

Multiple Model Adaptive Estimation with Filter Spawning

KENNETH A. FISHER

PETER S. MAYBECK, Fellow, IEEE
Air Force Institute of Technology

Multiple model adaptive estimation (MMAE) with filter spawning is used to detect and estimate partial actuator failures on the VISTA F-16. The truth model is a full six-degree-of-freedom simulation provided by Calspan and General Dynamics. The design models are chosen as 13-state linearized models, including first order actuator models. Actuator failures are incorporated into the truth model and design model assuming a “failure to free stream.” Filter spawning is used to include additional filters with partial actuator failure hypotheses into the MMAE bank. The spawned filters are based on varying degrees of partial failures (in terms of effectiveness) associated with the complete-actuator-failure hypothesis with the highest conditional probability of correctness at the current time. Thus, a blended estimate of the failure effectiveness is found using the filters’ estimates based upon a no-failure hypothesis, a complete actuator failure hypothesis, and the spawned filters’ partial-failure hypotheses. This yields substantial precision in effectiveness estimation, compared with what is possible without spawning additional filters, making partial failure adaptation a viable methodology in a manner heretofore unachieved.

Manuscript received September 21, 1999; revised November 29, 2001 and February 19, 2002; released for publication May 2, 2002.

IEEE Log No. T-AES/38/3/06425.

Refereeing of this contribution was handled by X. R. Li.

The views expressed in this article are those of the authors and do not reflect the official policy of the United States Air Force, Department of Defense, or the U.S. Government.

Authors’ address: Dept. of Electrical and Computer Engineering, Air Force Institute of Technology, 2950 P St., Bldg. 640, Wright-Patterson AFB, OH 45433-7765, E-mail: (Peter.Maybeck@afit.edu).

U.S. Government work not protected by U.S. copyright.

0018-9251/02/\$17.00 2002 IEEE

I. INTRODUCTION

Multiple model adaptive estimation (MMAE) with filter spawning [6, 8, 10, 11] is applied to the detection and estimation of partial actuator failures on the variable-stability in-flight simulator test aircraft (VISTA) F-16 to address the need for fault-tolerant and survivable flight control systems in aircraft. In particular, flight control systems should be able to detect and estimate actuator and sensor failures, and reconfigure the control law of the aircraft to obtain the least degradation in performance as possible. The success of such an algorithm may eliminate the need for layers of redundant sensors and actuators, improve performance during sensor failures, and reduce the hazards associated with actuator failures.

MMAE [12] has been chosen based on its ability to respond rapidly and correctly to real-world failures, and its detection, estimation, and control performance for *complete* actuator and sensor failures [4, 5, 9, 14, 15, 17–19, 21–23]. In considering *partial* actuator failures, “filter spawning” (described in detail in Section IV) has been chosen to include additional filters based on partial actuator failure hypotheses in the MMAE bank, once the MMAE has declared which specific actuator has undergone a failure of some as-yet-unknown extent. This minimizes the number of elemental filters required at any one time, while providing finer parameter discretization as needed to resolve the degree of failure of any partially failed actuator. Such accurate resolution is essential to subsequently declared elemental filter hypotheses for detecting any second failure: that the aircraft has undergone a partial failure of the estimated degree on that particular actuator and any other (or no other) complete failure. Multiple faults are handled efficiently by means of the hierarchical structure of [4, 5, 15, 21–23], requiring only the number of elemental filters for declaring a *single* fault to be online at any time.

To handle the onset of failures, i.e. time-varying parameter values or hypotheses about the current real-world failure status, a lower bound is imposed on the probabilities $p_k(t_i)$ (see (13)) computed by an MMAE based on static parameter assumptions [1, 13]. The alternative of a Markov model for $p_k(t_i)$ variations in time and the associated interacting multiple model (IMM) algorithm [2, 10, 11, 24] was considered but rejected for this application for two reasons. First, the IMM algorithm requires specification of an entire matrix of failure state transition probabilities for its Markov model, which often is significantly more difficult for a designer to provide accurately than a single lower bound to accomplish basically the same impact for this application. Secondly, the elemental filters within the IMM algorithm are continually

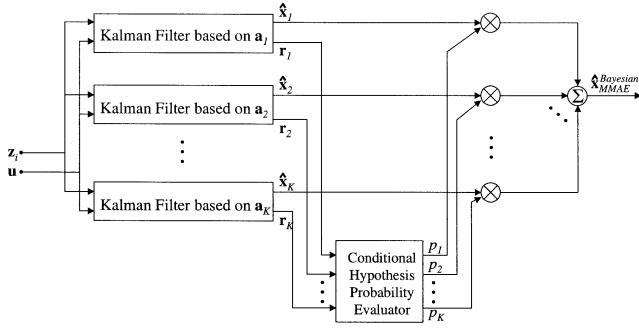


Fig. 1. MMAE Bayesian state estimate block diagram.

reinitialized, whereas an elemental filter is only restarted in the proposed MMAE when there is an indication of divergence (the quadratic form in the exponent of (14) exceeds a bound). The authors believe that this allows better disambiguation among different failure status hypotheses because the effects of those different modeling assumptions are allowed to accumulate in the filter residuals over multiple sample periods.

II. MMAE DEVELOPMENT

An MMAE is composed of a bank of K Kalman filters, each based on a different model of the system. For this research, each model incorporates a different realization of the uncertain parameter vector \mathbf{a} , denoting the type and extent of failure. The dynamics model for the k th filter is assumed to be a linear time-invariant, discrete-time system of the form

$$\mathbf{x}_k(t_{i+1}) = \Phi_k \mathbf{x}_k(t_i) + \mathbf{B}_{dk} \mathbf{u}(t_i) + \mathbf{G}_{dk} \mathbf{w}_{dk}(t_i) \quad (1)$$

$$\mathbf{z}(t_i) = \mathbf{H}_k \mathbf{x}_k(t_i) + \mathbf{v}_k(t_i) \quad (2)$$

where \mathbf{x} is the vector of system states, \mathbf{u} the control input vector, \mathbf{z} the measurement vector, and where dynamics driving noise $\mathbf{w}_{dk}(t_i)$ and measurement corruption noise $\mathbf{v}_k(t_i)$ are discrete-time, white Gaussian noises with the statistics:

$$\begin{aligned} E[\mathbf{w}_{dk}(t_i)] &= \mathbf{0} & E[\mathbf{w}_{dk}(t_i)\mathbf{w}_{dk}^T(t_i)] &= \mathbf{Q}_{dk} \\ E[\mathbf{v}_k(t_i)] &= \mathbf{0} & E[\mathbf{v}_k(t_i)\mathbf{v}_k^T(t_i)] &= \mathbf{R}_k \\ E[\mathbf{w}_{dk}(t_i)\mathbf{v}_k^T(t_i)] &= \mathbf{0}. \end{aligned} \quad (3)$$

Each of the K filters is propagated and updated using (6)–(12) presented later. The Bayesian method is used to form the state and parameter estimates from the MMAE shown in Fig. 1, expressed as

$$\hat{\mathbf{x}}_{\text{MMAE}}^{\text{Bayesian}}(t_i^+) = \mathbf{E}\{\mathbf{x}(t_i) \mid \mathcal{Z}(t_i) = \mathcal{Z}_i\} = \sum_{k=1}^K \hat{\mathbf{x}}_k(t_i^+) p_k(t_i) \quad (4)$$

$$\hat{\mathbf{a}}_{\text{MMAE}}^{\text{Bayesian}}(t_i) = \mathbf{E}\{\mathbf{a}(t_i) \mid \mathcal{Z}(t_i) = \mathcal{Z}_i\} = \sum_{k=1}^K \mathbf{a}_k(t_i) p_k(t_i) \quad (5)$$

where $\mathbf{a}_k(t_i)$ corresponds to each elemental filter's hypothesis, $\mathcal{Z}(t_i)$ is the measurement history through time t_i , and $p_k(t_i)$ is the MMAE-computed conditional probability that the k th hypothesis best describes the real world at time t_i (the computation itself is shown later in (13)). In this application, $\mathbf{a}_k(t_i)$ is actually a *scalar* failure status parameter, with discrete values denoting a fully functional aircraft, any single sensor having totally failed, any single surface/actuator having totally failed, any single surface/actuator having partially failed to a certain assumed level of effectiveness, etc. Partial failures can be due to some loss of hydraulic pressure, battle damage to a control surface, or the like, with the result being that the control surface delivers only a portion of the control effectiveness anticipated under unfailed conditions.

III. TRUTH AND DESIGN MODEL

This research is based on the VISTA F-16. The measurements used on-board are angle-of-attack; pitch, roll, and yaw rates; and normal and lateral acceleration (measured at the pilot's station). The control surfaces are the leading edge flaps, flaperons, stabilators, and a rudder. The flaperons span much of the trailing edge of the wings and may be commanded symmetrically as flaps to produce a pitching moment and asymmetrically as ailerons to produce a rolling moment. The stabilators compose the horizontal tail and may be commanded in the same manner as the flaperons. The redundant nature of the control surfaces makes this aircraft appealing for failure detection because an alternate input command can be found to produce an equivalent output response if a surface has failed [21, 22].

The truth model is the VISTA simulation which runs as part of the Simulation Rapid-prototyping Facility (SRF) and incorporates General Dynamics' VISTA F-16 simulation software with variable stability flight control system software provided by Calpsan. The SRF VISTA F-16 simulation provides a full six-degree-of-freedom truth model incorporating the aircraft's nonlinear equations of motion, advanced actuator modeling, the complete Block 40 controller operating at 64Hz (also the sample rate of the MMAE), and the aileron-to-rudder interconnect used to provide coordinated turns. Also, the truth model includes a zero-order Dryden wind model generating "moderate" turbulence [4, 5, 14, 19], failure simulation allowing single or dual and partial or complete failures [3–5, 9, 17, 18, 21, 22], sensor noise [4, 5], and a linear lateral acceleration measurement [21, 22]. All sensor noise variances and dynamics noise strengths in both the simulation and the MMAE elemental filter designs were representative of the VISTA F-16, and they are specified in [6].

A design model is found such that the MMAE and control redistribution algorithm based on it [21, 22] can be implemented on the 64Hz digital flight computer of the VISTA F-16, while capturing the important characteristics of the truth model. The design model is selected as a linear, time-invariant, discrete-time model, of reduced order compared with the truth model. The nonlinear, time-variant, continuous-time model is *linearized* about the nominal aircraft configuration “up-and-away” (leading edge flaps up and landing gear up, a typical configuration for steady, level flight) and a nominal flight condition of Mach 0.4 at 20,000 ft (making failure detection more difficult due to low dynamic pressure). The actuator design model for MMAE elemental filters is a first order lag, whereas the truth model incorporates a fourth order model.

Failures are incorporated into the truth model as well as the design model in terms of an *effectiveness factor* ϵ , with the range $0 \leq \epsilon \leq 1$. An effectiveness of 0% ($\epsilon = 0$) indicates a 100% failure, or a “full failure.” Conversely, an effectiveness of 100%, ($\epsilon = 1$), indicates a 0% failure, or “no failure.” Effectiveness values $0 < \epsilon < 1$ indicate “partial failures.” For sensor failures, the truth model is the loss of the desired noise-free signal portion of the sensor reading while still admitting sensor noise into the measurement. The design model matches the truth model by multiplying a *row* of the measurement matrix \mathbf{H} by the effectiveness factor ϵ (for $\epsilon = 0$ or 1). For actuator failures, the truth model multiplies the *commanded* actuator position by ϵ . For full failures ($\epsilon = 0$), the appropriate surface would be commanded to its trim position, approximating “failure to free stream.” For no failures ($\epsilon = 1$), the appropriate surface would be given the full command. For partial failures ($0 < \epsilon < 1$), scaling the command by ϵ results in the effect of reduced command authority. The design model matches the truth model by multiplying a *column* of the input matrix \mathbf{B}_d by the effectiveness factor ϵ .

Dither is introduced in order to excite the system and enhance identifiability, chosen as a sinusoid with a frequency of 15 Hz [9, 21, 22]. The limits on maximum allowable dither magnitude response used for this research is set to be ± 0.1 gs in the longitudinal channel and ± 0.2 gs in the lateral channel, considered *tolerable* to the pilot [20].

IV. FILTER SPAWNING

This research presents MMAE with filter spawning [6, 8, 10, 11] to consider the conditions of no failure, complete actuator failures, complete sensor failures, and partial actuator failures. Prior research has explored the *detection* and *estimation* of partial actuator failures without the use of spawning. One

method used the fully functional aircraft filter¹ and the completely failed actuator filter to obtain an estimate of the effectiveness value [3]. While this method is easy to implement, the discretization was too crude to produce good blending; that is, the probability tended to flow to *one* filter rather than to both filters in proportion to their “correctness.”

To resolve the discretization problem when *estimating* the extent of an actuator failure, filter spawning incorporates additional filters (based on various levels of effectiveness) into the MMAE bank. In the “filter spawning technique,” an actuator failure is *detected* using a maximum a posteriori (MAP) approach to be described; thus the decision logic *first* declares *which* specific actuator (or sensor) has failed. Then, the filter spawning technique *estimates* the effectiveness of that specific actuator using the Bayesian blending approach as described in (5). Additional filters based on the actuator failure detected but with different effectiveness values, called spawned filters, are included into the MMAE bank to provide for finer parameter discretization among hypothesized filters, increasing the number of discretization points used in blending. Filters are spawned *only* in the channel corresponding to the detected actuator failure, so as not to require an overwhelming number of elemental filters to be running in parallel at any one time.

The flow chart for the MMAE with Filter Spawning algorithm used in this research is shown in Fig. 2. The algorithm uses a hierarchical structure [7] where there are different sets of hypotheses, or banks, used for the filters within the MMAE. Using only one bank at a time, MMAE can cover a large number of hypotheses by changing banks while maintaining a small number of filters.

In the “Initialize” block, the models and “banks” used by the MMAE algorithm must be placed into memory and the initial conditions are defined. The algorithm uses one bank consisting of 12 filters as well as five additional banks consisting of 15 filters each, as described in Table I. Each bank contains the fully functional filter, the complete actuator failure filters, and the complete sensor failure filters. In addition, banks 2–6 contain three² spawned filters, each at a different effectiveness level. Notice the set of spawned filters for each of the five *spawning* banks is based on a different actuator failure. Each spawned filter is based on an effectiveness level $\epsilon_{i_a i_s}$, where i_a is the actuator index and i_s is the effectiveness level index. Since the first 12 filters are the same in

¹For convenience, the filter’s hypothesis is used as a descriptor for the filter, i.e., the “fully functional aircraft filter” is the filter hypothesizing a fully functional aircraft.

²Three spawned filters were chosen as a first-cut discretization. More filters results in finer discretization which provides a better estimate, but, the computational load is increased.

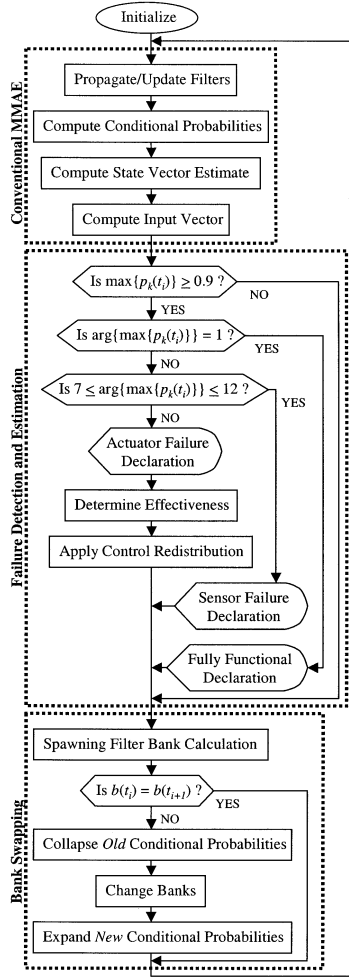


Fig. 2. MMAE with filter spawning flow chart.

banks 2–6, changing banks is actually the method used to decide *which* spawned filters, if any, will be used. The algorithm as implemented for this research only uses banks 2–6, using all 15 filters all the time. Bank 1 could be used to exclude the spawned filters when not needed, however an alternate approach is described later. The state vector estimate is initialized with trim condition values. The MMAE arbitrarily starts in bank 2 assuming a fully functional aircraft, where each failure filter's conditional probability will be equal to the probability lower bound 0.001 and the fully functional filter's conditional probability is assigned such that the sum of the probabilities equals one (0.986).

The “Conventional MMAE Computation” group in Fig. 2 propagates each elemental filter through

$$\hat{\mathbf{x}}_k(t_{i+1}^-) = \Phi \hat{\mathbf{x}}_k(t_i^+) + \mathbf{B}_{dk} \mathbf{u}(t_i) \quad (6)$$

$$\mathbf{P}_k(t_{i+1}^-) = \Phi \mathbf{P}_k(t_i^+) \Phi^T + \mathbf{Q}_d \quad (7)$$

and updates each elemental filter through

$$\mathbf{A}_k(t_i) = \mathbf{H}_k \mathbf{P}_k(t_i^-) \mathbf{H}_k^T + \mathbf{R} \quad (8)$$

$$\mathbf{K}_k(t_i) = \mathbf{P}_k(t_i^-) \mathbf{H}_k^T \mathbf{A}_k^{-1}(t_i) \quad (9)$$

$$\mathbf{r}_k(t_i) = \mathbf{z}_i - \mathbf{H}_k \hat{\mathbf{x}}_k(t_i^-) \quad (10)$$

$$\hat{\mathbf{x}}_k(t_i^+) = \hat{\mathbf{x}}_k(t_i^-) + \mathbf{K}_k(t_i) \mathbf{r}_k(t_i) \quad (11)$$

$$\mathbf{P}_k(t_i^+) = \mathbf{P}_k(t_i^-) - \mathbf{K}_k(t_i) \mathbf{H}_k \mathbf{P}_k(t_i^-) \quad (12)$$

where the subscript k was dropped from matrices independent from \mathbf{a}_k and the system matrices Φ , \mathbf{B}_{dk} , and \mathbf{H}_k are dependent on the current bank. The conditional probabilities can be computed [13] as

$$p_k(t_i) = \frac{f_{\mathbf{z}(t_i)|\mathbf{a}, \mathcal{Z}(t_{i-1})}(\mathbf{z}_i | \mathbf{a}_k, \mathcal{Z}_{i-1}) p_k(t_{i-1})}{\sum_{j=1}^K f_{\mathbf{z}(t_i)|\mathbf{a}, \mathcal{Z}(t_{i-1})}(\mathbf{z}_i | \mathbf{a}_j, \mathcal{Z}_{i-1}) p_j(t_{i-1})} \quad (13)$$

where

$$f_{\mathbf{z}(t_i)|\mathbf{a}, \mathcal{Z}(t_{i-1})}(\mathbf{z}_i | \mathbf{a}_k, \mathcal{Z}_{i-1}) = \beta_k e^{[-1/2 \mathbf{r}_k(t_i)^T \mathbf{A}_k^{-1}(t_i) \mathbf{r}_k(t_i)]}. \quad (14)$$

The β_k term is removed from (14) to reduce the number of sensor false alarms [16, 23], and the conditional probabilities in (13) are lower bounded [1, 13] with 0.001 [15, 23]. The state vector estimate is obtained using the Bayesian blending method, expressed as

$$\hat{\mathbf{x}}_{\text{MMAE}}^{\text{Bayesian}}(t_i^+) = \frac{\sum_{k \in \mathcal{B}} \hat{\mathbf{x}}_k(t_i^+) p_k(t_i)}{\sum_{k \in \mathcal{B}} p_k(t_i)} \quad (15)$$

where \mathcal{B} contains all k such that $p_k(t_i) \geq 0.005$. The blending lower bound of 0.005 is used to exclude state estimates with extremely low probabilities since the probability lower bound of 0.001 prevents probabilities from reaching zero. The measurement vector $\mathbf{z}(t_i)$ is reconstructed to reflect the state estimate [21, 22], expressed as

$$\hat{\mathbf{z}}_{\text{MMAE}}^{\text{Bayesian}}(t_i) = \mathbf{H} \hat{\mathbf{x}}_{\text{MMAE}}^{\text{Bayesian}}(t_i^+) \quad (16)$$

where the \mathbf{H} is the “no sensor failure” measurement matrix. The measurement vector estimate, $\hat{\mathbf{z}}_{\text{MMAE}}^{\text{Bayesian}}(t_i)$, is used to replace the raw measurement $\mathbf{z}(t_i)$ as an input to the Block 40 FCS, thereby allowing for normal operation despite sensor failures.

The “Failure Detection, Estimation, and Control” group in Fig. 2 applies MMAE to failure detection (using the conditional probabilities from models based on hypothesized failures). Before detection can occur, the probabilities must be redefined to assure a proper comparison. The conditional probability of failure of the actuator on which the spawned filters are based is dispersed among the complete actuator failure filter and the spawned filters modeling partial failures of that actuator. As a remedy, the conditional probability of the i_a th actuator filter is redefined

$$p'_{i_a}(t_i) = p_{i_a}(t_i) + p_{13}(t_i) + p_{14}(t_i) + p_{15}(t_i) \quad (17)$$

where, from Table I, i_a is the filter index for the complete actuator failure and 13–15 are the filter indices for the spawned filters. Thus, the sum of the conditional probabilities for the filters based on the i_a th actuator for any effectiveness will be compared

TABLE I
MMAE with Filter Spawning Bank Description

		Bank						
		1	2	3	4	5	6	
Filter	FF	1	FF	FF	FF	FF	FF	FF
	Complete	2	LS	LS	LS	LS	LS	LS
		3		RS	RS	RS	RS	RS
		4		LF	LF	LF	LF	LF
	Actuator	5	RF	RF	RF	RF	RF	RF
				6	R	R	R	R
	Sensor			7	AA	AA	AA	AA
		8	PR	PR		PR	PR	PR
		9	NA	NA		NA	NA	NA
		10	RR	RR		RR	RR	RR
		11	YR	YR		YR	YR	YR
		12	LA	LA		LA	LA	LA
	Failure	13		LS ϵ_{21}	RS ϵ_{31}	LF ϵ_{41}	RF ϵ_{51}	R ϵ_{61}
		14		LS ϵ_{22}	RS ϵ_{32}	LF ϵ_{42}	RF ϵ_{52}	R ϵ_{62}
		15		LS ϵ_{23}	RS ϵ_{33}	LF ϵ_{43}	RF ϵ_{53}	R ϵ_{63}

Note: fully functional (FF), left stabilator (LS), right stabilator (RS), left flaperon (LF), right flaperon (RF), rudder (R), angle of attack (AA), pitch rate (PR), normal acceleration (NA), roll rate (RR), yaw rate (YR), lateral acceleration (LA).

with the conditional probabilities of the remaining filters.

A failure is *detected* if the highest conditional probability exceeds some threshold, expressed as

$$\max_{1 \leq k \leq 12} [p'_k(t_i)] \geq P_{\text{threshold}} \quad (18)$$

where $p'_k(t_i)$ denotes the conditional probability $p'_{i_a}(t_i)$ as defined in (17) for $k = i_a$ and for $k \neq i_a$ the conditional probabilities as defined in (13). The threshold, $P_{\text{threshold}}$, for this research is 0.9; a lower threshold would detect failures faster but also increase false alarms [4, 5, 17, 18]. If no conditional probability exceeds this level, then no failure is assumed to have occurred and the failure

declaration remains at the previous declaration, i.e., $\hat{\mathbf{a}}_{\text{MMAE}}^{\text{MAP}}(t_i) = \hat{\mathbf{a}}_{\text{MMAE}}^{\text{MAP}}(t_{i-1})$. If a conditional probability does exceed this level, the failure is detected using a MAP approach expressed as

$$\hat{\mathbf{a}}_{\text{MMAE}}^{\text{MAP}}(t_i) = \mathbf{a}_j(t_i) \quad \text{where} \quad j = \arg \left\{ \max_{1 \leq k \leq 12} [p'_k(t_i)] \right\} \quad (19)$$

where $\hat{\mathbf{a}}_{\text{MMAE}}^{\text{MAP}}$ is the failure estimate and \mathbf{a}_j is the hypothesized failure for the j th filter. If $j = 1$ in (19), then $\hat{\mathbf{a}}_{\text{MMAE}}^{\text{MAP}}(t_i)$ corresponds to a fully functional aircraft occurring. If $7 \leq j \leq 12$, then $\hat{\mathbf{a}}_{\text{MMAE}}^{\text{MAP}}(t_i)$ corresponds to a sensor failure occurring. If $2 \leq j \leq 6$, then $\hat{\mathbf{a}}_{\text{MMAE}}^{\text{MAP}}(t_i)$ corresponds to an actuator failure occurring.

The extent of the failure is *estimated* in terms of effectiveness after a failure is detected. Because only complete sensor failures are considered in this research, the effectiveness estimate is $\hat{\epsilon} = 0$ for detected sensor failures. On the other hand, because complete *and* partial actuator failures are considered, actuator failures are *estimated* using the Bayesian blending approach shown in Fig. 2 as the block “Determine Effectiveness,” expressed as

$$\hat{\epsilon}(t_i) = \frac{p_j(t_i) + \epsilon_{j1}P_{13}(t_i) + \epsilon_{j2}P_{14}(t_i) + \epsilon_{j3}P_{15}(t_i)}{p_j(t_i) + p_{13}(t_i) + p_{14}(t_i) + p_{15}(t_i) + p_1(t_i)} \quad (20)$$

where, given that the failure index is j , as defined in (19), the hypothesized effectiveness values are $\epsilon_{j0} = 0$ for the complete actuator failure filter’s hypothesized effectiveness value; $\epsilon_{j1}, \epsilon_{j2}$, and ϵ_{j3} for the spawned filters’ hypothesized effectiveness values; and $\epsilon_{ff} = 1$ for the fully functional filter’s hypothesized effectiveness.

Control redistribution (CR) has been demonstrated to be an effective control scheme for the VISTA F-16 in the case of a single, complete actuator failure by *redistributing* control away from the failed actuator to the unfailed actuators [21, 22]. For partial failures, while it is recommended later to *refine* the effectiveness estimate using the empirically observed relationship between the estimate of the effectiveness and the true effectiveness rather than use $\hat{\epsilon}$ directly, the focus of this research is the detection and estimation, not control, of failures.

Because the banks only differ in the spawned filters, the “Bank Swapping” group in Fig. 2 simply chooses which spawned filters are used. The index corresponding to the *actuator* filter with highest probability at time t_i is used to decide which bank will be used at the next time step, t_{i+1} , expressed as

$$b(t_{i+1}) = \arg \left\{ \max_{2 \leq k \leq 6} [p'_k(t_i)] \right\} \quad (21)$$

where the maximum is taken only over the indices corresponding to the actuators. Now, the spawned filters in the MMAE bank used at the next time step, bank $b(t_{i+1})$, will correspond to the actuator with the highest probability at the current time step. The spawned filters are *always* used in this implementation; however, as an alternative, the spawned filters could be used only if the largest $p'_k(t_i)$ corresponds to an actuator failure. The spawned filters should not be used (i.e., use bank 1) if the largest $p'_k(t_i)$ is for a fully functional aircraft or sensor failure.

Notice in Table I that all banks share the same first 12 filters, but the spawned filters differ from bank to bank. Simply assigning the conditional probabilities of the old bank to the new bank is not valid. The spawned filters’ conditional probabilities must be “collapsed” into the complete failure actuator filter

on which the spawned filters are based. A rigorous “collapse” would be to recompute the conditional probabilities using (13) *without considering the spawned filters*. However, the formulation in (17) is used because it saves computation time and the probabilities only serve as initial conditions for the new bank. The probabilities are then “expanded” to the new spawned filters by dividing the probability equally among the complete-failure filter and the spawned filters. Although assigning a probability equal to some lower bound to the spawned filters and the remainder of the complete-actuator-failure filter’s original probability to itself could be used, there is no *prebiasing* of relative probabilities among the complete-failure filter and the three spawned filters in the “expansion” chosen.

V. SIMULATION RESULTS

The simulation starts at time equal to zero, with a fully functional aircraft in steady level flight at 0.4 Mach at 20,000 ft. At 1 s, a failure is introduced into the truth model. The failure is maintained for the duration of the simulation, or 7 additional seconds. While only actuator failures are shown, complete sensor failures were considered to *validate the design*. That is, the introduction of filter spawning is shown not to affect the detection of complete sensor failures as demonstrated in previous research [3, 9, 21, 22]. The following four discretization sets are considered where all actuators use the same level of discretization [6]: 1) 10%, 25%, 50%; 2) 20%, 40%, 60%; 3) 25%, 50%, 75%; and 4) 40%, 60%, 80%. Set 1 was chosen because other authors [3] hypothesized that estimating a *small* effectiveness value (i.e., a high percentage failure) would be easier than estimating a *large* effectiveness value. Set 2 was chosen after initial studies of set 1 showed that $\epsilon_{ia1} = 10\%$ was too close to $\epsilon = 0\%$ as hypothesized by the complete actuator failure filter. In contrast, set 4 was chosen to provide a discretization that favors estimating *large* effectiveness values over *small* effectiveness values. Set 3 was chosen to be impartial to small and large effectiveness values.

The “probability plot” is a plot of the conditional probability for each filter over time. The probability plot for a 20% left stabilator failure using the discretization set 1 is shown in Fig. 3. The subplots correspond to the 15 filters in the MMAE algorithm (see Table I for notation). In Fig. 3, the spawned filters $S_1 - S_3$ correspond to filters hypothesizing 10%, 25%, and 50% failures of the left stabilator, respectively. Ten Monte Carlo runs were performed for each simulation. (A subset of cases conducted with 20 runs revealed essentially no difference in computed statistics, and so ten-run analyses were employed for the many Monte Carlo studies performed to produce Figs. 3–8.) The mean is plotted as a solid

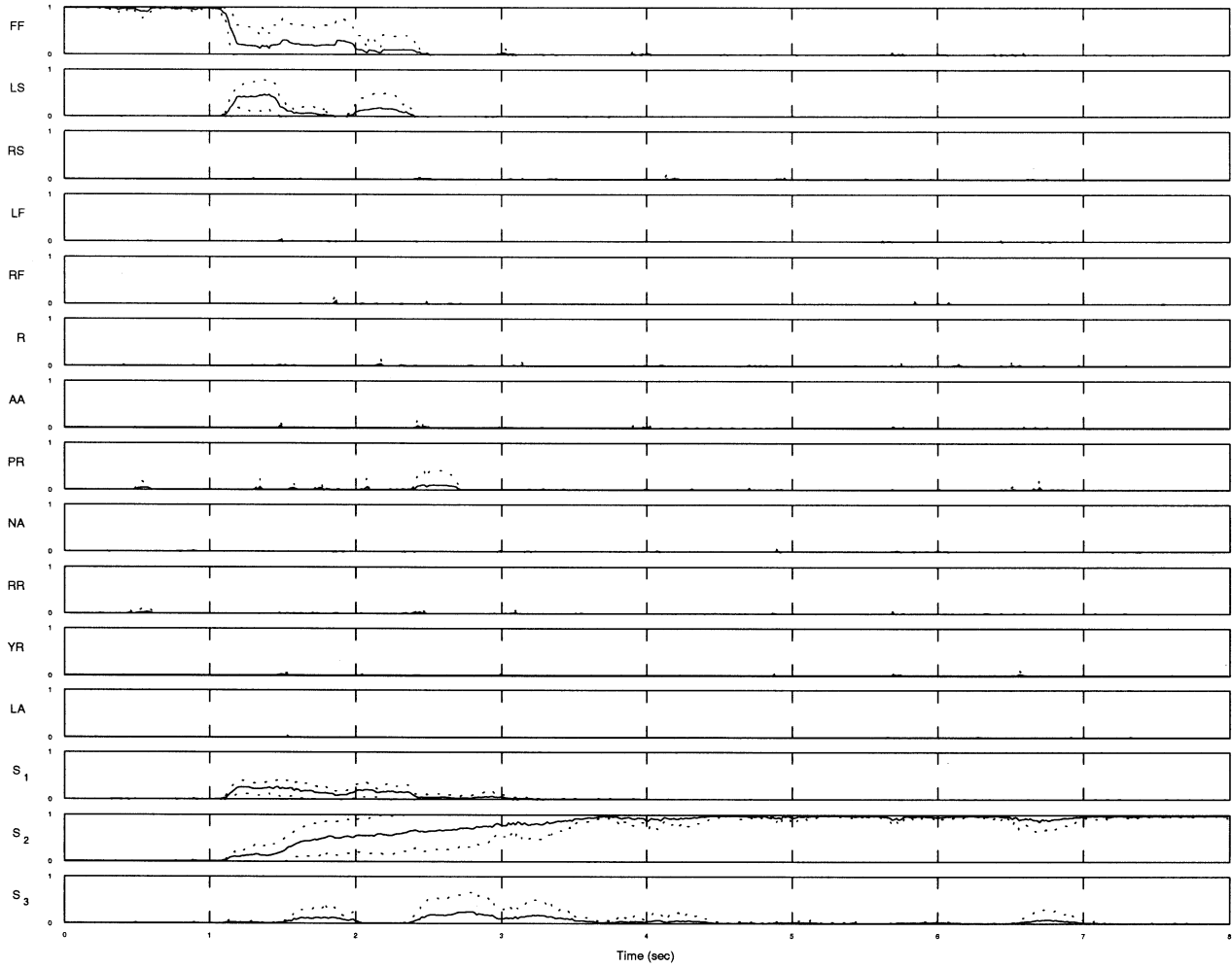


Fig. 3. Probability plot (all channels). Left stabilator failure, $\epsilon_{\text{true}} = 20\%$, spawned filter discretization: 10%, 25%, 50%.

line and $(\text{mean} \pm 1\sigma)$ bounds are plotted as dotted lines. Of most significance are the top two plots (probabilities for fully functional aircraft and complete left stabilator failure hypotheses) and the last three plots (probabilities for the three spawned filters). Notice that the channels with *wrong* hypotheses remain near zero during most of the simulation. These filters' probabilities verify that *wrong* filters do not obtain high probabilities.

A. Parameter Estimate Versus Time Plot

The “parameter estimate versus time plot” is a plot of the parameter estimate for each time sample, where the effectiveness estimate³, $\hat{\epsilon}$, is found using

³The parameter estimate consists of the failure index (i.e., identifying which actuator or sensor has failed) and the effectiveness in the case of an actuator failure. Here, since the algorithm has correctly detected the proper actuator failure, the remaining parameter estimate consists of an estimate of the effectiveness. Thus, parameter estimate and effectiveness estimate are used interchangeably.

(20) using the conditional probabilities for each filter at each time t_i and the effectiveness values of the spawned filters for the discretization set used. The left stabilator using set 4 spawned filters is shown in Fig. 4, where the subplots vary the true effectiveness from 0% to 100% in steps of 10%. Ten Monte Carlo runs were performed for each simulation, and the mean is plotted as a solid line, and $(\text{mean} \pm 1\sigma)$ bounds are plotted as dash-dot lines.

Plotting the parameter estimate versus time shows clearly the *convergence rate* of the parameter estimate. The mean *and* $\pm 1\sigma$ bounds give an indication of the time to converge to a solution. Notice the estimate converges by $t_i \approx 4$ s. Recalling the failure is introduced at 1 s, the estimate converges by 3 s after the failure. While the plots do not indicate it, a correct indication of *which* actuator has failed is found by $t_i \approx 2$ s. Thus, after a failure is *detected* (1 s after the failure), the algorithm must wait about 2 s (3 s after the failure) before an accurate *estimate* of actuator effectiveness (or its complement, degree of failure) is obtained.

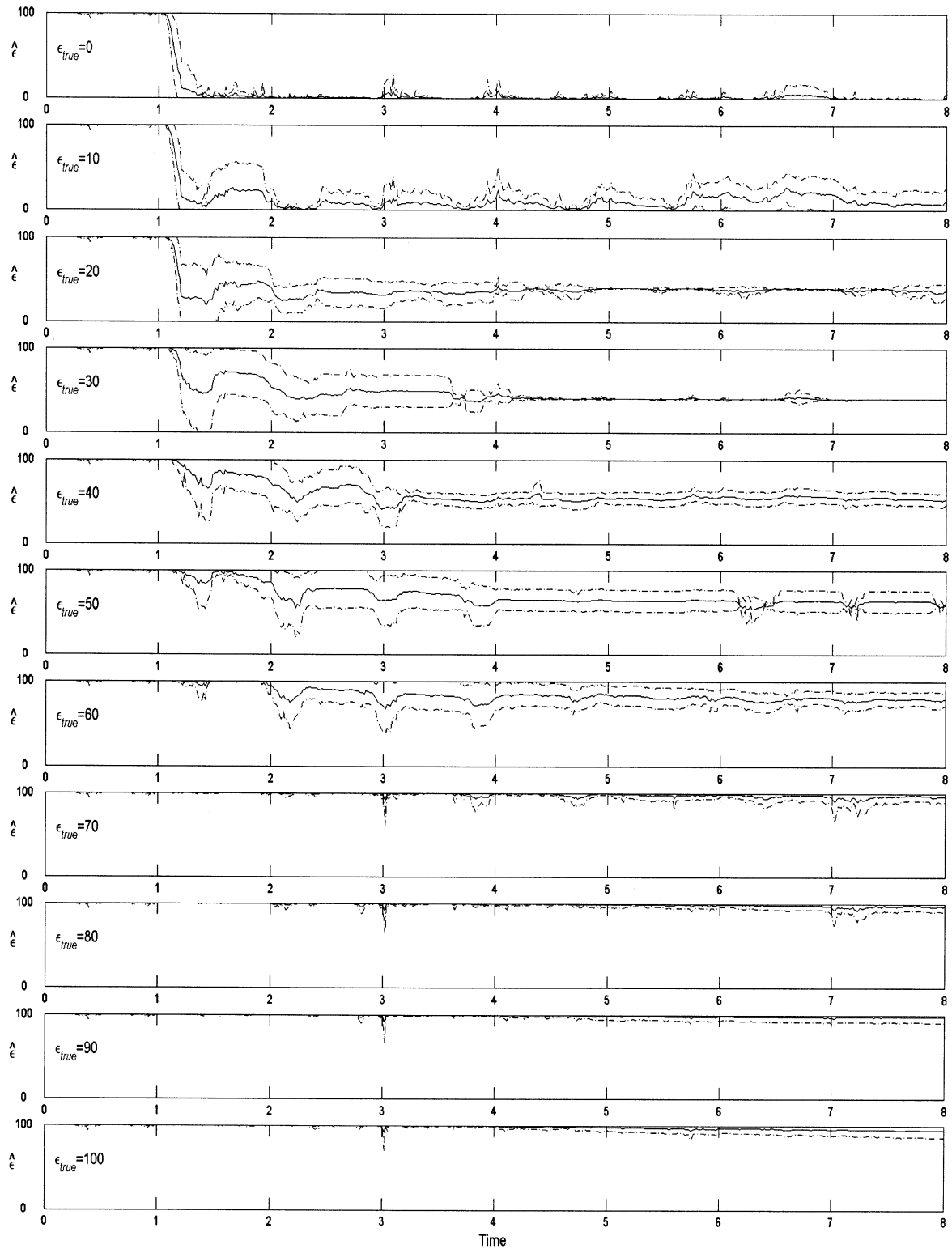


Fig. 4. $\hat{\epsilon}$ versus time plots for left stabilator failure using spawned filters 40%, 60%, 80%.

B. Parameter Estimate Versus True Parameter Plots

To demonstrate a more complete analysis, a plot of the parameter estimate versus time could be shown for the true parameter values $\epsilon_{true} = 0, 1, 2, \dots, 100$. Rather than do so and overwhelm the reader (or designer) with hundreds of plots, the

information is compacted instead into a digestible and useful form. The parameter estimates' mean and standard deviations are averaged over time for each true parameter value from the time when the estimate has essentially converged over the remainder of the simulation, i.e., from 4 s to 8 s.

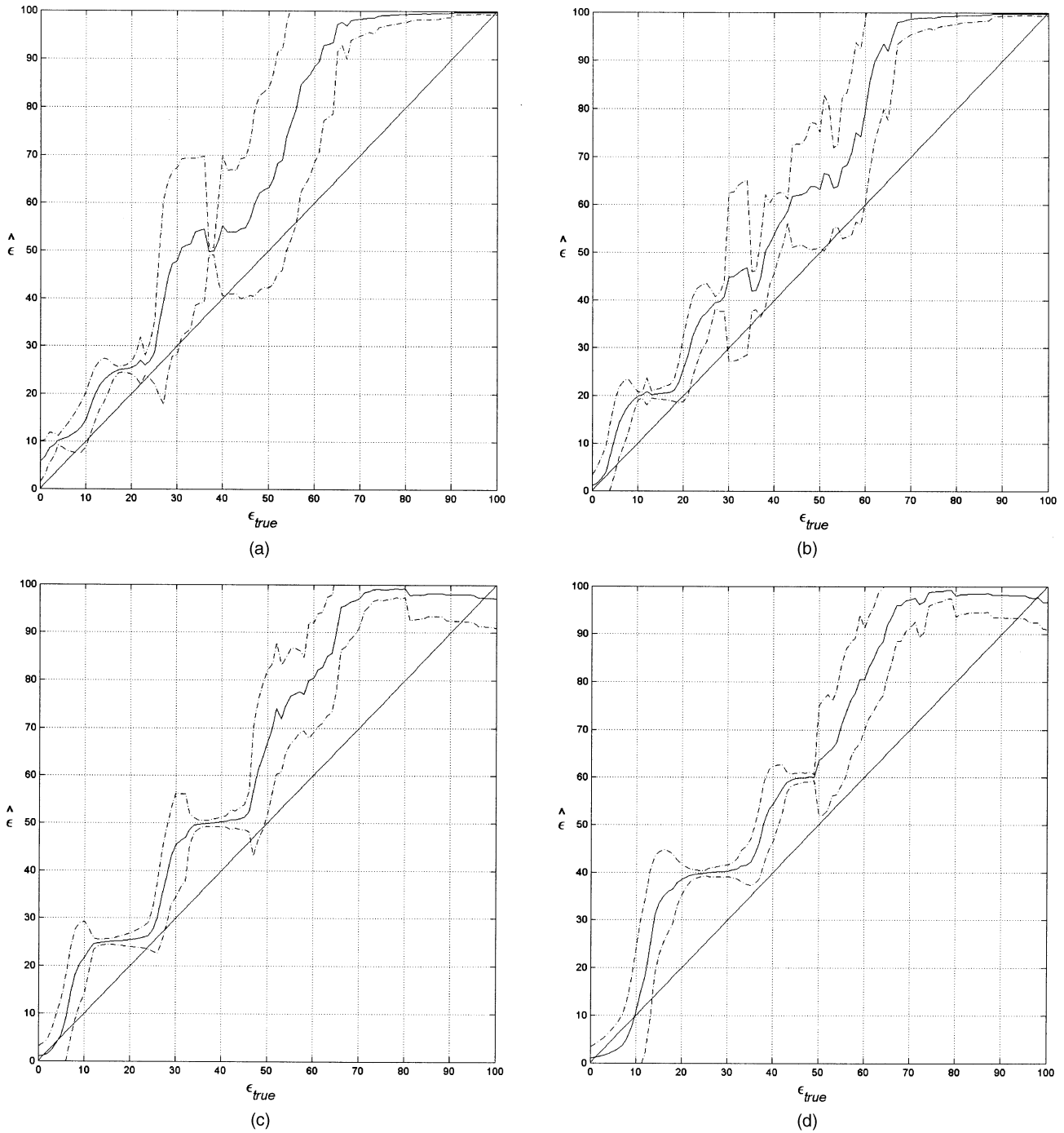


Fig. 5. Estimated versus true parameter: left stabilator failure. (a) Set: 10%, 25%, 50%. (b) Set: 20%, 40%, 60%. (c) Set: 25%, 50%, 75%. (d) Set: 40%, 60%, 80%.

Fig. 5 is a plot of the estimated versus true effectiveness parameters for a left stabilator failure, where the subplots (a)–(d) each use a different discretization set for the spawned filters. The *time-averaged mean* is plotted as a solid line, the *time-averaged mean* $\pm 1\sigma$ bounds are plotted as dash-dot lines symmetrically displaced about that mean, and another solid line represents $\hat{\epsilon} = \epsilon_{true}$. Because of the compact representation in Fig. 5, trends can be seen easily with one plot rather

than comparing hundreds of “probability plots” or “parameter estimate versus time plots.” A blending and uncertainty trade-off can be seen by noticing that in Fig. 5(b) the mean is blended well, but in Fig. 5(d) the mean flattens at each discretization level (except at 80%). In Fig. 5, no discretization set performs well for a true effectiveness greater than 70%. Thus, an effectiveness greater than 70% appears as a fully functional aircraft to the algorithm, or an actuator failure goes unnoticed until the effectiveness

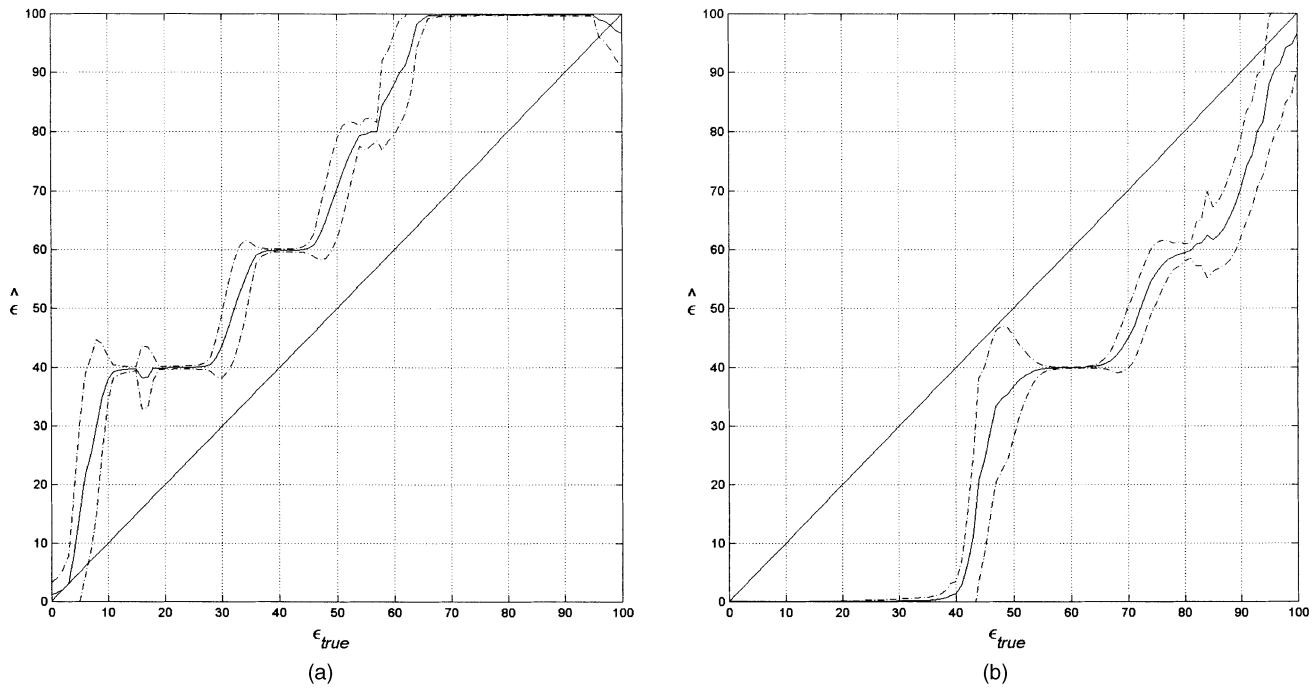


Fig. 6. Estimated versus true parameter. (a) Right flaperon. (b) Rudder.

is less than 70%. Improving performance for a true effectiveness greater than 70% was attempted by using discretization sets 40%, 60%, 90% and 70%, 80%, 90%. While still unsuccessful, the robust flight control system aboard the aircraft should accommodate this minor degradation in actuator effectiveness. Through the aid of the $\hat{\epsilon} = \epsilon_{\text{true}}$ line in Fig. 5 for all discretization sets ((a)–(d)), the effectiveness estimate is seen to be *larger* than the true effectiveness (except the fully functional and complete failure cases).

The parameter estimate versus true parameter plots allow a convenient comparison among the discretization sets. To decide which discretization set is “best,” consider that, for this research, a parameter *estimate* is sought for use in CR to maintain stability and control of the VISTA F-16 under partial and complete actuator failures. Further, if an actuator’s effectiveness is small, even if a good estimate of effectiveness could be obtained, the designer may choose *not* to send control authority to that actuator. With this in mind, discretization sets 3 and 4 are favored over sets 1 and 2 because they exhibit smaller $\pm 1\sigma$ bounds for most of the ϵ_{true} region. Set 4 is chosen over 3 because it drops off sharply in the true effectiveness interval of 10%–20%.

Fig. 6 plots the estimated versus true parameters for the right flaperon and rudder failures using the spawned filters in set 4 (40%, 60%, 80%). Only one side of the flaperons and stabilators is shown because the flaperons and stabilators produce similar results regardless of the side. Notice that the rudder behaves “opposite” the other actuators (physically, the rudder is “different” from the other surfaces).

C. Refined Parameter Estimate Plots

The estimate is *refined* before it is sent to the CR algorithm, based on the empirical relationship of the estimated versus true parameter, as seen in Fig. 5(d) or Fig. 6. Fig. 7 demonstrates this process for a left stabilator failure using spawned filters at 40%, 60%, and 80%. Subplot (a) repeats the $\hat{\epsilon}$ versus ϵ_{true} plot in Fig. 5(d). Subplot (b) shows the mapping $\epsilon_{\text{out}}(\hat{\epsilon})$ used in the MMAE with Filter Spawning to refine the estimate as follows (based only on the estimate $\hat{\epsilon}$ because ϵ_{true} is not known). 1) If the parameter estimate is less than 40%, then the refined estimate ϵ_{out} is considered a complete left stabilator failure estimate, assuring that actuators with small effectiveness values will be treated as complete failures in a redistributed control scheme. 2) If the parameter estimate is between 40% and 96%, then the estimate is refined by a piece-wise linear relationship to provide an ϵ_{out} more closely equal to ϵ_{true} . 3) If the parameter estimate is higher than 96%, then the refined estimate is considered a fully functional estimate so that control can be sent as usual. In words, given a parameter estimate $\hat{\epsilon}$, the refined estimate ϵ_{out} is determined and used by CR.

Fig. 7(c) shows the refined parameter estimate versus the true parameter. This subplot was formed using the refined parameter estimate given by the MMAE with Filter Spawning algorithm (incorporating the mapping given in Fig. 7(b)) at a time 2 s *after the failure was declared* based on the convergence trends seen in the parameter estimate versus time plot in Fig. 4. A single time point was chosen because it is desirable to form a redistributed control vector as soon

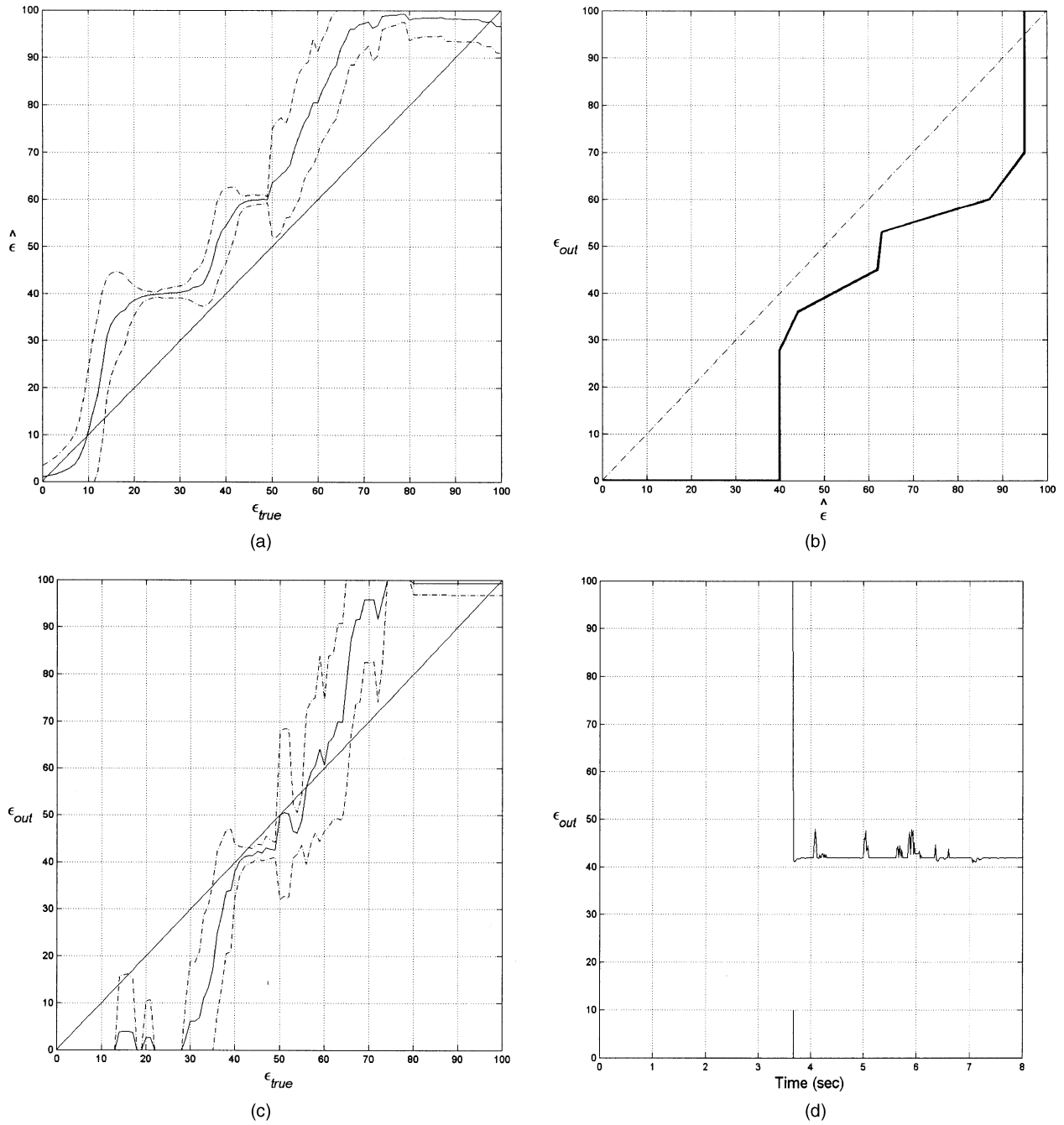


Fig. 7. Left stabilator failure. (a) $\hat{\epsilon}$ versus ϵ_{true} . (b) ϵ_{out} versus $\hat{\epsilon}$. (c) ϵ_{out} versus ϵ_{true} . (d) ϵ_{out} versus time.

as possible rather than wait online to time-average the refined estimate. The deviation between ϵ_{out} and ϵ_{true} in the range between 40% and 96% is due to the stochastic nature of the problem *and* the fact that ϵ_{out} is based on an empirical time-averaged estimate starting 2 s after detection. Thus, when $\hat{\epsilon}(t_i)$ does not match the time-averaged estimate $\hat{\epsilon}$, some error will exist in the ϵ_{out} estimate.

Because choosing a single point upon which to base the subplot Fig. 7(c) may bring doubt to the validity of the plot, a representative *single* Monte Carlo run at a single true effectiveness 50% is shown in Fig. 7(d). The time of failure

declaration in Fig. 7(d) is shown with a “tickmark” placed at the bottom of the plot at $t_i = 3.6$ s. The time at which the refined estimate is used (to be compared with the true effectiveness) is 2 s after this declaration. Fig. 7(d) shows that the refined parameter estimate remains quite consistent over time for this single representative Monte Carlo run. Fig. 8 shows the refined parameter estimate versus the true parameter for the right flaperon and rudder failures. Again, only one side of the flaperons and stabilators is shown because the flaperons and stabilators produce similar results regardless of the side.

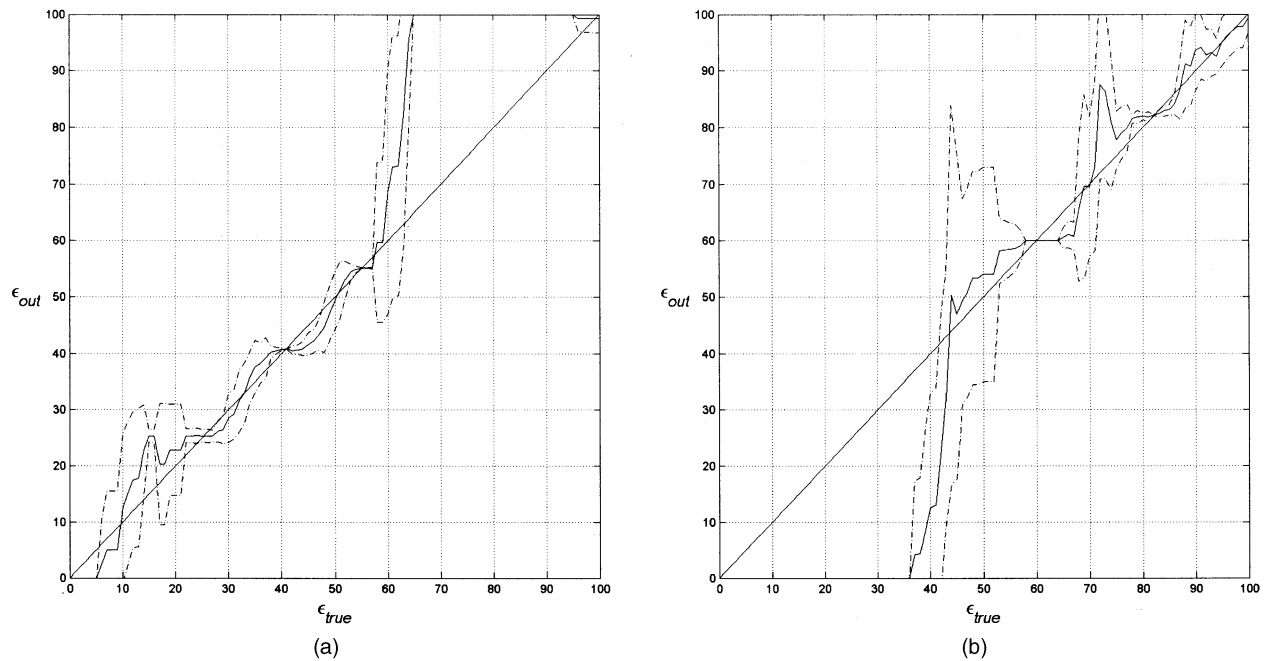


Fig. 8. Refined estimate versus true effectiveness. (a) Right flaperon. (b) Rudder.

D. Summary

The following procedure is accomplished *to design* the refined parameter estimate ϵ_{out} to be used for CR for flight control in the face of partial or complete actuator failures and full sensor failures, based on failure detection and estimation based on MMAE with Filter Spawning. 1) Choose a discretization set or sets for the spawned filters. 2) Calculate the conditional probabilities for each filter with the MMAE at each time sample for each of a set of true actuator effectiveness values (such as 0%, 1%, 2%, ..., 100%). 3) Using the conditional probabilities and the respective hypothesized parameter values, determine a blended parameter estimate at each time sample. 4) Using the time to convergence as seen in the parameter estimate versus time plots, time-average the parameter estimate mean value to obtain one steady state mean estimate for each true parameter value. 5) Refine the estimate to incorporate the relationship between the true parameter value and the parameter estimate.

The following procedure is accomplished *to implement* the refined parameter estimate ϵ_{out} in MMAE with Filter Spawning *at each time sample*.

- 1) Calculate the conditional probabilities for each filter.
- 2) Using the conditional probabilities and the respective hypothesized parameter values, determine a blended parameter estimate.
- 3) Refine the estimate.
- 4a) If detection has occurred less than two seconds prior to the current time, do not “trust” the effectiveness estimate yet and conservatively consider the actuator fully failed ($\hat{\epsilon}_{out} = 0\%$) in a CR scheme.
- 4b) If detection has occurred more than 2 s prior to

the current time, “trust” the effectiveness estimate and use it in a modified CR scheme to obtain the least degradation in performance as is possible.

VI. CONCLUSION

MMAE with Filter Spawning [6] has successfully estimated the failure effectiveness in the face of partial actuator failures. The performance of the refined estimates for each actuator failure in Fig. 7(c) and Fig. 8 demonstrates in a compact manner the ability of MMAE with Filter Spawning to obtain an accurate estimate of complete and partial actuator failures, as applied to the VISTA F-16. This excellent performance is in direct distinction to the inadequate performance of the same MMAE *without* filter spawning, as described in the beginning of Section IV. This research has concentrated on MMAE with Filter Spawning’s ability to *detect* and *estimate* complete and partial actuator failures (as well as complete sensor failures). To complete the application of MMAE with Filter Spawning, *control* should be applied using the refined estimate of the failure’s effectiveness. Such an investigation is currently being pursued. Stepaniak [21, 22] showed that, given the correct failure declaration, CR restored performance of an aircraft with failed actuators to nearly that of a fully functional aircraft.

Useful design tools were developed, allowing the designer to notice trends quickly, improve the parameter estimate, and give insight to how the estimate should be used in control redistribution. The designer has many options available for adjustment to enhance the refined estimate, such

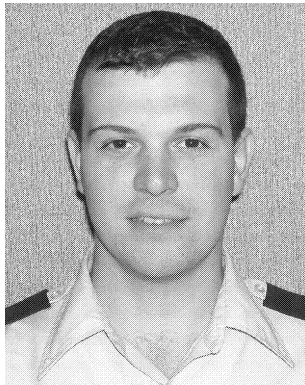
as the probability threshold for declaring a failure, the blending probability lower bound, using a bank without spawned filters when the spawned filters are not needed, alternatives to spawning filters (such as waiting until the conditional probability for an actuator failure hypothesis exceeds some threshold before spawning filters based upon that actuator), the mappings used to *refine* the estimate, and the time at which the *refined* estimate is given to a control scheme. Moreover, additional filters could be considered to determine *to what extent* performance is improved (to decide if the performance gained merits the additional computational loading accrued), and each actuator need not spawn filters with equally discretized effectiveness hypotheses.

REFERENCES

- [1] Athans, M., et al. (1977)
The stochastic control of the F-8C aircraft using a multiple model adaptive control (MMAC) method—Part I: Equilibrium Flight.
IEEE Transactions on Automatic Control, **AC-22**, 5 (Oct. 1977), 768–780.
- [2] Blom, H. A. P., and Bar-Shalom, Y. (1988)
The interacting multiple model algorithm for systems with Markovian switching coefficients.
IEEE Transactions on Automatic Control, **33**, 8 (Aug. 1988), 780–783.
- [3] Clark, C. S. (1997)
Multiple model adaptive estimation and control redistribution performance on the VISTA F-16 during partial actuator impairments.
MS Thesis, AFIT/GE/ENG/97D-23, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, Dec. 1997.
- [4] Eide, P. K. (1994)
Implementation and demonstration of a multiple model adaptive estimation failure detection system for the F-16.
MS Thesis, AFIT/GE/ENG/94D-06, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, Dec. 1994.
- [5] Eide, P. K., and Maybeck, P. S. (1996)
An MMAE failure detection system for the F-16.
IEEE Transactions on Aerospace and Electronic Systems, **32**, (3) (July 1996), 1125–1136.
- [6] Fisher, K. A. (1999)
Multiple model adaptive estimation using filter spawning.
MS Thesis, AFIT/GE/ENG/99M-09, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, Mar. 1999.
- [7] Fry, C. M., and Sage A. P. (1974)
On hierarchical structure adaption and system identification.
International Journal of Control, **20**, 3 (Sept. 1974), 433–452.
- [8] Gauvrit, M. (1984)
Bayesian adaptive filter for tracking with measurements of uncertain origin.
Automatica, **20** (Mar. 1984), 217–224.
- [9] Lewis, R. W. (1996)
Multiple model adaptive estimation and control redistribution for the VISTA F-16.
MS Thesis, AFIT/GE/ENG/96D-29, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, Dec. 1996.
- [10] Li, X. R., and Bar-Shalom, Y. (1996)
Multiple model estimation with variable structure.
IEEE Transactions on Automatic Control, **41**, 4 (Apr. 1996), 479–493.
- [11] Li, X. R., et al. (1997)
Multiple-model estimation with variable structure: Model-group switching algorithm.
In *Proceedings of the 36th IEEE Conference on Decision and Control*, San Diego, CA, (Dec. 1997), 3114–3119.
- [12] Magill, D. T. (1965)
Optimal adaptive estimation of sample stochastic processes.
IEEE Transactions on Automatic Control, **AC-10**, 4 (Oct. 1965), 434–439.
- [13] Maybeck, P. S. (1994)
Stochastic Models, Estimation, and Control, II.
New York: Academic Press, 1982. (Republished: Arlington, VA: Navtech, 1994.)
- [14] Maybeck, P. S., and Pogoda, D. L. (1989)
Multiple model adaptive controller for the STOL F-15 with sensor/actuator failures.
In *Proceedings of the 28th Conference on Decision and Control*, (Dec. 1989), 1566–1572.
- [15] Maybeck, P. S., and Stevens, R. D. (1991)
Reconfigurable flight control via multiple model adaptive control methods.
IEEE Transactions on Aerospace and Electronic Systems, **27**, 3 (May 1991), 470–480.
- [16] Maybeck, P. S., and Suizu, R. I. (1985)
Adaptive tracker field of view variation via multiple model filtering.
IEEE Transactions on Aerospace and Electronic Systems, **21**, 4 (July 1985), 529–537.
- [17] Menke, T. E. (1992)
Multiple model adaptive estimation applied to the VISTA F-16 with actuator and sensor failures.
MS Thesis, AFIT/GA/ENG/92J-01, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, June 1992.
- [18] Menke, T. E., and Maybeck, P. S. (1995)
Sensor/actuator failure detection in the VISTA F-16 by multiple model adaptive estimation.
IEEE Transactions on Aerospace and Electronic Systems, **31**, 4 (Oct. 1995), 1218–1229.
- [19] Pogoda, D. L. (1988)
Multiple model adaptive controller for the STOL F-15 with sensor/actuator failures.
MS Thesis, AFIT/GE/ENG/88D-23, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, Dec. 1988.
- [20] Schaefer, K. E. (1964)
Bioastronautics.
New York: Macmillan, 1964.
- [21] Stepaniak, M. J. (1995)
Multiple model adaptive control of the VISTA F-16.
MS Thesis, AFIT/GE/ENG/95D-04, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, Dec. 1995.
- [22] Stepaniak, M. J., and Maybeck, P. S. (1998)
MMAE-based control redistribution applied to the VISTA F-16.
IEEE Transactions on Aerospace and Electronic Systems, **34**, 4 (Oct. 1998), 1249–1260.

- [23] Stevens, R. D. (1989)
Characterization of a reconfigurable multiple model
adaptive controller using a STOL F-15 model.
MS Thesis, AFIT/GE/ENG/89D-52, School of
Engineering, Air Force Institute of Technology,
Wright-Patterson AFB, OH, Dec. 1989.

- [24] Zhang, Y. M., and Li, X. R. (1998)
Detection and diagnosis of sensor and actuator failures
using IMM estimation.
IEEE Transactions on Aerospace and Electronic Systems,
34, 4 (Oct. 1998), 1293–1312.



Kenneth A. Fisher was born in Pittsburgh, PA on August 3, 1975. In May 1997, he received the degree of Bachelor of Science in electrical engineering from Ohio Northern University. He received his commission as a Second Lieutenant through the Air Force ROTC program in May 1997. In March 1999, he received the degree of Masters of Science in electrical engineering from the Air Force Institute of Technology (AFIT).

From March 1999 through September 2001, he served as a Spectral Analyst in the Advanced Measurement and Signatures Intelligence (MASINT) Branch; MASINT Exploitation Division; Data Exploitation Directorate; Headquarters, National Air Intelligence Center (HQNAIC), Wright-Patterson AFB, OH. On 25 May 2001, Kenneth earned the rank of Captain in the USAF. Currently, he is pursuing a Doctorate of Philosophy in electrical engineering at the Air Force Institute of Technology.

Peter S. Maybeck (S'70—M'74—SM'84—F'87) was born in New York, NY on February 9, 1947. He received the B.S. and Ph.D. degrees in aeronautical and astronautical engineering from M.I.T., Cambridge, in 1968 and 1972, respectively.

In 1968, he was employed by the Apollo Digital Autopilot Group of The C. S. Draper Laboratory, Cambridge, MA. From 1972 to 1973, he served as a military control engineer for the Air Force Flight Dynamics Laboratory and then joined the faculty of the Air Force Institute of Technology in June of 1973. He is currently Professor of Electrical Engineering, responsible for the graduate sequence in estimation and stochastic control and for individual advanced digital filtering and control courses. Current research interests concentrate on using optimal estimation techniques for guidance systems, tracking, adaptive systems and failure detection purposes.

Dr. Maybeck is author of over one hundred papers on applied optimal filtering as well as the book, *Stochastic Models, Estimation and Control* (Academic Press, vol. 1—1979, vols. 2 and 3—1982; republished by Navtech in 1994). He is a member of Tau Beta Pi, Sigma Gamma Tau, Eta Kappa Nu, and Sigma Xi. He was recipient of the DeFlorez Award (ingenuity and competence of research), the James Means Prize (excellence in systems engineering) and the Hertz Foundation Fellowship at M.I.T. in 1968. In all years from 1975 to 2000, he received commendation as outstanding Professor of Electrical Engineering at A.F.I.T. In December of 1978, he received an award from the Affiliate Societies Council of Dayton as one of the twelve outstanding scientists in the Dayton, Ohio area. In March of 1980, he was presented with the Eta Kappa Nu Association's C. Holmes MacDonald Award, designating him as the outstanding electrical engineering professor in the United States under the age of 35 (he had placed second in this national competition for 1977 as well). In 1985, he received the Frederick Emmons Terman Award, the highest national award to a Professor of Electrical Engineering given by the American Society of Engineering Education. He is a member of the AIAA, and he is the current IEEE Dayton Section Student Activities Chairman and a member of the IEEE Executive Committee of Dayton, and he previously served as Chairman of the local Automatic Control Group.

