1971

Kalman filter with complementary constraint and integrated navigation systems applications

Leo Edward Ott
Iowa State University

72-12,580

OTT, Leo Edward, 1942-
KALMAN FILTER WITH COMPLEMENTARY CONSTRAINT AND INTEGRATED NAVIGATION SYSTEMS APPLICATIONS.

Iowa State University, Ph.D., 1971
Engineering, electrical

Kalman filter with complementary constraint and integrated navigation systems applications

by

Leo Edward Ott

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.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa

1971

TABLE OF CONTENTS

I. INTRODUCTION

The purpose of this thesis is to investigate the Kalman filter equations with complementary constraints and its applications. For the benefit of the reader a review of classical filter theory, the Wiener filter problem, and Kalman filter equations will be presented. This will be followed by the concept of a complementary filter and the complementary Kalman filter.

A. Classical Filter Theory

The determination of an appropriate electronic network configuration to yield a given frequency response is usually referred to as classical filter theory. There are basically four types of circuit configurations in this theory: low-pass, high-pass, band-pass, and band-stop filters. These filters are frequency-selective electronic devices that operate on voltage, current, or power. The low-pass filter is designed to pass all frequency spectra below some preset frequency and to attenuate all frequency spectra above this point. The preset frequency is usually referred to as the cutoff frequency or cutoff. The high-pass filter passes all frequencies above the cutoff frequency and attenuates those below cutoff. The band-pass filter passes frequencies between two desired cutoff frequencies, and the band-stop attenuates frequencies between the cutoff points.

Consider the case where a signal, being voltage, current, or power, consists of a specified frequency spectra. Suppose the signal is corrupted by noise with a differing frequency spectra. Using classical filter theory, it is possible to retrieve the signal from the signal and noise combination by using one of the above-mentioned filters or some combination of them. The output of the filter will give a good replica of the signal. The design of classical filter circuits can be found in almost any undergraduate textbook on linear circuit theory.

However, using the same theory, if the frequency spectra of the signal and the noise overlap, then there is no way of retrieving the signal without distorting it. Wiener (16) was the first to consider the resulting problem of the kind of filter necessary to give the best estimate of the signal in 1949.

B. Optimal Filter

A compromise has to be made when the signal and noise frequency spectra do overlap since the more one attenuates the noise, the more distorted the signal becomes. What then is the optimal filter for this compromise?

There is no single right answer to this question since the problem of optimization may be approached many ways depending upon the constraints placed upon the filter and the criteria used for best performance. However, the minimum rms error criterion used as a measure of optimal performance is common to nearly all the approaches. Therefore, the best filter is the one which minimizes the rms error subject to constraints; and the most obvious constraint is that the filter be physically realizable--that the response not precede the input.[1]

---

[1]In much of the literature this is called causal.

A commonly used method to minimize the rms error when the signal and noise frequency spectra overlap is referred to as the Wiener filter. Referring to Figure 1.1, the basic problem is to find the transfer function Y(s) which will minimize the rms difference between x(t) and $s(t + \alpha)$.

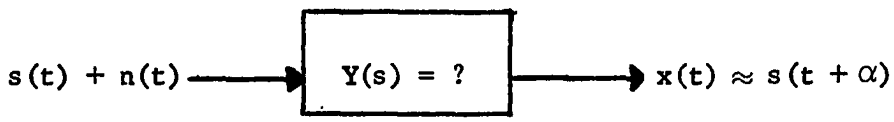


Figure 1.1. General Wiener filter problem

Note that this is the general Wiener filter problem where either delay or prediction is considered depending on the sign of α.

There are basically two different approaches in finding Y(s): the frequency domain approach used in the Bode-Shannon (3) solution, and the time domain approach found in the Wiener solution. These two methods are completely independent approaches to the same problem and both lead to the same result. However, as indicated by Brown and Nilsson (6), in certain respects the final form of the solution from Wiener's approach is easier to apply than the results of the Bode-Shannon solution.

The general procedure used to find the optimal filter is to write the error expression in terms of the weighting function, y(t), and then to use calculus of variations to find the optimum y(t), where y(t) is the inverse Laplace transform of Y(s). Note that the variational procedure will not lead directly to a solution for y(t) but only to an integral equation in y(t). This is referred to as the Wiener-Hopf integral equation which is derived in Brown and Nilsson (6) and

Levinson (12) and solved in Brown and Nilsson (6) and Davenport and Root (7). The Wiener filter problem includes basic assumptions: (1) that the filter must be physically realizable; (2) that the entire continuous past history is available for weighting.

Suppose that the input consists of discrete samples of the signal-plus-noise instead of continuous samples. The estimation of the signal must then be made on a sequence of discrete samples. This might be termed the discrete version of the Wiener filter problem. Brown (5) looks at the discrete-data filter problem from the weighting function approach, which is similar to the Wiener filter continuous data system where all past information is used to get an optimal estimate of the signal. Each measurement at every time interval is weighted. However, if there are too many measurements, the demand on the memory capabilities is very large because all past measurements must be stored. A solution to this problem describing a step-by-step recursive technique for solving the discrete data version of the least-squares smoothing and prediction problem was introduced by R. E. Kalman (11) in 1960.

It should be noted that the above arguments are not limited to the case of estimating one signal from one noisy measurement of the signal. Kalman's (11) step-by-step procedure may also be used for estimating many signals from noisy linear combinations of the signals.

C. Kalman Filter Equations

Kalman's (11) paper demonstrated a method of solving the discrete-data filter problem in the least-squares sense. With these results and the advent of the digital computer, problems could be solved that were never before realizable. The Kalman filter equations require less computer memory by updating the estimate of the signals between measurement times without requiring storage of all the past measurements.

The equations and presentation of the Kalman filter here are taken largely from unpublished notes by R. G. Brown (5) and only a very brief outline of the method is offered in this thesis. The reader is referred to these notes or to Sorenson (14) for a more complete derivation.

Most of the notation in this thesis is the same as that used by Brown (5) and is shown below:

1. A lower case letter denotes a column vector with the exception of b and ϕ.
2. An upper case letter is used to denote a matrix, as are b and ϕ which are also matrices.
3. A subscript k on any symbol is used to show that the symbol is evaluated at time t_k; e.g., $b_k = b(t_k)$ and $x_k = x(t_k)$.
4. A superscript T on any symbol denotes the transpose of that symbol.
5. A superscript -1 on any symbol denotes the inverse of that symbol.

A mathematical model of the system is assumed to be of the form

$$x_{k+1} = \phi_k x_k + g_k \tag{1.1}$$

$$y_k = M_k x_k + \delta y_k \tag{1.2}$$

where

x_k = State of the system at time t_k.

ϕ_k = Transition matrix.

g_k = Column vector of state responses due to all of the independent white-noise driving functions that occur in the interim between t_k and t_{k+1}. (Note that only white-noise driving functions are allowed in the mathematical model.)

y_k = Output vector (i.e., the "observable" or measured quantity, including noise).

δy_k = Observation noise.

M_k = Output matrix.

Furthermore, the measurement errors are assumed to be uncorrelated and unbiased timewise, i.e.,

$$E\,[\delta y_k \delta y_j^T] = \begin{cases} V_k & \text{for } k = j \\ 0 & \text{for } k \neq j \end{cases} \tag{1.3}$$

$$E\,[\delta y_k] = 0, \text{ for all } k \tag{1.4}$$

where V_k is a matrix whose terms are the variances and covariances of the respective measurement errors.

Begin with the linear estimation equation

$$\hat{x}_k = \hat{x}_k' + b_k(y_k - \hat{y}_k') \tag{1.5}$$

where y_k = the observed quantity at time t_k ,

$\hat{x}_k'$ = Best estimate of x_k based on all past measurements up through y_{k-1} (the a priori estimate of x_k) ,

$\hat{x}_k$ = Best estimate of x_k based on all measured data up through y_k (the a posteriori estimate of x_k),

b_k = "weighting" matrix or "gain" matrix.

Since the driving functions are white the a priori estimate $\hat{x}'_k$ of x_k is given by

$$\hat{x}'_k = \phi_{k-1} \hat{x}_{k-1} \tag{1.6}$$

Also, the output vector y corresponding to $\hat{x}'_k$ is given by

$$\hat{y}'_k = M_k \hat{x}'_k \tag{1.7}$$

The gain matrix b_k is now chosen such as to minimize the loss function L which is given by

$$L = E [(\hat{x}_k - x_k)^T (\hat{x}_k - x_k)] = E [e_k^T e_k] \tag{1.8}$$

where e_k is the estimation error. Note that L is a scalar and is just the sum of the variances of the estimation errors in the elements of the state vector. It can be shown that minimizing this sum is equivalent to minimizing each variance individually, so the Kalman filter minimizes the mean-square error associated with the estimation of the elements of the state vector x_k. This is justified in Sorenson (14).

Now define two error-covariance matrices as follows:

$$P_k = E [e_k e_k^T] \tag{1.9}$$

$$P_k^* = E [e'_k e'^T_k] \tag{1.10}$$

where $e'_k = (\hat{x}'_k - x_k)$ is the a priori estimation error.

The expression for the optimal gain matrix b_k is

$$b_k = P_k^* M_k^T (M_k P_k^* M_k^T + V_k)^{-1} \tag{1.11}$$

The derivation of this equation can be found in Brown (5) and Sorenson (14).

The recursive solution can be summarized as follows: a measurement y_k is taken at time t_k. Before this measurement can be used optimally, the a priori estimate $\hat{x}'_k$ and the corresponding error covariance matrix P^*_k must be known. Then the procedure is as follows:

1. Compute the optimum gain matrix b_k according to

$$b_k = P^*_k M^T_k (M_k P^*_k M^T_k + V_k)^{-1} \tag{1.12}$$

2. Revise the a priori estimate to get the a posteriori estimate according to

$$\hat{x}_k = \hat{x}'_k + b_k (y_k - \hat{y}'_k) \quad \text{where } \hat{y}'_k = M_k \hat{x}'_k \tag{1.13}$$

3. Compute the a posteriori error covariance matrix according to

$$P_k = P^*_k - b_k (M_k P^*_k M^T_k + V_k) b^T_k \tag{1.14}$$

4. Extrapolate ahead $\hat{x}_k$ and P_k to get

$$\hat{x}'_{k+1} = \phi_{k+1,k} \hat{x}_k \tag{1.15}$$

$$P^*_{k+1} = \phi_{k+1,k} P_k \phi^T_{k+1,k} + H_k \tag{1.16}$$

$$\text{where } H_k = E[g_k g^T_k] . \tag{1.17}$$

The process is now ready to be repeated for the next measurement y_{k+1}, ad infinitum. Equations 1.12 through 1.17 comprise the recursive solution for the Kalman filter. As is the case for any recursive process, initial values for P^*_k and $\hat{x}'_k$ must be specified.

It should be noted that our measurements y_k are assumed to be discrete samples in time. However, if the measurements are continuous rather than discrete, the Kalman filter equations can be extended to the time-continuous case by a limiting argument [See Sorenson (14)]. In much of the literature, the time-continuous filter is referred to as the Kalman-Bucy filter. Unlike the discrete data problem the solution of the time-continuous problem yields a set of matrix differential equations as follows:

1. The gain equation is

$$b^*(t) = P(t)M^T(t)V^{-1}(t) \tag{1.18}$$

2. The state differential equation is

$$\frac{d\hat{x}}{dt} = \phi(t)\hat{x} + b^*(t)[y(t) - M(t)\hat{x}] \tag{1.19}$$

3. The error covariance matrix differential equation is

$$\frac{dP}{dt} = \phi(t)P + P\phi^T(t) - PM^T(t)V^{-1}(t)M(t)P + G(t)H(t)G^T(t) \tag{1.20}$$

The derivation and solution of Equations 1.18 through 1.20 can be found in Sorenson (14).

Note that for all the above estimation or filter schemes the statistical behavior of the signal is known. Then the question arises, what is the best or optimal filter if nothing is known about the statistical properties of the signal? The answer is that the optimization scheme used must not in any way depend upon the nature of the signal. If there is only one measurement of the signal plus-noise, the optimal estimate would just be the measurement, which is a trivial solution. However, if two independent noisy measurements of the signal are available, a better estimate of the signal can be obtained through the use of complementary filtering as discussed below.

D. The Complementary Filter

The complementary filter was motivated from the case where nothing was known about the signal. For example, consider a situation where there are two independent noisy measurements of the same quantity; and it is wished to obtain the optimal estimate of the signal knowing only the spectral density functions of the noise. With reference to Figure 1.2, the problem is to choose $Y_1(s)$ and $Y_2(s)$ so as to minimize the mean square error and not to distort the signal.

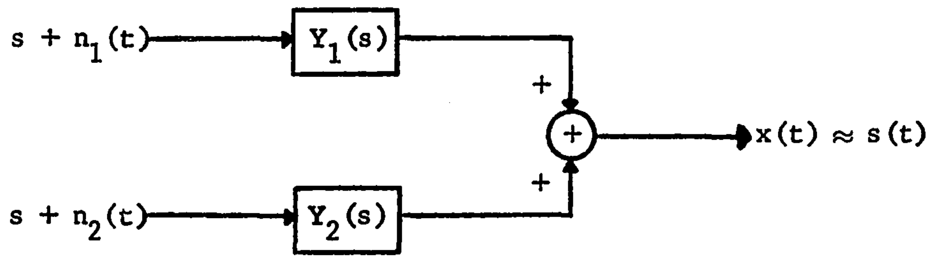


Figure 1.2. Linear combination of two independent noisy signals.

The expression for the output in transformed form is

$$X = Y_1(S + N_1) + Y_2(S + N_2) \tag{1.21}$$

If the following constraint between Y_1 and Y_2 is used

$$Y_2 = 1 - Y_1 \tag{1.22}$$

then Equation 1.21 becomes

$$X = S + [N_1Y_1 + N_2(1 - Y_1)] \tag{1.23}$$

Note that the term within the brackets of Equation 1.23 is the error term and the choice of Y_1 will not affect the signal portion of the output. The error term is then made as small as possible, using the minimum mean square error criterion, by the appropriate choice of Y_1. As shown by Brown and Nilsson (6) the solution for Y_1 is obtained in the same manner as in the Wiener filter problem, except in this case $n_1(t)$ and $n_2(t)$ play the roles of n(t) and s(t) respectively.

Note that this type of filtering might be called complementary filtering because each of the two transfer functions is the complement of the other. With reference to Equation 1.23, in the complete absence of noise, the output is exactly equal to the signal. Hence, signal distortion is not necessary to smooth the noise, as was the case in the Wiener filter problem. For this reason this method of filtering is also referred to as distortionless filtering.

The complementary or distortionless filter is not restricted to just the two input problem. Consider an m input problem as shown in Figure 1.3.

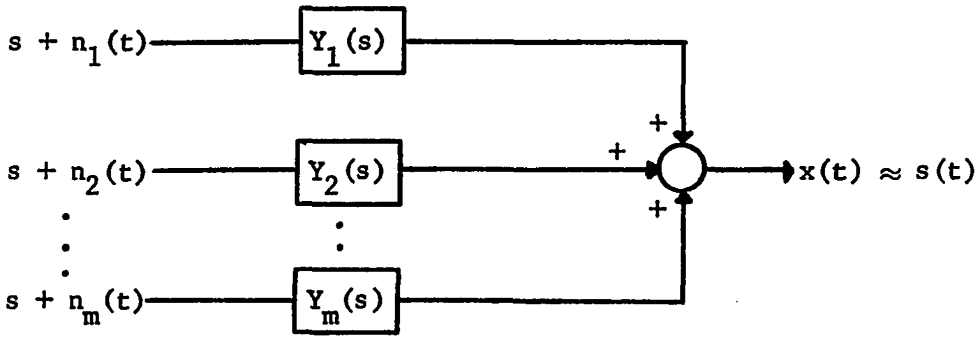


Figure 1.3. Linear combination of m sources of information.

In transformed form x(t) is

$$X = (S + N_1)Y_1 + (S + N_2)Y_2 + \cdots\cdots\cdots (S + N_m)Y_m \tag{1.24}$$

Now if $Y_m = 1 - Y_1 - Y_2 - \cdots\cdots\cdot Y_{m-1}$ then Equation 1.24 becomes

$$X = S + N_1Y_1 + N_2Y_2 + N_3(1 - Y_1 - Y_2 - \cdots\cdots\cdot Y_{m-1}) \quad (1.25)$$

The problem is to determine Y_1 through Y_{m-1} such that the mean square value of the error is a minimum. This is similar to the Wiener filter problem again, with the exception that there are m-1 degrees of freedom in the optimization process. An example of a two-dimensional problem can be found in Brown and Nilsson (6).

The purpose of this thesis is to explore in greater detail the Kalman filter equation with the complementary constraint. A large majority of the aided inertial navigation schemes proposed to date use the complementary constraint in one form or another in the estimate of position and velocity. Immediately, because of the use of this constraint, the question of the extent of knowledge of the behavior of the statistical signal arises. Some information about this is known; however, as shown by Brock and Schmidt (4), usually such statistics are too complicated or are too uncertain to be described analytically with confidence. Also, other factors are involved in the filter problem which are impossible, or nearly impossible, to describe mathematically. Trade off between performance and computer size is one. If one assumes some statistics for the signal which are not absolutely correct, it is possible to have very large errors. However, using the complementary filter and the least squares criterion, in essence an optimal estimate is obtained for the worst possible case. That is, the system is a min-max estimator.

A number of terrestrial navigation schemes include an inertial navigation unit and other aiding sources which give the optimal estimates of position and velocity. Generally, the signal variables are eliminated from the measurement equations and a new set of measurement equations are used that consist only of the noise variables. The optimal estimates of the noises are then determined, which in turn are subtracted from the original measurement equations to give the estimates of the signals. Specific examples can be found in Brock and Schmidt (4) and Huddle (10). In view of the reasons described above for the use of the complementary filter, a review of some of the methods in achieving the optimal complementary filter are in order.

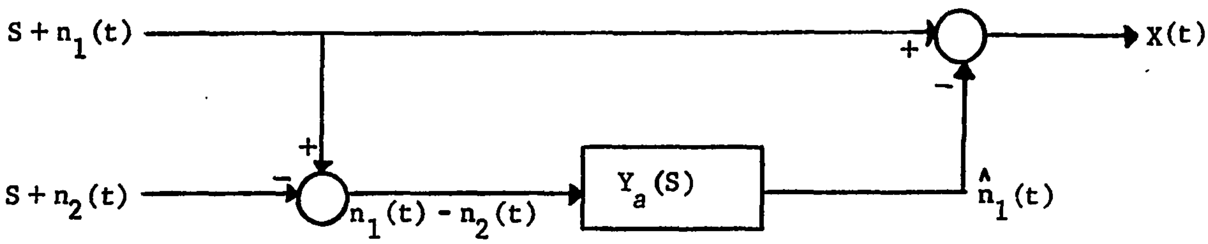
E. Review of Multiple-Input Complementary Filter

The purpose of this section is to give a review of some of the work that has been done on the multiple-input complementary filter. Both the continuous and discrete time systems will be studied via Wiener and Kalman filters.

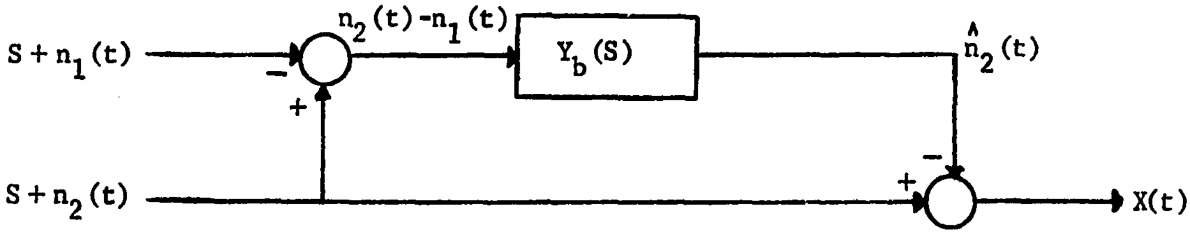
Benning (2) investigated the case of m inputs which consisted of known linear combinations of r signals plus an additive random noise with known spectral density functions when nothing was known about the signals, hence a multiple-input complementary filter. His method was an intuitive scheme for estimating the signals, which can best be demonstrated by a simple two-dimensional problem. The intuitive scheme is shown in Figure 1.4.

In Figure 1.4(a) the output is

$$X = (S + N_1) - (N_1 - N_2)Y_a = S + N_1(1 - Y_a) + N_2Y_a \qquad (1.26)$$



(a)



(b)

Figure 1.4. Two intuitive systems for estimating s(t).

Now if Y_a is equal to $Y_2 = 1 - Y_1$ from Equation 1.22, then Equation 1.26 becomes

$$X = S + [N_1Y_1 + N_2(1 - Y_1)] \tag{1.27}$$

which is identical to Equation 1.23. Also in Figure 1.4(b) if Y_b is equal to Y_1, then the expression for X is identical to 1.27.

Benning then extended the intuitive approach to the case where there were m measurements of n signals, as shown in Figure 1.5.

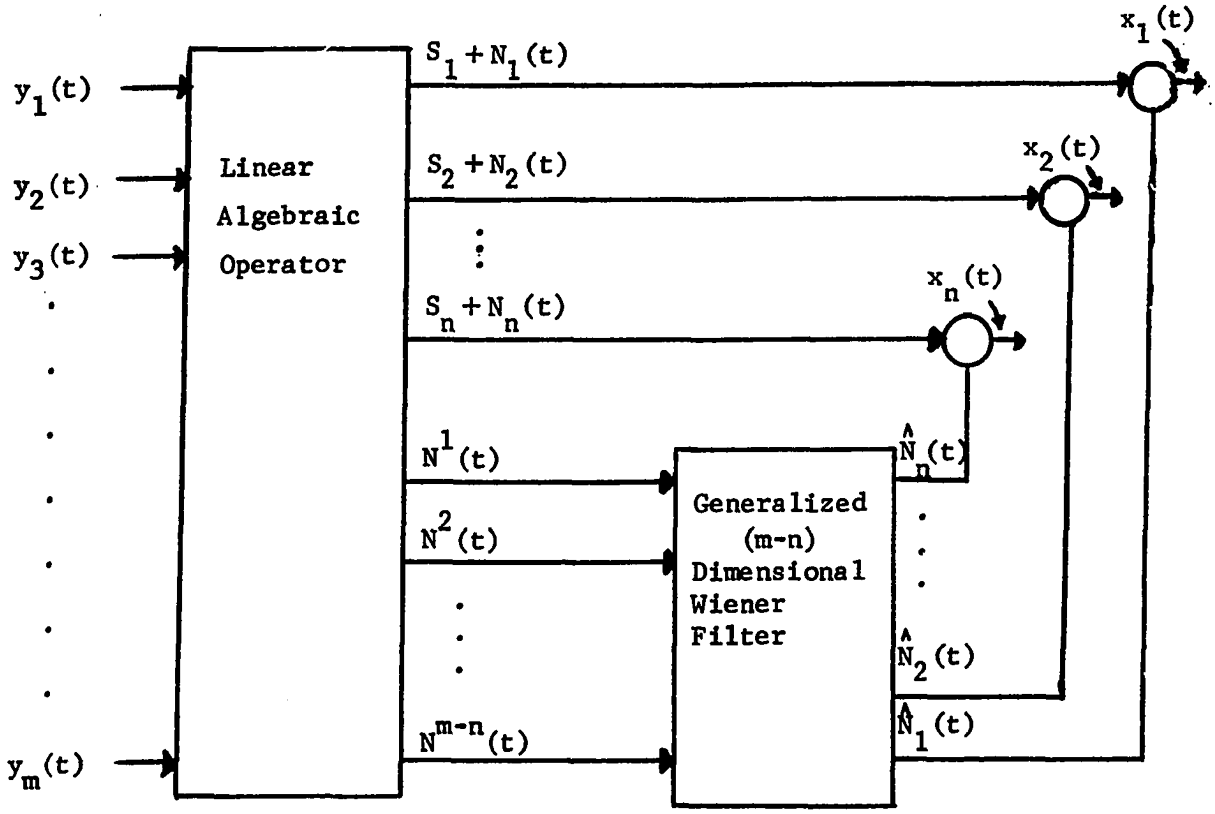


Figure 1.5. Block diagram of multiple-input intuitive complementary filter.

With reference to Figure 1.5, the symbols are:

$y_i(t)$ = Linear combination of r signals corrupted by additive noise.

$S_i + N_i(t)$ = One of the n signals corrupted by a linear combination of the m noises for the inputs.

$N^i(t)$ = Linear combination of the m noises from the inputs. [Note $N_i(t) \neq N^i(t)$].

The above system is satisfactory for continuous-time systems. However, when the inputs are discrete samples, the Kalman filter may be used. The (m - n) dimensional Wiener filter is replaced with a Kalman filter. Benning works out examples for both the time-continuous Wiener filter and the discrete-data Kalman filter.

Benning uses a linear algebraic operator to preprocess the measurements. This has the disadvantage that if one of the measurements is not available, then there has to be a new algebraic operation and a different Wiener filter configuration. The same argument can be used for the discrete-data filter. Thus, if one allows for single failures, there have to be m backup systems to account for all the possible losses of measurements. Also, if a fail-safe system is considered, there have to be backup systems for all combinations of two, three, etc. failures. The total number of backup systems needed for a fail-safe complementary filter, denoted by B, is

$$B = \sum_{i=1}^{m-n-1} \binom{m}{i} \tag{1.28a}$$

The summation only needs to be taken to (m - n - 1), since any number greater than this would not yield a complementary filter. The quantity $\binom{m}{i}$ is the number of possible combinations of i elements out of m total elements. In the Wiener filter a large amount of wiring and interconnections would be required if m were very large. In the Kalman filter an additional algorithm and a large amount of memory would have to be used to accommodate all possible failures. Conservative numbers for m and n might be 6 and 3 respectively. This might be the case where the signals were the three positions with three redundant measurements.

Then

$$B = \binom{6}{1} + \binom{6}{2} = 6 + 15 = 21 \tag{1.28b}$$

which is a fairly large number of backup systems. Finding some way to avoid the algebraic operator would help to alleviate this problem. If one were to operate on the measurements directly and then produce an optimal estimate of the signal with the complementary constraint, then this might be a workable solution to the problem of intermittent loss of measurements.

Bakker (1) investigated this method by deriving the Kalman filter equations with the complementary constraint. That is, the inputs to the filter were the m measurements and the outputs of the filter were the optimal estimates of the signal in the least squares sense, and at the same time the estimates satisfied the complementary or distortionless constraints.

A fairly detailed review of Bakker's work will be presented here since his results will be used in the next chapter. However, before proceeding, some partitioning of matrices and column vectors will be noted. Also, the time subscripts k will be omitted in the following equations in order to avoid confusion with the partitioned subscripts. All of the following equations are at time t_k unless otherwise noted.

The state variables can be partitioned as follows:

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \\ \hdashline x_{n+1} \\ \vdots \\ x_p \end{bmatrix} = \begin{bmatrix} x_S \\ \hdashline x_N \end{bmatrix} \tag{1.29}$$

where X_S in the n-dimensional signal variable and X_N is the (p-n) dimensional noise variable. The signal variables are those variables that are not to be distorted.

The state transition matrix is partitioned in the following manner:

$$\phi(t) = \left[\begin{array}{ccc|ccc} \phi_{11} & \cdots & \phi_{1n} & \phi_{1,n+1} & \cdots & \phi_{1p} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \phi_{n1} & \cdots & \phi_{nn} & \phi_{n,n+1} & \cdots & \phi_{np} \\ \hline \phi_{n+1,1} & \cdots & \phi_{n+1,n} & \phi_{n+1,n+1} & \cdots & \phi_{n+1,p} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \phi_{p1} & \cdots & \phi_{p,n} & \phi_{p,n+1} & \cdots & \phi_{pp} \end{array}\right] \triangleq \left[\begin{array}{c|c} \phi_S & \phi_3 \\ \hline \phi_4 & \phi_N \end{array}\right] \quad (1.30)$$

In addition

$$\phi_1 \triangleq [\phi_S \quad \phi_3] \quad (1.31)$$

and

$$\phi_2 \triangleq [\phi_4 \quad \phi_N] \quad (1.32)$$

Bakker assumed that $\phi_4 = 0$ for all k, which means that the value of the noise vector at time t_k must not depend on the value of the signal vector at time t_{k-1}.

The measurement matrix can be partitioned as:

$$M = \left[\begin{array}{ccc|ccc} m_{11} & \cdots & m_{1n} & m_{1,n+1} & \cdots & m_{1p} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ m_{n1} & \cdots & m_{nn} & m_{n,n+1} & \cdots & m_{np} \end{array}\right] \triangleq [M_S \quad M_N] \quad (1.33)$$

Next, the p^{th} order identity matrix is partitioned.

$$I^{(p)} = \left[\begin{array}{c|c} I^{(n)} & 0^{(n,p-n)} \\ \hline 0^{(p-n,n)} & I^{(p-n)} \end{array}\right] \triangleq [I_S \quad I_N] \tag{1.34}$$

The a priori estimate $\hat{x}'$ can be partitioned the same way as Equation 1.29. Then using Equations 1.29 through 1.34 the estimation Equation 1.5 can be written as:

$$\hat{x} = [I_S - bM_S]\,\hat{x}_S' + [I_N - bM_N]\,\hat{x}_N' + b_y \tag{1.35}$$

If the "noise" vector and the measurement noises happen to be zero for all k, then the filter must yield a perfect estimate of the signal. This is known as the complementary constraint. That is

$$\hat{x}_S = x_S \tag{1.36}$$

Bakker (1) shows that this constraint can be satisfied by requiring that the estimate of the state vector be independent of the a priori estimate of the signal vector. It can be seen from Equation 1.35 that this condition is satisfied if

$$[I_S - bM_S] = 0 \tag{1.37}$$

and hence this is called the distortionless or complementary constraint.

If the gain matrix b is partitioned between rows n and (n + 1) as

$$b = \left[\begin{array}{c} b_S \\ \hline b_N \end{array}\right] \tag{1.38}$$

then Equation 1.37 can be rewritten as the following two equations:

$$b_S M_S = I^{(n)} \tag{1.39}$$

$$b_N M_S = 0^{(p-n,n)} \tag{1.40}$$

The a priori and a posteriori covariance matrices can be partitioned similarly to the transition matrix

$$P = \left[\begin{array}{c|c} P_S & P_3 \\ \hline P_3^T & P_N \end{array}\right] \tag{1.41}$$

$$P^* = \left[\begin{array}{c|c} P_S^* & P_3^* \\ \hline P_3^{*T} & P_N^* \end{array}\right] \tag{1.42}$$

Bakker (1) determined the optimal gain matrix b^* which minimized the mean square error and at the same time satisfied the constaint Equations 1.38 through 1.40. He used the method of Lagrange multipliers and derived the following Equation for b^*:

$$b^* = \{ P^* M^T + [I_S - P^* M^T (MP^* M^T + V)^{-1} M_S][M_S^T (MP^* M^T + V)^{-1} M_S^T \}$$

$$(MP^* M^T + V)^{-1} \tag{1.43}$$

Using the partitioned form of P^* and b^* Equation 1.43 can be written as the following two equations:

$$b_S^* = P_3^* M_N^T (MP^* M^T + V)^{-1} \{ I - M_S [M_S^T (MP^* M^T + V)^{-1} M_S]^{-1} M_S^T (MP^* M^T + V)^{-1} \}$$

$$+ [M_S^T (MP^* M^T + V)^{-1} M_S]^{-1} M_S^T (MP^* M^T + V)^{-1} \tag{1.44}$$

$$b_N^* = P_N^* M_N^T (MP^* M^T + V)^{-1} \{ I - M_S [M_S^T (MP^* M^T + V)^{-1}$$

$$M_S^T (MP^* M^T + V)^{-1} \} \tag{1.45}$$

An alternate form is:

$$b_S^* = [M_S^T(M_N P_2^* M^T + V)^{-1} M_S]^{-1} M_S^T(M_N P_2^* M^T + V)^{-1} \tag{1.46}$$

$$b_N^* = P_N^* M_N^T (M_N P_2^* M^T + V)^{-1} [I - M_S b_S^*] \tag{1.47}$$

Equations 1.44 and 1.45 are equivalent to Equations 1.46 and 1.47 even though there is little resemblance. However, upon implementing this filter the latter two equations would probably be used instead of the previous two because they are generally simpler.

Because of the constraints put on the Kalman filter the a posteriori covariance matrix will be of a different form than the usual Kalman filter equation:

$$P = [I_N - b^* M_N] P_N^* [I_N - b^* M_N]^T + b^* V b^{*T} \tag{1.48}$$

The computations for Bakker's distortionless filter are done in the same order that was suggested earlier in Section C for the Kalman filter, with the exception that Equations 1.46 and 1.47 are used instead of Equation 1.12 and Equation 1.48 is used in place of Equation 1.14.

F. Objectives

Upon examining Bakker's (1) equations for the complementary Kalman filter, one finds the computation and the amount of memory needed are generally greater than in Benning's approach since Bakker's approach has a larger number of states (both signal and noise) than Benning's (noise states only). Also, the gain matrices are much more complicated in Bakker's equations. Even though this is true, Bakker's approach might still be better to use if a fail-safe system is desired. Thus, it should be worthwhile to further investigate the Kalman filter equations with the complementary constraint in order to achieve a more efficient system in both computation time and memory requirements.

Bakker (1) suggested a method that would involve less computation time than the method outlined above and demonstrated it with a simple example. The idea is intuitively sound but was not proven. Bakker (1) suggested that the complementary constraint can be satisfied by requiring that the estimate of the state vector be independent of the a priori estimate of the signal vector, i.e., $\hat{x}$ be independent of $\hat{x}_S'$. The elements along the major diagonal of P are the variances of the estimation errors. Similarly, the elements along the major diagonal of P^* are the variances of the errors in the a priori estimates. Intuitively, it would seem that if one of these elements along the major diagonal were very large then the a priori estimate of that state would receive very little weight in determining the new estimate $\hat{x}$. In the extreme case, if the variance was set to ∞, it should receive no weight at all. The intuitive approach involves setting the variances of the a priori signal vectors to infinity and hence not entering into the determination of the new estimate $\hat{x}$, which is precisely the complementary constraint. It should be noted that the optimal gain equation is simply the ordinary Kalman filter gain equation developed in Section C. The next chapter rests on the above discussion and shows that the Kalman filter with the complementary constraint can be obtained by simply taking the ordinary Kalman filter equations and setting the variances of the a priori signal vector equal to infinity. After the proof, an algorithm will be developed that will be simpler than the normal Kalman filter algorithm because advantage can be taken of many zero terms. Then a comparison will be made between this approach and the method described by Benning (2).

In Chapter IV the matrix differential equations for the Kalman-Bucy continuous-time filter with the complementary constraint will be developed, using a limiting argument similar to that employed by Sorenson (14).

The last objective will be to apply the complementary Kalman filter equations to two integrated navigation systems problems.

II. DIRECT COMPLEMENTARY KALMAN FILTER

A. Development of the Direct Complementary Kalman Filter

It is the purpose of this section to show that the normal Kalman filter equations can be altered to satisfy the complementary constraint. As mentioned in the previous chapter, intuitively, if one lets the a priori variances for the signal vectors approach infinity in the normal Kalman filter equations an optimal complementary Kalman filter is obtained.

For brevity, the optimal complementary Kalman filter to be derived here will be referred to as the "direct" filter and Benning's (2) filter equations will be referred to as the "indirect" filter. Again, the time subscripts k will be omitted to avoid confusion with the partitioned subscripts. As before, the equations are assumed to be at time t_k unless otherwise specified.

The normal optimal Kalman gain and a posteriori covariance equations are

$$b^* = P^*M^T(MP^*M^T + V)^{-1} \tag{2.1}$$

$$P = P^* - b^*(MP^*M^T + V)b^{*T} \tag{2.2}$$

Upon substituting Equation 2.1 into Equation 2.2 it becomes

$$P = P^* - P^*M^T(MP^*M^T + V)^{-1}MP^* \tag{2.3}$$

If V and P^* are positive definite matrices, then the gain equation and a posteriori covariance matrix can be written in an alternate form as described by Sorenson (14).

$$b^* = PMV^{-1} \tag{2.4}$$

$$P^{-1} = [P^{*-1} + M^T V^{-1} M] \tag{2.5}$$

Again let P^* be partitioned as before

$$P^* \triangleq \left[\begin{array}{c|c} P_S^* & P_3^* \\ \hline P_3^{*T} & P_N^* \end{array}\right] \tag{2.6}$$

where P_S^* = the covariance matrix of the a priori signal variables,

P_N^* = the covariance matrix of the a priori noise variables,

P_3^* = the covariances among the a priori signal and noise variables.

To apply the intuitive idea to the Kalman filter equation, let the variances of the a priori signal variables approach infinity, as described by the following equation

$$P_S^* = \lim_{a \to \infty} \begin{bmatrix} a & 0 & 0 & \cdots & 0 \\ 0 & a & 0 & . & 0 \\ . & & . & & . \\ 0 & & \cdots & & a \end{bmatrix} \tag{2.7}$$

Also P_3^* will be set equal to zero.

Then

$$P_S^* = \lim_{a \to \infty} \left[\begin{array}{cccc|c} a & 0 & \cdots & 0 & \\ 0 & a & \cdots & 0 & \bigcirc \\ \vdots & & & & \\ 0 & . & \cdots & a & \\ \hline & \bigcirc & & & P_N^* \end{array}\right] \tag{2.8}$$

Equation 2.5 requires that the inverse of Equation 2.8 be found. It can be shown (See Appendix A), upon taking the limit as $a \to \infty$, P^{*-1} becomes

$$P^{*-1} = \left[\begin{array}{c|c} O & O \\ \hline O & P_N^{*-1} \end{array}\right] \tag{2.9}$$

Then Equation 2.5 becomes

$$P^{-1} = \left[\begin{array}{c|c} O & O \\ \hline O & P_N^{*-1} \end{array}\right] + M^T V^{-1} M \tag{2.10}$$

Upon using the partitioned form of M, and after the indicated matrix multiplication is performed, Equation 2.10 becomes

$$P^{-1} = \left[\begin{array}{c|c} M_S^T V^{-1} M_S & M_S^T V^{-1} M_N \\ \hline M_N^T V^{-1} M_S & M_N^T V^{-1} M_N + P_N^{*-1} \end{array}\right] \tag{2.11}$$

In partitioned form

$$P = \left[\begin{array}{c|c} P_S & P_3 \\ \hline P_3^T & P_N \end{array}\right] \tag{2.12}$$

and denote P^{-1} as

$$P^{-1} = \left[\begin{array}{c|c} A & B \\ \hline B^T & C \end{array}\right] \tag{2.13}$$

From properties of matrices, the product of a matrix times its inverse will be an identity matrix. Thus

$$P^{-1}P = \left[\begin{array}{c|c} A & B \\ \hline B^T & C \end{array}\right] \left[\begin{array}{c|c} P_S & P_3 \\ \hline P_3^T & P_N \end{array}\right] = I \qquad (2.14)$$

After the indicated multiplication, Equation 2.14 can be written as the following four equations.

$$AP_S + BP_3^T = I \qquad (2.15)$$

$$AP_3 + BP_N = 0 \qquad (2.16)$$

$$B^TP_S + CP_3^T = 0 \qquad (2.17)$$

$$B^TP_3 + CP_N = I \qquad (2.18)$$

If the complementary constraint is to be satisfied, the matrix M_S is of rank r, where r represents the number of signal variables. Then the quantity $M_S^TV^{-1}M_S$ is an r x r matrix with rank r and is invertible. Also assuming that $(M_N^TV^{-1}M_N + P_N^{*-1})$ is invertible, the matrices A and C have an inverse. Using Equations 2.15 through 2.18 the following equations for P_S, P_N, and P_3 are obtained.

$$P_S = [A - BC^{-1}B^T]^{-1} \qquad (2.19a)$$

$$P_N = C^{-1} + C^{-1}B^TP_SBC^{-1} \qquad (2.19b)$$

$$P_3 = -P_SBC^{-1} \qquad (2.19c)$$

Substituting the identities for A, B, and C, and after much matrix manipulation the following equations are obtained.

$$P_S = [M_S^T(M_NP_N^*M_N^T + V)^{-1} M_S]^{-1} \qquad (2.20a)$$

$$P_N = P_N^* - P_N^* M_N^T [(M_N P_N^* M_N^T + V)^{-1} - (M_N P_N^* M_N^T + V)^{-1}$$

$$M_S P_S M_S^T (M_N P_N^* M_N^T + V)^{-1}] M_N P_N \quad (2.20b)$$

$$P_3 = -[M_S^T (M_N P_N^* M_N^T + V)^{-1} M_S]^{-1} M_S^T (M_N P_N^* M_N^T + V)^{-1} M_N P_N \quad (2.20c)$$

The algebra between Equations 2.19 and 2.20 is not shown here because of its great length (See Appendix B).

It will be shown that the direct filter described above is the optimal complementary filter. This will be proven by showing that Equations 2.20, which are the a posteriori covariance terms, are identical to Bakker's covariance terms. Thus, if both methods have identical covariance matrices, this means that the minimum mean square errors are identical. Since Bakker's (1) filter is optimal then the direct filter equation must be optimal, provided that the complementary constraint is satisfied in the direct filter.

Bakker's (1) a posteriori covariance matrix is given by

$$P = (I_N - b^* M_N) P_N^* (I_N - b^* M_N)^T + b^* V b^{*T} \quad (2.21)$$

Using the partitioned form of b^*, Equation 2.21 can be rewritten as

$$P = \begin{bmatrix} -b_S^* M_N \\ \hline I - b_N^* M_N \end{bmatrix} P_N^* [-M_N^T b_S^{*T} \;\vdots\; (I - M_N^T b_N^{*T}] + \left[\begin{array}{c:c} b_S^* V b_S^{*T} & b_S^* V b_N^{*T} \\ \hdashline b_N^* V b_S^{*T} & b_N^* V b_N^{*T} \end{array}\right] \quad (2.22)$$

After multiplying and collecting terms, P becomes

$$P = \left[\begin{array}{c|c} b_S^*(M_N P_N^* M_N^T + V) b_S^{*T} & -b_S^* M_N P_N^* + b_S^*(M_N P_N^* M_N^T + V) b_N^{*T} \\ \hline -P_N^* M_N^T b_S^{*T} + b_N^*(M_N P_N^* M_N^T + V) b_S^{*T} & P_N^* - b_N^* M_N P_N^* - P_N^* M_N^T b_N^{*T} + b_N^*(M_N P_N^* M_N^T + V) b_N^{*T} \end{array}\right] = \left[\begin{array}{c|c} P_S & P_3 \\ \hline P_S^* & P_N \end{array}\right] \quad (2.23)$$

Upon inserting the expressions for b_S^* and b_N^* (Equations 1.45 and 1.46) into Equation 2.23, and after lengthy matrix maneuvers, the following expressions are obtained

$$P_S = [M_S^T(M_N P_N^* M_N^T + V)^{-1} M_S]^{-1} \quad (2.24a)$$

$$P_N = P_N^* - P_N^* M_N [(M_N P_N^* M_N^T + V)^{-1} - (M_N P_N^* M_N^* + V)^{-1} (M_N P_N^* M_N + V)^{-1}] M_N P_N^* \quad (2.24b)$$

$$P_3 = -P_S M_S^T (M_N P_N^* M_N + V)^{-1} M_N P_N^* \quad (2.24c)$$

As can be seen, upon comparison, Equations 2.20 and Equations 2.24, respectively, are identical. The algebra omitted between Equations 2.23 and 2.24 is in Appendix C.

To show that the direct filter does satisfy the complementary constraint, the following two identities must be satisfied

$$b_S M_S = I \quad (2.25)$$

$$b_N M_S = 0 \quad (2.26)$$

Using Equation 2.4 and the partitioned forms of P, M, and b, we find that

$$b = \begin{bmatrix} b_S \\ \text{----} \\ b_N \end{bmatrix} = PM^TV^{-1} = \begin{bmatrix} P_S & \vdots & P_3 \\ \text{---} & + & \text{---} \\ P_3^T & \vdots & P_N \end{bmatrix} \begin{bmatrix} M_S^T \\ \text{---} \\ M_N^T \end{bmatrix} V^{-1} \tag{2.27}$$

After multiplying, Equation 2.27 becomes

$$\begin{bmatrix} b_S \\ \text{----} \\ b_N \end{bmatrix} = \begin{bmatrix} P_SM_S^TV^{-1} + P_3M_N^TV^{-1} \\ \text{-----------------} \\ P_3^TM_S^TV^{-1} + P_NM_N^TV^{-1} \end{bmatrix} \tag{2.28}$$

First look at b_SM_S given by

$$b_SM_S = P_SM_S^TV^{-1}M_S + P_3M_N^TV^{-1}M_S \tag{2.29}$$

Substituting for P_3, Equation 2.29 becomes

$$b_SM_S = P_SM_S^TV^{-1}M_S - P_SM_S^T(M_NP_N^*M_N^T + V)^{-1}M_NP_N^*M_N^TV^{-1}M_S \tag{2.30}$$

Let

$$W = (M_NP_N^*M_N^T + V) \tag{2.31}$$

Using this identity Equation 2.30 can be written as

$$b_SM_S = P_SM_S^TV^{-1}M_S - P_SM_S^TW^{-1}(W-V)V^{-1}M_S \tag{2.32}$$

After multiplying and canceling terms,

$$b_SM_S = P_SM_S^TW^{-1}M_S \tag{2.32}$$

Note that

$$M_S^TW^{-1}M_S = M_S^T(M_NP_N^*M_N^* + V)^{-1}M_S = P_S^{-1} \tag{2.33}$$

Then Equation 2.32 is

$$b_SM_S = P_SP_S^{-1} = I \tag{2.34}$$

Secondly, $b_N M_S$ is given by

$$b_N M_S = P_3^T M_S^T V^{-1} M_S + P_N M_N^T V^{-1} M_S \tag{2.35}$$

Substituting for P_3^T and P_N and using Equation 2.31, Equation 2.35 can be written as

$$b_N M_S = - P_N^* M_N^T (W^{-1}) M_S P_S M_S^T V^{-1} M_S + P_N^* M_N^T V^{-1} M_S$$

$$- P_N^* M_N^T [W^{-1} - W^{-1} M_S P_S M_S^T W^{-1}] M_N P_N^* M_N^T V^{-1} M_S \tag{2.36}$$

Replacing $M_N P_N^* M_N^T$ with (W-V) and further multiplication, Equation 2.36 becomes

$$b_N M_S = - P_N^* M_N^T W^{-1} M_S P_S M_S^T V^{-1} M_S + P_N^* M_N^T V^{-1} M_S - P_N^* M_N^T V^{-1} M_S + P_N^* M_N^T W^{-1} M_S$$

$$+ P_N^* M_N^T W^{-1} M_S P_S M_S^T V^{-1} M_S - P_N^* M_N^T W^{-1} M_S P_S P_S^T W^{-1} M_S \tag{2.37}$$

Canceling terms and factoring $P_N^* M_N^T$ yields

$$b_N M_S = P_N^* M_N^T [W^{-1} M_S - W^{-1} M_S P_S M_S^T W^{-1} M_S] \tag{2.38}$$

Using Equation 2.33,

$$b_N M_S = P_N^* M_N^T [W^{-1} M_S - W^{-1} M_S P_S P_S^{-1}] \equiv 0 \tag{2.39}$$

Thus, the direct filter satisfies the complementary constraint and has the identical a posteriori covariance matrix as Bakker's (1); therefore, the direct filter must be the optimal complementary filter.

An alternate gain equation for the direct filter can be found by substitution of Equation 2.24 into Equation 2.28 and using Equation 2.31. Then b_S is

$$b_S = P_S M_S^T V^{-1} - P_S M_S^T W^{-1} M_N P_N^* M_N^T V^{-1} = P_S M_S^T W^{-1} \qquad (2.40)$$

Substitution of P_S and W^{-1} gives

$$b_S = [M_S^T (M_N P_N^* M_N^T + V)^{-1} M_S]^{-1} M_S^T (M_N P_N^* M_N^T + V)^{-1} \qquad (2.41)$$

Similarly,

$$\begin{aligned} b_N &= P_N M_N^T V^{-1} - P_N^* M_N^T W^{-1} M_S P_S M_S^T V^{-1} \\ &= P_N^* M_N^T V^{-1} - P_N^* M_N^T [W^{-1} - W^{-1} M_S P_S M_S^T W^{-1}] (W-V) V^{-1} \\ &\quad - P_N^* M_N^T W^{-1} M_S M_S^T V^{-1} \end{aligned} \qquad (2.42)$$

Multiplying terms results in

$$\begin{aligned} b_N &= P_N^* M_N^T V^{-1} - P_N^* M_N^T V^{-1} + P_N^* M_N^T V^{-1} + P_N^* M_N^T W^{-1} M_S P_S M_S^T V^{-1} \\ &\quad - P_N^* M_N^T W^{-1} M_S P_S M_S^T W^{-1} - P_N^* M_N^T W^{-1} M_S P_S M_S^T V^{-1} \\ &= P_N^* M_N^T W^{-1} [I - M_S P_S M_S^T W^{-1}] \end{aligned} \qquad (2.43)$$

Using Equation 2.40 and substituting for W^{-1}

$$b_N = P_N^* M_N^T (M_N P_N^* M_N^T + V)^{-1} [I - M_S b_S] \qquad (2.44)$$

At this time, the difference between Bakker's (1) equations and those of the direct approach will be noted. Assume that a particular estimation problem requires the use of the complementary constraint. First, a model of the system is found. The state equation, state transistion matrix, measurement matrix, etc., will be identical for

Bakker's (1) and the direct methods. The only difference will be the gain equation and the a priroi covariance matrix. The a priori matrix in Bakker's method is the same as the normal Kalman filter equations. The direct method requires an extra step, i.e., to set diagonal terms of P_S to infinity and P_3 to zero. However, this step will require very little computation time or computer memory. The other difference between the two methods are in the gain matrices. Upon comparing Bakker's Equations 1.45 and 1.46 with the normal Kalman filter gain matrix (Equation 1.12), note that the latter equation has the advantage. That is, one less inverse is required. The savings in the inversion is reason enough to prefer the normal Kalman equations. As quoted by Sorenson (14),

> "The inversion on a digital computer of a matrix of large dimension is undesirable for several reasons--the amount of storage cells that must be used, the time that is comsumed in obtaining the inverse, and the accuracy of the end result. Thus, if the inversion can be circumvented, it is advisable to do so."

Therefore, the direct filter would probably be preferred over the method of Bakker (1).

One must realize that the direct filter can not be implemented exactly as described above because the a priori signal variance terms can not be set equal to infinity. However, as Bakker pointed out these terms wouldn't have to be set equal to infinity, but would only have to be on the order of 10 to 100 times larger than the largest element in the a priori P^* matrix. However, as will be shown in the next section the terms that are to be set equal to infinity can be circumvented to yield an exact solution.

B. An Algorithm for Sequential Processing in the Direct Kalman Filter

The purpose of this section is to develop an algorithm for the direct complementary filter, which will be derived so as to circumvent the infinite terms in the a priori covariance matrix. The use of the normal Kalman filter gain equation produces a savings in computation time and computer memory over Bakker's (1) equations. Also, if at each sampling time t_k, m statistically independent sources provide measurement data, then each measurement can be processed one at a time. This procedure is designated by the term sequential processing. The proof of the sequential processing procedure can be found in Sorenson (14).

It is not apparent to the author if sequential processing can be used in Bakker's (1) equations, since Sorenson's (14) proof dealt specifically with the normal Kalman filter equations. However, it might be worth investigation by some interested person.

The development of the algorithm with sequential processing proceeds in the following manner. Assume there are r state variables which are designated to be the signal variables. Furthermore, assume there are m independent measurements consisting of linear combinations of the r signal variables, each corrupted by additive noise and $m > r$. In the state equation there are r signal variables and n noise variables. The number of noise variables depends upon how the noises are modeled.

Assume at time t_k one has a priori estimates of the states $\hat{x}'_k$ and the associated covariance matrix P^*_k. The direct complementary filter requires that, before processing the measurements at time t_k, the signal variances of P_k^* be set to infinity as shown by

$$P^* = \left[\begin{array}{cccc|c} a & 0 & \cdots & 0 & \\ 0 & a & \cdots & 0 & \\ \cdot & \cdot & & \cdot & \bigcirc \\ 0 & 0 & \cdots & a & \\ \hline & & & & \\ & \bigcirc & & & P_N^* \\ & & & & \end{array}\right] \tag{2.45}$$

where a approaches infinity.

In order to avoid confusion, the time subscript k will be omitted and all matrices will be valid at time t_k unless otherwise specified. Since the adgorithm being developed uses sequential processing, the matrices will be subscripted such as to indicate which measurement is being processed. For example, M_i denotes the measurement matrix from the i^{th} measurement and b_i denotes the gain matrix associated with the processing of the i^{th} input. Furthermore, a second subscript will denote the partitioned form of that particular matrix. That is, M_{iS} denotes the signal portion of the i^{th} measurement matrix. The partitioned forms will be identical to those already used.

Before proceeding to the development of the algorithm three useful matrix identities will be shown.

Identity I. If R_i is a symmetric matrix and

$$b_i = R_i M_i^T (M_i R_i M_i^T)^{-1} \tag{2.46}$$

then

$$(I - b_iM_i)R_i(I - b_iM_i)^T = (I - b_iM_i)R_i = R_i(I - b_iM_i)^T \quad (2.47)$$

This can be shown by direct substitution of Equation 2.46 into the left side of Equation 2.47. That is,

$$\begin{aligned}(I - b_iM_i)R_i(I - b_iM_i)^T = \\ & R_i - R_iM_i^T(M_iR_iM_i^T)^{-1}M_iR_i - R_iM_i^T(M_iR_iM_i^T)^{-1}M_iR_i \\ & + R_iM_i^T(M_iR_iM_i^T)^{-1}M_iR_iM_i^T(M_iR_iM_i^T)^{-1}M_iR_i \\ & = (I - b_iM_i)R_i = R_i(I - b_iM_i)^T\end{aligned} \quad (2.48)$$

Identity II. If R_i is a symmetric matrix and

$$R_{(i+1)} = (I - b_iM_i)R_i(I - b_iM_i)^T \quad (2.49)$$

where

$$R_o = I \quad (2.50)$$

and

$$b_i = R_iM_i^T(M_iR_iM_i^T)^{-1} \quad (2.51)$$

then

$$M_iR_{(i-1)}M_i^T = 0 \quad (2.52)$$

if and only if

$$M_iR_{(i-1)} = 0 \quad (2.53)$$

This is shown by writing Equation 2.52 in terms of Equations 2.49 and 2.50. That is,

$$M_iR_{(i-1)}M_i^T = M_i(I - b_{(i-1)}M_{(i-1)})R_{(i-2)}(I - b_{(i-1)}M_{(i-1)})^TM_i^T \quad (2.54)$$

Iterating Equation 2.54 to R_o it becomes

$$M_iR_{(i-1)}M_i^T = M_i(I - b_{(i-1)}M_{(i-1)})(I - b_{(i-2)}M_{(i-2)}) \cdots$$
$$(I - b_1M_1)R_o(I - b_1M_1)^T \cdots$$
$$(I - b_{(i-2)}M_{(i-2)})^T(I - b_{(i-1)}M_{(i-1)})^TM_i^T \qquad (2.55)$$

Let

$$C = M_i(I - b_{(i-1)}M_{(i-1)})(I - b_{(i-2)}M_{(i-2)}) \cdots (I - b_1M_1) \qquad (2.56)$$

Then

$$M_iR_{(i-1)}M_i^T = CR_oC^T = CC^T \qquad (2.57)$$

Note that C is a column vector. Then the diagonal elements of CC^T are equal to the squares of the elements in C. Since the squares are all positive numbers, then the only way $CC^T = 0$ is if and only if each element in C is equal to 0. Upon using Identity I, C is simply

$$C = M_IR_{(i-1)} \qquad (2.58)$$

Thus, $M_iR_{(i-1)}M_i^T = 0$ if and only if $M_iR_{(i-1)} = 0$

Identity III. If R_i is a symmetric matrix and

$$M_iR_{(i-1)} = 0 \qquad (2.59)$$

then

$$R_{(i-1)}M_i^T = 0 \qquad (2.60)$$

and

$$(I - b_iM_i)R_{(i-1)}(I = b_iM_i)^T = R_{(i-1)} \qquad (2.61)$$

Since $M_iR_{(i-1)} = 0$ its transpose is also equal to zero. That is,

$$(M_iR_{(i-1)})^T = R^T_{(i-1)}M_i^T = R_{(i-1)}M_i^T = 0 \tag{2.62}$$

Then multiplying out Equation 2.61 and using Equations 2.59 and 2.60

$$\begin{aligned}(I - b_iM_i)R_{(i-1)}(I - b_iM_i)^T \\ &= R_{(i-1)}(I - M_i^Tb_i^T) - b_iM_iR_{(i-1)}(I - b_iM_i)^T \\ &= R_{(i-1)} - R_{(i-1)}M_i^Tb_i^T = R_{(i-1)}\end{aligned} \tag{2.63}$$

We are now in a position to develop the algorithm. Equation 2.45 can be written as the sum of two matrices.

$$P^* = \left[\begin{array}{c:c} \underset{}{\bigcirc}^{(r\times r)} & \bigcirc^{(r\times n)} \\ \hdashline \bigcirc^{(n\times r)} & P_N^{*\,(n\times n)} \end{array}\right] + a\left[\begin{array}{c:c} I^{(r\times r)} & \bigcirc^{(r\times n)} \\ \hdashline \bigcirc^{(n\times r)} & \bigcirc^{(n\times n)} \end{array}\right] \tag{2.64}$$

where a is very large.

The gain matrix for the first measurement is

$$b_1 = P^*M_1^T(M_1P^*M_1^T + V)^{-1} \tag{2.65}$$

Note that $(M_1P^*M_1^T + V)$ is a scalar so b_1 can be written as

$$b_1 = \frac{P^*M_1^T}{(M_1P^*M_1^T + V)} \tag{2.66}$$

Using Equation 2.59 and the partitioned form of M_1

$$b_1 = \frac{\begin{bmatrix} 0 & 0 \\ 0 & P_N^* \end{bmatrix} \begin{bmatrix} M_{1S}^T \\ M_{1N}^T \end{bmatrix} + a \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} M_{1S}^T \\ M_{1N}^T \end{bmatrix}}{[M_{1S} \mid M_{1N}] \begin{bmatrix} 0 & 0 \\ 0 & P^* \end{bmatrix} \begin{bmatrix} M_{1S}^T \\ M_{1N}^T \end{bmatrix} + a\, [M_{1S} \mid M_{1N}] \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} M_{1S}^T \\ M_{1N}^T \end{bmatrix} + V_1} \tag{2.67}$$

Carrying out the multiplication, Equation 2.67 becomes

$$b_1 = \frac{\begin{bmatrix} aM_{1S}^T \\ P_N M_N^T \end{bmatrix}}{aM_{1S}M_{1S}^T + V_1 + M_{1N}P_N^* M_{1N}^T} \tag{2.68}$$

Note that $M_{1S}M_{1S}^T \neq 0$, then choose a such that $aM_{1S}M_{1S}^T >> M_{1N}P_N^* M_{1N}^T + V_1$.

Then b_1 becomes

$$b_1 = \begin{bmatrix} \dfrac{M_{1S}^T}{M_{1S}M_{1S}^T} \\ 0 \end{bmatrix} = \begin{bmatrix} b_{1S} \\ b_{1N} \end{bmatrix} \tag{2.69}$$

To update the a priori covariance matrix the following expression is needed.

$$(I - b_1 M_1) = \begin{bmatrix} I - b_{1S}M_{1S} & - b_{1S}M_{1S} \\ 0 & I \end{bmatrix} \tag{2.70}$$

Upon using Equations 2.63 and 2.71, the a posteriori covariance matrix is

$$P_1 = \left[\begin{array}{c|c} (I - b_{1S}M_{1S}) & - b_{1S}M_{1S} \\ \hline 0 & I \end{array}\right] \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & P_N^* \end{array}\right] \left[\begin{array}{c|c} (I - b_{1S}M_{1S})^T & 0 \\ \hline - M_{1S}^T b_{1S}^T & I \end{array}\right]$$

$$+ \left[\begin{array}{c|c} b_{1S}V_1 b_{1S}^T & 0 \\ \hline 0 & 0 \end{array}\right] + a \left[\begin{array}{c|c} (I - b_{1S}M_{1S})(I - b_{1S}M_{1S})^T & 0 \\ \hline 0 & 0 \end{array}\right] \qquad (2.71)$$

after further multiplication and collection of terms, P_1 becomes

$$P_1 = \left[\begin{array}{c|c} b_{1S}(M_{1S}P_N^*M_{1S}^T + V_1)b_{1S}^T & - b_{1S}M_{1S}P_N^* \\ \hline - P_N^*M_{1S}^T b_{1S}^T & P_N^* \end{array}\right]$$

$$+ a \left[\begin{array}{c|c} (I - b_{1S}M_{1S})(I - b_{1S}M_{1s})^T & 0 \\ \hline 0 & 0 \end{array}\right] \qquad (2.72)$$

Let

$$Q_1 = \left[\begin{array}{c|c} b_{1S}(M_{1S}P_N^*M_{1S}^T + V)b_{1S}^T & - b_{1S}M_{1S}P_N^* \\ \hline - P_N^*M_{1S}^T b_{1S}^T & P_N^* \end{array}\right] \qquad (2.73)$$

and

$$R_1 = \left[\begin{array}{c|c} (I - b_{1S}M_{1S})(I - b_{1S}M_{1S})^T & 0 \\ \hline 0 & 0 \end{array}\right] \qquad (2.74)$$

Using Identity I. Equation 2.75 reduces to

$$R_1 = \left[\begin{array}{c|c} (I - b_{1S}M_{1S} & 0 \\ \hline 0 & 0 \end{array}\right] \tag{2.75}$$

Then P_1 can be written as

$$P_1 = Q_1 + aR_1 \tag{2.76}$$

The estimates of the states can be updated by Equation 1.13. The extrapolation of the states and covariance matrix will not be necessary since $\Delta t = 0$. This requires that $\phi(t) = I$ and $H(t) = 0$. Thus, the second input is ready to be processed and the gain matrix b_2 using Equation 2.77 as the a priori covariance matrix is

$$b_2 = (Q_1M_2^T + aR_1M_2^T)(M_2Q_1M_2^T + aM_2R_1M_2^T + V_2)^{-1} \tag{2.77}$$

Again the quantity under the inverse is a scalar, so b_2 can be written as

$$b_2 = \frac{Q_1M_2^T + aR_1M_2^T}{(M_2Q_1M_2^T + aM_2R_1M_2^T + V_2)} \tag{2.78}$$

If $R_1M_2^T = 0$, then $M_2R_1M_2^T = 0$ and b_2 can be written as

$$b_2 = \frac{Q_1M_2^T}{(M_2Q_1M_2^T + V_2)} \tag{2.79}$$

and thus

$$P_2 = Q_1 - b_2(M_2Q_1M_2^T + V)b_2^T + a(I - b_2M_2)R_1(I - b_2M_2)^T$$
$$= Q_1 - b_2(M_2Q_1M_2^T + V)b_2^T + aR_1 \qquad (2.80)$$

by use of Identity III. The remaining steps in the Kalman filter equations are as normal.

If $R_1M_2^T \neq 0$, then $M_2R_1M_2^T \neq 0$ by Identity II. Then choose a such that

$$aM_2R_1M_2^T \gg M_2Q_1M_2^T + V_2 \qquad (2.81)$$

then b_2 becomes

$$b_2 = \frac{R_1M_2^T}{M_2R_1M_2^T} \qquad (2.82)$$

In partitioned form this can be written as

$$b_2 = \begin{bmatrix} \dfrac{R_{1S}M_{2S}^T}{M_{2S}R_{1S}M_{2S}^T} \\ \text{---------} \\ 0 \end{bmatrix} \qquad (2.83)$$

The covariance matrix is

$$P_2 = (I - b_2M_2)Q_1(I - b_2M_2)^T + b_2V_2b_2^T$$
$$+ a(I - b_2M_2)R_1(I - b_2M_2)^T \qquad (2.84)$$

Define

$$Q_2 = (I - b_2M_2)Q_1(I - b_2M_2)^T + b_2V_2b_2^T \qquad (2.85)$$

and

$$R_2 = (I - b_2M_2)R_1(I - b_2M_2)^T = (I - b_2M_2)R_1 \qquad (2.86)$$

by Identity II; hence

$$P_2 = Q_2 + aR_2 \tag{2.87}$$

The third step or the ith step can be calculated as before. The gain matrix is

$$b_1 = \frac{Q_{i-1}M_i^T + aM_{i-1}M_i^T}{(M_iQ_{i-1}M_i^T + aM_iR_{i-1}M_i^T + V_i} \tag{2.88}$$

Again, if $R_{i-1}M_i^T = 0$ then

$$b_i = \frac{Q_{i-1}M_i^T}{(M_iQ_{i-1}M_i^T + V_i)} \tag{2.89}$$

and

$$P_i = Q_{i-1} - b_i(M_iQ_{i-1}M_i^T + V_i)b_i^T + aR_{i-1} \tag{2.90}$$

If $R_{i-1}M_i^T \neq 0$ then, $M_iR_{i-1}M_i^T \neq 0$, hence

$$b_i = \begin{bmatrix} \dfrac{R_{i-1}M_i^T}{M_iR_{i-1}M_i^T} \\ \text{---------} \\ 0 \end{bmatrix} \tag{2.91}$$

and

$$P_i = (I - b_iM_i)Q_{i-1}(I - b_iM_i)^T + a(I - b_iM_i)R_{i-1} \tag{2.92}$$

One would expect that $R_i = 0$ after there are enough measurements to give an estimate of all the signal variables. Then estimates of the variables would be obtained since the remaining measurements would be redundant information. For example, this can be shown by processing r inputs at once. Assume the first r measurements are such that the rows of M_S are n linear independent combinations of the r signal variables. Then M_S is an invertable matrix. The gain matrix is

$$b_1 = \begin{bmatrix} aM_S^T \\ \hline P_N^* M_N^T \end{bmatrix} [M_N P_N^* M_N^T + aM_S M_S^T + V]^{-1} \tag{2.93}$$

Factoring $\frac{1}{a}$,

$$b_1 = \frac{1}{a} \begin{bmatrix} aM_S^T \\ \hline P_N^* M_N^T \end{bmatrix} \begin{bmatrix} \dfrac{(M_S M_S^T + M_N P_N^* M_N^T + V)}{a} \end{bmatrix}^{-1} \tag{2.94}$$

Taking the limit as $a \to \infty$ b_1 is

$$b_1 = \begin{bmatrix} M_S^T \\ \hline 0 \end{bmatrix} [M_S M_S^T]^{-1} = \begin{bmatrix} M_S^{-1} \\ \hline 0 \end{bmatrix} \tag{2.95}$$

Then

$$(I - b_1 M_1) = \left[\begin{array}{c|c} 0 & -M_S^{-1} M_N \\ \hline 0 & I \end{array}\right] \tag{2.96}$$

The computation of R_1 is

$$R_1 = (I - b_1 M_1) \left[\begin{array}{c|c} I & 0 \\ \hline 0 & 0 \end{array}\right] (I - b_1 M_1)^T = \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & 0 \end{array}\right] \tag{2.97}$$

Thus, after r linearly independent measurements are processed R = 0 and P_i becomes finite. Then one can use the normal Kalman Equations to process the remaining inputs.

After all the measurements at time t_k have been processed, we need to extrapolate the a posteriori covariance matrix P_k. The calculation of P_S^* and P_3^* are not needed because they will be changed to accommodate the complementary constraint. Then P_{k+1}^* is

$$P_{k+1}^* = \phi P_k \phi^T + H \tag{2.98}$$

Using the partitioned form of ϕ, Equation 2.89 becomes

$$P_{k+1}^* = \left[\begin{array}{c|c} \phi_S P_S \phi_S^T + \phi_3 P_3 \phi_S^T + \phi_S P_S \phi_3^T & \phi_S P_S \phi_4^T + \phi_3 P_3^T \phi_4^T + \phi_S P_S \phi_N^T \\ + \phi_3 P_N \phi_3^T + H_S & + \phi_3 P_N \phi_N + H_2 \\ \hline \phi_4 P_S \phi_S^T + \phi_N P_3^T \phi_S^T + \phi_4 P_3 \phi_3^T & \phi_4 P_3 \phi_4^T + \phi_N P_3^T \phi_4^T + \phi_4 P_3 \phi_N^T \\ + \phi_N P_N \phi_3^T + H_3 & + \phi_N P_N \phi_N^T + H_N \end{array}\right] \tag{2.99}$$

The only term that needs to be retained is P_N^* which is

$$P_N^* = \phi_4 P_S \phi_4^T + \phi_N P_3^T \phi_4^T + \phi_4 P_3 \phi_N^T + \phi_N P_N \phi_N^T + H_N \tag{2.100}$$

If $\phi_4 = 0$ as indicated by Bakker (1) then

$$P_{k+1}^* = \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & \phi_N P_N \phi_N^T \end{array}\right] \tag{2.101}$$

The a priori states estimate is

$$\hat{x}'_{k+1} = \begin{bmatrix} \phi_S \underline{x}_S + \phi_3 \underline{x}_N \\ \hline \phi_4 \underline{x}_S + \phi_N \underline{x}_N \end{bmatrix} \tag{2.101}$$

Again if $\phi_4 = 0$, then

$$\hat{x}'_{k+1} = \begin{bmatrix} \phi_S \underline{x}_S + \phi_3 \underline{x}_N \\ \hline \phi_N \underline{x}_N \end{bmatrix} = \begin{bmatrix} \hat{x}'_S \\ \hline \hat{x}'_N \end{bmatrix} \tag{2.102}$$

However, if no weight is to be placed on the a priori signal terms the $\hat{x}'_S$ does not need to be calculated either and it can be arbitrarily set to zero.

The algorithm of the direct filter with sequential processing is now given. Assume at time t_k that P^*_k and $\hat{x}'_k$ are given, then the recommended procedure is:

1. Let $R_{i-1} = I$, where I is the identity matrix
2. Increment i starting with i = 1 until i = number of inputs, then go to step 14.
3. If $R_{i-1} = 0$, go to step 9; otherwise compute $R_{i-1}M^T_{iS}$; if = 0, go to step 9; otherwise go to step 4.
4. Calculate the gain matrix given by

$$b_i = \begin{bmatrix} \dfrac{R_{i-1}M^T_{iS}}{M_{iS}R_{i-1}M^T_{iS}} \\ \hline 0 \end{bmatrix}$$

5. Update the estimates by

$$\hat{x}_i = \hat{x}_{(i-1)} + \begin{bmatrix} b_{iS} \\ \hline 0 \end{bmatrix} (y_i - M_i \hat{x}_{i-1})$$

6. Calculate the a posteriori covariance matrix by

$$P_i = (I - b_iM_i)P_{i-1}(I - b_iM_i)^T + b_iV_ib_i^T$$

7. Compute R_i given by

$$R_i = (I - b_{iS}M_{iS})R_{i-1}$$

8. Go to step 2.

9. Calculate the gain matrix by

$$b_i = \frac{P_{i-1}M_i^T}{(M_iP_{i-1}M_i^T + V_i)}$$

10. Update the states by

$$\hat{x}_i = \hat{x}_{i-1} + b_i(y_i - M_i\hat{x}_{i-1})$$

11. Compute the a posteriori covariance matrix by

$$P_i = P_{i-1} - b_i(M_iP_{i-1}M_i^T + V_i)b_i^T$$

12. Let $R_i = R_{i-1}$.

13. Go to step 2.

14. Extrapolate the estimate of the states ahead by

$$\hat{x}'_N = \phi_N\hat{x}_N \qquad \hat{x}'_S = 0$$

15. Extrapolate the a posteriori covariance matrix to give the a priori covariance matrix by

$$P^*_N = \phi_NP_N\phi_N^T + H_N$$

If the above algorithm is used, an optimal estimate of the signals in the least squares sense is obtained. Also the estimate of the signals will satisfy the complementary constraint. The method described above does not require any matrix inversions, which in a large scale problem can amount to an appreciable time savings. Also note that the modeling of the signal variables is not critical since the ϕ_S matrix is not needed. Hence, the signal can be modeled any way that is desired. A look at some possible applications and uses for the above filters will be presented in Chapter IV.

III. LINEAR ESTIMATION FOR TIME-CONTINUOUS SYSTEMS WITH THE COMPLEMENTARY CONSTRAINT

The development of the complementary filter in Chapter II dealt with the discrete data input system. The purpose of this chapter is to consider the case where the inputs are continuous functions of time. The development of a Kalman-Bucy complementary filter is presented in this chapter. This is accomplished by a limiting technique similar to that employed by Sorenson (14). That is, the discrete time complementary Kalman filter equations are used and Δt is allowed to go to zero.

The development will produce a set of matrix differential equations. The solutions of these equations are not presented here, because they can be found in Reid (13). The differential equation will be presented in a block diagram to suggest a means of implementing the Kalman-Bucy complementary filter.

The filter equations for the discrete-time distortionless constraint are as follows. The gain matrix is

$$K_k \triangleq \begin{bmatrix} K_{Sk} \\ \hdashline K_{Nk} \end{bmatrix} \tag{3.1}$$

where

$$K_{Sk} = [M_{Sk}^T(M_{Nk}P_{Nk}^*M_{Nk}^T + V_k)^{-1}M_{Sk}]^{-1}M_{Sk}^T(M_{Nk}P_{Nk}^*M_{Nk}^T + V_k)^{-1} \tag{3.2}$$

$$K_{Nk} = P_{Nk}^*M_{Nk}^T(M_{Nk}P_{Nk}^*M_{Nk}^T + V_k)^{-1}[I - M_{Sk}K_{Sk}] \tag{3.3}$$

The a posteriori covariance matrix is

$$P_k = (I - K_kM_k)P_k^*(I - K_kM_k)^T + K_kV_kK_k^T \tag{3.4}$$

The estimation equation is

$$\hat{x}_k = \hat{x}'_k + K_k[y_k - M_k\hat{x}'_k] \tag{3.5}$$

where

$$\hat{x}'_k = \phi_{k,k-1}\hat{x}_{k-1} \tag{3.6}$$

Upon using the constraint as derived by Bakker (1)

$$K_{Sk}M_{Sk} = I \tag{3.7}$$

$$K_{Nk}M_{Sk} = 0 \tag{3.8}$$

Equation 3.4 becomes in partitioned form

$$\left[\begin{array}{c|c} P_{Sk} & P_{3k} \\ \hline P_{3k}^T & P_{Nk} \end{array}\right] = \left[\begin{array}{c|c} K_{Sk}(M_{Nk}P^*_{Nk}M^T_{Nk} + V_k)K^T_{Sk} & K_{Sk}(M_{Nk}P^*_{Nk}M^T_N + V_k)K^T_{Nk} \\ & - K_{Sk}M_{Nk}P^*_{Nk} \\ \hline K_{Nk}(M_{Nk}P^*_{Nk}M^T_{Nk} + V_k)K^T_{Sk} & P^*_{Nk} - P^*_{Nk}M^T_{Nk}K^T_{Nk} \\ - P^*_{Nk}M^T_{Nk}K^T_{Sk} & - K_{nk}M_{Nk}P^*_{Nk} \\ & + K_{Nk}(M_{Nk}P^*_{Nk}M^T_{Nk} + V_k)K^T_{Nk} \end{array}\right] \tag{3.9}$$

Equation 3.5 becomes in partitioned form

$$\left[\begin{array}{c} \hat{x}_{Sk} \\ \hline \hat{x}_{Nk} \end{array}\right] = \left[\begin{array}{c} - K_{Sk}M_{Nk}\hat{x}'_{Nk} + K_{Sk}(y_k) \\ \hline (I - K_{Nk}M_{Nk})\hat{x}'_{Nk} + K_{Nk}y_k \end{array}\right] \tag{3.10}$$

Equations 3.1 to 3.10 for the discrete-time models can be used to derive the Kalman filter for the time-continuous systems and measurement process with the distortionless constraint. This is accomplished with a limiting argument employed by Sorenson (14). Before the limiting argument the white noise sequences will be replaced with white noise processes.

Consider a dynamical system described by a linear, vector differential equation

$$\frac{dx}{dt} = A(t)x + G(t)\omega(t) \tag{3.11}$$

Let $\omega(t)$ be a gaussian white noise process with moments prescribed as

$$E[\omega(t)] = 0 \qquad \text{for all } t \tag{3.12}$$

$$E[\omega(t)\omega^T(t)] = Q(t)\delta(t - \tau) \qquad \text{for all } t, \tau \tag{3.13}$$

The $Q(t)$ is a symmetric, non-negative-definite matrix and $\delta(t - \tau)$ represents the Dirac delta function.

The measurement model is assumed to be

$$y(t) = M(t)x(t) + v(t) \tag{3.14}$$

where $v(t)$ is a gaussian white noise process with moments prescribed as

$$E[v(t)] = 0 \qquad \text{for all } t \tag{3.15}$$

$$E[v(t)v^T(t)] = V(t)\delta(t - \tau) \qquad \text{for all } t, \tau \tag{3.16}$$

The continuous Kalman filter with the distortionless constraint can be obtained from Equations 3.1 through 3.10 by letting $\Delta t \to 0$. However, fundamental differences exist between white noise processes and white noise sequences. These differences will be accounted for before introducing the limiting argument.

The covariance of the random sequence v_k has been defined as

$$E[v_k v_j^T] = V_k \delta_{kj} \qquad \text{for all } k, j \tag{3.17}$$

If the time interval Δt between adjacent sampling times is permitted to become arbitrarily small, the noise will contain no power. That is,

$$\lim_{\substack{n \to \infty \\ \Delta t \to o}} \sum_{j=1}^{n} V_k \delta_{kj} \Delta t = 0 \tag{3.18}$$

In other words for arbitrarily small Δt there is no noise in the measurement so the estimate problem is uninteresting.

To circumvent this difficulty, introduce the constraint that

$$E[v_k v_j^T] \, \Delta t = V_k \delta_{kj} \qquad \text{for all } k, j \tag{3.19}$$

for any sampling interval Δt and a prescribed matrix v_k. With this restriction it is apparent that

$$\lim_{\substack{n \to \infty \\ \Delta t \to o}} \sum_{j=1}^{n} E[v_k v_j^T] \, \Delta t = V_k \tag{3.20}$$

The constraint in Equation 3.19 is equivalent to requiring that the noise sequences $\{\omega_k\}$ and $\{v_n\}$ be replaced by

$$\frac{\omega_k}{(\Delta t)^{\frac{1}{2}}} \quad \text{and} \quad \frac{v_k}{(\Delta t)^{\frac{1}{2}}}$$

where ω_k and v_k are as described before. In other words, in Equations 3.11 and 3.14 replace v_k and ω_k with $\frac{V_k}{\Delta t}$ and $\frac{Q_k}{\Delta t}$ respectively.

Consider the dynamical system

$$x_k = \phi_{k,k-1} x_{k-1} + \Delta_{k,k-1} \frac{\omega_{k-1}}{(\Delta t)^{\frac{1}{2}}} \tag{3.21}$$

and the measurement process

$$y_k = M_k x_k + \frac{V_k}{(\Delta t)^{\frac{1}{2}}} \tag{3.22}$$

Now $\phi_{k,k-1}$ is the state transistion matrix and can be expanded as

$$\phi_{k,k-1} = I + A(t_{k-1})\Delta t + 0(\Delta t) \tag{3.23}$$

where $0(\Delta t)$ denotes terms of greater than first order in Δt.

Also, $\Delta_{k,k-1}$ can be written as

$$\Delta_{k,k-1} = G(t_{k-1})\Delta t + 0(\Delta t) \tag{3.24}$$

Now using Equations 3.23 and 3.24 in Equation 3.21 and rearranging terms,

$$\frac{x_k - x_{k-1}}{\Delta t} = A(t_{k-1})x_{k-1} + G(t_{k-1}) \frac{\omega_{k-1}}{(\Delta t)^{\frac{1}{2}}} + \frac{0(\Delta t)}{\Delta t} \tag{3.25}$$

Now define the processes $v(t)$ and $\omega(t)$ such that

$$v(t) \triangleq \frac{v_k}{(\Delta t)^{\frac{1}{2}}} \qquad \text{for } t_{k-1} \le t < t_k \tag{3.26}$$

$$\omega(t) \triangleq \frac{\omega_k}{(\Delta t)^{\frac{1}{2}}} \qquad \text{for } t_{k-1} \le t < t_k \tag{3.27}$$

and let

$$\lim_{\Delta t \to o} \frac{\delta_{kj}}{\Delta t} = \delta(t - \tau) \tag{3.28}$$

Letting $\Delta t \to o$ in Equation 3.25, Equation 3.11 is obtained and letting $\Delta t \to o$ in Equation 3.22, Equation 3.14 is obtained.

Now we are in a position to derive the Kalman filter equations. The extrapolated error covariance matrix for the modified noise sequence is

$$P_k^* = \phi_{k,k-1} P_{k-1} \phi_{k,k-1}^T + \Delta_{k,k-1} \frac{Q_{k-1}}{\Delta t} \Delta_{k,k-1}^T \qquad (3.29)$$

Substituting Equations 3.23 and 3.24 this becomes

$$P_k^* = P_{k-1} + A(t_{k-1})P_{k-1}\Delta t + P_{k-1}A^T(t_{k-1})\Delta t$$
$$+ G(t_{k-1})Q_{k-1}G^T(t_{k-1})\Delta t + 0(\Delta t) \qquad (3.30)$$

In partitioned form this becomes

$$\left[\begin{array}{c|c} P_{Sk}^* & P_{3k}^* \\ \hline P_{3k}^{*T} & P_{Nk}^* \end{array}\right] = \left[\begin{array}{c|c} P_{Sk-1} & P_{3k-1} \\ \hline P_{3k-1}^T & P_{Nk-1} \end{array}\right]$$

$$+ \left[\begin{array}{l|l} A_S(t_{k-1})P_{Sk-1}\Delta t + A_3(t_{k-1})P_{3k-1}^T\Delta t & A_S(t_{k-1})P_{3k-1}\Delta t + A_3(t_{n-1})P_{nk-1}\Delta t \\ + P_{Sk-1}A_S^T(t_{k-1})\Delta t + P_{3k-1}A_3^T(t_{k-1}\Delta t & + P_{3k-1}A_N^T(t_{n-1})\Delta t \\ \hline A_N(t_{k-1})P_{3k-1}^T\Delta t + P_{3k-1}^T A_S^T(t_{k-1})\Delta t & A_N(t_{k-1})P_{Nk-1}\Delta t + P_{Nk-1}A_N^T(t_{k-1})\Delta t \\ + P_{Nk-1}A_3^T(t_{k-1})\Delta t & \end{array}\right]$$

$$+ \left[\begin{array}{c|c} G_{Sk-1}Q_{k-1}G_{Sk-1}^T\Delta t & G_{Sk-1}Q_{k-1}G_{Nk-1}^T\Delta t \\ \hline G_{Nk-1}Q_{k-1}G_{Sk-1}^T\Delta t & G_{Nk-1}Q_{k-1}G_{Nk-1}^T\Delta t \end{array}\right] + 0(\Delta t) \qquad (3.31)$$

Now this can be written as three equations

$$P^*_{Sk} = P_{Sk-1} + A_S(t_{k-1})P_{Sk-1}\Delta t + A_3(t_{k-1})P^T_{3k-1}\Delta t$$

$$+ P_{Sk-1}A^T_S(t_{k-1})\Delta t + P_{3k-1}A^T_3(t_{k-1})\Delta t$$

$$+ G_{Sk-1}Q_{k-1}G^T_{Sk-1}\Delta t + 0(\Delta t) \qquad (3.32a)$$

$$P^*_{3k} = P_{3k-1} + A_S(t_{k-1})P_{3k-1}\Delta t + A_3(t_{k-1})P_{Nk-1}\Delta t$$

$$P_{3k-1}A^T_N(t_{k-1})\Delta t + G_{Sk-1}Q_{k-1}G_{Nk-1}\Delta t + 0(\Delta t) \qquad (3.32b)$$

$$P^*_{Nk} = P_{Nk-1} + A_N(t_{k-1})P_{Nk-1}\Delta t + P_{Nk-1}A^T_N(t_{k-1})\Delta t$$

$$+ G_{Nk-1}Q_{k-1}G^T_{Nk-1}\Delta t + 0(\Delta t) \qquad (3.32c)$$

Now look at

$$\lim_{\Delta t \to o} P_{Sk-1} = \lim_{\Delta t \to o} K_{Sk-1}(M_{Nk-1}P^*_{Nk-1}M^T_{Nk-1}$$

$$+ \frac{v_{k-1}}{\Delta t})K^T_{Sk-1} = \infty \qquad (3.33)$$

Thus $P^*_S = P_S = \infty$

Taking the limit of P_{3k-1} as $\Delta t \to 0$ yields

$$\lim_{\Delta t \to o} P_{3k-1} = \lim_{\Delta t \to o}[K_{Sk-1}(M_{Nk-1}P^*_{Nk-1}M^T_{Nk-1} + \frac{V_k}{\Delta t})K^T_{Nk-1}$$

$$- K_{Sk-1}M_{Nk-1}P^*_{Nk-1}]$$

$$= \lim_{\Delta t \to o} [K_{Sk-1}(M_{Nk-1}P^{*}_{Nk-1}M_{Nk-1}\Delta t + V_k) \frac{K^T_{Nk-1}}{\Delta t}$$

$$- K_{Sk-1}M_{Nk-1}P^{*}_{Nk-1}]$$

Determine the following limits:

$$\lim_{\Delta t \to o} K_{Sk-1} = [M^T_{Sk}V^{-1}_k M_{Sk}]^{-1}M^T_{Sk}V^{-1}_k \triangleq K'_S \tag{3.34}$$

$$\lim_{\Delta t \to o} \frac{K_{Nk-1}}{\Delta t} = P^{*}_{Nk}M^T_{Nk}V^{-1}_k[I - M_{Sk}(M^T_{Sk}V^{-1}_k M_{Sk})^{-1}M^T_{Sk}V^{-1}_k] \triangleq K'_N \tag{3.35}$$

$$\lim_{\Delta t \to o} K_{Nk-1} = 0 \tag{3.36}$$

Using Equation 3.34 and 3.35 then $\lim_{\Delta t \to o} P_{3k-1}$ becomes

$$\lim_{\Delta t \to o} P_{3k-1} = P_{3k} = -(M^T_{Sn}V^{-1}_k M_{Sk})^{-1}M^T_{Sk}V^{-1}_k M_{Nk}P^{*}_{Nk} \tag{3.37}$$

Therefore,

$$\lim_{\Delta t \to o} P^{*}_{3k} = P^{*}_{3}(t) = -[M^T_S(t)v(t)^{-1}M_S(t)]^{-1}.$$

$$M^T_S(t)v(t)^{-1}M_N(t)P^{*}_N(t) \tag{3.38}$$

Using Equation 3.9 and 3.32c, and rearranging terms

$$\frac{P^{*}_{Nk} - P^{*}_{Nk-1}}{\Delta t} = \frac{K_{Nk-1}}{\Delta t}(M_{Nk-1}P^{*}_{Nk-1}N^T_{Nk-1} + \frac{V_{k-1}}{\Delta t})K^T_{Nk-1}$$

$$- P^{*}_{Nk-1}M^T_{Nk-1}\frac{K^T_{Nk-1}}{\Delta t} - \frac{K_{Nk-1}}{\Delta t}M_{Nk-1}P^{*}_{Nk-1} + A_N(t_{k-1})P_{Nk-1}$$

$$+ P_{Nk-1}A^T_N(t_{k-1}) + G_{Nk-1}Q_{k-1}G^T_{Nk-1} + \frac{0(\Delta t)}{\Delta t} \tag{3.39}$$

$$\lim_{\Delta t \to 0} \frac{P^*_{Nk} - P^*_{Nk-1}}{\Delta t} = K'_N V_k K'^T_N - K'_N M_N P^*_{Nk} + A_N(t) P^*_{Nk} + P^*_{Nk} A^T_N(t)$$

$$+ G_{Nk} Q_k G^T_{Nk} \triangleq \frac{dP^*_N}{dt} \qquad (3.40)$$

Equation 3.35 can be rewritten as

$$K'_N = P^*_N M^T_N V^{-1} - P^*_N M^T_N V^{-1} M_S (M^T_S V^{-1} M_S)^{-1} M^T_S V^{-1} \qquad (3.41)$$

Now using this K'_N then

$$\frac{dP^*_N}{dt} = A_N(t) P^*_N + P^*_N A^T_N(t) - P^*_N M^T_N [V^{-1} - V^{-1} M_S (M^T_S V^{-1} M_S)^{-1} M^T_S V^{-1}]$$

$$M_N P^*_N + G_N Q(t) G^T_N \qquad (3.42)$$

This equation has the form of a matrix Ricatti equation and shall be discussed below.

The estimate Equation 3.10 for the signal variables is

$$\hat{x}_{Sk} = - K_{Sk} M_{Nk} \hat{x}'_{Nk} + K_k (y_{Sk} + y_{Nk}) \qquad (3.43)$$

$$\hat{x}'_k = \begin{bmatrix} \hat{x}'_{Sk} \\ \hat{x}'_{Nk} \end{bmatrix} = \phi_{k,k-1} \hat{x}_{k-1}$$

$$= \begin{bmatrix} \phi_{Sk,k-1} & \phi_{3k,k-1} \\ 0 & \phi_{Nk,k-1} \end{bmatrix} \begin{bmatrix} \hat{x}_{Sk-1} \\ \hat{x}_{Nk-1} \end{bmatrix} \qquad (3.44)$$

So

$$\hat{x}'_{Nk} = \phi_{Nk,k-1} \hat{x}_{Nk-1} = \hat{x}_{Nk-1} + A_N(t_{k-1}) \hat{x}_{Nk-1} \Delta t + 0(\Delta t) \qquad (3.45)$$

Using Equation 3.45 in Equation 3.43 it becomes

$$\hat{x}_{Sk} = - K_{Sk}M_{Nk}\hat{x}_{Nk-1} - K_{Sk}M_{Nk}A_N(t_{k-1})\hat{x}_{Nk-1}\Delta t +$$

$$+ K_{Sk}(y_k) + 0(\Delta t) \tag{3.46}$$

$$\lim_{\Delta t \to o} \hat{x}_{Sk} \triangleq \hat{x}_S = K'_S y(t) - K'_S M_N \hat{x}_N = K'_S[y(t) - M_N \hat{x}_N] \tag{3.47}$$

$$\hat{x}_{Nk} = (I - K_{Nk}M_{Nk})\hat{x}_{Nk-1} + (I - K_{Nk}M_{Nk})A_N(t_{k-1})\hat{x}_{Nk-1}\Delta t$$

$$+ K_{Nk}y_k + 0(\Delta t) \tag{3.48}$$

Upon rearranging terms and taking the limit

$$\lim_{\Delta t \to o} \frac{\hat{x}_{Nk} - \hat{x}_{Nk-1}}{\Delta t} \triangleq \frac{d\hat{x}_N}{dt} = A_N(t)\hat{x}_N - K'_N M_{Nk}\hat{x}_{Nk} + K'_N y_k$$

$$= A_N(t)\hat{x}_N + K'_N(t)[y(t) - M_N(t)\hat{x}_N] \tag{3.49}$$

The fact that $P_S = \infty$ makes sense because of the distortionless constraint. The distortionless constraint says to ignore the a priori statistic of the signal variables and hence, are not used in the gain equation or estimation equations.

A summary of the Kalman filter equation for the time-continuous filter with the distortionless constraint follows.

1. The covariance differential equation matrix is

$$\frac{dP_N}{dt} = A_N(t)P_N + P_N A_N(t)$$

$$+ P_N M_N[V^{-1} - V^{-1}M_S(M_S^T V^{-1}M_S)M_S^T V^{-1}]M_N^T P_N \tag{3.50}$$

2. The gain equations are

$$K_S'(t) = [M_S(t)v(t)^{-1}M_S^T(t)]^{-1}M_S^T(t)v(t)^{-1}$$

$$K_N'(t) = P_N M_N^T(t)[v(t)^{-1} - v(t)^{-1}M_S(t)$$

$$(M_S^T(t)v(t))^{-1}M_S^T(t)v(t)^{-1}] \tag{3.51}$$

3. The estimate equations are

$$\hat{x}_S = K_S'(t)[y(t) - M_N(t)\hat{x}_N] \tag{3.52}$$

$$\frac{d\hat{x}_N}{dt} = A_N(t)\hat{x}_N + K_N'(t)[y(t) - M_N(t)\hat{x}_N] \tag{3.53}$$

In block diagram form the filter is shown in Figure 3.1.

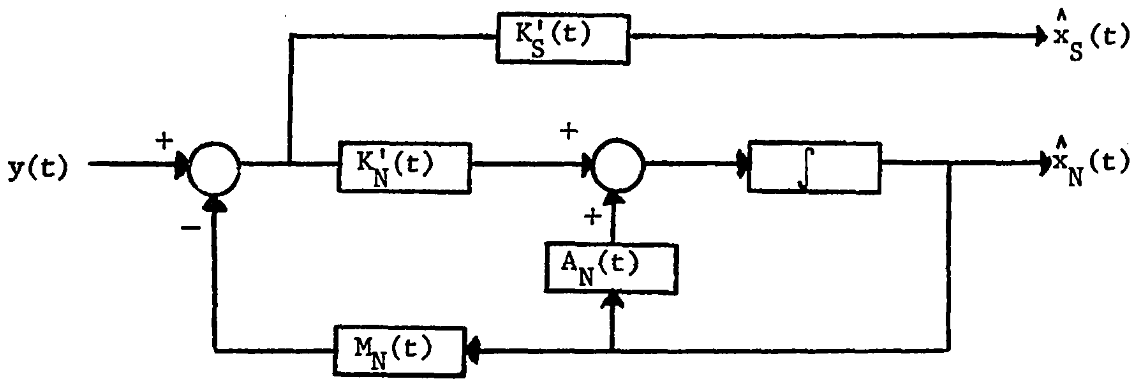


Figure 3.1. Block diagram of continuous complementary Kalman filter.

This completes the development of the continuous complementary filter. Before the filter can be implemented a solution must be found for the matrix differential Equation 3.50 through 3.53. The solutions are not shown here, but can be found in Reid (13).

IV. APPLICATIONS AND COMPARISON EFFORT INVOLVED IN THE DIRECT AND INDIRECT METHODS OF IMPLEMENTARY THE COMPLEMENTARY CONSTRAINT

A. Applications of Complementary Filters

The purpose of this section is to suggest some practical applications for the direct filter. The motivation for this thesis was to devise a totally integrated inertial navigation system. However, the author does not suggest that the direct method will solve all navigation problems. There have been certain restrictions and assumptions made that do not fit every navigation system. The direct filter yields a more complete integrated system than has been developed so far and in certain cases is even easier to implement than the methods proposed to date.

As far as a totally integrated inertial navigation system is concerned this thesis only considers the case where the complementary constraint is needed. This, in the author's viewpoint, is the area that needs to be explored. If one knows the statistics of the signal then the Kalman filter can be used to estimate the values of the signals. If the measurements are linearly independent then the inputs can be processed sequentially. Upon input failure or unavailability of inputs, the Kalman filter can merely omit these measurements and proceed with the remaining measurements. That is precisely the purpose of this thesis; to be able to sequentially process the measurements, such that if there is an input failure a backup system is not required. In Chapter II an algorithm was developed that precisely accomplishes this task. The inputs can be processed sequentially, and the result is an optimal estimation of the signal variables in the least squares sense with the complementary constraint.

The complementary constraint is used in virtually all terrestrial navigation systems, as pointed out by Brock and Schmidt (4). The reason is the statistics of the signal are not known well enough or are virtually impossible to describe mathematically. Also, in many cases where the statistics are known the complementary constraint does not degrade the performance appreciably as pointed outby Brock and Schmidt (4).

Huddle (10) described a navigation system that is typical of many systems today. He estimated position and velocity using an inertial navigation unit, doppler radar, Loran system and star tracker. He then considered the following modes of operation: free-inertial navigation mode, doppler inertial navigation mode, Loran-inertial navigation mode, and astro-inertial navigation mode. In the free-inertial mode, Huddle indicated that the errors grow with time. To keep the errors bounded and for a better estimate of the signals,aiding sources were used. Actual flight tests were made of the above mentioned modes and detailed error curves were plotted for each flight and mode. It would seem, however, that the best estimates of the signal would be obtained if all aiding sources were used at once instead of using the different modes. The reason all aiding sources are not used is the fact that they are not available at all times. For example, the Loran system only works if the vechicle is in range of Loran ground stations. The star tracker only works at night. The doppler radar may not work effectively over water. However, the direct filter as implemented in Chapter II, will allow one to use all the aiding sources that are available. This even allows for a failure in the inertial navigation system, which most systems propossed to date do not allow.

Another possible advantage might involve the case of spurious errors in the measurements, when a check can be made on the inputs to determine if they are acceptable or not. One way would be to compare the measurement y_i with $M_i\hat{x}_i'$ by the equation $(y_i - M_i\hat{x}_i') \leq F(t)$, where $F(t)$ is the maximum bound to be placed on the difference. From the dynamics of the system there will have to be an upper bound on this difference. For example, at time t_k assume $y_1 = S_1 + n_2(t)$ where S_1 is a velocity variable. From the previous data we have an a priori estimate of the velocity at time t_k. If the vehicle is an aircraft, it is obvious that it can only accelerate or deaccelerate at a maximum rate. Thus, the difference must lie within certain bounds. Figure 4.1 demonstrates this idea.

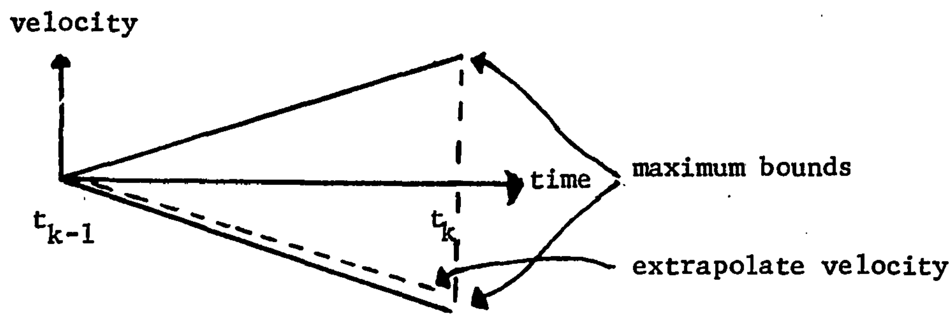


Figure 4.1. Bounds on velocity variables.

If the measurement at time t_k lies outside these bounds, then that measurement will be ignored. Since it processes all inputs sequentially ignoring an input does not affect the direct filter. Gaines (9) used a chi square test to protect the system from faulty measurements. The author does not suggest any one method, but indicates that the tests will be the same, for the direct and indirect filter and failure detection schemes will not be presented further.

Another advantage for the direct filter is the case where the statistics of the signal might be known only part of the time or where the statistics change drastically at some time. It is a trivial matter to change the normal Kalman filter equations to the direct filter, by just setting the a priori signal variance terms to a large number. This would involve no algorithm change with a mimimal amount of additional processing time. If the algorithm in Chapter II were being used, to switch from the complementary filter to the coventional Kalman filter would be accomplished by omitting steps 1 through 7.

A disadvantage of the direct filter is the fact that a larger number of states are involved, i.e., the direct filter models both signal and noise states while the filters to date model only estimate noise. Therefore larger matrices are involved in the direct filter. However, the next section indicates that computation time may be shorter if there are a large number of redundant measurements.

B. Computational Comparison of Direct to Indirect Filters

Based on the assumption that a better estimate of signals can be made if all aiding sources are used, computation time will be investigated. Unger and Ott (15) demonstrated that considerable improvement in accuracy can be obtained by using all additional redundant information as compared to pure inertial modes. Therefore, a comparison will be made between the direct filter and the conventional filter using all redundant information. Benning (2) introduced the general complementary filter which would be typical of most filter schemes to date as far as determining computational time. Benning's (2) method was described in Chapter I and referred to as the indirect filter. The indirect filter is shown in Figure 4.2.

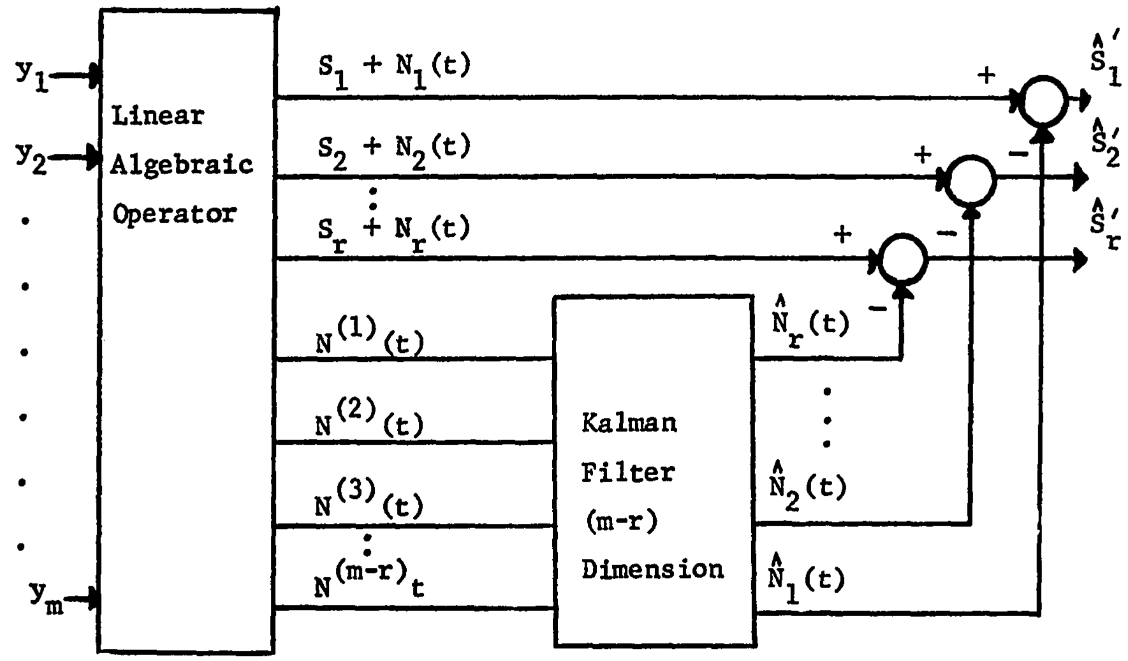


Figure 4.2. Block diagram of indirect filter.

The (m-r) dimensional Kalman filter operates on the (m-r) linear combinations of the noises to give the optimal estimate of $N_i(t)$. In the indirect filter, if one input fails or is not available the filter has to be changed to accommodate the remaining inputs, thus, requiring a backup system. If all possible combinations of errors are considered the number of backup systems required is given by Equation 4.1, which was derived in Chapter I.

$$B = \sum_{i=1}^{(m-r+1)} \binom{m}{i} \tag{4.1}$$

As indicated in Chapter I, B can be very large in a system with a large amount of redundancy. Instead of having backup systems for each case an alternate method would be to have an additional algorithm to recompute the algebraic operator and modify the Kalman filter when a failure occurred. The remaining inputs $y_i(t)$ would be solved to yield r equations of the form $S_i + N_i(t)$ and (m-r-1) equations consisting of linear combinations of noise.

A comparison between the indirect and direct filter computation time and computer memory will be made. A good comparison of the computational times involved is the number of multiplies required. Multiplies are usually an order of magnitude higher than simple additions. For example, the computer in Gaines's (9) paper has a speed of 24 μsec for a multiply and 4 μsec for an addition with a word size of 20 bits. Thus, we will examine numbers of multiplies that are required for the direct filter and the indirect filter. The signal states will depend on what types of quantities are desired to be estimated, i.e., position; velocity; attitude; pitch; roll; etc. The characteristics of the measurement noises will determine the number of noise state that are needed. That is, if the measurement noise is white, there will be no noise vector. However, if it is something other than white, in order to model it for our Kalman filter, we will have to think of it as an output of some shaping filter driven by white noise. Thus, we will have noise states the number of which depends on the characteristics of the shaping filter.

Let's consider the general case, where we have:

R signal variables, i.e., the number of variables that are to be estimated.

G noise variables

P measurements of the signal variables

Therefore, the filter will have R + G states in the direct filter and G states in the indirect filter. The equations for the number of multiplies for the direct filter is

$$M_D = 3R^4 - R^3 - R - 3RG + 4R^3G + 2R^2G^2 - 4R^2G - 2RG^2$$
$$+ 2G^3 + G^2 + P(2R^2 + 4RG + 2G^2 + 4R + 4G) \qquad (4.2)$$

The equation for the number of multiplies for the indirect filter is

$$M_I = P^2 + 2(P-R)(G^2 + G) + 2G(P - R)^2 + 2(P - R)^3$$
$$+ RG + 2G^3 + G^2 \qquad (4.3)$$

Equation 4.3 includes multiplies needed in order to make a fail-safe system. For brevity, the derivation is given in Appendix D.

Upon examining Equations 4.2 and 4.3 we find that the difference will be small, if R is small, P is large, and G is large. This is intuitively sound because the direct filter operates on the R + G variables and the indirect operates on G. The indirect must also take an inverse in its algorithm. If the noise states were large compared with the signal states, the indirect approach would take longer; and if P were large, this would make the noise states greater. Therefore, a look at the percentage increase of the direct filter over the indirect will be examined. This percentage will be denoted by P% and is

$$P\% = \frac{(M_D - M_I) \times 100}{M_I} \qquad (4.4)$$

Tables 1 through 8 list the percentage increase in multiplies for values of R from 1 to 8 and various values of G and P. Note values of P less than R are not listed, because it would then be impossible to have the complementary constraint. For constant values of G and R, and as P increased the percentages went negative. This means that as P becomes larger the direct filters computation time is shorter than the indirect filter.

In conclusion, if a fail-safe system using the complementary filter is desired, several factors must be considered. If many noise states and much redundant information exist, the designer must decide which type of filter mechanization will be used. The direct filter may require more computation time, but less memory allocation than the indirect filter. However, if there are a large number of redundant states and many noise variables compared to the signal states the computation time may even be smaller than the indirect filter. The percentage increase of the memory requirement of the indirect filter over the direct filter is also shown in Tables 1 through 8. Note, that the indirect filter generally requires more memory than the direct filter because of the matrices needed in order to be fail-safe. The percentage tends to increase as the redundancy increases.

In conclusion, if a fail-safe system using the complementary constraint is desired, several factors must be considered. The direct filter generally requires more computation time, but less memory requirements than the indirect filter. As indicated by Gaines (9), computation time is not the critical factor in Kalman filter mechanization, but it is usually the computer size that places the restriction on the mechanization. However, if there are a large number of redundant states the computation time of the direct filter may even be smaller. Then the direct filter would be superior to the indirect filter. The next chapter will present two examples demonstrating the direct and indirect filters.

Table 4.1. Percentage increase in multiplies of direct over indirect methods and percentage increase in computor memory of indirect over direct methods for (value of R) signal variables

Top numbers = Percentage increase in multiplies
Bottom numbers = Percentage increases in computer memory
G = Number of noise states
xxx = Number greater than 1000%

		Number of measurements											
R	G	1	2	3	4	5	6	7	8	9	10	11	12
1	1	240	106	8	-60								
		-55	10	72	145								
1	2	95	70	20	-21								
		-39	3	45	96								
1	3	53	48	22	-6								
		-29	0	31	68								
1	4	36	35	20	0	-21							
		-23	-1	22	50	82							
1	5	27	27	18	4	-12							
		-19	-2	17	39	65							
1	6	21	22	15	5	-6							
		-16	-2	14	32	53							
1	7	17	18	14	6	-3							
		-14	-2	11	26	44							
1	8	15	15	12	6	-1							
		-12	-2	9	22	37							
1	9	13	13	11	6	0	-7						
		-11	-2	8	19	32	47						
1	10	11	12	10	6	1	-5						
		-10	-2	7	17	28	41						
1	11	10	10	9	6	1	-3						
		-9	-2	6	15	25	36						
1	12	9	9	8	6	2	-2						
		-8	-2	5	13	23	33						

Table 4.2. Percentage increase in multiplies of direct over indirect methods and percentage increase in computer memory of indirect over direct methods for (value of R) signal variables

Top numbers = Percentage increase in multiplies
Bottom numbers = Percentage increases in computer memory
G = Number of noise states
xxx = Number greater than 1000%

		Number of measurements											
R	G	1	2	3	4	5	6	7	8	9	10	11	12
2	1		999	559	230	79	8	-41					
			-42	0	41	90	144	203					
2	2		578	366	197	90	27	-12					
			-37	-4	28	66	108	154					
2	3		320	242	158	88	38	2	-27				
			-31	-5	20	49	83	119	158				
2	4		206	172	126	81	42	12	-11				
			-26	-6	15	38	65	95	126				
2	5		147	129	102	72	43	18	-2				
			-22	-6	11	31	53	77	103				
2	6		112	102	85	64	42	22	4	-12			
			-19	-5	9	25	44	64	86	109			
2	7		90	84	72	57	40	23	8	-5			
			-17	-5	7	21	37	54	73	93			
2	8		75	71	62	51	38	24	11	-1			
			-15	-5	6	18	32	47	63	81			
2	9		64	61	55	46	35	24	13	2	-8		
			-14	-5	5	16	28	41	55	70	87		
2	10		55	53	49	42	33	24	14	4	-4		
			-13	-4	4	14	24	36	49	62	77		
2	11		49	47	44	38	31	23	15	6	-1		
			-12	-4	4	12	22	32	44	56	69		
2	12		44	42	39	35	29	22	15	7	0	-7	
			-11	-4	3	11	19	29	39	50	62	74	

Table 4.3. Percentage increase in multiplies of direct over indirect methods and percentage increase in computer memory of indirect over direct methods for (value of R) signal variables

Top numbers = Percentage increase in multiplies
Bottom numbers = Percentage increases in computer memory
G = Number of noise states
xxx = Number greater than 1000%

		Number of measurements											
R	G	1	2	3	4	5	6	7	8	9	10	11	12
3	1			999	999	742	359	171	73	17	-19		
				-33	-3	26	61	101	145	191	240		
3	2			999	999	607	339	184	93	37	1	-29	
				-32	-6	18	47	80	116	154	194	237	
3	3			962	697	469	298	182	104	52	16	-10	
				-30	-8	13	37	64	94	125	159	194	
3	4			614	487	363	255	171	107	61	27	2	-19
				-27	-8	10	30	52	77	104	132	161	192
3	5			429	361	287	217	156	106	66	36	12	-7
				-24	-8	7	24	43	64	87	111	136	163
3	6			321	280	233	186	141	102	69	41	19	1
				-21	-8	5	20	36	55	74	95	116	139
3	7			252	226	194	161	127	96	69	45	25	7
				-19	-8	4	17	31	47	64	82	101	121
3	8			206	188	165	141	115	90	67	47	29	13
				-17	-7	3	14	27	41	56	71	88	106
3	9			172	160	143	125	105	85	65	47	31	17
				-16	-7	2	12	24	36	49	63	78	94
3	10			148	138	126	111	95	79	63	47	33	20
				-15	-6	2	11	21	32	44	56	70	84
3	11			129	122	122	100	88	74	60	47	34	22
				-14	-6	1	10	19	29	39	51	63	75
3	12			114	108	101	91	81	69	58	46	35	24
				-13	-6	1	9	17	26	35	46	57	68

Number of measurements											
13	14	15	16	17	18	19	20	21	22	23	24
-15											
163											
-6											
141											
0	-13										
124	143										
4	-7										
110	127										
8	-2										
98	113										
11	1	-8									
88	102	116									
14	4	-4									
80	93	105									

Table 3. Continued

		Number of measurements											
R	G	1	2	3	4	5	6	7	8	9	10	11	12
3	13			102	98	91	84	75	65	55	45	35	25
				-12	-6	1	8	15	23	32	42	52	62
3	14			93	89	83	77	70	61	53	44	35	26
				-11	-5	1	7	14	21	29	38	47	57
3	15			84	81	77	71	65	58	50	42	34	26
				-10	-5	0	6	13	20	27	35	44	52
3	16			77	75	71	66	61	55	48	41	34	26
				-10	-5	0	6	12	18	25	32	40	48
3	17			72	69	66	62	57	52	46	40	33	26
				-9	-5	0	5	11	17	23	30	37	45
3	18			66	65	62	58	54	49	44	38	32	26
				-9	-4	0	5	10	16	22	28	35	42
3	19			62	60	58	55	51	47	42	37	32	26
				-8	-4	0	4	9	15	20	26	33	39
3	20			58	57	54	52	48	45	40	36	31	26
				-8	-4	0	4	9	14	19	25	31	37
3	21			55	53	51	49	46	43	39	35	30	26
				-8	-4	0	4	8	13	18	23	29	35
3	22			52	50	49	46	44	41	37	33	29	25
				-7	-4	0	4	8	12	17	22	27	33
3	23			49	48	46	44	42	39	36	32	29	25
				-7	-4	0	3	7	12	16	21	26	31
3	24			46	45	44	42	40	37	35	31	28	24
				-7	-4	0	3	7	11	15	20	24	29

Number of measurements											
13	14	15	16	17	18	19	20	21	22	23	24
16	7	0	-9								
73	84	96	108								
17	9	1	-5								
67	77	88	99								
18	11	4	-2								
62	71	81	92								
19	12	5	0	-7							
57	66	75	85	95							
20	13	7	1	-4							
53	61	70	79	88							
20	14	8	3	-2							
49	57	65	73	82							
21	15	10	4	0	-6						
46	53	61	69	77	85						
21	16	10	5	0	-4						
43	50	57	64	72	80						
21	16	11	6	2	-2						
41	47	54	61	68	75						
21	16	12	7	3	0	-5					
39	45	51	57	64	71	78					
21	17	12	8	4	0	-3					
36	42	48	54	60	67	74					
21	17	13	9	5	1	-2					
35	40	46	51	57	63	70					

Table 4.4. Percentage increase in multiplies of direct over indirect methods and percentage increase in computer memory of indirect over direct methods for (value of R) signal variables

Top numbers = Percentage increase in multiplies
Bottom numbers = Percentage increases in computer memory
G = Number of noise states
xxx = Number greater than 1000%

		Number of measurements											
R	G	1	2	3	4	5	6	7	8	9	10	11	12
4	1				xxx	xxx	xxx	897	493	276	153	78	30
					-26	-4	18	45	75	109	145	184	224
4	2				xxx	xxx	xxx	810	485	293	175	100	50
					-28	-7	13	36	62	91	122	154	189
4	3				xxx	xxx	xxx	689	450	292	187	116	66
					-27	-9	9	29	51	76	102	130	159
4	4				xxx	xxx	797	574	404	280	190	126	79
					-26	-9	6	23	43	64	86	111	136
4	5				950	783	622	477	357	261	187	131	87
					-24	-10	4	19	36	54	74	95	117
4	6				701	601	498	401	315	241	181	132	93
					-22	-9	3	16	31	47	64	82	102
4	7				543	478	410	342	278	221	172	130	96
					-20	-9	2	13	26	41	56	72	89
4	8				437	393	344	295	247	202	162	127	97
					-19	-9	1	11	23	36	49	64	79
4	9				362	330	295	258	221	186	153	123	96
					-17	-8	0	10	20	31	44	57	70
4	10				307	283	257	228	199	170	143	118	94
					-16	-8	0	8	18	28	39	51	63
4	11				266	247	226	204	181	157	134	113	92
					-15	-8	0	7	16	25	35	46	57
4	12				233	219	202	184	165	145	126	107	90
					-14	-7	-1	7	14	23	32	42	52

	Number of measurements											
	13	14	15	16	17	18	19	20	21	22	23	24
	0	-29										
	266	309										
	16	-8										
	224	261										
	31	5	-15									
	190	222	254									
	44	17	-2									
	162	190	218									
	54	28	7	-9								
	140	164	189	214								
	61	36	16	0	-15							
	122	143	165	187	210							
	67	43	24	7	-6							
	107	126	145	165	186							
	71	48	30	14	0	-11						
	95	112	129	147	165	184						
	73	52	35	19	6	-4						
	85	100	115	131	148	165						
	74	55	39	24	11	0	-9					
	76	90	104	118	133	149	165					
	74	57	41	28	16	5	-4					
	69	81	94	107	121	135	149					
	73	58	44	31	19	9	0	-9				
	63	74	86	98	110	123	136	150				

Table 4. Continued

		Number of measurements											
R	G	1	2	3	4	5	6	7	8	9	10	11	12
4	13				207	195	182	167	151	135	119	102	87
					-13	-7	-1	6	13	21	29	38	48
4	14				186	176	165	153	140	126	112	98	84
					-12	-7	-1	5	12	19	27	35	44
4	15				168	161	151	141	130	118	106	93	81
					-12	-6	-1	5	11	18	25	32	40
4	16				154	147	139	130	121	111	100	89	78
					-11	-6	-1	4	10	16	23	30	37
4	17				141	136	129	121	113	104	95	85	75
					-11	-6	-1	4	9	15	21	28	35
4	18				131	126	120	113	106	98	90	81	73
					-10	-6	-1	3	9	14	20	26	32
4	19				122	117	112	106	100	93	86	78	70
					-10	-5	-1	3	8	13	19	24	30
4	20				113	110	105	100	94	88	82	75	68
					-9	-5	-1	3	7	12	17	23	29
4	21				106	103	99	94	89	84	78	72	65
					-9	-5	-1	3	7	12	16	22	27
4	22				100	97	93	89	85	80	75	69	63
					-8	-5	-1	2	7	11	15	20	25
4	23				94	92	89	85	81	76	71	66	61
					-8	-5	-1	2	6	10	15	19	24
4	24				89	87	84	81	77	73	69	64	59
					-8	-5	-1	2	6	10	14	18	23

Number of measurements											
13	14	15	16	17	18	19	20	21	22	23	24
72	58	45	33	22	12	3	-4				
57	68	78	90	101	113	125	138				
71	58	46	35	25	15	7	0	-8			
53	62	72	82	93	104	115	127	139			
69	58	47	37	27	18	10	2	-4			
49	58	67	76	86	96	107	117	128			
67	57	47	38	29	20	12	5	-1			
45	53	62	71	80	89	99	109	119			
66	56	47	38	30	22	14	7	1	-5		
42	50	58	66	74	83	92	102	111	121		
64	55	47	39	31	23	16	9	3	-2		
39	46	54	62	70	78	86	95	104	113		
62	54	47	39	32	24	18	11	5	0	-6	
37	44	50	58	65	73	81	89	97	106	115	
60	53	46	39	32	25	19	13	7	1	-3	
35	41	47	54	61	69	76	84	92	100	108	
59	52	45	39	32	26	20	14	9	3	-1	
33	39	45	51	58	65	72	79	86	94	102	
57	51	45	39	33	27	21	15	10	5	0	-4
31	36	42	48	55	61	68	74	82	89	96	104
55	50	44	38	33	27	22	17	11	6	2	-2
29	34	40	46	52	58	64	71	77	84	91	98
54	49	43	38	33	28	22	17	13	8	3	0
28	33	38	43	49	55	61	67	73	80	86	93

Table 4.5. Percentage increase in multiplies of direct over indirect methods and percentage increase in computer memory of indirect over direct methods for (value of R) signal variables

Top numbers = Percentage increase in multiplies
Bottom numbers = Percentage increases in computer memory
G = Number of noise states
xxx = Number greater than 1000%

		Number of measurements											
R	G	1	2	3	4	5	6	7	8	9	10	11	12
5	1					xxx	xxx	xxx	xxx	xxx	632	391	244
						-22	-5	13	34	59	86	115	146
5	2					xxx	xxx	xxx	xxx	990	632	410	269
						-24	-8	9	28	49	73	99	126
5	3					xxx	xxx	xxx	xxx	894	605	412	282
						-25	-9	6	23	42	62	85	108
5	4					xxx	xxx	xxx	xxx	785	561	401	286
						-24	-10	4	19	35	53	73	94
5	5					xxx	xxx	xxx	897	681	512	382	283
						-23	-10	2	15	30	46	64	82
5	6					xxx	xxx	922	745	591	462	358	275
						-22	-10	1	13	26	40	56	72
5	7					xxx	879	750	627	515	416	333	263
						-21	-10	0	11	22	35	49	64
5	8					803	715	625	536	452	376	308	250
						-19	-10	-1	9	20	31	44	57
5	9					660	596	530	464	400	340	285	237
						-18	-10	-1	8	17	28	39	51
5	10					554	507	458	407	357	309	264	223
						-17	-9	-1	6	15	25	35	46
5	11					475	439	401	361	321	282	245	211
						-16	-9	-2	6	14	22	32	42
5	12					414	385	355	323	291	259	228	199
						-15	-9	-2	5	12	20	29	38

Number of measurements											
13	14	15	16	17	18	19	20	21	22	23	24
151	89	46	15	-7							
178	212	248	284	322							
175	110	65	32	7	-12						
154	184	215	247	279	313						
192	128	82	47	21	1	-16					
133	159	186	214	243	273	303					
203	142	96	61	34	13	-3					
116	139	163	187	213	239	266					
208	151	107	72	45	23	6	-8				
101	122	143	165	187	210	234	258				
209	156	115	81	55	33	15	0	-13			
89	107	126	146	166	187	208	230	252			
206	159	120	89	63	41	23	8	-4			
79	95	112	130	148	166	186	205	225			
205	159	124	94	69	48	30	15	3	-8		
71	85	101	116	133	149	167	184	203	221		
194	157	125	98	74	54	37	22	9	-1		
64	77	91	105	120	135	151	167	183	200		
187	154	125	100	78	59	42	28	15	4	-5	
57	69	82	95	109	122	137	151	167	182	198	
179	150	124	101	81	63	47	32	20	9	0	-9
52	63	75	87	99	112	125	138	152	166	181	195
171	146	122	101	82	65	50	37	25	14	4	-4
48	58	68	79	91	102	114	127	140	153	166	180

Table 5. Continued

		Number of measurements											
R	G	1	2	3	4	5	6	7	8	9	10	11	12
5	13					365	342	318	292	265	239	213	187
						-14	-8	-2	4	11	18	26	35
5	14					326	307	287	266	244	221	199	177
						-14	-8	-2	4	10	17	24	32
5	15					294	278	261	243	225	206	186	167
						-13	-8	-2	3	9	16	22	30
5	16					267	254	240	224	208	192	175	159
						-12	-7	-2	3	8	14	21	28
5	17					244	233	221	208	194	180	165	151
						-12	-7	-2	3	8	13	19	26
5	18					225	215	205	193	181	169	156	143
						-11	-7	-2	2	7	12	18	24
5	19					208	200	191	181	170	159	148	136
						-11	-7	-2	2	7	12	17	22
5	20					194	186	178	170	160	150	140	130
						-10	-6	-2	2	6	11	16	21
5	21					181	174	167	160	151	143	134	124
						-10	-6	-2	2	6	10	15	20
5	22					170	164	157	151	143	135	127	119
						-9	-6	-2	1	5	10	14	19
5	23					160	154	149	143	136	129	122	114
						-9	-6	-2	1	5	9	13	18
5	24					151	146	141	135	129	123	116	109
						-9	-6	-2	1	5	9	13	17

Number of measurements											
13	14	15	16	17	18	19	20	21	22	23	24
163	141	120	101	83	68	53	40	28	18	8	0
44	53	63	73	83	94	105	117	129	141	153	166
156	136	117	100	84	69	55	43	32	22	12	4
40	49	58	67	77	87	97	108	119	130	142	153
149	131	114	99	84	70	57	45	35	25	16	8
37	45	54	62	72	81	90	100	111	121	132	142
142	127	111	97	83	70	59	47	37	28	19	11
35	42	50	58	67	75	84	93	103	113	123	133
136	122	108	95	82	71	59	49	39	30	22	14
32	39	47	54	62	70	79	87	96	105	115	124
130	118	105	93	82	70	60	50	41	32	24	16
30	37	44	51	58	66	74	82	90	99	107	116
125	113	102	91	80	70	60	51	42	34	26	19
28	35	41	48	55	62	69	77	85	93	101	109
120	109	99	89	79	70	60	52	43	35	28	21
27	33	39	45	51	58	65	72	80	87	95	103
115	105	96	87	78	69	60	52	44	37	29	23
25	31	36	42	49	55	62	68	75	82	90	97
110	102	93	85	76	68	60	52	45	38	31	24
24	29	34	40	46	52	58	65	71	78	85	92
106	98	90	83	75	67	60	52	45	38	32	26
23	27	33	38	44	49	55	61	68	74	81	87
102	95	88	81	73	66	59	52	46	39	33	27
21	26	31	36	41	47	52	58	64	70	77	83

Table 4.6. Percentage increase in multiplies of direct over indirect methods and percentage increase in computer memory of indirect over direct methods for (value of R) signal variables

Top numbers = Percentage increase in multiplies
Bottom numbers = Percentage increases in computer memory
G = Number of noise states
xxx = Number greater than 1000%

		Number of measurements											
R	G	1	2	3	4	5	6	7	8	9	10	11	12
6	1						xxx	xxx	xxx	xxx	xxx	xxx	772
							-19	-5	10	28	47	70	94
6	2						xxx	xxx	xxx	xxx	xxx	xxx	779
							-22	-7	7	23	41	60	82
6	3						xxx	xxx	xxx	xxx	xxx	xxx	760
							-23	-9	4	18	35	52	71
6	4						xxx	xxx	xxx	xxx	xxx	988	722
							-23	-10	2	15	30	45	63
6	5						xxx	xxx	xxx	xxx	xxx	886	673
							-22	-11	1	12	25	40	55
6	6						xxx	xxx	xxx	xxx	xxx	790	621
							-22	-11	-1	10	22	35	49
6	7						xxx	xxx	xxx	xxx	863	703	569
							-21	-11	-1	8	19	31	43
6	8						xxx	xxx	xxx	887	751	628	521
							-20	-11	-2	7	17	27	39
6	9						xxx	983	872	763	659	563	477
							-19	-11	-2	6	15	24	35
6	10						912	830	747	664	584	508	438
							-18	-10	-3	5	13	22	31
6	11						777	714	650	585	521	460	403
							-17	-10	-3	4	12	20	29
6	12						672	623	572	521	469	419	372
							-16	-10	-3	3	10	18	26

Number of measurements											
13	14	15	16	17	18	19	20	21	22	23	24
513	346	234	157	102	62	32	9	-8			
119	146	174	204	234	266	298	331	365			
534	371	260	180	123	81	48	24	4	-12		
105	129	154	180	207	235	264	293	323	354		
540	387	278	200	142	98	64	38	17	0	-15	
92	113	136	159	183	208	234	260	287	314	342	
531	392	290	215	157	113	78	51	29	11	-3	
81	100	120	141	163	185	208	232	256	280	305	
512	389	296	225	169	125	91	63	40	21	6	-6
71	89	107	125	145	165	186	207	228	251	273	296
486	380	297	230	178	135	101	73	50	31	15	2
63	79	95	112	130	148	166	186	205	225	246	266
458	367	293	233	183	143	110	82	59	40	24	10
57	71	85	101	117	133	150	167	185	203	222	241
429	351	286	232	186	148	117	90	67	48	32	18
51	64	77	91	105	120	136	152	168	184	201	219
401	335	278	229	187	152	122	96	74	55	39	24
46	58	70	82	96	109	123	138	153	168	183	199
374	318	268	225	187	154	125	101	80	61	45	31
42	52	63	75	87	100	113	126	140	154	168	183
349	301	258	219	185	154	128	105	84	66	50	36
38	48	58	69	80	91	103	116	128	141	154	168
327	285	247	213	182	154	129	107	88	71	55	41
35	44	53	63	74	84	95	107	118	130	142	155

Table 6. Continued

		Number of measurements											
R	G	1	2	3	4	5	6	7	8	9	10	11	12
6	13						590	550	510	468	426	384	344
							-15	-9	-3	3	9	16	24
6	14						524	491	458	424	389	354	320
							-15	-9	-3	2	8	15	22
6	15						469	443	415	386	357	327	298
							-14	-9	-3	2	8	14	20
6	16						424	402	379	354	329	304	279
							-13	-8	-3	2	7	13	19
6	17						387	368	348	327	306	284	262
							-13	-8	-3	1	6	12	18
6	18						355	338	321	303	285	266	246
							-12	-8	-3	1	6	11	16
6	19						327	313	298	282	266	249	233
							-12	-8	-3	1	5	10	15
6	20						303	291	278	264	250	235	220
							-11	-7	-3	1	5	10	14
6	21						282	271	260	248	235	222	209
							-11	-7	-3	1	5	9	14
6	22						264	254	244	233	222	210	198
							-10	-7	-3	0	4	8	13
6	23						248	239	230	220	210	200	189
							-10	-7	-3	0	4	8	12
6	24						233	226	217	209	200	190	180
							-10	-6	-3	0	4	8	11

Number of measurements											
13	14	15	16	17	18	19	20	21	22	23	24
306	270	237	206	178	153	130	109	91	74	59	46
32	40	49	58	68	78	88	99	109	120	132	143
287	256	226	199	174	151	130	110	93	77	63	50
29	37	45	54	63	72	82	91	101	112	122	133
270	243	217	192	169	148	129	111	94	79	66	53
27	35	42	50	59	67	76	85	95	104	114	124
254	230	207	186	165	146	127	111	95	81	68	56
25	32	39	47	55	63	71	79	88	97	107	116
240	219	199	179	160	142	126	110	96	82	70	58
24	30	37	44	51	59	66	74	83	91	100	109
227	209	190	173	156	139	124	109	96	83	71	60
22	28	34	41	48	55	62	70	78	86	94	102
216	199	182	166	151	136	122	108	96	84	72	62
21	26	32	39	45	52	59	66	73	81	88	96
205	190	175	161	146	133	120	107	95	84	73	63
20	25	31	36	42	49	55	62	69	76	83	91
195	182	168	155	142	130	117	106	95	84	74	64
18	24	29	34	40	46	52	59	65	72	79	86
186	174	162	150	138	126	115	104	94	84	74	65
17	22	27	33	38	44	50	56	62	68	75	81
178	167	156	145	134	123	113	102	93	83	74	66
16	21	26	31	36	41	47	53	59	65	71	77
170	160	150	140	130	120	110	101	92	83	74	66
16	20	25	29	34	39	45	50	56	62	67	74

Table 4.7. Percentage increase in multiplies of direct over indirect methods and percentage increase in computer memory of indirect over direct methods for (value of R) signal variables

Top numbers = Percentage increase in multiplies
Bottom numbers = Percentage increases in computer memory
C = Number of noise states
xxx = Number greater than 1000%

		Number of measurements											
R	G	1	2	3	4	5	6	7	8	9	10	11	12
7	1							xxx	xxx	xxx	xxx	xxx	xxx
								-17	-5	8	23	39	58
7	2							xxx	xxx	xxx	xxx	xxx	xxx
								-19	-7	5	19	34	51
7	3							xxx	xxx	xxx	xxx	xxx	xxx
								-21	-9	3	15	29	45
7	4							xxx	xxx	xxx	xxx	xxx	xxx
								-21	-10	1	12	25	39
7	5							xxx	xxx	xxx	xxx	xxx	xxx
								-21	-11	0	10	22	34
7	6							xxx	xxx	xxx	xxx	xxx	xxx
								-21	-11	-2	8	19	30
7	7							xxx	xxx	xxx	xxx	xxx	xxx
								-20	-11	-2	7	16	27
7	8							xxx	xxx	xxx	xxx	xxx	xxx
								-20	-11	-3	5	14	24
7	9							xxx	xxx	xxx	xxx	xxx	871
								-19	-11	-3	4	13	22
7	10							xxx	xxx	xxx	xxx	895	780
								-18	-11	-4	3	11	19
7	11							xxx	xxx	991	892	795	703
								-17	-11	-4	3	10	18
7	12							xxx	947	868	790	712	637
								-17	-10	-4	2	9	16

Number of measurements											
13	14	15	16	17	18	19	20	21	22	23	24
xxx	915	640	454	326	234	167	117	79	50	27	8
78	100	122	146	172	197	224	252	280	309	339	269
xxx	928	664	482	353	260	191	138	98	66	41	21
69	89	109	131	154	177	201	226	252	278	305	332
xxx	915	672	499	374	281	211	157	115	82	55	34
61	79	98	117	138	159	181	203	226	250	274	299
xxx	883	666	506	387	297	227	173	130	96	69	46
54	70	87	105	123	143	162	183	204	225	247	269
xxx	838	648	504	393	307	240	187	144	109	81	58
48	63	78	94	111	128	146	165	184	203	223	244
992	785	623	495	394	314	249	198	155	121	92	68
43	56	70	85	100	116	132	149	167	184	203	221
900	731	593	481	390	316	255	206	165	131	102	78
38	51	63	77	91	105	120	136	151	168	184	201
816	678	561	464	382	315	259	212	172	139	111	87
35	46	57	70	82	96	109	124	138	153	168	184
741	628	529	444	372	311	259	215	178	145	118	94
31	41	52	64	75	87	100	113	127	140	154	169
675	581	498	424	361	305	258	217	181	151	124	101
28	38	48	58	69	80	92	104	116	129	142	155
617	539	468	404	348	298	255	217	184	154	129	107
26	35	44	53	63	74	85	96	107	119	131	143
566	500	440	385	335	290	251	216	185	157	133	111
24	32	40	49	59	68	78	89	99	110	121	133

Table 7. Continued

		Number of measurements											
R	G	1	2	3	4	5	6	7	8	9	10	11	12
7	13							895	833	770	706	643	581
								-16	-10	-4	2	8	15
7	14							791	740	689	636	584	532
								-15	-10	-4	1	7	13
7	15							706	664	621	578	534	490
								-15	-10	-4	1	6	12
7	16							636	601	565	528	491	454
								-14	-9	-4	1	6	11
7	17							577	547	517	485	454	422
								-14	-9	-4	0	5	10
7	18							527	502	476	449	421	394
								-13	-9	-4	0	5	10
7	19							485	463	440	417	393	368
								-13	-8	-4	0	4	9
7	20							448	429	409	388	367	346
								-12	-8	-4	0	4	8
7	21							416	399	382	364	345	326
								-12	-8	-4	0	4	8
7	22							388	373	358	342	325	308
								-11	-8	-4	0	3	7
7	23							363	350	336	322	307	292
								-11	-7	-4	0	3	7
7	24							341	329	317	304	291	277
								-11	-7	-4	-1	3	7

Number of measurements											
13	14	15	16	17	18	19	20	21	22	23	24
522	466	414	366	322	282	246	213	185	159	135	115
22	29	37	46	54	63	73	82	92	102	113	123
482	435	390	348	309	273	240	211	184	159	137	118
20	27	34	42	50	59	68	77	86	95	105	115
448	407	368	331	296	264	234	207	182	159	139	120
19	25	32	39	47	55	63	71	80	89	98	108
417	382	347	315	284	255	228	203	180	159	139	121
17	23	30	37	44	51	59	67	75	83	92	101
390	359	329	300	272	246	222	199	177	158	139	122
16	22	28	34	41	48	55	63	70	78	86	95
366	339	312	286	261	238	215	194	175	156	139	123
15	20	26	32	39	45	52	59	66	74	81	89
344	320	296	273	251	230	209	190	171	154	138	123
14	19	25	30	36	43	49	56	62	70	77	84
325	303	282	261	241	222	203	185	168	152	137	123
13	18	23	29	34	40	46	53	59	66	73	80
307	288	269	250	232	214	197	181	165	150	136	122
12	17	22	27	32	38	44	50	56	62	69	75
291	274	257	240	223	207	191	176	161	147	134	121
12	16	21	26	31	36	42	47	53	59	65	72
277	261	246	230	215	200	186	172	158	145	132	120
11	15	20	24	29	34	39	45	50	56	62	68
263	249	235	221	207	194	180	167	155	142	131	119
10	14	19	23	28	33	38	43	48	53	59	65

Table 4.8. Percentage increase in multiplies of direct over indirect methods and percentage increase in computer memory of indirect over direct methods for (value of R) signal variables

Top numbers = Percentage increase in multiplies
Bottom numbers = Percentage increases in computer memory
G = Number of noise states
xxx = Number greater than 1000%

		Number of measurements											
R	G	1	2	3	4	5	6	7	8	9	10	11	12
8	1								xxx -15	xxx -5	xxx 6	xxx 19	xxx 33
8	2								xxx -17	xxx -7	xxx 4	xxx 16	xxx 29
8	3								xxx -19	xxx -9	xxx 2	xxx 13	xxx 25
8	4								xxx -20	xxx -10	xxx 0	xxx 10	xxx 22
8	5								xxx -20	xxx -11	xxx -1	xxx 8	xxx 19
8	6								xxx -20	xxx -11	xxx -2	xxx 7	xxx 16
8	7								xxx -20	xxx -11	xxx -3	xxx 5	xxx 14
8	8								xxx -19	xxx -12	xxx -4	xxx 4	xxx 12
8	9								xxx -19	xxx -11	xxx -4	xxx 3	xxx 11
8	10								xxx -18	xxx -11	xxx -4	xxx 2	xxx 10
8	11								xxx -18	xxx -11	xxx -5	xxx 2	xxx 8
8	12								xxx -17	xxx -11	xxx -5	xxx 1	xxx 7

Number of measurements											
13	14	15	16	17	18	19	20	21	22	23	24
xxx	xxx	xxx	xxx	770	569	424	319	240	180	133	97
49	66	85	105	125	147	169	192	216	241	266	292
xxx	xxx	xxx	xxx	796	598	453	346	265	202	154	115
44	59	76	94	113	133	153	175	196	219	242	265
xxx	xxx	xxx	xxx	807	616	475	368	287	223	172	132
39	53	69	85	102	120	139	158	178	199	220	241
xxx	xxx	xxx	xxx	804	625	490	385	304	240	189	148
34	47	62	77	93	109	126	144	162	181	200	220
xxx	xxx	xxx	xxx	788	624	497	397	318	255	204	162
30	42	56	69	84	99	115	131	148	165	182	200
xxx	xxx	xxx	952	764	616	498	404	328	267	216	175
27	38	50	63	76	90	104	119	135	150	167	183
xxx	xxx	xxx	897	734	602	494	407	335	276	226	185
24	34	45	57	69	82	95	109	123	138	153	168
xxx	xxx	xxx	841	701	583	486	405	338	282	234	194
21	31	41	52	63	75	87	100	113	126	140	154
xxx	xxx	929	787	666	562	475	401	338	285	240	201
19	28	38	48	58	69	80	92	104	116	129	142
xxx	990	854	735	631	540	462	395	337	287	244	206
17	26	34	44	53	63	74	85	96	108	119	132
xxx	900	787	686	596	517	447	386	333	286	246	210
16	23	32	40	49	59	68	78	89	100	111	122
922	821	727	641	564	494	432	377	328	285	246	213
14	21	29	37	46	54	63	73	83	93	103	113

Table 8. Continued

		Number of measurements											
R	G	1	2	3	4	5	6	7	8	9	10	11	12
8	13								xxx -16	xxx -11	xxx -5	xxx 1	925 7
8	14								xxx -16	xxx -10	989 -5	913 0	837 6
8	15								xxx -15	951 -10	889 -5	826 0	763 5
8	16								909 -15	857 -10	805 -5	752 0	699 5
8	17								822 -14	778 -10	734 -5	689 0	644 4
8	18								749 -14	712 -9	673 -5	635 -1	596 4
8	19								686 -13	654 -9	621 -5	588 -1	554 3
8	20								632 -13	604 -9	576 -5	546 -1	517 3
8	21								586 -12	561 -9	536 -5	510 -1	484 3
8	22								545 -12	523 -8	501 -5	478 -1	455 2
8	23								509 -12	490 -8	470 -5	449 -1	429 2
8	24								477 -11	460 -8	442 -5	424 -1	405 2

Number of measurements											
13	14	15	16	17	18	19	20	21	22	23	24
837	753	674	600	533	472	416	366	322	282	246	214
13	20	27	34	42	50	59	68	77	86	96	106
764	693	626	563	504	450	401	355	315	278	244	215
12	18	25	32	39	47	55	63	72	81	90	99
701	641	583	529	477	429	385	344	307	273	242	214
11	17	23	30	37	44	51	59	67	75	84	93
646	595	545	498	452	410	370	333	299	268	239	213
10	16	22	28	34	41	48	56	63	71	79	87
599	554	511	469	429	392	356	322	291	262	236	211
9	15	20	26	32	39	45	52	59	67	74	82
557	518	480	444	408	374	342	312	283	257	232	209
9	14	19	24	30	36	43	49	56	63	70	77
520	486	453	420	388	358	329	301	275	251	228	206
8	13	18	23	29	34	40	46	53	59	66	73
487	457	428	399	370	343	317	291	268	245	223	203
7	12	17	22	27	32	38	44	50	56	63	69
458	431	405	379	353	329	305	282	260	239	219	200
7	11	16	21	25	31	36	42	47	53	59	66
431	408	384	361	338	315	294	273	252	233	214	197
6	11	15	19	24	29	34	40	45	51	56	62
408	386	365	344	323	303	283	264	245	227	210	193
6	10	14	18	23	28	33	38	43	48	54	59
386	367	348	329	310	291	273	255	238	221	205	190
6	9	13	18	22	26	31	36	41	46	51	57

V. EXAMPLES

Both examples demonstrate the use of the direct complementary filter to derive a distortionless estimate of the signal.

A. Example I

For the first example, consider the following measurement equations.

$$y_1(t_k) = S_1(t_k) + n_1(t_k) \tag{5.1}$$

$$y_2(t_k) = S_2(t_k) + n_2(t_k) \tag{5.2}$$

$$y_3(t_k) = \gamma_1 S_1(t_k) + \gamma_2 S_2(t_k) + n_3(t_k) \tag{5.3}$$

Assume that n_1, n_2, and n_3 are uncorrelated "white" measurement noises with variances v_{11}, v_{22}, and v_{33} respectively. The state vector has only two states. Let x_1 denote S_1 and x_2 denote S_2 where the time notation has been dropped. Then the measurement equation is

$$y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} \tag{5.4}$$

The covariance matrix will be a two by two and is written as

$$P^* = \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} \tag{5.5}$$

Even though this example has no P_N^* or $\hat{x}_N$, the theory of the direct filter will apply equally well. There is no reason to find a transistion matrix or an H matrix because they are not used in this example.

The steps of the algorithm will now be described in detail.

$$P^{*} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{5.6}$$

$$\hat{x}' = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{5.7}$$

$$R_o = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{5.8}$$

$$R_o M_1^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{5.9}$$

$$M_1 R_o M_1^T = 1 \tag{5.10}$$

$$b_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{5.11}$$

$$\hat{x}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} y_1 = \begin{bmatrix} y_1 \\ 0 \end{bmatrix} \tag{5.12}$$

$$b_1 M_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \tag{5.13}$$

$$I - b_1 M_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{5.14}$$

$$P_1 = \begin{bmatrix} v_{11} & 0 \\ 0 & 0 \end{bmatrix} \tag{5.15}$$

$$R_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{5.16}$$

Since R_1 is not zero, the following calculations are made:

$$R_1M_2^T = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{5.17}$$

$$M_2R_1M_2^T = 1 \tag{5.18}$$

The gain matrix is

$$b_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{5.19}$$

The update of $\hat{x}_1$ is

$$\hat{x}_2 = \begin{bmatrix} y_1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad y_2 \quad = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \tag{5.20}$$

Next, calculate

$$b_2M_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{5.21}$$

$$(I - b_2M_2) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \tag{5.22}$$

$$P_2 = \begin{bmatrix} v_{11} & 0 \\ 0 & v_{22} \end{bmatrix} \tag{5.23}$$

$$R_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{5.24}$$

The next input is ready for processing. The gain matrix now is

$$b_3 = \frac{\begin{bmatrix} v_{11} & 0 \\ 0 & v_{22} \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix}}{[\gamma_1\ \gamma_2] \begin{bmatrix} v_{11} & 0 \\ 0 & v_{22} \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} + v_{33}} \tag{5.25}$$

and reduces to

$$b_3 = \frac{\begin{bmatrix} \gamma_1 v_{11} \\ \gamma_2 v_{22} \end{bmatrix}}{(\gamma_1^2 v_{11} + \gamma_2^2 v_{22} + v_{33})} \tag{5.26}$$

The update of $\hat{x}_3$ is

$$\hat{x}_3 = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \left[\frac{1}{\gamma_1^2 v_{11} + \gamma_2^2 v_{22} + v_{33}}\right] \begin{bmatrix} \gamma_1 v_{11} \\ \gamma_2 v_{22} \end{bmatrix} (y_3 - \gamma_1 y_1 - \gamma_2 y_2) \tag{5.27}$$

or

$$\hat{x}_3 = \begin{bmatrix} y_1 + \dfrac{\gamma_1 v_{11}(y_3 - \gamma_1 y_1 - \gamma_2 y_2)}{\gamma_1^2 v_{11} + \gamma_2^2 v_{22} + v_{33}} \\ \\ y_2 + \dfrac{\gamma_2 v_{22}(y_3 - \gamma_1 y_1 - \gamma_2 y_2)}{\gamma_1^2 v_{11} + \gamma_2^2 v_{22} + v_{33}} \end{bmatrix} \tag{5.28}$$

Next consider solving the same problem by the indirect method. One method of implementing this is shown in Figure 5.1.

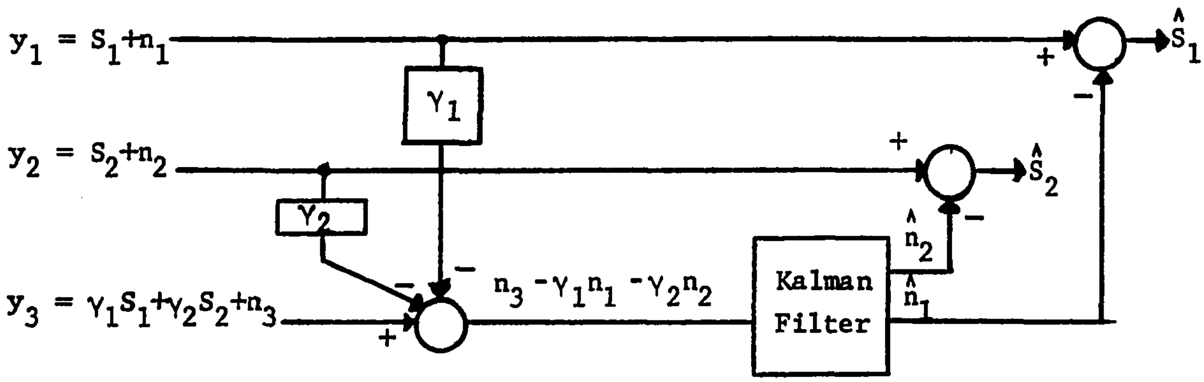


Figure 5.1. Block diagram of indirect filter.

The assumption was that n_1, n_2, and n_3 are white noises and are uncorrelated, then the Kalman filter in this case will be trivial. From Brown and Nilsson (6), the optimal transfer functions for estimating $\gamma_2\hat{n}_2$ and $\gamma_1\hat{n}_1$ are

$$\gamma_2\hat{n}_2 = \frac{\gamma_2 G_2(\omega)\, y_{IN}}{\gamma_1 G_1(\omega) + \gamma_2 G_2(\omega) + G_3(\omega)} \tag{5.29}$$

$$\gamma_1\hat{n}_1 = \frac{\gamma_1 G_1(\omega)\, y_{IN}}{\gamma_1 G_1(\omega) + \gamma_2 G_2(\omega)\; G_3(\omega)} \tag{5.30}$$

where $G_i(\omega)$ is the power spectral density function for $n_i(t)$ and $y_{IN} = (y_3 - \gamma_1 y_1 - \gamma_2 y_2)$. Also from Brown and Nilsson (6) if $n_i(t)$ is white noise then $\gamma_i n_i$ is white noise and $\gamma_i G_i(\omega) = \gamma_i^2 v_{ii}$.

Therefore, the optimal estimate of n_1 and n_2 becomes

$$\hat{n}_1 = \frac{-v_n\gamma_1(y_3 - \gamma_1 y_1 - \gamma_2 y_2)}{v_{33} + \gamma_1^2 v_{11} + \gamma_2^2 v_{22}} \tag{5.31}$$

$$\hat{n}_2 = \frac{-v_{22}\gamma_2(y_3 - \gamma_1 y_1 - \gamma_2 y_2)}{v_{33} + \gamma_1^2 v_{11} + \gamma_2^2 v_{22}} \tag{5.32}$$

The best estimate of the signal is

$$\hat{S}_1 = y_1 - \hat{n}_1 = y_1 + \frac{v_{11}\gamma_1(y_3 - \gamma_1 y_1 - \gamma_2 y_2)}{\gamma_1^2 v_{11} + \gamma_2^2 v_{22} + v_{33}} \tag{5.33}$$

and

$$\hat{S}_2 = y_2 - \hat{n}_2 = y_2 + \frac{v_{22}\gamma_2(y_3 - \gamma_1 y_1 - \gamma_2 y_2)}{\gamma_1^2 v_{11} + \gamma_2^2 v_{22} + v_{33}} \tag{5.34}$$

Note, these estimates of the signal are identical to those obtained in the direct filter. The purpose of this example was to demonstrate the algorithm for the direct filter and to reassure the reader that the results are identical to the indirect filter. It also indicates that the computation time in the direct approach is longer for this example:

$$P\% = 6818\% \tag{5.35}$$

B. Example II

For the second example, consider the case of altitude determination with the distortionless constraint. Assuming there are three measurements for altitude determination which are as follows:

1. Altitude derived from accelerometer measurements corrupted by white noise.

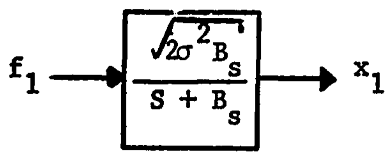
2. Altitude from barometer. (Which is a measure of true altitude

corrupted by Markov noise.)

3. Altitude from radar altimeter. (Which is a true measure of altitude corrupted by white noise.)

Let the signal state be altitude and let it be a Markov process. It has been shown that it can be chosen as any process that is desired, because it doesn't enter into the estimation of the signal.

The first step is to determine a model for this system and hence, the state equations. The Kalman filter requires that all inputs to the filter must be white noise processes, therefore, consider the following model for the altitude input.

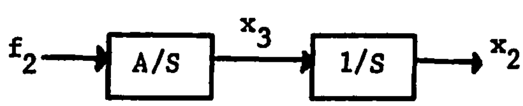


where f_1 is unity white noise and x_1 is the altitude variable.

The differential equation that describes the above system is

$$\dot{x}_1 = - B_S x_1 + \sqrt{2\sigma_S^2 B_S}\, f_1 \tag{5.36}$$

Assume the noise associated with the accelerometer measurement will be doubly integrated white noise, shown in the following block diagram.



It will take two states to describe this system and the differential equations are

$$\dot{x}_2 = x_3 \tag{5.37}$$

$$\dot{x}_3 = Af_2 \tag{5.38}$$

Assume the noise associated with the barometer will be a Markov process and will satisfy the following differential equation

$$\dot{x}_4 = -B_N x_4 + \sqrt{2\sigma_N^2 B_N}\, f_3 \tag{5.39}$$

Assume the noise associated with the radar altimeters is white so it requires no state.

Thus the state equation will have 4 states and can be written as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} -B_S & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -B_N \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} \sqrt{2\sigma_S^2 B_S}\, f_1 \\ 0 \\ A\, f_2 \\ \sqrt{2\sigma_N^2 B_N}\, f_3 \end{bmatrix} \tag{5.40}$$

The state transistion matrix is given by

$$\phi(t) = L^{-1}[SI-A]^{-1} = L^{-1} \begin{bmatrix} S+B_S & 0 & 0 & 0 \\ 0 & S & -1 & 0 \\ 0 & 0 & S & 0 \\ 0 & 0 & 0 & S+B_N \end{bmatrix}^{-1} \tag{5.41}$$

$\phi(t)$ becomes,

$$\phi(t) = \begin{bmatrix} e^{-B_S t} & 0 & 0 & 0 \\ 0 & 1 & t & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{-B_N t} \end{bmatrix} \tag{5.42}$$

H_n is (including only first order terms in Δt)

$$H_N = \begin{bmatrix} 2\sigma_S^2 B_S & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & A & 0 \\ 0 & 0 & 0 & 2\sigma_N^2 B_N \end{bmatrix} \Delta t \tag{5.43}$$

The measurement matrix will be

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} \delta y_1 \\ \delta y_2 \\ \delta y_3 \end{bmatrix} \tag{5.44}$$

Assume that δy_1, δy_2, δy_3 are white noise sequences and are uncorrelated such that

$$E[\delta y \; \delta y^T] = \begin{bmatrix} v_{11} & 0 & 0 \\ 0 & v_{22} & 0 \\ 0 & 0 & v_{33} \end{bmatrix} \tag{5.45}$$

The a posteriori covariance matrix will be a four by four matrix and in partitioned form is

$$P^* = \left[\begin{array}{c|c} P_S & P_3 \\ \hline P_S^T & P_N \end{array}\right] = \left[\begin{array}{c|ccc} P_{11} & P_{12} & P_{13} & P_{14} \\ \hline P_{12} & P_{22} & P_{23} & P_{24} \\ P_{13} & P_{23} & P_{33} & P_{34} \\ P_{14} & P_{24} & P_{34} & P_{44} \end{array}\right] \tag{5.46}$$

The a priori covariance matrix is given by the following:

$$P^* = \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & \phi P_N \phi^T \end{array}\right] + \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & H_N \end{array}\right] \tag{5.47}$$

The extrapolated state matrix is

$$\hat{x}' = \left[\begin{array}{c} 0 \\ \hline \phi x_N \end{array}\right] \tag{5.48}$$

Look at the estimate of the signal and noise states at time t_k. For simplicity assume that[1]

$$P_k^* = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix} \tag{5.49}$$

and

$$\hat{x}_k' = \begin{bmatrix} 0 \\ \hat{x}_2' \\ \hat{x}_3' \\ \hat{x}_4' \end{bmatrix} \tag{5.50}$$

Also assume that the variance of the measurement noises are

$$v_{11} = 1 , \quad v_{22} = 2 , \quad v_{33} = 3 . \tag{5.51}$$

---

[1]Note P_k^* does not represent a physically possible covariance matrix in this numerical example. It was chosen for numerical convenience and does not need to be positive definite in order to compare the direct and indirect algorithms.

The first step is to calculate the gain matrix according to

$$b_1 = \begin{bmatrix} \dfrac{I\,M_{1S}^T}{M_{1S}\;I\;M_{1S}^T} \\ \text{----------} \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.52}$$

Update the states by

$$\hat{x}_1 = \hat{x}_1' + b_1(y_1 - M_1\hat{x}_1') = \begin{bmatrix} y_1 - \hat{x}_2' \\ \hat{x}_2' \\ \hat{x}_3' \\ \hat{x}_4' \end{bmatrix} \tag{5.53}$$

Calculate the a posteriori covariance matrix by first finding

$$(I - b_1 M_1) = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.54}$$

Then

$$P_1 = (I_1 - b_1 M_1) P_1^* (I - b_1 M_1)^T + b_1 v_{11} b_1^T$$

$$= \begin{bmatrix} 2 & -1 & -2 & -1 \\ -1 & 1 & 2 & 1 \\ -2 & 2 & 1 & 1 \\ -1 & 1 & 1 & 2 \end{bmatrix} \tag{5.55}$$

Now

$$(I-b_{1S}M_{1S})\ I\ (I-b_{1S}M_{1S})^T = 0$$

Thus the remaining inputs will be processed as in steps 8 through 15 in Section B of Chapter 2.

The gain equation for the second measurement is

$$b_2 = \frac{P\ M_2^T}{(M_2P_1M_2^T + v_{22})} = \begin{bmatrix} 1/4 \\ 0 \\ -1/4 \\ 1/4 \end{bmatrix} \tag{5.56}$$

Update the states by

$$\hat{x}_2 = \hat{x}_1 + b_2(y_2 - M_2\hat{x}_1)$$

$$= \begin{bmatrix} \frac{3}{4}y_1 + \frac{1}{4}y_2 - \frac{1}{4}\hat{x}'_4 - \frac{3}{4}\hat{x}'_2 \\ \hat{x}'_2 \\ \hat{x}'_3 - \frac{1}{4}y_2 + \frac{1}{4}y_1 + \frac{\hat{x}'_4}{4} - \frac{1}{4}\hat{x}'_2 \\ \frac{3}{4}\hat{x}'_4 + \frac{1}{4}y_2 - \frac{1}{4}y_1 + \frac{1}{4}\hat{x}'_2 \end{bmatrix} \tag{5.57}$$

The update of the covariance matrix is given by

$$P_2 = P_1 - b_2(M_2P_1M_2^T + v_{22})b_2^T$$

$$= \begin{bmatrix} \frac{7}{4} & -1 & -\frac{7}{4} & -\frac{5}{4} \\ -1 & 1 & 2 & 1 \\ -\frac{7}{4} & 2 & \frac{3}{4} & \frac{5}{4} \\ -\frac{5}{4} & 1 & \frac{5}{4} & \frac{7}{4} \end{bmatrix} \tag{5.58}$$

The next measurement can be processed and the gain matrix is

$$b_3 = \frac{\begin{bmatrix} \frac{7}{4} \\ -1 \\ -\frac{7}{4} \\ -\frac{5}{4} \end{bmatrix}}{\frac{7}{4} + 3} = \begin{bmatrix} \frac{7}{19} \\ -\frac{4}{19} \\ -\frac{7}{19} \\ -\frac{5}{19} \end{bmatrix} \tag{5.59}$$

The update of the states is

$$\hat{x}_3 = \hat{x}_2 + \begin{bmatrix} \frac{7}{19} \\ -\frac{4}{19} \\ -\frac{7}{19} \\ -\frac{5}{19} \end{bmatrix} [y_3 - \frac{3}{4} y_1 - \frac{1}{4} y_2 + \frac{1}{4} \hat{x}'_4 + \frac{3}{4} \hat{x}'_2]$$

$$= \begin{bmatrix} \frac{9}{19} y_1 + \frac{3}{19} y_2 + \frac{7}{19} y_3 - \frac{3}{19} \hat{x}'_2 - \frac{3}{19} \hat{x}'_4 \\ \frac{16}{19} \hat{x}'_2 + \frac{3}{19} y_1 + \frac{1}{19} y_2 - \frac{4}{19} y_3 - \frac{1}{19} \hat{x}'_4 \\ \hat{x}'_3 + \frac{10}{19} y_1 - \frac{3}{19} y_2 - \frac{7}{19} y_3 - \frac{10}{19} \hat{x}'_2 + \frac{3}{19} \hat{x}'_4 \\ \frac{13}{19} \hat{x}'_4 - \frac{1}{19} y_1 + \frac{6}{19} y_2 - \frac{5}{19} y_3 + \frac{1}{19} \hat{x}'_2 \end{bmatrix} \tag{5.60}$$

The new covariance matrix is

$$P_3 = \begin{bmatrix} \frac{21}{19} & -\frac{15}{19} & -\frac{21}{19} & -\frac{15}{19} \\ -\frac{12}{19} & \frac{15}{19} & \frac{31}{19} & \frac{14}{19} \\ -\frac{21}{19} & \frac{31}{19} & -\frac{2}{19} & \frac{15}{38} \\ -\frac{15}{19} & \frac{14}{19} & \frac{15}{38} & \frac{27}{19} \end{bmatrix} \qquad (5.61)$$

Next work the same problem from the indirect approach where $n_1(t)$ is doubly integrated Markov noise and $n_2(t)$ is Markov and n_3 is white noise. The implementation is shown in Figure 5.2.

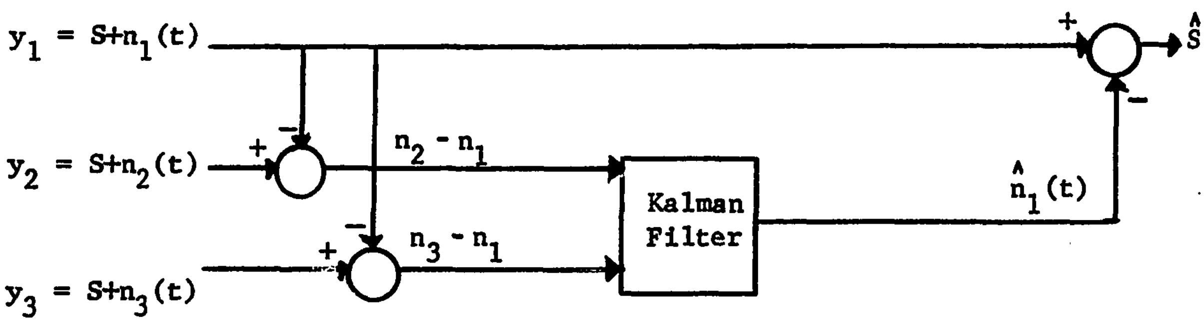


Figure 5.2. Block diagram of indirect filter.

Two states will be needed to describe $n_1(t)$.

$$\dot{x}_2 = x_3 \qquad (5.62)$$

$$\dot{x}_3 = Af_2 \qquad (5.63)$$

One state is needed to describe $n_2(t)$ and is

$$\dot{x}_4 = -B_N x_4 + \sqrt{2\sigma_N^2 B_N} f_3 \qquad (5.64)$$

No state will be required of $n_3(t)$.

The state equations can be written as

$$\begin{bmatrix} \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -B_N \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ A f_2 \\ \sqrt{2\sigma_N^2 B_N}\, f_3 \end{bmatrix} \tag{5.65}$$

The state transition matrix is

$$\phi(t) = \begin{bmatrix} \frac{1}{S^2} & \frac{1}{S^2} & 0 \\ 0 & \frac{1}{S} & 0 \\ 0 & 0 & \frac{1}{S+B_N} \end{bmatrix} \tag{5.66}$$

and (including only first order terms in Δt)

$$H_N = \begin{bmatrix} 0 & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & 2\sigma_N^2 P_N \end{bmatrix} \Delta t \tag{5.67}$$

The measurement matrix to the input of the Kalman filter will be

$$\begin{bmatrix} y_1' \\ y_2' \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} \delta y_2 - \delta y_1 \\ n_3 - \delta y_1 \end{bmatrix} \tag{5.68}$$

Then to be consistent the covariance matrix will be

$$P^* = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 1 \\ 1 & 1 & 2 \end{bmatrix} \tag{5.69}$$

Note that in the measurement matrix the signals will not be uncorrelated. Then

$$V = E\left[\begin{bmatrix}\delta y_2 - \delta y_1 \\ \hline n_3 - \delta y_1\end{bmatrix} \left| \delta y_2 - \delta y_1 \quad n_3 - \delta y_1 \right|\right]$$

$$= \begin{bmatrix} 3 & 1 \\ 1 & 4 \end{bmatrix} \tag{5.70}$$

Thus the measurements must all be processed at once. The gain matrix is then

$$b = P^*M^T(MP^*M^T+V)^{-1} = \begin{bmatrix} \frac{1}{19} & -\frac{4}{19} \\ -\frac{3}{19} & -\frac{7}{19} \\ \frac{6}{19} & -\frac{5}{19} \end{bmatrix} \tag{5.71}$$

Update the covariance matrix by

$$P = P^*-b(MP^*M^T+V)b^T = \begin{bmatrix} \frac{15}{19} & \frac{31}{19} & \frac{14}{19} \\ \frac{31}{19} & -\frac{2}{19} & \frac{15}{19} \\ \frac{14}{19} & \frac{15}{19} & \frac{27}{19} \end{bmatrix} \tag{5.72}$$

Note that it is identical to the P_N^* in Equation 5.61. Update the state matrix by

$$\hat{x} = \begin{bmatrix} \hat{x}_2' \\ \hat{x}_3' \\ \hat{x}_4' \end{bmatrix} + \begin{bmatrix} \frac{1}{19} & -\frac{4}{19} \\ -\frac{3}{19} & -\frac{7}{19} \\ \frac{6}{19} & -\frac{5}{19} \end{bmatrix} \begin{bmatrix} y_1' + \hat{x}_2' - \hat{x}_4' \\ y_2' + \hat{x}_2' \end{bmatrix}$$

Now

$$y_1' = y_2 - y_1 \tag{5.73}$$

$$y_2' = y_3 - y_1 \tag{5.74}$$

Substituting in Equation 5.73 and 5.74 and performing the indicated multiplication $\hat{x}$ becomes

$$\hat{x} = \begin{bmatrix} \frac{16}{19}\hat{x}_2' - \frac{1}{19}\hat{x}_4' + \frac{3}{19}y_1 + \frac{1}{19}y_2 - \frac{4}{19}y_3 \\ \hat{x}_3' - \frac{10}{19}\hat{x}_2' + \frac{3}{19}\hat{x}_4' + \frac{10}{19}y_1 - \frac{3}{19}y_2 - \frac{7}{19}y_3 \\ \frac{13}{19}\hat{x}_4' - \frac{1}{19}y_1 + \frac{6}{19}y_2 - \frac{5}{19}y_3 + \frac{1}{19}\hat{x}_2' \end{bmatrix} \tag{5.75}$$

This checks exactly with the noise states in the direct method.

In this example, R = 1, G = 3, P = 3, and P% = 22%. Also, C% = 31%.

In this example, the direct filter requires 22% longer than the indirect filter, but requires 31% less memory than the indirect. Note, if there was another measurement, the direct filter would be superior in both respects.

VI. SUMMARY

The goal of this thesis was to prove that the optimal complementary Kalman filter could be obtained from the normal Kalman filter equations by letting the variances of the signal variables approach infinity. Since the infinity terms could introduce errors in the computation, an algorithm was developed that would circumvent the infinity terms completely. The significance of this development was that the measurements could be processed sequentially, which lead to the conclusion that this filter would be fail-safe. That is, if failures in the measurements existed, they could be omitted in the processing, but optimal estimates of the signals would still be obtained from the remaining measurements. This is an advantage over many complementary filtering methods to date. That is, if there were a failure among the inputs, a backup system would be needed if estimates of the signals were to be made.

Another advantage is the ability to change the complementary Kalman filter equations to the usual Kalman filter equations. This can be accomplished by simple eliminating the first eight steps in the algorithm in Chapter II. Another, method would be to replace the variance of the signal variables with a large number in the usual Kalman filter equation to obtain the complementary Kalman filter.

In Chapter III the complementary Kalman filter was extended to the time continuous case. This was developed by letting the time increments of the discrete filter approach zero.

It was found that the calculation time for the complementary Kalman filter generally took longer than the indirect filter method. However, if there are a large number of redundant measurements, the computation time will approach that of the indirect filter or even be less. The second example demonstrated the case where less time could be involved.

A comparison of the amount of memory requirements was also made between the direct and indirect filters. It was found that the indirect filter required more memory than the direct for the cases of redundant measurements.

Two simple examples were worked to demonstrate the use of the algorithm of the complementary filter. These examples were also worked by using the indirect filter and the results were identical. The first example was the case where there were no noise states and it was obvious that the indirect filter was superior. The second example was typical of a navigation system for altitude determination. It was found that the direct filter could be superior to the indirect filter, both from computation time and memory requirements, if one more redundant measurement was added.

In general, the complementary Kalman filter is a fail-safe method to obtain the optimal estimate of the signals. Whether or not it is advantageous to use this method depends on the number of noise states and the amount of redundancy. With large redundancy the direct filter can be superior in all respects.

VII. LITERATURE CITED

1. Bakker, Marvin Lee. Recursive filtering with a distortionless constraint. Unpublished Ph.D. thesis. Ames, Iowa, Library, Iowa State University of Science and Technology. 1968.

2. Benning, Roger David. Multiple-input distortionless filters for estimating signals corrupted by noise. Unpublished Ph.D. thesis. Ames, Iowa, Library, Iowa State University of Science and Technology. 1966.

3. Bode, H. W. and Shannon, C. E. A simplified derivation of linear least square smoothing and prediction theory. Proc. I.R.E. 38: 417-425. April 1950.

4. Brock, Larry D. and Schmidt, George T. General questions on Kalman filtering in navigation systems. In Leondes, Cornelius T., editor. Theory and applications of Kalman filtering. Chapter 10. Pp. 203-230. U. S. Department of Commerce, National Technical Information Service AD 704 306. Feb. 1970.

5. Brown, R. G. Kalman filtering notes. Unpublished mimeographed notes for Electrical Engineering course 595K. Ames, Iowa Department of Electrical Engineering, Iowa State University of Science and Technology. 1967.

6. Brown, Robert Grover and Nilsson, James William. Introduction to linear systems analysis. New York, N.Y., John Wiley and Sons, Inc. c1962.

7. Davenport, Wilbur B., Jr. and Root, William L. An introduction to the theory of random signals and noise. New York, N.Y., John Wiley and Sons, Inc. 1958.

8. Derusso, Paul M., Roy, Rob J. and Close, Charles M. State variables for engineers. New York, N.Y., John Wiley and Sons, Inc. c1965.

9. Gaines, Harry T. Flight test of a Kalman filter doppler aided strapdown inertial navigator. Unpublished paper presented at Honeywell, Inc., Aerospace Division, St. Petersburg, Florida. 1971.

10. Huddle, J. R. Application of Kalman filtering theory to augmented inertial navigation systems. In Leondes, Cornelius T., editor. Theory and applications of Kalman filtering. Chapter 11. Pp. 231-268. U. S. Department of Commerce, National Technical Information Service Ad 704 306. Feb. 1970.

11. Kalman, R. E. A new approach to linear filtering and prediction problems. American Society of Mechanical Engineers Transactions (Journal of Basic Engineering) 82D: 35-45. 1960.

12. Levinson, N. Appendix C: Extrapolation, interpolation, and smoothing of stationary time series. In Wiener, N., author. The Interpolation, Extrapolation and Smoothing of Stationary Time Series. Pp. 149-160, New York, N.Y., John Wiley and Sons, Inc. 1949.

13. Reid, W. T. A matrix differential equation of Ricatti type. American Journal of Mathematics 68: 237-246. 1949.

14. Sorenson, H. W. Kalman filtering techniques. In Advances in Control Systems: Theory and Applications, Vol. 3. Pp. 219-292. Academic Press, New York, N.Y. 1966.

15. Unger, F. G. and Ott, N. Optimal use of redundant information in an inertial navigation system. In Leondes, Cornelius T., editor. Theory and applications of Kalman filtering. Chapter 18. Pp. 467-490. U. S. Department of Commerce, National Technical Information Service AD 704 306. Feb. 1970.

16. Wiener, N. The interpolation, extrapolation, and smoothing of stationary time series. New York, N.Y., John Wiley and Sons, Inc. 1949.

VIII. ACKNOWLEDGEMENTS

The author wishes to express his sincere thanks to Dr. R. G. Brown for the suggestion of this topic and his guidance throughout the development of this work. The author also wishes to thank his wife, Kay, and son, Jeff, who were very patient and understanding of the many evenings spent at the office working on this dissertation.

IX. APPENDIX A

The purpose of Appendix A is to determine the inverse of the following matrix as a approaches infinity.

$$P^* = \left[\begin{array}{cccc|c} a & 0 & \dots & 0 & \\ 0 & a & \dots & 0 & \bigcirc \\ \cdot & & & & \\ \cdot & & & & \\ 0 & \cdot & \dots & a & \\ \hline & & \bigcirc & & P_N^* \end{array}\right] \qquad \text{(A-1)}$$

A brief review of basic matrix manipulations are in order. The determinate of an (n x n) square matrix A is written as $|A|$. If the i^{th} row and j^{th} column of the determinate $|A|$ are deleted, the remaining n-1 rows and n-1 columns form a determinate $|M_{ij}|$. This determinate is called the minor of element a_{ij}. The cofactor of the element a_{ij} is equal to the minor of a_{ij}, with the sign $(-1)^{i+j}$ affixed. Thus, the cofactor (C_{ij}) of a_{ij} is defined as

$$C_{ij} = (-1)^{i+j}|M_{ij}| \qquad \text{(A-2)}$$

The Laplace expansion formula for the determinate of any (n x n) matrix A states that the determinant of A is given by the sum of the products of the elements of any single row or column and their respective cofactors.

Thus

$$|A| = \sum_{i=1}^{n} a_{ij}C_{ij} \qquad j = 1, \text{ or } 2, \cdots \text{ or } n \text{ (column expansion)}$$

(A-3)

$$|A| = \sum_{j=1}^{n} a_{ij}C_{ij} \qquad i = 1, \text{ or } 2, \cdots \text{ or } n \text{ (row expansion)}$$

The matrix formed by the cofactor's C_{ji} is defined as the adjoint matrix of A. That is, the adjoint matrix is the transpose of the matrix formed by replacing the elements a_{ij} by their cofactors. Then Derusso (8) defines the inverse of A as:

$$A^{-1} = \frac{\text{Adj } A}{|A|} \qquad \text{(A-4)}$$

Now to determine the inverse of Equation A-1, the determinate and adjoint matrix must be found. Using the row expansion of Equation A-3 the determinate of P^* is

$$|P^*| = \sum_{j=1}^{n} a_{ij}C_{ij} = a\ C_{11} \qquad \text{(A-5)}$$

since $a_{12}, a_{13}, \ldots a_{in} = 0$.

However,

$$C_{11} = \sum_{j=2}^{n} a_{2j}C_{2j}^{1,1} \qquad \text{(A-6)}$$

where $C_{ij}^{(1,1)}$ is the cofactor of P^* with the first row and column omitted. Expanding A-6 along row 2, C_{11} becomes

$$C_{11} = a\ C_{22}^{(1,1)} \qquad \text{(A-7)}$$

This process will be carried one step further

$$C_{22}^{(1,1)} = \sum_{j=3}^{n} a_{3j} C_{3j}^{(1,1),(2,2)} \qquad \text{(A-8)}$$

where $C_{3j}^{(1,1),(2,2)}$ is the cofactor of a_{3j} with rows 1 and 2 and columns 1 and 2 omitted.

Now P_S^* is of rank r then Equation A-5 can be iterated r times to yield

$$|P^*| = a^r C_{rr}^{(1,1)(2,2),..(r,r)} \qquad \text{(A-9)}$$

but $C_{rr}^{(1,1)(2,2),..(r,r)} = |P_N^*|$ so Equation A-9 is

$$|P^*| = a^r |P_N^*| \qquad \text{(A-10)}$$

The cofactors of the diagonal terms in P_S^* can be written as

$$C_{ii} = a^{r-1} |P_N^*| \qquad \text{for } i \leq r \qquad \text{(A-11)}$$

since C_{ii} does not contain the term a_{ii}. Also

$$C_{ij} = 0 \qquad \text{for } i \text{ and } j \leq r \text{ and } i \neq j \qquad \text{(A-12)}$$

This is true because the cofactor of C_{ij} is obtained y deleting row i and column j and the remaining determinate will have a complete row or column of zeros. An expansion on a row of zeros will yield a determinate equal to zero.

The cofactors of P_N^* will be of the following form

$$C_{ij} = a^r C_{ij}^{(1,1)(2,2),..(r,r),(i,j)} \quad \text{for } i \text{ and } j \geq r + 1 \quad \text{(A-13)}$$

The elimination of rows and columns greater than r does not delete any of the a's in P_S^*. Note that $C_{ij}^{(1,1)(2,2),..(r,r),(i,j)}$ for i and j ≥ r + 1 are simply the cofactors of submatrix P_N^*.

Thus P^{*-1} can be written as

$$P^{*-1} = \frac{\left[\begin{array}{cccc|c} C_{11} & 0 & \cdots & 0 & \\ 0 & C_{22} & \cdots & 0 & \bigcirc \\ 0 & . & \cdots & C_{rr} & \\ \hline & & \bigcirc & & a^r \text{Adj } P_N^* \end{array}\right]}{a^r \, |P_N^*|} \tag{A-14}$$

or

$$P^{*-1} = \left[\begin{array}{cccc|c} \frac{1}{a} & 0 & \cdots & 0 & \\ 0 & \frac{1}{a} & \cdots & 0 & \bigcirc \\ . & . & . & . & \\ 0 & . & \cdots & \frac{1}{a} & \\ \hline & & \bigcirc & & \dfrac{\text{Adj } P_N^*}{|P_N^*|} \end{array}\right] \tag{A-15}$$

Taking the limit as $a \rightarrow \infty$ P^{*-1} becomes

$$P^{*-1} = \left[\begin{array}{c|c} \bigcirc & \bigcirc \\ \hline \bigcirc & P_N^{*-1} \end{array}\right] \tag{A-16}$$

X. APPENDIX B

The inverse of the following equation is desired.

$$P^{-1} = \left[\begin{array}{c|c} M_S^T V^{-1} M_S & M_S^T V^{-1} M_N \\ \hline M_N^T V^{-1} M_S & M_N^T V^{-1} M_N + P_N^{*-1} \end{array}\right] \tag{B-1}$$

Let $A = M_S^T V^{-1} M_S$

$B = M_S^T V^{-1} M_N$

$C = M_N^T V^{-1} M_N + P^{*-1}$

then

$$P^{-1} = \left[\begin{array}{c|c} A & B \\ \hline B^T & C \end{array}\right] \tag{B-2}$$

The partitioned form of P is

$$P = \left[\begin{array}{c|c} P_S & P_3 \\ \hline P_3^T & P_N \end{array}\right] \tag{B-3}$$

A matrix multiplied by its inverse is the identity matrix, thus,

$$P^{-1}P = \left[\begin{array}{c|c} A & B \\ \hline B^T & C \end{array}\right] \left[\begin{array}{c|c} P_S & P_3 \\ \hline P_3^T & P_N \end{array}\right] = I \tag{B-4}$$

Performing the indicated multiplication yields the following four equations:

$$AP_S + BP_3^T = I \quad \text{(B-5)}$$

$$AP_3 + BP_N = 0 \quad \text{(B-6)}$$

$$B^T P_S + CP_3^T = 0 \quad \text{(B-7)}$$

$$B^T P_3 + CP_N = I \quad \text{(B-8)}$$

If the complementary constraint is to be used then the matrix M_S is of rank r. The quantity $M_S^T V M_S$ is an r x r matrix of rank r and, hence, has a inverse. Also assuming that $(M_N^T V^{-1} M_N + P_N^{*-1})$ is invertible, the matrices A and C have an inverse. Equations B-5 and B-7 can be used to obtain the following equation.

$$I = AP_S - BC^{-1}B^T P_S = [A - BC^{-1}B^T]P_S \quad \text{(B-9)}$$

Premultiplying both sides of Equation B-9 by $[A - BC^{-1}B^T]^{-1}$ gives the following equation for P_S.

$$P_S = [A - BC^{-1}B^T]^{-1} \quad \text{(B-10)}$$

Upon substituting the values for A, B, C

$$P_S = [M_S^T V^{-1} M_S - M_S^T V^{-1} M_N (M_N^T V^{-1} M_N + P_N^{*-1})^{-1} M_N V^{-1} M_S]^{-1} \quad \text{(B-11)}$$

At this time a Matrix Inversion Lemma as given by Sorenson (14) will be introduced.

Suppose (n x n) matrices B and R are positive-definites. Let H be any, possibly rectangular, matrix. Let A be an n x n matrix related to B, R, and H according to

$$A = B - BH^T[ABH^T + R]^{-1} HB \tag{B-12}$$

Then, A^{-1} is given by

$$A^{-1} = B^{-1} + H^T R^{-1} H \tag{B-13}$$

The proof is accomplished by direct multiplication and can be found in Sorenson (14) page 254.

Using Lemma I, in reverse Equation B-11 can be written as

$$P_S \{M_S^T V^{-1} M_S - M_S^T V^{-1} M_N [P_N^* - P_N^* M_N^T (M_N P_N^* M_N^T + V)^{-1} M_N P_N] M_N^T V^{-1} M_S \} \tag{B-14}$$

Introduce the identities

$$W = M_N P_N^* M_N^T + V \tag{B-15}$$

$$W - V = M_N P_N^* M_N^T \tag{B-16}$$

and upon substitution and rearrangement Equation B-14 becomes

$$P_S = [M_S^T V^{-1} M_S - M_S^T V^{-1} (W-V) V^{-1} M_S + M_S^T V^{-1} (W-V) W^{-1} (W-V) V^{-1} M_S]^{-1} \tag{B-17}$$

Collecting terms and expanding Equation B-17 becomes

$$P_S = \{M_S^T [V^{-1} - V^{-1}(W-V)V^{-1} + V^{-1}(W-V)W^{-1}(W-V)V^{-1}]M_S\}^{-1}$$

$$= \{M_S^T [V^{-1} - (I - W^{-1}V)V^{-1}]M_S\}^{-1}$$

$$= \{M_S^T [V^{-1} - V^{-1} + W^{-1}]M_S\}^{-1} = (M_S^T W^{-1} M_S)^{-1} \tag{B-18}$$

Substituting for W Equation B-18 is

$$P_S = [M_S^T(M_N P_N^* M_N^T + V)^{-1} M_S]^{-1} \tag{B-19}$$

Combining Equations B-6 and B-8 the expression for P_N is

$$P_N = C^{-1} + C^{-1} B^T P_S B C^{-1} \tag{B-20}$$

Using Lemma I and Equation B-15

$$C^{-1} = P_N^* - P_N^* M_N^T W^{-1} M_N P_N^* \tag{B-21}$$

The insertion of Equation B-21 and the expression for B into Equation B-20 produces

$$\begin{aligned} P_N = {} & P_N^* - P_N^* M_N^T W^{-1} M_N P_N^* \\ & + (P_N^* - P_N^* M_N^T W^{-1} M_N P_N^*) M_N^T V^{-1} M_S P_S M_S^T V^{-1} M_N (P_N^* - P_N^* M_N^T W^{-1} M_N P_N^*) \end{aligned} \tag{B-22}$$

Multiplying the terms gives

$$\begin{aligned} P_N = {} & P_N^* - P_N^* M_N^T W^{-1} M_N P_N^* + P_N^* M_N^T V^{-1} M_S P_S M_S^T V^{-1} M_N P_N^* \\ & - P_N^* M_N^T W^{-1} M_N P_N^* M_N^T V^{-1} M_S P_S M_S^T V^{-1} M_N P_N^* \\ & - P_N^* M_N^T V^{-1} M_S P_S M_S^T V^{-1} M_N P_N^* M_N^T W^{-1} M_N P_N^* \\ & + P_N^* M_N^T W^{-1} M_N P_N^* M_N^T V^{-1} M_S P_S M_S^T V^{-1} M_N P_N^* M_N^T W^{-1} M_N P_N^* \end{aligned} \tag{B-23}$$

Substituting Equation B-16 into Equation B-23 produces

$$\begin{aligned} P_N = {} & P_N^* - P_N^* M_N^T W^{-1} M_N P_N + P_N M_N^T W^{-1} M_S P_S M_S^T V^{-1} M_N P_N \\ & - P_N M_N^T V^{-1} M_S P_S M_S^T V^{-1} M_N P_N + P_N^* M_N^T V^{-1} M_S P_S M_S^T W^{-1} M_N P_N^* \end{aligned}$$

$$+ P_N^* M_N^T V^{-1} M_S P_S M_S^T V^{-1} (W-V) W^{-1} M_N P_N^*$$

$$- P_N^* M_N^T W^{-1} M_S P_S M_S^T V^{-1} (W-V) W^{-1} M_N P_N^* \qquad \text{(B-24)}$$

$$P_N = P_N^* - P_N^* M_N^T W^{-1} M_N P_N^* + P_N^* M_N^T W^{-1} M_S P_S M_S^T W^{-1} M_N P_N^* \qquad \text{(B-25)}$$

$$P_N = P_N^* - P_N^* M_N^T W^{-1} - W^{-1} M_S P_S M_S^T W^{-1} M_N P_N^* \qquad \text{(B-26)}$$

Using Equation B-15, Equation B-26 becomes

$$P_N = P_N^* - P_N^* M_N^T [(M_N P_N^* M_N^T + V)^{-1} - (M_N P_N^* M_N^T + V)^{-1}$$

$$M_S P_S M_S^T (M_N P_N^* M_N^T + V)^{-1}] M_N P_N^* \qquad \text{(B-27)}$$

Solving for P_3^T from Equation B-7

$$P_3^T = - C^{-1} B^T P_S \qquad \text{(B-28)}$$

or

$$P_3 = - P_S B (C^{-1})^T \qquad \text{(B-29)}$$

Using the equations for B, C^{-1}, and Equation B-15

$$P_3 = - P_S M_S^T V^{-1} M_N (P_N^* - P_N^* M_N^T W^{-1} M_N P_N^*) \qquad \text{(B-30)}$$

Expanding and using Equation B-16 gives

$$P_3 = - P_S M_S^T V^{-1} M_N P_N^* + P_S M_S^T V^{-1} (W-V) W^{-1} M_N P_N^*$$

$$= - P_S M_S^T V^{-1} M_N P_N^* + P_S M_S^T V^{-1} M_N P_N^* - P_S M_S^T W^{-1} M_N P_N^* \qquad \text{(B-31)}$$

Using Equation B-15

$$P_3 = - P_S M_S^T (M_N P_N^* M_N^T + V)^{-1} M_N P_N^* \qquad \text{(B-32)}$$

Then Equations B-19, B-27, and B-31 are the expressions to be used in the partitioned a posteriori covariance matrix P summerized here.

$$P = \left[\begin{array}{c|c} P_S & P_3 \\ \hline P_3^T & P_N \end{array}\right] \qquad \text{(B-33)}$$

where $$P_S = [M_S^T(M_N P_N^* M_N^T + V)^{-1} M_S]^{-1} \qquad \text{(B-34)}$$

$$P_N = P_N^* - P_N^* M_N^T [(M_N P_N^* M_N^T + V)^{-1} - (M_N P_N^* M_N^T + V)^{-1} M_S P_S M_S^T$$

$$(M_N P_N^* M_N^T + V)^{-1} M_N P_N^* \qquad \text{(B-35)}$$

$$P_3 = - P_S M_S^T (M_N P_N^* M_N^T + V)^{-1} M_N P_N^* \qquad \text{(B-36)}$$

XI. APPENDIX C

Bakker's (1) a posteriori covariance matrix, in partitioned form (see Equation 2.23), is written as

$$P = \left[\begin{array}{c|c} b_S^*(M_N P_N^* M_N^T + V)b_S^{*T} & - b_S^* M_N P_N^* \\ & + b_S^*(M_N P_N^* M_N + V)b_N^{*T} \\ \hline - P_N^* M_N^T b_S^{*T} & P_N^* - b_N^* M_N P_N^* - P_N^* M_N^T b_N^{*T} \\ + b_N^*(M_N P_N^* M_N^T + V)b_S^{*T} & + b_N^*(M_N P_N^* M_N^T + V)b_N^{*T} \end{array}\right] \qquad \text{(C-1)}$$

b_S^* and b_N^* (see Equation 1.45 and 1.46) are

$$b_S^* = [M_S^T(M_N P_2^* M^T + V)^{-1} M_S^T(M_N P_2^* M^T + V)^{-1} \qquad \text{(C-2)}$$

$$b_N^* = P_N^* M_N^T (M_N P_2^* M^T + V)^{-1} [I - M_S b_S^*] \qquad \text{(C-3)}$$

where $$P_2^* = [P_3^{*T} \mid P_N^*] \qquad \text{(C-4)}$$

P can be written in partitioned form as

$$P = \left[\begin{array}{c|c} P_S & P_3 \\ \hline P_3^T & P_N \end{array}\right] \qquad \text{(C-5)}$$

Introduce the following notation.

Let $$Z = (M_N P_2^* M^T + V) \qquad \text{(C-6)}$$

Using Equation C-4 and the partitioned form of M, Equation C-6 can be written as

$$Z = (M_N P_3^{*T} M_S^T + M_N P_N^* M_N^T + V) \qquad \text{(C-7)}$$

Also let

$$A = M_N P_3^{*T} M_S^T \tag{C-8}$$

Then

$$(M_N P_N^* M_N^T + V) = Z - A \triangleq W \tag{C-9}$$

A look at Equations C-1 and C-5 indicate that

$$P_S = b_S^* (M_N P_N^* M_N^T \div V) b_S^{*T} \tag{C-10}$$

Using Equation C-2 and C-6 through Equation C-9 Equation C-10 can be written as

$$P_S = (M_S^T Z^{-1} M_S)^{-1} M_S^T Z^{-1} (Z - A)(Z^T)^{-1} M_S [M_S^T (Z^T)^{-1} M_S]^{-1}$$

$$= (M_S^T Z^{-1} M_S)^{-1} M_S^T (I - Z^{-1} A)(Z^T)^{-1} M_S [M_S^T (Z^T)^{-1} M_S]^{-1}$$

$$= (M_S^T Z^{-1} M_S)^{-1} [M_S^T (Z^T)^{-1} M_S - M_S^T Z^{-1} A (Z^T)^{-1} M_S] (M_S^T (Z^T)^{-1} M_S)^{-1}$$

$$= (M_S^T Z^{-1} M_S)^{-1} [I - M_S^T Z^{-1} A (Z^T)^{-1} M_S (M_S^T Z^{-1} M_S)^{-1}] \tag{C-11}$$

Using Equation C-8, Equation C-11 becomes

$$P_S = (M_S^T Z^{-1} M_S)^{-1} [I - M_S^T Z^{-1} M_N P_3^{*T} (M_S^T Z^{T-1} M_S)(M_S^T Z^{-1} M_S)^{-1}$$

$$= (M_S^T Z^{-1} M_S)^{-1} [I - M_S^T Z^{-1} M_N P_3^{*T}] \tag{C-12}$$

Post multiply both sides by M_S^T and using Equation C-8 we have

$$P_S M_S^T = (M_S^T Z^{-1} M_S)^{-1} [M_S^T - M_S^T Z^{-1} A] \tag{C-13}$$

but A = (Z - W), therefore, Equation C-13 becomes

$$P_S M_S^T = (M_S^T Z^{-1} M_S)^{-1} [M_S^T - M_S^T Z^{-1} (Z-W)] = (M_S^T Z^{-1} M_S)^{-1}$$

$$[M_S^T - M_S^T + M_S^T Z^{-1} W] = (M_S^T Z^{-1} M_S)^{-1} M_S^T Z^{-1} W \tag{C-14}$$

Now post multiply both sides by $W^{-1}M_S$ which yields

$$P_S M_S^T W^{-1} M_S = (M_S^T Z^{-1} M_S)^{-1} M_S^T Z^{-1} M_S = I \tag{C-15}$$

Post multiplying both sides by $(M_S^T W^{-1} M_S)^{-1}$ yields

$$P_S = (M_S^T W^{-1} M_S)^{-1} = [M_S^T (M_N P_N^* M_N^T + V)^{-1} M_S]^{-1} \tag{C-16}$$

which is the same as the P_S from the direct approach.

Next, look at the P_N term which is

$$P_N = P_N^* - b_N^* M_N P_N^* - P_N^* M_N^T b_N^{*T} + b_N^* (M_N P_N^* M_N^T + V) b_N^{*T} \tag{C-17}$$

Using Equation C-2 and C-6 through C-9, Equation C-3 can be written as

$$b_N^* = P_N^* M_N^T Z^{-1} [I - M_S (M_S^T Z^{-1} M_S)^{-1} M_S^T Z^{-1}] \tag{C-18}$$

However, note in Equation C-14 if both sides are post multiplied by W^{-1} we have

$$P_S M_S^T W^{-1} = (M_S^T Z^{-1} M_S)^{-1} M_S^T Z^{-1} \tag{C-19}$$

Then using this result Equation C-18 becomes

$$b_N^* = P_N^* M_N^T Z^{-1} [I - M_S P_S M_S^T W^{-1}] \tag{C-20}$$

Using Equation C-20 and upon factoring Equation C-17, $P_N^* M_N^T$ becomes

$$P_N = P_N^* - P_N^* M_N^T \{ Z^{-1} - Z^{-1} M_S P_S M_S^T W^{-1} + (Z^T)^{-1} - W^{-1} M_S P_S M_S^T (Z^T)^{-1} \\ + [Z^{-1} - Z^{-1} M_S P_S M_S^T W^{-1}] W [W^{-1} M_S P_S M_S^T (Z^T)^{-1} - (Z^T)^{-1}] \} M_N P_N^* \tag{C-21}$$

Let X be equal to the term in the brackets {...}. Then multiplying and rearranging terms X becomes

$$X = Z^{-1}[I - M_S P_S M_S^T W^{-1}] + [I - M_S P_S M_S^T W^{-1}]^T (Z^T)^{-1}$$

$$+ Z^{-1}W\ [W^{-1}M_S P_S M_S^T (Z^T)^{-1} - (Z^T)^{-1}]$$

$$- Z^{-1}M_S P_S M_S^T W^{-1}\ [WW^{-1}M_S P_S M_S^T (Z^T)^{-1} - (Z^T)^{-1}] \qquad \text{(C-22)}$$

Further multiplication yields

$$X = Z^{-1}[I - M_S P_S M_S^T W^{-1}] + [I - M_S P_S M_S^T W^{-1}]^T (Z^T)^{-1}$$

$$+ Z^{-1}M_S P_S M_S^T (Z^T)^{-1} - Z^{-1}W(Z^T)^{-1} + Z^{-1}M_S P_S M_S^T W^{-1}(Z^T)^{-1}$$

$$- Z^{-1}M_S P_S M_S^T W^{-1}M_S P_S M_S^T (Z^T)^{-1} \qquad \text{(C-23)}$$

The last term in Equation C-23 can be reduced by noting that $M_S^T W^{-1} M_S = P_S^{-1}$, therefore

$$Z^{-1}M_S P_S M_S^T W^{-1}M_S P_S M_S^T (Z^T)^{-1} = Z^{-1}M_S P_S P_S^{-1} P_S M_S^T (Z^T)^{-1}$$

$$= Z^{-1}M_S P_S M_S^T (Z^T)^{-1} \qquad \text{(C-24)}$$

This term cancels the previous term in Equation C-23, and Equation C-23 becomes

$$X = Z^{-1} - Z^{-1}M_S P_S M_S^T W^{-1} - W^{-1}M_S P_S M_S^T (Z^T)^{-1}$$

$$+ Z^{-1}M_S P_S M_S^T (Z^T)^{-1} + Z^{-1}A(Z^T)^{-1} \qquad \text{(C-25)}$$

Pre-multiplying and post-multiplying both sides of Equation C-25 by W gives

$$W\ X\ W = WZ^{-1}W - WZ^{-1}M_S P_S M_S^T - M_S P_S M_S^T (Z^T)^{-1}W$$

$$+ WZ^{-1}M_S P_S M_S^T (Z^T)^{-1}W + WZ^{-1}A(Z^T)^{-1}W \qquad \text{(C-26)}$$

Observe that W = Z-A so

$$WZ^{-1} = I - AZ^{-1} \tag{C-27}$$

Using Equation C-27, Equation C-26 becomes

$$W\,X\,W = W - AZ^{-1}W - M_S P_S M_S^T + AZ^{-1}M_S P_S M_S^T$$

$$- AZ^{-1}M_S P_S M_S^T (Z^T)^{-1}W + A(Z^T)^{-1} - AZ^{-1}A(Z^T)^{-1}W \tag{C-28}$$

Note that

$$W = W^T = Z^T - A^T \tag{C-29}$$

then

$$(Z^T)^{-1}W = I - (Z^T)^{-1}A^T \tag{C-30}$$

Using this result Equation C-28 becomes upon canceling terms

$$W\,X\,W = W - M_S P_S M_S^T + (AZ^{-1}M_S P_S M_S^T - A + AZ^{-1}A)(Z^T)^{-1}A^T \tag{C-31}$$

Let Y equal the identity in the first parenthesis and substitute the following equation

$$P_S M_S^T = (M_S^T Z^{-1} M_S)^{-1} M_S^T Z^{-1} W \tag{C-32}$$

$$A = M_N P_3^{*T} M_S^T \tag{C-33}$$

to give for Y the following form

$$Y = M_N P_3^{*T} M_S^T Z^{-1} M_S (M_S^T Z^{-1} M_S)^{-1} M_S^T Z^{-1} W$$

$$- M_N P_3^{*T} M_S^T + M_N P_3^{*T} M_S^T Z^{-1} M_N P_3^{*T} M_S^T \tag{C-34}$$

which reduces to

$$Y = M_N P_3^{*T} M_S^T Z^{-1} W - M_N P_3^{*T} M_S^T + M_N P_3^{*T} M_S^T Z^{-1} M_N P_3^{*T} M_S^T \tag{C-35}$$

Now substituting Equation C-8 we have

$$Y = AZ^{-1}W - A + AZ^{-1}A \qquad \text{(C-36)}$$

Using Equation C-27 this reduces to

$$Y = A(I - AZ^{-1}) - A + AZ^{-1}A \equiv 0 \qquad \text{(C-37)}$$

Then Equation C-31 is

$$W X W = W - M_S P_S M_S^T \qquad \text{(C-38)}$$

Upon pre- and post-multiplying both sides by W^{-1}, Equation C-38 becomes

$$X = \{W^{-1} - W^{-1}M_S P_S M\} W^{-1} \qquad \text{(C-39)}$$

then substituting Equation C-21 we have

$$P_N = P_N^* - P_N^* M_N^T [W^{-1} - W^{-1}M_S P_S M_S^T W^{-1}] M_N P_N^* \qquad \text{(C-40)}$$

Using the expression for W^{-1}

$$P_N = P_N^* - P_N^* M_N^T [(M_N P_N^* M_N^T + V)^{-1} - (M_N P_N^* M_N^T + V)^{-1} M_S P_S M_S^T$$

$$(M_N P_N^* M_N^T + V)^{-1}] M_N P_N^* \qquad \text{(C-41)}$$

which is identical to the P_N obtained in the direct approach.

Next examine the P_3 term given by

$$P_3 = - b_S^* M_N P_N^* + b_S^* (M_N P_N^* M_N^T + V) b_N^{*T} \qquad \text{(C-42a)}$$

Using Equations C-3 and C-6 through C-9, P_3 becomes

$$P_3 = - b_S^* M_N P_N^* + b_S^* W (Z^T)^{-1} M_N P_N^*$$

$$- b_S^* W b_S^{*T} M_S^T (Z^T)^{-1} M_N P_N^* \qquad \text{(C-42b)}$$

Note that $b_S^* W b_S^{*T} = P_S$ and factoring $M_N P_N^*$ Equation C-41 can be written as

$$P_3 = [- b_S^* + b_S^* W(Z^T)^{-1} - P_S M_S^T (Z^T)^{-1}] M_N P_N^* \qquad \text{(C-42c)}$$

W can be written as Z-A; however, we know that $W = W^T = Z^T - A^T$ so

$$W(Z^T)^{-1} = I - A^T (Z^T)^{-1} \qquad \text{(C-43)}$$

Thus

$$P_3 = [- b_S^* A^T (Z^T)^{-1} - P_S M_S^T (Z^T)^{-1}] M_N P_N^* \qquad \text{(C-44)}$$

Substituting Equation C-2 and C-6, Equation C-43 is

$$P_3 = [- (M_S^T Z^{-1} M_S)^{-1} M_S^T Z^{-1} A^T (Z^T)^{-1} - P_S M_S^T (Z^T)^{-1}] M_N P_N^* \qquad \text{(C-45)}$$

Using Equation C-14

$$P_3 = [- P_S M_S^T W^{-1} A^T (Z^T)^{-1} - P_S M_S^T (Z^T)^{-1}] M_N P_N^* \qquad \text{(C-46)}$$

$$= P_S M_S^T [- W^{-1} A^T (Z^T)^{-1} - (Z^T)^{-1}] M_N P_N^* \qquad \text{(C-47)}$$

Observe that $A^T = Z^T - W$ so

$$P_3 = P_S M_S^T [- W^{-1} (Z^T - W)(Z^T)^{-1} - (Z^T)^{-1}] M_N P_N^*$$

$$= P_S M_S^T [- W^{-1} Z^T (Z^T)^{-1} + (Z^T)^{-1} - (Z^T)^{-1}] M_N P_N^*$$

$$= P_S M_S^T W^{-1} M_N P_N^* \qquad \text{(C-48)}$$

Substituting expression for W^{-1}

$$P_3 = P_S M_S^T (M_N P_N^* M_N^T + V)^{-1} M_N P_N^* \qquad \text{(C-49)}$$

which is identical to the P_3 in the direct approach.

XII. APPENDIX D

The number of multiplies will be counted for both the indirect and direct filter. This is accomplished by proceeding through the algorithms for the indirect and direct filters.

The number of multiplications when two matrices are multiplied together is first determined. A (BxC) matrix denotes B rows and C columns. The product of a (BxC) matrix times a (CxD) matrix is written as

$$(BxC) \times (CxD) \tag{D-1}$$

The number of multiplies involved in this calculation is given by

$$M = BCD \tag{D-2}$$

In order to determine the multiplies involved in the following equation, some nomenclature will be introduced. For example, the product of matrices XY will be written as

$$XY = (BxC) \times (CxD) = \text{"}BCD\text{"} \tag{D-3}$$

That is matrix X is a (BxC) matrix and Y is a (CxD) matrix. Thus, BCD multiplies are involved in the product of XY.

First, calculate the number of multiplications involved in the direct filter. The sizes of the respective matrices are as follows:

$$M_i = (1 \times (R + G))$$

$$M_{iS} = (1 \times R)$$

$$M_{iN} = (1 \times G)$$

$$P = ((R + G) \times (R + G))$$

$$P_N = (G \times G)$$

$x \quad = ((R + G) \times 1)$

$R_i \quad = (R \times R)$

$b \quad = (R \times 1)$

$b_S \quad = (R \times 1)$

$b_N \quad = (G \times 1)$

$\phi_N \quad = (G \times G)$

$P_N^* \quad = (G \times G)$

Assume that the first R measurement yield an independent linear combination of the R signal variables. That is, $R_i M_{iS}^T \neq 0$ for the first R measurements and $(I - b_R M_{RS}) R_{R-1} = 0$ at the R^{th} measurement. Thus, steps 3 through 7 in the algorithm will be calculated R times and steps 9 through 11 will be calculated (P-R) times. Each step of the algorithm will be listed and the number of multiplies will be counted.

1. No multiplies.
2. No multiplies.
3. Compute

$$R_{i-1} M_{iS}^T = (RxR) \times (Rx1) = "R^2"$$

However, the first step is trivial, because $R_o = I$. Therefore, no multiplies are needed for the first measurement. Thus, the total number of multiplies for this step is "$R^2(R-1)$".

4. Calculate the gain matrix by

$$b_i = \begin{bmatrix} \dfrac{R_{i-1} M_{iS}^T}{M_{iS} R_{i-1} M_{iS}^T} \\ ----------- \\ 0 \end{bmatrix}$$

$R_{i-1}M_{iS}^T$ is computed in steps 3, so the additional compution is

$$M_{iS}(R_{i-1}M_{iS}^T) = (1xR) \times (Rx1) = "R"$$

An additional "R" multiplies is required when $R_{i-1}M_{iS}^T$ is multiplied by $\dfrac{1}{M_{iS}R_{i-1}M_{iS}^T}$.

Therefore, the total number of multiplies for this step is "2R(R)" = "$2R^2$".

5. Update the estimate of the states by

$$\hat{x}_i = \hat{x}_i' + b_i(y_i - M_i\hat{x}_i')$$

$$M_i\hat{x}_i = (1\ x(R+G)) \times ((R+G) \times 1) = "R+G".$$

However, for the first measurement $\hat{x}_i'$ only contains terms for the noise variables, which only require "G" multiplies. Now,

$$b_i(y_i - M_i\hat{x}_i') = (Rx1) \times (1x1) = "R",$$

because $b_N = 0$.

Therefore the total number of multiplies for step five is

$$"(R+G)(R-1) + R^2 + G" = "2R^2 + RG - R" .$$

6. Update the covariance matrix by

$$P_i = (I - b_iM_i)P_{i-1}(I - b_iM_i)^T + b_iV_ib_i^T$$

$$b_iM_i = ((R+G)\ x1) \times (1\ x(R+G)) = "(R+G)^2"$$

$$(I - b_iM_i)P_{i-1} = (R\ x(R+G)) \times ((R+G)x(R+G)) = "R(R+G)^2" .$$

Then the product of

$$[(I - b_iM_i)P_{i-1}](I - b_iM_i)^T$$

$$= ((R+G)x(R+G))x((R+G)x R) = "R(R+G)^2".$$

Also,

$$b_iV_i = (Rx1) x (1x1) = "R"$$

and

$$(b_iV_i)b_i^T = (Rx1) x (1xR) = "R^2".$$

Therefore, the total number of multiplies in step 6 is

$$"R(R+G)^2 + 2R^2(R+G)^2 + R^3 + R^2"$$

$$= "2R^4 + 2R^3 + R^2 + 4R^3G + 2R^2G^2 + 2R^2G + RG^2"$$

7. Update R_i by

$$R_i = (I - b_{iS}M_{iS})R_{i-1}$$

which can be written as

$$R_i = (RxR) x (RxR) = "R^3"$$

The first step doesn't need to be calculated since $R_o = I$. Thus, the total number of multiplies for step seven is $"(R^4 - R^3)"$.

8. No multiplies involved.

9. Calculate the gain matrix by

$$b_i = \frac{P_{i-1}M_i^T}{(M_iP_{i-1}M_i^T + V_i)}$$

$$P_{i-1}M_i^T = ((R+G)x(R+G)) x ((R+G)x 1) = "(R+G)^2"$$

and

$$M_i(P_{i-1}M_i^T) = (1 \times (R+G)) \times ((R+G) \times 1) = "(R+G)" .$$

The product of $P_{i-1}M_i^T$ by $\frac{1}{(M_i P_{i-1} M_i^T + V_i)}$ will produce an additional "R+G" multiplies. Therefore, total number of multiplies for step nine is

$$"[(R+G)^2 + 2(R+G)](P-R)" .$$

10. Update the estimate by

$$\hat{x}_i = \hat{x}_i' + b_i(y_i - M_i\hat{x}_{i-1})$$

$$M_i\hat{x}_{i-1} = (1 \times (R+G)) \times ((R+G) \times 1) = "R+G" .$$

Also,

$$b_i(y_i - M_i\hat{x}_{i-1}') = ((R+G) \times 1) \times (1 \times 1) = "R+G" .$$

Therefore, step seven produces "2(R+G)(P-R)" multiplies.

11. Update the covariance matrix by

$$P_i = P_{i-1} - b_i(M_i P_{i-1} M_i^T + V_i)s_i^T$$

Now,

$$b_i(M_i P_{i-1} M_i^T + V_i) = P_{i-1}M_i^T$$

which is already calculated. Then

$$(P_{i-1}M_i^T)s_i^T = ((R+G) \times 1) \times (1 \times (R+G)) = "(R+G)^2" .$$

Therefore, the total number of multiplies in step eleven is "$(R+G)^2(P-R)$".

The remaining multiplies occur in the extrapolution of the states and covariance matrices to the next time interval. The estimate of the states are extrapolation by

$$\hat{x}_N' = \phi_N \hat{x}_N = (G\text{x}G) \text{ x } (G\text{x}1) = \text{"}G^2\text{"} .$$

The covariance matrix is given by

$$P_N^* = \phi_N P_N \phi_N^T + H_N .$$

$$Q_N P_N \phi_N^T = (G\text{x}G) \text{ x } (G\text{x}G) \text{ x } (G\text{x}G) = \text{"}2G^3\text{"} .$$

This completes the count on the number of multiplies for the direct filter between one time interval. Denote the total number of multiplies for the direct filter as M_D and summing the multiplies for each step M_D becomes

$$M_D = 3R^4 - R^3 - R - 3RG + 4R^3G + 2R^2G^2 - 4R^2G - 2RG^2$$
$$+ 2G^3 + G^2 + P(2R^2 + 4RG + 2G^2 + 4R + 4G) \qquad \text{(D-4)}$$

With reference to Figure 4.2, the number of multiplication for the indirect filter will now be determined. The algebraic operator will consist of two matrix multiplications to give the desired $S_i + N_i(t)$ equation and the $N^i(t)$ noise equations. That is there will be the following two matrix multiplies:

$$Cy = \begin{bmatrix} N \\ N^2 \\ \vdots \\ N^{(P-R)} \end{bmatrix} \qquad \text{(D-5)}$$

and

$$Dy = \begin{bmatrix} S_1 + N_1(t) \\ S_2 + N_2(t) \\ \vdots \\ S_R + N_R(t) \end{bmatrix} \tag{D-6}$$

Now since y is a (Px1) matrix, then C is a ((P-R) x P) matrix. The product of Cy will involve "P(P-R)" multiplies. Similarity Dy = (RxP) x (Px1) = "RP" .

Consider the inputs to the Kalman filter. They are linear combinations of the noise measurements. Thus, $N^1(t)$, $N^2(t)$, ⋯ $N^{(P-R)}(t)$ will not be uncorrelated with each other, which means that sequential processing is not possible with the (P-R) inputs to the Kalman filter. Thus, they must all be processed at once. The matrices involved with the Kalman filter part of the indirect filter are of the following size:

$M = ((P-R) \times G)$

$P = (G \times G)$

$P^* = (G \times G)$

$\phi = (G \times G)$

$x = (G \times 1)$

$b = (G \times (P-R))$

The first step in the Kalman filter is to calculate the gain matrix given by

$$b = P^* M^T (M P^* M^T + V)^{-1} \tag{D-7}$$

P^*M^T = (GxG) x (G x(P-R)) = "G^2(P-R)" and $M(P^*M^T)$ = ((P-R)x G) x (G x(P-R)) = "G$(P-R)^2$". However, a (P-R)x(P-R) inverse must now be performed, and a conservative number of multiplies of an inverse of this size is "2$(P-R)^3$". Then the product of P^*M^T times the inverse involves a (G x(P-R)) x ((P-R)x(P-R)) matrix which yields "G$(P-R)^2$" multiplies.

Therefore, the total number of multiplies in the calculate of the gain matrix is "$G^2(P-R) + 2G(P-R)^2 + 2(P-R)^3$".

The update of the a priori covariance matrix is

$$P = P^* - bMP^* = P^* - P^*M^Tb^T \tag{D-8}$$

P^*M^T has already been calculated, so the only multiplies involved is $(P^*M^T)b^T$ = (G x(P-R)) x ((P-R)x G) = "G^2(P-R)".

The update of the signal state is given by

$$\hat{x} = \hat{x}' + b(y - M\hat{x}') \tag{D-9}$$

$M\hat{x}'$ = ((P-R)x G) x (Gx1) = "G(P-R)" and then $b(y - M\hat{x}')$ = (G x(P-R)) x ((P-R)x 1) = "G(P-R)". Therefore, the total multiplies involved in updating the state estimate is "2G(P-R)".

The Kalman filter yields the optimal estimate of G noise variables. However, a matrix multiply is needed to get the best estimate of $N_i(t)$. This matrix multiply will be of the form

$$E\,\hat{x} = \begin{bmatrix} N_1 \\ \\ N_R \end{bmatrix} \tag{D-10}$$

and the number of multiplies involved is "RG".

The extrapolation of the covariance matrix and noise variables ahead in time will be identical to that of the direct filter. That is $P^* =$ "$2G^3$" multiplies and $\hat{x}' =$ "G^2" multiplies.

This completes the count of multiplies for the indirect filter at one time interval. Denote the total number of multiplies for the direct filter as M_I and summing it becomes

$$M_I = P^2 + 2(P-R)(G^2+G) + 2G(P-R)^2 + 2(P-R)^3 + RG + 2G^3 + G^2 \qquad \text{(D-11)}$$

The amount of memory for the indirect and direct filter are calculated next. It will be noted which matrices need to be stored. A count on the memory requirement will be done in the following manner. If an (n x n) matrix needs to be stored it will count as n^2 memory cells. This is, there are n^2 characters in an (n x n) matrix.

First, consider the direct filter. The matrices that need to be stored are as follows:

$R_i = (R \times R)$

$\hat{x}_i = ((R+G) \times 1)$

$P_i = (R+G) \times (R+G)$

$\phi_N = (G \times G)$

$H_N = (G \times G)$

$V = (1 \times 1)$, but there are P of them

$M = (P \times (R+G)$

$y_i = (P \times 1)$

The total of the memory cells needed so far is

$$"R^2 + (R+G) + (R+G)^2 + 2G^2 + 2P + P(R+G)"$$

Now we will proceed through the algorithm and determine what additional memory needs to be stored.

1. No additional memory needed.

2. No additional memory needed.

3. $R_{i-1,S}M_{iS}^T = (RxG)$, which is used later to calculate b_i, so needs to be stored.

4. $b_{iS} = \dfrac{R_{i-1}M_{iS}^T}{M_{iS}R_{i-1}M_{iS}^T}$ $\quad b_{iN} = 0$

No additional memory is needed here, because it can be computed without storing additional memory. However, b_{iS} will need to be stored because it is used later. Thus, $b_{iS} = ((R+G) x\ 1)$.

5. Compute $\hat{x}_i = \hat{x}_i' + b_{iS}(y_i - M_i\hat{x}')$

Now $M_i\hat{x}_i$ can be calculated then subtracted from y_i and then multiplied by b_{iS} which is then added to $\hat{x}_i'$ to give $\hat{x}_i$. Then $\hat{x}_i$ is put back in place of $\hat{x}_i'$ and hence, additional memory is not needed in this step.

6. $P_i = (I - b_iM_i)P_{i-1}(I - b_iM_i)^T + b_iV_ib_i^T$

Now b_iV_i can be calculated then post-multiplied by b_i^T , then this will need to be put in memory to be added later. This requires the storage of a (R x R) matrix. Now, b_iM_i can be calculated and then subtracted from I, but this will have to be put in memory to post-multiply by $(I-b_iM_i)^T$. Now this will require an (R+G)x(R+G) matrix to be stored. Now P_i will just replace P_{i-1}, so no more additional memory is needed.

7. $R_i = (I - b_{iS}M_{iS})P_{i-1}(I - b_{iS}M_{iS})^T$

This step will not require any additional memory because $(I - b_{iS}M_{iS})$ is already stored.

8. No memory needed.

9. $$b_i = \frac{P_{i-1}M_i^T}{(M_iP_{i-1}M_i^T + V_i)}$$

$P_{i-1}M_i^T$ will need to be stored and this is an ((R+G)x 1) matrix. The rest of the calculations will not require any more storage, except for b_i which will need to be stored for the remaining steps and it is a ((RxG) x 1) matrix.

10. This step will not require any memory, for the same reason as step 5.

11. This step will not require any memory because $b_i(M_iP_{i-1}M^T + V_i) = P_{i-1}M_i^T$ which is already stored and by B_i^T is also stored. Again P_i will just replace P_{i-1}.

12. - 15. The rest of the steps will not require additional memory because all quantities are already been stored and the new calculated matrix will just replace the old ones.

Note, we can eliminate memory required in steps 3 - 7, because they can be put in the slots of 9 through 10. Thus, the total memory cells needed for the direct filter, denoted by C_D, is

$$C_D = 2R^2 + 3R + 3G + 2RG + 3G^2 + 2P + PR + PG \qquad \text{(D-12)}$$

In the indirect filter the following matrices must be stored:

y_i = (P x 1)

ϕ = (G * G)

P = (G x G)

H = (G x G)

v = (P-R)x(P-R)

M = (P x(R+G))

$\hat{x}$ = (G x 1)

Also, the matrices that pre-multiplies the y_i to give the $S_i + N_i(t)$ equation and the $N^i(t)$ equation. These are a (R+P) matrix and a ((P-R)x P) matrix. Also needed is a matrix that multiplies the outputs of the Kalman filter to get the $\hat{N}_i(t)$. It is a (R+G) matrix. Also needed to be stored will be the $S_i + N_i(t)$ and the $N^i(t)$ which are (R x 1) and (P-R)x 1 matrices respectively. Also needed is the M matrix for the $N^i(t)$'s. This is ((P-R)x G) matrix.

The algorithm for the indirect filter will be gone through and the additional memory needed will be counted.

1. $b = P^*M^T(MP^*M^T + V)^{-1}$

P^*M^T can be calculated and will need to be stored. This is of size (G x(P-R)). Then $(MP^*M^T + V)$ can be calculated without additional memory, but the inverse will require a (P-R)x(P-R) memory cells in order to find the inverse. The gain matrix b = (G x(P-R)) matrix will be stored.

2. $P = P^* - b(MP^*M^T + V)b^T$ will not require additional memory, because $b(MP^*M^T + V) = P^*M^T$ is already stored and so is b^T. P will just replace P^*.

3. $\hat{x} = \hat{x}_i' + b_i(y_i - M_i\hat{x}_i)$ will not require any additional memory.

The update of the covariance and states will not require any more memory either. Thus, the total memory needed for the indirect filter, denoted by C_I, is

$$C_I = 2P + 3G^2 + 4P^2 - 5PR + 4R^2 - RG + G + 3PG \qquad \text{(D-13)}$$

This completes the count of memory cells needed for the indirect and direct filters.