JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 30, No. 1, January–February 2007

Sensitivity-Analysis Method for Inverse Simulation Application

Linghai Lu* and David J. Murray-Smith†
University of Glasgow, Glasgow, G12 8LT Scotland, United Kingdom
and
D. G. Thomson‡
University of Glasgow, Glasgow, G12 8QQ Scotland, United Kingdom

DOI: 10.2514/1.20722

An important criticism of traditional methods of inverse simulation that are based on the Newton–Raphson algorithm is that they suffer from numerical problems. In this paper these problems are discussed and a new method based on sensitivity-analysis theory is developed and evaluated. The Jacobian matrix may be calculated by solving a sensitivity equation and this has advantages over the approximation methods that are usually applied when the derivatives of output variables with respect to inputs cannot be found analytically. The methodology also overcomes problems of input-output redundancy that arise in the traditional approaches to inverse simulation. The sensitivity-analysis approach makes full use of information within the time interval over which key quantities are compared, such as the difference between calculated values and the given ideal maneuver after each integration step. Applications to nonlinear HS125 aircraft and Lynx helicopter models show that, for this sensitivity-analysis method, more stable and accurate results are obtained than from use of the traditional Newton–Raphson approach.

Nomenclature

$\boldsymbol{f}$	=	state equation
$\boldsymbol{f}_E$, $\boldsymbol{g}$	=	error function and output equation
$\boldsymbol{H}$	=	sensitivity function
h	=	height of the obstacle
$\boldsymbol{J}$	=	Jacobian matrix
l	=	total iterative number
M	=	integration number
m	=	order of state-variable vector
N	=	number of discretized intervals
n	=	iterative step
p	=	order of output vector
q	=	order of input vector
s	=	span distance
T_t	=	thrust input
t_m	=	time to complete hurdle-hop maneuver
t_{NR}, t_{SA}	=	computation time
$\boldsymbol{u}$	=	input vector
V_f, V_{fd}	=	actual and ideal resultant speed
$\boldsymbol{W}$	=	weighting matrix
$\boldsymbol{x}$	=	state-variable vector
x_e	=	displacement in the x-axis
$\boldsymbol{y}$, $\boldsymbol{y}_d$	=	output variable and ideal output
y_e	=	displacements in the y-axis
$\boldsymbol{Z}$	=	sensitivity function
z_d, z_e	=	ideal and actual displacements in the z-axis
$\boldsymbol{\alpha}$	=	time-varying parameter
Γ	=	operator mapping parameter to output
$\boldsymbol{\gamma}$, $\boldsymbol{\gamma}_0$	=	approximating parameter and initial value
Δt	=	discretized time interval
$\Delta\boldsymbol{\gamma}$	=	increment
δ_e	=	elevator angle
$\boldsymbol{\theta}$	=	constant parameter
θ_0, θ_{1c}	=	collective pitch and lateral cyclic
θ_{1s}, $\theta_{0\mathrm{tr}}$	=	longitudinal cyclic and tail rotor collective pitch
λ	=	convergence multiplier
$\boldsymbol{\mu}$	=	parameter disturbance
$\boldsymbol{\Pi}$	=	linear quadratic (LQ) performance index
Ψ	=	heading angle

Subscripts

d	=	desired value
e	=	earth fixed frame of reference

Superscript

T	=	transpose

I. Introduction

SENSITIVITY analysis (SA) is a well-known methodology which has been used extensively in reliability engineering, electrical circuit theory, and control engineering. In an editorial [1], Saltelli and Scott present 12 persuasive reasons to explain why and where SA should be considered. In spite of its simplicity, Ratto et al. [2] show that SA can help to evaluate the effectiveness of predictions and identify the most important aspects of a model. Sato et al. [3] have applied SA to determine the key parameters from hundreds of parameters for a conventional time-varying simulation model. However, no application of SA to the inverse simulation field appears to have been published to date.

Inverse simulation is a technique that generates the expected forward control inputs such that the mathematical model of a vehicle can follow a prescribed path. Generally, the techniques for inverse simulation can be divided into three categories: differentiation methods [4,5], integration methods [6,7], and optimization methods [8,9]. The first two approaches are based on the Newton–Raphson (NR) algorithm. The third set of methods adapts traditional numerical optimization algorithms for use in the inverse simulation field. Some optimization methods [7] involve an iterative approach similar to that adopted for the integration-based methods but with the NR algorithm replaced by other numerical procedures.

For the purposes of comparison this paper focuses on the integration technique based on the NR method because of its wide adoption. There are some known deficiencies in this algorithm. For

Received 25 October 2005; revision received 31 March 2006; accepted for publication 31 March 2006.

*Research Student, Department of Electronics & Electrical Engineering; linghai@elec.gla.ac.uk.
†Professor Emeritus, Department of Electronics & Electrical Engineering; d.murray-smith@elec.gla.ac.uk.
‡Senior Lecturer and Head, Department of Aerospace Engineering; d.thomson@aero.gla.ac.uk.

example, in aircraft applications, because the Jacobian matrix cannot generally be determined analytically, approximation methods are used to determine partial derivatives of output variables with respect to input variables. This will inevitably reduce the accuracy and Hess et al. [6,10] suggest that better techniques should be found to calculate the Jacobian matrix.

This integration-based method [6] may give rise to a form of high-frequency oscillations. It is a phenomenon involving severe oscillations superimposed on the desired variables. It may possibly be due to a number of factors, such as step tolerance limits, initial conditions, and choice of sampling rate, etc. [11–13]. Although oscillations originating from inappropriate values of sampling rate may be eliminated by reducing the basic time step (Δt) to a small enough value, Lin [14] suggested that this could excite uncontrolled state variables, providing a further source of high-frequency oscillations. In addition, the redundancy problem associated with the number of inputs being greater than the number of outputs is a further difficulty due to the nonsquare nature of the Jacobian matrix. Hess et al. [6] suggest that this issue could be avoided by means of the pseudoinverse. However, this could not always guarantee the convergence of the solutions [15]. Avanzini et al. overcame this problem in a helicopter application by reducing the model order through a two-timescale method and adding constraint conditions [16].

This paper describes the first application of SA theory to the problems of inverse simulation for dealing with the aforementioned issues and describes a new technique that has been termed "inverse sensitivity." Many ideas from sensitivity-analysis theory are incorporated into this method and this provides opportunities for a more thorough investigation of system properties through the inverse simulation process. The paper will begin by presenting the traditional inverse simulation approach based on the NR method. Then the mathematical development of the new SA method and the theory on which it is based are presented, including discussion of numerical stability and convergence properties. Comparisons with a traditional inverse simulation method are presented using results from an application of inverse simulation to a relatively simple fixed-wing aircraft model and a six-degree-of-freedom nonlinear model of a Lynx helicopter

II. Review of the Traditional Algorithm

This section provides a summary of the method of Hess et al. [6]. A nonlinear system may be described by equations of the form:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{1a}$$

$$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) \tag{1b}$$

where $\boldsymbol{f} \in \mathbb{R}^m$ is the set of nonlinear ordinary differential equations describing the original system, $\boldsymbol{g} \in \mathbb{R}^p$ is the set of algebraic equations that construct the expected outputs, and $\boldsymbol{u} \in \mathbb{R}^q$ is the input vector. $\boldsymbol{x} \in \mathbb{R}^m$ is the state-variable vector and $\boldsymbol{y} \in \mathbb{R}^p$ is the vector of output variables.

First the input–output relationship of the preceding system can be defined as follows:

$$\boldsymbol{x}(t_{k+1}) = \int_{t_k}^{t_{k+1}} \dot{\boldsymbol{x}}(t)\,\mathrm{d}t + \boldsymbol{x}(t_k) \qquad k = 1, 2, 3, \ldots, N-1 \tag{2a}$$

$$\boldsymbol{y}(t_{k+1}) = \boldsymbol{g}[\boldsymbol{x}(t_{k+1}), \boldsymbol{u}(\mathrm{t}_k)] \tag{2b}$$

where N is the total number of discretized intervals and t_k is the kth discretization point in the time period. Now the term on the left-hand side of Eq. (2b) is replaced by ideal output values $\boldsymbol{y_d}(t_{k+1})$. Thus, Eq. (2b) can be rewritten as

$$\boldsymbol{g}[\boldsymbol{x}(t_{k+1}), \boldsymbol{u}(t_k)] - \boldsymbol{y_d}(t_{k+1}) = 0 \tag{3}$$

In the traditional algorithm, the NR-based method is used to find $\boldsymbol{u}(t_k)$ by the following iterative relationship:

$$\boldsymbol{u}^{(n+1)}(t_k) = \boldsymbol{u}^{(n)}(t_k) - (\boldsymbol{J}\{\boldsymbol{g}[\boldsymbol{x}^{(n)}(t_{k+1}), \boldsymbol{u}^{(n)}(t_k)]\})^{-1}\boldsymbol{f}_E[\boldsymbol{x}^{(n)}(t_{k+1}), \boldsymbol{u}^{(n)}(t_k)] \tag{4}$$

where $\boldsymbol{f}_E[\boldsymbol{x}^{(n)}(t_{k+1}), \boldsymbol{u}^{(n)}(t_k)] = \boldsymbol{g}[\boldsymbol{x}^{(n)}(t_{k+1}), \boldsymbol{u}^{(n)}(t_k)] - \boldsymbol{y}_d(t_{k+1})$. In Eq. (4) $\boldsymbol{J}\{\cdot\}$ represents the Jacobian matrix of system outputs at the end of the time interval Δt (from t_k to t_{k+1}) with respect to input variables. The quantity n is the current step within the iterative process. If, in Eq. (1b), there is a direct analytical relationship between input and output the Jacobian matrix may be obtained directly. Otherwise an approximation technique must be used as follows:

$$\boldsymbol{J}_{ij} = \frac{\partial \boldsymbol{y}_i(t_{k+1})}{\partial \boldsymbol{u}_j(t_k)} \approx \frac{\partial \boldsymbol{y}_i(\boldsymbol{u}_j + \Delta \boldsymbol{u}_j)|_{t_{k+1}} - \partial \boldsymbol{y}_i(\boldsymbol{u}_j)|_{t_{k+1}}}{\Delta \boldsymbol{u}_j} \qquad i = 1, 2, 3, \ldots, p \quad \text{and} \quad j = 1, 2, 3, \ldots, q \tag{5}$$

where $\boldsymbol{u}_j$ and $\boldsymbol{y}_i$ are the jth and ith elements of the input and output vectors, respectively. $\Delta \boldsymbol{u}_j$ is the perturbation in at time t_k. In Eq. (5), the superscript n is omitted. In [12], Rutherford and Thomson have presented a modified approach for calculation of the Jacobian matrix by perturbing $\Delta \boldsymbol{u}_j$ in the negative and positive directions.

When a redundant situation exists, the Jacobian matrix is not square and it is not possible to use standard inversion in the NR iteration scheme as shown in Eq. (4). Hess et al. [6] propose the use of the pseudoinverse matrix as a solution for finding the roots of Eq. (3) when $\boldsymbol{J}$ is rectangular.

III. Development of the New Method

A. Derivation of the Algorithm

In the aircraft field of application, inverse simulation is usually carried out from a trimmed flight condition (equilibrium point) with suitable initial inputs $\boldsymbol{u}_0$. In addition, the inverse simulation process is ideal and it is assumed that no other factors such as exogenous disturbances and measurement noise are involved. Hence, the system output $\boldsymbol{y}$ can be viewed as being the result of the variation of the input vector. Now, if the input vector is regarded as a set of time-varying parameters, which is certainly independent of the state variables, this is a determining set of parameters because initial conditions (equilibrium points) of the system under investigation are assumed constant in the inverse simulation process. Moreover, $\boldsymbol{\alpha}$ is *complete* because all state variables of the investigated model can be uniquely determined with the given inputs during the ideal simulation process [17]. Therefore, the parameter set $\boldsymbol{\alpha}$ is the unique factor which determines future variations of all state-variable values because of its determining and complete properties. As a result, Eqs. (1a) and (1b) can be modified to take the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\alpha}) \tag{6a}$$

$$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\alpha}) \tag{6b}$$

In the traditional inverse simulation algorithm, the input vector is assumed constant within the time interval $t_k < t \leq t_{k+1}$. Therefore, it follows that the vector $\boldsymbol{\alpha}(t)$ will be a constant parameter vector $\boldsymbol{\alpha}_k$ in this time interval. In the following sections, this interval will be focused upon and $\boldsymbol{\alpha}_k$ will be replaced by $\boldsymbol{\theta}$ for readability.

For the system given in Eqs. (6a) and (6b) it can be shown that within the time interval $t_k < t \leq t_{k+1}$ the sensitivity function

$$\boldsymbol{Z}(\boldsymbol{\theta}, t) = \frac{\partial \boldsymbol{x}(\boldsymbol{\theta}, t)}{\partial \boldsymbol{\theta}^T} \tag{7}$$

exists [17], provided that the following are true:

1) The initial conditions for a general case depend uniquely on $\boldsymbol{\theta}$:

$$\boldsymbol{x}_k = \boldsymbol{x}_k(\boldsymbol{\theta}, t_k) \tag{8}$$

2) The variables in Eq. (8) are continuously differentiable with respect to $\boldsymbol{\theta}$.

3) The solution of $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\theta}, t)$ which satisfies Eq. (8) is also continuously differentiable with respect to $\boldsymbol{\theta}$.

4) The functions $\boldsymbol{f}$ and $\boldsymbol{g}$ are continuously differentiable with respect to their arguments.

From Eq. (8) it follows that

$$\frac{\mathrm{d}\boldsymbol{x}_k}{\mathrm{d}\boldsymbol{\theta}^T} = \frac{\partial \boldsymbol{x}_k}{\partial \boldsymbol{\theta}^T} + \boldsymbol{x}_k \frac{\mathrm{d}t_k}{\mathrm{d}\boldsymbol{\theta}^T} \tag{9}$$

Because t_k is a constant, the initial condition for the sensitivity equation shown as Eq. (7) is given by

$$\boldsymbol{Z}(\boldsymbol{\theta}, t_k) = \frac{\mathrm{d}\boldsymbol{x}_k}{\mathrm{d}\boldsymbol{\theta}^T} \tag{10}$$

In addition, it can be shown that if the conditions for Eq. (7) hold then, for a sufficiently small perturbation vector $\boldsymbol{\mu}$ of the vector $\boldsymbol{\theta}$, there exists a first-order approximation for $\boldsymbol{x}(t, \boldsymbol{\mu} + \boldsymbol{\theta}))$ within $t_k < t \le t_{k+1}$ as follows:

$$\begin{aligned} \Delta \boldsymbol{x}(t, \boldsymbol{\mu}) &= \boldsymbol{x}(t, \boldsymbol{\mu} + \boldsymbol{\theta}) - \boldsymbol{x}(t, \boldsymbol{\theta}) \\ \Delta \boldsymbol{x}(t, \boldsymbol{\mu}) &\approx \mathrm{d}\boldsymbol{x}(t, \boldsymbol{\theta}) = \sum_{i=1}^{q} \boldsymbol{Z}_i(t, \boldsymbol{\theta}) \boldsymbol{\mu}_i \end{aligned} \tag{11}$$

where q is the dimension of the $\boldsymbol{\theta}$ vector [17].

By transforming Eqs. (6a) and (6b) into the sensitivity-function form, it follows that

$$\frac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{\alpha}^T} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}^T} \cdot \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\alpha}^T} + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\alpha}^T} \tag{12a}$$

$$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{\alpha}^T} = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}^T} \cdot \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\alpha}^T} + \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\alpha}^T} \tag{12b}$$

If the following equations are defined,

$$\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\alpha}^T} = \boldsymbol{V} \qquad \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}^T} = \boldsymbol{A}(t) \qquad \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\alpha}^T} = \boldsymbol{B}(t) \qquad \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{\alpha}^T} = \boldsymbol{H}$$
$$\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}^T} = \boldsymbol{C}(t) \qquad \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\alpha}^T} = \boldsymbol{D}(t)$$

then Eqs. (12a) and (12b) can be expressed in the simplified form

$$\dot{\boldsymbol{V}} = \boldsymbol{A}(t) \cdot \boldsymbol{V} + \boldsymbol{B}(t) \qquad \boldsymbol{H} = \boldsymbol{C}(t) \cdot \boldsymbol{V} + \boldsymbol{D}(t) \tag{13}$$

Equations (13) are the continuous sensitivity equations, which provide an alternative approach to calculate the system output sensitivity function by a process involving forward simulation.

Because the parameter set $\boldsymbol{\theta}$ is complete ($\boldsymbol{\alpha}$ is complete) in the interval $t_k < t \le t_{k+1}$, the disturbance $\Delta\boldsymbol{\theta}$ results in the output variation $\Delta \boldsymbol{y}_{k+1}$. Hence, the inverse simulation process becomes an inverse problem for finding the value $\Delta\boldsymbol{\theta}$ from the following equation:

$$\Delta \boldsymbol{y}_{k+1} = \Gamma(\Delta\boldsymbol{\theta}) \tag{14}$$

where Γ represents the relationship between the variation Δy_{k+1} and the disturbance $\Delta\boldsymbol{\theta}$. In this paper, the method of *solution by inspection* [17] is adopted because it can apply to the situation where the relationship between $\Delta\boldsymbol{\theta}$ and $\Delta yk + 1$ in Eq. (14) is *incorrectly* posed [17]. The approximate value for $\Delta\boldsymbol{\theta}$ is found from a intermediate quantity $\Delta\boldsymbol{\gamma}_k$ in the region around $\boldsymbol{\theta}$ such that the distance between $\Delta \boldsymbol{y}_{k+1}$ and $\boldsymbol{g}(\Delta\boldsymbol{\gamma}_k)$ is minimal. Hence

$$\Delta\boldsymbol{\theta} \approx \Delta\boldsymbol{\gamma}_k = \sum_{n=0}^{l} \Delta\boldsymbol{\gamma}_k^{(n)} \tag{15}$$

$$\boldsymbol{\theta} \approx \boldsymbol{\gamma}_k^{(l)} = \boldsymbol{\gamma}_k^{(0)} + \Delta\boldsymbol{\gamma}_k \tag{16}$$

where $\boldsymbol{\gamma}_k^{(0)}$ is the initial condition for $\boldsymbol{\gamma}_k$.

Now a relationship can be defined as follows:

$$\boldsymbol{E} = \boldsymbol{y}_{d,k+1}(\boldsymbol{\theta}) - \boldsymbol{y}\left(\boldsymbol{\gamma}_k^{(n+1)}\right) \tag{17}$$

Using the preceding results, the term $\boldsymbol{y}(\boldsymbol{\gamma}_k^{(n+1)})$ in Eq. (17) can be replaced with its first-order sensitivity-function approximation. Furthermore, for inverse simulation in the context of aircraft applications, the conditions required are usually satisfied due to the fact that the nonlinear mathematical model of the aircraft and the defined output maneuver are continuous and smooth:

$$\boldsymbol{E} = \boldsymbol{y}_{d,k+1}(\boldsymbol{\theta}) - \left[\boldsymbol{y}\left(\boldsymbol{\gamma}_k^{(n)}\right) + \boldsymbol{H}_k^{(n)} \Delta\boldsymbol{\gamma}_k^{(n)}\right] = \Delta\boldsymbol{y}_{k+1}^{(n)} - \boldsymbol{H}_k^{(n)} \Delta\boldsymbol{\gamma}_k^{(n)} \tag{18}$$

where $\Delta\boldsymbol{y}_{k+1}^{(n)} = \boldsymbol{y}_{d,k+1}(\boldsymbol{\theta}) - \boldsymbol{y}(\boldsymbol{\gamma}_k^{(n)})$. To update Eq. (18), the current values $\boldsymbol{H}_k^{(n)}$ and $\boldsymbol{\gamma}_k^{(n)}$ have to be available in advance. Here it is possible to separate the calculation of $\boldsymbol{H}_k^{(n)}$ and $\boldsymbol{\gamma}_k^{(n)}$ by individually calculating $\boldsymbol{H}_k^{(n)}$ from Eq. (13) with the current value $\boldsymbol{\gamma}_k^{(n)}$, especially when the direct analytic input–output relationship of Eq. (6b) cannot be found. For such cases this method could show advantages in that it allows $\boldsymbol{H}_k^{(n)}$ values to be found directly by a smooth process involving a traditional one-step ($t_k < t \le t_{k+1}$) forward simulation of Eq. (13) with the initial conditions of Eq. (10), which determines sufficient conditions for the existence of $\boldsymbol{H}_k^{(n)}$. This approach is quite similar to the traditional integration-based method in that the latter calculates outputs with the current iterative input through one-step forward simulation. In conclusion, the proposed approach could obtain the Jacobian matrix through simulation instead of by means of the approximation [6].

It should be noted that the implementation of the proposed approach to calculate $\boldsymbol{H}_k^{(n)}$ increases the computational demands because the order of Eq. (13) is q times larger than that of the original system represented by Eqs. (6a) and (6b). Therefore, there is a need to balance the increase in the time required against the improved accuracy. If the analytic input–output relationship exists or can be constructed by finding the vector relative degree in advance [18], the traditional approach remains a candidate. The investigations of Lu et al. [19] have shown that with the constructed input–output relationship, the traditional inverse simulation provides greater stability and faster convergence than was previously believed possible. It can also overcome the problem [14] found by Lin. The approach of Rutherford and Thomson [12], which successfully eliminates the high-frequency oscillations, belongs to this class of methods. In addition, Eq. (13) could be suitable for situations when the system [Eqs. (1a) and (1b)] has no relative degree [20] or where high accuracy is required.

Regardless of the method selected, state variables are updated by a one-step forward simulation process. This avoids the necessity to use a *Lagrangian* approach which requires inclusion of Eq. (13) in the updating process. Hence, it is reasonable to define an LQ performance index:

$$\boldsymbol{\Pi}_k^{(n)} = \int_{t_k}^{t_{k+1}} \left(\Delta\boldsymbol{Y}_{k+1}^{(n)} - \boldsymbol{H}_k^{(n)} \Delta\boldsymbol{\gamma}_k^{(n)}\right)^T \boldsymbol{W} \left(\Delta\boldsymbol{Y}_{k+1}^{(n)} - \boldsymbol{H}_k^{(n)} \Delta\boldsymbol{\gamma}_k^{(n)}\right) \mathrm{d}t \tag{19}$$

By calculating the derivative of $\boldsymbol{\Pi}$ with respect to $\Delta\boldsymbol{\gamma}_k^{(n)}$, it is possible to find the minimum condition for Eq. (19):

$$\frac{\partial \boldsymbol{\Pi}_k^{(n)}}{\partial \Delta\boldsymbol{\gamma}_k^{(n)^T}} = \int_{t_k}^{t_{k+1}} \left(\boldsymbol{H}_k^{(n)^T} \boldsymbol{W} \boldsymbol{H}_k^{(n)} \Delta\boldsymbol{\gamma}_k^{(n)} - \boldsymbol{H}_k^{(n)^T} \boldsymbol{W} \Delta\boldsymbol{Y}_{k+1}^{(n)}\right) \mathrm{d}t = 0 \tag{20}$$

The following equations can be obtained after simplifying the preceding equation:

$$\boldsymbol{Q}_k^{(n)}\Delta\boldsymbol{\gamma}_k^{(n)} = \boldsymbol{P}_k^{(n)} \qquad \boldsymbol{Q}_k^{(n)} = \int_{t_k}^{t_{k+1}} \boldsymbol{H}_k^{(n)^T}\boldsymbol{W}\boldsymbol{H}_k^{(n)}\,\mathrm{d}t$$
$$\boldsymbol{P}_k^{(n)} = \int_{t_k}^{t_{k+1}} \boldsymbol{H}_k^{(n)^T}\boldsymbol{W}\Delta\boldsymbol{Y}_{k+1}^{(n)}\,\mathrm{d}t \tag{21}$$

According to Eq. (21), the values $\boldsymbol{Q}_k^{(n)}$ and $\boldsymbol{P}_k^{(n)}$ can be determined by integration over the interval and the value $\Delta\boldsymbol{\gamma}_k^{(n)}$ can then be found directly. Hence, the parameter $\Delta\boldsymbol{\gamma}_k^{(n)}$ can be updated at each time interval according to the following equation,

$$\boldsymbol{\gamma}_k^{(n+1)} = \boldsymbol{\gamma}_k^{(n)} + \boldsymbol{\lambda}_k \cdot \Delta\boldsymbol{\gamma}_k^{(n)} \tag{22}$$

and thus $\boldsymbol{Q}_k^{(n)}$ and $\boldsymbol{P}_k^{(n)}$ can also be updated. In Eq. (22), λ_k $(0 < \lambda_k < 1)$ is a selectively added multiplier, which can increase the convergence speed.

The algorithm will stop if the value $\Delta\boldsymbol{\gamma}_k^{(n)}$ is less than a threshold value defined in advance. The algorithm will then progress to the next time interval. Hence, the series of values of $\boldsymbol{\theta}$ can be found in the whole time space and these values again form the input vector.

It should be noted that in Eq. (21), the SA method suggests an alternative way to avoid the redundancy problem of the control inputs outnumbering the outputs [6] in that $\boldsymbol{Q}_k^{(n)}$ is calculated by the sensitivity-function matrix multiplied by its transpose. Therefore, the matrix $\boldsymbol{Q}_k^{(n)}$ is always square and this method can be applied to all kinds of systems with arbitrary input and output order. In addition, values of the weighting matrix can be selected to give priority to particular inputs and outputs. However, this may lead to more complexity and there is a need to balance this against the potential benefits in terms of the objectives.

B. Convergence Rate and Stability of the Algorithm

The algorithm in the preceding section has applied the modified approach of solution by inspection to the inverse simulation field. Thus the basic rules for the convergence and stability of the solution by inspection approach can also be applied here. Rosenwasser and Yusupov [17] show the convergence rate of this approach in the following $(n + 1)$th cost function:

$$\boldsymbol{\Pi}_k^{(n+1)} = \boldsymbol{\Pi}_k^{(n)} - \boldsymbol{P}_k^{(n)^T}\boldsymbol{Q}_k^{(n)^{-1}}\boldsymbol{P}_k^{(n)} \tag{23}$$

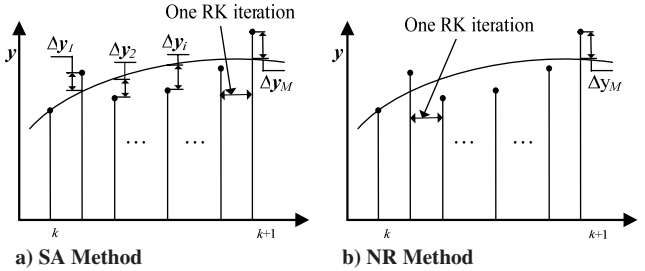
In Eq. (23), nonsingularity and the symmetric positive definite nature of the matrix $\boldsymbol{Q}_k^{(n)}$ at each iteration step is sufficient to ensure convergence for the iterative process. The symmetric positive definite property can be satisfied by selecting the weighting matrix to be symmetric positive definite in Eq. (21).

In addition, an algorithm with better stability properties, which is based on the *Hadamard inequality*, can be implemented. This method can guarantee convergence of the iterative process [17].

C. Comparisons of the SA Approach with the NR Method

Figure 1 shows the main difference between these two methods within the kth time interval. The fourth-order Runge–Kutta (RK) algorithm has been selected to carry out the one-step iteration within the interval $(\Delta t/M)$. In that period, both methods involve the same number (M) of integrations. However, the integration processes are different in that the SA method will measure the difference $\Delta\boldsymbol{y}_n$ between the ideal value and the value after each one-step RK iteration as shown in Fig. 1a. The SA method also calculates the sensitivity-function values after each RK iteration. The expressions for the matrices $\boldsymbol{Q}_k^{(n)}$ and $\boldsymbol{P}_k^{(n)}$ in Eq. (21) include these difference quantities. In contrast, the NR-based inverse simulation process only takes into account the values for the final point after the last one-step RK iteration and ignores the integration process, as shown in Fig. 1b. The NR method uses only these final values to update Eq. (4). In addition, the structural properties of the matrices $\boldsymbol{Q}_k^{(n)}$ and $\boldsymbol{P}_k^{(n)}$ provide more information compared with the single Jacobian matrix. The term $\boldsymbol{H}_k^{(n)^T}\boldsymbol{W}\boldsymbol{H}_k^{(n)}$ relates the elements of $\boldsymbol{H}_k^{(n)}$ to each other and the term $\boldsymbol{H}_k^{(n)^T}\boldsymbol{W}\Delta\boldsymbol{Y}_k^{(n)}$ connects $\boldsymbol{H}_k^{(n)}$ with the deviation from the ideal value.

Although the SA method is similar in some respects to the NR method it should be noted that the SA approach is not equivalent to carrying out an NR-based inverse simulation within a divided time interval $(\Delta t/M)$. If this equivalence existed, the NR method would require a different initial value for each divided time interval $(\Delta t/M)$. However, here the NR method keeps the assumed input constant over the whole interval Δt in the same way as the SA approach. Moreover, the minimizing optimization of the NR method is based on the difference between the actual and ideal values. However the SA approach does not minimize the output difference and deals instead with the input part $\Delta\boldsymbol{\gamma}_k^{(n)}$ during the inverse simulation process. Finally, the methodologies on which the two approaches are based are completely different. In practice, these results also suggest that further improvements in numerical accuracy may be obtained by increasing the integration number for calculation of $\boldsymbol{Q}_k^{(n)}$ and $\boldsymbol{P}_k^{(n)}$ for situations where the interval Δt is large, thus avoiding the instability discussed by Lin [14].



a) SA Method b) NR Method

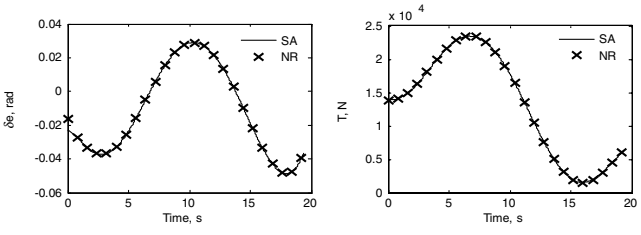
Fig. 1 Comparison of information used in the kth time interval.

IV. Numerical Examples

The case studies selected here relate to two mathematical models. One is a fixed-wing aircraft, the HS125 (Hawker 800) business jet and details of this model can be found in the Appendix; the other is a nonlinear Lynx helicopter model [21].

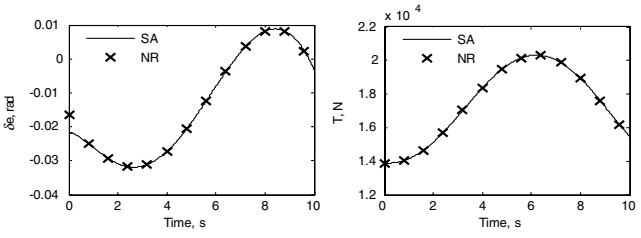
A. Application to a Fixed-Wing Aircraft

The thrust and the elevator angle act as the inputs in the algorithm implementations for inverse simulation based on the NR and SA methods. The maneuvers conducted are the hurdle-hop [12] and pop-up [22] maneuvers in the z–x vertical plane and will be characterized



a) b)

Fig. 2 Inverse simulation of hurdle-hop maneuver for the HS125 aircraft ($M = 10$, $\Delta t = 0.02$ s).



a) b)

Fig. 3 Inverse simulation of pop-up maneuver for the HS125 aircraft ($M = 10$, $\Delta t = 0.02$ s).

Table 1 Comparison of computer time and accuracy for the HS125 aircraft (z_e channel, $M = 10$)

Maneuver		$\Delta t = 0.01$ s		$\Delta t = 0.02$ s		$\Delta t = 0.025$ s		$\Delta t = 0.03$ s	
		Time, s	Var.	Time, s	Var.	Time, s	Var.	Time, s	Var.
Hurdle-hop	NR	5.6	0.0064	2.6	0.026	2.1	0.0409	1.7	0.0592
	SA	10.5	0.0019	4.4	0.0078	3.4	0.0122	2.8	0.0178
Pop-up	NR	2.7	0.0008	1.3	0.003	1.0	0.0047	0.89	0.0068
	SA	4.5	0.0002	2.1	0.0009	1.6	0.0013	1.3	0.0020

by the following polynomials, respectively:

$$z_d(t) = 64h[(t/t_m)^3 - 3(t/t_m)^2 + 3(t/t_m) - 1](t/t_m)^3 \text{ m}$$
$$V_{f_d}(t) = 61.87 \text{ m} \cdot \text{s}^{-1} \tag{24}$$

$$z_d(t) = -[6(t/t_m)^5 - 15(t/t_m)^4 + 10(t/t_m)^3]h \text{ m}$$
$$V_{f_d}(t) = 61.87 \text{ m} \cdot \text{s}^{-1} \tag{25}$$

In this application, the values of h for the hurdle-hop [Eq. (24)] and pop-up [Eq. (25)] maneuvers are 70 m and 50 m, respectively. As shown in Eqs. (24) and (25), V_f remains constant during the maneuver. The quantity t_m for the hurdle-hop maneuver is calculated by means of the following equation:

$$s = \int_0^{t_m} \dot{x}_e(t)\, \mathrm{d}t \tag{26}$$

where s is the total track distance and x_e is the longitudinal displacement. In this application, the value s for Eq. (26) is 1200 m. In addition, the value t_m for the pop-up maneuver is selected as 10 s.

The first priority is to define the calculated maneuvers in inverse simulation for both methods according to the vector relative degree, if it exists. The implemented model has a vector relative degree [2, 1]. Hence, the implemented maneuvers in the application of the inverse simulation techniques are defined in terms of acceleration. This guarantees the existence of the input–output analytic relationship. Therefore, the $\boldsymbol{H}$ matrix is available in advance for both methods. For this case, $\boldsymbol{W}$ is selected as a unit matrix and $\boldsymbol{H}$ and $\Delta\boldsymbol{\gamma}$ are calculated as shown in Eq. (27). A symbolic differentiation program is used to derive this analytic formulation [7].

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_{1\delta_e} & \boldsymbol{H}_{1T_t} \\ \boldsymbol{H}_{2\delta_e} & \boldsymbol{H}_{2T_t} \end{bmatrix} = \begin{bmatrix} \frac{\partial \ddot{z}_e}{\partial \delta_e} & \frac{\partial \ddot{z}_e}{\partial T_t} \\ \frac{\partial \dot{V}_f}{\partial \delta_e} & \frac{\partial \dot{V}_f}{\partial T_t} \end{bmatrix} \qquad \Delta\boldsymbol{\gamma} = \begin{bmatrix} \Delta\delta_e \\ \Delta T_t \end{bmatrix} \tag{27}$$

In all the following sections the conditions have been defined to give a fair comparison between NR-based and SA-based methods by selecting the same integration number for Eqs. (2a) and (21) within the same interval Δt. All simulations have been run on a Dell desktop PC which has a 2.8 GHz processor and a 1 Gbyte memory.

After a first application of these two maneuvers, the results are shown in Figs. 2 and 3.

Clearly Figs. 2 and 3 prove that the inverse simulation based on the SA method provides the same results compared with the traditional method based on the NR algorithm. The computational time and accuracy for the results are included in Table 1 for a variety of different Δt values. Here, the accuracy represented by the variance (Var.) describes the consistency of the results from the forward simulation with calculated inputs to the desired maneuver. During the simulation, the values for the V_f channel for both approaches are very similar in terms of accuracy. Thus, they are omitted here.

Table 1 shows that the results from the SA method can be about four times more accurate than results from the NR method for each time interval Δt. Although this is achieved at the cost of an increase in computation time, it is still acceptable for offline inverse simulation as the data show.

During the investigation, it was found that both algorithms show a high-frequency oscillation phenomenon within the sampling-rate

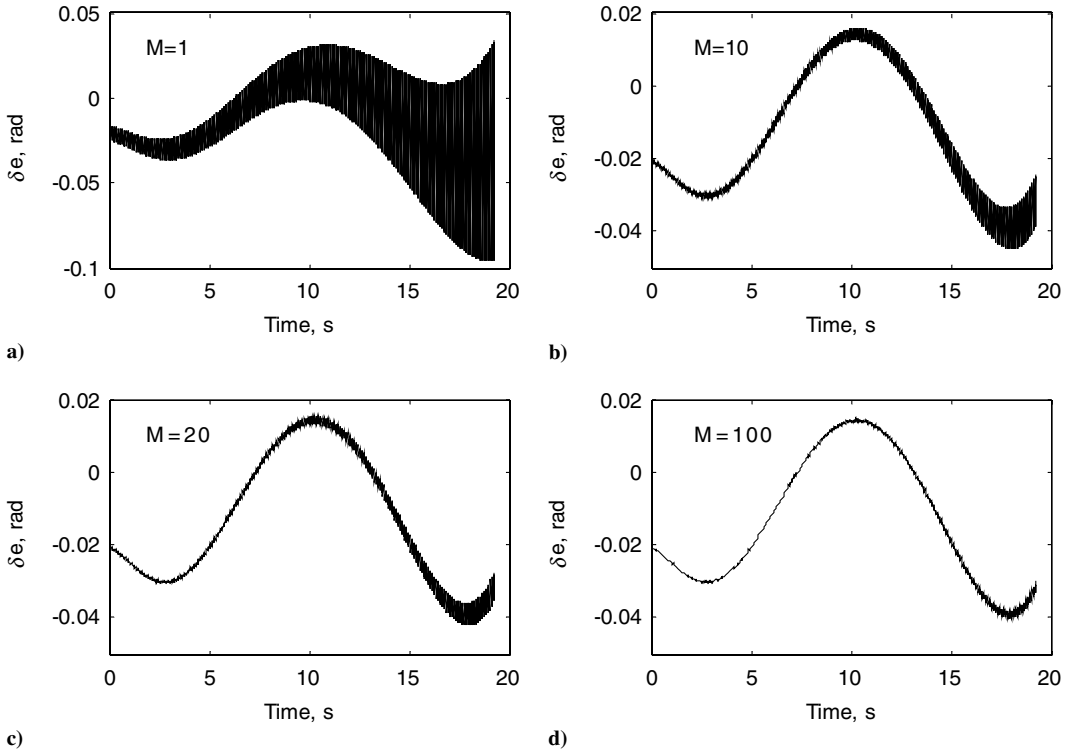


Fig. 4 Inverse simulation of hurdle-hop maneuver for the HS125 aircraft (SA, $\Delta t = 0.031$ s).

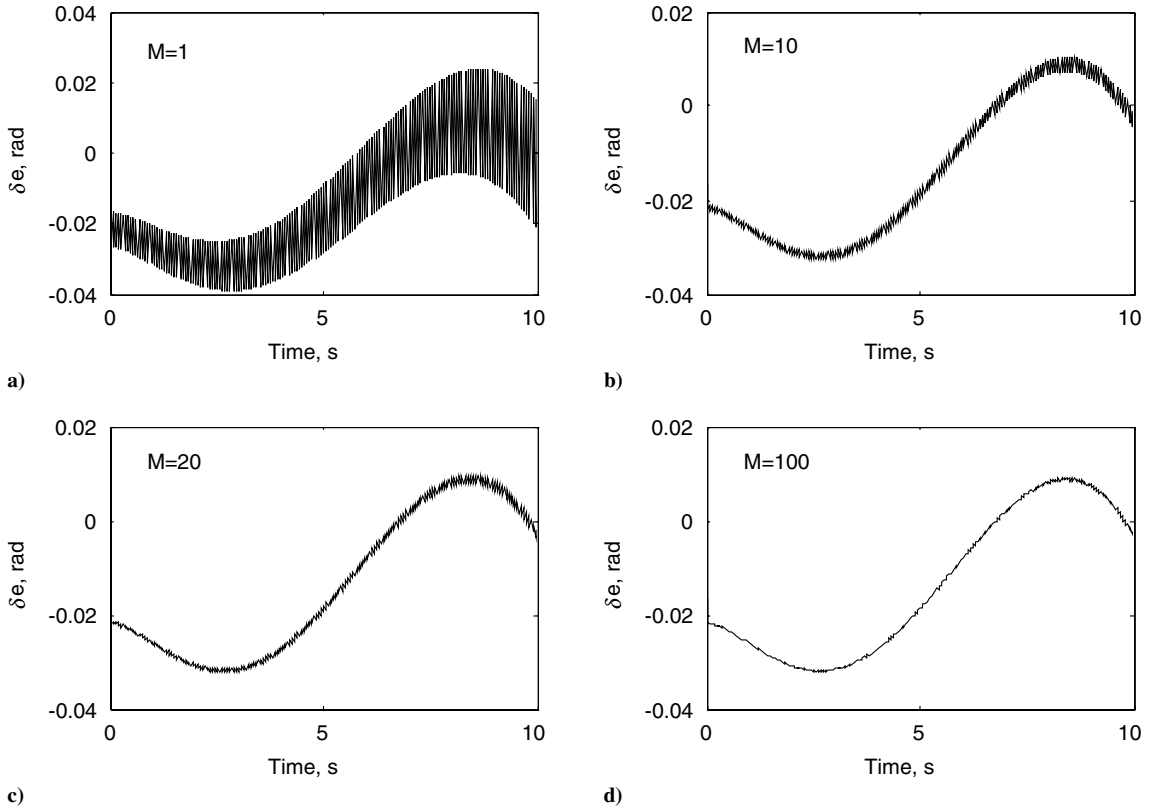


Fig. 5 Inverse simulation of pop-up maneuver for the HS125 aircraft (SA, $\Delta t = 0.031$ s).

interval (0.0285 s, 0.0315 s) for $M = 1$. Moreover, values of Δt smaller than the values within this interval can eliminate this problem whereas larger values can lead to convergence problems for both algorithms for $M = 1$. This phenomenon possibly arises from the fact that the sampling rates lie within the interval for which the algorithms encounter convergence problems. The similar convergence properties for both algorithms for $M = 1$ are probably due to the iterative nature of the calculations in both cases, as shown in Fig. 1.

It is of interest to investigate whether the high-frequency oscillation phenomenon can be substantially reduced by increasing the value of M. The results from the SA-based method are shown in Figs. 4 and 5. In addition, the same series of tests ($M = 1$ to $M = 100$) have been carried for the NR method and they provide results that are, in all cases, exactly the same as the highly oscillatory results shown in Figs. 4a and 5a. In other words, increasing the value of M in the NR method has no effect on results. The time required and the accuracy achieved by the two methods are shown in Table 2.

The results in Figs. 4 and 5 show that the high-frequency oscillation phenomenon can be substantially reduced in the SA method with a suitable selection of integration number M for Eq. (21). Table 2 provides information about the effect of M on the accuracy of results. For the SA method, an accuracy improvement is evident for values from $M = 1$ to $M = 20$. Beyond $M = 20$, these accuracy benefits are still obtained but are less marked and lead to significantly increased computation time. This suggests that there is a tradeoff between accuracy and computation time in the choice of M for practical applications. A large value M is preferred but is not essential. In addition, the results also show that it is not essential for the SA method to use a small Δt value to achieve good results, in contrast with the NR approach. Under some conditions, this means that the problems associated with small Δt values which lead to high-frequency oscillations in the NR approach [14] could be avoided. The good performance for the SA approach is probably due to the increase in the information used in the inverse simulation process, as discussed in Sec. III.

B. Application to a Lynx Helicopter Model

The model considered in this section is a nonlinear Lynx helicopter developed within the University of Glasgow [21]. The model involves of five subsystems: fuselage, tail plane, fin, rotor, and tail rotor. The rotor is modeled as an actuator disk using momentum theory and a blade element approach. Similar maneuvers to the HS125 case (i.e., hurdle-hop and pop-up) are used but with dimensions more appropriate to helicopter flight selected [23]. The four inputs are the traditional four control channels: collective pitch θ_0, longitudinal cyclic pitch θ_{1s}, lateral cyclic pitch θ_{1c}, and tail rotor collective pitch $\theta_{0\text{tr}}$. The four outputs are x_e, y_e, z_e, and Ψ.

The original NR-based software package [21] is designed on the basis of the second derivatives of the outputs. This can improve the numerical stability. To provide a useful comparative benchmark, the SA method is also implemented in a similar way. The Jacobian matrix in the original package [21] is calculated by the approximation method. Because some quantities within the Lynx

Table 2 Comparison of computer time and accuracy for the HS125 aircraft (z_e channel, $\Delta t = 0.031$ s)

Maneuver		$M = 1$		$M = 10$		$M = 20$		$M = 100$	
		Time, s	Var.	Time, s	Var.	Time, s	Var.	Time, s	Var.
Hurdle-hop	NR	0.34	0.0632	2.7	0.0632	4.4	0.0632	15.2	0.0632
	SA	0.44	0.0632	1.6	0.0191	7.4	0.0168	26.2	0.0161
Pop-up	NR	0.19	0.0073	0.88	0.0073	1.6	0.0073	7.42	0.0073
	SA	0.22	0.0073	1.28	0.0022	2.4	0.0020	12.5	0.0018

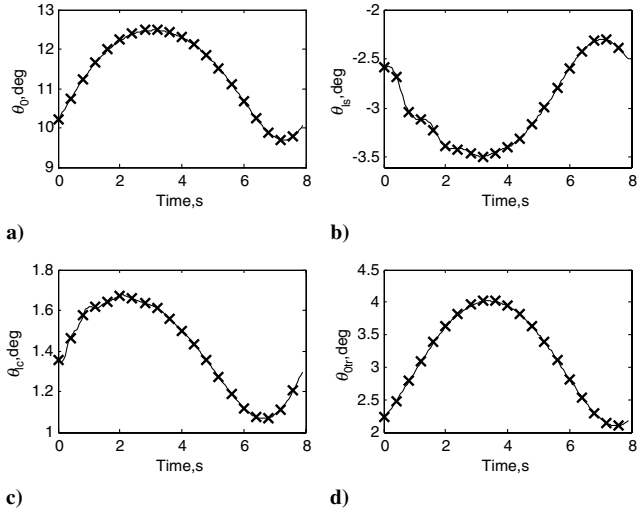


Fig. 6 Inverse simulation of pop-up maneuver for the Lynx helicopter ($M = 2$, $\Delta t = 0.05$ s, 80 kts), solid line, SA; x, NR.

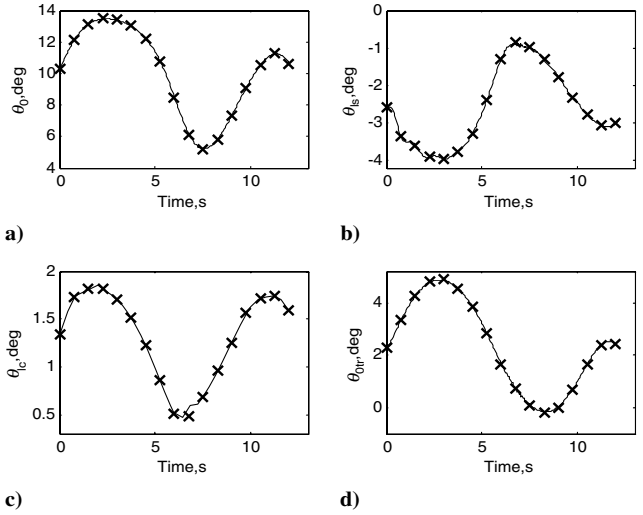


Fig. 7 Inverse simulation of hurdle-hop maneuver for the Lynx helicopter ($M = 2$, $\Delta t = 0.05$ s, 80 kts), solid line, SA; x, NR.

model are obtained by numerical methods and are not expressed in the form of simple analytical relationships in the model equations, it is impossible to calculate the sensitivity functions through the approach suggested in Eqs. (12a) and (12b). Therefore, the approximation method is used to calculate the $\boldsymbol{H}$ matrix, which has the following form:

$$\boldsymbol{H} = \begin{bmatrix} \frac{\partial \ddot{x}_e}{\partial \theta_0} & \frac{\partial \ddot{x}_e}{\partial \theta_{1s}} & \frac{\partial \ddot{x}_e}{\partial \theta_{1c}} & \frac{\partial \ddot{x}_e}{\partial \theta_{0tr}} \\ \frac{\partial \ddot{y}_e}{\partial \theta_0} & \frac{\partial \ddot{y}_e}{\partial \theta_{1s}} & \frac{\partial \ddot{y}_e}{\partial \theta_{1c}} & \frac{\partial \ddot{y}_e}{\partial \theta_{0tr}} \\ \frac{\partial \ddot{z}_e}{\partial \theta_0} & \frac{\partial \ddot{z}_e}{\partial \theta_{1s}} & \frac{\partial \ddot{z}_e}{\partial \theta_{1c}} & \frac{\partial \ddot{z}_e}{\partial \theta_{0tr}} \\ \frac{\partial \ddot{\Psi}_e}{\partial \theta_0} & \frac{\partial \ddot{\Psi}_e}{\partial \theta_{1s}} & \frac{\partial \ddot{\Psi}_e}{\partial \theta_{1c}} & \frac{\partial \ddot{\Psi}_e}{\partial \theta_{0tr}} \end{bmatrix} \qquad \Delta\boldsymbol{\gamma} = \begin{bmatrix} \Delta\theta_0 \\ \Delta\theta_{1s} \\ \Delta\theta_{1c} \\ \Delta\theta_{0tr} \end{bmatrix} \tag{28}$$

After a first application of these two maneuvers, the results are shown in Figs. 6 and 7.

Figures 6 and 7 show that the inverse simulation based on the SA method provides almost the same results as the traditional method based on the NR algorithm for the Lynx helicopter model. The reason that $M = 2$ is chosen is mainly reduction in the computation time but relates also to the validity of previous statements concerning the choice of M in Section IV.A. The data for the accuracy of the results are included in Table 3 for two different Δt values. In Table 3, the results for $M = 1$ from the SA and NR methods are ignored because they are exactly the same as for $M = 2$ from the NR method.

The maneuvers adopted here are carried out in the x–z plane. Hence, in comparing the accuracy of the results particular attention must be given to the output channels x_e and z_e. The results leading to the figures in Table 3 show, as in the previous examples, that the accuracy can be increased if the value of M becomes larger (although the results for $M = 1$ are not shown in the table). In addition to the benefits from the different M values, Table 3 also shows that the accuracy from the SA method can be more than four times that of corresponding results from the NR method for each time interval Δt. It should be noted that comparison of the computational time is of little relevance in this case, because even for the NR method, the inverse simulation process requires hours of the computer time for this complex nonlinear model. The time for the SA method is greater than for the NR method but by less than the factor of 2.

V. Conclusions

In this paper a procedure for inverse simulation based on sensitivity analysis is developed. Its stability and convergence properties have been discussed. This approach provides a new way to calculate the Jacobian matrix by solving a sensitivity equation. Although it involves increased computational complexity this method avoids the approximations involved in other published approaches. In addition, the new approach can be applied to arbitrarily redundant situations. The simulations with an HS125 aircraft model and a Lynx helicopter model for the hurdle-hop and pop-up maneuvers show that the new method provides more accurate results than the traditional approach with an acceptable increase of the computational time. In addition, it can deal with the high-frequency oscillation problem that appears in the inverse simulation process by increasing the integration number.

Appendix: Configuration Data and Vehicle Equations of Motion

The appropriate configuration data for the HS125 (Hawker 800) Business Jet are given in Table 1.

The vehicle equations of motion are

$$\dot{u} = -qw + \frac{X}{m_m} - g\sin\Theta \qquad \dot{w} = -qu + \frac{Z}{m_m} + g\cos\Theta$$

$$\dot{q} = \frac{M_m}{I_{yy}} \quad \text{and} \quad \dot{\Theta} = q \qquad \dot{x}_e = u\cos\Theta - w\sin\Theta$$

$$\dot{z}_e = u\sin\Theta + w\cos\Theta$$

where u, w are velocity components in body axes, q is pitch angular velocity component in body axes, and Θ is the fuselage-pitch attitude. X, Z, and M_m are the external forces and moment. The variables x_e and z_e are the longitudinal displacement and the vertical displacement.

Table 3 Comparison of output accuracy for the Lynx helicopter ($M = 2$, 80 kts)

Maneuver	Output	x_e, m		y_e, m		z_e, m		Ψ, deg	
	Δt	0.05 s	0.01 s	0.05 s	0.01 s	0.05 s	0.01 s	0.05 s	0.01 s
Hurdle-hop	NR	0.0024	$3.57e^{-5}$	0.0013	$4.26e^{-5}$	0.0534	0.0021	$9.34e^{-10}$	0.0000
	SA	0.0005	$1.72e^{-5}$	0.0008	0.0003	0.0131	0.0005	$3.35e^{-7}$	0.0000
Pop-up	NR	$1.69e^{-5}$	$6.56e^{-7}$	$9.44e^{-5}$	$4.53e^{-6}$	0.0297	0.0012	$3.13e^{-11}$	0.0000
	SA	$3.34e^{-6}$	$6.00e^{-7}$	$6.18e^{-5}$	$4.35e^{-6}$	0.0074	0.0003	$3.78e^{-12}$	0.0000

Acknowledgments

Linghai Lu gratefully acknowledges the award of a Glasgow University Scholarship and an Overseas Research Studentship from the British government. D. G. Thomson and D. J. Murray-Smith acknowledge support from the U.K. Engineering & Physical Science Research Council through grant *GR/S91024/01*. In addition, the authors wish to thank Marat Bagiev, Department of Aerospace of University of Glasgow, for supplying the Lynx helicopter model used in this paper. Finally, the helpful comments from the anonymous reviewers of this paper are gratefully acknowledged.

References

[1] Saltelli, A., and Scott, M., "Guest Editorial: The Role of Sensitivity Analysis in the Corroboration of Models and Its Link to Model Structural and Parametric Uncertainty," *Reliability Engineering and System Safety*, Vol. 57, No. 1, 1997, pp. 1–4.

[2] Ratto, M., Lodi, G., and Costa, P., "Sensitivity Analysis of a Fixed-Bed Gas Solid TSA: The Problem of Design with Uncertain Models," *Separations Technology*, Vol. 6, No. 4, 1996, pp. 235–245.

[3] Sato, J., Ueda, T., and Ohmori, H., "Sensitivity Analysis and Identification of Time Varying Parameters in Wastewater Treatment System Based on Activated Sludge Models," *Modelling and Control for Participatory Planning and Managing Water Systems*, International Federation of Automatic Control, Venice, Italy, 2004.

[4] Thomson, D. G., and Bradley, R., "Prediction of the Dynamic Characteristics of Helicopters in Constrained Flight," *Aeronautical Journal*, Vol. 94, No. 940, 1990, pp. 344–354.

[5] Kato, O., and Saguira, 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.

[6] Hess, R. A., Gao, C., and Wang, S. H., "Generalized Technique for Inverse Simulation Applied to Aircraft Maneuvers," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 5, 1991, pp. 920–926.

[7] Lee, S., and Kim, Y., "Time-Domain Finite Element Method for Inverse Problem of Aircraft Maneuvers," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, 1997, pp. 97–103.

[8] de Matteis, G., de Socio, L. M., and Leonessa, A., "Solution of Aircraft Inverse Problems by Local Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 3, 1995, pp. 567–571.

[9] Celi, R., "Optimization-Based Inverse Simulation of a Helicopter Slalom Manoeuvre," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 2, 2000, pp. 289–297.

[10] Gao, C., and Hess, R. A., "Inverse Simulation of Large-Amplitude Aircraft Manoeuvres," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 4, 1993, pp. 733–737.

[11] Anderson, D., "Modification of a Generalised Inverse Simulation Technique for Rotorcraft Flight," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 217, No. 1070, 2003, pp. 61–73.

[12] Rutherford, S., and Thomson, D. G., "Improved Methodology for Inverse Simulation," *Aeronautical Journal*, Vol. 100, No. 993, 1996, pp. 79–86.

[13] Rutherford, S., and Thomson, D. G., "Helicopter Inverse Simulation Incorporating an Individual Blade Rotor Model," *Journal of Aircraft*, Vol. 34, No. 5, 1997, pp. 627–634.

[14] Lin, K., "Comment on 'Generalized Technique for Inverse Simulation Applied to Aircraft Maneuvers'," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 6, 1993, pp. 1196–1197.

[15] Yip, K. M., and Leng, G., "Stability Analysis for Inverse Simulation of Aircraft," *Aeronautical Journal*, Vol. 102, No. 1016, 1998, pp. 345–351.

[16] Avanzini, G., de Matteis, G., and de Socio, L. M., "Two-Timescale Inverse Simulation of a Helicopter Model," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, 2001, pp. 330–339.

[17] Rosenwasser, E., and Yusupov, R., *Sensitivity of Automatic Control Systems*, Chapman & Hall/CRC, Boca Raton, FL, 2000.

[18] Sastry, S., *Nonlinear Systems: Analysis, Stability, and Control*, Springer–Verlag, New York, 1999.

[19] Lu, L., Murray-Smith, D., and Thomson, D. G., "Issues of Numerical Accuracy in Inverse Simulation," Systems and Control Report Series, Dept. of Electronics and Electrical Engineering, Univ. of Glasgow, Rept. 06/01, Glasgow, Scotland, 2005.

[20] Ramakrishna, V., Hunt, L. R., and Meyer, G., "Parameter Variations, Relative Degree, and Stable Inversion," *Automatica : the Journal of IFAC, the International Federation of Automatic Control*, Vol. 37, No. 6, 2001, pp. 871–880.

[21] Bagiev, M., Rotorcraft Inverse Simulation Package, Software Package, Ver. 2.1, Dept. of Aerospace Engineering, Univ. of Glasgow, Glasgow, Scotland, 2006.

[22] Thomson, D. G., and Bradley, R., "The Use of Inverse Simulation for Conceptual Design," *Proceeding of European Rotorcraft Forum*, CSA Illumina, Oxford, 1990, pp. 627–634.

[23] Thomson, D. G., and Bradley, R., "The Principles and Practical Application of Helicopter Inverse Simulation," *Simulation Practice and Theory*, Vol. 6, No. 1, 1998, pp. 47–70.