Uniform Approach to Model-Based Fuzzy Control System Design and Structural Optimisation

Yun Li and Kim Chwee Ng†

Centre for Systems and Control, and
Department of Electronics and Electrical Engineering,
University of Glasgow, Rankine Building,
Glasgow G12 8LT, U.K.
Email: Y.Li@elec.gla.ac.uk

Abstract. The design problem of a fuzzy logic control system is equivalent to a multi-modal multi-dimensional optimisation problem. In the context of optimal designs, such a design problem is usually “unsolvable” by analytical or conventional numerical means. This Chapter develops a genetic algorithm based soft computing paradigm for design automation, which efficiently reveals optimised system parameters and structures in a uniform manner without compromising to a fixed architecture. Controller specification, computation and robustness issues are also discussed. Various contributions to the genetic algorithm technique are reported. This methodology is demonstrated by design examples for controlling an asymmetric nonlinear and a critically stable uncontrollable nonlinear systems.

Keywords: Fuzzy logic control, fuzzy systems, genetic algorithms, evolutionary computing

1 Introduction

With the development of computational intelligence and soft computing techniques, the human inference oriented fuzzy logic control (FLC) has received increasing attention in the control community. A fuzzy controller incorporates uncertainty and abstract nature of human decision-making into an action mechanism. Reasoning by fuzzy logic, such a system flexibly implements functions in near human terms, i.e. IF-THEN linguistic rules. It also provides good reliability and capability in dealing with nonlinear systems that are complex, ill-defined or time-varying [3, 16, 21].

There exist, however, difficulties in the design of such nonlinear control systems arising from difficulties in applying analytical methods to FLC structure and parameter optimisations under practical constraint conditions, since the design

† Now at Singapore Technologies, Republic of Singapore.
criteria or specifications may not be differentiable. Existing numerical techniques widely used in computer-aided control system design (CACSD) are also inadequate, since they are based on derivative or \textit{a priori} gradient guidance and thus have difficulties in finding the global optimum in the multi-modal design space. Further, the objective functions needed in these numerical methods must be “well-behaved” and would not, therefore, reflect practical system constraints [10, 13, 17]. In addition, a modern paradigm for computer-aided control system design (CACSD) should also provide an environment that accepts the following challenges [13]:

(a) Complexity of practical systems;
(b) Required high quality and accuracy of design;
(c) Speed of design;
(d) Competition with available design tools (e.g., in terms of ease of use); and
(e) Robustness, reliability and safety arising from the design.

This Chapter develops a systematic and reusable evolutionary computing paradigm for solving generic FLC design problems, focusing on model based design automation using a genetic algorithm (GA). An FLC scheme that is easy for practising engineers to use is developed in Section 2. Design issues are discussed in Section 3. To avoid design difficulties mentioned above, it is discussed that the design problem could be transformed into an analysis problem, initially solvable in exponential time. In Section 4, a genetic algorithm is developed effectively to explore the design space, transforming the exponential problem into an NP-complete problem and leading to the globally optimised control system. It is illustrated that, by trading off precision slightly for improved tractability, robustness and ease of design, such computing paradigm would meet the above challenges. Section 5 subsequently illustrates this methodology through the control of an asymmetric stable and a symmetrical critically-stable nonlinear systems. Conclusions are highlighted in Section 6.

2 Fuzzy Logic Control

2.1 State-Space Control Law

A schematic of a fuzzy logic control system is shown in Fig. 1. The controller relates significant and observable variables to control actions using fuzzy relationships or an algorithm [15]. The time sequenced crisp variables are first converted to fuzzy variables to serve as conditions to a \textit{rule-base} or an \textit{inference engine}. Examples of crisp variables are the error \(e(k)\) and the rate of change of error \(\dot{e}(k)\), in a similar manner to state-space based sliding mode control (SMC) [13, 16]. In fuzzy control, however, control actions are inferred from the interpretation of the \textit{memberships} of \(e(k)\) AND/OR those of \(\dot{e}(k)\).

The application of fuzzy logic techniques to control is tied together with the concept of linguistic control rules. A fuzzy controller consists of a set of such control rules to be derived from a given plant condition. The process of the membership
conversion is termed *fuzzification*. The AND logic inference is naturally, and the most commonly, used in forming the rule-base, yielding a control action such as given by:

\[
\text{IF } <\text{Condition } X_e(k) > \text{ AND } <\text{Condition } X_\dot{e}(k) > \text{ THEN } <\text{Action } X_o(k) >
\]

![Fig. 1. A block diagram of a fuzzy logic control system.](image)

Upon this logic decision, follow-up *defuzzification* is then needed to reconvert the fuzzy control decision \(X_o(k)\) to a crisp value so as to actuate and to regulate the plant. Here \(X_e(k)\) and \(X_\dot{e}(k)\) are the fuzzified versions of discretised \(e(t)\) and \(\dot{e}(t)\), respectively. This example of the widely used fuzzy control strategy makes decisions upon the amount of the tracking discrepancy and the velocity of its change. It is in analogy of a *proportional plus derivative* (PD) controller, whose control action \(u_{PD}(t)\) is mathematically described by:

\[
u_{PD}(k) = K_P e(k) + K_D \dot{e}(k)
\]

(1)

To design a fuzzy controller by a developer, it is necessary to interpret rules that are based on experience or expert’s knowledge so as to form a decision table that gives the input and output values of the controller corresponding to situations of interest. In addition to the rules for the decision table, the membership functions are to be chosen to represent the human conception of the linguistic terms. However, in the construction of the rules, the choice for the input decision variables of the fuzzy controller is dependent on the developer’s preference. These make the design imperfect, difficult and time-consuming. The impossibility of analytical and the lack of tractable numerical design tools means that an off-line model-based design process has had to be based on manual trial-and-error simulations. This Chapter tends to analyse factors and criteria concerning the design, in view of formulating a single computerised design automation approach.

### 2.2 Scaling Factors

In Fig. 1, the traditional condition and action interfaces include:

(a) An estimator is included to determine the rate of change of the controlled
variables. This rate of change is normally treated by the controller as another input decision variable;
(b) Input scaling factors used to map the range of the values of the controlled variables into a pre-defined universe of discourse;
(c) A procedure to convert the fuzzy information supplied by the controller into one unique control action in the combination and defuzzification process; and
(d) Output scaling factor to map the action values into numerical values.

The choice of the input scaling factors has a significant effect on the controller performance [14] in terms of controller sensitivity, steady-state errors, stability, transients and number of rules being activated or required. After the combination and defuzzification process, the resultant fuzzy action value is output to the plant through an output scaling process. The output scaling factor has also an effect on the overall gain of the system [14]. A large value of the factor will cause the controller to operate in a bang-bang manner driving the system to saturation, whilst a small value will reduce the overall system gain giving a sluggish transient. In conventional manual trial-and-error based designs, however, scaling is difficult to optimise.

Such scaling factors can easily be optimised in the same evolution process that optimises the other FLC parameters [16, 17]. However, the use of the overall scaling factors are unnecessary if a GA is used to optimise the memberships in the actual universe of discourses, since the I/O ranges are usually limited and are known a priori to control engineers.

2.3 Membership Functions

Relatively more precisely to reflect the fuzzy nature of linguistic classifications, exponential (bell-type) membership functions are used in this Chapter. These are given by [16, 17]:

\[
\mu_i(x) = \exp \left( \frac{x - \alpha_i}{\sigma_i} \right)^{\beta_i}, \quad \forall x \in [-\text{Big}, +\text{Big}]
\]  

(2a)

where, representing the membership classification, \(i \in \{0, \pm 1, \pm 2, \pm 3\} = \{\text{Zero, Small, Medium, Big}\} = \{0, \pm S, \pm M, \pm B\}\) and

\[
\mu_{+B}(x) = 1, \quad \forall x > \alpha_{+B}
\]  

(2b)

\[
\mu_{-B}(x) = 1, \quad \forall x < \alpha_{-B}
\]  

(2c)

Here \(\alpha_i\) describing the centre point of the membership function along the universe of discourse is termed the position parameter, \(\beta_i\) resembling evolutionary shapes is the shape parameter, and \(\sigma_i\) modifying the base-length of the membership and determining the amount of overlap is the scale parameter.

In a design, pure trapezoidal or triangular membership functions may also be used as in many existing FLC schemes. The relevant mathematical formulae describing
them [16] are omitted here for simplicity. However, they are coded together with the exponential memberships of (2) in the GA as described in Section 4. Hence, an optimal set of membership functions for a given problem may be found by the GA as a combination of different types as depicted by Fig. 2. Such a mixed structure would, however, almost be impossible to obtain using manual design trials.

![Fig. 2. Examples of combined membership functions.](image)

### 2.4 Fuzzy Rule-Base and Robustness

Take the simple fuzzy PD controller as an example for the design of a two-dimensional rule-base. The initial rules can be formed using a phase plane for switching (or SMC) operations [16], as shown in Fig. 3. Tracking boundaries in the phase plane are related to the input variables. Usually, a phase plane diagram is partitioned according to the number of fuzzy sets used for the control actions. Note the trajectory of the tracking system is to be driven towards the “0” region, where no actions would be needed, allowing the trajectory to slide towards the origin that corresponds to the desired reference of the closed-loop control system.

As shown in Fig. 4, the inference engine utilises a rule-base formulated from a collection of rules in which the doublet \{ \( X_e(k) \), \( X_k(k) \) \} contracts to produce a joint single control action. From this diagram, the rule-base can be conveniently mapped into to help with the design. This results in a 2-D skew-symmetrical lookup table as shown in Fig. 4, which also implies that (2×7-1) memberships for the output variable may be needed for a finer resolution. The limitation of such a rule-base used in conventional designs is that the symmetrical property of the lookup table may be insufficient for an asymmetric nonlinear plant found in real world applications. Notice that, in manual designs, an asymmetric table is difficult to infer.

So far there exists no robustness theory on fuzzy logic control, although the insensitivity phenomenon has been experienced widely. However, from the above observed analogy to SMC, a proving argument can be made for FLC. It is proven that SMC is robust if the piece-wise gain or slope of the sliding region is large enough. This robustness is in terms of suppressing the uncertainty bounds (in \( L_\infty \)) of the model...
by the switching gain which satisfies the Lyapunov criterion. Here the discrete rules in FLC just act as piece-wise switching actions in SMC, while the no-action region acts as the sliding region. With the discontinuity of membership positions, there already exists a non-robust region outside which the closed-loop system will be robust, unless a saturating control action cannot compensate for the uncertainty bound (which is rare). Thus, an FLC system can always be asymptotically robust.

Fig. 3. A phase plane diagram usually found in sliding mode control [16].

Fig. 4. A common skew-symmetric lookup table in manual designs of a fuzzy PD controller.
2.5 Defuzzification

Usually FLC requires membership overlapping for a smooth operation. Combining control actions in defuzzification is thus essential as more than one rule may be activated for a given set of inputs. Among various defuzzification techniques, the centroid or the centre of gravity technique is widely used. This forms the resultant crisp control action from the centre of “mass” of the outputs of activated rules.

Another popular but simpler defuzzification method is the singleton technique. It is a special case of the centroid method and produces a fuzzy output set by direct comparison of multiple actions [3]. This approach requires less computational effort in defuzzification and, in design of output membership functions, the designer only needs to determine the positions of the memberships. Owing to its simplicity and that the shape and base-length of input memberships are to be sophisticatedly designed by a GA, this defuzzification method is used throughout this Chapter. It can be inferred that, however, different defuzzification methods will result in different control actions. Thus matching the memberships and rules with a particular defuzzification technique contributes to another degree of difficulty to the design.

2.6 Integral Action in Fuzzy Control

In addition to the traditional PD-type FLC, one can use other error variables as inputs in a similar way to conventional controllers. Different control laws will, however, have different effects on the controlled responses. For example, the PD type fuzzy controller may have a poor steady-state response but a relatively good transient performance [8, 16, 17].

If designed and implemented correctly, an integral term may be used to form a fuzzy proportional plus integral plus derivative (PID) controller. The amount of the integral action needs not to be significant, but the inclusion of such a term excels in dealing with steady-state errors. However, a fuzzy PID controller is usually indirectly implemented as an integration of a fuzzy P+D+D' controller. Clearly, this incurs excessive numerical errors due to the unnecessary differentiation and redundant integration. This implementation scheme often results in a sluggish transient and a less stable system when tending to achieve a relatively fast response [2, 9, 16].

If the integral of the error signal is directly used for fuzzification, however, difficulties may arise in defining the universe of discourse if range selection or scaling is not carefully optimised. Without the need of differentiating the error signal, a trade-off to this is a direct implementation (DI) technique [16] that also results in a reduced rule-base. To proceed, let \( u_{PID}(k) \) be the control action of a conventional PID controller extended form (1) and rewrite it as

\[
\begin{align*}
  u_{PID}(k) &= K_P e(k) + K_I \sum e(k)\Delta t + K_D \dot{e}(k) \\
  &= u_{PID}(k) + \Delta t u_1(k) 
\end{align*}
\]

where \( u_{PID}(k) \) is given by (1) and
Here, in conventional PID controller terms, $K_P$ is the proportional gain, $K_I$ the integral gain, $K_D$ the derivative gain, $\Delta t$ the sampling period and the proportional action in (3b) is given by:

$$u_p(k) = K_pe(k)$$

(3c)

In (1), the effect of $K_P$ and $K_D$ can be included in the position parameters of fuzzified variables and in the rule-base to be optimised. The joint P and D action requires a two-dimensional (2-D) rule base, while in (3c) a 1-D rule-base suffices. The fuzzy proportional action of (3c) is accumulated in a conventional way to form the integral action as given by (3b). In (3a), the switch $\phi$ acts as a structural selection parameter, turning on or off the integral action according to the nature of the plant (e.g., Type 0 or Type 1) and the design performance requirement (e.g., zero steady-state errors). This is important, since an integral term is unnecessary for a plant behaving Type 1, which can follow a step command with zero steady-state errors [11]. Such arrangement thus allows the designing genetic algorithm to be able to automatically optimise the FLC structure. It is to be determined in the same optimisation process of other parameters of the controller. This parameter is, however, mathematically analogous to $K_I$ specified in a conventional PID controller in the following manner:

$$\phi = K_I / K_P$$

(3d)

Thus, for such a fuzzy PID controller, separate 2-D and 1-D rule-bases are needed. It results in a nearly one-dimensional reduction of the rule-base, compared with a complete 3-D table, and is thus termed a “2.5-D” design. Notice that for this reduction arrangement optimal designs are needed, which are difficult to achieve by manual adjustments but can easily be achieved using a genetic algorithm.

Similarly, DI and rule-base reduction for a PID controller can be inferred from:

$$u_{PID}(k) = u_{PD}(k) + \phi \Delta t \sum_{i=0}^{k} u_{PD}(i)$$

(4a)

where

$$u_{PD}(k) = K_P e(k) + K_D \dot{e}(k)$$

(4b)

Here $K'_P$ and $K_D$ are also included in the position and rule-base optimisations and $\phi$ is used as the integral selection parameter also to be optimised in the same process. Clearly, a controlling switch parameter may also be placed on the PD action $u_{PD}(i)$. Note that the exclusion of $\Delta t$ in $\phi$ and $\varphi$ implies that these values are independent of the sampling period, which helps with real-time implementations.
3 Difficulties and Possibilities of an Automated Design

3.1 The Design Problem

In control system design practice, the design task is equivalent to optimising the controller parameters so as best to meet the design objectives. Let $P_i$ represent a candidate control system. Then a uniform vector representation of the system can be given by:

$$P_i = \left\{ p_1, \ldots, p_n \right\} \in \mathbb{R}^n$$

(5)

where $i$ stands for the $i$th design candidate, $n$ the number of numerical and structural parameters associated with the control law, $p_j \in \mathbb{R}$ the $j$th parameter of the $i$th design candidate with $j \in \{1, \ldots, n\}$, and $\mathbb{R}^n$ the $n$-dimensional real Euclidean space. Note that here $p_j$ may also be used to represent (i.e., to code) the controller structure.

In the case of an FLC, the vector representation should cover all parametric and structural factors, including those involved in the memberships, rule-base, defuzzification, scaling factors and integral action. In summary, the following issues need to be considered [16].

Coarse design:

(a) Number of fuzzy input and output decision variables, i.e., antecedents and consequences;
(b) Number of fuzzy sets for each decision variables, e.g., +B, -M, +S;
(c) The actual range of universe of discourse for each decision variable;
(d) Rule structure, e.g., the IF-THEN structure;
(e) Number of rules;
(f) Fuzzy inference methods, e.g., min-max operation, product-max;
(g) Defuzzification methods, e.g., centroid, singleton method; and
(h) Fuzzy control structures, i.e., PD-type, PI-type or PID-type, automatically to be determined by the structural parameter for the integral term, $\phi$ or $\varphi$.

Fine design:

(a) Types of membership functions, i.e., triangular, trapezoidal and bell types;
(b) Numerical and structural parameters of membership functions, $\alpha_i$, $\beta_i$ and $\sigma_i$; and
(c) Rule-base, e.g., both antecedents and consequences or consequences only.

Preceding a design, defining a performance index or, in reverse, a cost function $J(P_i) : \mathbb{R}^n \rightarrow \mathbb{R}^+$ that reflects design specifications is needed. Since the task of a closed-loop control system is accurately to follow a reference or command signal, a simple and generic cost function that reflects penalties in steady-state errors, a long rise-time, a long settling-time, oscillations/overshoots and a poor stability can be used [12, 13]. Such a cost function is given by:
where \( k \) is the discrete time index in the off-line simulations evaluating the design; \( N \) the duration of the simulation; \( e_k \) the error between the measured and the desired signals at \( k \); \( \dot{e}_k \) the rate of change of error at \( k \); and \( w \) a positive weighting constant used to balance the effort on reducing steady-state errors and on limiting oscillations. Note here the penalty is weighted by the time index \( k \) so that steady-state errors can be distinguished for suppression, as they are more important to eliminate for tracking. It is, however, worth to point out that defining a cost function (or a performance index) may be application-specific and other types of Euclidean norms or their variants can also be used [10, 13].

In summary, a control system design problem can be defined as the problem of finding a design \( P_0 \), such that:

\[
J(P_0) = \inf_{P} J(P)
\]

3.2 A Non-NP Problem for Automation

Practical fuzzy control system design problems are usually unsolvable problems in the analytical domain, but practical analysis problems are solvable problems in the numerical domain by simulations. Classification of computing problems is depicted in Fig. 5 [10], where the clear area represents unsolvable problems and the shaded areas solvable problems.

The solvable problems are further divided into three categories as follow:

\[ P = \{ \text{Problems solvable in Polynomial time by a deterministic algorithm} \} \]
\( \overline{\text{NP}} \) = \{Non-NP problems\}
= \{Solvable problems that can \textbf{NOT} be solved in polynomial time\}
= \{Problems that may be solved in \textit{exponential time}\};
\textbf{NP}-complete = \{Problems solvable in polynomial time by a \textit{nondeterministic algorithm} only and not by any deterministic algorithms\}.

Not directly shown in Fig. 5, the following definitions also exist:

\( \text{NP} \) = \{Problems solvable in \textbf{P}olynomial time by a \textbf{N}ondeterministic algorithm\}
= \{Problems that can always be solved in polynomial time\}
= \textbf{P} \cup \overline{\text{NP}}-complete;
\textbf{NP}-hard = \{Solvable problems that are at least as \textbf{hard} as an \textbf{NP}-complete problem\}
= \{Solvable problems that cannot be solved in polynomial time by any deterministic algorithms\}
= \textbf{NP}-complete \cup \overline{\text{NP}}.

The fact that the unsolvable FLC design problem in the analytical domain is solvable in the numerical domain as an analysis problem has led FLC developers to tempt designing the system using \textit{conventional numerical optimisation techniques}. These techniques are based on \textit{a priori} guidance using gradient or derivative information. They usually suffer from the following drawbacks \cite{10}:

(a) \textit{Multi-Objective Problem}: Conventional optimisation techniques can usually deal with one objective at a time;
(b) \textit{Existence Problem}: Gradient guidance can adjust \( P_i \) only when \( \nabla J(P_i) \) (and in some cases a monotonic second order derivative) exist \textit{a priori} or the objective functions have well-defined smooth slopes;
(c) \textit{Practical Problem}: Conventional techniques are almost impossible to work with hard constraint conditions found in practical applications. These constraints include direct domain constraints (such as parameter range requirements and fixed relationships) and indirect inequalities (such as voltage or current limits and other hard nonlinearities). Further, in practical applications, performance information may be noisy, discontinuous, incomplete, uncertain and/or imprecise;
(d) \textit{Multi-Modal Problem}: Sequential guiding usually leads to a local optimum. The use of parallelism may overcome this to a certain extent, but bare parallelism (multi-point search) includes no mechanism to exchange information among the search points;
(e) \textit{Prior Knowledge Problem}: It is difficult to incorporate knowledge and experience that a designer may already have on the design.

Therefore, using a CACSD package based on these techniques for design, the developer has to solve the above problems by heuristic simulations. First he/she needs to input certain \textit{a priori} or guessed parameters of the controller, such as those obtained from some preliminary analysis. Then, using the package, the developer could undertake simulations to evaluate the performance of candidate controllers. If
the performance does not meet specifications, the parameter values will be modified by engineering sense or guess. Subsequently, the engineer will need to run the simulations repeatedly until a “satisfactory” design emerges.

Clearly, such a design process is neither automated nor easily carried out, since mutual interactions among parameters are hard to predict (multi-dimensional problem). For such manual adjustment, much development time has to be devoted to the quest for optimal memberships and rule-base. Further, a symmetrical rule-base has often to be assumed, at a possible expense of sacrificing the controller performance, and the developer may encounter other difficulties discussed in the previous section. When a “satisfactory” FLC system is eventually obtained this way, the developer would not be certain if the system can be improved further, since there are multiple modes in the design space.

In summary, a CACSD package coupled with conventional numerical techniques cannot automate the design. It is also found that many CACSD systems are not capable of meeting design challenges discussed in Section 1 due to the drawbacks of conventional techniques.

One theoretically possible approach to computer-automated designs is the exhaustive search. However, such an approach can hardly be implemented in practice since it is non-NP. The complete search time increases exponentially with the number of parameters that need to be optimised. Even the highly regarded exhaustive scheme dynamic programming breaks down on problems of “moderate” dimensionality and complexity. One reason for such a long evaluation time is that this method cannot learn from the information gained during evaluation or make any use of existing design expertise, whereas a GA can.

4 Automation by Evolution Using an NP Genetic Algorithm

4.1 The Genetic Algorithm

An evolutionary program, such as that based on a GA, searches the design space “intelligently” by exchanging and varying candidate parameters in the way that simulates genetics and mutation in the natural evolution process. A GA generally uses coded strings (chromosomes) of Base-2 (binary) numbers in the search process. Such an algorithm is based on an analogy to the genetic code in the human DNA structure, where a coded chromosome consists of many genes. Each gene has 20 allele values (amino acids) coded by 3 of the 4 letters representing Adenine, Cytosine, Guanine and Thymine [10]. This analogy inspired the use of Base-7 coding [10, 12, 16, 17] and other integer coding mechanisms [10, 11, 13, 16, 19 20]. Compared with natural evolution, this emulated process is more controllable, efficient and yet more flexible for artificial optimisation.

Supported by the Schema Theory, a GA has shown to offer an exponentially reduced search time compared with enumerations and thus to allow a non-NP problem to be transformed into an NP-complete problem. Before the simulated evolution process begins, an initial population of multiple coded strings representing
random and/or *a priori* designs is first formed. Every such string is assigned a performance index, or a cost function such as (6), calculated against the design specifications. The designs with a higher performance (i.e., lower cost) will reproduce themselves favourably according to Darwin’s *survival-of-the-fittest* principle. Then information and search position exchanges (in an operation termed *crossover*) and variations (termed *mutation*) of some genes (i.e., coded parameters) in the reproduced population take place randomly and sparsely, as opposed to exhaustively. Once this is complete, the performance index of every new string (i.e. every new candidate design) is re-calculated. Such evolution process repeats itself until designs cannot be improved globally meaningfully.

A schematic of a binary evolutionary design automation process is shown in Fig. 6, where the fitness function, $f(P_i): R^n \rightarrow R^+$, acts in an inverse manner of the cost function to represent the design performance. Here $f(P_i)$ is used for searching its supreme value in the optimisation. During evolution, relatively fitter designs, such as those represented by $P_2$ and $P_3$, receive more attention for further refinements in a nondeterministic manner, converging to multiple optimised designs.

![Fig. 6. Evolution of coded designs with genetic operations guided by fitness evaluations.](image)

This method improves the chance of finding the global optima by simultaneously searching from multiple points (using the whole population) with effective exchange of information, as opposed to one point at a time. Since the performance index does not need to be differentiated, it can include both mathematical and linguistic/logic terms that best interpret the design specifications and customer requirements [10–13]. It can be concluded that a fuzzy system, as well as other decision and control systems, can always be designed automatically by the GA under the following conditions [10]:
(a) Candidate designs can be evaluated such as by simulations; and
(b) There exists a performance index that reveals more information than a simple True-or-False answer.

4.2 GA Automated Design for FLC

A genetic algorithm based design technique provides an alternative fuzzy control design approach and can offer unexpected high-performances in many cases. This Chapter treats the design of the FLC system in a uniform way. A candidate design is first coded by an integer vector, representing both numerical and structural parameters associated with the control law. Such vectors thus form a multi-dimensional solution space. In engineering and real-world applications, the model of the system to be controlled is usually known or obtained in approximation. This makes an off-line design possible. The use of a model is important to design when the system is nonlinear or presents an asymmetric input-output behaviour.

In general, the GA based approach overcomes design difficulties encountered in conventional methods and provides a tractable tool for control system design automation in the following ways:

(a) **Multi-objective problem** can be dealt with by a GA easily and its objective can include both numerical and logic terms;
(b) **Existence problem** is solved by evaluations, analysis and search, instead of by a priori gradient-guidance;
(c) **Practical problem** is eased by simulations, as practical constraints are more easily solved in analysis and evaluation than in designs;
(d) **Multi-modal problem** is largely solved by multi-point (population based) nondeterministic search with effective exchange of information;
(e) **Prior knowledge problem** is solved by incorporating known or potentially good designs in the initial population of the chromosomes, which usually results in a faster convergence [10–13];
(f) **Complexity of practical systems** is better tackled in simulations than in direct design optimisations;
(g) **Required high quality and accuracy of design** is easily achieved by population based genetic optimisation yielding multiple top solutions satisfying multi-objectives;
(h) **Speed of design** is exponentially increased by the NP algorithm and automation;
(i) **Competitions with available design tools** is invaluably helped by ease of use in the GA based automation; and
(j) **Reliability, robustness and stability** arising from the design can be obtained by incorporating their indices or penalties in the fitness function. Multiple optimised solutions are also readily available for further studies.

Over the past half decade, optimal FLC designs using genetic algorithms have widely been reported [1, 4–7, 14, 16, 17]. The parameters of a fuzzy controller that are commonly optimised by a GA are reported to be the fuzzy rule-base and/or the
position and base-length parameters only. These may lack completeness in the FLC design and thus this Chapter tends to offer a systematic design approach to complete design automation, including optimising the structure of the FLC system.

4.3 Formation of GA Based Designs

In the design of a fuzzy controller using a genetic algorithm, two issues concerning the design are to be considered. One is in terms of fuzzy controller choices as discussed in Sections 2 and 3, and the other of choices of the genetic algorithm itself as detailed below:

(a) Population size = 100 (in view of the size of the FLC optimisation problem);
(b) Integer encoding (for a reduced Hamming distance, short chromosome length, small memory usage, fast operation and enhanced resolution) with parameter range selection [10, 13, 16];
(c) Prior knowledge and the initial symmetrical rule-base incorporated in the initial population (which is however optional) for a good start of the search and a faster convergence [10, 13, 16];
(d) Cost function given by (6) for good overall, transient and steady-state performances [10, 13, 16];
(e) Selection by tournament in conjunction with ranking [10, 16] (which performs more reliably and efficiently than the traditional roulette-wheel selection scheme shown in Fig. 6 and is applicable directly to both fitness and cost functions);
(f) Crossover rate = 70% for multi-point crossover (better than single-point crossover for long chromosomes needed in FLC);
(g) Adaptive mutation scheme [13, 16] (similar to evolution strategy [10]) with a maximum rate limited to 20%, which needs no prior knowledge in fixing the mutation rate; and
(h) A generation gap method used to maintain some genes for diversity of species with no extra fitness evaluations needed [10, 16].

In this Chapter, each category of the memberships and element of the rule-base is coded in Base-7 [12, 17], limited to represent the 7 choices in the discrete interval [-B, +B]. Each position, base-length and controller structure parameter is also coded by a signed integer, but is augmented with an additional digit to explore the range of the coded value, the simplex form of which is the position of the decimal point. Here, the adaptive mutation implemented also helps with reducing unnecessary diversity of the ranges when the evolution converges.

The shape structure parameter $\beta$ is, however, coded for the range [0, 9], where a coded value in the interval [1, 9] is used to represent $\beta$ for the bell shape given by (2a). For $\beta \in (0, 1)$, a trapezoid is signalled with a top-length to base-length ratio of $\beta$. If the coded value of $\beta$ is zero, then a triangle emerges. This integer thus represents not only numerals but also flags for logic selections. Note that the number of memberships needs not to be fixed at 7, but can also be coded for further structural optimisation if need. The seven memberships are however regarded as adequate and
as already computationally intensive in real-time implementations for many fuzzy systems. Thus this number is used as the upper limit of the number of memberships, while redundant ones can be found automatically by the GA.

5 Design Examples

5.1 Control of an Asymptotically Stable Asymmetric Nonlinear System

A laboratory liquid-level regulation system that simulates plants widely involved with dairy, chemical or heat-balancing processes is experimented in this Chapter. It is a coupled asymmetric nonlinear system as shown in Fig. 7 and as described by the state-space differential equation set:

\[
\begin{bmatrix}
\dot{h}_1 \\
\dot{h}_2
\end{bmatrix} = 
\begin{bmatrix}
\frac{C_1 a_1}{A} \sqrt{2g(h_1 - h_2)} \\
\frac{C_2 a_2}{A} \sqrt{2g(h_1 - h_2)} - \frac{C_2 a_2}{A} \sqrt{2g(h_2 - h_0)}
\end{bmatrix} + 
\begin{bmatrix}
\frac{1}{A} & 0 \\
0 & \frac{1}{A}
\end{bmatrix}
\begin{bmatrix}
u \\
d
\end{bmatrix}
\] (8)

Here, \(h_1(t)\) and \(h_2(t)\) are the liquid levels of Tank 1 and Tank 2, respectively; \(u(t)\) is an input flow rate mapped from a pump voltage; \(d(t)\) is also a pumped input but is used to test the rejection of disturbances when need; \(C_1 = C_2 = 0.58\) are discharge constants; \(a_1 = 0.386\) cm\(^2\) and \(a_2 = 0.976\) cm\(^2\) are orifice areas; \(A = 100\) cm\(^2\) is the cross-sectional area of both tanks; and \(g = 981\) cm s\(^{-2}\) is the gravitational constant. There are two practical constraints imposed on this system. One is by its physical structure, being \(h_0 = 3\) cm, the minimum liquid level bounded by the height of the orifices. The other is by the upper limit of the pump capacity, being \(\max(u) = 33.3\) cm\(^3\) s\(^{-1}\). Here the pre-amplifier gain is adjusted to map the full D/A converter output voltage (4.56 V) to this maximum flow rate.

![Diagram](Image)

**Fig. 7.** Twin-tank liquid-level regulation: an asymmetric nonlinear system.

The objective of this control system is to drive, through the input to Tank 1, the
liquid level at Tank 2 towards the desired level of 10 cm as fast as possible with minimal overshoots and steady-state errors. Subsequently a step-down command of 5 cm is given at 800 s with similar objectives to the step-up operation. In this system, a transportation delay of 6 s is observed from the physical laboratory system. Implemented in Turbo Pascal, the genetic algorithm is used in the design of an FLC controller, combined with the 4th-order Runge-Kutta integration in the simulations. The equivalent control law is given by (3) and the rule-structure is:

**PD part:** If $h_2$ is $A$ and $\dot{h}_2$ is $B$ Then $u_{PD}$

**I part:** If $h_2$ is $C$ Then $u_I$ (and then accumulating for $u_I$)

and the GA optimised rule-base is shown in Fig. 8.

![Control Law Table]

Fig. 8. A GA optimised lookup table for an asymmetric PID FLC system.

Since the rule tables can be optimised asymmetrically and freely, symmetric memberships have been used to simplify the design in this example. The optimised membership parameters are listed below.

Position parameters:

$e$: $\alpha_{e_B} = \pm 92.7$; $\alpha_{e_M} = \pm 34.6$; $\alpha_{e_S} = \pm 18.3$; $\alpha_0 = 0$ (fixed)

$\Delta e$: $\alpha_{\Delta e_B} = \pm 969.0$; $\alpha_{\Delta e_M} = \pm 658.0$; $\alpha_{\Delta e_S} = \pm 217.4$; $\alpha_0 = 0$ (fixed)

$I$ term: $\alpha_{I_B} = \pm 867.0$; $\alpha_{I_M} = \pm 708.9$; $\alpha_{I_S} = \pm 559.5$; $\alpha_0 = 0$ (fixed)

Note: The input flow rates are physically bounded by $[0, 33.3]$ and the system has asymmetric dynamics. Thus $\alpha_{I_B}$ and $\alpha_{I_B}$ for the outputs are fixed to +3 and -3 so as
to map into 4.56 V (i.e., 33.3 cm\(^3\) s\(^{-1}\)) and 0 V (i.e., 0 cm\(^3\) s\(^{-1}\)), respectively. Since the equilibrium point changes with the command for different levels of Tank 2, it is not used to map \(\alpha_0\). Instead, \(\alpha_0 = 0\) maps into the middle voltage, 2.28 V.

**Base-length parameters:**

\[ e: \begin{align*} \sigma_{EB} &= 21.9; \quad \sigma_{2M} = 21.9; \quad \sigma_{3S} = 27.7; \quad \sigma_0 = 9.7; \end{align*} \]

\[ \Delta e: \begin{align*} \sigma_{EB} & = 262.4; \quad \sigma_{2M} = 255.7; \quad \sigma_{3S} = 336.5; \quad \sigma_0 = 222.1; \end{align*} \]

I term: \( \sigma_{EB} = 343.2; \quad \sigma_{2M} = 343.2; \quad \sigma_{3S} = 361.3; \quad \sigma_0 = 222.8; \)

Note: There are no output base-length parameters as singleton defuzzification is used.

**Shape parameters:**

\[ e: \begin{align*} \beta_x = 2.0; \quad \beta_{2M} = 2.5; \quad \beta_{3S} = 3.5; \quad \beta_0 = 0; \end{align*} \]

\[ \Delta e: \begin{align*} \beta_{EB} = 1.5; \quad \beta_{2M} = 0.35; \quad \beta_{3S} = 3.5; \quad \beta_0 = 2.0; \end{align*} \]

I term: \( \beta_{EB} = 0.3; \quad \beta_{2M} = 0; \quad \beta_{3S} = 0.3; \quad \beta_0 = 4.5; \)

Note: “T” stands for a trapezoid/triangle is signalled.

**Controller selection:**

\[ \phi = 0.007 \]

This FLC system has yielded a simulated response depicted in Fig. 9. In [12] and [17], it was shown that fuzzy controllers outperformed conventional PID controllers in terms of smoothness and the rejection of disturbances. To test the sensitivity of this 2.5-D fuzzy PID controller, a disturbance inflow of 8.33 cm\(^3\) s\(^{-1}\) was also added. As can be seen from Fig. 9, it did reject such disturbances well in the simulation. The performance is very close to that obtained in [12], although the rule-bases are quite different. This indirectly implies that FLC is intrinsically robust. The closeness of this result also means that the controller outperforms those obtained from a full 3-D and a reduced 3-D (with “don’t-care” states) rule-bases [12]. The implemented performance of this fuzzy controller on a laboratory test system is shown in Fig. 10. As can be expected, it is subject to measurement noise.

### 5.2 Control of a Critically Stable Uncontrollable Nonlinear System

The GA based uniform design approach has also been tested in a symmetrically nonlinear cart-pole balancing system, which is critically stable. The typical differential equations of such a system is given by [7]:

\[
\ddot{x} = \frac{F_u + m_P(\dot{\theta}^2 \sin \theta - \ddot{\theta} \cos \theta) - \mu C \dot{x}}{m_C + m_P}
\]  

(9a)
\[
\dot{\theta} = \frac{-g \sin \theta + \cos \theta \left( -Fu - m_p L \dot{\theta}^2 \sin \theta + \mu_C \ddot{x} \right)}{m_C + m_p} - \frac{\mu_p \dot{\theta}}{m_p} \left( L \left( \frac{4}{3} - \frac{m_p \cos^2 \theta}{m_C + m_p} \right) \right)
\]

(9b)

where \( m_C = 1.0 \) kg is the mass of cart; \( m_p = 0.1 \) kg, mass of pole; \( L = 0.5 \) m, length of pole; \( g = 9.81 \) m s\(^{-2}\), gravitational constant; \( \mu_C = 0.0005 \), friction coefficient of the cart track; \( \mu_p = 0.000002 \), equivalent friction coefficient of the pole on the cart; \( u(t) \in [-1, 1] \), control input to vary the magnitude of the directional force to the cart; \( F = 10 \) N V\(^{-1}\), force constant applied to cart centre of mass by \( u(t) \); \( x(t) \), position of cart from the centre point; and \( \theta(t) \), angle of the pole from its vertical axis.

In this application, the cart is free to move along a one-dimensional track while the pole is free to rotate only in the vertical plane of the cart and track. A force of varying magnitude can be applied by \( Fu(t) \) to the centre of the mass of the cart at discrete time intervals. The objective of this control system is to apply controlling forces to the cart until it is stationary at the centre of the track while simultaneously positioning the pole in the vertical position without any angular velocity [7]. This is an uncontrollable problem in control theory, since the controllability matrix does not exist due to the number of variables to control (i.e., \( x \) and \( \theta \)) is greater than the number of the controlling input (i.e., \( u \)).

Fig. 9. Simulated response of the 2.5-D fuzzy PID controller designed by the GA.
Fig. 10. Implemented response of the 2.5-D fuzzy PID controller designed by the GA.

It has been reported widely that an FLC system can successfully control this classically uncontrollable system, e.g., in [1, 2, 7]. In the design of a fuzzy logic controller for this system, it is important to identify appropriate decision variables that constitute to an effective control action. As given in (9a) and (9b), four decision variables identified as being important are the current positions of cart and pole and the rate of changes of these positions, namely, $x$, $\dot{x}$, $\theta$, and $\dot{\theta}$ [7]. Due to these four inputs, however, a 4-D rule-base will be needed. This has led to the use of a small number (chosen as 3 in this Chapter) of fuzzy sets for each variable. It is in analogy to the use of the bang-bang control scheme for such a system, the control signal provided by which is either switched on positive, on negative or off. However, the FLC system is naturally expected to provide a smoother control action. The fuzzy control rule here is given by:

$$\text{If } \theta \text{ is } A \text{ and } \dot{\theta} \text{ is } B \text{ and } x \text{ is } C \text{ and } \dot{x} \text{ is } D \text{ Then } u$$

This is, in effect, similar to devising “PD” control, since designing a traditional PID type FLC system would be over complicated. Thus the control law equivalent to (4) is adopted here to achieve a full PID action based on the added accumulation of the above PD action. It is automatically designed by the GA. The structural parameter controlling the integral action is found to be $\varphi = 0.047$ and other optimised parameters as well as a 4-D rule-base can be found in [16]. Simulated closed-loop response of the cart-pole balancing problem is depicted in Fig. 11 with an initial condition of $x = -2.0$ m, $\theta = 0.2$ rad ($11.5^\circ$) and zero velocities. It can be seen that
the cart and pole of the control system have reached and stabilised at their desired positions within approximately 2 seconds. Small oscillations of the pole are observed. Fig. 12 shows the control actions taken by the fuzzy controller.

![Graph showing control actions taken by the fuzzy controller.](image)

**Fig. 12.** Control actions taken by the fuzzy PID controller.

The physical system however experienced more oscillations, but still stabilised to zero steady-state errors eventually. The response was quite similar to that shown in Fig. 11 and the overall performance of the fuzzy controller was satisfactory. Tests on other initial conditions also yielded similarly performance. To test the controller robustness in term of rejecting sensitivities to parameter variations, this fuzzy control system has also been evaluated with variations of the mass of cart $m_c$ by ±20%. Fig. 13 shows the corresponding performances of the controller for the initial conditions.
as used in the first test. As can be seen, the performances are quite satisfactory. However, it was also observed that the pole oscillated very heavily when the mass is increased by more than 20% due to a larger inertia.

![Graph showing simulated responses of the fuzzy controller with ±20% parameter variations.](image)

**Fig. 13.** Simulated responses of the fuzzy controller with ±20% parameter variations.

### 6 Conclusions

This Chapter has provided a thorough analysis of the design problem of FLC systems. It has been reported the application of genetic algorithms to FLC system design automation, which avoids a painstaking trial-and-error process arising from the complexity of design and from the lack of analytical and numerical design tools. Due to the fact that the problem of fuzzy logic control system design is equivalent to a multi-dimensional, multi-modal and multi-objective optimisation problem, the genetic algorithm provides a tractable tool for design automation with structural optimisation. The methodology is particularly useful in engineering systems, since a practical system always has constraints imposed by physical limitations.

Although some globally optimised design can emerge from evolution, off-line design can by no means replace on-line tuning completely, since the plant model of an engineering system cannot be infinitely accurate. The robustness of the FLC system, however, can compensate for the model uncertainty in a high degree. Further work is currently undertaken at Glasgow on on-line automatic tuning using a fast micro-GA and hill-climbing and results will be reported in due course.

### Acknowledgement

Dr. Y. Li’s work is partially supported by the EPSRC funding granted to him (Evolutionary programming for nonlinear control, K24987). Financial support form University of Glasgow and CVCP to Dr. K.C. Ng during 1992-95 is also gratefully acknowledged.
References


