or "performance table" derived from the Self Organizing Controller (SOC) suggested by Procyk and Mamdani [28]. Both applications follow the same general outline, beginning with a fixed error/change-in-error rulebase. The current error and change-in-error are inputs to the controller rulebase, which generates a control effort via the usual FLC mechanism. The performance table is also indexed using the error and change-in-error, and generates a change in control effort. This change in control effort is applied to the rules which fired m steps in the past. In this technique, like that of Layne *et al.*, the number of rules and the region of each rule in antecedent space are fixed. However, the output of each rule is adjusted independently, using the performance table as a guide. Unlike the SOC, no explicit inverse model is used to distribute control effort corrections among the rules.

A Satellite Positioning Controller [10] uses a Self Organizing Controller approach to generate the rulebase to control the attitude of a simulated satellite along three orthogonal axes. The actuators in this application are orthogonally mounted rotating wheels; the torque to these wheels is the control input to the plant. The physical structure of the satellite, namely solar panel placement and structural asymmetry, introduces dynamic coupling among the actuators. The attitude error and change-in-error of the plant along each axis index into the performance table, generating a measure of the required change in control effort. This measure is propagated to the rulebase via a quantitatively derived inverse model. Rulebase adjustment uses the performance measure to create new rules and delete others. Here the fuzzy sets which characterize antecedent error and change-in-error are left unchanged, while the rule consequents and the rules themselves are adjusted to achieve local adaptation.

Table 3 Control Policies for the Mobile Robot Application [30]

Policy	Type
"go forward to goal"	Regulator
"avoid obstacles"	Constraint enforcement (on proximity)
"avoid collisions"	Exception Handler

An adaptive control example which does not employ Local Adaptation is the Water Tank Temperature Controller of Daugherity *et al.* [11]. This FLC uses an error/change-in-error construction for the temperature control of a water tank with flow. Using measures of rise time, oscillation and overshoot, this controller adjusts the scaling factors used to map crisp process measurements into fuzzy sets. These scale factors operate across the range of the inputs, modifying the degree of fulfillment of every rule antecedent. While the controller succeeds in regulating the tank temperature, the FLC benefit of Local Adaptation is not employed.

Relying on the Locality of Control attribute of FLC, Local Adaptation allows the designer to ensure that adaptation on active rules affects only the antecedent-space regions covered by those rules. Two application prerequisites may be stated for the Local Adaptation benefit:

1) An appropriate inverse model, fuzzy or crisp, must be available to map errors between the reference model and the plant into modifications of fuzzy rules or consequents.
2) The rulebase modification mechanism must have a local effect in antecedent space.

E. Generalized Constraint Enforcement (GCE)

Generalized Constraint Enforcement (GCE) is related to Exception Handling. But whereas Exception Handling is employed when there is a need to change among control policies, GCE reflects a time invariant control policy.

Consider the Mobile Robot application [30] mentioned in the Exception Handling discussion above. The application actually implements three control policies (see Table 3).

The distinction between the Constraint Enforcement and Exception Handling is this: the constraint enforcement policy is always present with the regulator, and they may compete while producing the interpolated control action. The Exception Handling policy, "avoid collisions" is not interpolated. When a collision condition is approached, the "go forward" policy is deactivated. Thus the controller may exhibit "go forward while avoiding obstacles," but does not exhibit "avoid collision and go forward to goal."

Generalized Constraint Enforcement permits introduction of graded enforcement of constraints, but at the same time allows a system to behave as if the constraint did not exist when operating in state-space regions removed from the constraint. For example, the designer may elect to use a "bathtub" shaped cost curve, markedly different from the "quadratic bowl" cost curve which corresponds to a linear

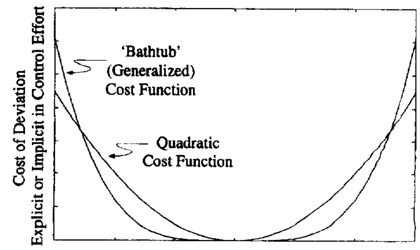


Fig. 8. Representative cost functions.

quadratic regulator (LQR). Fig. 8 provides representative cost curves for both situations. In a controller exploiting GCE, as the system state approaches the constraint, the constraint is weighted more heavily.

The flexible wing roll controller [6], discussed above in the context of Generalized Damping, also demonstrates the Generalized Constraint Enforcement benefit. Torque on the wing spars has limits which must not be exceeded. In this case, the FLC uses the current wing torque, the current deflection command and the rate-of-change of the deflection command to infer a correction to the deflection command. The firing strength of the correction rulebase increases rapidly as wing torque approaches the limit. However, when torque is not near the limit, no correction is generated. Unlike controllers designed via quadratic regulator theory, GCE exacts no performance penalty in state-space regions removed from the constraint. The FLC in the Flexible Wing application exploits the fact that is not necessary to minimize wing loading, but only to keep wing torque within suitable bounds.

In the Automatic Train control of the Sendai subway [50], secondary control objectives, such as rider comfort and energy conservation, are reflected as soft constraints on the regulator control policy that tracks the desired trajectory. The cost of violating the constraints, along with the cost of violating a safety related hard constraint, is weighed in an optimization step against the cost of deviating from the desired trajectory. The optimization step is comparable to that in Generalized Predictive Control [8], [9]. The control for normal cruising is thus a static compound regulator/maximizer with constraints. The capture of designer knowledge played an important role in the successful implementation of this system.

Controllers enjoying the GCE benefit have qualifying clauses in their control policies. Thus a controller implementing a policy of "reach the goal while avoiding sensed obstacles," as seen in the Mobil Robot, is characterized as a compound dynamic regulator with constraints. The Flexible Wing Roll Controller, having the policy "drive the roll rate error to zero without exceeding wing torque limits," is a compound static regulator with constraints.

Each of the GCE applications demonstrate the Capture of Operator or Artisan Knowledge benefit to some degree. However, these applications also demonstrate a distinct benefit, Generalized Constraint Enforcement, which implements the qualifying statements which characterize human articulation of complex control policies. Several application preconditions may be stated for the GCE benefit:

1) It must be possible to measure or infer the constraint condition.
2) The designer must be able to articulate control actions to enforce the constraint.
3) A constraint condition is used to qualify a regulator or maximizer policy, rather than directly generating control due to the constraint condition alone.

The Interpolation Among Rules, the Locality of Control and the Information Equity attributes of FLC combine to produce the Generalized Constraint Enforcement benefit. Interpolation Among Rules is important for being able to implement a rule system without inducing limit cycles. As with Exception Handling, an implementation of GCE may exhibit a high gain nonlinearity as the constraint is approached. In high order systems, or systems with time delay, a limit cycle may result. In each of the GCE examples, the control input determines either the constrained variable or its derivative, and limit cycles were not reported. Locality of Control allows the designer to represent the constraint condition in appropriate antecedent-space regions. Finally, Information Equity allows qualitative knowledge of the constraint to be utilized in the control implementation.

F. Input/Output Maps via Interpolation (IOI)

An alternate view of FLC is as an interpolation engine. The Input/Output Maps via Interpolation (IOI) benefit is realized by obtaining controllers which perform well within local regions of state space, and forming the net control action by interpolating among these controllers according to the current controller input. A Waste Incinerator application [36], two vehicle navigation applications [37], [38], an environmental emissions application [40], and a nuclear reactor control application [29] demonstrate this benefit.

In one form of IOI, a fuzzy plant model is identified as a collection of crisp, linear, discrete-time, transfer-function models on fuzzy state-space partitions [39]. The high-temperature, waste incinerator application of Sugeno and Kang [36] demonstrates this mode of IOI. The multistage incinerator is a high-order, strongly coupled, nonlinear plant with time lag. In the existing control, local PID loops controlling input air regulated temperature at several of the incinerator stages, but operation was not consistent and operator intervention was frequently required. The operator control itself left much to be desired, and it was determined that capturing operator knowledge was not an approach to satisfactory control. For this system, a fourth order, linear model was identified on each of eight fuzzy state-space partitions, using plant input/output data. For each partition, linear control was designed for the identified models by minimizing a quadratic performance index. To produce the final control, contributions of the separate linear controllers were combined and resolved into a crisp number according to membership of the current state in each partition and center of gravity defuzzification.

An extension of this technique places an operator in the control loop and seeks to model the operator's actions. In [37] and [38] control laws were identified on fuzzy state-space partitions. In these applications, directed toward parking a car and negotiating a curve respectively, a human operator provides the input-output data for identification. Again, several local controllers were constructed, and the composite output of these controllers is generated using center of gravity defuzzification.

An alternative approach to the IOI benefit uses a detailed plant model to evaluate and modify control alternatives off-line, seeking a sheaf of control trajectories which maximize a figure of merit. The approach is demonstrated in an Automobile Engine Idle Control application, [45], which uses a "phase portrait algorithm" to generate manifold pressure and engine shaft speed control trajectories. The algorithm has strong ties to dynamic programming [3]. The selected transitions appear as rules in the FLC rule base.

To obtain the IOI benefit, the designer must have:

1) The means to design local control laws:
    a. Plant input/output data (operator actions, step response, random input response, etc.) is available to perform model identification and build local controllers.
    b. Alternatively, a predictive, quantitative model of the plant, or other means of computing a figure of merit for control actions, is available to solve the dynamic programming problem posed by the weighted graph approach.
2) The means to partition state space to facilitate local identification or control design, either through designer knowledge or techniques such as analysis of variance.

The IOI benefit is the result of a design strategy which departs from that seen in the COAK and EH benefits. Rather than relying on a human articulation of the control policy, an FLC obtaining the IOI benefit discovers that policy off-line in data generated by the plant. The Locality of Control attribute of FLC allows the construction of controllers local to state-space regions. The Interpolation Among Rules attribute combines the effect of these controllers, generating a continuous control action as the plant traverses state-space regions. Together, the LOC and IAR attributes deliver the Input/Output Maps via Interpolation benefit seen in these applications.

VI. Conclusions

In the years since Zadeh's coining of the term fuzzy logic and Mamdani's early demonstrations of FLC, great progress has been made in constructing the mathematical foundation of fuzzy set theory as well as developing tools for and demonstrating applications of FLC. But the utility and role of FLC remains controversial. This is true in part because fuzzy logic is not a control design methodology in the ordinary sense. Unlike Linear Quadratic Regulator theory, or even Bode plots, fuzzy logic does not prescribe a controller. Rather, it replaces the multiplications and additions (or capacitors and inductors) of the traditional control implementation. As an implementation technique, FLC places a special burden on the user to justify its use.

This paper is a contribution to this end. Based on the direct evidence of the applications literature, a taxonomy of demonstrated benefits of FLC is presented with example applications. The attributes of FLC which contribute to obtaining the benefits are discussed, as are aspects of the applications which the examples show to play an important role. Linking aspects of applications to examples demonstrating benefits provides a tool for the controls engineer: by comparing aspects of the application at hand with benefit preconditions and example papers, the engineer with a specific application will be better able to justify or refute the use of FLC. Providing a tool to support this decision has been the principle objective of this work.

To build the taxonomy, it has been important to consider the control objectives of the application designers, that is to say, the control policy. Arising out of this work, a characterization of control policies is presented which captures distinctions along three axes: type, composition and temporal behavior. It is interesting to note that while a large portion of classical and modern control theory is concerned with the design of static regulators, all of the applications of the taxonomy implement compound control policies which combine maximization or constraint enforcement with regulation. Many of the control policies implemented are dynamic.

An application benefit is the short answer to the question "What did the designer obtain though the use of FLC?" In each benefit described, a distinct answer to this question is developed. By examining applications, characterizing control policies, observing the interplay among FLC attributes, and extracting application prerequisites, the discussion of each benefit illuminates a different facet of FLC. It has been said that the problem with FLC is that nobody knows what it's good for. This is no longer true.

REFERENCES

[1] D. Abramovitch, "Some crisp thoughts on fuzzy logic," *Proc. 1994 Amer. Contr. Conf.*, pp. 168–172, AACC, Baltimore, MD, 1994.

[2] K. Aström and B. Wittenmark, *Adaptive Control.* Reading, MA: Reading: Addison-Wesley, 1989.

[3] R. E. Bellman and S. E. Dreyfus, *Applied Dynamic Programming.* Princeton, NJ: Princeton Univ. Press, 1962.

[4] S. Bennett, *A History of Control Engineering.* Sterenage: Peter Peregrinos, 1979.

[5] J. A. Bernard, "Use of a rule-based system for process control," *IEEE Control Systems Magazine*, vol. 8, no. 5, pp. 3–13, 1988.

[6] S. Chiu, S. Chand, D. Moore, and A. Chaudhary, "Fuzzy logic for control of roll and moment for a flexible wing aircraft," *IEEE Control Systems Magazine*, vol. 11, no. 4, pp. 42–48, June 1991. (See also: A Schwartz, "Comments on 'fuzzy logic control of roll and moment for a flexible wing aircraft,'" *IEEE Control Systems Magazine*, Feb. 1992 pp. 61–62; and the Authors reply, *ibid.*, pp. 62–63.

[7] P. Christian, "Fuzzy logic control of a hypervelocity interceptor," *Proc. 2nd IEEE Conf. on Fuzzy Syst.*, pp. 877–882, 1993.

[8] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized predictive control—part I, the basic algorithm," *Automatica*, vol. 23, no. 2, pp. 137–148, 1987.

[9] ——, "Generalized predictive control—part II, extensions and interpretations. *Automatica*, vol. 23, no. 2, pp. 149–160, 1987.

[10] S. Daley and K. F. Gill, "Comparison of a fuzzy logic controller with a P + D control law," *Trans. ASME*, vol. 111, pp. 128—137, June 1989.

[11] W. Daugherity, B. Rathakrishnan, and J. Yen, "Performance evaluation of a self tuning fuzzy controller," *Proc. 1st Int. Conf. on Fuzzy Syst.*. San Diego: IEEE, 1992, pp. 389–397.

[12] D. Driankov, *An Introduction to Fuzzy Control* Berlin: Springer-Verlag, 1993.

[13] I. Flugge-Lotz, *Discontinous Automatic Control.* Princeton: Princeton Univ. Press, 1953.

[14] J. A. Freeman and D. M. Skapura, *Neural Networks, Algorithms, Applications and Programming Techniques.* Reading, MA: Addison-Wesley, 1991.

[15] J. E. Gibson, *Nonlinear Automatic Control.* New York: McGraw-Hill, 1963.

[16] C. J. Harris and A. B. Read, *Knowledge Based Fuzzy Motion Control of Autonomous Vehicles: Artificial Intelligence in Real-Time Control*, M. G. Rodd and G. J. Suski, Eds. New York: Pergamon, 1989, pp. 139–144.

[17] L. P. Holmblad and J. J. Ostergaard, "Control of a cement kiln by fuzzy logic," in *In Fuzzy Information and Decision Processes,* M. M. Gupta and E. Sanchez, Eds. Amstedam: North-Holland, 1982, pp. 389–399.

[18] M.-H. Kim and S. Engell, "Speed-up of linear programming for time-optimal control," in *Proc. 1994 Amer. Contr. Conf.*, pp. 2667–2670, Baltimore: AACC, 1994.

[19] B. Kosko, *Neural Networks and Fuzzy Systems.* Englewood Cliffs, NJ: Prentice Hall, 1992.

[20] J. Layne and K. Passino, "Fuzzy model reference learning control for cargo ship steering," *IEEE Control Systems Magazine*, vol. 13, no. 6, pp. 23–34, 1993.

[21] J. R. Layne, K. M. Passino, and S. Yurkovich, "Fuzzy learning control for anti-skid braking systems," *IEEE Trans. Control Syst. Tech.*, vol. 1, pp. 122—129, Feb. 1993.

[22] F. L. Lewis, *Optimal Control.* New York: Wiley Interscience, 1986.

[23] R. J. Marks II, Ed. *Fuzzy Logic Technology and Applications.* Piscataway, NJ: IEEE, 1994.

[24] R. Meier, J. Nieuwland, A. M. Zbinden, and S. S. Hacisalihzade, "Fuzzy control of blood pressure during anesthesia," *IEEE Control Syst. Magazine*, vol. 12, no. 6, pp. 12–17, 1992 (see also [23]).

[25] Y. Nakamura, *Advanced Robotics, Redundancy and Optimization* Menlo Park, CA: Addison Wesley, 1991.

[26] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems.* Englewood Cliffs, NJ: Prentice Hall, 1989.

[27] K. Oishi, M. Tominaga, A. Kawato, Y. Abe, S. Imayasu, and A. Nanba, "Application of fuzzy control theory to the sake brewing process,"*J. Fermentation and Bioeng.*, vol. 72, no. 2, pp. 115–121, 1991.

[28] T. J. Procyk and E. H. Mamdani, "A linguistic self-organizing process controller," *Automatica*, vol. 15, no. 1, pp. 15–30, 1979.

[29] P. Ramaswamy, R. M. Edwards, and K. Y. Lee, "An automatic tuning method of a fuzzy logic controller for nuclear reactors," *IEEE Trans. Nuclear Sci.*, vol. 40, pp. 1253–1262, Aug. 1993.

[30] A. Saffiotti, E. Ruspini, and K. Konolige, "Blending reactivity and goal-directedness in a fuzzy controller," in *Proc. 2nd IEEE Conf. on Fuzzy Syst.*, pp. 134–139, 1993 (see also [23]).

[31] S. Sakaguchi, I. Sakai, and T. Haga, " Application of fuzzy logic to shift scheduling method for automatic transmission," *Proc. 2nd IEEE Conf. on Fuzzy Syst.*, 1993 pp. 52–58 (see also [23]).

[32] E. M. Scharf and N. J. Mandic, "The application of a fuzzy controller to the control of a multi-degree-of freedom robot arm," in *Industrial Applications of Fuzzy Control,* M. Sugeno, Ed. Amsterdam: North-Holland, 1985, pp. 41–61.

[33] S. M. Sharuz, G. Langari, and M. Tomizuka, "Optimal damping ratio for linear second-order systems," *J. Optimization Theory and Applications*, vol. 73, no. 3, pp. 563–576, 1992.

[34] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control.* Englewood Cliffs, NJ: Prentice Hall, Inc., 1991.

[35] M. Stienberg, " Development and simulation of an F/A-18 fuzzy logic automatic carrier landing system," *Proc. 2nd IEEE Conf. on Fuzzy Syst.*, pp. 797—802, 1993 (see also [23]).

[36] M. Sugeno and G. T. Kang, " Fuzzy modeling and control of multilayer incinerator," *Fuzzy Sets and Systems*, vol. 18, no. 3, pp. 329–345, 1986.

[37] M. Sugeno and K. Murakami, "An experimental study on fuzzy parking control using a model car," in *Industrial Applications of Fuzzy Control*, M. Sugeno, Ed. Amsterdam: North Holland, 1985, pp. 125–138.

[38] M. Sugeno and M. Nishida, "Fuzzy control of a model car," *Fuzzy Sets and Systems*, vol. 16, pp. 103–113, 1985.

[39] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control," *IEEE Trans. Syst., Man and Cybern.*, vol. SMC-15, pp. 116–132, 1985.

[40] K. Tanaka, M. Sano, and H. Watanabe, "Identification and analysis of fuzzy model for air pollution—an approach to self-learning control of {CO} concentration," *Fuzzy Logic Technology and Applications*, R. J. Marks, Ed. Piscataway, NJ: IEEE, 1994, pp. 261–266.

[41] T. Terano, K. Asai, and M. Sugeno, *Applied Fuzzy Systems*. Cambridge, MA: AP Professional, 1994.

[42] D. E. Thomas, "Fuzzy logic control—a taxonomy of novel benefits," M.S. thesis, Univ. Wisconsin–Milwaukee, 1994.

[43] M. Togai and O. Yamano, " Analysis and design of an optimal learning control scheme for industrial robots: A discrete system approach," *Proc. 24th IEEE Conf. on Decision and Control*, pp. 1399–1404, 1985.

[44] R. M. Tong, M. B. Beck, and A. Latten, "Fuzzy control of the activated sludge wastewater process," *Automatica*, vol. 16, no. 6, pp. 695–701, 1980.

[45] G. Vachtsevanos, S. S. Farinwata, and D. K. Pirovolou, "Fuzzy logic control of an automotive engine," *IEEE Control Syst. Magazine*, vol. 13, no. 3, pp. 62–86, 1993.

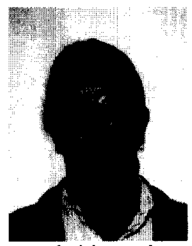
[46] C. von Altrock, B. Krause, and H.-J. Zimmermann, "Advanced fuzzy logic control of a model car in extreme situations," *Fuzzy Sets and Systems*, vol. 48, no. 1, pp. 41–52, 1992.

[47] L. X. Wang and J. M. Mendel, "An RLS adaptive filter, with application to nonlinear channel equalization," in *Fuzzy Logic Technology and Applications*, R. M. Marks, Ed. Piscataway: IEEE, 1994, pp. 389–394.

[48] O. Yagishita, O. Itoh, and M. Sugeno, "Application of fuzzy reasoning to the water purification process," in *Industrial Applications of Fuzzy Control*, M. Sugeno, Ed. Amsterdam: North-Holland, pp. 19–39.

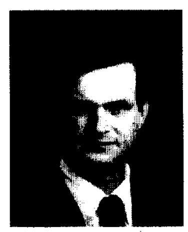
[49] S. Yasunobu and T. Hasegawa, "Evaluation of an automatic container crane operation system based on predictive fuzzy control," *Control - Theory and Advanced Applications*, vol. 2, no. 3, pp. 419–432, 1986.

[50] S. Yasunobu and S. Miyamoto, "Automatic train operation system by predictive fuzzy control," in *Industrial Applications of Fuzzy Control*, M. Sugeno, Ed. Amsterdam: North-Holland, 1985, pp. 115–128.



David E. Thomas was born in Milwaukee, WI. He received the B.Sc. in mechanical engineering and the M.Sc. in engineering from the University of Wisconsin–Milwaukee in 1985 and 1994, respectively.

Since 1985 he has been employed as a product development engineer with an emphasis on embedded, real-time software systems for analytical and biomedical instrumentation. He is currently a Senior Staff Analyst with Computer People Unlimited. He interests include intelligent decision and control systems, software engineering practices and methods, digital signal processing, and graphical description languages.



Brian Armstrong-Hélouvry received B.Sc. degrees in physics and mechanical engineering from MIT in 1980, and the M.Sc. and Ph.D. degrees in electrical engineering from Stanford University, in 1984 and 1988.

He joined the Electrical Engineering and Computer Science faculty of the University of Wisconsin–Milwaukee in 1989. In 1986, 1988, and 1991 he held visiting positions with Mitsubishi Electric Company, Osaka, Japan; INRIA, then in Rennes, France; and the State University of Campinas, Campinas, Brazil. His interests include problems of identification and control in robotics and other nonlinear systems.