Research Article

Offline Model Predictive Control Based on Weighted Projection over Polytopes

Fabio Andrade de Almeida

Instituto de Aeronáutica e Espaço (IAE), 12228-904 São José dos Campos, SP, Brazil

Correspondence should be addressed to Fabio Andrade de Almeida; almeidafaa@iae.cta.br

This work presents a novel offline model predictive control technique for tracking of constrained systems. The quadratic programming problem, commonly found in constrained control methods, is replaced by sequential offline set projections based on priority given to the decision variables. If a preference is established in terms of which decision variables are more desirable, the optimization problem can be solved by sequentially choosing the most important variables and performing a membership test with the projection of the constraint closed set over the related dimensions. Thus, real-time optimization is replaced by offline projection operations and online one-dimensional membership tests. This concept of decision variable prioritization is then applied to a form of model predictive control: feasible target tracking. Three quadratic programming problems are replaced by the proposed method. In the first problem, attainable steady-state demands are computed based on the performance of the plant. The reachable target command is then filtered in terms of dynamic admissibility, creating feasible inputs to the plant. Finally, the control is computed considering the current state and disturbance vectors along with the feasible and attainable command. Simulations of the method executing a path-following task are presented, demonstrating its benefits with negligible online computational burden.

1. Introduction

Model predictive control has been a matter of extensive research in recent decades because it provides a straightforward way to address the control of constrained multivariate systems [1]. The computed control to be applied in the plant is such that it satisfies the system constraints; therefore, an online constrained optimal control must be computed at every time instant. Though being initially conceived for applications in the process industry, other engineering fields are taking advantage of the developed techniques, such as in space systems [2] and aeronautical guidance and control [3, 4]. The facility of MPC to deal with constraints and its common appearance with traditional linear-quadratic controllers makes this technique suitable for the design of fault-tolerant flight control systems and also to allow operations near vehicle operational bounds.

Despite the fact that successful flight tests have already been executed to perform constrained control of fast dynamical modes of an aircraft [5], the related computational load is still high for the present generation of flight control computers. However, in the last decade, new approaches have been presented to decrease the computational burden of the online control calculations. Explicit MPC [6] has become the main strategy for moving offline most of the constrained control law computations. In the other direction, new methods were recently proposed [7, 8] to improve numerical performance during real-time execution.

Nevertheless, the online computation of the control actions depends mostly on linear or quadratic programming solvers, which do not offer guarantees of finite-time execution because of their iterative procedures. This makes MPC a non-certifiable technique for civil aviation control software, where deterministic algorithms are mandatory [9]. Explicit MPC could be an alternative for this problem, but it is still restricted for control of low-order systems and for regulation purposes [10]. An attempt to perform robust variable tracking, instead of nominal regulation, can increase the complexity of the state-space formulation in a way that makes it hard or even impossible to compute offline solutions.

Therefore, the main contribution of this work is the proposal of a novel method to move offline the nondeterministic computations of MPC, leaving only set membership verifications during real-time operation. This is achieved by proposing a new form to solve quadratic convex optimization problems and applying it on the feasible target tracking model predictive control (FTT-MPC) method [11], producing a new form of offline MPC. Based on the assumption of the existence of priority among controlled outputs and manipulated variables (inputs), the traditional quadratic programming problem is replaced by offline computation and projection of polyhedral sets. This new model predictive control method is then applied to perform airspeed, flight path angle, and turn rate tracking of a small UAV model during a path-following task with constraints on angle of attack, throttle, and aerodynamic surfaces deflections as well as with considerable plant-model mismatches, demonstrating adequate flight envelope estimation and protection.

The structure of this paper is organized as follows. Section 2 establishes the constrained control problem and initial assumptions. Section 3 presents a new method to compute the projection of a point over a polytope, based on prioritization of the decision vector components. Section 4 is the main part of this work, where the feasible target tracking MPC is converted to an offline implementation. Finally, Section 5 applies the technique to the longitudinal and lateral control of an unmanned aircraft, showing a practical and potential application of the proposed MPC method.

2. Problem Formulation

Let the discrete-time state-space model be defined by

$$
\begin{aligned}
x_{k+1} &= \Phi x_k + \Gamma u_k + \Gamma_d d_k, \\
d_{k+1} &= d_k, \\
y_k &= E x_k, \\
z_k &= H y_k,
\end{aligned}
\tag{1}
$$

where $x_k \in \mathbb{R}^n$ is the state vector, $u_k \in \mathbb{R}^m$ is the control vector, $y_k \in \mathbb{R}^p$ is the vector of observed variables, $d_k \in \mathbb{R}^q$ is the constant vector of disturbances, and $z_k \in \mathbb{R}^q$, $q \le m$, is the vector of controlled variables. The disturbance vector is included to model mismatches between the plant and the nominal model as well as constant external disturbances acting on the plant. Note that the number of disturbance variables is equal to the number of controlled outputs q.

The objective of the control system is to asymptotically eliminate the tracking error given a piecewise constant reference command r_c, which is

$$
z_k \xrightarrow[k \to \infty]{} r_c
\tag{2}
$$

in the presence of a disturbance d_k and given constraints on the state and control vectors $x_k \in \mathbf{X}$, $u_k \in \mathbf{U}$, where $\mathbf{X}$ and $\mathbf{U}$ are closed, bounded, and convex sets expressed by linear inequalities. In case r_c is not achievable, a reachable target r_{ss} will be computed with some priority among the controlled

variables. Additionally, the dynamical protection characteristic of the controller maintains the outputs inside their admissible ranges by computing feasible target values of z_{ss}.

Before proceeding with the assumptions, some useful definitions are given. A polyhedron is the intersection of a finite number of closed halfspaces:

$$
\mathbf{P} = \left\{ v \in \mathbb{R}^n \mid L_c v \le k_c \right\},
\tag{3}
$$

where L_c and k_c are, respectively, a matrix and a vector which represent linearly the set of halfspaces. A polytope is a bounded polyhedron. Given a polyhedron $\mathbf{P} \subset \mathbb{V} \times \mathbb{Y}$, where $\mathbb{V}$ and $\mathbb{Y}$ are subspaces, that is, sets with lower dimensions that $\mathbf{P}$, the projection π of $\mathbf{P}$ over $\mathbb{V}$, is defined as

$$
\pi_{\mathbb{V}} \mathbf{P} = \left\{ v \in \mathbb{V} \mid \exists y \in \mathbb{Y}, \ (v, y) \in \mathbf{P} \right\},
\tag{4}
$$

and the *slice* σ of $\mathbf{P}$ at a vector $r \in \mathbb{R}^p$ is defined as

$$
\sigma_r \mathbf{P} = \left\{ v \in \mathbb{R}^n \mid (v, r) \in \mathbf{P} \right\}.
\tag{5}
$$

Assumption 1. The pair (Φ, E) is assumed to be detectable with E full row rank. The following condition is equivalent:

$$
\operatorname{rank} \begin{bmatrix} \Phi - I & \Gamma \\ E & 0 \end{bmatrix} = n + p.
\tag{6}
$$

Assumption 2. It is assumed that the state and disturbance vectors are estimated by a nominally asymptotically stable observer. Because $d_k \in \mathbb{R}^q$, the observer proposed by Maeder et al. [12] is chosen for this work. Other choices of observers [13] would increase the dimension of the disturbance vector and add additional numerical complexity to the controller synthesis.

Assumption 3. The disturbance is bounded and lies in a compact convex polyhedron containing the origin in its interior.

3. Weighted Projection of a Point on a Polytope

In the next section, it will be presented a new form of MPC that basically shapes input references and computes the constrained control, by projecting points on convex polyhedral sets. The projected point is commonly computed by solving a quadratic programming (QP) problem [14] and several solvers are available to calculate the solution; however, they comprise iterative procedures with no guarantee of finite time execution. This section presents a novel and simple way to project a point over a polytope without need of online optimization.

The Euclidean orthogonal projection of v_0 on a polytope is given by the solution of the QP [14]:

$$
\min_v \ J(v_0) = (v - v_0)^T (v - v_0)
\tag{7}
$$

$$
\text{subject to: } v \in \mathbb{V}.
$$

We also refer to oblique 2-norm projection of the minimizer of the following weighted cost function:

$$\min_{y} \quad J\left(y_0\right) = \left(y - y_0\right)^T W \left(y - y_0\right)$$

$$\text{subject to:} \quad y \in \mathbb{Y}, \tag{8}$$

where $W \in \mathbb{R}^{n \times n}$ is a positive definite diagonal matrix. Clearly, both formulations are equivalent by doing the variable transformation $v_i = \sqrt{w_i} y_i, i = 1, \ldots, n$. Here the weights are chosen to influence the solution, giving relative different emphasis to some components in the minimization of the distance between the optimal solution y^* and y_0.

It is proposed to solve (8) with a special choice of each diagonal element of the W matrix: the largest individual weight penalizes the most important component in terms of distance minimization, followed by the second component, and so on. Giving enough scale separation between the diagonal elements, it is possible to solve (8) through sequential projections and slices of polytopes.

The proposed prioritized computation of y_p^* is given as follows. Let us consider the feasible set $\mathbb{Y}$. Its projection on the most important component y_1 creates the one-dimensional set $\mathbb{Y}_1$. Thus, the prioritized value $y_{1_p}^*$ is obtained by verifying whether $y_{1_0} \in \mathbb{Y}_1$ or not. If it is, $y_{1_p}^* = y_{1_0}$. If not, $y_{1_p}^*$ will be equal to the nearest extremum of $\mathbb{Y}_1$. Next, the set $\mathbb{Y}$ is projected over the two most important components, generating $\mathbb{Y}_2$. Because $y_{1_p}^*$ has been already selected, the choice of $y_{2_p}^*$ is done over the slice of $\mathbb{Y}_2$ at $y_{1_p}^*$, in which it is also a one-dimensional set. This sequence of projections and slices continues until the last component y_n.

Figure 1 exemplifies and compares, using a two-dimensional case, two solutions of (8), one considering $W = I$, and the other with weighting prioritization, that is, $w_1 \gg w_2$. The established setpoint (y_{1_0}, y_{2_0}) is not achievable because it lies outside the admissible set. In the unweighted projection computation, the pair (y_1^*, y_2^*) is the closest point to (y_{1_0}, y_{2_0}) and is perpendicular to the related active constraint. However, considering y_1 as the preferred variable to attain, the demand prioritization technique computes a solution where $y_{2_p}^*$ is the result of the prioritized choice $y_{1_p}^*$. Its selection is simply performed by a membership test with the projection of the two-dimensional polytope on the first dimension, as shown in the lower part of Figure 1.

The extension of the proposed method to compute decision variables in higher-dimensional spaces and its application to the problem of constrained control must take into consideration the definition of the projection operation. As stated by (4), the projection of a high-dimensional decision space on the most important variables subspace guarantees the existence of the remaining decision elements. Doing sequential projections clearly decreases the sets of the variables with low priority, even with degeneration to a point, but they will not be empty.

The presented method of solving (8) is particularly helpful for online applications. It is worth observing that the successive projections of the set $\mathbb{Y}$ can be done completely
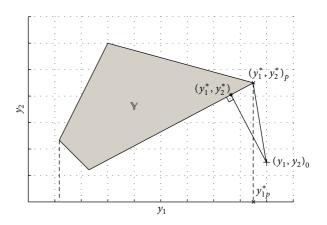

Figure 1: Orthogonal and prioritized projections of a point on a polytope.

offline. It only remains to calculate online the sequence of slices, which are computationally inexpensive.

4. Offline Feasible Model Predictive Control

While input constraints are relatively straightforward to manipulate, output constraints must be mapped onto the input space through some sort of dynamical prediction. Thus, the input applied to the system will consider the future evolution of the constrained outputs. Even basic formulations of MPC naturally compute optimal inputs that force the outputs to stay inside their limitations. However, not all reference values to be tracked are numerically feasible; that is, the imposed limitations and the system demand may not be compatible. In addition, even if feasibility is found, closed-loop stability may not be assured because the constraints are active not only at the time instant of the input computation, but also at all future instants.

Previous work presented the feasible target tracking MPC technique [11], a three-stage control method where each part comprises a quadratic programming problem. In the present work, the offline feasible model predictive control method (OFF-MPC) is proposed, replacing each optimization of the FTT-MPC by prioritized projections on polytopes that comprises the relevant system constraints.

The offline feasible MPC technique, schematically shown in Figure 2, guarantees both feasibility and closed-loop stability by first computing statically reachable commands, which respects the complete set of constraints when the system reaches the steady-state condition. The disturbance vector of (1) is required to correct the steady-state output limitations, that is, the admissible envelope, when differences between the model and the real system occur. After this first computation, dynamically feasible commands are calculated, to assure that the input and output constraints will be satisfied during the transient. In this step, both state and disturbance vectors are necessary because the dynamically feasible demand depends on the state at each time instant. Finally, the optimal constrained control is obtained in the MPC, considering a feasible and reachable demand.
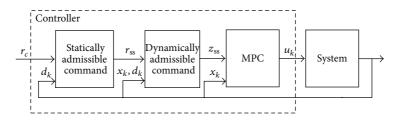
FIGURE 2: Overall scheme of the offline feasible MPC.

Off-line computation
(1) Sort the reference vector r_c in order of importance
(2) Create the polyhedron $\mathbf{P}_{ss}$
(3) Project $\mathbf{P}_{ss}$ over $\begin{bmatrix} r_{ss}^T & d_k^T \end{bmatrix}^T$ to create the polyhedron $\mathbf{P}_w$
(4) For $i = 1$ to q
    (a) Project $\mathbf{P}_w$ over $\begin{bmatrix} r_{1_{ss}} & \cdots & r_{i_{ss}} & d_k^T \end{bmatrix}^T$ to create $\mathbf{P}_{w_i}$
On-line computation
(1) Slice $\mathbf{P}_{w_1}$ at estimated values of d_k
(2) Check if r_1 lies inside the admissible range of $r_{1_{ss}}$
(3) For $i = 2$ to q
    (a) Slice $\mathbf{P}_{w_i}$ at selected $\begin{bmatrix} r_{1_{ss}} & \cdots & r_{(i-1)_{ss}} \end{bmatrix}^T$ and estimated d_k
    (b) Check if r_i lies inside the admissible range of $r_{i_{ss}}$

ALGORITHM 1: Sets of statically admissible commands.

4.1. Sets of Statically Admissible Commands. The separation between the determination of steady-state target values and constrained control computation creates distinct optimization problems, which are individually solvable through the demand prioritization technique. Thus, in a first step, the polyhedral set of input, state, and output constraints that act in the steady-state is created and projected offline over the highest priority variable to track, creating an interval of admissible and reachable values. In the sequence, another projection is performed over the second most important variable, and this operation repeats until the last variable, whose computation is simply performed by online slicing the complete constraint set with the choice of prioritized variables.

The equality constraints represented by the state-space and target equations can be converted into inequality constraints through introduction of arbitrarily small tolerance vectors ϵ_x and ϵ_z. Hence, (1) can be stated, at steady state, as

$$(\Phi - I) x_{ss} + \Gamma u_{ss} + \Gamma_d d_k \leq \epsilon_x I,$$

$$(\Phi - I) x_{ss} + \Gamma u_{ss} + \Gamma_d d_k \geq -\epsilon_x I,$$

$$H E x_{ss} - r_{ss} \leq \epsilon_z I,$$

$$H E x_{ss} - r_{ss} \geq -\epsilon_z I,$$

$$u_{ss} \in \lambda \mathbf{U},$$
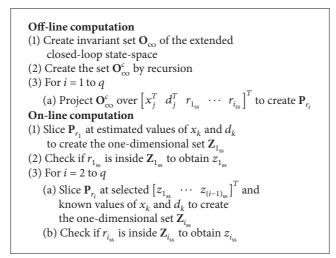
$$x_{ss} \in \lambda \mathbf{X},$$

(9)

which defines the polyhedron $\mathbf{P}_{ss}$ spanned by the augmented vector $\begin{bmatrix} x_{ss}^T & x_{ss}^T & r_{ss}^T & d_k^T \end{bmatrix}^T$. The projection of $\mathbf{P}_{ss}$ over $\begin{bmatrix} r_{ss}^T & d_k^T \end{bmatrix}^T$ generates a polyhedron $\mathbf{P}_w$ that encodes statically admissible commands r_{ss} to the real plant, provided that the disturbance is known or estimated. The scalar $\lambda \in (0, 1)$ slightly shrinks the constraints at steady state to allow the set computations of the next subsection.

If there is no priority among references to be tracked, the statically admissible vector r_{ss} is the orthogonal projection of r_c on the slice of $\mathbf{P}_w$ at estimated values of d_k. However, the prioritization and ordering of each element of r_c give the possibility of avoiding online quadratic programming computation, being replaced by an offline sequence of polyhedra projections.

The first required offline operation is a projection of $\mathbf{P}_w$ over $\begin{bmatrix} r_{1_{ss}} & d_k^T \end{bmatrix}^T$ to create the polyhedron $\mathbf{P}_{w_1}$. The procedure follows with the projection of $\mathbf{P}_w$ over $\begin{bmatrix} r_{1_{ss}} & r_{2_{ss}} & d_k^T \end{bmatrix}^T$ to compute $\mathbf{P}_{w_2}$. This is performed by all ordered elements of r_c.

The online slice of $\mathbf{P}_{w_1}$ with estimated values of d_k gives the admissible range of $r_{1_{ss}}$. Thus, in the sequence, slices are performed with the disturbance and selected attainable reference values. This sequence of offline projections and online slices is followed until the last demand r_q, as shown by Algorithm 1.

4.2. Sets of Dynamically Admissible Commands. Once the attainable target r_{ss} is computed, the next step of the OFF-MPC method is the computation of the dynamically

---

Off-line computation
(1) Create invariant set $\mathbf{O}_\infty$ of the extended
    closed-loop state-space
(2) Create the set $\mathbf{O}_\infty^c$ by recursion
(3) For $i = 1$ to q
    (a) Project $\mathbf{O}_\infty^c$ over $\begin{bmatrix} x_j^T & d_j^T & r_{1_{ss}} & \cdots & r_{i_{ss}} \end{bmatrix}^T$ to create $\mathbf{P}_{r_i}$
On-line computation
(1) Slice $\mathbf{P}_{r_1}$ at estimated values of x_k and d_k
    to create the one-dimensional set $\mathbf{Z}_{1_{ss}}$
(2) Check if $r_{1_{ss}}$ is inside $\mathbf{Z}_{1_{ss}}$ to obtain $z_{1_{ss}}$
(3) For $i = 2$ to q
    (a) Slice $\mathbf{P}_{r_i}$ at selected $\begin{bmatrix} z_{1_{ss}} & \cdots & z_{(i-1)_{ss}} \end{bmatrix}^T$ and
        known values of x_k and d_k to create
        the one-dimensional set $\mathbf{Z}_{i_{ss}}$
    (b) Check if $r_{i_{ss}}$ is inside $\mathbf{Z}_{i_{ss}}$ to obtain $z_{i_{ss}}$

ALGORITHM 2: Sets of dynamically admissible commands.

admissible target z_{ss}, which guarantees the existence of a feasible control action to steer states and controls to the target values $x_{ss} \in \lambda\mathbf{X}$ and $u_{ss} \in \lambda\mathbf{U}$. The starting point is the proposed control law predicted for N steps [15]:

$$u_j = -K\left(x_j - x_{ss}\right) + u_{ss} + c_j \quad j = k, \ldots, k + N - 1,$$
$$u_j = -K\left(x_j - x_{ss}\right) + u_{ss} \quad j \geq k + N. \tag{10}$$

The model predictive controller calculates the contribution c_j over N steps. A reparameterization that relates the target state vector with the target control vector and disturbance is also proposed:

$$x_{ss} = (I - \Phi)^{-1}\Gamma u_{ss} + (I - \Phi)^{-1}\Gamma_d d_k. \tag{11}$$

Therefore, assuming that r_{ss} is piece-wise constant, it will be convenient to extend the state-space vector as $x_a = \begin{bmatrix} x_j^T & d_j^T & u_{ss}^T & r_{ss}^T \end{bmatrix}^T$, leading to the following extended linear model:

$$\begin{bmatrix} x_{j+1} \\ d_{j+1} \\ u_{ss} \\ r_{ss} \end{bmatrix} = \begin{bmatrix} \Phi - \Gamma K & \Theta_d \Gamma_d & \Gamma\Theta & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \begin{bmatrix} x_j \\ d_j \\ u_{ss} \\ r_{ss} \end{bmatrix} + \begin{bmatrix} \Gamma \\ 0 \\ 0 \\ 0 \end{bmatrix} c_j, \tag{12}$$

where $\Theta = K(I - \Phi)^{-1}\Gamma + I$ and $\Theta_d = \Gamma K(I - \Phi)^{-1} + I$. In a similar way of that established in the previous section, the relation between the statically admissible demand r_{ss}, steady-state control u_{ss}, and the disturbance $d_k = d_{ss}$ is found through the formulation of the following constraint:

$$HE(I - \Phi)^{-1}\Gamma_d d_k + HE(I - \Phi)^{-1}\Gamma u_{ss} - r_{ss} \leq \epsilon_r I$$
$$-HE(I - \Phi)^{-1}\Gamma_d d_k - HE(I - \Phi)^{-1}\Gamma u_{ss} + r_{ss} \leq -\epsilon_r I \tag{13}$$

in addition to state limitations $x_j \in \mathbf{X}$. Because the reference value r_{ss} is statically admissible, u_{ss} fully respects the control limitations and does not need to be accounted for again. For stability purposes, it is assumed that, after N steps,

the extended state vector x_a at $j = k + N$ lies inside an invariant set $\mathbf{O}_\infty$. This extension of the state-space vector is similar to those proposed by Chisci and Zappa [16] and to recent techniques of economic MPC [17].

One possible way of interpreting the role of the MPC control is that the inputs $\{c_j\}_{j=k}^{k+N-1}$ are responsible for steering the extended state in N that moves into the invariant set $\mathbf{O}_\infty$. The set of all states that can be steered to $\mathbf{O}_\infty$ after N steps is the domain of attraction $\mathbf{O}_\infty^{(N)}$ of the complete closed-loop constrained system. This set is computed by first performing recursive substitutions of the extended dynamic system into the constraints, creating the set $\mathbf{O}_\infty^c$ and subsequently projecting it over x_{a_k} to eliminate the MPC contributions in the domain of attraction formulation.

Here the priority among the demands again eliminates the need for online optimization. Performing a projection of $\mathbf{O}_\infty^c$ on the subspace spanned by $\begin{bmatrix} x_j^T & d_j^T & r_{1_{ss}} \end{bmatrix}^T$ removes the dependency on u_{ss}, creating the polyhedron $\mathbf{P}_{r_1}$. The slice of $\mathbf{P}_{r_1}$ with estimated values of x_k and d_k gives the dynamically admissible range of $r_{1_{ss}}$. It is worth noting the similarity with the procedure adopted to obtain the static set in the previous subsection, but instead of considering only the estimated disturbance vector, the current state vector is also considered here to guarantee transient constraint satisfaction.

Once the most important demand is filtered in a dynamic sense and compared to the admissible range, a new command $z_{1_{ss}}$ is created. The sequence of dynamic filtering, which is summarized in Algorithm 2, follows the same steps of the static admissibility.

Remark 4. The reparameterization presented before is suitable for both square and nonsquare systems, where $q \leq m$ and $(I - \Phi)$ admits an inverse. If one or both conditions do not hold, the steady-state reparameterization suggested by Shead and Rossiter [18] is applicable, and the extension of the state space is performed considering the null space of $\begin{bmatrix} \Phi - I & \Gamma \end{bmatrix}$, instead of u_{ss}.

4.3. Control Calculation. The computation of z_{ss} introduces one additional degree of freedom in the constrained controller. Because the dynamically filtered demand is calculated from $\mathbf{O}_\infty^c$, the state and control vectors at steady state respect the related constraints. The feedforward control input u_{ss} is calculated as

$$u_{ss} = \left(HE \left(I - \Phi \right)^{-1} \Gamma \right)^{-1} \left[z_{ss} - HE \left(I - \Phi \right)^{-1} \Gamma_d d_k \right]. \quad (14)$$

Assumption 5. There are no elements of the state vector integrating elements of the controlled vector. The following condition is equivalent:

$$\text{rank} \begin{bmatrix} \Phi - I & \Gamma \\ HE & 0 \end{bmatrix} = n + q. \quad (15)$$

Assumption 6. (1) Let $\Psi = \Psi^T \in \mathbb{R}^{m \times m}$ be a diagonal positive definite matrix.

(2) Let $K \in \mathbb{R}^{m \times n}$ be a feedback gain such that $\Phi - \Gamma K$ is Hurwitz.

The constrained control problem is to determine the optimal control sequence $C_N^* = \{c_j\}_{j=k}^{k+N-1}$ that minimizes, at each time instant k, the following cost function:

$$J_k = \sum_{j=k}^{k+N-1} c_j^T \Psi c_j \quad (16)$$

subject to $c_j \in \text{slice}(\mathbf{O}_\infty^c, [\hat{x}_k^T \quad \hat{d}_k^T \quad z_{ss}^T]^T)$, $j \in (k, k + N - 1)$.

Theorem 7. *Assume that Assumptions 1 through 6 hold. Assume that the piecewise constant command r_{ss} is statically admissible. Thus, for any initial estimated state, disturbance and command $(\hat{x}_k, \hat{d}_k, r_{ss}) \in \mathbf{O}_\infty^{(N)}$, the dynamically admissible command z_{ss} and control law $u_k = \kappa(\hat{x}_k, \hat{d}_k, r_{ss})$ steer the system to the statically admissible setpoint r_{ss}, while respecting the constraints.*

Proof. First, the control law $u_k = -K(\hat{x}_k - x_{ss}) + u_{ss} + c_k$ is a feasible solution for the constrained control problem because $x_{a_k} \in \mathbf{O}_\infty^{(N)}$. Let us define the optimal cost function at the sample instant k as $J_k^* = \sum_{j=k}^{k+N-1} c_j^T \Psi c_j$ with feasible control sequence $C_N^* = \{c_j\}_{j=k}^{k+N-1}$. At the following sample time $k + 1$, define the control sequence $C_{N-1} = \{c_j\}_{j=k+1}^{k+N-1}$ and also $J_{k+1} = \sum_{j=k+1}^{k+N} c_j^T \Psi c_j$. Thus, $J_k^* - J_{k+1} = c_k^T \Psi c_k$. Considering that at $k + 1$ the optimal cost is J_{k+1}^*, it follows that $J_{k+1} \geq J_{k+1}^*$ and therefore $J_{k+1}^* - J_k^* \leq c_k^T \Psi c_k$.

Hence, the optimal cost sequence $\{J_k^*\}_{k=0}^\infty$ can never increase. Moreover, if $J_k^* = J_{k+1}^*$, then $c_k^T \Psi c_k = 0$, which means that the extended state-space x_{a_k} already lies already in the invariant set $\mathbf{O}_\infty$, from where the unconstrained linear feedback law $u_{j+N} = -K(x_{j+N} - x_{ss}) + u_{ss}$ steers the controlled vector to r_{ss} when $k \to \infty$. $\square$

Remark 8. The fact that r_{ss} is piecewise constant means that the above stability result holds, only if changes on r_{ss} occur after the system reaches its equilibrium around steady state. In this case, stability can be assured, but not when the reference is modified while the system dynamics is still prevailing.

The calculation of the MPC contributions can also follow the concept of decision variable prioritization, ordering the elements $c_1, c_2, \ldots, c_m$. From the polyhedron $\mathbf{O}_\infty^c$, one can project it offline over variables $\begin{bmatrix} x_j^T & d_j^T & r_{ss}^T & c_1 \end{bmatrix}^T$ to create the set $\mathbf{P}_{c_1}$. The known values of x_k, d_k, and $r_{ss} = z_{ss}$ are then used to online slice this set and provide an admissible range for c_1. The procedure is repeated to generate the sets $\mathbf{P}_{c_1}, \ldots, \mathbf{P}_{c_m}$ and is summarized in Algorithm 3. Note that the MPC contribution will be zero unless it is required to steer the states to the invariant set $\mathbf{O}_\infty$. Additionally, note that it is necessary to compute only the first m elements of C_N^* because the remaining inputs are not applied in the plant at time instant k. Thus, the online computational burden does not depend on the control horizon size, which affects only the offline projections of $\mathbf{O}_\infty^c$ on Algorithms 2 and 3.

4.4. Optimality of the Prioritized Constrained Solution. During normal operation, inside the system limitations, the control consists of a regulation around state and control steady-state values. This regulation is governed by the feedback linear gain K. Thus, optimality is easily achieved if K is obtained through optimal control methods (e.g., LQR).

Given a statically admissible reference to track, the controller offers the computation of both z_{ss} and c_k to respect system constraints. First, if some constraint becomes active, the dynamical filter is the first subsystem to act (Algorithm 2), computing z_{ss} in such a way that the MPC is capable of steering the states inside the extended invariant set after N moves. The ordered choice of the elements of z_{ss} does not affect the size of the domain of attraction of the controller because there are MPC nonlinear contributions c_k to maintain the system stable while tracking. This is assured by the projection operation on $\mathbf{O}_\infty^c$.

The ordering of controlled variables seems suitable for multivariable systems where there are interdependence among demands. This type of relationship can occur when operating at the limits of the system performance, what it is exactly MPC does, or following a desired optimal trajectory computed by some kind of higher level optimization layer. For example, the optimal climb schedule for an aircraft generally relates flight path angle (the angle that the airspeed vector makes with the ground) with the airspeed. Thus, rate or angle of climb is first selected, followed by the selection of airspeed. Depending on the aircraft performance, the airspeed is unique for a given altitude change demand. The example of Section 5 will explore this aspect, presenting the prioritization of demands as a mean to achieve better aircraft climb performance.

After the first filter, the contribution c_k is calculated (Algorithm 3) considering the previous filtered values of x_{ss} and u_{ss}. The existence of nonempty sets $\mathbf{P}_{c_i}$, $i = 1$ to m, is guaranteed by the computation of z_{ss}. Again, the prioritization is only a method to select the contributions inside each set $\mathbf{P}_{c_i}$. Therefore, the way that each constrained

Off-line computation
(1) Sort the control contribution vector c_k
(2) For $i = 1$ to m
  (a) Project $\mathbf{O}_\infty^c$ over $\begin{bmatrix} x_j^T & d_j^T & r_{ss}^T & c_1 & \cdots & c_m \end{bmatrix}^T$ to create $\mathbf{P}_{c_i}$
On-line computation
(1) Calculate x_{ss} and u_{ss} from the dynamically
   admissible vector z_{ss}
(2) Slice $\mathbf{P}_{c_1}$ at estimated values of x_k, d_k and $r_{ss} = z_{ss}$
(3) Check if $c_1 = 0$ lies inside the admissible values
(4) For $i = 2$ to m
  (a) Slice $\mathbf{P}_{c_i}$ at computed $\begin{bmatrix} c_1 & \cdots & c_{(i-1)} \end{bmatrix}^T$ and at
     values of x_k, d_k and $r_{ss} = z_{ss}$
  (b) Check if $c_i = 0$ lies inside the admissible values
(5) Compute $u_k = -K\left(x_k - x_{ss}\right) + u_{ss} + c_k$

ALGORITHM 3: Control calculation.

FIGURE 3: Closed-loop system for UAV trajectory control.

decision variable is computed (orthogonal or prioritized projection) does not enlarge or reduce either the sets $\mathbf{Z}_{i_{ss}}$, $i = 1$ to q, or $\mathbf{P}_{c_i}$.

Any choice of the MPC correction inside slice($\mathbf{O}_\infty^c$, $\begin{bmatrix} \hat{x}_k^T & \hat{d}_k^T & z_{ss}^T \end{bmatrix}^T$) is such that keeps the system inside the limitations. The minimization of the cost function given by (16) provides an optimal choice among all possible solutions. If the cost function prioritizes some variables among others, through selection of individual diagonal elements of Ψ, the presented technique solves the problem of oblique projection of a point on a set, finding an optimal solution for the problem stated in (16). Based on what is depicted in Figure 1, it is possible that the prioritized projection leads, in the control inputs with less priority, to larger c_k components than those computed through orthogonal unweighted projection. Thus, the ordering of control inputs makes more use of the last elements to maintain the closed-loop system inside its limitations, since the first components of C_N^* have preference to be set closer or equal to zero.

5. Simulation Example

The test aircraft used for the simulations is a six-degree-of-freedom model of the unmanned aerial vehicle Acauã, which also takes into account actuator and engine dynamics [19].

Acauã is a propeller-driven fixed-wing unmanned testbed used at the Brazilian Aeronautics and Space Institute (IAE). Linear models for control synthesis have been obtained with airspeed of 27 m/s and a pressure altitude of 620 m, at zero sideslip.

The simulation scenario consists of following a path defined by geodetic waypoints and related courses, as shown by Table 1. In this work, two-dimensional Dubins paths [20, 21] were used to connect each waypoint. Briefly, a 2D Dubins path is a path that flies over a certain waypoint on its desired course makes a right or left turn followed by a straight line and finishes with a right or left turn, flying over the next waypoint on a certain course. The selected turn radius for the circular transitions was 190 m, which corresponds to a nominal rate of turn of 8.14 deg/s at an airspeed of 27 m/s. The on-off turn rate commands created by 2D Dubins paths can practically be considered as piecewise constant references to be followed by the MPC-based autopilot, which is favorable in terms of closed-loop stability (Theorem 7).

Thus, the lateral guidance system [22] will be responsible for creating autopilot demands to steer the vehicle straight or through right or left turns with a constant radius. In the vertical channel, piecewise constant flight path angle demands were generated to explore the full flight envelope of the aircraft model. Figure 3 shows schematically the complete closed-loop system for the simulation.

Number	Lat. (deg.)	Long. (deg.)	Course (deg.)
1	−21.9842	−47.3426	0
2	−21.9765	−47.3474	120
3	−21.9765	−47.3377	240

5.1. Longitudinal Constrained Autopilot. Consider the unmanned aerial vehicle continuous-time linearized longitudinal model described in [19]:

$$
A_{\text{long}} = \begin{bmatrix} -0.1849 & 0.9732 & -0.1182 & -9.8067 \\ -0.016 & -2.498 & 0.9673 & 0 \\ 0.0227 & -33.97 & -5.924 & -0 \\ 0 & 0 & 1 & 0 \end{bmatrix},
$$

$$
B_{\text{long}} = \begin{bmatrix} 3.582 & -0.7128 \\ -0.0008 & -0.197 \\ 0.0012 & -41.17 \\ 0 & 0 \end{bmatrix}. \tag{17}
$$

The state vector is composed of the true airspeed (m/s), angle of attack (rad), pitch rate (rad/s), and the pitch angle (rad), respectively. The control vector is composed of the engine throttle and the elevator angular deflection. Full state observation ($E = I$) is assumed, and it is intended to control both airspeed and flight path angle without offset with priority given to the second variable. Thus,

$$
H_{\text{long}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}. \tag{18}
$$

The feedback gain K was obtained by first defining an LQR continuous-time cost function and subsequently performing the discretization [23]. The elements of the Q matrix in the LQR cost function were chosen to penalize deviations of airspeed and flight path angle; thus, $Q = H_{\text{long}}^T \text{diag}([1 \ 3300]) H_{\text{long}}$. The R matrix was set to $R = \text{diag}([100 \ 820])$. The dynamical system and the weight matrices were discretized with $T_s = 0.05$ s, from which the matrices Φ and Γ are obtained, along with the feedback gain K. The disturbance model was chosen to have two variables, and the related distribution matrix was set to $\Gamma_d = \Gamma$. The observer gains were computed using the linear Kalman method.

The inputs are constrained as follows: $\|\delta_{t_c}\|_\infty \leq 0.5$ and $\|\delta_{e_c}\|_\infty \leq 10$ deg. The disturbances are assumed to respect the bounds $\|d_1\|_\infty \leq 0.5$, $\|d_2\|_\infty \leq 0.05$. It is required to protect the aircraft against aerodynamic stall while maneuvering, imposing $\alpha \in (-8.08, 3.36)$ deg, which corresponds to a total admissible interval of $(-3.06, 8.38)$ deg. The constraints at steady state were scaled through $\lambda = 0.99$.

The sets $\mathbf{P}_{w_1}$, $\mathbf{P}_{w_2}$ (Algorithm 1); $\mathbf{P}_{r_1}$, $\mathbf{P}_{r_2}$ (Algorithm 2); and $\mathbf{P}_{c_1}$ and $\mathbf{P}_{c_2}$ (Algorithm 3) were computed offline through the appropriate numerical routine [24] in the MATLAB environment, setting $\epsilon_x = \epsilon_z = \epsilon_r = 10^{-4.2}$ and the control horizon $N = 2$.

5.2. Lateral Constrained Autopilot. The same values of the control horizon N, tolerances ϵ_z, ϵ_r, and the sampling time from the longitudinal controller were used in the synthesis of the constrained lateral autopilot. The state-space continuous-time linear lateral model is given by

$$
A_{\text{lat}} = \begin{bmatrix} -0.2123 & 0.0868 & -0.9868 & 0.3618 \\ -5.6541 & -1.7311 & 0.4654 & 0 \\ 6.8358 & -0.0697 & -0.5304 & 0 \\ 0 & 1.0000 & 0.0875 & 0 \end{bmatrix},
$$

$$
B_{\text{lat}} = \begin{bmatrix} 0 & 0.0858 \\ 6.1382 & 1.1299 \\ -0.2630 & -5.0436 \\ 0 & 0 \end{bmatrix}, \tag{19}
$$

where the state vector is composed of the sideslip (rad), roll rate (rad/s), yaw rate (rad/s), and the bank angle (rad). Aileron and rudder angular deflections (rad) define the control vector. The track priority is given first to maintain zero sideslip and subsequently to follow the turn rate. Hence,

$$
H_{\text{lat}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \tag{20}
$$

After discretization of the state-space formulation, the discrete lateral gain is given by

$$
K_{\text{lat}} = \begin{bmatrix} 0.9302 & 0.2528 & -0.1069 & 0.5671 \\ 0.3871 & 0.0666 & -1.8854 & 0.1874 \end{bmatrix}. \tag{21}
$$

Full state feedback ($E = I$) is considered. Constraints on control angular displacements were defined as $\|\delta_a\|_\infty \leq 10$ deg and $\|\delta_r\|_\infty \leq 10$ deg. In addition, the related sets of Algorithms 2 and 3 were computed offline, with disturbance vector elements constrained to $\|d_1\|_\infty \leq 0.05$ and $\|d_2\|_\infty \leq 0.05$.

It is important to mention that Assumptions 1 through 6 are fully respected in the two-controller synthesis. Assumptions 1 and 2 are accomplished by the choice of the estimator (Kalman) and full state observation. Bounds for the disturbance vector are established (Assumption 3). Finally, the choice of the controlled vector in both autopilots, that is, no states as pure integrators of controlled variables and the LQR feedback gains, guarantees Assumptions 5 and 6.

5.3. Simulation Analysis. Two cases were considered for the simulation. First, nominal engine power and longitudinal stability were considered in the computation of the controllers. The second case, however, simulates an underpowered Acauã (60% of maximum nominal power) with 50% of the original static stability, with the nominal controllers. Figure 4 shows the horizontal path followed by the aircraft in both cases. During the flyover of waypoints 2 and 3, an abrupt modification in the turn direction was required, but the vehicle strictly followed the computed Dubins path due to the high-gain lateral controller. The maximum lateral deviation during the two cases was 8.1 meters but with immediate guidance and control regulation to zero. The feasible commands z_{ss} and aircraft responses in the first case are shown in Figure 5. On

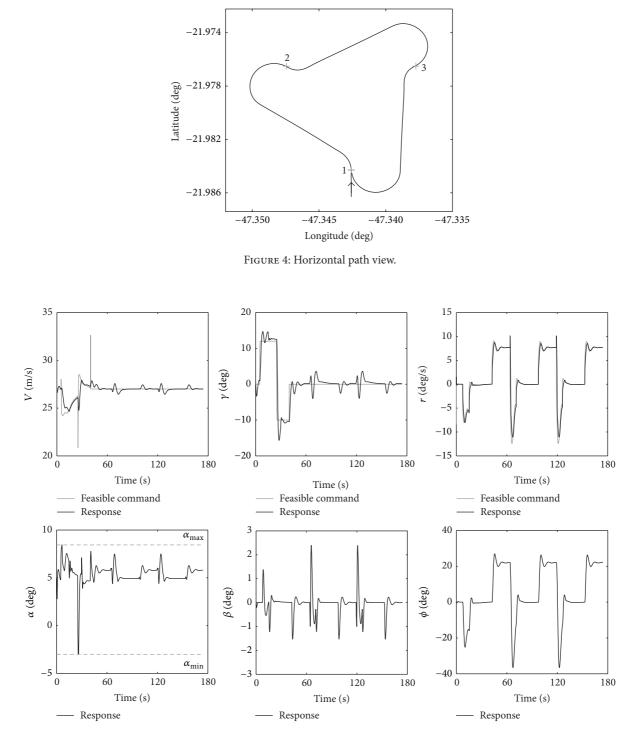FIGURE 4: Horizontal path view.

FIGURE 5: Controlled outputs and state variables in the nominal simulation.

two occasions, aggressive roll reversals were demanded by the Dubins path and guidance from the lateral autopilot, but fast and stable tracking of the turn rate was achieved with small variations of the sideslip angle.

In addition to the lateral tracking task, an aggressive variation of the flight path angle was simulated by an arbitrarily large command. Regardless of the airspeed command, the highest attainable climb angle (higher priority) was selected.

The angle of attack has changed between maximum and minimum values but never exceeded the limitations. The full exploration of the flight envelope for climbing and descent is shown in Figure 6. Engine throttle is demanded at its maximum and minimum values, and the elevator deflection reached its minimum deflection twice. An adequate lateral path-following performance was achieved through a high-gain lateral feedback control, which manipulated aileron and
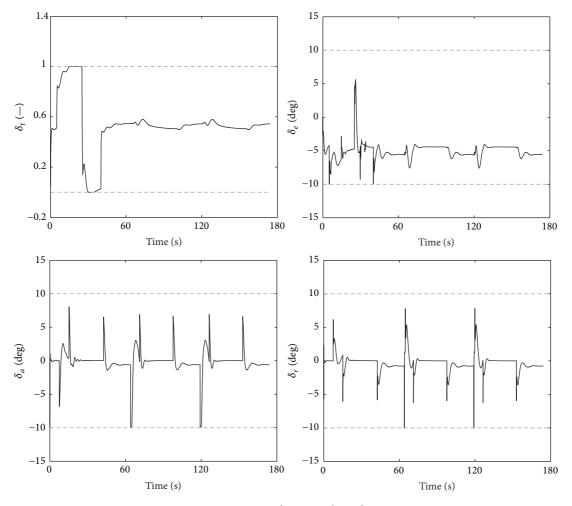
FIGURE 6: Inputs in the nominal simulation.

rudder deflection to their limits as shown also in Figure 6. Thus, unconstrained control strategies would most likely fail to track the considered turn rates.

The ability to cope with plant-model mismatches is demonstrated in the case of an underpowered vehicle with reduced static stability. Figure 7 shows related feasible demands and aircraft responses given the same path to be followed and flight path angle command. With only 60% of the nominal maximum power, the aircraft is not able to achieve the previous nominal climb angles, but due to the slice of $\mathbf{P}_{w_1}$ and $\mathbf{P}_{w_2}$ at the estimated disturbance values, real attainable commands are online computed. It is worth noting the demanded reduction of airspeed, which is crucial in the task of tracking the maximum climb angle. The longitudinal controller maintained performance and stability despite the fact that the real aircraft was less stable. Figure 8 shows that the elevator deflection required to maneuver was noticeably less than in the first case. Additionally, the reduced available power is shown in the evolution of the engine throttle. Again, the key aspect of offset-free tracking is the disturbance vector of the longitudinal autopilot, as shown in Figure 9. While the disturbance was practically zero, that is, a small plant-model mismatch, in the first simulation, the mismatch is present but properly taken into account in this case.

5.4. Aspects of Computational Burden and Optimality. The elimination of online iterative numerical routines provided by OFF-MPC reflects the drastic runtime reduction. Using the MATLAB command `tic toc`, the complete constrained control computation (online sections of Algorithms 1, 2, and 3) required, on average, 347 μs in both vertical and lateral autopilots. The simulations were performed on a PC with a 3.20 GHz Intel Core i5 CPU and 4 GB of RAM. A previous work [11] presented larger runtimes, in the range of milliseconds. Recently, Hartley et al. [2] presented numerical simulations of MPC with execution times in the range of microseconds, but using iterative solvers that demand specific initialization tuning. Although the offline set projections are based on iterative numerical routines, the online segment is only a direct sequence of algebraic operations, completely eliminating the dependency on iteration stop criteria. It is worth mentioning that this numerical performance was achieved using only MATLAB-coded routines, without previous compilation that would speed up the constrained computation.

Another comparison is done with the purpose to demonstrate the optimality achieved when the problem stated by (8) is solved with offline projections and online slices. Figure 10 shows engine throttle (left) and elevator deflection (right) MPC corrections during a capture of 10 deg of flight path
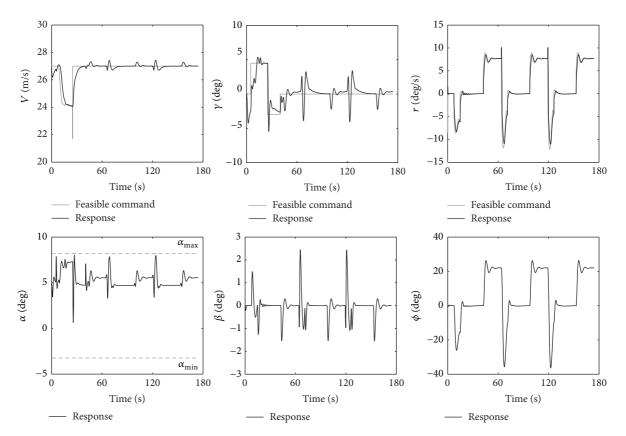
FIGURE 7: Controlled outputs and state variables in the simulation with reduced power and stability.

angle, using the proposed offline method and an active-set quadratic programming solver [25] to obtain the optimal solution of (16). Note the large nonlinear correction required to maintain the elevator deflection inside the limits.

Although the weighting matrix Ψ was set equal to identity, both numerical methods computed identical solutions. This means that the proposed technique of computing the projection of a point on a polytope is capable of providing the same results as those calculated via quadratic programming solvers. Also, by design, elevator is the aerodynamic control surface to provide fast vertical change in flight path. This type of analytical priority is captured by both methods, with advantage to the OFF-MPC, which offers the optimal solution using less computational time. In this simulation of 6 seconds, the active-set QP required, on average, 3.5 ms, against 283 μs demanded by OFF-MPC.

6. Conclusion

The protection of the closed-loop system against excursions outside relevant limitations is an important task that must be performed by modern control systems. Moreover, performance limitations must also be taken into account to avoid commands with unattainable outputs. In this paper, a technique was presented to perform tracking on constrained systems. A new offline model predictive control technique was proposed based on the prioritization of quadratic programming decision variables. In this way, a sequence of

constraint set projections over incremental subspaces of the decision variables creates polyhedral sets that, after substitution of the previously computed decision variables, degenerates to a one-dimensional set, where the choice of the optimal decision variable turns into a simple membership verification.

The concept of decision variable prioritization was applied to convert the feasible target tracking model predictive control technique into an offline constrained control method. First, whether the reference vector to be tracked is statically admissible is determined. If the demand is beyond the performance limitations of the plant, an attainable solution is calculated starting with the highest priority reference, and the remaining are determined through substitutions in the incremental projections of the static constraint set.

In the sequence, the statically admissible demand is filtered, taking into consideration the feasibility of the MPC problem. Extending the state space with the reference to be followed and the disturbance model, offline sequential projections of the domain of attraction of the controller and online substitutions generate one-dimensional feasible sets from which dynamically feasible target outputs are easily computed. Finally, the nonlinear MPC contributions are calculated following the same method.

The complete offline MPC method was demonstrated to perform tracking of trajectory variables of an unmanned aerial vehicle model, with constraints on throttle, aerodynamic surfaces, and angle of attack. The flight envelope was
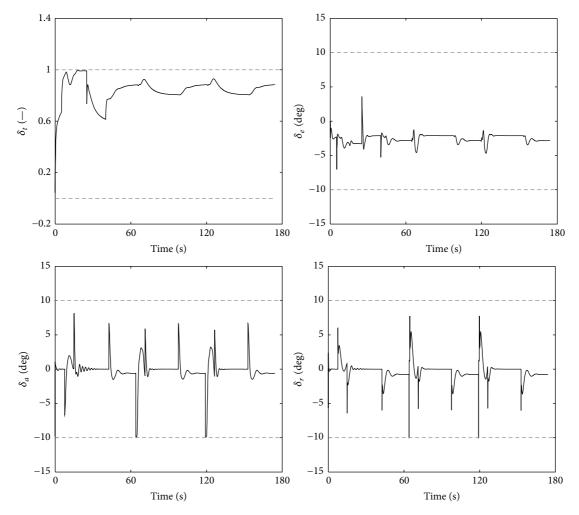
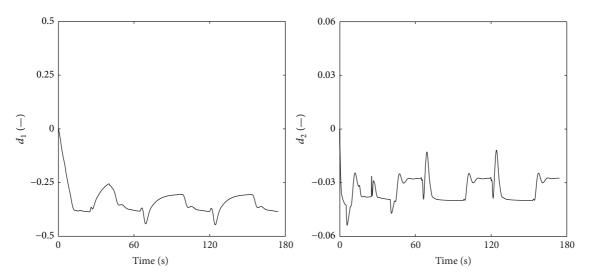Figure 8: Inputs in the simulation with reduced power and stability.

Figure 9: Estimated disturbances of the longitudinal controller in the simulation with reduced power and stability.
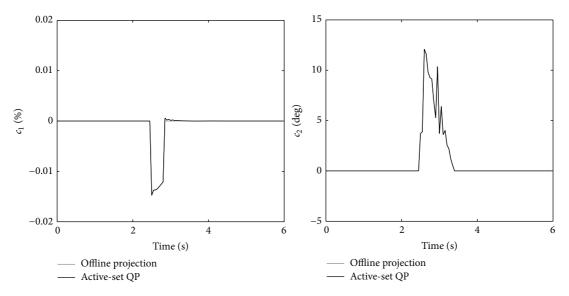
FIGURE 10: Nonlinear MPC corrections during capture of flight path angle.

estimated and protected even when plant-model mismatches were considered, yielding offset-free tracking and a complete satisfaction of constraints.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

References

[1] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*, chapter 1, Nob Hill Publishing, Madison, Wis, USA, 2009.

[2] E. N. Hartley, P. A. Trodden, A. G. Richards, and J. M. Maciejowski, "Model predictive control system design and implementation for spacecraft rendezvous," *Control Engineering Practice*, vol. 20, no. 7, pp. 695–713, 2012.

[3] A. Richards and J. P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *Proceedings of the American Control Conference*, vol. 3, pp. 1936–1941, Anchorage, Alaska, USA, May 2002.

[4] T. Keviczky and G. J. Balas, "Software-enabled receding horizon control for autonomous unmanned aerial vehicle guidance," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 3, pp. 680–694, 2006.

[5] F. A. de Almeida and D. Leissling, "Fault-tolerant model predictive control with flight-test results," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 2, pp. 363–375, 2010.

[6] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[7] S. Richter, C. N. Jones, and M. Morari, "Computational complexity certification for real-time MPC with input constraints based on the fast gradient method," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1391–1403, 2012.

[8] J. L. Jerez, G. A. Constantinides, and E. C. Kerrigan, "Towards a fixed point QP solver for predictive control," in *Proceedings of the 51st IEEE Conference on Decision and Control*, pp. 3851–3861, Maui, Hawaii, USA, 2012.

[9] S. A. Jacklin, "Closing the certification gaps in adaptive flight control software," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA-2008-6988, Honolulu, Hawaii, USA, August 2008.

[10] M. Monnigmann and M. Jost, "Vertex based calculation of explicit MPC laws," in *Proceedings of the American Control Conference (ACC '12)*, pp. 423–428, Montreal, Canada, June 2012.

[11] F. A. de Almeida, "Reference management for fault-tolerant model predictive control," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 1, pp. 44–56, 2011.

[12] U. Maeder, F. Borrelli, and M. Morari, "Linear offset-free model predictive control," *Automatica*, vol. 45, no. 10, pp. 2214–2222, 2009.

[13] G. Pannocchia and E. C. Kerrigan, "Offset-free receding horizon control of constrained linear systems," *AIChE Journal*, vol. 51, no. 12, pp. 3134–3146, 2005.

[14] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 7th edition, 2009.

[15] J. A. Rossiter, B. Kouvaritakis, and M. J. Rice, "A numerically robust state-space approach to stable-predictive control strategies," *Automatica*, vol. 34, no. 1, pp. 65–73, 1998.

[16] L. Chisci and G. Zappa, "Dual mode predictive tracking of piecewise constant references for constrained linear systems," *International Journal of Control*, vol. 76, no. 1, pp. 61–72, 2003.

[17] J. B. Rawlings, D. Angeli, and C. N. Bates, "Fundamentals of economic model predictive control," in *Proceedings of the 51st*

IEEE Conference on Decision and Control (CDC '12), pp. 3851–3861, Maui, Hawaii, USA, December 2012.

[18] L. R. Shead and J. A. Rossiter, "Feasibility for non-square linear MPC," in *Proceedings of the American Control Conference (ACC '07)*, pp. 4464–4469, IEEE, New York, NY, USA, July 2007.

[19] F. A. de Almeida, E. B. Guerra, F. A. D'Oliveira, and A. W. Mello, "Constrained linear quadratic tracker for optimal flight performance," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 6, pp. 1911–1918, 2012.

[20] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *The American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, 1957.

[21] A. Tsourdos, B. White, and M. Shanmugavel, *Cooperative Path Planning of Unmanned Aerial Vehicles*, chapter 2, John Wiley & Sons, Chichester, UK, 2010.

[22] D. G. Galisteu and F. A. de Almeida, "Three-dimensional guidance filter for autonomous collision avoidance," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA-2014-0961, National Harbor, Md, USA, January 2014.

[23] A. E. Bryson, *Control of Spacecraft and Aircraft*, Princeton University Press, Princeton, NJ, USA, 1994.

[24] M. Kvasnica, *Real-Time Model Predictive Control via Multi-Parametric Programming: Theory and Tools*, VDM, Saarbrucken, Germany, 1st edition, 2009.

[25] R. Fletcher, *Practical Methods of Optimization*, A Wiley-Interscience Publication, John Wiley & Sons, West Sussex, UK, 2nd edition, 1987.