NEW RESULTS IN 2-D SYSTEMS THEORY, 2-D STATE-SPACE MODELS — REAL-ETC(U)
1976 B LEVY, S Y KUNG, M MORF
AFOSR-TR-77-0370
F44620-69-C-0101
NL
NEW RESULTS IN 2-D SYSTEMS THEORY, 2-D STATE-SPACE MODELS - REALIZATION AND THE NOTIONS OF CONTROLLABILITY, OBSERVABILITY AND MINIMALITY

by B. Levy, S.Y. Kung and M. Morf
Information Systems Laboratory, Stanford University, Stanford, California 94305

AFOSR - TR-77-0370

Abstract

A short comparison between the different state-space models is presented. We discuss proper definitions of state, controllability and observability and their relation to minimality of 2-D systems. We also present new circuit realizations of 2-D filters and for 2-D digital filter hardware implementations of 2-D transfer functions, as well as a 2-D generalization of Levinson's algorithm.

1. Introduction

A short comparison between the different state-space models is presented. We discuss proper definitions of state, controllability and observability and their relation to minimality of 2-D systems. We also present new circuit realizations of 2-D filters and for 2-D digital filter hardware implementations of 2-D transfer functions, as well as a 2-D generalization of Levinson's algorithm.

2. State-Space Models for 2-D Systems

During recent years, several authors: Attasi [1], [2], Fornasini-Marchesini [3], [4] and Givone-Roesser [5] have proposed different state-space models for 2-D systems. In [4], Fornasini and Marchesini were using the algebraic point of view of nonde equivalence and were the first to realize that a major difference between 1-D and 2-D systems is that we can introduce a global state and a local state in the 2-D case. The global state (which is of infinite dimension in general) preserves the past information while the local state gives us the size of the recursions to be performed at each step by the 2-D filter. However, their state does not obey a first-order difference equation (the notion of first-order difference equation for linear systems on partially ordered sets has been defined by Mullins and Elliott in [7]). Attasi's model [1], [2] suffers from the same drawback.

Givone and Roesser in [8] and [5] have used a "circuit approach" to the problem of state space realization for some 2-D transfer functions. They present a model in which the local state is divided into an horizontal and a vertical state which are propagated respectively horizontally and vertically by first-order difference equations.

3. Circuit Realizations and Hardware Designs

First, we can note that the notion of "dynamic elements", "multipliers" and "adders" is at the center of circuit theory. In the 1-D discrete-time case, the elements used are (time) delay...
elements. The 1-D realization problems have been well studied and, given any transfer function, it is well known that the realization can be readily found in certain standard (e.g. controller canonical) forms [11]. For the realization of a 2-D transfer function, a major difference is that two types of dynamic elements are needed - "horizontal delay element" ($z^{-1}$) and "vertical delay element" ($\omega^{-1}$). Now an important problem is that of how to use 2-D dynamic elements, multipliers and adders to realize a 2-D digital filter with the transfer function:

$$H(z^{-1},\omega^{-1}) = \frac{\sum_{i=0}^{n} b_i z^{-i}\omega^{-j}}{\sum_{i=0}^{m} a_i z^{-i}\omega^{-j}}.$$  (3.1)

We can do this in two steps. First we rewrite (3.1) in a rational-gain representation, i.e.

$$H(z^{-1},\omega^{-1}) = \frac{\sum_{i=0}^{n} b_i(\omega^{-1}) z^{-i}}{\sum_{i=0}^{m} a_i(\omega^{-1}) z^{-i}}.$$  (3.2)

Without loss of generality, we can assume $a_{00} = 1$ and we denote

$$a_0(\omega^{-1}) \equiv 1 + \tau_0(\omega^{-1})$$

Thus, using the 1-D realization technique, we write down a realization, where the gains of the multipliers are represented in $F[\omega^{-1}]$.

Figure 3.1

The realization is almost achieved; in addition to the $n$ horizontal delay elements we need only $m$ vertical delay elements to implement the feedback gains ($a_i(\omega^{-1})$, $i=0, 1...m$) and $m$ other vertical delay elements to implement the readout gains ($b_i(\omega^{-1})$, $i=0, 1...m$). Thus, the complete realization shown in Figure 3.2 requires only $n+2m$ dynamic elements.

This realization is a standard (canonical) one; its structure is very simple and it involves only real gains. Note also that we need fewer dynamic elements than the implementations in [9].

**State Space Model Representation**

As remarked in Section 2, circuit implementations with delay elements $z^{-1}$ and $\omega^{-1}$ are in a one-to-one correspondence with state space models of Roesser's type. The output of the $z^{-1}$ delays are the horizontal states and the outputs of the $\omega^{-1}$ delays are the vertical states.

Thus, the implementation of Figure 3.2 can be transformed readily into the following state space model:

$$\begin{align*}
x_h(1+i, j) &= [x_h(1+i, j) x_v(1+i, j)] + b u(i, j) \\
x_v(1+i, j) &= [x_v(1+i, j)]
\end{align*}$$  (3.3)

where

$$A = \begin{bmatrix} A_{11} & -b_0 e_1 & 0 \\
A_{12} & B & 0 \\
A_{13} & -b_0 e_1 & C
\end{bmatrix}$$

with

$$A_{11} \Delta \begin{bmatrix} z_n \Delta a_0 & 0 & \cdots & 0 \\
a_1 & a_2 & \cdots & 0 \\
a_2 & \cdots & \cdots & \cdots \\
s_0 & \cdots & \cdots & 0
\end{bmatrix}$$

and

$$\begin{align*}
A_{1j} &\Delta \begin{bmatrix} 1 & 1, 1, \cdots, 1 & 1, 1, \cdots, 1 \\
1 & 1, 1, \cdots, 1 & 1, 1, \cdots, 1 \\
1 & 1, 1, \cdots, 1 & 1, 1, \cdots, 1 \\
1 & 1, 1, \cdots, 1 & 1, 1, \cdots, 1
\end{bmatrix} \\
B_{1j} &\Delta \begin{bmatrix} 1 & 1, 1, \cdots, 1 & 1, 1, \cdots, 1 \\
1 & 1, 1, \cdots, 1 & 1, 1, \cdots, 1 \\
1 & 1, 1, \cdots, 1 & 1, 1, \cdots, 1 \\
1 & 1, 1, \cdots, 1 & 1, 1, \cdots, 1
\end{bmatrix} \\
C_{1j} &\Delta \begin{bmatrix} 1 & 1, 1, \cdots, 1 & 1, 1, \cdots, 1 \\
1 & 1, 1, \cdots, 1 & 1, 1, \cdots, 1 \\
1 & 1, 1, \cdots, 1 & 1, 1, \cdots, 1 \\
1 & 1, 1, \cdots, 1 & 1, 1, \cdots, 1
\end{bmatrix}
\end{align*}$$

(\text{with} \ a_{00} = 1, \ a_{01}, \ \cdots, \ a_{0m} \ \text{and} \ b_{00} = 1, \ b_{01}, \ \cdots, \ b_{0m})
state space models can be found in [18].

Hardware Design of 2-D Digital Filter

The idea of using two types of dynamic elements is not very abstract; it is very natural in delay-differential systems. However, before considering its practical application to image systems, two remarks have to be made:

1) Because the "spatial" dynamic elements seem unimplementable, (except as index operators in a digital computer, for example,) we can replace them by time-delay elements.

2) In order to have a finite order description, we shall only consider a bounded frame system, i.e. we assume that the picture frame of interest is an MxN frame (with vertical width M and horizontal length N).

Note that in order to use time delay elements we need first to find a way to code a 2-D spatial system into a 1-D (discrete-time) system and vice versa. Thus we shall propose the following implementation of a 2-D filter:

i) The input scan generator codes the 2-D spatial input into 1-D (time) data according to the mapping function t(,).

\[ t(i,j) = iM + jN \]  (3.4)

where M and N are relatively prime integers.

ii) A 1-D (discrete-time) digital filter processes the 1-D data generated by (i). This subsystem is implemented by replacing \( z^{-1} \) by \( \delta \) , \( \omega^{-1} \) by \( \Delta \) in a 2-D circuit realization (e.g. 2-D controller form). \( \delta \) and \( \Delta \) are chosen as

\[ \delta = D^M \text{ N-units delay element,} \]
\[ \Delta = D^N \text{ N-units delay element.} \]  (3.5)

iii) The output frame generator decodes the 1-D (discrete-time) output of the 1-D digital filter described above into a 2-D (discrete-spatial) picture according to the inverse mapping of (3.4).

\[ (i(t), j(t)) = (P \text{ mod } M, (t - (P \text{ mod } M)N)/N) \]  (3.6)

where \( P \) is the unique integer such that

\[ PN - QM = 1 \text{ and } 0 < P < M \]  (3.7)

Verification: Let us note the 1-D (discrete-time) output will be

\[ y(D) = H(D)u(D) \]
\[ = H(z^{-1},\omega^{-1})u(z^{-1},\omega^{-1}) \]
\[ = \sum_{i,j} y_{i,j} z^{-i} \omega^{-j} \]  (3.8)

where \( \{y_{i,j}\} \) represents the 2-D (discrete-spatial) output data field. Note also that

\[ y(D) \Delta = \sum_{t} y_{i,j} D^{-t} \]  (3.9)

Since the system is a causal system,

\[ y_{i,j} = 0 \text{ if } i,j < 0 \]  (3.11)

Let us consider only the integer \( t \) with \( iM + jN \), \( 1 < N, j < M \) then (3.10) and (3.11) give

\[ y_{t} = y_{i,j} \]

since, for this special case, the summation set of (3.10) contains only one nonzero point. Therefore, we will obtain a bona fide output picture inside the MxN frame.

This 2-D image scanning and display system is not as complicated as it looks, it can be simple:

Example: Problem: Design a 2-D digital filter for

\[ H(z^{-1},\omega^{-1}) = \frac{1}{1+0.3z^{-1}+0.2\omega^{-1}+0.1z^{-1}w^{-1}} \]

for a frame: MxN = 100x101. Assume D=0.01 ms.

Solution. (i) ISC

In this special frame (with M=11+1), the input scanning generator is indeed very simple, as shown in Figure 3.3.

Scanning time: 0.01 ms/pixel = 1 ms/line
Scanning angle: 45°

Fig. 3.4 Input Scan Generator & Output Frame Generator

(ii) 1-D digital filter

Constructing the 2-D realization of Figure 3.2 and then replacing \( z^{-1} \) by \( \delta \) and \( \omega^{-1} \) by \( \Delta \) we have the 1-D realization shown in Figure 3.4.
The output frame generator does the reverse of the TSO, displaying the picture instead of scanning.

**Dimensionality of Global State**

Considering a bounded frame (MxN) system, it is interesting to know the dimension of the global state (or initial conditions) needed to process the MxN "future" data field. Since vertical states convey information vertically, all the vertical states along the X-axis are necessary initial conditions and their dimension is mW. Similarly, all the horizontal states along the Y-axis are necessary initial conditions (with dimension mN) since they convey information horizontally. Therefore, in the bounded frame case a total number of mN+mW is needed to summarize the "past" information.

This very same idea can be used again from a computational point of view. Indeed, the number of required storage elements for recursive computations is also equal to mN+mW if initial conditions are not zero. However, the initial conditions are often zero, then the size of storage required can be reduced to mN (resp. mW) by storing the updated data row by row (resp. column by column). No storage is needed for the rest of the initial conditions - mW horizontal states (resp. mN vertical states) - since they are assumed to be zero. This is consistent with the result of Read [12] derived from a direct polynomial approach.

Another interesting observation concerns the dimension of the 1-D digital filter contained in our 2-D digital filter design discussed above. Since it needs n N-unit-delays and m W-unit-delays, the corresponding 1-D state-space has also a dimension equal to mN+mW. Note that, despite the high dimension of the corresponding 1-D filter, its high sparsity is very encouraging for further studies.

In short, our studies on the dimensionality of 2-D global states have reached a consistent conclusion from either theoretical or practical approaches.

4. **Global and Local Controllability and Observability**

For reasons of space we deferred this Section to [18], part II.

5. **Modal Controllability (Observability) and Minimality**

In the 1-D case, the relative primeness concepts could also be used to define controllability and observability. In [16] Rosenbrock proved that A, B was controllable if and only if zI-A, B were left coprime.

C, A was observable if and only if zI-A were right coprime.

This approach can be generalized very easily to 2-D systems and will also provide a definition of minimality.

**Definition 5.1** Let $H(z,\omega) = VT^{-1}U$ where V, T, U are 2-D polynomial matrices. It is a minimal description of $H(z,\omega)$ if and only if V, T are right coprime and T, U are left coprime.

This amounts to requiring that there is no cancellation in the 2-D transfer function $H(z,\omega)$. In [18] part I we also provide the important property that if $(V, T, U)$ and $(V_1, T_1, U_1)$ are two minimal descriptions of $H$, $[T] = [T_1]$. We also presented an algorithm to extract the greatest common right (left) divisor of two polynomial matrices, which enables us to find a minimal description of $H$ from a nonminimal one.

**Definition 5.2**

(1) $A, B$ is modally controllable if and only if $A(z,\omega), B$ are left coprime (5.1) and $C, (A(z,\omega))]$ are right coprime (5.2)

(2) $C, A(z,\omega)$ is modally observable if (5.2) holds.

These definitions are clearly connected to minimality but the state space significance of controllability and observability disappears in this formulation. This is why we shall give now an equivalent state-space characterization of the notions of modal controllability and observability. Another consequence is that for a single input-single output system, if $H(a,\omega) = b(a,\omega)$ and if $b$ and $a$ are coprime with $\Delta = n\omega_2 = m$, then if $CA_{pq}(z,\omega)$ is a minimal realization of $H(z,\omega)$ we must have $|A(z,\omega)| = a(z,\omega)$ and hence $p = n$ and $q = m$.

Hence the validity of our definition of minimality of a state-space model will depend on our ability to realize a transfer function of order $(n, m)$ with $n+m$ states. This problem was considered in Section 6 of [18], part II.

A consequence of the relative primeness criterion for 2-D polynomial matrices given in [18], part I is that $C$ and $A(z,\omega)$ are right coprime if and only if $\text{rank } [A(z,\omega)] = n+m$ for any generic point $(\xi_1, \xi_2)$ of any irreducible algebraic curve $V_1$ appearing in the decomposition of $V$, the algebraic curve defined by $|A(z,\omega)| = 0$. It is to be noted that the rank is considered over the field $K(\xi_1, \xi_2)$. A proof of this is given in [18], part II, along with some illustrating examples.

6. **Minimality of State-Space Model**

It is shown in the last section and in [18], part II, that only a state-space realization with order $(n, m)$ - i.e. the same order as the transfer function - can be both modally controllable and modally observable. Now the question is whether such a realization exists at all.

The best way to prove the existence of such realization is by construction. Note that, in the
2-D state-space model, the particular transform
\[
\begin{bmatrix}
    x' \\
    x''
\end{bmatrix} =
\begin{bmatrix}
    T_{xy} & 0 \\
    0 & T_{xy}
\end{bmatrix}
\begin{bmatrix}
    x' \\
    x''
\end{bmatrix} = T
\begin{bmatrix}
    x' \\
    x''
\end{bmatrix}
\tag{6.1}
\]

enables us to change the basis of the state-space.

The matrices \(A, B, C, D\) are transformed to
\[
\begin{align*}
    A &= T_{xy} A T^{-1} \\
    B &= TB \\
    C &= C T^{-1} \\
    D &= D
\end{align*}
\tag{6.2}
\]

In fact, it is more convenient to work with a canonical form under the "similarity transform" defined by (6.2).

In the 1-D case, all minimal state-space model can be transformed to the controller canonical form. Similarly, almost all \(2\)-D state-space model can be transformed to the following modal controller form \((A, B, C)\) (assuming \(D = 0\))
\[
\begin{bmatrix}
    z_{11} & z_{12} & \cdots & z_{1m} \\
    \vdots & \ddots & \cdots & \vdots \\
    z_{m1} & \cdots & \cdots & z_{mm}
\end{bmatrix}
\begin{bmatrix}
    B^1 \\
    \vdots \\
    B^m
\end{bmatrix}
\begin{bmatrix}
    C^1 \\
    \vdots \\
    C^m
\end{bmatrix}
\tag{6.3}
\]

where \(z_{ij}, B^i, C^i\) were defined in (3.3) and the entries of \(A_{12}\) and \(A_{21}\) are to be chosen such that
\[
\det[A(z,w)] = a(z,w)
\tag{6.4}
\]

and
\[
\det\begin{bmatrix}
    A(z,w) & B \\
    -C & 0
\end{bmatrix} = b(z,w)
\tag{6.5}
\]

It is easy to check that, in (6.4), the coefficients \(a_{i0}, 0 \leq i < n\) and \(a_{0j}, 0 \leq j < m\) have already been matched. Similarly, in (6.4), the coefficients \(b_{10}, 0 \leq i < n\) and \(b_{0j}, 0 \leq j < m\) have already been matched. Therefore, only \(2nm\) nonlinear equations \(\{a_{1i}, 1 \leq i \leq n, 1 \leq j \leq m\}\) and \(\{b_{1j}, 1 \leq i \leq n, 1 \leq j \leq m\}\) are to be matched. In other words, there are totally \(2nm\) (nonlinear) equations to be satisfied. Coincidently, the number of free parameters in matrices \(A_{12}\) and \(A_{21}\) is also \(2nm\). Therefore it is natural to conjecture that a solution (or, more precisely, a finite number of solutions) should always exist.

Now let us examine the plausibility of this conjecture by taking a low-order example.

Example 6.1 (1,1) order case

For ease of notation, let \(A_{11} = a, A_{21} = b\).

Also (without loss of generality) let us assume that \(b_{10} \neq 0\) (otherwise, we may have to use another canonical form). Then (4) becomes
\[
zw+z_{a10}+z_{t10}+z_{d} = zw+a_1 z+b_1 z_{a10}+z_{t10}+z_{d}
\tag{6.6}
\]

and (6.5) becomes
\[
b_{01} + b_{10} + (a_1 b_{01} + a_0 b_{10} + a_1 b_{10}) = b_{01} + b_{10} + (a_1 b_{01} + a_0 b_{10} + a_1 b_{10})
\tag{6.7}
\]

Since \(b_{10} \neq 0, (6.6)\) and (6.7) have solutions

\[\Sigma \text{Sontag (Univ. of Florida) independently arrived at the same conjecture recently (private communication).}\]
realization. Therefore, it is worth mentioning that some special types of transfer function can be easily realized in (n,m) order real-gain realizations. There are two important special types of transfer functions:

1. with separable denominator.
2. with separable numerator.

Let us first consider the separable denominator case. Assuming

$$H(z^{-1},w^{-1}) = \frac{b(z^{-1},w^{-1})}{a(z^{-1},w^{-1})} = a_0 + \sum_{j=0}^{m} b_j z^{-1} w^{-j}$$

then its circuit realization is shown in Figure 6.1

**Controller-Observer-Type Form**

Secondly, let us consider the separable numerator case, which is to say a system with transfer function

$$\tilde{H}(z^{-1},w^{-1}) = [H(z^{-1},w^{-1})]^{-1} = \frac{\alpha(z^{-1},w^{-1})}{\beta(w^{-1})} = \sum_{i=0}^{n} \sum_{j=0}^{m} b_{ij} z^{-1} w^{-j}$$

(6.17)

At first sight, it seems quite difficult. However, in actuality, the realization can be readily obtained by using the inversion rule by Kung [17]. More precisely, to realize the inverse system of Figure 6.1, we first note that the path "input -- $\tilde{G}_0$ -- $\delta_0$ -- input" is a "feed through" path (i.e., a path connecting input and output with only constant gains). The second step is to invert all the gains and reverse all the arrows on the path (in our case, replace $b_{00}$ by $b_{00}^{-1}$). Lastly, change signs of the gains of the branches which are entering this path. These steps complete the realization of $H'(z^{-1},w^{-1})$, in [18], part II the implementation is given.

**REFERENCES**

Appendix: 2-D Levinson Algorithms: The following set of results were motivated by the problem of determining stability of 2-D recursive filters.

In the 1-D case the connections between stability of orthogonal polynomials and the Levinson recursions are by now well known.

In the 2-D case, Shank conjectured that the least-squares inverse, say \( b(z,\omega) \), of an unstable 2-D polynomial \( a(z,\omega) \) of degree \( n \) in \( z \), \( m \) in \( \omega \) is stable, i.e.,

\[
\sum_{j=0}^{n} b(j) z^j = b(z,\omega) \leq 1/a(z,\omega),
\]

where \( b(z,\omega) \) minimizes

\[
\|1 - a(z,\omega)b(z,\omega)\|^2
\]

or

\[
\|e_0 - a b\|^2
\]

with

\[
e_0 \Delta [1,0,\ldots,0], \quad [1 \times (2n+1)(m+1)]
\]

\[
b' = [b_{00},b_{01},b_{10},\ldots,b_{nm}]
\]

and the Toeplitz block Toeplitz matrix \( A \) containing the coefficients of \( a(z,\omega) \) such that \([A\, b]\) is the vector of the coefficients of the product polynomial \([a(z,\omega)b(z,\omega)]\. Then \( b \) is given by the solution of

\[
\mathbb{R}b = [a' \, b]b = e_0 a_{00} (A.3)
\]

By applying the Levinson (LRW) recursions [19] for block matrices developed by Robinson and Wiggins [20], to (A.3) \( b \) can be obtained from the first column of the block solution \( \mathbb{R}b = [I_m,0,\ldots,0]' \). Or with

\[
w \Delta [1,0,\ldots,0]^t
\]

\[
b(z,\omega) = \sum_{i=0}^{n} b_i(z) \omega^i = w' b(z)
\]

Using the property of the LRW recursions that \([A(z)] \) has its roots inside the unit circle, the g.c.d.

of \([b_i(z)] \), which divides \([B(z)] \) contains a subset of these roots.

We can therefore conclude that the nonprimitive factors of \( b(z,\omega) \)---the contents in \( z \) and \( \omega \)---are indeed stable.

However Genin and Kamp [21] proved that \( b(z,\omega) \), therefore the primitive factors, are in general not stable for \( n,m > 1 \).

A 2-D Levinson Algorithm: Genin and Kamp developed a 2-D generalization of the orthogonal polynomials on the unit circle. We give here an equivalent recursion in the time-domain using a stochastic framework (see, e.g., [19]).

We consider a finite window of a scalar 2-D stationary stochastic process \( \{y_{kl}\}, \ i \in \{0,n\}, j \in \{0,m\} \) with zero mean and covariance

\[
E[y_{ij} y_{kl}] = \delta_{ik} \delta_{jl} - k\delta_{ij} - l
\]

Genin and Kamp [21] independently also developed such recursions (private communication).
\[ \frac{\alpha_r(n,m)}{\alpha_r(n,m)} = \frac{\alpha(n,m)}{\alpha(n,m)} \]

where

\[ \alpha(n,m) = \text{I}_{n+1} \bigotimes_k e(n,m) \]

Note that diagonal entries of block composed of 0th row of \( k \)th block entry of \( \alpha(n,m) \) equal unity.

Also, the first column of \( \alpha_r(n,m) \) and of \( \alpha(n,m) \) are the same. Then by (A.5), (A.8) and (A.8')

\[ E(\alpha(n,m) \bigotimes_k e(n,m)) = \text{I}_{n+1} \]

and also by (A.6) and (A.6')

\[ E(\alpha_r(n,m) \bigotimes_k e_r(n,m)) = [e_r(n,m), e_r(n,m)] \]

are the 2-D Levinson equations, therefore we have \( n+1 \) auxiliary solutions. Also, the first column of \( \alpha_r(n,m) \) (or of \( q_r(n,m) \) since they are the same) corresponds to the 2-D causal estimate of \( y(n,m) \) given \( (0,0) \leq (i,j) < (n,m) \), i.e., it is the \( \rightarrow \) predictor of \( y(n,m) \) (one quadrant-predictor). The last column of \( \alpha_r(n,m) \) gives the \( \rightarrow \) predictor and the last column of \( \alpha(n,m) \) gives the \( \leftarrow \) predictor.

The Levinson Recursions: First define

\[ (J_n \bigotimes J_m) \alpha(n,m) = \alpha_r(n,m), \]

hence

\[ (J_n \bigotimes J_m) \alpha_r(n,m) = [e_r(n,m), e_r(n,m)] \]

and we multiply

\[ \begin{pmatrix} J_n & 0 \\ 0 & J_m \end{pmatrix} \]

on the right and denote

\[ \alpha_r^{\ast}(n,m) = (J_n \bigotimes J_m) \alpha_r(n,m) \]

\[ \alpha_c^{\ast}(n,m) = (J_n \bigotimes J_m) \alpha(n,m) \]

and

\[ \alpha_r^{\ast}(n,m) = (J_n \bigotimes J_m) \alpha_r(n,m) \]

\[ \alpha_c^{\ast}(n,m) = (J_n \bigotimes J_m) \alpha(n,m) \]

Then

\[ \alpha_r^{\ast}(n,m) \alpha_c^{\ast}(n,m) = (e_r(n,m), e_r(n,m)) \]

and

\[ \alpha_r^{\ast}(n,m) \alpha_c^{\ast}(n,m) = (e(n,m), e(n,m)) \]

Now, the 2-D Levinson recursions can be described as follows. Increase in \( n \): \( n \rightarrow n+1, m \rightarrow m \), see Figure A1.

\[ \begin{pmatrix} 0_{n+1,m} \\ \alpha_r(n,m) \\ \alpha_r(n+1,m) \end{pmatrix} = \begin{pmatrix} \alpha(n+1,m) \alpha_c(n+1,m) \end{pmatrix} \]

\[ \begin{pmatrix} \alpha_r(n+1,m) \\ \alpha_c(n+1,m) \\ \alpha_r(n+1,m) \end{pmatrix} = \begin{pmatrix} \alpha_c(n+1,m) \alpha_r(n+1,m) \end{pmatrix} \]

Now, let

\[ \begin{pmatrix} \alpha_r(n,m) \\ \alpha_c(n,m) \end{pmatrix} \]

\[ \begin{pmatrix} \alpha(n+1,m) \\ \alpha_c(n+1,m) \end{pmatrix} \]

and

\[ \begin{pmatrix} \alpha_r(n+1,m) \\ \alpha_c(n+1,m) \end{pmatrix} \]

\[ \begin{pmatrix} \alpha(n+1,m) \\ \alpha_c(n+1,m) \end{pmatrix} \]

and

\[ \begin{pmatrix} \alpha_r(n+1,m) \\ \alpha_c(n+1,m) \end{pmatrix} \]

\[ \begin{pmatrix} \alpha(n+1,m) \\ \alpha_c(n+1,m) \end{pmatrix} \]

where

\[ \begin{pmatrix} \alpha_r(n+1,m) \\ \alpha_c(n+1,m) \end{pmatrix} \]

and the diagonal of the top block of \( \alpha_c(n+1,m) \)

equals Diag(\( \alpha_r(n+1,m) \)) \( \alpha_c(n+1,m) \) so that \( \alpha_c(n+1,m) \)

is just obtained by re-normalizing the columns of
\[ \epsilon^{-1}(n+1,k;n+1,m) \neq 0 \text{ otherwise deleting the } k\text{th} \]
\[ \text{column and } k\text{th row of } \gamma^{(n+1,m)}, \text{ we could find } \gamma_k: \gamma^{(n+1,m)}_k = 0. \]
But \( \gamma^{(n+1,m)}_k \) is a covariance and this would mean that the estimation problem is singular.

\[ \epsilon^{(n+1,k;n+1,m)} \neq 0 \text{ otherwise} \]
there would be \( \gamma^{(n+1,m)}_k = 0. \) Also

\[ \gamma^{(n+1,m)}_r = \begin{bmatrix} 0^{m+1}_n \\ \vdots \\ 0^{m+1}_{n+1} \\ \vdots \\ \vdots \\ 0^{m+1}_n \\ \end{bmatrix} \] \[ \gamma^{(n+1,m)}_c = \begin{bmatrix} \beta^{(n,m)}_c \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \beta^{(n,m)}_c \\ \end{bmatrix} \]
\[ \gamma^{(n+1,m)}_r = \begin{bmatrix} 1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 1 \\ \end{bmatrix} \]
and let

\[ \gamma^{(n+1,m)}_r = \gamma^{(n,m)}_r - \gamma^{(n+1,m)}_c \beta^{(n,m)}_c \gamma^{(n,m)}_r \] (A.17)

then

\[ \gamma^{(n+1,m)}_r = \begin{bmatrix} \gamma^{(n+1,m)}_c \\ \gamma^{(n+1,m)}_r \\ \end{bmatrix}, \text{ (n+2 columns) (A.18)} \]

where \( \gamma^{(n+1,m)}_c \) is the first column of \( \gamma^{(n+1,m)}_c \).

To obtain the recursions for an increase in \( m \), we just have to reorder \( \gamma^{(n,m)}_c \) in blocks of size \((n+1)x(n+1)\) then the roles of \( m \) and \( n \) are exchanged as well as \( \gamma^{(n+1,m)}_c \) and \( \gamma^{(n+1,m)}_r \) and we can use the same recursion as the one just described.

This version of the recursion enables us to increase \( n \) and \( m \) separately, instead of the scheme proposed by Genin and Kamp where \( m = n \).

Note 3: The inversions required by these recursions have additional structure, i.e., the matrices are typically non-Toeplitz, but sums of products of Toeplitz matrices. One can take advantage of such a structure by using generalized Levinson recursions [22] to find a representation of the inverse of such matrices also in terms of sums of products of Toeplitz matrices. Expressions with Toeplitz matrices, since they are related to convolutions, can be evaluated using Fast Fourier Transforms (FFT's)
A short comparison between the different state-space models is presented. We discuss proper definitions of state, controllability and observability and their relation to minimality of 2-D systems. We also present new circuit realizations and 2-D digital filter hardware implementations of 2-D transfer functions, as well as a 2-D generalization of Levinson's algorithms.