

# A New Approach to Aircraft Robust Performance Analysis

Irene M. Gregory\*

Dynamics and Controls Branch  
NASA Langley Research Center  
Hampton, VA

Jorge E. Tierno

Science and Research Center  
Honeywell  
Minneapolis, MN

## Abstract

A recently developed algorithm for nonlinear system performance analysis has been applied to an F16 aircraft to begin evaluating the suitability of the method for aerospace problems. The algorithm has a potential to be much more efficient than the current methods in performance analysis for aircraft. This paper is the initial step in evaluating this potential.

**Keywords:** aircraft performance, uncertain system, nonlinear performance.

## 1 Introduction

Real world systems are necessarily nonlinear and the range in which these systems can operate safely and within specifications is of paramount interest to both system designers and their eventual users. Typically, controllers are designed and analyzed using linear methods at different operating conditions. In the final design stages, the nonlinear system is analyzed through repetitive simulation to determine its actual performance. The currently accepted practice in industry and elsewhere is to select a number of potential operating parameters, select their operating range, and then randomly simulate the system [1]. The process is known as Monte Carlo simulation. In order to ensure that the full range of the system's behavior is covered, a very large number of simulations must be performed. In a typical scenario of an aircraft landing, 5 weight conditions, 5 center of gravity locations, and 2 flap settings are considered. The number of Monte Carlo simulations associated with this scenario, in order to ensure sufficient exploration of noise and disturbance space, is 5000. This scenario constitutes a small part of the operational envelope of an airplane, although an important one. As the number of parameters allowed to vary increases, the problem of checking performance robustness of a realistic system with this method grows exponentially in size and complexity.

Any methodology that would provide for a reduction

in the number of simulations necessary to ascertain robust performance of an airplane would be very useful in both control system development and in assisting the certification process. In recent work, we have developed a numerical tool for nonlinear robust performance analysis that has the potential to reduce the number of simulations necessary to assess aircraft performance and thus ameliorate the exponential growth of the problem [5]. Emphasis was placed on creating a computationally sound tool, requiring only information usually available on the process being analyzed. This analysis tool only requires a simulation program for the plant. The development in many respects parallels that of robustness analysis for linear systems. We would like to ascertain the suitability of the developed algorithm for solving problems encountered in aerospace applications. As the first test of the methodology, the algorithm is applied to an F16 fighter aircraft executing a maneuver to determine the worst deviation from the trajectory.

For linear time invariant (LTI) systems with complex, structured uncertainty, analysis of robust performance can be reduced to searching for the solution of a set of algebraic equations which give bounds on the achievable performance. One is thus able to find computationally efficient solutions, such as the power algorithm for the  $\mu$  lower bound, without doing an explicit parameter search involving repeated simulation. This works because the system is linear and the performance and uncertainty descriptions are chosen so as to give computationally attractive solutions, even for large problems.

Performance analysis for nonlinear systems is difficult due to the wide variety of behavior and structures which can occur. The algorithm summarized in this paper addresses the lower bound on the worst case performance. We will consider the problem of robust trajectory tracking: given a nominal trajectory for an uncertain, noisy nonlinear system, a feedback controller which stabilizes the trajectory and a description of the desired performance, find a lower bound on the worst case performance. The numerical tool we employ is a

---

\*E-mail: i.m.gregory@larc.nasa.gov

power algorithm to solve a specific two point boundary value problem that is similar in computational nature to the power algorithm for the lower bound of  $\mu$ , and so does not rely on classical descent techniques.

The rest of this paper proceeds as follows. In Section 2 a brief summary of the algorithm is given. In Section 3 the F16 airplane model is described, the trajectory tracking problem is set up, and the results of applying the algorithm are discussed. This is an initial application of the recently developed algorithm to an aircraft problem. The ability of the algorithm to handle tabular data, a relatively high number of parameters, and the class of nonlinearities present in aircraft, all typical characterizations of an aircraft problem, was unknown until this application. The last section concludes the paper with a discussion of results, issues encountered in this application, and new work.

## 2 Brief Summary of Robust Tracking Problem Method

Many nonlinear analysis problems of engineering interest can be reduced to a problem of tracking a nominal trajectory. Be it an airplane taking a turn or an idling engine going through a sudden change in load, the designer has in mind an appropriate path, to be completed in a finite predetermined time, and builds his control system accordingly. Since the real system is not exactly the one used for the design, and since it is also subject to noise, the system will not follow the intended trajectory. However, the design can still be considered successful if it remains close enough to it in an appropriate norm.

In this paper we consider a restricted version of this problem. The original presentation of the algorithm's development and a more detailed discussion can be found in [5]. Our performance measure will be the 2-norm of the error signal (i.e. the difference between the nominal and the actual trajectory). If needed, the error signal can be weighted by a multiplicative time function. Noise signals will be bounded in the 2-norm. Unmodeled dynamics will be norm bounded operators. The only information available on these operators is their induced 2-norm. We will not restrict the operators to be causal. The system equations will be allowed to depend on a set of real parameters varying in closed intervals. The initial conditions for some or all of the state variables will also be allowed to vary in given closed intervals.

To simplify the notation we will work in the following with a system having one uncertain parameter, one unmodeled norm bounded component and one noisy input. All the results presented, however, generalize

naturally.

Let  $u$  be the noise signal perturbing the system, and let  $y$  be the error signal, that is the difference between the nominal and the actual trajectories. Denote by  $v$  the output of the uncertain dynamical block, and  $\delta$  the real uncertain parameter. The equations describing the system will then be

$$\begin{aligned}\dot{x} &= f(x, u, v, \delta, t) \\ y &= g(x, u, v, \delta, t) \\ z &= h(x, u, v, \delta, t)\end{aligned}$$

with the following constraints

$$\begin{aligned}|\delta| &\leq 1 \\ \|u\|_2 &= 1 \\ \|v\|_2 &= \|z\|_2.\end{aligned}\tag{1}$$

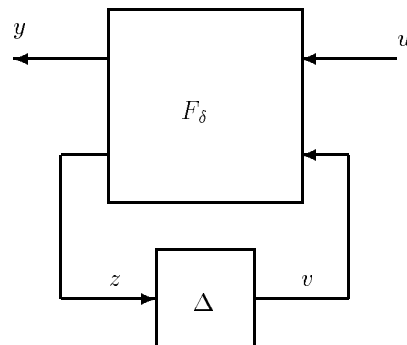


Figure 1: Schematic representation for the robust trajectory tracking problem.

Figure 1 gives a schematic representation of these equations. In order to determine how close the nominal trajectory is being tracked we need to compute

$$\max_{|\delta| \leq 1, \|u\|=1, \|v\|=\|z\|} \|y\|.\tag{2}$$

The preceding problem is a nonlinear constrained optimization problem. It is in general non-convex and some of the optimization variables live in an infinite dimensional space. An exact solution is thus out of the question: we have to settle for upper and lower bounds. Note that this is true even if the system is linear: the complexity of the problem does not come just from nonlinearities in the system but from the nonlinear nature of the optimization index and the constraints. In the following section we briefly summarize the algorithm for computing a lower bound, based on the search for locally worst case signals.

## 2.1 Necessary Conditions for Worst Case Signals

Any evaluation of the function  $\|y\|_2$ , for valid values of the parameters and signals is a lower bound on (2). So a simple way of getting lower bounds is through repeated simulation of the system for different values of the uncertain signals in the model. This is at present the state of the art of nonlinear analysis as applied in industry: good simulation models are developed and designs are tested through extensive simulation, usually selecting the uncertain signals at random. This approach is practical, since it requires information from the plant that is usually available, and often gives reasonable results. The algorithm applied in this paper improves on this approach without sacrificing in simplicity or in the generality of the information required. Instead of simulating at random points, the algorithm looks for points that are good candidates for being local maximums and this search is conducted through a “power-like” method. The robust trajectory tracking problem can be written as a standard Euler-Lagrange optimization problem. For a detailed development the reader is referred to [5]. The robust trajectory tracking problem is equivalent to optimizing the performance index

$$J = \int_{t_i}^{t_f} L dt = \frac{1}{2} \|y\|^2 \quad (3)$$

for the system verifying the differential equation

$$\begin{aligned} \dot{x} &= f(x, u, v, x_\delta) \\ \dot{x}_u &= \frac{1}{2} u^* u \\ \dot{x}_\Delta &= \frac{1}{2} (z^* z - v^* v) \\ \dot{x}_\delta &= 0, \end{aligned} \quad (4)$$

where

$$\begin{aligned} y &= g(x, u, v, x_\delta) \\ z &= h(x, u, v, x_\delta) \end{aligned}$$

with given initial conditions

$$x(t_o) = x_o, \quad x_u(t_o) = 0, \quad x_\Delta(t_o) = 0, \quad x_\delta(t_o) = \delta \quad (5)$$

and final conditions

$$x_u(t_f) = \frac{1}{2}, \quad x_\Delta(t_f) = 0. \quad (6)$$

So a set of signals  $u$ ,  $v$ , and a parameter  $\delta$  achieve the worst case value of the performance index  $J$  only if

there exists  $\Lambda = (\lambda_x, \lambda_u, \lambda_\Delta, \lambda_\delta) \in \mathbf{R}^{n \times [t_o, t_f]}$ , satisfying

$$\dot{\lambda} = - \left( \frac{\partial f}{\partial x} \right)^t \lambda - \left( \frac{\partial h}{\partial x} \right)^t z \lambda_\Delta - \left( \frac{\partial g}{\partial x} \right)^t y \quad (7)$$

$$\dot{\lambda}_u = 0 \quad (8)$$

$$\dot{\lambda}_\Delta = 0 \quad (9)$$

$$\dot{\lambda}_\delta = - \left( \frac{\partial f}{\partial \delta} \right)^t \lambda - \left( \frac{\partial h}{\partial \delta} \right)^t z \lambda_\Delta - \left( \frac{\partial g}{\partial \delta} \right)^t y \quad (10)$$

with final state conditions

$$\begin{aligned} \lambda(t_f) &= 0 \\ \lambda_\delta(t_f) &= 0 \end{aligned} \quad (11)$$

and satisfying the following alignment conditions

$$\begin{aligned} \left( \frac{\partial f}{\partial u} \right)^t \lambda + u \lambda_u + \left( \frac{\partial h}{\partial u} \right)^t z \lambda_\Delta + \left( \frac{\partial g}{\partial u} \right)^t y &= 0 \\ \left( \frac{\partial f}{\partial v} \right)^t \lambda + \left( \left( \frac{\partial h}{\partial v} \right)^t z - v \right) \lambda_\Delta + \left( \frac{\partial g}{\partial v} \right)^t y &= 0 \end{aligned} \quad (12)$$

and such that the initial state verifies:

$$\lambda_\delta(t_i) = 0, \text{ or } \begin{cases} \delta = -1 \\ \text{and} \\ \lambda_\delta(t_i) < 0 \end{cases} \quad \text{or} \quad \begin{cases} \delta = 1 \\ \text{and} \\ \lambda_\delta(t_i) > 0. \end{cases} \quad (13)$$

**Remarks:** Equations (7) and (10) describe a linear time varying dynamical system whose inputs are the (scaled) outputs of the original system. We will refer to this system as the adjoint or co-system.

Equations (12) can be interpreted as an alignment condition between the outputs of the adjoint system and the inputs to the original dynamical system. Thus, these equations describe two dynamical systems interconnected in a feedback loop.

Equation (13) states that at an optimum, either the derivative of the performance index with respect to the value of the parameter is zero, or it is negative and the parameter is at the lower end of the interval or it is positive and the parameter is at the higher end of the interval.

If we consider both the equations for the system, the co-system, and the alignment conditions together, we have a two point boundary value problem, i.e. a set of differential equations with boundary conditions at two distinct time instants.

Several methods for solving the general two point boundary value problem have been devised and thoroughly studied. (See for example [3], [2]). However,

the standard methods are based on gradient descent. In what follows we present a method to solve this particular instance of the two point boundary value problem that avoids the problems of gradient descent methods. The algorithm is a generalization of the power algorithm for the lower bound of  $\mu$ . In fact when applied to linear systems the proposed algorithm reduces to the standard power algorithm for  $\mu$  as described by Young and Doyle [6].

## 2.2 A Power Algorithm

For a trajectory that meets the necessary conditions for a critical point, the Euler Lagrange conditions can be naturally separated into (i) a dynamical system with initial conditions only; (ii) a dynamical system with final conditions only; (iii) two sets of aligning conditions between the inputs and outputs of the two systems; and (iv) conditions relating the initial conditions of both systems. It is also important to note that the adjoint system depends on the trajectory of the original one.

So, if the perturbations signals achieve the necessary conditions, the following composition of mappings yields the identity map:

- Simulate the system along the current inputs.
- Compute the co-system along the current trajectory, and simulate it backwards in time.
- From the alignment conditions in (12) compute updated values for  $u$ ,  $v$ ,  $\lambda_u$ , and  $\lambda_\Delta$ .
- Update the value of  $x_\delta$  with the following rule:

$$\begin{aligned} \chi &= x_\delta + \lambda_\delta(t_o) \\ x_\delta &= \begin{cases} -1 & \chi < -1 \\ \chi & -1 \leq \chi \leq 1 \\ 1 & \chi > 1. \end{cases} \end{aligned}$$

Denote this composition by

$$(u^1, v^1, \delta^1, \lambda_u^1, \lambda_\Delta^1) = \Phi(u^o, v^o, \delta^o, \lambda_u^o, \lambda_\Delta^o). \quad (14)$$

**Remarks:** From the first equation in (12), using the old values of  $\lambda_\Delta$  and of the state trajectory we can compute  $u\lambda_u$ . Since  $\lambda_u$  is a scalar, and we know the norm of  $u$  we can separate this product into its components.

From the second equation in (12) we can compute  $\left(\left(\frac{\partial h}{\partial v}\right)^t z - v\right)\lambda_\Delta$ . Since we know the norm of  $v$ , and the value of  $\left(\frac{\partial h}{\partial v}\right)^t z$  we can determine  $v$  and  $\lambda_\Delta$  by intersecting the line passing through the origin and

with direction  $\left(\left(\frac{\partial h}{\partial v}\right)^t z - v\right)$  with the circle centered at  $\left(\frac{\partial h}{\partial v}\right)^t z$  and with radius  $\|v\|$ .

When solving the differential equations with a numerical integrator, we will obtain values for all the signals at a finite number of time instants. The number of operations necessary to the signal updating described grows linearly with the number of time instants.

The following iterative algorithm searches for fixed points of  $\Phi$ , by evaluating it repeatedly.

1. Simulate the system with  $u = 0$ ,  $v = 0$ ,  $x_\delta = 0$ . Use the time steps generated by the simulation routine as a time axis.
2. Select random values for  $u$  along the time axis. (the signal is to be interpolated in between time-steps). Normalize  $u$  to fit the norm requirement. Set  $v^o = 0$ , and  $x_\delta^o = 0$ . Set  $\lambda_\Delta^o = 1$ .

### 3. Repeat

$$(u^{i+1}, v^{i+1}, x_\delta^{i+1}, \lambda_u^{i+1}, \lambda_\Delta^{i+1}) := \Phi(u^i, v^i, x_\delta^i, \lambda_u^i, \lambda_\Delta^i)$$

### 4. until

$$(u^{i+1}, v^{i+1}, x_\delta^{i+1}, \lambda_u^{i+1}, \lambda_\Delta^{i+1}) = (u^i, v^i, x_\delta^i, \lambda_u^i, \lambda_\Delta^i)$$

**Remarks:** If the algorithm converges, it converges to a fixed point of  $\Phi$  and thus to a set of signals that meet the necessary conditions for a critical point.

In order to prove convergence we would have to prove that  $\Phi$  is a contraction around fixed points. That has not been proved even for the simpler case when the system is linear. Since we are interested in a preponderance of experimental evidence that this algorithm does in fact work with aerospace systems, we apply it to an F16 fighter executing a climb and turn maneuver as our initial example.

## 3 Application to an F16

We want to determine whether the algorithm, summarized in the previous section, is suitable for aerospace applications. As a first step, the algorithm's ability to handle a model that includes a number of nonlinear equations and tabular data with a relatively high number of parameters that are allowed to vary, all characteristic of a typical aircraft, must be ascertained.

The aircraft used in this example application is an F16. The aerodynamic model is a reduced version of the full model obtained in wind tunnel tests at NASA Langley in 1979 [4]. It consists of tabular data with typical interpolation routines and nonlinear equations

of motion. The engine model is that of an afterburning turbofan. The airplane model utilized in this application is defined for speed range of up to Mach 0.6 and angle of attack interval between -10 and 45 degrees. The model includes four traditional controls (elevator, aileron, rudder, and throttle) and thirteen states (velocity vector, attitude angles, angular velocities, navigational position, altitude, and engine power). Furthermore, the aerodynamic coefficients are built up in a traditional way and the equations of motion are full nonlinear flat earth equations.

Several different trajectories were analyzed, but due to lack of space only one is presented as an illustrative example. The trajectory is a constant climb, constant g coordinated turn. The effective center of gravity (cg) location is set at  $0.2\bar{c}$ , the x-coordinate of the reference cg position normalized by the maximum aerodynamic chord length ( $\bar{c}$ ) of the wing, which makes the aircraft statically stable. The performance variables were chosen to be the turning radius error and the altitude error, both measured from a nominal condition characterized by absence of any exogenous disturbances as well as any uncertainty or parameter variation. The aircraft initiates the maneuver at 10,000 ft flying at 500 fps. The F16 is then trimmed to climb at 50 fps while maintaining a 4.5 g coordinated turn. This is the aircraft's nominal trajectory as illustrated in figure 2 (solid line).

During the maneuver the aircraft is subjected to atmospheric turbulence in vertical, horizontal, and lateral directions modeled by Van Karman spectra and implemented by Dryden filters [1]. The magnitude of the allowable gusts is limited to 50 fps. In addition, seven parameters in the model are allowed to vary individually on a closed interval. These parameters include variation in cg position as well as uncertainty in the aerodynamic forces and moments along each axis. For the example presented in this paper the numerical values for the variations are as follows. Cg variation is on the interval between  $0.195\bar{c}$  and  $0.205\bar{c}$ . The aerodynamic force coefficients are allowed to vary  $\pm 2.5\%$  for  $C_x$ ,  $\pm 1.5\%$  for  $C_y$ ,  $\pm 3\%$  for  $C_z$ . The aerodynamic moment coefficients vary  $\pm 5\%$  individually for rolling ( $C_l$ ), pitching ( $C_m$ ), and yawing ( $C_n$ ) moments.

The algorithm is asked to find the combination of parameters and wind gusts that produce the largest norm of the performance variable vector, i.e. turning radius and altitude error. The worst case combination produced by the algorithm gives the value of each of the parameters at the end point of the allowable interval of variation, even though the entire interval is searched. Numerically these are Cg at  $0.195\bar{c}$ ,  $C_x$  at 102.5% from nominal,  $C_y$  at 98.5%,  $C_z$  at 97%,  $C_l$  at 95%,  $C_m$  at 95%, and  $C_n$  at 105%. The resulting 2-

norm of the performance variables is 1260 ft, which is given for future comparison rather than physical meaning.

The model simulation used by the algorithm was built up in a Simulink diagram, figure 6. The behavior of the airplane under the worst case parameter variation selected by the algorithm is illustrated in figures 2-5. The solid line in all the figures represents the nominal trajectory while the dashed line represents the perturbed trajectory.

To compare with more traditional ways of evaluating nonlinear system behavior, Monte Carlo simulations were run. For each parameter the endpoints of the interval of variation were selected as allowable values. A system simulation with random turbulence subjected to the same restrictions as those imposed by the algorithm, i.e. random noise with normal distribution and limited energy input to Dryden filters, is run for each possible combination of parameter values, 128 in this case. For each of these parameter combinations 10 simulations are performed. The resulting 2-norm of each simulation is plotted in figure 7. The figure shows the 2-norm of the performance vector for each of the simulations as well as the worst case 2-norm. A total of 1280 simulations were performed.

As can be seen from figure 7, the 2-norm of the worst case parameter combination with atmospheric winds shaped by the algorithm is indeed larger than any combination of parameters with random atmospheric winds. The two vertical lines demarcate the interval that corresponds to the combination of parameters selected by the algorithm as the worst case. Thus, the algorithm gives us a combination of parameters that is particularly bad. While this combination is not unique, as is evident from the figure, it does provide us with a better lower bound on the worst case behavior of the airplane for the allowable set of parameter variations than the Monte Carlo method. In terms of computational efficiency, the worst case algorithm is at least four times faster than the Monte Carlo simulations in this particular case. It is important to note that both the algorithm and Monte Carlo simulations provide only a lower bound. It is possible that even worse performance might be achieved by some other combination of parameters and atmospheric turbulence. There are currently no methods that compute the global maximum for a problem of this complexity.

## 4 Conclusions and Future Work

An application of a recently developed algorithm for robust performance assessment of nonlinear systems

to an F16 aircraft is presented in this paper. The algorithm successfully handles a 13 state nonlinear aero/propulsion model based on wind tunnel data in tabular form that is subjected to atmospheric turbulence and has a relatively high number of uncertain parameters. These results confirm the applicability of the nonlinear robust performance analysis method [5] to aircraft performance problems.

In order to better match the needs of the aircraft designers and of the certification process, several issues must still be addressed. The first issue of interest is comparison between bounded energy and stochastic noise signals. The current practice in industry is to rely on stochastic performance measurements. Furthermore, for the certification the FAA requires stochastic performance measures. Future research will focus on incorporating the stochastics into the algorithm.

Certification guidelines also establish probability distributions for uncertain parameters. We believe this can be incorporated into this method by subdividing the distribution curve into intervals and evaluating the worst case performance of the system while the parameter falls into that interval is evaluated. We are currently studying this point.

Although this algorithm cannot completely replace the Monte Carlo simulations necessary for the certification process, it does enhance the simulation results. It answers the question of how bad can performance of a system really get and under what circumstances would that behavior occur. This answer is extremely valuable during the development of control laws since it can be done cheaply in comparison to a large number of Monte Carlo simulations and in parallel with linear robustness analysis.

## Acknowledgements

The authors would like to thank Dr. Eugene Morelli of NASA Langley for his help in obtaining the F16 model and Dr. Richard Murray of Caltech for helpful comments, suggestions, and lots of encouragement. This work was partially supported by NASA, NSF and AFOSR.

## References

- [1] Federal Aviation Administration. Advisory circular: Automatic landing systems, January 1971. AC No 20-57.
- [2] U.M Ascher, R.M. Mattheij, and R.D. Russel. *Numerical Solution of Boundary Value Problems for*

*Ordinary Differential Equations*. Prentice Hall, 1988.

- [3] H.B. Keller. *Numerical methods for two point boundary-value problems*. Blaisdell, 1968.
- [4] B. L. Stevens and F. L. Lewis. *Aircraft Control and Simulation*. John Wiley & Sons, Inc., 1992.
- [5] J. Tierno, R. Murray, and J.C. Doyle. An efficient algorithm for performance analysis of nonlinear control systems. In *Proceedings of the 1995 American Control Conference*, 1995.
- [6] P. M. Young and J. C. Doyle. Computation of  $\mu$  with real and complex uncertainties. In *Proceedings of the 29<sup>th</sup> Conference on Decision and Control*, pages 1230–1235. IEEE, 1990.

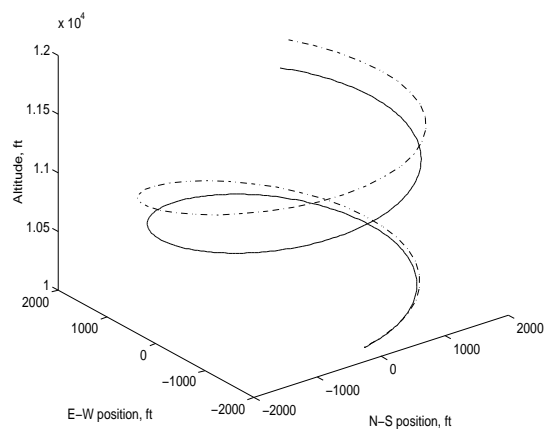


Figure 2: Spatial view of the trajectory.

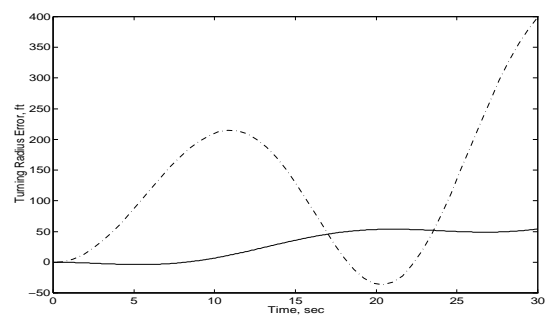


Figure 4: Turning radius error.

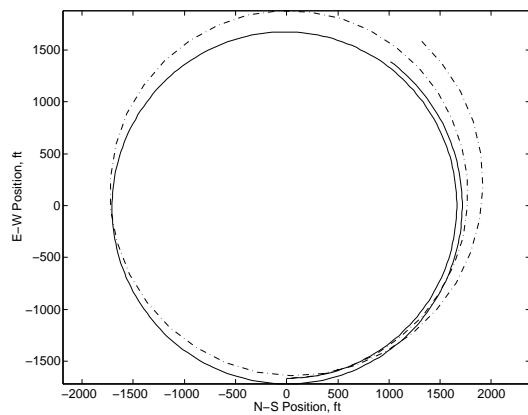


Figure 3: Ground track of the trajectory.

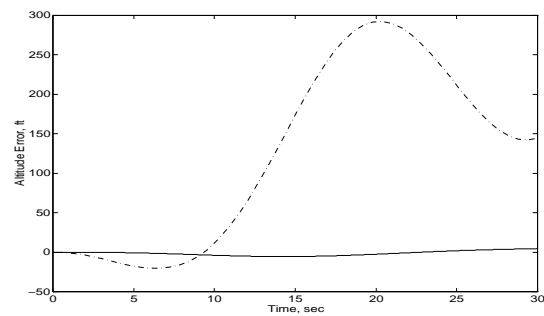


Figure 5: Altitude error.

