Fifth International Symposium on Signal Processing and its Applications, ISSPA '99, Brisbane, Australia, 22-25 August, 1999
Organised by the Signal Processing Research Centre, QUT, Brisbane, Australia

DESIGN OF PERFECT RECONSTRUCTION INTEGER-MODULATED FILTER BANKS

Alfred Mertins
University of Wollongong
School of Electrical, Computer and Telecommunications Engineering
Wollongong, NSW 2522, Australia

Tanja Karp
University of Mannheim
Lehrstuhl für Elektrotechnik, B6/26
D-68131 Mannheim, Germany

Jörg Kliewer
University of Kiel
Institute for Circuits and Systems Theory
Kaiserstr. 2, D-24143 Kiel, Germany

ABSTRACT

In this paper, new methods for the design of integer-modulated filter banks are presented. The novelty lies in the design of the modulation vectors, which are not restricted to certain structures like dyadic symmetry. In addition, an efficient implementation based on Householder factorizations is presented. In all cases considered, both the filter prototype and the modulation sequences are composed of integers while maintaining perfect reconstruction for arbitrary input signals.

1. INTRODUCTION

In modulated filter banks, all analysis and synthesis filters are modulated versions of a single prototype. Such systems are widely used in signal processing applications, because they allow very efficient implementations based on polyphase filtering and a modulating transform. Both perfect reconstruction [1, 2, 3] and near perfect reconstruction [4] schemes are known.

When a perfect reconstruction (PR) filter bank is to be implemented on a processor with finite-precision arithmetic, the prototype and modulating sequences usually have to be quantized and the PR property gets lost. It is therefore of significant interest to have filter banks that allow PR with finite precision arithmetic. This means that both the prototype and the modulation sequences should be composed of integers. Different design methods for integer coefficient prototypes can be found in [5, 6]. Completely integer modulated PR filter banks have been presented in [7] where the integer modulation sequences are designed on the basis of the dyadic symmetry principle [8]. However, the design in [7] is restricted to filter-lengths $L = 2M$, where M is the number of bands.

In this paper, a new approach to the design of integer-modulated filter banks is presented where the modulating sequences are not restricted to dyadic symmetry. This allows better approximation of the ideal cosine functions when integer sequences are desired and PR has to be maintained. Furthermore, no restriction on the filter length is imposed.

2. COSINE-MODULATED FILTER BANKS

In cosine-modulated filter banks the analysis and synthesis filters, denoted as $h_k(n)$ and $g_k(n)$, $k = 0, \ldots, M-1$ are derived from FIR prototypes via cosine modulation. Various modulation schemes and filter design strategies have been proposed in the literature. Examples are critically subsampled filter banks based on linear-phase prototypes, biorthogonal filter banks with low reconstruction delay, and oversampled filter banks.

In this work, we consider critical subsampling, an even number of bands, M, and usage of the same prototype filter for both analysis and synthesis. A suitable modulation scheme for this case is [3, 9, 10]

$$h_k(n) = 2p(n)\cos\left[\frac{\pi}{M}\left(k+\frac{1}{2}\right)\left(n-\frac{D}{2}\right)+\phi_k\right] \quad (1)$$

$$g_k(n) = 2p(n)\cos\left[\frac{\pi}{M}\left(k+\frac{1}{2}\right)\left(n-\frac{D}{2}\right)-\phi_k\right] \quad (2)$$

with $n = 0, \ldots, L-1$, $k = 0, \ldots, M-1$ and $\phi_k = (-1)^k\pi/4$. The parameter L denotes the length of the prototype, and D is the overall delay of the analysis-synthesis system.

To outline the perfect reconstruction (PR) conditions we use the polyphase notation. The analysis polyphase matrix can be written as [10]

$$\boldsymbol{E}(z) = \boldsymbol{T}_1 \begin{bmatrix} \boldsymbol{P}_0(z^2) \\ z^{-1}\boldsymbol{P}_1(z^2) \end{bmatrix}, \quad (3)$$

where

$$\begin{aligned} [\boldsymbol{T}_1]_{k,j} &= 2\cos\left[\tfrac{\pi}{M}\left(k+\tfrac{1}{2}\right)\left(j-\tfrac{D}{2}\right)+\phi_k\right], \\ & k = 0, \ldots, M-1,\ j = 0, \ldots, 2M-1, \end{aligned} \quad (4)$$

and

$$\begin{aligned} \boldsymbol{P}_0(z^2) &= \operatorname{diag}\left[P_0(-z^2), \ldots, P_{M-1}(-z^2)\right], \\ \boldsymbol{P}_1(z^2) &= \operatorname{diag}\left[P_M(-z^2), \ldots, P_{2M-1}(-z^2)\right]. \end{aligned} \quad (5)$$

The terms $P_j(z)$ denote the type-1 polyphase components of the prototype, given by

$$P_j(z) = \sum_\ell p(2\ell M + j)\, z^{-\ell}, \quad j = 0, \dots, 2M-1. \quad (6)$$

For the synthesis polyphase matrix we get accordingly

$$\boldsymbol{R}(z) = \left[z^{-1}\boldsymbol{Q}_1(z^2),\ \boldsymbol{Q}_0(z^2)\right]\ \boldsymbol{T}_2^{\mathrm{T}}, \quad (7)$$

where

$$[\boldsymbol{T}_2]_{k,j} = 2\cos\left[\tfrac{\pi}{M}\left(k+\tfrac{1}{2}\right)\left(2M-1-j-\tfrac{D}{2}\right) - \phi_k\right], \quad k = 0, \dots, M-1,\ j = 0, \dots, 2M-1, \quad (8)$$

and

$$\begin{aligned} \boldsymbol{Q}_0(z^2) &= \operatorname{diag}\left[P_{M-1}(-z^2), \dots, P_0(-z^2)\right], \\ \boldsymbol{Q}_1(z^2) &= \operatorname{diag}\left[P_{2M-1}(-z^2), \dots, P_M(-z^2)\right]. \end{aligned} \quad (9)$$

Although the delay can be chosen independently of the filter length and the number of channels, we here assume (for the sake of brevity) the most common form of delay, given by $D = 2sM + 2M - 1$ with s being an integer. The PR conditions for the prototype can then easily be derived by inserting (3) and (7) into the polyphase-domain PR conditions given by

$$\boldsymbol{R}(z)\boldsymbol{E}(z) = z^{-2s-1}\,\boldsymbol{I}_M \quad (10)$$

and considering the fact that [3]

$$\boldsymbol{T}_2^{\mathrm{T}}\boldsymbol{T}_1 = (-1)^s\, 2M\, \boldsymbol{I}_{2M} + 2M \begin{bmatrix} \boldsymbol{J}_M & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{J}_M \end{bmatrix}. \quad (11)$$

This yields

$$P_{2M-1-k}(z)\,P_k(z) + P_{M+k}(z)\,P_{M-1-k}(z) \stackrel{!}{=} \frac{z^{-s}}{2M} \quad (12)$$

for $k = 0, \dots, \frac{M}{2} - 1$. In the special case of a linear-phase prototype of length $L = 2(s+1)M$, eq. (12) can be rewritten as

$$\tilde{P}_k(z)P_k(z) + \tilde{P}_{M+k}(z)P_{M+k}(z) = \frac{1}{2M}. \quad (13)$$

The interesting fact is that the PR conditions for the prototype are somehow separated from those for the modulation scheme. Given a prototype $P(z)$ whose polyphase components satisfy (12) one can maintain the PR property if $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ are replaced by other matrices than the ones defined in (4) and (8), as long as the new matrices jointly satisfy (11). This separability will be used in the next section in order to derive integer modulation schemes.

3. INTEGER MODULATION SCHEMES

In this section, we discuss possibilities of replacing the cosine matrices $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ by other matrices $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ that solely contain integers and satisfy (11) up to a scale factor ε/M:

$$\boldsymbol{U}_2^{\mathrm{T}}\boldsymbol{U}_1 = (-1)^s\, 2\varepsilon\, \boldsymbol{I}_{2M} + 2\varepsilon \begin{bmatrix} \boldsymbol{J}_M & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{J}_M \end{bmatrix}. \quad (14)$$

This problem has also been addressed in [7], where the principle of dyadic symmetry was used to design integer matrices. We here outline new design techniques, which are not restricted to having the symmetry property and allow a better match between $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ and the ideal cosine-based matrices $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$.

It is easily verified that the condition (14) is met by matrices $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ of the type

$$\begin{aligned} \boldsymbol{U}_1 &= \boldsymbol{V}_1\left[\boldsymbol{I}_M + \boldsymbol{J}_M,\ \boldsymbol{I}_M - \boldsymbol{J}_M\right] \\ \boldsymbol{U}_2 &= (-1)^s\,\boldsymbol{V}_2\left[\boldsymbol{I}_M + \boldsymbol{J}_M,\ \boldsymbol{I}_M - \boldsymbol{J}_M\right] \end{aligned} \quad (15)$$

where $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$ are $M \times M$ matrices that satisfy

$$\boldsymbol{V}_2^T\boldsymbol{V}_1 = \varepsilon\,\boldsymbol{I}_M. \quad (16)$$

Note that the original cosine matrices may also be expressed in the form (15). For example, $\boldsymbol{T}_1$ can be written as

$$\boldsymbol{T}_1 = \boldsymbol{V}_c\left[\boldsymbol{I}_M + \boldsymbol{J}_M,\ \boldsymbol{I}_M - \boldsymbol{J}_M\right]. \quad (17)$$

Given $\boldsymbol{T}_1$, the matrix $\boldsymbol{V}_c$ can be found as

$$\boldsymbol{V}_c = \frac{1}{4}\boldsymbol{T}_1\left[\boldsymbol{I}_M + \boldsymbol{J}_M,\ \boldsymbol{I}_M - \boldsymbol{J}_M\right]^T. \quad (18)$$

The task of designing suitable integer matrices $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ is equivalent to finding integer matrices $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$ that are inverse to one another up to the factor ε. In the design process we may either consider a general approach with two different matrices $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$, or we may look at the special case $\boldsymbol{V}_2 = \boldsymbol{V}_1 = \boldsymbol{V}$. We follow the latter one, because here the same modulation sequences are used on the analysis and synthesis sides.

We next discuss two methods of designing a matrix $\boldsymbol{V}$ with integer entries that is unitary up to a scale factor:

$$\boldsymbol{V}^T\boldsymbol{V} = \varepsilon\,\boldsymbol{I}. \quad (19)$$

3.1. Design based on Householder Factorizations

Every $M \times M$ unitary matrix $\boldsymbol{V}_c$ can be decomposed into a set of M Householder matrices $\boldsymbol{H}_i$

$$\boldsymbol{V}_c = \boldsymbol{H}_M \cdots \boldsymbol{H}_1\boldsymbol{D}, \quad \boldsymbol{H}_i = \boldsymbol{I} - 2\boldsymbol{u}_i\boldsymbol{u}_i^T \quad (20)$$

where

$$\boldsymbol{u}_i = \frac{\boldsymbol{v}_i \pm ||\boldsymbol{v}_i||\boldsymbol{e}_i}{||\boldsymbol{v}_i \pm ||\boldsymbol{v}_i||\boldsymbol{e}_i||}, \quad \boldsymbol{D} = \mathrm{diag}(\pm 1, \dots, \pm 1) \quad (21)$$

and $\boldsymbol{v}_i$, $\boldsymbol{e}_i$ being the i-th column of $\boldsymbol{V}_c$, $\boldsymbol{I}$, respectively. The $\pm$ signs have to be chosen such that the vectors $\boldsymbol{u}_i$ are non-zero. As described in [11], we can quantize the vectors $\boldsymbol{u}_i$ to obtain an integer implementation based on non-unit norm vectors $\hat{\boldsymbol{u}}_i$, obtaining the following realization:

$$\boldsymbol{V} = \hat{\boldsymbol{H}}_M \cdots \hat{\boldsymbol{H}}_1 \boldsymbol{D}, \quad \hat{\boldsymbol{H}}_i = ||\hat{\boldsymbol{u}}_i||^2 \boldsymbol{I} - 2\hat{\boldsymbol{u}}_i \hat{\boldsymbol{u}}_i^T \quad (22)$$

The matrix $\boldsymbol{V}$ obtained this way contains integer elements and satisfies (19). However, notice that when M increases the elements in $\boldsymbol{V}$ tend to become very large, even if the elements of $\hat{\boldsymbol{u}}_i$ are of moderate size. Instead of implementing $\boldsymbol{V}$ directly, it is therefore advantageous to implement $\boldsymbol{V}$ in the factorized form (22). The multiplication of a vector $\boldsymbol{x}$ with a matrix $\boldsymbol{H}_i$ can then be carried out as $\boldsymbol{H}_i \boldsymbol{x} = ||\hat{\boldsymbol{u}}_i||^2 \boldsymbol{x} - 2\hat{\boldsymbol{u}}_i[\hat{\boldsymbol{u}}_i^T \boldsymbol{x}]$.

3.2. Design based on Projection Matrices

To design matrix $\boldsymbol{V}$, we start with a single arbitrary vector $\boldsymbol{v}_1$ that contains integer values. This may be an integer approximation of one of the columns of $\boldsymbol{V}_c^T$. A second vector $\boldsymbol{v}_2$ that is orthogonal to $\boldsymbol{v}_1$ is easily found as $\boldsymbol{v}_2 = \boldsymbol{Z}\boldsymbol{v}_1$ with $\boldsymbol{Z} = \boldsymbol{J}_M \,\mathrm{diag}\,[1, -1, \dots, 1, -1]$. The two vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ will later form two columns of the matrix $\boldsymbol{V}^T$.

The next step is to store $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ in a matrix $\boldsymbol{A} = [\boldsymbol{v}_1, \boldsymbol{v}_2]$ and to compute the projection matrix

$$\boldsymbol{B} = \boldsymbol{I}_M - \boldsymbol{A}\,[\boldsymbol{A}^T \boldsymbol{A}]^{-1} \boldsymbol{A}^T \quad (23)$$

Then, we form a matrix $\boldsymbol{C}$ from a subset of columns of $\boldsymbol{B}$ and express the vector $\boldsymbol{v}_3$ as

$$\boldsymbol{v}_3 = \boldsymbol{C}\,\boldsymbol{p}_3, \quad (24)$$

where $\boldsymbol{p}_3$ is a parameter vector that needs to be found. Clearly, all linear combinations of the columns of $\boldsymbol{C}$ are orthogonal to $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$. Gathering only a linearly independent subset of the columns of $\boldsymbol{B}$ in $\boldsymbol{C}$ has the effect that the number of unknowns can be reduced. The problem of finding a vector $\boldsymbol{v}_3$ that solely contains integers and has the same ℓ_2-norm as $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ remains. These conditions are usually met by a number of integer parameter vectors $\boldsymbol{p}_3$, so that one can choose a solution which leads to a close match between $\boldsymbol{v}_3$ and one of the columns of $\sqrt{\varepsilon}\,\boldsymbol{V}_c^T$. Then, given $\boldsymbol{v}_3$, a vector $\boldsymbol{v}_4$ can easily be found as $\boldsymbol{v}_4 = \boldsymbol{Z}\boldsymbol{v}_3$.

In the next iteration we define a new matrix $\boldsymbol{A} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4]$, compute a new matrix $\boldsymbol{B}$ according to (23), form $\boldsymbol{C}$, design $\boldsymbol{v}_5$ as $\boldsymbol{v}_5 = \boldsymbol{C}\,\boldsymbol{p}_5$ and create $\boldsymbol{v}_6 = \boldsymbol{Z}\boldsymbol{v}_5$. The procedure can be continued until M vectors $\boldsymbol{v}_1, \dots, \boldsymbol{v}_M$ are found, and $\boldsymbol{V}$ can finally be formed as $\boldsymbol{V} = [\boldsymbol{v}_1, \dots, \boldsymbol{v}_M]^T$. To reduce the search effort, initial guesses $\tilde{\boldsymbol{p}}_k$ can be computed as $\tilde{\boldsymbol{p}}_k = \mathrm{round}(\alpha[\boldsymbol{C}^T\boldsymbol{C}]^{-1}\boldsymbol{C}^T\boldsymbol{v}_k^c)$. Here, $\boldsymbol{v}_k^c$ is the column of $\boldsymbol{V}_c$ that is to be approximated, and α is a scale factor that ensures that $\tilde{\boldsymbol{p}}_k^T \tilde{\boldsymbol{p}}_k \approx \varepsilon$.

4. INTEGER PROTOTYPES

The design of integer prototypes is straightforward for length-$2M$ filters. In this case, the polyphase filters $P_j(z)$ degenerate to single coefficients, $P_j(z) = p(j)$, and the condition (13) for the linear-phase case becomes $p^2(j) + p^2(M+j) = \gamma$, where $\gamma \in \mathbb{N}$ is a scale factor that has to be introduced if the coefficients are integers. For certain values of γ a number of integer combinations $\{p(j), p(M+j)\}$ can be found, and reasonably well prototypes can be designed [7].

A design method for prototypes with $L > 2M$ is described in [6]. We here list coefficients of some PR integer prototypes designed with this method that will be used in the design examples discussed in the next section.

Table 1

Perfect reconstruction prototypes with integer coefficients. (a) and (b): 4-band linear-phase prototypes; (c) and (d): 8-band linear-phase prototypes.

n:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
(a)	-1	0	0	2	4	6	7	8								
(b)	-14	-6	7	33	56	96	112	132								
(c)	-1	-1	0	0	0	0	2	2	4	4	6	6	7	7	8	8
(d)	-6	-4	0	-6	7	0	8	17	24	33	41	48	56	62	66	68

5. DESIGN EXAMPLES

Examples of integer matrices $\boldsymbol{V}$ for $M = 4$ and $M = 8$ are given in Table 2. The frequency responses of entire integer-modulated filter banks are depicted in Figure 1. As Figure 1(b) shows, for $M = 4$ a good performance can be obtained with relatively small integers for both the prototype and the modulation matrix. For $M = 8$ (or more) a high stopband attenuation requires a high precision of the modulation sequences, as the comparison of Figures 1(a)-(e) shows. The best performance for $M = 8$ is given in Figure 1(e), which depicts the result for an implementation via Householder building blocks. The vectors $\boldsymbol{u}_i$ used to construct these blocks are listed in Table 3. Their elements are small integers, allowing an efficient implementation of the modulation in factorized form. The entries of the entire matrix $\boldsymbol{V}$, however, are extremely large (up to 3×10^{20}) in this case. Fortunately, the full precision is not needed if an integer input signal is to be reconstructed perfectly. A worst case analysis with the given matrix $\boldsymbol{V}$ shows that vectors $\boldsymbol{x}$ containing B bit integers can be reconstructed from their transform coefficients $\boldsymbol{y} = \boldsymbol{V}\boldsymbol{x}$ without error if a $B + 4$

Table 2

Examples of matrices $\boldsymbol{V}$. (a) and (b): $M = 4$; (c) and (d): $M = 8$. The matrices (a), (b) and (c) are designed via the projection method, whereas matrix (d) is designed via Householder factorization.

$$\begin{bmatrix} 2 & 1 & 1 & 0 \\ 0 & -1 & 1 & -2 \\ 1 & 0 & -2 & -1 \\ -1 & 2 & 0 & -1 \end{bmatrix}$$

(a)

$$\begin{bmatrix} 35 & 30 & 20 & 7 \\ 7 & -20 & 30 & -35 \\ 30 & -7 & -35 & -20 \\ -20 & 35 & -7 & -30 \end{bmatrix}$$

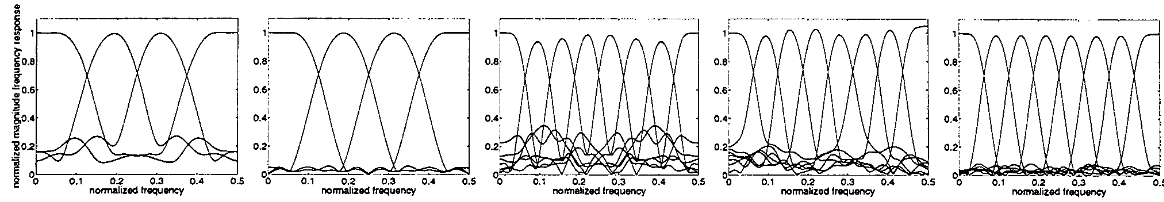
(b)

$$\begin{bmatrix} 2 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 & -1 & 1 & -2 \\ 0 & 2 & -1 & -1 & 1 & -1 & 0 & 1 \\ 1 & 0 & -1 & -1 & -1 & 1 & 2 & 0 \\ -1 & 1 & 2 & 0 & -1 & 0 & 1 & 1 \\ 1 & -1 & 0 & 1 & 0 & -2 & 1 & 1 \\ 1 & 1 & 0 & 0 & -2 & -1 & -1 & -1 \\ -1 & 1 & -1 & 2 & 0 & 0 & 1 & -1 \end{bmatrix}$$

(c)

$$\begin{bmatrix} -720 & -1080 & -1080 & -1080 & -1080 & -1080 & 0 & 0 \\ -1485 & -810 & -495 & 450 & 450 & 1395 & 945 & 0 \\ -1296 & -432 & 1080 & 1080 & 72 & -936 & -1008 & -504 \\ -621 & 738 & 1305 & -270 & -1278 & -81 & 1197 & 756 \\ 1011 & -1238 & -55 & 1450 & -342 & -489 & 693 & 924 \\ 657 & -726 & 555 & 30 & -1314 & 1017 & -189 & -1512 \\ 168 & -1204 & 1120 & -1120 & 504 & 588 & -756 & 1092 \\ 282 & -536 & 710 & -620 & 1116 & -858 & 1386 & -1092 \end{bmatrix}$$

(d)



(a) (b) (c) (d) (e)

Figure 1: Magnitude frequency responses of integer-modulated filter banks. (a) $M = 4$, $P(z)$ and $\boldsymbol{V}$ from Tables 1(a) and 2(a); (b) $M = 4$, $P(z)$ and $\boldsymbol{V}$ from Tables 1(b) and 2(b); (c) $M = 8$, $P(z)$ and $\boldsymbol{V}$ from Tables 1(c) and 2(c); (d) $M = 8$, $P(z)$ and $\boldsymbol{V}$ from Tables 1(d) and 2(d); (e) $M = 8$, $P(z)$ from Table 1(d) and vectors $\boldsymbol{u}_i$ from Table 3.

bit representation for the elements of $\boldsymbol{y}$ and all intermediate results is used.

6. CONCLUSIONS

We have presented two methods for the design of modulation sequences for integer-modulated perfect reconstruction filter banks. The subspace projection method is well suited for the design of sequences containing small integers, while the Householder factorization typically results in matrices of large integers. However, an implementation based on Householder building blocks allows to circumvent the need for high word lengths and results in PR for integer input signals at a very low cost.

Table 3

Vectors $\boldsymbol{u}_i$ for the construction of Householder building blocks. $M = 8$; $\boldsymbol{D} = \boldsymbol{I}_8$.

$\boldsymbol{u}_1$:	12	-11	2	7	7	-7	2	2
$\boldsymbol{u}_2$:	14	-3	-9	8	-4	4	-3	-2
$\boldsymbol{u}_3$:	16	-4	4	4	-4	5	4	7
$\boldsymbol{u}_4$:	15	-3	2	-8	-2	1	-5	3
$\boldsymbol{u}_5$:	16	4	-1	7	1	-1	-5	4
$\boldsymbol{u}_6$:	17	5	-8	-4	-2	-5	3	3
$\boldsymbol{u}_7$:	15	3	9	-2	-5	-5	-1	-4
$\boldsymbol{u}_8$:	11	10	-3	-9	3	7	3	0

7. REFERENCES

[1] H. Malvar, "Extended lapped transforms: Fast algorithms and applications," *IEEE Trans. Signal Processing*, vol. 40, November 1992.

[2] T. Ramstad and J. Tanem, "Cosine modulated analysis-synthesis filter bank with critical sampling and perfect reconstruction," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Toronto, Ont., Canada, pp. 1789–1792, May 1991.

[3] R. Koilpillai and P. Vaidyanathan, "Cosine-modulated FIR filter banks satisfying perfect reconstruction," *IEEE Trans. Signal Processing*, vol. 40, pp. 770–783, April 1992.

[4] J. Rothweiler, "Polyphase quadrature filters - a new subband coding technique," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 1280–1283, 1983.

[5] T. Karp, A. Mertins, and T. Nguyen, "Efficiently VLSI-realizable prototype filters for modulated filter banks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Munich, Germany, vol. 3, pp. 2445–2448, April 1997.

[6] A. Mertins, "Subspace approach for the design of cosine-modulated filter banks with linear-phase prototype filter," *IEEE Trans. Signal Processing*, vol. 46, pp. 2812–2818, October 1998.

[7] M. Bi, S. Ong, and Y. Ang, "Integer-modulated FIR filter banks for image compression," *IEEE Trans. Circ. and Syst. for Video Technology*, vol. 8, pp. 923–927, December 1998.

[8] W. Cham, "development of integer cosine transforms by the principle of dyadic symmetry," *Proc. Inst. Elect. Eng., Part I*, vol. 136, pp. 276–282, August 1989.

[9] H. Malvar, *Signal Processing with Lapped Transforms*. Norwood, MA: Artech House, 1992.

[10] P. N. Heller, T. Karp, and T. Q. Nguyen, "A general formulation for modulated filter banks," *IEEE Trans. Signal Processing*, vol. 47, pp. 986–1002, April 1999.

[11] P. Vaidyanathan, *Multirate Systems and Filter Banks*. Prentice Hall, 1993.