

A longitudinal flight control law to accommodate sensor loss in the RECONFIGURE benchmark

E. N. Hartley, K. Siaulys, J. M. Maciejowski*

*University of Cambridge Department of Engineering, Trumpington Street, Cambridge.
CB2 1PZ. United Kingdom.*

Abstract

The feedback gains in state-of-the-art flight control laws for commercial aircraft are scheduled as a function of values such as airspeed, mass, and centre of gravity (CoG). If measurements or estimates of these are lost due to multiple simultaneous sensor failures, the pilot must revert to an alternative control law, or, in the ultimate case, directly command control surface positions. This work develops a robust backup load-factor tracking control law, that does not depend on these parameters, based on application of theory from robust MPC and \mathcal{H}_2 optimal control. Firstly, the methods are applied with loss only of airdata, and subsequently also with loss of mass and CoG estimates. Local linear analysis indicates satisfactory performance over a wide range of operating points. To keep the aircraft within an acceptable operating region, an outer protection loop is implemented using an override approach, based on ground speed, a model of the trim angle of attack and variation of load factor with respect to angle of attack, and *a priori* bounds on the wind speed. Finally, the resulting control laws are demonstrated on the nonlinear RECONFIGURE benchmark, which is derived from an Airbus high fidelity, industrially-validated simulator.

Keywords: Optimal control, Robust control, Aircraft control, Aerospace, Fault-tolerant control, Flight envelope protection

Contents

1 Introduction	1	3.2.4 Climb rate limit	10
2 Inner loop load factor control law	2	3.2.5 Speed limit	10
2.1 Control problem	2	3.2.6 Override rules	10
2.2 Theoretical grounding	3	3.3 Implementation aspects	10
2.2.1 Output-feedback transformation	3	4 Performance evaluation on a high fidelity model	11
2.2.2 Robust control theory	4	4.1 Inner loop load factor control law	11
2.2.3 Robustification to gain scheduling	5	4.1.1 Airspeed measurement failure	11
2.2.4 Parameterised dynamic control law	6	4.1.2 Airspeed, mass and CoG estimates missing	12
2.3 Design using linear models	6	4.2 Flight envelope protection	12
2.3.1 Airspeed measurement failure	6	5 Conclusions	13
2.3.2 Airspeed, mass and CoG estimates missing	7	6 Acknowledgements	13
2.4 Implementation aspects	7	1. Introduction	
3 Flight envelope protection	8	Automatic control systems are integral to the operation of modern civil airliners (Favre, 1994), reducing the pilot's workload through stability augmentation and providing a consistent response to commands through the whole flight envelope. The open-loop response of the aircraft varies considerably, and when a linear feedback control law is used, it is usual to schedule its parameters based on the flight point. For example, when controlling the longitudinal short-period mode, load factor (n_z) and pitch rate (q) are fed-back, but the control law parameters are scheduled as	
3.1 Objectives	8		
3.2 Design	8		
3.2.1 Angle of attack limit	8		
3.2.2 Pitch angle limit	9		
3.2.3 Flight path angle limit	10		

*Corresponding author

Email addresses: enh20@eng.cam.ac.uk (E. N. Hartley),
ks555@cam.ac.uk (K. Siaulys), jmm@eng.cam.ac.uk (J. M.
Maciejowski)

a function of a combination of airspeed, altitude, Mach number, mass and centre-of-gravity of the aircraft. Each parameter used must therefore be measured or estimated.

Under some specific circumstances, erroneous flight parameters can propagate downstream to the flight control law computation, making the aircraft difficult to handle. Hardware redundancy is the typical mitigating measure (Brière et al., 1995; Goupil, 2011). To achieve robustness to sensor failure, multiple sensors can be employed and a “voting” mechanism implemented to detect and compensate for a large class of faults (Goupil, 2011). Analytical redundancy (where multiple signals are combined to reconstruct an estimate of a parameter without explicitly measuring it) can also be exploited, providing additional fault-tolerance without the burden of additional physical hardware. However when too many simultaneous sensor faults occur, the signals must be considered as polluted and ignored.

The key purpose of this paper is not the development of new theory, but to present an approach for addressing the requirements of a challenging, industrially-motivated application. Section 2 presents an extension of an approach explored in Hartley and Maciejowski (2015), for the control of the longitudinal dynamics of a large commercial aircraft in a scenario where airspeed data is lost. Whilst relatively unusual, this scenario can emerge due to multiple simultaneous faults on air data and angle-of-attack sensors (so angle of attack cannot be used to estimate airspeed). In this situation, one recourse is to switch to a direct control law, where elevator deflection is commanded directly. However, the presented approach aims to maintain a load-factor control law with reasonable robustness and handling qualities, so as to limit the inevitable additional workload falling on the pilot. Stable operation is also subsequently demonstrated where estimates of mass and centre of gravity are lost. In contrast to Hartley and Maciejowski (2015), the control synthesis approach also considers stability of the interpolated control laws used between design points. Prior work by Puyou and Ezerzere (2012) achieved robustness to loss of flight parameter measurements through applying non-smooth optimisation to obtain a fixed-complexity controller robust to loss of mass and centre of gravity estimates. The remaining scheduling information was introduced through an inner-loop nonlinear dynamic inversion (NDI) controller due to difficulties of interpolating dynamic systems. Similarly, Varga et al. (2014) proposed a non-scheduled backup C^* control law, tuned using multi-objective optimisation. In contrast, the presently proposed approach does not require an additional inner control loop, yet allows a load factor control law to be maintained.

Section 3 considers flight envelope protection (e.g., Well (2006); Falkena et al. (2011)). Flight envelope protection is becoming a common feature of commercial aircraft, which can automatically override the pilot (or autopilot) command to prevent the aircraft exiting the envelope within which it has been designed to operate. This type of system allows the pilot to reflexively perform full stick de-

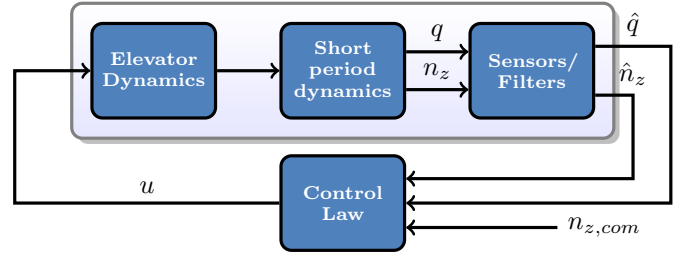


Figure 1: Schematic of inner loop control design setup

flections in response to unanticipated situations without worrying about causing a stall. To protect against over-speed, under-speed, and angle-of-attack limit violations, conventionally, measurements of the airspeed and angle of attack are employed. If the integrity of the airspeed and angle of attack measurements is lost, the existing systems must be disengaged. One of the requirements specified in the RECONFIGURE Benchmark Scenario Description is to “keep the aircraft in a safe region” in the case of detected loss of angle of attack and airspeed measurements. The controller described in Section 2 does not, on its own, fulfil this requirement. A replacement flight envelope protection system is therefore proposed in Section 3, employing ground speed measurements, a model of the aerodynamic behaviour, and *a priori* bounds on the wind speed.

Section 4 presents results obtained from testing the robust inner loop controller and the outer loop flight envelope protection system in an industrial high-fidelity nonlinear simulator provided by Airbus for use in the RECONFIGURE project. Details of the simulator and the wider scope of the project can be found in Goupil et al. (2014, 2015).

2. Inner loop load factor control law

2.1. Control problem

Airbus has provided the RECONFIGURE consortium with linearisations of the longitudinal dynamics of an aircraft in straight-and-level flight at 234 different flight points. These cover an envelope of altitudes, airspeeds, masses and centre-of-gravity (CoG), with 214 of the flight points covering “clean” aerodynamic configurations with slats and flaps fully retracted, and a further 20 covering configurations with the high-lift devices extended to various degrees and landing gear also extended in some configurations. Altitudes range from 5000 ft to 35000 ft, the ratio of maximum to minimum airspeed for a given altitude-mass-CoG combination is up to 1.6, and the highest mass is 2.15 times the lowest. The rigid-body models are accompanied by simplified linear sensor, filter and actuator models. This study considers a setup where all elevators act in common mode, and neglect the trimmable-horizontal-stabiliser (THS), which in any case can only control at much lower frequency ranges than those considered. A sampling period $T_s = 0.04$ s is used.

The objective is to control only the short-period dynamics (leaving the pilot or an outer loop to control the phugoid mode). Classically the short-period dynamics are modelled with the pitch rate q and angle-of-attack α as states, with q , α and vertical “load factor” n_z as outputs. (Strictly, the load factor is the ratio of lift to weight, however in this paper its deviation from trim (i.e., the acceleration normal to the aircraft body divided by acceleration due to gravity) is universally considered). Usually, α is not available at sufficiently high bandwidth to be used for the innermost control loop, so q and n_z are used as feedback variables. The control input is the elevator deflection (multiple elevators operating in common mode). For design purposes, the short-period model at each flight point is augmented with a first-order-plus delay actuator model for the elevator and first-order low-pass linear sensor models on q and n_z , followed by a first-order low-pass filter yielding estimates \hat{q} and \hat{n}_z of the true values (Figure 1). These approximate the higher-order “true” filters, which also include notches to attenuate certain structural modes. There is an implicit assumption that the existing filters are unalterable.

Let x denote the combined state vector of the elevator dynamics, short-period mode and sensors/filters and y denote the measured output $[\hat{q}, \hat{n}_z]^T$. The short-period dynamics vary with the current airspeed, altitude, CoG, and mass. The parameters that determine the flight point are denoted as ϑ . The augmented linearised plant model at a given flight point ϑ sampled at time step k , with period T_s can be described by the parameterised linear difference equations:

$$x(k+1) = A(\vartheta)x(k) + B(\vartheta)u(k) \quad (1a)$$

$$y(k) = C(\vartheta)x(k) + \cancel{D(\vartheta)u(k)}. \quad (1b)$$

Delays in the model result in it being strictly proper, i.e. $D(\vartheta) = 0$.

The specification for the RECONFIGURE project (Goupil et al., 2014, 2015) states that the closed-loop response should have the following time-domain characteristics. First, the response to a step change in commanded n_z should be “substantially finished” within 6 s. The corresponding pitch rate q should not overshoot its steady state value by more than 30%, the load factor should not overshoot its setpoint by more than 10%, and the “control anticipation parameter” (CAP) should be “consistent” throughout the flight envelope. For an ideal second order model of the short period mode, CAP is defined as $\dot{q}(0)/n_z(\infty)$ in response to a step input. The consistency requirement reflects the importance of pitch acceleration as a cue for the pilot during changes in load factor. In addition it is desirable to have a local 60° phase margin and a 6 dB gain margin at the linear design points, although in degraded conditions, it may not be possible to achieve all of these simultaneously on top of the nominal design uncertainty.

2.2. Theoretical grounding

Let $\mathcal{I} \triangleq \{1, \dots, 234\}$ be an index for the 234 design points, and ϑ_i , for $i \in \mathcal{I}$ denote the flight parameters for the

i th flight point. Define subsets of the flight points $\mathcal{J}_j \subseteq \mathcal{I}$, $j = 0, \dots, j_{\max}$, as “flight groups” such that $\mathcal{J}_m \cap \mathcal{J}_n = \emptyset$, $\forall m \neq n$, and let ϑ_{ji} denote the parameters of the flight point indexed by the i th element of \mathcal{J}_j . The design objective can be posed as the finding j_{\max} control laws $\kappa_j(z)$ that each stabilise all flight points in \mathcal{J}_j , with satisfactory tracking performance. It is assumed that parameters vary slowly in comparison to the controlled dynamics and can be locally approximated as time-invariant.

2.2.1. Output-feedback transformation

Since the model state cannot be measured directly, and the sensors have their own dynamics, this is an *output feedback* control problem. One approach to output feedback control design is to use an observer, but for an observer-based control law, a nominal plant model would need to be chosen (the choice of which is not obvious). Moreover, since non-zero setpoints are to be tracked, the usual assumption that observer error converges to zero is invalid. On the other hand, a static, or arbitrary order feedback law might be difficult to tune. The alternative used here is to transform (1) for each flight point into a non-minimal input-output form whose “state” is a finite time history of inputs and outputs. The “state-observer” for this is then simply a set of shift registers. This form was commonly used in early implementations of MPC (Maciejowski, 2002), and has also been recently highlighted in Granado et al. (2005); Ding and Zou (2014) for applying state-feedback control techniques for uncertain systems in an output-feedback setting. This form also results in the controller state having the same physical interpretation at all flight points, and thus interpolating a feedback gain between flight points using the remaining available parameters is a possible strategy. The present application has the complication of multiple measured outputs but the same principles hold. It should be emphasised that the transformation used here for the multiple-output case is not as simple as using MATLAB’s “`ss2tf`” command and extracting the coefficients. The following proposes a systematic method to compute such a form.

Assume that the plant (1) is m -step observable (in the sense that a sequence of input and output measurements at m consecutive time steps is sufficient to reconstruct the state), and that the number of states in (1) is a whole multiple of m . Define the augmented state $\tilde{x}(k) = [\mathbf{u}(k)^T \quad \mathbf{y}(k)^T]^T$ where $\mathbf{u}(k) = [u(k-m) \quad \dots \quad u(k-1)]^T$ and $\mathbf{y}(k) = [y(k-m)^T \quad \dots \quad y(k-1)^T]^T$. Letting dependency on ϑ be implicit (and approximated as time-invariant), define

$$\Phi = \begin{bmatrix} C \\ \vdots \\ CA^{m-2} \\ CA^{m-1} \end{bmatrix}, \quad \Gamma = \begin{bmatrix} D & & & & \\ \dots & \ddots & & & \\ \dots & CB & D & & \\ \dots & CAB & CB & D & \end{bmatrix}$$

and

$$\Psi = [CA^{m-1}B \quad \dots \quad CA^2B \quad CAB \quad CB]$$

where for genericity, we have included D , despite the fact that $D = 0$ for the application in this paper.

Theorem 1. $y(k) = (\Psi - CA^m\Phi^{-1}\Gamma)\mathbf{u}(k) + CA^m\Phi^{-1}\mathbf{y}(k) + Du(k)$.

Proof. At a given flight point, it holds that

$$\mathbf{y}(k) = \Phi x(k-m) + \Gamma\mathbf{u}(k).$$

Since the plant is m -step observable,

$$x(k-m) = \Phi^{-1}(\mathbf{y}(k) - \Gamma\mathbf{u}(k)).$$

From (1),

$$\begin{aligned} y(k) &= CA^m x(k-m) + \Psi\mathbf{u}(k) + Du(k) \\ &= CA^m\Phi^{-1}(\mathbf{y}(k) - \Gamma\mathbf{u}(k)) + \Psi\mathbf{u}(k) + Du(k) \\ &= (\Psi - CA^m\Phi^{-1}\Gamma)\mathbf{u}(k) + CA^m\Phi^{-1}\mathbf{y}(k) + Du(k). \end{aligned}$$

□

Thus, the non-minimal state-space system is:

$$\begin{aligned} \bar{x}(k+1) &= \begin{bmatrix} \frac{0}{0} & \vdots & -I_{(m-1)n_y} & -\frac{0}{0} \\ \frac{0}{0} & \vdots & 0 & -\frac{0}{0} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{0}{\Psi - CA^m\Phi^{-1}\Gamma} & \vdots & -CA^m\Phi^{-1} & \vdots \end{bmatrix} \bar{x}(k) + \begin{bmatrix} 0 \\ I \\ 0 \\ D \end{bmatrix} u(k) \\ y(k) &= \begin{bmatrix} \Psi - CA^m\Phi^{-1}\Gamma & CA^m\Phi^{-1} \end{bmatrix} \bar{x}(k) + Du(k). \end{aligned} \quad (2)$$

This is generalised to systems with a higher observability index and systems where the state dimension is not an integer multiple of the outputs by increasing the length of the time history of past inputs and outputs, m , until Φ is of full row rank, and using the Moore-Penrose Pseudo-inverse, Φ^+ instead of Φ^{-1} . The nonminimal system realisation is denoted by the matrices $\tilde{A}(\vartheta)$, $\tilde{B}(\vartheta)$, $\tilde{C}(\vartheta)$, $\tilde{D}(\vartheta)$.

2.2.2. Robust control theory

The conventional way of designing a control law that is robust to large degrees of parametric uncertainty is to choose a nominal plant model and then characterise uncertainty around it. However, such a choice is not obvious. Instead, the tool applied here is based on robust MPC theory and \mathcal{H}_2 control theory. Kothare et al. (1996) proposed a method based on linear matrix inequalities (LMIs) to simultaneously stabilise a set of linear plants. At each time step, an optimisation is solved to find a controller of the form $u(k) = K(x(k))x(k)$ that minimises an upper bound on a quadratic cost function over an infinite horizon from the current state. Input and output constraints can also be accommodated, albeit conservatively. Cuzzola et al. (2002) suggests a less conservative method based on multiple Lyapunov functions (De Oliveira et al., 1999, 2002). This is prohibitively computationally demanding for the present application. However, Wan and Kothare

(2002) proposes an offline method, based on interpolation of gains computed at judiciously selected states based on set membership conditions. The present work does not consider constraints: the tracking nature of the problem adds complications when both constraints and large amounts of plant/model mismatch are present. For tracking, the classical technique of augmenting the plant with an integral of the error between the load factor to be tracked and the reference is employed. Let $H = [0, 1]$, and $A_i = 1$, $B_i = T_s$ be the state space matrices of the (discrete-time) integrator, then the augmented system is

$$\begin{aligned} \underbrace{\begin{bmatrix} e(k+1) \\ \tilde{x}(k+1) \end{bmatrix}}_{\bar{x}(k+1)} &= \underbrace{\begin{bmatrix} A_i & -B_i H \tilde{C}(\vartheta) \\ 0 & \tilde{A}(\vartheta) \end{bmatrix}}_{\tilde{A}(\vartheta)} \underbrace{\begin{bmatrix} e(k) \\ \tilde{x}(k) \end{bmatrix}}_{\bar{x}(k)} \\ &+ \underbrace{\begin{bmatrix} -B_i H \tilde{D}(\vartheta) \\ \tilde{B}(\vartheta) \end{bmatrix}}_{\tilde{B}(\vartheta)} u(k) + \underbrace{\begin{bmatrix} B_i \\ 0 \end{bmatrix}}_{B_r} r(k). \end{aligned} \quad (3)$$

The objective now reduces to synthesising control gains K_j that stabilise our integrator-augmented short-period-plus-elevator-plus-sensor input-output composite models. With $\bar{x}(k)$ defined in (3), and weighting matrices Q , R , and S , the control synthesis objective is, for each group \mathcal{J}_j , to minimise an upper bound on

$$\min_{K_j} \max_{\vartheta_{j_i} \in \mathcal{J}_j} \sum_{k=0}^{\infty} \bar{x}(k)^T (Q + K_j^T R K_j + S K_j + K_j^T S^T) \bar{x}(k). \quad (4)$$

The cross term, S , in the quadratic cost function is not always considered in the literature, but turns out to be helpful in this application when addressing tuning (see Section 2.3).

Lemma 1. $K^T R K + S K + K^T S^T = (K^T + S R^{-1}) R (K + R^{-1} S^T) - S R^{-1} S^T$

Proof. By expansion. □

Using this, the result of Cuzzola et al. (2002) can be applied for a general quadratic cost function:

Theorem 2. $\sum_{k=0}^{\infty} \bar{x}(k)^T (Q + K_j^T R K_j + S K_j + K_j^T S^T) \bar{x}(k) \leq \gamma_j$ if $Y_j = K_j G_j$ and for all $\vartheta_{j_i} \in \mathcal{J}_j$,

$$\begin{bmatrix} G_j + G_j^T - X_{j_i} & * & * & * \\ (\tilde{A}(\vartheta_{j_i}) + \tilde{B}(\vartheta_{j_i})) Y & X_{j_i} & 0 & 0 \\ (Q - S R^{-1} S^T)^{1/2} G_j & 0 & \gamma_j I & 0 \\ R^{1/2} (Y_j + R^{-1} S^T G_j) & 0 & 0 & \gamma_j I \end{bmatrix} \geq 0 \quad (5a)$$

$$\begin{bmatrix} 1 & \bar{x}(k)^T \\ \bar{x}(k) & X_{j_i} \end{bmatrix} \geq 0. \quad (5b)$$

Corollary 1. The control law $u(k) = Y_j G_j^{-1} \bar{x}(k) = K_j \bar{x}(k)$, obtained by solution of

$$\min_{\gamma, G_j, X_{j_i}, Y_j} \gamma_j \text{ subject to (5)} \quad (6)$$

minimises an upper bound on the cost function (4).

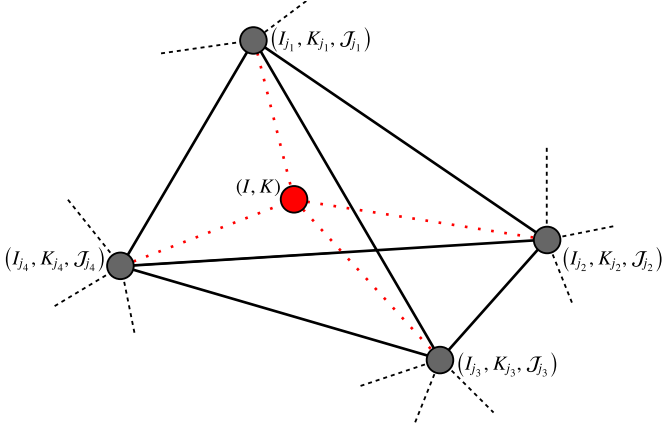


Figure 2: Delaunay Triangulation and barycentric interpolation

Solving (6) online for each new $\bar{x}(k)$ is not appropriate for an aircraft inner control loop where the sampling period is in the order of tens of milliseconds because of the complexity of the required software, the required solution time, and the limited computational hardware. Instead, following the lead of Wan and Kothare (2002) the problem is solved offline for a fixed value of $\bar{x}(k) = \bar{x}_0$ and the resulting control gain used for all $\bar{x}(k)$ for a given flight group. Thus, the final implementation used here is no longer “MPC” in the conventional sense, but can rather be interpreted as a form of \mathcal{H}_2 controller.

2.2.3. Robustification to gain scheduling

A controller designed using Theorem 2 and Corollary 1 for a given “flight group” will stabilise any realisation of the system (3) in that flight group. However, the flight groups only correspond to samples of the envelope, corresponding to discrete values of mass, altitude, and CoG. In reality, these parameters can also take on values between the samples, and a gain-scheduling strategy (Rugh and Shamma, 2000) must be employed. Linear interpolation is an approach that can be used to accommodate this practical issue.

The computation of the control gain K in the control law $u(k) = K\bar{x}(k)$ for a general point in the flight envelope is illustrated in Figure 2. Consider the case when all flight points in the flight group \mathcal{J}_j , $j = 0, \dots, j_{\max}$ have the same mass m_j , altitude z_j and CoG c_j , but differing airspeed. Firstly, Delaunay Triangulation is performed over the set of points $I_j = (m_j, z_j, c_j)$, returning a set of vertices $T_j = \{I_{j_1}, I_{j_2}, I_{j_3}, I_{j_4}\}$, each of which defines a tetrahedral simplex in the flight envelope. Then, given a particular point in the flight envelope $I = (m, z, c)$, the simplex T_j it belongs to is found, and the control gain K for that point is computed via barycentric interpolation of the gains $K_{j_1}, K_{j_2}, K_{j_3}, K_{j_4}$ synthesised for flight groups $\mathcal{J}_{j_1}, \mathcal{J}_{j_2}, \mathcal{J}_{j_3}, \mathcal{J}_{j_4}$ by solving (6).

Strictly, this interpolated control law does not guarantee stability, since the Lyapunov function for the flight group

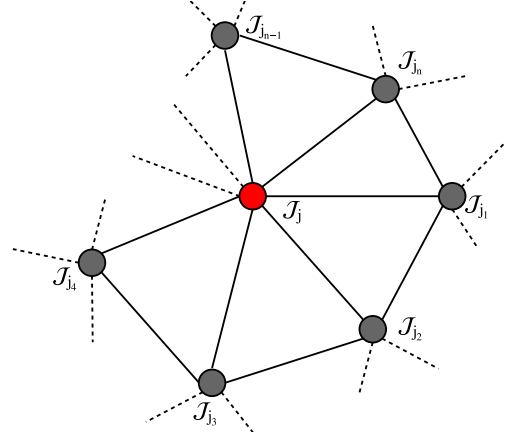


Figure 3: Flight group \mathcal{J}_j with its neighbouring flight groups $N_j = \{\mathcal{J}_{j_1}, \dots, \mathcal{J}_{j_n}\}$ in Delaunay Triangulation

\mathcal{J}_{j_1} , obtained implicitly while synthesizing control gain K_{j_1} , is not necessarily a Lyapunov function for flight groups $\mathcal{J}_{j_2}, \mathcal{J}_{j_3}, \mathcal{J}_{j_4}$, and vice versa. The solution to this is to add additional LMIs to the optimisation problem in (6), guaranteeing that a Lyapunov function for the flight group \mathcal{J}_{j_1} is also a Lyapunov function for all “neighbouring” flight groups $N_{j_1} = \{\mathcal{J}_{j_{11}}, \dots, \mathcal{J}_{j_{1n}}\}$ that are connected by an edge with \mathcal{J}_{j_1} in Delaunay Triangulation.

Proposition 1. *Let \mathcal{J}_j , $j = 0, \dots, j_{\max}$ be one of the flight groups with a set of neighbouring flight groups $N_j = \{\mathcal{J}_{j_1}, \dots, \mathcal{J}_{j_n}\}$, as shown in Figure 3. Also, let the index j_{mk} denote the k -th flight point in the j_m -th neighbouring flight group ($j_m \in \{j_1, \dots, j_n\}$). Then, if the optimisation problem*

$$\begin{aligned} & \min_{\gamma, G_j, X_{j_i}, Y_j, Q_{j_{mk}}} \gamma_j \\ & \text{subject to (5) and} \\ & \begin{bmatrix} G_j + G_j^T - Q_{j_{mk}} & * \\ \bar{A}(\vartheta_{j_{mk}})G_j + \bar{B}(\vartheta_{j_{mk}})Y_j & Q_{j_{mk}} \end{bmatrix} \geq 0 \end{aligned}$$

is feasible, the control law $u(k) = K\bar{x}(k)$ is stabilising for any point in the designed-for flight envelope.

Remark. *The flight groups should be chosen to cover a sufficiently dense sampling of the flight parameters so that linear interpolation provides a good approximation of the behaviour between the design points. No formal claim is made about the stability if the current flight point is outside of the region defined by the convex hull of points within the flight groups used for design, although linear robustness analysis at the design points (Section 2.3) indicates that small excursions that do not dramatically change the linearisation can be tolerated. An approach to maintain the true flight parameters within an acceptable envelope without knowledge of exact airspeed and AoA is considered in Section 3.*

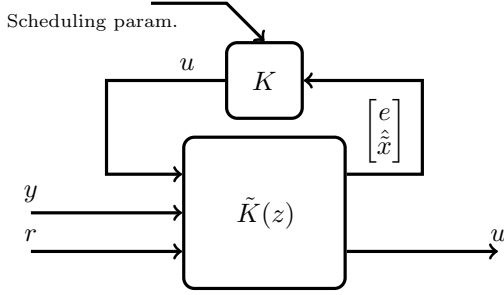


Figure 4: Linear fractional transformation representation of inner loop controller

2.2.4. Parameterised dynamic control law

The inner-loop load factor control law, $\kappa(z)$, that is finally implemented consists of two components: a flight-group dependent gain K , computed as described in Section 2.2.3; and a fixed dynamic linear system with transfer function $\tilde{K}(z)$. Using A_i , B_i and H as defined in Section 2.2.2, and letting A_s , B_{su} , and B_{sy} denote the shift registers formed as in (2), and \hat{x} denote the estimate of $\tilde{x}(k)$ contained therein, the fixed component $\tilde{K}(z)$ can be described by the system:

$$\begin{bmatrix} e(k+1) \\ \hat{x}(k+1) \\ e(k) \\ \hat{x}(k) \\ u(k) \end{bmatrix} = \begin{bmatrix} A_i & 0 & 0 & -B_i H & B_i \\ 0 & A_s & B_{su} & B_{sy} & 0 \\ I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \end{bmatrix} \begin{bmatrix} e(k) \\ \hat{x}(k) \\ u(k) \\ y(k) \\ r(k) \end{bmatrix}.$$

Letting $\mathcal{F}_u(\cdot, \cdot)$ denote the upper linear fractional transformation (e.g., Zhou et al. (1996)), the static component K , and the dynamic system are then combined in feedback, as shown in Figure 4, so that

$$\kappa(z) = \mathcal{F}_u(\tilde{K}(z), K).$$

2.3. Design using linear models

For synthesis, we model the actuator dynamics with a second order discrete time transfer function. The combined pitch rate sensor and filter, and the combined load factor sensor and filter are each modelled as third order transfer functions. These constitute a low-order approximation of the real ones. The short-period dynamic model considers only the states α , and q , a single lumped elevator input and outputs q and n_z .

For tuning, we define a performance output $z(k) = [e(k), q(k-1), n_z(k-1), \Delta q(k-1), \Delta n_z(k-1), u(k), \Delta u(k)]^T$, where $\Delta u(k) = u(k) - u(k-1)$, and $z(k) = C_p(\vartheta)\bar{x}(k) + D_p(\vartheta)u(k)$, and the objective is to thus, minimise an upper bound on $\sum_{k=0}^{\infty} z(k)^T Q z(k)$ over all $\vartheta_{ji} \in \mathcal{J}_j$, where we define Q with tuning weights on the diagonal as

$$Q \triangleq \text{diag} \{1, Q_q, Q_{n_z}, Q_{\Delta q}, Q_{\Delta n_z}, R, R_{\Delta}\}.$$

Thus the weights in (4) become $Q = C_p^T Q C_p$, $R = D_p^T Q D_p$, $S = C_p^T Q D_p$. As a rule of thumb, increasing Q_q can damp q overshoot and increase phase margin, but increase rise

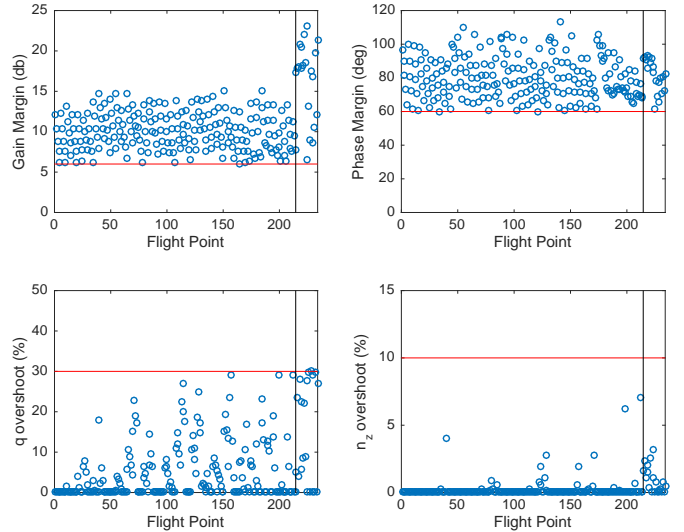


Figure 5: Satisfaction of linear stability margin and step response overshoot requirements

time. Increasing Q_{n_z} increases rise time. Increasing $Q_{\Delta q}$ increases phase margin, but can also increase rise time. It is more effective than Q_q in this respect but does not reduce q overshoot so much. Increasing R slows the response, increases high-frequency roll off and increases gain margin, but can reduce phase margin and increase overshoot. $Q_{\Delta n_z}$ and R_{Δ} have less pronounced effects.

We choose \bar{x}_0 as a vector of zeros with unity on the state corresponding to the previous input (this appeared most reliable at the challenging high altitude, high mass, rear CoG flight points).

In Hartley and Maciejowski (2015), an optimisation-based approach was employed to obtain the tuning weights automatically. In the present setting (including the synthesis refinements of Section 2.2.3), the controller has been tuned manually. For each flight group the initial value $Q_{n_z} = 1.1$ is chosen to get an n_z settling time of the desired order, and $R = 0.008$ to attenuate high frequency input commands. The remaining values are set to 10^{-6} so as to be positive but negligible and the values of R , Q_q , $Q_{\Delta q}$ are then manually increased to address design requirements on pitch rate overshoot and phase margin based on the aforementioned rules of thumb.

The optimisation (6) is formulated using YALMIP (Löfberg, 2004) and solved using MOSEK. In its natural coordinates, the problem can become very ill conditioned. To remedy this, the chosen \bar{x}_0 can be scaled, or the full problem can be scaled by similarity transformation on $\bar{A}(\vartheta)$ and equivalent scaling on all other matrices.

2.3.1. Airspeed measurement failure

The satisfaction of step response overshoot and linear stability margin requirements described in Section 2.1 is illustrated in Figure 5, where flight points to the left of vertical lines (1–214) correspond to “clean” aerodynamic

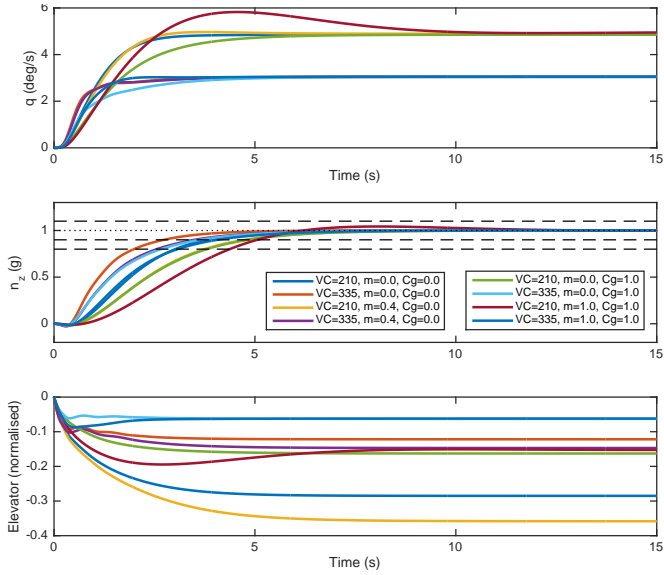


Figure 6: Step response for linear models at 5000 ft (clean configuration)

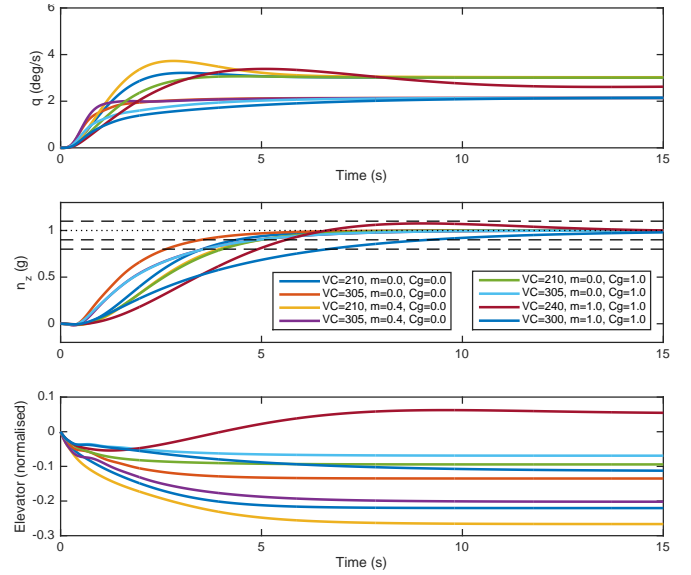


Figure 7: Step response for linear models at 35000 ft (clean configuration)

configurations while flight points to the right (215–234) cover high-lift configurations. Figure 6 shows the closed-loop step responses of the linearised short-period model (which includes models of the filters and elevator actuator) with the control law described in Section 2.2.3, for a clean slat/flap configuration, over the range of airspeeds at vertices of the $[0, 1]$ -normalised mass/CG envelope at an altitude of 5000 ft. The rise-time to 90% (6 s) and overshoot in q (30%) and n_z (10%) are met, and the response, whilst not as consistent as possible with a fully scheduled control law, varies in a predictable manner. Figure 7 shows the closed-loop step responses over the vertices of the airspeed, mass and CoG envelope at 35000 ft. Most flight points meet the required specifications, except at the highest-mass, most rear CoG, where it is very difficult to simultaneously stabilise the upper and lower airspeeds. Here, there are slight violations of rise time requirement.

2.3.2. Airspeed, mass and CoG estimates missing

When mass and CoG estimates are also polluted, a control law can be designed to simultaneously stabilise all vertices of the flight envelope for a given altitude, i.e., each group \mathcal{J}_j contains flight points at a given altitude. Figure 8 shows the step response at 5000 ft (for the clean slat/flap configuration). This has been tuned to achieve at maximum 40% overshoot in q , 10% overshoot in n_z and a linear phase margin of 40° at each of the design points so as to achieve the required n_z response.

2.4. Implementation aspects

Figure 9 shows a high-level functional description of the online control task. The control law must also be defined for small deviations outside the nominal flight envelope. Since the grid describing the flight groups is not completely

regular, firstly, “fictitious” flight groups are used to “pad” the flight envelope to allow it to be bounded by simple box constraints. This is done by repeating the data for the nearest defined flight point before performing the Delaunay Triangulation explained in Section 2.2.3. To clarify this, Figure 10(a) shows the triangulation. Mass and CoG positions are normalised into the range $[0, 1]$ corresponding to the bounds of their admissible values. Green markers indicate the extended padding used to allow online use of simple box saturation. The point-location and barycentric interpolation task is conceptually simple and can be performed using standard MATLAB tools, however, the online task can be further simplified by re-sampling the interpolation offline (red markers) to partition the padded flight envelope into cuboids, as depicted in Figure 10(b). This allows implementation in SIMULINK using (for the case where mass and CoG are available) a pre-lookup block and an array of “Interpolation Using Prelookup” blocks. The scenario where mass and CoG are polluted interpolates the gains with respect to altitude only.

Flight Control Computer (FCC) software coding practices are often quite restrictive in order to facilitate verification of stringent data integrity and real-time requirements, and to fit with current certification processes. Therefore, the controller has been re-implemented using a limited library of SIMULINK-based blocks reflecting those limitations. This library allows the use of only scalar signals and operations and forbids the use of operations such as loops, online optimisation and iterative code. These restrictions allow resource usage to be easily analysed. Complexity of the implementation of our backup control law is such that implementing it within the constraints of this library is a straightforward process.

The designed controller is a backup control law to be

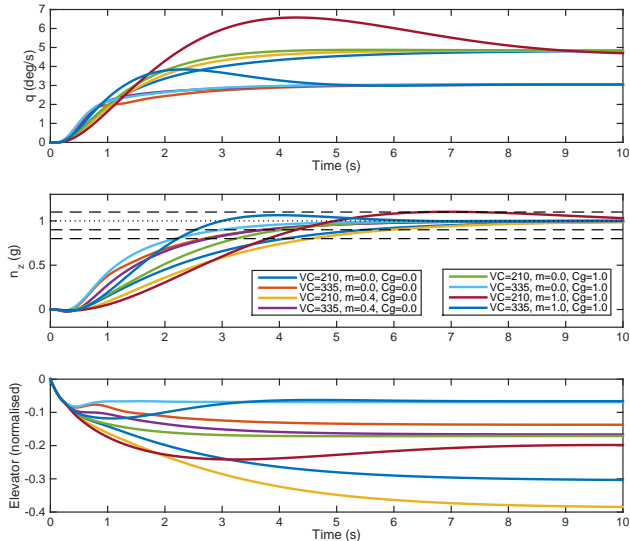


Figure 8: Step response for linear models at 5000 ft: no mass or CoG data used in controller

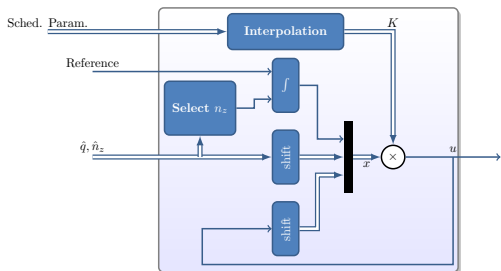


Figure 9: High-level functional description of implementation

used in the event of multiple simultaneous sensor failures. Therefore, a switching strategy from the nominal fully scheduled control law to the robust backup control law is required. This switching strategy achieves continuity of the control input u at the moment of switching by appropriately initialising shift registers and the integrator based on the past values of the nominal control law. Consequently, this leads to reduced transients at the instant of switching.

3. Flight envelope protection

3.1. Objectives

Modern civil airliners have a flight envelope protection facility to prevent them exiting the intended region of operation (Well, 2006; Falkena et al., 2011). For example, flight envelope protection can avoid getting close to a stall due to excessive angle of attack caused by either a large load factor command, or too low an airspeed. However, in a scenario when airspeed and angle of attack are neither reliably measured nor estimated, existing systems that rely upon their values cannot be applied. The control design presented in Section 2 is intended to track pilot load factor commands, whilst being robust to unknown airspeed, but does not itself address keeping the airspeed within the

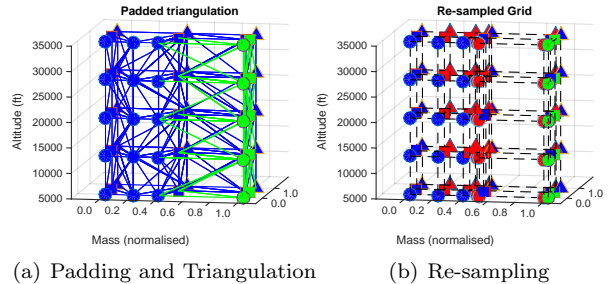


Figure 10: Flight envelope padding, triangulation and re-sampling (normalised-CoG into the page)

designed uncertainty bounds, nor preventing commands that lead to high angles of attack.

To complement the design of Section 2, this section presents a flight envelope protection based on ground speed and a simplified model of the aircraft lift. The RECONFIGURE Benchmark Scenario Description specifies as a requirement that in the case of detected loss of angle of attack and airspeed measurements, the aircraft should be kept in a safe region of operation, and that the design should be operable with up to 60 kts (≈ 30 m/s) head or tail wind and up to 20 kts (≈ 10 m/s) vertical wind. The accommodation of vertical wind means that “angle of attack with respect to ground” (the difference between the pitch angle and the flight path angle) is not a suitable surrogate parameter since the resulting conservatism would be excessive.

3.2. Design

3.2.1. Angle of attack limit

Let W_x denote the head/tail wind velocity, using the convention that a positive wind is blowing in the direction of the aircraft travel, so the true airspeed $V_{TAS} = V_{GND} - W_x$, and let the *a priori* bounds on W_x be denoted as $W_{x,min}$ and $W_{x,max}$, and thus

$$V_{TAS} \in [V_{GND} - W_{x,max}, V_{GND} - W_{x,min}].$$

For aeronautical applications, calibrated airspeed, V_{CAS} , is often the preferred measurement. Let $f_{TAS}^{CAS}(V_{TAS}, z_p)$ be a function that converts from true to calibrated airspeed. Estimates of bounds on V_{CAS} can be computed from the bounds on V_{TAS} using a standard atmospheric model. Thus for altitude z_p :

$$V_{CAS} \in [f_{TAS}^{CAS}(V_{GND} - W_{x,max}, z_p), f_{TAS}^{CAS}(V_{GND} - W_{x,min}, z_p)].$$

For the RECONFIGURE benchmark aircraft with a given mass and slat/flap configuration, the angle of attack at trim, which we will denote as α_0 , can be modelled as a function of calibrated airspeed and mass (CoG has little effect). Letting m be the aircraft mass, based on data obtained from simulations using the nonlinear RECONFIGURE benchmark, and p_i be coefficients obtained by

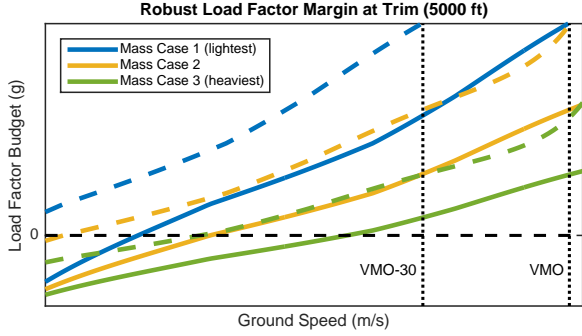


Figure 11: Load factor margin (solid lines: robust to ± 30 m/s wind, dashed lines: zero wind)

least squares regression, for each slat/flap configuration an expression for α_0 is obtained, of the form:

$$\alpha_0(V_{CAS}, m) = \sum_{i=0}^3 p_i \left(\frac{V_{CAS}}{100} \right)^i + m \sum_{i=0}^3 p_{i+4} \left(\frac{V_{CAS}}{100} \right)^i.$$

The scaling factor of 1/100 is to ensure p_i are well scaled. The coefficients of this expression can also be used to form a model for the load factor as a function of α as:

$$L_\alpha(V_{CAS}, m) = \partial n_z / \partial \alpha = (m(\partial \alpha_0 / \partial m))^{-1}.$$

Making the approximation that the effect of pitch rate and the direct effect of elevator deflection on the load factor is small compared to the effect of the angle of attack,

$$n_z \approx L_\alpha(V_{CAS}, m) \times (\alpha - \alpha_0(V_{CAS}, m)).$$

The maximum permitted angle of attack, $\alpha_{\max}(M)$ is a function of the Mach number. Letting V_s be the speed of sound (computed from an atmospheric model as function of z_p). For a given slat/flap configuration, mass and V_{CAS} , the available load factor budget is:

$$n_{z,\max} = L_\alpha(V_{CAS}, m) \times (\alpha_{\max}(V_{TAS}/V_s) - \alpha_0(V_{CAS}, m)).$$

Since the scenario considered does not permit measurements of V_{CAS} or V_{TAS} to be used, a conservative load factor budget is computed based on ground speed and the wind bounds:

$$\hat{n}_{z,\max}^\alpha = \min_{W_x \in [W_{x,\min}, W_{x,\max}]} n_{z,\max} \quad (7)$$

$$\begin{aligned} \text{s.t. } V_{CAS} &= f_{TAS}^{CAS}(V_{GND} - W_x, z_p) \\ M &= (V_{GND} - W_x)/V_s \\ n_{z,\max} &= L_\alpha(V_{CAS}, m)[\alpha_{\max}(M) - \alpha_0(V_{CAS}, m)]. \end{aligned}$$

The values of VMO and VS1G come from a lookup table as a function of mass, CoG, slat/slap configuration and altitude. For $W_x \in [-30, 30]$ m/s, Figure 11 shows how the load factor budget varies as a function of ground speed (solid lines) for 3 different mass cases (ordered low to high), at an altitude of 5000 ft. For comparison, the load factor

budget for the zero wind scenario (where $V_{TAS} = V_{GND}$) is shown as a dashed line, and ground speeds of the maximum operating speed VMO and VMO-30 m/s (to account for wind bounds) are highlighted with vertical dashed lines. Inevitably, using V_{GND} and bounds on W_x reduces the available load factor budget (the solid lines in Figure 11 lie below the dashed ones of the corresponding colour), and increases the minimum acceptable trim speed (the solid lines intersect the horizontal axis to the right of the dashed ones of the corresponding colour). However, there remains some manoeuvrability, and with sufficient ground speed, changing the flight path angle is still feasible.

A negative load factor budget is indicative of at least one realisation of the wind uncertainty causing an under-speed condition. To prevent an overly aggressive descent, in this situation, a lower saturation limit of zero is placed on the load factor budget when the wings are level. Instead, the flight path angle limits are modified to encourage speed gain (Section 3.2.5). Whilst this paper does not address the control of the lateral dynamics, when there is a non-zero bank angle (ϕ), additional lift is required to maintain a given flight path angle. This must be subtracted from the load factor budget, which was computed based on α in level flight. Combining these conditions gives:

$$n_{z,\max}^\alpha = \max\{\hat{n}_{z,\max}^\alpha, 0\} - (1 - \cos \phi).$$

The inner loop control laws are designed to have a settling time of less than 6 s, so, in reality, the aircraft does not respond to a load factor command immediately. At high angles of attack it is reasonable to assume that the airspeed is decreasing. This means that the nominal equilibrium angle of attack will be increasing and the lift derivative decreasing simultaneously, and thus the maximum deliverable load factor will be decreasing. To compensate for this dynamic behaviour it is useful to include some anticipatory behaviour to accommodate the finite time response of the inner load factor tracking loop. The “measurement” of V_{GND} can be adjusted by using the current measurement of horizontal load factor n_x to determine the current acceleration, and projecting a period τ_α into the future assuming constant acceleration.

3.2.2. Pitch angle limit

Let θ be the pitch angle and θ_{\min} and θ_{\max} denote its desired operational limits. Assuming constant airspeed, the pitch rate q_{ss} , for a steady load factor $n_{z,ss}$ satisfies:

$$|q_{ss}| \leq (180/\pi)|n_{z,ss}g/(V_{GND} - W_{x,\max})|.$$

Upper and lower bounds on the load factor to meet these requirements are specified as

$$\begin{aligned} n_{z,\max}^\theta &= \pi(V_{GND} - W_{x,\max})(\theta_{\max} - \theta)/(180g\tau_\theta) \\ n_{z,\min}^\theta &= \pi(V_{GND} - W_{x,\max})(\theta_{\min} - \theta)/(180g\tau_\theta) \end{aligned}$$

where τ_θ is a tuning parameter corresponding to a lower bound on the time to reach θ_{\max} or θ_{\min} from a given θ at a particular time instant ($\theta + q_{ss}\tau_\theta \leq \theta_{\max}$).

3.2.3. Flight path angle limit

The flight path angle γ is the angle between the ground velocity vector projected onto the ground, and its vertical component. Its rate of change is proportional to the load factor: $\dot{\gamma} = (180/\pi)gn_z/V_{\text{GND}}$. Using the same argument as for the pitch angle, one can choose a tuning parameter τ_γ corresponding to a lower bound on the time to reach γ_{max} or γ_{min} and define corresponding limits on the load factor:

$$\begin{aligned} n_{z,\text{max}}^\gamma &= (\pi/180)V_{\text{GND}}(\gamma_{\text{max}} - \gamma)/(g\tau_\gamma) \\ n_{z,\text{min}}^\gamma &= (\pi/180)V_{\text{GND}}(\gamma_{\text{min}} - \gamma)/(g\tau_\gamma). \end{aligned}$$

3.2.4. Climb rate limit

The climb rate should be limited to $[\dot{z}_{\text{min}}, \dot{z}_{\text{max}}]$ by limiting the range of the flight path angle γ . Using small angle approximations:

$$\gamma_{\text{max}}^z = (180/\pi)\dot{z}_{\text{max}}/V_{\text{GND}}, \quad \gamma_{\text{min}}^z = (180/\pi)\dot{z}_{\text{min}}/V_{\text{GND}}.$$

3.2.5. Speed limit

Since in the scenario addressed in this paper airspeed measurements are notably absent, ground speed in combination with *a priori* bounds on wind speed is used as an alternative metric for maintaining the flight envelope. Firstly, it is necessary to define upper and lower bounds V_{min} and V_{max} respectively, on the ground speed V_{GND} . Letting $f_{\text{CAS}}^{\text{TAS}}(V_{\text{CAS}}, z_p)$ be a model-based conversion from calibrated airspeed to true airspeed, bounds are designed as follows. The stall speed and the maximum operating speed are converted from their values specified in terms of V_{CAS} , into values in terms of V_{TAS} for the current altitude. Then these are adjusted based on the bounds on the wind, and finally a positive safety margin δV_{min} is added to the lower bound. We choose a value of $\delta V_{\text{min}} = 8.5$ m/s, but this is a tuning parameter. So:

$$\begin{aligned} V_{\text{min}} &= f_{\text{CAS}}^{\text{TAS}}(VS1G, z_p) + W_{x,\text{max}} + \delta V_{\text{min}} \\ V_{\text{max}} &= f_{\text{CAS}}^{\text{TAS}}(VMO, z_p) + W_{x,\text{min}}. \end{aligned}$$

Assuming that thrust approximately cancels drag, then the main gain or loss of speed is due to gain or loss of altitude, with the following approximate relationship: $\dot{V}_{\text{GND}} \approx -(\pi/180)g\gamma$. Following our previous pattern, we define τ_V as a tuning parameter with the physical interpretation: $V_{\text{min}} \leq V_{\text{GND}} + \dot{V}_{\text{GND}}\tau_V \leq V_{\text{max}}$. This places a constraint on the admissible flight path angle, and can be re-arranged to compute γ_{max}^V and γ_{min}^V . Therefore:

$$\gamma_{\text{min}}^V = \frac{180(V_{\text{GND}} - V_{\text{max}})}{\pi g \tau_V} \quad \gamma_{\text{max}}^V = \frac{180(V_{\text{GND}} - V_{\text{min}})}{\pi g \tau_V}.$$

3.2.6. Override rules

Let $n_{z,\text{ref}}$ be the pilot's load factor command. We can consider enforcing limits on the command passed to the inner loop controller in the following order:

$$1. n_{z,\text{ref}} \leftarrow \max\{n_{z,\text{ref}}, n_{z,\text{min}}^\gamma\}$$

2. $n_{z,\text{ref}} \leftarrow \min\{n_{z,\text{ref}}, n_{z,\text{max}}^\gamma\}$
3. $n_{z,\text{ref}} \leftarrow \max\{n_{z,\text{ref}}, n_{z,\text{min}}^\theta\}$
4. $n_{z,\text{ref}} \leftarrow \min\{n_{z,\text{ref}}, n_{z,\text{max}}^\theta\}$
5. $n_{z,\text{ref}} \leftarrow \min\{n_{z,\text{ref}}, n_{z,\text{max}}^\alpha\}$.

The first and second overrides ensure the flight path angle is within design limits, and the third and fourth ensure that the pitch angle is within design limits. The fifth override ensures that any positive load factor commanded (either by the pilot, or one of the other overrides) does not cause an angle of attack limit violation.

The flight path angle limit is a function of the desired climb rate ranges and the over and under-speed limits. In principle, the limits should be:

$$\gamma_{\text{max}} = \min\{\gamma_{\text{max}}^V, \gamma_{\text{max}}^z\} \quad \gamma_{\text{min}} = \max\{\gamma_{\text{min}}^V, \gamma_{\text{min}}^z\}.$$

Before doing this, it is essential to take care to ensure that $\gamma_{\text{max}} \geq \gamma_{\text{min}}$. If the current point is already significantly underspeed then it is possible for γ_{max}^V to be more negative than γ_{min}^z . An analogous situation holds for overspeed. In such a case, we choose to prioritise the climb rate constraint and override

$$\gamma_{\text{max}}^V \leftarrow \max\{\gamma_{\text{max}}^V, \gamma_{\text{min}}^z\}, \quad \gamma_{\text{min}}^V \leftarrow \min\{\gamma_{\text{min}}^V, \gamma_{\text{max}}^z\}.$$

To avoid an unintended descent or climb as a result of the protection loop competing with the autothrust, if for some reason it is enabled (i.e., if this design were to be applied in a non-fault condition, or engines were used to maintain a ground speed instead of an airspeed), we also override:

$$\gamma_{\text{max}} \leftarrow \max\{\gamma_{\text{max}}, 0\} \quad \gamma_{\text{min}} \leftarrow \min\{\gamma_{\text{min}}, 0\}.$$

3.3. Implementation aspects

Most of the numerical operations in the protection loop are simple. Here we propose some approximations to the two most complex tasks. Since the value of $\alpha_{\text{max}}(\cdot)$ comes from a lookup table, the optimisation (7) can be non-trivial. However an approximate solution can be obtained by evaluating the cost over a grid (for example, $W_{x,\text{min}}$, 0, and $W_{x,\text{max}}$) and taking the minimum value. Also, the exact function for converting from TAS to CAS $f_{\text{TAS}}^{\text{CAS}}(V_{\text{TAS}}, z_p)$ involves the computation of some non-integer roots, which is not supported by the flight code library that is targeted. For online implementation, we approximate the conversion using a polynomial in V_{TAS} and z_p , with terms up to third order in each parameter.

To accommodate the scenario where mass estimation has been invalidated, an upper bound on the aircraft mass can be passed to the protection loop. This will impose the most conservative set of bounds on the flight envelope given the remaining assumptions on the aircraft model and the wind bounds, and consequently a reduced level of performance.

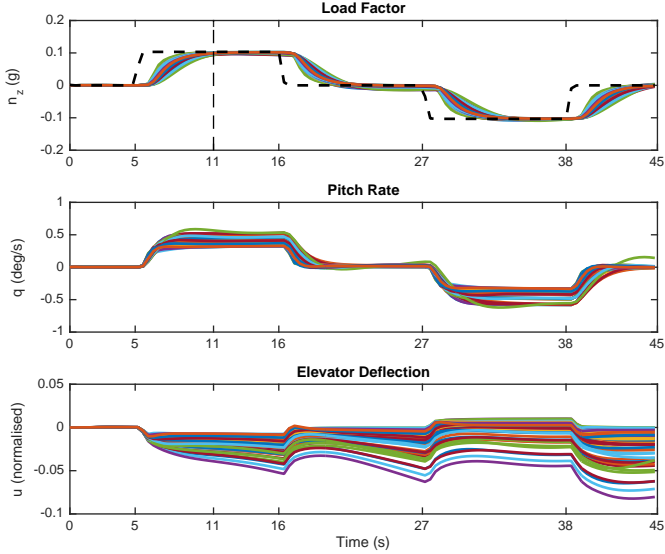


Figure 12: Closed-loop simulation (no airspeed) in nonlinear RECONFIGURE benchmark: all design points for clean configuration at 5000 ft with mass-CoG-altitude scheduled control law

4. Performance evaluation on a high fidelity model

The RECONFIGURE benchmark simulator (Goupil et al., 2014, 2015) is a high-fidelity industrially-validated nonlinear simulator provided to the RECONFIGURE consortium by Airbus. It is comprised of two parts: a compiled “black box” running on a Linux server, which simulates the flight dynamics; and a SIMULINK-based interface containing parts of the flight control computer logic with sufficient hooks to allow replacement of the built-in benchmark control law by a custom design. The two parts communicate over a network link. To demonstrate the behaviour of the proposed designs, the control law is implemented as a “Model Reference Block” in SIMULINK and connected in place of the nominal (fully scheduled) benchmark control law.

4.1. Inner loop load factor control law

Firstly, the inner loop is demonstrated. To prevent distortion of the load factor command, and to allow visual analysis of the consistency of the responses at different flight points, all protection and autothrust laws are disabled.

4.1.1. Airspeed measurement failure

When airspeed measurements are unavailable, but mass and CoG are usable, the design from Section 2.3.1 is demonstrated, interpolated based on mass, CoG and altitude. Figure 12 shows the closed-loop responses to a sequence of pilot sidestick commands corresponding to a small backward deflection at $t = 5$ s followed by a return to neutral at $t = 16$ s, a small forward deflection at $t = 27$ s and neutral at $t = 38$ s, from straight-and-level trim points at an altitude of 5000 ft, for each of the range of mass, CoG and airspeed combinations corresponding to the design points.

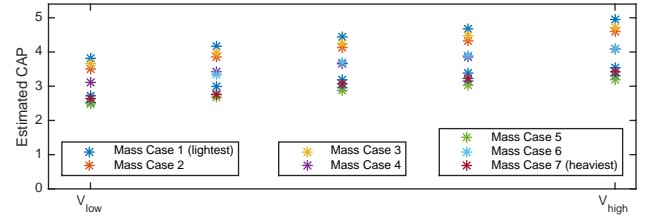


Figure 13: Estimated CAP (no airspeed) in nonlinear RECONFIGURE benchmark for clean configuration at 5000 ft with mass-CoG-altitude scheduled control law

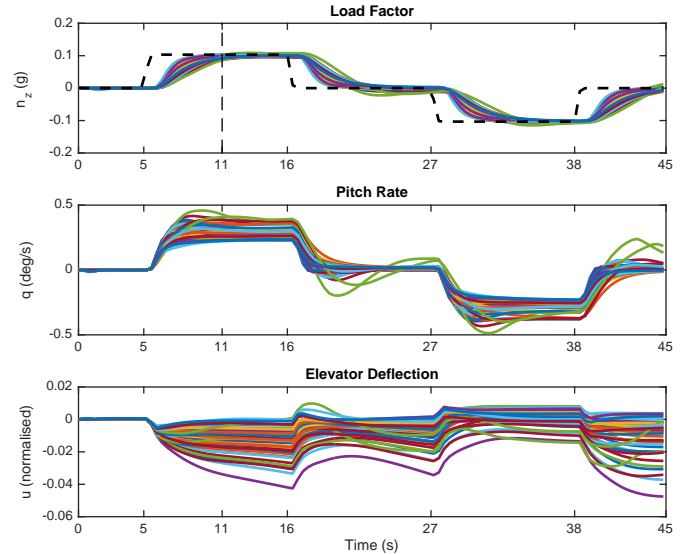


Figure 14: Closed-loop simulation (no airspeed) in nonlinear RECONFIGURE benchmark: all design points for clean configuration at 27500 ft with mass-CoG-altitude scheduled control law

The load factor command is plotted as a black dashed line, and the point 6 s after the first change in setpoint is highlighted to emphasise that the load factor transient is substantially finished within this time. The initial gradients of the pitch rate trajectories following the command are similar, indicating a consistent control anticipation parameter (CAP). An estimate of the CAP, formally for an ideal two state model with a step at $t = 0$ defined as $\dot{q}(0)/n_z(\infty)$, but in this non-ideal case with a highly augmented system, approximated as the maximum pitch acceleration divided by the load factor setpoint during the period from 5 s to 11 s is shown in Figure 13. The variation in CAP is small, and moreover its variation with respect to mass and airspeed is predictable. There is minimal overshoot on the n_z and q trajectories. The drift on the input trajectories is a function of the gradual reduction in airspeed as the aircraft gains altitude. Figures 14 and 15 show the corresponding simulations at an altitude of 27500 ft. Again, the load factor transient is substantially finished 6 s after the first change in setpoint, and the CAP is consistent. The CAP at this altitude is slightly lower than at 5000 ft but its variation with mass and airspeed remains predictable. Two

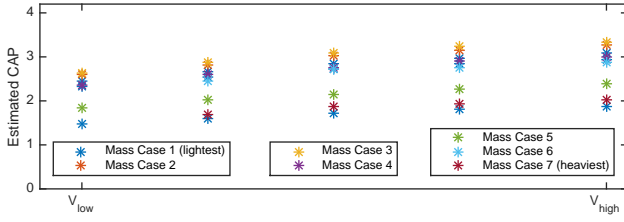


Figure 15: Estimated CAP (no airspeed) in nonlinear RECONFIGURE benchmark for clean configuration at 27500 ft with mass-CoG-altitude scheduled control law

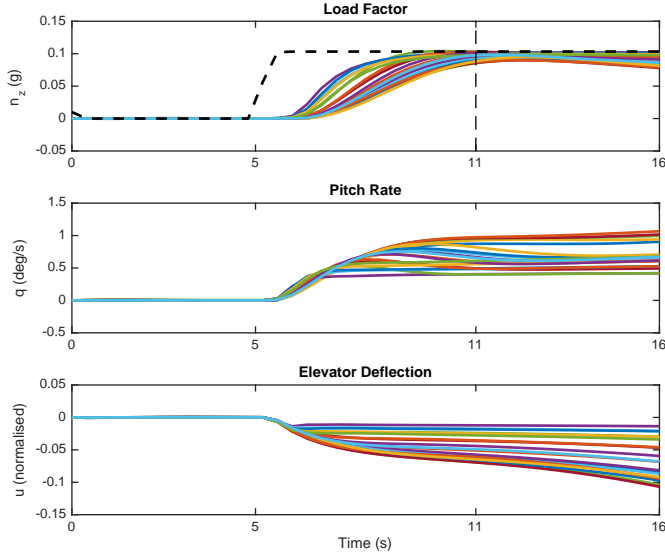


Figure 16: Closed-loop simulation (no airspeed) in nonlinear RECONFIGURE benchmark: all design points for non-clean (high lift) configurations at 5000 ft with mass-CoG-altitude scheduled control law

of the pitch rate trajectories in Figure 14 appear anomalous with larger pitch rate overshoots, despite adequate load factor trajectories. These correspond to the two highest mass cases (both substantially above maximum landing weight) at the lowest airspeeds in the grid of design points. The overshoot is commensurate with that achieved by the fully scheduled benchmark control law (not shown) with autothrust disabled (and the flight points are below “VLS”, the lowest selectable speed with autothrust enabled).

Figure 16 shows responses for the 20 non-clean (high lift) design points. These design points cover a substantially lower set of air speeds than for the clean configurations. Only the initial transient is shown, since with the protection loops disabled some of the trajectories enter a stall due to low airspeed. The CAP varies more than in the clean configuration (seen from the initial gradient of q), but it should be noted that these simulations cover 4 distinct slat/flap configurations. The slight tracking offset is to be expected, due to airspeed changes in the absence of autothrust acting as a ramp disturbance, whilst the control law only has a single integrator.

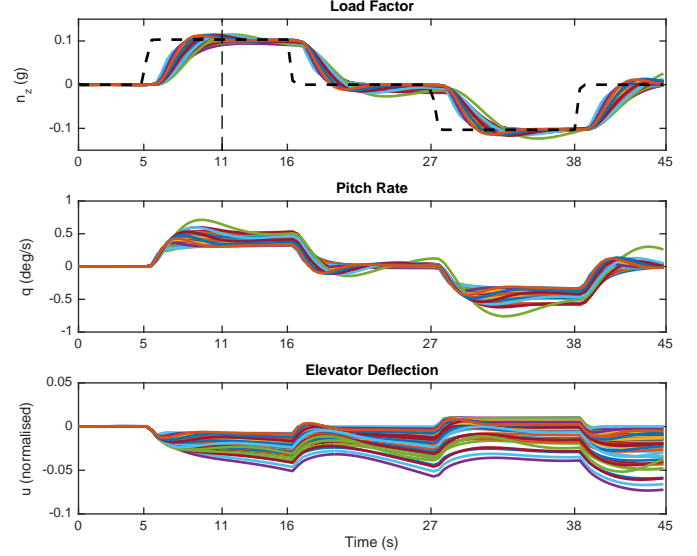


Figure 17: Closed-loop simulation (altitude only) in nonlinear RECONFIGURE benchmark: all design points at 5000 ft with altitude-only scheduled control law

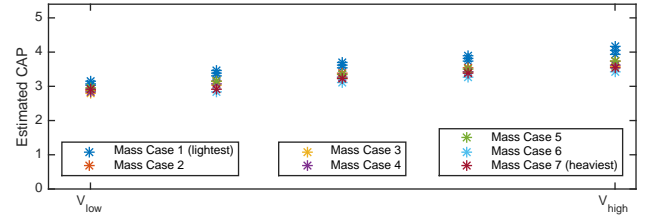


Figure 18: Estimated CAP (no airspeed) in nonlinear RECONFIGURE benchmark for clean configuration at 5000 ft with altitude-only scheduled control law

4.1.2. Airspeed, mass and CoG estimates missing

Figure 17 shows the closed-loop responses for an altitude of 5000 ft to a control law synthesised as described in Section 2.3.2 to depend upon neither airspeed, CoG nor mass estimates, i.e. for a given altitude each simulation will use the same control law. Performance is adequate, except for the high-mass, rear CoG and low airspeed scenario, in which, due to absence of protection loops and autothrottle, the airspeed deviates from the design envelope and the quality of control deteriorates. Elevator responses are more “cautious” than the scenario where CoG and mass are available. The estimated CAP for these simulations is shown in Figure 18. It is a little bit lower than for the mass-CoG-altitude scheduled control law, but still consistent (this can also be seen from the initial gradient of q being similar for all scenarios in Figure 17).

4.2. Flight envelope protection

Figure 19 shows the results using the nonlinear RECONFIGURE benchmark with the same inner loop control law as demonstrated in Section 4.1, and the outer loop protection system described in Section 3. (Numerical val-

ues of units cannot be shown due to the RECONFIGURE confidentiality constraints.) The unused, but implicitly constrained, parameters V_{CAS} and α are also shown in this experiment to verify the effectiveness of the approach. The simulations consider an initial altitude of 5000 ft, for a mass case at the upper limit of the permitted envelope, the CoG at its rearmost position, an initial V_{CAS} approximately mid-way between stall speed and maximum speed, and a clean slat/flap configuration. Between $t = 5$ and $t = 15$, the maximum load factor is commanded. The engine thrust is held at the trim equilibrium value. Three scenarios are shown — the first with no wind, the second with a tailwind entering at $t = 10$, and the third with a headwind entering at $t = 10$. In all cases, the load factor command is attenuated substantially, but an initial increase in flight path angle is visible. As the speed nears its lower bound, the flight path angle falls back. In the case of no wind, and a headwind, the angle of attack remains far from the upper limit. On the other hand, in the case of the tailwind, the angle of attack transiently reaches its upper limit before converging to a value slightly below it. It should be noted, that to recover a climb whilst meeting the imposed envelope limits, more engine thrust would be needed to increase the speed. The pilot must command this, or a ground-speed based autothrust law could be implemented, but that is beyond the scope of this paper.

5. Conclusions

A robust load-factor tracking control law for the longitudinal dynamics of an aircraft has been proposed and demonstrated in linear and nonlinear simulation. The performance requirements are met over a large region of the flight envelope, and a further degraded control law which does not require even mass or centre of gravity estimates has also been demonstrated. Heavy computation is relegated to the offline design phase, and the online computational task involves only simple operations. In comparison to Hartley and Maciejowski (2015), where a preliminary version of this work was presented, the behaviour between design points has been more rigorously considered. In addition, a substitute for the invalidated outer angle of attack and airspeed protection loops has been implemented, based upon command limiting as a function of ground speed, *a priori* bounds on the wind speed and an *a priori* model of the trim angle of attack and load factor behaviour. The proposed approaches do not pose any fundamental incompatibility with existing software implementation approaches. Nor do they require substantial conceptual changes in high level architecture, with a one-to-one mapping between their roles and the systems that they are intended to take over from during the fault scenario. As part of the concluding stages of the RECONFIGURE project, the designs have been submitted for verification and benchmarking using the tool described in Fernández et al. (2015). Future work will relax the architectural and coding compatibility considerations, and investigate to what degree replacing the override rules

in the protection loop with a more complex online optimisation based MPC scheme can reduce the conservatism of the ultimate achievable, compliant, load factor.

6. Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement number 314 544, project “RECONFIGURE”. The authors would like to thank Philippe Goupil and Josep Boada-Bauxell from Airbus Operations SAS for their support with the RECONFIGURE benchmark simulator.

- Brière, D., Favre, C., Traverse, P., 1995. A family of fault-tolerant systems: electrical flight controls, from Airbus A320/330/340 to future military transport aircraft. *Microprocessors and Microsystems* 19 (2), 75–82.
- Cuzzola, F. A., Geromel, J. C., Morari, M., 2002. An improved approach for constrained robust model predictive control. *Automatica* 38 (7), 1183–1189.
- De Oliveira, M. C., Bernussou, J., Geromel, J. C., 1999. A new discrete-time robust stability condition. *Syst. Control Lett.* 37 (4), 261–265.
- De Oliveira, M. C., Geromel, J. C., Bernussou, J., 2002. Extended H_2 and H_∞ norm characterizations and controller parametrizations for discrete-time systems. *Int. J. Control* 75 (9), 666–679.
- Ding, B., Zou, T., 2014. A synthesis approach for output feedback robust model predictive control based-on input-output model. *J. Process Contr.* 24 (3), 60–72.
- Falkena, W., Borst, C., Chu, Q. P., Mulder, J. A., 2011. Investigation of practical flight envelope protection systems for small aircraft. *Journal of Guidance, Control, and Dynamics* 34 (4), 976–988.
- Favre, C., 1994. Fly-by-wire for commercial aircraft: the Airbus experience. *Int. J. Control* 59 (1), 139–157.
- Fernández, V., Montañó, J., Recupero, C., Kerr, M., Rosa, P., Boada-Bauxell, J., Dalbies, L., September 2–4 2015. A tool for industrial verification and benchmarking of FDD/FTC designs. In: *Proc. 9th IFAC Symp. on Fault Detection, Supervision and Safety for Technical Processes*. Paris, pp. 1006–1011.
- Goupil, P., 2011. AIRBUS state of the art and practices on FDI and FTC in flight control system. *Control Eng. Pract.* 19 (6), 524–539.
- Goupil, P., Boada-Bauxell, J., Marcos, A., Cortet, E., Kerr, M., Costa, H., August 24–29 2014. AIRBUS efforts towards advanced real-time fault diagnosis and fault tolerant control. In: *Proc. 19th IFAC World Congress*. Cape Town, South Africa, pp. 3471–3476.
- Goupil, P., Boada-Bauxell, J., Marcos, A., Rosa, P., Kerr, M., Dalbies, L., September 2–4 2015. An overview of the FP7 RECONFIGURE project: industrial, scientific and technological objectives. In: *Proc. 9th IFAC Symp. on Fault Detection, Supervision and Safety for Technical Processes*. Paris, pp. 976–981.
- Granado, E., Colmenares, W., Bernussou, J., García, G., December 12–15 2005. LMI based robust output feedback MPC. In: *Proc. 44th IEEE Conf. Decision and Control, and European Control Conf. (CDC-ECC)*. Seville, pp. 5764–5769.
- Hartley, E. N., Maciejowski, J. M., September 2–4 2015. A longitudinal flight control law based on robust MPC and H_2 methods to accommodate sensor loss in the RECONFIGURE benchmark. In: *Proc. 9th IFAC Symp. on Fault Detection, Supervision and Safety for Technical Processes*. Paris, pp. 1000–1005.
- Kothare, M. V., Balakrishnan, V., Morari, M., 1996. Robust constrained model predictive control using linear matrix inequalities. *Automatica* 32 (10), 1361–1379.
- Löfberg, J., 2004. YALMIP: A toolbox for modeling and optimization in MATLAB. In: *Proc. CACSD Conference*. Taipei, Taiwan. URL <http://control.ee.ethz.ch/~joloef/yalmip.php>
- Maciejowski, J. M., 2002. *Predictive Control with Constraints*. Pearson Education.

Puyou, G., Ezerzere, P., June 20–22 2012. Tolerance of aircraft longitudinal control to the loss of scheduling information: toward a performance oriented approach. In: Proc. 7th IFAC Symp. Robust Control Design. Aalborg, Denmark, pp. 393–399.

Rugh, W. J., Shamma, J. S., 2000. Research on gain scheduling. *Automatica* 36 (10), 1401–1425.

Varga, A., Ossmann, D., Joos, H.-D., August 24–29 2014. A fault diagnosis based reconfigurable longitudinal control system for managing loss of air data sensors for a civil aircraft. In: Proc. 19th IFAC World Congress. Cape Town, South Africa, pp. 3489–3496.

Wan, Z., Kothare, M. V., 2002. Robust output feedback model predictive control using off-line linear matrix inequalities. *J. Process Contr.* 12 (7), 763–774.

Well, K. H., August 21–24 2006. Aircraft control laws for envelope protection. In: AIAA Guidance, Navigation, and Control Conference and Exhibit. Keystone, CO.

Zhou, K., Doyle, J. C., Glover, K., 1996. *Robust and Optimal Control*. Prentice Hall.

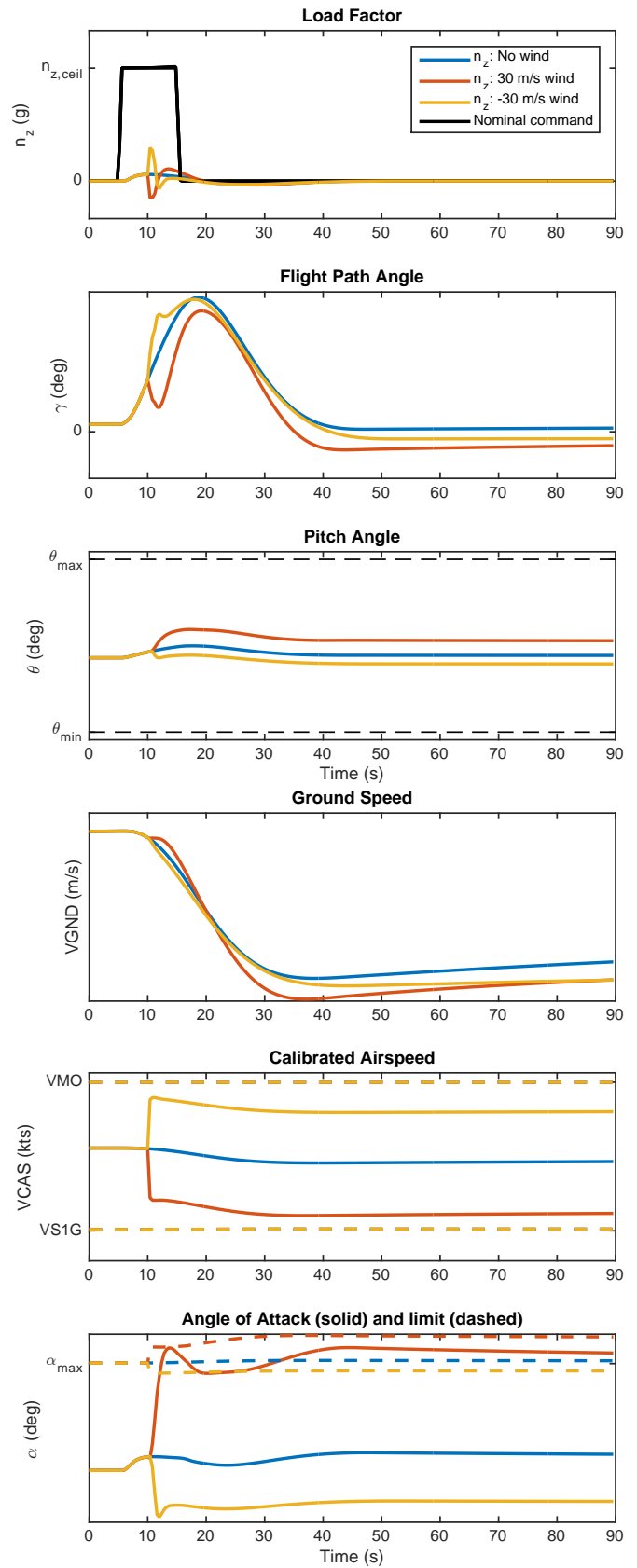


Figure 19: Demonstration of flight envelope protection for clean slat/flap configuration at 5000 ft, for heavy, rear CoG aircraft