DESIGN OF A PILOT-ACTIVATED RECOVERY SYSTEM USING GENETIC SEARCH METHODS

G. D. Sweriduk* and P. K. Menon†
Optimal Synthesis Inc.
470 San Antonio Road, Suite 200
Palo Alto, CA 94306-4646

M. L. Steinberg‡
Aeromechanics Division
Naval Air Systems Command
Patuxent River, MD 20670

Abstract

Control design tasks often require the use of trial-and-error search methods to obtain a satisfactory solution. Depending upon the nature and the number of "tuning" parameters or functions, the search process can be very discontinuous and nonconvex. The genetic search methods are a recently developed family of techniques for optimization which offer certain advantages over other techniques. These include greater freedom in defining cost functions and constraints, and the ability to automate the design process. Most notably, though, is the ability to construct new control laws and the potential to generate non-intuitive solutions as well. This paper demonstrates the application of genetic search methods to design a Pilot-Activated Recovery System (PARS) for a modern fighter aircraft. The PARS is a guidance law that transfers the aircraft from any initial attitude to a wings-level, nose-up, recovered flight condition. This system is useful in cases of pilot disorientation. A 6 degree-of-freedom nonlinear model of a modern, high-performance aircraft is used for design. The genetic search seeks to produce nonlinear feedback functions to meet the specified goals and constraints. This intricate problem highlights some of the advantages of this emerging search technique.

I. Introduction

Various methods are available for designing guidance and control laws for aerospace vehicles. Sometimes, guidance laws are obtained through optimal control theory. In order to use this approach the engineer must define some performance index. The performance index must have certain properties, including continuity, smoothness, and a practical limit on the complexity, so that it will lend itself to mathematical manipulations. Often, it can be difficult to accurately capture the true design goals in a single performance function. In aerospace applications there is often a large set of design goals. The control design process relies on the intuition of the engineer to select an appropriate performance index. However, it may not lend itself to further analysis. Automating this process would have considerable benefits.

Traditional optimization methods are strictly numeric, so that they work only on parameters, while the guidance or control law structure remains fixed. However, the recent development of genetic methods offers the possibility of broadening the scope of computer-based optimization to include explicit functions and expressions. Two distinct advantages of genetic methods are the capability to define non-traditional performance indices, and the capability to manipulate the control law itself, rather than just parameters. Moreover, these methods do not require continuity or smoothness of the performance index or constraints.

This paper demonstrates the use of genetic search methods to design a Pilot-Activated Recovery System (PARS) for a modern fighter aircraft. The PARS is a guidance law that transfers the aircraft from any initial attitude to wings-level, nose-up, recovered flight. The PARS logic is designed to work through the aircraft's command augmentation system. The overall system is shown in Fig. 1. Several studies have considered the PARS problem[1,2], or the related ground collision avoidance system (GCAS)[3]. References 1 and 2 described a system in which separate maneuvers were defined, depending on the initial condition. The goal of the present work is to obtain a single guidance law that will work for arbitrary initial conditions, excluding extreme conditions such as spins. What makes this problem a good example for genetic methods is that the objective is to achieve recovery without violating the constraints. Constraint violations are determined by examining time histories. Therefore, this problem does not lend itself well to customary methods of analysis. The genetic search process is provided with mathematical functions and operations through the initial population, which it uses to synthesize new control laws, and simulates the closed-loop system to obtain a fitness value (i. e. cost).

---

* Research Scientist, Senior Member, AIAA
† President, Associate Fellow, AIAA
‡ Flight Control Engineer, Senior Member, AIAA

Section II will discuss genetic search methods, the problem will be defined in Section III, and Section IV will present some results.

II. Genetic Algorithms and Genetic Programming

The basic concept of genetic algorithms has been around for a long time, but the modern revival began with the work of Holland[4], who showed how the evolutionary process could be applied to artificial systems. The genetic algorithm is a mathematical algorithm which transforms a set (population) of mathematical objects (members) into a new set using operations similar to the process of natural selection, as described by Charles Darwin in his well-known treatise[5]. The main operations are reproduction, crossover, and mutation. Each new population is called a generation. The fitness of each member of the current generation is evaluated according to some specified function. The members with better fitness are more likely to be selected to be carried over to the next generation (reproduction) or used to create offspring (crossover) which will be included in the next generation. Members with poor fitness are more likely to be eliminated from the population. Members can also be selected at random and altered (mutation).

Throughout the 1980’s, extensions to the standard genetic algorithm were proposed[6,7]. In the standard genetic algorithm the members are usually fixed-length strings. The strings are made up of binary numbers which can represent real numbers or actions (i.e. fast vs. slow, high vs. low, etc.). A string may therefore represent a set of numbers or a sequence of actions. By breeding and mutating these strings, new combinations are formed, and the new strings are evaluated for fitness. However, the length of the string, and the structure of the solution, is always fixed in genetic algorithms. In the more recent genetic programming methodology, the complexity of the members undergoing adaptation is much greater.[7] The members may be mathematical expressions or rules such as logical operators. In this way a genetic algorithm can be used to create computer programs to solve a specific problem. According to Ref. 7, “...the structures undergoing adaptation in genetic programming are active. They are not passive encodings of the solution to the problem. Instead, given a computer on which to run, the structures in genetic programming are active structures that are capable of being executed in their current form.”[7] It is this process of creating programs that leads to the term “genetic programming.” There are variations of the classical genetic algorithm and genetic programming, but they all share the basic concepts, so they are referred to here collectively as genetic search methods.

In recent years, several authors have applied genetic methods to flight control problems. In Ref. 8, genetic search methods were used to design nonlinear control laws for a longitudinal model of an A-4 aircraft. Both autopilot controllers and guidance laws were developed using this methodology. In Ref. 9 fixed-order dynamic compensators were designed with the objective of pole placement. In Ref. 10 genetic algorithms were used to design decentralized controllers by matching performance with existing, centralized controllers. Classical control was used with genetic algorithms in Refs. 11 and 12. A modified LQR design was used in Ref. 13, and in Ref. 14 H-infinity controllers were designed using genetic methods. In these problems only parameters were selected in the design. The present study will use a genetic search to design a PARS, where an assortment of mathematical functions will be used to construct nonlinear feedback guidance laws.

III. Problem Definition

This section will describe the model used, the fitness, the structure of the system, and the procedure for obtaining a solution.

Model

The model used for design is a 6 degree-of-freedom simulation of an modern fighter aircraft and is implemented in the system simulation environment SIMULINK®. The rigid body is modeled while flexible modes are not, although the aerodynamic model does partly account for flexibility. Actuator and sensor models were excluded. A command augmentation system, described in Ref. 15, is used to control the aircraft. Commands to the closed-loop system are normal acceleration and roll rate; an additional loop is included to control velocity via the throttle. By convention, the positive z direction is downward, and therefore, straight and level flight is described by -1 g normal acceleration.

Fitness

In Ref. 1, the objective was to bring the aircraft from any initial attitude to a wings-level, nose-up, recovered state. “Nose-up” was defined by the final flight path angle being between 1 and 10 degrees. In this work, the “recovered state” is interpreted to mean zero roll angle, 1 g load factor (zero inertial normal acceleration), zero roll rate and pitch rate, and the pitch angle close to the flight path angle, which can also be stated as having a small angle of attack. Since angle of attack and load factor are related for small perturbations, the use of normal acceleration in the fitness calculation should keep the pitch angle close to the flight path angle. No guidelines were given regarding the final airspeed and altitude, so these were left open, within reasonable limits.

For the benefit of the pilot, commands should not be too severe. The limits specified in Ref. 1 and used here were as follows:

roll rate: 180 deg/sec max.
roll acceleration: 285 deg/sec^2 max.
load factor: 5 g max., 0.5 g min.
load factor onset: 4 g/sec max.

As a practical matter, the following constraints were also imposed:

dynamic pressure: 50 psf min
altitude: 50,000 ft. max., 100 ft. min.

In order to use any optimization technique, some performance criterion or criteria must be established. Performance objectives were not easy to quantify in this case, since the recovery process was mostly a problem of constraint satisfaction. However, in the event that more than one member satisfies all constraints, some way is needed to distinguish among them. It was decided that attitude error and altitude change during the recovery would form the basic fitness functional. This was realized with the time-weighted integral of error squared for the attitude variables. In each channel the function is

$$\int_{t_i}^{t_f} \frac{t}{w}\left(x - x_f\right)^2 dt$$

where w is a weighting function, x is the variable in question, and x_f is the desired final value. Including the time t in the function effectively increases weight as time increases, which favors a quicker settling time. The goal is to achieve wings-level, nose-up flight, so the variables which would be integrated are roll angle, normal acceleration, and flight path angle. Altitude change was included in the fitness as well, the penalty being the integral of the difference between the current altitude and the altitude at the time of initiation. The fitness value is determined by simulating the closed-loop system and evaluating the integrals.

In addition to the basic fitness, penalties were added for exceeding limits (constraint violations). The penalty functions were:

roll angle	$(50*\phi(t_f))^4$
flight path angle	$(50*(\gamma(t_f)-0.0873))^4$
roll rate	$(100*p(t_f))^4$
pitch rate	$(100*q(t_f))^4$
max acceleration	$1000*(nz_{max} + 5.0)^2$ if $nz_{max} < -5.0$ g
min acceleration	$1000*(nz_{min} + 0.5)^2$ if $nz_{min} > -0.5$ g
max onset	$1000*(dg/dt_{max} - 4.0)^2$ if $dg/dt_{max} > 4.0$ g/s
max altitude	1000 if max(h) > 50000 ft.
min altitude	1000 if min(h) < 100 ft.
dynamic pressure	2000 if min(qbar) < 50 psf

where angles and rates are in radians, and t_f indicates final time. It is evident from this formulation that the objective is to minimize the fitness. If altitude goes below zero the simulation is halted and the fitness is assigned a very high value.

The PARS block shown in fig. 1 is composed of several subsystems, shown in fig. 2. The PARS logic block, implemented as an S-function in SIMULINK®, uses a nonlinear function determined by the genetic search to compute a normal acceleration command, and also determines the desired roll angle and velocity. The normal acceleration command is fed through a second-order filter to help shape the command. The roll angle is fed into a roll rate command generator which uses a first-order model and limits. The velocity command goes straight to the velocity controller, which is just a proportional control law, with limits. Since the engine response is relatively slow, there is no need to shape the command. Each of these subsystems will be discussed in more detail in what follows.

<u>PARS logic</u>

The logic block contains the control law that comes from the genetic search. The control law specifies the normal acceleration, but the logic determines the other commands based on the acceleration command. The PARS logic is set up so that whatever function is supplied, the normal acceleration command will always be negative in sign, or what pilots would call positive gs, since acceleration in the opposite direction is undesirable. A conditional statement is used that makes the acceleration command -1 when the value of the flight path angle, γ , is between 1 and 8 deg., which is the desirable range for the recovered state. The desired roll attitude is determined according to the value of γ. The roll command will roll the aircraft upright if the aircraft is descending and will roll inverted if the aircraft is ascending. If the desired roll angle is ±180 deg., a small bias of ±10 deg. is added to prevent dithering around 180, where the sign of the roll angle changes. The roll command tries to coordinate roll attitude with the anticipated normal acceleration command. Once the aircraft reaches its desired roll attitude, the normal acceleration command will be in the correct direction for recovery.

Normal acceleration control

The command generated by the PARS logic is fed into a filter to help shape the command. The poles of this filter have a natural frequency of 3 and a damping ratio of 0.9; the DC gain is 1. This filter was included mainly because the command augmentation system that was used permitted higher onset rates and overshoots than the specifications would permit.

Roll rate control

To develop a control law for the roll rate which covers the entire envelope would require a very complicated function, because roll rate performance varies considerably over the envelope. If a genetic search had been used, more iterations would have been required and more simulations per iteration would have been necessary. Since the limits were already known, these were imposed in the PARS setup via the command generator. The command generator is shown in fig. 3.

The input is a desired roll angle which comes from the PARS logic block, either 0, +170, or -170 deg.; the 10 degree offset from ±180 is used to prevent dithering where the angle changes sign. A first-order model of the desired roll rate response, with a pole at -4, represents the aircraft response. The output of this model goes through a rate limiter to enforce the roll acceleration constraint. The output of this block goes into the limit block, which is detailed in fig. 4. The output of the limit block is fed to a relational operator, and also to a product block, where it is multiplied by the sign of the rate input. The relational operator decides which signal has the smallest magnitude and controls the switch block to pass either the input rate signal or the limit signal with the appropriate sign. The output of this block is the roll rate command, which is fed to the command augmentation system and to an integrator block, whose output will be the estimated value of the roll angle. This estimated value is fed back to the input of the first-order model via a proportional gain. This gain was selected to give a satisfactory roll response.

Speed control

The velocity is controlled with the throttle using a proportional gain on velocity error. The commanded velocity is a fixed value which is a rough estimate of the corner velocity at 10,000 ft, since all maneuvers in this study were performed below that altitude. If and when the maneuver is completed, the PARS block makes the velocity command equal to whatever the velocity is at the end of the maneuver. The controller is shown in fig. 5

Parametrization

The initial population is comprised of the basic operations of addition, subtraction, multiplication and division, and assorted mathematical functions including trigonometric and inverse-trigonometric functions, hyperbolic and inverse-hyperbolic functions, natural and common logarithms, exponentials, exponents, square root, and absolute value. Variables which might be useful for the control law and are assumed to be available for feedback are also included in the initial population. These variables include altitude (h), altitude rate (hdot), flight path angle (gamma), initial flight path angle (gamma_i), total velocity (Vt), Mach number (M), dynamic pressure (qbar), and pitch rate (q). The integers 1 - 9 are represented by a1 - a9, powers of ten from -4 to +4 are represented by b0 - b8, and -1 is represented by m1; addition and multiplication of these variables can be used to represent any number. A few of the members in the initial population were designed to be able to recover, although not optimally, while the majority of members were formed arbitrarily, simply to introduce specific functions and constants into the population. It is essential with genetic search methods to introduce as much variability into the initial population as possible, since the final result can be comprised of only those terms, or “traits”, which appear in the initial population.

Algorithm

Figure 6 illustrates the basic procedure for designing the PARS. The algorithm was implemented in the MATLAB® programming environment. A MATLAB® toolbox was developed which provides functions for carrying out the genetic operations shown[16]. In SIMULINK® it was a simple matter to add integrators to compute the components of the fitness function as the simulation was running. It was also simple to add the additional penalties by evaluating the output data from the simulations.

IV. Results

For the genetic search, three initial conditions were used to evaluate each member: 10,000 ft, M 1.2, 57.3 degree inverted dive; 5,000 ft, M 0.6, 57.3 degree inverted dive; 5,000 ft, M 0.6, 40 degree climb. The simulation in each case lasts 20 seconds. Three runs were conducted using purely random selection of members for crossover. The initial populations contained 20 – 30 members that were constructed mostly randomly from the various mathematical functions. In each run the population size is limited to 500 members. The number of generations was between 2100 and 3000 for the three runs. The results were:

run	guidance law	fitness
1	(γ*γ + a3 * a5) / h * hdot	2114
2	sqrt((a4*gamma*b4*a4*(a1*b4+sin((b5*a2))*b1/(hdot*hdot/b8*h*a1*a4))/(a1*h)*hdot*b4))*a2	4017
3	(a7*b4 + a1*b5)*hdot / h	3444

The responses of the best design at various initial conditions are shown in the figs. 7-12. It can be seen in fig. 7 that the normal acceleration exceeds the design goal of 5 gs slightly, but overall the performance is good.

One of the difficulties with genetic methods is knowing when to stop. With gradient-based optimization methods, assuming the problem is smooth, the user can examine the norm of the gradient and if it is sufficiently small, one can assume that no further improvement is possible. With genetic methods, however, no such indication is available. In theory, the fitness can continue to improve as long as the search is run. In practice, the algorithm ususally reaches a point where the fitness does not decrease after many generations, and many of the members of the population look very similar. Koza[7] recommends starting with large initial populations and not running for many generations. It is also suggested that several runs be made with different starting populations.

Another disadvantage of this technique is the amount of time necessary to generate results. Each of the three runs lasted well over 100 hours on a Pentium II 233 MHz processor. The simulations themselves required the most amount of time, but with modifications they could be made to run faster. Implementing the problem in lower level programming languages would undoubtedly increase the speed considerably, at the loss of the ease of implementation.

V. Summary

An example was presented which demonstrates the ability of genetic methods to develop nonlinear control laws. In the future it would be desirable to include the speed brake to provide more rapid deceleration and thus greater maneuverability. It would also be desirable to include terrain (i.e. ground slope) in the problem as well. Another area of research would be to expand the envelope for which the PARS is capable of recovering, such as spins and other adverse conditions, assuming that the aircraft model could be enhanced to simulate these conditions.

VI. References

1 Combs, S.R., Gousman, K.G., and Tauke, G.J., “Pilot Activated Recovery System on the F-117A,” AIAA paper 92-1126, 1992.

2 Enns, D., et al., “Application of Multivariable Control Theory to Aircraft Control Laws; Interim Report: Definition of Control Problems and Requirements,” Air Force Report WL-TR-96-3071, January, 1996.

3 Fitzgerald, T.R., and Brunner, M.T., "Use of a High-Fidelity Simulation in the Development of an F/A-18 Active Ground Collision Avoidance System," AIAA paper 92-4140, 1992.

4 Holland, J.H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975.

5 Darwin, C.A., *On the Origin of Species by Means of Natural Selection*, John Murray, London, 1859.

6 DeJong, K.A., “On Using Genetic Algorithms to Search Program Spaces,” *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, J. J. Grefenstette, editor, Erlbaum Press, 1987.

7 Koza, J.R., *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, 1992.

8 Menon, P.K., Yousefpor, M., Lam, T., and Steinberg, M.L., “Nonlinear Flight Control System Synthesis Using Genetic Programming,” *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Baltimore, MD, August 7-10, 1995, pp. 461-470.

9 Schmitendorf, W. E., Shaw, O., Benson, R., and Forrest, S., "Using Genetic Algorithms for Controller Design: Simultaneous Stabilization and Eigenvalue Placement in a Region", *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Hilton Head Island, SC, August 10-12, 1992, pp. 757-761.

10 Krishnakumar, K., Swaminathan, R., Garg, S., and Narayanaswamy, S., "Solving Large Parameter Optimization Problems Using Genetic Algorithms," *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Baltimore, MD, August 7-10, 1995, pp. 449-460.

11 Perhenschi, M., "A Modified Genetic Algorithm for the Design of an Autonomous Helicopter Control System," *Proceedings of the AIAA Guidance, Navigation and Control Conference*, New Orleans, LA, August 11-13, 1997, pp. 1183-1192.

12 Porter, B., and Hicks, D. L., "Genetic Design of Digital Model-Following Flight-Control Systems", *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Monterey, CA, August 9 - 11, 1993, pp. 1648-1657.

13 Marrison, C.I., and Stengel, R.F., "Design of Robust Control Systems for a Hypersonic Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 1, 1998, pp. 58-63.

14 Sweriduk, G.D., Menon, P. K., and Steinberg, M.L., "Robust Command Augmentation System Design Using Genetic Search Methods," *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Boston, MA, August 10 - 12, 1998, pp. 296-304.

15 Menon, P. K., "Nonlinear Command Augmentation System for a High-Performance Aircraft," *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Monterey, CA, August 9 - 11, 1993, pp. 720-730.

16 *Genetic Search Toolbox* User's Manual, Optimal Synthesis Inc., 1998.

Acknowledgments

This work was supported by the United States Navy, contract number N62269-5361-1291. The authors would like to acknowledge the contribution of the Boeing Company, St. Louis, for validating the model and conducting additional simulations of the PARS.

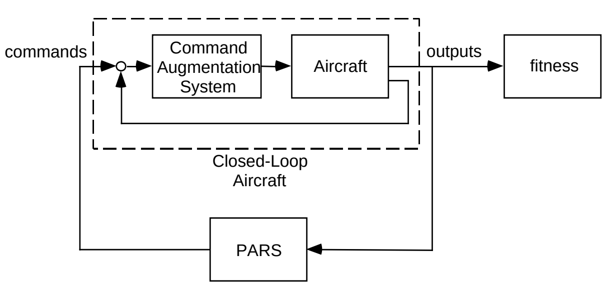

Figure 1. Closed-Loop System

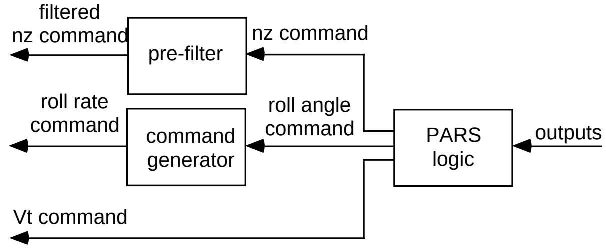

Figure 2. PARS Subsystems

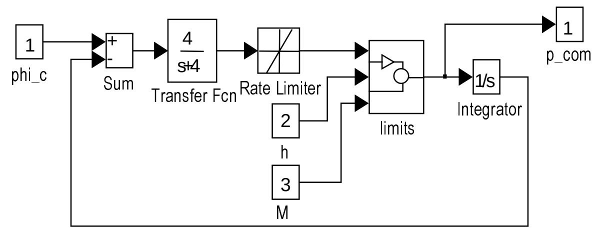

Figure 3. Roll Rate Command Generator

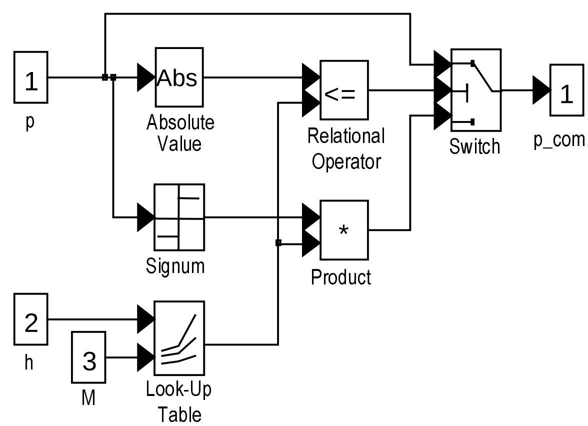

Figure 4. Limit Block

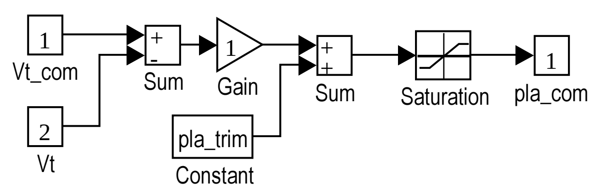

Figure 5. Velocity Controller

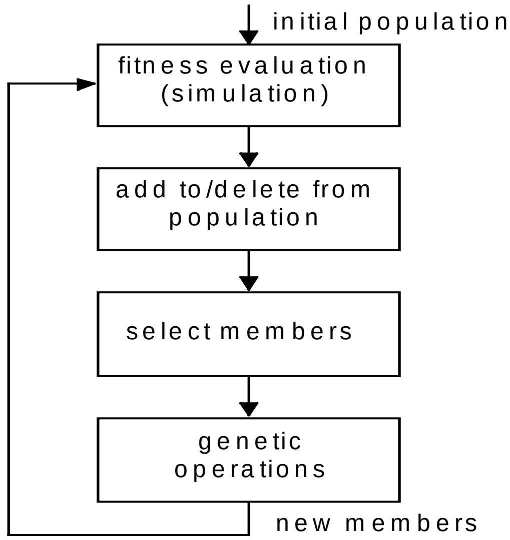

Figure 6. Genetic Search Procedure for PARS

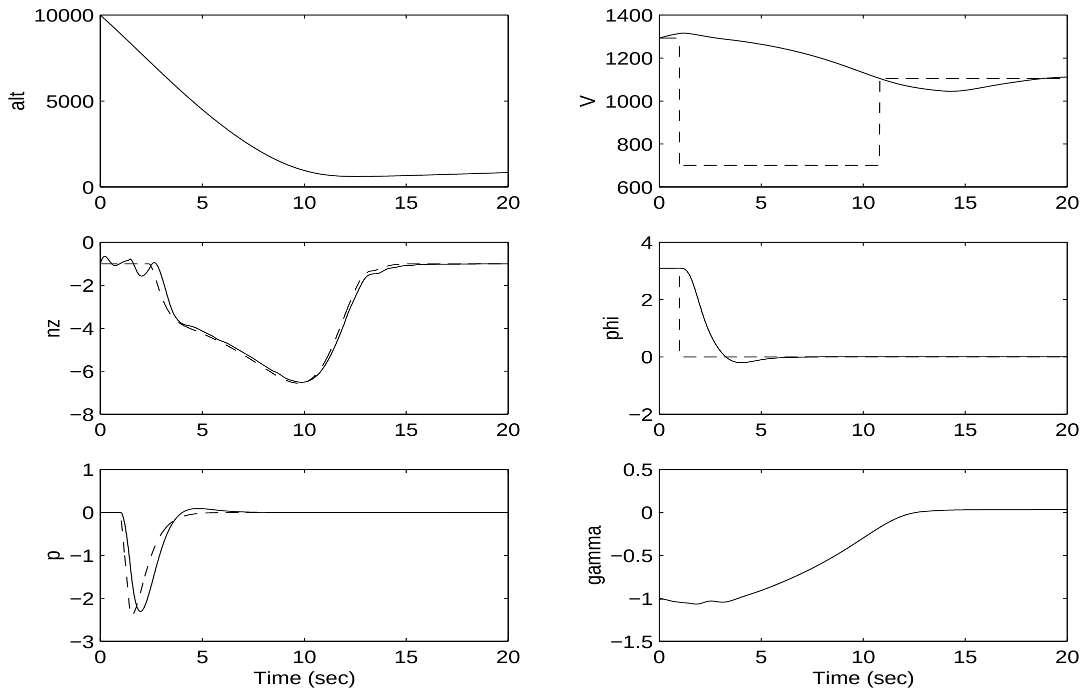

Figure 7. Initial Conditions: h = 10,000 ft, M = 1.2, γ = -57.3 deg., ϕ = 180 deg.

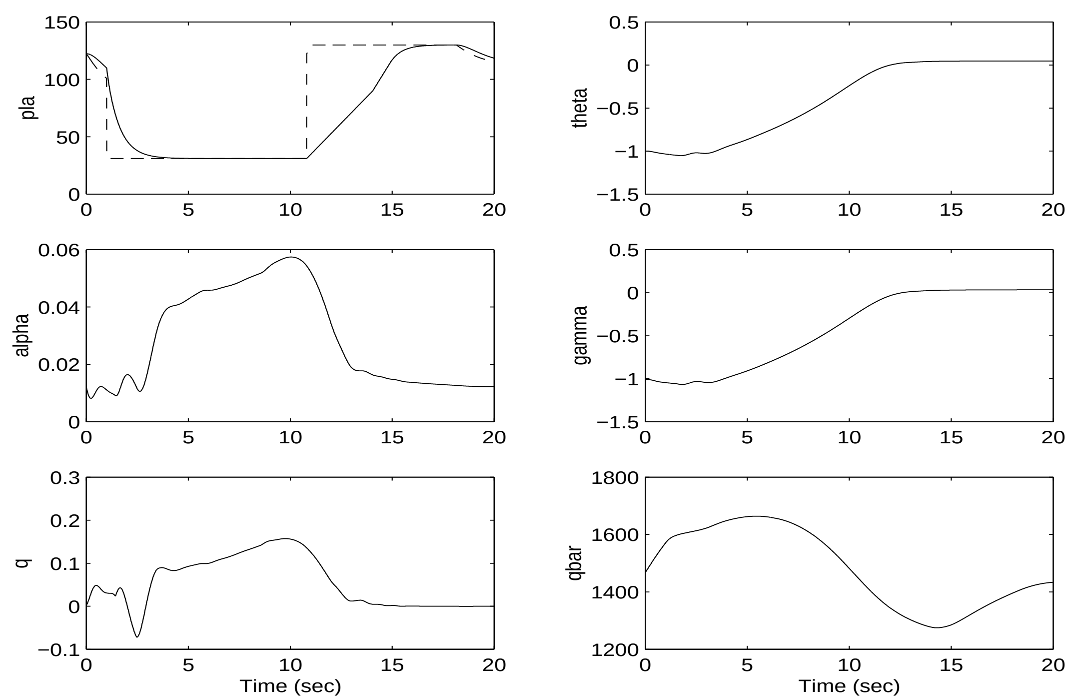

Figure 8. Initial Conditions: h = 10,000 ft, M = 1.2, γ = -57.3 deg., ϕ = 180 deg.

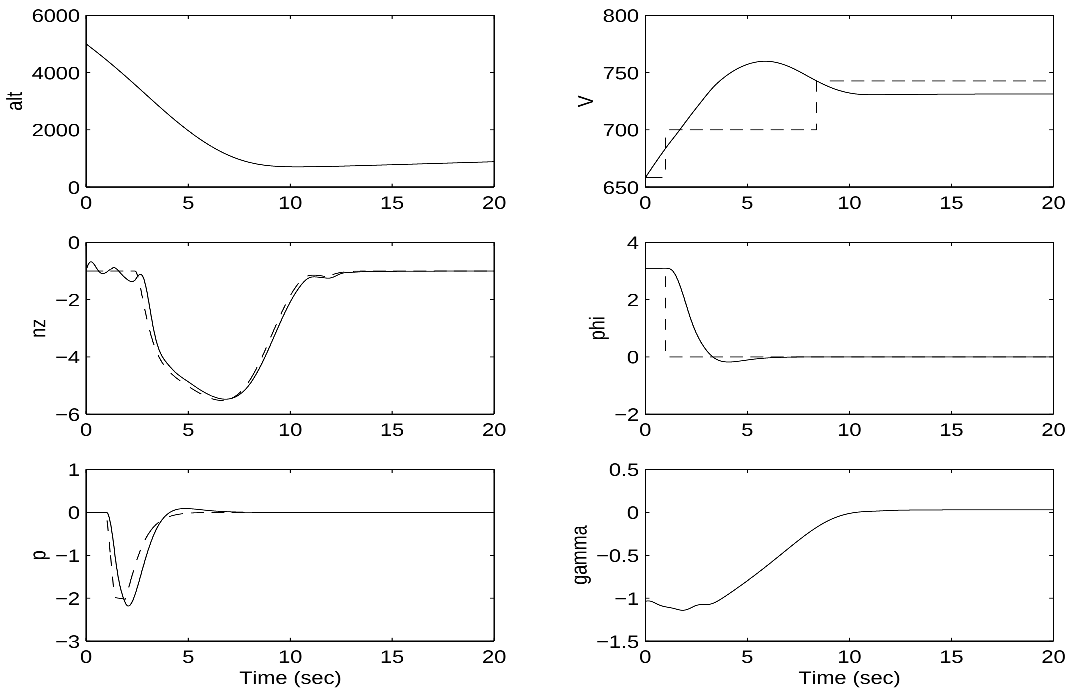

Figure 9. Initial Conditions: h = 5,000 ft, M = 0.6, γ = -57.3 deg., ϕ = 180 deg.

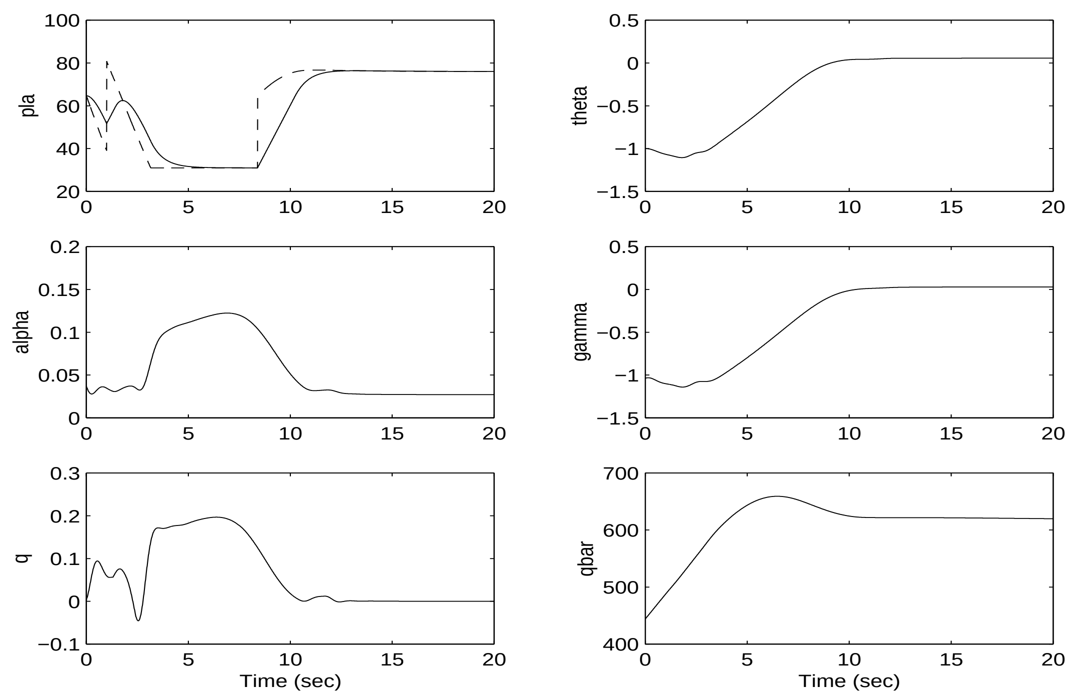

Figure 10. Initial Conditions: h = 5,000 ft, M = 0.6, γ = -57.3 deg., ϕ =180 deg.

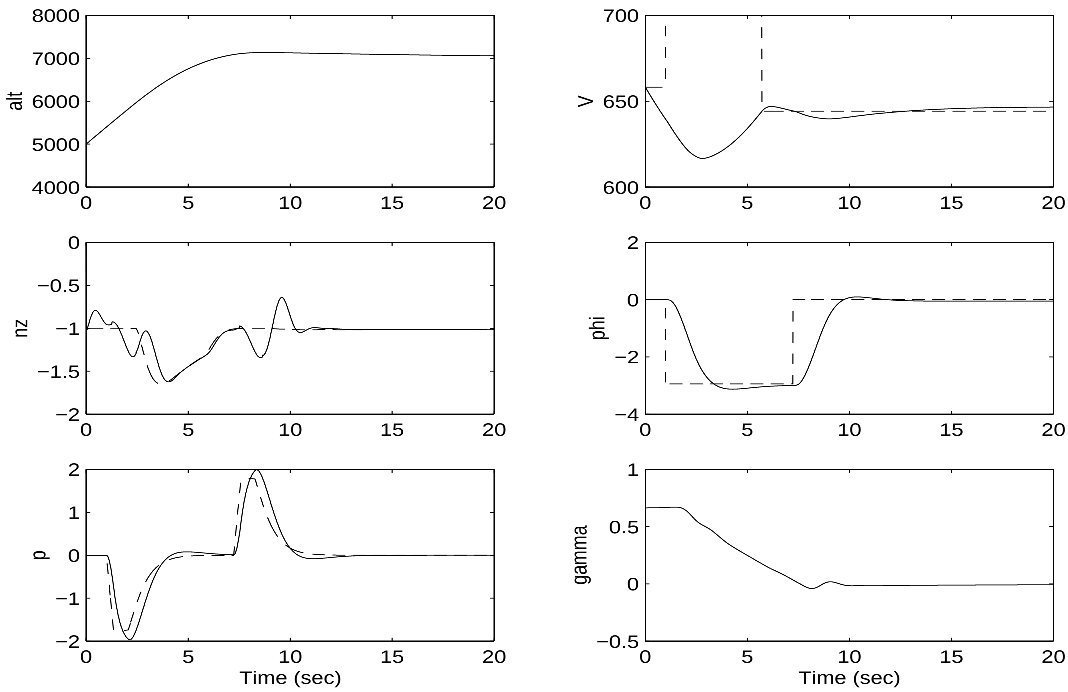

Figure 11. Initial Conditions: h = 5,000 ft, M = 0.6, γ = 40 deg., ϕ = 0 deg.

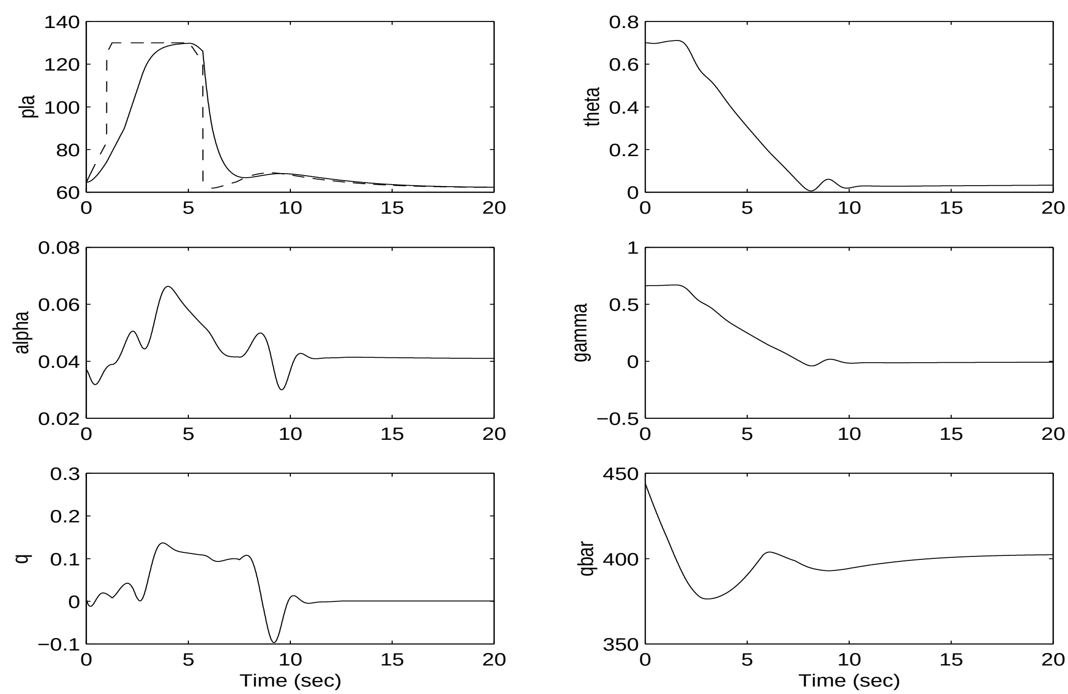

Figure 12. Initial Conditions: h = 5,000 ft, M = 0.6, γ = 40 deg., ϕ = 0 deg.