

# Exact and Approximate Polynomial Decomposition Methods for Signal Processing Applications

Sefa Demirtas, Guolong Su, and Alan V. Oppenheim

Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge MA, 02139

Email: sefa@mit.edu

**Abstract**—Signal processing is a discipline in which functional composition and decomposition can potentially be utilized in a variety of creative ways. From an analysis point of view, further insight can be gained into existing signal processing systems and techniques by reinterpreting them in terms of functional composition. From a synthesis point of view, functional composition offers new algorithms and techniques with modular structure. Moreover, computations can be performed more efficiently and data can be represented more compactly in information systems represented in the context of a compositional structure. Polynomials are ubiquitous in signal processing in the form of  $z$ -transforms. In this paper, we summarize the fundamentals of functional composition and decomposition for polynomials from the perspective of exploiting them in signal processing. We compare exact polynomial decomposition algorithms for sequences that are exactly decomposable when expressed as a polynomial, and approximate decomposition algorithms for those that are not exactly decomposable. Furthermore, we identify efficiencies in using exact decomposition techniques in the context of signal processing and introduce a new approximate polynomial decomposition technique based on the use of Structured Total Least Norm (STLN) formulation.

—**Keywords:** polynomials, composition, decomposition, STLN

## I. INTRODUCTION

Functional composition can be defined as the application of one function  $F(x)$  to the results of another function  $G(x)$  to obtain the composition  $H(z) = F(G(x))$ . Conversely, functional decomposition is the process of obtaining two or more functions which, when composed, yield the original function. The functions  $F$ ,  $G$  and  $H$  involved in these operations may belong to general classes such as continuous or discrete functions, polynomials, boolean functions or simply tabular functions given in the form of a truth table, decision table or more generally as a list of possible multi-valued inputs and corresponding multi-valued outputs.

Factorization, namely the representation of computationally complex functionals or tasks as a cascade of sub-functionals or sub-tasks, has been used widely in the fields of mathematics, computation and signal processing, including singular value decomposition of matrices and implementation of LTI systems as a cascade of at most second order systems among many others. Composition and decomposition, which are two operations different from factorization in their nature, also have naturally arisen or been used in signal processing such as frequency or phase modulation. Filter sharpening is another example where a given filter is interconnected repeatedly with adders and multipliers to obtain sharper filter characteristics than the original filter, the structure of which can conveniently be represented using functional composition [1], [2], [3]. Aside from naturally arising in such applications, functional composition has been exploited in various signal processing and filter design techniques by a number of authors [4], [5], [6], [7]. For instance, as an extension of filter sharpening method proposed in [1], Saramaki [3] proposed designing filters in the form of composed polynomials, which breaks the problem of

designing the overall filter into two simpler filter design problems and makes optimization easier and more efficient. Although incorporating functional composition and decomposition in these applications has proven to be useful, attempts to more broadly exploit composition and decomposition in signal processing have been limited.

Polynomial composition and decomposition deserve particular attention since polynomials are ubiquitous in the form of the  $z$ -transform representation of discrete-time filters and signals. As a potential application of polynomial decomposition, a discrete time signal  $h[n]$  can be represented by fewer parameters than the number of its nonzero coefficients if its  $z$ -transform  $H(z)$  is decomposable as  $F \circ G(z)$  since in general the order of  $H(z)$  is larger than the sum of the orders of  $F(z)$  and  $G(z)$ . This offers the potential, for example, for more efficient signal representation and for highly modular system implementation. In this paper, we briefly summarize the existing mathematical literature on the fundamentals of polynomial decomposition as well as exact and approximate polynomial decomposition techniques [8], [9], [10], [11], [12], [13], [14]. Furthermore, we report improved performance for a certain exact decomposition technique when it is specifically used in the context of signal processing; and also introduce a new approximate polynomial decomposition technique based on the Structured Total Least Norm (STLN) formulation.

## II. POLYNOMIAL COMPOSITION AND DECOMPOSITION

Consider  $F(x)$ , the polynomial that represents a length- $(M + 1)$  sequence  $f_n$ ,

$$F(x) = \sum_{n=0}^M f_n x^n, \quad (1)$$

which corresponds to the  $z$ -transform of  $f_n$  for  $x = z^{-1}$ . Composing  $F(x)$  with another polynomial that represents a length- $(N + 1)$  sequence  $g_n$ , we obtain

$$H(x) = F(G(x)) = \sum_{n=0}^M f_n G^n(x). \quad (2)$$

Hence  $h_n$ , the sequence represented by  $H(x)$  becomes

$$h_n = f_0(g^{(0)}) + f_1(g^{(1)}) + f_2(g^{(2)}) + f_3(g^{(3)}) + \dots \quad (3)$$

where  $g^{(i)}$  corresponds to  $i$  self-convolutions of the sequence  $g_n$ . Equivalently

$$\mathbf{h} = \mathbf{C}\mathbf{f} \quad (4)$$

where the  $k^{\text{th}}$  column of the matrix  $\mathbf{C}$  consists of  $g^{(k-1)}$ ; and  $\mathbf{f}$  and  $\mathbf{h}$  are the coefficient vectors of  $F(x)$  and  $H(x)$  in the ascending order, respectively. Therefore, it is relatively straightforward to obtain the coefficients of the composition polynomial  $H(x)$  given the coefficients of its components  $F(x)$  and  $G(x)$ . The inverse problem is, however, more difficult.

This work was supported by Texas Instruments Leadership University Program, and Bose Corporation.

While decomposable polynomials have the form of equation (2) or (4), decomposability cannot be easily inferred directly from the polynomial coefficients. A key observation first made by Fried and MacRae [9] is that a polynomial  $H(x)$  has another polynomial  $G(x)$  as a decomposition factor if and only if the bivariate polynomial

$$\phi_G(y, z) \triangleq \frac{G(y) - G(z)}{y - z} \quad (5)$$

divides  $\phi_H(y, z)$  resulting in a bivariate polynomial in  $y$  and  $z$ . This follows easily by expressing  $\phi_H(y, z)$  using equations (1) and (2)

$$\phi_H(y, z) = \frac{H(y) - H(z)}{y - z} = \sum_{n=1}^M f_n \frac{G^n(y) - G^n(z)}{y - z}, \quad (6)$$

and factoring  $\phi_G(y, z)$  out from the summation. The bivariate polynomials  $\phi_{(\cdot)}$  defined as in equation (5) have a specific symmetry in their coefficients, namely the terms that have the same total order of  $y$  and  $z$  have the same coefficients.

A stronger statement regarding decomposability of a given polynomial  $H(x)$  without the need for testing against a potential decomposition factor  $G(x)$  was also provided by Fried ([15], Theorem 1), and refined by Turnwald more recently ([16], Theorem 1). Specifically, a given polynomial  $H(x)$  with a non-prime order is decomposable if and only if  $\phi_H(y, z)$  is factorable. Moreover, if  $\phi_H(y, z)$  is factorable, then at least one of its factors has to be of the form  $\phi_G(y, z)$  since  $H(x)$  is decomposable. We note that in the context of signal processing, a sequence that corresponds to a nondecomposable polynomial with a prime order may easily be modified to obtain a non-prime order polynomial leading to a potentially decomposable polynomial through delaying the sequence in time.

Factorability of  $\phi_H(y, z)$  can be determined using a particular test for bivariate polynomial factorization that was introduced by Ruppert [17], [18], [19], [20]. Specifically,  $\phi_H(y, z)$  is factorable if and only if its associated Ruppert matrix  $\mathbf{R}$  is rank deficient, where  $\mathbf{R}$  is a  $(4P^2 - 10P + 6) \times (2P^2 - 3P)$  matrix the entries of which are linear functions of the coefficients of  $H(x)$ . Therefore rank deficiency of  $\mathbf{R}$  directly implies decomposability of  $H(x)$ .

The linearity of the Ruppert matrix in the coefficients of  $\phi_H(y, z)$  allows rewriting  $\mathbf{R}$  as the linear combination of a basis for Ruppert matrices. More specifically

$$\mathbf{R} = \mathbf{R}\{H(x)\} = \mathbf{R}\left\{\sum_{i=1}^P h_i x^i\right\} = \sum_{i=1}^P h_i \mathbf{R}\{x^i\} \triangleq \sum_{i=1}^P h_i \mathbf{R}_i. \quad (7)$$

In equation (7), the summation index starts from 1 since the Ruppert matrix of a constant polynomial is the zero matrix. Although each monomial term  $x^i$  has order less than or equal to  $P$ , the matrices  $\mathbf{R}_i$ ,  $i = 1 \dots P$  have the same dimensions as  $\mathbf{R}$  since they correspond to the Ruppert matrix of the degenerate  $P^{\text{th}}$  order polynomial  $0x^P + x^i$  and can be considered to be a basis for Ruppert matrices of all polynomials of order  $P$ . The formulation of the Ruppert matrix as in equation (7) will provide a basis for certain approximate polynomial decomposition techniques discussed in Section IV.

### III. EXACT DECOMPOSITION OF POLYNOMIALS

In this section, we introduce four polynomial decomposition algorithms which obtain the components  $F(x)$  and  $G(x)$  if the polynomial  $H(x)$  is a composition, i.e.,  $H(x) = F(G(x))$ . These algorithms focus on obtaining the right decomposition factor  $G(x)$  first since  $F(x)$  can be obtained relatively easily from the linear relationship given in equation (4) once  $G(x)$  is known.

Based on the observations of Fried and MacRae [9] as summarized in Section II, Barton and Zippel [10] proposed a decomposition algorithm which, given  $H(x)$ , obtains the bivariate polynomial  $\phi_H(y, z)$  and examines all of the factors of  $\phi_H(y, z)$  to find a factor that has the form of  $\phi_G(y, z)$  described in Section II. The requirement to examine all combinations of the factors to obtain a factor of the form  $\phi_G(y, z)$  makes this algorithm computationally inefficient since the number of combinations is exponential in the number of factors. A slightly more efficient algorithm, proposed by Alagar and Thanh [11], uses the fact that the derivative of a decomposable polynomial  $H(x)$  as in equation (2) has  $G'(x)$  as one of its factors since

$$H'(x) = F'(G(x))G'(x).$$

However, it still requires examining each factor of  $H'(x)$  with appropriate order as a candidate for  $G'(x)$ .

A more systematic polynomial decomposition algorithm is given by Kozen and Landau [12].  $M$  and  $N$ , the orders of  $F(x)$  and  $G(x)$  respectively, are required as part of the input. The algorithm uses the fact that the coefficients of the terms with the highest  $N$  orders in  $H(x)$  are determined only by  $f_M$ , namely the coefficient of the highest order term in  $F(x)$ , and all the coefficients of  $G(x)$  as seen from equation (3).

As the first step,  $H(x)$  is scaled to be monic, i.e. to have unity as the coefficient of the highest order term, which does not affect decomposability. Restricting  $G(x)$  and  $F(x)$  to be also monic, the coefficients of  $G(x)$  are obtained in the order of decreasing powers through solving  $N$  equations systematically involving the coefficients of  $G(x)$ . After the decomposition is obtained for the monic polynomial, the scaling is undone. This algorithm is computationally much more efficient than the previous algorithms, but requires knowledge of the degrees of the decomposition components. If this information is not available, the algorithm is then implemented repetitively for candidate orders  $M$  and  $N$ , which are factors of the order of  $H(x)$ .

A different class of decomposition algorithm proposed by Aubry and Valibouze [21] utilizes the relationship between the coefficients of a polynomial and the power sums of its roots, known as the Newton identities. More specifically, the coefficients of an  $N^{\text{th}}$  order monic polynomial  $G(x)$  can be uniquely determined from the  $n^{\text{th}}$  power sums of its roots for  $n = 1, \dots, N$  defined as

$$\sum_{i=1}^N r_{g,i}^n, \quad n = 1 \dots N, \quad (8)$$

where  $r_{g,i}$   $i = 1, \dots, N$  are the roots of  $G(x)$ . A decomposable monic polynomial  $H(x) = F(G(x))$  can be written as

$$H(x) = \prod_{j=1}^M (G(x) - r_{f,j}) \triangleq \prod_{j=1}^M \tilde{G}_j(x), \quad (9)$$

where  $r_{f,j}$   $j = 1 \dots M$  are the roots of  $F(x)$ . Newton's identity suggests a method to obtain the coefficients of  $\tilde{G}_j(x)$  except its constant term from the coefficients of  $N$  highest order terms in  $H(x)$  since the roots of  $\tilde{G}_j(x)$  are also the roots of  $H(x)$ . This establishes the basis of a decomposition algorithm since each polynomial  $\tilde{G}_j(x)$ ,  $j = 1 \dots M$  has the same coefficients as  $G(x)$  except the constant term, which can be chosen freely for a valid decomposition.

Both Kozen and Landau's algorithm [12] and Aubry and Valibouze's algorithm [21] are based on using the coefficients of  $N$  highest order terms in  $H(x)$ . Due to representation of these coefficients and their manipulations with finite precision, the performance of both algorithms deteriorates with increasing polynomial orders. This problem can partially be avoided in the context of signal

processing applications where the roots of polynomials might be more relevant than its coefficients to a specific application since signals and systems often are represented through their poles and zeros. Computation of  $n^{\text{th}}$  power sums in equation (8) can be performed directly from the poles and zeros in the implementation of Aubry and Valibouze's algorithm leading to significantly enhanced precision for the decomposition factors  $G(x)$  and  $F(x)$ . Figure 1 shows a comparison of the performance of three algorithms, namely Kozen and Landau's algorithm and Aubry and Valibouze's algorithm implemented using firstly the coefficients and secondly the roots of  $H(x)$ . The polynomials  $H(x)$  were obtained by composing random polynomials  $F(x)$  and  $G(x)$  with the coefficient of the highest orders fixed to be unity and where the respective orders  $M$  and  $N$  were chosen equal and varied from 5 to 75 with increments of five. For easy comparison of algorithms, constant term of  $G(x)$  was set to zero. The decomposition is considered successful if the SNR is more than 80dB, where the error is defined as the difference between the true and the obtained decomposition factors. The two algorithms using coefficients show an almost identical success rate whereas the implementation of the A-V algorithm using the roots of  $H(x)$  directly outperforms the others significantly. All of the polynomials of order 625 were successfully decomposed by this algorithm while 76 polynomials of order 4900 were decomposed successfully.

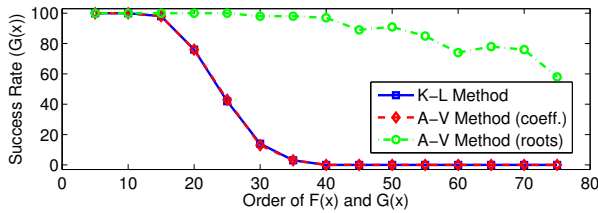


Fig. 1. The comparison of number of successful decompositions of  $H(x) = F \circ G(x)$  by Kozen and Landau's method and Aubry and Valibouze's method implemented using coefficients and the roots of polynomials.

#### IV. APPROXIMATE DECOMPOSITION OF POLYNOMIALS

Section III focused on algorithms for obtaining the decomposition factors when it is known that a given polynomial is decomposable. Modeling nondecomposable polynomials by decomposable ones is also of significant interest, particularly in such applications as signal representation and compression because of the inherent reduction in the number of free parameters. In this section, we introduce two approximate polynomial decomposition algorithms, which can be viewed as an extension of the exact decomposition algorithms.

Corless et al [13] proposed two approximate decomposition methods that start from an initial guess for the decomposition factors  $F(x)$  and  $G(x)$ , which are obtained using Kozen-Landau algorithm in Section III, and iteratively obtain a nearby decomposable polynomial. The first algorithm determines  $\Delta G(x)$  at each iteration to minimize

$$\begin{aligned} & \|H(x) - F_k(G_k(x) + \Delta G(x))\| \\ & \approx \|H(x) - F_k(G_k(x)) - F'_k(G_k(x))\Delta G(x)\|, \end{aligned} \quad (10)$$

where the subscript  $k$  represents the current iteration step,  $\|\cdot\|$  denotes the  $l_2$  norm and only the first term in the Taylor series is taken into account since  $\Delta G$  is assumed to be small at each iteration. This algorithm approximates a nonlinear optimization problem with a simpler one and if it converges, the convergence rate is linear. The second algorithm attempts to solve the nonlinear problem of minimizing  $\|H(x) - F(G(x))\|$  directly using Newton iteration where convergence is quadratic at the expense of increased computational

complexity. The quality of the decomposition obtained by these algorithms is highly dependent on the validity of the assumption that there is a decomposable polynomial close to  $H(x)$  since the initial guess is obtained through Kozen-Landau decomposition algorithm.

Given an overdetermined and nonconsistent set of linear equations  $\mathbf{A}\mathbf{x} \approx \mathbf{b}$ , the solution that minimizes  $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|$  is given by the well known least squares solution. This solution leads to  $\mathbf{A}\hat{\mathbf{x}} = \mathbf{b} + \Delta\mathbf{b}$ , i.e. only the entries of  $\mathbf{b}$  are altered to satisfy the equation and  $\mathbf{A}$  remains intact. A generalization of this problem is called total least squares (TLS) where the entries of  $\mathbf{A}$  are also subject to possible change. More specifically the Euclidean norm of the matrix  $[\Delta\mathbf{A}|\Delta\mathbf{b}]$  is minimized such that

$$(\mathbf{A} + \Delta\mathbf{A})\mathbf{x} = \mathbf{b} + \Delta\mathbf{b}. \quad (11)$$

This is equivalent to finding the closest rank deficient matrix  $[\mathbf{A} + \Delta\mathbf{A}|\mathbf{b} + \Delta\mathbf{b}]$  since equation (11) can be expressed as

$$[\mathbf{A} + \Delta\mathbf{A}|\mathbf{b} + \Delta\mathbf{b}]\mathbf{y} = \mathbf{0}, \quad (12)$$

where  $\mathbf{y} = [\mathbf{x}^T, -1]^T$ . The solution is obtained by suppressing the smallest singular value of the matrix  $[\mathbf{A}|\mathbf{b}]$ , however in general  $\mathbf{A}$  and  $[\mathbf{A}|\mathbf{b}]$  do not retain their previous structure such as sparsity or the structure of a Ruppert matrix. The collective algorithms that assert retaining any matrix structure are referred to as Structured Total Least Square Norm (STLN) [22] for a general norm and reduce to Structured Total Least Square (STLS) for the choice of  $l_2$  norm [23].

As an alternative method for approximate polynomial decomposition, Botting [24] proposed a solution in the STLS framework to the problem of finding a rank deficient Ruppert matrix for which the corresponding polynomial is close in  $l_2$  to  $H(x)$  since the equivalence of polynomial decomposability to having a rank deficient Ruppert matrix is established. Specifically, the approximate decomposition problem reduced to the optimization problem specified as ([23])

$$\min_{\tilde{h}_i, \mathbf{w}} \sum_{i=1}^P (\tilde{h}_i - h_i)^2 \quad \text{such that} \quad \tilde{\mathbf{R}}\mathbf{w} = \mathbf{0} \quad \text{and} \quad \mathbf{w}^T\mathbf{w} = 1, \quad (13)$$

where  $\tilde{h}_i$ ,  $i = 1, \dots, P$  are the coefficients of a decomposable polynomial  $\tilde{H}(x)$  and  $\tilde{\mathbf{R}}$  is its Ruppert matrix. The second constraint ensures that  $\mathbf{w}$  is not identically zero so that  $\tilde{\mathbf{R}}$  has a nontrivial null space. The optimization problem (13) is shown to be equivalent to a nonlinear generalized singular value decomposition referred to as Riemannian SVD problem in [23], namely finding the triplet  $(\mathbf{u}, \tau, \mathbf{v})$  corresponding to the smallest scalar  $\tau$  that satisfies

$$\mathbf{R}\mathbf{v} = \mathbf{D}_v\mathbf{u}\tau, \quad \mathbf{u}^T\mathbf{D}_v\mathbf{u} = 1, \quad \mathbf{R}^T\mathbf{u} = \mathbf{D}_u\mathbf{v}\tau, \quad \mathbf{v}^T\mathbf{D}_u\mathbf{v} = 1$$

where  $\mathbf{D}_u$  and  $\mathbf{D}_v$  are matrices with entries quadratic in the vectors  $\mathbf{u}$  and  $\mathbf{v}$  and also a heuristic iterative solution is provided leading to a decomposable polynomial with coefficients  $\tilde{h}_i = h_i - \mathbf{u}^T\mathbf{R}_i\mathbf{v}\tau$ . The iteration ends when the smallest singular value of  $\tilde{\mathbf{R}}$  gets less than a given threshold, however no guarantee for convergence exists.

#### V. STLN FORMULATION FOR APPROXIMATE DECOMPOSITION

The exploitation of structure preserving low rank approximation (SPLRA) formulations such as STLN as described in [22] for finding a rank deficient Ruppert matrix has been suggested as a potentially useful method ([20], Remark 6), however no implementations or results were reported. In this section, we propose a new STLN-based algorithm for approximate polynomial decomposition in order to solve equation (12) using the iterative methods proposed in [22] and [25] to find a rank deficient Ruppert matrix the associated polynomial of which is close to a given nondecomposable polynomial. This

method requires solving a simple quadratic optimization problem at each iteration step.

The rank deficiency of the Ruppert matrix can be ensured by writing one of its column in terms of other columns. In order to formulate this problem as in equation (12), we set the column vector  $\mathbf{b}$  as one of the columns of the Ruppert matrix  $\mathbf{R}$  and  $\mathbf{A}$  equal to  $\mathbf{R}$  excluding the column vector  $\mathbf{b}$ . This column can be specified as the one that minimizes the residual when expressed in terms of other columns, i.e.

$$\arg \min_{\mathbf{b}} \left( \min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 \right). \quad (14)$$

If  $[\Delta\mathbf{A}|\Delta\mathbf{b}]$  is constrained to be of the form  $\sum_{i=1}^P \alpha_i [\mathbf{A}_i|\mathbf{b}_i]$  for real scalars  $\alpha_i$ ,  $i = 1 \dots P$ , where  $\mathbf{A}_i$  and  $\mathbf{b}_i$  are obtained from  $\mathbf{R}_i$  consistent with the column index of  $\mathbf{b}$  in  $\mathbf{R}$ , the resulting matrix  $[\mathbf{A} + \Delta\mathbf{A}|\mathbf{b} + \Delta\mathbf{b}]$  in equation (12) will retain the same structure as  $[\mathbf{A}|\mathbf{b}]$  due to the linear relationship given in equation (7). We are interested in minimizing the change in the coefficients of the polynomial to be decomposed subject to the rank deficiency constraint in equation (12), i.e.

$$\min_{\alpha_i, \mathbf{y}} \sum_{i=1}^P \alpha_i^2 \quad \text{such that } \hat{\mathbf{r}} = [\mathbf{A} + \Delta\mathbf{A}|\mathbf{b} + \Delta\mathbf{b}]\mathbf{y} = 0, \quad (15)$$

where  $\hat{\mathbf{r}}$  is the residual that is ideally zero. The similarity of the optimization problems given in equations (13) and (15) is obvious since we chose to minimize the  $l_2$  norm of the coefficient perturbation vector  $\boldsymbol{\alpha} = [\alpha_1 \ \alpha_2 \ \dots \ \alpha_P]^T$ . An explicit constraint for  $\mathbf{y}$  to be nonzero is not required in equation (15) since it is restricted to be of the form  $\mathbf{y} = [\mathbf{x}^T, -1]^T$  with this formulation.

Consider a relaxation of the nonlinear optimization problem in (15)

$$\min_{\alpha_i, \mathbf{y}} \boldsymbol{\alpha}^T \boldsymbol{\alpha} + \lambda^2 \hat{\mathbf{r}}^T \hat{\mathbf{r}} \quad (16)$$

where  $\lambda$  is the penalty parameter for any nonzero residual  $\hat{\mathbf{r}}$  and is required to be chosen appropriately large for a good approximation to the original problem. An iterative algorithm for the solution of the nonlinear optimization problem in (16) is given in [22]. In the context of this algorithm, the residual  $\hat{\mathbf{r}}$  defined in equation (15) is

$$\hat{\mathbf{r}} = \mathbf{A}\mathbf{x} + \Delta\mathbf{A}\mathbf{x} - \mathbf{b} - \Delta\mathbf{b} \triangleq \mathbf{A}\mathbf{x} + \mathbf{X}\boldsymbol{\alpha} - \mathbf{b} - \mathbf{Q}\boldsymbol{\alpha} \quad (17)$$

where the matrix  $\mathbf{X}$  is obtained by

$$\Delta\mathbf{A}\mathbf{x} = \sum_{i=1}^P \alpha_i \mathbf{A}_i \mathbf{x} = \sum_{i=1}^P (\mathbf{A}_i \mathbf{x}) \alpha_i \triangleq \mathbf{X}\boldsymbol{\alpha}. \quad (18)$$

More specifically the  $i^{\text{th}}$  column of  $\mathbf{X}$  consists of  $\mathbf{A}_i \mathbf{x}$ . Similarly,  $i^{\text{th}}$  column of  $\mathbf{Q}$  consists of  $\mathbf{b}_i$ . The steps of the iteration are summarized in Algorithm 1.

#### ALGORITHM 1

**Input:**  $H(x)$  with coefficients  $h_i$ ,  $i = 1, \dots, P$ .

**Output:** Decomposable  $\tilde{H}(x)$  with coefficients  $\tilde{h}_i$ .

Specify  $\mathbf{A}$ ,  $\mathbf{b}$  from  $\mathbf{R}$  as in (14). Set  $\mathbf{x}^{(k)} = \arg \min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2$  and  $\boldsymbol{\alpha}^{(k)} = \mathbf{0}$ . Obtain  $\mathbf{A}_i$ ,  $\mathbf{b}_i$  from  $\mathbf{R}_i$ ,  $i = 1, \dots, P$ . Set  $\mathbf{Q} = [\mathbf{b}_1 | \dots | \mathbf{b}_P]$ .

1. Set  $\mathbf{X}^{(k)} = [\mathbf{A}_1 \mathbf{x}^{(k)} | \dots | \mathbf{A}_P \mathbf{x}^{(k)}]$ ,  $\mathbf{K}^{(k)} = \mathbf{X}^{(k)} - \mathbf{Q}$ . In first step only, set  $\lambda = (\sigma_{\mathbf{K}^{(k)}})^{-1}$ , the inverse of minimum singular value of  $\mathbf{K}^{(k)}$ .
2. Solve the following quadratic program:
$$\underset{\Delta\boldsymbol{\alpha}, \Delta\mathbf{x}}{\text{minimize}} \left\| \begin{bmatrix} \lambda \mathbf{K}^{(k)} & \lambda \mathbf{A}^{(k)} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{pmatrix} \Delta\boldsymbol{\alpha} \\ \Delta\mathbf{x} \end{pmatrix} + \begin{pmatrix} \lambda \hat{\mathbf{r}}^{(k)} \\ \boldsymbol{\alpha}^{(k)} \end{pmatrix} \right\|_2.$$
3. Set  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \Delta\mathbf{x}$ ,  $\boldsymbol{\alpha}^{(k+1)} = \boldsymbol{\alpha}^{(k)} + \Delta\boldsymbol{\alpha}$ ,  $\tilde{h}_i = h_i + \alpha_i^{(k+1)}$ .
4. Exit if  $\tilde{\mathbf{R}}^{(k+1)} = \mathbf{R} + \sum_{i=1}^P \alpha_i^{(k+1)} \mathbf{R}_i$  rank deficient; else, go to 1.

## VI. SIMULATIONS

In this section, the performance of the Riemann SVD (RiSVD) formulation summarized in Section IV and the Structured Total Least Norm (STLN) formulation developed in Section V are compared in the context of approximating Ruppert matrices with those that are rank deficient. One hundred decomposable polynomials  $\hat{H}(x)$  were obtained by composing a random  $M^{\text{th}}$  order polynomial  $F(x)$  with an  $N^{\text{th}}$  order random polynomial  $G(x)$ , the coefficients of both of which are selected from a standard normal distribution except the highest order terms that are fixed to be unity to avoid degenerate compositions. The coefficient vector  $\hat{\mathbf{h}}$  of each polynomial  $\hat{H}(x)$  is then perturbed by an error vector  $\mathbf{e}$  to obtain a nondecomposable polynomial  $H(x)$ , where the coefficients of  $\mathbf{e}$  are obtained from a standard normal distribution and scaled so that  $\|\mathbf{e}\|_2 = 10^{-2} \|\hat{\mathbf{h}}\|_2$ , i.e. the SNR is 40dB.

The iterations in both RiSVD and STLN methods were ended when the Ruppert matrix is considered numerically rank deficient where this is defined as the existence of a significantly large ratio between any two consecutive singular values among the smallest twenty singular values of the Ruppert matrix. More specifically, the Ruppert matrix is considered to be rank deficient when the maximum ratio between consecutive singular values are greater than one hundred times that of the original Ruppert matrix or  $10^4$ , whichever is smaller.

Table I summarizes the results of the iterations for the STLN and RiSVD methods tested against nondecomposable polynomials of different orders. The success rates are calculated as the ratio of the number of cases where the ending criterion was met before one hundred iterations to the total number of polynomials that did not have numerically rank deficient Ruppert matrices at the initial stage. The STLN method proves to be more successful than the RiSVD method for all orders.

The approximate decomposition method introduced by Corless et al [13] and summarized in Section IV obtains a decomposable approximation to  $H(x)$  in all the cases shown in Table I, however a comparison cannot be made between this method and the STLN or RiSVD methods since Corless' method is not based on approximating Ruppert matrices. However, when Kozen and Landau's exact decomposition algorithm [12] is applied to the polynomials obtained by STLN method, a faithful decomposition cannot be obtained. In fact, the polynomials obtained by RiSVD yield more consistent decomposition factors when decomposed by this method. This suggests that our rank deficiency definition does not necessarily represent decomposability by Kozen and Landau's method. We motivate exploring a more consistent rank deficiency metric for Ruppert matrices consistent with exact decomposability of the corresponding polynomials and other exact decomposition algorithms that faithfully decompose polynomials obtained through STLN method.

TABLE I  
SUCCESS RATE OF APPROXIMATE DECOMPOSITION METHODS (%)

$\text{deg}(F)$	$\text{deg}(G)$	$\text{deg}(F \circ G)$	STLN	RiSVD
2	2	4	100.0	73.0
2	3	6	97.0	2.0
3	2	6	96.0	9.0
2	4	8	92.0	5.0
4	2	8	94.8	7.3
3	3	9	86.0	5.0
2	5	10	81.0	1.0
5	2	10	90.0	10.0
2	6	12	79.0	2.0
3	4	12	83.7	12.2
4	3	12	82.2	10.0
6	2	12	95.0	11.3

## REFERENCES

- [1] J. Kaiser and R. Hamming, "Sharpening the response of a symmetric nonrecursive filter by multiple use of the same filter," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 25, no. 5, pp. 415–422, Oct 1977.
- [2] S. Nakamura and S. K. Mitra, "Design of FIR digital filters using tapped cascaded FIR subfilters," *Circuits, Systems, and Signal Processing*, vol. 1, pp. 43–56, 1982.
- [3] T. Saramaki, "Design of FIR filters as a tapped cascaded interconnection of identical subfilters," *Circuits and Systems, IEEE Transactions on*, vol. 34, no. 9, pp. 1011–1029, Sep 1987.
- [4] A. Oppenheim, D. Johnson, and K. Steiglitz, "Computation of spectra with unequal resolution using the Fast Fourier Transform," *Proceedings of the IEEE*, vol. 59, no. 2, pp. 299–301, Feb. 1971.
- [5] C. Asavathiratham, P. Beckmann, and A. Oppenheim, "Frequency warping in the design and implementation of fixed-point audio equalizers," in *Applications of Signal Processing to Audio and Acoustics, 1999 IEEE Workshop on*, 1999, pp. 55–58.
- [6] R. Mersereau, W. Mecklenbrauker, and J. Quatieri, T., "McClellan transformations for two-dimensional digital filtering-part i: Design," *Circuits and Systems, IEEE Transactions on*, vol. 23, no. 7, pp. 405–414, Jul 1976.
- [7] M. R. Schroeder, "Natural Sounding Artificial Reverberation," *J. Aud. Eng. Soc.*, vol. 10, no. 3, pp. 219–223, 1962.
- [8] J. F. Ritt, "Prime and composite polynomials," *Transactions of the American Mathematical Society*, vol. 23, no. 1, pp. 51–66, 1922.
- [9] M. D. Fried and R. E. MacRae, "On curves with separated variables," *Mathematische Annalen*, vol. 180, pp. 220–226, 1969.
- [10] D. R. Barton and R. E. Zippel, "A polynomial decomposition algorithm," in *Proceedings of the third ACM Symposium on Symbolic and Algebraic Computation*, ser. SYMSAC '76. New York, NY, USA: ACM, 1976, pp. 356–358.
- [11] V. Alagar and M. Thanh, "Fast polynomial decomposition algorithms," in *EUROCAL '85*, ser. Lecture Notes in Computer Science, B. Caviness, Ed. Springer Berlin / Heidelberg, 1985, vol. 204, pp. 150–153.
- [12] D. Kozen and S. Landau, "Polynomial decomposition algorithms," *Journal of Symbolic Computation*, vol. 7, pp. 445–456, May 1989.
- [13] R. M. Corless, M. W. Giesbrecht, D. J. Jeffrey, and S. M. Watt, "Approximate polynomial decomposition," in *Proceedings of ISSAC '99*, 1999, pp. 213–219.
- [14] M. Giesbrecht and J. May, "New algorithms for exact and approximate polynomial decomposition," in *Proceedings of the SNC Workshop*, July 2005.
- [15] M. Fried, "On a conjecture of Schur," *Michigan Mathematical Journal*, vol. 17, pp. 41–55, 1970.
- [16] G. Turnwald, "On schur's conjecture," *Journal of the Australian Mathematical Society (Series A)*, vol. 58, no. 03, pp. 312–357, 1995.
- [17] W. Ruppert, "Reducibility of polynomials  $f(x, y)$  modulo  $p$ ," *Journal of Number Theory*, vol. 77, no. 1, pp. 62–70, July 1999.
- [18] S. Gao, "Factoring multivariate polynomials via partial differential equations," *Mathematics of Computation*, vol. 72, pp. 801–822, 2003.
- [19] S. Gao, E. Kaltofen, J. May, Z. Yang, and L. Zhi, "Approximate factorization of multivariate polynomials via differential equations," in *Proceedings of the 2004 international symposium on Symbolic and algebraic computation*, ser. ISSAC '04, New York, NY, USA, 2004, pp. 167–174.
- [20] E. Kaltofen, J. P. May, Z. Yang, and L. Zhi, "Approximate factorization of multivariate polynomials using singular value decomposition," *Journal of Symbolic Computation*, vol. 43, no. 5, pp. 359–376, 2008.
- [21] P. Aubry and A. Valibouze, "Algebraic computation of resolvents without extraneous powers," *European Journal of Combinatorics*, vol. 33, no. 7, pp. 1369–1385, 2012.
- [22] J. Rosen, H. Park, and J. Glick, "Total least norm formulation and solution for structured problems," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 1, pp. 110–126, 1996.
- [23] B. De Moor, "Total least squares for affinely structured matrices and the noisy realization problem," *Signal Processing, IEEE Transactions on*, vol. 42, no. 11, pp. 3104–3113, nov 1994.
- [24] B. Botting, M. Giesbrecht, and J. May, "Using Riemannian SVD for problems in approximate algebra," in *Proceedings of the International Workshop of Symbolic-Numeric Computation*, 2005, pp. 209–219.
- [25] P. Lemmerling, *Structured total least squares: analysis, algorithms and applications. PhD Thesis*. Katholieke Universiteit Leuven, Belgium, 1999.