Generalized Technique for Inverse Simulation Applied to Aircraft Maneuvers

R. A. Hess,* C. Gao,† and S. H. Wang‡
University of California, Davis, Davis, California 95616

Inverse simulation techniques are computational methods that determine the control inputs to a dynamic system that produce desired system outputs. Such techniques can be powerful tools for the analysis of problems associated with maneuvering flight. An algorithm is developed that serves as an efficient inverse simulation tool for analyzing maneuvering flight. As opposed to current inverse simulation methods, which require numerical time differentiation in their implementation, the generalized technique offered is essentially an integration algorithm. Examples of inverse solutions for a large-amplitude aircraft maneuver and a nap-of-the-Earth helicopter maneuver are presented.

Introduction

THE inverse problem of general aircraft motion is receiving increased attention in the literature.[1-4] As addressed herein, this problem involves determining control inputs that will force a vehicle to complete some prescribed large amplitude maneuver. Although inverse methods have been applied to control system design, this is not the thrust of the research to be described. The inverse problem is, of course, not new, with the earliest aircraft-related applications dating back some 50 years.[5] Solutions to inverse problems tend to be computationally intensive, and the recent resurgence of interest in the technique both as an analysis tool for aircraft maneuvering and as a design tool for aircraft flight control systems,[6] stems from the availability of small, powerful computers and their associated software.

Computer simulation programs that solve vehicle equations of motion in forward as opposed to inverse fashion invoke the process of numerical time integration, which is accurate and stable. However, most of the inverse techniques that have been applied to flight control problems invoke, at some point in their solution, the process of numerical time differentiation, which can be inaccurate and unstable. The research to be described is aimed at reformulating the inverse problem as an integration process.

Generalized Inverse Simulation Technique

Overview

Inverse simulation problems can be conveniently divided into three categories, depending upon whether the number of outputs are greater than, equal to, or less than the number of inputs. In the first category, it would appear that there is little promise of developing a general solution technique; this category will not be discussed herein. In what follows, the second category will be referred to as nominal and the third as redundant.

The input-output relationship for a nonlinear dynamic system can be represented as a transformation from a vector $u(kT)$ in some input space to a vector $y(kT)$ in an output space, i.e.,

$$y(kT) = G[u(kT)] \tag{1}$$

where G represents the mapping function and T the discretization interval, which is typically an order of magnitude larger than the integration interval used in the simulation algorithm itself. Assuming that a desired output trajectory $y_D(kT)$ has been selected, Eq. (1) can be rewritten

$$G[u(kT)] - y_D(kT) = 0 \tag{2}$$

Newton's method[7] is generally recognized as the most powerful algorithm for solving systems like those of Eq. (2). As an iterative procedure, the solution to Eq. (2) can be approached by defining an error vector F_E, as

$$F_E[u_n(kT)] = G[u_n(kT) - Y_D(kT)] \tag{3}$$

and finding successively better estimates of $u(kT)$ that allow $F_E \to 0$, as n, the iteration index, increases. For the system of Eq. (2), Newton's method becomes

$$u_{n+1}(kT) = u_n(kT) - (J\{G[u_n(kT)]\})^{-1} \cdot F_E[u_n(kT)] \tag{4}$$

where n is the iteration index, $F_E[\,]$ defines the error vector between the actual and the desired output at time kT, and $J[\,]$ is the Jacobian matrix defined as

$$J\{G[u_n(kT)]\} = Y_U \tag{5}$$

where ny is the number of outputs and

$$Y_U|_{ij} = \partial y_i(kT)/\partial u_j(kT) \qquad i,j = 1{:}ny \tag{6}$$

The partial derivatives in Eq. (6) cannot, in general, be determined analytically. In the algorithm used here, the partial derivatives are approximated as

$$\partial y_i(kT)/\partial u_j(kT) \approx [y_i(u_j + \Delta u_j)|_{(k+1)T} - y_i(u_j)|_{kT}]/\Delta u_j \tag{7}$$

where Δu_j represents a perturbation in u_j and can be created as some fixed percentage of u_j. Although Eq. (7) is an obvious example of numerical differentiation, the fact that it is a differentiation with respect to control inputs rather than time means that the derivative can be determined from simulation with an accuracy that does not compromise the solution of the inverse simulation problem.

Received April 20, 1990; revision received Sept. 9, 1990; accepted for publication Oct. 12, 1990. Copyright © 1990 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

*Professor, Department of Mechanical, Aeronautical, and Materials Engineering. Associate Fellow AIAA.

†Graduate Student, Department of Mechanical, Aeronautical, and Materials Engineering.

‡Professor, Department of Electrical Engineering and Computer Science.

The entire inverse simulation process begins with the definition of a desired output vector y_D over a specified time interval. The vector thus defined must, of course, lie in a reachable domain; i.e., it must be able to be generated by physically realizable control inputs. Typically, the trajectory in question begins from a trimmed flight condition. It is not necessary to know the values of the control inputs in the trim condition, as the inverse simulation algorithm, itself, can determine these.

The inverse simulation process is thus seen to consist of a normal forward simulation algorithm with an integration step size $\Delta t \ll T$ in which the control inputs are changed, in step fashion, every T seconds on the basis of a solution of Eq. (4).

Redundant Case

In the nominal case, where the number of inputs equals the number of outputs, the Jacobian is square and calculating the required inverse in Eq. (4) poses no problems. Often in flight control problems, however, the redundant problem is encountered. In this case, the Jacobian is rectangular and the Moore-Penrose generalized inverse, or pseudoinverse,[7] J^+ is computed using singular value decomposition (SVD).[8]

This technique for approaching the redundant inverse simulation problem has been adopted from the robotics literature, where it has been employed to solve robot inverse kinematics problems, i.e., converting Cartesian-space position and orientation to joint-space displacement.[9] Manipulators with redundant joints (the number of joints greater than the degrees of freedom of the robot's end effector) are of interest to the robotics community since they can achieve singularity avoidance, work space obstacle avoidance, and exhibit improved dexterity.

Singular value decomposition allows the factoring of the Jacobian matrix (here considered to be $n \times m$) as:

$$J = [y][\Sigma][U^T] \tag{8}$$

where U and Y are orthogonal matrices and

$$\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r, 0, 0, \ldots, 0) \tag{9}$$

with σ_i and r being the singular values and rank of J, respectively. Finally,

$$J^+ = [U][\Sigma^+][Y^T] \tag{10}$$

Figure 1 shows the flow chart for the inverse simulation algorithm.

Comparing Differentiation and Integration Inverse Methods

What has been termed the differentiation inverse method refers to techniques such as those found in Refs. 1-4, whereas the integration inverse method refers to that proposed here. A very simple formulation allows a brief comparison of the two procedures. Consider the following linear vehicle model:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{11a}$$

$$y(t) = Cx(t) \tag{11b}$$

Let nx be the number of states, nu the number of inputs, and ny the number of outputs. It is desired to determine $u(t)$ for $t_0 \le t \le t_1$ such that $y(t) = y_D(t)$ for $t_0 \le t \le t_1$.

Differentiation Inverse Method

Consider first the simplest case in which $nx = nu = ny$. One begins by differentiating y_D and then finding $\dot{x}(t) = C^{-1}\dot{y}_D(t)$. Substituting the resulting $\dot{x}(t)$ into Eq. (10) and rearranging yields

$$u(t) = B^{-1}[C^{-1}\dot{y}_D(t) - Ax(t)] \tag{12}$$

The solution of Eq. (12) is undertaken at each time step. Now, consider the much more common situation where $nx > nu$, and $nx > ny$. For example, in a typical flight mechanics problem, $nx = 12$, $nu = 4$, and $ny \le 4$. Obvious complications arise as the differentiated output equation can no longer provide the derivatives of all of the state variables needed in solving Eq. (11a) for $u(t)$.

The necessity of calculating the derivatives of these intermediate state variables in iterative fashion at each time step using numerical differentiation is the primary disadvantage of the differentiation approach both in terms of accuracy and computational effort. At each time step, the iterative procedure is initiated by making guesses of vehicle attitude angles. As reported in Ref. 4, "The inverse solution was found to be very sensitive to initial value. This sensitivity is directly related to the severity of the maneuver." Finally, the algorithms often take on a somewhat ad hoc nature and are tailored to the particular vehicle at hand, e.g., fixed vs rotary wing.

Integration Inverse Method

The integration inverse method is summarized by Eqs. (3-7). These equations and the associated inverse simulation algorithm outlined in Fig. 1 provide a generalized approach to inverse simulation. No problems are encountered with intermediate states and, as pointed out in the preceding, only numerical differentiation with respect to the input vector is required, as shown in Eq. (7). In addition, the method is independent of the type of vehicle being simulated and the particular values of nx, nu, and ny, provided, of course that $ny \le nu$. The simplifications inherent in the proposed integration inverse method, as summarized by Eqs. (3-7), can be appreciated more fully by comparing it with the formulations associated with the derivative method, as described in Refs. 1, 3, and 4.

As with any formulation involving Newton's method, a problem that can arise with the integration inverse method concerns multiple solutions of Eq. (2). For example, consider the case where a solution of Eq. (2) lies near a relative minimum of $F_E[u(kT)]$. It is possible that another solution may

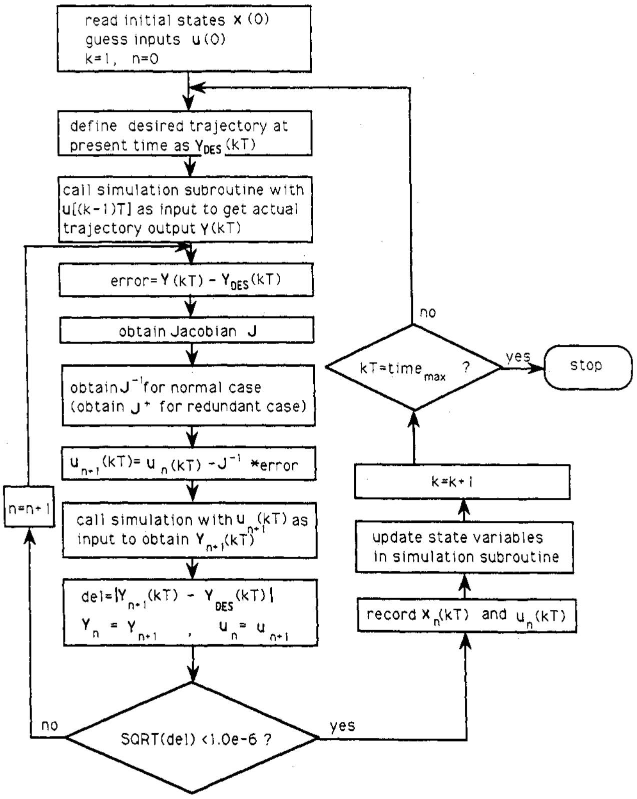
read initial states x (0)
guess inputs u(0)
k=1, n=0
define desired trajectory at present time as $Y_{DES}(kT)$
call simulation subroutine with u[(k-1)T] as input to get actual trajectory output Y(kT)
error= Y(kT) - $Y_{DES}(kT)$
obtain Jacobian J
obtain J^{-1} for normal case
(obtain J^+ for redundant case)
$u_{n+1}(kT) = u_n(kT) - J^{-1}$ *error
call simulation with $u_{n+1}(kT)$ as input to obtain $Y_{n+1}(kT)$
del=$|Y_{n+1}(kT) - Y_{DES}(kT)|$
$Y_n = Y_{n+1}$, $u_n = u_{n+1}$
SQRT(del) <1.0e-6 ?
no
yes
n=n+1
record $x_n(kT)$ and $u_n(kT)$
update state variables in simulation subroutine
k=k+1
kT=time$_{max}$?
no
yes
stop

Fig. 1 Flow chart for generalized inverse simulation algorithm.

exist in the neighborhood of the minimum and that the solution of Eq. (2) by way of Eq. (4) may alternate between these solutions for successive time steps in the simulation. This problem appeared to arise in some of the maneuvers analyzed here and is discussed further in the next section.

Input Filtering

In some of the inverse solutions to be discussed, the control inputs obtained from the simulation exhibited low-amplitude, high-frequency oscillations superimposed on the low-frequency waveform. It is hypothesized that this phenomenon may be caused by the multiple solution problem just discussed. These oscillations were filtered by the vehicle dynamics and had minimal impact on the solution quality. They were removed from the solutions to be discussed by use of a fifth-order, low-pass digital filter in the simulation process. The filter had a cutoff frequency of 10 rad/s. The control inputs were passed through the filter twice (forward and backward) to avoid time shifting in the output data.[10]

Examples

Illustrations of the inverse simulation method will employ a pair of aircraft models and flight regimes. The first will involve a fighter aircraft engaged in a pair of similar large-disturbance maneuvers at high subsonic speeds, and the second will involve a small helicopter engaged in a maneuver charac-

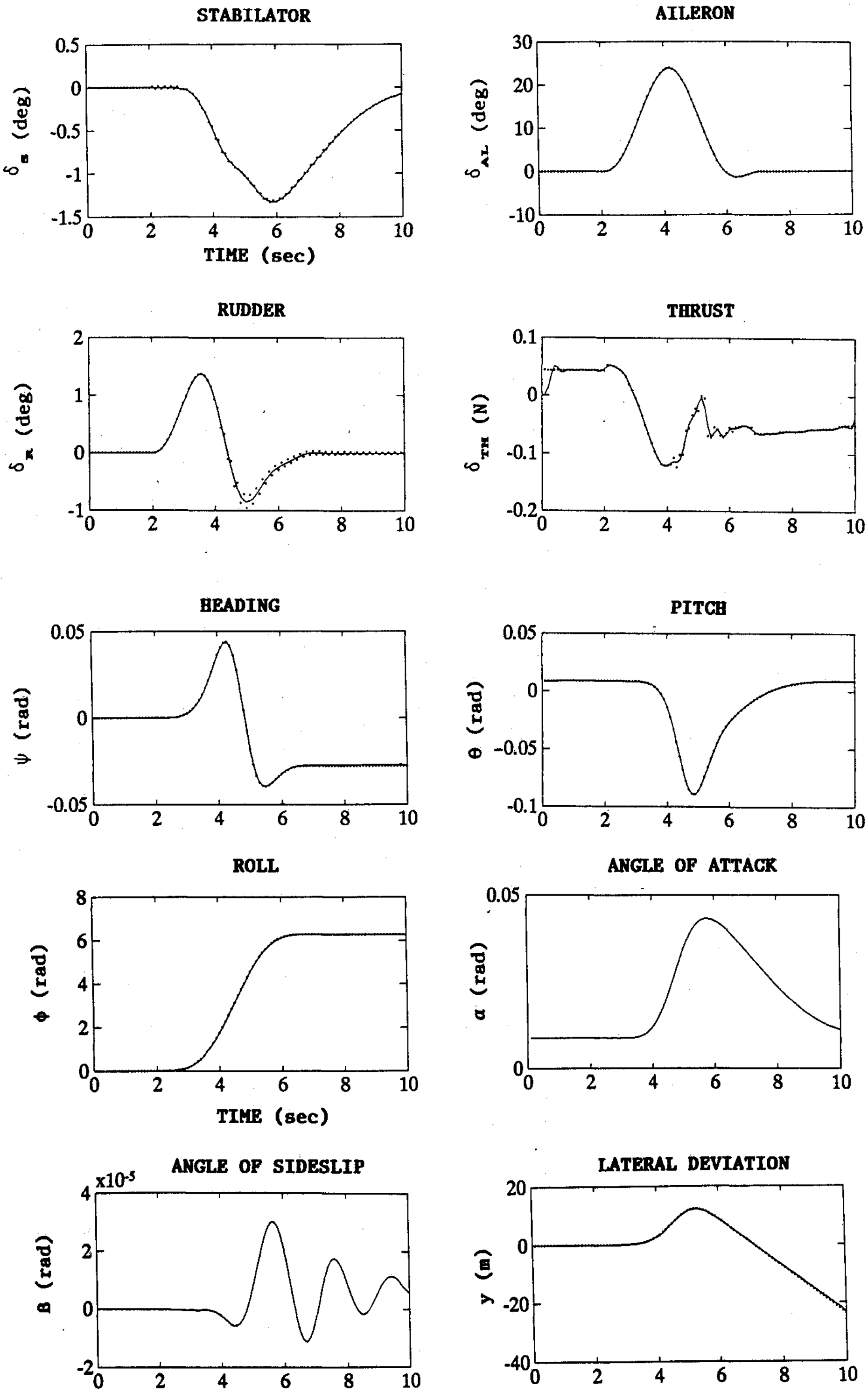

Fig. 2 F-4C aileron roll maneuver: x-axis velocity, sideslip, and roll attitude constrained.

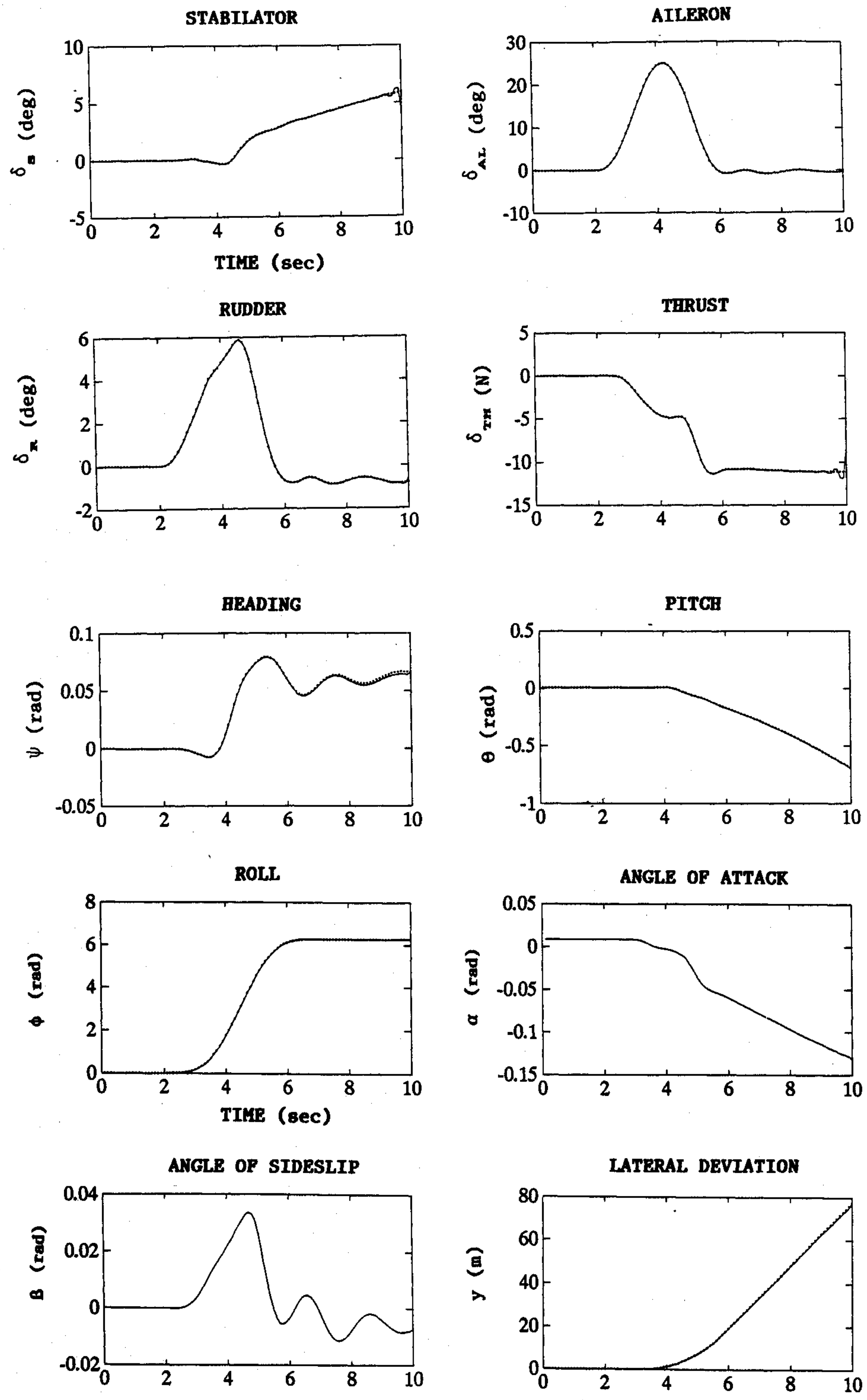

Fig. 3 F-4C aileron roll maneuver: x-axis velocity and roll attitude constrained.

teristic of low-speed, nap-of-the-Earth flight and will be performed with and without a high-bandwidth stability augmentation system (SAS). The aircraft maneuvers will require only redundant solutions, whereas the helicopter maneuver will require a nominal solution technique. The vehicle models in both cases utilized nonlinear dynamic models albeit with linear aerodynamics. The basic equations of motion are shown in the Appendix. The nonlinear kinematic equations, Euler angle definitions, etc., can be found in any standard text on the subject.[11] In all of the simulations to be discussed, a discretization interval $T = 0.1$ s was used. An integration step size $\Delta t = 0.01$ s was maintained except for the helicopter with SAS, where $\Delta t = 0.001$ s was used. All of the solutions were obtained on an IBM PC-AT using FORTRAN, LINPACK, and MATLAB software.

In the time histories associated with the examples to be discussed, dotted curves represent the results of the inverse simulation. For the control time histories, solid curves represent the filtered control inputs, whereas for the output time histories, solid curves represent the results of a forward simulation conducted with the filtered time histories as inputs. The forward simulation results are of obvious importance for they demonstrate both the validity of the inverse solution and the errors introduced by the input filtering just described.

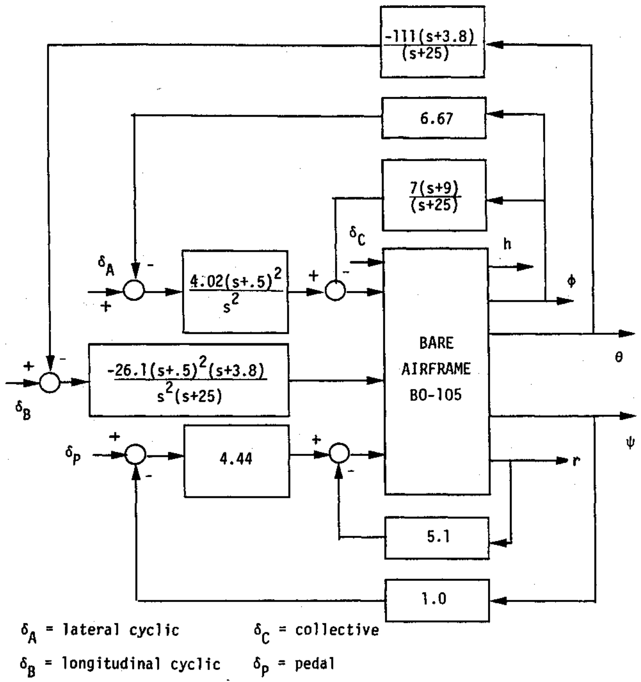

Fig. 4 Stability augmentation system for BO-105 model: attitude command/attitude hold in pitch, roll, and heading.

Large Amplitude Aircraft Maneuvers

The aircraft model used here is the F-4C fighter aircraft from Ref. 12. The longitudinal and lateral vehicle dynamics are not coupled aerodynamically. The flight condition was one of level flight at 4572 m at a Mach number of 0.9. In this flight condition, the phugoid mode is represented by a pair of real roots, one of which is unstable. The two maneuvers to be considered all involve aileron rolls; however, they differ in the constraint variables. The four control inputs are stabilator angle δ_S (rad), rudder angle δ_R (rad), aileron angle δ_{AL} (rad), and thrust δ_{TH} (N).

Aileron Roll with x-Axis Velocity, Sideslip, and Roll Attitude Constrained

Here, $u = U_0 = 290$ m/s, where u is the x body-axis component of the aircraft velocity, sideslip $\beta(t) = 0$, and roll attitude, $\phi(t)$, given by

$$\phi(t) = 0 \text{ rad} \qquad \text{for} \qquad 0 \le t < 2 \text{ s} \tag{13a}$$

$$\phi(t-2) = (2\pi/16)\{\cos 3\pi[(t-2)/5] - 9\cos\pi[(t-2)/5] + 8\} \text{ rad} \qquad \text{for} \qquad 2 \le t \le 7 \text{ s} \tag{13b}$$

$$\phi(t-7) = 2\pi \qquad \text{for} \qquad t > 7 \text{ s} \tag{13c}$$

The basic waveform for the roll time history is taken from Ref. 1 and represents a 360-deg aileron roll of 5-s duration.

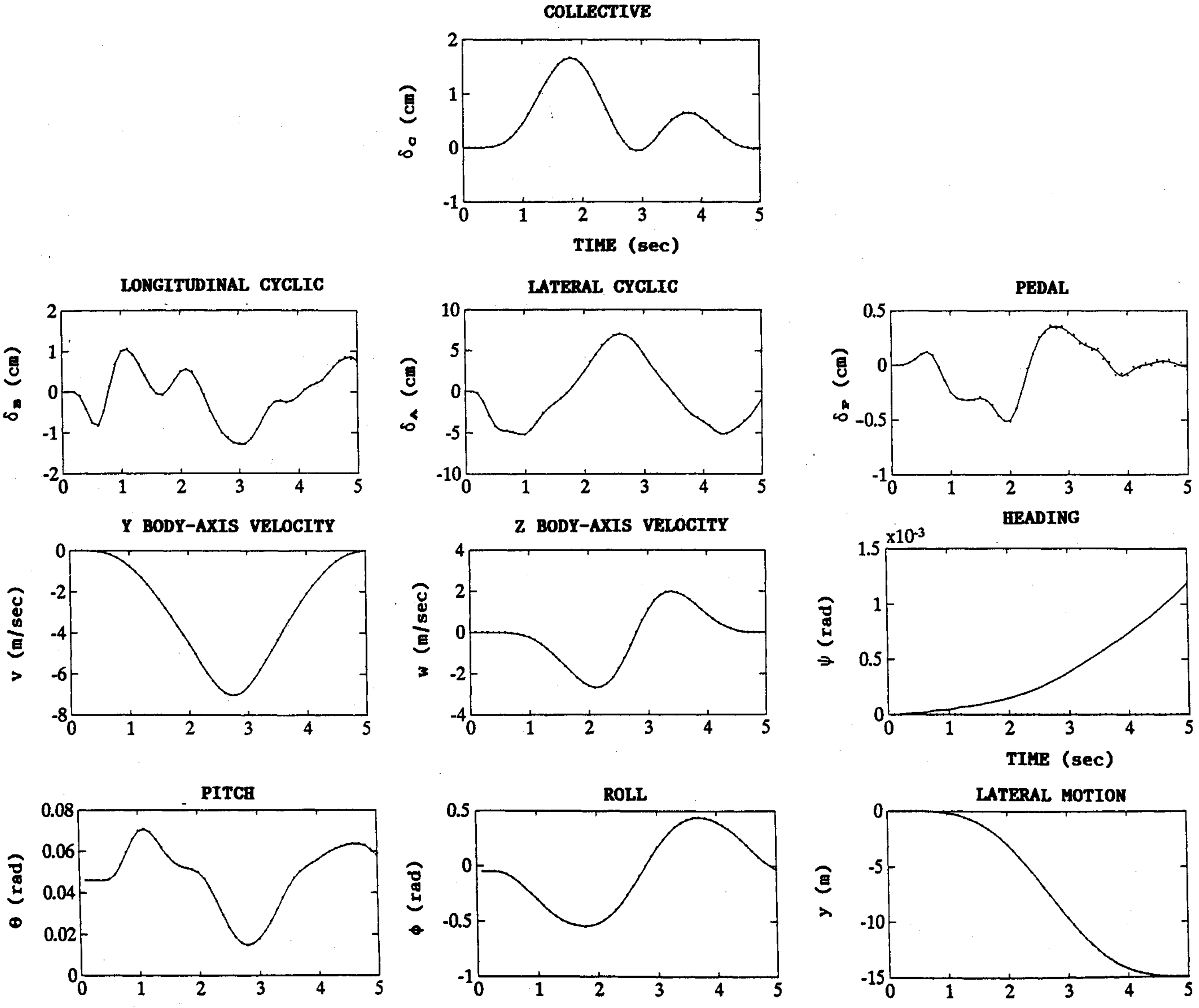

Fig. 5 BO-105 left side-step maneuver, without SAS.

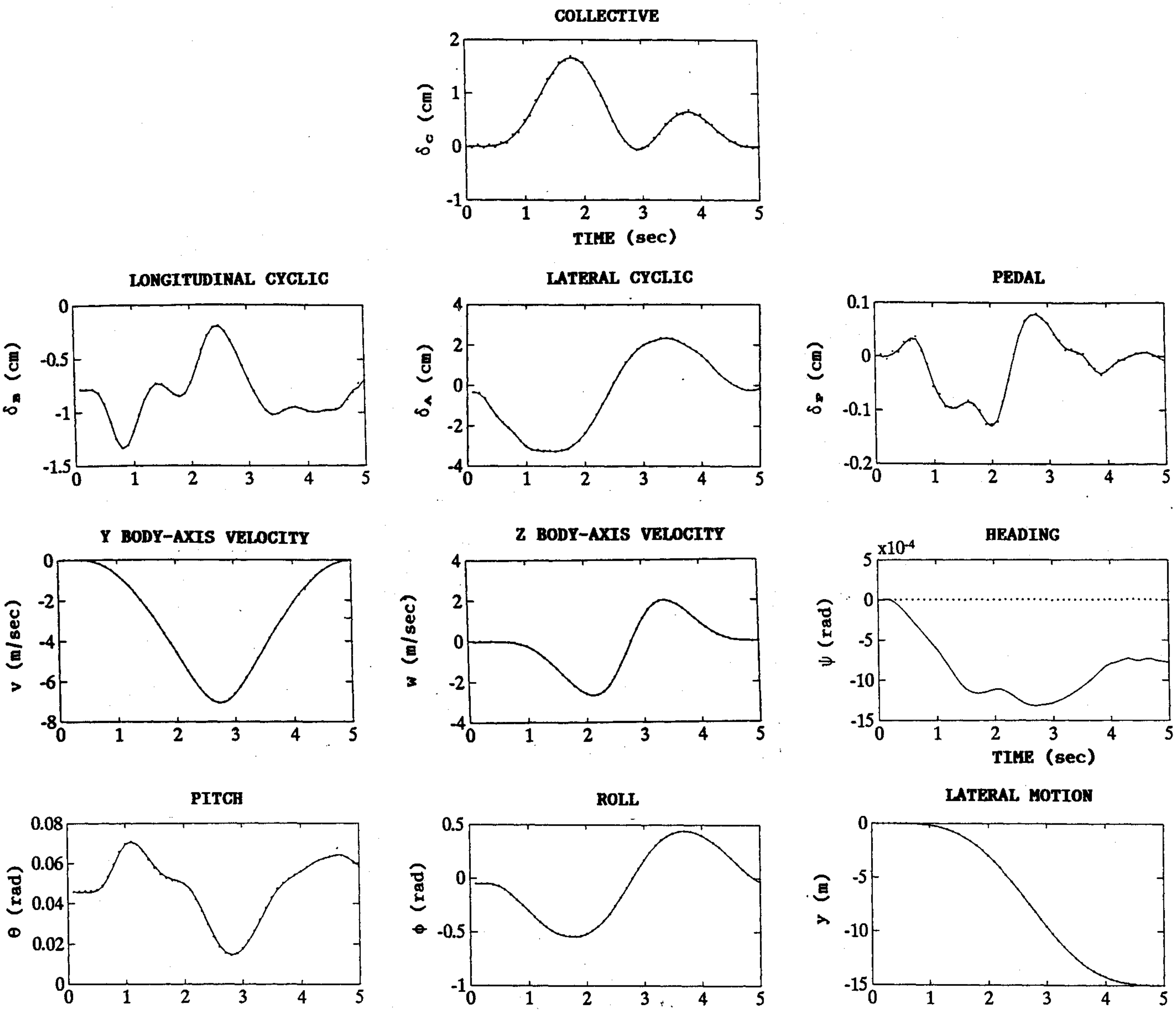

Fig. 6 BO-105 left side-step maneuver, with SAS.

This represents a redundant maneuver because only three outputs are constrained and four control inputs are used. Figure 2 shows the control inputs, attitude angles, angles of attack, and sideslip and lateral path deviation resulting from the analysis. No changes in the x component of the aircraft velocity occurred, and this graph is not shown. Also note that the thrust changes are minimal. At $t = 7$ s, the aircraft had lost about 50-m altitude.

Aileron Roll with x-Axis Velocity and Roll Attitude Constrained

Here, two outputs are constrained, with four control inputs. Figure 3 shows the same variables as selected for Fig. 2. Note the control inputs that this maneuver induced as compared with those of Fig. 2. Changes in the x-axis velocity were less than 0.01%. At $t = 7$ s, the aircraft had lost about 70-m altitude.

Helicopter Nap-of-the-Earth Maneuvers

The helicopter model used here is the BO-105 helicopter.[13] As opposed to the F-4C model, the longitudinal and lateral modes of the BO-105 model are coupled aerodynamically. The nominal flight condition is hover at sea level. The maneuver to be discussed is a left side-step maneuver from the hover condition. The side step will be analyzed with and without a SAS. It should be noted that the helicopter is unstable without a SAS in this flight condition. The four control inputs are longitudinal cyclic δ_B (cm), lateral cyclic δ_A (cm), collective δ_C (cm), and pedal δ_P (cm). As opposed to the F-4C vehicle, these inputs are control manipulator deflections in the cockpit. The SAS dynamics are included as part of the effective vehicle in the inverse simulation, i.e., the control inputs that are calculated with and without the SAS are those that would be generated in the cockpit. Figure 4 shows the SAS feedback structure and represents attitude command/attitude hold systems for the pitch, roll, and heading axes. The bandwidths of the systems are quite high, approximately 4 rad/s in pitch and roll and 1 rad/s in heading. In each of the simulations to be described, the runs were initiated at $t = -5$ s to allow the algorithm to determine trim control inputs accurately. This 5-s period is not shown on the figures.

Side-Step Maneuver with x-y-z Inertial Position and Heading Constrained, No Stability Augmentation System

The side-step maneuver is characterized by a rapid, constant altitude and heading, 15-m lateral translation, starting and terminating in a stabilized hover. Using the same general waveform as in Eq. (13), the 5-s lateral translation $y(t)$ is given by

$$\begin{aligned} y(t) &= -(15/16)[\cos 3\pi(t/5) - 9\cos\pi(t/5) + 8]\ \text{m} \quad \text{for} \quad 0 \le t \le 5\ \text{s} \\ &= -15\ \text{m} \quad \text{for} \quad t > 5\ \text{s} \end{aligned} \tag{14}$$

Figure 5 shows the control inputs, the y- and z-body axis components of velocity (v and w), attitude angles, and lateral displacement for the side step. The x-axis velocity was <0.006 m/s throughout the maneuver and the helicopter lost 0.03 m in altitude. The heading deviations shown in Fig. 5 are quite small, with the largest being 0.069 deg.

Side-Step Maneuver with x-y-z Inertial Position and Heading Constrained, with Stability Augmentation System

Figure 6 shows the corresponding variables for the left side-step maneuver with the SAS in operation. The primary effect of the SAS is seen in the simplification of the lateral cyclic input and its direct correspondence with vehicle roll attitude. This correspondence is what would be expected from a high-bandwidth attitude command SAS. The x-axis velocity, altitude, and heading deviations were comparable to those with no SAS in operation.

Conclusions

1) A generalized technique for inverse simulation formulated as an integration process appears to offer advantages over those that require time differentiation.

2) The technique can be applied to flight control problems in which the number of controls equal or exceed the number of constrained outputs.

3) Multiple-axis stability and control augmentation systems can be accommodated in a straightforward manner, i.e., as modifications of the dynamics of the effective vehicle.

4) The necessity of input filtering does not appear to detract from the utility of the technique in the analysis of aircraft maneuvers, over and above the effort required for filtering itself.

Appendix: Vehicle Equations of Motion

$$\dot{u} = rv - qw - g\sin(\Theta) + X_{\text{AERO}} + X_{\text{TRIM}} + X_{\text{CONTROL}}$$

$$\dot{v} = pw - ru - g\cos(\Theta)\sin(\phi) + Y_{\text{AERO}} + Y_{\text{TRIM}} + Y_{\text{CONTROL}}$$

$$\dot{w} = uq - vp - g\cos(\Theta)\cos(\phi) + Z_{\text{AERO}} + Z_{\text{TRIM}} + Z_{\text{CONTROL}}$$

$$\dot{p} = (r + pq)(I_{XZ}/I_{XX}) - qr(I_{ZZ} - I_{YY})/I_{XX} + L_{\text{AERO}} + L_{\text{TRIM}} + L_{\text{CONTROL}}$$

$$\dot{q} = -pr(I_{XX} - I_{ZZ})/I_{YY} + (r^2 - p^2)(I_{XZ}/I_{YY}) + M_{\text{AERO}} + M_{\text{TRIM}} + M_{\text{CONTROL}}$$

$$\dot{r} = (\dot{p} - qr)(I_{XZ}/I_{ZZ}) - pq(I_{YY} - I_{XX})/I_{ZZ} + N_{\text{AERO}} + N_{\text{TRIM}} + N_{\text{CONTROL}}$$

where u,v,w are velocity components in body axes and p,q,r are angular velocity components in body axes.

The trim forces and moments are defined as

$$X_{\text{TRIM}} = w_0q_0 - v_0r_0 + g\sin(\Theta)$$

$$Y_{\text{TRIM}} = u_0r_0 - w_0p_0 - g\cos(\Theta_0)\sin(\phi_0)$$

$$Z_{\text{TRIM}} = v_0p_0 - u_0q_0 - g\cos(\Theta_0)\cos(\phi_0)$$

$$L_{\text{TRIM}} = [p_0r_0(I_{ZZ} - I_{YY}) - p_0q_0I_{XZ}]/I_{XX}$$

$$M_{\text{TRIM}} = [p_0r_0(I_{XX} - I_{ZZ}) - r_0^2I_{XZ} + p_0^2I_{XZ}]/I_{YY}$$

$$N_{\text{TRIM}} = [p_0q_0(I_{YY} - I_{XX}) + q_0r_0I_{XZ}]/I_{ZZ}$$

where subscript 0 is the trim condition.

References

[1] Kato, O., and Suguira, I., "An Interpretation of Airplane General Motion and Control as Inverse Problem," *Journal of Guidance, Control, and Dynamics,* Vol. 9, No. 2, 1986, pp. 198–204.

[2] McKillip, R. M., Jr., and Perri, T. A., "Helicopter Flight Control System Design and Evaluation for NOE Operations Using Controller Inversion Techniques," *Proceedings of the 45th Annual Forum of the American Helicopter Society,* American Helicopter Society, Alexandria, VA, May 1989, pp. 669–680.

[3] Kato, O., "Attitude Projection Method for Analyzing Large-Amplitude Maneuvers," *Journal of Guidance, Control, and Dynamics,* Vol. 13, No. 1, 1990, pp. 22–29.

[4] Thomson, D. G., and Bradley, R., "Development and Verification of an Algorithm for Helicopter Inverse Simulation," *Vertica,* Vol. 14, No. 2, 1990, pp. 185–200.

[5] Jones, R. T., "A Simplified Application of the Method of Operators to the Calculation of Disturbed Motions of an Airplane," NACA TR 560, 1936.

[6] Meyer, G., and Cicolani, L., "A Formal Structure for Advanced Automatic Flight-Control Design—System Concepts and Flight Evaluations," *Theory and Application of Optimal Control in Aerospace Systems,* edited by I. P. Kant, AGARDograph 251, 1981.

[7] Luenberger, D. G., *Optimization by Vector Space Methods,* Wiley, New York, 1969.

[8] Goloub, G. H., and Van Loan, C. F., *Matrix Computations,* Johns Hopkins University Press, Baltimore, MD, 1983.

[9] Hsia, T. C., and Guo, Z. Y., "Joint Trajectory Generation for Redundant Robots," *Proceedings of the 1989 IEEE International Conference on Robotics and Automation,* May 1989, pp. 277–282.

[10] Anon., "PC-MATLAB User's Guide," Mathworks, Inc., South Natick, MA, 1987.

[11] Etkin, B., *Dynamics of Flight,* 2nd ed., Wiley, New York, 1982, Chap. 4.

[12] Heffley, R. K., and Jewell, W. F., "Aircraft Handling Qualities Data," NASA CR-2144, Dec. 1972, pp. 61–107.

[13] Heffley, R. K., Jewell, W. F., Lehman, J. M., and Van Winkle, R. A., "A Compilation and Analysis of Helicopter Handling Qualities Data, Vol. 1: Data Compilation," NASA CR-3144, March 1979, pp. 67–114.