Multivariable Adaptive Algorithms for Reconfigurable Flight Control

Marc Bodson*
Department of Electrical Engineering
University of Utah
Salt Lake City, UT 84112
and
Joseph E. Groszkiewicz
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

The application of multivariable adaptive control techniques to flight control reconfiguration is considered. The objective is to redesign automatically flight control laws to compensate for actuator failures or surface damage. Three adaptive algorithms for multivariable model reference control are compared. The availability of state measurements in this application leads to relatively simple algorithms. The respective advantages and disadvantages of the adaptive algorithms are discussed, considering their complexity and the assumptions that they require. An equation-error based algorithm is found to be preferable. Simulations obtained using a full nonlinear model of a twin-engine jet aircraft are presented. The results demonstrate the ability of the adaptive algorithms to maintain trim after a failure, to restore tracking of the pilot commands despite the loss of actuator effectiveness, and to coordinate the use of the remaining active control surfaces in order to guarantee the decoupling of the rotational axes. A new adaptive algorithm with a variable forgetting feature is also used and is found to yield a useful alternative to covariance resetting as a solution to covariance wind-up in least-squares algorithms.

1 Introduction

Reconfiguration is likely to be a feature of future generations of flight control systems. The main motivation for reconfiguration is greater survivability, attained through the ability of the feedback sys-

*Effort sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant numbers F49620-92-J-0386 and F49620-95-1-0341. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation. The views and conclusions contained in the paper are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. Government.

tem to reorganize itself in the presence of actuator failures and surface damage. High-performance aircrafts are often unstable to the point of exceeding the control capabilities of a human pilot. Instability follows either from requirements of maneuverability, or from efficiency considerations, and it is expected to become an increasingly common characteristic of future aircrafts. The need for fault tolerant control methods is therefore critical. The benefits of reconfigurable flight control systems extend beyond the immediate considerations of safety. Indeed, reconfigurable systems reduce the need for other forms of reliability, such as redundant actuators. Therefore, increased maintainability and reduced costs are expected to result from this technology [5].

Two main approaches can be distinguished for flight control reconfiguration. This first is based on the concept of failure detection and identification. The resulting system consists in a fast and efficient method to detect the failure among a set of pre-planned conditions, and in procedures to handle each of the cases. This approach works well in restricted cases, but suffers from significant drawbacks. The first is that, as the number of failures grows, it becomes increasingly difficult and time-consuming to carry out the detection and classification. Even with a large number of pre-planned failures, there is also no reason to believe that a failure that has not been categorized will not cause the whole system to fail. In the case of flight control, there are multiple possible actuator failures (multiple actuators and multiple failure modes, such as locked or floating) and an infinite variety of possible surface damages. In addition, because failure detection relies on models of the unfailed system, any discrepancy between the model and reality can lead to false detection. Because of the nonlinearity and complexity of aircraft dynamics (especially engine dynamics and aerodynamics), this is a nontrivial problem in reconfigurable flight control.

A totally different approach to the problem of flight control reconfiguration consists in identifying the dynamic behavior of the aircraft in real-time, and in designing a controller automatically. Because such an approach does not rely on failure classification, it is expected that the resulting system will tolerate a larger class of failures, including some that may not have been anticipated. In this paper, we discuss several multivariable adaptive control algorithms that may be used with that objective in mind. We make assumptions that are realistic in the flight control problem, yet allow to considerably simplify the algorithms available in the literature. We also present the results of a simulation study using a detailed nonlinear model of a twin-engine aircraft. The results demonstrate the ability of the adaptive algorithms to maintain trim after a failure, to restore tracking of the pilot commands despite the loss of actuator effectiveness, and to coordinate the use of the remaining active control surfaces in order to guarantee the decoupling of the rotational axes.

1.1 Failure Detection and Classification

In 1984, the U.S. Air Force began a program called the *Self-Repairing Flight Control System Program* [12]. The objectives of the program were to "improve the reliability, maintainability, survivability and life cycle cost" of aircraft. There were two main thrusts involved in achieving the goals of the program. The first was the development of reconfigurable flight control systems (FCS), meaning control systems that are modified to compensate for changing conditions. The changing conditions consisted in a failure of the airplane's control surfaces or sensors, or in damage to the aircraft. The second component of the program was the use of on-line diagnostics to identify failures. Improvements would come as reductions in maintenance and repair costs.

In the reconfigurable FCS, the failure was identified through a Failure Detection and Isolation (FDI) procedure. A local FDI algorithm was used to detect actuator failures, while a global FDI algorithm determined surface damage. The global FDI used a model of the aircraft to compare the measured output with the expected output. The error between the two was passed through several filters. The output of each filter represented a likelihood that the failure represented by the filter had occurred. These likelihoods were used by a Pseudo-Surface Resolver (PSR) to determine how control of the aircraft could be maintained. The PSR used a "modified pseudo-inverse that minimizes changes in control deflections after failure to maintain forces and moments" [12]. The maintenance diagnostics worked in a similar manner. The approach proved successful in flight tests and was capable of handling various types of failures of the right stabilator. Urnes, Yeager & Stewart [28] concluded that "the test results of the Self-Repairing Flight Control System installed on an F-15 aircraft indicate high potential for the concepts evaluated."

Another major study of reconfigurable flight control systems was sponsored by NASA Langley and carried out by Alphatech (*cf.* [30]). The study considered the application of reconfiguration strategies to stable, commercial aircraft (Boeing 737). A single flight condition was assumed but a large variety of possible failures were simulated. The approach was the precursor of the SRFCS approach described above, consisting of two main components: a failure detection & identification module, and a control reconfiguration module. The control design in the NASA/Alphatech study was not based on the approximation of the unimpaired control actions, but on a redesign of the control law using linear quadratic (LQ) regulator theory. This approach is described in [17] and consists in specifying weighting matrices in an LQ problem so that the resulting closed-loop system satisfies some bandwidth constraints. Only changes in the B matrix of the state-space representation were considered, but the study included the possibility of incorporating knowledge of uncertainties in the estimated B matrix in the design.

The Alphatech project also studied extensively the problem created by changes in trim conditions, which act as constant disturbances to be added to the state-space model. An automatic trim algorithm was developed, based on an optimization procedure, and is reported in [31]. Of related interest is the work of Ostroff [23], which also considers the automatic control redesign for a Boeing 737 aircraft, but suggests the incorporation of integral action in the control law to solve the trim problem (essentially considering it as a disturbance rejection problem). Simulations for a mildly unstable aircraft model are also reported in this work.

Several other researchers have worked on design methods based on linear quadratic (LQ) techniques, assuming that the detection problem was solved independently. Huang & Stengel [15] presented an automatic redesign method based on implicit model following, incorporating integral action. Moerder *et al.* [20] studied the application of LQ controllers, but assumed that control gains were to be scheduled according to the decision of the failure detection and identification logic (as opposed to being calculated in real-time).

Of related interest is the work of Maybeck & Stevens [18], which suggests a somewhat different approach. While assuming that the possible failures have been categorized, the method relies on a bank of Kalman filters to estimate the states of the system based on the different assumptions. Residual errors are used to calculate the probabilities of individual failures and the control input is the weighted average of the signals calculated under these respective assumptions. This is significantly different from the SRFCS approach where the control input corresponding to the most likely failure is chosen.

In [19], a multiple model adaptive estimation approach is used. Failures are represented as a vector of unknown system parameters, related to the effectiveness of various actuators and sensors. Fault detection is achieved through the use of filters, where each filter is essentially a model of the VISTA F-16 in which a certain actuator or sensor has failed. How well the filter response and the airplane response match gives an estimate of how likely it is that the particular failure has occurred. If a partial failure has occurred, then the probability that the failure has occurred is taken to represent the loss of effectiveness of the sensor or control surface. Each failure has a set of control law gains associated with it. The controller combines the gains based on the probabilities generated. One of the difficulties of this approach is that there must be a non-zero input to the system. If not, false failures are sometimes detected, or real failures are not detected. Normally, the commands necessary to perform a maneuver are sufficient to excite the system. During steady level flight, however, there is not sufficient excitation. The solution to this problem was to add a small sinusoidal input to the system to ensure sufficient excitation.

1.2 Nonlinear and Adaptive Control

In contrast to the methods described above, several researchers have searched for methods that do not depend on the identification of the failures before taking action. A variety of directions have been pursued.

A logical first step consists in looking for a robust linear control law that would be satisfactory for all possible impaired aircrafts, and would achieve the required performance for the unimpaired aircraft. Schneider, Horowitz & Houpis [25] considered the use of quantitative feedback theory for that purpose. However, Chandler [6] illustrated with several examples that it is generally not possible to design a robust linear control law that guarantees stability for the impaired conditions while providing satisfactory performance for the nominal unfailed conditions. He advocated the design of a robust control law as a first line of defense to failures, giving time for the reconfigurable control law to take action, but implied that some form of reconfiguration, for example nonlinear or adaptive control, would be necessary.

Dittmar [10] investigated the use of an adaptive control approach based on hyperstability and the algorithms developed by Landau [16]. A simulation study concluded that the performance was equal or better than the SRFCS scheme, and could do so with less computer memory while accommodating a larger number of failure modes.

Morse & Ossman [21] also considered an adaptive control approach for the AFTI/F-16, using algorithms of Sobel & Kaufmann [26]. The authors developed their own design method for the selection of the parameters of the algorithms and showed that the method was successful even in the presence of multiple failures.

Gross & Migyanko [13] considered the use of "supercontroller" technology, which is a form of nonlinear control based on polynomial networks. Coefficients of the polynomials were adjusted using an optimization program and a data base of optimal responses. This method was further developed in [1] and connected to recent work in neural networks. Sofge & White [27] also mention efforts at McDonnell Douglas in the neural network area. It is interesting to note that while the implementation of the controllers does not rely on explicit failure recognition, the training of the networks does, so that this method can be considered a hybrid between the two approaches discussed in this brief overview of the literature on reconfigurable flight control systems.

Research on the application of adaptive methods to reconfigurable control has also been recently carried out at Wright-Patterson Air Force Base [7], [8], [9]. The emphasis of the research has been on the use of constrained least-squares identification methods and model predictive control. The studies have addressed the control of single-input single-output pitch axis models of an unstable aircraft, and have successfully included actuator rate saturation in the design, as well as prior information on the stability derivatives.

2 Adaptive Control Algorithms

2.1 Aircraft Model

The kinematic behavior of an airplane is governed by a set of nonlinear differential equations. For flight control system design, these equations can be approximated effectively by a set of linear differential equations.

$$\begin{aligned} \dot{x} &= Ax + Bu + d \\ y &= Cx \end{aligned} \tag{1}$$

A disturbance term d is included to account for the trim values of the input necessary to maintain steady flight at the operating point. By explicitly including this disturbance, the reconfigurable flight control system can automatically calculate the trim, freeing the pilot from having to. Because the trim values may change radically after a failure, such a feature is quite important.

The states of the aircraft are represented by x. The longitudinal states are α (angle of attack), q (pitch rate), h (altitude), and v (velocity). The lateral states are β (sideslip), p (roll rate), r (yaw rate), ϕ (roll angle), and ψ (yaw angle). For the design of stability augmentation flight control systems, the state vector can be reduced to only five states: α, q, β, p, r. The slow dynamics related to angular motions, and the actuator dynamics, constitute unmodelled dynamics against which the control system must be robust. The control inputs u are also divided into lateral and longitudinal inputs. The longitudinal inputs are δ_E (elevator command) and δ_T (thrust command). The lateral inputs are δ_A (aileron command) and δ_R (rudder command). For stability augmentation flight control system design, δ_T is usually not considered.

There are several choices available for the control output y. The states q, p, and r are good choices for low dynamic pressure and limited angle of attack (*cf.* [14]). Generally, the problem is that of a three-input, three-output, linear time-invariant system. Because of symmetry in the unfailed aircraft, the longitudinal and lateral axes are usually decoupled. However, after a failure the airplane is usually no longer symmetric. Therefore, the longitudinal and lateral axes cannot be decoupled.

2.2 Adaptive Control Algorithms

Several adaptive algorithms can be used for reconfigurable flight control. The control objective considered here is based on model reference control. A motivation is that this objective allows us to easily incorporate considerations of tracking and decoupling in the design. Further, very simple algorithms are obtained. Several adaptation mechanisms have been proposed in the literature (see [24]). Those presented here are modified slightly to include the constant disturbance d, and to exploit the fact that all the states and their derivatives are measured. State variable filters are not needed to reconstruct the state and a state feedback control law can be used.

Assumptions and Reference Model

We consider the state-space model for the plant (1), where $x \in R^n$, $u \in R^m$, $y \in R^m$, and $d \in R^n$. We assume that the whole state x is available for measurement, although only the output y is to be tracked. The objective is for y to match the output y_M of a reference model

$$\dot{y}_M = A_M\, y_M + B_M\, r \tag{2}$$

where $y_M \in R^m$ and $r \in R^m$. The matrices A_M and B_M are arbitrary square matrices, with A_M stable. For the model reference control problem to have a relatively simple solution, we assume:

Assumption 1: The plant has relative degree 1, *i.e.* $\det(CB) \neq 0$.

Assumption 2: The plant transfer function is minimum phase, *i.e.* the zeros of transmission of the system are in the open left-half-plane.

The first assumption guarantees that the closed-loop transfer function of the plant can be made to match the transfer function of the reference model (2) using a proper compensator. If the assumption is not satisfied, the model reference control problem may still be solvable, but a more complex reference model would have to be chosen, so as to match the so-called *Hermite normal form* of the plant (*cf.* [24]). When the first assumption is satisfied, this Hermite form is simply $H(s) = \text{diag}\{1/s\}$, which means that the behavior of the plant at infinity is that of a multivariable integrator. The matrix CB is called the *high-frequency gain matrix* of the plant and is usually denoted K_P in the adaptive control literature. It is a critical parameter for adaptive algorithms.

The second assumption is a necessary assumption to guarantee the internal stability of the model reference control algorithm. The dimension of the state-space for the reference model is m, while the dimension of the state-space for the plant is n. Therefore $n - m$ modes must be made unobservable or uncontrollable. It can be shown that the model reference control law places m modes of the plant at the desired model reference locations, and makes the others unobservable by placing them at the locations of the transmission zeros.

Model Reference Control Law

We consider the state feedback control law

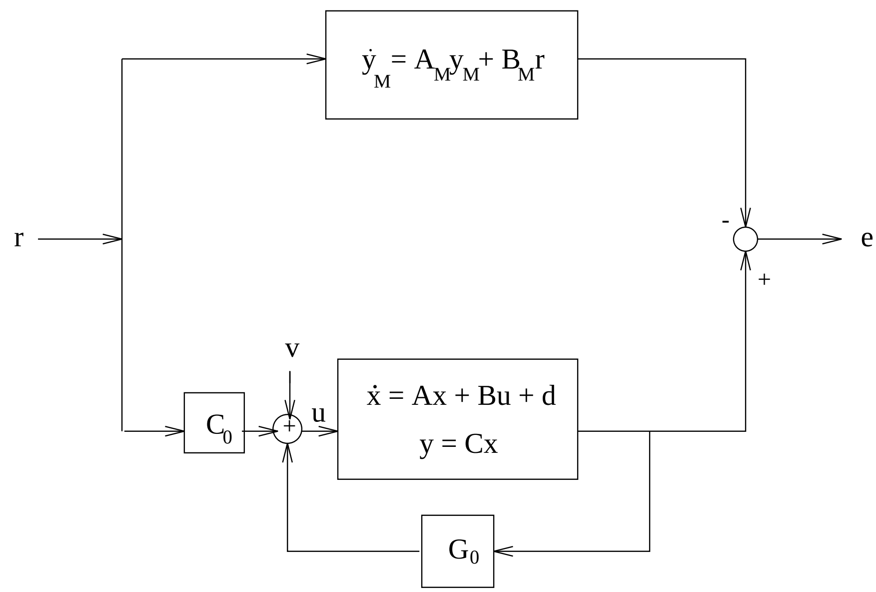

Figure 1: Model Reference Control Loop

$$u = C_0 r + G_0 x + v \tag{3}$$

where $C_0 \in R^{m \times m}, G_0 \in R^{m \times n}, v \in R^m$ are free controller parameters. The control law is represented in Figure 1. The closed-loop dynamics are given by

$$\begin{aligned} \dot{x} &= Ax + BG_0 x + Bv + d \\ y &= Cx \end{aligned} \tag{4}$$

or, in terms of the output y,

$$\dot{y} = (CA + CBG_0)x + CBC_0 r + CBv + Cd \tag{5}$$

(5) leads to the same input/output relationship as that of the reference model (2) for the so-called *nominal* values of the controller parameters

$$C_0^* = (CB)^{-1} B_M$$

$$
\begin{aligned}
G_0^* &= (CB)^{-1}(A_M C - CA) \\
v^* &= -(CB)^{-1}(Cd)
\end{aligned} \tag{6}
$$

If the plant was known, these would be the controller parameters that one would use to achieve the model reference control objective.

Indirect Adaptive Control Indirect adaptive control involves two stages. First, estimates of the plant parameters A, B, and d are generated. Once the plant parameters have been estimated, the estimates are used to generate controller parameters. If $\hat{A}$, $\hat{B}$, $\hat{d}$ are the estimates of A, B, d, an error vector can be defined by

$$
e_1 = \hat{A}x + \hat{B}u + \hat{d} - \dot{x} \tag{7}
$$

Given the assumptions, the error vector can also be expressed as

$$
e_1 = (\hat{A} - A)x + (\hat{B} - B)u + (\hat{d} - d) \tag{8}
$$

A least-squares algorithm can be used to find the estimates of A, B, and d which minimize the sum of the norm of e_1 evaluated at several sampling instants. Each row of (7) can be treated independently in this process. The algorithm requires the measurement of $\dot{x}$. The derivative can often be obtained by filtered differentiation of x. For flight control systems, differentiation is usually not needed, as the derivatives can be reconstructed from accelerometer measurements.

Once the estimates are obtained, the controller parameters C_0, G_0, and v in (3) are obtained from

$$
\begin{aligned}
C_0 &= (C\hat{B})^{-1}B_M \\
G_0 &= (C\hat{B})^{-1}(A_M C - C\hat{A}) \\
v &= -(C\hat{B})^{-1}(C\hat{d})
\end{aligned} \tag{9}
$$

This is the same relationship as (6), replacing the plant parameters with their estimates. In addition to the issue of stability, a major question must be resolved, namely what to do when $C\hat{B}$ is singular.

There are few methods which satisfactorily address this problem, from a practical point-of-view, but it has been a subject of recent research in the adaptive control field.

Direct Adaptive Control Instead of estimating the plant parameters, a direct adaptive control algorithm estimates the controller parameters C_0, G_0, and v. We discuss two main approaches: output error and input error.

Output Error The output error $e_0 = y - y_M$ is defined as the difference between the plant output and the output from the reference model, when given the same input as the plant. Because the system is assumed to have relative degree 1, there is no need for the so-called augmented error, and a simple algorithm results. The following fact is used.

Fact 1: *The output error* e_0 *satisfies*

$$e_0 = (sI - A_M)^{-1}(CB)[(C_0 - C_0^*)r + (G_0 - G_0^*)x + (v - v^*)] \tag{10}$$

Proof: Define

$$\dot{x}_M = A_M x_M + BC_0^* r + BG_0^* x_M + Bv^* + d \tag{11}$$

so that $y_M = Cx_M$ for some appropriate choice of initial conditions for x_M. Since $\dot{e}_0 = C\dot{x} - \dot{y_M} = C\dot{x} - A_M C x_M - B_M r$, along with (6), we find

$$\dot{e}_0 = (CA + (CB)G_0^*)(x - x_M)(CB)[(C_0 - C_0^*)r + (G_0 - G_0^*)x + (v - v^*)] \tag{12}$$

which leads to (10).

Equation (10) can be expressed compactly in the form

$$e_0 = (sI - A_M)^{-1}(CB)[\Phi \cdot w] \tag{13}$$

where

$$\Phi = ((C_0 - C_0^*)\ (G_0 - G_0^*)\ (v - v^*)) \tag{14}$$

is the $m\,\mathrm{x}\,(m+n+1)$ matrix of parameter errors, and the *regressor vector* $w^T = (r^T, x^T, 1)$ has dimension $m+n+1$. The product $\Phi \cdot w$ is carried out in the time domain, and the resulting signal is applied to the linear time-invariant operator $(sI - A_M)^{-1}(CB)$.

Fact 2: The update law $\dot{\Phi} = -Ge_0 w^T$ leads to an adaptive system that is stable in the sense of Lyapunov, with the property that e_0 tends to zero as $t \to \infty$, provided that $(sI - A_M)^{-1}$ is a strictly positive real transfer function matrix, and $(CB)^T G^{-1}$ is a positive definite matrix.

Proof: follows from Kalman-Yacubovich-Popov lemma (*e.g.* [22]).

This algorithm is similar to other algorithms, such as the one available in [22], with some adjustments needed because of the constant disturbance d and the state measurement. The strictly positive real condition can be satisfied by choosing A_M so that $A_M + A_M^T$ is negative definite. The other condition requires prior knowledge of the high-frequency gain matrix CB, and an appropriate choice for the adaptive gain matrix G.

Input Error Another formulation of the direct adaptive algorithm uses the following fact.

Fact 3: The following identity is satisfied for all time.

$$u = C_0^* B_M^{-1}(\dot{y} - A_M y) + G_0^* x + v^* \tag{15}$$

Proof: From (1), we find

$$\dot{y} - A_M y = CAx + CBu + Cd - A_M Cx \tag{16}$$

Using the nominal parameter values in (6), we get

$$\dot{y} - A_M y = (CB)(-G_0^* x + u - v^*) \tag{17}$$

from which (15) follows.

A new error equation can be defined from (15).

$$e_2 = C_0 B_M^{-1}(\dot{y} - A_M y) + G_0 x + v - u \tag{18}$$

We call e_2 the *input error.* An adaptive scheme based on this error belongs to the class of so-called equation-error based schemes. Given the assumptions, (18) can be expressed as

$$e_2 = \Phi \cdot z \tag{19}$$

where Φ is the controller parameter error defined in (14), and z is a new regressor vector defined as

$$z = \begin{pmatrix} B_M^{-1}(\dot{y} - A_M y) \\ x \\ 1 \end{pmatrix} \tag{20}$$

The main difference between the error equations (19) and (13) is the absence of the transfer function between the parameter error and the error signal. This eliminates the strictly positive real condition necessary for the stability of the algorithm, including the condition on the high-frequency gain matrix. Also, it makes possible the use of least-squares algorithms.

Comparison The three algorithms presented above all achieve the same model reference control objective, but with different structures and different assumptions. The following issues should be considered:

Number of Parameters: Both direct methods estimate the controller parameters C_0, G_0, and v. These have a total of $m^2 + mn + m$ elements. The indirect algorithm estimates A, B and d, which have a total of $n^2 + mn + n$ elements. Since $m < n$ in general, the direct algorithms estimate fewer parameters. For the reduced-order aircraft model, $n = 5$ and $m = 3$. The indirect algorithm estimates 45 parameters, while the direct algorithms only estimate 27.

Prior Information: The direct output error algorithm requires the most restrictive assumptions, by imposing a positive definiteness condition on the product of the transpose of the high-frequency gain matrix with the inverse of the adaptation gain matrix. There is no obvious way to enforce this condition. For all practical purposes, this condition is a symmetry and positive definiteness condition on the high-frequency gain matrix itself, and is not easily guaranteed. The direct input error and indirect algorithms place less stringent conditions on the high-frequency gain matrix. The indirect algorithm requires that the estimate of the high-frequency gain matrix be nonsingular at all times. It can be shown that the direct input error algorithm requires that the parameter C_0 must be nonsingular at all times. In [11], it was shown that the stability of the overall adaptive system could be guaranteed by incorporating a modification based on a sort of hysteresis into the algorithm with a least-squares update, and requires an upper bound on the norm of the high-frequency gain matrix.

Adaptation Algorithms: The indirect and direct input error approaches can be used with least-squares algorithms, in addition to gradient algorithms. The least-squares algorithms, which are faster and more efficient, can be used in their batch or recursive forms. The batch forms lend themselves to monitoring of the estimation quality [2].

Flexibility: The indirect algorithm is the most flexible of the three approaches. Other control strategies than model reference can be used, such as model predictive control. The direct algorithms are not easily modified away from the reference model formulation. The indirect algorithm also has the flexibility to incorporate prior knowledge on the plant parameters [8].

Given these considerations, it was decided that the direct input error algorithm was best suited to the problem of reconfigurable flight control. Given the large number of computations required, it was felt that fewer parameters was a significant advantage. Also, the least-squares algorithms are known to converge much faster than the gradient algorithms. This is especially important considering that rapid reconfiguration may be critical. A positive definiteness condition on the high-frequency gain matrix could also hardly be justified in this application. Therefore, the output error algorithm was not further considered. For the remainder of this paper, it is implied that the direct input error formulation of adaptive control is being used.

Least-Squares and Data Forgetting

Given a set of linear equations with unknown coefficients, the least-squares algorithm constitutes a fast and efficient way to find the set of coefficients which most closely match the equations. The direct input error algorithm is easily formulated so that least-squares estimation can be used. Define the regressor vector z as in (20), and the parameter matrix θ as

$$\theta = \begin{pmatrix} C_0^T \\ G_0^T \\ v^T \end{pmatrix} \tag{21}$$

where C_0, G_0, and v are the control law parameters in (3). The input error e_2 can then be expressed as

$$e_2 = \theta^T z - u \tag{22}$$

where u is the system input.

For problems where parameters vary and adaptation is needed, a forgetting factor is usually introduced. Unfortunately, the least-squares algorithm with forgetting factor becomes unstable if there is insufficient excitation. This situation is expected to occur with aircraft, as steady level flight does not provide adequate excitation for convergence of the parameters. The regressor vector z must be *persistently exciting* in order to guarantee convergence of the parameters to the nominal values [24]. One possible solution is to add a small perturbation to the controls, such as white noise. However, it will affect the flight of the aircraft. Another solution is the use of covariance resetting. This modification induces sharp discontinuities and transients in the responses of the algorithm.

A stabilized version of the least-squares with forgetting factor was derived in [3], based on a concept proposed in [29]. The algorithm was obtained by including an additional term in the error function used by the least-squares

$$J(\theta[N]) = \sum_{k=1}^{N} |\theta^T[N]z[k] - u[k]|^2 \lambda^{N-k} + \alpha|\theta[N] - \theta[N-1]|^2 \tag{23}$$

The additional term penalizes changes in the parameter matrix, θ. Setting $\frac{\partial J}{\partial \theta[N]} = 0$ yields

$$\theta[N] = \left(\sum_{k=1}^{N} z[k]z^T[k]\lambda^{N-k} + \alpha I \right)^{-1} \left(\sum_{k=1}^{N} z[k]u^T[k]\lambda^{N-k} + \alpha\theta[N-1] \right) \tag{24}$$

This is the equivalent of the batch solution of the least-squares, but it is not truly a batch solution, because of the dependence on $\theta[N-1]$. The matrix

$$P[N] = \left(\sum_{k=1}^{N} z[k]z^T[k]\lambda^{N-k} + \alpha I \right)^{-1} \tag{25}$$

is defined as the covariance matrix of this algorithm. A recursive form for the inverse of the covariance is

$$P^{-1}[N] = \lambda P^{-1}[N-1] + z[N]z^T[N] + \alpha(1-\lambda)I \tag{26}$$

with the initial condition $P^{-1}[0] = \alpha I$. The recursive formula for θ is

$$\theta[N] = \theta[N-1] + P[N]z[N](u^T[N] - z^T[N]\theta[N-1]) + \alpha\lambda P[N](\theta[N-1] - \theta[N-2]) \qquad (27)$$

One problem is to transform (26) into a recursion for $P[N]$. The recursive least-squares with forgetting factor algorithm makes use of the matrix inversion lemma

$$(A + BC)^{-1} = A^{-1}B(I + CAB)^{-1}CA^{-1} \qquad (28)$$

in the recursive update of $P[N]$. The inversion of the matrix $(I + CAB)$ is simplified because the product $z^T[N]P[N-1]z[N]$ is a scalar. For the update law (26), the matrix inversion lemma can still be used, but with $B = C^T$ defined as

$$B = \begin{pmatrix} z[N] & \sqrt{\alpha(1-\lambda)}I \end{pmatrix} \qquad (29)$$

However the product $B^T P[N-1]B$ is not a scalar, but a matrix whose dimension is one greater than $P[N]$. It would be easier to update $P^{-1}[N]$ and invert it than to update $P[N]$.

An alternate solution is to replace $\alpha(1-\lambda)I$ in the update law by $p\alpha(1-\lambda)e_i e_i^T$, where e_i is a vector of zeros, except for the i^{th} position which is one. p is the dimension of z, that is, the number of parameters in each row of the parameter matrix. As time progresses, i is incremented and returned to one when the end of the vector is reached. The matrix B is given by

$$B = \begin{pmatrix} z[N] & \sqrt{p\alpha(1-\lambda)}e_i \end{pmatrix} \qquad (30)$$

With this modification, the matrix $B^T P[N-1]B$ is only 2 x 2, which is easily inverted. Averaging analysis [3] shows that the averaged system responses are identical for both implementations.

The stabilized recursive least-squares with forgetting factor has the property that the covariance matrix and its inverse are bounded [3]. The only condition is that $z[N]$ must be bounded. The approximate algorithm using (30) was used in this research, in part because it only required the inversion of a matrix of size 2x2, while the dimension would otherwise have been 9x9.

3 Implementation

3.1 Aircraft Model, Assumptions, and Design Considerations

Simulations were carried out using a detailed simulation of a twin-engine aircraft, developed at NASA-Dryden [4]. The model is a complete nonlinear aircraft simulation, including full envelope aerodynamics, atmospheric model, detailed engine dynamics, and actuator dynamics. The reconfigurable control system design, on the other hand, is based on the reduced-order model using the five states α,q, β, p, r. The control inputs are denoted δ_H, δ_A, and δ_R. There is a cross-feed between aileron command and elevator command. Specifically, the actual elevator command δ_E is the sum of the symmetric deflection δ_H and an antisymmetric deflection set to $\frac{5}{6}\delta_A$. For the reconfigurable control law, there is nothing that forces the same reduction of the five independent control surfaces to three control inputs (*i.e.*, it is not necessary to keep the same coupling matrix). However, it was found convenient to keep the same structure for compatibility with the original control law, and because treating the five control surfaces as independent inputs would require that these surfaces be actuated by linearly independent signals for the parameters of the B matrix to be identifiable.

The controlled outputs are chosen to be q, p, r. It was checked that this choice corresponded to minimum phase zeros for the flight condition under consideration. In general, it is possible to enforce minimum phase properties by replacing q and r by $q+K_\alpha\alpha$ and $r-K_\beta\beta$ [14]. A justification is that one has, approximately, $\dot{\alpha} = q$ and $\dot{\beta} = -r$ (yet, the precise location of the transmission zeros depend on the other coefficients in the A and B matrices). The choice of q, p, and r as tracked outputs leads to a CB matrix that is a 3x3 matrix whose elements are the 3 moments created by each of the 3 control inputs. As long as the three vectors of moments are linearly independent (*i.e.*, moments in all three directions can be independently created), the matrix CB is nonsingular. This is usually the case, so that the assumptions under which the algorithms were derived are satisfied.

3.2 Reference Model and Autopilot

The objective of the model reference control law is for the airplane with feedback to have dynamics which approximate those of a chosen model. The reference model must have the same relative degree as the plant, but is otherwise arbitrary. The reference model chosen for the reconfiguration application is $H(s) = \frac{a}{s+a}I_3$, where I_3 is the identity matrix of dimension 3. We let $a = 2.5$ (*cf.*

[14]). The input vector is $u^T = (\delta_H \; \delta_A \; \delta_R)$ and the output vector $y^T = (q \; p \; r)$.

If we assume that the reference model is matched, an autopilot can also be designed around that model. A choice for an autopilot is one that tracks the angles θ, ϕ, and ψ. Assuming that the reference model is matched, we have

$$\begin{aligned} \dot{q} &= -2.5q + 2.5q_c \\ \dot{p} &= -2.5p + 2.5p_c \\ \dot{r} &= -2.5r + 2.5r_c \end{aligned} \tag{31}$$

where q_c, p_c, and r_c are the elements of the reference input in (2). We can use the relationships

$$\begin{aligned} \dot{\theta} &= q \\ \dot{\phi} &= p \\ \dot{\psi} &= r \end{aligned} \tag{32}$$

with the commands

$$\begin{aligned} q_c &= g(\theta_c - \theta) \\ p_c &= g(\phi_c - \phi) \\ r_c &= g(\psi_c - \psi) \end{aligned} \tag{33}$$

so that the angles track the desired angles with the transfer function

$$\frac{2.5g}{s^2 + 2.5s + 2.5g} \tag{34}$$

With the constant g set to 1.6, the closed-loop poles are located at -1.25 $\pm j$ 1.56. Note that an outer loop could also be designed around the pitch angle command to regulate altitude in a similar manner.

3.3 Batch LS Results

The first implementation of a model reference control law used a batch least-squares identification, with data collected from the simulation. Both the indirect and direct input error methods were used, and control performance was found to be similar. The results shown here and afterwards are for the direct input error algorithm. The identifications were performed at the flight condition of Mach 0.5 at altitude 9,800 feet. Identifications were performed for the original aircraft and for the aircraft with a locked left horizontal tail surface. Ten seconds of data were used to perform the identification.

The identified matrices C_0, G_0, v were used to control the airplane. In turn, each of the reference inputs was given a series of step changes, while the other inputs were held at zero. The results for the direct input error identification are shown in Figure 2. The expected output y_M is represented by the solid lines, and the actual output y is shown as dashed lines. The actual output closely matches the expected output.

Figure 3 shows responses from tests with the aircraft after failure, using the matrices identified for the unfailed aircraft. The pitch rate response to a pitch rate command is seen to be significantly less than the desired response (less than 30%), due to the loss of pitching moment after the elevator failure. The roll rate response to a roll rate command is also smaller than specified, because roll control is partly achieved through the elevators in this aircraft. Finally, there is a very large cross-coupling from the pitch rate command to roll rate (1.5 deg/s roll rate for a 1 deg/s pitch rate command). This is due to the loss of symmetry in the elevator response and, as a consequence, to the production of a large rolling moment. Figure 4 shows the responses with the matrices identified from the failed system. The responses show that tracking of the commands was restored to the desired values. The main cross-coupling that appeared after the failure, as a roll response to a pitch rate command, was reduced by a factor of 2.

Fig. 5 shows the deflections of the right horizontal tail surface. On the left are the responses for the unfailed aircraft, and on the right for the failed aircraft, using the reconfigured control law. As expected, the deflections are about twice as large after the failure. However, they are well within limits. It is sometimes believed that model reference control laws require large control activity. However, this belief is rooted in early model reference control laws based on high-gain feedback and

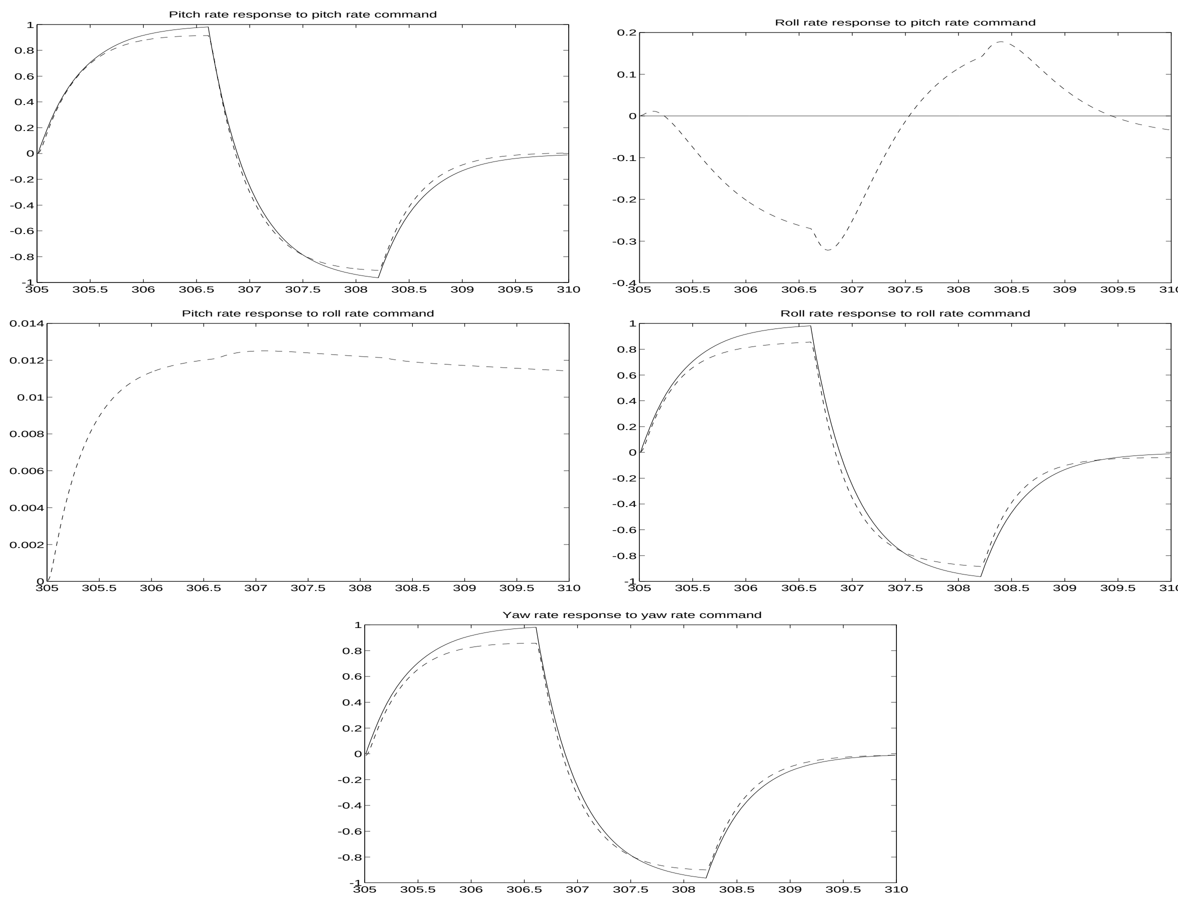

Figure 2: Batch LS: Responses for aircraft without failure

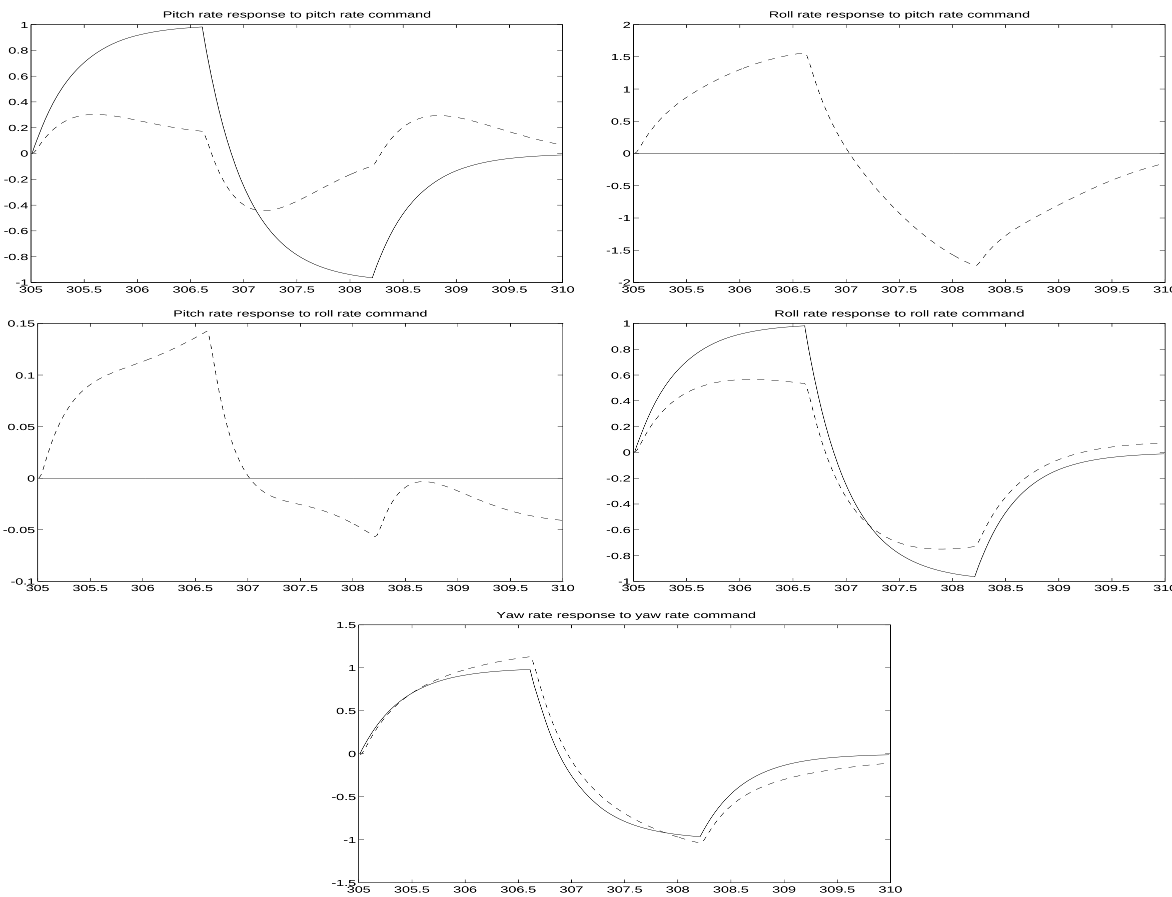

Figure 3: Batch LS: Responses for aircraft with failure, using parameters of unfailed aircraft

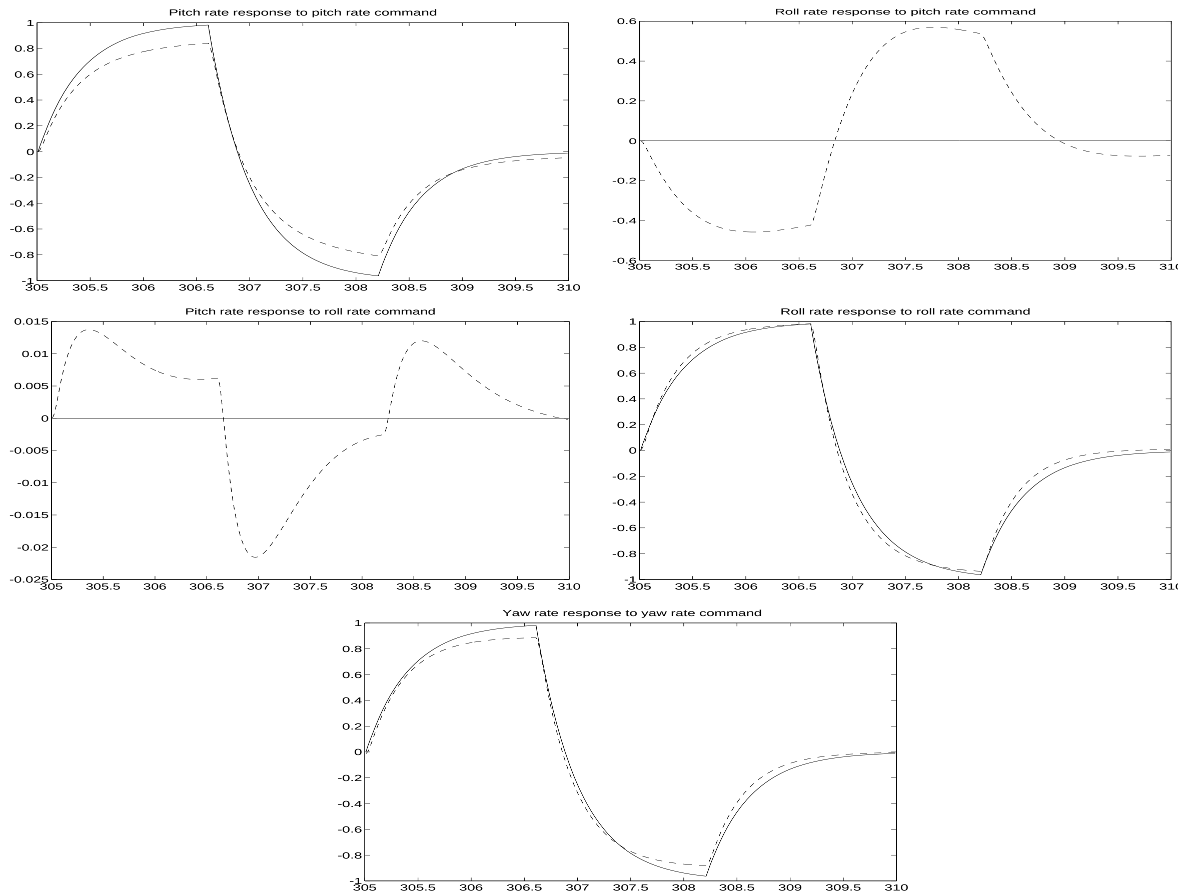

Figure 4: Batch LS: Responses for aircraft with failure

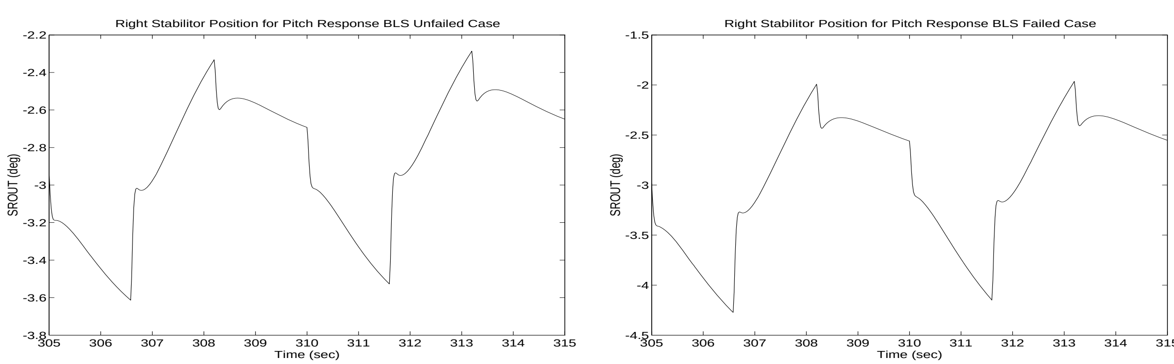

Figure 5: Batch LS: Right horizontal tail deflection, without and with failure

prefiltering of the control inputs. The control laws discussed in this paper do not rely on high-gain feedback, and do not require large control actions as long as reasonable choices of the reference model are made.

An interesting observation concerns the trim of the aircraft. The aircraft was trimmed at an angle of attack of about 4.6 degrees, requiring a command δ_H of approximately -2.86 degrees. The identification procedure did not identify the trim value for δ_H explicitly, or have α_{trim} available. However, the trim value for δ_H can be calculated for the previous angle of attack to be $v(1) + G_0(1,1)\,\alpha_{trim}$ which is equal to $0.02 - 0.64 * 4.6 = -2.92$ and is remarkably close to the actual trim value. The trim value is also observed on Fig. 5 as the average value of the actuator deviations. The least-squares procedure is successful in determining the trim value required to maintain level flight, without actually being told what that flight condition is.

3.4 Recursive LS with Forgetting Factor Results

The next implementation was a recursive least-squares with forgetting factor (RLSFF) algorithm. Performance was tested in the same manner as it was tested for the off-line batch LS identification. However, the control law was applied simulataneously in this case. Figures 6 and 7 compare the results obtained with the batch LS algorithm (dashed lines) to those obtained from the RLSFF algorithm (dash-dot lines). Figure 6 is for the aircraft without a failure and Figure 7 for the aircraft with a failure. As can be seen in the figures, the system output for the recursive algorithm follows the reference model output very well after a short transient. A notable difference with the batch algorithm is that cross-couplings are significantly reduced, from 1.5 deg/s in the original control law and 0.5 deg/s in the batch-redesigned control law to 0.1 deg/s in the recursive law. Over periods of time longer than those shown on the plots, the couplings were found to be further reduced by the recursive law, and to reach negligible values.

3.5 Stabilized RLSFF Results

The recursive least-squares with forgetting factor algorithm becomes unstable when there is insufficient excitation. Figure 8 shows the (1,1) elements of the parameter matrix C_0 and the covariance matrix P. During a period of insufficient excitation, such as during steady level flight, the matrices become unbounded. This instability is the motivation for the stabilized RLSFF algorithm.

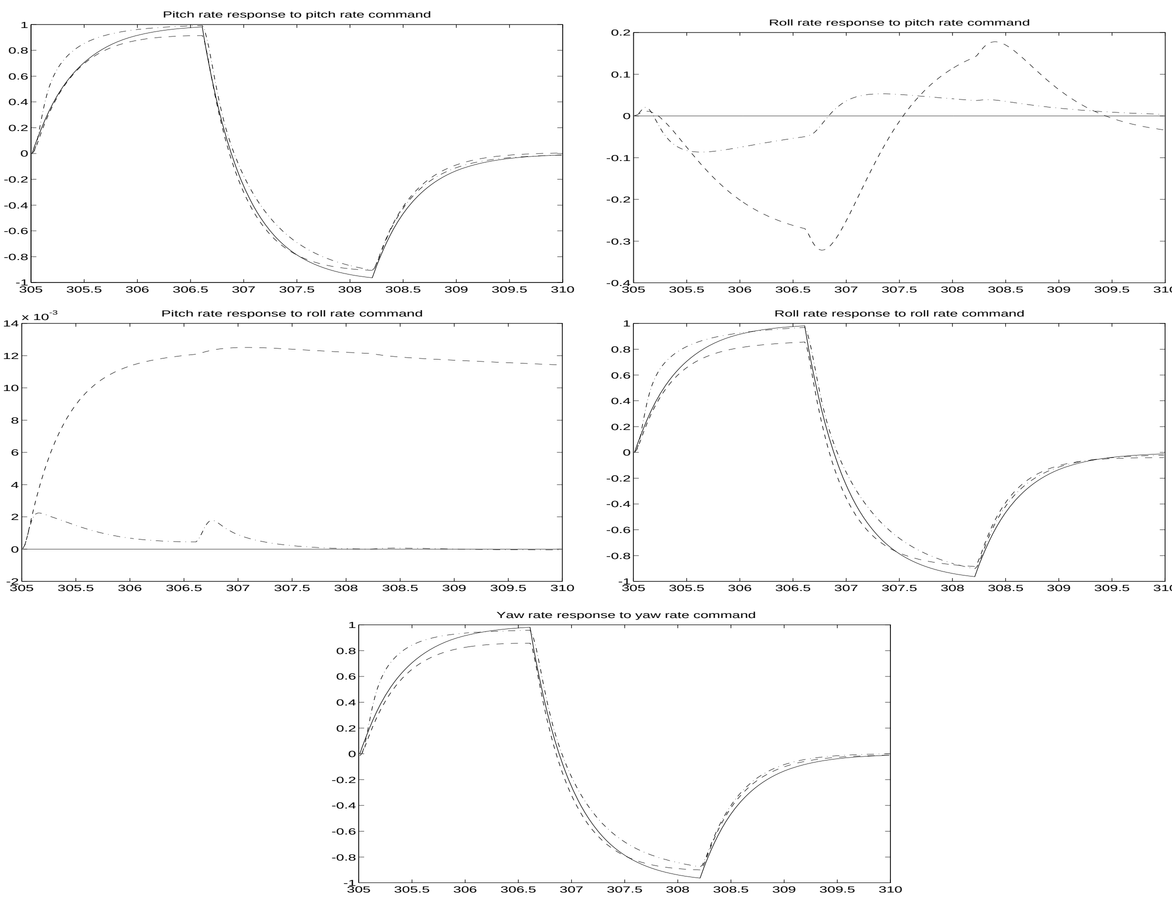

Figure 6: Comparison of batch and recursive LS results, aircraft without failure

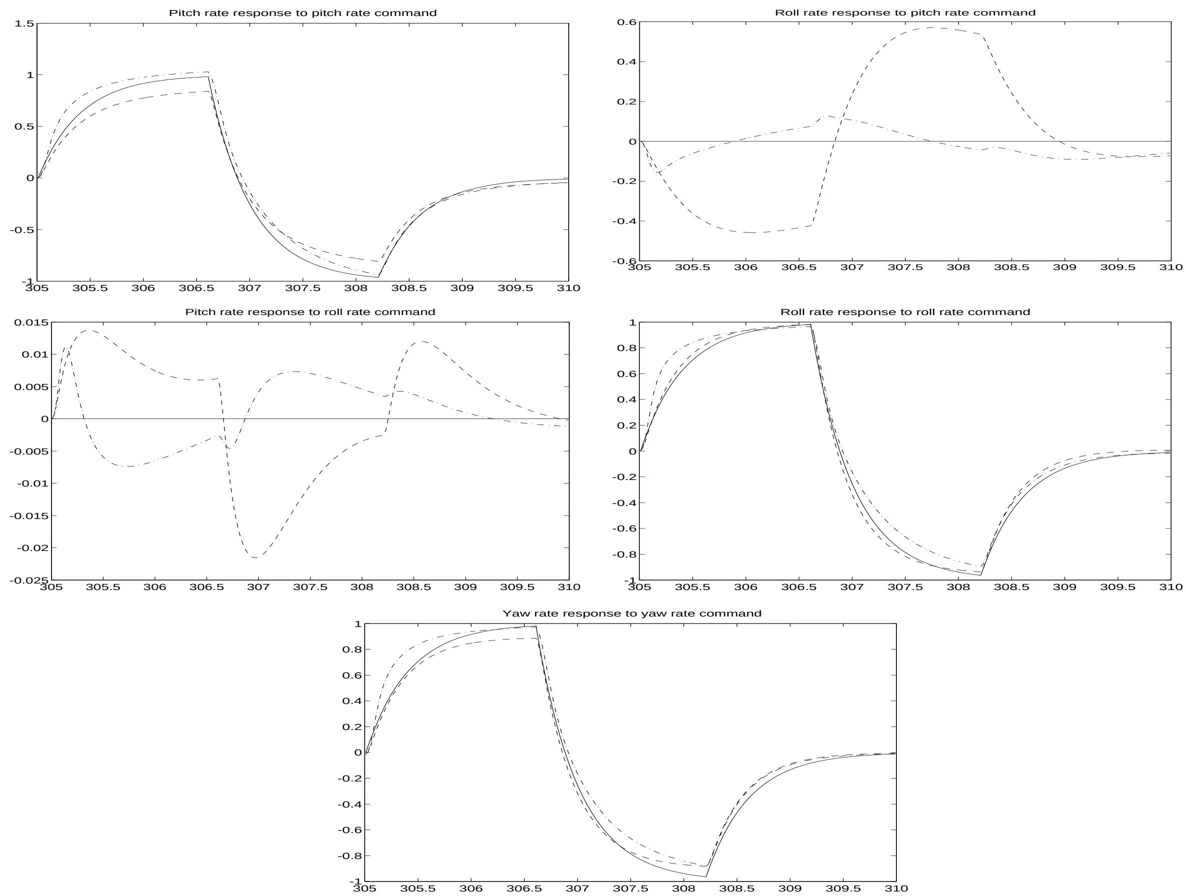

Figure 7: Comparison of batch and recursive LS results, aircraft with failure

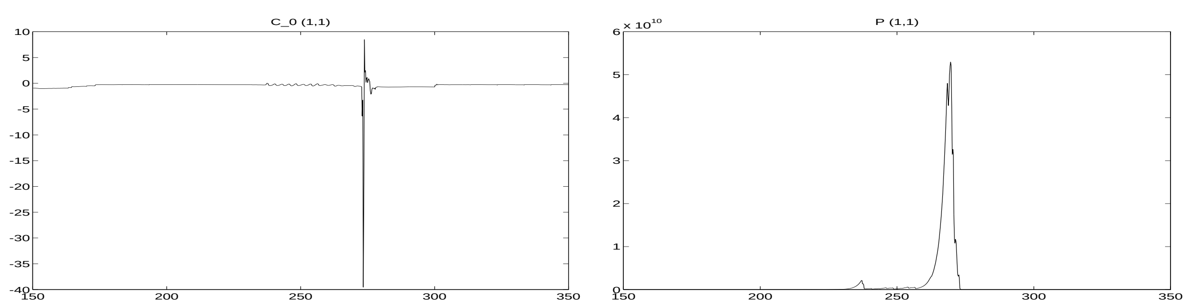

Figure 8: RLSFF: Low excitation results in instability

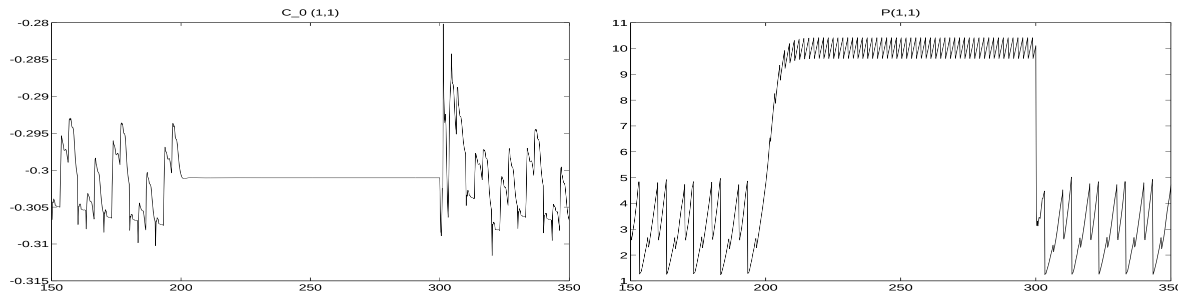

Figure 9: Stabilized RLSFF: Low excitation does not cause instability

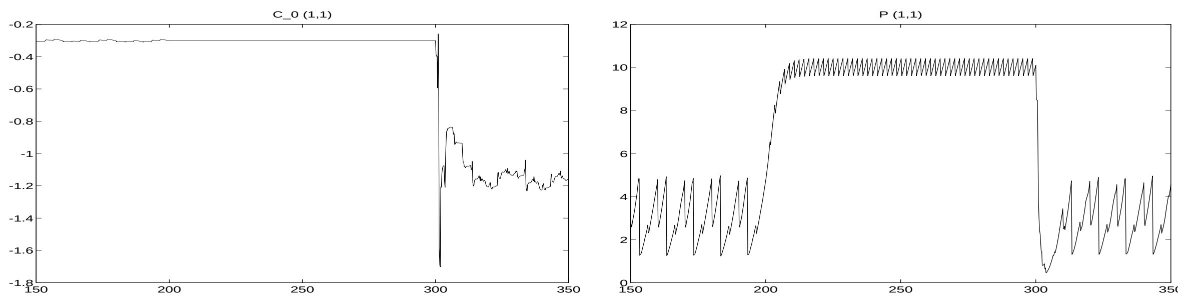

Figure 10: Stabilized RLSFF, aircraft with failure, at t = 250 seconds

In its unmodified form, the stabilized recursive least-squares with forgetting factor algorithm requires the inverse of a pxp matrix, where p is the number of parameters that are identified per row. In our case, $p = 9$. The modification proposed in equations (29) and (30) allows implementation of the algorithm using the inverse of a 2x2 matrix. This reduced algorithm was implemented on the airplane simulation. The test with the quiet period was performed again with the stabilized RLSFF. Figure 9 shows that the covariance and parameter matrices remain stable. Figure 10 shows the results of a similar test, but with a failure occurring in the middle of the quiet period. While in the quiet period, the algorithm does not adapt to the failure, because there is no excitation which can be used to identify the changes in the parameters. When there is excitation, the algorithm correctly identifies the unknown parameter.

One issue that presented itself was that numerical errors caused the covariance matrix update to become unstable. Enforcing the symmetry of P resolved this problem. Another solution would consist in using a square-root algorithm. This was not found to be necessary, however.

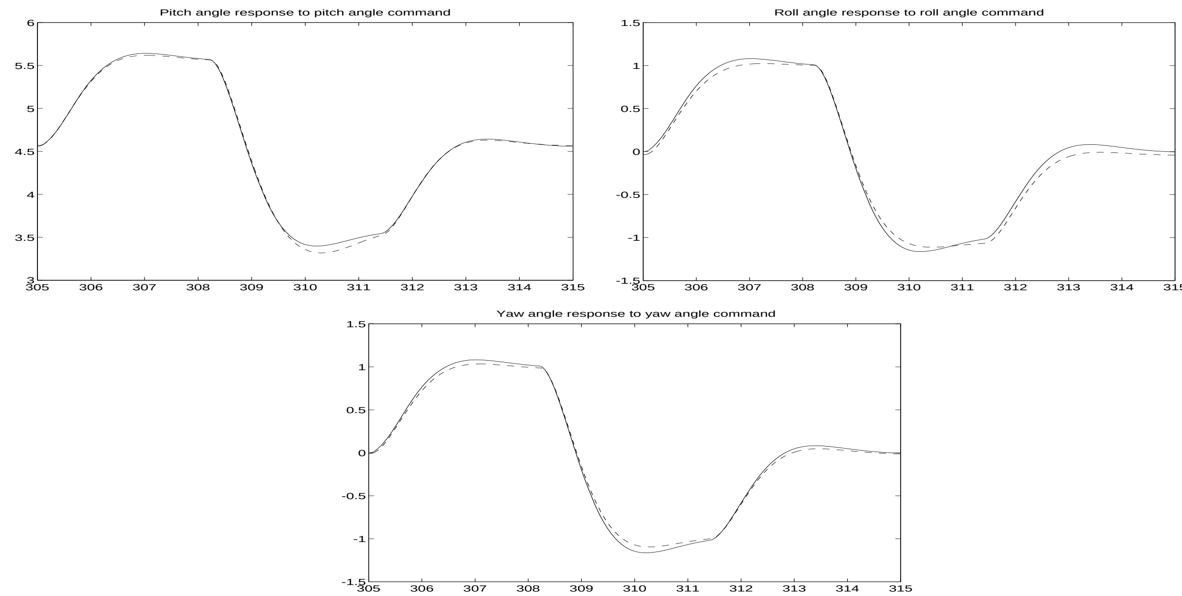

Figure 11: Tracking of θ, ϕ, and ψ by the auto pilot

3.6 Autopilot Angle Tracking

An autopilot was described in section 3.2 that could be used to control the angles θ, ϕ, and ψ of the airplane. This autopilot was implemented with the batch LS identification matrices, for demonstration of the concept. If the reference model is exactly matched, the transfer function $\frac{4}{s^2+2.5s+4}$ describes the relationship from each angle command to the output angle.

The autopilot was tested by holding two of the angle commands constant, while making step changes in the third command. Figure 11 shows some of the results from the tests. Note that there is a constant command for θ of 4.6 degrees. This is the angle of attack required to maintain trim flight. These responses show that once an inner loop is closed for the tracking of roll, pitch, and yaw rate commands, outer loops can easily be added for the control of other (slower) variables.

4 Conclusions

A reconfigurable flight control system is expected to perform three tasks. First, the system must adjust the trim values for the inputs, as a failure may cause rapid changes in the control inputs needed to maintain level flight. Second, the system is expected to decouple the inputs and outputs.

In the aircraft without a failure, this is easily achieved, because of symmetry. Once a failure occurs, however, the aircraft usually loses its symmetry, introducing strong cross-couplings. Finally, the closed-loop system must ensure tracking of the pilot commands, despite a reduction in control effectiveness. We have demonstrated that, with model reference adaptive control, it is feasible to achieve all three of these goals with satisfactory results.

The input error direct algorithm has important advantages over other algorithms. The indirect algorithm requires more parameters to be identified, 45 versus 27. Since there are a large number of calculations in the identification algorithm, a large number of parameters is a handicap. The output error direct algorithm estimates the same number of parameters as the input error direct algorithm, but has more rigid stability conditions than either the indirect or input error direct algorithms. Also, least-squares adaptation algorithms can be used with the indirect and input error direct algorithms, but not with the output error direct algorithm. Least-squares algorithms give faster convergence than gradient algorithms, a property which is critical for reconfiguration.

We investigated the use of a new algorithm, the stabilized recursive least-squares with forgetting factor algorithm (SRLSFF). The SRLSFF achieves stability during periods of low excitation by penalizing changes of the parameter matrix θ in the error function. This change results in an algorithm with relatively weak conditions for stability. At the same time, no sharp discontinuities are introduced in the responses, and implementation is computationally feasible.

Several issues would deserve to be examined. The performance of the algorithms should be tested when there is noise present in the system, as well as for other failures and at other flight conditions throughout the operational envelope of the aircraft. At other flight conditions, other output variables may be preferable. One such criterion might be control of the aircraft based on the acceleration experienced by the pilot. No consideration was given here to input saturation, which may be more restrictive at other flight conditions or for other maneuvers. The maneuvers considered in this paper were small, and did not induce actuator saturation or rate saturation. One possibility would be to incorporate the method proposed in [9], in order to handle the problem of input saturation through an outer-loop design. Considering more complicated algorithms, one should remember that computations must usually be performed at a rate of approximately 50-100 Hz. This requirement severely limits the complexity of the algorithms that can be realistically implemented with current flight control computers. The algorithms studied in this work are probably the simplest adaptive algorithms available that account for possible cross-couplings in a multivariable design.

References

[1] R.L. Barron, R.L. Cellucci, R.L., Jordan, N.E. Beam, P. Hess, & A.R. Barron, "Applications of Polynomial Neural Networks to FDIE and Reconfigurable Flight Control," *Proc. of the National Aerospace and Electronics Conference*, Dayton, OH, pp. 507-519, 1990.

[2] M. Bodson, "Identification with Modeling Uncertainty and Reconfigurable Control", *Proc. of the IEEE Conference on Decision and Control,* San Antonio, TX, pp. 2242-2247, 1993.

[3] M. Bodson, "An Adaptive Algorithm with Information-Dependent Data Forgetting", *Proc. of the Automatic Control Conference*, Seattle, WA, pp. 3485-3489, 1995.

[4] R.W. Brumbaugh, "An Aircraft Model for the AIAA Controls Design Challenge", *Proc. of the AIAA Guidance, Navigation, and Control Conference,* New Orleans, LA, 1991.

[5] P. Chandler, "Self-Repairing Flight Control System Reliability & Maintainability Program Executive Overview," *Proc. of the National Aerospace and Electronics Conference,"* Dayton, OH, pp. 586-590, 1984.

[6] P.R. Chandler, "Issues in Flight Control Design for Robustness to Failures and Damage," *Proc. of the IEEE International Conference on Control and Applications*, Jerusalem, Israel, paper WP-2-7, 1989.

[7] P.R. Chandler, M. Pachter, & M. Mears, "On-Line Optimizing Networks for Reconfigurable Control," *Proc. of the IEEE Conference on Decision and Control*, San Antonio, TX, pp. 2272-2277, 1993.

[8] P.R. Chandler, M. Pachter, & M. Mears, "System Identification for Adaptive and Reconfigurable Control," *J. Guidance, Control, & Dynamics,* vol. 18, no. 3, pp. 516-524, 1995.

[9] M. Pachter, P.R. Chandler, & M.J. Mears, "Reconfigurable Tracking Control with Saturation," *J. Guidance, Control, & Dynamics,* vol. 18, no. 5, pp. 1016-1022, 1995.

[10] C.J. Dittmar, "A Hyperstable Model-Following Flight Control System Used for Reconfiguration Following Aircraft Impairment," *Proc. of the Automatic Control Conference*, Atlanta, GA, pp. 2219-2224, 1988.

[11] M. de Mathelin, & M. Bodson, "Multivariable Model Reference Adaptive Control without Constraints on the High-Frequency Gain Matrix," *Automatica*, vol. 31, no. 4, pp. 597-604, 1995.

[12] R.A. Eslinger, & P.R. Chandler, "Self-Repairing Flight Control System Program Overview", *1988 IEEE National Aerospace and Electronics Conference (NAECON 1988)*, Dayton, OH, May 1988.

[13] H.N. Gross & B.S. Migyanko, "Application to Supercontroller to Fighter Aircraft Reconfiguration," *Proc. of the Automatic Control Conference*, Atlanta, GA, pp. 2232-2237, 1988.

[14] Honeywell Technology Center, *Multivariable Control Design Guidelines*, Draft of the report for the program "Design Guidelines for Application of Multivariable Control Theory to Aircraft Control Laws," Minneapolis, MN, 1995.

[15] C.Y. Huang & R.F. Stengel, "Restructurable Control Using Proportional-Integral Implicit Model Following," *Journal of Guidance, Control and Dynamics*, vol. 13, no. 2, pp. 303-309, 1990.

[16] Y. Landau, *Adaptive Control: The Model Reference Approach*, Marcel Dekker, 1979.

[17] D.P. Looze, J.L. Weiss, J.S. Eterno, & N.M. Barrett, "An Automatic Redesign Approach for Restructurable Control Systems," *IEEE Control Systems Magazine*, pp. 16-22, May 1985.

[18] P.S. Maybeck & R.D. Stevens, "Reconfigurable Flight Control via Multiple Model Adaptive Control Methods," *Proc. of the IEEE Conference on Decision and Control*, Honolulu, Hawaii, pp. 3351-3356, 1991.

[19] T.E. Menke, & P.S. Maybeck, "Sensor/Actuator Failure Detection in the VISTA F-16 by Multiple Model Adaptive Estimation", *Proc. of the American Control Conference*, San Francisco, CA, June 1993.

[20] D.D. Moerder, N. Halyo, J.R. Broussard, & A.K. Caglayan, "Application of Precomputed Control Laws in a Reconfigurable Aircraft Flight Control System," *Journal of Guidance, Control and Dynamics*, vol. 12, no. 3, pp. 325-333, 1989.

[21] W.D. Morse & K.A. Ossman, "Model Following Reconfigurable Flight Control System for the AFTI/F-16," *Journal of Guidance, Control and Dynamics*, vol. 13, no. 6, pp. 969-976, 1990.

[22] K.S. Narendra & A. Annaswamy, *Stable Adaptive Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[23] A.J. Ostroff, "Techniques for Accommodating Control Effector Failures on a Mildly Statically Unstable Airplane," *Proc. of the Automatic Control Conference*, Boston, MA, pp. 906-913, 1985.

[24] S. Sastry, & M. Bodson, *Adaptive Control: Stability, Convergence and Robustness*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[25] D.L. Schneider, I.M. Horowitz & C.H. Houpis, "QFT Digital Flight Control Design as Applied to the AFTI/F-16," *Proc. of the National Aerospace and Electronics Conference*, Dayton, OH, pp. 616-623, 1987.

[26] K.M. Sobel & H. Kaufman, "Direct Model Reference Adaptive Control for a Class of MIMO Systems," *Control and Dynamic Systems*, Academic Press, 1986.

[27] D. Sofge & D. White, "NSF Workshop on Aerospace Applications of Neurocontrol," *IEEE Control Systems Magazine*, pp. 80-81, April 1991.

[28] J.M. Urnes, R.B. Yeager, & J. Stewart, "Flight Demonstration of the Self-Repairing Flight Control System in a NASA F-15 Aircraft", presented at the *National Aerospace and Electronics Conference*, Dayton, OH, 1990.

[29] D.G. Ward & R.L. Barron, "A Self-Designing Receding Horizon Optimal Flight Controller," *Proc. of the American Control Conference,* Seattle, WA, pp. 3490-3494, 1995.

[30] J.L. Weiss & J.Y. Hsu, "Integrated Restructurable Flight Control System Demonstration Results," NASA Contractor Report 178305, Langley Research Center, Hampton, VA, May 1987.

[31] J.L. Weiss, J. Eterno, D. Grunberg & D. Looze, "Investigation of an Automatic Trim Algorithm for Restructurable Aircraft Control," *Proc. of the National Aerospace and Electronics Conference*, Dayton, OH, pp. 400-406, 1986.