Genetic Algorithm-Based Fuzzy System Design Using a New Representation Scheme

Spyros. G. Tzafestas*

Spyros. N. Raptis*

Triantafyllos Pimenides**

*Intelligent Robotics and Automation Laboratory
Division of Signals, Control and Robotics
National Technical University of Athens
Zographou, Athens, GR 15773
GREECE
E-mail: tzafesta@softlab.ntua.gr

**Systems and Control Division
Department of Electrical and Computer Engineering
University of Patras
Patras, GR 26500
GREECE
E-mail: pimenide@ee.upatras.gr

Abstracts: Genetic algorithms (GA's) are powerful stochastic search algorithms for general problem solving. Their effectiveness as a tool for evolving other systems has been early identified and has gained increasing research interest. Fuzzy systems may strongly benefit from GA's since they involve a quite large number of parameters that need to be tuned for the system to achieve the required performance. Such parameters include (but are not limited to) the definition of the fuzzy sets stored in the fuzzy rule base. The parameter tuning process becomes more important for the cases where the fuzzy system is meant to be used for function approximation. The evolution of a fuzzy system via GA's involves stochastic varying of the parameters defining the fuzzy sets. It is, thus, important for the "robustness" of the overall process that these parameters are appropriately defined. The conventional representation of fuzzy sets through their membership function values at discrete points or through the a-level set representation do not possess the required characteristics to be directly exposed to GA search. A more efficient representation scheme based on the latter is proposed and its properties are investigated. The advantages of this scheme are illustrated through the non-toy application of evolving a fuzzy system for inverse robot kinematics by the use of GA's. It is argued that the proposed fuzzy set representation possesses better properties from the stochastic search point of view.

Spyros Raptis: Diploma in Electrical and Computer Engineering of the National Technical University of Athens (NTUA) Greece (1994). Ph.D. in Soft Computing carried out at the Intelligent Robotics and Automation Laboratory of NTUA (2001). Presently, Researcher at the Institute of Language Processing (ILP), Athens, Greece. Several publications in neural, fuzzy and genetic algorithm applications in robotics, decision making and control. *Current interests :* Intelligent Systems, language and speech processing, multi-agent systems, neural networks, fuzzy logic and genetic algorithms. Member of the Greek Technical Chamber.

Triantafyllos Pimenides: He received Degree in Mathematics from University of Athens 1974, Diploma in Electrical Engineering from University of Patras 1981, Ph.D in 1984 from the University of Patras. He is an Associate Professor of Systems & Control of the University of Patras. His research interests include Control of Multivariable Dynamical Systems, Robotics, N-dimensional Systems, Computer–aided Design, Analysis Methods and Optimal Control. He has published over than fifty papers and three books. Mr. Pimenides T. is a member of the Technical Chamber of Greece and a member of the Greek Mathematic Society.

Tzafestas Spyros G.: full professor, Director of the Institute of Communication and Computer Systems (ICCS), the Signals, Control and Robotics Division and the Intelligent Robotics and Automation Laboratory (IRAL) of the National Technical University of Athens (NTUA). Holder of Ph.D. and D.Sc. in Control and Automation. Recipient of Honorary Doctorates of the International University (D.Sc. (Hon.)) -and the Technical University of Munich (Dr.-Ing. E.h.) .Fellow of IEEE (N.Y.) and IEE (London) ; Member ofASME (N.Y.) , New York Academy of Sciences , IMACS (Rutgers, N.J.) and SIRES (Brussels). Member of IFAC SECOM and MIM TCs. Project evaluator of national european and international projects (USA, Canada, Italy, Hong Kong, Japan). Project coordinator of national and EU projects in the fields of robotics, CIM and IT (ESPRIT, BRITE-EURAM , TIDE, INTAS , SOCRATES, EUREKA, GROWTH etc.). Publications: 30 research books, 60 book chapters, over 700 journal and conference technical papers. Editor-in-Chief of the Journal of Intelligent and Robotic Systems and the book series "Microprocessor-Based and Intelligent Systems Engineering" (K1uwer). Organizer of several international conferences (IEEE, IFAC, IMACS, IASTED, SIRES etc.). Listed in several international biographical volumes. Current interests include: control, robotics and CIM.

1. Introduction

Genetic Algorithms

Genetic algorithms (GA's) have become one of the main attractions for researchers and practitioners and there is a very good reasons for that: they work! They are model-free estimators, since they require no information relatively to the system that they are trying to model or optimize except for a fitness measure, i.e. a measure of how well the algorithm did in the selection of a specific parameter set.

Although the mechanics of GA's are rather simple, their remarkable success in dealing with complex real problems has not yet been thoroughly explained. Put in other words, they do work but we don't fully know why and how. What is certain is that they process a large amount of useful information, the so-called *schemata*, and they do it rather fast through their *implicit parallelism* [1]. Another unique feature of GA's is that they work with a population of points rather than a single search point and use probabilistic transition rules rather than deterministic. This enables them to escape local minima in multimodal surfaces and proceed with their quest for global optimality.

Possessing such attributes, genetic algorithms qualify as an excellent general purpose tool which explains the vast number of diverge areas they have been successfully used for.

Like any process that involves probabilistic search, GA's are better suited for off-line processes since their response time (i.e. the time required to find a "satisfactory" parameter set) cannot be known in advance or even be considered constant.

From the problem categories they address, it has been early identified that one of the most interesting is that of the design and/or tuning of a system that will exhibit a required behavior, based on a performance measure. GA's are possibly not very well suited for on-line performance themselves but they are certainly suited to design systems to exhibit it.

To stress this point through a simple example, one practice could be to use a GA to solve the inverse kinematic problem of a robot arm based solely on its forward kinematics. But a better practice would be to use a GA to evolve an appropriate system (e.g. a fuzzy or neural system) that could perform the same task. It is obvious that the latter is much more meaningful. This very task we will undertake at a later section.

Fuzzy Systems

On the other hand, fuzzy systems are well suited for on-line performance but require detailed domain-specific knowledge in the form of linguistic production rules. Often such knowledge is difficult or expensive to obtain, or it simply does not exist, especially for multivariable complex systems and processes. Even when such knowledge can be obtained, it is a common (and necessary) engineering practice to fine-tune it based on precise numerical data.

It is quite tempting (and rewarding) to try to automatically design a fuzzy system by the use of an appropriately arranged genetic algorithm. Designing a fuzzy system mainly consists of building its rulebase, which decomposes to defining and associating appropriate fuzzy sets. At some cases a specialized inference mechanism may also be required.

No doubt, one of the most demanding tasks required by a fuzzy system is that of the approximation of an unknown function based on input-output data since the error is of major importance (as opposed to the use of fuzzy systems for classification purposes). In this endeavor, it is crucial that the values of the fuzzy system parameters are accurately set, i.e. the membership functions' shape and position are optimized.

The use of GA's to determine the parameters of a fuzzy system is not new; numerous related publications can be found that use various techniques to merge GA's with fuzzy and neural models, some indicative ones being [4] and [5]. In this paper we address this problem fundamentally: from the representation of the fuzzy set itself.

2. Representation of Fuzzy Sets for GA Purposes

In the sequel, only *convex* and *normal* fuzzy sets will be considered since they are the dominating and most useful form of fuzzy sets for inference purposes. There are two widely used ways to represent such a fuzzy set (figure 1):

1. Quantization of the universe of discourse and storing of the values of the membership function at each point.
2. Quantization of the membership values and storing of the respective intervals at each level.

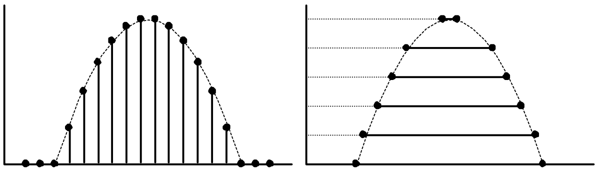

Figure 1: The Various Representation Schemes. Quantization of (a) the Universe of Discourse, (b) the Membership Grades.

While the former is the most common, the latter is argued to be the more efficient [2] and makes use of the *resolution identity theorem* to express a fuzzy set (and the related operations) as a superposition of crisp sets, namely its α-level sets (or α-cuts):

$$P = \bigcup_{0<\alpha\leq 1} \alpha P_\alpha$$

From a representation point of view, the first method uses *n* parameters for a fuzzy set. It is clear that a random change of one of the representation values may result in problems: the fuzzy set may cease to be convex and/or normal as illustrated in figure 2 (a).

The second method uses a number of parameters proportional to the number of required levels, say *m*. This method has more severe problems when exposed to a genetic search: the fuzzy set may become nonsense as illustrated in figure 2 (b).

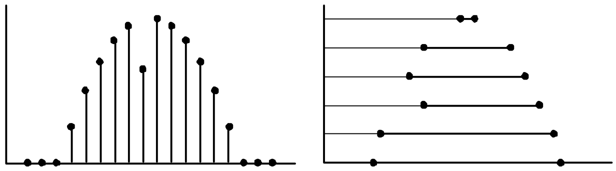

Figure 2: Problems with the Representations

Invoking a GA to search directly the definition space of a fuzzy set represented either by the first or the second method, will most probably result in undesirable situations when subject to mutation and crossover operators. To avoid such situations, a common way around would be to perform thorough checking each time the crossover and mutation operations are performed. This is both time consuming and makes the problem much harder for GA to handle since:

- it introduces additional “artificial” nonlinearities in the problem,
- important similarities and highly fit schemata will be difficult to find, and
- areas of the problem space will be cut off during checking.

After all, GA’s are blind since they work with a coding of the parameters and not the parameters themselves. These problems imply a representation that is poorly suited for direct GA search.

3. The Proposed Representation Scheme

The proposed GA-oriented representation is based on α-cuts but not in the left-point-right-point interval form proposed in [2]. A rather recursive way is used to represent each cut based on the previous cut.

This representation:

- Guarantees that the fuzzy set will remain convex and normal for *every* combination of the involved parameters.
- Like the other methods mentioned, it guarantees a *unique* representation, i.e. each set of parameters corresponds to a single fuzzy set and vice versa.
- Is not more memory consuming than [2] since it uses the same number of parameters (two per level set).

- Possesses interesting geometric properties.

Consider a universe of discourse X, and a fuzzy set P defined on X. Furthermore, assume a set of ordered levels α_i, i=1,..., m, such that: $0<\alpha_1<\alpha_2<...<\alpha_m=1$. Then P can be decomposed into α-level sets according to the resolution identity theorem:

$$P = \bigcup_i \alpha_i P_i \tag{1}$$

In [2], each level set P_i is expressed by the interval:

$$\left[x_i^l, x_i^r\right] \text{ where } x_i^l, x_i^r \in X \tag{2}$$

We introduce a different representation that makes use of two different parameters, namely c_i and l_i such that:

$$c_i \in (-1, 1) \text{ and } l_i \in (0, 1] \tag{3}$$

These two parameters do not define P_i explicitly (as the x_i^l and x_i^r pair) but with respect to the immediate lower level set, P_{i-1} (figure 3). The role of c_i is to position P_i with respect to P_{i-1}:

$$c_i = \frac{center(P_{i-1}) - center(P_i)}{length(P_{i-1})/2} \tag{4}$$

A value of zero indicates that the two centers coincide while a value of +1 (-1, respectively) indicate that the center of P_i is at the rightmost (leftmost) position of P_{i-1}.

The role of l_i is to capture the width of P_i but not with respect to the width of P_{i-1} but to the maximum width it could have at the specific position c_i, which is:

$$length(P_{i-1}) - 2\left|center(P_i) - center(P_{i-1})\right|$$

Thus:

$$l_i = \frac{length(P_i)}{length(P_{i-1})\left(1 - |c_i|\right)} \tag{5}$$

This choice guarantees that for all combinations of c_i and l_i the representation will always be both valid and unique. Thus, employing a GA on a representation scheme of this kind is straightforward requiring no checking during mutation and crossover (or any other possible bit-altering operators) and the search space can be argued to be “smoother”.

Switching from and to the Proposed Representation

To keep the formulas short, we introduce:

$$C_j = cente(P_j) = \frac{x_j^l + x_j^r}{2} \text{ and } L_j = length(P_j) = x_j^r - x_j^l \tag{6}$$

From (4) and (5), taking into account (6), it is straightforward to express c_i and l_i as a function of x_i^l, x_i^r, x_{i-1}^l, and x_{i-1}^r.

The other way round we can easily arrive at:

$$x_i^l = C_{i-1} - \frac{c_i L_{i-1}}{2} - \frac{l_i L_{i-1}}{2}\left(1 - |c_i|\right)$$
$$x_i^r = C_{i-1} - \frac{c_i L_{i-1}}{2} + \frac{l_i L_{i-1}}{2}\left(1 - |c_i|\right)$$

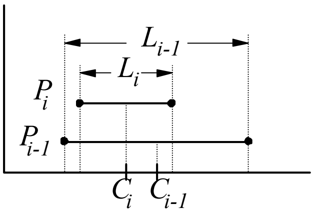

Figure 3: Switching from and to the Proposed Representation.

Geometric Properties

The proposed representation possesses some interesting properties. For example, decreasing the parameter l_i of one of the level sets, will result in an proportional shrink of all the higher level sets. Moving the center c_i of a level set, moves all the higher level sets the same way.

Thus, a "smooth" change in one of the representation parameters of a set causes a "smooth" alteration of the set. Moreover, the lower the cut whose parameters are being changed, the greater the effect that these changes will have on the overall set.

Implementation

To illustrate the recursive nature of the proposed definition, we provide a C-like code fragment that performs the required conversion from the proposed representation to the standard α-level set representation. Note that the first α-cut in the proposed representation is defined with respect to the universe of discourse.

```
//Index j stands for i-1
xjl=left-boundary; xjr=right-boundary;

for (uint i=0; i<nCuts; i++) {
    Cj=(xjr+xjl)/2.;
    Lj=xjr-xjl;

    xil=Cj-ci*Lj/2.-li*Lj/2.*(1.-fabs(ci));
    xir=Cj-ci*Lj/2.+li*Lj/2.*(1.-fabs(ci));

    //xil and xir are now valid and the
    //membership value for the [xil, xir]
    //interval is 1./nCuts*(i+1)

    xjl=xil; xjr=xir;
}
```

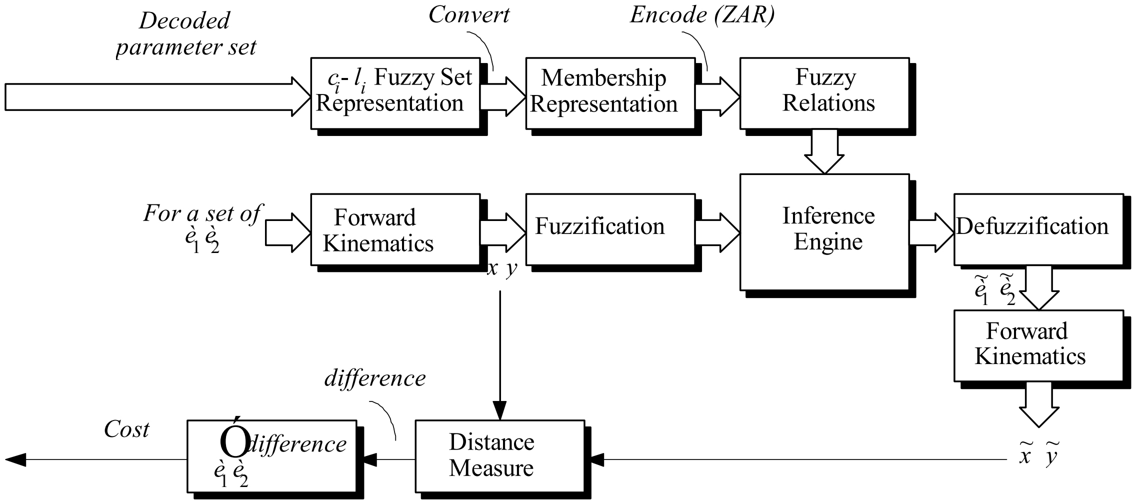

Figure 4: Fitness Evaluation for the Simulation

4. Simulation

To evaluate the proposed representation, we address the problem of automatically designing a fuzzy system for inverse robot kinematics by the use of genetic algorithms, based solely on the forward model of robot at hand. The design of the fuzzy system implies the design of the rule base and the involved fuzzy sets.

For simplicity, we will consider a simple planar 2R robot arm. The forward kinematic formulations for the specific robot is given by:

$$x = d_1 \cos(\theta_1) + d_2 \cos(\theta_1+\theta_2)$$
$$y = d_1 \sin(\theta_1) + d_2 \sin(\theta_1+\theta_2)$$

and the inverse kinematic formulations by:

$$\varphi = \tan(y/x)$$
$$\theta_1 = \text{atan2}(y - d_2 \sin(\varphi), x - d_2 \cos(\varphi))$$
$$\theta_2 = \varphi - \theta_1$$

Addressing this problem with a fuzzy system will require fuzzy rules of two-inputs and two-outputs. So, each of the rules will have the form:

IF x is A AND y is B THEN θ_1 is C AND θ_2 is D

We will demand from the GA to produce and optimize *nRules* such fuzzy rules. Each of the four fuzzy sets involved in each rule will use the proposed representation scheme and will be decomposed to *nLevels* level sets. Taking into account that for each level set two parameters are needed and assigning *nBits* bits per parameter, we arrive at a total of:

$$nBits\frac{bits}{parameter} \times 2\frac{parameters}{level} \times$$
$$\times nLevels\frac{levels}{set} \times 4\frac{sets}{rule} \times$$
$$\times nRules\frac{rules}{individual}$$

bits per individual of the GA population. The module that will be responsible for calculating the fitness of each individual, is depicted in figure 4.

The following remarks can be made:

1. The second joint angle has been appropriately constrained so as to exclude redundant solutions, specifically $\theta_2 \geq 0$ (e.g. $\theta_1 \in [0, \pi/2]$ and $\theta_2 \in [0, \pi]$).

2. The link lengths were chosen in such a way that $l_1 > l_2$ so as to avoid negative values in the *y* axis.

3. With the above assumptions, the universes of discourse for the input variables will be: $x \in [-d_2, d_1+d_2]$ and $y \in [0, d_1+d_2]$. To define input fuzzy sets it is much more convenient for the GA to augment these intervals by a small proportion so that the sets can be efficiently positioned near the limits.

4. Although the outputs of the fuzzy system are the joint angles θ_1 and θ_2, the error is calculated based on the difference of actual and required *x* and *y*. This is a common practice in inverse kinematics analyses and assures that we are trying to minimize the correct error criterion.

5. The fuzzy sets produced by the GA are first converted to the standard membership representation in order for standard inference techniques to be employed. An alternative would be to convert them to the left-right-point interval representation of α-cuts and apply the operations proposed in [2].

6. It is possible that some parameter sets define such fuzzy sets that no rule is activated for a specific pair of joint angles θ_1 and θ_2. In these cases, we "punish" the respective individual of the GA population by adding a large quantity to its cost (e.g. 100 or 1000). In this way we ensure that the search is soon directed to individuals that cover the complete input universes.

7. If membership representation was used for the fuzzy sets, then instead of the 2×*nLevels* coefficient in the bit calculation formula, there would be a *nQuanta* coefficient. It is obvious that to achieve an acceptable accuracy, *nQuanta* should be significantly larger than 2×*nLevels*. Moreover, with membership representation additional checking for convexity (and normality) would be required.

Results

Clearly the accuracy of the resulting database inference process, is directly affected by the specific values chosen for *nBits*, *nLevels*, and *nRules*. For the test purposes of this simulation, the follow values were used:

nBits = 16, *nLevels* = 5, and *nRules* = 5

An equally important parameter in this case, is the value of *nQuanta*, which affects the inference process and was chosen to be equal to 20. An instance of the simulation execution (with *nLevels* = 3 and *nQuanta* = 10) is provided in figure 5. An indicative progress of the best individual fitness of the GA popu lation is given in figure 6.

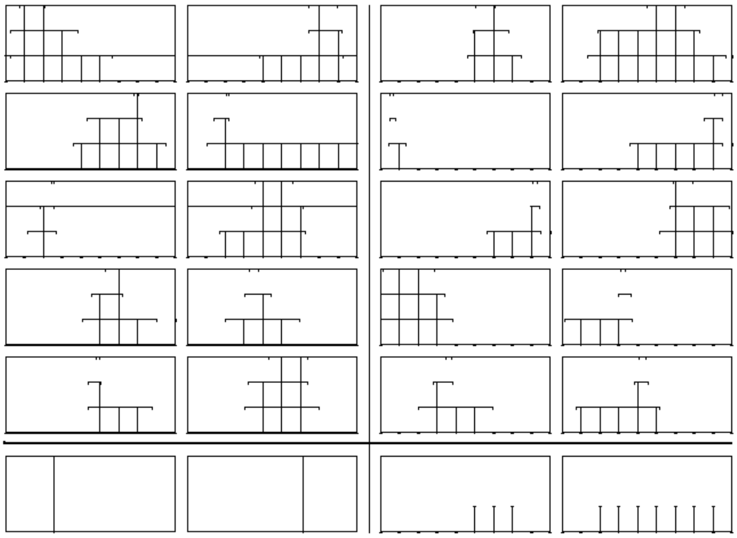

Figure 5: An Instance of the Fuzzy Database Evolution

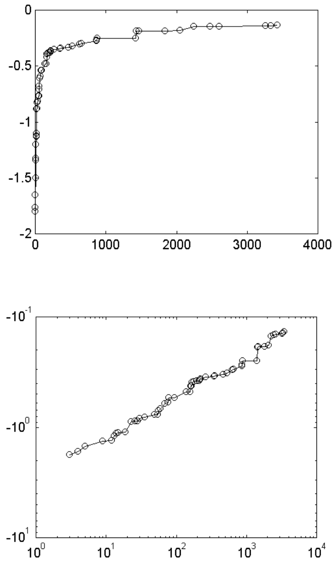

Figure 6: Evolution of the Best Fitness

5. Conclusions - Future Work

In the lack of global systematic methodologies to dictate efficient encoding of parameters for use with GA's, the need for "robust" representations is central since it greatly affects the performance and search capabilities of GA's. Imposing constraints to the individuals of a genetic algorithm externally introduces severe obstacles in the search process as opposed to building constraints intrinsically into the representation.

The novel fuzzy set representation proposed, captures the specific constraints relating to the fuzzy inference process. It intrinsically imposes convexity and normality to the fuzzy sets, thus relieving the GA from time-consuming checking and discarding of unacceptable individuals that the standard representation schemes tolerate.

The recursive definition of fuzzy sets under this representation makes it rather interesting to extract an equality measure for two fuzzy sets defined this way. This equality measure could be directly used for inference purposes.

It would also be interesting to use adaptive fuzzy systems techniques and error-minimization techniques inspired by neural networks in combination with the proposed fuzzy set representation [3]. However, most of these techniques suffer from the single point local searching curse: local minima.

REFERENCES

1. D. E. GOLDBERG, **Genetic Algorithms in Search, Optimization and Machine Learning**, Reading, MA: Addison-Wesley (1989)

2. K. UEHAR, M. FUJISE, **Fuzzy Inference Based on Families of α-Level Sets**, IEEE Trans. Fuzzy Systems, Vol. 1, No. 2 (1993)

3. L. -X. WANG, **Adaptive Fuzzy Systems and Control: Design and Stability Analysis**, Prentice Hall, Englewood Cliffs, NJ 07632 (1994)

4. C. L. KARR, E. J. GENTRY, **Fuzzy Control of pH Using Genetic Algorithms**, IEEE Trans. Fuzzy Systems, Vol. 1, No. 1, pp. 46-53 (1993)

5. D. PARK, A. KANDEL, G. LANGHOLZ, **Genetic-Based New Fuzzy Reasoning Models with Application to Fuzzy Control**, IEEE Trans. Systems, Man, and Cybernetics, Vol. 24, No. 1, pp. 39-47 (1994)