GENETIC ALGORITHM TUNED FUZZY LOGIC FOR GLIDING RETURN TRAJECTORIES*

Bradley T. Burchett†
Rose-Hulman Institute of Technology

prepared by
Bradley Burchett
Department of Mechanical Engineering
Rose-Hulman Institute of Technology
Terre Haute, IN 47803

under NASA grant number NAG8-1915.

---

* This work was supported by NASA Contract number NAG8-1915.

† Assistant Professor, Member AIAA

ABSTRACT

The problem of designing and flying a trajectory for successful recovery of a reusable launch vehicle is tackled using fuzzy logic control with genetic algorithm optimization. The plant is approximated by a simplified three degree of freedom non-linear model. A baseline trajectory design and guidance algorithm consisting of several Mamdani type fuzzy controllers is tuned using a simple genetic algorithm. Preliminary results show that the performance of the overall system is shown to improve with genetic algorithm tuning.

Contents

List of Figures:

1. NOMENCLATURE

ψ	heading angle with respect to runway (deg)
ϕ	bank angle (deg)
γ	flight path angle (deg)
g	gravity constant = 32.174 ft/s^2
V	forward velocity (ft/s)
x	vehicle cg distance from runway threshold (ft)
y	vehicle cg distance from runway centerline (ft)
h	vehicle cg height above runway (ft)
N_Z	vehicle z axis normal acceleration (g)
D	drag force (lb$_f$)
C_D	drag coefficient
ρ	atmospheric density (slug/ft^3)
S	vehicle frontal area (ft^2)
m	vehicle mass (slugs)
E_W	vehicle energy divided by weight (ft)
X_{MAX}	X_{HAC} value for maximum energy approach
X_{FTC}	Nominal value for X_{HAC}
X_{ALI}	X runway coordinate of baseline Auto Landing Interface (ALI) point
X_{MEP}	X runway coordinate for Minimum Entry point approach
X_{HAC}	X runway coordinate of heading alignment cone center
r_{HAC}	radius of heading alignment turn (ft)
HAC	heading alignment cone
E_{MEP}	Energy over weight for minimum entry point approach
E_N	Energy over weight for nominal approach

SS	Energy over weight for small s-turn approach
SM	Energy over weight for medium s-turn approach
SL	Energy over weight for large s-turn approach
V_{HAC}	expected velocity at HAC initiation (ft/s)
ϕ_{AVG}	average bank angle during HAC turn (deg)

The following acronyms are used repeatedly in describing the Fuzzy Logic input and output membership functions:

LN	Large Negative
MN	Medium Negative
SN	Small Negative
Z	Zero
SP	Small Positive
MP	Medium Positive
LP	Large Positive

2. INTRODUCTION

NASA is currently studying the application of advanced and intelligent control methodologies to the successful recovery of Reusable Launch Vehicles (RLVs). In recent years several authors have contributed to this work. Hanson [7,8] provides a fairly comprehensive overview of the work currently funded through NASA Marshall.

The return to earth consists of three phases--Entry, Terminal Area Energy Management (TAEM), and Approach and Landing. Entry is defined as taking the spacecraft from 190,000 ft to 90,000 ft above mean sea level. TAEM takes the spacecraft from 90,000 ft above mean sea level and Mach 3 to 10,000 ft and Mach 0.5 and aligns the craft with extended runway centerline. Approach and Landing takes the vehicle from 10,000 ft to wheel stop on the runway. The Space Shuttle is programmed to fly all three phases of flight automatically, and under normal circumstances the astronaut-pilot takes manual control only during the Approach and Landing phase. The automatic control algorithms used in the Shuttle for TAEM and Approach and Landing have been developed over the past 30 years. They are computationally efficient, and based on careful study of the spacecraft's flight dynamics, and heuristic reasoning. The gliding return trajectory is planned prior to the mission, and only minor adjustments are made during flight for perturbations in the vehicle energy state.

In order to provide more flexibility during recovery, especially in the case of off nominal energy conditions at Entry / TAEM interface or control surface failures, several authors have been investigating the application of advanced control technologies to autonomously design the trajectory in real time.

The bulk of work published to date deals primarily with the approach and landing phase of flight. Ref [1] focuses on the auto landing trajectory where changes in heading angle are small and the distance to runway threshold is monotonically decreasing. Ref [2] proposes new methods to assess the robustness of auto landing trajectories. Ref [4] shows results for the subsonic portion of TAEM. Ref [6] uses an adaptive-critic neural network approach to optimize trajectory design for the approach and landing phase of flight. Once again, the mathematics used are only applicable when the heading changes are small and the distance to the runway is monotonically decreasing.

The goal of this work is to take a set of baseline fuzzy inference systems designed for autonomous trajectory design and guidance [3], and tune the internal parameters using a simple genetic algorithm. Results are shown for the supersonic and subsonic portions of TAEM. Trajectory design and guidance during TAEM has been challenging for most approaches because of the large changes in heading angle that are allowed during this phase, and the large changes in aerodynamic drag during transonic flight.

3. SIMPLIFIED PLANT MODEL

An overly simplified plant model was used for all tuning simulations in this work. The details from our previous work [3] are repeated here for completeness. The simplified model of a gliding aircraft can be found by treating the vehicle as a point mass, and applying Newton's second law in the aircraft *y-z* and *x-z* planes separately. Considering the *y-z* plane first, and assuming no acceleration in the *z* direction, Newton's second law renders

$$L\cos\phi - mg = 0 \tag{1}$$

Which can be solved for lift L yielding

$$L = \frac{mg}{\cos\phi} \tag{2}$$

In the y direction we find

$$L\sin\phi = m\frac{V^2}{r} \tag{3}$$

Eq. 2 is then substituted for lift in Eq. 3, resulting in an expression for turn radius that is independent of aircraft size or type.

$$r = \frac{V^2}{g}\frac{1}{\tan\phi} \tag{4}$$

Now substituting $\dot{\psi}^2 r = \frac{V^2}{r}$ in Eq 4 and solving for turn rate $\dot{\psi}$, results in the following:

$$\dot{\psi} = \frac{g}{V}\tan\phi \tag{5}$$

In the x-z plane first show the force balance along the aircraft z axis

$$L - mg\cos\gamma = m\frac{V^2}{r} = N_Z m \tag{6}$$

Solving for turn radius in the vertical plane yields

$$r = \frac{mV^2}{L - mg\cos\gamma} \tag{7}$$

Solving Eq 6 for normal acceleration N_Z renders

$$N_Z = \frac{L}{m} - g\cos\gamma \tag{8}$$

Now substitute Eq 7 and $\dot{\gamma}^2 r = \frac{V^2}{r}$ into Eq 6 and solve for $\dot{\gamma}$

$$\dot{\gamma} = \frac{1}{V}\left(\frac{L}{m} - g\cos\gamma\right) \tag{9}$$

Finally, substituting Eq 8 into Eq 9 will eliminate lift and thus make the model independent of specific aircraft aerodynamic properties.

$$\dot{\gamma} = \frac{N_Z}{V} \tag{10}$$

The force-acceleration balance along the aircraft x axis results in an equation for forward velocity.

$$\dot{V} = -\frac{D}{m} - g\sin\gamma \tag{11}$$

Where drag D is given by

$$D = \frac{1}{2}\rho V^2 S C_D \tag{12}$$

By definition of heading angle and flight path angle, the aircraft position in x-y-h space is governed by Eqs. (13-15).

$$\dot{x} = V\cos\psi \tag{13}$$

$$\dot{y} = V\sin\psi \tag{14}$$

$$\dot{h} = V \sin\gamma \quad (15)$$

Atmospheric density is determined from the exponential model:

$$\rho = \begin{cases} 0.0023784722\left(1 - 6.8789 \times 10^{-6} h\right)^{4.258} & h < 35332 \\ 0.00072674385 e^{-4.78 \times 10^{-5}(h - 35332)} & h \geq 35332 \end{cases} \quad (16)$$

Eqs. (5), (10), (11), and (13-16) provide a generic aerospace vehicle model where only the drag coefficient is aircraft specific. The coefficient of drag was taken to be a function of Mach number only and was approximated by the zero angle of attack portion of the drag table directly from a high-fidelity non-linear simulation of the X-33 Venturestar. These drag coefficient data are shown in Fig. 1.

Induced drag was ignored which is probably the biggest weakness in the simplified model. The resulting model was programmed in Matlab / Simulink to serve as a test bed for rapid prototyping, tuning and testing of fuzzy inference systems for trajectory design and guidance.

4. TRAJECTORY DESIGN AND GUIDANCE SYSTEM

The basic connections between plant, trajectory designer, and guidance are shown in Fig. 2.

During the low fidelity simulations used to tune the trajectory designer and guidance systems in this work, the trajectory designer provides a desired ground track and vertical path, and the guidance system provides the appropriate bank angle and negative *z* axis acceleration to intercept and maintain the desired path. Since the model is simplified to three degrees of freedom, the vehicle is assumed to follow the guidance commands immediately and perfectly, that is, there are no dynamics causing actual vehicle attitude to differ from commanded attitude.

The baseline trajectory design and guidance fuzzy inference systems were developed based on existing Shuttle TAEM guidance, and instrument approach procedure techniques used by military pilots[3]. The ground path in this work is essentially the same as current Shuttle guidance when using a straight-in approach. That is, at Entry / TAEM interface, the vehicle will turn in the shortest direction to a heading that will intercept the Heading Alignment Cone (HAC) in a tangent fashion. The craft then flies in a straight line until intercepting the HAC. HAC interception should occur as the craft reaches subsonic speeds. The HAC is so named, because an aircraft flying a constant bank descending turn with decreasing airspeed will actually describe a decreasing radius helix, hence the surface of a cone. In this work, the HAC is a constant radius turn and could be aptly renamed 'Heading Alignment Cylinder'. Once the spacecraft is within 45 degrees of runway heading and the distance to extended runway centerline is decreasing, the bank guidance system switches to a mode that will intercept extended runway centerline. The basic ground track in shown in Fig. 3.

4.1 Trajectory Designer

The trajectory designer is a fuzzy inference system with two inputs and three outputs. The inputs are the quotient of energy over predicted downrange distance to ALI E_W/R_{ALI}, and an integer denoting the degree of control surface health. Energy is computed as the sum of kinetic and potential energy divided by weight, and has dimensions of length.

$$E_W = h + \frac{V^2}{2g} \tag{17}$$

Distance to ALI is computed using the expected ground track from the previous design iteration. The calculations are identical to those used in Shuttle TAEM guidance. The membership functions for E_W/R_{ALI} are tunable. The trajectory designer outputs are described next.

Two parameters that determine the ground track are the heading alignment turn radius (Y_{HAC}), and position from the runway threshold (X_{HAC}). The output membership functions for these parameters will be tunable.

The vertical trajectory is constrained by initial and final conditions. The spacecraft must reach the Auto-Land Interface (ALI) at approximately 10,000 ft above the runway, 20,000 ft from the runway threshold at a flight path angle depressed 30 deg from horizontal. To maintain continuity throughout the trajectory, the vertical path is defined as a cubic polynomial which intersects the ALI at the appropriate altitude and slope.

$$h_{ref} = c_4 R_{ALI}^3 + c_3 R_{ALI}^2 - c_2 R_{ALI} + h_{ALI}$$

Where R_{ALI} is the predicted ground track distance to ALI and h_{ALI} is the desired height at ALI and c_3 is an adjustable parameter allowing real-time updates of the reference vertical path to match off-nominal energy conditions. The trajectory designer will also determine c_3 through a fuzzy decision with tunable membership functions.

4.2 Bank Guidance Commands

The bank guidance commands are generated by a fuzzy inference system with seven inputs 112 rules, and a single output. The bank output command is defined as a proportion of available bank. Available bank is limited by the degree of control surface health. The conventional sub-phases of TAEM, which are acquisition, HAC turn, and pre-final [11], are used to limit the number of inputs which must be considered during a single iteration. During the acquisition turn, the only input considered is turn angle to the HAC center, $\Delta\Psi_{aq}$ which is partitioned into five membership functions. In this phase, the fuzzy inference system acts as a proportional controller with five rules connected to the five output membership functions. The acquisition turn ends when the spacecraft is pointed within one-half degree of the heading that will take it tangent to the HAC, or when it reaches a range of 1.85 HAC radii from the HAC center, to facilitate intercepting the HAC arc.

During the HAC turn, the inputs which are considered are turn radius error and bearing to the HAC center. Bearing to the station is divided into thirteen membership functions. Turn radius error is defined as the quotient of actual turn radius over desired turn radius, and is divided into seven membership functions. Eighty-nine rules are used in this phase. The rules are based on the techniques for intercepting and flying a TACAN arc contained in Air Force Manual 11-217 volume 1.[13] The bearing, turn radius error and bank angle membership functions were tunable, and were tuned using a simulation of the HAC interception and maintenance phase only.

For transition to final approach, the distance to extended runway centerline, and rate of change of this distance are taken as tunable inputs. Proportion of available bank angle is the output, but is not tuned during this phase of flight.

4.3 Negative Z-Acceleration Guidance

The negative z-axis acceleration guidance has four inputs. Dynamic pressure, and Mach number are not tunable, since they represent the stall and structural limits of the spacecraft. Altitude error and vertical velocity error are considered tunable. The output, which is the guidance portion of negative z acceleration and ranges from -0.6*g* to 0.6*g* is considered tunable, however, as discussed in the sequel, the membership functions are constrained to this predetermined limit.

5. GENETIC ALGORITHM TUNING OF MAMDANI TYPE FUZZY CONTROLLERS

Genetic algorithms have been used to tune fuzzy logic controllers of the Mamdani type for several years. Previous merging of genetic algorithms and fuzzy logic has provided well-tuned low-order controllers for satellite docking control and chemical engineering [9] examples. Other methods of adaptive fuzzy control use the backpropagation algorithm, or tuning based on gradient information, and require that the controller use differentiable membership functions (MFs).[12] GAs provide a viable method of adapting fuzzy logic controllers for optimum performance even when the membership functions of the controller are not differentiable (i.e. triangles or trapezoids).

Prior to genetic algorithm tuning, a specific fuzzy logic architecture is chosen. That is, a controller with at least marginally adequate performance should be known, and coded as a baseline. The human control designer will thus know an appropriate universe of discourse for input and output variables, and a possible partitioning of the input and output spaces, including number, and shape of membership functions.

In this work, adjustable parameters of the controller are coded as unsigned binary integers. Trapezoidal membership functions are reduced to three parameters; mean, support and spread. These parameter definitions are illustrated in Fig. 4.

Triangular membership functions require only the mean and spread to be coded. This convention was used successfully by Linkens and Nyongesa.[10] This particular coding scheme is chosen so that, when the genetic algorithm produces randomly selected mutations, the shape and monotonicity of the membership functions will be preserved. Thus, the inference engine will give valid control outputs regardless of what numbers the GA picks for a particular membership function. The mean of each membership function is coded as a proportion of the expected universe of discourse. That is:

$$b = \left\lfloor \frac{P - P_{\min}}{P_{\max} - P_{\min}} \left(2^m - 1\right) \right\rfloor \tag{18}$$

Where $P_{\max}$ and $P_{\min}$ define the limits of the relevant universe of discourse. P is the mean as a real number, and b is the mean coded as an integer between 0 and $2^m - 1$. Spread and support are, by definition, positive quantities, thus, Eq. 18 is used to code these with $P_{\min}=0$. Each integer b is then coded as a binary string using a standard decimal to binary function.

In order to initialize the search, a controller based on human knowledge was used as the baseline [3]. A population size of 30 was chosen, in order to keep the computation time required for propagation of a single generation within reason. This population size is supported by the findings of Linkens and Nyongesa, where population sizes of 20 to 40 were used on bit strings of length 2080. [10]

5.1 Coding of Bank guidance Parameters

The total number of membership functions for bank guidance commands was seven for the turn radius error input, thirteen for the bearing to station input, and seven for the bank command output. Of these, turn radius error has three trapezoidal, and four triangular MFs, bearing has seven trapezoidal and six triangular, and bank command has MFs that are all triangular. Each membership function parameter (mean, support, and spread) was coded using 8 bits. Counting three adjustable parameters for trapezoidal MFs, and two for triangular ones, we have 64 total parameters, and 512 total bits. The combined chromosome for a candidate controller is produced by concatenating the bit strings for individual parameters.

The initial population was filled by taking random mutations of the baseline controller. Each new string had 52 alleles or approximately 10% altered from the baseline. Note these alterations are completely random and no insight to the effect on the fuzzy inference system parameters is required for successful adaptation.

5.2 Measure of Performance and Fitness Levels

The current example is one of tracking control. that is, the objective of the fuzzy inference system is to produce actuator commands that will cause the system to track a prescribed trajectory. Thus, a quadratic cost function based on tracking error is used as the measure of controller performance.

$$J = \sum_i \left(r_{ref_i} - r_{act_i} \right)^2$$

In order to measure performance, each candidate bit string is decoded into the corresponding fuzzy logic controller. This is done by stripping eight bits at a time from the chromosome, and associating those eight bits with the corresponding controller parameters. The eight bit binary string is then decoded to an integer value *b*. The corresponding real number parameter *P* is then found by inverting Eq. 18. For spread and support parameters, $P_{\min}$ in Eq. 18 is replaced with zero.

Each candidate controller is then used for a simulation of part of the gliding trajectory of the RLV. Control commands are updated every 0.5s, and are held, during a contiuous simulation of the system dynamics. The tracking error is also computed every 0.5s. The simulation is run until the vehicle reaches a goal, or a time limit is reached. In general, this time limit is set to twice the time required by the baseline controller to reach the goal.

Fig. 5 illustrates tracking error performance for the bank command guidance of the X-33 simulator.

Since GAs seek to maximize the 'fitness level', each candidate controller was assigned a fitness level equal to the difference between the maximum cost candidate controller cost divided by the maximum cost. That is

$$f_i = \left(\max\left(J_{pop}\right) - J_i\right) / \max\left(J_{pop}\right)$$

Typical J values for the baseline controller were $O(10^{11})$. This scaling gave values of f such that $0 \le f_i \le 1 \,\forall\, i$.

5.3 Implementation of the Simple Genetic Algorithm

Once fitness levels are established the next generation of chromosomes is produced by the simple genetic algorithm with the following specific characteristics.

1. The mating pool is selected using a biased roulette wheel.

2. Parent strings are paired randomly

3. Probability of crossover is 1, and crossover sites are chosen randomly. In particular, this means that the GA operates on the entire chromosomes with no regard to where bit substrings for individual parameters begin and end.

4. Proportion of mutation was set at 0.34%. That is, mutation is performed on 52 randomly chosen of the total 15360 alleles of each new generation. This is the most significant departure from the simple GA as presented by Goldberg [5]. Goldberg uses a mutation probability which is then applied to each individual allele, to determine which ones are altered during the mutation step.

Since membership function means are allowed to be moved anywhere in the universe of discourse, the rule base may be effectively changed by this type of adaptation. That is, although the linguistic names associated with various membership functions will not change, the crisp input / output values associated with the linguistic names are allowed to change. Thus, for instance, the membership function for 'Large Positive', may actually end up to the left of the membership function for 'Zero', thus resulting in an effective rule change.

This is illustrated in Figs. 6-11.

6. RESULTS

The Bank and Negative Z Axis acceleration fuzzy inference systems were tuned using the procedure described above. The bank guidance commands are essentially split into two separate fuzzy inference systems, one for the HAC turn, and one for transition to final approach. Fig. 5 shows the performance improvement of the HAC turn phase due to GA tuning. After tuning the input and output membership functions for the HAC turn phase, the GA was used to tune the input membership functions of the transition to final approach fuzzy system. In this case, the inputs are distance from extended runway centerline, and runway y coordinate component of horizontal velocity. Figure 12 below shows the improvement in transition to final aproach performance.

The bank guidance commands were tuned for a right HAC turn and right turn to final only. Thus, after tuning, the input and output membership functions which were not exercised during the tuning were set symmetric to the ones which were exercised. That is, for example, the LN, MN, ans SN membership functions shown 'scrunched' together to the left of Figure 11, were reset to reflect the values of LP, MP, and SP, only negative.

Fig. 13 shows the improvement in tracking the planned vertical trajectory as a result of GA tuning of the Negative Z Axis acceleration fuzzy guidance.

Once the guidance commands had been tuned, the trajectory design fuzzy inference system was tuned using the simple GA. In this case, in order to design for the widest possible energy envelope, the simulation was run for four different initial conditions for each candidate fuzzy system. The four initial conditions are shown in Table 1.

The cost function for each simulation was taken as the sum of squared error in four dimensions from the desired end state. The desired end state is shown in Table 2.

The fourth dimension is difference in final heading and runway heading in degrees. The errors in the three Cartesian dimensions are computed in feet/100 before squaring so that error in final heading angle has approximately the same amount of influence in the final cost. Thus, the cost for a single simulation is

$$J = \sqrt{(x_{err}/100)^2 + (y_{err}/100)^2 + (h_{err}/100)^2 + \psi_{err}^2}$$

Since four simulations are computed to determine the performance of each candidate trajectory designer the cost is taken as the sum of costs for the four simulations. Table 3a shows a comparison of the cost for the baseline controller and how the cost is minimized during tuning with the GA using a sum of cost criterion. Note that the genetic algorithm seeks to minimize the sum of the four costs shown, therefore, the cost for initial condition three is allowed to increase as long as the total cost decreases. Also, cost for initial condition four is diminished but still rather large after 45 generations of adaptation. Figure 14 shows the corresponding ground track taken by the vehicle when using the trajectory designer from this adaptation. It is obvious that for initial conditions three and four, the vehicle is not in a good position to interecept final approach. In order to provide for success on all four initial conditions, we also ran the adaptation with cost defined as the max cost over the four trials. Table 3b shows how the max cost evolves over 42 generations. Note that the adaptation seeks only to reduce the max cost, so after 24 generations, all of the numbers are rather large, and the trajectory designer does not reach the desired end state for any of the initial conditions, thus, we prefer the sum cost criterion.

7. CONCLUSIONS

In this work, a simple genetic algorithm has been employed to tune the parameters of several fuzzy inference systems used in trajectory design and guidance of a reusable launch vehicle. The Terminal Area Energy Management portion of flight is considered. Such tuning has provided a way to optimize the design of fuzzy logic controllers for this application. Although the GA was only allowed to adjust the membership functions, the association of crisp input / output values with linguistic names used in the inference engine could change, resulting in an effective rule change. After tuning, the spacecraft is flown to a prescribed end point with a greater degree of precision for widely varied initial conditions. The result is an optimized trajectory design and guidance algorithm which has been demonstrated to control a simplified model of the plant from Entry / TAEM interfact to auto-land interface with a great degree of success.

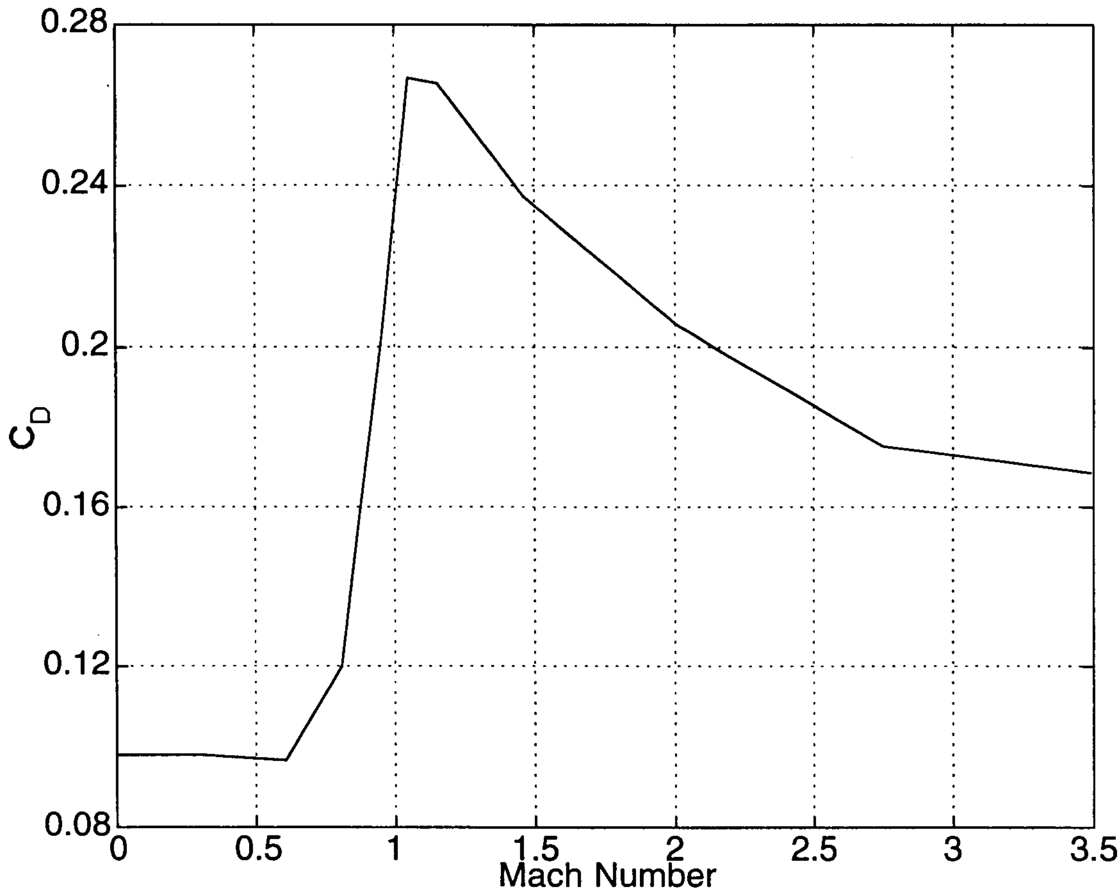

Figure 1: Drag Coefficient as a function of Mach Number

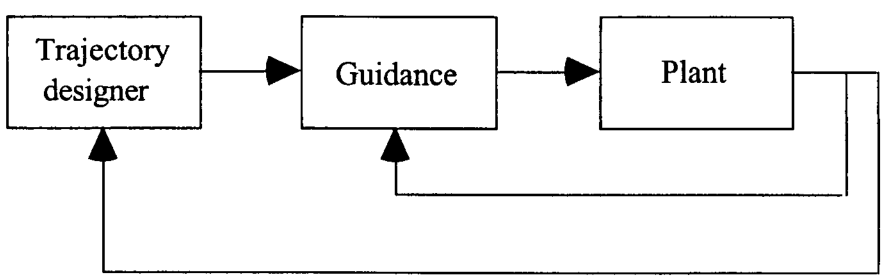

Figure 2: Trajectory Design and Guidance Connections

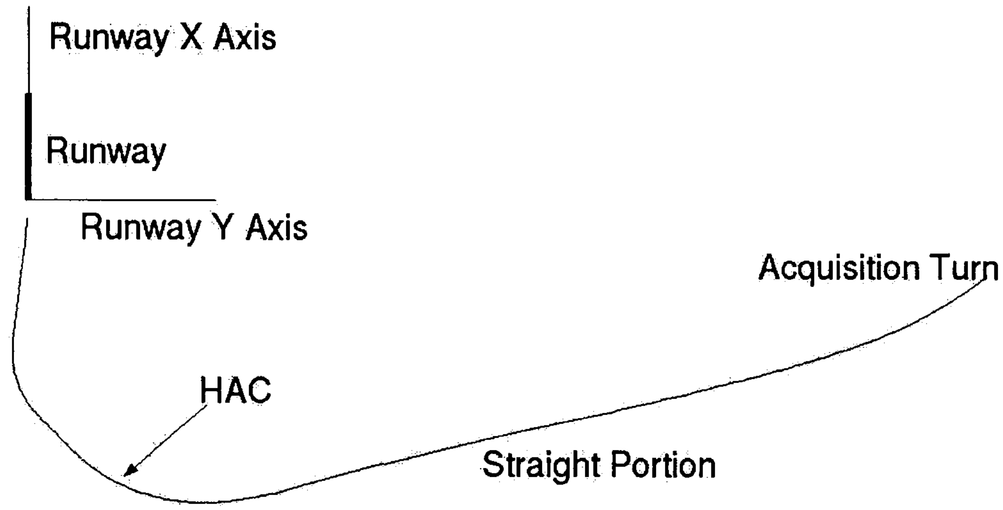

Figure 3: Basic Ground Track Shape

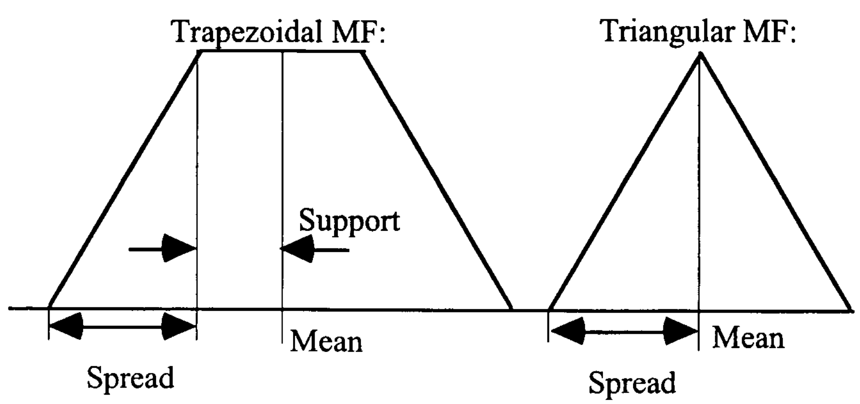

Figure 4: Illustration of Membership Function Coding

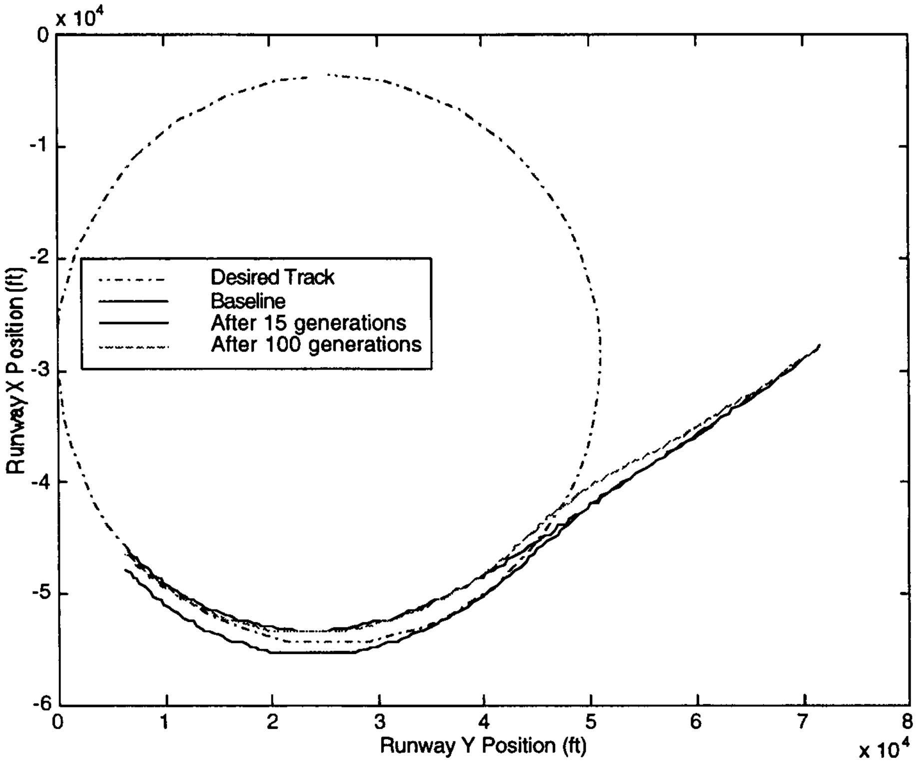

Figure 5: Tracking Error Guidance Performance Comparison

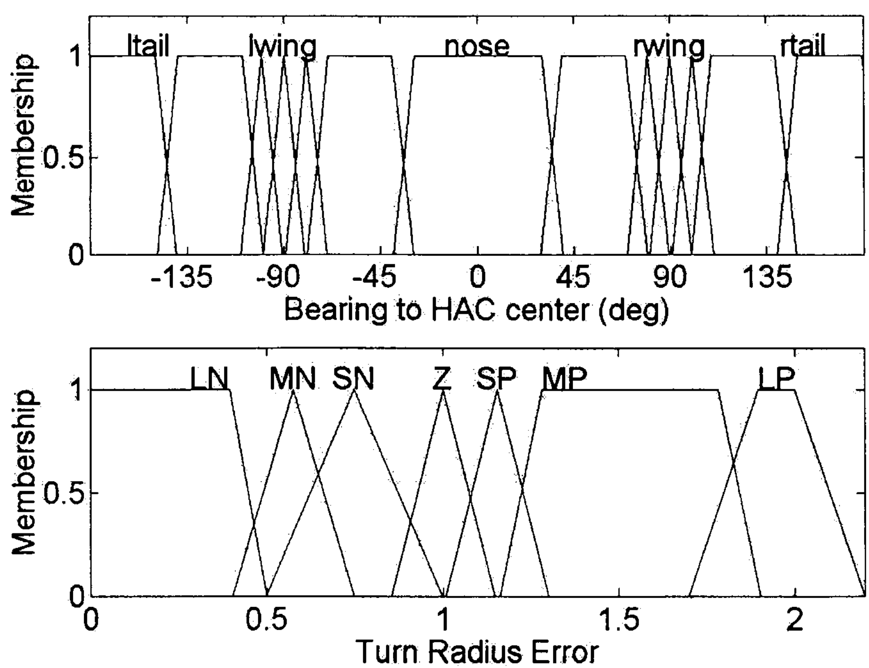

Figure 6: Baseline Membership Functions for Turn Radius Error and Bearing

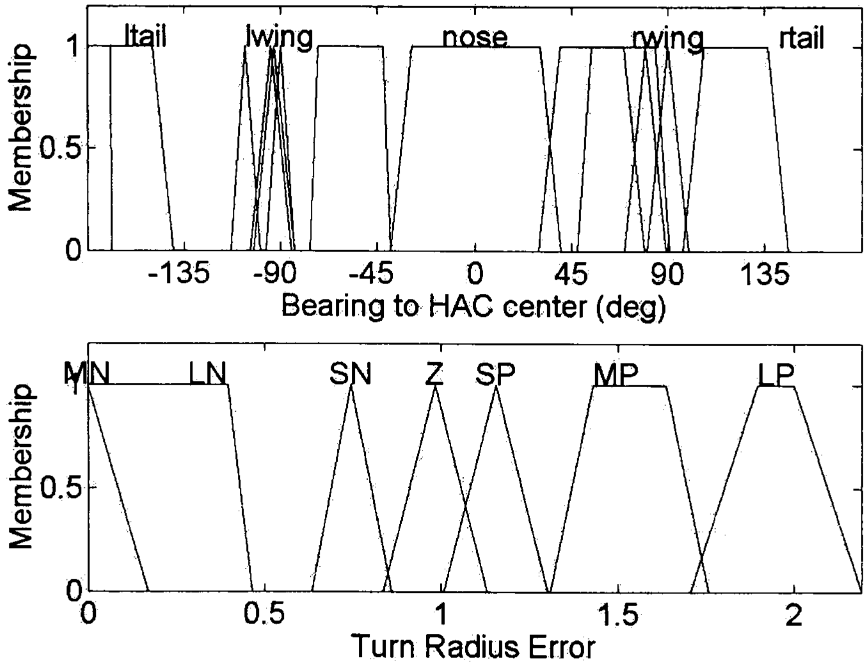

Figure 7: Membership functions for turn radius error and bearing after 15 generations

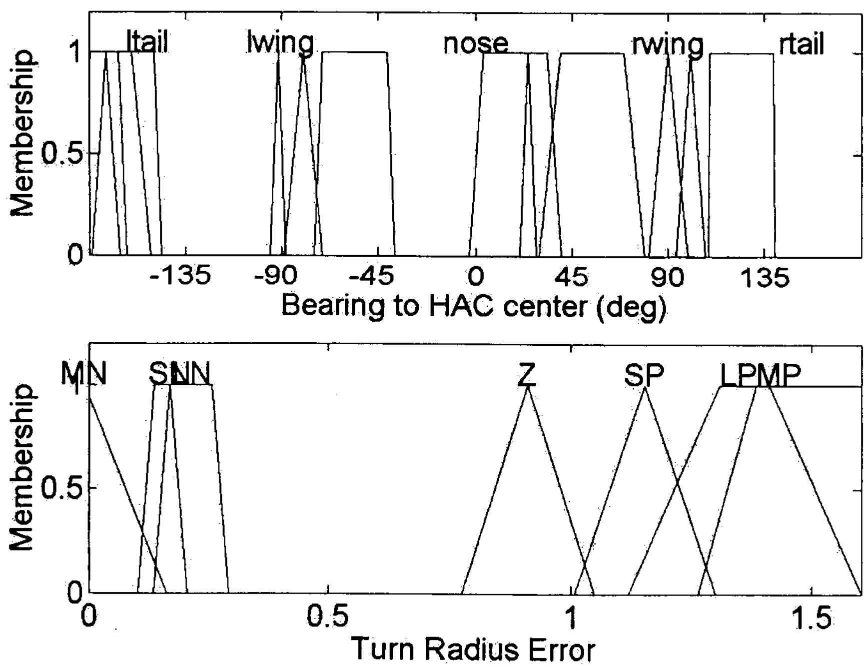

Figure 8: Bearing and Turn Radius Error Membership Functions after 100 generations

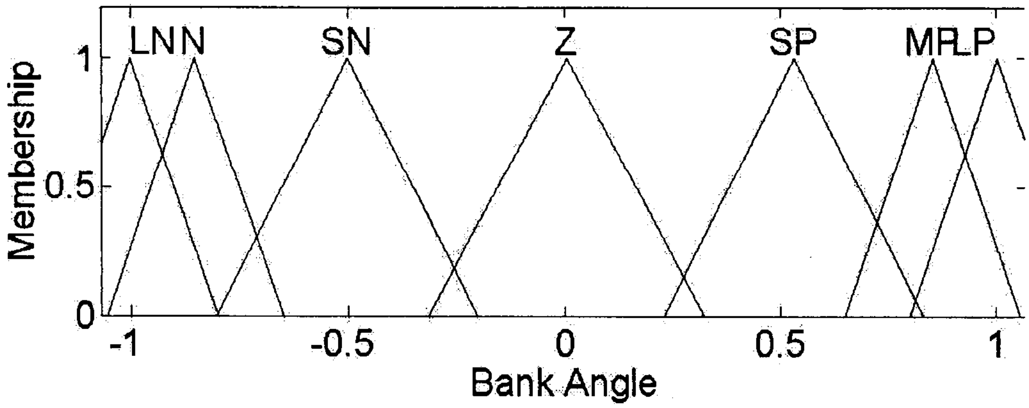

Figure 9: Baseline Bank Angle Membership Functions

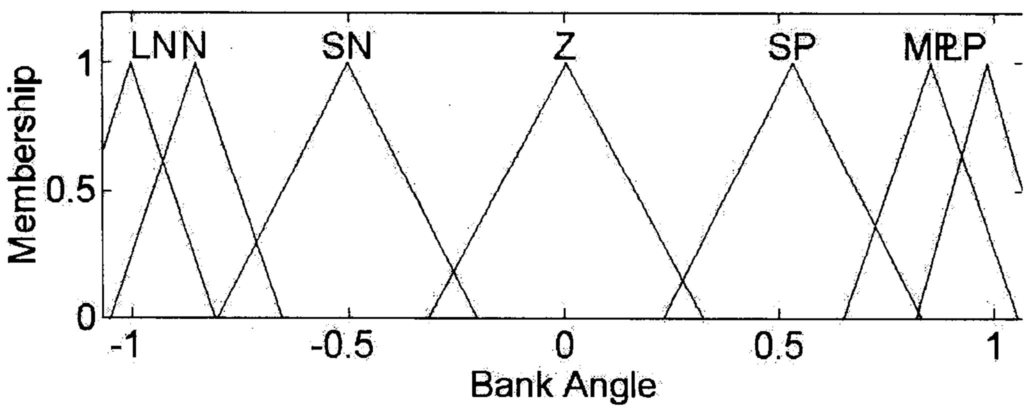

Figure 10: Bank Angle Membership Functions after 15 Generations

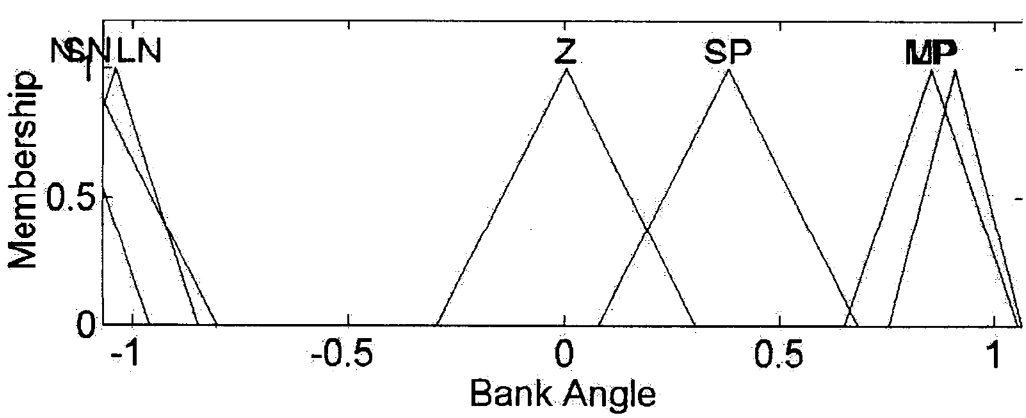

Figure 11: Bank Angle Membership Functions after 100 Generations

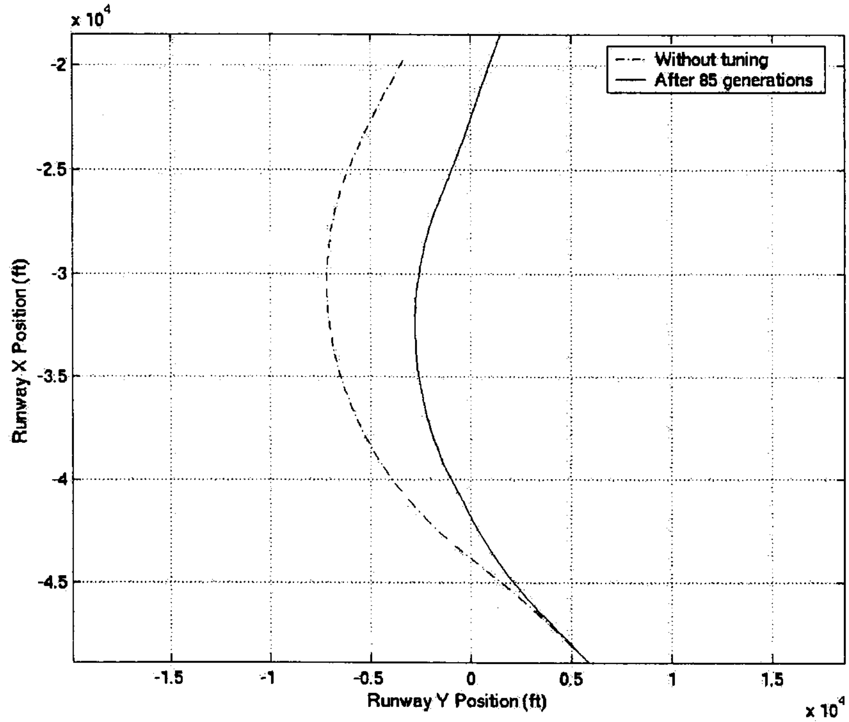

Figure 12: Improvement in Final Approach Alignment

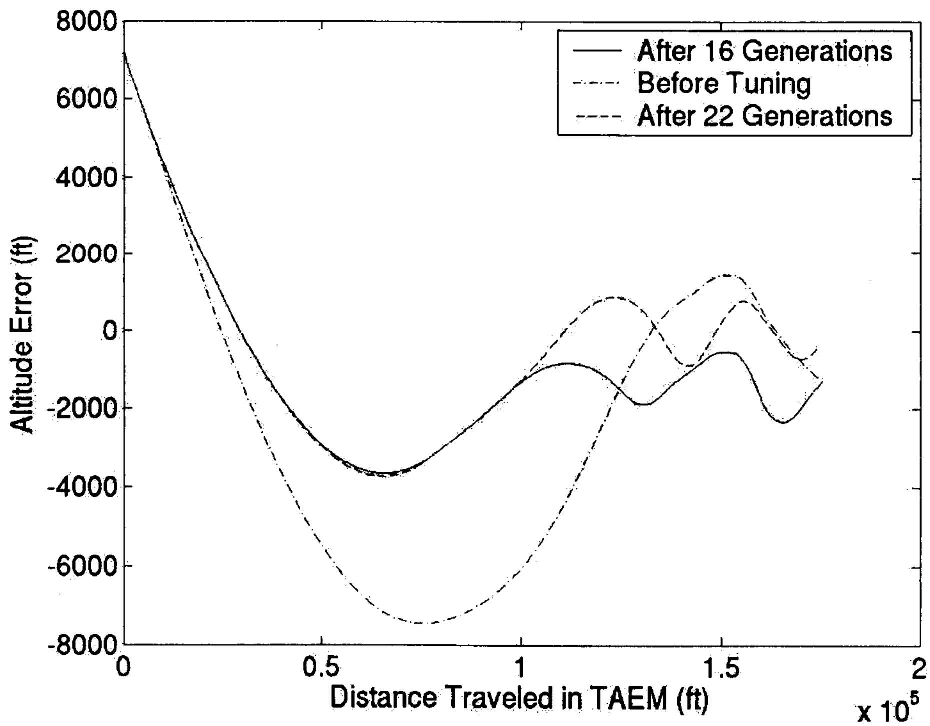

Figure 13: Improvement in Vertical Channel Guidance

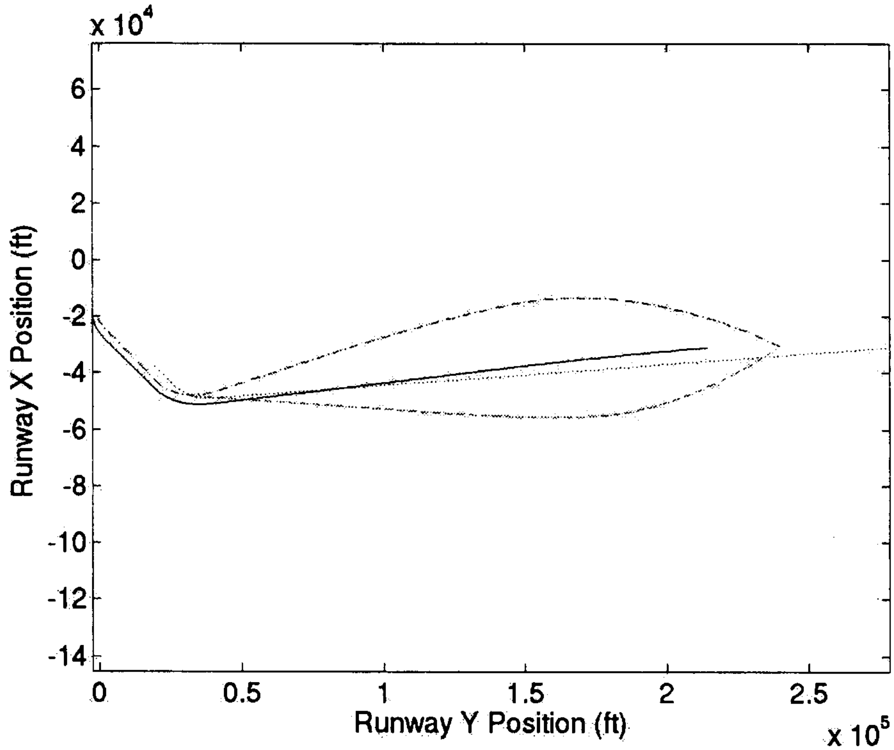

Figure 14: Ground Track for the Four Initial Conditions after Tuning the Trajectory Designer

Table 1: Initial Conditions for Trajectory Designer Training

Variable	IC 1	IC 2	IC 3	IC 4
x (ft)	-30614	-31038	-30587	-30974
y (ft)	239900	213840	239690	278610
ψ (deg)	-64	-94	-124	-94
$\dot{y}$ (ft/s)	-2966	-2958	-2945	-2960
h (ft AGL)	96500	87880	79480	87880
V (ft/s)	3000	3000	3000	3000
γ (rad)	-0.1076	-0.1076	-0.1076	-0.1076
$\bar{q}$ (psf)	133.6	213.8	319.3	213.8
Mach	3.02	3.04	3.06	3.04
$\dot{h}$ (ft/s)	-309	-268	-249	-258

Table 2: Desired End State

Coordinate	Desired Value
Runway X Position	-20000 (ft)
Runway Y Position	0 (ft)
Altitude	10000 (ft AGL)
Heading	Aligned with Runway

Table 3a: Evolution of Cost Function for Trajectory Designer (sum cost criterion)

Training	Cost for IC 1	Cost for IC 2	Cost for IC 3	Cost for IC 4
None	136.9383	292.7708	385.0947	288.4938
14 generations	35.5167	30.7042	455.3514	276.9252
28 generations	27.2860	28.6652	461.0296	264.2302
45 generations	27.3110	28.6318	457.6460	264.4193

Table 3b: Evolution of Cost Function for Trajectory Designer (max cost criterion)

Training	Cost for IC 1	Cost for IC 2	Cost for IC 3	Cost for IC 4
None	136.9383	292.7708	385.0947	288.4938
21 generations	331.9659	175.3205	260.7934	221.4817
42 generations	317.3839	199.2199	260.3012	220.6867

8. REFERENCES

1. Barton, G. H., ``Autolanding Trajectory Design for the X-34," AIAA Paper 99-4161, 1999.
2. Barton, G. H., ``New Methodologies for Assessing the Robustness of the X-34 Autolanding Trajectories," *Advances in the Astronautical Sciences*, Volume 107 Guidance and Control 2001, pp. 193-214.
3. Burchett, B. T., "Fuzzy Logic Trajectory Design and Guidance for Terminal Area Energy Managment," Submitted to the AIAA 42nd Aerospace Sciences Meeting, Reno, NV 2004.
4. Girerd, A., Barton, G., ``Next Generation Entry Guidance--Onboard Trajectory Generation for Unpowered Drop Tests," AIAA Paper 2000-3960, 2000.
5. Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning,* Addison-Wesley, Reading, MA, 1989.
6. Grantham, K. ``Adaptive Critic Neural Network Based Terminal Area Energy Management / Entry Guidance," AIAA Paper 2003-305.
7. Hanson, J. "A Plan for Advanced Guidance and Control Technology for 2nd Generation reusable launch vehicles," AIAA Paper 2002-4557.
8. Hanson, J. M. "New Guidance for New Launchers," *Aerospace America* , March 2003, pp. 36-41.
9. Karr, C. L., *Practical Applications of Computational Intelligence for Adaptive Control,* CRC Press, Boca Raton, FL, 1999.
10. Linkens, D. A., and Nyongesa, H. O., "Genetic Algorithms for Fuzzy Control, Part 1: Offline System Development and Application," *IEE Procedures on Control Theory Applications,* Vol 142, No 3, pp. 161-176.
11. Moore, T. E. ``Space Shuttle Terminal Area Energy Management," NASA Technical Memorandum 104744, Lyndon B. Johnson Space Center, 1991.
12. Wang, L. X., *Adaptive Fuzzy Systems and Control: Design and Stability Analysis,* Prentice Hall, Englewood Cliffs, NJ, 1994.

13. Air Force Manual 11-217 Volume 1, *Instrument Flight Procedures*, 29 December 2000, http://afpubs.hq.af.mil