Context Adaptation in Fuzzy Processing and Genetic Algorithms

Ricardo Gudwin and Fernando Gomide

State University of Campinas, Faculty of Electrical and Computer Engineering, CP 6101 - 13.083-970 - Campinas - SP - Brasil,
Email: gudwin@dca.fee.unicamp.br, gomide@dca.fee.unicamp.br

Witold Pedrycz

University of Manitoba, Department of Electrical and Computer Engineering, Winnipeg, Canada R3T 2N2
Email:pedrycz@ee.umanitoba.ca

In this paper we introduce the use of contextual transformation functions to adjust membership functions in fuzzy systems. We address both linear and nonlinear functions to perform linear or nonlinear context adaptation, respectively. The key issue is to encode knowledge in a standard frame of reference, and have its meaning tuned to the situation by means of an adequate transformation reflecting the influence of context in the interpretation of a concept. Linear context adaptation is simple and fast. Nonlinear context adaptation is more computationally expensive, but due to its nonlinear characteristic, different parts of base membership functions can be stretched or expanded to best fit the desired format. Here we use a genetic algorithm to find a nonlinear transformation function, given the base membership functions and a set of data extracted from environment classified by means of fuzzy concepts.

I. INTRODUCTION

One of the most challenging areas of investigation concerning fuzzy information processing is the adjustment of membership functions to best represent concepts in real environments. The fine tune of such membership functions is critical when evaluating the effectiveness of fuzzy solutions to engineering problems[1], e.g., fuzzy control, fuzzy modeling and fuzzy clustering.

Linguistic terms, viewed as a family of fuzzy sets, do provide a sort of linguistic space quantization. The linguistic terms play an instrumental role in encoding both, numerical and non-numerical information[2]. The main problem is how to find the fuzzy sets that best represent the linguistic terms they are associated with, to obtain, e.g., satisfactory models or controls.

Different attempts have been made to solve this problem. Earlier approaches used some form of scaling the underlying numerical variables to fit them into normalized universes of discourse. Actually, this is what is got when gains are used in fuzzy controllers[2,3]. They provide a way to implement linear scaling of numerical variables prior to encoding them in terms of fuzzy sets. These gains were initially considered as design parameters. More recently, adaptive techniques were used to determine the gains dynamically through learning[4]. Alternative methods determine the membership functions but some of them do not consider the relation among linguistic terms and the semantic they carry, and generally address fuzzy concepts directly[5,6,7]. Hybrid approaches adjust both, gains and membership functions. An example of such approach[8], in the framework of fuzzy control, adjusts membership functions of the input variables and the gain for the output variable. The use of linear and polynomial-based modification of the universe of discourse as a way of adjusting membership functions has also been proposed[9]. Recently[10], the authors introduced the idea of using contextual information for the assignment of meaning to linguistic terms. The key point here is that many types of knowledge are general and independent of the specific context. For instance, intuitively we agree that BIG is greater than MEDIUM, which in turn is greater than SMALL. Thus these concepts may be represented by base membership functions defined not in a particular universe of discourse, but in a reference one e.g., the unit interval [0,1]. When assigning meaning to a linguistic term, a context is generated e.g., in a form of an interval [a,b][10], based on samples collected from the environment. The corresponding membership functions are determined by a mapping from the base membership functions, scaled to fit the context interval. This is a form of linear context adaptation[10]. An extension of this scheme was devised for non-linear mappings implemented through neural networks[11]. The advantage of nonlinear mappings is that the membership function shape can also be adjusted. Clearly, this is not possible when using linear context adaptation.

It should be noted that the problem we are concerned with could be viewed in two main perspectives. First, as a mechanism to fine-tune fuzzy systems (e.g., controllers or models), and second as an attempt to answer a key question about the generality of linguistic terms, i.e., how context influences concept understanding.

In this paper we introduce the use of contextual information in the assignment of meaning to fuzzy concepts. After a general discussion about the influence of context in perception, linear and nonlinear context adaptation is reviewed, including a formalization of the context adaptation problem. Next we develop a scheme to find nonlinear transformation functions using a genetic algorithm. Simulation experiments and an application example are also included to illustrate the usefulness of context adaptation. Final remarks and some issues deserving further research conclude the paper.

II. CONTEXT AND MEANING

Psychologists have long been intrigued with the finding that context appears to influence perception. The same stimulus information in different contexts can produce different perceptual events. The effect of context in perception has been widely investigated by cognitive psychology. Early studies have been made analyzing context influence in anaphoric reference resolution[12], conceptual combination[13] and speech and word recognition[14,15,16].

Considering concepts representable by fuzzy sets, the main effect of a context can be related with some sort of filtering. That is, the same base concept can be perceived in different situations, provided it is filtered to suit the context particularities.

When a context is fixed, we restrict the working universe of a system. That is, even if the universe is large enough, the context induced by a particular behavior may imply in using part of the universe only. Thus, working with the same concept in a restricted universe may alter the perception of that concept. This observation suggests a mechanism to characterize previously known concepts within the scope of new situations. Instead of building a new representation from scratch, it would be better to derive new representations from the known concept through a transformation function that we call here a context adaptation. In this sense, context adaptation is a transformation function that provides a reevaluation of a given concept, considering that a different context is present. In other words, a known information is modified (transformed) to adapt itself with a new situation. To further clarify this issue, consider the following example.

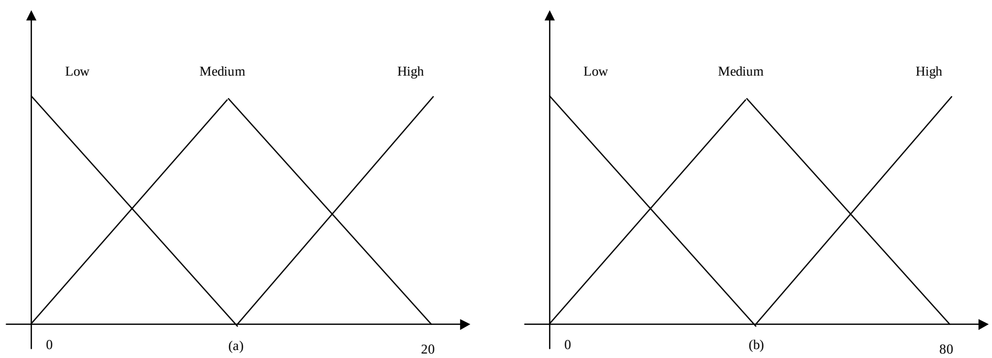

Figure 1. Linguistic variable "waiting time" in a fuzzy elevator group controller

Assume that the fuzzy sets shown in figure 1 represent the meanings associated with the linguistic variable "waiting time" in a fuzzy elevator group controller[17]. In (a) we have the case in which the system is working under normal traffic condition and in (b) the case of up-peak traffic condition. Note that when in normal condition, a waiting time of 20 seconds is considered high, but when in up-peak condition the same value is viewed as low and medium with the corresponding degrees. This is an instance where the same concepts (low, medium, and high) can be used in different contexts without loosing their original meaning (i.e., high is greater than medium, medium is greater than low, etc.). In the example, it is obvious that we can derive (b) from (a) or vice versa, but a more interesting and general approach is to define a standard frame of reference, figure 2.

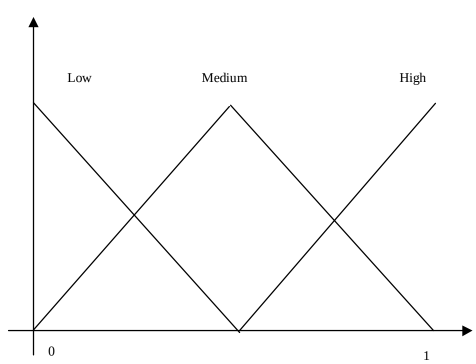

Figure 2. Frame of reference

In this case, the fuzzy sets attached to each context are derived through a transformation function (linear, in the example of figure 1) applied to the base set. The procedure underlying this general approach is given in figure 3.

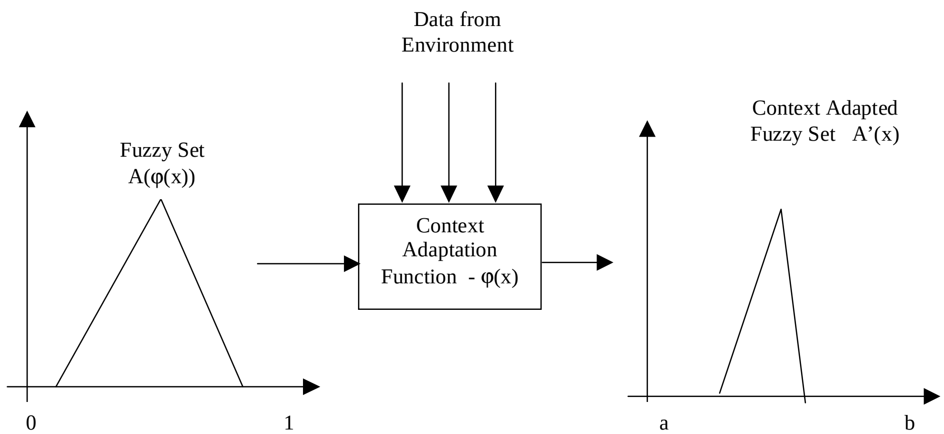

Figure 3. Context adaptation procedure

The use of a context adaptation procedure can be useful in many different circumstances, including the following typical ones:

a) Samples of the fuzzy sets we want to get are available (e.g. due to experimental outcomes). Thus, we arbitrate a collection of fuzzy sets that we judge to be suitable to represent the concepts under analysis (base membership functions defined in standard universe of discourse). Next, we derive a context which, when used with the base membership functions, best approximates the data.
b) Non fuzzy data samples and an intuitive description of the membership functions format are given. This description is used to build the base sets in a standard universe, and data used to determine a context. After, we derive the adapted fuzzy sets.
c) An arbitrary collection of fuzzy sets is provided, and its use as a model for the same concepts in different contexts is required. Assuming characterizations of the contexts as known, the fuzzy sets associated with each context are derived.
d) An arbitrary collection of fuzzy sets is given, and a changing context is known to be the case. Thus, context is continuously adapted to fit the fuzzy sets to each new context.

In all cases above, the context adaptation procedure is more or less the same: we arbitrate a collection of fuzzy sets defined on a normalized universe, determine the context, and apply a transformation function to derive the new fuzzy sets adapted to the underlying context.

Let us now properly formalize the context adaptation procedure. Consider a collection of fuzzy sets (linguistic terms) assembling a frame of reference (cognition)[3]

$$A = \{A_1, A_2, \dots, A_c\}$$

defined in [0,1]. As usual, we require that A satisfies some requirements of semantic integrity[3] such as unimodality and normality of the membership functions of A_i 's.

Given a data set of experimental outcomes (coming e.g. from a certain process of expert polling), they can be arranged in the form of N (c+1)-tuples, namely

$(d_1 , (\mu_{11} , \mu_{12} , ... , \mu_{1c}))$
$(d_2 , (\mu_{21} , \mu_{22} , ... , \mu_{2c}))$
...
$(d_N , (\mu_{N1} , \mu_{N2} , ... , \mu_{Nc}))$.

Our intent is to accommodate these data to the highest extent by adapting the context of A (case (a) from above). The essence of this process is to map the unit interval of the base universe of discourse to a more suitable universe of discourse, e.g. [a,b] with $a = \min_{i=1...N} d_i$, $b = \max_{i=1...N} d_i$ (see section IV for other examples).

The essence of the context adaptation is that adopting a unity interval [0,1] as the base set, it is mapped onto [a,b] in an adequate form. If φ is a linear function, then all the original membership functions of A are contracted or expanded, moved to left or to right, depending on φ's parameters. If φ is nonlinear, then some of the regions of [0,1] might be contracted while others could be expanded. After applying this mapping we obtain the family of fuzzy sets

$$A' = \{A'_1 , A'_2 , ... A'_c \}.$$

It is remarkable that the effect of a nonlinear mapping may result in the modification of the granularity of fuzzy sets of the frame of cognition adapted to the new data.

Let us now define the mapping $\varphi : [a,b] \rightarrow [0,1]$ in more detail. The formal requirements for φ involve:

(i) **Continuity**

(ii) **Monotonicity** - more precisely, we require that φ is nondecreasing (it can be almost constant over some regions). By doing that we assure that the meanings of A_i 's do not change.

(iii) **Boundary Conditions** - the boundary conditions $\varphi(a) = 0$ and $\varphi(b) = 1$ allow us to accommodate all experimental data.

We can also request that φ is differentiable, which is merely a matter of optimization convenience. Once φ has been defined, it is used to generate the new frame of cognition A' whose fuzzy sets are computed as:

$$A'_i (x) = A_i (\varphi (x)), x \in [a,b], \varphi(x) \in [0,1] .$$

In this sense, a context can be properly formalized by C = (a,b,φ). Thus, the proper determination of a context involves two steps:

a) find the upper and lower bounds involved,
b) find an appropriate transformation function φ.

The information needed in context adaptation could be acquired from data collected from the environment of interest, but we usually have to arbitrate function φ's format. If we consider φ as linear, then from the above conditions we have a trivial solution:

$$\varphi(x) = \frac{(x - a)}{(b - a)}.$$

If φ is nonlinear, then it is to be found or we may guess its format and find the corresponding parameters.

III. DATA COLLECTION PROCEDURES

As shown in the previous section, a context C is a tuple (a,b,φ). Thus, to adapt a context we have to obtain data from environment to get [a,b], and next to find a function φ. As summarized in figure 4, different procedures can be used to generate [a,b] from data.

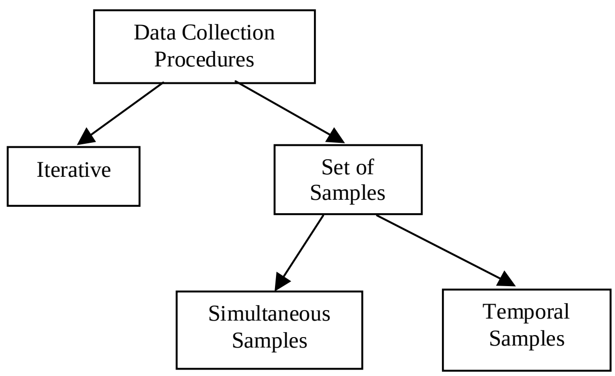

Figure 4. Data collection procedures

The basic issue here is how to collect a suitable set of data to generate the interval [a,b]. Useful techniques include some sort of iterative procedure, or a comparison procedure with a set of samples. If we choose the second we still may consider the simultaneous availability of all samples, or data collected in a time frame.

III.1. Iterative Procedure

In this case, the interval [a,b] is generated for each new sample coming from the system. Thus, it is not needed to collect a complete set of data before computing the values a and b. This class of procedure is frequently used by humans to generate contexts. For example, consider a variable, which we do not have any information about such as the concentration of a reagent in a chemical process. If it is given information that the reagent concentration in a first sample is 3 ppm, one cannot, in principle, classify this information as "low", "medium" or "high" (unless classification is done using an arbitrarily defined context based on a similar experiment). But, if we say that last sample measured 18 ppm, one is inclined to classify the 3 ppm sample as "low", as a particular context is emerging from the given information. But, if it is the case that most samples measure between 2 and 4 ppm, we can reevaluate our classification and say that the first sample concentration of a reagent is "medium", because the context definition is becoming well defined.

III.2. Simultaneous Sample Procedure

If we assume the simultaneous availability of samples for a given variable, they can be compared to determine the interval [a,b]. For example, suppose that we are classifying the arrival rate of persons at an elevator hall, at a given time interval, among three classes: “low”, “medium” and “high”. Since we have a hall at each building floor, we can compare the arrival rate in a specific hall with all others. This comparison allows us classify a particular hall rate as "low", "medium", or "high" because the context under evaluation is considering multiples instances of the variable “arrival rate of persons in hall” within the same time interval.

III.3. Temporal Sample Procedure

When no multiple samples of a variable is available, like the arrival rate of persons in a hall in the example above, we must compare temporally distributed samples to determine a context. In this case, we have to collect samples within a time window and use them to determine the interval [a,b].

Next, we review some basic methods for context adaptation and suggest additional ones as useful techniques for adaptive systems.

IV. CONTEXT ADAPTATION METHODS

IV.1. Absolute Limit Interval Determination

The absolute limit method is based on a collection samples (either simultaneously or temporally acquired) in which the interval [a,b] is generated as follows. The lower bound a is taken as the minimum, and the upper bound b is taken as the maximum values among samples available. For example, in the elevator group control problem, consider the variable "call waiting time", which measures how much time a call is waiting to be served. In this case, we can compare all simultaneous samples (all calls in the building). Suppose that the waiting times of these calls are the following:

$$\{10, 21, 17, 15, 12\}.$$

This set is our collection of samples, and the absolute limit method defines a lower bound $a = 10$ and an upper bound $b = 21$. Associated with the function φ, the interval [a,b] defines the context at a given time instant. The main characteristic of this method is its simplicity. Besides being very simple to be implemented it works quite well, especially when samples are well distributed. When this is not the case, some problems may occur. For instance, consider the waiting time samples as

$$\{51, 1, 55, 57, 53\}.$$

If we use the absolute limit method, the differences among the four greatest values could be misevaluated. To define the sample set, in the case of simultaneous samples comparisons, the sample set dimension is the multiplicity of variable's instances. When doing temporal comparisons, it is necessary to define a time-slice where samples are collected, and after taking all samples, the method can be applied. More sophisticated sample set definition could be the recording of a number of past states, in a FIFO buffer, and using those recordings as the sample set, or doing averages of past states and recording them in FIFO buffers, generating different orders of time magnitude for temporal evaluation. (Something like the context of last 10 minutes and the context of last 24 hours).

IV.2. Elastic Limit Interval Determination

The elastic limit procedure is a method that continuously adapts the bounds of a context interval through an exponential filter. For a given a sample, the method first determines which bounds will be adapted:

If the sample s is greater than the upper bound b, then only b is adapted.
If s is less than the lower bound a, then only a is adapted.
If a <s < b, both LB and UB are adapted.

The law of adaptation is the following:

$$\text{NewBound} = \alpha\ (\text{Old Bound}) + (1-\alpha)\ \text{Sample} \qquad 0 \le \alpha \le 1$$

The parameter α determines the convergence rate. The closer α is to 1, the slower the bound changes, and the closer to 0, the faster the adaptation. The value α is a design parameter. In some cases α can be variable.

IV.3. Statistic Context Determination

The statistic method is performed with samples collected either simultaneously or temporally. It attempts to avoid the problems encountered with the absolute limit method, i.e., when the samples are not well distributed. Actually, it is not a single, but a family of methods. The simplest one uses the mean and standard deviation of the samples, and assigns the lower and upper bounds as follows:

$$M = \sum_{i=1}^{N} \frac{x_i}{N} \qquad SD = \left(\sum_{i=1}^{N} \frac{(x_i - M)^2}{N} \right)^{1/2} \qquad \begin{array}{l} a = M - 3SD \\ b = M + 3SD \end{array}$$

This method demands more processing time than the absolute limit method, but it provides better solutions to poorly conditioned samples. Clearly, other statistic measures can be used.

IV.4. Neural Networks Based Context Generation

If the fuzzy set modeling the concept is discrete, then we can use a neural network to generate a context and to perform context adaptation. A simple illustrative example is shown here. A more elaborated case has been given elsewhere[11]. Below, we assume a neural network, with a learning rule similar to Kohonen[18] adaptation law. The neural network reflects the statistical distribution of inputs in its weights and is well suited for context information applications. In this case, context is a finite set in which each element defining the context is a neuron output. The input to fuzzy system will not be a single value, as usual, but a discrete fuzzy set, defined as the values of output layer of the neural network. This property allows better integration with other sub-nets, considering a kind of neuro-fuzzy processing. In figure 5 we show the main scheme of this neural network.

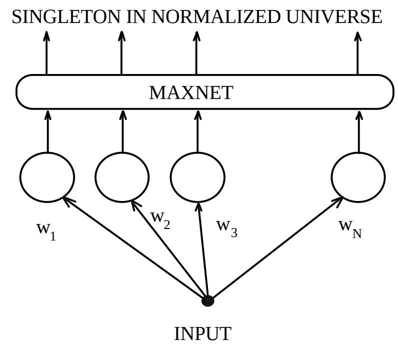

Figure 5. Example of Neural Network

The single input is connected to to each neuron through a weight w_i. Each neuron is adapted by a Kohonen learning procedure, as for example, the one described in [19]. After a significant number of samples (inputs) were presented, the neural network will have as weights, an ordered range of numbers proportional to statistical sample distribution. In figure 5, the MAXNET subnet is the one outlined in [19], where only the neuron with the closest value to input will remain active, all other will be innative. Outputs may then be used by a fuzzy procedure to do the matching phase. Despite this feature, it is necessary to detach that as weights are updated closer to sample distribution, its evolution in time corresponds to a context adaptation. So, if membership factors are associated to the weights that link MAXNET output nodes to a superior sub-network, this sub-network could implement the matching phase, with matches the input singleton to a particular fuzzy concept being evaluated.

V. GENERAL NONLINEAR CONTEXT ADAPTATION

In the general case, context adaptation means to determine the interval [a,b] and a nonlinear transformation function φ as well. There are a number of potential candidates for φ. The one to be considered here from now on is the linear combination of sigmoidal-like functions:

$$\varphi(x) = \alpha \,.\, \sum_{i=1}^{c} \frac{k_i}{1+\exp\left(\frac{-(x-m_i)}{s_i}\right)} + \beta$$

where c denotes the number of sigmoidal functions with parameters k_i (scale), m_i (translation) and s_i (steepness). We have that

$k_i \geq 0$
$m_i \in [a,b]$
$s_i > 0$

The scale factors α and β are computed as:

$$\alpha = \frac{1}{\sum_{i=1}^{c} \frac{k_i}{1+\exp\left(\frac{-(b-m_i)}{s_i}\right)} - \sum_{i=1}^{c} \frac{k_i}{1+\exp\left(\frac{-(a-m_i)}{s_i}\right)}}$$

$$\beta = -\alpha \, . \sum_{i=1}^{c} \frac{k_i}{1+\exp\left(\frac{-(a-m_i)}{s_i}\right)}$$

Figure 6 illustrates the underlying idea

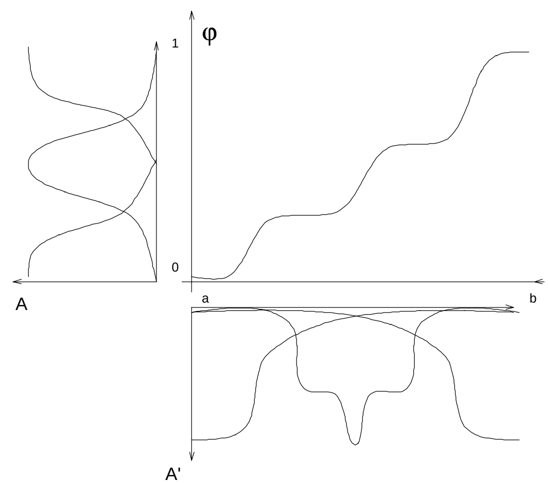

Figure 6. Nonlinear Context Transformation

VI. COMPUTING THE TRANSFORMATION FUNCTION

Once we arbitrate on the format of function φ, the whole determination of φ is completed via optimization computations. For the discussed method, we assume pairs of data

$$(d_i, (\mu_{i1}, \mu_{i2}, \ldots, \mu_{ic})) \qquad i = 1,\ldots,N .$$

Furthermore, considering the MSE (Mean Squared Error) criterion, one gets:

$$Q = \sum_{i=1}^{N} \sum_{j=1}^{c} \left(\mu_{ij} - A(\varphi(d_i)) \right)^2$$

Remembering that φ is a function with 3c parameters k_1 , m_1 , s_1 , ... , k_c , m_c , s_c , the determination of φ reduces to:

$$\min_{k_1, m_1, s_1, \ldots, k_c, m_c, s_c} Q \quad \text{s.a.} \begin{cases} k_i \geq 0 \\ m_i \in [a,b] \\ s_i > 0 \end{cases}$$

The solution of this problem by means of standard optimization computations is very complicated, as it will fall in a system of nonlinear equations. To solve this problem we use, instead, a genetic algorithm.

VII. GENETIC ALGORITHM PROCEDURE

As usual in genetic algorithms (GA) the parameters to be optimized are encoded by chromosomes. Each chromosome has the following format: a gene of the chromosome is a floating-point number representing the following sequence of parameters:

$$[\, k_1 , m_1 , s_1 , k_2 , m_2 , s_2 , \ldots , k_c , m_c , s_c \,].$$

In the GA procedure, first a initial population of chromosomes is generated randomly, and the value of Q, used as the performance index, is computed for each of them. In addition, several populations, with the same number of individuals M as initially, are generated next using the initial population as a reference. They are generated using the following schemes:

1- crossover using individuals of the reference population selected by a roulette wheel algorithm, and by the performance indices[20] of the individuals; crossover is done with 50% of genes.

2- inductive mutation of the best individual; to each gene of this individual a random value ranging from -50% to +50% of its value is added.

3- standard mutation applied to individuals of the reference population chosen by the roulette wheel algorithm. In this case, 50% of the genes are randomly changed to values within the valid range for each gene.

4- inductive mutation of the best individual; to each gene of this individual a random value within -1% to +1% of the corresponding range is added.

5- uniform crossover in which a chromosome is taken from the reference population using the roulette wheel algorithm, and the mating chromosome is chosen randomly among all the individuals of the populations generated in the steps 2, 3 and 4 above. Crossover is performed with 50% of the genes.

After generating these additional populations, all the individuals are mixed and ranked according to their respective performance indices. Next, the best M individuals are selected to remain in the next reference generation. After a number of generations, the best individual is chosen as the optimizing parameters.

VIII. COMPUTATIONAL RESULTS

For simulation purposes, we assume the form of φ and membership functions of A (Gaussian) as given to generate test data. During the experiments, we arbitrate an input universe of discourse and a nonlinear context function φ as explained in the previous sessions. Thus, for equally spaced values x on the chosen universe of discourse, we calculate φ(x) and A_i (φ(x)). The values of x are stored as samples d_i and the values A_i (φ(x)) stored as the corresponding μ_{ij}. The GA finds the original parameters using these samples. Here we consider two candidate functions. The first, F1 (see Figure 7), is a strictly monotone function. For this case the genetic algorithm has always converged to the correct solution. For the second, F2 (see Figure 8), however, the genetic algorithm always converged to a set of parameters, but sometimes these parameters did not correspond to the correct solution, but to a local minimum. The function shape differed from the original one, but kept consistent with it in the least squares sense. Therefore, in the case of local minimum, the result is an approximation for the actual transforming function. The linguistic terms defined in [0,1] were characterized by Gaussian membership functions of the form $A_I = G(x, m_I, \sigma_I)$ where $G(x,m,\sigma) = \exp(-(x-m)^2 / \sigma^2)$.

For all the experiments, we used the following values:

a = 5;   b = 10;   c = 3;   N = 20
$A_1 = G(x,0,0.25)$
$A_2 = G(x,0.5,0.25)$
$A_3 = G(x,1,0.25)$.

VIII.1. First Experiment: Function F1

Original parameters :

K_0:0.30 m_0:5.10 s_0:0.70 K_1:0.40 m_1:5.50 s_1:0.80 K_2:0.50 m_2:6.00 s_2:0.90

Obtained parameters:

K_0:0.02 m_0:6.56 s_0:0.57 K_1:0.01 m_1:5.03 s_1:0.98 K_2:0.32 m_2:5.09 s_2:0.98

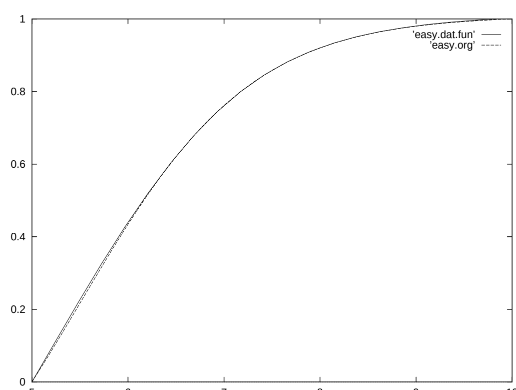

Figure 7. Function F1 for Experiment 1

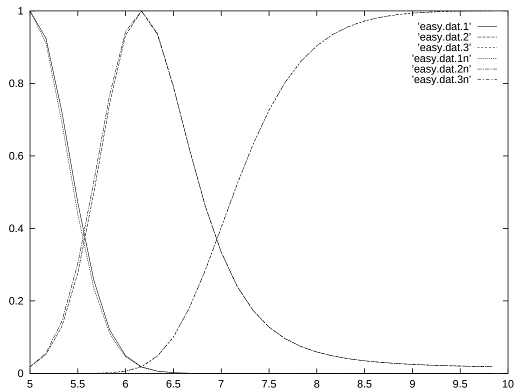

Figure 8. Membership Functions of A´ for Experiment 1

VIII.2. Second Experiment : Function F2

Original parameters:

K_0:0.75 m_0:6.25 s_0:0.10 K_1:0.74 m_1:7.50 s_1:0.10 K_2:0.75 m_2:8.75 s_2:0.10

Obtained parameters: first trial

K_0:0.48 m_0:6.84 s_0:0.36 K_1:0.22 m_1:8.76 s_1:0.09 K_2:0.01 m_2:8.31 s_2:0.36

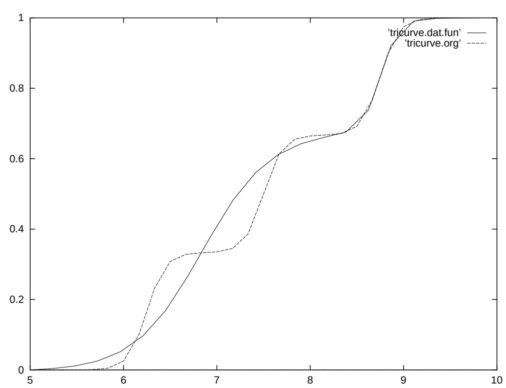

Figure 9. Function F2 for Experiment 2, Trial 1

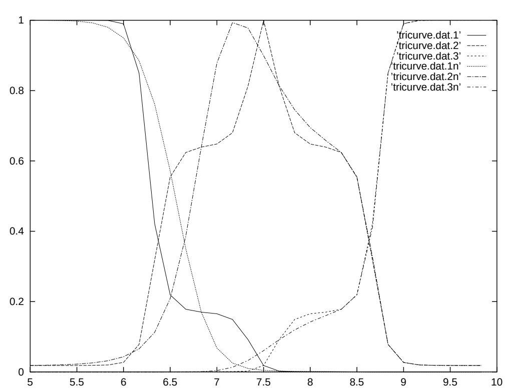

Figure 10. Membership Functions for Trial 1

Obtained parameters: second trial

K_0:0.00 m_0:6.32 s_0:0.63 K_1:0.22 m_1:6.24 s_1:0.08 K_2:0.63 m_2:8.20 s_2:0.57

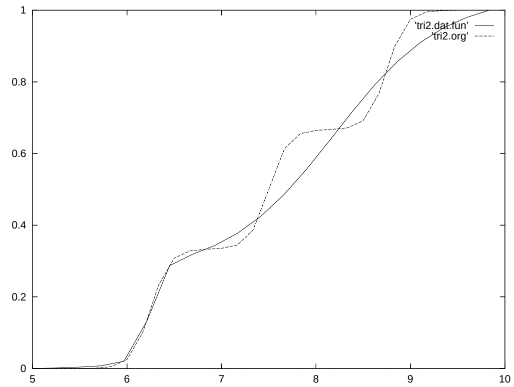

Figure 11. Function F2 for Experiment 2, Trial 2

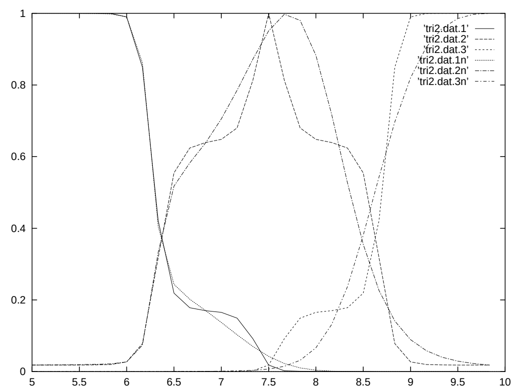

Figure 12. Membership Functions for Trial 2

Obtained parameters : third trial

K_0:0.75 m_0:6.25 s_0:0.10 K_1:0.74 m_1:7.50 s_1:0.10 K_2:0.75 m_2:8.75 s_2:0.10

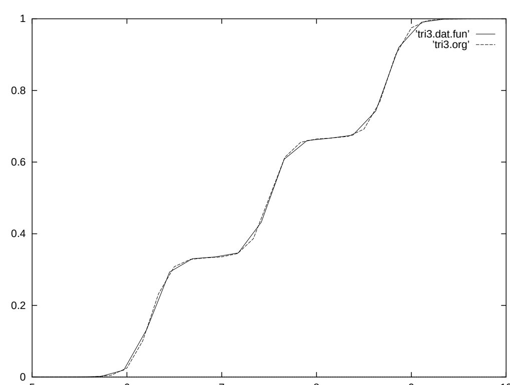

Figure 13. Function F2 for Experiment 2, Trial 3

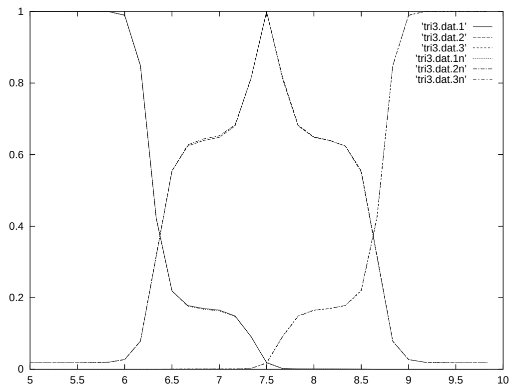

Figure 14. Membership Functions for Trial 3

IX. CONCLUSIONS

The primary intent of this paper was to pose the problem of nonlinear context adaptation and to come up with a relevant optimization framework.

The results have shown that, once some requirements on the membership functions hold, for strict monotone transforming functions the context adaptation scheme proposed always provided the appropriate, correct answers. Additional experiments (not reported here) have confirmed this property. It remains to formally proof this statement. This is a task to be pursued in the future. For non-strictly monotone transforming functions, the results have shown that only (reasonable) approximations are guaranteed because the algorithm may stop at a local minimum. However, as usual in practice, result validation should be done with judgment. Of course, this may imply several trials before the solution is found.

The importance of the addressed idea is evident at the conceptual level. By raising the problem of calibrating fuzzy sets we tackled the issue of attaching meaning of generic linguistic terms that hold even in different contexts.

The direct application aspects have not been discussed. They could be quite easily envisioned in adaptation of rule-based control and fuzzy modeling. These will be analyzed in a separate study.

References

1. M. Braae, D.A. Rutherford. “Selection of Parameters for a Fuzzy Logic Controller” - *Fuzzy Sets and Systems*, 2, 1979, pp. 185-199.

2. W. Pedrycz. *Fuzzy Control and Fuzzy Systems*. Research Studies Press, J. Wiley, Taunton, New York, 1989.

3. W. Pedrycz. *Fuzzy Sets Engineering*. CRC Press, Boca Raton, Florida, 1995.

4. L. Magdalena. “Adapting Gain and Sensibility of FLCs with Genetic Algorithms” - *Proceedings of the 6th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems - IPMU’96* - vol. II, Granada, Spain, 1-5 July, 1996, pp. 739-744.

5. O. Cordón, F. Herrera. “Generating and Selecting Fuzzy Control Rules Using Evolution Strategies and Genetic Algorithms” - *Proceedings of the 6th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems - IPMU’96* - vol. II, Granada, Spain, 1-5 July, 1996, pp. 733-738.

6. F. Herrera, M. Lozano, J.L. Verdegay. “Design of a Control Rules Base Based on Genetic Algorithms” - *Proceedings of the 6th International Fuzzy Systems Association World Congress IFSA’95* - vol. I, São Paulo, Brazil, 1995, pp.265-268.

7. Velasco, L. Magdalena. “Genetic Learning Applied to Fuzzy Rules and Fuzzy Knowledge Bases” - *Proceedings of the 6th International Fuzzy Systems Association World Congress IFSA’95* - vol. I, São Paulo, Brazil, 1995, pp.257-260.

8. Oh, D.J. Park. “Self-Tuning Fuzzy Controller with Variable Universe of Discourse” - *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Vancouver, Canada, 1995, pp. 2628-2632.

9. J.C.T. van der Veen. *Fuzzy Sets, Theoretical Reflections, Application to Ship Steering*. M.Sc. Thesis. Dept. of Electrical Eng., Delft University of Technology, Delft, the Netherlands, 1976.

10. R.R. Gudwin, F.A.C. Gomide. “Context Adaptation in Fuzzy Processing”- *Proceedings of the I Brazil- Japan Joint Symposium in Fuzzy Systems*, Campinas, SP, Brazil, July 1994, pp. 15-20.

11. W. Pedrycz, R.R. Gudwin, F.A.C. Gomide. “Nonlinear Context Adaptation in the Calibration of Fuzzy Sets” - *Fuzzy Sets and Systems*, **88**, 91-97 (1997).

12. E.M. Roth, E.J. Shoben - "The Effect of Context on the Structure of Categories" *Cognitive Psychology* **15**, 346-378 (1983).

13. D.L. Medin, E.J. Shoben - "Context and Structure in Conceptual Combination" *Cognitive Psychology* **20**, 158-190 (1988).

14. D.W. Massaro - "Testing between the TRACE Model and the Fuzzy Logical Model of Speech Perception" - *Cognitive Psychology* **21**, 398-421 (1989).

15. J.L. McClelland - "Stochastic interactive Processes and the Effect of Context on Perception" - *Cognitive Psychology* **23**, 1-44 (1991).

16. D.W. Massaro - "Integration versus Interactive Activation: The Joint Influence of Stimulus and Context in Perception" - *Cognitive Psychology* **23**, 558-614 (1991).

17. R.R. Gudwin - *A Kernel for Real-Time Knowledge Processing* - M.S. Thesis, Campinas State University, Unicamp (1992) (in portuguese).

18. T. Kohonen - *Self-Organization and Associative Memory* Berlin, Germany: Springer-Verlag (1988).

19. R.P. Lippman - "An Introduction to Computing with Neural Nets" - *IEEE ASSP Magazine*, 4(2), 4-22 (1987).

20. D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning* - Addison-Wesley Publishing Company, Inc. 1989.

Acknowledgement: The second author is grateful to CNPq, the Brazilian National Research Council, for research grant # 300729/86-3.