設計自動化 62－8
(1992. 5.27)

ギブンズ回転に基づく巡回格子形多チャンネル適応フィルタ

酒井英昭
京都大学　工学部
京都市左京区吉田本町

あらまし　本論文ではギブンズ回転ユニットを用いた巡回格子形多チャンネル適応フィルタについて述べる。近年、適応信号処理の分野における逐次最小2乗法に対し、ギブンズ回転を用いたQR法に基づくシストリックアレーが注目されてきたが、これは修正グラム・シュミット法に基づく事前誤差を用いたアレーと等価であることが知られている。この等価性を逆に利用して、高速信号処理法として知られる格子形アルゴリズムをギブンズ回転ユニットを用いて実現することが提案されている。

本論文では多チャンネル信号処理のためのスカラー演算のみから成る巡回格子形アルゴリズムにこの手法を適用し、CORDICで実現できる適応フィルタの構造を示す。

The Circular Lattice Filter For Multichannel Signal Processing Using Givens Rotation Units

Hideaki Sakai
Faculty of Engineering, Kyoto University
Yoshida Honmachi Sakyo-ku, 606 Kyoto-shi

Abstract　This paper presents the circular lattice (CL) filter for adaptive multichannel signal processing using Givens rotation units.

Recently, it has been shown that a triangular systolic array by using the QR method based on Givens rotations and the array based on the *a priori* errors by the modified Gram-Schmidt method are in a sense equivalent. Using this equivalence in reverse for the least squares CL algorithm, an adaptive filter based on the Givens rotation is proposed. Most salient features of this filter are that though we treat vector signals, there are no matrix vector operations and it is highly uniform and can be implemented by CORDIC.

1 Introduction and Summary

In this paper we present the circular lattice (CL) filter for adaptive multichannel signal processing using Givens rotation units.

Recently, for the recursive least squares (RLS) problem in adaptive signal processing, a triangular systolic array has been proposed by using the QR method based on Givens rotations. It has been also shown that this array and the array based on the *a pripori* errors by the modified Gram-Schmidt (MGS) method are in a sense equivalent [1]. Using this equivalence in reverse for the faster lattice algorithm, Ling [1] proposed the Givens rotation based lattice algorithm.

In this paper we generalize the result in [1] to the multichannel case where we use the circular lattice (CL) structure [2]. Most salient features of the CL structure are that though we treat vector signals, there are no matrix vector operations and it is highly uniform consisting of two kinds of computational units.

2 Systolic Array Based on QR Method

In this section we briefly review the triangular systolic array for RLS of approximating a desired response $y(n)$ by a linear combination of input signals $x_1(n),\ x_2(n), \cdots, x_d(n)$ based on the Givens rotation method. Denote the $n \times d$ weighted data matrix as

$$\boldsymbol{X}(n) = \Lambda(n)\,[\boldsymbol{x}(1)\ \boldsymbol{x}(2)\ \cdots \boldsymbol{x}(n)]^T = \begin{bmatrix} \lambda^{1/2}\boldsymbol{X}(n-1) \\ \cdots\cdots\cdots\cdots \\ \boldsymbol{x}^T(n) \end{bmatrix} \tag{1}$$

and the $n \times 1$ weighted desired response vector as

$$\boldsymbol{y}(n) = \begin{bmatrix} \lambda^{1/2}\boldsymbol{y}(n-1) \\ \cdots\cdots\cdots\cdots \\ y(n) \end{bmatrix} \tag{2}$$

where $\boldsymbol{x}(n)$ is given by

$$\boldsymbol{x}(n) = (x_1(n)\ x_2(n) \cdots x_d(n))^T \tag{3}$$

and $\Lambda(n)$ is an $n \times n$ "forgetting" matrix of the form

$$\Lambda(n) = \mathrm{diag}\left(\lambda^{\frac{n-1}{2}}, \cdots, \lambda^{\frac{1}{2}},\ 1\right) \tag{4}$$

with $0 < \lambda \leq 1$. Suppose that at the $(n-1)th$ update we have the following QRD

$$\boldsymbol{Q}(n-1)\boldsymbol{X}(n-1) = \begin{bmatrix} \boldsymbol{R}(n-1) \\ 0 \end{bmatrix} \tag{5}$$

where $\boldsymbol{Q}(n-1)$ is an $(n-1)\times(n-1)$ orthogonal matrix and $\boldsymbol{R}(n-1)$ is a $d \times d$ upper triangular matrix. But from (1) and (5) it follows that

$$\begin{pmatrix} \boldsymbol{Q}(n-1) & 0 \\ 0^T & 1 \end{pmatrix} \boldsymbol{X}(n) = \begin{pmatrix} \lambda^{1/2}\boldsymbol{R}(n-1) \\ 0 \\ \boldsymbol{x}^T(n) \end{pmatrix}. \tag{6}$$

To annihilate the last low in the right hand side of (6), a series of the Givens transformations are used . Thus we have

$$\boldsymbol{G}(n) \begin{pmatrix} \lambda^{1/2}\boldsymbol{R}(n-1) \\ 0 \\ \boldsymbol{x}^T(n) \end{pmatrix} = \begin{pmatrix} \boldsymbol{R}(n) \\ 0 \end{pmatrix} \tag{7}$$

and the new $\boldsymbol{Q}(n)$ is

$$\boldsymbol{Q}(n) = \boldsymbol{G}(n)\ \mathrm{diag}(\boldsymbol{Q}(n-1),\ 1). \tag{8}$$

Now write

$$\begin{pmatrix} \boldsymbol{R}(n) & \boldsymbol{u}(n) \\ 0 & \boldsymbol{v}(n) \end{pmatrix} = \boldsymbol{Q}(n)(\boldsymbol{X}(n)\ \boldsymbol{y}(n)) \tag{9}$$

where $\boldsymbol{u}(n)$ is $d \times 1$ and $\boldsymbol{v}(n)$ is $(n-d) \times 1$. The original least squares problem is to find the minimizer of $\| \boldsymbol{y}(n) - \boldsymbol{X}(n)\boldsymbol{c} \|$. From (9) and the orthogonality of $\boldsymbol{Q}(n)$, the minimizer $\boldsymbol{c}(n)$ is given by $\boldsymbol{R}(n)\boldsymbol{c}(n) = \boldsymbol{u}(n)$ and the residual vector $\boldsymbol{\epsilon}(n)$ is given by

$$\boldsymbol{\epsilon}(n) = \boldsymbol{y}(n) - \boldsymbol{X}(n)\boldsymbol{c}(n). \tag{10}$$

The last element of the above vector is called the *a posteriori error* and is given by

$$\epsilon(n) = y(n) - \boldsymbol{x}^T(n)\boldsymbol{c}(n) \tag{11}$$

This can be computed efficiently by a triangular systolic array in Fig. 1 without computing $\boldsymbol{c}(n)$ explicitly. The components of $d \times d$ $\boldsymbol{R}(n)$ are stored and updated in part A of this figure . Also, the components of $d \times 1$ $\boldsymbol{u}(n)$ are stored and updated in the column B in the same fashion as the off-diagonal components of $\boldsymbol{R}(n)$ with the input $y(n)$ from the top of this column and the output $v(n)$ from the bottom of this column. This $v(n)$ is in a sense a geometric mean of the *a posteriori error* $\epsilon(n)$ and the *a priori error* $e(n)$ defined by

$$e(n) = y(n) - \boldsymbol{x}^T(n)\boldsymbol{c}(n-1). \qquad (12)$$

The operation of the boundary cell (angle computer) denoted by a circle in Fig. 2 (a) is as follows.

$$d' = \sqrt{\lambda d^2 + y_{in}^2},\ c = \lambda^{1/2} d/d'$$
$$s = y_{in}/d',\ d \leftarrow d',\ \gamma_{out}^{1/2} = c\gamma_{in}^{1/2}. \qquad (13)$$

The operation of the internal cell (rotor) denoted by a square in Fig. 2 (b) is as follows.

$$x_{out} = cx_{in} - s\lambda^{1/2} r,$$
$$r' = sx_{in} + c\lambda^{1/2} r,\ r \leftarrow r'. \qquad (14)$$

Based on the *a priori error* formulation of the MGS method, Ling [1] has shown that the above cell operations can be changed to be square-root-free. The operation of the boundary cell in Fig. 3 (a) is as follows.

$$\Delta' = \lambda\Delta + \delta\overline{y}_{in}^2,\ \overline{s} = \delta\overline{y}_{in}/\Delta'$$
$$\delta' = \delta - \delta^2\overline{y}_{in}^2/\Delta',\ \Delta \leftarrow \Delta'. \qquad (15)$$

The operation of the internal cell in Fig. 3 (b) is as follows.

$$\overline{x}_{out} = \overline{x}_{in} - \overline{y}_{in}\overline{r},$$
$$\overline{r}' = \overline{r} + \overline{s}\ \overline{x}_{out},\ \overline{r} \leftarrow \overline{r}' \qquad (16)$$

The relations between the quantities in both realizations are $\sqrt{\Delta} = d$, $\sqrt{\Delta}\overline{r} = r$, $\sqrt{\delta}\overline{y}_{in} = y_{in}$, $\sqrt{\delta}\overline{x}_{in} = x_{in}$, $\overline{c} = c^2$, $\delta' = \gamma_{out}$, $\delta = \gamma_{in}$.

3 The Circular Lattice Algorithm

Now we consider the problem of estimating $y(n)$ by a linear combination of $\boldsymbol{x}(n)$, $\boldsymbol{x}(n-1), \cdots, \boldsymbol{x}(n-p)$. This problem arises in, for example, broad band beamforming. For this multichannel FIR filtering problem the least squares circular lattice (LSCL) algorithm has been derived by using a geometric method [2]. Here we review the result and derive its new form using the *a priori* errors.

First we define the ket vector by

$$|z\rangle_N = [z(1)\ z(2) \ldots z(N)]^T \qquad (17)$$

for scalar time series $\{z(t)\}$ of length N and the bra vector by $\langle z|_N = |z\rangle_N^T$. We also define the inner product beween $|z\rangle_N$ and $|w\rangle_N$ in this space by

$$\langle z|w\rangle_N = \sum_{i=1}^{N} z(i)\lambda^{N-i} w(i) \qquad (18)$$

where λ is the forgetting factor and $0 < \lambda \le 1$.

Now we have d channel time series data $\{\boldsymbol{x}(1)\ \boldsymbol{x}(2) \ldots \boldsymbol{x}(N)\}$. Writing the ith element of $\boldsymbol{x}(t)$ as $x_i(t)$, we define the following channel data vector for $i = 1, \ldots, d$ by

$$|x_i\rangle_N = [x_i(1)\ x_i(2) \ldots x_i(N)]^T. \qquad (19)$$

To these vectors, an operator s^{-1} is defined as follows :

$$|s^{-1}x_1\rangle_N = [0\ x_d(1)\ x_d(2) \ldots x_d(N-1)]^T$$
$$|s^{-1}x_i\rangle_N = |x_{i-1} >_N \ (i = 2, \ldots, d).$$

Let $Y_{j,k,N}$ be the subspace spanned by the vectors $|x_j\rangle_N, \ldots, |x_k\rangle_N$ $(j \le k)$ and $P_{j,k,N}$ be the orthogonal projection operator on $Y_{j,k,N}$ defined by

$$P_{j,k,N} = |Y_{j,k}\rangle_N \langle Y_{j,k}|Y_{j,k}\rangle_N^{-1} \langle Y_{j,k}|_N$$

where $|Y_{j,k}\rangle_N = [|x_j\rangle_N \ldots |x_k\rangle_N]$. Also let $|\varepsilon_{m,i}^f\rangle_N$, $|\varepsilon_{m,i}^b\rangle_N$ be the projection error vectors of $|x_i\rangle_N$ and $|x_{i-m}\rangle_N$ on $Y_{i-m,i-1,N}$ and $Y_{i-m+1,i,N}$, respectively. *That is,*

$$|\varepsilon_{m,i}^f\rangle_N = P_{i-m,i-1,N}^{\perp}|x_i\rangle_N$$
$$|\varepsilon_{m,i}^b\rangle_N = P_{i-m+1,i,N}^{\perp}|x_{i-m}\rangle_N \qquad (20)$$

where $P^{\perp} = I - P$. The last components of (20) satisfy the recursions

$$\varepsilon_{m+1,i}^f(N) = \varepsilon_{m,i}^f(N) - k_{m+1,i}^f(N)\varepsilon_{m,i-1}^b(N)$$
$$\varepsilon_{m+1,i}^b(N) = \varepsilon_{m,i-1}^b(N) - k_{m+1,i}^b(N)\varepsilon_{m,i}^f(N) \qquad (21)$$

where we define

$$\alpha_{m,i}^f(N) = \langle\varepsilon_{m,i}^f|\varepsilon_{m,i}^f\rangle_N,$$
$$\alpha_{m,i-1}^b(N) = \langle\varepsilon_{m,i-1}^b|\varepsilon_{m,i-1}^b\rangle_N,$$
$$\beta_{m,i}(N) = \langle\varepsilon_{m,i}^f|\varepsilon_{m,i-1}^b\rangle_N, \qquad (22)$$
$$k_{m+1,i}^f(N) = \frac{\beta_{m,i}(N)}{\alpha_{m,i-1}^b(N)},$$
$$k_{m+1,i}^b(N) = \frac{\beta_{m,i}(N)}{\alpha_{m,i}^f(N)}$$

and we also note

$$\varepsilon^b_{m,0}(N) = \varepsilon^b_{m,d}(N-1).$$

The time-updated recursions for $\alpha^f_{m,i}(N)$, $\alpha^b_{m,i-1}(N)$ and $\beta_{m,i}(N)$ can be obtained from the general formula [2]. We define the quantity called "likelihood variable" by

$$cos^2\theta_{i-m,i-1}(N) = \langle\pi|P^{\perp}_{i-m,i-1}|\pi\rangle_N. \quad (23)$$

For $m=0$, $Y_{i,i-1,N}$ is empty, so that $P^{\perp}_{i,i-1,N} = I$, and $cos^2\theta_{i,i-1}(N) = 1$. Then the time-updated recursions are given by

$$\begin{aligned}
\alpha^f_{m,i}(N) &= \lambda\alpha^f_{m,i}(N-1) \\
&\quad +(\varepsilon^f_{m,i}(N))^2 sec^2\theta_{i-m,i-1}(N) \\
\alpha^b_{m,i-1}(N) &= \lambda\alpha^b_{m,i-1}(N-1) \\
&\quad +(\varepsilon^b_{m,i-1}(N))^2 sec^2\theta_{i-m,i-1}(N) \qquad (24) \\
\beta_{m,i}(N) &= \lambda\beta_{m,i}(N-1) \\
&\quad +\varepsilon^f_{m,i}(N)\varepsilon^b_{m,i-1}(N) sec^2\theta_{i-m,i-1}(N).
\end{aligned}$$

The order-updated recursion for $cos^2\theta_{i-m,i-1}(N)$ is obtained from (37) and (44) as

$$\begin{aligned}
cos^2\theta_{i-m-1,i-1}(N) &= cos^2\theta_{i-m,i-1}(N) \\
&\quad -(\varepsilon^b_{m,i-1}(N))^2/\alpha^b_{m,i-1}(N). \qquad (25)
\end{aligned}$$

Next we present the parameter estimation algorithm of the joint process circular lattice . We denote the least squares estimator of $y(N)$ based on $Y_{d-m,d,N}$ and its estimation error by $z_m(N)$ and $\nu_m(N)$, respectively, where $0 \le m \le (p+1)d-1$. *That is,*

$$|\nu_m\rangle_N = P^{\perp}_{d-m,d,N}|y\rangle_N \qquad (26)$$

The last component of (26) satisfies

$$\begin{aligned}
\nu_m(N) &= \nu_{m-1}(N) - \delta_m(N)\varepsilon^b_{m,d}(N) \qquad (27) \\
\delta_m(N) &= \frac{\langle\varepsilon^b_{m,d}|\nu_{m-1}\rangle_N}{\langle\varepsilon^b_{m,d}|\varepsilon^b_{m,d}\rangle_N} = \frac{\Delta_m(N)}{\alpha^b_{m,d}(N)}.
\end{aligned}$$

The time-updated recursion for $\Delta_m(N)$ is given similarly by

$$\begin{aligned}
\Delta_m(N) &= \lambda\Delta_m(N-1) \\
&\quad +\nu_{m-1}(N)\varepsilon^b_{m,d}(N) sec^2\theta_{d-m+1,d}(N).
\end{aligned}$$

Finally, we present the *a priori* error form of the above LSCL algorithm. First, we define the *a priori* forward and backward prediction errors by

$$\begin{aligned}
e^f_{m,i}(N) &= \frac{\varepsilon^f_{m,i}(N)}{\gamma_{m,i}(N)} \\
e^b_{m,i-1}(N) &= \frac{\varepsilon^b_{m,i-1}(N)}{\gamma_{m,i}(N)} \qquad (28)
\end{aligned}$$

where we define

$$\gamma_{m,i}(N) = cos^2\theta_{i-m,i-1}(N). \qquad (29)$$

Substituting (28) into (21) we have

$$\begin{aligned}
e^f_{m+1,i}(N) &= \frac{\gamma_{m,i}(N)}{\gamma_{m+1,i}(N)}(e^f_{m,i}(N) \\
&\quad -k^f_{m+1,i}(N)e^b_{m,i-1}(N)) \qquad (30) \\
e^b_{m+1,i}(N) &= \frac{\gamma_{m,i}(N)}{\gamma_{m+1,i+1}(N)}(e^b_{m,i-1}(N) \\
&\quad -k^b_{m+1,i}(N)e^f_{m,i}(N)).
\end{aligned}$$

Also, using (24) in (22) we have

$$\begin{aligned}
k^b_{m+1,i}(N) &= (\lambda\beta_{m,i}(N-1) \\
&\quad +\gamma_{m,i}(N)e^f_{m,i}(N)e^b_{m,i-1}(N))/\alpha^f_{m,i}(N) \\
&= k^b_{m+1,i}(N-1) + \frac{\gamma_{m,i}(N)e^f_{m,i}(N)}{\alpha^f_{m,i}(N)} \times \\
&\quad (e^b_{m,i-1}(N) + k^b_{m+1,i}(N-1)e^f_{m,i}(N))
\end{aligned}$$

and

$$\begin{aligned}
k^f_{m+1,i}(N) &= k^f_{m+1,i}(N-1) + \frac{\gamma_{m,i}(N)e^b_{m,i-1}(N)}{\alpha^b_{m,i-1}(N)} \\
&\quad \times(e^f_{m,i}(N) + k^f_{m+1,i}(N-1)e^b_{m,i-1}(N))
\end{aligned}$$

In turn, substituting these into (30) gives

$$\begin{aligned}
e^f_{m+1,i}(N) &= \xi_{m,i}(N)(e^f_{m,i}(N) \\
&\quad -k^f_{m+1,i}(N-1)e^b_{m,i-1}(N)) \\
e^b_{m+1,i}(N) &= \zeta_{m,i}(N)(e^b_{m,i-1}(N) \\
&\quad -k^b_{m+1,i}(N-1)e^f_{m,i}(N))
\end{aligned}$$

where

$$\xi_{m,k}(N) = \frac{\gamma_{m,i}(N)}{\gamma_{m+1,i}(N)}\left(1 - \frac{\gamma_{m,i}(N)(e^b_{m,i-1}(N))^2}{\alpha^b_{m,i-1}(N)}\right)$$

$$\zeta_{m,k}(N) = \frac{\gamma_{m,i}(N)}{\gamma_{m+1,i+1}(N)}\left(1 - \frac{\gamma_{m,i}(N)(e^f_{m,i}(N))^2}{\alpha^f_{m,i}(N)}\right).$$

But from (25) and the defintions (28) and (29) it is easy to see that $\xi_{m,k}(N) = 1$. Also, from (23) it follows that

$$\gamma_{m+1,i+1}(N) = \langle \pi | P^{\perp}_{i-m,i} | \pi \rangle_N$$

$$= \gamma_{m,i}(N) - \frac{(\gamma_{m,i}(N)e^f_{m,i}(N))^2}{\alpha^f_{m,i}(N)} \qquad (31)$$

so that $\zeta_{m,i}(N) = 1$. Summarizing the above results, we have the following algorithm.

Table 1 *A Priori Error Form of The LSCL Algorithm*

i) Initial conditions ($N = 0$)
For all $i = 1, \ldots, d$ and m, set

$$\begin{aligned} \alpha^f_{m,i}(0) &= \alpha^b_{m,i}(0) = \text{small positive value} \\ k^f_{m,i}(0) &= k^b_{m,i}(0) = 0 \end{aligned}$$

ii) Main CL part
Do for $i = 1, \ldots, d$

$$\begin{aligned} e^f_{0,i}(N) &= e^b_{0,i}(N) = x_i(N) \\ \gamma_{0,i}(N) &= 1 \end{aligned}$$

Do for $m = 0, \ldots, pd + i - 1$
a) order update ($m \to m+1$)

$$\begin{aligned} e^f_{m+1,i}(N) &= e^f_{m,i}(N) \\ &\quad -k^f_{m+1,i}(N-1)e^b_{m,i-1}(N) \\ e^b_{m+1,i}(N) &= e^b_{m,i-1}(N) \\ &\quad -k^b_{m+1,i}(N-1)e^f_{m,i}(N) \\ \gamma_{m+1,i+1}(N) &= \gamma_{m,i+1}(N) \\ &\quad -\frac{(\gamma_{m,i+1}(N)e^b_{m,i}(N))^2}{\alpha^b_{m,i}(N)} \end{aligned}$$

b) time update ($N-1 \to N$)

$$\alpha^f_{m,i}(N) = \lambda\alpha^f_{m,i}(N-1) + (e^f_{m,i}(N))^2\gamma_{m,i}(N)$$

$$\alpha^b_{m,i}(N) = \lambda\alpha^b_{m,i}(N-1) + (e^b_{m,i}(N))^2\gamma_{m,i+1}(N)$$

$$\begin{aligned} k^f_{m+1,i}(N) &= k^f_{m+1,i}(N-1) \\ &\quad +\frac{\gamma_{m,i}(N)e^b_{m,i-1}(N)}{\alpha^b_{m,i-1}(N)}e^f_{m+1,i}(N) \\ k^b_{m+1,i}(N) &= k^b_{m+1,i}(N-1) \\ &\quad +\frac{\gamma_{m,i}(N)e^f_{m,i}(N)}{\alpha^f_{m,i}(N)}e^b_{m+1,i}(N) \end{aligned}$$

where

$$\alpha^b_{m,0}(N) = \alpha^b_{m,d}(N-1),\ e^b_{m,0}(N) =$$

$$e^b_{m,d}(N-1),\ \gamma_{m,1}(N) = \gamma_{m,d+1}(N-1)$$

iii) Joint process part
c) initial condition ($N = 0$)
Set for all m as

$$\delta_m(0) = 0$$

d) initial condition ($m = 0$)

$$\bar{\nu}_{-1}(N) = y(N)$$

Do for $m = 0$ to $pd + d - 1$
e) order update ($m - 1 \to m$)

$$\bar{\nu}_m(N) = \bar{\nu}_{m-1}(N) - \delta_m(N-1)e^b_{m,d}(N)$$

f) time update ($N - 1 \to N$)

$$\delta_m(N) = \delta_m(N-1) + \frac{\gamma_{m,d+1}(N)e^b_{m,d}(N)}{\alpha^b_{m,d}(N)}$$

$$\times\bar{\nu}_m(N)$$

Comparing the above algorithm and (31) with (15) and (16) we see that both have the same structure. Since (15) and (16) are equivalent to (13) and (14), the above algorithm can be also implemented by a network of the angle computer and the rotor using CORDIC. Fig. 4 (a) shows the whole block diagram of our adaptive CL filter. Fig. 4 (b) shows the detailed structure of each building block.

References

[1] F. Ling, "Givens rotation based least squares lattice and related algorithms," IEEE Trans. Signal Process., vol. 39, pp. 1541-1551, 1991.

[2] T. Kawase, H. Sakai, and H. Tokumaru, "Recursive least squares circular lattice and escalator estimation algorithms," IEEE Trans. Acoust.,Speech, Signal Process., vol. ASSP-31, pp. 228-231, 1983.

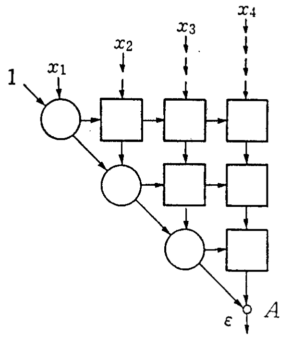


Fig. 1 The block diagram of the triangular systolic array where $y = x_4$ (d=3).

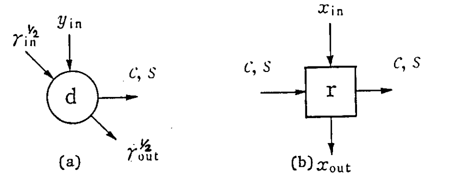


Fig. 2 The cell description (a) angle comuter (13) (b) rotor (14).

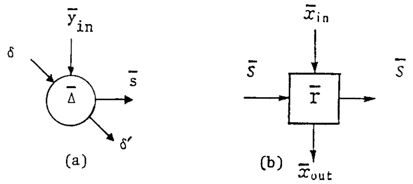


Fig. 3 The cell description (a) angle computer (15) (b) rotor (16) (square-root-free realization).

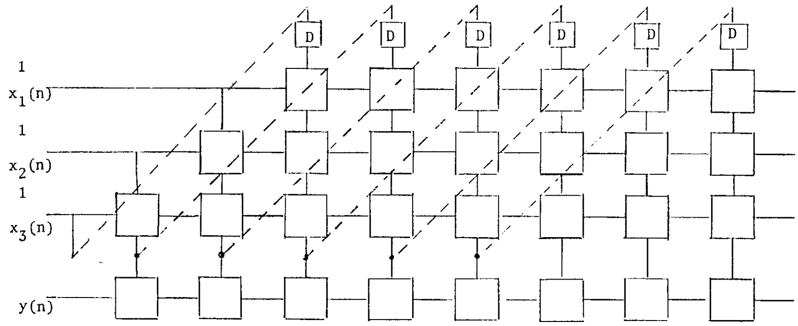
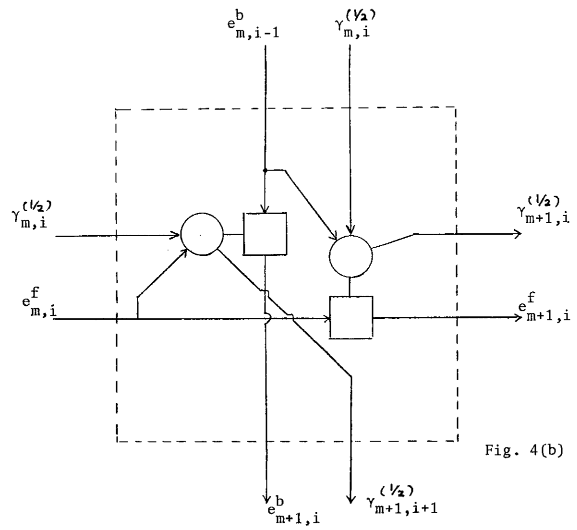


Fig. 4 (a) The block diagram of the CL filter where the detail of each unit is shown in Fig. 4 (b) (d = 3, p = 2).



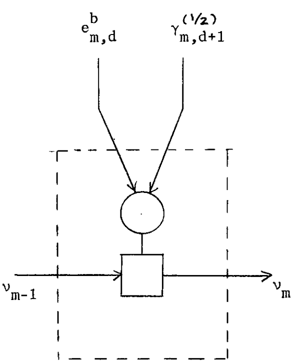


Fig. 4(b) The block diagram of each unit where a circle denotes the angle computer and a square denotes the rotor.