Chapter 2
Fault Tolerant Flight Control - A Survey

Jan Maciejowski, and Hafid Smail

2.1 Why Fault Tolerant Control?

Nowadays, control systems are involved in nearly all aspects of our lives. They are all around us, but their presence is not always really apparent. They are in our kitchens, in our DVD-players, computers and our cars. They are found in elevators, ships, aircraft and spacecraft. Control systems are present in every industry, they are used to control chemical reactors, distillation columns, and nuclear power plants.

Michel Verhaegen
Delft University of Technology, Delft Center for Systems and Control,
Mekelweg 2, 2628CD Delft, The Netherlands
e-mail: m.verhaegen@moesp.org

Stoyan Kanev
ECN Wind Energy, P.O.Box 1, 1755ZG Petten, The Netherlands
e-mail: kanev@ecn.nl

Redouane Hallouzi
ReliaCon, Rotterdamseweg 145, 2628AL Delft, The Netherlands
e-mail: hallouzi@reliacon.nl

Colin Jones
ETH Zurich, Automatic Control Laboratory ETL K14.2,
Physikstrasse 38092 Zurich, Switzerland
e-mail: cjones@control.ee.ethz.ch

Jan Maciejowski
University of Cambridge, Engineering Department, Trumpington Street,
Cambridge CB2 1PZ, United Kingdom
e-mail: jmm@eng.cam.ac.uk

Hafid Smaili
National Aerospace Laboratory NLR, Anthony Fokkerweg 2, 1059 CM Amsterdam,
The Netherlands
e-mail: smaili@nlr.nl

They are constantly and inexhaustibly working, making our life more comfortable and more efficient . . . until the system fails.

Faults in technological systems are events that happen rarely, and come mostly unexpectedly. In [43] the following definition for a fault is made:

> A **fault** is an unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable/usual/standard condition.

Faults are difficult to accurately predict in time, and to prevent. The impact of a fault can be a small reduction in efficiency, but could also lead to overall system failure. In safety critical systems this can lead to catastrophic events with significant costs, both economically and in terms of human life. Several such examples are

- the explosion at the nuclear power plant at Chernobyl, Ukraine, on 26th April 1986 [67]. About 30 people were killed immediately, while another 15,000 were killed and 50,000 left handicapped in the emergency clean-up after the accident. It is estimated that five million people were exposed to radiation in Ukraine, Belarus and Russia.
- the crash of the AMERICAN AIRLINES flight 191, a McDonnell-Douglas DC-10 aircraft, at Chicago O'Hare International Airport on 25 May 1979 (see Chapter 1). In this incident 271 persons on board and 2 on the ground were killed when the aircraft crashed into an open field [74, 75].
- the explosion of the Ariane 5 rocket on 4th June 1996, where the reason was a fault in the Internal Reference Unit that had the task to provide the control system with altitude and trajectory information. As a result, incorrect altitude information was delivered to the control unit [67].

The question that immediately arises is *"Could something have been done to prevent these disasters?"*. While in most situations the occurrences of faults in the systems cannot be prevented, subsequent analysis often reveals that *the consequences of the faults could be avoided* or, at least, that their severity (in terms of economic losses, casualties, etc.) could be minimized. If faults could be detected and diagnosed rapidly enough, then, in many cases, it is possible to subsequently reconfigure the control system so that it can safely continue its operation (though with degraded performance) until the time comes when it can be switched off to allow repair. In order to minimize the chances for such catastrophic events as those summarized above, safety-critical systems must possess the properties of increased reliability and safety.

A way to offer increased reliability and safety is by means of a fault-tolerant control (FTC) system design. An FTC system could have been designed to lead to a safe shutdown of the Chernobyl reactor way before it exploded [67]. Subsequent studies following the McDonnell-Douglas DC-10 crash showed that the crash could have been avoided [75]. In the last minutes of the Ariane 5 crash the normal altitude information had been replaced by some diagnostic information that the control system was not designed to understand [67]. Fortunately, there are also examples,

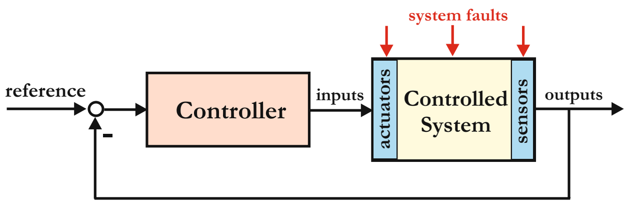

Fig. 2.1 According to their location, faults are classified into sensor, actuator and component faults.

which show that taking appropriate measures can indeed prevent disasters (see also Chapter 1):

1. A McDonnell-Douglas DC-10 aircraft executing flight 232 of UNITED AIRLINES from Denver to Minneapolis experienced a disastrous failure in the hydraulic lines that left the plane without any control surfaces at 37,000 ft. The crew then improvised a control strategy that used only the throttles of the two wing engines and managed to successfully crash-land the plane in Sioux City, Iowa, saving the lives of 184 out of the 296 passengers on board [66].
2. In the DELTA AIRLINES flight 1080 an elevator became jammed at 19 degrees. The pilot was not given any indication of what had actually occurred but still was able to reconfigure the remaining lateral control elements to land the aircraft safely [75].

All these examples clearly motivate the need for increased fault-tolerance in order to improve to the maximum possible extent the safety, reliability and availability of controlled systems. This is particularly true as modern systems become increasingly complex. The examples above also explain the large amount of research in the field of fault detection, diagnosis and fault-tolerant control. An overview of this research is provided in this chapter.

2.2 Fault Classification

Faults are events that can take place in different parts of the controlled system. In the FTC literature faults are classified according to their location of occurrence in the system (see Figure 2.1).

Actuator faults: they represent *partial* or *total* (complete) loss of control action. An example of a completely lost actuator is a “stuck” actuator that produces no (controllable) actuation regardless of the input applied to it. Total actuator faults can occur, for instance, as a result of a breakage, cut or burned wiring, short circuits, or the presence of a foreign body in the actuator. Partially failed actuators produce only a part of the normal (i.e. under nominal operating conditions) actuation. This can result from hydraulic or pneumatic leakage, increased resistance or a fall in the supply voltage, etc. Duplicating the actuators in the system in order to achieve increased fault-tolerance is often not an option due to their high prices and large size and mass.

Sensor faults: these faults represent incorrect readings from the sensors that the system is equipped with. Sensor faults can also be subdivided into *partial* and *total*. Total sensor faults produce information that is not related to the value of the measured physical parameter. They can be due to broken wires, lost contact with the surface, etc. Partial sensor faults produce readings that are related to the measured signal in such a way that useful information could still be retrieved. This can, for instance, be a gain reduction so that a scaled version of the signal is measured, a biased measurement resulting in a (usually constant) offset in the reading, or increased noise. Due to their smaller sizes sensors can be duplicated in the system to increase fault tolerance. For instance, by using three sensors to measure the same variable one may consider it reliable enough to compare the readings from the sensors to detect faults in (one and only one) of them. The so-called "majority voting" method can then be used to pinpoint the faulty sensor. This approach usually implies significant increases in the related costs.

Component faults: these are faults in the components of the plant itself, i.e. all faults that cannot be categorized as sensor or actuator faults will be referred to as component faults. These faults represent changes in the physical parameters of the system, e.g. mass, aerodynamic coefficients, damping constant, etc., that are often due to structural damage. They often result in a change in the dynamical behaviour of the controlled system. Due to their diversity, component faults cover a very wide class of (unanticipated) situations, and as such are the most difficult ones to deal with.

Further, with respect to the way faults are modelled, they are classified as *additive* and *multiplicative*, as depicted in Figure 2.2. Additive faults are suitable for representing component faults in the system, while sensor and actuator faults are in practice most often multiplicative by nature.

Faults are also classified according to their time characteristics (see Figure 2.3) as *abrupt*, *incipient* and *intermittent*. Abrupt faults occur instantaneously often as a result of hardware damage. They can be very severe since, if they affect the performance and/or the stability of the controlled system, prompt reaction from the FTC system is required. Incipient faults represent slow parametric changes, often as a result of aging. They are more difficult to detect due to their slow time characteristics,

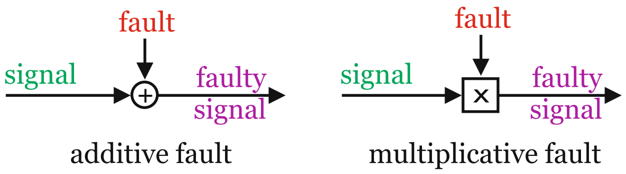

Fig. 2.2 According to their representation, faults are divided into additive and multiplicative.

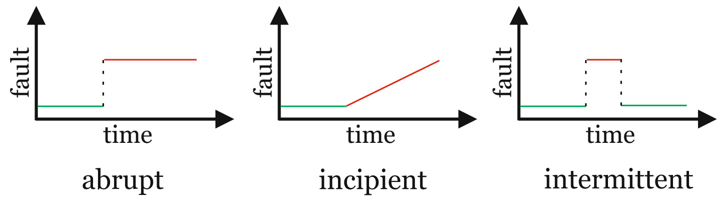

Fig. 2.3 With respect to their time characteristics faults can be abrupt, incipient and intermittent.

but are also less severe. Finally, intermittent faults are faults that appear and disappear repeatedly, for instance due to partially damaged wiring.

2.3 Modelling Faults

As already mentioned in Section 2.2, faults are often represented as additive or multiplicative adjustments to the nominal behaviour. In this section we further concentrate on the mathematical representation of these faults and will provide a discussion on when and why one representation is more appropriate than the other.

Throughout this chapter the state-space representation of dynamical systems is used, so that the relation from the system inputs $u \in \mathbb{R}^m$ to the measured outputs $y \in \mathbb{R}^p$ is written in the form

$$S_{nom} : \begin{cases} x_{k+1} = Ax_k + Bu_k \\ y_k = Cx_k + Du_k, \end{cases} \tag{2.1}$$

where $x_k \in \mathbb{R}^n$ denotes the state of the system at time instance k, and A, B, C and D are matrices (possibly time-varying) of appropriate dimension.

2.3.1 Multiplicative Faults

Multiplicative modelling is mostly used to represent sensor and actuator faults.

Actuator faults represent malfunctioning of the actuators of the system, for example as a result of hydraulic leakages, broken wires, or stuck control surfaces in an aircraft. Such faults can be modelled as an abrupt change of the nominal control action from u_k to

$$u_k^f = u_k + (I - \Sigma_A)(\bar{u} - u_k), \tag{2.2}$$

where $\bar{u} \in \mathbb{R}^m$ is a (not necessarily constant) vector that cannot be manipulated, and where

$$\Sigma_A = \mathbf{diag}\{\left[\,\sigma_1^a,\ \sigma_2^a,\ \ldots,\ \sigma_m^a\,\right]\},\ \sigma_i^a \in \mathbb{R}.$$

In this way $\sigma_i^a = 0$ represents a total fault (i.e a complete failure) of the i-th actuator of the system so that the control action coming from this i-th actuator becomes equal to the i-th element of the uncontrollable offset vector $\bar{u}$, i.e. $u_k^f(i) = \bar{u}(i)$. On the other hand, $\sigma_i^a = 1$ implies that the i-th actuator operates normally ($u_k^f(i) = u(i)$). The quantities σ_i^a, $i = 1, 2, \ldots, m$ can also take values in between 0 and 1, making it possible to represent partial actuator faults. Substituting the nominal control action u_k in equation (2.1) with the faulty u_k^f results in the following state-space model

$$S_{mult,af} : \begin{cases} x_{k+1} = Ax_k + B\Sigma_A u_k + B(I - \Sigma_A)\bar{u} \\ \quad y_k = Cx_k + D\Sigma_A u_k + D(I - \Sigma_A)\bar{u}. \end{cases} \tag{2.3}$$

Models in the form (2.3) are referred to as multiplicative fault models and have been widely used in the literature (see, for example [86, 73]).

It needs to be noted that while such multiplicative actuator faults do not directly affect the dynamics of the controlled system itself, they can significantly affect the dynamics of the closed-loop system, and may even affect the controllability of the system. Figure 2.4 presents a simple example with a 50% actuator fault that results in instability of the closed-loop system. In the example of Figure 2.4 a system consisting of the transfer function $S(s) = 1/(s-1)$ is controlled by a PI controller with transfer function $C(s) = 1.5 + \frac{5}{s}$, so that a sinusoidal reference signal is tracked under normal operating conditions (i.e. during the first 20 seconds of the simulation). At time instance $t = 20$ sec, a 50% loss of control effectiveness is introduced and as a result the closed-loop system stability is lost. This example makes it clear that even "seemingly simple" faults may significantly degrade the performance and can even destabilize the system.

Similarly, sensor faults occurring in the system (2.1) represent incorrect reading from the sensors, so that as a result the real output of the system y_k^{real} differs from the variable being measured. Multiplicative sensor faults can be modelled in the following way

$$y_k^f = y_k + (I - \Sigma_S)(\bar{y} - y_k), \tag{2.4}$$

where $\bar{y} \in \mathbb{R}^p$ is an offset vector, and

$$\Sigma_S = \mathbf{diag}\{\,[\,\sigma_1^s, \ \ldots, \ \sigma_p^s\,]\,\}, \ \sigma_i^s \in \mathbb{R},$$

so that $\sigma_j^s = 0$ represents a total fault of the j-th sensor, and $\sigma_j^s = 1$ models the normal mode of operation of the j-th sensor. Partial faults are then modelled by taking $\sigma_j^s \in (0,\ 1)$. Substitution of the nominal measurement y_k in (2.1) with its faulty counterpart y_k^f results in the following state-space model that represents multiplicative sensor faults

$$S_{mult,sf} : \begin{cases} x_{k+1} = Ax_k + Bu_k \\ \quad y_k = \Sigma_S Cx_k + \Sigma_S Du_k + (I - \Sigma_S)\bar{y}. \end{cases} \tag{2.5}$$

In this way, combinations of multiplicative sensor and actuator faults are represented in the following way

$$S_{mult} : \begin{cases} x_{k+1} = Ax_k + B\Sigma_A u_k + b(\Sigma_A, \bar{u}) \\ \quad y_k = \Sigma_S Cx_k + \Sigma_S D\Sigma_A u_k + d(\Sigma_A, \Sigma_S, \bar{u}, \bar{y}), \end{cases} \tag{2.6}$$

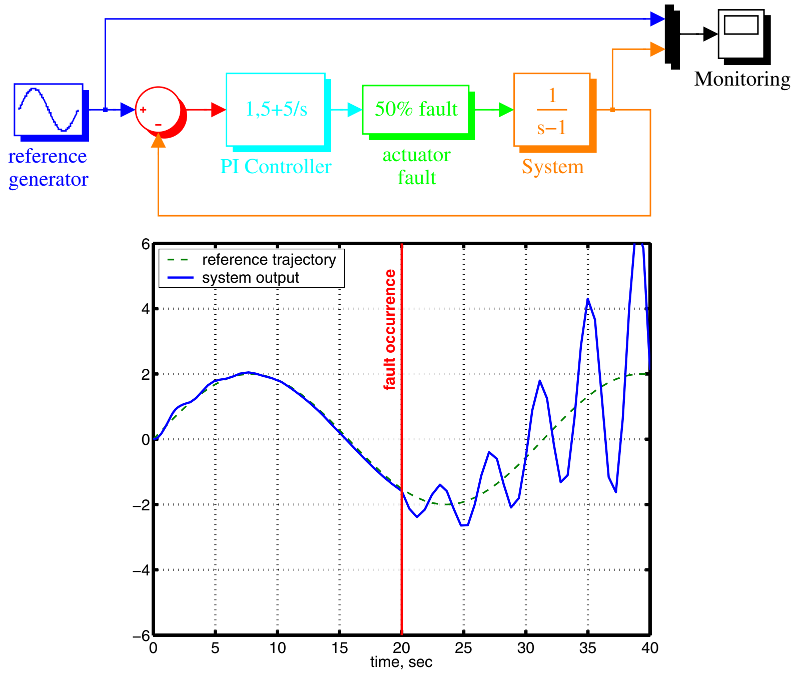

Fig. 2.4 After a multiplicative fault the system may become unstable if no reconfiguration takes place.

with

$$\begin{aligned} b(\Sigma_A, \bar{u}) &= B(I - \Sigma_A)\bar{u}, \\ d(\Sigma_A, \Sigma_S, \bar{u}, \bar{y}) &= \Sigma_S D(I - \Sigma_A)\bar{u} + (I - \Sigma_S)\bar{y}. \end{aligned}$$

The multiplicative model is thus a "natural" way to model a wide variety of sensor and actuator faults, but cannot be used to represent more general component faults. This fault model representation is most often used in the design of the controller reconfiguration scheme of an active FTC system since for controller redesign one usually needs the state-space matrices of the faulty system.

2.3.2 Additive Faults

The additive faults representation is more general than the multiplicative one. A state-space model with additive faults has the form

$$S_{add} : \begin{cases} x_{k+1} = Ax_k + Bu_k + Ff_k \\ y_k = Cx_k + Du_k + Ef_k, \end{cases} \tag{2.7}$$

where $f_k \in \mathbb{R}^{n_f}$ is a signal describing the faults. This representation may, in principle, be used to model a wide class of faults, including sensor, actuator, and

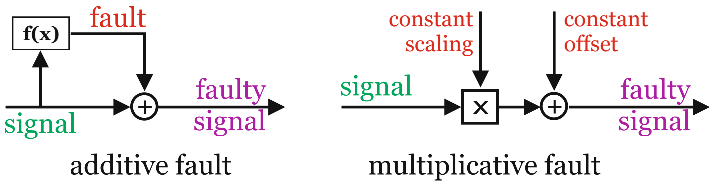

Fig. 2.5 Using additive fault representation to model total sensor (or actuator) faults results in a fault signal that depends on y_k (u_k). This is not the case with the multiplicative model where the fault magnitude and the offset are independent on the signals in the state-space model.

component faults. Using model (2.7), however, often results in the signal f_k becoming related to one or more of the signals u_k, y_k and x_k. For instance, when using this additive fault representation to model a total fault in all actuators ($\Sigma_A = 0$ and $\bar{u} = 0$ in equation (2.2)) then in order to make model (2.7) equivalent to model (2.3) one needs to take a signal f_k such that $\begin{bmatrix} F \\ E \end{bmatrix} f_k = -\begin{bmatrix} B \\ D \end{bmatrix} u_k$ holds, making f_k dependent on u_k. Clearly, the fault signal being a function of the control action is not desirable for controller design. On the other hand, f_k is independent of u_k when multiplicative representation is utilized. Figure 2.5 illustrates this.

Another disadvantage of the additive model when used to represent sensor and actuator faults is that, in terms of input-output relationships, these two faults become difficult to distinguish. Indeed, suppose that the model

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + f_k^a \\ y_k &= Cx_k + Du_k + f_k^s, \end{aligned}$$

is used to represent faults in the sensors and actuators. By writing the corresponding transfer function

$$y(z) = (C(zI - A)^{-1}B + D)u_k + C(zI - A)^{-1}f_k^a + f_k^s,$$

it becomes clear that the effect of an actuator fault on the output of the system can be modelled not only by the signal f_k^a, but also by f_k^s.

An advantage is, as already mentioned, that the additive representation can be used to model a more general class of faults than multiplicative ones. In addition, it is more suitable for the design of FDD schemes because the faults are represented by one *signal* rather than by changes in the state-space matrices of the system as is the case with the multiplicative representation. For that reason the majority of FDD methods are focused on additive faults [33, 3, 57].

2.3.3 Component Faults

The class of component faults was defined in Section 2.2 as the most general as it includes faults that may bring changes in practically any element of the system. It was defined as the class of all faults that cannot be classified as sensor or actuator faults. A component fault may introduce changes in each matrix of the state-space representation of the system due to the fact they may all depend on the same physical parameter that undergoes a change. Component faults are often modelled in the form of a linear parameter-varying (LPV) system

$$\begin{aligned} x_{k+1} &= A(f)x_k + B(f)u_k \\ y_k &= C(f)x_k + D(f)u_k, \end{aligned} \tag{2.8}$$

where $f \in \mathbb{R}^{n_f}$ is a parameter vector representing the component faults. It should be noted that this model might also be used for modelling sensor and actuator faults. Due to the fact the matrices may depend in a general, nonlinear, way on the fault signal f_k this model is less suitable for fault detection and diagnosis.

2.4 Main Components in an FTC System

FTC systems are generally divided into two classes: *passive* and *active*. Passive FTC systems are based on robust controller design techniques and aim at synthesizing a single, robust controller that makes the closed-loop system insensitive to anticipated faults. This approach requires no online detection of the faults, and is therefore computationally more attractive. Its applicability, however, is very restricted due to its serious disadvantages:

- In order to achieve robustness to faults, usually a very restricted subset of the possible faults can be considered; often only faults that have a "small effect" on the behaviour of the system can be treated in this way.
- Achieving increased robustness to certain faults is only possible at the expense of decreased nominal performance. Since faults are effects that happen very rarely it is not reasonable to significantly degrade the fault-free performance of the system only to achieve some insensitivity to a restricted class of faults.

However, using passive FTC systems can also have its advantages. One advantage is that a fixed controller has relatively modest hardware and software requirements. Another advantage is that passive FTC systems, due to their lower complexity compared to active FTC systems, can be made more reliable according to classical reliability theory [84]. Examples of passive FTC systems can be found in [61, 72, 97].

As opposed to passive methods, the *active* approach to the design of FTC systems is based on controller redesign, or selection/mixing of predesigned controllers. This technique usually requires a fault detection and diagnosis (FDD) scheme that has the task of detecting and localizing the faults if they occur in the system. The structure of an active FDD-based FTC system is presented in Figure 2.6. The FDD part uses input-output measurement from the system to detect and localize the faults. The estimated faults are subsequently passed to a reconfiguration mechanism (RM) that changes the parameters and/or the structure of the controller in order to achieve an acceptable post-fault system performance.

Depending on the way the post-fault controller is formed, active FTC methods are further subdivided into *projection-based* methods and *on-line redesign* methods.

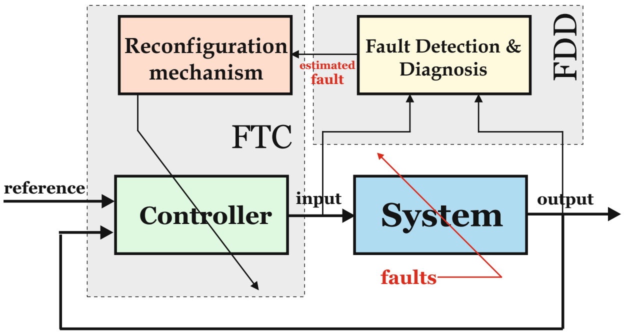

Fig. 2.6 Main components of an active FTC system.

The projection based methods rely on the controller selection from a set of off-line predesigned controllers. Usually each controller from the set is designed for a particular fault situation and is switched on by the RM whenever the corresponding fault pattern has been diagnosed by the FDD scheme. In this way only a restricted, finite class of faults can be treated. The on-line redesign methods involve on-line computation of the controller parameters, referred to as *reconfigurable control*, or recalculation of both the structure and the parameters of the controller, called *restructurable control*. Comparing the achievable post-fault system performances, the on-line redesign method is superior to the passive method and the off-line projection-based method. However, it is computationally the most expensive method as it often boils down to on-line optimization.

There are a number of important issues when designing active FTC systems. Probably the most significant one is the *integration between the FDD part and the FTC* part. The majority of approaches in the literature are focused on one of these two parts by either considering the absence of the other or assuming that it is perfect. To be more specific, many FDD algorithms do not consider the closed-loop operation of the system and, conversely, many FTC methods assume the availability of perfect fault estimates from the FDD scheme. The interconnection of such methods is potentially infeasible and there can be no guarantees that a satisfactory post-fault performance, or even stability, can be maintained by such a scheme. It is therefore very important that the designs of the FDD and FTC, when carried out separately, are each performed bearing in mind the presence and imperfections of the other. For making the interconnection possible, one should first investigate what information from the FDD is needed by the FTC, as well as what information can actually be provided by the FDD scheme. Imprecise information from the FDD that is incorrectly interpreted by the FTC scheme might lead to a complete loss of stability of the system.

The usual situation in practice is that after the occurrence of a fault in the system there is initially not enough information in terms of input/output measurements from the system to make it possible for the FDD scheme to diagnose the fault. For this reason, only after some time elapses and more information becomes available can the FDD scheme detect that a fault has occurred. Even more time is required to localize the fault and its magnitude. As a result, the information that is provided to the FTC part is initially more imprecise (i.e. with larger uncertainty), and it gets more and more accurate (with less uncertainty) as more data becomes available from the system. The FTC scheme should be able to deal with such situations. Therefore, the FTC should necessarily be capable of dealing with *uncertainty in the FDD* information/estimates, and should perform satisfactorily (guaranteeing at least the stability) during the *transition period* that the FDD scheme needs to diagnose the fault(s).

Very often the dynamics of real physical systems cannot be represented accurately enough by linear dynamical models so that *nonlinear models* have to be used. This necessitates the development of techniques for FTC system design that can explicitly deal with nonlinearities in the mathematical representation of the system. Nonlinearities are, in fact, very often encountered in the representations of complex safety-critical controlled systems like aircraft and spacecraft. To reduce the inherent complexity of the control design, it is usual that the lateral and longitudinal dynamics of an aircraft are decoupled so that they have no effect on each other. This significantly simplifies the model of the aircraft and makes it possible to design the corresponding controllers independently. This decoupling condition can approximately be achieved for a healthy aircraft, but certain faults can easily destroy it, so that the two controllers could not be considered separately.

An important issue in FTC system design is that even for a fixed operating region, where a nonlinear system allows approximation by a linear model, it is very difficult to obtain an accurate linear representation, either due to the fact that the physical parameters in the nonlinear model are not exactly known or because they vary with time. Even the nonlinear model is often derived after some simplifying assumptions, so that it only approximates the behaviour of the system. Even more, this uncertainty is further increased due to the linearization that basically consists in truncating second and higher order terms in the Taylor series expansion of the nonlinear function. As a result only a representation with *uncertainty* is available. It is important that the FTC system is designed to be robust to such uncertainties within the model.

Another very important issue is that every real-life controlled system has *control action saturation*, i.e. the input and/or output signals cannot exceed certain values. In the design phase of a control system usually the effect of the saturation is accommodated by making sure that the control action will not get overly active and will remain inside the saturation limits under normal operating conditions. Faults, however, can have the effect that the control action stays at the saturation limit. For instance, when a partial 50% loss of effectiveness in an actuator has been diagnosed, a standard and easy way to accommodate the fault is to re-scale the control action by two so that the resulting actuation approximates the fault-free actuation. As a result the control action becomes twice as big and may go to the saturation limits. Clearly, in such situations one should not try to completely accommodate the fault but one should be willing to accept certain performance degradation imposed by the saturation. In other words, a trade-off between achievable performance and available actuator capability might need to be made after the occurrence of a fault. This situation is often referred to as *graceful performance degradation* [95].

2.5 FTC Problem Formulation

The dynamics of a real-life physical system can be represented in state-space in the following general form

$$S(p_k): \begin{cases} x_{k+1} = f(x_k, u_k, p_k), \\ y_k = h(x_k, u_k, p_k), \\ x_0 = \hat{x}_0, \end{cases} \tag{2.9}$$

where the vector $x_k \in \mathscr{X} \subseteq \mathbb{R}^n$ represents the state of the system $S(p_k)$, $u_k \in \mathscr{U} \subseteq \mathbb{R}^{m+n_\xi}$ represents the inputs to the system, $y_k \in \mathbb{R}^{p+n_z}$ denotes the outputs of the system. At each time instance t the system $S(p_k)$ is parameterized by a (possibly unknown) parameter vector $p_k \in \mathscr{P} \subseteq \mathbb{R}^{n_p}$. The vector p_k may represent uncertain physical parameters in the system or system faults.

Nonlinear models of systems are in general inconvenient to work with due to their complexity and due to the lack of a well-developed theory for analysis and synthesis for general nonlinear models. The usual strategy to deal with them is either by approximating them with more convenient models (e.g. by means of blending of a set of local linear models as in the multi-model and in the Fuzzy control theories) or by assuming certain structure (e.g. bilinear systems, Hammerstein-Wiener systems, linearity in the input, etc.).

In the multiple model approach the state space $\mathscr{X}$ is divided into N representative and disjoint regions $\mathscr{X}_i$, with $\bigcup_{i=1}^{N} \mathscr{X}_i \equiv \mathscr{X}$, and in each region a point $(x^{(i)}, u^{(i)}) \in \mathscr{X}_i \times \mathscr{U}$ is chosen around which the nonlinear system $S(p_k)$ is approximated by a linear model. Under the assumption that $f(\cdot), g(\cdot) \in C^1$, the local linear approximation $M_i(p_k)$ of the system $S(p_k)$ within the open-ball neighbourhood

$$\mathscr{B}(x^{(i)}, u^{(i)}) = \left\{ (x,u) \in \mathscr{X} \times \mathscr{U} : \left\| \begin{bmatrix} x - x^{(i)} \\ u - u^{(i)} \end{bmatrix} \right\|_2 < \varepsilon \right\},$$

is called the p_k-parameterized local linear model

$$M_i(p_k): \begin{cases} x^{(i)}_{k+1} = A_i(p_k)x^{(i)}_k + B_i(p_k)u_k + b_i(p_k), \\ y^{(i)}_k = C_i(p_k)x^{(i)}_k + D_i(p_k)u_k + c_i(p_k), \\ x^{(i)}_0 = \bar{x}_0, \end{cases}$$

with

$$\begin{aligned} & A_i(p_k) = \partial_x f(x^{(i)}, u^{(i)}, p_k), \;\; B_i(p_k) = \partial_u f(x^{(i)}, u^{(i)}, p_k) \\ & C_i(p_k) = \partial_x h(x^{(i)}, u^{(i)}, p_k), \;\; D_i(p_k) = \partial_u h(x^{(i)}, u^{(i)}, p_k) \\ & b_i(p_k) = f(x^{(i)}, u^{(i)}, p_k) - A(p_k)x^{(i)} - B(p_k)u^{(i)} \\ & c_i(p_k) = h(x^{(i)}, u^{(i)}, p_k) - C(p_k)x^{(i)} - D(p_k)u^{(i)}, \end{aligned}$$

where $\partial_x f$, $\partial_u f$, $\partial_x h$, and $\partial_u h$ represent the partial derivatives of the functions $f(\cdot)$ and $h(\cdot)$ with respect to the vectors x and u.

Each local linear model $M_i(p_k)$ describes the behaviour of the nonlinear system within one regime $\mathscr{X}_i$. A global approximation can then be formed by interpolating the local models using smooth interpolation functions $\phi_i(x_k, u_k, p_k) > 0$ that depend on the operating point (x_k, u_k) as well as on the parameter vector p_k, i.e.

$$\hat{y}_k = \sum_{i=1}^{N} \mu_k^{(i)} y_k^{(i)}, \text{ with } \mu_k^{(i)} = \frac{\phi_i(x_k, u_k, p_k)}{\sum_{i=1}^{N} \phi_i(x_k, u_k, p_k)}. \tag{2.10}$$

Such approximations are widely used in the literature (see, for instance, [47]). In fact it is shown in [46] that, under certain smoothness properties, the nonlinear system $S(p_k)$ can be approximated to any desired accuracy on a compact subset of the state and input spaces by means of the representation (2.10) for a sufficiently large number of local models.

The multiple model representation (2.10) is both intuitive and attractive, and is related to the Takagi-Sugeno fuzzy model, where the weights $\mu_k^{(i)}$ in the linear combination of the local outputs are called degrees of membership.

Suppose that the parameter vector p_k is formed by two vectors, $\delta_k \in \Delta \subseteq \mathbb{R}^{n_\delta}$ and $f_k \in \mathscr{F} \subseteq \mathbb{R}^{n_f}$, so that

$$p_k = \begin{bmatrix} \delta_k \\ f_k \end{bmatrix}, \tag{2.11}$$

where the vector δ_k is used to represent unknown, time-varying physical parameters of the system, and where the vector f_k represents faults in the system. For consistency in terms of dimensions $n_\delta + n_f = n_p$. While both vectors are unknown, the fault vector f_k is assumed to be estimated by an FDD scheme, and its estimate is denoted here as $\hat{f}_k$. Let $\delta_0 \in \Delta$ represent the nominal values of the uncertain parameters, and $f_0 \in \mathscr{F}$ represent the fault-free mode of operation.

Collect all local models $M_i(p_k)$ into a model set

$$\mathscr{M}(p_k) = \{M_1(p_k), M_2(p_k), \ldots, M_N(p_k)\}, \tag{2.12}$$

and consider only one element of the set $\mathscr{M}(p_k)$ which, due to (2.11), is denoted as $M(\delta, f)$. For simplicity of notation, the time symbol is omitted in $M(\delta, f)$.

The following objectives are considered:

- *passive robust FTC:* design one controller K that achieves some desired performance for the model $M(\delta, f)$ for all possible uncertainties $\delta_k \in \Delta$ and faults $f_k \in \mathscr{F}$,
- *active robust FTC:* given an estimate $\hat{f}$ of the fault vector f by some FDD scheme, design a controller $K(\hat{f})$ that achieves some desired performance for the model $M(\delta, f)$ for all possible uncertainties $\delta_k \in \Delta$ and faults $f_k \in \mathscr{F}$,
- *active MM-based FTC:* design a controller that achieves some desired performance for the nonlinear system $S(p_k)$ for some fixed $\delta_k = \delta_0 \in \Delta$ (i.e. in the case of no uncertainty) and for all possible faults $f_k \in \mathscr{F}$.

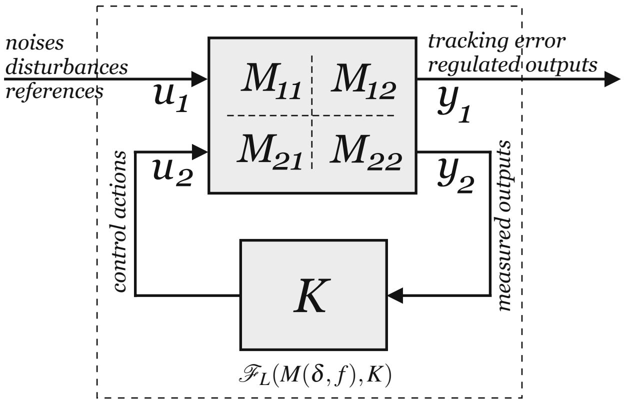

Fig. 2.7 Partitioning of the model $M(\delta, f)$ and forming the closed-loop with the controller K.

A natural continuation of this research activity is to combine the MM-based representation of the nonlinear system with the passive and active approaches to FTC in an attempt to deal with nonlinear systems with uncertainty as in (2.9).

We will next provide some technical insight into the above objectives. Suppose that a continuous map, the *performance index*, is given by

$$J:\ \mathscr{R}^{n_z \times n_\xi} \mapsto \mathbb{R}^+,$$

such that $J(M) = \infty$ for any $M \notin \mathscr{RH}_\infty$, where $\mathscr{R}^{n_z \times n_\xi}$ denotes the set of rational transfer $n_z \times n_\xi$ matrices, and $\mathscr{RH}_\infty$ denotes the set of stable real rational transfer matrices. Let $M(\delta, f) \in \mathscr{R}^{(p+n_z)\times(m+n_\xi)}$ be partitioned as follows

$$M(\delta,f) = \begin{bmatrix} M_{11}(\delta,f) & M_{12}(\delta,f) \\ M_{21}(\delta,f) & M_{22}(\delta,f) \end{bmatrix},$$

where, as depicted in Figure 2.7, the subsystem $M_{22}(\delta, f) \in \mathscr{R}^{p\times m}$ gives the relationships between the control actions and the measured output signals, and the subsystem $M_{11}(\delta, f) \in \mathscr{R}^{n_z \times n_\xi}$ describes the relationships between all exogenous inputs (such as noises, disturbances, reference signals) and the regulated (controlled) outputs that are related to the performance of the system (e.g. tracking errors). The feedback interconnection of the model $M(\delta, f)$ with some controller $K \in \mathscr{R}^{m\times p}$ is represented by the lower linear fractional transformation

$$\mathscr{F}_L(M(\delta,f),K) = M_{11}(\delta,f) + M_{12}(\delta,f)K(I - M_{22}(\delta,f)K)^{-1}M_{21}(\delta,f).$$

For a fixed controller K, the performance of the resulting closed-loop is therefore represented by $J(\mathscr{F}_L(M(\delta, f),K))$.

2.5.1 Passive Fault Tolerant Control

The passive robust FTC problem is then defined as the following optimization problem

Passive FTC:

$$K_P = \arg\min_{K} \sup_{\substack{\delta \in \Delta \\ f \in \mathscr{F}}} J(\mathscr{F}_L(M(\delta, f), K)). \tag{2.13}$$

In this way a controller needs to be found that minimizes the worst-case performance over all possible values for the uncertainty vector δ and the fault vector f. This problem is considered in [51] where methods are developed for robust controller design in the presence of structured uncertainty.

In practice, two main difficulties arise with the optimization problem (2.13), both being related to convexity. In the case when the state vector x_k is directly measured (or, equivalently, when $y_k = x_k$), the optimization problem (2.13) is convex in the controller parameters for many standard performance indices (e.g. $J(\cdot) = \|\cdot\|_2$, $J(\cdot) = \|\cdot\|_\infty$, etc.) provided that the set $\{M(\delta, f) : \ \delta \in \Delta, \ f \in \mathscr{F}\}$ is a convex polytope. In such cases (2.13) can be represented as a linear matrix inequality (LMI) optimization problem, for which there exist very efficient and computationally fast solvers. If $M(\delta, f)$ is not a convex set, however, the original problem (2.13) is also nonconvex and the LMI solvers cannot be used. A "brute force" way to deal with this problem is to embed the set $M(\delta, f)$ into a convex set. This, however, introduces unnecessary conservatism that for some problems might be unacceptable or undesirable.

In order to deal with such problems a probabilistic design approach is proposed in [51] that is basically applicable for any bounded set $M(\delta, f)$, as long as (2.13) can be rewritten as a robust LMI optimization problem (as for most state-feedback controller design problems). This method is basically an iterative algorithm that at each iteration generates a random uncertainty sample for which an ellipsoid is computed with the properties that (a) it contains the solution set (the set of all solutions to the robust LMI problem), (b) it has a smaller volume than the ellipsoid at the previous iteration. The approach is proved to converge to the solution set in a finite number of iterations with probability one.

In the output-feedback case the probabilistic method described in [51] cannot be directly applied because the optimization problem (2.13) cannot be rewritten as a robust LMI optimization problem. The reason for that is that the output-feedback problem in the presence of uncertainty is a bilinear matrix inequality (BMI) problem, and BMI problems are not convex. Actually, such problems have been shown to be NP-hard meaning that they cannot be expected to have polynomial time complexity. A local BMI optimization approach is developed in [51] that is guaranteed to converge to a local optimum of the cost function $J(\mathscr{F}_L(M(\delta, f), K))$.

2.5.2 Active Fault Tolerant Control

Whenever an estimate $\hat{f}$ of the fault vector f is provided by some FDD scheme, and if the imprecision in this estimate is described by an additional uncertainty $\Delta_f \in \boldsymbol{\Delta}_f$ so that $f = (I + \Delta_f)\hat{f}$, the active robust FTC can be defined as the problem:

$$\begin{aligned}&\text{given } f = (I+\Delta_f)\hat{f}, \text{ evaluate}\\ &\tilde{K}_A(\hat{f}) = \arg\min_{K(\hat{f})} \sup_{\substack{\delta \in \boldsymbol{\Delta} \\ \Delta_f \in \boldsymbol{\Delta}_f}} J(\mathscr{F}_L(M(\delta, f), K(\hat{f}))).\end{aligned} \tag{2.14}$$

The resulting controller would, in this way, be scheduled by the fault estimate $\hat{f}$ and will be robust with respect to uncertainties both in the model $M(\delta, f)$ and in the estimate of f. Clearly, the way in which the scheduling parameter $\hat{f}$ enters the controller needs to be assumed before one could proceed with the optimization.

In the above, Δ_f represents the FDD uncertainty that, as already discussed, usually increases after the occurrence of a fault. This will then subsequently decrease as the FDD scheme refines the estimate based on the availability of more input-output data from the impaired system. As a result the "maximal uncertainty" is only active for some relatively short periods of time compared with the lifetime of the system. Therefore, assuming a maximal uncertainty size during the complete operation might be overly conservative since the robust controller effectively trades off performance for increased robustness to uncertainties. Hence, it is interesting to allow the controller to deal with an FDD uncertainty with time-varying size. To this end, however, the FDD scheme should be capable of providing not only an estimate of the fault but also an upper bound on the magnitude of the uncertainty on this estimate. The size of the FDD uncertainty might, for instance, be represented by a scalar $\gamma_f(k)$ such that $f_k = (I + \gamma_f(k)\bar{\Delta}_f)\hat{f}_k$ with $\|\bar{\Delta}_f\|_2 \leq 1$. In this way the size of the uncertainty set is allowed to vary with time. In fact $\gamma_f(k)$ might be a vector to make it possible to assign different uncertainty sizes on the different entries of the fault vector f_k. Therefore, provided that the FDD scheme produces $(\hat{f}_k, \gamma_f(k))$ at each time instance, the achievable performance in (2.14) may further be improved by computing the controller by solving the following optimization problem

Active FTC:

$$\begin{aligned}&\text{given } f = (I+\gamma_f\bar{\Delta}_f)\hat{f}, \text{ evaluate}\\ &K_A(\hat{f}, \gamma_f) = \arg\min_{K(\hat{f},\gamma_f)} \sup_{\substack{\delta \in \boldsymbol{\Delta} \\ \bar{\Delta}_f \in \bar{\boldsymbol{\Delta}}_f \\ \underline{\gamma_f} \leq \gamma_f \leq \bar{\gamma}_f}} J(\mathscr{F}_L(M(\delta, f), K(\hat{f}, \gamma_f))),\end{aligned} \tag{2.15}$$

where $\bar{\boldsymbol{\Delta}}_f = \{\Delta \in \boldsymbol{\Delta}_f : \ \|\Delta\| \leq 1\}$, and where the vectors $\{\underline{\gamma_f}, \bar{\gamma}_f\}$, assumed known *a-priori*, define a lower and an upper bound on the possible uncertainty sizes. In this way methods can be developed for the design of robust active FTC for one uncertain local model $M(\delta, f)$. The robust active FTC design problem is considered in [51].

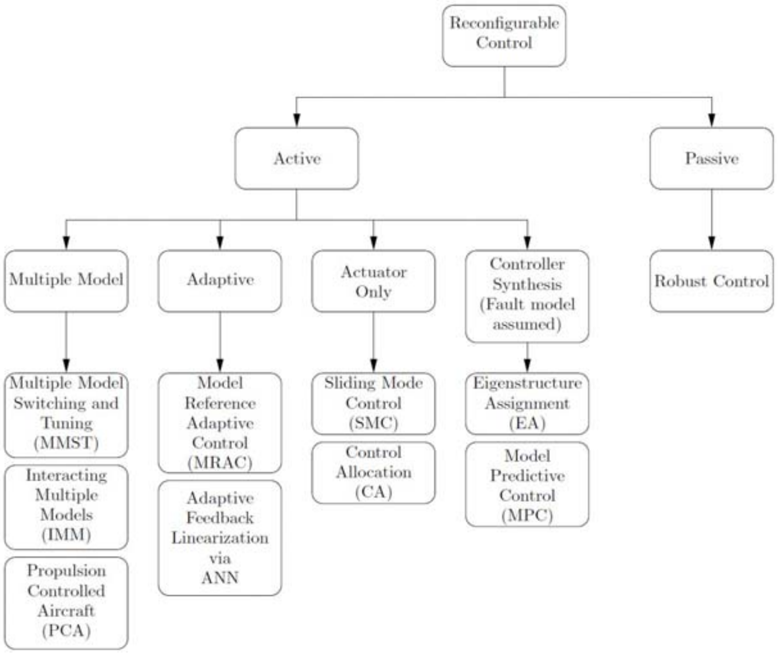
Reconfigurable Control
Active
Passive
Multiple Model
Adaptive
Actuator Only
Controller Synthesis (Fault model assumed)
Robust Control
Multiple Model Switching and Tuning (MMST)
Interacting Multiple Models (IMM)
Propulsion Controlled Aircraft (PCA)
Model Reference Adaptive Control (MRAC)
Adaptive Feedback Linearization via ANN
Sliding Mode Control (SMC)
Control Allocation (CA)
Eigenstructure Assignment (EA)
Model Predictive Control (MPC)

Fig. 2.8 Classification of approaches to reconfigurable flight control.

2.6 State-of-the-Art in Fault Tolerant Flight Control

In this section an overview of the existing work in the area of fault tolerant control is given, an area that has been gaining increasing attention in the aerospace community in recent years. Some overview books and papers in the field of FTC are [36, 45, 5, 96].

Due to their improved performance and their ability to deal with a wider class of faults, active FTC methods have gained much more attention in the literature than the passive FTC methods. In the following, a survey is given focussed on current active FTC methods of which several have been evaluated within this GARTEUR action group. The survey starts with a classification of the described and evaluated FTC methodologies to approach the problem of reconfigurable flight control.

2.6.1 Classification of Reconfigurable Control

Many methods have been proposed to solve the problem of fault tolerant control. As shown in Figure 2.8 they fall into two main categories: active and passive.

Passive methods are essentially robust control techniques which are suitable for certain types of structural failures that can be modelled as uncertainty regions around a nominal model. Any failure which doesn't push the system outside of the stability radius given by the robust controller will still have satisfactory stability and performance guarantees. However, any controller with a large enough stability radius to encompass most failure situations will likely be unnecessarily conservative and there is no guarantee that unanticipated or multiple failures could be handled or even that such a controller exists. There are also many types of common failures, such as actuator or sensor faults, which cannot be adequately modelled as uncertainty. These problems motivate the need for a controller which more directly addresses the situation.

The active methods differentiate themselves from passive approaches in that they take fault information explicitly into account and do not assume a static nominal model. Reconfigurable flight control is for the most part still an academic notion. Although there have been very few controllers implemented on physical systems and none on commercial aircraft, over the last 20 years several research programs have been formed to investigate their potential and as a result there are a variety of active methods. The following sections give an overview of each approach.

2.6.2 Multiple Model Control

The multiple model (MM) method is an active approach to FTC that belongs to the class of projection based methods rather than to the on-line re-design methods. The MM method is frequently used for FDD/FTC purposes [92, 78, 27, 37]. The MM method is based on a finite set of linear models M_i, $i = 1, 2, \ldots, N$ that describe the system in different operating conditions, i.e. in the presence of different faults in the system. For each such local model M_i a controller C_i is designed (off-line). The key in the design is to develop an on-line procedure that determines the global control action through a (probabilistically) weighted combination of the different control actions that can be taken. The control action weighting is usually based on a bank of Kalman filters, where each Kalman filter is designed for one of the local models M_i. On the basis of the residuals of the Kalman filters, the probability $1 \geq \mu_i \geq 0$ of each model to be in effect, is computed. The control action is then computed as the weighted combination

$$u(k) = \sum_{i=1}^{N} \mu_i(k) u_i(k), \quad \sum_{i=1}^{N} \mu_i = 1, \tag{2.16}$$

where $u_i(k)$ is the control action produced by a controller designed for the i-th local model.

The multiple model method is a very attractive tool for modelling and control of nonlinear systems. However, these approaches usually only consider a finite number of *anticipated* faults and proceed by building one local model for each anticipated fault. In this way, at each time instance only one model, say model M_i, is assumed to be in effect, so that its corresponding weight μ_i is approximately equal to unity and all the other weights μ_j, $j \neq i$ are close to zero. In such cases at each time instance one local controller is "active", namely the one corresponding to the model M_i that is in effect. The disadvantage here is that if the current model is not in the predesigned

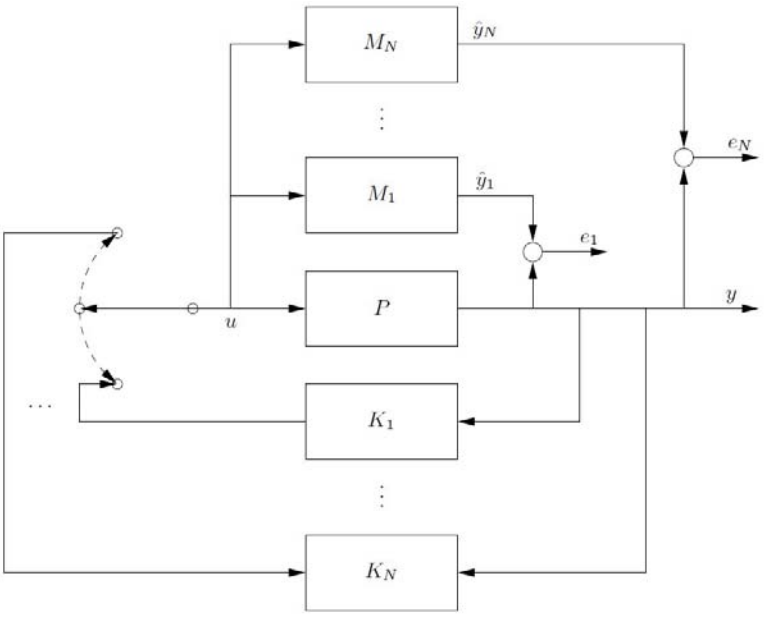

Fig. 2.9 Multiple Model Switching and Tuning

model set and is instead formed by some convex combination of the local models in the model set (representing, for instance, *unanticipated* faults) then, in general, the control action (2.16) is not the optimal one for this model. It can easily be shown that forming the global control action as in (2.16) can even lead to instability of the closed-loop system. In order to avoid that when dealing with unanticipated faults, an approach is proposed in [51] that uses a bank of predictive controllers and forms the global control action in an optimal way, so that the optimal control action for the current model is used at each time instance instead of (2.16). Another disadvantage of the MM approaches is that model uncertainties, as well as uncertainties in the weights $\mu_i(k)$, cannot be considered.

There are three types of reconfigurable control that fall under the heading of multiple model control: Multiple Model Switching and Tuning (MMST), Interacting Multiple Model (IMM) and Propulsion Controlled Aircraft (PCA). In the first two cases all expected failure scenarios are enumerated during a Failure Modes and Effects Analysis (FMEA) and fault models constructed which cover each situation. When a failure occurs, MMST switches to a pre-computed control law corresponding to the current failure situation. Rather than using the model which is closest to the current failure scenario, IMM computes a fault model as a convex combination of all pre-computed fault models and then uses this new model to make control decisions. PCA is a special case of MMST, where the only anticipated fault is a total hydraulics failure, and in this case only the engines are used for control. The following sections discuss these three approaches.

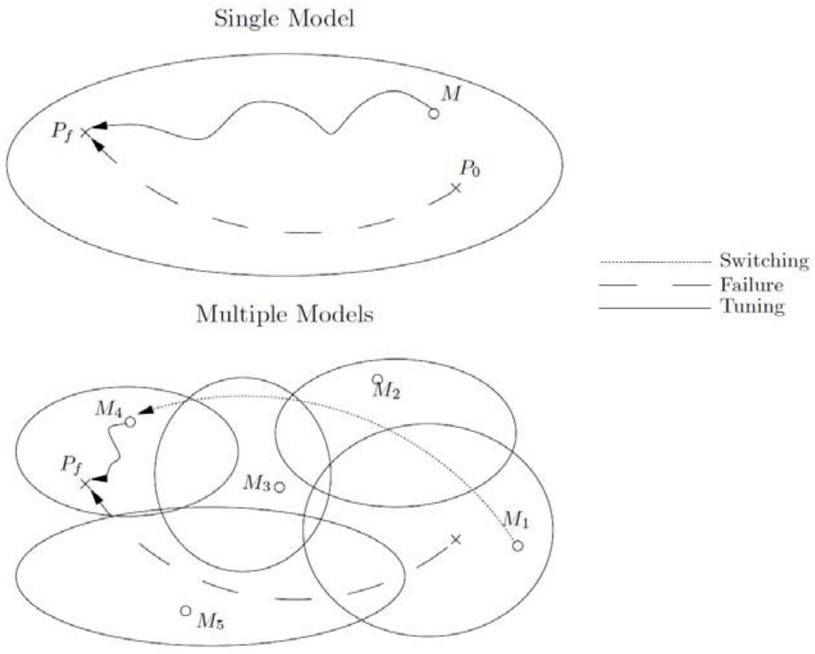

Fig. 2.10 Single Model vs. Multiple Model Adaptation

2.6.2.1 Multiple Model Switching and Tuning (MMST)

Although the idea of multiple model control has been around for many years, it has seen some interest in the reconfigurable control literature in the last few years [13, 34, 14, 10, 11, 12, 53, 25]. In MMST, the dynamics of each fault scenario is described by a different model. These models are referred to as the identification models [13] and are setup in parallel, with each one having a corresponding controller as shown in Figure 2.9. The problem then becomes one of choosing which model/controller pair to switch to at each time instant.

Figure 2.10 helps to motivate the use of MMST in reconfigurable control systems. During a failure the plant is assumed to move from some nominal model P_0 to a failure model P_f some distance away in parameter space. The top half of the figure shows an adaptive control scheme which is using only a single model, and the lower a MMST method. For certain plants, the MMST converges to the correct fault model faster than a single model approach.

Consider a system of the form

$$P = \begin{cases} \dot{x} = A_0(p(t))x + B_0(p(t))u \\ y = C_0(p(t))x \end{cases} \tag{2.17}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $y \in \mathbb{R}^k$, $A_0 \in \mathbb{R}^{n\times n}$, $B_0 \in \mathbb{R}^{n\times m}$, $C_0 \in \mathbb{R}^{k\times n}$ and $p(t) \in \mathscr{S} \subseteq \mathbb{R}^l$ are the plant parameters. The quantity $p(t)$ varies in time in an abrupt fashion and represents the various failure scenarios.

Definition 6.1 (Model Set). *The model set $\mathscr{M}$ is a set of N linear models*

$$\mathscr{M} : \{M_1, \dots, M_N\}$$

such that

$$M_i : \begin{cases} \dot{x}_i = A_i x_i + B_i u \\ y_i = C_i x_i \end{cases}$$

where model M_i corresponds to a particular set of parameters $p_i \in \mathscr{S}$.

A stabilizing controller K_i is designed for each model $M_i \in \mathscr{M}$.

The control law proceeds as follows. At each time step, the model which is closest to the current system is determined by computing a performance index $J_i(t)$, which is a function of the errors $e_i(t)$ between the estimated outputs of model M_i and the measurements at time t. A commonly used index is [71]

$$\begin{gathered} J_i(t) = \alpha e_i^2(t) + \beta \int_0^t e^{-\lambda(t-\tau)} e_i^2(\tau) d\tau \\ \alpha \geq 0, \beta > 0, \lambda > 0 \end{gathered}$$

where α and β are chosen to give a desired combination of instantaneous and long-term accuracy measures. The forgetting factor λ ensures the boundedness of $J_i(t)$ for bounded e_i. The model/controller, M_i/K_i with the smallest index is switched to and a waiting period of $T_{min} > 0$ is allowed to pass in order to prevent arbitrarily fast switching. Most MMST algorithms include a 'tuning' part which occurs during the period while a controller K_i is active, during which time the parameters of the corresponding model, and only the corresponding model M_i, are being updated using an appropriate identification technique (e.g. [2]).

Recent interest in this approach arises from the following stability result:

Theorem 6.2 [71]. *Consider the switching and tuning system described above, where the N models are all fixed and the proposed switching scheme is used with β, λ, $T_{min} > 0$, and $\alpha \geq 0$. Then, for each plant with parameter vector $p \in \mathscr{S}$, there is a positive number $T_{\mathscr{S}}$ and a function $\mu_{\mathscr{S}}(p, T_{min}) > 0$, such that if:*

- *the waiting time $T_{min} \in (0, T_{\mathscr{S}})$*
- *there is at least one model M_i with parameter error $||\hat{p}_i - p|| < \mu_{\mathscr{S}}(p, T_{min})$*

then all the signals in the overall system, as well as the performance indices $\{J_i(t)\}$, are uniformly bounded. Here $T_{\mathscr{S}}$ depends only upon $\mathscr{S}$, and $\mu_{\mathscr{S}}$ also depends upon α, β, λ and $\mathscr{S}$.

In essence, Theorem 6.2 states that the MMST system is stable if the set of models M_i is dense enough in the parameter space $\mathscr{S}$ and the sampling rate T_{min} is fast enough. How dense and how fast depend on the particular system and Theorem 6.2 gives no insight into the selection of $\mathscr{M}$ or T_{min}.

Despite the limitations of Theorem 6.2, there are several papers which have applied these methods. In [13, 10, 11, 12] a MMST controller is developed for the highly over-actuated tailless advanced fighter aircraft (TAFA). Eleven fault models are required to cover the scenario of right wing damage ranging from 0% to 100% and a switching interval of $25ms$ is needed for stability. Clearly, this approach will not scale well to the situation where more than one failure, or multiple failures are considered. Ref. [14] describes a MMST scheme which can handle locked, floating, hard-over or loss of effectiveness actuator failures for an F-18 aircraft carrier landing manoeuvre. Only five models are needed for satisfactory performance, but again, multiple failures cannot be accommodated. Ref. [13] introduced a new method of failure parameterizations for jammed actuators, enabling multiple complete failures of control surfaces for an F-18 to be handled using a large number of simple models.

For systems with relatively few and well understood failure modes, multiple model switching and tuning has advantages in being fast and provably stable. However, the main limitation is that there may be failure scenarios that were not modelled, which would likely be the case for multiple or structural failures. A severe limitation for larger systems is that the number of models required increases exponentially with the number of simultaneous failures considered.

2.6.2.2 Interacting Multiple Models (IMM)

The method of interacting multiple models (IMM) attempts to deal with the key limitation of MMST, namely that every fault scenario must be modelled, by considering fault models which are convex combinations of models in a model set.

The primary assumption of IMM is that every possible failure can be modelled as a convex combination of models in a pre-determined model set $\mathscr{M}$ as defined above in Definition 6.1

$$M_f = \sum_{i=1}^{N} \mu_i M_i = \mu^T \begin{bmatrix} M_1 \\ \vdots \\ M_N \end{bmatrix}, \quad M_i \in \mathscr{M},\ \mu_i > 0 \in \mathbb{R}, \sum_{i=1}^{N} \mu_i = 1, \tag{2.18}$$

Then M_f is the system:

$$M_f : \begin{cases} \dot{x} = \begin{bmatrix} A_1 & 0 & \dots & 0 \\ 0 & A_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_n \end{bmatrix} x + \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{bmatrix} u \\ \\ y = \begin{bmatrix} \mu_1 C_1 & \mu_2 C_2 & \dots & \mu_N C_N \end{bmatrix} x \end{cases} \tag{2.19}$$

It is still an open question how to choose this model set or when the assumption that the failure model can be written as a convex combination of the models in the set, is valid.

Fault detection and modelling is then done online by identifying the variables μ_i in Equation (2.18). Two proposed methods exist for computing the coefficients μ. In the first, a Kalman filter is designed for each $M_i \in \mathscr{M}$ and all filters are run in parallel. The probability that each of these models represents the true state of the system can be computed and the coefficients μ are set to these probabilities. This method is named Multiple Model Adaptive Estimation (MMAE) and is used in [68, 93]. In the second approach, the previous k_f time instants are considered and the estimated output at each point is computed as a function of μ, which is then selected to minimize this difference. This approach is advocated in [52, 54].

Once a fault model has been identified, there are a variety of methods for control law calculation. Refs. [52] and [54] suggest a Model Predictive Control (MPC) scheme where the minimization of the past tracking error, and therefore of μ, is included in the cost function. Ref. [93] proposes an Eigenstructure Assignment (EA) (see Section 2.6.6) method and [68] uses a fixed controller, using the fault model M_f only for state estimation.

IMM is attractive in its ability to handle multiple failure scenarios by combining single failure models. However, the requirement of finding the coefficients μ after a failure makes this an adaptive algorithm and not a model-switching one. As a result it loses some of the speed of the MMST approach. The formulation of IMM as an MPC problem given in [54] also offers the potential of handling actuator constraints naturally.

2.6.2.3 Propulsion Controlled Aircraft (PCA)

After the possibility of control using only the engine throttles was demonstrated by the Sioux City accident (see Chapter 1), and following a recommendation from the National Transportation Safety Board of America, the PCA problem was taken up by the NASA Dryden Flight Research Center [16, 17] in order to provide a backup in case of total hydraulic failure. PCA is a specific instance of a multi-model approach where the fault model is identical to the nominal one, but in which all control surfaces are free floating. In 1995, a demonstration was made during which a MD-11 (Figure 2.11) and a F-15 recovered from a complete hydraulic failure and landed successfully under propulsion-only control [18]. PCA is a useful and important idea and solves a very practical problem. However, it clearly is not sufficient to solve the general reconfigurable control problem.

2.6.3 Control Allocation (CA)

Control allocation is the problem of producing a desired set of forces and moments from a (usually large) set of actuators. For example, as shown in Figure 2.12, the output of the control law can be a set of desired moments and the job of the control

Fig. 2.11 Landing demonstration of MD-11 Propulsion Controlled Aircraft (PCA), NASA Dryden, 2001 (copyright NASA)

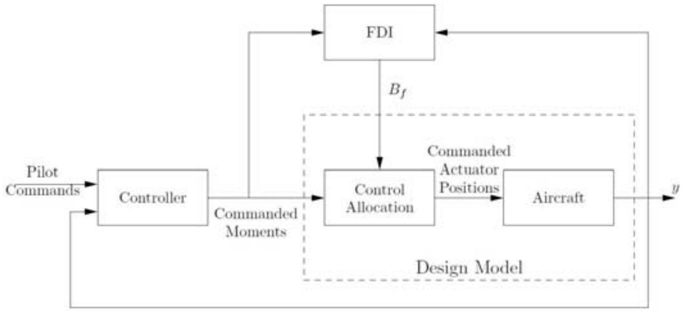

Fig. 2.12 Control Allocation scheme

allocation block is then to select appropriate setpoints for the actuators which will produce those moments.

The control allocation algorithm takes as inputs the desired moments and an estimation of the input derivatives (adaptive B_f matrix) from either a FDI or a system identification algorithm. The algorithm therefore has the ability to adapt the way actuation forces are generated from the available actuators, to the faults that have occurred. For example, if the effectiveness of a certain actuator becomes 0% due to a fault, the corresponding column in B_f will also become 0. This actuator is then not considered anymore by the control allocation method. Instead, the remaining actuators can be used to generate the desired actuation forces. The goal is then to produce the desired moments u_d by selecting the appropriate inputs to the system u. Whether this can be done depends on the difference between the size of $u_d \in \mathbb{R}^m$ and the column rank of $B_f \in \mathbb{R}^{n \times k}$. There are three cases to consider:

- If $m < k$ the moments can be selected exactly and the remaining degrees of freedom can be used (for example) to drive the actuators towards a desired position u_p by minimizing [90, 15, 20]:

$$\tfrac{1}{2}||u-u_p||_{W_p} = \tfrac{1}{2}(u-u_p)^T W_p (u-u_p) \text{ where } W_p = W_p^T > 0$$
subject to $Bu = u_d$

where W_p is a weighting matrix prioritizing critical actuators.

- If $m = k$ then there is only one solution which places the moments exactly

$$u = B^{-1}u_d$$

- In the case when $m > k$ there are not enough degrees of freedom to achieve u_d and so a compromise must be made by (for example) minimizing the weighted norm

$$\frac{1}{2}||Bu-u_d||_{W_d}$$

Control allocation has been heavily studied in relation to over-actuated systems (see [29] for a survey) and has received a great deal of attention in the literature for reconfigurable systems as it allows actuator failures to be handled without the need to modify the control law. However, there are two major limitations to this approach to reconfiguration. Firstly, the system will not necessarily be stable, even with a stabilizing control law, when $m > k$, as the input seen by the system may not be equal to that intended by the controller. Secondly, the dynamics and limitations of the actuators after a failure are not taken into account in the control law. This means that the controller will still be attempting to achieve the original system performance even though the actuators are not capable of achieving it.

Control allocation has received considerable attention from the field of aerospace engineering. Extensions to the simple control allocation problem presented here have been considered in the literature. In [9] and [28] the problem of control allocation with magnitude and rate limits on the actuators is considered, [24] develops a control allocation controller for the extremely over-actuated Innovative Control Effector (ICE) aircraft and [98] looks at restoring as much of the performance of the original B matrix as possible after an actuator failure. Other examples of work in the area of control allocation for aerospace applications can be found in [7] and [38].

2.6.4 Adaptive Feedback Linearization via Artificial Neural Network

This section examines a method primarily developed by Calise *et al* [42, 48, 41, 19, 21, 90, 20] involving a Model Reference Adaptive Control (MRAC) scheme through adaptive feedback linearization augmented by an Artificial Neural Network (ANN). This approach has been successfully demonstrated via simulation on the Tailless Advanced Fighter Aircraft (TAFA) [90, 20] and the X-36 [21]. The approach presented here splits the dynamics of the plane into three SISO subsystems, each of which has a model reference adaptive controller: roll, pitch and yaw. The output of each controller is a command specifying a desired roll, pitch or yaw moment and it is then the job of the Integrated Control Effector Management (ICEM) [15, 90], a form of control allocation, to generate these moments using the available control surfaces. In the next three sections, a brief overview of the principles of feedback linearization on SISO systems will be given, review the particulars and benefits of its use in reconfiguration and finally discuss the ICEM and its role in the proposed method.

2.6.4.1 Single-Input Single-Output (SISO) Feedback Linearization

Consider the SISO nonlinear system

$$\begin{aligned} \dot{x} &= f(x,u) \\ y &= h(x) \end{aligned} \quad x \in \mathbb{R}^n, u, y \in \mathbb{R} \tag{2.20}$$

In feedback linearization the goal is to design a control law for the SISO nonlinear system given in Equation 2.20 such that the closed loop system is linear and controllable. Assuming the relative degree of h is $r = n$, the r^{th} derivative of the output is the first derivative that is directly affected by the control. As a result, we can write the system dynamics in the normal form ([44], Section 4.2):

$$\begin{array}{llll} \Phi_1(x) = & h(x) & = z_1 & = y \\ \Phi_2(x) = & \frac{dh(x)}{dt} & = \dot{z}_1 & = z_2 \\ \Phi_3(x) = & \frac{d^2h(x)}{dt^2} & = \dot{z}_2 & = z_3 \\ \vdots & \vdots & \vdots & \\ \Phi_r(x) = & \frac{d^rh(x)}{dt^r} & = \dot{z}_{r-1} & = z_r \\ & \dot{z}_r = h_r(z,u) & & \end{array} \tag{2.21}$$

where $\Phi(x) = z = [z_1, \ldots, z_r]'$.

We now define the ‘pseudo control signal’ ν

$$\nu = \hat{h}_r(\Phi(x), u)$$

where $\hat{h}_r(\Phi(x), u)$ is an invertible estimate of $h_r(z,u)$. Then the system dynamics can be expressed as

$$\begin{aligned} \dot{z}_i &= z_{i+1}, \quad 1 \le i \le r-1 \\ \dot{z}_r &= \nu + \Delta \\ y &= z_1 \end{aligned} \tag{2.22}$$

where

$$\Delta = \Delta(z,u) = h_r(z,u) - \hat{h}_r(y,u)$$

In effect, the transformation places r integrators between the pseudo control ν and the system output y, with the error Δ acting as a disturbance signal. This is now a linear and controllable system.

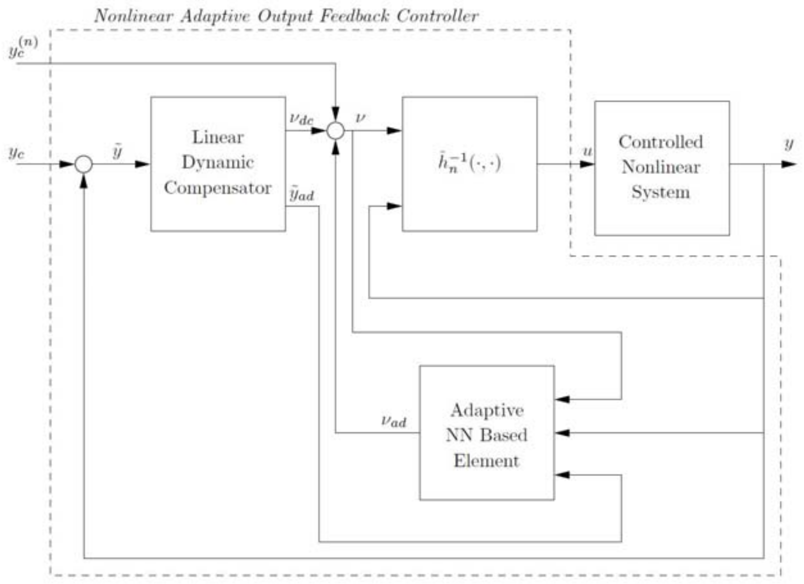
Nonlinear Adaptive Output Feedback Controller
$y_c^{(n)}$
y_c
$\tilde{y}$
Linear Dynamic Compensator
ν_{dc}
ν
$\tilde{y}_{ad}$
$\hat{h}_n^{-1}(\cdot,\cdot)$
u
Controlled Nonlinear System
y
Adaptive NN Based Element
ν_{ad}

Fig. 2.13 Nonlinear Adaptive Output Feedback Controller

2.6.4.2 Feedback Linearization for Reconfigurable Control

Feedback linearization can be used in a model-following configuration by choosing the pseudo control to have the form [19]

$$\nu = y_c^r + \nu_{dc} - \nu_{ad},$$

where ν_{dc} is the output of a stabilizing linear compensator for the linearized system given by Equation (2.22) with $\Delta = 0$. The quantity ν_{ad} is an adaptive signal designed to cancel Δ and y_c^r is the r^{th} derivative of the signal to be tracked. The signal y_c^r can be obtained from an (at least) r^{th} order reference model which defines the desired dynamics.

If the model of the system is perfect, $\Delta = 0$ and we could simply apply the input $u = \hat{h}_r^{-1}(x,\nu) = h_r^{-1}(x, y_c^r + \nu_{dc})$ and the system would track the reference trajectory. However, as there will always be modelling errors, the error Δ needs to be compensated online and for this an ANN can be used. Neural networks can be trained to approximate any function with an arbitrary precision. As a result, the ANN can estimate the modelling error and hence cancel it. The benefit of this approach is that no model structure needs to be assumed in order to estimate the error. Figure 2.13 shows the structure of the full controller, and Figure 2.14 that of the linear compensator.

This control technique was proposed as a method of reconfigurable control in combination with Wise's ICEM [15]. This scheme is suited to reconfigurable control, as the adaptation makes no assumptions about the structure of the system after

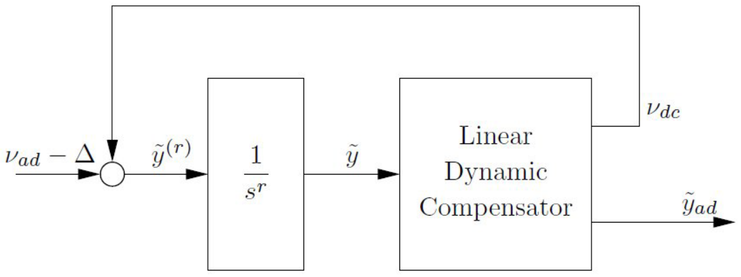

Fig. 2.14 Block Diagram of the Error Dynamics

the failure. Since the ANN can approximate any nonlinear function, it can track and cancel any structural failures which may occur under the assumption of sufficient control authority and excitation for adaptation. The techniques presented in this section have been developed and expanded upon in several publications: Single Input Single Output (SISO) stability proofs [19], input saturation [48], combined aero/engine control [42] and highly over-actuated systems [21].

2.6.5 Sliding Mode Control (SMC)

This section reviews the work in [82]. The proposed controller is setup in a two-loop cascade configuration, with the ultimate goal of tracking a trajectory given by roll, pitch and yaw angle setpoints. The outer-loop takes roll, pitch and yaw setpoints and provides angular rate commands to the inner-loop, which is assumed to track the commands using the inputs to the actuators.

The outer-loop is designed using standard robust SMC techniques. The inner-loop is also a robust sliding mode controller but has an adaptive feature to handle actuator magnitude and rate limitations. In [82] it is shown that modifying the size of the boundary layer online can ensure that integrators do not wind up, as well as ensuring that actuator magnitude and rate limits are satisfied. There is a direct trade-off between the size of the boundary layer and tracking performance. Therefore, this procedure provides an intuitive method of maximizing tracking while ensuring actuator limits.

The benefits of this controller to reconfigurable control are two-fold. Firstly, being a robust control technique, it can handle all structural failures which modify the dynamics of the plant less than the assumed uncertainty. Secondly, the online adaptation of the boundary layer can handle partial loss of actuator surfaces, while avoiding limits and integrator windup by reducing the tracking performance. Although this technique provides benefits to aircraft control, there are limitations due to the use of SMC when it is presented with the full reconfigurable problem.

1. There must be one and only one control surface for every controlled variable and second, none of the control surfaces can ever be lost. This is handled in [82] by only considering failures which cause a partial loss of effectiveness of

the control surfaces, which is not realistic as floating or jammed actuators are certainly possible failure scenarios. This problem could be addressed by placing a control allocation algorithm (see Section 2.6.3) between the requested outputs and the physical actuators.

2. The method proposes to use robust control to handle all structural failures. This requires a de-tuning of the controller to the point that it can handle uncertainties including all possible structural failures, which may well result in an excessively conservative controller in the non-failure situation.

2.6.6 Eigenstructure Assignment (EA)

Eigenstructure Assignment (EA) was made popular in the 1980s primarily by Andry, Shapiro and Chung in their paper [1] where the method of Direct Eigenstructure Assignment (DEA) was introduced. The idea behind the method is to place the eigenvalues of a linear system using state feedback and then use any remaining degrees of freedom to align the eigenvectors as accurately as is possible. The eigenvalues determine the natural frequency and damping of each mode while the eigenvectors control how much each mode contributes to a given output. The following sections first give a brief overview of the theory behind EA and then a review of its use in reconfigurable control.

2.6.6.1 Introduction to Eigenstructure Assignment

The eigenstructure assignment (EA) method [63] to controller reconfiguration is a more intuitive approach than the Pseudo Inverse method (Section 6.6.3). It aims at matching the eigenstructures (i.e. the eigenvalues and the eigenvectors) of the A-matrices of the nominal and the faulty closed-loop systems. The main idea is to exactly assign some of the most dominant eigenvalues while at the same time minimizing the 2-norm of the difference between the corresponding eigenvectors. The procedure has been developed both under constant state-feedback [89] and output-feedback [26]. More specifically, in the state-feedback case, if λ_i, $i = 1, 2, \dots, n$ are the eigenvalues of the A-matrix of the nominal closed-loop system formed as the interconnection of (2.25) with the constant state-feedback control action $u_k = Fx_k$, and if v_i are their corresponding eigenvectors, the EA method computes the state-feedback gain F_R for the faulty model (2.26) as the solution to the following problem

$$\text{EA}: \begin{cases} \text{Find } F_R \\ \text{such that } (A_f + B_f F_R)v_i^f = \lambda_i v_i^f,\ i = 1, \dots, n, \\ \text{and } v_i^f = \arg\min\limits_{v_i^f} \|v_i - v_i^f\|_{W_i}^2, \end{cases} \tag{2.23}$$

where $\|v_i - v_i^f\|_{W_i}^2 = (v_i - v_i^f)^T W_i (v_i - v_i^f)$. In other words, the new gain F_R needs to be such that the poles of the resulting closed-loop system coincide with the poles of the nominal closed-loop system and, in addition, the eigenvectors of the closed-loop A-matrices are as close as possible. As both the eigenvectors and the eigenvalues determine the shape of the time response of the closed-loop system, this method can be thought of as trying to preserve the nominal closed-loop system time-response after the occurrence of faults. Thus, the objective of the EA method seems more "natural" than that of the Pseudo Inverse Method (PIM) and, moreover, the stability is guaranteed. The computational burden of the approach is not high since an analytic expression for the solution to (2.23) is available, i.e. no on-line optimization is necessary. The disadvantage is that model and FDD uncertainties cannot be easily incorporated in the optimization problem, and that only static controllers are considered. The references [22, 58] further describe the use of Eigenstructure Assignment.

2.6.6.2 Reconfigurable Eigenstructure Assignment

Although a method for choosing appropriate eigenvectors and eigenvalues is not immediately obvious for aircraft, some studies have been made on the effects of the eigenstructure (eigenvalues and eigenvectors) on flying qualities [23]. Methods which propose EA for use in reconfigurable flight control systems [58, 4, 94] first assume a linear fault model which has been given to the controller by a FDI system.

$$\begin{aligned} \dot{x} &= A_f x + B_f u \\ y &= C_f x \end{aligned}$$

The goal is then to design a stabilizing output feedback law K_f

$$u = K_f C_f x \tag{2.24}$$

such that the new eigenstructure closed-loop system $A_f + B_f K_f C_f$ is as close as possible to that of the original closed-loop system $A + BKC$.

The choice of K_f can be made in a variety of ways, but the placement of the eigenspace is limited by Theorem 2.1. Generally the eigenvalues of the failed system, λ_i^f are ordered from most important to least and then the top $\max(m,k)$ are made to exactly match those of the non-failed system λ, while the remainder are kept stable. Similarly, the most important $\max(m,k)$ eigenvectors of the failed system, v_i^f, are made close to those of the original system v_i in the least squares sense.

Theorem 2.1. *[23] Consider a controllable and observable system with the output feedback law of* (2.24) *and the assumption that the matrices B and C are full rank. Then, there exists a matrix $K \in \mathbb{R}^{m \times k}$ such that*

1. $\max(m,k)$ *closed-loop eigenvalues can be assigned*
2. $\max(m,k)$ *eigenvectors can be partially assigned with* $\min(m,k)$ *entries in each vector arbitrarily chosen*

There are several limitations to this approach when applied to reconfiguration. Firstly, only linear systems have been considered and actuator limitations have not been taken into account. Secondly, a perfect fault model is assumed and the effects of uncertainty have not been extensively studied. Finally, the effect of the eigenvectors in the failed system not being exactly equal to those in the nominal system is not well understood. The result of these significant limitations is that only a few researchers have proposed this approach.

2.6.6.3 Pseudo Inverse Method (PIM)

The *pseudo-inverse method* (PIM) [31] is one of the most cited active methods to FTC due to its computational simplicity and its ability to handle a very large class of system faults. The basic version of the PIM considers a nominal linear system

$$\begin{cases} x_{k+1} = Ax_k + Bu \\ y_k = Cx_k, \end{cases} \tag{2.25}$$

with a linear state-feedback control law $u_k = Fx_k$, under the assumption that the state vector is available for measurement. The method allows for a very general post-fault system representation

$$\begin{cases} x^f_{k+1} = A_f x^f_k + B_f u^R_k \\ y^f_k = C_f x^f_k, \end{cases} \tag{2.26}$$

where the new, reconfigured control law is taken with the same structure, i.e. $u^R_k = F_R x^f_k$. The goal is then to find the new state-feedback gain matrix F_R in such a way that the "distance" (defined below) between the A-matrices of the nominal and the post-fault closed-loop systems is minimized, i.e.

$$\text{PIM} : \begin{cases} F_R = \arg\min_{F_R} \|(A+BF) - (A_f + B_f F_R)\|_F \\ \quad\;\; = B_f^\dagger (A + BF - A_f), \end{cases} \tag{2.27}$$

where $B_f^\dagger$ is the pseudo-inverse of the matrix B_f. The advantages of this approach are that it is very suitable for on-line implementation due to its simplicity, and moreover, that it allows for changes in all state-space matrices of the system as a consequence of the faults. A very strong disadvantage is, however, that the optimal control law computed by equation (2.27) does not always stabilize the closed-loop system. Simple examples that confirm this fact can easily be generated, see for example [31]. To circumvent this problem, the *modified pseudo-inverse method* was developed in [31] that basically solves the same problem under the additional constraint that the resulting closed-loop system remains stable. This, however, results in a constrained optimization problem that increases the computational burden. A similar approach is also discussed in [77, 62], where the reconfigured control action u^R_k is directly computed from the nominal control u_k as $u^R_k = B_f^\dagger B u_k$. Other modifications of this approach that were proposed include the consideration of additive faults on the state equation and additive terms on the control action to compensate for them in [73] and static output-feedback in [59].

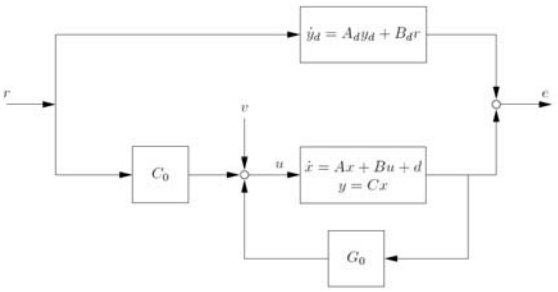

Fig. 2.15 Model Reference Adaptive Control

2.6.7 Model Reference Adaptive Control (MRAC)

Aström defines an adaptive controller as "a controller with adjustable parameters and a mechanism for adjusting those parameters" ([2], Page 1). Clearly, all methods presented in this survey are adaptive to some degree (save for robust control techniques) as they require the identification of a fault model in order to compute a control law. The approach we consider here is Model Reference Adaptive Control (MRAC) which can be effective for many types of structural failures and is often used as a final stage in other algorithms.

The goal of adaptive model-following is to force the plant output to track a reference model. We consider linear plants of the form

$$\begin{aligned}\dot{x} &= Ax + Bu + d\\ y &= Cx\end{aligned} \tag{2.28}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $y \in \mathbb{R}^k$ and a reference model of the form

$$\dot{y}_d = A_d y_d + B_d r \tag{2.29}$$

where $y_d \in \mathbb{R}^k$ and $r \in \mathbb{R}^k$. A_d and B_d are arbitrary square matrices with A_d stable. State feedback of the form shown in Figure 2.15 is considered.

$$u = C_0 r + G_0 x + v$$

where $C_0 \in \mathbb{R}^{k\times k}$, $G_0 \in \mathbb{R}^{k\times n}$ and $v \in \mathbb{R}^k$ are free controller parameters. The closed loop dynamics are then

$$\dot{y} = (CA + CBG_0)x + CBC_0 r + CBv + Cd \tag{2.30}$$

The goal is now to make the closed loop dynamics given by Equation (2.30) match the desired dynamics of Equation (2.29). If the model shown in Equation (2.28) was known exactly, the controller parameters C_0, G_0 and v could be computed to achieve this. However, since post-failure the model in (2.28) is not known exactly, the controller parameters need to be adapted. There are two methods to achieve this: direct and indirect adaptation.

2.6.7.1 Indirect Adaptation

There are two stages in indirect adaptive control. Firstly the matrices A, B and d are estimated and then under the assumption that these estimates are correct the control parameters G_0, C_0 and v are computed such that the closed-loop system matches the desired dynamics.

A least squares algorithm can be used to compute the estimates $\hat{A}, \hat{B}$ and $\hat{d}$ ([2]), which can then be used to compute the controller parameters such that the closed loop dynamics (2.30) match the desired ones (2.29).

$$\begin{aligned} C_0 &= (C\hat{B})^{-1}B_d \\ G_0 &= (C\hat{B})^{-1}(A_dC - C\hat{A}) \\ v &= (C\hat{B})^{-1}(Cd) \end{aligned}$$

where we must assume that $\det(C\hat{B}) \neq 0$.

The idea of identifying the model online and then computing a control law under the assumption that the estimated model is perfect is common in the reconfigurable control literature. For example, the EA algorithms of Section 2.6.6 and the IMM algorithms of Section 2.6.2.2 assume this type of structure.

2.6.7.2 Direct Adaptation

Direct adaptive control attempts to estimate the controller parameters G_0, C_0 and v directly rather than first computing the model parameters. We define $G_0^\star, C_0^\star$ and $v^\star$ as the 'correct' values of the controller parameters which will force the plant to track the reference model. A problem can then be formulated such that a least squares routine can be used to estimate the correct controller parameters [8]. The idea of direct adaptation is seen in algorithms such as the adaptive feedback linearization approach presented in Section 2.6.4.

The basic model-reference adaptive control techniques described here are not by themselves suitable for reconfigurable control for two main reasons. Firstly, in order for these approaches to work a model structure must be assumed. However, the types of failures addressed in reconfigurable control may well cause the plant structure to change drastically. Secondly, adaptive control requires the system parameters to change slowly enough for the estimation algorithm to track them. Faults may well cause abrupt and drastic changes in the parameters moving the system instantaneously to a new region of the parameter space. There is no guarantee that the system will be stable during the transient period in which the adaptive algorithm is identifying the faulty plant. Despite the limitations of adaptive control for reconfiguration, some researchers have attempted to apply it in slightly modified forms [6, 35, 8]. As a result adaptive control on its own is not enough to handle the general problem, but may well be an important part of a reconfigurable algorithm.

2.6.8 Model Predictive Control

After its introduction in the 1970s, model predictive control (MPC) has become a popular strategy in the field of industrial process control. The main reasons for this popularity are the abilities of MPC to control multivariable systems and to handle constraints. Initially, MPC was primarily applied to relatively slow processes such as the plants encountered in the process industry. The reason for this is that MPC can require considerable computational effort to generate the control signals as a result of an optimization that has to be performed at each time instance. This optimization is based on matching a prediction of the system output to some desired reference trajectory. The latter is assumed to be known in advance. For the relatively slow plants in the process industry, the considerable computational effort of MPC was not an issue because of the low sampling frequency of the controllers. However, for faster systems, higher frequencies were required that prevented on-line implementation of MPC for such systems. More recently, MPC has become a viable alternative for faster systems as a result of the increase in computational power that is available in modern control systems. For example, in [79] MPC has been used for real-time control of a miniature hovercraft. Another example is [56], in which MPC has been used for real-time control of an unmanned aerial vehicle.

As discussed in [65], the MPC architecture allows fault-tolerance to be embedded in a relatively easy way by: (a) redefining the constraints to represent certain faults (usually actuator faults), (b) changing the internal model, (c) changing the control objectives to reflect limitations due to the faulty mode of operation. In such a way there is practically no additional optimization that needs to be executed on-line as a consequence of a fault being diagnosed, so that this method can be viewed as having an inherent self-reconfiguration property. However, if state-feedback MPC is used in an interconnection with an observer one should also take care to also reconfigure the observer appropriately in order to achieve fault-tolerant state estimation. Examples of the application of MPC to FTC are numerous [66, 51, 76, 50, 56].

Model predictive control has been proposed as a method for reconfigurable flight control due to its ability to handle constraints and changing model dynamics systematically. MPC relies on an internal model of the system and so, like many of the approaches presented in this survey, a fault model is required. There are two general classifications of aircraft faults: actuator and structural. As noted in [69], these failures can be handled naturally in a MPC framework via changes in the input constraints and internal model. Actuator limit and rate constraints can be written as:

$$\begin{aligned} u_i^l &\leq u_i(t) \leq u_i^u \\ du_i^l &\leq \dot{u}_i(t) \leq du_i^u \end{aligned}$$

for actuator inputs u_1 through u_m. If actuator i becomes jammed at position $u_i^\star$ the MPC controller can be made to compensate by simply changing the constraints on input i to

$$\begin{aligned} u_i^\star &\leq u_i(t) \leq u_i^\star \\ 0 &\leq \dot{u}_i(t) \leq 0 \end{aligned}$$

The result will be similar to the control allocation approach where other input channels are used to create the same effect. As noted in [64], an MPC controller can be designed so that it has an intrinsic ability to handle jammed actuators without the need to explicitly model the failure. Structural failures can also be handled in a natural fashion by changing the internal model used to make prediction in either an adaptive fashion [52], a multi-model switching scheme [13] or by assuming an FDI scheme which provides a fault model [40, 39, 55, 66].

An important issue when using MPC is the robustness with respect to model uncertainties. Since MPC heavily depends on how well the controlled system is represented by the model used, measures should be taken in case of model uncertainty. One method to do so is to define an uncertainty region around the nominal model and to ensure that the MPC algorithm achieves a certain minimum performance level for the whole uncertainty region. MPC methods that take model uncertainty explicitly into account are referred to as robust MPC methods. One of the first research efforts that addresses the issue of robust MPC was performed by [60]. This issue has been addressed in the context of FTC in [51].

Like most active FTC methods, MPC-based FTC requires availability of fault information to accommodate faults. This requirement limits the ability of MPC-based FTC to deal with unanticipated fault conditions for which fault information cannot be obtained most of the time. An FTC algorithm that has this ability is therefore very desirable. Such an algorithm is subspace predictive control (SPC). This algorithm consists of a predictor that is derived using subspace identification theory [87], making it a data-driven control method. This subspace predictor is subsequently integrated into a predictive control objective function. The basic SPC algorithm was introduced by [30] and has since been used by various researchers [91, 49, 88]. If the subspace predictor is updated on-line with new input-output data when it becomes available, then SPC has the ability to adapt to changing system conditions, which can also include unanticipated faults. Besides having this ability, another important advantage of the SPC algorithm is that the issue of robustness with respect to model uncertainty is implicitly addressed because of the adaptation of the predictor. In [37] the SPC algorithm is used for FTC of the GARTEUR benchmark model.

2.6.9 Model Following

The model following method is another approach to active FTC. Basically, the method considers a reference model of the form

$$\begin{aligned} x_{k+1}^M &= A_M x_k^M + B_M r_k, \\ y_k^M &= x_k^M, \end{aligned}$$

where r_k is a reference trajectory signal. The goal is to compute matrices K_r and K_x such that the feedback interconnection of the open-loop system (2.25) and the state-feedback control action

$$u_k = K_r r_k + K_x x_k$$

matches the reference model. To this end the reference model and closed-loop system are written in the form

$$\begin{aligned} y_{k+1}^M &= A_M x_k^M + B_M r_k, \\ y_{k+1} &= (CA + CBK_x)x_k + CBK_r r_k, \end{aligned}$$

so that *perfect model following* (PMF) can be achieved by selecting

$$\text{PMF:} \begin{cases} K_x = (CB)^{-1}(A_M - CA), \\ K_r = (CB)^{-1}B_M, \end{cases} \tag{2.31}$$

provided that the system is square (i.e. $\dim(y) = \dim(u)$), and that the inverse of the matrix CB exists. When the exact system matrices (A,B) in (2.31) are unknown, they can be substituted by some estimated values $(\hat{A},\hat{B})$, resulting in *the indirect (explicit) method* [8]. The indirect method provides no guarantees for closed-loop stability, and in addition, the matrix $(C\hat{B})$ may not be invertible. In order to avoid the need for estimating the plant parameters, the *direct (implicit) method* of model following can be used, which directly estimates the controller gain matrices K_r and K_x by means of an adaptive scheme. Two approaches to direct model following exist, the output error method and the input error method. Examples of the application of the model following approach can be found in [8, 70, 85]. We note here, that the direct model following method is based on adaptation rules and as such is also a candidate for the group of adaptive control methods.

The model following methods have the advantage that they usually do not require an FDD scheme. A strong drawback is, however, that they are not applicable to sensor faults. In addition, these methods do not deal with model uncertainty.

2.6.10 Adaptive Control

Adaptive control methods form a class of methods that is very suitable for active FTC. Due to their ability to automatically adapt to changes in the system parameters, these methods could be called “self-reconfiguable”, i.e. they often don’t require the “reconfiguration mechanism” and “FDD” components, as in Figure 2.6. This, however is mostly true for component faults and actuator faults, but not for some sensor faults. If one, for instance, makes use of an adaptive control scheme based on output-feedback design to compensate for sensor faults it will make the faulty measurement (rather than the true signal) track a desired reference signal, and this in turn may even lead to instability. Indeed, in a case of a total sensor failure an adaptive controller may try to increase the control action to make the faulty measured signal equal to the desired value which will not be possible due to the complete failure of the sensor. In such cases an FDD scheme is needed to detect the sensor failure, and a reconfiguration mechanism would have to appropriately reconfigure the adaptive controller. We note here that the direct model following and MM approaches, discussed above, also belong to the class of adaptive control algorithms. LPV control methods for FTC design are also members of this class. In [51] LPV FTC methods are developed that deal with structured parametric and FDD uncertainty. Furthermore, these methods are applicable to a wide class of faults as the fault signal is allowed to enter the state-space matrices of the system in any way as long as the matrices remain bounded. Other applications of LPV control for FTC can be found, for example in [80, 32].

2.7 Comparison of Fault Tolerant Flight Control Methods

The table on the next page presents a comparison of the fault tolerant control methods, applicable for reconfigurable flight control, considered in this survey. Filled circles mean that the method has the indicated property while empty circles imply that an author has suggested that the approach could be modified to incorporate the property. The columns are explained as follows:

- Failures: Types of failures that the method can handle
- Robust: The method uses robust control techniques
- Adaptive: The method uses adaptive control techniques
- Fault Model:
  - FDI: An FDI algorithm is incorporated into the method
  - Assumed: The method assumes an algorithm which provides a fault model
- Constraints: The method can handle actuator constraints
- Model Type: The type of internal model used

The table also shows the fault tolerant control methodologies that have been selected for further evaluation in this action group. Their application in the different control designs using the GARTEUR FTFC benchmark and achieved real-time performances are described in the subsequent chapters of this book.

Method	Failures		Robust	Adaptive	Fault Model		Constraints	Model Type	
	Actuator	Structural			FDI	Assumed		Linear	Nonlinear
Multiple Model Switching and Tuning (MMST)		●		●	●			●	
Interacting Multiple Model (IMM)		●		●	●		○	●	
Propulsion Controlled Aircraft (PCA)	●		○			●		●	●
Control Allocation (CA)*	●					●	○	●	
Feedback Linearization	●	●		●	●				●
Sliding Mode Control (SMC)*	○[1]	●	●[2]				●		●
Eigenstructure Assignment (EA)		●				●		●	
Pseudo Inverse Method (PIM)		●				●		●	
Model Reference Adaptive Control (MRAC)*		●		●	●			●	○
Model Predictive Control (MPC)*	●	●	○	○	●	●	●	●	●

Comparison of reconfigurable control methods
* Evaluated in this Action Group
1: Can handle partial loss of effectiveness of actuators, but not complete loss
2: Assumes robust control can handle all forms of structural failures

References

1. Andry, A.N., Shapiro, E.Y., Chung, J.C.: Eigenstructure assignment for linear systems. IEEE Transactions on Aerospace Electronic Systems 19(5) (September 1983)
2. Aström, K.J., Wittenmark, B.: Adaptive control, 2nd edn. Addison-Wesley Publishing Company, Reading (1995)
3. Basseville, M.: On-board component fault detection and isolation using the statistical local approach. Automatica 34(11), 1391–1415 (1998)
4. Belkharraz, A.I., Sobel, K.: Fault tolerant flight control for a class of control surface failures. In: Proceedings of the American Control Conference, June 2000. IEEE, Los Alamitos (2000)
5. Blanke, M., Kinnaert, M., Lunze, J., Staroswiecki, M.: Diagnosis and fault-tolerant control, 2nd edn. Springer, Heidelberg (2006)
6. Bodson, M.: Multivariable adaptive algorithms for reconfigurable flight control. In: Proceedings of the 33rd Conference on Decision and Control, December 1994. IEEE, Los Alamitos (1994)
7. Bodson, M.: Evaluation of optimization methods for control allocation. Journal of Guidance, Control, and Dynamics 25(4), 703–711 (2002)
8. Bodson, M., Groszkiewicz, J.E.: Multivariable adaptive algorithms for reconfigurable flight control. IEEE Transactions on Control Systems Technology 5(2), 217–229 (1997)
9. Bordignon, K.A., Durham, W.C.: Closed-form solutions to constrained control allocation problem. Journal of Guidance, Control and Dynamics 18(5) (September 1995)
10. Boskovic, J.D., Li, S.M., Mehra, R.K.: Reconfigurable flight control design using multiple switching controllers and on-line estimation of damage-related parameters. In: Proceedings of the 2000 IEEE International Conference on Control Applications, September 2000. IEEE, Los Alamitos (2000)
11. Boskovic, J.D., Li, S.M., Mehra, R.K.: Study of an adaptive reconfigurable control scheme for tailless advanced fighter aircraft (TAFA) in the presence of wing damage. In: Position Location and Navigation Symposium, pp. 341–348. IEEE, Los Alamitos (2000)
12. Boskovic, J.D., Li, S.M., Mehra, R.K.: Robust supervisory fault-tolerant flight control system. In: Proceedings of the American Control Conference (June 2001)
13. Boskovic, J.D., Mehra, R.K.: A multiple model-based reconfigurable flight control system design. In: Proceedings on the 37th IEEE Conference on Decision & Control, December 1998. IEEE, Los Alamitos (1998)
14. Boskovic, J.D., Mehra, R.K.: Stable multiple model adaptive flight control for accommodation of a large class of control effector failures. In: Proceedings of the American Control Conference (June 1999)
15. Brinker, J.S., Wise, K.A.: Flight testing of reconfigurable control law on the X-36 tailless aircraft. Journal of Guidance, Control and Dynamics 24(5) (September 2001)
16. Burcham, F.W., Burken, J.J., Maine, T.A., Bull, J.: Emergency flight control using only engine thrust and lateral center-of-gravity offset: a first look. Technical report, NASA (1997)
17. Burcham, F.W., Burken, J.J., Maine, T.A., Fullerton, C.G.: Development and flight test of an emergency flight control system using only engine thrust on an MD-11 transport airplane. Technical report, NASA (October 1997)
18. Burken, J.J., Burcham, F.W.: Flight-test results of propulsion-only emergency control system on MD-11 airplane. Journal of Guidance, Control and Dynamics 20(5) (October 1997)

19. Calise, A.J., Hovakimyan, N., Idan, M.: Adaptive output feedback control of nonlinear systems using neural networks. Automatica 37(8) (March 2001)
20. Calise, A.J., Lee, S., Sharma, M.: Direct adaptive reconfigurable control of a tailless fighter aircraft. In: AIAA Guidance, Navigation and Control Conference, Boston, MA (August 1998)
21. Calise, A.J., Lee, S., Sharma, M.: Development of a reconfigurable flight control law for the X-36 tailless fighter aircraft. In: AIAA Guidance, Navigation, and Control Conference (August 2000)
22. Davidson, J.B., Andrisani, D.: Gain weighted eigenspace assignment. Technical report, NASA (May 1994)
23. Davidson, J.B., Andrisani, D.: Lateral-directional eigenvector flying qualities guidelines for high performance aircraft. Technical report, NASA (December 1996)
24. Davidson, J.B., Lallman, F.J., Bundick, W.T.: Real-time adaptive control allocation applied to a high performance aircraft. In: 5th SIAM Conference on Control & Its Applications (2001)
25. Demetriou, M.A.: Adaptive reorganization of switched systems with faulty actuators. In: Proceedings of the 40th IEEE Conference on Decision and Control (December 2001)
26. Duan, G.R.: Parametric eigenstructure assignment via output feedback based on singular value decompositions. IEE Proceedings - Control Theory and Applications 150(1), 93–100 (2003)
27. Ducard, G., Geering, H.P.: Efficient nonlinear actuator fault detection and isolation system for unmanned aerial vehicles. Journal of Guidance, Control, and Dynamics 31(1), 225–237 (2008)
28. Durham, W.C., Bordignon, K.A.: Multiple control effector rate limiting. Journal of Guidance, Control and Dynamics 19(1) (February 1996)
29. Enns, D.F.: Control allocation approaches. In: Proceedings of AIAA GNC Conference (August 1998)
30. Favoreel, W.: Subspace methods for identification and control of linear and bilinear systems. PhD thesis, Faculty of Engineering, K.U. Leuven, Belgium (1999)
31. Gao, Z., Antsaklis, P.: Stability of the pseudo-inverse method for reconfigurable control systems. International Journal of Control 53(3), 717–729 (1991)
32. Gáspár, P., Bokor, J.: A fault-tolerant rollover prevention system based on an LPV method. International Journal of Vehicle Design 42(3-4), 392–412 (2006)
33. Gertler, J.: Designing dynamic consistancy relations for fault detection and isolation. International Journal of Control 73(8), 720–732 (2000)
34. Gopinathan, M., Boskovic, J.D., Mehra, R.K., Rago, C.: A multiple model predictive scheme for fault-tolerant flight control design. In: Proceedings of the 37th IEEE Conference on Decision & Control, December 1998. IEEE, Los Alamitos (1998)
35. Groszkiewicz, J.E., Bodson, M.: Flight control reconfiguration using adaptive methods. In: Proceedings of the 34th Conference on Decision & Control, December 1995. IEEE, Los Alamitos (1995)
36. Hajiyev, C., Caliskan, F.: Fault diagnosis and reconfiguration in flight control systems. Kluwer Academic Publishers, Dordrecht (2003)
37. Hallouzi, R.: Multiple-model based diagnosis for adaptive fault-tolerant control. PhD thesis, Delft University of Technology (2008)
38. Härkegård, O.: Dynamic control allocation using constrained quadratic programming. Journal of Guidance, Control, and Dynamics 27(6), 1028–1034 (2004)
39. Huzmezan, M., Maciejowski, J.M.: Reconfiguration and scheduling in flight using quasi-LPV high-fidelity models and MBPC control. In: Proceedings of the American Control Conference (June 1998)

40. Huzmezan, M., Maciejowski, J.M.: Reconfigurable flight control of a high incidence research model using predictive control. In: UKACC International Conference on CONTROL (September 1998)
41. Idan, M., Johnson, M., Calise, A.J.: A hierarchical approach to adaptive control for improved flight safety. AIAA Journal on Guidance, Control and Dynamics (July 2001)
42. Idan, M., Johnson, M., Calise, A.J., Kaneshige, J.: Intelligent aerodynamic/propulsion flight control for flight safety: a nonlinear adaptive approach. In: American Control Conference, ACC (2001)
43. Isermann, R., Ballé, P.: Trends in the application of model-based fault detection and diagnosis of technical processes. Control Engineering Practice 5(5), 709–719 (1997)
44. Isidori, A.: Nonlinear control systems, 2nd edn. Springer, Heidelberg (1989)
45. Jiang, J.: Fault-tolerant control systems - an introductory overview. Acta Automatica Sinica 31(1), 161–174 (2005)
46. Johansen, T.A.: Operating regime based process modeling and identification. The Norwegian Institute of Technology, University of Trondheim, ph.d. thesis, itk-report 94-109-w edition (1994)
47. Johansen, T., Foss, B.: Identification of non-linear system structure and parameters using regime decomposition. Automatica 31(2), 321–326 (1995)
48. Johnson, E.N., Calise, A.J.: Neural network adaptive control of systems with input saturation. In: American Control Conference (ACC), Arlington, Virginia (June 2001)
49. Kadali, R., Huang, B., Rossiter, A.: A data driven subspace approach to predictive controller design. Control Engineering Practice 11(3), 261–278 (2003)
50. Kale, M.M., Chipperfield, A.J.: Stabilized MPC formulations for robust reconfigurable flight control. Control Engineering Practice 13(6), 771–788 (2005)
51. Kanev, S.: Robust fault-tolerant control. PhD thesis, University of Twente (2004)
52. Kanev, S., Verhaegen, M.: Controller reconfiguration for non-linear systems. Control Engineering Practice 8, 1223–1235 (2000)
53. Kanev, S., Verhaegen, M.: A bank of reconfigurable LQG controllers for linear systems subjected to failures. In: 39th IEEE Conference on Decision and Control (December 2000)
54. Kanev, S., Verhaegen, M., Nijsse, G.: A method for the design of fault-tolerant systems in case of sensor and actuator faults. In: European Control Conference, ECC (September 2001)
55. Kerrigan, E.: Fault-tolerant control of the COSY ship propulsion benchmark using model predictive control. Technical report, University of Cambridge (November 1998)
56. Keviczky, T., Balas, G.J.: Software-enabled receding horizon control for autonomous unmanned aerial vehicle guidance. Journal of Guidance, Control, and Dynamics 29(3), 680–694 (2006)
57. Kinnaert, M.: Fault diagnosis based on analytical models for linear and nonlinear systems - a tutorial. In: Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS 2003), Washington D.C., USA, pp. 37–50 (2003)
58. Konstantopoulos, I.K., Antsaklis, P.J.: Eigenstructure assignment in reconfigurable control systems. Technical report, Interdisciplinary Studies of Intelligent Systems (January 1996)
59. Konstantopoulos, I.K., Antsaklis, P.J.: An optimization approach to control reconfiguration. Dynamics and Control 9(3), 255–270 (1999)
60. Kothare, M.V., Balakrishnan, V., Morari, M.: Robust constrained model predictive control using linear matrix inequalities. Automatica 32(10), 1361–1379 (1996)

61. Liao, F., Wang, J.L., Yang, G.H.: Reliable robust flight tracking control: an LMI approach. IEEE Transactions on Control Systems Technology 10(1), 76–89 (2002)
62. Liu, W.: An on-line expert system-based fault-tolerant control system. Expert Systems with Applications 11(1), 59–64 (1996)
63. Liu, G., Patton, R.: Eigenstructure assignment for control systems design. John Wiley & Sons, Chichester (1998)
64. Maciejowski, J.M.: The implicit daisy-chaining property of constrained predictive control. Applied Math and Computer Science 8(4), 695–711 (1998)
65. Maciejowski, J.M.: Predictive control with constraints. Prentice Hall, Englewood Cliffs (2002)
66. Maciejowski, J.M., Jones, C.N.: MPC fault-tolerant flight control case study: Flight 1862. In: Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS 2003), Washington D.C., USA, pp. 121–126 (2003)
67. Mahmoud, M., Jiang, J., Zhang, Y.: Active fault tolerant control systems: stochastic analysis and synthesis. Springer, Berlin (2003)
68. Maybeck, P.S.: Multiple model adaptive algorithms for detecting and compensating sensor and actuator/surface failures in aircraft flight control systems. International Journal of Robust and Nonlinear Control 9, 1051–1070 (1999)
69. Mignone, D.: Control and estimation of hybrid systems with mathematical optimization. PhD thesis, Swiss Federal Institute of Technology, ETH (January 2002)
70. Morse, W., Ossman, K.: Model-following reconfigurable flight control system for the AFTI/F-16. Journal of Guidance, Control, and Dynamics 13(6), 969–976 (1990)
71. Narendra, K.S., Balakrishnan, J.: Adaptive control using multiple models. IEEE Transactions on Automatic Control 42(2) (February 1997)
72. Niemann, H., Stoustrup, J.: Passive fault tolerant control of a double inverted pendulum - case study. Control Engineering Practice 13(8), 1047–1059 (2005)
73. Noura, H., Sauter, D., Hamelin, F., Theilliol: Fault-tolerant control in dynamic systems: application to a winding machine. IEEE Control Systems Magazine 20(1), 33–49 (2000)
74. NTSB. Aircraft accident report - american airlines, inc. DC-10-10. Technical Report NTSB-AAR-79-17, National Transpotration Safety Board, USA (1979)
75. Patton, R.: Fault tolerant control: the 1997 situation. In: Proceedings of the 3rd Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS 1997), pp. 1033–1054. Hull University, Hull (1997)
76. Prakash, J., Narasimhan, S., Patwardhan, S.C.: Integrating model based fault diagnosis with model predictive control. Industrial & Engineering Chemistry Research 44(12), 4344–4360 (2005)
77. Rauch, H.: Intelligent fault diagnosis and control reconfiguration. IEEE Control System Magazine 14(3), 6–12 (1994)
78. Ru, J., Li, X.R.: Variable-structure multiple-model approach to fault detection, identification, and estimation. IEEE Transactions on Control Systems Technology 16(5), 1029–1038 (2008)
79. Seguchi, H., Ohtsuka, T.: Nonlinear receding horizon control of an underactuated hovercraft. International Journal of Robust and Nonlinear Control 13(3-4), 381–398 (2003)
80. Shin, J.-Y., Belcastro, C.M.: Performance analysis on fault tolerant control system. IEEE Transactions on Control Systems Technology 14(5), 920–925 (2006)
81. Shtessel, Y.B.: Sliding mode control: overview and applications to aerospace control. Talk notes (2001)
82. Shtessel, Y.B., Buffington, J.: Multiple time scale flight control using reconfigurable sliding modes. AIAA Journal on Guidance, Control and Dynamics 22(6), 873–883 (1999)

83. Slotine, J.J.E., Li, W.: Applied Nonlinear Control. Prentice-Hall International, Inc., Englewood Cliffs (1991)
84. Stoustrup, J., Blondel, V.D.: Fault tolerant control: A simultaneous stabilization result. IEEE Transactions on Automatic Control 49(4), 305–310 (2004)
85. Tao, G., Chen, S., Joshi, S.: An adaptive actuator failure compensation controller using output feedback. IEEE Transactions on Automatic Control 47(3), 506–511 (2002)
86. Tao, G., Ma, X., Joshi, S.: Adaptive state feedback and tracking control of systems with actuator failures. IEEE Transactions on Automatic Control 46(1), 78–95 (2001)
87. Verhaegen, M., Verdult, V.: Filtering and system identification: an introduction. Cambridge University Press, Cambridge (2007)
88. Wang, X., Huang, B., Chen, T.: Data-driven predictive control for solid oxide fuel cells. Journal of Process Control 17(2), 103–114 (2007)
89. Wang, G.S., Lv, Q., Liang, B., Duan, G.R.: Design of reconfiguring control systems via state feedback eigenstructure assignment. International Journal of Information Technology 11(7), 61–70 (2005)
90. Wise, K.A., Brinker, J.S., Calise, A.J., Enns, D.F., Elgersma, M.R., Voulgaris, P.: Direct adaptive reconfigurable flight control for a tailless advanced fighter aircraft. International Journal of Robust and Nonlinear Control 9(14), 999–1022 (1999)
91. Woodley, B.R., How, J.P., Kosut, R.L.: Subspace based direct adaptive $\mathscr{H}_\infty$ control. International Journal of Adaptive Control and Signal Processing 15, 535–561 (2001)
92. Yen, G.G., Ho, L.-W.: Online multiple-model-based fault diagnosis and accommodation. IEEE Transactions on Industrial Electronics 50(2), 296–312 (2003)
93. Zhang, Y., Jiang, J.: An interacting multiple-model based fault detection, diagnosis and fault-tolerant control approach. In: Proceedings of the 38th Conference on Decision & Control (December 1999)
94. Zhang, Y., Jiang, J.: Integrated design of reconfigurable fault-tolerant control systems. Journal of Guidance 24(1), 133–136 (2000)
95. Zhang, Y.M., Jiang, J.: Fault tolerant control system design with explicit consideration of performance degradation. IEEE Transactions on Aerospace and Electronic Systems 39(3), 838–848 (2003)
96. Zhang, Y., Jiang, J.: Issues on integration of fault diagnosis and reconfigurable control in active fault-tolerant control systems. In: Proceedings of the IFAC SAFEPROCESS, Beijing, China (August 2006)
97. Zhang, D., Wang, Z., Hu, S.: Robust satisfactory fault-tolerant control of uncertain linear discrete-time systems: an LMI approach. International Journal of Systems Science 38(2), 151–165 (2007)
98. Zhenyu, Y., Huazhang, S., Zongji, C.: The frequency-domain heterogeneous control mixer module for control reconfiguration. In: Proceedings of the 1999 IEEE International Conference on Control Applications, August 1999. IEEE, Los Alamitos (1999)