71-14,224

FINCH, Robert Bruce, 1944-
SMOOTHING FOR DELAYED STATE MODEL WITH APPLICATIONS TO AIDED INERTIAL NAVIGATION SYSTEMS.

Iowa State University, Ph.D., 1970
Engineering, electrical

Smoothing for delayed state model with applications to aided inertial navigation systems

by

Robert Bruce Finch

A Dissertation Submitted to the
Graduate Faculty in Partial Fulfillment of
The Requirements for the Degree of
DOCTOR OF PHILOSOPHY

Major Subject: Electrical Engineering

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

Head of Major Department

Signature was redacted for privacy.

Dean of Graduate College

Iowa State University
Ames, Iowa

1970

TABLE OF CONTENTS

I. INTRODUCTION

Although many contributions have been made to estimation theory, the contribution which made the first great impact on engineering was made by Wiener (22). His research was concerned with the continuous estimation problem. That is, the measurement data is a continuous record. The development presented by Wiener, using Fourier analysis, arrives at the celebrated Wiener-Hopf equation which must be solved in order to obtain estimators. This approach to the estimation problem includes the parameter, frequency, therefore, allowing the engineer to gain his all important "feel" for the estimators in terms of filtering theory.

With the coming of state space and the digital computer the discrete estimation problem came to light. Inherent in any estimation problem is the determination of the expression for the best estimator. The methods for defining the best estimator are many, leading to a number of assorted expressions. One criterion for a best estimator is least mean squared error. Such a criterion leads to a conditional expected value to describe the best estimator. Throughout the rest of this work the least mean squared error will be considered the criterion for the best estimator (21). Therefore, under this assumption the discrete estimator is characterized by the conditional expected value

$$\hat{x}(k/i) = E[x(k)/y_1, y_2, \ldots, y_i]$$

where $x(k)$ is the state vector at time t_k and $\{y_1, y_2, \ldots, y_i\}$ is the set of measurement vectors through time t_i. The vector $\hat{x}(k/i)$ is interpreted as the estimate of x at time t_k given the data up through and including t_i.

Estimation theory itself is divided into three parts depending on the relationship of k and i. When $k = i$ the above expression represents the definition of the filtering algorithm. When $k > i$, the prediction algorithm is defined, and when $k < i$, the smoothing algorithm is defined.

In order to expand the conditional expected value given above there has to be some given relationships between the state vectors and between the state vector and the observables. These relationships are procured by modeling some physical system in which an interest lies. The modeling process consists of taking system parameters and fitting them into a specific format. Note that one may degrade or improve the modeling process by the manner in which the system parameters are shaped into a given format. By changing formats the modeling procedure is changed.

There have been many methods used to write expressions for the above conditional expected value. One undesirable feature of most of these expressions was "growing memory." In other words, all the data, y_i vectors, must be remembered;

and of course as i increased, more and more data accumulates for memorization. When all the data are processed simultaneously, the processing is referred to as batch processing. Kalman arrived at a different scheme (10). He noticed that estimates of the state vector, x, were indeed functions of all the past data. So, Kalman devised a recursive equation, consisting of a previous estimated value of x plus a function of the last data point or variable. This, of course, eliminated the growing memory problems. The recursive aspect provided an extremely convenient procedure for digital computer implementation and is well suited to many on-line applications.

The objective of this research is to explore recursive smoothing algorithms for models involving a delayed state in the measurement equation. Every time the format for modeling a system is changed a new set of equations must be developed. For the standard modeling procedure, as has been mentioned previously, Kalman developed the recursive filtering equations. Also, for this model, much work has been carried out on smoothing equations. This paper develops the recursive smoothing equations for a relative new type of modeling scheme, which was presented by Brown and Hartmann (3). After development, these equations are used in an aided inertial navigation example. The importance of these equations, of course, depends entirely on the importance of the delayed state modeling.

Also in this paper is the extension of the work of Friedland (7) on the recursive filtering equations of Kalman to the smoothing algorithm. First of all, his decoupling ideas are extended to smoothing equations presented by Meditch (13). Next, they are extended to the recursive filtering equations derived by Brown and Hartmann for the delayed state model (3). Finally, his decoupling ideas are applied to the recursive delayed state smoothing equations developed earlier in this work.

At this point, the Kalman filtering problem and solution will be summarized (21). The system considered is composed of two parts. First, the process being estimated is assumed to be described by the state equation

$$x(k+1) = \Phi_{k+1,k}\, x(k) + g_k$$

and the measurement data is related to the state by

$$y_k = H_k\, x(k) + \delta y_k$$

where $\{g_k\}$ and $\{\delta y_k\}$ represent independent white-noise sequences. All capital letters represent matrices and all small case letters represent vectors. The initial state $x(0)$ has a mean value of $\hat{x}(0/-1)$ and is independent of $\{g_k\}$ and $\{\delta y_k\}$. The covariance of the estimation error at $t = 0$ is $P(0/-1)$. The noise sequences are assumed to have zero means and second-order statistics

$$E[\delta y_k \delta y_j'] = R_k \delta_{kj} \qquad E[g_k g_j'] = Q_k \delta_{kj}$$

$$E[\delta y_k g_j'] = 0 \text{ for all } g, j$$

where the prime indicates a transpose and δ_{kj} is the Kronecker delta. An estimate $\hat{x}(k/k)$ of the state $x(k)$ is to be computed from the data $y_1, y_2, \ldots, y_k$ so as to minimize the mean square error in the estimate.

The solution of this recursive, linear, mean-square estimation problem can be determined from the orthogonality principle (15) as well as in many other ways, and is presented below.

$$\hat{x}(k/k-1) = \Phi_{k,k-1}\hat{x}(k-1/k-1)$$

$$\hat{x}(k/k) = \hat{x}(k/k-1) + K_k(y_k - H_k\Phi_{k,k-1}\hat{x}(k-1/k-1)) \ .$$

The gain matrix, K_k, minimizes

$$E[(x(k) - \hat{x}(k/k))' (x(k) - \hat{x}(k/k))]$$

and is

$$K_k = P(k/k-1)H_k'[H_k P(k/k-1)H_k' + R_k]^{-1} \ .$$

The matrix $P(k/k-1)$ is the covariance of the error in the a priori estimate, $\hat{x}(k/k-1)$, and is

$$P(k/k-1) = E[(x(k) - \hat{x}(k/k-1))(x(k) - \hat{x}(k/k-1))']$$

$$= \Phi_{k,k-1} P(k-1/k-1) \Phi'_{k,k-1} + Q_{k-1} \quad .$$

The matrix $P(k/k)$ is the covariance of the error in the a posteriori estimate, $\hat{x}(k/k)$, and is

$$P(k/k) = E[(x(k) - \hat{x}(k/k))(x(k) - \hat{x}(k/k))']$$

$$= [I - K_k H_k] P(k/k-1) \quad .$$

It is evident that the gains, K_k and the covariance matrices, $P(k/k-1)$ and $P(k/k)$, could be computed for all possible k without computing any of the state estimators. In this manner the quality of the modeling process could be observed by comparing the covariance matrices of the estimation errors with the covariance matrices of the true error. The order in which the above equations are used is:

1) compute the optimum gain matrix K_k
2) revise the a priori estimate to get the a posteriori estimate $\hat{x}(k/k)$
3) compute the a posteriori error covariance matrix $P(k/k)$
4) extrapolate ahead the a posteriori estimate and covariance to get $\hat{x}(k+1/k)$ and $P(k+1/k)$.

In many filtering problems the original state assignment is supplemented with additional states due to modeling problems. One such problem is a system of difference equations with other than a white noise driving function. For example, the drift of a gyro in inertial navigation systems is usually modeled as a Markov process.

Some systems are modeled such that there are bias states appearing in the difference equation. Friedland (7) has offered an approach to estimate the states of a system by a linear combination of the bias estimate and the bias-free estimate of the states, which can be computed separately. His technique reduces the size of the system and the computation difficulties due to system size. He also points out that his decoupling in the calculation has its optimal effect when the size of the bias-free state vector is equal to the size of the bias vector. It seems that whatever makes his method advantageous for recursive filtering would also make it desirable for recursive smoothing. In Chapter III the recursive smoothing algorithm presented in Meditch (13) is decoupled by extending the work of Friedland. It should be pointed out that even though smoothing is an off-line operation, i.e., done after the fact, there could be reason to model a system with biases just to take advantage of the decoupled smoothing equations. The trade-off between the computing costs and the degraded models has to be evaluated for each situation.

Brown and Hartmann have suggested a delayed state model to be used for certain aided inertial navigation systems (3).

In such systems position and velocity errors are the states to be estimated and the difference between inertial and non-inertial velocities is the observable. If the measurement noise is white, the variance of the measurement noise is infinite which does not fit the Kalman filter assumptions. If the samples of observable are replaced with average samples, where the average is over some small time Δt, the difficulty is eliminated. This average precipitates the delayed state in the measurement equations. In the article cited the recursive filtering equations for the delayed state model are derived. Chapter IV of this presentation uses the delayed state model and derives the recursive smoothing equations. The algorithm developed is an off-line computation scheme inverting only a matrix of the order of the measurement vector. But, the measurements must be remembered to carry out the scheme. The situations calling or delayed state modeling occurs often enough for important physical systems to justify extending Friedland's decoupling idea. In inertial systems especially, there are a good number of the states in the state assignment which could be modeled as biases. Chapter V presents the decoupled solution of the recursive filtering equations and recursive smoothing equations for the delayed state model.

All the mentioned developments have been for recursive equations and for good reason. The recursive equations are readily implemented on computers, the memory requirements are not as demanding as they are in other methods, and the size of the actual computation can be held down so they involve only system size matrices. Periodically, one must go back and make the comparison between the recursive solution and the batch processing schemes. In Chapter VI two batch processing schemes are explored. The system size is chosen to be consistent with the example to follow in Chapter VII.

The example in Chapter VII is a system presented by Brown (2). The coefficient matrices of the state and delayed state vectors in the measurement equation contain a number of terms which were assigned as states, making the system nonlinear. Because of the nonlinear aspects of this system, the Kalman filter alone will not produce desirable results. Reviewing the techniques available for solving nonlinear equations it was found that in some cases iteration schemes are relied upon for solutions. With this in mind an iteration scheme involving filtering and smoothing will be worked out for this system. The first thing that must be done is that the system is linearized about a nominal value. Then the Kalman filter is used as usual. The linearized model is then corrected by the smoothed information and the Kalman filter is rerun. Because of the short bursts of data obtained in this example the smoothing computation should be easily implemented. The data for this example was obtained from actual test flights, and the aircraft position was determined by precision radar or accurately known check points. Therefore, the system errors, those trying to be estimated, were known. These facts made the above system into an ideal example for testing the smoothing algorithm.

II. SMOOTHING

The main emphasis in this paper is smoothing. This chapter will try to establish a common starting place. As was mentioned earlier, the definition for smoothing is

$$\hat{x}(k/i) = E[x(k)/y_1, y_2, \ldots, y_i]$$

where $i > k$. This definition implies that given the data through the present, the estimate of some state vector in the past is being updated. Using this idea, every estimate from the present time to some fixed or arbitrary one in the past can be updated with each new data point. To become a little more specific, three different types of smoothing are defined. They are:

1) fixed-interval smoothing
2) fixed-point smoothing
3) fixed-lag smoothing.

The fixed point algorithm is characterized by the smoothed estimate,

$$\hat{x}(k/j),\ j = k+1, k+2, \ldots \qquad k = \text{fixed integer.}$$

This is to say that the estimate at some fixed time point, k, is updated with each new piece of data. The fixed-lag algorithm is characterized by the smoothed estimate,

$$\hat{x}(k/k+N),\ k = 0, 1, \ldots \qquad N = \text{fixed positive integer.}$$

This is to say that the estimate at some fixed interval from the last data point is updated with each new piece of data. The fixed interval algorithm is characterized by the smoothed estimate, $\hat{x}(k/N)$, $k = 0,1,\ldots,N-1$, N = fixed positive integer. Which says that all the past estimates are updated with each new piece of data.

The main difference in the three smoothing algorithms is the manner in which the data after the time point, k, is used. Therefore, all one has to do is permute the developmental philosophy of one type of smoothing to precipitate another type of smoothing. With this in mind, the emphasis from here on will be placed on the fixed interval algorithm.

There are different mathematical schemes for the fixed interval smoothing problems. A standard scheme is presented by Meditch (13). He says that

$$\hat{x}(k/n) = E[x(n)/y_1, y_2, \ldots y_{n-1}, \tilde{y}(n/n-1)]$$

where

$$\tilde{y}(n/n-1) = y_n - \hat{y}(n/n-1)$$

and

$$\hat{y}(n/n-1) = E[y_n/y_1, y_2, \ldots, y_{n-1}] \quad .$$

Since it is assumed that the random variables are Gaussian, the above is the same as the general definition. With $E[x(k)] = 0$ then

$$\hat{x}(k/n) = E[x(k)/y_1, y_2, \ldots, y_{n-1}] + E[x(k)/\tilde{y}(n/n-1)]$$

$$= \hat{x}(k/n-1) + E[x(k)\tilde{y}(n/n-1)']E[\tilde{y}(n/n-1)\tilde{y}'(n/n-1)]^{-1}$$

$$\tilde{y}(n/n-1) \ .$$

Starting with $k = n-1$ and evaluating the above expression Meditch obtains the one step back smoothing equation. Continuing on with $k = n-2, n-3, \ldots$ and with the use of induction he obtains the general fixed interval smoothing equations.

$$\hat{x}(k/N) = \hat{x}(k/k) + A(k)[\hat{x}(k+1/N) - \hat{x}(k+1/k)]$$

$$A(k) = P(k/k)\Phi'(k+1,k)P^{-1}(k+1/k)$$

$$P(k/N) = P(k/k) + A(k)[P(k+1/N) - P(k+1/k)]A'(k) \ .$$

This is a recursive solution to the fixed interval smoothing problem. The main disadvantage computationally to this scheme is that the a priori covariance matrix has to be inverted. It should also be noted, however, that the covariance matrix for the smoothed estimate is not needed to compute the smoothed estimate. This fact is true for most smoothing schemes.

Another scheme used in the fixed interval smoothing problem is presented by Cox (5). He starts by minimizing a cost function. In doing this, Cox ends up with a two point boundary value problem to solve. From this TPBVP he obtains the Kalman filtering equations plus the fixed interval smoothing equations.

$$\hat{x}(k/N) = \hat{x}(k/k) + P(k/k)\Phi'(k+1,k)\lambda(k)$$

$$\lambda(k-1) = \Phi'(k+1,k)\lambda(k) + M'(k)R^{-1}(k)[y(k) - M(k)\hat{x}(k/N)]$$

$$\lambda(N) = 0 \quad .$$

This is also a recursive solution to the fixed interval smoothing problem. In this scheme a matrix inverse operation is also required. But, the dimension of the matrix being inverted here is a m x m matrix where m is the size of the measurement vector. In many estimation problems the size of the measurement vector is less than that of the state vector due to the number of augment states. Therefore, inverting the R matrix is more desirable than inverting the P(k+1/k) matrix. The trade-off here is that now the measurements must be remembered.

Another scheme is presented by Rauch (19). The differences between this scheme and the one above are many. Rauch's equations have a growing sumation. His equations are best used when finding the estimates at one time point; and they propagate in the forward direction.

$$\hat{x}(k/N) = \hat{x}(k/k) + \sum_{i=k+1}^{n} K(k,i)[y(i) - M(i)\hat{x}(i/i-1)]$$

$$K(k,i) = \left\{P(k/k)\Phi'(i,k) - \sum_{j=k+1}^{i-1} K(k,j)M(j)P(j/j-1)\Phi'(i,j)\right\}$$

$$\cdot\ M'(i)[M(i)P(i/i-1)M(i) + R(N)]^{-1}$$

$$i = k+1, k+2, \ldots, N \quad .$$

It is true that the K(k,i) is computed recursively. But, the sumations make the computations in arriving at the smoothed estimates grow with the number of steps taken in the past.

These are just three schemes that have been arrived at for the fixed interval smoothing problem. They all have some common features. The computations in two of the above schemes are done such that the smoothed estimates are obtained by going from the last data point backwards in time to an earlier data point. This feature is almost expected, but some of the outcomes of it causes some problems. For example, in each method the filtering covariance matrices must be incorporated in the backwards computations. This implies either a backwards recursive equation is needed or that the covariance matrices must be memorized. In large systems the memorization would be near impossible. The backwards recursive equation for the covariance equations requires an inverse of the transition matrix, which is something that should be avoided.

It seems that at the present time we have a choice of schemes for the fixed interval smoothing problem. The only difficulty that can be seen is implementing these schemes. These problems could often be attacked individually and overcome (8).

III. A SMOOTHING ALGORITHM USING FRIEDLAND'S DECOUPLING SCHEME

As has been mentioned previously, smoothing should be considered an off-line computation. Usually it will be carried out after the physical operation or experiment has been completed. This allows a researcher to use devices with much more accurate computations and does not have to worry about storage space and time. Therefore, there would be no justification in modeling Markov states as biases just to be able to use a decoupled smoothing scheme. But, if the true model of a system contains bias states the decoupling scheme should present a little savings in computation.

In this chapter a decoupling of a smoothing scheme already in existence will be presented. The equations that will be decoupled are presented by Meditch (13). But first Friedland's paper will be outlined. Then using the same technique the smoothing equations will be derived.

A. Treatment of Bias Variables

The problem of estimating the state x of a linear system in the presence of a constant but unknown bias vector b or of a Markov state with a very long time constant, which will appear like a constant during the time the filter is being used, is considered. This bias state influences the dynamics and/or the observations. It was shown by Friedland (7) that the optimum estimate $\hat{x}$ of the state can be expressed as

$$\hat{x} = \tilde{x} + V_x\hat{b}$$

where $\tilde{x}$ is the bias-free estimate, computed as if no bias were present, $\hat{b}$ is the optimum estimate of the bias, and V_x is a matrix which can be interpreted in the scalar case as the ratio of the covariance of $\tilde{x}$ and $\hat{b}$ to the variance of $\hat{b}$. The computation of the optimum estimate $\hat{x}$ is effectively decoupled from the estimate of the bias $\hat{b}$, except for the final addition.

Friedland's notation is as follows:

$x(k)$ - original or physical process state (at k^{th} observation instant) n components

$b(k)$ - bias vector (r components)

g_k - process noise vector, with $E[g_k g_n'] = Q_k \delta_{kn}$

δy_k - observation noise vector, with $E[\delta y_k \delta y_n'] = R_k \delta_{kn}$.

Assuming that g_k and δy_k are independent for all k and n the dynamic equations can be written.

$$x(k) = A_{k-1}x(k-1) + B_{k-1}b(k-1) + g_{k-1}$$

$$b(k+1) = b(k)$$

$$y(k) = H_k x(k) + C_k b(k) + \delta y_k \quad \text{(measurement equation)}.$$

Most people when attacking a problem of this sort augment the state vector to include the bias terms as states. Then using the new dynamical equations proceed with the Kalman filter equations. So, a vector Z(k) will be defined as

$$Z(k) = \begin{bmatrix} x(k) \\ --- \\ b(k) \end{bmatrix} .$$

Now the dynamical equations may be rewritten as

$$Z(k) = F_{k-1}Z(k-1) + G\, g_{k-1}$$

$$y(k) = L_k Z(k) + \delta y_k$$

where

$$F_{k-1} = \left[\begin{array}{c|c} A_{k-1} & B_{k-1} \\ \hline 0 & I \end{array}\right], \quad G = \begin{bmatrix} I \\ - \\ 0 \end{bmatrix}, \quad L_k = [H_k \mid C_k] .$$

The Kalman filter equations can be written for the augmented dynamical equations

$$\hat{Z}(k/k) = F_{k-1}\hat{Z}(k-1/k-1) + K(k)[y(k) - L_k F_{k-1}\hat{Z}(k-1/k-1)]$$

where

$$K(k) = P(k/k-1)L_k'[L_k P(k/k-1)L_k' + R_k]^{-1}$$

$$= P(k/k)L' R_k^{-1}$$

$$P(k/k) = [I - K(k)L_k]P(k/k-1)$$

$$P(k+1/k) = F_k P(k/k)F_k' + GQ_k G' \quad . \tag{1}$$

Friedland (7) now defines $\tilde{P}(k/k-1)$ as the covariance which is a solution to Equation 1 for the initial conditions

$$P(0/-1) = \left[\begin{array}{c|c} \tilde{P}_x(0/-1) & 0 \\ \hline 0 & 0 \end{array}\right] .$$

He then shows that the solution to Equation 1 can be written as

$$P(k/k-1) = \tilde{P}(k/k-1) + U(k)M(k)U'(k)$$

where $M(k)$ is an $r \times r$ symmetric matrix and $U(k)$ is an $n \times r$ matrix, and

$$U(k+1) = F_k[I-\tilde{P}(k/k-1)L_k'(L_k\tilde{P}(k/k-1)L_k' + R_k)^{-1}L_k]U(k)$$

$$= F_kV(k)$$

$$M(k+1) = M(k) - M(k)U'(k)L_k'[L_k\tilde{P}(k/k-1)L_k' + R_k$$

$$+ L_kU(k)M(k)U'(k)L_k']^{-1} L_kU(k)M(k) .$$

The last term of the solution for Equation 1 is due to the fact that the cross term P_{xb} and the lower diagonal term P_b of the partitioned $P(0/-1)$ matrix are not zero, i.e.,

$$P_{xb}(0/-1) \neq 0$$

$$P_b(0/1) \neq 0 .$$

The above equation leads to the following

$$P(k/k) = \tilde{P}(k/k) + V(k)M(k+1)V'(k)$$

$$V(k) = [I - \tilde{K}(k)L_k]U(k) \quad .$$

By writing the component equation for the bias and the state forms, it can be shown that from the above set of matrix equations

$$U_b(k) = V_b(k) = U_b(0) = \text{constant} \quad .$$

This fact and the assumption of x and b being independent at $k = 0$ leads Friedland to choose

$$U_b(0) = I \text{ and } U_x(0) = 0$$

which precipitates the following equations

$$V_x(k) = U_x(k) - \tilde{K}_x(k)[H_k U_x(k) + C_k] = U_x(k) - \tilde{K}_x(k)S(k)$$

$$U_x(k+1) = A_k V_x(k) + B_k$$

$$P_x(k/k-1) = \tilde{P}_x(k/k-1) + U_x(k)M(k)U'_x(k)$$

$$P_{xb}(k/k-1) = U_x(k)M(k)$$

$$P_b(k/k-1) = M(k)$$

$$P_x(k/k) = \tilde{P}_x(k/k) + V_x(k)M(k+1)V'_x(k)$$

$$P_{xb}(k/k) = V_x(k)M(k+1)$$

$$P_b(k/k) = M(k+1)$$

$$M(k+1) = M(k) - M(k)S'(k)[H_k\tilde{P}(k/k-1)H_k'+R_k+S(k)M(k)S'(k)]^{-1}$$

$$\cdot\ S(k)M(k)$$

$$K_x(k) = \tilde{K}_x(k) + V_x(k)K_b(k)$$

$$K_b(k) = M(k+1)[V_x'(k)H_k' + C_k']\ R_k^{-1} \quad .$$

Using these equations and splitting the augmented filter equations into components, Friedland arrives at the following decoupled equations

$$\hat{b}(k/k) = \hat{b}(k-1/k-1) + K_b[\gamma_k - S(k)\hat{b}(k-1/k-1)]$$

$$\hat{x}(k/k) = A_{k-1}\hat{x}(k-1/k-1) + \tilde{K}_x(k)\gamma_k$$

$$\gamma_k = y(k) - H_kA_{k-1}\hat{x}(k-1/k-1)$$

which can be arranged to prove the following expression

$$\hat{x}(k/k) = \tilde{x}(k/k) + V_x(k)\hat{b}(k/k) \quad .$$

Although these results have been derived from an assumption of constant bias, they can be readily extended to the deterministic process, i.e.,

$$b_{n+1} = W_nb_n$$

which can also be written as

$$b_{n+1} = W^*_{n+1}b_o$$

where

$$W^*_{n+1} = \prod_{i=0}^{n} W_i$$

and W_i is the transition matrix which transforms b_i to b_{i+1}. Therefore, W^*_{n+1} is the transition matrix which transforms b_o to b_{n+1}. Inherent in the deterministic process problem is that the dynamics of the bias are known, but not the initial conditions. Therefore, with the above observation the time-varying bias problem can be reduced to the constant bias problem by redefining the sum of the coefficients as indicated below.

$$x(k) = A_{k-1}x(k-1) + B_{k-1}W^*_{k-1}b_o(k-1) + g_{k-1}$$

$$b_o(k+1) = b_o(k)$$

$$y(k) = H_k(k) + C_kW^*_kb_o(k) + \delta y_k \quad .$$

Therefore, if in all the previous equations the coefficients B_k and C_k are changed to read $B_kW^*_k$ and $C_kW^*_k$ the results will be identical.

B. Smoothing Algorithm

The idea of decoupling the estimation equations will now be extended to the smoothing equations. By the use of induction, the smoothing equations presented by Meditch (13) will be decoupled. The one step back smoothing equations will be decoupled first.

$$\hat{Z}(k/k+1) = \hat{Z}(k/k) + D(k)[\hat{Z}(k+1/k+1) - F_k\hat{Z}(k/k)]$$

$$D(k) = P(k/k)F_k'P^{-1}(k+1/k)$$

where

$$Z(k/k) = \begin{bmatrix} \hat{X}(k/k) \\ \hline \hat{b}(k/k) \end{bmatrix} .$$

In order to decouple the above equations the smoothing gain matrix, D(k), must be partitioned into components.

$$D(k) = \left[\begin{array}{c|c} D_x(k) & D_{xb}(k) \\ \hline D_{bx}(k) & D_b(k) \end{array}\right]$$

$$= \left[\begin{array}{c|c} P_x(k/k) & P_{xb}(k/k) \\ \hline P_{bx}(k/k) & P_b(k/k) \end{array}\right] \left[\begin{array}{c|c} A_k' & 0 \\ \hline B_k' & I \end{array}\right] \left[\begin{array}{c|c} R_{11} & R_{12} \\ \hline R_{21} & R_{22} \end{array}\right]$$

where R_{ij} are the partitions of the matrix $P^{-1}(k+1/k)$ which are

$$R_{11}^{-1} = P_x(k+1/k) - P_{xb}(k+1/k)P_b^{-1}(k+1/k)P_{x'b}(k+1/k)$$

$$R_{22}^{-1} = P_b(k+1/k) - P_{xb}'(k+1/k)P_x^{-1}(k+1/k)P_{xb}(k+1/k)$$

$$R_{12} = -P_x^{-1}(k+1/k)P_{xb}(k+1/k)R_{22}$$

$$R_{21} = -P_b^{-1}(k+1/k)P_{x'b}(k+1/k)R_{11} .$$

Using the equations Friedland presents for the a priori covariance matrices the above equations can be rewritten.

$$R_{11}^{-1} = \tilde{P}_x(k+1/k) + U_x(k+1)M(k+1)U_x'(k+1) - U_x(k+1)M(k+1)$$

$$\cdot\ M^{-1}(k+1)M(k+1)U_x'(k+1) = P_x(k+1/k)$$

$$R_{21} = -M^{-1}(k+1)M(k+1)U_x'(k+1)\tilde{P}_x^{-1}(k+1/k)$$

$$= -U_x'(k+1)\tilde{P}_x^{-1}(k+1/k) \quad .$$

Since $P^{-1}(k+1/k)$ is a symmetric matrix

$$R_{12} = R_{21}' = -\tilde{P}_x^{-1}(k+1/k)U_x(k+1) \quad .$$

Using the matrix inverse identity (20) R_{22} reduces as follows

$$R_{22} = P_b^{-1}(k+1/k) - P_b^{-1}(k+1/k)P_{xb}'(k+1/k)[-P_x(k+1/k)$$

$$+ P_{xb}(k+1/k)P_b^{-1}(k+1/k)P_{xb}'(k+1/k)]^{-1}P_{xb}(k+1/k)$$

$$\cdot\ P_b^{-1}(k+1/k)$$

$$= M^{-1}(k+1)M^{-1}(k+1) + U_x'(k+1)P_x(k+1/k)U_x(k+1) \quad .$$

Now the components for the gain matrix are

$$D_x(k) = P_x(k/k)A_k'R_{11} + P_{xb}(k/k)[B_k'R_{11} + R_{21}]$$

$$= \tilde{P}_x(k/k)A_k'\tilde{P}_x^{-1}(k+1/k) \equiv \tilde{D}_x(k)$$

$$D_{bx}(k) = P'_{xb}(k/k)A'_kR_{11} + P_b(k/k)[B'_kR_{11} + R_{21}]$$

$$= 0$$

$$D_b(k) = P'_{xb}(k/k)A'_kR_{12} + P_b(k/k)[B'_kR_{12} + R_{22}]$$

$$= I$$

$$D_{xb}(k) = P_x(k/k)A'_kR_{12} + P_{xb}(k/k)[B'_kR_{12} + R_{22}]$$

$$= V_x(k) - \tilde{D}_x(k)U_x(k+1) \quad .$$

With the gain equation partitioned as above, the augmented smoothed equations can be separated into the following

$$\hat{x}(k/k+1) = \hat{x}(k/k) + D_x(k)[\hat{x}(k+1/k+1) - A_k\hat{x}(k/k) - B_k\hat{b}(k/k)]$$

$$+ D_{xb}(k)[\hat{b}(k+1/k+1) - \hat{b}(k/k)]$$

and

$$\hat{b}(k/k+1) = \hat{b}(k/k) + D_{bk}(k)[\hat{x}(k+1/k+1) - A_kx(k/k) - B_k\hat{b}(k/k)]$$

$$+ D_b(k)[\hat{b}(k+1/k+1) - \hat{b}(k/k)] \quad .$$

Using the result

$$\hat{x}(k/k) = \tilde{x}(k/k) + V_x(k)\hat{b}(k/k)$$

and the derived expression for the gain components, the above equation reduces to

$$\hat{x}(k/k+1) = \tilde{x}(k/k) + \tilde{D}_x(k)[\tilde{x}(k+1/k+1) - A_k\tilde{x}(k/k)]$$

$$+ \ [V_x(k) - V_x(k) + \tilde{D}_x(k)U_x(k+1) - \tilde{D}_x(k)U_x(k+1)]\hat{b}(k/k)$$

$$+ \ [\tilde{D}_x(k)V_x(k+1) + V_x(k) - \tilde{D}_x(k)U_x(k+1)]\hat{b}(k+1/k+1)$$

$$= \tilde{x}(/k+1)+[\tilde{D}_x(k)V_x(k+1)+V_x(k)-\tilde{D}_x(k)U_x(k+1)]\hat{b}(k+1/k+1)$$

$$\hat{b}(k/k+1) = \hat{b}(k/k) + \hat{b}(k+1/k+1) - \hat{b}(k/k) = \hat{b}(k+1/k+1) \quad .$$

Therefore, the decoupled one step back smoothing equations are

$$\hat{x}(k/k+1) = \tilde{x}(k/k+1) + T(k/k+1)\hat{b}(k/k+1)$$

$$T(k/k+1) = \tilde{D}_x(k)V_x(k+1) + V_x(k) - \tilde{D}_x(k)U_x(k+1) \quad .$$

Now proceeding with the two step back smoothing equations and the same gain components, the components of the augmented equation are

$$\hat{x}(k/k+2) = \hat{x}(k/k) + D_x(k)[\hat{x}(k+1/k+2) - A_k\hat{x}(k/k) - B_k\hat{b}(k/k)]$$

$$+ \ D_{xb}(k)[\hat{b}(k+1/k+2) - \hat{b}(k/k)]$$

and

$$\hat{b}(k/k+2) = \hat{b}(k/k) + D_{bx}[\hat{x}(k+1/k+2) - A_k\hat{x}(k/k) - B_k\hat{b}(k/k)]$$

$$+ \ D_b[\hat{b}(k+1/k+2) - \hat{b}(k/k)] \quad .$$

By using the equation derived for the one step back smoothing problem, the two step equations can be reduced in a similar manner as those in the one step situation.

$$\hat{x}(k/k+2) = \tilde{x}(k/k) + \tilde{D}_x(k)[\tilde{x}(k+1/k+2) - A_k\tilde{x}(k/k)]$$

$$+ \ [V_x(k) - \tilde{D}_k(k)U_x(k+1) - V_x(k) + \tilde{D}_x(k)U_x(k+1)]\hat{b}(k/k)$$

$$+ \ [D_x(k)T(k+1/k+2) + V_x(k) - \tilde{D}_x(k)U_x(k+1)]\hat{b}(k+1/k+2)$$

$$= \tilde{x}(k/k+2) + [\tilde{D}_x(k)T(k+1/k+2) + V_x(k) - \tilde{D}_x(k)U_x(k+1)]$$

$$\cdot \ \hat{b}(k+1/k+2)$$

$\hat{b}(k/k+2) = \hat{b}(k/k) + \hat{b}(k+1/k+2) - \hat{b}(k$ [illegible]$) = \hat{b}(k+1/k+2)$.

Therefore, the decoupled two step [illegible] smoothing equations are

$$\hat{x}(k/k+2) = \tilde{x}(k/k+2) + T(k/k+2)\hat{b}(k/k+2)$$

$$T(k/k+2) = \tilde{D}_x(k)T(k+1/k+2) + V_x(k) - \tilde{D}_x(k)U_x(k+1) \ .$$

Proceeding in the same manner, the $N-k^{th}$ step smoothing equations or the fixed interval smoothing equations are found. In carrying out the same manipulations as was indicated in deriving the one and two step equations one arrives at the following

$$\hat{x}(k/N) = \tilde{x}(k/N) + T(k/N)\hat{b}(k/N)$$

$$T(k/N) = \tilde{D}_x(k)T(k+1/N) + V_x(k) - \tilde{D}_x(k)U_x(k+1) \ .$$

The above equation represents a recursive technique to handle the fixed interval smoothing problem when there are bias states in the system model. It should be noted that since

$$\hat{b}(N/N) = \hat{b}(N-1/N) = \ldots = \hat{b}(k/N)$$

that $T(k/N)$ and $\tilde{x}(k/N)$ are the only matrices which have to be recalculated for each step. The flow chart in Figure 1 indicates the calculation of the smoothed estimates of the state variables.

One has the usual memory problem with the regular smoother plus he has to remember the extra matrices $V_x(k)$ and $U_x(k+1)$ from the filtering procedure to form the correct smoothed estimate. The saving comes from having to invert a smaller matrix to form $\tilde{D}_x(k)$, the usual gain matrix.

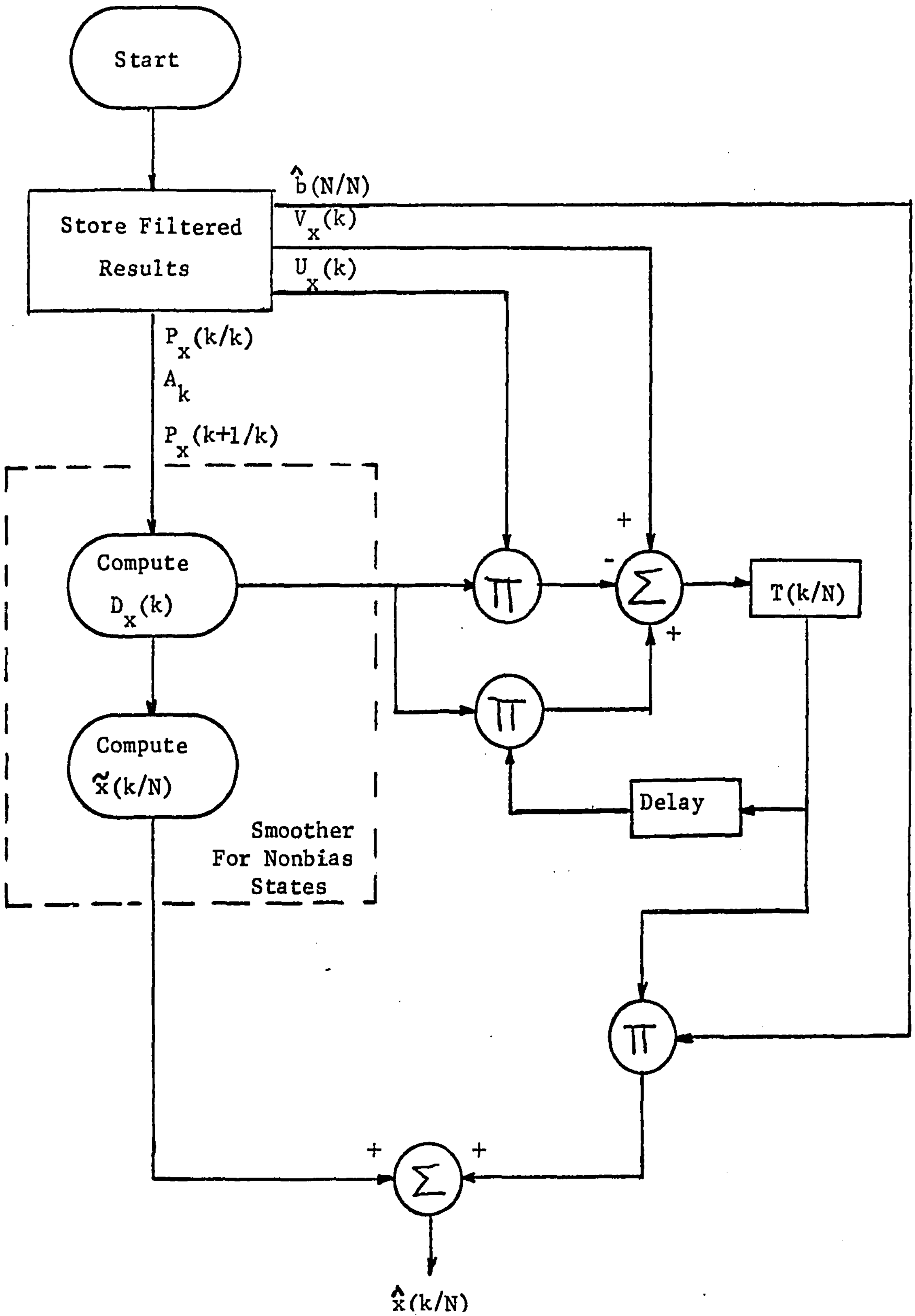

Figure 1. Flow chart for decoupled smoothing equations

IV. SMOOTHING EQUATIONS FOR DELAYED STATE MODEL

There are physical situations which do not fit the present model for the Kalman filter. One such situation could be a certain aided inertial systems. Here a noisy measurement of velocity is obtained by comparing the inertial and noninertial velocity. If a large amount of high frequency noise is present in the measurement it may be better to use an average measurement over the recursive interval rather than the measurement. The idea of an average indicates an integral, which points to two states, one being at the present time and one being at the previous time, in the measurement equation.

$$y_k = M_k x(k) + N_k x(k-1) + \delta y_k \quad .$$

With the augmented model the filter equations must be rederived. The development presented of these equations will follow that suggested by Brown and Hartmann (3). Following these filtering equations, the fixed interval smoothing equation will be developed. The development will follow the method presented by Meditch (13). The order of the inverse in the algorithm will be the same as the dimension of the measurement, thus implementing some of the ideas presented by Cox (5) and Rauch (19). The end result is a very interesting algorithm similar to that of Cox.

A. Delayed State Kalman Filter

The system model will include the delayed state in the measurement equation, i.e.,

$$x(k+1) = \Phi(k+1,k)x(k) + g_k$$

$$y_k = M_k x(k) + N_k x(k-1) + \delta y_k \quad .$$

The first thing that should be done is to twist the new system model into the usual Kalman format. Therefore, the difference equation will be

$$\begin{bmatrix} x_{k+1} \\ \hline x_k \end{bmatrix} = F_k \begin{bmatrix} x_k \\ \hline x_{k-1} \end{bmatrix} + \begin{bmatrix} g_k \\ \hline 0 \end{bmatrix}$$

where

$$F_k = \left[\begin{array}{c|c} \Phi(k+1,k) & 0 \\ \hline I & 0 \end{array}\right] .$$

and the measurement equation will be

$$y_k = L_k \begin{bmatrix} x_k \\ \hline x_{k-1} \end{bmatrix} + \delta y_k$$

where

$$L_k = [M_k \mid N_k] \quad .$$

With the above equations the filter equation can be written by inspection

$$\begin{bmatrix} \hat{x}(k+1/k+1) \\ \hline \hat{x}(k/k+1) \end{bmatrix} = F_k \begin{bmatrix} \hat{x}(k/k) \\ \hline \hat{x}(k-1/k) \end{bmatrix} + B(k+1)\{y_{k+1} - L_{k+1} F_k \begin{bmatrix} \hat{x}(k/k) \\ \hline \hat{x}(k-1/k) \end{bmatrix}\}$$

Following the method used by Sorenson (21) the gain matrix is found to be

$$B(k+1) = \begin{bmatrix} b_1(k+1) \\ \hline b_2(k+1) \end{bmatrix} = \begin{bmatrix} P(k+1/k)\, M'_{k+1} + \Phi(k+1,k)\, P(k/k)\, N'_{k+1} \\ \hline P(k/k)\, \Phi(k+1/k)\, M'_{k+1} + P(k/k) N'_{k+1} \end{bmatrix} C^{-1}_{k+1}$$

where

$$C_{k+1} = M_{k+1}P(k+1/k)M'_{k+1} + M_{k+1}\Phi(k+1,k)P(k/k)N'_{k+1}$$
$$+ N_{k+1}P(k/k)\Phi'(k+1,k)M'_{k+1} + N_{k+1}P(k/k)N'_{k+1} + R_{k+1}$$

and

$$P(k+1/k+1) = P(k+1/k) - b_1(k+1)C_{k+1}b'_1(k+1)$$

$$P(k+1/k) = \Phi(k+1/k)P(k/k)\Phi'(k+1,k) + Q_k \quad .$$

Now, separating the filter equation into components, two equations can be written. The first is the filter equation for the delayed state model and the second is the one step back smoothing equation.

$$\hat{x}(k+1/k+1) = \Phi(k+1/k)\hat{x}(k/k) + b_1(k+1)[y_{k+1} - M_{k+1}\Phi(k+1/k)\hat{x}(k/k)$$
$$- N_{k+1}\hat{x}(k/k)]$$

$$\hat{x}(k/k+1) = \hat{x}(k/k) + b_2(k+1)[y_{k+1} - M_{k+1}\Phi(k+1,k)\hat{x}(k/k) - N_{k+1}\hat{x}(k/k)] \quad .$$

Notice, by proceeding in the above fashion more was obtained than was bargained for. This should indicate a method for deriving the multiple step smoothing equation. By adding the appropriate unitary matrices to the F_k matrix and the $x_{k-1}, x_{k-2}, \ldots$, etc. to the state vector, the recursive estimation scheme illustrated above should yield the solution to the two-step back smoothing problem, the three-step back smoothing problem, and so forth, along with what has been presented. However, obtaining these additional solutions could be quite time consuming and messy.

B. Fixed Interval Smoothing Equation for the Delayed State Kalman Filter

Instead of proceeding as was indicated in the last section, the development of the desired smoothing equations will follow the development presented by Meditch (13). The philosophy being used is to start at the final time point of the existing data and develop first the one-step back smoothing equations, then the two-step back smoothing equations, and then by induction the general equations.

The one-step back equation was derived when the delayed state filtering equation was obtained in the previous section. This equation will now be rewritten so that the development will start from the last data point at t_n.

$$\hat{x}(n-1/n) = \hat{x}(n-1/n-1) - b_2(n-1)\ \ \tilde{y}(n/n-1)$$

where

$$\tilde{y}(n/n-1) = y_n - [M_n \Phi_{n,n-1} + N_n]\hat{x}(n-1/n-1)$$

$$b_2(n-1) = P(n-1/n-1)[M_n \Phi_{n,n-1} + N_n]' C_n^{-1}$$

$$C_n = M_n P(n/n-1) M_n' + M_n \Phi_{n,n-1} P(n-1/n-1) N_n'$$

$$+ N_n P(n-1/n-1) \Phi_{n,n-1}' M_n' + N_n P(n-1/n-1) N_n' + R_n \ .$$

Now the two-step back problem will try to be solved, that is, find an expression for $\hat{x}(n-2/n)$ given $\{y_1, y_2, \ldots y_{n-1}, y_n\}$. By definition then

$$\hat{x}(n-2/n) = E[x(n-2)/y_1, \ldots, y_{n-1}, \tilde{y}(n/n-1)]$$

where $\tilde{y}(n/n-1)$ is independent of the set of measurements $\{y_1, \ldots, y_{n-1}\}$. Thus, since the random variables are Gaussian

$$\hat{x}(n-2/n) = E[x(n-2)/y_1, \ldots, y_{n-1}] + E[x(n-2)/\tilde{y}(n/n-1)] \ .$$

The first term above is just the definition of the one-step smoothing problem which has already been solved. The second term is evaluated by using theorem 9.11 from (14), which means

$$\hat{x}(n-2/n) = \hat{x}(n-2/n-1) + P_{x\tilde{y}} P_{\tilde{y}\tilde{y}}^{-1} \tilde{y}(n/n-1)$$

where

$$\tilde{y}(n/n-1) = y_n - \hat{y}(n/n-1)$$

$$= (M_n \Phi_{n,n-1} + N_n)\tilde{x}(n-1) + M_n g_{n-1} + \delta y_n$$

$$P_{\tilde{y}\tilde{y}} = E[\tilde{y}(n/n-1)\tilde{y}'(n/n-1)] = C_n$$

$$\tilde{x}(n-1) = x(n-1) - \hat{x}(n-1/n-1)$$

$$= \{\Phi_{n-1,n-2} - b_1(n-1)[M_{n-1}\Phi_{n-1,n-2} + N_{n-1}]\}\tilde{x}(n-2)$$

$$+ [I - b_1(n-1)M_{n-1}]g_{n-2} - b_1(n-1)\delta y_{n-1}$$

$$Px\tilde{y} = E[x(n-2)\tilde{y}'(n/n-1)] = E\{[\tilde{x}(n-2) - \hat{x}(n-2/n-2)]\tilde{x}'(n-1)$$

$$\cdot [M_n\Phi_{n,n-1} + N_n]'\} = P(n-2/n-2)\{\Phi_{n-1,n-2} - b_1(n-1)$$

$$[M_{n-1}\Phi_{n-1,n-2} + N_{n-1}]\}' \; [M_n\Phi_{n,n-1} + N_n]' \quad .$$

Therefore,

$$\hat{x}(n-2/n) = \hat{x}(n-2/n-2) + P(n-2/n-2)[M_{n-1}\Phi_{n-1,n-2} + N_{n-1}]'$$

$$\cdot C_{n-1}^{-1}\tilde{y}(n-1/n-2) + P(n-2/n-2)\{\Phi_{n-1,n-2} - b_1(n-1)$$

$$\cdot [M_{n-1}\Phi_{n-1,n-2} + N_{n-1}]\}'[M_n\Phi_{n,n-1} + N_n]'$$

$$\cdot C_n^{-1}\tilde{y}(n/n-1) \quad .$$

Proceeding as above, a solution to the three-step back smoothing problem will be found. By definition

$$\tilde{x}(n-3/n) = E[x(n-3)/y_1, \ldots, y_{n-1}, \tilde{y}(n/n-1)]$$

$$= E[x(n-3)/y_1, \ldots, y_{n-1}] + E[x(n-3)/\tilde{y}(n/n-1)]$$

$$= \hat{x}(n-3/n-2) + Px\tilde{y} \; P\tilde{y}\tilde{y}^{-1} \; \tilde{y}(n/n-1)$$

where

$$\tilde{x}(n-1) = x(n-1) - \hat{x}(n-1/n-1)$$

$$= \{\Phi_{n-1,n-2} - b_1(n-2)[M_{n-1}\Phi_{n-1,n-2} + N_{n-1}]\}\{\Phi_{n-2,n-3} - b_1(n-2)$$

$$\cdot[M_{n-2}\Phi_{n-2,n-3} + N_{n-2}]\}\tilde{x}(n-3) + \{\Phi_{n-1,n-2} - b_1(n-1)$$

$$\cdot[M_{n-1}\Phi_{n-1,n-2} + N_{n-1}]\}\{I - b_1(n-2)M_{n-2}\}g_{n-3}$$

$$+ [I - b_1(n-1)M_{n-1}]g_{n-2} - \{\Phi_{n-1,n-2} - b_1(n-1)$$

$$\cdot[M_{n-1}\Phi_{n-1,n-2} + N_{n-1}]\}b_1(n-2)\delta y_{n-2} - b_1(n-1)\delta y_{n-1}$$

$$Px\tilde{y} = E[x(n-3)\tilde{y}'(n/n-1)] = E\{(\tilde{x}(n-3) + \hat{x}(n-3/n-3))\tilde{x}'(n-1)$$

$$\cdot[M_n\Phi_{n,n-1} + N_n]'\} = P(n-3/n-3)[\{\Phi_{n-1,n-2} - b_1(n-1)$$

$$\cdot[M_{n-1}\Phi_{n-1,n-2} + N_{n-1}]\}\{\Phi_{n-2,n-3} - b_1(n-2)$$

$$\cdot[M_{n-2}\Phi_{n-2,n-3} + N_{n-2}]\}'[M_n\Phi_{n,n-1} + N_n]' \quad .$$

Therefore,

$$\hat{x}(n-3/n) = \hat{x}(n-3/n-3) + P(n-3/n-3)[M_{n-2}\Phi_{n-2,n-1} + N_{n-2}]'$$

$$\cdot C_{n-2}^{-1}\tilde{y}(n-2/n-3) + P(n-3/n-3)\{\Phi_{n-2,n-3} - b_1(n-2)$$

$$\cdot[M_{n-2}\Phi_{n-2,n-3} + N_{n-2}]\}'[M_{n-1}\Phi_{n-1,n-2} + N_{n-1}]'$$

$$\cdot C_{n-1}^{-1}\tilde{y}(n-1/n-2) + P(n-3/n-3)\{\Phi_{n-2,n-3} - b_1(n-2)$$

$$\cdot[M_{n-2}\Phi_{n-2,n-3} + N_{n-2}]\}'\{\Phi_{n-1,n-2} - b_1(n-1)$$

$$\cdot [M_{n-1}\Phi_{n-1,n-2} + N_{n-1}]\}'[M_n\Phi_{n,n-1} + N_n]'C_n^{-1}\tilde{y}(n/n-1)$$

This can be simplified as follows

$$\hat{x}(n-3/n) = \hat{x}(n-3/n-3) + P(n-3/n-3)\sum_{i=n-2}^{n} d(n-3,i)[M_i\Phi_{i,i-1} + N_i]'$$

$$\cdot C_i^{-1}\tilde{y}(i/i-1)$$

where

$$d(n-3,n-2) = I$$

$$d(n-3,i) = \prod_{j=n-2}^{i-1}\{\Phi_{j,j-1} - b_1(j)[M_j\Phi_{j,j-1} + N_j]\}'$$

$$\text{for } i=n-1,n \quad .$$

Now proceeding with the induction, it is assumed that

$$\hat{x}(k+1/n) = \hat{x}(k+1/k+1) + P(k+1/k+1)\sum_{i=k+2}^{n} d(k+1,i)[M_i\Phi_{i,i-1} + N_i]'$$

$$\cdot C_i^{-1}\tilde{y}(i/i-1)$$

where

$$d(k+1,k+2) = I$$

$$d(k+1,i) = d(k+1,i-1)\{\Phi_{i-1,i-2} - b(i-1)[M_{i-1}\Phi_{i-1,i-2}$$

$$+ N_{i-1}]\}' \quad \text{for } i=k+3,\ldots,n \quad .$$

Using the assumption and proceeding in the same manner as the above, the $n-k^{th}$ step smoothing problem will be solved. By definition

$$\hat{x}(k/n) = \hat{x}(k/n-1) + Px\tilde{y}\, P\tilde{y}\tilde{y}^{-1}\, \tilde{y}(n/n-1)$$

since

$$\tilde{x}(n-1) = \{\Phi_{n-1,n-2} - b_1(n-1)[M_{n-1}\Phi_{n-1,n-2} + N_{n-1}]\}'$$

$$\cdot\{\Phi_{n]2,n-3} - b_1(n-2)[M_{n-2}\Phi_{n-2,n-3} + N_{n-2}]\}' \cdots$$

$$\cdot\{\Phi_{k+1,k} - b_1(k+1)[M_{k+1}\Phi_{k+1,k} + N_k]\}'\,\tilde{x}(k)$$

$$+ \underline{f}[g_2,\ldots,g_{n-2}] + \underline{h}[\delta y_{k+1},\ldots,\delta y_{n-1}]$$

where

$\underline{f}(\cdot)$ and $\underline{h}(\cdot)$ are linear functions and

$$Px\tilde{y} = P(k/k)\prod_{i=k+1}^{n-1}\{\Phi_{i,i-1} - b_1(i)[M_i\Phi_{i,i-1} + N_i]\}'(M_n\Phi_{n,n-1}+N_n]'$$

it may be written that

$$\hat{x}(k/n) = \hat{x}(k/k) + P(k/k)\sum_{i=k+1}^{n-1} d(k,i)[M_i\Phi_{i,i-1} + N_i]'C_i^{-1}\tilde{y}(i/i-1)$$

$$+ P(k/k)d(k,n)[M_n\Phi_{n,n-1} + N_n]'C_n^{-1}\tilde{y}(n/n-1)$$

$$= \hat{x}(k/k) + P(k/k)\sum_{i=k+1}^{n} d(k,i)[M_i\Phi_{i,i-1} + N_i]'C_i^{-1}\tilde{y}(i/i-1)$$

where

$$d(k,k+1) = I$$

$$d(k,i) = d(k,i-1)\{\Phi_{i-1,i-2} - b_1(i-1)[M_{i-1}\Phi_{i-1,i-2}+N_{i-1}]\}'$$

$$\text{for } i=k+2,\ldots,n \quad .$$

The induction proof is complete and gives the general fixed interval smoothing equation for the delayed state model. This equation can be put into a more useful form by noting that

$$d(k,i) = d(k,k+2)d(k+1,i) \quad i=k+3,\ldots,n$$

and if

$$z(k,n) = \sum_{i=k+1}^{n} d(k,i)[M_i\Phi_{i,i-1} + N_i]' \, C_i^{-1}\tilde{y}(i/i-1)$$

$$= [M_{k+1}\Phi_{k+1,k} + N_{k+1}]' \, C_{k+1}^{-1}\tilde{y}(k+1/k)$$

$$+ d(k,k+2) \sum_{i=k+2}^{N} d(k+1,i)(M_i\Phi_{i,i-1} + N_i)' C_i^{-1}\tilde{y}(i/i-1)$$

$$= d(k,k+2)z(k+1,n) + (M_{k+1}\Phi_{k+1,k} + N_{k+1})' C_{k+1}^{-1}\tilde{y}(k+1/k)$$

then

$$\hat{x}(k/n) = \hat{x}(k/k) + P(k/k)z(k,n)$$

where

$$z(k,n) = d(k,k+2)z(k+1,n) + (M_{k+1}\Phi_{k+1,k} + N_{k+1})' C_{k+1}^{-1}\tilde{y}(k+1/k)$$

$$z(n-1,n) = [M_n\Phi_{n,n-1} + N_n]' \, C_n^{-1}\tilde{y}(n/n-1)$$

$$z(n,n) = 0 \quad .$$

Since $z(k,n)$ has the same dimensions as $\hat{x}(k/n)$, this equation is similar to the fixed interval smoothing equation developed by Cox (5). Therefore, this algorithm has the same problems as the one by Cox and others. In the computation the recursive expression for the covariance matrix does not follow the backward movement of the equations. There are three ways to compute $\hat{x}(k/n)$. The first is to place in memory all the a posteriori covariances that are needed. This method becomes ridiculous when the size of the system is large. The second method would be to write a recursive relationship that will propagate the covariances in the backward direction to match the rest of the computations. But, such a relationship would involve an inverse of a matrix of the order of the system. This is one of the problems that is trying to be circumvented by writing the smoothing equations with all the data present. The last method, which seems to be a reasonable one when the system is large, is computing in the backward direction just the $z(k,n)$ matrix for the whole interval of interest. Then, calculate $\hat{x}(k/n)$ in the forward direction carrying along the calculations of the covariance matrix. The only things that are then needed to be remembered other than the estimates, which are needed in any method, are the vector $z(k,n)$ and the initial covariance matrix, to start the covariance computation. This method is illustrated in the flow chart in Figure 2.

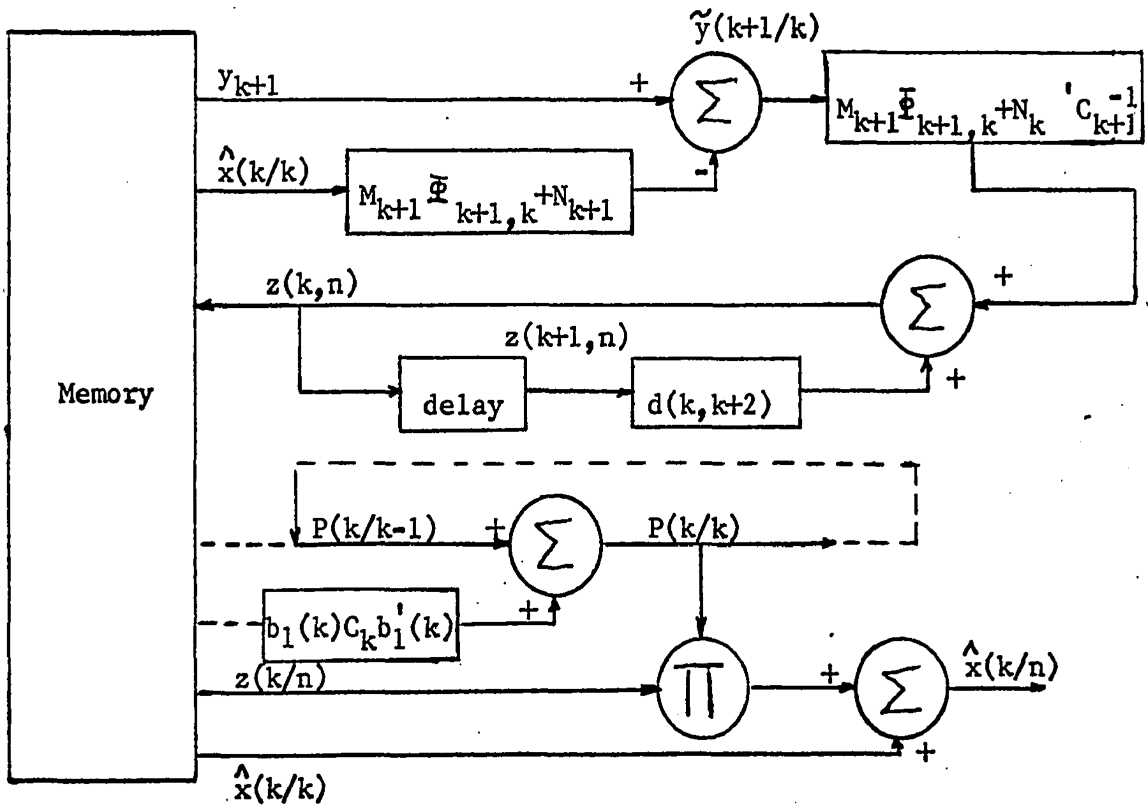

Figure 2. Flow chart for the smoothing equations for the delayed state model

C. Fixed Point Smoothing Equations for the Delayed State Kalman Filter

In order to derive the fixed point smoothing equation one must stop sooner in the development of the fixed interval equations. Going back to the induction proof used to derive the fixed interval smoothing equation, it was written by definition

$$\hat{x}(k/n) = \hat{x}(k/n-1) + P_{x\tilde{y}} P_{\tilde{y}\tilde{y}}^{-1} \tilde{y}(n/n-1)$$

which could have been left to be

$$\hat{x}(k/n) = \hat{x}(k/n-1) + P(k/k)d(k,n)[M_n \Phi_{n,n-1} + N_n]' \cdot C_n^{-1}\tilde{y}(n/n-1) \quad .$$

This equation can be interpreted as being the fixed point smoothing equation. That being that the estimate of the state variable at time k given the data through time n-1 is up-dated with weighted data at time n to form the estimate at time k given the data through time n.

By this development, it is evident that the only difference in the two smoothing algorithms, fixed point and fixed interval is the method in which the smoothed estimates are calculated. The most important difference in the computations is the fact that the fixed point algorithm is on-line, where the fixed interval computations are done after the fact. But, if n-k, the number of points in the fixed interval, is small and if a large computer was available, it could be possible to form an on-line fixed interval smoother made up of n-k fixed point smoothers.

D. Covariance Equation for the Smoothing Scheme

The covariance matrix for the smoothed estimate was not needed to form the smoothed estimate. This is true in most smoothing schemes. To add a measure of completeness to the new scheme presented in the second section, the error covariance equation for the smoothed estimate will be formed in this section.

By definition, the error covariance is

$$P(k/n) = E[\tilde{x}(k/n)\tilde{x}'(k/n)]$$

where

$$\tilde{x}(k/n) = x(k) - \hat{x}(k/n) .$$

Substituting the error equation into the smoothing equations, it follows that

$$\begin{aligned}\tilde{x}(k/n) &= x(k) - \hat{x}(k/k) - P(k/k)z(k,n)\\ &= \tilde{x}(k) - P(k/k)z(k,n)\end{aligned}$$

which implies that

$$\begin{aligned}P(k/n) &= E[\tilde{x}(k/n)\tilde{x}'(k/n)]\\ &= E[\tilde{x}(k)\tilde{x}'(k)] - E[\tilde{x}(k)z'(k,n)P'(k/k)] - E[P(k/k)\\ &\quad \cdot z(k,n)\tilde{x}'(k)] + E[P(k/k)z(k,n)z'(k,n)P'(k/k)]\end{aligned}$$

$$= P(k/k) - E[\tilde{x}(k) z'(k,n)]P'(k/k) - P(k/k)$$
$$\cdot E[z(k,n)\tilde{x}'(k)] + P(k/k)E[z(k,n)z'(k,n)]P'(k/n)$$

where

$$z(k,n) = \sum_{i=k+1}^{n} d(k,i)[M_i \Phi_{i,i-1} + N_i]' C_i^{-1} \tilde{y}(i/i-1)$$

$$\tilde{y}(i/i-1) = (M_i \Phi_{i,i-1} + N_i)\tilde{x}(i-1) + M_i g_{i-1} + \delta y_i$$

$$\tilde{x}(k) = \{\Phi_{k,k-1} - b_1(k)[M_k \Phi_{k,k-1} + N_k]\}\tilde{x}(k-1) + [I - b_1(k) M_k]$$
$$\cdot g_{k-1} - b_1(k)\delta y_k \quad .$$

Looking at the third term of the error covariance equation term by term, it follows that for i=k+1

$$E[z(k,n)\tilde{x}'(k)] = (M_{k+1}\Phi_{k+1,k} + N_{k+1})' C_{k+1}^{-1} (M_{k+1}\Phi_{k+1,k} + N_{k+1})$$
$$P(k/k) \ .$$

For i=k+2

$$E[z(k,n)x'(k)] = d(k,k+2)(M_{k+2}\Phi_{k+2,k+1} + N_{k+2})' C_{k+2}^{-1} (M_{k+2}\Phi_{k+2,k+1}$$
$$+ N_{k+2}) d'(k,k+2) P(k/k)$$

and in general, when i=j where $k+2 < j < n$

$$E[z(k/n)\tilde{x}'(k)] = d(k,j)(M_j \Phi_{j,j-1} + N_j)' C_j^{-1} (M_j \Phi_{j,j-1} + N_j)$$
$$\cdot d'(k,j) P(k/k) \quad .$$

Therefore, summing up all the terms

$$E[z(k,n)\tilde{x}'(k/k)] = [\sum_{i=k+1}^{n} d(k,i)(M_i\Phi_{i,i-1} + N_i)' C_i^{-1}$$

$$\cdot (M_i\Phi_{i,i-1} + N_i) d'(k,i)]P(k/k) \quad .$$

Now the two middle terms in the equation for the error covariance for the smoothed estimate can be written as

$$E[\tilde{x}(k)z'(k,n)]P'(k/k) - P(k/k)E[z(k,n)\tilde{x}'(k/k)]$$

$$= 2P(k/k)[\sum_{i=k+1}^{n} d(k,i)P_z(i-1/i)d'(k,i)]P(k/k)$$

where

$$P_z(i-1/i) = (M_i\Phi_{i,i-1} + N_i)' C_i^{-1}(M_i\Phi_{i,i-1} + N_i) \quad .$$

Now, to finish the evaluation of the error covariance, the term $E(z(k,n)z'(k,n))$ must be evaluated. Using the definition of $P_z(i-1/i)$ and expressing it in more general terms

$$P_z(i/j) = E[z(i,j)z'(i,j)]$$

a recursive relationship will be formed so that $P_z(i/j)$ can be evaluated as the process steps along. The first terms of such a relationship can be written by inspection

$$P_z(n-1/n) = (M_n\Phi_{n,n-1} + N_n)' C_n^{-1}(M_n\Phi_{n,n-1} + N_n) \quad .$$

The second term is by definition

$$P_z(n-2/n) = E[z(n-2,n)z'(n-2,n)]$$

$$= d(n-2,n)E[z(n-1,n)z'(n-1,n)]d'(n-2,n)$$

$$+ \; d(n-2,n)E[z(n-1,n)\tilde{y}'(n-1/n-2)]C_{n-1}^{-1}(M_{n-1}\Phi_{n-1,n-2}$$

$$+ \; N_n) + (M_{n-1}\Phi_{n-1,n-2} + N_{n-1})' \, C_{n-1}^{-1} \; E[\tilde{y}(n-1/n-2)$$

$$\cdot \; z'(n-1,n)]d'(n-2,n) + (M_{n-1}\Phi_{n-1,n-2} + N_{n-1})'$$

$$\cdot \; C_{n-1}^{-1} \; E[\tilde{y}(n-1/n-2)\tilde{y}'(n-1/n-2)]C_{n-1}^{-1}(M_{n-1}\Phi_{n-1,n-2}$$

$$+ \; N_{n-1})$$

where

$$E[z(n-1,n)\tilde{y}(n-1/n-2)] = (M_n\Phi_{n,n-1} + N_n)' \, C_n^{-1}(M_n\Phi_{n,n-1} + N_n)$$

$$\cdot \; d'(n-2,n)P(n-2/n-2)(M_{n-1}\Phi_{n-1,n-2} + N_{n-1})$$

$$+ \; (M_n\Phi_{n,n-1} + N_n)' \, C_n^{-1}(M_n\Phi_{n,n-1} + N_n)[I-b_1(n-1)M_{n-1}]$$

$$\cdot \; Q_{n-2}M'_{n-1}$$

$$- \; (M_n\Phi_{n,n-1} + N_n)' \, C_n^{-1}(M_n\Phi_{n,n-1} + N_n)b_1(n-1)R_{n-1}$$

which reduces to

$$P_z(n-2/n) = d(n-2,n)P_z(n-1/n)d'(n-2,n) + P_z(n-2/n-1) \quad .$$

Now by assuming

$$P_z(k+1/n) = d(k+1,k+3)P_z(k+2/n)d'(k+1,k+3) + P_z(k+1/k+2)$$

and proceeding as was done previously it can be shown that

$$P_z(k/n) = d(k,k+2)P_z(k+1/n)d'(k,k+2) + P_z(k/k+1)$$

which proves by induction that the above equation is the expression for $E[z(k,n)z'(k,n)]$. Using the identities for $d(k,n)$ given in the preceding sections the above expression for $E[z(k,n)z'(k,n)]$ can be rewritten.

$$P_z(k/n) = \sum_{i=k+1}^{n} d(k,i)P_z(i-1/i)d'(k,i)$$

where

$$P_z(i-1/i) = (M_i\Phi_{i,i-1} + N_i)' C_i^{-1} (M_i\Phi_{i,i-1} + N_i) \quad .$$

Therefore,

$$P(k/n) = P(k/k) - 2P(k/k)\Big[\sum_{i=k+1}^{n} d(k,i)P_z(i-1/i)d'(k,i)\Big]P(k/k)$$

$$+ P(k/k)\Big[\sum_{i=k+1}^{n} d(k,i)P_z(i-1/i)d'(k,i)\Big]P(k/k)$$

$$= P(k/k) - P(k/k)P_z(k/n)P(k/k) \quad .$$

V. DECOUPLING OF THE DELAYED STATE MODEL FILTERING AND SMOOTHING EQUATIONS

To add completeness to the delayed state model equations, these too will be decoupled when bias states exist. Therefore, the effective system size will be reduced by applying the idea of parallel computations as Friedland (7) has done for the regularly modeled systems. Because of the systems that can be twisted into the delayed state model the decoupling of the equations for this model should be useful. For example, it has been mentioned previously that aided inertial systems fall into this class of systems especially those systems with bias states defined. Here computer size and computation time are at a premium. So, if decoupling the equations alleviates some of these specifications it would be well worth the time and effort.

The first section of this chapter presents the decoupling of the filtering equations derived by Brown and Hartmann (3) and presented in the last chapter. Many additions had to be made to Friedland's method because of the delayed state present in the measurement equation. After much juggling of equations it will be shown that the estimate of the state vector x at time k given k data points can be decoupled when the system has been augmented because of biases present.

Proceeding further, the second section presents the decoupling of the smoothing equations derived in the last chapter when the model of the system has bias states present. The most difficult part of this development is writing the expressions for the portions of the $z(k,n)$ matrice. The results of this development do not appear as simple as those presented in Chapter III. This is because of the cross coupling involved due to the presence of the delayed state vector in the measurement equation.

A. Delayed State Kalman Filter with Bias States

Friedland has offered a scheme to decouple the computation of bias estimates and the state estimates to save time and money. Of course the saving is greatest in large systems with about the same number of bias variables as there are states. This saving would also be of interest to people trying to use the delayed state Kalman filter equations. Therefore, the rest of this section will consist of the development of the decoupling of the delayed state Kalman filter with bias states.

The dynamical equation for this development are the same as before except the delayed state explicitly occurs in the measurement equation.

$$x(k+1) = A_k x(k) + B_k b(k) + g_k$$

$$b(k+1) = b(k)$$

$$y(k) = M_k x(k) + N_k x(k-1) + C_k b(k) + \delta y_k \quad .$$

If the state vector is augmented to include the bias state by redefining the state vector as

$$Z(k) = \begin{bmatrix} x(k) \\ \hline b(k) \end{bmatrix} \quad ,$$

the dynamical equations are

$$Z(k+1) = F_k Z(k) + G g_k$$

$$y(k) = L_k Z(k) + J_k Z(k-1) + \delta y_k$$

where

$$F_k = \left[\begin{array}{c|c} A_k & B_k \\ \hline 0 & I \end{array}\right]$$

$$G = \begin{bmatrix} I \\ \hline 0 \end{bmatrix}$$

$$L_k = [H_k \mid C_k]$$

$$J_k = [N_k \mid 0] \quad .$$

From the review of the delayed state Kalman filter, the equations for the best estimate of Z at time k+1 given the set of observations $\{y(1), y(2), \ldots, y(k+1)\}$ are as follows

$$\hat{Z}(k+1/k+1) = F_k \hat{Z}(k/k) + W(k+1)[y(k+1) - (L_{k+1}F_k + J_{k+1})\hat{Z}(k/k)]$$

where

$$W(k+1) = P(k+1/k)L'_{k+1}D_{k+1}^{-1}$$

$$D_{k+1} = L_{k+1}P(k+1/k)L'_{k+1} + L_{k+1}F_k\ P(k/k)J'_{k+1} + J_{k+1}P(k/k)F'_kL'_{k+1}$$

$$+ J_{k+1}P(k/k)J'_{k+1} + R_{k+1}$$

$$P(k/k) = [I - W(k)L_k]P(k/k-1) - W(k)J_kP(k-1/k-1)F'_{k-1}$$

$$P(k+1/k) = F_kP(k/k)F'_k + GQ_kG' \quad .$$

Now, define $\tilde{P}(k/k-1)$ as the covariance which is the solution to the above equation for the a priori covariance for the initial conditions

$$\tilde{P}(0/-1) = \left[\begin{array}{c|c} \tilde{P}_x(0/-1) & 0 \\ \hline 0 & 0 \end{array}\right] \quad .$$

Therefore, a general solution may be written as

$$P(k/k-1) = \tilde{P}(k/k-1) + U(k)M(k)U'(k)$$

where here again $M(k)$ is an r x r symmetric matrix. The last term of the above equation comes from the fact that

$$P_{xb}(0/-1) \neq 0$$

$$P_b(0/-1) \neq 0$$

and

$$U(k+1) = F_k[I - \tilde{W}(k)E_k]U(k)$$

$$M(k+1) = M(k) - M(k)U'(k)E'_kD_k^{-1}E_kU(k)M(k)$$

where

$$\tilde{W}(k) = \tilde{P}(k/k-1)L'_k\tilde{D}_k^{-1} + F_{k-1}\tilde{P}(k-1/k-1)J'_k\tilde{D}_k^{-1}$$

$$\tilde{D}_k = L_k\tilde{P}(k/k-1)L'_k + L_kF_{k-1}\tilde{P}(k-1/k-1)J'_k + J_k\tilde{P}(k-1/k-1)F'_{k-1}L'_k$$

$$+ J_k\tilde{P}(k-1/k-1)J'_k + R_k$$

$$E_k = L_k + J_kF_{k-1}^{-1} = [L_kF_{k-1} + J_k]F_{k-1}^{-1} = \bar{E}_kF_{k-1}^{-1} \quad .$$

The proof of the above equations is in Appendix A. Proceeding on as was done in Friedland's original paper (7) the a posteriori covariance equation is written as

$$P(k/k) = \tilde{P}(k/k) + V(k)M(k+1)V'(k) \quad .$$

Using this fact and the two previous equations for the a priori covariance equation it follows that

$$U(k+1) = F_kV(k)$$

which implies that

$$V(k) = [I - \tilde{W}(k)E_k]U(k)$$

or which can be written as

$$V(k) = [F_{k-1} - \tilde{W}(k)\bar{E}_k]F_{k-1}^{-1}U(k) = [F_{k-1} - W(k)\bar{E}_k]\bar{U}(k) \quad .$$

Next, the expression for $\tilde{D}_k$ will be derived by using the identity for $\bar{E}$ and the a priori covariance equation

$$\tilde{D}_k = L_k\tilde{P}(k/k-1)L'_k + L_k[\tilde{P}(k/k-1) - GQ_kG'](F'_{k-1})^{-1}J'_k$$

$$+ J_k F_{k-1}^{-1}[\tilde{P}(k/k-1) - GQ_kG']L_k'$$

$$+ J_k F_{k-1}^{-1}[\tilde{P}(k/k-1) - GQ_kG'](F_{k-1}')^{-1}J_k' + R_k$$

$$= E_k[\tilde{P}(k/k-1) - GQ_kG']E_k' + L_kGQ_kG'L_k' + R_k$$

$$= \bar{E}_k F_{k-1}^{-1}[\tilde{P}(k/k-1) - GQ_kG'](F_{k-1}')^{-1}\bar{E}_k' + L_kGQ_kG'L_k' + R_k \quad .$$

Now the expression above is expanded taking small portions of the equation at a time.

$$F_{k-1}^{-1}[P(k/k-1) - GQ_kG'](F_{k-1}')^{-1} = \tilde{P}(k-1/k-1) = \left[\begin{array}{c|c} \tilde{P}_x(k-1/k-1) & 0 \\ \hline 0 & 0 \end{array}\right]$$

$$L_kGQ_kG'L_k' = [H_k \mid C_k]\left[\begin{array}{c|c} Q_k & 0 \\ \hline 0 & 0 \end{array}\right]\left[\begin{array}{c} H_k' \\ \hline C_k' \end{array}\right] = H_kQ_kH_k'$$

$$\bar{E}_k = [H_kA_{k-1} + N_k \mid H_kB_{k-1} + C_k] \; .$$

Therefore,

$$\tilde{D}_k = [H_kA_{k-1} + N_k]\tilde{P}_x(k-1/k-1)[H_kA_{k-1} + N_k]' + H_kQ_kH_k' + R_k$$

$$= H_k\tilde{P}(k/k-1)H_k' + H_kA_{k-1}\tilde{P}_x(k-1/k-1)N_k' + N_k\tilde{P}_x(k-1/k-1)$$

$$\cdot A_{k-1}'H_k' + N_k\tilde{P}_x(k-1/k-1)N_k' + R_k \quad .$$

Next, the expression for $\tilde{W}(k)$ will be derived by expanding the following

$$\tilde{W}(k) = \tilde{P}(k/k-1)E_k'\tilde{D}_k^{-1} - GQ_kG'(F_{k-1}')^{-1}J_k'\tilde{D}_k^{-1}$$

$$= [\tilde{P}(k/k-1)(F'_{k-1})^{-1}\bar{E}'_k - GQ_kG'(F'_{k-1})^{-1}J'_k]\tilde{D}_k^{-1}$$

where

$$\tilde{P}(k/k-1)(F'_{k-1})^{-1} = \left[\begin{array}{c|c} \tilde{P}_x(k/k-1)(A'_k)-1 & 0 \\ \hline 0 & 0 \end{array}\right]$$

$$\tilde{P}(k/k-1)(F'_{k-1})^{-1}\bar{E}_k^{-1} = \left[\begin{array}{c} \tilde{P}_x(k/k-1)H'_k+\tilde{P}_x(k/k-1)(A'_k)^{-1}N'_k \\ \hline 0 \end{array}\right]$$

$$GQ_kG'(F'_{k-1})^{-1}J'_k = \left[\begin{array}{c} Q_k(A'_{k-1})^{-1}N'_k \\ \hline 0 \end{array}\right] .$$

Then it follows that

$$\tilde{W}(k) = \left[\begin{array}{c} \tilde{P}_x(i/k-1)H'_k+(\tilde{P}_x(k/k-1)-Q_k)(A'_{k-1})^{-1}N'_k \\ \hline 0 \end{array}\right]\tilde{D}_k^{-1}$$

$$= \left[\begin{array}{c} \tilde{P}_x(k/k-1)H'_k\tilde{D}_k^{-1}+A_{k-1}\tilde{P}_x(k-1/k-1)N'_k\tilde{D}_k^{-1} \\ \hline 0 \end{array}\right]$$

$$= \left[\begin{array}{c} \tilde{W}_x(k) \\ \hline 0 \end{array}\right] .$$

Now, the expression linking U(k) and V(k) must be expanded. First, however, some identities must be established from past equations. Since

$$U(k) = F_{k-1}\bar{U}(k)$$

then

$$\begin{bmatrix} U_x(k) \\ \hline U_b(k) \end{bmatrix} = \left[\begin{array}{c|c} A_{k-1} & B_{k-1} \\ \hline 0 & I \end{array}\right] \begin{bmatrix} \bar{U}_x(k) \\ \hline \bar{U}_b(k) \end{bmatrix} = \begin{bmatrix} A_{k-1}\bar{U}_x(k)+B_{k-1}\bar{U}_b(k) \\ \hline \bar{U}_b(k) \end{bmatrix} .$$

Also,

$$U(k+1) = \begin{bmatrix} U_x(k+1) \\ \hline U_b(k+1) \end{bmatrix} = F_k V(k) = \left[\begin{array}{c|c} A_k & B_k \\ \hline 0 & I \end{array}\right] \begin{bmatrix} V_x(k) \\ \hline V_b(k) \end{bmatrix} = \begin{bmatrix} A_k V_x(k)+B_k V_b(k) \\ \hline V_b(k) \end{bmatrix}$$

and

$$\begin{bmatrix} V_x(k) \\ \hline V_b(k) \end{bmatrix} = (F_{k-1} - \tilde{W}(k)\bar{E}_k)\bar{U}(k)$$

$$= \left[\begin{array}{c|c} A_{k-1} - \tilde{W}_x(k)H_k A_{k-1} + N_k & B_{k-1} - W_x(k)H_k B_{k-1} + C_k \\ \hline 0 & I \end{array}\right] \begin{bmatrix} \bar{U}_x(k) \\ \hline \bar{U}_b(k) \end{bmatrix}$$

$$= \begin{bmatrix} A_{k-1}\bar{U}_x(k)+B_{k-1}\bar{U}_b(k)-\tilde{W}_x(k)\{H_k[A_{k-1}\bar{U}_x(k)+B_{k-1}\bar{U}_b(k)]+N_k U_x(k)+C_k U_b(k) \\ \hline \bar{U}_b(k) \end{bmatrix}$$

$$= \left[\frac{U_x(k) - W_x(k) H_k U_x(k) + N_k \bar{U}_x(k) + C_k U_b(k)}{U_b(k)} \right] .$$

The above equations imply that

$$U_b(k) = V_b(k) = U_b(0) = \text{Constant, for all } k \quad .$$

If it is assumed x and b are independent at k = 0, i.e.,

$$P_{xb}(0/-1) = 0$$

and if

$$U_b(0) = I \quad \text{and} \quad U_x(0) = 0 \ ,$$

then

$$V_x(k) = U_x(k) - \tilde{W}_x(k) [H_k U_x(k) + N_k \bar{U}_x(k) + C_k]$$

$$= U_x(k) - \tilde{W}_x(k) T(k)$$

$$M(k+1) = M(k) - M(k) T'(k) D_k^{-1} T(k) M(k)$$

$$U_x(k+1) = A_k V_x(k) + B_k \quad .$$

The above expressions also allows the components of the covariance matrices to be simplified as follows

$$P_x(k/k-1) = \tilde{P}(k/k-1) + U_x(k) M(k) U_x'(k)$$

$$P_{xb}(k/k-1) = U_x(k) M(k)$$

$$P_b(k/k-1) = M(k)$$

$$P_x(k/k) = \tilde{P}(k/k) + V_x(k)M(k+1)V'_x(k)$$

$$P_{xb}(k/k) = V_x(k)M(k+1)$$

$$P_b(k/k) = M(k+1) \quad .$$

To be able to write the equation for the best estimate of the augmented state vector into the partitioned equations, the estimate of the bias and state vectors, the gain matrix M(k) must be partitioned. The equation for W(k) is as follows

$$W(k) = P(k/k-1)L'_k D_k^{-1} + F_{k-1}P(k-1/k-1)J'_k D_k^{-1}$$

where

$$P(k/k-1)L'_k = \left[\begin{array}{c|c} P_x(k/k-1) & P_{xb}(k/k-1) \\ \hline P'_{xb}(k/k-1) & P_b(k/k-1) \end{array}\right] \left[\begin{array}{c} H'_k \\ \hline C'_k \end{array}\right]$$

$$= \left[\begin{array}{c} P_x(k/k-1)\ H'_k + P_{xb}(k/k-1)\ C'_k \\ \hline P'_{xb}(k/k-1)H'_k + P_b(k/k-1)\ C'_k \end{array}\right]$$

$$F_{k-1}P(k-1/k-1)J'_k = \left[\begin{array}{c|c} A_{k-1} & B_{k-1} \\ \hline 0 & I \end{array}\right] \left[\begin{array}{c|c} P_x(k-1/k-1) & P_{xb}(k-1/k-1) \\ \hline P'_{xb}(k-1/k-1) & P_b(k-1/k-1) \end{array}\right] \left[\begin{array}{c} N'_k \\ \hline 0 \end{array}\right]$$

$$= \left[\begin{array}{c} A_{k-1}P_x(k-1/k-1)N'_k + B_{k-1}P'_{xb}(k-1/k-1)N'_k \\ \hline P'_{xb}(k-1/k-1)N'_k \end{array}\right]$$

Therefore

$$W(k) = \begin{bmatrix} W_x(k) \\ \hline W_b(k) \end{bmatrix}$$

$$= \begin{bmatrix} P_x(k/k-1)H'_k + P_{xb}(k/k-1)C'_k + A_{k-1}P_x(k-1/k-1)N'_k + B_{k-1}P'_{xb}(k-1/k-1)N'_k \\ \hline P'_{xb}(k/k-1)H'_k + P_b(k/k-1)C'_k + P'_{xb}(k-1/k-1)N'_k \end{bmatrix}$$

$$\cdot\ D_k^{-1} \quad .$$

Looking at the equation for $W_b(k)$, it may be reduced by using the expression for the components of the covariance matrices.

$$\begin{aligned} W_b(k) &= [P'_{xb}(k/k-1H'_k + P_b(k/k-1)C'_k + P'_{xb}(k-1/k-1)N'_k]D_k^{-1} \\ &= [M(k)U'_x(k)H'_k + M(k)C'_k + M(k)V'_x(k-1) \; N'_k]D_k^{-1} \\ &= M(k)[U'_x(k)H'_k + C'_k + \bar{U}'_x(k)N'_k]D_k^{-1} \\ &= M(k)T'(k)D_k^{-1} \quad . \end{aligned}$$

Now proceeding in the same manner the gain $W_x(k)$ is reduced.

$$\begin{aligned} W_x(k) &= [P_x(k/k-1)H'_k + P_{xb}(k/k-1)C' + A_{k-1}P_x(k-1/k-1)N'_k \\ &\quad + B_{k-1}P'_{xb}(k-1/k-1)N'_k]D_k^{-1} \\ &= \tilde{W}_x(k) + V_x(k)W_b(k) \quad . \end{aligned}$$

The proof of the above equation is found in Appendix B.

The best estimate of the augment state vector is written, as before,

$$\hat{z}(k/k) = F_{k-1}\hat{z}(k-1/k-1) + W(k)[y_k - (L_k F_{k-1} + J_k)\hat{z}(k-1/k-1)]$$

which can be rewritten as two equations ,

$$\hat{x}(k/k) = A_{k-1}\hat{x}(k-1/k-1) + B_{k-1}\hat{b}(k-1/k-1)$$
$$+ W_x(k)[y_k - (H_k A_{k-1}+N_k)\hat{x}(k-1/k-1) - (H_k B_{k-1}+C_k)$$
$$\cdot\ \hat{b}(k-1/k-1)]$$

and

$$\hat{b}(k/k) = \hat{b}(k-1/k-1) + W_b[y_k - (H_k A_{k-1} + N_k)\hat{x}(k-1/k-1)$$
$$- (H_k B_{k-1} + C_k)\hat{b}(k-1/k-1)] \quad .$$

Let $\tilde{x}(k/k)$ be the bias-free estimate, i.e.,

$$\tilde{x}(k/k) = A_{k-1}\tilde{x}(k-1/k-1) + \tilde{W}_x(k)[y_k - (H_k A_{k-1}+N_k)\tilde{x}(k-1/k-1)]$$

and $\tilde{W}_x(k)$ is the bias-free gain given earlier. The result that is to be shown is

$$\hat{x}(k/k) = \tilde{x}(k/k) + V_x(k)\hat{b}(k/k) \quad .$$

To prove this result, the residuals of the partitioned equation will be written as

$$y_k - (H_k A_{k-1} + N_k)\hat{x}(k-1/k-1) - (H_k B_{k-1} + C_k)\hat{b}(k-1/k-1)$$

$$= y_k - (H_k A_{k-1} + N_k)\tilde{x}(k-1/k-1) - (H_k A_{k-1} V_x(k-1)$$

$$+ N_k V_x(k-1) + H_k B_{k-1} + C_k)\hat{b}(k-1/k-1)$$

$$= \tilde{\gamma}_k - (H_k U_x(k) + V_k \bar{U}_x(k) + C_k)\hat{b}(k-1/k-1)$$

$$= \tilde{\gamma}_k - T(k)\hat{b}(k-1/k-1)$$

where $\tilde{\gamma}_k = y_k - H_k A_{k-1}\tilde{x}(k-1/k-1)$ = the residual of the bias-free estimation. Using this result the expression for $b(k/k)$ is

$$\hat{b}(k/k) = \hat{b}(k-1/k-1) + W_b(k)[\tilde{\gamma}_k - T(k)\hat{b}(k-1/k-1)]$$

$$= [I - W_b(k)T(k)]\hat{b}(k-1/k-1) + W_b(k)\tilde{\gamma}_k$$

and the expressions for $\hat{x}(k/k)$ is

$$\hat{x}(k/k) = A_{k-1}\tilde{x}(k-1/k-1) + [A_{k-1}V_x(k-1) + B_{k-1} - W_x(k)T(k)]$$

$$\cdot\ \hat{b}(k-1/k-1) + W_x(k)\tilde{\gamma}_k$$

which implies that

$$\hat{x}(k/k) = \tilde{x}(k/k) + V_x(k)\hat{b}(k/k) = \tilde{x}(k/k) + V_x(k)[I-W_b(k)T(k)]$$

$$\cdot\ \hat{b}(k-1/k-1) + V_x(k)W_b(k)\tilde{\gamma}_k$$

$$= A_{k-1}\tilde{x}(k-1/k-1) + [A_{k-1}V_x(k-1) + B_{k-1} + W_x(k)T(k)]$$

$$\cdot\ \hat{b}(k-1/k-1) + W_x(k)\tilde{\gamma}_k$$

for all k, $\tilde{\gamma}_k$, and $\hat{b}(k/k)$. This requires that

$$V_x(k)[I - W_b(k)T(k)] = A_{k-1}V_x(k-1) + B_{k-1} - W_x(k)T(k)$$

since

$$W_x(k) = \tilde{W}_x(k) + V_x(k)W_b(k)$$

which can be reduced to

$$V_x(k) = V_x(k)W_b(k)T(k) + A_{k-1}V_x(k-1) + B_{k-1} - W_x(k)T(k)$$

$$= U_x(k) - \tilde{W}_x(k)T(k)$$

which is the expression for $V_x(k)$ previously derived. Hence the desired result,

$$\hat{x}(k/k) = \tilde{x}(k/k) + V_x(k)\hat{b}(k/k)$$

has been proved.

This result is the same as Friedland obtained in his paper except some of the matrices in the development are different because of the delayed state. Therefore, there is also a savings in the delayed state Kalman filter by modeling a system with biases and then decoupling the computation, as was indicated for the regular Kalman filter.

B. Fixed Interval Smoothing Equations for the Decoupled Delayed State Kalman Filter

The justification for deriving this decoupled fixed-interval smoothing equation will be the same as was given in Chapter III when that smoothing equation was derived for the decoupled Kalman filter.

Again the development is started with the augmented difference equation, which are repeated here for convenience.

$$Z(k+1) = F_k Z(k) + G g_k$$

$$y_k = L_k Z(k) + J_k Z(k-1) + \delta y_k$$

where

$$Z(k) = \begin{bmatrix} x(k) \\ \hline b(k) \end{bmatrix} \quad F_k = \left[\begin{array}{c|c} A_k & B_k \\ \hline 0 & I \end{array}\right] \quad G = \begin{bmatrix} I \\ \hline 0 \end{bmatrix} \quad L_k = [H_k \mid C_k] \quad J_k = [N_k \mid 0] \quad .$$

Using the above notation, the fixed-interval smoothing equation will be

$$\hat{Z}(k/n) = Z(k/k) + P(k/k)z(k,n)$$

$$z(k,n) = d(k,k+2)z(k+1,n) + [L_{k+1}, F_k + J_{k+1}]' D_{k+1}^{-1}\tilde{y}(k+1/k)$$

where

$$\tilde{y}(k+1,k) = y_{k+1} - (L_{k+1}F_k + J_{k+1})\hat{Z}(k/k)$$

$$d(k,k+2) = d(k,k+1)\{F_k - W(k+1)[L_{k+1}F_k + J_k]\}'$$

$$d(k,k+1) = I \quad .$$

Since $z(k,n)$ can be partitioned as follows

$$z(k,n) = \begin{bmatrix} z_x(k,n) \\ \hline z_b(k,n) \end{bmatrix}$$

the augmented smoothing equation may be written as the following two equations:

$$\hat{x}(k/n) = \hat{x}(k/k) + P_x(k/k)z_x(k,n) + P_{xb}(k/k)z_b(k,n)$$

$$\hat{b}(k/n) = \hat{b}(k/k) + P'_{xb}(k/k)z_x(k,n) + P_b(k/k)z_b(k,n) \quad .$$

Using the results of the first section of the chapter, it follows that

$$\hat{x}(k/n) = \tilde{x}(k/k) + \tilde{P}_x(k/k)z_x(k,n) + V_x(k)\hat{b}(k/n) \quad .$$

In order to decouple the smoothing equation fully, $z_x(k,n)$ must be written as a function of $\tilde{z}_x(k,n)$, that term from the smoothing equation for the bias-free system. The needed relationship is

$$z_x(k,n) = \tilde{z}_x(k,n) - e(k+1,n)$$

where

$$e(k+1,n) = (H_{k+1}A_k+N_k)'\,\tilde{D}_{k+1}^{-1}T(k+1)\hat{b}(k+1/n)$$
$$+ \tilde{d}(k,k+2)e(k+2,n) \quad .$$

The development of this relationship is in Appendix C. Therefore, it follows that

$$\hat{x}(k/n) = \tilde{x}(k/n) + V_x(k)\hat{b}(k/n) - \tilde{P}_x(k/k)e(k+1,n)$$

where

$$e(k+1,n) = (H_{k+1}A_k+N_k)'\,\tilde{D}_{k+1}^{-1}\hat{b}(k+1/n)+\tilde{d}(k,k+2)e(k+2,n) \quad .$$

Notice that because of the delayed state in this model of the system, the best smoothed estimate of the bias at time k given the data through time n is not $\hat{b}(n/n)$ as was the case in Chapter III. Therefore, as is indicated, the smoothed estimate of the state $\hat{x}(k/n)$ is dependent not only on the decoupled terms, $\tilde{x}(k/n)$ and $\hat{b}(k/n)$, but also on all the previous smoothed estimates of the bias, $\hat{b}(k+1/n)$, $\hat{b}(k+2/n),\ldots,\hat{b}(n/n)$.

VI. COMPARISON OF RECURSIVE SMOOTHING VERSUS BATCH PROCESSING

The idea behind batch processing is to manipulate all the data at once, where as in a recursive scheme the data is processed one point at a time. Of course, hidden in the recursive method is the use of all previous data via the use of the past estimates. The idea of batch processing was dropped as an on line method when the recursive schemes were presented because of the size and number of computations that are involved. Another disadvantage of the batch processing methods was that each time a new data point was received the order of the problem or equation that would have to be manipulated would increase. So, when Kalman introduced his recursive equations, his method had many advantages just because of their recursive aspect. To reaffirm ones faith in recursive equations, two methods of batch processing will be compared to the recursive smoothing equation developed in the previous chapter. In order to make the comparison, system size and smoothing interval must be chosen. The smoothing interval will be 36 steps and the system size will be 16 states. This choice is consistent with the example to be given later.

A. Fixed Interval Batch Processing

The fixed interval batch processing could be rationalized from the fact that any estimate is a function of the input data. Therefore, the set of equations for the set of smoothed estimate for the 36 point interval can be written as

$$\hat{x}(1/36) = k_1(1)y(1) + k_1(2)y(2) + \ldots + k_1(36)y(36)$$

$$\hat{x}(2/36) = k_2(1)y(1) + k_2(2)y(2) + \ldots + k_2(36)y(26)$$

$$\vdots$$

$$\hat{x}(36/36) = k_{36}(1)y(1) + k_{36}(2)y(2) + \ldots + k_{36}(36)y(36)$$

where $k_i(j)$ are the weighting coefficients. These coefficients have to be chosen in some optimal fashion. Looking at the top term of any one of the above equations, the mean squared error is given by

$$\overline{e_n^2} = \overline{[\hat{x}(n/36) - x(n)]^2} = \overline{\hat{x}^2(n/36)} - \overline{2\hat{x}(n/36)x(n)} + \overline{x^2(n)}$$

$$= [k_n^2(1)y^2(1) + k_n^2(2)y^2(2) + \ldots + k_n^2(36)y^2(36)]$$

$$+ 2[k_n(1)k_n(2)\overline{y(1)y(2)} + k_n(1)k_n(3)\overline{y(1)y(3)} + \ldots$$

$$+ k_n(2)k_n(3)\overline{y(2)y(3)} \ldots] - 2[k_n(1)\overline{y(1)x(n)}$$

$$+ k_n(2)\overline{y(2)x(n)} + \ldots + k_n(36)\overline{y(36)x(n)}] + \overline{x^2(n)}$$

for $n = 1, 2, \ldots, 36$.

The $k_n(j)$ components must be found to minimize $\overline{e_n^e}$. Therefore, if the derivative of $\overline{e_n^2}$ is taken with respect to $k_n(j)$ components, $i = 1,2,\ldots,36$ and set equal to zero for all n, the following set of equations must be solved.

$$\begin{bmatrix} \overline{y^2(1)} & \overline{y(1)y(2)} & \cdots & \overline{y(1)y(36)} \\ \overline{y(2)y(1)} & \overline{y^2(2)} & \cdots & \overline{y(2)y(36)} \\ \vdots & & & \\ \overline{y(36)y(1)} & \overline{y(36)y(2)} & \cdots & \overline{y^2(36)} \end{bmatrix} \begin{bmatrix} k_n(1) \\ k_n(2) \\ \vdots \\ k_n(36) \end{bmatrix} = \begin{bmatrix} \overline{y(1)x(n)} \\ \overline{y(2)x(n)} \\ \vdots \\ \overline{y(36)x(n)} \end{bmatrix}.$$

This equation looks innocent enough and is easily solved, given a computer. However, this is just part of the situation. The first clue as to the situation is the size of the constants, $k_n(j)$, $j = 1,2,\ldots,36$. If it is assumed that the measurements are scalars, the best possible condition to reduce size, the constants are 16×1 vectors. Therefore, in order to solve for the constants from the above equation, 16 systems of 36 equations in 36 unknowns must be dealt with. Plus, if all 36 smoothed estimates are to be computed, a system of equations of the same order must be solved 36 times. But, if this were not enough, each known variance and covariance indicated in the above matrix equation must be evaluated. Using the set of difference equations used in Chapter IV to describe the delayed state system, it follows that

$$y(i)y(j) = H(i)x(i)x'(j)H'(j) + N(i)x(i-1)x'(j)H'(j) + \delta y_i x'(j)H'(j) + H(i)x(i)x'(j-1)N'(j) + N(i)x(i-1)x'(j-1)N'(j) + \delta y_i x'(j-1)N'(j) + H(i)x(i)\delta y'_j + N(i)x(i-1)\delta y'_j + \delta y_i \delta y'_j$$

where

$$X(i) = \Phi_{i,i-1}x(i-1) + g_{i-1} = \Phi_{i,i-2}x(i-2) + \Phi_{i,i-1}g_{i-2} + g_{i-1}$$

$$= \dots = \Phi_{i,0}x(0) + \sum_{j=1}^{i} \Phi_{i,j}g_{j-1} \quad .$$

In order to evaluate $\overline{y(i)y(j)} = E[y(i)y(j)]$ the following equation must be evaluated

$$E[x(i)x(j)] = E[\Phi_{i,0}x(0)x'(0)\Phi'_{j,0} + \sum_{n=1}^{i} \sum_{k=1}^{j} \Phi_{i,n}g_{n-1}g'_{k-1}\Phi'_{j,k}]$$

$$= \Phi_{i,0}E[x(0)x'(0)]\Phi'_{i,0} + \sum_{n=1}^{i} \sum_{k=1}^{j} \Phi_{i,n}E[g_{n-1}g'_{k-1}]\, \Phi'_{j,k} \quad .$$

Since the first term is a constant or bias term it can be subtracted out of the equation. Then all that is left is

$$E[x(i)x(j)] = \sum_{k=1}^{j} \Phi_{i,0}\Phi_{k,0}^{-1}Q_k(\Phi'_{k,0})^{-1}\Phi'_{j,0} \quad .$$

Since all the matrices in the above equation are 16 x 16 it will be a formidable task to form the above equation even if Q_k a diagonal matrix. Note that the above equation is just one term of $E[y(i)y(j)]$ which is just one term of the 36 x 36 matrix needed to find the coefficients $k_n(j)$, $j = 1,2,\ldots,36$. There are approximately 5,266 triple sums of the form indicated above to be evaluated in order to solve the 16 systems of 36 equations needed to form the smoothed estimates. One can see that if the number of smoothed estimates increases the number of operations given here increase many times. So, it is fair to say that in this batch processing scheme the size of the system, not the nature of the computations, makes the smoother a very difficult operation.

B. Fixed Point Batch Processing

This method uses the recursive filter equations to derive the best estimate of the state vector given all the data from the 36 point interval. Instead of the usual differences equations, the following will be used.

$$x(k+1) = \Phi_{k+1,k}x(k) + g_k$$

$$\underline{\bar{Y}}_{k+1} = A_{k+1}x(k) + \delta\underline{\bar{Y}}_{k+1}$$

where

$$\underline{\bar{Y}}_{k+1} = \begin{bmatrix} y_{k+1} \\ y_{k+2} \\ \vdots \\ y_{k+36} \end{bmatrix}$$

To find A and $\delta\underline{\overline{Y}}$, the measurement equations

$$y_1 = M_1 x(1) + N_1 x(0) + \delta y_1$$

$$y_2 = M_2 x(2) + N_2 x(1) + \delta y_2$$

$$\vdots$$

$$y_{36} = M_{36} x(36) + N_{36} x(35) + \delta y_{36}$$

must be put into the form indicated above. After the filtering equations have been applied the results will be the 36 smoothed estimates of the state variable.

To start with let k+1 = 1 then

$$x(i) = \Phi_{1,0} x(0) + g_0$$

$$\underline{\overline{Y}}_1 = A_1 x(0) + \delta\underline{\overline{Y}}_1 \quad .$$

Now A and $\underline{\overline{Y}}$ must be found. It follows that

$$y_1 = (M_1\Phi_{1,0} + N_1) x(0) + M_1 g_0 + \delta y_1$$

$$y_2 = (M_2\Phi_{2,1} + N_2)\Phi_{1,0} x(0) + (M_2\Phi_{2,1} + N_2) g_0 + M_2 g_1 + \delta y_2$$

$$y_3 = (M_3\Phi_{3,2} + N_3)\Phi_{2,1}\Phi_{1,0} x(0) + (M_3\Phi_{3,2} + N_3)\Phi_{2,1} g_0$$

$$+ (M_3\Phi_{3,2} + N_3) g_1 + M_3 g_2 + \delta y_3$$

$$\vdots$$

Therefore,

$$\underline{\overline{Y}}_1 = A_1 x(0) + B_1 G_1 + \begin{bmatrix} \delta y_1 \\ \delta y_2 \\ \cdot \\ \cdot \\ \cdot \\ \delta y_{36} \end{bmatrix}$$

where

$$A_1 = \begin{bmatrix} (M_1 \Phi_{1,0} + N_1) \\ (M_2 \Phi_{2,1} + N_2)\Phi_{1,0} \\ (M_3 \Phi_{3,2} + N_3)\Phi_{2,0} \\ \cdot \\ \cdot \\ \cdot \\ (M_{36}\Phi_{36,35} + N_{36})\Phi_{35,0} \end{bmatrix} \qquad (36 \times 16)$$

$$B_1 =$$

$$\begin{bmatrix} M_1 & \underline{0} & \underline{0} & & \underline{0} \\ (M_2\Phi_{2,1} + N_2) & M_2 & \underline{0} & & \underline{0} \\ (M_3\Phi_{3,2} + N_3)\Phi_{2,1} & (M_3\Phi_{3,2} + N_3) & M_3 & & \underline{0} \\ \cdot \quad \cdot & \cdot & \cdot & & \cdot \\ \cdot \quad \cdot & \cdot & \cdot & & \cdot \\ \cdot \quad \cdot & \cdot & \cdot & & \cdot \\ (M_{36}\Phi_{36,35} + N_{36})\Phi_{35,1} & (M_{36}\Phi_{36,35} + N_{36})\Phi_{35,2} & (M_{36}\Phi_{36,35} + N_{36})\Phi_{35,3} & \cdots & M_{36} \end{bmatrix}$$

$$(36 \times 36 \cdot 16)$$

$$G_1 = \begin{bmatrix} g_0 \\ g_1 \\ \cdot \\ \cdot \\ \cdot \\ g_{35} \end{bmatrix} \qquad (36 \cdot 16 \times 1)$$

Since

$$\delta \underline{\bar{Y}}_1 = B_1 G_1 + \begin{bmatrix} \delta y_1 \\ \cdot \\ \cdot \\ \cdot \\ \delta y_{36} \end{bmatrix}$$

the covariance of $\delta \underline{\bar{Y}}$ is

$$V_1 = B_1 \bar{Q}_1 B_1' + \bar{R}_1 \qquad (36 \times 36)$$

where

$$\bar{Q}_1 = \begin{bmatrix} Q_0 & 0 & 0 & \cdots & 0 \\ 0 & Q_1 & 0 & \cdots & 0 \\ 0 & 0 & Q_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & Q_{35} \end{bmatrix} \qquad (36 \times 36)$$

$$\bar{R}_1 = \begin{bmatrix} R_1 & 0 & \cdots & 0 \\ 0 & R_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_{36} \end{bmatrix} \qquad (36 \times 36)$$

The next operation is to find the gain matrix W, which is

$$W_1 = PA'(A_1 P A_1' + V_1)^{-1}$$

where P is the usual error covariance matrix for the state variable. Note that one has to invert a 36 x 36 matrix to evaluate W. Also, within this equation there are large matrix multipliers which are time consuming. Now the smoothed estimate will be

$$\hat{x}(1/36) = \Phi_{1,0}\hat{x}(0/0) + W_1[\underline{\bar{Y}}_1 - \hat{\underline{\bar{Y}}}_1]$$

where

$$\hat{\underline{\bar{Y}}}_1 = A_1\hat{x}(0/0) \quad .$$

To obtain the smoothed estimate x(2/36) the same procedure is followed.

$$x(2) = \Phi_{2,1}x(1) + g_1$$

$$\underline{\bar{Y}}_2 = A_2 x(1) + \delta\underline{\bar{Y}}_2 \quad .$$

Arranging the measurement equation into the desired form implies that

$$A_2 = \begin{bmatrix} (M_2\Phi_{2,1} + N_2) \\ (M_3\Phi_{3,2} + N_3)\Phi_{2,1} \\ \vdots \\ (M_{36}\Phi_{36,35} + N_{36})\Phi_{35,1} \end{bmatrix} \qquad (35 \times 16)$$

$$B_2 = \begin{bmatrix} M_2 & \underline{0} & \cdots & \underline{0} \\ (M_3\Phi_{3,2} + N_3) & M_3 & \cdots & \underline{0} \\ \vdots & \vdots & \ddots & \vdots \\ (M_{36}\Phi_{36,35}+N_{36})\Phi_{35,2} & (M_{36}\Phi_{36,35}+N_{36})\Phi_{35,3} & \cdots & M_{36} \end{bmatrix}$$

$$(35 \times 35 \cdot 16)$$

$$G_2 = \begin{bmatrix} g_1 \\ \vdots \\ g_{35} \end{bmatrix} \qquad (35 \cdot 16 \times 1)$$

Now one can go ahead and find the gain matrix and write down the smoothed estimate. It should be pointed out that the sizes of the matrices had been reduced which will relieve slightly the computational difficulties.

This method is to be continued until all the desired smoothed estimates are found. The last step should just be the nominal filtering equations for the delayed state model and the size of the matrices involved in the computation will be that of the system itself. Until this step inverse operations and matrix multiplies were carried out on large matrices causing this method to be inferior when compared to the smoothing scheme derived earlier.

VII. THE USE OF SMOOTHING IN AN INTEGRATED INERTIAL/DOPPLER--SATELLITE NAVIGATION SYSTEM

The block diagram of Figure 3 summarizes the system used in this example, which was presented by Brown (2). This system was flight tested and the inertial and Doppler satellite systems were operated independently and data from each was recorded. Also, the true error curves were obtained by accurate radar or check points makes this system valuable as an example.

The idea of the Transit system is to aid the inertial system by passing on position information. At the present, the navigation satellites are circling the earth giving their positions as they pass. Of course, under this arrangement there will times when the inertial system is out of range of any of the circling satellites. During this time the Kalman filter for the delayed state model propagates the errors just through the dynamics until another satellite pass occurs.

The properties of the inertial system are:

1. The system is basically terrestial (near-earth), and the vertical (altitude) channel is implemented by other than inertial means.
2. The inertial system is strapped-down. This means the body mounted and the computer coordinate frames

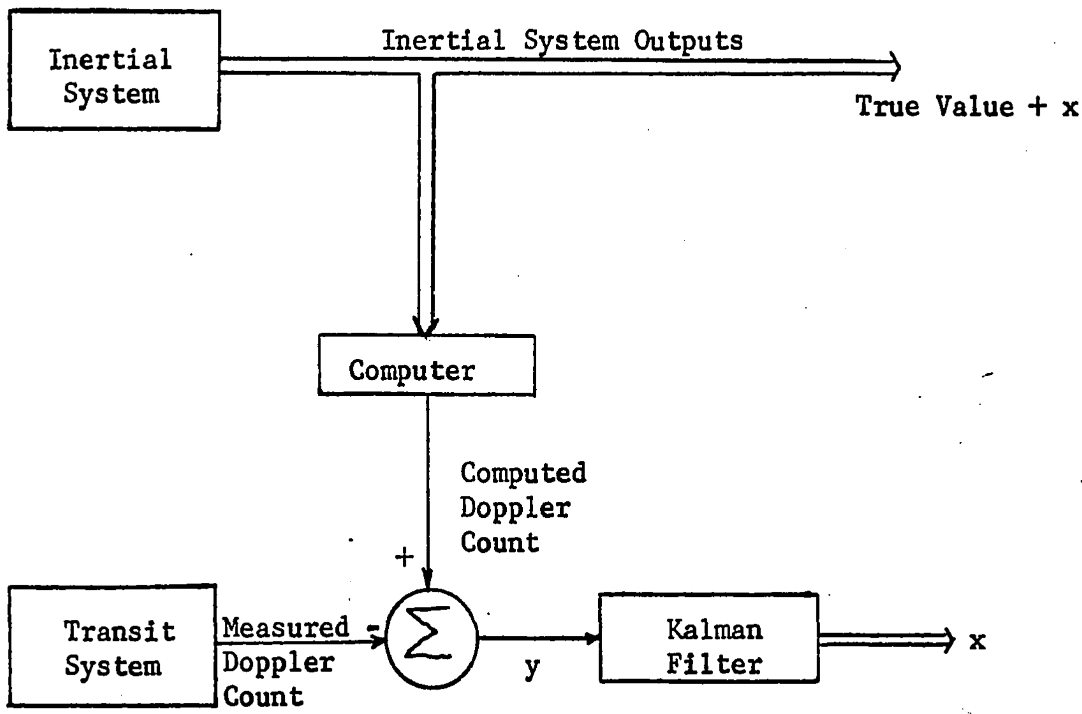

Figure 3. Block diagram for aided inertial system

are related by a direction-cosine matrix that is continuously updated.

Using the theory of error propagation in inertial systems given by Pitman (18), the system states are defined as follows (2):

$$\left.\begin{array}{l} x_1 = \psi_x \\ x_2 = \psi_y \\ x_3 = \psi_z \\ x_4 = \delta\Theta_x \\ x_5 = \delta\Theta_x/\omega_0 \\ x_6 = \delta\Theta_y \\ x_7 = \delta\dot{\Theta}_y/\omega_0 \end{array}\right\} \quad \text{from the } \psi\text{-equation and Schuler dynamics } (R = \text{earth radius, } g = \text{gravity constant, and } \omega_0^2 = g/R)$$

$$x_8 = \delta R_z/R \qquad \text{altitude error}$$

$$\left.\begin{array}{l} x_9 = \varepsilon_x'/\omega_0 \\ x_{10} = \varepsilon_y'/\omega_0 \\ x_{11} = \varepsilon_z'/\omega_0 \end{array}\right\} \quad \text{body-mounted gyro biases}$$

$$\left.\begin{array}{l} x_{12} = \delta a_x'/R\omega_0^2 \\ x_{13} = \delta a_y'/R\omega_0^2 \\ x_{14} = \delta a_z'/R\omega_0^2 \end{array}\right\} \quad \text{body-mounted accelerometer biases}$$

$$x_{15} = \delta\dot{R}_z/R\omega_0 \qquad \text{altitude rate error}$$

$$x_{16} = \delta N \qquad \text{Doppler-count bias error}$$

The gyro drifts and the accelerometer errors are modeled as slowly varying Markov processes. The Kalman filter input modeled by Brown (2) is

$$y_k = M_k x(k) + N_k x(k-1) + \delta y_k$$

where

$$M_k = [0 \;\; 0 \;\; 0 \;\; b_k \;\; 0 \;\; c_k \;\; 0 \;\; Ra_k \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 1]$$

$$N_k = [0 \;\; 0 \;\; 0 \;\; -b_{k-1} \;\; 0 \;\; -c_{k-1} \;\; 0 \;\; -Ra_{k-1} \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0]$$

The parameters in the M_k and N_k matrices are dependent on the position of the inertial system and the satellite coordinates. Thus this problem is non-linear and the Kalman filter should not be optimal. Some method had to be used to take away the nonlinear aspect of the system. One way would be to use the inertial system data and calculate the a, b, c parameter for each new estimate. The accuracy of this correction method can be checked very nicely by using the true error data obtained from the flight test. It turned out this method did not give enough accuracy.

Recent analytical studies made by Brown have shown that if the a, b, c parameters are calculated with a corrected position the Kalman filter estimation of the position errors are very close to the actual error. In an attempt to correct the position data a smoothing scheme was implemented into the system.

The system will not include the Kalman filter with the a, b, c parameters computed with the raw inertial data. Then reprocessing all the satellite data recursively with the smoothing scheme, better estimates of position are obtained. Using these better estimates the a, b, c parameters can be remcomputed and the Kalman filter rerun. It was found that after two cycles of filtering and then smoothing the estimates of the errors were very close to those obtained using the true latitude and longitude in the a, b, c parameter computation.

Figures 4 and 5 indicate the estimation error in latitude and longitude channels of the system during the satellite pass. The curves were obtained by using the iteration technique discussed above. One can see that after two iterations the error curves coincide for each channel. The subscripts in Figures 4 and 5 indicate the order in which the two types of curves were generated. The order of the curves generated by the iterative program is A_1, B_1, A_2, B_2, and A_3.

Figures 6 and 7 indicate the error curves for a whole flight. These graphs indicate the accuracies one may expect from iteration technique previously discussed.

To implement the smoothing scheme, a subroutine plus iteration logic was added to the filtering program used for this example. The iterations were made easier because all the dynamical data for computing the transition matrices and the covariance matrices for the system noise vectors were on tape. Therefore, it was just a matter of rewinding the tape to iterate instead of storing all the data in the computer. Also, the fact that the measurement equation was a scalar made the filtering and smoothing computationally easier because the inverses in these algorithms were trivial. Figure 8 is a rough flow chart of the computer program used in this example. Figure 9 is a flow chart for the program used to compute the smoothed estimates. In this subroutine the values of $z(k,n)$ are computed starting from the last point in the pass to the first point of the pass. And then, the smoothed estimates are computed starting at the first point of the pass to the last point of the pass. This procedure was outlined when the smoothing equations were developed.

The block diagram of Figure 10 summarizes the new system.

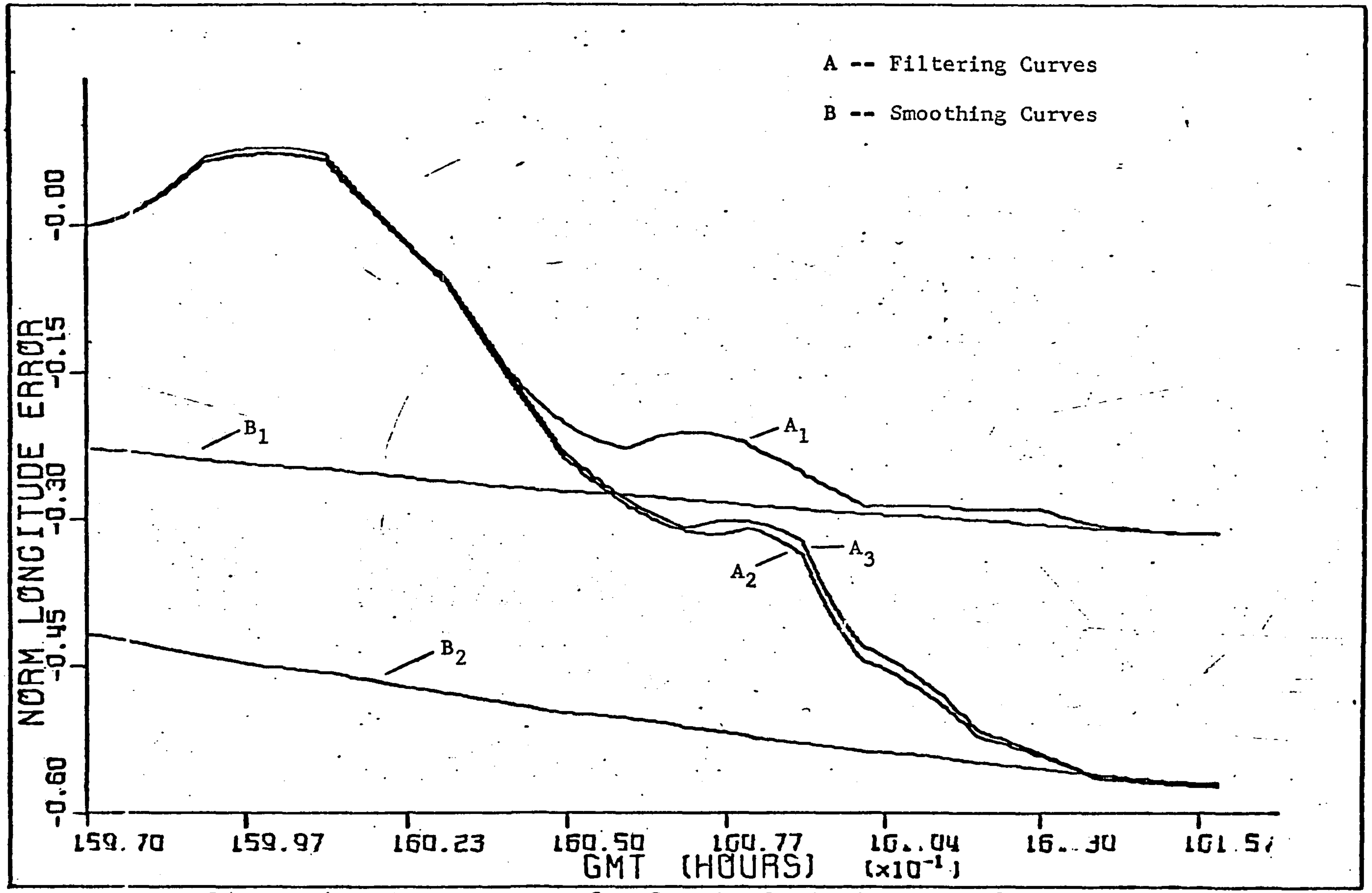

Figure 4. Error curves for longitude during satellite pass

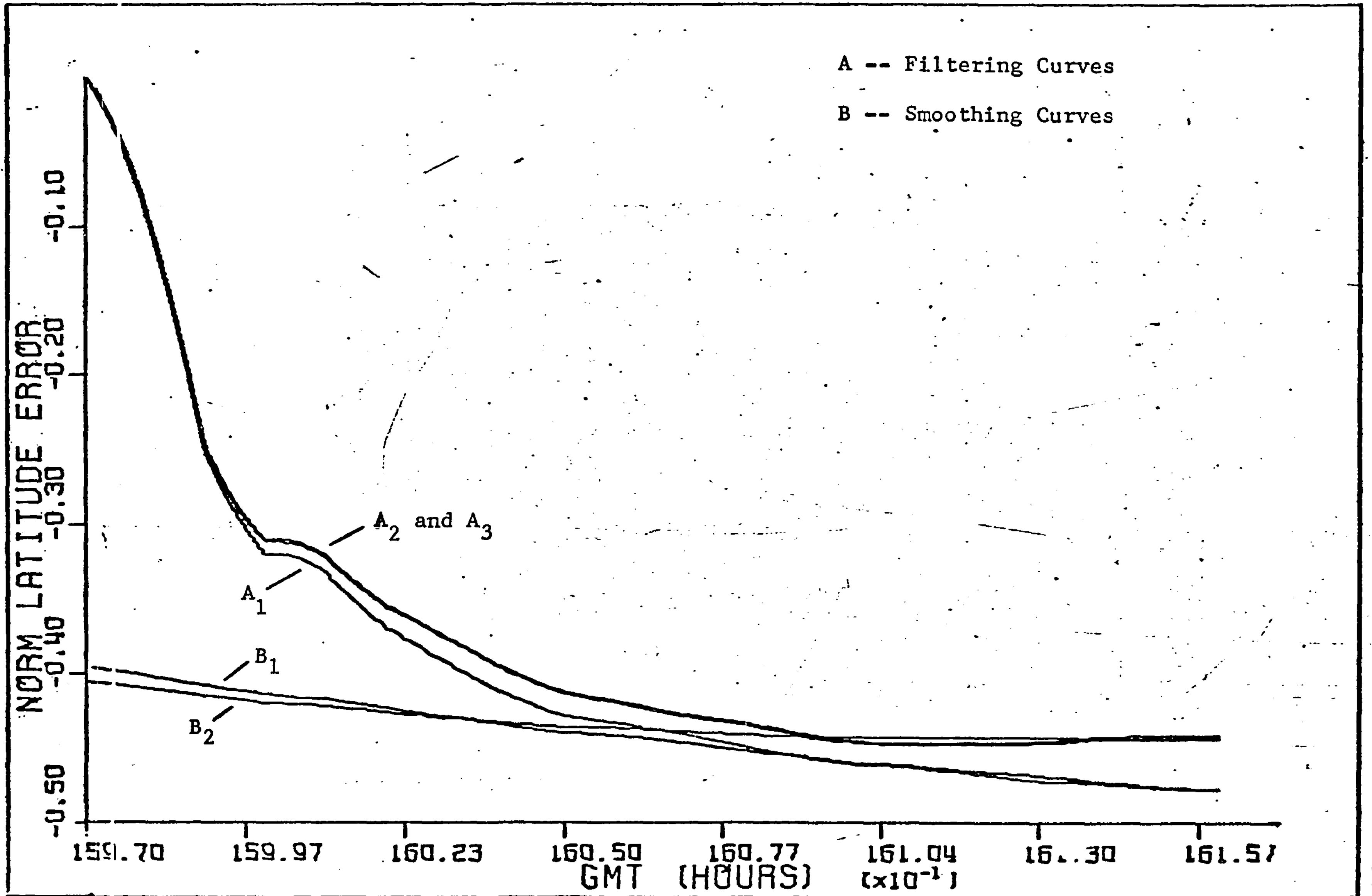

Figure 5. Error curves for latitude during satellite pass

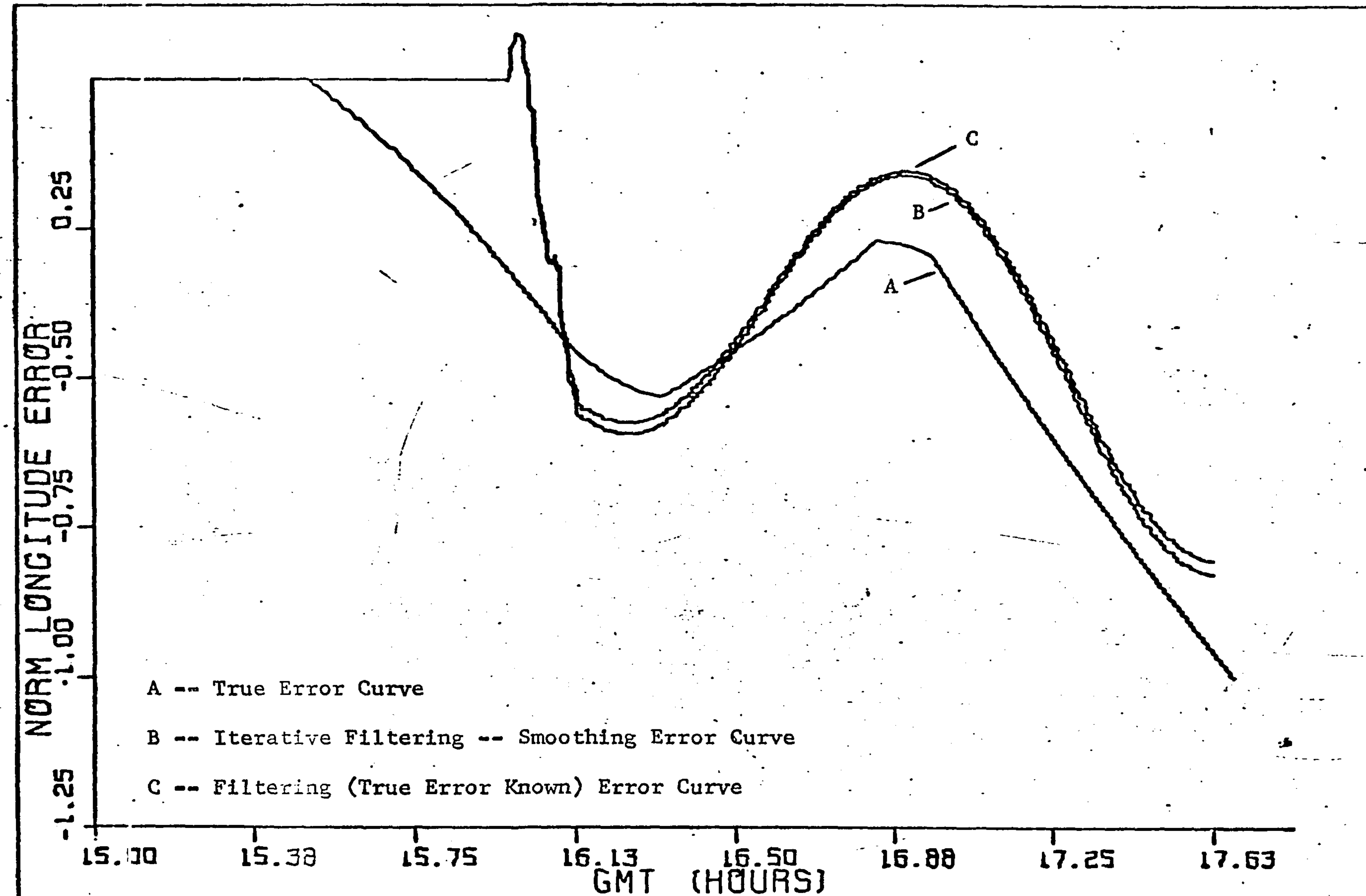

Figure . Error curves for first 2 1/2 hours of flight

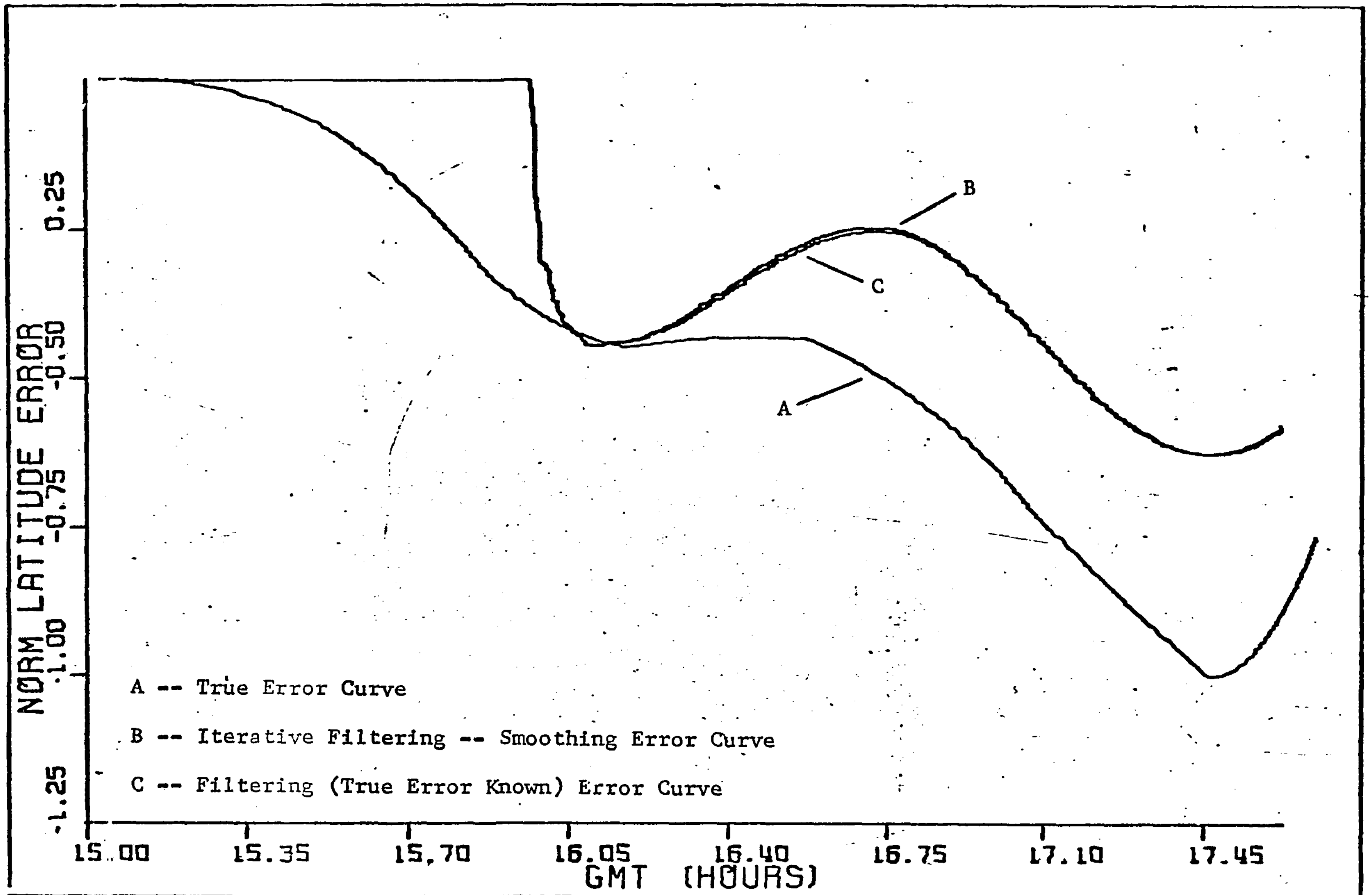

Figure 7. Error curves for first 2 1/2 hours of flight

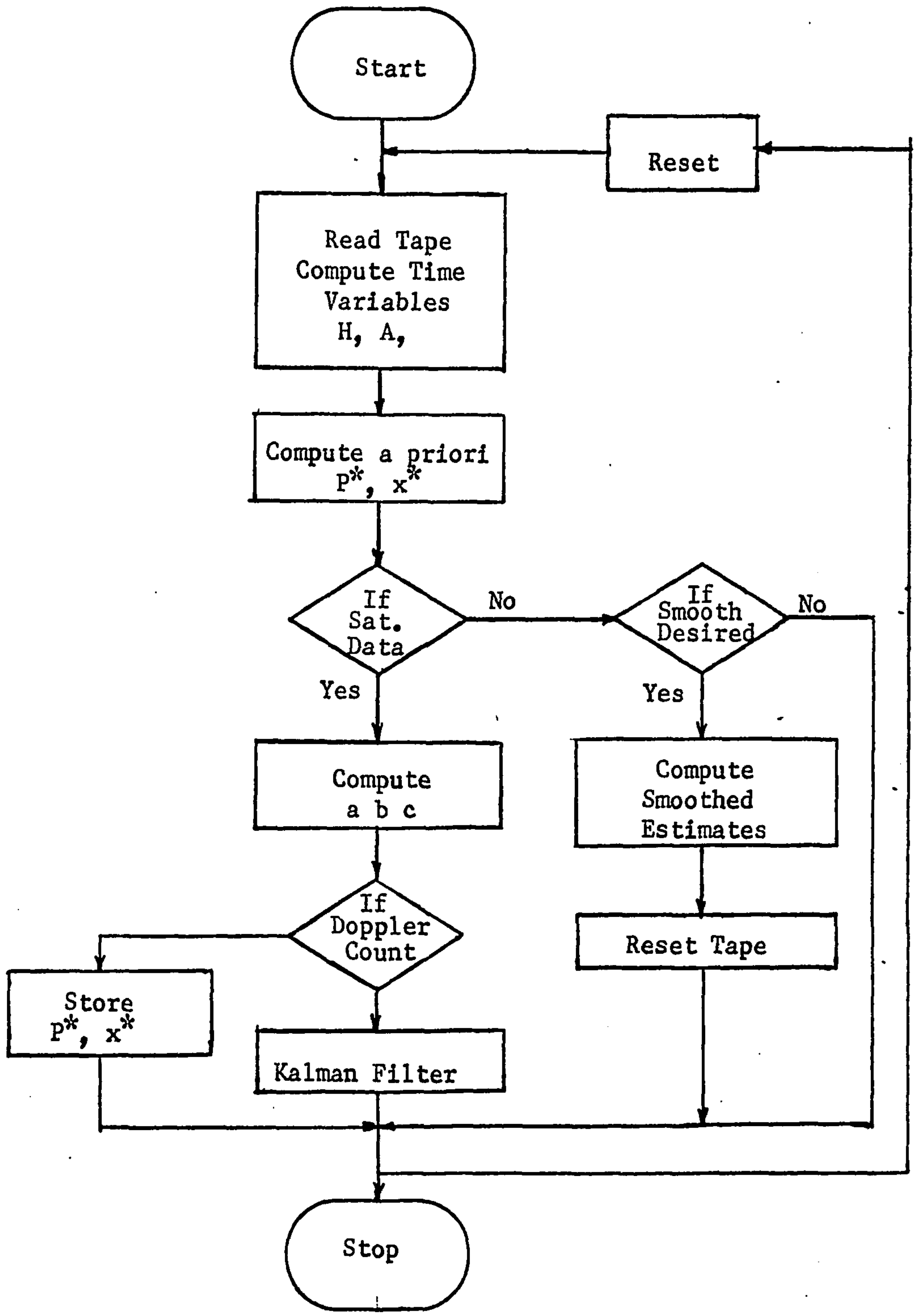

Figure 8. Flow chart for computer program

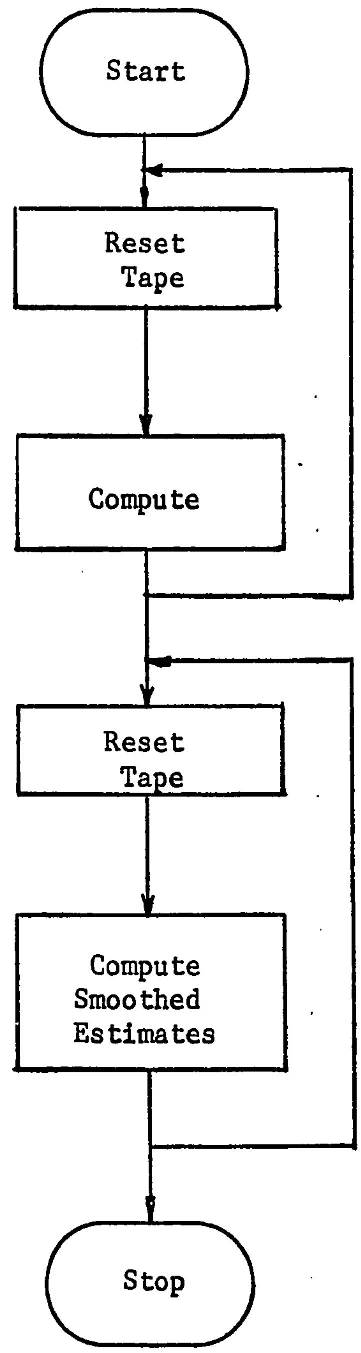

Figure 9. Flow chart for smoothing computer program

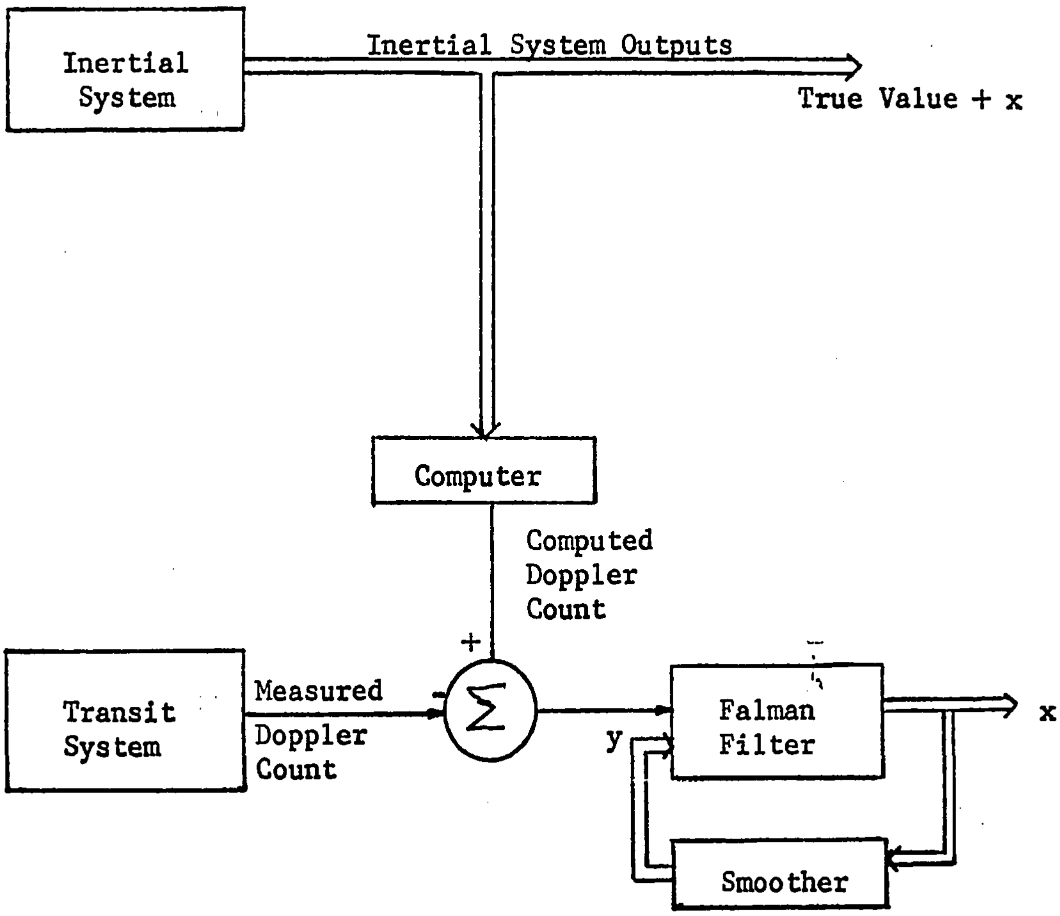

Figure 10. Block diagram for aided inertial system with smoother

VIII. CONCLUSIONS

When averaging is involved in the measurement process, the delayed state automatically appears in the measurement model. Therefore, the Kalman filtering equations have been developed for the delayed state model. As has been mentioned before, this type of modeling can be used quite nicely for certain aided inertial nagivation systems.

The importance of the delayed state model suggests that there could be a need for more than just the filtering equations. Therefore, in this work the smoothing equations for the delayed state model were developed. Realizing that in many of the systems being used, the measurement vector is smaller in size than the state vector; the smoothing equations were derived to involve inverse operations on matrices of the order of the measurement vector. The trade-off being that all the measurement matrices must be remembered.

One of the inherent problems of the smoothing algorithm in general could be eliminated by proper use of the smoothing equations derived in this work. In most smoothing schemes some provision must be made to step the covariance matrix backwards in time. Upon carrying this out one must invert a matrix of the system size; hence, defeating the reason for writing the smoothing equations in the stated manner. But, by proper ordering of the computation of the terms in the smoothing equation presented in this work, the covariance matrix can be stepped forward in time. The method or equations for this are obtained directly from the filtering equations. This procedure is illustrated in the example in Chapter VII.

In the process of modeling systems in the Kalman format, extra states could be defined. Friedland (7) has discussed a method for decoupling the recursive equations when bias states are present in the system equations. He has taken the serial computation problem, that presented by the augmented state assignment, and decoupled it into two computations in parallel. His method could amount to some savings in computation.

Of course, the advantages of the above decoupling scheme should still be present for the delayed state modeling. The delayed state occurring in the measurement offered a formidable equation obstacle to the mathematical development of the decoupled equations. By being consistent with Friedland's notation, the equations derived for the delayed state model look very much like those derived by Friedland.

The application of the above mentioned decoupling sounds rather restrictive. However, it must be remembered that slowly varying noise could be thought of as bias states over a relatively short period of time. In the case of the smoothing equations, the decoupling aspect remains restrictive unless one's budget is very limited. This is because the general philosophy behind smoothing makes it an off-line or after the fact computation. Therefore, there are no time or computer size restrictions on the smoothing calculations. This means that one would not want to degrade the models of the systems just to take advantage of a computation method. However, changing the conditions under which the smoothing computation must be made may yield a more favorable attitude toward the remodeling of some systems.

An interesting side light to the whole idea of smoothing has arisen. In the example presented in Chapter VII, the smoothing equations were used to help solve a nonlinear problem. The idea of using smoothing algorithms to help solve nonlinear problems needs much more investigation, but has been demonstrated to work in the example presented. After two smoothing and filter iterations it was found that the solution was as "good" as could be obtained by using the already known true error to get rid of the nonlinear aspect of the problem. The speed at which the iteration method converged is due to "goodness" of the initial system model. The position errors were relatively small as compared to the vehicle's actual position on the earth. Therefore, like the second order steepest descent method (12), convergence occurs only if the initial approximation is sufficiently close to the solution.

IX. LITERATURE CITED

1. Bode, H. W. and Shannon, C. E., "A simplified derivation of linear least square smoothing and prediction theory," Proc. IRE, vol. 38, pp. 417-425, April, 1950.

2. Brown, R. G., "Analysis of an integrated inertial/ Doppler-satellite navigation system," Part I ERI-62600, Iowa State University, Ames, Iowa, December, 1969.

3. Brown, R. G. and Hartmann, G. L., "Kalman filter with delayed states as observables," ERI-279, Iowa State University, Ames, Iowa, July, 1968.

4. Bryson, A. E. and Ho, Y., Applied Optimal Control, Waltham, Mass.: Blaisdel, 1969.

5. Cox, H., "On the estimation of state variables and parameters for noisy dynamic systems," IEEE Trans. Autonatic Control, vol. AC-9, pp. 5-12, January, 1964.

6. Deutsch, R., Estimation Theory, Englewood Cliffs, N.J.: Prentic-Hall, 1965.

7. Friedland, B., "Treatment of bias in recursive filtering," IEEE Trans. Autonatic Control, vol. AC-14, pp. 359-366, August, 1969.

8. Galles, W. B., "Numerical feasible solutions to the discrete optimal smoothing problems," Memorandum 48-509-M, August, 1966.

9. Griffin, R. E. and Sage, A. P., "Sensitivity analysis of discrete filtering and smoothing algorithms," AIAAJ, vol. 7, No. 10, pp. 1890-1897, October, 1969.

10. Kalman, R. E., "A new approach to linear filtering and prediction problems," Trans. ASME J. Basic Eng., vol. 82D, pp. 35-45, 1960.

11. Lee, R. C. K., Optimal Estimation, Identification and Control, Cambridge, Mass.: MIT Press, 1964.

12. Luenberger, D. G., Optimization by Vector Space Methods, New York: Wiley, 1969.

13. Meditch, J. S., Stochastic Optimal Linear Estimation and Control, New York: McGraw-Hill, 1969.

14. Mood, A. M. and Graybill, F. A., Introduction to the Theory of Statistics, New York: McGraw-Hill, 1963.

15. Nahi, N. E., Estimation Theory and Applications, New York: Wiley, 1969.

16. Parzen, E., "A new approach to the synthesis of optimal smoothing and prediction problem," Applied Math. Stat. Lab., Stanford University, Stanford, California, 1960.

17. Perlis, S., Theory of Matrices, Cambridge, Mass.: Addison-Wesley, 1952.

18. Pitman, G. R., Jr., Ed., Inertial Guidance, New York: Wiley, 1962.

19. Rauch, H. E., Tung, F. and Striebel, C. T., "Maximum likelihood estimation of linear dynamic systems," AIAA Journal, vol. 3, pp. 1445-1450, August, 1965.

20. Sage, A. P., Optimum Systems Control, Englewood Cliffs, N.J.: Prentice-Hall, 1968.

21. Sorenson, H. W., "Kalman filtering technique," in Advances in Control Systems, vol. 3, C. T. Leondes, Ed., New York: Academic Press, 1966.

22. Wiener, N., The Extrapolation, Interpolation and Smoothing of Stationary Time Series. New York: Wiley, 1949.

23. Wayne, D. Q., "A solution of the smoothing problem for linear dynamic systems," Automatica, vol. 4, pp. 73-92, 1966.

X. ACKNOWLEDGMENTS

The author would like to thank Dr. R. G. Brown for his encouragement and help during the development stage of this work. Thanks is also due to Mr. Larry Levy for his inspiration, Mr. Donely Winger for his computer program for the delayed-state Kalman filter, and Mr. Terry Cline for his plotting program.

Special thanks is due to the author's wife, both for her help in proofreading and her patience during the time necessary to complete this research.

This project was supported in part by the Engineering Research Institute under Themis contract N00014-68-A-0162. The author also wishes to acknowledge stipend support from a National Defense Education Act Fellowship during a large portion of his graduate program.

XI. APPENDIX A

Consider the covariance equation

$$P(k+1/k) = F_k[I - W(k)L_k]P(k/k-1)F_k' - F_kW(k)J_kP(k-1/k-1)F_{k-1}'F_k' + GQ_kG'$$

and

$$\tilde{P}(k+1/k) = F_k[I - \tilde{W}(k)L_k]\tilde{P}(k/k-1)F_k' - F_k\tilde{W}(k)J_k\tilde{P}(k-1/k-1)F_{k-1}'F_k' + GQ_kG'$$

where, from the derivation in Chapter III,

$$S_k = P(k/k-1) - \tilde{P}(k/k-1) = U(k)M(k)U'(k)$$

$$S_{k+1} = P(k+1/k) - \tilde{P}(k+1/k) = U(k+1)M(k+1)U'(k+1) \quad .$$

Note that the x subscript has been omitted for convenience, therefore,

$$\begin{aligned} S_{k+1} &= F_k[P(k/k-1) - \tilde{P}(k/k-1)]F_k' - F_k[W(k)L_kP(k/k-1) \\ &\quad - \tilde{W}(k)L_k\tilde{P}(k/k-1)]F_k' - F_k[W(k)J_kP(k-1/k-1)F_{k-1}' \\ &\quad - \tilde{W}(k)J_k\tilde{P}(k-1/k-1)F_{k-1}']F_k' \\ &= F_k\{S_k - W(k)[L_k + J_kF_{k-1}^{-1}]P(k/k-1) + W(k)J_kF_{k-1}^{-1}GQ_kG' \\ &\quad + \tilde{W}(k)[L_k + J_kF_{k-1}^{-1}]\tilde{P}(k/k-1) - \tilde{W}(k)J_kF_{k-1}^{-1}GQ_kG'\}F_k' \end{aligned}$$

where

$$W(k) = P(k/k-1)[L_k' + (F_{k-1}')^{-1}J_k']D_k^{-1} - GQ_kG'(F_{k-1}')^{-1}J_k'D_k^{-1}$$

$$\tilde{W}(k) = \tilde{P}(k/k-1)[L_k' + (F_{k-1}')^{-1}J_k']\tilde{D}_k^{-1} - GQ_kG'(F_{k-1}')^{-1}J_k'\tilde{D}_k^{-1}$$

$$E_k = L_k + J_kF_{k-1}^{-1}$$

$$P(k/k-1) = \tilde{P}(k/k-1) + S_k \quad .$$

Making substitutions, S_{k+1} becomes

$$\begin{aligned} S_{k+1} = F_k\{ & S_k - S_kE_k'D_k^{-1}E_kS_k - S_kE_k'D_k^{-1}E_kP(k/k-1) \\ & + S_kE_k'D_k^{-1}J_kF_{k-1}{}^{-1}GQ_kG' - \tilde{P}(k/k-1)E_k'D_k^{-1}E_kS_k \\ & + GQ_kG'(F_{k-1}')^{-1}J_k'D_k^{-1}E_kS_k + \tilde{P}(k/k-1)E_k'[\tilde{D}_k^{-1} - D_k^{-1}] \\ & \cdot E_k\tilde{P}(k/k-1) - \tilde{P}(k/k-1)E_k'[\tilde{D}^{-1} - D_k^{-1}]J_kF_{k-1}^{-1}GQ_kG' \\ & - GQ_kG'(F_{k-1}')^{-1}J_k[\tilde{D}_k^{-1} - D_k^{-1}]E_k\tilde{P}(k/k-1) \\ & + GQ_kG'(F_{k-1}')^{-1}J_k'[\tilde{D}_k^{-1} - D_k^{-1}]J_kF_{k-1}^{-1}GQ_kG'\}F_k' \quad . \end{aligned}$$

Now, if it is assumed that

$$U(k+1) = F_k[I - \tilde{W}(k)E_k]U(k)$$

and

$$M(k+1) = M(k) - M(k)U(k)'E_k'D_k^{-1}E_kU(k)M(k)$$

then

$$S_{k+1} = U(k+1)M(k+1)U(k+1) = F_k\{(I - \tilde{W}(k)E_k)[S_k - S_kE_k'D_k^{-1}E_kS_k]$$

$$\cdot (I - \tilde{W}(k)E_kJ'\} F_k'$$

$$= F_k\{S_k - S_kE_k'D_k^{-1}E_kS_k + \tilde{P}(k/k-1)E_k'[\tilde{D}^{-1}E_kS_kE_k'D_k^{-1} - \tilde{D}_k^{-1}]E_kS_k$$

$$+ GQ_kG'(F_{k-1}')^{-1}J_k'[\tilde{D}_k^{-1} - \tilde{D}_k^{-1}E_kS_kE_k'D_k^{-1}]E_kS_k$$

$$+ S_kE_k'[D_k^{-1}E_kS_kE_k'\tilde{D}_k^{-1} - \tilde{D}_k^{-1}]E_k\tilde{P}(k/k-1)$$

$$+ S_kE_k'[\tilde{D}_k^{-1} - D_k^{-1}E_kS_kE_k\tilde{D}_k^{-1}]J_kF_{k-1}^{-1}GQ_kG'$$

$$+ \tilde{P}(k/k-1)E_k'[\tilde{D}_k^{-1}E_kS_kE_k'\tilde{D}_k^{-1} - \tilde{D}_k^{-1}E_kS_kE_k'D_k^{-1}E_kS_kE_k'\tilde{D}_k^{-1}]$$

$$\cdot E_k\tilde{P}(k/k-1) + \tilde{P}(k/k-1)E_k'[\tilde{D}_k^{-1}E_kS_kE_k'D_k^{-1}E_kS_kE_k'\tilde{D}_k^{-1}$$

$$- \tilde{D}_k^{-1}E_kS_kE_k'\tilde{D}_k^{-1}]J_kF_{k-1}^{-1}GQ_kG'$$

$$+ GQ_kG'(F_{k-1}')^{-1}J_k'[\tilde{D}_k^{-1}E_kS_kE_k'D_k^{-1}E_kS_kE_k'\tilde{D}_k^{-1}$$

$$- \tilde{D}_k^{-1}E_kS_kE_k'\tilde{D}_k^{-1}]E_k\tilde{P}(k/k-1) + GQ_kG'(F_{k-1}')^{-1}J_k'[\tilde{D}_k^{-1}E_kS_kE_k'\tilde{D}_k^{-1}$$

$$- \tilde{D}_k^{-1}E_kS_kE_k'D_k^{-1}E_kS_kE_k'\tilde{D}_k^{-1}]J_kF_{k-1}^{-1}GQ_kG'\}F_k' \quad .$$

Since the two equations for S_{k+1} are equal, their difference must be zero

$$S_{k+1} - S_{k+1} = F_k\{S_kE_k'[D_k^{-1} - \tilde{D}_k^{-1} + D_k^{-1}E_kS_kE_k'D_k^{-1}]E_k\tilde{P}(k/k-1)$$

$$- S_kE_k'[D_k^{-1} - \tilde{D}_k^{-1} + D_k^{-1}E_kS_kE_k'\tilde{D}_k^{-1}]J_kF_{k-1}^{-1}GQ_kG'$$

$$+\ \tilde{P}(k/k-1)E_k'[D_k^{-1} - \tilde{D}_k^{-1} + \tilde{D}_k^{-1}E_kS_kE_k'D_k^{-1}]E_kS_k$$

$$-\ GQ_kG'(F_{k-1}')^{-1}J_k'[D_k^{-1} - \tilde{D}_k^{-1} + \tilde{D}_kE_kS_kE_k'D_k^{-1}]E_kS_k$$

$$+\ \tilde{P}(k/k-1)E_k'[D_k^{-1} - \tilde{D}_k^{-1} + \tilde{D}_k^{-1}E_kS_kE_k'\tilde{D}_k^{-1}$$

$$-\ \tilde{D}_k^{-1}E_kS_kE_k'D_k^{-1}E_kS_kE_k'\tilde{D}_k^{-1}]E_k\tilde{P}(k/k-1) - \tilde{P}(k/k-1)E_k'[D_k^{-1}$$

$$-\ \tilde{D}_k^{-1} + \tilde{D}_k^{-1}E_kS_kE_k'\tilde{D}_k^{-1} - \tilde{D}_k^{-1}E_kS_kE_k'D_k^{-1}E_kS_kE_k'\tilde{D}_k^{-1}]J_kF_{k-1}^{-1}GQ_kG'$$

$$-\ GQ_kG'(F_{k-1}')^{-1}J_k'[D_k^{-1} - \tilde{D}_k^{-1} + \tilde{D}_k^{-1}E_kS_kE_k'\tilde{D}_k^{-1}$$

$$-\ \tilde{D}_k^{-1}E_kS_kE_k'D_k^{-1}E_kS_kE_k'\tilde{D}_k^{-1}]E_k\tilde{P}(k/k-1)$$

$$+\ GQ_kG'(F_{k-1}')^{-1}J_k'[D_k^{-1} - \tilde{D}_k^{-1} + \tilde{D}_k^{-1}E_kS_kE_k'\tilde{D}_k^{-1}$$

$$-\ \tilde{D}_k^{-1}E_kS_kE_k'D_k^{-1}E_kS_kE_k'\tilde{D}_k^{-1}]J_kF_{k-1}^{-1}GQ_kG'\}F_k'$$

$$= 0 \quad .$$

To show that the above equation is true, it must be shown that

$$D_k^{-1} - \tilde{D}_k^{-1} + D_k^{-1}E_kS_kE_k'\tilde{D}_k^{-1} = 0$$

$$D_k^{-1} - \tilde{D}_k^{-1} + \tilde{D}_k^{-1}E_kS_kE_k'D_k^{-1} = 0$$

$$D_k^{-1} - \tilde{D}_k^{-1} + \tilde{D}_k^{-1}E_kS_kE_k'\tilde{D}_k^{-1} - \tilde{D}_k^{-1}E_kS_kE_k'D_k^{-1}E_kS_kE_k\tilde{D}_k^{-1} = 0 \quad .$$

The following identity is needed to show the above equations:

$$D_k = L_k P(k/k-1) L_k' + L_k [P(k/k-1) - GQ_k G\ ](F_{k-1}')^{-1} J_k'$$

$$+ J_k F_{k-1}^{-1} [P(k/k-1) - GQ_k G'] L_k'$$

$$+ J_k F_{k-1}^{-1} [P(k/k-1) - GQ_k G'] (F_{k-1}')^{-1} J_k' + R_k$$

$$= \tilde{D}_k + L_k S_k L_k' + L_k S_k (F_{k-1}')^{-1} J_k' + J_k F_{k-1}^{-1} S_k L_k'$$

$$+ J_k F_{k-1}^{-1} S_k (F_{k-1}')^{-1} J_k'$$

$$= \tilde{D}_k + E_k S_k E_k' \quad .$$

The second equation reduces as follows

$$D_k^{-1} - \tilde{D}_k^{-1} + \tilde{D}_k^{-1} E_k S_k E_k' D_k^{-1} = D_k^{-1} - \tilde{D}_k^{-1} (I - E_k S_k E_k' D_k^{-1})$$

$$= D_k^{-1} - \tilde{D}_k^{-1} (D_k - E_k S_k E_k') D_k^{-1} = D_k^{-1} - \tilde{D}_k^{-1} \tilde{D}_k D_k^{-1}$$

$$= 0 \quad .$$

The first equation reduces as follows

$$D_k^{-1} + D_k^{-1} E_k S_k E_k' \tilde{D}_k^{-1} - \tilde{D}_k^{-1} = D_k^{-1} + D_k^{-1} (E_k S_k E_k' - D_k) \tilde{D}_k^{-1}$$

$$= D_k^{-1} - D_k^{-1} \tilde{D}_k \tilde{D}_k^{-1} = 0 \quad .$$

The last equation reduces as follows

$$D_k^{-1} - \tilde{D}_k^{-1} + \tilde{D}_k^{-1} E_k S_k E_k' \tilde{D}_k^{-1} - \tilde{D}_k^{-1} E_k S_k E_k' D_k^{-1} E_k S_k E_k' \tilde{D}_k^{-1}$$

$$= \tilde{D}_k^{-1} E_k S_k E_k' \tilde{D}_k^{-1} - D_k^{-1} E_k S_k E_k \tilde{D}_k^{-1} - \tilde{D}_k^{-1} E_k S_k E_k' D_k^{-1} E_k S_k E_k' \tilde{D}_k^{-1}$$

$$= (\tilde{D}_k^{-1} - D_k^{-1}) E_k S_k E_k' \tilde{D}_k^{-1} - \tilde{D}_k E_k S_k E_k' D_k^{-1} E_k S_k E_k' \tilde{D}_k^{-1}$$

$$= [\tilde{D}_k^{-1} - D_k^{-1} - \tilde{D}_k E_k S_k E_k' D_k^{-1}] E_k S_k E_k' \tilde{D}_k^{-1} = 0 \quad .$$

Thus the assumptions made earlier are correct. Therefore,

$$U(k+1) - F_k[I - \tilde{W}(k) E_k] U(k)$$

$$M(k+1) = M(k) - M(k) U'(k) E_k' D_k^{-1} E_k U(k) M(k) \quad .$$

XII. APPENDIX B

Considering the gain component for the state vector x

$$W_x(k) = [P_x(k/k-1)H_k' + P_{xb}(k/k-1)C_k' + A_{k-1}P_x(k-1/k-1)N_k'$$

$$+ B_{k-1}P_{xb}'(k-1/k-1)N_k']D_k^{-1}$$

$$= \{[\tilde{P}_x(k/k-1) + U_x(k)M(k)U_x'(k)]H_k' + U_x(k)M(k)C_k'$$

$$+ A_{k-1}\tilde{P}_x(k-1/k-1)N_k' + A_{k-1}V_x(k-1)M(k)V_x'(k-1)N_k'$$

$$+ B_{k-1}M(k)V_x'(k-1)N_k'\}D_k^{-1}$$

$$= \{\tilde{P}_x(k/k-1)H_k' + A_{k-1}\tilde{P}_x(k-1/k-1)N_k' + U_x(k)M(k)[U_x'(k)H_k'$$

$$+ C_k'] + (A_{k-1}V_x(k-1) + B_{k-1})M(k)V_x'(k-1)N_k'\}D_k^{-1}$$

$$= \tilde{W}_x(k)\tilde{D}_kD_k^{-1} + U_x(k)M(k)T'(k)D_k^{-1} \quad .$$

Using the identity proved in Appendix A and the equations for E_k and $T(k)$, the following reduction can be made.

$$D_k = \tilde{D}_k + E_kS_kE_k' = \tilde{D}_k + E_kU(k)M(k)U'(k)E_k'$$

where

$$E_kU(k) = (L_k + J_kF_{k-1}^{-1})U(k)$$

$$= [H_k \quad C_k] + [N_k \quad 0]\left[\begin{array}{c|cc} A_{k-1} & -A_{k-1}^{-1} & B_{k-1} \\ \hline 0 & & I \end{array}\right]\left[\begin{array}{c} U_x(k) \\ \hline I \end{array}\right]$$

$$= [H_k + N_k A_{k-1}^{-1} \quad C_k - N_k A_{k-1}^{-1} B_{k-1}]\left[\begin{array}{c} U_x(k) \\ \hline I \end{array}\right]$$

$$= [(H_k + N_k A_{k-1}^{-1}) U_x(k) \quad (C_k - N_k A_{k-1}^{-1} B_{k-1})]$$

$$E_k U(k) M(k) U'(k) E_k' = (H_k + N_k A_{k-1}^{-1}) U_x(k) M(k) U_x'(k) (H_k + N_k A_{k-1}^{-1})'$$

$$+ (C_k - N_k A_{k-1}^{-1} B_{k-1}) M(k) (C_k - N_k A_{k-1}^{'} B_{k-1})'$$

$$= [H_k U_x(k) + N_k A_{k-1}^{-1}(U_x(k) - B_{k-1}) + C_k] M(k)$$

$$\cdot [H_k U_x(k) + N_k A_{k-1}^{-1}(U_x(k) - B_{k-1}) + C_k]'$$

$$= T(k) M(k) T'(k) \quad .$$

Therefore,

$$D_k = \tilde{D}_k + T(k) M(k) T'(k) \quad .$$

Using the inverse lemma presented by Sorenson (21) take the inverse of D_k

$$D_k^{-1} = \tilde{D}_k^{-1} - \tilde{D}_k^{-1} (Tk) [T'(k) \tilde{D}_k^{-1} T(k) + M^{-1}(k)]^{-1} T(k)' \tilde{D}_k^{-1} \quad .$$

Using the inverse lemma again on the bracketed quantity

$$[T'(k)\tilde{D}_k^{-1}T(k) + M^{-1}(k)]^{-1} = M(k) - M(k)T'(k)[T(k)M(k)T'(k) + \tilde{D}_k]^{-1}T(k)M(k)$$

$$= M(k) - M(k)T'(k)D_k^{-1}T(k)M(k)$$

and

$$D_k^{-1} = \tilde{D}_k^{-1} - \tilde{D}_k^{-1}T(k)[M(k) - M(k)T'(k)D_k^{-1}T(k)M(k)][T'(k)\tilde{D}_k^{-1}$$

$$= \tilde{D}_k^{-1} - \tilde{D}_k^{-1}T(k)M(k)T'(k)\tilde{D}_k^{-1}$$

$$+ \tilde{D}_k^{-1}T(k)M(k)T'(k)D_k^{-1}T(k)M(k)T'(k)\tilde{D}_k^{-1} \quad .$$

Using these results in the gain equations, it follows that

$$W_x(k) = \tilde{W}_x(k) - \tilde{W}_x(k)T(k)M(k)T'(k)\tilde{D}_k^{-1}$$

$$+ \tilde{W}_x(k)T(k)M(k)T'(k)D_k^{-1}T(k)M(k)T'(k)\tilde{D}_k^{-1}$$

$$+ U_x(k)M(k)T'(k)D_k^{-1} - V_x(k)M(k)T'(k)D_k^{-1}$$

$$+ V_x(k)M(k)T'(k)D_k^{-1}$$

$$= \tilde{W}_x(k) - \tilde{W}_x(k)T(k)M(k)T'(k)\tilde{D}_k^{-1}$$

$$+ \tilde{W}_x(k)T(k)M(k)T'(k)D_k^{-1}[I + T(k)M(k)T'(k)\tilde{D}_k^{-1}]$$

$$+ \ V_x(k)W_b(k)$$

$$= \tilde{W}_x(k) + V_x(k)W_b(k) - \tilde{W}_x(k)T(k)M(k)T'(k)\tilde{D}_k^{-1}$$

$$+ \ \tilde{W}_x(k)T(k)M(k)T'(k)D_k^{-1}[\tilde{D}_k + T(k)M(k)T'(k)]\tilde{D}_k^{-1}$$

$$= \tilde{W}_x(k) + V_x(k)W_b(k) - \tilde{W}_x(k)T(k)M(k)T'(k)\tilde{D}_k^{-1}$$

$$+ \ \tilde{W}_x(k)T(k)M(k)T'(k)D_k^{-1}D_k\tilde{D}_k^{-1}$$

$$= \tilde{W}_x(k) + V_x(k)W_b(k) \quad .$$

XIII. APPENDIX C

In order to write $z_x(k,n)$ as a function of $\tilde{z}_x(k,n)$ a look will be taken at $z_x(n-1/n)$, $z_x(n-2/n)$ and $z_x(n-3/n)$ in an attempt to get at the general expression, $z_x(k,n)$.

For $k = n-1$, the augmented $z(n-1,n)$ will be written as two equations. If

$$z(n-1,n) = (L_n F_{n-1} + J_n)' D_n^{-1} \tilde{y}(n/n-1)$$

then

$$z_x(n-1,n) = (H_n A_{n-1} + N_n)' D_n^{-1} [y_n - (H_n A_{n-1} + N_n)\hat{x}(n-1/n-1) - (H_n B_{n-1} + C_n)\hat{b}(n-1/n-1)]$$

$$z_b(n-1,n) = (H_n B_{n-1} + C_n)' D_n^{-1} [y_n - (H_n A_{n-1} + N_n)\hat{x}(n-1/n-1) - (H_n B_{n-1} + C_n)\hat{b}(n-1/n-1)] \quad .$$

Using the identity for $\tilde{z}_x(n-1,n)$ and the equality that gives D_n as a function of $\tilde{D}_n$, it may be shown that

$$z_x(n-1,n) = \tilde{z}_x(n-1,n) - (H_n A_{n-1} + N_n)' \tilde{D}_n^{-1} T(n)\hat{b}(n/n) \quad .$$

Proceeding in the same manner and using the equation

$$z(n-2,n) = d(n-2,n)z(n-1,n) + (L_{n-1} A_n + J_{n-1})' D_{n-1}^{-1} \tilde{y}(n-1/n-2)$$

it was found that

$$z_x(n-2,n) = \tilde{z}_x(n-2,n) - (H_{n-1}A_{n-2} + N_{n-1})'\tilde{D}_{n-1}^{-1}T(n-1)$$

$$\cdot\ \hat{b}(n-1/n) - \tilde{d}(n-2,n)(H_nA_{n-1} + N_n)'\tilde{D}_n^{-1}T(n)$$

$$\cdot\ \hat{b}(n/n)$$

and proceeding on it was found also that

$$z_x(n-3,n) = \tilde{z}_x(n-3,n) - \tilde{d}(n-3,n-1)\tilde{d}(n-2,n)(H_nA_{n-1} + N_n)'$$

$$\cdot\ \tilde{D}_n^{-1}T(n)\hat{b}(n/n) - \tilde{d}(n-3,n-1)(H_{n-1}A_{n-2} + N_{n-1})'$$

$$\cdot\ \tilde{D}_{n-1}^{-1}T(n-1)\hat{b}(n-1/n) - (H_{n-2}A_{n-3} + N_{n-2})'$$

$$\cdot\ \tilde{D}_{n-2}^{-1}T(n-2)\hat{b}(n-2/n) \quad .$$

The above equations imply that

$$z_x(k+1,n) = \tilde{z}_x(k+1,n) - \sum_{i=k+2}^{n} \tilde{d}(k+1,n)(H_iA_{i-1} + N_i)'\tilde{D}_i^{-1}T(i)\hat{b}(i/n)$$

Using this fact and that

$$z(k,n) = d(k,k+2)z(k+1,n) + (L_{k+1}F_k + J_{k+1})'D_{k+1}^{-1}\tilde{y}(k+1/k)$$

it was found that

$$z_x(k,n) = \tilde{z}_x(k,n) - \sum_{i=k+1}^{n} \tilde{d}(k,n)(H_iA_{i-1} + N_i)'\tilde{D}_i^{-1}T(i)\hat{b}(i/n)$$

thus completing the induction proof. Using the identities of Chapter IV for reducing the fixed interval smoothing equations to a more desirable state, the above equation is reduced to

$$z_x(k,n) = \tilde{z}_x(k,n) - (H_{k+1}A_k + N_k)'\tilde{D}_{k+1}^{-1}T(k+1)\hat{b}(k+1/n)$$

$$- \tilde{d}(k,k+2) \sum_{i=k+2}^{n} \tilde{d}(k+1,i)(H_i A_{i-1} + N_i)'\tilde{D}_i^{-1}T(i)\hat{b}(i/n)$$

$$= \tilde{z}_x(k,n) - e(k+1,n)$$

where

$$e(k+1,n) = \sum_{i=k+1}^{n} \tilde{d}(k,i)(H_i A_{i-1} + N_i)'\tilde{D}_i^{-1}T(i)\hat{b}(i/n)$$

$$= (H_{k+1}A_k + N_{k+1})'\tilde{D}_{k+1}T(k+1)\hat{b}(k+1/n)$$

$$+ \tilde{d}(k,k+2)e(k+2,n) \quad .$$