Journal of Mathematical Systems, Estimation, and Control
Vol. 5, No. 4, 1995, pp. 1–37

Graph Structure and Recursive Estimation of Noisy Linear Relations*

Ramine Nikoukhah Darrin Taylor† Bernard C. Levy‡
Alan S. Willsky†

Abstract

This paper examines estimation problems specified by noisy linear relations describing either dynamical models or measurements. Each such problem has a graph structure, which can be exploited to derive recursive estimation algorithms only when the graph is acyclic, i.e., when it is obtained by combining disjoint trees. Aggregation techniques appropriate for reducing an arbitrary graph to an acyclic one are presented. The recursive maximum likelihood estimation procedures that we present are based on two elementary operations, called reduction and extraction, which are used to compress successive observations, and discard unneeded variables. These elementary operations are used to derive filtering and smoothing formulas applicable to both linear and arbitrary trees, which are, in turn applicable to estimation problems in settings ranging from 1-D descriptor systems to 2-D difference equations to multiscale statistical models of random fields. These algorithms can be viewed as direct generalizations to a far richer setting of Kalman filtering and both two-filter and Rauch-Tung-Striebel smoothing for standard causal state space models.

*Received March 29, 1993; received in final form July 27, 1994. Summary appeared in Volume 5, Number 4, 1995.

†The work of both Professors Taylor and Willsky was supported by The Office of Naval Research under Grant N00014-91-J-1004, by The National Science Foundation under Grants MIP-9015281 and INT-9002393, and by The Army Research Office under Grant DAAL03-92-G-0115.

‡The work of this author was supported by the National Science Foundation under Grant MIP-9015271 and the Office of Naval Research under Grant USN-N00014-89-J-3153.

1 Introduction

In this paper we investigate the problem of recursive estimation for a set of unknown variables subject to noisy, linear constraints. Our motivation for this is to provide a unifying framework that includes not only the standard, causal Kalman filter and its information filter counterpart [1] but also applies equally well to a much richer set of problems in which some of the natural, simplifying aspects of the standard problem, that are usually taken for granted, don't apply, requiring more careful analysis. To understand part of our perspective, consider for the moment the standard Kalman filtering problem

$$x_{k+1} = A_k x_k + B_k w_k \quad , \quad k \geq 0 \tag{1.1}$$

$$y_k = C_k x_k + r_k \quad , \quad k \geq 1 \tag{1.2}$$

where w_k and r_k are independent, zero-mean Gaussian random vectors with identity covariances, and where x_0 is a Gaussian random vector, independent of w and r, with mean m_0 and covariance P_0. While there are a variety of ways in which to derive optimal estimation algorithms for this standard problem, as we will see, the one that we must use in our general case involves adopting a maximum likelihood (ML) perspective in which initial condition, dynamics (1.1) and observation (1.2) are all viewed as "measurements", or perhaps, more appropriately, as noisy dynamic constraints. That is, we wish to estimate the sequence of unknowns x_0, x_1, $\cdots$ from the sequence of measurements

$$\begin{pmatrix} m_0 \\ 0 \\ y_1 \\ 0 \\ y_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} I & 0 & 0 & \cdots \\ -A_0 & I & 0 & \cdots \\ 0 & C_1 & 0 & \cdots \\ 0 & -A_1 & I & \cdots \\ 0 & 0 & C_2 & \\ \vdots & \vdots & \vdots & \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ \vdots \\ \vdots \end{pmatrix} + \begin{pmatrix} \tilde{x}_0 \\ B_0 w_0 \\ r_1 \\ B_1 w_1 \\ r_2 \\ \vdots \end{pmatrix} \tag{1.3}$$

where $\tilde{x}_0 = x_0 - m_0$.

Note three things about this formulation. First of all, thanks to the lower-bi-diagonal structure of the matrix on the right-hand side of (1.3), we can readily obtain recursive algorithms for the computation of the desired filtered or smoothed estimates, essentially by Gaussian elimination. Secondly, in many cases the dynamic noise $B_k w_k$ is *not* full rank (e.g. think about a second-order system driven by a single noisy input). Consequently if we adopt ML perspective we apparently have to deal with a singular estimation problem, since the "measurement noise" in (1.3) has a singular covariance. Thirdly, thanks to the identity blocks in the matrix in (1.3),

it is easy to see that the matrix relating $x_0, \cdots, x_k$ to the measurements in (1.3) has full column rank implying that this ML estimation problem is well-posed in that we do indeed have information about the full state vector. However, in some situations it is not realistic to assume that we have useful prior information about x_0 so that we either eliminate the first block row of (1.3) or replace it by only partial prior information about x_0. In this case, of course, the matrix in (1.3) may no longer have full column rank.

In this standard framework the two apparent sources of singularity that we have pointed out typically cause us no real difficulty. In particular, thanks again to the presence of the identity blocks in (1.3) – or more fundamentally to the recursive nature of the dynamic constraint in (1.1), the calculations corresponding to the incorporation of these dynamics in the estimation procedure are reduced to the essentially trivial prediction step of the Kalman filter, so that the singularity of $B_k B_k^T$ causes no difficulty. Also, if x_0 is partially or completely unknown, we can use the information form of the filter, involving inverse covariances, which yields well-defined quantities. In other words, the Kalman filter and associated smoothing algorithms, in principle, have no difficulty in dealing with perfect information, corresponding to singularity of error covariances, while the information filter and its smoothing counterparts have no problem in dealing with a complete lack of information, corresponding to singularity of the *inverses* of error covariances.

However, what happens if we may have a complete lack of information about part of the state and perfect information about another part, so that neither the error covariance nor its inverse may be well-defined? In addition, what if the relationships between unknowns x_k and observations do not have as obvious a sequential structure as that displayed by the lower bi-diagonal matrix in (1.3)? When and how can we determine recursive estimation structures for such problems, generalizing both Kalman filtering and optimal smoothing algorithms for linear stochastic systems? In this paper we answer these questions by analyzing a rather general linear estimation problem whose study enables us to expand the range of applicability of Kalman filtering techniques to systems which are far more diverse and general than the usual state-space models. Such systems include for example both 1-D and 2-D stochastic descriptor systems, where the class of 2-D descriptor systems that we consider contains as special cases the 2-D state-space models of Roesser [2] and Fornasini and Marchesini [3], and can be used to model 2-D stochastic nearest-neighbor models of the type considered in [4]. In addition, the multiscale stochastic modeling and estimation framework developed in [5, 6, 7] also falls within the class of systems captured in our formalism.

The general estimation formulation adopted here is strongly influenced

by our earlier work on the filtering and smoothing of 1-D descriptor systems [8], [9], where a general and flexible maximum likelihood (ML) approach was employed to derive recursive estimation algorithms. In addition, in [10] Chisci and Zappa independently developed a square-root Kalman filter filter for essentially the same problem studied in [9]. The main feature of the ML approach, which was itself motivated by earlier work of Whittle [11], Chapter 11, and of Bierman [12] in the context of square-root Kalman filtering, is that no distinction is made between system dynamics and observations. Specifically, all dynamic relations and initial or boundary conditions are viewed as observations, i.e. as noisy constraints on the state variables. Given a stream of observations, all observations considered up to a certain point can be compressed in such a way that the ML estimates based on the original observations or their compressed version are the same. Furthermore, observations concerning variables that are no longer of interest can be discarded. This process of compressing past observations and discarding unneeded variables is extended and generalized here in several ways. First, and most importantly, we introduce the concept of an *xo-graph*, which provides a unifying perspective for recursive estimation as well as an extremely convenient visualization of the structure of general linear estimation problems which in turn can be exploited to determine the structure of recursive estimation algorithms for the broad array of problems mentioned previously. Secondly, with the exception of [9], the previous work (e.g. in [8], [10]) on estimation for 1-D descriptor systems under singular covariance and/or information matrix conditions has focused on "causal" filtering, i.e. recursive estimation of the "current state" given "past and present" observations. In this paper we consider "noncausal" smoothing as well, providing both a generalization and a conceptually and notationally far simpler solution of the smoothing problem first analyzed in [9].

In the next section we state the general linear estimation problem of interest here, introduce its *xo*-graph representation, and illustrate by example the rich set of problems that are captured in this framework. It turns out that recursive estimation algorithms can be derived only for the class of so-called *acyclic* xo-graphs, and, since all xo-graphs are not necessarily acyclic, in Section 3 we develop aggregation operations on xo-graphs that can be used to reduce any such graph to an acyclic one. As we will see, this reduction directly provides a recursive structure for an estimation problem by identifying a grouping and sequential ordering of both the variables to be estimated and the observations to be processed. The general form of this reduction is that of a *tree*, leading to a generalization of the estimation problems considered in [5, 6, 7] for multiresolution stochastic processes. In Section 4 we then introduce the core operations required for recursive filtering and smoothing by considering several basic facts about

ML estimation. In particular, it is shown that two operations, called reduction and extraction, can be employed to compress observations, and discard unneeded variables. These two operations are then employed to derive recursive and numerically robust filtering and smoothing algorithms for ML estimation problems represented by trees. The case of linear trees is first discussed in Section 5, and the merge operation necessary to handle arbitrary trees is discussed in Section 6.

2 XO-Graphs for Linear Estimation

The problem we consider consists of estimating a set of vectors $X = \{x_i \ , \ i \in I\}$ with $x_i \in \mathbf{R}^{n_i}$, based on all or part of the set of linear observations:

$$o_k : \ z_k = \sum_j^I A_{kj} x_j + G_k u_k \ , \ k \in K \tag{2.1}$$

where the u_k's are zero-mean, independent Gaussian vectors with covariance $E[u_k u_k^T] = I$. For this estimation problem to possess a nontrivial recursive structure, each observation in (2.1) should couple together only a limited number of x_j's. For example, in (1.3) each observation involves only a single x_j or two successive values. The dependency structure existing between observations and vectors to be estimated can be described by a special type of graph, called an *xo-graph*. An xo-graph has two types of nodes: x nodes corresponding to the unknown vectors x_i, and o nodes corresponding to the observations o_k. Each measurement (2.1) is represented by a set of J_k arcs, where J_k is the number of values of j for which $A_{kj} \neq 0$, and where the node o_k is connected to node x_j if the matrix $A_{kj} \neq 0$, i.e. if the unknown vector x_j contributes to the observation o_k. An important property of xo-graphs is that to go from a given x-node to another x-node, we must go through an o-node, and vice-versa. Graphs with this property are called bipartite [13].

Let us illustrate these ideas with several examples, first a simple one that we will use to illustrate some aspects of our construction and several others that indicate the generality of this framework.

Example 1 The vectors to be estimated and observations are given by

$$X = \{x_1, x_2, x_3, x_4, x_5\} \tag{2.2a}$$
$$O = \{o_1, o_2, o_3, o_4\} \tag{2.2b}$$

with

$$o_1 : \ z_1 = A_{11} x_1 + A_{12} x_2 + G_1 u_1 \tag{2.3a}$$

$$o_2: \quad z_2 = A_{21}x_1 + A_{22}x_2 + A_{23}x_3 + G_2u_2 \tag{2.3b}$$
$$o_3: \quad z_3 = A_{33}x_3 + A_{34}x_4 + G_3u_3 \tag{2.3c}$$
$$o_4: \quad z_4 = A_{44}x_4 + A_{45}x_5 + G_4u_4\ , \tag{2.3d}$$

and the corresponding xo-graph is shown in Fig. 1.

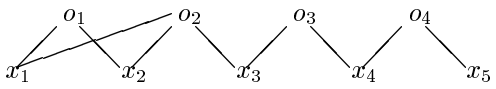

Figure 1: Xo-graph representing observations (2.3)

Note that by labeling each arc of the xo-graph with the corresponding A_{kj} matrix, and each node o_k with the observation vector z_k and matrix G_k, the given estimation problem can be totally represented by an xo-graph. The advantage of such a representation is that the structure of the xo-graph can be exploited to perform operations on the observations o_k aimed at estimating the unknown vectors x_j recursively. Also, a given xo-graph is said to be acyclic if the graph obtained by ignoring the x and o labels of the xo-graph does not contain any cycle. For example, the xo-graph of Fig. 1 contains the x_1–o_1–x_2–o_2–x_1 cycle. In the next section we describe how to reduce such xo-graphs to acyclic ones. Since xo-graphs can always be decomposed into separate connected components, corresponding here to decoupled estimation problems, the xo-graphs that we shall consider when discussing recursive estimation algorithms are therefore *trees*, i.e. connected acyclic graphs.

Example 2 The Kalman filtering and smoothing problems for descriptor systems can be formulated in the form (2.1). To see this, consider the descriptor system

$$E_{k+1}x_{k+1} = A_kx_k + B_ku_k\ , \quad 0 \le k \le N-1 \tag{2.4}$$

with observations

$$y_{k+1} = C_{k+1}x_{k+1} + D_ku_k\ , \quad 0 \le k \le N-1\ . \tag{2.5}$$

This model reduces to a standard state-space model when $E_k = I$, so that the descriptor estimation problem includes the corresponding problem for linear state-space models as a special case. To transform the system (2.4)–(2.5) to the form (2.1), the main step is to view the system dynamics (2.4) as observations linking the state vectors x_k and x_{k+1}. Combining these

observations with (2.5) yields

$$o_{k+1}: \quad z_{k+1} = \begin{pmatrix} 0 \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} -E_{k+1} \\ C_{k+1} \end{pmatrix} x_{k+1} + \begin{pmatrix} A_k \\ 0 \end{pmatrix} x_k + \begin{pmatrix} B_k \\ D_k \end{pmatrix} u_k, \tag{2.6}$$

with $0 \leq k \leq N-1$. We assume also that some information is separately available about the initial and final state vectors x_0 and x_N, which is represented by the observations

$$o_0 : z_0 = V_0 x_0 + G_{-1} u_{-1} \tag{2.7a}$$

$$o_{N+1} : z_{N+1} = V_N x_N + G_N u_N . \tag{2.7b}$$

The xo-graph corresponding to this estimation problem is shown in Fig. 2. Clearly, it is connected and acyclic, so that it forms a tree. Since each node in the interior of the tree is connected to exactly two other nodes, we call it a *linear* tree. In this context, the estimate of x_k based on all o's is the smoothed smoothed estimate of x_k, and its estimate based on the observations o_j such that $j \leq k$ is the filtered estimate.

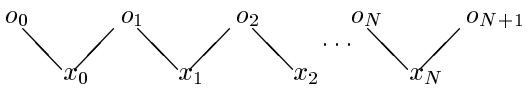

Figure 2: Xo-graph for descriptor systems

Example 3 In Example 2, it was assumed that the boundary conditions for the initial and final states x_0 and x_N were decoupled. For a two-point boundary value descriptor system (TPBVDS) [14]–[15], these decoupled conditions are replaced by a single boundary condition coupling x_0 and x_N, which can be modeled by an an observation of the form

$$o_0: \quad z_N = V_0 x_0 + V_N x_N + G_N u_N . \tag{2.8}$$

The xo-graph for this example contains the cycle x_0–o_1–x_1 ... o_N–x_N–o_0–x_0, as shown in Fig. 3.

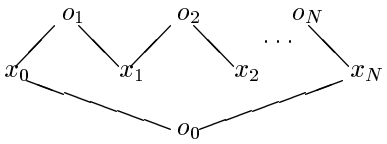

Figure 3: Xo-graph for boundary-value descriptor systems

Example 4 In the preceding examples the index sets I (for the x_i's) and K (for the z_k's) are simply subsets of the integers. The framework we develop here can handle more general index sets including the 2-D index set used in 2-D systems. Many 2-D systems, such as those obtained by discretizing linear stochastic partial differential equations, can be described by a 2-D descriptor model of the form

$$A_0 x_{i+1j+1} + A_1 x_{ij+1} + A_2 x_{i+1j} + A_3 x_{ij} = B u_{ij} \,, \tag{2.9}$$

where u_{ij} is a 2-D white Gaussian noise sequence with unit covariance. This model includes as special cases the 2-D state-space models introduced by Roesser [2] and Fornasini and Marchesini [3], which correspond respectively to the choices

$$A_0 = 0 \quad A_1 = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \quad A_2 = \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} \tag{2.10}$$

and

$$A_0 = I \,. \tag{2.11}$$

It is also easy to verify that the 2-D nearest-neighbor stochastic models

$$z_{ij} = A_E z_{i-1j} + A_W z_{i+1j} + A_S z_{ij-1} + A_N z_{ij+1} + B u_{ij} \tag{2.12}$$

considered in [4] can be rewritten in the form (2.9) provided that we select

$$x_{ij}^T = [z_{ij}^T \; z_{i-1j-1}^T] \tag{2.13}$$

as partial state vector. For simplicity, it is assumed that the descriptor model (2.9) is defined over the rectangle $0 \leq i \leq N$, $0 \leq j \leq M$, and the boundary conditions are of Dirichlet type, so that x_{ij} is known on the edges of the domain of definition. Then, given the observations

$$y_{ij} = C x_{ij} + D u_{ij} \,, \tag{2.14}$$

we seek to find the ML estimate of x_{ij} based on all observations. To convert this estimation problem into the format (2.1), the dynamics (2.9) and observations (2.14) can be combined into a single observation o_{ij}, in

the same manner as for the 1-D descriptor systems of Example 2. The resulting observation o_{ij} depends on x_{ij}, x_{i+1j}, x_{ij+1} and x_{i+1j+1}, so that the corresponding xo-graph has the structure shown in Fig. 4. Clearly this graph contains many elementary cycles.

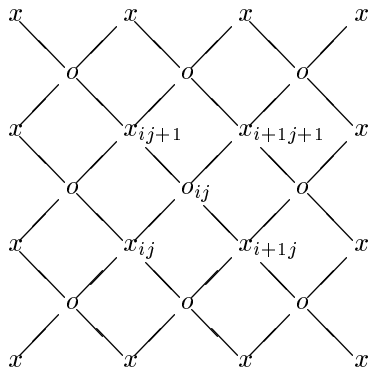

Figure 4: Xo-graph for 2-D descriptor systems

Example 5 In [5, 6, 7] a class of recursive models on dyadic trees is introduced and studied as the basis for multiresolution modeling and processing of stochastic processes. In this setting the index set I consists of nodes on the tree corresponding to scale/translation pairs (m, n), where m denotes scale and n translational offset. As we move from one scale (say m) to the next finer one, $(m + 1)$, the number of points doubles and finer detail is added to the coarser description. The general form of such a coarse-to-fine recursion is

$$x_{\alpha k} = A_{\alpha k} x_k + B_{\alpha k} w_{\alpha k} \tag{2.15}$$

$$x_{\beta k} = A_{\beta k} x_k + B_{\beta k} w_{\beta k} \tag{2.16}$$

where if k denotes the pair (m, n), then $\alpha k = (m + 1, 2n)$ and $\beta k = (m + 1, 2n + 1)$ are the two descendents of k. If we also have observations

$$y_k = C_k x_k + v_k \,, \tag{2.17}$$

the xo-graph has a tree structure as depicted in Fig. 5, where o_k consists of (2.15)–(2.17).

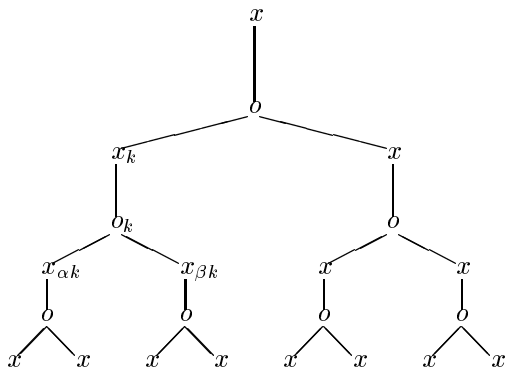

Figure 5: Xo-graph for systems over a dyadic tree

Finally, note that several of the examples given above have the feature that, based on a subset of the full observation set, the vectors to be estimated contain components which are not estimable as well as components which are known perfectly. For example, consider the descriptor system (2.4)–(2.5) with noiseless boundary conditions of the form (2.7a)–(2.7b), i.e., where $G_{-1} = G_N = 0$, and where we assume that the ranks of V_0 and V_N are each less than the corresponding dimensions of x_0 and x_N. Then, suppose we seek to estimate x_k recursively based on (2.4), (2.5) and (2.7a)–(2.7b). In this case, we start with (2.7a), from which x_0 is not estimable (since V_0 has rank less than the dimension of x_0), although some of its components are known perfectly because $G_{-1} = 0$. Furthermore, if

$$\begin{pmatrix} E_{k+1} \\ C_{k+1} \end{pmatrix}$$

does not full column rank, then x_k will not be recursively estimable based on o_j, $j \leq k$. Said another way, if we collect the dynamics (2.4), observations (2.5) and boundary conditions (2.7a)–(2.7b) into one set of simultaneous equations as we did in (1.3), the matrix we obtain is again lower bi-diagonal:

$$\begin{pmatrix} V_0 & 0 & 0 & \cdots & & \\ -A_0 & E_1 & 0 & & & \\ 0 & C_1 & 0 & & & \\ & & & \ddots & & \\ & & & & -A_N & E_N \\ & & & \cdots & 0 & C_N \\ & & & \cdots & 0 & V_N \end{pmatrix} . \tag{2.18}$$

In this case, while the full matrix may in fact be full rank (so that each x_k *is* estimable based on the use of *all* observations, including those for $j > k$),

the upper left-hand submatrices may *not* be, so that the straightforward application of Gaussian elimination, as in standard Kalman filtering and smoothing fails.

The above discussion illustrates one key feature of the recursive ML estimation problems considered in this paper, namely the necessity to cope simultaneously with the absence of information, as well as the existence of perfect information, about the vectors we seek to estimate.

3 XO-Graph Reduction

In this section we describe two types of operations that can be employed to reduce an arbitrary xo-graph to an acyclic one.

A X-aggregation

This operation consists of combining several x-nodes to form a larger x-node. The effect of this operation on the matrices A_{kj} appearing in the observation relations (2.1) is straightforward. Consider the case of Example 1. Then, the operation consisting of aggregating the nodes x_1, x_2 and x_3 of Fig. 1 into a larger node is equivalent to stacking the corresponding vectors into a single column vector

$$x(1:3) = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} , \tag{3.1}$$

and the observations can be expressed in terms of this new vector as

$$o_1: \quad z_1 = [A_{11}\ A_{12}\ 0]x(1:3) + G_1 u_1 \tag{3.2a}$$
$$o_2: \quad z_2 = [A_{21}\ A_{22}\ A_{23}]x(1:3) + G_2 u_2 \tag{3.2b}$$
$$o_3: \quad z_3 = [0\ 0\ A_{33}]x(1:3) + A_{34}x_4 + G_3 u_3 \tag{3.2c}$$
$$o_4: \quad z_4 = A_{44}x_4 + A_{45}x_5 + G_4 u_4 \, . \tag{3.2d}$$

The corresponding reduced xo-graph is depicted in Fig. 6.

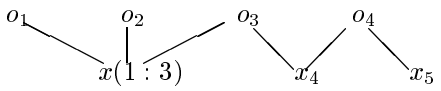

Figure 6: X-aggregation for Example 1

B O-aggregation

This operation is similar to x-aggregation except that we now combine several o nodes to form a single observation. This is accomplished by stacking the z vectors corresponding to the observations that we want to aggregate. Consider for example the xo-graph of Fig. 1, and suppose that we want to aggregate the nodes o_1 and o_2. Combining the measurement vectors z_1 and z_2 of o_1 and o_2 into a single measurement yields the aggregated observation

$$o(1:2): \quad \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} A_{11} \\ A_{31} \end{pmatrix} x_1 + \begin{pmatrix} A_{12} \\ A_{22} \end{pmatrix} x_2 + \begin{pmatrix} 0 \\ A_{23} \end{pmatrix} x_3 + \begin{pmatrix} G_1 & 0 \\ 0 & G_2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} . \quad (3.3)$$

C Reduction to acyclic form

The two aggregation operations introduced above can be used to reduce an arbitrary xo-graph into an acyclic one. In general this can be done in a number of ways, resulting in different levels of aggregation and graph structures. The simplest way of accomplishing this objective consists of aggregating all x nodes and all o nodes together, which results in a trivial graph with a single x–o arc. This reduction technique is of course not very interesting since it destroys completely the dependency structure of the original xo-graph. A more sensible approach consists in using as few aggregations as possible to reduce the given xo-graph into an acyclic one. The procedure that we employ, which is an adaptation of a technique proposed by Lauritzen and Spiegelhalter [16] to perform local computations on statistical models with a graph structure, consists in triangulating the chordless cycles of the given xo-graph. The new branches we introduce by triangulation may be of x to x, o to o, or x to o type. Then, when no chordless cycle remains, all x-nodes, and all o-nodes, which form cliques, i.e. sets of mutual neighbors, in the final graph need to be aggregated together (the aggregation can also be performed concurrently with the triangulation). Obviously, the reduction procedure that we have just described is not unique, since different fill-in strategies can be used to triangulate a given xo-graph. However, provided the new branches are selected judiciously, the final aggregated acyclic xo-graph will preserve a large part of the structure of the original graph.

To see how this approach works out in practice, consider the xo-graph of Fig. 1. It turns out that this graph can be triangulated by adding a single branch between the x_1 and x_2 nodes, so that the graph becomes acyclic after we aggregate the nodes x_1 and x_2, as shown in Fig. 7.

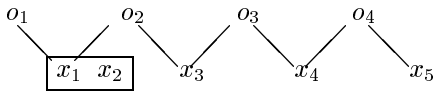

Figure 7: Reduced acyclic xo-graph for Example 1

For the TPBVDS estimation problem represented by the xo-graph of Fig. 3 the situation is slightly more complicated. To reduce this xo-graph to acyclic form, we can first aggregate the state vectors x_i and x_{N-i} for $i = 0, 1, ...$ This yields the xo-graph of Fig. 8.

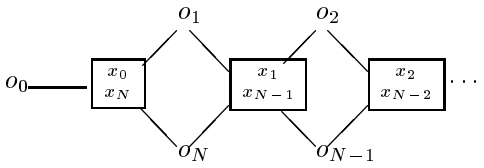

Figure 8: X-reduction of the TPBVDS xo-graph

To eliminate the remaining cycles from the graph of Fig. 8, we can aggregate the observations o_i and $o_{N-(i-1)}$ for $i = 1, 2, \ldots$, which gives the acyclic xo-graph of Fig. 9.

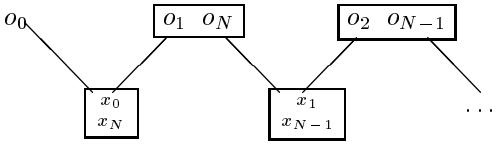

Figure 9: O-reduction of the TPBVDS xo-graph

The xo-graph of Fig. 4, which corresponds to the 2-D descriptor estimation problem of Example 4 is slightly more difficult to reduce. One reduction technique that leads to an acyclic xo-graph consists in organizing the observations o_{ij} and partial states x_{ij} into concentric regions, as shown in Fig. 10. This means that the corresponding recursive estimation algorithms for estimating x_{ij} from the given model and observations will operate either outwards from the center of the observation region toward the edges, or inwards from the edges toward the center. However, this scheme is not the only one that can be used to reduce the given graph to an acyclic one. For example, by organizing the o_{ij}'s and x_{ij}'s columnwise or rowwise, it is possible to process the data from left to right or right to left, or from top to bottom, and vice-versa.

An interesting feature of the xo-trees obtained in figures 9 and 10 for

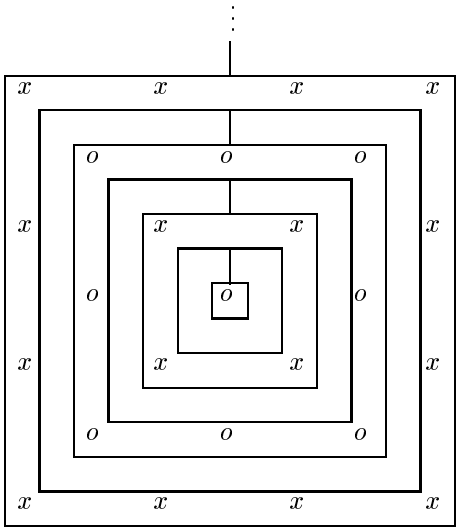

Figure 10: Reduced xo-graph for 2D descriptor estimation

the TPBVDS and 2-D descriptor estimation problems, respectively, is that they are *linear*, i.e. they contain only one branch. A consequence of this property is that the recursive filtering and smoothing algorithms for standard Gauss-Markov state-space models will be applicable with only minor modifications to the above problems, since they correspond to the same xo-tree. In constrast, more general trees, such as the dyadic tree of Fig. 5 require an interesting variation on the standard Kalman filter, described for the multiresolution model of Example 5 in [5, 6, 7] and generalized in Section 6. One property of these algorithms on trees is that they have an inherently parallel structure, as the processing on disjoint subtrees can be carried out in parallel. Indeed it is also possible to use aggregation to construct a tree structure for the 2-D problem of Fig. 4. Specifically, in this representation we first aggregate together the x_{ij}'s and o_{ij}'s along the central column of the 2-D array of Fig. 4, yielding the nodes at the top of the tree and dividing the array into two halves. Each of these is then subdivided by aggregating the x_{ij}'s and o_{ij}'s along the central row of each half. Repeating this procedure on each of the disjoint rectangular regions produced in the preceding stage yields a tree structure that is used in [17] to develop multiresolution, pyramidal models for Markov random fields.

4 Recursive Maximum Likelihood Estimation

Let x be an unknown vector in $\mathbf{R}^n$. Consider the problem of finding the maximum likelihood (ML) estimate of x given the observation

$$z = Ax + Gu \tag{4.1}$$

where z belongs to $\mathbf{R}^p$, and u is a zero-mean Gaussian vector in $\mathbf{R}^m$, with covariance matrix I_m. The problems motivating this paper lead us to consider the most general form of (4.1), where first we do not assume that the covariance GG^T of the noise Gu is is invertible. In other words, we allow the possibility that some measurements may be *perfect*. As noted earlier, this extension is motivated by the fact that if we view the dynamics of a stochastic linear system as measurements, some of the dynamic relations for the system may not be affected by noise, and will therefore specify perfect, i.e. noiseless, measurements for some of the system variables. Another important feature of the above problem is that the matrix A may not have full column rank. This means that all components of the vector x need not be estimable from the measurement (4.1). The need to consider ML estimation in such generality stems from the observation that the objective of recursive estimation is to incorporate progressively more information about a given system. In this context, although the final ML estimation problem may be well-posed, i.e. x may be estimable given all available measurements, as was observed at the end of Section 2, this is not necessarily the case for intermediate estimation problems based on a measurement subset.

ML estimation problems where both A and GG^T are singular have been studied in the literature on generalized linear regression models, and the reader is referred to [18] for a survey of results in this area. The ML estimation problem requires maximizing the probability density

$$p(u) = \frac{1}{(2\pi)^{m/2}} \exp -\frac{1}{2} u^T u \tag{4.2}$$

or equivalently, minimizing the quadratic cost $J(u) = u^T u/2$, under the constraint (4.1). The solution of this problem is well known and is given by the following theorem, which appears as Proposition 2.3 of [18] (see also section 6.4 of [19]).

Theorem 1 *$\hat{x}$ is a ML estimate of x if and only if for some λ, $\hat{x}$ satisfies*

$$\begin{pmatrix} GG^T & A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ \hat{x} \end{pmatrix} = \begin{pmatrix} z \\ 0 \end{pmatrix} . \tag{4.3}$$

The estimation error $\tilde{x} = x - \hat{x}$ corresponding to such an estimate obeys

$$\begin{pmatrix} GG^T & A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ -\tilde{x} \end{pmatrix} = \begin{pmatrix} Gu \\ 0 \end{pmatrix} , \tag{4.4}$$

so that the bias vectors $m_\lambda = E[\lambda]$ *and* $b = x - E[\hat{x}]$ *satisfy*

$$\begin{pmatrix} GG^T & A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} m_\lambda \\ -b \end{pmatrix} = 0\,. \tag{4.5}$$

Furthermore the error covariance matrix

$$\mathcal{P} \triangleq \begin{pmatrix} P_{\lambda\lambda} & -P_{\lambda\tilde{x}} \\ -P_{\tilde{x}\lambda} & P_{\tilde{x}\tilde{x}} \end{pmatrix} \tag{4.6a}$$

of the vector $[\lambda^T \ -\tilde{x}^T]^T$ *obeys*

$$\begin{pmatrix} GG^T & A \\ A^T & 0 \end{pmatrix} \mathcal{P} \begin{pmatrix} GG^T & A \\ A^T & 0 \end{pmatrix} = \begin{pmatrix} GG^T & 0 \\ 0 & 0 \end{pmatrix}. \tag{4.6b}$$

The operations that will be employed in the remainder of this section to develop a recursive ML estimation procedure rely on the following canonical decomposition of an observation of the form (4.1). This decomposition appears in section 6 of [18], but a proof is included below, since it contains the precise numerical algorithm required to implement our subsequent results.

Lemma 1 *Given a linear observation of the form (4.1), there exists a transformation* (S, W) *with* S *invertible and* W *orthonormal such that*

$$SA = \begin{pmatrix} M_1 \\ M_2 \\ 0 \\ 0 \end{pmatrix} \qquad SGW^T = \begin{pmatrix} I & 0 \\ 0 & 0 \\ 0 & I \\ 0 & 0 \end{pmatrix} \tag{4.7}$$

with $M = \begin{pmatrix} M_1 \\ M_2 \end{pmatrix}$ *onto, so that if*

$$Sz = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} \qquad Wu = \begin{pmatrix} u_1 \\ u_3 \end{pmatrix}, \tag{4.8}$$

the observation (4.1) takes the form

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} M_1 \\ M_2 \\ 0 \\ 0 \end{pmatrix} x + \begin{pmatrix} u_1 \\ 0 \\ u_3 \\ 0 \end{pmatrix}, \tag{4.9}$$

where $u_1 \sim \mathcal{N}(0, I)$ *and* $u_3 \sim \mathcal{N}(0, I)$ *are independent.*

Proof: Let

$$G = V \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} U \,, \tag{4.10}$$

with V and U orthonormal and Σ diagonal positive definite, be a singular value decomposition of G. Then, consider the matrix

$$T = \begin{pmatrix} T_1 \\ T_2 \end{pmatrix} = \begin{pmatrix} U^T \Sigma^{-1} & 0 \\ 0 & I \end{pmatrix} V^T \,. \tag{4.11}$$

Denoting

$$\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} T_1 \\ T_2 \end{pmatrix} z \qquad \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} = \begin{pmatrix} T_1 \\ T_2 \end{pmatrix} A \,, \tag{4.12}$$

the observation (4.1) can be decomposed as

$$z_1 = Z_1 x + u \qquad z_2 = Z_2 x \,. \tag{4.13}$$

By performing a QR decomposition of Z_2, we can premultiply z_2 by an orthonormal matrix Q such that

$$Q z_2 = \begin{pmatrix} y_2 \\ y_4 \end{pmatrix} = \begin{pmatrix} M_2 \\ 0 \end{pmatrix} x \,, \tag{4.14}$$

where M_2 has full row rank. Next, we project the rows of Z_1 onto the row space of M_2. Letting

$$P_2 = M_2^T (M_2 M_2^T)^{-1} M_2 \tag{4.15a}$$

be the projection matrix onto the row space of M_2, the rows of

$$\tilde{Z}_1 = Z_1 (I - P_2) \tag{4.15b}$$

are orthogonal to those of M_2. We can then consider the modified measurement

$$\tilde{z}_1 = z_1 - Z_1 M_2^T (M_2 M_2^T)^{-1} y_2 = \tilde{Z}_1 x + u \,. \tag{4.16}$$

By performing a QR decomposition of $\tilde{Z}_1$, we can find an orthonormal matrix W such that

$$W \tilde{z}_1 = \begin{pmatrix} y_1 \\ y_3 \end{pmatrix} = \begin{pmatrix} M_1 \\ 0 \end{pmatrix} x + \begin{pmatrix} u_1 \\ u_3 \end{pmatrix} \,, \tag{4.17}$$

where M_1 has full row rank and the covariance of the noise

$$\begin{pmatrix} u_1 \\ u_3 \end{pmatrix} = W u \tag{4.18}$$

is identity, so that u_1 and u_3 are independent. Thus, we have constructed a transformation (S, W) which brings the observation (4.1) to the canonical form (4.7)–(4.9), where by construction $[\, M_1^T M_2^T \,]^T$ has full row rank. □

A Reduction

A key aspect of recursive estimation in general is that we recursively compute and propagate a reduced or compressed version of the information collected – e.g. in standard causal Kalman filtering we propagate the estimate of the current state of the system. The generalization of this concept that we need for the general recursive ML estimation approach developed here is that of propagating a *sufficient statistic*, i.e. a compressed version of the collected information that is statistically equivalent to the original measurements for the purposes of recursive estimation. More precisely, we shall freely replace a set of measurements for x by another, usually smaller, set of measurements that would yield the same family of ML estimates. This replacement procedure is justified by the introduction of the following concept.

Definition 1 *Two observations*

$$o_1: \quad z_1 = A_1 x_1 + G_1 u_1 \tag{4.19a}$$

$$o_2: \quad z_2 = A_2 x_1 + G_2 u_2 \tag{4.19b}$$

are said to be equivalent if for any other observation

$$o_3: \quad z_3 = A_3 x_1 + A_4 x_2 + G_3 u_3\,, \tag{4.20}$$

with u_3 independent of u_1 and u_2, the set of ML estimates ($\hat{x}_1$, $\hat{x}_2$) based on o_1 and o_3 is identical to the set of ML estimates ($\hat{x}_1$,$\hat{x}_2$) based on o_2 and o_3.

Thus, two sets of observations are equivalent if they provide the same "information" about x. The idea of replacing a set of measurements by another containing the same information is not new and has been used informally in much of the recursive ML estimation and square-root Kalman filtering literature [12]. A notion of equivalence similar to the one introduced here was proposed recently in [10]. The definition we consider is slightly more general, since we require that equivalence should be "context-free." Specifically, given two sets of measurements for a vector x_1, for these two measurements to be equivalent, we require not only that they should yield the same ML estimates of x_1, but also that they should be equivalent in terms of estimating any other vector x_2 for which additional measurements coupling x_1 and x_2 can be obtained. The motivation for including the additional measurement o_3 in our definition of equivalence is that at any given time, we do not know whether new measurements will become available at a later stage, so that we require that any operation we perform on the existing data should not degrade our ability to estimate subsequent data.

Among all equivalent measurements of a vector x, we can then ask ourselves whether it is possible to construct one which provides the most compact representation for the information contained in the given measurement vector, i.e. a minimal sufficient statistic. The feature that determines whether a measurement has been maximally compressed is as follows.

Definition 2 *The observation*

$$o: \quad z = Ax + Gu \tag{4.21}$$

is called reduced if A has full row rank.

To gain some intuition about the information contained in a reduced observation, consider the case of a reduced measurement where x is estimable, i.e. such that A has full column rank. Then the ML estimate of x is given by $\hat{x} = A^{-1}z$, and without loss of information we can premultiply (4.21) by A^{-1}, which yields

$$\hat{x} = x + \tilde{G}u \tag{4.22}$$

where $\tilde{G} = A^{-1}G$ and $P_{\tilde{x}\tilde{x}} = \tilde{G}\tilde{G}^T$ is the ML error covariance. Thus, in this case, a reduced observation just encodes the ML estimate of x and its error covariance. More generally, when x is not estimable, the reduced observation (4.21) can be viewed as encoding the ML estimate and error variance for the estimable part $x' = Ax$ of x.

The introduction of the concept of reduced observation is justified by the following result.

Theorem 2 *Any observation o admits an equivalent reduced observation. Furthermore, if*

$$o_1: \quad z_1 = A_1x + G_1u_1 \tag{4.23a}$$

$$o_2: \quad z_2 = A_2x + G_2u_2 \tag{4.23b}$$

are two equivalent reduced observations, there exists an invertible matrix T such that

$$A_1 = TA_2 \tag{4.24a}$$

$$G_1G_1^T = TG_2G_2^TT^T\,. \tag{4.24b}$$

The relations (4.24a)–(4.24b) show that the reduced observation corresponding to a given measurement is unique up to left multiplication by an invertible matrix.

Proof: Without loss of generality o can be assumed to be in the canonical form (4.9). The new observation

$$R\{o\}: \quad \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} M_1 \\ M_2 \end{pmatrix} x + \begin{pmatrix} u_1 \\ 0 \end{pmatrix} \tag{4.25}$$

obtained by retaining only the first two block rows of (4.9) is reduced since M is onto. To verify $R\{o\}$ is equivalent to o, consider a measurement

$$o' : \quad z' = Ex + Fx' + G'u' \tag{4.26}$$

coupling x with another variable x'. Because the last two blocks of the decomposition (4.9) do not contain x and the noise u_3 is independent of both u_1 and u', the ML estimates of x and and x' based respectively on o and o', and on $R\{o\}$ and o' are identical.

To complete the proof of Theorem 2, we need to prove that if o_1 and o_2 are two equivalent reduced observations of the form (4.23a) and (4.23b), they are necessarily related through (4.24a)–(4.24b). Note that the ML estimates of x based on o_i with $i = 1, 2$ satisfy

$$\begin{pmatrix} G_1 G_i^T & A_i \\ A_i^T & 0 \end{pmatrix} \begin{pmatrix} \lambda_i \\ \hat{x} \end{pmatrix} = \begin{pmatrix} z_i \\ 0 \end{pmatrix} . \tag{4.27}$$

Since the observation o_i is reduced A_i^T has full column rank, so that $\lambda_i = 0$, and (4.27) reduces to

$$z_i = A_i \hat{x} \tag{4.28}$$

for $i = 1, 2$. In order for the solutions of equation (4.28) to coincide for $i = 1, 2$, A_1 and A_2 must have the the same right null space. Since A_1 and A_2 have full row rank, this means that they must have the same reduced row echelon form, so that there exists an invertible matrix T such that (4.24a) holds. In addition, to ensure that the solutions of (4.28) are the same for $i = 1, 2$, we must also have $z_1 = Tz_2$, which in combination with (4.24a) implies (4.24b). □

B Extraction

Another operation that is needed in deriving recursive estimation algorithms involves discarding unneeded variables that are no longer of interest. Equivalently, given a measurement, we want to be able to extract a submeasurement concerning only the variables in which we are still interested. For example in standard Kalman filtering, the prediction step in fact corresponds to an extraction of relevant information about x_{k+1} from previous measurements and current dynamics and the dropping of the estimate of x_k from the set of statistics to be updated when the next measurement is to be incorporated. This operation, which will be called *extraction*, was first described in [8], [9] and was also introduced in [10] in the context of square-root Kalman filtering for descriptor systems. The main difficulty in performing an extraction is that we want to ensure that we are not throwing away any useful information information concerning the variables that are of interest. This requirement can be expressed as follows.

Definition 3 *The observation*

$$o_1: \quad z_1 = Ax_1 + G_1 u_1 \tag{4.29}$$

is said to be an extraction of the vector x_1 from the measurement

$$o_0: \quad z_0 = A_0 x_0 + A_1 x_1 + G_0 u_0 \tag{4.30}$$

if for all observations

$$o_2: \quad z_2 = A_2 x_1 + A_3 x_2 + G_2 u_2\,, \tag{4.31}$$

with u_2 independent of u_0 and u_1, the set of ML estimates $\hat{x}_1$, $\hat{x}_2$ based on o_1 and o_2 is identical to the set of ML estimates $\hat{x}_1$, $\hat{x}_2$ based on o_0 and o_2.

The following result provides a general mechanism for performing extractions.

Theorem 3 *The observation*

$$o_1: \quad Lz = LA_1 x_1 + LG_1 u \tag{4.32}$$

is an extraction of x_1 from

$$o: \quad z = A_0 x_0 + A_1 x_1 + G_1 u \tag{4.33}$$

if L is a basis of the left null space of A_0, i.e., it is a matrix of maximum rank such that $LA_0 = 0$.

Proof: First observe that to construct L we need only to perform a QR factorization of A_0. Specifically, let Q be an orthonormal matrix such that

$$\begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} A_0 = \begin{pmatrix} M \\ 0 \end{pmatrix}\,, \tag{4.34}$$

where M has full row rank. Then $L = Q_2$. To prove that (4.32) is an extraction of (4.33), note that the ML estimates of x_1 and x_2 based on (4.33) and (4.31) must satisfy

$$\begin{pmatrix} G_1 G_1^T & 0 & A_0 & A_1 & 0 \\ 0 & G_2 G_2^T & 0 & A_2 & A_3 \\ A_0^T & 0 & 0 & 0 & 0 \\ A_1^T & A_2^T & 0 & 0 & 0 \\ 0 & A_3^T & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \hat{x}_0 \\ \hat{x}_1 \\ \hat{x}_2 \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{4.35}$$

for some λ_1, λ_2 and $\hat{x}_0$, whereas the ML estimates of x_1 and x_2 based on the extracted measurement (4.32) and (4.31) obey

$$\begin{pmatrix} LG_1 G_1^T L^T & 0 & LA_1 & 0 \\ 0 & G_2 G_2^T & A_2 & A_3 \\ A_1^T L^T & A_2^T & 0 & 0 \\ 0 & A_3^T & 0 & 0 \end{pmatrix} \begin{pmatrix} \lambda_e \\ \lambda_2 \\ \hat{x}_1 \\ \hat{x}_2 \end{pmatrix} = \begin{pmatrix} Lz_1 \\ z_2 \\ 0 \\ 0 \end{pmatrix}\,. \tag{4.36}$$

Note that the second and last block rows of (4.35) and (4.36) are identical.

To show that the ML estimates $\hat{x}_1$ and $\hat{x}_2$ which solve (4.35) and (4.36) are the same, assume first that λ_1, λ_2, $\hat{x}_0$, $\hat{x}_1$ and $\hat{x}_2$ satisfy (4.35). The third block row

$$A_0^T \lambda_1 = 0 \tag{4.37}$$

of this equation implies that λ_1 is in the left null space of A_0, and since L is a basis of this null space, there exists a vector λ_e such that

$$\lambda_1 = L^T \lambda_e \, . \tag{4.38}$$

Substituting this relation inside the fourth block row of (4.35) yields the third block row of (4.36), and multiplying the first block row of (4.35) by L and taking into account $LA_0 = 0$, we obtain the first block row of (4.36). Thus, the vectors λ_e, λ_2, $\hat{x}_1$ and $\hat{x}_2$ satisfy (4.36).

Conversely, let the vectors λ_e, λ_2, $\hat{x}_1$ and $\hat{x}_2$ obey (4.36). Then define the vector λ_1 through (4.36). This implies that the third block row of (4.36) is the same as the fourth block row of (4.41), and since L is a basis of the left null space of A_0, we have

$$A_0^T \lambda_1 = 0 \, , \tag{4.39}$$

so that the third block row of (4.35) is satisfied. Finally, consider the first block row

$$L(G_1 G_1^T \lambda_1 + A_1 \hat{x}_1 - z_1) = 0 \tag{4.40}$$

of (4.36). This implies that the vector

$$a \stackrel{\triangle}{=} G_1 G_1^T \lambda_1 + A_1 \hat{x}_1 - z_1 \tag{4.41}$$

is orthogonal to the left null space of A_0, so that it must be in its column space, i.e. we can find $\hat{x}_0$ such that $a = -A_0 \hat{x}_0$, which implies that the first block row of (4.35) is satisfied. This shows that the ML estimates $\hat{x}_1$ and $\hat{x}_2$ based on (4.31) and (4.33) are the same as those based on (4.32) and (4.33), so that (4.32) is an extraction of (4.33). □

The extraction operation admits also the following interpretation [20] in terms of the canonical decomposition of Lemma 1. Consider the observation o given in (4.33). Suppose that we view x_1 as known and let

$$M_{x_1}\{o\} : \quad z - A_1 x_1 = \tilde{z} = A_0 x_0 + G_1 u_1 \tag{4.42}$$

be the observation obtained by letting x_1 be part of the measurement vector. Then, decomposing $M_{x_1}\{o\}$ as in (4.9), and retaining the last two blocks of the decomposition, i.e., those that are discarded when performing a reduction with respect to x_0, yields an extraction $X_{x_1}\{o\}$ of x_1 from o.

C Recursive Estimation

We have now all the elements necessary to develop recursive ML estimation algorithms. The algorithms considered below will be based entirely on the following *elementary operations*:

(i) Aggregating two observations: $o = o_1 \wedge o_2$

(ii) Reducing an observation o: $o_r = R\{o\}$.

(iii) Extracting a vector x from an observation o: $o_x = X_x\{o\}$.

(iv) The equivalence of two observations o_1 and o_2: $o_1 \equiv o_2$. when o_1 and o_2 are both reduced, this means that they can be obtained from each other by left multiplication by an invertible matrix.

Note that since the reduction and extraction operations described above rely on numerically stable techniques such as the singular value decomposition, QR factorizations, and orthonormal projections, all algorithms obtained by combining such operations can be implemented in a numerically reliable manner. The recursive estimation algorithms described in the next two sections rely on the following result, which is a direct consequence of theorems 2 and 3.

Theorem 4 *Consider the observations*

$$o_1: \quad z_1 = A_1x_1 + A_2x_2 + G_1u_1 \tag{4.43a}$$

$$o_2: \quad z_2 = A_3x_2 + A_4x_3 + G_2u_2\,. \tag{4.43b}$$

Then

$$R\{X_{x_3}\{o_1 \wedge o_2\}\} \equiv R\{X_{x_3}\{R\{X_{x_2}\{o_1\}\} \wedge o_2\}\}\,. \tag{4.44}$$

Thus, we can either perform extraction and reduction operations in one step by working with all observations together, or we can perform these operations recursively, as more observations become available. That is, we can first process (4.43a), keeping only a reduced form of the information concerning x_2 contained in o_1 and then can combine this with (4.43b), allowing us to repeat the process by extracting x_3 and again transforming this into reduced form.

5 Estimation on Linear Trees

In this section, we consider estimation problems corresponding to linear xo-trees, which are trees with a single branch. The main motivation for considering such trees arises from the observation that standard state-space estimation problems have exactly this tree structure. Thus, provided that

they are expressed abstractly in terms of the elementary operations introduced in the previous section, all the standard Kalman filtering and smoothing algorithms are applicable to this family of trees. Another motivation for considering such trees is that many problems that do not obviously have such a structure can, through the use of the x- and o-aggregation operations introduced in Section 2, be reduced to linear trees. This is the case for example of the TPBVDS and 2-D descriptor estimation problems whose reduced graphs appear in Fig. 9 and Fig. 10, respectively.

Consider the estimation problem

$$\begin{aligned} o_0: \quad z_0 &= E_0 x_0 + G_0 u_0 && (5.1a)\\ o_k: \quad z_k &= E_k x_k + A_{k-1} x_{k-1} + G_{k-1} u_{k-1}\,, \quad 1 \le k \le N && (5.1b)\\ o_{N+1}: \quad z_{N+1} &= A_N x_N + G_{N+1} u_{N+1} && (5.1c) \end{aligned}$$

whose xo-graph is the linear tree of Fig. 2.

A Filtering

The Kalman filtering algorithm for this problem can be described as follows. Let $\hat{o}_k^f$ be the observation obtained by extracting x_k from the combination of all observations o_j with $0 \le j \le k$, and then reducing the resulting extracted observation, i.e.

$$\hat{o}_k^f \stackrel{\triangle}{=} R\{X_{x_k}\{\wedge_{j=0}^k o_j\}\}\,. \tag{5.2}$$

Then, Theorem 4 implies that $\hat{o}_k^f$ satisfies the recursion

$$\begin{aligned} \hat{o}_0^f &= R\{o_0\} && (5.3a)\\ \hat{o}_{k+1}^f &= R\{X_{x_{k+1}}\{\hat{o}_k^f \wedge o_{k+1}\}\} && (5.3b) \end{aligned}$$

for $0 \le k \le N-1$, which is the abstract form of the forward Kalman filter. In these recursions, the observation $\hat{o}_k^f$ represents a complete summary of the information about x_k contained in the past observations o_j, $0 \le j \le k$. To the observation $\hat{o}_k^f$, we can then associate a family of ML estimates $\hat{x}_k^f$ of x_k based on the past observations, where the estimate $\hat{x}_k^f$ is unique only if x_k is estimable from the observations o_j with $0 \le j \le k$. This is ensured in particular if the matrices E_k have full rank for all k. When x_k is estimable, the observation $\hat{o}_k^f$ can be represented as

$$\hat{o}_k: \quad \hat{x}_k^f = x_k + P_k^{f\,1/2} v_k\,, \tag{5.4}$$

where $P_k^{f\,1/2}$ is a square-root of the error covariance matrix P_k^f of the estimate $\hat{x}_k^f$, and v_k is a zero-mean Gaussian vector with unit variance.

Example 2, continued: The Kalman filtering problem for descriptor systems has been investigated in [21], [22], [8], [10]. When x_k is estimable from past observations, a general 3-block form for the optimum filter and its associated Riccati equation were obtained in [8], which includes also a complete analysis of the steady-state convergence of the optimum filter. The case when x_k is not estimable was subsequently examined in [9], [23], [10].

For descriptor systems, it is shown in Section 2 that the observations (5.1a)–(5.1c) take the form (2.6). Then, the recursive filtering scheme (5.3a)–(5.3b), when applied to these observations, reduces exactly to the square-root algorithm of [10]. It is also easy to verify that when x_k is estimable, the recursions (5.3a)–(5.3b) yield the 3-block Kalman filter of [8]. To see this, note that the first step of the recursion (5.3b) requires extracting x_{k+1} from the combination of $\hat{o}_k$ and observation (2.6). But, when x_k is estimable, $\hat{o}_k^f$ can be expressed in the form (5.4), and the extraction step consists in backsubstituting $x_k = \hat{x}_k^f - P_k^{f\,1/2} v_k$ inside (2.6). This yields the observation

$$\begin{pmatrix} A\hat{x}_k^f \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} E_{k+1} \\ C_{k+1} \end{pmatrix} x_{k+1} + \begin{pmatrix} A_k P_k^{f\,1/2} & -B_k \\ 0 & D_k \end{pmatrix} \begin{pmatrix} v_k \\ u_k \end{pmatrix}, \qquad (5.5)$$

where v_k and u_k are independent. From this observation, we see that x_{k+1} will be estimable provided the matrix $[E_{k+1}^T \; C_{k+1}^T]^T$ has full column rank, which is precisely the condition obtained in [8]. Then, the next step of the recursion (5.3b) requires reducing the observation (5.5). One way to achieve this is to apply the reduction procedure described in the proof of Lemma 1. However, the resulting filter has an implicit form. To obtain a closed-form expression for the filter, we need only to note that according to Theorem 1, the ML estimate $\hat{x}_{k+1}^f$ of x_{k+1} satisfies

$$\begin{pmatrix} A_k P_k^f A_k^T + Q_k & -S_k & E_{k+1} \\ -S_k^T & R_k & C_{k+1} \\ E_{k+1}^T & C_{k+1}^T & 0 \end{pmatrix} \begin{pmatrix} \xi_k \\ \lambda_k \\ \hat{x}_{k+1}^f \end{pmatrix} = \begin{pmatrix} A_k \hat{x}_k^f \\ y_{k+1} \\ 0 \end{pmatrix} \qquad (5.6a)$$

with

$$Q_k = B_k B_k^T \quad S_k = B_k D_k^T \quad R_k = D_k D_k^T \,, \qquad (5.6b)$$

from which we deduce the 3-block expression

$$\hat{x}_{k+1}^f = \begin{pmatrix} 0 & 0 & I \end{pmatrix} \begin{pmatrix} A_k P_k^f A_k^T + Q_k & -S_k & E_{k+1} \\ -S_k^T & R_k & C_{k+1} \\ E_{k+1}^T & C_{k+1}^T & 0 \end{pmatrix}^{\dagger} \begin{pmatrix} A_k \hat{x}_k^f \\ y_{k+1} \\ 0 \end{pmatrix} \qquad (5.7)$$

which was obtained for the filter in [8]. Here $M^\dagger$ denotes the Moore-Penrose pseudo-inverse ([24], p. 243) of a matrix M. Similarly, the expression (4.6b) for the error covariance of a ML estimate yields the 3-block Riccati equation

$$P^f_{k+1} = -\begin{pmatrix} 0 & 0 & I \end{pmatrix} \begin{pmatrix} A_k P^f_k A^T_k + Q_k & -S_k & E_{k+1} \\ -S^T_k & R_k & C_{k+1} \\ E^T_{k+1} & C^T_{k+1} & 0 \end{pmatrix}^\dagger \begin{pmatrix} 0 \\ 0 \\ I \end{pmatrix}. \tag{5.8}$$

□

B Smoothing

With the exception of [9], [23], most treatments of recursive ML estimation for problems of the type considered here (or special cases thereof as in [8]) have dealt exclusively with problems of filtering. In this subsection we present the generalizations of the two-filter and double-sweep smoothing algorithms [25], [26], [27] for standard state-space models to the general ML estimation problem on linear trees. As we will see, while the two-filter generalization is straightforward, the double-sweep or Rauch-Tung-Striebel (RTS) algorithm has a subtle twist, due to the nature of the extraction and reduction processes, in order to deal with the fact that variables that are not estimable using past data may become estimable when future data is included as well.

To begin, we construct a backward Kalman filter which is the counterpart of the forward filter (5.3a)–(5.3b) in the sense that it starts from the other end of the tree and propagates the opposite direction. Let $\hat{o}^b_k$ be the observation obtained by extracting the vector x_k from the combination of the observations o_j such that $k+1 \le j \le N+1$, and then reducing the resulting extracted observation, i.e.

$$\hat{o}^b_k = R\{X_{x_k}\{\wedge^{N+1}_{k+1} o_j\}\}. \tag{5.9}$$

Then, $\hat{o}^b_k$ can be computed recursively with the backward Kalman filter

$$\hat{o}^b_N = R\{o_{N+1}\} \tag{5.10a}$$

$$\hat{o}^b_k = R\{X_{x_k}\{\hat{o}^b_{k+1} \wedge o_{k+1}\}\}. \tag{5.10b}$$

Consider now the smoothed observation $\hat{o}^s_k$ obtained by extracting x_k from all observations, and then reducing the resulting extracted observation, so that

$$\hat{o}^s_k = R\{X_{x_k}\{\wedge^{N+1}_{j=0} o_j\}\}. \tag{5.11}$$

According to Theorem 4, $\hat{o}^s_k$ can be constructed by extracting x_k separately from the past and future observations, and reducing the resulting observations, which gives $\hat{o}^f_k$ and $\hat{o}^b_k$, and then reducing the combination of these

two observations. Thus, we have

$$\hat{o}_k^s = R\{\hat{o}_k^f \wedge \hat{o}_k^b\}\,, \tag{5.12}$$

which is the *two-filter* smoothing formula for the given tree estimation problem.

It reduces to the usual two-filter smoothing formula when x_k is estimable separately from the past and future observations, and the corresponding covariance matrices P_k^f and P_k^b are positive definite. This can be verified by noting that under these assumptions, the observations $\hat{o}_k^f$ and $\hat{o}_k^b$ can be expressed as

$$\hat{o}_k^f : \quad d_k^f \;\stackrel{\triangle}{=}\; (P_k^f)^{-1/2}\hat{x}_k^f = (P_k^f)^{-1/2}x_k + u_k^f \tag{5.13a}$$

$$\hat{o}_k^b : \quad d_k^b \;\stackrel{\triangle}{=}\; (P_k^b)^{-1/2}\hat{x}_k^b = (P_k^b)^{-1/2}x_k + u_k^b\,, \tag{5.13b}$$

where u_k^f and u_k^b are two independent zero-mean Gaussian vectors with unit intensity. Then, when the reduction operation described in the proof of Theorem 2 is applied to the combination $\hat{o}_k^f \wedge \hat{o}_k^b$, it requires finding an orthonormal matrix T such that

$$T \begin{pmatrix} (P_k^f)^{-1/2} & d_k^f \\ (P_k^b)^{-1/2} & d_k^b \end{pmatrix} = \begin{pmatrix} (P_k^s)^{-1/2} & d_k^s \\ 0 & a_k \end{pmatrix}\,, \tag{5.14}$$

where $\hat{x}_k^s$ and P_k^s denote the smoothed estimate of x_k and its error variance, $d_k^s = P^{s-1/2}\hat{x}_k^s$, and a_k is an arbitrary vector. Premultiplying (5.14) by its transpose, and taking into account the orthogonality of T, yields

$$(P_k^s)^{-1} \;=\; (P_k^f)^{-1} + (P_k^b)^{-1} \tag{5.15a}$$

$$(P_k^s)^{-1}\hat{x}_k^s \;=\; (P_k^f)^{-1}\hat{x}_k^f + (P_k^b)^{-1}\hat{x}_k^b \tag{5.15b}$$

which are the usual two-filter smoothing relations.

Turning to the generalization of the RTS smoothing formula for a linear xo-tree, we find that we must be a bit careful in developing this result. In particular the usual RTS smoother for causal systems consists of a forward Kalman filter to process the data and a reverse sweep that processes the filtered estimates alone in order to produce the smoothed estimates. If one were to write this smoothing problem as a large, static estimation problem – i.e. as in (1.3) – we would find that this procedure corresponds simply to a Gaussian elimination step (the Kalman filter) on the block tridiagonal normal equations arising from (1.3) and a back substitution (the reverse sweep) to yield the smoothed estimate. In our more general problem of ML estimation on linear trees, we also have a tridiagonal set of normal equations, and the Kalman filtering reduction/extraction procedure described previously corresponds to the Gaussian elimination step with one

significant difference, namely that because of the possibility that x_k may not be estimable based only on the past, *the filtering procedure given by (5.3b) may discard some measurements that are of no value for filtering but may be of use for smoothing.*

In particular, suppose that we have $\hat{o}_k^f$ as in (5.2) and suppose that some part of x_k is not estimable based on $\hat{o}_k^f$. More precisely, suppose that some part of $A_k x_k$ is not estimable. In this case the incorporation of o_{k+1} as in (5.3b) will include some nonestimable portion of x_k. For recursive *filtering*, however, at this point we are longer interested in x_k but rather in x_{k+1}, and the result is that the extraction process for x_{k+1} in (5.3b) will discard that portion of o_{k+1} that contains nonestimable parts of x_k. that is, at each step in the Gaussian elimination process we *discard* some of the equations, i.e. we ignore some measurements, which are not important for filtering. However, these discarded pieces of information may very well be of value for *smoothing* (e.g. if $E_{k+1}x_{k+1}$ is estimable based on future data, then o_{k+1} will provide useful information for the part of $A_k x_k$ not estimable based solely on past information).

The net result of all of this is that, unlike the standard causal case, the backward sweep of the general RTS algorithm will also involve the processing of at least a part of the raw data (corresponding to the discarded measurements). To see how to do this, we note that the culprit here is the coupling between x_k and x_{k+1} in o_{k+1}. Thus, let us collect *all* information about these two variables into three sets: the set prior to o_{k+1} (which involves x_k but not x_{k+1}), o_{k+1} itself, and the information subsequent to o_{k+1} (involving x_{k+1} but not x_k). That is, we observe that all the information about x_k and x_{k+1} contained in the observations o_j with $0 \leq j \leq N+1$ is also contained in the compressed observation $\hat{o}_k^f \wedge o_{k+1} \wedge \hat{o}_{k+1}^b$. Since we do not necessarily assume that x_k is separately estimable from the past or future observations alone, the forward and backward observations $\hat{o}_k^f$ and $\hat{o}_{k+1}^b$ can be assumed to take the general form

$$\hat{o}_k^f : \quad z_k^f = L_k^f x_k + G_k^f u_k^f \tag{5.16a}$$

$$\hat{o}_{k+1}^b : \quad z_{k+1}^b = L_{k+1}^b x_{k+1} + G_{k+1}^b u_{k+1}^b \tag{5.16b}$$

where u_k^f and u_{k+1}^b are independent with unit variance. Then, according to Theorem 1, the smoothed estimates $\hat{x}_k^s$ and $\hat{x}_{k+1}^s$ satisfy the system

$$\begin{pmatrix} G_k^f G_k^{fT} & 0 & 0 & L_k^f & 0 \\ 0 & G_k G_k^T & 0 & A_k & E_{k+1} \\ 0 & 0 & G_{k+1}^b G_{k+1}^{bT} & 0 & L_{k+1}^b \\ L_k^{fT} & A_k^T & 0 & 0 & 0 \\ 0 & E_{k+1}^T & L_{k+1}^{bT} & 0 & 0 \end{pmatrix} \begin{pmatrix} \lambda_f \\ \xi \\ \lambda_b \\ \hat{x}_k^s \\ \hat{x}_{k+1}^s \end{pmatrix} = \begin{pmatrix} z_k^f \\ z_{k+1} \\ z_{k+1}^b \\ 0 \\ 0 \end{pmatrix} \tag{5.17}$$

for some vectors λ_f, ξ and λ_b. Dropping the third and fifth block rows of (5.17) gives

$$\begin{pmatrix} G_k^f G_k^{fT} & 0 & L_k^f \\ 0 & G_k G_k^T & A_k \\ L_k^{fT} & A_k^T & 0 \end{pmatrix} \begin{pmatrix} \lambda_f \\ \xi \\ \hat{x}_k^s \end{pmatrix} = \begin{pmatrix} z_k^f \\ z_{k+1} - E_{k+1}\hat{x}_{k+1}^s \\ 0 \end{pmatrix} . \quad (5.18)$$

If we now assume that x_k is estimable from *all* observations, the relation (5.18) yields the Rauch-Tung-Striebel smoothing recursion

$$\hat{x}_k^s = \begin{pmatrix} 0 & 0 & I \end{pmatrix} \begin{pmatrix} G_k^f G_k^{fT} & 0 & L_k^f \\ 0 & G_k G_k^T & A_k \\ L_k^{fT} & A_k^T & 0 \end{pmatrix}^\dagger \begin{pmatrix} z_k^f \\ z_{k+1} - E_{k+1}\hat{x}_{k+1}^s \\ 0 \end{pmatrix} , \quad (5.19)$$

where the smoothed estimate $\hat{x}_k^s$ is obtained by first propagating the Kalman filter (5.3b) in the forward direction, which gives the observations $\hat{o}_k^f$, and then propagating (5.19) in the backwards direction, so that the whole tree is swept twice in opposite directions. In general, *both* of these sweeps use the original data z_k. However, by carefully organizing the observations (5.1b) in the standard causal case, it is straightforward to recover the usual RTS algorithm.

6 Arbitrary Trees

In [5, 6, 7] the estimation problem for multiscale processes on dyadic trees, as described by the model (2.15)–(2.16) of Example 5, is considered, and the generalization of the RTS algorithm is developed for this problem. In this section we consider the more general problem of possibly singular measurements on arbitrary trees, and we describe the extensions of both the two-filter and RTS algorithms to this setting. To begin this development, it is useful to observe that the forward and backward Kalman filters for linear trees were initialized at each extremity of the tree, and then combined all tree observations sequentially, in the order in which they were encountered. The same principle applies for arbitrary trees: the filtering and smoothing algorithms that we develop in this section rely on initializing a filter at each extremity of the tree, and then merging the outputs of different filters as we move inward from the extremities of the tree. The key additional operation needed to perform this task is the generalization of the *merge operation* introduced in [5, 6, 7], whereby the outputs of Kalman filters characterizing the observations contained in nonoverlapping subtrees are combined to yield estimates which now summarize the observations on the subtree formed by the union of the merging subtrees.

A Merge operation

Consider an o-node connected to the nodes x_j, with $1 \leq j \leq N$, as shown in Fig. 11. Then, if $\mathcal{T}$ denotes the given tree, the tree $\mathcal{T} - \{o\}$ obtained by removing from the node o and the arcs connected to it from $\mathcal{T}$ can be partitioned into N subtrees $\mathcal{T}_j$, where a node belongs to the subtree $\mathcal{T}_j$ if the unique path connecting it to o passes through node x_j. Note that since o is only connected to the nodes x_j, $1 \leq j \leq N$, any path leading to o must necessarily go through one of these nodes. Let O be the observation obtained by o-aggregation of all observations contained in the tree $\mathcal{T}$. Let also O_j be the observation obtained by aggregating all observations of the subtree $\mathcal{T}_j$. The observation O can be decomposed as

$$O = O_1 \wedge \check{O}_1 \tag{6.1a}$$

$$\check{O}_1 = \left(\wedge_{j=2}^{N} O_j\right) \wedge o\,, \tag{6.1b}$$

where $\check{O}_1$ corresponds to the observation obtained by removing all observations contained in subtree $\mathcal{T}_1$ from O, or equivalently, by aggregating the observations O_j for $j \neq 1$ with the o-node observation. Then, let $\hat{o}_j$ be the observation obtained by extracting x_j from O_j, and reducing the resulting observation, so that

$$\hat{o}_j = R\{X_{x_j}\{O_j\}\}\,. \tag{6.2}$$

Similarly, let $\check{o}_1$ be the observation obtained by extracting x_1 from $\check{O}_1$ and reducing the resulting observation, i.e.

$$\check{o}_1 = R\{X_{x_1}\{\check{O}_1\}\}. \tag{6.3}$$

Then the merge operation

$$\check{o}_1 = R\{X_{x_1}\{\left(\wedge_{j=2}^{N}\hat{o}_j\right) \wedge o\}\} \tag{6.4}$$

is the extension to arbitrary trees of the forward and backward Kalman filtering identities obtained in the previous section. It is a direct consequence of Theorem 4, and provides a mechanism for recursively processing the tree observations, starting from the extremities of the tree and moving inwards.

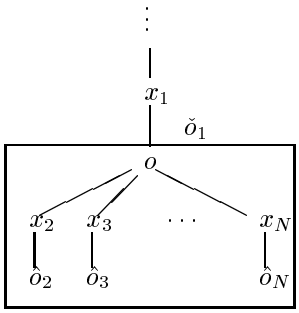

Figure 11: Merge operation

B Smoothing

Both the two-filter and RTS smoothing algorithms can be extended to trees, although, as we now show, the structure of the tree leads to some differences in the structure of these algorithms. To begin, we consider the two-filter algorithm.

Suppose that we wish to compute the smoothed estimate at a particular node x on an arbitrary tree. If we remove this x node and the arcs connected to it, we break the tree into disjoint components. The observations contained into each component can be processed recursively through the use of merging steps, so as to get a measurement summarizing the information about x contained in each subtree. All the subtree measurements can then be combined, thus yielding a smoothed observation characterizing the information about x contained in the whole tree.

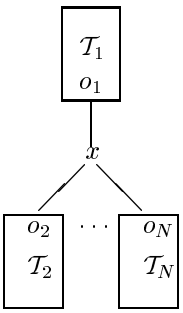

Figure 12: Smoother geometry

In detail, consider an x-node of a tree $\mathcal{T}$, which is connected to nodes

o_j with $1 \leq j \leq M$, as shown in Fig. 12. The tree $\mathcal{T} - \{x\}$ obtained by removing x and the arcs connected to it from $\mathcal{T}$ is partitioned into M disconnected subtrees $\mathcal{T}_j$, where a node belongs to the subtree $\mathcal{T}_j$ if the unique path connecting this node to x goes through the node o_j. Then, let O_j be the observation obtained by o-aggregation of all observations contained in the subtree $\mathcal{T}_j$, including o_j, and let $\hat{o}_j$ be the observation obtained by extracting x from O_j and reducing the resulting observation, i.e.

$$\hat{o}_j = R\{X_x\{O_j\}\}\,. \tag{6.5}$$

Let also $\hat{o}^s$ be the smoothed observation obtained by extracting x from the o-aggregation of all observations in the tree, and reducing the resulting observation, so that

$$\hat{o}^s = R\{X_x\{O\}\}\,. \tag{6.6}$$

Then, the analog of the two-filter formula for arbitrary trees is given by

$$\hat{o}^s = R\{\wedge_{j=1}^{M} \hat{o}_j\}\,, \tag{6.7}$$

which shows that $\hat{o}^s$ can be obtained by combining the information about x contained in each subtree, and reducing the resulting observation. Note that (6.7) is actually an "M-filter" smoothing formula, where M denotes the degree of the node x in the graph, i.e., the number of branches connected to x.

In (6.7) the observations $\hat{o}_j$ can be constructed recursively by using merge operations to progressively collapse the tree $\mathcal{T}_j$ from its extremities towards node x. This can be performed systematically by giving a root structure to the tree $\mathcal{T}$, where x is selected as the root, the observations o_j are located on the first level of the tree, to the o_j's are put on the second level, etc. Then, by using merge operations to move from the higher to lower levels of the tree, one can progressively compress the observations contained in each subtree $\mathcal{T}_j$, so as to place ourselves in the situation corresponding to Fig. 12.

Note that the generalization of the two-filter algorithm has a very simple structure with considerable symmetry. However, in contrast to the case of a linear tree, the computational structure of the algorithm is far more complex in the case of an arbitrary tree. Specifically, suppose that we wish to obtain the best estimate of x_k at every node of the tree. As we have just seen, the computation for each such node corresponds to breaking the tree into disjoint subtrees by removing the node x_k and by performing recursive processing toward x_k in each subtree. For a linear tree, this leads to *two* recursive filters, one from each extremity to the other, providing *all* of the subtree estimates required for optimal smoothing at *all* x-nodes. For an arbitrary tree, with several extremities, this is obviously *not* the case,

implying that there are many more recursive filters (essentially from each extremity toward every other extremity).

The complexity of the above M-filter smoother motivates the consideration of alternative processing strategies which would be globally optimal for the whole tree. One such structure is the RTS algorithm, which was first derived in [5, 6, 7] for the nonsingular estimation problem over dyadic trees described in Example 5, and was recently extended to arbitrary trees and arbitrary (possibly singular) observations in [28]. This algorithm works as follows. From each extremity of the tree, we start a recursive filter. These recursive filters move towards the center of the tree, processing observations and estimating variables, until a node of degree $M > 2$ is encountered, at which time the filter stops its progression until $M - 1$ filters have reached the node. If the node is an o-node, a merge operation is performed, and if it is an x-node, so that the geometry of Fig. 12 is applicable, the observations o_2, ... , o_N are reduced into a single observation. Then a single filter carrying the merged information of the $M - 1$ filters is sent along the last edge from which a filter has not arrived yet. Proceeding in this manner, there comes a time when M filters meet at a node of degree M, which plays the role of "center" of the tree. These M filters merge their information and start backtracking along their earlier trajectories. At each node of degree $M > 2$, they transmit the merged information to the waiting $M - 1$ filters, which in turn start retracing their earlier paths. The resulting RTS algorithm needs therefore to traverse each edge of the tree only twice, once in each direction, to compute the smoothed estimates at each x-node of the tree, so that it is globally optimal.

Finally, note that the above RTS algorithm can also be applied to 2-D estimation problems, provided that graph transformations of the type described in [23] are employed to give a tree structure to the graph depicted in Fig. 4.

7 Conclusions

In this paper we have developed a general framework for deriving recursive ML estimation algorithms for problems specified by noisy linear relations describing either linear stochastic models or measurements. An xo-graph structure was associated to each estimation problem. It was then shown that if any xo-graph can be reduced to acyclic form through the use of x- or o-aggregation operations, and for any such acyclic form it is possible to derive recursive estimation algorithms for the corresponding reduced estimation problem. The recursive ML estimation algorithms we have developed rely on two elementary operations, called reduction and extraction, which can be used to compress observations, and extract the information about certain variables contained in these observations. The

resulting filtering and smoothing algorithms were illustrated for both linear and arbitrary trees. These results are very general, since they apply to 1-D and multidimensional stochastic systems, systems with singular dynamics, and stochastic processes defined at multiple resolution levels. Furthermore, the procedures employed to perform reduction and extraction operations rely on numerically stable methods, of the same type as to those arising in square-root Kalman filtering, and thus yield numerically reliable estimation techniques. This general framework appears to offer considerable promise for a wide variety of estimation problems, including those for 2-D processes and those involving scale-recursive process descriptions.

References

[1] B.D.O. Anderson and J. B. Moore. *Optimal Filtering.* Englewood Cliffs, NJ: Prentice Hall, 1979.

[2] R.P. Roesser. "A discrete state-space model for linear image processing," *IEEE Trans. Automat. Control* **20** (Feb. 1975), 1–10.

[3] E. Fornasini and G. Marchesini. "State-space realization theory of two-dimensional filters," *IEEE Trans. Automat. Control* **21** (1976), 484–491.

[4] B.C. Levy, M.B. Adams, and A.S. Willsky. "Solution and linear estimation of 2-D nearest-neighbor models," *Proc. IEEE* **78** (April 1990), 627–641.

[5] K.C. Chou, A.S. Willsky, A. Benveniste, and M. Basseville. "Recursive and iterative estimation algorithms for multi-resolution stochastic processes," in *Proc. 28th IEEE Conf. Decision and Control*, Tampa, FL., Dec. 1989, pp. 1184–1189.

[6] K.C. Chou. "A stochastic modeling approach to multiscale signal processing," Ph. D. thesis, Dept. of Electrical Engineering and Computer Science, and Report LIDS-TH-2036, Laboratory for Information and Decision Systems, M.I.T., Cambridge, MA, May 1991.

[7] M. Basseville, A. Benveniste, K. Chou, S. Golden, R. Nikoukhah, and A.S. Willsky. "Modeling and Estimation of Multiresolution Stochastic Processes," *IEEE Trans. Inform. Theory* **38** (April 1992), 766–784.

[8] R. Nikoukhah, A.S. Willsky, and B.C. Levy. "Kalman filtering and Riccati equations for descriptor systems," *IEEE Trans. Automat. Control* **37** (Sept. 1992), 1325–1342/.

[9] D. Taylor and A.S. Willsky, "Maximum likelihood estimation for two-point boundary-value descriptor systems," in *Proc. 1991 Conf. Information Sciences and Systems*, The Johns Hopkins Univ., March 1991.

[10] L. Chisci and G. Zappa. "Square-root Kalman filtering of descriptor systems," *Systems Control Lett.* **19** (1992), 325–334.

[11] P. Whittle. *Prediction and Regulation by Linear Least-Square Methods*, 2nd ed. Minneapolis, MN: University of Minnesota Press, 1983.

[12] G.J. Bierman. *Factorization Methods for Discrete Sequential Estimation*. New York: Academic Press, 1977.

[13] G. Chartrand and L. Lesniak. *Graphs and Digraphs*, 2nd ed. Monterey, CA: Wadsworth, 1986.

[14] R. Nikoukhah, A.S. Willsky, and B.C. Levy. "Boundary-value descriptor systems: well-posedness, reachability and observability," *Internat. J. Control* **46** (1987), 1715–1737.

[15] R. Nikoukhah. "A deterministic and stochastic theory for two-point boundary-value descriptor systems," Ph.D. thesis, Dept. of Electrical Engineering and Computer Science, and Report LIDS-TH-1820, Laboratory for Information and Decision Systems, M.I.T., Cambridge, MA, 1988.

[16] S.L. Lauritzen and D.J. Spiegelhalter. "Local computations with probabilities on graphical structures and their application to expert systems (with discussion)," *J. Royal Stat. Soc., Series B* **50** (1988), 157–224.

[17] M. Luettgen, W.C. Karl, A.S. Willsky, and R.R. Tenney. "Multiscale representations of Markov random fields," *IEEE Trans. Signal Proc.* **41** (Dec. 1993), 3377–3396.

[18] C.R. Rao. "Inference from linear models with fixed effects: recent results and some problems," in *Statistics: an Appraisal*, (H.A. David and H.T. David, eds.). Ames, IA: Iowa State Univ. Press, 1984, pp. 345–369.

[19] S.L. Campbell and C.D. Meyer. *Generalized Inverses of Linear Transformations*. London: Pitman, 1979.

[20] B.C. Levy, A. Benveniste, and R. Nikoukhah. "High-level primitives for recursive maximum-likelihood estimation," Technical report no. 767, Institut de Recherche en Informatique et Systèmes Aléatoires, Rennes, France, Oct. 1993.

[21] L. Dai. "Filtering and LQG problems for discrete-time stochastic singular systems," *IEEE Trans. Automat. Control* **34** (Oct. 1989), 1105–1108.

[22] X-M. Wang and P. Bernhard. "Filtrage et lissage des systèmes implicites discrets," Technical report no. 1083, Institut National de Recherche en Informatique et Automatique, Rocquencourt, France, Aug. 1989.

[23] D. Taylor. "Parallel estimation on one and two dimensional systems," Ph.D thesis, Dept. of Electrical Engineering and Computer Science, and Report LIDS-TH-2092, Laboratory for Information and Decision Systems, M.I.T., Cambridge, MA, 1992.

[24] G.H. Golub and C.H. Van Loan. *Matrix Computations*, 2nd edition. Baltimore, MD: The Johns Hopkins Univ. Press, 1989.

[25] D.Q. Mayne. "A solution of the smoothing problem for linear dynamic systems," *Automatica* **4** (Nov. 1966), 72–92.

[26] D.C. Fraser. "A new technique for the optimal smoothing of data," D.Sc. Dissertation, Dept. Aeronautics and Astronautics, M.I.T., Cambridge, MA, Jan. 1967.

[27] H.E. Rauch, F. Tung, and C.T. Striebel. "Maximum likelihood estimates of linear systems," *AIAA Journal*, **3** (Aug. 1965), 1445–1450.

[28] E. Fabre. "Lissage de processus Markoviens sur un graphe," Technical report no. 799, Institut de Recherche en Informatique et Systèmes Aléatoires, Rennes, France, Feb. 1994.

INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex, France

Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts, 02139

Dept. of Electrical and Computer Engineering, University of California, Davis, CA, 95616

Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts, 02139

Communicated by Anders Lindquist