# MPC FAULT-TOLERANT FLIGHT CONTROL CASE STUDY: FLIGHT 1862

**Jan M. Maciejowski** * **Colin N. Jones** *,[1]

* *Control Group, Department of Engineering*
*University of Cambridge, Trumpington Street*
*Cambridge CB2 1PZ, UK*
{*jmm,cnj*}*@eng.cam.ac.uk*

Abstract: We demonstrate that the fatal crash of El Al Flight 1862 might have been avoided by using MPC-based fault-tolerant control. Simulation on a detailed nonlinear model shows that it is possible to reconfigure the controller so that the aircraft is flown successfully down to ground level, without entering the condition in which it was lost. We use a reference-model based approach, in which an MPC controller attempts to restore the original functionality of the pilot's controls. For the purposes of simulation, we emulate the pilot by another MPC controller, running at a lower sampling rate. We assume in this paper that an FDI function delivers information about actuator damage, and about changes to aerodynamic coefficients in the failed condition.

Keywords: Model predictive control, Fault-tolerant control, Flight control.

## 1. INTRODUCTION

In (Maciejowski, 1997*a*; Maciejowski, 1997*b*) it was argued that Model Predictive Control (MPC) provides a suitable 'implementation architecture' for fault-tolerant control. The representation of both faults, and of control objectives, is relatively natural and straightforward in MPC. Actuator faults such as jams and slew-rate reductions can be represented by modifying the constraints in the MPC problem definition (Camacho and Bordons, 1999; Maciejowski, 2002). Other faults can be represented by modifying the internal model used by MPC — either detailed modifications to a detailed first-principles model, of the kind being used increasingly in the process industries, or modifications of the overall behaviour of a 'black-box' model. In addition, MPC has a degree of fault-tolerance to actuator faults under certain conditions, even if the fault is not detected (Maciejowski, 1998).

The previous paragraph assumes that at least the effects of a (possibly multiple) fault can be identified. This is undoubtedly the most difficult part of solving the fault-tolerant control problem — although this difficulty is being radically affected by the availability of self-diagnosing actuators and sensors, and increasingly effective condition monitoring schemes. Nevertheless this assumption will be maintained in this paper, in order to show, by careful investigation of a specific historical incident, that it is at least plausible — no more will be claimed here — that MPC can provide effective solutions for fault-tolerant control.

The possibility of fault-tolerant control arises only if there is, in some sense, enough redundancy in the system being controlled to allow the effects of

[1] Partially supported by the Natural Sciences and Engineering Research Council of Canada and the Cambridge Commonwealth Trust

a fault to be in some way circumvented. Flight control is a promising application area for fault-tolerant control, because aircraft, in addition to being very fully instrumented, usually have some actuator redundancy. Civilian airliners, for example, have spoilers (air brakes) which are sometimes used to provide a rolling moment at low speeds, additional to that available from conventional ailerons. Advanced aircraft concepts, such as the 'flying wing', promise many multi-function control surfaces, each capable of being deflected independently of the others.

There have been several instances of aircraft incidents in which the pilot(s) has successfully used the redundancy of actuators to save an aircraft from an apparently hopeless failure condition. Perhaps the most spectacular is the well-known Sioux City DC-10 incident (Hughes and Dornheim, 1989), in which the aircraft was saved despite total loss of hydraulic power, by clever manipulation of the thrust from the two surviving engines. There have also been several incidents in which the crew has not managed to save the aircraft, although post-flight analysis showed that it was possible to do so. Such incidents illustrate that there is scope for automatic fault-tolerant flight control systems. In this paper we investigate one such incident, that of El Al Flight 1862, which lost two engines on taking off from Schiphol Airport in Amsterdam.

Before proceeding, let us address one aspect which is sometimes raised as an objection to our concept of fault-tolerant flight control — whether such a flight control law could ever be accepted and certified. We point out that it is not necessary for such a system to be operating continuously. It could, for example, be activated only by an express command from the pilot. And once activated, it need not be thought of as a kind of autopilot, taking over from the human pilot; it could form no more than an advisory system, proposing courses of action, or showing consequences (ahead of real time) of actions initiated by the pilot.

## 2. ROBUST, ADAPTIVE AND FAULT-TOLERANT CONTROL

How does 'fault-tolerant' control differ from 'robust' or 'adaptive' control? It is not clear that, given a flight control system, it is possible to distinguish whether it is 'robust' or 'adaptive' from observation of its behaviour (Wang and Zames, 1991). The same is probably true of a 'fault-tolerant' control system. However, examination of its internal structure (algorithms) may reveal into which of these categories it falls. Intuitively, 'adaptive' systems can cope with a greater range of variations in system behaviour and external environment than 'robust' systems. But much of the literature of both fields assumes that the objectives and performance specifications of the control system remain unchanged. Some of the literature makes much more severe assumptions, for example that the plant under control undergoes parameter variations only, while its structure remains unchanged.

In our view 'fault-tolerant' control certainly considers the possibility of structural changes in the plant being controlled (including changes in dynamic behaviour, changes in available actuators, and changes in available sensors), as well as the possibility that control objectives may need to be changed. In this paper we consider an example in which major structural changes occur to the plant. We do not address directly the question of changing control objectives here. However, the architecture which we adopt leaves a human operator (in this case the pilot of the aircraft) in the loop. Although our reconfiguration attempts to restore the initial effects of the pilot's controls, even in the failed condition, one can to some extent rely on the pilot's experience to moderate the demands made of the aircraft, and in this way modify the control objectives. One could argue that if the reconfiguration compensated perfectly for the effects of the fault, then the presence of the fault might be hidden from the pilot. However, this is unlikely to be the case, and one can imagine other indicators being given to the pilot that there is a problem; for example, some measure of the reconfiguration of actuators — unusual patterns of use — could be devised.

An extensive survey of fault-tolerant control is given by (Patton, 1997). In this survey 'robust control' is included in the category of 'passive fault-tolerance', whereas both adaptive control and reconfigurable control are considered to be included in 'active fault-tolerance'. The survey also rightly emphasises the importance of using, rather than replacing, a human operator.

## 3. PILOT-BASED FAULT-TOLERANT FLIGHT CONTROL

Of the 421 fatal accidents involving large jet aircraft between 1990 and 1999, not one was caused by loss of pilot (Civ, 2000). Because of this fact, we make the assumption that for large passenger planes there will always be a pilot (or several) available to fly the plane. In a failure situation the pilot is a valuable resource who must, if possible, be utilised. Removing him from the loop is a last resort and should only be done if it can be shown that he cannot fly the plane. For this reason, we propose a method which, if possible,

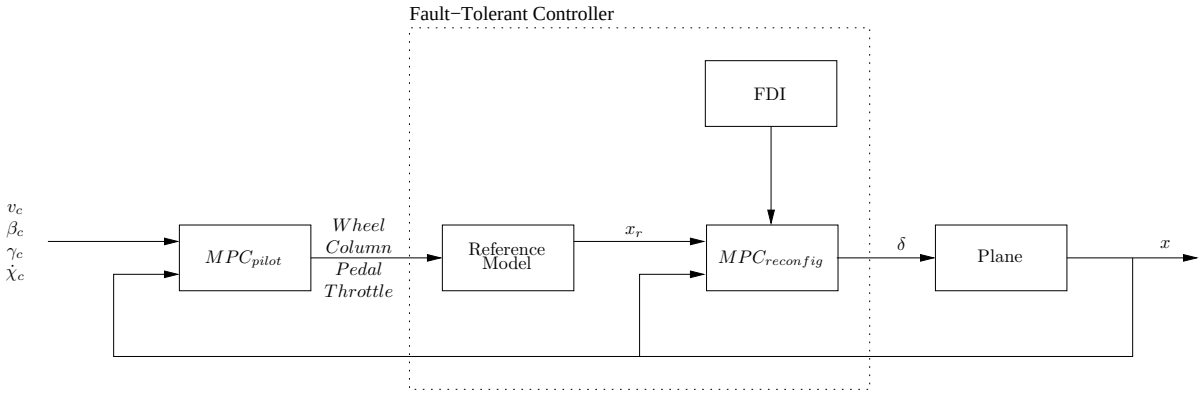


Fig. 1. MPC Fault-Tolerant Controller

compensates for failures while allowing the pilot to continue flying the plane.

A sufficient condition for 'pilot fly-ability' is for the aircraft to operate in a fashion which is close to its nominal specification, i.e., from the pilot's point of view, the column, wheel, pedal and throttle inputs cause close to the same effect as they do on the working plane. This can be cast as a constrained model-following problem. Figure 1 contains a box labelled *'Fault-Tolerant Controller'*, which comprises three components: the block 'FDI' which performs identification of the fault's effects (and whose presence we assume, as discussed above), a *'Reference Model'* which uses pilot commands to generate a reference trajectory for the aircraft's state vector, and the block $MPC_{reconfig}$ which is an MPC controller whose objective is to track the reference trajectory, using the output of *'FDI'* to update its internal model, constraints, etc. The pilot gives commands to the reference model and the goal of the controller is to cause the plane to track the resultant trajectory. At each time step, the MPC controller chooses an input sequence which minimises the difference between the predicted future trajectory, given by the reference model under the assumption that the pilot's inputs are constant over the prediction horizon, and the predicted trajectory of the aircraft, computed from the FDI model. A very similar concept has previously been proposed by (Pachter *et al.*, 1995), and a less similar 'model-matching' concept has been proposed by (Yu and Jonckheere, 1999).

We propose the use of a model-following MPC controller in this application for several reasons. As stated above, MPC is a good framework for fault-tolerant control, as many kinds of aircraft failures can be handled online in an adaptive fashion via modifications to the internal model. The achievable performance of an aircraft will often be reduced after a failure. This too can be dealt with via MPC through changes to the objective function or through the use of a multi-objective formulation as discussed in, for example (Kerrigan and Maciejowski, 2002). Finally, the loss of some actuators will often require the remaining controls to be driven to their limits, requiring any fault-tolerant scheme to deal with actuator constraints which, again, can be handled naturally in MPC.

There are three key benefits of this formulation. First, it is unlikely that any fault-tolerant method which takes full control of the aircraft during an emergency would be used while there is a pilot available to fly. Therefore, keeping the pilot in the loop is desirable in any fault-tolerant flight control scheme. Second, the controller benefits from the heuristic knowledge and experience of the pilot. Finally, the model-following formulation implies that the damaged plane will respond to pilot input, if possible, in a way similar to that for which he was trained. This behaviour will reduce the need for pilots to be trained for particular failure situations and will free them from learning the effects of a failure on the aircraft and trying to compensate for the dynamics of a faulty plane during an emergency. The assumption behind our approach is that there are too many possible failure scenarios for pilots to be trained explicitly for each of them.

## 4. EL AL FLIGHT 1862

### *4.1 Scenario*

On 4 October 1992, a Boeing 747-200F freighter flying out of Schiphol Airport in Amsterdam suffered separation of both engines from the starboard wing. Despite this failure, the crew continued flying for almost 15 minutes, giving considerable time for identification of the failure and for the online design of a new controller. At the end of that period, however, the crew lost control, and the aircraft crashed into an apartment building, causing considerable loss of life. This particular incident is a good test case for studying fault-tolerant control for several reasons. First, the plane was clearly controllable, as the crew

stayed in the air for almost 15 minutes after the failure. Second, despite the fact that considerable time was available, current emergency procedures could not handle the situation as the plane did eventually crash. Finally, there was sufficient damage to the plane to make it extremely difficult to fly, thus making it a challenging and interesting problem. A detailed 6 degree-of-freedom nonlinear Simulink model has been developed by M.H.Smaili, who also showed by careful analysis of the recovered flight data that it was possible to fly the aircraft beyond the point at which the crew lost control. All details of the incident, and the simulation model used in this section, are taken from (Smaili, 1997).

Modern aircraft are designed to be controllable despite multiple single-wing engine failures without separation. However, due to the damage to the right wing caused by the separation of the engines, there was a significant loss of both lateral and directional control:

- Right wing leading edge severely damaged
- Right wing leading edge flaps partially lost
- Right outboard aileron floating
- No outboard aileron available, caused by outboard trailing edge flaps failure
- Loss of six out of ten spoilers
- Unusually long lagging behind of the lower rudder (for unknown reasons)
- Reduced effectiveness of right-hand inboard aileron, because of disturbed airflow caused by right-wing damage and loss of pylon no. 3
- Engines no. 1 and 2 at high thrust settings

### *4.2 Simulation*

For the purposes of demonstration, a 'pilot' is needed to control the plane and give wheel, pedal, throttle and column inputs to the fault-tolerant controller. A second MPC controller is used for this purpose, which has the reference plane as its internal model, in order to track forward velocity ($v$), sideslip ($\beta$), flight path angle ($\gamma$), and the heading derivative ($\dot{\chi}$). The second MPC controller is shown as the block labelled $MPC_{pilot}$ in Figure 1.

The Boeing 747 has a flight control system which maps the pilot's commands in the form of throttle, wheel, rudder and column inputs to control inputs, which consist of: four engine throttles, two inboard and two outboard ailerons, ten spoilers, two inboard and two outboard elevators and the lower and upper rudders. In order to make the actuators independently accessible by the fault-tolerant controller this flight control system has been removed from the simulation, and has been replaced by the fault-tolerant controller shown in Figure 1. The reader is referred to (Jones, 2002) for details.

The simulations presented in this paper attempt to replicate the conditions during Flight El Al 1862 shortly before the pilot lost control: the aircraft is initially flying at an altitude of $1500m$, a speed of $128m/s$ and an angle of attack of $8°$. The test reference trajectory is similar to that which was required for Flight El Al 1862 to line up with the runway, including both right and left turns which test the ability of the controller to handle the asymmetric nature of the damaged aeroplane.

### *4.3 Fault-tolerant MPC design*

The fault-tolerant MPC controller (inner loop in Figure 1) runs at a frequency of 10 Hz, which is chosen to be faster than the fastest mode of the Boeing 747. The 'pilot emulator' MPC (outer loop in Figure 1) runs at 1 Hz. The MPC prediction horizons were chosen experimentally to be 15 samples for the outer loop and 12 for the inner, while the control horizons were chosen to be 15 and five for the outer and inner loops, respectively. The outer loop is tuned such that it will track changes in the references $\gamma$ and $\dot{\chi}$ while maintaining the velocity $v$ to within $10m/s$ of a setpoint and holding the sideslip angle $\beta$ at zero. The pilot's inputs are bounded by their physical limitations and a rate limit of $\frac{1}{2}$ their full range per second. The inner loop is tuned such that it will track the velocity ($v$,$\alpha$,$\beta$) and the orientation ($\phi$,$\theta$,$\psi$) of the reference model. Actuator inputs are constrained in both magnitude and rate by their physical limitations (in the failed condition). Tuning of the fault-tolerant controller was specific to this scenario and no claim is made that the controller would work for all failures.

### *4.4 Results*

Simulations were carried out on both the working and failed nonlinear models using the same controller. The results of the simulations are shown in Figures 2, 3, and 4. Due to space constraints, only the lateral channels are shown here, although the model and controller have six degrees of freedom, and control of this asymmetric aircraft requires coordination between the lateral and longitudinal controls. The full results are available in (Jones, 2002).

Figure 3 displays the ability of the pilot to control the plane and track the trajectory required to lineup with the runway. One can see that the pilot has sufficient authority to fly the same path in the failed aeroplane as with the working one. Figure 4 shows that the pilot is able to control

the damaged aircraft in a manner very similar to how he flew the nominal craft through the same manoeuvres; although there is an offset of both wheel and pedals after the failure, the patterns of movements of these pilot inputs are the same as for the working aircraft. This indicates that the reference model was being successfully tracked by the fault-tolerant controller and that the pilot did not need to take any special action in order to compensate for the damage to the aircraft. Finally, Figure 2 shows that during the flight on the damaged plane, the controller has used the available actuators in a drastically different fashion from the working case, despite achieving a similar result. Note also that one of the engines and one of the rudders are at their physical limits, indicating that the ability of MPC to handle constraints was necessary in this example.

Starting from the same initial conditions as in the real incident about 5 minutes into the flight, our controller performs similar manoeuvres to those performed by the real aircraft, but avoids the condition which led to the fatal crash, and successfully descends to ground level — though we have not simulated the final landing flare.

## 5. CONCLUSIONS

We do not claim to have solved the problem of fault-tolerant flight control. But we do believe that the results presented in this paper are sufficiently good and non-trivial to indicate that fault-tolerant control on the basis of MPC as an 'implementation architecture' is a plausible proposition.

Major problems remain to be solved before the concept can be considered to be proved. The first is that considerable tuning of the (inner loop) MPC controller was necessary in order to obtain the results shown in the simulation. Either a faster way of tuning MPC must be found, or a set of controller parameters must be found which is generic for a large number of fault conditions (within the restricted domain of flight control, and possibly restricted to a single aircraft type).

Another potential problem is that of choosing objectives for the MPC controller. If the fault/damage is so serious that the original objectives cannot be met, then a fault-tolerant controller must try to achieve less ambitious objectives. How to detect that condition, and how to decide what the revised objectives should be, are questions to which the answers are currently unknown. An advantage of our model-following formulation is that the MPC controller's objectives relate only to following the reference model. This allows the pilot to use experience and intuition to moderate the demands made on the aircraft.

An even more fundamental problem is the requirement for identification of the effects of a failure on the aircraft behaviour. Aircraft are comprehensively instrumented, and in the case of El Al 1862 there was sufficient time to collect plenty of data. Furthermore, intuitively it does not appear to be necessary for the new aircraft behaviour to be known accurately; even crude information on some aspects of behaviour might be sufficient. For example, even after flying the damaged aircraft for 15 minutes the pilot had not realised that he had lost both engines on one side of the aircraft, although he knew that these engines were not producing any thrust (Smaili, 1997). This had two effects: firstly, the crew thought that the engines were probably on fire, and that they therefore had to land as quickly as possible, and this might have led them to hasty decisions. Secondly, they did not consider the possibility that the wing's aerodynamic performance might have changed as a result of damage. Knowing how much information about behaviour is needed for effective control, and how quickly that information can be acquired, are topics on which very little is known at present, and which deserve further study. (The existing literature on fault detection and identification — such as (Basseville and Nikiforov, 1993) — does not address such questions.) We note, however, that there was enough data available on-board to enable (Smaili, 1997) to derive a model of the failed aircraft, albeit not in real time.

## 6. ACKNOWLEDGEMENTS

We would like to thank M.H.Smaili, of the Dutch Aerospace Research Laboratory (NLR), and the University of Leicester, for making the *Simulink* nonlinear simulation model of El Al 1862 available to us.

## REFERENCES

Basseville, M. and I.V. Nikiforov (1993). *Detection of Abrupt Changes*. Prentice-Hall. New York.

Camacho, E.F. and C. Bordons (1999). *Model Predictive Control*. Springer. London.

Civ (2000). Civil Aviation Authority, Aviation Saftey Review 1990-1999.

Hughes, D. and M.A. Dornheim (1989). United DC-10 crashes in Sioux City, Iowa. *Aviation Week and Space Technology* **131**(4), 96–97.

Jones, C.N. (2002). Reconfigurable flight control. Technical report. Engineering Dept, University of Cambridge, UK, `www-control.eng.cam.ac.uk/ cnj22/ docs/yearone.pdf`.

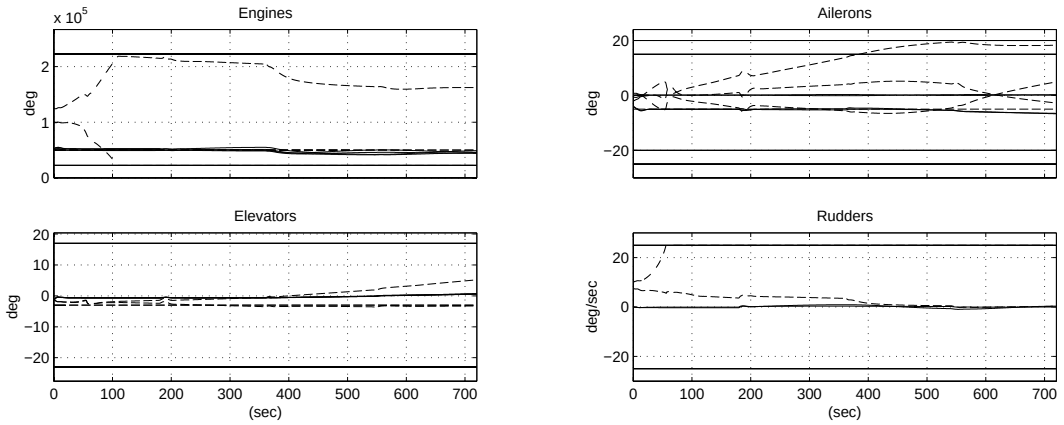


Fig. 2. Actuator Positions; Solid: Working Aircraft, Dashed: Failed Aircraft


Kerrigan, E. C. and J. M. Maciejowski (2002). Designing model predictive controllers with prioritised constraints and objectives. In: *Proceedings of the IEEE Conference on Computer Aided Control System Design*. Glasgow, Scotland.

Maciejowski, J.M. (1997*a*). Modelling and predictive control: enabling technologies for reconfiguration. In: *IFAC Conference on System Structure and Control*. Bucharest. Reprinted in Annual Control Reviews, vol.23, 1999, pp.13–23.

Maciejowski, J.M. (1997*b*). Reconfiguring control systems by optimization. In: *European Control Conference ECC*. Brussels.

Maciejowski, J.M. (1998). The implicit daisy-chaining property of constrained predictive control. *Applied Mathematics and Computer Science* **8**(4), 101–117.

Maciejowski, J.M. (2002). *Predictive Control with Constraints*. Prentice-Hall. Harlow, England.

Pachter, M., P. Chandler and M.J. Mears (1995). Reconfigurable tracking control with saturation. *AIAA Journal of Guidance, Control and Dynamics*.

Patton, R.J. (1997). Fault tolerant control: the 1997 situation. In: *Proc. IFAC Safeprocess Conf.*. Hull, UK. pp. 1033–1054.

Smaili, M.H. (1997). Flight data reconstruction and simulation of El Al Flight 1862. Master's thesis. Technical University Delft.

Wang, L.Y. and G. Zames (1991). Local-global double algebras for slow $H^\infty$ adaptation: part II — optimization of stable plants. *IEEE Trans. Automatic Control* **36**(2), 143–151.

Yu, G-R. and E.A. Jonckheere (1999). Propulsion control of crippled aircraft by $H_\infty$ model-matching. *IEEE Trans. Control Systems Technology* **7**(6), 142–159.


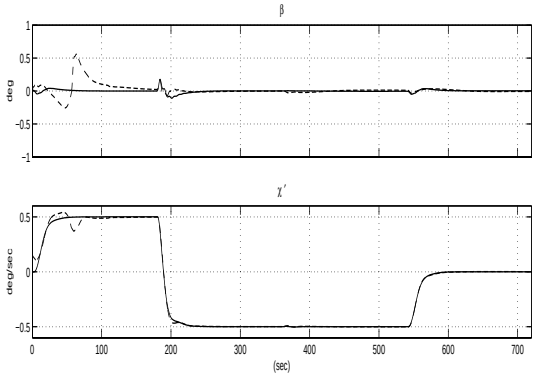


Fig. 3. Lateral Tracking Performance; Solid: Working Aircraft, Dashed: Failed Aircraft

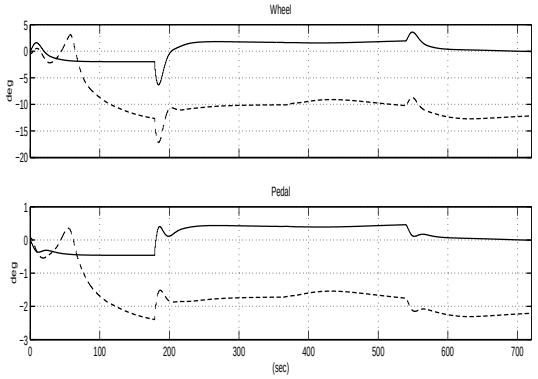


Fig. 4. Lateral Pilot Inputs; Solid: Working Aircraft, Dashed: Failed Aircraft