A Novel Strategy for Fault Tolerant Control

Mihai Huzmezan [*] Jan Maciejowski
Department of Engineering, University of Cambridge,
Cambridge CB2 1PZ, United Kingdom.

Abstract

This paper describes advances towards reconfiguration and scheduling in flight control systems using high fidelity models expressed in a quasi-LPV form, Fault Detection and Isolation (FDI), model approximation/simplification and constrained Model Based Predictive Control ($MBPC$). The strategy is applied to a well known missile example. This represents a new Reconfigurable Flight Control System (RFCS). The paper also contains the exploration of the reconfiguration strategy by employing it to a High Incidence Research Model (HIRM).

1 The $MBPC$ Reconfigurable Flight Control

Control reconfiguration is a critical technology. All methods presume a certain initial knowledge of the system. In time, this may change gradually due to environmental effects or rapidly on account of faults. Hence, various impairments may significantly degrade the performance of the aircraft as well as the decoupling of the aircraft longitudinal and lateral axes. A reconfigurable controller possesses the capability to redistribute and coordinate the control effort during a system failure among the aircraft's remaining effective control surfaces, such that satisfactory flight performance is retained if possible. As the system changes the necessity for updating the model and the control strategy increases. The flight control system is designed not only to cope with failures but to be robust with respect to variations in aerodynamic stability derivatives, high frequency unmodelled dynamics, time delays, sensor noise and atmospheric turbulence. If the aircraft dynamics is altered due to uncontrollable changes such as failed sensors, actuators or damaged structural elements the reconfigurable control should ensure a response to pilot's command inputs still close to the nominal performance if possible.

Reconfigurable control tries to avoid excessive hardware replication. Of course, the set of redundant components allows the flight control system to continuously perform its functions in spite of the failures some of its components might have. The trade off in terms of replication is between the weight, size, complexity and power consumption of such redundant sets of instruments. Instead of relying on hardware redundancy a reconfigurable control system has to exploit inherent control redundancies in order to

[*] e-mail: mh2,jmm@eng.cam.ac.uk

restructure in real time and preserve stability and tracking. Increased survivability and reduced support requirements involve redundant aerodynamic control surfaces and spare power during manoeuvres.

As a validation scheme the adaptive control reconfiguration concepts use the full nonlinear simulation of the aircraft, the main performance goal being to land the aircraft safely whenever possible. Owing to the time constraints, in many failure scenarios the automated control law redesign process has to make use of the most numerically efficient algorithms possible. When an impairment occurs the most urgent task is to stabilise the system so as to avoid catastrophic situations. Once the stabilisation has been carried out, there is more time for the control reconfiguration mechanism to manipulate the control in order to obtain better system performance.

The main contribution of this paper is the integration of:

- Fault Detection and Isolation (FDI)
- High fidelity models
- Model approximation and simplifications techniques
- Constrained Model Based Predictive Control ($MBPC$)

in a framework which provides a solution for scheduling and reconfiguration in response to failures.

We approach reconfigurable control as a multi-model adaptive control problem, namely one in which adaptation occurs by discrete changes to the control algorithm, rather than continuous tracking of a gradually-changing model. Scheduling a means of adaptation has already been proved successful in flight control applications [Hyd95, Woo95].

The $MBPC$ control strategy can be formulated in multi-model setting. This feature together with its special ability to handle constraints [RM93, MBT97] and its inherent self-reconfiguration property [Mac97] led us to consider it as a potential tool for reconfigurable control.

We propose to represent the controlled plant by a high-fidelity nonlinear model. We therefore do not use an identification module, at least for ordinary 'scheduling'. For reconfiguration in case of failures some fault detection module is required; for the purpose of this work we assume the existence of such a module.

The paper will firstly introduce all the ingredients of our fault tolerant strategy. Next it shows how these can be integrated to give a strategy for scheduling and reconfiguration. The solution will be illustrated via two examples.

2 A new reconfiguration strategy

The four system components shown in Figure 5 work together as follows:

1. When a failure occurs the FDI system pinpoints the nature of it. Updated parameters are passed to the high fidelity model. Actuator failure information is passed to the $MBPC$ controller as constraint modifications, if appropriate.

2. Freezing the high fidelity model with respect to the parameter vector yields a linear model which is used by the $MBPC$ as its internal model. An important advantage for real-time implementation is that this is a computationally inexpensive way of obtaining, in adaptive fashion, a linear internal model, compared with other possibilities such as identification.

3. The constrained $MBPC$ controller provides inputs to the plant. Giving it enough degrees of freedom (a large enough set of control inputs) enables it to keep the plant close to the required trajectory. We are assuming here that any failures are compatible with maintaining this trajectory.

All the components performing the functions mentioned above are shown in Figure 5. We claim that the system ensures the following successive layers of reconfiguration:

- A simple reconfiguration occurs if there are actuator failures which are consistent with maintenance of the required flight conditions. This can occur even without FDI information [Mac97].
- Small structural failures are reflected in the high-fidelity model, as shown in Section 4, and hence in the linear internal model used by the $MBPC$ controller. No tuning is necessary for the controller; its original cost function still reflects well the various criteria for which it was designed.
- Major failures will again lead to a change in the linear internal model, but now retuning of the controller will be necessary. The approach which keeps the horizons fixed but adjusts the weights in (1) appears to be the most promising at present. This solution is presented in Section 8.

3 Why $MBPC$ for reconfigurable control?

Among the possible applications of $MBPC$ for flight control we can enumerate reconfigurable flight control, a very challenging problem that can emphasise several features of predictive controllers and push the scheme to its limits.

Before addressing various issues connected with constrained predictive control we give here some fundamentals of the method. The defining feature of Model Based Predictive Control ($MBPC$) is the repeated optimisation of a performance objective over a prediction horizon (N_2). Given a set-point, a reference $r(k+l)$ is produced and used within the cost function (1). Manipulating the control variable $u(k+l)$, over the control horizon (N_u), the algorithm drives the predicted output $\tilde{y}(k+l)$, over the prediction horizon, towards the reference.

The future control movement over the control horizon (N_u) is determined by minimising the cost function:

$$J(k) = \sum_{l=N_1}^{N_2} \|(\tilde{y}(k+l) - r(k+l)\|^2_{Q(l)} + \sum_{l=0}^{N_u-1} \|\Delta u(k+l)\|^2_{R(l)} \tag{1}$$

subject to constraints on:

- the inputs levels: $u_l(l) \leq u(l) \leq u_u(l)$
  where $k \leq l \leq k + N_u - 1$

- the input rates of change: $\Delta u_l(l) \leq \Delta u(l) \leq \Delta u_u(l)$
  where $k \leq l \leq k + N_u - 1$

- the output (and state) levels:
  $y_l(l) \leq \tilde{y}(l) \leq y_u(l)$ where $k + N_1 \leq l \leq k + N_2$

$Q(l)$ and $R(l)$ are weights independent of time k and the norm $\|.\|_Q$ is defined as $\|\alpha\|_Q^2 = \alpha^T Q \alpha$. It is assumed that $\Delta u(l) = 0$ for $l \geq k + N_u$. This optimisation is a quadratic program (QP) which is solved using a standard algorithm. Only the first command increment $\Delta u(k)$ is implemented, and the optimisation is re-solved at each step.

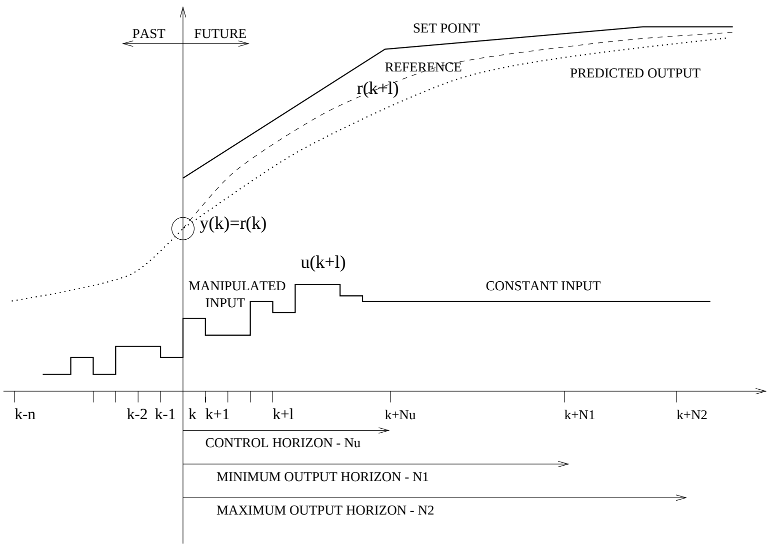

Figure 1: Prediction strategy

The main question raised while implementing the adaptive mechanism was which model should be used to provide predictions. Two strategies have been considered for use during manoeuvres which cause significant α variations:

- No *a priori* trajectory information. A single model was used across the whole prediction horizon (*ie* for $l = 1, \ldots, N_y$) but changed at each current time step (*ie* different for each k) in accordance with the measured α value.

- *A priori* trajectory information available. This allows the internal model to vary over the prediction horizon, but one needs to predict α over the prediction horizon in order to do this. It is important to base this prediction on the commanded trajectory, rather than the achieved one; this is more approximate, but it retains the QP structure of the optimisation problem. Basing it on the achieved trajectory would lose even the convexity property, so should be avoided if at all possible [CKC96]. For the missile it is possible to obtain a prediction for α, based on the commanded trajectory for n_Z.

Note that these two strategies become the same in the special case when $N_y = N_u = 1$. In our evaluations we have simulated only the first prediction strategy.

It is advisable to take one of the above suggested paths. Otherwise, obtaining the models through the plant future dynamics, the resulting cost function is no longer quadratic in the command increment. This results in the standard $MBPC$ quadratic optimisation being transformed into a nonlinear constrained optimisation for which it is hard to guarantee a global solution [CKC96].

The general structure of $MBPC$ schemes is the following:

Optimiser computes the future manipulated variable moves. Constraints reflect input (actuator) and output (flight envelope) limitations.

Predictor employs the internal model and the measurement or estimate of the current state to provide the optimiser with predicted values of states and outputs.

Internal model represents the plant. In this paper we use a state-space linear time invariant model for the plant. This is a natural choice for flight control.

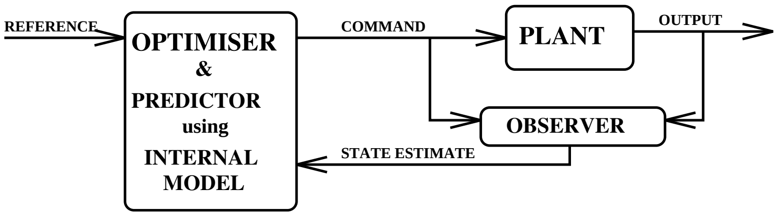

Figure 2: The structure of $MBPC$ schemes

Hence, we have decided to use predictive control as a support for our fault tolerant strategy due the following reasons:

1. the possibility of having a multi-model formulation laws

2. the simple form of the cost function which can lead to a gain type of controller possible to be parametrised in the unconstrained case with respect to the elements of output and control increment diagonal weights

3. the property of handling constraints in a straightforward manner.

For instance, to offer a better insight into the last feature mentioned and its implications in reconfigurable control we present here an example of reconfiguration performed by an MBPC flight controller, when the rudder of an aircraft jams, and it is required to change the aircraft heading (yaw angle). A linearised model of the RCAM model, as in [HM96], is used for this example. This is as well the internal model used by the MBPC controller.

There are 3 controlled variables: the yaw angle, the roll angle, and the side-slip, and 4 actuators: the rudder, the ailerons, the tail-plane, and the engine thrust. A rudder jam (at the neutral position) is simulated by disconnecting the rudder demand signal (issued by the controller) from the rudder. A step demand is then made on the yaw angle. The results are shown in Figures 3 and 4. Each of the sub-figures in these figures shows 5 cases:

Case 1 Normal operation of the rudder. The rudder and aileron positions are constrained to be within ± 2 units on the graphs, which represents $\pm 20^\circ$ in each case.

Case 2 Jammed rudder. No FDI information supplied to the controller, so that the controller is not aware of the failure. The constraints on the rudder position demand are removed, in order to see the effects of actuator constraints (known to the controller) in this scenario. (The rudder demand has been reduced by a factor of 300 in the figure for this case.) Of all the cases, this gives the slowest yaw angle response.

Case 3 Jammed rudder. No FDI information supplied to the controller, but the usual $\pm 20^\circ$ constraints are restored on the rudder position. The yaw response is significantly faster in this case, because the controller stops relying on the rudder sooner, and makes more use of the ailerons, as can be seen from the larger roll angle.

Case 4 Jammed rudder. FDI information supplied — the constraint on the rudder demand has now been tightened to $\pm 0^\circ$, so that the controller knows that it cannot move the rudder. In fact this makes so little difference that the plots for Cases 3 and 4 cannot be distinguished from each other. The reason for this is that in each case the controller moves the rudder demand to its constraint almost immediately, and then uses the ailerons and other actuators. Since the actual rudder position is the same in both cases, the two behaviours are virtually identical.

Case 5 Jammed rudder. FDI information supplied, as in Case 4. But now the weight on roll errors in the cost function has been reduced by a factor of 3, approximately. This leads to a much faster response of the yaw angle. It can be seen that a much larger roll angle develops during the first 10 seconds of the manoeuvre when this weight has been reduced, and the lift then has a larger component in the horizontal plane, which changes the aircraft's heading.

Because the manoeuvre simulated here is rather artificial we can consider a more practical requirement than changing only the yaw angle which would be to change the aircraft heading, without much concern for what combination of body angles was most appropriate to achieve it.

This example shows that such a higher level role is also appropriate for reconfiguration of a rather simple but common type of failure, since an MBPC-based flight manager would issue appropriate yaw, roll and side-slip set-points. In this case, however, the rudder jam would no longer be just an 'actuator fault', but what we earlier called an 'internal fault'. However, a complicated model update would not be needed if the actuator position were a state variable of the model, since the failure could then be represented as a state constraint.

Our approach is one that, like in the model following case, computes on-line the closed loop feedback matrix based on an update of the internal model. For this model no special restrictions apart from its controllability and observability are requested. The classic pseudo-inverse method was extended in the $MBPC$ case such that an automatic tuning procedure for the predictive controller is provided. We have imagined this procedure not only as a tool to provide off-line tuning of the controller having an internal model based on the impaired plant such as to match the performance of the nominal one, but as a potential method to be applied on-line in the event of a major failure.

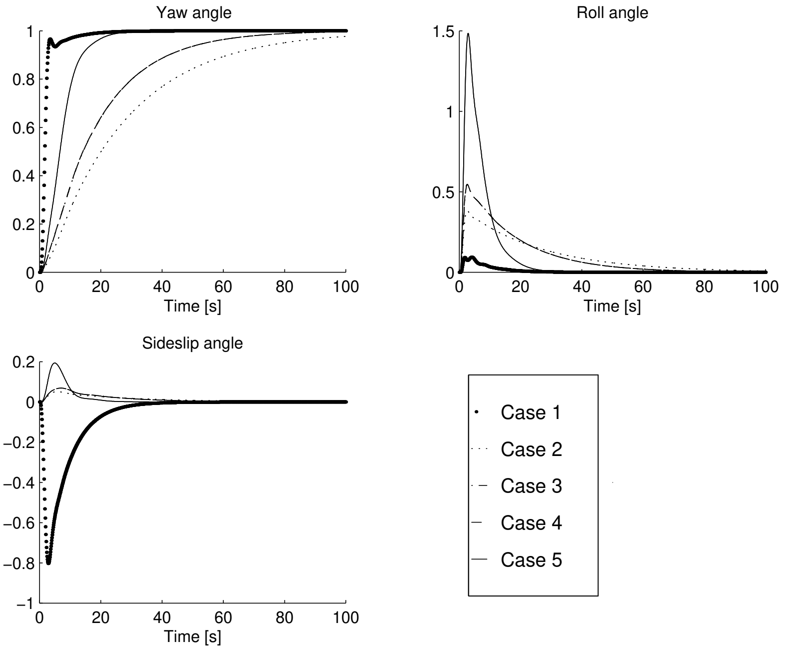

Figure 3: Yaw angle step demand with failed rudder: Controlled outputs

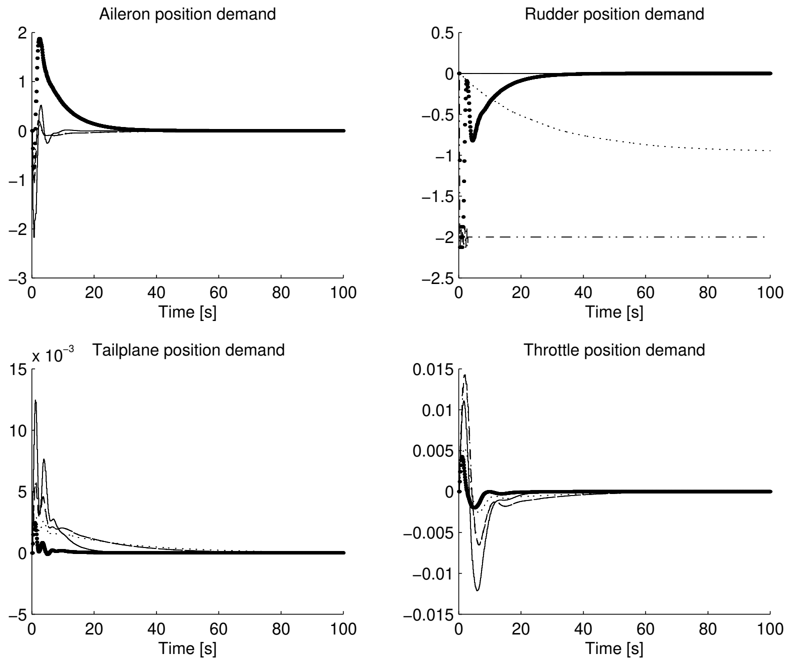

Figure 4: Yaw angle step demand with failed rudder: Actuator demands. (Key as for previous figure.)

4 Reconfiguration and scheduling using high fidelity models written in the quasi-LPV form

High fidelity dynamic models are increasingly built for complex plants. This has been the case in the aerospace industry for many years. This section will firstly introduce quasi-LPV models. Next it shows how these can be integrated to give a strategy for scheduling and reconfiguration. The solution will be illustrated via a well known missile example.

In [SC93] a quasi-LPV model that embeds the plant nonlinearities without interpolating between point-wise (Jacobian) linearisations is presented. The main characteristic of these models compared with the usual LPV way of representing systems is that the scheduling variable is a state of the model.

Such a way of modelling the plant gives the clarity required by industry and eases certification since it retains a physical meaning for the elements of the model LTV matrices. It is advisable to have the scheduling variable as a system output rather than an estimate.

The quasi-LPV approach is mostly suited for systems exhibiting output nonlinearities (*e.g.* aerospace applications). A principal requirement for a nonlinear system to be transformed into a quasi-LPV system is that the number of available equations has to be equal to the number of states plus the number of outputs minus the number of scheduling variables. When it is impossible to embed all the system nonlinearities in the output then the transformations used in producing the quasi-LPV model, see equation (4), have to be approximated up to first order terms in all the states except the scheduling parameters.

To develop the quasi-LPV model we start with a nonlinear missile model with the states α (angle of attack) and q (pitch rate), and the command δ (horizontal fin angle) of which complete details are given in [SC93].

Then we cast the nonlinear in model such a form that the nonlinearities depend only on the scheduling variable α:

$$\frac{d}{dt}\begin{bmatrix}\alpha\\ q\end{bmatrix} = f(\alpha) + \begin{bmatrix}A_{11}(\alpha) & A_{12}(\alpha)\\ A_{21}(\alpha) & A_{22}(\alpha)\end{bmatrix}\begin{bmatrix}\alpha\\ q\end{bmatrix} + \begin{bmatrix}B_{11}(\alpha)\\ B_{21}(\alpha)\end{bmatrix}\delta \tag{2}$$

A family of equilibrium states, parametrised by the scheduling variable α, is obtained by setting the state derivatives to zero:

$$0 = f(\alpha) + A(\alpha)\begin{bmatrix}\alpha\\ q_{eq}(\alpha)\end{bmatrix} + B(\alpha)\delta_{eq}(\alpha)$$

Providing that there exist continuously differentiable functions $q_{eq}(\alpha)$ and $\delta_{eq}(\alpha)$, we are able to write the system (2) in the following form [SC93]:

$$\begin{aligned}&\tfrac{d}{dt}\begin{bmatrix}\alpha\\ q - q_{eq}(\alpha)\end{bmatrix} = \\ &\begin{bmatrix}0 & A_{12}(\alpha)\\ 0 & A_{22} - \frac{d}{d\alpha}q_{eq}(\alpha)A_{12}(\alpha)\end{bmatrix}\begin{bmatrix}\alpha\\ q - q_{eq}(\alpha)\end{bmatrix} + \\ &\begin{bmatrix}B_{11}(\alpha)\\ B_{21}(\alpha) - \frac{d}{d\alpha}q_{eq}(\alpha)B_{11}(\alpha)\end{bmatrix}(\delta - \delta_{eq}(\alpha))\end{aligned} \tag{3}$$

This form gives a different α-dependent family than would be obtained by point-wise linearisation.

In order to use (3), the function $\delta_{eq}(\alpha)$ must be known. This can be estimated by using an 'inner loop' [SC93] but, because of model uncertainty, this can reduce the robustness of the main control loop in a way which is difficult to predict at the design stage. Like in [SC93] we avoid the problem generated by the existence of an inner loop required to compute $\delta_{eq}(\alpha)$ by adding an integrator at the plant input. As a result we have the following form for the system dynamics:

$$
\begin{array}{l}
\frac{d}{dt}\begin{bmatrix} \alpha \\ q - q_{eq}(\alpha) \\ \delta - \delta_{eq}(\alpha) \end{bmatrix} = \\
\begin{bmatrix} 0 & A_{12}(\alpha) & B_{11}(\alpha) \\ 0 & A_{22} - \frac{d}{d\alpha} q_{eq}(\alpha) A_{12}(\alpha) & B_{21}(\alpha) - \frac{d}{d\alpha}[q_{eq}(\alpha)] B_{11}(\alpha) \\ 0 & -\frac{d}{d\alpha}[\delta_{eq}(\alpha)] A_{12}(\alpha) & -\frac{d}{d\alpha}\delta_{eq}(\alpha) B_{11}(\alpha) \end{bmatrix} \times \\
\begin{bmatrix} \alpha \\ q - q_{eq}(\alpha) \\ \delta - \delta_{eq}(\alpha) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \nu
\end{array}
\qquad (4)
$$

This final form is actually the representation used for control purposes. If α remains constant, the $MBPC$ controller stabilising this model will drive the states to zero which means that the δ input to the plant will be set at the true δ_{eq} trim value. The $MBPC$ internal model will be the corresponding LTI system obtained at each value of the scheduling parameter α.

The controlled output of the plant is the normal acceleration (n_Z), which is a nonlinear function of α and $\delta - \delta_{eq}(\alpha)$. In the internal model a linearised approximation of n_Z with respect to α is used, as in [SC93].

The normal force (C_Z) and pitch moment (C_m) aerodynamic coefficients are embedded in the expressions for $q_{eq}(\alpha)$, $\delta_{eq}(\alpha)$, $A(\alpha)$ and $B(\alpha)$. The representation of C_Z and C_m is approximated by the following expressions:

$$
\begin{aligned}
C_Z &= \Phi_Z(\alpha) + b_Z\delta \\
C_m &= \Phi_m(\alpha) + b_m\delta
\end{aligned}
$$

where the expressions for b_Z and b_m are given in [SC93], and $\Phi_Z(\alpha)$, $\Phi_m(\alpha)$ are polynomials in α. Let k_Z and k_m, respectively, be the vectors of coefficients of these polynomials. We assume that k_Z and k_m will be the outputs of an FDI module, as shown in Figure 5. These expressions for C_Z and C_m explicitly contain contributions from all the missile's surfaces, and are based on data obtained from wind tunnel tests. Such data may be available in the form of a database [Hoa78].

5 The HIRM nonlinear model and its on-line point-wise linearisation

In this section we restrict our attention to the longitudinal channel of the High Incidence Research Model. We can assume that the side slip angle and the roll and yaw rates, are zero. The model is based on the following equations of motion:

$$
\begin{aligned}
\dot{\gamma} &= -\left(\frac{g}{V}\right) cos(\gamma) + \left(\frac{L}{mV}\right) + \left(\frac{F}{mV}\right) sin(\alpha) \\
\dot{\alpha} &= q - \dot{\gamma} \\
\dot{q} &= M\left(\frac{\bar{c}}{J_y}\right) - Z_{ATP}F \\
\dot{V} &= -gsin(\gamma) - \left(\frac{D}{m}\right) + \left(\frac{F}{m}\right) \\
\dot{H} &= V sin(\gamma)
\end{aligned}
$$

In these equations γ, α, q, V, H denote the vertical flight path angle, angle of attack, pitch rate, airspeed, and altitude, respectively. The variables m, $\bar{c}$, J_y are constants denoting, respectively, the aircraft mass, the reference mean aerodynamic chord, and pitch moment of inertia.

The expressions of the aerodynamic drag (D), lift (L) and pitch moment (M) expressed in body axis are:

$$
\begin{aligned}
D &= \bar{q}S\left[C_{X_{\delta_{TS}}}(\alpha,\delta_{TS}) + C_{X_{\delta_{CS}}}(\alpha,\delta_{TS})\delta_{CS}\right] \\
L &= \bar{q}S\Big[C_{Z_{\delta_{TS}}}(\alpha,\delta_{TS}) + C_{Z_{\delta_{CS}}}(\alpha,\delta_{TS})\delta_{CS} + \\
&\quad + C_{Z_q}(\alpha,\delta_{CS})\frac{q\bar{c}}{2V}\Big] \\
M &= \bar{q}S\bar{c}\Big[C_{M_{\delta_{TS}}}(\alpha,\delta_{TS}) + C_{M_{\delta_{CS}}}(\alpha,\delta_{TS})\delta_{CS} + \\
&\quad + C_{M_q}(\alpha,\delta_{CS})\frac{q\bar{c}}{2V}\Big]
\end{aligned}
$$

where $\bar{q} = \frac{1}{2}\rho(H)V^2$ is the dynamic pressure and $\rho(H)$ the air density which varies as a function of altitude.

In the above equations the meaning of F, δ_{TS}, δ_{CS} is thrust vector, symmetrical taileron deflection, and symmetrical canard deflection, respectively.

The non dimensional coefficients in these equations $C_{X_{\delta_{TS}}}(\alpha,\delta_{TS})$, $C_{X_{\delta_{CS}}}(\alpha,\delta_{CS})$, $C_{Z_{\delta_{TS}}}(\alpha,\delta_{TS})$, $C_{Z_{\delta_{CS}}}(\alpha,\delta_{CS})$, $C_{Z_q}(\alpha,\delta_{CS})$, $C_{M_{\delta_{TS}}}(\alpha,\delta_{TS})$, $C_{M_{\delta_{CS}}}(\alpha,\delta_{CS})$, $C_{M_q}(\alpha,\delta_{CS})$ are given in look-up tables obtained as a result of wind tunnel tests [MBT97, Wil97]. In the high fidelity model their values are computed by linearly interpolating between the values stored as functions of the variable α and the corresponding surface deflection.

The way the aerodynamics forces together with the engine thrust act upon the 6-DOF nonlinear simulation model is defined based on wind tunnel data and simple first order actuator models which include as well delays. At the moment the model is programmed in C using the Matlab Simulink S-function template and run on a SPARCstation 20 as compiled code.

We propose to obtain on-line linearisation of the system by using Taylor series to derive the analytical expressions for the linear equations. This is done around the current input and state pair, which are not, in general, equilibrium points (that is, the linearisation is not performed about a "trimmed" condition of the aircraft). The values of the derivatives of the aerodynamic coefficients are stored in look-up tables.

As an example, we show here the expression for $\Delta V = V - V_c$, where V is the total airspeed and V_c is the current airspeed value at which the linearisation is performed:

$$
\begin{aligned}
\dot{V}c + \Delta\dot{V} \quad = \quad & -gcos(\gamma_c)(\gamma - \gamma_c) + \frac{1}{m}(F - F_c) - \\
& -\frac{\bar{q}(H)S}{m}\Bigg[\Bigg(\left(\frac{\partial C_{X_{\delta_{TS}}}(\alpha, \delta_{TS})}{\partial\alpha}\right)_{(\alpha_c, \delta_{TS_c})}(\alpha - \alpha_c) + \\
& + \left(\frac{\partial C_{X_{\delta_{TS}}}(\alpha, \delta_{TS})}{\partial\delta_{TS}}\right)_{(\alpha_c, \delta_{TS_c})}(\delta_{TS} - \delta_{TS_c})\Bigg) + \\
& \Bigg(\left(\frac{\partial C_{X_{\delta_{CS}}}(\alpha, \delta_{TS})}{\partial\alpha}\right)_{(\alpha_c, \delta_{TS_c})}\delta_{CS_c}(\alpha - \alpha_c) + \\
& + \left(\frac{\partial C_{X_{\delta_{CS}}}(\alpha, \delta_{TS})}{\partial\delta_{TS}}\right)_{(\alpha_c, \delta_{TS_c})}\delta_{CS_c}(\delta_{TS} - \delta_{TS_c}) + \\
& + C_{X_{\delta_{CS}}}(\alpha_c, \delta_{TS_c})(\delta_{CS} - \delta_{CS_c})\Bigg)\Bigg] + \mathcal{C}
\end{aligned}
$$

where the subscript c denotes the current value of a variable and $\mathcal{C}$ collects all the values determined at the current state and input from the Taylor series expansion. In trim $\dot{V}_c$ and $\mathcal{C}$ are zero.

This procedure allows linearised models to be obtained quickly enough for real-time use in the case of $MBPC$ because of the way the internal model is written in Δx and Δu. Of course this assumes that the FDI system can update the look-up tables quickly enough.

For the simulations shown in Section 10 we have employed numerical linearisation instead (the Simulink function linmod), for convenience. In principle this gives the same linearised model because of the way the A, B, C, D matrices are computed in the case of a trim or non-trim pair of inputs and states.

6 The autopilot structure

In Figure 5 all the essential components of the system which perform the functions referred in the Section 2 for the missile case are depicted.

The quasi-LPV model runs in parallel with the nonlinear plant providing the states equilibrium values $q_{eq}(\alpha)$, $\delta_{eq}(\alpha)$ together with the $A(\alpha)$, $B(\alpha)$, C, D matrices depending on the current value of the parameter α.

The structure of the $MBPC$ controller consists of standard modules. States, including α, are assumed measurable, since they are usually available in flight control. We use a state-space LTI model for the plant, different at each time step and corresponding to the parameter value α.

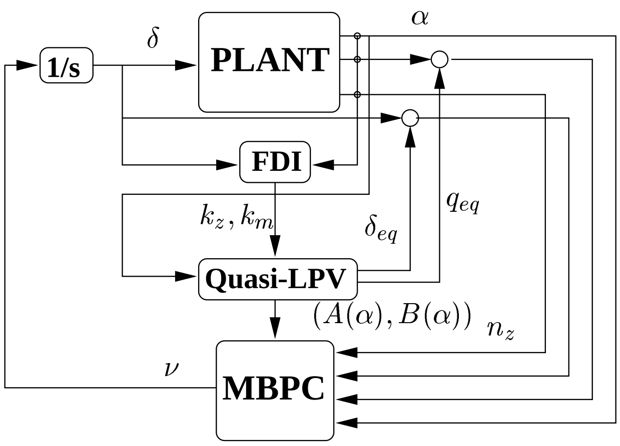

Figure 5: The reconfiguration and scheduling $MBPC$ based strategy: case 1

The other alternative of the on-line point-wise linearisations used in the HIRM case is depicted in Figure 6. Here the strategy is the same as in the previous case with the exception of the way the $MBPC$ internal model is produced.

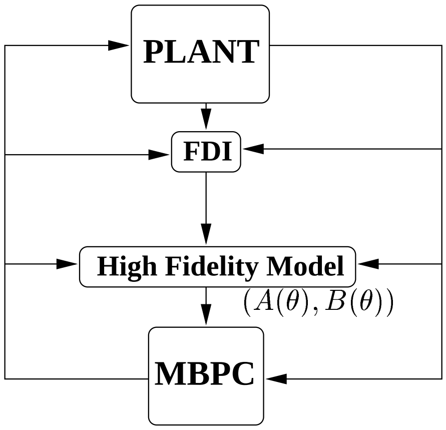

Figure 6: The reconfiguration and scheduling $MBPC$ based strategy: case 2

7 Designing the $MBPC$ controller for the unfailed condition

7.1 The missile case

The controller design followed the general procedure used in conventional $MBPC$ cases. This involved tuning the $MBPC$ controller for a plant model obtained by freezing the healthy quasi-LPV model at $\alpha = 0$. Note that this model is unstable. The plant model used for control had the following (approximated) continuous time transfer function:

$$\frac{n_Z}{\nu} = \frac{(s - 31.58)(s + 31.30)}{s(s + 6.0029)(s - 5.3984)}$$

A sampling time of $T_s = 0.01s$ will provide a good approximation over the whole open-loop bandwidth.

Horizon dimensions were defined according with the general guide lines used in tuning $MBPC$ controllers [MBT97, HM96]. The tuning was performed in the unconstrained case.

The control horizon was chosen to be the same as the number of states ($N_u = 3$ in this case) which will ensure stability even in the worst case, namely when all the modes are unstable, should that ever occur [RM93].

The small sampling time has determined a long prediction horizon of $N_y = 30$ in order to predict $0.3s$ ahead. This was enough to ensure the performance criteria required [SC93].

The $MBPC$ control weighting matrix had to be made relatively small $R = 0.005$ due to the small effectiveness of the fin in this particular model. The constraint predictive controller tracking weighting matrix $Q = 1$ reflects the requirements on the time response characteristics (rise time, settling time and overshoot) for the closed loop system. These were obtained after some trial and error, starting with unit values for both weighting matrices.

The next step in design, after checking the unconstrained closed loop behaviour, was to enforce the constraints and check the constrained $MBPC$ performance. Once these steps were passed the controller was checked when performing adaptation with respect to α.

7.2 The HIRM case

As in the missile case the controller design followed the general procedure which we have developed for conventional $MBPC$ control [MBT97, HM96]. This involved tuning the $MBPC$ controller for a plant model obtained by linearising the healthy high-fidelity model for an equilibrium point corresponding to a level flight with Mach number $M = 0.30$ at a hight of $H = 5000\ ft$ with parametric uncertainties in M of 3%.

A sampling time of $T_s = 0.05s$ provided for the discrete time model a good approximation of the continuous time one over the whole open-loop bandwidth (the smallest constant which appears in the model is on the transfer function from symmetrical taileron (δ_{TS}) to pitch rate (q) and it is defined by a cross-over of about $10 rad/s$).

The HIRM longitudinal model contains actuator models for which the bandwidths are at $60 rad/s$, $10 rad/s$ and $1 rad/s$ for canards, taileron and engines, respectively.

In order to speed up the process the tuning was performed first for the unconstrained case. The control horizon was chosen to be the same as the number of states $N_u = 11$ but the small sampling time has determined a long prediction horizon of $N_2 = 60$ in order to predict 3 s ahead. This was enough to ensure the achievement of the performance criteria.

The $MBPC$ control weighting matrix had to be made relatively big
$R = \begin{bmatrix} 9 \times 10^4 & 3 \times 10^4 & 1 \times 10^4 \end{bmatrix}$, a typical choice for longitudinal channel of the aircraft.

The $MBPC$ tracking weighting matrix $Q = \begin{bmatrix} 700 & 1 \end{bmatrix}$ reflects the requirements on the time response characteristics (rise time, settling time and overshoot) for the closed loop system. These were obtained after some trial and error, starting with unit values for both weighting matrices.

8 Automatic Tuning for $MBPC$ During Reconfiguration

In [Mac97, HM98a, HM98b] and the previous section a particular combination of high fidelity models, fault detection and isolation (FDI), model approximation/simplification and constrained predictive control which enables a generic solution to reconfiguration and adaptation was described.

As it will be shown in Section 9 it is not too simplistic to believe that it is enough to change the internal model and the constraints limits in order to reflect a fault for which the $MBPC$ controller has to provide satisfactory control inputs. But, in the most general case the controller needs to be re-tuned in order to achieve satisfactory performance. "Tuning" here means the adjustment of the $MBPC$ cost function weights and possibly the relaxation of some of the constraints in order to make the problem feasible.

Therefore, in this section, on-line optimisation to provide the tuning parameters that will enable the reconfigured controller to provide good performance is investigated. In Section 10.1 a practical problem of reconfiguring a missile on-line in the presence of major impairments is solved.

8.1 Matching the reference closed loop eigenvalues via on-line optimisation

Our main goal is to have in the case of the impaired system the same responses as the ones provided by the healthy plant and its corresponding controller. Before discussing about the optimisation problem involved to solve the matching of the two closed loops we would like to point out two assumptions made, valid for both Section 8.1 and 8.2.

First one assumes the horizons as fixed parameters for both the nominal and the impaired system cost functions. In fact horizon dimensions are defined according with the general guide lines [Soe92] used in tuning $MBPC$ controllers. The control horizon was chosen to be the same as the number of model states which will ensure stability even in the worst case when all the plant modes are unstable [RM93]. The prediction horizon is determined by the sampling time and the performance to be reached with the closed loop.

The second assumption refers to the order of the impaired system. Because in our case the impaired model is obtained employing the high fidelity model written in the quasi-LPV form [HM98a] the number of states is considered to be the same in spite of major structural failures.

Authors like [GA91] applying pseudo-inverse reconfiguration methods, translate the matching of the healthy and impaired system responses into the goal of keeping the closed loop eigenvalues of the impaired system in the same region as in the nominal case, if possible.

Initially, we matched the $MBPC$ controller by performing a Frobenius norm optimisation of the form:

$$\min_{\Theta} \|K_f(\Theta) - K_n\|_F$$

The results obtained drove us to the conclusion that a matching of the closed-loop eigenvalues with the ones corresponding to the nominal closed-loop will be more appropriate.

As result the following optimisation was employed:

$$\min_{\Theta} \|\bar{A}_f(\Theta) - \bar{A}_n\|_F$$

The minimisation was carried out using the `fminu` function from the Matlab Optimisation Toolbox, a nonlinear optimisation algorithm based on quasi-Newton method using a mixed quadratic and cubic search procedure. This was employed due to the particular nonlinear dependency of A_f on Θ.

The form of the cost function (1) depending on Q and R explains the existence of a non-unique solution. In other words, for the SISO case, the non-uniqueness of the solution can be explained observing that the actual parameter in the optimisation problem is the ratio between the weights Q and R.

The reason of choosing the Frobenius norm as a measure for the distance between the two closed loop matrices is not only related to the optimisation feasibility but as well to stability guarantees that we provide when the reconfigured controller is employed. In the case of a small perturbation upon the eigenvalues of the closed loop (A_f) from the nominal closed loop (A_n) the Bauer-Fike theorem gives a condition for the location of the eigenvalues of the reconfigured closed loop. The stability guarantees are restricted to the assumption that A_n is non defective in the sense that it can be reduced to a diagonal form by a similarity transformation.

Theorem 8.1 *If μ is an eigenvalue of the matrix $A_f \in \mathbb{R}^{q\times q}$, $X^{-1}A_nX =$ $= diag([\lambda_1, \dots, \lambda_q]) = D$ where X is the matrix of the A_n eigenvectors, $k_2(X)$ is the X matrix condition number and λ is an eigenvalue of A_n ($\lambda \in \lambda(A_n)$) then:*

$$\min_{\lambda\in\lambda(M_n)} |\lambda - \mu| \leq k_2(X) \parallel E \parallel_F \tag{5}$$

where $E = A_f - A_n$

Basically the theorem defines a disk, centred on the old eigenvalue, in which a new eigenvalue will be located once the optimisation performed. This bound is given by the condition number of the eigenvector matrix of A_n and the value of the cost function.

The eigenvalues of A_f lie in the union of disks $|\lambda - \mu|$ having a radius $k_2(X) \parallel E \parallel_F$. We recommend a nominal design that will place the nominal closed loop eigenvalues centred within the open unit disk ensuring an increased level of robust stability to the system. Using the above procedure, for a good conditioning of eigenvectors of the nominal A_n, the minimisation performed provides a closed loop behaviour of the reconfigured system close to the nominal one. The increased control effort used, that requires enough actuator authority, might be considered as a disadvantage of the current design procedure which made us to consider the solution presented in the next section.

If $k_2(X)$ is large then small changes in A_f can induce large changes in the closed loop eigenvalues. In the case of a practical problem a poor conditioning of the matrix of eigenvectors of A_n (X) is possible. This is a problem located at the level of the nominal design. Further research has to be carried out in order to provide a good conditioning for the $k_2(X)$ number when designing the $MBPC$ controller for the nominal plant.

Despite the stability guarantees and a relatively simple optimisation procedure the above approach suffers from the fact that only the closed-loop poles are matched but not the zeros, which are known to have as well a big effect on the system response.

8.2 Control reconfiguration automatic design procedure

The natural evolution from the previous way of addressing the problem of reconfiguration was to solve a minimisation which accounts for both closed-loop poles and zeros matching the system four-block transfer matrix.

The automatic tuning procedure employed as part of the overall control mechanism follows the following steps:

1. A nominal controller is designed for the given performance specifications having the nominal realisation $\left[\begin{array}{c|c} A_n & B_n \\ \hline C_n & D_n \end{array}\right]$ as the internal model.

2. Given a plant model $\left[\begin{array}{c|c} A_f & B_f \\ \hline C_f & D_f \end{array}\right]$ reflecting an impairment a closed loop realisation is produced depending on the elements of the cost function weighting matrices Q and R which we denote by the parameter vector Θ.

3. A minimisation procedure is performed with respect to Θ of the ∞-norm of the difference between the transfer matrix defining the nominal reference closed loop $\hat{M}_n\left(\left[\begin{array}{c|c} A_n & B_n \\ \hline C_n & D_n \end{array}\right]\right)$ and the transfer matrix $\hat{M}_f\left(\left[\begin{array}{c|c} A_f & B_f \\ \hline C_f & D_f \end{array}\right], \Theta\right)$ which is defined on the basis of the impaired model controlled by the $MBPC$, depending on the parameter Θ.

The procedure can be frequency weighted via W_i and W_o, as common in model matching setups, leading to the following optimisation problem with respect to the parameter Θ:

$$\min_{\Theta} \left\| W_o \left[\hat{M}_f\left(\left[\begin{array}{c|c} A_f & B_f \\ \hline C_f & D_f \end{array}\right], \Theta\right) - \hat{M}_n\left(\left[\begin{array}{c|c} A_n & B_n \\ \hline C_n & D_n \end{array}\right]\right)\right] W_i \right\|_{\infty} \tag{6}$$

The algorithm chosen to perform this minimisation is based on a fairly brute force procedure but in exchange it provides feasibility together with good control of the level of matching. This procedure proved successful when severe impairments were encountered.

Therefore here we include the steps of the algorithm which produces a solution for Θ that gives the best matching:

1. Perform a gridding of the parameter space Θ with an appropriate step.

2. Compute for the current value of the parameter the value of the cost function from (6).

3. Store the value of Θ and the corresponding norm if smaller than figures previously obtained.

4. Iterate Steps 2 and 3 for all the grid vertices.

Together with the ability of this method to provide automatic tuning for the $MBPC$ controller while reconfiguring in case of failures we consider other features included. For instance using MIL type of specifications for the closed loop it is possible to employ this method to tune the controller in such a way as to match a reference model. The approximation of other state feedback controllers with an $MBPC$ regulator is possible in a similar manner.

9 Evaluation results

9.1 Evaluating the scheduling and reconfiguration capabilities

The two different capabilities of the controller – scheduling on α and reconfiguration – are investigated in the following simulations.

Firstly we show the behaviour of the closed loop with the different controllers when the system is subjected to a step of 0.1 in the non-dimensional controlled variable n_Z.

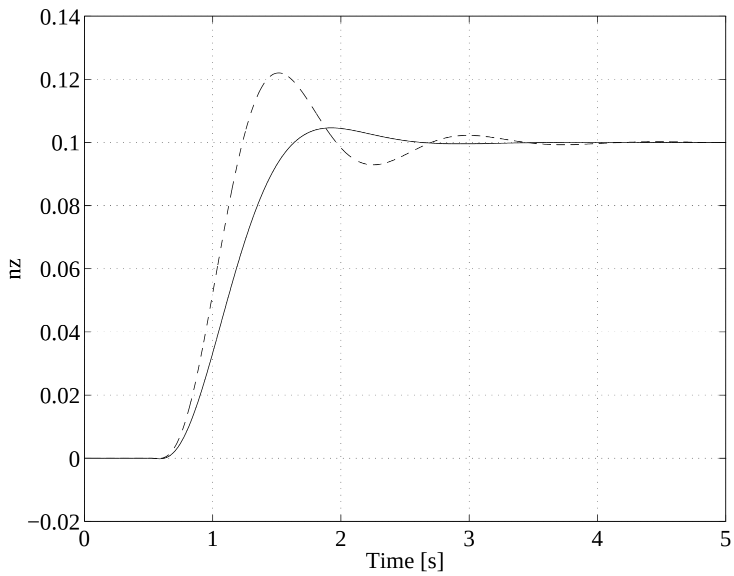

Figure 7: Comparison between the fixed and the adaptive $MBPC$ (adaptive–solid, fixed–broken)

The first controller is represented by the nominal design for which a fixed internal model, defined at $\alpha = 0$ is used, even though α varies during the manoeuvre.

In the second case the controller uses the fixed values for the cost function tuning parameters (N_y, N_u, Q, R), but the internal model is updated to follow the evolution of α, as described in Section 4

For each such case we have ensured stability by an adequate choice of the control horizon (N_u greater than the system order). In this first case no failures are assumed for the controlled system.

The response of the system controlled by the adaptive $MBPC$ provides less overshoot and a shorter settling time as demonstrated in Figure 7.

In the second simulation we show the reconfiguration capability of the new approach. Successive structural failures of the missile were simulated: a failure on the body at $t =$

0.3s, and one on the fin at $t = 0.5$s. These failures modified the aerodynamic characteristics by 10% and 50%, respectively. Figure 8 shows how the controller coped with each failure.

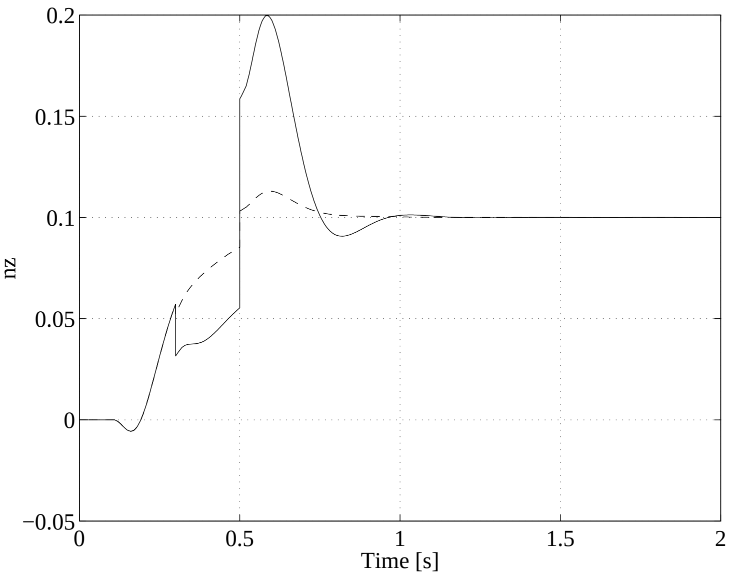

Figure 8: The reconfiguration capabilities provided by the new approach (update internal model–broken, nominal internal model–solid)

The failure was simulated by manipulating the nonlinear model dynamics. The impairment was represented in the high fidelity model by changing the coefficients relating the contributions of various aerodynamic surfaces to C_Z and C_m.

In both situations the new controller behaved well, providing scheduling and reconfiguration as required.

10 Simulation results for several failure scenarios in the HIRM case

10.0.1 Actuator failures

One of the most frequent failures encountered in real life is actuator failure. The reduction of control authority leads to reduced performance, or even instability.

The simulation result shown in Figure 9 shows the system (HIRM) response when the canard actuator failed during a pitch rate demand of 6 deg/s. The step in pitch rate demand occurred at time 5 s followed by the failure, a stuck symmetrical canard, which occurred at time 8 s.

This failure, was represented by tightening the constraints on the allowed canard deflection in the $MBPC$ optimisation. Due to the $MBPC$ feature of handling constraints explicitly no other change was necessary in the structure of the controller.

The solid line in Figure 9 shows the response when no information about the failure is passed to the controller.

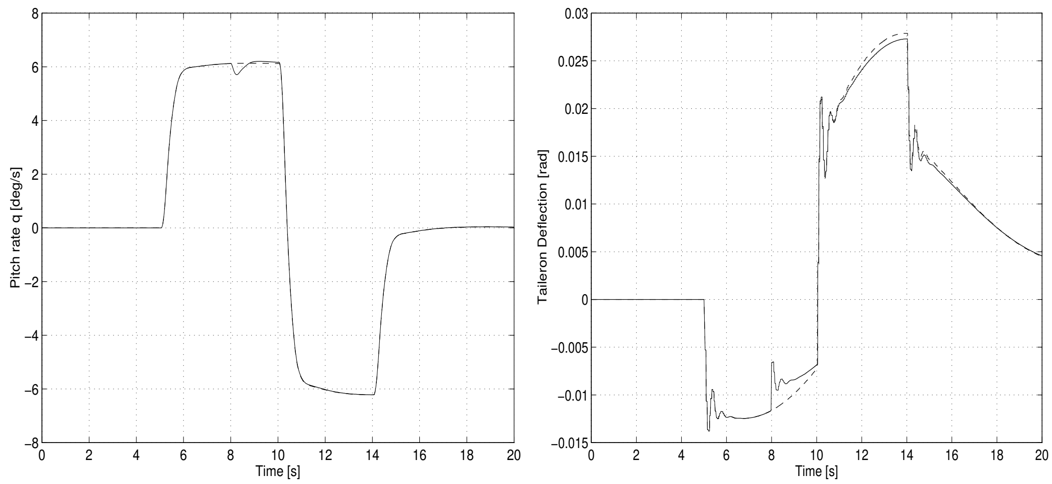

Figure 9: The reconfiguration capabilities provided by the new approach in the case of actuator failures (nominal – broken, stuck canard – solid)

By looking at the performance criteria stated in the GARTEUR Design Challenge [MBT97], for this manoeuvre, we consider the controller performance in the presence of the failure to be satisfactory.

10.0.2 Structural failures

For a fighter aircraft, we can assume that one of the typical failures that its controller has to cope with is battle damage which, in general falls in the class of structural failures.

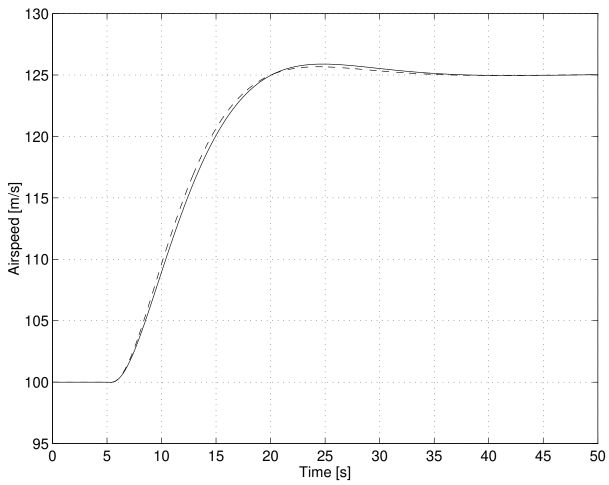

Figure 10: The reconfiguration capabilities provided by the new approach in the case of structural failures (nominal – solid, 20 % change in aerodynamic coefficients – broken)

The simplest way to simulate structural failures in the aircraft high fidelity model is by means of the non-dimensional aerodynamic coefficients, which are modified accordingly.

For instance a missing part of an aerodynamic surface translates into more drag and a reduced lift, which in their turn are translated into new values of the aerodynamic coefficients. This is according with a study [Hoa78] which relates various types of structural failures to the aerodynamic model.

In our scenario we assume that such a structural failure, which changed the values of the aerodynamic coefficients by 20 %, occurred at time 8 s while a change of 25 m/s in speed set-point was demanded at time 5 s, see Figure 10. A fairly large set-point change was chosen as part of the scenario because we want to show the controller fault tolerant features. During this manoeuvre the actuator limits were not reached.

The performance of the reconfigurable flight controller, shown in Figure 10, complies with the criteria mentioned in GARTEUR Design Challenge document [MBT97].

10.0.3 Combined failures

Now we consider a simultaneous actuator and structural failure. This is typical of battle damage or of fatigue fracture.

In Figure 11 the performance of the controller is investigated when at time 5 s a change of 25 m/s in the air speed was demanded, followed at time 8 s by an actuator failure and at time 15 s by a structural failure similar to the ones described in Sections 10.0.1 and 10.0.2.

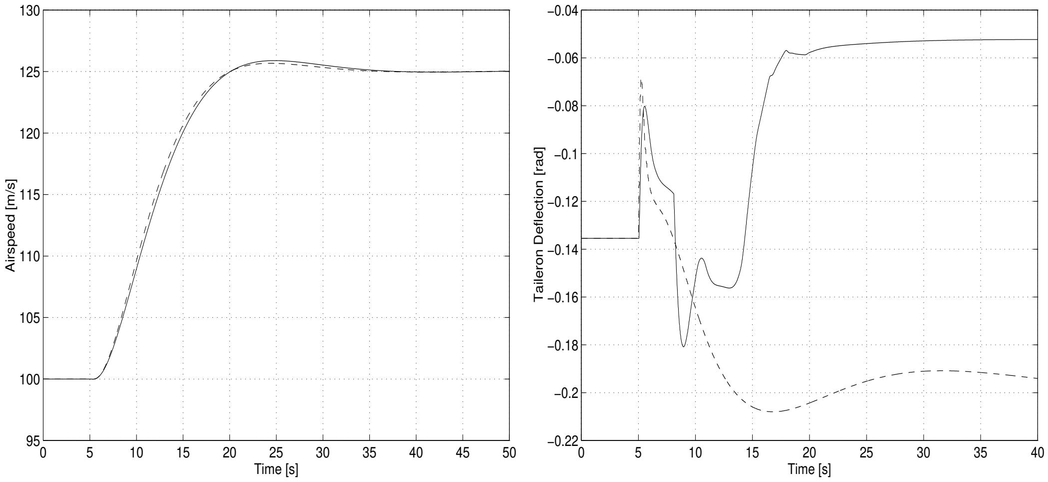

Figure 11: The reconfiguration capabilities provided by the new approach in the case of a combined failure (nominal – broken, combined structural and actuator failure – solid)

The model chosen to illustrate the algorithm was subjected to several failure scenarios which showed the ability of the controller to meet robust performance criteria mentioned in the GARTEUR design challenges.

10.1 Checking the automatic tuning procedure with the missile model

In this section we concentrate on the same missile model of which complete details are given in [SC93]. In flight scheduling for the nominal loop and reconfiguration were performed using quasi-LPV high-fidelity models and $MBPC$ control as described in the previous section.

The $MBPC$ internal model is based on the plant having the approximated discrete time transfer function (7) obtained for a sampling time of $T_s = 0.01$s. This model was defined at $\alpha = 0$, even though α varies during the manoeuvre, is the same one as in the previous section described now in discrete time:

$$\frac{n_Z}{\nu} = \frac{(z-1.3732)(z-0.073)}{(z-1)(z-0.9417)(z-1.055)} \tag{7}$$

The structural failure was simulated by manipulating the nonlinear model dynamics. The failure model (8) was obtained for values of the parameters k_Z and k_m, see [HM98a], reflecting a failure that modified the aerodynamic characteristics. The resulting plant model is:

$$\frac{n_Z}{\nu} = \frac{(z-0.77)(z-1.42)}{(z-1)(z-1.12)(z-0.975)} \tag{8}$$

As seen from (7) and (8) major failures lead to a change in the linear internal model. The severity of these impairments requires controller re-tuning.

The automatic tuning procedure keeps the horizons fixed but adjusts the weights in (1) via the optimisation (6) performed with respect to Θ. The control horizon was chosen to be the same as the number of states ($N_u = 3$ in this case) which will ensure stability even in the worst case, namely when all the modes are unstable, should that ever occur [RM93]. The small sampling time has determined a long prediction horizon of $N_y = 30$ in order to predict 0.3s ahead. This was enough to ensure the performance criteria required in [SC93]. The nominal values chosen for Q and R were 0.005 and 1, respectively.

The new weights $R = 0.035$ and $Q = 1.12$ obtained from the optimal Θ were computed using the procedure described in the Section 8.2. The value for the grid step was in this case 0.01 but for other cases exhibiting a large ratio between Q and R it has to be much finer. Of course the grid step influences the solution speed.

We show in Figure 12 the behaviour of the closed loop with the different controllers when the system is subjected to a step of 0.1 in the non-dimensional controlled variable n_Z.

In Figure 12 the dash-dotted line depicts the behaviour of the nominal plant with an $MBPC$ controller based on a nominal internal model and nominal tuning. The broken line shows the impaired plant controlled by the $MBPC$ having information about the failure via the internal model but still using the cost function nominal tuning. The solid line shows the behaviour of the impaired plant this time controlled by the re-tuned $MBPC$ using the newly developed automatic procedure.

11 Comments on the FDI system

Critical failures in flight control may result from sensor failure, actuator, structure damage or breakdown. The objective of most of RFCS – to permit a graceful degradation of system performance in the presence of failures, damage or malfunctions – require the presence

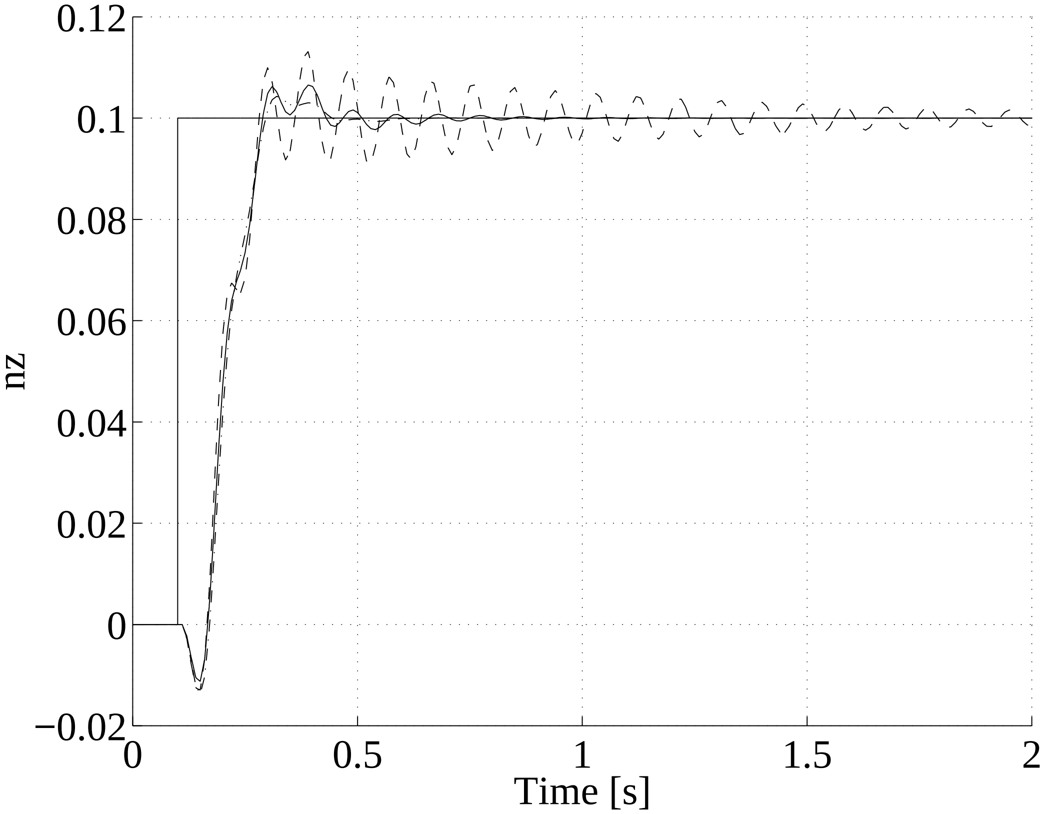

Figure 12: Evaluation of the $MBPC$ automatic tuning

of a Fault Detection and Isolation (FDI) system. The FDI is required to quickly and reliably determine the faulty components in the presence of noise and modelling errors. Primarily requirements for a succesfull fault detection are: low false alarm, robustness against disturbances and manoeuvres and minimal detection delay. Although developing the FDI is beyond the purpose of these work we give a brief review of main strategies used at the moment. This is due to the possible development and their use in conjunction with high fidelity models.

Involving into the plant model information about the delay of the FDI system in providing information about the failed plant is a difficult task. Before the failure the controller based on the nominal model is used. At the time of the failure until the FDI decides the new characteristics of the plant and hence the model in use we require a certain degree of robustness to be built in the controller. After enough information was acquired about the plant the FDI system will be able to select from a collection of pre-computed controllers the one which provides the best performance and stability. The filters carefully tuned have to avoid the choice of a "bad" model. Of course, robust control design enable us as well to reduce this number up to an implementable one.

An usual FDI system contains two levels:

1. An actuator failure FDI system used to detect and isolate the malfunction of the actuator which could have severe impact on the system response. Typically the first layer consists of devices that detect the failure by one of the following methods: hardware and parity checks and/or detection filters. Fault detection and isolation filters consists of a set of estimators arranged in several cascade layers to test for various failures.

2. A second layer refers the system itself having the main task of providing information regarding structural damage of the aircraft

According with the generally accepted terminology FDI systems perform the following tasks:

- Detection — the indication that something is going wrong in the system behaviour.
- Isolation — the determination of the exact location of the failure.
- Identification — the determination of the failure size.

As a result typical features for FDI systems are:

- Isolability — the ability to isolate certain specific faults provided their size is large enough; this test depend on the system matrices structure being a statistical based algorithm.
- Sensitivity — a qualitative measure defining the size of faults that can be isolated under certain conditions; it depends on the size of system matrices elements and noise properties having closely related the time in which the FDI produces a solution
- Robustness — the property that enables the existence of modelling errors; a serious interference with fault isolation has to be overcome using time history

It is required an enhanced robust stability and performance region achieved with each model and the corresponding controller in order to ensure blending and switching of them. The placement of the threshold, based on the ability of the FDI system and of individual controllers to provide robust closed loops, it is very important. By placing it at the significantly higher level than the detection one it is possible to reduce false alarms without a major decrease in the closed loop failure's sensitivity.

Several classifications of FDI methods were provided in literature but the one used here is based on the type of method used:

- failure sensitive filters
- voting systems
- multiple hypothesis filter detectors
- jump process formulations
- innovation based detection systems

In all these methods *a priori* knowledge of the plant dynamic model is required with more emphasise on this issue in the first case described above. For this case actually methods of detecting and isolating faults accurately are subject of current research.

In fact the research that has been carried out until now has not revealed completely how much further information can be provided by the FDI unit or what are the best mechanisms to perform the integrated design of the FDI and the reconfigurable controller.

Looking at the interaction between the FDI system and the high fidelity model we need to establish, if it is possible, how quick and accurate an update of this model can be such as to reflect the dimension of the failure. Our main concern is that the existence of one structural failure in general is driving towards a series of failures which can hardly be addressed by a system which deals with predetermined situations. For instance in the case quoted in the work of [Sma97] the separation of engines 3 and 4 of a Boeing 747-200F caused several structural and actuator failures:

1. right hand wing leading edge severely damaged
2. right hand wing leading edge flaps partly lost
3. no outboard aileron available due to the outboard trailing edge flaps failure
4. right hand inboard aileron less effective due to disturbed airflow caused by right wing damage and loss of pylon 3
5. the loss of pylons 3 and 4
6. partly lost spoiler system
7. malfunction of the lower rudder

In [Sma97] insight is provided into the flying capabilities of an impaired aircraft. In effect the author was updating off line an high fidelity model based on the data obtained from the Digital Flight Data Recorder (DFDR) such as to account for the right wing multiple engine separations and for the contribution of the right wing leading edge damage. This example represents an important argument to support our fault tolerant strategy assuming the existence of an FDI system which could have done in automatic manner what the author of [Sma97] achieved after certain effort during his research (*i.e.* an accurate update of the high fidelity model). For instance the augmentation of the aircraft aerodynamic surfaces and actuators with a system – an active component of the FDI – which informs the about the possible losses of parts or surfaces in flight can represent a potential alternative.

Going forward we can imagine a solution which involves a high fidelity model embedding all possible failure modes encountered in the past period of time in various accidents combined with an FDI system capable of detecting and isolating these. The FDI system probably will have the high-fidelity model – written as a function of the aerodynamic coefficients susceptible to modification as parameters. Then the identifying these parameters on line the computational burden can be a reduced and in the same time more knowledge about the model is built in the FDI system. The possible drawback of such scheme is the existence of a nonlinear optimisation that has to be employed for the current values of the model parameters. In principle operating such a system in real time will enable us to tackle the difficult problem of the on board implementation and to account for the failure modes encountered during various accidents.

12 Conclusions

Patcher et. al. [PCM95] were the first to view a simplified version of constrained predictive control as a promising tool for reconfiguration. Their scheme was an indirect adaptive approach with the system identification module employing a mixed identification method (optimisation involving flight mechanics information) and the controller being split into two loops (an LQR inner loop and an outer linear programme on-line optimisation to enforce constraints a step ahead).

In our case the objective was to find a systematic way of reconfiguring control systems in the event of actuator and structural failures. Therefore we consider that the particular combination of high fidelity models, fault detection and isolation (FDI), model approximation/simplification and constrained predictive control enables a generic solution to reconfiguration and adaptation [Mac97].

Assuming reliable FDI, the proposed approach gives:

- A systematic and compact in some cases way of representing high-fidelity models,
- Automatic updating of models following failure,
- Real-time production of internal models for $MBPC$ control,
- An autopilot structure which integrates the components shown in Figure 5.

The exploitation of the quasi-LPV models fits naturally in the $MBPC$ framework. Since no linearisation is involved these models are not limited to standard gain-scheduled designs. Moreover, there is a wide class of systems exhibiting output nonlinearities. They are particularly common in aerospace, but there are also process models which can be expressed in a similar form.

There has been some progress in analysing the stability of certain multi-model control schemes [NB97, Mor93], although it is too early to know whether these approaches will be helpful for our proposal.

The automatic tuning procedure represents an extension to a general reconfiguration strategy described in [HM98a]. Depending on the bounds defined by the ∞-norm computation we can state for a practical problem how big the failure can be to have the method coping with it.

The examples tackled and the failure scenarios suggested are common enough to enable the reader to have a documented opinion about the performance of the fault tolerant method suggested. Of course more things have to be done in this direction by setting up a benchmark in terms of failure scenarios but this initial attempts opened already the path on which we are going to pursue our future research. For a while we are going to concentrate on the HIRM model but a possible move towards civil aviation aircraft can be foreseen.

Although our approach is quite complex, we believe that the complexity is worthwhile if effective scheduling and reconfiguration can be achieved. The results of Section 9 are encouraging, and papers such as [PCS97] indicate that others also take this view. Addressing the problem in such a manner will have a serious economical impact upon the sales of the aircraft producing companies. Therefore embedding such a fault tolerant system into the aircraft's flight controller will result in less incidents or accidents which will enable open funding of future research.

13 Acknowledgements

We would like to acknowledge Pembroke College Cambridge, Lundgren Fund, ORS scholarship, Cambridge Overseas Trust and CT Taylor Fund by which Mihai Huzmezan is supported. We acknowledge the assistance given by the DERA – Bedford in support of the work covered in this paper. We would like to acknowledge as well G. Papageorgiou for the discussions held regarding the topic of this paper.

References

[CKC96] C.M. Chow, A.G. Kuznetsov, and D.W. Clarke. Using multiple model predictive control. In *Proceedings of the spring school "Adaptive and Predictive Control"*, Oxford, 21–22 Mar. 1996. Oxford University.

[GA91] Z. Gao and P.J. Antsaklis. Stability of the pseudo-inverse method for reconfigurable control systems. *International Journal of Control*, 53(3):717–729, 1991.

[GL96] G.H. Golub and C.F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, third edition, 1996.

[HM96] M. Huzmezan and J.M. Maciejowski. RCAM Design Challenge Presentation Document: The Model Based Predictive Control Approach. Technical Report FM(AG08) TP-088-20, Group for Aeronautical Research and Technology in Europe (GARTEUR), 1996.

[HM98a] M. Huzmezan and J.M. Maciejowski. Reconfiguration and Scheduling in Flight Using Quasi-LPV High-Fidelity Models and *MBPC* Control. In *Submitted to the American Control Conference*, July 1998.

[HM98b] M. Huzmezan and J.M. Maciejowski. Reconfigurable Flight Control of a High Incidence Research Model Using Predictive Control. In *IEE Proceedings-D*, January 1998.

[Hoa78] D.E. Hoak. USAF Stability and Control DATCOM. Technical Report AFWAL-TR-83-3048, Air Force Wright Aeronautical Lab., Wright-Patterson AFB, 1978.

[Hyd95] R.A. Hyde. H_∞ *Aerospace Control Design.* Springer–Verlag, 1995.

[Mac97] J.M. Maciejowski. Reconfigurable Control Using Constrained Optimisation. In *Proceedings of ECC'97*, Bruxelles, Belgium, 1997.

[MBT97] JF. Magni, S. Bennani, and J. Terlouw. *Robust Flight Control: A Design Challenge, GARTEUR*, volume 224 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, 1997.

[Mor93] A.S. Morse. Supervisory control of families of linear set point controllers. In *Proceedings of the 32nd Conference on Decision and Control*, San Antonio, Texas, Dec 1993. IEEE.

[NB97] K.S. Narendra and J. Balakrishnan. Adaptive Control Using Multiple Models. *IEEE Transactions on Automatic Control*, 42(2):171–187, 1997.

[PCM95] M. Patcher, P.R. Chandler, and M. Mears. Reconfigurable Tracking Control with Saturation. *Journal of Guidance Control and Dynamics*, 18(5):1016–1022, Sep.–Oct. 1995.

[PCS97] M. Patcher, P.R. Chandler, and L. Smith. Maneuvering Flight Control. In *Proceedings of the American Control Conference*, July 1997.

[RM93] J. Rawlings and K. Muske. The stability of constrained receding horizon control. *IEEE Transactions on Automatic Control*, 38:1512–1516, 1993.

[SC93] J.S. Shamma and J.R. Cloutier. Gain-Scheduled Missile Autopilot Design Using Linear Parameter Varying Transformations. *Journal of Guidance Control and Dynamics*, 16(2):256–263, Mar.–Apr. 1993.

[Sma97] M.H. Smaili. Flight Data Reconstruction and Simulation of EL AL Flight 1862. Technical report, Delft University of Technology, Faculty of Aerospace Engineering, November 1997.

[Soe92] A.R.M. Soeterboek. *Predictive Control - A unified approach*. Prentice Hall, New York, first edition, 1992.

[Wil97] S.W. Willcox. Private communication, DERA. Technical report, 1997.

[Woo95] G.D. Wood. *Control of Parameter–Dependent Mechanical Systems*. PhD thesis, University of Cambridge, 1995.