Evaluation of a Multiple-Model Failure Detection System for the F-16 in a Full-Scale Nonlinear Simulation

P. Eide, P. Maybeck
Department of Electrical and Computer Engineering
Air Force Institute of Technology (AFIT/ENG)
Wright-Patterson AFB, OH 45433

Abstract

A Multiple Model Adaptive Estimation (MMAE) algorithm is implemented with the fully nonlinear six-degree-of-motion, Simulation Rapid-Prototyping Facility (SRF) VISTA F-16 software simulation tool. The algorithm is composed of a bank of Kalman filters modeled to match particular hypotheses of the real world. Each presumes a single failure in one of the flight-critical actuators, or sensors, and one presumes no failure. The algorithm is demonstrated to be capable of identifying flight-critical aircraft actuator and sensor failures at a low dynamic pressure (20,000 ft., 4 Mach). Research considers single hardover failures. Tuning methods for accommodating model mismatch, including addition of discrete dynamics pseudonoise and measurement pseudonoise, are discussed and demonstrated. Robustness to sensor failures provided by MMAE-based control is also demonstrated.

1 Introduction

Current state-of-the-art flight control systems (FCS) rely on costly physical redundancies to provide required aircraft reliability and survivability. What is desired is a FCS which can reduce, or even eliminate, these redundancies by exploiting the functional redundancies inherent in control surfaces and sensors. Such a FCS must be able to operate safely when its control surfaces and flight-critical sensors fail. Our goal, then, is a fault tolerant, reconfigurable FCS. The first step in that direction is a reliable failure detection algorithm.

The Air Force Institute of Technology (AFIT/ENG) has been working with Wright Laboratory (WL/FIGS) to develop multiple model, Kalman-filter-based, algorithms for flight control applications [4, 6, 10, 13-17]. Past research utilized linearized “truth models” for performance simulations with good results. This effort has sought to increase confidence in these results and further demonstrate the multiple model adaptive estimator (MMAE) approach to failure detection by applying the algorithm to a more realistic, high-order, nonlinear, aircraft truth model: the most complete simulation model currently available. In addition, with an eye toward fault-tolerant control, we look ahead to Multiple Model Adaptive Control (MMAC) by investigating fault tolerance to sensor failures already present in the MMAE.

The following sections include a statement of the problems which our research addresses, a short overview of the MMAE algorithm, a discussion of the models used, our results, and some concluding remarks.

2 Problem Statement

The task at hand is to implement an existing MMAE failure detection algorithm into as real a simulation environment of the F-16 as possible in order to facilitate a more realistic assessment of the algorithm's true potential [5, 6, 12]. Specifically, we will characterize the algorithm's failure detection performance, at a single point within the flight envelope, for single hard-over failures of twelve flight critical actuators and sensors. Failures include the left and right stabilator (LS, RS), left and right flaperon (LF, RF), and rudder (RUD) actuators, and forward velocity (VEL), angle of attack (AOA), pitch rate (PIT), normal acceleration (Az), roll rate (ROL), yaw rate (YAW), and lateral acceleration (Ay) sensors. If necessary, the algorithm will be tuned to achieve acceptable failure detection performance. MMAE-based control's robustness to sensor failures will also be investigated.

3 MMAE/MMAC Overview

Figure 1 shows a functional block diagram of the MMAE algorithm. Its primary feature is a bank of steady-state,
discrete, Kalman filters operating in parallel, with a vector of sensed measurements \( z_t \) and a vector of actuator control commands \( u \) as the algorithm's inputs. All filter designs are based on the same reduced-order, linearized equations of motion for our nominal point in the flight envelope, but each hypothesizes a different failure condition. At every sample period, each of these \( K \) filters produce a state estimate \( \tilde{x}_k \), and a vector of residuals \( r_k \), for \( k = 1, 2, \ldots, K \). The idea is that the filter which produces the most well-behaved residuals, contains the model which best matches the true failure status of the aircraft [4, 8, 9].

\[ p_k(t) = \frac{f_z(t_0\mid t, z_{t-1})(z_t\mid a_k, Z_{t-1})p_k(t_{t-1})}{\sum_{j=1}^{K} f_z(t_0\mid t, z_{t-1})(z_t\mid a_j, Z_{t-1})p_j(t_{t-1})} \]

(1)

where

\[ f_z(t_0\mid t, z_{t-1})(z_t\mid a_k, Z_{t-1}) = \frac{1}{(2\pi)^{m/2}|A_k(t_0)|^{1/2}}e^{-1/2} \]

(2)

and

\[ [-] = -\frac{1}{2}a_k(t_0)^T A_k^{-1} a_k(t_0) + r_k(t) \]

(3)

In these equations, \( f_z(t_0\mid t, z_{t-1})(z_t\mid a_k, Z_{t-1}) \) is the probability density function of the current measurement \( z(t_0) \), conditioned on the hypothesized failure status \( a = a_k \) and previously observed measurement history \( Z(t_{t-1}) \), based on a filter's residuals \( r_k \) and internally precomputed residual covariance \( A_k \). When actual residuals are in consonance with filter-computed covariance \( A_k \), the exponential term of Equation (2) (Equation (3)) is approximately \( -\frac{m}{2} \), where \( m \) is the measurement dimension.

With an incorrect hypothesis, the magnitude of that exponential is much greater, yielding a deweighted \( p_k \) for that hypothesis from Equations (1) through (3). In practice the \( p_k \)'s are artificially lower bounded (\( p_{k\text{min}} = 0.001 \)) to prevent "lock-out" and the leading term \( \frac{1}{(2\pi)^{m/2}|A_k(t_0)|^{1/2}} \) is stripped to prevent a bias towards declaring sensor failures [1, 5, 6, 11-16]. The algorithm is started up with the no-failure hypothesis presumed. The output of this block is a vector of probabilities which can be used to declare a failure and also weight the state estimates as also shown in Figure 1. The output of the algorithm is a probability weighted state estimate. In this application, the state estimate and linear combinations thereof are then passed directly as inputs to the existing Block 40 FCS for the VISTA F-16 (rather than using the raw measurements as in the existing FCS). This results in the "MMAE-based" control shown in Figure 1.

Not shown in Figure 1, but nonetheless important to effective operation of the MMAE algorithm is dithering. Dithering is the term given to purposeful commands sent to the actuators which "shake up" the system state. For the MMAE to detect failures, it must have some activity in the state vector from which to discern aberrant behavior. Past research has investigated various forms of dithering which are subliminal to the pilot [12]. This application uses those results directly.

The architecture of Figure 1 can be expanded to include elemental controllers in series with the Kalman filters. Each controller can be designed to operate "optimally" in its hypothesized failure configuration, resulting in fault-tolerant control. Such an architecture is called Multiple Model Adaptive Control (MMAC) and will be the subject of future research. In this effort, we recognize that different elemental controllers aren't needed for hypothesized sensor failures. The reconstructed measurement vector, \( \hat{z} \), contains the failed sensor's information which is missing in the raw measurement vector, \( z \). As a first step towards MMAC, we will therefore demonstrate this capability.

4 Models

The "truth model" used in this research is the VISTA F-16 aircraft simulation hosted in the Simulation Rapid Prototyping Facility (SRF) at the Flight Controls Division of the Flight Dynamics Directorate, Wright Laboratory [2, 3]. This is a high-order, fully nonlinear, six-degree-of-freedom simulation which includes the Block-40 FCS. During non-realtime simulation runs, the truth model provides required inputs \( u \) and \( z \) to the MMAE and the MMAE returns its estimated \( \hat{z} \) to the FCS.

The design models used in the Kalman filters are based on small-perturbation-assumed, linearized, state-space equations of motion which are provided by a subroutine within the SRF VISTA F-16 software. Simple first-order lags (i.e., reduced-order models) are augmented to the state-space formulation to account for actuator dynamics. Wind buffeting is accounted for by incorporating a zero-order Dryden wind model, and sensor noise is included as well [7]. A full development of all models can be found in [4].

![Figure 1: Multiple Model Adaptive Estimator Block Diagram](image-url)
5 Failure Identification Results

Untuned MMAE. The MMAE design described in Sections 3 and 4 is implemented into the SRF VISTA F-16 simulation. All twelve single failures, as well as the no-failure case, are each run a total of 10 times to give results with statistical confidence. Figure 2 displays the results. This figure contains 13 strip plots, each of which shows the mean (over 10 runs) time history of the probability that the failure hypothesis on which it is based is correct when that failure is actually introduced at 3.0 seconds into the simulation. It is readily apparent from the plots that the design is unacceptable at this point. Clearly, there are higher order effects for which our filter design models cannot account. While there is good stimulation within the appropriate channels when a failure is introduced, there are just too many false alarms and missed alarms.

Figure 2: Single Failure Probability Convergence - Untuned

Tuned MMAE. In order to account for unknown higher order effects and improve on the performance of Figure 2, we go about tuning the filters through addition of pseudonoise. In this research, we explore four methods of tuning which can exploit any available insight into the problem, three of which will be incorporated into the final design.

Method 1: Direct Pseudonoise on Longitudinal Diagonal $Q_d$ Entries

In the first method, we look to compensate for our reduced-order dynamics model equations by adding pseudonoise to the dynamics model. From Figure 2 we note significant false alarming in the normal acceleration ($A_N$) channel and some in the angle of attack (AOA) channel as well. This insight directs us to do tuning in the longitudinal channel first. To accomplish this, we insert a fictitious source of discrete white noise ($w_d$) of diagonal covariance $Q_d$ into the filters' dynamics model equation:

$$x_k(t_{k+1}) = \Phi x_k(t_k) + B_d u(t_k) + G_d w_d(t_k) + w_d(t_k)$$

and specifically adjust the longitudinal diagonal entries ($Q_{a_0}, Q_{a_a}, Q_{a_d}, Q_{a_i}$). The advantage here is that one can tune directly on individual states (pitch angle, velocity, angle of attack, pitch rate) and, with insight, correct for dynamics model deficiencies. Figure 3 shows the probability convergence results when using this tuning method.

The improvement is quite good. With a few exceptions, the forward velocity sensor (VEL) and left stabilator actuator (L ST) most notably, unambiguous single failure detection has been achieved. False alarms and missed alarms are nearly eliminated. It should be noted however that with increases in pseudonoises can come decreases in speed of performance. Performance improvements gained in some channels through addition of pseudonoise can cause sluggish performance in others, and in the extreme,
lead to missed alarms. This tradeoff must always be considered in the context of any application.

**Method 2: Direct Pseudonoise on R Entries**

A second method which we use to improve performance in the forward velocity (VEL) channel adds white noise directly to the entries of $R_k$, which is the covariance of the measurement corruption noise $v_k(t_k)$ in the assumed measurement model:

$$z_k(t_k) = H_k x_k(t_k) + v_k(t_k)$$

This enters into the Kalman filter equations via the filter-computed residual covariance:

$$A_k(t_k) = H_k(t_k) P_k(t_k^{-1}) H_k^T(t_k) + R_k(t_k)$$

which impacts Equations (1) through (3) directly. The advantage here is that one can tune directly in a sensor channel which is experiencing performance difficulty by varying the values of the diagonal entries in $R$ ($R_u, R_a, R_y, R_{AX}, R_y, R_{AY}$).

**Method 3: Direct Pseudonoise on Lateral Diagonal $Q_d$ Entries**

For the third method of tuning, we return to the concept of Method 1, adding pseudonoise directly to states through $Q_d$. This time however, we seek to eliminate the ambiguity still present in the left stabilator (L ST) channel. The L ST plot in Figure 4 shows this probability “bouncing” at the onset of failure. From data not explicitly shown in Figure 4, the ambiguity is determined to be coming from the yaw channel. We therefore seek to demonstrate the technique of Method 1 further and improve performance by tuning the lateral diagonal $Q_d$ entry $Q_{dL}$.

![Figure 4: Single Failure Probability Convergence - $Q_{dL}$ and R Tuning](image)

Figure 4 shows the performance improvement which can be achieved using this method. Comparing the VEL plot in Figures 3 and 4, one can see that we've gained unambiguous failure detection at the cost of response.

![Figure 5: Single Failure Probability Convergence - $Q_{dL}$, R, and Qdlat Tuning](image)

Figure 5 displays the result of this tuning method. A comparison of the L ST and YAW plots from Figures 4 and 5 shows that we can eliminate the ambiguity in the L ST channel at the cost of some performance degradation in the YAW channel.

Having performed the tuning in Methods 1, 2, and 3, we arrive at a design which provides unambiguous failure detection capability in all channels with no false alarms. No attempts are made to do any optimization with the tuning, since explicit performance specifications are not provided at the outset. Rather, Figure 5 demonstrates the potential of the MMAE failure detection algorithm.
**Method 4: Tuning for Actuator Uncertainty**

This last method attempts to account for model deficiencies related to actuator effects. It is not included in our final design, but presented for possible future use. Here, a fictitious source of scalar white noise \( \omega(t) \), of strength \( q \), is added to the model dynamics equation, from which Equation (4) is generated as an equivalent discrete-time model [8]:

\[
\dot{x}(t) = Ax(t) + Bu(t) + Gw(t) + B_{acte} \omega(t) \quad (7)
\]

where \( B_{acte} \) is the column of the original \( B \) corresponding to the actuator of interest. Here, the continuous pseudonoise is brought through the relevant actuator dynamics stripped from the basic unaugmented state-space representation; the corresponding covariance differential equation for the Kalman filter propagation cycle would then be:

\[
\dot{P}(t) = AP(t)A^T + GQG^T + B_{acte} \Sigma B_{acte}^T \quad (8)
\]

The advantage is that one can directly tune in order to address poor single actuator failure detection performance.

**6 MMAC Demonstration**

Having designed a multiple model adaptive estimator (MMAE) which can detect and isolate single failures, we wish to demonstrate the resulting MMAE-based controller’s robustness to sensor failures. As stated in section 3, this line of research endeavors ultimately to proceed to the fault tolerant control of a Multiple Model Adaptive Controller (MMAC). Before designing the elemental controllers however, we can already show the algorithm’s tolerance to sensor failures.

To demonstrate this feature, we use the MMAE design which includes tuning methods 1, 2, and 3. We simulate the system first with no failure inserted and record the trajectories of the state vector. This serves as a baseline of unfailed operation. Next, we disable the MMAE-based control of Figure 1 by removing the \( z \), and provide the raw vector of inputs \( z \) instead. A sensor failure is then inserted (at 3.0 seconds) and simulated with the state vector again being recorded. Figure 6 shows both the unfailed trajectories (dashed lines), and the trajectories corresponding to a failed angle of attack sensor (solid lines). This demonstrates the type of departure experienced when a critical sensor fails.

We then restate the MMAE-based control scheme of Figure 1 and run the failed angle of attack sensor simulation once again. Figure 7 displays the results. As before, the unfailed trajectories are shown with a dashed line and the solid line represents operation with the failed sensor. Figure 7 shows that the trajectories are virtually identical. Therefore, operation is demonstrated to be the same with or without the presence of a failed sensor. Similar results are obtained for other sensor failures as well. State reconstruction generated by the Kalman filter approach provides the additional benefit, and different elemental controllers aren’t necessary for sensor failures within a MMAC. Our MMAE-based control is already robust to sensor failures.

**7 Conclusions**

The SRF VISTA F-16 simulation provides an opportunity to apply the MMAE failure detection algorithm in as near a real-world environment as is currently possible and further determine its potential. From this application, we learn that model mismatch due to reduced order, linearized modeling leads to false alarms and must be accounted for via addition of pseudonoise to assumed design models. Upper limits on pseudonoise addition do exist, however, which if passed, result in missed alarms. Intelligent pseudonoise addition such as that demonstrated in Methods 1 through 4, though, can result in acceptable probability convergence. Further, the algorithm is demonstrated to possess robustness to sensor failures, which is part of the ultimate goal of fault tolerance and MMAC.

Overall, the MMAE performed well against the fully nonlinear simulation “truth model” environment. With no performance specifications driving our research, there was no pressing effort to optimize our results.
Implementation-specific requirements must always be considered, with the techniques just described being applied appropriately. The results of this effort though, argue strongly for continued research and development in this area.

References


