Gröbner Bases and Multidimensional FIR Multirate Systems

Hyungju Park* Ton Kalker† Martin Vetterli‡

Abstract

The **polyphase representation** with respect to sampling lattices in multidimensional (M-D) multirate signal processing allows us to identify perfect reconstruction (PR) filter banks with unimodular Laurent polynomial matrices, and various problems in the design and analysis of invertible MD multirate systems can be algebraically formulated with the aid of this representation. While the resulting algebraic problems can be solved in one dimension (1-D) by the Euclidean Division Algorithm, we show that Gröbner bases offers an effective solution to them in the M-D case.

1 Introduction

It has been well known that the **polyphase representation** with respect to sampling lattices is a natural representation of multirate systems in studying their algebraic properties. As demonstrated in [1], it allows us to identify various problems in the design and analysis of invertible MD multirate systems with the following mathematical question:

> *Given a matrix of polyphase components, can we effectively decide whether or not that matrix has a left inverse, and give a complete parametrization of all the left inverses of that matrix?*

*Department of Mathematical Sciences, Oakland University, Rochester, MI 48309 (park@oakland.edu).

†Philips Research Laboratories, Prof. Holstlaan 4, 5656 AA, Eindhoven, The Netherlands (kalker@natlab.research.philips.com).

‡Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA 94720 (martin@eecs.berkeley.edu).

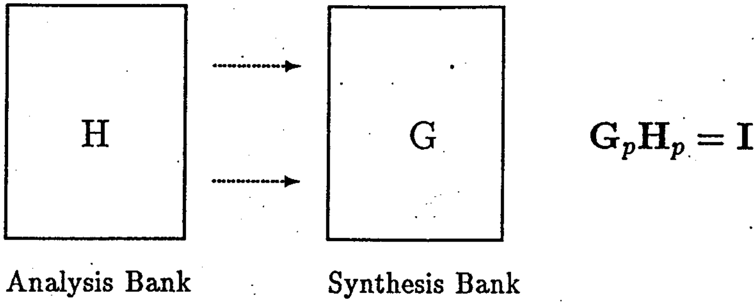

Figure 1: An analysis/synthesis system with PR property.

In this paper, based on the methods initiated in [1], we will further investigate this algebraically simplified problem, and all the systems henceforth will be assumed to be FIR (Finite Impulse Response).

Let us start by reminding the reader that the following three problems were proposed in [1] as demonstrating examples of this theme.

1. Given an MD FIR low-pass filter $G(z)$, decide effectively whether or not $G(z)$ can occur as an analysis filter in a critically downsampled, 2-channel, perfect reconstructing (PR) FIR filter bank. When this decision process yields a positive answer, find all such filter banks.

2. Given an oversampled MD FIR analysis filter bank, decide effectively whether or not there is an FIR synthesis filter bank such that the overall system is PR. When this decision process yields a positive answer, provide a complete parametrization of all such FIR synthesis filter banks[1].

3. Given a sample rate conversion scheme consisting of upsampling, filtering with an MD FIR filter, and downsampling, decide effectively whether or not this scheme is FIR invertible.

Some of these questions have been studied in 1-D multirate systems (see [3], [4], [5], [2], and [6]), in which the Euclidean Division Algorithm plays a central role. In the M-D case, however, the questions are substantially harder to answer, and it is the goal of this paper to show how Gröbner bases can be used to effectively answer them.

While comprehensive accounts of this theory can be found in [7], [8], [9], [10], [11] and [12], a short review of the Gröbner bases theory is presented

[1]This is often called the **complementary filter** problem [2].

in the following section. A heuristic review of this theory can also be found in [13].

2 Gröbner Bases

In order to define Gröbner basis, we first have to introduce the notion of *monomial order.* A monomial in $\mathbb{C}[\boldsymbol{x}] = \mathbb{C}[x_1, \ldots, x_m]$ is a power product of the form $x_1^{e_1} \cdots x_m^{e_m}$, and we denote by $T(x_1, \ldots, x_m)$, or simply by T, the set of all monomials in these variables. In the univariate case, there is a natural monomial order, that is,

$$1 < x < x^2 < x^3 < \cdots.$$

In the multivariate case, we define a monomial order $\leq$ to be a linear order on T satisfying the following two conditions.

1. $1 \leq t$ for all $t \in T$.

2. $t_1 \leq t_2$ implies $t_1 \cdot s \leq t_2 \cdot s$ for all $s, t_1, t_2 \in T$.

Once a monomial order is given, we can talk about the leading monomial (or leading term), $\mathrm{lt}(f(\boldsymbol{x}))$, of $f(\boldsymbol{x}) \in \mathbb{C}[\boldsymbol{x}]$. It should be noted that, if we change the monomial order, then we may have a different $\mathrm{lt}(f(\boldsymbol{x}))$ for the same $f(\boldsymbol{x})$. Now, fix a monomial order on T, and let $I \subset \mathbb{C}[\boldsymbol{x}]$ be an ideal (i.e a set which includes all the elements which it can generate by taking linear combinations). Define $\mathrm{lt}(I)$ by

$$\mathrm{lt}(I) \quad = \quad \{\mathrm{lt}(f(\boldsymbol{x})) \mid f \in I\}.$$

Definition 2.1 *$\{f_1(\boldsymbol{x}), \ldots, f_t(\boldsymbol{x}\} \subset I$ is called a* **Gröbner basis** *of I if*

$$(\mathrm{lt}(f_1(\boldsymbol{x})), \ldots, \mathrm{lt}(f_t(\boldsymbol{x}))) \quad = \quad \mathrm{lt}(I)$$

i.e. if the ideal generated by $\mathrm{lt}(f_1(\boldsymbol{x})), \ldots, \mathrm{lt}(f_t(\boldsymbol{x})$ *coincides with* $\mathrm{lt}(I)$.

Example 2.2 Fix the degree lexicographic order on $\mathbb{C}[x, y]$, and let I be the ideal generated by $(f(\boldsymbol{x}), g(\boldsymbol{x}))$, with $f(\boldsymbol{x}) = 1 - xy$ and $g(\boldsymbol{x}) = x^2$. Then the relation

$$(1 + xy)f(\boldsymbol{x}) + y^2 g(\boldsymbol{x}) = 1$$

implies that $I = \mathbb{C}[x, y]$, and therefore $\mathrm{lt}(I) = \mathbb{C}[x, y]$. But

$$(\mathrm{lt}(f(\boldsymbol{x})), \mathrm{lt}(g(\boldsymbol{x}))) = (-xy, x^2) \subset (x).$$

Therefore, $\{f(\boldsymbol{x}), g(\boldsymbol{x})\}$ is not a Gröbner basis of the ideal I. □

The main reason that Gröbner basis is useful for us comes from the following analogue of the Euclidean division algorithm.

Theorem 2.3 *(Division Algorithm)*
Let $\{\mathrm{lt}(f_1(\boldsymbol{x})), \ldots, \mathrm{lt}(f_t(\boldsymbol{x}))\} \subset \mathbb{C}[\boldsymbol{x}]$ *be a Gröbner basis w.r.t. a fixed monomial order, and let* $h(\boldsymbol{x}) \in \mathbb{C}[\boldsymbol{x}]$. *Then there is an algorithm for writing* $h(\boldsymbol{x})$ *in the form*

$$h(\boldsymbol{x}) \quad = \quad \lambda_1(\boldsymbol{x}) f_1(\boldsymbol{x}) + \cdots + \lambda_t(\boldsymbol{x}) f_t(\boldsymbol{x}) + r(\boldsymbol{x})$$

such that $h(\boldsymbol{x}) \in I$ *if and only if* $r(\boldsymbol{x}) = 0$.

The polynomial $r(\boldsymbol{x})$ in the above is called the normal form of $f(\boldsymbol{x})$ w.r.t. $\{\mathrm{lt}(f_1(\boldsymbol{x})), \ldots, \mathrm{lt}(f_t(\boldsymbol{x}))\}$. Now, in order to solve our problem, we just compute the normal form of $h(\boldsymbol{x})$ w.r.t. the given set of polynomials (assuming that this set is a Gröbner basis. Otherwise, we first have to transform it to another set of polynomials which is a Gröbner basis. There is a standard algorithm for this transformation). If it is 0, then $h(\boldsymbol{x})$ can be written as a linear combination of the polynomials $f_i(\boldsymbol{x})$ and we have at the same time found the polynomials $\lambda_i(\boldsymbol{x})$.

Remark 2.4 There are some results known on the complexity of Gröbner bases computation. If we let

r = # of variables
d = the maximum degree of the polynomials
s = the degree of the Hilbert polynomial (this is one less than the dimension, and is between 0 and $r-1$)
b = the **worst case** upper bound for the degree of the elements of the Gröbner basis (of the given polynomials),

then it is known that

$$b = ((r+1)(d+1)+1)^{(2^{s+1})(r+1)},$$

i.e. is potentially doubly exponential in the number of variables.

This estimate is so large that it seems to suggest that Gröbner bases would be useless in practice. Fortunately, this is not at all the case, and the algorithm (in actual use) terminates quite quickly on very many problems of interest. There is a partial understanding of why this is so, and various other bounds are known in some special cases. It is also known that the monomial order being used for the computation affects the complexity, i.e. you have to choose a good monomial order in order to shorten the computation time.

Reverse degree lexicographic order behaves particularly well in many cases. The papers [14], [15], [16], and [17] contain some results on this complexity issue.

Remark 2.5 One of the reviewers of this paper asked to include some results on the issue of sensitivity of round-off errors to Gröbner bases computations. We must however note that we are not aware of any results in this direction. The importance of this issue is however recognized by the authors.

3 Unimodularity and Perfect Reconstruction

Since the polyphase representation of an FIR system always gives rise to a rectangular matrix with **Laurent polynomial** entries (see below for a definition), our problem is essentially reduced to the following:

Find left inverses to Laurent polynomial matrices.

Definition 3.1 *Let k be one of the sets $\mathbb{Q}, \mathbb{R}, \mathbb{C}$.*

1. *A* **Laurent polynomial** *f over k in m variables $x_1, \ldots, x_m$ is an expression of the form*

$$f(x_1, \ldots, x_m) = \sum_{i_1=l_1}^{d_1} \cdots \sum_{i_m=l_m}^{d_m} a_{i_1 \cdots i_m} x_1^{i_1} \cdots x_m^{i_m},$$

where $l_1 \leq d_1, \ldots, l_m \leq d_m$ are all integers (positive or negative) and each $a_{i_1 \cdots i_m}$ is an element of k.

2. *The set of all the Laurent polynomials over k in $x_1, \ldots, x_m$ is called a* **Laurent polynomial ring,** *and denoted by*

$$k[x^{\pm 1}] := k[x_1^{\pm 1}, \ldots, x_m^{\pm 1}].$$

In order to see why a Laurent polynomial (rather than a polynomial) arises naturally, consider a filter with frequency response

$$H(\omega) = 2\sin(\omega) - 3\cos(2\omega).$$

Then, letting $z := e^{iw}$, we get

$$\begin{aligned} H &= 2\frac{e^{iw}-e^{-iw}}{2} - 3\frac{e^{2iw}+e^{-2iw}}{2} \\ &= -\frac{3e^{-2iw}}{2} - e^{-iw} + e^{iw} - \frac{3e^{2iw}}{2} \\ &= -\frac{3}{2z^2} - \frac{1}{z} + z - \frac{3z^2}{2}, \end{aligned}$$

which is a Laurent polynomial in z.

Definition 3.2 *Let $R := k[\boldsymbol{x}^{\pm 1}]$ be a Laurent polynomial ring.*

1. *Let $\mathbf{v} = (v_1, \ldots, v_n)^t \in R^n$ for some $n \in \mathbb{N}$. Then $\mathbf{v}$ is called a* **unimodular column vector** *if its components generate R, i.e. if there exist $g_1, \ldots, g_n \in R$ such that $v_1g_1 + \cdots + v_ng_n = 1$.*

2. *A matrix $\mathbf{A} \in \mathrm{M}_{pq}(R)$ is called a* **unimodular matrix** *if its maximal minors generate R.*

Remark 3.3 When $R = \mathbb{C}[\boldsymbol{x}] := \mathbb{C}[x_1, \ldots, x_m]$ is a polynomial ring over $\mathbb{C}$ and the polynomials $v_1, \ldots, v_n \in R$ do not have a common root, Hilbert Nullstellensatz states that there always exist $g_1, \ldots, g_n \in R$ such that $v_1g_1 + \cdots + v_ng_n = 1$, i.e. $\mathbf{v} = (v_1, \ldots, v_n)^t \in R^n$ is unimodular. In this case, Gröbner bases theory offers a way to find such g_i's (see [7]).

Example 3.4 Consider the polynomial matrix $\mathbf{H}_p$ given by

$$\mathbf{H}_p = \begin{pmatrix} xy - y + 1 & 1 - x \\ yz + w & -z \\ -y & 1 \end{pmatrix} \in \mathrm{M}_{32}(k[x, y]).$$

Computing (the determinants of) the maximal minors, we find $D_{12}(\boldsymbol{x}) = -w + wx - z$, $D_{13}(\boldsymbol{x}) = -1$ and $D_{23}(\boldsymbol{x}) = w$. Since $0 \cdot D_{12}(\boldsymbol{x}) + (-1)D_{13}(\boldsymbol{x}) + 0 \cdot D_{23}(\boldsymbol{x}) = 1$, $\mathbf{H}_p$ is trivially unimodular. □

Now the following important result simplifies our problem significantly.

Theorem 3.5 *A $p \times q$ Laurent polynomial matrix ($p \geq q$) has a left inverse if and only if it is unimodular.*

A proof of this assertion in the case of polynomial matrices can be found in [18], and this result was extended to the case of Laurent polynomial matrices in [19]. An immediate corollary of this theorem is,

Corollary 3.6 *An M-D FIR filter bank can be the analysis portion of an M-D PR filter bank if and only if its polyphase matrix is a unimodular Laurent polynomial matrix.*

Example 3.7 Consider the polyphase matrix $\mathbf{H}_p$ of the Example 3.4. Since it was shown to be unimodular, it has to have a left inverse. Actually, one verifies easily that

$$\mathbf{G}_p := \begin{pmatrix} 1 & 0 & x-1 \\ y & 0 & xy-y+1 \end{pmatrix} \in \mathrm{M}_{23}(k[x,y])$$

satisfies $\mathbf{G}_p\mathbf{H}_p = \mathbf{I}$. This left inverse, however, is far from being unique. On the contrary, a computation using Gröbner bases shows (see [19, page 30–31, page 116]) that an arbitrary matrix of the form,

$$\begin{pmatrix} 1 & 0 & x-1 \\ y & 0 & xy-y+1 \end{pmatrix} + \begin{pmatrix} u_1 w & -u_1 & u_1(xw-z-w) \\ u_2 w & -u_2 & u_2(xw-z-w) \end{pmatrix}$$

for any Laurent polynomials $u_1, u_2 \in k[x^{\pm 1}, y^{\pm 1}]$, is also a left inverse of $\mathbf{H}_p$. Even more strikingly, this parametrization of the left inverses of $\mathbf{H}_p$ in terms of the two parameters u_1 and u_2 turns out to be complete (i.e. exhaustive) and canonical (i.e. minimal with unique representation).

Therefore, the analysis filter bank H whose polyphase matrix is $\mathbf{H}_p$ is FIR invertible, and together with the synthesis filter bank obtained by a backward polyphase superposition of $\mathbf{G}_p$, makes a PR FIR filter bank. And the above parametrization in terms of the two Laurent polynomial parameters $u_1, u_2 \in k[x^{\pm 1}, y^{\pm 1}]$ gives a complete and canonical parametrization of the complementary FIR filter banks. Thus, the degree of freedom with which we can design a PR pair of H is precisely 2. □

Therefore, mathematically, we are dealing with the problem of determining if a given **Laurent polynomial** matrix is unimodular, and in case it is, if we can explicitly find all the (not unique in non-square cases) left inverses for it. This allows us to see the study of perfect reconstructing FIR filter banks as the study of unimodular matrices over Laurent polynomial rings.

4 Causal FIR systems and General FIR systems

Many of the known methods for unimodular matrices are developed mainly over polynomial rings, i.e. when the matrices involved are unimodular **polynomial** matrices rather than Laurent polynomial matrices. In system theoretic terminology, causal invertibility of causal filters are therefore covered by these methods. Geometrically, this demonstrates the relative simplicity associated with affine systems compared to toric systems.

The situation, however, is more complicated partly because an FIR-invertible causal filter may not be causal-invertible. For an example, consider the polynomial vector $\begin{pmatrix} z \\ z^2 \end{pmatrix} \in (k[z])^2$. While the relation $\frac{1}{2z} \cdot z + \frac{1}{2z^2} \cdot z^2 = 1$ clearly shows the FIR-invertibility of this vector, it is not causal-invertible since there are no polynomials $f(z), g(z) \in k[z]$ satisfying

$$f(z) \cdot z + g(z) \cdot z^2 = 1$$

as we can see easily by evaluating both sides at $z = 0$.

Now, in order to extend any affine results (i.e. causal cases) to general FIR systems, we need an effective process of converting a given Laurent polynomial matrix to a polynomial matrix while preserving unimodularity i.e. we have to perform a preparatory process to convert the problems to causal problems.

We already have presented a systematic method to this effect in [1]: for every variable z_i we introduce two new variables x_i and y_i. Substituting x_i^m for every positive power z_i^m and y_i^k for every negative power z_i^{-k}, we transform the original set of Laurent polynomials into a set of regular polynomials. We then enlarge this set by adding the polynomials $x_i y_i - 1$. One verifies that the constant 1 is a linear combination of the original set of Laurent polynomials if and only if the same is true for the constructed set of regular polynomials. Moreover, given a linear combination of polynomials , we find a linear combination of Laurent polynomials by back substitution: x_i and y_i are replaced by z_i and z_i^{-1} respectively.

There are, however, some drawbacks with this method. First, it significantly increases the complexity of the problem by introducing extra variables and by enlarging the size of the given polynomial vector. Also, a complete parametrization of solutions needs separate computation.

To remedy the situation, an alternative systematic process for the same purpose was developed in [19], and was named the **LaurentToPoly** Algorithm. An input-output description of this algorithm is given in the box of Figure 2. An overview of the main ingredients of this algorithm is presented in the Appendix (see [19] for a complete description of this algorithm).

In this paper, we will mainly rely on this result to reduce the FIR problems to causal FIR problems. A graphical demonstration of this process is shown in the Figure 3.

Input:	$\mathbf{v}(x) \in (k[x^{\pm 1}])^n$, a Laurent polynomial column vector
Output:	$x \to y$, a change of variables $\mathbf{T}(x) \in \mathrm{GL}_n(k[x^{\pm 1}])$, a square unimodular Laurent polynomial matrix
Specification:	(1) $\hat{\mathbf{v}}(y) := \mathbf{T}(x)\mathbf{v}(x) \in (k[y])^n$ is a polynomial column vector in the new variable y (2) $\mathbf{v}(x)$ is unimodular over $k[x^{\pm 1}]$ if and only if $\hat{\mathbf{v}}(y)$ is unimodular over $k[y]$

Figure 2: Algorithm **LaurentToPoly**

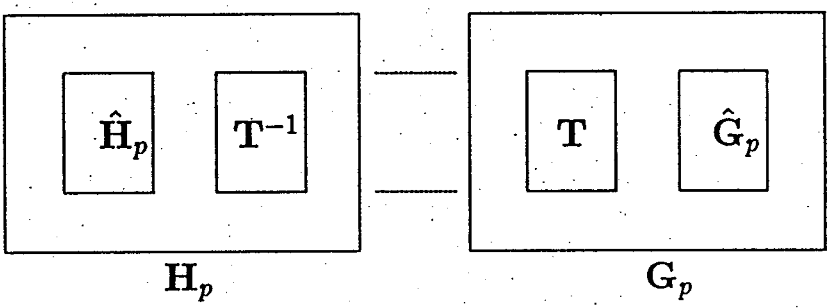

Figure 3: Conversion of an FIR system H to a causal system $\hat{H}$

Now finding an FIR inverse G to the given FIR filter H is equivalent to finding a causal inverse $\hat{G}$ to the causal filter $\hat{H}$.

Example 4.1 Consider the unimodular Laurent polynomial vector

$$\mathbf{v} = (z, z^2)^t \in k[z^{\pm 1}].$$

As was pointed out in the Section 4, this vector is not unimodular as a polynomial vector. If we let the transformation matrix $\mathbf{T}$ be

$$\mathbf{T} = \begin{pmatrix} 1/z & 0 \\ 0 & z \end{pmatrix},$$

then the converted vector, $\hat{\mathbf{v}} = \mathbf{T}\mathbf{v} = (1, z^3)^t$, is unimodular as a polynomial vector. □

Remark 4.2 At this point, we would like to make a remark on the opinion expressed in [20]. In that paper, it is stated that the class of 1D *cafacafi* systems is more tractable than the more general class of FIR systems with FIR inverse. The **LaurantToPoly** algorithm, however, shows that this is not necessarily true: using **LaurentToPoly**, any invertible FIR system is translated into the mathematically well understood domain of invertible polynomial matrices. At the same time, **LaurentToPoly** is also applicable in the MD case.

5 1-D Case

In this section, we will present a complete solution to our problem in the transparent one-dimensional (1-D) setup. And our main tool in this section will be the Euclidean Division Algorithm for the univariate polynomial ring.

5.1 Determination of 1-D FIR Invertibility

Let H be an FIR filter bank whose polyphase matrix is $\mathbf{H}_p \in \mathrm{M}_{st}(k[x^{\pm 1}])$, a $s \times t$ univariate Laurent polynomial matrix ($s \geq t$). Suppose H is FIR invertible. Then, since a Laurent polynomial matrix has a left inverse if and only if it is unimodular, there exist $l := \binom{s}{t}$ Laurent polynomials $D_i(x)$ such that

$$\sum_{i=1}^{l} D_i(x) M_i(x) = 1, \qquad (1)$$

where $M_i(x)$ ranges over the maximal minors of $\mathbf{H}_p$.

So, determining the FIR invertibility of H is equivalent to determining the unimodularity of the Laurent polynomial matrix $\mathbf{H}_p$, or equivalently, the unimodularity of the Laurent polynomial vector $(M_1, \ldots, M_l) \in (k[x^{\pm 1}])^l$.

When $k = \mathbb{C}$, this unimodularity determination problem can be readily solved once we notice that, due to the the Laurent polynomial analogue of Hilbert Nullstellensatz over $\mathbb{C}$, $\sum_i D_i(x) M_i(x) = 1$ is possible if and only if the Laurent polynomials $M_i(x)$'s, $1 \leq i \leq \binom{s}{t}$, have no nonzero common roots, i.e. no roots in $\mathbb{C}^*$. Since each univariate Laurent polynomial $M_i(x)$ has only finitely many zeros which can be explicitly found using any existing computer algebra packages, we can tell if $M_i(x)$'s have a nonzero common root or not, and thereby determining if $\mathbf{H}_p$ is unimodular.

Example 5.1 Consider a sample rate conversion scheme consisting of upsampling by $p = 3$, filtering with an FIR filter $U(z)$ and downsampling by $q = 2$, where $U(z)$ is given by

$$\begin{aligned} U(z) = {} & \frac{3}{z^6} + \frac{6}{z^5} + \frac{6}{z^3} + \frac{3}{z^2} - 2 + 29z + 25z^3 + 2z^5 - 2z^6 - 4z^7 + 2z^8 - \\ & 23z^9 - 2z^{10} + 4z^{11} + 2z^{12} - 20z^{13} - 16z^{15} + 20z^{17} + 20z^{21}. \end{aligned}$$

Then we get the polyphase decomposition $U(z) = \sum_{i=0}^{5} z^i U_i(z^6)$ of $U(z)$ where $U_i(z)$'s are found as

$$\begin{aligned} U_0(z) &= \frac{3}{z} - 2 - 2z + 2z^2 \\ U_1(z) &= \frac{6}{z} + 29 - 4z - 20z^2 \\ U_2(z) &= 2z \\ U_3(z) &= \frac{6}{z} + 25 - 23z - 16z^2 + 20z^3 \\ U_4(z) &= \frac{3}{z} - 2z \\ U_5(z) &= 2 + 4z + 20z^2. \end{aligned}$$

Now, as demonstrated in [1], the FIR invertibility of the given scheme is equivalent to the FIR invertibility of the following polynomial matrix:

$$\mathbf{U} = \begin{pmatrix} U_0(z) & U_3(z) \\ U_4(z) & U_1(z) \\ U_2(z) & U_5(z) \end{pmatrix}.$$

The three maximal minors of $\mathbf{U}$ are

$$\begin{aligned} M_1(z) &= \begin{vmatrix} U_0(z) & U_3(z) \\ U_4(z) & U_1(z) \end{vmatrix} = -1 \\ M_2(z) &= \begin{vmatrix} U_0(z) & U_3(z) \\ U_2(z) & U_5(z) \end{vmatrix} = \frac{6}{z} - 4 - 2z + 2z^2 \\ M_3(z) &= \begin{vmatrix} U_4(z) & U_1(z) \\ U_2(z) & U_5(z) \end{vmatrix} = \frac{6}{z} - 2z \end{aligned}$$

which obviously don't have any common roots.

Consequently the given scheme is FIR invertible. □

5.2 Parametrization of 1-D PR Pairs

Let the polyphase matrix be $\mathbf{A}$, a $p \times q$ Laurent polynomial matrix, $p \geq q$. Since this polyphase matrix has a left inverse if and only if it is unimodular, we can first determine its unimodularity by the method outlined in the above. If this test shows the unimodularity of $\mathbf{A}$, we first apply the algorithm **LaurentToPoly** to $\mathbf{A}$, converting $\mathbf{A}$ to a unimodular polynomial matrix $\hat{\mathbf{A}}$. Then, by using Euclidean Division Algorithm, we apply a succession of elementary operations to $\hat{\mathbf{A}}$ to reduce it to the following $p \times q$ matrix

$$\begin{pmatrix} \mathbf{I}_q \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix} \in \mathrm{M}_{pq}(k),$$

where $\mathbf{I}_q$ is the $q \times q$ identity matrix, and $\mathbf{0}$ is the q-dimensional zero row vector.

This means that we can find $\mathbf{E} \in E_p(k[z])$, a product of elementary matrices, such that

$$\mathbf{EA} = \begin{pmatrix} \mathbf{I}_q \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix}.$$

Now take the first q rows of $\mathbf{E}$ to make a $q \times p$ matrix $\mathbf{F}$, i.e.

$$\mathbf{F} := (\mathbf{I}_q, \mathbf{0}, \ldots, \mathbf{0})\mathbf{E}.$$

Then $\mathbf{F}$ is a desired left inverse of $\mathbf{A}$. Note here that $\mathbf{A} = \mathbf{E}^{-1} \begin{pmatrix} \mathbf{I}_q \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix}$ implies $\mathbf{E}^{-1} \in \mathrm{GL}_p(k[z])$ is a unimodular completion of $\mathbf{A}$.

To get a complete parametrization of all the possible left inverses of $\mathbf{A}$, let $\mathbf{B} \in \mathrm{M}_{qp}(k[z])$ be an arbitrary left inverse of $\mathbf{A}$. Then

$$\mathbf{BA} = \mathbf{I}_q.$$

Now, since $\mathbf{E}^{-1}$ is a unimodular completion of $\mathbf{A}$,

$$\mathbf{BE}^{-1} = (\mathbf{I}_q, \mathbf{u}_1, \ldots, \mathbf{u}_{p-q})$$

for some $\mathbf{u}_1, \ldots, \mathbf{u}_{p-q} \in (k[z^{\pm 1}])^q$. Now, regarding $\mathbf{u}_1, \ldots, \mathbf{u}_{p-q}$ as free parameters ranging over q-dimensional Laurent polynomial vectors, we get a complete parametrization of the left inverses to $\mathbf{A}$ in terms of $(p-q)q$ parameters ranging over the Laurent polynomials in $k[z^{\pm 1}]$:

$$\mathbf{B} = (\mathbf{I}_q, \mathbf{u}_1, \ldots, \mathbf{u}_{p-q})\mathbf{E}. \tag{2}$$

Remark 5.2 If $p = q$, i.e. if the polyphase matrix $\mathbf{A}$ is a square unimodular matrix, then the number of free parameters is $(p-q)q = 0$. This coincides with the fact that a square unimodular matrix has a unique inverse.

Example 5.3 Consider an oversampled 1-D FIR analysis filter bank whose polyphase matrix is the matrix $\mathbf{U}$ of the Example 5.1. We already saw in that example that

$$\mathbf{U} = \begin{pmatrix} \frac{3}{z} - 2 - 2z + 2z^2 & \frac{6}{z} + 25 - 23z - 16z^2 + 20z^3 \\ \frac{3}{z} - 2z & \frac{6}{z} + 29 - 4z - 20z^2 \\ 2z & 2 + 4z + 20z^2 \end{pmatrix}$$

is unimodular, so there is an FIR synthesis filter bank such that the overall system is PR. Now we want to find all such FIR synthesis filter banks.

Closely following the algorithm outlined in the above, we get

$$\mathbf{EU} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix},$$

where the 3×3 matrix $\mathbf{E}$ is found as

$$\begin{pmatrix} \frac{z}{18}(-18-125z-188z^2+252z^3-215z^4+178z^5+6z^6) & \frac{z}{3}(-2-27z+30z^2+z^3) & \frac{(-12-89z+51z^2-60z^3-2z^4)}{6} \\ \frac{z}{6}(3+19z-32z^2+23z^3-9z^4-8z^5+6z^6) & z(4-3z-z^2+z^3) & \frac{9}{2}-4z+\frac{3z^2}{2}+z^3-z^4 \\ z(-4z+\frac{23z^2}{3}-5z^3+z^4+\frac{8z^5}{3}-2z^6) & 2z(-3+2z+z^2-z^3) & -6+6z-z^2-2z^3+2z^4 \end{pmatrix}.$$

Now a general left inverse of $\mathbf{U}$ is in the form

$$\begin{pmatrix} 1 & 0 & u \\ 0 & 1 & v \end{pmatrix} \mathbf{E},$$

where u, v are arbitrary Laurent polynomials in $k[z^{\pm 1}]$. □

Now we consider a real world problem.

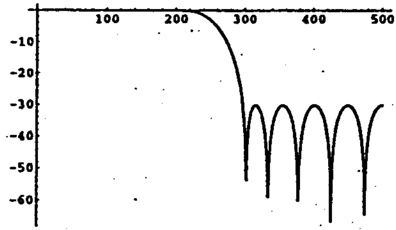

Figure 4: Frequency Response of the Lowpass Filter $H(z)$

Example 5.4 Consider a causal lowpass filter $H(z)$ given by

$$\begin{aligned} H(z) &= 0.1605+0.4156z+0.4592z^2+0.1487z^3-0.1643z^4-0.1245z^5+0.0825z^6+ \\ &\quad 0.0887z^7-0.0508z^8-0.0608z^9+0.0351z^{10}+0.0399z^{11}-0.0256z^{12}- \\ &\quad 0.0244z^{13}+0.0186z^{14}+0.0135z^{15}-0.0131z^{16}-0.0074z^{17}+0.0129z^{18}-0.0050z^{19} \end{aligned}$$

whose lowpass characteristic is shown in the Figure 4.

This is decomposed into polyphase components as

$$H(z) = H_0(z^2) + zH_1(z^2),$$

where $H_0(z)$ and $H_1(z)$ are

$$\begin{aligned} H_0(z) &= 0.1605+0.4592z-0.1643z^2+0.0825z^3-0.0508z^4+0.0351z^5-0.0256z^6+ \\ &\quad 0.0186z^7-0.0131z^8+0.0129z^9, \\ H_1(z) &= 0.4156+0.1487z-0.1245z^2+0.0887z^3-0.0608z^4+0.0399z^5-0.0244z^6+ \\ &\quad 0.0135z^7-0.0074z^8-0.0050z^9. \end{aligned}$$

the Euclidean Division yields

$$H_0(z) = -2.5893H_1(z) + r(z)$$

with the remainder

$$\begin{aligned} r(z) &= 1.2367+0.8442z-0.4867z^2+0.3123z^3-0.208349z^4+0.138472z^5-0.0888109z^6+ \\ &\quad 0.0536797z^7-0.0323696z^8. \end{aligned}$$

Carrying out the corresponding elementary operation gives

$$\mathbf{E}_{12}(2.5893)\begin{pmatrix} H_0(z) \\ H_1(z) \end{pmatrix} = \begin{pmatrix} r(z) \\ H_1(z) \end{pmatrix}.$$

Repeating the same procedure to the polynomial vector $\begin{pmatrix} r(z) \\ H_1(z) \end{pmatrix}$, we eventually get $\mathbf{C} \in \mathrm{E}_2(\mathbb{C}[z])$, a product of 10 elementary matrices, such that

$$\mathbf{C}\begin{pmatrix} H_0(z) \\ H_1(z) \end{pmatrix} = \begin{pmatrix} 0.7661 \\ 0 \end{pmatrix}.$$

Let $\mathbf{E} := \frac{1}{0.7661}\mathbf{C}$. Then

$$\mathbf{E}\begin{pmatrix} H_0(z) \\ H_1(z) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

An explicit computation shows

$$\begin{aligned}
E_{11}(z) &= -0.4138+0.5743z-0.3989z^2+0.2652z^3-0.1667z^4+0.0960z^5-0.0478z^6+\\
&\quad 0.0164z^7+0.0154z^8\\
E_{12}(z) &= 2.5658-0.6827z+0.3689z^2-0.2369z^3+0.1658z^4-0.1189z^5+0.0839z^6-\\
&\quad 0.0572z^7+0.0398z^8\\
E_{21}(z) &= -0.7081-0.2533z+0.2121z^2-0.1512z^3+0.1036z^4-0.0679z^5+0.0416z^6-\\
&\quad 0.0231z^7+0.0127z^8+0.0085z^9\\
E_{22}(z) &= 0.2735+0.7824z-0.2799z^2+0.1406z^3-0.0865z^4+0.0599z^5-0.0436z^6+\\
&\quad 0.0317z^7-0.0223z^8+0.0220z^9.
\end{aligned}$$

Now by the parametrization formula (2), any left inverse $\mathbf{B}$ is of the form

$$\begin{aligned}
\mathbf{B}(u) &= (1 \quad u)\mathbf{E}\\
&= (E_{11}+uE_{21} \quad E_{12}+uE_{22})\\
&= (E_{11} \quad E_{12}) + u(E_{21} \quad E_{22}),
\end{aligned}$$

where $u \in k[z^{\pm 1}]$ is an arbitrary Laurent polynomial. Now, the 1 parameter family of filters

$$F(z,u(z)) = B_{11}(z^2,u(z^2)) + zB_{12}(z^2,u(z^2)), \quad u \in k[z^{\pm 1}]$$

describes all the synthesis filters, and making a good choice of $u \in k[z^{\pm 1}]$ will give us a synthesis filter with a more desirable frequency response. □

6 Gröbner Bases and M-D FIR Systems

Let an $s \times t$ matrix $\mathbf{H}_p \in \mathrm{M}_{st}(k[x^{\pm 1}])$ be the polyphase matrix of a given M-D FIR filter bank H $(s \geq t)$. Now, suppose we want to find a synthesis filter bank G so that G together with H makes a perfect reconstructing system.

Applying the **LaurentToPoly** Algorithm to $\mathbf{H}_p \in M_{ts}(k[x^{\pm 1}])$ to obtain $\hat{\mathbf{H}}_p \in M_{ts}(k[x])$, we see that this problem is equivalent to finding a $t \times s$ matrix $\hat{\mathbf{G}}_p \in M_{ts}(k[x])$ such that $\hat{\mathbf{G}}_p\hat{\mathbf{H}}_p = \mathbf{I}_q$. After getting such a $\hat{\mathbf{G}}_p$, we can apply the **LaurentToPoly** Algorithm *backwards* to $\hat{\mathbf{G}}_p$ to obtain $\mathbf{G}_p \in M_{ts}(k[x^{\pm 1}])$. Then it follows that $\mathbf{G}_p\mathbf{H}_p = \mathbf{I}_q$, and the filter bank G with its polyphase matrix being $\mathbf{G}_p$ is a desired synthesis filter bank.

So, we have reduced our problem to

> For a given **polynomial matrix** $\mathbf{A} \in M_{st}(k[x])$ $(s \geq t)$, find a (particular) left inverse $\mathbf{B} \in M_{ts}(k[x])$ of $\mathbf{A}$.

This is in fact possible using Gröbner bases, and the following method was introduced in [18], and was exploited in our context in [19]:

The column vectors of the unimodular matrix $\mathbf{A}^t = (f_{ji}) \in M_{st}(k[x])$ span the free $k[x]$-module $(k[x])^q$. Therefore, we can use Gröbner bases to express the standard basis vectors $\mathbf{e}_1, \ldots \mathbf{e}_q \in (k[x])^q$ as linear combinations (with polynomial coefficients) of the column vectors of $\mathbf{A}^t$.

More explicitly, denoting the i-th column vector of $\mathbf{A}^t$ by $\mathbf{w}_i$, $1 \leq i \leq p$, we have $\mathbf{w}_i := \begin{pmatrix} f_{i1} \\ \vdots \\ f_{iq} \end{pmatrix}$. Now, use Gröbner bases to find g_{ij}'s such that

$$\mathbf{e}_1 := \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = g_{11}\mathbf{w}_1 + \cdots + g_{1p}\mathbf{w}_p = g_{11}\begin{pmatrix} f_{11} \\ \vdots \\ f_{1q} \end{pmatrix} + \cdots + g_{1p}\begin{pmatrix} f_{p1} \\ \vdots \\ f_{pq} \end{pmatrix}$$

$$\vdots$$

$$\mathbf{e}_q := \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} = g_{q1}\mathbf{w}_1 + \cdots + g_{qp}\mathbf{w}_p = g_{q1}\begin{pmatrix} f_{11} \\ \vdots \\ f_{1q} \end{pmatrix} + \cdots + g_{qp}\begin{pmatrix} f_{p1} \\ \vdots \\ f_{pq} \end{pmatrix}.$$

Denoting the $q \times p$ matrix (g_{ij}) by $\mathbf{B}$, we can rewrite the above set of equations as

$$\mathbf{I}_q = \begin{pmatrix} g_{11} & \cdots & g_{1p} \\ \vdots & & \vdots \\ g_{q1} & \cdots & g_{qp} \end{pmatrix}\begin{pmatrix} f_{11} & \cdots & f_{1q} \\ \vdots & & \vdots \\ f_{p1} & \cdots & f_{pq} \end{pmatrix} = \mathbf{BA}.$$

And this **B** is precisely (one of) what we want.

Example 6.1 Again, consider the polyphase matrix

$$\mathbf{H}_p = \begin{pmatrix} xy - y + 1 & 1 - x \\ yz + w & -z \\ -y & 1 \end{pmatrix} \in \mathrm{M}_{32}(k[x, y])$$

of Example 3.4. The following is a SINGULAR[2] script implementing the algorithm of this section to find a $\mathbf{G}_p \in \mathrm{M}_{23}(k[x, y])$ such that $\mathbf{G}_p\mathbf{H}_p = \mathbf{I}_2$.

```
ring r=0,(x,y,z,w),(c,dp);option(redSB);
vector v(1)=[xy-y+1,1-x];vector v(2)=[yz+w,-z];
vector v(3)=[-y,1];
module M=v(1),v(2),v(3);
module G=std(M); matrix T=lift(M,G);
```

And the results are

```
> G;
G[1]=[0,1]
G[2]=[1]

> T;
T[1,1]=y
T[1,2]=1
T[2,1]=0
T[2,2]=0
T[3,1]=xy-1y+1
T[3,2]=x-1
```

Since $\{(1,0), (0,1)\}$ is a Gröbner basis of the row vectors of $\mathbf{H}_p$, $\mathbf{H}_p$ is unimodular, and the relation G = MT translates to

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \mathbf{H}_p^t \mathrm{T}.$$

[2]SINGULAR is a computer algebra system for singularity theory and algebraic geometry, developed in the University of Kaiserslautern, Germany. It is being alpha-tested, and is freely available by anonymous ftp. For more information, see [21].

By taking transpose of both sides, we get $T^t \mathbf{H}_p = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, i.e.

$$\begin{pmatrix} 1 & 0 & x-1 \\ y & 0 & xy-y+1 \end{pmatrix} \mathbf{H}_p = \mathbf{I}_2.$$

Hence, $\mathbf{G}_p := \begin{pmatrix} 1 & 0 & x-1 \\ y & 0 & xy-y+1 \end{pmatrix}$ is a left inverse of $\mathbf{H}_p$. □

Example 6.2 Consider the 2D sample rate conversion scheme which consists of vertical upsampling by a factor 3, filtering with a filter $H(z) = H(z_1, z_2)$ and horizontal downsampling with a factor 2. We assume that H is FIR, and we would like to know if this scheme has an FIR inverse. To be more precise, we are looking for an FIR filter $G(z)$, such that horizontal upsampling by a factor 2, filtering with $G(z)$ and vertical downsampling with a factor 3, cancels the effect of the first sample rate conversion scheme.

Let the filter $H(z)$ be given by $H(z) = \sum h_{i,j} z_1^i z_2^j$. Following the method outlined in Section 5, but now for this 2D case, we construct the 3×2 polynomial matrix $H_{k,l}(z) = \sum h_{3i+k,2j+l} z_1^i z_2^j$, where $0 \le k \le 2$ and $0 \le l \le 1$.

Assume momentarily that $H(z)$ is a separable filter $H^h(z_1)H^v(z_2)$. It is easily seen that in this case the filters $H_{k,l}(z)$ are products of 1D polyphase components, i.e. $H_{k,l}(z) = H_k^h(z_1)H_l^v(z_2)$. Consequently, all the maximal minors of $H_{k,l}(z)$ have determinants equal to 0. Therefore the 2D analogue of Eq. 1 cannot be satisfied, and inversion is impossible.

Now we consider a non-separable case, where the filter $H(z)$ is given by the 4×6 (horizontal $\times$ vertical) impulse response

$$\begin{pmatrix} 2 & 3 & 2 & 1 & 3 & 2 \\ 3 & 5 & 3 & 1 & 3 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 1 & 1 & 1 \end{pmatrix}.$$

The polyphase component matrix $\mathbf{H}_p = (H_{kl})$ of $H(z)$ is defined by

$$H(z) = \sum_{k=0}^{2} \sum_{l=0}^{1} z_1^k z_2^l H_{kl}(z_1^2, z_2^3),$$

which is found as

$$\mathbf{H}_p = \begin{pmatrix} 2 + z_1 + z_2 + z_1 z_2 & 3 + 2z_1 + z_2 + z_1 z_2 \\ 3 + z_1 + 3z_2 + z_1 z_2 & 5 + 2z_1 + 3z_2 + z_1 z_2 \\ 2 + z_1 + 2z_2 + z_1 z_2 & 3 + 2z_1 + 2z_2 + z_1 z_2 \end{pmatrix}.$$

Computing the determinants of the maximal minors we find $D_0(z) = -1 - z_2$, $D_1(z) = -z_2 - z_1 z_2$ and $D_2(z) = 1 - z_2 - z_1 z_2$. These determinants are proper multivariable expressions and the Euclidean algorithm will therefore not work. In this case one easily verifies that $D_2 - D_1 = 1$, and therefore $\mathbf{H}_p$ is unimodular and there exist an inverse FIR filter $G(z)$. To find $G(z)$ we first need to find a left inverse $\mathbf{G}_p$ to $\mathbf{H}_p$.

The following is an example SINGULAR script used to compute a left inverse of $\mathbf{H}_p$. For notational convenience, we let $x := z_1$, $y := z_2, \mathbf{A} := \mathbf{H}_p$, and $\mathbf{B} := \mathbf{G}_p$.

```
ring r=0,(x,y),(c,dp); option(redSB);
vector v(1)=[2+x+y+xy,3+2*x+y+xy];
vector v(2)=[3+x+3*y+xy,5+2*x+3*y+xy];
vector v(3)=[2+x+2*y+xy,3+2*x+2*y+xy];
module M=v(1),v(2),v(3);
module G=std(M); matrix T=lift(M,G)
```

The output from SINGULAR is as follows:

```
> G;
G[1]=[0,1]
G[2]=[1]

> T;
T[1,1]=0
T[1,2]=1
T[2,1]=x+2
T[2,2]=-2x-3
T[3,1]=-1x-3
T[3,2]=2x+4
```

Since $\{(1,0),(0,1)\}$ is a Gröbner basis of the row vectors of $\mathbf{A}$, $\mathbf{A}$ is unimodular, and the relation $\mathtt{G} = \mathtt{MT}$ translates to

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \mathbf{A}^t\mathrm{T}.$$

By taking transpose of both sides, we get $\mathrm{T}^t\mathbf{A} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, i.e.

$$\begin{pmatrix} 1 & -2x-3 & 2x+4 \\ 0 & x+2 & -x-3 \end{pmatrix} \mathbf{A} = \mathbf{I}_2.$$

Hence, $\mathbf{B} := \begin{pmatrix} 1 & -2z_1 - 3 & 2z_1 + 4 \\ 0 & z_1 + 2 & -z_1 - 3 \end{pmatrix}$ is a left inverse of $\mathbf{A}$.

□

Appendix

A LaurentToPoly Algorithm

In this appendix, we present an algorithm that transforms a Laurent polynomial column vector to a polynomial column vector while preserving unimodularity. A schematic description of this **LaurentToPoly** Algorithm was given in Section 4, and we refer to the Figure 2 for the notations. This process is very powerful essentially because the unimodularity of the Laurent polynomial vector $\mathbf{v}(\boldsymbol{x}) \in (k[\boldsymbol{x}^{\pm 1}])^n$ is converted to the unimodularity of the polynomial vector $\hat{\mathbf{v}}(\boldsymbol{y}) \in (k[\boldsymbol{x}])^n$. For the results stated in this appendix without proof, see [19].

We start with a theorem (without proof) that can be seen as an analogue of the Noether Normalization Lemma. The Noether Normalization Lemma states that, for any given polynomial $f \in k[\boldsymbol{x}]$, by defining new variables $y_1, \ldots, y_m$ by $x_1 = y_1, x_2 = y_2 + y_1^l, \ldots, x_m = y_m + y_1^{l^{m-1}}$ for a sufficiently large $l \in \mathbb{N}$ and regarding f as a polynomial in the new variables $y_1, \ldots, y_m$, we can make f a monic polynomial in the first variable y_1. Now, we extend this to the Laurent polynomial ring $k[\boldsymbol{x}^{\pm 1}] = k[x_1^{\pm 1}, \ldots, x_m^{\pm 1}]$.

Figure 5: **LaurentNoether**

Input:	$f \in k[\boldsymbol{x}^{\pm 1}]$
Output:	$\boldsymbol{x} \to \boldsymbol{y}$, a change of variables
Specification:	the leading and the lowest coefficients of $f \in k[\boldsymbol{y}^{\pm 1}]$ with respect to the first variable y_1 are units in the ring $k[y_2^{\pm 1}, \ldots, y_m^{\pm 1}]$

Theorem A.1 (Laurent polynomial analogue of Noether Normalization) *Let $f \in k[\boldsymbol{x}^{\pm 1}]$ be a Laurent polynomial, and define new variables*

$y_1, \dots, y_m$ by $x_1 = y_1, x_2 = y_2 y_1^l, \dots, x_m = y_m y_1^{l^{m-1}}$. Then, for a sufficiently large $l \in \mathbb{N}$, the leading and the lowest coefficients of $f \in k[\mathbf{x}^{\pm 1}]$ with respect to the first variable y_1 are units in the ring $k[y_2^{\pm 1}, \dots, y_m^{\pm 1}]$.

With this theorem at hand, we can now describe the **LaurentToPoly** Algorithm.

Let $n \geq 2$, $S = k[x_2^{\pm 1}, \dots, x_m^{\pm 1}]$, and $\mathbf{v} = (v_1, \dots, v_n)^t \in (k[\mathbf{x}^{\pm 1})^n = (S[x_1])^n$. By using the algorithm **LaurentNoether**, we may assume that the leading and the lowest coefficients of v_1 w.r.t. x_1 are invertible elements of S. Write

$$v_1 \;=\; a_p x_1^p + a_{p+1} x_1^{p+1} + \cdots + a_q x_1^q$$

where a_p and a_q are units of S.

- **Step 1:** Using the invertibility of $a_p \in S$, define $\mathbf{D} \in \mathrm{M}_n(S[x_1^{\pm 1}])$ and $\mathbf{v}' \in (S[x_1^{\pm 1}])^n$ by

$$\mathbf{D} \;:=\; \begin{pmatrix} a_p^{-1} x_1^{-p} & 0 & \\ 0 & a_p x_1^p & \\ & & I_{n-2} \end{pmatrix}$$
$$\mathbf{v}' \;=\; (v_1', \dots, v_n')^t := \mathbf{D}\mathbf{v}.$$

  Note here that the matrix

$$\mathbf{D} \;=\; \mathbf{E}_{21}(a_p x_1^p)\mathbf{E}_{12}(1 - a_p^{-1} x_1^{-p})\mathbf{E}_{21}(1)\mathbf{E}_{12}(1 - a_p x_1^p)$$

  is realizable over $S[x_1^{\pm 1}]$, and

$$v_1' \;=\; a_p^{-1} x_1^{-p} v_1 = 1 + a_{p+1}/a_p x_1 + \cdots + a_q/a_p x_1^{q-p}$$

  is a **polynomial** in $S[x_1]$.

- **Step 2:** Since the constant term of $v_1' \in S[x_1]$ is 1, by adding suitable multiples of v_1' to v_i''s, $i = 2, \dots, n$, we can make $v_2', \dots, v_n'$ polynomials in $S[x_1]$ whose constant terms are zero, i.e. find $\mathbf{E} \in E_n(k[\mathbf{x}^{\pm 1}])$ such that

$$\mathbf{E}\mathbf{v}' = \hat{\mathbf{v}} = \begin{pmatrix} \hat{v}_1 \\ \vdots \\ \hat{v}_n \end{pmatrix} \in (S[x_1])^n,$$

  where $\hat{v}_1 \equiv 1 \bmod x_1$ and $\hat{v}_i \equiv 0 \bmod x_1$ for all $i = 2, \dots, n$.

- **Step 3:** Choose a sufficiently large number $l \in \mathbb{N}$ so that, with the following change of variables,

$$\begin{aligned} x_1 &= y_1 \cdot (y_2 \cdots y_m)^l \\ x_2 &= y_2 \\ &\vdots \\ x_m &= y_m, \end{aligned}$$

all the $\hat{v}_i$'s become polynomials in $k[y]$. Then $\hat{v}_1 \equiv 1 \bmod y_1 \cdots y_m$. Now give the transformation matrix $\mathbf{T} := \mathbf{ED}$ as the output.

□

It still remains to show that the outcome of this algorithm is what we want. This, however, follows from the following theorem (without proof).

Theorem A.2 *With the notations as in the above,* $\mathrm{v}(x)$ *is unimodular over* $k[x^{\pm 1}]$ *if and only if* $\hat{\mathrm{v}}(y)$ *is unimodular over* $k[y]$.

References

[1] T. Kalker, H. Park, and M. Vetterli, "Groebner Bases Techniques in Multidimensional Multirate Systems," *Proceedings of IGASSP 95*, 1995.

[2] M. Vetterli and J. Kovačević, *Wavelets and Subband Coding.* Prentice Hall Signal Processing Series, Prentice Hall, 1995.

[3] M. Vetterli, "Filter banks allowing perfect reconstruction," *IEEE Transactions on Signal Processing*, vol. 10, pp. 219–244, April 1986.

[4] A. J. E. M. Janssen, "Note on a linear system occurring in perfect reconstruction," *IEEE Transactions on Signal Processing*, vol. 18, no. 1, pp. 109–114, 1989.

[5] M. Vetterli and C. Herley, "Wavelets and filter banks: Theory and design," *IEEE Transactions on ASSP*, vol. 40, pp. 2207–2232, Sept. 1992.

[6] P. P. Vaidyanathan, *Multirate Systems and Filter Banks.* Prentice Hall Signal Processing Series, Prentice Hall, 1993.

[7] B. Buchberger, "Gröbner bases – an algorithmic method in polynomial ideal theory," in *Multidimensional systems theory* (N. K. Bose, ed.), ch. 6, pp. 184–232, Dordrecht: D. Reidel, 1985.

[8] D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms.* Undergraduate Texts in Mathematics, Springer-Verlag, 1992.

[9] B. Mishra, *Algorithmic Algebra.* Texts and Monographs in Computer Science, Springer-Verlag, 1993.

[10] W. Adams and P. Loustaunau, *An Introduction to Gröbner Bases*, vol. 3 of *Graduate Studies in Mathematics.* American Mathematical Society, 1994.

[11] T. Becker and V. Weispfenning, *Gröbner Bases*, vol. 141 of *Graduate Texts in Mathematics.* Springer-Verlag, 1993.

[12] D. Eisenbud, *Introduction to Commutative Algebra with a View Towards Algebraic Geometry.* Graduate Texts in Mathematics, Springer-Verlag, 1995.

[13] X. Chen, I. Reed, T. Helleseth, and T. K. Truong, "Use of Gröbner bases to Decode Binary Cyclic Codes up to the True Minimum Distance," *IEEE Transactions on Information Theory*, vol. 40, pp. 1654–1660, Sept. 1994.

[14] D. Bayer and M. Stillman, "A theorem on redefining division orders by the reverse lexicographic orders," *Duke J. of Math.*, vol. 55, pp. 321–328, 1987.

[15] D. Bayer and M. Stillman, "On the complexity of computing syzygies," *J. Symb. Comp.*, vol. 6, pp. 135–147, 1988.

[16] Gruson, Lazarsfeld, and Peskine, "On a theorem of Castelnuovo, and the equations defining space curves," *Invent. Math.*, vol. 72, pp. 491–506, 1983.

[17] Winkler, "On the complexity of the Gröbner basis algorithm over $k[x, y, z]$," in *Springer Lect. Notes in Computer Sci. 174* (J. Fitch, ed.), Springer-Verlag, 1984.

[18] A. Logar and B. Sturmfels, "Algorithms for the Quillen-Suslin theorem," *Journal of Algebra*, vol. 145, pp. 231–239, 1992.

[19] H. Park, *A Computational Theory of Laurent Polynomial Rings and Multidimensional FIR Systems.* PhD thesis, University of California at Berkeley, 1995.

[20] P. P. Vaidyanathan and T. Chen, "Role of anticausal inverses in multirate filter banks – part i: System-theoretic fundamentals," *IEEE Transactions on Signal processing*, vol. 43, pp. 1090–1102, May 1995.

[21] G. Greuel, G. Pfister, and H. Schoenemann, *Singular User Manual.* University of Kaiserslautern, version 0.90 ed., 1995. available by anonymous ftp from helios.mathematik.uni-kl.de.

[22] H. Park and C. Woodburn, "An Algorithmic Proof of Suslin's Stability Theorem for Polynomial Rings," *to appear in Journal of Algebra*, 1995.

[23] S. Basu and H. M. Choi, "Multidimensional causal, stable, perfect reconstruction filter banks," *Proceedings of the ICIP-94*, vol. 1, pp. 805–809, Nov. 1994.

[24] Z. Cvetković and M. Vetterli, "Oversampled Filtered Banks," *submitted to IEEE Transactions on Signal Processing*, 1994.