The NF-16D VISTA Simulation System

George M. Siouris

Abstract: Called VISTA (Variable-stability In-flight Simulator Test Aircraft), the one-of-a-kind NF-16D has a simulation system that can mimic several aircraft. Though housed in an F-16 Fighting Falcon airframe, VISTA can also act like the F-15 Eagle or the Navy's F-14 Tomcat. More importantly, such flexibility allows for improved training and consolidation of some sorties. Consequently, USAF Test Pilot School students will have an opportunity to learn how to test future integrated cockpits. In this paper, we will use the multiple model adaptive estimation (MMAE) and the multiple model adaptive controller (MMAC) techniques to model the aircraft's flight control system containing the longitudinal and lateral-directional axes. Single and dual actuator and sensor failures will also be included in the simulation. White Gaussian noise will be included to simulate the effects of atmospheric disturbances.

Keywords: VISTA, multiple model adaptive estimation, Kalman filtering, failure detection and isolation, flight control

I. Introduction

The USAF's VISTA NF-16D aircraft is one-of-a-kind aircraft that can mimic the flying qualities of the F-15 Eagle and the Navy's F-14 Tomcat. The VISTA aircraft provides the USAF Test Pilot School student an opportunity to learn how to test future integrated cockpits. Such flexibility allows for improved training and consolidation of some sorties. It also provides a better value for flying-hour costs. On a single sortie, students can evaluate the specific flying qualities of the F-14's direct lift control feature, or experience the unique pitch control characteristics of a delta wing aircraft like the F-106 *Delta Dart*, which is no longer in the Air Force inventory. Flying different types of aircraft is key since some students come from the Navy, Marine Corps and allied countries. This aircraft will save money and time, since students don't have to fly in each kind of aircraft to get the same quality learning experience. Moreover, the system lets students "peal back" the layers of flight control features in today's digital fly-by-wire systems to see the effects of each. In one demonstration, students start flying the basic F-16, then sequentially remove flight control features until they are flying the aircraft manually. This gives vivid and dramatic proof for the need and effects of computerized flight controls on inherently unstable aircraft like the F-16.

At the heart of the aircraft is a simulation system with a suite of high-speed digital computers. They sense and digitally record all necessary aircraft data and transmit them in real-time via telemetry downlink. That process was useful when the aircraft was used to help develop such leading-edge designs like the F-22 Raptor, the Joint Strike Fighter and NASA's X-38 shuttle lifeboat. The system has an automatic safety monitoring system. It is easily reconfigured from one aircraft to another during flight and has an all-attitude simulation capability. The aircraft also features an integrated programmable helmet-mounted display (HMD).

In this paper, a complete F-16 flight control system is investigated, with particular emphasis in modeling the longitudinal and lateral-directional axes. In particular, the multiple model adaptive estimation (MMAE) and multiple model adaptive controller (MMAC) algorithms will be used in the Variable-stability NF-16D In-flight Simulator Test Aircraft (VISTA). Single and dual actuator and sensor failures are simulated, which include the following: a) complete actuator failures, b) partial actuator failures, c) increased sensor noise, d) sensor biases, and e) combinations of actuator and sensor failures. These failures are examined in light of both maneuvering and straight and level flight conditions. White Gaussian noise is included in order to simulate the effects of atmospheric disturbances, and white Gaussian noise is added to the measurements in order to simulate the effects of sensor noise.

As stated above, the present investigation makes use of both the MMAE and MMAC algorithms. Although these algorithms were developed in the early seventies, the application of the algorithm was not practical at the time. The reason is that the MMAE algorithm uses multiple elemental controllers each of which can represent a specific failure condition. That is, application of the algorithm was not feasible until the development of parallel processing digital computers. However, the complexity and inherently nonlinear nature of the multiple adaptive controller prevents complete theoretical analysis of its closed-loop performance.

Previous research efforts have addressed a number of interesting modifications to the MMAE/MMAC algorithms for a variety of different aircraft, that the NF-16D VISTA aircraft is able to simulate [1], [5]. However, these research efforts consider only the longitudinal axis with no cross-axis coupling. In the present investigation, we will investigate the ability of MMAC-based controller to monitor the current state of the NF-16D aircraft in both longitudinal and lateral-directional axes, at a low dynamic pressure flight condition (i.e., 0.4 Mach at 20,000-ft.). It should be noted that a low dynamic pressure test case provides an environment in which control surfaces are less effective and dynamic damping is poor. Therefore, this case will demonstrate the algorithm's characteristics under unfavorable circumstances. Moreover, it should be pointed out that the most obvious limitation in this investigation is the use of a linearized VISTA NF-16D aerodynamic model (note that in the sequel we will call the VISTA NF-16D aircraft simply VISTA). In the present investigation, we will simulate the Dryden wind model, using the appropriate shaping filters driven by white Gaussian noise of the appropriate strength.

Manuscript received: July 26, 2001, Accepted: Nov. 16, 2001.

George M. Siouris: AF Institute of Technology, and Consultant-Avionics and Weapon Systems, 4425 E. Entrada Drive, Dayton, OH 45431, U.S.A.(gsiouris@worldnet.att.net)

II. Model methodology and development

1. Aircraft model

The aircraft model is represented as a linear continuous-time dynamics equation given by

$$dx/dt = Ax(t) + Bu(t) \tag{1}$$

with an output equation

$$y(t) = Cx(t) + Du(t) \tag{2}$$

and a discrete-time measurement equation given by

$$z(t_l) = Hx(t_l) + Du(t_l) + v(t_l) \tag{3}$$

where $u(t)$ is the adaptive control vector, and $y(t)$, $z(t_i)$, $v(t_i)$ are identical in dimension and physical interpretation of components for the truth and filter models. Generally, the truth models are of higher state dimensionality than the filter models due to higher order actuator models, discrete gust models, wind models, etc. Additionally, the truth model contains actuator rate and position limiting functions. The noise vector $v(t_l)$ is a zero-mean white Gaussian noise with covariance R. The covariance matrix represents measurement inaccuracies caused by sensor noise. Note that the high frequency sensor dynamics are omitted from the aircraft dynamics model since these are secondary effects. The aircraft velocity can be expressed as

$$du(t_l)/dt = X_\theta'\theta + X_u'u + X_\alpha'\alpha + X_q'q + \\ + X_{\delta S}'\delta S + X_{\delta F}'\delta F + X_{\delta LEF}'\delta LEF \tag{4}$$

with $X_l' = X_l / (mI_{xx})$, where m is the aircraft mass and I_{xx} is the moment of inertia about the aircraft x-axis. The stabilator and flaperon inputs are expressed as

$$0.5X_{\delta S}'(\delta S_L + \delta S_R) \quad \text{and} \quad .0.5X_{\delta F}'(\delta F_L + \delta F_R),$$

respectively. In state space format, the aircraft equation of motion is as follows:

$$\begin{matrix} \theta \\ u \\ \alpha \\ d/dt\ q \\ \phi \\ \beta \\ p \\ r \end{matrix} = \begin{matrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ X_\theta' & X_u & X_\alpha' & X_q' & 0 & 0 & 0 & 0 \\ X_\theta' & X_u & X_\alpha' & X_q' & 0 & 0 & 0 & 0 \\ M_\theta' & M_u' & M_\alpha' & M_q' & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & Y_\phi' & Y_\beta' & Y_p' & Y_r' \\ 0 & 0 & 0 & 0 & 0 & L_\beta' & L_p' & L_r' \\ 0 & 0 & 0 & 0 & 0 & N_\beta' & N_p' & N_r' \end{matrix} \begin{matrix} \theta \\ u \\ \alpha \\ q \\ \phi \\ \beta \\ p \\ r \end{matrix}$$

$$+ \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5X_{\delta S}' & 0.5X_{\delta S}' & 0.5X_{\delta F}' & 0.5X_{\delta F}' & 0 & X_{\delta LEF} \\ 0.5Z_{\delta S}' & 0.5Z_{\delta S}' & 0.5Z_{\delta F}' & 0.5Z_{\delta F}' & 0 & Z_{\delta LEF} \\ 0.5M_{\delta S}' & 0.5M_{\delta S}' & 0.5M_{\delta F}' & 0.5M_{\delta F}' & 0 & M_{\delta LEF} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5Y_{\delta S}' & -0.5Y_{\delta S}' & 0.5Y_{\delta F}' & -0.5Y_{\delta F}' & Y_{\delta R}' & 0 \\ 0.5L_{\delta S}' & -0.5L_{\delta S}' & 0.5L_{\delta F}' & -0.5L_{\delta F}' & L_{\delta R}' & 0 \\ 0.5N_{\delta S}' & -0.5N_{\delta S}' & 0.5N_{\delta F}' & -0.5N_{\delta F}' & N_{\delta R}' & 0 \end{matrix} \begin{matrix} \delta S_L \\ \delta S_R \\ \delta F_L \\ \delta F_R \\ \delta R \\ \delta LEF \end{matrix} \tag{5}$$

Conventionally, positive deflections are defined as stabilator trailing edge down, left flaperon trailing edge down, and rudder trailing edge right (i.e., those deflections that produce positive moments). Table 1 summarizes the various state variables, while table 2 summarizes the control variables.

Table 1. Aircraft state variables.

State Variables	Units
θ - pitch angle	[rad]
u - forward vel.	[ft/s]
α - angle-of-attack	[rad]
q - pitch rate	[rad/s]
ϕ - bank angle	[rad]
β - sideslip angle	[rad]
p - roll rate	[rad/s]
r - yaw rate	[rad/s]

Table 2. Control variables.

Control Variables	Units
δS_L - Left stabilator	[rad]
δS_R - Right stabilator	[rad]
δF_L - Left flaperon	[rad]
δF_R - Right flaperon	[rad]
δF - Rudder	[rad]
δLEF – Leading edge flap	[rad]

Note that the above two tables lists the eight state variables and six control variables, respectively. Furthermore, each of the state variables can be expressed in terms of dimensional derivatives.

The aircraft model was run on a simulation program residing at the Flight Dynamics Laboratory at Wright-Patterson AFB. Also, the dimensional derivatives of Eq.(5) were developed using this simulation program. This program contains the VISTA flight control system as well as a complete nonlinear aerodynamic database. The code allows the user the ability to linearize the model about a trim condition. The thrust remains constant for this application. A FORTRAN coded flight control system was used for the MMAE simulation with linearized model against the full VISTA flight control system with nonlinear aerodynamic database.

Next, the normal and lateral acceleration at the center of gravity were computed by the following equations:

$$A_{n\text{-}cg} = -(Z_\alpha\alpha + Z_q q + Z_u u + Z_{\delta S}\delta S + Z_{\delta F}\delta F + \\ + Z_{\delta LEF}\,\delta LEF) \tag{6}$$

$$A_{y\text{-}cg} = -(Y_\beta\beta + Y_p p + Y_F F + \\ + Y_{(\delta SL\text{-}\delta SR)}(\delta S_L - \delta S_R) + Y_{\delta F}\delta F + Y_{\delta R}\delta R) \tag{7}$$

where $A_{n\text{-}cg}$ and $A_{y\text{-}cg}$ are the longitudinal and lateral accelerations at the center of gravity (cg) and the dimensional derivatives are shown in the unprimed body axes for convenience. Using Eq.(3), the measurement equation is given by

$$z(t_l) = \begin{matrix} u \\ \alpha \\ q \\ A_{n\text{-}cg} \\ p \end{matrix} = \begin{matrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & z_{n1} & z_{n2} & z_{n3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{matrix} \begin{matrix} \theta \\ u \\ \alpha \\ q \\ \phi \end{matrix} +$$

$$\begin{matrix} r \\ A_{y\text{-}cg} \end{matrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & z_{y1} & z_{y2} & z_{y3} \end{bmatrix} \begin{matrix} \beta \\ p \\ r \end{matrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ z_{n4} & z_{n5} & z_{n6} & z_{n7} & 0 & z_{n8} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ z_{y4} & z_{y5} & z_{y6} & z_{y7} & z_{y8} & 0 \end{bmatrix} \begin{matrix} \delta S_L \\ \delta S_R \\ \delta F_L \\ \delta F_R \\ \delta R \\ \delta LEF \end{matrix} + v(t_I) \tag{8}$$

where the subscripts n and y represent the normal and lateral axes, respectively.

2. Evaluation of measurement noise covariance matrix

The sensor noise variances were determined by using flight test data from NASA's F-8 *Crusader* aircraft. Based upon the noise correlation time constants (0.02 sec for angular rates, 0.01 sec for accelerations, and 0.005 sec for the angle-of-attack), the bandwidth of the sensor noise is very large as compared to the frequency response of the aircraft. We therefore conclude that the sensor noise is essentially white over the bandwidth of the aircraft. The rms (root mean square) noise values are listed in table 3:

Table 3. Sensor noise rms. Values.

Variable	rms Noise	Units
u	0.1	[ft/s]
α	0.10 – 0.30	[deg]
q	0.15 – 0.50	[deg/s]
$A_{n\text{-}cg}$	0.02 – 0.06	[g's]
p	0.75 – 2.00	[deg/s]
r	0.15 – 0.50	[deg/s]
$A_{y\text{-}cg}$	0.003 – 0.04	[g's]

Conservative estimates, relative to anticipated VISTA sensor performance, for σ values are established as 0.1 ft/sec, 0.004 rad, 0.006 rad, 1 ft/sec^2, 0.02 rad/sec, 0.006 rad/sec, and 0.6 ft/sec^2, respectively.

3. Actuator dynamics model

The VISTA control surface actuators are modeled as fourth order actuators. The transfer function used to describe each of the actuators is given by

$$\begin{aligned} \mathbf{T(s) = \delta C/\delta C_{cmd}} \\ &= (20.2)((141.4)(5214.5)/[(s + 20.2)(s + 141.4)\cdot \\ &\quad \cdot (s^2 + 107.0s + 5214.5) \end{aligned} \tag{9}$$

where δC is the actuator (control surface) position and δC_{cmd} is the command surface position. The left and right stabilators, flaperons, and the rudder use this fourth-order model to represent the surface dynamics. The leading edge flap is represented as a first-order lag given by the equation [3]

$$T_{LEF}(s) = 16.0 / (s + 16.0). \tag{10}$$

The MMAE software allows these control surfaces to be modeled as either a first-order, second-order, or fourth-order representation. The first-order approximation, implemented in the filters, allows for a realistic simulation without the complexity of a higher order state-space model:

$$T_C(s) = 20.2 / (s + 20.2) \tag{11}$$

It should be noted here that previous efforts encountered robustness difficulties associated with reduced-order actuator models. Therefore, as mentioned earlier, the leading edge flap is always modeled as a first-order lag. However, for the most part, the first-order model approximation of Eq.(11) is used in both truth the model and the filters. Some cases were run with the fourth-order representation in the model and the first-order representation in the filters. The results indicated some minor performance degradation. Since the actual model in the aircraft is effectively fourth-order, actual implementation would require the filters to contain the fourth-order actuator model to maximize the algorithm's performance. Final implementable controllers would most likely incorporate first-order filter design models, and the truth model repeats this first-order model in order to keep the computational load down, without any severe misrepresentations of performance attributes. Actuator position and rate limiting are included in the truth model. Note that, and as mentioned earlier, the flight control system contains other limiting functions, such as internal signal and rate limiting, and pilot command limiting.

4. Actuator and sensor failures

In general, the performance of the MMAE algorithm in the present investigation can be characterized as good. This is true for single and multiple hard failures. The results are good for all single hard failures of actuators and sensors using a pulse train dither signal. In this section, actuator and sensor failures will be discussed in view of hard and soft failures.

1) Actuator failures: Actuator failures are commonly grouped into two categories: a) hard, and b) soft. Hard failures are represented, as the complete loss of actuator control with the surface trailing edge positioned as zero degrees of deflection. The leading edge flap is not included in the failure scenarios. This surface is not flight-critical and its inclusion only complicates the ambiguity problems with the other flight-critical surfaces. While the leading edge flap's omission simplifies the ambiguity issue, a filter could be designed to detect this failure if desired. Hard actuator failures are modeled by zeroing out the columns of the B-matrix (i.e., in Eq.(1)). Soft failures are represented as a partial loss of actuator control authority (as opposed to modeling a soft failure as a partial loss of actuator rate capability). Arbitrary percentages are selected to represent the loss of control authority. The critical issue is whether the algorithm will be able to blend the outputs of the fully functional filter with the appropriate hard failure filter to achieve the proper control for the soft failure scenario. Soft failures are introduced by multiplying the appropriate columns of the B-matrix in the truth model with the percentage of control authority available.

2) Soft failures: Sensor failures are also grouped into two categories, that is, hard and soft. Hard sensor failures are modeled by assuming the complete loss of the sensor's output. All of the flight-critical sensors are included in this model. Zeroing out the appropriate row in the H-matrix represents a hard

sensor failure. Unlike actuator failures in which one surface can affect many different residuals, sensor failures have a direct relationship to the associated scalar residuals and are easily detected. Soft failures are modeled by assuming increased sensor noise (as opposed to a bias). Increasing the appropriate sensor variance matrix, R, within the truth model represents soft sensor failures. Detection of a soft sensor failure is considerably harder when the elemental variance increment, from the sensor noise covariance matrix is small. We conclude this section by mentioning multiple failures. Multiple failures include all of the two-failure combinations: dual hard actuator failures, dual hard sensor failures, hard actuator and sensor failure; dual soft actuator failures, dual soft sensor failures, soft actuator and sensor failures; combinations of hard actuator with soft sensor failures, and combinations of soft actuator with hard failures. Multiple failures are implemented using the "moving bank" hierarchical structure in Section 4.4. Moreover, multiple failures are implemented according to a delay sequence, with the time between failures selected by the user. Multiple failures require the generation of a bank of filters designed for the first failure and any second failure including a no-failure filter. Partial failures provide an example of the robustness of the MMAE algorithm, to be handled by blending the estimates from the appropriate hard-failure elemental filter and the fully functional aircraft elemental filter.

5. The Dryden wind model

The development of the Dryden wind model is taken from MIL-STD-1797A [9]. The design model computed in this investigation assumes the value of turbulence rms. value, σ = 1(flight turbulence). Pilot perception of turbulence is often significantly different from the categories listed in the MIL-STD-1797A. Sufficient evidence exists to suggest a value of σ = 1 may more accurately describe light to moderate turbulence. The addition of the Dryden wind model modifies the aircraft dynamics model of Eq.(5). Relating aircraft motion to air mass motion (i.e., wind) in an inertial reference frame yields:

$$v_{a/i} = v_{a/m} + v_{m/i} \tag{12}$$

where the subscripts a, i, and m are aircraft, inertial, and air mass, respectively. Then the u from Eq.(5) can be expressed as follows:

$$du/dt = X_\theta'(\theta - \theta_g) + X_u'(u - u_g) + X_\alpha'(\alpha - \alpha_g) + \\ +X_q'(q - q_g) + X_{\delta S}'\delta S + X_{\delta F}'\delta F + X_{\delta LEF}'\delta LEF \tag{13}$$

where θ_g, u_g, and α_g represent the gust or air mass velocity effect on the state variables. Note that the elements from Eq.(13) must be added (i.e., augment) to Eq.(5).

In the present investigation the discrete gust model was used. From MIL-STD-1797A the magnitude of the wind shear is computed from the relations

$$v_x = k\sigma_u, \quad v_y = k\sigma_v, \quad v_z = k\sigma_w \tag{14}$$

where k varies from 0.4 to 2.6, and is tuned for each gust to excite the natural frequencies of the aircraft. The rms intensities of forward, lateral, and vertical turbulence are described by σ_u, σ_v, and σ_w, respectively. The magnitude of the wind shear in each component direction are described by v_x, v_y, and v_z. After calculating the velocities for a given disturbance level the gust variables are converted into the state variables of the Dryden wind model. The discrete gust velocities are defined as follows:

$$v_x = \Delta u_g, \quad v_y = \Delta v_g, \quad v_z = \Delta w_g \tag{15a}$$

with the equalities

$$\Delta\alpha_g = (1/V_T)\Delta w_g, \ \Delta\beta_g = (1/V_T)\Delta v_g \tag{15b}$$

where V_T is the aircraft velocity. From these equations result the following relationship:

$$\Delta u_g = v_x. \tag{16}$$

Substituting the values Δu_g, $\Delta\alpha_g$, and $\Delta\beta_g$ into the Dryden wind model equations provides additional terms with

$$\Delta\alpha_g = v_z / V_T \tag{17a}$$

$$\Delta\beta_g = v_y / V_T \tag{17b}$$

which to augment the wind disturbance model, with an additional term $d_g(t)$, as

$$dx_g/dt = A_g x_g(t) + G_g w_g(t) + d_g(t) \tag{18}$$

where $d_g(t)$ is given by the equation

$$d_g(t) = [(-V_T/L_u)v_x \quad 0 \quad (-1/2L_w)v_z \quad (-\pi V_T/\delta b L_w) \\ 0 \quad 0 \quad (-1/2L_v)v_y \quad (\pi V_T/\delta b L_v)v_y \quad 0^T]^T. \tag{19}$$

III. Truth model

The aircraft truth model is represented as a continuous-time dynamic system (see Eqns.(1 – 3))

$$dx_{TM}/dt = A_{TM}x_{TM}(t) + B_{TM}u_{TM}(t) + G_{TM}w_{TM}(t) + \\ + d_{TM}(t) \tag{20}$$

with an output equation given by

$$y(t) = C_{TM}x_{TM}(t) + D_{TM}u_{TM}(t) \tag{21}$$

and the discrete-time measurement equation

$$z(t_l) = H_{TM}x_{TM}(t_l) + D_{TM}u_{TM}(t_l) + v(t_l) \tag{22}$$

where $w_{TM}(t)$ is a zero-mean white Gaussian noise of strength Q_{TM}, statistically representing aircraft disturbances generated by random wind buffet. For the 4^{th}-order actuator representations with state elements δS_L, δS_R, etc. $u_{AC}(t)$ becomes δL_{Lcmd}, δS_{Rcmd}, etc. The $\delta(\cdot)_{cmd}$ represents the input command to the actuator model, and $\delta(\cdot)$ is the output of the actuator which is provided as input to the aircraft model. The aircraft state model, Eq.(5), is augmented to incorporate the actuator and wind disturbance models. In this investigation, the wind models are the only source of disturbances to the aircraft model. Explicitly, the nonzero components in the augmented truth model matrices G_{TM} and d_{TM} come from the wind model matrices G_g and d_g.

Assuming an inertial navigation system onboard the aircraft and using the F-16 dynamics, the velocities and gyro related white noise terms of the filter state model are given by the following equations [10]:

$$d(\Delta V_x)/dt = -g_E(\Delta\theta_y + \psi_y) + (\rho_z + 2\Omega_z)\Delta V_y + \\ + (-\psi_y\Omega_x + \psi_x\Omega_y - \theta_z)V_y + \Delta A_x \quad (23)$$

$$d(\Delta V_y)/dt = g_E(\Delta\theta_x + \psi_x) - (\rho_z + 2\Omega_z)\Delta V_x - \\ - (-\psi_y\Omega_x + \psi_x\Omega_y - \theta_z)V_x + \Delta A_y \quad (24)$$

$$d(\Delta\theta_x)/dt = -(\Delta V_y - \psi_z V_y)/R_E + \rho_z\Delta\theta_y \quad (25)$$

$$d(\Delta\theta_y)/dt = (\Delta V_x + \psi_z V_x)/R_E - \rho_z\Delta\theta_x \quad (26)$$

$$d\psi_x/dt = -(\rho_y + \Omega_y)\psi_z + (\rho_z + \Omega_z)\psi_y + \\ + e_x + \Delta e_x \quad (27)$$

$$d\psi_y/dt = (\rho_x + \Omega_x)\psi_z - (\rho_z + \Omega_z)\psi_x + \\ + e_y + \Delta e_y \quad (28)$$

$$d\psi_z/dt = -(\rho_x + \Omega_x)\psi_y - (\rho_y + \Omega_y)\psi_x + \\ + e_z + \Delta e_z \quad (29)$$

where $\Delta\theta_x$, $\Delta\theta_y$ = INU position error in polar coordinates
Ω_x, Ω_y, Ω_z = Earth-rate torquing terms in platform coordinates
ρ_x, ρ_y, ρ_z = vehicle-rate torquing terms
e_x, e_y, e_z = INU gyro bias terms
Δe_x, Δe_y, Δe_z = gyro-related white noise
ψ_x, ψ_y, ψ_z = INU misalignment angles
ΔA_x, ΔA_y = accelerometer-related white noise
ΔV_x, ΔV_y = INS velocity errors
R_E = local Earth radius of curvature
g_E = local gravity.

Note that the vertical velocity error, ΔV_z, is not estimated by the filter, it is calculated to calibrate V_z with, say, *GPS* (Global Positioning System) velocity whenever GPS is available and is processed as a filter measurement. The Kalman filter contains a model that describes how the INU (Inertial Navigation Unit) errors evolve with time, and a model relating the reference sensor data (such as GPS) to the INU data.

The design models in this investigation were created with first-order representations, while the full-scale wind model is replaced with the "appropriately scaled" white noise. Moreover, the white noise used in the truth model is scaled relative to strength levels used in the design models. An empirical approach is taken to provide white noise that is noticeable, but not offensive from a pilot's perspective. The noise strength levels were chosen to resemble light turbulence. Also, the flight control system incorporates cross-channel effects. Limitations in the size and scope of this investigation and the unnecessary computational loading precluded the addition of these models.

IV. Multiple model adaptive control and estimation

In this section we will briefly review the well-known MMAC and MMAE-based controller methodologies.

1. The MMAC algorithm: an overview

The implementation of the multiple model adaptive model controller for this application requires the design of a series of individual elemental controllers corresponding to a set of failure modes at a given flight condition. Moreover, these controllers can be gain-scheduled to form an integrated flight control system capable of functioning anywhere within the specified flight envelope. Failure modes can be developed by analyzing the flight control system, which is within its proposed operating environment. For example, a fighter is designed to operate in a hostile environment. The failure modes for a fighter might include the partial or complete loss of an actuator or control surface. One would also include the biasing or complete loss of any sensor, etc. In its most simplest implementation using a constant-gain controller, each of the elemental controllers consists of a Kalman filter based upon the aircraft plant corresponding to the failed condition, cascaded with a deterministic optimal controller gain, both based on an assumed parameter value, say, a_k. If a is a parameter representing a failure space, then a_k is one of the possible discrete realization of a, in this case one possible failure mode. Each of the elemental Kalman filters utilize the same measurement realization vector

$$z(t_i) = z_i \quad (30)$$

and generate state estimates $x(t_i^+)$, and a measurement residual vector $r_k(t_i)$, where

$$r_k(t_i) = z_i - H_k(t_i)x_k(t_i^-) \quad (31)$$

where the subscript k corresponds to the parameter value of a_k assumed by the particular elemental filter. The residual vector is difference between the observed measurement vector and the predicted value from the filter. $H_k(t_i)$ is the measurement matrix value from the *kth* filter, and $x_k(t_i^-)$ is the Kalman filter estimate of the state vector before the incorporation of the measurement at time t_i. Figure 1 illustrates the MMAC structure with its bank of k elemental controllers, each of which corresponds to a given failure mode.

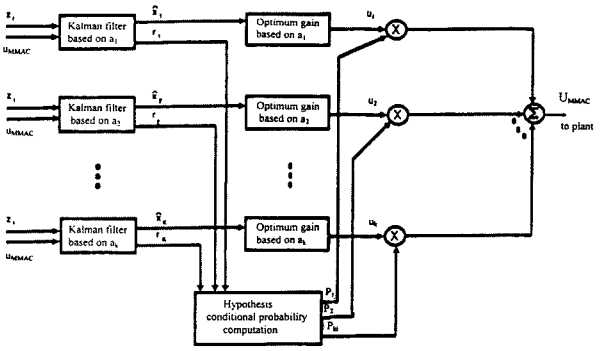

Fig. 1. The multiple model adaptive controller.

Now, if the *kth* filter represents the current plant (i.e., the failure condition), the magnitude of the residual from the filter will be small, on the order of the size of the residual's standard deviation as computed within the *kth* filter itself. The magnitudes of the residuals from the other "mismatched" filters, representing other failure scenarios, will be larger than anticipated by the filter-computed standard deviation. Finally, the elemental controller with the most accurate representation of the current plant configuration will provide the smallest residual to the MMAC algorithm. Therefore, the algorithm will assign a probability to each of the filters (i.e., assumed failure scenarios) based upon the magnitude of the associated residual

Referring to Fig.1, we note that after construction of the residuals, a number of techniques can be used to assign probabilities to each of the elemental controllers and generate an adaptive control vector, u_{MMAC}. In addition, we note that previous research efforts have addressed the formulation of the control vector u_{MMAC} using a maximum a posteriori (MAP) approach, along with a Bayesian approach [5]. In the MAP technique, the single u_k corresponding to the highest probability p_k (i.e., with residuals having the smallest magnitude relative to the filter-computed standard deviation) would be selected as the control vector to apply to the plant. From the above discussion, we note that the Bayesian form of the MMAC algorithm produces a probability-weighted average of the form [2], [6]

$$u_{MMAC}(t_i) = \sum_{k=1}^{K} u_k[x_k(t_i^+), t_i]\, p_k(t_i) \tag{32}$$

where K is the total number of elemental filter-controlled combinations, $u_k[x_k(t_i^+), t_i]$ is a deterministic optimal full-state feedback control law, and $x_k(t_i^+)$ is the state estimate generated by a Kalman filter for the *kth* failure mode modeled within the multiple model adaptive controller. Thus, the control applied to the plant is the probability-weighted average of all of the individual elemental controller outputs. The Bayesian form is ideally suited to address failures that may not be explicitly modeled within the MMAC hierarchy. As such, it is capable of blending the output of two or more elemental controllers to adapt to an unmodeled failure condition.

While the Bayesian form of the MMAC algorithm is a powerful control technique, it requires some modification for practical implementation. That is, the blending of the outputs of each of the elemental controllers allows for the adaptation of the algorithm for failure scenarios not explicitly modeled within the MMAC hierarchy. However, an optimum solution would suggest that only the output of models with hypotheses that "bound" the actual failure be included in the probability-weighted control vector u_{MMAC}.

2. MMAE algorithm development

Let a denote a vector of uncertain parameters in a given linear stochastic state model for a dynamic system. In the present investigation, we will assume that a resides in the space of actuator and/or sensor failures within a flight control system. In essence, the a parameters can affect the model structure or the noises entering it. Also, the continuous range of a is discretized in order to provide a computationally feasible solution. Define now the hypothesis conditional probability $p_k(t_i)$ as the probability that a assumes a value a_k (for $k = 1, 2, \ldots, K$), conditioned on the observed measurement history to time t_i:

$$p_k(t_i) = Prob(a = a_k \mid Z(t_i) = Z_i) \tag{33}$$

then it can be shown that $p_k(t_i)$ can be evaluated recursively for all k via the iteration [3], [4]:

$$p_k(t_i) = f_{z(ti)\mid a,\, Z(ti-1)}(z_i \mid a_k, Z_{i-1}) \cdot p_k(t_{i-1}) \,/ \sum_{j=1}^{K} f_{z(ti)\mid a,\, Z(ti-1)}(z_i \mid a_j, Z_{i-1}) \cdot p_j(t_{i-1}). \tag{34}$$

The measurement history of the random vector $Z(t_i)$ is made up of partitions $z(t_1) \ldots z(t_i)$ which correspond to the measurement vectors available at the sample times $t_1 \ldots t_i$. The realization Z_i of the measurement history vector has partitions $z_1 \ldots z_i$. The Bayesian minimum mean square error estimate of the state is the probability weighted average [7], [8]:

$$x(t_i^+) = \mathrm{E}\{x(t_i) \mid Z(t_i) = Z_i\} = \sum_{k=1}^{K} x_k(t_i^+) p_k(t_i) \tag{35}$$

where $x_k(t_i^+)$ is the state estimate generated by a Kalman filter based on the assumption that the parameter vector equals a_k. More explicitly, let the model corresponding to a_k be described by (an "equivalent discrete-time model" for a continuous-time system with sampled data measurements) [4], [11]:

$$x_k(t_{I+1}) = \Phi_k(t_{I+1}, t_i) x_k(t_i) + B_k(t_i) u(t_i) + G_k(t_i) w_k(t_i) \tag{36}$$

$$z(t_i) = H_k(t_i) + v_k(t_i) \tag{37}$$

where x_k is the state, u is a control unit, w_k is a discrete-time zero-mean white Gaussian dynamic noise of covariance $Q_k(t_i)$ at each t_i, z is the measurement vector, and v_k is a discrete-time zero-mean white Gaussian measurement noise of covariance $R_k(t_i)$ at t_i, assumed independent (and thus uncorrelated) of w_k. The initial condition $x(t_o)$ is modeled as Gaussian, with mean x_{ko} and covariance P_{ko} and is assumed independent of w_k and v_k. Based on this set of models, the usual Kalman filter is specified by the measurement update as follows:

$$A_k(t_i) = H_k(t_i) P_k(t_i^-) H_k^T(t_i) + R_k(t_i) \tag{38}$$

$$K_k(t_i) = P_k(t_i^-) H_k^T(t_i) A_k^{-1}(t_i) \tag{39}$$

$$x_k(t_i^+) = x_k(t_i^-) + K_k(t_i)[z_i - H_k(t_i) x_k(t_i^-)] \tag{40}$$

$$P_k(t_I^+) = P_k(t_I^-) - K_k(t_i) H_k(t_i) P_k(t_I^-) \tag{41}$$

and the time propagation relations:

$$x_k(t_{I+1}^-) = \Phi_k(t_{I+1}, t_i) x_k(t_I^+) + B_k(t_i) u(t_i) \tag{42}$$

$$P_k(t_I^-) = \Phi_k(t_{I+1}, t_i) P_k(t_I^+) \Phi_k^T(t_{I+1}, t_i) + G_k(t_i) Q_k(t_i) G_k^T(t_i). \tag{43}$$

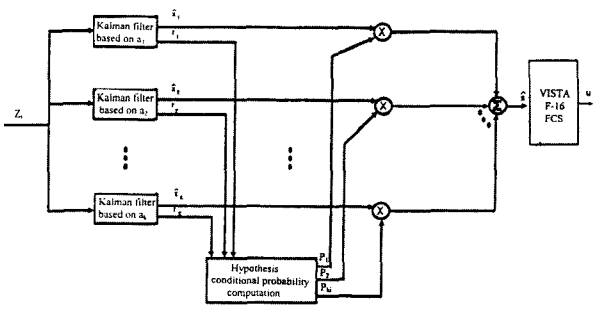

Fig. 2. Multiple model adaptive estimation block diagram.

From Fig.2 it is noticed that the multiple model adaptive estimation filtering algorithm is composed of a bank of K separate Kalman filters, each of which is based on a particular

value $a_1 \ldots a_K$ of the parameter vector a, as depicted in Fig.2. Therefore, when the measurement z_i becomes available at time t_i, the residuals $r_1(t_i) \ldots r_K(t_i)$ are generated in the K filters as shown by the bracketed term in Eq.(40), and used to compute $p_1(t_i) \ldots p_K(t_i)$ via Eq.(34). Each numerator density function in Eq.(34) is given by the following well-known relation [4]:

$$f_{z(ti)\ a,\ Z(ti-1)}(z_i \mid a_k, Z_{i-1}) = (1 / [(2\pi)^{m/2} \mid A_k(t_i) \mid^{1/2}) \cdot \cdot \exp [-(1/2) r_k^T(t_i) A_k^{-1}(t_i) r_k(t_i)] \quad (44)$$

where m is the measurement dimension and $A_k(t_i)$ is the residual covariance calculated in the *kth* Kalman filter in Eq.(38). The denominator in Eq.(34) is simply the sum of all the computed numerator terms and thus is the scale factor required to ensure that the $p_k(t_i)$'s sum to one [2]. Thus, the filter probabilities must satisfy two constraints. These are:

$$p_k(t_i) \geq 0 \quad (45)$$

$$\sum_{k=1}^{K} p_k(t_i) = 1. \quad (46)$$

In operation, one expects the residuals of the Kalman filter with the hypothesis that best matches the true system state to have mean squared value most in consonance with its internally computed covariance, $A_k(t_i)$. The performance of the algorithm depends on the existence of significant differences in the characteristics of "correct" vs. "mismatched" filters' residuals. Each filter should be tuned for best performance when the true values of the uncertain parameters match the assumed value for these parameters. The addition of substantial amounts of the dynamics model, often used to guard against filter divergence, should be avoided as it tends to mask the differences between good and bad models. Although the MMAE algorithm appears similar to the MMAC controller, fundamental differences exist between the two algorithms. The intent of the MMAE algorithm is to generate a composite $x_{MMAE}(t_i^+)$ from each of the K elemental filter estimates as shown in Fig.2. In the present investigation, we are compensating for parameter uncertainties in B and H representing actuator and sensor failures; a hard actuator failure is modeled with a column of B going to zero, and a sensor hard failure is represented by a row of H being zeroed. The multiple model adaptive estimation algorithm produces a composite state estimate vector used by the flight control system.

3. MMAE-based control

The MMAE-based was selected over the MMAC-based control for various reasons. In this investigation, the control system was selected from existing control systems. Previous efforts designed control systems with the objective of reconfiguring the estimator and controller portions of the system upon the isolation of a failure. The purpose of the present investigation is to demonstrate the algorithm's robustness and emerging maturity by applying the algorithm to a full-scale control system in use with the USAF. In addition, the primary focus of this investigation is the detection and isolation of failures, not reconfigurability of the control laws. (the possibility of loss of control for some failure combinations could be anticipated). Another important consideration for the anticipated). Another important consideration for the selection of the MMAE-based control is the demonstration of this technique on a modern fighter test aircraft. As mentioned earlier, the residuals from each of the Kalman filters (see Fig.2), representing a different hypothesized failure determine the appropriate magnitude of the probability assigned to each filter. Consequently, failures are detected and identified according to the probability assigned to the filter.

4. Hierarchical modeling

The MMAE/MMAC algorithm provides flexibility in its application to single and multiple failures. That is, given a relatively small finite set of failure conditions, it might be feasible to design an elemental controller for every failure condition within the set. However, for a flight control system application, this is not practical. In essence, modeling the single and dual failures of K sensors and/or actuators, would require $[1 + K + (K!/(K-2)!2!)]$ elemental filters, with one no-failure filter, K single-failure filters, and $K!/(K-2)!2!$ dual-failure filters. An alternative method used in previous flight control research involves a "moving bank" of $(1 + K)$ filters. This technique significantly reduces the computational burden by requiring fewer elemental filters to be "on line" at any given time. Figure 3 identifies the hierarchical structure of the "moving bank" of filters.

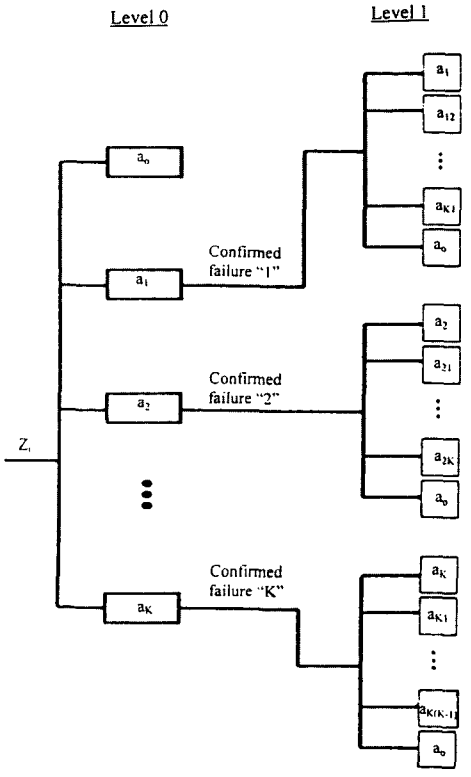

Fig. 3. Hierarchical modeling.

Note that at level 0, filters a_j through a_K correspond to the single-failure filters. Filter a_o is the no-failure filter. The algorithm continuously monitors the system with the $(1 + K)$ Level 0 filters. Therefore, at the isolation of a single failure, the al-

gorithm will activate the appropriate Level 1 bank of $(1 + K)$ elemental filters designed for the isolated failure alone, and any other possible second failure. The addition of a no-failure filter allows the algorithm to "back out" of the decision tree structure and return to Level 0 in the event of a first misidentified failure, intermittent failure, etc.

For a multi-sensor system, the K local estimators and the global estimator share the same *a priori* information on the state x. Furthermore, x is assumed to be Gaussian with mean x and variance M; $v_o(t_i)$, $i = 1,2,\ldots,K$ are independent Gaussian vectors with zero-mean and variance R_i. Let now the error covariance of the local and global estimates be

$$P \equiv \mathrm{E}\{(x - x_i)\,(x - x_i)^T\} \tag{47}$$

where x is the estimate of x. Then, we have the following Lemma:

Lemma:

$$P^{-1} = \sum_{i=1}^{K} P_i^{-1}(K-1)M^{-1} \tag{48a}$$

$$P^{-1}x = \sum_{i=1}^{K} P_i^{-1}x_i - (K-1))M^{-1}x. \tag{48b}$$

Proof:

Standard estimation results give

$$P^{-1} = M^{-1} + \sum_{i=1}^{K} H_i^T R_i^{-1} H_i \tag{49a}$$

$$P^{-1} = M^{-1} + H_i^T R_i^{-1} H_i \tag{49b}$$

$$x = x + P_i \sum_{i=1}^{K} H_i^T R_i^{-1}(z_i - H_i x) \tag{49c}$$

$$x = x + P_i H_i^T R_i^{-1}(z_i - H_i x). \tag{49d}$$

Thus, in this particular case, if a hierarchical scheme of estimation is used, it is only necessary to transmit the local estimates and error covariances to the higher level.

V. Simulation and discussion of the results

The purpose of this investigation was to construct a multiple model adaptive estimator-based controller using a computer simulation of the VISTA flight control system. The program uses many subroutines, however, it should be noted that the VISTA subroutine can be removed and replaced with any other flight control subroutine. However, changing the subroutine will require a change to the MMAE code since it is unlikely that the number of sensors and actuators would be equivalent. The flight control system in this investigation differs from the full VISTA slight control system by the following aspects:

1) Dynamics above 40 rad/sec are ignored (secondary dynamic effects).

2) First-order models replace fourth-order actuator models

3) The flight control system is modeled as a continuous system using Euler integration techniques instead of a 40 Hz digital controller. Eventually, the sampled data nature of the controller will be captured.

The flight control system does include both the longitudinal and later-directional axes, gain scheduling, position and rate limits (in the controller software as well as in the physical hardware implementation), angle-of-attack limiting, the leading edge flap controller, trim effects, command and position breakouts, surface biases, and the yaw structural filter, as in the full VISTA flight control system. The aerodynamic model is linearized about a flight condition about a flight condition of 0.4 Mach at 20,000-ft. The model includes the pitch attitude, velocity, angle-of-attack, pitch rate, sideslip angle, roll angle, roll rate, and yaw rate. The control vector includes the left and right stabilator positions, the left and right flaperon positions, the rudder, and the leading edge flap. Since the model is linearized about a selected flight condition, the thrust is constant in this perturbation model. The performance analysis of the MMAE-based control algorithm begins by inducing hard modeled failures, soft modeled failures, and finally unmodeled failures. Following the evaluation of single-failure performance, dual-failure scenarios were investigated. Initially, the time between the first failure and the induction of the second failure will be large. As confidence in he ability of the system to isolate both failures increases, the time between the failures will be reduced.

Undoubtedly, the most obvious limitation throughout this investigation is the use of a linearized VISTA aerodynamic model. However, since the interest in this investigation lies within the area of failure detection and isolation, as opposed to control, this is not considered a restrictive limitation. Another limitation is the use of reduced order actuators, which allow a simpler implementation in the model. Frequency response characteristics for first vs. fourth-order actuators are essentially equivalent for the frequencies of interest. The Dryden wind model was simulated using the appropriate shaping filters driven by white Gaussian noise of the appropriate strength Within the elemental filters, sensor dynamics were presented as essentially instantaneous, with the second output being corrupted by discrete-time white Gaussian noise of appropriate covariances. One of the strengths of this investigation is its realistic models, particularly of the VISTA flight control system. The limitations listed within this investigation do not seriously bound the validity of the results. In the present investigation, position and rate limits for the VISTA aircraft were as shown in table 4.

Table 4. Actuator position and rate limits.

Control Surface	Limits	Units
δS - Stabilator	± 21.0 ± 60.0	[deg] [deg/s]
δF - Flaperons	+ 20.0 - 23.0 ± 61.0	**[deg]** [deg/s]
δR – Rudder	± 30.0 ±120.0	[deg] [deg/s]

Fig. 4 shows the modeled states.

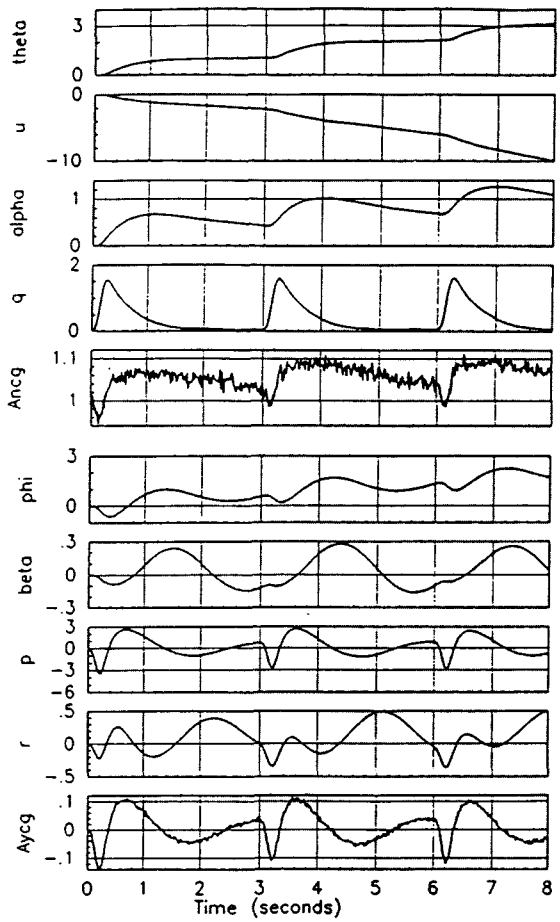

Fig. 4. States for no failure scenario.

Specifically, Fig.4 contains the modeled states. The algorithm correctly identifies the no-failure filter from the starting probability of 0.75 within 0.25 seconds. The probability grows from 0.75 to 0.988 (maximum) within this time. The excitation of the pulse forced the residuals, corresponding to each of the filters with incorrect hypotheses, to grow large as compared to their internally computed covariances. In contrast, the residuals of the filter with the correct hypothesis fall within the 3σ bounds. The MMAE algorithm assigns probabilities based upon the residuals characteristics. Except for a minor probability dropout at 5.95 seconds, the probability is locked on the fully functional (i.e., no failure) filter. The minor probability dropout occurs near the six-second pulse, a time at which ambiguity arises due to the application of the pulse. The pulse excites the system, however, the pulse also provides a transient. The dropout at 5.95 seconds is picked up by the angle-of-attack failure filter. This phenomenon is consistent throughout the hard- failure scenarios. It is an anomaly caused by a 3σ noise spike which violates the 3σ bounds representing he internally computed covariance and causes a momentary loss of probability, which happens to be picked up by the angle-of-attack failure filter. Note the probability spike is momentary, consistent between cases, and the algorithm instantly returns the probability to the correct filter, namely, the fully functional aircraft filter. In Fig.4, the pulse dither signal can clearly be observed at 0.3 and 0.6 seconds. All of the variables are centered around zero with the exception of the normal acceleration (centered at 1-*g*). These variables are relative to the trim conditions. The trim angle-of-attack is 10.06°, at the trim pitch angle. The trim velocity is 414.8 ft/sec. All other trim states are zero with the exception of the trim normal acceleration which is 1-*g*. Finally, a small change in the pulse amplitude can alter the ramping with no change in performance, and the ramping can be easily controlled.

VI. Conclusion

A multiple model adaptive estimation algorithm is applied to the NF-16D variable stability in-flight simulator test aircraft (VISTA) at a low dynamic pressure flight condition (i.e., 0.4 Mach at 20,000-ft.). A complete F-16 flight control system is modeled containing the longitudinal and lateral-directional axes. Single and dual actuator and sensor failures were simulated including the following: complete actuator failures, partial actuator failures, complete sensor failures, increased sensor noise, sensor biases, dual complete actuator failures, dual complete sensor failures, and combinations of actuator and sensor failures. Failure scenarios are examined in both maneuvering and straight and level flight conditions. Single scalar residual monitoring techniques were evaluated with suggestions for improved performance. A Kalman filter is designed for each hypothesized failure condition.

Based upon the projected cost of future military aircraft, sufficient motivation exists within the Air Force to increase the survivability of present and future aircraft. The MMAE/MMAC algorithm may fulfill that need. The application of this algorithm to a sophisticated flight control system currently in service with the Air Force provides a realistic and valuable test. In this investigation, the construction of the MMAE/MMAC algorithm demonstrated the hierarchical structure utilized in addressing multiple failures, was discussed. The assignment of elemental controller probabilities using a modified Bayesian form allows the algorithm to control outputs in an effort to address unmodeled failures. Additional voting techniques may be necessary to resolve ambiguities in the detection and isolation of failures. A large number of questions remain to be answered concerning the algorithm's ability to isolate single and multiple failures, distinguish between simultaneous failures, and differentiate between longitudinal and lateral directional failures.

Finally, in this investigation thirteen elemental Kalman filters are designed encompassing: a no failure filter left stabilator failure filter, a right stabilator failure filter, a left flaperon failure filter, a right flaperon failure filter, a rudder failure filter, a velocity sensor failure filter, an angle-of-attack sensor failure filter, a pitch rate sensor failure filter, a normal acceleration sensor filter, a roll rate sensor failure filter, a yaw rate sensor failure filter, and a lateral acceleration sensor failure filter. The blended state estimates are sent to the VISTA flight control system. A hierarchical "moving bank" structure is utilized for multiple failure scenarios. White Gaussian noise is

included to simulate the effects of atmospheric disturbances, and to simulate the effects of sensor noise. Each elemental Kalman filter is compared to the truth model with a selected failure. Filters with residuals that have mean square values most in consonance with their internally computed covariance are assigned the higher probabilities.

References

[1] M. Athans and C.B.Chang, "Adaptive estimation and parameter identification using the multiple model estimation algorithm," Technical Note 1976-28, ESD-TR-1976, Lincoln Laboratory, Lexington, Mass., June 1976.

[2] M. Athans et al., "Stochastic control of the F-8C aircraft using a multiple adaptive control (MMAC) method – Part I: Equilibrium flight, *IEEE Trans. on Auto. Cont.*, vol. AC-22, no. 5, pp. 768-780, October, 1977.

[3] J.J.D'Azzo and C.H.Houpis, *Linear Control System Analysis and Design*, 2nd edition, McGraw-Hill, New York, 1981.

[4] C.K.Chui and G.Chen, *Kalman Filtering with Real-Time Applications*, 3rd edition, Springer, New York, 1999.

[5] C.S.Greene and A.S.Wilsky, "An analysis of the multiple model adaptive control algorithm," *Proc. IEEE Conf. Decision and Control*, Albuquerque, New Mexico, pp. 1142-1145, December., 1980.

[6] P.A.Ioanou and J.Sun, *Robust Adaptive Control*, Prentice-Hall International edition, 1996.

[7] D.T.Magill, "Optimal adaptive estimation of sampled stochastic processes," *IEEE Trans. AC*, vol. AC-10, no. 5, pp. 434-439, October, 1965.

[8] P.S.Maybeck and K.Hentz, "Investigation of a moving-bank multiple model adaptive algorithm," *AIAA Journal of Guidance, Control, and Dynamics*, 10, 1, pp. 90-96, Jan.-Feb. 1987.

[9] Military Specification – "Flying qualities of piloted airplanes," MIL-STS-1797A, Government Printing Office, Washington, D.C., January 30, 1990.

[10] G.M.Siouris, *Aerospace Avionics Systems: A Modern Synthesis*, Academic Press, New York, 1993.

[11] G.M.Siouris, *An Engineering Approach to Optimal Control and Estimation Theory*, Wiley-Interscience, New York, 1996.

George M. Siouris

He received the B.E.E. degree in 1958 and the M.S. degree in mathematics in 1962, both from New York University, and the D.Eng.Sc. degree in aerospace engineering in 1971 from the Technical University of Berlin, Germany. Dr. Siouris has 16 years of industrial research and development experience from his association with the Sperry Gyroscope Company, Honeywell Aeronautical Division, the M.I.T Instrumentation Laboratory, and Dornier System in Germany. His professional experience includes theory and applications of inertial navigation, missile guidance, and satellite attitude control and stabilization systems. In 1979, he joined the Aeronautical Systems Division (now Aeronautical Systems Center/AFMC), Directorate of Avionics Engineering, Wright-Patterson AFB, Ohio, where he was engaged in the theory, applications, and modeling of inertial navigation systems for aircraft and missiles, air-to-air and air-to-ground missile guidance, Inertial Navigation System/Global Positioning System integration, and ring laser/fiber optic gyro technology. From January 1993 until his retirement in January 1997, he was with the Development Planning Directorate, Studies and Analyses Division, where he was engaged in the design/analysis of flight control systems, Situational Awareness/Situational Assessment models for use in real time man-in-the-loop air combat simulations, and acted as consultant to System Program Offices. In 1981 he was appointed lecturer in the Department of Electrical and Computer Engineering, AF Institute of Technology, and in 1985 Adjunct Professor at the same Department. He remains one to this date. Upon his retirement from the Dept. of the Air Force, he received the Air Force's *Outstanding Civilian Service Award.*

Presently, he is a consultant in avionics and weapon systems. He is the author of numerous technical papers, book reviews, and conference papers. From 1983-1993 he was the editor for Navigation/Tracking Systems in the *IEEE Transactions on Aerospace and Electronic Systems*, and associate editor from 1993-1996. His primary research interests includes theory and applications of inertial navigation and guidance systems, modeling of physical systems, weapon systems, and applications of modern control and estimation theory.