J Intell Robot Syst (2012) 68:69–85
DOI 10.1007/s10846-012-9669-6

Terrain Avoidance Nonlinear Model Predictive Control for Autonomous Rotorcraft

Bruno J. N. Guerreiro · Carlos Silvestre · Rita Cunha

Received: 15 September 2010 / Accepted: 4 March 2012 / Published online: 25 March 2012

Abstract This paper describes a terrain avoidance control methodology for autonomous rotorcraft applied to low altitude flight. A simple nonlinear model predictive control (NMPC) formulation is used to adequately address the terrain avoidance problem, which involves stabilizing a nonlinear and highly coupled dynamic model of a helicopter, while avoiding collisions with the terrain as well as preventing input and state saturations. The physical input saturations are made intrinsic to the model, such that the control is always admissible and the MPC design is simplified. A comparison of several optimization approaches is provided, where the performance of the traditional gradient method with fixed step is compared with the quasi-Newton method and a line search algorithm. The simulation results show that the adopted strategy achieves good performance even when the desired path is on collision course with the terrain.

Keywords Helicopter control · Obstacle avoidance · Model-based control · Predictive control · Nonlinear models

Mathematics Subject Classification (2010) 49J15 · 49N90

This work was partially supported by project FCT [PEst-OE/EEI/LA0009/2011], by project FCT AMMAIA (PTDC/HIS-ARQ/103227/2008), and by project AIRTICI from AdI through the POS Conhecimento Program that includes FEDER funds. The work of Bruno Guerreiro was supported by the PhD Student Grant SFRH/BD/21781/2005 from the Portuguese FCT POCTI program.

B. J. N. Guerreiro (✉) · C. Silvestre · R. Cunha
Institute for Systems and Robotics (ISR),
Instituto Superior Técnico (IST),
Torre Norte Piso 8, Av. Rovisco Pais 1,
1049-001 Lisbon, Portugal
e-mail: bguerreiro@isr.ist.utl.pt

C. Silvestre
e-mail: cjs@isr.ist.utl.pt

R. Cunha
e-mail: rita@isr.ist.utl.pt

C. Silvestre
Faculty of Science and Technology,
University of Macau, Taipa, Macau

1 Introduction

This paper addresses the problem of low altitude terrain avoidance flight control of autonomous rotorcraft. Within the scope of Unmanned Aerial Vehicles, autonomous rotorcraft have been steadily growing as a major topic of research. They have the potential to perform high precision tasks in challenging and uncertain operation scenarios as new sensor technology and increasingly powerful computational systems are available.

Missions like aerial surveillance, automatic infrastructure inspection (Fig. 1), or 3-D surface mapping in unknown environments demand highly adaptable autonomous vehicles that can meet low altitude flight requirements, thus emphasizing the importance of terrain avoidance strategies.

To address this problem model predictive control (MPC) techniques are adopted. The MPC formulation amounts to solving at each sampling instant a discrete finite horizon optimal control problem, subject to input and state constraints, which results in a sequence of optimal control actions, and applying the first element of this sequence to the plant. When MPC techniques are based on nonlinear models, non-quadratic cost-functionals, and/or general nonlinear constraints, they are referred to as nonlinear model predictive control (NMPC). The basic concepts inherent to MPC techniques can be traced back to the 60s, when Zadeh and Whalen [36] recognized the connections between minimum time optimal control and linear programming, and Propoi [29] introduced the moving horizon approach. Interestingly enough, the first algorithms closely related to the present form of MPC were developed and implemented in the context of the process industry during the 70s and 80s, such as model predictive heuristic control [32], dynamic matrix control [7], model algorithmic control [23] and internal model control [10]. The main focus of the first MPC techniques was the performance achieved in real petro-chemical and process industry applications, gaining popularity due to its natural ability to cope with constraints, nonlinearities and uncertainties. These methods did not ensure stability a priori and required fine tuning of the design parameters, using Monte Carlo simulations to attest for the desired behavior of the overall closed loop system. With few results during the 80s [3, 16], it was in the 90s that the research community devoted considerable effort to the stability analysis of MPC techniques. The main approach was to change the optimization problem, by adding a terminal equality constraint, terminal constraint set, and/or terminal cost functional, yielding a plethora of slightly different methods. Such methods can be found in [1, 25, 31] and also in [4, 8, 15, 21], where the main focus is the suppression of the terminal constraint. Some recent results focus on explicit solutions of the optimization problem [2, 13] or on numerical methods for fast online solution of large scale NMPC problems [37]. As for MPC applications, the vast majority is documented in the process industry, where there is a strong economic drive to push the systems toward their limits of operation. The models used in this kind of applications are inherently nonlinear, constrained and with very large time constants, allowing for the use of large sampling times [11, 30]. With the availability of increasingly faster CPU capabilities in the last decade, MPC techniques are being considered for the control of faster systems. For instance, the control of autonomous vehicles using MPC, as documented in [17, 18, 33, 34], is progressively revealing promising results. Further details

Fig. 1 Helicopter for infrastructure inspection

on the theoretical developments and applications of MPC techniques can be found in [9, 11, 20, 22].

Related work using different control methodologies in the context of terrain following for autonomous helicopters, can be found in [28]. The authors use gain-scheduling control techniques to control and guide the vehicle, in such a way that it is possible to incorporate preview information to achieve terrain following. In [19], an MPC-based nap-of-earth flight trajectory optimization for a helicopter is designed, resorting to an input-output mapping of control functions and resultant system trajectories, while in [35], MPC obstacle avoidance techniques are applied in the field of autonomous ground vehicles.

In this paper a simple NMPC methodology is proposed in order to simultaneously meet the conflicting control objectives of trajectory tracking and terrain avoidance. Since in NMPC the optimal control problem is solved on-line, it is straightforward to add state saturation constraints as a penalty function to the cost functional and the vehicle dynamic model constraint can be readily incorporated in the cost functional using Lagrange multipliers. In an effort to simplify the optimization algorithm, the input saturation is directly incorporated in the nonlinear model instead of being added as a constraint to the optimization problem. To enforce the terrain avoidance, a virtual repulsive field is defined around the helicopter such that any obstacle within its range is weighted in the optimal control cost functional, guiding the vehicle trajectory away from collisions. The resulting unconstrained optimization problem is then numerically solved using the gradient and quasi-Newton methods to compute the search direction and using the Wolfe conditions for the line search algorithm to solve the step size optimization subproblem [26]. The vehicle model considered in this work is a helicopter nonlinear dynamic model, derived from first-principles and specially suited for model-scale helicopters [5, 6]. The control algorithm relies on a simplified version of the referred model to compute the sequence of state vectors given a sequence of input vectors.

The key contribution of this work is the use of simple and well established optimization techniques to solve the terrain avoidance trajectory tracking NMPC problem for a complex nonlinear helicopter model. The comparison of the gradient method with fixed step size and the quasi-Newton method with line search provides evidence that the later optimization approach is a simple yet effective way to reduce the computation burden inherent to NMPC techniques. Furthermore, the intrinsic saturation described in this paper simplifies the controller design by removing the need to include input constraints in the optimization problem.

The paper is organized as follows. Section 2 presents a brief summary of the helicopter dynamic model, including the intrinsic actuation saturation functions, the discretization and the time delay modeling. Section 3 formulates the terrain avoidance NMPC problem, describing the control problem, the optimization algorithm and the saturation, terrain and model constraints. Implementation issues and simulation results are presented in Section 4. Section 5 summarizes the main ideas of this paper and discusses directions for future work. A preliminary version of these results was presented at the 17th IFAC World Congress and can be found in [14].

2 Helicopter Model

This section briefly describes the helicopter dynamic model, as well as the intrinsic input saturation, the discretization and the time delay modeling. A comprehensive coverage of helicopter flight dynamics can be found in [5, 27] and [6].

Consider the helicopter modeled as a rigid body driven by the resultant force and moment applied at the helicopter's center of mass, which include the contributions of the helicopter components and gravitational force. Let $\left({}^{I}\mathbf{p}_B, {}^{I}_{B}\mathbf{R}\right) \in SE(3) \triangleq \mathbb{R}^3 \times SO(3)$ denote the configuration of the body frame $\{B\}$ (attached to the vehicle's center of mass) with respect to the inertial frame $\{I\}$. Consider also the Euler angles vector $\boldsymbol{\lambda}_B = \left[\phi_B\ \theta_B\ \psi_B\right]'$, denoting the orientation of $\{B\}$ relative to $\{I\}$ such that ${}^{I}_{B}\mathbf{R} = \mathcal{R}(\boldsymbol{\lambda}_B)$, where $\theta_B \in (-\frac{\pi}{2}, \frac{\pi}{2}]$ and $\phi_B, \psi_B \in \mathbb{R}$. The linear and angular body velocities are defined as $\mathbf{v}_B$ and

$\boldsymbol{\omega}_B$, respectively, where $\mathbf{v}_B = {}^B_I\mathbf{R}\,{}^I\dot{\mathbf{p}}_B \in \mathbb{R}^3$, $\boldsymbol{\omega}_B = {}^B_I\mathbf{R}\,{}^I\boldsymbol{\omega}_B$ and ${}^I\boldsymbol{\omega}_B$ is the angular velocity of $\{B\}$ relative to $\{I\}$. For the sake of simplicity, the superscript of the position vector is dropped, so that $\mathbf{p}_B = {}^I\mathbf{p}_B$, and the time dependence of the state and input variables is omitted.

Using this notation, the helicopter combined dynamics and kinematics state equations can be written as

$$\begin{cases} \dot{\mathbf{v}}_B = -\boldsymbol{\omega}_B \times \mathbf{v}_B \\ \qquad\quad + \dfrac{1}{m}\left[\mathbf{f}_h\left(\mathbf{v}_B, \boldsymbol{\omega}_B, \mathbf{u}_B\right) + \mathbf{f}_g\left(\phi_B, \theta_B\right)\right] \\ \dot{\boldsymbol{\omega}}_B = -\mathbf{I}^{-1}\left(\boldsymbol{\omega}_B \times \mathbf{I}\,\boldsymbol{\omega}_B\right) + \mathbf{I}^{-1}\,\mathbf{n}_h\left(\mathbf{v}_B, \boldsymbol{\omega}_B, \mathbf{u}_B\right) \\ \dot{\mathbf{p}}_B = \mathcal{R}\left(\boldsymbol{\lambda}_B\right)\,\mathbf{v}_B \\ \dot{\boldsymbol{\lambda}}_B = \mathcal{Q}\left(\boldsymbol{\lambda}_B\right)\,\boldsymbol{\omega}_B \end{cases} \tag{1}$$

In the above equations, m denotes the vehicle mass, $\mathbf{I}$ is the tensor of inertia about the frame $\{B\}$, $\mathbf{u}_B$ is the input vector, $\mathbf{f}_h$ and $\mathbf{n}_h$ are the external force and moment vectors due to the helicopter components, and $\mathbf{f}_g$ stands for the gravitational force vector, all expressed in body coordinates, and $\mathcal{Q}$ is the transformation from angular rates to Euler angle derivatives. The state vector is defined as $\mathbf{x}_B = [\mathbf{v}'_B\ \boldsymbol{\omega}'_B\ \mathbf{p}'_B\ \boldsymbol{\lambda}'_B]'$, noting that $\mathbf{x}_B \in \mathcal{X} \subset \mathbb{R}^{n_x}$ with $n_x = 12$. The input vector, $\mathbf{u}_B \in \mathcal{U} \subset \mathbb{R}^{n_u}$ with $n_u = 4$, is defined as $\mathbf{u}_B = \left[\theta_{c_0}\ \theta_{c_{1c}}\ \theta_{c_{1s}}\ \theta_{c_{0t}}\right]'$ and comprises the main rotor collective input θ_{c_0}, the main rotor cyclic inputs $\theta_{c_{1c}}$ and $\theta_{c_{1s}}$, and the tail rotor collective input $\theta_{c_{0t}}$.

There are five main components on a helicopter that contribute for the overall force and moment vectors: main rotor, tail rotor, fuselage, horizontal tail plane and vertical tail fin, respectively denoted by the subscripts mr, tr, fus, tp and fn. Hence, the force and moment vectors can be decomposed as $\mathbf{f}_h = \mathbf{f}_{mr} + \mathbf{f}_{tr} + \mathbf{f}_{fus} + \mathbf{f}_{tp} + \mathbf{f}_{fn}$ and $\mathbf{n}_h = \mathbf{n}_{mr} + \mathbf{n}_{tr} + \mathbf{n}_{fus} + \mathbf{n}_{tp} + \mathbf{n}_{fn}$.

2.1 Main Rotor

As the primary source of lift, propulsion and control, the main rotor dominates the helicopter dynamic behavior. As a result of the aerodynamic lift forces generated at the surface of its blades, the main rotor is responsible for the helicopter's distinctive ability to operate in low-speed regimes, which include hovering and vertical maneuvering.

To present the main rotor equations of motion, two additional frames need to be introduced. The first denotes the Hub/Wind frame, $\{hw\}$, which has its origin at the hub, x-axis aligned with the component of the helicopter linear velocity relative to the fluid that is parallel to the hub plane, and z-axis aligned with the hub shaft. The second coordinate frame, $\{b\}$, is attached to the blade, with the y-axis aligned with the blade chord and describes rotation, flapping, and pitching motions.

Most of the helicopter maneuvering capabilities result from effectively controlling the main rotor aerodynamic loads. This is achieved by means of the swashplate—a mechanism responsible for applying a different blade pitch angle θ_m at each blade azimuth angle ψ_m, such that $\theta_m(\psi_m) = \theta_{c_0} + \theta_{1c}\cos\psi_m + \theta_{1s}\sin\psi_m$. The collective command θ_{c_0} is directly applied to the main rotor blades, whereas the cyclics θ_{1c} and θ_{1s} result from combining the cyclic commands $\theta_{c_{1c}}$ and $\theta_{c_{1s}}$ with the flapping motion of the Bell-Hiller stabilizing bar, also called flybar. This combined motion can be described by the first order system

$$\begin{aligned} &\Omega\,\mathbf{A}_{\dot{\theta}}\,\dot{\boldsymbol{\theta}}_1 + \Omega^2\,\mathbf{A}_{\theta}(\mu)\,\boldsymbol{\theta}_1 \\ &\quad = \Omega^2\left(\mathbf{B}_{\theta}(\mu)\,\boldsymbol{\theta}_{c_1} + \mathbf{B}_{\omega}\,\boldsymbol{\omega}_B + \mathbf{B}_{\lambda}(\mu)\,\boldsymbol{\lambda}\right), \end{aligned} \tag{2}$$

considering that $\boldsymbol{\lambda} = \left[\mu_z\ -\lambda_0\ \lambda_{1c}\ \lambda_{1s}\right]'$, $\boldsymbol{\theta}_1 = \left[\theta_{1c}\ \theta_{1s}\right]'$, $\boldsymbol{\theta}_{c_1} = \left[\theta_{c_{1c}}\ \theta_{c_{1s}}\right]'$, $\boldsymbol{\omega} = \left[\bar{p}\ \bar{q}\right]'$, and defining $\Omega = \dot{\psi}_m$ as the rotor speed. The variables μ and μ_z are the normalized x and z-components of the hub linear velocity and $\bar{p}$ and $\bar{q}$ are the normalized x and y-components of the angular velocity, all expressed in the frame $\{hw\}$. Detailed expressions for the matrices $\mathbf{A}_{\dot{\theta}}$, $\mathbf{A}_{\theta}(\mu)$, $\mathbf{B}_{\theta}(\mu)$, $\mathbf{B}_{\omega}$, and $\mathbf{B}_{\lambda}(\mu)$ can be found in [5]. Due to this additional dynamic component, the state vector becomes $\mathbf{x}_B = \left[\mathbf{v}'_B\ \boldsymbol{\omega}'_B\ \mathbf{p}'_B\ \boldsymbol{\lambda}'_B\ \boldsymbol{\theta}'_1\right]'$ and $n_x = 14$.

As a result of the thrust force generated at the surface of a rotating blade, the air is accelerated downwards creating a flow field, usually called induced downwash. This flow field can be decomposed in Fourier Series, yielding

$$\lambda(\psi_m) = \lambda_0 + r_m\left(\lambda_{1c}\cos\psi_m + \lambda_{1s}\sin\psi_m\right), \tag{3}$$

where r_m denotes the rotor radius integration variable and the second and higher order terms are neglected. As a consequence of the rotation and pitching motions and their interaction with the motion of the helicopter, the blades describe flap and lag motions. These two blade motions can be roughly characterized by pulling the tip of the blade upwards and backwards, respectively. In this work, the blades are assumed rigid and linked to the hub through flap hinge springs with stiffness k_β, the lag motion is neglected, and the flap motion is approximated by the steady state solution of the Fourier Series expansion without second and higher order terms, resulting in

$$\boldsymbol{\beta} = \mathbf{A}_0^{-1}(\mu)\left[\mathbf{B}_1(\mu)\,\boldsymbol{\theta} + \mathbf{B}_2(\mu)\,\boldsymbol{\omega} + \mathbf{B}_3(\mu)\,\boldsymbol{\lambda}\right] , \quad (4)$$

where $\boldsymbol{\beta} = \left[\beta_0\ \beta_{1c}\ \beta_{1s}\right]'$, $\boldsymbol{\theta} = \left[\theta_{c_0}\ \theta_{1c}\ \theta_{1s}\right]'$, and the matrices $\mathbf{A}_0(\mu)$, $\mathbf{B}_1(\mu)$, $\mathbf{B}_2(\mu)$, and $\mathbf{B}_3(\mu)$ are defined in [5].

The forces and moments generated by the main rotor are the sum of the contributions of each blade expressed in the hub frame. The contribution of the main rotor to the total force acting on the helicopter can be written as $\mathbf{f}_{mr} = {}^{B}_{hw}\mathbf{R}\,{}^{hw}\mathbf{f}_{mr}$, with the expression for ${}^{hw}\mathbf{f}_{mr}$ given by

$${}^{hw}\mathbf{f}_{mr} \simeq \frac{n_b}{2}\begin{bmatrix} -Y_{1s} \\ -Y_{1c} \\ 2\,Z_0 \end{bmatrix} + \frac{n_b}{2}\begin{bmatrix} -Z_{1c} & -Z_0 - \dfrac{Z_{2c}}{2} & -\dfrac{Z_{2s}}{2} \\ Z_{1s} & \dfrac{Z_{2s}}{2} & Z_0 - \dfrac{Z_{2c}}{2} \\ 0 & 0 & 0 \end{bmatrix}\boldsymbol{\beta} , \quad (5)$$

where n_b is the number of blades. The terms $Y_{(.)}$ and $Z_{(.)}$ are the components of the Fourier Series decomposition of the aerodynamic force generated at each blade. Similarly, the main rotor contribution to the overall moment is computed using $\mathbf{n}_{mr} = {}^{B}_{hw}\mathbf{R}\,{}^{hw}\mathbf{n}_{mr} + {}^{B}\mathbf{p}_{mr} \times \mathbf{f}_{mr}$, where ${}^{B}\mathbf{p}_{mr}$ denotes the position of the main rotor relative to the body frame, and the expression for ${}^{hw}\mathbf{n}_{mr}$ can be written as

$${}^{hw}\mathbf{n}_{mr} \simeq n_b \begin{bmatrix} 0 \\ 0 \\ N_0 \end{bmatrix} + \frac{n_b}{2}\begin{bmatrix} -N_{1c} & -N_0 - \dfrac{N_{2c}}{2} & -k_\beta - \dfrac{N_{2s}}{2} \\ N_{1s} & -k_\beta + \dfrac{N_{2s}}{2} & N_0 - \dfrac{N_{2c}}{2} \\ 0 & 0 & 0 \end{bmatrix}\boldsymbol{\beta} , \quad (6)$$

noting that the terms $N_{(.)}$ are the components of the Fourier Series decomposition of the aerodynamic yaw moment generated at each blade.

2.2 The Other Components

The tail rotor, placed at the tail boom in order to counteract the moment generated by the rotation of the main rotor, provides yaw control of the helicopter. The same principles adopted for the main rotor can be used to model the tail rotor, further neglecting the blade flapping and pitching motions, which have little significance for small rotor diameters. The tail rotor contribution to the total force can be approximated by

$$\mathbf{f}_{tr} = {}^{B}_{tr}\mathbf{R}\,{}^{tr}\mathbf{f} \simeq \left[0\ -n_{b_t}\,Z_{0_t}\ 0\right]' , \quad (7)$$

where n_{b_t} is the number of blades of the tail rotor, Z_{0_t} is the thrust force produced by each blade of the tail rotor and ${}^{B}_{tr}\mathbf{R}^{\,tr}$ is the rotation from the tail rotor frame $\{tr\}$ to the body frame $\{B\}$. Similarly, the moment expression is given by

$$\mathbf{n}_{tr} = \left[0\ -n_{b_t}\,N_{0_t}\ 0\right]' + {}^{B}\mathbf{p}_{tr} \times \mathbf{f}_{tr} , \quad (8)$$

where N_{0_t} is the torque generated by each blade of the tail rotor and ${}^{B}\mathbf{p}_{tr}$ denotes the position of the tail rotor relative to the body frame.

Accurate modeling of the aerodynamic forces and moments generated by the flow surrounding the helicopter fuselage is a demanding task. In this work these loads are modeled as functions of the mean flow speed v_{fus}, the incidence angle α_{fus} and the side-slip angle β_{fus}. The horizontal tail plane and vertical tail fin are modeled as normal wings, whose aerodynamic force contributions can be

approximated by functions of the angle of attack and sideslip, respectively.

2.3 Intrinsic Input Saturation

Within the MPC literature, input saturation constraints are included in the optimization problem to guarantee that the resulting control sequence is always admissible. However, when using NMPC these and other complex physical constraints can be defined implicitly in the nonlinear model of the plant, instead of explicitly in the optimization problem. The rationale is to separate between basic control saturations, corresponding to physical limitations of the platform or model, and mission dependent control saturations, where the former are made intrinsic to the nonlinear model and the latter are defined as an additional constraint to the optimization problem. This procedure simplifies the optimization problem and yields a self contained model of the plant that is valid for any inputs.

The autonomous helicopter dynamic model described in Eq. 1 can be written as

$$\dot{\mathbf{x}}_B(t) = \mathbf{f}_B\left(\mathbf{x}_B(t), \mathbf{u}_B(t)\right) \ . \tag{9}$$

Let the new input $\bar{\mathbf{u}}_B$ be defined as a smoothly saturated function of the regular input $\mathbf{u}_B$, so that the dynamic equation is now given by $\dot{\mathbf{x}}_B = \mathbf{f}_B\left(\mathbf{x}_B, \bar{\mathbf{u}}_B(\mathbf{u}_B)\right)$. In brief, considering that $\mathbf{u}_B \in \mathcal{U}$, the procedure described below defines the new saturated input vector as $\bar{\mathbf{u}}_B \in \mathbb{R}^{n_u}$, simplifying the optimization problem formulation.

The saturation functions are derived from a basic function $\bar{a}(a) = \frac{a}{1+|a|}$, applying translation and scaling operations both to the function and its derivative, such that inside the bounds $\bar{a} = a$ and outside the bounds $\bar{a}$ tends smoothly to the maximum value $a_{\max}$ or to the minimum value $a_{\min}$. The generic saturation function used in this work is defined as

$$\bar{a}(a) = \begin{cases} a & , \ \varepsilon_{\max} \le a \le \varepsilon_{\min} \\ \varepsilon_{\max} + \dfrac{a - \varepsilon_{\max}}{1 + \frac{a-\varepsilon_{\max}}{\varepsilon}} & , \ a > \varepsilon_{\max} \\ \varepsilon_{\min} + \dfrac{a - \varepsilon_{\min}}{1 - \frac{a-\varepsilon_{\min}}{\varepsilon}} & , \ a < \varepsilon_{\min} \end{cases} , \tag{10}$$

with $\varepsilon_{\max} = a_{\max} - \varepsilon$, $\varepsilon_{\min} = a_{\min} + \varepsilon$, where $0 < \varepsilon < 1$ is a constant, that defines the length of the smooth transition. The derivative of this function is defined as

$$\frac{\mathrm{d}\,\bar{a}}{\mathrm{d}a} = \begin{cases} 1 & , \ \varepsilon_{\max} \le a \le \varepsilon_{\min} \\ \dfrac{1}{\left(1 + \frac{a-\varepsilon_{\max}}{\varepsilon}\right)^2} & , \ a > \varepsilon_{\max} \\ \dfrac{1}{\left(1 - \frac{a-\varepsilon_{\min}}{\varepsilon}\right)^2} & , \ a < \varepsilon_{\min} \end{cases} . \tag{11}$$

For simplicity and with a slight abuse of notation, the input vector $\mathbf{u}_B$ is used hereafter to denote the result of the smooth saturation, $\bar{\mathbf{u}}_B$.

2.4 Discretization and Delay Modeling

In the NMPC approach used in this paper it is necessary to find a discrete representation of the equations of motion of the vehicle. There are several methods available, with different complexity and integration errors, from which the simplest is to use the forward Euler discretization, resulting in the difference equation

$$\mathbf{x}_B((k+1)T_s) = \mathbf{x}_B(kT_s) + T_s\mathbf{f}_B\left(\mathbf{x}_B(kT_s), \mathbf{u}_B(kT_s)\right) \ , \tag{12}$$

where T_s is the sampling interval. The previous equation can be rewritten using a compact notation as

$$\mathbf{x}_{B_{k+1}} = \mathbf{f}_d\left(\mathbf{x}_{B_k}, \mathbf{u}_{B_k}\right) \ . \tag{13}$$

The intensive computational requirements of NMPC techniques discard the possibility of neglecting the processing time in comparison with the sampling interval. A fair assumption is to consider that the time needed for the computation of the control law is smaller than the sampling interval. In recent literature, some new algorithms are proposed to tackle this problem, such as the advanced step algorithm described in [37]. For simplicity, the classic approach is used in this work, which considers that the delay between the instant the state variables are measured and the instant when new control action is applied coincides with the sampling period T_s. The model is augmented with an extra delay state yielding the new state vector $\mathbf{x}_k = [\mathbf{x}'_{B_k} \ \mathbf{x}'_{u_k}]'$, the input vector

is defined as $\mathbf{u}_k = \mathbf{u}_{B_k}$, and the model function can be denoted as

$$\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) = \begin{bmatrix} \mathbf{f}_d\left(\mathbf{x}_{B_k}, \mathbf{x}_{u_k}\right) \\ \mathbf{u}_{B_k} \end{bmatrix} , \tag{14}$$

resulting in the discrete-time model

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \ . \tag{15}$$

3 Model Predictive Control Problem

To address the two conflicting objectives of trajectory tracking and terrain avoidance, the NMPC problem is formulated as a discrete-time open-loop optimal control problem with finite horizon, subject to the discrete nonlinear dynamic model equations, the saturation constraints, and the terrain avoidance constraint.

Recalling Eq. 15, the vehicle dynamics can be modeled as a discrete-time state-space equation with state $\mathbf{x}_k \in \mathcal{X}$ and input $\mathbf{u}_k \in \mathcal{U}$, where $\mathcal{X} \subset \mathbb{R}^{n_x}$ and $\mathcal{U} \subset \mathbb{R}^{n_u}$ denote the feasibility sets of the state and control vectors, respectively. Let N be the prediction horizon of the control problem, $\mathbf{U}_k = \{\mathbf{u}_k, \ldots, \mathbf{u}_{k+N-1}\}$ the sequence of control inputs at iteration k, and $\mathbf{X}_k = \{\mathbf{x}_k, \ldots, \mathbf{x}_{k+N}\}$ the sequence of state vectors generated by that control sequence. Further note that the state sequence is a function of the initial state vector and the control sequence, i. e. $\mathbf{X}_k = \mathbf{X}(\mathbf{x}_k, \mathbf{U}_k)$.

The saturation constraints of the state and input sequences are defined by the conditions $\mathbf{X}_k \in \boldsymbol{\mathcal{X}}_N$ and $\mathbf{U}_k \in \boldsymbol{\mathcal{U}}_N$, considering the sets $\boldsymbol{\mathcal{X}}_N = \{\mathbf{X}_k : \mathbf{x}_i \in \mathcal{X}, \forall_{i=k,\ldots,k+N}\}$ and $\boldsymbol{\mathcal{U}}_N = \{\mathbf{U}_k : \mathbf{u}_i \in \mathcal{U}, \forall_{i=k,\ldots,k+N-1}\}$. Using Eq. 15, the model constraint for an horizon of N steps ahead can be written as

$$\mathbf{f}_M(\mathbf{X}_k, \mathbf{U}_k) = \begin{bmatrix} \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) - \mathbf{x}_{k+1} \\ \vdots \\ \mathbf{f}(\mathbf{x}_{k+N-1}, \mathbf{u}_{k+N-1}) - \mathbf{x}_{k+N} \end{bmatrix} = \mathbf{0} \ . \tag{16}$$

The terrain constraint is denoted by

$$\mathbf{f}_T(\mathbf{X}_k) = \begin{bmatrix} F_T(\mathbf{x}_k) \\ \vdots \\ F_T(\mathbf{x}_{k+N}) \end{bmatrix} = \mathbf{0} \ , \tag{17}$$

where $F_T(.)$ weights the distance between the vehicle and the terrain such that, as discussed below, $F_T(\mathbf{X}_k)$ goes exponentially fast to zero when the distance increases. Given these constraints, the NMPC problem can be defined as

$$\mathbf{U}_k^* = \arg\min_{\mathbf{U}_k} J_{N,k} \tag{18}$$

$$s.t. \quad \mathbf{X}_k \in \boldsymbol{\mathcal{X}}_N \ , \ \mathbf{U}_k \in \boldsymbol{\mathcal{U}}_N \tag{19}$$

$$\mathbf{f}_M(\mathbf{X}_k, \mathbf{U}_k) = \mathbf{0} \tag{20}$$

$$\mathbf{f}_T(\mathbf{X}_k) = \mathbf{0} \tag{21}$$

where

$$J_{N,k} = J_N(\mathbf{X}_k, \mathbf{U}_k) = F_{k+N} + \sum_{i=k}^{k+N-1} L_i \ , \tag{22}$$

$$F_i = F(\mathbf{x}_i) = \frac{1}{2} (\mathbf{x}_i - \bar{\mathbf{x}}_i)' \mathbf{P} (\mathbf{x}_i - \bar{\mathbf{x}}_i) \ , \tag{23}$$

$$L_i = L(\mathbf{x}_i, \mathbf{u}_i) = \frac{1}{2} \left[(\mathbf{x}_i - \bar{\mathbf{x}}_i)' \mathbf{Q} (\mathbf{x}_i - \bar{\mathbf{x}}_i) + (\mathbf{u}_i - \bar{\mathbf{u}}_i)' \mathbf{R} (\mathbf{u}_i - \bar{\mathbf{u}}_i) \right] \tag{24}$$

noting that $\bar{\mathbf{x}}_i$ for all $i = k, \ldots, k+N$ and $\bar{\mathbf{u}}_i$ for all $i = k, \ldots, k+N-1$ are the known reference state and input vector sequences, respectively, and $\mathbf{P}$, $\mathbf{Q}$, and $\mathbf{R}$ are symmetric positive definite matrices to be defined in the simulation results section. In brief, the NMPC objective is to find, at each iteration k, the optimal control sequence $\mathbf{U}_k^*$ with horizon N, such that the distance to the full-state trajectory and the control effort are minimized through the cost functional $J_{N,k}$, without violating the state and input constraints defined in Eq. 19 and keeping the vehicle within a safety distance from the terrain.

The constrained optimization problem presented above can be reformulated as an unconstrained optimization problem and gradient methods can be used to estimate the optimal solution. While constraint 20 is eliminated using Lagrange multipliers, constraints 19 and 21 are incorporated in the cost functional resorting to penalty methods. In the following subsections each constraint function and respective inclusion in the unconstrained optimization cost functional are described.

3.1 Terrain Avoidance Constraint

The terrain constraint function introduced in Eq. 17 is designed so as to enable a collision free trajectory, even if the reference trajectory goes through the terrain. This constraint can be implemented by defining a repulsive field around the origin of the body frame $\{B\}$, obtained by weighting exponentially the minimum distance between the helicopter position $\mathbf{p}$ and the closest point on the terrain, $\mathbf{p}_m(\mathbf{p})$, which is a function of the position of the helicopter. Consider the position error between these two points to be defined as $\tilde{\mathbf{p}} = \mathbf{p} - \mathbf{p}_m$ and let r_S be the radius of a safety sphere around the helicopter, such that if $\tilde{\mathbf{p}}'\,\tilde{\mathbf{p}} > r_S^2$ then the vehicle is sufficiently distant from the terrain. A function $g(\mathbf{x})$ that gives a measure of distance between the terrain and the safety sphere can be defined as

$$g\,(\mathbf{x}) = \tilde{\mathbf{p}}'\,\tilde{\mathbf{p}} - r_S^2\,, \tag{25}$$

such that $g\,(\mathbf{x}) = 0$ when the terrain touches the safety sphere. The terrain avoidance constraint function can be written as

$$F_T(\mathbf{x}) = c\,\mathrm{e}^{-g(\mathbf{x})}\,, \tag{26}$$

where $c > 0$ is a scalar weight selected such that, when $g(\mathbf{x}) = 0$ the respective value of the terrain constraint, $F_T(\mathbf{x}) = c$, is adequate to drive the vehicle away from the terrain in the foreseen operation scenarios. The zero value is forced whenever the distance $\|\tilde{\mathbf{p}}\|$ is greater than a predefined outer radius r_O where the influence of the terrain is negligible, that is $F_T = 0$ if $\|\tilde{\mathbf{p}}\| > r_O$, thus avoiding unnecessary computational burden. Note that even if this procedure is not performed, the deviation from the desired trajectory will be negligible, since the terms L_i of the cost function will always be dominant when the vehicle is not close to the terrain. The derivative of this terrain constraint is computed using

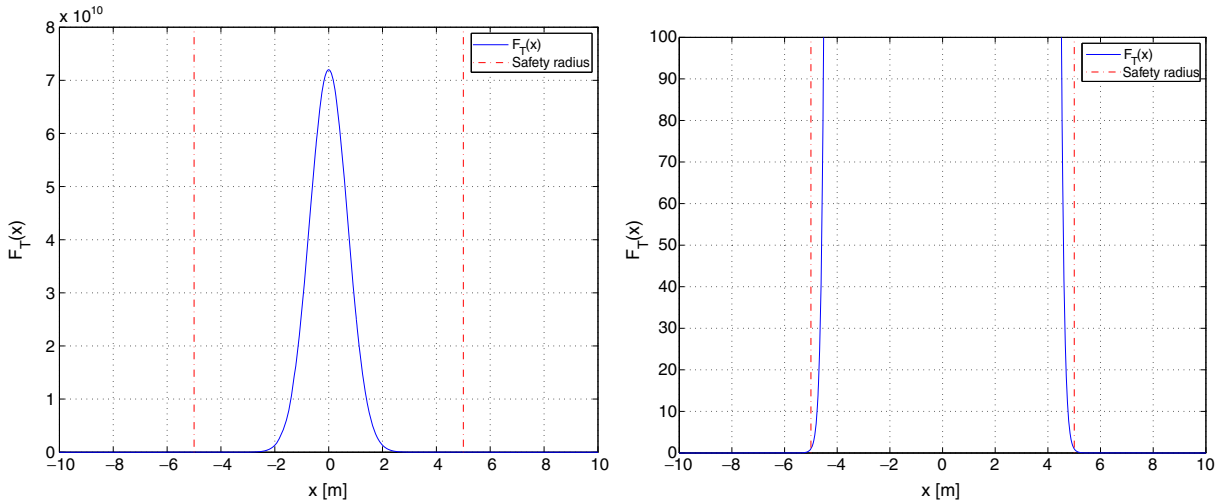
$$\frac{\partial\, F_T}{\partial \mathbf{x}'} = -2\,c\,\tilde{\mathbf{p}}'\,\mathrm{e}^{-g(\mathbf{x})}.\,\frac{\partial\,\tilde{\mathbf{p}}}{\partial \mathbf{x}'} \tag{27}$$

and

$$\frac{\partial\,\tilde{\mathbf{p}}}{\partial \mathbf{x}'} = \left[\mathbf{0}_{3\times 3}\;\mathbf{0}_{3\times 3}\;\frac{\partial\,\tilde{\mathbf{p}}}{\partial \mathbf{p}'}\;\mathbf{0}_{3\times 3}\right]'\,, \tag{28}$$

where the derivative $\frac{\partial\,\tilde{\mathbf{p}}}{\partial \mathbf{p}'}$ is computed numerically, based on the available information of the terrain.

The values used throughout this work for the constants described earlier are $c = 1$, $r_0 = 10m$, and $r_S = 5m$. An example of a one-dimension version of this function can be seen in Fig. 2. This type of function has the advantage of being defined for all $\mathbf{x} \in \mathbb{R}^{n_x}$, while the ubiquitous infinity barrier type of constraint functions, applied to this safety sphere concept, may lead to problems during ini-



(a) Global view (b) Detail on safety sphere boundary

Fig. 2 One-dimensional function $c\,\mathrm{e}^{-g(x)}$, with $g(x) = x^2 - 5^2$ and $c = 1$

tialization or when avoiding the terrain. For instance, for takeoff maneuvers, the vehicle would be initialized with the terrain inside the safety sphere and the proposed terrain constraint alone would drive its trajectory away from the terrain, whereas the infinity barrier constraint is not coherently defined for such a situation and would have the opposite effect of keeping the vehicle close to the terrain.

3.2 State and Input Saturation Constraint

The saturation constraints defined in Eq. 19 are included in the optimization problem to complement the intrinsic input saturation described in Section 2.3, enabling the definition of mission specific bounds for the state and input vectors. If there is no special requirement in terms of actuation there is no need to have input saturations in the optimization problem, since the physical constraints will be always satisfied through the intrinsic saturations. On the other hand, a mission where the payload includes sensitive equipment may require all actuation signals to be further constrained into a smaller actuation set, thus avoiding large acceleration maneuvers.

These constraints can be incorporated in the cost functional as a penalty function, $F_R(\mathbf{x}, \mathbf{u})$, which is zero-valued for $\mathbf{x} \in \mathcal{X}$ and $\mathbf{u} \in \mathcal{U}$, and behaves as a quadratic function outside these sets. Assuming that the feasibility sets for state and input vectors are given by $\mathcal{X} = \left\{\mathbf{x} \in \mathbb{R}^{n_x} : x^{(j)}_{\min} \leq x^{(j)} \leq x^{(j)}_{\max}\ \forall_{j=1,\ldots,n_x}\right\}$ and $\mathcal{U} = \left\{\mathbf{u} \in \mathbb{R}^{n_u} : u^{(l)}_{\min} \leq u^{(l)} \leq u^{(l)}_{\max}\ \forall_{l=1,\ldots,n_u}\right\}$, respectively, the penalty function can be defined as

$$F_R(\mathbf{x}, \mathbf{u}) = \sum_{j=1}^{n_x} f_R(x^{(j)}) + \sum_{l=1}^{n_u} f_R(u^{(l)})\ , \tag{29}$$

where

$$f_R(a) = \frac{1}{2}\, h^2(|a - a_{\text{center}}| - a_{\text{range}})\, w_a\ , \tag{30}$$

w_a is a positive scalar weight, $a_{\text{center}} = (a_{\max} + a_{\min})/2$, $a_{\text{range}} = a_{\max} - a_{\text{center}}$, and

$$h(a) = \begin{cases} a & , \text{if } a > 0 \\ 0 & , \text{otherwise} \end{cases}\ . \tag{31}$$

The derivative of this constraint is computed analytically using the derivative of the generic function $f_R(a)$, given by

$$\frac{\mathrm{d}\, f_R(a)}{\mathrm{d}a} = \text{sign}(a - a_{\text{center}}) \times h(|a - a_{\text{center}}| - a_{\text{range}})\, w_a\ . \tag{32}$$

3.3 Unconstrained Optimization Problem

The unconstrained optimization control problem can be defined as

$$\mathbf{U}^*_k = \arg\min_{\mathbf{U}_k} \bar{J}_{N,k} \tag{33}$$

$$s.t. \quad \mathbf{f}_M(\mathbf{X}_k, \mathbf{U}_k) = \mathbf{0} \tag{34}$$

where the terrain and saturation constraints are incorporated in the new cost functional by using

$$\bar{J}_{N,k} = \bar{J}_N(\mathbf{X}_k, \mathbf{U}_k) = \bar{F}_{k+N} + \sum_{i=k}^{k+N-1} \bar{L}_i\ , \tag{35}$$

$$\bar{F}_i = \bar{F}(\mathbf{x}_i) = F_i + F_R(\mathbf{x}_i, \mathbf{0}) + F_T(\mathbf{x}_i)\ , \tag{36}$$

$$\bar{L}_i = \bar{L}(\mathbf{x}_i, \mathbf{u}_i) = L_i + F_R(\mathbf{x}_i, \mathbf{u}_i) + F_T(\mathbf{x}_i)\ , \tag{37}$$

and the model constraint 34 is solved by the elimination method using Lagrange multipliers. Introducing the Lagrange multiplier sequence $\Lambda_k = \{\boldsymbol{\lambda}_{k+1}, \ldots, \boldsymbol{\lambda}_{k+N}\}$ and the Hamiltonian $H_i = H(\mathbf{x}_i, \mathbf{u}_i) = \bar{L}_i + \boldsymbol{\lambda}'_{i+1}\, \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i)$, the cost functional can be rewritten as

$$\bar{J}_{N,k} = \bar{F}_{k+N} - \boldsymbol{\lambda}'_{k+N}\, \mathbf{x}_{k+N} + \sum_{i=k+1}^{k+N-1} \left[H_i - \boldsymbol{\lambda}'_i\, \mathbf{x}_i\right] + H_k. \tag{38}$$

For a fixed initial state $\mathbf{x}_k$, the first order condition of optimality yields

$$\frac{\partial\, \bar{J}_{N,k}}{\partial \mathbf{x}_i} = \frac{\partial\, H_i}{\partial \mathbf{x}_i} - \boldsymbol{\lambda}'_i = \mathbf{0}\ ,\quad \forall_{i=k+1,\ldots,k+N-1}\ , \tag{39}$$

$$\frac{\partial\, \bar{J}_{N,k}}{\partial \mathbf{x}_{k+N}} = \frac{\partial\, \bar{F}_{k+N}}{\partial \mathbf{x}_{k+N}} - \boldsymbol{\lambda}'_{k+N} = \mathbf{0}\,, \tag{40}$$

$$\frac{\partial\, \bar{J}_{N,k}}{\partial_i} = \frac{\partial\, H_i}{\partial \mathbf{u}_i} = \mathbf{0}\ ,\quad \forall_{i=k,\ldots,k+N-1}\ , \tag{41}$$

where $\frac{\partial H_i}{\partial \mathbf{u}_i} = \frac{\partial \bar{L}_i}{\partial \mathbf{u}_i} + \boldsymbol{\lambda}'_{i+1} \frac{\partial \mathbf{f}_i}{\partial \mathbf{u}_i}$ and $\frac{\partial H_i}{\partial \mathbf{x}_i} = \frac{\partial \bar{L}_i}{\partial \mathbf{x}_i} + \boldsymbol{\lambda}'_{i+1} \frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_i}$. The lagrange multipliers are defined as

$$\boldsymbol{\lambda}'_{k+N} = \frac{\partial \bar{F}_{k+N}}{\partial \mathbf{x}_{k+N}} \text{ and } \boldsymbol{\lambda}'_i = \frac{\partial H_i}{\partial \mathbf{x}_i}, \quad \forall_{i=k+1,\ldots,k+N-1}, \tag{42}$$

so that the first order conditions of optimality reduce to Eq. 41.

An iterative algorithm based on the first order gradient method can be applied to estimate $\mathbf{U}_k^*$, updating the control sequence at each iteration according to

$$\mathbf{U}_k \leftarrow \mathbf{U}_k + s\,\boldsymbol{\Delta}_k\,, \tag{43}$$

where the step size is denoted by s and the search direction by $\boldsymbol{\Delta}_k$. The optimization algorithm can be summarized as follows.

Algorithm 1 Minimization algorithm for the NMPC unconstrained problem.

1. Initialize $\mathbf{X}_k, \mathbf{U}_k$;
2. Compute $\{\boldsymbol{\lambda}_i\}, i = k+N, \ldots, k$;
3. Compute $\left\{\frac{\partial H_i}{\partial \mathbf{u}_i}\right\}, i = k, \ldots, k+N-1$;
4. Compute the search direction $\boldsymbol{\Delta}_k$;
5. Compute the step size s;
6. Compute the new $\mathbf{U}_k$ using Eq. 43 and the new $\mathbf{X}_k = \{\mathbf{x}_i\}$ using $\mathbf{x}_{i+1} = \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i)$, for $i = k+1, \ldots, k+N$;
7. Test stop conditions: if false go to step (2), if true let $\hat{\mathbf{U}}_k = \mathbf{U}_k$ be the final estimate, apply first input vector $\hat{\mathbf{u}}_k$ to system and set the next initial solution to $\mathbf{U}_k \leftarrow \{\hat{\mathbf{u}}_{k+1}, \ldots, \hat{\mathbf{u}}_{k+N-1}, \hat{\mathbf{u}}_{k+N-1}\}$.

The search direction is denoted by the sequence $\boldsymbol{\Delta}_k = \{\boldsymbol{\delta}_k, \ldots, \boldsymbol{\delta}_{k+N-1}\}$ and can be obtained using either the gradient or the quasi-Newton methods, given respectively by

$$\boldsymbol{\delta}_i = -\frac{\partial H_i}{\partial \mathbf{u}_i} \quad \text{and} \quad \boldsymbol{\delta}_i = -D_i \frac{\partial H_i}{\partial \mathbf{u}_i}\,, \tag{44}$$

where the matrices D_i are estimates for the inverse second-order derivatives $\frac{\partial^2 H_i}{\partial \mathbf{u}_i \partial \mathbf{u}_i}$, computed as in [26].

The line search optimization subproblem, used to estimate the optimal step size s^*, is numerically solved using the Wolfe rule. This approach guarantees a decrease of the cost functional, as the well known Armijo rule does, and ensures reasonable progress by ruling out unacceptably short steps [26]. Consider the step size optimization subproblem defined as

$$s^* = \arg\min_{s \geq 0} \phi(s)\,, \tag{45}$$

where the cost functional is $\phi(s) = \bar{J}^+_{N,k} = \bar{J}_N(\mathbf{X}_k^+, \mathbf{U}_k^+)$, with $\mathbf{U}_k^+ = \mathbf{U}_k + s\,\boldsymbol{\Delta}_k$, $\mathbf{X}_k^+ = \mathbf{X}(\mathbf{x}_k, \mathbf{U}_k + s\,\boldsymbol{\Delta}_k)$, and the first order condition of optimality is given by

$$\frac{\mathrm{d}\,\phi(s)}{\mathrm{d}s} = \frac{\mathrm{d}\,\bar{J}^+_{N,k}}{\mathrm{d}s} = 0\,. \tag{46}$$

The superscript $(.)^+$ denotes the result of updating a variable using the step size s. Noting that $\bar{J}_{N-i,k+i} = \bar{L}_{k+i} + \bar{J}_{N-i-1,k+i+1}$ for all $i = 0, \ldots, N-1$, the derivative $\frac{\mathrm{d}\,\bar{J}^+_{N,k}}{\mathrm{d}s}$ can be defined using

$$\frac{\mathrm{d}\,\bar{J}^+_{N-i,k+i}}{\mathrm{d}s} = \frac{\partial\,\bar{L}^+_{k+i}}{\partial \mathbf{x}_{k+i}}\,\boldsymbol{\eta}_{k+i} + \frac{\partial\,\bar{L}^+_{k+i}}{\partial \mathbf{u}_{k+i}}\,\boldsymbol{\delta}_{k+i} + \frac{\mathrm{d}\,\bar{J}^+_{N-i-1,k+i+1}}{\mathrm{d}s}\,, \tag{47}$$

$$\frac{\mathrm{d}\,\bar{J}^+_{0,k+N}}{\mathrm{d}s} = \frac{\partial\,\bar{F}^+_{k+N}}{\partial \mathbf{x}_{k+N}}\,\boldsymbol{\eta}_{k+N}\,, \tag{48}$$

$$\boldsymbol{\eta}_{k+i} = \frac{\mathrm{d}\,\mathbf{x}^+_{k+i}}{\mathrm{d}s} = \frac{\partial\,\mathbf{f}^+_{k+i-1}}{\partial \mathbf{x}_{k+i-1}}\,\boldsymbol{\eta}_{k+i-1} + \frac{\partial\,\mathbf{f}^+_{k+i-1}}{\partial \mathbf{u}_{k+i-1}}\,\boldsymbol{\delta}_{k+i-1}\,, \tag{49}$$

Let also $\mu_1 = \phi(0) + \sigma_1 \frac{\mathrm{d}\,\phi(0)}{\mathrm{d}s}\,s$ and $\mu_0 = \sigma_0 \frac{\mathrm{d}\,\phi(0)}{\mathrm{d}s}$, where σ_1 and σ_0 are parameters of the search algorithm. The Wolfe conditions classify a step size according to the sets

$$\begin{aligned}
\mathcal{A} &= \left\{s > 0 : \phi(s) \leq \mu_1 \ \wedge\ \frac{\mathrm{d}\,\phi(s)}{\mathrm{d}s} \geq \mu_0\right\} \\
\mathcal{D} &= \{s > 0 : \phi(s) > \mu_1\} \\
\mathcal{E} &= \left\{s > 0 : \phi(s) \leq \mu_1 \ \wedge\ \frac{\mathrm{d}\,\phi(s)}{\mathrm{d}s} < \mu_0\right\}\,,
\end{aligned} \tag{50}$$

that define the acceptable, the right unacceptable, and the left unacceptable step sizes, respectively. By using a basic search algorithm an acceptable

step size is obtained, yielding an estimate of the optimal step size.

4 Simulation Results

The terrain avoidance NMPC algorithm presented in this work is designed to provide the unmanned rotorcraft with low altitude flight capabilities even in operation scenarios where the predefined trajectory collides with the terrain. In this section the performance of the proposed method is evaluated in simulation.

Since the problem of terrain acquisition and representation is not the main focus of this paper, it is assumed that the terrain is represented by an elevation function and that this information enters the control algorithm in the computation of the helicopter-terrain distance. In the results presented hereafter, the helicopter model described in Section 2 is parameterized for the Vario X-treme model scale helicopter and used to close the simulation loop [5]. A simplified, yet highly nonlinear, version of this model is used in the control algorithm. Full analytical expressions for the first order derivatives of this model were obtained in order to efficiently compute the actuation at each sampling time. Simulations were carried out in Matlab®/Simulink® with C mex-functions, using an Intel® Core™2 T9550 at 2.66GHz with 3GB of RAM running Ubuntu 10.10 operative system.

The simulation results presented in this section are two-fold. Firstly, a terrain avoidance simulation is presented to demonstrate the capabilities of the algorithms in collision avoidance situations and to compare the performance of each optimization algorithm. Secondly, a shorter simulation is presented to illustrate particular aspects of this NMPC, such as the natural previewing ability of MPC strategies or the activation of intrinsic saturations and saturation constraints included in the optimization problem.

4.1 Terrain Avoidance Results

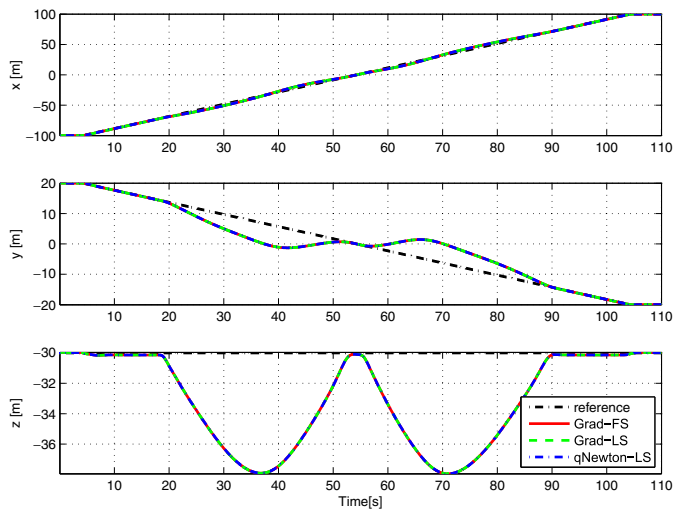
The terrain used in the first simulation resembles a winding water stream and the reference trajectory comprises three sections: (1) hovering flight at the initial position; (2) forward flight trajectory with constant speed $\|\mathbf{v}\| = 2\ \mathrm{ms}^{-1}$; and (3) hovering flight at the final position. The sample time is $T_s = 0.02$ s and the horizon is $N = 50$ sample times, or equivalently 1 s. This horizon is sufficiently large to allow for the algorithm to predict impacts with the terrain and change the helicopter trajectory to avoid it. The precision of the solution is determined by the algorithm stop condition given by $\|\nabla_{\mathbf{U}_k} \bar{J}_{N,k}\| / \|\nabla_{\mathbf{U}_k} \bar{J}^{(0)}_{N,k}\| < 10^{-3}$, where $\bar{J}^{(0)}_{N,k}$ is the initial value of the cost functional. Noting that the final state vector is defined as $\mathbf{x}_k = \left[\mathbf{v}'_{B_k}\ \boldsymbol{\omega}'_{B_k}\ \mathbf{p}'_{B_k}\ \boldsymbol{\lambda}'_{B_k}\ \boldsymbol{\theta}'_{1_k}\ \mathbf{u}'_{B_{k-1}}\right]'$, the cost functional matrices used in the simulation results are

$$\mathbf{R} = \mathrm{diag}(150,\ 300,\ 300,\ 150)\ , \tag{51}$$

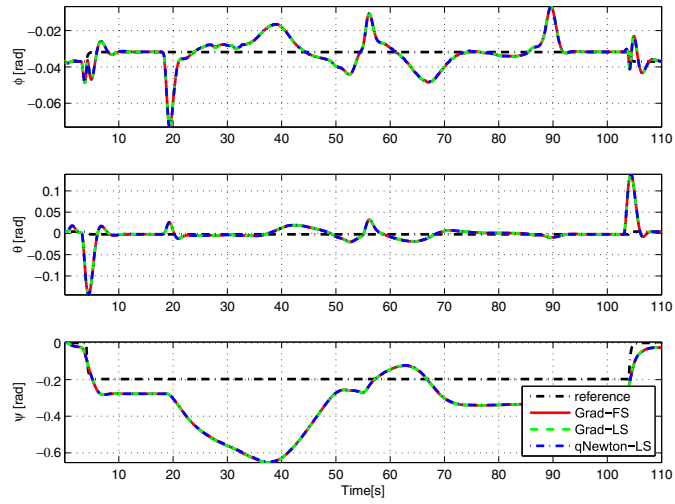
$$\mathbf{Q} = \mathrm{diag}(\mathbf{I}_3,\ 3\,\mathbf{I}_2,\ 10,\ 5\,\mathbf{I}_3,\ \mathbf{I}_2,\ 10,\ \mathbf{I}_2,\ \mathbf{R})\ , \tag{52}$$

and the $\mathbf{P}$ matrix is computed as the steady-state solution of the discrete-time algebraic Riccati equation using matrices $\mathbf{Q}$ and $\mathbf{R}$, for the helicopter model linearized around hover. The operator diag(.) stands for the block diagonal matrix formed by the arguments, and $\mathbf{I}_n$ is the $n \times n$ identity matrix.

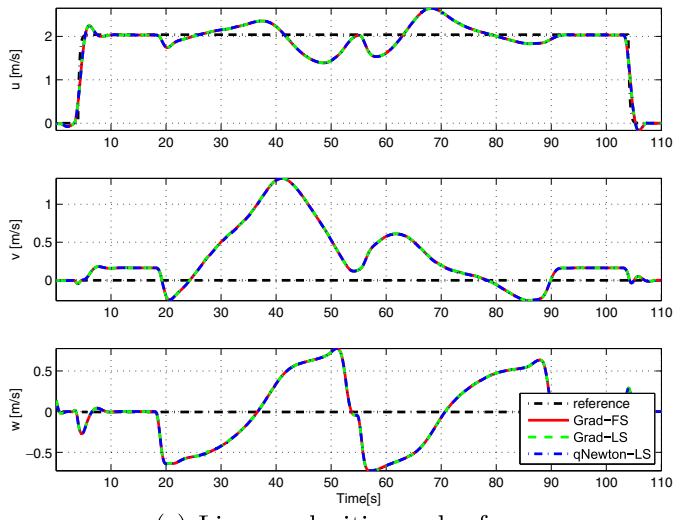
The simulation results presented in Figs. 3 and 4, feature the time evolution of the position, Euler angles, linear velocity, actuation and the 3-D trajectory described by the helicopter. These results include the values for each of the three optimization algorithms: gradient method with fixed step (Grad-FS), gradient method with line search (Grad-LS); and quasi-Newton method with line search (qNewton-LS). For the Grad-FS algorithm, the value $s = 10^{-6}$ was chosen for the step size so that the algorithm converges at each instant of the desired simulation with a good rate of convergence. Since the optimization problem and the termination tolerance are the same for each of these algorithms, the performance in terms of minimization of the cost function should be the same, apart from numerical issues. Note that the cost function minimization includes several terms: trajectory following, terrain avoidance, control effort, and saturations. It can be seen in the simulation plots that all the variables of the three



(a) Position and reference



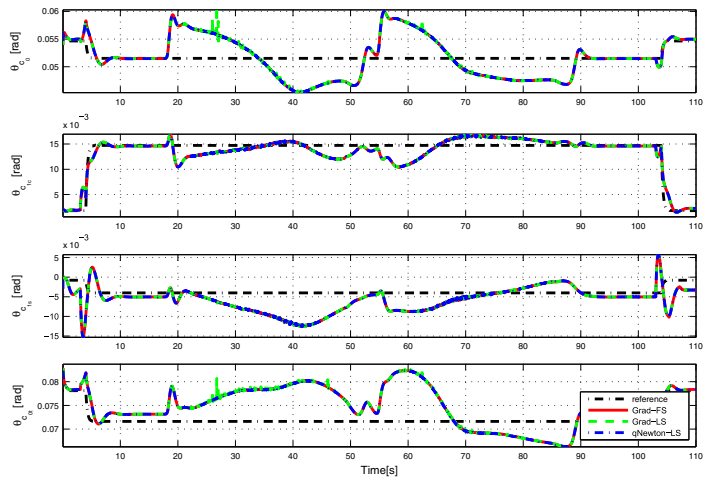
(b) Euler angles and reference



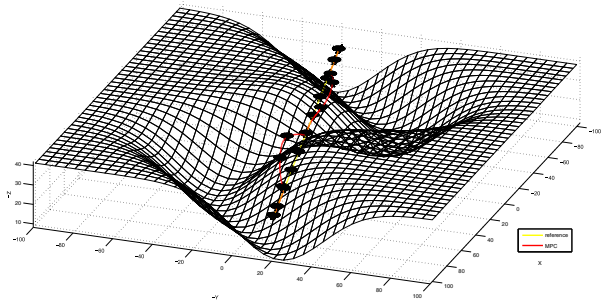
(c) Linear velocities and reference

Fig. 3 Trajectory tracking with terrain avoidance

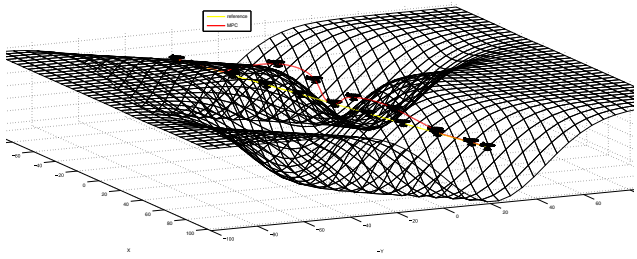
Fig. 4 Trajectory tracking with terrain avoidance



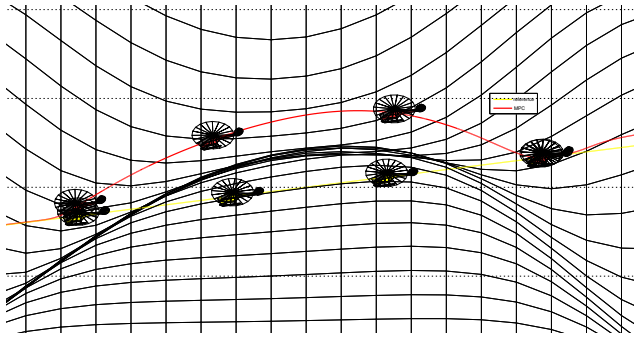
(a) Actuation



(b) 3D Trajectory: Global View 1



(c) 3D Trajectory: Global View 2



(d) 3D Trajectory: Detail View

Table 1 Iterations and computation times

Method	avg(n_{it})	avg(t_{CPU})	max(t_{CPU})
Gradient with Fixed Step (Grad-FS)	1129.9	41.4 s	84.5 s
Gradient with Line Search (Grad-LS)	136.3	12.5 s	46.3 s
Quasi-Newton with Line Search (qNewton-LS)	20.6	4.6 s	23.4 s

algorithms are overlapping, which indicate that they have identical minimization performance.

The improvements introduced by the different methods are presented in Tables 1 and 2, regarding the average computation time avg(t_{CPU}), the maximum computation time max(t_{CPU}), and the average number of iterations avg(n_{it}). For instance, the reduction percentage of the average CPU time from algorithm A to B is computed using

$$\text{Reduction}_{\text{A}\to\text{B}}(t_{CPU}) = 100\,\frac{\text{avg}(t^{\text{A}}_{CPU}) - \text{avg}(t^{\text{B}}_{CPU})}{\text{avg}(t^{A}_{CPU})}\ \%\ .$$

From this data it is evident that using the line search algorithm and the quasi-Newton method is more advantageous than using the traditional gradient method with a fixed step. According to these results, the control algorithm solution combines the two conflicting objectives of trajectory tracking and effective terrain avoidance, redirecting the vehicle to avoid the terrain while keeping the shortest possible distance with respect to the reference trajectory. Notwithstanding these improvements in computation times, the algorithms are not yet prepared for a real-time implementation that is compatible with the high sample rate nature of the envisioned platform (50 Hz). For this reason, future work will focus on implementation efficiency, further simplification of the helicopter model used within the MPC algorithm and redefinition of the terrain representation. With the steadily increasing computational power, new generation processors and algorithm refinements will make these algorithms usable in the near future.

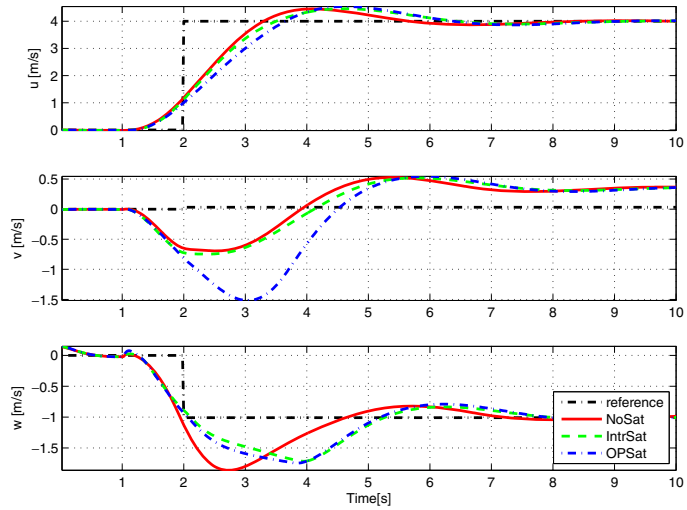
The presented terrain avoidance methodology is meant to be the core of a larger system, which would include a supervisor and a reference trajectory planner. Since this optimal control strategy with terrain avoidance can lead to terrain local minima, the global system should detect that the vehicle is approaching a local minimum and redesign the reference trajectory to redirect the vehicle to a different path.

4.2 Saturation Results

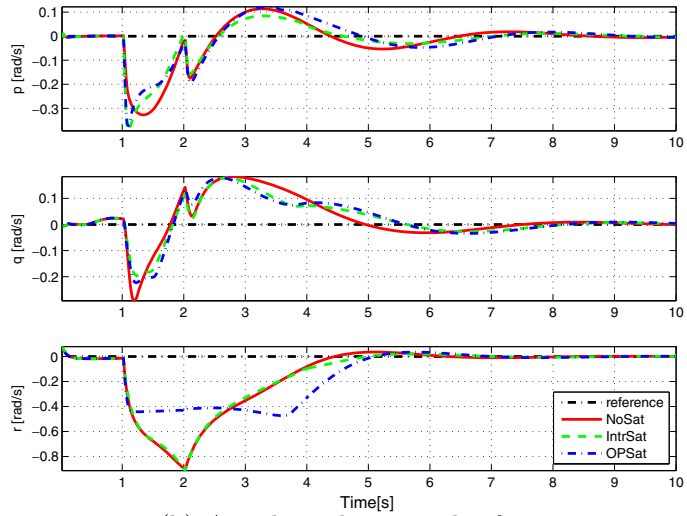
The second simulation results intend to shed some light on particular aspects of this NMPC implementation, which are not clear in a normal operation scenario such as that of the previous simulation results. The simulation is performed in the same conditions as the previous one, but the trajectory is changed to feature an abrupt change in the attitude of the vehicle: (1) hovering flight at the initial position facing north; (2) forward flight trajectory with constant speed $\|\mathbf{v}\| = 4\ \text{m}\,\text{s}^{-1}$ facing west. The simulation data comprises three different runs with: no saturations (NoSat); intrinsic saturations in the inputs (IntrSat); and optimization problem saturation constraints (OPSat), in both the input and state variables. The input saturation limits, used for both the IntrSat and the OPSat, are $\mathbf{u}_{\min} = \begin{bmatrix}0.0 & -0.1 & -0.015 & 0.0\end{bmatrix}'$ rad and $\mathbf{u}_{\max} = \begin{bmatrix}0.065 & 0.1 & 0.015 & 0.3\end{bmatrix}'$ rad. For the state saturation used in OPSat, all variables will remain within the bounds, except for the yaw rate, for which

Table 2 Relative improvements in number of iterations and CPU time between the algorithms

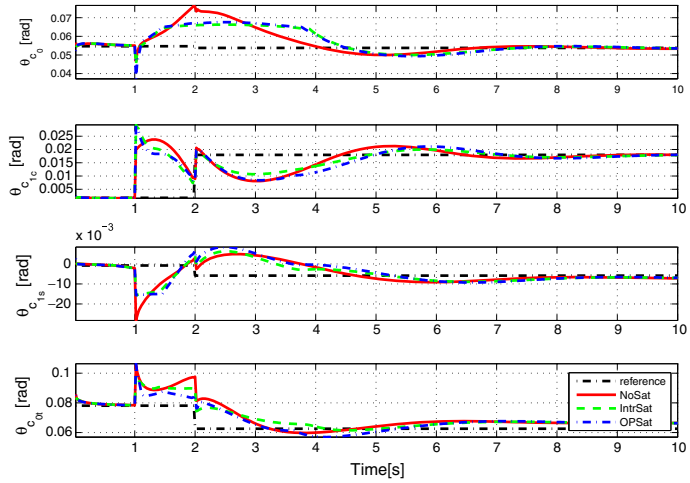
	avg(n_{it}) (%)	avg(t_{CPU}) (%)	max(t_{CPU}) (%)
Reduction from Grad-FS to Grad-LS	87.9	69.7	45.3
Reduction from Grad-LS to qNewton-LS	84.9	63.0	49.4
Reduction from Grad-FS to qNewton-LS	98.2	88.8	72.3



(a) Linear velocity and reference



(b) Angular velocity and reference



(c) Actuation

Fig. 5 Saturation test with: (i) no saturations (NoSat), (ii) intrinsic saturations in the inputs (IntrSat), and (iii) optimization problem saturation constraints (OPSat)

the absolute value is saturated at 0.4 rad s^{-1}. The bounds considered for this simulation results are more restrictive than the real bounds used in the terrain avoidance simulation in order to show the behavior of the saturation strategies. It can be observed in Fig. 5, that both saturation strategies, IntrSat and OPSat, are effectively keeping the main rotor collective and lateral cyclic controls within the bounds. Moreover, the state saturation in the optimization problem is also constraining the yaw rate absolute value below 0.4 rad s^{-1}. It can also be seen that the control algorithm starts changing the control effort N steps ahead of the change in the reference trajectory, demonstrating the preview ability of this NMPC strategy.

As discussed above, the objective of having intrinsic saturations is to make sure that the physical limitations of the platform and the numerical limitations of the model are always taken into account in the form of input saturations within the model function, even if there are no saturation constraints in the optimization problem.

5 Conclusions

Motivated by the use of autonomous rotorcraft in low altitude flight applications, this paper presented a simple NMPC-based strategy for terrain avoidance and motion control of helicopters. In addition to imposing state and input saturation constraints, the proposed solution enforces terrain avoidance by defining a repulsive field around the helicopter that grows exponentially fast as the distance between the vehicle and the closest point on the terrain decreases.

In contrast to the standard approach in NMPC literature, the actuation constraints were incorporated into the model so that every control action provided by the control algorithm is always valid and the controller design is simplified. The constrained optimization problem was reformulated as an unconstrained optimization problem using penalty methods to accommodate the saturation and terrain constraints, and Lagrange multipliers to eliminate the helicopter dynamic model constraint. The optimization problem was solved using an iterative algorithm that relies on the gradient and quasi-Newton methods to find the search direction and on the Wolfe conditions to find an estimate of the optimal step size. The simulation results were obtained using a simplified nonlinear helicopter model in the MPC algorithm and the full model as the real plant, showing that the presented methodology can achieve effective terrain avoidance while steering the vehicle along a reference trajectory. Moreover, it is shown that resorting to the quasi-Newton method with line search drastically reduced the number of iterations and CPU time relatively to the standard gradient method with fixed-step.

Similarly to most NMPC strategies used in high sampling rate platforms, the proposed methodology faces the challenge of real time implementation. Nevertheless, the constant technological advancements in terms of processing capability will enable the use of refined versions of these algorithms in the near future. In terms of CPU time consumption, the most critical tasks involve the helicopter model computations and the algorithm to determine the closest point on the terrain (which with modern sensors, like LADARs or time-of-flight cameras, can be obtained with negligible processing). Therefore, future work will focus on implementation efficiency, simplification of the helicopter model and also the simplification of the closest point computation. Furthermore, future algorithms shall be compared with other existing obstacle avoidance methods [12], using a standard set of benchmarks such as in [24].

References

1. Alamir, M., Bornard, G.: Stability of a truncated infinite constrained receding horizon scheme: the general discrete nonlinear case. Automatica **31**(9), 1353–1356 (1995)
2. Bemporad, A., Morari, M., Pistikopoulos, E.N., Dua, V.: The explicit linear quadratic regulator for constrained systems. Automatica **38**(1), 3–20 (2002)
3. Chen, C.C., Shaw, L.: On receding horizon feedback control. Automatica **18**(3), 349–352 (1982)
4. Chen, H., Allgöwer, F.: A quasi-infinite horizon nonlinear model predictive control with guaranteed stability. Automatica **34**(10), 1205–1217 (1998)
5. Cunha, R.: Modeling and Control of an Autonomous Robotic Helicopter. Master's thesis, Department of Electrical and Computer Engineering, Instituto Superior Técnico, Lisbon, Portugal (2002)

6. Cunha, R., Guerreiro, B., Silvestre, C.: Vario-xtreme Helicopter Nonlinear Model: Complete and Simplified Expressions. Technical report, Instituto Superior Técnico, Institute for Systems and Robotics (2005)
7. Cutler, C.R., Ramaker, B.L.: Dynamic matrix control: a computer control algorithm. In: Proceedings of the Joint Automatic Control Conference. San Francisco, CA (1980)
8. de Nicolao, G., Magnani, L., Magni, L., Scattolini, R.: A stabilizing receding horizon controller for nonlinear discrete time systems. In: American Control Conference. Chicago, Illinois (2000)
9. Findeisen, R., Imsland, L., Allgöwer, F., Foss, B.A.: State and output feedback nonlinear model predictive control: an overview. Europ. J. Contr. **9**, 179–195 (2003)
10. Garcia, C.E., Prett, D.M., Morari, M.: Model predictive control: theory and practice—a survey. Automatica **25**(3), 335–348 (1989)
11. Garcia, C.E., Morari, M.: Internal model control. A unifying review and some new results. Ind. Eng. Chem. Process Des. Dev. **21**(2), 308–323 (1982)
12. Goerzen, C., Kong, Z., Mettler, B.: A survey of motion planning algorithms from the perspective of autonomous uav guidance. J. Intell. Robot. Syst. **57**, 65–100 (2010)
13. Grancharova, A., Kocijan, J., Johansen, T.A.: Explicit stochastic predictive control of combustion plants based on gaussian process models. Automatica **44**, 1621–1631 (2008)
14. Guerreiro, B., Silvestre, C., Cunha, R.: Terrain avoidance model predictive control for autonomous rotorcraft. In: Proceedings of the 17th IFAC World Congress (IFAC'08), pp. 1076–1081. Seoul, Korea (2008)
15. Jadbabaie, A., Yu, J., Hauser, J.: Unconstrained receding-horizon control of nonlinear systems. IEEE Trans. Automat. Contr. **46**(5), 776–783 (2001)
16. Keerthi, S.S., Gilbert, E.G.: Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: stability and moving-horizon approximations. J. Optim. Theory Appl. **57**(2), 265–293 (1988)
17. Keviczky, T., Balas, G.J.: Receding horizon control of an f-16 aircraft: a comparative study. Control Eng. Pract. **14**(9), 1023–1033 (2006)
18. Kim, H., Shim, D., Sastry, S.: Nonlinear model predictive tracking control for rotorcraft-based unmanned aerial vehicles. In: American Control Conference, vol. 5, pp. 3576–3581. Anchorage, AK (2002)
19. Lapp, T., Singh, L.: Model predictive control based trajectory optimization for nap-of-earth flight including obstacle avoidance. In: American Control Conference, pp. 891–892. Boston, MA (2004)
20. Magni, L.: On robust tracking with non-linear model predictive control. Int. J. Control **75**(6), 399–407 (2002)
21. Magni, L., Raimondo, D.M., Allgöwer, F. (eds.): Nonlinear Model Predictive Control: Towards New Challenging Applications. Lecture Notes in Control and Information Sciences, vol. 384. Springer, New York (2009)
22. Mayne, D., Rawlings, J., Rao, C., Scokaert, P.: Constrained model predictive control: stability and optimality. Automatica **36**, 790–814 (2000). Survey Paper
23. Mehra, M., Rouhani, R., Eterno, J., Richalet, J., Rault, A.: Model algorithmic control: review and recent development. In: Proceedings of the Eng. Foundation Conf. on Chemical Process Control II, pp. 287–310. Sea Island, Georgia (1982)
24. Mettler, B., Goerzen, C., Kong, Z., Whalley, M.: Benchmarking of obstacle field navigation algorithms for autonomous helicopters. In: Proceedings of the American Helicopter Society 66th Annual Forum. Phoenix, AZ (2010)
25. Michalska, H., Mayne, D.Q.: Robust receding horizon control of constrained nonlinear systems. IEEE Trans. Automat. Contr. **38**(11), 1623–1633 (1993)
26. Nocedal, J., Wright, S.: Numerical Optimization. Springer Series in Operation Reasearch. Springer, New York (1999)
27. Gareth, D., Padfield.: Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modeling. AIAA Education Series. AIAA, Washington DC (1996)
28. Paulino, N., Silvestre, C., Cunha, R.: Affine parameter-dependent preview control for rotorcraft terrain following. AIAA J. Guid. Contr. Dynam. **29**(6), 1350–1359 (2006)
29. Propoi, A.I.: Use of linear programming methods for synthesizing sample-data automatic systems. Autom. Remote Control **24**, 837 (1963)
30. Qin, S.J., Badgwell, T.A.: A survey of industrial model predictive control technology. Control Eng. Int. **11**, 733–764 (2003)
31. Rawlings, J.B., Muske, K.R.: The stability of constrained receding horizon control. IEEE Trans. Automat. Contr. **38**(10), 1512–1516 (1993)
32. Richalet, J., Rault, A., Testud, J.L., Papon, J.: Model predictive heuristic control: applications to industrial processes. Automatica **14**, 413–428 (1978)
33. Shim, D., Kim, H., Sastry, S.: Decentralized reflective model predictive control of multiple flying robots in dynamic environment. In: Conference on Decision and Control, USA (2003)
34. Sutton, G., Bitmead, R.: Computational implementation of nonlinear model predictive control to nonlinear submarine. In: Allgöwer, F., Zheng, A. (eds.) Nonlinear Model Predictive Control, Progress in Systems and Control Theory, pp. 461–471. Birkhäuser, Basel-Boston-Berlin (2000)
35. Yoon, Y., Shin, J., Kim, H., Park, Y., Sastry, S.: Model-predictive active steering and obstacle avoidance for autonomous ground vehicles. Control Eng. Pract. **17**, 741–750 (2009)
36. Zadeh, L.A., Whalen, B.H.: On optimal control and linear programming. IRE Trans. Automat. Contr. **7**(4), 45–46 (1962)
37. Zavala, V.M., Biegler, L.T.: The advanced-step nmpc controller: optimality, stability, and robustness. Automatica **45**(1), 83–93 (2009)