INTERNATIONAL JOURNAL OF ROBUST AND NONLINEAR CONTROL

Int. J. Robust. Nonlinear Control 2010; **00**:2–29

Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/rnc

Fault-Tolerant Reference Generation for Model Predictive Control with Active Diagnosis of Elevator Jamming Faults

L. Ferranti[1]*, Y. Wan[2], and T. Keviczky[1]

[1]*Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD, Delft, The Netherlands.*

[2] *Massachusetts Institute of Technology, 77 Massachusetts Ave, 02139, Cambridge (MA), USA.*

SUMMARY

This paper focuses on the longitudinal control of an Airbus passenger aircraft in the presence of elevator jamming faults. In particular, in this paper, we address permanent and temporary actuator jamming faults using a novel reconfigurable fault-tolerant predictive control design. Due to their different consequences on the available control authority and fault duration, the above two actuator jamming faults need to be distinguished so that appropriate control reconfigurations can be adopted accordingly. Their similarity in symptoms, however, prevents effective discrimination of the root cause of the jamming when using only a passive fault-diagnosis approach. Hence, we propose the use of model predictive control (MPC) as fault-tolerant controller to actively help the fault-detection (FD) unit discriminate between a permanent and a temporary jamming fault, while ensuring the performance of the aircraft. The MPC controller and FD unit closely interact during the detection and diagnosis phases. In particular, every time a fault is detected, the FD module commands the MPC controller to perform a predefined sequence of reconfigurations to diagnose the root cause of the fault. An artificial reference signal that accounts for changes in the actuator operative ranges is used to guide the system through this sequence of reconfigurations. Our strategy is demonstrated on an Airbus passenger aircraft simulator. Copyright © 2010 John Wiley & Sons, Ltd.

KEY WORDS: Fault-tolerant control; predictive control; model-based control; active diagnosis; reconfigurable control; reference generator; flight control; aerospace.

1. INTRODUCTION

The ability to automatically handle faults and component malfunctions while preserving overall performance is the main characteristic of a fault-tolerant control (FTC) system [1]. Fault-tolerant control systems have been largely investigated in the context of flight control taking into account the occurrence of faults on sensors and actuators [2, 3, 4, 5, 6, 7, 8, 9].

In this work, we focus on faults that can occur on the aircraft actuators (i.e., actuator jamming faults). Actuator jamming faults have long been investigated in the field of fault-tolerant flight control (e.g., [10, 3, 11, 8]). Among other techniques, we focus on the use of model predictive control (MPC) as fault-tolerant control. MPC provides a well-recognized framework for fault tolerance [12, 13, 10, 14]. On one hand, MPC (even) without reconfiguration has some inherent *self-reconfiguration* properties that allows one to reallocate the control effort in the presence of actuator faults [15]. On the other hand, reconfigurable MPC further improves fault tolerance capabilities by exploiting extra fault information in a structured manner, especially when it comes to dealing with constraints [15].

In practical applications, the control design has to take into account that the information concerning the fault is provided by a fault-detection (FD) module. Hence, in these scenarios, the design of a reconfigurable MPC controller must be integrated with a FD module. Robustness and guaranteed fault tolerance of this integrated fault-tolerant MPC (FTMPC) scheme was analyzed with set theoretic methods in [16, 17].

In most literature, actuator jamming is attributed to a permanent jamming (or *stuck fault*), during which the actuator is locked at a certain position. The study of temporary jamming due dynamics manoeuvres (combined with the presence of heavy aerodynamic forces), however, has been only investigated by few researchers (e.g., the authors of [11] propose a sliding mode fault tolerant control

*Correspondence to: Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD, Delft, The Netherlands. E-mail: l.ferranti@tudelft.nl

Contract/grant sponsor: European Union's Seventh Framework Programme FP7/2007-2013 entitled "Reconfiguration of Control in Flight for Integral Global Upset Recovery (RECONFIGURE)"; contract/grant number: AAT-2012-RTD-2314544.

Prepared using rncauth.cls DOI: 10.1002/rnc

scheme to detect and compensate the effects of the temporary and permanent jamming faults). This temporary jamming—known as *stall load* or blow-down [18, 11] for aerospace applications—leads to more stringent control limits for a bounded period of time. The original limitations of the actuators can be recovered once either the control command is consequently adjusted or the aerodynamic forces become smaller [18]. Although both stuck fault and stall load lead to a jammed actuator, their consequences on the control limits and jamming duration are significantly different. Therefore, we must be able to identify the root cause of the actuator jamming (i.e., identify whether the actuator is temporarily or permanently jammed). Furthermore, in case of stall load, we must be able to determine its end to apply suitable reconfiguration strategies from the control design perspective.

Conventional FD cannot achieve this goal because the fault phenomena of a permanent or temporary jamming have high similarity. We propose to integrate reconfigurable MPC with active FD to address the challenge above. Instead of passively monitoring actuator behaviors, we exploit a sequence of reconfiguration strategies using the MPC controller to assist the FD module, not only to distinguish the root cause of the actuator jamming, but also to actively detect the end of a stall load (in case of a temporary jamming). Then, once the root cause of the jamming is detected, the MPC controller adopts suitable successive reconfigurations, aimed to improve the overall control performance. All these improvements from both FD and control perspective cannot be achieved without using active reconfigurations to assist FD.

The use of active FD in the context of FTMPC has been rather limited so far and focused only on permanent faults [19, 20, 21]. In contrast, our contribution lies in discriminating between a permanent and temporary jamming (i.e., stuck fault and stall load, respectively) that share highly similar fault symptoms. Compared to the approach we proposed in [22], we rely on *(i)* an improved FD strategy, *(ii)* a different MPC formulation for tracking, and *(iii)* a modified disturbance observer to incorporate plant-model mismatches. From the detection perspective, in [22] the FD unit relies only on the information from a single control surface, without exploiting the actuator redundancy. In this work, we combine the detection strategy in [22] with an additional check that compares the behavior of the single elevator with the others. This has the additional benefit that if only one (or two)

Prepared using rncauth.cls
DOI: 10.1002/rnc

control surfaces are subject to faults, the fault can be detected quickly by monitoring the deviation of the residual signal from the normal behavior of the others. This strategy is useful especially for permanent jamming faults that are more likely to involve only one control surface. Temporary faults that are more likely to affect all the control surfaces can still be detected by monitoring if the residual signal of each actuator exceeds a predetermined threshold. From the control perspective, in [22] we made the assumption that the desired reference during a manoeuvre could not lead to infeasible solutions and all the control reconfigurations were performed on the actuator constraints directly, without affecting the desired reference signal. In contrast to [22], in this work we exploit a strategy similar to the artificial reference tracking proposed by [23, 24]. In [23, 24], the concept of artificial reference is used to enlarge the region of attraction of the proposed control while ensuring closed-loop stability guarantees. We reinterpret this idea for fault-tolerant control purposes. In particular, this approach can be used to compute artificial reference signals for the state and the actuator commands in order to compensate for the occurrence of faults that can suddenly affect the feasible region of the MPC controller. In particular, the sequence of reconfigurations used to detect and diagnose the root cause of the jamming is not performed directly on the actuators' constraints, but on the constraints associated with the artificial reference signal. By doing so, when a fault is detected the reference followed by the states and the actuators is adapted to the faulty feasible region. Consequently, if the desired reference signal becomes unfeasible in the presence of a fault, the artificial reference acts as a *fault-tolerant* reference signal to avoid infeasibility (and possible instability) issues. Finally, compared to [22], we incorporate the effects of plant-model mismatches directly in the definition of the artificial reference constraints using the information provided by an improved disturbance estimator module. We demonstrate the effectiveness of our approach using an Airbus civil aircraft simulator [25].

In the following, Section 2 presents the Airbus simulator used to evaluate our design. Section 3 describes our fault-tolerant control architecture. Section 4 introduces the proposed detection and diagnosis strategy and highlights the interactions between the FD module and the MPC. Section 5

Prepared using rncauth.cls
DOI: 10.1002/rnc

compares the behavior of the MPC with and without the proposed active reconfigurations when multiple faults occur on the elevators. Finally, Section 6 concludes this paper.

2. BENCHMARK MODEL AND SCENARIO DEFINITION

This section describes the RECONFIGURE benchmark model, that is, an Airbus civil aircraft simulator [25] (Section 2.1), and details the actuator fault scenarios we focus on in this work (Section 2.2).

2.1. The aircraft longitudinal model

This work focuses on the longitudinal control of an Airbus passenger aircraft in the presence of actuator jamming faults. Our proposed FTC architecture relies on MPC, which is a model-based technique. Hence, a mathematical description of the longitudinal dynamics of the aircraft (i.e., the model) is necessary to ensure performance of our FTC scheme. In this respect, in the control design phase, we can rely on linearized aircraft models at given operating points (or *trim conditions*) to build the prediction model of the MPC controller. In the following, we describe the augmented aircraft model (i.e., the cascade actuator-aircraft dynamics depicted in Figure 2) and introduce the notation used to design our MPC control (Section 3).

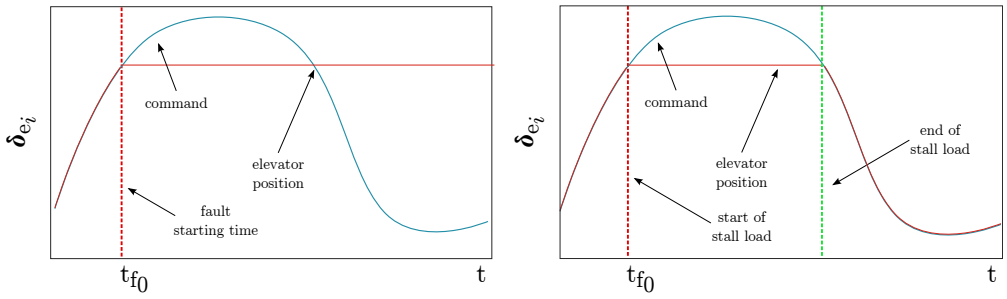
The linearized and discretized longitudinal dynamics of the aircraft can be described as follows:

$$x_{\text{A/C}}(t+1) = A_{\text{A/C}}x(t) + B_{\text{A/C}}u_{\text{A/C}}(t) \tag{1a}$$

$$y_{\text{A/C}}(t) = C_{\text{A/C}}x(t) + D_{\text{A/C}}u_{\text{A/C}}(t), \tag{1b}$$

where $x_{\text{A/C}} := [q\, p\, v\, \alpha\, \vartheta\, h]^{\text{T}} \in \mathcal{X}_{\text{A/C}} \subseteq \mathbb{R}^{n_{\text{A/C}}}$ is the state vector, which includes the pitch rate, roll rate, ground speed, angle of attack, pitch angle, and altitude, respectively, $u_{\text{A/C}} := [\delta_{\text{e}_{\text{li}}}\, \delta_{\text{e}_{\text{ri}}}\, \delta_{\text{e}_{\text{lo}}}\, \delta_{\text{e}_{\text{ro}}}] \in \mathcal{U} \subseteq R^{n_{\text{u}}}$ is the control input with $\delta_{\text{e}_{\text{li}}}$, $\delta_{\text{e}_{\text{ri}}}$, $\delta_{\text{e}_{\text{lo}}}$, and $\delta_{\text{e}_{\text{ro}}}$ representing the left inner, right inner, left outer and right outer elevator deflections, respectively, and $y_{\text{A/C}} := [n_{\text{z}}\, x^{\text{T}}]^{\text{T}} \in \mathcal{Y}_{\text{A/C}} \subseteq \mathbb{R}^{n_{y_{\text{A/C}}}}$ is the output vector with n_{z} representing the vertical load factor, which is a quantity related to the acceleration on

Prepared using rncauth.cls
DOI: 10.1002/rnc



(a) Stuck Fault. (b) Stall load.

Figure 1. Overview of the elevator jamming scenarios considered in the paper.

the vertical axis. All the states describing the longitudinal dynamics are measurable using dedicated sensors. These measurements are, however, affected by delays that must be compensated in the control design (Section 3).

The elevator dynamics in the RECONFIGURE benchmark model can be modeled as third-order linear time-invariant (LTI) systems. The following model describes the elevator dynamics:

$$x_{\text{el}}(t+1) = A_{\text{el}}x_{\text{el}}(t) + B_{\text{el}}u(t) \tag{2a}$$

$$y_{\text{el}}(t) = C_{\text{el}}x_{\text{el}}(t) + D_{\text{el}}u(t) \tag{2b}$$

where $x_{\text{el}} \in \mathcal{X}_{\text{el}} \in \mathbb{R}^{n_{\text{el}}}$ (the components of x_{el} are the elevator position, velocity, and acceleration), $u \in \mathcal{U}_{\text{MPC}} \subseteq \mathbb{R}^{n_u}$, and $y_{\text{el}} \equiv u_{\text{A/C}}$ (i.e., the elevator position).

Finally, we assume that $\mathcal{X}$, $\mathcal{U}$, $\mathcal{Y}$, $\mathcal{X}_{\text{el}}$, and $\mathcal{U}_{\text{MPC}}$ are polyhedral sets that contain the origin in their interior. Furthermore, in the remainder of the paper, we use $\overline{\delta}_{\text{e}_i}$ and $\underline{\delta}_{\text{e}_i}$ to indicate the upper and the lower bounds of the i-th elevator output δ_{e_i} ($i \in \mathcal{I} := \{\text{li},\ \text{ri},\ \text{lo},\ \text{ro}\}$).

2.2. *Fault Description*

This work focuses on elevator jamming scenarios. In these scenarios, one or more elevators remain fixed at an unpredictable value $\delta^{\text{f}}_{\text{e}_i}$ ($i \in \mathcal{I}$), which might differ from their normal saturation limits. The elevator jamming can be attributed to two different root causes exemplified in Figure 1:

Prepared using rncauth.cls DOI: 10.1002/rnc

- *Stuck Fault*. The elevator is permanently jammed at a certain position $\delta^{\mathrm{f}}_{\mathrm{e}_i}$ and cannot be recovered (Figure 1a). This effect can be modelled as a permanent change at time t_{f} in the elevator's upper and lower operating bounds that become both equal to the jammed position $\delta^{\mathrm{f}}_{\mathrm{e}_i}$ $\forall t \geq t_{\mathrm{f}}$.
- *Stall load* [18]. The elevator is temporarily jammed during a dynamic manoeuvre, due to heavy aerodynamic forces preventing the elevator to achieve its commanded control surface deflection (Figure 1b). In this situation, the elevator can still move within its reduced control limits $[-\underline{\delta}_{\mathrm{e}_i}, \delta^{\mathrm{f}}_{\mathrm{e}_i}]$ or $[-\delta^{\mathrm{f}}_{\mathrm{e}_i}, \overline{\delta}_{\mathrm{e}_i}]$, determined by the jammed position $\delta^{\mathrm{f}}_{\mathrm{e}_i}$. The stall load ends if either the manoeuvre becomes less dynamic or the aerodynamic forces acting on the control surface become smaller.

Considering their different consequences on the control limits and jamming duration, a stuck elevator and stall load need to be distinguished and require adopting different reconfiguration strategies in FTC. Nevertheless, because of the high similarity in the jamming phenomena, it is difficult to distinguish these two root causes. Hence, our proposed integrated FTC approach actively modifies the control strategies to help the FD module discriminate between the two root causes of the jamming, as detailed in Section 4.

Remark 1

This work focuses on jamming faults for which it is nontrivial to distinguish the root cause of the jamming. Although in some practical situations the stall load limits might change overtime leading to control challenges, from the diagnosis point of view we can still distinguish the root cause of the jamming easily in this case (when the fault is detected it is evident that the actuator is not permanently stuck at a given position). Hence, given that our goal is to design the interactions between the FD unit and the MPC controller to diagnose the root cause of a jamming fault, we do not focus on stall load scenarios with time-varying limits.

Prepared using rncauth.cls
DOI: 10.1002/rnc

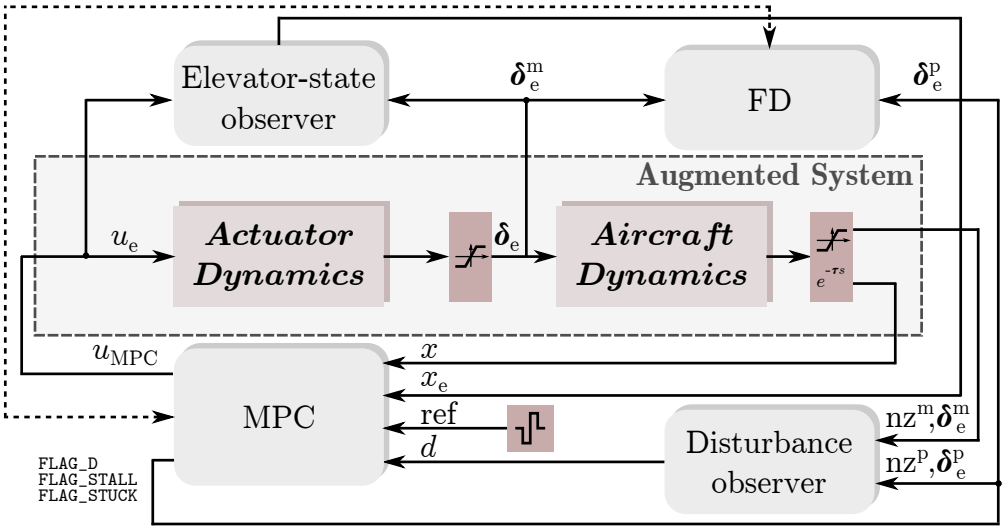


Figure 2. Proposed control architecture.

3. FTC ARCHITECTURE

This section focuses on our proposed FTC architecture. In this respect, Figure 2 provides an overview of our proposed FTC design and show the interactions among the different components of our control system and the controlled plant. In particular, Figure 2 highlights *(i)* in dark grey the main components of the plant (i.e., the augmented aircraft model described in Section 2.1, the constraints depicted as saturation blocks, and the sensor delays) and *(ii)* in light grey the main components of our fault-tolerant controller. A detailed description of these components is provided in the remainder of the section.

3.1. Elevator-state observer

The elevator states are needed by the MPC controller to build the predictions. By using the elevator model (2), four Luenberger observers [26], characterized by a constant gain L, are constructed. The gain L is the same for all the operating points, given that the elevators are LTI systems (according to their description in the RECONFIGURE model). Each observer independently monitors one elevator. On one hand, the elevator-state estimates are needed to exploit the elevator dynamics in the MPC problem formulation. On the other hand, these elevator-state estimates are used to compute predicted elevator outputs δ_e^p for the disturbance observer and the FD module.

Prepared using rncauth.cls
DOI: 10.1002/rnc

The realization we adopt for the elevators is such that, for each elevator, the state associated with the elevator position corresponds to the output of the elevator. Hence, when a saturation is detected on the i-th elevator position, the other two states (associated with the velocity and acceleration of the i-th elevator) are set to zero and the estimated position value is set to the measured elevator output. This allows us to estimate the elevator states without requiring a more advanced state estimator to handle saturation.

Note that if the model of the elevators is nonlinear or depends on the flight condition the gain L should also vary accordingly. As previously stated, in this work we adopt the elevator description provided in the RECONFIGURE benchmark model, which assumes the elevators to be LTI systems.

3.2. Disturbance observer

The disturbance observer is used to compensate constant measurement errors, reduce the effects of plant-model mismatches, and provide useful information to help the FD module detect jamming faults. The proposed observer strongly relies on the information provided by the MPC controller and on the plant measurements.

The observer is composed by two modules used to compensate *(i)* measurement errors and *(ii)* plant-model mismatches, respectively. In particular, the first module estimates a constant disturbance signal (that is then used by the MPC controller) as follows. First, we take into account that the MPC controller does not model the sensor and filter dynamics in the predictor to reduce the number of decision variables (and, consequently, the computation time). Hence, the proposed observer monitors $e_{n_z} := n_{\mathrm{z}}^{\mathrm{m}} - n_{\mathrm{z}}^{\mathrm{p}}$, that is, the mismatch between the measured and the predicted load factor. Second, the observer monitors $e_{\delta_{\mathrm{e}_i}} := \delta_{\mathrm{e}_i}^{\mathrm{m}} - \delta_{\mathrm{e}_i}^{\mathrm{p}}$, that is, the mismatch between the measured and the predicted elevator outputs, for elevator-jamming detection purposes. Hence, the first module of the disturbance observer estimates $d := [d_{n_z}\ d_{\mathrm{e}}^{\mathrm{T}}]^{\mathrm{T}}$ as follows:

$$d(t+1) = d(t) + \begin{bmatrix} e_{n_z} \\ \\ e_{\delta_{\mathrm{e}_i}} \end{bmatrix}. \tag{3}$$

Prepared using rncauth.cls
DOI: 10.1002/rnc

This estimated disturbance $d \in \mathbb{R}^{n_d}$ ($n_d = 5$) affects the predicted elevator outputs, the aircraft states, and the aircraft outputs. Hence, we must consider this disturbance as an additional state in the MPC prediction model as explained below.

The second module of the disturbance observer takes into account plant-model mismatches and, eventually, nonlinearities in the plant that are not modelled in the MPC controller, given that only linearized plant models are used to build the predictions. In this respect, we define an upper bound on these plant-model mismatches as $\epsilon_{\text{nl}} := \|\hat{x}_t - x_{t|t-1}\|_2$, where $\hat{x}_t$ is the measured state of the aircraft (we omitted the subscript A/C to simplify the discussion) at time t and $x_{t|t-1}$ is the value of the state at time t predicted (by the MPC controller) according to the value of the measured state at time $t-1$. This upper bound monitors the distance between the predicted behavior of the plant and the real behavior and can be used (as explained below) to design a robust reference signal to avoid constraint violations in the MPC problem formulation.

Remark 2

The strategy described in (3) can only be used to estimate disturbances that can be modeled as constant values. Hence, given that the plant-model mismatches and the nonlinearities in the plant cannot be modelled as constant disturbances, we decided to include their effects in the definition of the MPC constraints as explained below.

3.3. Fault-detection module

The Fault-Detection (FD) module relies on the elevator-output prediction error $e_{\delta_{\text{e}_i}}$ to compute the residual signal used for the detection of jamming faults. The generated residual for each elevator is evaluated by its root mean square (RMS) value

$$J_i(t) := \sqrt{\frac{1}{N_{\text{eval}}} \sum_{k=t-N_{\text{eval}}+1}^{t} e^2_{\delta_{\text{e}_i}}(k)}\,, \qquad i \in \mathcal{I} \tag{4}$$

over a sliding window $[t - N_{\text{eval}} + 1, t]$. N_{eval} is selected according to the slowest mode of the actuators. This is an empirical choice to give sufficient time to the physical system to register the

Prepared using rncauth.cls DOI: 10.1002/rnc

jamming fault. The choice of N_{eval} is a trade-off between reducing the risks of miss detection/false alarms and detection delay.

The fault detection decision is made by comparing each residual evaluation value $J_i(t)$ with the related threshold J_i^{th}, that is,

$$\text{FD Logic} : \begin{cases} J_i(t) \leq J_i^{\text{th}} & \Rightarrow \quad \text{fault-free in elevator } i \\ J_i(t) > J_i^{\text{th}} & \Rightarrow \quad \text{jamming in elevator } i. \end{cases} \tag{5}$$

After fixing the length of the sliding evaluation window, the thresholds $\{J_i(t)\}$ are determined by the plant-model mismatch of the elevator model (2). In practice, each threshold J_i^{th} can be selected as the peak value of $J_i(t)$ in a large set of fault-free scenarios. In this work, we determine the thresholds by using dynamic fault-free manoeuvre, that is, when stall loads might be more likely to occur. Its choice is a trade-off between reducing the miss detections/false alarms and, at the same time, reducing detection delays.

Remark 3

Note that in this work we rely on a simple fault-detection logic with fixed J_i^{th} to present our integrated approach. Nevertheless, the proposed approach can be extended with the use of more sophisticated detection techniques to select the threshold J_i^{th} (for example, when an explicit description of multiplicative model uncertainties is taken into account).

Furthermore, we add an additional check to improve the detection of isolated faults for which we can exploit redundancy, that is, the presence of redundant control surfaces. In fault-free conditions, the residual signal of each elevators are sufficiently small and close to each others (in terms of magnitude). Suppose that one of the residual signals starts deviating from the others. This abnormal behavior is an indicator that the elevator associated with that residual signal might be jammed. This strategy is useful when we have to deal with isolated faults on one or two actuators. For example, this strategy is useful to anticipate the detection of a stuck fault, because a permanent jamming is more likely to occur on a single elevator.

Prepared using rncauth.cls
DOI: 10.1002/rnc

Remark 4

The detection logic described above is insufficient to identify the root cause of jamming by itself given that it only informs the controller that the actuator is jammed. At this stage the controller does not know whether the jamming is permanent or temporary. In Section 4, we combine the detection logic (5) with different active reconfigurations to capture more detailed fault information.

3.4. Model Predictive Controller

MPC controllers rely on *(i)* the plant description to build predictions of the plant behavior over a predefined time window (called prediction horizon), *(ii)* the information on state, input, and output constraints, and *(iii)* current measurements from the plant, such as state measurements and desired reference signals. These controllers offer an intuitive and structured framework to compute the optimal control law to simultaneously satisfy the control objectives and constraints on the plant. This control law is computed by solving (either offline or online [27, 28, 29, 30, 31], depending on the number of decision variables) an optimization problem (usually a quadratic programming problem). For more details on MPC refer to [32, 33, 34] and the references within.

Remark 5

In this work, we solve the MPC optimization problem online. This requires solving a quadratic programming (QP) problem of size proportional to the number of decisions variables and length of the prediction horizon. The solution of this optimization problem in an embedded environment can be challenging, due to small sampling times and limited hardware and software resources (the availability of a QP solver is usually not guaranteed). First-order solvers, such as proximal-gradient and splitting methods (refers to [35, 36] and the references within for an overview) are valid solutions for this problem. In this respect, in the context of aerospace applications, in [37], we show on the RECONFIGURE benchmark model how we can efficiently compute the MPC problem by relying on these first-order solvers (in particular, by combining the use of Nesterov's dual fast gradient and the alternating direction method of multipliers).

Prepared using rncauth.cls
DOI: 10.1002/rnc

With this framework in mind, we define the model used to compute the predictions in the the MPC controller. In particular, given (1)-(2), this model is computed as follows:

$$x(t+1) = Ax(t) + Bu(t) \quad t \geq 0, \tag{6a}$$

$$y(t) = Cx(t) + Du(t) \quad t \geq 0, \tag{6b}$$

where $x := [\bar{x}_{\text{A/C}}^{\text{T}}\, x_{\text{el}}^{\text{T}}\, \hat{d}^{\text{T}}]^{\text{T}} \in \mathcal{X}_{\text{MPC}} \subseteq \mathbb{R}^n$ (where $\bar{x}_{\text{A/C}}^{\text{T}} := [q\, v\, \alpha\, h]$ takes into account a subset of the longitudinal states to maintain the size of the prediction model small and $n := n_{\text{A/C}} - 2 + n_{\text{el}} + n_d$), and $y := [y_{\text{A/C}}^{\text{T}}\, y_{\text{el}}^{\text{T}}]^{\text{T}} \in \mathcal{Y}_{\text{MPC}} := \mathcal{Y} \times \mathcal{U} \subseteq \mathbb{R}^{n_{y_{\text{A/C}}} + n_u}$. The structure of $A,\ B,\ C$, and D follows from the choice of the state, input, and output for the cascade actuator-aircraft dynamics depicted in Figure 2 (namely, the augmented system) and by describing the disturbance dynamics as constant, that is, $\hat{d}(t+1) = \hat{d}(t)$, where $\hat{d}(t) = d(t)$ (3).

Remark 6

Note that we use linearized aircraft models in the MPC problem formulation (as described in Section 2.1 as well) to explain our algorithm. Nevertheless, the approach can potentially be extended to linear-parameter varying (LPV) or linear time-varying (LTV) models [38, 39, 40, 41].

In the remainder of the paper, we consider the following assumption:

Assumption 1

The augmented system is stabilizable.

Our goal is to control the longitudinal dynamics of the aircraft. In particular, our goal is to steer the output of system (6) to a desired reference value denoted by ν, which is generated by a pilot stick command. The reference value is measured at each sampling time and we assume that is constant along the length of the prediction horizon in the MPC problem formulation. Furthermore, we have to take into account the constraints acting on state, input, and output, that are, $\mathcal{X}_{\text{MPC}}, \mathcal{U}$, and $\mathcal{Y}_{\text{MPC}}$, respectively. Hence, compared to [42], we rely on a modified version of the MPC for tracking

Prepared using rncauth.cls
DOI: 10.1002/rnc

formulation proposed in [23, 24]. In particular, we can formulate our MPC problem as follows:

$$\mathcal{V}^*(\nu, x_{\text{init}}) := \underset{x,u,\theta}{\text{minimize}} \sum_{t=0}^{N} l_t(\nu, x_t, u_t, \theta_t) \tag{7a}$$

$$\text{subject to: } Ax_t + Bu_t = x_{t+1}, \quad t = 0, \dots, N, \tag{7b}$$

$$\begin{bmatrix} \hat{x}_t \\ \hat{u}_t \end{bmatrix} = M_\theta \theta_t, \quad t = 0, \dots, N, \tag{7c}$$

$$G_x x_t + G_u u_t + g \leq 0 \quad t = 0, \dots, N, \tag{7d}$$

$$G_x \hat{x}_t + G_u \hat{u}_t + g_\theta + E\epsilon_{\text{nl}} \leq 0 \quad t = 0, \dots, N, \tag{7e}$$

$$s \geq 0 \tag{7f}$$

$$\hat{y}_t = N_\theta \theta_t \quad t = 0, \dots, N, \tag{7g}$$

$$x_0 := x_{\text{init}}, \tag{7h}$$

where $x_t \in \mathbb{R}^n$, $u_t \in \mathbb{R}^{n_u}$ indicate the t-step-ahead state and control predictions, respectively. In addition, (7d) represents the constraints on the predicted state, input, and output ($G_x \in \mathbb{R}^{c \times n}$, $G_u \in \mathbb{R}^{c \times n_u}$, and $g_\theta = g$ in fault-free operating conditions) that follow from the definition of $\mathcal{X}_{\text{MPC}}$, $\mathcal{U}$, and $\mathcal{Y}_{\text{MPC}}$. Furthermore, $\theta_t \in \mathbb{R}^{n_u}$ is the vector of parameters used to generate the *artificial* steady state, input, and output $\hat{x}_t$, $\hat{u}_t$, and $\hat{y}_t$, respectively. M_θ and N_θ are suitable matrices (refer to [23] for details). For a prediction horizon of length N, the cost l_t in (7a) is described as follows:

$$l_t(\nu, x_t, u_t, \theta_t) := \|x_t - \hat{x}_t\|_Q^2 + \|u_t - \hat{u}_t\|_R^2 + \rho_1 \|\hat{y}_t - \nu\|_2^2, \tag{8}$$

where $Q = Q^{\text{T}} \in \mathbb{S}_{\geq 0}^n$, $R = R^{\text{T}} \in \mathbb{S}_{>0}^m$, and $\rho_1 > 0$.

The main idea of the artificial reference associated with the parameters θ_t in Problem (7) is to generate a reference for the states and the control inputs that achieves the control objectives (i.e., the tracking of the reference ν) while satisfying the constraints on the system. This strategy allows one to compromise between tracking performance and feasibility of the solution when the

Prepared using rncauth.cls
DOI: 10.1002/rnc

commanded reference ν does not lead to feasible state and control trajectories. In this respect, note that in the cost the distance between the desired reference and $\hat{y}_t$ is penalized by a quantity $\rho_1 > 0$ (which is a tuning parameter of our design) in order to generate an output trajectory close to the desired one. At the same time, the constraints (7e) prevent that the generated trajectory along the prediction horizon becomes infeasible. This strategy has the following advantage compared to the one proposed in [22]. At every problem instance, if a jamming fault is detected on the actuators, with a simple reconfiguration of the constraints on θ_t (i.e., by changing the definition of g_θ according to the severity of the fault, but without changing the initial feasible region of the states and control commands) we can generate a feasible reference signal for the state, input, and output that steers the system towards the new (post fault) feasible region. This reference signal is clearly suboptimal (note that we are using the 2-norm in (8) to penalize the distance from ν, which is not an exact penalty), but ensures a safer transition to the after-fault feasible region of the controller.

Remark 7

One concern when using this approach is related to the stability of the system controlled by the MPC controller. In [24] a terminal set for tracking is introduced in the MPC problem formulation to guarantee stability. When a jamming fault occurs, this impacts the definition of the terminal set that shrinks according to the severity of the fault. While a rigorous stability proof is out of the scope of this manuscript (our main focus is to provide a strategy for active diagnosis of jamming faults using control reconfiguration and, consequently, in the remainder of the paper, we consider maneuvers that do not impact the stability of the system), we provide different possible strategies/guidelines to design a robust MPC controller in the presence of faults:

1. The jamming faults can be considered as (possibly persistent) disturbances bounded in a given set $\mathcal{W}$ computed based on some heuristics (for example, by considering different fault combinations). The robust terminal set for tracking computed based on the worst combination of faults can then be used in the MPC formulation (leading to a tube-based MPC design [43] for tracking).

Prepared using rncauth.cls
DOI: 10.1002/rnc

2. If in the current setup we include a terminal set for tracking (according to [24]), when a fault occurs, the only reconfigurations in the MPC problem formulation affect the parameters θ used to generate the artificial reference signal. The optimizer computes the best artificial reference trajectory to compromise between tracking performance and constraint satisfaction. Hence, if we tighten (according to the severity of the fault) the constraints associated with the parameters θ this should directly prevent violation of the original terminal set for tracking (which remains unmodified for the states and control commands).
3. Alternatively, if we include a terminal set for tracking in the current MPC formulation (as in the previous point), a solution could be to tighten the terminal set by an amount proportional to the fault and uncertainties in the model. The terminal set associated with the augmented aircraft model takes into account also the dynamics of the actuators. Consequently changes in the actuator bounds will impact the dynamics and the choice of the associated tightening parameters.

An interesting alternative to be investigated (as part of our future research and out of the scope of this manuscript) is related to the use of infinite horizon MPC formulations [44, 45, 46], that are recently gaining increasing attention and can remove the requirements of a terminal set in the MPC problem formulation.

Note that the constraints on the artificial states (7e) are tightened (E is the matrix used to select the subset of state constraints where the tightening occurs), compared to (7d), by a quantity ϵ_{nl}, which is computed by the disturbance observer (presented in Section 3.2) at each sampling time. This additional tightening allows the controller to take into account the effects of the plant model-mismatches/nonlinearities, which are not modelled in the prediction model (7b) and cannot be modelled as constant disturbances (3). Consequently, the pairs $(\hat{x}_t, \hat{u}_t)$ are generated to take into account these plant-model mismatches leading to a *robust* artificial reference generation, without directly affecting the feasible region of the states and control inputs. Note that constraint tightening is a technique used in robust MPC to avoid infeasibility in the presence of disturbances (the interested reader can refer to [47] and the references within).

Prepared using rncauth.cls
DOI: 10.1002/rnc

In general, the MPC controller solves Problem (7) every time new measurements are available from the plant and returns an optimal sequence of states and control inputs that minimizes the cost (7a). Let the optimal sequence be defined as follows:

$$\{\mathbf{x}, \mathbf{u}, \boldsymbol{\theta}\} := \{x_0, \ldots, x_N^*, u_0^*, \ldots, u_{N-1}^*, \theta_0^*, \ldots, \theta_N^*\}. \tag{9}$$

Only the first element of $\mathbf{u}$ is implemented in closed-loop, that is, the control law obtained using the MPC controller is given by:

$$\kappa_{\text{MPC}}(\nu, x_{\text{init}}) = u_0^*, \tag{10}$$

and the closed-loop system is described by

$$x(t+1) = Ax(t) + B\kappa_{\text{MPC}}(\nu, x_{\text{init}}). \tag{11}$$

With this framework in mind, the next section details the interactions between the FD module and the MPC controller to actively detect and diagnose the root cause of jamming faults.

4. INTERACTION FD-MPC

This section aims to describe the close interactions between the FD module and the MPC controller (described in Sections 3.3 and 3.4, respectively) in our proposed integrated FTMPC approach. Figure 3 summarizes these interactions. In the following, we show how the fault information obtained by the FD module is exploited by the MPC controller and how the MPC controller actively modifies its reconfiguration strategies to assist the FD module in diagnosing the root cause of a detected elevator jamming.

4.1. Detection

As Figure 3 shows, during the detection phase, the FD module constantly monitors each elevator by evaluating its corresponding residual signal $e_{\delta_{e_i}}$ with J_i in(4) ($i \in \mathcal{I}$). If the residual evaluation signal

Prepared using rncauth.cls
DOI: 10.1002/rnc

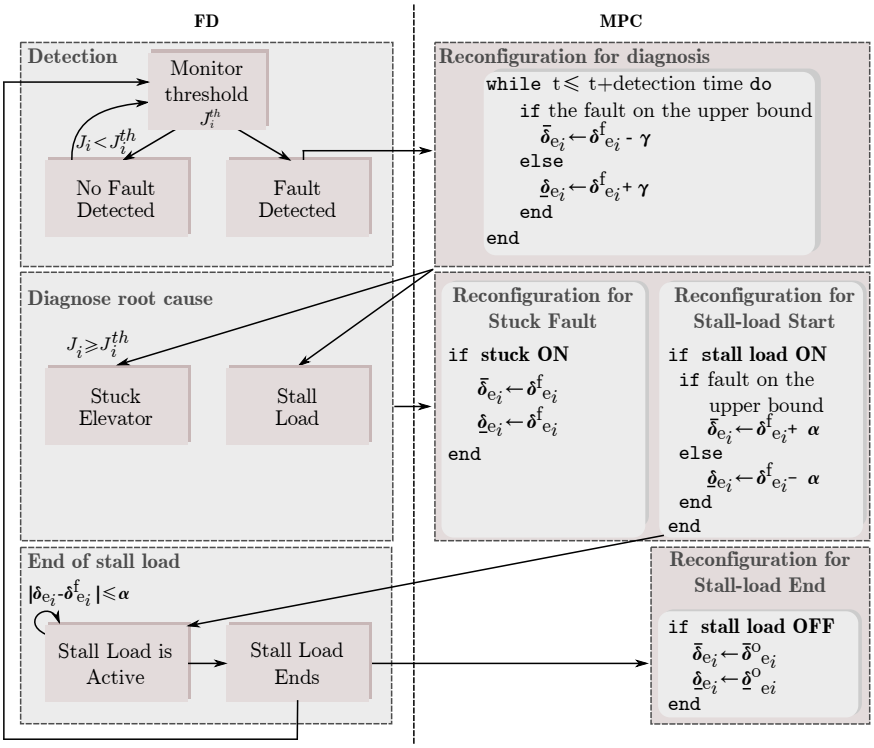


Figure 3. Description of the interaction FD-MPC.

J_i associated with the i-th elevator at time t_{f_i} exceeds the predefined threshold J_i^{th} or differ from the others as described in Section 3.3, the FD module detects that the i-th elevator is jammed. At this stage, the root cause of jamming is still unknown. Hence, the FD module sends a message to the MPC controller to activate the first reconfiguration (i.e., reconfiguration for diagnosis in Figure 3).

4.2. *Reconfiguration for diagnosis*

The aim of the reconfiguration for diagnosis is to help the FD module understand the root cause of the jamming fault. The MPC controller checks the sign of $e_{\delta_{e_i}}$ at time t_{f_i} to decide whether to modify $\overline{\delta}_{e_i}$ or $\underline{\delta}_{e_i}$, that is, the upper or the lower bounds of the i-th elevator. Note that this modification in the MPC problem formulation affects only the definition of g_θ (i.e., the feasible region of the parameters θ used to generate the artificial reference signal). The idea is to temporarily set the jammed elevator bound to a tightened value $\delta^f_{e_i} \pm \gamma$, where $\delta^f_{e_i}$ is the measured value of the elevator at time t_{f_i} and γ is a positive constant that should be tuned sufficiently small to preserve the performance of the controller, but, at the same time, large enough to allow the size of residual

Prepared using rncauth.cls
DOI: 10.1002/rnc

signal exceed the predefined threshold J_i^{th} for a stuck elevator. Note that the positive or negative ($\pm$) sign depends on the bound that the MPC modifies, according to the description in Figure 3. The MPC maintains this new γ-tightened bound for τ samples. On one hand, τ must be selected sufficiently large to ensure that the control commands u have time to adjust to the updated (in terms of feasible region) parameters θ. On the other hand, τ must be small enough to preserve performance (especially in case of false alarms or stuck faults). It is reasonable to set τ proportional to the prediction horizon N.

4.3. Diagnosis of the root cause

If $J_i(t_{\text{f}_i} + \tau) < J_i^{\text{th}}$ at the end of the diagnosis period, the FD module confirms a *stall load* as the root cause of the jamming fault, because the controller showed (using the reconfiguration for diagnosis) that jammed elevator can still move within its reduced bounds. If $J_i(t_{\text{f}_i} + \tau) \geq J_i^{\text{th}}$, the FD module confirms a *stuck elevator* as the root cause of the jamming fault, because the faulty elevator was unable to reach the tightened bound.

4.4. Reconfiguration for stuck fault

As soon as the FD module communicates the root cause of the jamming fault, the MPC controller performs the second reconfiguration. If the diagnosis is that the elevator is stuck, the MPC controller performs the reconfiguration for the stuck elevator by setting both $\underline{\delta}_{\text{e}_i}$ and $\overline{\delta}_{\text{e}_i}$ in the definition of g_θ to $\delta_{\text{e}_i}^f$, as Figure 3 shows. In this way, the artificial reference is generated to take into account that the i-th elevator is permanently stuck at the fault position and adapts the reference for the remaining healthy elevators accordingly. This second reconfiguration is also the last one for the stuck elevator.

4.5. Reconfiguration for stall-load start

If the diagnosis is stall load on the i-th elevator, the MPC controller performs the reconfiguration for stall-load start to allow the detection of the end of the stall load. In this respect, the controller sets the previously modified bound ($\overline{\delta}_{\text{e}_i}$ or $\underline{\delta}_{\text{e}_i}$ depending on the sign of $e_{\delta_{\text{e}_i}}$ at time t_{f_i}) to the new value $\delta_{\text{e}_i}^f \pm \alpha$, that is, the controller allows a $\alpha > 0$ larger feasible region for the i-th elevator, but does

Prepared using rncauth.cls
DOI: 10.1002/rnc

not restore yet the original bound ($\overline{\delta}^{\mathrm{o}}_{\mathrm{e}_i}$ or $\underline{\delta}^{\mathrm{o}}_{\mathrm{e}_i}$) yet. This new limit allows one to detect the elevators deviate from the temporarily jammed position at the end of the stall load.

Remark 8

Setting $\alpha = 0$ could prevent the FD module to monitor the end of the stall load because the elevator cannot follow a command that exceeds its reduced bound. The reduced bounds of elevators due to miss detecting the end of a stall load may lead to sever control performance degradation.

4.6. Detection of end of stall load

During the reconfiguration for stall-load start, the FD module constantly monitors the discrepancy between the measured elevator position $\delta^{\mathrm{m}}_{\mathrm{e}_i}$ and its previously jammed position $\delta^{f}_{\mathrm{e}_i}$. If $|\delta^{\mathrm{m}}_{\mathrm{e}_i} - \delta^{f}_{\mathrm{e}_i}| \leq \alpha$, the FD module communicates that the stall load is still active on the i-th elevator and the MPC controller maintains its current formulation. When this condition is violated, the FD module communicates the end of the stall load to the controller and returns to monitor the residual value.

4.7. Reconfiguration for stall-load end

When the stall load ends, the MPC must restore the original saturation limit (i.e., $g_\theta = g$), which is the last reconfiguration for the stall load.

Remark 9

The MPC reconfiguration can handle more than one elevator fault at a time, thanks to the decoupled structure of the FD module, which monitors each elevator independently. In this work, however, we consider symmetric faults, that is, if a jamming fault occurs on the left inner elevator, the same fault occurs on the right inner elevator. The reason for this choice is related to the fact that nonsymmetric faults affect the lateral behavior of the aircraft and would require a different (more complex) model to build the MPC predictions.

Remark 10

Compared to [22], all the reconfigurations in the MPC problem formulation does not affect the states and the control commands, but only the feasible region of the parameters θ. These reconfigurations

Prepared using rncauth.cls
DOI: 10.1002/rnc

affect the way the artificial reference is generated and allows a smoother transition from the fault-free region to the faulty feasible region (by generating a feasible reference signal for the states and actuators at every problem instance).

4.8. Discussion

The proposed algorithm relies on the interactions between the FD unit and the MPC controller. In this work, we proposed a simple FD design and an LTI MPC formulation to simplify the presentation of our approach (as pointed out in Remarks 3 and 6).

The success of our proposed algorithm depends on the accuracy of the detection and diagnosis. In general, the fault detection and diagnosis accuracy depends mainly on N_{eval}, J_i^{th} and τ. These parameters determine the delay from fault occurrence to control reconfiguration. On one hand, if we set these parameters so that the delay is short, the FD results are less accurate. Consequently, control performance is sacrificed. On the other hand, if we set those parameters so that the delay is larger, the FD results are more accurate, but the control performance would still be sacrificed (due to the larger delay). This suggests a trade-off in the waiting time for the reconfiguration. Detailed theoretical analysis of such an integration for FD parameter tuning is an open theoretical challenge [48]. Nevertheless, the intuitive understanding above provides a guideline for tuning.

The proposed design is robust to scenarios that might lead to misdetection or misdiagnosis of actuator faults. For example, if the *reconfiguration for diagnosis* is triggered by a misdetection in the FD unit, a temporary reconfiguration of the actuator bounds will be performed leading to τ time instances of conservative behavior. In most cases, the redundancy in the number of actuators (that allows to reallocate the control action on the healthy control surfaces) will mitigate the conservatism due to the misdetection.

A more severe situation that the proposed algorithm does not address is related to the misdiagnosis of a stuck fault. In particular, suppose that τ is too short and the residual signal does not have enough time to decrease during the diagnosis phase. In this scenario, a stuck fault for an healthy elevator is diagnosed by our algorithm. This misdetection can seriously affect the performance, especially

Prepared using rncauth.cls
DOI: 10.1002/rnc

if all the longitudinal control surfaces are erroneously diagnosed as stuck. The algorithm can be modified to include additional control surfaces (e.g., the ones associated to the lateral dynamics) to compensate for the fault, or techniques to recover from the misdiagnosis of a fault must be implemented for this particular scenario.

5. SIMULATION RESULTS

This section presents numerical results of our integrated control strategy on an Airbus simulator that has been the benchmark model of the RECONFIGURE project [25].

The threshold J_i^{th} in the FD module is selected according to the guideline of Section 3 and is equal to 0.40 for the inner elevators and to 0.65 for the outer elevators (the thresholds are different given the differences between the inner and outer elevator models). In addition, we implemented the detection strategy that exploits redundancy described in Section 3.3. In this respect, the FD unit detects a fault on the i-th elevator if $J_i >= 4J_j$, $i \neq j$, $i, j \in \mathcal{I}$, that is, when the residual signal of the i-th elevator is four times larger than the residual signals of the other elevators. In addition, we selected the time required for the diagnosis of the root cause of the jamming as $\tau = N\,T_s$ ($N = 20$ is the length of the prediction horizon and $T_s := 0.04$ sec is the sampling time of the system), that is,τ is selected proportional to the prediction horizon used in the MPC problem formulation. Another parameter that requires a trade-off between performance and accuracy is γ, used to tighten the faulty-elevator constraints during the *reconfiguration-for-diagnosis* phase. We noticed that a small value of γ (e.g., 1% of the maximum allowed control command) is sufficient for the diagnosis. Finally, we selected α sufficiently large (e.g., 3γ) to avoid false alarms in the detection of the stall-load end right after the diagnosis phase.

We trimmed the aircraft at an altitude of $12,500$ feet and calibrated airspeed of 335 knots (inside the flight envelope) and we used the linearized model of the aircraft at the trimmed operating condition to build the MPC prediction model. Our aim is to track a doublet signal on the vertical load factor, that is, $\nu := n_{z_{\text{ref}}}$. Specifically, we consider the sequence of two doublets of different

Prepared using rncauth.cls
DOI: 10.1002/rnc

amplitude. The first doublet starts at 0.04 sec and ends at 20.04 sec and its value exceeds the allowed constraints on the vertical load factor. The second doublet starts at 30.04 sec and ends at 50.04 sec and its value remains within the constraints of the vertical load factor. We study the performance of our integrated design in the following scenarios:

- Stall load occurring at 2.65 sec from the beginning of the simulation on the inner elevators.
- Stuck fault occurring on the inner elevators at 2.65 sec from the beginning of the simulation.

The baseline to evaluate the performance of the proposed integrated design is the behavior of the system controlled by the MPC, in the fault-free case. Note that we simulate the occurrence of the faults during the first doublet when the reference signal starts exceeding the vertical load factor bounds. Furthermore, in the following, recall that all the reconfigurations operate on the feasible region of the artificial reference signal (as discussed in Section 4) and do not affect the original feasible region of the states and actuators.

5.1. Stall Load

Figures 4-5 present the results obtained using the proposed algorithm (i.e., the integrated FD-MPC design) in case of stall load on the inner elevators. In this scenario the outer elevators are healthy.

Figure 4 details the behavior of the vertical load factor. During the first part of the manoeuvre the stall load occurs. The proposed algorithm allows the controller to avoid the constraint violation of the vertical load factor (that would have occurred without a tailored control reconfiguration) with a minor loss of performance compared to the fault-free case (dot-dashed green line). Figure 5[†] details the behavior of the elevators and of the residual signals during the detection and diagnosis of the stall load. Once the fault is detected, the MPC controller immediately updates the lower bound of the faulty inner elevators. Consequently, the outer elevators (second row) compensate for the temporary loss of the inner elevators (first row) leading to an overall control action (third row) that is comparable to the one in the fault-free case.

[†]The units on the vertical axis of the elevator plots (Figures 5, 7) have been removed upon request of our industrial partners.

Prepared using rncauth.cls
DOI: 10.1002/rnc

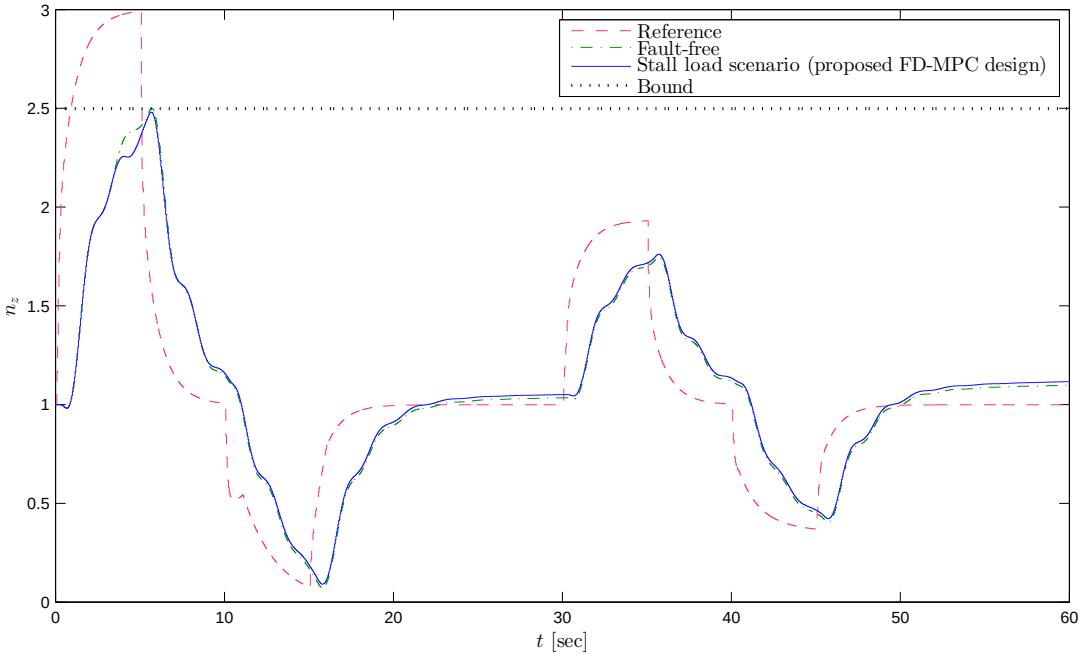


Figure 4. Comparison of the vertical load factor tracking performance in the fault-free case (dot-dashed green line) and when a stall load on the inner elevators (at 2.65 sec from the beginning of the simulation) is detected and diagnosed using the proposed integrated design (solid blue line).

The detection and diagnosis of the fault is fundamental for the performance of the controller. In particular, as shown in the last row of Figure 5, the FD unit alerts the MPC controller as soon as the residual signal J_i of the inner elevators starts to abnormally increase with respect to the one of the outer elevators. When the anomaly is detected the MPC proceeds to perform the reconfiguration for fault diagnosis (first row) and adapt the reference signal to maintain feasibility. At the end of the detection time, given that the residual signal is below the threshold, the FD unit notifies the MPC controller of the occurrence of a stall load. Note that, at the end of the detection phase, the inner elevators are no longer in stall, but they remain close (within α) to the lower bound. Hence, the FD unit maintains the stall load on (highlighted in grey in Figure 5). As soon as the inner elevators move away from their reduced saturation bounds the stall load ends and the MPC controller restores the original elevator bounds.

Prepared using rncauth.cls
DOI: 10.1002/rnc

5.2. *Stuck Fault*

Figures 6-7 present the results obtained using the proposed algorithm in case of permanent jamming of the inner elevators. In this scenario the outer elevators are healthy.

Figure 6 details the behavior of the vertical load factor. During the first part of the manoeuvre the inner elevators become jammed. The proposed algorithm, thanks to the detection and diagnoses of the root cause of the jamming fault, allows the controller to avoid the constraint violation of the vertical load factor with a minor loss of performance. Note that without the proposed sequence of reconfigurations for detection and diagnosis, due to the severity of the fault, the MPC controller would not be able to maintain the system within its feasible region and ensure stability.

Figure 7 presents the behavior of the elevators and of the residual signals during the detection and diagnosis of the stuck fault. Once the fault is detected, the MPC controller immediately performs the first reconfiguration (as done in the previous case for the temporary jamming) to update the bounds associated with the inner elevators in the feasible region of the artificial reference. During the detection phase, compared to the previous scenario, the residual signal of the inner elevators (solid blue line on the last row of Figure 7) increases. At the end of the detection time the residual signal associated with the inner elevators is still above the predefined threshold and the FD module can diagnose the permanent jamming of the inner elevators. After the diagnosis, the MPC controller performs the reconfiguration for stuck fault by updating the upper and lower bound of the faulty elevators in the MPC problem formulation (as also the first row of Figure 7 depicts). The tracking performance is maintained (compared to the fault-free case depicted in dash-dotted green lines) with limited loss thanks to the reallocation of the control authority on the healthy outer elevators (second and third row of Figure 7). The minor performance loss is due mainly to the inner elevator being stuck to a nonzero value and the presence of physical rate limitations in the actuators that affect the response of the outer elevators to the loss of the inner ones.

Prepared using rncauth.cls
DOI: 10.1002/rnc

6. CONCLUSIONS

We presented a novel fault-tolerant controller tailored to aerospace applications. Our approach relies on the close interaction between a fault-detection (FD) module and a model predictive controller (MPC). The FD module exploits the controller to diagnose the root cause of the elevator jamming and the MPC exploits the information provided by the FD module to better handle the jamming. We showed on an Airbus passenger aircraft simulator the benefits that our strategy can bring to the performance of the control system.

As the numerical example showed, the proposed design is effective for the detection and active diagnosis of jamming faults that can occur on the aircraft actuators. Furthermore, the reconfiguration and fault-tolerant reference generation allows one to preserve the tracking performance after the occurrence of the fault.

A limitation of the current approach is related to the definition of the threshold used to activate the diagnosis. Exploiting the information provided by the other actuators helps the early detection of the faults, but if all the control surfaces are affected by a fault (e.g., in case of temporary jamming) the choice of the threshold remains critical. As part of our future work, we plan to investigate different strategies on the threshold selection (for example, by exploring the relationship with the amplitude of the reference signal and disturbances) to improve the detection of the fault.

REFERENCES

1. Zhang Y, Jiang J. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control* 2008; **32**(2):229 – 252. DOI:10.1016/j.arcontrol.2008.03.008.
2. Cieslak J, Henry D, Zolghadri A, Goupil P. Development of an active fault-tolerant flight control strategy. *Journal of guidance, control, and dynamics* 2008; **31**(1):135–147. DOI:10.2514/1.30551.
3. Edwards C, Smaili H, Lombaerts T. *Fault Tolerant Flight Control: A Benchmark Challenge*. Springer-Verlag, 2010.
4. Hartley E, Maciejowski J. A longitudinal flight control law based on robust MPC and H2 methods to accommodate sensor loss in the RECONFIGURE benchmark. *9th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS)* 2015; **48**(21):1000–1005. DOI:10.1016/j.ifacol.2015.09.657.

Prepared using rncauth.cls
DOI: 10.1002/rnc

5. Rosa P, Vasconcelos J, Kerr M. A mixed-μ approach to the integrated design of an FDI/FTC system applied to a high-fidelity industrial airbus nonlinear simulator. *9th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS)* 2015; **48**(21):988–993. DOI:10.1016/j.ifacol.2015.09.655.
6. Ossmann D. Fault tolerant control design for the longitudinal aircraft dynamics using quantitative feedback theory. *AIAA Guidance, Navigation, and Control Conference*, 2015; 1–16. DOI:10.2514/6.2015-1310.
7. Péni T, Vanek B, Szabò Z, Bokor J. Supervisory fault tolerant control of the GTM UAV using LPV methods. *International Journal of Applied Mathematics and Computer Science* 2015; **25**(1):117–131. DOI:10.1515/amcs-2015-0009.
8. Yu X, Liu Z, Zhang Y. Fault-tolerant flight control design with finite-time adaptation under actuator stuck failures. *IEEE Transactions on Control Systems Technology* 2017; **25**(4):1431–1440. DOI:10.1109/TCST.2016.2603072.
9. Zhang Y, Jiang J. Fault tolerant control system design with explicit consideration of performance degradation. *IEEE Transactions on Aerospace and Electronic Systems* 2003; **39**(3):838–848. DOI:10.1109/TAES.2003.1238740.
10. Maciejowski JM, Jones CN. MPC fault-tolerant flight control case study: Flight 1862. *Proc. of IFAC Safeprocess Sympoisum*, 2003; 119–124.
11. Chandra KPB, Chen L, Alwi H, Edwards C. Actuator faults and blow-down limit detection, and fault tolerant control for the RECONFIGURE benchmark problem. *2016 IEEE Conference on Control Applications (CCA)*, 2016; 1544–1549. DOI:10.1109/CCA.2016.7588020.
12. Almeida FAD, Leißling. Fault-tolerant model predictive control with flight-test results. *Journal of Guidance, Control, and Dynamics* 2010; **33**:363–375.
13. Kale MM, Chipperfield AJ. Stabilized MPC formulations for robust reconfigurable flight control. *Control Engineering Practice* 2005; **13**:771–788.
14. Lew J. Robust predictive control for structures under damage condition. *Journal of Guidance, Control, and Dynamics* 2013; **36**:1824–1829.
15. Maciejowski JM. The implicit daisy-chaining property of constrained predictive control. *Applied Mathematics and Computer Science* 1998; **8**:695–712.
16. Stoican F, Olaru S. *Set-theoretic Fault-tolerant Control in Multisensor Systems*. John Wiley & Sons, Inc., 2013.
17. Yetendje A, Seron MM, De Doná JA. Robust multiactuator fault-tolerant MPC design for constrained systems. *International Journal of Robust and Nonlinear Control* 2013; **23**:1828–1845.
18. Goupil P, Boada-Bauxell J, Marcos A, Cortet E, Kerr M, Costa H. AIRBUS efforts towards advanced real-time fault diagnosis and fault tolerant control. *Proc. of the 19th IFAC World Congress*, 2014; 3471–3476.
19. Puncochar I, Siroky J, Simandl M. Constrained active fault detection and control. *IEEE Transactions on Automatic Control* 2015; **60**:253–258.
20. Raimondo DM, Marseglia GR, Braatz RD, Scott JK. Fault-tolerant model predictive control with active fault isolation. *Proc. of 2013 Conference on Control and Fault-Tolerant Systems*, Nice, France, 2013.
21. Xu F, Olaru S, Puig V, Ocampo-Martnez C, Niculescu S. Sensor-fault tolerance using robust MPC with set-based state estimation and active fault isolation. *Proc. of 53rd Conference on Decision and Control*, Los Angeles, CA,

Prepared using rncauth.cls
DOI: 10.1002/rnc

2014; 4953–4958.

22. Ferranti L, Wan Y, Keviczky T. Predictive flight control with active diagnosis and reconfiguration for actuator jamming. *Proc. of 5th IFAC Conference on Nonlinear Model Predictive Control*, 2015; 166–171. DOI:10.1016/j.ifacol.2015.11.278.
23. Limón D, Alvarado I, Alamo T, Camacho EF. MPC for tracking of piece-wise constant references for constrained linear systems. *Automatica* 2008; **44**:2382–2387. DOI:10.1016/j.automatica.2008.01.023.
24. Ferramosca A, Limón D, Alvarado I, Alamo T, Camacho EF. MPC for tracking with optimal closed-loop performance. *Automatica* 2009; **45**(8):1975–1978. DOI:10.1016/j.automatica.2009.04.007.
25. Goupil P, Boada-Bauxell J, Marcos A, Rosa P, Kerr M, Dalbies L. An overview of the FP7 RECONFIGURE project: industrial, scientific and technological objectives. *Proc. of the 9th IFAC Symposium on SAFEPROCESS*, 2015; 976–981.
26. Luenberger D. Observers for multivariable systems. *IEEE Transactions on Automatic Control* 1966; **11**(2):190–197. DOI:10.1109/TAC.1966.1098323.
27. Bemporad A, Morari M, Dua V, Pistikopoulos EN. The explicit linear quadratic regulator for constrained systems. *Automatica* 2002; **38**(1):3–20. DOI:10.1016/S0005-1098(01)00174-1.
28. Ferreau HJ, Bock HG, Diehl M. An online active set strategy to overcome the limitations of explicit MPC. *International Journal of Robust and Nonlinear Control* 2008; **18**(8):816–830. DOI:10.1002/rnc.1251.
29. Rao CV, Wright SJ, Rawlings JB. Application of Interior-Point Methods to Model Predictive Control. *Journal of optimization theory and applications* 1998; **99**(3):723–757. DOI:10.1023/A:1021711402723.
30. Patrinos P, Bemporad A. An accelerated dual gradient-projection algorithm for embedded linear model predictive control. *IEEE Transactions on Automatic Control* 2014; **59**(1):18–33. DOI:10.1109/TAC.2013.2275667.
31. Zeilinger MN, Colin NJ, Raimondo DM, Morari M. Real-time MPC-Stability through Robust MPC design. *Proc. of the 48th IEEE Conference on Decision and Control held jointly with the 28th Chinese Control Conference*, 2009; 3980–3986. DOI:10.1109/CDC.2009.5400903.
32. Mayne D, Rawlings J, Rao C, Scokaert P. Constrained model predictive control: Stability and optimality. *Automatica* 2000; **36**(6):789–814. DOI:10.1016/S0005-1098(99)00214-9.
33. Maciejowski JM. *Predictive control: with constraints*. Pearson education, 2002.
34. Borrelli F, Bemporad A, Morari M. *Predictive Control for linear and hybrid systems*. 2015. Available at http://www.mpc.berkeley.edu/mpc-course-material.
35. Parikh N, Boyd S. Proximal algorithms. *Foundations and Trends® in Optimization* 2014; **1**(3):127–239. DOI:10.1561/2400000003.
36. Stathopoulos G, Shukla H, Szucs A, Pu Y, Jones CN. Operator splitting methods in control. *Foundations and Trends® in Systems and Control* 2016; **3**(3):249–362.
37. Ferranti L, Keviczky T. Operator-splitting and gradient methods for real-time predictive flight control design. *Journal of Guidance, Control, and Dynamics* 2016; DOI:10.2514/1.G000288.

38. Marcos A, Balas GJ. Development of linear-parameter-varying models for aircraft. *Journal of Guidance, Control, and Dynamics* 2004; **27**(2):218–228.

39. Scherer C. LPV control and full block multipliers. *Automatica* 2001; **37**(3):361–375. DOI:10.1016/S0005-1098(00)00176-X.

40. Falcone P, Tufo M, Borrelli F, Asgari J, Tseng HE. A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems. *Proc. of the 46th IEEE Conference on Decision and Control*, IEEE, 2007; 2980–2985.

41. Prodan I, Olaru S, Bencatel R, de Sousa JB, Stoica C, Niculescu SI. Receding horizon flight control for trajectory tracking of autonomous aerial vehicles. *Control Engineering Practice* 2013; **21**(10):1334–1349. DOI:10.1016/j.conengprac.2013.05.010.

42. Ferranti L, Keviczky T. MPC design for the longitudinal motion of a passenger aircraft based on operator-splitting and fast-gradient methods. *Proc. of the European Control Conference (ECC'16)*, 2016; 1562–1567.

43. Langson W, Chryssochoos I, Raković S, Mayne DQ. Robust model predictive control using tubes. *Automatica* 2004; **40**(1):125–133.

44. Scokaert POM, Rawlings JB. Constrained linear quadratic regulation. *IEEE Transactions on Automatic Control* 1998; **43**(8):1163–1169.

45. Stathopoulos G, Korda M, Jones CN. Solving the infinite-horizon constrained lqr problem using accelerated dual proximal methods. *IEEE Transactions on Automatic Control* 2017; **62**(4):1752–1767.

46. Ferranti L, Stathopoulos G, Jones CN, Keviczky T. Constrained lqr using online decomposition techniques. *Proc. of the 55th IEEE Conference on Decision and Control (CDC)*, IEEE, 2016; 2339–2344.

47. Richards A, How J. Robust stable model predictive control with constraint tightening. *Proc. of the American Control Conference*, IEEE, 2006; 6–pp.

48. Zhang Y, Jiang J. Issues on integration of fault diagnosis and reconfigurable control in active fault-tolerant control systems. *IFAC Proceedings Volumes* 2006; **39**(13):1437–1448.

Prepared using rncauth.cls
DOI: 10.1002/rnc

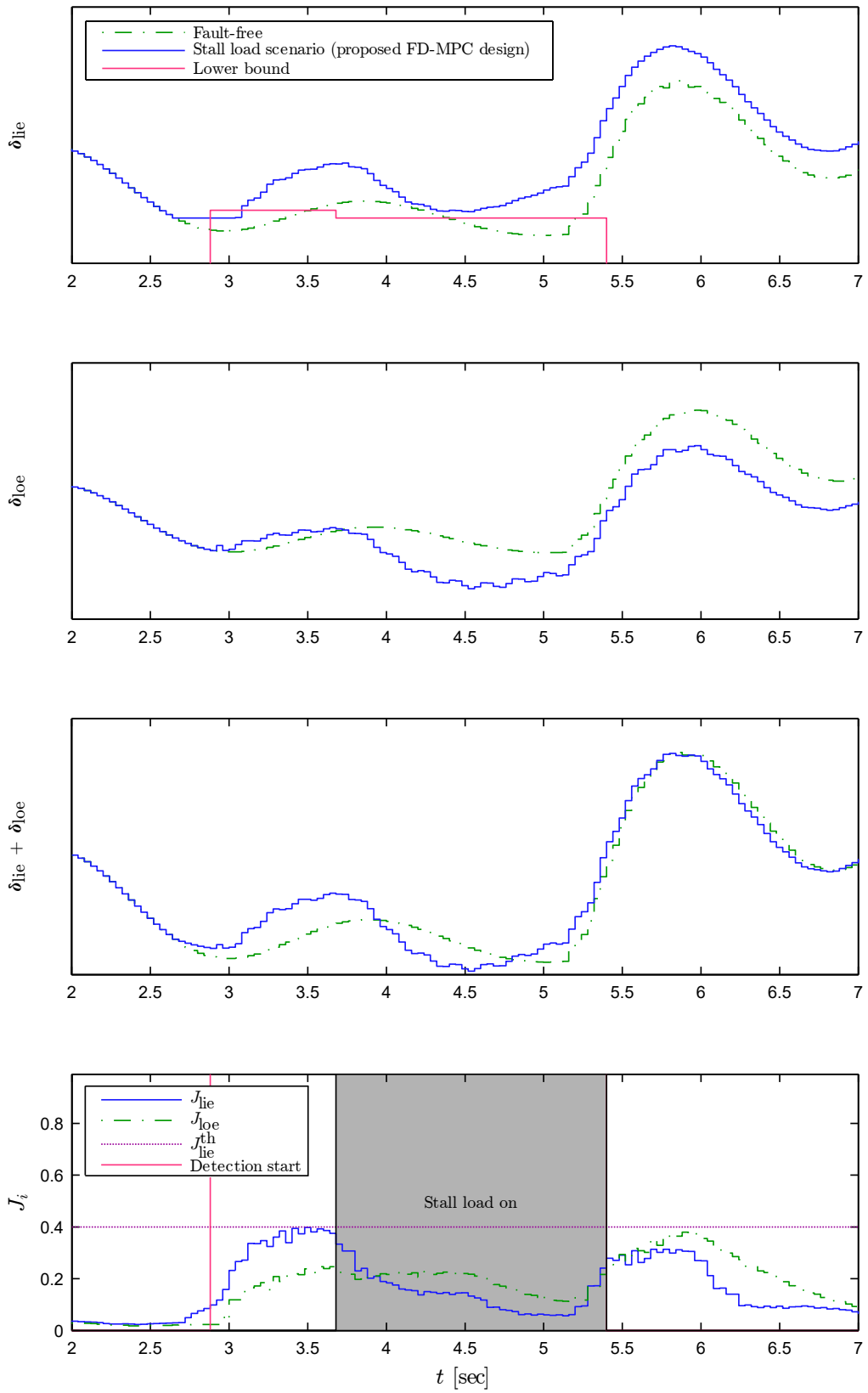


Figure 5. Comparison of the elevator behaviors (rows 1–3) in the fault-free case (dot-dashed green line) and when a stall load on the inner elevators is detected and diagnosed using the proposed integrated design (solid blue line). The last row depicts the behavior of the residual signals used to detect and diagnose the fault. The grey area highlights the duration of the reconfiguration for stall load start.

DOI: 10.1002/rnc

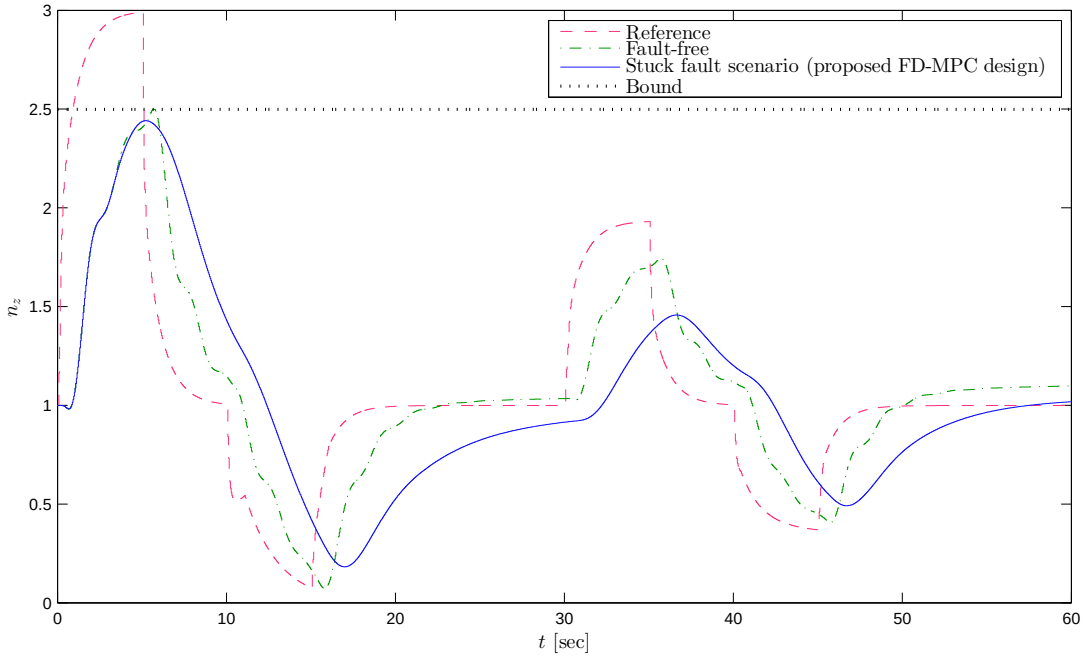


Figure 6. Comparison of the vertical load factor tracking performance in the fault-free case (dot-dashed green line) and when a permanent jamming of the inner elevators (at 2.65 sec from the beginning of the simulation) is detected and diagnosed using the proposed integrated design (solid blue line).

Prepared using rncauth.cls
DOI: 10.1002/rnc

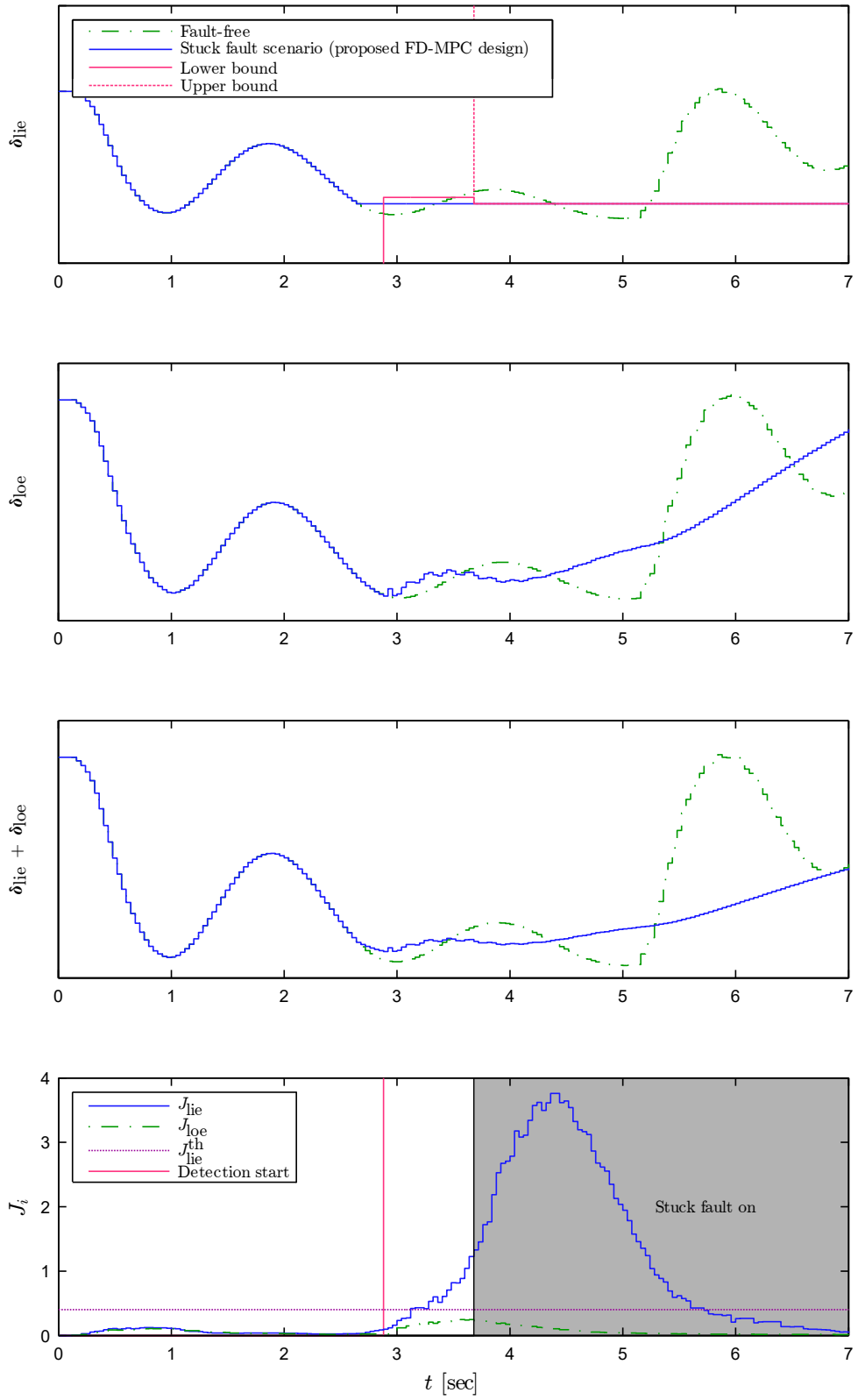


Figure 7. Comparison of the elevator behaviors (rows 1–3) in the fault-free case (dot-dashed green line) and when a stuck fault on the inner elevators is detected and diagnosed using the proposed integrated design (solid blue line). The last row depicts the behavior of the residual signals used to detect and diagnose the fault. The grey area highlights the reconfiguration for stuck fault.

DOI: 10.1002/rnc