A State-Space Approach to Orthogonal Digital Filters

Uday B. Desai, *Member, IEEE*

***Abstract*—Using a state-space approach, a new algorithm for designing MIMO orthogonal digital filters is developed. The algorithm consists of three parts: (i) orthogonal embedding, (ii) transformation of the embedded orthogonal transition matrix to the α-*extended upper Hessenberg* form, and (iii) factorization of this new form into Givens (planar) rotations. Appropriately interconnecting the rotors leads to the pipelined orthogonal filter structure. As a consequence of our approach, for the SISO case, an essentially orthogonal structure is obtained for the inverse filter; only one of the Givens rotors gets replaced by a hyperbolic rotor.**

I. Introduction

Orthogonal digital filters are lossless digital filters possessing some very nice properties like low sensitivity to finite precision arithmetic, absence of limit cycle and overflow oscillations, stability in spite of parameter quantization, and VLSI implementability. The results presented in this paper are motivated by the pioneering work on orthogonal digital filters of Deprettere, Dewilde [1], Dewilde, Deprettere, and Nouta [2], Deprettere, Dewilde, and Rao [3], Rao and Kailath [4], [5], Henrot and Mullis [6], and Vaidyanathan [7]. In this context we would also like to mention the work of Regalia, Mitra, and Vaidyanathan [11], Vaidyanathan [12], and the book by Roberts and Mullis [13]. We shall use the definition of orthogonal digital filters as introduced in [1]–[4], and restrict our attention to the kind of pipelined orthogonal digital filter structure developed in Rao and Kailath [4].

In this paper we develop a new algorithm for designing orthogonal digital filters using a purely state-space approach; this is in contrast to [1]–[4], which use the transfer function approach. There are several reasons that motivate the state-space approach, some of which are:

i) It leads to an algorithm that we believe will have better numerical properties than the ones reported thus far. Our algorithm involves applying orthogonal transformations that are known to be numerically very reliable.

ii) Using the state variable notation the generalization to multi-input multi-output (MIMO) filters and time-varying filters is relatively simple.

iii) There is of course the pedagogical reason; the state-space approach will provide insight that is not provided by the transfer function approach.

The algorithm involves three major steps:

i) orthogonal embedding; this step is similar to [1]–[4];

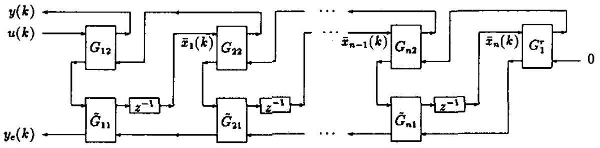
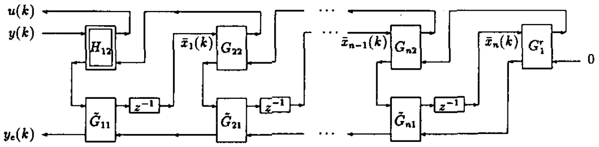


Fig. 1. SISO orthogonal filter structure.



$$\left(G_{ij}=\begin{bmatrix} c_{ij} & s_{ij} \\ -s_{ij} & c_{ij}\end{bmatrix}, \tilde{G}_{ij}=\begin{bmatrix} 0 & 1 \\ 1 & 0\end{bmatrix}G_{ij}\right)$$

Fig. 2. SISO inverse filter structure.

ii) transformation of the embedded orthogonal transition matrix to a special form referred to as the *α-extended upper Hessenberg* form;

iii) factorization of this new form into product of Givens rotations. Appropriately interconnecting the rotors with proper delay elements leads to the pipelined orthogonal filter structure depicted in Fig. 1 for the single-input single-output (SISO) case, and Fig. 3 for the MIMO case.

As a byproduct of the state-space approach, in the SISO case we obtain an essentially orthogonal structure for the inverse filter in a very simple fashion; only one of the Givens rotors is inverted to a hyperbolic rotor; the rest of the structure remains the same (Fig. 2).

We would like to remark that the transformation approach developed here is motivated by the similar work of Mullis and Roberts [13]. Also, some other aspects of the state-space approach to orthogonal digital filters are discussed in Rao [8] and Rao and Dewilde [9]. The state-space approach was also used by Donganata, Vaidyanathan, and Nguyen [14] for the synthesis of FIR lossless transfer matrices. The *discrete-time bounded real lemma* used by us is intimately related to Anderson's *bounded real lemma* [18].

II. Orthogonal Embedding

Consider a MIMO digital filter represented in the state-space form as

$$\begin{bmatrix} y(k) \\ x(k+1)\end{bmatrix}=\underbrace{\begin{bmatrix} D & C \\ B & A\end{bmatrix}}_{\mathcal{A}}\begin{bmatrix} u(k) \\ x(k)\end{bmatrix} \tag{1}$$

Manuscript received June 12, 1989; revised July 2, 1990. This paper was recommended by Associate Editor E. J. Coyle.

The author is with the Department of Electrical Engineering, Indian Institute of Technology, Powai, Bombay 400 076, India.

IEEE Log Number 9041053.

0098-4094/91/0200-0160$01.00 ©1991 IEEE

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY BOMBAY. Downloaded on November 3, 2008 at 23:27 from IEEE Xplore. Restrictions apply.

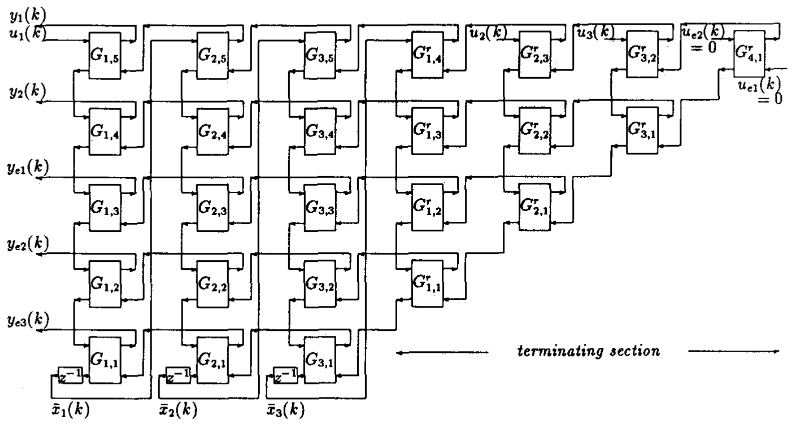


Fig. 3. MIMO orthogonal filter structure (three states, three inputs, two inputs).

where $u(\cdot)$ and $y(\cdot)$ are, respectively, $m \times 1$ input and $p \times 1$ output vectors; $x(\cdot)$ is an $n \times 1$ state vector; and A, B, C, D are, respectively, $n \times n, n \times m, p \times n, p \times m$ real matrices. We refer to A as the transition matrix. The transfer function of (1) is

$$H(z) = D + C(zI - A)^{-1}B.$$

Realization (1) is said to be *orthogonal* if a) $H(z)$ is stable ($|\lambda(A)| < 1$), b) $H(z)$ is real for real z, and c) $H'(e^{-j\omega})H(e^{j\omega}) = I_p$ for $\omega \in [0, 2\Pi]$ (primes will denote matrix transpose, and I_p denotes a $p \times p$ matrix). Note that this definition of orthogonality holds true for $p \leqslant m$. Transfer functions satisfying the above property are also referred to as *all-pass* or *lossless bounded real* transfer functions.

An equivalent characterization for (1) to be orthogonal is that (i) $|\lambda(A)| < 1$, and (ii) $\exists$ an invertible transformation T such that

$$A_T = \begin{bmatrix} D & CT^{-1} \\ TB & TAT^{-1} \end{bmatrix}$$

is orthogonal, i.e., $A_T' A_T = I$.

In general, (1) will not be orthogonal and as such we need to embed it into an orthogonal filter. This idea was first proposed by Deprettere and Dewilde [1]; our formulation is somewhat different since it is based on the state-space approach. In order to achieve this we assume that $H(z)$ is *bounded real*, i.e., a) $H(z)$ is stable ($|\lambda(A)| < 1$), b) $H(z)$ is real for real z, and c) $H'(e^{-j\omega})H(e^{j\omega}) \leqslant I_p$ for $\omega \in [0, 2\Pi]$ if $p \leqslant m$ ($H(e^{-j\omega})H'(e^{j\omega}) \leqslant I_p$ if $p \geqslant m$).

We formulate the orthogonal embedding problem as follows. Consider the digital filter (1) embedded into a square $(m+p)$-input $(m+p)$-output digital filter as

$$\begin{bmatrix} \bar{y}(k) \\ y_e(k) \\ x_e(k+1) \end{bmatrix} = \begin{bmatrix} D & C & \delta_2 \\ \delta_1 & \gamma & \delta_3 \\ B & A & \beta \end{bmatrix} \begin{bmatrix} u(k) \\ x_e(k) \\ u_e(k) \end{bmatrix} \tag{2}$$

where $u_e(k)$ is $p \times 1$, $y_e(k)$ is $m \times 1$, and $x_e(k)$ is the state vector for the embedded filter. By fixing the size of u_e and y_e we are *a priori* fixing the size of the embedding parameters. The intuitive basis for this is what one would have for the SISO case. Nevertheless, the justification follows from Lemma 1 below. Note that when $u_e(k) = 0$, $\bar{y}(k) = y(k)$. In order for (2) to be an orthogonal embedding we need to find the parameters $\beta, \gamma, \delta_1, \delta_2, \delta_3$, and a transformation matrix T such that the embedded transition matrix

$$F_e = \begin{bmatrix} D & CT^{-1} & \delta_2 \\ \delta_1 & \gamma T^{-1} & \delta_3 \\ TB & TAT^{-1} & T\beta \end{bmatrix} \overset{\text{def}}{=} \begin{bmatrix} D & C_e & \delta_2 \\ \delta_1 & \gamma_e & \delta_3 \\ B_e & A_e & \beta_e \end{bmatrix} \tag{3}$$

is orthogonal, i.e.,

$$F_e' F_e = I_{n+m+p}$$
$$F_e F_e' = I_{n+m+p}. \tag{4}$$

Lemma 1: If the transfer function $H(z)$ is bounded real and the corresponding realization $\{A, B, C, D\}$ is minimal, then there exists an invertible transformation T such that F_e is orthogonal; T is given by any factor of N satisfying

$$T'T = N \tag{5}$$

where N satisfies

$$A'NA + C'C + \gamma'\gamma = N. \tag{6}$$

The embedding parameters are given by (superscript $1/2$ denotes the full rank factor, and superscript $T/2$ denotes its transpose, i.e., a symmetric nonnegative definite matrix has the full rank factorization $R = R_1^{1/2} R_1^{T/2}$)

$$\gamma = (N - A'NA - C'C)^{T/2}$$
$$\delta_1 = (I_m - B'NB - D'D)^{T/2}. \tag{7}$$

β, δ_2, and δ_3 satisfy

$$\underbrace{\begin{bmatrix} A'T' & C' & \gamma' \\ B'T' & D' & \delta_1' \end{bmatrix}}_{\overset{\text{def}}{=} \Phi} \begin{bmatrix} T\beta \\ \delta_2 \\ \delta_3 \end{bmatrix} = \mathbf{0},$$

and

$$\beta' T' T \beta + \delta_2' \delta_2 + \delta_3' \delta_3 = I_p \tag{8}$$

i.e., $T\beta$, δ_2, and δ_3 are p orthogonal basis vectors for the null space of Φ.

Moreover, assuming $I_m - B'NB - D'D$ is invertible, N is given by the symmetric positive definite solution of the

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY BOMBAY. Downloaded on November 3, 2008 at 23:27 from IEEE Xplore. Restrictions apply.

algebraic Riccati equation

$$N = A'NA + (A'NB + C'D)(I_m - B'NB - D'D)^{-1} \cdot (A'NB + C'D)' + C'C. \quad (9)$$

Proof: Since $H(z)$ is bounded real and $\{A, B, C, D\}$ minimal, from Vaidyanathan [10], we know that there exists a symmetric positive definite matrix N, an $m \times n$ matrix γ, and an $m \times m$ matrix δ_1 satisfying

$$\begin{aligned} A'NA + C'C + \gamma'\gamma &= N \\ A'NB + C'D + \gamma'\delta_1 &= 0 \\ B'NB + D'D + \delta_1'\delta_1 &= I_m. \end{aligned} \quad (10)$$

Then, for the moment assuming N is available, we compute γ and δ_1 using the first and the third equations in (10) as full rank factors of $(N - A'NA - C'C)$ and $(I_m - B'NB - D'D)$, respectively. This then leads to (7). Now factor $N = T'T$ and consider the $(n+m)\times(n+m+p)$ matrix Φ defined in (8). Using (10), we see that

$$\Phi\Phi' = \begin{bmatrix} N & \mathbf{0} \\ \mathbf{0} & I_m \end{bmatrix}_{(n+m \times n+m)} \quad (11)$$

is a positive definite matrix. Consequently Φ has a p-dimensional null space. We next define the remaining embedding parameters $[\beta'T' \quad \delta_2' \quad \delta_3']'$ as the p orthogonal basis vectors of the null space of Φ, i.e.,

$$\Phi \begin{bmatrix} T\beta \\ \delta_2 \\ \delta_3 \end{bmatrix} = \mathbf{0}, \qquad [\beta'T' \quad \delta_2' \quad \delta_3'] \begin{bmatrix} T\beta \\ \delta_2 \\ \delta_3 \end{bmatrix} = I_p.$$

Then with the above choice of the embedding parameters it is easily verified that F_e is orthogonal.

In order to obtain the algebraic Riccati equation we substitute (7) in the second equation in (10), this gives

$$A'NB + C'D + (-A'NA - C'C + N)^{1/2} \cdot (I_m - B'NB - D'D)^{T/2} = 0.$$

Now, since we have assumed $(I_m - B'NB - D'D)$ to be invertible, upon multiplying from the right by $(I_m - B'NB - D'D)^{-T/2}$, we obtain

$$(A'NB + C'D)(I_m - B'NB - D'D)^{-T/2} + (-A'NA - C'C + N)^{1/2} = 0.$$

Finally, multiplying the above from the right by its transpose gives the desired equation (9). The size of the various embedding parameters is evident from their computation formulas. □

Remark 1: In the above we assumed that $p \leqslant m$. Now if $p \geqslant m$, the proof can be modified *mutatis mutandis* by assuming $H'(z)$ to be bounded real, i.e.,

$$\begin{bmatrix} A' & C' \\ B' & D' \end{bmatrix}$$

to be the transition matrix and the corresponding transfer function to be bounded real.

It is easily established that if the transition matrix F_e is orthogonal, then the corresponding transfer function satisfies the conditions of orthogonality. In particular, all that is needed is to verify $H'(e^{-j\omega})H(e^{j\omega}) = I_p$ for $\omega \in [0, 2\Pi]$.

III. Transformation and Factorization

Lemma 1 shows how to construct an (embedded) orthogonal F_e. Now, in order to obtain an orthogonal implementation of the digital filter (1), one could factor F_e as a product of various Givens (planar) rotations, as described, for example, in Murnaghan [11]. In the present situation such an approach is not desirable for it may lead to more number of rotations than actually required; moreover, it may or may not yield a pipelined architecture. In order to achieve the desired objective we first transform F_e to a special form, which can be factored into a minimal number of Givens rotations. The following lemma will be useful in achieving the desired transformation.

Lemma 2: Given an $n \times n$ matrix A and an $n \times 1$ vector b, there exists an orthogonal matrix Q such that

$$Q'AQ = H, \qquad Q'b = [* \quad 0 \quad \cdots \quad 0]'$$

where H is an upper Hessenberg matrix.

Proof is given in Appendix A by giving an algorithm for constructing such a Q.

In order to clearly explicate the various transformation and factorization steps we first consider the SISO case and then the MIMO case.

3.1. SISO Orthogonal Structure

For the SISO case, the input and output matrices B and C are replaced by lower case letters b and c. First we apply Lemma 2 to the pair (A_e, b_e), and obtain the transformed embedded matrix F having the structure depicted below.

$$F \overset{\text{def}}{=} \begin{bmatrix} d & c_e Q & \delta_2 \\ \delta_1 & \gamma_e Q & \delta_3 \\ Q'b_e & Q'A_e Q & Q'\beta_e \end{bmatrix} = \begin{bmatrix} * & * & * & \cdots & * & * & * \\ * & * & * & \cdots & * & * & * \\ * & * & * & \cdots & * & * & * \\ 0 & * & * & \cdots & * & * & * \\ 0 & 0 & * & \cdots & * & * & * \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & * & * & * \end{bmatrix}. \quad (12)$$

The above F is very much like an upper Hessenberg matrix except, in general, it has one more nonzero line parallel to the diagonal. Thus if $f_{i,j}$ is the (i,j)th entry of F, then $f_{i,j} = 0$ whenever $i \geqslant j + 2$. We refer to such a matrix as an *α-extended upper Hessenberg matrix*, $\alpha = 1$. α represents a number such that $f_{i,j} = 0$ whenever $i \geqslant j + \alpha + 1$. The transformed embedded filter is

$$\begin{bmatrix} \bar{y}(k) \\ y_e(k) \\ \bar{x}(k+1) \end{bmatrix} = F \begin{bmatrix} u(k) \\ \bar{x}(k) \\ u_e(k) \end{bmatrix}. \quad (13)$$

Note that the original state $x(\cdot) = T^{-1}Q\bar{x}(\cdot)$ and $\bar{y}(k) = y(k)$, when $u_e(k) = 0$.

Our objective now is to obtain a pipelined architecture for the two-input two-output orthogonal filter (13). This is

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY BOMBAY. Downloaded on November 3, 2008 at 23:27 from IEEE Xplore. Restrictions apply.

achieved by appropriately factoring F as a product of Givens rotations. This factorization is obtained in two steps: (i) A sequence of 3×3 orthogonal Hessenberg matrices H_i and a 2×2 orthogonal matrix F_{n+1} are constructed which when applied to F will transform it to an identity matrix; (ii) next the H_i's are factored into products of two Givens rotations; note F_{n+1} determines the terminating section. Orderly collection of all these rotations leads to the desired factorization and the pipelined orthogonal structure. The details are presented below as Algorithm 3.1. Though steps (i) and (ii) can be combined, we have presented them separately to indicate the modular factor H_i, which repeats itself in a regular fashion.

Algorithm 3.1:

1) Solve the algebraic Riccati equation (9) for N, compute the transformation matrix T from (5), and the embedding parameters (7) and (8), and then form the embedded orthogonal transition matrix F_e (3).

   Remark 2: In case $H(z)$ is all-pass, then all the embedding parameters will be zero. Moreover, from (10) it should be clear that N can be computed by solving a Lyapunov equation ($N = A'NA + c'c$). Computation of T remains the same.

2) Using Algorithm A.1 in Appendix A, compute an orthogonal Q that will transform F_e to an extended upper Hessenberg matrix F (12).

3) Let $I_0 = 0, I_i$ be an $i \times i$ identity matrix, and $F_1 \overset{\text{def}}{=} F$. For $i = 1$: n Do (a)-(c):

   a)

$$\rho_i = \left(\left(f^i_{i+1,i}\right)^2 + \left(f^i_{i+2,i}\right)^2 \right)^{1/2}$$

$$H_i = \begin{bmatrix} f^i_{i,i} & f^i_{i+1,i} & f^i_{i+2,i} \\ -\rho_i & \left(f^i_{i,i} f^i_{i+1,i}\right)/\rho_i & \left(f^i_{i,i} f^i_{i+2,i}\right)/\rho_i \\ 0 & -f^i_{i+2,i}/\rho_i & f^i_{i+1,i}/\rho_i \end{bmatrix}$$

where $f^i_{j,k}$ is the $(j-i+1, k-i+1)$th element of F_i, an orthogonal 1-extended upper Hessenberg matrix of size $(n-i+3)\times(n-i+3)$:

$$F_i = \begin{bmatrix} f^i_{i,i} & f^i_{i,i+1} & \cdots & f^i_{i,n} & f^i_{i,n+1} & f^i_{i,n+2} \\ f^i_{i+1,i} & f^i_{i+1,i+1} & \cdots & f^i_{i+1,n} & f^i_{i+1,n+1} & f^i_{i+1,n+2} \\ f^i_{i+2,i} & f^i_{i+2,i+1} & \cdots & f^i_{i+2,n} & f^i_{i+2,n+1} & f^i_{i+2,n+2} \\ 0 & f^i_{i+3,i+1} & \cdots & f^i_{i+3,n} & f^i_{i+3,n+1} & f^i_{i+3,n+2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & f^i_{n+2,n} & f^i_{n+2,n+1} & f^i_{n+2,n+2} \end{bmatrix}.$$

   b)

$$\underbrace{\begin{bmatrix} I_{i-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & H_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{n-i} \end{bmatrix}}_{J_i'} \begin{bmatrix} I_{i-1} & \mathbf{0} \\ \mathbf{0} & F_i \end{bmatrix} = \begin{bmatrix} I_i & \mathbf{0} \\ \mathbf{0} & F_{i+1} \end{bmatrix}.$$

This gives the following factorization for F:

$$F = J_1 J_2 \cdots J_{n-1} J_n J_{n+1} \tag{14}$$

where $J_{n+1} = \begin{bmatrix} I_n & \mathbf{0} \\ \mathbf{0} & G_1^r \end{bmatrix}$, $G_1^r = F_{n+1}$ is a 2×2 orthogonal matrix representing the terminating section.

   c) Factor H_i' as a product of two rotations:

$$H_i' = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & G_{i1} \end{bmatrix} \begin{bmatrix} G_{i2} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}$$

where G_{i1} and G_{i2} are constructed from the elements of H_i as

$$G_{i1} = \frac{1}{\rho_i} \begin{bmatrix} f^i_{i+1,i} & -f^i_{i+2,i} \\ f^i_{i+2,i} & f^i_{i+1,i} \end{bmatrix}$$

$$G_{i2} = \begin{bmatrix} f^i_{i,i} & -\rho_i \\ \rho_i & f^i_{i,i} \end{bmatrix}.$$

Using the above in the definition of J_i, the following factorization is obtained: for $i = 1, \cdots, n$,

$$J_i = J_{i1} J_{i2} = \left[\begin{array}{c|cc|c} I_{i-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & 1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & G_{i1} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & I_{n-i} \end{array}\right] \left[\begin{array}{c|cc|c} I_{i-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & G_{i2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 1 & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & I_{n-i} \end{array}\right].$$

4) Substituting the factorization of J_i in (14), we obtain the final factorization of F:

$$F = J_{11} J_{12} J_{21} J_{22} \cdots J_{n-1,1} J_{n-1,2} J_{n1} J_{n2} J_{n+1}. \tag{15}$$

Use of (15) in (13) leads to the embedded factorized state variable equation

$$\begin{bmatrix} \bar{y}(k) \\ y_e(k) \\ \bar{x}_1(k+1) \\ \vdots \\ \bar{x}_n(k+1) \end{bmatrix} = \left[\begin{array}{cc|c} 1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & G_{11} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & I_{n-1} \end{array}\right] \left[\begin{array}{cc|c} G_{12} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & I_{n-1} \end{array}\right]$$

$$\cdots \left[\begin{array}{c|cc} I_{n-1} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & G_{n1} \end{array}\right] \left[\begin{array}{c|cc} I_{n-1} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & G_{n2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 1 \end{array}\right]$$

$$\cdot \left[\begin{array}{c|cc} I_{n-1} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & G_1^r \end{array}\right] \begin{bmatrix} u(k) \\ \bar{x}_1(k) \\ \vdots \\ \bar{x}_n(k) \\ u_e(k) \end{bmatrix}. \tag{16}$$

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY BOMBAY. Downloaded on November 3, 2008 at 23:27 from IEEE Xplore. Restrictions apply.

Connecting the various rotations in accordance with (16), we obtain the orthogonal pipelined filter structure depicted in Fig. 1. Note, in Fig. 1, G_{ij} and $\tilde{G}_{ij}$ have the form shown below.

$$G_{ij} = \begin{bmatrix} c_{ij} & s_{ij} \\ -s_{ij} & c_{ij} \end{bmatrix}$$

$$\tilde{G}_{ij} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} G_{ij}. \tag{17}$$

$\tilde{G}_{ij}$ is indeed an orthogonal planar rotation, often referred to as a *Givens reflection* or a *Householder matrix* (see [16], p. 44). We have expressed the filter structure of Fig. 1 using two kinds of rotations G_{ij} and $\tilde{G}_{ij}$ simply to indicate that it is possible to have the orthogonal structure without any crossing of the lines.

Remark 3:

a) The transition matrix for each orthogonal section constituted by the pair $(\tilde{G}_{i1}, G_{i2}))$ is a lower Hessenberg matrix H_i' with two inputs, two outputs, and 1 state variable (see Fig. 1).

b) In Algorithm 3.1 we are successively peeling off different layers of the orthogonal filter from the transition matrix F: first $J_{11}(G_{11})$, then $J_{12}(G_{12})$, and so on. In this sense we could view Algorithm 3.1 as a state-space layer peeling algorithm.

c) The above algorithm simplifies considerably when applied to an AR filter. If $1/A(z)$ is the AR transfer function, then we consider the transition matrix corresponding to $z^n A(z^{-1})/A(z)$, and apply Algorithm 3.1 starting with Remark 2, Step 1. We need to solve a Lyapunov equation as opposed to a Riccati equation, and as expected, this leads to the well known AR lattice of Gray and Markel [10].

d) Realization (13), viewed as a two-input two-output filter is a balanced realization with both the Grammians equal to the identity matrix.

3.2. MIMO Orthogonal Structure

The approach to the MIMO case is similar to the SISO case, nevertheless there are some significant differences. For one, a rearrangement of the F_e matrix is needed. Moreover, the factorization involves some additional steps and the structure of the terminating section is very different. We define

$$F_e = \begin{bmatrix} d_1 & C_e & \overline{D} & \delta_2 \\ \delta_{11} & \gamma_e & \bar{\delta}_1 & \delta_3 \\ b_{e1} & A_e & \overline{B}_e & \beta_e \end{bmatrix} \tag{18}$$

where d_1, δ_{11}, and b_{e1}, respectively, represent the first column of D, δ_1, and B; while $\overline{D}$, $\bar{\delta}_1$, and $\overline{B}_e$, respectively, represent the remaining columns of D, δ_1, and B. In this format the MIMO embedded filter has the form

$$\begin{bmatrix} \bar{y}_1(k) \\ \vdots \\ \bar{y}_p(k) \\ y_{e1}(k) \\ \vdots \\ y_{em}(k) \\ x_e(k+1) \end{bmatrix} = F_e \begin{bmatrix} u_1(k) \\ x_e(k) \\ u_2(k) \\ \vdots \\ u_m(k) \\ u_{e1}(k) \\ \vdots \\ u_{ep}(k) \end{bmatrix}. \tag{19}$$

Now we apply Algorithm A.1 to the pair (A_e, b_{e1}) to obtain an F that is analogous to the one depicted in (12), namely

$$F \overset{\text{def}}{=} \begin{bmatrix} d_1 & C_eQ & \overline{D} & \delta_2 \\ \delta_{11} & \gamma_eQ & \bar{\delta}_1 & \delta_3 \\ Q'b_{e1} & Q'A_eQ & Q'B_e & Q'\beta_e \end{bmatrix}$$

$$= \left[\begin{array}{c|ccccc|cc} * & * & * & \cdots & * & * & \cdots & * \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ * & * & * & \cdots & * & * & \cdots & * \\ \hline * & * & * & \cdots & * & * & \cdots & * \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ * & * & * & \cdots & * & * & \cdots & * \\ \hline * & * & * & \cdots & * & * & \cdots & * \\ 0 & * & * & \cdots & * & * & \cdots & * \\ 0 & 0 & * & \cdots & * & * & \cdots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & * & * & \cdots & * \end{array}\right] \begin{array}{l} \left.\vphantom{\begin{array}{c} * \\ \vdots \\ * \end{array}}\right\} p \\ \left.\vphantom{\begin{array}{c} * \\ \vdots \\ * \end{array}}\right\} m \\ \left.\vphantom{\begin{array}{c} * \\ 0 \\ 0 \\ \vdots \\ 0 \end{array}}\right\} n \end{array}. \tag{20}$$

As defined in the SISO case, F is an α-extended Hessenberg matrix with $\alpha = m + p - 1$. Note that the size of F is $(n + m + p) \times (n + m + p)$.

The problem now is to factor F into product of Givens rotations, in a specific order, so that it will lead to a pipelined orthogonal structure. Though a direct algorithm for this can be developed, we prefer to present it in two stages. In the first stage a sequence of $(m + p + 1) \times (m + p + 1)$ Hessenberg matrices H_i and an $(m + p) \times (m + p)$ orthogonal matrix F_{n+1} are computed, such that when these are applied to F, as shown below, the result is an identity matrix. F_{n+1} represents the terminating section. In the second stage each H_i is factored into a product of $(m + p)$ Givens rotations, and F_{n+1} into a product of $(m + p)(m + p - 1)/2$ Givens rotations. We first present Algorithm 3.2 which explicitly specifies the above stages and then present an illustrative example. Perhaps, we are preempting a concluding remark, but nevertheless it is worthwhile to mention that in the MIMO case the terminating section may in fact determine the complexity of the filter structure.

Algorithm 3.2:

1) Solve the algebraic Riccati equation (9) for N, compute the transformation T and the embedding parameters, and form the embedded orthogonal transition matrix F_e (18).

2) Compute Q by applying Algorithm A.1 of Appendix A to the pair (A_e, b_{e1}), and form the matrix F (20).

3) Formation of the Hessenberg matrices and the terminating matrix:

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY BOMBAY. Downloaded on November 3, 2008 at 23:27 from IEEE Xplore. Restrictions apply.

Let $I_0 = 0$, I_i be an $i \times i$ identity matrix, and $F_1 \overset{\text{def}}{=} F$.
For $i = 1$: n do a)–c):
a)

$$\rho_{i,1} = \left[\left(f^i_{i+1,i}\right)^2 + \left(f^i_{i+2,i}\right)^2 + \cdots + \left(f^i_{i+m+p,i}\right)^2 \right]^{1/2}$$

$$\rho_{i,2} = \left[\left(f^i_{i+2,i}\right)^2 + \left(f^i_{i+3,i}\right)^2 + \cdots + \left(f^i_{i+m+p,i}\right)^2 \right]^{1/2}$$

$$\vdots$$

$$\rho_{i,m+p-1} = \left[\left(f^i_{i+m+p-1,i}\right)^2 + \cdots + \left(f^i_{i+m+p,i}\right)^2 \right]^{1/2}$$

$$H_i = \begin{bmatrix} f^i_{i,i} & f^i_{i+1,i} & f^i_{i+2,i} & \cdots & f^i_{i+m+p-1,i} & f^i_{i+m+p,i} \\ -\rho_{i,1} & \dfrac{f^i_{i+1,i} f^i_{i,i}}{\rho_{i,1}} & \dfrac{f^i_{i+2,i} f^i_{i,i}}{\rho_{i,1}} & \cdots & \dfrac{f^i_{i+m+p-1,i} f^i_{i,i}}{\rho_{i,1}} & \dfrac{f^i_{i+m+p,i} f^i_{i+1,i}}{\rho_{i,1}} \\ \mathbf{0} & -\dfrac{\rho_{i,2}}{\rho_{i,1}} & \dfrac{f^i_{i+2,i} f^i_{i+1,i}}{\rho_{i,1}\rho_{i,2}} & \cdots & \dfrac{f^i_{i+m+p-1,i} f^i_{i+1,i}}{\rho_{i,1}\rho_{i,2}} & \dfrac{f^i_{i+m+p,i} f^i_{i+1,i}}{\rho_{i,1}\rho_{i,2}} \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & -\dfrac{\rho_{i,m+p-1}}{\rho_{i,m+p-2}} & \dfrac{f^i_{i+m+p-1,i} f^i_{i+m+p-2,i}}{\rho_{i,m+p-2}\rho_{i,m+p-1}} & \dfrac{f^i_{i+m+p,i} f^i_{i+m+p-2,i}}{\rho_{i,m+p-2}\rho_{i,m+p-1}} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & -\dfrac{f^i_{i+m+p-1,i}}{\rho_{i,m+p-1}} & \dfrac{f^i_{i+m+p-1,i}}{\rho_{i,m+p-1}} \end{bmatrix}$$

where $f^i_{j,k}$ is the $(j-i+1, k-i+1)$th entry of F_i, an $(m+p-1)$-extended upper Hessenberg matrix of size $(n+m+p-i+1) \times (n+m+p-i+1)$.
b)

$$\underbrace{\begin{bmatrix} I_{i-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & H_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{n-i} \end{bmatrix}}_{J_i'} \begin{bmatrix} I_{i-1} & \mathbf{0} \\ \mathbf{0} & F_i \end{bmatrix} = \begin{bmatrix} I_i & \mathbf{0} \\ \mathbf{0} & F_{i+1} \end{bmatrix}.$$

Steps a) and b) give the following factorization for F:

$$F = J_1 J_2 \cdots J_{n-1} J_n J_{n+1}$$

where $J_{n+1} = \begin{bmatrix} I_n & 0 \\ 0 & F_{n+1} \end{bmatrix}$, F_{n+1} of size $(m+p) \times (m+p)$.

c) Using Algorithm A.2 factor each H_i' into a product of $(m+p)$ Givens rotations:

$$H_i' = \begin{bmatrix} I_{m+p-1} & \mathbf{0} \\ \mathbf{0} & G_{i,1} \end{bmatrix} \begin{bmatrix} I_{m+p-2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & G_{i,2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 1 \end{bmatrix} \cdots \begin{bmatrix} G_{i,m+p} & \mathbf{0} \\ \mathbf{0} & I_{m+p-1} \end{bmatrix}$$

where $G_{i,j}$ is a 2×2 Givens rotation matrix.

4) Factorization of the terminating section:
Define $R_1 \overset{\text{def}}{=} F_{n+1}$. Let $r^i_{j,k}$ be the $(j-i+1, k+i+1)$th entry of R_i.

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY BOMBAY. Downloaded on November 3, 2008 at 23:27 from IEEE Xplore. Restrictions apply.

For $i=1$: $m+p-1$ do a)–c):

a)

$$\gamma_{i,i}=\left[\left(r^i_{i+1,i}\right)^2+\left(r^i_{i+2,i}\right)^2+\cdots+\left(r^i_{m+p,i}\right)^2\right]^{1/2}$$

$$\gamma_{i,i+1}=\left[\left(r^i_{i+2,i}\right)^2+\left(r^i_{i+3,i}\right)^2+\cdots+\left(r^i_{m+p,i}\right)^2\right]^{1/2}$$

$$\vdots$$

$$\gamma_{i,m+p-1}=\left[\left(r^i_{m+p-1,i}\right)^2+\cdots+\left(r^i_{m+p,i}\right)^2\right]^{1/2}$$

$$H_i^r=\begin{bmatrix} r^i_{i,i} & r^i_{i+1,i} & r^i_{i+2,i} & \cdots & r^i_{m+p-1,i} & r^i_{m+p,i} \\ -\gamma_{i,i} & \dfrac{r^i_{i+1,i}r^i_{i,i}}{\gamma_{i,i}} & \dfrac{r^i_{i+2,i}r^i_{i,i}}{\gamma_{i,i}} & \cdots & \dfrac{r^i_{m+p-1,i}r^i_{i,i}}{\gamma_{i,i}} & \dfrac{r^i_{m+p,i}r^i_{i,i}}{\gamma_{i,i}} \\ \mathbf{0} & -\dfrac{\gamma_{i,i+1}}{\gamma_{i,i}} & \dfrac{r^i_{i+2,i}r^i_{i+1,i}}{\gamma_{i,i+1}\gamma_{i,i}} & \cdots & \dfrac{r^i_{m+p-1,i}r^i_{i+1,i}}{\gamma_{i,i+1}\gamma_{i,i}} & \dfrac{r^i_{m+p,i}r^i_{i+1,i}}{\gamma_{i,i+1}\gamma_{i,i}} \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & -\dfrac{\gamma_{i,m+p-2}}{\gamma_{i,m+p-3}} & \dfrac{r^i_{m+p-1,i}r^i_{m+p-2,i}}{\gamma_{i,m+p-3}\gamma_{i,m+p-2}} & \dfrac{r^i_{m+p,i}r^i_{m+p-2,i}}{\gamma_{i,m+p-3}\gamma_{i,m+p-2}} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & -\dfrac{r^i_{m+p,i}}{\gamma_{i,m+p-2}} & \dfrac{r^i_{m+p-1,i}}{\gamma_{i,m+p-2}} \end{bmatrix}.$$

b)

$$H_i^rR_i=\begin{bmatrix}1 & \mathbf{0}\\ \mathbf{0} & R_{i+1}\end{bmatrix}.$$

Using a) and b) we obtain the following factorization for R_1:

$$R_1=(H_1^r)'\begin{bmatrix}1 & \mathbf{0}\\ \mathbf{0} & (H_2^r)'\end{bmatrix}\begin{bmatrix}I_2 & \mathbf{0}\\ \mathbf{0} & (H_3^r)'\end{bmatrix}\cdots\begin{bmatrix}I_{m+p-2} & \mathbf{0}\\ \mathbf{0} & (H^r_{m+p-1})'\end{bmatrix}.$$

c) Using Algorithm A.2 factor each $(H_i^r)'$ as a product of $(m+p-i)$ Givens rotations:

$$(H_i^r)'=\begin{bmatrix}I_{m+p-i-2} & \mathbf{0}\\ \mathbf{0} & G^r_{i,1}\end{bmatrix}\begin{bmatrix}I_{m+p-i-3} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & G^r_{i,2} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & 1\end{bmatrix}\cdots\begin{bmatrix}G^r_{i,m+p-i} & \mathbf{0}\\ \mathbf{0} & I_{m+p-i-2}\end{bmatrix}.$$

5) Substituting the factorization of H_i and H_i^r in Step 3b) above, we obtain the final factorization of F as

$$F=J_{1,1}J_{1,2}\cdots J_{1,m+p}\cdots J_{n,1}J_{n,2}\cdots J_{n,m+p}$$
$$J^r_{1,1}\cdots J^r_{1,m+p-1}J^r_{2,1}\cdots J^r_{2,m+p-2}\cdots$$
$$J^r_{m+p-2,1}J^r_{m+p-2,2}J^r_{m+p-1,1}$$

where each $J_{i,j}$ and $J^r_{i,j}$ agree with an $(n+m+p)\times(n+m+p)$ identity matrix, except for a rotation $G_{i,j}$ or $G^r_{i,j}$ in an appropriate 2×2 principal submatrix location. Rotations with a subscript r constitute the terminating section.

Remark 4:

a) Thus from the above factorization, we see that the terminating section requires $(m+p-1)(m+p)/2$ rotors, while the remaining portion requires $n(m+p)$ rotors, giving in all a total of $(n+(m+p-1)/(2))(m+p)$ rotors. The following argument given by Reviewer 1 shows that the proposed embedding has minimum number of input–output lines. A realization with n states, m inputs, and p outputs has $n(m+p)+mp$ independent parameters. From the above factorization of F we see that an orthogonal realization with n states, and r inputs and outputs has $nr+(r-1)r/2$ independent parameters. Thus the requirement that the inequality

$$nr+\frac{(r-1)r}{2}\geqslant n(m+p)+mp$$

hold true for all n leads to $r=m+p$ as the minimal size of the embedding. Thus, with $r=m+p$, the embedded system has an excess of $((m+p-1)(m+p)/2-mp)$ parameters. Intuitively, one expects to have more number of parameters in the orthogonally embedded filter. In the SISO case, surprisingly, this does not happen; both the original and the orthogonally embedded realizations have the same number of parameters, $2n+1$. The intuition is borne out by the MIMO case where one does require extra parameters. Of course the extra parameters provide the required termination to orthogonalize the filter. At present, it is difficult to ascertain the precise significance of these extra parameters.

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY BOMBAY. Downloaded on November 3, 2008 at 23:27 from IEEE Xplore. Restrictions apply.

b) The state variable filter with F as the transition matrix is an $(m+p)$-input $(m+p)$-output balanced realization with both the Grammians equal to identity.

Remark 5: Algorithms 3.1 and 3.2 involve applying orthogonal transformations, which are numerically very reliable. The embedding part involves three key computational steps.

a) Solution of the algebraic Riccati equation.
b) Factorization of N, and full rank factorization of symmetric nonnegative definite matrices to compute γ and δ_1.
c) Finding an orthogonal basis for the null space of Φ.

The Riccati equation has been extensively researched, and consequently there are numerous algorithms available for solving it (for a survey see [19]). We believe that well-tested numerical algorithms are available for solving the Riccati equation, and hence it *may* have advantages over the spectral factorization approach. There are software packages like Matlab or Matrix$_X$ that can be exploited for this task.

N is a symmetric positive definite matrix; now for factorization of such matrices, several numerically reliable algorithms are available (Golub and Van Loan [16]). As an example, one could use the orthogonal eigenvector factorization; [16] also contains good numerical algorithms for computing full rank factors of symmetric nonnegative definite matrices.

A numerically reliable way to compute the orthogonal basis for the null space of Φ is to use the QR decomposition of Φ'.

For the purpose of illustration consider the case of a three state ($n=3$) digital filter with three inputs ($m=3$) and two outputs ($p=2$). In this case F will be a 4-extended upper Hessenberg matrix. Moreover, from Remark 4 we see that in all 25 rotors will be required, 10 of which will be for the terminating section. After implementing Algorithm 3.2 the transformed and factorized state variable structure will be as follows:

$$
\begin{aligned}
\begin{bmatrix} y_1(k) \\ y_2(k) \\ y_{e1}(k) \\ y_{e2}(k) \\ y_{e3}(k) \\ \bar{x}_1(k+1) \\ \bar{x}_2(k+1) \\ \bar{x}_3(k+1) \end{bmatrix}
={}& \begin{bmatrix} I_4 & & \\ & G_{1,1} & \\ & & I_2 \end{bmatrix}
\begin{bmatrix} I_3 & & \\ & G_{1,2} & \\ & & I_3 \end{bmatrix}
\begin{bmatrix} I_2 & & \\ & G_{1,3} & \\ & & I_4 \end{bmatrix}
\begin{bmatrix} 1 & & \\ & G_{1,4} & \\ & & I_5 \end{bmatrix}
\begin{bmatrix} G_{1,5} & \\ & I_6 \end{bmatrix} \\
& \begin{bmatrix} I_5 & & \\ & G_{2,1} & \\ & & 1 \end{bmatrix}
\begin{bmatrix} I_4 & & \\ & G_{2,2} & \\ & & I_2 \end{bmatrix}
\begin{bmatrix} I_3 & & \\ & G_{2,3} & \\ & & I_3 \end{bmatrix}
\begin{bmatrix} I_2 & & \\ & G_{2,4} & \\ & & I_4 \end{bmatrix}
\begin{bmatrix} 1 & & \\ & G_{2,5} & \\ & & I_5 \end{bmatrix} \\
& \begin{bmatrix} I_6 & \\ & G_{3,1} \end{bmatrix}
\begin{bmatrix} I_5 & & \\ & G_{3,2} & \\ & & 1 \end{bmatrix}
\begin{bmatrix} I_4 & & \\ & G_{3,3} & \\ & & I_2 \end{bmatrix}
\begin{bmatrix} I_3 & & \\ & G_{3,4} & \\ & & I_3 \end{bmatrix}
\begin{bmatrix} I_2 & & \\ & G_{3,5} & \\ & & I_4 \end{bmatrix} \\
& \begin{bmatrix} I_6 & \\ & G^r_{1,1} \end{bmatrix}
\begin{bmatrix} I_5 & & \\ & G^r_{1,2} & \\ & & 1 \end{bmatrix}
\begin{bmatrix} I_4 & & \\ & G^r_{1,3} & \\ & & I_2 \end{bmatrix}
\begin{bmatrix} I_3 & & \\ & G^r_{1,4} & \\ & & I_3 \end{bmatrix} \\
& \begin{bmatrix} I_6 & \\ & G^r_{2,1} \end{bmatrix}
\begin{bmatrix} I_5 & & \\ & G^r_{2,2} & \\ & & 1 \end{bmatrix}
\begin{bmatrix} I_4 & & \\ & G^r_{2,3} & \\ & & I_2 \end{bmatrix} \\
& \begin{bmatrix} I_6 & \\ & G^r_{3,1} \end{bmatrix}
\begin{bmatrix} I_5 & & \\ & G^r_{3,2} & \\ & & 1 \end{bmatrix}
\begin{bmatrix} I_6 & \\ & G^r_{4,1} \end{bmatrix}
\begin{bmatrix} u_1(k) \\ \bar{x}_1(k) \\ \bar{x}_2(k) \\ \bar{x}_3(k) \\ u_2(k) \\ u_3(k) \\ u_{e1}(k) \\ u_{e2}(k) \end{bmatrix}
\end{aligned}
$$

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY BOMBAY. Downloaded on November 3, 2008 at 23:27 from IEEE Xplore. Restrictions apply.

with $u_{e1}(\cdot)=0$ and $u_{e2}(\cdot)=0$. Implementing the above equation using rotors and delay units leads to the MIMO orthogonal structure of Fig. 3. In the MIMO case it does not seem possible to avoid crossing of the lines, and as such, unlike the SISO case we have not made use of $\tilde{G}_{i,j}$. Note that when $u_{ei}(\cdot)=0$, the original state vector $x(\cdot)=T^{-1}Q'\bar{x}(\cdot)$.

IV. Inverse Filter Structure for the SISO Case

A rather interesting feature of the filter structure depicted in Fig. 1 is that the inverse filter structure is very easily obtained. All one has to do is replace the Givens rotor G_{12} by a hyperbolic rotor H_{12} and change the direction of the input and output; this is depicted in Fig. 2.

To explicate this, consider the orthogonally factored state variable filter (16) with $u_e(k)=0$. Then, for the first section consisting of G_{11} and G_{12} we have the following input–output relationship:

$$\begin{bmatrix} y(k) \\ y_e(k) \\ \bar{x}_1(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_{11} & s_{11} \\ 0 & -s_{11} & c_{11} \end{bmatrix} \begin{bmatrix} c_{12} & s_{12} & 0 \\ -s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u(k) \\ \alpha_{12} \\ \alpha_{11} \end{bmatrix} \tag{21}$$

where α_{11} and α_{12} are the inputs to G_{11} ($\tilde{G}_{11}$) and G_{12}, and are obtained as outputs of G_{21} ($\tilde{G}_{21}$) and G_{22}, respectively. Now, using some simple algebra the roles of $u(\cdot)$ and $y(\cdot)$ are easily interchanged, leading to the inverse filter equation

$$\begin{bmatrix} u(k) \\ y_e(k) \\ \bar{x}_1(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & G_{11} \end{bmatrix} \begin{bmatrix} H_{12} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} y(k) \\ \alpha_{12} \\ \alpha_{11} \end{bmatrix} \tag{22}$$

where

$$H_{12}=\begin{bmatrix} 1/c_{12} & -s_{12}/c_{12} \\ -s_{12}/c_{12} & 1/c_{12} \end{bmatrix},$$

$$H_{12}\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} H'_{12}=\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Consequently, H_{12} is a hyperbolic rotation matrix. The factorized state variable equation for the inverse filter will be just like (16) except $u(\cdot)$and $y(\cdot)$ will interchange and G_{12} will get replaced by H_{12}. Finally, the pipelined inverse filter structure will be as shown in Fig. 2. Note that except for H_{12}, the filter structure of Fig. 2 consists of orthogonal sections; thus we could refer to it as an *essentially orthogonal inverse filter structure.*

An alternate way to see why only one hyperbolic rotation appears in the inverse filter structure of Fig. 2 is to consider its transition matrix. Using (13), the transition matrix for the inverse filter is obtained as

$$F_I=\begin{bmatrix} d^{-1} & -d^{-1}c_f & -d^{-1}\delta_2 \\ \delta_1 d^{-1} & \gamma_f-\delta_1 d^{-1}c_f & \delta_3-\delta_1 d^{-1}\delta_2 \\ b_f d^{-1} & A_f-b_f d^{-1}c_f & \beta_f-b_f d^{-1}\delta_2 \end{bmatrix}$$

where the parameters with the subscript f denote the parameters obtained after applying the transformation Q (see (12)). Now, in contrast to the orthogonality of F, for the inverse filter we have

$$F_I \begin{bmatrix} -1 & 0 & \mathbf{0} \\ 0 & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_n \end{bmatrix} F_I'=\begin{bmatrix} -1 & 0 & \mathbf{0} \\ 0 & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_n \end{bmatrix} \tag{23}$$

which shows that except for one -1 in (23) F_I would be an orthogonal matrix. This -1 is reflected as the only hyperbolic rotation H_{12} in Fig. 2.

We would like to mention that one could obtain a completely orthogonal inverse filter structure by first inverting (1) and then using its parameters in Algorithm 3.1. This will lead to a completely different state-space coordinate and correspondingly a different set of parameters for the various rotors. Our objective was to obtain a pipelined inverse filter structure using the parameters of the filter of Fig. 1 and maintaining the same state-space coordinates.

V. Conclusion

We believe that Algorithms 3.1 and 3.2 will be numerically very reliable because they involve applying orthogonal transformations. Nevertheless, in spite of extensive literature on algorithms for solving algebraic Riccati equations and full rank factorization of symmetric nonnegative definite matrices, numerical robustness of the steps involved in the embedding need a detailed investigation. Also, another topic for future investigation could be the significance of the excess parameters required in the MIMO case to achieve orthogonal embedding. With a new approach there is always the possibility of obtaining a new filter structure. At present we have obtained the structure reported in [4], but we are hopeful that some new structures will emerge. This aspect is currently being investigated. Also, as a byproduct of the state-space approach, in the SISO case, an essentially orthogonal pipelined structure for the inverse filter is obtained.

Appendix A

We present the proof of Lemma 3.1 by giving an algorithm for constructing the desired Q. For computation of the Householder matrices required below, see [16].

Algorithm A.1

1) Compute a Householder matrix P_0 such that

$$P_0'b=[\,*\quad 0\quad \cdots\quad 0\,],\quad \text{and let}\quad A_1=P_0'AP_0.$$

2) For $k=1$: $n-2$ Do (a) and (b):
   a) Compute the Householder matrix $\bar{P}_k$ such that

$$\bar{P}_k\begin{bmatrix} a^k_{k+1,k} \\ \vdots \\ a^k_{n,k} \end{bmatrix}=\begin{bmatrix} * \\ 0 \\ \vdots \\ 0 \end{bmatrix},\quad \text{and let}\quad P_k=\begin{bmatrix} I_k & \mathbf{0} \\ \mathbf{0} & \bar{P}_k \end{bmatrix}$$

   where $a^k_{i,j}$ is the (i,j)th entry of A_k.
   b) Construct $A_{k+1}=P_k'A_kP_k$.

3) Let $Q\overset{\text{def}}{=}P_0P_1\cdots P_{n-2}$, then $A_{n-1}=Q'AQ\overset{\text{def}}{=}H$ will be an upper Hessenberg matrix. Moreover,

$$Q'b=[\,*\quad 0\quad \cdots\quad 0\,].$$

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY BOMBAY. Downloaded on November 3, 2008 at 23:27 from IEEE Xplore. Restrictions apply.

Algorithm A.2 — Algorithm for Factoring an Orthogonal Lower Hessenberg Matrix as a Product of Givens Rotations

Let M be a $q \times q$ orthogonal lower Hessenberg matrix:

$$M = \begin{bmatrix} m_{1,1} & m_{1,2} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ m_{q-2,1} & m_{q-2,2} & m_{q-2,3} & \cdots & 0 & 0 \\ m_{q-1,1} & m_{2,q-1} & m_{3,q-1} & \cdots & m_{q-1,q-1} & m_{q-1,q} \\ m_{q,1} & m_{q,2} & m_{q,3} & \cdots & m_{q,q-1} & m_{q,q} \end{bmatrix} \overset{\text{def}}{=} M_1.$$

For $i = 1$: $q - 2$ Do

$$\begin{bmatrix} i_{q-i-1} & \mathbf{0} \\ \mathbf{0} & G_i' \end{bmatrix} M_i = \begin{bmatrix} M_{i+1} & 0 \\ \mathbf{0} & 1 \end{bmatrix}$$

where

$$G_i = \begin{bmatrix} m^i_{q-i+1,q-i+1} & m^i_{q-i,q-i+1} \\ -m^i_{q-i,q-i+1} & m^i_{q-i+1,q-i+1} \end{bmatrix}$$

and $m^i_{j,k}$ is the $(j-i+1, k-i+1)$th entry of M_i.

Define

$$M_{q-1} \overset{\text{def}}{=} G_{q-1}.$$

Then

$$M = \begin{bmatrix} I_{q-2} & \mathbf{0} \\ \mathbf{0} & G_1 \end{bmatrix} \begin{bmatrix} I_{q-3} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & G_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 1 \end{bmatrix}$$

$$\cdots \begin{bmatrix} 1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & G_{q-2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{q-3} \end{bmatrix} \begin{bmatrix} G_{q-1} & \mathbf{0} \\ \mathbf{0} & I_{q-2} \end{bmatrix}.$$

References

[1] E. Deprettere and P. Dewilde, "Orthogonal cascade realization of real multiport digital filters," *Int. J. Circuit Theory and Applications*, vol. 8, pp. 245–272, 1980.

[2] P. Dewilde, E. Deprettere, and R. Nouta, "Parallel and pipelined VLSI implementation of signal processing algorithms," in *VLSI and Modern Signal Processing*, S. Y. Kung, H. J. Whitehouse, and T. Kailath, Eds. Englewood Cliffs, NJ: Prentice Hall, 1984.

[3] E. Deprettere, P. Dewilde, and P. Rao, "Orthogonal filter design and VLSI implementation," in *Proc. Int. Conf. Computers, Systems, and Signal Processing*, Bangalore, India, pp. 779–790, 1984.

[4] S. K. Rao and T. Kailath, "Orthogonal digital filters for VLSI implementation," *IEEE Trans. Circuits Syst.*, vol. CAS-31, pp. 933–945, Nov. 1984.

[5] ——, "VLSI arrays for digital signal processing: Part I—A model identification approach to digital filter realizations," *IEEE Trans. Circuits Syst.*, vol. CAS-32, pp. 1105–1118, Nov. 1985.

[6] D. Henrot and C. T. Mullis, "A modular orthogonal digital filter structure for parallel processing," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, pp. 623–626, Boston, MA, 1983.

[7] P. P. Vaidyanathan, "An unified approach to orthogonal digital filters and wave digital filters, based on LBR two-pair extraction," *IEEE Trans. Circuits Syst.*, vol. CAS-32, pp. 673–686, July 1985.

[8] C. V. K. Prabhakar Rao, "A systems approach to the multiport lossless inverse scattering problem," Ph.D. dissertation, Dep. Electrical Engineering, Delft University of Technology, Delft, The Netherlands, 1985.

[9] C. V. K. Prabhakar Rao and P. Dewilde, "System theory for lossless wave scattering," in *Operator Theory: Advances and Applications, Vol. 19*. Basel, Switzerland: Birkhauser Verlag, 1986, pp. 333–358.

[10] A. H. Gray and J. D. Markel, "A normalized digital filter structure," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-23, pp. 486–494, Oct. 1975.

[11]. P. A. Regalia, S. K. Mitra, and P. P. Vaidyanathan, "The digital all-pass filter: A versatile signal processing block," *Proc. IEEE*, vol. 76, pp. 19–37, Jan. 1988.

[12] P. P. Vaidyanathan, "Passive cascaded-lattice structures for low-sensitivity FIR filter design, with application to filter banks," *IEEE Trans. Circuits Syst.*, vol. CAS-33, pp. 1045–1064, Nov. 1986.

[13] R. A. Roberts and C. T. Mullis, *Digital Signal Processing*. Reading, MA: Addison Wesley, 1987.

[14] Z. Doganata, P. P. Vaidyanathan, and T. Q. Nguyen, "General synthesis procedures for FIR lossless transfer matrices, for perfect reconstruction filter bank applications," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-36, Oct. 1988.

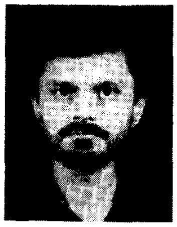
[15] P. P. Vaidyanathan, "The discrete-time bounded-real lemma in digital filtering," *IEEE Trans. Circuits Syst.*, vol. CAS-32, pp. 918–924, Sept. 1985.

[16] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD: The Johns Hopkins Univ. Press, 1983.

[17] F. D. Murnaghan, *The Unitary and Rotation Group*. Washington, DC: Spartan Books, 1962.

[18] B. D. O. Anderson and S. Vongpanitlerd, *Network Analysis and Synthesis: A Modern Systems Theory Approach*. Englewood Cliffs, NJ: Prentice Hall, 1973.

[19] M. Jamshidi, "An overview of the solution of the algebraic Riccati equation and related problems," *J. Large Scale Syst.*, vol. 1, pp. 167–192, 1980.



Uday B. Desai (S'75–M'79) received the B. Tech. degree from Indian Institute of Technology, Kanpur, India, in 1974, the M.S. degree from the State University of New York, Buffalo, in 1976, and the Ph.D. degree from The Johns Hopkins University, Baltimore, MD, in 1979, all in electrical engineering.

From 1979 to 1984 he was an Assistant Professor in the Electrical Engineering Department at Washington State University, Pullman, WA, and an Associate Professor at the same place from 1984 to 1987. He is presently a Professor in the Electrical Engineering Department at the Indian Institute of Technology, Bombay. His research interests are in the areas of statistical and adaptive signal processing, signal processing architectures, artificial neural networks, stochastic modeling, and optimal approximation theory. He is the editor of *Modeling and Applications of Stochastic Processes* (Kluwer Academic Press, Boston, MA, 1986).

Dr. Desai is a member of Eta Kappa Nu and Sigma Xi.

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY BOMBAY. Downloaded on November 3, 2008 at 23:27 from IEEE Xplore. Restrictions apply.