INVERSE SIMULATION METHODS AND APPLICATIONS

David J. Murray-Smith, Linghai Lu

Department of Electronics and Electrical Engineering,
University of Glasgow,
Glasgow G12 8LT, Scotland, U.K.

djms@elec.gla.ac.uk (David Murray-Smith)

Abstract

This paper provides background information for a tutorial which introduces the concepts of inverse simulation. Conventional simulation methods involve finding the response of a system to a particular form of input or disturbance for a given set of initial conditions. Inverse simulation methods reverse this process and attempt to find the control inputs required to achieve a particular response. This inverse approach has been applied with success in a number of fields but it is in aeronautical applications that it has found most favour so far, particularly in the context of helicopter flight mechanics. The piloting strategy required for an aircraft to perform a defined manoeuvre is predicted and can be used to analyse handling qualities, pilot workload, agility and control system performance. Recently the methods have also been applied to problems of ship manoeuvring and navigation and to underwater vehicles. The methods are particularly well suited to investigation of actuator performance and the effects of actuator limiting in control system design. The paper outlines the algorithms most widely used for inverse simulation and introduces some relatively novel approaches which are believed to have some advantages for applications involving control system performance investigations. The paper includes discussion of possible numerical problems encountered when using these algorithms and more fundamental issues associated with the dynamic properties of inverse models. The impact of inverse simulation in providing enhanced understanding about the dynamic properties of the system under investigation is emphasized. Although aeronautical applications are dominant in published work in this field the approach is of general applicability and any dynamic system may be treated in the same way.

Keywords: Inverse, Simulation, Model, Manoeuvre, Helicopter, Ship.

Presenting Author's biography

David Murray-Smith is an Emeritus Professor of the University of Glasgow. Until October 2005 he was Professor of Engineering Systems and Control in the Department of Electronics and Electrical Engineering, where he is still actively involved in research. His current research interests are in system modelling and control techniques applied to a range of engineering and biomedical systems. With Linghai Lu (Research Student and co-author of this paper) he is engaged on research on the further development and application of inverse simulation methods and their application in feed-forward control system design.

1 Introduction

The conventional process of forward modelling and simulation is not the only approach possible for gaining an understanding of the behaviour of complex dynamic systems. The idea of inverting a system to determine the inputs that are necessary to give a specified set of system output variables can be traced back to the 1930s when Jones [1] investigated the effects of gusts on an aircraft by inverting a linearised aircraft model.

Linear single-input single-output models in transfer function form, with the same order of numerator and denominator, can be inverted directly so that the numerator of the original model becomes the denominator of the inverted model. For other classes of system the situation is more complicated but, when inversion is possible, new insight can often be obtained which is important both for system design and for control.

In the 1960s and 1970s a class of methods, generally termed ***dynamic inversion***, was developed for multivariable nonlinear minimum-phase (MP) systems by some pioneers (e.g. by Brockett [2]; Dorato [3] and Hirschorn [4]). The procedure of dynamic inversion involves the transformation of the original nonlinear system into a linear and controllable system by means of a nonlinear change of state space coordinates and a nonlinear state feedback control law with the application of differential geometry concepts. Subsequently, further efforts were made to develop approaches for nonminimum-phase (NMP) systems (e.g. by Isidori *et al.* [5], [6]) and in the mid 1990s, Devasia *et al.* [7] achieved further significant progress by developing a method based on noncausal *Picard-like* iteration to obtain bounded inversions. In recent years, based on this earlier work, several approaches have been developed to achieve the causal inversion of NMP systems (e.g. [8-11] as well as methodology for the non-hyperbolic problem [12,13].

This group of approaches has attracted considerable attention since these techniques offer a potentially powerful methodology for control system design. In 1995 Reiner *et al.* published an account [14] of the design of an outer-loop flight controller using structured singular value (μ) synthesis, based on the linearised model obtained by an inner-loop controller designed by dynamic inversion. In 2003 Hu *et al.* [15] followed a similar approach to design a controller for ship course keeping.

However, all of the above approaches based on model inversion principles have some shortcomings that restrict their applications. Firstly, the available mathematical approaches are quite tedious and difficult to apply even for MP systems, especially in applications involving high-order models. Secondly, most available methods only achieve noncausal inversion and depend on the whole desired output trajectory. They also usually require specific properties of smoothness of the desired trajectories to provide the ideal input vector. Also, the valid domain of these methods is mainly restricted to tracking of trajectories involving small amplitudes due to the form of the algorithms [16] and these techniques may therefore be unsuitable for more severe types of manoeuvre.

During the period when major developments were taking place in terms of the techniques for model inversion, a numerical process termed ***inverse simulation*** was being developed to allow inversion of linear and nonlinear dynamic models using numerical methods instead of an analytical approach. This idea attracted particular attention within the field of aerospace engineering and in some other application areas. It is an approach that generates the forward control inputs such that a mathematical model of a system can follow a prescribed trajectory in state space.

Inverse simulation aims to determine, by numerical processes, the system inputs required to produce a given output response. Interest in inverse simulation methods has been particularly strong in the field of aircraft flight mechanics and this approach has received special attention in the case of helicopters and other forms of rotorcraft, which involve complex and highly nonlinear models. For such an application the input to the inverse simulation is the required flight path and the output information represents the piloting commands needed to achieve this trajectory.

Many contributions have been made to the field of inverse simulation since the early 1990s. Much of the research has been concerned with the establishment of robust and numerically stable approaches to inverse simulation (e.g. through developments by Kato & Saguira [17] ; Thomson et al. [18-20]; Anderson [21]; Hess *et al*, [22,23]; de Matteis *et al.*, [24]; Lee & Kim [25]; Avanzini *et al.* [26]; Celi [27]; Lu *et al.* [28]).

The reasons for the popularity of inverse simulation methods arise from the practical usefulness of this approach in various fields. Thomson and Bradley [29, 30] highlighted the value of inverse simulation techniques for the investigation of the handling qualities, manoeuvrability, and agility of a hypothetical battlefield utility helicopter at the conceptual design stage. The quantitative assessment of helicopter handling qualities and validation of the model was approached by analysing specific measures of aircraft response known, as attitude quickness criteria, using simulated flight in conjunction with standard Mission Task Elements (MTEs) which can be used to describe specific forms of aircraft trajectory (e.g. [31-33]). Secondly, inverse simulation has been shown to facilitate investigation of required actuator characteristics and control actions. A specific example of an investigation of this kind involved study of the control actions required following engine failures during takeoff from offshore platforms [34]. In addition, inverse simulation has been investigated for output-tracking and inversion-based controllers (e.g. [26], [35-37]).

As with dynamic inversion, traditional inverse simulation techniques suffer from some well known problems. Numerical issues, such as non-convergence, rounding errors and phenomena involving sustained high-frequency oscillations have been found in most of the techniques that are available (e.g. [22,23], [32], [38-40]). Secondly, redundancy issues when the number of inputs is greater than the number of outputs may also lead to non-convergence of inverse solutions (e.g. [24,25], [38], [40]). Thirdly, in some cases, oscillations of much lower frequency also appear in the results. This phenomenon, which has been termed “constraint oscillations”, often gives stable results but with gently damped oscillating components (e.g. [21], [29]). Fourthly, the numerical processes in traditional methods of inverse simulation involve use of derivative information, such as in the Jacobian matrix or the Hessian matrix. This limits their application only to smooth trajectories and models which have no input constraints.

Another, more fundamental, issue is that the stability of inverse simulation methods for NMP systems has received relatively little attention. Also, rather surprisingly, little consideration has been given in most previously published work to the relationship between model inversion and inverse simulation techniques.

2 Classification of Inverse Simulation Approaches

Inverse simulation is commonly carried out either by a direct approach based on differentiation or iteratively using integration methods. The first significant published accounts of the problems of inverse simulation for aircraft applications were those of Kato and Saguira [17] and Thomson [18]. Their methods involved numerical differentiation of the vehicle state variables, with respect to time. The main advantage of this approach is fast convergence speed. However, it may suffer from problems of numerical rounding error and involves ad-hoc approaches for specific applications for different types of vehicles. Sentoh and Bryson [35] defined the inverse process, in the context of an aircraft application, as a LQ optimal problem that minimises the integral of a weighted square sum of the deviations from a straight flight-path and control surface deflections. They demonstrated the approach by an application to feed-forward control. However, this method was found to suffer from significant practical limitations and involved a relatively cumbersome procedure. Moreover, this method is not suitable for the redundancy problems that can arise where the number of control inputs is greater than the number of path constraints [26].

In the early 1990s members of a research group at the University of California, Davis [22,23] proposed what is now the most commonly used approach that formulates the inverse problem as an iterative procedure involving repeated forward simulation involving a conventional integration process. This approach does not require time differentiation of the specified path constraints. Instead, it involves a procedure that calculates the partial derivative of the output vector with respect to the input vector through a numerical algorithm. In addition, redundancy problems can be overcome by use of the Moore-Penrose inverse. Unlike the earlier approaches based on differentiation, the structure of the algorithm in this case means that the integration-based method is less model-specific. Thus it can accommodate different models without restructuring the algorithm itself. One of the drawbacks of this technique is that it is an order of magnitude slower than the approach involving differentiation with respect to time.

In an approach similar to the integration-based algorithm, de Matteis *et al.* [24] presented an alternative local optimisation concept to eliminate the control redundancy problem. This involved adding new path constraints at the cost of evaluating the Hessian matrix numerically. The implementation by de Matteis *et al.* involved a modified *Broyden, Fletcher, Goldfarb and Shannon* (BFGS) quasi-Newton method. It should be noted that, in practice, it is not always feasible to construct new path constraints for a special performance requirement as this approach may not always lead to a solution due to the searching region being restricted. By incorporating a two timescale approach to simplify the complexity of aircraft models, this method has been successfully demonstrated on a F-16 fighter aircraft model [26] and a Bell aH-1G single rotor helicopter model [40].

Lee and Kim [25] formulated the inverse simulation problem as a general optimisation problem by defining a performance index constrained by equality conditions that are a function of state variables. Then the performance index is discretized by the finite element method and the final governing equation is solved by the *Levenberg-Marquardt* (LM) algorithm. This can avoid the control redundancy problem by appropriate selection of the performance index and constraint condition. Therefore, the procedure does not involve numerical differentiation or integration processes. As a result, it overcomes the problem of ill conditioning and sensitivity issues associated with initial guessed values. However, the performance improvement is achieved at the cost of enormously increased complexity of the inverse simulation process.

Celi [27] solved the inverse problem by borrowing some ideas from the optimisation field. In fact, unlike the approaches based on local optimisation, his method considers pilot inputs as design variables in the global space. Celi's approach (which has been applied to helicopter problems) can generate pilot inputs that differ from those expected from the application of other techniques due to the existence of a family of valid trajectories. There may also be difficulty with this approach if it is required to calculate the whole trajectory at one time. If this is the case, the results may show poor consistency between the converged solution and the desired trajectory. This problem can be solved by performing the optimisation over overlapping consecutive segments of the trajectory rather than over the entire time history. In addition, problems of multiple solutions may appear. Multiple solutions may actually assist the investigator in aircraft handling-qualities studies but creates difficulties if the inverse solution is being used for some other purposes, such as simulation model validation. As a consequence, additional constraints are required to achieve a unique solution.

Finally, a number of other researchers also have made contributions to the inverse simulation field. Anderson [21] proposed an enhanced NR method by combining Hess's approach with a bisection method through which each change of controls is multiplied by an additional scale factor. He stated that the numerical stability of the inverse simulation can be significantly improved and an order of magnitude reduction can be achieved in both the tracking error and the control deflections This is shown from results obtained from an application to a helicopter model with an individual blade representation. In fact, his method can be considered as another modification of the calculation of the Jacobian matrix and is quite similar to the inverse Broyden method [41], but simpler. However the bisection method has some drawbacks. One disadvantage is that when the searching interval for a real root is decreased, the speed of convergence becomes very slow due to the computational load. It is also difficult to achieve high accuracy using this approach.

Lu *et al*. [28] have recently proposed an approach based on sensitivity-analysis (SA) to solve some numerical problems existing in the traditional integration method. In addition, Lu *et al* [42] have developed a derivative-free approach which leads to improved numerical stability. This approach also allows the inclusion of actuator saturation and other limits in the model being investigated as well as discontinuous manoeuvres.

One approach that has generated interest for some specific applications is based on principles that were used very successfully in analogue computers to produce divider elements from analogue multipliers and inverse functions from simple function generators. The idea is to embed a forward simulation of the system under investigation within a high gain feedback system. Then, in the simple linear case at least, it is clear from feedback theory that it is possible to generate an inverse simulation in a very simple and straightforward fashion. This involves using the required output time history as reference. The necessary input may then be found automatically by monitoring the input signal that is applied to the forward simulation within the high-gain feedback loop. Although this may appear very straightforward, it does require considerable effort to design an appropriate feedback system, especially for situations involving multi-input multi-output simulation models, NMP systems or nonlinear simulation models. Further details may be found in the work of von Grünhagen *et al.* (e.g. [36],[43]) where these ideas have been used successfully for inverse simulation for helicopter applications.

Techniques based on the properties and methods for solution of differential-algebraic equations (DAE) provide another interesting avenue of approach which has been successfully explored by a number of researchers (e.g. [44]). These DAE methods have attracted most attention within communities that make use of software tools such as *Modelica* and *Dymola* that provide facilities involving the use of algorithms for the solution of DAEs.

2.1 Methods in which derivative information is used

Based on the above literature review, the various techniques in which derivative information is used may be categorised as follows:

2.1.1 Optimisation methods involving local optimisation

Approaches that fall within this category include: The LQ problem [35]; the local optimisation approach with the BFGS algorithm [25]; the two timescale approach [26]; the method based on sensitivity analysis [28].

2.1.2 Optimisation methods involving global optimisation

Approaches in this category include the general optimisation problem involving equality conditions [25] and also the optimisation approach of Celi [27].

2.1.3 Differentiation methods

This approach involves numerical differentiation of the vehicle constrained variables, with respect to time, until the control variables can be solved explicitly (e.g. [17,18]). More discussion of this approach may be found in an earlier review paper by Murray-Smith [45] and in a very recent and extensive review by Thomson and Bradley [46].

2.1.4 Integration methods

In integration-based methods the value of the control variables that satisfy the constraints are found iteratively within a sampling interval using a form of Newton-Raphson algorithm or a variation thereof (e.g.

[21-23]). Integration-based algorithms are the most widely used techniques at the present time and provide a benchmark against which other methods are commonly compared.

2.2 Optimisation methods that do not involve derivative information

Methods for inverse simulation that do not use derivative information are essentially optimisation methods that are search based. One approach of this kind that has been used successfully is a derivative free method based on the Nelder-Mead (NM) algorithm [42].

2.3 Methods based on the principles of high-gain feedback systems.

The application of high-gain feedback principles for inverse simulation has, so far, involved applications that have specific requirements for high-speed (often real-time) inverse simulation (e.g. [36]). Particular attention has been given to linear models using this approach (e.g. [43]).

2.4 Methods based on the solution of differential algebraic equations (DAEs)

Methods based on differential algebraic equations (DAEs) have attracted particular attention within research groups that make extensive use of *Modelica* and the associated simulation tool *Dymola* (e.g. [44]). A continuous *Modelica* model is mapped to a DAE model of the form:

$$0 = f(\dot{\boldsymbol{x}}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}) \tag{1}$$

where $\boldsymbol{x}(t)$ are variables that appear differentiated in the model, $\boldsymbol{y}(t)$ are algebraic variables and $\boldsymbol{u}(t)$ are known input functions of time. An inverse model of the DAE is constructed by simply exchanging the meaning of variables. Previously unknown variables from the vectors $\boldsymbol{x}$ and $\mathbf{y}$ in this DAE model are treated as known inputs and an appropriate subset of the vector $\boldsymbol{u}$ is treated as involving unknowns. The resulting equation remains a DAE and this can be handled using standard methods for such a mathematical description.

3 Outline descriptions of some selected inverse simulation methods

3.1 The differentiation-based approach

Consider a nonlinear system described by equations of the form:

$$\dot{x} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{2}$$

$$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) \tag{3}$$

where $\boldsymbol{f} \in \mathrm{R}^m$ is the set of nonlinear ordinary differential equations describing the system, $\boldsymbol{g} \in \mathrm{R}^p$ is the set of algebraic equations that construct the expected outputs, and $\boldsymbol{u} \in \mathrm{R}^q$ is the input vector. The vector $\boldsymbol{x} \in \mathrm{R}^m$ is the state-variable vector and $\boldsymbol{y} \in \mathrm{R}^p$ is the vector of output variables. This form follows the traditional definition used in most inverse simulation investigations [22]. In addition, Eqs. (2) and (3) can be discretized as:

$$\frac{\boldsymbol{x}(t_k) - \boldsymbol{x}(t_{k-1})}{t_k - t_{k-1}} = \boldsymbol{f}[\boldsymbol{x}(t_k), \boldsymbol{u}(t_k)] \tag{4}$$

$$\boldsymbol{y}(t_k) = \boldsymbol{g}[\boldsymbol{x}(t_k), \boldsymbol{u}(t_k)] \tag{5}$$

$$\text{both for } \quad k = 1, 2, 3, \ldots\ldots N-1$$

where N is the total number of discretized intervals and t_k is the k_{th} discretization point in the time period. Now define two functions $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$ to calculate the values of the unknown variables $\boldsymbol{x}(t_k)$ and $\boldsymbol{u}(t_k)$.

$$\boldsymbol{F}_1[\boldsymbol{x}(t_k), \boldsymbol{u}(t_k)] = \boldsymbol{f}[\boldsymbol{x}(t_k), \boldsymbol{u}(t_k)] - \frac{\boldsymbol{x}(t_k) - \boldsymbol{x}(t_{k-1})}{t_{k-1} - t_k} \tag{6}$$

$$\boldsymbol{F}_2[\boldsymbol{x}(t_k), \boldsymbol{u}(t_k)] = \boldsymbol{g}[\boldsymbol{x}(t_k), \boldsymbol{u}(t_k)] - \boldsymbol{y}_d(t_k) \tag{7}$$

where the term on the left-hand side of Eq. (5) is replaced by ideal output values $\boldsymbol{y}_d(t_k+1)$, and where the subscript d is used to represent the desired value. The Newton-Raphson (NR) method is applied to solve these last two equations so that the values $\boldsymbol{x}(t_k)$ and $\boldsymbol{u}(t_k)$ make the right hand sides of these equations approximately equal to zero. The updated equations (Eq. (8)) are shown as follows:

$$\begin{bmatrix} \boldsymbol{x}^{(n)}(t_k) \\ \boldsymbol{u}^{(n)}(t_k) \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}^{(n\text{-}1)}(t_k) \\ \boldsymbol{u}^{(n\text{-}1)}(t_k) \end{bmatrix} - \begin{bmatrix} \dfrac{\partial \boldsymbol{F}_1}{\partial \boldsymbol{x}} & \dfrac{\partial \boldsymbol{F}_1}{\partial \boldsymbol{u}} \\ \dfrac{\partial \boldsymbol{F}_2}{\partial \boldsymbol{x}} & \dfrac{\partial \boldsymbol{F}_2}{\partial \boldsymbol{x}} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{F}_1[\boldsymbol{x}^{(n\text{-}1)}(t_k), \boldsymbol{u}^{(n\text{-}1)}(t_k)] \\ \boldsymbol{F}_2[\boldsymbol{x}^{(n\text{-}1)}(t_k), \boldsymbol{u}^{(n\text{-}1)}(t_k)] \end{bmatrix} \tag{8}$$

where the quantity n is the current step within the iterative process. After the values $\boldsymbol{x}(t_k)$ and $\boldsymbol{u}(t_k)$ that make $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$ zero are found, the inverse simulation will move to the next time step t_{k+1}. By similar sequential steps, the complete time histories of $\boldsymbol{x}(t_k)$ and $\boldsymbol{u}(t_k)$ can eventually be obtained.

3.2 The integration-based approach

This section provides a summary of the integration-based method of Hess *et al.* [22]. By discretization of Eqs. (2) and (3), the input-output relationship of the nonlinear system can be defined as follows:

$$\boldsymbol{x}(t_{k+1}) = \int_{t_k}^{t_{k+1}} \dot{x}(t)dt + \boldsymbol{x}(t_k) \tag{9}$$

$$\boldsymbol{y}(t_{k+1}) = \boldsymbol{g}[\boldsymbol{x}(t_{k+1}), \boldsymbol{u}(t_k)] \tag{10}$$

The term on the left-hand side of Eq. (10) is now replaced by the ideal output values $\boldsymbol{y}_d(t_{k+1})$, where the subscript *d* is used to represent the desired value. Thus, Eq. (10) can be rewritten as:

$$\boldsymbol{g}[\boldsymbol{x}(t_{k+1}),\boldsymbol{u}(t_k)]-\boldsymbol{y}_d(t_{k+1})=0 \qquad (11)$$

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

$$\boldsymbol{u}^{(n+1)}(t_k)=\boldsymbol{u}^{(n)}(t_k)-\boldsymbol{J}^{-1}\boldsymbol{f}_E[\boldsymbol{x}^{(n)}(t_{k+1}),\boldsymbol{u}^{(n)}(t_k)] \qquad (12)$$

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. (12) the term $\boldsymbol{J}$ 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.

If, in Eq. (3), 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:

$$\boldsymbol{J}_{ij}=\frac{\partial \boldsymbol{y}_i(t_{k+1})}{\partial \boldsymbol{u}_j(t_k)}\approx\frac{\partial \boldsymbol{y}_i(\boldsymbol{u}_j+\Delta\boldsymbol{u}_j)\,|_{t_{k+1}}-\partial \boldsymbol{y}_i(\boldsymbol{u}_j)\,|_{t_{k+1}}}{\Delta\boldsymbol{u}_j} \qquad (13)$$

for $i = 1, 2, 3, \ldots, p$ and $j = 1, 2, 3, \ldots, q$

where $\boldsymbol{u}_j$ and $\boldsymbol{y}_i$ are the j_{th} and i_{th} elements of the input and output vectors, respectively. $\Delta\boldsymbol{u}_j$ is the perturbation in $\boldsymbol{u}_j$ at time t_k. In Eq. (13), the superscript *n* is omitted. Rutherford and Thomson [20] have also 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 methods of matrix inversion in the NR iteration scheme, as shown in Eq. (12). Hess *et al.* [22] proposed the use of the pseudo-inverse matrix as a solution for finding the roots of Eq. (11) when $\boldsymbol{J}$ is rectangular.

3.3 The constrained Nelder-Mead method

Both the differentiation and integration-based methods introduce additional derivative calculations, such as those associated with the Jacobian matrix or Hessian matrix. However, the direct gradient information is not always available from the model. Direct search methods, being derivative free and thus avoiding issues associated with discontinuity and input saturation that cause problems with the differentiation and integration-based methods, provide an alternative approach that has been shown to have advantages for some types of application. This approach is presented here in more detail than the other methods because there are fewer published sources of information currently available about this approach.

Lewis *et al.* [47] have reviewed the history and development of direct search methods of optimisation and point out that they remain popular because of their simplicity, flexibility, and reliability. Among direct search methods, the most widely used is the downhill simplex method of Nelder and Mead [48]. The Nelder-Mead (NM) approach is a popular method for minimizing a scalar-valued nonlinear function of *q* real variables using only function values, without any derivative information (explicit or implicit). The latest developments of this method (e.g. [49-51]) have expanded its functions so that it can be used to tackle multimodal, discontinuous, and constrained optimization problems. However, these developments inevitably make the algorithm more complex. The algorithm outlined in this section is based on the version of Lagarias *et al.*, [52] with an additional input-constrained function [53].

As with the NR method, the NM approach is developed in the interval $[t_k, t_{k+1}]$. One of the distinct differences between the NR and NM methods is that the former updates the input values by means of Eq. (12), but the latter relies exclusively on values of the cost function to find the optimal solution [47]. Hence, it is important for the NM method to define a good form of the cost function, which may be described by equations of the form:

$$\begin{cases}\min\limits_{\boldsymbol{u}\in\mathrm{R}^q} L[\boldsymbol{u}(t_k)], \text{ where} \\ L[\boldsymbol{u}(t_k)]=\sum\limits_{i=1}^{p}\{\boldsymbol{g}_i[\boldsymbol{u}(t_k),\boldsymbol{x}(t_{k+1})]-\boldsymbol{y}_{d_i}(t_{k+1})\}^2\end{cases} \qquad (14)$$

subject to

$$\begin{cases}\boldsymbol{u}_{min,j}\le\boldsymbol{u}_j(t_k)\le\boldsymbol{u}_{max,j} \qquad j=1,2,\ldots,q \\ \dot{\boldsymbol{x}}=\boldsymbol{f}[\boldsymbol{x}(t_k),\boldsymbol{u}(t_k)]\end{cases} \qquad (15)$$

where $L[\cdot]$ is the cost function. If the NM algorithm fails for the quadratic cost-function form of Eq. (14), the following equation based on the absolute value can provide an alternative:

$$\begin{cases}\min\limits_{\boldsymbol{u}\in\mathrm{R}^q} L[\boldsymbol{u}(t_k)], \text{ where} \\ L[\boldsymbol{u}(t_k)]=\sum\limits_{i=1}^{p}\left|\boldsymbol{g}_i[\boldsymbol{u}(t_k),\boldsymbol{x}(t_{k+1})]-\boldsymbol{y}_{d_i}(t_{k+1})\right|\end{cases} \qquad (16)$$

It has been found that the problem can best be approached by using the structure of the integration-based approach so that the process to find solutions is divided into two sub-processes: one-forward simulation to obtain $\boldsymbol{x}(t_{k+1})$ and then calculation of the solution $\boldsymbol{u}(t_k)$ from Eq. (14) or Eq. (16) with the available values $\boldsymbol{x}(t_{k+1})$. In the case of the inverse simulation application it is assumed that only the input saturation conditions are of interest. Hence, for the method outlined here, the inequalities in Eq. (15) are solved by two-step transformations before the solution process of Eqs. (14-16), as shown in the following [53]:

Step 1- transformation of input constraints:

The purpose of this step is to transform the original domain of the input variables into a new space before searching for the solution. The unconstrained input variables will be left alone. If an input variable is constrained by only a lower or an upper bound, a quadratic transformation will be performed by means of the following equations:

$$\begin{cases} \text{if } \boldsymbol{u}_j(t_k) \le \boldsymbol{u}_{min,j} \text{ or } \boldsymbol{u}_j(t_k) \ge \boldsymbol{u}_{max,j} \\ \boldsymbol{u}_{a,j} = 0, \text{ otherwise} \\ \boldsymbol{u}_{a,j} = \sqrt{\boldsymbol{u}_j(t_k) - \boldsymbol{u}_{min,j}} \text{ or } \boldsymbol{u}_{a,j} = \sqrt{\boldsymbol{u}_{max,j} - \boldsymbol{u}_j(t_k)} \end{cases} \tag{17}$$

where $\boldsymbol{u}_a$ is the transformed input vector. If both the lower and upper bounds are required, a *sin* transformation can be defined as follows:

$$\begin{cases} \text{if } \boldsymbol{u}_j(t_k) \le \boldsymbol{u}_{min,j} \text{ or } \boldsymbol{u}_j(t_k) \ge \boldsymbol{u}_{max,j} \\ \boldsymbol{u}_{a,j} = -\pi/2 \text{ or } \boldsymbol{u}_{a,j} = \pi/2, \text{ respectively, otherwise} \\ \boldsymbol{u}_{a,j} = 2 \cdot \pi + \arcsin[-1, (1, 2 \cdot \dfrac{\boldsymbol{u}_j(t_k) - \boldsymbol{u}_{min,j}}{\boldsymbol{u}_{max,j} - \boldsymbol{u}_{min,j}} - 1)_{min}]_{max} \end{cases} \tag{18}$$

where the added term 2π is introduced to avoid problems at zero in the NM algorithm. If this is not done the initial simplex is vanishingly small.

Step 2- solution finding by means of the NM algorithm:

This step is used to transform the new input domain back into the original domain. However, this domain has been constrained before the each evaluation of the cost function or after the solution is finally found. Thus, based on the values transformed from Eq. (17) and Eq. (18), the actual input values for the NM algorithm have to be obtained by application of a second transformation.

In the approach adopted [53], the unconstrained input variables remain unchanged. For the input variables that are constrained by only a lower or an upper bound, the transformation is applied as follows:

$$\begin{cases} \text{For the lower bound}: \boldsymbol{u}_{b,j} = \boldsymbol{u}_{min,j} + \boldsymbol{u}_{a,j}^2(t_k) \\ \text{For the upper bound}: \boldsymbol{u}_{b,j} = \boldsymbol{u}_{max,j} - \boldsymbol{u}_{a,j}^2(t_k) \end{cases} \tag{19}$$

where $\boldsymbol{u}_a$ is the transformed or finally calculated input values. If both the lower and upper bounds are required, a *sin* transformation can be defined as follows:

$$\begin{cases} \text{For the lower and upper bounds}: \\ \boldsymbol{u}_{b,j} = \dfrac{1}{2} \cdot \{\sin[\boldsymbol{u}_{a,j}(t_k)] + 1\} \cdot (\boldsymbol{u}_{max,j} - \boldsymbol{u}_{min,j}) + \boldsymbol{u}_{min,j} \end{cases} \tag{20}$$

Thus the constrained conditions in the cost function may be been handled successfully by the above two steps. As a result, the transformed values $\boldsymbol{u}_{b,j}$ are bounded for the NM algorithm. The final solutions from the NM algorithm have to be transformed back to the original domain by the second step. The above whole process can be illustrated by the following flow chart:

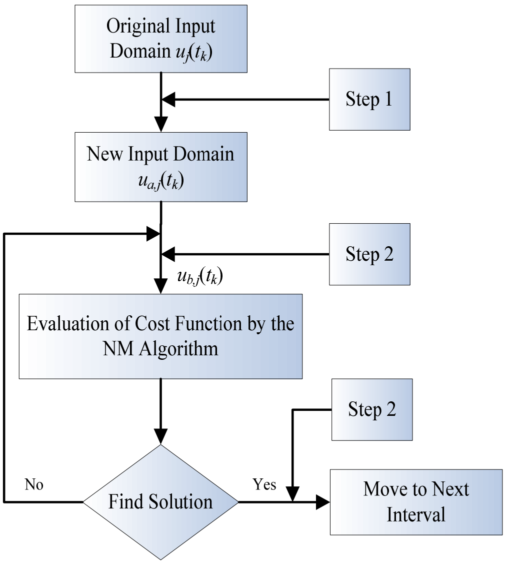
Original Input Domain $u_j(t_k)$
Step 1
New Input Domain $u_{a,j}(t_k)$
Step 2
$u_{b,j}(t_k)$
Evaluation of Cost Function by the NM Algorithm
Step 2
No
Find Solution
Yes
Move to Next Interval

Fig. 1 Flow chart for the constrained NM algorithm for the k_{th} interval of the inverse simulation.

The operation of the modified NM algorithm [52] which forms the basis of this approach can be summarised in the following way. This algorithm first characterises a simplex in q-dimensional space by q+1 distinct vertices. Then, based on four rules that involve processes of reflection (ρ), expansion (χ), contraction (γ) and shrinkage (σ), a new point in or near the current simplex is generated at each step of the search. Afterwards, a new simplex can be constructed by replacing a vertex in the old simplex, after the function value at the new point is compared with the function's values at the vertices of the old simplex. This process is repeated until the diameter of the simplex is less than the specified tolerance. The solutions are thus found for the step under consideration. If each step converges successfully, the complete input time histories $\boldsymbol{u}(t)$ will be formed from the solutions obtained at each step.

The values of the four important coefficients: ρ, x, γ, and σ used are those recommended by Lagarias *et al.* [52]. These are also almost universal choices for the standard NM algorithm and are

$$\rho = 1 \quad \chi = 2 \quad \gamma = 0.5 \quad \sigma = 0.5$$

The initial guess values for $\boldsymbol{u}_{k+1,0}$ are the calculated values $\boldsymbol{u}_k$ from the previous step. Thus, if the manoeuvre is smooth and continuous, this could be a good starting point. Even if discontinuous points are included in this interval, the probability is still high that the NM algorithm can find a global solution because of the reduced vector space involved in the search (from $\boldsymbol{u}_k$ to $\boldsymbol{u}_{k+1}$) instead of the whole space $\boldsymbol{u}(t)$. Hence, the problem of discontinuous points in the manoeuvre may be avoided.

4 Discretization processes in inverse simulation

Analysis of the inverse simulation process for the case of a nonlinear system of the type shown in Eqs. (2) and (3) is difficult. Therefore, it is more appropriate to first consider a linear system having the form shown in Eq. (21).

$$\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{Ax} + \boldsymbol{Bu}\\ \boldsymbol{y} &= \boldsymbol{Cx} + \boldsymbol{Du}\end{aligned} \tag{21}$$

where $\boldsymbol{x}$ is the vector of system state variables, $\boldsymbol{u}$ is the input vector, $\boldsymbol{y}$ is the output vector, and the matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ are the system matrices with the appropriate dimensions.

The inverse simulation procedure based on the integration process may be divided into two stages: first the discretisation process and then the solution by means of the appropriate numerical algorithms. This division is useful because many other inverse simulation methodologies also involve a two-stage approach, using other numerical algorithms instead of the NR approach (e.g. [24-26], [28], [39]). The stability of the second stage usually relates to the numerical stability and convergence properties of the chosen algorithm itself. This involves numerical issues more than questions of dynamical stability. As a result, only the discussion of the first stage is presented and for the second stage convergence is assumed to be achievable.

After discretizing Eq. (21), the following formulae can be obtained:

$$\begin{aligned}\boldsymbol{x}(t_{k+1}) &= \boldsymbol{Px}(t_k) + \boldsymbol{Hu}(t_k)\\ \boldsymbol{y}(t_k) &= \boldsymbol{Cx}(t_k) + \boldsymbol{Du}(t_k)\end{aligned} \tag{22}$$

where the terms $\boldsymbol{P}$ and $\boldsymbol{H}$ are:

$$\begin{aligned}\boldsymbol{P} &= e^{\boldsymbol{A}\Delta t}\\ \boldsymbol{H} &= (\int_0^{\Delta t} e^{\boldsymbol{A}t}\boldsymbol{d}t)\boldsymbol{B}\end{aligned} \tag{23}$$

For the integration-based method the state variables are first updated using the fourth-order *Runge-Kutta* (RK) algorithm. If the RK algorithm is applied for the integration process of the right side of Eq. (23), Eq. (9) can be expressed by the following equation after transformation and simplification:

$$\boldsymbol{x}(t_{k+1}) = \boldsymbol{Q}(\boldsymbol{A}, M, \Delta t)\boldsymbol{x}(t_k) + \boldsymbol{W}(\boldsymbol{A}, \boldsymbol{B}, M, \Delta t)\boldsymbol{u}(t_k) \tag{24}$$

where the variable M is the number of iterative RK steps for one integration step from t_k to t_{k+1}. The function $\boldsymbol{Q}$ is dependent on the algebraic relationship of the three variables $\boldsymbol{A}$, M, and Δt. The function $\boldsymbol{W}$ also depends on the matrix $\boldsymbol{B}$ in addition to the three other quantities shown. When the value M is increased, the accuracy of the results from Eq. (24) will be improved at the cost of greatly increased complexity. If M tends to infinity, Eq. (24) will be identical to Eq. (22). It can thus be concluded that the inverse simulation approximates to the process of discretization and the accuracy of this approximation depends on the value of M. In addition, the zeros of the system in Eq. (22) can be relocated in the z-plane by varying the sampling rate Δt. In the practical inverse simulation process, the values $\boldsymbol{A}$, $\boldsymbol{B}$, and M in Eq. (24) are usually fixed. Hence, by changing the value Δt in Eq. (24), it may be possible to redistribute the zeros in the z-plane in Eq. (22) and to avoid the NMP problem.

The application of the new method to the NMP problem can be explained as follows. Assume first that the system shown in Eq. (22) is a NMP system and has right half plane (RHP) zeros, regardless of the distribution of poles. This process of disregarding the poles is possible because only the RHP zeros will affect the dynamic stability of the inverse system. As mentioned above, by changing the value of Δt it is possible to move zeros originally in the RHP into the left-half plane (LHP). This can guarantee the stability of the inverse simulation process in terms of the system structure at the first stage. Hence, there may exist some sampling-rate intervals or critical Δt_c values where the magnitudes of all the zeros are less than one.

Moreover, even if some magnitudes are greater than unity, inverse simulation may still provide good convergence because of the fact that it approximates to but is not exactly the same as a traditional discretization process. However, it is difficult to obtain Δt_c directly from Eq. (24) due to the complicated structures of the two functions $\boldsymbol{Q}$ and $\boldsymbol{W}$. Furthermore, this complexity is greatly increased when the value of M is increased. In practical terms, Δt_c can be obtained from Eq. (22) by plotting a diagram showing the distribution of magnitudes of zeros versus the sampling-rate variation. These values of Δt_c can then be taken as the reference Δt values for Eq. (24). As M tends to infinity, values obtained from Eq. (22) should be quite close to those obtained from Eq. (24).

The analysis presented in this section is different from that given by Yip and Leng [40]. They addressed the stability analysis using an assumption of fast convergence of the NR method instead of the two-stage division. This assumption of fast convergence may not be appropriate for cases where the inverse simulation does converge but at a relatively slow rate. Moreover, their assumption is made for the case of small Δt values. However, it is well known that small Δt values will lead to some numerical instabilities such as the high-frequency oscillations discussed previously [38]. Secondly, in practice, the assumption of the constant Jacobian matrix, or the existence of the direct analytic relationship between input and out, may not be satisfied for many situations [22, 23]. This assumption of small Δt values can be avoided entirely in the new approach described here. Thirdly, the two methods are based on different standpoints in terms of investigation of the stability of the inverse simulation process. The Yip and Leng approach [40] mainly focuses on the approximation of the NR algorithm. In contrast, the approach presented in this section is concerned more with the first stage- the discretization process.

5 Applications

The inverse simulation approach is illustrated here through a number of different applications to allow a more detailed description and demonstration of the methodology.

5.1 A nonlinear minimum-phase system

The simulation study selected here relates to a nonlinear longitudinal mathematical model of a fixed-wing aircraft, the HS125 (Hawker 800) business jet [54]. It can be shown that the linearised model for this aircraft around the chosen equilibrium point is a minimum phase (MP) system since there are no right-half-plane (RHP) zeros for this model. The thrust T (N) and the elevator angle δ_e (deg) act as the inputs for implementation of the algorithm for inverse simulation involving the NR approach. The manoeuvre conducted is a constant forward-speed hurdle-hop manoeuvre [20] in the z-x plane (altitude versus distance travelled). It may be characterised by the following polynomials:

$$Z_d(t) = 64h\left[\left(\frac{t}{t_m}\right)^3 - 3\left(\frac{t}{t_m}\right)^2 + 3\left(\frac{t}{t_m}\right) - 1\right]\left(\frac{t}{t_m}\right)^3 \mathrm{m} \qquad (25)$$

$$V_{fd}(t) = 61.87\mathrm{m}\cdot\mathrm{s}^{-1}$$

where t_m is the time to complete the manoeuvre and h is the height. This equation also shows that the total flight speed V_f remains constant during the manoeuvre.

In this application the first priority is to define the calculated manoeuvre based on the vector relative degree, if it exists. Calculations show that the model has a vector relative degree [2, 1]. This means that the manoeuvre must be defined in terms of acceleration for application of the model inversion approach. To guarantee a fair comparison, the ideal manoeuvre is also defined as the acceleration in the inverse simulation, although it is not essential in this case. This is one of the advantages of implementation of inverse simulation to derive the require inputs.

The simulation results are generated for the conditions defined below and are shown in Figs. 2 and 3.

$$h = 150 \text{ m}; \quad s = 500 \text{ m}$$

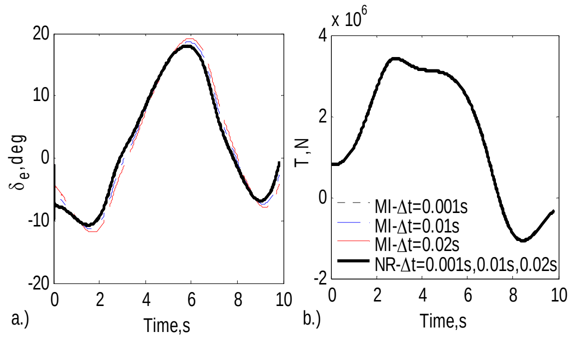

Fig. 2: Inputs from inverse simulation (NR) and model inversion (MI) for the HS125 model for different Δt values.

Figs. 2 and 3 show that for this case inverse simulation shows more accurate results compared with model inversion for the larger Δt values such as 0.01s and 0.02s. In Fig. 2, both methods obtain the same thrust (T) for all Δt values being investigated. However, for the elevator angle (δ_e) channel (Fig. 2a), there are differences between the results for Δt =0.01s and Δt =0.02s. The results from the forward simulation with these calculated inputs, as shown in Fig. 3, further illustrates the poor consistency of the model inversion for Δt =0.01s and Δt =0.02s. This approach only achieves good results for Δt =0.001s. However, use of this smaller Δt value means increased computation time.

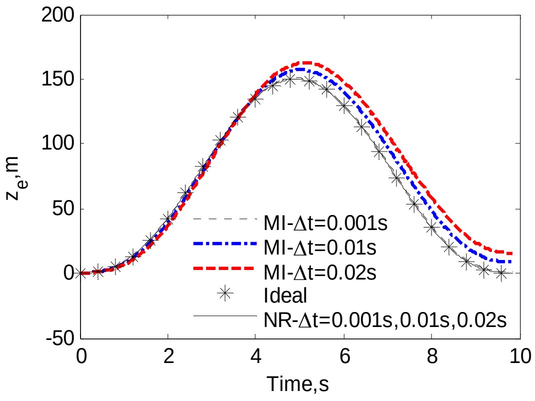

Fig. 3: Comparisons of outputs from forward simulation for the ideal manoeuvre for the HS125 model

In addition to the increased accuracy for this case, compared with the model inversion techniques, inverse simulation is easier and more feasible in terms of implementation. Properties of the algorithm mean that there are no demands on the system in terms of the vector relative degree. It is therefore suggested that for MP systems, particularly for applications where the model is quite complex, such as in a helicopter or ship model, it would be more convenient to adopt inverse simulation, by selecting a suitable sampling interval, rather than apply model inversion. The chosen Δ*t* value should satisfy two important conditions: a.) to guarantee the convergence of the inverse simulation process; and b.) to ensure that the zeros in Eq. (24) remain within the LHP in the discretization process. These two requirements follow the property of the two-stage-division analysis, as already mentioned. The latter requirement must be included because inverse simulation approximates to the discretization process.

5.2 A linear SISO nonminimum-phase system

Consider a linear SISO NMP system given by the following four system matrices:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -11 & -6 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ -7 \\ 81 \end{bmatrix} \tag{26}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \qquad D = [0]$$

This system has two RHP zeros: 0.5000 ± 7.0534*i*. Obviously, the method of Devasia *et al.* [7], can be applied to overcome this NMP problem but it quite tedious and is also a noncausal process. Instead, for implementation of the method outlined in Section 4 above, a plot of the magnitude of the zeros versus Δ*t* values is created, as shown in Fig. 4.

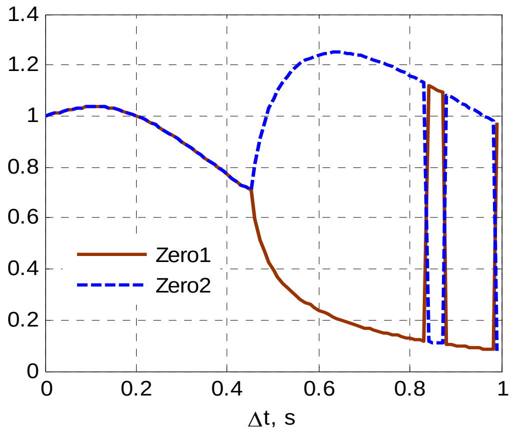

Fig. 4 Variation of magnitude of the zeros with Δ*t*

The interval over which the magnitudes of the zeros are smaller than one, can be determined directly from examination of Fig. 4. In this case the interval is [0.2s, 0.46s]. For the interval [0, 0.2s], there are clearly zeros with magnitude slightly larger than one and for the range above 0.46s magnitudes again become greater than unit as the interval increases. The range of intervals that should be considered first for Δ*t* in the inverse simulation algorithm is therefore [0.2s, 0.46s]. However, it should be noted that the interval [0.0.2s] may not necessarily be invalid and a trial and error process may be used to check whether or not it is usable. According to the analysis presented above, the inverse simulation approximates to but is not exactly the same as a traditional discretization process and the interval [0, 0.46s] therefore may be considered for the process of Δ*t* value selection. Simulation results support the fact that the point 0.46s is a critical limit for convergence of the inverse simulation. However, in addition to the reasons relating to Fig. 4, convergence problems may also be linked to numerical limitations of the NR method implemented in the inverse simulation algorithm. Therefore, the critical point value 0.46s is a combination of effects from the discretization process and from the NR algorithm. The results with Δ*t* values in the selected interval are shown in Fig. 5 for the hurdle-hop manoeuvre.

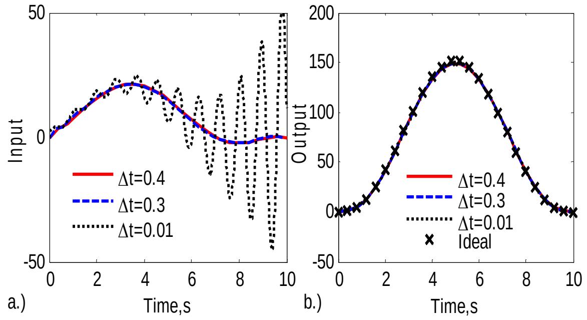

Fig. 5 Comparisons of results from inverse simulation with the different Δ*t* values

Fig. 5a shows that for the sampling rates 0.4s and 0.3s, the inverse simulation achieved perfectly bounded inputs. However, for Δ*t* =0.01s the calculated input is combined with slowly increasing oscillations. This is consistent with the above analysis that states that the results obtained for values outside the interval [0.2s, 0.46s] are of lower quality and even invalid compared with Δ*t* =0.4s and Δ*t* =0.3s. Furthermore, the poorer results of this case conflict with the traditional idea that smaller Δ*t* values in discretization will lead to more accurate results [22].

An interesting phenomenon shown in Fig. 5b is that the results from the forward simulation with the three different calculated inputs completely satisfy the requirements for the ideal trajectory. This actually shows a multi-solution phenomenon with regard to the selection of the different Δ*t* values for a NMP system. Therefore, special attention should be paid to deal with the selection of a suitable Δ*t* value for a NMP system since this is a special case. All in all, this example demonstrates the validity of earlier statements concerning the application of inverse simulation to NMP systems.

5.3 A linear MIMO nonminimum-phase system

In this example, a helicopter model is implemented in terms of an eighth-order description representative of a combat helicopter similar to the Westland Lynx, linearised around the hover situation. The model has the standard state space form. Its state variable vector $\boldsymbol{x}$ contains the following system variables :

State Variables	Description	Units
θ	Pitch attitude	rad
ϕ	Roll attitude	rad
p	Roll rate	$\text{rad}\cdot\text{s}^{-1}$
q	Pitch rate	$\text{rad}\cdot\text{s}^{-1}$
r	Yaw rate	$\text{rad}\cdot\text{s}^{-1}$
U	Forward velocity	$\text{ft}\cdot\text{s}^{-1}$
V	Lateral velocity	$\text{ft}\cdot\text{s}^{-1}$
W	Vertical velocity	$\text{ft}\cdot\text{s}^{-1}$

Table 1: State variables for the Westland Lynx linearised helicopter model

The four channels of heave velocity ($\dot{H}$), roll rate (p), pitch rate (q), and heading rate ($\dot{\Psi}$) are selected to be the outputs. The inputs are the four basic control channels (i.e. main rotor collective pitch (θ_0), main rotor longitudinal cyclic pitch (θ_{ls}), lateral cyclic pitch (θ_{lc}), and tail rotor collective pitch (θ_{tr})). The desired manoeuvres of these four channels are taken from the standard heave axis response (e.g. [55]) and redefined based on the latest version of the US Army helicopter handling qualities requirements ADS-33E-PRF [56]. The desired vertical rate response is thus defined as having the qualitative appearance of a first-order lag with an additional pure delay, as shown in Eq. (27). The other three channels p, q, and $\dot{\Psi}$ are set to be zero in terms of their desired responses.

$$\dot{H}(s) = \frac{10}{0.8225\cdot s+1}e^{-0.162\cdot s} \qquad (27)$$

It can be shown easily that this Lynx-like model, for the chosen flight condition, has vector relative degree $\boldsymbol{r} = \begin{bmatrix}1 & 1 & 1 & 1\end{bmatrix}$. Thus, the inverse simulation is carried out using the first-order derivative of the variables for each channel for the chosen manoeuvre to get a more accurate Jacobian matrix by avoiding the traditional approximation method. The calculation to determine the zeros of the model has shown that the system has two RHP zeros and therefore is a NMP system.

As previously explained, the first step is to plot the magnitude of the zeros in the z-plane versus the sampling rate Δt. After being discretized, the system has more than two RHP zeros. The results in terms of the magnitude plot are shown in Fig. 6.

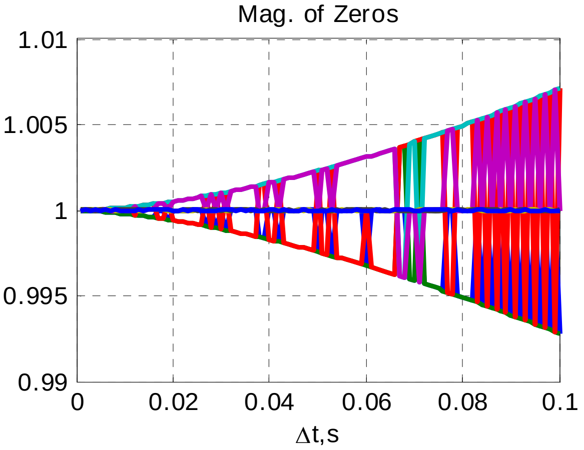

Fig. 6 Magnitude variation of zeros with respect to the sampling interval Δt

Fig. 6 shows the distribution of the magnitudes of the eight zeros of the discretized system of the Westland Lynx-like linearised model. The figure of eight zeros is determined from a series of discretization processes within the interval [0, 0.1s]. In addition, it can be seen from Fig. 6 that there always exist zeros whose magnitudes are larger than unity. Moreover, when the sampling time is increased, the magnitudes of the RHP zeros become larger. According to the previous suggestion, this means that to assure the convergence of the inverse simulation, small sampling intervals are preferred. Besides, the convergence of the NR algorithm needs to be taken into consideration. The final simulations have shown that inverse simulation can achieve convergence only when the Δt value is less than 0.05s. Thus $\Delta t = 0.01s$ is selected to ensure satisfaction of the combined requirements of accuracy, numerical stability and good convergence. The results from the simulation experiments based on this choice of Δt are shown in Figures 7 and 8.

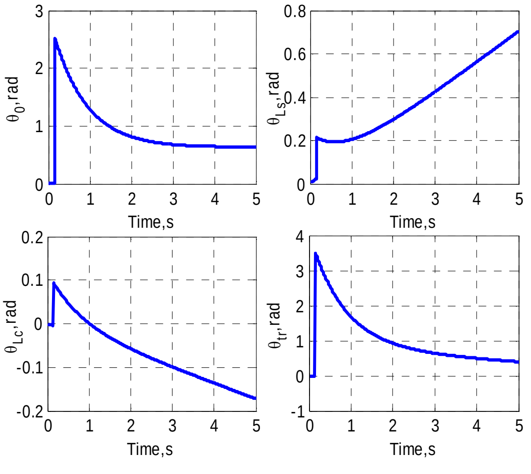

Fig. 7 The calculated inputs from inverse simulation ($\Delta t = 0.01$s)

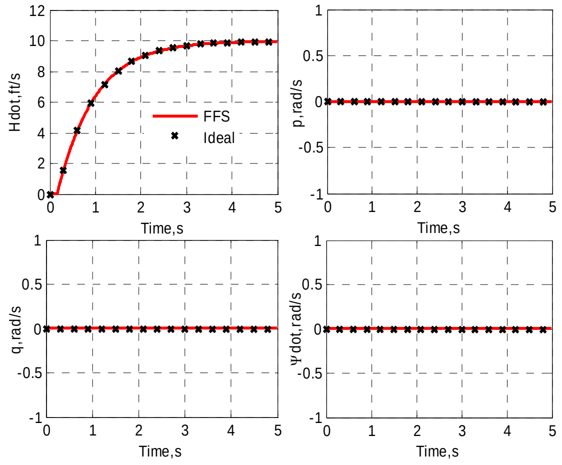

Fig. 8 Comparisons of the calculated outputs with the ideal manoeuvres ($\Delta t = 0.01$s)

Figure 7 shows the inputs obtained from the inverse simulation. The amplitudes of these inputs are quite large and may not have physical meaning for the linear system which is being used as a benchmark. The main rotor collective pitch (θ_0) first initiates a step input to produce vertical acceleration and then decreases to a steady value after a while in order to maintain the required heave velocity. Meanwhile, a step input in the tail rotor collective pitch (θ_{tr}) is applied to balance the main rotor effect. Coupling effects can also be observed in the main rotor longitudinal (θ_{ls}) channel and the lateral cyclic pitch (θ_{lc}) channel to make the roll and pitch angles as small as possible. Fig. 8 shows good consistency between the ideal manoeuvres and results obtained from the forward simulation using those calculated inputs. The heave velocity ($\dot{H}$) follows the required step response while the other three channels involving roll rate (p), pitch rate (q), and heading rate ($\dot{\Psi}$) are kept at zero.

These figures show that the inverse simulation can obtain perfect results regardless of the fact that the original system (for Δt =0.01s) has three RHP zeros with magnitudes very close to one. This is consistent with that fact, mentioned previously, that the inverse simulation process can be linked to the traditional discretization process. The latter process provides an analytical method for selecting a sub-optimal sampling interval or a reference interval for the inverse simulation. Trial and error may be involved in this process. Other tests have also been done and the results show that the inverse simulation algorithm can converge well, in this example, for values of Δt below 0.05s. Beyond this critical point, the inverse simulation algorithm cannot converge.

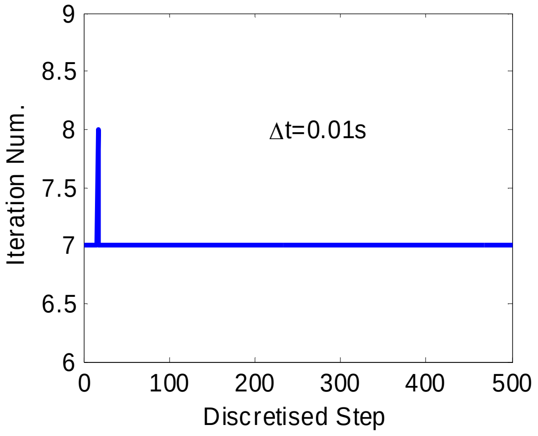

Fig. 9 Iterations required in inverse simulation for each discretized step

The iterative steps required during the inverse simulation process for this example are plotted in Fig. 9 and this shows that the inverse simulation process needs at least seven steps for the NR algorithm to achieve convergence in the interval $[t_k, t_{k+1}]$.

5.4 An application with input saturation and discontinuous manoeuvres

To compare the new NM algorithm with the NR method, one case study has been selected. It relates to the nonlinear Norrbin type of ship model which has been used extensively for ship manoeuvring studies involving both deep and confined waters (e.g. [57]). The structure of the Norrbin ship model can be represented by the following equation:

$$T\ddot{\Psi} + H_N(\dot{\Psi}) = K\delta \qquad (28)$$

where δ and Ψ represents the rudder and heading angles, respectively. K and T are constants and the nonlinear term $H_N(\dot{\Psi})$ is defined as:

$$H_N(\dot{\Psi}) = \alpha_3\dot{\Psi}^3 + \alpha_2\dot{\Psi}^2 + \alpha_1\dot{\Psi} + \alpha_0 \qquad (29)$$

where α_i $(i = 0,1,2,3)$ are called Norrbin's coefficients. For most ships, α_3 = α_2 = 0. Therefore, Eqs. (28) and (29) can be simplified to form Eq. (30):

$$\delta = m\ddot{\Psi} + d_1\dot{\Psi} + d_2\dot{\Psi}^3 \qquad (30)$$

where $m = \dfrac{T}{K}$, $d_1 = \dfrac{\alpha_1}{K}$, and $d_3 = \dfrac{\alpha_3}{K}$. These coefficients will vary according to chosen operating point in terms of the steady forward speed U. The coefficient values correspond to the forward speeds from 1 m/s to 20 m/s can be found in Table 2. Now Eq. (30) can be transformed into a state-space form, as shown in Eq. (31), for facilitating the investigation of inverse simulation.

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= -\frac{d_1}{m}x_2 - \frac{d_3}{m}x_2^3 + \frac{1}{m}\delta \qquad (31)\\ \dot{\delta} &= \frac{1}{\tau}(\delta_c - \delta)\end{aligned}$$

where $x_1 = \Psi$ and τ is the time constant. The third equation in Eq. (31) is related to the steering machine structure, as described in Fig.10, where the rudder and rudder rate limiters are involved in the model.

In this model input saturation is included through limits on the rudder position and the rudder rate is also constrained, as shown in Fig. 10.

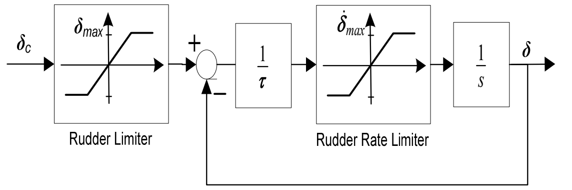

Fig. 10: Diagram illustrating rudder amplitude and rate limit [57].

The value for the time constant τ in Eq. (31) is selected to be 1 s. The three kinds of manoeuvres investigated are the turning circle, a zigzag, and a pullout.

A third-order reference model, as shown in Eq. (32), is used to generate the desired heading response.

$$\frac{\Psi_d}{\Psi_r} = \frac{c_m}{s^3 + a_m s^2 + b_m s + c_m} \qquad (32)$$

where a_m, b_m, and c_m are constants. In the current application, these values are selected as follows [58]:

$a_m = 0.9341,\ b_m = 0.2040$ and $c_m = 0.0182$

This choice of reference model can guarantee sufficient smoothness of the heading acceleration.

U (m/s)	T	K	m	d_1	d_3/K
1	155.0	0.1	1550.0	10.00	100.00
2	77.5	0.2	387.5	5.00	12.50
3	51.7	0.3	172.2	3.33	3.70
4	38.8	0.4	96.9	2.50	1.56
5	31.0	0.5	62.0	2.00	0.80
6	25.8	0.6	43.1	1.67	0.46
7	22.1	0.7	31.6	1.43	0.29
8	19.4	0.8	24.2	1.25	0.19
9	17.2	0.9	19.1	1.11	0.14
10	15.5	1.0	15.5	1.00	0.10
11	14.1	1.1	12.8	0.91	0.07
12	12.9	1.2	10.8	0.83	0.06
13	11.9	1.3	9.2	0.77	0.05
14	11.1	1.4	7.9	0.71	0.04
15	10.3	1.5	6.9	0.67	0.03
16	9.7	1.6	6.1	0.63	0.02
17	9.1	1.7	5.3	0.59	0.02
18	8.6	1.8	4.8	0.56	0.0171
19	8.2	1.9	4.3	0.53	0.0146
20	7.8	2.0	3.9	0.50	0.0125

Table 2: Values of parameters of Norrbin model of ROV Zeefakkel for variation of forward speed U [58]

Inverse simulation has been carried out for a Norrbin type model of the ROV Zeefakkel (RZ) ship [58] with the forward speed U = 10m/s and the set heading angles are 20 deg and 50 deg. The quality of the results from inverse simulation has been validated by the feedforward simulation (FFS) with the calculated inputs. This procedure can be described by the following diagram:

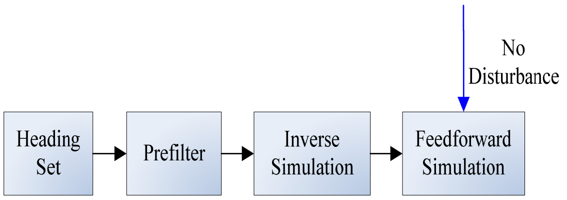

Fig. 11: Validation of the inverse simulation process

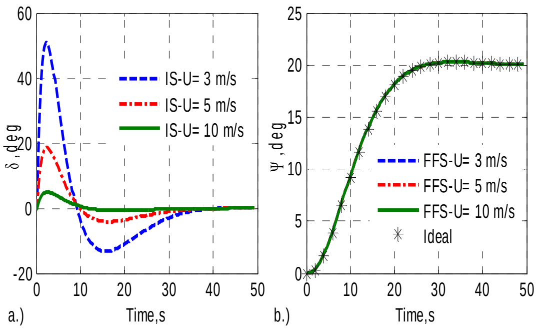

Fig. 12: Inverse simulation of the RZ ship without saturation limits (Δ*t* =0.2 s, NR)

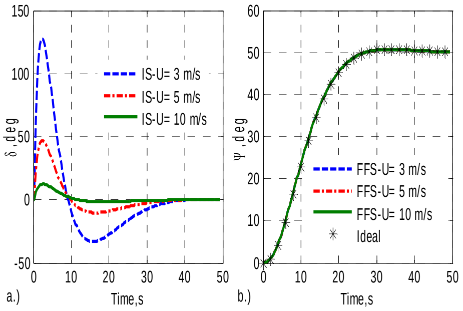

Fig. 13: Inverse simulation of the RZ ship without saturation limits (Δ*t* =0.2 s, NR)

As shown in Fig. 12 and Fig. 13, the results from the inverse simulation and the FFS agree well with each other for both heading angles of 20 and 50 degrees. The right-hand parts of Fig. 12 and Fig. 13 also contain results from three forward speeds ($U = 3, 5, 10$ m/s). These results demonstrate the successful application of inverse simulation to the nonlinear RZ ship model without saturation limits. From these figures, it may also be seen that the rudder angles are beyond their limits for part of the time history for the case where U = 3m/s and also for U = 5m/s for a heading angle of 50 degrees. In addition, a further series of tests of inverse simulation based on both the NR and NM algorithms have been successfully run on the RZ model with the forward speeds varying from 1 m/s to 20 m/s. The results are not presented here since they are similar to those shown in Figs. 12 and 13.

Inverse simulation also has been investigated on the models with the saturation limits included. The rudder limit in this case is 35 degrees and the rudder-rate limit is 7 deg/s. This situation is quite different from the one without the limiters. The inverse simulation based on the NR algorithm fails to converge for the set heading angle of 20 deg if the forward speed U is less than 9 m/s, or for the set heading angle of 50 deg if U is less than 15 m/s. This problem of convergence failure is a well-known feature of the NR algorithm when saturation limits are reached. In this application it arises to a considerable extent from the fact that the inputs or the rate of change of inputs required to track the ideal manoeuvres become larger as the forward speed decreases, as is clearly shown in Fig. 12 and Fig. 13. In addition, the larger the required heading angle the larger will be the required control effort. Therefore, the case of the set heading angle of 50 deg will show problems of convergence failure at a larger value of speed compared with the case of the set heading angle 20 deg.

However, the inverse simulation based the NM algorithm, which was developed to overcome the problem of input saturation, can be used to deal with this kind of situation. Two typical cases with forward speeds 3 m/s and 8 m/s have been investigated. As mentioned above, the NR algorithm failed for both cases. The results from simulations are shown in Fig. 14 and Fig. 15.

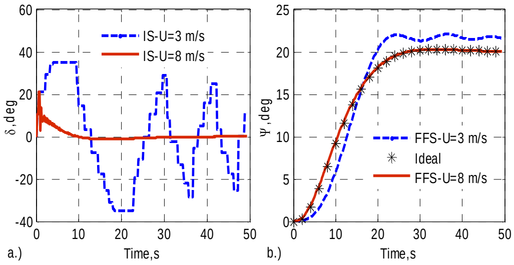

Fig. 14: Inverse simulation of the RZ ship with saturation limits (Δ*t* =0.2 s) using the NM algorithm

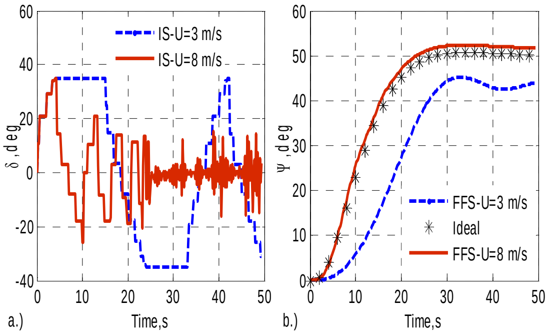

Fig. 15: Inverse simulation of the RZ ship with saturation limits (Δ*t* =0.2 s) using the NM algorithm

Fig. 14 shows that the NM-based approach achieves good convergence. For a heading angle of 20 deg and the forward speed 8 m/s, the required input is within the saturation limits, as shown in Fig. 14a and the trajectory from the FFS also complies well with the ideal manoeuvre in Fig. 14b. However, the rate limit value for the rudder on this vessel is 7 m/s and the rates encountered without the limiting reached 34 m/s. Therefore, it is not the rudder amplitude limiter but the rudder-rate limiter that leads the NR-based approach to fail to converge. For a heading angle 20 deg and the forward speed 3 m/s, the amplitude of the required input reaches the saturation level at two periods around the time points 8 and 20 seconds. These saturations further lead to the slight discrepancy between the results of the FFS and the ideal manoeuvre, as shown in Fig.14b. In addition, the input result shows an oscillatory shape in this case, which is also shown in the results of the FFS.

The situation becomes worse for the case where the heading angle of 50 deg is considered. Fig. 15a shows that the results are affected by high-frequency oscillations after the 22 second time point, although the case for U = 8m/s does converge. However, the smooth results from the FFS comply with the ideal manoeuvre, as shown in Fig. 15b. This may be due to the filtering effect of the model. The case of u = 3m/s is quite challenging since the required inputs are significantly larger. The inverse simulation process is in the saturation state during two long periods, shown in Fig. 15a and this naturally leads to a large discrepancy from the ideal manoeuvre shown in Fig. 15b.

It can be concluded, from the evidence in this example and from other case studies, that the results show that the new method of inverse simulation provides better convergence and numerical stability for cases involving input saturation or discontinuous manoeuvres than traditional inverse simulation methods. However, for severe manoeuvres such as the zigzag and complex models such as the AUV, a multi-solution phenomenon may appear in the results.

It is suggested that the multi-solution phenomenon has potential advantages in dealing with control reallocation and may allow the optimal control effort to be found by modification of the cost-function definition. In addition, the NM method can form a useful reference method that can allow a better understanding of some numerical problems associated with the other commonly used methods.

6 Discussion

This paper has reviewed some aspects of inverse simulation, putting emphasis on a number of methods that have been applied successfully in the past to a range of practical applications, drawn mainly from the aeronautical engineering field. Particular emphasis has been placed on methods of inverse simulation that are based on use of an integration algorithm in conjunction with the Newton-Raphson (NR) technique for determination of the input needed to achieve a specified output. These techniques are widely used and have been applied successfully to a range of problems. Numerical and other limitations have been discussed and a number of different linear and nonlinear examples have been considered.

It has been shown that for a suitable discretization interval, for the case of minimum phase (MP) systems, inverse simulation can provide results that are almost identical to those obtained by model inversion. This was illustrated by the application involving the nonlinear HS125 fixed-wing aircraft model. For linear non-minimum-phase (NMP) systems, the results shown that inverse simulation can be used successfully for causal calculation of inputs. In addition, compared with model inversion, the inverse simulation process is easier and more feasible in terms of practical implementation for such cases.

Analysis has shown that the inverse simulation process can be viewed as depending upon zero redistribution and this provides a useful link between the linear inverse system and its discrete counterpart in a mathematical sense. This has been successfully demonstrated using the example of the eighth-order linear Lynx-like helicopter model. However, the investigation of inverse simulation for the case of nonlinear NMP systems requires further consideration.

A two-stage approach to inverse simulation has been presented which is more general and less restricted than previously implemented methods. It does not require assumptions of a constant Jacobian matrix or fast convergence. While other methods (e.g. [40]) focus on the approximation of the NR algorithm the emphasis in this paper is on the approximation to the discretization process. As a result it can be concluded that the stability of the whole inverse simulation process is affected both by the discretization process and the NR algorithm.

In the control field, it is well known that the performance of a controller may be degraded if the control system designer fails to take account of input saturation effects. These exist in real physical systems due to inevitable limitations of mechanical or electrical sub-systems. However, in the inverse simulation field, few previous investigations have given particular consideration to situations involving saturation constraints or discontinuities in the model and manoeuvres.

In fact, limit effects present a challenge to traditionally established approaches, involving not only the integration-based approaches but also the differentiation-based methods and some optimisation approaches (e.g. [26], [27]). All these techniques involve derivative or gradient information since these approaches depend on continuous and smooth properties of the model and the manoeuvre for inverse simulation.

To avoid the above problems and achieve increased numerical stability with additional physical insight, a new algorithm for inverse simulation based on the constrained Nelder-Mead (NM) method has been outlined. It is well known that the NM algorithm can handle discontinuities satisfactorily, particularly if they do not occur near the optimum solution. Furthermore, the derivative-free property can facilitate investigation of some of the numerical issues that exist in the more traditional inverse simulation methods.

7 Acknowledgements

The authors wish to acknowledge with thanks the many helpful discussions that they have enjoyed with Dr Douglas G. Thomson, Dr David Anderson and Dr

Marat Bagiev of the Department of Aerospace Engineering at the University of Glasgow on the subject of inverse simulation methods and their application to problems of helicopter flight mechanics.

David Murray-Smith acknowledges the financial support of the UK Engineering and Physical Sciences Research Council through grant GR/S91024/01.

Linghai Lu acknowledges the support of the University of Glasgow and the UK ORS programme in terms of a Postgraduate Scholarship and a Studentship respectively.

8 References

[1] R. Jones. A s*implified application of the method of operators to the calculation of disturbed motions of an airplane*, NACA TR560, 1936.

[2] R.W. Brockett. Poles, zeros, and feedback: state space interpretation. *IEEE Transactions on Automatic Control*, 10: 129-135, 1965.

[3] P. Dorato. On the inverse of linear dynamical systems. *IEEE Transactions on Systems Science and Cybernetics*, 5: 43-48, 1969.

[4] R.M. Hirschorn, Invertibility of multivariable nonlinear control system. *Journal of Guidance, Control, and Dynamics*, AC-24: 855-865, 1979.

[5] A. Isidori, Nonlinear Control Systems: An Introduction, *2*nd ed., London: Springer, 1989.

[6] A. Isidori and C.I. Byrnes, Output regulation of nonlinear systems. *IEEE Transactions on Automatic Control*, 35: 3544-8, 1990.

[7] S. Devasia, D. Chen and B. Paden, Nonlinear inversion-based output tracking. *IEEE Transactions on Automatic Control*, 41: 930-942, 1996.

[8] X. Wang and D. Chen, Causal inversion of nonminimum phase system. *Proceedings of the 40th IEEE Conference on Decision and Control*; Dec. 2001, Orlando, FL, USA, 73-8, IEEE 2001.

[9] X. Wang, X. and D. Chen, Output tracking control of nonminimum phase systems via causal inversion. *Proceedings of the 45th Midwest Symposium on Circuits and Systems*; Aug 4-7, Tulsa, OK, USA, 125-8, 2002.

[10] Q. Zou and S. Devasia, Preview-based stable-inversion for output tracking. *Proceedings of the American Control Conference*; Jun 2-4, 1999, San Diego, CA, USA, 3544-8, 1999.

[11] Q. Zou. and S. Devasia, Preview-based inversion of nonlinear nonminimum-phase systems: VTOL example. *Automatica*, 43: 117-127, 2007.

[12] S. Devasia, Output tracking with nonhyperbolic and near nonhyperbolic internal dynamics: helicopter hover control. *Journal of Guidance, Control, and Dynamics*, 20: 573-580, 1997.

[13] S. Devasia, Approximated stable inversion for nonlinear system with nonhyperbolic internal dynamics. *IEEE Transactions on Automatic Control*, 44: 1419-1425, 1999.

[14] J. Reiner, G.J. Balas and W.L. Garrard, Robust dynamic inversion for control of highly maneuverable aircraft. *Journal of Guidance, Control, and Dynamics*, 18: 18-24, 1995.

[15] S.S. Hu, P.H. Yang, J.Y. Juang, and B.C. Chang, Robust nonlinear ship course-keeping control by H I/O linearization and synthesis. *International Journal of Robust and Nonlinear Control*, 13: 55-70, 2003.

[16] S. Sastry, Nonlinear systems: analysis, stability, and control. New York: Spring-Verlag, 1999.

[17] O. Kato and I. Saguira, An interpretation of airplane general motion and control as inverse problem. *Journal of Guidance, Control, and Dynamics*, 9: 198-204, 1986.

[18] D.G. Thomson, *Evaluation of helicopter agility through inverse solution of the equation of motion*. PhD Thesis, Faculty of Engineering, University of Glasgow, Scotland, UK, 1987.

[19] D.G. Thomson, F. Coton, and R. Galbraith, A simulation study of helicopter ship landing procedures incorporating measured flow-field data. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 219: 411-427, 2005.

[20] S. Rutherford and D.G. Thomson, Improved methodology for inverse simulation. *Aeronautical Journal*, 100: 79-86, 1996.

[21] D. Anderson, Modification of a generalised inverse simulation technique for rotorcraft flight. *Proceedings of the Institution of Mechanical Engineers*, *Part G: Journal of Aerospace Engineering*, 217: 61-73, 2003.

[22] R.A. Hess, C. Gao and S.H. Wang, A generalized technique for inverse simulation applied to aircraft maneuvers. *Journal of Guidance, Control and Dynamics*, 14**:** 920-6, 1991.

[23] C. Gao, R.A. Hess, Inverse simulation of large-amplitude aircraft manoeuvres. *Journal of Guidance, Control and Dynamics*, 16: 733-7, 1993.

[24] G. de Matteis, L.M. de Socio and A. Leonessa, Solution of aircraft inverse problems by local optimization. *Journal of Guidance, Control and Dynamics*, 18: 567-571, 1995.

[25] S. Lee and Y. Kim, Time-domain finite element method for inverse problem of aircraft maneuvers. *Journal of Guidance, Control and Dynamics*, 20: 97-103, 1997.

[26] G. Avanzini, G. de Matteis and L.M. de Socio, Two-timescale-integration method for inverse simulation. *Journal of Guidance, Control and Dynamics*, 22: 395-401, 1999.

[27] R. Celi, Optimization-based inverse simulation of a helicopter slalom manoeuvre. *Journal of Guidance, Control and Dynamics*, 23: 289-297, 2000.

[28] L. Lu, D.J. Murray-Smith and D.G. Thomson, A sensitivity-analysis method for inverse simulation. *Journal of Guidance, Control and* Dynamics, 30: 114-121, 2007.

[29] D.G. Thomson and R. Bradley, The use of inverse simulation for conceptual design. *Proceedings of 16th European Rotorcraft Forum*; Sept. 1990, Glasgow, Scotland, UK, 1990.

[30] D.G. Thomson and R. Bradley, The contribution of inverse simulation to the assessment of helicopter handling qualities. *Proceedings of the 19th ICAS Conference*; Sept 18-23, 1994, Anaheim, USA, 229-238, 1994.

[31] R. Bradley and D.G. Thomson, The development and potential of inverse simulation for the quantitative assessment of helicopter handling qualities, piloting vertical flight aircraft. *Proceedings of the AHS/NASA Conference on Piloting Vertical Flight Aircraft: Flying Qualities and Human Factors*; Jan. 1993, San Francisco, USA, 359-371, 1993.

[32] S. Rutherford and D.G. Thomson, Helicopter inverse simulation incorporating an individual blade rotor model. *Journal of Aircraft*, 34: 627-634, 1997.

[33] D.G. Thomson and R. Bradley, The principles and practical application of helicopter inverse simulation. *Simulation Practice and Theory*, 6: 47-70, 1998.

[34] D.G. Thomson, C.D. Taylor, R. Talbot, R. Ablett, and R. Bradley, An investigation of piloting strategies for engine failure during takeoff from offshore platforms. *Aeronautical Journal*, 99: 15-25, 1995.

[35] E. Sentoh and A. Bryson, Inverse and optimal control for desired outputs. *Journal of Guidance, Control and Dynamics*, 15: 687-691, 1992.

[36] G.J. Gray, W. von Grünhagen, 1998. An investigation of open loop and inverse simulation as nonlinear model validation tools for helicopter flight mechanics. *Mathematical and Computer Modelling of Dynamical Systems*, 4: 32-57, 1998.

[37] D.P. Boyle, G.E. Chamitoff, Autonomous maneuver tracking for self-piloted vehicles. *Journal of Guidance, Control and Dynamics*, 22: 58-67, 1999.

[38] K. Lin, Comment on generalized technique for inverse simulation applied to aircraft maneuvers. *Journal of Guidance, Control and Dynamics*, 16: 1196-7, 1993.

[39] G. Avanzini and G. de Matteis, Two-timescale inverse simulation of a helicopter model. *Journal of Guidance, Control and Dynamics*, 24: 330-9, 2001.

[40] K.M. Yip and G. Leng, Stability analysis for inverse simulation of aircraft. *Aeronautical Journal*, 102: 345-351, 1998.

[41] W. Cheney and D. Kincaid, D., Numerical Mathematics and Computing, 5th ed. USA: Brooks/Cole Publishing Company, 2004.

[42] L. Lu, D.J. Murray-Smith and E.W. McGookin, Feedforward controller design from a constrained derivative-free inverse simulation process. *Control Engineering Practice*. Submitted for Publication, 2007.

[43] J.J. Buchholz and W. von Grünhagen, *Inversion impossible?*, Technical Note, Hochschule Bremen and DLR Braunschweig, September 2004.

[44] G. Looye, M.Thümmel, M. Kurze, M. Otter and J. Bals, Nonlinear inverse models for control. In G. Schmitz (ed.), *Proceedings 4th Intl. Modelica Conference*, Hamburg, March 7-8 2005, 267-279, 2005

[45] D.J. Murray-Smith, The inverse simulation approach: a focused review of methods and applications. *Mathematics and Computers in Simulation*, 53: 239-247, 2000.

[46] D.Thomson and R. Bradley, Inverse simulation as a tool for flight dynamics research – Principles and application, *Progress in Aerospace Sciences*, 42: 174-210, 2006.

[47] R.M. Lewis, V. Torczon and M.W. Trosset, Direct search methods: then and now. *Journal of*

Computational and Applied Mathematics, 124: 191-207, 2000.

[48] J.A. Nelder and R. Mead, A simplex method for function minimization. *Computer Journal*, **7:** 308-313, 1965.

[49] R. Chelouah and P. Siarry, Genetic and Nelder-Mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions. *European Journal of Operational Research*, 148: 335-348, 2003.

[50] M.A. Luersen, R. Le Richem and F. Guyon, A constrained, globalised, and bounded nelder-mead method for engineering optimization. *Structural and Multidisciplinary Optimization,* 27:, 43-54, 2004.

[51] S. Wolff, *A local and globalized, constrained and simple bounded Nelder-Mead method [Ver. 2.0]*. Computer Program. Bauhaus. University Weimar, Germany, 2004.

[52] J.C. Lagarias, J.A. Reeds, M.H. Wright and P.E. Wright, Convergence properties of the Nelder-Mead simplex method in low dimensions. *Siam. J. Optim*., 9: 112-147, 1998.

[53] J.D. Errico, 2005. *Bound constrained optimization*. Computer Program. MATLAB and Simulink Centre, the MathWorks, Inc., 2005

[54] D.G. Thomson, D.G. 2004. *Mathematical Modelling and Simulation of Fixed Wing Aircraft.* Course Notes, Department of Aerospace Engineering, Faculty of Engineering, University of Glasgow, Scotland, UK, 2004.

[55] D.J. Walker and I. Postlethwaite, Advanced helicopter flight control using two-degree-of-freedom H infinity optimization. *Journal of Guidance, Control and Dynamics*, 19: 461-8, 1996.

[56] Anon, *Aeronautical design standard performance specification handling qualities requirements for military rotorcraft ADS-33E-PRF*. US Army Aviation and Missile Command, Aviation Engineering Directorate, Redstone Arsenal, Alabama, USA, 2000.

[57] T.I. Fossen, Guidance and Control of Ocean Vehicles. UK: John Wiley & Sons Ltd., 1994.

[58] M.A. Unar, *Ship steering control using feedforward neural networks.* PhD Thesis, Department of Electronics and Electrical Engineering, Faculty of Engineering, University of Glasgow, UK, 1999.