

FFT Interpolation from Nonuniform Samples Lying in a Regular Grid

J. Selva

Abstract

This paper presents a method to interpolate a periodic band-limited signal from its samples lying at nonuniform positions in a regular grid, which is based on the FFT and has the same complexity order as this last algorithm. This kind of interpolation is usually termed “the missing samples problem” in the literature, and there exists a wide variety of iterative and direct methods for its solution. The one presented in this paper is a direct method that exploits the properties of the so-called erasure polynomial, and provides a significant improvement on the most efficient method in the literature, which seems to be the burst error recovery (BER) technique of Marvasti’s et al. The paper includes numerical assessments of the method’s stability and complexity.

I. INTRODUCTION

In a variety of applications, a band-limited signal is converted from the analog to the discrete domain, but some of the resulting samples are lost due to various causes. Then, the problem is to interpolate the lost samples from the available ones, assuming the average rate of the latter fulfills the Nyquist condition. Just to cite a few applications in which this problem arises, it is a task required whenever a sampled signal is sent through a packet network and there exist losses [1]. Also, it is a basic spectral estimation problem whenever a channel spectrum must be estimated from its nonuniform samples in OFDM systems [2], [3], (pilot-aided estimation). It is equivalent to the error calculation step for the so-called Bose-Chaudhuri-Hocquenghem (BCH) DFT codes, in which the coding is performed in the real field, before quantization, [4]–[6]. Finally, in time-interleaved analog-to-digital converters (TI-ADCs), some samples at arbitrary positions can be unavailable due to a jitter calibration process, and they must be recovered, [7], [8].

Copyright (c) 2015 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org. The author is with the Dept. of Physics, Systems Engineering and Signal Theory (DFISTS), University of Alicante, P.O.Box 99, E-03080 Alicante, Spain (e-mail: jesus.selva@ua.es). This work was supported by the Spanish Ministry of Economy and Competitiveness (MINECO) under Project TEC2011-28201-C02-02.

In sampling theory, this problem is usually termed “the missing samples problem”, and is addressed assuming the signal’s bandwidth is unknown but fulfills the Nyquist condition. The basic interpolation model is then the trigonometric one, i.e, the signal is viewed as a trigonometric polynomial, and the problem reduces to computing the polynomial’s coefficients and from them the missing samples [9], [10, Ch. 17]. As can be easily deduced, this task is equivalent to solving a linear system for which there exist various standard techniques. There are, however, two main issues. The first is the numerical stability, due to the fact that the round-off errors accumulate heavily if there is a large number of consecutive missing samples. The second is the complexity, given that the linear system size is large and the complexity order of the standard techniques depends cubically on it. Though there exist conventional stable techniques like the pseudo-inverse, no conventional technique seems able to deal with both the stability and complexity drawbacks. This situation has led to the development of various direct and iterative algorithms for recovering the lost samples during the last decades. Probably, the earliest solution in the literature was the Papoulis-Gerchberg algorithm [11], [12], which is an iterative method based on the FFT. Standard techniques like the conjugate gradient and Lagrange interpolation methods have also been employed [13, Sec. 3]. [1] presents another iterative method and several ways to speed up its convergence using extrapolation. The BER technique of Marvasti et al. in [13] seems to be the most efficient technique to date. The numerical experiments show that this technique is stable, and achieves the complexity order $O(NP)$, where N is the total number of samples and P the number of known ones. This complexity order is a clear improvement compared to the order of the standard methods which is $O(P^3)$.

The purpose of this paper is to present a new direct solution for the missing sampling problem whose complexity order is $O(N \log N)$. If a denotes the ratio of total to known samples N/P and is assumed constant, then the complexity of the BER technique is $O(N^2)$ while that of the proposed method is $O(N \log N)$. Thus, the proposed method provides a significant improvement in terms of complexity. Actually, its arithmetic operations count is up to factor twenty smaller than the corresponding count of the BER technique, for typical values of N . The method proposed in this paper is based on two theorems. The first gives a procedure for obtaining the missing samples which consists of two FFTs plus three weighting operations. The coefficients of two of the three weighting operations depend on the sampling positions, and thus the procedure is efficient but only usable if these last coefficients have been pre-computed. The second theorem provides a solution to this last shortcoming, by specifying a procedure to compute the weighting coefficients in just two FFTs plus the computation of one complex exponential per sample. The combination of these two theorems yields the proposed method whose complexity is $O(N \log N)$.

The efficient recovery of a band-limited signal from its own nonuniform samples has been a fundamental topic during the last decades. In [14], the authors proposed a filterbank approach for reducing the implementation complexity, while in [9] efficiency was achieved by processing the signal's samples block-wise. In [15], [16], the sampling instants varied sample-wise and the signal was assumed to have a known bandwidth strictly below the Nyquist rate. In these last references, the efficiency was the result of exploiting the properties of the Lagrange and barycentric interpolators. The method in the present paper assumes perfect knowledge of the samples' positions. Therefore, it can be a useful tool for more complex problems in which these positions are unknown, as those dealt with in areas like compressed sensing and random sampling [17]–[20], though is not directly applicable.

The paper has been organized as follows. In the next two sub-sections, we introduce the notation and recollect several basic results about periodic signals and the FFT. Then, in Sec. II we introduce the missing samples problem, comment on standard solution methods like the pseudo-inverse, and discuss in detail the BER technique. Afterward, we present in Secs. III and IV the two theorems making up the proposed method. The complexity order of the standard pseudo-inverse, BER and proposed methods are then discussed in Sec. V. Finally, these methods are compared in terms of numerical stability and computational burden in Sec. VI.

A. Notation and basic concepts

We will employ the following notation:

- Throughout the paper, $t \in \mathbb{R}$ will denote the time variable, and n, p and q integer variables.
- Definitions of new symbols and functions will be written using ' \equiv '.
- Vectors will be denoted in lower-case bold face, (\mathbf{s} , \mathbf{d}), and matrices in upper-case bold face (\mathbf{A} , $\mathbf{\Phi}$).
- \mathbf{A}^\dagger is the pseudo-inverse of \mathbf{A} .
- $[\mathbf{v}]_n$ will denote the n th component of vector \mathbf{v} .
- For integer $M \geq 0$, I_M will denote the set

$$\{0, 1, \dots, M-1\}.$$

- For any signal or sequence f and a set of integers A , $\{f(n), n \in A\}$ will denote the set of samples $f(n)$ taken at positions $n \in A$. With this definition, we jointly give information about the values $f(n)$ and their corresponding sampling positions n . A rigorous definition is

$$\{f(n), n \in A\} \equiv \{(n, f(n)) : n \in A\},$$

i.e., $\{f(n), n \in A\}$ is the set of pairs formed by the sampling positions and their corresponding values.

- $\text{DFT}\{\mathbf{v}\}$ and $\text{IDFT}\{\mathbf{v}\}$ will respectively denote the DFT and IDFT of vector \mathbf{v} , but computed using a fast algorithm based on the FFT. Note that there exist fast algorithms of this kind for any vector length, like the chirp transform, [21, Sec. 6.9.2].
- For two sets, A and B , $A \setminus B$ will denote the set of elements in A not in B .
- $\mathbf{a} \odot \mathbf{b}$ will denote the component-wise product of \mathbf{a} and \mathbf{b} , i.e., for \mathbf{a} and \mathbf{b} of equal length, $[\mathbf{a} \odot \mathbf{b}]_n = [\mathbf{a}]_n [\mathbf{b}]_n$.
- For two N -period discrete sequences, $a(n)$ and $b(n)$, $(a \stackrel{N}{*} b)(n)$ will denote their cyclic convolution, defined by

$$(a \stackrel{N}{*} b)(n) \equiv \sum_{p=q}^{q+N-1} a(p)b(n-p),$$

where q can be any integer, given that $a(n)$ and $b(n)$ have period N . The cyclic convolution can be efficiently evaluated using the FFT by means of the formula

$$(a \stackrel{N}{*} b)(n) = [\text{IDFT}\{\text{DFT}\{\mathbf{a}\} \odot \text{DFT}\{\mathbf{b}\}\}]_{n+1}, \quad n \in I_N, \quad (1)$$

where

$$[\mathbf{a}]_{n+1} \equiv a(n), \quad [\mathbf{b}]_{n+1} \equiv b(n), \quad n \in I_N.$$

B. The basic set of trigonometric polynomials and two evaluation procedures based on the FFT

In the paper, a key concept will be the set of trigonometric polynomials of the form

$$v(t) = \sum_{p=0}^{N-1} V_p e^{j2\pi pt/N}, \quad (2)$$

where $N > 0$ is the number of polynomial coefficients, $V_p \in \mathbb{C}$ and $t \in \mathbb{R}$. This set will be denoted \mathcal{F}_N , and we will exploit two evaluation procedures for its elements based on the DFT.

The first procedure is the usual way to switch the time and frequency domains using the DFT/IDFT pair. More precisely, for $v(t) \in \mathcal{F}_N$ we may group the time samples and coefficients in corresponding vectors, \mathbf{v} and $\tilde{\mathbf{v}}$,

$$[\mathbf{v}]_{n+1} \equiv v(n), \quad [\tilde{\mathbf{v}}]_{p+1} \equiv V_p, \quad (n, p \in I_N), \quad (3)$$

and then obtain $\tilde{\mathbf{v}}$ from \mathbf{v} and vice versa using the equations

$$\begin{aligned} \mathbf{v} &= N \text{IDFT}\{\tilde{\mathbf{v}}\}, \\ \tilde{\mathbf{v}} &= \frac{1}{N} \text{DFT}\{\mathbf{v}\}. \end{aligned}$$

Also, we may express these relations as

$$\{v_n, n \in I_N\} \xrightarrow{\text{DFT}} \{NV_p, p \in I_N\}. \quad (4)$$

The second procedure is the method to differentiate $v(t) \in \mathcal{F}_N$ using the DFT/IDFT pair, and exploits the fact that \mathcal{F}_N is closed under differentiation, i.e, if $v(t) \in \mathcal{F}_N$ then $v'(t) \in \mathcal{F}_N$. This last property is obvious since the Fourier coefficients of $v'(t)$ are $\{j2\pi pV_p/N, p \in I_N\}$. If \mathbf{v}' denotes the derivative samples vector of $v(t)$,

$$[\mathbf{v}']_{n+1} \equiv v'(n), n \in I_N,$$

then the procedure is expressed by the formula

$$\mathbf{v}' = \text{IDFT}\{\text{DFT}\{\mathbf{v}\} \odot \mathbf{d}\}, \quad (5)$$

where

$$[\mathbf{d}]_{p+1} \equiv j2\pi p/N, p \in I_N. \quad (6)$$

II. THE MISSING SAMPLES PROBLEM

A basic interpolator for a band-limited signal $s_o(t)$ is the trigonometric one, i.e, it consists of viewing $s_o(t)$ as a trigonometric polynomial of the form

$$s_o(t) \approx \sum_{p=p_1}^{p_1+P-1} S_{o,p} e^{j2\pi p t/T}, \quad (7)$$

where we assume that $s_o(t)$ is interpolated in the range $[0, T]$ with $T > 0$, $S_{o,p}$ denotes the p th coefficient, p_1 the first polynomial index, and $P > 0$ the number of coefficients. If (7) is sufficiently accurate and $s_o(t)$ is sampled with period T/N for integer $N \geq P$, then it is well-known that the coefficients $S_{o,p}$ and the value of $s_o(t)$ at any $t \in [0, T]$ can be efficiently computed from the set of samples $\{s_o(nT/N), n \in I_N\}$ using algorithms from the FFT family [22].

In some applications, however, $N - P$ samples from the set $\{s_o(nT/N), n \in I_N\}$ are lost due to various causes, and then the problem consists of recovering these missing samples from the known ones in a numerically stable way and with low complexity. More precisely, if J denotes the indices n of the known samples, then J has P elements and the objective is to obtain the samples $\{s_o(nT/N), n \in J^c\}$, where J^c is the complement of J relative to I_N ,

$$J^c \equiv I_N \setminus J.$$

In this problem, the initial index p_1 and time period T are irrelevant, given that we may scale $s_o(t)$ so that its period is N and its first frequency is zero. So, in order to simplify the notation, we may state the problem in terms of the following normalized signal

$$s(t) \equiv s_o(tT/N)e^{-j2\pi p_1 t/N}.$$

From (7) we have that $s(t)$ has the form

$$s(t) = \sum_{p=0}^{P-1} S_p e^{j2\pi p t/N}, \quad (8)$$

where $S_p \equiv S_{o,p+p_1}$. In terms of $s(t)$, the problem consists of computing the samples $\{s(n), n \in J^c\}$, assuming the samples $\{s(n), n \in J\}$ are known.

As can be readily checked, the solution to this problem just involves the inversion of the linear system

$$s(n) = \sum_{p=0}^{P-1} S_p e^{j2\pi p n/N}, \quad n \in J, \quad (9)$$

in which the unknowns are the coefficients S_p , followed by the computation of the desired samples using (8) for $t \in J^c$. This inversion can in principle be tackled using conventional linear algebra techniques. However, (9) is ill conditioned if there are long sequences of missing samples, and it is then necessary to resort to a method ensuring stability. The usual one is based on the pseudo-inverse and its first step consists of computing estimates $\{\hat{S}_{ps,p}, p \in I_P\}$ of the coefficients $\{S_p, p \in I_P\}$ through the equation

$$\tilde{\mathbf{s}}_{ps} = \mathbf{\Phi}^\dagger \mathbf{s}, \quad (10)$$

where

$$[\tilde{\mathbf{s}}_{ps}]_{p+1} \equiv \hat{S}_{ps,p}, \quad [\mathbf{\Phi}]_{k,p+1} \equiv e^{j2\pi p n(k)/N},$$

$$[\mathbf{s}]_k \equiv s(n(k)), \quad (p, k \in I_P),$$

and $n(k)$ runs through the elements of J for $k \in I_P$. Then, $\{s(n), n \in J^c\}$ can be readily obtained by evaluating (8) with $\hat{S}_{ps,p}$ in place of S_p for $t \in J^c$. The main drawback of conventional methods like this one is their $O(P^3)$ complexity order, which is too high for most practical applications. Specifically, in the pseudo-inverse method just stated, the computation of $\mathbf{\Phi}^\dagger$ in (10) involves one singular value decomposition (SVD) with $O(P^3)$ complexity. This drawback has led to the development of a variety of iterative and non-iterative methods with lower complexity during the last decades; (see [10, Ch. 17] for a review on this topic).

In [13], Marvasti et al. presented the so-called BER technique for this problem whose complexity order is $O(NP)$. This order is a clear improvement compared to the $O(P^3)$ order of the standard solutions,

and relative to other methods like the Lagrange interpolation and conjugate gradient methods. The key of the BER method consists of two relations between the following three polynomials:

- $s_J(t)$: Element of \mathcal{F}_N such that $s_J(n) = s(n)$ if $n \in J$ and $s_J(n) = 0$ if $n \in J^c$.
- $s_{J^c}(t)$: Polynomial with the same definition as $s_J(t)$ but with J and J^c switched.
- $\phi(t)$: Erasure polynomial. This is the monic element of \mathcal{F}_N of degree $N - P$ with one simple zero at each missing sample instant (set J^c), i.e, the polynomial

$$\phi(t) \equiv \prod_{n \in J^c} (e^{j2\pi t/N} - e^{j2\pi n/N}). \quad (11)$$

To introduce the first relation in the BER technique, note that to compute the desired samples $\{s(n), n \in J^c\}$ is equivalent to compute $\{s_{J^c}(n), n \in I_N\}$, given that $s_{J^c}(n) = 0$ if $n \in J$. Additionally, from the definitions of $s_J(t)$ and $s_{J^c}(t)$, it is clear that

$$s(n) = s_J(n) + s_{J^c}(n), \quad n \in I_N. \quad (12)$$

This equation can be written in the coefficients (frequency) domain using (4),

$$S_p = S_{J,p} + S_{J^c,p}, \quad p \in I_N, \quad (13)$$

where $S_{J,p}$ and $S_{J^c,p}$ respectively denote the Fourier coefficients of $s_J(t)$ and $s_{J^c}(t)$. But $S_p = 0$ if $P \leq p < N$ and, therefore, (13) implies

$$S_{J^c,p} = -S_{J,p}, \quad P \leq p < N. \quad (14)$$

So, the DFT of the samples $\{s_J(n), n \in I_N\}$ gives partial information about $s_{J^c}(t)$, namely its coefficients $S_{J^c,p}$ for $P \leq p < N$.

The second relation links $s_{J^c}(t)$ with the erasure polynomial $\phi(t)$ and is the following

$$s_{J^c}(n)\phi(n) = 0, \quad n \in I_N. \quad (15)$$

This relation is also a direct consequence of the definitions of $s_{J^c}(t)$ and $\phi(t)$, given that $J \cup J^c = I_N$, $s_{J^c}(n) = 0$ if $n \in J$, and $\phi(n) = 0$ if $n \in J^c$. If we take (15) to the frequency domain using the DFT (4), we have that (15) is turned into a cyclic convolution of the coefficients of $s_{J^c}(t)$ and $\phi(t)$. More precisely, we have

$$\sum_{p=0}^{N-P} \phi_{N-P-p} S_{J^c,q+p} = 0, \quad q \in \mathbb{Z}, \quad (16)$$

where ϕ_p denotes the coefficients of $\phi(t)$, and we take $S_{J^c,p}$ as a periodic sequence, i.e, $S_{J^c,p+N} = S_{J^c,p}$, $p \in \mathbb{Z}$. This second relation can be written as a recursive formula for computing $S_{J^c,q}$, if $S_{J^c,q+p}$ is known for $1 \leq p < N - P$. For this, just note from (11) that $\phi_{N-P} = 1$ and solve for $S_{J^c,q}$ in (16),

$$S_{J^c,q} = - \sum_{p=1}^{N-P} \phi_{N-P-p} S_{J^c,q+p}. \quad (17)$$

We have that (14) already provides the coefficients $S_{J^c, q+p}$ in this sum if $q = P$. So we may recursively apply this last formula for $q = P, P-1, \dots, 0$, in order to compute the missing coefficients $S_{J^c, q}$, $0 \leq q < P$.

Finally, from $\{S_{J^c, p}, p \in I_N\}$ we obtain the desired samples $\{s_{J^c}(n), n \in J^c\}$ through one inverse DFT,

$$s(n) = s_{J^c}(n) = [\text{IDFT}\{\mathbf{s}_{J^c}\}]_{n+1}, \quad n \in J^c, \quad (18)$$

where

$$[\tilde{\mathbf{s}}_{J^c}]_{p+1} \equiv S_{J^c, p}, \quad p \in I_N.$$

We can see in this method that the insertion of zeros, either in a vector or using the erasure polynomial, is the key to obtaining an efficient solution. Actually, the zero insertion in the definitions of $s_J(t)$ and $s_{J^c}(t)$ permits the use of the DFT in going from (12) to (13). And the multiplication by the erasure polynomial in (15) produces a zero sequence and the corresponding cyclic convolution in (16). There is, however, a more powerful way to insert zeros, that leads to a method entirely based on the DFT and weighting operations with complexity $O(N)$. The method is based on considering the properties of the signal $s(t)\phi(t)$ and is presented in the next section. It yields the desired samples in just two DFTs, if some samples of $\phi(t)$ and its derivative are known.

III. PROPOSED METHOD FOR FIXED SAMPLING POSITIONS

We present in the sequel a theorem specifying the first part of the proposed method. More precisely, it gives an efficient method to compute the desired samples $\{s(n), n \in J^c\}$ assuming that the samples $\{s(n), n \in J\}$, index set J , and values $\{\phi(n), n \in J\}$ and $\{\phi'(n), n \in J^c\}$ are known. The re-computation of these last two sets of values will be addressed in the next section. The theorem is the following.

Theorem 1. *The desired samples are given by the formula*

$$s(n) = \frac{1}{\phi'(n)} [\text{IDFT}\{\text{DFT}\{\mathbf{s}_\phi\} \odot \mathbf{d}\}]_{n+1}, \quad n \in J^c, \quad (19)$$

where \mathbf{d} was defined in (6) and

$$[\mathbf{s}_\phi]_{n+1} \equiv \begin{cases} s(n)\phi(n) & \text{if } n \in J \\ 0 & \text{if } n \in J^c. \end{cases} \quad (20)$$

The implementation of the procedure specified in this theorem is straight-forward. First, it is necessary to form the nonuniformly zero-padded vector \mathbf{s}_ϕ in (20), and then perform the steps specified in (19), i.e.,

- 1) Compute the DFT of \mathbf{s}_ϕ .

- 2) Weight the result component-wise using \mathbf{d} .
- 3) Compute the inverse DFT.
- 4) Multiply the samples with $n \in J^c$ by $1/\phi'(n)$.

If what is required is the set of Fourier coefficients $\{S_p, p \in I_P\}$, then they can be computed from $\{s(n), n \in I_N\}$, through one FFT using the formula

$$S_p = \text{DFT}\{\mathbf{s}_1\}, p \in I_P,$$

with

$$[\mathbf{s}_1]_{n+1} \equiv s((N/Q)n), n \in I_Q,$$

where Q is the smallest divisor of N such that $Q \geq P$.

Proof of theorem 1. Consider the signal

$$s_\phi(t) \equiv s(t)\phi(t)$$

and two key facts related with it. The first is that we know its value at all instants in the regular grid I_N . This is so because either $n \in J$ and then both factors of the product $s(n)\phi(n)$ are known, or $n \in J^c$ and then $s(n)\phi(n) = 0$ because $\phi(n) = 0$. As a consequence, we have enough information to form the vector \mathbf{s}_ϕ in (20), akin to \mathbf{v} in (3), given that the only samples of $s(t)$ appearing in (20) are the known ones, $[s(n), n \in J]$.

The second fact is that $s_\phi(t)$ belongs to \mathcal{F}_N . We can see that this is so if we view (8) and (11) as polynomials in the variable $z = e^{j2\pi t/N}$. Since the right-hand side of (8) has degree $P - 1$ and (11) has degree $N - P$ (number of elements of J^c), then $s(t)\phi(t)$ has degree $N - 1$ in z . In other words, $s_\phi(t)$ has the form in (2). As a consequence, we may compute the derivative samples of $s_\phi(t)$ using (5). We have

$$\mathbf{s}'_\phi = \text{IDFT}\{\text{DFT}\{\mathbf{s}_\phi\} \odot \mathbf{d}\}, \quad (21)$$

where

$$[\mathbf{s}'_\phi]_{n+1} \equiv s'_\phi(n), n \in I_N.$$

Finally, the product differentiation rule allows us to obtain the desired samples $s(n), n \in J^c$, from \mathbf{s}'_ϕ , given that $\phi(t)$ has placed zeros at the instants $n \in J^c$. Specifically, since $\phi(n) = 0$ if $n \in J^c$, we have

$$s'_\phi(n) = s'(n)\phi(n) + s(n)\phi'(n) = s(n)\phi'(n).$$

So, solving for $s(n)$ we obtain

$$s(n) = \frac{s'_\phi(n)}{\phi'(n)} = \frac{[\mathbf{s}'_\phi]_{n+1}}{\phi'(n)}, n \in J^c.$$

Note that the division by $\phi'(n)$ is valid because the instants $n \in J^c$ are simple zeros of $\phi(n)$. The theorem's formula in (19) is the result of substituting (21) into this last equation. \square

IV. COMPUTATION OF THE ERASURE POLYNOMIAL WEIGHTS $\phi(n)$ AND $\phi'(n)$

In Theorem 1, the samples of $\phi(n)$ and $\phi'(n)$ depend on the sampling scheme and, therefore, they must be re-computed whenever the set J changes. If this re-computation is performed using (11) directly, then the cost of obtaining $\{\phi(n), n \in J\}$ is $O((N - P)P)$. As to the samples $\{\phi'(n), n \in J^c\}$, they can be computed from the derivative of (11),

$$\phi'(t) = \frac{j2\pi}{N} e^{j2\pi t/N} \sum_{k \in J^c} \prod_{n \in J^c \setminus \{k\}} (e^{j2\pi t/N} - e^{j2\pi n/N})$$

with complexity $O((N - P)^2(N - P - 1))$. These complexities are too high for real-time systems. The following theorem presents a method to compute these values with complexity $O(N \log N)$. It involves the computation of two size- N FFTs and N complex exponentials.

Theorem 2. Consider the N -period sequence specified by

$$\alpha(n) \equiv \begin{cases} \log(1 - e^{-j2\pi n/N}), & 1 \leq n < N \\ 0, & n = 0 \end{cases} \quad (22)$$

and $\alpha(n + N) = \alpha(n)$, $n \in \mathbb{Z}$. Let $\beta(n)$ denote the cyclic convolution

$$\beta(n) \equiv (1_{J^c} * \alpha)(n), \quad (23)$$

where $1_{J^c}(n)$ is the cyclic indicator sequence for J^c , defined by

$$1_{J^c}(n) \equiv \begin{cases} 1 & \text{if } n \in J^c \\ 0 & \text{if } n \in J \end{cases} \quad (24)$$

and $1_{J^c}(n) = 1_{J^c}(n + N)$, $n \in \mathbb{Z}$. The samples of $\phi(t)$ and $\phi'(t)$ required in theorem 1 are given by

$$\phi(n) = \exp\left(-\frac{j2\pi nP}{N} + \beta(n)\right), \quad n \in J, \quad (25)$$

$$\phi'(n) = \frac{j2\pi}{N} \exp\left(-\frac{j2\pi nP}{N} + \beta(n)\right), \quad n \in J^c. \quad (26)$$

Note that the sequence $\alpha(n)$ is independent of the sampling scheme and, therefore, it can be computed offline. This theorem implies that the computation of the required samples of $\phi(n)$ and $\phi'(n)$ just requires the cyclic convolution in (23) and the computation of one complex exponential per sample. Since the cyclic convolution can be performed using the FFT [Eq. (1)], the total computational cost is $O(N \log N)$. In computing the cyclic convolution, the DFT of the sequence $\alpha(n)$ can be spared, given that it can be performed offline. So to update $\phi(n)$ and $\phi'(n)$ just requires two FFTs.

Proof of theorem 2. Let us write (11) in terms of $\alpha(n)$, taking into account that J^c has $N - P$ elements.

If $n \in J$, we have:

$$\phi(n) = \prod_{k \in J^c} (e^{j2\pi n/N} - e^{j2\pi k/N}) \quad (27)$$

$$\begin{aligned} &= \prod_{k \in J^c} (e^{j2\pi n/N} (1 - e^{j2\pi(k-n)/N})) \\ &= \prod_{k \in J^c} (e^{j2\pi n/N} e^{\alpha(n-k)}) \\ &= e^{j2\pi n(N-P)/N} \prod_{k \in J^c} e^{\alpha(n-k)} \\ &= \exp \left(-\frac{j2\pi nP}{N} + \sum_{k \in J^c} \alpha(n-k) \right). \end{aligned} \quad (28)$$

In this last step, note that the summation is the cyclic convolution of $\alpha(n)$ with the indicator sequence of J^c in (24); i.e.,

$$\sum_{k \in J^c} \alpha(n-k) = (1_{J^c} *^N \alpha)(n),$$

and we have from (28)

$$\phi(n) = \exp \left(-\frac{j2\pi nP}{N} + (1_{J^c} *^N \alpha)(n) \right), \quad n \in J. \quad (29)$$

Thus we have proved (25).

For deriving (26), we must consider first the signal $\phi_1(t)$ with the same definition as $\phi(t)$ in (11), but with J in place of J^c , i.e., the signal

$$\phi_1(t) \equiv \prod_{n \in J} (e^{j2\pi t/N} - e^{j2\pi n/N}). \quad (30)$$

For $\phi_1(t)$, we may repeat the derivations in (27) to (29) already performed for $\phi(t)$ and, as can be easily checked, the result is the formula in (29) but with J and J^c switched and $N - P$ in place of P in the first term of the exponent. Specifically, we obtain

$$\phi_1(n) = \exp \left(-\frac{j2\pi n(N-P)}{N} + (1_J *^N \alpha)(n) \right), \quad n \in J^c, \quad (31)$$

where $1_J(n)$ is the indicator sequence of J , defined by $1_J(n+N) = 1_J(n)$, $n \in \mathbb{Z}$, and

$$1_J(n) \equiv \begin{cases} 1 & \text{if } n \in J \\ 0 & \text{if } n \in J^c. \end{cases}$$

Next, we require two simple results about $\alpha(n)$ and the indicators $1_{J^c}(n)$ and $1_J(n)$. The first is the property

$$\sum_{n=0}^{N-1} \alpha(n) = \log(N), \quad (32)$$

which is proved in Ap. A. The second is the fact that we may write (31) in terms of $1_{J^c}(n)$ rather than $1_J(n)$, because these two indicator functions are complementary; i.e, since $J \cup J^c = I_N$ and $J \cap J^c = \emptyset$, we have

$$1_{\text{all}}(n) = 1_J(n) + 1_{J^c}(n), \quad (33)$$

where $1_{\text{all}}(n)$ is the all-ones sequence.

Now, using (32) and (33) we have that $(1_J * \alpha)(n)$ can be obtained from $(1_{J^c} * \alpha)(n)$:

$$\begin{aligned} (1_J * \alpha)(n) &= (1_{\text{all}} * \alpha)(n) - (1_{J^c} * \alpha)(n) \\ &= \log(N) - (1_{J^c} * \alpha)(n). \end{aligned}$$

And substituting this formula into (31), we obtain a result of the form in (29) but for $\phi_1(n)$,

$$\phi_1(n) = N \exp \left(-\frac{j2\pi n(N-P)}{N} - (1_{J^c} * \alpha)(n) \right), \quad n \in J^c. \quad (34)$$

Let us derive the formula for $\phi'(n)$, $n \in J^c$. For this, consider the product $\phi(t)\phi_1(t)$. From (27) and (30), we have that this product is a monic polynomial whose root set is $\{e^{j2\pi n/N}, n \in I_N\}$, given that $J \cup J^c = I_N$ and $J \cap J^c = \emptyset$. So, we have

$$\begin{aligned} \phi(t)\phi_1(t) &= \prod_{n=0}^{N-1} (e^{j2\pi t/N} - e^{j2\pi n/N}) \\ &= \prod_{n=0}^{N-1} (z - e^{j2\pi n/N}) \Big|_{z=e^{j2\pi t/N}} \\ &= (z^N - 1) \Big|_{z=e^{j2\pi t/N}} = e^{j2\pi t} - 1. \end{aligned} \quad (35)$$

In this derivation, we have used the fact that the monic N th-order polynomial with root set $\{e^{j2\pi n/N}, n \in I_N\}$ is $z^N - 1$.

Next, let us apply the product differentiation rule to the equation derived in (35),

$$\phi(t)\phi_1(t) = e^{j2\pi t} - 1,$$

at $t = n$, $n \in J^c$. For its left-hand side, we have

$$\begin{aligned} (\phi(t)\phi_1(t))'_{t=n} &= (\phi'(t)\phi_1(t) + \phi(t)\phi_1'(t))_{t=n} \\ &= \phi'(n)\phi_1(n), \end{aligned} \quad (36)$$

given that $\phi(n) = 0$ if $n \in J^c$. And for its right-hand side, we have

$$(e^{j2\pi t} - 1)'_{t=n} = j2\pi. \quad (37)$$

Operation	Flops
Real sum	1
Complex sum	2
Real multiplication	1
Complex multiplication	6
Complex exponential	7
Size- N FFT, IFFT	$5N \log_2 N$

Fig. 1. Flop counts for basic operations.

So, the combination of (36) and (37) yields

$$\phi'(n) = \frac{j2\pi}{\phi_1(n)}, \quad n \in J^c.$$

Finally, substituting (34) into this last formula we obtain

$$\phi'(n) = \frac{j2\pi}{N} \exp \left(-\frac{j2\pi nP}{N} + (1_{J^c} * \alpha)(n) \right), \quad n \in J^c,$$

which is (26). □

V. COMPLEXITY ANALYSIS

In this section, we present counts of the number of floating point operations (flops) for the pseudo-inverse, BER, and proposed methods. Since the complexity of basic operations like multiplication and complex exponential may vary wildly with the hardware implementation, we have employed the convention in Fig. 1 for measuring the complexity.

The pseudo-inverse method consists of two steps with the following flop counts:

- Computation of coefficients $\{\tilde{S}_{ps,p}, p \in I_P\}$ from (10) using a complex singular value decomposition (SVD), [23, p. 293]: $96P^3$.
- Evaluation of (8) for $t \in J^c$: $(N - P)(8P - 1)$.

Thus, the total cost of the pseudo-inverse method is

$$96P^3 + (N - P)(8P - 1). \quad (38)$$

The flop count of each step in the BER technique, as explained in Sec. II, is the following:

- Computation of $\{\phi(n), n \in J\}$ using (11),

$$10P(N - P) - 11P + 3.$$

- DFT of the sequence $\{\phi(n), n \in I_N\}$ for obtaining the coefficients $\{\phi_p, p \in I_{N-P}\}$ in (17): $5N \log_2 N$.
- DFT of sequence $\{s_J(n), n \in I_N\}$, for computing $\{S_{J,p}, p \in I_N\}$: $5N \log_2 N$.
- Computation of recursive formula in (17),

$$8P(N - P) - P.$$

- Inverse DFT for obtaining the final result $\{s_{J^c}(n), n \in I_N\}$: $5N \log_2 N$.

The total cost of the BER technique is the following

$$18P(N - P) - 12P + 3 + 15N \log_2 N. \quad (39)$$

The implementation of the proposed method has the following flop counts:

- Computation of $\{\alpha(n), n \in I_N\}$ in (22). We assume zero cost for this operation, given that it can be performed offline.
- Computation of $\{\beta(n), n \in I_N\}$ in (23). This operation involves two FFTs plus N complex multiplications. The cost is

$$10N \log_2 N + 8N - 1.$$

- Computation of samples $\phi(n), n \in J$, in (20) using (25). We assume the factor $-j2\pi P/N$ in the exponent of (25) has been pre-computed. The cost of this operation is

$$18P - 3.$$

- Computation of second factor in (19). This operation involves two DFTs and N real-to-complex products with total cost

$$10N \log_2 N + 4N - 2.$$

Computation of $\{1/\phi'(n), n \in J^c\}$ from (26), and product with the output of the previous step. The cost is

$$20(N - P) - 4.$$

The total cost of the proposed technique is the following

$$20 \log_2 N + 32N - 2P - 10.$$

By comparing (39) with this last equation, we can readily see that the complexity of the proposed method is free of quadratic terms, while the complexity of the BER techniques is dominated by these terms whenever P is separated from 0 and N . From (38), we can see that the pseudo-inverse has the highest complexity order among the three methods, $O(P^3)$.

VI. NUMERICAL EXAMPLES

A. Numerical stability

The linear system in (9) is ill conditioned if there are long sequences of missing samples. This implies that a given method can be mathematically correct but not usable in practice, due to the accumulation of round-off errors. A well-known instance of such method is the Gaussian elimination procedure. For the proposed method, we can see from theorems 1 and 2 that it consists of a concatenation of a small number of stable operations. Specifically, there are three weighting operations, four DFTs, and N complex exponentials, and all these operations are stable numerically. This fact suggests that the proposed method is stable, though this assertion must be validated numerically. We perform this validation in the sequel by comparing the following three methods using double precision arithmetic:

- *Pseudo-inverse method*: Solution based on (10) and (8).
- *BER technique*: Combination of (14), (17), and (18).
- *Proposed method*: Method in theorem 1 using the computation procedure for $\phi(n)$ and $\phi'(n)$ in theorem 2.

In the examples that follow, we employ test signals of the form in (8) with $S_p = S_{R,p} + jS_{I,p}$, where $S_{R,p}$ and $S_{I,p}$ are independent realizations of a uniform distribution in the interval $[-1, 1]$. The figures are based on 100 Monte Carlo trials.

We present two examples. In the first, we assume a sampling grid which is the result of shifting the samples of a uniform grid (jittered sampling). In the second, we address the extrapolation problem, i.e., the sampling grid has a long gap that must be filled.

1) *Round-off error for a jittered sampling scheme*: In this example, we fix an oversampling factor $a = 8$ and relate N and P through $N = aP$. Then, we take sampling instants $t_p = 8p + u_p$, $p = 0, 1, \dots, P-1$ where u_p is randomly taken from the set $\{0, 1, \dots, a-1\}$ with uniform distribution (jittered sampling). Fig. 2 shows the round-off error versus the number of output samples N . The ordinate in this figure is the largest error among the $N - P$ interpolated samples and all trials. The proposed method improves on the BER technique slightly, and these two methods show a slight accuracy loss (one to two decimal digits) compared to the pseudo-inverse solution. The error of the proposed method is sufficiently small for most applications.

2) *Round-off error for extrapolation*: In this example, we fix $N = 64$ and take as input samples those at $t = 0, 1, \dots, P-1$. The objective is to interpolate the signal at $t = P, P+1, \dots, N-1$. Fig. 3 shows the maximum round-off error versus P among all interpolated samples and trials. Note that there is little

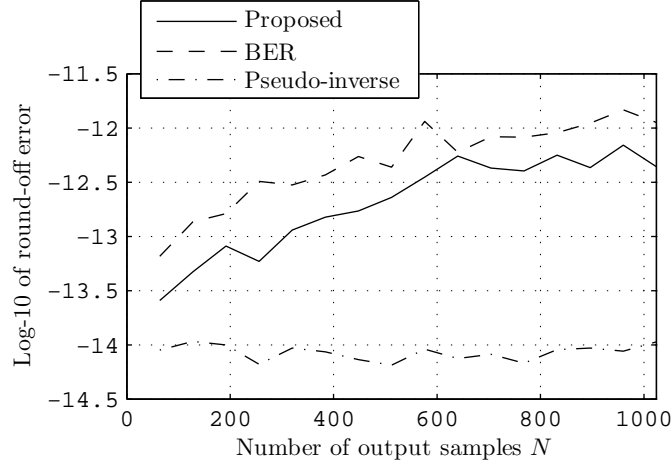


Fig. 2. Maximum round-off error among all interpolated samples and trials versus number of output samples (N), for the proposed, BER and pseudo-inverse methods.

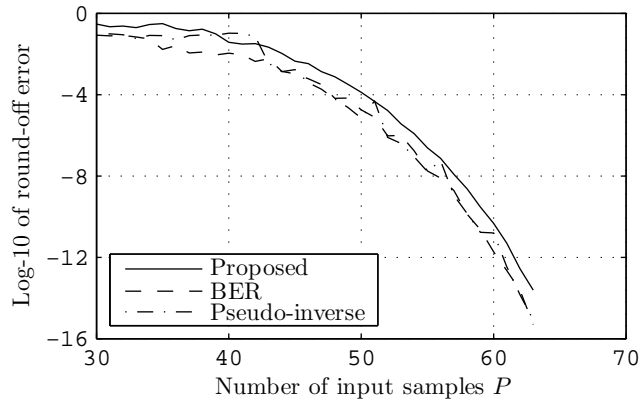


Fig. 3. Maximum round-off error among all interpolated samples and trials versus number of input samples P , for the proposed, BER and pseudo-inverse methods.

difference between the performances of the three methods, with the BER technique having a slightly better performance.

B. Computational burden

In this section, we evaluate the computational burden of the proposed method relative to the pseudo-inverse and BER techniques, using the results in Sec. V.

1) *Complexity versus grid size N* : Fig. 4 shows the flop counts for the three methods versus the grid size, assuming $a = N/P = 8$. There are two variants of the proposed method in this figure. In variant “Prop. A”, $\beta(n)$ in (23) is computed using the FFT (1), while in variant “Prop. B” (23) is evaluated

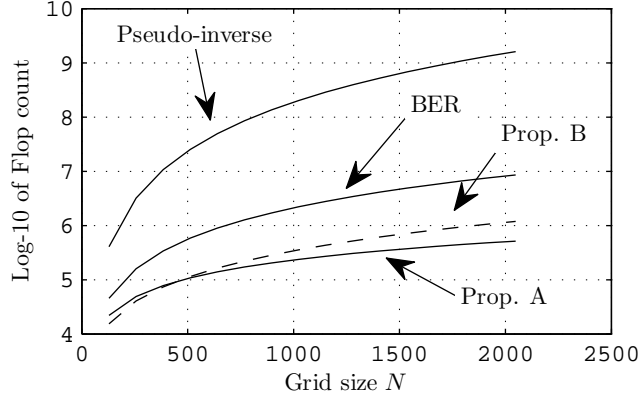


Fig. 4. Complexity versus grid size N for two variants of the proposed method and the BER and pseudo-inverse techniques. Variant “Prop. A” computes $\beta(n)$ in (23) using the FFT, while variant “Prop. B” performs this last computation by directly evaluating the convolution in (23).

directly. The proposed methods shows a clear improvement compared to the BER technique. For large N , “Prop. A” is roughly a factor 16 less complex than the BER technique. Also, note that “Prop. B” improves on “Prop. A” for small N . This is due to the fact that the cyclic convolution in (23) can be directly evaluated without any multiplications. Finally, the pseudo-inverse is by far the most expensive technique due to the computation of one SVD.

2) *Complexity versus N/P ratio:* Fig. 5 shows the ratio

$$\frac{\text{BER tech. flop count}}{\text{“Prop. A” flop count}},$$

versus the factor $a = N/P$ for $N = 1024$, where “Prop. A” was described in the previous sub-section. This figure shows that “Prop. A” improves on the BER technique for all a values except for the very small or very large. Actually, the BER technique is more efficient only if $a < 0.0049$ or $a > 0.96$, ($P \leq 5$ or $P \geq 1019$). The maximum improvement is factor 20 roughly.

3) *Complexity compared with the zero-padding FFT algorithm :* If N/P is an integer and J is a regular grid with spacing N/P , then the missing samples problem can be solved using the zero-padding FFT (ZP-FFT) algorithm, [24, Sec. 3.11]. Fig. 6 shows the complexity of this well-known algorithm and that of the method in this paper. The curve “Proposed, no weight comp.” is the count of “Prop. A” but discounting the complexity of computing $\{\phi(n), n \in J\}$ and $\{\phi'(n), n \in J^c\}$, given that the sampling grid is constant. This figure shows that the proposed method is, in rough terms, only factor two more complex than ZP-FFT if the weight factors are available, and factor 4 if not.

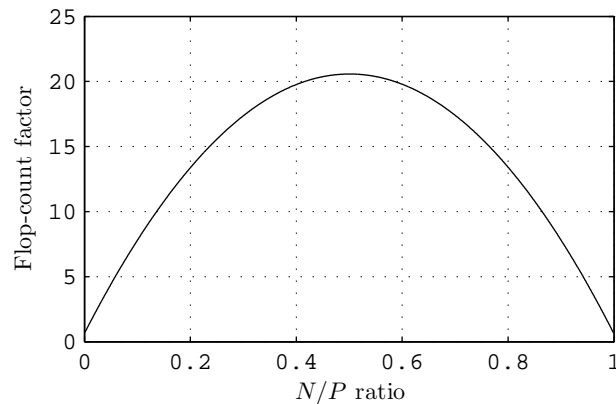


Fig. 5. Improvement factor of the proposed method relative to the BER technique in terms of flop count, (proposed method's count / BER technique's count).

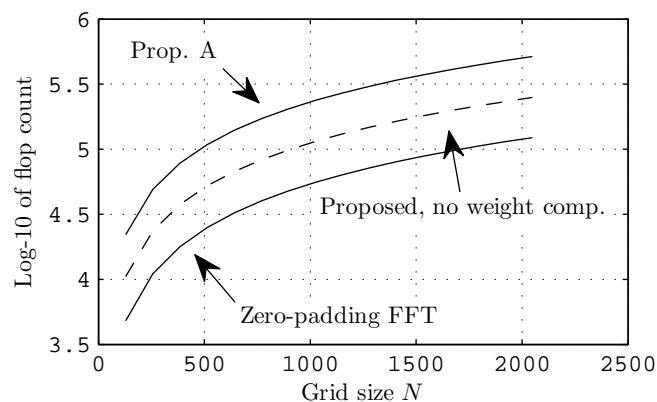


Fig. 6. Complexity versus the zero-padding FFT algorithm.

VII. CONCLUSIONS

We have presented a solution for the missing samples problem based on the FFT. The method has complexity $O(N \log N)$ and consists of four FFTs plus several operations of order $O(N)$. It provides a significant improvement on the burst error recovery (BER) technique, which seems to be the most efficient method in the literature. For typical values of N , the complexity is reduced up to factor 20, relative to this last technique. The method has been assessed in terms of numerical stability and computational burden numerically.

APPENDIX A

PROOF OF (32)

In order to prove (32), write the summation as the logarithm of a polynomial in $z = e^{j2\pi t/N}$:

$$\sum_{n=0}^{N-1} \alpha(n) = \log \left(\prod_{n=1}^{N-1} (z - e^{-j2\pi n/N}) \right) \Big|_{z=1}. \quad (40)$$

Note that $z^N - 1$ is the monic polynomial with roots $e^{j2\pi n/N}$, $n = 0, 1, \dots, N-1$, and these roots also appear in (40), except for the root at $z = 1$. So we have that the polynomial in (40) is actually $(z^N - 1)/(z - 1)$ and

$$\begin{aligned} \sum_{n=0}^{N-1} \alpha(n) &= \log \left(\frac{z^N - 1}{z - 1} \right) \Big|_{z=1} \\ &= \log \left(\sum_{n=0}^{N-1} z^n \right) \Big|_{z=1} = \log(N). \end{aligned}$$

REFERENCES

- [1] M. Ghandi, Mahdi Jahani Yekta, and F. Marvasti, "Some nonlinear/adaptive methods for fast recovery of the missing samples of signals," *Signal Processing*, vol. 88, no. 3, pp. 624 – 638, 2008.
- [2] M.K. Ozdemir and H. Arslan, "Channel estimation for wireless OFDM systems," *IEEE Communications Surveys Tutorials*, vol. 9, no. 2, pp. 18–48, 2007.
- [3] P. Fertl and G. Matz, "Channel estimation in wireless OFDM systems with irregular pilot distribution," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3180–3194, June 2010.
- [4] M. Vaezi and F. Labeau, "Distributed source-channel coding based on real-field BCH codes," *IEEE Transactions on Signal Processing*, vol. 62, no. 5, pp. 1171–1184, March 2014.
- [5] Paulo J. S. G. Ferreira and José M. N. Vieira, "Stable DFT codes and frames," *IEEE Signal Processing Letters*, vol. 10, no. 2, pp. 50–53, 2003.
- [6] G. Rath and C. Guillemot, "Subspace-based error and erasure correction with DFT codes for wireless channels," *IEEE Transactions on Signal Processing*, vol. 52, no. 11, pp. 3241–3252, 2004.
- [7] A.K.M. Pillai and H. Johansson, "A sub-band based reconstructor for M-channel time-interleaved ADCs with missing samples," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 4126–4130.
- [8] K. M. Tsui and S. C. Chan, "A novel iterative structure for online calibration of M-channel time-interleaved ADCs," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 2, pp. 312–325, 2014.
- [9] T. E. Tuncer, "Block-based methods for the reconstruction of finite-length signals from nonuniform samples," *IEEE Transactions on Signal Processing*, vol. 55, no. 2, pp. 530–541, Feb 2007.
- [10] F. Marvasti, Ed., *Nonuniform sampling, theory and practice*, Kluwer academic/Plenum Publishers, 2001.
- [11] A. Papoulis, "A new algorithm in spectral analysis and band-limited extrapolation," *IEEE Transactions on Circuits and Systems*, vol. 22, no. 9, pp. 735–742, Sep 1975.
- [12] P.J.S.G. Ferreira, "Interpolation and the discrete Papoulis-Gerschberg algorithm," *IEEE Transactions on Signal Processing*, vol. 42, no. 10, pp. 2596–2606, Oct 1994.

- [13] F. Marvasti, M. Hasan, M. Echhart, and S. Talebi, "Efficient algorithms for burst error recovery using FFT and other transform kernels," *IEEE Transactions on Signal Processing*, vol. 47, no. 4, pp. 1065–1075, Apr 1999.
- [14] Y. C. Eldar and A. V. Oppenheim, "Filterbank reconstruction of bandlimited signals from nonuniform and generalized samples," *IEEE Transactions on Signal Processing*, vol. 48, pp. 2864–2875, 2000.
- [15] J. Selva, "Functionally weighted Lagrange interpolation of band-limited signals from nonuniform samples," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 168–181, Jan 2009.
- [16] J. Selva, "Design of barycentric interpolators for uniform and nonuniform sampling grids," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1618–1627, Mar 2010.
- [17] D.L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.
- [18] Tang G., B.N. Bhaskar, P. Shah, and B. Recht, "Compressed sensing off the grid," *IEEE Transactions on Information Theory*, vol. 59, no. 11, pp. 7465–7490, Nov 2013.
- [19] M. F. Duarte and R. G. Baraniuk, "Spectral compressive sensing," *Applied and Computational Harmonic Analysis*, vol. 35, no. 1, pp. 111–129, 2013.
- [20] J. Laska, S. Kirolos, Y. Massoud, R. Baraniuk, A. Gilbert, M. Iwen, and M. Strauss, "Random sampling for analog-to-information conversion of wideband signals," in *Design, Applications, Integration and Software, 2006 IEEE Dallas/CAS Workshop on*, Oct 2006, pp. 119–122.
- [21] Alan V. Oppenheim and Ronald W. Schaffer, *Discrete-time signal processing*, Prentice Hall, 1989.
- [22] M. Frigo and S.G. Johnson, "The design and implementation of fftw3," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 216–231, Feb 2005.
- [23] Gene H. Golub and Charles F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, fourth edition, 2013.
- [24] R. G. Lyons, *Understanding Digital Signal Processing*, Prentice Hall, 2001.



J. Selva received the M.Sc. degree in Communications Engineering from the Polytechnic University of Valencia (UPV), Valencia, Spain, in 1994, the M.Sc. degree in Mathematics from the UNED University, Madrid, Spain, in 2000, and the Ph.D. degree in Communications Engineering from the Polytechnic University of Catalonia (UPC), Barcelona, Spain, in 2004. During 1995, he was a grant holder at the Polytechnic University of Valencia. From 1996 to the end of 1998, he was a trainee of the Spanish Ministry of Education at the European Space Agency (ESA) sites in Frascati, Italy, and Noordwijk, The Netherlands. From 1998 to 2004, he was researcher with the German Aerospace Centre (DLR), Oberpfaffenhofen, Bayern, Germany. From 2004 to 2011, he was "Juan de la Cierva" and "Ramon y Cajal" Research Fellow at the University of Alicante, Spain. From 2012 to date he has been associate professor in this last university. His main research interests include interpolation for signal processing, sampling theory, navigation systems, synchronization circuits, array processing, and estimation theory.