Nonuniform Filter Banks: New Results and Open Problems

Sony Akkarakaran and P.P. Vaidyanathan

Abstract. A nonuniform filter bank (FB) is one whose channel decimation rates need not all be equal. While the theory and design of uniform FBs is a very well developed subject, there are several interesting open issues in the area of nonuniform FBs. Most nonuniform FB designs either result in approximate or *near–perfect* reconstruction, or involve cascading uniform FBs in tree structures. This leaves unanswered many important theoretical issues involved in obtaining perfect reconstruction (PR) in nonuniform FBs. The purpose of this paper is to address these issues. We only study FBs with *integer* decimation rates, as FBs with rational decimators can also be shown to be transformable to them. The central problem of interest is as follows: Let S be a set of positive integers obeying maximal decimation (i.e., with reciprocals summing to unity). Find necessary and sufficient conditions on S for existence of a PRFB belonging to some FB class $\mathcal{C}$ and using S as its set of decimators. The class $\mathcal{C}$ is defined by some constraint on the filters of its constituent FBs; examples of interest are the class of all rational FBs (FBs with rational filters), FIR FBs, orthonormal FBs, etc. A condition that immediately suggests itself is the one stating that the integers be arrangeable in a tree so that the required PRFB can be built by cascading uniform PRFBs in a tree structure. However, this condition, while clearly sufficient, is *not necessary* for many classes $\mathcal{C}$ of interest. In fact there are sets violating it which can be used to build *delay-chain* PRFBs (in which all filters are delays). Many of our new results focus on the class of rational FBs. We strengthen considerably the known necessary conditions in this case, and provide new ones. The basic problem remains unresolved — necessary and sufficient conditions are still unknown, however we believe our work is an important step towards a full solution. We conclude by listing all known conditions, studying their inter-relationship, and pointing out several open problems.

1. Introduction

Figure 1 shows an M-channel nonuniform filter bank (FB). The FB is said to be maximally decimated if the channel decimation rates n_i are integers satisfying

$$\sum_{i=0}^{M-1} \frac{1}{n_i} = 1. \qquad \text{(maximal decimation condition)} \qquad (1.0.1)$$

Beyond Wavelets
J. Stoeckler and G. V. Welland (Eds.), pp. 1–53.

ISBN 0-12-xxxxxx-x

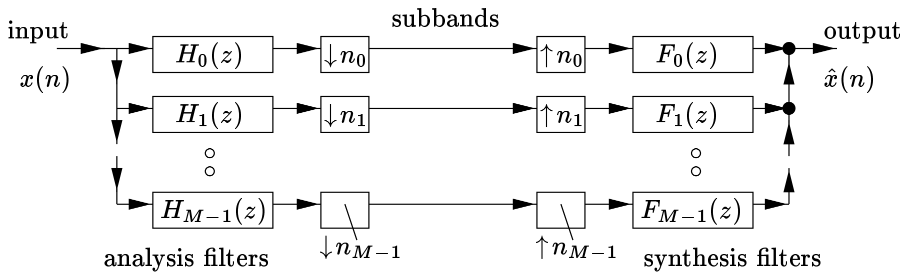


Figure 1. Nonuniform filter bank.

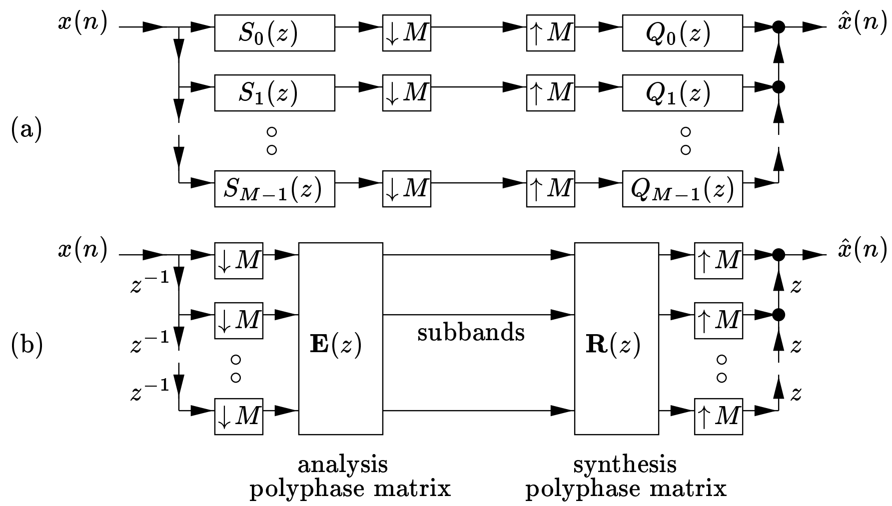


Figure 2. Uniform maximally decimated filter bank. (a) Showing analysis and synthesis filters. (b) Polyphase representation.

Figure 2a shows a maximally decimated *uniform* FB, which is a special case of Fig. 1 where $n_i = M$ for all i. For this case, the system can be equivalently redrawn using the analysis and synthesis polyphase matrices $\mathbf{E}(z)$ and $\mathbf{R}(z)$, as shown in Fig. 2b. The condition for perfect reconstruction (PR) is then easily expressed as $\mathbf{R}(z) = \mathbf{E}^{-1}(z)$. Due to this, the theory and design of uniform PRFBs is an extremely well developed subject. Numerous parameterization results list all possible M-channel uniform PRFBs with various sets of properties such as paraunitariness, FIR filters, linear phase filters, etc.

In contrast, several issues involved in achieving PR in nonuniform FBs remain unresolved. For example, given a general set of positive integers n_i obeying maximal decimation (1.0.1), how do we determine whether or not there exists a rational PRFB (i.e., one with rational filters) using the n_i as decimators? If the n_i are all equal, clearly such a FB exists (as it is then uniform). Similarly, it also exists if the n_i are arrangeable in a tree so that such a PRFB can be built by cascading uniform PRFBs in a tree structure (Section 4.1). This is the most common approach to achieving PR in nonuniform FBs. In particular, it is used to build the FBs that implement the dyadic wavelet transforms [11],[12]: Such a FB has a *dyadic* decimator–set, i.e., one of form $\{2, 2^2, \ldots, 2^{r-1}, 2^r, 2^r\}$ for some integer $r \geq 1$, and is built using a dyadic tree (i.e., one built from a cascade of r 2-channel FBs). However, there are sets of decimators n_i that cannot be arranged in a tree as described above, and yet permit existence of rational PRFBs in which in fact all filters are *delays.* Further, even if the decimators are arrangeable in a tree, it is possible that there are PRFBs using those decimators that cannot be realized using the tree. These facts will be discussed in detail with examples in Section 4.2. Thus *tree structures of uniform PRFBs are far from being a full solution* to the PR problem for nonuniform FBs.

Derivability of decimators from a tree (as described above) is a sufficient condition for existence of rational PRFBs using the decimators. There are certain other conditions that are known to be necessary, e.g., there are no rational PRFBs using the decimator–set $\{2, 3, 6\}$ because no two decimators of such a FB can be coprime (Section 6.1,[4]). However, a *condition that is both necessary and sufficient remains unknown.* The present work studies this and related problems. An important part of our study is to significantly improve upon the known conditions, i.e., to derive new ones, strengthen necessary conditions and weaken sufficient ones. Another contribution is to study the conditions for reducibility of PRFBs to tree structures. For example, it has been shown [3],[10] that all rational PRFBs with dyadic decimator–sets *must* be derivable from dyadic trees. In Section 7, we will considerably generalize this result. Although these problems in their full generality remain unresolved, we believe the present work to be an important step towards a complete understanding of this subject — an area so rich in open problems even after over two decades of filter bank research.

1.1. Relevant earlier work

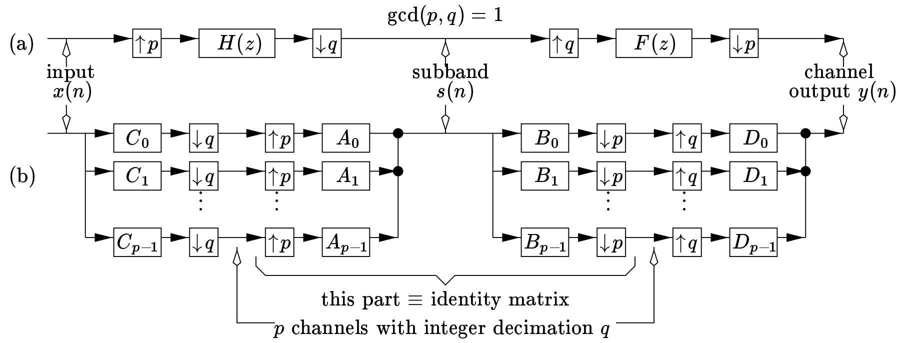
Trees of uniform FBs, and near–PR designs: A very common approach to nonuniform PRFB design is to cascade uniform PRFBs in a tree-structure, e.g., as is done to implement dyadic wavelet transforms [11],[12]. However, as stated earlier, there are nonuniform PRFBs that cannot be built in this manner. Many works deal with *approximate reconstruction* (or 'near–PR') nonuniform FBs, e.g., the frequency domain approaches of Li et al. [7], the time domain methods of Nayebi et al. [8], and other references therein. These are very useful from a practical standpoint, giving FBs with excellent filter responses and low aliasing distortions. However, they do not address the many theoretical issues involved in obtaining *exact* reconstruction.

FBs with fractional decimators: Kovačević and Vetterli have studied a more general system [6] where each channel of the FB has a decimation rate that is fractional, i.e., of form q/p where p, q are coprime positive integers. Such a channel, shown in Fig. 3a, is completely equivalent to the system of Fig. 3b. By this we mean that given any one of these systems, we can choose the filters in the other so that the same input $x(n)$ for both systems always produces the same signals $s(n)$ and $y(n)$ as shown. A choice ensuring this is shown in Fig. 3c (polyphase vectors are defined in Section 1.3). The equivalence under this choice is provable using the discussion on fractional decimation in [11, Section 4.3.3]. If the $A_i(z)$ differ from the special choice of Fig. 3c, we can replace them by this choice and modify the $C_i(z)$ so that the signal $s(n)$ is unaffected. This is done by performing a p-th order polyphase decomposition of the $A_i(z)$, using the fact that p, v are coprime, and moving the resulting polyphase matrix to the left. A similar comment holds for the $B_i(z)$.

From the equivalence shown in Fig. 3, we conclude that the PR problems for integer–decimated and rationally decimated FBs are fully equivalent. Another concern besides PR in rationally decimated FBs is the nature of their spectral analysis: Does a subband represent a contiguous portion of the input spectrum, or do the decimators and expanders in Fig. 3b cause it to contain separate parts, possibly mirrored and shuffled in order? This issue is studied in [6].[1] However, as far as the PR problem is concerned, it is enough to study FBs with integer decimators, and that is the approach we shall use.

Other more general multirate structures: As we will see in Section 2.2, nonuniform PRFBs are hard to design because of certain structural constraints that their associated polyphase matrices must obey. This is the origin of the central problem studied in our work: These structural constraints cannot be obeyed by rational FBs unless their decimators satisfy various conditions, which we aim to characterize. However, the constraints vanish if we use more general systems in the channels of the FB, e.g., if the filters are allowed to be periodically time–

[1] It becomes less serious if we allow modulators at appropriate points within the FB.



(c)

$C_i(z) = z^{iu} R_i(z)$ $\quad$ $A_i(z) = z^{-iv},\ B_i(z) = z^{iv}$ $\quad$ $D_i(z) = z^{-iu} E_i(z)$

$(R_0(z), \ldots, R_{p-1}(z))^T$: p-th order synthesis polyphase vector of $H(z)$ $\quad$ $up - vq = 1$ $\quad$ $(E_0(z), \ldots, E_{p-1}(z))$: p-th order analysis polyphase vector of $F(z)$

Figure 3. FB with rational decimators. (a) Single channel with decimator q/p. (b) Equivalent system of p channels with decimator q. (c) A possible set of filter choices ensuring the equivalence.

varying (Section 2.3). Chen and Qiu [2] and Shenoy [9] have studied multirate and FB design using such more general structures. The PR design then allows as much or even more freedom than that in the well–studied PR designs for the traditional *uniform* FB of Fig. 2. Our work is restricted to the usual nonuniform FB structure of Fig. 1 that does not use such generalized multirate structures.

PR conditions on decimators, and reducibility to tree structures: A necessary condition on the (integer) decimators for PR with rational FBs was first stated in [5]. Called the compatibility condition, it was generalized by Djokovic and Vaidyanathan [4], who also pointed out another such condition (pairwise noncoprimeness). We will considerably generalize these conditions. Another related work has involved showing derivability of FBs using dyadic decimator–sets from dyadic trees [10],[3], as explained earlier. These results too will be significantly strengthened. Among various more general situations studied include certain non–dyadic sets, unconstrained FBs, and tree structures whose constituent FBs need not be uniform.

1.2. Outline

Section 2 reviews the PR conditions on the filters of uniform FBs, and their generalization to nonuniform FBs, derivable using a transformation of nonuniform FBs to equivalent uniform ones. It shows how in spite of this transform, the nonuniform PRFB design does not reduce to a uniform PR design, unless the filters of the nonuniform FB are allowed to be *time varying.* In Section 3 we formally state the central problem, and study its solution for classes of unconstrained FBs (where the filters of the FB have no constraints such as rationality). Section 4 analyzes the role of tree structures in the study of the main problem. It shows how tree structures of uniform PRFBs do not provide a full solution (Section 4.2), and how trees can be used to improve upon known PR conditions on the decimators (Section 4.3). Section 5 solves the central problem of the paper for the class of *delay-chains* (FBs in which all filters are delays): It states the necessary and sufficient condition for a set of decimators to be usable to build a PR delay-chain, and presents algorithms to test the condition. Subsequent sections focus mainly on the class of rational FBs. Section 6 states the earlier known necessary conditions on decimators of rational PRFBs, and generalizes them in several ways. Section 7 generalizes [10],[3] by finding weaker conditions on decimators under which all PRFBs using them can be derived from certain tree structures. Section 8 summarizes all known necessary PR conditions on the decimators, and studies their inter-relationships. We conclude by noting many open problems in the area.

1.3. Notations, definitions and assumptions

Standard notation: Superscripts (*) and (^{T}) denote the complex conjugate and matrix (or vector) transpose respectively. We use boldface letters for matrices and vectors. We use lowercase letters for discrete sequences and uppercase letters for Fourier and z–transforms. Sometimes lowercase boldface letters are used for vector z–transforms. For sequences $\mathbf{h}(n)$ without z–transforms that are rational functions of z, the notation $\mathbf{H}(z)$ is an abbreviation for the Fourier transform $\mathbf{H}(e^{j\omega})$. For LTI transfer matrices $\mathbf{H}(z)$, the 'paraconjugate' $\mathbf{H}^{*T}(1/z^*)$ is denoted by $\widetilde{\mathbf{H}}(z)$. The L-th root of unity, $e^{-j2\pi/L}$ is denoted by W_L, or by W if the subscript value L is understood. The Kronecker delta function is denoted by δ ($\delta(0) = 1$ and $\delta(x) = 0$ if $x \neq 0$).

Polyphase concepts [11]: The M–fold decimator and expander are represented by $\downarrow_M$ and $\uparrow_M$ respectively, as in Fig. 1. Given a sequence $\mathbf{h}(n)$ with z–transform $\mathbf{H}(z)$, its M–fold decimated version is the sequence $\mathbf{g}(n) = \mathbf{h}(Mn)$, with z–transform denoted by $(\mathbf{H}(z))\downarrow_M$. Likewise, the M–fold expanded version of $\mathbf{h}(n)$ is

$$\mathbf{f}(n) = \begin{cases} \mathbf{h}(n/M) & \text{if } n/M \text{ is an integer} \\ \mathbf{0} & \text{otherwise} \end{cases}$$

with z–transform denoted by $(\mathbf{H}(z))\uparrow_M$. With $W = e^{-j2\pi/M}$, we have

$$(\mathbf{H}(z))\downarrow_M = \frac{1}{M}\sum_{i=0}^{M-1}\mathbf{H}(z^{1/M}W^i), \quad \text{and} \quad (\mathbf{H}(z))\uparrow_M = \mathbf{H}(z^M) \qquad (1.3.1)$$

Given filters $H_0(z), H_1(z), \ldots, H_{N-1}(z)$, their M-th order analysis polyphase matrix $\mathbf{E}(z)$ is the $N \times M$ matrix defined by

$$\mathbf{h}(z) \stackrel{\triangle}{=} (H_0(z), H_1(z), \ldots, H_{N-1}(z))^T = \mathbf{E}(z^M)\mathbf{d}(z),$$

where $\mathbf{d}(z) = (1, z^{-1}, \ldots, z^{-(M-1)})^T$ is the length M delay vector. Thus, $\mathbf{E}(z)$ has i-th column $\big(z^i \mathbf{h}(z)\big) \downarrow_M$. Similarly, the M-th order synthesis polyphase matrix of the filters $F_0(z), F_1(z), \ldots, F_{N-1}(z)$ is the $M \times N$ matrix $\mathbf{R}(z)$ obeying

$$\mathbf{f}(z) \stackrel{\triangle}{=} (F_0(z), F_1(z), \ldots, F_{N-1}(z)) = \widetilde{\mathbf{d}}(z)\mathbf{R}(z^M).$$

Thus the i-th row of $\mathbf{R}(z)$ is $\big(z^{-i}\mathbf{f}(z)\big) \downarrow_M$. If the $H_i(z), F_i(z)$ are respectively the analysis and synthesis filters of a FB, then $\mathbf{E}(z), \mathbf{R}(z)$ are respectively said to be the M-th order analysis and synthesis polyphase matrices of the FB. An easily proved result that we often use is the following:

Lemma 1: Polyphase lemma. Let $\mathbf{e}(z), \mathbf{r}(z)$ be the M-th order analysis and synthesis polyphase matrices of the filters $H(z)$ and $F(z)$ respectively. Thus $\mathbf{e}(z)$ is a row vector and $\mathbf{r}(z)$ is a column vector. Then,

$$\mathbf{e}(z)\mathbf{r}(z) = (H(z)F(z)) \downarrow_M \tag{1.3.2}$$

Maximal decimation: **All FBs** studied in the paper are **maximally decimated** with **integer decimation** rates, even if this is not explicitly stated. Similarly, references to a 'set of decimators' (or 'decimator–set') always implicitly mean a set of positive integers (not necessarily distinct) obeying (1.0.1).

2. Background: Equivalent Uniform FBs; PR Equations

The main focus of the paper is to find conditions on the decimators that permit existence of various types of nonuniform perfect reconstruction (PR) FBs with those decimators. To do this, we must first know what conditions on the filters of the FB guarantee the PR property. This section begins by reviewing the PR conditions for uniform FBs. We then review the transformation of a nonuniform FB with decimators n_i to an equivalent uniform FB with a decimation rate L that is a multiple of all the n_i. This yields the general PR conditions for nonuniform FBs, that will be used in all the later sections. In spite of the possible transformation to uniform FBs, the nonuniform PRFB design problem by no means reduces to the uniform PR design. However, such a reduction *does* occur if the nonuniform FB is allowed to have filters that are LPTV(L) (linear periodically time varying with period L) instead of LTI. With LTI filters, achieving PR is tougher, and is the subject of the later sections.

2.1. PR for uniform FBs, and the nonuniform to uniform transform

For the uniform FB of Fig. 2, the problem of achieving PR is very well understood. The following are three equivalent necessary and sufficient conditions on the filters for PR in this case [11]:

1. *Biorthogonality condition.* $(S_i(z)Q_j(z))\downarrow_M = \delta(i-j)$.
2. *AC matrix formulation.* Let $W = e^{-j2\pi/M}$. Then,

$$\begin{bmatrix} A_0(z) \\ A_1(z) \\ \vdots \\ A_{M-1}(z) \end{bmatrix} \triangleq \underbrace{\begin{bmatrix} S_0(z) & \dots & S_{M-1}(z) \\ S_0(zW) & \dots & S_{M-1}(zW) \\ \vdots & \ddots & \vdots \\ S_0(zW^{M-1}) & \dots & S_{M-1}(zW^{M-1}) \end{bmatrix}}_{\text{alias cancellation (AC) matrix } \mathbf{S}(z)} \begin{bmatrix} Q_0(z) \\ Q_1(z) \\ \vdots \\ Q_{M-1}(z) \end{bmatrix} = \begin{bmatrix} M \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{2.1.1}$$

   For any uniform FB (PR or otherwise), the $A_i(z)$ defined above are called the 'aliasing gains'. The PR condition (2.1.1) thus specifies all aliasing gains. It arises from the frequency domain relation between the output $\widehat{X}(z)$ and input $X(z)$ of *any* uniform FB (PR or otherwise):

$$\widehat{X}(z) = \frac{1}{M}\sum_{i=0}^{M-1} A_i(z)X(zW^i) \tag{2.1.2}$$

3. *Polyphase formulation.* If $\mathbf{E}(z), \mathbf{R}(z)$ are respectively the M-th order analysis and synthesis polyphase matrices of the FB (as in Fig. 2b), then $\mathbf{R}(z) = \mathbf{E}^{-1}(z)$. That this is equivalent to the biorthogonality condition stated earlier follows from the polyphase lemma (Section 1.3), which shows that the ij-th entry of $\mathbf{E}(z)\mathbf{R}(z)$ is precisely the quantity $(S_i(z)Q_j(z))\downarrow_M$ occurring in the biorthogonality condition.

Now any nonuniform FB (as in Fig. 1) is transformable into a uniform FB, which we will call its equivalent uniform FB [1],[4],[5],[6]. This transform is described by Fig. 4, which shows how a single channel with decimator n_k is replaceable by p_k channels with decimators $L = n_k p_k$. Repeating this process on all channels of the nonuniform FB, with L as any common multiple of all its decimators n_i (usually $L = \text{lcm}\{n_i\}$), yields a *uniform* L-channel FB. The nonuniform FB has PR if and only if the equivalent uniform FB has PR. The filters in the uniform FB are various delayed versions of those in the nonuniform one. Inserting these relations between the filters into the PR conditions for uniform FBs gives the PR conditions for nonuniform FBs. These conditions, described next, generalize the uniform FB PR conditions, and are heavily used later.

2.2. The general PR conditions for nonuniform FBs

Biorthogonality condition. The uniform FB biorthogonality condition, when applied to the uniform FB derived from the nonuniform one of Fig. 1, is equiv-

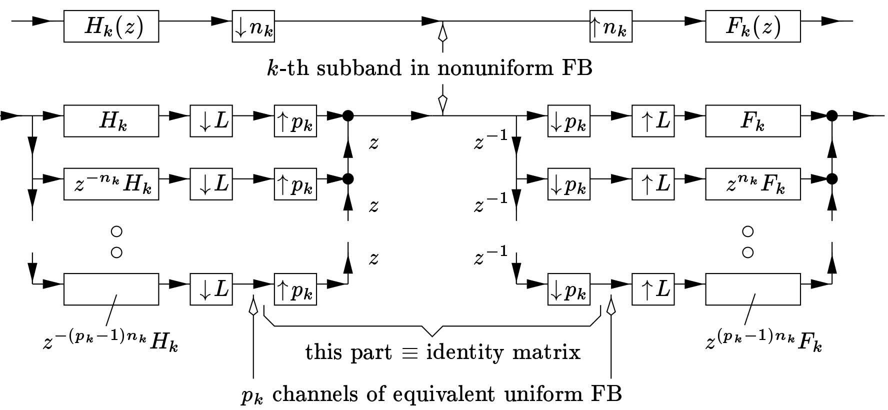


Figure 4. Transforming a nonuniform FB to an equivalent uniform FB.

alent to

$$(H_i(z)F_j(z))\downarrow_{\gcd(n_i,n_j)} = \delta(i-j) \qquad \text{(biorthogonality condition)} \tag{2.2.1}$$

This has been observed earlier [4],[10]. Appendix A contains a proof for easy reference. The condition gets its name from its time-domain equivalent. To describe this, let $h_i(n), f_i(n)$ respectively be the impulse responses of $H_i(z), F_i(z)$. We define two sets of sequences

$$\{\mu_{ik}(n) = h_i^*(kn_i - n) \ \mid i = 0,1,\ldots,M-1,\ k = \text{any integer}\} \tag{2.2.2}$$

$$\{\eta_{jl}(n) = f_j(n - ln_j) \ \mid j = 0,1,\ldots,M-1,\ l = \text{any integer}\} \tag{2.2.3}$$

The action of the FB on its input $x(n)$ is now elegantly expressible using these sequences: The j-th subband signal $c_j()$ and the FB output $\widehat{x}()$ are given by

$$c_j(l) = \Big\langle x(n), \mu_{jl}(n) \Big\rangle = \sum_{n=-\infty}^{\infty} x(n)h_j(ln_j - n), \qquad \text{and}$$

$$\widehat{x}(n) = \sum_{j=0}^{M-1}\sum_{l=-\infty}^{\infty} c_j(l)\eta_{jl}(n) = \sum_{j=0}^{M-1}\sum_{l=-\infty}^{\infty} c_j(l)f_j(n - ln_j)$$

Here $\langle a(n), b(n)\rangle = \sum_n a(n)b^*(n)$ is the inner product of the sequences $a(n)$ and $b(n)$ (in the space of all sequences $x(n)$ for which $\sum_n |x(n)|^2$ is finite). Thus, the FB output $\widehat{x}(n)$ is a linear combination of the sequences from (2.2.3), using weights $c_j(l)$ that are inner products of the input $x(n)$ with the sequences from (2.2.2). Thus PR (i.e., $\widehat{x}(n) = x(n)$) is achieved if the two sets (2.2.2),(2.2.3) form a biorthogonal system, i.e., if

$$\langle \mu_{ik}(n), \eta_{jl}(n)\rangle = \delta(i-j)\delta(k-l)$$

$$\left[\begin{array}{ccc|ccc|ccc} E_0^k & \cdots & E_{n_k-1}^k & E_{n_k}^k & \cdots & E_{(p_k-1)n_k-1}^k & E_{(p_k-1)n_k}^k & \cdots & E_{L-1}^k \\ z^{-1}E_{(p_k-1)n_k}^k & \cdots & z^{-1}E_{L-1}^k & E_0^k & \cdots & E_{(p_k-2)n_k-1}^k & E_{(p_k-2)n_k}^k & \cdots & E_{(p_k-1)n_k-1}^k \\ z^{-1}E_{(p_k-2)n_k}^k & \cdots & z^{-1}E_{(p_k-1)n_k-1}^k & z^{-1}E_{(p_k-1)n_k}^k & \cdots & E_{(p_k-3)n_k-1}^k & E_{(p_k-3)n_k}^k & \cdots & E_{(p_k-2)n_k-1}^k \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ z^{-1}E_{n_k}^k & \cdots & z^{-1}E_{2n_k-1}^k & z^{-1}E_{2n_k}^k & \cdots & z^{-1}E_{L-1}^k & E_0^k & \cdots & E_{n_k-1}^k \end{array}\right]$$

Figure 5. Polyphase matrix structure for uniform FBs derived from nonuniform ones.

This can indeed be shown to be the 'time domain' equivalent of (2.2.1).

AC matrix formulation [4]. In (2.1.1), we set $M = L, W = e^{-j2\pi/L}$, and the filters as those of the uniform FB derived as in Fig. 4, from the nonuniform FB of Fig. 1. The i-th row in (2.1.1) is a sum of filter product terms $S_j(zW^i)Q_j(z)$. We group terms arising from the k-th subband in Fig. 1, i.e., those with $S_j(z) = z^{-ln_k}H_k(z)$ and $Q_j(z) = z^{ln_k}F_k(z)$ for $l = 0, 1, \ldots, p_k - 1$ where $n_k p_k = L$ (see Fig. 4). This yields a sum of form $H_k(zW^i)F_k(z)A_{ik}$, where

$$A_{ik} = \sum_{l=0}^{p_k-1} W^{-iln_k} = \sum_{l=0}^{p_k-1} e^{-j2\pi il/p_k} = \begin{cases} p_k & \text{if } i \text{ is a multiple of } p_k \\ 0 & \text{otherwise} \end{cases}$$

Thus, we can rewrite (2.1.1) using a new L–row AC matrix $\mathbf{H}(z)$ that has only M columns (one for each analysis filter of the nonuniform FB), as follows:

$$\underbrace{\begin{bmatrix} \mathbf{h}_0(z) & \ldots & \mathbf{h}_{M-1}(z) \end{bmatrix}}_{\text{AC matrix } \mathbf{H}(z)} \begin{bmatrix} F_0(z) \\ \vdots \\ F_{M-1}(z) \end{bmatrix} = \begin{bmatrix} L \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \text{where} \qquad (2.2.4)$$

$$\mathbf{h}_i(z) = p_i \begin{bmatrix} \mathbf{h}_i'(z) \\ \mathbf{h}_i'(zW^{p_i}) \\ \vdots \\ \mathbf{h}_i'(zW^{(n_i-1)p_i}) \end{bmatrix}, \text{ and } \mathbf{h}_i'(z) = \begin{bmatrix} H_i(z) & \underbrace{0 \ \ldots \ 0}_{p_i-1 \text{ zeros}} \end{bmatrix}^T \qquad (2.2.5)$$

If $n_i = M$ and $p_i = 1$ for all i (i.e., if the FB is uniform), the form of $\mathbf{H}(z)$ indeed reduces to that of (2.1.1).

Polyphase formulation. The PR condition is $\mathbf{R}(z) = \mathbf{E}^{-1}(z)$, just as for uniform FBs. However, as the equivalent uniform FB has interdependencies between the filters, its analysis polyphase matrix $\mathbf{E}(z)$ has a special structure [1]: Its rows can be partitioned into groups, where the k-th group corresponds to the k-th subband analysis filter $H_k(z)$ in Fig. 1. This group has $p_k = L/n_k$ rows as shown in Fig. 5. The first row is the L-th order analysis polyphase matrix (vector) of $H_k(z)$. Each subsequent row is formed by shifting length - n_k blocks of the previous row to the right, with the last block multiplied by z^{-1}

and circulated back to the left end.[2] These rows are the polyphase vectors of filters $z^{-an_k}H_k(z)$ for $a = 1, 2, \ldots, p_k - 1$. Similarly, the synthesis polyphase matrix $\mathbf{R}(z)$ of the equivalent uniform FB has *columns* arrangeable into groups. The k-th group has a form like the transpose of that in Fig. 5, with the $E_l^k(z)$ replaced by the entries $R_l^k(z)$ of the L-th order synthesis polyphase vector of the synthesis filter $F_k(z)$, and the z^{-1} factors replaced by z elements.

The paraunitary case. The uniform FB of Fig. 2 is said to be paraunitary (or orthonormal) if $\mathbf{E}^{-1}(z) = \widetilde{\mathbf{E}}(z)$; or in other words, if PR is obtained with $\mathbf{R}(z) = \widetilde{\mathbf{E}}(z)$, or equivalently with $Q_i(z) = \widetilde{S}_i(z)$. By generalization, the nonuniform FB of Fig. 1 is said to be orthonormal if PR is obtained (i.e., (2.2.1) is obeyed) with $F_i(z) = \widetilde{H_i}(z)$. From the relations between the filters of the nonuniform and the equivalent uniform FB, we see that each of these is paraunitary if and only if the other is. Notice that the two sets of (2.2.2),(2.2.3) which form a biorthogonal system in any PRFB, will *coincide*, hence forming an *orthonormal* system, if and only if the FB is paraunitary. This is because $F_i(z) = \widetilde{H_i}(z)$ is equivalent to $\eta_{jl}(n) = \mu_{jl}(n)$ in (2.2.2),(2.2.3). A general PRFB that is not necessarily orthonormal is often called a biorthogonal FB, due to the condition (2.2.1). Two other properties of orthonormal FBs, proved for uniform FBs in [11], are the unit energy and power complementarity properties, stated respectively as

$$\frac{1}{2\pi}\int_0^{2\pi} \left|H_i(e^{j\omega})\right|^2 d\omega = 1, \quad \text{and} \quad \sum_{i=0}^{M-1} \frac{H_i(z)\widetilde{H_i}(z)}{n_i} = 1$$

We can prove these for nonuniform FBs using the result for uniform ones and the transformation of Fig. 4.

2.3. Relation between the nonuniform and uniform PR designs

Transforming a nonuniform FB to an equivalent uniform one helps to find the PR conditions on its filters. These two FBs also share several properties (i.e., each has the property iff the other does). Examples are PR and paraunitariness; and rationality, stability, and FIR nature of filters. However, the equivalent uniform FB does not help in designing nonuniform PRFBs. This is due to its special structure: It has groups of filters that are delayed versions of each other. There are no known uniform PRFB design methods that allow imposition of this structure. Notice that the delayed versions of a filter have the *same* magnitude response, while uniform PRFB designs usually approximate ideal *nonoverlapping* analysis filter responses.

Most choices of the analysis filters $H_i(z)$ of Fig. 1 yield an equivalent uniform FB with an *invertible* analysis polyphase matrix $\mathbf{E}(z)$. However, this is

[2] The submatrix of $\mathbf{E}(z)$ shown in Fig. 5 is *block pseudocirculant* with block size $1 \times n_k$ (generalizing the notion of pseudocirculants [11]).

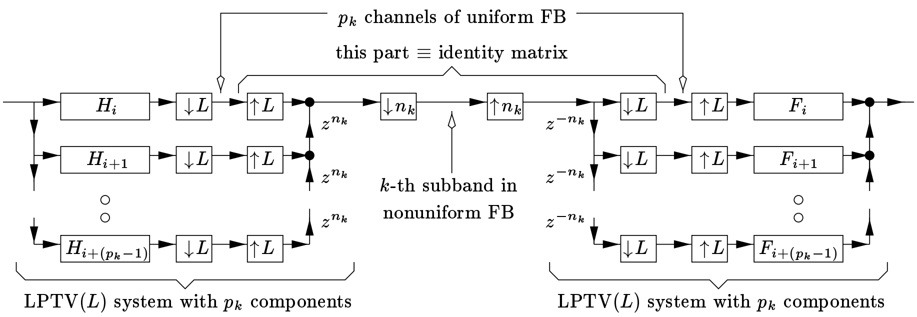


Figure 6. Equivalence between uniform FBs and nonuniform FBs with LPTV filters.

not sufficient for existence of LTI synthesis filters ($F_i(z)$ of Fig. 1) resulting in PR: For this we further require that the inverse $\mathbf{R}(z) = \mathbf{E}^{-1}(z)$ have the special structure described in Section 2.2. This added constraint is not always easy to satisfy. If $\mathbf{E}(z)$ is paraunitary, then $\mathbf{R}(z)$, being equal to $\widetilde{\mathbf{E}}(z)$, automatically has the desired structure, and a nonuniform (paraunitary) PRFB is possible. However, again none of the many known parameterizations of uniform paraunitary FBs [11] allow imposition of the special structure of Section 2.2 that $\mathbf{E}(z)$ must have in order to represent a nonuniform FB.

The structural constraints on $\mathbf{E}(z)$ and $\mathbf{R}(z)$ can however be completely given up if the filters in the nonuniform FB are allowed to be LPTV(L) instead of LTI [1]. This is shown by Fig. 6, in which $p_k = L/n_k$ channels of a uniform L-channel (maximally decimated) FB are converted into a single channel with decimator n_k. The analysis and synthesis filters in this channel are LPTV(L). The procedure is repeated for each k using disjoint subsets of channels of the uniform FB. Clearly the nonuniform FB has the PR property if and only if the uniform one does. In the rest of the paper, we assume all analysis and synthesis filters of all FBs to be LTI. The nonuniform PR design is then significantly harder.

3. Problem Statement, and Unconstrained FBs

3.1. Problem statement

The nonuniform perfect reconstruction (PR) FB design problem in its full generality can be stated as follows:

1. **Conditions on decimators for PR.** Given a set of positive integers n_i satisfying the maximal decimation condition (1.0.1), find necessary and sufficient conditions on the n_i for existence of a PRFB in some specified class $\mathcal{C}$ of FBs, having the n_i as decimators.

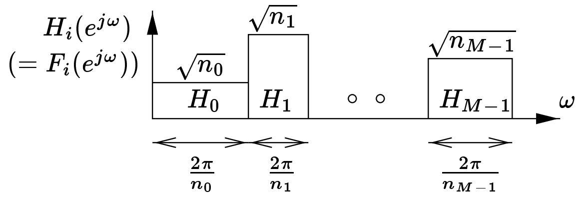


Figure 7. Ideal contiguous–stacked complex coefficient brickwall FB.

2. **Parameterization of the PRFBs.** When the n_i satisfy such a condition, find *all* possible PRFBs in $\mathcal{C}$ having n_i as decimators.

The FB class $\mathcal{C}$ here is defined by some constraint on the filters of its constituent FBs. Important examples that we will consider are delay-chains (FBs in which all filters are delays), rational FBs and FIR FBs. Other constraints that the class $\mathcal{C}$ can impose are realness of filter coefficients, stability of filters, and paraunitariness (or orthonormality). Note that in general the class definition does not directly by itself impose any constraint on either the number of channels or the nature of the decimators in the FB. However, the requirement that a FB in the class be maximally decimated and have PR could impose various constraints on these parameters. The statement of the problem is to characterize (a) the nature of these constraints, and (b) all PRFBs in $\mathcal{C}$ having a general decimator–set that obeys these constraints.

The solution to the problem of course depends on the FB class $\mathcal{C}$. It is completely known for delay-chains, but unknown for rational FBs. Notice that the parameterization problem depends on first finding conditions on the decimators for PR, which can be quite tough in itself. So we will mainly focus on finding conditions for PR. Our aim will be to weaken the sufficient conditions and strengthen necessary ones until we obtain a set of necessary *and* sufficient conditions (the final goal, which we do not always achieve). We will also derive some results on the parameterization problem, especially in connection with tree structures.

3.2. FBs with unconstrained complex and real coefficient filters

Let the class $\mathcal{C}$ in the above formulation be simply the class of all FBs, with no filter constraints (i.e., allowing ideal brickwall filters etc.). Then a PRFB in $\mathcal{C}$ always exists, no matter what the decimators n_i are (of course, provided they obey (1.0.1)). This is because the FB in Fig. 7, with ideal contiguous–stacked brickwall filters, always has PR. In fact it is a paraunitary FB. We will hence exclude this class $\mathcal{C}$ from all further discussion.

Note that the filters of Fig. 7 always have complex coefficients. Now let $\mathcal{C}$ be the class of all *real coefficient* FBs (i.e., FBs in which all filters have real

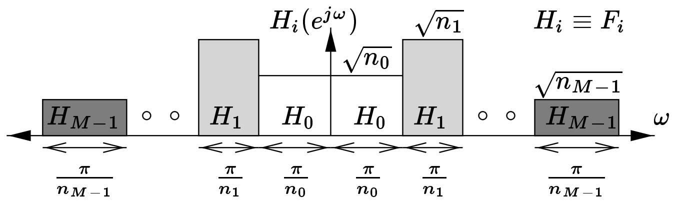


Figure 8. Ideal contiguous–stacked real coefficient brickwall FB.

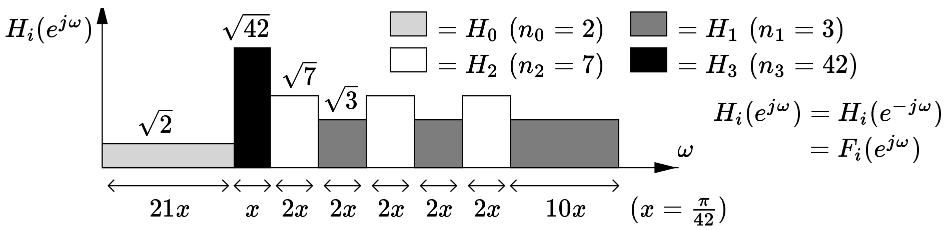


Figure 9. Non–contiguous stacked ideal real coefficient brickwall FB.

coefficients). No other constraint is imposed, so the filters could still be ideal. However, it is now more difficult to find conditions on the decimators for existence of PRFBs in $\mathcal{C}$. Taking a cue from Fig. 7, we can examine *brickwall* FBs, i.e., FBs as in Fig. 1 where the filters $H_i(e^{j\omega})$ have nonoverlapping supports, are constant on their supports and $H_i(z) = F_i(z)$. Since the H_i partition the input spectrum, PR is possible if and only if for each i, the i-th channel perfectly reconstructs all inputs that are bandlimited to the passband of $H_i(e^{j\omega})$. (In fact we then get a paraunitary PRFB, by suitable scaling of the filters.) This equivalently means that $H_i(e^{j\omega})$ has an aliasfree(n_i) support. For the (real coefficient) FB of Fig. 8, the bandpass sampling theorem states that this happens iff the band edges of H_i are at integer multiples of π/n_i [6]. Thus, the FB of Fig. 8 has PR if and only if

$$\sum_{i=0}^{k} \frac{1}{n_i} \text{ is an integer multiple of } \frac{1}{n_{k+1}} \text{ for all } k = 0, 1, \ldots, M-2. \tag{3.2.1}$$

Thus, a given set of decimators n_i can be used to build a real coefficient PRFB of the form of Fig. 8 if and only if (3.2.1) holds for some ordering of the n_i. For example, the set $\{2, 3, 6\}$ obeys this condition (with ordering (2,6,3) or (3,6,2)).

The set $\{2, 3, 7, 42\}$ violates the condition (it is the only such set with ≤ 4 decimators). However, this does not preclude existence of PRFBs with more complicated stackings of nonoverlapping real coefficient brickwall filters, e.g., as in Fig. 9. Given a set S of decimators, does such a PRFB using the set S *always* exist? Does its nonexistence imply that there is no PRFB using S with real coefficient filters (ideal or otherwise) at all? The answers are not currently known to the authors.

An important class of FBs studied in the later sections is that of all *rational* FBs, i.e., those in which all filters have rational transfer functions. As Section 6.1 will show, neither of the above decimator–sets $\{2, 3, 6\}$ and $\{2, 3, 7, 42\}$ permit existence of a rational PRFB (since they have pairs of coprime decimators). Thus it is tempting to conclude that the decimators of rational PRFBs are more restricted than those of real coefficient PRFBs. Indeed, intuition suggests that for any decimator–set S, existence of rational PRFBs using S implies that of real coefficient rational PRFBs using S. This is in fact true for all sets S for which rational PRFBs are currently known to exist. However, as we will see later, there are many sets for which it is not known whether either rational PRFBs or PRFBs with real coefficient filters (rational or otherwise) exist. Thus, in general we do not know whether existence of the former implies that of the latter. The constraint of realness of filter coefficients will not be applied or studied further in the rest of the paper.

4. Tree Structures

Cascading uniform PRFBs in a tree structure is the most common method of designing nonuniform PRFBs. As pointed out in Section 1, this method, though useful, is far from providing a complete PR theory of nonuniform FBs, i.e., a full solution to either of the two basic problems posed in Section 3.1. However, tree structures do provide very useful tools in the study of these problems. This section aims at analyzing their role in this study. Section 4.1 defines some basic terminology we will often use later in describing and studying tree structures. Section 4.2 analyzes the method of cascading uniform PRFBs in tree structures, and shows with examples how it falls short of a full PR theory of nonuniform FBs. Section 4.3 presents general methods that use trees to improve upon known PR conditions on the decimators of nonuniform FBs belonging to various FB classes. By 'improving a PR condition' we mean strengthening a necessary condition, or weakening a sufficient one. These methods will be applied to specific conditions later on.

4.1. Basics and terminology

A **tree structured FB** is one of the form shown in Fig. 10, built by repeated insertion of FBs into the subbands of other FBs. These constituent FBs of the tree structure will be called its **units**. They could be either uniform or nonuniform FBs, and may themselves be tree structured FBs. The terms **parent, child, root and leaf units** will often be used to describe the relative positions of the units in the tree; their meanings are presumed to be self–evident or clear from the examples shown in Fig. 10. We also use the term **descendant**, an

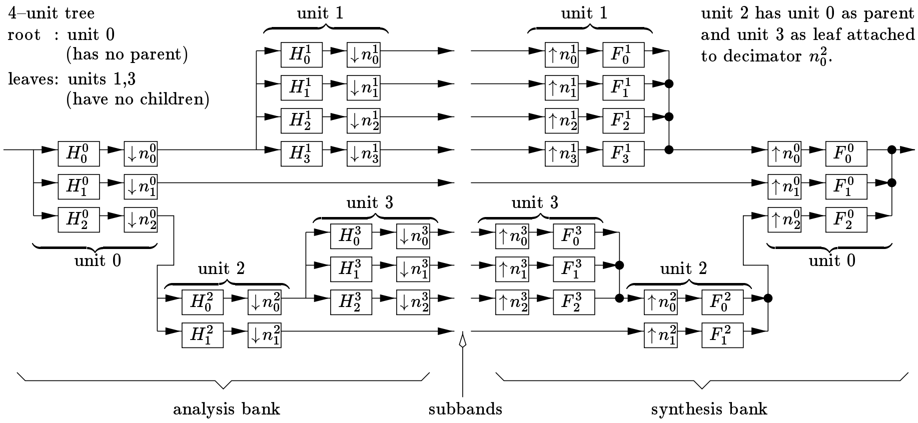


Figure 10. Tree structure of filter banks.

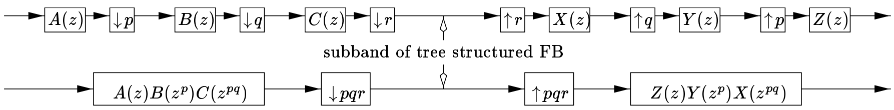


Figure 11. Structure of a channel of a tree structured FB.

obvious extension of 'child'.

Figure 11 shows how the decimators and filters of a tree structured FB are related to those of its units. The same FB may be derivable from many trees, differing in the choice of filters in the units (e.g., in Fig. 11, replace filter $A(z)$ by $A(z)/T(z^p)$ and $B(z)$ by $B(z)T(z)$ for arbitrary $T(z)$) or even in the sets of decimators in the units and the number of units (e.g., combine units 2,3 of Fig. 10 into a single FB). Every FB is derivable from a **trivial tree**, which by definition is one with only one unit, i.e., one whose root is also a leaf. We will also use the notion of a **tree structured set of decimators**. Shown in Fig. 12, this is defined exactly as a tree structured FB, except that the units of the tree are now just sets of decimators rather than FBs. The distinction is made because while a decimator–set S may sometimes be derivable from many tree structures, a FB using S need not always be derivable from all of these. In fact derivability from *all* these trees usually occurs only in very special cases (e.g., when the set S is *dyadic*, Section 7). Often we have the other extreme where the trivial tree is the only one that the FB is derivable from. Two other useful notions are as follows:

Uniform–trees. A uniform–tree structure of FBs or decimator–sets is a tree structure in which all units are uniform. (A uniform decimator–set, like a uniform FB, is one in which all decimators are equal.) Its importance, elaborated

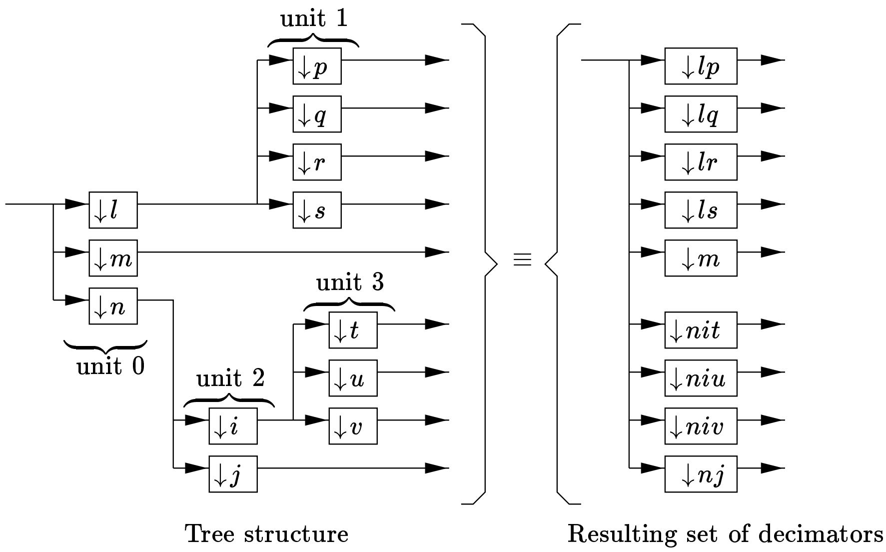


Figure 12. Tree structure of sets of decimators.

in Section 4.2, stems from the fact that design of uniform FBs is so well understood.

Properties preserved by trees. It is fairly clear that a tree structure whose units are PRFBs generates a (tree structured) PRFB. Similarly a tree of rational FBs generates a rational FB. In general we say that a property of FBs is **preserved by tree structures** if it is true that whenever all units of a tree of FBs obey the property, so does the resulting tree structured FB. A similar statement holds for properties of decimator–sets. Two obvious but important properties of FBs preserved by trees are PR and maximal decimation. From Fig. 11, we can infer that the property of having filters that are rational, stable, real coefficient, FIR, or delays, and also the paraunitariness property, are all preserved by tree structures. The property of being a uniform FB is clearly *not* preserved by trees. Other useful nontrivial examples will be presented later (Section 6.3). As Section 4.3 will show, the ability of trees to strengthen known PR conditions on the decimator–sets depends crucially on whether or not certain properties are preserved by trees.

4.2. Uniform–trees: An incomplete PR theory for nonuniform FBs

A uniform–tree of FBs or decimator–sets is one in which each unit is uniform (i.e., all its decimators are equal). Its role in the central problem of Section 3.1 can be summarized as follows:

Role of uniform–trees. Derivability of a decimator–set S from a uniform–tree is a *sufficient* condition on S for existence of PRFBs using S and belonging to the specified FB class $\mathcal{C}$, for all $\mathcal{C}$ of interest in this work.

This statement follows from the simple fact that a uniform–tree whose units are rational PRFBs generates a rational PRFB, and so on. More generally, the statement holds for every FB class $\mathcal{C}$ having two features, namely (a) $\mathcal{C}$ contains uniform PRFBs with all decimation rates, and (b) the property of being in $\mathcal{C}$ is preserved by tree structures. All $\mathcal{C}$ of interest here, e.g., the rational and FIR FB classes, have these features. Thus it is important to have an algorithm to test whether or not a given decimator–set S is derivable from a uniform–tree. Such derivability is assured, for instance, if S has no more than two *distinct* decimators, or if each decimator divides every decimator greater than itself (e.g., when they are all powers of the same integer). Appendix B proves this, and gives complete algorithms to test for derivability from uniform–trees.

Due to the common use of uniform–trees to design nonuniform PRFBs, the term 'tree structure' in the literature sometimes implicitly refers only to uniform–trees. In this work however, trees are always more general, i.e., unless explicitly referred to as uniform–trees, they could have nonuniform units too. This is necessary, for as we now show, uniform–trees do not provide a complete PR theory for nonuniform FBs.

Deficiencies of uniform–trees.

1. *Uniform–tree condition is not necessary for PR:* There are decimator–sets S that are not derivable from uniform–trees, but can be used to build PR *delay-chain* FBs, i.e., FBs in which all filters have the form z^{-k} for integer k. An example [4] is the set $S = \{6, 10, 15, 30, \ldots, 30\}$ (30 occurring 20 times), discussed in detail and generalized in Section 5.3. A delay-chain belongs to every FB class $\mathcal{C}$ of interest here (e.g., the FIR class). Thus, the uniform–tree condition is not necessary for any of these classes.

2. *Uniform–tree FBs are not a full parameterization:* Even if a decimator–set S is derivable from a uniform–tree, there may be PRFBs using S which are not derivable from any uniform–tree of FBs. We will now illustrate two examples of such a situation.

Example 1: Based on modifying filters of tree structured FBs. We take the analysis bank of a tree structured PRFB, find all subbands with a fixed decimation rate N, and transform them using an invertible square matrix $\mathbf{E}(z)$. If $\mathbf{h}(z)$ is the vector of analysis filters in the channels being transformed, the transform is equivalent to replacing $\mathbf{h}(z)$ by $\mathbf{h}'(z) = \mathbf{E}(z^N)\mathbf{h}(z)$. We preserve PR by effecting a corresponding change of synthesis filters using the inverse transform $\mathbf{E}^{-1}(z)$. Now if all the subbands being transformed come from the same unit FB in the tree, the transform can be effected by modifying only the filters of this unit, and the tree structure is preserved. More generally, if $\mathbf{E}(z)$ is block–diagonal with each block acting on subbands coming from the same unit of the tree, then again the tree structure can be preserved. However, this is no longer possible in general once we choose $\mathbf{E}(z)$ to avoid this degeneracy. In fact it is then fairly easy to ensure that the special structure of $\mathbf{h}(z)$ imposed by the tree is absent in the new filter vector $\mathbf{h}'(z)$. Thus, the new FB has the same decimator–set S but cannot be derived from the same tree. In particular if we choose a set S with a unique uniform–tree representation which is chosen as the starting tree in the above construction, the new FB is not derivable from a uniform–tree though its decimators are. An example of this kind is shown in [10], using FIR orthonormal FBs having the decimator–set $\{6, 6, 6, 6, 9, 9, 9\}$.

Example 2: Based on PR delay-chains. Consider the set with decimators 6,10,15,30 occurring 2,4,1,6 times respectively. From Section 5.2 we can show that this set can be used to build a PR delay-chain FB, which is clearly not derivable from a uniform–tree as the gcd of its decimators is unity. Now we build a tree in which the root is a uniform 2-channel PR delay-chain and the leaves are two such (identical) FBs, both children of the root. This yields a new (tree structured) PR delay-chain FB in which the gcd of the decimators is 2. Thus if this FB is to be derived from a *uniform–tree*, the root of the tree must be uniform with decimator 2. From its construction, this implies that in fact the new FB is *not* derivable from a uniform–tree. However, its set of decimators *is* derivable from a uniform–tree (in fact, in multiple ways). Note that this example cannot be produced starting from a uniform–tree of FBs in the manner used to create Example 1 above. Thus it shows a *deeper reason* for the incompleteness of FB parameterizations using uniform–trees.

4.3. Using trees to improve PR conditions on the decimators

Weakening sufficient conditions

Let S be a general decimator–set (obeying (1.0.1)). We seek conditions on S that permit existence of a PRFB that uses S as its decimator–set and belongs to some specified FB class $\mathcal{C}$. For all $\mathcal{C}$ of interest here, the most elementary but very strong *sufficient* condition on S for this purpose is that S be uniform (i.e., all its decimators be equal), as uniform FBs can always be built. However, using the fact that the FB class definition (i.e., property of being in the class $\mathcal{C}$) is preserved by tree structures, Section 4.2 has obtained a much weaker (and hence improved) sufficient condition, namely that S be derivable from a *uniform–tree*.

The process just described can be easily generalized to improve (i.e., weaken) *any* sufficient condition $\mathcal{P}$ on the decimator–set S (rather than merely the condition that S be uniform). The only requirement for this process to work is that the FB class definition be preserved by trees (which holds for all classes of interest here). The improved sufficient condition, denoted by $\mathcal{P}'$, states that S be derivable from a tree in which all units obey the original sufficient condition $\mathcal{P}$. Because S is always derivable from the trivial tree, the new condition is indeed weaker, i.e., $\mathcal{P}$ implies $\mathcal{P}'$. It is also easy to test for $\mathcal{P}'$ once we have a test for the original condition $\mathcal{P}$: We simply list all possible tree representations of S and run the test for $\mathcal{P}$ on all units of each of them. (Of course the specific nature of $\mathcal{P}$ could make even faster tests possible.)

It may happen that $\mathcal{P}' \equiv \mathcal{P}$, i.e., the 'weaker' condition is *not strictly weaker.* For example, suppose $\mathcal{P}$ itself is preserved by trees. Then if S is derivable from a tree in which each unit obeys $\mathcal{P}$ (i.e., $\mathcal{P}'$ holds), it implies that S itself obeys $\mathcal{P}$. Thus $\mathcal{P}' \equiv \mathcal{P}$ here. In fact a little further thought shows that $\mathcal{P}' \equiv \mathcal{P}$ if and only if $\mathcal{P}$ is preserved by trees. Note that by its definition, $\mathcal{P}'$ itself is preserved by trees. Thus repeated application of the above method cannot weaken the sufficient condition $\mathcal{P}$ any more than the first one does.

The only currently known instance where the above method strictly weakens a sufficient PR condition on decimator–sets is the one mentioned at the beginning of this section, leading to the uniform–tree sufficient condition. (Of course we can create other artificial instances, which lead to sufficient conditions that are stronger, and hence not as useful.) In the next section we will show a method to improve *necessary* conditions, and see that there are more nontrivial examples where this method causes a strict improvement.

Strengthening necessary conditions

We begin by illustrating the general method using a specific necessary condition that follows from Theorem 4 of Section 6.2. The condition states that the decimator–set of a rational PRFB cannot have a subset of $g+1$ decimators within which the gcd of any pair is g. The set $S = \{2, 4, 8, 12, 24\}$ can be seen to obey this condition. Suppose there is a rational PRFB using decimator–set S. We can create tree structures whose units are this and other rational PRFBs. The resulting FB also is a (tree structured) rational PRFB. Hence its decimator–set must obey the above necessary condition too. Thus we can obtain a new and stronger necessary condition on S by applying the original one to all the tree structured decimator–sets created from S as just described. In the present case, this new condition is *strictly stronger:* Using a two–unit tree in which the leaf is uniform with decimator 2 and is attached to the decimator 2 in the root S, we obtain the decimator–set $\{4, 4, 4, 8, 12, 24\}$ which violates the original condition. (Its subset $\{4, 4, 4, 8, 12\}$ has 5 decimators within which the gcd of any pair is 4.)

The new necessary condition $\mathcal{P}''$ created as above from the original condition $\mathcal{P}$ will be referred to as the **tree version** of the necessary condition $\mathcal{P}$. It is stronger, i.e., $\mathcal{P}''$ implies $\mathcal{P}$, because in particular the tree chosen in the above construction can be taken as the trivial one with S as its only unit. Generalizing the above example, we summarize the method of strengthening necessary conditions as follows:

Theorem 1: Tree versions of necessary conditions. Let $\mathcal{C}$ be a FB class such that the property of being in $\mathcal{C}$ is preserved by trees. Let $\mathcal{P}$ be a *necessary* condition on a general decimator–set S for existence of a PRFB in $\mathcal{C}$ with S as its set of decimators. Consider any decimator–set S'' derivable from a tree structure in which each unit is either identical to S or allows building of PRFBs in $\mathcal{C}$ (i.e., obeys some sufficient condition). Let $\mathcal{P}''$ be the condition that all such sets S'' satisfy $\mathcal{P}$. Then $\mathcal{P}''$ is also a necessary condition on S, called the *tree version* of the necessary condition $\mathcal{P}$.

Remarks:

1. We have just defined tree versions of necessary conditions, which are stronger necessary conditions. Earlier we had defined tree versions of sufficient conditions, which are weaker sufficient conditions. Some basic differences exist between these two methods of using trees to improve known conditions. For example, the above definition of the necessary condition $\mathcal{P}''$ involves a known sufficient condition. The weaker this sufficient condition, the stronger $\mathcal{P}''$ becomes. This is notably different from the earlier situation for tree versions of sufficient conditions.

2. *Algorithm to test* $\mathcal{P}''$*.* The condition $\mathcal{P}''$ on S demands that $\mathcal{P}$ hold for several sets S'' derived from S (including S itself) as described above. As there are infinitely many of the S'', we cannot state a general algorithm that tests for $\mathcal{P}''$. One needs specific tests designed using the features of $\mathcal{P}$ and the sufficient condition used to define $\mathcal{P}''$. This is again unlike the situation for tree versions of sufficient conditions.

3. *When are tree versions not* strictly *stronger?* Suppose $\mathcal{P}$ is preserved by tree structures. Then if S obeys $\mathcal{P}$, all units in the tree generating S'' obey $\mathcal{P}$, and hence so does S''. Thus, S obeys $\mathcal{P}''$ too, i.e., $\mathcal{P}'' \equiv \mathcal{P}$. Here too, as with tree versions of sufficient conditions, $\mathcal{P}''$ is preserved by trees, and is hence unchanged by forming its tree version. The only difference is that now we cannot in general claim that $\mathcal{P}'' \equiv \mathcal{P}$ implies that $\mathcal{P}$ is preserved by tree structures.

Tree versions of necessary conditions have not been observed earlier. A possible reason for this is that the simplest known necessary conditions for the rational FB class (compatibility and pairwise noncoprimeness, Section 6.1) are preserved by trees, and are hence identical to their tree versions. Section 6.3 shows another necessary condition that is made strictly stronger by forming its tree version.

5. Delay-chains

A delay-chain FB is one in which all filters are delays, i.e., of form z^{-k} for integer k. (We call z^{-k} a delay even though it is actually an ‘advance’ for $k < 0$.) Such a FB, while quite useless from a practical standpoint, is a useful tool in solving the problems of Section 3.1. This section presents a complete solution to these problems when the class $\mathcal{C}$ of FBs under study is that of delay-chain FBs. Because delays trivially obey various filter properties, delay-chain PRFBs belong to every class $\mathcal{C}$ studied in this paper: They are FBs with FIR, rational, stable, real coefficient filters, and we will see that they are also paraunitary. Thus, solving the problem of Section 3.1 for the class of delay-chains yields a *sufficient* condition on the decimators for existence of PRFBs in any of these classes. We will see that this condition is strictly weaker than the other sufficient condition studied earlier in Section 4.2, namely derivability from uniform–trees. In fact existence of a delay-chain is the weakest known sufficient condition for all of the earlier mentioned rational FB classes.

5.1. PR condition on the set of decimators

In Fig. 1, if $H_k(z) = z^{-l_k}$ for all k, the k-th subband signal is $c_k(n) = x(n_k n - l_k)$, i.e., it contains a certain subset of the input samples $x(n)$. Let $L = \text{lcm}\{n_i\}$, and consider any L consecutive samples of $x(n)$. We see that the k-th subband contains exactly L/n_k of these samples. Due to maximal decimation, we have

$$\sum_k (L/n_k) = L \tag{5.1.1}$$

Thus, if any of the L chosen input samples occurs in more than one subband, there must be a sample that does not occur in any subband. In this case, PR is clearly impossible no matter what the choice of synthesis filters. On the other hand, if none of the input samples occurs in more than one subband, then (5.1.1) implies that each of them occurs in exactly one subband. We can then achieve PR by appropriately interleaving the subband samples, which is done by the choice of synthesis filters as $F_k(z) = z^{l_k}$. Thus, PR is possible iff no input sample occurs in more than one subband. This condition means that if $i \neq j$, then $n_i n - l_i \neq n_j m - l_j$, i.e., $l_i - l_j \neq n_i n - n_j m$, for any integers n, m. As n, m range over all integers, the right side here ranges over all multiples of $\text{gcd}(n_i, n_j)$. Thus the PR condition may be summarized as follows:

Theorem 2: Delay-chain PRFBs. In Fig. 1, if $H_k(z) = z^{-l_k}$ for integers l_k, PR is possible iff no input sample occurs in more than one subband. Under this condition, PR is obtained with the unique choice $F_k(z) = \widetilde{H}_k(z) = z^{l_k}$, yielding a PR delay-chain FB, which is thus always paraunitary. The necessary and sufficient condition on the decimators n_i for existence of such a FB is that there exist integers l_i satisfying

$$(l_i - l_j) \not\equiv 0 \pmod{\text{gcd}(n_i, n_j)} \quad \text{if } i \neq j \tag{5.1.2}$$

5.2. Testing the PR condition

Given the decimators n_k, it is required to test for existence of integers $l_0, \ldots, l_{M-1}$ obeying (5.1.2). Now if (5.1.2) holds for some integers l_k, then it also holds if each l_k is replaced by $l_k + m_k n_k + C$ for any integers m_k and any fixed integer C. Hence, without loss of generality we can assume that $0 \leq l_k < n_k$ and $l_0 = 0$. This makes the number of possible sets of l_k *finite*, so clearly there is an algorithm for our purpose.

For example, we can try to assign the l_k sequentially, as follows: Suppose we have $l_0, l_1, \ldots, l_{N-1}$ obeying (5.1.2) for some $N < M$. We assign to l_N all possible values obeying $0 \leq l_N < n_N$ and $(l_N - l_j) \not\equiv 0 \pmod{\gcd(n_N, n_j)}$ for $j = 0, 1, \ldots, N-1$. Each value yields a larger set $l_0, l_1, \ldots, l_N$ obeying (5.1.2), and we can now repeat the process on this set. If there is no valid choice for l_N, we must restart with another valid set of choices for $l_0, l_1, \ldots, l_{N-1}$. Initializing this recursive scheme using $l_0 = 0$, we can thus list *all* sets $\{l_0, l_1, \ldots, l_{M-1}\}$ obeying (5.1.2). In particular this finds whether or not there exist such sets. This solves both problems of Section 3.1 for the class $\mathcal{C}$ of delay-chain FBs. To determine only the existence of a PR delay-chain, the above algorithm can often be accelerated using the following result:

Fact 1. Let integers $l_0, \ldots, l_{N-1}$ obey (5.1.2) for some $N < M$, and let n_N be a common multiple of $n_0, n_1, \ldots, n_{N-1}$. Then there is an integer l_N such that $0 \leq l_N < n_N$ and $l_0, \ldots, l_{N-1}, l_N$ obey (5.1.2) too.

Proof: From the premise, $l_0, \ldots, l_{N-1}, l_N$ will satisfy (5.1.2) if and only if

$$(l_N - l_j) \not\equiv 0 \pmod{\gcd(n_N, n_j)} \quad \text{for } j = 0, 1, \ldots, N-1. \tag{5.2.1}$$

Also $\gcd(n_N, n_j) = n_j$. Thus (5.2.1) is equivalent to $l_N \neq l_j + n n_j$ for all integers n, for $j = 0, 1, \ldots, N-1$. For each j there are n_N/n_j integers of this form $l_j + n n_j$ in the range $[0, n_N)$. Thus, of the n_N integers in $[0, n_N)$, at most $B = \sum_{k=0}^{N-1} n_N/n_k$ of them are excluded as possible choices of l_N (in fact we can even show that exactly B choices are excluded). As $N < M$, maximal decimation (1.0.1) means that $B < n_N$, so there are still valid choices of l_N in the interval $[0, n_N)$. $\triangledown\triangledown\triangledown$

Thus, suppose there is a decimator n_N such that each $n_j \geq n_N$ is a multiple of all $n_i < n_j$. It then suffices to verify existence of valid delays l_k obeying (5.1.2) for all $n_k < n_N$. As an extreme case, if every n_j is a multiple of all $n_i < n_j$ (i.e., every n_j divides all $n_i > n_j$), then a delay-chain PRFB always exists. In fact the decimator–set is then derivable from a uniform–tree (Appendix B). Fact 1 is also useful in proving Theorem 3 which follows soon.

Nonuniqueness of delay-chains: When a decimator–set allows building of a PR delay-chain FB, in general this delay-chain is not unique. The non-uniqueness can be much deeper than that caused simply by adding a constant delay to all the filters. For example, when several delay-chains are possible, it could happen that some of them are also derivable from uniform–trees, while some others are not, as seen in Section 4.2.

5.3. Delay-chains vs. uniform–trees

Our study of tree structures showed that (a) known PR conditions on decimators can sometimes be strengthened using trees (Section 4.3), and (b) derivability of the decimators from a uniform–tree is a *sufficient* PR condition for all FB classes that we study (Section 4.2). Does this teach us more about delay-chains? Firstly, the condition (5.1.2) is both necessary and sufficient for existence of PR delay-chains. Hence it remains unaltered by the procedures of Section 4.3. Next, the uniform–tree condition is *not* necessary, as we now show:

Theorem 3: PR delay-chains without uniform–trees. There are infinitely many PR delay-chain FBs that cannot be derived from uniform–trees. Such FBs can be built using every set of decimators of the form

$$S = \{n_0, n_1, n_2, L, L, \ldots, L\}, \quad \text{where} \quad L = \mathrm{lcm}(n_0, n_1, n_2),$$

and $n_0 = m_1 m_2$, $n_1 = m_2 m_0$, $n_2 = m_0 m_1$ where the m_i are pairwise coprime integers greater than unity. (Here L occurs $L(1 - \sum_{i=0}^{2}(1/n_i))$ times in S.)

Proof: By Fact 1, decimators of S allow building of a PR delay-chain FB iff there are integers l_0, l_1, l_2 obeying (5.1.2) for $i, j \in \{0, 1, 2\}$. This condition is easily ensured, in fact it holds iff $\gcd(n_i, n_j) \geq 2$ for $i, j \in \{0, 1, 2\}$ with strict inequality for at least one $i \neq j$. (We can then make a valid choice of the l_i from the numbers 0,1,2.) Further if $\gcd(n_0, n_1, n_2) = 1$, the set cannot be derived from a uniform–tree (Appendix B). Both these requirements are satisfied by the choice of n_i stated by the theorem. $\bigtriangledown \bigtriangledown \bigtriangledown$

An example of a delay-chain PRFB not derivable from a uniform–tree was first shown in [4]. Its set of decimators $\{6, 10, 15, 30, 30, \ldots, 30\}$ (30 occurring 20 times) is a special case of the construction of Theorem 3 with $(m_0, m_1, m_2) = (5, 3, 2)$. This is not the only way to produce such examples: Delay-chain PRFBs can also be built with the decimator values $6, 10, 15, 30$ when the number of their respective occurrences are $2, 4, 1, 6$ or $2, 3, 2, 7$. The former set of decimators is the *smallest* such example.[3] It can be used as the root of a tree to derive the example of [4], but not the latter example. In all these cases, the decimators have no common factor, ensuring that they are not derivable from uniform–trees. In fact if the decimators of a delay-chain PRFB do have a common factor, the FB can be built from smaller PR delay-chains as follows:

Fact 2. Let all decimators in a PR delay-chain FB have common factor $K > 1$. Then the FB can be derived from a tree structure in which each unit is a PR delay-chain FB and the root is uniform with decimator K.

[3]This is true when size is measured by either the number of decimators, or their lcm, or the largest one. In fact there is no other example with 13 or fewer decimators. This is verifiable by exhaustive search aided by a computer and Fact 2.

Proof: Let $x(n)$ be the FB input. For $0 \leq k < K$, let $f_k(n) = x(Kn - k)$, which is the k-th subband signal in a uniform K channel delay-chain PRFB. Now consider the i-th channel of the given PR delay-chain, with decimator n_i, analysis filter z^{-l_i}, and hence, subband signal $x(n_i n - l_i)$. Since n_i is a multiple of K, either *all* its samples lie in the sequence $f_k(n)$, or none of them do (depending on whether or not $l_i \equiv k \pmod{K}$). We now collect all subbands whose samples do lie (entirely) in $f_k(n)$. Due to the PR condition for delay-chains (Theorem 2), these subbands collectively contain all samples of $f_k(n)$ (as none of the other subbands have any of them), and each of these samples occurs in exactly one of these subbands. Further the delays in all these subbands are equal (to k) modulo K. Thus these subbands can be generated by inserting a suitable delay-chain PRFB as a child (in a tree) in the k-th subband signal $f_k(n)$ of a uniform K channel delay-chain PRFB. Repeating this process for $k = 0, 1, \ldots, K-1$ yields the desired tree structure. $\bigtriangledown\bigtriangledown\bigtriangledown$

Remarks:

1. The above result does not generalize easily to other classes of FBs (besides delay-chains). For example, consider the decimators $\{4, 4, 4, 4\}$, having common factor $K = 2$. These decimators can be used to build *rational* and *FIR* PRFBs that are not derivable from *any* tree structure (besides the trivial one).

2. A common factor $K > 1$ among all decimators does not by itself ensure their derivability from a tree whose root is uniform with K as decimator.[4] However, if the decimators also allow building of a delay-chain PRFB, then by Fact 2, there is at least one such tree, as the FB itself is derivable from such a tree.

3. All decimators of a PR delay-chain FB need not have a common factor $K > 1$ (see the example in Theorem 3). However, further conditions on the decimators can force such a common factor to exist, thus making Fact 2 apply. For example, suppose the PR delay-chain has a decimator of value m occurring $m - 1$ times (m is thus the smallest decimator). Then all decimators must have m as a factor. This is provable by a slight extension of the proof of Fact 2. In fact it even generalizes to rational FBs in place of delay-chains (Theorem 5, Section 7), although this is harder to prove.

6. The Class of Rational FBs

In this section and most of Section 7, the FB class $\mathcal{C}$ of interest is that of rational FBs, i.e., FBs in which all filters are rational. We seek necessary and sufficient conditions on a decimator–set S for existence of a rational PRFB using S.

[4]The set of decimators $\{4, 6, 6, 10, 10, 10, 10, 60\}$ shows this for $K = 2$. The choice of root prevents the leaves from obeying (1.0.1).

The *weakest known sufficient condition* is that of existence of a PR *delay-chain* (Section 5). This is clearly sufficient since delay-chains are rational FBs, but *is it also necessary*? Or is there a decimator–set which does not permit existence of PR delay-chains, but allows building of rational PRFBs (whose filters are not all delays)? This is a major open question in the PR theory of nonuniform FBs.

A possible approach to answer the above question is to try to build a rational PRFB with decimators that do not allow building of PR delay-chains. However, starting with an arbitrary decimator–set such as $S = \{2, 3, 6\}$ does not help, as S violates a known *necessary* condition (called 'compatibility', Section 6.1) on the decimators of a rational PRFB. Such sets must be excluded, and to this end it helps to derive more necessary conditions. This is our main contribution in this section. The previously known necessary conditions for PR are described in Section 6.1. Each subsequent subsection develops a new necessary condition that is strictly stronger than a previously known one. Table 1 (Section 8) presents a comprehensive summary of all known conditions, many of which are new results of the present work. The table studies the interrelationship between the conditions, and lists example decimator–sets illustrating their use.

All the new necessary conditions we develop still collectively remain insufficient for existence of *delay-chain* PRFBs, and thus it is *still not known* whether they are sufficient for existence of rational PRFBs. Our work reduces the 'gap' between the necessary conditions and the sufficient one. Proving that the sufficient condition is in fact necessary would in some sense render obsolete most of the present section. However this appears tough to do, in fact the statement may not even be true. Our work is a step towards the truth.

6.1. Previously known necessary conditions on decimators

1. *Pairwise noncoprimeness.* No two decimators of a rational PRFB can be coprime [4]. If $\gcd(n_i, n_j) = 1$ for two decimators n_i, n_j in Fig. 1, the biorthogonality condition (2.2.1) for PR implies $H_iF_j = 0$ and $(H_iF_i)\downarrow_{n_i} = (H_jF_j)\downarrow_{n_j} = 1$. This is impossible for a rational FB, as $H_iF_j = 0$ forces $H_i \equiv 0$ or $F_j \equiv 0$.

2. *Compatibility.* Every decimator occurring only once must divide some other decimator [1, 5, 4]. In particular, the largest decimator must occur at least twice. As Section 6.4 will show, without this condition the rational FB cannot even be a nonzero LTI system, let alone have PR.

3. *Strong compatibility.* This condition, developed in [4], places a lower bound $b_j \geq 1$ on the number of occurrences N_j of each decimator n_j. The condition is stated as follows:

$$N_j \geq b_j \stackrel{\triangle}{=} \frac{1}{p_j}\left(\min_{p_i \neq p_j} \operatorname{lcm}(p_i, p_j)\right), \quad \text{where} \quad p_i = \frac{L}{n_i}, \tag{6.1.1}$$

where $L = K(\mathrm{lcm}\{n_i\})$ for any integer $K > 0$. This will be shown in Section 6.4, which in fact proves a new condition strictly stronger than the above.

Note that the bound b_j of (6.1.1) is independent of the integer K. Also, it only needs verification for *distinct* decimator values, because if $n_i = n_j$ then $N_i = N_j$, $b_i = b_j$. For a *uniform* set of decimators, $p_i = p_j$ for all i, j, so we define $b_j = 1$ here (so that the bound holds). Excluding this case, $b_j = 1$ iff p_j is a multiple of some $p_i \neq p_j$, i.e., iff n_j divides some distinct decimator n_i. So the bound need not be checked for such decimators. Also, strong compatibility implies compatibility, because it demands that any n_j occurring only once (i.e., with $N_j = 1$) must have $b_j = 1$, i.e., must divide some other decimator. In fact strong compatibility is a *strictly* stronger necessary condition than compatibility, as shown by the set of decimators $\{2, 4, 6, 24, 24\}$. However it does not imply pairwise noncoprimeness [1] (shown by $\{2, 5, 10, 10, 10\}$). Likewise, a set could satisfy pairwise noncoprimeness but violate compatibility (and hence strong compatibility), e.g., $\{2, 4, 6, 12\}$.

6.2. The pairwise gcd test

Theorem 4: Pairwise gcd test. Among the decimators of a rational PRFB, there cannot be a subset of $g + 1$ decimators such that the gcd of any two elements from the subset is a factor of g. In particular (for $g = 1$), this implies the pairwise noncoprimeness condition (Section 6.1).

Proof: As with pairwise noncoprimeness, the proof uses the biorthogonality condition (2.2.1) for PR. Let $g + 1$ decimators $n_0, n_1, \ldots, n_g$ be such that the gcd of any pair divides g. From (2.2.1), $(H_i(z)F_j(z)) \downarrow_{\gcd(n_i,n_j)} = 0$ if $i, j \in \{0, 1, \ldots, g\}$, $i \neq j$. In this case $g/\gcd(n_i, n_j)$ is given to be an integer, so decimating both sides by it,

$$(H_i(z)F_j(z)) \downarrow_g = 0, \quad \text{for } i, j \in \{0, 1, \ldots, g\},\, i \neq j \tag{6.2.1}$$

Form the g-th order analysis polyphase matrix $\mathbf{E}(z)$ (of size $(g + 1) \times g$) of the filters $H_i(z)$, and the g-th order synthesis polyphase matrix $\mathbf{R}(z)$ (of size $g \times (g + 1)$) of the F_i. Thus, from the polyphase lemma (Section 1.3), for $i, j \in \{0, 1, \ldots, g\}$, $(H_i(z)F_j(z)) \downarrow_g$ is the ij-th entry in the product $\mathbf{P}(z) = \mathbf{E}(z)\mathbf{R}(z)$. Hence by (6.2.1), $\mathbf{P}(z)$ is a $((g + 1) \times (g + 1))$ *diagonal* matrix. Its i-th diagonal entry is the filter $(H_i(z)F_i(z)) \downarrow_g$, with impulse response $c_i(gn)$, where $c_i(n)$ is the impulse response of H_iF_i. From (2.2.1), $(H_i(z)F_i(z)) \downarrow_{n_i} = 1$, so $c_i(n_in) = \delta(n)$ i.e., $c_i(0) = 1$. Hence $c_i(gn) \not\equiv 0$, i.e., no diagonal element of $\mathbf{P}(z)$ is identically zero. Thus, as these elements are rational filters, there is a z such that $\mathbf{P}(z)$ has full rank $g + 1$. However, this is impossible from the sizes of $\mathbf{E}(z), \mathbf{R}(z)$. $\triangledown\triangledown\triangledown$

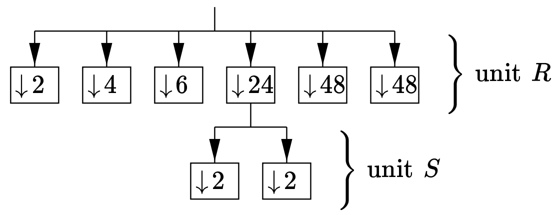


Figure 13. Showing that strong compatibility is not preserved by trees.

6.3. Tree version of strong compatibility

In Section 4.3, we saw how given a necessary condition $\mathcal{P}$ on the decimators for PR, we could form its 'tree version' $\mathcal{P}''$, which is a stronger (though not necessarily *strictly* stronger) necessary condition. We can apply this process to the conditions of Section 6.1. Some thought shows that both the pairwise non-coprimeness and the compatibility conditions are preserved by tree structures, and are hence identical to their tree versions (as seen in Section 4.3). However, the same is not true with strong compatibility: Its tree version is strictly stronger than itself. This is shown by the two–unit tree in Fig. 13. Both units R and S are strong compatible, and S allows building of rational PRFBs (as it is uniform). However the resulting set of decimators is not strong compatible. Hence, though R obeys strong compatibility, it violates its tree version.

A complete algorithm to test this new necessary condition is described in Appendix E. Its derivation involves characterizing trees similar to that in Fig. 13. This is done by the following results:

Fact 3. Consider a set T of decimators derived from a 2-unit tree structure having root R and leaf S attached to decimator m_0 of R. Suppose R, S are strong compatible but T is not. Then S is a uniform unit, i.e., all its decimators have equal value K. The decimator m_0 of R does not occur in T, i.e., it occurs only once in R. The decimator m_0K of T obtained at the leaf S also occurs in R. Decimators of this value m_0K are the only ones violating the strong compatibility lower bound on the number of their occurrences in T.

Fact 4. Let a set D of decimators satisfy strong compatibility but violate its tree version. Then there is a tree $\mathcal{T}$ generating a set T of decimators, such that $\mathcal{T}$ and T have the following properties:

1. The tree $\mathcal{T}$ has root D. All leaves of the tree are uniform and are children of its root. All decimators obtained at the leaves have equal value d.

2. If d_i are the decimators of D to which leaves are attached in $\mathcal{T}$, then no decimator in T has value d_i.

3. If $d \notin D$, then $d = \mathrm{lcm}\{d_i\}$. Hence, if $d \notin D$, the d_i are not all equal (for otherwise, $d = d_i \in D$).

4. Decimator $d \in T$ violates the strong compatibility lower bound on the number of its occurrences in T.

Fact 3 is proved in Appendix C and used to prove Fact 4 in Appendix D. Fact 4 gives an algorithm to test whether the set D obeys the tree version of the strong compatibility condition: We find all trees with root D and properties 1,2 and 3 listed in Fact 4. It can be seen that there are only finitely many such trees, and from Fact 4, D violates the condition if and only if one of these trees also obeys property 4. This idea is the basis of the detailed algorithm of Appendix E.

6.4. The AC-matrix test

The necessary condition derived here relies heavily on the AC matrix formulation (2.2.4),(2.2.5) of the PR condition on the filters of the FB. The algorithm to test the condition is described in Appendix F, and may be taken as the statement of the condition (i.e., this condition, unlike the earlier ones, does not have a short / simple statement). Like the test of Section 6.3, this test also strictly strengthens strong compatibility, but in an independent direction. In this section we derive two lemmas that explain the operation of the test, illustrate the test with examples, and thus justify the algorithm of Appendix F. Deriving the new test also *proves* the necessity of strong compatibility for PR; a result *assumed* in deriving the test of Section 6.3. We further show that (simple) compatibility is necessary even if we allow the rational FB to violate PR but merely insist that it be an *LTI* system (i.e., an *aliasfree* FB) that is not identically zero.

Two key results used by the test

Lemma 2. In Fig. 1, if $\sum_k H_{i_k}(z)F_{i_k}(z) = 0$ for any set of i_k, $0 \leq i_k < M$, then the FB cannot have PR.

Proof: If the FB of Fig. 1 has PR, $(H_{i_k}(z)F_{i_k}(z)) \downarrow_{n_{i_k}} = 1$ by biorthogonality (2.2.1). Let $L = \mathrm{lcm}\{n_i\}$, thus $(H_{i_k}(z)F_{i_k}(z)) \downarrow_L = (1) \downarrow_{(L/n_{i_k})} = 1$. So $(\sum_k H_{i_k}(z)F_{i_k}(z)) \downarrow_L \neq 0$, violating $\sum_k H_{i_k}(z)F_{i_k}(z) = 0$. $\bigtriangledown\bigtriangledown\bigtriangledown$

Lemma 3. Given *rational* filters $B_i(z), C_i(z)$, $0 \leq i < N$, let $W = e^{-j2\pi/M}$ and $G_l(z) = \sum_{i=0}^{N-1} B_i(zW^l)C_i(z)$. If $G_l(z) = 0$ for N values of l occurring consecutively in an arithmetic progression, then $G_l(z) = 0$ *for all* values of l in this progression. (The lemma in fact holds for any nonzero complex W.)

Proof: If $N = 1$, the lemma must be taken to mean that if $B_0(zW^l)C_0(z) \equiv 0$ for some l, then it holds for all l. This is clearly true: Rational filters $B_0(z), C_0(z)$ obey $B_0(zW^l)C_0(z) \equiv 0$ iff $B_0 \equiv 0$ or $C_0 \equiv 0$ or both. (Note however that this is in general false if we remove the rationality constraint.) Hence, let

$N > 1$. Let the N given consecutive values of l in arithmetic progression be $s, s+d, s+2d, \ldots, s+(N-1)d$. The lemma can then be restated by defining

$$\mathbf{b}(z) = [B_0(zW^s), \ldots, B_{N-1}(zW^s)], \quad \mathbf{c}(z) = [C_0(z), \ldots, C_{N-1}(z)]^T,$$

as follows: If $\mathbf{b}(zW^{nd})\mathbf{c}(z) = 0$ for $n = 0, 1, \ldots, N-1$ then it is true for all integers n. To show this, form the square matrix $\mathbf{B}(z)$ with rows $\mathbf{b}(zW^{nd})$, $0 \leq n < N$. By the premise of the lemma, $\mathbf{B}(z)\mathbf{c}(z)$ is the zero vector. This implies linear dependence of the columns of $\mathbf{B}(z)$, and hence of its rows, as it is square. So $\sum_{n=0}^{N-1} \alpha_n(z)\mathbf{b}(zW^{nd}) = 0$ for some rational filters $\alpha_n(z)$ not all identically zero. Let r be the maximum n for which $\alpha_n(z) \not\equiv 0$. Divide the above relation by $\alpha_r(z)$. (This is allowed solely due to the rationality assumption: Otherwise $\alpha_r(e^{j\omega})$ could for instance be zero in an interval.) This yields

$$\mathbf{b}(zW^{rd}) = \sum_{n=0}^{r-1} \beta_n(z)\mathbf{b}(zW^{nd}) \qquad \text{for some rational filters } \beta_n(z).$$

Replacing z by zW^d and postmultiplying by $\mathbf{c}(z)$ shows $\mathbf{b}(zW^{nd})\mathbf{c}(z) = 0$ for $n = r+1$. Using this and repeating the process shows the same for $n = r+2$. Carrying on this way, the result is shown for all $n \geq 0$. For $n < 0$ we use a similar process, now taking r as the *minimum* n obeying $\alpha_n(z) \not\equiv 0$. $\bigtriangledown\bigtriangledown\bigtriangledown$

Deriving the test from the AC matrix and Lemmas 2, 3

Examine closely Equations (2.2.4),(2.2.5) which are equivalent to PR. Number the rows of the AC matrix from 0 to $L-1$, and the columns from 0 to $M-1$, and let $W = e^{-j2\pi/L}$. The l-th row equation in (2.2.4) has form

$$\sum_k p_{i_k} H_{i_k}(zW^l)F_{i_k}(z) = 0 \tag{6.4.1}$$

The summation here ranges over all indices i_k for which the i_k-th column in the AC matrix has a nonzero entry in the l-th row. This happens if and only if l is a multiple of $p_{i_k} = L/n_{i_k}$. Thus, suppose that for all integers m in some set S, the number $l = mp_0$ is not a multiple of any $p_i \neq p_0$. For all these $l = mp_0$, (6.4.1) holds with the summation being over the same set of filters, i.e., those corresponding to decimator value n_0. Thus, if N_0 is the number of occurrences of decimator n_0, (6.4.1) takes the form

$$\sum_{i=0}^{N_0-1} p_0 H_i(zW^l)F_i(z) = 0, \quad \text{for } l = mp_0, \quad \text{for } m \in S \tag{6.4.2}$$

This is very similar to the system $\sum_{i=0}^{N-1} B_i(zW^l)C_i(z) = 0$ of Lemma 3, with $N = N_0$. The only difference is that here the premise of the lemma may or may not hold, i.e., (6.4.2) may or may not hold for N_0 values of l occurring consecutively in an arithmetic progression.

The main idea of the test we are developing is to find all progressions for which the premise of Lemma 3 actually holds, and then use the lemma. This may sometimes allow us to deduce that (6.4.2) actually holds for other values of l too, besides those stated in (6.4.2) itself. If $l = 0$ turns out to be one of these, then by Lemma 2 we can conclude that PR is impossible, i.e., the given set of decimators fails the test. To perform such a test, we must use the known values of l from (6.4.2) to find progressions obeying the premise of Lemma 3. From (6.4.2) it clearly suffices to examine progressions of integers whose common difference d is an integer multiple of p_0. There are infinitely many such progressions, each an infinite sequence. However, since $W^L = 1$, it suffices to consider the progressions modulo L, and to restrict their common difference d as $d < L$. In fact even $d \leq \lfloor L/2 \rfloor$ suffices, as any progression with common difference $L - d$ can be generated in reverse order by one with difference d. We will now show examples of the working of the above test.

Example 1: Compatibility is a special case, and is necessary even for aliasfree FBs. Suppose p_0 occurs only once and is not a multiple of any $p_i \neq p_0$. In other words, n_0 occurs only once and does not divide any other n_i, i.e., compatibility (Section 6.1) is violated. Then (6.4.2) holds with $N_0 = m = 1$, and a trivial use of Lemma 3 shows that indeed (6.4.2) also holds for $l = 0$ (i.e., $H_0 F_0 \equiv 0$). Thus by Lemma 2, PR is impossible. Hence, passing our new test implies compatibility of the decimators. In fact, even if the rational FB does not have PR but is *LTI* (i.e., aliasfree) with transfer function $T(z)$, (2.2.4),(2.2.5) still hold with L replaced by $LT(z)$ in (2.2.4). Hence, if n_0 violates compatibility as described above, the conclusion $H_0 F_0 \equiv 0$ still holds. Thus the FB input–output relation is preserved even if we drop the 0-th channel, making the FB *overdecimated*. As the input of such a FB cannot be recovered from its output using any LTI system, we must have $T(z) \equiv 0$ (else we could use the LTI inverse $1/T(z)$). Thus, compatibility is necessary even if all we demand of the rational FB is that it be aliasfree and not identically zero $T(z) \not\equiv 0$ (as opposed to having PR).

Example 2: A specific set of decimators. Consider the set $\{4, 6, 6, 6, 10, 20, 20, 20\}$. This has 8 decimators with lcm $L = 60$, and $p_5 = p_6 = p_7 = 60/20 = 3$. For $m = 7, 9, 11$, the numbers $l = 3m$ are not multiples of any $p_i \neq 3$. Thus for these l, the l-th row equation in (2.2.4) reads as

$$\sum_{i=5}^{7} 3H_i(zW^l)F_i(z) = 0 \quad \text{for } l = 3m,\ m = 7, 9, 11 \qquad (6.4.3)$$

The sum has three terms, and the three values of l occur consecutively in an arithmetic progression. Thus by Lemma 3, (6.4.3) holds for all l in this progression, specifically for $l = 3 \times 5 = 15$, which is not a multiple of any p_i

besides $p_5 = 3$ and $p_0 = 15$. The 15-th row equation in (2.2.4) initially reads as $15H_0(zW^l)F_0(z)+3\sum_{i=5}^{7} H_i(zW^l)F_i(z) = 0$ (for $l = 15$), but in the light of the above conclusion it now further says that $H_0(zW^l)F_0(z) = 0$. Now another application of Lemma 3 (for the trivial case of $N = 1$) shows that $H_0(z)F_0(z) = 0$ and thus PR is impossible by Lemma 2.

Based on the above discussion and examples, Appendix F shows a complete general algorithm to test the necessary condition derived above using Lemmas 2 and 3. This test, called the AC matrix test, implies not merely compatibility (Example 1) but strong compatibility too (Appendix F), and is in fact strictly stronger than strong compatibility: The decimators of Example 2 are strong compatible and yet fail the test.

7. Conditions Based on Reductions to Tree Structures

As seen in Section 4.2, a decimator–set may be derivable from trees in many ways, but a FB using the decimator–set may not be derivable from all of these trees. However, all PRFBs using decimators obeying certain conditions must be derivable from certain nontrivial trees associated with the conditions. Fact 2 is a result of this type for delay-chains. Another example is as follows: If the decimator–set comes from a dyadic or 'wavelet' tree (i.e., has form $\{2, 2^2, \ldots, 2^{r-1}, 2^r, 2^r\}$ for some integer $r > 0$), then *all* PRFBs using those decimators are derivable from this tree. This was proved in [10],[3] for rational orthonormal and biorthogonal FBs respectively. It parameterizes all FBs with dyadic decimator–sets, i.e., solves problem 2 of Section 3.1 for such sets. However, it does not reveal any new conditions on decimators for existence of rational PRFBs (problem 1 of Section 3.1). This is because it concerns only dyadic decimator–sets, which, being derivable from uniform–trees, are already known to allow building of PRFBs in every FB class of interest here.

Suppose on the other hand that we have a condition on a more general decimator–set S that allows us to conclude that every rational PRFB using S is derivable from some (nontrivial) tree. Such a condition provides a parameterization result for FBs using such decimator–sets S. Further, it reduces the problem of existence of rational PRFBs using S to that of existence of rational PRFBs using the smaller decimator–sets in the units of the tree. We can obtain a new necessary condition on S for existence of such a FB, by applying all the known conditions on these smaller sets. In this section, we derive three such conditions (Theorems 5,6,7) all of which yield as a special case, the result on dyadic FBs mentioned earlier. We refer the reader to Table 1 (Section 8) for example decimator–sets showing the use of the new necessary conditions generated by these results. Finally, we present two other results (Theorems 8,9) that also pertain to other filter constraints besides rationality such as orthonormality, stability and the FIR property.

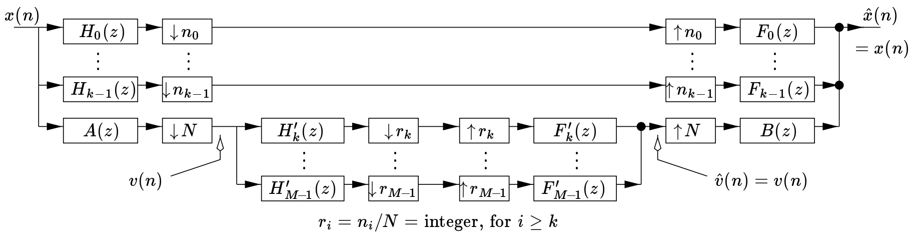


Figure 14. Root extraction test (Theorem 5): Showing equivalent tree structure for any rational PRFB with decimators obeying the premise (7.0.1) of the test.

Theorem 5: Root extraction test. Let a set of decimators $n_0, n_1, \ldots, n_{M-1}$ obey

$$\sum_{i=k}^{M-1} \frac{1}{n_i} = \frac{1}{N}, \quad \text{where } N = \text{integer multiple of } n_0, n_1, \ldots, n_{k-1}. \tag{7.0.1}$$

Let there be a rational PRFB using these decimators. Then all n_i for $i \geq k$ are multiples of N, and the FB is always derivable from a two unit tree structure of rational PRFBs in which the root has decimators $n_0, n_1, \ldots, n_{k-1}, N$, as shown in Fig. 14.

This result is a special case of Theorem 6 (which is proved in Appendix G). A corollary obtained with $N = 2$ is that any rational PRFB having a decimator of value 2 must be derivable from a two unit tree of rational PRFBs in which the root is a uniform two band FB. Repeated use of this corollary shows the result of [3] on derivability of rational biorthogonal FBs with dyadic decimators from trees. The corresponding result of [10] for orthonormal FBs does not directly follow: Theorems 5,6 do not themselves show how to ensure orthonormality of all the units of the tree, given that of the overall FB. This can be done using Theorem 8 which follows later. Finally, note that even nonrational PRFBs with decimators n_i obeying (7.0.1) must be derivable from trees as in Fig. 14, *provided* the n_i have the property that all n_i for $i \geq k$ are multiples of N. In other words, this property no longer follows from (7.0.1) (as is clear from Section 3.2), but the derivability from a tree follows if the property is made an additional premise (Appendix G).

Theorem 6: Generalized root extraction test. Let the given decimator–set

$D=\{n_0,\ldots,n_{M-1}\}$ have disjoint[5] subsets S, T_1, T_2 such that S is nonempty and

$$\sum_{n_i \in S} \frac{1}{n_i} = \frac{1}{N} \text{ for some integer } N, \tag{7.0.2}$$

$$T_1 = \{n_i : n_i \in D, n_i = \text{factor of } N\}, \tag{7.0.3}$$

$$\gcd(n_i, n_j) = \text{factor of } N \quad \text{whenever } n_i \in S \cup T_2, n_j \in T_2, i \neq j, \tag{7.0.4}$$

$$\sum_{n_i \in T_1} \frac{N}{n_i} + |T_2| = N - 1 \quad (\text{where } |T_2| = \text{number of elements in } T_2). \tag{7.0.5}$$

Then, if a rational PRFB exists with these decimators, all $n_i \in S$ are multiples of N, and the FB is derivable from a 2-unit tree of rational PRFBs. This tree has root decimator–set obtained from D by replacing $S \subset D$ by a single decimator of value N. The leaf–decimator set is derived from S by dividing all its elements by N.

Comments on Theorem 6. This result, proved in Appendix G, is more complicated to state but also more general than Theorem 5. Theorem 5 represents the special case when T_2 is empty and $S \cup T_1 = D$ (in which case (7.0.2) and (7.0.3) imply (7.0.5), due to (1.0.1)). Note that while one of the sets T_1, T_2 can be empty, (7.0.5) shows that they cannot *both* be empty except in the trivial case where $N = 1$ and $S = D$. With S, T_1 defined as in (7.0.2),(7.0.3), their disjointedness is equivalent to $|S| > 1$, which ensures that each entry of T_1 is less than all entries of S. Disjointedness of T_2 from S, T_1 is a separate requirement that does not follow from (7.0.2)–(7.0.5). Both S and T_1 can have multiple occurrences of a given decimator value; in fact from (7.0.3), every $n_i \in T_1$ occurs as many times in T_1 as it does in D. However, entries of T_2 are all distinct from each other, for else by (7.0.4), T_2 would have some elements that are factors of N and are hence in T_1 too, violating their disjointedness. Unlike Theorem 5, Theorem 6 is not obeyed by nonrational FBs even if the condition that all $n_i \in S$ be multiples of N is made an additional premise (a counterexample can be created with brickwall FBs).

Theorem 7: AC matrix based leaf extraction test. Consider Fig. 15a, showing a subset of the channels of some maximally decimated FB. Suppose the system in Fig. 15a is not identically zero, and all its filters are rational. Then the following statements are equivalent:

(a) Let $W = \exp(\frac{-j2\pi}{KM})$ and $G_l(z) \stackrel{\triangle}{=} \sum_{i=0}^{K-1} H_i(zW^l)F_i(z)$. Then, $G_l(z) = 0$ for all $l \in \{0, 1, \ldots, KM-1\}$ that are not integer multiples of K (or equivalently, as $W^{KM} = 1$, for all such integers l).

[5] Most 'sets' in our work, including D, S, T_1 here, are really 'multisets', i.e., can contain multiple occurrences of the same decimator value. However, *disjointedness* here has its usual set–theoretic meaning. Thus, here if one of the sets S, T_1, T_2 has a decimator of value m, the other two sets cannot have any decimator of value m even if D has several such decimators.

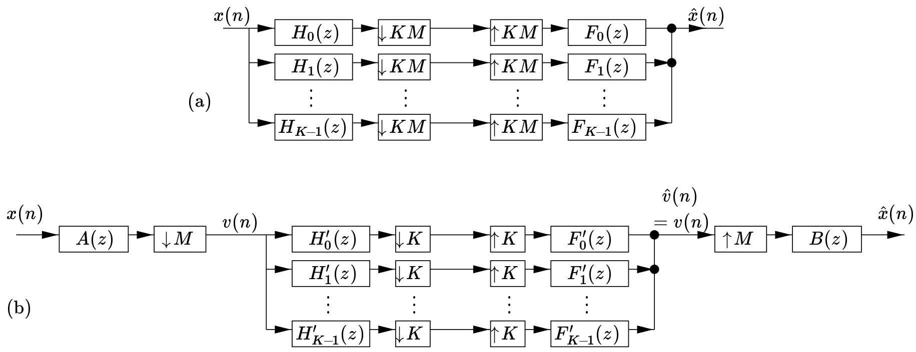


Figure 15. Leaf extraction test (Theorem 7). (a) K channels with decimator KM. (b) Equivalent structure under the premise of the test.

(b) There are rational filters A, B, H_i', F_i' such that the systems of Figs. 15a, 15b are equivalent (i.e., for $i = 0, 1, \ldots, K-1$, $H_i(z) = A(z)H_i'(z^M)$, $F_i(z) = B(z)F_i'(z^M)$) and the H_i', F_i' form a K band PRFB.

Application of Theorem 7. The result is proved in Appendix G, we mainly use the fact that (a) implies (b) in its statement. Suppose precisely K decimators of a rational PRFB have value KM. Examine the k-th row on the left side of the L–row AC matrix equation (2.2.4) of the FB (where L is a multiple of all decimators in the FB). For $k = l(L/(KM))$ (l a positive integer), this evaluates to the sum of $G_l(z)$ (defined in Theorem 7) and other terms coming from channels whose decimators n_i are such that k is a multiple of L/n_i. If there are no such terms, $G_l(z) = 0$. Even if there are such terms, we have seen in deriving the AC matrix test (Section 6.4) how one can sometimes deduce that they sum to zero (and hence that $G_l(z) = 0$) using filter rationality and the other rows in (2.2.4). Suppose the decimators are such that such a deduction of $G_l(z) = 0$ is possible for all l that are not integer multiples of K. The condition (a) of Theorem 7 is then obeyed by the K channels with decimator KM for all rational PRFBs with this set of decimators. Thus, Theorem 7 implies that all these PRFBs are derivable from a two unit tree of rational PRFBs, in which the leaf is uniform with decimator K and generates the K channels with decimator KM. Thus for such a decimator–set, existence of rational PRFBs is equivalent to existence of rational PRFBs using the smaller decimator–set in the root of the above–mentioned tree. This technique can be applied to dyadic decimator sets to deduce the result of [3]. However this result follows more easily from Theorem 5. Finally, note that even if we remove all rationality restrictions in the statement of Theorem 7, (b) still implies (a) (Appendix G). The converse (which is more useful) is however no longer true (a counter example can be created using brickwall FBs).

Theorems 5—7 involve a decimator–set D having a subset S whose entries have reciprocals summing to $1/N$ for some integer N. Given a rational PRFB using D, the goal is to derive the subset of its channels corresponding to S from a single channel with decimator N, by attaching a leaf FB using the decimator–set S/N. (S/N is obtained from S by dividing each of its entries by N.) The theorems give various conditions on D under which this can be done for all rational PRFBs using decimator–set D. That S/N is a set of integers is either an assumption or a conclusion. Note that certain *rational* PRFBs are shown to be derivable from trees whose units are all rational PRFBs. If the original PRFB obeys a constraint other than (or besides) rationality, such as orthonormality or the FIR constraint, then can all units of the tree also be chosen to obey this constraint? A partial answer (for certain constraints) lies in the following result:

Theorem 8. Consider any of the following properties of FBs: (a) PR, (b) orthonormality, (c) stable filters, and (d) FIR filters. Suppose a tree structure of rational FBs yield a (necessarily rational) FB which obeys one particular property from this list. Then the filters in each unit of the tree can be modified without changing the overall FB, in such a way that each unit also becomes a FB that satisfies that property.

This result is proved in Appendix H. There is an important point to note about the list of properties in its statement. One could consider adding to the list, *combinations* of the listed properties, i.e., (e) PR and stable filters, (f) orthonormality and stable filters, (g) PR and FIR filters, (h) orthonormality and FIR filters. However, these *have not been listed*. Thus, for instance if the overall FB has PR with stable filters, all Theorem 8 assures is that the individual FBs can be altered to have either PR or stable filters — whether they can have *both* is left undecided. Indeed, it is an *open problem* as to whether or not Theorem 8 holds with any of the properties (e)–(h) added to its list of properties (though we believe that it probably does hold even in this case). That it holds for property (h) has been proved *for dyadic trees* in [10, Th. 2]. This proof can be extended to cover both properties (f) and (h) for all uniform–trees in which no unit has more than one child (dyadic trees being a special case). Further extensions (to arbitrary trees) are unknown.[6]

Our last result is one that, given a PRFB with decimator–set S, deduces existence of another PRFB, which has a possibly different decimator–set S_1 and preserves certain properties of the original FB such as rationality and orthonormality. Thus, given a necessary condition $\mathcal{P}$ on S for existence of a PRFB using S with such a property, we can get a stronger necessary condition $\mathcal{P}_1$ by applying $\mathcal{P}$ not merely to S but also to S_1. It may turn out that $S_1 \equiv S$, or that S_1 is derivable from a tree using S as root, in which case $\mathcal{P}_1$ is automatically tested once we test for the tree version $\mathcal{P}''$ of $\mathcal{P}$ (i.e., $\mathcal{P}''$ is even stronger

[6] [10, Th. 4] appears to show Theorem 8 with property (h) for all uniform–trees, but in fact it does not: In its proof, $T_i'(z) = T_i(z)/B(z^b)$ has not been shown to be FIR. Similarly, [3, Sec. 6] seems to account for property (e), but in fact it only covers stability (property (c)).

than $\mathcal{P}_1$). However, this does not always happen, i.e., sometimes we indeed get a new condition. The result is as follows:

Theorem 9: Subset extension test. Consider any subset of channels of any PRFB. Let the decimators in this subset have lcm L and reciprocals that sum to p/L. Then there exist $L-p$ channels with decimation L which when augmented to the chosen subset, extend it into a PRFB. If the original PRFB has any *one* of the following properties: (a) rational filters, (b) orthonormality, (c) orthonormality and rational filters, (d) orthonormality and FIR filters, then the new 'extended' FB can also be chosen to have that property.

Proof: We use the equivalence between the biorthogonality condition (2.2.1) and the polyphase formulation of the PR condition on the filters. From Section 2.2, Fig. 4 and Appendix A, we can see that for a specific i, j, (2.2.1) is equivalent to $p_i \times p_j$ equations of the form $(S_i(z)Q_j(z)) \downarrow_L = \delta(i-j)$. Here $L = n_i p_i = n_j p_j$, and these equations come from choosing S_i, Q_j respectively as the delayed versions of H_i, F_j in Fig. 4. The left side of each such equation can be written as an 'inner product' of length L vectors using the polyphase lemma (Section 1.3). In order to arrange all these equations into a single polyphase matrix equation, L was chosen in Section 2.2 as a multiple of *all* decimators n_i. However, if we restrict attention to a subset of channels of the nonuniform FB (as in Theorem 9), it suffices to let L be a multiple of the decimators in this subset. Thus, the subset chosen in the theorem statement corresponds to a matrix equation $\mathbf{E}(z)\mathbf{R}(z) = \mathbf{I}$ where $\mathbf{E}(z), \mathbf{R}(z)$ are of sizes $p \times L$ and $L \times p$ respectively. The theorem then follows by augmenting these matrices into $L \times L$ ones whose product is still the identity. The augmented matrices are the polyphase matrices of the new FB, and the added rows and columns are the L-th order polyphase vectors of the filters in the added channels. Clearly if $\mathbf{E}(z), \mathbf{R}(z)$ are rational, these vectors can also be chosen to be rational. If the original FB is orthonormal, $\mathbf{E}(e^{j\omega})\mathbf{E}^{*T}(e^{j\omega}) = \mathbf{I}$, so we can extend $\mathbf{E}(e^{j\omega})$ into a unitary matrix (for each ω). Further if the original FB is rational or FIR, the extension can be forced to preserve these properties by using the unitary statespace realizations [11, Chap. 14] of $L \times p$ paraunitary systems.[7] $\bigtriangledown \bigtriangledown \bigtriangledown$

8. Summary and Comparison of Necessary Conditions

Table 1 lists all currently known necessary conditions on the decimators of rational PRFBs, many of which have been developed in this paper. The following remarks are in order:

1. For each of the tests numbered $\#i = 1, 2, \ldots, 11$, we have an example decimator–set D_i violating the test. We have chosen D_i so that the only other listed tests it fails are (a) any tests that imply test $\#i$, as shown

[7] The extension is made by extending their 'rectangular unitary' *realization matrices* into (square) unitary ones.

Test type	Test # (i)	Test	Section #	Tests implied	Example decimator–set D_i violating test #i
Pairwise gcd based	1	Pairwise noncoprimeness	6.1	—	3,4,12,12,12,12,12
	2	Pairwise gcd test	6.2 (Th. 4)	1	6,10,14,210,210,...,210
Generalizing compatibility	3	Compatibility	6.1	—	6,6,6,9,9,9,9,18
	4	Strong compatibility	6.1	3	6,6,6,6,9,9,27,27,45,270,...,270
	5	Tree version of #4	6.3, App. E	3,4	4,6,6,12,16,16,16,18,18,72,144,144,144
	6	AC matrix test	6.4, App. F	3,4	6,6,6,6,9,9,24,72,72,72,72,72
Tree reduction based	7	Root extraction (RE) test	7 (Th. 5)	—	4,4,4,10,20,20,20
	8	Generalized RE test	7 (Th. 6)	7	4,4,6,8,24,24,24,24,24
	9	AC–based leaf extraction	7 (Th. 7)	—	4,6,6,12,12,12,20,20,30,60,60
	10	Subset extension test	7 (Th. 9)	—	4,6,6,8,12,12,36,48,48,72,72,72,72
Tree versions (Section 4.3)	11	Tree version of #2	?	1,2	3,6,6,12,18,30,180,...,180
	12	Tree version of #6	?	3,4,6	??
	13	Tree version of #7–10	?	7–10	??

Table 1. Necessary Conditions on Decimators of Rational PRFBs..

in the second-last column of the table, and (b) *possibly* one or more of the tests #11–13, which we have not designed algorithms to perform, and hence, cannot currently decide whether or not they are violated. This shows that except for these last three tests, the interdependencies between the tests are exactly as described in the table (in its second-last column). For example, passing the AC matrix test implies nothing about passing the tree version of the strong compatibility test, and vice versa.[8]

2. The above remark applies in particular to the example set D_{11}, which passes all tests #1–10. It fails test #11 because attaching a uniform leaf with decimator 2 to its decimator of value 3 yields a set with a subset of 7 decimators within which the gcd of any pair is 6. Currently we do not have such examples for tests #12,13. Though we have not devised an algorithm to test for the tree version of the pairwise gcd test, the set D_{11} shows that the tree version is strictly stronger than the original test.

3. The AC matrix test (#6) is also strictly strengthened by forming its tree version (#12), as shown by the set $\{3, 4, 8, 12, 12, 24, 24, 24\}$. This set passes test #6, but fails its tree version because attaching uniform leaves with decimator 2 to its decimators of value 12 yields a set that fails test #6. However, this example also fails another test from Table 1, i.e., pairwise noncoprimeness.

4. Each test $\mathcal{P}$ of rows #11–13 is the tree version of some test $\mathcal{P}_1$. As seen in Section 4.3, the test $\mathcal{P}$ is well defined, but involves applying $\mathcal{P}_1$ to an infinite family of decimator–sets. Devising a finite algorithm for this process can take ingenuity or hard work, as seen in Section 6.3 (and Appendix E) for the tree version of strong compatibility. This is especially

[8]Example D_5 actually also fails test #10, but in a manner that makes the use of test #10 equivalent to using the tree version of another test (see discussion on Theorem 9).

true given the complex nature of some of the $\mathcal{P}_1$ (e.g., the AC matrix test). Design of algorithms for these tests is left for future work.

5. There are decimator–sets that obey all the known necessary conditions #1–10 which we have algorithms to verify, and yet do not allow building of delay-chain PRFBs. Examples are the sets $\{6, 6, 6, 6, 9, 12, 36, 36, 36, 36, 36\}$ and $\{4, 6, 6, 12, 12, 12, 16, 24, 48, 48, 48\}$ (all examples have ≥ 11 decimators). Thus, these necessary conditions, taken together, are still not equivalent to the most general known sufficient condition for existence of rational PRFBs, namely, existence of PR delay-chains. We currently do not know whether or not sets of the kind listed above allow building of rational PRFBs. Thus, the main problem of this paper (Section 3.1) remains unsolved for the rational FB class.

9. Concluding Remarks

We have presented several new conditions on the decimators of rational PRFBs, considerably generalizing many earlier known ones. Our work still leaves necessary and sufficient conditions unknown. The weakest known sufficient condition, when obeyed, allows the decimators to be used to build PRFBs as specialized as delay-chains. Thus, if we impose various less restrictive conditions on the filters of a rational FB (e.g., FIR filters, linear phase filters, orthonormality, etc.), we get many more FB classes for which we do not know the necessary and sufficient conditions on the decimators of PRFBs in the class. It has been shown [4] that existence of rational PRFBs implies that of rational *orthonormal* FBs with stable filters (i.e., all analysis filters have all poles inside the unit circle). However, whether this implies existence of *FIR orthonormal* FBs is not known. Even when a decimator–set is known to allow building of rational PRFBs, complete parameterizations of the possible PRFBs are not known, except in the restricted cases of uniform and dyadic decimator–sets. Partial parameterizations using trees have been presented in Section 7. Other specific open problems encountered in our study are listed below:

1. *Non–tree–structured PRFBs with tree structured decimator–sets:* Section 4.2 has shown two different constructions leading to such FBs. Are there any more?

2. *Forcing properties of a tree structured FB on all the tree units:* Theorem 8 (Section 7) shows that this is possible for rational FBs with certain properties (e.g., PR, FIR filters), but it is not known whether it is possible for certain others (e.g., PR *and* FIR filters).

3. *Real coefficient FBs (Section 3.2):* Do they always exist? Does existence of rational PRFBs with decimator–set S imply that of real coefficient PRFBs (rational or otherwise) using S?

4. *Algorithms for tree versions of necessary conditions:* These have not been designed for certain necessary conditions due to their complicated nature (e.g., AC matrix test), see Table 1 (Section 8).

Appendix A: Proof of Nonuniform Biorthogonality Condition (2.2.1)

The (uniform) biorthogonality condition on the uniform FB derived from a nonuniform one is equivalent to

$$\left(z^{-n_i c} H_i(z) z^{n_i d} F_i(z)\right) \downarrow_L \;=\; \delta(c-d) \qquad \text{(A.1)}$$

$$\left(z^{-n_i a} H_i(z) z^{n_j b} F_j(z)\right) \downarrow_L \;=\; 0 \quad \text{if} \quad i \neq j \qquad \text{(A.2)}$$

Here $c, d, a \in \{0, 1, \ldots, p_i - 1\}$ and $b \in \{0, 1, \ldots, p_j - 1\}$, where $L = n_i p_i = n_j p_j$. We now use the noble identity $\left(X(z^M)Y(z)\right) \downarrow_M = X(z)(Y(z)) \downarrow_M$. This shows that (A.1) is equivalent to $\left((H_i(z)F_i(z)) \downarrow_{n_i} z^{d-c}\right) \downarrow_{p_i} = \delta(c-d)$. If $c = 0$, the left side here for $d = 0, 1, \ldots, p_i - 1$ is the d-th entry in the p_i-th order analysis polyphase vector of $(H_i(z)F_i(z)) \downarrow_{n_i}$. So the equation is clearly equivalent to

$$(H_i(z)F_i(z)) \downarrow_{n_i} = 1. \qquad \text{(A.3)}$$

Next, having (A.2) hold for the said values of a, b is equivalent to having it hold for all integers a, b. This is because $L = n_i p_i = n_j p_j$, and $(A(z)) \downarrow_L = 0$ is equivalent to $\left(z^{qL} A(z)\right) \downarrow_L = 0$ for any integer q and transfer function $A(z)$ (by noble identity). As a, b take all integer values, $-n_i a + n_j b$ takes values $k \gcd(n_i, n_j)$ for all integers k. Thus, using the noble identity, with $A_{ij}(z) \stackrel{\triangle}{=} (H_i(z)F_j(z)) \downarrow_{\gcd(n_i,n_j)}$, (A.2) is equivalent to

$$\left(A_{ij}(z)\; z^k\right) \downarrow_{L/\gcd(n_i,n_j)} = 0 \quad \text{for all integers } k \quad \text{if} \quad i \neq j \qquad \text{(A.4)}$$

The left side here includes all entries of the order $L/\gcd(n_i, n_j)$ polyphase vector of $A_{ij}(z)$, so (A.4) is equivalent to

$$A_{ij}(z) = (H_i(z)F_j(z)) \downarrow_{\gcd(n_i,n_j)} = 0 \quad \text{if} \quad i \neq j. \qquad \text{(A.5)}$$

Thus (A.1),(A.2) are equivalent to (A.3),(A.5) respectively; proving the nonuniform biorthogonality equation (2.2.1).

Appendix B: Derivability of Decimator–sets from a Uniform-tree

Claim: If a set S of decimators satisfying (1.0.1) is derivable from a uniform–tree, then it is derivable from a uniform–tree in which the root has decimator g, where g is the gcd of all elements of S.

Proof: (Can be skipped without losing continuity.) Use induction on the number N of units in the tree. Let r be the root decimator in the given uniform–tree. Clearly r divides all decimators in S, so r divides g. Now the root can have at most r children. If it has less than r children, then r is a decimator in S. Hence $g <= r$, implying $g = r$, i.e., the root already has decimator g. This proves the claim for $N = 2$, as the root of a 2-unit tree has 1 child and $1 < r$. If the root has all r children, consider any child along with all its descendants. These units form another uniform–tree. All decimators generated by this new tree are multiples of g/r. Let g' be their gcd, thus $g' = k(g/r)$ for some integer k. The new tree has $\leq N-1$ units. So by the induction hypothesis, the decimators it generates can be rederived from a uniform–tree with root decimator g'. Since $g' = k(g/r)$, this root unit can then be rederived from a uniform–tree having decimator g/r for root and k for all the g/r leaves, each of which is a child of the root. After making all these replacements on the starting tree, all children of its root now have decimator g/r, hence the root and its children can be replaced by a single uniform unit with decimator g. This proves the claim. $\bigtriangledown\bigtriangledown\bigtriangledown$

The above result suggests an algorithm [6] that tries to build a uniform–tree starting from its root:

Root–to–leaves Algorithm. (Tests derivability of a given set S of decimators from a uniform–tree)

1. Find gcd g of all elements of S. If $g = 1$, then S is not derivable from a uniform–tree.

2. Divide all entries of S by g (represents choosing root decimator g). Find *all* possible partitions of the resulting set into g groups each of which is a valid set of decimators obeying maximal decimation (1.0.1).

3. We can derive S from a uniform–tree if and only if among these partitions, there is at least one in which each group is derivable from a uniform–tree.

The algorithm is recursive. At Step 2, dividing the entries of S by g yields a set S' lower–bounded by unity. In the ensuing partition of S', any unity element in S' is all by itself a valid group viewed as derivable from a uniform–tree for purposes of Step 3. Such a group denotes absence of a child of the root, just as groups with more than one element represent children of the root. There may possibly be *no* valid partition at Step 2, e.g., when $S = \{4, 6, 6, 10, 10, 10, 10, 120, 120\}$. This of course means that there is no uniform–tree.

Note that though Step 2 can always be implemented in principle, doing it with a simple and efficient algorithm can be tricky. An alternative method builds the tree starting from a leaf and avoids this problem. Its basic idea is in identifying a leaf: Given an arbitrary decimator d in a set S derivable from a uniform–tree, it is not clear whether d is obtained at a leaf unit of the tree.

However, this *must* be the case if d is the *maximum* element in S, and further the leaf decimator must of course then divide d. Based on this, we have:

Leaf–to–root Algorithm. (Tests derivability of a given set S of decimators from a uniform–tree)

1. If S has no more than two *distinct* decimators, it is derivable from a tree.
2. Find the maximum entry m in S, and the number of times N that it occurs. For each factor k of m such that $1 < k \leq N$, form a smaller set S_k by setting $S_k = S$ and then replacing k of the elements of value m in S_k by one element of value m/k (i.e., form a leaf unit that is uniform with decimator k).
3. The set S is derivable from a uniform–tree if and only if at least one of the S_k above is.

This is another recursive algorithm, more elegant and simpler to implement, though it may be somewhat unclear whether or not it is faster. Its only step still requiring justification is Step 1. This is easily done: Suppose $S = \{m_0, \ldots, m_0, m_1, \ldots, m_1\}$ with m_i occurring N_i times $(i = 0, 1)$. Let $m_i = gd_i$ where $g = \gcd(m_0, m_1)$. Maximal decimation implies that $N_0 d_1 + N_1 d_0 = g d_0 d_1$. As d_0, d_1 are coprime, this means that $N_i = D_i d_i$ for integers D_i, $i = 0, 1$, where $D_0 + D_1 = g$. Thus S is derivable from a uniform–tree in which the root has decimator g, all its children are leaves, and D_i leaves have decimator d_i $(i = 0, 1)$.

Only necessary and only sufficient conditions. Presence of no more than two distinct decimators, as shown above, is an example of a *sufficient* condition for derivability from a uniform–tree. It is by no means necessary. Another such example is the condition that each decimator divides every decimator larger than itself (a special case is when all of them are powers of the same number). This condition neither implies nor is implied by the earlier one, and neither condition is necessary, as exemplified by the set $\{4, 4, 6, 6, 12, 12\}$. Sufficiency of the new condition is proved using the root–to–leaves algorithm: Clearly $g > 1$ at Step 1, as g is the smallest decimator. At Step 2 in formation of the partition, if we sequentially select elements from the smallest upwards, the condition ensures that at some stage the reciprocals of the selected elements will sum to unity. Repeating this process results in a valid partition, and further each of its groups also satisfies the condition. Thus the proof is completed by induction on the number of decimators.

Derivability of a set of decimators from uniform–trees implies existence of various types of PRFBs (including PR delay-chains) using those decimators. Thus, any conditions necessary for such existence are also necessary for derivability from uniform–trees. Their necessity is often provable directly from the above algorithms. For example, without pairwise noncoprimeness (Section 6.1),

$g = 1$ at Step 1 of the root–to–leaves algorithm. If compatibility (Section 6.1) is violated, i.e., if a decimator d does not divide any other decimator, then eventually $m = d$ and $N = 1$ at Step 2 of the leaf–to–root algorithm, i.e., there are no sets S_k. As tests for such necessary conditions are inconclusive whenever they are satisfied, they cannot replace the earlier complete algorithms, though they can potentially increase their efficiency.

Appendix C: Proof of Fact 3

Let $R = \{m_0, \ldots, m_{M-1}\}$, $S = \{k_0, \ldots, k_{K-1}\}$. So $T = \{n_0, \ldots, n_{K+M-2}\}$ with $n_i = m_0 k_i$ for $i = 0, 1, \ldots, K-1$ and $n_{K-1+i} = m_i$ for $i = 1, 2, \ldots, M-1$. Let $L = \mathrm{lcm}\{n_i\}$, $p_i = L/n_i$. Let n_i occur N_i times in T. Let b_i be the strong compatibility lower bound on N_i. The proof is in two parts:

Part 1: Uniformity of S. Suppose S is not a uniform unit, we will then show that $b_j \leq N_j$ for all j, i.e., T is strong compatible. Indeed for $j = 0, 1, \ldots, K-1$ we have from (6.1.1),

$$p_j b_j = \min_{p_i \neq p_j} \mathrm{lcm}(p_i, p_j) \quad \leq \quad \min_{p_i \neq p_j,\, 0 \leq i < K} \mathrm{lcm}(p_i, p_j) \tag{C.1}$$

$$\leq \quad p_j N_j \qquad \text{(as } S \text{ is strong compatible)} \tag{C.2}$$

The minimization on the right side of (C.1) is not over an empty set because S is nonuniform, i.e., $p_i \neq p_j$ for at least one i such that $0 \leq i < K$. The right side of (C.1) thus equals $p_j b_j^S$ where b_j^S is the strong compatibility (lower) bound on the number N_j^S of occurrences of k_j in S. This bound holds by strong compatibility of S, and $N_j^S \leq N_j$. This justifies (C.2), and thus $b_j \leq N_j$ for $0 \leq j < K$. For $j \geq K$, if $n_j = m_0$ then $b_j = 1 \leq N_j$, because n_j divides a distinct decimator $n_0 = n_j k_0$. If $n_j \neq m_0$, then $N_j \geq N_j^R$, the number of occurrences of $n_j = m_{j-(K-1)}$ in R. Let b_j^R be the strong compatibility lower bound on N_j^R. Thus $b_j^R \leq N_j^R \leq N_j$, and with

$$A = \min_{p_{K-1+i} \neq p_j,\, i \geq 1} \mathrm{lcm}(p_{K-1+i}, p_j) \quad \text{and} \quad B = \mathrm{lcm}(L/m_0, p_j), \quad \text{we have} \tag{C.3}$$

$$p_j b_j^R = \min(A, B), \quad \text{while} \quad p_j b_j = \min_{p_i \neq p_j} \mathrm{lcm}(p_i, p_j) \tag{C.4}$$

Thus if $A \leq B$ in (C.3) (e.g., this holds if $m_i = m_0$ for some $i > 0$), then clearly $b_j \leq b_j^R \leq N_j$. Even if $A > B$,

$$p_j b_j^R = \mathrm{lcm}(L/m_0, p_j) \geq \min_{p_i \neq p_j,\, 0 \leq i < K} \mathrm{lcm}(p_i, p_j) \geq p_j b_j,$$

as $p_i = L/(m_0 k_i)$ for $i < K$, and nonuniformity of S again ensures that $\mathrm{lcm}(p_i, p_j)$ is not being minimized over an empty set. (Nonuniformity of S is not needed here if $L/(m_0 K) \neq p_j$.) So again $b_j \leq b_j^R \leq N_j$. Thus, $b_j \leq N_j$

for all j, i.e., T is strong compatible, contradicting the premise of Fact 3. Hence S must be a uniform unit, i.e., $k_0 = k_1 = \ldots = k_{K-1} = K$. $\bigtriangledown\bigtriangledown\bigtriangledown$

Part 2: Necessary conditions for $b_j > N_j$. We have already shown in Part 1 that if $j \geq K$, then $b_j > N_j$ is possible only if m_0 occurs only once in R and $m_0 K = n_j$. The proof of Fact 3 will be completed if we show a similar statement for $j < K$, i.e., that $b_j > N_j$ is possible only if m_0 occurs only once in R and $m_0 K = n_i$ for some $i \geq K$. To show this, note that for all $j < K$, all the n_j are identical (shown by Part 1), and hence the same holds for the N_j and the b_j. Also $N_j \geq K$. Thus it suffices to show that $b_0 \leq K$ if either $m_0 = m_l = n_{K-1+l}$ for some $l > 0$, or $m_0 K \neq m_i$ for all $i > 0$. If $m_0 = m_l = n_{K-1+l}$ for some $l > 0$, then

$$p_0 b_0 = \min_{p_i \neq p_0} \operatorname{lcm}(p_i, p_0) \leq \operatorname{lcm}(p_{K-1+l}, p_0) = \operatorname{lcm}\left(\frac{L}{m_0}, \frac{L}{m_0 K}\right) = \frac{L}{m_0} = p_0 K,$$

hence $b_0 \leq K$. If on the other hand m_0 occurs exactly once in R, then $m_0 F = m_l = n_{K-1+l}$ for some $F > 1$, $l > 0$ since R is compatible. Thus if $m_0 K \neq m_i$ for all $i > 0$, then

$$p_0 b_0 = \min_{p_i \neq p_0} \operatorname{lcm}(p_i, p_0) \leq \operatorname{lcm}(p_{K-1+l}, p_0) = \operatorname{lcm}\left(\frac{L}{m_0 F}, \frac{L}{m_0 K}\right) \leq \frac{L}{m_0} = p_0 K,$$

hence $b_0 \leq K$ again. This establishes the claim, hence proving Fact 3. $\bigtriangledown\bigtriangledown\bigtriangledown$

Appendix D: Proof of Fact 4

From the premise of Fact 4, there is a tree $\mathcal{T}'$ in which each unit is either D or allows building of rational FBs (e.g., uniform units), such that $\mathcal{T}'$ generates a set of decimators that is not strong compatible. Note that every unit in $\mathcal{T}'$ is strong compatible. We now perform a series of operations on $\mathcal{T}'$, each yielding a new tree with all the properties of $\mathcal{T}'$, until finally we get the tree $\mathcal{T}$ with the desired properties as in Fact 4.

If the root of $\mathcal{T}'$ has a child that is not a leaf, then this child, along with all its descendants, forms a tree with fewer units than $\mathcal{T}'$. We can assume that this tree generates a strong compatible decimator–set (else we can replace $\mathcal{T}'$ by this tree and repeat the process). We then view this tree as a *single* unit. This makes every child of the root of $\mathcal{T}'$ a strong compatible leaf. Next, we delete any leaf such that the residual tree generates a decimator–set that is not strong compatible. This yields the desired tree $\mathcal{T}$ having all properties of $\mathcal{T}'$. We now show that $\mathcal{T}$ and the decimator–set T it generates have all the properties listed in Fact 4.

Properties 1,2,4: For any leaf S of $\mathcal{T}$, we see that $\mathcal{T}$ can be redrawn as a 2-unit tree with strong compatible units R and S. However $\mathcal{T}$ itself generates the set T that is not strong compatible. Thus we can use Fact 3 to conclude the following: (a) All leaves of $\mathcal{T}$ are uniform. (b) For any decimator value obtained at a leaf of $\mathcal{T}$, decimators of T with that value are the only ones in T that violate the strong compatibility lower bound on the number of their occurrences in T. (c) Property 2 of Fact 4 holds. Now (b) implies that all decimators obtained at the leaves have the same value d. Also, (a) implies that $\mathcal{T}$ has root D: Otherwise the root allows building of rational PRFBs, and hence, so does $\mathcal{T}$ (as all children of its root are uniform leaves); violating the fact that T is not strong compatible. This completes the proof of property 1. Property 4 follows from this and conclusion (b) listed above. Thus we have shown properties 1,2,4 of Fact 4. $\bigtriangledown\bigtriangledown\bigtriangledown$

Property 3: Let k_i be the decimator value of the leaf attached to $d_i \in D$ to form $\mathcal{T}$. As $d_i k_i = d$, we have $d = C\,\mathrm{lcm}\{d_i\}$ where $C = \gcd\{k_i\}$. We must show that if $d \notin D$, then $C = 1$. In fact, this may be *false.* Our approach is to assume that $d \notin D$, and then create a new tree $\mathcal{T}^*$ generating a decimator–set T^* with all the properties of Fact 4. This is done by replacing every leaf decimator k_i with k_i/C. (If $k_i = C$ this means deleting the leaf.) Clearly property 1 of Fact 4 continues to hold, with the decimators obtained at the leaves now having value $d^* = d/C = \mathrm{lcm}\{d_i\}$. To prove property 2, let decimator d_i of D have a leaf attached to it in $\mathcal{T}^*$. Then it also has a leaf (uniform with decimator k_i) attached in $\mathcal{T}$. As $d_i \notin T$ (by property 2 for $\mathcal{T}$), the only way to have $d_i \in T^*$ is that d_i be the newly formed decimator d/C. This however means that $k_i = C$ (as $d = d_i k_i$), i.e., the leaf attached to d_i in $\mathcal{T}$ has been deleted in $\mathcal{T}^*$, contradicting the assumption on d_i. Thus $d_i \notin T^*$, i.e., $\mathcal{T}^*$ obeys property 2. Next we prove property 3. As already seen, if $k_j = C$ for some j, then $d^* = d/C = d_j \in D$. Thus, if $d^* \notin D$, then $k_j > C$ for all j, i.e., decimators d_i with leaves attached in $\mathcal{T}$ are the same as those with leaves attached in $\mathcal{T}^*$. So property 3 holds for $\mathcal{T}^*$ from $d^* = d/C = \mathrm{lcm}\{d_i\}$. Lastly, we show property 4, i.e., that d^* violates the strong compatibility lower bound b^* on the number N^* of its occurrences in T^*. Let N be the number of occurrences of d in T, and let b be the strong compatibility lower bound on N. Let L be any common multiple of the decimators of T. We must show that $b^* > N^*$. Since $\mathcal{T}$ obeys property 4, we have $b > N$. Also, by construction of $\mathcal{T}^*$ and the hypothesis $d \notin D$, we have $N^* \geq N/C$. The inequality is strict only if $d/C \in T$, but this would imply (by definition (6.1.1) of b) that $b \leq \left(\frac{d}{L}\right) \mathrm{lcm}(\frac{L}{d}, \frac{LC}{d}) = C$. Since $N \geq k_i \geq C$, we get $b \leq N$, a contradiction. Thus $d/C \notin T$, and hence $N^* = N/C$. Lastly, $b^* = \left(\frac{d}{LC}\right) \mathrm{lcm}(\frac{LC}{d}, \frac{L}{m})$ for some $m \in T^*$, $m \neq d/C$. Thus $m \in D$ and $m \in T$ too, and $m \neq d$ by the hypothesis $d \notin D$. So $b \leq \left(\frac{d}{L}\right) \mathrm{lcm}(\frac{L}{d}, \frac{L}{m}) \leq C\left(\frac{d}{LC}\right) \mathrm{lcm}(\frac{LC}{d}, \frac{L}{m}) = Cb^*$. Hence, $b^* \geq b/C > N/C = N^*$ (using $b > N$). Thus $b^* > N^*$ as required. $\bigtriangledown\bigtriangledown\bigtriangledown$

Appendix E: Testing Tree Version of Strong Compatibility

Given a decimator–set D, let $V = \{v_0, v_1, \ldots, v_{K-1}\}$ be the set of *distinct* decimator values in D, with v_i occurring N_i times in D. Let L be any multiple of all the v_i, i.e., of $\mathrm{lcm}\{v_i\}$, and let $p_i = L/v_i$. Then D satisfies the tree version of strong compatibility if and only if Routine 1 below returns the value 'TRUE' for all $v_i \in V$ *and* Routine 2 returns value 'TRUE'.

Routine 1: (To be performed for all $v_i \in V$)

1. Initialization: Set $M = N_i$, $A = V$ and delete v_i from A.
2. If A is empty, return(TRUE). Else, let $j = l$ minimize $\mathrm{lcm}(p_i, p_j)$ over all j such that $v_j \in A$. If $M < \mathrm{lcm}(p_i, p_l)/p_i$, return(FALSE).
3. If v_l does not divide v_i, return(TRUE). Else, add $N_l(v_i/v_l)$ to M and delete v_l from A. This represents attaching to every decimator of value v_l, a leaf that is uniform with decimator v_i/v_l. Then go to Step 2.

Routine 2:

1. Find all subsets S of V having at least two but less than $K-1$ elements, such that the lcm $l(S)$ of all elements of S does not divide any $v_j \in V$.
2. For each S of Step 1, let $\sigma(S)$ be the sum of all the numbers $N_i(l(S)/v_i)$ for all $v_i \in S$. Let $b(S)$ be the minimum of $\left(\frac{l(S)}{L}\right) \mathrm{lcm}(\frac{L}{l(S)}, \frac{L}{v_i})$ over all $v_i \notin S$. This step represents attaching to every decimator whose value v_i lies in S, a leaf unit that is uniform with decimator $l(S)/v_i$, so that all decimators thus obtained at the leaves have value $l(S)$. In the resulting tree structured set of decimators, $\sigma(S)$ is the number of occurrences of decimator $l(S)$ and $b(S)$ is the strong compatibility lower bound on $\sigma(S)$.
3. If $\sigma(S) \geq b(S)$ *for all* S above, return(TRUE). Else return(FALSE).

The action of the routines is independent of which multiple of $\mathrm{lcm}\{v_i\}$ we choose L to be. To explain how the above test works, refer to the statement of Fact 4. Routine 2 lists all trees $\mathcal{T}$ obeying properties 1,2,3 of Fact 4 such that $d \notin D$ (see property 3), and returns a 'FALSE' value if any of these obey property 4. The set S of Step 1 represents choice of the d_i of property 2. We demand that S must have at least two elements, and that $l(S) \neq v_j$ for all $v_j \in V$, to ensure that property 3 holds with $d \notin D$. In fact we further demand that $l(S)$ must not divide any $v_j \in V$, for if it does, $b(S) = 1$ at Step 2. We also exclude sets S with $\geq K-1$ elements, for then $\mathcal{T}$ generates a set with at most two *distinct* decimators. Such a set, being derivable from a uniform–tree (Appendix B), is always strong compatible, i.e., $\sigma(S) \geq b(S)$ will hold at Step 3.

Routine 1 becomes a test for strong compatibility if we delete Step 3 in it. Hence we can assume strong compatibility of the given set of decimators. Thus the only task remaining is to examine whether there is a tree $\mathcal{T}$ obeying all properties of Fact 4 with $d \in D$ in property 3. This is achieved by the addition of Step 3. To see this, let there be such a tree $\mathcal{T}$, with $d = v_i$, producing a set T of decimators. The quantity $b = \text{lcm}(p_i, p_l)/p_i$ of Step 2 is the lower bound on N_i, which holds by assumption of strong compatibility. Now the number N_T of occurrences of v_i in T is at least N_i. Further if $v_l \in T$, then the strong compatibility lower bound on N_T does not exceed b, and hence cannot be violated. Thus $v_l \notin T$, i.e., all decimators of value v_l must have leaves attached to them to convert them into decimators of value v_i. This justifies Step 3.

In the special case when $L \stackrel{\triangle}{=} \text{lcm}\{v_j\} \in V$, Routine 2 can be skipped (it always returns 'TRUE'), and Routine 1 needs execution only for $v_j = L$ (it returns 'TRUE' for all other v_j). This is provable from the fact that for $v_j = L$, $p_j = 1$. In general, Routine 1 appears to be the important part of the test: There are relatively fewer decimator–sets for which violation of the test is detected by Routine 2 but not by Routine 1 (examples of such sets being $\{2, 3, 24, 24, 36, 36, 36\}$ and $\{2, 4, 6, 48, 48, 72, 72, 72\}$).

Appendix F: Algorithm for the AC Matrix Test

In the given set of decimators, let $v_0, v_1, \ldots, v_{K-1}$ be the *distinct* decimator values, with v_j occurring N_j times. Let L be any common multiple of the v_j, and let $p_j = L/v_j$. The algorithm is then as follows:

1. *Initialization.* Create a matrix $\mathbf{U}$ with rows numbered 0 to $L - 1$ and columns 0 to $K - 1$, where the lj-th entry u_{lj} is 1 if l is a multiple of p_j, and zero otherwise. Thus $\mathbf{U}$ is initialized to describe the positions of the zero and nonzero entries in the AC matrix (2.2.4),(2.2.5). In particular, $u_{0j} = 1$ for all j.

2. Set $\mathbf{U}' = \mathbf{U}$ (saving the current value of $\mathbf{U}$ in $\mathbf{U}'$). For all l, j such that u_{lj} is the only entry in the l-th row having value unity, set $u_{lj} = 2$. This identifies sets of filters having the same decimator value v_j, and satisfying an equation of the form $\sum_i B_i(zW^l)C_i(z) = 0$.

3. For each $d = kp_j$ for integer k obeying $1 \le kp_j \le \lfloor L/2 \rfloor$, let $c_s^d(n) = s + nd$ for $s = 0, p_j, 2p_j, \ldots, d - p_j$. If $u_{lj} = 2$ for $l \equiv c_s^d(n) \pmod{L}$ for N_j consecutive integers n, set $u_{lj} = 2$ for $l \equiv c_s^d(n) \pmod{L}$ for *all* integers n. Do this for each $j = 0, 1, \ldots, K - 1$. (This represents use of Lemma 3.)

4. If $u_{0j} = 2$ for any j, the given set of decimators fails the AC matrix test. (This is where we apply Lemma 2.) If $\mathbf{U}' = \mathbf{U}$, the set passes the test. If neither of these happens, go to Step 2.

Passing the above test is a necessary condition on the decimators of any rational PRFB, as the discussion of Section 6.4 proves. The test outcome is independent of which common multiple of the v_j we choose L to be. The above algorithm may be made more efficient in many ways (e.g., we can declare the test as passed if $\mathbf{U}' = \mathbf{U}$ after Step 2); our main purpose here is to state a correct (rather than highly efficient) algorithm.

Lastly, we prove that the above test implies strong compatibility. Consider any fixed $j \in \{0, 1, \ldots, K-1\}$, and find the smallest $l > 0$ such that u_{lj} is not set to value 2 at Step 2. This is the smallest nonzero multiple of p_j that is also a multiple of some $p_i \neq p_j$, i.e., it is $\min_{p_i \neq p_j} \operatorname{lcm}(p_i, p_j) = p_j b_j$ where b_j is as in (6.1.1). Thus, after Step 2, $u_{lj} = 2$ for $l = kp_j$ for $k = 1, 2, \ldots, b_j - 1$. So if $N_j < b_j$, Step 3 will use the sequence $c_0^{p_j}(n)$ to set $u_{lj} = 2$ for all $l = np_j$. In particular it sets $u_{0j} = 2$, which means that the test is failed (see Step 4). Hence if the test is passed, we have $N_j \geq b_j$ for all j, which is the strong compatibility condition (6.1.1).

Appendix G: Proofs of Theorems 6,7

Proof of Theorem 6: We will prove the claim of the theorem after replacing its premises (7.0.2)–(7.0.5) about the decimator–set D by the following premise: The set D has two nonempty disjoint subsets S, T such that

$$\begin{aligned} \sum_{n_i \in S} \frac{1}{n_i} &= \frac{1}{N} \text{ for some integer } N, && \text{(G.1)} \\ |T| &= N - 1, \quad \text{and} && \text{(G.2)} \\ \gcd(n_i, n_j) &= \text{factor of } N \quad \text{whenever} \quad n_i \in S \cup T,\, n_j \in T,\, i \neq j && \text{(G.3)} \end{aligned}$$

This suffices because from a rational PRFB obeying (7.0.2)–(7.0.5), we can create one obeying (G.1)–(G.3) by inserting in each of its channels with decimator $n_i \in T_1$, a uniform rational PRFB with decimator N/n_i. This process preserves the channels corresponding to the decimator subset S, and creates $\left(\sum_{n_i \in T_2} (\frac{N}{n_i})\right)$ new decimators each of value N. The set T consists of T_2 and these new decimators; thus (G.2) follows from (7.0.5), and (G.3) from (7.0.4) and the fact that the new decimators have value N. Having proved the claim using (G.1)–(G.3), we remove the inserted uniform leaf FBs to prove it under the original premise (7.0.2)–(7.0.5).

Part 1: Proof under additional assumption that all $n_i \in S$ are multiples of N. Let us be given a rational PRFB with decimator–set D and filters as in Fig. 1, such that D has disjoint subsets S, T obeying (G.1)–(G.3). Let $\mathbf{E}(z), \mathbf{R}(z)$ respectively be the N-th order analysis and synthesis polyphase matrices of the analysis and synthesis filters corresponding to channels with decimators $n_i \in T$. Let $\mathbf{e}_i(z)$ be the N-th order analysis polyphase vector of $H_i(z)$ where $n_i \in S$. From (G.2), $\mathbf{E}(z), \mathbf{R}(z)$ have sizes $(N-1) \times N$ and $N \times (N-1)$ respectively. We use (G.3) with the PR condition (2.2.1) and the polyphase lemma, as in Section 6.2. This shows that $\mathbf{e}_i(z)\mathbf{R}(z) = \mathbf{0}$, and that $\mathbf{E}(z)\mathbf{R}(z)$ is a $(N-1)\times(N-1)$ diagonal matrix, none of whose diagonal entries is identically zero. This implies (using rationality of the filters) that $\mathbf{R}(z)$ has $N-1$ linearly independent columns. All the $\mathbf{e}_i(z)$, being 'orthogonal' to all these columns, must be 'proportional', i.e., $\mathbf{e}_i(z) = H_i'(z)\mathbf{a}(z)$ for some rational filters $H_i'(z)$ and vector $\mathbf{a}(z)$. Let $A(z)$ be the filter with $\mathbf{a}(z)$ as its N-th order analysis polyphase vector. Computing $H_i(z)$ from $\mathbf{e}_i(z)$ shows that $H_i(z) = A(z)H_i'(z^N)$. A similar argument shows that for all i such that $n_i \in S$, $F_i(z) = B(z)F_i'(z^N)$ for some rational $B(z), F_i'(z)$. Thus, under the additional assumption that all decimators in S are multiples of N, we see that the given rational PRFB is derivable from a two unit tree of rational FBs. The units of the tree have decimator–sets exactly as desired, and using Theorem 8, their filters can further be modified so that they also have PR. This completes Part 1 of the proof.

Part 2: Extending Part 1 to nonrational FBs in the setting of Theorem 5. When the original premises (7.0.2)–(7.0.5) of Theorem 6 are obeyed in the special manner that results in the premise of Theorem 5, the effect on (G.1)–(G.3) is to cause $D = S \cup T$ and $n_j = N$ for all $n_j \in T$. Now in Part 1, the diagonal elements of $\mathbf{E}(z)\mathbf{R}(z)$ are $(H_j(z)F_j(z))\downarrow_N$ where $n_j \in T$ (by polyphase lemma). Thus, in the above special case, by (2.2.1), in fact $\mathbf{E}(z)\mathbf{R}(z)$ is the identity. Hence we can choose the $A(z), B(z)$ of Part 1 to have N-th order analysis and synthesis polyphase vectors $\mathbf{a}(z), \mathbf{b}(z)$ respectively, such that the $N\times N$ matrices $\begin{bmatrix} \mathbf{E}(z) \\ \mathbf{a}(z) \end{bmatrix}$ and $\begin{bmatrix} \mathbf{R}(z) & \mathbf{b}(z) \end{bmatrix}$ have product equal to identity. This possible even without any rationality restriction on the filters (of course A, B are then nonrational in general). These matrices now become the polyphase matrices of the root FB. Thus, the root automatically has PR, and hence so does the leaf (since the overall FB has PR), without the need to use Theorem 8 (which requires filter rationality). Thus, for the special case of Theorem 5 (as distinct from the general setting of Theorem 6), we have extended Part 1 to nonrational FBs.

Part 3: Proving the additional premise used in Part 1, using filter rationality. For each i such that $n_i \in S$ we insert a q_i channel uniform rational PRFB within the i-th channel of the given PRFB, where $q_i = \mathrm{lcm}(N, n_i)/n_i$. This forms q_i new decimators of value $n_i q_i$. Let S' be the set of these decimators. Then, the newly formed tree–structured rational PRFB also has a decimator–set satisfying the premises (G.1)–(G.3), with S replaced by S' and T unchanged. Indeed, (G.1),(G.2) obviously hold, while (G.3) follows from the observation that if $\gcd(n_i, n_j)$ is a factor of N and q_i contains precisely the factors of N that are not present in n_i (i.e., $q_i = \mathrm{lcm}(N, n_i)/n_i$) then $\gcd(n_i q_i, n_j)$ is also a factor of N. Further S' also obeys the additional assumption that its elements are multiples of N, by the choice of the q_i. Let $q_i > 1$ and consider two analysis filters $C_i^l(z)$, $l = 0, 1$ of the q_i band leaf FB inserted in the channel with decimator $n_i \in S$. The corresponding analysis filters of the new tree–structured FB are

$H_i(z)C_i^l(z^{n_i})$. However, using Theorem 6 (which Part 1 has proved for the new FB), these filters have the form $A(z)D_i^l(z^N)$ for some rational $D_i^l(z), A(z)$ where $A(z)$ is independent of l, i. Taking ratios of these filters (a crucial step that requires filter rationality) shows that

$$\frac{C_i^0(z^{n_i})}{C_i^1(z^{n_i})} = \frac{D_i^0(z^N)}{D_i^1(z^N)}, \tag{G.4}$$

which implies that each equals $X_i(z^{\text{lcm}(N,n_i)})$ for some rational $X_i(z)$. Replacing z by z^{1/n_i} and using the definition of q_i, we have $\frac{C_i^0(z)}{C_i^1(z)} = X_i(z^{q_i})$. This means that the q_i-th order analysis polyphase vectors $\mathbf{e}_i^l(z)$ of $C_i^l(z)$, $l = 0, 1$, are linearly dependent, as $\mathbf{e}_i^0(z) = \mathbf{e}_i^1(z)X_i(z)$. Thus, the inserted q_i band uniform leaf FB with the filters $C_i^l(z)$, while assumed to have PR, has an analysis polyphase matrix that is not invertible (since it contains the rows $\mathbf{e}_i^l(z), l = 0, 1$). This contradiction disproves the assumption that $q_i > 1$. Hence $q_i = 1$, or in other words, n_i is a multiple of N. $\triangledown\triangledown\triangledown$

Proof of Theorem 7: We first write the input–output relations, analogous to (2.1.2), of the systems of Fig. 15:

$$\hat{X}(z) = \frac{1}{KM}\sum_{l=0}^{KM-1} X(zW^l)G_l(z) \qquad \text{for Fig. 15a} \tag{G.5}$$

$$\hat{X}(z) = \frac{1}{M}\sum_{l=0}^{M-1} A(zW^{Kl})B(z)X(zW^{Kl}) \quad \text{for Fig. 15b} \tag{G.6}$$

Here G_l are as defined in statement (a) of Theorem 7, and (G.6) uses the PR property of the FB formed by the H_i', F_i'. That (b) implies (a) in Theorem 7 follows directly by comparing (G.5) and (G.6), and holds even without any rationality requirements on the filters. We now prove that (a) implies (b) (for which the rationality is essential). Form the M-th order AC matrix $\mathbf{H}(z)$ (of size $M \times K$) using analysis filters $H_i(z)$, i.e., let the q-th row of $\mathbf{H}(z)$ be $(H_0(zW^{Kq}), H_1(zW^{Kq}), \ldots, H_{K-1}(zW^{Kq}))$ for $q = 0, 1, \ldots, M-1$. Let $\mathbf{f}(z) = (F_0(z), F_1(z), \ldots, F_{K-1}(z))^T$. Thus, the condition (a) is equivalent to $\mathbf{H}(zW^l)\mathbf{f}(z) = \mathbf{0}$ for $l = 1, 2, \ldots, K-1$. Replacing z by zW^{-l}, $\mathbf{H}(z)\mathbf{f}(zW^{-l}) = \mathbf{0}$. Now the $K-1$ columns $\mathbf{f}(zW^{-l})$, $l = 1, 2, \ldots, K-1$ are linearly independent. For otherwise, there are rational filters $\alpha_l(z)$ such that $\sum_{l=j}^{K-1} \alpha_l(z)\mathbf{f}(zW^{-l}) = 0$ for all z, where $1 \le j < K$ and $\alpha_j(z) \not\equiv 0$. Dividing this by $\alpha_j(z)$ and replacing z with zW^j shows that $\mathbf{H}(z)\mathbf{f}(z) = \mathbf{0}$ too. This would mean that $G_l(z) = 0$ *for all* integers l. This shows, by (G.5), that the system of Fig. 15a is identically zero, contradicting the premise of the theorem. Thus, the $K-1$ columns $\mathbf{f}(zW^{-l})$, $l = 1, 2, \ldots, K-1$ are linearly independent, and each row of $\mathbf{H}(z)$ is 'orthogonal' to all these columns (i.e., their product is identically zero). Hence all these rows must be 'proportional' to each other, i.e., $\mathbf{h}_1(z) = C(z)\mathbf{h}_0(z)$ for some scalar filter $C(z)$, where $\mathbf{h}_i(z)$ is the i-th row of $\mathbf{H}(z)$. This means that $H_i(zW^K)/H_0(zW^K) = H_i(z)/H_0(z) \stackrel{\triangle}{=} D_i(z)$, i.e., $D_0(z) = 1$ and for $i = 1, 2, \ldots, K-1$, $D_i(e^{j\omega}) = D_i(e^{j(\omega + \frac{2\pi}{M})})$, i.e., $D_i(e^{j\omega})$ is periodic with period $\frac{2\pi}{M}$. So $D_i(e^{j\omega}) = P_i(e^{j\omega M})$, i.e., by rationality, $D_i(z) = P_i(z^M)$. Thus, $H_i(z) = A(z)H_i'(z^M)$ where $A(z) = H_0(z)$ and $H_i'(z) = P_i(z)$, showing that the analysis banks of Figs. 15a and 15b can be made equivalent. Next, replacing z with zW^{-l} in condition (a) of the theorem shows that the condition holds even if each H_i is interchanged with F_i. Hence the same process can be repeated for the synthesis banks.

The above process may not ensure PR for the K band FB formed by the H_i', F_i' (which we will refer to as the leaf FB). However, G_l now takes the form $G_l(z) = A(zW^l)B(z)\sum_{i=0}^{K-1} H_i'(z^M W^{Ml})F_i'(z^M) = A(zW^l)B(z)G_l'(z^M)$, where $G_l'(z) = \sum_{i=0}^{K-1} H_i'(zW_K^l)F_i'(z)$ (where $W_K = W^M = \exp(\frac{-j2\pi}{K})$). Thus, condition (a) implies that $G_l'(z) = 0$ for $l = 1, 2, \ldots, K-1$. (The alternative $A(zW^l)B(z) = 0$ is infeasible as it makes the systems identically zero.) Now the input–output relation of the leaf FB is $\hat{V}(z) = \frac{1}{K}\sum_{i=0}^{K-1} V(zW_K^l)G_l'(z)$ (analogous to (2.1.2)). Thus the leaf FB is LTI with (rational) transfer function $U(z) = G_0'(z)/K$. Hence, dividing all the $H_i'(z)$ by $U(z)$ and multiplying $A(z)$ by $U(z^M)$ gives a new system with all the properties desired in condition (b). This proves that (a) implies (b). $\bigtriangledown\bigtriangledown\bigtriangledown$

Appendix H: Proof of Theorem 8

It suffices to prove the result for 2–unit trees, as we can continue by induction. A general 2–unit tree is specifiable as follows: The triples of (analysis filter, synthesis filter, decimator) are $(H_i(z), F_i(z), m_i), i = 0, 1, \ldots, M-1$ for the root and $(A_i(z), B_i(z), k_i), i = 0, 1, \ldots, K-1$ for the leaf, which is attached to decimator m_0 of the root. Thus the filters allowing and requiring modification are H_0, F_0 and the leaf filters A_i, B_i. The overall FB is unaffected iff the modifications preserve all the products $H_0(z)A_i(z^{m_0})$ and $F_0(z)B_i(z^{m_0})$.

Realizing stability, FIR filters: Let all the $H_0(z)A_i(z^{m_0})$ be stable. Then for every unstable pole $z = p$ of $A_j(z)$, there are m_0 unstable poles in $A_j(z^{m_0})$, one at each m_0-th root of p. To cancel these, we must have $H_0(z) = H_0'(z)C(z^{m_0})$ where H_0, H_0' have the same set of poles and $C(z) = (1 - z^{-1}p)$, so that $C(z^{m_0})$ is FIR with m_0 zeroes at the right places. Hence, replacing H_0 by H_0' and the A_i by A_iC removes the unstable pole of A_j and preserves the analysis filters of the overall FB. Thus all A_i can be made stable. Similarly if H_0 has an unstable pole p, each $A_i(z^{m_0})$ must have a zero at p, and hence for each i, $A_i(z) = A_i'(z)(1 - p^{m_0}z^{-1})$ where A_i, A_i' have the same set of poles. Thus, replacing A_i by A_i' and $H_0(z)$ by $H_0(z)(1 - p^{m_0}z^{-m_0})$ removes the unstable pole of H_0. Thus all filters can be made stable while preserving the overall FB. Similarly, if all the $H_0(z)A_i(z^{m_0})$ are FIR, the above argument can be repeated for *all* poles (rather than just the unstable ones), and all analysis filters can be made FIR.

Realizing PR, orthonormality: If the overall FB has PR, from (2.2.1) we get

$$\begin{aligned} (H_0(z)A_i(z^{m_0})F_0(z)B_j(z^{m_0})) \downarrow_{\gcd(m_0k_i,m_0k_j)} &= \\ ((H_0(z)F_0(z)) \downarrow_{m_0} A_i(z)B_j(z)) \downarrow_{\gcd(k_i,k_j)} &= \delta(i-j) \qquad \text{(H.1)} \end{aligned}$$

With rational filters $X(z), Y(z)$ defined such that $XY = (H_0F_0)\downarrow_{m_0}$, let $A_i' = A_iX$, $B_i' = B_iY$ for all i. Thus from (H.1), $\left(A_i'(z)B_j'(z)\right)\downarrow_{\gcd(k_i,k_j)} = \delta(i-j)$, i.e., replacing each A_i by A_i' and B_i by B_i' causes the leaf FB to obey (2.2.1) and hence to have PR. The overall FB is preserved on replacing $H_0(z)$ by $H_0'(z) = H_0(z)/X(z^{m_0})$ and $F_0(z)$ by $F_0'(z) = F_0(z)/Y(z^{m_0})$. Since now both the leaf and the overall FB have PR, the root must have PR too. Thus the root and leaf have been modified as desired. Further if the overall FB is orthonormal, then it has PR with $F_0(z)B_i(z^{m_0}) = \widetilde{T}_i(z)$ where $T_i(z) = H_0(z)A_i(z^{m_0})$ (and of course, $F_i = \widetilde{H}_i$ for $i > 0$). Using $\widetilde{PQ} = \widetilde{P}\widetilde{Q}$, this means that (H.1) holds with F_0, B_i replaced by $\widetilde{H}_0, \widetilde{A}_i$ respectively. So we repeat with these substitutions, the earlier arguments used to make the root and leaf PR, and choose X such that $Y = \widetilde{X}$, i.e., such that $X\widetilde{X} = \left(H_0\widetilde{H}_0\right)\downarrow_{m_0} \stackrel{\triangle}{=} W(z)$. (This is possible by spectral factorization, as $W(z)$ is rational and $W(e^{j\omega}) \geq 0$.) This ensures that the root and leaf are modified to be PR with $F_0' = \widetilde{H_0'}$ and $B_i' = \widetilde{A_i'}$. In other words, for all FBs, PR is obeyed and the synthesis filter corresponding to a given analysis filter D is $\widetilde{D}$. Thus both the root and leaf have been modified to be orthonormal rational FBs.

References

[1] S. Akkarakaran and P.P. Vaidyanathan, New results and open problems on nonuniform filter banks, in *Proc. IEEE ICASSP*, Phoenix, AZ, Mar. 1999.

[2] T. Chen and L. Qiu, *General multirate building structures with application to nonuniform filter banks*, IEEE Trans. Ckts. Syst.–II, **45** (1998), 948-958.

[3] S. Dasgupta and A. Pandharipande, *On biorthogonal nonuniform filter banks*, preprint.

[4] I. Djokovic and P.P. Vaidyanathan, *Results on biorthogonal filter banks*, Appl. Comp. Harmonic Anal., **1** (1994), 329-343.

[5] P.-Q. Hoang and P.P. Vaidyanathan, Non-uniform multirate filter banks: Theory and design, in *Proc. IEEE ISCAS*, Portland, Oregon, May 1989, pp.371-374.

[6] J. Kovačević and M. Vetterli, *Perfect reconstruction filter banks with rational sampling factors*, IEEE Trans. Sig. Proc., **41** (1993), 2047-2066.

[7] J. Li, T.Q. Nguyen, and S. Tantaratana, *A simple design method for near-perfect-reconstruction nonuniform filter banks*, IEEE Trans. Sig. Proc., **45** (1997), 2105-2109.

[8] K. Nayebi, T.P. Barnwell,III, and M. Smith, *Nonuniform filter banks: A reconstruction and design theory*, IEEE Trans. Sig. Proc., **41** (1993), 1114-1127.

[9] R.G. Shenoy, *Multirate specifications via alias–component matrices*, IEEE Trans. Ckts. Syst.–II, **45** (1998), 314–320.

[10] A.K. Soman and P.P. Vaidyanathan, *On orthonormal wavelets and paraunitary filter banks*, IEEE Trans. Sig. Proc., **41** (1993), 1170-1183.

[11] P.P. Vaidyanathan, *Multirate Systems and Filter Banks*, Englewood Cliffs, NJ: Prentice-Hall, 1993.

[12] M. Vetterli and J. Kovačević, *Wavelets and Subband Coding*, Englewood Cliffs, NJ: Prentice-Hall, 1995.

Sony Akkarakaran and P.P. Vaidyanathan
Department of Electrical Engineering 136-93
California Institute of Technology
Pasadena, CA 91125
`{sony,ppvnath}@systems.caltech.edu`