

CONTINUOUS AND DISCRETE FRACTIONAL FOURIER DOMAIN DECOMPOSITION

İ. Şamil Yetik¹, M. A. Kutay², Hakan Özaktas³, Haldun M. Ozaktas¹

¹ Bilkent University, Department of Electrical Engineering, TR 06533 Bilkent, Ankara, Turkey

² Drexel University, Department of Electrical and Computer Engineering, Philadelphia, PA 19104 USA

³ East Mediterranean University, Department of Industrial Engineering, Gazi Magusa, TRNC

ABSTRACT

We introduce the fractional Fourier domain decomposition for continuous and discrete signals and systems. A procedure called pruning, analogous to truncation of the singular-value decomposition, underlies a number of potential applications, among which we discuss fast implementation of space-variant linear systems.

1. INTRODUCTION

The continuous spectral decomposition (or expansion) and its discrete counterpart, the singular value decomposition (SVD), plays a fundamental role in signal and system analysis, representation and processing. The spectral decomposition of a function $h(u, u')$ is

$$h(u, u') = \int_{-\infty}^{\infty} \psi_v(u) \lambda_v \psi_v^*(u') dv, \quad (1)$$

where the λ_v are the eigenvalues and the $\psi_v(u)$ are the eigenfunctions of $h(u, u')$ (that is, they are solutions of the equation $\int_{-\infty}^{\infty} h(u, u') f(u') du' = \lambda f(u)$).

Turning our attention to the discrete case, the SVD of an arbitrary $N_r \times N_c$ complex matrix \mathbf{H} is

$$\mathbf{H}_{N_r \times N_c} = \mathbf{U}_{N_r \times N_r} \mathbf{\Sigma}_{N_r \times N_c} \mathbf{V}_{N_c \times N_c}^\dagger, \quad (2)$$

where \mathbf{U} and \mathbf{V} are unitary matrices. The superscript \dagger denotes Hermitian transpose. $\mathbf{\Sigma}$ is a diagonal matrix whose elements σ_j (the singular values) are the nonnegative square roots of the eigenvalues of $\mathbf{H}\mathbf{H}^\dagger$ and $\mathbf{H}^\dagger\mathbf{H}$. The number of strictly positive singular values is equal to the rank R of \mathbf{H} . The SVD can also be written in the form of an outer-product (or spectral) expansion

$$\mathbf{H} = \sum_{j=1}^R \sigma_j \mathbf{u}_j \mathbf{v}_j^\dagger \quad (3)$$

where \mathbf{u}_j and \mathbf{v}_j are the columns of \mathbf{U} and \mathbf{V} . It is common to assume that the σ_j are ordered in decreasing value.

In this paper we define the continuous and discrete fractional Fourier domain decomposition (FFDD). While the FFDD may not match the SVD's central importance, we believe it is of fundamental importance in its own right as an alternative which may offer complementary insight and understanding. Although exploring the full range of properties and applications of the FFDD is beyond the scope of this paper, we illustrate its usefulness by showing that it can be used for fast implementation of space-variant linear systems. We believe the FFDD has the potential to become a useful tool in signal and system analysis, representation, and processing (especially in time-frequency space), in some cases in a similar spirit to the SVD.

We refer the reader to [1, 2, 3] for an introduction to the fractional Fourier transform as well as an extensive list of references. Here we briefly mention a few vital properties. The a th order fractional Fourier transform $f_a(u) = \int_{-\infty}^{\infty} K_a(u, u') f(u') du'$ (the explicit form of $K_a(u, u')$ is given in the references) is unitary, the zeroth order fractional Fourier transform corresponds to the identity operation and the first order fractional Fourier transform corresponds to the ordinary Fourier transform. Furthermore, the fractional Fourier transform is index additive, that is, the a_1 th order fractional Fourier transform of the a_2 th order fractional Fourier transform is equal to the $(a_1 + a_2)$ th order fractional Fourier transform. The a th order fractional Fourier transform corresponds to a counter-clockwise rotation of the Wigner distribution of the function by an angle $\alpha = a\pi/2$ in the time-frequency plane. All of these properties are also satisfied by the discrete fractional Fourier transform [4, 5]. Finally, we note that the fractional Fourier transform can be simulated with an $O(N \log N)$ algorithm.

2. THE CONTINUOUS FRACTIONAL FOURIER DECOMPOSITION

Let $h(u, u')$ be a two-dimensional function, representing either an image or the kernel of a one-dimensional linear system. Its fractional Fourier domain decomposition is defined as

$$h(u, u') = \int_{-2}^2 \int_{-\infty}^{\infty} K_{-a}(u, u'') c(a, u'') K_a(u'', u') du'' da, \quad (4)$$

where $c(a, u'')$ is a family of one-dimensional weighting functions with parameter a . The integration interval is limited to $[-2, 2]$, since the fractional Fourier transform is periodic in a with period 4. Comparing the fractional Fourier decomposition with the spectral decomposition given in (1), we can see that the integrands in both expressions consist of three terms. The definition of the FFDD can be rewritten in the form

$$h(u, u') = \int_{-2}^2 \int_{-\infty}^{\infty} c(a, u'') P_a(u, u', u'') du'' da \quad (5)$$

where we have defined

$$P_a(u, u', u'') = K_a(u'', u') K_{-a}(u, u'') \quad (6)$$

Equation (5) can be interpreted as an expansion of $h(u, u')$ in terms of the basis functions $P_a(u, u', u'')$ with $c(a, u'')$ corresponding to the expansion coefficients.

The basis functions in (6) can easily be shown to be linearly independent as a direct consequence of the fact that $(K_a(u'', u), K_{a'}(u, u''))_u$ is nonzero for all a, a' . Here $(\cdot, \cdot)_u$ denotes a one-dimensional inner product with respect to the variable u .

3. THE DISCRETE FRACTIONAL FOURIER DOMAIN DECOMPOSITION

We now turn our attention to the discrete FFDD. Let \mathbf{H} be a complex $N_r \times N_c$ matrix and $\{a_1, a_2, \dots, a_N\}$ a set of $N = \max(N_r, N_c)$ distinct real numbers such that $-1 < a_1 < a_2 < \dots < a_N \leq 1$. For instance, we may take the a_k uniformly spaced in this interval. The corresponding fractional Fourier domains are illustrated in figure 1b. We define the FFDD of \mathbf{H} as

$$\mathbf{H}_{N_r \times N_c} = \sum_{k=1}^N \mathbf{F}_{N_r}^{-a_k} [\mathbf{\Lambda}_k]_{N_r \times N_c} (\mathbf{F}_{N_c}^{-a_k})^\dagger, \quad (7)$$

where $\mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \dots, \mathbf{\Lambda}_N$ are diagonal matrices each of whose $N' = \min(N_r, N_c)$ elements c_{kj} , $j = 1, 2, \dots, N'$ are in general complex numbers. It will sometimes be a convenient way to represent these diagonal elements

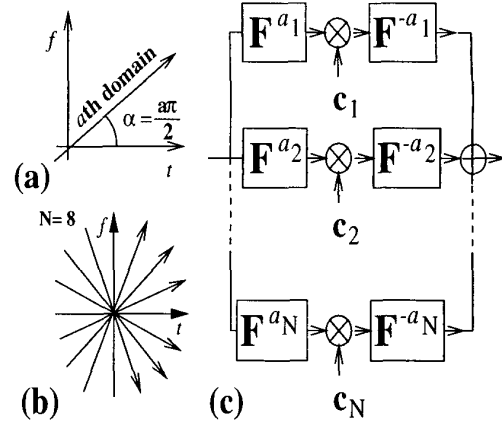


Figure 1: (a) The a th fractional Fourier domain. The $a = 0$ th and $a = 1$ st domains are the ordinary time (t) and frequency (f) domains. (b) N equally spaced fractional Fourier domains. (c) Block diagram of the FFDD.

$c_{k1}, c_{k2}, \dots, c_{kN'}$ for any k in the form of a column vector \mathbf{c}_k . When \mathbf{H} is Hermitian (skew-Hermitian), \mathbf{c}_k is real (imaginary). We also note that $(\mathbf{F}_{N_c}^{-a_k})^\dagger = \mathbf{F}_{N_c}^{a_k}$. The FFDD always exists and is unique, as will be discussed further below.

Comparing and contrasting the FFDD with the SVD will help gain insight into the FFDD. If we compare one term on the right-hand side of (7) with the right-hand side of (2), we see that they are similar in that they both consist of 3 terms of corresponding dimensionality, the first and third being unitary matrices and the second being a diagonal matrix. But whereas the columns of \mathbf{U} and \mathbf{V} constitute orthonormal bases specific to \mathbf{H} , the columns of $\mathbf{F}_{N_r}^{-a_k}$ and $\mathbf{F}_{N_c}^{-a_k}$ constitute orthonormal bases for the a_k th fractional Fourier domain. Customization of the decomposition is achieved through the coefficients c_{kj} (and perhaps also the orders a_k).

Denoting the j th columns of $\mathbf{F}_{N_r}^{-a_k}$ and $\mathbf{F}_{N_c}^{-a_k}$ as $[\mathbf{F}_{N_r}^{-a_k}]_j$ and $[\mathbf{F}_{N_c}^{-a_k}]_j$ respectively, it now becomes possible to write the k th term of the summation in (7) as an outer product of the form $\sum_{j=1}^{N'} c_{kj} [\mathbf{F}_{N_r}^{-a_k}]_j ([\mathbf{F}_{N_c}^{-a_k}]_j)^\dagger$ so that (7) can be rewritten as

$$\mathbf{H} = \sum_{k=1}^N \sum_{j=1}^{N'} c_{kj} [\mathbf{F}_{N_r}^{-a_k}]_j ([\mathbf{F}_{N_c}^{-a_k}]_j)^\dagger. \quad (8)$$

To a certain extent, the inner summation resembles the outer-product form of the SVD given in (3). The $N_r \times N_c$ matrices $[\mathbf{F}_{N_r}^{-a_k}]_j ([\mathbf{F}_{N_c}^{-a_k}]_j)^\dagger$ are of unit rank since they are the outer product of vectors. We will

denote these matrices by \mathbf{P}_{kj} so that

$$\mathbf{H} = \sum_{k=1}^N \sum_{j=1}^{N'} c_{kj} \mathbf{P}_{kj}. \quad (9)$$

This equation is simply an expansion of \mathbf{H} in terms of the *basis matrices* \mathbf{P}_{kj} , $1 \leq k \leq N$, $1 \leq j \leq N'$, where the c_{kj} serve as the weighting coefficients of the expansion.

When \mathbf{H} is a square matrix of dimension N , the FFDD takes the simpler form

$$\mathbf{H} = \sum_{k=1}^N \mathbf{F}^{-a_k} \mathbf{A}_k (\mathbf{F}^{-a_k})^\dagger, \quad (10)$$

where all matrices are $N \times N$.

Equation (9) is a linear relation between the matrices \mathbf{H} and c_{kj} with the four-dimensional tensor \mathbf{P}_{kj} representing the transformation between them. Let \mathcal{H} denote a column ordering of the matrix \mathbf{H} , with dimensions $N_c N_r \times 1$. Also let \mathcal{C} denote the $NN' \times 1$ column vector obtained by stacking the column vectors $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_N$ on top of each other. Notice that we always have $NN' = \max(N_r, N_c) \min(N_r, N_c) = N_r N_c$. These column orderings determine a corresponding ordering which converts the four-dimensional tensor (or two-dimensional array of matrices) \mathbf{P}_{kj} into a square matrix \mathcal{P} of dimensions $N_c N_r \times N_c N_r$. (The vector obtained as the column ordering of the matrix \mathbf{P}_{kj} for a specific kj , goes into the $[(k-1)N' + j]$ th column of the matrix \mathcal{P} .) Now, we can write (9) as the linear square matrix equation $\mathcal{H} = \mathcal{P}\mathcal{C}$. This equation will have a unique solution for \mathcal{C} and thus c_{kj} if and only if the columns of \mathcal{P} are linearly independent. Since the columns of \mathcal{P} are merely column orderings of the basis matrices \mathbf{P}_{kj} , this is the same as linear independence of these basis matrices. Recalling the definition of these matrices (just before (9)), their linear independence follows from the fact that the inner product of any column of \mathbf{F}^a with any column of $\mathbf{F}^{a'}$ ($a' \neq a$) is always nonzero. Thus *the FFDD always exists and is unique* (for given a_k).

4. APPLICATIONS

Let \mathbf{H} denote a linear matrix operator. Equation (7) represents a decomposition of this operator into N terms. Each term, taken by itself, corresponds to filtering in the a_k th fractional Fourier domain [6, 2], where an a_k th order forward transform is followed by multiplication with a filter function \mathbf{c}_k and concluded with an inverse a_k th order transform. If $a_k = 1$, this corresponds to ordinary Fourier domain filtering. If $a_k = 0$,

this corresponds to multiplication of a signal with a filter function directly in the time domain. All terms taken together, the FFDD can be represented by the block diagram shown in figure 1c and interpreted as the decomposition of an operator into fractional Fourier domain filters of different orders. An arbitrary linear system \mathbf{H} will in general not correspond to multiplicative filtering in the time or frequency domain or in any other single fractional Fourier domain. However, \mathbf{H} can always be expressed as a combination of filtering operations in different fractional domains. *A sufficient number of different-ordered fractional Fourier domain filtering operations "span" the space of all linear operations.* The fundamental importance of the FFDD is that it shows how an arbitrary linear system can be decomposed into this complete set of domains in the time-frequency plane.

If \mathbf{H} represents a time-invariant system, all filter coefficients except those corresponding to $a_k = 1$ will be zero. More generally, different domains will make varying contributions to the decomposition. By eliminating domains for which the coefficients $c_{k1}, c_{k2}, \dots, c_{kN'}$ are small, significant savings in storing and implementing \mathbf{H} becomes possible. This procedure, which we refer to as *pruning* the FFDD, is the counterpart of truncating the SVD. An alternative to this selective elimination procedure will be referred to as *sparsening*, in which we simply work with a more coarsely spaced set of domains.

In any event, the resulting smaller number of domains will be denoted by $M < N$. The upper limit of the summation in (7) is replaced by M and the equality is replaced by approximate equality, leading us to $\mathcal{H} \approx \mathcal{P}\mathcal{C}$. If we solve this in the least-squares sense, minimizing $\|\mathcal{H} - \mathcal{P}\mathcal{C}\|$, we can find the coefficients resulting in the best *M-domain approximation* to \mathbf{H} . This procedure amounts to projecting \mathbf{H} onto the subspace spanned by the M basis matrices, which now do not span the whole space.

Since the fractional Fourier transform can be computed in $O(N \log N)$ time, implementation of the pruned version of figure 1c takes $O(MN \log N)$ time. If an acceptable approximation to \mathbf{H} can be found with a relatively small value of M , this can be much smaller than the time $O(N_r N_c)$ associated with direct implementation of the linear system. Likewise, optical implementation requires a space-bandwidth product of $O(MN)$, as opposed to $O(N_r N_c)$ for direct implementation [7]. In passing, we note that the pruned FFDD is directly related to the concept of *parallel filtering* [8, 9].

As an example, we consider the problem of recovering a signal consisting of multiple chirp-like components, which is buried in white Gaussian noise such that

the signal-to-noise ratio is 0.1. We assume the signal consists of 6 chirps with uniformly distributed random amplitudes and time shifts, and that the chirp rates are known with a $\pm 5\%$ accuracy. We find that the general linear optimal Wiener filter \mathbf{H} for this problem can be approximated with a mean-square error of 5.2% by using only $M = 6$ domains. \mathbf{H} can also be approximated by truncating (3) to M terms, leading to an implementation time of $O(MN)$. For the present example, $M = 6$ results in an error of 20%, demonstrating an instance where the FFDD yields better accuracy than the SVD.

Next we consider restoration of images blurred by a space-varying point-spread function whose diameter increases linearly with position. This time we use the M -domain expansion as a constraint on the linear recovery filter and optimize directly over the coefficients c_{kj} to minimize the mean-square estimation error. The error is found to be 7% for $M = 5$. One may construct a similar constrained optimization problem by using the truncated SVD. However, this leads to a much more difficult nonlinear optimization problem because \mathbf{u}_j and \mathbf{v}_j in (3) are also unknowns, whereas the only unknowns in (7) are the Λ_k , leading to a linear optimization problem.

Other potential applications other than fast implementation of linear systems include data compression, statistically-optimum filtering, and regularization of ill-posed inverse problems, all of which may be based on the same basic idea of appropriately pruning or weighting the different domains.

We have not addressed the problem of optimally choosing the orders a_k , which corresponds to choosing the basis matrices. When $M = N$, the basis matrices form a complete set and any choice is acceptable. However, certain choices may offer better numerical stability. When $M < N$, the choice of a_k may reflect our knowledge about the ensemble of matrices \mathbf{H} we wish to approximate. This knowledge may be statistical or in the form of restrictions on the set of matrices possible. It is also possible to choose the orders optimally for a single specific matrix.

A natural extension of the FFDD would be the linear canonical domain decomposition (LCDD) based on linear canonical transforms [10].

5. REFERENCES

- [1] H. M. Ozaktas, M. A. Kutay and D. Mendlovic, "Introduction to the fractional Fourier transform and its applications", in: P. W. Hawkes, ed., *Advances in Imaging and Electron Physics 106*, Academic Press, San Diego, California, 1999, pp. 239-291.
- [2] H. M. Ozaktas, B. Barshan, D. Mendlovic, and L. Onural, "Convolution, filtering, and multiplexing in fractional Fourier domains and their relation to chirp and wavelet transforms", *J. Opt. Soc. Am. A* vol. 11, 1994, pp. 547-559.
- [3] L. B. Almeida, "The fractional Fourier transform and time-frequency representations", *IEEE Trans. Sig. Proc.* vol. 42, 1994, pp. 3084-3092.
- [4] C. Candan, M. A. Kutay, H. M. Ozaktas "The Discrete Fractional Fourier Transform" to appear in *IEEE Trans. Sig. Proc.*, 2000.
- [5] S. C. Pei and M. H. Yeh, "Improved discrete fractional Fourier transform", *Opt. Lett.*, vol. 22, no. 14, 1995, pp. 1047-1049.
- [6] M. A. Kutay, H. M. Ozaktas, O. Arikan, and L. Onural, "Optimal filtering in fractional Fourier domains", *IEEE Trans. Sig. Proc.* vol. 45, 1997, pp. 1129-1143.
- [7] H. M. Ozaktas and D. Mendlovic, "Fractional Fourier optics", *J. Opt. Soc. Am. A* vol. 12, 1995, pp. 743-751.
- [8] M. A. Kutay, M. F. Erden, H. M. Ozaktas, O. Arikan, C. Candan, and O. Guleryuz, "Cost-Efficient Approximation of Linear Systems with Repeated and Multi-channel Filtering Configurations," *Proc. IEEE Int. Conf. Acous., Speech and Sig. Proc.*, Seattle, May 12-15 1998.
- [9] M. A. Kutay, M. F. Erden, H. M. Ozaktas, O. Arikan, O. Guleryuz, and C. Candan, "Space-Bandwidth Efficient Realizations of Linear Systems", to appear in *Opt. Lett.*
- [10] B. Barshan, M. A. Kutay, and H. M. Ozaktas, "Optimal Filtering with Linear Canonical Transforms", *Opt. Commun.*, vol. 135, 1997, pp. 32-36.

- [1] H. M. Ozaktas, M. A. Kutay and D. Mendlovic, "Introduction to the fractional Fourier transform and its applications", in: P. W. Hawkes, ed., *Advances in Imaging and Electron Physics 106*, Aca-