Kalman Filtering in Two Dimensions

JOHN W. WOODS, MEMBER, IEEE, AND CLARK H. RADEWAN, MEMBER, IEEE

***Abstract*—The Kalman filtering method is extended to two dimensions. The resulting computational load is found to be excessive. Two new approximations are then introduced. One, called the strip processor, updates a line segment at a time; the other, called the reduced update Kalman filter, is a scalar processor. The reduced update Kalman filter is shown to be optimum in that it minimizes the post update mean-square error (mse) under the constraint of updating only the nearby previously processed neighbors. The resulting filter is a general two-dimensional recursive filter.**

I. Introduction

In RECENT YEARS there have been many attempts to extend Kalman filtering to two dimensions [1]–[3]. This is a direct result of the method's great success in one dimension. In the two-dimensional (2-*D*) case, the enormity of the data calls particularly for an efficient recursive processor. Unfortunately, the previous efforts to achieve a truly recursive 2-*D* Kalman filter were of only limited success because of both the difficulty in establishing a suitable 2-*D* recursive model and also the high dimension of the resulting state vector. In fact, a straightforward extension of one-dimensional (1-*D*) Kalman filter techniques would result in a number of state variables proportional to N for the filtering of an $N \times N$ digital image.

In this paper, we propose two schemes which to a large extent overcome the computational problems that have precluded the use of 2-*D* Kalman-like processors. We develop two new approximations: one to the 2-*D* Kalman vector processor (updates a line at a time), called the Kalman strip filter; and another to the 2-*D* Kalman scalar processor, (updates a point at a time) called the reduced update filter. Both of these processors offer computationally effective solutions to the problem of 2-*D* Kalman filtering. These results have application to a wide range of problems involving estimation on 2-*D* data fields, including the image processing problem where the utility of linear filtering is well-established. The theory includes the case of space-variant models, allowing a better match to local source statistics, in turn permitting a greater noise reduction. Further, it should be possible to treat nonlinear space-variant models via techniques similar to those of the 1-*D* extended Kalman filtering [5].

We start with a brief review of the concept of state and its role in 1-*D* Kalman filtering. Then we define the 2-*D* Kalman scalar and vector filters, and we point out their undesirable computational properties in that the state vector grows with the image size. Next, we present the Kalman strip filter and the reduced update Kalman filter. Finally, we present examples of application of the filters in a simulated data environment.

The Concept of State

The concept of state is central to linear system theory in one dimension. The state is defined as the minimum information about the past and present needed to determine all future responses given the future input [4]. For systems governed by differential or difference equations, the state is simply a sufficient set of initial conditions for the one-point boundary value problem. The state concept carries over to the case of stochastic differential and difference equations in Kalman filtering. In fact, the state vector of the dynamical system signal model determines the order of the Kalman calculations. In this random case, the state can be defined as the minimum amount of information about past and present estimates needed to determine an optimal causal estimate of future response given future noisy observations. The dynamical model for this case is given by [5]

$$\boldsymbol{s}(m) = \boldsymbol{F}\boldsymbol{s}(m-1) + \boldsymbol{G}\boldsymbol{u}(m) \tag{1}$$

$$\boldsymbol{r}(m) = \boldsymbol{H}\boldsymbol{s}(m) + \boldsymbol{v}(m), \tag{2}$$

where $\boldsymbol{s}$ is the signal state vector, $\boldsymbol{r}$ is the observation or received signal, $\boldsymbol{v}$ is the observation noise, and $\boldsymbol{u}$ is the random process which generates the signal. The matrices $\boldsymbol{F}$, $\boldsymbol{G}$, and $\boldsymbol{H}$ are, respectively, the system, drive, and observation matrices.

In (1) and (2), $\boldsymbol{u}$ and $\boldsymbol{v}$ are uncorrelated white Gaussian zero mean noise sources with correlation

$$\boldsymbol{E}[\boldsymbol{u}(m)\boldsymbol{u}^T(k)] = \boldsymbol{Q}_u\delta_{mk}$$

and

$$\boldsymbol{E}[\boldsymbol{v}(m)\boldsymbol{v}^T(k)] = \boldsymbol{Q}_v\delta_{mk},$$

where $\boldsymbol{Q}_u$ and $\boldsymbol{Q}_v$ are correlation matrices and δ_{mk} is the Kronecker delta. The Kalman filtering equations for this model are [5]

$$\hat{\boldsymbol{s}}_b(m) = \boldsymbol{F}\hat{\boldsymbol{s}}_a(m-1) \tag{3}$$

$$\boldsymbol{P}_b(m) = \boldsymbol{F}\boldsymbol{P}_a(m-1)\boldsymbol{F}^T + \boldsymbol{G}\boldsymbol{Q}_u\boldsymbol{G}^T \tag{4}$$

$$\boldsymbol{K}(m) = \boldsymbol{P}_b(m)\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{P}_b(m)\boldsymbol{H}^T + \boldsymbol{Q}_v)^{-1} \tag{5}$$

$$\hat{\boldsymbol{s}}_a(m) = \hat{\boldsymbol{s}}_b(m) + \boldsymbol{K}(m)[\boldsymbol{r}(m) - \boldsymbol{H}\hat{\boldsymbol{s}}_b(m)] \tag{6}$$

Manuscript received May 3, 1976; revised October 25, 1976. This work was supported in part by the U.S. Energy Research and Development Administration under contract number W-7504-ENG-48 and in part by the Defense Advanced Research Projects Agency of the Department of Defense under ARPA Order No. 2436.

J. W. Woods was with Lawrence Livermore Laboratory, Livermore, CA. He is now with the Electrical and Systems Engineering Department, Rensselaer Polytechnic Institute, Troy, NY 12181.

C. H. Radewan is with the Lawrence Livermore Laboratory, Livermore, CA 94550.

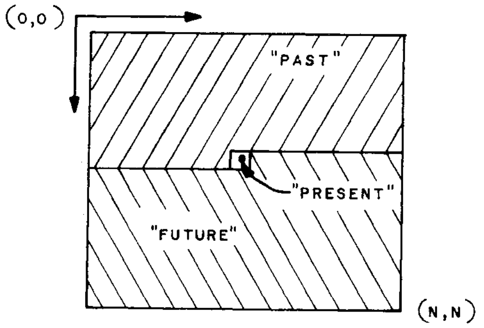

Fig. 1. Scalar scanning rectangular region using raster scan.

$$\boldsymbol{P}_a(m) = [\boldsymbol{I} - \boldsymbol{K}(m)\boldsymbol{H}]\boldsymbol{P}_b(m), \qquad (7)$$

where

$$\boldsymbol{P}_i(m) \triangleq \boldsymbol{E}[(\boldsymbol{s} - \hat{\boldsymbol{s}})_i(\boldsymbol{s} - \hat{\boldsymbol{s}})_i^T], \qquad i = a,b$$

is the error covariance matrix. The subscripts a and b indicate "after" and "before" updating, respectively.

These equations have the following simple interpretation. First, in (3), we project the last estimate forward using the dynamics of the system model. Then in (6) we update this estimate using the new observation $\boldsymbol{r}(m)$. Equation (5) is the gain matrix for the updating in (6). The remaining equations (4) and (7) are the error covariances necessary to calculate the new gain matrix.

II. Kalman Filtering in Two Dimensions

A) A Scalar 2-D Kalman Filter

When we try to extend the above formalism to two dimensions, we encounter a problem concerning the 1-*D* definition of state as we will now show. First, we consider a particular scanning of a discrete 2-*D* square region consisting of an $N \times N$ regularly spaced lattice. Since the scanning does not qualitatively affect the results, we will assume the raster scan, i.e., left to right, advance one line, then repeat. Continuing with this procedure for the next N-2 lines we complete the scan. Thus at any point in the picture, some points will be the "past," one point will be the "present," and the remaining points will be the "future." These words will thus have their conventional meaning with respect to the order in which the points are processed (see Fig. 1). Next, we generalize our dynamical model (1). We obtain, by application of a recently developed 2-*D* spectral factorization routine [6], the $(M \times M)$th order (see Appendix) scalar equation,[1]

$$s(m,n) = \sum_{\mathcal{R}_{\oplus +}} c_{kl} s(m-k,n-l) + u(m,n), \qquad (8)$$

where $\mathcal{R}_{\oplus +} = \{m \geq 0, n \geq 0\} \cup \{m < 0, n > 0\}$, which is to be compared to the scalar version of (1),

$$s(m) = \sum_{k=1}^{M} c_k s(m-k) + u(m). \qquad (9)$$

The model of (8) is called a nonsymmetric half-plane (NSHP) model [7].

Now we are in a position to appreciate the difficulty with state in two dimensions. For while (8) only uses $O(M^2)$ points for each computation of a new output point, so that the amount of computation is related to the order of the filter as in (9), when (8) moves across the image, values of s will be needed from the previous M lines. Thus the memory requirements of (8) are not of order related to the order of the filter as in (9). Because of this large memory requirement, the state vector for (8) must contain the M previous lines and hence has $O(MN)$ components.

It is convenient to take the part of the state vector on the support of the filter and give it a special name. Thus we define a new term *pseudo-state* as the minimum part of the state vector needed to compute the present output given the present input. This pseudo-state has a number of elements related to the order of the filter and hence its size determines the order of computation of the filter. However, the storage requirements are related to the size of the state vector. In the ordinary 1-*D* case, the pseudo-state vector is the state vector.

Next we will consider the Kalman filtering equations for this model. We will find that both the per point computation and the memory grow with N^2, the square of the width of the image field. Because of this unsatisfactory situation, we then turn to some approximate filters which greatly reduce the computation while only slightly affecting the accuracy of the estimate.

To obtain the Kalman filtering equations for the signal model of (8) and the observation equation

$$r(m,n) = s(m,n) + v(m,n), \qquad (10)$$

where u and v are independent white Gaussian sources, we note that the scanning operation transforms the 2-*D* problem into an equivalent 1-*D* problem. Thus (8) can be put into the form (1) by defining the state vector of $M(N + 1)$ components

$$\boldsymbol{s}(m,n) = [s(m,n), s(m-1,n), \cdots, s(1,n); s(N,n-1), \cdots, s(1,n-1); \cdots; s(N,n-M), \cdots, s(m-M,n-M)]^T.$$

With this transformation, we obtain

$$\boldsymbol{s}(m,n) = \boldsymbol{F}\boldsymbol{s}(m-1,n) + \boldsymbol{G}u(m,n) \qquad (11)$$

with corresponding observational model

$$r(m,n) = \boldsymbol{H}\boldsymbol{s}(m,n) + \boldsymbol{v}(m,n) \qquad (12)$$

which is identical in form to (1) and (2) with line by line scanning understood. Thus we could immediately write down the Kalman equations. They would be (3) to (7) with the above interpretation of the $\boldsymbol{s}$ vector. The difficulty with these equations is the amount of computation and memory requirements associated with them. For example, consider an $(M \times M)$th order system model as in (8) and an observation region consisting of an $N \times N$ square with $N \gg M$. Then the dimension of the matrix equations (3) to (8) is

[1] Inclusion of zeroes in the model leads to additional complications. To simplify the discussion, we consider only all-pole dynamical systems.

approximately MN. This means that in general the order of the computation is $O(M^3N^3)$. However, taking advantage of the spatial invariant structure of the signal model, this can be reduced to $O(M^2N^2)$ as will be shown below. For $M = 4$ and $N = 100$, we require on the order of 160 000 multiplies and adds per output point. The overall total computation for the 10 000 element picture would be $O(10^9)$. At 1 μs/operation, the computer time for such a calculation would be 10^3 s or approximately 20 min. In addition, the data storage problems are immense. To store $\boldsymbol{P}_{b,a}$ at each stage we need $O(MN)^2$ storage locations. For the above M and N, this is 160 000 words of storage to be accessed at each picture point! These numbers tell us that the exact 2-D Kalman scalar filter is computationally unmanageable, given today's state of the computer art. However, not only that, it is also wasteful of computation as will be pointed out below.

Next we consider the 2-D Kalman vector filter [8]. This is also an exact processor as is the scalar one mentioned above; however, the vector processor observes a line at a time instead of a point at a time.

B) *A Kalman Vector Filter*

In this formulation, we consider a vector scanning [8] consisting of a (horizontal) line at a time, as is necessary to do optimal Kalman vector filtering. We can write a recursive signal model

$$\boldsymbol{s}(n) = \boldsymbol{F}\boldsymbol{s}(n-1) + \boldsymbol{G}\boldsymbol{u}(n) \tag{13}$$

with vector observations

$$\boldsymbol{r}(n) = \boldsymbol{H}\boldsymbol{s}(n) + \boldsymbol{v}(n). \tag{14}$$

Here

$$\boldsymbol{s}(n) \triangleq [s(1,n), \cdots, s(N,n); s(1,n-1), \cdots, s(N,n-1); \cdots; s(1,n-M+1), \cdots, s(N,n-M+1)]^T$$

and

$$\boldsymbol{r}(n) = [r(1,n), \cdots, r(N,n)]^T.$$

Since (13) and (14) are of the form of (1) and (2), the Kalman equations are (3) thru (7) with the above interpretations for the $\boldsymbol{s}$ and $\boldsymbol{r}$ vectors. We note that these matrix equations are of order $O(MN)$ as in the 2-D Kalman scalar filter. However, each iteration of the vector filter yields N estimates (a whole line), so for the example given for the scalar filter, the vector filter computation times are reduced by a factor of N yielding an $O(M^3N^2)$ computation. The storage requirements are the same and hence this is again a computationally unmanageable solution.[2]

We will see below how splitting the picture up into partially overlapping strips can greatly reduce the computation of this vector filter. Further reductions in the order of the computation can be realized by using the special block form of the matrices $\boldsymbol{F}$, $\boldsymbol{G}$, and $\boldsymbol{H}$ in (13) and (14). These reductions in computation permit the practical implementation of the Kalman vector filter. As in the 2-D scalar case, we can construct the recursive dynamical model (13) by using a 2-D spectral factorization procedure [6].

III. New Developments

In this section, we will summarize new approximate methods for realizing the 2-D Kalman scalar and vector filters [10], [11]. While both these techniques provide 2 to 3 orders of magnitude reduction in computation and storage, it can be anticipated that the reduction in the optimality of the estimate will be only slight. The reason for this surprising result is that the cause of the excessive computations is "purely theoretical," that is, the very uncommon but possible occurrence of high correlation at great distances. We will first develop an approximate Kalman vector filter called the strip processor. Then we will present a scalar filter called the reduced update Kalman filter.

A) *A Kalman Strip Processor*

The Kalman strip filter arises from the following considerations. First, although the vector filter has a state vector whose dimension is the width of the picture as is required for theoretical optimality, experience suggests that very few images will have the large correlation at great distances to justify such computation. Rather, most images show significant correlation only over rather compact neighborhoods of a given point. Thus we consider processing strips of the picture independently. The strips are overlapped with only the middle points being used as the final estimate. If the strip width is greater than the correlation distance of the signal and noise field, then close to optimum performance can result for points near the center. Processing the picture in these strips can greatly reduce the order of the computation and storage.

Second, we observe that if the data are stationary in the horizontal direction, we can save computation and storage by the following scheme: take the strips in the vertical direction and note that the filters for the different strips will all be the same. Thus the Kalman covariance equations (4) and (7) and the gain equation (5) need be calculated only for the first strip. Then the corresponding line in each of the remaining strips may be updated via (6) using the same gain matrix. In this way, storage need be provided for the covariance and gain matrices only in the first strip. Thus these two simplifications allow us to construct a good approximation to the Kalman vector filter which is computationally manageable. We will now elaborate on the details of constructing such a Kalman strip filter. This will consist of two parts: first, a discussion of the model, second, a discussion of the filter for that model.

1) The Strip Dynamical Model: The recursive signal model is the same as given in (13) with vector observations given in (14) with the exception that the vectors now are

[2] Note that the term vector processor was used in [8] to refer to a processor less than a line wide. However, we will reserve this term for the full width processor.

of dimension MW, where W is the strip width ($W \ll N$). We start the model development from a general NSHP recursive model (see Appendix), and develop from it a vector recursive model,

$$\boldsymbol{s}(n) = \boldsymbol{F}\boldsymbol{s}(n-1) + \boldsymbol{G}\boldsymbol{u}(n) \tag{15}$$

$$\boldsymbol{r}(n) = \boldsymbol{H}\boldsymbol{s}(n) + \mathbf{v}(n), \tag{16}$$

where

$$\boldsymbol{s}(n) \triangleq [s(1,n),\cdots,s(W,n);s(1,n-1),\cdots,s(W,n-1); \cdots;s(1,n-M+1),\cdots,s(W,n-M+1)]^T$$

and

$$\boldsymbol{r}(n) \triangleq [r(1,n),\cdots,r(W,n)]^T.$$

To obtain $r = s + v$, choose

$$\boldsymbol{H} = [I:0:\cdots:0]^T.$$

Now $\boldsymbol{F}$, $\boldsymbol{G}$, and $\boldsymbol{Q}_u$ must be chosen so that s approximately satisfies (8). The conversion from (8) to (15) is most easily seen in the Z-transform domain. Taking the Z-transform of (8), we see that the system function transforming s into u is

$$\frac{U(z_1,z_2)}{S(z_1,z_2)} = 1 - C(z_1,z_2), \tag{17}$$

where

$$C(z_1,z_2) = \sum_{\mathcal{R}_{\oplus+}} c_{pq} z_1^{-p} z_2^{-q}. \tag{18}$$

Equation (17) can be factored as

$$1 - C(z_1,z_2) = (1 - C(z_1,\infty))(1 - C^{(1)}(z_1,z_2)), \tag{19}$$

where

$$C(z_1,\infty) = \sum_{p>0} c_{p0} z_1^{-p},$$

and where $C^{(1)}(z_1,z_2)$ contains only $q > 0$ terms and is generally of infinite order in z_1^{-1}.

Now $C^{(1)}$ can be approximately represented by $\boldsymbol{F}$ since it represents an operation on the previous rows. The first factor on the right side of (19) can be inverted and represented by $\boldsymbol{G}$. In this way we arrive at the following form for $\boldsymbol{F}$:

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \cdots & \boldsymbol{A}_M \\ \boldsymbol{I} & \boldsymbol{O} & \cdots & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I} & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \cdots & \boldsymbol{O}\ \boldsymbol{I} & \boldsymbol{O} \end{bmatrix}, \tag{20}$$

which is $MW \times MW$ with $W \times W$ blocks $\boldsymbol{I}$ and $\boldsymbol{A}_m$; $m = 1,\cdots,M$. Here

$$\boldsymbol{A}_m = [\boldsymbol{I} - \boldsymbol{C}_0]^{-1}\boldsymbol{C}_m \tag{21}$$

where the $\boldsymbol{C}_m$ are given by

$$(\boldsymbol{C}_m)_{ij} = c_{m,i-j}. \tag{22}$$

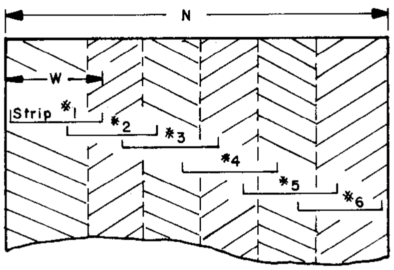

Fig. 2. Kalman strip processing example (dotted lines indicate estimate boundaries).

In like manner, we could set

$$\boldsymbol{G} = [(\boldsymbol{I} - \boldsymbol{C}_0)^{-1}:\boldsymbol{O}:\boldsymbol{O}:\boldsymbol{O}:\cdots:\boldsymbol{O}]^T.$$

However, the edge effects due to the small strip width W would be expected to cause significant errors. To lessen this problem, we choose $\boldsymbol{G}$ to make the correlations of models (15) and (8) agree exactly in the current row. This procedure leaves uncorrected slight errors in the correlations with the $(M-1)$ previous rows in the model. Since the errors are both small and concentrated near the strip edges, they are expected to be insignificant in effect on the "saved" estimates in the middle section of the strip.

2) *The Kalman Strip Filter:* The Kalman equations for this model are

$$\hat{\boldsymbol{s}}_b^i(n) = \boldsymbol{F}\hat{\boldsymbol{s}}_a^i(n-1) \tag{23}$$

$$\hat{\boldsymbol{s}}_a^i(n) = \hat{\boldsymbol{s}}_b^i(n) + \boldsymbol{K}^1(n)[\boldsymbol{r}^i(n) - \boldsymbol{H}\hat{\boldsymbol{s}}_b^i(n)], \tag{24}$$

where i is the number of the strip. In addition, to calculate the gain matrix for the nth line we must compute

$$\boldsymbol{P}_b^1(n) = \boldsymbol{F}\boldsymbol{P}_a^1(n-1)\boldsymbol{F}^T + \boldsymbol{G}\boldsymbol{Q}_u\boldsymbol{G}^T \tag{25}$$

$$\boldsymbol{K}^1(n) = \boldsymbol{P}_b^1(n)H^T[\boldsymbol{H}\boldsymbol{P}_b^1(n)\boldsymbol{H}^T + \boldsymbol{Q}_v]^{-1} \tag{26}$$

$$\boldsymbol{P}_a^1(n) = [\boldsymbol{I} - \boldsymbol{K}^1(n)\boldsymbol{H}]\boldsymbol{P}_b^1(n). \tag{27}$$

We note that for the horizontally stationary case, (25) through (27) need only be computed once for each line, while the estimates (23) and (24) are computed for all the strips at the nth line before proceeding to line $(n+1)$. Initial conditions are needed for (23) and (25) at the first line. They are given by

$$\boldsymbol{s}_b^i(1) = \boldsymbol{E}[\boldsymbol{s}^i(1)] = \boldsymbol{o} \tag{28}$$

and

$$\boldsymbol{P}_b^1(1) = \boldsymbol{E}[\boldsymbol{s}^i(1)\boldsymbol{s}^{iT}(1)] = \boldsymbol{R}_s, \tag{29}$$

where $\boldsymbol{R}_s$ is the $MW \times MW$ correlation matrix of the model.

Fig. 2 shows a diagram for strip processing where the middle third of the elements are retained for the final estimate of s. The overlap would require a factor of three in computation. In the scheme used in the example of Section IV, a width of $W = 25$ is used with retention of the middle 15 elements. In this case, the "extra" computation due to the overlap is reduced to a factor of two. The amount of "extra" computation can be traded off for uniformity in the estimation error.

3) Optimization of Strip Width: To minimize the amount of computation for the completely nonstationary case, we can choose the number of retained elements N_R and the strip width W consistent with obtaining near optimum performance. To keep the saved points at least D from the strip edge, where D is the "correlation distance" of the signal s, we get

$$W = 2D + N_R.$$

The number of computations per point is proportional to

$$\frac{(2D + N_R)^3}{N_R}, \qquad \text{for } N \gg 2D.$$

The minimum occurs at

$$N_R^o = D \tag{30}$$

which yields

$$W^o = 3D \tag{31}$$

to minimize computation.

When the data are stationary horizontally, then the number of computations is asymptotically

$$\frac{(2D + N_R)^2}{N_R}, \qquad \text{for } N \gg 2D.$$

The minimum turns out to be

$$N_R^o = 2D$$

which yields

$$W^o = 4D. \tag{32}$$

B) Reduced Update Kalman Filtering

In this section, we will treat an approximation to the 2-D Kalman scalar processor presented in Section II-A. This involves the dynamical model (8) and the scalar observation (10). As indicated earlier, the state vector for this filter is $O(MN)$ dimensional, where M is the order of the recursive model and N is the width of the picture. Thus all these points must be updated in an equation equivalent to (6). The main concept of the reduced update filter is that the Kalman equations are composed of two steps: a prediction part and an update part. Now the prediction part, as set forth in (3), is a computationally straightforward projection of $O(M^2)$ previous estimates. However, the update part involves calculations involving each of the $O(MN)$ random variables in the state vector. Since $N \gg M$, we can reduce the bulk of the computation by reducing the update process. Thus we choose to update only those elements of the state vector within a certain distance of the point currently being processed, (m,n). We can expect this procedure to result in a good approximation because significant updates will be confined to a region around the observation at (m,n). Therefore, omitting the update of points far away should only minimally impact performance. For convenience of notation, we take this region to be the support of the NSHP recursive signal model in (8). Hence, there are only $O(M^2)$ points to update, i.e., we only update the pseudo-state vector for each observation. In the remainder of this section, we will derive the equations for the optimal updating of the pseudo-state vector. We will see that these equations can point the way to further simplifications which are expected again to result in only slightly suboptimum performance.

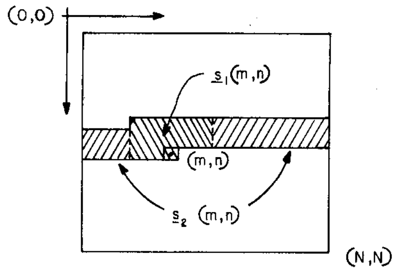

Fig. 3. Assignment of points to partitioned state vector.

Let the signal model be generated by

$$s(m,n) = \sum_{\mathcal{R}_{\oplus+}} c_{kl} s(m-k, n-l) + w(m,n), \tag{33}$$

where $\{w(m,n)\}$ is a zero mean homogeneous white Gaussian source with variance σ_w^2. The received signal is

$$r(m,n) = s(m,n) + v(m,n), \tag{34}$$

where $\{v(m,n)\}$ is zero mean homogeneous white Gaussian noise, independent of $\{w(m,n)\}$, with variance σ_v^2.

Let the field $\{r(m,n)\}$ be observed over an $N \times N$ rectangular region. Next, introduce a vector notation for the pseudo-state vector corresponding to a scalar line by line scan, viz.,

$$\begin{aligned} \boldsymbol{s}_1(m,n) \triangleq [&s(m,n), \cdots, s(m-M,n); s(m+M,n-1), \\ &\cdots, s(m-M,n-1); \\ &\cdots; s(m+M,n-M), \cdots, s(m-M,n-M)]^T. \end{aligned} \tag{35}$$

Order the remaining points of the state vector onto $\boldsymbol{s}_2(m,n)$. Thus the resulting assignment of points is as shown in Fig. 3. With this convention the state vector can be written

$$\boldsymbol{s}(m,n) = (\boldsymbol{s}_1^T(m,n), \boldsymbol{s}_2^T(m,n))^T. \tag{36}$$

The state dynamical model can then be written

$$\boldsymbol{s}(m,n) = \boldsymbol{C}\boldsymbol{s}(m-1,n) + \boldsymbol{w}(m,n), \tag{37}$$

where $\boldsymbol{C}$ is the system propagation matrix determined by $\{c_{kl}\}$ and by the ordering of the state vector $\boldsymbol{s}(m,n)$. Note that (37) holds for all the points (m,n), except near the boundaries where boundary conditions must be incorporated. The drive vector is given as

$$\boldsymbol{w}(m,n) \triangleq (w(m,n), 0, \cdots, 0)^T. \tag{38}$$

The scalar observation equation is

$$r(m,n) = \boldsymbol{H}\boldsymbol{s}(m,n) + v(m,n), \tag{39}$$

where

$$\boldsymbol{H} = (1, 0, \cdots, 0). \tag{40}$$

We can partition the matrix C similarly to $\boldsymbol{s}$ as

$$\boldsymbol{C} = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}. \tag{41}$$

It then turns out that $\boldsymbol{C}_{11}$ and $\boldsymbol{C}_{12}$ contain all the $\{c_{kl}\}$ terms, and the remainder of $\boldsymbol{C}$ constitutes a shift transformation. Thus (37) may be rewritten as

$$\boldsymbol{s}_1(m,n) = \boldsymbol{C}_{11}\boldsymbol{s}_1(m-1,n) + \boldsymbol{w}_1(m,n) + \boldsymbol{C}_{12}\boldsymbol{s}_2(m-1,n), \tag{42}$$

where $\boldsymbol{w}$ has been partitioned similarly to $\boldsymbol{s}$. Equation (42) focuses on the computation to be performed at (m,n). That is, $\boldsymbol{s}_2$ requires only the shifting of previously computed values. We can similarly partition $\boldsymbol{H} = (\boldsymbol{H}_1,\boldsymbol{H}_2)$ with $\boldsymbol{H}_2 = \boldsymbol{O}$ to get a new observation equation

$$r(m,n) = \boldsymbol{H}_1\boldsymbol{s}_1(m,n) + v(m,n). \tag{43}$$

1) Derivation of Reduced Update Filter: Assume that the received array $\{r(m,n)\}$ is scanned in line by line fashion. Then the Kalman filtering equations can be written immediately from (3) through (7) as discussed in Section I.

a) Extrapolation:

$$m \rightarrow m+1$$

$$\boldsymbol{P}_b(m,n) = \boldsymbol{C}\boldsymbol{P}_a(m-1,n)\boldsymbol{C}^T + \boldsymbol{G}\boldsymbol{Q}_w\boldsymbol{G}^T \tag{44}$$

$$\hat{\boldsymbol{s}}_b(m,n) = \boldsymbol{C}\hat{\boldsymbol{s}}_a(m-1,n) \tag{45}$$

b) Update:

$$\boldsymbol{K}(m,n) = \boldsymbol{P}_b(m,n)\boldsymbol{H}^T[\boldsymbol{H}\boldsymbol{P}_b(m,n)\boldsymbol{H}^T + \sigma_v^2]^{-1} \tag{46}$$

$$\hat{\boldsymbol{s}}_a(m,n) = \hat{\boldsymbol{s}}_b(m,n) + \boldsymbol{K}(m,n)[r(m,n) - \boldsymbol{H}\hat{\boldsymbol{s}}_b(m,n)] \tag{47}$$

$$\boldsymbol{P}_a(m) = [\boldsymbol{I} - \boldsymbol{K}(m,n)\boldsymbol{H}]\boldsymbol{P}_b(m,n). \tag{48}$$

If an arbitrary gain matrix is used instead of the one prescribed in (46), the covariance $\boldsymbol{P}_a$ given in (48) changes to [5]

$$\boldsymbol{P}_a(m,n) = (\boldsymbol{I} - \boldsymbol{K}(m,n)\boldsymbol{H})\boldsymbol{P}_b(m,n)(\boldsymbol{I} - \boldsymbol{K}(m,n)\boldsymbol{H})^T + \boldsymbol{K}(m,n)\sigma_v^2\boldsymbol{K}^T(m,n). \tag{49}$$

Let $\boldsymbol{K}(m,n)$ have the form

$$\boldsymbol{K}(m,n) = \left[\begin{array}{c} \boldsymbol{K}_1(m,n) \\ \hdashline \boldsymbol{O} \end{array}\right], \tag{50}$$

where the partitioning corresponds to that of $\boldsymbol{s}(m,n)$. We now choose $\boldsymbol{K}_1(m,n)$ to minimize the trace of $\boldsymbol{P}_a(m,n)$ as given in (49) for $\boldsymbol{K}(m,n)$ as given in (50). The result is the reduced update Kalman filter.

c) Extrapolation:

$$m \rightarrow m+1$$

$$\boldsymbol{P}_b(m,n) = \boldsymbol{C}\boldsymbol{P}_a(m-1,n)\boldsymbol{C}^T + \boldsymbol{G}\boldsymbol{Q}_w\boldsymbol{G}^t \tag{51}$$

$$\hat{\boldsymbol{s}}_{1b}(m,n) = \boldsymbol{C}_{11}\hat{\boldsymbol{s}}_{1a}(m-1,n) + \boldsymbol{C}_{12}\hat{\boldsymbol{s}}_{2a}(m-1,n). \tag{52}$$

d) Update:

$$\boldsymbol{K}_1(m,n) = \boldsymbol{P}_{11,b}(m,n)\boldsymbol{H}_1^T(\boldsymbol{H}_1\boldsymbol{P}_{11,b}(m,n)\boldsymbol{H}_1^T + \sigma_v^2)^{-1} \tag{53}$$

$$\hat{\boldsymbol{s}}_{1a}(m,n) = \hat{\boldsymbol{s}}_{1b}(m,n) + \boldsymbol{K}_1(m,n)[r(m,n) - \boldsymbol{H}_1\hat{\boldsymbol{s}}_{1b}(m,n)] \tag{54}$$

$$\boldsymbol{P}_{11,a}(m,n) = [\boldsymbol{I} - \boldsymbol{K}_1(m,n)\boldsymbol{H}_1]\boldsymbol{P}_{11,b}(m,n) \tag{55a}$$

$$\boldsymbol{P}_{12,a}(m,n) = [\boldsymbol{I} - \boldsymbol{K}_1(m,n)\boldsymbol{H}_1]\boldsymbol{P}_{12,b}(m,n), \tag{55b}$$

where $\boldsymbol{P}_a$ and $\boldsymbol{P}_b$ have been partitioned similarly to $\boldsymbol{s}$.

Equations (51)–(55) can provide great computational savings over the standard Kalman filtering equations. To understand these equations better, it is helpful to convert them back to scalar notation. First, we note that (52) will become, in scalar notation,

$$\hat{s}_b^{(m,n)}(m,n) = \sum_{\mathcal{R}_{\oplus+}} c_{kl}\hat{s}_a^{(m-1,n)}(m-k,n-l), \tag{56}$$

since it represents propagation of the previous estimates through the dynamics of the system. In these scalar equations, the superscript indicates the step in the filtering, while the argument represents the position of the data. Equation (51) represents the error in this predicted estimate; thus it becomes

$$R_b^{(m,n)}(m,n;k,l) = \sum_{op} c_{op}R_a^{(m-1,n)}(m-o,n-p;k,l), \qquad (k,l) \in \mathcal{S}_{\oplus+}^{(m,n)} \tag{57}$$

$$R_b^{(m,n)}(m,n;m,n) = \sum_{kl} c_{kl}R_b^{(m,n)}(m,n;m-k,n-l) + \sigma_w^2, \tag{58}$$

where $\mathcal{S}_{\oplus+}^{(m,n)}$ is the support of the state vector $\boldsymbol{s}(m,n)$.

Equation (53) computes a $\boldsymbol{K}_1(m,n)$ which has the same support as $\boldsymbol{s}_1(m,n)$, namely, $\mathcal{R}_{\oplus+}^{(m,n)}$, the support of the pseudo-state vector. The scalar equation identical to (53) is

$$K^{(m,n)}(i,j) = \frac{R_b^{(m,n)}(m,n;i,j)}{R_b^{(m,n)}(m,n;m,n) + \sigma_v^2}, \qquad (i,j) \in \mathcal{R}_{\oplus+}^{(m,n)}. \tag{59}$$

Similarly, (54) becomes

$$\hat{s}_a^{(m,n)}(i,j) = \hat{s}_b^{(m,n)}(i,j) + K^{(m,n)}(m-i,n-j)[r(m,n) - \hat{s}_b^{(m,n)}(m,n)], \qquad (i,j) \in \mathcal{R}_{\oplus+}^{(m,n)}. \tag{60}$$

Finally, (55a) and (55b) both are expressible as the set of scalar equations

$$R_a^{(m,n)}(i,j;k,l) = R_b^{(m,n)}(i,j;k,l) - K^{(m,n)}(m-i,n-j)R_b^{(m,n)}(m,n;k,l), \quad \text{for } (i,j) \in \mathcal{R}_{\oplus+}^{(m,n)};\ (k,l) \in \mathcal{S}_{\oplus+}^{(m,n)}. \tag{61}$$

The reduced update Kalman filter comprising (56) to (61) has been derived as an optimal approximation to the 2-*D* Kalman scalar filter. The prediction part of the Kalman filter is left unchanged, and the update is optimized under the constraint of updating only the nearest previously processed neighbors. It is thus perhaps not clear that the resulting reduced update filter is overall optimal in the class of 2-*D* recursive filters of the same order as itself because the prediction part of the Kalman filter was

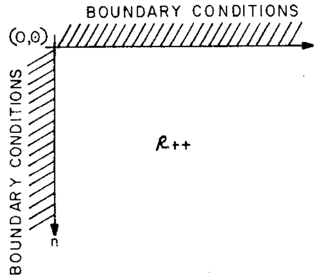

Fig. 4. Data region to be filtered by reduced update equations.

held fixed in the above derivation. However, it turns out that this procedure is indeed overall optimal as will now be seen.

Theorem: The reduced update Kalman filter is optimal under its constraint, i.e., it minimizes the mse over the class of spatially varying, linear, 2-*D* recursive filters of similar order.

Proof: Consider filtering the region $\mathcal{R}_{\oplus+}$ with boundary conditions (bc) on the upper and left edges as shown in Fig. 4. The method of proof will be induction. The signal model is given by (33) with observations given by (34). For the first point $(m,n) = 0$, obviously the best prediction will be

$$\hat{s}_b^{(0,0)}(0,0) = \sum_{\mathcal{R}_{\oplus+}} c_{kl} s^{(\mathrm{bc})}(-k,-l). \tag{62}$$

If this is followed by the optimal constrained update, we obtain

$$\hat{s}_a^{(0,0)}(0,0) = \hat{s}_{\mathrm{mmse}}(0,0),$$

the best mse estimate subject to constraint. Now assume we have the best constrained mse estimate at (m,n). Then the best prediction based on the pseudo-state is

$$\hat{s}_b^{(m+1,n)}(m+1,n) = \sum c_{kl}\hat{s}_a^{(m,n)}(m+1-k,n-l), \tag{63}$$

as can be seen by applying to (33) the conditional expectation operator $E[\cdot \mid \hat{s}_1(m,n)]$. Finally, the optimal constrained update generates the optimal constrained mse estimate at $(m+1,n)$. This completes the proof.

From this theorem follows the very important fact that for homogeneous data the Kalman reduced update filter will converge to an optimal 2-*D* NSHP recursive filter for estimating the data. Now, in one dimension, such a filter could be obtained via Wiener's spectral design procedure [9]. However, the same procedure in the 2-*D* case leads to an infinite-order filter, thus the optimal finite-order filter is not obtained. Hence, the reduced update Kalman filter is also particularly attractive from the standpoint of design of spatially invariant filters for homogeneous random fields.

2) *Order of Computation:* Next we investigate the order of computation of the reduced update Kalman filter. We will consider each of (56) thru (61) separately. First, (56) will be $O(M^2)$ for an Mth order NSHP model. Equation (56) will thus be $O(M^2)$ for each (k,l). Since there are $O(MN)$ points in $\mathcal{S}_{\oplus+}^{(m,n)}$, we obtain $O(M^3N)$ for (57). Equation (58) is $O(M^2)$, as are (58) and (60). Equation (61) is simple, but has to be computed for each pair (i,j) and (k,l) with $(k,l) \in \mathcal{S}_{\oplus+}^{(m,n)}$, thus giving a computational total of $O(M^3N)$ as in (57). Summing up, we get the overall total computation per point as $O(M^3N)$. This is to be compared with $O(M^3N^3)$ for the general Kalman scalar filter and $O(M^2N^2)$ for the $(M \times M)$th order half-plane filter model.

The overall savings of a factor N^2 results from two simplifications. First, the reduced update has reduced the orders of the matrices from $MN \times MN$ to $M^2 \times MN$. Second, for the $(M \times M)$th order filter model, the scalar equations (56) to (61) write only the newly computed values at point (m,n). Equation (51), for example, contains many more error covariance values than (57) and (58). However, only the first row (and column) of $\boldsymbol{P}_b(m,n)$ actually changes. The other elements simply get shifted. The scalar equations were written with respect to a fixed reference, the origin in the data plane. Thus no shifting appears in that notation. Another way of looking at this is as follows. The $\boldsymbol{C}$ matrix is composed of zeroes and ones except for the first row. These zeroes and ones simply serve to shift the data in the state vector, one place down as each new element is put in. This computation can be avoided by using indirect addressing and simply writing the new element over the oldest element in storage. Then a pointer or indirect address can be used to keep track of the "head" of the vector. This is essentially the reason for not counting the shifting operations, as they can be avoided with minimal computational effort on many machines.

For the reasons outlined in the previous paragraph, (56) thru (61) not only provide a convenient way to perceive the two-dimensional nature of the reduced update filter, but also present its essential computational aspects. The main computation was observed to be in (57) and (61). Equation (56) computes the error covariance between the "predicted point" (m,n) and the previous estimates in the filters state vector. Experience suggests this will be peaked at the point (m,n) with rapid decay with distance from (m,n). Thus it is a reasonable approximation to compute (57) only in a fixed size region including $\mathcal{R}_{\oplus+}^{(m,n)}$. This reasoning can also be applied to (61), where (k,l) would be restricted to a region significantly smaller than $\mathcal{S}_{\oplus+}^{(m,n)}$ and of fixed size for increasing N. Calling such a region $\mathcal{T}_{\oplus+}^{(m,n)}$ we can rewrite (57) and (61) as

$$R_b^{(m,n)}(m,n;k,l) = \sum c_{op} R_a^{(m-1,n)}(m-o,n-p;k,l), \qquad (k,l) \in \mathcal{T}_{\oplus+}^{(m,n)} \tag{64}$$

$$\begin{aligned} R_a^{(m,n)}(i,j;k,l) &= R_b^{(m,n)}(i,j;k,l) \\ &\quad - K^{(m,n)}(m-i,n-j)R_b^{(m,n)}(m,n;k,l), \\ &\quad \text{for } (i,j) \in \mathcal{R}_{\oplus+}^{(m,n)};\ (k,l) \in \mathcal{T}_{\oplus+}^{(m,n)}. \end{aligned} \tag{65}$$

These approximate reduced update equations reduce the order of the computation to $O(M^4)$, a constant with respect to N. Fig. 5 sets forth the envisioned region assignment. As mentioned above, the adverse effects of substituting $\mathcal{T}$

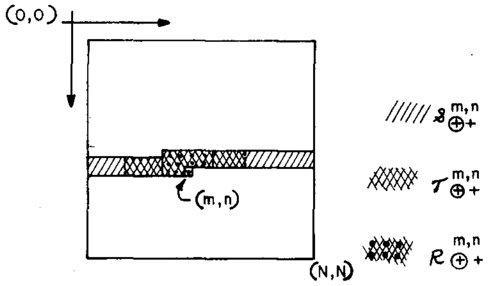
(0,0)
(m,n)
(N,N)
$\mathcal{S}_{\oplus+}^{m,n}$
$\mathcal{T}_{\oplus+}^{m,n}$
$\mathcal{R}_{\oplus+}^{m,n}$

Fig. 5. Region assignment for $\mathcal{T}_{\oplus+}^{(m,n)}$, covariance update region.

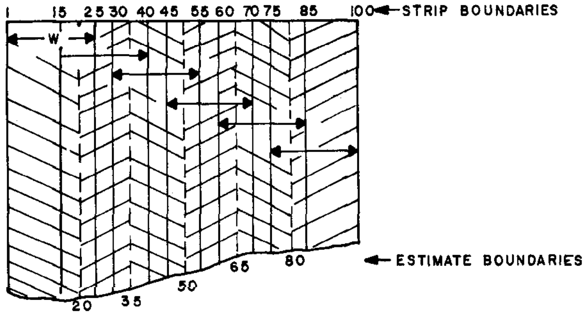

Fig. 6. Strip arrangement detail.

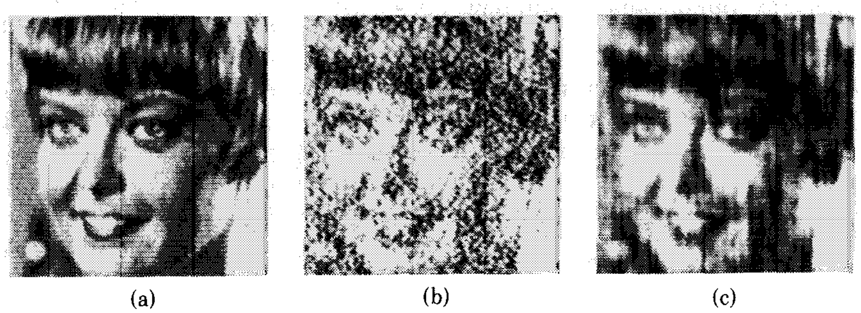
(a) (b) (c)

Fig. 7. Strip processing example. (a) Original. (b) Noisy (SNR = 1). (c) Estimate.

for $\mathcal{S}$ are expected to be minimal for most pictures. This is so both for the reasons mentioned previously and also because (59), the Kalman gain calculation does not directly make use of points outside $\mathcal{R}_{\oplus+}^{(m,n)}$. Thus the effect of this "truncation" would seem to be at worst second-order.

If we look at memory requirements, we find that they are dominated by the need to store the $\boldsymbol{R}_a$ error covariances. For the 2-D Kalman scalar filter, this storage will be (M^2N^2). For the reduced update filter we get the same amount of storage, only a portion of which is accessed at every point. For the approximate reduced update filter, we obtain $O(M^3N)$. A little thought reveals that this is the minimal possible dependence on N, i.e., linear, for a spatially varying processor.[3] Thus we cannot hope for further improvements here. However, for homogeneous regions, these storage requirements can be greatly relaxed. Then one only has to run the processor over a much smaller region to obtain near convergence to the steady-state filter. Subsequently, only (56) and (60) need be computed at a substantially lower amount of computational effort.

IV. Experimental Results

A noisy picture was processed with the Kalman strip filter. The signal was a 100×100 element center segment from a standard SMPTE test picture. The 8-bit original had an estimate of its mean removed prior to processing. The picture was then scaled so that its variance equaled one. White Gaussian noise was added at unit variance to produce a SNR of unity. The original with mean removed was also used to generate a (10×4)th order $+\oplus$ NHSP model [6], [7]. (See Appendix). This turns out to be a model for the transposed pictures. Thus the noisy image was transposed and inserted into the Kalman strip filter program with the above NSHP model. The strip arrangement is shown in detail in Fig. 6. There are six strips, each 25-points wide, with five-element overlap on each side for the interior strips. The middle 15 elements of the strips cover the picture and provide a fairly uniform error covariance. The mean value was added back and the estimate images transposed for presentation.

Fig. 7(a) shows the original image. Fig. 7(b) shows the noisy image. Fig. 7(c) shows the strip filter estimate. The measured SNR improvement was 7.7 dB. This compares favorably with previous results. As the model and noise were stationary, convergence was obtained; in this case, in 10 lines. The computer time was under 15 s on a CDC 7600 computer. The error covariance values were within 3 percent of their minimum value over the 15-element middle region of the strip. On the strip edge, it was as much as 60 percent above the minimum error variance value which was near the center.

The noisy picture of Fig. 7(b) was also modeled with a (3×3)th order $\oplus+$ NSHP factor and processed by the approximate reduced update Kalman filter for various sizes of $\mathcal{T}_{\oplus+}$. In particular, region half-widths of 3, 5, 7, and 9 were tried. The results were approximately the same in all cases, indicating both that the error decorrelates fast for our model and that the filter is fairly robust with respect to this type of error. The measured SNR improvement was 8.3 dB with output shown as Fig. 8. Convergence

[3] To see this, note that the processor must minimally have access to the error variances for the entire previous line.

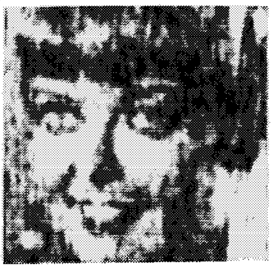

Fig. 8. Reduced update estimate.

was obtained in 10 lines or less with a run time of 40 s. This run time could be greatly shortened if the filter were updated only for the first 10 columns in the first 10 lines instead of the entire first 10 lines. The error covariance values of a sampling of the columns indicate that such a change in the algorithm would have negligible effect on the final result. This is because the chosen covariance model, as is typical of images, does not show substantial correlation over distances on the order of ten or more pixels.

V. Conclusions

The Kalman strip filter was introduced as a new approximation to the Kalman vector filter which processes a line at a time. The new filter is not limited to separable correlation functions and provides nearly uniform estimation error over the image by incorporating strip overlap. An optimum strip width was chosen to minimize computation. The reduced update Kalman filter was introduced and shown to be optimum in that it minimizes the post update mse under the constraint of updating only the nearby previously processed neighbors. It was shown that for all-pole data models, the Kalman reduced update filter coverges to an optimum 2-D recursive filter in the homogeneous case.

Appendix

Spectral Factorization in Two Dimensions [6]

In this Appendix, we review some of the properties of two-dimensional (2-D) spectral factorization that are pertinent to the development of 2-D recursive models. Since our recursive model consists of a recursive filtering of spatially white Gaussian noise, the problem can be thought of as a filter design problem.

In [7] it is shown that under mild conditions, one may factor a 2-D spectrum into nonsymmetric half-plane (NSHP) factors. The support of the filter numerator a and denominator b is restricted to a nonsymmetric half-plane, that is, a half-plane with the negative axis removed, as shown in Fig. 9. This removal allows the filter to be recursible. Fig. 10 shows how the NSHP support of b permits the 2-D recursion from left to right and top to bottom. This recursion can be accomplished precisely, because only previously computed points are used in the computation of the present output.

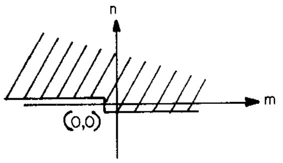

Fig. 9. Nonsymmetric half-plane.

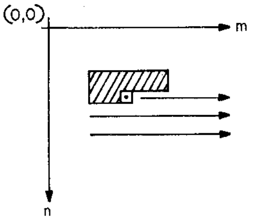

Fig. 10. Recursion of filter with NSHP support.

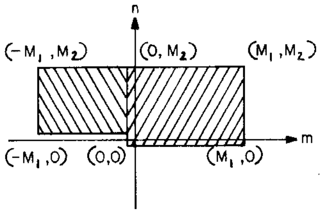

Fig. 11. Diagram illustrating concept of order of NSHP filter.

By the order of an NSHP filter, we mean the following: let the support of b and a be given by the union of an $(M_1 + 1) \times (M_2 + 1)$ square and an $M_1 \times M_2$ square situated as shown in Fig. 11. We will agree to call such a filter an $(M_1 \times M_2)$th-order NSHP filter. This notation is then indicative of the highest powers of z_1, z_2, z_1^{-1}, and z_2^{-1} in the filter's 2-D Z-transform

$$B(z_1,z_2) = \sum_{m=0}^{M_1} \sum_{n=0}^{M_2} b_{mn} z_1^{-m} z_2^{-n} + \sum_{m=-1}^{-M_1} \sum_{n=-1}^{-M_2} b_{mn} z_1^{-m} z_2^{-n}. \tag{A1}$$

As mentioned above, these recursive factors arise from a spectral factorization, in this case, that of the signal spectrum S_s. In general, the exact factors of S_s will be infinite-order, however, finite-order approximations can result in a specified small error. We can obtain such an approximation, in the simplest case, by simply truncating the factors to finite support. Thus we can solve the following rational approximation problem.

Given $\epsilon > 0$, choose the NSHP filter order $M_1 \times M_2$ high enough so that

$$\left|\left| S_s(u,v) - \left|\frac{A(e^{ju},e^{jv})}{B(e^{ju},e^{jv})}\right|^2 \right|\right| < \epsilon, \tag{A2}$$

where A and B are the 2-D Z-transforms of the filter numerator a and denominator b, respectively, and where $||\cdot||$ is a suitable functional norm, e.g., L^1, L^2, L^∞, etc., and most importantly where the resulting 2-D recursive filter is *stable*. In [6], a window method is advanced as a simple way to design these filters. This method was used to obtain the models in this paper. More elaborate and efficient design methods are presently under investigation.

References

[1] A. Habibi, "Two-dimensional Bayesian estimate of images," *Proc. IEEE*, vol. 60, p. 878–883, July 1972.

[2] N. E. Nahi, "Role of the recursive estimation in statistical image enhancement," *Proc. IEEE*, vol. 60, p. 872–877, July 1972.

[3] S. R. Powell and L. M. Silverman, "Modeling of two-dimensional covariance functions with application to image restoration," *IEEE Trans. Auto. Control*, vol. AC-19, p. 8–13, Feb. 1974.

[4] P. Padulo and M. A. Arbib, *System Theory.* Philadelphia, PA: W. B. Saunders, 1974, p. 21.

[5] A. P. Sage and J. L. Melsa, *Estimation Theory with Applications to Communications and Control.* New York: McGraw-Hill, 1971, p. 89–90.

[6] M. P. Ekstrom and J. W. Woods, "Two-dimensional spectral factorization with applications in recursive digital filtering," *IEEE Trans. Acoust., Speech and Signal Proc.*, vol. ASSP-24, pp. 115–128, Apr. 1976.

[7] J. W. Woods and M. P. Ekstrom, "Nonsymmetric half-plane recursive filters—characterization, stability theory and test," in *Proceedings 1975 Int'l Sympos. on Circuits and Systems*, Newton, MA, Apr. 1975, p. 447–450.

[8] N. E. Nahi and C. A. Franco, "Recursive image enhancement by vector scanning," *IEEE Trans. Commun.*, vol. C-21, pp. 305–311, April 1973.

[9] N. Wiener, *The Extrapolation, Interpolation, and Smoothing of Stationary Time Series.* New York: Wiley, 1949.

[10] J. W. Woods and C. H. Radewan, "The Kalman strip filter—a two-dimensional recursive vector processor," in *Proc. 9th Asilomar Conference Circuits, Sys. & Comp.*, Pacific Grove, CA, Nov. 1975.

[11] J. W. Woods and C. H. Radewan, "Reduced update Kalman filter—a two-dimensional recursive processor," presented at *Johns Hopkins Conf. on Inform. Sci. & Sys.*, Baltimore, MD, Apr. 1976.

Approximate Likelihood Ratio Detectors for Linear Processes

LESTER F. EASTWOOD, JR., MEMBER, IEEE, AND ROBERT LUGANNANI, SENIOR MEMBER, IEEE

***Abstract*—Approximately optimum detectors for problems in which either the signal or noise is a linear process are developed. An approximate expression for the characteristic function of a class of linear processes is inverted to obtain a convergent series approximation for the joint probability density. From this series, a detection algorithm is designed that is optimum when applied to the series and approximately optimum when applied to the original process. This technique is applied and evaluated in several cases that model coherent or instantaneous amplitude signal detection in reverberation noise, multipath transmission, and similar problems.**

I. Introduction

THE PROBLEM of optimum detection of signals in noise has been solved completely only under stringent assumptions. Many investigators assume that the noise and the signal, if it is also random, are Gaussian. However, in some important practical cases, the Gaussian assumption is invalid [1]–[4], and using the Gaussian optimum detector risks unexpectedly poor performance.

Though nonparametric detectors eliminate this risk, we still seek detectors that are optimum for non-Gaussian problems. One motivation for this effort is that the optimal performance is a standard of comparison for that of simply implemented suboptimum designs. Moreover, there may be problems for which only optimum performance is acceptable.

The most general signal detection problem, the detection of a stochastic signal in noise, has been extensively studied. For example, Kailath [5] describes the optimum detector for a broad class of random signals in white Gaussian noise. Unfortunately, implementation of this detector requires the causal minimum mean-squared error estimator for the signal, and the structure of this device is usually unknown if the signal is non-Gaussian. Choosing a different approach, Schwartz [6] describes a Bayes optimum detection algorithm for making a decision based on a single sample of the received waveform. This algorithm assumes that a convergent series expansion for the test statistic is known; it makes optimum decisions based on a truncated version of this series.

In the present paper, we will derive approximately optimum likelihood ratio detectors for problems in which the received random processes belong to a suitably restricted class of linear processes. Our approach is based upon a generalization of Schwartz's method and is concerned with the detection of instantaneous amplitude samples of the received waveform, rather than with its envelope. We first derive a convergent series expansion for the probability densities by extending a method of Middleton [7]. A truncated version of this series is then used to develop an approximation of the optimum test statistic for a variety of problems, including models of signal detection in reverberation noise and multipath communication. Evaluation of the approximate text statistic is shown to be implementable by an algorithm that is guaranteed to terminate. This algorithm is optimum for the series approxi-

Manuscript received June 10, 1976; revised November 8, 1976. This work was supported in part by the Air Force Office of Scientific Research, Air Force Systems Command, USAF, under Grant AFOSR-74-2689, in part by the National Science Foundation under Grant GK-24187, and in part by the U.S. Army Research Office under Contract DAHC04-69-0012. This paper was presented at the Tenth Annual Allerton Conference on Circuit and System Theory, Monticello, IL, 1972.

L. F. Eastwood, Jr. is with the Center for Developmental Technology and the Department of Electrical Engineering, Washington University, St. Louis, MO 63130.

R. Lugannani is with the Department of Applied Physics and Information Science, University of California, San Diego, La Jolla, CA 92093.