

AUTOPILOT DESIGN OF AN UNSTABLE FIGHTER USING MODEL PREDICTIVE CONTROL

Marcos Toshio Ueda[1], Marko A. Rempel[2], Mauricio A. V. Morales[1] & João E. M. Fernandes[1]

[1]Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, 12228-900, Brazil
[2]Embraer,Gavião Peixoto, SP, 14813-000, Brazil

Abstract

Fault tolerance is a major concern in conventional control system design and is often achieved through physical redundancy. Model Predictive Control (MPC) has been extensively investigated for flight control systems due to its inherent fault-handling capability, which is achieved through analytical redundancy. This work investigates the design of autopilot functions using nonlinear MPC (NMPC). The proposed NMPC architecture performs altitude-hold and flight-level-change functions evaluated using the nonlinear ADMIRE model. The results confirm the inherent fault-tolerant characteristics and constraint handling.

Keywords: NMPC, fault tolerance, unstable fighter, autopilot functions

1. Introduction

Modern fighter aircraft, particularly those with delta-canard configurations such as the JAS-39 Gripen, are designed with reduced or negative static stability to enhance maneuverability. This inherent instability demands advanced fly-by-wire flight control systems (FCS) to maintain safe flight. Within such systems, autopilot functions reduce pilot workload and allow pilots to focus on strategic mission tasks, but must be both reliable and fault-tolerant across the operational envelope.

Classical autopilot designs, based on PID or Linear Quadratic Regulator (LQR) approaches, handle fault tolerance mainly through physical redundancy. Model Predictive Control (MPC) offers an alternative paradigm: it achieves analytical redundancy by redistributing control effort across available actuators at every sample instant through online optimization. This makes MPC naturally suited for reconfigurable flight control.

In the aerospace industry, MPC has been extensively explored for flight control applications. In recent decades, inner loop, outer loop, and guidance designs have been implemented using MPC. Most efforts have focused on reconfigurable flight control systems, with only a few on envelope protection and maneuver limiting [1]. For fighters, a reconfigurable flight control system design aims to address the following challenges [2]:

- A failure or loss of symmetry due to damage to the aircraft and hence cross-couplings between lateral-directional and longitudinal modes.
- Significant variations in aircraft dynamics under fault conditions
- A fighter may become more unstable, and the control reconfiguration must be very reactive
- Control effort redistribution to avoid saturation of the remaining actuators.

This paper presents the design and validation of altitude-hold, climb, and descent autopilot functions for the longitudinal axis of ADMIRE, using nonlinear MPC. The work demonstrates fault tolerance through a nominal and an extreme canard hardover scenario.

2. MPC Problem Statement

Model Predictive Control (MPC) is a method used to determine an optimal control for both single-input single-output (SISO) or multiple-input multiple-output (MIMO) systems over a receding horizon based on a predictive model of the system. The key elements of the general MPC architecture are depicted in Figure 1:

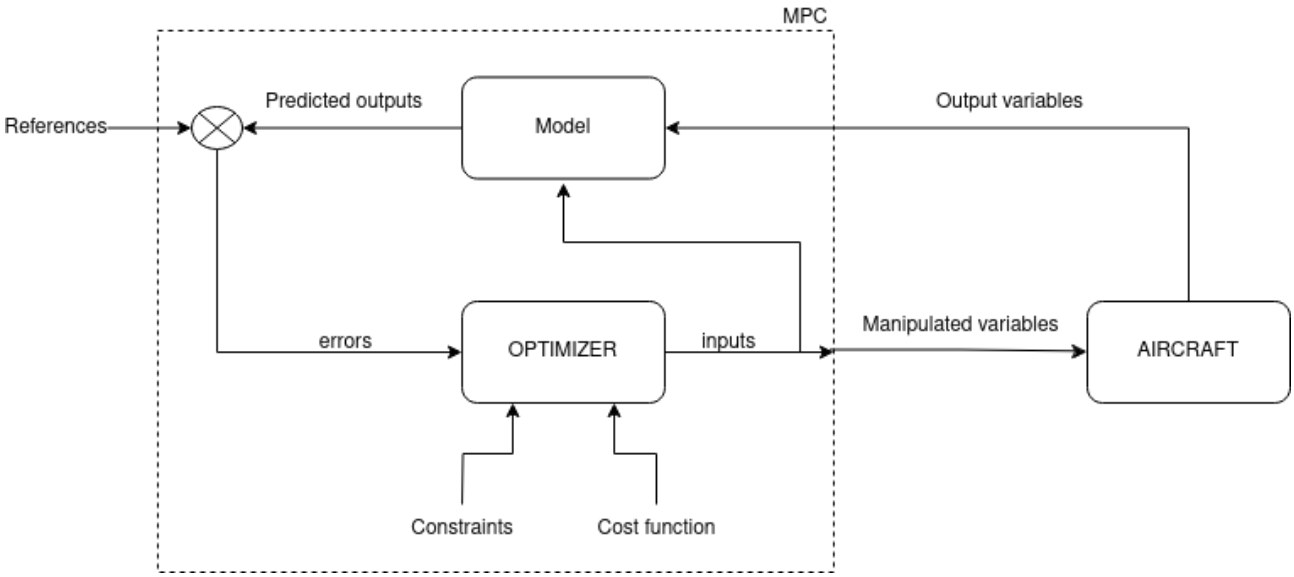


Figure 1 – MPC architecture

- Model: It predicts the future states of the system based on the current state and control inputs.
- Prediction horizon: It defines how far into the future the prediction model looks to carry out the optimization.
- Constraints: It allows for limiting the control inputs and state variables during optimization.
- Cost function: It evaluates the system's behavior given a reference. The objective is to minimize the cost function, respecting the constraints.

Given the system below:

$$x(k+1) = f\big(x(k), u(k)\big) \tag{1}$$

$$y(k) = h\big(x(k)\big) \tag{2}$$

Where x is an n-dimensional vector of state variables, u is an m-dimensional vector of control input variables, and y is a p-dimensional vector of controlled output variables. MPC minimizes the cost function J:

$$\begin{aligned}
&\min_{u} \quad J\big(x(k), u(\cdot)\big) \\
&\min_{u} \quad \sum_{i=Nd}^{Np} J_1\big[y(k+i)\,|\,k\big] + \sum_{j=0}^{Nu} J_2\big[u(k+j)\,|\,k,\ \Delta u(k+j)\,|\,k\big] \\
&\text{s.t.} \quad y_{lb} \le y(k+i\,|\,k) \le y_{ub}, \\
&\qquad u_{lb} \le u(k+j\,|\,k) \le u_{ub}, \\
&\qquad \Delta u_{lb} \le \Delta u(k+j\,|\,k) \le \Delta u_{ub}, \\
&\qquad \forall i \in \{Nd, \ldots, Np\}, \quad j \in \{0, \ldots, Nu\}.
\end{aligned} \tag{3}$$

Where J_1 and J_2 are functions to be minimized, y_{lb} and y_{ub} are the lower and upper limits of the outputs, respectively, u_{lb} and u_{ub} are the lower and upper values of the control inputs, and Δu_{lb} and Δu_{ub} are the lower and upper values of the rate-of-change of the control inputs. For quadratic functions, it becomes a quadratic optimization problem.

2.1 MPC Stability and Feasibility

Unlike LQR, a finite-horizon MPC does not guarantee stability by default. Stability can be ensured by including terminal cost, terminal set, and terminal control law ingredients in the formulation [3]. Feasibility under infeasible references is handled via slack variables, which soften state and output constraints by adding a penalty term to the cost function. For tracking time-varying references, Ferramosca et al. [4] propose an artificial steady-state extension that ensures feasibility and closed-loop stability. Formal closed-loop stability guarantees were not investigated in this work; instead, the controller stability and performance were evaluated through nonlinear time-domain simulations.

3. ADMIRE model

The ADMIRE model resembles a multipurpose JAS-39 Gripen aircraft, a layout that went through several optimizations to meet the Swedish Air Force's requirements and that enhances flight performance through artificial stability provided by a FBW system [5]. The dynamics equations are nonlinear, and the model without any FCS is unstable.

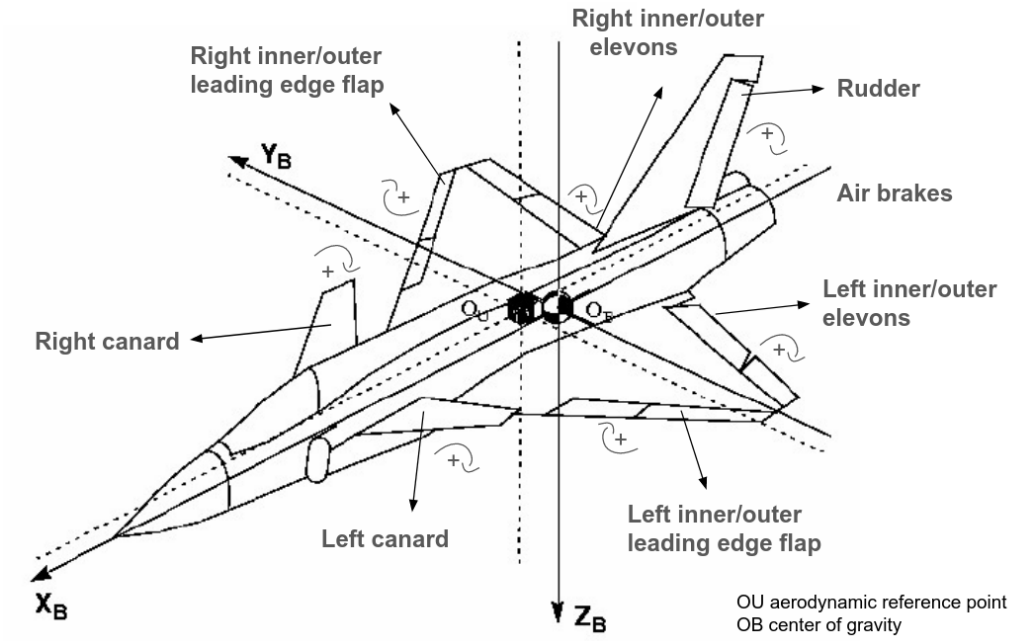


Figure 2 – Layout configuration - Figure adapted from [6]

The Aero-Data Model In a Research Environment (ADMIRE) is a small single-seat fighter aircraft model with a delta-canard configuration (Figure 2). It results from the Generic Aerodata Model (GAM), developed by Saab AB, and other models such as engine, actuator, and dynamics[6].
Its control surfaces include canards ($3.2m^2$), inner and outer elevons ($1.2m^2$ and $1.35m^2$ per side), leading-edge flaps, a rudder, and air brakes. Thrust vectoring via a nozzle is also modeled. Actuator dynamics are represented by a first-order lag ($\tau = 0.05s$) with a Padé approximation delay ($T = 20ms$). The nominal configuration (Table 1) was used in the simulations (Section 5.).

4. Architecture

The proposed architecture can execute the following autopilot functions: Altitude hold, climb, and descent. It uses two Nonlinear Model Predictive Controllers (NMPCs) for aircraft control. One NMPC manages the fast dynamics for flight path control (inner loop), while the other addresses slow dynamics for altitude control. The desired altitude is the only input required. This architecture holds the

Parameter	Value	Unit
wing area	45.00	m^2
wing span	10.00	m
wing chord (mean)	5.20	m
Mass	9100	kg
I_x	21000	kgm^2
I_y	81000	kgm^2
I_z	101000	kgm^2
I_{xz}	2500	kgm^2

Table 1 – Nominal configuration data [6]

initial speed, which could also be an input. In case of a jammed surface, the speed is reduced to 100 m/s, and the aircraft is leveled off.

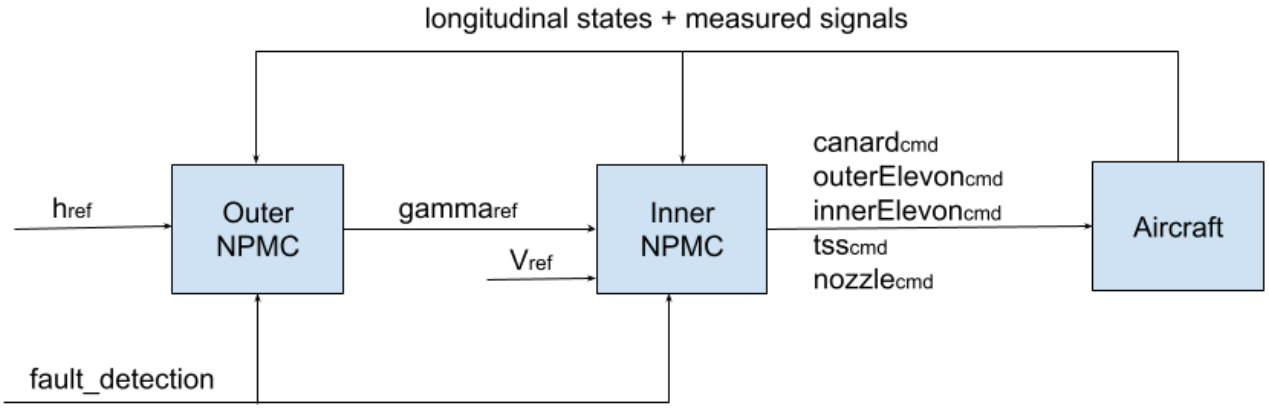


Figure 3 – NMPC architecture

4.1 Outer loop

The outer loop will generate a flight path angle reference to guide the inner loop to the desired altitude. The following kinematic equation is used to predict the altitude:

$$\dot{h}(k) = V_T(k_0) \cdot \Big(\cos(\alpha_0) \cdot \sin\big(\gamma_{\text{cmd}}(k) + \alpha(k_0)\big) - \sin(\alpha_0) \cdot \cos\big(\gamma_{\text{cmd}}(k) + \alpha(k_0)\big)\Big) \tag{4}$$

where: the velocity and alpha are constant within the prediction horizon, and flight path angle is the manipulated variable.
The outer loop minimizes the error between the admissible reference altitude (i.e., aircraft can follow that reference) and the current altitude. There is a secondary objective to ensure that the plant is following flight path angle. Besides, “e" represents a soft constraint penalty if violated.

$$J = \sum_{i=1}^{p-1} W_{z_1} \cdot \left(Z^i_{\text{pred}} - Z^i_{\text{ref}}\right)^2 + W_{z_2} \cdot \left(Z^p_{\text{pred}} - Z^p_{\text{ref}}\right)^2 + \sum_{i=1}^{p} W_\gamma \cdot \left(\gamma^i_{\text{ac}} - \gamma^i_{\text{cmd}}\right)^2 + 10 \cdot e^2, \tag{5}$$

where:

$$\gamma^i_{ac} = \theta^i_{ac} - \alpha^i_{ac} \tag{6}$$

The optimization problem can be stated using equations 4 and 5:

$$\begin{aligned} \min_{\{u_i\}_{i=0}^{N-1},\{x_i\}_{i=1}^{N}} \quad & J \\ \text{s.t.} \quad & x_i = h_i, \\ & x_{i+1} = x_i + T_s \dot{h}_i(x_i, u_i) \\ & x_i \in \mathbb{X}, \quad u_i \in \mathbb{U}, \quad \forall i, \\ & \Delta u_i = u_i - u_{i-1}, \quad |\Delta u_i| \leq \Delta U_{\max} \end{aligned} \tag{7}$$

4.2 Inner loop

The inner loop runs at a higher frequency to capture the fast dynamics related to pitch control and plant stabilization. It runs the ADMIRE's longitudinal model with five longitudinal modes: velocity, alpha, pitch rate, theta, and altitude. The controller follows the path angle reference provided by the outer loop. A first-order filter between the two controllers was used to smooth the square signal, improving convergence.

The cost function comprises five main terms: Cost function related to references, states, control commands, control command rates, and a penalty "e" due to the violation of the constraints.

$$J = J_y + J_x + J_u + J_{\Delta u} + 10 \cdot e^2 \tag{8}$$

The first term (equation 9) relates to flight path angle error and velocity error. The velocity term only considers the last term of the prediction horizon to prevent a sudden acceleration of the aircraft to keep the velocity constant (e.g., Mach 0.5). The terminal cost of flight path angle improves the convergence rate.

$$J_y = \sum_{i=1}^{p-1} W_{\gamma 1} \cdot \left(\gamma^i_{\text{pred}} - \gamma^i_{\text{ref}}\right)^2 + W_{\gamma 2} \cdot \left(\gamma^p_{\text{pred}} - \gamma^p_{\text{ref}}\right)^2 + \sum_{i=p-3}^{p} W_V \cdot \left(V^i_{\text{pred}} - V^i_{\text{ref}}\right)^2 \tag{9}$$

The reference cost function considers three states (alpha, theta, and velocity). One more state, pitch rate (equation 10), had been proven necessary to ensure longitudinal stability in extreme scenarios such as a hardover. However, it may compromise the aircraft's performance.

$$J_x = \sum_{i=1}^{p} W_q \cdot \left(Q^i_{\text{pred}}\right)^2 \tag{10}$$

The control inputs of the three pairs of surfaces (equation 11) are proportional to their surface area, so canard has a higher penalty for the same deflection. Although the elevons are independently operated, it is a design choice from Gripen [5] and Admire controller [6] to move elevons on the same side as a unit. The second term ensures that both pairs of elevons have the same deflection.

The engine has a special condition to control the activation of the afterburner. NMPC would activate the afterburner whenever thrust is required, which is not optimal when other solutions are available. Therefore, a penalty for high throttle settings prevents its activation.

$$\begin{aligned} J_u = & \sum_{i=1}^{p} W_c \cdot \left(U^i_{C_{\text{pred}}}\right)^2 + \sum_{i=1}^{p} W_{io} \cdot \left(U^i_{ei_{\text{pred}}} - U^i_{eo_{\text{pred}}}\right)^2 + \sum_{i=1}^{p} W_{ei} \cdot \left(U^i_{ei_{\text{pred}}}\right)^2 \\ & + \sum_{i=1}^{p} W_{eo} \cdot \left(U^i_{eo_{\text{pred}}}\right)^2 + \sum_{i=1}^{p} W_n \cdot \left(U^i_{n_{\text{pred}}}\right)^2 + \sum_{i=1}^{p} W_t \cdot \left(U^i_{t_{\text{pred}}}\right)^2 + \sum_{i=1}^{p} W_{ab} \cdot \left(U^i_{ab_{\text{pred}}}\right)^2 \end{aligned} \tag{11}$$

The last term (equation 12) prioritizes small control input adjustments by penalizing big variations of canard, inner elevon, outer elevon, throttle, and nozzle inputs, respectively.

$$
\begin{aligned}
J_{\Delta u} = &\sum_{i=1}^{p} W_c \cdot \left(\Delta U^i_{C_{\text{pred}}}\right)^2 + \sum_{i=1}^{p} W_{ei} \cdot \left(\Delta U^i_{ei_{\text{pred}}}\right)^2 + \sum_{i=1}^{p} W_{eo} \cdot \left(\Delta U^i_{eo_{\text{pred}}}\right)^2 \\
&+ \sum_{i=1}^{p} W_t \cdot \left(\Delta U^i_{t_{\text{pred}}}\right)^2 + \sum_{i=1}^{p} W_n \cdot \left(\Delta U^i_{n_{\text{pred}}}\right)^2
\end{aligned} \tag{12}
$$

The optimization problem statement:

$$
\begin{aligned}
\min_{\{u_i\}_{i=0}^{N-1}, \{x_i\}_{i=1}^{N}} \quad & J \\
\text{s.t.} \quad & x_0 = x_k, \\
& x_{i+1} = f_{\text{admire}}(x_i, u_i), \quad i = 0, \ldots, N-1, \\
& x_i \in \mathbb{X}, \quad u_i \in \mathbb{U}, \quad \forall i, \\
& \Delta u_i = u_i - u_{i-1}, \quad |\Delta u_i| \leq \Delta U_{\max}
\end{aligned} \tag{13}
$$

4.3 Optimization

For practical reasons, real-time optimization is a major concern when using a nonlinear model. It is not part of the scope to evaluate the computational feasibility or reduce computational burden. A benchmark was done to verify good practices and algorithms to be used in the implementation phase. The optimization problems are solved using *fmincon* function from MATLAB's Optimization Toolbox, which applies the Sequential Quadratic Programming (SQP) algorithm [7]. SQP is designed to handle nonlinear constraints and objective functions efficiently. It can solve a sequence of quadratic programming subproblems, breaking the original problem into small ones. It computes the Hessian Matrix, which improves the convergence rate of the optimization. Moreover, an Active Set Management manages active sets of constraints that help maintain feasibility and optimality in NMPC problems. These characteristics make the SQP algorithm well-suited for NMPC problems.

4.4 Fault detection system

The development of a Fault Detection and Diagnostics (FDD) system is beyond the scope of this work. The time required for detection is influenced by various factors, presenting a complex challenge. It is assumed that ADMIRE is equipped with a modern system capable of measuring control surface displacements. According to [8] [9], detecting a hardover condition within milliseconds is crucial to allow the pilot to maintain control of the aircraft. Modern electromechanical actuators can detect failures within 120 to 160 milliseconds [2]. In this work, a hardover is caused by a malfunction that causes the surface to reach its maximum deflection at the actuator's maximum rate (50 deg/s). The NMPC receives this information after 0.5 seconds.

5. Results

The three autopilot functions (altitude-hold, climb, and descent) were first tested under nominal conditions (fault-free) to evaluate the proposed architecture's baseline performance. Subsequently, a canard hardover fault was introduced to assess the controller's capability to redistribute control efforts among available actuators. Various combinations of prediction and control horizons were also examined in this scenario to evaluate their impact on control performance. In all tests, the ADMIRE was trimmed at Mach 0.5 and 4000 m as shown in Table 2

For the controllers, Table 3 shows the configurations of both controllers. The sample time was based on aircraft dynamics (0.05 seconds for the inner loop and 0.3 seconds for the outer loop). kufoalorNMPC used 0.05 seconds and 0.5 seconds, respectively. One second period (20 steps horizon) for the prediction horizon of the inner loop and 12 seconds (40 steps horizon) for the prediction horizon of the outer loop.

In general, the weight magnitude order follows the criteria considering the variables are normalized:

- 10^0 - Primary objective

Velocity	162.2893 m/s
AoA	3.4911 °
Pitch rate	0 °/s
Pitch angle	3.4911 °
Altitude	4000 m
Canard	0.0215 °
Elevons	-0.2396 °
Throttle	0.1587
Nozzle	0 °

Table 2 – Initial conditions.

	NMPC (outer loop)	NMPC (inner loop)
Prediction horizon	40	20
Control horizon	20	10
Sample time [s]	0.3	0.05

Table 3 – Inner and outer horizons/sample time

- 10^{-1} - Secondary objective
- 10^{-2} - Tertiary objective

The control surfaces are proportional to their areas. Although the inboard and outboard elevons have different areas, it is a requirement that they move as a unit by designing (see Section 4.1). Therefore, equal weights are assigned to the inboard and outboard elevons, with a penalty applied to their difference to enforce equal deflection. The throttle setting has less important command weight but a higher command rate weight to favor a steady command to prevent premature engine wear. Nozzle deflection is penalized to remain minimal during flight, as excessive deflection typically results in high throttle setting. The thrust vectoring model seems to generate a high pitch moment for very small nozzle deflection (10^{-1}) and may cause some noisy responses. Flight path is the primary objective, as well as the pitch rate proven necessary in case of hardover. The velocity penalty was sufficient to keep the velocity around the initial one.

	Command weight	Command rate weight
Canard	0.2783	0.2783
Outboard elevon	0.2217	0.2217
Inboard elevon	0.2217	0.2217
Nozzle	0.8	1
Elevon difference	0.5	0
Throttle setting	0.02/0.021*	0.5
	Weight	Terminal weight
Flight path	1	5
Pitch rate	0.8	0
Velocity	0	0.03

Table 4 – Inner loop MPC weights used in Figure 5

**The first weight corresponds to the throttle operation range without afterburner (0-0.76), and the second weight applies when the afterburner is activated (0.76-1). This difference helps to prevent frequent activation of the afterburner.*

For the outer loop, assigning equal weight provided a suitable trade-off between reference tracking and pursuing a flight path angle. A higher weight on flight path angle is needed, given that a kinematic model cannot predict the aircraft's inertial response at the beginning of the maneuver.

	Weight	Terminal weight
Altitude	0.5	1
Flight path	0.5	1

Table 5 – Inner loop MPC weights used in scenario 5

5.1 Altitude-hold, climb and descent performance without any failure

To evaluate the three autopilot function performances, the following flight profile was simulated:

- t = 0s, altitude-hold
- t = 5s, climb 150m in 10s (15 m/s)
- t = 15s, altitude-hold 4150m
- t = 25s, descent 150m in 10s (15 m/s)
- t = 35s, altitude-hold 4000m

As depicted in Figure 4, the NMPC in the outer loop translates the altitude reference into a flight path angle reference. The NMPC in the inner loop follows this reference while stabilizing the aircraft. The velocity is a secondary goal and can vary along the flight within the limits.
At the climb phase, as the leading edge of the canard goes down, the leading edge of the elevons follows this tendency to control pitch rotation and generate upwards lift. These commands allow the aircraft to “step up" without increasing the angle of attack, as in conventional aircraft, such as Embraer 195-E2. The engine nozzle also contributes to pitching and lifting the aircraft. The afterburner is activated to recover the initial velocity. A similar maneuver is performed at the descent phase “step down", but lift is generated downwards by deflecting the surface in the opposite direction. This capability of generating lift without directly depending on the angle of attack is a compelling technique in designing control laws for specific missions.

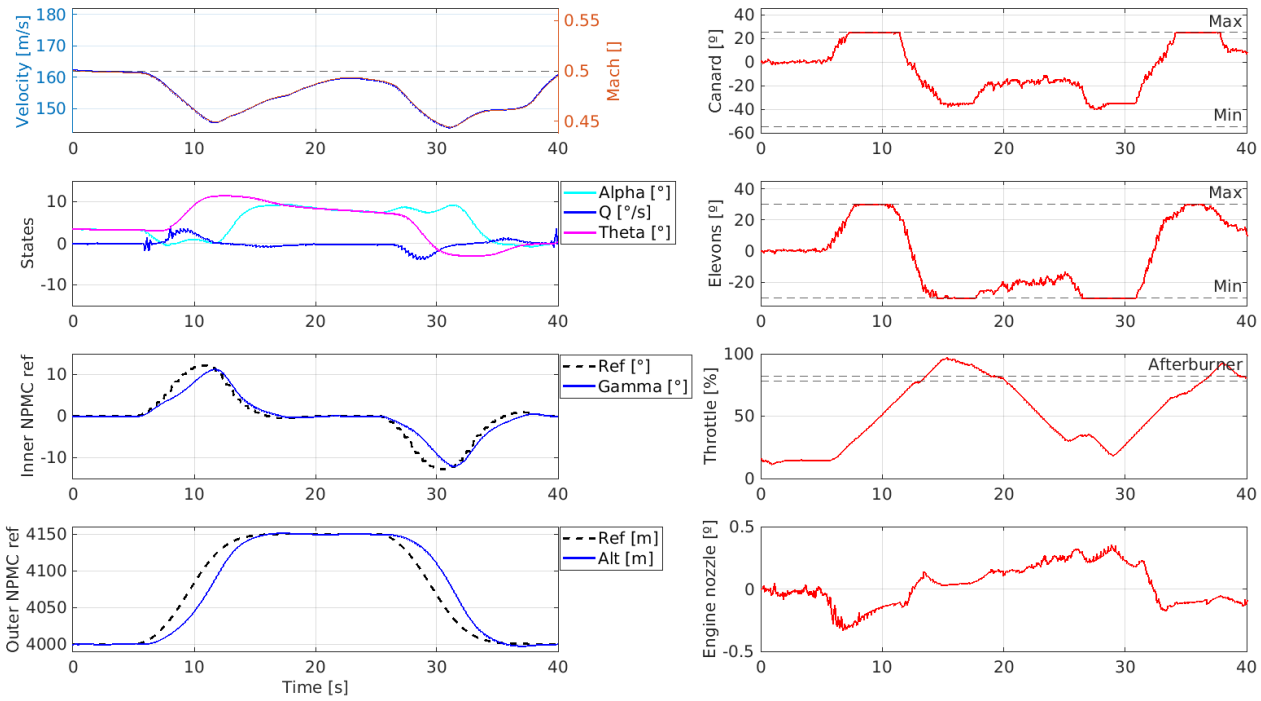


Figure 4 – Simulation of ADMIRE - NMPC - Climb and descent performance

The aircraft usually accelerates during descent because potential energy is transformed into kinetic energy. However, the results show a decrease in velocity and an increase in throttle. Two factors account for this:

- This maneuver requires a large deflection angle of the control surfaces, which generates a considerable amount of drag
- The engine nozzle consumes energy to maneuver the aircraft, so more thrust is needed.

From a flight control perspective, powerful actuators capable of resisting substantial hinge moments, particularly when the aircraft reaches its deflection limits at high speeds, are required. It may lead to heavier and bulkier actuators that may not be a viable solution or cause premature actuator wear. Furthermore, the actuator's energy consumption is considerable, given the frequent oscillation between end stops. One way to prevent these massive surface displacements is to increase the penalty for these commands. As a result, MPC finds new optimum commands as shown in Figure 5

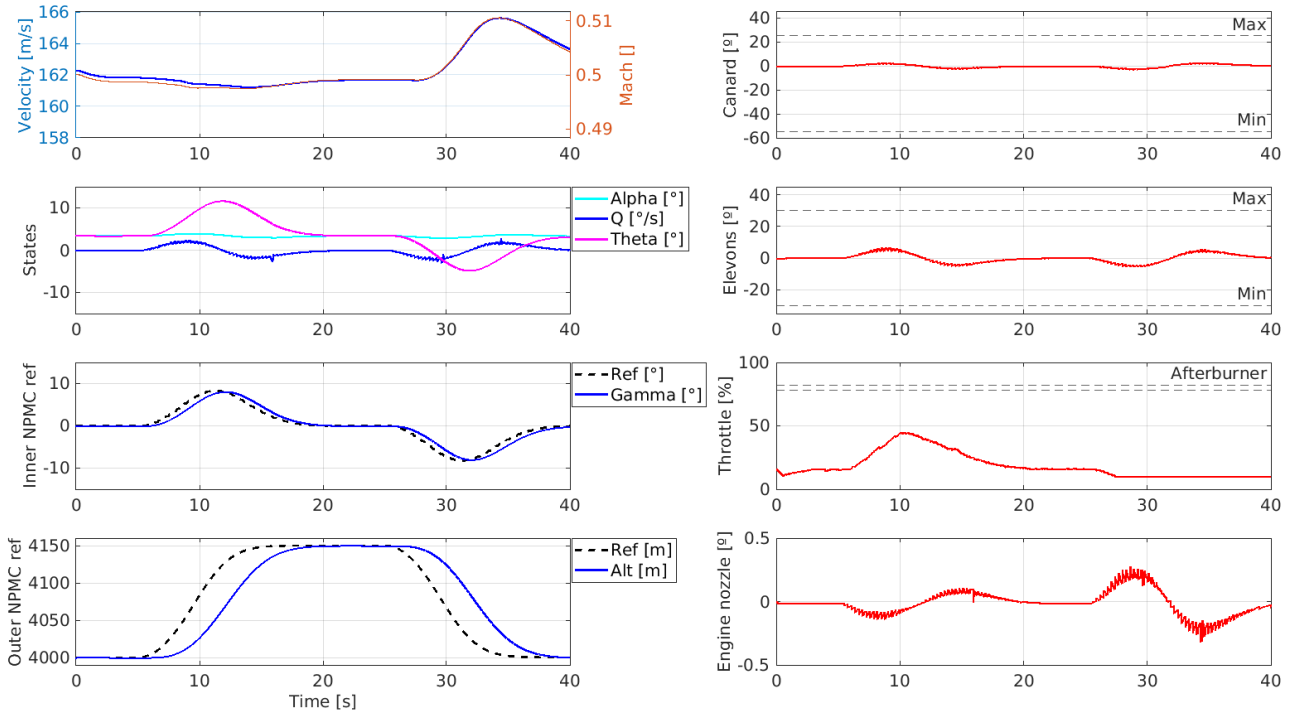


Figure 5 – Simulation of ADMIRE - NMPC - Climb and descent performance with actuator penalties

Due to implementation challenges, the actuator and engine dynamics are not included in the prediction model. The actuator dynamics, being significantly faster than the remaining aircraft dynamics, are neglected and omitted from the prediction model [10]. Nevertheless, the engine dynamics are comparatively slower, which might affect the overall dynamics. For instance, based on the first-order filter approximation of the ADMIRE engine dynamics, the transition from idle to maximum thrust position requires approximately 10 seconds. So, the engine rate can be limited directly on the NMPC. Nevertheless, the inner loop prediction horizon of one second can lead to a suboptimal and non-constant value. Only applying a terminal weight for the velocity reduces this variation. It may still overuse the engine while a steadier solution is probably available. A complementary model on the outer loop could address this problem by anticipating the next maneuver.
In Figure 6, the same simulation as in Figure 5, with the addition of actuators, nozzle, and engine dynamics. As expected, the inclusion of actuator dynamics has a minor impact, and NMPCs demonstrate satisfactory reference-tracking performance. The noise is attenuated by the actuator dynamics. Nevertheless, it exhibits difficulty in effectively commanding the throttle, leading to noticeable variation in velocity.

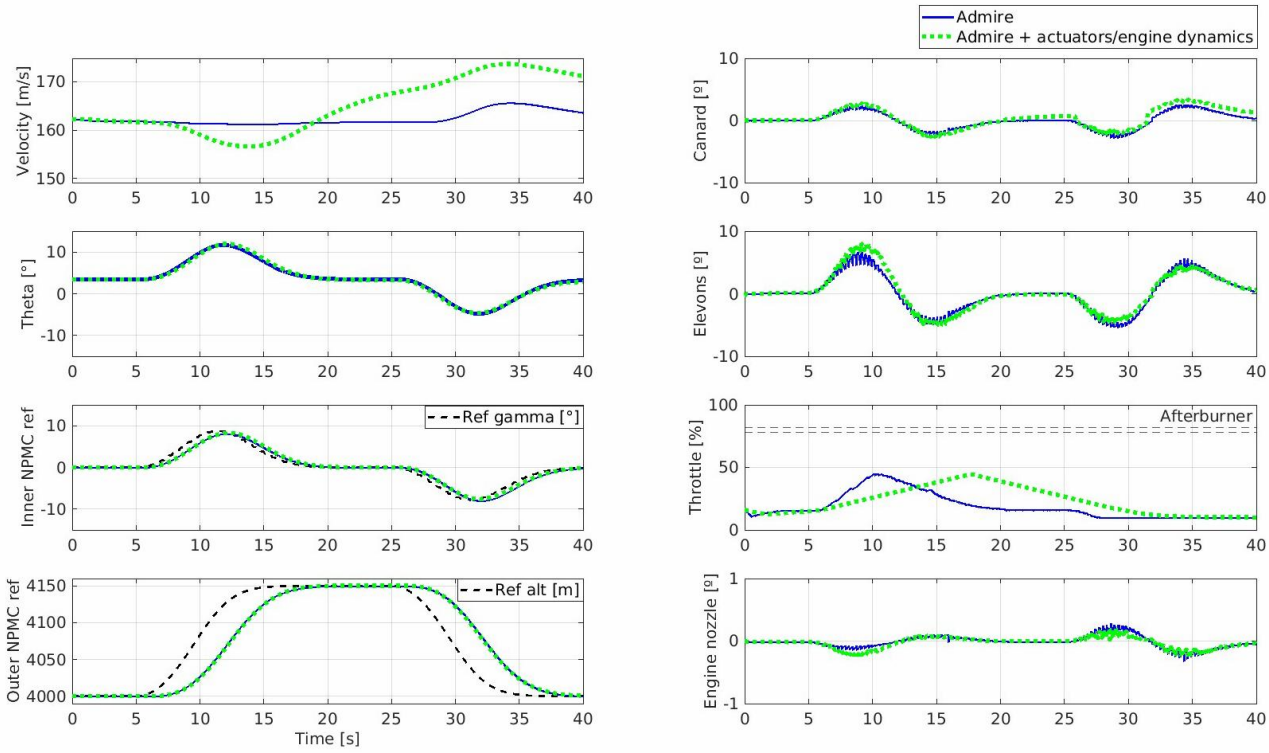


Figure 6 – Simulation of ADMIRE - NMPC - Climb and descent performance with actuator penalties including actuator/engine dynamics

5.2 Climb followed by canard hardover

As discussed previously, NMPC provides a suitable framework for fault accommodation through control redistribution and constraint management. The canard is the largest control surface on the ADMIRE model, with 3.2 m^2 against 1.35 m^2 and 1.2 m^2 of the outboard and inboard elevons, respectively. To test NMPC in an extreme scenario, which may be a catastrophic event, a hardover will be applied while climbing to force canard deflection from 25° to -55° . Besides, after hardover detection, NMPC will level off and reduce the velocity to 100 m/s (initial guess of safety velocity in this scenario). It is an attempt to translate a possible pilot action in this kind of situation (e.g., aviate before navigating or communicating) while giving time to the pilot to understand the situation (situation awareness).

- t = 0s, altitude-hold
- t = 5s, climb 150m in 10s (15 m/s)
- t = 10s, canard hardover
- t = 10.5s, canard hardover announcement
- t = 12.5s, aircraft levels off
- t = 26s, aircraft reaches 100 m/s

Overall, NMPC capabilities are impressive in dealing with a hardover, which took only 3 seconds and less than 30m of altitude after the hardover announcement to level off the aircraft. NMPC receives the current surface positions, and when the hardover occurs, the NMPC reacts to the resulting aircraft state deviations before the fault is declared by the FDD system.

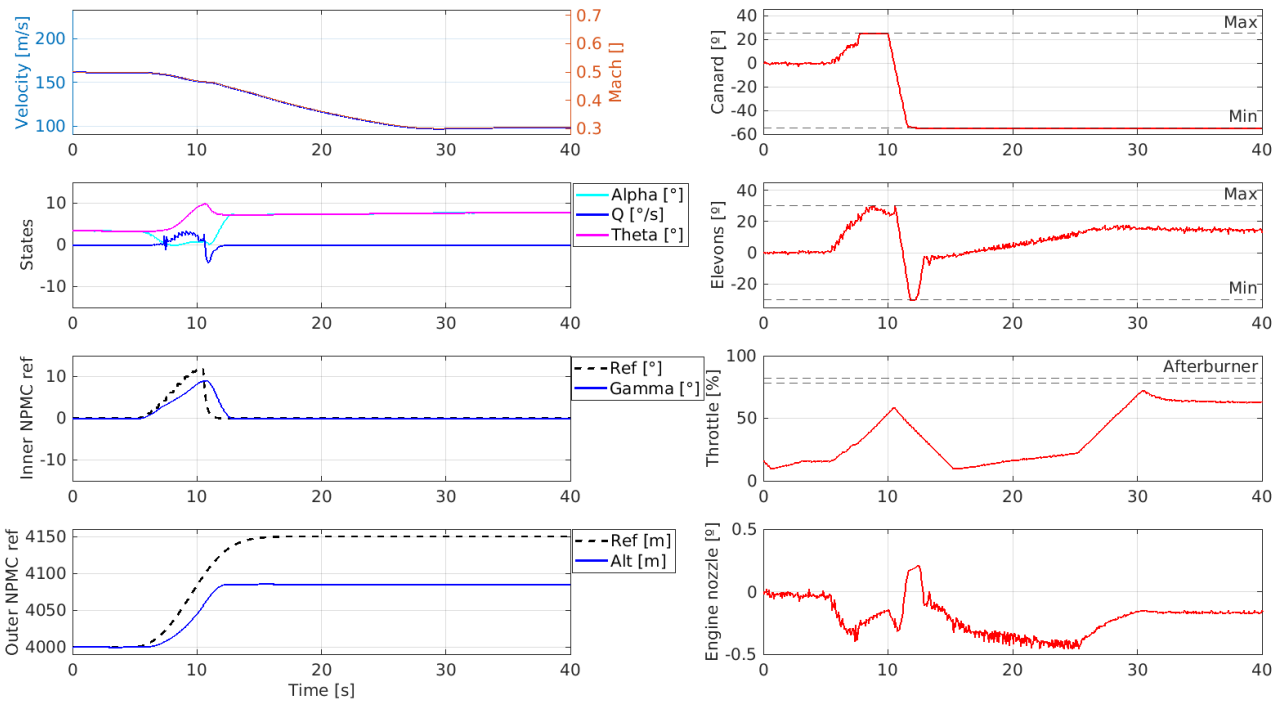


Figure 7 – Simulation of ADMIRE - NMPC - Climbing followed by canard hardover

5.3 Descending followed by canard hardover

A similar test was performed while descending:

- t = 0s, altitude-hold
- t = 5s, descend 150m in 10s (15 m/s)
- t = 10s, canard hardover
- t = 10.5s, canard hardover announcement
- t = 12.5s, aircraft levels off
- t = 26s, aircraft reaches 100 m/s

At the moment of hardover announcement, the aircraft descended at around 15 m/s and took less than 3 seconds and 35m of height to level off. The NMPC can stabilize the aircraft using the remaining resources even when the fault contributes to a downward tendency.
An intriguing result is how a small deflection of the engine nozzle, in the order of 0.5 degrees, generates a considerable amount of pitch moment to counteract canard hardover.

6. Conclusion

This paper presents the design and validation of autopilot functions by applying the NMPC technique to the nonlinear ADMIRE model, with performance evaluated in extreme scenarios. The nonlinear model allows the NMPC to acquire complete information on flight dynamics and find an optimum solution. Even when actuator and engine dynamics are present in the plant but not in the prediction model, it can follow the references while stabilizing the aircraft. Overall, the NMPC performed adequately and demonstrated fault accommodation capabilities under the considered hardover scenario. This architecture is robust and straightforward in tracking a feasible altitude reference even in a failure scenario. The simulations show that the proposed architecture satisfactorily recovers from a canard hardover, considering an announcement delay of 0.5 seconds. In the inner loop, the hard constraints are respected, and the soft constraints are respected whenever possible. In the outer loop, the problem was not sufficiently constrained to fully exploit the potential of NMPC. Additional tests are required to properly evaluate the actual benefits of employing a second NMPC. The cost function

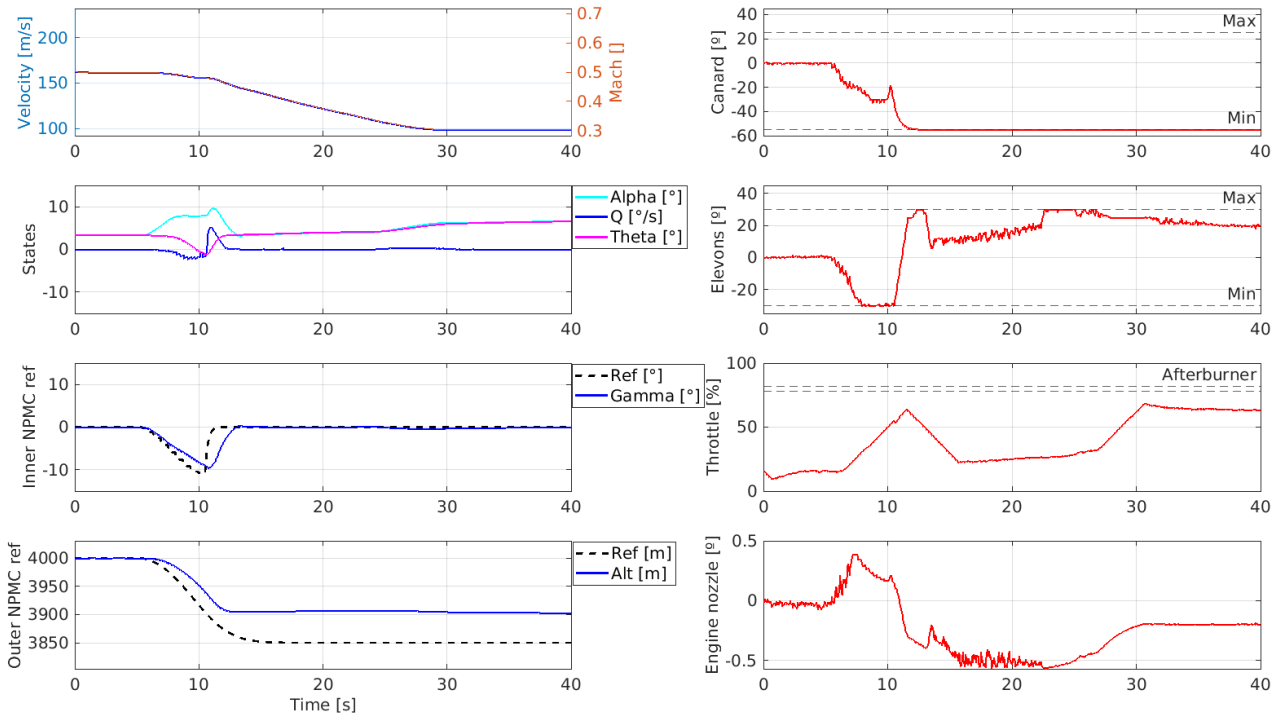


Figure 8 – Simulation of ADMIRE - NMPC - Descent followed by canard hardover

terms and their weight reflect the plant's behavior. For example, afterburner tends to be activated more frequently as the engine's weight is lowered. Although the results are promising, the NMPC real-time application for fast dynamic systems is challenging.

The prediction horizon and sampling time are a trade-off between the dynamics to be captured and computational resources. Solving a nonlinear programming problem consumes extensive computational resources, and up to 60 seconds were recorded for a one-second simulation time. Therefore, the present implementation should be regarded as a proof-of-concept rather than a real-time flight implementation.

The MPC takes a different approach to designing control laws in industry, such as PID and LQR. It shifts the focus to the models and directly incorporates operational constraints and restrictions of the SISO/MIMO system into the optimization process. Therefore, the manipulated variables are physically realistic and implementable, which is a valuable characteristic in fault-tolerant and safety-critical systems.

7. Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.

References

[1] Daniel Simon, Ola Härkegård, and Johan Löfberg. Angle of attack and load factor limiting in fighter aircraft using command governors. 01 2017.

[2] MM Kale and AJ Chipperfield. Stabilized mpc formulations for robust reconfigurable flight control. *Control Engineering Practice*, 13(6):771–788, 2005.

[3] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[4] A. Ferramosca, D. Limon, I. Alvarado, T. Alamo, and E.F. Camacho. Mpc for tracking of constrained nonlinear systems. In *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 7978–7983, 2009.

[5] Karl-Erik Modin and Clareus. Aerodynamic design evolution of the saab 39 gripen aircraft. In *Aircraft Design and Operations Meeting*, page 3195.

[6] Lars S. Forssell and Ulrik Nilsson. Admire the aero-data model in a research environment version 4.0. Technical report, Swedish Defence Research Agency, Stockholm, Sweden, 2005.

[7] *Sequential Quadratic Programming*, pages 529–562. Springer New York, New York, NY, 2006.

[8] Arda Çalışkan, D Yildiz, and V Kargin. Extensive analysis of hardover and trim-runaway failures on tluh mathematical model based on cs-29 requirements. 2018.

[9] Shawn T Donley and Valentine A Freitag. Autopilot hardover failure protection system, August 23 1977. US Patent 4,043,526.

[10] Ola Härkegård. Dynamic control allocation using constrained quadratic programming. *Journal of Guidance, Control, and Dynamics*, 27(6):1028–1034, 2004.

8. Contact Author Email Address

Marcos Toshio Ueda: ueda_toshio@hotmail.com