Abstract—We propose an entirely redesigned framework of bandlimited signal reconstruction for the time encoding machine (TEM) introduced by Lazar and Tóth in [1]. As the encoding part of TEM consists in obtaining integral values of a bandlimited input over known time intervals, it theoretically amounts to applying a known linear operator on the input. We then approach the general question of signal reconstruction by pseudo-inversion of this operator. We perform this task numerically and iteratively using projections onto convex sets (POCS). The algorithm can be implemented exactly in discrete time with multiplications that are all reduced to scaling by signed powers of two, thanks to the use of relaxation coefficients. Meanwhile, the algorithm achieves a rate of convergence similar to that of [1]. For real-time processing, we propose an approximate time-varying FIR implementation, which avoids the splitting of the input into blocks. We finally propose some preliminary semi-convergence analysis of the algorithm under data noise.

Index Terms—bandlimited signals, nonuniform sampling, time encoding machine, interpolation, minimal norm, pseudo-inverse, Kaczmarz method, POCS, frame algorithm, semi-convergence.

I. INTRODUCTION

Reconstructing bandlimited signals from nonuniform samples is an old problem [2], [3], [4], [5] that attracted relatively low attention in signal processing for quite a long time due to the necessity of complex digital postprocessing for signal recovery. This topic has gained revived interest with the recent trend of event-based signal processing [6]. The main motivations behind this movement of research has been the higher demand for low power and low complexity acquisition devices, while digital postprocessing is becoming more accessible. One of the main approaches has been to resolve the typical weakness of analog circuits with the acquisition of amplitude values by exploiting their superior accuracy in time detection. A breakthrough in this direction has been the use by Lazar and Tóth of an asynchronous Sigma-Delta modulator (ASDM) to extract the integral values of a bandlimited input \( x(t) \) over consecutive intervals [1], [7]. This method owes its appeal to the high simplicity of the time encoder, together with its built-in robustness to analog circuit imperfections. This outstanding acquisition device has given higher motivations to invest in the difficult problem of signal recovery, regarding theory as well as practical implementation. Most of the reconstruction methods available for this time encoding technique have lied in the publications of Lazar and Tóth [1], [8], [9]. In this paper, we propose to revisit signal reconstruction in this problem, all the way from theoretical foundations to real-time implementations.

From the switching instants of an ASDM with an input \( x(t) \), [1] showed a way to extract with high precision a sequence \( (t_i, x_i)_{i \in \mathbb{Z}} \) such that

\[
    x_i = \int_{t_{i-1}}^{t_i} x(t) \, dt, \quad i \in \mathbb{Z}. \tag{1}
\]

Here, \( \mathbb{Z} \) denotes either \( \mathbb{Z} \) or a finite index set \( \{1, \cdots, N\} \). This can be formally rewritten as

\[
    x_i = \langle h_i, x \rangle, \quad i \in \mathbb{Z} \tag{2}
\]

where \( \langle \cdot, \cdot \rangle \) is the canonical inner-product of \( L^2(\mathbb{R}) \) and

\[
    h_i(t) := \begin{cases} 
    1, & t \in [t_{i-1}, t_i) \\
    0, & \text{otherwise} 
\end{cases} \tag{3}
\]

One can go one step further in formalization. Assuming that \( x(t) \) belongs to a known space \( \mathcal{B} \) of bandlimited signals and \( (t_i - t_{i-1})_{i \in \mathbb{Z}} \) is upper bounded, (2) amounts to writing

\[
    x := (x_i)_{i \in \mathbb{Z}} = S x. \tag{4}
\]

where \( S \) is the linear operator

\[
    S : \mathcal{B} \rightarrow \ell^2(\mathbb{Z}) \\
    u \mapsto (\langle h_i, u \rangle)_{i \in \mathbb{Z}}
\]

and \( \ell^2(\mathbb{Z}) \) is the space of square summable sequences indexed by \( \mathbb{Z} \). Under this presentation, the approach of [1] was to first find a sufficient condition for the equation

\[
    S u = x \tag{5}
\]

to yield a unique solution, then obtain \( x = S^{-1} x \) by iterating a contraction mapping. The approach of the present paper is more generally to reconstruct the signal

\[
    x^\dagger = S^\dagger x \tag{6}
\]

where \( S^\dagger \) is the pseudo-inverse of \( S \), whenever it exists. This solution has the advantage to coincide with \( x(t) \) whenever (5) has a unique solution, while providing the minimal-norm solution of (5) as an estimate of \( x(t) \) when \( S \) is a many-to-one mapping. But another advantage is the elegant reconstruction framework that this implies: \( x^\dagger \) is the unique solution of (5) that lies in the closed linear span

\[
    S g := \operatorname{span}(g_i)_{i \in \mathbb{Z}},
\]

where \( g_i \) is the bandlimited version of \( h_i \). The coefficients of the expansion of \( x^\dagger \) in \( (g_i)_{i \in \mathbb{Z}} \) thus provide an exact discrete-time characterization of \( x^\dagger \). This gives the structure under
which exact discrete-time iterative reconstructions are possible and will be performed in this paper.

To be consistent with real practice, we later assume that $Z = \{1, \cdots, n\}$, in which case $S^\dagger$ always exists. The equivalent characterization of $x^\dagger$ as the minimal-norm solution of $Su = x$ leads us to employ the Kaczmarz method [10] for its reconstruction. While this algorithm finds its natural implementation in $B$, a more efficient version of it is obtained in the augmented space of $L^2(\mathbb{R})$, taking advantage of the orthogonality of $(\langle h_i, v \rangle)_{i \in Z}$. This invokes the more general framework of projection onto convex sets (POCS) [11, 12]. Interestingly, this method has been simultaneously adopted in [13, 14] for its ability to deal with multi-channel time-encoding machines. The optimized POCS’s appear to yield similar convergence rates to the method of [1]. But the outstanding feature of interest of the POCS’s in this paper is the flexible use of relaxation coefficients, which allows us to eliminate all full resolution multiplications in its discrete-time implementation by reducing them to scaling with signed powers of 2, implementable in hardware by mere bit shifts [15]. A remaining difficulty is the involvement of inner-products of the type $\langle g_i, h_j \rangle$ in the computation. By generalizing a technique from [16], we show that these coefficients can be obtained from a one-variable lookup table, up to some refined time quantization and some additions.

The next contribution of the present paper is a different approach to practical implementation. When trying to invert a time-varying linear operator such as $S$, one cannot rely on Fourier techniques and often resorts to matrix algebra. Due to the virtually infinite size of this operator compared to the practical windows of operation, the technique adopted in [9] is to perform within overlapping blocks of the input, an approximate modeling of finite-dimensional input space and an exact algebraic matrix inversions with the approximated space. Our approach is on the contrary to preserve the original virtually infinite dimensional signal space and approximate instead the POCS-based inversion process in the form of finite-length sliding window operations. This allows regular pipeline circuit architectures similar to time-varying FIR filters, suitable for real-time signal processing [17].

As a final contribution, we propose some preliminary analysis of the POCS algorithm in the presence of sampling noise.

II. TIME ENCODING MACHINE

We briefly review the principles of time encoding introduced in [1] of particular interest to us. All continuous-time signals are assumed to be in the real Hilbert space $L^2(\mathbb{R})$ equipped with the inner-product $(u, v) := \int_\mathbb{R} u(t)v(t)dt$ and the norm $\|u\| := (u, u)^{1/2}$. To simplify, we assume that $B$ is the subspace of bandlimited functions of Nyquist period 1.

A. Encoder

The time encoding of a signal $x(t)$ of $B$ as proposed in [1] consists in feeding it into an ASDM as shown in Fig. 1 and recording the successive instants $\tau_i$ when the output $z(t)$ switches between +1 and −1. It is shown in [1] that

$$\int_{\tau_i}^{\tau_{i+1}} x(t) \, dt = (−1)^i((\tau_{i+1}−\tau_i)−2d) \quad (6)$$

where ±d are the thresholds of the Schmitt trigger. The integral value dependence with the circuit parameter d can be eliminated by considering only the integrals of $x(t)$ between the even-indexed instants $\tau_{2i}$. Defining

$$t_i := \tau_{2i} \quad \text{and} \quad x_i := (\tau_{2i}−\tau_{2i−1})−(\tau_{2i−1}−\tau_{2i−2}), \quad (7)$$

one easily obtains the relation (1) for each $i \in Z$. The remaining impact of the parameter d is however in the density of the instants $t_i$. Defining

$$T_m := \sup_{i \in Z} (t_i−t_{i−1})$$

it is also shown that $T_m \leq 2d/(1−x_m)$ where $x_m$ is the maximum amplitude of $x(t)$.

B. Signal reconstruction

The method proposed in [1] to recover $x(t)$ for this estimation can be presented as follows. For any given estimate $u$ of $x$, let $r_i(u)$ be the $i$th residual error

$$r_i(u) := x_i − \langle h_i, u \rangle \quad (8)$$

and $\varphi(t)$ be the sinc function of Nyquist period 1

$$\varphi(t) := \sin(\pi t)/(\pi t).$$

Then, [1] performs in $B$ the iteration

$$x^{(n+1)} := Qx^{(n)} \quad (9)$$

where for all $u \in B$,

$$Qu := u + \sum_{i \in Z} r_i(u) \varphi_i, \quad (10)$$

$$\varphi_i(t) := \varphi(t−s_i) \quad \text{and} \quad s_i := \frac{1}{2}(t_i−t_{i−1}).$$

Given the expression of $r_i(u)$, $Q$ is an affine transformation. Calling $M$ its linear part, we have $Qu = Mu + c$ where

$$Mu := u − \sum_{i \in Z} \langle h_i, u \rangle \varphi_i \quad \text{and} \quad c := \sum_{i \in Z} x_i \varphi_i. \quad (11)$$

Due to (2), $r_i(x) = 0$ for all $i \in Z$. Hence, $x$ is a fixed point of $Q$, i.e., $Qx = x$. Thus,

$$Qu − x = Qu − Qx = Mu − Mx = M(u−x).$$

By applying $n$ times this identity on $u = x^{(0)}$, one obtains

$$x^{(n)} − x = M^n(x^{(0)}−x).$$

It is shown in [1] that $\|M\| \leq T_m$. So by adjusting $d < (1−x_m)/2$, $\|M\|$ is less than 1. Thus, $Q$ is a contraction and $x^{(n)}$ tends to $x$. 

![Fig. 1. Encoder of the time encoding machine](image-url)
III. Sampling pseudo-inversion

A. Sampling operator

Before characterizing the pseudo-inverse $S^\dagger$ of $S$ defined in (4), we need to finalize a number of points. One first needs to verify that $S$ does map \mathcal{B} into $\ell^2(\mathbb{Z})$. By Cauchy-Schwarz inequality
\[ |\langle h_i, u \rangle|^2 \leq \int_{t_i}^{t_i} 1 dt \int_{t_{i-1}}^{t_{i-1}} |u(t)|^2 dt \leq T_m \int_{t_{i-1}}^{t_i} |u(t)|^2 dt. \]
So, $\sum_{i \in \mathbb{Z}} |\langle h_i, u \rangle|^2 \leq T_m |u|^2$. This shows moreover that $S$ is a bounded operator. For a proper definition of $S$, the second issue is that the functions $\langle h_i \rangle_{i \in \mathbb{Z}}$ are not in its domain $\mathcal{B}$. This is easily fixed as follows. For each $i \in \mathbb{Z}$, the function $g_i$, which can also be presented as
\[ g_i := \varphi * h_i, \]
is the orthogonal projection of $h_i$ onto $\mathcal{B}$. Consequently,
\[ \forall u \in \mathcal{B}, \quad \langle h_i, u \rangle = \langle g_i, u \rangle. \]
Thus, $S$ is equivalently described as
\[ S: \mathcal{B} \rightarrow \ell^2(\mathbb{Z}) \]
\[ u \mapsto (\langle g_i, u \rangle)_{i \in \mathbb{Z}}. \]

B. Pseudo-inverse of operator

The pseudo-inverse $S^\dagger$ is defined whenever the range of $S$ is closed [18, Lemma 2.5.1]. To the best of the authors’ knowledge, the most general condition on $(t_i)_{i \in \mathbb{Z}}$ for this to be realized for the specific operator $S$ of (14) is not currently established. But whenever this is satisfied, $x^\dagger := S^\dagger x$ yields the following properties:

(a) $x^\dagger$ is the unique minimal-norm solution of equation (5).
(b) $x^\dagger$ is the orthogonal projection of any solution of (5) (including $x$) onto $\mathcal{S}_g$.
(c) $x^\dagger$ is the unique solution of (5) that is in $\mathcal{S}_g$. The justification of the first two properties can be found in [18] and specifically as follows. Property (a) is a consequence of [18, Theorem 2.5.3]. Property (b) involves the adjoint operator $S^*$ of $S$. By [18, Lemma 3.1.1], $S^*$ is defined by
\[ S^*: \ell^2(\mathbb{Z}) \rightarrow \mathcal{B} \]
\[ (c_i)_{i \in \mathbb{Z}} \mapsto \sum_{i \in \mathbb{Z}} c_i g_i. \]
As stated in [18, Lemma 2.5.2(ii)], $S^\dagger S$ is the orthogonal projection of $\mathcal{B}$ onto the range of $S^*$, which is closed by [18, Lemma 2.4.1(ii)] and coincides with $\mathcal{S}_g$. This justifies property (b). Property (c) is an easy consequence of (b).

C. Finite encoding

The eternal pain in the analysis of bandlimited signals is the discrepancy between the obvious constraint of finite time window of signal processing in practice, and the necessity of an infinite time window for the rigorous description of bandlimited signals. In real operations, $Z$ is always a finite set
\[ Z = \{1, \ldots , N\} \]
which we assume from now on in the paper. When using an infinite-sampling-based perfect reconstruction method with a finite number of samples, one can only expect to obtain an estimate $\hat{x}(t)$ that matches $x(t)$ asymptotically within the acquisition interval as $t$ gets away from its boundaries, with a behavior near the boundaries that is not under explicit control. Meanwhile, the minimal-norm reconstruction $x^\dagger(t)$ has a designed controlled behavior at the boundaries, while carrying the same asymptotic properties as $\hat{x}(t)$ inside the interval. The fact is that $S^\dagger$ always exists when $Z$ is finite since the range of $S$ is closed by finite dimension. From property (c), $x^\dagger$ is then nothing but a linear combination of $(g_i)_{i \in \mathbb{Z}}$.

But the more general advantage of pseudo-inversion is that $x^\dagger$ is well defined for any sequence $(t_i)_{i \in \mathbb{Z}}$. When $T_m > 1$, which corresponds to “undersampling”, one has a precise characterization of $x^\dagger$ as a particular solution of (5) which can still be of use for the estimation of $x$.

The minimal-norm reconstruction of a bandlimited signal from a finite number of nonuniform samples was initially introduced by Yen in [3] in the case where $g_i = \varphi(t-t_i)$ for all $i \in \mathbb{Z}$. Although the connection to pseudo-inversion was not made, it was shown that $x^\dagger$ is a linear combination of $(g_i)_{i \in \mathbb{Z}}$ by more constrained minimization via Lagrange multipliers.

D. Discrete-time algorithm and continuous-time output

In absence of closed form expression for $S^\dagger$, $x^\dagger(t)$ can only be numerically approached by successive approximations $(x^{(n)}(t))_{n \geq 0}$ via some algorithm. Iterations in continuous time are however to be avoided due to the precision limitation of analog circuits. By forcing $x^{(n)}$ to be in $\mathcal{B}$, a standard signal processing reflex is to work in discrete time with its uniform Nyquist-rate samples. But one faces again a theoretical difficulty as an infinite number of them would still be needed to give an exact description of $x^{(n)}$. Now, pseudo-inversion is such that $x^\dagger$ belongs to the finite-dimensional space $\mathcal{S}_g$. It is then natural to seek the estimates $x^{(n)}$ in $\mathcal{S}_g$. This is achieved by having
\[ x^{(n)} = S^* c^{(n)} \]
where $c^{(n)} = (c_i^{(n)})_{i \in \mathbb{Z}}$ is some $N$-dimensional vector. The goal is then to build a discrete-time algorithm for the sequence $(c_i^{(n)})_{n \geq 0}$ such that $x^{(n)}$ theoretically tends to $x^\dagger$. Once $c^{(n)}$ has been obtained after a satisfactory number $n$ of iterations, then $x^{(n)}$ is output by applying the transformation (16) only once. Given (12), we have
\[ x^{(n)}(t) = \varphi(t) * \sum_{i \in \mathbb{Z}} c_i^{(n)} h_i(t). \]
This is nothing but the bandlimited version of the piecewise constant function equal to $c_i^{(n)}$ in $(t_{i-1}, t_i)$ for each $i \in \mathbb{Z}$. This is implemented in circuits by a zero-order hold followed by a lowpass filter.

IV. Minimal-norm reconstruction by POCS

A. POCS method in $\mathcal{B}$

Given the description of $S$ in (14), an estimate $u \in \mathcal{B}$ satisfies the equation $Su = x$ of (5) if and only if $u$ is in
the intersection
\[ \mathcal{I} := \mathcal{G}_1 \cap \mathcal{G}_2 \cap \cdots \cap \mathcal{G}_N \]
where for each \( i \in \mathbb{Z} \),
\[ \mathcal{G}_i := \{ u \in \mathcal{B} : \langle g_i, u \rangle = x_i \}. \] (17)

This set is affine and closed. For any closed and affine subspace \( \mathcal{A} \), let us denote by \( P_\mathcal{A} \) the orthogonal projection onto \( \mathcal{A} \). For any \( u, P_\mathcal{A} u \) is the unique element of \( \mathcal{A} \) such that \( u - P_\mathcal{A} u \) is in the orthogonal complement of the direction of \( \mathcal{A} \), which we denote by \( \mathcal{A}^\perp \). Given that the sets \( \mathcal{G}_i \) are more specifically hyperplanes, it was shown by Kaczmarz [19] that the sequence \( (x^{(n)})_{n \geq 0} \) recursively defined by
\[ x^{(n+1)} = P_{\mathcal{G}_1} \cdots P_{\mathcal{G}_n} x^{(n)} \] (18)
converges to a point of \( \mathcal{I} \). It is known from [20] that \( x^{(n)} \) specifically tends to \( P_{\mathcal{I}} x^{(0)} \). By taking \( x^{(0)} = 0 \), \( P_{\mathcal{I}} x^{(0)} \) is exactly the minimal-norm element \( x^\dagger \) of \( \mathcal{I} \).

This type of method was previously used by Feichtinger and Grochenig in [4] for the interpolation of bandlimited signals from nonuniform point samples. It was introduced under the more general framework of projections onto convex sets (POCS) [11], [12]. In this sampling application, it was however found that the POCS iteration suffers slow convergence. This explains the low popularity of this method in bandlimited nonuniform sampling until now.

B. POCS method in \( L^2(\mathbb{R}) \)

We bring back the POCS method to the forefront in this paper due to two outstanding properties. The first one results from a particular feature of the kernel functions \( (g_i)_{i \in \mathbb{Z}} \) that does not exist with point sampling of bandlimited functions. By construction in (12), they are the orthogonal projections onto \( \mathcal{B} \) of \( (h_i)_{i \in \mathbb{Z}} \), which are orthogonal in the larger Hilbert space of \( L^2(\mathbb{R}) \). Returning to the original sampling description of (2), the solution set \( \mathcal{I} \) can be alternatively described as
\[ \mathcal{I} := \mathcal{H}_1 \cap \mathcal{H}_2 \cap \cdots \cap \mathcal{H}_N \cap \mathcal{B} \]
where for each \( i \in \mathbb{Z} \),
\[ \mathcal{H}_i := \{ u \in L^2(\mathbb{R}) : \langle h_i, u \rangle = x_i \}. \] (19)

The new iteration
\[ x^{(n+1)} = P_\mathcal{H}_1 \cdots P_\mathcal{H}_n R_\lambda x^{(n)} \] (20)
then also converges to \( x^\dagger \) with \( x^{(0)} = 0 \). With the orthogonality of \( (h_i)_{i \in \mathbb{Z}} \), we will see in Section IV-H that this method has a convergence rate closer to that of Lazar and Tóth.

C. Relaxation coefficients

The second outstanding property of the POCS method is its maintained convergence with relaxation coefficients. For any closed and affine subspace \( \mathcal{A} \) and \( \lambda \in \mathbb{R} \), consider the relaxed projection
\[ P_\lambda x := x + \lambda(P_\mathcal{A} x - x). \] (21)

For a given sequence \( (\lambda^{(n)})_{n \geq 0} \) of \( N \)-dimensional vectors, consider the iteration
\[ x^{(n+1)} = R_\lambda x^{(n)} \] (22)
where for any vector \( \lambda = (\lambda_1, \cdots, \lambda_N) \),
\[ R_\lambda := P_\mathcal{B} P_{\mathcal{H}_1}^{\lambda_1} \cdots P_{\mathcal{H}_2}^{\lambda_2} P_{\mathcal{H}_1}^{\lambda_1}. \] (23)

This more general iteration is known to converge to a point of \( \mathcal{I} \) provided that \( \lambda^{(n)} \) remains in \( [e, 2 - e]^N \) for some constant \( e > 0 \) [21], [10]. A relaxation coefficient in \([e, 2-e]\) could also be applied to \( P_\mathcal{B} \), but we maintain it to 1 so that \( R_\lambda \) maps \( \mathcal{B} \) into itself.

As a generalization of [20], the limit \( x^{(\infty)} \) of \( x^{(n)} \) is also obtained. Since \( P_\mathcal{I} x^{(0)} = \lambda(P_\mathcal{A} x^{(0)}) \), this vector is in \( \mathcal{A}^\perp \). It is then easy to see from (23) that \( R_\lambda x^{(n)} = x^{(0)} \). Since \( \mathcal{I} \) is included in \( \mathcal{B} \) and \( \mathcal{H}_i \) for all \( i \in \mathbb{Z} \), \( \mathcal{B}^\perp \) and \( \mathcal{H}_i^\perp \) are included \( \mathcal{I}^\perp \). Then \( R_\lambda x = x \in \mathcal{I}^\perp \). By induction, one concludes from (22) that \( x^{(n)} \) is \( \mathcal{I}^\perp \) for all \( n \geq 0 \). Since \( x^{(n)} \) is \( \mathcal{I}^\perp \), the limit \( x^{(\infty)} \) exists in \( \mathcal{I}^\perp \). Again, with \( x^{(0)} = 0 \), \( x^{(0)} \) tends to \( x^\dagger \).

D. Explicit expression of \( R_\lambda \)

The orthogonality of \( (h_i)_{i \in \mathbb{Z}} \) allows a closed form expression for \( R_\lambda \).

**Proposition 4.1:** For any given \( \lambda = (\lambda_1, \cdots, \lambda_N) \),
\[ R_\lambda u = u + \sum_{i \in \mathbb{Z}} \lambda_i r_\lambda(u) \frac{h_i}{\|h_i\|^2} h_i \] (24)

**Proof:** Using the function \( r_\lambda(u) \) of (8), one can easily verify that for any given \( u \in L^2(\mathbb{R}) \),
\[ v := u + \frac{r_\lambda(u)}{\|h_i\|^2} h_i \]
is equal to \( P_{\mathcal{H}} u \), by seeing that \( v - u \in \mathcal{H}^\perp \) and checking that \( \langle h_i, v \rangle = x_i \). Since \( P_{\mathcal{H}}^\lambda u = u + \lambda_i (v - u) \) from (21), then
\[ P_{\mathcal{H}}^\lambda u = u + \alpha_i(u) h_i \]
where \( \alpha_i(u) := \lambda_i \frac{r_\lambda(u)}{\|h_i\|^2} \).

Let us show that
\[ P_{\mathcal{H}^k} \cdots P_{\mathcal{H}_2} P_{\mathcal{H}_1}^\lambda u = u + \sum_{i=1}^k \alpha_i(u) h_i \] (25)
for all \( k = 1, \cdots, N \) and \( u \in L^2(\mathbb{R}) \). Because \( (h_i)_{i \in \mathbb{Z}} \) is orthogonal, note from (12) that \( r_\lambda(u+\alpha h_j) = r_\lambda(u) \) for all \( \alpha \in \mathbb{R} \) and any distinct \( i, j \in \mathbb{Z} \). Therefore \( P_{\mathcal{H}_i}^\lambda(u+\alpha h_j) = u + \alpha h_j + \alpha_i(u) h_i = P_{\mathcal{H}_i}^\lambda u + \alpha h_j \). For any \( k \in \{1, \cdots, N\} \), we conclude that
\[ P_{\mathcal{H}^k} \left( u + \sum_{i=1}^{k-1} \alpha_i(u) h_i \right) = P_{\mathcal{H}^k} u + \sum_{i=1}^{k-1} \alpha_i(u) h_i = u + \sum_{i=1}^{k-1} \alpha_i(u) h_i. \]

One then obtains (25) by induction on \( k \). Finally, (24) is obtained for all \( u \in \mathcal{B} \) by applying \( P_\mathcal{B} \) to the members of (25) with \( k = N \).

It is clear from (24) that \( R_\lambda \) maps \( S_\delta \) into itself. Starting with \( x^{(0)} = 0 \), the iteration of (22) thus provides a sequence of estimates \( x^{(n)} \) in \( S_\delta \).
E. Qualitative comparison with the method of [1]

The expression of $R_\lambda$ in (24) can be seen as derived from $Q$ in (10) by replacing $\varphi$ by $\lambda_i g_i/\|h_i\|^2$. While the former function is $\varphi * \delta_z$ where $\delta_z$ is the Dirac located at $t = s_z$, the latter function is $\alpha_i \varphi * h_i$ with $\alpha_i := \lambda_i/\|h_i\|^2$. As opposed to $\delta_z$, the rectangular function $h_i$ “fills the gap” between $t_{i-1}$ and $t_i$. This is physically more satisfactory, and also explains intuitively why our present method can work without any condition on $T_m$. Meanwhile, we will see by linear algebra in the next section that $R_\lambda$ is mathematically a contraction, again without any assumption on $T_m$ unlike $Q$. Beyond this, an attractive feature of the iteration (22) is the degree of freedom in the adjustment of $(\lambda^{(n)})_{n \geq 0}$. We will see its double potential for convergence accelerations and hardware implementation simplification.

F. Contraction property of $R_\lambda$ in $S_g$

The convergence of Lazar & Tóth’s method was based on the contraction property of the transformation $Q$ when $T_m < 1$. A similar property can be shown with $R_\lambda$ with no condition on $T_m$. Similarly to (11), one finds for all $u \in \mathbb{B}$ that

$$R_\lambda u = M_\lambda u + c_\lambda$$

where

$$M_\lambda u := u - \sum_{i \in Z} \lambda_i \frac{\langle h_i, u \rangle}{\|h_i\|^2} g_i \quad \text{and} \quad c_\lambda := \sum_{i \in Z} \frac{x_i}{\|h_i\|^2} g_i.$$  

(26)

Since $x^\dagger$ is a solution of (5), $r_i(x^\dagger) = 0$ for all $i \in Z$, and hence $x^\dagger$ is a fixed point of $R_\lambda$, i.e., $R_\lambda(x^\dagger) = x^\dagger$. Then,

$$R_\lambda u - x^\dagger = R_\lambda u - R_\lambda x^\dagger = M_\lambda(u - x^\dagger).$$  

(27)

With $u = x^{(n)}$ and $\lambda = \lambda^{(n)}$, we then obtain

$$\|x^{(n+1)} - x^\dagger\| \leq \|M_\lambda^{(n)}\| \|x^{(n)} - x^\dagger\|$$

where

$$\|M_\lambda\| := \sup_{u \in S_g \setminus \{0\}} \frac{\|M_\lambda u\|}{\|u\|}.$$  

Note that $\|M_\lambda\|$ is the operator norm of $M_\lambda$ restricted to $S_g$. We show in Appendix A the following result.

**Theorem 4.2:** For any $\epsilon \in (0,1]$, there exists a positive constant $\gamma_\epsilon < 1$ such that

$$\forall \lambda \in [\epsilon, 2-\epsilon]^N, \quad \|M_\lambda\| \leq \gamma_\epsilon.$$  

(28)

Hence, by maintaining $\lambda^{(n)}$ in $[\epsilon, 2-\epsilon]^N$ for some $\epsilon > 0$, we obtain the contraction $\|x^{(n+1)} - x^\dagger\| \leq \gamma_\epsilon \|x^{(n)} - x^\dagger\|$ for all $n \geq 0$. This could have been shown as a consequence of the general POCS result of [12, Theorem 5.7] by working in the space $S_g$ and using property (c) of Section III-B. But the above theorem and its proof give insight that is more typical of linear algebra and will also be useful for further analysis.

G. Frame algorithm and over-relaxation

While the previous section gave broad conditions for the iteration (22) to converge to $x^\dagger$, one wishes to have some analytical insight on the effect of $\lambda^{(n)}$ on the rate of convergence. While this is a difficult problem in the general context of the Kaczmarz method [22], this analysis appears to be somewhat easier with the iteration (22), primarily due to the fact that the linear part $M_\lambda$ of $R_\lambda$ turns out to be self-adjoint within $\mathcal{B}$. This is fundamentally linked to the orthogonality of $(h_i)_{i \in Z}$ in $L^2(\mathbb{R})$. When $\lambda^{(n)}$ has all its coordinates equal to a constant $\lambda$, (22) moreover coincides with a frame algorithm [2], [23]. For convenience, we write

$$\lambda^{(n)} = \lambda 1 \quad \text{where} \quad 1 := (1,1,\cdots,1) \in \mathbb{R}^N.$$  

(29)

The theory of frames provides a technique to find the optimized relaxation value

$$\lambda_m := \underset{\lambda \in \mathbb{R}}{\text{argmin}} \|M_\lambda 1\|.$$  

Let us define

$$\hat{h}_i := h_i/\|h_i\| \quad \text{and} \quad \hat{g}_i := g_i/\|h_i\|.$$  

(30)

Since $\langle h_i, x^\dagger \rangle = x_i$, it is easy to see from (24), (8) and (13) that (22) takes the generic form of the frame algorithm

$$x^{(n+1)} = x^{(n)} + \lambda P(x^\dagger - x^{(n)})$$

where

$$Pu := \sum_{i \in Z} \langle \hat{h}_i, u \rangle \hat{g}_i = \sum_{i \in Z} \langle g_i, u \rangle g_i = \sum_{i \in Z} \langle \hat{g}_i, u \rangle \hat{g}_i$$

for all $u \in \mathbb{B}$. The transformation $P$ is exactly the frame operator of [4, (53)]. As $S_g$ is of finite dimension, we can define the bounds

$$A := \min_{u \in S_g} \sum_{i \in Z} |\langle \hat{g}_i, u \rangle|^2 \quad \text{and} \quad B := \max_{u \in S_g} \sum_{i \in Z} |\langle \hat{g}_i, u \rangle|^2.$$  

This implies that for all $u \in S_g$,

$$A\|u\|^2 \leq \sum_{i \in Z} |\langle \hat{g}_i, u \rangle|^2 \leq B\|u\|^2.$$  

Under this perspective, it was proved in [2] that

$$\lambda_m = \frac{2}{A+B} \quad \text{and} \quad \|M_{\lambda_m} 1\| = \frac{B-A}{2A}.$$  

With our specific operator $P$, we obtain the following.

**Proposition 4.3:** $0 < A \leq B < 1$.

**Proof:** For any $u \in S_g \setminus \{0\}$, $\sum_{i \in Z} |\langle \hat{g}_i, u \rangle|^2 > 0$. So $A > 0$. Next, we can also write $P u = P_B \sum_{i \in Z} \langle \hat{h}_i, u \rangle \hat{h}_i = P_B P_B u$ where $P_B$ is the linear span of $(\hat{h}_i)_{i \in Z}$, since this family is orthonormal. As a nonzero bandlimited function cannot be piecewise constant, $u$ is not in $S_B$, so $\|P u\| \leq \|P_{B_B} u\|$. With Cauchy-Schwarz inequality, $\sum_{i \in Z} |\langle \hat{g}_i, u \rangle|^2 \leq \langle u, Pu \rangle \leq \|u\| \|P u\| < \|u\|^2$. Thus $B < 1$.

This implies that $\lambda_m > 1$. This falls in the case of over-relaxation, which is typically the result of optimal relaxation with parallel projections [24], but derived here by connection to the frame algorithm. Meanwhile, contrary to usual frame results, there is no condition here on the instants $(t_i)_{i \in Z}$ as we worked within the space $S_g$. In practice, as $A$ and $B$ may not be analytically available, the value of $\lambda$ is often optimized by experimentation.
over one period of an input $x(t)$. As a reference, we show the result of applying the basic Kaczmarz iteration of (20) in (a). The plotted MSE is averaged over 1500 drawn samples. From (4) and (16) that

$$r^{(n)} = x - Sx^{(n)} = x - Ac^{(n)}$$

and $e_i$ designates the $i$th coordinate basis vector of $\mathbb{R}^N$. As $S^*e_i = g_i$, one sees recursively that $S^*c^{(n)} = x^{(n)}$ by applying $S^*$ to the members of (31). We are going to see that the pair $(b^{(n)}, r^{(n)})$ yields a simple recursion. Since $r_i(x^{(n)}) = x_i - (h_i, x^{(n)})$, it follows from (33)

$$h_i^2 = T_i := t_i - t_{i-1}.$$  

The sequence $(c^{(n)})_{n \geq 0}$ is then recursively obtained by iterating the system

$$b^{(n+1)} = \sum_{i \in Z} \lambda_i^{(n)} e_i / T_i$$  

starting with $(r^{(0)}, c^{(0)}) = (x, 0)$. From (14) and (15), one finds that $A$ is the $N \times N$ matrix

$$A = \left[ \langle g_i, g_j \rangle \right]_{i,j \in Z}. $$

We will see in Section VII how the coefficients $\langle g_i, g_j \rangle$ can be obtained using a one-variable lookup table plus a few additions.
B. Relaxation function

The next goal is to adjust the coefficients \( \lambda^{(n)} \) so that the global complexity of the system (32) is low. From (32a), \( b^{(n)} \) is just the vector whose \( i \)th component is

\[
b_i^{(n)} = \lambda_i^{(n)} r_i^{(n)}/T_i. \tag{34}
\]

For systematic procedure, consider taking \( b_i^{(n)} \) directly as

\[
b_i^{(n)} = \beta_i^{(n)}(r_i^{(n)})\tag{35}
\]

where \( \beta_i(r) \) is some low complexity function such that \( \beta_i(0) = 0 \). This amounts to having (34) with

\[
\lambda_i^{(n)} := \left\{ \begin{array}{ll} T_i \beta_i(r_i^{(n)})/r_i^{(n)}, & r_i^{(n)} \neq 0 \\ 1, & r_i^{(n)} = 0 \end{array} \right. \tag{36}
\]

By imposing the function \( \beta_i(r) \) to satisfy for every \( i \in \mathbb{Z} \) the condition

\[
\beta_i(0) = 0 \quad \text{and} \quad \forall r \neq 0, T_i \beta_i(r)/r \in [\epsilon, 2-\epsilon], \tag{37}
\]

we guarantee that \( \lambda_i^{(n)} \in [\epsilon, 2-\epsilon] \) for all \( n \geq 0 \). We thus ensure the convergence of \( x^{(n)} \) from (22) to \( x^\dagger \) when \( x^{(0)} = 0 \). With (35), the system (32) takes the simpler form

\[
\begin{align*}
b^{(n)} &= B(r^{(n)}) \tag{38a} \\
r^{(n+1)} &= r^{(n)} - A b^{(n)} \tag{38b} \\
c^{(n+1)} &= c^{(n)} + b^{(n)} \tag{38c}
\end{align*}
\]

starting with \( (r^{(0)}, c^{(0)}) = (x, 0) \), where

\[
B(r) := \left( \beta_1(r_1), \ldots, \beta_N(r_N) \right), \quad r \in \mathbb{R}^N. \tag{39}
\]

C. Power-of-2 valued functions \( \beta_i(r) \)

Under the constraint of (37), it is possible to force \( \beta_i(r) \) to have values that are signed powers of 2. In this way, all multiplications involved in the product \( A b \) of (38b) are reduced to bit shifts. There are various ways to achieve this goal. In this paper, we consider functions \( \beta_i(\cdot) \) of the form

\[
\beta_i(r) := \rho(\lambda r/T_i) \tag{40}
\]

where \( \lambda \) is some chosen constant in \( (0, 2) \) and

\[
\rho(r) := \text{sign}(r) \max_{2^i \leq |r|} 2^i \tag{41}
\]

for all \( r \neq 0 \), with \( \rho(0) := 0 \). For any \( r > 0 \), it is clear that \( \frac{1}{2} r < \rho(r) \leq r \). So \( \rho(r)/r \in (\frac{1}{2}, 1] \), and as a result

\[
T_i \beta_i(r)/r = \lambda \frac{\rho(\lambda r/T_i)}{\lambda r/T_i} \in (\frac{1}{2}, 1] \tag{42}
\]

for all \( r > 0 \). By odd symmetry of the function \( \rho(\cdot) \), this is also true for all \( r \neq 0 \). As \( \lambda \in (0, 2) \), we obtain (37) with \( \epsilon = \min(\frac{1}{2}\lambda, 2-\lambda) > 0 \).

An apparent shortcoming of the function \( \beta_i(\cdot) \) of (40) is that it involves a multiplication and a division. There is a way to avoid them. Note that for any \( r \neq 0 \) and \( a > 0 \),

\[
\rho(\frac{r}{a}) = \text{sign}(r) \max_{2^i a \leq |r|} 2^i.
\]

This value is then found by simple inspection of the binary expansions of \(|r|\) and \(a\). Next, we calculate \( \beta_i(r) \) in the form

\[
\beta_i(r) = \rho(\frac{r}{T_i/\lambda}). \tag{43}
\]

The division by \( \lambda \) is not eliminated, but \( T_i/\lambda \) is to be computed only once for each \( i \in \mathbb{Z} \) before the iteration. Moreover, \( \lambda \) is only a constant parameter that is roughly and empirically adjusted to accelerate the convergence. We will see in the next section that good results are obtained with a value of \( \lambda \) of very low binary complexity.

D. Experimental results

Under the experimental conditions of Section IV-H, we plot in Fig. 2(f) \( ||x^{(n)} - x||^2 \) versus \( n \), where \( x^{(n)} = Sc^{(n)} \) and \( c^{(n)} \) is recursively obtained from the discrete-time system (38). The function \( B \) is defined by (39) and (43) where \( \lambda \) is taken to be \( (2^{-1}+2^{-4})^{-1} \approx 1.8 \). In this case, the division \( T_i/\lambda \) in (43) only requires a few bit shifts and one addition. Meanwhile, the error decay rate of curve (f) is higher than that of [1] in (a). The whole system (38) is multiplierless.

VI. REAL-TIME CIRCUIT IMPLEMENTATION

A. Approximate iteration

Concisely, the system (38) performs the transformation

\[
(r^{(n+1)}, c^{(n+1)}) = R(r^{(n)}, c^{(n)}) \tag{44}
\]

starting from \( (r^{(0)}, c^{(0)}) = (x, 0) \), where

\[
R(r, c) := (r - AB(r), c + B(r)), \quad r, c \in \mathbb{R}^N.
\]

The transformation \( B \) defined in (39) depends on the choice of functions \( \beta_1(r), \ldots, \beta_N(r) \) and is in general nonlinear. It is however memoryless when thinking of the components of \( r \) as a sequence of time. The issue is the multiplication by the matrix \( A \). Although \( A \) is theoretically of finite size, it is virtually infinite compared to the practical time windows of operation. Now, its coefficients \( \langle g_i, g_j \rangle \) typically tend to 0 when \(|i-j|\) tends to infinity. Like in rectangular windowing for the FIR implementation of lowpass filters, we consider truncating these coefficients as soon as \(|i-j|\) is larger than some parameter \( L \geq 0 \). This amounts to replacing \( A \) by the matrix \( \hat{A} \) of coefficients

\[
\hat{a}_{i,j} := \begin{cases} \langle g_i, g_j \rangle, & i, j \in \mathbb{Z} \text{ and } |i-j| \leq L \\ 0, & \text{otherwise} \end{cases}
\]

So, in real implementation, (44) is replaced by

\[
(r^{(n+1)}, c^{(n+1)}) = \hat{R}(r^{(n)}, c^{(n)}) \tag{46}
\]

where

\[
\hat{R}(r, c) := (r - \hat{A}B(r), c + B(r)), \quad r, c \in \mathbb{R}^N. \tag{47}
\]

B. Sliding-window pipeline implementation

We show in Fig. 3(a) a real-time pipeline implementation of the single transformation \( (r', c') = \hat{R}(r, c) \). It is derived as follows. From (47), we have

\[
r' = r - p \quad \text{and} \quad c' = c + b \tag{48}
\]

where \( p := \hat{A}b \) and \( b := B(r) \).

Using explicitly the multiplierless functions \( \beta_i \) of (43), the components of \( b \) are

\[
b_i = \beta_i(r_i) = \rho(\frac{r_i}{T_i/\lambda}). \tag{49}
\]
Meanwhile, the components of \( p \) are
\[
p_k = \sum_{j=k-L}^{k+L} \hat{a}_{k,j} b_j.
\]
Note that \( p_k \) depends on \( b_{k+L} \). So at a given instant \( k \), only \( p_{k-L} \) can be obtained in a causal manner. We have
\[
p_{k-L} = \sum_{j=k-L}^{k} \hat{a}_{k-L,j} b_j = \sum_{\ell=0}^{2L} \hat{a}_{k}^{\ell} b_{k-\ell}
\tag{50}
\]
where for each \( \ell \in \{0, \cdots, 2L\}, \)
\[
\hat{a}_{k}^{\ell} := \hat{a}_{k-L,k-\ell} = \left\{ \begin{array}{ll}
(2b_{k-L}, 2k-\ell), & k-L, k-\ell \in \mathbb{Z} \\
0, & \text{otherwise}
\end{array} \right.
\tag{51}
\]
Equations (48), (49) and (50) can then be mapped to the block diagram of Fig. 3(a). Each node signal is a function of the discrete-time index \( k \), which is incremented in real time from \( k-1 \) at the switching instant \( t_k \). The symbol \( D \) represents the delay operation with respect to \( k \). The dashed frame highlights the structure of time-varying FIR filter operating on the sequence \( (b_k)_{k \in \mathbb{Z}} \) of signed powers of 2.

Fig. 3(b) shows the global pipeline architecture for the computation of \( (x^{(n)}, c^{(n)}) = \tilde{R}^n(x, 0) \). The operation \( D^L \) is the delay by \( L \) discrete-time instants. We will show in Section VII how the coefficients \( \hat{a}_{k}^{\ell} \) can be obtained in real time by table lookup.

C. Relaxed bandlimitation

The coefficients \( \langle g_i, g_j \rangle \) are expected to decay with
\[
T_{i,j} := t_i - t_j
\tag{52}
\]
at the slow rate of \( 1/|T_{i,j}| \) due to their connection to the sinc function. As a classically known phenomenon, a plain truncation of such a sequence of coefficients is expected to induce disappointingly large errors. Advanced techniques of windowing are available for linear and time-invariant DSP, but not for the present case of time-varying operations. Moreover, the truncated operator \( \tilde{R} \) is iterated, making the process sensitive to in-band distortions. With the lack of knowledge in this problem, we propose to maintain the abrupt truncation of the coefficients \( \langle g_i, g_j \rangle \) but relax the bandlimiting function \( \varphi(t) \) involved in the definition (12) of \( g_i(t) \). Specifically, we maintain the flat in-band frequency response of \( \varphi(t) \) but allow a smooth cutoff transition (of cosine type) between the angular frequencies of \( \pi \) and \( \pi r \) for some coefficient \( r > 1 \). The purpose is to induce a faster decay rate of \( \varphi(t) \) to limit the damages due to truncation and eventually limit in-band distortions. Mathematically, this amounts to replacing \( P_B \) in the expression (23) of \( R_x \) by a non-ideal bandlimitation. One will naturally expect degradations in the efficiency of the POCS’s.

D. Experimental results

We show in Fig. 4 the effect of the various practical approximations on the multiply- and reconstruction scheme of Fig. 2(f), which is reproduced as curve (a) in Fig. 4. For a fair assessment, we only measure the final in-band mean squared error \( \|P_B(x^{(n)} - x)\|^2 \), which is the specific portion of the error that is irreversible. We report in (c) the performance degradation due to bandwidth relaxation alone with \( r = 1.4 \), as presented in the previous section. Under this condition, we next apply the truncation approximation of (46) with \( L = 17 \), which yields the result of curve (d). Although the experiment is performed on an input of period 257, it is representative of aperiodic inputs as the window of operation resulting from the truncation is only of approximate length 19 in average, which is small compared to the input period. As shown in the figure, 6 iterations are needed to obtain a reconstruction resolution of 8.5 bits. The total number of adders required by the system for \( n \) iterations is \( n(2L+2) + (7L+8) \) where \( 2L+2 \) is the complexity of \( \tilde{R} \) in Fig. 3(a) and \( 7L+8 \) is the required
how the sequence with additional non-idealities: (a) ideal case and the signal statistics on an idea introduced in complexity to compute the multidimensional input
\[
\hat{a}_k := (\hat{a}_{k,0}^1, \hat{a}_{k,1}^1, \cdots, \hat{a}_{k,2L}^1)
\] (53)
as will be shown in Section VII. With \( L = 17 \) and \( n = 6 \), this implies 343 adders. Roughly, we have observed that each additional bit of reconstruction resolution requires a doubling of the computation complexity. According to our observations, the bottleneck of reconstruction accuracy is the slow decay of the sinc function required for exact bandlimitation.

It is also interesting to see the behavior of the algorithm with additional noise. As a concrete source of noise, we choose the quantization in time of the switching instants \( \tau_n \) of the encoder. This implies errors on both \( t_k \) and \( s_k \) as can be seen in (7). We show the resulting additional degradation in curve (c) with the time-quantization step size of \( 2^{-12} \). This time resolution has been chosen by observing its effect in absence of all other distortions, as shown in curve (b). In fact, time quantization is necessary not only for digital processing, but also to limit the possible values of \( \langle g_i, g_j \rangle \) to a finite number so that they can be precalculated and stored in a lookup table. According to a method presented in Section VII and the signal statistics of the present experiment, this lookup table is evaluated to fit in a memory of less than 100 KB.

VII. MATRIX COEFFICIENTS BY TABLE LOOKUP

In the systems of Fig. 3, we assumed the availability of the vector \( \hat{a}_k \) of (53). Its components however need to be computed. In steady state, it follows from (51) that
\[
\hat{a}_k^\ell = \langle g_{k-L}, g_{k-\ell} \rangle, \quad \ell = 0, \ldots, 2L.
\] (54)
This is precisely satisfied for all \( k = 2L+1, \ldots, N \). Based on an idea introduced in [16] and following more elaborate derivations from [26], we show that these coefficients can be derived by pure additions from pre-stored values of a certain single-variable analytical function.

A. Expression of \( \langle g_i, g_j \rangle \)

We recall from (12) that \( g_i(t) = \varphi(t) * h_i(t) \), where \( \varphi(t) \) is ideally a sinc function. The next derivations are however valid with any function \( \varphi(t) \). This allows the use of functions of faster decay as was motivated in Section VI-C. Let us define
\[ a_\varphi(t) := \varphi(t) * \varphi(-t). \] (55)

**Proposition 7.1:**
\[
\langle g_i, g_j \rangle = \langle \varphi * h_i, \varphi * h_j \rangle = \langle h_i, a_{\varphi} * h_j \rangle = \int_{t_{i-1}}^{t_i} (a_{\varphi} * h_j)(t)dt.
\] (56)
where \( T_{i,j} := t_i - t_j \) as defined in (52) and
\[
f(t) = \int_0^1 (t - \tau) a_\varphi(\tau) d\tau.
\] (57)

**Proof:** We have \( \langle g_i, g_j \rangle = \langle \varphi * h_i, \varphi * h_j \rangle = \langle h_i, a_{\varphi} * h_j \rangle = \int_{t_{i-1}}^{t_i} (a_{\varphi} * h_j)(t)dt \). Next, \( (a_{\varphi} * h_j)(t) = \int_{t_{j-1}}^{t_j} a_\varphi(t-\tau) d\tau \) where \( h(\tau) := \int_0^1 a_{\varphi}(s) ds \). Thus,
\[
\langle g_i, g_j \rangle = \int_{t_{i-1}}^{t_i} h(t-t_{j-1}) dt - \int_{t_{i-1}}^{t_j} h(t-t_j) dt.
\] (58)

Defining \( f(t) := \int_0^1 h(\tau) d\tau \), we have for any \( k \),
\[
\int_{t_{i-1}}^{t_i} h(t-t_k) dt = \int_{t_{i-1}}^{t_i} h(t-t_{j-1}) dt - \int_{t_{i-1}}^{t_j} h(t-t_j) dt.
\] (59)

A slight numerical issue with (56) is that \( \lim|t|\to\infty f(t) = \infty \), while \( \lim_{|i-j|\to\infty} \langle g_i, g_j \rangle = 0 \). We show in Appendix B how this problem can be fixed.

B. Real-time computation of \( \hat{a}_k \)

To obtain the coefficients \( \hat{a}_k^\ell \), we need to express \( \langle g_{k-L}, g_{k-\ell} \rangle \) as required by (51). Let us define the coefficients
\[
f_k^\ell := f(T_{k-L,k-\ell}).
\] (58)
After verifying that \( f_k^{\ell-j'} = f(T_{k-L,j,k-\ell-j'}) \) and taking various values of \( j, j' \in \{0, 1\} \), one easily obtains from (54) and (56) that
\[
\hat{a}_k^\ell = f_{k-1}^{\ell-1} - f_{k-2}^{\ell-2} - f_k^{\ell} + f_{k-1}^{\ell-1}.
\] (59)
The values of \( f_k^\ell \) in (58) can be obtained from the time values \( T_{k-L,k-\ell} \) by table lookup.

A difficulty is the real-time transformation of the sequence of switching instants \( \{t_{i} 0 \leq i \leq N \} \) into the required values \( T_{k-L,k-\ell} \). In practice, the time encoder typically provides this sequence in the form of the successive differences \( T_k = t_k - t_{k-1} \). We show in Fig. 5 how the sequence \( T_k \) can be manipulated in real discrete time to eventually output the
requires values of $T_{k,i}$. The proposed technique is to consider the generalized sequence

$$T_k^n := T_{k,k-n} = t_k - t_{k-n}$$

and use the following recursive relations

$$T_k^n = T_{k-1}^{n-1} + T_k$$
$$T_k^n = T_{k-1}^{n-1} - T_{k-n}$$

(60, 61, 62)

easy to verify from (60). With (58) and (59), $\hat{a}_k^\ell$ can then be obtained from the lookup table by the successive operations

$$f_k^\ell = f(T_{k-1}^{L-\ell})$$
$$d_k^\ell = f_k^\ell - f_{k-1}^{\ell-1}$$
$$\hat{a}_k^\ell = d_{k-1}^{\ell-1} - d_k^\ell$$

The global system requires $7L+8$ adders. Note that it induces a delay of two discrete-time instants.

Assuming that the sequence $T_k$ is bounded, the argument $T_{k-L,k-\ell}$ to the function $f(\cdot)$ of (58) remains bounded. With time quantization, it can therefore only take a finite number of values, thus allowing a lookup table of finite size. In the experiment of Section VI-D, we recall that its size was evaluated to be less than 100 KB.

VIII. PRELIMINARY ANALYSIS OF DATA NOISE EFFECT

We had a glimpse at the behavior of the algorithm with some data noise in the experiment of Section VI-D and more specifically in Fig. VI-D. It would be desirable to get a little more analytical insight on the effect of noise on the estimates, especially given the tendency for sampling to generate ill-conditioned operators [27]. We propose some preliminary analysis of this on a simple version of the reconstruction iteration (22) where $(\lambda^{(n)})_{n\geq 0}$ is a constant sequence of vectors.

A. General objective

Assume that the acquired sample values are no longer $x_i$ from (2) but instead

$$\tilde{x}_i := x_i + z_i$$

(63)

where $z_i$ is some error value for each $i \in Z$. The reconstruction iteration then becomes

$$x^{(n+1)} = \tilde{R}_\lambda x^{(n)}$$

where

$$\tilde{R}_\lambda u := \lambda + \sum_{i \in Z} \tilde{r}_i(u) g_i$$

and

$$\tilde{r}_i(u) := \tilde{x}_i - \langle h_i, u \rangle$$

for all $u \in \mathcal{B}$. The estimate $x^{(n)}$ is expected to deviate by an amount that depends on the noise vector $z = (z_i)_{i \in Z}$. We are interested in the new evolution of the error signal

$$e^{(n)} := x^{(n)} - x^\dagger.$$  

B. Error derivation

Clearly, $\tilde{r}_i(u) = r_i(u) + z_i$. Therefore,

$$\tilde{R}_\lambda u = R_\lambda u + z$$

where $z := \sum_{i \in Z} z_i^\dagger g_i$ (64)

for all $u \in \mathcal{B}$. It then follows from (27) that

$$\tilde{R}_\lambda u - x^\dagger = M_\lambda (u - x^\dagger) + z$$

(65)

where $M_\lambda$ is defined in (26). By taking $u = x^{(n)}$ and $\lambda = \lambda^{(n)}$, we finally have

$$e^{(n+1)} = M_\lambda e^{(n)} + z^{(n)}, \quad n \geq 0. \quad (66)$$

Since $x^{(0)} = 0$, then $e^{(0)} = -x^\dagger$. Thus, both $e^{(n)}$ and $z^{(n)}$ belong to $S_g$. Obviously from (26), $M_\lambda$ maps $S_g$ into itself. Then $e^{(n)} \in S_g$ for all $n \geq 0$.

C. Constant relaxation vector $\lambda$

The analysis of (66) is difficult due to the variations of $\lambda^{(n)}$. To obtain some preliminary idea on the behavior of $e^{(n)}$, we propose to limit ourselves to the case where $\lambda^{(n)}$ is a constant vector $\lambda \in (0, 2)^N$, so that

$$e^{(n+1)} = M_\lambda e^{(n)} + z^{(n)}, \quad n \geq 0. \quad (67)$$

This includes as a particular case the iteration of (20) which corresponds to $\lambda = 1$. As was pointed in Section IV-G, the outstanding property of $M_\lambda$ that makes the analysis of (67) feasible is that it is self-adjoint. Indeed, one can easily...
check from (26) that \( \langle u, M_N v \rangle = \langle M_N u, v \rangle \). Since \( S_g \) is invariant under \( M_N \), \( S_g \) yields an orthonormal basis \( (\psi_i)_{i \in Z} \) of eigenvectors of \( M_N \) of real eigenvalues \( (\mu_i)_{i \in Z} \) (it can be shown that \( (g_i)_{i \in Z} \) are independent, so the dimension of \( S_g \) is \( N \)). In Appendix A, Lemma A.1 states that \( \| M_N v \| < \| v \| \) for all \( v \in S_g \setminus \{0\} \). Therefore \( |\mu_i| < 1 \) for all \( i \in Z \).

D. Semi-convergence analysis

The decomposition of \( e^{(n)} \) in the orthonormal basis \( (\psi_i)_{i \in Z} \) yields
\[
e^{(n)} = \sum_{i \in Z} e_i^{(n)} \psi_i \quad \text{where} \quad e_i^{(n)} := \langle \psi_i, e^{(n)} \rangle.
\]
Note that \( \langle \psi_i, M_N e^{(n)} \rangle = \langle M_N \psi_i, e^{(n)} \rangle = \mu_i \langle \psi_i, e^{(n)} \rangle \). By taking the inner-product of the members of (66) with \( \psi_i \), we then obtain that
\[
e_i^{(n+1)} = \mu_i e_i^{(n)} + \langle \psi_i, z \lambda \rangle
\]
for each \( i \). Since \( e^{(0)} = -x^\dagger \), then \( e_i^{(0)} = -\langle \psi_i, x^\dagger \rangle \). By induction with \( n \) and geometric series, it follows that
\[
e_i^{(n)} = -\mu_i^n \langle \psi_i, x^\dagger \rangle + (1-\mu_i^n) \langle \psi_i, z \lambda \rangle \frac{1}{1-\mu_i}.
\]
The first term gives the zero-noise component of the error and thus quantifies the intrinsic convergence behavior of the algorithm. The second term isolates the contribution of data noise. This type of error decomposition is typically performed in the semi-convergence analysis of an algorithm [28]. As expected, the first term always tends to 0 since \( |\mu_i| < 1 \). Meanwhile, the noise term tends to \( \langle \psi_i, z \lambda \rangle \) at infinite iteration. This tends to amplify the input noise component \( \langle \psi_i, z \lambda \rangle \) when \( \mu_i \) is close to 1, which happens when the sampling is badly conditioned. In the experimental condition of Fig. 2(e) however, we find numerically that \( |\mu_i| < 0.3 \) for all \( i \in Z \), among which less than 1% satisfy \( \mu_i > 0.17 \). This shows the good conditioning of the time encoding machine, and hence implies its good behavior with respect to noise.

IX. DISCUSSION

Beyond the proposal of practical solutions for signal reconstruction in TEM, the present article addressed some fundamental issue with nonuniform sampling in general. When a bandlimited signal is sampled uniformly, one obtains a discrete description from which the original signal can be easily recovered, and in which linear and time-invariant manipulations of the original signal can be achieved by means of discrete-time convolution. With nonuniform samples, a common reflex of discrete-time processing is to keep working with the space of Nyquist-rate samples (or a Fourier-isomorphic version of it). But one faces two new difficulties: (i) unsynchronized data flows between Nyquist-rate samples and non-uniform samples, (ii) necessity to invert linear transformations that are not time invariant. For the second aspect, one often relies on block-based matrix algebra, which artificially maps exact algebraic operations on distorted signal spaces while implying implementation complications.

In our reconstruction framework, pseudo-inversion is not only a natural approach to inverting a linear transformation, but it also sets a working signal space, namely \( S_g \), whose elements are rigorously described by discrete coefficients that are synchronized in rate with the nonuniform samples. Instead of block-based algebraic operations, we perform the sampling pseudo-inverse numerically, by iteration of POCS-based affine transformations in the subspace \( S_g \) of truly bandlimited signals. These continuous-time operations can be performed exactly in discrete-time by working on the signal expansion coefficients in \( S_g \). This is similar to Nyquist-rate DSP except for the loss of time invariance and orthogonality. The generating family of \( S_g \) being \( (g_i)_{i \in Z} \), the discrete-time equivalent transformations must involve the inner-products \( \langle \langle g_i, g_j \rangle \rangle_{i,j \in Z} \). These values can however be obtained using a single-variable lookup table. In analogy to traditional DSP but in absence of time-invariance, the final practical implementation of these discrete-time transformations takes the form of time-varying FIR filters. The bottleneck in the reconstruction accuracy is all in the sliding window truncations. This is an old issue of standard DSP when dealing with bandlimitation, but this is a virgin topic when the filters are both time-varying and iterated. The paper contains preliminary results that can potentially be improved with further research.

A special feature of the reconstruction technique proposed in this paper is the absence of multipliers in the iteration. In practice, the above mentioned time-varying FIR filters operate on sequences of signed powers of 2, thus involving only bit shifting. The multiplierless technique is in fact possibly be improved with further research.

APPENDIX

A. Proof of Theorem 4.2

Theorem 4.2 is mostly based on the following result.

Lemma A.1:

\[ \forall \lambda \in (0, 2)^N, \quad \forall v \in S_g \setminus \{0\}, \quad \| M_N v \| < \| v \|. \]

Proof: For all \( v \in B, M_N v = P_B Q_N v \) where
\[ Q_N v := v - \sum_{i \in Z} \lambda_i \langle \hat{h}_i, v \rangle \| \hat{h}_i \|^2 \hat{h}_i = v - \sum_{i \in Z} \lambda_i \langle \hat{h}_i, v \rangle \hat{h}_i \]
and \( \hat{h}_i := \hat{h}_i/\| \hat{h}_i \| \). So \( \| M_N v \| \leq \| Q_N v \| \). Let \( S_h \) be the linear span of \( \{\hat{h}_i\}_{i \in Z} \). It is easy to see that \( Q_N \) leaves \( S_h \) invariant and is identity in \( S_h^\perp \). Let \( v \in S_g \setminus \{0\} \). Writing the decomposition of \( v = u + w \) in \( S_h \oplus S_h^\perp \), one obtains
\[ \| v \|^2 = \| u \|^2 + \| w \|^2 \quad \text{and} \quad \| Q_N v \|^2 = \| Q_N u \|^2 + \| w \|^2. \]

As \( u \) yields the expansion \( u = \sum_{i \in Z} \langle \hat{h}_i, u \rangle \hat{h}_i \) by orthonormality of \( \{\hat{h}_i\}_{i \in Z} \), then \( Q_N \) due to (68) yields
\[ \| Q_N u \|^2 = \sum_{i \in Z} (1-\lambda_i)^2 \langle \hat{h}_i, u \rangle^2 \leq \lambda u \| u \|^2 \]

(69)
where \( m_\lambda := \max_{i \in Z} (1-\lambda_i)^2 < 1 \). For any \( i \in Z \), note that 
\[
(h_i, u) = (h_i, v) = (g_i, v) \quad \text{due to (13).}
\] Since \( v \in S_\delta \setminus \{0\} \), 
\( (g_i, v) \) must be nonzero for some \( i \in Z \). So \( u \neq 0 \). Then (69) implies that 
\[
\|Q_\lambda u\| < \|u\|,
\] and as a result, (68) implies that 
\[
\|Q_\lambda v\| < \|v\|.
\]

Let \( U \) be the unit sphere of \( S_\delta \). Since \( \|M_\lambda v\| \) is a continuous function of \((\lambda, v)\) and the set \( \mathcal{C} := \{\varepsilon, 2 - \varepsilon\} \times U \) is compact, the value \( \gamma_\epsilon := \sup_{(\lambda, v) \in \mathcal{C}} \|M_\lambda v\| \) is reached at some pair \((\lambda_0, v_0) \in C\). Lemma A.1 then implies that \( \gamma_\epsilon < \|v_0\| = 1 \). When \( \lambda \in \{\varepsilon, 2 - \varepsilon\} \) and \( v \in S_\delta \setminus \{0\} \), \( \|M_\lambda v\|/\|v\| = \|M_\lambda (v/\|v\|)\| \leq \gamma_\epsilon \), which implies (28).

### B. Growth control of function \( f(t) \) of (57)

**Proposition A.2**: For any distinct \( i, j \in Z \), the inner product 
\[
\langle g_i, g_j \rangle \equiv h(T_{i,j-1}) - h(T_{i-1,j-1}) - h(T_{i,j}) + h(T_{i-1,j})
\]
with any function of the type \( h(t) = f(t) - (\alpha t + \beta) \).

**Proof**: Let \( i \) and \( j \) be given integers in \( Z \) and let us write 
\[
(T_{i-1,j-1}, T_{i,j-1}, T_{i,j}, T_{i-1,j}) = (d_0, d_1, d_2, d_3)
\]
for convenience. When \( i \neq j \), it is easy to see that \( d_0, d_1, d_2, d_3 \) all have the sign of \( -i-j \) (including the possibility of a 0 value). As \( o_\psi(t) \) is an even function, it can be checked from (57) that \( f(t) \) is even as well. So is \( h(t) \). Without loss of generality, we can then assume that \( i > j \). In this case, 
\[
\langle g_i, g_j \rangle = f(d_0) - f(d_1) - f(d_2) + f(d_3) = h(d_0) - h(d_1) - h(d_2) + h(d_3) + \alpha(d_0 - d_1 - d_2 - d_3)
\]
where the last term is easily checked to be 0.

The growth of \( h(t) \) can be limited by taking \( \alpha = \int_0^\infty a_\psi(s)ds \) as \( f(t) \to \alpha t \) when \( t \) goes to infinity as mentioned in Section VII-A. With this value of \( \alpha \), it can be proved that there even exists \( \beta \) such that \( h(t) \) vanishes at infinity, at least when \( a_\psi(t) = O(t^{-\gamma}) \) for some \( \gamma > 2 \). This is also the case when \( \psi(t) \) is the sinc function \( \sin(\pi/t)/(\pi/t) \) with \( \alpha = \frac{1}{2} \) and \( \beta = -\frac{1}{\pi} \) as can be checked with Mathematica.

To calculate \( \langle g_i, g_j \rangle \), however, one needs to return to the original formula (56) which yields 
\[
\langle g_i, g_j \rangle = 2f(T_i) \] using the even symmetry of \( f(t) \) mentioned in the above proof. This has the drawback to require a separate lookup table for \( f(t) \). The length of this table however remains limited since \( T_i \) remains of the order of the Nyquist period. In the system of Fig. 5, this table would be specifically used to calculate the output coefficient \( \hat{\alpha}_k L \) as 
\[
\langle g_{k-2}, g_{k-2} \rangle = 2f(T_{k-2}).
\]

### References

[1] A. Lazar and L. T. Tóth, “Perfect recovery and sensitivity analysis of time encoded bandlimited signals,” IEEE Trans. Circ. and Syst.-I, vol. 51, pp. 2060–2073, Oct. 2004.

[2] R. Duffin and A. Schaeffer, “A class of nonharmonic Fourier series,” Transactions of the American Mathematical Society, vol. 72, pp. 341–366, Mar. 1952.

[3] J. L. Yen, “On nonuniform sampling of bandwidth-limited signals,” IRE Trans. Circ. Theory, vol. CT-3, pp. 251–257, Dec. 1956.

[4] H. G. Feichtinger and K. Gröchenig, “Theory and practice of irregular sampling,” in Wavelets: Mathematics and Applications (J. Benedetto, ed.), pp. 318–324, Boca Raton: CRC Press, 1994.

[5] F. Marvasti, Nonuniform Sampling: Theory and Practice. New York: Kluwer, 2001.

[6] M. Miskowicz, Event-Based Control and Signal Processing. Embedded Systems, CRC Press, 2018.

[7] D. Kościelnik and M. Miskowicz, “Time-to-digital converters based on event-driven successive charge redistribution: A theoretical approach,” Measurement, vol. 45, no. 10, pp. 2511 – 2528, 2012. Special Volume.

[8] A. A. Lazar, E. K. Simonyi, and L. T. Tóth, “Fast recovery algorithms for time encoded bandlimited signals,” in Proceedings. (ICASSP ’05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005., vol. 4, pp. iv/237–iv/240 Vol. 4, March 2005.

[9] A. A. Lazar, E. K. Simonyi, and L. T. Tóth, “An overcomplete stitching algorithm for time decoding machines,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 55, pp. 2619–2630, Oct 2008.

[10] G. T. Herman, A. Lent, and P. H. Lutz, “Iterative relaxation methods for image reconstruction,” in Proceedings of the 1975 Annual Conference, ACM ’75, (New York, NY, USA), pp. 169–174, ACM, 1975.

[11] P. L. Combettes, “The foundations of set theoretic estimation,” Proceedings of the IEEE, vol. 81, pp. 182–208, Feb 1993.

[12] H. H. Bauschke and J. M. Borwein, “On projection algorithms for solving convex feasibility problems,” SIAM Rev., vol. 38, no. 3, pp. 367–426, 1996.

[13] K. Adam, A. Scholefield, and M. Vetterli, “Sampling and reconstruction of bandlimited signals with multi-channel time encoding,” arXiv:1907.05657v1.

[14] K. Adam, A. Scholefield, and M. Vetterli, “Encoding and decoding mixed bandlimited signals using spiking integrate-and-fire neurons,” arXiv:1910.09413.

[15] A. Chandra and S. Chattopadhyay, “Design of hardware efficient FIR filter: A review of the state-of-the-art approaches,” Engineering Science and Technology, an International Journal, vol. 19, no. 1, pp. 212 – 226, 2016.

[16] D. Hand and M.-W. Chen, “A non-uniform sampling ADC architecture with embedded alias-free asynchronous filter,” in Global Communications Conference (GLOBECOM), 2012 IEEE, pp. 3707–3712, Dec 2012.

[17] H. Johansson and P. Löwenborg, “Reconstruction of nonuniformly sampled bandlimited signals by means of time-varying discrete-time FIR filters,” EURASIP J. Adv. Signal Process, vol. 2006, pp. 105–105, Jan. 2006.

[18] O. Christensen, Frames and bases. Applied and Numerical Harmonic Analysis, Birkhäuser Boston, Inc., Boston, MA, 2008. An introductory course.

[19] S. Kaczmarz, “Angenäherte auflösung von systemen linearer gleichungsn,” Bulletin de l’Académie des Sciences de Pologne, vol. A33, pp. 355–357, 1937.

[20] I. Halperin, “The product of projection operators,” Acta Scientiarum Mathematicarum (Sieged), vol. 23, no. 1, pp. 96–99, 1962.

[21] L. G. Gubin, B. T. Polyak, and E. V. Raik, “The method of projections and projection operators,” Numer. Algorithms, vol. 1, no. 3, pp. 259–274, 2006.

[22] N. T. Thao, “Sliding-window signal reconstruction for time encoding machine,” in 2016 Second International Conference on Event-based Control, Communication, and Signal Processing (EBCBSP), pp. 1–7, June 2016.

[23] N.T.Thao, “Event-based data acquisition and reconstruction - Mathematical background,” in Event-Based Control and Signal Processing (M. Miskowicz, ed.), pp. 379–407, CRC/Taylor and Francis, 2015.

[24] H. Choi and D. C. Munson, “Analysis and design of minimax-optimal interpolators,” IEEE Transactions on Signal Processing, vol. 46, pp. 1571–1579, Jun 1998.

[25] T. Eltving, P. C. Hansen, and T. Nikazad, “Semi-convergence properties of Kaczmarz’s method,” Inverse Problems, vol. 30, no. 5, pp. 055007, 16, 2014.