Intelligent Control Systems (0640734)

Lecture (10)

Genetic Algorithms in Control

Prof. Kasim M. Al-Aubidy
Philadelphia University-Jordan

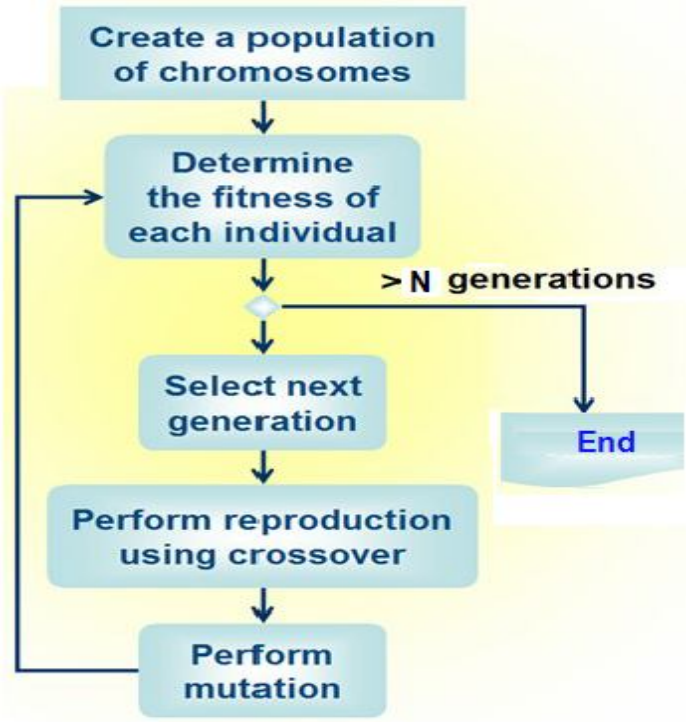
GA Design Outline:

The following points must be considered to solve any optimization problem using GA;

1. A genetic representation of candidate solutions,
2. A way to create an initial population of solutions,
3. An evaluation function which describes the quality of each individual,
4. Genetic operators that generate new variants during reproduction, and
5. Values for the parameters of the GA, such as population size, number of generations and probabilities of applying genetic operators.

The GA Flowchart:

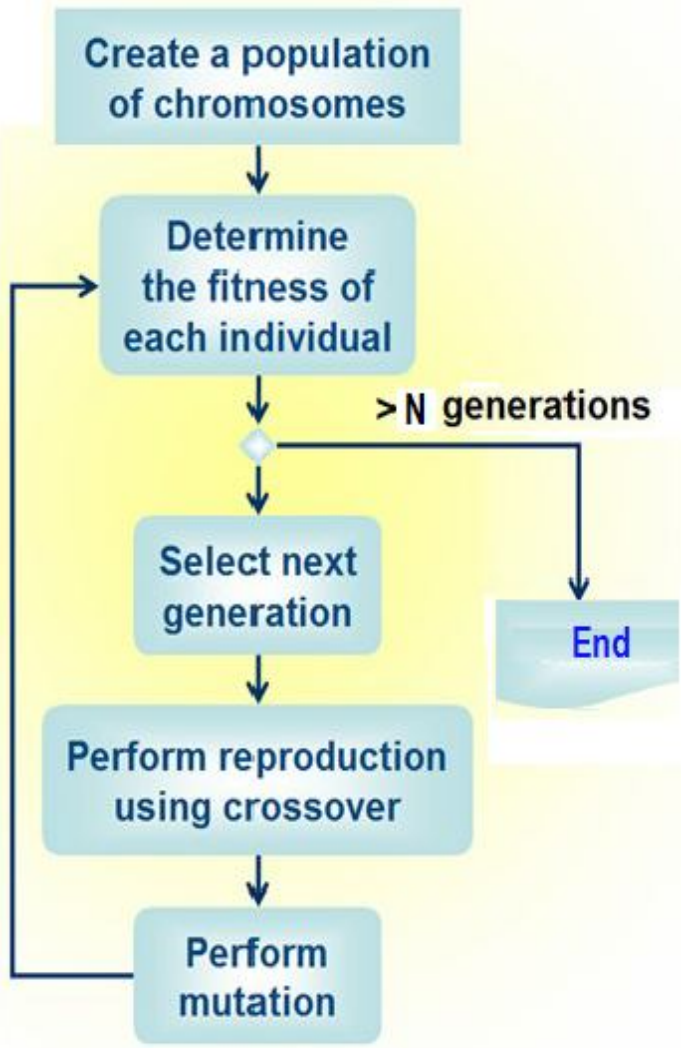
1. **Start Step:** Generate random population of N chromosome, the individuals of this population represent the Possible solutions.
2. **Select Fitness Function:** Evaluate the fitness f(x) of each chromosome x in the population.



3. **New population:** Create a new population by repeating the following steps until the New population is complete;

- **Selection:** Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to get selected).
- **Crossover:** With a crossover probability, cross over the parents to form new offspring (children). If no crossover was performed, offspring is the exact copy of parents.
- **Mutation:** With a mutation probability, mutate new offspring at each locus (position in chromosome)
- **Accepting:** Place new offspring in the new population.

4. **Replace**] Use new generated population for a further sum of the algorithm.
5. **Test**] If the end condition is satisfied, stop, and return the best solution in current population.
6. **Loop**] Go to step2 for fitness evaluation



GA-Fuzzy Systems:

GA-fuzzy systems are in fact the most common evolution based fuzzy system.

GA is used as the optimization algorithm and name the resulting hybrid system as GA-fuzzy system.

Fuzzy expert knowledge can be divided into two basic components:

1. **The Domain knowledge;** is generally the conscious operating knowledge about a particular system such as the membership functions and the fuzzy rule set.
2. **The Meta knowledge;** is the unconscious knowledge that is also needed to completely define a fuzzy system such as the mechanism of executing the fuzzy rules, methods of implication, rule aggregation, and defuzzification.

Optimization Methods:

Most of the existing methods in evolutionary fuzzy systems attempt to optimize parameters of the domain knowledge only (namely membership functions and rule set) while ignoring the effect of meta knowledge.

There are 4 basic methods of optimization as follows;

1. Automatic optimization of membership functions while there is a fixed and known rule set;
2. Automatic selection of the rule set with fixed membership functions;
3. Optimization of both the membership functions and rule set in two steps. First selecting the optimal rule set with fixed known membership functions and then tuning the membership functions with the resulting rule set; and
4. Simultaneous optimization of fuzzy rule set and membership functions.

Note that the number of membership functions or rules can also be optimized in the algorithm. There may be various reasons for a method to be selected.
Some of those advantages and disadvantage are mentioned below:

1. Since the rule set and membership functions are codependent, they should be defined simultaneously. This can lead to more optimal solutions.
2. Since the performance of a fuzzy system is more dependent on fuzzy rules rather than membership functions, fine tuning of the fuzzy system is better possible by tuning of membership functions. So it seems that it is better first to select the optimal rule set (coarse tuning) and then tune the membership functions (third method).
3. Even though various methods exist to encode both the rule base and membership functions, such encoding can have several potential difficulties. In addition to the level of complexity and large number of optimization parameters, the problem of competing conventions may arise and the landscape may unnecessarily become multi-modal.

Design of Encoding Function:

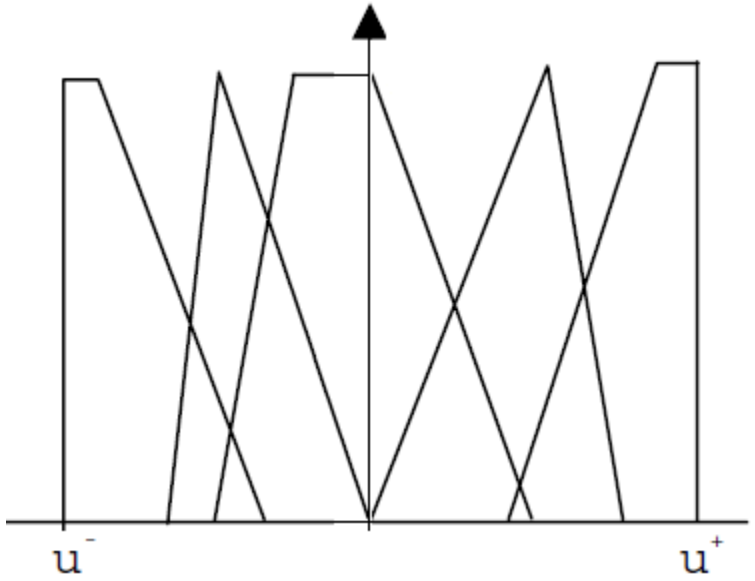
GA can optimize membership functions and rules of a fuzzy expert system.

Membership Functions Encoding:

Membership functions can be all or some part of a chromosome of the system. Their genetic representation is named MFC (membership function chromosome).

Every fuzzy set is defined by its type and shape;

- **Type of the membership functions:** triangular, trapezoidal, gaussian, ...
- **Shape of the membership function:** such as left base, center, base width, etc.



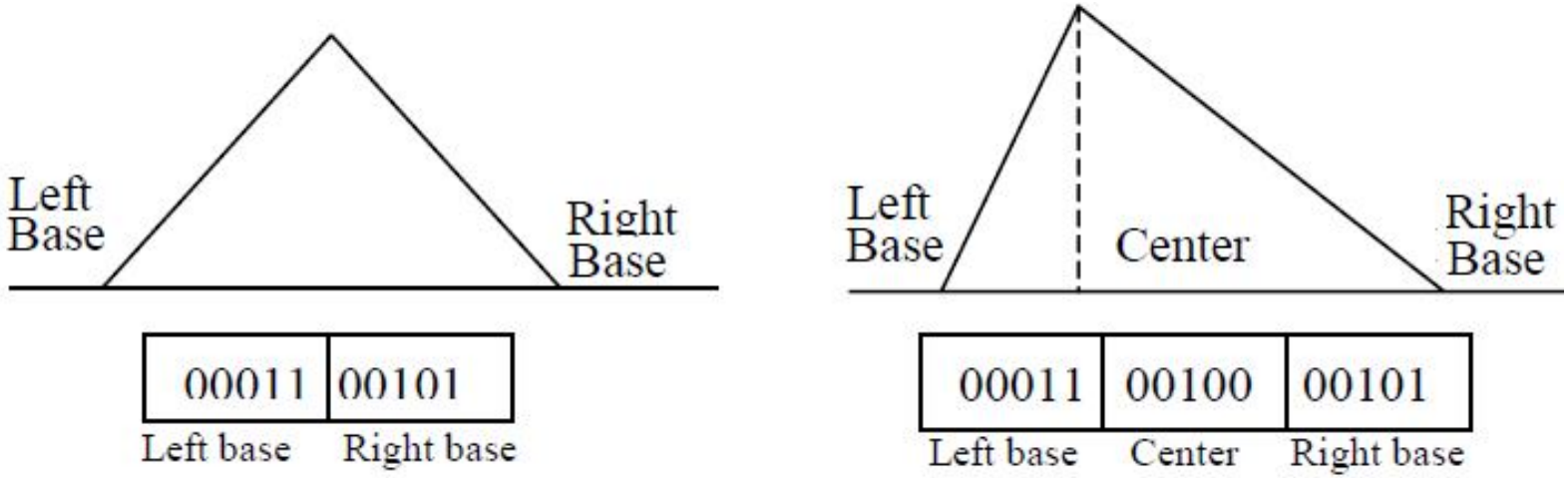
Thus the encoding problem is divided into two parts:

1. **Selection of free parameters:** the GA designer has to decide which parameters to fix and which parameters to tune.
2. **Encoding of Chosen Parameters:** Several methods exist for encoding MF parameters; among them, binary string encoding is the most common.

Triangular Membership Functions:

There are different types of coding methods for triangular membership functions as discussed below,

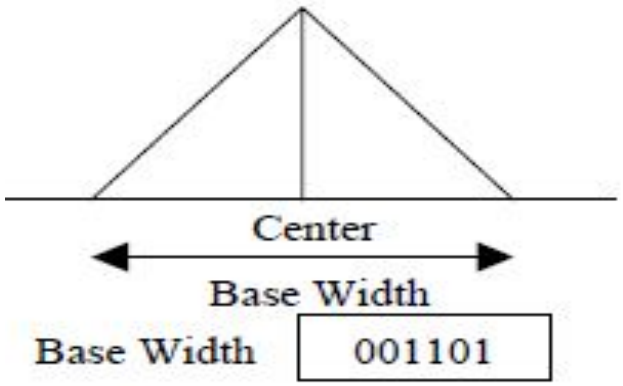
1. In this method a triangular membership function is defined by its three parameters: left base, center, and right base. A binary string MFC is developed as shown in the figure below where each parameter is encoded as a binary string.



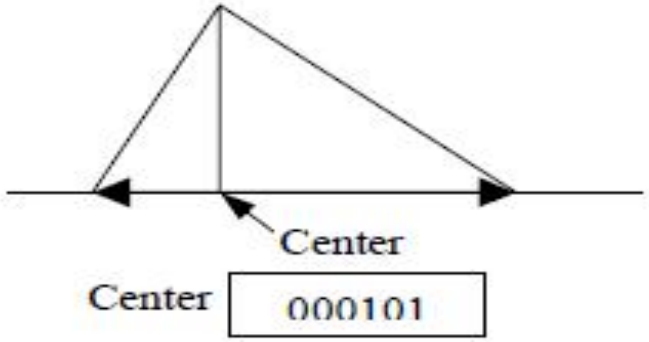
2. Symmetric triangular membership functions are assumed here; thus two parameters are sufficient to define a membership function, left base (starting point) and right base (ending point).

Coding Methods for Triangular MFs:

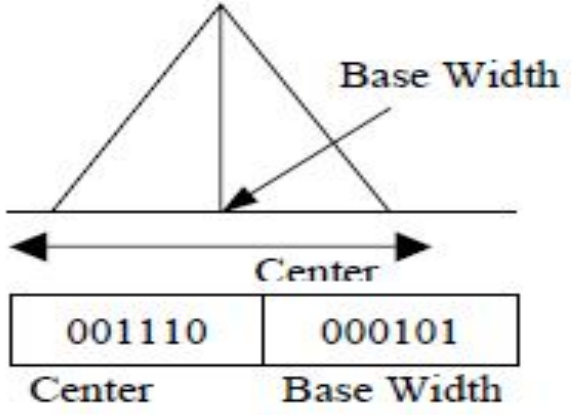
3. The triangular membership functions are symmetric and have fixed centers, only their base widths are tuned.



4. The triangular membership functions with fixed base width are assumed and only their centers are encoded and tuned. Thus there is only one free parameter.



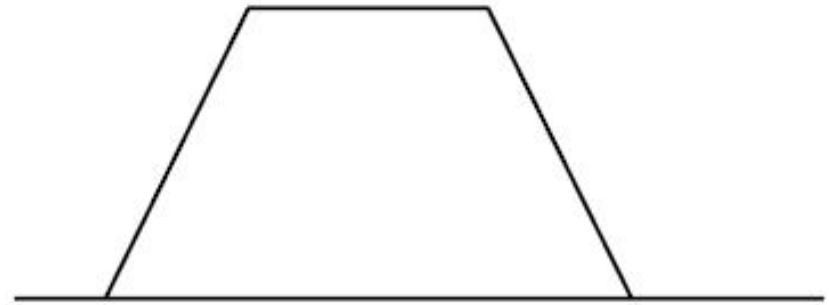
5. Symmetric triangular membership functions are assumed while their centers and widths are encoded and tuned, yielding two free parameters.



Non-Triangular Membership Functions:

To simplify the coding problem, use only symmetric membership functions and thus encode every membership function with three parameters; the type of the function, starting point, and ending point, with a fixed ratio of points in between starting and ending points.

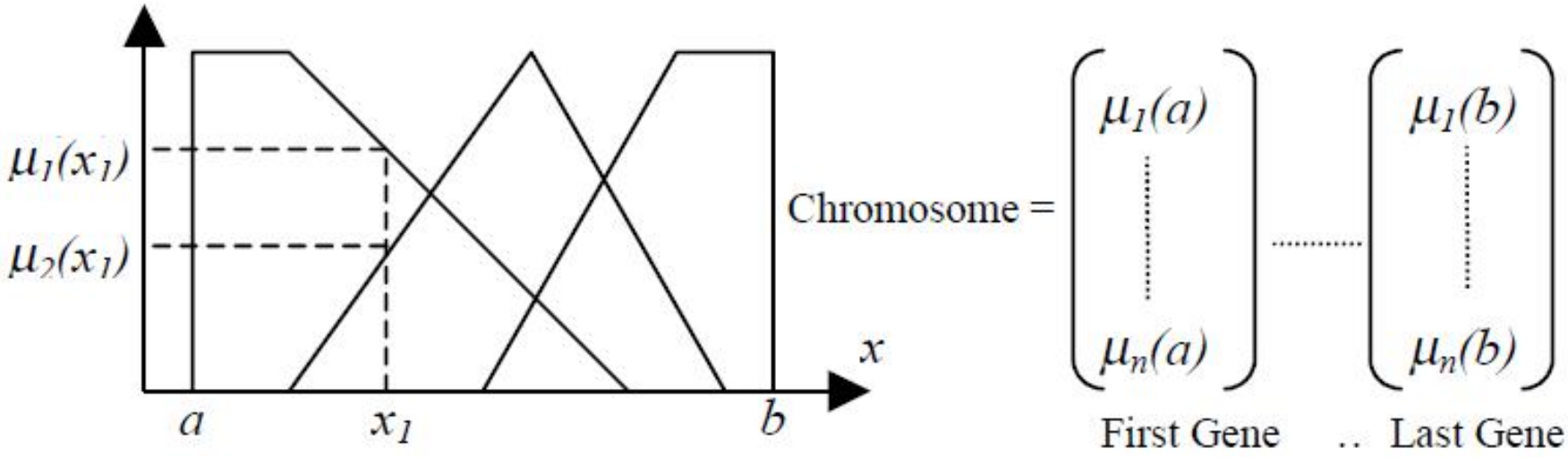
001	Triangular membership function
010	Trapezoidal membership function
011	Gaussian membership function
100	Sigmoidal membership function
…	…



010	00011	00101
Type	Starting Point	Ending Point

General Method of MF Encoding:

- All the MFs in a domain of a variable are encoded together in a matrix form.
- Every column of this matrix is a gene and is associated with a real value x.
- The gene is a vector having n elements, where n is the number of the MFs in that partition. Every element is the membership value of all MFs at x.
- In practice, if p points in the domain are considered and there are n MFs, then n*p parameters are encoded.
- This figure is a genetic representation of the domain where a and b are the starting and ending points of the domain.



Rule Encoding:

Rule set encoding can be more complicated than membership function encoding, since each rule is in competition with others for being selected in the rule set, the impact of each rule in the system is dependent on other rules that concurrently exist in the rule set. For example, two fuzzy rules may be highly fitted if they both exist in the rule set while neither of the two rules may be desirable separately.

Rule set optimization is widely used in fuzzy classification problems, which can be generally categorized as either the Michigan approach or the Pittsburgh approach.

1. The Michigan approach:

Every individual in the GA is a fuzzy rule encoded as a string with fixed length. The GA operates on the individual rules and more fit rules are combined together via genetic operators to create the next population of rules. The fitness function is designed so as to show the fitness of one rule. Its disadvantage is the problem of competing convention.

2. The Pittsburgh approach:

Every individual in the GA is a fuzzy rule set encoded as a string with variable length. Fitness function, therefore, operates on the rule sets and higher fit rule sets are combined via genetic operators to produce rule sets with higher fitness. This method is more desirable because of the competing convention problem.

Example:

Consider the following rules;

If (x1 is IMF11 & x2 is IMF21) then y is OMF2

If (x1 is IMF11 & x2 is IMF22) then y is OMF10

If (x1 is IMF12 & x2 is IMF22) then y is OMF7

	IMF_{11}	IMF_{12}	IMF_{13}	IMF_{14}
IMF_{21}	OMF_2	OMF_1	OMF_{13}	OMF_{14}
IMF_{22}	OMF_{10}	OMF_7	OMF_9	OMF_5
IMF_{23}	OMF_3	OMF_{11}	OMF_8	OMF_4
IMF_{24}	OMF_{16}	OMF_6	OMF_{12}	OMF_{15}

Matrix of Indices:

Every fuzzy rule set is defined with Rmax free parameters. These parameters are indices that represent an output MF among q MFs in OSET.

Use only indices of the MFs and build a matrix of indices named P.

$P=[p_{kl}]=[p(k,l)], \quad p_{kl} \in Z, \quad 0 \leq p_{kl} \leq q$

p_{15}	p_{14}	p_{13}	p_{12}	p_{11}
p_{25}	p_{24}	p_{23}	$\boldsymbol{p_{22}}$	p_{21}
p_{35}	p_{34}	$\boldsymbol{p_{33}}$	$\boldsymbol{p_{32}}$	$\boldsymbol{p_{31}}$
p_{45}	p_{44}	p_{43}	$\boldsymbol{p_{42}}$	p_{41}
p_{55}	p_{54}	p_{53}	p_{52}	p_{51}

Genetic Representation

Having the indices array of the rule set, **P**, it is possible to use either string or array representation of the rule set.

String Representation:

String representation of the rule set table can be obtained in two steps:

Step 1: encoding all the elements of matrix P. Binary encoding is a common method to encode the parameters. S is the resulting matrix after encoding.

$$S_{kl}=\text{Decimal\_to\_Binary}(p_{kl})$$

Step 2: Obtain the string representation of the table using the rows of the **S** matrix as;

Chromosome = s11 s12 s13.s1m1 s21 s22 s23.s2m1. . . sm21sm22 .sm2m1

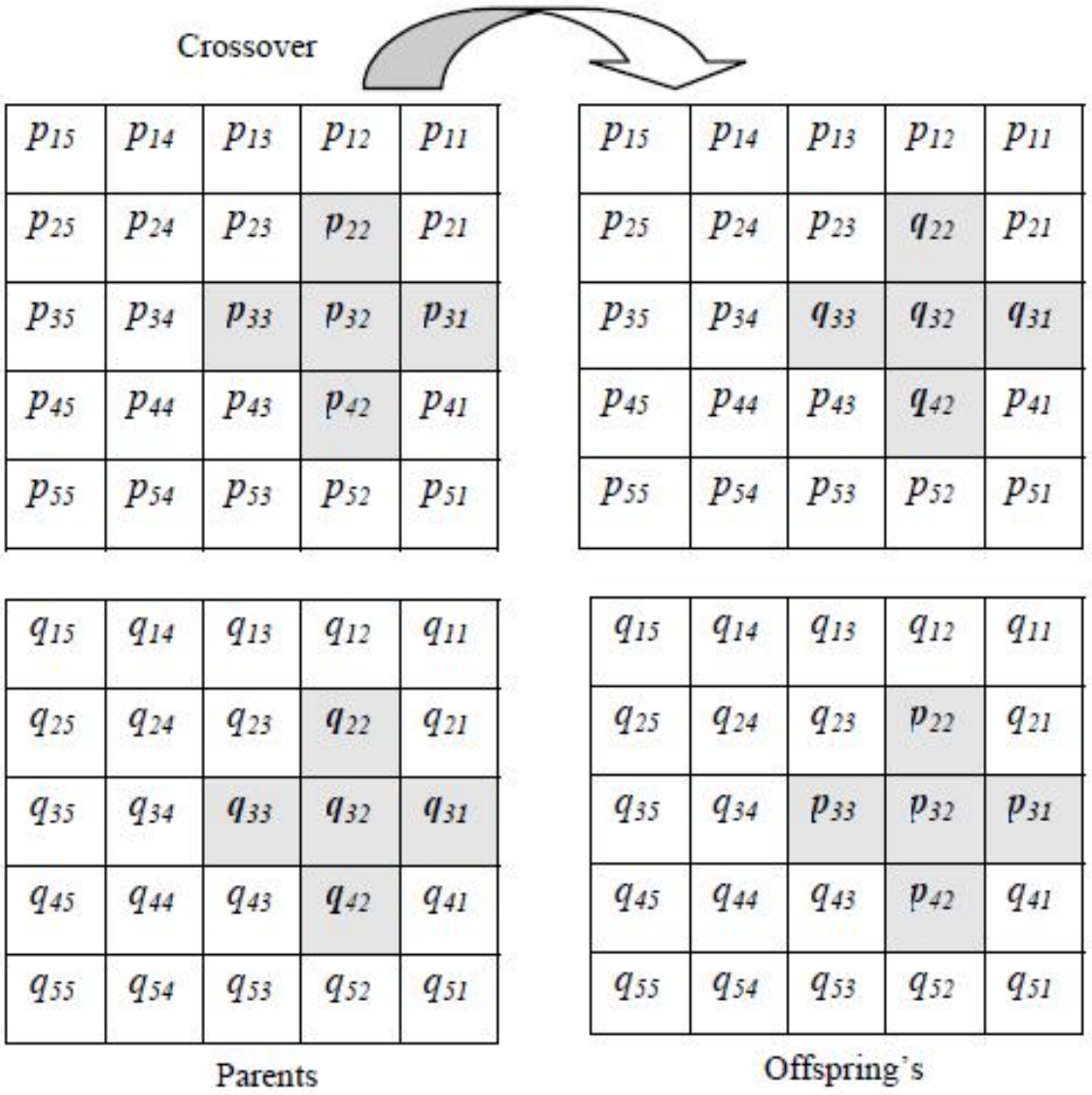
Number of the bits in every chromosome will be $N = (\prod_{i=1}^{n} m_i) * K$

where K is the number of the bits in every element of S (s_{kl})

The genetic operators in this type of representation can be the same as the standard genetic algorithms.

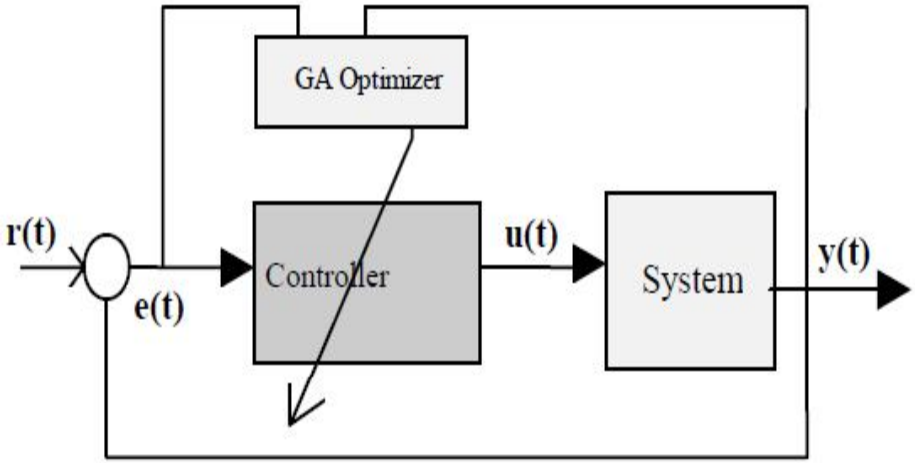
Matrix Representation:

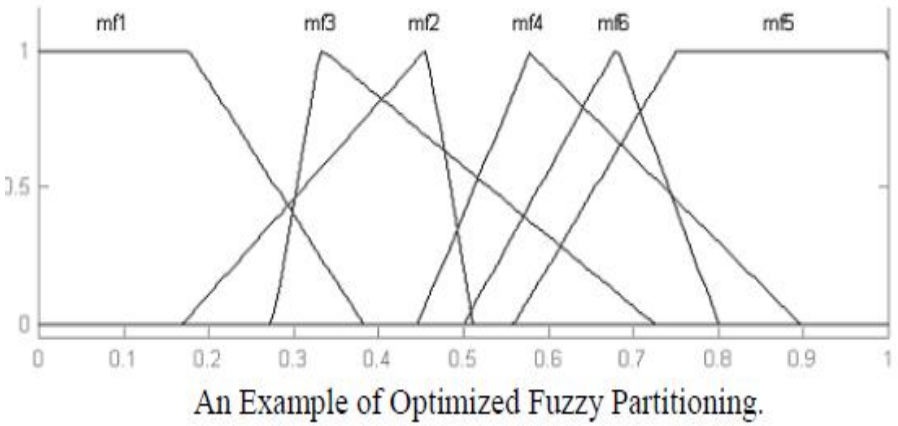
A matrix chromosome and a set of genetic operators are needed to operate on this chromosome. In this method, a Point-Radius crossover is used where each crossover is determined with a circle with known center and radius. The region in the table that is surrounded by the circle is exchanged with the similar region in the other chromosome



Designing Fuzzy Controllers using Genetic Algorithms

Ref: Tan, G. V. and Hu, X., On Designing Fuzzy Controllers using Genetic Algorithms, *IEEE Int. Conf. on Fuzzy Systems, 905−911, 1996.*



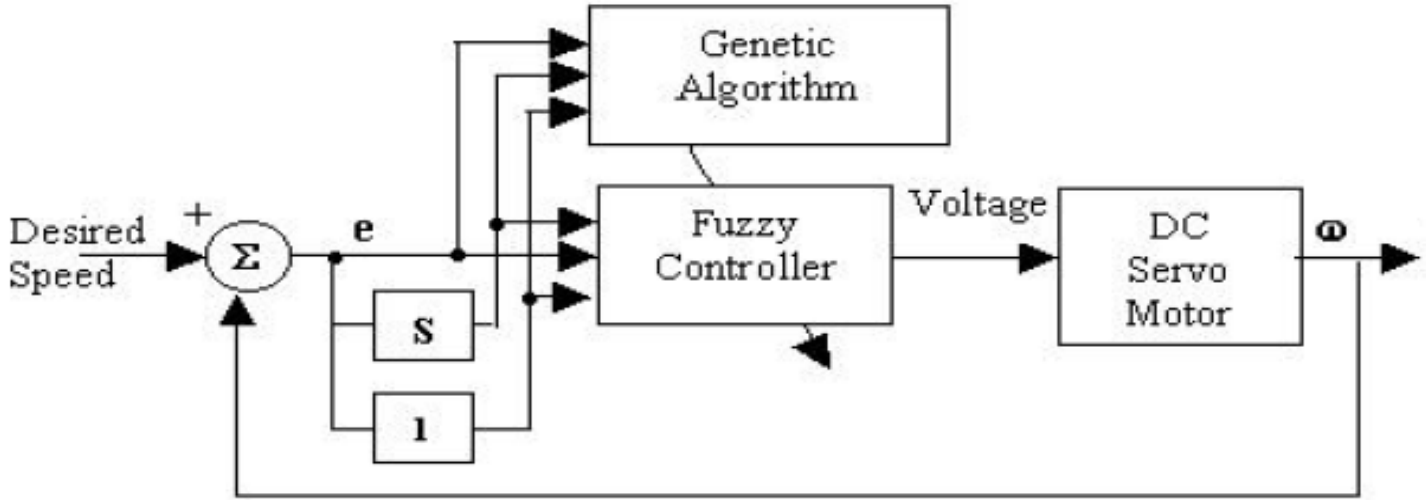


An Example of Optimized Fuzzy Partitioning.

This paper presents some of the undesired side effects that can surface when using evolutionary fuzzy systems. The main problem is that evolutionary optimization algorithms may alter the control system architecture to the extent that the resulting system would no longer be identifiable by a human operator. This is particularly true for fuzzy systems where an intuitive human understanding of the control system is a significant aspect of a control system. Indeed, the meaningful relation between MFs and associated fuzzy rules may be lost from a human point of view.

SPEED REGULATION OF A DC MOTOR:

Ref: Akbarzadeh-T, M. R. et al., Evolutionary Fuzzy Speed Regulation for a DC Motor, *29th Southeastern Symp. on Syst. Theory, Cookeville,* TN, March 1997.

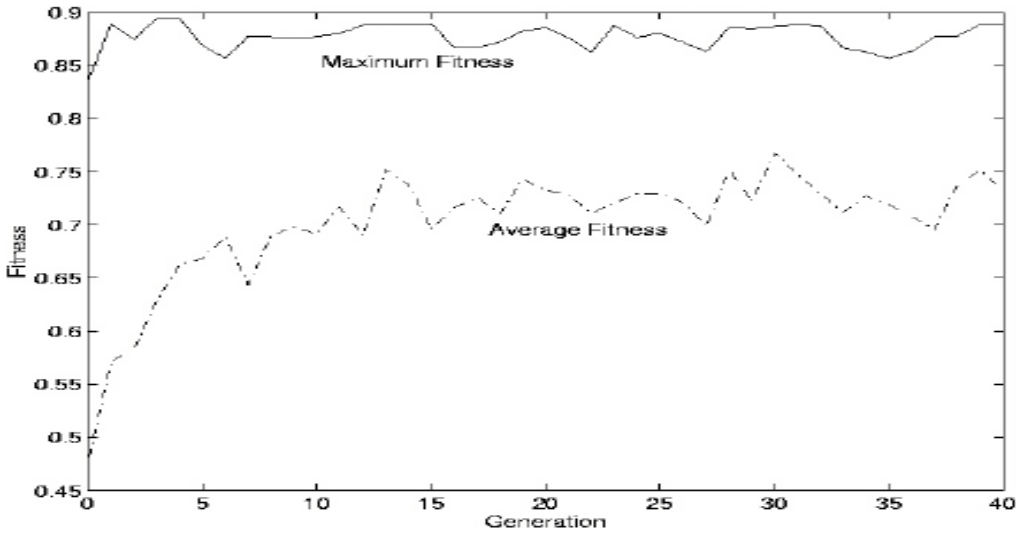


- Through the GA, various combinations of candidate solutions are evaluated and the best, fittest, solution is chosen to control the actual system.
- The GA has the capability to alter the shape of the MFs of individual inputs.
- The following fitness function was used to evaluate various individuals within a population of potential solutions:

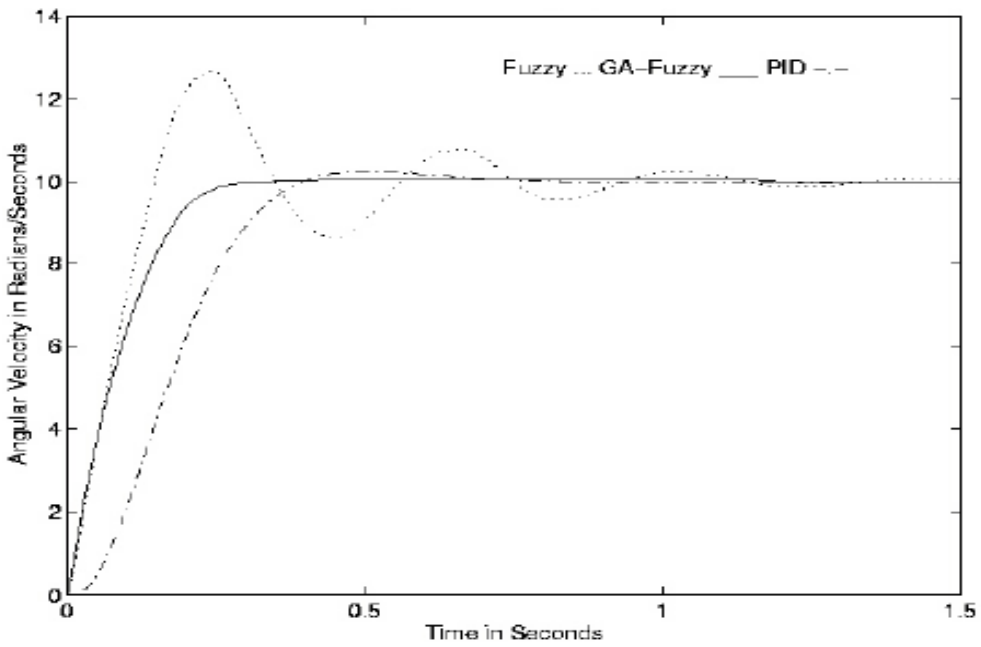
$$fitness = \frac{1}{t_f - t_i} \int_{t_i}^{t_f} \frac{1}{\left(k_1 e^2 + k_2 \dot{e}^2 + k_3 \gamma^2 + 1\right)} dt$$

Results:

A total of 40 generations were simulated; each generation included 100 individuals
The performance measure never reaches steady state since it is constantly trying out new directions of search through mutation.



The curve for the maximum fitness converges very quickly and within the first two generations. However, the fitness of the whole population converges within 20 generations.
The mutation rate for creating the initial population was set to 0.1.
The mutation rate= 0.033.
The probability of crossover= 0.6.



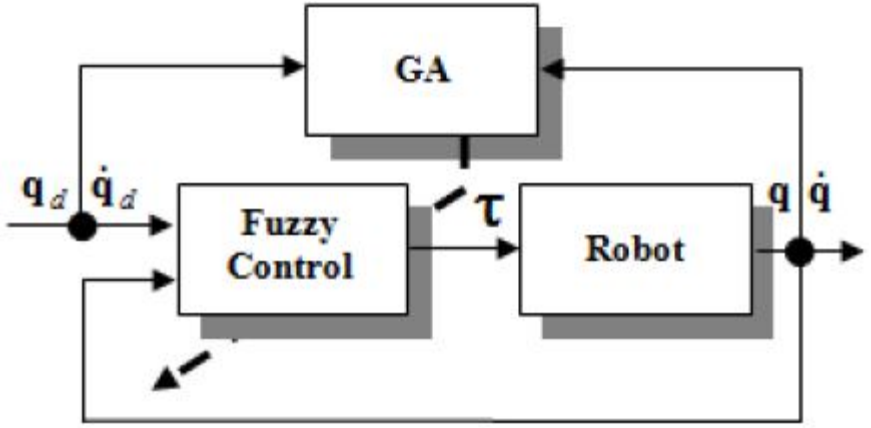
FUZZY CONTROL SYSTEMS BASED ON GAs:

Ref: Gyula Mester, "DESIGN OF THE FUZZY CONTROL SYSTEMS BASED ON GENETIC ALGORITHM FOR INTELLIGENT ROBOTS", *Interdisciplinary Description of Complex Systems 12(3), 245-254, 2014.*

- Optimization of fuzzy control systems based on GA in the MATLAB environment.
- The GAs are applied for fuzzy rules set, scaling factors and membership functions optimization.
- The fuzzy control structure initial consist of the 3 membership functions and 9 rules for the 4 DOF SCARA Robot control.
- Fuzzy controller with the generalized bell membership functions can provide better dynamic performance of the robot than with the triangular membership functions.

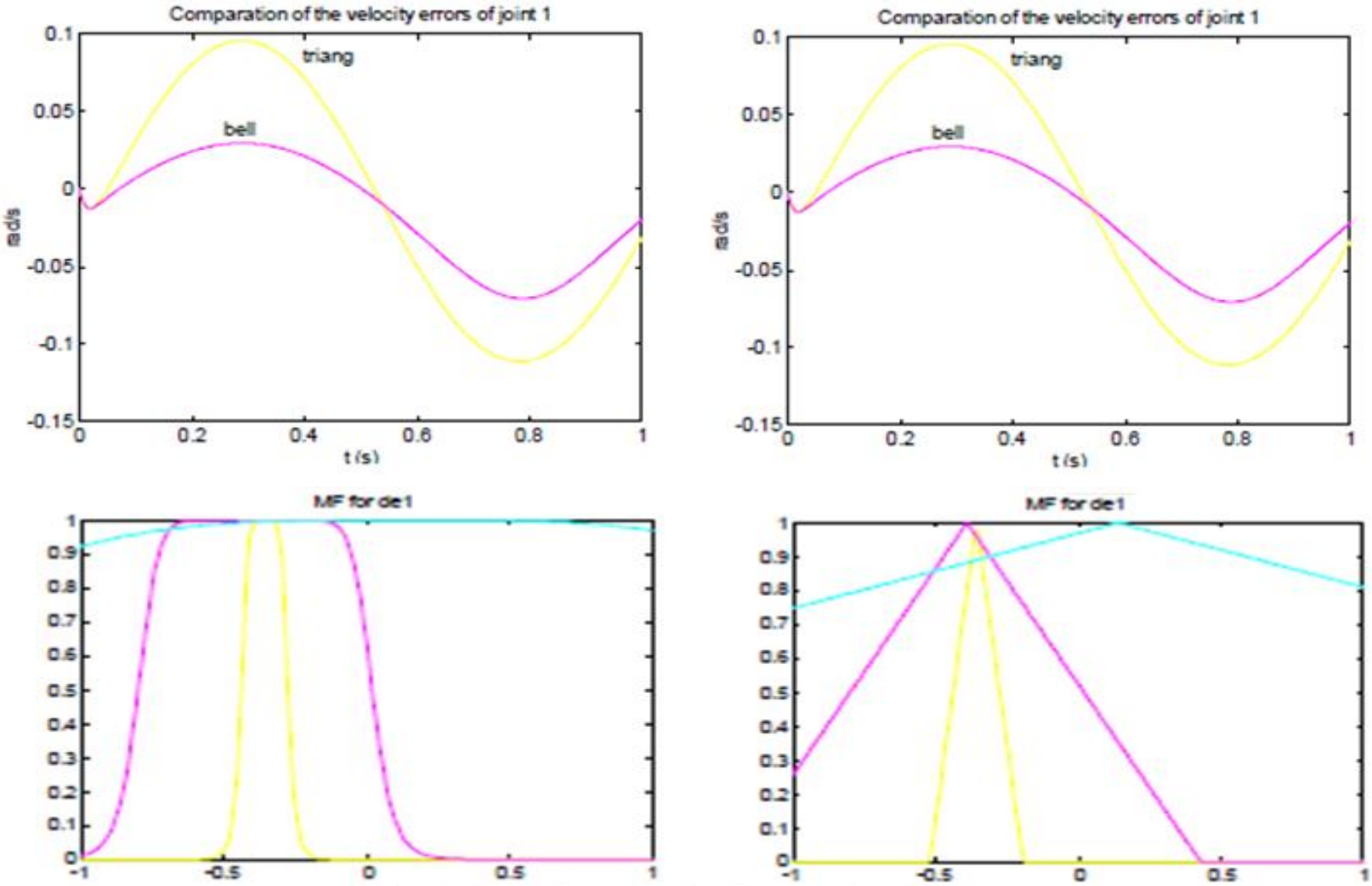
The generalized bell membership function
Depends on three parameters [a, b and c]:

$$f(x;,a,b,c) = \frac{1}{1+\left|\frac{x-c}{a}\right|^{2b}}$$



- The GA is applied for fuzzy rules set, scaling factors and MFs optimization. The GA starts by randomly generating initial population of strings. The generation, crossover probability and mutation probability are 200, 0,8 and 0,01, respectively.
- Generalized bell and triangular-shaped MFs can be parameterized by 3 parameters.
- The chromosome of the simple GA for each robot joints includes two parts:
  - ✓ 3* 9 = 27 consequent variables on the fuzzy control rule base (9 rules) and
  - ✓ 2*3*3=18 parameters of the MFs (3 MFs for e and 3 MFs for de).
- The numbers 0, 1, 2, 3 and 4 on the fuzzy control rule table represent the linguistic values (NB, NS, ZE, PS and PB).
- The 27 genes in the chromosome are the elements of the control rule table.
- The fitness function is composed of the error and the error rate of the systems step response for each joint:

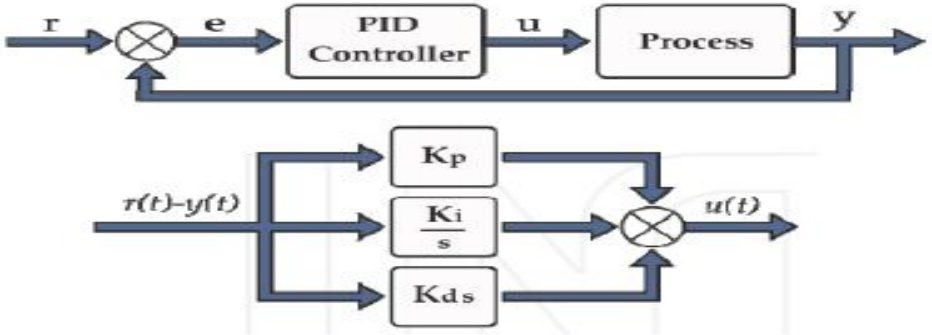
$$\text{Fitness} \leftrightarrow -\sum_{i=1}^{4}\left[k(q_{di}-q_i)^2+(dq_{di}-dq_i)^2\right]$$



Optimized bell membership functions for de_1.

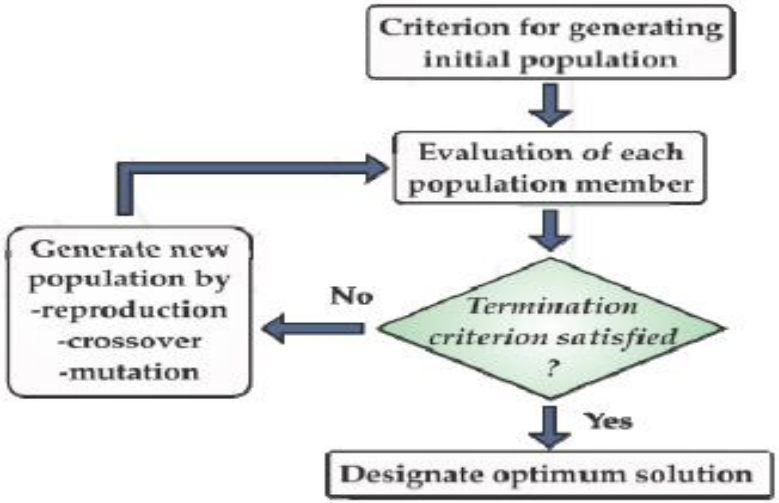
PID-Controller Tuning Optimization with GA:

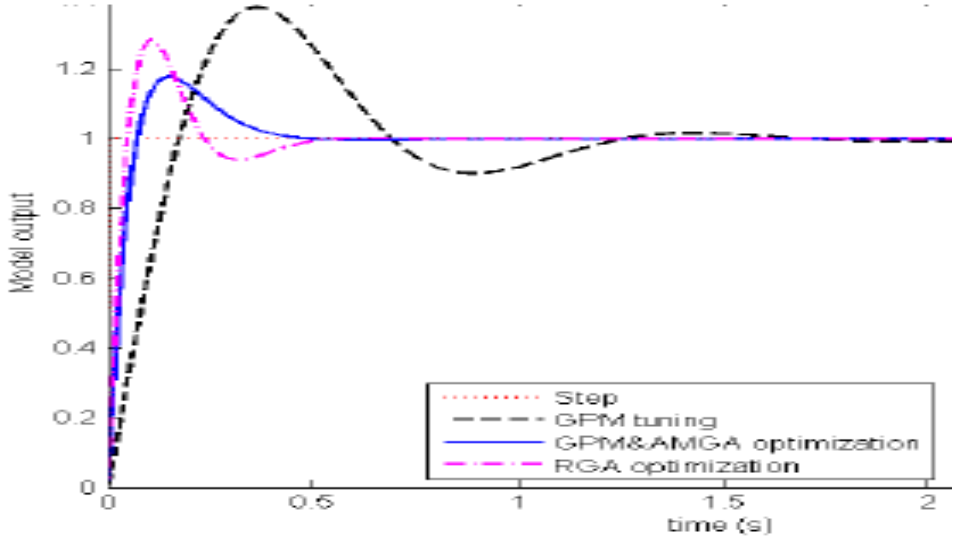
Ref: Arturo Y., et. Al., "PID-Controller Tuning Optimization with Genetic Algorithms in Servo Systems", International Journal of Advanced Robotic Systems, 2013, Vol. 10, 324:2013.



Equivalence of parameters for PID controllers.

$C(z)$	a_0	a_1	a_2	b_1	b_2
$C(s)$	$K_p + K_i + K_d$	$-(K_p + 2K_d)$	K_d	1	0





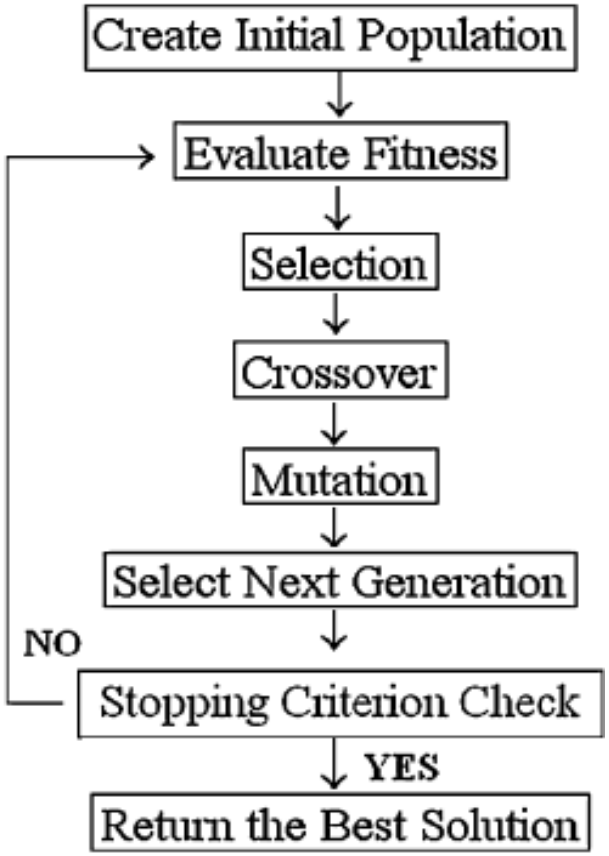
Tuning method	OS (%)	ISE Counts	MSE (%)
CNC Lathe			
GPM-AMGA	6.9	3.7470 x10⁹	0.6660
RGA	19.28	4.1936 x10⁹	0.7452
GPM	27.3	4.9102 x10⁹	0.8729
CNC Milling Machine			
GPM-AMGA	5.65	3.7436x10⁹	0.6649
RGA	29.26	3.6888 x10⁹	0.6553
GPM	13.0	4.3736e x10⁹	0.769
Industrial PUMA Robot			
GPM-AMGA	17	4.5668x10⁷	0.0081
RGA	18.16	4.7432x10⁷	0.0084
GPM	37.16	5.0649x10⁷	0.0090

Genetic Algorithms for the Temperature Control of a Batch Reactor:

Ref: Javier Causa, et.al., "Hybrid fuzzy predictive control based on genetic algorithms for the temperature control of a batch reactor", Computers & Chemical Engineering, Vol.32, No.12, 2008, pp.3254–3263.

The optimization based on a GA can be described by the following steps:

(1) Initialize a random population of individuals corresponding to the feasible solutions.
(2) Evaluate the objective function for each individual of the current population.
(3) Select random parents from the current population.
(4) Apply genetic operators like *crossover* and/or *mutation* to the parents, for a new generation.
(5) Evaluate the objective function for all the individuals of the generation.
(6) Choose the best individuals according to the best values of the objective function.
(7) Replace the weakest individuals of the previous generation with the best ones of the new generation obtained in Step 6.
(8) If either the value of the objective function reaches a certain tolerance or the maximum number of generations is reached, then the algorithm stops. Otherwise, go to Step 2.



The procedure for the HFPC-GA consists of:

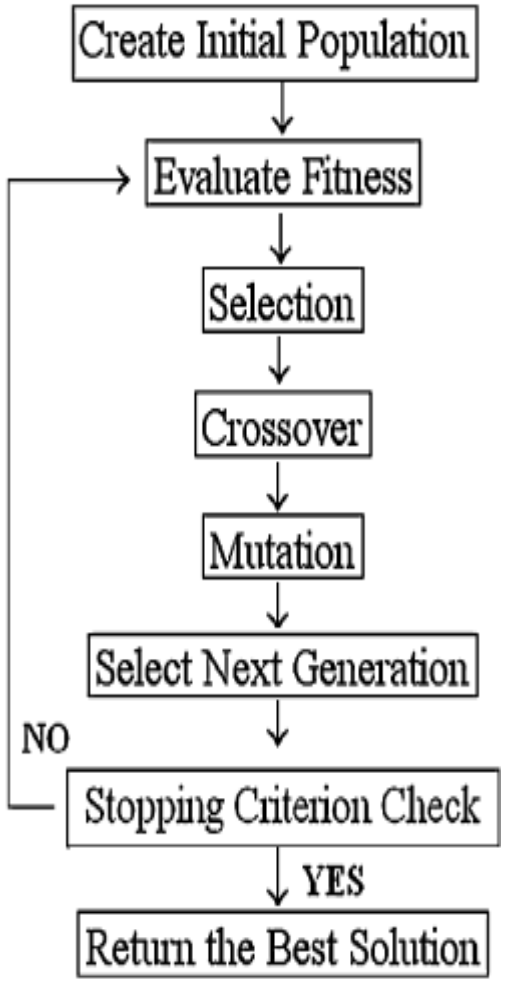
(1) Initialize a random population of individuals, i.e., create random-integer feasible solutions of manipulated variables for the hybrid fuzzy predictive control problem. As an example, the size of the population could be seven individuals per generation. Then, as the control horizon is 4, there are 7^4 possible individuals. However, for the GA per generation, the following population is considered:

$$\text{Population}_i = \begin{bmatrix} \text{individual}_1 \\ \text{individual}_2 \\ \text{individual}_3 \\ \text{individual}_4 \\ \text{individual}_5 \\ \text{individual}_6 \\ \text{individual}_7 \end{bmatrix} = \begin{bmatrix} 0163 \\ 2100 \\ 5423 \\ 3634 \\ 4131 \\ 2543 \\ 0301 \end{bmatrix}$$

(2) Evaluate the fitness function for all the initial individuals of the population using Eq. (11). Note that the prediction $\hat{y}(k+h)$ is calculated recursively by using the future control action. In general

$$\hat{y}(k+h) = f(\hat{y}(k+h-1), \ldots, u(k+h-1), \ldots)$$

where f is a non-linear function defined by a hybrid fuzzy model.



(3) Select random parents from the population (different vectors of the future control actions). For example, Individual 1 and Individual 5 are chosen as the parents:

$$\underbrace{\overbrace{[0\ 1]}^{1A}\ \overbrace{[6\ 3]}^{1B}}_{individual_1} \qquad \underbrace{\overbrace{[4\ 1]}^{5A}\ \overbrace{[3\ 1]}^{5B}}_{individual_5}$$

(4) Apply crossover and mutation to the parents in order to generate an offspring.

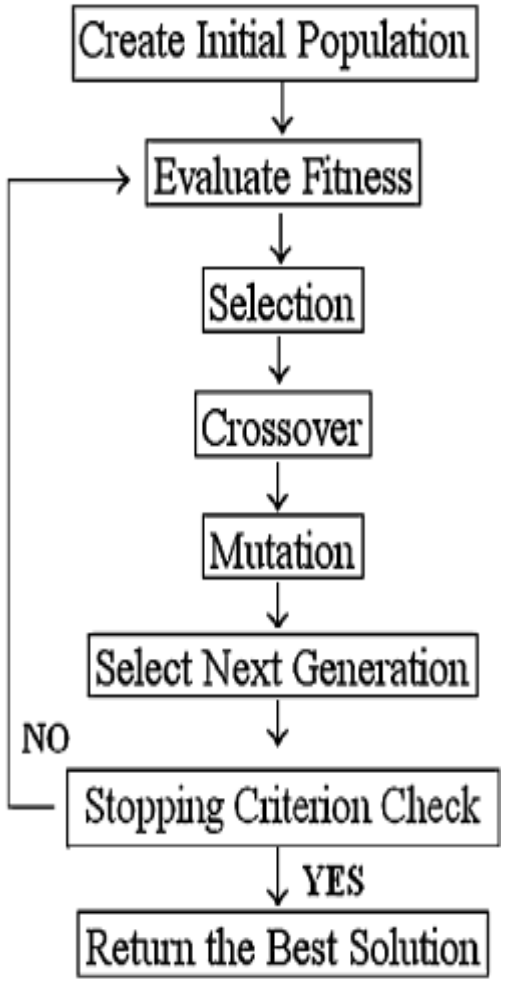
After the crossover step

$$\underbrace{\overbrace{[0\ 1]}^{1A}\ \overbrace{[3\ 1]}^{5B}} \qquad \underbrace{\overbrace{[4\ 1]}^{5A}\ \overbrace{[6\ 3]}^{1B}}$$

After the mutation step

$$\overbrace{[0\ 1\ 2\ 1]}^{New\ Individual_1} \qquad \overbrace{[0\ 1\ 6\ 3]}^{New\ Individual_2}$$

(5) Evaluate the fitness given by the objective function (11) of all the individuals of the offspring population.
(6) Select the best individuals according to the objective function.
(7) Replace the weakest individuals from the previous generation with the strongest individuals of the new generation selected in step 6.
(8) If the objective function value reaches the defined tolerance or the maximum generation number is reached (stopping criteria), then stop. Otherwise, go to step 2.

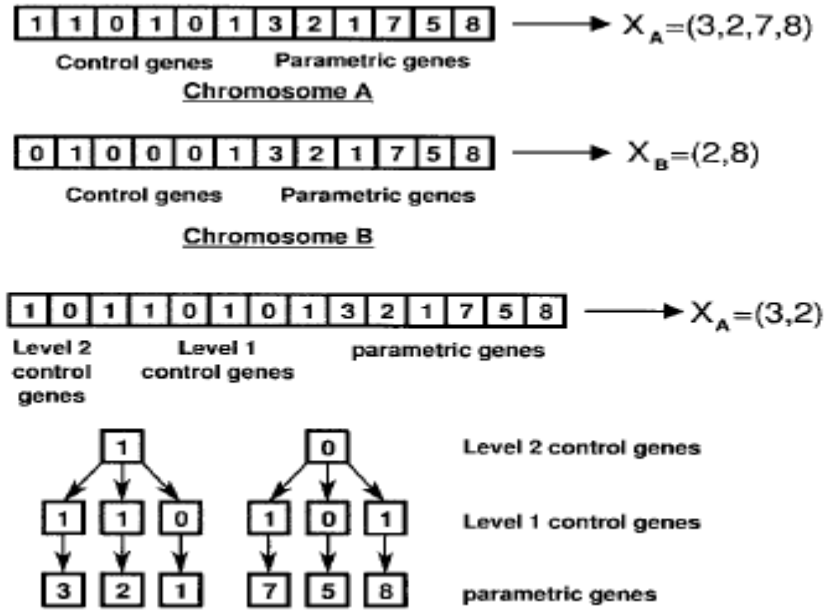


Fuzzy Memberships and Rules Using Hierarchical GAs:

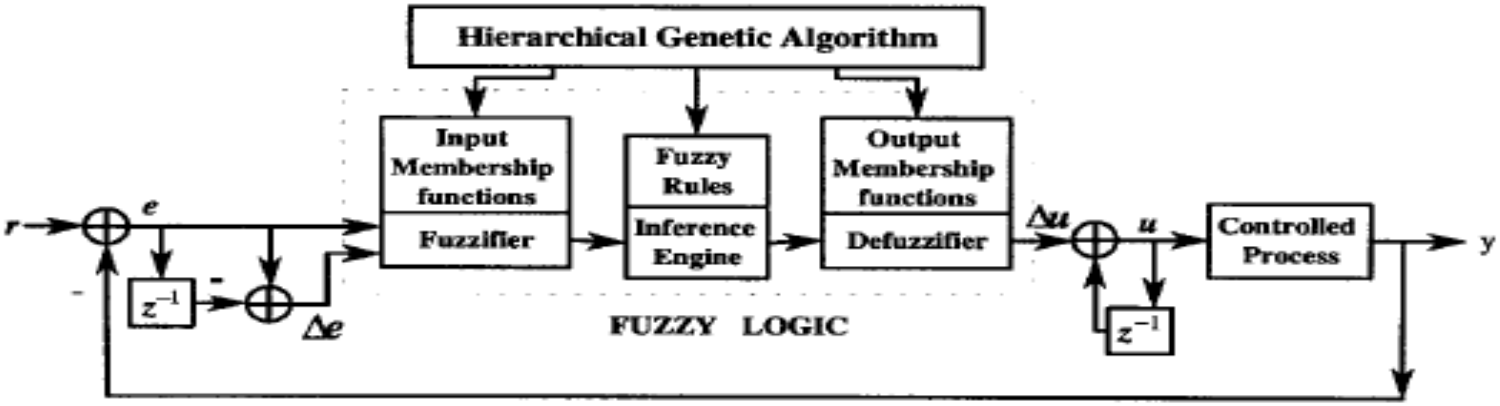
Kit-sang Tang, *Kim-fung Man, Zhi-feng Liu, and Sam Kwong,* "Minimal Fuzzy Memberships and Rules Using Hierarchical Genetic Algorithms", IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 45, NO. 1, FEBRUARY 1998.

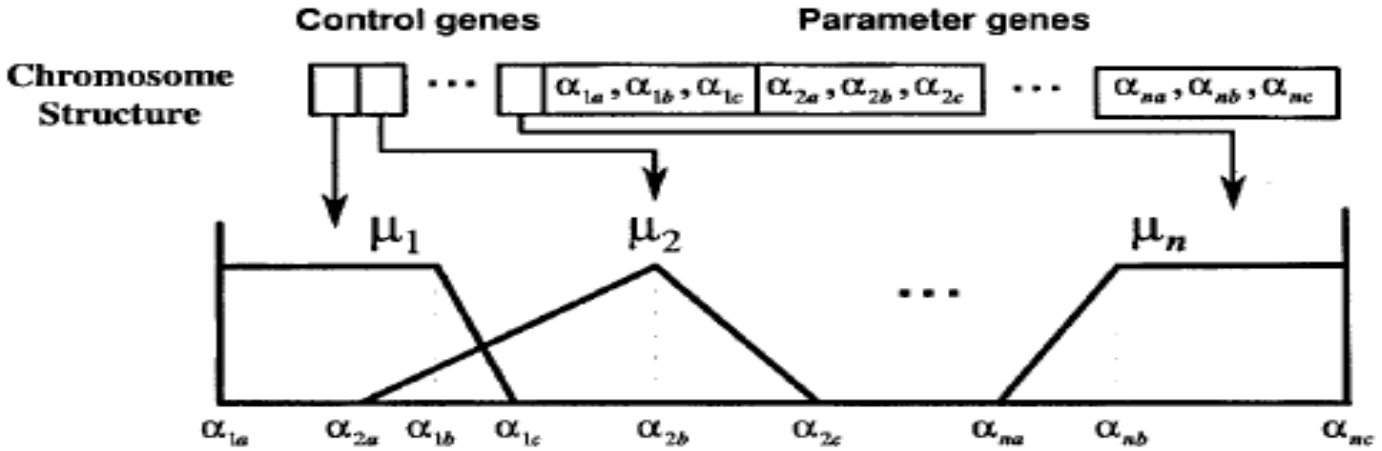
Optimal fuzzy subsets and rules using GAs is proposed. The genes of the chromosome can be arranged in a hierarchical form, where one type of genes controls the other type of genes.
The effectiveness of this genetic formulation enables the fuzzy subsets and rules to be optimally reduced, then the system performance is well maintained.
The paper covers all procedures for coding the fuzzy membership function and rules into the chromosome.
The proposed scheme is applied to control a constant water pressure pumping system.

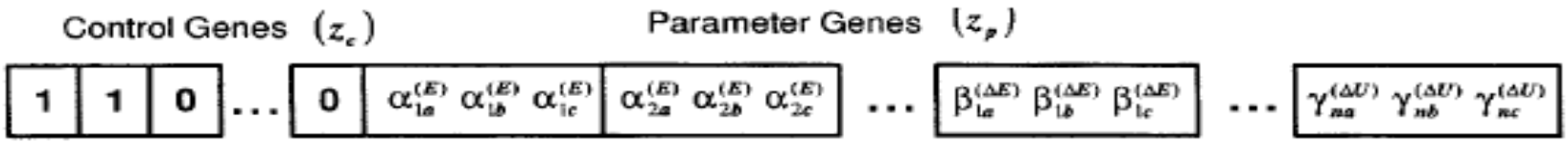
Example:
Consider a chromosome formed with 6-b control genes and 6-integer parametric genes.
The lengths of X_A and X_B are 4 and 2, respectively, which means that the phenotype in different lengths is available within the same chromosome formulation.
The HGA will search over all possible lengths, including the parameters to met the final objective requirement.
The hierarchical levels can be increased within the Chromosome to formulate a multilayer chromosome.



Formulation of HGA Fuzzy Logic Control System:

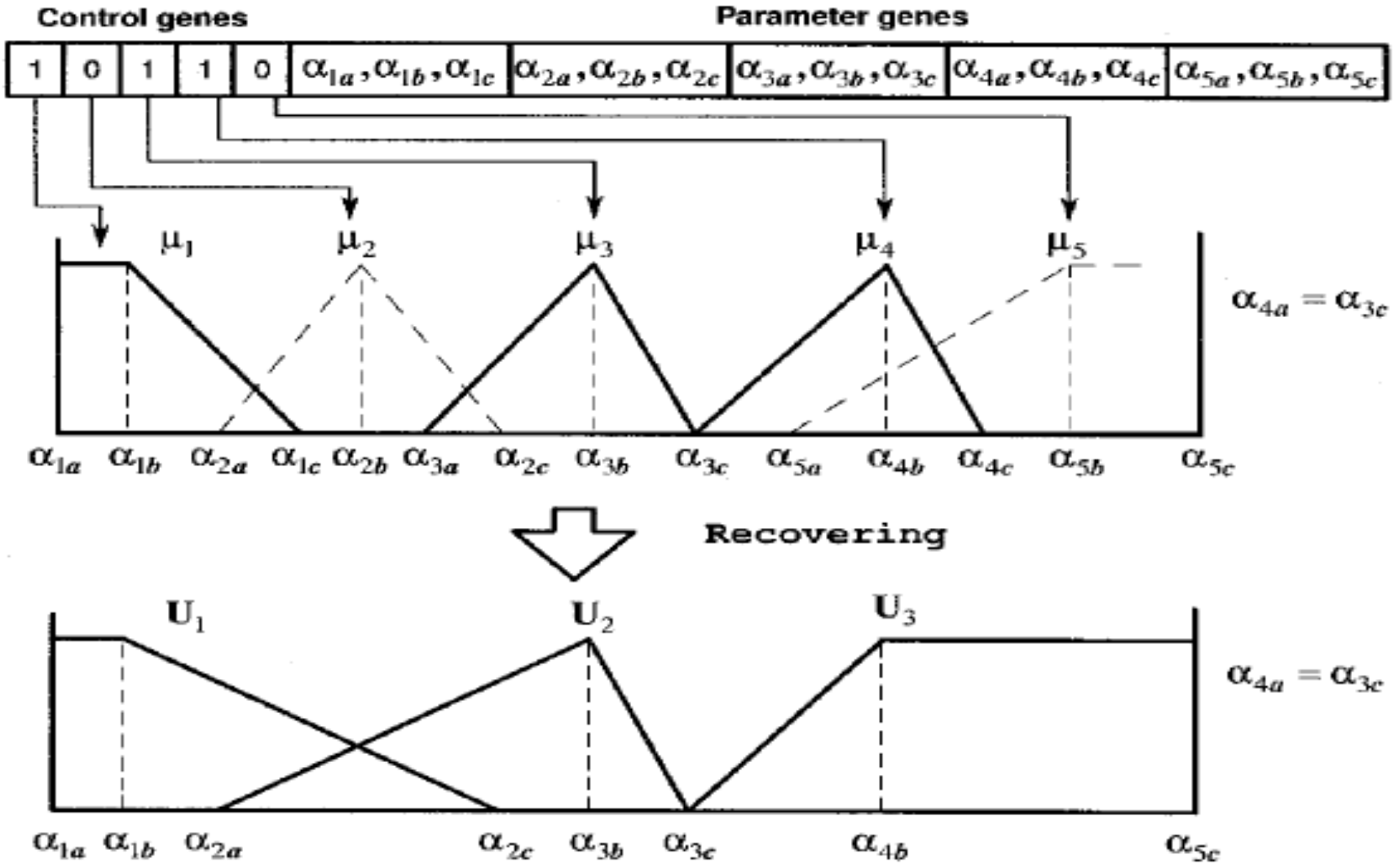






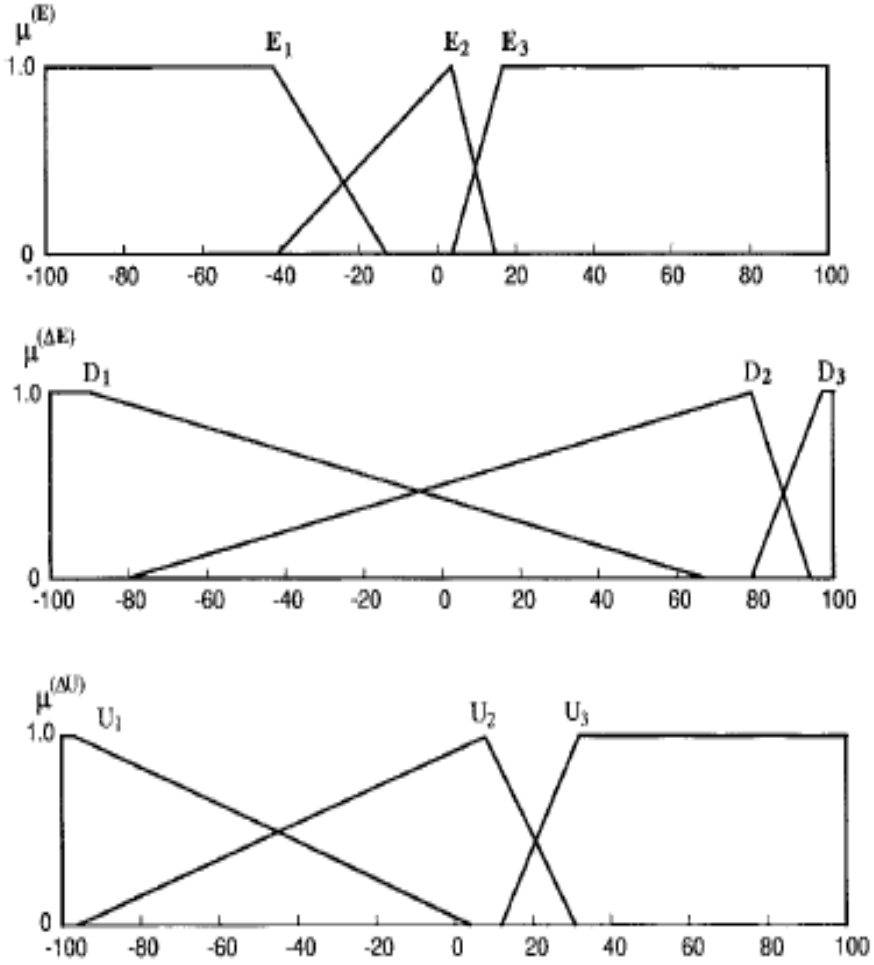
GA-Based MFs Tuning:

Recovery of invalid fuzzy membership functions.



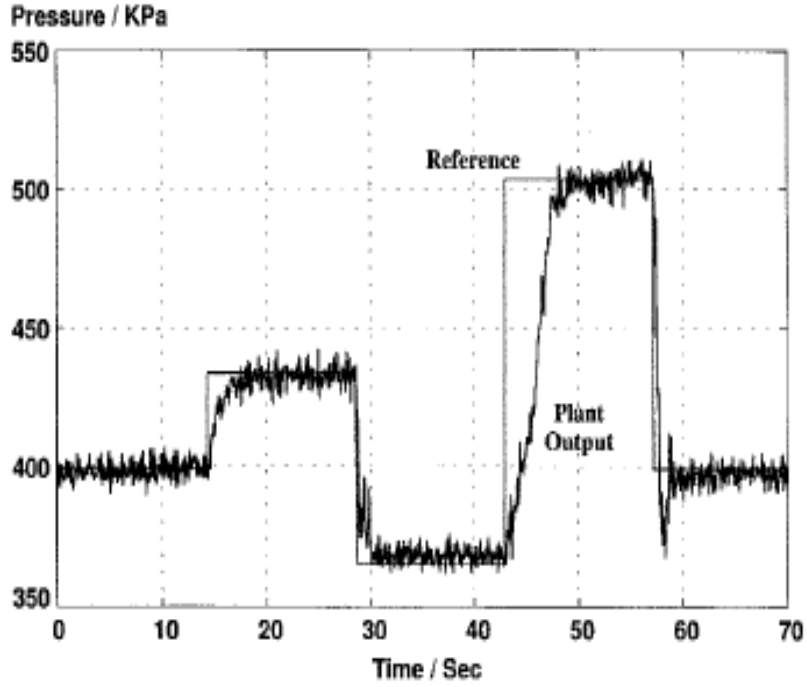
Results:

Final fuzzy subsets and membership function for water pump control system.



OPTIMAL RULE TABLE

		Error Rate Fuzzy Set D_1	D_2	D_3
Error	E_1	U_1	U_2	U_2
Fuzzy	E_2	U_2	U_2	U_3
Set	E_3	U_2	U_3	U_3



References:

1. S. Rajasekaran & G.A. Vijayalakshmi Pai, "Neural Networks, Fuzzy Logic, and Genetic Algorithms: Synthesis & Applications", ISBN: 81-203-2186-3, Prentice Hall, 2006.
2. A. Zilouchian and M. Jamshidi, "Intelligent Control Systems Using Soft Computing Methodologies", CRC Press, 2001.
3. S.N. Sivanandam and S.N. Deepa, *Introduction to Genetic Algorithms, Springer, New* York, , 2008.
4. Z. Michalewicz, "Genetic Algorithms and Data Structures: Evolution Programs", Springer-Verlag, 1999.
5. Tan, G. V. and Hu, X., On Designing Fuzzy Controllers using Genetic Algorithms, *IEEE Int. Conf. on Fuzzy Systems, 905−911, 1996.*
6. Akbarzadeh-T, M. R. et al., Evolutionary Fuzzy Speed Regulation for a DC Motor, *29th Southeastern Symp. on Syst. Theory, Cookeville,* TN, March 1997.
7. Ref: Gyula Mester, "DESIGN OF THE FUZZY CONTROL SYSTEMS BASED ON GENETIC ALGORITHM FOR INTELLIGENT ROBOTS", *Interdisciplinary Description of Complex Systems 12(3), 245-254, 2014.*
8. Arturo Y., et. Al., "PID-Controller Tuning Optimization with Genetic Algorithms in Servo Systems", International Journal of Advanced Robotic Systems, 2013, Vol. 10, 324:2013.
9. Arturo Y., et. Al., "PID-Controller Tuning Optimization with Genetic Algorithms in Servo Systems", International Journal of Advanced Robotic Systems, 2013, Vol. 10, 324:2013.
10. Kit-sang Tang, *Kim-fung Man, Zhi-feng Liu, and Sam Kwong,* "Minimal Fuzzy Memberships and Rules Using Hierarchical Genetic Algorithms", IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 45, NO. 1, FEBRUARY 1998.