ARTICLE IN PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Control Engineering Practice ▮ (▮▮▮▮) ▮▮▮–▮▮▮

CONTROL ENGINEERING PRACTICE

www.elsevier.com/locate/conengprac

Stabilized MPC formulations for robust reconfigurable flight control

M.M. Kale, A.J. Chipperfield*

School of Engineering Sciences, Computational Engineering Design Group, University of Southampton, Highfield, Southampton, England SO17 1BJ, UK

Received 4 August 2003; accepted 6 September 2004

Abstract

Model predictive control (MPC) solves an optimization at every sampling instance to achieve commanded set points and control objectives subject to constraints on control inputs and system states. Such online optimization can incorporate many important factors to enable the control of systems subject to faults, changing dynamics, changing control objectives, failed control inputs and large disturbances. Reconfigurable flight control is one such application where the capabilities and flexibility of optimization-based control methods can be fully utilized and exploited. This paper presents formulations and experimental evaluations of various MPC schemes applied to a realistic full envelope non-linear model of a fighter aircraft. Investigations are carried out by exploring a variety of scenarios of fault and disturbance combinations along with modified and robust formulations of online constrained optimization.
© 2004 Elsevier Ltd. All rights reserved.

Keywords: Model predictive control; Fault tolerance; Constraint softening; Pre-stabilization

1. Introduction

One principle objective of control system design is to accommodate faults and changes in underlying system dynamics while achieving tracking and regulation of performance variables. A modern fighter aircraft is a complex system with many non-linearities affecting its dynamics over the flight envelop. Depending on the flight parameters such as airspeed, angle of attack (AoA) and altitude, the dynamics of the aircraft change significantly. In realistic scenarios such as high-amplitude manoeuvres and high AoA flights, stall-induced non-linearities and abrupt changes in dynamic behaviour pose hard problems to underlying control and stabilization systems. Furthermore, many modern agile fighters are inherently unstable and the control system must augment stability to such an unstable airframe so that the pilot can benefit from high agility and manoeuvrability (Pratt, 1999). At the same time, increased agility should not overly increase the pilot's workload.

In addition to the above factors, fault tolerance to system and component failures is a desired feature of modern flight control systems. An aircraft is controlled by the deflection of aero-surfaces such as canards and elevons. Common faults that can arise are failures of sensors measuring critical aircraft parameters (such as body rates and aerodynamic angles), and failures of actuators affecting the control surfaces. The conventional approach to achieving fault tolerance to such failures is through physical redundancy. Essentially, the control and measurement channels are duplicated in hardware and, for flight control systems triple or quadruple redundancy is relatively common. The major disadvantage of such physical redundancy is increased cost and complexity. Replicated channels also add weight to the aircraft and increase maintenance requirements. To overcome these and other difficulties, the concept of reconfigurable control has been developed. Reconfigurable control is based on the principle of exploiting inherent redundancies within system dynamics, i.e. analytic redundancy (Maybeck & Stevens, 1991; Patcher & Miller, 1997; Bodson, 1997). For example, input redundancy is often possible where a controlled variable of the dynamic system can be affected by

*Corresponding author. Tel.: +44 23 8059 8344; fax: +44 23 8059 7802.
E-mail address: a.j.chipperfield@soton.ac.uk (A.J. Chipperfield).

0967-0661/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.conengprac.2004.09.001

more than one system input. Then, based on the dynamic models of the system, a suitable method can often be devised to accommodate loss of a control input by a suitable combination of other inputs. The severity of such faults in system, of course, increases in the case of battle damages. Here the partial or complete loss of aerodynamic surfaces may occur. A reconfigurable control architecture should also effectively and efficiently accommodate such fault scenarios.

Some challenges for reconfigurable flight control system (RFCS) design have been identified (Bodson, 1993) as:

- After a fault occurs, strong cross couplings between modes usually appear. The aircraft loses its symmetry after surface damage (i.e. damage to body, wings or movable surfaces) and conventional simplified separated longitudinal and lateral direction control approaches may not be applicable.
- Under fault conditions, the dynamics of an aircraft can change significantly. Hence, trim values and the system $\boldsymbol{A}$ and $\boldsymbol{B}$ matrices (i.e. linearized dynamic model) used by the controller change after failures. Thus, the continuous use of a non-linear or adaptive control algorithm is required.
- The system may be highly unstable, leaving very little time for control reconfiguration. This demands extremely efficient online identification and controller redesign algorithms.
- After an aircraft has sustained damage to a surface, its ability to produce the required control forces and moments degrades. Hence, the demands on healthy and available actuators (i.e. deflection and deflection rates) will increase. This also aggravates the control saturation problem (Patcher, Chandler, & Mears, 1995).

A reconfigurable controller should be capable of redistributing and co-ordinating available control effort during system failures among the aircraft's remaining effective control surfaces, such that satisfactory flight performance is retained if at all possible. Modern fighter aircraft usually have multiple control surfaces, each of which is capable of independent movement. Controller reconfiguration should therefore exploit such redundancy to achieve acceptable performance. Perhaps the most significant difficulty in designing control laws for a damaged aircraft is accounting for non-linearities. High-amplitude manoeuvres and damaged aero-surface conditions render small perturbation linearizations of aircraft dynamics unreliable. Hence, an RFCS requires a continuous parameter identification technique. This is similar to indirect adaptive control (Eberhardt & Ward, 1999; Ward, Monaco, & Bodson, 1998). The online parameter identification estimates stability and control derivatives of the aircraft for use by the control law.

Online system identification coupled with model predictive control (MPC) design is a promising starting point for such control reconfiguration schemes. Tight tracking requirements usually call for an effective high open-loop gain that can lead to actuator saturations and state limit violations. Such hard limits on the system severely limit the achievable performance of linear controllers designed using conventional optimal control methods (such as LQR, H_∞ and μ synthesis) and may cause instability in a feedback stabilized linearly stable control system. However, unlike linear control design methods for RFCS, MPC can directly accommodate system input and output constraints (Markerink, Bennani, & Mulder, 1997). Related work includes RFCS designs presented in Eberhardt and Ward (1999) and Ward et al. (1998). As discussed previously, under fault conditions when some of the control inputs malfunction, more demands are placed on other healthy inputs to accommodate the functionality of failed inputs while achieving appropriate tracking responses to commands. Due to these increased demands, the problem of control input saturation becomes more pronounced and frequent. Hence, a realistic reconfigurable control methodology must consider such hard saturation limits during online redesign of the control laws. Actuator rate saturation is another dominant feature that must be considered in such control redesign. The ability to incorporate input and state constraints directly in the control algorithm is a unique feature of MPC. Explicit handling of constraints in online controller redesign reduces the need for tuning and manual intervention. This attribute also reduces the need for ad hoc 'safety jackets' and other software layers to handle problems such as integrator wind-up and daisy chaining of actuators. Also, if MPC uses a linear system model for prediction and optimization, results from linear system theory can be readily applied to the synthesis and analysis procedures. Recent developments in interior-point (IP) algorithms (Rao, Wright, & Rawlings, 1997) and active set methods (Bartlett, Wachter, & Beigler, 2000), coupled with significant improvements in onboard computing hardware are making the issues related to computational complexity of online optimization gradually diminish. One desired feature of reconfigurable/adaptive control algorithms is the ability to use (or account for) uncertainly in the online parameter estimates. Aircraft dynamics are expected to be highly non-linear and uncertain under severe failure conditions. Hence, the reconfigurable control algorithm should be robust to unmodeled and/or uncertain dynamics (Bodson, 1993).

The outline of this paper is as follows. The aircraft simulation model and background information are discussed in Section 2. The experimental set-up and required control objectives are then summarized in Section 3. In Section 4, MPC theory with its reconfigurable control capabilities and appropriate formulations from the point of practical implementation are derived. Experimental results are presented in Section 5. Various MPC formulations from Section 4 are tested for combinations of faults, disturbances and model uncertainties evaluating strengths and drawbacks of each. Conclusions are drawn in Section 6.

2. Description of aircraft model

A full flight envelope, non-linear, six degrees of freedom simulation model of a small rigid fighter aircraft is used for the experimental work presented in this paper. The model data is available from the Aeronautical Research Institute of Sweden (FFA, URL http://www.foi.se/admire). It is designated as ADMIRE (Aero data Model in Research Environment) and it incorporates the Generic Aero data Model (GAM) originally developed by Saab AB. ADMIRE also includes a full non-linear engine model, sensor models and actuator dynamics along with actuator position and rate limits. The aircraft configuration for the model data is of delta-canard type as shown in Fig. 1. It is similar but slightly larger in dimensions to the SAAB JAS Gripen. ADMIRE is implemented in the Matlab/Simulink environment as C S-functions. The flight operational region for ADMIRE is upto Mach number 1.2 ($\approx$400 m/s) and an altitude upto 6000 m. The envelop for AoA is upto 30° and for sideslip angle upto 20°. As the airspeed (V_{air}) increases, the envelop limits placed on aerodynamic angles and control surface deflections decrease due to structural and aerodynamic reasons. The variations in envelop characteristics are also shown in Fig. 1.

The bare aircraft model associated with GAM has 12 internal states which are airspeed (V_{air}), two aerodynamic angles—AoA and sideslip, three rotational body rates (p,q,r), three rotational angles and three linear displacements. Available control effectors are left and right canard, four leading edge flaps (grouped together), four elevons (left and right, inboard–outboard; LIE, LOE, RIE, ROE), rudder and a thrust setting. The model is also equipped with thrust vectoring capabilities in the **Y** and **Z**-axis (dt_y, dt_z). Atmospheric turbulence enters the model in the form of external disturbances. Depending on the amount of fuel loaded, certain combinations of the mass and inertia coefficients can be set as a baseline configuration before the start of simulations. The nominal case simulations assume 60% fuel load along with default mass and inertia parameter values. A detailed description of such adjustments and parameters is available in the ADMIRE documentation (Backstrom, 1997). The dynamic states of the bare airframe model define a flight condition and are inter-related with a set of 12 first-order non-linear differential equations. These equations are defined in the conventional manner. The aircraft aero-data model consists of aero-data tables, interpolation routines and aero-data algorithms. This is a standard way of performing aerodynamic modelling today. Based on the current Mach number and other suitable aircraft states, various aerodynamic coefficients are calculated using aero-data tables and associated interpolation algorithms. The final six aerodynamic coefficients corresponding to total aerodynamic forces and moments along three translational and three rotational axes are then obtained. The geometry reference data is used to convert force and moment coefficients into forces and moments. The geometric data corresponds to wing areas, mean chord lengths and span. These aerodynamic coefficients, forces, moments and engine thrust contributions are then substituted in differential equations to propagate the states further in time using numerical integration. A detailed discussion of such aerodynamic modelling can be found in Etkin and Reid (1996). Contribution of atmospheric disturbances to total aerodynamic forces and moments is also considered in a similar manner. The available atmospheric inputs are three translational body axis wind disturbances and a rotational contribution around the longitudinal axis. For the simulation studies the body axis reference frame is considered. Engine thrust calculations are based on similar data tables which interpolate thrust using altitude and Mach number along with throttle settings. First-order lag dynamics are added to the engine model to represent time taken to accelerate/decelerate the rotating

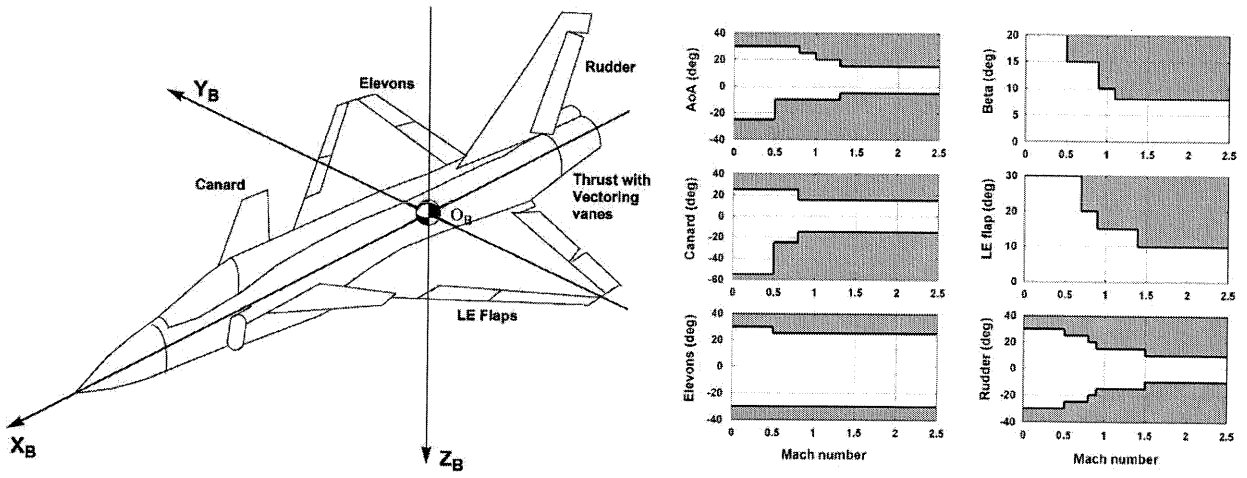

Fig. 1. Delta-canard configuration of aircraft model in ADMIRE data set and its envelop and validity.

Table 1
Configuration-specific values used in the ADMIRE simulations at 0.45 Mach and 3000 m

Trimmed flight data		Envelop limits		Control design limits	Rate limits on actuator deflections	
Vair	147.86 m/s	Roll rate		+50° to −50°/s	Canard	50°/s
Altitude	3000 m	Pitch rate		+25° to −25°/s	Elevons	50°/s
Alpha	3.66767°	Yaw rate		+25° to −25°/s	Rudder	50°/s
Canard	0.01256°	AoA	+30° to −25°	+20° to −15°	dt_y	20°/s
Elevon	0.0998°	Sideslip	+20° to −20°	+5° to −5°	dt_z	20°/s
Throttle	0.07520	Canard	+25° to −55°	+15° to −33°		
Mass	9100 kg	Elevons	+30° to −30°	+18° to −18°		
Fuel	60%	Rudder	+30° to −30°	+20° to −20°		
		Horizontal thrust vanes dt_y	+10° to −10°	+10° to −10°		
		Vertical thrust vanes dt_z	+15° to −15°	+15° to −15°		

parts of engine. The dynamic models for sensors take different forms depending on the measured quantity. For air data sensors such as airspeed, AoA and sideslip, first-order lag dynamics are considered. For inertial sensors measuring body rates and accelerations, second-order lead lag dynamics are used. Aero-surface actuator dynamics are represented with first-order lag models. The sign of the actuator deflections follows a standard right-hand rule. A positive deflection means movement of control surfaces towards the base of the aircraft. Some important configuration-specific values of ADMIRE are specified in Table 1. For control design purposes, the sensor and actuator dynamics are not considered, hence the control laws are expected to be robust to such ignored dynamics.

Trimming and linearization routines are provided with the ADMIRE model set. The non-linear simulation model can be trimmed at a desired flight condition of Mach number, altitude and AoA. The trimming routine performs a non-linear search and optimization to find various deflections of control surfaces and a thrust setting which maintain the flight condition in a steady manner. The linearization facility is then used to obtain a small perturbation linear model of the bare aircraft around trimmed flight condition. Such a linear model is then used for control system design and further analysis.

3. Experimental set-up

The nominal flight condition considered for the experiments reported in this paper is Mach number 0.45 at an altitude of 3000 m. This is an unstable point in the pitch axis and a likely condition during a segment of combat mission. At the start of each simulation run, the non-linear aircraft model is first trimmed at this flight condition. The control design is based on a reduced-order linear model using six states of the aircraft which are: V_{air} (airspeed), AoA (α), sideslip (β), p (roll rate), q (pitch rate) and r (yaw rate).

The control inputs are left and right canards, left and right inboard and outboard elevons, rudder, and, vertical and horizontal thrust vectoring deflections. As there is only one engine for propulsion, reconfiguration for velocity control is not considered. The reconfigurable MPC is expected to provide inner loop stability and command augmentation system which can tolerate various faults while maintaining acceptable command tracking performance.

A block schematic of various components within the experimental set-up is shown in Fig. 2. The Atmospheric Disturbance Model block uses Dryden wind turbulence models. The turbulence is considered as a stochastic process defined by the standard Dryden velocity spectra and is implemented by passing band-limited white noise through appropriate forming filters as described in Etkin and Reid (1996). In addition to baseline turbulence, triangular shape wind gusts active in vertical and lateral axis are also considered.

The Stick Shaping and Non-linear Compensation module accepts stick commands from the pilot. The pilot commanded pitch rate (q_c), wind axis roll rate (p_{wc})and sideslip angle (βc) are shaped through stick filters and limited according to structural limits and failure information. This block also contains non-linear compensations that are needed for the turn co-ordination (i.e. minimum coupling of rolling and yawing motions in a steady turn; see, e.g. McLean, 1990). As the velocity vector (wind axis) roll rate is commanded, commands for both body axis roll rate and body axis yaw rate are generated. All these command signals are then passed through low-order system models that specify handling qualities required. The standard flight handling qualities specifications are used to define the objectives of the proposed reconfigurable control strategies. A damaged aircraft is often significantly less symmetric (and therefore less decoupled) than the original one. The control objectives hence need to be adjusted according to damage conditions. The military aircraft handling qualities standard specifies lower-order equivalent system (LOES)

algorithm. To implement online system identification like functionality, Model Change Logic modifies the linear dynamic models of the aircraft used by MPC according to flight conditions and fault descriptions. The trimming and linearization routines were used to obtain a number of linear models at different flight conditions based on changes in Mach number, altitude and AoA around the starting point of 0.45 Mach and 3000 m. The Model Change Logic selects an appropriate model depending on the measured flight conditions, then changes it according to fault and uncertainty settings and finally updates it after FDI delay for the use of MPC algorithm.

4. Reconfiguration strategies using MPC

The linearized aircraft dynamics can be represented by the following time varying non-homogeneous continuous time linear differential equations:

$$\dot{x}(t) = \boldsymbol{A}_c(t)x(t) + \boldsymbol{B}_c(t)\boldsymbol{u}(t) + \boldsymbol{b}(t), \tag{1}$$

where $\boldsymbol{A}_c(t)$ are stability derivatives, $\boldsymbol{B}_c(t)$ are the control derivatives and $\boldsymbol{b}(t)$ is a bias that includes higher-order terms and accelerations due to non-equilibrium or off-trim conditions. $\boldsymbol{x} \in \Re^n$ and $\boldsymbol{u} \in \Re^m$ are the state and control vectors, respectively. For the studies considered here, full state measurements are assumed. Measurement noise is not explicitly considered and the existence of a suitable Kalman state estimator is assumed. The continuous time model is discretized appropriately for controller design.

The principle behind MPC is based on the repetitive minimization of cost function:

$$V(k) = \sum_{l=1}^{H_p} ||\hat{x}(k+l|k) - \hat{\boldsymbol{T}}(k+l|k)||_{\boldsymbol{Q}}^2 + \sum_{l=0}^{H_u-1} ||\hat{\boldsymbol{u}}(k+l|k)||_{\boldsymbol{R}}^2 \tag{2a}$$

subject to constraints

$$\hat{\boldsymbol{x}}(k+1) = \boldsymbol{A}\hat{\boldsymbol{x}}(k) + \boldsymbol{B}\hat{\boldsymbol{u}}(k), \tag{2b}$$

$$\hat{\boldsymbol{x}}(k|k) = \boldsymbol{x}(k), \tag{2c}$$

$$\boldsymbol{u}_{\min}(l) \leqslant \boldsymbol{u}(l) \leqslant \boldsymbol{u}_{\max}(l) \text{ where } k \leqslant l \leqslant k + H_u - 1, \tag{2d}$$

$$\Delta\boldsymbol{u}_{\min}(l) \leqslant \Delta\boldsymbol{u}(l) \leqslant \Delta\boldsymbol{u}_{\max}(l) \text{ where } k \leqslant l \leqslant k + H_u - 1, \ \Delta\boldsymbol{u} = \boldsymbol{u}_k - \boldsymbol{u}_{K-1}, \tag{2e}$$

$$\boldsymbol{x}_{\min}(l) \leqslant \boldsymbol{x}(l) \leqslant \boldsymbol{x}_{\max}(l) \text{ where } k + 1 \leqslant l \leqslant k + H_p, \tag{2f}$$

where $\boldsymbol{T}$ is the future state target trajectory vector, $\boldsymbol{Q}$ and $\boldsymbol{R}$ are weights independent of time k and, H_p and H_u are prediction and control horizons, respectively. The target trajectory is generated based on a given set point or command reference. The MPC algorithm drives the predicted state over the prediction horizon, towards the target trajectory and yields a sequence of future control inputs. It is also assumed that the dynamic system defined by the model $(\boldsymbol{A},\boldsymbol{B})$ is controllable. The controllability condition is required to ensure that the MPC optimization solved at each step is feasible in the nominal case. This optimization can be cast as a quadratic programming (QP) problem. To ensure a well-posed optimization problem, the constraints defined on control inputs and states must be consistent and convex. Such a restriction guarantees a convex feasible region containing the origin. The controllability and convexity conditions remained valid during all the simulation studies presented in this paper.

The MPC strategy can accommodate various failures in the following manner (Huzmezan & Maciejowski, 1998):

- A minor fault in an actuator, such as limited deflection, can be represented as a change in position limits used by the MPC optimization algorithm. If an actuator floats it can simply be represented by making the corresponding column of $\boldsymbol{B}$ matrix zero. If there is an actuator hard-over, it can be represented by removing the corresponding column of $\boldsymbol{B}$ and adding a constant disturbance of magnitude $\boldsymbol{b}_f \times u_f$ where u_f is the stuck actuator position and $\boldsymbol{b}_f$ is the corresponding column of $\boldsymbol{B}$ matrix.
- Faults such as the loss of an aero-surface or damage to the wings or body will necessarily change the aircraft's dynamics. Employing a system identification module, $\boldsymbol{A}$ and $\boldsymbol{B}$ matrices representing changed aircraft dynamics can be determined (see Monaco, Ward, & Bird, 1997; Ward et al., 1998). The MPC algorithm uses this new model as its internal model for the prediction and further optimization.
- In cases when the damage and failures are severe, the parameters used in the MPC algorithm, such as prediction and control horizons or state and input weightings may be changed. More advanced MPC algorithms capable of guaranteeing stability and robustness are of interest in such cases. Such formulations are presented in this paper.

5. MPC formulations for fault tolerant and reconfigurable control

5.1. Tracking formulation with integral error augmentation

To achieve zero error set point or command tracking in the presence of model uncertainties and disturbances, it is necessary to augment integral error states to the state space model of the aircraft. The MPC uses the augmented state equation for prediction calculations and subsequent optimizations.

5.2. Soft constraints

A major practical problem with the MPC formulation of Eq. (2) is that the constrained optimization may become infeasible. Assuming well-posed constraint conditions, the infeasibility can occur because of unachievable targets due to restrictions on control inputs and states, limited control moves due to short horizons, excursions into the infeasible region due to uncontrollable disturbances and, large plant-model mismatch (i.e. uncertainty in dynamic models). Hence, when implementing MPC, it is essential to take steps either to avoid posing an infeasible problem or to have an alternative method of computing an appropriate control signal. Constraint softening is one method of avoiding infeasibilities. A soft-constraints-based approach to handling any infeasibility involves the introduction of slack variables. Slack variables are defined such that they are non-zero only if the corresponding constraints are violated. In a soft-constrained MPC, violations of the state constraints are usually allowed. Additional terms penalizing these violations are introduced into the objective function. From a practical perspective, keeping the state constraints tight is not appropriate because of the presence of noise, disturbances and numerical errors. However, the saturation limits on control input variables are, in principle, hard and cannot be softened.

5.3. Zone regions

The standard MPC formulation assumes that all variables of the state vector have a reference trajectory. If some state variables do not have any explicit reference command, the simplest formulation is to assume a zero reference trajectory. However, in most applications, exact values of such states (even if non-zero) are not important, so long as they remain within specified boundaries or zones. For flight control applications, variables such as AoA are usually not commanded but must be restricted within certain ranges due to aerodynamic considerations. In exceptional conditions, such as large disturbances, these ranges might be violated. The zone regions may also be necessary for over-specified systems. If there are too many input and output constraints, the system may not be able to meet all the set points/commanded references simultaneously. If a set point is changed to zone region, the performance specifications are slightly relaxed. Introducing zone regions for appropriate variables thus increases the probability of a system meeting the specifications for other commanded variables (Wang, 2002). A further advantage of using zone regions in multivariable/multiobjective systems arises due to the possibility of reference trajectories for different outputs possibly being inconsistent with each other. For example, in the case of flight control, AoA and pitch rate variables are closely inter-related and any command tracking of pitch rate demands a certain AoA profile that depends on the aircraft's longitudinal dynamics. Attempting to keep perturbations in AoA to a minimum whilst following a certain pitch rate command can introduce conflicting objectives. Such a control problem can greatly benefit by the formulation of a zone region for AoA. The zone-based MPC is reformulated in terms of soft constraints assigned to the zone region violations.

5.4. Stabilized predictions

A naïve application of the MPC to an unstable system can lead to severe numerical problems in evaluating the prediction equations. The prediction equations involve computing A^i, hence for unstable systems and large values of i, some elements of A^i may become extremely large relative to others and also relative to elements in lower powers of A. Hence, finite precision computer arithmetic can sometimes lead to incorrect results. As demonstrated by Rossiter, Kouvaritakis, and Rice (1998), the optimization solved on the basis of such predictions can be highly ill-conditioned and the solution will be far from optimal, resulting in instability and a lack of robustness. An effective way to prevent this anomaly is to pre-stabilize the prediction equations. The standard MPC prediction equation assumes that the system is operating in 'open-loop' and control signals are computed as deviations from a nominal value. Furthermore, at the end of control horizon, input signals are assumed to remain constant thereafter. Clearly, there is a plenty of scope to explore this baseline 'do-nothing' control policy. A straightforward strategy is to assume state feedback as a baseline controller to which MPC control signals are added. The introduction of state feedback has multiple

implications as it provides an effective framework for robust MPC design (Kothare, Balakrishnan, & Morari, 1996). If the state feedback makes system stable or matrix A nilpotent, numerical and ill-conditioning problems associated with open-loop unstable predictions are also avoided. The pre-stabilization is also an effective tool to guarantee nominal closed-loop stability using the MPC controller (assuming the existence of a feasible solution to the constrained optimization). Many modifications of MPC that guarantee stability, usually introduce additional constraints or modify the objective function with appropriate terms. Some of these methods and corresponding implementations involve adding terminal inequality constraint sets at the end of prediction horizon. Such an optimization can be implicitly reformulated in terms of pre-stabilization instead of explicit stability constraints (Rossiter et al., 1998). Stability proofs of such formulations are given in Keerthi and Gilbert (1988) Rawling and Muske (1993) and Maciejowski (2002).

Pre-stabilization also offers significant benefits when disturbances are present in the system. This holds true for both stable and unstable systems (and a similar argument applies to modelling of uncertainties as well). The use of stabilized predictions attempts to keep the online constrained optimization feasible and reduce conservatism in optimal control moves by counteracting to the adverse effects of disturbances and model uncertainties. This is because, in open-loop due to baseline 'do-nothing' control policy, the effect of disturbances is passively suffered and as a result, the uncertainty produced by the disturbances grows (accumulates) with time over the prediction horizon. Hence, the control input calculated on the basis of such contrived predictions can be conservative as well as pessimistic about the actual feasibility of the online constrained optimization (Bemporad, 1998). Assuming a base-line stabilizing state feedback controller, consider a state feedback gain $\boldsymbol{K}_u$ as

$$\hat{\boldsymbol{u}}(k+i|k) = \begin{cases} -\boldsymbol{K}_u\hat{\boldsymbol{x}}(k+i|k) + \boldsymbol{q}_i, & i = 0,\ldots,H_u-1 \\ -\boldsymbol{K}_u\hat{\boldsymbol{x}}(k+i|k), & i \geqslant H_u \end{cases} \quad \text{and } \hat{\boldsymbol{x}}(k+i|k) \to 0, \quad i \to \infty. \tag{3a}$$

As the baseline feedback controller remains active after H_u, it must also satisfy related control constraints. Hence, the MPC control moves must bring the state at the end of horizon within a terminal region, such that the baseline control law remains valid thereafter. To maximize this terminal set, the feedback gain can be chosen to be one that merely pushes the unstable modes inside the unit circle. The cost of the baseline control is then added to MPC cost function as $\boldsymbol{x}(k+Hu|k)T\bar{\boldsymbol{Q}}\cdot\boldsymbol{x}(k+Hu|k)$ where, $\bar{\boldsymbol{Q}}$ is the terminal penalty matrix computed by solving an appropriate Lyapunov equation as described in Lee (2000).

5.5. MPC algorithm solution

Assuming, $\boldsymbol{A}_K = \boldsymbol{A} - \boldsymbol{B}_U\boldsymbol{K}_u$, state equation for the system with integral error augmentation as in Fig. 3 can be written as

$$\underbrace{\begin{bmatrix} x(k+1) \\ e(k+1) \end{bmatrix}}_{\substack{new \\ x(k+1)}} = \underbrace{\begin{bmatrix} \boldsymbol{A}_K & \boldsymbol{0} \\ -\boldsymbol{H} & \boldsymbol{I} \end{bmatrix}}_{\substack{new \\ A}} \underbrace{\begin{bmatrix} \boldsymbol{x}(k) \\ e(k) \end{bmatrix}}_{\substack{new \\ x(k)}} + \underbrace{\begin{bmatrix} \boldsymbol{B}_U \\ \boldsymbol{0} \end{bmatrix}}_{\substack{new \\ B_U}} \boldsymbol{q}(k) + \underbrace{\begin{bmatrix} \boldsymbol{B}_F \\ \boldsymbol{0} \end{bmatrix}}_{\substack{new \\ B_F}} \boldsymbol{u}_F(k) + \underbrace{\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{H} \end{bmatrix}}_{B_2} \boldsymbol{R}_x(k). \tag{3b}$$

In order to obtain the sequence of control inputs, Eq. (2) needs to be solved while taking into account the issues discussed in previous subsections. To achieve this, the performance objective, system dynamic equations and constraints are formulated as a QP problem. Assuming the availability of the current measured state, $\boldsymbol{x}(k)$, previously applied input, $\boldsymbol{u}(k-1)$, and fault information from actuators, $\boldsymbol{u}_F$, at instance k, the state equation can be expanded over

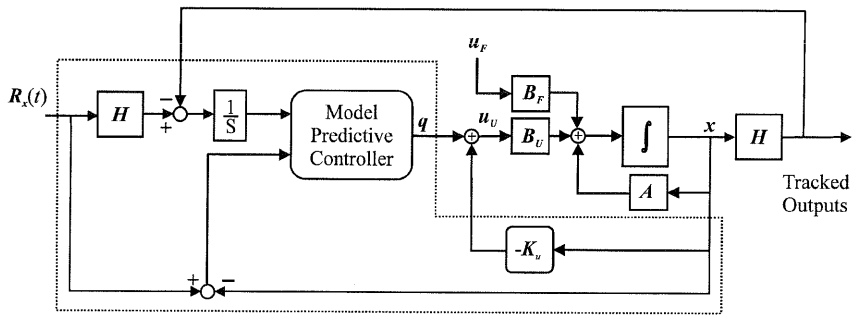

Fig. 3. Pre-stabilization-based MPC configuration.

the prediction and control horizons and rewritten in matrix form as

$$\boldsymbol{X}(k) = \psi x(k) + \Pi \boldsymbol{R}_x(k) + Y_F \boldsymbol{u}_F(k) + \Theta \mathbf{Q}(k) \tag{4a}$$

with

$$\boldsymbol{X}(k) = [\,\hat{\boldsymbol{x}}(k+1|k)^{\mathrm{T}} \quad \cdots \quad \hat{\boldsymbol{x}}(k+H_u|k)^{\mathrm{T}} \quad \cdots \quad \hat{\boldsymbol{x}}(k+H_p|k)^{\mathrm{T}}\,]^{\mathrm{T}},$$
$$\mathbf{Q} = [\,\hat{\boldsymbol{q}}(k|k)^{\mathrm{T}} \quad \hat{\boldsymbol{q}}(k+1|k)^{\mathrm{T}} \quad \hat{\boldsymbol{q}}(k+2|k)^{\mathrm{T}} \quad \cdots \quad \hat{\boldsymbol{q}}(k+H_u-1|k)^{\mathrm{T}}\,]^{\mathrm{T}},$$

$$\psi = \begin{bmatrix} \boldsymbol{A} \\ \vdots \\ \boldsymbol{A}^{H_u} \\ \vdots \\ \boldsymbol{A}^{H_p} \end{bmatrix}, \quad \Pi = \begin{bmatrix} \boldsymbol{B}_2 \\ \vdots \\ \sum_{i=0}^{H_u-1} \boldsymbol{A}^i \boldsymbol{B}_2 \\ \vdots \\ \sum_{i=0}^{H_p-1} \boldsymbol{A}^i \boldsymbol{B}_2 \end{bmatrix}, \quad Y_F = \begin{bmatrix} \boldsymbol{B}_F \\ \vdots \\ \sum_{i=0}^{H_u-1} \boldsymbol{A}^i \boldsymbol{B}_F \\ \vdots \\ \sum_{i=0}^{H_p-1} \boldsymbol{A}^i \boldsymbol{B}_F \end{bmatrix},$$

$$\Theta = \begin{bmatrix} \boldsymbol{B}_U & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}^{H_u-1}\boldsymbol{B}_U & \boldsymbol{A}^{H_u-2}\boldsymbol{B}_U & \cdots & \sum_{i=0}^{H_u-H_u} \boldsymbol{A}^i \boldsymbol{B}_U \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{A}^{H_p-1}\boldsymbol{B}_U & \boldsymbol{A}^{H_p-2}\boldsymbol{B}_U & \cdots & \sum_{i=0}^{H_p-H_u} \boldsymbol{A}^i \boldsymbol{B}_U \end{bmatrix}.$$

Similarly, the outputs can be described as

$$\boldsymbol{U}_U(k) = Y_X \boldsymbol{x}(k) + Y_{UF} \boldsymbol{u}_F(k) + Y_R \boldsymbol{R}_x(k) + \Phi \mathbf{Q}(k) \tag{4b}$$

with

$$Y_X = \begin{bmatrix} -\boldsymbol{K} \\ -\boldsymbol{K}\boldsymbol{A} \\ -\boldsymbol{K}\boldsymbol{A}^2 \\ \vdots \\ -\boldsymbol{K}\boldsymbol{A}^{H_u-1} \end{bmatrix}, \quad Y_{UF} = \begin{bmatrix} \mathbf{0} \\ -\boldsymbol{K}\boldsymbol{B}_F \\ -\boldsymbol{K}[\boldsymbol{A}\boldsymbol{B}_F + \boldsymbol{B}_F] \\ \vdots \\ -\boldsymbol{K}\sum_{i=0}^{H_u-2} \boldsymbol{A}^i \boldsymbol{B}_F \end{bmatrix}, \quad Y_R = \begin{bmatrix} \mathbf{0} \\ -\boldsymbol{K}\boldsymbol{B}_2 \\ -\boldsymbol{K}[\boldsymbol{A}\boldsymbol{B}_2 + \boldsymbol{B}_2] \\ \vdots \\ -\boldsymbol{K}\sum_{i=0}^{H_u-2} \boldsymbol{A}^i \boldsymbol{B}_2 \end{bmatrix},$$

$$\Phi = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ -\boldsymbol{K}\boldsymbol{B}_U & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ -\boldsymbol{K}\boldsymbol{A}\boldsymbol{B}_U & -\boldsymbol{K}\boldsymbol{B}_U & \mathbf{I} & \vdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\boldsymbol{K}\boldsymbol{A}^{H_u-2}\boldsymbol{B}_U & -\boldsymbol{K}\boldsymbol{A}^{H_u-3}\boldsymbol{B}_U & -\boldsymbol{K}\boldsymbol{A}^{H_u-4}\boldsymbol{B}_U & \cdots & \mathbf{I} \end{bmatrix},$$

where $\boldsymbol{K} = [\boldsymbol{K}_u \; \mathbf{0}]$ is the feedback matrix for new integral error augmented state. The differential outputs are written as

$$\Delta \boldsymbol{U}_U(k) = \Xi \boldsymbol{x}(k) + Y_{DR} \boldsymbol{R}_x(k) + Y_{DUF} \boldsymbol{u}_F(k) + Y \boldsymbol{u}(k-1) + \Lambda \mathbf{Q}(k) \tag{4c}$$

with $\Delta U_U(k) = [\Delta\hat{u}(k|k)^T \quad [\Delta\hat{u}(k+1|k)^T \quad \cdots \quad \Delta\hat{u}(k+H_u-1|k)^T]^T$,

$$\Xi = \begin{bmatrix} -K \\ -K[A-I] \\ -KA[A-I] \\ -KA^2[A-I] \\ \vdots \\ -KA^{H_u-2}[A-I] \end{bmatrix}, \quad Y_{DR} = \begin{bmatrix} 0 \\ -KB_F \\ -KAB_F \\ -KA^2B_F \\ \vdots \\ -KA^{H_u-2}B_F \end{bmatrix}, Y_{DUF} = \begin{bmatrix} 0 \\ -KB_2 \\ -KAB_2 \\ -KA^2B_2 \\ \vdots \\ -KA^{H_u-2}B_2 \end{bmatrix}, \quad Y = \begin{bmatrix} -I \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

$$\Lambda = \begin{bmatrix} \mathbf{I} & 0 & 0 & 0 & \cdots & 0 \\ -[KB_U + \mathbf{I}] & I & 0 & 0 & \cdots & 0 \\ -\mathrm{K}[A - \mathbf{I}]B_U & -[KB_U + \mathbf{I}] & \mathbf{I} & 0 & \cdots & 0 \\ -KA[A-I]B_U & -K[A-\mathbf{I}]B_U & -[KB_U+\mathbf{I}] & \mathbf{I} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -KA^{H_u-3}[A-I]B_U & -KA^{H_u-4}[A-I]B_U & -KA^{H_u-5}[A-I]B_U & -KA^{H_u-6}[A-I]B_U & \cdots & \mathbf{I} \end{bmatrix}.$$

Set point/tracking variables and zone region variables are defined as $y_{sp} = C_{sp}\,x$ and $y_{zr} = C_{zr}\,x$, respectively. Slack variables are defined as, $y_{zr} > y_H \Rightarrow \varepsilon = y_{zr} - y_H$ and $y_{zr} < y_L \Rightarrow \varepsilon = y_L - y_{zr}$. The cost function in Eq. (2) then becomes

$$V(k) = [\mathbf{C}_{SP}X(k) - \mathbf{T}_{SP}(\mathrm{k})]^T\mathbf{Q}_{SP}[\mathbf{C}_{SP}X(\mathrm{k}) - \mathbf{T}_{SP}(\mathrm{k})] + U_U(\mathrm{k})^T\mathbf{R}U_U(\mathrm{k}) + ||\boldsymbol{E}(\mathrm{k})||,$$

where $\mathbf{C}_{SP}$ and $\boldsymbol{E}(k)$ are defined as

$$\mathbf{C}_{\mathrm{SP}} = \mathrm{diag}[\,C_{sp_1} \quad C_{sp_2} \quad \cdots \quad C_{sp_{Hp}}\,],$$
$$\boldsymbol{E}(k) = [\hat{\varepsilon}(k+1|k)^T \quad \hat{\varepsilon}(k+2|k)^T \quad \cdots \quad \hat{\varepsilon}(k+H_p|k)^T]^T,$$

$\mathbf{C}_{ZR}$, $\mathbf{Q}_{SP}$ and $\mathbf{R}$ are defined in similar manner with $\boldsymbol{C}_{zr}$, $\boldsymbol{Q}_{sp}$ and $\boldsymbol{R}$. The modified MPC optimization problem then becomes

$$\begin{aligned} \min_{\mathbf{Q}(k)} V(k) &= \mathbf{Q}(k)^T[\Theta^T\mathbf{C}_{SP}^T\mathbf{Q}_{SP}C_{SP}\Theta + \Phi^T\mathbf{R}\Phi]\mathbf{Q}(k) \\ &\quad + 2\mathbf{Q}(k)^T[\Phi^T\mathbf{N}(k) - \Theta^T\mathbf{C}_{SP}^T\mathbf{Q}_{SP}\varepsilon_{sp}(k)] \\ &\quad + \mathbf{N}(k)^T\mathbf{R}\mathbf{N}(k) + \varepsilon_{sp}(k)^T\mathbf{Q}_{SP}\varepsilon_{sp}(k) \\ \text{where } \mathbf{N}(k) &= Y_X\boldsymbol{x}(k) + Y_R\boldsymbol{R}_{sp}(k) + Y_{UF}\boldsymbol{u}_F(k) \\ \text{and } \varepsilon_{sp}(k) &= \mathbf{T}_{SP}(k) - \mathbf{C}_{SP}[\Psi\boldsymbol{x}(k) + \Pi\mathbf{R}_{sp}(k) + Y_F\boldsymbol{u}_F(k)]. \end{aligned} \tag{4d}$$

The above cost function needs to be minimized subject to the constraints in Eq. (2). The constraints can be expanded over the horizon and expressed in compact form as

$$[\mathbf{P} \quad \mathrm{s}]\begin{bmatrix} \Delta U_U(k) \\ 1 \end{bmatrix} \leqslant \mathbf{0}, \quad [\mathbf{F} \quad \mathrm{f}]\begin{bmatrix} U_U(k) \\ 1 \end{bmatrix} \leqslant 0 \quad \text{and} \quad [\Gamma \quad \mathrm{g}]\begin{bmatrix} X(k) \\ 1 \end{bmatrix} \leqslant 0, \tag{4e}$$

where the matrices $\mathbf{P}$, s, $\mathbf{F}$, f, Γ and g are derived from the corresponding constraint boundaries expanded over the horizons (see Maciejowski (2002) for details). Expressing these in terms of $\mathbf{Q}(k)$, the constraint equations become

$$\begin{bmatrix} \mathbf{P}\Lambda \\ \mathbf{F}\Phi \\ \Gamma\Theta \\ \mathbf{C}_{ZR}\Theta \\ -\mathbf{C}_{ZR}\Theta \end{bmatrix}\mathbf{Q}(k) \leqslant \begin{bmatrix} -\mathbf{P}[\Xi\boldsymbol{x}(k) + Y_{DR}\boldsymbol{R}_{sp}(k) + Y_{DUF}\boldsymbol{u}_F(k) + Y\boldsymbol{u}_U(k-1)] - \mathrm{s} \\ -\mathbf{F}[Y_X\boldsymbol{x}(k) + Y_R\boldsymbol{R}_{sp}(k) + Y_{UF}\boldsymbol{u}_F(k)] - \mathrm{f} \\ -\Gamma[\Psi\boldsymbol{x}(k) + Y_F\boldsymbol{u}_F(k) + \Pi\boldsymbol{R}_{sp}(k)] - \mathrm{g} \\ -\varepsilon_H(k) \\ -\varepsilon_L(k) \end{bmatrix}, \tag{4f}$$

where $\varepsilon_H(k)$ and $\varepsilon_L(k)$ are predicted zone error violations defined similar to $\varepsilon_{sp}(k)$. The first two lines in Eq. (4f) represent hard actuator rate and position limits. The third line represents state constraints. These constraints are

softened and, as described in Kerrigan and Maciejowski (2000), an l_1 or l_∞ norm-based constraint violations must be used to make penalty functions exact with finite values of penalty weights.

5.6. Target recalculations

For most constrained systems, perfect tracking is usually unachievable in spite of sufficient degrees of freedom in underlying dynamic equations. This happens because, the input and output constraints do not allow the system to reach all output target values exactly. Such unachievable targets must be adjusted with target recalculations at every time step before the MPC optimization (Muske, 1995). The target recalculations also offer a safety check for the commands applied to the MPC algorithm. By recalculating targets (based on a small prior optimization), the likelihood of MPC demanding excessive control input saturations is reduced when large input commands are to be followed. Such a scenario can exist in the case of multiple control input failures as the remaining control inputs would be required to replace the authority of those parts of the system that are no longer functioning. Preventing control saturations in the case of component/system failures can assist in improving the stability and robustness of the overall system as shown in Section 5. Consider a pre-stabilized system

$$\boldsymbol{x}(k+1) = \boldsymbol{A}_K\boldsymbol{x}(k) + \boldsymbol{B}_u\boldsymbol{q}(k) + \boldsymbol{B}_F\boldsymbol{u}_F, \tag{5a}$$

state constraints and input constraints are rewritten as

$$[\,G \quad J\,]\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{q}\end{bmatrix} \leqslant -g \text{ and } \mathrm{F}\boldsymbol{x} \leqslant -\mathrm{f}. \tag{5b}$$

Given an existing command, $\boldsymbol{y}_{sp}$, new target states, $\boldsymbol{x}_t$, and inputs, $\boldsymbol{q}_t$ are calculated by solving a small QP as

$$\min_{\boldsymbol{x}_t,\boldsymbol{q}_t}[\,\boldsymbol{x}_t^{\mathrm{T}} \quad \boldsymbol{q}_t^{\mathrm{T}}\,]\begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{R}_t\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_t\\ \boldsymbol{q}_t\end{bmatrix}, \tag{5c}$$

$$\text{subject to : } \begin{bmatrix}\mathbf{I} - \boldsymbol{A} + \boldsymbol{B}_u\boldsymbol{K}_u & -\boldsymbol{B}_u\\ \boldsymbol{C}_{sp} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_t\\ \boldsymbol{q}_t\end{bmatrix} = \begin{bmatrix}\boldsymbol{B}_F\boldsymbol{u}_F\\ \boldsymbol{y}_{sp}\end{bmatrix} \text{ and } \begin{bmatrix}\boldsymbol{C}_{zr} & \mathbf{0}\\ -\boldsymbol{C}_{zr} & \mathbf{0}\\ \mathrm{F} & \mathbf{0}\\ \mathrm{G} & \boldsymbol{J}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_t\\ \boldsymbol{q}_t\end{bmatrix} \leqslant \begin{bmatrix}\boldsymbol{y}_H\\ \boldsymbol{y}_L\\ -\mathrm{f}\\ -\mathrm{g}\end{bmatrix}. \tag{5d}$$

The equality constraints in Eq. (5d) represent given commands, $\boldsymbol{y}_{sp}$, and the condition for an equilibrium point at new target values. If it is not possible to satisfy the equality constraints, a suitable quadratic penalty for the error is also included in the cost function (5c). A simple method of achieving this is to convert the equality constraints into inequality constraints. A small tolerance band is created around the right-hand side of the equality constraints and these inequalities are then treated as soft constraints. The zone region constraints are assigned with least penalty weights for violations. The target value constraint in (5d) is given the highest priority, i.e. the penalty weights for its violations are the largest among all softened constraints. In these target recalculations, the state and input limit constraints are not softened.

6. Experimental results

As mentioned in Section 4.1, to meet the tracking performance specifications and rejection of unmodeled disturbances, integral states for the tracking errors in body rates (p, q and r) are introduced. It is important to note the use of integral error states for body rates, rather than for AoA and sideslip angle, is more meaningful. This is necessary for the controller to automatically find new trim values that can stabilize the aircraft along a steady flight path under fault conditions. For a damaged aircraft, small finite errors in AoA and sideslip might be tolerable but to reduce the pilot load, it is more important to make body rates zero under steady flight conditions. Also, the non-minimum phase responses associated with AoA and sideslip can lead to slow oscillations of control surfaces if integral error states for these variables are used in the control optimization. This is indeed the case, as discovered during the initial simulation studies (due to lack of space not reported here). A further consideration is the separation of velocity from other state variables. This is logical as airspeed response times and dynamics are considerably slower when compared to variables such as p, q and r. Hence, the optimization problem solved online by the MPC without velocity is expected to be better conditioned and easier to solve. This also addresses concerns such as the MPC controller sometimes achieving a step decrease command in velocity by decreasing throttle and deflecting some control surfaces (i.e. using drag to reduce

Table 3
Fault cases used in simulation studies

Case	Pilot commands (time) vs. (deg/s)	Faults (occurs at 3 s)	Uncertainty	Atmospheric turbulence (speed) and (scale length)
1	p 11–14–17 s: +25 to 0 to −25	Rudder: stuck at 10° and,	10%/0.5 or 15%/0.9	No turbulence OR
	q 6–10–13 s: +10 to 0 to −7	ROE: stuck at 0°		Intensity 3 m/s and 525 m
2	p 11–14–17 s: +5 to 0 to −5 q 6–10–14 s: +10 to 0 to −10	LOE: 50% loss of area, LIE: stuck at 2°	30%/0.9	Intensity 3 m/s and 525 m Triangular wind gust: Y axis: 7–10–13 s: 20m/s peak Z axis: 11–15–19 s: 20m/s peak
3	p 11–14–17 s: +25 to 0 to −25	All canards and elevons:	10%/0.5	Intensity 3 m/s and 525 m OR
	q 6–10–13 s: +20 to 0 to −15	40% loss of area		Intensity 6 m/s and 525 m

velocity which is clearly undesirable). Initially, airspeed was included as one of the states used by the MPC. To reduce the likelihood of problems with this formulation a separate auto-throttle was implemented for velocity control.

The actuator position limits are shown in Table 2 and are obtained based on 60% control authority. Hence, the controller limits on actuator deflections are smaller than envelop limits. The controller limits on AoA and sideslip are treated as zone regions. To accommodate structural considerations, the controller is also expected to keep body rates within the limits shown in Table 2. The actual hard limits may be greater; hence all of the state limits are softened by the MPC algorithm.

Simulation results for pitch rate (q_c) and velocity vector (wind-axis) roll rate (p_{wc}) commands are presented. A sampling time of 50 ms is employed with a prediction horizon of 15 and a control horizon of nine sample time steps. The state weights are $\boldsymbol{Q}$=[1 1 25 450 25] for AoA, sideslip, roll rate, pitch rate and yaw rate respectively and control weights are $\boldsymbol{R}$=[80 40 10 20] for canards, elevons, rudder and thrust vectoring, respectively. Integral error weights for the body rates are $\boldsymbol{Q}_i$=[0.1 1.0 0.1]. These values were obtained after few initial trials runs. However, some guidelines were also taken into consideration during the selection of the weights. The weights nominally indicate the importance associated with corresponding variables. To begin with, pitch rate was assigned the highest weight since the aircraft dynamics are unstable in longitudinal axis. Similarly, AoA and sideslip angle weights indicate a relatively low importance associated with the actual value of these variables due to corresponding zone region formulations. The weights assigned to control inputs were adjusted to keep the utilization of all the aero-surfaces nearly equal under nominal conditions. The values of $\boldsymbol{Q}$, $\boldsymbol{R}$ and $\boldsymbol{Q}_i$ remained unchanged thereafter and no retuning of these weights is required for the various fault cases presented. Loss of effective area of the aircraft control surfaces is considered and is simulated by multiplying the corresponding columns of the $\boldsymbol{B}$ matrix (control derivatives) by a constant between 0 and 1. Stuck or runaway actuator faults are also introduced. The uncertainty and identification errors in the $\boldsymbol{A}$ matrix (stability derivatives) due to these faults are simulated by changing all the elements of $\boldsymbol{A}$ matrix at random with a probability. Similarly, all the elements of the changed $\boldsymbol{B}$ matrix are also perturbed at random with a probability. The nominal values of such uncertainty are 10% perturbations with a probability of 0.5. MPC uses these $\boldsymbol{A}$ and $\boldsymbol{B}$ matrices as an internal model of the damaged aircraft (assumed to be provided by a system identification module). Such parametric uncertainty is an unavoidable characteristic of online system identification algorithms and the reconfigurable controller must be tolerant to these errors. The dynamic models of the aircraft are updated by the Model Change Logic block based on Mach number, AoA, altitude and faults at intervals of 0.5 s. This is equivalent to a practical scenario where FDI and system identification modules are operating at a sampling rate of 10 times that of the MPC controller. In effect, a FDI delay of 0.5 s is always present when the faults occur. Table 3 shows the fault cases implemented for simulation studies.

6.1. Improvements with soft constraints and zone regions

Fig. 4 shows aircraft states and inputs in response to roll rate and pitch rate commands with simulation settings of Case 1 in Table 3. Initially, simulations without atmospheric turbulence and with 10% model uncertainty are discussed. Besides indicating reconfigurable control capabilities of MPC, a comparison is made between hard constrained MPC and MPC with zone regions and soft constraints. The fault considered introduces considerable asymmetric aerodynamic forces and moments. Since the stuck rudder at 10° position produces a large yawing moment, sideslip angle deviates from the desired value of zero. When these deviations reach the zone limit of 5°, the hard

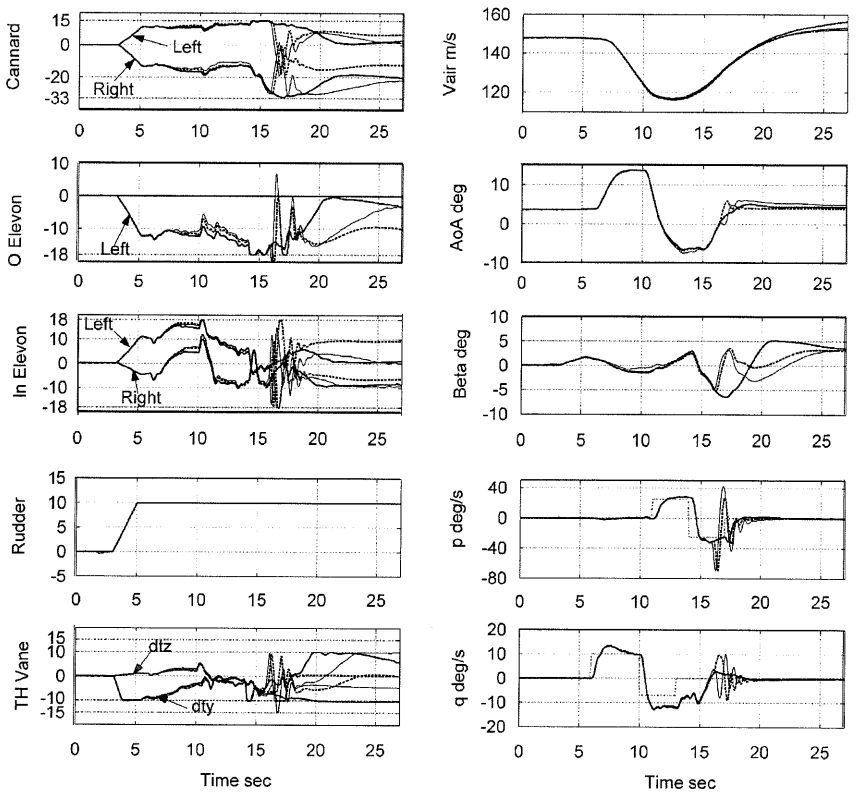

Fig. 4. Aircraft responses to case 1 setting in Table 3: (- - -) hard constrained MPC, (—) soft constrained MPC, (—) hard constrained MPC with disturbances and increased model uncertainty.

constrained MPC acts aggressively producing large movements of control surfaces to move sideslip away from the limits. Such aggressive control action leads to rapid saturation of elevon positions. This also results in considerable oscillations in roll rate and yaw rate. In particular, the peaks in roll rate and pitch rate responses are highly undesirable and can lead to acceleration (g) limit violations. Command tracking performance is also severely affected. On the other hand, MPC with zone regions and soft constraints gives a much better response to the commands. The actuator deflections and state trajectories are smooth indicating at least Level II (or Level I in pitch axis) handling qualities.

The soft MPC does not cause oscillations when zone limits on sideslip are approached and a slight violation of $-5°$ zone limit is allowed. Such violation may be unavoidable in presence of the major runaway fault considered in rudder.

In terms of control reconfiguration capabilities, the cross couplings in aircraft dynamics due to the jammed rudder position at 10° are compensated by asymmetric deflections of canards, elevons and thrust vectoring vanes. Effectively, by means of optimization, MPC exploits the inherent redundancy within control inputs. The horizontal thrust deflection acts to counteract the stuck rudder while residual rolling moments induced by the rudder are counteracted by suitable differential deflections of elevons. A careful observation of the figure also reveals that input and output deflections are approaching to steady values indicating the stable operation of MPC controller and discovery of new trim values for new steady flight conditions. The loss of velocity during the manoeuvre is gradually recovered by a separate auto-throttle controller.

Further, Fig. 4 also shows the simulation results with atmospheric turbulence and increased model uncertainty. The turbulence and uncertainty settings are shown in Table 3, Case 1. For soft MPC, these changes have no effect on the responses. But the hard MPC results in increased amplitude oscillations of control inputs as well as in tracked outputs. Clearly, soft MPC is more robust that hard constrained MPC and hence more suitable for fault tolerant reconfigurable control.

6.2. *Improvements with pre-stabilization*

Fig. 5 shows the results for the case 2 scenario in Table 3. This case considers a combination of roll rate and pitch rate commands, asymmetric faults, high model uncertainty and atmospheric turbulences along with large wind gusts.

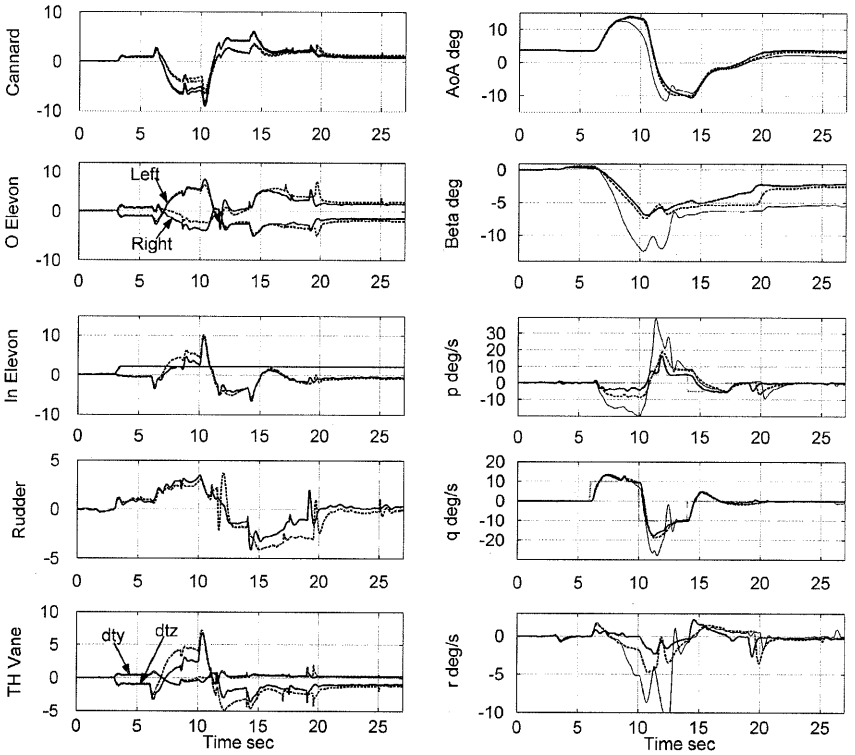

Fig. 5. Aircraft responses to case 2 setting in Table 3: (---) MPC without pre-stabilization, (—) MPC with stabilization, (—) MPC without pre-stabilization and increased disturbances.

The constraint softening and zone regions are also incorporated in MPC algorithms during this simulation set-up. The lateral and vertical axis wind gusts act as disturbances on the system. Fig. 5 indicates, the MPC without pre-stabilization has a poor disturbance rejection capability as the aircraft roll rate response shows a considerable effect of the lateral axis wind gust starting at 7 s. Similarly, the yaw rate response shows an overshoot when the wind gust reaches its peak at around 10 s. On the other hand, pre-stabilized MPC roll rate and yaw rate responses show a much reduced coupling to the lateral wind gust. Also, the sideslip angle in case of pre-stabilized MPC response returns quickly inside zone region of $-5°$ and approaches a steady value faster as compared to the response of MPC without pre-stabilization.

Fig. 6 shows responses of other aircraft variables during the manoeuvre. The advantage of pre-stabilization is more prominent in these plots. The pitch angle (theta) approaches to steady small positive value, indicating return to a level flight condition after the end of manoeuvre. Similarly, the airspeed also approaches a steady value. However, the MPC without pre-stabilization results in an unsteady flight response after the end of manoeuvre as the aircraft pitches down thereafter. In practice, this means an increased pilot load; since he is required to use more efforts to maintain aircraft in a steady flight. Figs. 5 and 6 also show aircraft responses when the lateral (Y) axis wind gust has an increased peak magnitude of 15 instead of 5 m/s.

In this situation, the pre-stabilized MPC produced responses slightly different than the ones with a gust magnitude of 5 m/s, hence are not shown. On the other hand there is a severe deterioration of responses produced by MPC without pre-stabilization. The sideslip angle and aircraft body rates show considerable overshoots and oscillations. Clearly, the handling qualities are below Level II in this case. A prominent effect of the increased wind gust magnitude is visible in Fig. 6. Without pre-stabilization in MPC, the aircraft is loosing altitude rapidly as it is now pitched down to a steady angle of $-20°$. However, with pre-stabilization, the overall aircraft responses to commands are smoother with at least Level I handling qualities in pitch axis and Level II in lateral axis effectively maintained throughout the simulation run.

6.3. *Effect of target recalculations*

Finally, results for case 3 are shown in Figs. 7 and 8. The constraint softening, zone regions and pre-stabilizations are also incorporated during this simulation set-up. Target recalculations mainly play an important role when the system is pushed to the limits of available control power and allowable state boundaries. The ADMIRE aircraft model

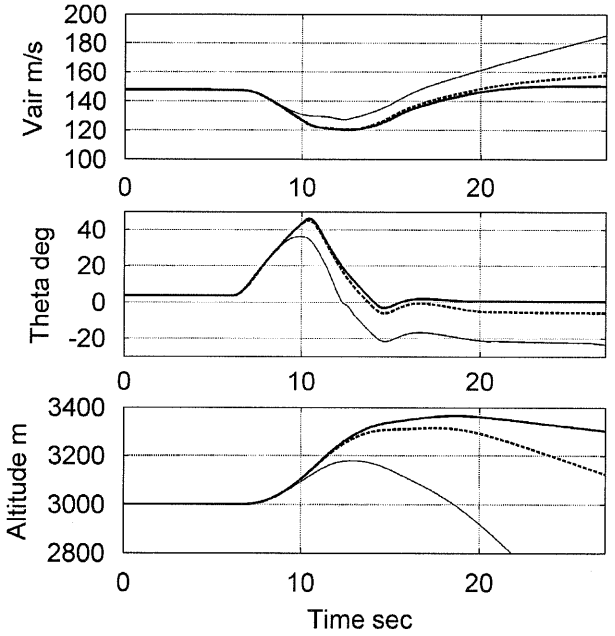

Fig. 6. Aircraft responses to case 2 settings in Table 3.

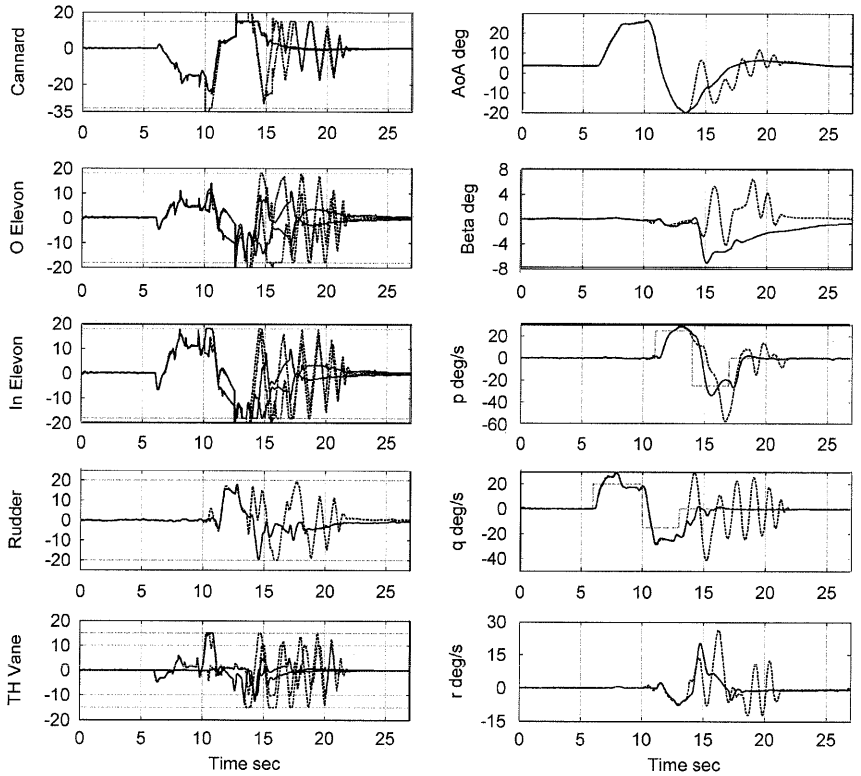

Fig. 7. Aircraft responses to case 3 setting in Table 3: (- - -) MPC without target recalculations, (—) MPC with target recalculations.

has a total of 9 control inputs and, hence the available control power is sufficient for manoeuvring under nominal conditions. As a result, initially the advantages of target recalculations remained undetected when moderate levels of faults and pilot commands were considered. To reveal the benefits of target recalculations, some extreme combinations of failures and pilot commands were evaluated. However, care has been taken to maintain realistic fault situations. In case

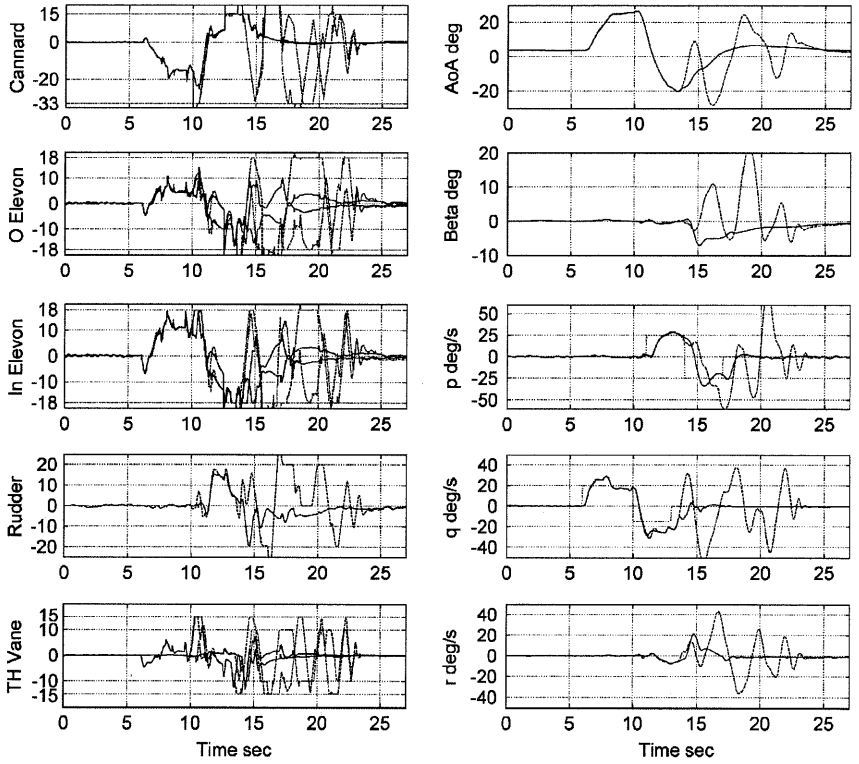

Fig. 8. Aircraft responses to case 3 setting and increased disturbances.

3, both canards and all elevons have 40% reduction in their effectiveness. Such failures even though seemingly extreme, are sometimes possible to take place in real life scenarios. A typical example is loss of pressure in the hydraulic actuation system of the aircraft. This kind of fault occurs due to leakage or servo valve malfunctioning. The sufficiently high-amplitude commands for roll rate and yaw rate result in control surface movements close to the saturation limits. As the faults in this case are symmetric, the left and right surface movements of canards and elevons overlap in Fig. 7. As Fig. 7 shows, the response of MPC without target recalculations is clearly oscillatory and unsatisfactory. To achieve high-amplitude commands, MPC without target recalculations demands large deflections of control surfaces. This causes frequent saturations of the available control inputs as the system under this fault conditions is inherently incapable of attaining such command values with available control power. The handling qualities achieved are well below Level II. However, with target recalculations, tracked roll rate and pitch rate commands are systematically lowered by solving a small optimization at every time step prior to the MPC action. Hence, the control demands are smaller and saturation peaks are less frequent and of smaller durations when compared to MPC without target recalculations.

The handling qualities achieved are at least Level II and sometimes Level I. Essentially, the target recalculations prevent the MPC from over-driving the actuators. If the pilot commands are too large for the available control power, new less demanding commands are generated. In the case of fault tolerant control design, such a feature is desirable as control saturations in effect mean open-loop operation of the system when there is greater uncertainty present due to the failures.

Fig. 8 shows the responses for the same simulation set-up with increased atmospheric turbulence of 6 m/s. The MPC without target recalculations clearly leads to unacceptable oscillations in the control loop. The actuators saturate more frequently and for longer durations. Since during such saturations the system is partially open-loop, increased disturbances can easily drive the system out of desired operational limits. To counteract these violations, MPC acts aggressively leading to further oscillations of control inputs and aircraft states. On the contrary, the MPC with target recalculations produced similar responses when the turbulence intensity of 3 m/s was used. This proves the robustness to disturbances even in the presence of extreme faults and high-amplitude manoeuvres.

6.4. Computational issues

For the simulation studies in this paper, a sampling time of 50 ms is assumed. Though no hardware-in-loop simulations are performed, some important conclusions can be drawn based on the time required by the MPC block

within Simulink framework. The simulation studies are carried out on a 1.2 GHz PC. On average, the time spent in MPC calculations was in the range of 40–55 ms. The MPC computations and the QP code are implemented in the form of a non-optimized Matlab code. Clearly, an optimized and carefully written C-code can perform much better and hence it is possible to implement the algorithms presented in a real time environment.

Further, the QP implemented in the presented formulations of MPC can be solved using active set based or IP-based optimization algorithms. Although a detailed analysis is not yet available, it is worth noting that an IP-based method offers a significant benefit over the active set method in terms of computation time. This effect is more pronounced when multiple failures are considered and, due to large commands, available actuators frequently saturate. When such saturation occurs, the active set method takes more time to find the next feasible set and corresponding control inputs. On the other hand, the IP method is less sensitive to such effects.

7. Conclusions

A number of formulations for MPC from a reconfigurable control perspective have been presented and considered. These formulations have been gradually refined in terms of their robustness and stability for performance on a flight control system. Advantages of using soft constraints, zone regions, pre-stabilization and target recalculation in MPC-based RFCS design have been demonstrated with results presented from a non-linear 6 DOF aircraft simulation. The simulation studies show the proposed control formulations performing satisfactorily in the presence of multiple control input failures as well as severe atmospheric disturbances. Although a significant delay in FDI is introduced along with the uncertainty in system models used by the controller, the MPC formulations show not only robust and stable operation but also the conformance to appropriate handling quality models for pilot commands. The time to recover aircraft control in a satisfactory manner after the occurrence of failures is short; indicating fast and efficient adaptive characteristics of the MPC algorithms for reconfigurable flight control. As no online tuning or learning of controller parameters is involved, the time required to adapt to faults is, in effect, governed by the FDI delay. The simulation studies also reveal that the MPC framework can systematically treat and respect limits on the actuators and states whenever possible. However, there is scope to further improve the RFCS methodology presented if the uncertainty in online estimated parameters can be explicitly considered in the MPC optimization. Further research effort will be devoted to such robust MPC formulations.

Acknowledgements

The first author gratefully acknowledges the financial support for this research by the Faculty of Engineering and Applied Sciences, University of Southampton, England.

References

Backstrom, H. (1997). *Report on the usage of the generic aerodata model.* Linkoping: SAAB Aircraft AB.

Bartlett, R. A., Wachter, A., & Beigler, L. T. (2000). Active set vs interior point strategies for model predictive control. *Proceedings of the American control conference*, pp. 4229–4233.

Bemporad, A. (1998). Reducing conservativeness in predictive control of constrained systems with disturbances. *Proceedings of the 37th IEEE conference on decision and control*, pp. 1384–1389.

Bodson, M. (1993). Identification with modelling uncertainty and reconfigurable Control. *Proceedings of the 32nd IEEE conference on decision and control*, pp. 2242–2247.

Bodson, M. (1997). Multivariable adaptive algorithms for reconfigurable flight control. *IEEE Transactions on Control System Technology*, *5*(2), 217–229.

Burken, J. J., Lu, P., Wu, Z., & Bahn, C. (2001). Two reconfigurable flight-control design methods: robust servomechanism and control allocation. *Journal of Guidance, Control and Dynamics*, *24*(3), 482–493.

Eberhardt, R. L., & Ward, D. G. (1999). Indirect adaptive flight control system interactions. *International Journal of Robust and Nonlinear Control*, *9*, 1013–1031.

Etkin, B., & Reid, L. D. (1996). *Dynamics of flight: Stability and Control* (3rd ed.). New York: Wiley.

Huzmezan, M., & Maciejowski, J. M. (1998). Reconfigurable flight control of a high incidence research model using predictive control. *UKACC international conference on control*, pp. 1169–1174.

Keerthi, S. S., & Gilbert, E. G. (1988). Optimal infinite horizon feedback laws for a general class of constrained systems: stability and moving horizon approximations. *Journal of Optimisation Theory and Applications*, *57*, 265–293.

Kerrigan, E. C., & Maciejowski, J. M. (2000). Soft constraints and exact penalty functions in model predictive control. *Proceedings of the UKACC international conference*, Cambridge, UK, September 2000.

Kothare, M. V., Balakrishnan, V., & Morari, M. (1996). Robust constrained model predictive control using linear matrix inequalities. *Automatica, 32*(10), 1361–1379.

Lee, J. W. (2000). Exponential stability of constrained receding horizon control with terminal ellipsoid constraints. *IEEE Transactions on Automated Control, 45*(1), 83–88.

Maciejowski, J. M. (2002). *Predictive control.* Englewood Cliffs, NJ: Prentice-Hall.

Markerink, J., Bennani, S., & Mulder, B. (1997). Design of a robust scheduled controller for the HIRM using μ-synthesis. *GARTEUR technical report*, TP-088-29.

Maybeck, D. Q., & Stevens, R. D. (1991). Reconfigurable flight control via multiple model adaptive control method. *IEEE Transactions on Aerospace and Electronic Systems, 27*, 470–479.

McLean, D. (1990). *Automatic flight control systems.* Englewood Cliffs, NJ: Prentice Hall International (UK).

Monaco, J., Ward, D., & Bird, R. (1997). Implementation and flight test assessment of an adaptive, reconfigurable flight control system. *Proceedings of the AIAA GNC*, pp. 1443–1454.

Muske, K. R. (1995). *Linear model predictive control of chemical processes.* Ph.D. Thesis, University of Texas at Austin.

Patcher, M., Chandler, P. R., & Mears, M. (1995). Reconfigurable tracking control with saturation. *Journal of Guidance, Control and Dynamics, 18*(5), 1016–1022.

Patcher, M., & Miller, R. B. (1997). Manoeuvring flight control with actuator constraints. *Journal of Guidance, Control and Dynamics, 20*(4), 729–734.

Pratt, R. W. (1999). *Flight control systems: practical issues in design and implementation.* Institution of Electrical Engineers.

Rao, C. V., Wright, S. J., & Rawlings, J. B. (1997). Application of interior-point methods to model predictive control. *Journal of Optimization Theory and Applications, 99*, 723–757.

Rawling, J. B., & Muske, K. R. (1993). The stability of constrained receding horizon control. *IEEE Transactions on Automated Control, 38*(10), 1512–1516.

Rossiter, J. A., Kouvaritakis, B., & Rice, M. J. (1998). A numerically robust state-space approach to stable predictive control strategies. *Automatica, 34*(1), 65–73.

Ward, D. G., Monaco, J. F., & Bodson, M. (1998). Development and flight testing of a parameter identification algorithm for reconfigurable control. *Journal of Guidance, Control and Dynamics, 21*(6), 956–984.

Wang, Y. (2002). *Robust model predictive control.* Ph.D. Thesis, University of Wisconsin-Madison.