

Design of Multiplier-free State-Space Digital Filters

Tamal Bose, Zhurun Zhang, Miloje Radenkovic, and Ojas Chauhan

Abstract—In this paper, a novel approach is presented for designing multiplier-free state-space digital filters. The multiplier-free design is obtained by finding power-of-2 coefficients and also quantizing the state variables to power-of-2 numbers. Expressions for the noise variance are derived for the quantized state vector and the output of the filter. A “structure-transformation matrix” is incorporated in these expressions. It is shown that quantization effects can be minimized by properly designing the structure-transformation matrix. Simulation results are very promising and illustrate the design algorithm.

Keywords—Digital filters, minimum noise, multiplier-free, quantization, state-space.

I. INTRODUCTION

Multiplier-free filters have received widespread attention in the signal processing community due to their high computational speed and low implementation costs. Different methods have been proposed for the design of FIR multiplier-free filters [1–5] but the development for IIR multiplier-free filters has been limited. It has been shown [6], [7] that a multiplier-free IIR filter can be obtained by deriving a special class of digital filter transfer functions from the elliptic minimal Q-factors. In [8] multiplierless IIR filter has been obtained by appropriate extension of the approach given in [7]. In [9] and [10], the authors have used periodically shift-varying (PSV) filters for designing multiplierless IIR filters in the state-space (SS) domain.

In this paper, we provide a novel approach for designing multiplier-free IIR SS filters. A SS structure with power-of-2 coefficients and quantized state variables is employed to design the proposed multiplier-free filter. This makes the filter fast and low cost in VLSI implementation. Expressions for the noise variance are formulated for the quantized state vector and the output of the filter. A structure transformation matrix is incorporated in these expressions. It is shown in the paper that the power-of-

2 effects can be minimized by properly designing the structure transformation matrix.

The remainder of the paper is organized as follows. Section II introduces the notation and describes the filter design problem. Noise analysis is done in Section III. The procedure to obtain the structure transformation matrix is presented in Section IV. Simulation results are given in Section V and Section VI is the conclusion..

II. PROBLEM DEFINITION

A. Multiplier-Free SS Structure

An IIR digital filter can be described by the SS equations

$$\begin{aligned}\lambda(n+1) &= A\lambda(n) + Bx(n) \\ y(n) &= C\lambda(n) + Dx(n),\end{aligned}\quad (1)$$

where $x(n)$ is the input, $y(n)$ is the output, $\lambda(n)$ is an M dimensional state vector, and A, B, C , and D are, respectively, $M \times M$, $M \times 1$, $1 \times M$, and 1×1 constant matrices. In fact, there is an infinite number of equivalent realizations obtained by using a similarity transformation matrix T as

$$\{A', B', C', D'\} = \{T^{-1}AT, T^{-1}B, CT, D\}.\quad (2)$$

In this paper, we assume that D is power-of-2. This is easily done by scaling the output in (1), i.e., $\frac{1}{\alpha}y(n) = \frac{C}{\alpha}\lambda(n) + \frac{D}{\alpha}x(n)$, where α is a scaling constant.

Based on the transformed SS structure $\{A', B', C', D'\}$, the multiplier-free implementation is given by

$$\begin{aligned}\lambda(n+1) &= A'Q_2[\lambda(n)] + B'x(n) \\ y_q(n) &= C'Q_2[\lambda(n)] + D'x(n),\end{aligned}\quad (3)$$

where $Q_2[\cdot]$ represents quantization to the closest power-of-2 and $D' = D$ is power-of-2. By properly designing T , $B' = T^{-1}B$ can also be made a power-of-2 matrix. This design procedure is described in the following sections. Each multiplication in (3) now has a power-of-2 operand, requiring only shifters in their hardware realization.

Tamal Bose is with the Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24060; Zhurun Zhang is with Microsoft Research, Redmond, WA; Miloje Radenkovic is with the Department of Electrical Engineering, University of Colorado Denver, CO 80217; Ojas Chauhan's affiliation is unknown.

This work was supported in part by NASA Grants NAG5-10716 and NNG06GE95G.

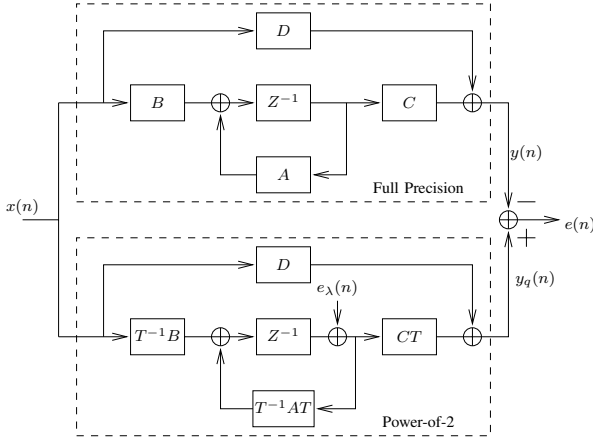


Fig. 1. Structure and design

B. Quantization Noise

Quantization effect is modeled as a noise sequence added to the original sequence, giving the actual finite word-length realization of the multiplier-free filter in (3) as

$$\begin{aligned}\lambda(n+1) &= A'(\lambda(n) + e_\lambda(n)) + B'x(n) + \alpha(n) + \beta(n) \\ y_q(n) &= C'(\lambda(n) + e_\lambda(n)) + D'x(n) + \gamma(n) + \delta(n),\end{aligned}\quad (4)$$

where $\lambda(n)$ is the actual state, $y_q(n)$ is the actual output, and $\alpha(n), \beta(n), \gamma(n)$, and $\delta(n)$ are, respectively, the error vectors caused by the shifters in $A'(\lambda(n) + e_\lambda(n))$, $B'x(n)$, $C'(\lambda(n) + e_\lambda(n))$, and $D'x(n)$. We call $e_\lambda(n)$, the “power-of-2 state noise.”

$e_\lambda(n)$ is typically much greater in magnitude compared to $\alpha(n), \beta(n), \gamma(n)$, and $\delta(n)$. The noise model for (3) can then be approximated as

$$\begin{aligned}\lambda(n+1) &= A'(\lambda(n) + e_\lambda(n)) + B'x(n) \\ y_q(n) &= C'(\lambda(n) + e_\lambda(n)) + D'x(n).\end{aligned}\quad (5)$$

The architecture of the full precision SS filter in (1) and the multiplier free version in (5) are both illustrated in Fig. 1.

C. The Filter Design Problem

The problem of designing the multiplier-free SS filter (5) can be formulated as follows: Given an arbitrary SS filter $\{A, B, C, D\}$ in (1), design a nonsingular matrix T such that the filter $\{A', B', C', D'\} = \{T^{-1}AT, T^{-1}B, CT, D\}$ in (5) satisfies the following requirements.

- 1) The desired output $y_q(n)$ of (5) in response to the same white noise input $x(n)$ is as close as possible, in some sense, to the output $y(n)$ of (1).

- 2) $B' = T^{-1}B$ is a power-of-2 matrix.

The mean square error $E[e^2(n)] = E[|y_q(n) - y(n)|^2]$ is used as a cost function. The output $y_q(n)$ in (5) can be separated into two components. One is the response to input $x(n)$ and other is the response to $e_\lambda(n)$. Since $\{A, B, C, D\}$ and $\{T^{-1}AT, T^{-1}B, CT, D\}$ represent the same filter, the output noise $e(n) = y_q(n) - y(n)$ is the response to $e_\lambda(n)$ only. Hence the first requirement mentioned above can be equivalently stated as designing T to minimize $E[e^2(n)]$.

III. NOISE ANALYSIS

In this section, we formulate the expressions for the variance of the power-of-2 state noise $e_\lambda(n)$ and the output noise $e(n)$.

A. Power-of-2 Quantization Noise

For power-of-2 quantization, we have $Q_2[x(n)] = x(n) + \Delta x(n)$, where $x(n)$ is the input, $Q_2[x(n)]$ is the quantized output, and $\Delta x(n)$ is the quantization noise. If we assume that $x(n)$ is zero mean white noise that is properly scaled between the upper and lower boundary, then $\Delta x(n)$ is also zero mean and has a variance proportional to the variance of $x(n)$. This relation can be formulated as

$$E[\Delta x^2(n)] \approx \kappa E[x^2(n)], \quad (6)$$

where $\kappa = 0.038$ is a constant obtained through simulations. It will be seen that the optimal matrix T is independent of κ , and therefore exact value of κ is unimportant.

B. Power-of-2 State Noise

From (5) we have $\lambda(n) = (\lambda(n) + e_\lambda(n)) + (-e_\lambda(n)) = Q_2[\lambda(n)] + \Delta\lambda(n)$, where $\lambda(n) = [\lambda_1(n), \dots, \lambda_M(n)]^T$ and $e_\lambda(n) = -\Delta\lambda(n)$. The covariance matrix of $e_\lambda(n)$ is $E[e_\lambda(n)e_\lambda^T(n)] = E[\Delta\lambda(n)\Delta\lambda^T(n)]$. Assuming no cross correlation and using (6), we get,

$$\begin{aligned}& E[e_\lambda(n)e_\lambda^T(n)] \\ &= \begin{bmatrix} E[e_{\lambda_1}(n)e_{\lambda_1}^*(n)] & \cdots & E[e_{\lambda_1}(n)e_{\lambda_M}^*(n)] \\ \vdots & \ddots & \vdots \\ E[e_{\lambda_M}(n)e_{\lambda_1}^*(n)] & \cdots & E[e_{\lambda_M}(n)e_{\lambda_M}^*(n)] \end{bmatrix} \\ &= \kappa \begin{bmatrix} E[\lambda_1(n)\lambda_1^*(n)] & & 0 \\ & \ddots & \\ 0 & & E[\lambda_M(n)\lambda_M^*(n)] \end{bmatrix} \\ &= \kappa \text{diag}(E[\lambda(n)\lambda^T(n)]).\end{aligned}$$

Defining $\lambda(z) = [\lambda_1(z), \dots, \lambda_M(z)]^T$, and considering (5) without $e_\lambda(n)$, the input state relation of the

filter is described by $z\lambda(z) = A'\lambda(z) + B'X(z)$ and $T^{-1}AT\lambda(z) + T^{-1}BX(z)$. Hence,

$$\lambda(z) = T^{-1}(zI - A)^{-1}BX(z). \quad (7)$$

Using (7) and Parseval's theorem, the covariance matrix of the state vector can be expressed as

$$\begin{aligned} & E[\lambda(n)\lambda^T(n)] \\ &= E[[\lambda_1(n), \dots, \lambda_M(n)]^T [\lambda_1(n), \dots, \lambda_M(n)]] \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \lambda(e^{j\omega}) \lambda^T(e^{-j\omega}) d\omega \\ &= \sigma_x^2 \frac{1}{2\pi} \int_{-\pi}^{\pi} [(T^{-1}(e^{j\omega}I - A)^{-1}B) \\ &\quad \cdot (T^{-1}(e^{j\omega}I - A)^{-1}B)^*] d\omega \\ &= \sigma_x^2 \cdot [T^{-1}GT^{-T}], \end{aligned} \quad (8)$$

where $G = \frac{1}{2\pi} \int_{-\pi}^{\pi} [(e^{j\omega}I - A)^{-1}B][(e^{j\omega}I - A)^{-1}B]^* d\omega$ is a Hermitian matrix and σ_x^2 is the variance of the white noise input. From (7) and (8), we have

$$E[e_\lambda(n)e_\lambda^T(n)] = \kappa \cdot \sigma_x^2 \cdot \text{diag}(T^{-1}GT^{-T}), \quad (9)$$

where G and $T^{-1}GT^{-T}$ can be interpreted as the covariance matrices of power-of-2 state noise for the original and transformed filters, respectively.

C. Output Noise

Output noise $e(n)$ is the response of the multiplier-free filter to input $e_\lambda(n)$, which is a white noise vector with covariance matrix given by (9). If we define $e_\lambda(z) = [e_{\lambda_1}(z), \dots, e_{\lambda_M}(z)]^T$, the state-output relation of the filter in (5) can be described by the equations

$$\begin{aligned} z\lambda(z) &= A'(\lambda(z) + e_\lambda(z)) = T^{-1}AT(\lambda(z) + e_\lambda(z)) \\ e(z) &= C'(\lambda(z) + e_\lambda(z)) = CT(\lambda(z) + e_\lambda(z)). \end{aligned} \quad (10)$$

After some algebraic manipulations $e(z)$ in (10) can be expressed as

$$\begin{aligned} e(z) &= CT[(zI - T^{-1}AT)^{-1}(T^{-1}AT) + I] \cdot e_\lambda(z) \\ &= C((zI - A)^{-1}A + I) \cdot T \cdot e_\lambda(z) \\ &= C((zI - A)^{-1}(A - zI + zI) + I) \cdot T \cdot e_\lambda(z) \\ &= C(zI - A)^{-1}zI \cdot T \cdot e_\lambda(z). \end{aligned} \quad (11)$$

Define

$$\begin{aligned} H(e^{j\omega}) &= C(zI - A)^{-1}zI \cdot T|_{z=e^{j\omega}} \\ &= [h_1(e^{j\omega}), \dots, h_M(e^{j\omega})]. \end{aligned} \quad (12)$$

Using (11) and (12) we get

$$e(n) = h_1(n) * e_{\lambda_1}(n) + \dots + h_M(n) * e_{\lambda_M}(n),$$

$$\begin{aligned} & E[e^2(n)] \\ &= E[(h_1(n) * e_{\lambda_1}(n) + \dots + h_M(n) * e_{\lambda_M}(n))^2] \\ &= \sum_{i=1}^M \sum_{j=1}^M E[(h_i(n) * e_{\lambda_i}(n)) \cdot (h_j(n) * e_{\lambda_j}(n))^*] \\ &= \sum_{i=1}^M \sum_{j=1}^M E[e_{\lambda_i}(n) * e_{\lambda_j}^*(n)] * h_i(n) * h_j(-n) \end{aligned} \quad (13)$$

Using Parseval's theorem on (13) and using (12), we have where $W = \frac{1}{2\pi} \int_{-\pi}^{\pi} [C(e^{j\omega}I - A)^{-1}e^{j\omega}I][C(e^{j\omega}I - A)^{-1}e^{j\omega}I]^T d\omega$ is a Hermitian matrix. Substituting (9) into (14), we finally get

$$\begin{aligned} & E[e^2(n)] \\ &= \text{tr} \{ \kappa \cdot \sigma_x^2 \cdot \text{diag}(T^{-1}GT^{-T}) \cdot T^TWT \} \\ &= \kappa \cdot \sigma_x^2 \cdot \text{tr} \{ \text{diag}(T^{-1}GT^{-T}) \cdot \text{diag}(T^TWT) \} \end{aligned} \quad (15)$$

IV. DESIGNING TRANSFORMATION MATRIX T

We now design the real non-singular matrix T to minimize the cost function given in (15), and achieve power-of-2 $B' = T^{-1}B$. However, a closed form solution of T , for this problem, is very difficult to establish. We provide an iterative algorithm to find T in a scaled orthogonal subspace. That is, we design T which has the form

$$T = P \cdot S = P \cdot \begin{bmatrix} s_{11} & & 0 \\ & \ddots & \\ 0 & & s_{MM} \end{bmatrix}, \quad (16)$$

where P is an orthogonal matrix ($P^T P = I$) and S is a diagonal scaling matrix with nonzero diagonal elements ($s_{ii} \neq 0$, $i = 1, \dots, M$). The design procedure is divided into two parts: 1) Designing the orthogonal matrix P , and 2) Determining the scaling matrix S . The design methodology will utilize the following theorem.

Theorem 1: If $f(T) = \text{tr} \{ \text{diag}(T^{-1}GT^{-T}) \cdot \text{diag}(T^TWT) \}$, where G and W are Hermitian matrices and T has the form given in (16), then $f(T) = f(P)$.

Proof: Using $P^T P = I$, and (16) we have

$$T^{-1} = [P \cdot S]^{-1} = \begin{bmatrix} \frac{1}{s_{11}} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{s_{MM}} \end{bmatrix} \cdot P^T.$$

Assume $P = [p_1, \dots, p_M]$ where p_i , ($i = 1, \dots, M$) are column vectors. Then Similarly,

$$\text{diag}(T^TWT) = \begin{bmatrix} s_{11}^2(p_1^T W p_1) & & 0 \\ & \ddots & \\ 0 & & s_{MM}^2(p_M^T W p_M) \end{bmatrix}.$$

$$\begin{aligned}
& E[e^2(n)] \\
&= \sum_{i=1}^M \sum_{j=1}^M \left\{ E[e_{\lambda_i}(n) * e_{\lambda_j}^*(n)] \cdot \frac{1}{2\pi} \int_{-\pi}^{\pi} [h_i(e^{j\omega}) h_j^*(e^{j\omega})] d\omega \right\} \\
&= \text{tr} \left\{ E[e_{\lambda}(n) \cdot e_{\lambda}^T(n)] \cdot \frac{1}{2\pi} \int_{-\pi}^{\pi} [H^T(e^{j\omega}) H(e^{j\omega})] d\omega \right\} \\
&= \text{tr} \left\{ E[e_{\lambda}(n) \cdot e_{\lambda}^T(n)] \cdot T^T \left[\frac{1}{2\pi} \int_{-\pi}^{\pi} [C(e^{j\omega} I - A)^{-1} e^{j\omega} I] \cdot [C(e^{j\omega} I - A)^{-1} e^{j\omega} I]^T d\omega \right] T \right\} \\
&= \text{tr} \{ E[e_{\lambda}(n) \cdot e_{\lambda}^T(n)] \cdot [T^T W T] \}, \tag{14}
\end{aligned}$$

$$\begin{aligned}
& \text{diag}(T^{-1} G T^{-T}) \\
&= \text{diag} \left(\begin{bmatrix} \frac{1}{s_{11}} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{s_{MM}} \end{bmatrix} P^T G P \begin{bmatrix} \frac{1}{s_{11}} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{s_{MM}} \end{bmatrix} \right) \\
&= \begin{bmatrix} \left(\frac{1}{s_{11}}\right)^2 (p_1^T G p_1) & & 0 \\ & \ddots & \\ 0 & & \left(\frac{1}{s_{MM}}\right)^2 (p_M^T G p_M) \end{bmatrix}. \tag{17}
\end{aligned}$$

Using (17) and (18),

$$\begin{aligned}
f(T) &= \text{tr} \{ \text{diag}(T^{-1} G T^{-T}) \cdot \text{diag}(T^T W T) \} \\
&= \sum_{i=1}^M \left(\frac{1}{s_{ii}} \right)^2 (p_i^T G p_i) s_{ii}^2 (p_i^T W p_i) \\
&= \text{tr} \{ \text{diag}(P^T G P) \cdot \text{diag}(P^T W P) \} \\
&= \text{tr} \{ \text{diag}(P^{-1} G P^{-T}) \cdot \text{diag}(P^T W P) \} \\
&= f(P). \quad \square
\end{aligned}$$

A. Designing the Scaling Matrix S

Theorem 1 claims that once the orthogonal matrix P is fixed, the cost function $f(P S)$ is unaffected by the scaling matrix S . The degree of freedom in choosing S provides the opportunity to achieve power-of-2 $B' = T^{-1} B$. Assuming $P = [p_1, \dots, p_M]$ is given, then from (16) we have

$$B' = \begin{bmatrix} b'_1 \\ \vdots \\ b'_M \end{bmatrix} = \begin{bmatrix} \frac{1}{s_{11}} p_1^T B \\ \vdots \\ \frac{1}{s_{MM}} p_M^T B \end{bmatrix},$$

where B' is the desired power-of-2 matrix (b'_1, \dots, b'_M can be chosen arbitrarily), and

$$s_{ii} = \frac{p_i^T B}{b'_i} \quad (i = 1, \dots, M). \tag{18}$$

Therefore, once the orthogonal matrix P is fixed, and the power-of-2 matrix B' is chosen, the scaling matrix S given by

$$S = \begin{bmatrix} \frac{p_1^T B}{b'_1} & & 0 \\ & \ddots & \\ 0 & & \frac{p_M^T B}{b'_M} \end{bmatrix}$$

can be used without affecting the cost function.

B. Designing the Orthogonal Matrix P

The problem of finding the orthogonal matrix P can be formulated as follows.

$$\min_P f(P) = \text{tr} \{ \text{diag}(P^{-1} G P^{-T}) \cdot \text{diag}(P^T W P) \}$$

where subject to $P^T \cdot P = I$, G and W are given Hermitian matrices.

An iterative algorithm is derived to solve this problem. At each iteration, the algorithm uses the steepest gradient update procedure and an orthogonalization procedure in order to satisfy the orthogonality constraint.

In the steepest gradient update procedure, we first assume $P = [p_1, \dots, p_M]$ is orthogonal and then compute the gradient of $f(P)$ with respect to P as

$$\nabla f(P) = \left[\frac{\partial f(P)}{\partial p_1}, \frac{\partial f(P)}{\partial p_2}, \dots, \frac{\partial f(P)}{\partial p_M} \right],$$

where

$$\begin{aligned}\frac{\partial f(P)}{\partial p_i} &= \frac{\partial}{\partial p_i} \text{tr} \{ \text{diag}(P^{-1}GP^{-T}) \cdot \text{diag}(P^TWP) \} \\ &= \frac{\partial}{\partial p_i} \text{tr} \{ \text{diag}(P^TGP) \cdot \text{diag}(P^TWP) \} \\ &= (p_i^T G p_i) W p_i + (p_i^T W p_i) G p_i \\ &\quad (i = 1, \dots, M).\end{aligned}\quad (19)$$

Then an update of P in the steepest-descent direction is performed as

$$P^{(k_1+1)} = P^{(k_1)} - \mu_1 \nabla f(P^{(k_1)}), \quad (20)$$

where μ_1 is the iteration step and k_1 is the iteration index. Equations (19) and (20) are repeated several times to reduce the cost function $f(P)$.

Now we satisfy the orthogonality constraint, $P^T P = I$. We use the following cost function

$$\begin{aligned}R(P) &= \|P^T P - I\|_F^2 \\ &= \text{tr} \{ (P^T P - I)^T (P^T P - I) \} \\ &= \sum_{i=1}^M \sum_{j=1, j \neq i}^M [(p_i^T p_j)^2 + (p_i^T p_i - 1)^2],\end{aligned}\quad (21)$$

where $\|\cdot\|_F$ is the Frobenius norm. P is orthogonalized when $R(P)$ is reduced to zero. A steepest descent method shown in the following equations is used to minimize (21).

$$\begin{aligned}\nabla R(P) &= \left[\frac{\partial R(P)}{\partial p_1}, \frac{\partial R(P)}{\partial p_2}, \dots, \frac{\partial R(P)}{\partial p_M} \right] \\ \frac{\partial R(P)}{\partial p_i} &= 4 \sum_{j \neq i} (p_i^T p_j) p_j + 2(p_i^T p_i - 1) p_i \\ P^{(k_2+1)} &= P^{(k_2)} - \mu_2 \nabla R(P^{(k_2)}),\end{aligned}\quad (22)$$

where μ_2 is the iteration step and k_2 is the iteration index. The algorithm to find the orthogonal matrix P is summarized in Table I.

The complete procedure for designing the multiplier-free SS filter is summarized in Table II.

V. SIMULATION RESULTS

The simulation procedure is illustrated in Fig. 3 with three different filters. The full precision SS filter described by $\{A, B, C, D\}$ is the desired filter. The transformed multiplier-free filter is the filter we designed. The semi multiplier-free SS filter has the same power-of-2 quantization structure as the transformed multiplier-free SS filter, except that it does not use the structure transformation matrix T . Since B and D are not power-of-2 matrices, the shifters can only be used in $AQ_2[\lambda(n)]$ and $CQ_2[\lambda(n)]$. Hence, the name semi multiplier-free. In

TABLE I
FINDING ORTHOGONAL MATRIX P TO MINIMIZE $f(P)$

Input: Hermitian symmetric matrices G and W , No. of steepest descent iterations ($N^\#$), μ_1 and μ_2 step size of steepest-descent itr., ε minimum cost for $R(P)$.

Initialization: Randomly initialize P .

Procedure:

(1) Steepest-descent update of P .

for $k_1 = 1:N^\#$

$$\nabla f(P^{(k_1)}) = \left[\frac{\partial f(P^{(k_1)})}{\partial p_1}, \frac{\partial f(P^{(k_1)})}{\partial p_2}, \dots, \frac{\partial f(P^{(k_1)})}{\partial p_M} \right],$$

$$\text{where } \frac{\partial f(P^{(k_1)})}{\partial p_i} = (p_i^T G p_i) W p_i + (p_i^T W p_i) G p_i \quad (i = 1, \dots, M).$$

$$P^{(k_1+1)} = P^{(k_1)} - \mu_1 \nabla f(P^{(k_1)}).$$

end

(2) Orthogonalize P .

while $R(P) > \varepsilon$

$$\nabla R(P^{(k_2)}) = \left[\frac{\partial R(P^{(k_2)})}{\partial p_1}, \frac{\partial R(P^{(k_2)})}{\partial p_2}, \dots, \frac{\partial R(P^{(k_2)})}{\partial p_M} \right],$$

$$\text{where } \frac{\partial R(P)}{\partial p_i} = 4 \sum_{j \neq i} (p_i^T p_j) p_j + 2(p_i^T p_i - 1) p_i.$$

$$P^{(k_2+1)} = P^{(k_2)} - \mu_2 \nabla R(P^{(k_2)}).$$

end

TABLE II
MULTIPLIER-FREE SS FILTER DESIGN PROCEDURE

Input: Arbitrary SS filter $\{A, B, C, D\}$

Procedure:

(1) Scale the output to achieve power-of-2 D .

$$\frac{1}{\alpha} y(n) = \frac{C}{\alpha} \lambda(n) + \frac{D}{\alpha} x(n)$$

(2) Calculate the Hermitian symmetric matrices G and W .

$$\begin{aligned}G &= \frac{1}{2\pi} \int_{-\pi}^{\pi} [(e^{j\omega} I - A)^{-1} B] [(e^{j\omega} I - A)^{-1} B]^* d\omega \\ W &= \frac{1}{2\pi} \int_{-\pi}^{\pi} [C(e^{j\omega} I - A)^{-1} e^{j\omega} I] \\ &\quad \cdot [C(e^{j\omega} I - A)^{-1} e^{j\omega} I]^T d\omega\end{aligned}$$

(3) Design P using Algorithm 1.

(4) Find the scaling matrix S .

$$B^d = [b_1^d, b_2^d, \dots, b_M^d]^T = Q_2[B]$$

$$s_{ii} = \frac{p_i^T B}{b_i^d} \quad (i = 1, \dots, M)$$

(5) Get $T = PS$.

(6) Return the multiplier-free SS filter with

$$\{A', B', C', D'\} = \{T^{-1}AT, T^{-1}B, CT, D\}$$

fact, the transformed SS multiplier-free filter equals the semi multiplier-free SS filter when $T = I$. Thus, the difference between $E[e_T^2(n)]$ and $E[e_T^2(n)]$ is interpreted as the improvement that we achieve by using the structure transformation matrix T .

Example 1: Consider a 4th order Butterworth low pass filter with passband edge frequency at 0.3π . The SS structure of this filter is

$$A = \begin{bmatrix} -0.0914 & -0.4630 & 0 & 0 \\ 0.4630 & 0.7641 & 0 & 0 \\ 0.1430 & 0.5449 & 0.2124 & -0.6178 \\ 0.0729 & 0.2776 & 0.6178 & 0.6852 \end{bmatrix}, B = \begin{bmatrix} 0.6547 \\ 0.3336 \\ 0.1030 \\ 0.0525 \end{bmatrix},$$

$$C = [0.0258 \quad 0.0982 \quad 0.2184 \quad 0.5958], D = [0.0186].$$

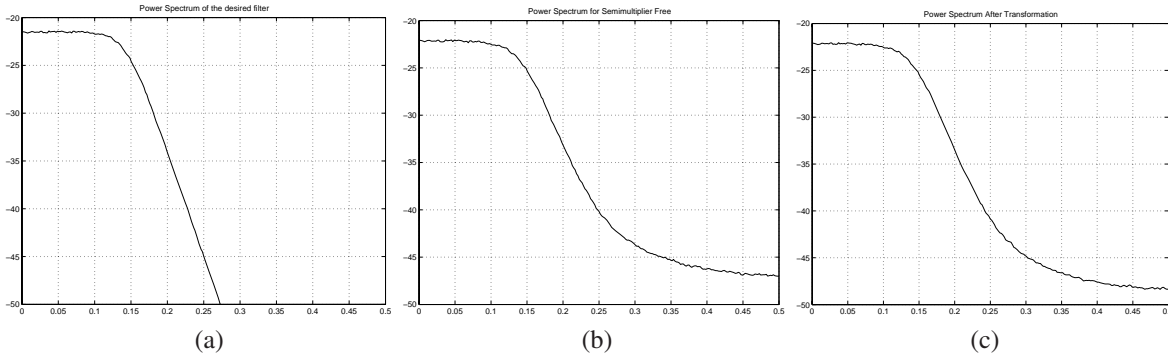


Fig. 2. Spectrum of (a) Desired filter, (b) Semi multiplier-free filter, (c) Transformed multiplier-free filter.

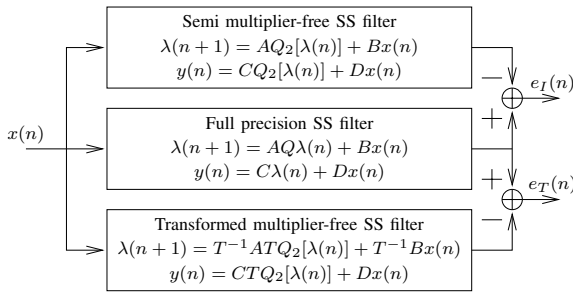


Fig. 3. Simulation Procedure

After scaling the output $y(n)$ with factor $\alpha = \frac{2^{-6}}{0.0186} = 0.8401$ we get

$$C = [0.0217 \quad 0.0826 \quad 0.1838 \quad 0.5015], D = [2^{-6}].$$

Following the procedure in Algorithm 2 we find the transformation matrix

$$T = \begin{bmatrix} 0.8588 & 0.0347 & -0.7364 & -0.1303 \\ 0.4284 & 0.2813 & 0.5574 & -0.7537 \\ -0.2219 & 0.8676 & -0.1880 & 0.1995 \\ 0.6827 & 0.0619 & 0.5154 & 0.7018 \end{bmatrix}.$$

Using (2) we obtain the following transformed SS structure.

$$A' = \begin{bmatrix} 0.3116 & 0.2553 & 0.0967 & 0.3359 \\ 0.1533 & 0.4363 & -0.1201 & -1.0210 \\ 0.7739 & 0.4414 & 0.3399 & -0.2317 \\ -0.1552 & 0.3280 & 0.1487 & 0.4826 \end{bmatrix}, B' = \begin{bmatrix} 2^{-1} \\ 2^{-2} \\ -2^{-2} \\ -2^{-2} \end{bmatrix}$$

$$C' = [0.3556 \quad 0.2145 \quad 0.2540 \quad 0.3235], D' = [2^{-6}]$$

We feed the filter with Gaussian white noise with variance $\sigma_x^2 = 0.001$. Power-of-2 quantization values used are $\{0, \pm 2^0, \pm 2^{-1}, \dots, \pm 2^{-8}\}$. From (15), the variance of the output noise is

$$10 \log_{10}(E[e_T^2(n)]) = -37.5819 \text{ dB}.$$

For the semi multiplier-free SS filter, the variance of output noise is obtained from (15) by setting $T = I$. The result is

$$10 \log_{10}(E[e_I^2(n)]) = -36.3846 \text{ dB}.$$

There is an $37.5819 - 36.3846 = 1.1973 \text{ dB}$ improvement in theory. From simulation we get $E[e_T^2(n)] = -37.7289 \text{ dB}$ and $E[e_I^2(n)] = -36.5949 \text{ dB}$, so the actual improvement is 1.134 dB . The difference between the theoretical and actual value is due to the ignored cross correlation terms in (9). The power spectrum of the filters are shown in Fig. 2.

VI. CONCLUSION

In this paper, a new approach is presented for designing multiplier-free SS digital filters. Expressions for noise variance are formulated for the quantized state vector and the output noise. A steepest-descent type method is developed for the design algorithm. The simulation results are very promising and yield low noise multiplier-free filters.

REFERENCES

- [1] Y. Lim and S. C. Parker, "FIR filter design over a discrete powers-of-two coefficient space," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 31, no. 3, pp. 583–591, June 1983.
- [2] Y. C. Lim and B. Liu, "The design of cascade form FIR filters for discrete space implementation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 7, pp. 1735–1739, Nov. 1988.
- [3] H. Samueli, "An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients," vol. 36, no. 7, pp. 1044–1047, July 1989.
- [4] S. P. Ghanekar, S. Tantarana, and L. E. Franks, "A class of high-precision multiplier-free FIR filter realizations with periodically time-varying coefficients," *IEEE Trans. Signal Processing*, vol. 43, no. 4, pp. 822–830, Apr. 1995.
- [5] —, "Design and architecture of multiplier-free FIR filter using periodically time-varying ternary coefficients," vol. 40, no. 5, pp. 364–370, May 1993.
- [6] L. D. Milic and M. D. Lutovac, "Design of multiplierless elliptic IIR filters with a small quantization error," *IEEE Trans. Signal Processing*, vol. 47, no. 2, pp. 469–479, Feb. 1999.
- [7] M. D. Lutovac and L. D. Milic, "Design of computationally efficient elliptic IIR filters with a reduced number of shift-and-add operations and multipliers," *IEEE Trans. Signal Processing*, vol. 45, pp. 2242–2430, Oct. 1997.

- [8] L. Milic and M. D. Lutovac, "Design of multiplierless IIR filters," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., ICASSP*, Munich, Germany, Apr. 1997, pp. 2201–2204.
- [9] F. L. Kitson and L. Griffiths, "Design and analysis of recursive periodically time-varying digital filters with highly quantized coefficients," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 5, pp. 647–685, May 1988.
- [10] S. P. Ghanekar, S. Tantarana, and L. E. Franks, "Multiplier-free IIR filter realization using periodically time-varying state-space structure—Part I: Structure and design," *IEEE Trans. Signal Processing*, vol. 42, no. 5, pp. 1008–1017, May 1994.