



Avanzini, G., Thomson, D., and Torasso, A. (2013) Model predictive control architecture for rotorcraft inverse simulation. *Journal of Guidance, Control, and Dynamics*, 36 (1). pp. 207-217. ISSN 0731-5090

Copyright © 2013 American Institute of Aeronautics and Astronautics, Inc

A copy can be downloaded for personal non-commercial research or study, without prior permission or charge

The content must not be changed in any way or reproduced in any format or medium without the formal permission of the copyright holder(s)

When referring to this work, full bibliographic details must be given

<http://eprints.gla.ac.uk/75035>

Deposited on: 11 February 2013

Enlighten – Research publications by members of the University of Glasgow
<http://eprints.gla.ac.uk>

Model Predictive Control Architecture for Rotorcraft Inverse Simulation

Giulio Avanzini¹

Università del Salento, Brindisi, 72100, Italy

Douglas Thomson²

University of Glasgow, Glasgow, G12 8QQ, Scotland UK

Alberto Torasso³

Politecnico di Torino, Turin, 10129, Italy

A novel inverse simulation scheme is proposed for applications to rotorcraft dynamic models. The algorithm adopts an architecture which closely resembles that of a model predictive control scheme, where the controlled plant is represented by a high-order helicopter model. A fast solution of the inverse simulation step is obtained on the basis of a lower-order, simplified model. The resulting control action is then propagated forward in time using the more complex one. The algorithm compensates for discrepancies between the models by updating initial conditions for the inverse simulation step and introducing a simple guidance scheme in the definition of the tracked output variables. The proposed approach allows for the assessment of handling quality potential on the basis of the most sophisticated model while keeping model complexity to a minimum for the computationally more demanding inverse simulation algorithm. The reported results, for an articulated blade, single main rotor helicopter model, demonstrate the validity of the approach.

¹ Professor, Faculty of Industrial Engineering, Edificio 14 Cittadella della Ricerca S.S. 7 Km.7.3, AIAA Senior Member.

² Senior Lecturer, Department of Aerospace Engineering, James Watt South Building, AIAA Senior Member.

³ Ph.D. Candidate, Department of Mechanical and Aerospace Engineering, Corso Duca degli Abruzzi 24, AIAA Member.

Nomenclature

A_{1s}, B_{1s}	lateral and longitudinal cyclic pitch commands
a_y	lateral airframe acceleration
\dot{h}	rate of climb
K	guidance gain
N	number of forward simulation steps in the receding horizon
N_b	number of blades
p, q, r	angular speed components in body frame
T	receding time horizon
t_k	time at the beginning of the k -th inverse simulation step
u, v, w	speed components in body frame
$\mathbf{u}, \mathbf{x}, \mathbf{y}$	control, state and output vectors
V	airspeed
\mathbf{y}_{des}	desired output vector
<i>Greek Symbols</i>	
$\beta_0, \beta_c, \beta_s$	coning, longitudinal and lateral flapping multiblade coordinates
$\tilde{\beta}_0, \tilde{\beta}_{1c}, \tilde{\beta}_{1s}$	first order harmonic flapping coefficients for low-order rotor models
β_i	flap angle of the i -th blade
Δt	forward simulation step
$\Delta \mathbf{y}^*$	desired increment for the tracked outputs
ζ_i	lead-lag angle of the i -th blade
θ_0, θ_{0TR}	main and tail rotor collective pitch command
$\nu_0, \nu_{1c}, \nu_{1s}$	uniform, cosine and sine components of nondimensional rotor inflow velocity
ϕ, θ, ψ	fuselage attitude angles
φ_i	dynamic twist angle of the i -th blade
Ψ	reference blade azimuth angle
χ	heading angle
Ω	rotor angular speed

Subscripts

F at final time

I at initial time

0 at trim

I. Introduction

The scope of this paper is to present a novel method for solving inverse simulation problems for rotorcraft motion, where the control action that successfully tracks a prescribed trajectory is determined on the basis of a low-order simplified model and then implemented on a high order, more accurate one. Helicopter inverse simulation [1] has been an active topic of research since its first development over two decades ago, with the works of Thomson and Bradley [2] and Hess and Gao [3].

Inverse simulation (IS) can be used for systematically evaluating helicopter performance in manoeuvring flight and/or for an extensive analysis of handling qualities (HQ), as suggested in [1]. Once a Manoeuvre Task Element (MTE) is chosen that fulfills a particular HQ requirement, the determination of a feasible control action that successfully tracks the desired trajectory by means of an IS algorithm indicates that the considered MTE lies within the manoeuvring envelope of the vehicle, at least in terms of limits on command travel, available power, maximum shaft torque, etc.

The solution of the inverse problem thus requires the determination of control inputs that allow a helicopter model to fly a specified manoeuvre. A wide plethora of methods for solving IS problems in flight mechanics has been considered in the past for both fixed- and rotary-wing aircraft. These methods can be grouped into three major categories: 1) differential methods [4], suitable for nominal problems only, where the number of control inputs equals that of the tracked variables; 2) integration methods [5], where the required control action is evaluated over a discrete time interval, with the possibility of solving redundant problems (*e.g.* by means of a local optimisation approach [6]); and 3) global methods [7], where the time-history of the control variables is determined over the whole duration of the tracked manoeuvre by means of a variational approach.

As underlined in Ref. [1], the solution of the inverse problem is a task significantly more chal-

lenging for the rotorcraft case than for a conventional airplane, especially when individual blade dynamics are incorporated in the model [8]. Moreover, the issues related to the presence of transmission zeros and non-minimum phase response affect rotorcraft dynamics more seriously than fixed-wing aircraft models [9].

Among other approaches, one of the advantages of integration methods is represented by their capability of dealing with complex, high order mathematical models of the vehicle on the basis of a solution scheme that can be applied with only minor variations to dynamical models of various order and complexity, provided that the issue of unconstrained states is properly addressed. Although computational efficiency can be increased by application of a two-time-scale approach [9], the short-term behaviour of unconstrained states remains a major concern in performing inverse simulation of complex high order models.

Bagiev et al. [10] have shown that a modified scheme that includes a predictive step is able to provide more realistic solutions to the inverse simulation problem. Especially when dealing with aggressive manoeuvres, a baseline IS algorithm may predict values which exceed the physical limits of the real vehicle, such as mechanical limitations on control travel or control rates (based on hydraulic actuator stroke and other characteristics), limits on main and tail rotor torque, total required power or even structural limits of critical components. A predictive step is thus introduced in order to identify in advance if and when such limits are approached during a manoeuvre, so that the control task can be modified and turned into a feasible one. The use of a receding horizon for predicting violations of limits on vehicle performance and control travel paved the way to the derivation of a novel approach for inverse simulation, with an architecture that closely follows the structure of a Model Predictive Control scheme.

The use of Model Predictive Control (MPC) [11] in the aerospace field is not new. Several control techniques based on prediction schemes have been proposed in the literature [12–14] as a possible approach for the design of high performance controllers. The evaluation of the control law results from the solution of a finite horizon open-loop optimal control problem, using the current state of the plant as the initial state. The optimisation yields an optimal control sequence and the first control in this sequence is applied to the plant for a discrete time interval until the next control

step, when the same procedure is repeated. Usually, the control objective is required to follow a user defined trajectory $\mathbf{y}(t) = \mathbf{y}_{des}(t)$, where $\mathbf{y}_{des}(t)$ is the desired evolution for the components of the vector of tracked outputs. The optimisation problem is aimed at minimizing a stage cost based on the difference between real and desired output variables as well as on control activity during each time step plus a terminal cost evaluated at the end of the integration, that is, at time $t_F = t_k + T$ (where T is the length of the receding horizon).

These applications are often developed for linear systems. As a first contribution, the algorithm proposed in this paper introduces a nonlinear MPC step for the solution of the inverse simulation problem for rotorcraft dynamics. A high-order rotorcraft model replaces the controlled plant, and its dynamics are driven by the precise and fast solution of a nonlinear inverse simulation problem over a given time horizon T . The complex model which needs to be analyzed is substituted in the inverse simulation step by a lower-order model that 1) requires a significantly shorter CPU time to solve the inverse problem and 2) is less sensitive to those problems that affect most IS schemes when applied to rotary-wings aircraft, such as non-minimum-phase response and the presence of uncontrolled dynamics [8, 9]. The control action evaluated for the low-order model is then propagated forward in time using the complex, high-order one.

The scheme proposed in the paper is based on a special case of the optimisation process, where a nominal MPC problem (such that the number of algebraic conditions on the final output matches the number of control problem unknowns) is stated in terms of constraints on the terminal state only. In particular, the increments of four output variables are required to be equal to their desired values at the end of the integration interval. A correction term, generated by means of a simple guidance scheme, is added to the desired output 1) to compensate for discrepancies between the desired output variables and their actual values obtained from the complex model at the end of the previous forward simulation step, 2) to prevent the models from drifting away from the prescribed flight path during unsteady manoeuvres, and 3) to asymptotically recover the correct airspeed, climb and turn rates during steady-state flight phases. An exchange of information on the values of control and (a possibly selected subset of) state variables between the two models is required.

In the present work, focused on the IS scheme, the possibility of preventing violations of con-

straints by means of the predictive step is not included, as it is already presented and discussed in detail in Ref. [10]. The major contribution of this paper lies in the determination of a novel IS scheme that handles complex, high-order models at a computational cost only marginally higher than that necessary for solving the same IS problem based on a less reliable low-order model of the same vehicle. In this latter respect, one should note that the determination of the feasibility for a given test manoeuvre is demonstrated for the more accurate high-order model, which in turn allows for a more reliable assessment of vehicle performance limits and manoeuvring potential. This is a relevant feature, if IS is used for evaluating helicopter HQ [1].

As an example, total required power provides one of the constraints for manoeuvre feasibility, as the required power should remain within acceptable limits (zero to maximum) during the entire manoeuvre. The manoeuvre can be made more demanding (*e.g.* by increasing the required displacement or turn rate), in order to identify vehicle performance limits for a particular configuration. This analysis can be performed on the pure baseline helicopter model, without the need of implementing stability augmentation systems used on the actual rotorcraft for improving flying qualities, when mild instabilities characterise its dynamic response to controls. In this respect, the IS algorithm acts as a feedback that can easily compensate for those unstable modes that lie within a bandwidth in the range of standard pilot tasks, such as phugoid or spiral modes at high speed or pendular modes at low speed and hover.

The inverse simulation scheme is presented in detail in the next section. Three aggressive manoeuvre examples, a hurdle-hop, a slalom, and a lateral reposition are then proposed and discussed in section III, where two different lower-order, simplified models are used for the inverse simulation step, thus demonstrating that the algorithm can accommodate various degrees of model complexity.

II. MPC Scheme for IS

A. Basic features of MPC

In order to highlight similarities between the IS scheme proposed in the present paper and the MPC approach, a few relevant features of MPC are briefly recalled. Details on MPC control approach can be found, among many other papers, in Ref. [11]. The plant to be controlled is assumed

to be a continuous-time system, described by means of a set of n first-order ordinary differential equations in the form

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{u}) \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{u} \in \mathbb{R}^m$ are state and control vectors, respectively. In the most general case, the cost function V is given by

$$V(t, \mathbf{x}, \mathbf{u}) = \int_t^{t+T} \mathbf{l}[\mathbf{x}(s), \mathbf{u}(s)] ds + \mathbf{F}[\mathbf{x}(t+T)] \quad (2)$$

where $\mathbf{l}[\mathbf{x}(t), \mathbf{u}(t)]$ and $\mathbf{F}[\mathbf{x}(t+T)]$ are the stage and terminal costs respectively. T is the length of the prediction horizon over which the optimisation process is performed, often named the receding horizon. The constraints on states and controls are defined as $\mathbf{x}(t) \in \mathbb{X}$ and $\mathbf{u}(t) \in \mathbb{U}$ for all $s \in [t, t+T]$, where \mathbb{X} and \mathbb{U} are the admissible state and control subsets, respectively. A terminal constraint set can be added to the problem formulation, defined as $\mathbf{y}(t+T) = \mathbf{g}[\mathbf{x}(t+T)] \in \mathbb{Y}$ where $\mathbf{y}(t+T)$ is the objective function at the end of the receding horizon and \mathbb{Y} is the admissible (or desired) output subset.

The control action is given by the solution of a finite-horizon open-loop optimal control problem defined by Eq. (2) for the system represented by Eq. (1), i.e. the minimization of the cost function V using the state of the plant at the current time t as the initial state. Even if the plant is described by a continuous-time model, the control $\mathbf{u}(t)$ is often determined in terms of a piecewise constant function, where the control sampling time is given by $\Delta t = T/N$, with N usually between 3 and 10. As a result, the control action evaluated by means of a MPC law is a discrete process, where the first element in the control sequence evaluated by means of the solution of the open-loop optimal control problem stated above is applied to the plant. The remaining terms are used as an initial guess in the following control step, when the same optimisation procedure is repeated. Depending on the complexity of the plant model and on the types of constraints, different techniques are used to solve the optimisation problem, which range from algebraic Riccati equations for linear (or locally linearised) models without constraints to complex numerical schemes when nonlinear models and constraints are considered [15].

Comparing the proposed Inverse Simulation scheme to the general MPC problem, the controlled

plant is substituted by a high-order helicopter model, whereas the plant model $\mathbf{f}(t, \mathbf{x}, \mathbf{u})$ is given by a low order helicopter model. No stage or terminal cost is used in the present formulation of the IS scheme, where the problem is solved by enforcing a terminal constraint only. Finally, and differently from most MPC schemes, the control action $\mathbf{u}(t)$ is required to remain constant over the whole receding horizon T , but similarly to the general MPC problem the control sampling time Δt is a fraction of T .

B. Models

The study is based on different high- and low-order models of the same helicopter, a Sikorsky UH-60 ‘Black Hawk.’ Relevant data for the vehicle are reported in Table 1. An individual blade model of this rotorcraft, based on Refs. [16] and [17], is used as the reference model for the analysis, and it will be indicated as Model A. The fuselage’s aerodynamic model is represented by a database of force and moment coefficients depending on aerodynamic angles α_{fus} and β_{fus} . The representation of the blade includes flap, lag, and dynamic twist degrees of freedom. The triangular inflow model for the main rotor is taken from Ref. [18], whereas a simple uniform inflow model is considered for the tail rotor. The dynamics of the reference model are represented in terms of the evolution of a 37 element state vector, that can be partitioned as

$$\mathbf{x} = (\mathbf{x}_B, \mathbf{x}_R, \mathbf{x}_{In})^T$$

where $\mathbf{x}_B = (u, v, w, p, q, r, \phi, \theta, \psi)^T$ collects fuselage rigid body states, $\mathbf{x}_R = (\mathbf{x}_{R_1}^T, \mathbf{x}_{R_2}^T, \dots, \mathbf{x}_{R_{N_b}}^T)^T$, with $\mathbf{x}_{R_i} = (\beta_i, \dot{\beta}_i, \zeta_i, \dot{\zeta}_i, \varphi_i, \dot{\varphi}_i)^T$, $i = 1, 2, 3, 4$, lists rotor flap, lag and twist angles and their derivatives for main rotor blades, and finally the vector $\mathbf{x}_{In} = (\nu_0, \nu_{1c}, \nu_{1s}, \nu_{0TR})^T$ features main rotor and tail rotor inflow states.

Aerodynamic loads acting on each blade are evaluated by means of strip theory, as in Ref. [16], where lift and drag force coefficients on the airfoil are determined as a function of local angle of attack and Mach number from look-up tables. This allows to take into account the effects of compressibility on the advancing blade and retreating blade stall. No blade bending and unsteady aerodynamic effects are included in the model.

In compact form, the reference model is defined by a set of 37, time variant, nonlinear ordinary

Table 1 UH-60 geometric and mass data [16].

Fuselage		
mass	m	6792 kg
moments of inertia	I_{xx}	6317 kg m ²
	I_{yy}	52215 kg m ²
	I_{zz}	49889 kg m ²
	I_{xz}	2552 kg m ²
Main rotor		
number of blades	N_b	4
radius	R	8.18 m
chord	c	0.53 m
rotor speed	Ω	27 rad s ⁻¹
Tail rotor		
number of blades	$N_{b,TR}$	4
radius	R_{TR}	1.68 m
cant angle	Γ	70 deg
tail rotor speed	Ω_{TR}	124.6 rad s ⁻¹

differential equations

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\Psi(t), \mathbf{x}, \mathbf{u}) \\ \mathbf{y} &= \mathbf{g}(\mathbf{x}) \end{aligned} \tag{3}$$

where the control vector $\mathbf{u} = (\theta_0, A_{1s}, B_{1s}, \theta_{0TR})^T$ contains commands on main rotor collective, lateral and longitudinal cyclic pitch and tail rotor collective pitch, whereas Ψ is the rotor azimuth angle. A constant rotor angular speed is assumed, such that $\Psi(t) = \Omega t$. Finally the reference model output vector \mathbf{y} contains the output variables needed for guidance law and inverse simulation step. The choice of a suitable set of output variables will be discussed later in this section.

Rutherford and Thomson [8] demonstrated that it is possible to solve an IS problem even for this class of helicopter models, but the solution is computationally demanding and numerically difficult. If the model used for the inverse simulation step becomes significantly simpler than that used for

forward simulation, considerable savings in terms of CPU time necessary for each IS step can be obtained compared to the solution of the inverse problem for the reference, high-order model.

In the present analysis a first simplified model (Model B) adopts a second order tip-path-plane (TPP) representation of main rotor blade flap motion. Linear aerodynamics is assumed for blade airfoil, which allows one to analytically derive average rotor loads transmitted to the fuselage [19]. The resulting 19 elements of the state vector $\tilde{\mathbf{x}}$ of the simplified model can be partitioned as in the previous case in the form $\tilde{\mathbf{x}} = (\tilde{\mathbf{x}}_B, \tilde{\mathbf{x}}_R, \tilde{\mathbf{x}}_{In})^T$, with the same fuselage rigid body and inflow states, $\tilde{\mathbf{x}}_F$ and $\tilde{\mathbf{x}}_{In}$, respectively. Rotor states represent first order flap harmonic coefficients and their derivatives, that is, $\tilde{\mathbf{x}}_R = (\tilde{\beta}_0, \dot{\tilde{\beta}}_0, \tilde{\beta}_{1c}, \dot{\tilde{\beta}}_{1c}, \tilde{\beta}_{1s}, \dot{\tilde{\beta}}_{1s})^T$, where $\tilde{\beta}_0$, $\tilde{\beta}_{1c}$, and $\tilde{\beta}_{1s}$ are coning, longitudinal and lateral flapping coefficients, respectively. Compared to Model A, Model B neglects blade lag and twist dynamics, together with compressibility effects and retreating blade stall. This means that important effects are missed, which are relevant for a reasonably accurate identification of performance limits in manoeuvring flight by means of IS.

The dynamic properties of the simplified model are thus represented by means of a set of 19 nonlinear time-invariant ordinary differential equations, in the form

$$\begin{aligned}\dot{\tilde{\mathbf{x}}} &= \tilde{\mathbf{f}}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) \\ \tilde{\mathbf{y}} &= \tilde{\mathbf{g}}(\tilde{\mathbf{x}})\end{aligned}\tag{4}$$

where $\tilde{\mathbf{u}} = (\tilde{\theta}_0, \tilde{A}_{1s}, \tilde{B}_{1s}, \tilde{\theta}_{0TR})^T$ is the command vector and $\tilde{\mathbf{y}}$ is the output vector. Note that states, commands and outputs of the model used for the inverse simulation step are defined by symbols with a “tilde” in order to underline the fact that, in general, they may assume different values with respect to their counterparts in the reference models due to the difference in modelling level and tracking error of the output variables during the procedure.

The MPC-IS scheme is also tested by solving the IS step for a minimum-complexity model (Model C), where further simplifying assumptions are used in order to drop inflow states and some rotor variables [20]. In particular, main and tail rotor inflow is assumed uniform and quasi-steady. The values for inflow non-dimensional velocity parameters, ν_0 and ν_{0TR} , are determined at each time step by means of a simple iterative procedure based on momentum theory. The description of fuselage aerodynamics is based on parasite drag area, rather than the aerodynamic database used

Table 2 Rotorcraft models.

Model label	Main rotor			Fuselage model	No. of states
	Blade dynamics	Airfoil	Inflow		
A	Articulated flap, lag, twist	C_l & C_d from tables	3 states	Forces & moments	37
B	2 nd order TPP dynamics	linear C_l &	3 states	in tabular form	19
C	1 st order decoupl. lat and long tilt	$C_d \approx \text{const.}$	uniform	drag area	11

for Models A and B. Finally, a first-order dynamic model is assumed for rotor states, namely coning, longitudinal and lateral flap coefficients. The dynamics of this lowest-order model is thus described in terms of just 12 state variables. The state vector is partitioned in this case as $\hat{\mathbf{x}} = (\hat{\mathbf{x}}_B, \hat{\mathbf{x}}_R)^T$, where $\hat{\mathbf{x}}_B = (\hat{u}, \hat{v}, \hat{w}, \hat{p}, \hat{q}, \hat{r}, \hat{\phi}, \hat{\theta}, \hat{\psi})^T$ represents as usual fuselage states and $\hat{\mathbf{x}}_R = (\hat{\beta}_0, \hat{\beta}_{1c}, \hat{\beta}_{1s})^T$ represents rotor states.

Note that a “hat” symbol is used for indicating Model C state and control variables, in order to denote their difference with respect to those used for Models A and B. The most important features of the three models are summarised in Table 2. More details on the models and their characteristics can be found in the cited literature [16, 17, 19, 20], together with the full sets of equations of motion that represent their dynamics.

C. Initial trim conditions and MPC-IS scheme initialization

All the manoeuvres dealt with in this paper start from a given flight condition either in hover or in forward flight. In order to start the MPC-IS algorithm it is thus necessary to provide both models with the correct initial trim condition. Model A, based on an individual blade approach, is inherently time variant and oscillations in every state variable are expected at a frequency equal to (or multiple of) blade rotational speed, Ω , assumed to be constant. As a consequence, trim conditions cannot be enforced in an algebraic way by simply setting to zero all state derivatives as with fixed wing aircraft. A periodic trim needs to be found by enforcing a periodicity condition on all the states in the form

$$\mathbf{x}(t) = \mathbf{x}(t + 2\pi/\Omega)$$

for a constant value of the controls, \mathbf{u}_0 . The values of control variables are chosen so as to determine (on average) a desired flight condition, defined in terms of airspeed, V , climb rate, \dot{h} , and heading angle, χ (or turn rate, $\dot{\psi}$). The mean value of states over one rotor revolution

$$x_{i_0} = \frac{\Omega}{2\pi} \int_t^{t+2\pi/\Omega} x_i dt \quad (5)$$

is used for defining the state variables at trim. Several techniques can be found in the literature for solving the problem of helicopter periodic trim. In particular harmonic balance, periodic shooting, and autopilot techniques have been proposed over the years and compared in Ref. [21]. In the present work, a periodic shooting approach derived from the work by McVicar and Bradley [22] is used for trimming Model A.

For Models B and C, a set of nonlinear time-invariant ordinary differential equations, Eq. (4), describes the vehicle dynamics, so that the helicopter model can be trimmed by means of algebraic tools, simply enforcing the condition

$$\tilde{\mathbf{f}}(\tilde{\mathbf{x}}_0, \tilde{\mathbf{u}}_0) = 0$$

where $\tilde{\mathbf{x}}_0$ and $\tilde{\mathbf{u}}_0$ are the state and control variables at trim.

Since each model generates slightly different values for state and control variables at trim for the same flight condition, the variation of states and controls from their trim values is used, rather than their absolute values, as far as the latter ones are biased by this slight initial difference. This difference, integrated over time, would result in a significant drift between the models used in the MPC-IS scheme. By means of this elementary procedure, the initial difference between equilibrium states for the two models has no impact on the evaluation of the dynamic behaviour of the vehicle.

In what follows, the symbols $\Delta\mathbf{u} = \mathbf{u} - \mathbf{u}_0$ indicate control variable increments with respect to the considered reference trim condition for Model A. Similarly, state vector increments for Model A are defined as $\Delta\mathbf{x}_R = \mathbf{x}_R - \mathbf{x}_{R_0}$, $\Delta\mathbf{x}_F = \mathbf{x}_F - \mathbf{x}_{F_0}$, $\Delta\mathbf{x}_{In} = \mathbf{x}_{In} - \mathbf{x}_{In_0}$ for rotor, fuselage and inflow states, respectively. Similar definitions hold for Models B (*e.g.* $\Delta\tilde{\mathbf{u}} = \tilde{\mathbf{u}} - \tilde{\mathbf{u}}_0$) and C (*e.g.* $\Delta\hat{\mathbf{x}} = \hat{\mathbf{x}} - \hat{\mathbf{x}}_0$). Note that each model has a different vector of rotor states. This characteristic affects the IS scheme, and it will be discussed in detail in the following subsection.

D. Inverse simulation algorithm

The approach for the solution of the inverse problem is described in Fig. 1. Three major blocks form the basis of the algorithm architecture. The forward simulation block performs the forward simulation of the reference Model A. The inverse simulation block evaluates the command increment $\Delta \mathbf{u}$ that achieves a prescribed increment $\Delta \mathbf{y}^*$ for the tracked output variables on the basis of either Model B or C. Finally, the guidance block generates the prescribed output increment $\Delta \mathbf{y}^* = \Delta \mathbf{y}_{des} + \Delta \mathbf{y}_{guid}$ for the inverse simulation block, based on the desired trajectory $\mathbf{y}_{des}(t)$, plus a correction $\Delta \mathbf{y}_{guid}$ that aims at limiting the drift between actual and desired output variables. Details on how $\Delta \mathbf{y}_{guid}$ is generated are provided in subsection IIG.

The nonlinear IS problem is solved by means of an integration algorithm. In a standard inverse simulation approach [23], once a desired variation with time $\mathbf{y}_{des}(t)$ of the output is prescribed (*i.e.* a manoeuvre profile like those required by ADS-33 specifications [24]), helicopter equations of motion are integrated from an initial condition $\mathbf{x}_I = \mathbf{x}_k$ at time t_k over a time interval T for a piece-wise constant value \mathbf{u}_k^* of the control variables. The resulting value $\mathbf{y}_F = \mathbf{g}(\mathbf{x}_F)$ of the output variables at time $t_F = t_k + T$ can thus be represented in terms of a function $\mathbf{y}_F = \mathbf{F}(\mathbf{x}_k, \mathbf{u}_k^*)$ of the (given) initial state \mathbf{x}_k and of the (unknown) constant control action, \mathbf{u}_k^* . The unknown control vector, \mathbf{u}_k^* , is evaluated iteratively by means of some suitable numerical approach (Newton-Raphson [23], local optimisation [6], etc.), until \mathbf{y}_F matches the desired output at the final time, $\mathbf{y}_{des}(t_F)$. The control action is then propagated forward in time for only a fraction $\Delta t = T/N$ of the inverse simulation time interval [23].

In this latter respect, a proper choice of the receding horizon T and time-step Δt is crucial, in order to obtain adequate numerical performance and, at the same time, a feasible and reasonable inverse solution. The selection of T and N results from a trade-off between computational time and stability of the method. A short integration time may excite uncontrolled dynamics and lead to an unstable or highly oscillatory response of the system, both of which should be discarded as poor and/or impractical solutions of the inverse problem. The value of T must thus be sufficiently large, in order to allow non-minimum-phase response to settle down, but if N is large, large fractions of the time-history from the IS solution are dropped in the forward simulation step, and the computation

time becomes obviously longer.

This approach is common practice in Model Predictive Control, as recalled above (see [11] for details). For the present work the integration time-interval T is selected so that $T = N\Delta t$, with $N = 3$ and $\Delta t = 0.2$ s for all the results proposed in the next section. Similar results are obtained when $N = 5$, with a corresponding increment of approximately 66% in terms of CPU time, whereas $N = 2$ results in an unacceptable solution for the inverse problem during critical phases of the most aggressive manoeuvres.

The approach described above becomes even more demanding from the computational point of view for individual blade models featuring as many as 37 states, such as Model A for the present analysis. The resulting computational time becomes considerably high, even on modern CPUs, and a problem of uncontrolled states and non-minimum phase response can harm the convergence of the scheme and/or the practical feasibility of the command law, in the presence of large amplitude oscillations.

The computational burden is reduced by solving the inverse problem on the basis of a lower-order, simplified helicopter model. Some changes to the inverse simulation integration method are required in order to achieve robustness and improve tracking performance. Assuming that Model B is used for the inverse solution of vehicle motion, at every time step t_k the inverse simulation block evaluates the control action $\tilde{\mathbf{u}}_k^*$, that achieves the desired variation $\Delta\mathbf{y}^* = \Delta\mathbf{y}_{des} + \Delta\mathbf{y}_{guid}$ at the end of the IS step, that is, for $t_F = t_k + T$, where $\Delta\mathbf{y}_{des} = \mathbf{y}_{des}(t_k + T) - \mathbf{y}_{des}(t_k)$ and $\Delta\mathbf{y}_{guid}$ is proportional to the difference between the output variables of Model A at time t_k and their desired values for the tracked trajectory at the same instant.

As in any other IS algorithm, the value $\tilde{\mathbf{y}}_F = \tilde{\mathbf{g}}(\tilde{\mathbf{x}}_F)$ of the output variables for the simplified model at time $t_F = t_k + T$ depends on a known initial state $\tilde{\mathbf{x}}_k$ at time t_k and on the unknown control action, $\tilde{\mathbf{u}}_k^*$, assumed piece-wise constant. The value of the control increment $\Delta\tilde{\mathbf{u}} = \tilde{\mathbf{u}}_k^* - \tilde{\mathbf{u}}_0$ is then passed to the forward simulation as command displacement from trim condition, assuming $\Delta\mathbf{u} = \Delta\tilde{\mathbf{u}}$. From the knowledge of the initial condition for state variables at time t_k and controls at trim, \mathbf{u}_0 , the forward simulator integrates the equations of motion for Model A, assuming a constant value of the control variables, $\mathbf{u} = \mathbf{u}_0 + \Delta\mathbf{u}$, over a time step equal to $\Delta t = T/N$.

A perfectly analogous scheme is easily implemented if Model C is used instead of Model B in the IS block. Prior to implementing this IS scheme, a few issues need to be properly taken into account: 1) the choice of constrained output variables, 2) initialization of the IS step, and 3) definition of the guidance logic. The first issue is typical of all aeronautical applications of IS schemes, whereas the remaining two characterise the development of the MPC-based IS scheme.

E. IS problem constraints

If the helicopter is required to follow a prescribed trajectory, the flight task element can be enforced by setting as constraints at every time step either the inertial position, inertial velocity components, or inertial acceleration, as discussed in Ref. [2]. Choosing the acceleration components as constraints makes the problem numerically more stable, but at the same time it may lead to large drift from the desired trajectory, as the system integrates twice the error on the considered constraints, whereas setting the position as desired variables may lead to instability in the algorithm. Inertial velocity components were thus chosen as the baseline desired output to be tracked by means of the inverse simulation technique.

At this point, a redundant problem with $m > p$ is obtained, where 4 controls are available for tracking 3 trajectory variables. A further constraint can be added to make the problem nominal [23], such that $m = p$. An additional constraint that depend on the type of manoeuvre considered is thus added to the basic velocity tracking task, to provide the manoeuvre with desired characteristics, such as nose pointing (when a value to either yaw or sideslip angle is assigned) or zero lateral acceleration, that results into the execution of coordinated turns only. If well posed, a nominal problem can be solved by means of standard numerical techniques, such as Newton–Raphson (NR) method, as in the present case. The elements J_{ij} of the Jacobian matrix

$$J_{ij} = \frac{\partial}{\partial u_j} (\Delta y_i^* - \Delta \tilde{y}_i)$$

are evaluated by means of a centred finite-difference approach. An absolute termination tolerance $\varepsilon = 10^{-5}$ is used for stopping the algorithm when $|\Delta y_i^* - \Delta \tilde{y}_i| \leq \varepsilon$ for $i = 1, \dots, m = p$.

F. Initial conditions for IS step

Since a reduced order model (either B or C) is adopted for the IS step, both states $\Delta \mathbf{x}(t_k + \Delta t)$ and output variables $\mathbf{y}(t_k + \Delta t)$ achieved at the end of the simulation step for Model A are (hopefully only slightly) different from their counterparts for the IS step, $\Delta \tilde{\mathbf{x}}(t_k + \Delta t)$ and $\tilde{\mathbf{y}}(t_k + \Delta t)$, determined on the basis of a simplified model. These discrepancies need to be taken into account when the initial conditions for the simplified model at each initial time t_k are defined and the control objectives for the IS step prescribed.

The ideal choice of setting $\tilde{\mathbf{x}}_I = \mathbf{x}_k$ for Model B (or $\hat{\mathbf{x}}_I = \mathbf{x}_k$, when Model C is adopted in the IS scheme) is ruled out by the fact that the two state vectors contain different sets of variables. Moreover, some of the states would not be accessible to direct measurements, if the algorithm is implemented as an MPC controller for an actual vehicle, rather than an off-line inverse simulation method for a complex helicopter model. For this reason, the issue of state initialization for Models B (or C) at the beginning of every IS time step t_k needs to be properly addressed.

For the inverse simulation step, the initialization of states must rely (at least partially) on the knowledge of the states of the reference model which is integrated forward in time, in order to prevent a drift between the two models and consequent loss of control when implementing the control action derived from the simplified model on the full-order one, which is flying the same trajectory for slightly different values of the common state variables. Two options are considered. In the first option as much information as possible is passed from the complete model to the reduced order one. In what follows, this technique will be referred to as full state initialization, inasmuch as initial conditions for all the state variables of the simplified model are derived from the knowledge of the states for Model A at the end of the previous time-step. In particular, increments for fuselage and inflow variables are evaluated and the initial states for the inverse simulation step are given by

$$\tilde{\mathbf{x}}_B(t_k) = \tilde{\mathbf{x}}_{B_0} + [\mathbf{x}_B(t_k) - \mathbf{x}_{B_0}] \quad (6)$$

$$\tilde{\mathbf{x}}_{In}(t_k) = \tilde{\mathbf{x}}_{In_0} + [\mathbf{x}_{In}(t_k) - \mathbf{x}_{In_0}] \quad (7)$$

where \mathbf{x}_{B_0} and $\tilde{\mathbf{x}}_{B_0}$ are the values at trim of rigid body states for reference and inverse models, respectively, and $\mathbf{x}_B(t_k)$ is the vector of rigid body states at the end of the previous forward integration step. Similarly \mathbf{x}_{In_0} and $\tilde{\mathbf{x}}_{In_0}$ represent inflow states for the reference and inverse models,

and $\mathbf{x}_{In}(t_k)$ is the reference model inflow state at the end of the previous forward integration step.

As for rotor states, coning, longitudinal and lateral flapping coefficients at time t_k are evaluated by means of multiblade coefficients [25]:

$$\begin{aligned}\beta_0(t_k) &= \frac{1}{N_b} \sum_{j=1}^{N_b} \beta_j(t_k) \\ \beta_s(t_k) &= \frac{2}{N_b} \sum_{j=1}^{N_b} \beta_j(t_k) \sin \psi_j \\ \beta_c(t_k) &= \frac{2}{N_b} \sum_{j=1}^{N_b} \beta_j(t_k) \cos \psi_j\end{aligned}\tag{8}$$

where N_b is the number of blades. Provided that the time derivatives of multiblade coefficients can be analytically derived from Eq. (8), and letting $\boldsymbol{\beta} = (\beta_0, \dot{\beta}_0, \beta_c, \dot{\beta}_c, \beta_s, \dot{\beta}_s)^T$, the initial condition for rotor states is defined as

$$\tilde{\mathbf{x}}_R(t_k) = \tilde{\mathbf{x}}_{R_0} + [\boldsymbol{\beta}(t_k) - \boldsymbol{\beta}_0]\tag{9}$$

A second option is also analyzed, based on the hypothesis that only rigid body states \mathbf{x}_B of the reference model are truly observable, as it would happen in a real-time application of the algorithm in the form of an actual MPC scheme. This technique will be referred to as partial state initialization in the sequel. In this case the same displacement of fuselage states from their values at trim is assumed for the initial condition at time t_k of the simplified model, as prescribed by Eq. (6), whereas inflow and rotor states are not updated from the corresponding values obtained for Model A at the end of the previous forward integration interval. In this respect, inflow and rotor states are assumed as not observable and therefore they are initialised with the value achieved at the end of the last inverse simulation run $\tilde{\mathbf{x}}_{R/In}(t_k) = \tilde{\mathbf{x}}_{R/In}(t_{k-1} + \Delta t)$ for the simplified Model B.

Note that, when Model C is adopted in the IS block, an equivalent definition for its initial conditions at each time t_k is easily derived by dropping inflow states and substituting the tilde with a hat. Also remember that, in this second case, first-order dynamics are assumed for flapping coefficients, so that, from the definition of $\hat{\mathbf{x}}_R = (\hat{\beta}_0, \hat{\beta}_{1c}, \hat{\beta}_{1s})^T$, only the current values of multiblade coordinates derived from Eqs. (8) are needed, whereas their derivatives are no longer necessary.

The choice of selecting rigid body states only as observable states maintains a link to Model Predictive Control procedures. If a real system replaces the forward simulation model, only some

states would be accessible to direct measurements. In particular, linear and angular velocities as well as attitude variables are available from GNC sensors and, as a consequence, they are fed to initialize the inverse simulation step, that acts like a controller for the plant. Rotor and inflow states are in general not subject to direct measurement, and therefore no feedback of their actual value from the controlled plant could be provided to the inverse simulation model in a realistic scenario. At the same time, and more importantly in the present off-line inverse simulation framework, partial state initialization seriously challenges the robustness of the MPC-IS algorithm. This allows one to fully assess the capabilities of the method.

G. Desired output for IS (with a guidance scheme)

When a standard IS scheme is adopted, the same vehicle model is used for the solution of the inverse problem and forward propagation of the control action. The model follows the desired variation of the outputs, $\mathbf{y}_{des}(t)$, whenever a successful inverse solution is obtained. This is no longer true for the MPC-IS scheme, where the output variables $\mathbf{y}(t_k + \Delta t)$ for Model A achieve different values compared to those obtained at the same time instant for the simplified model (either B or C) during the solution of the IS step and, as a consequence, they are different from the desired output $\mathbf{y}_{des}(t_k + \Delta t)$.

Several reasons contribute to this difference: 1) the constraints on the output are exactly enforced during the IS step for the simpler model only; 2) the control action is propagated for just a fraction of the receding horizon T , so that the actual output is evaluated at a time $t_{k+1} \neq t_k + T$; and 3) the output variable increment required over the IS step for the simplified model, $\Delta \mathbf{y}^* = \Delta \mathbf{y}_{des} + \Delta \mathbf{y}_{guid}$, includes the contribution $\Delta \mathbf{y}_{guid}$ generated by the guidance term required for limiting the drift between the two models, so that $\Delta \mathbf{y}_{des} = \mathbf{y}_{des}(t_k + T) - \mathbf{y}_{des}(t_k) \neq \Delta \mathbf{y}^*$.

Letting the actual increment achieved by Model B at time t_F be defined as $\Delta \tilde{\mathbf{y}} = \tilde{\mathbf{y}}_F - \tilde{\mathbf{y}}_I = \tilde{\mathbf{g}}(\tilde{\mathbf{x}}_F) - \tilde{\mathbf{g}}(\tilde{\mathbf{x}}_k) = \tilde{\mathbf{F}}(\tilde{\mathbf{x}}_k, \tilde{\mathbf{u}}_k^*) - \tilde{\mathbf{g}}(\tilde{\mathbf{x}}_k)$, the inverse problem can be stated again in terms of a set of algebraic equations in the form

$$\tilde{\mathbf{F}}(\tilde{\mathbf{x}}_k, \tilde{\mathbf{u}}_k^*) - \tilde{\mathbf{g}}(\tilde{\mathbf{x}}_k) = \Delta \mathbf{y}^*. \quad (10)$$

Note that, as a further variation with respect to a standard IS method, a different definition of

the algebraic system is adopted in this paper, where, rather than directly enforcing the constraints in terms of actual desired values for the tracked variables at time t_F , their increments over the time interval T between t_I and t_F are required to be equal. The guidance term included in the definition of $\Delta \mathbf{y}^*$ corrects the desired output variables increment by means of a term proportional to the error exhibited by the reference model at the end of the previous forward simulation step.

A simple linear guidance scheme is adopted, where

$$\Delta \mathbf{y}_{guid} = K [\mathbf{y}_{des}(t_I) - \mathbf{g}(\mathbf{x}_k)]$$

such that

$$\Delta \mathbf{y}^* = \Delta \mathbf{y}_{des} + \Delta \mathbf{y}_{guid} = [\mathbf{y}_{des}(t_F) - \mathbf{y}_{des}(t_I)] + K [\mathbf{y}_{des}(t_I) - \mathbf{g}(\mathbf{x}_k)] \quad (11)$$

The additional guidance term also enforces asymptotic convergence on the tracked variables when they achieve a constant value during steady-state flight segments (*e.g.* at the end of a manoeuvre).

For both partial and full state initialization techniques, fuselage states for the simplified model are always updated to their actual values achieved by Model A at the end of the forward simulation step, Δt . Thus, it is $\tilde{\mathbf{g}}(\tilde{\mathbf{x}}_k) = \mathbf{g}(\mathbf{x}_k)$, and it is possible to rearrange Eqs. (10) and (11) in the form

$$\tilde{\mathbf{F}}(\tilde{\mathbf{x}}_k, \tilde{\mathbf{u}}_k^*) = \mathbf{y}_{des}(t_F) + (K - 1) [\mathbf{y}_{des}(t_I) - \mathbf{g}(\mathbf{x}_k)]$$

This formulation for the IS problem constraints allows one to highlight the following facts. For $K = 0$ the guidance term disappears and one simply requires that the increment of the actual output variables at the end of the whole inverse simulation step $T = t_F - t_I$ equals the increment for the desired variation of \mathbf{y}_{des} over the same interval, without taking into account the initial error. In this case the error on the output slowly grows during the manoeuvre. If a value $K = 1$ is used, the second term between square brackets, multiplied by $K - 1$, disappears. The lower order model is thus required to exactly follow the desired variation with time of the output, that is, the IS scheme no longer works on the desired output increment. At the same time no information on the error on the tracked variables for the more complex model is available at the beginning of the IS step. This causes the inverse solution to rapidly diverge. An intermediate value between 0 and 1 needs to be found which is suitable for the considered application. In this research a value of $K = 0.3$ was adopted throughout.

III. Results and Discussion

A. Test cases

The approach described in the previous section is demonstrated for a longitudinal and two lateral-directional manoeuvres, taken from the inverse simulation literature [1, 9, 23] and from ADS-33 E requirements [24], respectively. The first manoeuvre is a hurdle-hop performed in 20 s at a constant speed $V = 30$ m/s, with a vertical displacement equal to 30 m and a maximum rate of climb $\dot{h}_{max} \approx 7$ m/s. The second manoeuvre is a slalom with gates displaced 15 m away from the centerline, performed in 13 s at a constant speed $V = 35$ m/s, resulting in a maximum rate of turn $\dot{\psi}_{max} \approx 17^\circ/\text{s}$. Finally, a lateral reposition manoeuvre is simulated, which requires the helicopter to start from a hover condition, move laterally 120 m in 16 s and come to a complete stop, maintaining a constant nose-pointing during the manoeuvre.

Trim conditions for Models A and B (or A and C) are evaluated for initializing the procedure. Then, following the approach previously described, the inverse solutions based on the simplified models (either B or C) are calculated for the two manoeuvres using inertial velocities as desired output variables, plus an additional constraint introduced to make the IS problem nominal. The condition $\psi = 0$ is enforced for the longitudinal hurdle-hop and for the lateral reposition, that results in a constant nose-pointing during the manoeuvre. A constraint on zero lateral acceleration, $a_y = 0$, is introduced for the slalom manoeuvre, performed by means of coordinated turns.

For each manoeuvre, three solutions are compared. The reference one, based on a standard integration method for the individual blade model (IS Model A), is represented in the figures by a dashed line. A second solution, represented by means of dotted lines, is obtained for the same problems by inverse simulation of the simplified model (IS Model B or C); finally, a continuous line is used to represent state and control variables obtained from the solution of the same inverse problems by means of the novel MPC-IS approach. In all the figures, commands are scaled with respect to their maximum travel. In particular main rotor collective θ_0 varies between 0 and 1, whereas longitudinal and lateral cyclic pitch and tail rotor collective are scaled between -1 and $+1$.

B. MPC-IS with IS step based on Model B

In the first set of test cases discussed, Model B is used as the simplified, low order model for the inverse simulation step for the MPC-IS technique. When full state initialization of the inverse problem is adopted, as much information as possible from the complete model is provided to the simplified one at the end of each simulation step, and the results of the MPC-IS scheme perfectly match the inverse solution obtained with an integration method based on Model A alone. Only results produced by means of partial state initialization, with update limited to fuselage rigid body states, are presented, as this technique challenges the robustness of the new IS scheme.

Figure 2 represents the projection on the X - Z inertial plane for the trajectory during a hurdle-hop. Lateral displacements (not reported) remain small during the entire manoeuvre, with peak values in the order on 1 m only, for the MPC-IS scheme, and around 0.1 m for IS based on Model A and B alone. All the three techniques evaluate a feasible solution that tracks the desired trajectory with great precision for this aggressive manoeuvre. A small deviation from the nominal path is barely visible, especially in the final part of the manoeuvre, for the MPC-IS solution. In this case, during the fast descent phase, differences between Models A and B used for forward and inverse simulation, respectively, become more significant and the guidance term modifies the desired output of the IS problem for limiting drift between the two models. This introduces a minor delay in tracking the trajectory variables, but accuracy remains satisfactory.

In spite of the trajectories being almost identical in the three cases, a few discrepancies between the solutions are apparent in the time-histories of control and state variables. These differences are related to both the numerical technique and the effects of the simplifying assumptions at the basis of the simplified model. Figures 3 and 4 compare the time histories for the controls in terms of absolute position and displacement from trim, respectively, whereas Fig. 5 represents attitude angles during the manoeuvre. Initial and final differences at trim between the IS for Model A and the MPC-IS on one side, and the IS for Model B on the other one, are visible in Fig. 3. The difference on main rotor collective is related to the fact that Model B neglects rotor blade dynamic twist, and a bias in θ_0 is present. Other effects are analyzed in more detail in Ref. [23], focused on the impact of helicopter modeling technique on the results obtained from inverse simulation.

This initial bias is removed when command displacements from trim is considered (Fig. 4). In this case the variation of collective pitch is identical in the three cases. The inverse solution of Mod. B shows more significant differences on the other command channels, namely A_{1s} , B_{1s} , and $\theta_{0_{TR}}$. These differences are again related to the different rotor model employed, especially during the initial and final transient phases, when linear aerodynamics for rotor blades provides a worst approximation for the forces developed by the rotor in ascending and descending flight.

Helicopter pitch response, represented in Fig. 5.b, is similar for all the techniques. Minor differences are present in lateral-directional states, such as roll and yaw angles, shown in Figs. 5.a and 5.c, that exhibit a smaller range of variation for this longitudinal test-case.

Command travel, state variables and required power (not reported in the figures) evaluated by means of the MPC-IS technique remain very close to those obtained from the computationally more demanding solution of the inverse problem by means of the integration method applied to Model A. Command time histories match almost perfectly those obtained for the IS of Model A. Minor differences are present only for a short interval at the beginning of the descent phase, when some higher frequency oscillations in longitudinal pitch rate in the IS results for Model A are smoothed by the MPC-IS technique. In the latter solution the transition to horizontal flight is slightly more gradual, thanks to the effects of the guidance term generated for tracking the velocity variables in the presence of errors introduced by the use of different models in the forward and inverse simulation steps. A marginally less accurate tracking of the velocity variables thus results in an overall smoother control action.

Figures 6 to 9 represent the results obtained for the slalom manoeuvre. Again, as shown in Fig. 6, where the projection of the trajectory in the X - Y plane is reported, the solution evaluated by means of the MPC-IS approach provides an inverse solution that correctly tracks the prescribed manoeuvre. Variation of altitude during the manoeuvre remains smaller than 1 m for all three IS schemes. Also in this second test-case, the conventional integration method provides a feasible solution to the inverse problem that follows the desired trajectory, when applied to both Models A and B, but discrepancies in the corresponding command time histories and resulting state variables are more significant, especially on the cyclic pitch control channels, A_{1s} and B_{1s} (Fig. 7.b and c).

These differences are related to the type of rotor model implemented in the two cases, the response of which is affected by the complex coupling of longitudinal and lateral dynamics during the slalom. In spite of this, when Models A and B are implemented independently in a conventional IS scheme, the use of the inverse solution obtained from Model B in the MPC-IS scheme provides an evolution of the control variables that follows the same trend of the inverse solution for Model A alone. This means that differences between the models are correctly compensated by the updating process of the initial conditions for rigid-body fuselage states and the guidance term introduced in the definition of the desired output for the IS step. Minor discrepancies on maximum command travel are visible, such as lower peaks for cyclic pitch and tail rotor collective. Such differences are reduced when a full-state initialization technique is adopted, without disappearing, contrary to what happens for the longitudinal manoeuvre considered previously, when full-state initialization makes the MPC-IS scheme recover almost exactly the inverse solution obtained from Model A.

For an aggressive slalom manoeuvre, the smoothing action of the guidance term on the evolution of the tracked variables plays a more significant role, resulting in a reduction of the peaks on command displacement from the initial trim values. It should be remembered that the guidance logic corrects the drift from the desired trajectory adding a correction term that is determined at the end of the IS time step Δt and acts on the following one. This means that the correction is enforced with a small delay. The effect of this delay, together with the discrepancies between the responses of Models A and B when partial state initialization is adopted, are at the basis of this small, yet evident discrepancy in the inverse solution. If only fuselage rigid body states for Model B are updated to the values achieved by Model A at the end of the step, the control action determined during the IS step needs to account for a settling time for those states (*i.e.* TPP and inflow states) that are not correctly updated.

The evolution of Euler angles (shown in Fig. 8) and that of angular rates (not reported in the figures) are again very similar for all three solutions. Differences smaller than half a degree are present for helicopter attitude angles when the MPC-IS approach is compared to the inverse solution of Model A obtained by means of a conventional integration method. Roll and yaw angles follow exactly the same patterns. The only visible difference in Fig. 8 is on the pitch angle, and

mainly for a matter of resolution, given the limited range of the variable. Again, differences on the inverse solution obtained from Model B within the IS step of the MPC-IS algorithm are successfully compensated by the initial condition updating scheme and the guidance term adopted.

Inverse simulation schemes can be used to evaluate the feasibility of a given manoeuvre, in terms of command travel and required power. Limiting performance is obtained by challenging the vehicle model on more demanding tasks (*e.g.* increasing the required displacement of the gates from the centerline, for the slalom manoeuvre), until a limit on maximum command travel or maximum available power is violated. The feasibility of the control action is readily available from the inverse solution, whereas the constraint on available power requires an estimate of power absorbed by main and tail rotors, taking into account gearbox efficiency and an estimate of on-board system required power. More reliable estimates of manoeuvre feasibility or limiting performance are obviously obtained from more accurate models. In this respect, Fig. 9 shows that the linear aerodynamics assumed for the main rotor in Model B provides a significantly different estimate of required power during the slalom manoeuvre, whereas the estimate of required power based on the MPC-IS solution remains close to the results obtained for the integration method applied to Model A alone. This means that the two solutions, although obtained by means of different numerical approaches, provide consistent estimate for manoeuvre feasibility and limiting performance for the considered test-case in terms of both command travel and required power.

The technique is tested also on a lateral reposition manoeuvre. The three techniques determine reasonable controls for tracking the trajectory (not reported in the figure). In this case the commands evaluated with the MPC-IS approach are very similar to those evaluated with the integration method based on model A. Minor differences are visible only for lateral-directional commands (in particular tail rotor pitch) as well as in yaw attitude (Fig. 10.c). This is due to the fact that a minor error on heading angle results as a consequence of differences between the two models used in the MPC-IS scheme. Nevertheless, the error drops rapidly back to zero during the final phase thanks to the guidance block. The results for roll and pitch angles during the lateral reposition are very similar for the three approaches, as shown in Fig. 10.a and 10.b.

C. MPC–IS with IS step based on Model C

As demonstrated above, the MPC–IS approach based on Model B represents adequately all the main features of the inverse solution for the considered flight tasks. The approach is then tested using a minimum complexity model (Model C) for the IS step. Figure 11 presents the control action for the hurdle–hop manoeuvre based on Model C. The solution of the inverse problem with integration method based on Model C presents relevant differences, especially for longitudinal and lateral commands, compared with the solution for the same task obtained from Model A by means of the same IS integration method (indicated by means of dashed and dotted lines in Fig. 4, respectively). These differences, discussed in detail in Ref. [23], are related to the simplifying hypothesis assumed for the derivation of Model C.

The MPC–IS approach is again capable of evaluating a feasible command action that tracks well the desired trajectory, also when partial state initialization is adopted, as in the example shown. Similar results are obtained for the slalom manoeuvre. In both cases maximum command travel on all commands and required power are precisely estimated by the MPC–IS approach. No difficulties in converging to the inverse solution are encountered, even when very aggressive manoeuvres are dealt with, in spite of the differences between Model C, used for the IS step, and Model A, adopted for the forward propagation of the control action. Less brilliant results are obtained for the lateral reposition manoeuvre, where the uniform inflow approximation jeopardises the accuracy of the inverse solution.

D. Computation effort

The computational burden necessary for the solution of an inverse problem depends on the solution scheme and on the complexity of the model. Table 3 presents the CPU times (in seconds) for the solution of a hurdle–hop, slalom and lateral reposition manoeuvres, based on models written in Matlab and running on a 1.6 GHz CPU. Together with total CPU time, also a percentage of a reference CPU time is provided in Table 3, using as a reference (100%) the time required for solving the IS problem by means of a conventional integration method applied to Model A. In all the solutions, a time step $\Delta t = 0.2$ s and a receding horizon equal $T = 0.6$ s are used as a tradeoff

Table 3 Computational time for the solution of the inverse problems (CPU at 1.6 GHz)

		IS Mod A	IS Mod B	IS Mod C	MPC A+B	MPC A+C
Hurdle-hop	CPU-time (s)	3992	1230	1114	1274	1148
	%	100	30.8	27.9	31.9	28.8
Slalom	CPU-time (s)	2506	876	831	937	857
	%	100	35.0	33.2	37.4	34.2
Lateral	CPU-time (s)	3090	1026	1080	1053	1118
Reposition	%	100	33.2	34.9	34.0	36.2

between computational effort (that grows for shorter Δt and longer T) and stability of the numerical method. Small values of T and/or poor choices of the ratio $\Delta t/T$ may not allow transient response to settle down, resulting in strong and possibly diverging oscillations in the control action that threaten stability.

CPU time grows quite obviously with the duration of the manoeuvre (20 s for the hurdle-hop, 16 s for the lateral reposition, and 13 s for the slalom), but it also depends on the complexity of the control task, where more challenging ones require more iterations for converging during critical flight phases. When comparing the time saved by using the MPC-IS approach, there is a 6% difference between the hurdle-hop and the slalom manoeuvre, where the advantage of using a lower order model is apparently greater for the hurdle-hop. This is due to the fact that the integration method based on Model A encounters difficulties in the convergence to an inverse solution during the descending portion of the manoeuvre. In the slalom manoeuvre, Model A always converges to the inverse solution in fewer iterations and, as a consequence, the computational time saved by using Models B or C is reduced.

The solution of the IS problem by means of an integration method based on Model B (see [23] for details) is approximately 3 time faster than the solution of the same problem based on Model A. Using the MPC-IS approach the computational burden is almost identical, as the forward simulation performed with Model A requires a relatively small amount of CPU time, when it remains outside of the iterative process at the basis of the solution of the inverse problem. This confirms that a feasible and accurate inverse solution for Model A is evaluated at a cost only slightly higher than

the significantly less accurate inverse solution based on the simplified model only.

When Model C is used, only marginal improvements in terms of computation time are achieved (in two of the three manoeuvres), compared with the MPC-IS solution based on Model B, in spite of the reduced number of states (12 instead of 19) and the simpler main rotor and fuselage aerodynamic models. This happens because 1) a computationally expensive iterative process is necessary at every simulation step for evaluating main and tail rotor inflow, and 2) the computational cost of the integration scheme remains high, in spite of the simplified model dynamics, because more iterations are required for converging to a solution for the IS step, when an initial condition obtained from a different, more complex model is used. In particular, in the lateral reposition manoeuvre, due to the desired trajectory which lies completely in the lateral plane, the absence of a 3-states inflow in Model C has a negative effect on the results obtained from the controls evaluated by the inverse simulation step. A longer computational time is required to converge to the solution, when the next step is initialised on the basis of the current state of Model A, featuring larger tracking errors and more significant differences in the rotor states between the two models. In this respect, the use of a minimum complexity helicopter model does not appear to be justified by the modest gain in terms of CPU time (if any).

IV. Conclusions

A novel approach to the solution of inverse simulation problems for helicopter manoeuvres based on a model predictive control scheme is proposed. The approach significantly reduces the computational cost required by the inverse simulation of a complex nonlinear helicopter model by using a lower-order model in the inverse simulation step. In this framework, the standard integration approach to the solution of the inverse simulation problem is modified introducing an update scheme for the initial conditions of the lower-order model at the end of the forward simulation step, performed on the more complex one, and a simple guidance law that avoids the build-up of errors while tracking the prescribed variation of the output variables.

The approach, tested on three manoeuvres used for the analysis of rotorcraft handling qualities, solves the considered inverse problems with good convergence characteristics, generating accurate

trajectories that almost overlap the desired one. Time-histories for controls, state variables, and required power are very similar to those generated by a standard integration algorithm applied to the same problem, allowing for the determination of helicopter limiting performance on demanding manoeuvring tasks. When a minimum complexity helicopter model is employed, tracking performance is still adequate, but only a marginal gain in terms of computational time is obtained for two of the three test cases.

References

- [1] Thomson, D.G., and Bradley, R., "Inverse simulation as a tool for flight dynamics research - Principles and applications," *Progress in Aerospace Sciences*, Vol. 42, No. 3, May 2006, pp. 174-210.
- [2] Thomson, D.G., and Bradley, R., "Development and Verification of an Algorithm for Helicopter Inverse Simulation," *Vertica*, Vol. 14, No. 2, 1990, pp. 185-200.
- [3] Hess, R.A., and Gao, C., "A Generalized Algorithm for Inverse Simulation Applied to Helicopter Manoeuvring Flight," *Journal of the American Helicopter Society*, Vol. 16, No. 5, 1993, pp. 3-15.
- [4] Kato, O., and Sugiura, I., "An interpretation of airplane general motion and control as inverse problem," *Journal of Guidance, Control & Dynamics*, Vol. 9, No. 2, Mar.-Apr. 1986, pp. 198-204.
- [5] Hess, R.A., Gao, C., and Wang, S.H., "Generalised technique for inverse simulation applied to aircraft manoeuvres," *Journal of Guidance, Control & Dynamics*, Vol. 14, No. 5, Sep.-Oct. 1991, pp. 920-926.
- [6] De Matteis, G., De Socio, L.M., and Leonessa, A., "Solution of aircraft inverse problems by local optimization," *Journal of Guidance, Control & Dynamics*, Vol. 18, No. 3, May-Jun. 1995, pp. 567-571.
- [7] Borri, M., Bottasso, C.L., and Montelaghi, F., "Numerical Approach to Inverse Flight Dynamics," *Journal of Guidance, Control & Dynamics*, Vol. 20, No. 4, Jul.-Aug. 1997, pp. 742-747.
- [8] Rutherford, S., and Thomson, D.G., "Helicopter Inverse Simulation Incorporating an Individual Blade Rotor Model," *Journal of Aircraft*, Vol. 34, No. 5, Sep.-Oct. 1997, pp. 627-634.
- [9] Avanzini, G., and De Matteis, G., "Two-Timescale Inverse Simulation of a Helicopter Model," *Journal of Guidance, Control & Dynamics*, Vol. 24, No. 2, Mar.-Apr. 2001, pp. 330-339.
- [10] Bagiev, M., Thomson, D.G., Anderson, D., and Murray-Smith, D., "Hybrid inverse methods for helicopters in aggressive manoeuvring flight". *17th IFAC Symposium on Automatic Control in Aerospace*, June 2007, Toulouse, France.
- [11] Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M., "Constrained model predictive control: Stability and optimality", *Automatica*, Vol. 36, NO. 6, June 2000, pp. 789-814.

- [12] Hess, R.A., Jung, Y.C., “An Application of Generalized Predictive Control to Rotorcraft Terrain-Following Flight”, *IEEE transactions on systems, man, and cybernetics*, Vol. 19, No. 5, Sept.-Oct 1989, pp. 955-962.
- [13] Jung, Y.C., Hess, R.A., “Precise Flight-Path Control Using a Predictive Algorithm”, *Journal of Guidance, Control & Dynamics*, Vol. 14, No. 5, Sep.-Oct. 1991, pp. 936-942.
- [14] Bottasso, C.L., Nicasastro, R., Savini, B., and Riviello, L., “Adaptive Reference-Augmented Predictive Control”, *Proceedings of the 2007 American Control Conference*, New York City, USA, July 11-13, 2007.
- [15] Camacho, E.F., Bordons, C., *Model Predictive Control (Advanced Textbooks in Control and Signal Processing)*, 2nd ed., Springer, London 2004, Chap. 9
- [16] Howlett, J.J., “UH-60A ‘Black Hawk’ Engineering Simulation Program, Volume 1: Mathematical Model,” NASA CR 166309, Moffet Field (CA), 1981.
- [17] Howlett, J.J., “UH-60A ‘Black Hawk’ Engineering Simulation Program, Volume 2: Background Report,” NASA CR 166310, Moffet Field (CA), 1981.
- [18] Peters, D.A., and HaQuang, N., “Dynamic inflow for practical applications,” *Journal of the American Helicopter Society*, Vol. 33, No. 4, Oct. 1988, pp. 64-68.
- [19] Talbot, P.D., Tinling, B.E., Decker, W.A., and Chen, R.T.N., “A mathematical model of a single main rotor helicopter for piloted simulation,” NASA TM-84281, Moffet Field (CA), 1982.
- [20] Heffley, R.K., and Mních, M.A., “Minimum Complexity Helicopter Simulation Math Model,” NASA Technical Report CR-177476, 1988.
- [21] Peters, D.A., and Barwey, D., “A General Theory of Rotorcraft Trim”, *Mathematical Problems in Engineering*, Vol. 2, No. 1, Jan. 1996, pp. 1–34.
- [22] McVicar, J.S.G., and Bradley, R., “Robust and efficient trimming algorithm for application to advanced mathematical models of rotorcraft”, *Journal of Aircraft*, Vol. 32, No. 2, Feb. 1995, pp. 439-442.
- [23] Avanzini, G., De Matteis, G., and Torasso, A., “Modelling Issues in Helicopter Inverse Simulation,” 36th *European Rotorcraft Forum ERF2010*, Paris, Sept. 7 – 9, 2010.
- [24] Anonymous, “Handling Qualities Requirements for Military Rotorcraft,” US Army Aviation and Missile Command, ADS-33E-PRF, 2000.
- [25] Padfield, G., *Helicopter Flight Dynamics*, Blackwell Publishing, Oxford, 2007, Chap. 3.

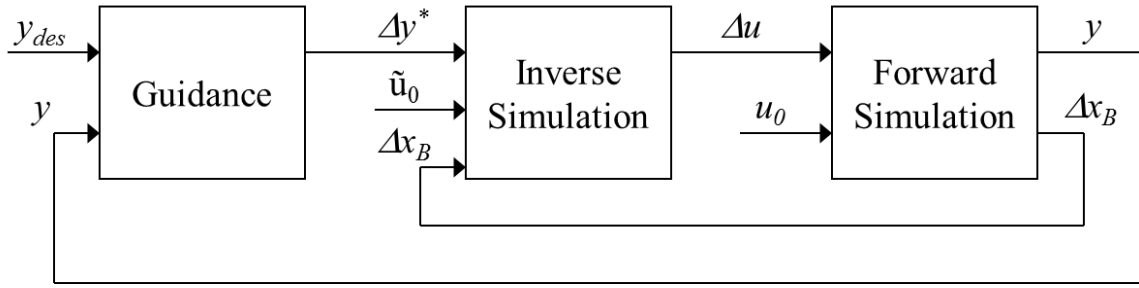


Fig. 1 Architecture of the MPC-IS scheme.

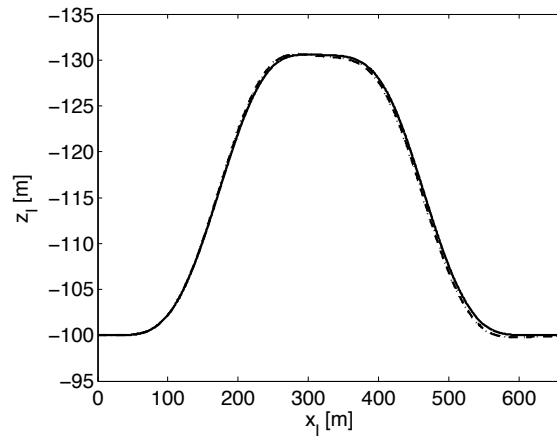


Fig. 2 Hurdle-hop trajectory (—: MPC-IS; - - - IS for Model A; ··· IS for Model B).

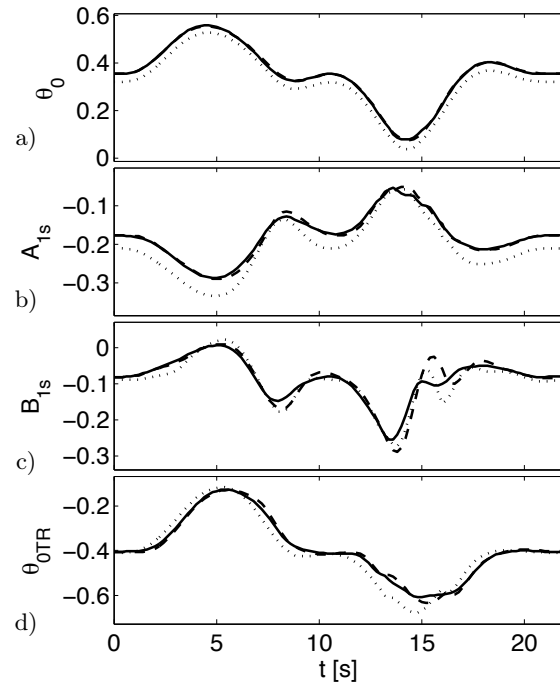


Fig. 3 Normalised command travel during a hurdle-hop manoeuvre (—: MPC-IS; - - - IS for Model A; \cdots IS for Model B).

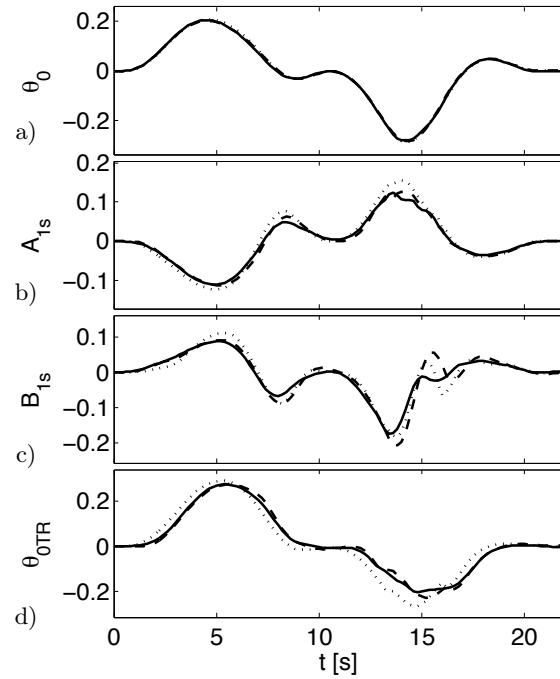


Fig. 4 Normalised command displacement from trim during a hurdle-hop manoeuvre (—: MPC-IS; - - - IS for Model A; \cdots IS for Model B).

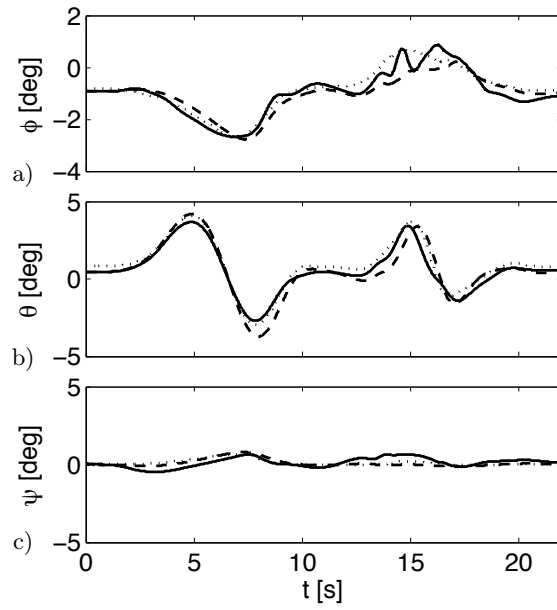


Fig. 5 Roll, pitch and yaw angles during a hurdle-hop manoeuvre (—: MPC-IS; - - - IS for Model A; ··· IS for Model B).

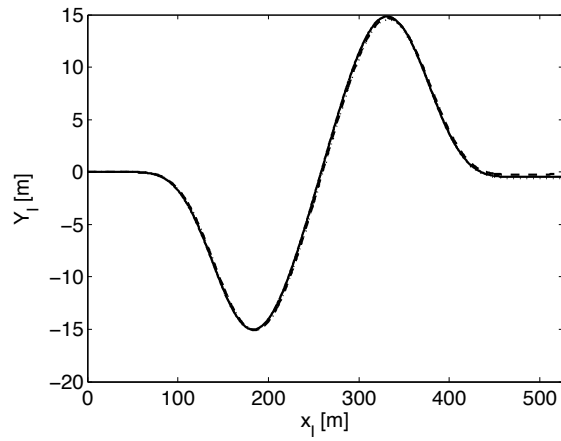


Fig. 6 Slalom trajectory (—: MPC-IS; - - - IS for Model A; ··· IS for Model B).

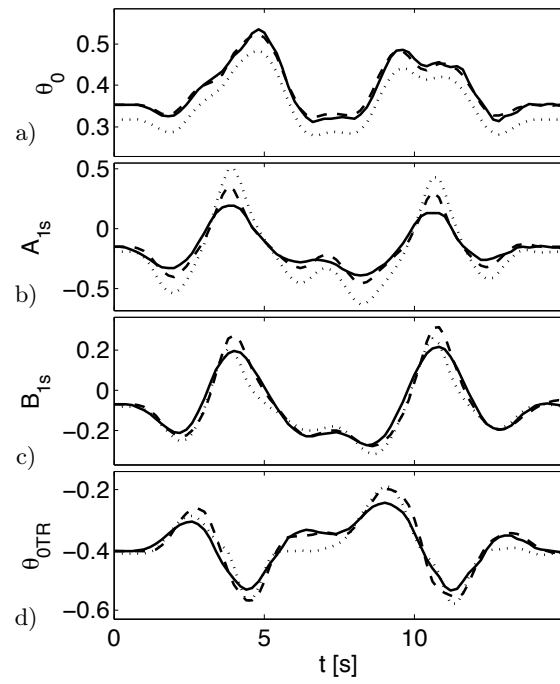


Fig. 7 Normalised command travel for a slalom manoeuvre (—: MPC-IS; - - - IS for Model A; \cdots IS for Model B).

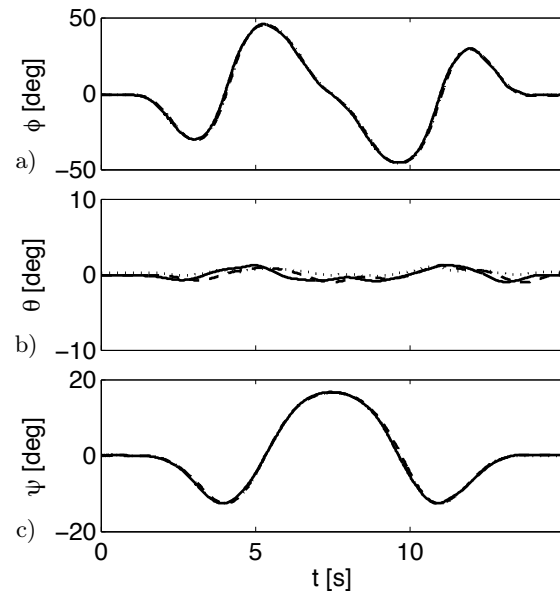


Fig. 8 Roll, pitch and yaw angles during a slalom manoeuvre (—: MPC-IS; - - - IS for Model A; \cdots IS for Model B).

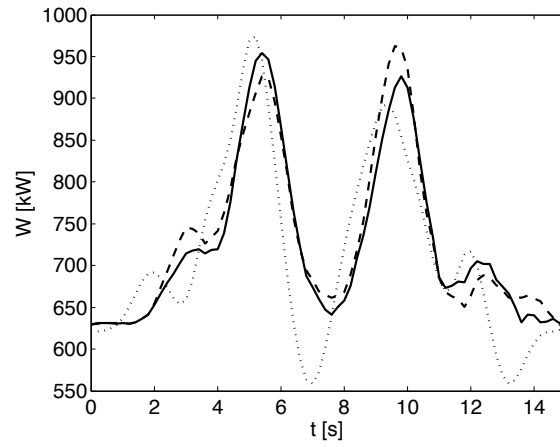


Fig. 9 Required power during a slalom manoeuvre (—: MPC-IS; - - - IS for Model A; ··· IS for Model B).

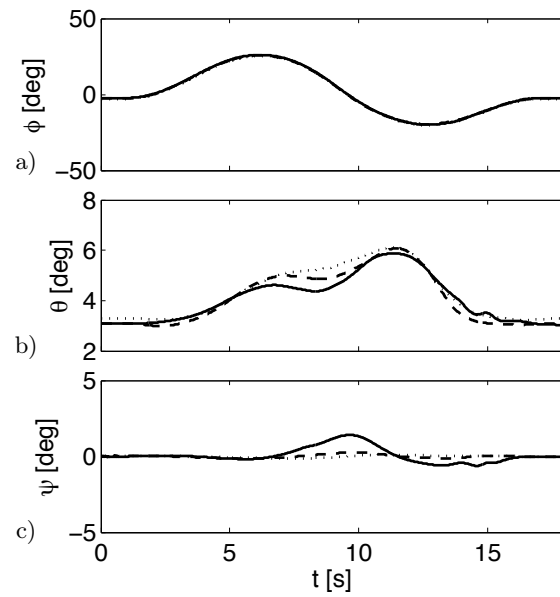


Fig. 10 Roll, pitch and yaw angles during a lateral reposition manoeuvre (—: MPC-IS; - - - IS for Model A; ··· IS for Model B).

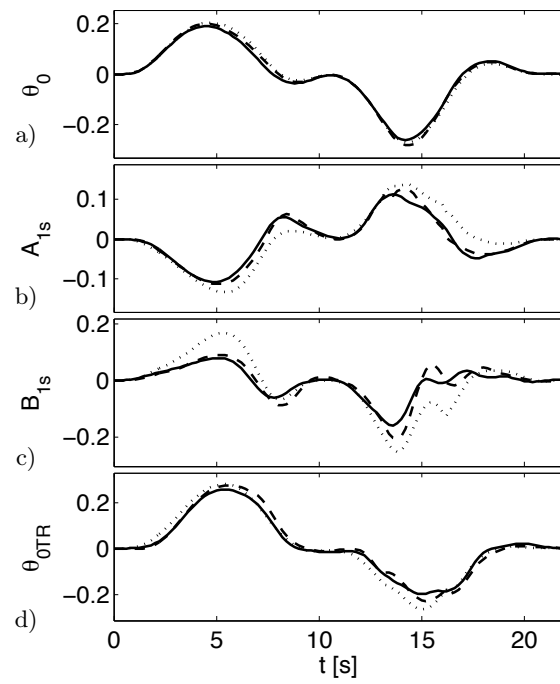


Fig. 11 Normalised command displacement from trim during a hurdle-hop manoeuvre (—: MPC-IS using Model C in the IS step; - - - IS for Model A; \cdots IS for Model C).