

DESIGN OF FUZZY LOGIC CONTROLLERS FOR OPTIMAL PERFORMANCE*

Mohamed B. Trabia, Professor

Department of Mechanical Engineering

University of Nevada, Las Vegas

Las Vegas, NV 89154-4027

*Based on, Trabia, M. And W. McCarthy, "Design of Fuzzy Logic Controllers for Optimal Performance," Journal of Intelligent & Fuzzy Systems, Volume 6, Number 4, December 1998, pp. 459-470.

INTRODUCTION

- An expert in a particular system may be able to easily design a fuzzy logic controller for it since he is aware of the relations between its inputs and outputs.
- While fuzzy logic controllers are generally robust, the performance of a system whose behavior is not well understood or that has a large number of coupled inputs and outputs may be less than optimal.

Literature Survey

- Procyk and Mamdani (1977) developed a method for self-organized controllers. In their method, the rules are continually adjusted to improve the performance of the controller.
- Sugeno (1985) used structure and parameter identification process that is similar to that used in traditional control, to select the rules of the fuzzy controller based on experimental data.
- Langari and Berenji (1992) presented a method for self-organizing fuzzy control system that has a "critic" program. The "critic" evaluated the performance of the system and a "performance optimizer" program that rewards or penalizes the rules according to the "critic" output.
- Berenji (1992a and 1992b) suggested using neural networks to train fuzzy logic controllers.
- Jang (1993) proposed a method for tuning fuzzy inference system using the backpropagation neural-network algorithm based on collection of input-output data.
- Ishibuchi et al. (1993) presented a learning method of neural networks that uses the numerical data and the fuzzy inference rules. The network adjusted these rules automatically.

- Lee et al. (1996) tuned the parameters for the shape of antecedent linguistic terms and the relative degree of importance of these terms using neural networks. The fuzzy model they used consisted of two neural networks: the first evaluated antecedent matching degrees while the other reflected the matching degrees to consequence and defuzzified them.
- Isaka and Sebald (1992) used simulated annealing to produce a set of optimal membership functions.
- Atahalye et al. (1993) presented a method for optimizing a fuzzy system by adjusting the membership functions.
- Ramaswamy et al. (1993) used a method to automate the tuning of a fuzzy logic controller using a simplified Kalman filter approach.
- Manorajan et al. (1995) presented a method for systematic approach for determining the grades of membership of fuzzy sets using the simplex optimization method.
- Perneel et al. (1995) compared genetic algorithms to neural networks in the optimization of fuzzy expert systems. They concluded that genetic algorithms are superior when it is difficult to provide an efficient database for training the neural network. Most of these researchers concentrated on tuning the membership sets or the fuzzy inference rules separately.

Objectives:

In this paper, nonlinear programming techniques are used to improve the performance of a fuzzy logic controller. The algorithm modifies:

1. the fuzzy inference rules,
2. the membership sets of both controller and sensor outputs.

PERFORMANCE INDEX

- The performance of a fuzzy logic controller is generally dependent on the choice of the membership functions and the fuzzy inference rules.
- Intuition or experience can be used to determine the inference rules and the membership sets.
- This process becomes increasingly difficult if the system is nonlinear or when the number of controller outputs and sensor outputs are large.
- While even a poorly formulated fuzzy logic controller is usually robust, its performance may be less than optimal. Therefore, a method to optimize the performance of a fuzzy logic controller is needed.
- A controller may have many, or may be an infinite number of, "acceptable" designs. The optimization process starts by defining a performance index that measures the controller's performance over time such as,

$$PI = \int_0^T (\bar{y}_d - \bar{y})^T \cdot (\bar{y}_d - \bar{y}) dt$$

where, y_d is the desired output of the system. The above equation is a possible representation of the objective function.

- It is proposed to minimize the performance index by modifying:
 - i. The fuzzy inference rules. These rules are represented numerically so the nonlinear programming techniques can be applied to them.
 - ii. The range and the shapes of membership functions of the fuzzy linguistic terms of the controller and sensors outputs.
- The algorithm does not reduce number of inference rules. However, it has the ability to reduce the range of a membership set to a near-zero width. This reduction practically eliminates a fuzzy linguistic term from the inference rules.

Numerical Representation of the Fuzzy Inference Rules

- The fuzzy inference rules for a system with n controller outputs and m sensor outputs may be represented using n hyper-surfaces.
- Each hyper-surface is $m+1$ dimensional. To quantitatively describe various fuzzy linguistic terms, appropriate numerical values are assigned to each membership set such as those in Table I values to describe their relative weights.

Table I Possible Numerical Values for Fuzzy Linguistic Terms

Set	NB	NM	NS	Z	PS	PM	PB
Numerical Value	-3	-2	-1	0	1	2	3

- These numbers may be used to identify the fuzzy inference rules. Therefore, the fuzzy inference rules can be written as,

$$I_i(j, k, \dots, m) = n$$

where, I_i is the fuzzy term describing the i th controller output.

j, k, \dots, m are the fuzzy terms describing the sensor outputs 1 through m .

Example:

$$I_1(-2,0) = 1$$

- The preceding equation shows that the first controller output belongs to the *Positive Small* set when the output of the first sensor is of the *Negative Big* set and the output of the second sensor is of the *Zero* set.
- These variables are subject to the following constraint,

$$S_{I_j \min} \leq I_j \leq S_{I_j \max}$$
$$S_{O_k \min} \leq O_k \leq S_{O_k \max}$$

where:

- I_j and O_k are the j th controller output and the k th sensor output respectively.
- $S_{I_j \min}$ and $S_{I_j \max}$ are the minimum and maximum permissible numerical values for this variable.

Representation and Rules for Trapezoidal Fuzzy Membership

- A trapezoidal fuzzy membership set can be described using the variable,

$V_{i,j,k,l}$ where,

$i = y$ (sensor output) or u (controller output),

$j =$ output number,

$k =$ the fuzzy linguistic term numerical value and,

$l =$ vertex number (ranging from one to four).

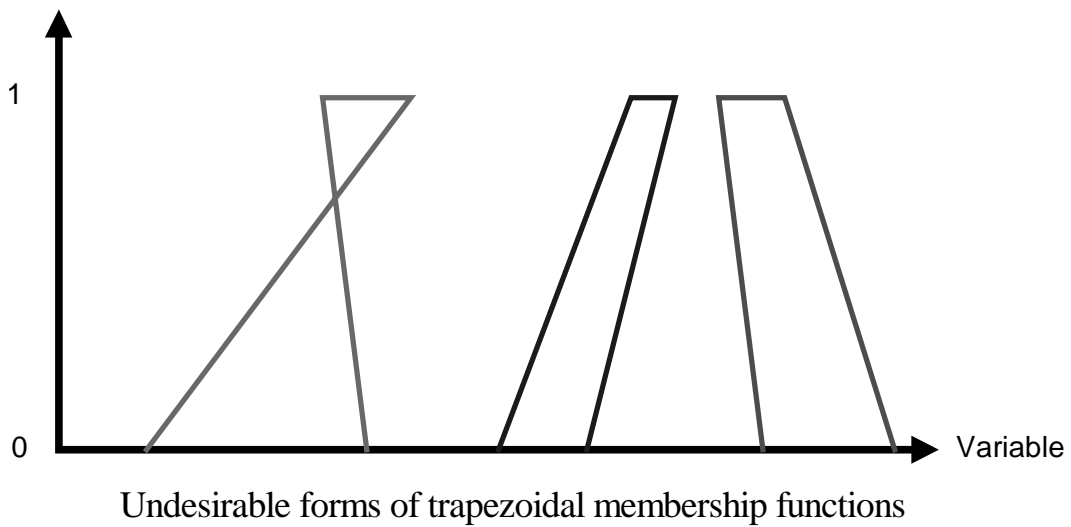


Trapezoidal Membership Function of a Fuzzy Linguistic Term

Constraints:

$$V_{i,j,k,l+1} \geq V_{i,j,k,l} \quad l = 1,2,3$$

- This constraint ensures that the truthfulness value of any variable value within a set is unique. Figure 3 shows cases when this constraint was not imposed. Additional constraints may be added to describe a particular system.



Optimization of the Fuzzy Logic Controller

- The objective of the optimization is to minimize the performance index for a fuzzy logic controller by modifying:
 1. the fuzzy inference rules and,
 2. the membership sets of both controller output and sensor outputs.

Example

- A fuzzy logic controller for a system with two sensors and one controller output is to be optimized.
- If each of the sensor outputs and the controller input is described using three fuzzy linguistic terms, the maximum number of fuzzy inference rules is nine.
- The nine membership sets are defined using thirty-six vertices.
- The total number of variables is forty-five.
- Nine of these variables, the fuzzy inference rules, are discrete.

First Stage of Tuning

- The optimization process starts by tuning the fuzzy inference rules while the membership sets are fixed. Tuning the membership sets follows. This decoupling is necessary to avoid sensitivity issues that may arise from having different scales for the rules and the membership sets vertices.
- The proposed algorithm uses a modified form of Hooke-Jeeves optimization method.

Hooke-Jeeves optimization Algorithm

1. Define:

Initial guess $x(0)$

Increment vector Δx_i ($i=1, 2, \dots, N$)

Step reduction factor $\alpha > 1$

Termination parameters $\epsilon_1, \epsilon_2 > 0$

Iteration counter $k = 0$

2. Test two points, along x_1 direction, at $\pm \Delta x_1$ with respect to $x(k)$. Move to the point with the better performance index.

3. Repeat the previous step sequentially for the other variables (2 through N) as long as no constraint is violated. If a constraint is violated, or if the program cannot find a better neighboring point while searching in the x_i direction, reduce the step Δx_i as follows,

$$\Delta x_i = \frac{\Delta x_i}{\alpha}$$

This process is repeated till a feasible point is obtained. The resulting point is labeled $x(k+1)$.

4. Move to a new point,

$$x(k+2) = x(k+1) + (x(k+1) - x(k))$$

If the new point is not feasible, reduce the distance to the new point by a ratio α successively till $x(k+2)$ becomes feasible.

5. Let, $k=k+2$

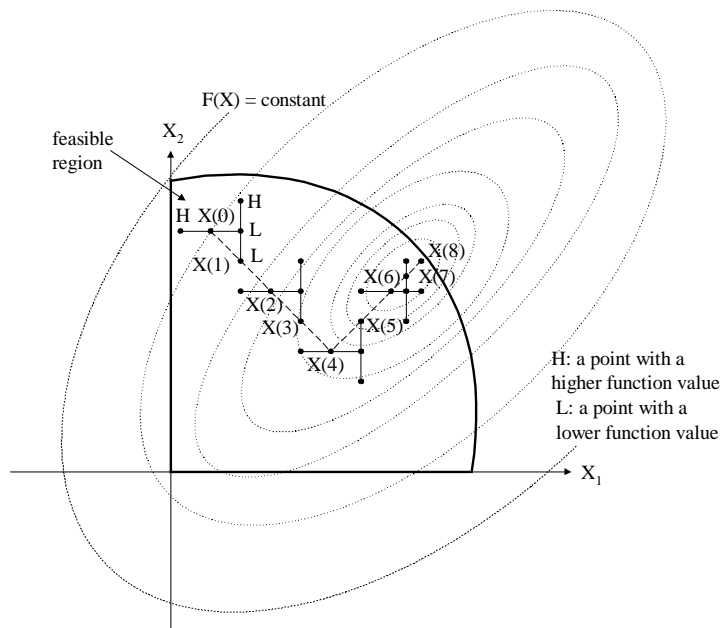
6. The search terminates when either of the following conditions is satisfied:

i. $\|\Delta x\| < \epsilon_1$

ii. the number of steps exceeds the maximum allowable number

iii. $|f(x(k)) - f(x(k-1))| < \epsilon_2$

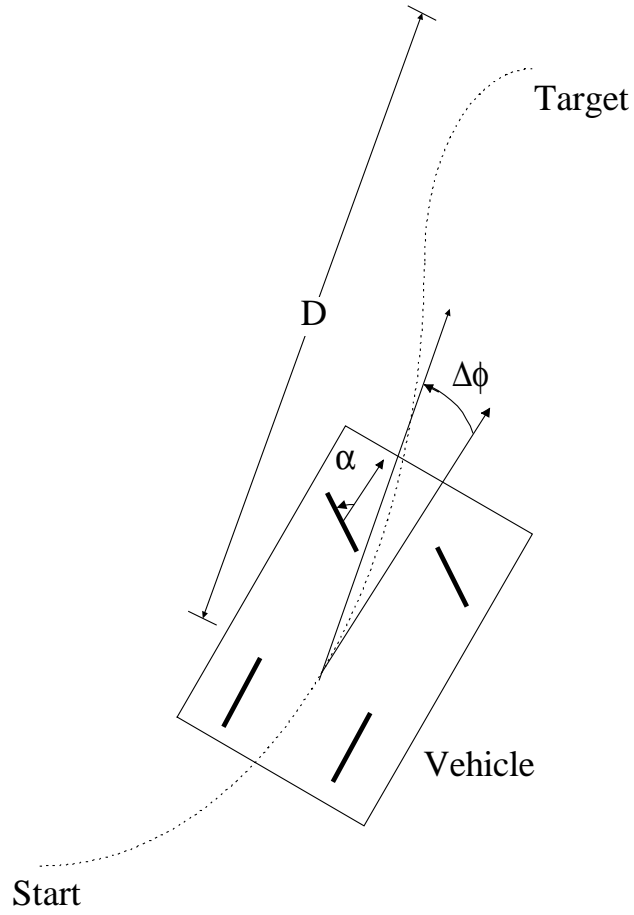
7. If no termination criterion is satisfied, go to step 2.



Graphical Representation of the Modified Hooke-Jeeves Nonlinear Programming Algorithm

EXAMPLE: OPTIMIZATION OF A FUZZY CONTROLLER FOR THE NAVIGATION OF AN AUTONOMOUS VEHICLE

The problem of guiding a car-like autonomous vehicle to a target is considered. If the vehicle is moving at a constant speed, it will have one controller output, which is the change in the steering angle, $\Delta\alpha$. The measured outputs are the steering angle α , which is measured with respect to the vehicle axis, the vehicle current orientation error with respect to the target $\Delta\phi$, and the distance to the target, D . These variables are shown in Figure 5.

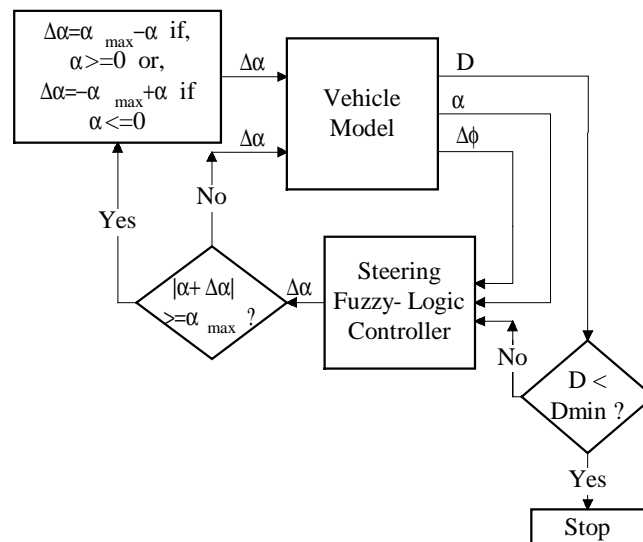


Steering of an Autonomous Vehicle Toward a Target

- The controller determines the change in the steering angle at every time step, δt , to guide the vehicle toward a target. The controller calculates the new steering angle, α , as follows,

$$\alpha = \alpha + \Delta\alpha$$

- $\Delta\alpha$ is constrained such as the steering angle, α , is within the mechanical limits of $\pm\alpha_{\max}$ degrees, which is the maximum steering angle to either direction of the vehicle.
- The distance to the target D is not expressed in terms of fuzzy linguistic terms since it is only used to stop the vehicle once it reaches the neighborhood of the target.



Block Diagram for the Navigation of an Autonomous Vehicle

Objective function:

$$PI = \sum_{t=0}^T \left((x(t) - x(t - \Delta t))^2 + (y(t) - y(t - \Delta t))^2 \right)$$

Controller Setup:

- The steering angle (α) and the vehicle current orientation error with respect to the target $\Delta\phi$ are labeled as sensors output number 1 and output number 2 respectively.
- Each of the three variables, α , $\Delta\phi$, and $\Delta\alpha$, is described using the following seven fuzzy linguistic terms:
- Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM), and Positive Big (PB).
- These terms are given numerical values starting by NB as -3 through PB as 3 as shown in Table I.
- Fuzzy logic controller has a maximum of forty-nine fuzzy inference rules.

- The fuzzy membership sets are subject to the following additional constraints:
 - i. To ensure that no preference to left or right steering occurs, all the membership sets are symmetrically placed around the zero value of each variable. This restriction results in the following constraint on the membership set of the Z term,

$$V_{i,j,0,1} = -V_{i,j,0,4} \qquad V_{i,j,0,2} = -V_{i,j,0,3}$$

- ii. Similarly, PS and NS, PM and NM, and PB and NB sets are symmetric.

This constraint can be described using these equations,

$$\begin{aligned} V_{i,j,-k,1} &= -V_{i,j,k,4} & V_{i,j,-k,2} &= -V_{i,j,k,3} \\ V_{i,j,-k,3} &= -V_{i,j,k,2} & V_{i,j,-k,4} &= -V_{i,j,k,1} \end{aligned}$$

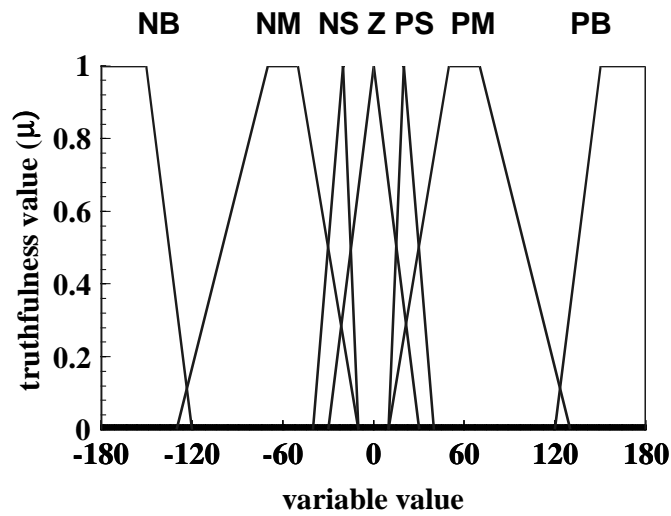
where, $k = 1..3$

- iii. Experience shows that a membership set can be used to describe the same fuzzy linguistic term for the three variables $\Delta\alpha$, α , and $\Delta\phi$ since all of them may be described using similar linguistic definitions.
- iv. To ensure meaningful results, the NB membership sets for α and $\Delta\phi$ variables always end at 180 degrees.
- v. Similarly, PB membership set for α and $\Delta\phi$ variables ends at 180 degrees as shown in the following equations,

$$\begin{aligned} V_{y,1,k,l} &= V_{y,2,k,l} = V_{u,1,k,l} & k &= -2..2 & l &= 1..4 \\ V_{y,1,3,3} &= V_{y,2,3,3} = V_{y,1,3,4} = V_{y,2,3,4} & & & & = 180^\circ \end{aligned}$$

- All these constraints result in limiting the number of vertices that will be variables in the optimization process to fourteen. These variables are,

$$\begin{aligned}
 &V_{u,1,-3,l} && l = 1..4 \\
 &V_{y,1,k,l} && k = -2,-1 \quad l = 1..4 \\
 &V_{y,1,0,1}, V_{y,1,0,2}
 \end{aligned}$$



Symmetrical Membership Sets

Initial Guesses:

- By interviewing drivers, inference rules and membership sets, Table II through Table IV, for navigational controller are proposed.
- The controller is trying to guide the vehicle from a starting point at (0,0) m to a target at (100,100) m. The simulation stops if the vehicle is at distance less than 1.5 m (D_{min}).
- The vehicle original orientation is at -135 degrees with respect to the horizontal.
- The original steering angle is zero degrees.
- Changing these conditions does not affect the results that are presented in the next section significantly.

Table II Initial Fuzzy Rules for the Steering Controller

$\Delta\phi \rightarrow$ $\alpha \downarrow$	NB	NM	NS	Z	PS	PM	PB
NB	Z	PS	PM	PB	PB	PB	PB
NM	NS	Z	PS	PM	PM	PM	PB
NS	NM	NS	Z	PS	PS	PM	PB
Z	NB	NM	NS	Z	PS	PM	PB
PS	NB	NM	NS	NS	Z	PS	PM
PM	NB	NM	NM	NM	NS	Z	PS
PB	NB	NB	NB	NB	NM	NS	Z

Table III Initial Guess for the Membership Sets of the Steering Angle and the Orientation Error Fuzzy Terms (α , $\Delta\phi$) (in degrees)

Vertices (j=1,2) \rightarrow Fuzzy Term (K) \downarrow	$V_{y,j,K,1}$	$V_{y,j,K,2}$	$V_{y,j,K,3}$	$V_{y,j,K,4}$
0 (Z)	-2.0	0.0	0.0	2.0
1 (PS)	1.5	2.5	10.0	12.0
2 (PM)	10.0	15.0	27.0	34.0
3 (PB)	30.0	45.0	180.0	180.0

Table IV Initial Guess for the Membership Sets of the Correction of Steering Angle Fuzzy Terms ($\Delta\alpha$) (in degrees)

Vertices → Fuzzy Term (K) ↓	$V_{u,1,K,1}$	$V_{u,1,K,2}$	$V_{u,1,K,3}$	$V_{u,1,K,4}$
0 (Z)	-2.0	0.0	0.0	2.0
1 (PS)	1.5	2.5	10.0	12.0
2 (PM)	10.0	15.0	27.0	34.0
3 (PB)	30.0	45.0	55.0	55.0

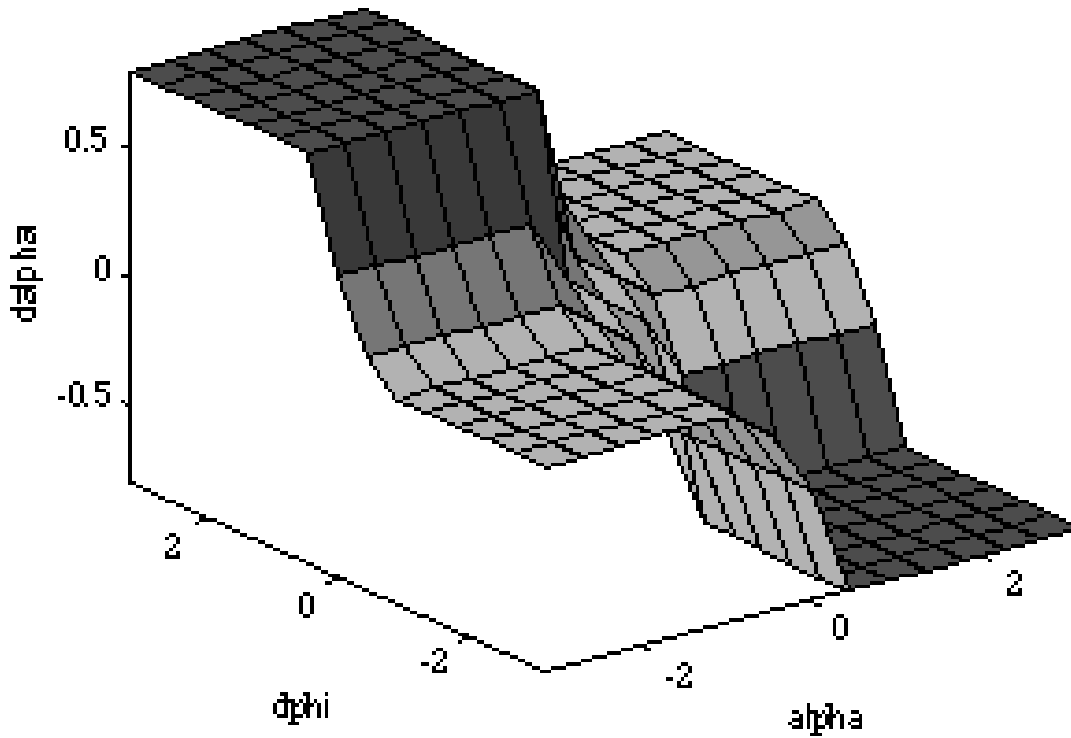


Figure 8 Initial Rule Surface

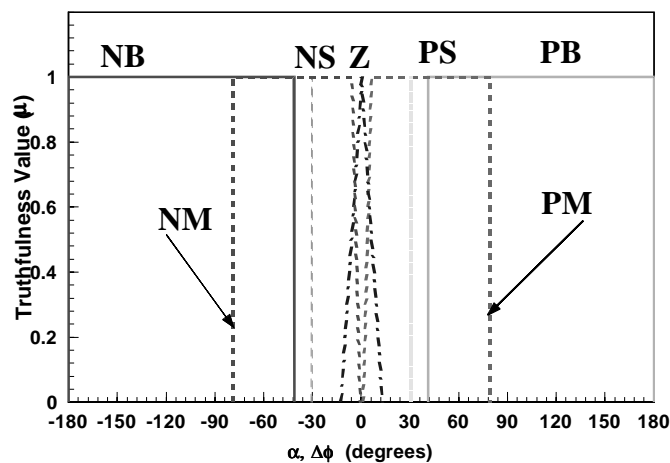
Results and Comparison

Table V Optimized Fuzzy Rules for the Steering Controller

$\Delta\phi \rightarrow$ $\alpha \downarrow$	NB	NM	NS	Z	PS	PM	PB
NB	Z	PS	PS	PM	PM	PB	PB
NM	NS	Z	PS	PS	PM	PM	PB
NS	NS	NS	Z	PS	PS	PM	PM
Z	NM	NS	NS	Z	PS	PS	PM
PS	NM	NM	NS	NS	Z	PS	PS
PM	NB	NM	NM	NS	NS	Z	PS
PB	NB	NB	NM	NM	NS	NS	Z

Table VI Optimized Membership Sets for the Steering Angle and the Orientation Error Fuzzy Terms (α , $\Delta\phi$) (in degrees)

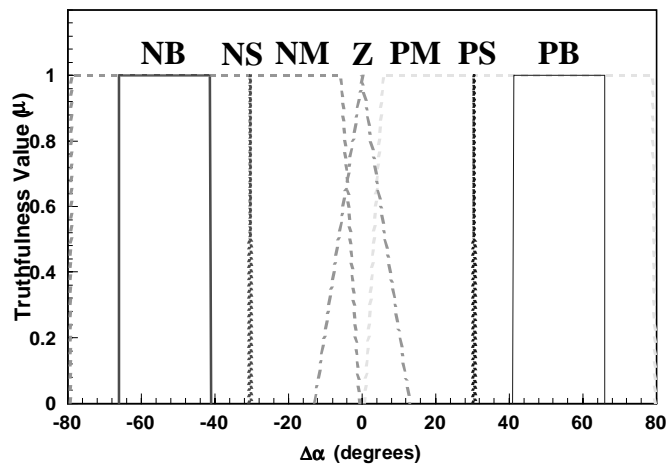
Vertices (j=1,2) \rightarrow Fuzzy Term (K) \downarrow	$V_{y,j,K,1}$	$V_{y,j,K,2}$	$V_{y,j,K,3}$	$V_{y,j,K,4}$
0 (Z)	-13.0	0	0	13.0
1 (PS)	29.95	30.23	30.52	30.81
2 (PM)	0.57	5.51	79.23	79.52
3 (PB)	41.0	41.3	180.0	180.0



Optimized Membership Sets for the Steering Angle and the Orientation Error Fuzzy Terms (α , $\Delta\phi$) (in degrees)

Table VII Optimized Membership Sets for the Correction of Steering Angle
Fuzzy Terms ($\Delta\alpha$) (in degrees)

Vertices \rightarrow Fuzzy Term (K) \downarrow	$V_{u,1,K,1}$	$V_{u,1,K,2}$	$V_{u,1,K,3}$	$V_{u,1,K,4}$
0 (Z)	-13.0	0	0	13.0
1 (PS)	29.95	30.23	30.52	30.81
2 (PM)	0.57	5.51	79.23	79.52
3 (PB)	41.0	41.3	66.0	66.0



Optimized Membership Sets for the Correction of Steering Angle Fuzzy Terms
($\Delta\alpha$) (in degrees)

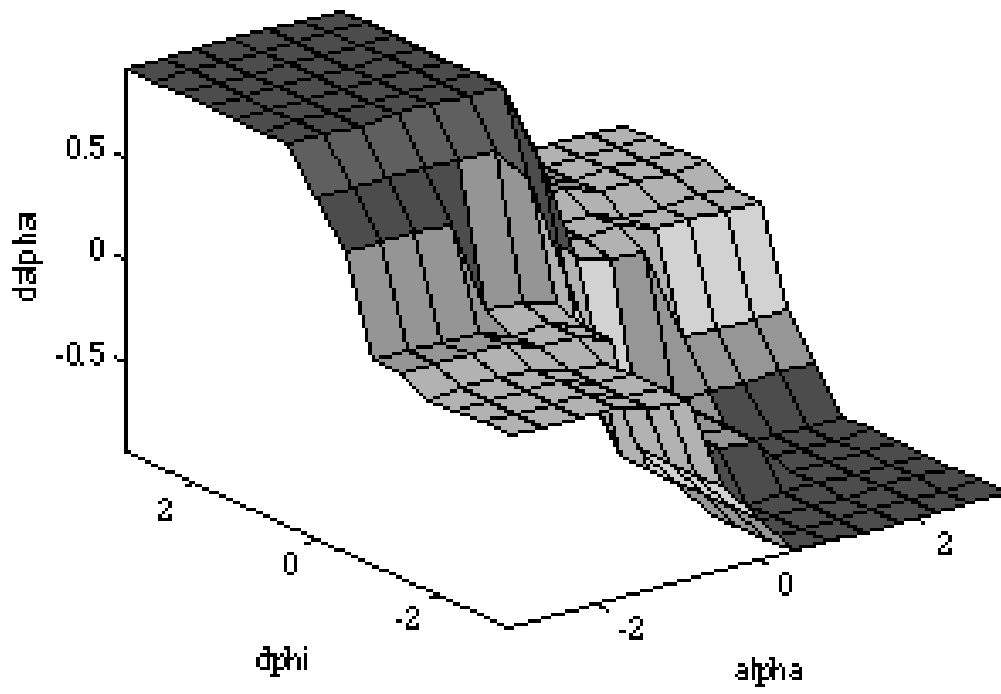


Figure 11 Optimized Rule Surface

- The optimization algorithm required a total of seventy-three iterations to reach the solution.
- Comparing rule surfaces shows that:
 - i. The optimized controller has a lesser area where the steering correction is zero.
 - ii. The optimized controller produces a more gradual response to changes in the two measured outputs.

- Results also show that the NS and PS sets are completely enclosed within the NM and PM sets respectively.
- Thus, the optimization process simplified the proposed fuzzy logic controller by practically eliminating the PS and NS sets from the terms describing α , $\Delta\phi$, and $\Delta\alpha$. The rules for the reduced controller are listed in Table VIII.
- The reduced controller produced the same response as the optimized controller.
- Similarly, the NB and PB sets of the correction of steering angle $\Delta\alpha$ are completely enclosed within the NM and PM sets respectively.
- The enclosure of the Big sets shows that, from a linguistic point of view, the optimization scheme exchanged the *medium* and *big* terms.
- While relabeling the sets after this redefinition may be desirable, it is not a necessary step.

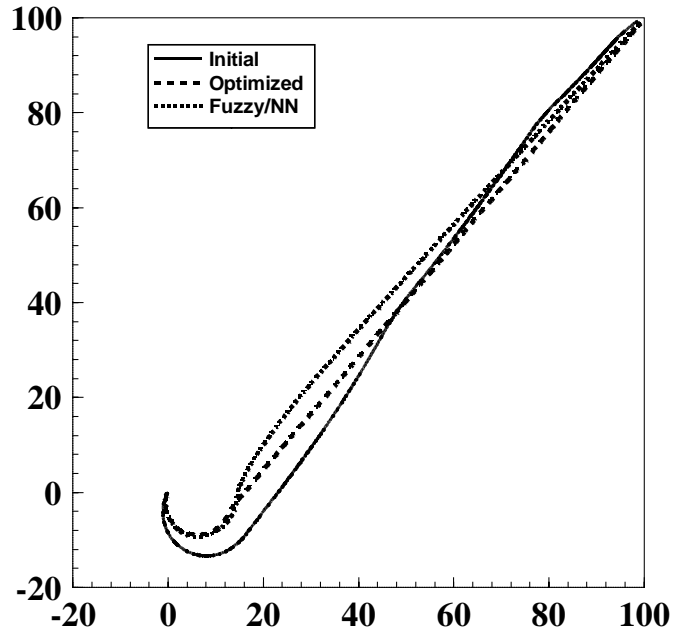
Table VIII Reduced Fuzzy Rules for the Steering Controller

$\Delta\phi \rightarrow$ $\alpha \downarrow$	NB	NM	Z	PM	PB
NB	Z	PS	PM	PB	PB
NM	NS	Z	PS	PM	PB
Z	NM	NS	Z	PS	PM
PM	NB	NM	NS	Z	PS
PB	NB	NB	NM	NS	Z

Comparison with ANFIS:

- The initial and the optimized controllers are compared with that produced by training a fuzzy logic controller using ANFIS, which is an adaptive neuro-fuzzy inference, Jang (1993).
- ANFIS uses Sugeno-style fuzzy inference, Sugeno (1985) with four first-order fuzzy rules.
- The controller is tuned using the backpropagation algorithm.
- The training data needed are obtained by running the initial controller. The controller is trained for eighty epochs.

Figure 10 shows the paths of the vehicle when using the initial, the optimized, and the ANFI fuzzy logic controllers respectively. The results, which are also listed in Table IX, show that the proposed optimization scheme produces satisfactory results.



Path of the Autonomous Vehicle Using the Initial, Optimized, and Fuzzy-Neural-Net Controller

Table IX Comparison of the Performance of the Three Fuzzy logic Controllers (Absolute minimum path length is equal to 155.004 m)

	Initial Controller	Optimized Controller	Fuzzy/ NN Controller
Path Length	178.2	155.8	157.2
Percentage Improvement in Path Traversal Time as Compared to that of the Initial Controller	0.0	5.6	4.7

CONCLUSION

- The design of fuzzy logic controllers requires knowledge of the appropriate fuzzy inference rules and fuzzy membership sets that describe the fuzzy linguistic terms.
- This process is usually done by questioning a skilled operator of the controlled.
- The performance of such controller may not be optimal especially if the controlled system is complex or if precise information about it is not readily available.
- A novel method for the design of a fuzzy logic controller for optimal performance is presented.
- The method starts by representing fuzzy inference rules numerically. Constraints for ensuring meaningful shapes of the membership sets are also imposed.
- The optimization algorithm adjusts fuzzy inference rules and shapes of membership sets of the fuzzy linguistic terms using Hooke-Jeeves nonlinear programming method.
- The objective of the optimization is to minimize a controller's performance index.

- The proposed method can simplify the controller by suggesting a reduction of the number of the fuzzy inference rules.
- The algorithm is successfully applied to the problem of designing a fuzzy logic controller for navigating an autonomous vehicle. The performance of the optimized controller is compared to:
 - i. A controller whose rules and sets are based on interviewing drivers.
 - ii. A fuzzy controller that is trained using neural-networks.
- The results show a performance improvement of the optimized controller over both controllers.