Surface Effectiveness Estimation for Control Reconfiguration of Impaired Aircraft

by

Derek Paul Laufenberg

S.B., Aeronautics and Astronautics, Massachusetts Institute of Technology
(1988)

SUBMITTED TO THE DEPARTMENT OF
AERONAUTICS AND ASTRONAUTICS
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
March 1989

Signature of Author__

Department of Aeronautics and Astronautics
March 2, 1989

Certified by__________

Professor Walter Hollister

Certified by____________

Dr. Tahm Sadeghi
[illegible] Company - ACSD

Certified by_____________

Professor Harold Y. Wachman
Chairman, Departmental Graduate Committee

Surface Effectiveness Estimation for Control Reconfiguration of Impaired Aircraft

by

Derek Paul Laufenberg

Submitted to the Department of Aeronautics and Astronautics
on March 10, 1989
In Partial Fulfillment of the Requirements
For the Degree of Master of Science

ABSTRACT

The problem of variations in control effectiveness caused by control surface damage was examined in this thesis. Three different filters were implemented to estimate remaining surface effectiveness after surface impairment. The first filter, an Extended Kalman Filter (EKF), was used as a parameter estimator for the surface effectiveness of a single surface. The second estimator, a Multiple Model Estimator (MME), was designed and compared to the Kalman Estimator. The MME was derived from simplifications made to an Adaptive Kalman Filter algorithm. The simplifications were done to facilitate real-time operation of the estimator. The third estimator, also an Extended Kalman Filter, was used to address the multiple damaged surface problem. A natural extension of the single surface EKF increases the capability, so that it can track damage to all of the aircraft surfaces simultaneously. Because the expanded filter monitors all of the surfaces, it can be used as an impairment detector. This is an improvement over the other two estimators which only monitor one surface and require an external impairment detection mechanism. The cost of this extension was found in the time required for the multiple surface EKF to converge. The addition of constraints to the multiple surface EKF was shown to reduce estimation time. Convergence properties and computational costs are presented for each of three estimators. These calculations show that the EKF is a computationally more efficient solution to both the single and multiple damaged surface problem.

Thesis Supervisor: Walter Hollister
Title: Professor of Aeronautics and Astronautics

Thesis Supervisor: Tahm Sadeghi, Manager
Title: Advanced Flight Control Engineering
General Electric Company - ACSD

Acknowledgments

I am fortunate to have a family that has always believed in me - even during those times that I did not. My mother taught me to be emotionally strong, inquisitive in nature and persistent at my undertakings. For this and her unconditional love and support, I thank her. I would also like to thank my grandmother for her emotional and financial support. Much of the credit for this work should go to my wife Pamela. She, most of all, helped me overcome my impediments and frustrations during the course of this work.

In addition to my family, there have been many other people who have been helpful. Mr. Chic Dittmar supported and guided me when I first started as a Co-op at the General Electric Company and later was a helpful sounding board for this thesis. There are other people at General Electric without whose help and guidance I would have been lost: Dr. Tahm Sadeghi, who, in addition to being my company thesis supervisor, provided me with insight and encouragement; Mr. Mark Stifel, who tolerated my questions and tangents while we discovered the mysteries of Kalman filtering; Mr. Paul Higgins, who was always there with a word of encouragement; and Mr. Donald Pogoda, a friend who provided technical advice and encouragement.

Help is gratefully acknowledged from the General Electric Company - Aircraft Control Systems Division for the use of its computation, simulation and other facilities.

Permission is granted to the General Electric Company, to reproduce and distribute copies of this thesis in whole or in part.

To Pamela

Table of Contents

List of Symbols and Abbreviations

a_y	Y body axis acceleration
a_z	Z body axis acceleration
$\mathbf{a}$	effectiveness parameter vector for MME
$\mathbf{a}_i$	discrete parameter vector for MME
a_i	discrete scalar effectiveness for MME
$\mathbf{B}$	surface control derivative matrix
$\mathbf{C}$	covariance matrix for normal probability
$\mathbf{e}$	estimate error vector
$\mathbf{e}^-$	*a priori* estimate error vector
$\mathbf{E}$	diagonal surface effectiveness matrix
$\mathbf{ez}$	residual acceleration vector (measured - modeled accelerations)
EKF	Extended Kalman Filter
$\mathbf{H}$	observation matrix
$\mathbf{H}_m$	measurement observation matrix
$\mathbf{H}_p$	process/model observation matrix
$\mathbf{K}$	Kalman gain matrix
MME	Multiple Model Estimator
$\mathbf{I}$	identity matrix
$\mathbf{\Phi}$	state transition matrix
$\dot{p}$	body axis roll acceleration
p()	probability density function
$\mathbf{P}$	error covariance matrix for Kalman filter
$\mathbf{P}^-$	*a priori* error covariance matrix
$\dot{q}$	body axis pitch acceleration
$\mathbf{Q}$	process noise covariance matrix
$\dot{r}$	body axis yaw acceleration
$\mathbf{R}$	measurement noise covariance matrix
$\mathbf{s}$	aircraft sensed acceleration vector
$\mathbf{u}$	aircraft control input vector
$\mathbf{v}$	measurement noise

$\mathbf{w}$	process or model noise
$\mathbf{x}$	EKF state estimate vector
$\hat{\mathbf{x}}$	EKF/MME parameter estimate
$\mathbf{z}_{measured}$	measured accelerations
$\mathbf{z}_{modeled}$	modeled accelerations
$\mathbf{z}^*$	set of all residual acceleration measurements

Chapter 1

Introduction

Perspective on Effector Estimation

1.1 Introduction

This chapter presents the rationale and motivation for research into the subject of real-time surface effectiveness estimation. Current flight control systems, while robust, do not offer the ability to deal with aircraft damage. Damage to control surfaces can result in a needless mission abortion or aircraft loss. Estimation of remaining surface effectiveness can be used by a suitable control system to maintain flyability and survivability of damaged aircraft.

For this study, an advanced fighter plane was used for the simulation tests. Such an aircraft would have a digital fly-by-wire flight control system (Command Augmentation System) with redundant control surfaces. The redundant surfaces can be used to compensate for battle damage and other impairments. Battle damage is taken to mean any change in the structure of the control surface; such as that which would result from gun fire.

Three different estimators will be presented in this thesis. The first estimator is an extended Kalman filter. The Extended Kalman Filter (EKF) will be used to estimate the impairment factor of a single damaged surface. For this estimator, it is assumed that the surface which is damaged is known *a priori*.

The Kalman filter optimally combines measurement data with modeling information to produce an estimate. In this case, the estimate produced is the surface effectiveness. The effectiveness is represented by a factor on the surface's control derivative. The factor will be in the range of zero to one; a value of one would represent a fully effective surface and zero would be a useless surface. The effectiveness factor is a parameter in the Kalman filter's system model. The estimator adjusts the model's effectiveness parameter to minimize the difference between the measured accelerations and those predicted by the filter's model. Once an estimate is obtained, it can be used by the flight control to compensate for the damage.

The second type of estimator is a Multiple Model Estimator (MME). The MME uses a number of different aircraft models to test for the effect of the damaged surface on the aircraft. As with the first method, the multi-model estimator assumes *a priori* knowledge of which surface is impaired.

Both the EKF and MME estimators are based on a single impairment assumption. Only one surface will be damaged at a time; however, sequential impairments to the same surface are possible.

The third estimator which will be investigated is an expanded form of the EKF. It is expanded so that each surface is estimated at the same time. With this expansion, the estimator can determine the surface effectiveness factors for each aircraft surface when there is simultaneous damage to aircraft surfaces. The expansion of the EKF estimator may be desirable for two reasons. First, it is possible that when one surface is damaged, another will be damaged as well. Second, there is no need for explicit *a priori* information about which surface is damaged, as all surfaces are considered at the same time.

An overview of a reconfigurable flight control system will be presented to demonstrate how surface effectiveness estimation can be combined with a reconfigurable controller to improve the damaged aircraft performance. Requirements for the speed and accuracy of the estimator, and the impairment model which was used for the experiment, are also described in this chapter.

1.2 Motivation for Treating the Problem

1.2.1 Increased Reliability

Reconfigurable flight control systems offer additional fault tolerance and performance advantages over conventional flight control systems. Simply stated, reconfigurable flight control systems redistribute the control commands in the event of a surface impairment. The redistributed commands compensate for surface impairments through the control effector redundancy inherent to the aircraft. Three basic types of impairments are considered for a functional reconfigurable flight control system: stuck surface, floating surface, and partial surface loss. Impairments can result from incidents such as battle damage, object strikes and actuator failure. A surface is considered stuck when the actuator cannot move the surface. A floating surface condition exists when the surface can no longer be controlled by the actuator and moves freely with the airflow. Stuck and floating surface fault detection is beyond the scope of this thesis. A partially damaged control surface is one which is missing some fraction of the surface.

A partial surface loss changes the control characteristics of the aircraft. The details of the impairment model for control surface damage will be explained in Section 1.5. There is a wide range of possible damage a surface can sustain. The surface loss can be minor, requiring little more than some additional deflection, or it can be a major impairment which requires deflection changes in all surfaces to maintain flight path. In both cases, a standard flight control will have difficulty maintaining aircraft stability and control.

Surface effectiveness estimation used for control redistribution provides compensation which will maintain aircraft stability and control. Redundancies inherent among the aircraft surfaces are exploited to increase the reliability of the aircraft in the event of a control surface impairment. Estimation is key in determining the proper surface deflections to command to both the damaged and undamaged surfaces. By fully utilizing all surfaces, including those with partial surface losses, aircraft control can be maintained. This can mean the difference between losing or landing the aircraft. This in itself is a major benefit of control estimation and redistribution.

1.2.2 Greater Mission Survivability

Closely related to reliability is the mission completion and success. This class of reconfigurable flight control system is applicable to both military and commercial aircraft. Redundancy of flight critical systems is a decisive factor in surviving surface damage or completing a mission in a hostile environment. Maintaining aircraft control and an operational flight envelope is a distinct advantage. Large degrees of damage can be withstood before the aircraft would have to be abandoned or the mission aborted.

Given the design of many of today's advanced fighters, there is exceptional surface redundancy inherent to the airframe. Independent stabilators and canards, collective and differential ailerons, and maneuvering flaps all have capabilities which can be exploited to compensate for damage. Independent use of canards, for example, compensates for loss of a primary pitch or roll control surface. The number of redundant surfaces available on military aircraft, combined with advanced digital flight control systems which can exploit inherent redundancy, will allow improvements in mission reliability and safety of flight [15,16].

1.3 Overview of a Reconfigurable Flight Control System

This section will describe a typical reconfigurable flight control system which would make use of surface effectiveness estimation. Discussion will cover each of the three main aspects of control reconfiguration: detection, estimation, and control restoration.

1.3.1 Failure Detection and Classification

The detection function analyzes linear and angular accelerations and rates, to determine if a failure has occurred. Once a failure is identified, this function then determines which aircraft surface is at fault as well as the nature of the impairment. If the failure is a stuck or a floating surface, control surface deflection will be reallocated to completely avoid the impaired surface. No estimation is thus required. For partial surface loss, however, it is important to determine the extent of the impairment. For the single surface impairment scenario, it will be assumed that the damaged surface is identified before the effectiveness estimation takes place. In the case of the expanded estimator, where all surfaces are estimated simultaneously, no such assumption will be made.

1.3.2 Estimation

There are a number of different ways to build a parameter estimator for surface effectiveness estimation. Each will have its own advantages and disadvantages. Surface effectiveness estimation is inherently a constrained estimation problem. The constraint is due to the assumption that the effect of the impairment is to reduce the effectiveness of the surface. The reduction factor is constrained to a range of zero to one. It will be shown empirically that proper use of this constraint will improve convergence time.

Issues of concern for the estimator are speed, accuracy and behavior of the estimator. The faster and the more accurate the estimates are, the better the control system will be. The accuracy of the estimate, for example, is a function of the signal to noise ratio of the aircraft's sensors, and a function of how well the aircraft dynamics are modeled. The non-linear dynamic model of the aircraft embedded in the estimator performs a crucial role in the fidelity of the estimator's convergence. If the aircraft model is not accurately modeled, the measurements taken from the aircraft sensors will cause a false alarm. The simulation work done in this thesis makes use of a non-linear aerodynamic database for an advanced fighter aircraft. The non-linear nature of the surface effectiveness requires the use of a non-linear estimator. The details involved in each of the three estimators tested for this thesis will be explained in Chapter 2; however, a brief survey is in order here.

The standard Kalman filter works well for linear systems. The class of filters known as Extended Kalman Filters (EKF) are approximations to non-linear systems. The

EKF updates the filter model with the current state estimate. The filter model is then linearized about the current state. References [3, 10] provide the details of both Kalman and extended Kalman filters. Reference [9] discusses the behavior of the Extended Kalman Filter as a parameter estimator.

Another form of non-linear Kalman estimator is the linearized Kalman filter, reference [3, 10]. The filter model for this estimator is linearized about a nominal trajectory. This method was not investigated because there is no way to know *a priori* what the nominal trajectory would be.

Another approach to this problem of on-line parameter estimation is multiple model Bayesian estimation. An approximation to the true solution of the Bayesian mean-square-error estimate is found from the probability density functions for the state estimates. The conditional probability density function for the measurement, and an *a priori* probability density function for the estimate are approximated in each computational cycle by employing statistical models and *a priori* assumptions. Then a new estimate is formed based on the mode (average) of the *a posteriori* probability of the state estimate conditioned on the measurement.

Since a system model is required for each possible element in the parameter space, this method is known as Multiple Model Estimation (MME). The Multi-model concept has been successfully developed for several applications. One notable application of the multi-model structure is the adaptive control system for NASA's F-8C, [1]. The origins of adaptive parameter estimation appear to come from Magill, [11].

These two different approaches will be tested and evaluated in this thesis. Both the multi-model and the extended Kalman filter estimators will be applied to the single surface impairment problem of assessing the remaining control authority of a partially damaged surface. The effectiveness estimate will then be used for the purpose of control reconfiguration. An expanded form of the first method, EKF, will then be developed to demonstrate how a system which monitors each surface could deal with simultaneous impairments to different surfaces.

1.3.3 Restoring Control

The final step in the reconfiguration process is to redistribute the surface commands to counter the impairment. This is accomplished by nulling the acceleration errors caused by the damage to the control surface. To properly understand how this is to be accomplished, one must first look at the system under control. Equations (1-1) and (1-2) describe the linearized system dynamics. An impairment to a control surface will cause a change in the control matrix, **B**. It is assumed that the stability matrix, **A**, remains unchanged by the impairment.

$$\dot{x} = Ax + Bu \tag{1-1}$$

$$y = Cx + Du \tag{1-2}$$

After the impairment, Equation (1-1) can be restated with a new control matrix, **B'**. An error vector for the accelerations can be calculated by subtracting Equation (1-3) from Equation (1-1) as shown in Equation (1-4).

$$\dot{x}_{impaired} = Ax + B'u' \tag{1-3}$$

$$error = Bu - B'u' \tag{1-4}$$

A new control vector, u', is sought for the damaged system which will minimize the error vector shown in Equation (1-4). In general, this is done by finding the minimum normal solution to:

$$B'u' = Bu \tag{1-5}$$

Where **B** and **B'** are, respectively, the unimpaired and impaired control effectiveness matrices. Physically, **Bu** represents the unimpaired commanded accelerations caused by the commanded surface deflection vector u, and **B'u'** represents the resulting impaired commanded accelerations. The control redistribution function executes a Penrose pseudo inverse algorithm, [14], to compute the impaired surface deflection, u'. The solution found by taking the Penrose inverse of **B'** has the form:

$$\underline{u}' = B'^{\#}B\underline{u} \qquad \text{\# denotes Penrose inverse} \tag{1-6}$$

The Penrose inverse is required because **B'** is typically not a square matrix and the solution is assumed to be under determined. This assumption is based on the premise that there are more surfaces than degrees of freedom being controlled. Control effector estimation is used to provide a flight control system with information about surface effectiveness. From this discussion, we can see the need for estimating the impaired control effectiveness matrix, **B'**, which directs the reconfiguration.

1.4 Estimation Requirements

As discussed in Section 1.3.2, the speed and accuracy of the estimator are important parameters to be investigated. A slow rate of convergence could waste valuable real-time required to save the aircraft. In some cases, there are only a few seconds before the plane departs to a point where it is no longer recoverable [16]. Inaccuracy in the estimates can lead to large reconfiguration transients within the aircraft control.

1.4.1 Speed of Convergence

The filter for this application requires quick identification of the impairment factor. Speed is crucial because the estimate of the surface effectiveness factor is needed to compensate for the damage done to the aircraft's surface. Quickness, however, needs to be defined more precisely. The driving factors determining the speed of convergence are the complexity of aircraft model, extent of surface damage, which surface is impaired, and the type of flight control system.

The type of aircraft is important because different airframes respond more quickly to the impairment than others. For example, when a primary longitudinal effector is impaired on an aircraft with relaxed longitudinal static stability, the aircraft would depart faster than an aircraft with the same impairment but with more static stablility.

Obviously, this is not as critical if a lateral-directional control effector is similarly damaged.

The amount of surface damage sustained can change the amount of time available before the aircraft is lost. A small loss of 5% of the surface may not be significant, and the Command or Stability Augmentation System (CAS or SAS) may cover the impairment adequately. However, for the more severe cases, the CAS will not have enough information to compensate for the damage. Further, the CAS does not have the ability to maximize the aircraft's performance capabilities, while a reconfigurable control system does. A conservative upper bound on the allowable estimation time would be the time it takes to lose control of the aircraft for a 100% loss to a primary longitudinal control surface.

Reference [16] discusses a number of different accidents which were deemed recoverable if the pilot had responded in time. The time available ranged from as little as five seconds to as much as twenty-five seconds. Being conservative, a maximum estimation time of one second would allow four seconds for the controls to be redistributed to the remaining surfaces. The aircraft used for this study was a stable aircraft. Five seconds may be too slow for some unstable aircraft where there may be only a few tenths of a second for the plane to be recovered. A maximum estimation time which was used as a goal in this thesis was two tenths of a second.

In the case where there is plenty of time to save the aircraft, the effect of a longer estimation time will be seen as a larger reconfiguration transient. Once the reconfigured state is reached, however, there should be no further transients. The transient occurs because the nominal flight control (CAS) attempts to compensate for the damage. When CAS compensation for an impairment is inadequate, but the reconfiguration system has not assessed the extent of the damage, the CAS will attempt to maintain aircraft flight path. The CAS states will command larger deflections to compensate for the impairment. When the reconfiguration system finally does take over, the CAS commands are too large and they imperfectly compensate for the fault. The commands are redistributed, and the CAS returns to performing the role it had before the impairment. The transient results from allowing the CAS too much time to improperly attempt to compensate for the damage.

1.4.2 Accuracy of Effector Estimates

The accuracy required for proper control redistribution must be understood in terms of the CAS and reconfiguration system interaction. When a Command or Stability Augmentation System is designed, there is always the problem of imperfect knowledge of control and stability derivatives. Thus these systems are designed to be very robust and tolerant of errors in the derivatives. The reconfiguration system acts to return the aircraft's damaged response within the tolerance of the C/SAS. The C/SAS acts as a fine tuning mechanism while the reconfiguration system is the coarse tuning. The acceptable tolerance for the estimate is plus or minus five percent ($\pm 5\%$) of the total range of the estimate.

1.5 Impairment Model and Surface Effectiveness Factors

In order to make the estimation problem more tractable, a number of simplifying assumptions have been made for the effects of a partial surface loss. At this point there is one predominant assumption which needs to be stated: impairment of a surface affects all axes by the same amount. This is done as a convenience to reduce the number of elements to be estimated. A more detailed impairment model would have independent estimates for each axis, but the added detail would greatly increase the complexity of the estimator.

This assumption leads to the idea of a single surface effectiveness parameter for each surface. The surface effectiveness factor is a non-dimensional term which models the effect of damage on a surface's ability to produce aerodynamic forces and moments. The parameter is a number between 0.0 and 1.0 (inclusive) which represents the surface's effectiveness. A value of 1 means the surface is unimpaired, while, at the other end of the spectrum, a zero states the surface is useless. This parameter is used in the **B** matrix on the nominal control derivatives. Thus the equation for the impairment model adopts the following form:

$$\mathbf{B}' = \mathbf{B}\mathbf{E}$$

Where **B** and **B'** are as previously described, and **E** is a non-dimensional *diagonal* matrix of effectiveness factors. There is one factor for each surface. Unimpaired surfaces will have unity values.

The surface effectiveness factor is a useful modeling device for simplifying the effects of damage on the surface control derivatives. The unpredictable effect of surface damage is a very complex problem and beyond the scope of this thesis. However, a more complex and accurate model, if developed, could be used in place of the simple effectiveness factor model in each of the three estimators that will be presented.

Chapter 2

Methods of Estimation

2.1 Introduction

This chapter will present the theory and mathematical descriptions of each of the three estimators studied in this thesis. Emphasis will first be placed on the generalized equations of the estimator. Then the focus will be narrowed to the problem of surface effectiveness estimation. This chapter will begin with an explanation of the basic linear Kalman filter.

2.2 Kalman Filter Estimator

2.2.1 Brief Historical Note on the Linear Kalman Filter

In 1960, R. E. Kalman changed the way linear filtering problems were used. Kalman reformulated the least-squares filtering problem using state space methods. This new formulation used vector modeling of the random processes and recursive data processing of the noisy measurements. Unlike most filtering problems, the Kalman filter was designed originally as a discrete time filter. The equations are independent and are not approximations to a continuous filter. In fact, it was not until about a year later when Kalman co-authored a paper with R. S. Bucy to present the continuous filter solution. The derivation of the continuous Kalman filter from the original discrete filter is found in Reference [3].

2.2.2 Theory of Operation for the Linear Kalman Filter

We will begin by assuming the form of a random linear vector process [3,6] can be described as follows:

$$\mathbf{x}_{k+1} = \mathbf{\Phi}_k \mathbf{x}_k + \mathbf{w}_k \qquad (2.2\text{-}1)$$

The discrete time state vector $\mathbf{x}$ is transformed at each time step by the state transition matrix $\mathbf{\Phi}$. The Gaussian white noise vector, $\mathbf{w}$, is an uncorrelated sequence with a known covariance structure. The noise is then added directly to the state.

The measurements are related to the state by a linear observation matrix, $\mathbf{H}_k$. The observation matrix represents the ideal connection between the state vector and the measuring device. Noise is again directly added to the measurement. The resulting equation is shown below:

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k. \qquad (2.2\text{-}2)$$

Where $\mathbf{z}$ is the set of measurements, $\mathbf{x}$ is the state process vector and $\mathbf{v}$ is the measurement noise vector.

It will be assumed that the two noise vectors $\mathbf{w}$ and $\mathbf{v}$ have known statistics; i.e., covariance structure. Further, it is also assumed that they are uncorrelated with each other. This last assumption is needed to make the calculation of the estimate covariance more tractable. The covariances for each of the noise terms can be summarized as follows:

$$E\{\mathbf{w}_k \mathbf{w}_j\} = \begin{array}{ll} \mathbf{Q}_k & k=j \\ 0 & k \neq j \end{array} \qquad (2.2\text{-}3)$$

$$E\{\mathbf{v}_k \mathbf{v}_j\} = \begin{array}{ll} \mathbf{R}_k & k=j \\ 0 & k \neq j \end{array} \qquad (2.2\text{-}3)$$

$$E\{\mathbf{w}_k \mathbf{v}_j\} = 0. \qquad \text{all } j,k \qquad (2.2\text{-}3)$$

Where the notation E{} represents the expected value or average function operation. Note that it is proper to claim that $\mathbf{Q}$ and $\mathbf{R}$ are covariance matrices because the noise processes are assumed to be zero mean. Given these assumptions, we are now ready to look at the derivation of the filter.

At this point it is convenient to introduce a notation that will assist us in looking at the filter equations. Assume for the moment that we have a value for the estimate $\hat{\mathbf{x}}_k^-$. The "hat" implies that this is the estimated quantity. The superscript minus sign refers to the value before incorporating the measurement information at time t_k.

First an error term is defined for the estimate, $\hat{\mathbf{x}}_k^-$. The estimation error vector, $\mathbf{e}_k^-$, is defined as the difference between the true state, $\mathbf{x}_k$, and the estimate $\hat{\mathbf{x}}_k^-$. It is the goal of the filter to minimize the error:

$$\mathbf{e}_k^- = \mathbf{x}_k - \hat{\mathbf{x}}_k^- \qquad (2.2\text{-}6)$$

The covariance matrix of the *a priori* error vector can also be defined as follows:

$$\mathbf{P}_k^- = E\{\mathbf{e}_k^- \mathbf{e}_k^{-T}\} = E\{(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T\} \qquad (2.2\text{-}7)$$

The desired minimum mean-squared error estimate will be the one for which the *a posteriori* error covariance matrix is minimized. The *a posteriori* error covariance matrix represents the estimate's covariance <u>after</u> the current measurement information is incorporated. The *a posteriori* covariance matrix is defined similar to Equation (2.2-7);

$$\mathbf{P}_k = E\{\mathbf{e}_k \mathbf{e}_k^T\} = E\{(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T\} \qquad (2.2\text{-}8)$$

To minimize the *a posteriori* error covariance, we start with the prior estimate of state, $\hat{\mathbf{x}}_k^-$, and combine it with the current measurement, $\mathbf{z}_k$, to produce an improved estimate. A linear combination of $\hat{\mathbf{x}}_k^-$ and $\mathbf{z}_k$ is selected to improve the estimate. Equation (2.2-9) describes the linear relation.

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-) \qquad (2.2\text{-}9)$$

The only remaining task for the filter is to find the optimal value for $\mathbf{K}$. The optimal $\mathbf{K}$ will minimize the mean-squared error of the estimate. That is the same as minimizing the trace of the covariance matrix in Equation (2.2-8).

Starting with the Equation (2.2-8) for the *a posteriori* error covariance, one can substitute Equation (2.2-9) for $\mathbf{x}_k$ and Equation (2.2-2) for $\mathbf{z}_k$ to obtain the following result:

$$\mathbf{P}_k = E\{[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) - \mathbf{K}_k(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-)] \\ [(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) - \mathbf{K}_k(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-)]^T\} \qquad (2.2\text{-}10)$$

Performing the required expectation on this rather unwieldy equation leaves the somewhat simpler form shown in Equation (2.2-11).

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{R}_k\mathbf{K}_k\mathbf{R}_k^T \qquad (2.2\text{-}11)$$

This is a general equation for the error covariance matrix. It can be used for any K_k, optimal or otherwise. The gain matrix can now be found. The objective is to find K by minimizing the trace of P_k. The solution to the optimal gain is that K which minimizes the trace of P_k. This can be done by the method of completing the squares, [3], or by matrix differentiation.

Differentiating the trace of P_k, in Equation (2.2-11), with respect to the matrix K and setting the result to zero, leads to the familiar Kalman gain Equation (2.2-12). This particular value for K will minimize P_k.

$$K_k = P_k^- H_k (H_k P_k^- H_k^T + R_k)^{-1} \qquad (2.2\text{-}12)$$

It is common to simplify the expression for the covariance matrix associated with the <u>optimal</u> estimate by substituting the expression for the optimal gain into the general form for the covariance matrix. This yields the following equation for the optimal P_k:

$$P_k = (I - K_k H_k) P_k^- \qquad (2.2\text{-}13)$$

It is important to remember that expression (2.2-13) is only valid for the estimate found when using the optimal gain K.

Each of the three key Equations, (2.2-9), (2.2-12), and (2.2-13) require *a priori* values for $\hat{x}_k^-$ and P_k^-. To get these quantities we must project ahead in time the current values $\hat{x}_k$ and P_k. The state transition matrix can be used to do this for $\hat{\mathbf{x}}_{k+1}^-$ and $\mathbf{e}_{k+1}^-$. Then $\mathbf{P}_{k+1}^-$ can be found as before from the error $\mathbf{e}_{k+1}^-$.

First project $\hat{\mathbf{x}}_k$ to $\hat{\mathbf{x}}_{k+1}^-$:

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{\Phi}_k \hat{\mathbf{x}}_k \tag{2.2-14a}$$

Next project the error vector:

$$\begin{aligned} \mathbf{e}_{k+1}^- &= \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^- \\ &= (\mathbf{\Phi}_k \mathbf{x}_k + \mathbf{w}_k) - \mathbf{\Phi}_k \hat{\mathbf{x}}_k \\ &= \mathbf{\Phi}_k \mathbf{e}_k + \mathbf{w}_k \end{aligned} \tag{2.2-14b}$$

Using the fact that $\mathbf{w}_k$ and $\mathbf{e}_k$ are uncorrelated, $\mathbf{P}_{k+1}^-$ can be found by taking the expected value of the vector outer product of $\mathbf{e}_{k+1}^-$ with itself. This yields equation (2.2-15).

$$E\{\mathbf{e}_{k+1}^- \mathbf{e}_{k+1}^{-T}\} = \mathbf{P}_{k+1}^- = \mathbf{\Phi}_k \mathbf{P}_k^- \mathbf{\Phi}_k^T + \mathbf{Q}_k \tag{2.2-15}$$

The five equations which form the complete linear Kalman filter are summarized by Equation (2.2-16). These equations form an iterative process for updating the state measurements. Initial conditions for the estimate and the error covariance matrix are, of course, needed to start the process. Figure (2.2-1) displays the initialization and the looping of the filter process.

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} && \text{(a)} \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) && \text{(b)} \\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- && \text{(c)} \\ \hat{\mathbf{x}}_{k+1}^- &= \mathbf{\Phi}_k \hat{\mathbf{x}}_k && \text{(d)} \\ \mathbf{P}_{k+1}^- &= \mathbf{\Phi}_k \mathbf{P}_k^- \mathbf{I}_k^T + \mathbf{Q}_k && \text{(e)} \end{aligned} \tag{2.2-16}$$

The filter process will incorporate the measurements and form the optimal weighted least-squares estimate. However, for the problem of surface effectiveness estimation, the observation matrix, **H**, is a function of the surface effectiveness estimate. Thus this estimator requires a slightly different filter approach. This approach is covered in detail in Section 2.3.

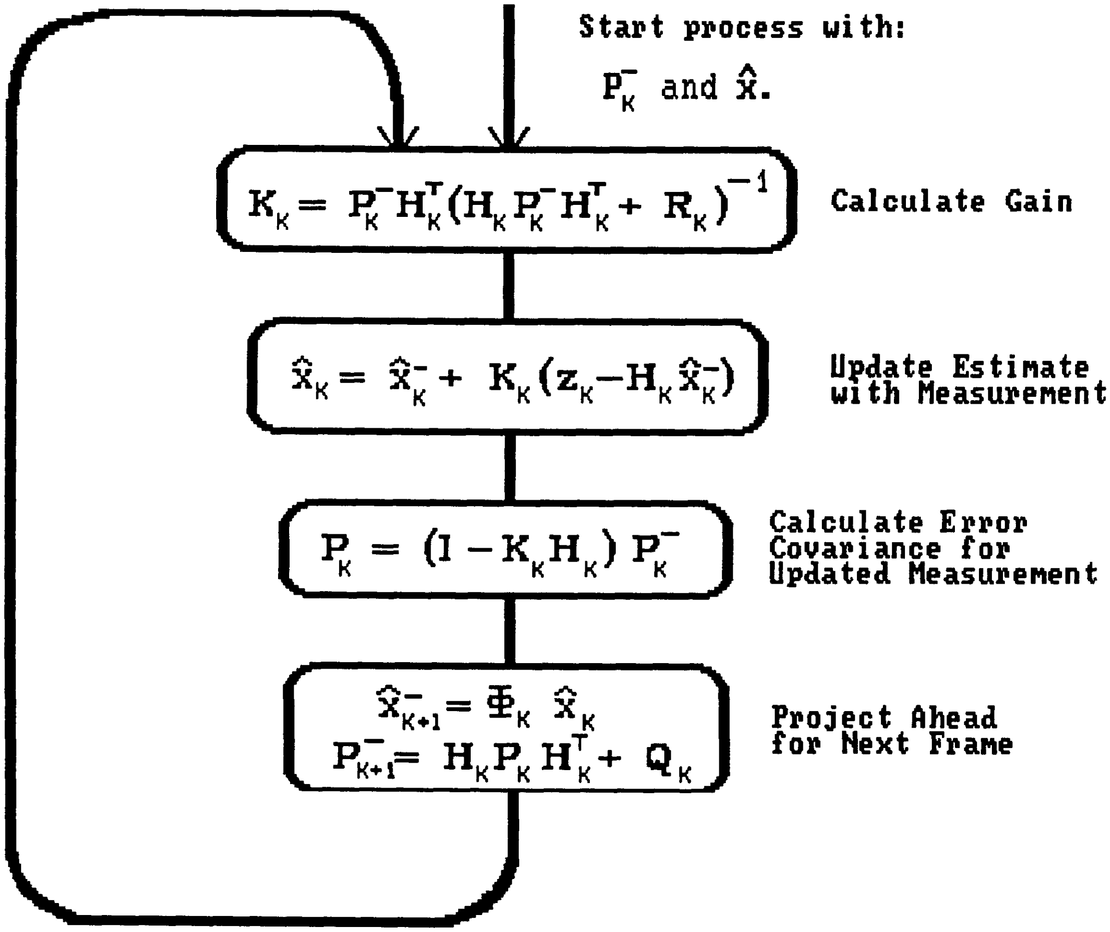


Figure 2.2-1 Kalman Filter Process

2.3 Extended Kalman Filter Estimator

The extended Kalman filter is needed when the system dynamics are a function of the state estimate.

2.3.1 Theory of Operation

The system equations (2.3-1) and (2.3-2) describe how the plant dynamics evolve in time. The process must be linearized and put into the same form as the set of Equations (2.2-16, a-e). This is done by linearizing about the current state estimate.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u},t) + \mathbf{w}(t) \tag{2.3-1}$$

$$\mathbf{z} = \mathbf{h}(\mathbf{x},\mathbf{u},t) + \mathbf{v}(t) \tag{2.3-2}$$

Linearization is accomplished by taking the partial derivatives of both $\mathbf{f}$ and $\mathbf{h}$ with respect to the state $\mathbf{x}$ and evaluating at a given flight condition. Then the first order Taylor series approximation is formed about the current estimate, $\hat{\mathbf{x}}_k$. These approximations are used to propagate the current state estimate and its associated covariance matrix. The propagation equations are given without derivation by Equations (2.3-3) and (2.3-4). The derivation is found in reference [6]. It is important to note that these equations are approximations to the true conditional mean equations.

$$\dot{\hat{\mathbf{x}}} = \mathbf{f}(\hat{\mathbf{x}}(t),\mathbf{u}), \quad t_{k-1} < t < t_k \tag{2.3-3}$$

$$\dot{\mathbf{P}}(t) = \mathbf{F}(\hat{\mathbf{x}}(t),\mathbf{u},t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}(\hat{\mathbf{x}}(t),\mathbf{u},t)^T + \mathbf{Q}(t) \tag{2.3-4}$$

Equation (2.3-5) defines the matrix $\mathbf{F}(\mathbf{x}(t),t)$. It is the Jacobian of $\mathbf{f}$ with respect to $\mathbf{x}$ evaluated at the current state estimate, $\mathbf{x} = \hat{\mathbf{x}}_k$.

$$\mathbf{F}(\mathbf{x}(t),\mathbf{u},t) = \frac{d\mathbf{f}(\mathbf{x}(t),\mathbf{u},t)}{d\mathbf{x}} \Bigg|_{\mathbf{x}(t) = \hat{\mathbf{x}}_k^-} \tag{2.3-5}$$

The estimation equations retain the same form as the linear Kalman filter (Equations 2.2-16, a-c), with the exception of the observation matrix, $\mathbf{H}$. $\mathbf{H}$ is a nonlinear function of the estimate as shown by Equation (2.3-6). The linearazation of $\mathbf{h}$ is done in the same fashion as $\mathbf{f}$. Thus the observation matrix used is a linear approximation to the nonlinear function.

$$\mathbf{H}(\mathbf{x}(t),\mathbf{u}(t),t) = \frac{d\mathbf{h}(\mathbf{x}(t),\mathbf{u}(t),t)}{d\mathbf{x}} \Bigg|_{\mathbf{x}(t) = \hat{\mathbf{x}}_k^-} \tag{2.3-6}$$

Since $\mathbf{H}$ is a function of the current estimate and the measurements, the Kalman gain and the covariance matrix sequences can no longer be pre-computed as they could be for the linear case. The complex expression for $\mathbf{H}$ shown in Equation (2.3-6) is used in the Kalman filter equations derived in Section 2.2. It should be stated that the Kalman gain and the error covariance become random variables themselves. They depend on the time history of $\hat{\mathbf{x}}$ which is a random variable. The most important effect of this dependency is that the accuracy of the estimate becomes trajectory dependent. We will see in Section 2.3.3 that for some trajectories there is not enough information in the measurement to properly improve the estimate.

The inverse required by the Kalman Filter equations is guaranteed to exist because both **R** and **P** are symmetric positive definite matrices. The term $(\mathbf{HPH}^T + \mathbf{R})$ is therefore a symmetric positive definite matrix which is invertible.

2.3.2 Specifics of Surface Effectiveness Estimation

Returning to the problem of surface effectiveness estimation, we find that there are a number of simplifications that can be made to the equations presented in Section 2.3.1. This is of great use since the goal of this project is to build a real time estimator which can be used on current flight control computer systems. The origin of each of the variables in the system and physically what each represents will be discussed next.

The filter state vector, $\hat{\mathbf{x}}$, reduces to a scalar because only a single surface impairment is estimated. As we will see in Section 2.5, when there are more surfaces, the full vector equations will be used. As stated before, the value of $\hat{x}$ is in the range of zero to one. This is a dimensionless parameter.

The observation matrix $\mathbf{H}(\hat{x})$ relates the scalar surface effectiveness parameter, $\hat{x}$, to the acceleration vector. The acceleration vector represents the accelerations caused by the impaired surface. That is, only the fraction of the total acceleration which is caused by the impaired surface being estimated. The remaining portion of the aircraft accelerations are subtracted from the measurement vector. The measurement vector, **z**, is then the collection of acceleration measurements taken from the aircraft sensors with modeled accelerations removed. Equations (2.3-7) through (2.3-12) should clarify the matter.

$$\hat{\mathbf{x}} = [\ x_1\ \ x_2\ \ x_3\ \ \ldots\ \ x_n\ ] \quad \text{surface parameter vector} \tag{2.3-7}$$

$$H = \begin{bmatrix} \frac{d\dot{P}}{d\mathbf{x}_1} & \frac{d\dot{P}}{d\mathbf{x}_2} & \cdots & \frac{d\dot{P}}{d\mathbf{x}_n} \\ \frac{d\dot{Q}}{d\mathbf{x}_1} & \frac{d\dot{Q}}{d\mathbf{x}_2} & \cdots & \frac{d\dot{Q}}{d\mathbf{x}_n} \\ \frac{d\dot{R}}{d\mathbf{x}_1} & \frac{d\dot{R}}{d\mathbf{x}_2} & \cdots & \frac{d\dot{R}}{d\mathbf{x}_n} \\ \frac{dAy}{d\mathbf{x}_1} & \frac{dAy}{d\mathbf{x}_2} & \cdots & \frac{dAy}{d\mathbf{x}_n} \\ \frac{dAz}{d\mathbf{x}_1} & \frac{dAz}{d\mathbf{x}_2} & \cdots & \frac{dAz}{d\mathbf{x}_n} \end{bmatrix} \tag{2.3-8}$$

$$\mathbf{z}_{surface} = \mathbf{H}\hat{\mathbf{x}} \quad \text{Estimated acceleration due to surfaces} \tag{2.3-9}$$

$$\mathbf{z}_{measured} = [\text{aircraft acceleration measurements}] = [\ \dot{p}\ \ \dot{q}\ \ \dot{r}\ \ a_y\ \ a_z\ ] \tag{2.3-10}$$

$$\mathbf{z}_{modeled} = [\text{aircraft accelerations without surfaces}] = [\ \dot{p}_{modeled}\ \ \dot{q}_{modeled}\ \ \dot{r}_{modeled}\ \ a_{y_{modeled}}\ \ a_{z_{modeled}}\ ] \quad (2.3\text{-}11)$$

$$\mathbf{z} = \mathbf{z}_{measured} - \mathbf{z}_{modeled} \quad (2.3\text{-}12)$$

$$\mathbf{ez} = \mathbf{z} - \mathbf{H}\hat{\mathbf{x}} \quad (2.3\text{-}13)$$

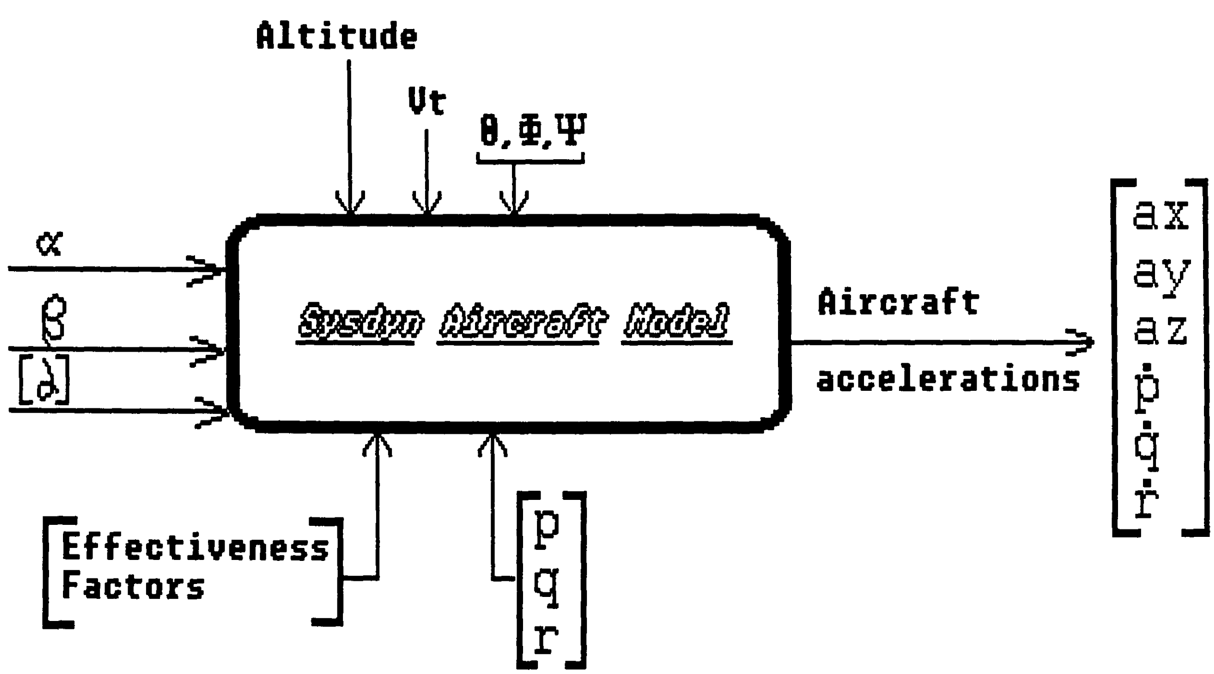
Equation (2.3-12) corresponds to the measurement vector which is used in the Kalman filter equations derived in Section 2.2. The organization of this estimator differs from the norm. Normally, one can think of a Kalman filter as a combination of two parts: the measurement system and the model system. The filter optimally combines the measurements with the modeled values to produce the estimate. In this application, the measurement portion contains a very extensive model of the aircraft dynamics and aerodynamics.

Through an algebraic manipulation of Equation (2.3-13), as shown below, the filter's error vector, **ez**, reduces to measured total accelerations minus the modeled total aircraft accelerations. This is important because it permits a self-contained aircraft model to be used in place of using the equations for the aircraft model as it was conceptually presented by Equations (2.3-7) through (2.3-13). Equation (2.3-14) demonstrates this manipulation.

$$
\begin{aligned}
\mathbf{ez} &= \mathbf{z} - \mathbf{H}\hat{\mathbf{x}} \\
&= \mathbf{z}_{measured} - \mathbf{z}_{modeled} - \mathbf{H}\hat{\mathbf{x}} \\
&= \mathbf{z}_{measured} - (\mathbf{z}_{modeled} + \mathbf{H}\hat{\mathbf{x}}) \\
&= \mathbf{z}_{measured} - \mathbf{f}(\hat{\mathbf{x}},\mathbf{u}) \\
&= \text{(measured accelerations)} - \text{(modeled accelerations)}
\end{aligned} \qquad (2.3\text{-}14)
$$

In our application, the measured accelerations are provided by the aircraft instrumentation package. The modeled accelerations are obtained from General Electric - ACSD's SYSDYN aircraft model. The SYSDYN module contains the non-linear aircraft dynamics as a function of measureable aircraft state. SYSDYN is short for system dynamic model. The diagram below shows SYSDYN's input and output variables.

Figure 2.3-1 SYSDYN Aircraft Model



SYSDYN is a mathematical aircraft model which contains both the dynamic and aerodynamic information. SYSDYN uses state information such as airspeed, altitude, surface deflections and other aircraft state information to calculate the expected aircraft accelerations.

2.3.3 Singularities in Observation Matrix

As well as calculating the modeled accelerations, SYSDYN calculates the observation matrix, **H**. The observation matrix relates the surface effectiveness parameter to the accelerations produced by the surface. At certain angles of attack and surface deflection, the surface produces little or no lift. When no lift is being produced, it is impossible to determine how effective the surface is. There is no information in the measurement signal during these times. Equation 2.2-12 shows that when **H** is identically zero, the gain will also be zero, causing spikes in the surface effectiveness estimate.

It was found that enforcing a lower limit on the elements of **H** was not the most desireable way to remove the transition spikes. The measurement covariance matrix, **R**, turned out to have values which were uncharacteristically small for the residual signals being measured. The small covariance made the filter highly sensitive to model and sensor errors in the system. Time histories of the residuals were studied and a new set of the values for the **R** matrix were determined. The new values were approximately two orders of magnitude larger than the old. With this change to the measurement covariance matrix, the problem with spikes in the estimate was gone.

2.4 Multiple Model Estimator

This section will present an alternative method of estimation which will be studied in this thesis. The theory and fundamental equations for the Multiple Model Estimator will be explained and then simplified for the single surface estimation problem, [3,10].

2.4.1 Theory of Operation

The multiple model estimator uses a collection of different models to form a minimum squared error estimate, [10]. Each model uses a different value of the parameter to be estimated. Values for the model's parameter are selected to span the estimation space. The output of each model is used to form a residual vector. The residuals are then used to propagate the conditional probability density functions for each model. The estimate is then calculated from the weighted average of the modeled parameters. The weighting factors are the conditional probabilities.

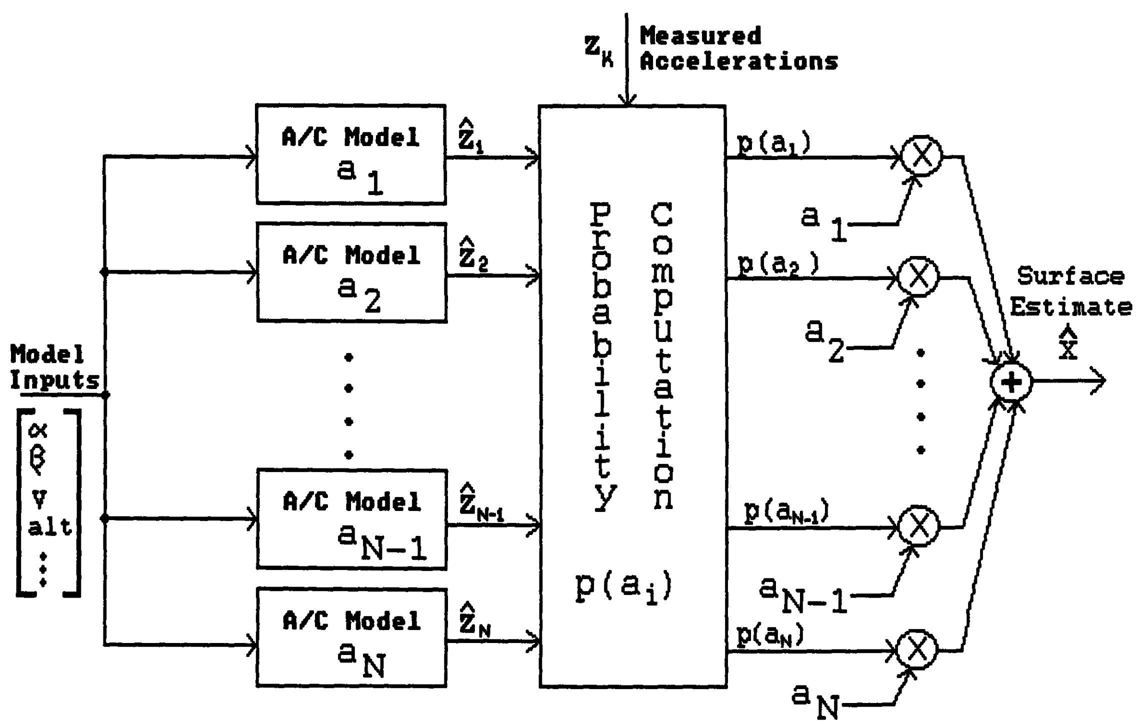
To begin the derivation of the multiple model estimator we will start with the estimate being a weighted sum of the probabilities. This is shown in Equation (2.4-1) for the continous spectrum of **a**. The integral shown in Equation (2.4-1) is very difficult, if not impossible, to calculate in a real-time estimator; therefore the integral is converted to a finite number of values for **a** which can be summed.

$$\hat{\mathbf{a}} = \int_0^1 \mathbf{a} p(\mathbf{a} | \mathbf{z}^*) d\mathbf{a} \qquad (2.4\text{-}1)$$

$$\hat{x} = \sum_{i=1}^{L} a_i p(a_i | z^*) \quad L = \text{number of models} \qquad (2.4\text{-}2)$$

Equation (2.4-2) states that the estimate, $\hat{x}$, is found by taking each possible value for **a** and weighting it by the probability of the value a_i conditioned on the set of measurements z^*. The vector z^* is the set of measurements for all time, while the vector **z** represents only one measurement. The difference is important, because we want to weight the potential estimates based on all of the measurements. A pictoral demonstration of this estimator can be seen in Figure 2.4-1.

Figure 2.4-1 Multiple Model Estimator



Typically, a bank of Kalman filters is used to generate the set of estimates from which the $\mathbf{a}_i$ terms in equation (2.4-2) are obtained. However, there is no need for that complexity in this problem. When a bank of Kalman filters is used, the resulting system is referred to as an adaptive Kalman filter. The system adapts by using the measurement residuals to select which filter is the most accurate.

An estimate of the surface effectiveness is the only parameter which is sought in this problem, thus the complex bank of filters can be replaced with different constant values for the vector parameter $\mathbf{a}_i$. The vector $\mathbf{a}_i$ can be reduced to a set of scalar constants, a_i. This reduction is possible because an estimate for a single surface is sought. These constants are then weighted by the probabilites to form the estimate of the surface effectiveness.

There still remains a fair degree of complexity because the probabilities still need to be calculated. In the more completed case of the adaptive Kalman filter, the conditional probabilities are calculated from the *a posteriori* error covariance matrix which is taken from each of the filters. Without the collection of Kalman filters, the probabilities cannot be calculated from continually updated error covariance matrices in the Kalman filters. In a much simpler manner the measurement and process noise statistics are used to calculate the error covariance matrix. The details of this calculation are covered in Section 2.4.3.

The system starts off with an *a priori* probability density function. This probability function contains no information about which of the values in discrete estimation space is correct. The uniform density function will weight all estimate candidates equally. Measurement data and the statistical information are used to calculate the conditional probabilities which are propagated forward each frame. Each frame of new information draws the estimator closer to the correct estimate. The probability functions shift from containing no information about the surface damage to a state in which the best estimate has the highest probability. It is this shift in the weighting factors or probabilites that is at the heart of the multiple model estimator. The next section will examine how the measurement information is incorporated in the probability density functions.

2.4.2 Probability Density Function Calculation

To calculate the probability density functions, we turn to Bayes Theorem for conditional probability. We are looking for the probability of a_i given the measurements $\mathbf{z}$, $p(a_i|\mathbf{z})$. Bayes Theorem is used to relate quantities which can be calculated directly to the desired conditional probability for the surface effectiveness, a_i.

$$p(a_i|\mathbf{z}^*) = \frac{p(\mathbf{z}|a_i)p(a_i)}{p(z)} \qquad (2.4\text{-}3)$$

Three terms are required for Equation (2.4-3): they are the *a priori* probability function for the effectiveness factor, $p(a_i)$, the probability of the measurements, $p(\mathbf{z})$, and the conditional probability of the measurements given a surface effectiveness.

The *a priori* probability function for the effectiveness factor starts with a uniform distribution. Initially, it contains no information about the surface effectiveness.

$$p_o(a_i) = \frac{1}{L} \qquad (2.4\text{-}4)$$

(where L is the number of models)

After the first frame, the *a priori* probability density function is taken to be the result of the previous iteration's $p(a_i|\mathbf{z})$. Taking the conditional probability of $p(a_i|\mathbf{z})$ and using it as the next iteration's $p(a_i)$ is, in effect, a recursive calculation of $p(a_i|\mathbf{z}^*)$. It is in this manner that information about all the previous measurements is carried forward to the next cycle's calculations.

The second term in Equation (2.4-3) is found by the following manipulation:

$$p(\mathbf{z}) = \sum_{j=1}^{L} p(\mathbf{z},a_j) \qquad (2.4\text{-}6)$$

$$= \sum_{j=1}^{L} p(\mathbf{z}|a_j)p(a_j)$$

The conditional probability function for the measurements, given the surface effectiveness, is found by assuming a Gaussian process. We can write $p(\mathbf{z}|a_j)$ as a multivariate normal density function as shown in Equation (2.4-6) below.

$$p(\mathbf{z}|a_j) = \frac{1}{(2\ )^{m/2}|\mathbf{C}(j)|^{1/2}} \exp\{.5[\mathbf{ez}_j^T\mathbf{C}(j)^{-1}\mathbf{ez}_j]\} \qquad (2.4\text{-}6)$$

(where m is the number of elements in **ez**)

The vector $\mathbf{ez}_j$ is residual between the measurement vector, **z**, and the model as a function of the surface effectiveness factor, a_j. This definition is shown in Equation (2.4-7). The same measurement vector is compared to all the different modeled vectors and thus the residual vectors, $\mathbf{ez}_j$, are formed. Each model then has its own conditional probability based on the same measurements. The model with the highest probability is the model which best matches the true surface impairment found in the measurements.

The matrix **C**(j) is the covariance matrix for the error vector $\mathbf{ez}_j$. In the more complex formulation where a bank of Kalman filters is used, **C** would be calculated from the error covariances of the model's Kalman filters. In this case, where there is no such set of filters, the matrix is determined in a different manner. The next section covers this calculation.

2.4.3 Covariance Calculation for Normal Probability

The covariance of the residual is required to properly calculate the probability density function for the residual vector. The residiual or error vector, **ez**, is defined as measured accelerations minus the modeled accelerations:

$$\mathbf{ez}_j = \mathbf{z}_{measured} - \mathbf{z}(a_j)_{modeled} \qquad (2.4\text{-}7)$$

Where $\mathbf{z}$ is the acceleration measurement which is treated in accordance with the model shown below:

$$\mathbf{z}_{measured} = \mathbf{H}_m\mathbf{s} + \mathbf{v} \qquad (2.4\text{-}8)$$

The vector, $\mathbf{s}$, is the aircraft state accelerations. Further, the process model is defined in much the same way as the measurment model.

$$\mathbf{z}(\mathbf{a}_j)_{modeled} = \mathbf{H}_p\mathbf{a}_j + \mathbf{w} \qquad (2.4\text{-}9)$$

For the problem at hand, the measurement observation matrix, $\mathbf{H}_m$, reduces to simply the identity matrix. This is because it is assumed that there is direct access to each of the aircraft acceleration. The modeled estimate of the accelerations, $\mathbf{z}(\mathbf{a}_j)_{modeled}$, is calculated with General Electric's Aircraft Model, SYSDYN. The matrix $\mathbf{H}_p$ relates the impairment factor vector, $\mathbf{a}_j$, to the aircraft accelerations. $\mathbf{H}_p$ is a non-linear function of aircraft states such as altitude, airspeed, angle of attack, and surface deflections.

The derivation of the covariance matrix with the observation matrices is presented next. For the actual application of the estimator, the measurement matrix $\mathbf{H}_m$ was set to the identity matrix.

It is assumed that the measurement and process noise statistics of $\mathbf{v}$ and $\mathbf{w}$ are known *a priori*. It is assumed that they are stationary with zero mean. Additionally, $\mathbf{v}$,

$\mathbf{w}$, $\mathbf{a}_j$, and $\mathbf{s}$ are uncorrelated. The statistics for $\mathbf{v}$ and $\mathbf{w}$ are as follows:

$$E\{\mathbf{v}\mathbf{v}^T\} = \mathbf{R} \qquad (2.4\text{-}10)$$

$$E\{\mathbf{w}\mathbf{w}^T\} = \mathbf{Q}$$

$$E\{\mathbf{v}\mathbf{w}^T\} = E\{\mathbf{w}\mathbf{v}^T\} = 0.0$$

The covariance of $\mathbf{ez}$ is defined as $E\{\mathbf{ez}\cdot\mathbf{ez}^T\} - E\{\mathbf{ez}\}^2$. The symbol E{} denotes the expected value operation.

First $E\{\mathbf{ez}_j\}$:

$$\begin{aligned} E\{\mathbf{ez}_j\} &= E\{\ \mathbf{H_m s} + \mathbf{v} - \mathbf{H_p a}_j - \mathbf{w}\} \\ &= E\{\ \mathbf{H_m s} - \mathbf{H_p a}_j\ \} \\ &= \mathbf{H_m s} - \mathbf{H_p a}_j \end{aligned} \qquad (2.4\text{-}11)$$

Second $E\{\mathbf{ez}_j\cdot\mathbf{ez}_j^T\}$:

$$\begin{aligned} E\{\mathbf{ez}_j\cdot\mathbf{ez}_j^T\} = E\{\ &(\mathbf{H_m s} + \mathbf{v} - \mathbf{H_p a}_j - \mathbf{w})\cdot \\ &(\mathbf{H_m s} + \mathbf{v} - \mathbf{H_p a}_j - \mathbf{w})^T\ \} \end{aligned} \qquad (2.4\text{-}12)$$

$$\begin{aligned} E\{\mathbf{ez}_j\cdot\mathbf{ez}_j^T\} = E\{\ &\mathbf{H_m s s^T H_m}^T + \mathbf{v}\mathbf{v}^T \\ &- \mathbf{H_p a}_j\mathbf{s}^T\mathbf{H_m}^T - \mathbf{H_m s a}_j^T\mathbf{H_p}^T \\ &+ \mathbf{H_p a}_j\mathbf{a}_j^T\mathbf{H_p}^T + \mathbf{w}\mathbf{w}^T\ \} \end{aligned}$$

$$\begin{aligned} E\{\mathbf{ez}_j\cdot\mathbf{ez}_j^T\} = (\ &\mathbf{H_m s s^T H_m}^T - \mathbf{H_p a}_j\mathbf{s}^T\mathbf{H_m}^T \\ &- \mathbf{H_m s a}_j^T\mathbf{H_p}^T + \mathbf{H_p a}_j\mathbf{a}_j^T\mathbf{H_p}^T \\ &+ \mathbf{R} + \mathbf{Q}\ ) \end{aligned} \qquad (2.4\text{-}13)$$

Subtracting the outer product of Equation (2.4-11) from Equation (2.4-13) leaves the desired covariance matrix of the residual.

$$C(j) = \text{Covar}(\mathbf{ez}_j) = E\{\mathbf{ez}_j \cdot \mathbf{ez}_j^T\} - E\{\mathbf{ez}_j\}^2 \qquad (2.4\text{-}14)$$

$$= \mathbf{Q} + \mathbf{R}$$

The covariance matrix **R** is obtained directly from the statistics for the sensors that are used. Biases in sensors are calibrated leaving only zero mean noise. The value for **Q**, on the other hand, is not as obvious. Ideally, it represents the process noise in the aircraft model, but practically, it is used to tune the estimator. Tuning requires simulation of the aircraft and the estimator model. Different values of **Q** are tested until an acceptable one is found. Small values in the **Q** matrix indicate a high degree of confidence in the model's representation of the true aircraft. For example, a null matrix for **Q** will cause the estimate to be more sensitive to errors between the model and the measurements.

2.5 Kalman Filter for Simultaneous Failures

The two previous estimators work only for impairments to a single surface, and only when that surface is known to be damaged. By expanding the Kalman filter to process all the surfaces, the estimator can detect and measure simultaneous damage to multiple surfaces. This section will examine the salient differences between the single surface problem and the multiple surface case. Similarities between the multiple surface estimator and impairment detection and classification problem will also be discussed.

2.5.1 Filter Expansion for Multiple Surfaces

The Kalman filter demonstrated in Section 2.3 was for an impairment to a single surface. It had one element in its state vector, and required a detection mechanism to determine which surface to evaluate. When the order of the state vector is increased to include an estimate for each surface, simultaneous failures can be tracked and compensated for. The aircraft which was used for this project has five surfaces: two stabilators, two ailerons and a rudder. The order of the estimator is thus increased to five.

The measurement and modeling data remain the same as in the single surface case. As stated above, the Kalman estimator state vector is increased from a scalar to a five element vector, and the associated observation matrix is increased from a five by one (5 x 1) matrix to a five by five (5 x 5) matrix. Other than the size increase of the matrices, the equations described in Section 2.3 remain unchanged. The larger sizes of the estimator state and observation matrix are not without cost. This formulation of the estimator takes more computational effort than its single surface counterpart.

2.5.2 Similarity to Impairment Detection Process

The multiple surface Kalman estimator is in some ways similar to the surface impairment detection mechanism which is required for the two single surface estimators. The residuals used for the multiple surface estimator are the same as those used by the impairment detection process. In the detection process, the residuals are projected onto an impairment signature vector. Each surface has its own failure signature and the signature which best matches the residual vector is declared impaired. The Kalman estimator, on the other hand, projects the residual vector directly onto an effectiveness estimate for each surface. When a surface is impaired the estimate for that surface is reduced based on the residual, while the detection process determines a match with one of the surface impairment signatures.

2.5.3 Constraints on the Surface Estimates

It was found that the multiple surface Kalman estimator would sometimes transiently confuse an impairment of one surface with anonther surface. Physically impossible estimates, those greater than one or less than zero, occurred when the same residual could be generated with a surface that is more effective than normal. The multiple surface estimator would eventually find the correct answer, but that required more measurements and hence more time. An example is in order to explain why the Kalman estimator would search in an impossible area of the solution space.

One of the test flight conditions had the aircraft trimmed at five degrees of sideslip (Beta). In this filght condition, both ailerons are deflected by a few degrees to counter the roll due to the rudder deflection. An impairment to one of the ailerons would generate a large residual in the roll axis. The same roll residual could also be generated by making the other aileron more effective than normal. While this is clearly impossible, the estimator has no way to know this and its estimate moves off towards an impossible value. The estimator will return and end up with the correct value, eventually. This is because the information from the other axes does not possess the high degree of symmetry the roll axis does for this flight condition.

In an effort to improve the speed of the estimate, it was found that a simple constraint on the estimates prevented unreasonable estimates and improved the time response. The estimate for each surface was checked against the valid range of zero to one. If a surface's estimate was found to be outside the range, it was limited. Estimation otherwise proceeded as normal. The constraint provides the estimator with additional knowledge about the parameters being estimated. This additional information speeds up the estimation process. Section 4.4.3 will provide time histories proving the effectiveness of the constraint on the estimate.

Chapter 3

Computational Issues

3.1 Introduction

The computational aspects of the estimators cannot be neglected. Since each of the estimators is designed to operate in a real-time environment, it is important to know what the computational costs are so the flight control computer is not over tasked. The basic unit of measure will be the floating point operation or FLOP. A FLOP is defined as one multiplication or one addition. The operation of a division is counted as two multiplications for the purpose of measuring execution costs. References [7] and [13] are used as justification for the double cost of division and for the equal cost of multiplication and addition. Using this rating, an accurate assessment of the processing requirements is obtained.

The algorithm for the single surface Kalman estimator is the same as for the multiple surface case. Thus only one section on the computation costs for the Kalman estimators will be presented.

3.2 Extended Kalman Filter Estimator Algorithm Costs

The equations presented in Chapter 2 are examined here in terms of their fundamental computational costs. Only the code in which computations (multiplication, division, addition and subtraction) are performed was counted. The additional code which makes up the estimator is, for the most part, overhead, dependent on the environment of the application. The assumption made is that the computer spends significantly more time performing the floating-point calculations than overhead logic and control code.

There are two parameters that determine the size of the matrices for the Kalman estimator. The first parameter is the number of estimator states. The single surface estimator contained one state, while the multiple surface estimator has a five-state estimate vector. The second parameter is the number of measurements made each cycle. In both cases, there were five measurements to be used by the estimator.

Let N be the number of estimator states (surfaces) and M be the number of measurements made each cycle. The total computation cost of the estimator will be expressed by a polynomial in N. Equation (3.1) shows the calculations done in the Kalman loop. The equations were broken into two parts to demonstrate the high cost of the matrix inverse as compared to the rest of the loop.

N = Number of States (3.1)
M = Number of Measurements

Filter Equations: $(3M)N^2 + (2M^2 + M)N$ Multiplies

$(2M + 1)N^2 + (3M^2 - M)N - M^2$ Additions

Matrix Inverse : $2M^2$ Divides

$2M^3 - 2M^2$ Multiplies

$2M^3 - 2M^2$ Additions

These results can now be summed to get a total cost. To express the expense in terms of FLOPs, the additions will be included with the number of multiplications. Equation (3.2) shows the resulting approximate cost:

$$\text{Cost} = (5M + 1)N^2 + (5M^2)N + (4M^3 - M^2) \quad \text{FLOPs} \tag{3.2}$$

Notice that the size of the inverse is related to the number of measurements only. Further, its cost is of the order M cubed. Clearly, this could get computationally expensive if M were large.

There is one other computational cost which is of some importance: that of the aircraft model used by the estimators. One pass through the aircraft model must be made in each cycle and its costs must be included. For the model used in this research, the cost was approximately 600 FLOPs.

3.3 Multiple Model Estimator Algorithm

The Multiple Model Estimator (MME) will be examined in the same manner as the Kalman filter. The main parameter in the cost function for the MME is not the number of surfaces being estimated, but rather the number of models used by the estimator. So for this discussion, N will be the number of models. Eleven different models were used to generate the data presented in Chapter 4.

Examination of the code in Appendix II reveals that the multiple model has following the operational cost per model:

Operation	FLOPs
6 Multiplies	6
2 Divides	4
8 Additions	8
1 Exponential	15
A/C Model	470
Total	503

Converting to an approximate number of FLOPs yields a cost of 33 FLOPs. The exponential was counted at 15 FLOPs, [7]. Just as in the Kalman estimator, the model costs must be included. This gives a total cost of approximately 503 FLOPs per model used. A total of 130 FLOPs out of the total 600 FLOPs are required for the Kalman H matrix and are not needed for the MME. A disproportionate part of the costs come from the model. This suggests that the model should be examined to save any of the calculation when executing the different passes through the model. If that were the case, the model's cost could be broken into a fixed cost plus a recurring cost. Only the recurring cost would need to be multiplied by the number of models used. This would greatly reduce the total cost of the MME as shown by Equation (3.3).

$$\text{MME Cost} = M_{fixed} + N(33 + M_{recurring}) \tag{3.3}$$

Where N is the number of models used and M is the cost of the model in FLOPs.

Fortunately, only the portion of the model that calculates the change in aircraft acceleration due to the impairment needs to be computed for each model in every iteration. The nominal accelerations can be calculated and saved for use with each different model. The change due to the impairment can be added as a correction to the nominal values for each model. Broken down this way, the aircraft model (SYSDYN) has a fixed cost of 435 FLOPs and a recurring cost of 35 FLOPs, as shown by Equation (3.4).

$$\begin{aligned} \text{MME Cost} &= 435 + N(33 + 35) \\ &= 1183 \quad \text{FLOPs for } N = 11 \end{aligned} \tag{3.4}$$

3.4 Comparison of Single Surface Estimation Costs

Both single surface estimators perform well, but the computational costs favor the Kalman estimator by a factor of almost two. Using Equation (3.2) one can see that the single surface Kalman estimator has a computational cost of 626 FLOPs for each estimator cycle. A cost ratio of MME estimator over the EKF estimator can be used as a direct comparison. In this case, the ratio of MME to EKF is 1.90. The cost ratio between the two estimators grows even larger when the Multiple Model Estimator is used for multiple surface damage. When multiple surfaces are involved, N in Equation (3.4) grows exponentially with the number of surfaces while Equation (3.2) remains quadratic. This makes the direct application MME techniques computationally prohibitive for more than one or two surfaces.

The MME technique is possibly best for cases where vastly different models are under consideration, such as different sensor failure modes. The Kalman filter structure avoids the problem of having to descritize the parameter space, thus there is no need for the large number of models.

Chapter 4

Results for Each Estimation Method

4.1 Introduction

This chapter presents the results for each of the estimators: single surface Kalman estimator, multiple model estimator, and the multiple surface Kalman estimator. Comparisons of the estimates made by the different estimators show the advantages and disadvantages of each method. Non-linear simulation data will be used to demonstrate the workings of the estimators.

With any complicated system, it is possible to have hundreds of test cases to check and verify that the system is working correctly. In order to keep the volume of the data down to a minimum, the same flight condition was used for each of the cases. Additional test conditions were used during the testing and development of the systems; however, these tests do not offer much additional insight into the capabilities of each system.

Because of the proprietary nature of the aircraft database used in this study, the aircraft response data cannot be published.

The flight condition used for each of the test cases is shown in Table 4.1. This condition was selected because all of the aircraft's surfaces are used to trim the aircraft. The testing procedure was to trim the aircraft, then let the simulation run for a few tenths of a second before turning on the estimator. For the single surface estimators, sequential impairments were made to the right stabilator. The estimators then tracted the damage to the surface.

Table 4.1: Aircraft Trim Condition

altitude	20,000.0	feet
airspeed	725.0	feet/second
alpha (approximately)	2.0	degrees
beta	5.0	degrees
bank angle	0.0	degrees
gross weight	28,800.0	pounds

4.2 Single Surface Extended Kalman Filter Results

4.2.1 Sequential Impairments to a Single Surface

The single surface Kalman estimator performed admirably. It followed the impairment very closly as shown in Figure 4.1. The estimator is started at six tenths of a second. This is before the first impairment occurs so we can see that the estimator will indicate the lack of an impairment. The first impairment of 35% loss of the right stabilator was started at 1.2 seconds into the test. The estimator followed the true impairment almost exactly. Two more changes in the surface impairment were made to demonstrate the sequential failure capability of the estimator. These failures were 60 and 80 percent losses to the stabilator at times 2.6 and 3.2 seconds, respectively. Again, the estimator tracks the surface impairment well.

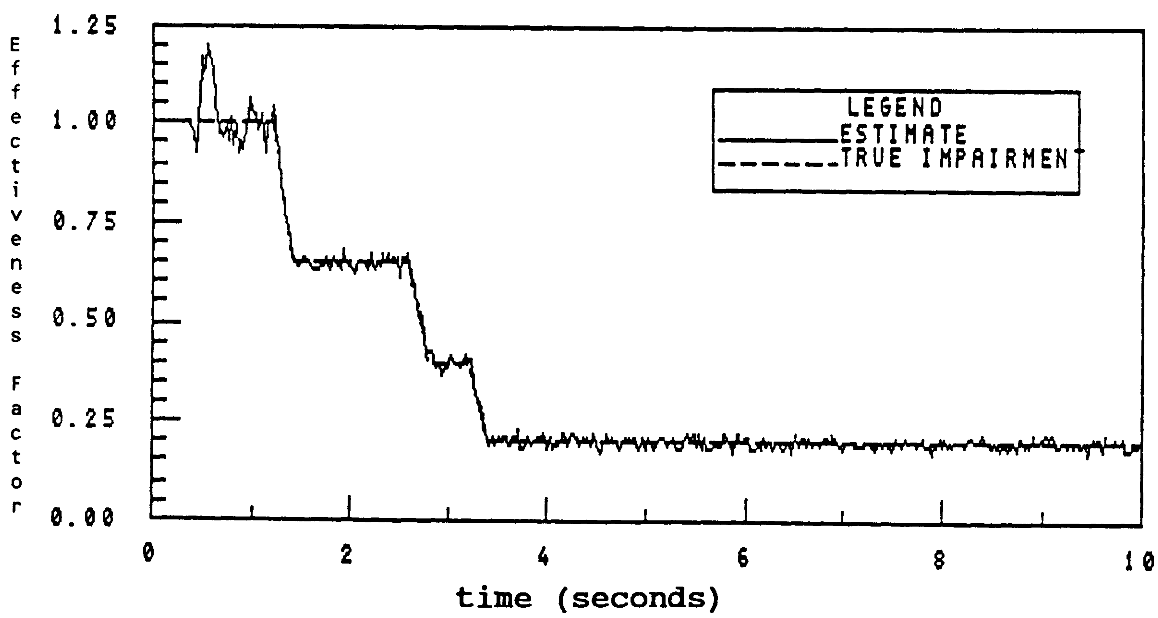


Figure 4.1: Single Surface Kalman Filter Estimate

The estimator must be tuned or calibrated to the statistics of the sensors and the model of the estimator state. The filter parameters that were used for this run are shown in Table 4.2 below:

Table 4.2: Single Surface Kalman Filter Parameters

P_o	0.005	(surface factor)^2
Q	0.005	(surface factor)^2
R_{pdot}	0.000034	(rad/sec^2)^2
R_{qdot}	0.000034	(rad/sec^2)^2
R_{rdot}	0.000034	(rad/sec^2)^2
R_{ay}	0.01	(feet/sec^2)^2
R_{az}	0.01	(feet/sec^2)^2

The five noise covariances for the diagonal matrix **R** were selected based on the strength of the noise added to each of the measured accelerations in the simulation. Noise was also added to the other measured aircraft state variables which were used to drive the SYSDYN aircraft model.

The initial covariance of the estimate, P_o, was arbitrarily set. The estimate is not sensitive to the initial value. As long as P_o is non-zero, the filter's projected (next cycle's) error covariance will settle to a value close to Q. Figure 4.2 shows the estimate's error variance. One may wonder why Q is non-zero. In the case of a single impairment assumption, Q would be zero. A detection system would signal that a failure has occurred and the estimator process would be started. The single impairment would be considered a random constant and a zero Q is the correct choice. The initial value for P_o represents the uncertainty of the begining estimate. The error variance will decrease as more measurements are taken. It is important to note that a value of zero for Q is only possible if an impairment detection system is used. Without a seperate detection system, the filter would have to run continously and P would drop to zero, thus preventing any change in the estimate.

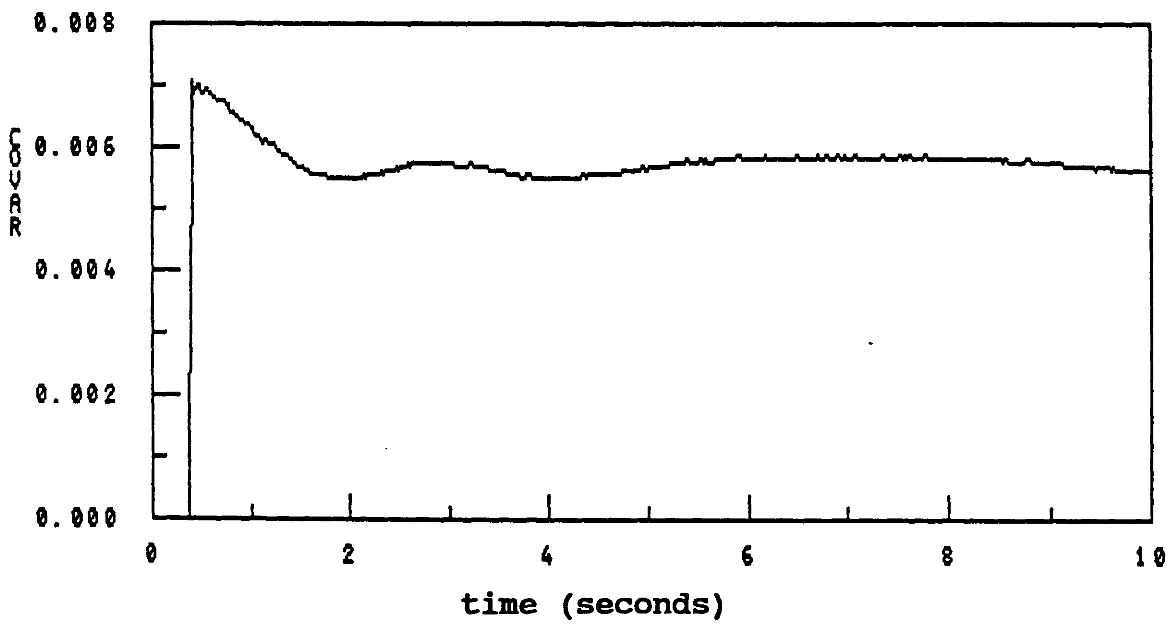


Figure 4.2: Single Surface Kalman Filter Error Variance

A zero value for Q implies thate the filter's state model would be that of a random constant. If this were the case, each new measurement would be weighted the same as all the previous measurements. As time passes, the estimator would become very sluggish and resist any change in the estimate. The error covariance would approach zero, and any changes in the surface damage would be detected slowly or missed completely. To prevent the filter from being sluggish, the surface factor being estimated is treated as a random walk rather than a random constant. A non-zero Q also has the benefit of speeding the estimation response.

The random walk model for the surface impairment has a non-zero variance Q. This model expects the impairment to change over time and the amount of change expected is set by the magnitude of Q. A small value for Q is all that is required to prevent the sluggish behavior of the filter. The larger the value of Q, the faster the filter will be able to respond to a change in the surface effectiveness estimate. This improvement in response is not without penalty, however. Too large of a variance for Q will pass too much noise to the surface estimate. Tuning the estimator consisted of selecting a value of Q which balanced the response speed with the noise attenuation of the estimate. The random walk model for the surface impairment model proved to be more robust than the random constant model.

4.2.2 Multiple Impairments for Single Surface Estimator

To see the potential advantage of the multiple surface estimator we now turn to the case where the single surface Kalman estimator is given a multi-surface impairment profile. The impairment profile was the same multiple surface profile as used for Figure 4.7. Each surface was impaired by the amount shown in Table 4.4. The single surface estimator was set to determine the effectiveness factor for the right stabilator. That is the same surface that was used for the other case already presented in Section 4.2.1. Figure 4.3 shows the estimator response:

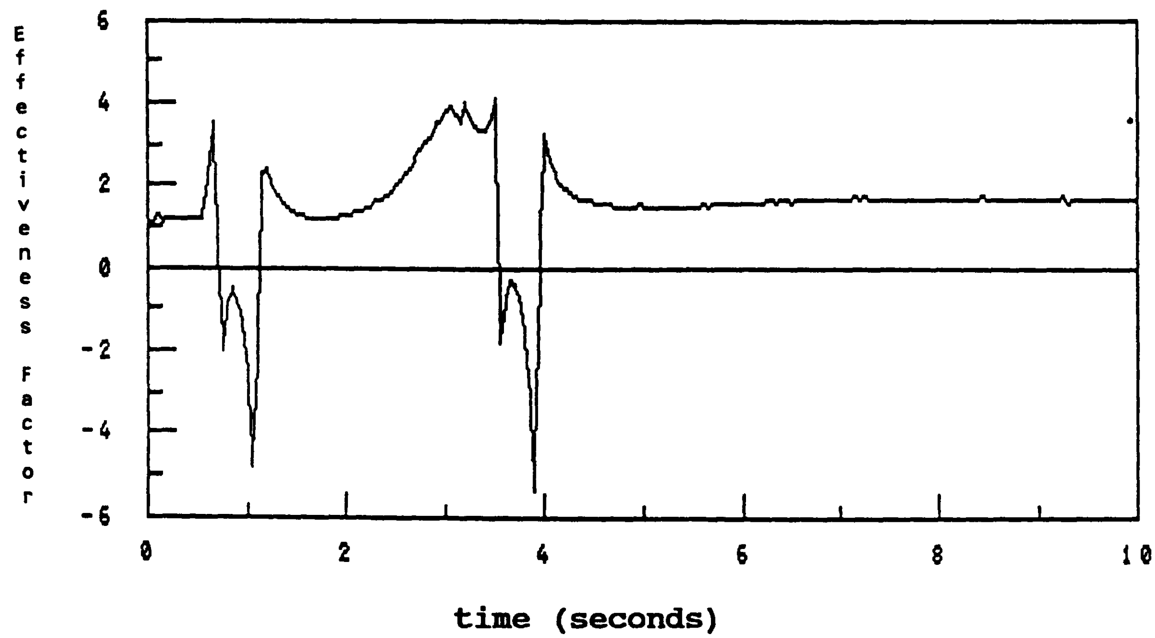


Figure 4.3: Single Surface Kalman Estimator Multiple Surface Impairment Profile

The resulting estimate is not close to the correct value. The steady state value of 1.7 which is reached for this example is not close to the correct value of 0.7 for the right stabilator. The error residuals are all attributed to the right stabilator which is being estimated. The system has no other place to allocate the errors seen in the residuals, so all the errors are put into a wrong estimate. While it is true that the estimate minimizes the difference between the impaired aircraft and the filter's model, the estimate is not useful for aircraft reconfiguration. The single surface system is overloaded.

4.3 Multiple Model Estimator Results

4.3.1 Sequential Impairments to a Single Surface

The same impairment profile was used to demonstrate the working of the multiple model estimator. This estimator also tracked the impairments well. Eleven different models were used for the estimator. Each of the models had a different value for the surface effectiveness parameter. The values ranged between zero and one by increments of 0.1 for a total of eleven models. This spread covered the parameter space with sufficient resolution to produce an accurate estimate.

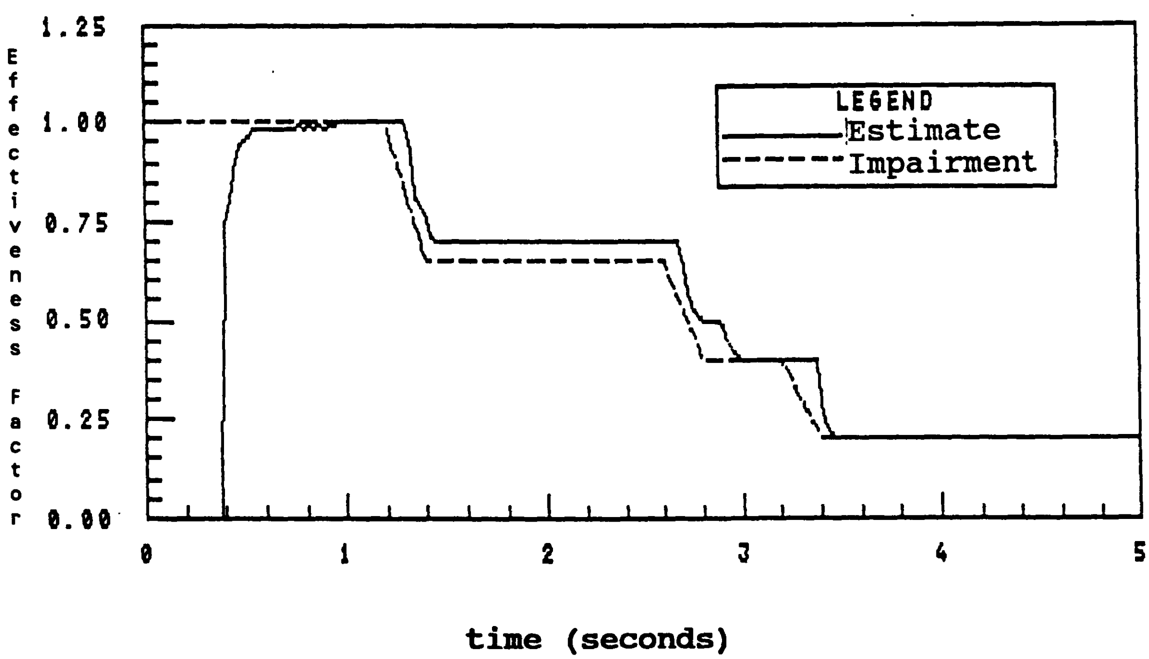


Figure 4.4: Multiple Model Estimate

Figure 4.4 shows the response of the estimator to the sequence of impairments. The true value of the impairment is plotted as a reference. The estimate approaches the impairment exactly when the real impairment matches one of the system's models. When the impairment does not match any of the models, the estimator selects the closest match. A blending of model values was expected when the actual impairment fell between two models. Unfortunately this was not seen. Between 1.4 and 2.6 seconds of the test shown in Figure 4.4, the true value of the impairment factor was 0.65. The estimator reaches a value of 0.7 and holds right there. The estimate never makes it to the correct value of 0.65 (an equal blend of 0.7 and 0.6). Each model has a strong point of attraction for impairments near the modeled parameter. The value of 0.7 was reached before the value of 0.6, so this was the value which was locked on to. The estimate transitions shown in Figure 4.4 indicate that the impairment factor moves by about 0.1 or more on the effectiveness scale before the estimate breaks free from its current modeled value. This effect is also seen in the vicinity of 2.8 seconds. The estimate holds a value of 0.5 on its way to the correct value of 0.4. In this case however, there is enough error to draw the estimate down to the correct value. Figures 4.5 and 4.6 show the time histories of the individual probabilites for the estimate shown in Figure 4.4.

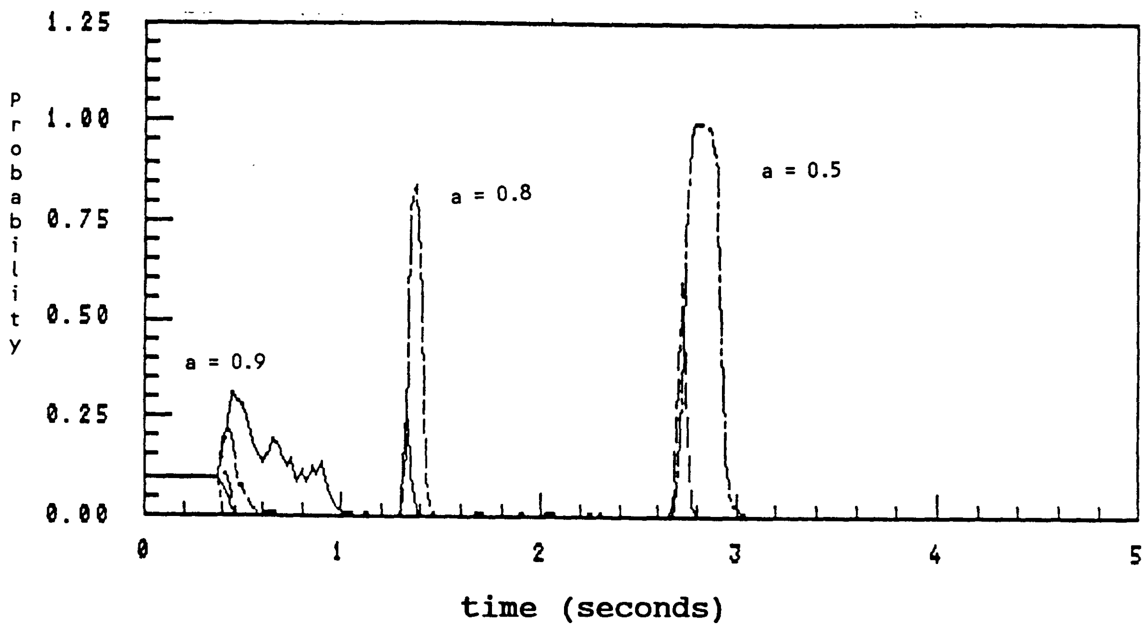


Figure 4.5: Multiple Model Conditional Probabilities Models (0.0, 0.1, 0.5, 0.8, 0.9)

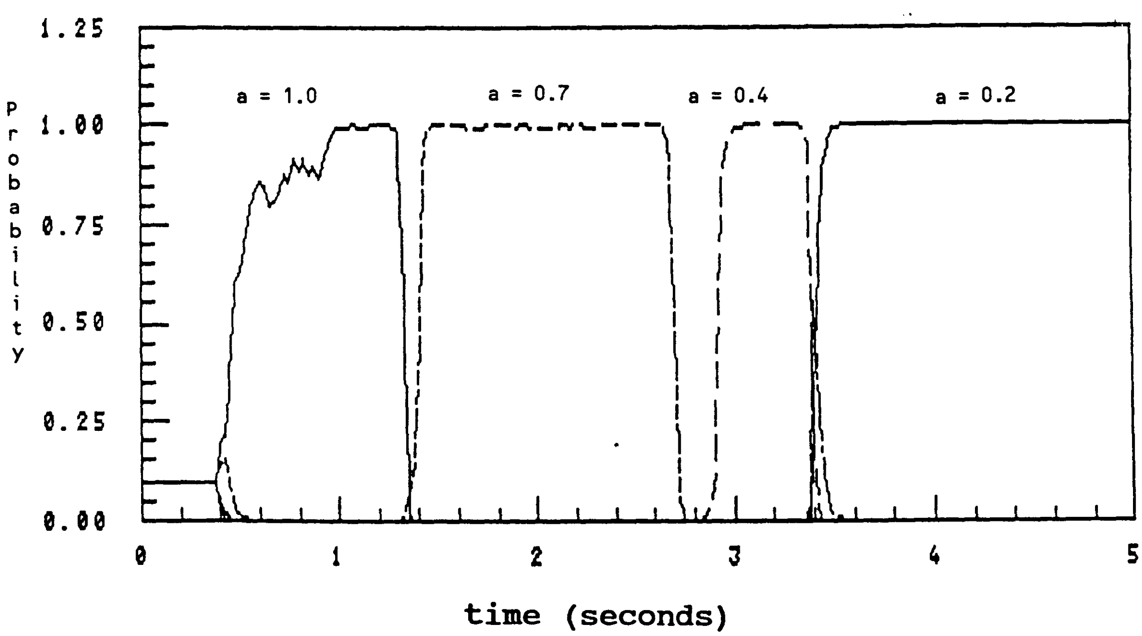


Figure 4.6: Multiple Model Conditional Probabilities Models (1.0, 0.7, 0.4, 0.2)

Each of the attraction points draws the conditional probability to the estimate which minimizes the error. If these attraction points are very strong it will be difficult to pull the estimate free: larger errors between the measured and modeled data are required. Further, blending is also difficult because of the strong attraction to the nearest point. There are two ways to improve the system in this respect. The first is to use more models. A greater spread of models will provide more locations for the estimate to fall into, thus a better estimate will be made despite the strong attraction points in the probabilities. The second way to flatten the steep attraction slopes is to increase the process noise term, Q. An increase in the diagonal elements of Q will cause the probabilities to have a weaker attraction: a better blending will be realized.

The estimate reaches one of the models when the probability for that model has become large. As the probability for one of the models increases, the other probabilities decrease. Given enough time, the decreasing probabilities would reach zero due to the numerical limitations imposed by a finite word length of the computer. Once a probability drops to zero, that model's probability can never be raised. This is a problem for sequential failure detection because the elimination of models will leave nowhere for the estimator to go.

To prevent the probabilities from going to zero, a minimum value is selected. Any probability that drops below this level is set to the minimum level. This level will cause errors in the estimate; however, the errors are small and bounded. A worst case error is found by assuming the actual impairment value is the smallest modeled value. In this case, that would be the zero surface effectiveness model. The estimate in the error would then be:

$$E = \sum_{i=1}^{L} a_i P_{min} \tag{4.3-1}$$

where P_{min} is the minimum allowable probability and a_i is the parameter value. With an additional limiting algorithm, this error can be accounted for and removed from the estimate. Because the error is small, however, the additional limiting algorithm was not implemented. The error in the estimate due to P_{min} can be kept small by the selection of a small value for P_{min}. The value used for this experiment was 0.00001, which kept the error to down to 0.000055. This value is well below the expected noise level of the system. Table 4.3 shows the sensor and process noise variances used to generate the data shown in Figures 4.4-6.

Table 4.3: Multiple Model Estimator Parameters

P_{min}	0.00001	dimensionless
Q_{pdot}	0.002	$(rad/sec^2)^2$
Q_{qdot}	0.002	$(rad/sec^2)^2$
Q_{rdot}	0.002	$(rad/sec^2)^2$
Q_{ay}	0.02	$(feet/sec^2)^2$
Q_{az}	0.04	$(feet/sec^2)^2$
R_{pdot}	0.000034	$(rad/sec^2)^2$
R_{qdot}	0.000034	$(rad/sec^2)^2$
R_{rdot}	0.000034	$(rad/sec^2)^2$
R_{ay}	0.01	$(feet/sec^2)^2$
R_{az}	0.01	$(feet/sec^2)^2$

4.3.2 Multiple Impairments for Multiple Model Estimator

The multiple model estimator was presented with the multi-surface impairment profile shown in Table 4.4. This resulted in a divide by zero error in the conditional probability calculation. To protect the estimator from a problem with numerical overflow, the exponent in the probability was limited to 80. When the multiple surface impairment profile was given to this estimator, all of the exponents calculated were greater than 80. The resulting large exponents imply that none of the models were very good candidates for the impairment. Preferably, the estimator would have selected one of the models. Ideally, the model which best matches the right stabilator effectiveness would have been selected. It is possible that double precision arithmetic could be used to allow larger exponents. This would prevent the divide by zero error and permit smaller probabilities. This way an effectiveness factor could still be calculated. If a surface factor is estimated, the potential to develop an interacting multiple model estimator exists. More will be said on this matter in Section 5.2.

4.3.3 Comparison of Single Surface Estimators

A comparison of the two single surface estimators is shown in Figure 4.7. It represents both surface factor estimates superimposed with the true value of surface effectiveness. The Kalman estimate follows the impairment better, but noise is seen in the estimate. The multiple model's estimate is smooth, but does not track off model impairments as well.

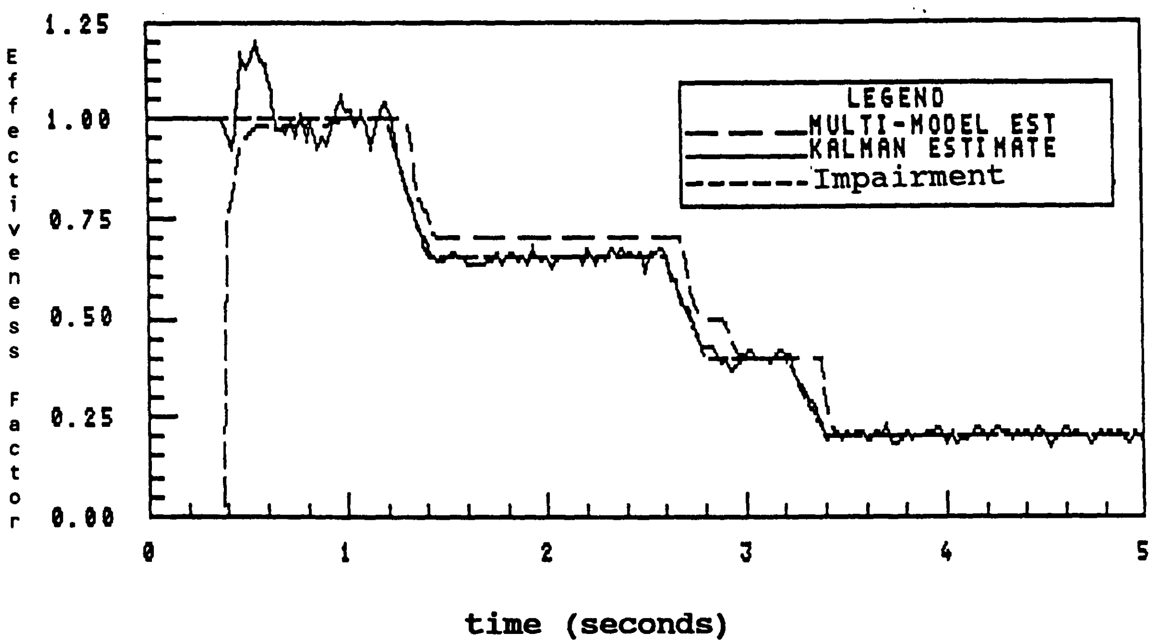


Figure 4.7: Comparison of Single Surface Estimators

4.4 Multiple Surface Extended Kalman Estimator Results

4.4.1 Response to Simultaneous Surface Impairments

The multiple surface Kalman estimator performed well. To test its capabilities, each of the aircraft surfaces was simultaneously impaired to a different value. Figure 4.8 shows the result of the impairments on the system. Notice that the estimates are not as steady as those of the single surface estimators. The reasons for this are discussed in section 4.5.1.

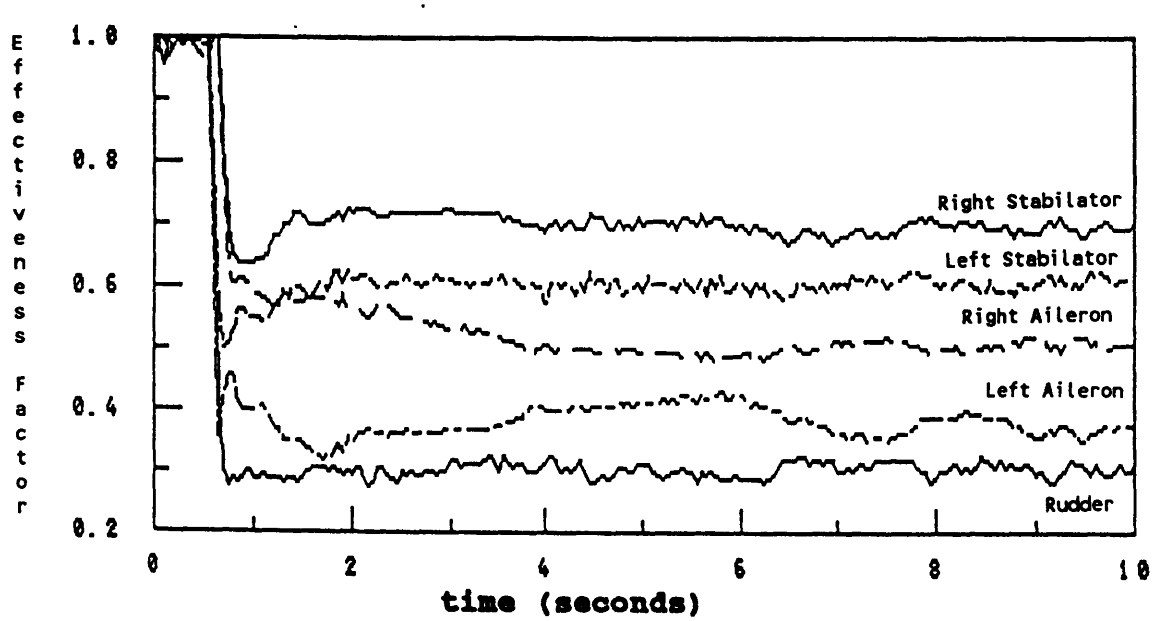


Figure 4.8: Multiple Surface Kalman Estimator

Table 4.4: Surface Effectiveness Values for Multiple Surface Impairment Test

SURFACE	EFFECTIVENESS
Right Stabilator	0.7
Left Stabilator	0.6
Right Aileron	0.5
Left Aileron	0.4
Rudder	0.3

The impairment levels for this run are shown in Table 4.4. Each surface was impaired a different amount to show the ability of the estimator to detect effectiveness levels in each surface at the same time. All five impairments go into effect at 0.6 seconds. The estimator is started at time zero for this test. Figure 4.8 shows that the estimates are at 1.0 for each surface before the impairments occur. At the time of the impairments, all the surfaces start to move towards the correct values.

As in the case of the single surface Kalman estimator, the speed and noise suppression are a function of the process noise matrix **Q**. In the multiple surface case, the diagonal of **Q** represents the amount of change expected in the process (surface factor) state. The random walk process (non-zero **Q**) was found to be most robust when the impairment is expected to change. Thus the random walk was used for the multiple surface case as well. It is important to make the multiple surface estimator robust to changes, because it does not have the same detection mechanism as the single surface system. The estimator is its own detection mechanism and examines all surfaces continously. The system runs continuously, and its estimates are at or near unity until a surface loss has occurred. The estimator cannot be allowed to get sluggish due to increased probability of missed detection.

The process variance values for each surface do not need to be the same. For example, the rudder's control authority is very different from that of the other surfaces. It was found that the rudder could have a larger process noise without much penalty on the smoothness of that estimate. This, of course, decreases the time it takes to settle on the rudder's estimate. The variances for the remaining four surfaces could not be made as large as the rudder's variance because of the noise content in the measurement data. The primary sensors for the rudder estimate are the side force and yaw acceleration residuals. The modeled component of these signals is tolerant to noise in the aircraft model. Normal acceleration, on the other hand, is very sensitive to noise on the angle of attack measurement (Alpha). The effect of noise on Alpha is seen on all surfaces except the rudder, thus the rudder's estimation time could be decreased without much penalty. The process variances used for the figures presented are shown in Table 4.5 below:

Table 4.5: Surface Process Variances (Q Matrix Diagonal)

SURFACE	Random Walk Variance
Right Stabilator	0.00001
Left Stabilator	0.00005
Right Aileron	0.00001
Left Aileron	0.00005
Rudder	0.0005

It was also found during the tuning process for the Q matrix that the estimator performed better when the left and right surfaces had different process variances. This too is reflected in Table 4.5. It did not make any difference whether the left surfaces had the larger variance or the right did, just as long as they were not the same. The reason for this is unclear. More work is required to investigate the cause.

4.4.2 Signal Information Content

The multiple surface Kalman estimator does much more than its single surface counterpart. It uses the same residuals as the single surface system but it estimates are for all five surfaces. Futher, it is designed to operate all the time as a detection mechanism. It is unrealistic to expect all this without some cost. To demonstrate this cost, a test case was run with the same impairment profile that was used to test the single surface estimator. The estimates are shown in Figure 4.9 below:

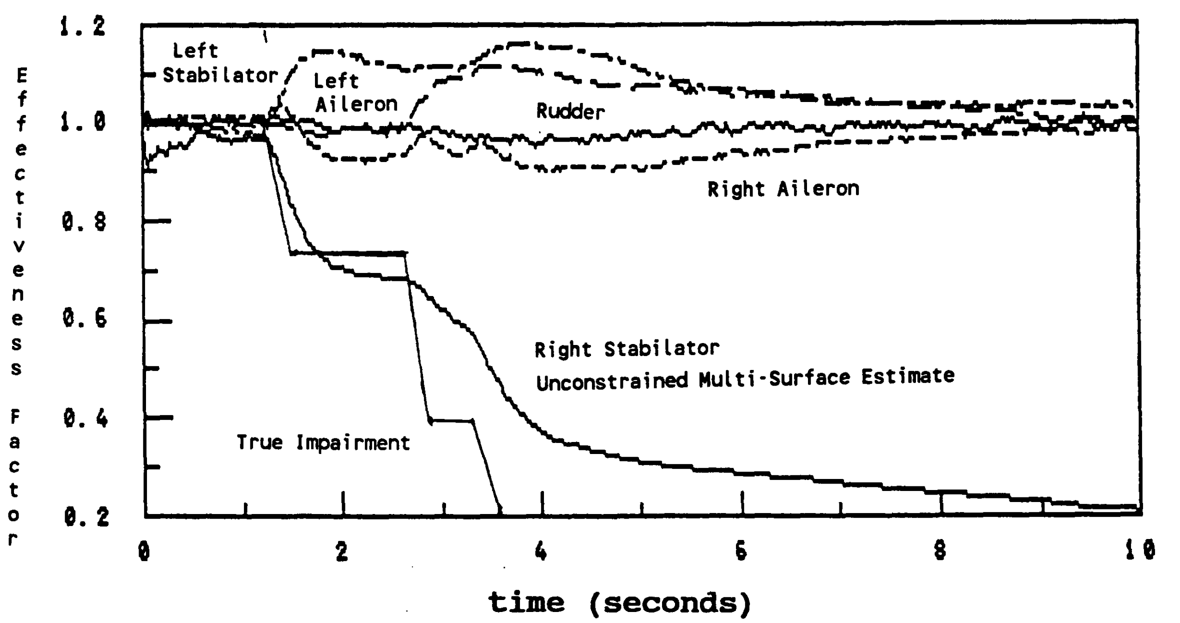


Figure 4.9: Multiple Surface Kalman Estimator
Single Surface Impairment Profile

Notice here that the estimate takes much longer to reach the final value of 0.2. It also does not converge to any of the intermediate impairment values. The other four surfaces also move from their unimpaired values. Clearly, this is incorrect. The interaction between the surfaces acts to slow the estimate and to cause the estimator to calculate wrong effectiveness values. Two of the surface estimates, the left stabilator and left aileron, converge to values significantly above one. By definition, this is wrong. The next section will discuss the change made to correct this problem. From Figure 4.9 we see the basic cost of the increased number of surfaces, which is the time. It takes longer for the estimator to isolate which surface is impaired. For the single surface system, this is not a problem because all residual information is assumed to belong to the surface being estimated; the knowledge of the surface being impaired is externally supplied.

4.4.3 Effects of Constraining the Estimate

Figure 4.9 demonstrated the problem of an unconstrained estimate. Impossible estimates were reached because they matched the residual data. An impairment on one side of the aircraft could be interpreted as an increase in effectiveness on the opposite side. Inherently, the filter has no mechanism, other than the aircraft model, to rule out these unrealistic estimates. It does eventually reach the correct value, but too slowly.

The constrained nature of the estimates can be included in the filter by simply checking each of the surface estimates after the adjustment of the previous frame's estimates. Each surface estimate is compared to the maximum and minimum values. Any estimate found to be outside this range is reset to the nearest acceptable value. That is, one for values greater than one and zero for those less than zero. This prevents the build-up of incorrect estimates and reinforces the correct direction for the estimate. Figure 4.10 shows the proof of this result. In this figure the single surface estimate and the true impairment can be compared to the constrained multiple surface estimator. The constrained estimator response is superior to the unconstrained shown in Figure 4.9, but it is still not as fast or accurate as the single surface estimator. The true time penalty due to the multiple surfaces is seen in this figure.

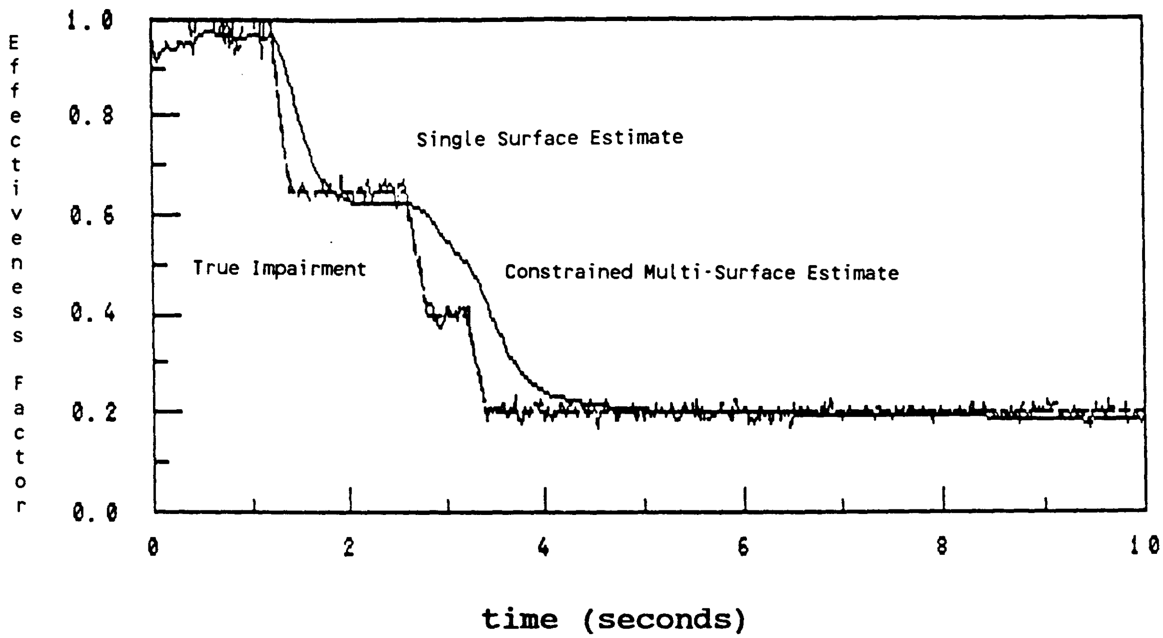


Figure 4.10: Constrained Multiple Surface Kalman Estimator Single Surface Impairment Profile

Chapter 5

Summary and Recomendations for Additional Research

5.1 Summary and Contributions of Thesis

The main contribution of this thesis has been the examination and comparison of three types of surface effectiveness estimators: two for the single surface impairment problem, and one for multiple surface impairment. The motivation for surface effectiveness estimation was explained in Chapter 1 with an emphasis on aircraft control reconfiguration as a means to compensate for damage. Each of the estimators was tested in a high accuracy, non-linear, six degree of freedom research simulation to prove the feasability of surface effectiveness estimation under realistic conditions. Benefits and limitations for the three systems were presented and discussed.

The single surface extended Kalman filter successfully tracked the single surface impairment profile. The resulting estimate was extremely close to the true impairment value. The estimate contained a small amount of high frequency noise, but this was not a problem since it was less than five percent of the maximum surface effectiveness. If it were a concern, a low pass filter could be used at the estimator's output to remove the noise. Response time of the single surface Kalman estimator was exceptional.

The multiple model estimator resulted from major simplifications made to an adaptive Kalman filter algorithm. The individual Kalman filters in the adaptive filter were replaced with algebraic aircraft models. Satisfactory results were demonstrated with the simplified adaptive filter. The computational cost associated with the bank of Kalman filters was found unnecessary; the simpler multiple model estimator can be used.

It was shown that an expansion of the extended Kalman filter, used for single surface estimation, could be used to provide estimates for all surfaces on the aircraft. A penalty in the amount of time for the estimation was the main cost. This was due to the increase in the number of surfaces. Further, the expanded estimator acted as an impairment detection and classification mechanism: an impairment in any surface would be reflected in the filter's estimate for that surface.

5.2 Recommendations for Further Research

There are a number of interesting areas related to surface effectiveness estimation that need further study and investigation. The impairment model used for this study was a very simple one. There is considerable work to be done in the area of impairment modeling. Wind tunnel data on surface damage would have been useful for developing a more sophisticated impairment model. Theoretical research using computational fluid dynamics may also lead to useful impairment models.

Each of the estimators presented share a number of desirable features. They also share the same Achille's heel. Damage to any of the sensors used by the estimator would result in incorrect estimate. Sensor failure and redundancy need to be accounted for to prevent incorrect aircraft reconfiguration. Additional Kalman filter states or additional models could be used to incorporate this information.

Methods for expanding the multiple model estimator so it can track simultaneous impairments should be investigated. The problem is that the number of models required for each additional surface increases on the order of the number of surfaces. For example, let the parameter space be divided into ten different failure values. (The single surface estimator designed for this thesis used eleven different models.) For two surfaces there would be 100 possible combinations of parameters. For three surfaces, there would be 1000 different combinations for the impairments to take. This pattern, unfortunately, continues so even a two surface model appears intractable.

However, the potential for an interactive multiple model estimator exists. An interactive model estimator would estimate the impairment for one surface at a time. After one surface has converged, the estimator would move to the next surface and estimate for that surface. The process would cycle from one surface to the next until all surfaces have converged. The estimate from one surface would be carried forward to the bank of models for the next surface's estimation cycle. This way the estimator moves closer to the true impairment with each surface switch. After a number of times through the cycle all the impaired surfaces should be accounted for. The interaction of the surface estimates is done by keeping the previous surface's estimate in each of the models for the current surface's estimate. This method may be a way to use the multiple model estimator without the problem of trying to calculate thousands of different models.

References

1. M. Athans, D. Castanon, K.P. Dunn, C.S. Greene, "The Stochastic Control of the F-8C Aircraft Using a Muliple Model Adaptive Control (MMAC) Method - Part 1: Equilibrium Flight," IEEE Trans. Automatic Control, Vol. AC-22, No. 5, pp. 768-780, October 1977.

2. H.A. Blom and Y. Bar-Shalom, "The Interacting Multiple Model Algorithm for System with Markovian Switching Coefficients," IEEE Trans. Automatic Control, Vol. AC-33, pp. 780-783, August 1988.

3. R.G. Brown, " Introduction to Random Signal Analysis and Kalman Filtering," John Wiley & Sons, New York, 1983.

4. S. Cramer and H. Sorenson, "Bayesian Parameter Estimation," IEEE Trans. Automatic Control, Vol. AC-33, No. 2, pp. 217-222, Feburary 1988.

5. G. Franklin and D. Powell, "Digital Control of Dynamic Systems," Addison-Wesley, Reading, Massachusetts, 1980.

6. A. Gelb, "Applied Optimal Estimation," M.I.T. Press, Cambridge, Massachusetts, 1974.

7. Intel Corporation, "iAPX 86,88 User's Manual," Intel Corporation, 1981.

8. D.G. Lainiotis, "Optimal Adaptive Estimation: Structure and Parameter Adaptation," IEEE Trans. Automatic Control, Vol. AC-16, pp. 160-170, April 1971.

9. L. Ljung, "Asymptotic Behavior of the Extended Kalman filter as a Parameter Estimator for Linear Systems," IEEE Trans. Automatic Control, Vol. AC-24, pp. 36-50, Feburary 1979.

10. P.S. Maybeck, "Stochastic Models, Estimation and Control," Vol. 2. Academic Press, New York, New York, 1982.

11. D. T. Magill, "Optimal Adaptive Estimation of Sampled Stochastic Processes," IEEE Trans. Automatic Control, Vol. AC-10, pp. 434-439, October 1965.

12. R.M. du Plessis, "Poor Man's Explanation of Kalman Filtering," North American Aviation,Inc., Autonetic Division, June 1967.

13. Rolm Corporation, "Hawk/32 Product Brief," Rolm Corporation, 1984.

14. B. Rust, "A Simple Algorithm for Computing the Generalized Inverse of a Matrix," Communications of the ACM, Vol. 9, No. 5, May,1966.

15. T. Sadeghi and G. Rosenthal, "A Control Law for the Self-Repairing/Digital Flight Control System," Proceedings of the National Aerospace and Electronics Conference, 1984, pp.591-595.

16. E.A. Wagner, "On-Board Automatic Aid and Advisory for Pilots of Control-Impaired Aircraft," Ph.D. Thesis, MIT, Cambridge Massachusetts, February, 1988.

Appendix I

Fortran Code for the Single Surface Extended Kalman Filter

The following FORTRAN source listing is the single surface Kalman filter code developed for this thesis.

```
      SUBROUTINE SINGEKF(NSURF)
C
C  SUBROUTINE SINGEKF IS CALLED BY THE EXECUTIVE PROGRAM
C  EVERY SAMPLE
C  THIS SUBROUTINE PERFORMS DATA PROCESSING,CALLS THE
C  RESIDUAL COMPUTATION SUBROUTINE EVERY SAMPLE,AND CALLS
C  THE EFFECTIVENESS ESTIMATION SUBROUTINE WHEN TRIGGERED
C  BY THE EXECUTIVE PROGRAM
C
C
C  R(5) - MEASUREMENT NOISE COVARIANCE.  Diagonal Matrix 5x5
C  Q - PROCESS NOISE COVARIANCE.
C  P - COVARIANCE OF ESTIMATE
C  P0 - INITIAL VARIANCE OF ESTIMATE
C
C  EZS(5) - ARRAY CONTAINING THE RESIDUALS COMPUTED BY RESID
C
C
      IMPLICIT NONE

      INTEGER*2 NSURF

      include 'EKF_OUT.inc'

      COMMON/WINDEG/ RSPD,RSQD,RSRD,RSNY,RSNZ
      COMMON /HPAST/ HDATA(5,5)
      REAL*4 RSPD,RSQD,RSRD,RSNY,RSNZ,HDATA

      COMMON /EGE_DATASET/Q_CHOICE(5),P0_CHOICE(5)

      REAL*4 Q_CHOICE, P0_CHOICE


      CHARACTER*75 LABEL


      COMMON /CONVER/ OFF_TOLER, NOFF, MIX_TOLER, NMIX,
     .                EST_WIND(10),NCNT,IOFF,IMIX
      REAL*4 MIX_TOLER,EST_WIND

      COMMON /COVARI/ Q,R(5),P0
      REAL*4 Q,R,P0

      COMMON /S_KALOUT/ XS_EST,EZS(5), HIN(5), P, KG(5)
      REAL*4 XS_EST, EZS, HIN,P,KG
      DATA XS_EST /1.0/

      REAL*4 THRESH
      INTEGER*4 I,J

      INTEGER*2 PSLID_LATCH
      data PSLID_LATCH /0/
```

```
      logical INIT_FLAG
      DATA INIT_FLAG /.TRUE./

C
C     LOAD THE DATA FOR THE RUN
C
      IF (INIT_FLAG) THEN
          OPEN(UNIT=35,FILE='COVAR.QN',STATUS='OLD')
 125  FORMAT(A75)
          READ(35,125) LABEL
          READ(35,*) (P0_CHOICE(I), I = 1,5)
          READ(35,125) LABEL
          READ(35,*) (Q_CHOICE(I), I = 1,5)
          READ(35,125) LABEL
          READ(35,*) (R(I), I = 1,5)
          CLOSE(UNIT=35,STATUS='KEEP')

          INIT_FLAG = .FALSE.
          DO I= 1,5
            EST_SAV(I) = 1.0
          END DO
        ENDIF      ! POWER UP


      IF (NSURF .NE. PSLID_LATCH) THEN
        P = P0_CHOICE(NSURF)
        Q = Q_CHOICE(NSURF)

        XS_EST = 1.0

        DO I=1,5
          EST_SAV(I) = 1.0
        ENDDO

        IOFF = 0
        IMIX = 0
        NCNT = 1
        PSLID_LATCH = NSURF
      ENDIF

      IF (NSURF.EQ.0) RETURN

C
C   If there is a failure, call the Kalman filter estimator
C
C       Form the error vector ez
C
        EZS(1) = RSPD
        EZS(2) = RSQD
        EZS(3) = RSRD
        EZS(4) = RSNY
        EZS(5) = RSNZ
C
C    XS_EST =  A  (effectiveness estimate)
```

```
C
C                    T
C   SETUP THE H AND H MATRICES
C
        DO J = 1,5
          HIN(J) = HDATA(NSURF,J)
        END DO

        CALL KALEST

      RETURN
      END


      SUBROUTINE KALEST
C
C     XS_EST - STATE ESTIMATE VECTOR
C     EZS - MEASUREMENT ERROR VECTOR  (from RESID)
C     H - OBSERVATION MATRIX
C
C     P - STATE ESTIMATE VARIANCE
C     Q - PROCESS NOISE COVAIRANCE
C     R - MEASUREMENT NOISE COVARIANCE MATRIX
C     SIG3 - 3 standard deviations for noise covariance

      IMPLICIT NONE

      COMMON /S_KALOUT/ XS_EST,EZS(5), H(5), P, KG(5)
      REAL*4 XS_EST, EZS, H,P, KG

      INCLUDE 'EKF_OUT.INC'

      COMMON /COVARI/ Q,R(5)
      REAL*4 Q,R

      COMMON /CONVER/ OFF_TOLER, NOFF, MIX_TOLER, NMIX,
     .                EST_WIND(10),NCNT,IOFF,IMIX
      REAL*4 MIX_TOLER,EST_WIND,OFF_TOLER
      INTEGER*4 NOFF,NMIX,NCNT,IOFF,IMIX

      REAL*4 SUM
C
C     Space for local temporary variables
C
      DIMENSION HP(5),HPH_R(5,5),PZINV(5,5)
      REAL*4 HP,HPH_R,PZINV,DX

      INTEGER*4 I,J,RANK

C
C    Calculate Kalman Gain
C
C                   -1
C     K = PH'(HPH'+R)
```

```
C

      do i= 1,5
        HP(i) = H(i)*P
      end do

      do i = 1,5
        do j = 1,5
          HPH_R(i,j) = HP(i)*H(j) ! HPH'(vector outer
                                         product)
        end do
        HPH_R(i,i) = HPH_R(i,i) + R(i)  ! HPH' + R
                                         ! (symetric pos.
                                           definite matrix)
      end do

      CALL GMINV(5, 5, 5, HPH_R, PZINV, RANK)

      do i = 1,5
        KG(i) = 0.0
        do j = 1,5
          KG(i) = KG(i) + HP(j) * PZINV(j,i)
                                 ! note: HP = PH
        end do
      end do

C                         ^
C     Update the estimate X
C       ^   ^              ^
C       X = X + KG(z - Hx) = X + KGez
C
      DX = 0.0
      do i = 1,5
        DX = DX + KG(i) * EZS(i)
      end do
      XS_EST = XS_EST + DX

C
C     Compute the error covariance P for this estimate
C     Uses optimal gain K assumption
C
C       P = (I - KH)P  = P - KHP
C
      SUM = 0.0
      do i = 1,5
        SUM = SUM + KG(i) * HP(i)
      end do
      P = P - SUM
C
C     Project ahead the error covariance P for
C     the next measurement cycle
C
C      P = P + Q
C
```

```
P = P + Q

RETURN
END
```

Appendix II

Fortran Code for the Multiple Model Estimator

The following FORTRAN source listing is the single surface Multiple Model Estimator code developed for this thesis.

```
        SUBROUTINE multi(SURF)
C*************************************************************
C
C        Multi-model Estimation
C
C*************************************************************
C
C       SURF        number corresponding to failed surface
C
C**************************************************************

        integer*4 SURF

        integer*4 i,j
        logical*2 init

        include 'sensor.inc'   ! Generic A/C sensor package

        data init /.false./
c
c       interface common block to A/C model
c
        common /MM_OUT/ theta_hat,p(20)
        real*4 theta_hat,p

        common /MMest/ NQUAN,theta(20),pd_m(20), qd_m(20),
     .                rd_m(20),ay_m(20),az_m(20),S(5)

        COMMON /MMCOVARI/ Q(5),R(5)

        integer*4 NQUAN
        real*4    theta,pd_m,qd_m,rd_m,ay_m,az_m,S

        dimension v(5), temp(5), S_1(5)
        real*4 v,temp,S_1

        real*4 l(20),m(20),detS,B, MINPROB

        integer*4 pmax,adj_cnt

 125    FORMAT(A75)
c
c       Initialize the estimation space
c
        NQUAN = 11
        if (.not.init) then
c         Read to Covariances Q and R
          pi2 = (2.*3.14159265)

          OPEN(UNIT=35,FILE='mmCOVAR.qn',STATUS='OLD')
          READ(35,125) LABEL
          READ(35,*)  (Q(i), i = 1, 5)
          READ(35,125) LABEL
          READ(35,*)  (R(i), i = 1, 5)
```

```
          READ(35,125) LABEL
          READ(35,*)  MINPROB
          CLOSE(UNIT=35,STATUS='KEEP')
c
c     Calculate the covariance matrix, Q,R are constant
c     matrices
c
          do j = 1, 5                         ! pd,qd,rd,ay,az
            S(j) = Q(j) + R(j)          ! S = HQH + R, but H = I
            S_1(j) = 1./S(j)            !for multi mdl
          end do
c                    5         -.5
c       B = [(2*pi)^ * |S^ |]^
c
          detS = S(1)
          do j = 2,5
            detS = detS * S(j)           ! S is the diagonal
          end do                         ! noise covairance matrix

          B = 1.0/sqrt(detS*(pi2**5))
c
c      initialize the probabilites
c
          do i=1, NQUAN
            theta(i) = real(i-1)/(NQUAN-1)
            p(i) = 1./(NQUAN)
          end do

          init = .true.
        endif
c
c       end initialization
c
        if (SURF.EQ. 0) return

        call Mmodel(SURF) ! calculate the modeled
                          ! accelerations
                          ! for the surface in question for
                          ! each theta(i)

c
c       Start here, looping through each of the possible
C       estimate values.  theta(i) | i = 1,2 ... NQUAN
c
        do i = 1, NQUAN               ! loop through the
                                      ! parameter space
c
c         form the residual vector
c          v = measured - modeled accelerations
c
          v(1) = PDOTS - pd_m(i)
          v(2) = QDOTS - qd_m(i)
          v(3) = RDOTS - rd_m(i)
          v(4) = AYS   - ay_m(i)
```

```
          v(5) = AZS    - az_m(i)

c
c       calculate and limit the exponents
c               -1
c       m = v' S^ v/2
c
            m(i) = 0.0
            do j = 1,5
              m(i) = m(i) + v(j)* S_1(j)*v(j)
            end do
            m(i) = m(i)/2.
c
c       limit the exponent
c
          if (m(i) .gt. 80.) then
            l(i) = 0.0
          else
            l(i) = B*exp(-m(i))
          endif
        end do   ! parameter space loop (i=1, NQUAN)

c
c       calculate the a posteriori probabilites
c
        sum = 0.0
        do j = 1, NQUAN   ! sum over the whole space
          sum = sum + l(j)*p(j)
        end do

        adj_cnt = 0
        pmax = 1
        do j = 1, NQUAN
          p(j) = l(j)*p(j)/sum   ! weight and normalize
          if (p(j) .lt. MINPROB) then
            p(j) = MINPROB
            adj_cnt = adj_cnt + 1
          endif
          if (p(pmax) .lt. p(j)) then
            pmax = j
          endif
        end do


c
c       calculate the estimate
c
        theta_hat = 0.0
        do i = 1, NQUAN
          theta_hat = theta_hat + p(i)*theta(i)
        end do
        return
        end

        SUBROUTINE Mmodel(SURF)
```

```
C*************************************************************
C
C        Multi-model - generates the acceleration vector for
C                      each of the aircraft models.
C
C        Uses General Electric's SYSDYN Model
C
C*************************************************************
C       SURF         number corresponding to failed surface
C
C*************************************************************
       integer*4 SURF

        include 'SYSDYN.INC' ! interface to GE's SYSDYN
        include 'SENSOR.INC' ! interface to A/C sensors

        COMMON /sktable/ ISKIPTBL    ! so AERO model table
                                     ! lookups can be
        INTEGER*4 ISKIPTBL           ! skipped

C
C       interface common block for multiple A/C models
C
        common /MMest/ NQUAN,theta(20),pd_m(20), qd_m(20),
                   rd_m(20),ay_m(20),az_m(20),S(5,20)

        integer*4 NQUAN
        real*4    theta,pd_m,qd_m,rd_m,ay_m,az_m,S
C
C     setup SYSDYN input parameters with SENSOR data
C
        ALPH = ALS
        BETA = BETAS
        THE=PITS
        PHI=ROLS
        PB=PS
        QB=QS
        RB=RS
        VT = VT_SENSOR

C
C   SET THE SURFACE POSITIONS
C
      DELTA(1,1) = STABRS
      DELTA(2,1) = STABLS
      DELTA(3,1) = AILRS
      DELTA(4,1) = AILLS
      DELTA(5,1) = RUDS

      ISKIPTBL = 0
      IEGEPASS = 1         ! Tell SYSDYN this is an estimator
                           ! pass.
        do i = 1, NQUAN
          A(SURF) = theta(i)   ! set the model's estimate
```

```
    call SYSDYN
    pd_m(i) = PDOT(1)      ! Store the accelerations
    qd_m(i) = QDOT(1)
    rd_m(i) = RDOT(1)
    ay_m(i) = AY(1)
    az_m(i) = AZ(1)
    ISKIPTBL = 1           ! ONLY DO TBL LOOKUPS ONCE
  end do  ! i = 1, NQUAN
IEGEPASS = 0
ISKIPTBL = 0
RETURN
END
```

Appendix III

Fortran Code for the Multiple Surface Extended Kalman Filter

The following FORTRAN source listing is the multiple surface Kalman filter code developed for this thesis.

```
      SUBROUTINE MULTIEKF
C
C  QDOT - MEASURED PITCH ACCELERATION  (RAD/SEC**2)
C  PDOT - MEASURED ROLL ACCELERATION
C               (RAD/SEC**2)
C  RDOT - MEASURED YAW ACCELERATION
C               (RAD/SEC**2)
C
C  IPASS - INITIALIZATION CODE (1 FOR INITIALIZE)
C  R(5,5) - MEASUREMENT NOISE COVARIANCE.
C  Q(5,5) - PROCESS NOISE COVARIANCE.
C  P(5,5) - COVARIANCE OF ESTIMATE
C  P0(5)  - INITIAL VALUE FOR COVARIANCE MATRIX DIAGONAL
C
C  EZ(5) - ARRAY CONTAINING THE RESIDUALS COMPUTED BY RESID
C
C
      LOGICAL NINIT,NCLEAR

      REAL NYS,NZS,NYSI,NZSI

      include 'EKF_out.inc'

      COMMON /INDYN/ ALPHS,BETASY,DELTA(6,2),THES, PHIS,
     . PBS, QBS,RBS,AXBH,ALT,VTS,AS(6)

      COMMON /OUTDYN/ FILLER(12), B(5,6), H(5,6)


      CHARACTER*75 LABEL
      DATA EST_SAV /1.,1.,1.,1.,1./
      DATA X_EST /1.,1.,1.,1.,1./

      COMMON/WINDEG/ RSPD,RSQD,RSRD,RSNY,RSNZ

      COMMON /HPAST/ HDATA(5,5)


      COMMON /COVARI/ P(5,5),Q(5,5),R(5,5),P0(5)

      COMMON /KALOUT/ X_EST(5),EZ(5)

      DIMENSION HIN(5,5), HINT(5,5)

      DATA NINIT /.TRUE./
      DATA NCLEAR /.TRUE./
      DATA IPASS /1/

        IF (NINIT) THEN         ! "power on"
          OPEN(UNIT=35,FILE='COVAIR.QN',STATUS='OLD')
 125  FORMAT(A75)
          READ(35,125) LABEL
```

```
          READ(35,*) (P0(I), I = 1,5)
          READ(35,125) LABEL
          READ(35,*) (Q(I,I), I = 1,5)
          READ(35,125) LABEL
          READ(35,*) (R(I,I), I = 1,5)
          CLOSE(UNIT=35,STATUS='KEEP')

          NINIT = .FALSE.
          DO I= 1,5
            EST_SAV(I) = 1.0
          END DO
        ENDIF      ! POWER UP

C
C     START INITIALIZATION
C
      IF (IPASS.EQ.1) THEN
        DO I = 1,5
          X_EST(I) = 1.0
        END DO
        ICONVR = 0

        DO I=1,5
          EST_SAV(I) = AS(I)
        ENDDO
      ENDIF
C
C   END OF INITIALIZATION
C

C
C  CLEAR OUT OLD WINDOW VALUES
C
      IF (IPASS.EQ.1) THEN
        IF (NCLEAR) THEN
          NCLEAR = .FALSE.
          IPASS = 0
          DO I = 1,5
            DO J = 1,5
              P(I,J) = 0.0
            END DO
          END DO
          DO J = 1,5
            P(J,J) = P0(J)
          END DO
          DO I = 1,5
            EST_SAV(I) = X_EST(I)
          END DO
        ENDIF
      ELSE
        NCLEAR = .TRUE.
      ENDIF
```

```
      EZ(1) = RSPD
      EZ(2) = RSQD
      EZ(3) = RSRD
      EZ(4) = RSNY
      EZ(5) = RSNZ
C
C    X_EST() =  RTSTAB
C               LTSTAB
C               RTAIL
C               LTAIL
C               RUDDER
C
C                    T
C   SETUP THE H AND H MATRICES
C
      DO J = 1,5     ! ACCELERATION
        DO I = 1,5   ! NUMBER OF SURFACES 1,5
          HIN(J,I) = HDATA(I,J)
          HINT(I,J) = HDATA(I,J)
        END DO
      END DO

      CALL KALEST(X_EST,EZ,HIN,HINT)

      RETURN
      END


      SUBROUTINE KALEST(XHAT,EZ,H,HT)
C
C     XHAT - STATE ESTIMATE VECTOR
C     EZ - MEASUREMENT ERROR VECTOR  (from RESID)
C     H,HT - OBSERVATION MATRIX AND ITS TRANSPOSE
C
C     P - STATE ESTIMATE COVARIANCE MATRIX
C     Q - PROCESS NOISE COVAIRANCE
C     R - MEASUREMENT NOISE COVARIANCE
C
      include 'EKF_out.inc'

      DIMENSION XHAT(5),EZ(5),H(5,5),HT(5,5)

      COMMON /COVARI/ P(5,5),Q(5,5),R(5,5)

C
C     Space for local temporary variables
C
      DIMENSION HP(5,5),TEMP(5,5),TEMP2(5,5),PZINV(5,5)
      DIMENSION K(5,5), DX(5)
      REAL*4 K

C
C    Calculate Kalman Gain
C
```

```
C                        -1
C      K = PH'(HPH'+R)
C

       CALL MATMUL(5, H, P, 5, 5, 5, HP)
       CALL MATMUL(5, P, HT, 5, 5, 5, TEMP)
       CALL MATMUL(5, HP, HT, 5, 5, 5, TEMP2)
       CALL MATADD(5, TEMP2, R, 5, 5)
       CALL GMINV(5, 5, 5, TEMP2, PZINV, IRANK)
       CALL MATMUL(5, TEMP, PZINV, 5, 5, 5, K)

C                         ^
C     Update the estimate X
C
C       x = x + K(z - Hx) = x + Kez
C

       CALL MATMUL(5, K, EZ, 5, 5, 1, DX)
       CALL VADD(5,1.0,XHAT,DX)
C
C     Constrain the estimate
C
       DO I = 1,5
         IF (XHAT(I) .GT. 1.0) THEN
           XHAT(I) = 1.0
         ENDIF
         IF (XHAT(I) .LT. 0.0) THEN
           XHAT(I) = 0.0
         ENDIF
       END DO

C     Compute the error covariance P for this estimate
C     Uses optimal gain K assumption
C
C       P = (I - KH)P  = P - KHP
C
       CALL MATMUL(5, K, HP, 5, 5, 5, TEMP)
       CALL MATSUB(5, P, TEMP, 5, 5)

C
C     Project ahead the error covariance P for the
C     next measurement
C
C      P = P + Q
C
C      Projection ahead of x is done in call to sysdyn
C
       CALL MATADD(5, P, Q, 5, 5)

       RETURN
       END
```