

Article

# A Modulated Wideband Converter Model Based on Linear Algebra and Its Application to Fast Calibration

Gilles Burel , Anthony Fiche  and Roland Gautier 

Université de Bretagne Occidentale, Lab-STICC, CNRS, UMR 6285, 6 Avenue Le Gorgeu, 29200 Brest, France

\* Correspondence: gilles.burel@univ-brest.fr

**Abstract:** In the context of cognitive radio, smart cities and Internet-of-Things, the need for advanced radio spectrum monitoring becomes crucial. However, surveillance of a wide frequency band without using extremely expensive high sampling rate devices is a challenging task. The recent development of compressed sampling approaches offers a promising solution to these problems. In this context, the Modulated Wideband Converter (MWC), a blind sub-Nyquist sampling system, is probably the most realistic approach and was successfully validated in real-world conditions. The MWC can be realized with existing analog components, and there exist calibration methods that are able to integrate the imperfections of the mixers, filters and ADCs, hence allowing its use in the real world. The MWC underlying model is based on signal processing concepts such as filtering, modulation, Fourier series decomposition, oversampling and undersampling, spectrum aliasing, and so on, as well as in-flow data processing. In this paper, we develop an MWC model that is entirely based on linear algebra, matrix theory and block processing. We show that this approach has many interests: straightforward translation of mathematical equations into simple and efficient software programming, suppression of some constraints of the initial model, and providing a basis for the development of an extremely fast system calibration method. With a typical MWC acquisition device, we obtained a speed-up of the calibration computation time by a factor greater than 20 compared with a previous implementation.



**Citation:** Burel, G.; Fiche, A.; Gautier, R. A Modulated Wideband Converter Model Based on Linear Algebra and Its Application to Fast Calibration. *Sensors* **2022**, *22*, 7381. <https://doi.org/10.3390/s22197381>

Academic Editors: Yuh-Shyan Chen, Ilsun You and Shih-Lin Wu

Received: 28 July 2022  
Accepted: 22 September 2022  
Published: 28 September 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** compressed sampling; hardware calibration; spectrum monitoring; linear algebra; matrix theory; modulated wideband converter

## 1. Introduction

Digital wireless radio signals are often composed of a small number of narrow-band transmissions spread across a wide spectrum range. For example, the Internet-of-Things (IoT) communications have recently emerged in contexts such as smart cities. Cognitive radios are able to manage the spectrum dynamically but require advanced sensing techniques for spectrum monitoring.

Basically, spectrum monitoring is based on the Shannon–Nyquist sampling theorem [1,2]. This theorem states that the signal must be sampled at a rate greater than its Nyquist frequency, which is twice its frequency band. However, when we have to monitor a large frequency band, this requirement can exceed the capabilities of existing Analog to Digital Converters (ADC) or require very expensive components. Furthermore, sampling at a very high rate may require huge storage capacities to store the digital samples.

Recently, new approaches have been proposed allowing sampling at sub-Nyquist rates. These approaches, known as compressed sensing, or compressive sampling [3], have emerged as a promising framework for signal acquisition in difficult applications, such as monitoring a wideband spectrum [4]. The basic idea of compressed sampling is to take advantage of the fact that a signal that has a sparse representation on a given basis can theoretically be recovered from a small set of linear measurements [5,6]. The price to pay is the need to develop sophisticated signal processing algorithms to reconstruct the signal from this small set of measurements, these algorithms being much more complex than the usual demodulators.

A great deal of the theoretical aspects of compressed sampling has been addressed in the literature. For example, many studies have been proposed in relation to the design of the measurement scheme as in [7,8]. However, few studies have considered the practical limitations of compressed acquisition. In fact, designing measurement schemes and applying them to practical acquisition systems remains a significant challenge.

In this context, the Modulated Wideband Converter (MWC) has been proposed as an efficient system for real-world compressed sampling [9]. The MWC does sub-Nyquist sampling without prior information about the spectral support of the transmitters present in the monitored wideband. It can be realized with existing devices [10] and has been successfully tested on real-world problems, including surveillance of a wideband spectrum [11].

A few real-world compressed sensing acquisition systems have already been proposed [11–18]. An analog circuit board with discrete components was designed by us to prototype compressed sensing based on the MWC principle [19].

A necessary step when using MWC in real-world conditions is the calibration of the acquisition system. Indeed, analog components are never ideal, especially when fed with wideband signals. Then, using a purely theoretical model leads to extremely poor performances of signal reconstruction. An efficient calibration method, which is considered a reference, has been proposed in [13]. It consists of estimating the sensing matrix column after column by injecting sine waves at a specific frequency and recording the corresponding output signals. The procedure is repeated by changing the input frequency until all columns are estimated. Some researchers have exploited this work to calibrate their systems or to propose variants of the calibration algorithms [18,20–23]. While this procedure is efficient, it can be time-consuming because the number of columns to estimate is usually at least a few dozen. That is why a few authors [24,25], including us [19], have recently proposed alternative calibration algorithms requiring only one input signal.

The MWC theoretical background is signal processing theory (filters, modulation, Fourier series, sampling theory, spectrum aliasing, etc.). Most signal processing theoretical tools are asymptotic. However, when signals are processed in real-world conditions, they are always finite; thus, block processing and purely matrix-based algorithms may be more natural and efficient.

Moreover, a quick look at the literature shows that most people use Matrix-based programming tools, such as Octave or Matlab, for signal processing in this context, but without really exploiting the power of these tools. To take full advantage of the power of Matrix-oriented software, it would then be preferable to process data by blocks instead of in-flow. It is, therefore, interesting to view the whole MWC acquisition and reconstruction in terms of block processing. The most natural framework to achieve this objective is matrix theory and linear algebra.

In this paper, we elaborate on an MWC model using linear algebra only (without any signal processing theory). While this approach will probably appear less intuitive than the approach based on signal processing, because most people are less familiar with linear algebra than with signal processing, it has strong advantages:

- Once the model is established, programming it becomes extremely simple, straightforward and efficient.
- Furthermore, computational complexity is significantly reduced.
- The border effects are implicitly taken into account in the model. Indeed, using a signal processing model, people have to deal with the fact that the signals processed in the real world are not infinite, while when using a linear algebra model, the finite nature of the data is implicitly taken into account and the mathematical equations are exact and not approximate.
- In the original version of the MWC, the number of physical branches is increased by a factor  $q$ , which must be an odd integer due to signal processing considerations. An interesting aspect of the linear algebra model is that it allows even integers for  $q$ .

The main contributions of this paper are:

- The development of a pure linear-algebra model of the MWC. Despite the fact that establishing this model is rather hard because it requires non-trivial matrix manipulations, once established, it is extremely simple and allows programming MWC-related software, such as calibration, in a very fast, compact and efficient way.
- Its application to the development of a very fast calibration method. With typical choices of parameters, the calibration is more than 20 times faster than our previous method (this previous method being itself very fast compared to a reference method because it required only one calibration signal instead of dozens of sinusoidal signals in the reference method).

The remainder of this paper is organized as follows: Section 2 provides the main mathematical tools used in the paper. Then, Section 3 presents an overview of our hardware acquisition board and the MWC principle. Section 4 establishes a system model based on linear algebra, and an equivalent model, useful for signal reconstruction and system calibration, is then derived in Section 5. In Section 6, we show how this model allows us to considerably improve a calibration method that we proposed previously, leading to speeding up the process by a factor greater than 20. Then, some experimental results are shown in Section 7.

## 2. Mathematical Background

### 2.1. Notations

Unless otherwise stated, lowercase symbols denote row vectors (e.g.,  $x, p$ ), uppercase symbols denote matrices (e.g.,  $C, Z$ ),  $\bar{x}$  stands for the DFT (Discrete Fourier Transform) of  $x$ . The symbols  $N, K, L, a, b$  will be used to denote the size of vectors or matrices.

We will denote  $D_x$  as the square diagonal matrix whose diagonal is vector  $x$ .

The vectorization of a  $K \times L$  matrix  $Q$ , denoted  $\text{vec}(Q)$ , is the  $1 \times KL$  row vector obtained by reading the matrix row after row, from top to bottom:

$$\text{vec}(Q) = ( q_{11} \cdots q_{1L} \quad q_{21} \cdots q_{2L} \quad \cdots \quad q_{K1} \cdots q_{KL} ) \quad (1)$$

$M^*$  stands for the Hermitian transpose of matrix  $M$ .

$I_K$  stands for the  $K \times K$  identity matrix.

The nearest lower or equal integer will be noted  $\lfloor \cdot \rfloor$  and the nearest greater or equal integer  $\lceil \cdot \rceil$ .

### 2.2. Circulant Matrices

Let  $x$  be a  $1 \times N$  row vector. A circulant matrix  $C_x$  is a square matrix whose first row is  $x$  and each next row is a circular shift one element to the right of the preceding row. That is:

$$C_x = \begin{pmatrix} x_0 & x_1 & x_2 & \cdots & x_{N-1} \\ x_{N-1} & x_0 & x_1 & \cdots & x_{N-2} \\ x_{N-2} & x_{N-1} & x_0 & \cdots & x_{N-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1 & x_2 & x_3 & \cdots & x_0 \end{pmatrix} \quad (2)$$

It is convenient to define the cyclic permutation matrix as the  $N \times N$  matrix below:

$$J_N = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & & 1 \\ 1 & 0 & 0 & \cdots & 0 \end{pmatrix} \quad (3)$$

Then,  $C_x$  is a polynomial in  $J_N$ :

$$C_x = \sum_{n=0}^{N-1} x_n J_N^n \quad (4)$$

The effect of multiplication of a matrix  $M$  by  $J_N$  is as follows. The rows of  $MJ_N$  are the rows of  $M$  circularly shifted one element to the right. The columns of  $J_N M$  are the columns of  $M$  circularly shifted one element to the top.

Matrices  $J_N$  and  $C_x$  commute:

$$J_N C_x = C_x J_N \quad (5)$$

because

$$J_N C_x = J_N \left( \sum_{n=0}^{N-1} x_n J_N^n \right) \quad (6)$$

$$= \sum_{n=0}^{N-1} x_n J_N^{n+1} \quad (7)$$

$$= \left( \sum_{n=0}^{N-1} x_n J_N^n \right) J_N \quad (8)$$

$$= C_x J_N \quad (9)$$

### 2.3. Discrete Fourier Transform (DFT)

Let us note  $\omega$  the  $N^{\text{th}}$  square root of unity below:

$$\omega = \exp\left(-i\frac{2\pi}{N}\right) \quad (10)$$

The DFT matrix  $F_N$  is an  $N \times N$  square symmetric matrix whose element at row  $l$  column  $k$  is  $\omega^{lk}$  (assuming row 0 is the first row, and column 0 the first column):

$$F_N = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \dots & \omega^{(N-1)^2} \end{pmatrix} \quad (11)$$

The inverse DFT matrix is

$$F_N^{-1} = \frac{1}{N} F_N^* \quad (12)$$

The DFT of a vector  $x$  is

$$\bar{x} = x F_N \quad (13)$$

and the inverse DFT (IDFT) is given by  $\bar{x} F_N^{-1}$ .

A circulant matrix is diagonalized by the DFT matrix. That is

$$C_x = F_N D_{\bar{x}} F_N^{-1} \quad (14)$$

It follows that the elements of  $\bar{x}$  are the eigenvalues of  $C_x$  and the columns of  $F_N^{-1}$  are the eigenvectors.

We also have

$$D_{\bar{x}} = F_N^{-1} C_x F_N \quad (15)$$

and

$$\frac{1}{N}C_{\bar{x}} = F_N^{-1}D_x F_N. \quad (16)$$

#### 2.4. Kronecker Product

The Kronecker product is a bilinear matrix operation, denoted by  $\otimes$ . If  $A$  is a  $K \times L$  matrix and  $B$  is a  $M \times N$  matrix, it produces the  $KM \times LN$  block matrix  $C$  below:

$$C = A \otimes B = \begin{pmatrix} a_{11}B & \cdots & a_{1L}B \\ \vdots & \ddots & \vdots \\ a_{K1}B & \cdots & a_{KL}B \end{pmatrix} \quad (17)$$

A useful property about the inverse is:

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1} \quad (18)$$

Assuming the sizes are such that one can form the matrix products  $AC$  and  $BD$ , an interesting property, known as the mixed-product property, is:

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD) \quad (19)$$

The product is not commutative, but there exist permutation matrices (shuffle matrices) such that if  $A$  is an  $a \times a$  square matrix and  $B$  a  $b \times b$  square matrix, then [26]:

$$A \otimes B = P_{a,b}(B \otimes A)P_{b,a} \quad (20)$$

Matrix  $P_{a,b}$  represents the permutation obtained when elements, written row by row in an  $a \times b$  matrix, are read column by column. For instance, set  $a = 2$  and  $b = 3$ , and write the elements 1, 2, 3, 4, 5, 6 row by row in a  $a \times b$  matrix

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \quad (21)$$

Reading column by column, we obtain 1, 4, 2, 5, 3, 6. The permutation matrix is then the matrix such that:

$$(1 \ 4 \ 2 \ 5 \ 3 \ 6) = (1 \ 2 \ 3 \ 4 \ 5 \ 6)P_{2,3} \quad (22)$$

That is:

$$P_{2,3} = \begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & 1 & & & & \\ & & & & 1 & \\ & & & & & 1 \end{pmatrix} \quad (23)$$

If  $N = KL$ , an interesting property with the permutation matrix defined in (3) is

$$J_N^K = J_L \otimes I_K. \quad (24)$$

#### 2.5. General Radix Identity

If  $N$  is a composite number, i.e.,  $N = KL$ , then [26]:

$$F_N = (F_K \otimes I_L)T_{K,L}(I_K \otimes F_L)P_{K,L} \quad (25)$$

where  $T_{K,L}$  is a diagonal matrix (twiddle matrix) and  $P_{K,L}$  a permutation matrix (shuffle matrix defined in Section 2.4).

The twiddle matrix  $T_{K,L}$  is an  $N \times N$  diagonal matrix, the diagonal of which is  $vec(Q_{K,L})$  with  $\omega$  defined in Equation (10) and

$$Q_{K,L} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{L-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(L-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{K-1} & \omega^{2(K-1)} & \cdots & \omega^{(K-1)(L-1)} \end{pmatrix} \quad (26)$$

For instance, with  $K = 2$  and  $L = 3$ , we have

$$Q_{2,3} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & e^{-i\pi/3} & e^{-2i\pi/3} \end{pmatrix} \quad (27)$$

and the diagonal of  $T_{2,3}$  is

$$diag(T_{2,3}) = (1 \ 1 \ 1 \ 1 \ e^{-i\pi/3} \ e^{-2i\pi/3}) \quad (28)$$

Let us note  $\theta_K$  and  $1_K$  as the  $(1 \times K)$  vectors below

$$\theta_K = [1 \ 1 \ \cdots \ 1] \quad (29)$$

$$1_K = [1 \ 0 \ \cdots \ 0] \quad (30)$$

Note that for any  $1 \times L$  vector  $p$ , we have

$$(1_K \otimes p)T_{K,L} = (1_K \otimes p) \quad (31)$$

because only the first  $L$  elements of  $1_K \otimes p$  are non null, and the  $L$  first elements of  $T_{K,L}$  are ones.

Note also that

$$(1_K \otimes p)P_{K,L} = p \otimes 1_K \quad (32)$$

when elements of  $1_K \otimes p$  are written row by row in a  $K \times L$  matrix, the elements of  $p$  go on the first row and the  $K - 1$  next rows are null. Then, when this matrix is read column by column, we obtain elements of  $p$  separated by  $K - 1$  zeroes, that is,  $p \otimes 1_K$

Similarly, it is easy to check that

$$T_{a,b}^{-1}(I_a \otimes 1_b^T) = I_a \otimes 1_b^T \quad (33)$$

and

$$P_{a,b}^{-1}(I_a \otimes \theta_b^T) = \theta_b^T \otimes I_a. \quad (34)$$

## 2.6. Selection Matrix

Let us define the selection matrix  $S_{N,K}^{(r)}$  as the  $N \times K$  matrix below:

$$S_{N,K}^{(r)} = \begin{pmatrix} 0_{r \times K} \\ I_K \\ 0_{(N-K-r) \times K} \end{pmatrix} \quad (35)$$

If  $x = (x_0 \cdots x_{N-1})$  is a  $1 \times N$  vector, then  $y = xS_{N,K}^{(r)}$  is the  $1 \times K$  vector below:

$$y = (x_r \cdots x_{r+K-1}) \quad (36)$$

We consider the indices modulo  $N$ , so  $r$  may be negative.

### 2.7. Moore–Penrose Pseudo-Inverse

Let us consider a rectangular matrix  $Z$  whose size is  $L \times K$  with  $L \leq K$ . The Moore–Penrose pseudo-inverse [27] of  $Z$ , denoted  $Z^+$ , is a  $K \times L$  matrix, which generalizes the concept of inverse and, among other interesting properties, provides a mean to compute a least squares solution to a system of linear equations that lacks an exact solution. The pseudo-inverse is defined and unique for all complex matrices. It is usually computed using the singular value decomposition (SVD).

Let us note the SVD of  $Z$  as [28]:

$$Z = USV^* \quad (37)$$

where  $U$  is a  $L \times L$  unitary matrix (i.e.,  $UU^* = U^*U = I$ ),  $V$  is a  $K \times K$  matrix with orthonormal columns (i.e.,  $V^*V = I$ ) and  $S$  is a diagonal matrix whose diagonal elements are the singular values (non-negative real numbers, ranked by decreasing order). The SVD exists for all complex matrices.

Here we consider a version of the SVD usually called “thin-SVD”, which is a compact version of the more general SVD decomposition (in which matrices  $S$  and  $V$  are larger), because this compact version is sufficient for the purpose of computing the pseudo-inverse. The computational cost of computing the thin-SVD is  $6KL^2 + 20L^3$  flops ([28], p. 254). Note that for complex matrices, it is usual to redefine the floating point operation (flop) in order to count only one flop for the product of two complex numbers, while in reality it requires four real multiplications. Since only ratios between the computational costs of algorithms are of interest, applying this does not change the result.

The pseudo-inverse is given by:

$$Z^+ = VS^+U^* \quad (38)$$

where  $S^+$  is the pseudo-inverse of  $S$ . It is a diagonal matrix in which the diagonal contains the inverses of the singular values of  $S$ , which are above a small tolerance value, and 0 elsewhere.

The cost of the inversion plus the computation of the matrix product is  $2L + KL^2 \simeq KL^2$ .

Overall, the cost of computing the pseudo-inverse is  $7KL^2 + 20L^3$ .

### 3. Acquisition Device and System Parameters

The MWC is a compressed sampling device that samples a signal  $x(t)$  at a sampling frequency  $F_s$  significantly lower than its Nyquist frequency  $F_{nyq}$ . The input signal is assumed sparse in the frequency domain. From the outputs of this acquisition device, one can reconstruct the input signal using a compressed sensing algorithm, such as Orthogonal Matching Pursuit (OMP) [29].

The principle of the MWC is shown in Figure 1:

- The input signal  $x(t)$  is multiplied (using a mixer) by a scrambling signal  $s(t)$ .
- The resulting signal  $v(t)$  goes through a low-pass filter whose impulse response is  $h(t)$ .
- Then, the filter output  $w(t)$  is sampled by an Analog to Digital Converter (ADC), providing the output samples  $y[n]$ .

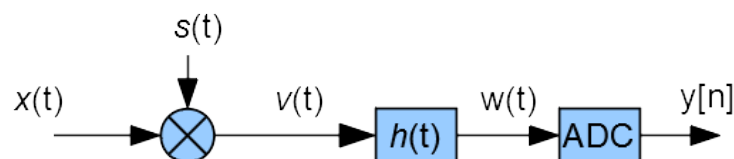


Figure 1. Principle of MWC acquisition (one physical branch).

The scrambler  $s(t)$  is a periodic signal: it is a basic waveform  $p(t)$  repeated  $F_p$  times per second. The analog waveform  $p(t)$  itself is generated at sampling frequency  $F_{nyq}$  from

an  $L$  samples digital sequence, which is usually a pseudo-random sequence. Consequently, we have  $F_p = F_{nyq} / L$ .

The performance of the system can be enhanced by using  $M$  parallel branches with different scrambling signals. However, since generalization to  $M$  branches is trivial, we will restrict the discussions below to one branch.

The digital outputs  $y[n]$  are provided at  $F_s$  samples per second.

In previous practical realizations, in order to reduce aliasing, the ADC output samples go through a digital filter, which provides properly filtered samples at a frequency  $F_{ss}$  lower than  $F_s$ . In the original MWC model,  $F_{ss}$  is an odd multiple of  $F_p$ , that is  $F_{ss} = qF_p$  with  $q$  an odd integer. In this paper, since the linear algebra model allows a less constrained post-processing, this digital filter is not required and  $q$  is not necessarily odd. Indeed, we will see that the linear algebra model also allows even values of  $q$ .

When designing an actual acquisition device, we have to choose some parameters:

- The sampling frequency  $F_{nyq}$  of the scramblers, which will impact the Nyquist frequency of acceptable input signals (i.e., input signal maximum frequency must remain under  $F_{nyq} / 2$ ).
- The sampling frequency  $F_s$  of the ADC, which should be significantly lower than  $F_{nyq}$  (otherwise the system would have no interest compared to direct sampling at  $F_{nyq}$ ). This frequency determines the subsampling factor  $b = F_{nyq} / F_s$ .
- The length  $L$  of the scrambler periodic pattern. This parameter determines the frequency of repetition  $F_p = F_{nyq} / L$  of the scrambling pattern.

Figure 2 illustrates, in a very simplified case ( $L = 5$ ), an example of scrambling signal. It is formed from a length- $L$  binary pseudo-random sequence, which is repeated. In the time domain, the duration of a binary symbol is  $1/F_{nyq}$ , and the period of the signal is  $1/F_p$  (where  $F_{nyq}$  and  $F_p$  are the frequencies defined above).

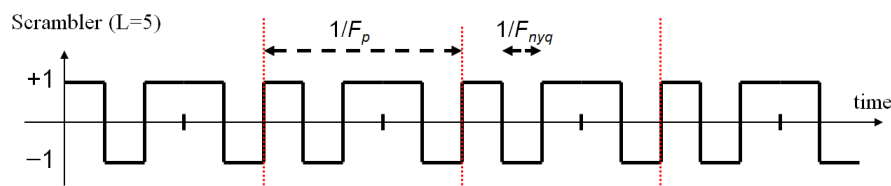


Figure 2. Illustration of the scrambling signal.

Figure 3 illustrates the low-pass filter response. The scrambled signal, which occupies a frequency band of width  $F_{nyq}$ , goes through a low-pass filter. The filter output is sampled at a frequency  $F_s$ , high enough to avoid aliasing.

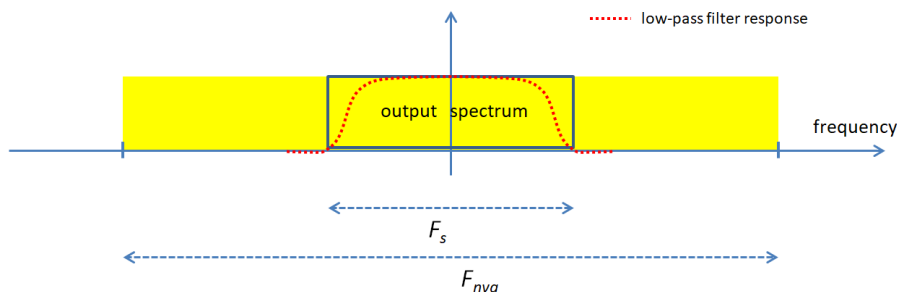


Figure 3. Illustration of the low-pass filter response and output spectrum.

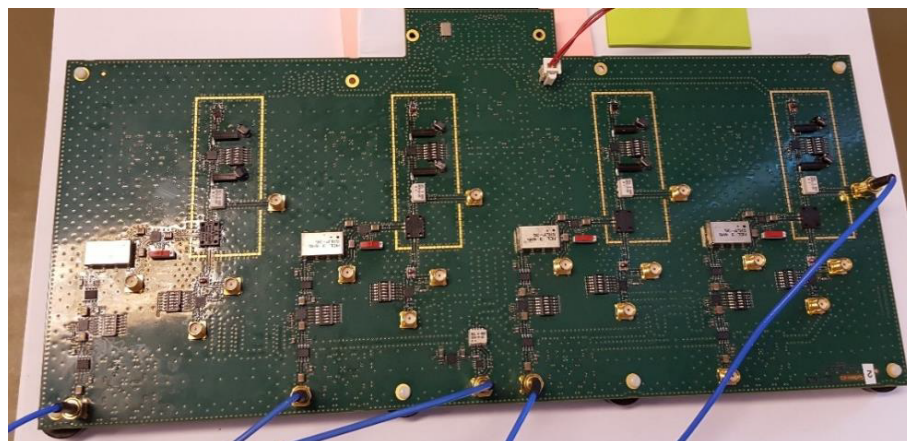
The scrambler and the ADC are controlled by a common central clock to avoid synchronization problems.

Reconstruction of the input signal, and calibration of the system, are based on the information provided by a block of  $a$  output samples. In order to avoid unnecessary mathematical complications, the value of  $a$  is chosen such that it corresponds to an integer number  $K$  of scrambling patterns, then  $a = KL/b$ . This output block then corresponds to



$N = KL$  scrambler samples (and also to  $N$  input samples if the input signal were sampled at  $F_{nyq}$ ). The size of the block determines the frequency resolution  $F_s/a = F_{nyq}/N$ .

For our experiments on real-world data, we designed a 4-channel MWC analog board (Figure 4), which was described in more detail in a previous paper [19]. The scramblers are sampled at  $F_{nyq} = 1$  GHz and their length is  $L = 96$ . Therefore, their repetition frequency is  $F_p = F_{nyq}/L = 10.41667$  MHz. The device is then able to monitor a wideband spectrum of 1 GHz.



**Figure 4.** Our analog acquisition board.

The prototype includes  $M = 4$  physical channels. Each channel features an M1-0008 mixer from MArki©, and an SXLP-36+ low-pass filter from Mini-Circuits©. The filter cut-off frequency is 40 MHz (at -3 dB). The SXLP-36+ filter was chosen because it is quite close to the ideal low-pass filter. Indeed, it has a sharp cut-off, linear phase and flat band (attenuation < 0.5 dB) in the frequency range (0–36) MHz. The ADC sampling frequency is  $F_s = 10F_p = 104.1667$  MHz (at  $F_s/2$  the filter attenuation is more than 30 dB); therefore, the downsampling factor is  $b = 9.6$ . Table 1 sums up the main parameters.

**Table 1.** Parameters of our MWC prototype.

Symbol	Meaning	Value
$M$	Number of physical channels	4
$L$	Length of scramblers	96
$F_{nyq}$	Sampling frequency of scramblers = bandwidth to monitor	1 GHz
$F_s$	Sampling frequency of physical ADC	104.1667 MHz
$b = F_{nyq}/F_s$	Physical subsampling factor	9.6
$F_p = F_{nyq}/L$	Repetition frequency of scramblers	10.41667 MHz

The frequency response of the low-pass filter implemented on our acquisition board is shown in Figure 5 and its phase in Figure 6. Details on filter calibration can be found in one of our previous papers [30].

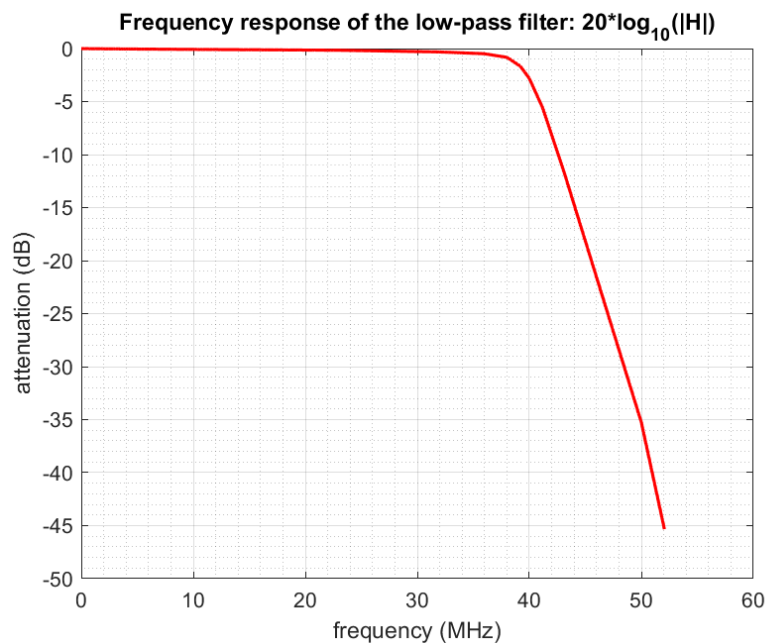


Figure 5. Frequency response of the low-pass filter.

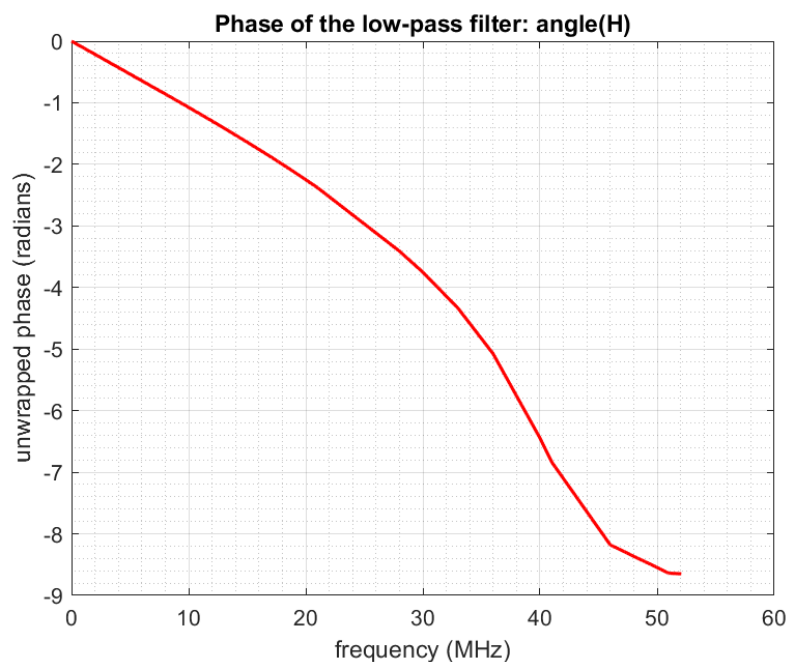


Figure 6. Phase of the low-pass filter.

$F_{nyq}$  being the Nyquist frequency of the input signal, we can consider a digital equivalent model at  $F_{nyq}$  without loss of information. Furthermore, as previously mentioned, since calibration and signal processing are always performed on a limited amount of data in real-world applications, we can consider an input block of  $N$  samples (at  $F_{nyq}$ ).

Modern implementations of the FFT [31] contain a special code to handle splittings not only of size 2 but also of sizes 3 (and sometimes 5 and 7). Therefore, for the efficiency of the FFT, we will preferably choose block sizes whose prime factors belong to  $\{2, 3, 5, 7\}$ . In our experiments, we have taken  $K = 448$ ,  $N = KL = 43,008 = 2^{11} \times 3 \times 7$  and  $a = 4480 = 2^9 \times 5 \times 7$ . The frequency resolution is then  $F_{nyq}/N \simeq 23$  kHz, which is far sufficient unless we would like to detect extreme narrow-band transmitters.

### 4. System Model and Matrix Representation

#### 4.1. System Equations in the Time Domain

Let us note:

- $x$  the vector representing the input signal.
- $s$  the vector representing the scrambling signal.
- $v$  the vector representing the scrambler output.
- $h$  the vector representing the low-pass filter impulse response.
- $w$  the vector representing the low-pass filter output.
- $y$  the  $1 \times a$  vector containing the digital output samples (at  $F_s$ ).

All of these vectors, except for  $y$ , are  $(1 \times N)$  vectors and represent the signals at  $F_{nyq}$  samples per second. In Figure 7, we show the links between these vectors. In the time domain (top of the figure), the signals are represented by vectors. Symbol  $*$  stands for cyclic convolution. These vectors can be transposed in the frequency domain using a multiplication by matrix  $F_N$  or  $F_a$ . A post-processing, described later, is then performed in the frequency domain. The post-processing outputs  $q$  vectors  $\bar{y}_n$  of size  $1 \times K$ .

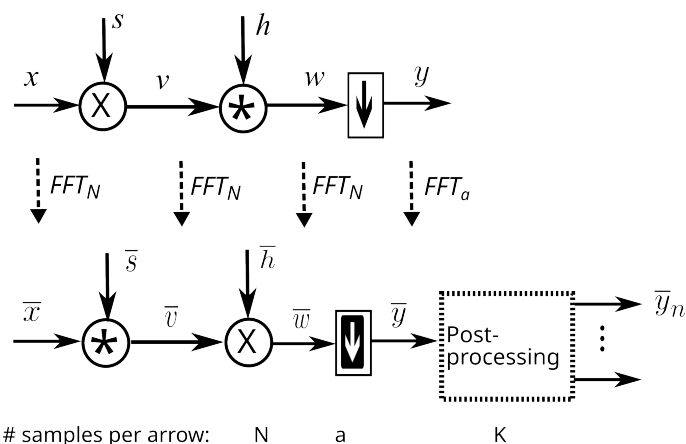


Figure 7. Principle of the system, using vector notations, in the time and frequency domains

In the figure, we have used different symbols for down-sampling because the operation in the time and frequency domains are different. For instance, when  $b$  is an integer, down-sampling in the time domain consists of picking one sample out of  $b$  while its equivalent in the frequency domain is a multiplication of the down-sampling matrix  $\Xi$ , which will be defined later.

Notations used below have already been defined in Section 2. Since the system is linear, in the time domain, we have

$$y = xB \tag{39}$$

where  $B$  is an  $N \times a$  matrix. The structure of  $B$  can be easily computed from the system model (Figure 7):

$$B = D_s C_h (I_a \otimes 1_b^T) \tag{40}$$

Indeed, the scrambler output is given by:

$$v = xD_s \tag{41}$$

The filter output is:

$$w = vC_h \tag{42}$$

For the moment, let us consider that  $b$  is an integer (we will see later that this is not a requirement). In that case, down-sampling consists of picking one sample out of  $b$  in  $w$ . Mathematically, that is:

$$y = w \left( I_a \otimes 1_b^T \right) \quad (43)$$

Otherwise, down-sampling can be modeled using an interpolation matrix. However, we will not detail this because only the equations in the frequency domain will be useful for our purpose. We will see later that in the frequency domain, thanks to the presence of a low pass filter,  $b$  being an integer is not a requirement anymore.

#### 4.2. System Equations in the Frequency Domain

Multiplying Equation (39) by  $F_a$  and inserting the identity  $F_N F_N^{-1}$  where appropriate, we obtain:

$$y F_a = (x F_N) \left( F_N^{-1} B F_a \right) \quad (44)$$

That is

$$\bar{y} = \bar{x} A \quad (45)$$

where  $A$  is an  $N \times a$  matrix.

$$A = F_N^{-1} B F_a \quad (46)$$

The structure of  $A$  can be detailed further. Replacing  $B$  with its expression (Equation (40)) and inserting the identity  $F_N F_N^{-1}$  where it is appropriate, we obtain:

$$A = \left( F_N^{-1} D_s F_N \right) \left( F_N^{-1} C_h F_N \right) \left( F_N^{-1} \left( I_a \otimes 1_b^T \right) F_a \right) \quad (47)$$

Then, using Equations (15) and (16) we obtain:

$$A = \frac{1}{N} C_{\bar{s}} D_{\bar{h}} \Xi \quad (48)$$

As proved in the appendix (see Appendix A.1), the frequency-domain down-sampling matrix  $\Xi$  is:

$$\Xi = \frac{1}{b} \theta_b^T \otimes I_a \quad (49)$$

That is:

$$\Xi = \frac{1}{b} \begin{pmatrix} I_a \\ \vdots \\ I_a \end{pmatrix} \quad (50)$$

where sub-matrix  $I_a$  is repeated  $b$  times. Here we remind that  $h$  is a low-pass filter. Since the ADC sampling frequency is  $F_s$ , we assume that the elements of  $\bar{h}$ , which correspond to frequencies outside the interval  $]-F_s/2, F_s/2[$  are almost null. Since  $\bar{h}$  contains  $N$  elements, the frequency resolution is  $F_{nyq}/N$ , so  $F_s/2$  corresponds to index  $(F_s/2)/(F_{nyq}/N) = N/(2b)$ , that is,  $a/2$ . Let us note

$$c = \lfloor a/2 \rfloor \quad (51)$$

and

$$\delta = a \bmod 2 \quad (52)$$

Therefore, the elements of  $\bar{h}$  are almost null for indices outside the interval  $[-c, c + \delta]$  (the indices are considered modulo  $N$ ). Hence, we can redefine  $\Xi$  as the  $N \times a$  matrix below:

$$\Xi = \frac{1}{b} \begin{pmatrix} I_{c+\delta} & 0 \\ 0 & 0 \\ 0 & I_c \end{pmatrix} \quad (53)$$

without changing the product  $D_{\tilde{h}}\Xi$ . Here, the zeros stand for null sub-matrices. We see that, thanks to the low-pass pass filter which leads to this structure of  $\Xi$ , it is not required any more that  $b$  is an integer (this requirement was only due to the need for an integer number of occurrences of  $I_a$  in Equation (50)).

Finally, let us define the  $(1 \times a)$  vector  $\tilde{h}$ :

$$\tilde{h} = [ \tilde{h}_0 \quad \cdots \quad \tilde{h}_{c+\delta-1} \quad \tilde{h}_{-c} \quad \cdots \quad \tilde{h}_{-1} ] \tag{54}$$

where the indices are modulo  $N$ . We then have:

$$D_{\tilde{h}}\Xi = \Xi D_{\tilde{h}} \tag{55}$$

and the expression of matrix  $A$  becomes:

$$A = \frac{1}{N} C_{\tilde{s}} \Xi D_{\tilde{h}} \tag{56}$$

### 4.3. Unconstrained System Equations in the Time Domain

Now we can go back to the time domain to obtain a matrix  $B$ , which does not require  $b$  being an integer. We have:

$$y = \tilde{y} F_a^{-1} \tag{57}$$

$$= \tilde{x} A F_a^{-1} \tag{58}$$

$$= (\tilde{x} F_N^{-1}) (F_N A F_a^{-1}) \tag{59}$$

$$= x B \tag{60}$$

where

$$B = F_N A F_a^{-1} \tag{61}$$

### 4.4. Fast Simulation of the Acquisition System

A first interest of the linear algebra model is that it makes the design of a fast simulator obvious. Indeed, multiplication by a diagonal matrix  $D$  is efficiently implemented as an element-by-element vectors product, and multiplication by a Fourier matrix  $F$  (or its inverse) is efficiently implemented by Fast Fourier Transform (FFT). On the contrary, multiplications by circulant matrices  $C$  should be avoided because of their computational cost. Then, the method to design a fast simulator is to insert identities  $FF^{-1}$  or  $F^{-1}F$  where it is appropriate in order to suppress the circulant matrices. For instance, we have:

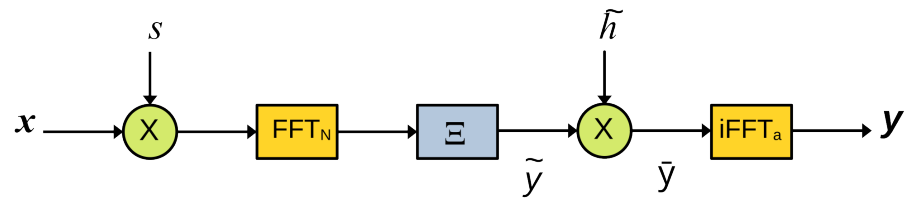
$$y = x B \tag{62}$$

$$= x F_N \frac{1}{N} C_{\tilde{s}} \Xi D_{\tilde{h}} F_a^{-1} \tag{63}$$

$$= x \left( F_N \frac{1}{N} C_{\tilde{s}} F_N^{-1} \right) F_N \Xi D_{\tilde{h}} F_a^{-1} \tag{64}$$

$$= x D_s F_N \Xi D_{\tilde{h}} F_a^{-1} \tag{65}$$

using Equation (16). Here we have only fast operations, as shown in Figure 8.



**Figure 8.** Fast simulation. Dotted arrows are for  $a$  elements, full arrows are for  $N$  elements.

## 5. Equivalent Model and Post-Processing

### 5.1. Equivalent Model

Until now, we have not taken advantage of the periodicity of the scrambler. This opens the way to an equivalent model with interesting properties.

The scrambler is a  $(1 \times N)$  vector  $s$ , which contains  $K$  replica of a basic waveform represented by a  $(1 \times L)$  vector  $p$ . Then, the scrambler can be written:

$$s = \theta_K \otimes p \quad (66)$$

and we have (see proof in Appendix A.2)

$$\bar{s} = K\bar{p} \otimes \mathbf{1}_K \quad (67)$$

It follows that  $\bar{s}$  is sparse (only one element out of  $K$  is non-zero). It will be easier to take benefit of the sparsity of  $\bar{s}$  if we permute  $\bar{x}$  and  $\bar{s}$  in the expression of  $\bar{y}$ :

$$\bar{y} = \frac{1}{N} \bar{x} C_{\bar{s}} \Xi D_{\bar{h}} \quad (68)$$

$$= \frac{1}{N} \bar{s} C_{\bar{x}} \Xi D_{\bar{h}} \quad (69)$$

The proof is trivial: since the multiplication is commutative, we can permute  $x$  and  $s$  (see Figure 7); therefore, we can also permute  $\bar{x}$  and  $\bar{s}$ .

Let us define the  $L \times N$  matrix  $C_{\bar{x}}^{(K)}$  obtained by picking one row out of  $K$  in  $C_{\bar{x}}$ . That is:

$$C_{\bar{x}}^{(K)} = (I_L \otimes \mathbf{1}_K) C_{\bar{x}} \quad (70)$$

More explicitly, that is:

$$C_{\bar{x}}^{(K)} = \begin{pmatrix} \bar{x}_0 & \cdots & \bar{x}_{N-1} \\ \bar{x}_{-K} & \cdots & \bar{x}_{N-K-1} \\ \vdots & & \vdots \\ \bar{x}_{-(L-1)K} & \cdots & \bar{x}_{K-1} \end{pmatrix} \quad (71)$$

where the indices are considered modulo  $N$ .

Let us denote

$$\bar{p} = \frac{1}{L} \bar{p} \quad (72)$$

Using Equation (67), the mixer output becomes:

$$\frac{1}{N} \bar{s} C_{\bar{x}} = \frac{1}{N} (K\bar{p} \otimes \mathbf{1}_K) C_{\bar{x}} \quad (73)$$

$$= \frac{1}{L} ((\bar{p} I_L) \otimes \mathbf{1}_K) C_{\bar{x}} \quad (74)$$

$$= \frac{1}{L} \bar{p} (I_L \otimes \mathbf{1}_K) C_{\bar{x}} \quad (75)$$

$$= \bar{p} C_{\bar{x}}^{(K)} \quad (76)$$

An interesting property of matrix  $C_{\bar{x}}^{(K)}$ , which will be exploited later, is (see proof in Appendix A.3):

$$C_{\bar{x}}^{(K)} J_N^K = J_L C_{\bar{x}}^{(K)} \tag{77}$$

Finally, let us define

$$\tilde{y} = \bar{y} D_{1/\bar{h}} \tag{78}$$

We then have:

$$\tilde{y} = \frac{1}{N} \bar{s} C_{\bar{x}} \Xi \tag{79}$$

$$= \tilde{p} C_{\bar{x}}^{(K)} \Xi \tag{80}$$

### 5.2. Post-Processing

The post-processing extracts frequency blocks of  $K$  samples from  $\tilde{y}$ . Using definition (35), let us note  $S_n$  the  $a \times K$  selection matrix below

$$S_n = b S_{a,K}^{(r+nK)} \tag{81}$$

and  $R_n$  the  $N \times K$  selection matrix below

$$R_n = S_{N,K}^{(r+nK)} \tag{82}$$

Denoting  $\tilde{y}_n$  a  $1 \times K$  vector representing the selected data, we have:

$$\tilde{y}_n = \tilde{y} S_n \tag{83}$$

$\tilde{y}_n$  contains the elements of  $\tilde{y}$  whose indices (modulo  $a$ ) are in the interval  $\Phi = [r + nK, r + nK + K - 1]$ .

The indices are considered modulo  $a$ , so  $r$  may be negative. We will consider that  $\Phi \subset [-c, c + \delta]$ , so

$$\Xi S_n = \Xi b S_{a,K}^{(r+nK)} \tag{84}$$

$$= S_{N,K}^{(r+nK)} \tag{85}$$

$$= R_n \tag{86}$$

We can note that:

$$R_n = J_N^{-nK} R_0 \tag{87}$$

This is a matrix similar to  $R_0$  but with sub-matrix  $I_K$  circularly shifted  $nK$  positions downwards. We can note that we have also:

$$R_{n+1} = J_N^{-K} R_n \tag{88}$$

Eventually, using Equations (76) and (77) we have:

$$\tilde{y}_n = \tilde{p} C_{\bar{x}}^{(K)} \Xi S_n \tag{89}$$

$$= \tilde{p} C_{\bar{x}}^{(K)} R_n \tag{90}$$

$$= \tilde{p} C_{\bar{x}}^{(K)} J_N^{-nK} R_0 \tag{91}$$

$$= (\tilde{p} J_L^{-n}) (C_{\bar{x}}^{(K)} R_0) \tag{92}$$

$$= \tilde{p}_n Z_{\bar{x}} \tag{93}$$

where

$$\tilde{p}_n = \tilde{p} J_L^{-n} \tag{94}$$

and

$$Z_{\bar{x}} = C_{\bar{x}}^{(K)} R_0 \quad (95)$$

More explicitly,  $Z_{\bar{x}}$  is the  $L \times K$  matrix below:

$$Z_{\bar{x}} = \begin{pmatrix} \bar{x}_r & \cdots & \bar{x}_{r+K-1} \\ \bar{x}_{r-K} & \cdots & \bar{x}_{r-1} \\ \vdots & & \vdots \\ \bar{x}_{r-(L-1)K} & \cdots & \bar{x}_{r-(L-2)K-1} \end{pmatrix} \quad (96)$$

were the indices are considered modulo  $N$ . The interesting feature in Equation (93) is that  $Z_{\bar{x}}$  does not depend on  $n$ . Hence, we can take  $q$  different values of  $n$  and write

$$\begin{pmatrix} \vdots \\ \tilde{y}_n \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ \tilde{p}_n \\ \vdots \end{pmatrix} Z_{\bar{x}} \quad (97)$$

assuming we know the filter frequency response (which should be the case because the filter is part of the acquisition system). More compactly, this fundamental equation can be noted:

$$Y = PZ \quad (98)$$

were  $Y$  is the  $(q \times K)$  matrix below:

$$Y = \begin{pmatrix} \vdots \\ \tilde{y}_n \\ \vdots \end{pmatrix} \quad (99)$$

$P$  is the  $(q \times L)$  matrix below:

$$P = \begin{pmatrix} \vdots \\ \tilde{p}_n \\ \vdots \end{pmatrix} \quad (100)$$

and  $Z$  is the  $(L \times K)$  matrix below:

$$Z = Z_{\bar{x}} \quad (101)$$

Therefore, the sizes of the matrices appearing in Equation (98) are  $(q \times K)$ ,  $(q \times L)$ ,  $(L \times K)$ . Then, if the number of non-zero rows in  $Z$  is less than  $q$  the matrix  $Z$  can be reconstructed from this equation using an algorithm such as OMP [29]. Eventually, from  $Z$  we can retrieve  $\bar{x}$  as shown below. Indeed, it is easy to see that  $\bar{x}$  can be rebuilt from  $Z$  with

$$\bar{x} = \text{vec}(A_L Z) J_N^{r+K} \quad (102)$$

where  $A_L$  is the  $K \times K$  anti-diagonal matrix:

$$A_L = \begin{pmatrix} 0 & \cdots & 0 & 1 \\ \vdots & \ddots & 1 & 0 \\ 0 & \ddots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{pmatrix} \quad (103)$$

The effect of the multiplication of a matrix by  $A_L$  on the left is to reverse the order of its rows.



If we have  $M$  channels instead of one in the physical system, the number of rows of  $Y$  becomes  $qM$ ; hence, we can theoretically rebuild the signal if the number of non-zero rows in  $Z$  is less than  $qM$ .

Let us note  $q = 2\rho + \tau$  the Euclidean division of  $q$  by 2. In our experiments, we have set  $r = 0$  for  $q$  even and  $r = -\lfloor K/2 \rfloor$  for  $q$  odd. For  $n$  we take the integers in the interval  $-\rho$  to  $\rho + \tau - 1$ . This choice, while not compulsory, is designed to take into account equally distributed values around frequency 0 in  $\tilde{y}$ , which is a priori the best choice. Indeed, the indices of the samples taken into account go from  $r - \rho K$  to  $r + (\rho + \tau)K - 1$ , that is:

- For  $q$  odd: from  $-\rho K - \lfloor K/2 \rfloor$  to  $\rho K + \lceil K/2 \rceil - 1$
- For  $q$  even: from  $-\rho K$  to  $\rho K - 1$

For this choice, Figure 9 illustrates how the elements of  $\bar{x}$  are arranged into matrix  $Z_{\bar{x}}$  and Figure 10 illustrates how the elements of  $\tilde{y}$  are arranged into matrix  $Y$ .

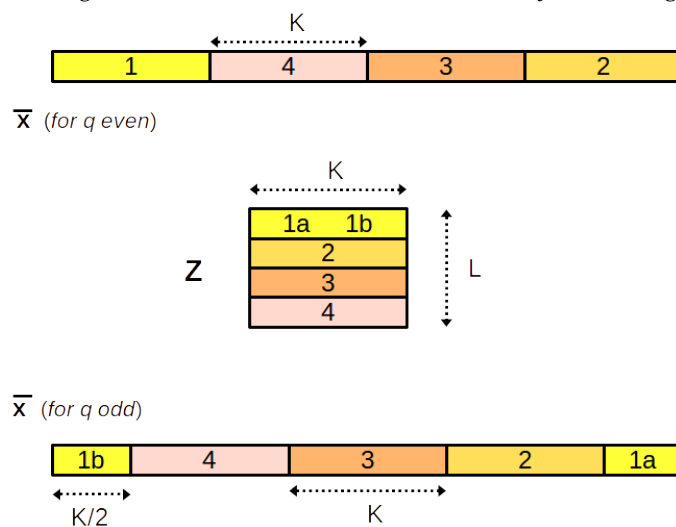


Figure 9. Arrangement of the elements of  $\bar{x}$  into matrix  $Z_{\bar{x}}$ , for  $L = 4$ .

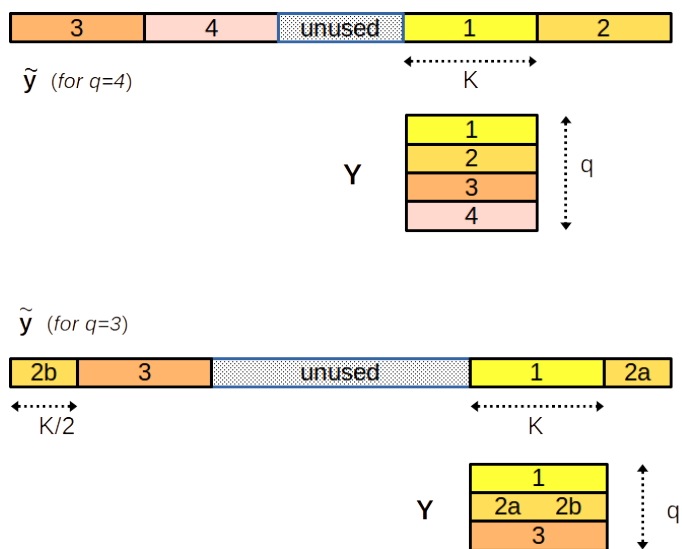


Figure 10. Arrangement of the elements of  $\tilde{y}$  into matrix  $Y$ , for  $q = 4$  and  $q = 3$ .

### 5.3. Application of the Equivalent Model to Reconstruction

The input of the reconstruction algorithm is the vector  $y$  provided by the acquisition device. The output is an estimation of  $\bar{x}$ .

We assume that:

- The frequency response  $\tilde{h}$  of the low-pass filter is known (or has been estimated). Then  $\tilde{h}$  can be precomputed using Equation (54).
- Matrix  $P$  has been precomputed, using Equations (72), (94) and (100), or (better) has been estimated by the calibration process (see Section 6).

The procedure is as follows:

1. Using an  $a$ -points FFT compute  $\tilde{y}$
2. Compensate the filter by computing  $\tilde{y}$  (Equation (78))
3. Extract  $q$  sub-vectors  $\tilde{y}_n$  from  $\tilde{y}$  (Equation (83))
4. Compute matrix  $Y$  (Equation (99))
5. Use a compressed sensing algorithm, such as OMP [29], to estimate matrix  $Z$  from Equation (98)
6. Reconstruct  $\tilde{x}$  from  $Z$  using Equation (102)

As an illustration of how the linear algebra model makes things simple from the programming point of view, this is the Octave program, which builds matrix  $Y$  from  $y$  and then obtains  $x$  from the reconstructed matrix  $Z$ :

```
ytilde = fft(y, [], 2) ./ htilde;
ind = 1+mod(r+[-rho*K:(rho+tau)*K-1], a);
Y = reshape(ytilde(ind), K, q) .';
% Insert here estimation of Z from Y and P (using OMP, for instance)
xbs = reshape(Z(L:-1:1, :)) .', 1, N);
xb = circshift(xbs, [0 r+K]);
x = ifft(xb, [], 2);
```

If there are  $M > 1$  physical channels, the  $q \times K$  matrices  $Y$  corresponding to each channel are stacked vertically, leading to a  $qM \times K$  matrix  $Y$ .

#### 5.4. Interest of $Q$ Even

The linear algebra model allows even values of  $q$ , instead of previous models that required  $q$  to be odd. The main interest is that it puts lower constraints on the design of the acquisition board. If the acquisition board is already available, it may also allow a better use of the MWC output data if the acquisition board is not perfectly optimized.

Let us consider our own acquisition board, which was designed before we established the linear-algebra model. We remind that the ADC sampling frequency is  $F_s \simeq 104.2$  MHz and the scramblers repetition frequency is  $F_p = F_{nyq}/L \simeq 10.42$  MHz. Using  $F_s = F_{nyq}/b$  and  $N = KL = ab$ , it is easy to see that  $F_p = KF_s/a$ . In the frequency domain, the  $a$  output samples correspond to  $F_s$ , then  $qK$  output samples correspond to  $qKF_s/a = qF_p$ .

- With  $q = 7$ , we put into  $Y$  a total of  $qK$  samples corresponding to a frequency half-band  $qF_p/2 = 36.47$  MHz. This choice perfectly fits the frequency response of the low-pass filter, which is an almost perfectly flat and linear phase in [DC-36MHz] (see Figures 5 and 6).
- With  $q = 6$ , we would put into  $Y$  samples corresponding to a frequency half-band  $qF_p/2 = 31.26$  MHz. This corresponds to an even better area of the filter response, but in doing this, we would not use all the available information.
- On the contrary, with  $q = 8$ , we would put it into  $Y$  samples corresponding to a frequency half-band  $qF_p/2 = 41.68$  MHz. This allows us to take more information into account, but we see that we take into account some samples corresponding to a lower quality of the filter response.

This result is not surprising, because our acquisition board was designed and optimized for  $q = 7$ , but for a future design of a new board, the possibility to have  $q$  may even be interesting because it puts fewer constraints on the choice of the commercial filters.

Indeed, applying the proposed method to our analog board did not need any change because odd values of  $q$  are allowed by our method. If a new analog board were to be designed, our method adds an additional degree of freedom because it allows even values

of  $q$  also. This would allow, for example, the designer of the analog board to use a low-pass filter with a cut-off frequency of about 34 MHz (using  $q = 6$ ) instead of 40 MHz (using  $q = 7$ ).

## 6. Application to Fast Calibration

### 6.1. Proposed Approach

The objective of calibration is to estimate the true matrix  $P$ . Indeed, for real-world applications, using the theoretical matrix (Equation (100)) leads to very poor results [13]. In a previous paper [19], we proposed an approach that uses a single wideband signal for calibration, contrary to previous approaches that required successive injections of sinusoids in the system. In that paper, we presented spectrum reconstruction performances and examples of spectrum reconstruction obtained with our calibration method. In the present paper, we will mainly focus on simplifying and speeding up the method thanks to the linear algebra model.

The signal we used for calibration has a spectrum that is totally flat in the  $\left[-F_{nyq}/2, F_{nyq}/2\right]$  frequency band and random phase. Compared to methods based on iterative injections of sinusoidal waves, our new calibration method is time-saving and is more practical in terms of the simplicity of implementation because it requires a single measurement to perform the calibration. The calibration method uses advanced resynchronization preprocessing. Our calibration method offers slightly better spectrum reconstruction performances compared to reference method [13].

If we know matrices  $Y$  and  $Z$ , using Equation (98), we can estimate matrix  $P$  by:

$$\hat{P} = YZ^+ \quad (104)$$

where  $Z^+$  is the Moore–Penrose pseudo-inverse [27] of  $Z$ . Matrix  $Y$  depends on the MWC outputs, and matrix  $Z$  depends on its input signal. The problem in a real-world context is that we cannot reliably synchronize the input of the MWC with the ADC sampling, which provides the output, and even if a costly synchronization device was implemented, there are delays in the analog board itself, which are intractable. Then, the input signal must be designed such that a synchronization can be performed numerically. Otherwise, in Equation (104), we would multiply matrices  $Y$  and  $Z^+$  corresponding to desynchronized data, which would make no sense.

In order to allow an efficient numerical synchronization, we used, for the MWC input, a periodical signal with a flat spectrum and random phase. More precisely, the period of this signal corresponds to the chosen block size, that is  $N/F_{nyq}$ , and one period can be represented by a length- $N$  row vector  $x$ . This vector is generated as follows:

1. A length- $N$  vector  $\bar{x}$  is generated such that, for any of its elements  $\bar{x}(k)$ , we have  $|\bar{x}(k)| = 1$  and  $Arg(\bar{x}(k))$  is random in  $[0, 2\pi[$  under the constraint  $Arg(\bar{x}(-k)) = -Arg(\bar{x}(k))$  (this constraint ensures that  $x$  is real).
2.  $x$  is deduced from  $\bar{x}$  by an inverse FFT:  $x = \bar{x}F_N^{-1}$

Reminding that matrix  $Z$  contains the elements of  $\bar{x}$  (Equation (101)), the constant modulus  $|\bar{x}(k)| = 1$  ensures that no element of  $Z$  is privileged or disadvantaged by the input signal. Furthermore, the random phase ensures that the input signal has good localization properties, which is desired for efficient synchronization. Finally, choosing a periodic signal has a strong advantage: a time shift of a block taken on the input signal is equivalent to a cyclic permutation of vector  $x$ .

On the programming point of view, building  $Z$  from  $x$  is very simple:

```
xb = fft(x, [], 2);
xbs = circshift(xb, [0 -(r+K)]);
Z(L:-1:1, :) = reshape(xbs, K, L) .';
```

In the following, we will note  $x_0 = x$  as the signal pattern and  $x_d$  a cyclic permutation,  $d$  positions to the right, of the pattern. This means that

$$x_d = xJ_N^d \quad (105)$$

The procedure that we propose is as follows:

1. Feed the acquisition device with a periodic signal, which is a repetition of a known pattern  $x_0$
2. Record  $a$  samples at the output of the acquisition device (this is vector  $y$ ), then compute matrix  $Y$  using steps 1 to 4 of the reconstruction procedure.
3. Perform cyclic permutations of the input pattern  $x_0$ . For each vector  $x_d$ , compute matrix  $Z_d = Z_{\bar{x}_d}$  (using Equations (96) and (101)) and then  $\hat{P}$  (using Equation (104)).

Let us note the residue

$$R_d = Y - \hat{P}_d Z_d. \quad (106)$$

The criterion to determine the best cyclic permutation of the input pattern  $x_0$  is the inverse Frobenius norm of  $R_d$  (the Frobenius norm is the square root of the sum of the square modules of the elements of the matrix).

The computational cost per iteration (i.e., per value of  $d$  tested) can be estimated as follows:

- A FFT is required to compute  $\bar{x}_d$ , that is  $LK \log_2(LK)$  flops (because  $N = LK$ ).
- Computation of the pseudo-inverse  $Z_d^+$ :  $7KL^2 + 20L^3$  flops (see Section 2.7).
- Computation of  $\hat{P} = YZ_d^+$  (the sizes of the matrices are  $qM \times K$  and  $K \times L$ ):  $qMKL$  flops.
- Computation of  $\hat{P}Z_d$  requires  $qMLK$  flops.
- Computation of the Frobenius norm requires  $KL$  flops

Globally, since the computation of the Frobenius norm can be neglected compared to the other terms, the algorithm requires about  $L(2qMK + 7LK + 20L^2 + K \log_2(LK))$  flops per iteration.

## 6.2. Fast Update of Matrix Z

While evaluating all possible shifts  $d$ , computation of matrix  $Z_d = Z_{\bar{x}_d}$  requires an  $N$ -points FFT to obtain  $\bar{x}_d$ , which requires approximately  $N \log_2(N)$  multiplication at each iteration. However, we can reduce the complexity just by computing the first matrix and then updating it at each iteration as described below. Let us consider a vector  $x_d$ , which is a cyclic permutation,  $d$  positions to the right of pattern  $x$ . We have:

$$x_d = x_0 J_N^d \quad (107)$$

Then

$$\bar{x}_d = x_0 J_N^d F_N \quad (108)$$

$$= x_0 F_N F_N^{-1} J_N^d F_N \quad (109)$$

$$= \bar{x}_0 D_{\bar{\alpha}_d} \quad (110)$$

where

$$\alpha_d = \left[ \underbrace{0 \cdots 0}_{d} 1 0 \cdots 0 \right] \quad (111)$$

Indeed, since  $J_N^d = C_{\alpha_d}$ , using Equation (15) we have:

$$F_N^{-1} J_N^d F_N = F_N^{-1} C_{\alpha_d} F_N \quad (112)$$

$$= D_{\bar{\alpha}_d} \quad (113)$$

Since  $\bar{\alpha}_d = \alpha_d F_N$ , it is easy to see that  $\bar{\alpha}_d$  is the  $(d+1)^{th}$  row of  $F_N$ , that is (see Equations (10) and (11)):

$$\bar{\alpha}_d = \left[ 1 \quad \omega^d \quad \omega^{2d} \quad \dots \quad \omega^{(N-1)d} \right] \quad (114)$$

Then, we have:

$$Z_d = Z_{\bar{\alpha}_d} \bullet \bar{x}_0 \quad (115)$$

$$= Z_{\bar{\alpha}_d} \bullet Z_0 \quad (116)$$

where  $\bullet$  stands for element by element multiplication. This equation can also be written

$$Z_d = \omega^{-rd} \Theta_d^* Z_0 \Omega_d \quad (117)$$

with

$$\Theta_d = \text{diag} \left[ 1 \quad \omega^{Kd} \quad \omega^{2Kd} \quad \dots \quad \omega^{(L-1)Kd} \right] \quad (118)$$

$$\Omega_d = \text{diag} \left[ 1 \quad \omega^d \quad \omega^{2d} \quad \dots \quad \omega^{(K-1)d} \right] \quad (119)$$

To see where this formula comes from, we must remind that multiplication by a diagonal matrix on the left (right) multiplies the rows (columns) by the elements of the diagonal. Then, denoting  $\omega_d = \omega^d$  we can see that:

$$\omega_d^{-r} \text{vec}(\Theta_d)^* \text{vec}(\Omega_d) = \omega_d^{-r} \begin{bmatrix} 1 \\ \omega_d^{-K} \\ \dots \\ \omega_d^{-(L-1)K} \end{bmatrix} \left[ 1 \quad \omega_d \quad \dots \quad \omega_d^K \right] \quad (120)$$

$$= \omega_d^{-r} \begin{bmatrix} 1 & \omega_d & \dots & \omega_d^{K-1} \\ \omega_d^{-K} & \omega_d^{-K+1} & \dots & \omega_d^{-1} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_d^{-(L-1)K} & \omega_d^{-(L-1)K+1} & \dots & \omega_d^{-(L-2)K-1} \end{bmatrix} \quad (121)$$

$$= Z_{\bar{\alpha}_d} \quad (122)$$

If we evaluate by step  $g$ , we can use:

$$Z_d = Z_{\bar{\alpha}_g} \bullet Z_{d-g} \quad (123)$$

Matrix  $Z_{\bar{\alpha}_g}$  can be precomputed. Therefore, at each iteration, we need only  $N$  multiplications, which is less complex than computing  $\bar{x}_d$  each time. This update requires only  $N = LK$  multiplications at each iteration, instead of approximately  $N \log_2(N)$ .

If we want to allow sub-sample precision (i.e.,  $g < 1$ ), we just have to note  $\eta = N/2$  and write  $\bar{\alpha}_g$  as follows:

$$\bar{\alpha}_g = \left[ 1 \quad \omega^g \quad \dots \quad \omega^{\eta-1} \quad \omega^{-\eta g} \quad \dots \quad \omega^{-g} \right] \quad (124)$$

This is very interesting because sub-sample precision is then allowed without any additional cost due to oversampling (with this method, no oversampling is required). Matrix  $Z_{\bar{\alpha}_g}$  is computed as follows:

```
alpha = exp(-i*2*pi*g*[0:eta-1 -eta:-1]/N);
alpha = circshift(alpha, [0 -(r+K)]);
Zalpha(L:-1:1,:) = reshape(alpha,K,L).';
```

Then, at each iteration, updating matrix  $Z$  is achieved by:

```
Z = Z .* Zalpha;
```

### 6.3. Fast Update of Matrix $Z^+$

The approach is similar to the pseudo-inverse: we can reduce the complexity just by computing the first pseudo-inverse matrix and then updating it at each iteration, as described below. This update requires only  $N$  multiplications at each iteration.

According to discussions above, denoting  $Z_0 = USV^*$  as the SVD of  $Z_0$ , we have

$$Z_d = \omega^{-rd} \Theta_d^* (USV^*) \Omega_d \quad (125)$$

$$= (\omega^{-rd} \Theta_d^* U) S (\Omega_d^* V)^* \quad (126)$$

This equation directly provides the SVD of  $Z_d$ . It follows that  $Z_d^+$  is

$$Z_d^+ = (\Omega_d^* V) S^+ (\omega^{-rd} \Theta_d^* U)^* \quad (127)$$

$$= \omega^{rd} \Omega_d^* Z_0^+ \Theta_d \quad (128)$$

This update can be realized by element-by-element multiplication :

$$Z_d^+ = Z_{\alpha_d}^* \bullet Z_0^+ \quad (129)$$

If we evaluate by step  $g$  we can update the matrix at each iteration by:

$$Z_d^+ = Z_{\alpha_g}^* \bullet Z_{d-g}^+ \quad (130)$$

where  $Z_{\alpha_g}^*$  can be precomputed.

### 6.4. Fast Synchronization

To obtain a synchronization, we must evaluate the Frobenius norm of  $R_d$  at each iteration.

We can evaluate the computational complexity of this fast algorithm as follows:

- Updating matrix  $Z_d$  and  $Z_d^+$  requires  $2LK$  flops.
- Computing  $Y_d Z_0^+$  requires  $qMKL$  flops.
- Multiplying  $Y_d Z_0^+$  by  $Z_0$  requires  $qMKL$  flops.
- Computing the Frobenius norm requires  $KL$  flops.

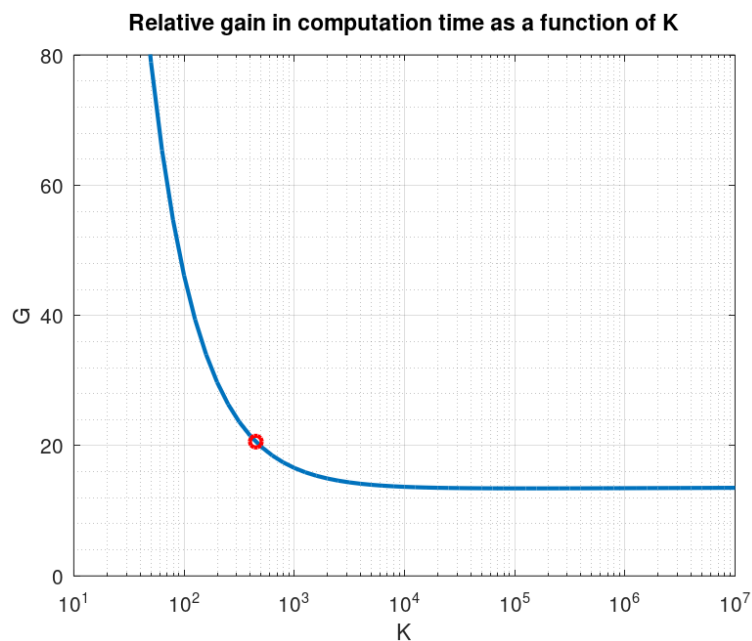
Globally, the algorithm requires  $2qMKL + 3KL \simeq 2qMKL$  flops per iteration, which is to be compared to  $L(2qMK + 7LK + 20L^2 + K \log_2(LK))$  for the previous version. The gain is, approximately:

$$G \simeq 1 + \frac{7L}{2qM} + \frac{10L^2}{qMK} + \frac{\log_2(L)}{2qM} + \frac{\log_2(K)}{2qM} \quad (131)$$

$$\simeq 21 \quad (132)$$

Computation time on Octave is 54 s for the slow version and 1.6 s for the fast version, which is then 34 times faster.

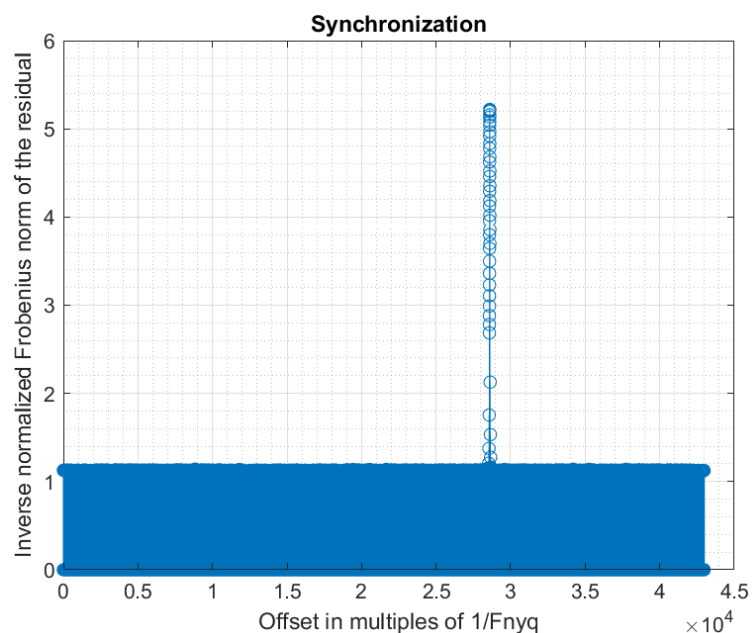
As a function of  $K$ , the gain decreases until  $K = 20 \ln(2)L^2 = 127761$  (which is a huge value, not expected for real-world acquisition devices), where it reaches a minimum of 13.4, then increases slowly, behaving asymptotically as  $\log_2(K)/(2qM)$ , as shown in Figure 11.



**Figure 11.** Gain as a function of  $K$  (the red dot shows the values corresponding to our acquisition board).

### 7. Experimental Results

Using a step  $g > 1$  for the evaluation of the synchronization criterion allows a decrease in the computation time by a factor  $g$ . A good strategy is to obtain a coarse synchronization with a step  $g > 1$ , and then to perform a fine synchronization around the detected peak. The fine synchronization may even be realized at sub-sample precision, if desired. However, a too large initial step must be avoided because it may lead to missing the synchronization peak during the coarse synchronization. In our experiments, we first used a coarse synchronization with step  $g = 16$ , then a fine synchronization with step  $g = 1$  around the coarse synchronization peak. Figure 12 shows the obtained synchronization data. Figures 13 and 14 present are magnifications of the synchronization peak to show more details.



**Figure 12.** Synchronization (overview).

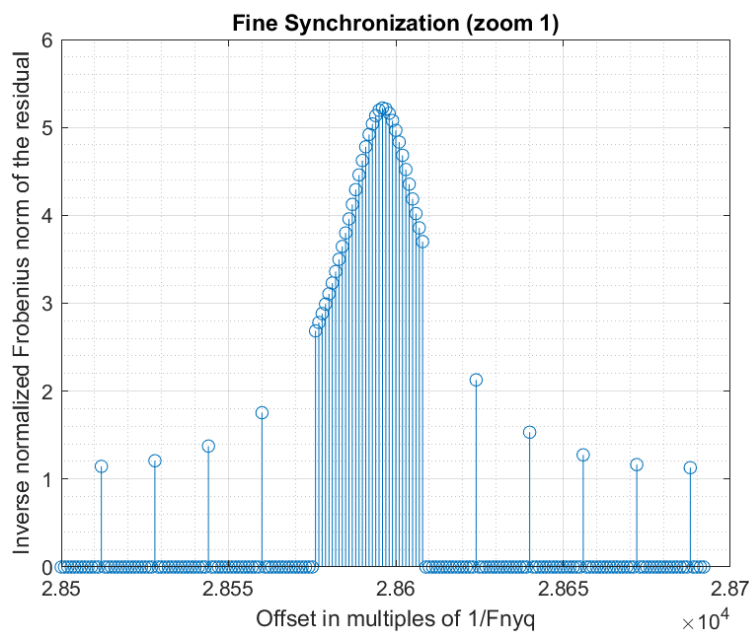


Figure 13. Synchronization (zoom 1).

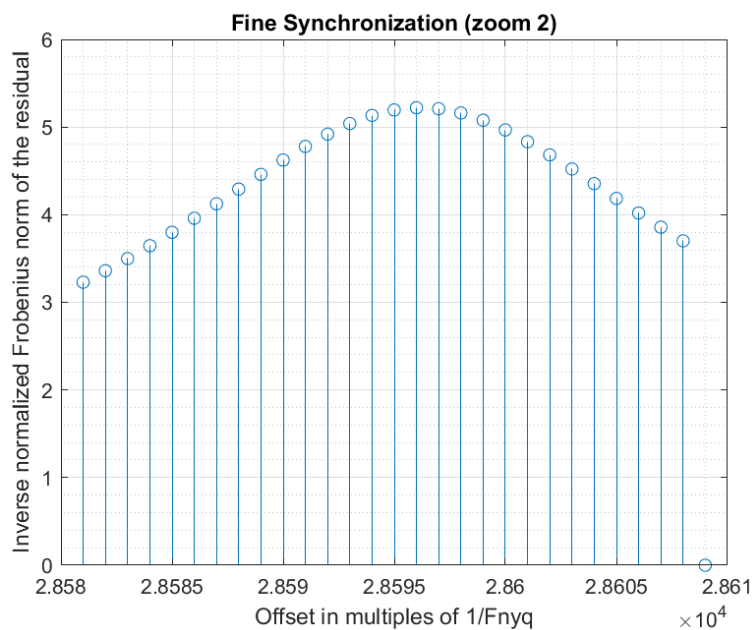
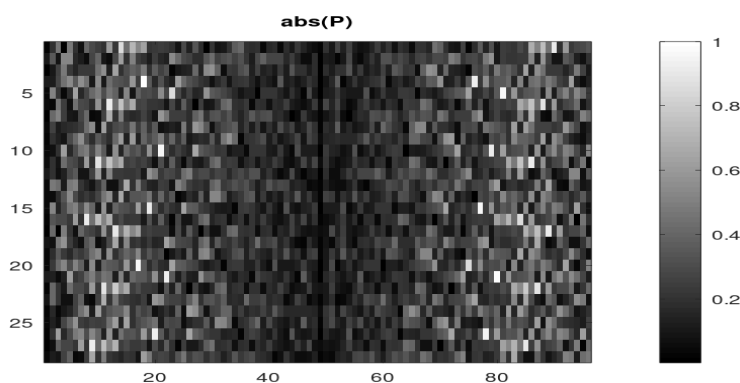


Figure 14. Synchronization (zoom 2).

If the response of the filter is not taken into account in Equation (78) (i.e., assuming an ideal low-pass filter), the synchronization peak is only slightly lower (4.75 instead of 5.22). This is due to the fact that the low-pass filter used in our analog board has good characteristics (almost flat response, and almost linear phase, in the band of interest). The difference would be higher with a lower quality filter. Anyway, it is always better to integrate the filter response in the equations, as we did, because the additional computational cost is negligible.

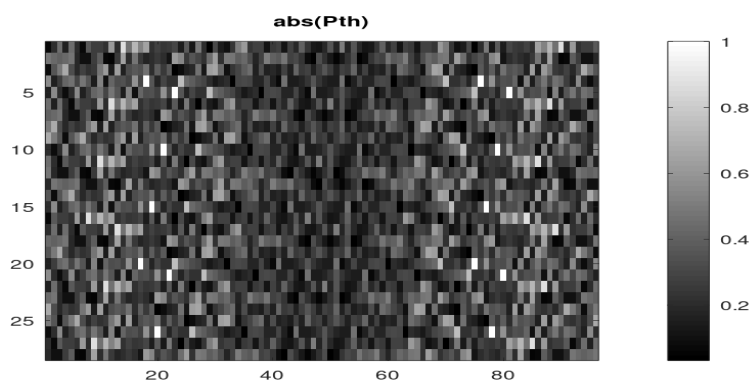
Once the signal is synchronized, the estimated matrix  $\hat{P}$  is obtained using Equation (104) (at no additional cost because this computation was already part of the synchronization process). Figure 15 shows the modulus of the matrix elements. Here, since we have  $M = 4$  physical channels and we have taken  $q = 7$ , the matrix has  $qM = 28$  rows (and  $L = 96$  columns). The first  $q = 7$  rows correspond to the first physical channel, then the next  $q$  rows to the second physical channel, and so on.





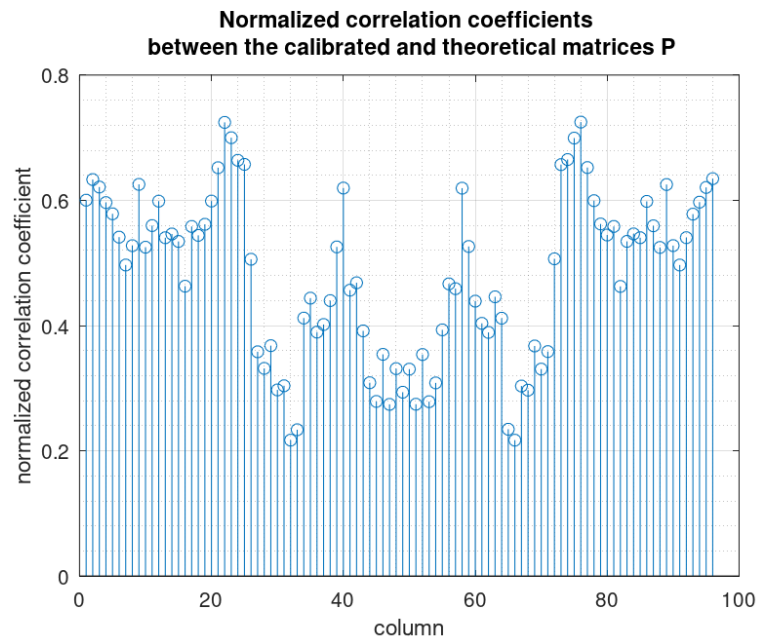
**Figure 15.** Estimated mixture matrix  $P$  (modulus).

The theoretical (ideal) matrix  $P_{th}$  can be computed using Equation (100). Since the analog scrambling sequence is the output of a Digital to Analog Converter (DAC), fed with a digital pseudo-random sequence, it is (ideally) piecewise constant in the time domain. In the spectral domain, this is equivalent to a multiplication by the *sinc* function in which first zero is at  $F_{nyq}$ . We take that into account when computing our theoretical matrix in order to be as close as possible to the real-world matrix. Figure 16 shows the modules of the elements of this matrix.



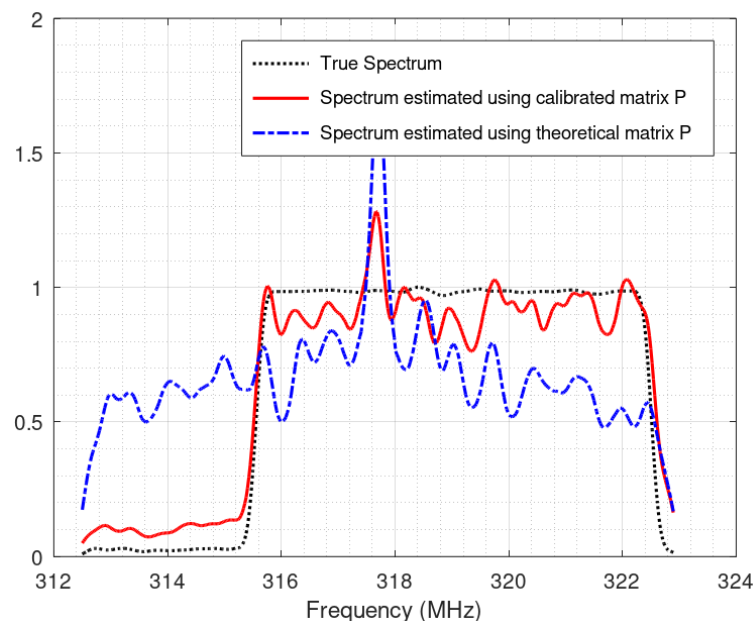
**Figure 16.** Theoretical mixture matrix  $P_{th}$  (modulus).

The estimated (real-world) mixture matrix  $\hat{P}$  may be compared with the theoretical (ideal) mixture matrix  $P_{th}$ . On the basis of the elements modules, we can see that their overall aspects are close despite noticeable differences. In fact, the main differences are in the phases of the elements. If we draw the normalized correlation coefficients between the columns of both matrices (Figure 17), we obtain low values, which confirms significant differences. We remind that a normalized correlation coefficient is the absolute value of the cosine of the angle between two vectors (here the columns of both matrices), then values around 0.5 mean that the angle is about 60 degrees, and thus, the columns are significantly different.



**Figure 17.** Normalized correlation coefficients.

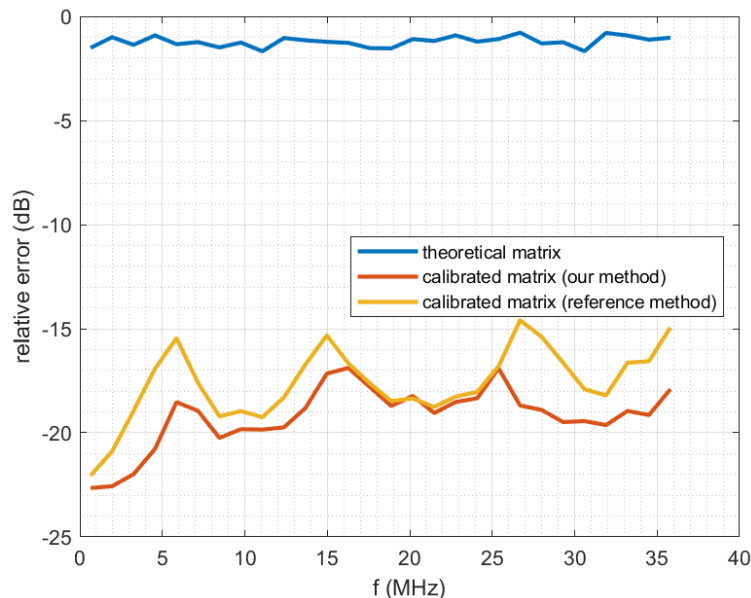
In a previous paper [19], we showed that despite the good quality of our real-world acquisition board, calibration of the system is absolutely required: using the theoretical matrix leads to poor reconstruction performances. Without calibration, the system usually incorrectly detects the active sub-bands, and even when the active sub-bands are correctly identified, the spectrum reconstruction provided by the uncalibrated system is extremely poor, as illustrated in Figure 18.



**Figure 18.** Example of sub-band spectrum reconstruction with and without calibration.

Figure 19 shows the relative error between computed and observed system output, as a function of output frequency. The computation is performed using a formula similar to the criterion used in [32].

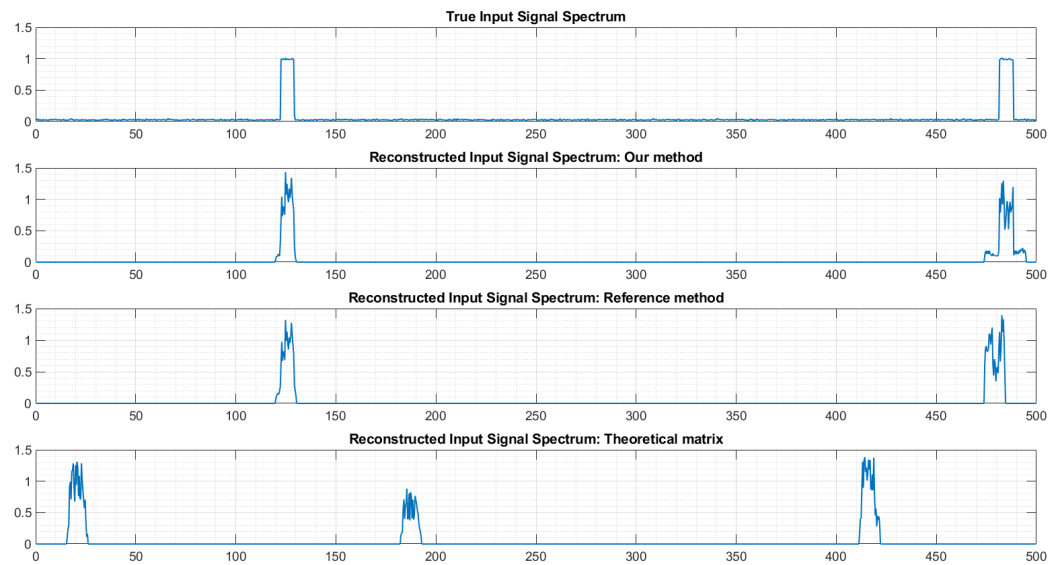
$$\varepsilon(f) = 20 \log_{10} \left( \frac{\|output_{real}(f) - output_{computed}(f)\|}{\|output_{real}(f)\|} \right) \quad (133)$$



**Figure 19.** Relative error between computed and observed system output, as a function of output frequency.

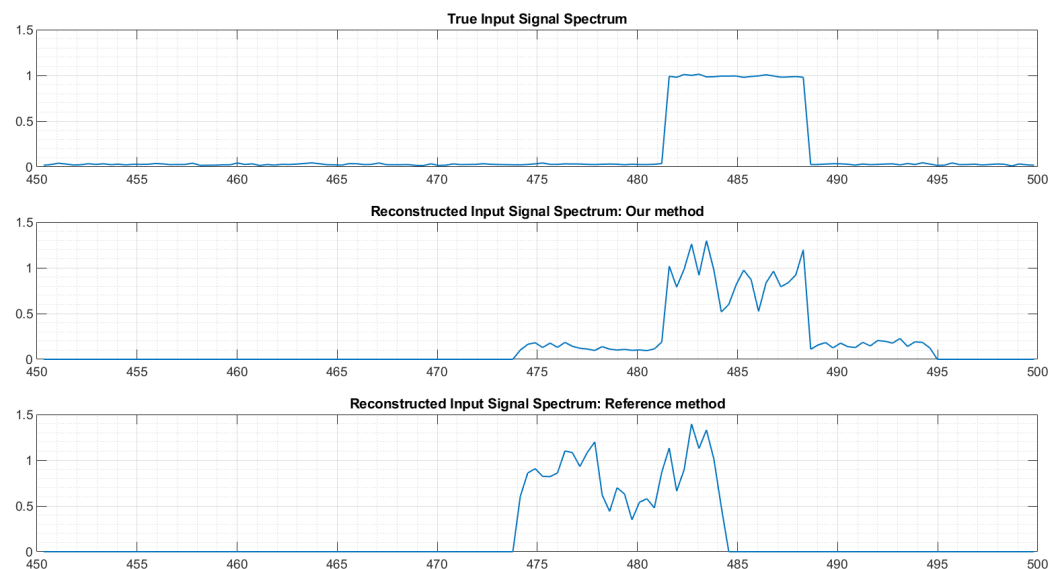
Here the frequency band of the acquisition system output signal has been divided into 28 subbands, and  $output_{real}$  is the real output signal comprised in the subband centered on  $f$ . Similarly,  $output_{computed}(f)$  is the computed output signal comprised in the subband centered on  $f$ , “computed” meaning that it is computed from the input signal using the system model and a given matrix  $P$ . The three curves correspond to using different matrices  $P$  in this computation. It can be seen that our method and the reference method provide a good prediction of the real output (relative errors around  $-18$  dB), while the theoretical uncalibrated matrix provides poor results (large relative errors). We remind that in any case, even for the theoretical matrix, we always use a low-pass filter that is calibrated: the true frequency response of the filter (Figures 5 and 6) is being taken into account in the equations through the diagonal matrix  $D_{\tilde{h}}$ .

To further illustrate the interest of the approach, Figure 20 shows the spectrum of the reconstructed signal using different matrices  $P$ . On top, it is the true spectrum. Here we have two transmitters in the monitored bandwidth. We can see that our method, as well as the reference method [13], correctly identifies the presence of two active transmitters. On the contrary, when using the theoretical matrix, the number of transmitters and their frequency locations are incorrect.



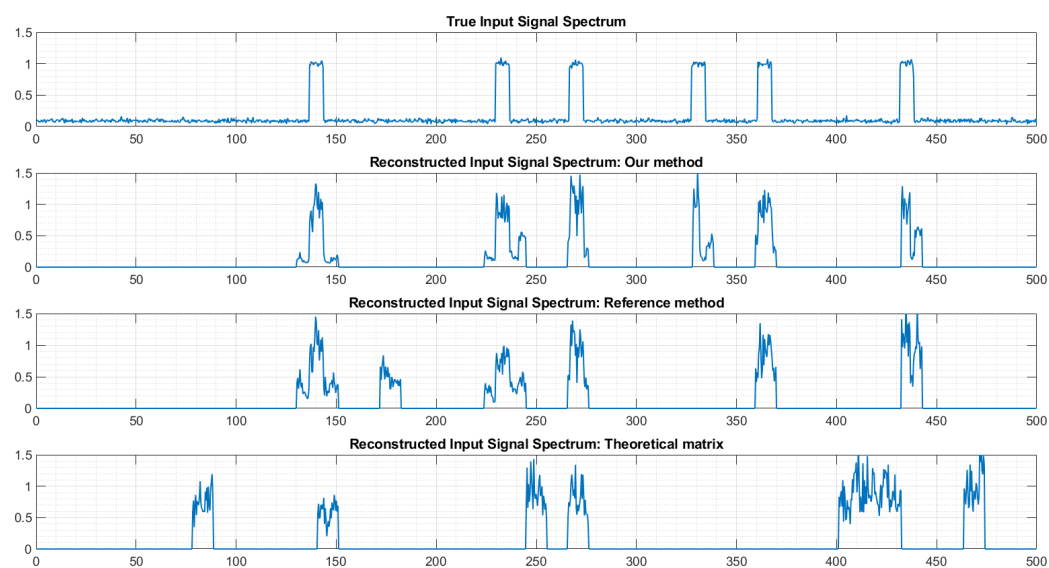
**Figure 20.** Example of spectrum reconstruction with 2 transmitters. The horizontal axes are frequencies, labeled in MHz.

If we zoom in on Figure 20, we can see (Figure 21) that the frequency location of the second transmitter is more precise when using matrix  $P$  provided by our method than when using the matrix provided by the reference method.



**Figure 21.** Example of spectrum reconstruction with 2 transmitters (zoom). The horizontal axes are frequencies, labeled in MHz.

Let us now consider a much more difficult case, where six transmitters are simultaneously active in the monitored bandwidth (Figure 22). Using the matrix  $P$  provided by our approach leads to a correct estimation of the number of transmitters and of their frequency location, despite imperfect shapes of the spectrum reconstruction for some transmitters. The reference method produced quite good results also, but missed one transmitter (around 330 MHz) and detected a non-existing transmitter (around 180 MHz). Finally, as anticipated, using the theoretical matrix  $P$  lead to a reconstruction that is almost totally false.



**Figure 22.** Example of spectrum reconstruction with 6 transmitters. The horizontal axes are frequencies, labeled in MHz.

This shows that calibration is unavoidable. An interest of the extremely fast calibration procedure proposed in this paper is the possibility of performing quick recalibration of the system as soon as the performances appear to decrease. Indeed, many factors, such as temperature, external perturbation, components aging, etc., modify the characteristics of the system, making a recalibration necessary.

## 8. Conclusions

In this paper, we have established an MWC model solely based on linear algebra. It is very convenient as a basis for fast and efficient programming of simulation, reconstruction and calibration algorithms related to MWC. It suppresses a previous restriction on the channels augmentation factor, hence providing more degrees of liberty to the systems designer. It also allowed us to develop an extremely fast implementation of a previously proposed calibration algorithm, leading to a gain of a factor greater than 20 on the computation time. This fast calibration allows quick recalibration of the system as soon as it becomes necessary. Our future work will include more in-depth exploitation of the advantages and interesting properties of this model.

**Author Contributions:** Conceptualization, G.B.; methodology, G.B., A.F., R.G.; software, G.B.; validation, G.B., A.F., R.G.; formal analysis, G.B., A.F., R.G.; investigation, G.B., A.F., R.G.; resources, G.B., A.F., R.G.; data curation, G.B., A.F., R.G.; writing—original draft preparation, G.B.; writing—review and editing, G.B., A.F., R.G.; visualization, G.B., A.F., R.G.; supervision, R.G.; project administration, A.F. and R.G.; funding acquisition, G.B. and R.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research is partly supported by the IBNM (Brest Institute of Computer Science and Mathematics) CyberIoT Chair of Excellence at the University of Brest. The design of our MWC analog board has been partly supported by the RAPID REACC-RF project, whereas the MWC model based on linear algebra was developed after the end of the REACC-RF project.

**Institutional Review Board Statement:** Not applicable

**Informed Consent Statement:** Not applicable

**Data Availability Statement:** Not applicable

**Acknowledgments:** The authors would like to thank the company Syrlinks for the design of the analog board.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## Abbreviations

The following abbreviations are used in this manuscript:

ADC	Analog to Digital Converter
DAC	Digital to Analog Converter
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
flop	floating point operation
MWC	Modulated Wideband Converter
SVD	Singular Value Decomposition

## Appendix A. Mathematical Proofs

### Appendix A.1. Frequency-Domain Downsampling Matrix $\Xi$

Using the general radix identity, with  $N = ab$ , the inverse DFT matrix can be decomposed as:

$$F_N^{-1} = P_{a,b}^{-1}(I_a \otimes F_b^{-1})T_{a,b}^{-1}(F_a^{-1} \otimes I_b) \quad (\text{A1})$$

Then, the frequency-domain downsampling matrix  $\Xi$  is:

$$\Xi = F_N^{-1}(I_a \otimes 1_b^T)F_a \quad (\text{A2})$$

$$= P_{a,b}^{-1}(I_a \otimes F_b^{-1})T_{a,b}^{-1}(F_a^{-1} \otimes I_b)(I_a \otimes 1_b^T)F_a \quad (\text{A3})$$

$$= P_{a,b}^{-1}(I_a \otimes F_b^{-1})T_{a,b}^{-1}(F_a^{-1} \otimes 1_b^T)F_a \quad (\text{A4})$$

$$= P_{a,b}^{-1}(I_a \otimes F_b^{-1})T_{a,b}^{-1}(I_a \otimes 1_b^T) \quad (\text{A5})$$

$$= P_{a,b}^{-1}(I_a \otimes F_b^{-1})(I_a \otimes 1_b^T) \quad (\text{A6})$$

$$= \frac{1}{b}P_{a,b}^{-1}(I_a \otimes \theta_b^T) \quad (\text{A7})$$

$$= \frac{1}{b}\theta_b^T \otimes I_a \quad (\text{A8})$$

### Appendix A.2. Periodic Scrambler

Using the general radix identity, with  $N = KL$ , the DFT matrix can be decomposed as:

$$F_N = (F_K \otimes I_L)T_{K,L}(I_K \otimes F_L)P_{K,L} \quad (\text{A9})$$

Then, we have:

$$\bar{s} = sF_N \quad (\text{A10})$$

$$= (\theta_K \otimes p)(F_K \otimes I_L)T_{K,L}(I_K \otimes F_L)P_{K,L} \quad (\text{A11})$$

$$= (\theta_K F_K) \otimes (p I_L)T_{K,L}(I_K \otimes F_L)P_{K,L} \quad (\text{A12})$$

$$= K(1_K \otimes p)T_{K,L}(I_K \otimes F_L)P_{K,L} \quad (\text{A13})$$

$$= K(1_K \otimes p)(I_K \otimes F_L)P_{K,L} \quad (\text{A14})$$

$$= K(1_K I_K) \otimes (p F_L)P_{K,L} \quad (\text{A15})$$

$$= K(1_K \otimes \bar{p})P_{K,L} \quad (\text{A16})$$

$$= K\bar{p} \otimes 1_K \quad (\text{A17})$$

### Appendix A.3. Commutation Property of $C_{\bar{x}}^{(K)}$

$$C_{\bar{x}}^{(K)} J_N^K = (K I_L \otimes 1_K) C_{\bar{x}} J_N^K \quad (A18)$$

$$= K(I_L \otimes 1_K) J_N^K C_{\bar{x}} \quad (A19)$$

$$= K(I_L \otimes 1_K)(J_L \otimes I_K) C_{\bar{x}} \quad (A20)$$

$$= K(I_L J_L) \otimes (1_K I_K) C_{\bar{x}} \quad (A21)$$

$$= J_L (K I_L \otimes 1_K) C_{\bar{x}} \quad (A22)$$

$$= J_L C_{\bar{x}}^{(K)} \quad (A23)$$

where we have used Equation (24).

## References

- Shannon, C.E. Communication in the presence of noise. *Proc. IRE* **1949**, *37*, 10–21. [\[CrossRef\]](#)
- Nyquist, H. Certain topics in telegraph transmission theory. *Trans. Am. Inst. Electr. Eng.* **1928**, *47*, 617–644. [\[CrossRef\]](#)
- Upadhyaya, V.; Salim, M. Compressive Sensing: Methods, Techniques, and Applications. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Jeju Island, Korea, 12–14 March 2021; IOP Publishing: Bristol, UK, 2021; Volume 1099, p. 012012.
- Fang, J.; Wang, B.; Li, H.; Liang, Y.C. Recent Advances on Sub-Nyquist Sampling-Based Wideband Spectrum Sensing. *IEEE Wirel. Commun.* **2021**, *28*, 115–121. [\[CrossRef\]](#)
- Candès, E.J.; Wakin, M.B. An introduction to compressive sampling. *IEEE Signal Process. Mag.* **2008**, *25*, 21–30. [\[CrossRef\]](#)
- Donoho, D.L. Compressed sensing. *IEEE Trans. Inf. Theory* **2006**, *52*, 1289–1306. [\[CrossRef\]](#)
- Zelnik-Manor, L.; Rosenblum, K.; Eldar, Y.C. Sensing matrix optimization for block-sparse decoding. *IEEE Trans. Signal Process.* **2011**, *59*, 4300–4312. [\[CrossRef\]](#)
- Abolghasemi, V.; Ferdowsi, S.; Makkiabadi, B.; Sanaei, S. On optimization of the measurement matrix for compressive sensing. In Proceedings of the European Signal Processing Conference, Aalborg, Denmark, 23–27 August 2010.
- Mishali, M.; Eldar, Y.C. From theory to practice: Sub-Nyquist sampling of sparse wideband analog signals. *IEEE J. Sel. Top. Signal Process.* **2010**, *4*, 375–391. [\[CrossRef\]](#)
- Mishali, M.; Hilgendorf, R.; Shoshan, E.; Rivkin, I.; Eldar, Y.C. Generic sensing hardware and real-time reconstruction for structured analog signals. In Proceedings of the 2011 IEEE International Symposium of Circuits and Systems (ISCAS), Rio de Janeiro, Brazil, 15–18 May 2011; pp. 1748–1751.
- Mishali, M.; Eldar, Y.C.; Dounaevsky, O.; Shoshan, E. Xampling: Analog to digital at sub-Nyquist rates. *IET Circuits Devices Syst.* **2011**, *5*, 8–20. [\[CrossRef\]](#)
- Laska, J.N.; Kirolos, S.; Duarte, M.F.; Ragheb, T.S.; Baraniuk, R.G.; Massoud, Y. Theory and implementation of an analog-to-information converter using random demodulation. In Proceedings of the IEEE International Symposium on Circuits and Systems, New Orleans, LA, USA, 27–30 May 2007; pp. 1959–1962.
- Israeli, E.; Tsiper, S.; Cohen, D.; Shoshan, E.; Hilgendorf, R.; Reysenson, A.; Eldar, Y.C. Hardware calibration of the modulated wideband converter. In Proceedings of the Global Communications Conference (GLOBECOM), Austin, TX, USA, 8–12 December 2014; pp. 948–953.
- Adams, D.; Eldar, Y.C.; Murmann, B. A Mixer Front End for a Four-Channel Modulated Wideband Converter With 62-dB Blocker Rejection. *IEEE J. Solid-State Circuits* **2017**, *52*, 1286–1294. [\[CrossRef\]](#)
- Mishali, M.; Eldar, Y.C.; Dounaevsky, O.; Shoshan, E. Sub-Nyquist acquisition hardware for wideband communication. In Proceedings of the 2010 IEEE Workshop on Signal Processing Systems, San Francisco, CA, USA, 6–8 October 2010; pp. 156–161.
- Eldar, Y.C. *Sampling Theory: Beyond Bandlimited Systems*; Cambridge University Press: Cambridge, UK, 2015.
- Cohen, D.; Tsiper, S.; Eldar, Y.C. Analog-to-Digital Cognitive Radio: Sampling, Detection, and Hardware. *IEEE Signal Process. Mag.* **2018**, *35*, 137–166. [\[CrossRef\]](#)
- Liu, W.; Huang, Z.; Wang, X.; Sun, W. Design of a single channel modulated wideband converter for wideband spectrum sensing: Theory, architecture and hardware implementation. *Sensors* **2017**, *17*, 1035. [\[CrossRef\]](#) [\[PubMed\]](#)
- Burel, G.; Fiche, A.; Gautier, R.; Martin-Guennou, A. A Modulated Wideband Converter Calibration Technique Based on a Single Measurement of a White Noise Signal with Advanced Resynchronization Preprocessing. *Electronics* **2022**, *11*, 774. [\[CrossRef\]](#)
- Wang, P.; You, F.; He, S. An improved signal reconstruction of modulated wideband converter using a sensing matrix built upon synchronized modulated signals. *Circuits Syst. Signal Process.* **2019**, *38*, 3187–3210. [\[CrossRef\]](#)
- Fu, N.; Jiang, S.; Deng, L.; Qiao, L. Successive-phase correction calibration method for modulated wideband converter system. *IET Signal Process.* **2019**, *13*, 624–632. [\[CrossRef\]](#)
- Park, J.; Jang, J.; Lee, H.N. A calibration for the modulated wideband converter using sinusoids with unknown phases. In Proceedings of the 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), Milan, Italy, 4–7 July 2017; pp. 951–955.

23. Alp, Y.K.; Korucu, A.B.; Karabacak, A.T.; Gürbüz, A.C.; Arikan, O. Online calibration of modulated wideband converter. In Proceedings of the 2016 24th Signal Processing and Communication Application Conference (SIU), Zonguldak, Turkey, 16–19 May 2016; pp. 913–916.
24. Byambadorj, Z.; Asami, K.; Yamaguchi, T.; Higo, A.; Fujita, M.; Iizuka, T. A Calibration Technique for Simultaneous Estimation of Actual Sensing Matrix Coefficients on Modulated Wideband Converters. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2020**, *67*, 5561–5573. [[CrossRef](#)]
25. Byambadorj, Z.; Asami, K.; Yamaguchi, T.J.; Higo, A.; Fujita, M.; Iizuka, T. High-Precision Sub-Nyquist Sampling System Based on Modulated Wideband Converter for Communication Device Testing. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2021**, *69*, 378–388. [[CrossRef](#)]
26. D’Angeli D.; Donno A. Shuffling Matrices, Kronecker Product and Discrete Fourier Transform. *Discret. Math.* **2017**, *233*, 1–18. [[CrossRef](#)]
27. Penrose, R. A generalized inverse for matrices. In *Mathematical Proceedings of the Cambridge Philosophical Society*; Cambridge University Press: Cambridge, UK, Volume 51, 1955; pp. 406–413.
28. Golub, G.H.; Van Loan, C.F. *Matrix Computations*, 3rd ed.; The Johns Hopkins University Press: Baltimore, London, 1996; ISBN 0-8018-5413-8.
29. Cai, T.T.; Wang, L. Orthogonal matching pursuit for sparse signal recovery with noise. *IEEE Trans. Inf. Theory* **2011**, *57*, 4680–4688. [[CrossRef](#)]
30. Nguyen, L.L.; Fiche A., Gautier, R.; Canaff, C.; Radoi, E.; Burel, G.; Implementation of Modulated Wideband Converter compressed sensing scheme based on COTS low-pass filter with amplitude and phase compensation for spectrum monitoring. In Proceedings of the 2018 IEEE International Conference on Advanced Video and Signal-based Surveillance (AVSS), Auckland, New Zealand, 27–30 November 2018. [[CrossRef](#)]
31. Sondergaard, P.L. Next Fast FFT Size. Available Online: <https://lftat.github.io/notes/lftatnote017.pdf> (accessed on 23 July 2022).
32. Daponte, P.; De Vito, L.; Iadarola, G.; Iovini, M.; Rapuano, S. Experimental comparison of two mathematical models for Analog-to-Information Converters. In Proceedings of the 21st IMEKO TC4 International Symposium and 19th International Workshop on ADC Modelling and Testing Understanding the World through Electrical and Electronic Measurement, Budapest, Hungary, 7–9 September 2016.