Genetic Algorithms Training of Neural Nets for Aircraft Guidance and Control

Glenn Gebert and Murray Anderson
Sverdrup Technology, Inc., Eglin AFB, FL 32542

Johnny Evers
Air Force Research Laboratory (AFRL/MN), Eglin AFB, FL

Abstract

As airframe become more and more complex, and are called upon to perform increasingly stressful maneuvers, autopilots must be robust enough to adequately stabilize the airframe in the highly non-linear, strongly cross-coupled environments. Classic autopilot design can achieve stability throughout the flight envelope, but generally lack robustness for design and environmental changes. Guidance and control routines composed of a neural net architecture offer a promising ability to process multiple inputs, generate the appropriate outputs, and provide greater robustness. However, difficulty can arise in the training process of the neural nets. In the present study, a feed-forward neural net was used for the guidance and control routines on typical airframes. The neural nets were trained through genetic algorithms. The work attempts to model the biological process of the "thinking" aspect of the airframes by the use of a neural nets trained through natural selection as put forth in the Theory of Evolution. The present study produced an autopilot that learned to control its rates and maneuver (with a full six degrees-of-freedom) across an arena to a target.

Introduction

Aircraft and missile autopilots are becoming increasingly complex as the airframes are required to perform extreme maneuvers. Generally, the autopilots on current airframes employ a closed-loop architecture, feeding back the critical positions and rates required to maintain control. During extreme maneuvers, the aerodynamics, and subsequently the airframe response, becomes increasingly non-linear. Cross-coupling between the channels becomes significant, and classic autopilot design techniques fail in their ability to optimize the autopilot.

Researchers[1-6] have attempted to address the highly coupled autopilot issues by replacing the classic autopilot design with a neural net. The net can allow for full cross coupling between channels while maintaining stable flight mode. The apparent robustness of a neural net makes its application to flight control systems appealing. However, it is difficult to verify its stability throughout the flight regime.

Neural nets require training. During the training process the weights of the neural net are adjusted to achieve a particular set of outputs for a given set of inputs. Generally, this is accomplished by adjusting the weights of the net to match the outputs with the inputs for an existing dataset. It is then desired that the net will interpolate and extrapolate from the training dataset to more general situations. An alternate method for training the neural net is to adjust its weights by attempting to realize a particular goal for the system. In this case, the exact outputs for a set of inputs are not known. However, the entire system is encouraged to achieve a specific goal or set of goals. It is expected that this technique is less efficient. However, it is often the case of complex systems that the precise outputs for a set of inputs are not known and are therefore unavailable for the training process. The training of the neural net through the encouraging of specific goals strongly lends itself to the implementation of genetic algorithms.

The present analysis attempted to strongly model the processes of nature (according to Darwin's evolution theory) where the design of the flight control neural net is created and proved by pursuing a set of goals in a arena environment. For this study a population of geometrically similar aircraft was examined. All members of the population had the exact same size, geometry, mass properties, aerodynamic properties, and propulsion system. The "brains" of each aircraft is a neural net, which will position the control surfaces and throttle setting. The only differences between the aircraft are the weights of the neural nets, which were determined through a genetic algorithm.

The Arena

Each aircraft must fly through a tournament arena. Figure 1 shows a typical formation of an arena. The flat green plane represents the ground. Above and penetrating the ground is a system of (red) spheres. The spheres represent obstacles to reaching the goal, which is the single black hemi-sphere at ground level. The arena has dimensions of 132,000 ft by 132,000 ft along the ground. Each aircraft starts on the left-most side of the arena as shown in Fig. 1. Its initial altitude is a random value between 1,000 ft and 15,000 ft above the ground level, and it is in the middle 50% of the left side starting plane. There are a random number of obstacles from 3 to 30. The obstacles are positioned randomly within the arena at center-point altitudes from ground level to

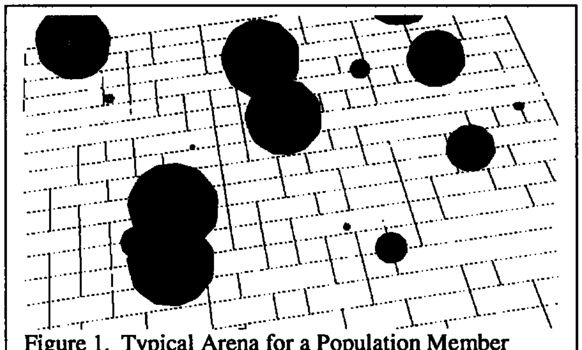

Figure 1. Typical Arena for a Population Member

15,000 ft. The radius of the obstacles varies randomly from 1,000 ft to 10,000 ft. The target is randomly located at least 90% across the arena.

Each aircraft sees a different arena than all other aircraft. By the luck of the draw, some aircraft fly through an arena with relatively few and/or small obstacles, while others obtain obstacle intense arena and perhaps a low starting altitude. The desire in continuously modifying the arena is to obtain a guidance/autopilot, which not only will navigate through a particular arena, but will learn to avoid all obstacles in any arena.

The Neural Net

For this application a relatively simplistic neural net was used to replace the typical guidance and control routines. Figure 2 shows a general construction of the feed-forward neural net.

Typical autopilot and guidance algorithms feedback multiple parameters to control and maneuver the vehicle. It is felt that these parameters fall into three categories, those being:

- Attitude Control
- Flight Stabilization
- Maneuvering Guidance

The parameters associated with each of these categories are

- Attitude Control – Angle of Attack α, Angle of Sideslip β, Flight Mach number M, Altitude z
- Flight Stabilization – Roll, Pitch, and Yaw Angular Body Rates, p, q, and r, respectively
- Maneuvering Guidance – Obstacle Proximity Function ϕ_0, and the Normalize Vector Pointing to the Target, $\vec{i}_g = x_g\vec{i} + y_g\vec{j} + z_g\vec{k}$, where $\vec{i}$, $\vec{j}$, and $\vec{k}$ are the usual Cartesian unit vectors and x_g, y_g, and z_g of the components of the $\vec{i}_g$ vector, which points from the airframe to the target.

It was felt that all parameters in a category should enter the net at a given level, but not all of the categories should enter at the same level. For example, the Attitude Control inputs could enter at the second level, while the Flight Stabilization and Maneuvering Guidance inputs might both enter at the first level. Since the best positioning of the entry of the inputs is not known in advance, it was decided to let the genetic algorithm determine the appropriate entry point.

There may be a large number of obstacles. Feeding in the position and size of each obstacle into the net would lead to an unacceptably large number of inputs. Rather, it was decided to replace the position/size information of the obstacles with a single function, which is

$$\phi_0 = \sum_{i=1}^{N_{ob}} \frac{1}{\sqrt{(x_{cg} - x_i)^2 + (y_{cg} - y_i)^2 + (z_{cg} - z_i)^2} - R_i} \tag{1}$$

where x_{cg}, y_{cg}, and z_{cg} are the current inertial coordinates of the airframe's center of gravity, N_{ob} is the number of obstacles, x_i, y_i, and z_i defines the location for the i^{th} obstacle, and R_i is the radius of the i^{th} obstacle. The obstacle function does not impart precise information on the location of each obstacle to the neural net, but it does approach infinity as the airframe nears an obstacle.

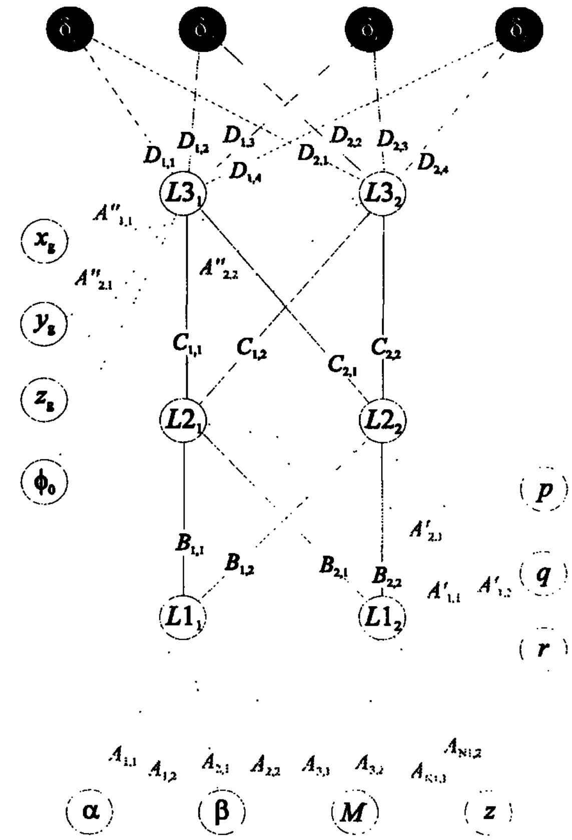

Figure 2. Schematic of the Neural Net Used to Control the Airframes.

Part of the "genetics" of the routine is to determine the best position for the entering of the various types of inputs. The interior nodes are the typical summation points of the lines into each node. The outputs of the neural net are δ_e, δ_a, δ_r, and δ_t, which respectively correspond to the elevator, aileron, rudder deflections, and the throttle setting. The topology of the neural net makes each output a linear combination of the input.

The neural net required a large number of weighting constants. In an effort to reduce the parameters of the problem, the net was constructed to have three interior layers with only two nodes each.

The angles and angular rates enter the net in terms of radians and radians per second, respectively. The input altitude z, which is calculated in units of feet is divided by 100,000. The target pointing vector is a normalized vector continuously pointing from the aircraft center of gravity to the target.

Thus, all of the inputs into the net are on the order of unity (except the Obstacle Proximity Function ϕ_0, which can tend to infinity). The output (where the angles are also measured in radians) of the net are also expected to be on the order of unity. The throttle setting δ_t is a normalized quantity where $\delta_t = 0.0$ corresponds to motor off, and $\delta_t = 1.0$ implies full throttle.

The Genetic Algorithm

All of the airframes examined had the same geometry, aerodynamics, propulsion, and mass properties. The only difference between them was the weighting of the connections of the neural net. These weighting constants became the genes on the chromosomes for the genetic algorithm searching technique.

Figure 3 shows the process in which the chromosomes are created. An upper and lower bound is specified for each parameter, as well as a resolution factor. The genetic algorithm routine determines the binary coding of the parameter. All of the individual binary values (the genes) of the parameters are concatenated together to form a single chromosome.

In the case of this study, the minimum value allowed for the weighting constants was −1.0 and the maximum was 1.0.

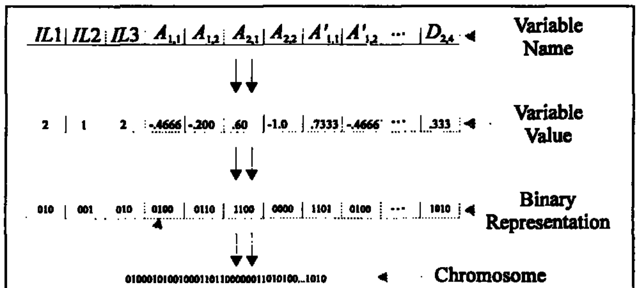

Figure 3. Process for Creating the Chromosomes

The resolution for each was approximately 0.15. The particular genetic algorithm has been documented in Refs. 7 and 8.

The Airframes

The aerodynamic and mass properties used in this study were approximately those of the general aviation airplane, NAVION, shown schematically in Fig. 4. Linear aerodynamics were generally assumed. The only non-linear aerodynamic terms were the total lift and associated drag. It was required to limit the lifting ability of the airframe, so that it would not ever increase its angle of attack to achieve better and better lift.

Thus, a rough estimation of the stall and post-stall behavior of the wings and its effect on the drag was made. The steady, no control surface deflection, lift was fit with a third order polynomial. The typical lift induced drag was used up to the stall angle. However, in the post-stall regime, both the lift and drag coefficients were assigned a sinusoidal component of a total normal force.

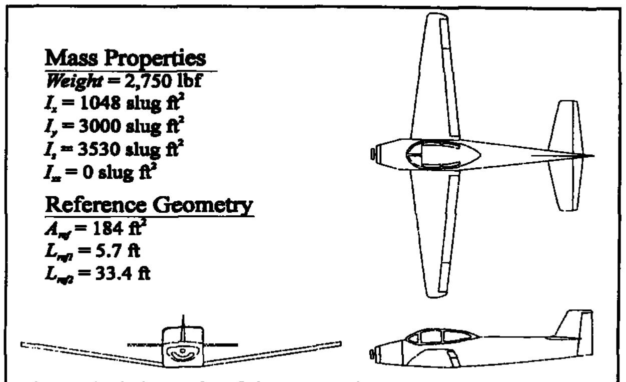

Figure 4. Schematic of the General Aviation Aircraft

The basic aerodynamics were of the form

$$C_L = f_{CL}(\alpha) + C_{L_q}\bar{q} + C_{L_{\delta e}}\delta_e \tag{3a}$$

$$C_D = f_{CD}(\alpha) + C_{D_{\delta e}}|\delta_e| + C_{D_{\delta a}}|\delta_a| + C_{D_{\delta r}}|\delta_r| \tag{3b}$$

$$C_Y = C_{Y_\beta}\beta + C_{Y_{\delta r}}\delta_r \tag{3c}$$

$$C_l = C_{l_\beta}\beta + C_{l_p}\bar{p} + C_{l_r}\bar{r} + C_{l_{\delta a}}\delta_a + C_{l_{\delta r}}\delta_r \tag{3d}$$

$$C_m = C_{m_\alpha}\alpha + C_{m_{\dot\alpha}}\bar{\dot\alpha} + C_{m_q}\bar{q} + C_{m_{\delta e}}\delta_e \tag{3e}$$

and

$$C_n = C_{n_\beta}\beta + C_{n_p}\bar{p} + C_{n_r}\bar{r} + C_{n_{\delta a}}\delta_a + C_{n_{\delta r}}\delta_r \tag{3f}$$

where C_L, C_D, and C_Y are the lift, drag, and side force coefficients, respectively, the C_l, C_m, and C_n are the rolling pitching, and yawing moments, respectively. A bar over a rate variable indicated that it is non-dimensionalized by multiplying by its reference length and dividing by twice the upstream velocity. Each coefficient on the right hand side of Eqs. (3a-f) was constant and defined in Ref. 9. The single exception was the variable C_{n_β}, which apparently has a sign error in Ref. 9 and will be discussed further later.

The angle of attack dependence for C_L was defined as

$$f_{CL}(\alpha) = \begin{cases} \frac{\alpha}{|\alpha|}0.817\cos\alpha & \alpha + \alpha_{L=0} < -16^\circ \\ 5.65(\alpha-\alpha_{L=0}) + 5.02(\alpha-\alpha_{L=0})^2 - 27.91(\alpha-\alpha_{L=0})^3 & |\alpha+\alpha_{L=0}| < 16^\circ \\ 1.227\cos\alpha & 16^\circ < \alpha + \alpha_{L=0} \end{cases} \tag{4a}$$

and for the C_D,

$$f_{CD}(\alpha) = \begin{cases} \frac{\alpha}{|\alpha|}0.817\sin\alpha & \alpha + \alpha_{L=0} < -11^\circ \\ 0.006 + 0.015C_L^2 & |\alpha+\alpha_{L=0}| < 11^\circ \\ 1.227\sin\alpha & 11^\circ < \alpha + \alpha_{L=0} \end{cases} \tag{4b}$$

where $\alpha_{L=0}$ is the zero lift angle of attack.

The propulsion systems on the airframes have maximum thrust levels, and fuel consumption consistent with a propeller driven general aviation airplane.

Results

A population size of 200 members was chosen. The weighting constants of each member were defined by their genetics. Originally, the genes were randomly determined. The original single criterion in selecting the superior members of the population was their final distance to the target. Since the original weighting constants were randomly generated, the first generation did not perform very well. Figure 5 shows the trajectory (the blue three-dimensional curve) of a typical member of the first generation.

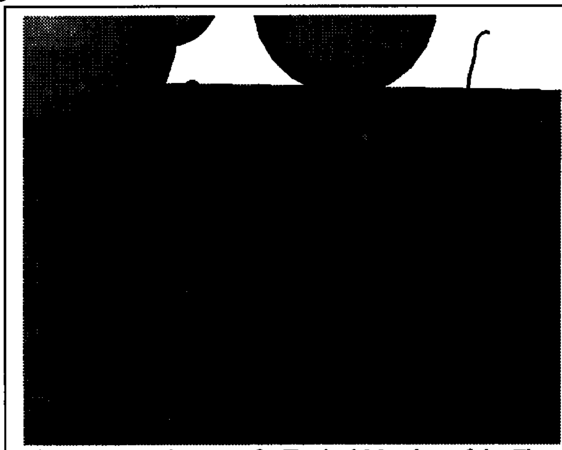

Figure 5. Trajectory of a Typical Member of the First Generation

The original analysis continued to select superior population members based solely on their ability to impact the ground near the target. Of course, no information was given to the aircraft that it was desired to avoid the ground, avoid the obstacles and approach the target. However, being allowed to produce offspring was the reward of those that did.

Following the first 100 generation, it was observed that the superior aircraft were the ones that were lucky enough to start from the highest altitude, set the control surfaces hard over, and tumble more or less ballistically as far as possible across the arena. It was likely, that in time a superior set of aircraft would evolve to a more elegant solution. However, it was deemed that some encouragement could also be provided. A second goal was established which rewarded population members that controlled their angular rates. The following function was established,

$$f_{rate} = \frac{1}{T}\int_0^T \sqrt{p^2 + q^2 + r^2}\,dt \tag{8}$$

where f_{rate} is the average rate function, t is time, and T is the total time of the flight. In addition, to the goal of reaching the target, successful population members minimize the f_{rate} function.

Clearly, some maneuvering is desirable to negotiate to the target, and avoid obstacles. However, as an instructional tool. It was felt that population members that could control their attitude should be encouraged first with reaching the target as a secondary goal. Once the population evolved to the point where it could control its attitude, then the goal of reaching the target could be reemphasized.

Figure 6 shows the best performing member of the population after 100 generations, where the minimization of f_{rate} was strongly emphasized and reaching the target was the secondary goal. The airframes learned rather quickly to sets their control surfaces to stop the tumbling. Further, the active control system damped out the phugiod oscillation quicker than the just stick-fixed aerodynamic damping. It was at this point that a problem in the aerodynamic model of the six Degree-of-Freedom (6-DOF) simulation was discovered. The aerodynamics associated with the airframes were taken from Ref. 9, and modified only the lift at drag at the high angles of attack. The remaining aerodynamic coefficients were linear, and the values found in Ref. 9. The derivative of the yawing moment with respect to the angle of sideslip C_{n_β} was give as -0.071 in Ref. 9, which has an apparent sign error. The negative sign in the coefficient makes the airframe statically unstable in yaw. The remarkable thing is the neural net autopilot was able to stabilize the airframe in spite of its static instability.

Figure 6. Trajectory of the Best Performer after 100 Generations of Encouraging Rate Control

Even though the neural net was able to control the unstable airframe, it was felt that that proper aerodynamics should be examined. Thus, the sign was changed on the C_{n_β}, and the analysis started again from the first generation.

With the correct aerodynamics, the results after 100 generations looked quite similar to those shown in Figure 6. Since the airframes had largely learned to control their rates, the emphasis was changed to encourage close proximity to the target, and the rate control was changed to a secondary objective.

The trajectory shown in Fig. 7 is that of the best performer after 300 generations. The airframes had largely retained their ability and desire to control their rates. However, all successful population members learned to maintain a zero command for rudder and aileron deflections. They learned how to adjust their elevator and throttle to fly across the arena to reach approximately the correct longitudinal position of the target, but no corrections were made for lateral adjustments.

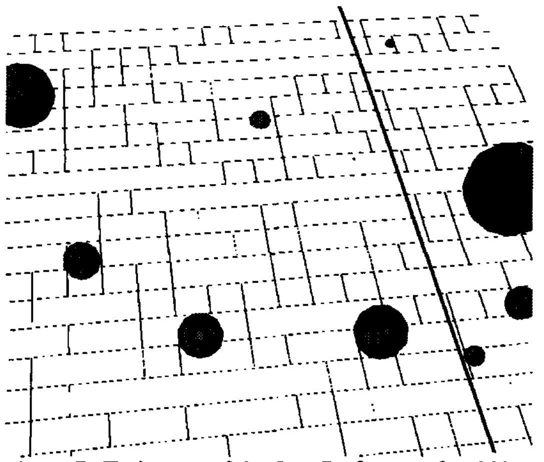

Figure 7. Trajectory of the Best Performer after 300 Generations.

Such was the status of the airframes for a large number of generations. The local optimization of commanding no lateral motion reigned superior to other tested options. The airframes got somewhat better at finding the correct longitudinal location, and would learn to nose over if flying to far, but no success was found in commanding aileron and/or rudder. Finally, at approximately the 1,600th generation, successful airframes emerged that command lateral maneuvers.

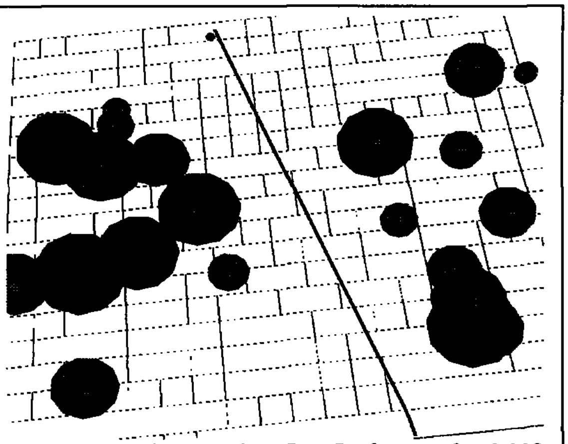

Figure 8. Trajectory of the Best Performer after 2,000 Generations.

Figures 8 and 9 show the trajectory of the best performer after 2000 generations. The best performers learned to provide an early course correction to points towards the target, and upon flying over the target, they learned to nose over.

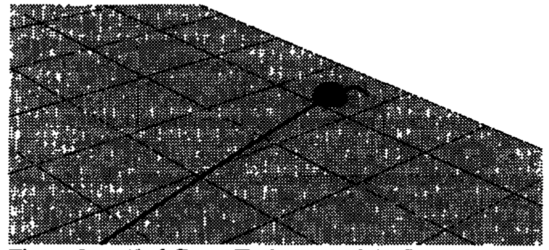

Figure 9. End Game Trajectory of the Best Performer after 2,000 Generations.

The airframes have not seemed to learn to avoid obstacles. The successful airframes are those that obtain an arena that presents no obstacles between the starting position and the target. It will likely require a large number of additional generations to fine-tune the targeting and obstacle avoidance of the neural net.

The combination of 40 weighting constants (each having a possibility of 16 values) and three input position constants (each having a possibility of 3 values) produces a total of 3.94×10^{49} different configuration to the neural net. After 2,000 generations with 200 members on the population, only 400,000 different combinations have been examined. The genetic algorithm has produced rather remarkable results, considering the small amount of the sample space that has been examined. Clearly, further refining is required to find the more optimal solution.

Figures 10 and 11 respectively show the convergence histories for the target miss distance and the rate function of the population versus the generation. Shown on both charts is the population average performance as well as the best performer of the population. Results for each generation were computed for the first 300 generations, and then for every 100 generations after that.

Note in Fig. 10 that there is a significant initial drop off in the miss distance of both the best performer and the population average. The initial drop is due to the airframes learning to control their rates as shown in Fig. 11. It can be seen in Fig. 11 that at the 100th generation the primary objective was changed to encourage close approaches to the target. Even though the rate function begins to increase after 100 generations, the miss distance is not significantly affected.

There is a small decrease in miss distance (average and best) following the 300th generation. The decrease is largely due to a learning of the airframes to nose over if they are extending too far down range. Finally, after the 1,600th generation, the best performers start learning to maneuver toward the target. It is noted that the 1,900th generation loses some performance, but this was due to obstacle

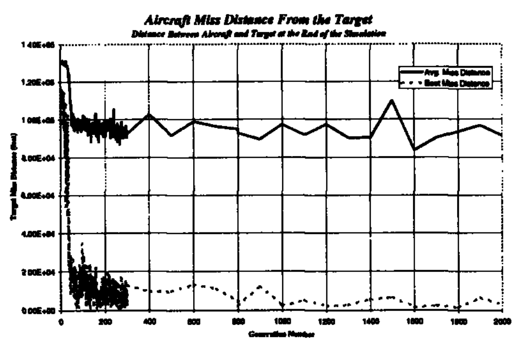

Figure 10. Miss Distance from Target Convergence History.

placement between the airframe and the target for the best performers at that generation.

The trends of the average population members for both the miss distance and the rate function are very encouraging. With continued execution of the genetic algorithm, better solutions are likely.

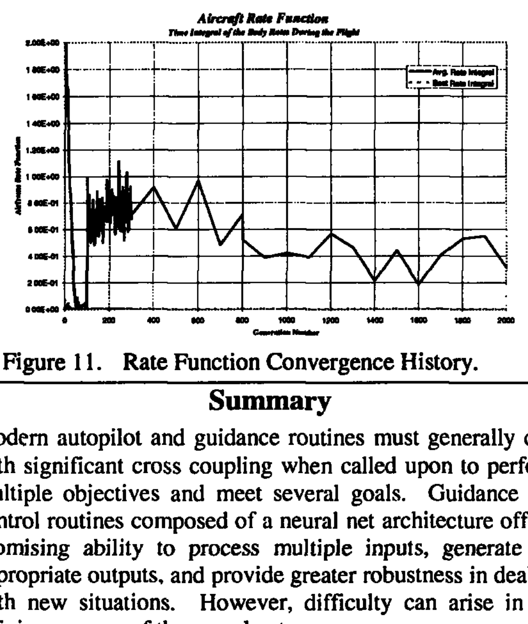

Figure 11. Rate Function Convergence History.

Summary

Modern autopilot and guidance routines must generally deal with significant cross coupling when called upon to perform multiple objectives and meet several goals. Guidance and control routines composed of a neural net architecture offer a promising ability to process multiple inputs, generate the appropriate outputs, and provide greater robustness in dealing with new situations. However, difficulty can arise in the training process of the neural nets.

In an effort to mimic processes associated with Darwin's Theory of Evolution, a genetic algorithm was employed to facilitate the learning of the neural net. A population of similar aircraft was generated. Each aircraft had the same aerodynamics, configuration, mass properties, and propulsion system. The aircraft's guidance and control algorithms were composed of a neural net receiving flight attitude, position and rate information, outputting control surface deflections and throttle setting. The weights of the neural net were the genes associated with the genetic algorithm search, which were passed on and modified in subsequent generations. The genetic algorithm statistically preferences the best performers to reproduce offspring and pass on their general characteristics to the next generation.

In the present study, the best performers were the aircraft that could control their angular rates and/or impact the ground within the closest distance from an arbitrarily placed target, while negotiating through an arena of spherical obstacles. After 2,000 generations, the best performers learn to control their rates very well, and could set their throttle and elevator to damp the phugoid motion and traverse the distance of the arena. Further, they developed a general notion to deflect rudder and/or aileron early in their trajectory to maneuver toward the target, and nose over onto the target when approaching too high. However, little learning to avoid obstacles had occurred.

Further generations are required to optimize the solution. The sample space of problem is so large that the genetic algorithm continues to find local extrema to the guidance and control, but continues to search for the global optima. From the work thus far, it appears that a neural net can be trained to meet the desired goals. However, it is likely to require tens of thousands of generations to reach that point. Although the computational expense is significant, the resulting guidance and control routine should be able to avoid arbitrary obstacles and guide to a randomly located target, which will have tremendous applications for weapon system implementation in urban (cluttered) environements.

References

1. Bateman, A. J. D., Ward, D. G., Barron, R. L., Whalley, M. S., "Piloted Simulation Evaluation of a Neural Network Limit Avoidance System for Rotorcraft", Paper 99-4252, *AIAA Atmos Flt Mech Conf*, Portland, OR, Aug. 1999
2. McFarland, M. B., "Augmentation of Gain-Scheduled Missile Autopilots Using Adaptive Neural Networks", Paper 98-4491, *AIAA Guide, Nav and Cntrl Conf*, Boston, MA, Aug. 1998: p. 1786-1792.
3. Feng, X., Lin, C., Yu, T., Coleman, N., "Intelligent Control Design and Simulation Using Neural Networks", Paper 97-3528, *AIAA, Guide, Nav and Cntrl Conf*, New Orleans, LA, Aug. 11-13, 1997
4. McFarland, M. B.; Calise, A. J., "Neural Networks for Stable Adaptive Control of Air-to-Air Missiles", Paper 95-3315, *AIAA, Guide, Nav and Cntrl Conf*, Baltimore, MD, Aug. 7-10, 1995
5. Werbos, P. J., "Neural networks and flight control - Overview of Capabilities and Emerging Applications", Paper 95-3272, *AIAA, Guide, Nav and Cntrl Conf*, Baltimore, MD, Aug. 7-10, 1995
6. Schaechter, D. B., Rauch, H. E., "Applications of Neural Networks to Control Systems", Paper 92-4389, *AIAA Guide, Nav and Cntrl Conf*, Hilton Head Island, SC, Aug. 10-12, 1992
7. Anderson, M.B., and Lawrence, W.R., "Launch Conditions and Aerodynamic Data Extraction By An Elitist Pareto Genetic Algorithm", Paper 96-3361, *AIAA Atmos Flt Mech Conf*, San Diego, CA, July 1996.
8. Anderson, M. B., Burkhalter, J.E., and Jenkins, R.M. "Missile Aerodynamic Shape Optimization Using Genetic Algorithms", Paper 99-0261, *AIAA Aerospace Sci Mtg*, Reno, NV, Jan. 1999.
9. Nelson, R. C., Flight Stability and Automatic Control, 2nd Ed., McGraw-Hill, Boston, MA, 1998, p. 400-401.