



Article

Signal Filtering Using Neuromorphic Measurements

Dorian Florescu ^{1,*}  and Daniel Coca ^{2,*} 

¹ Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, UK

² School of Engineering, Newcastle University, Newcastle upon Tyne NE1 7RU, UK

* Correspondence: d.florescu@imperial.ac.uk (D.F.); d.coca@sheffield.ac.uk (D.C.)

Abstract: Digital filtering is a fundamental technique in digital signal processing, which operates on a digital sequence without any information on how the sequence was generated. This paper proposes a methodology for designing the equivalent of digital filtering for neuromorphic samples, which are a low-power alternative to conventional digital samples. In the literature, filtering using neuromorphic samples is performed by filtering the reconstructed analog signal, which is required to belong to a predefined input space. We show that this requirement is not necessary, and introduce a new method for computing the neuromorphic samples of the filter output directly from the input samples, backed by theoretical guarantees. We show numerically that we can achieve a similar accuracy compared to that of the conventional method. However, given that we bypass the analog signal reconstruction step, our results show significantly reduced computation time for the proposed method and good performance even when signal recovery is not possible.

Keywords: event-driven sampling; time encoding machines; filter design



Citation: Florescu, D.; Coca, D. Signal Filtering Using Neuromorphic Measurements. *J. Low Power Electron. Appl.* **2023**, *13*, 63. <https://doi.org/10.3390/jlpea13040063>

Academic Editor: Costas Psychalinos

Received: 18 October 2023

Revised: 22 November 2023

Accepted: 30 November 2023

Published: 6 December 2023



Copyright: © 2023 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/>).

1. Introduction

In contemporary computing, the majority of tasks necessitate some form of digital signal processing [1]. With the escalating computational capabilities of digital processing systems, there is a concurrent surge in power consumption. This is further exacerbated by the rapid advancements in the field of artificial intelligence. Neuromorphic sampling, or time encoding, is an alternative to traditional digital encoding that transforms an analog signal into a sequence of low-power time-based pulses, often referred to as a spike train. Neuromorphic sampling draws its inspiration from neuroscience and introduces a paradigm shift by significantly reducing power consumption during encoding and transmission [2,3]. Despite these advantages, as of now, there exist no equivalents of digital signal processing operations tailored to neuromorphic sampling. This unexplored territory holds the promise of groundbreaking developments in low-power and efficient signal processing.

In this paper, we address the problem of filtering a signal using its neuromorphic measurements, thus extending the principle of digital signal filtering for the case of neuromorphic sampling. In posing this problem, we do not seek alternatives for spiking neural networks, but rather theoretically validated analytical approaches. Moreover, the proposed problem is not to replace existing conventional digital signal processing, but it is posed under the assumption that the communication protocol is emitting and receiving spike trains [4,5]. In this case, the proposed problem is to perform the mathematical operation of filtering on the analog signals encoded in these spike trains.

In the literature, the concept of spike train filtering predominantly refers to convolving the filter function with a sequence of Diracs centered in the spike times [6]. This is not an operation on the analog signal that resulted in those neuromorphic samples. Moreover, in the context of neuromorphic measurements generated from multidimensional signals, such as those generated by event cameras [7], filtering also refers to performing multidimensional spatial convolution [8,9]. The case of filtering the analog signal via its neuromorphic

measurements was proposed in [10], but the process also involves signal reconstruction. A filtering approach of neuromorphic signals without reconstruction was first studied in [11].

Therefore, conventionally, to process the signal underlying a sequence of spike measurements, the signal is recovered in the analog domain, followed by filtering and neuromorphic sampling. There are a number of drawbacks with this approach. First, this method does not exploit the power consumption advantage of neuromorphic measurements. Second, this approach is heavier computationally due to the complexity of signal reconstruction from neuromorphic samples [12,13]. Third, reconstruction is only possible if the input satisfies some restrictive smoothness or sparsity constraints. However, for conventional sampling, the process of digital filtering is independent of the characteristics of the signal that generated the measurements.

In this paper, we derive a direct mapping between the time encoded inputs and outputs of an analog filter. The proposed mapping forms the basis for a practical filtering algorithm of the underlying signal corresponding to some given neuromorphic measurements, without direct access to the signal. We introduce theoretical guarantees and error bounds for the measurements generated with the proposed algorithm. Through numerical simulations, we demonstrate the performance of our method in terms of speed, but also reduced restrictions on the input signal, in comparison with the existing conventional method.

This paper is structured as follows. Section 2 presents a brief review of the time encoding model used in this paper and associated input reconstruction methods. Section 3 introduces the proposed problem. Section 4 describes the proposed filtering method. Numerical results are presented in Section 5. Section 6 presents the concluding remarks.

2. Time Encoding

The time encoding machine (TEM) is a conceptualization of neuromorphic sampling that maps input $u(t)$ into sequence of samples $\{t_k\}_{k \in \mathbb{Z}}$. The particular TEM considered here is the integrate-and-fire (IF) TEM, which is inspired by neuroscience. Consequently, sequence $\{t_k\}_{k \in \mathbb{Z}}$ is called a spike train, where the spike refers to the firing of an action potential, representing the information transmission method in the mammalian cortex. Previously, the IF model was used for system identification of biological neurons [14,15] to perform machine learning tasks [16,17] but also for input reconstruction [13,18–21]. The IF model adds input $g(t)$ with bias parameter b , and subsequently integrates the output to generate a strictly increasing function $y(t)$. When $y(t)$ crosses threshold δ , the integrator is reset and the IF generates output spike time t_k . The IF TEM is described by the following equations:

$$\int_{t_k}^{t_{k+1}} u(s) ds = \delta - b(t_{k+1} - t_k), \quad k \in \mathbb{Z}_+^*. \quad (1)$$

Without reducing the generality, it is assumed that $t_0 = 0$. A common assumption is that the input is bounded by $c \in \mathbb{R}_+$, such that $|u(t)| \leq c < b$. This bound enables derivation of the following density guarantees [2]:

$$\frac{\delta}{b+c} \leq t_{k+1} - t_k \leq \frac{\delta}{b-c}. \quad (2)$$

Signal $u(t)$ is, in the general case, not recoverable from $\{t_k\}_{k \in \mathbb{Z}}$. To ensure it can be reconstructed, we require imposing restrictive assumptions. A common assumption is that $u(t)$ belongs to PW_Ω , the space of functions bandlimited to Ω rad/s that are also square integrable, i.e., $u \in L^2(\mathbb{R})$. If this assumption is satisfied, then $u(t)$ can be recovered from $\{t_k\}_{k \in \mathbb{Z}}$ if

$$\frac{\delta}{b-c} < \frac{\pi}{\Omega}. \quad (3)$$

To offer an intuition on the recovery condition and its link to the Nyquist rate, we note that TEM Equation (1) can be re-written as

$$\langle u, \mathbb{1}_{[t_k, t_{k+1}]} \rangle_{L^2} = \delta - b(t_{k+1} - t_k), \quad (4)$$

where $\langle \cdot, \cdot \rangle_{L^2}$ denotes the inner product in Hilbert space $L^2(\mathbb{R})$ of square integrable functions and $\mathbb{1}_{[t_k, t_{k+1}]}$ is the characteristic function of $[t_k, t_{k+1}]$. We note that although $\mathbb{1}_{[t_k, t_{k+1}]} \in L^2(\mathbb{R})$, it is not bandlimited, i.e., $\mathbb{1}_{[t_k, t_{k+1}]} \notin PW_\Omega$. However, due to the properties of the inner product [22],

$$\langle u, \mathbb{1}_{[t_k, t_{k+1}]} \rangle_{L^2} = \langle u, \varphi_k \rangle_{L^2} = \delta - b(t_{k+1} - t_k), \quad (5)$$

where $\varphi_k(t) \triangleq \frac{\sin(\Omega \cdot)}{\pi \cdot} * \mathbb{1}_{[t_k, t_{k+1}]}(t)$ denotes the projection of $\mathbb{1}_{[t_k, t_{k+1}]}$ in PW_Ω . In the case of the Nyquist rate condition, the uniform samples can be described as

$$u(kT) = \left\langle u, \frac{\sin(\Omega(\cdot - kT))}{\pi(\cdot - kT)} \right\rangle_{L^2}.$$

The Nyquist rate criterion [23] ensures that $\{u(kT)\}_{k \in \mathbb{Z}}$ uniquely identify $u(t)$ if $T < \frac{\pi}{\Omega}$, which is guaranteed by the fact that functions $\left\{ \frac{\sin(\Omega(\cdot - kT))}{\pi(\cdot - kT)} \right\}_{k \in \mathbb{Z}}$ form a basis in PW_Ω . The same is not true in the case of time encoding, which is a form of nonuniform sampling [24–26], where $\{\varphi_k\}_{k \in \mathbb{Z}}$ do not form a basis. They can, however, form the more general concept of frame [27], which guarantees that $u(t)$ is uniquely determined by $\langle u, \varphi_k \rangle_{L^2}$ if sequence $\{t_k\}_{k \in \mathbb{Z}}$ is dense enough. Via (2), we can use $\frac{\delta}{b-c}$ as a measure of density of sequence $\{t_k\}_{k \in \mathbb{Z}}$, yielding Nyquist-like criterion (3).

Input $u(t)$ is then recovered from $\{t_k\}_{k \in \mathbb{Z}}$ as

$$u(t) = \sum_{m \in \mathbb{Z}} \tilde{c}_m \cdot \frac{\sin \Omega(t - s_k)}{\pi(t - s_k)},$$

where $s_k = \frac{t_k + t_{k+1}}{2}$ are the midpoints of intervals $[t_k, t_{k+1}]$ and $\tilde{c}_k$ are the solution in the least square sense of the following system [22]:

$$\sum_{n \in \mathbb{Z}} c_n \int_{t_k}^{t_{k+1}} \frac{\sin \Omega(t - s_k)}{\pi(t - s_k)} dt = \delta - b(t_{k+1} - t_k). \quad (6)$$

Unlike uniform sampling, the input recovery for an IF TEM is much more complex computationally because, for each new input, functions $\frac{\sin \Omega(t - s_k)}{\pi(t - s_k)}$ have to be computed and System (6) needs to be solved. This becomes very demanding computationally for long sequences $\{t_k\}$. Alternative recovery approaches are based on optimizing a smoothness-based criterion instead of aiming to uniquely recover the input [28,29]. Moreover, the problem of input recovery was shown to be equivalent to that of system identification of a filter in series with an IF TEM in the case of linear [30,31] and nonlinear filters [32].

The methods presented so far are assuming the input is bandlimited. Further generalizations were introduced for the case where $u(t)$ is a function in a shift-invariant space [13,19] or in a space with finite rate of innovation [20,21]. However, if $u(t)$ does not belong to one of the classes above, or if it is bandlimited and does not satisfy (3), then the conventional theory does not allow any processing of signal $u(t)$ via its samples t_k . The same is not true for conventional digital signals, which can be processed even when they are not sampled at the Nyquist rate. We show that some types of processing such as filtering are still doable even when (3) is not true.

3. Problem Statement

Here, we formulate the proposed signal filtering problem as follows. We assume the neuron input is continuous, i.e., $u \in C(\mathbb{R})$. To satisfy the neuron encoding requirement in Section 2, we assume the input is bounded, such that $|u(t)| \leq c < b, \forall t \in \mathbb{R}$. Furthermore, we assume that the input is absolutely integrable $u \in L^1(\mathbb{R})$ and square integrable $u \in L^2(\mathbb{R})$. Following the same idea as in digital filtering, we do not impose any general conditions on the bandwidth, smoothness, or sparsity of the analog signal in order to compute the filter output.

The filter is assumed to be linear, with impulse response $g(t)$ that is continuous, $g \in C(\mathbb{R})$, and absolutely integrable, $g \in L^1(\mathbb{R})$. The output of the filter then satisfies

$$y(t) = \int_{-\infty}^{\infty} u(\tau)g(t-\tau)d\tau, \quad (7)$$

$$|y(t)| \leq \int_{-\infty}^{\infty} |u(\tau)| \cdot |g(t-\tau)|d\tau \leq c\|g\|_{L^1} \leq c,$$

where the last inequality assumes that $\|g\|_{L^1} \leq 1$, which is introduced to ensure that $|y(t)| \leq c$, which in turn allows $y(t)$ sampling by the same neuron. According to the properties of the convolution operator, we also have $y \in L^2(\mathbb{R}) \cap L^1(\mathbb{R})$, where $L^2(\mathbb{R})$ denotes the space of square integrable functions.

We let $\{t_k^u\}_{k \in \mathbb{Z}}$ and $\{t_k^y\}_{k \in \mathbb{Z}}$ be the neuromorphic samples of signals u and y , respectively, computed using an IF neuron with parameters δ, b . The proposed problem is to compute t_k^y knowing t_k^u , sampling parameters δ, b and filter $g(t)$. This problem, illustrated in Figure 1, is inspired by digital signal processing, where a digital filter is applied directly to the samples of a signal. The conventional way to address this problem would be to recover $u(t)$ from t_k^u , apply filter $g(t)$ in the analog domain, and subsequently sample output $y(t)$ with the same IF model to obtain t_k^y . We refer to this as the indirect method for filtering. The first step of recovery, however, is not possible unless we impose some further restrictive conditions on $u(t)$ such as being bandlimited [22], shift-invariant [13,19], or having a finite rate of innovation [18,20]. Therefore, the proposed problem is not solvable in its full generality using conventional approaches.

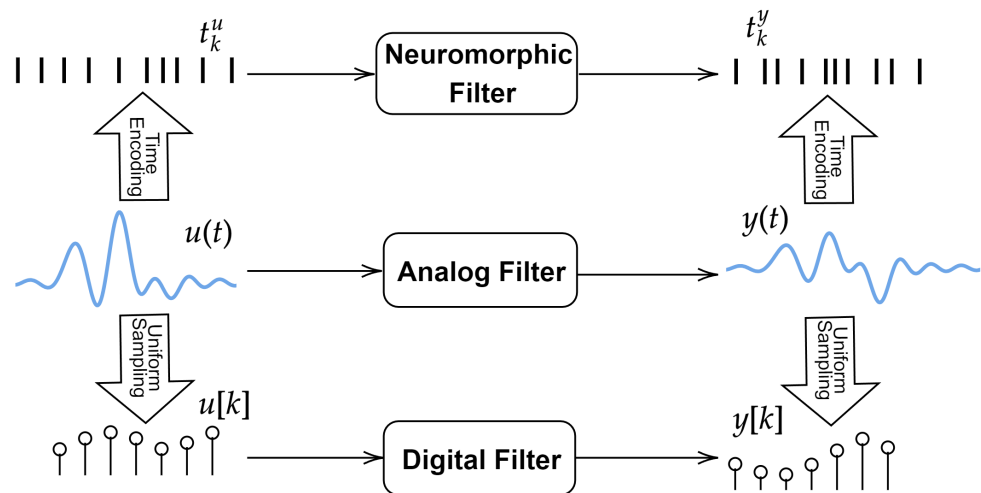


Figure 1. Approximating the output of an analog filter using input measurements. In the case of uniform sampling measurements (**bottom**), digital filters are well known and have been studied exhaustively. Mapping the filter input to the output was not studied extensively for neuromorphic measurements. The conventional approach is to reconstruct the input and compute filtering in the analog domain, which is time consuming and not always possible.

However, if we replace neuromorphic sampling by conventional uniform sampling, this problem leads to the widely used operation of digital filtering. The operation itself does not require any special conditions on the analog signal that generated these samples. Therefore, an equivalent of this solution for the case of neuromorphic sampling is highly desirable.

4. The Proposed Neuromorphic Direct Filtering Method

In this section, we describe the proposed direct filtering method. To compute t_k^y from t_k^u we need to create an analytical link between the integrals of the underlying analog signals $u(t)$ and $y(t)$ due to the integral operators in (1) and (7). To this end, we define the following auxiliary functions:

$$U(t) \triangleq \int_0^t u(\tau) d\tau, \quad Y(t) \triangleq \int_0^t y(\tau) d\tau. \quad (8)$$

We note that U satisfies $|U(t)| \leq \|u\|_{L^1}$. Using Young's convolution inequality, we obtain $|Y(t)| \leq \|y\|_{L^1} \leq \|u\|_{L^1} \cdot \|g\|_{L^1}$. Using these functions, we derive the equivalent of t -transform Equations (1) as

$$U(t_k^u) = k\delta - bt_k^u, \quad Y(t_k^y) = k\delta - bt_k^y, \quad \forall k \in \mathbb{Z}. \quad (9)$$

Assuming we know $Y(t)$, the target spike train t_k^y satisfies $f_k(t_k^y) = t_k^y$, where

$$f_k(t) \triangleq \frac{1}{b}(k\delta - Y(t)). \quad (10)$$

The following result shows that t_k^y can be uniquely computed using $f_k(t)$.

Lemma 1 (Exact Output Samples Computation). *Function $f_k(t)$ has a unique fixed point $t = t_k^y$. Furthermore, we let $t_{k,m}^y$ be computed recursively such that $t_{k,0}^y \in \mathbb{R}$ is arbitrary and*

$$t_{k,m+1}^y = f_k(t_{k,m}^y). \quad (11)$$

Then, $\lim_{m \rightarrow \infty} t_{k,m}^y = t_k^y$ and $|t_{k,m}^y - t_k^y| < |t_k^y - t_{k,0}^y| \left(\frac{c}{b}\right)^m$.

Proof. We assume, by contradiction, that $\exists \bar{t} \neq t_k^y$ such that $f_k(\bar{t}) = \bar{t}$. It follows that

$$b\bar{t} + Y(\bar{t}) = \int_0^{\bar{t}} (y(\tau) + b) d\tau = k\delta.$$

On the other hand, we know that $|y(t)| \leq c < b$ due to (7), and thus $\int_0^{\bar{t}} (y(\tau) + b) d\tau$ is a strictly increasing function of $\bar{t}$, which ensures that $\int_0^{\bar{t}} (y(\tau) + b) d\tau = k\delta$ has a unique solution. Using (9), we obtain $\bar{t} = t_k^y$, which invalidates our initial assumption and proves uniqueness.

From (9), it follows that $\forall k \in \mathbb{Z}, f_k(t_k^y) = t_k^y$. The following holds:

$$|f_k'(t)| = \left| -\frac{y(t)}{b} \right| \leq \frac{c}{b},$$

and thus

$$\forall \zeta \geq 0, \forall t \in [t_k^y - \zeta, t_k^y + \zeta], f_k(t) \in [t_k^y - \zeta \frac{c}{b}, t_k^y + \zeta \frac{c}{b}]. \quad (12)$$

We let $\zeta = |t_k^y - t_{k,0}^y|$ and $t = t_{k,0}^y$. It follows that

$$|f_k(t_{k,0}^y) - t_k^y| = |t_{k,1}^y - t_k^y| \leq |t_k^y - t_{k,0}^y| \frac{c}{b}. \quad (13)$$

Similarly, by choosing $\zeta = |t_k^y - t_{k,1}^y|$ and $t = t_{k,1}^y$, we obtain

$$|t_k^y - t_{k,2}^y| \leq |t_k^y - t_{k,1}^y| \frac{c}{b} \leq |t_k^y - t_{k,0}^y| \left(\frac{c}{b}\right)^2,$$

and the process continues recursively, which completes the proof via $c < b$. $\square$

Therefore, t_k^y can be computed by solving the fixed-point equation $f_k(t) = t$. This equation requires knowing $Y(t)$, which satisfies

$$\begin{aligned} Y(t) &= \int_0^t \int_{-\infty}^{\infty} g(\tau) u(s - \tau) d\tau ds = \int_{-\infty}^{\infty} g(\tau) \int_0^t u(s - \tau) ds d\tau \\ &= \int_{-\infty}^{\infty} g(\tau) [U(t - \tau) - U(-\tau)] d\tau = \int_{-\infty}^{\infty} U(\tau) [g(t - \tau) - g(-\tau)] d\tau, \end{aligned}$$

where the last equality uses the variable change $\tau \rightarrow t - \tau$. In reality, however, $Y(t)$ is unknown, since it could only be precisely computed using $U(t)$. Given that we only know t_k^u and do not impose any smoothness or sparsity conditions on $u(t)$, we do not have access to $U(t)$, but only to its samples $U(t_k^u)$ via (9). In the following, we show that $Y(t)$, $f_k(t)$ and subsequently t_k^y can be estimated using a piecewise constant approximation of $U(t)$ at points t_k^u . We let $\tilde{f}_k : \mathbb{R} \rightarrow \mathbb{R}$ be defined by

$$\tilde{f}_k(t) = \frac{1}{b} (k\delta - \tilde{Y}(t)), \quad (14)$$

where

$$\tilde{Y}(t) = \int_{-\infty}^{\infty} I_1 U(\tau) [g(t - \tau) - g(-\tau)] d\tau, \quad (15)$$

where $I_1 U(t)$ is the piecewise constant interpolant to U at points $\{t_k^u\}_{k \in \mathbb{Z}}$, such that $I_1 U(t) = U(t_k^u)$ for $t \in [t_k^u, t_{k+1}^u)$. The next proposition derives some properties of $\tilde{Y}(t)$.

Proposition 1. *Function $\tilde{Y}(t)$ is continuous and satisfies*

$$\tilde{Y}(t) = \sum_{k \in \mathbb{Z}} U(t_k^u) (G(t - t_k^u) + G(-t_k^u)) - U(t_k^u) (G(t - t_{k+1}^u) - G(-t_{k+1}^u)), \quad (16)$$

where $G(t) \triangleq \int_0^t g(s) ds$.

Proof. Using (15), we obtain

$$\begin{aligned} \tilde{Y}(t) &= \sum_{k \in \mathbb{Z}} \int_{t_k^u}^{t_{k+1}^u} I_1 U(\tau) [g(t - \tau) - g(-\tau)] d\tau \\ &= \sum_{k \in \mathbb{Z}} U(t_k^u) \left[\int_{t_k^u}^{t_{k+1}^u} g(t - \tau) d\tau - \int_{t_k^u}^{t_{k+1}^u} g(-\tau) d\tau \right] \\ &= \sum_{k \in \mathbb{Z}} U(t_k^u) \left[\int_{t - t_{k+1}^u}^{t - t_k^u} g(\tau) d\tau - \int_{-t_{k+1}^u}^{-t_k^u} g(\tau) d\tau \right], \end{aligned} \quad (17)$$

which proves (16). It follows that $\tilde{Y}(t)$ is a linear combination of continuous functions; thus, it is itself continuous. $\square$

Proposition 1 shows that, unlike $Y(t)$ and $f_k(t)$, function $\tilde{Y}(t)$ and, consequently, also $\tilde{f}_k(t)$, are fully known from the IF parameters and input samples t_k^u . The remaining challenge is to show that the fixed point equation $\tilde{f}_k(t) = t$ can be solved and to provide an error bound for estimating t_k^y . This challenge is addressed rigorously in the next theorem.

Moreover, the result allows computing recursively a sequence of estimations $\tilde{t}_{k,m}^y$ that converges to a vicinity of t_k^y .

Theorem 1 (Estimating Output Samples from Input Samples). *We let $u(t)$ be a signal satisfying $u \in L^1(\mathbb{R}) \cap L^2(\mathbb{R}) \cap C(\mathbb{R})$, $|u(t)| \leq c < b$. Furthermore, we let $g(t)$ be the impulse response of a filter satisfying $g \in L^1(\mathbb{R}) \cap C(\mathbb{R})$, $\|g\|_{L^1} \leq 1$, and let $y(t)$ be the filter output in response to input $u(t)$. Signals $u(t)$ and $y(t)$, sampled with an IF neuron with parameters δ, b , generate values $\{t_k^u\}_{k \in \mathbb{Z}}$ and $\{t_k^y\}_{k \in \mathbb{Z}}$, respectively. Then, the following hold true:*

(a) $\forall k \in \mathbb{Z}, \exists \tilde{t}_k^y \in \left[t_k^y - \frac{2\delta c}{(b-c)^2}, t_k^y + \frac{2\delta c}{(b-c)^2} \right]$ such that $\tilde{f}_k(\tilde{t}_k^y) = \tilde{t}_k^y$, where $\tilde{f}_k(t)$ satisfies (14).

(b) We let $\tilde{t}_{k,m}^y$ be a sequence defined recursively as $\tilde{t}_{k,m+1}^y = \tilde{f}_k(\tilde{t}_{k,m}^y)$, where $\tilde{t}_{k,0}^y \in \mathbb{R}$. Then,

$$|\tilde{t}_{k,m}^y - t_k^y| \leq |t_k^y - \tilde{t}_{k,0}^y| \cdot \left(\frac{c}{b}\right)^m + \frac{2\delta c}{b(b-c)} \cdot \sum_{i=1}^m \left(\frac{c}{b}\right)^{i-1}. \quad (18)$$

(c) For $\tilde{t}_{k,m}^y$ defined above, $\exists m_0 \in \mathbb{Z}$ such that $\tilde{t}_{k,m}^y \in \left[t_k^y - \frac{2\delta c}{(b-c)^2}, t_k^y + \frac{2\delta c}{(b-c)^2} \right], \forall m > m_0$.

Proof. (a) Function $\tilde{Y}$ satisfies

$$\begin{aligned} |Y(t) - \tilde{Y}(t)| &\leq \int_{-\infty}^{\infty} |U(\tau) - I_1 U(\tau)| \cdot |g(t - \tau) - g(-\tau)| d\tau \\ &\leq \Delta \cdot \sup_{\tau \in \mathbb{R}} |U'(\tau)| \cdot \int_{-\infty}^{\infty} |g(t - \tau) - g(-\tau)| d\tau \leq E, \end{aligned} \quad (19)$$

where $E = 2\Delta c \|g\|_{L^1}$, $\Delta = \sup_{k \in \mathbb{Z}} (t_{k+1}^u - t_k^u)$. From (12), the following holds:

$$\forall \zeta \geq 0, \forall t \in \left[t_k^y - \zeta, t_k^y + \zeta \right], f_k(t) \in \left[t_k^y - \zeta \frac{c}{b}, t_k^y + \zeta \frac{c}{b} \right]. \quad (20)$$

Using $|f_k(t) - \tilde{f}_k(t)| \leq \frac{E}{b}, \forall t \in \mathbb{R}$,

$$\forall \zeta \geq 0, \forall t \in \left[t_k^y - \zeta, t_k^y + \zeta \right], \tilde{f}_k(t) \in \left[t_k^y - \zeta \frac{c}{b} - \frac{E}{b}, t_k^y + \zeta \frac{c}{b} + \frac{E}{b} \right]. \quad (21)$$

Unlike in the case of Lemma 1, in this case, applying $\tilde{f}_k(t)$ recursively does not guarantee the exact computation of t_k^y . However, we observe that by picking $\zeta = \frac{E}{b-c}$, we obtain identical intervals for t and $\tilde{f}_k(t)$:

$$\forall t \in \left[t_k^y - \frac{E}{b-c}, t_k^y + \frac{E}{b-c} \right], \tilde{f}_k(t) \in \left[t_k^y - \frac{E}{b-c}, t_k^y + \frac{E}{b-c} \right]. \quad (22)$$

This observation is very useful as it enables applying Brouwer's fixed-point theorem which states that for any continuous function $f : \mathbb{S} \rightarrow \mathbb{S}$ where $\mathbb{S}$ is a nonempty compact convex set, there is a point t_0 such that $f(t_0) = t_0$. Given that $\tilde{Y}(t)$ is continuous due to Proposition 1, it follows that $\tilde{f}_k$ is also continuous. By applying Brouwer's fixed point theorem for $f(t) = \tilde{f}_k(t)$ and $\mathbb{S} = \left[t_k^y - \frac{E}{b-c}, t_k^y + \frac{E}{b-c} \right]$, it follows that $f_k(t)$ has a fixed point in $\mathbb{S}$. We recall that $E = 2\Delta c \|g\|_{L^1}$, which, using (2), leads to

$$E \leq \frac{2\delta c}{b-c} \|g\|_{L^1} \leq \frac{2\delta c}{b-c}. \quad (23)$$

It follows that $\mathbb{S} \subseteq \left[t_k^y - \frac{2\delta c}{(b-c)^2}, t_k^y + \frac{2\delta c}{(b-c)^2} \right]$, which yields the required result.

(b) We approach this proof using mathematical induction. We select $t = \tilde{t}_{k,0}^y$ in (21). Using (23), it follows that

$$\left| \tilde{f}_k(\tilde{t}_{k,0}^y) - t_k^y \right| \leq \zeta \frac{c}{b} + \frac{2\delta c}{b(b-c)}, \quad \text{s.t.} \quad \tilde{t}_{k,0}^y \in \left[t_k^y - \zeta, t_k^y + \zeta \right]. \quad (24)$$

We note that $\tilde{t}_{k,0}^y \in \left[t_k^y - \zeta, t_k^y + \zeta \right]$ is always true for $\zeta = \left| t_k^y - \tilde{t}_{k,0}^y \right|$, which yields

$$\left| \tilde{f}_k(\tilde{t}_{k,0}^y) - t_k^y \right| \leq \left| t_k^y - \tilde{t}_{k,0}^y \right| \frac{c}{b} + \frac{2\delta c}{b(b-c)}. \quad (25)$$

This demonstrates that (18) is true for $m = 1$. To finalize the induction, we assume (18) to be true, and show it is true for $m + 1$ as follows:

$$\left| \tilde{t}_{k,m+1}^y - t_k^y \right| = \left| \tilde{f}_k(\tilde{t}_{k,m}^y) - t_k^y \right| \leq \zeta \frac{c}{b} + \frac{2\delta c}{b(b-c)}. \quad (26)$$

Finally, as before, we use the fact that $\zeta = \left| t_k^y - \tilde{t}_{k,m}^y \right|$ guarantees $\tilde{t}_{k,m}^y \in \left[t_k^y - \zeta, t_k^y + \zeta \right]$. We also use the fact that ζ is bounded by (18), which, when substituted in (26), leads to the desired result via (23).

(c) Equation (18) can be expanded into

$$\tilde{t}_{k,m}^y \leq t_k^y + \left| t_k^y - \tilde{t}_{k,0}^y \right| \left(\frac{c}{b} \right)^m + \frac{2\delta c}{(b-c)^2} \cdot \left(1 - \left(\frac{c}{b} \right)^m \right), \quad (27)$$

$$\tilde{t}_{k,m}^y \geq t_k^y - \left| t_k^y - \tilde{t}_{k,0}^y \right| \left(\frac{c}{b} \right)^m - \frac{2\delta c}{(b-c)^2} \cdot \left(1 - \left(\frac{c}{b} \right)^m \right). \quad (28)$$

The required result follows from $\lim_{m \rightarrow \infty} \left(\frac{c}{b} \right)^m = 0$ via (23). $\square$

Theorem 1 shows that one can construct sequence $\tilde{t}_{k,m}^y$ that approximates t_k^y with error $\frac{2\delta c}{(b-c)^2}$. We note that this error is dependent only on neuron parameters, and thus can be made arbitrarily small by changing the IF model. In practice, we use a finite sequence of input measurements $\{t_l^u\}_{l=0,\dots,L}$ to approximate output samples $\{t_k^y\}_{k=0,\dots,K}$ that satisfy $t_0^u \leq t_k^y \leq t_L^u, \forall k = 0, \dots, K$. The proposed direct filtering method is summarized in Algorithm 1.

We note that, in practice, convergence was achieved in Algorithm 1 for $M \leq 4$ iterations in all examples we evaluated. Moreover, we note that, in the proposed manuscript, computing $\tilde{t}_{k+1}^y$ uses $\tilde{t}_k^y$ as an initial condition. When new input data samples become available, Algorithm 1 incorporates them in computing new output data samples, but does not need to re-compute the output samples that are already known. In the next section, we numerically evaluate the proposed algorithm.

Algorithm 1 Computing the neuromorphic output data samples via the proposed method.

Data: $\{t_k^u\}, g(t), \delta, b;$

Result: $\{\tilde{t}_k^y\};$

Step 1. Set $k = 1$ and $\tilde{t}_0^y = 0$. While $\tilde{t}_{k-1}^y < t_L^u$,

Step 1a. $\tilde{t}_{k,0}^y = \tilde{t}_{k-1}^y;$

Step 1b. Compute $\tilde{t}_{k,m+1}^y = \tilde{f}_k(\tilde{t}_{k,m}^y)$ for $m = 1, \dots, M$, where $\tilde{f}_k(t) = \frac{1}{b}(k\delta - \tilde{Y}(t))$,

$$\tilde{Y}(t) = \sum_{l=0}^L U(t_l^u)(G(t - t_l^u) - G(-t_l^u)) - U(t_l^u)(G(t - t_{l+1}^u) + G(-t_{l+1}^u)), \quad (29)$$

and $G(t) = \int_0^t g(\tau)d\tau, U(t_l^u) = l\delta - bt_l^u, l = 0, \dots, L;$

Step 1d. $\tilde{t}_k^y = \tilde{t}_{k,M}^y;$

Step 1e. $k = k + 1;$

Step 2. Set $K = k - 2$.

5. Numerical Study

In order to evaluate the performance of the proposed method, we define the following approximation errors:

1. Output inter-spike time error between the output of the time-encoded filter and the time-encoded output of the conventional filter:

$$e_j^t(k) = (t_{k+1}^y - t_k^y) - (\tilde{t}_{k+1}^{y_j} - \tilde{t}_k^{y_j}), \quad \forall k = 1, \dots, K,$$

where $\tilde{t}_k^{y_j}$ is the output spike train prediction with Algorithm 1 ($j = 1$) and the conventional method, involving the reconstruction of $u(t)$ ($j = 2$).

2. Output error between the decoded output of the time-encoded filter and the output of the conventional digital filter:

$$e_j^y(t) = y(t) - \tilde{y}_j(t), \quad \forall t \in [0, 2], j = 1, 2.$$

We compute the errors as percentages of true quantities $(t_{k+1}^y - t_k^y)$ and $y(t)$, respectively, by defining the following normalized error metrics:

$$ERR_j^t = 100 \cdot \frac{\|e_j^t\|_{\ell^2}}{\sqrt{\sum_{k=1}^K (t_{k+1}^{y_j} - t_k^{y_j})^2}} (\%), \quad ERR_j^y = 100 \cdot \frac{\|e_j^y\|_{L^2}}{\|y\|_{L^2}} (\%).$$

Furthermore, we denote by Time₁ and Time₂ the computation times of the output spike train with the proposed and indirect methods, respectively. In our numerical examples, we discretize the time using a simulation time step of $\varepsilon = 1 \times 10^{-4}$ s. The parameters used, errors and computing times are summarized in Table 1.

Table 1. The parameters used in our numerical simulations section are summarized below.

Example	Input	Filter	δ	ERR ₁ ^t (%)	ERR ₂ ^t (%)	ERR ₁ ^y (%)	ERR ₂ ^y (%)	Time ₁ (s)	Time ₂ (s)
5.1	Low pass	Low pass	0.05	0.49	0.75	0.82	3.35	0.62	2.93
5.2	Uniform Noise	Low pass	0.1	0.6	1.06	25.2	26.6	0.32	0.81
5.3	Uniform Noise	Low pass	0–0.1	0.5–1.3	1–3	22–25	24–32	0.1–1	1–20
5.4	Uniform Noise	Wavelet	0.04	2.85	3.2	N/A	N/A	0.2	3

5.1. Low-Pass Filtering of a Bandlimited Signal

The selected filter has the following transfer function:

$$G(s) = \frac{\omega_c}{s + \omega_c}, \quad (30)$$

where $\omega_c = 10$ rad/s. Furthermore, the input is generated as

$$u(t) = \sum_{k=1}^N c_k \cdot \frac{\sin(\Omega(t - k\pi/\Omega))}{\Omega(t - k\pi/\Omega)}, \quad t \in [0, 2], \quad (31)$$

where $N = 191$ and c_k are random coefficients drawn from the uniform distribution on $(-1, 1)$, and $\Omega = 100$ rad/2. We select $c_i = 0, i \in \{1, 2, 3, N-2, N-1, N\}$ in order to avoid boundary errors for input recovery. The input and output of system $G(s)$ are sampled with an integrate and fire neuron with parameters $\delta = 0.05, b = 6$. Signals $u(t), y(t)$ and corresponding spike times are depicted in Figure 2a,b, respectively.

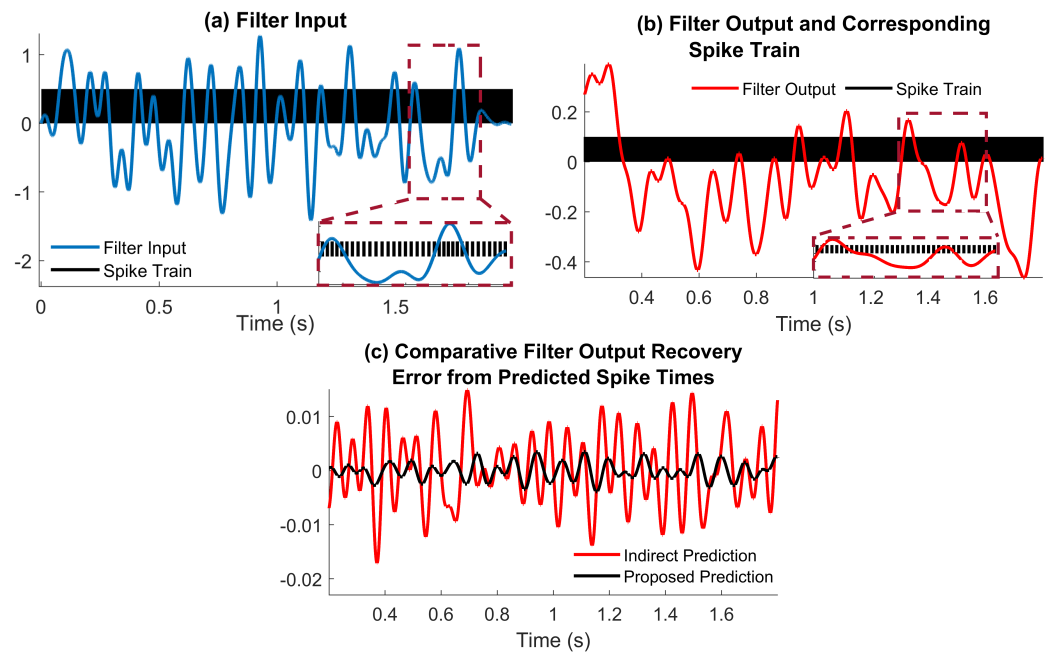


Figure 2. Output spike train prediction for a bandlimited input. (a) Filter input and the corresponding spike train. (b) Filter output and the corresponding spike train. (c) Recovery errors $e_2^y(t)$ and $e_1^y(t)$ with the indirect and proposed method, respectively.

We estimate spike times t_k^y in two different ways, via the indirect method presented in Section 3 and the proposed direct method in Section 4. The results are depicted in Figure 3, and the recovery errors are $ERR_1^t = 0.49\%$ for the proposed method and $ERR_2^t = 0.75\%$ for the indirect method. Furthermore, the filter output errors are $ERR_1^y = 0.82\%$ and $ERR_2^y = 3.35\%$. However, the computing times are 0.62 s for the proposed method and 2.93 s for the indirect method. We note that a major contribution in the error induced by the conventional method is due to boundary errors, which are a known artefact of reconstruction [12,33].

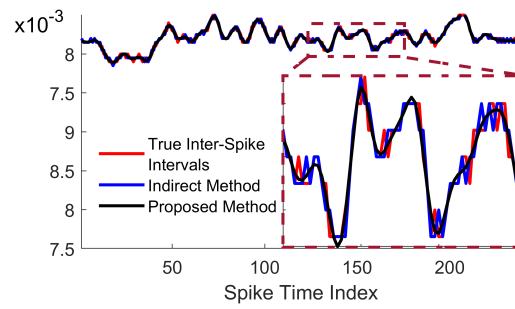


Figure 3. Spike train prediction comparative performance: the true inter-spike intervals, interposed on the predicted inter-spike intervals with the indirect and proposed methods, respectively.

5.2. Low-Pass Filtering of Uniform White Noise

To show that the proposed method has no limitations in terms of input bandwidth, as indicated by the bounds in Theorem 1, here, we choose $u(t)$ to be drawn from the random uniform distribution on $[-1.5, 1.5]$. Given that in this case, filter $G(s)$ reduces significantly the noise amplitude, we amplify the output by 10 using the new filter, $G(s) = 10 \cdot \frac{\omega_c}{s + \omega_c}$, where $\omega_c = 10$ rad/s. The neuron parameters are $\delta = 0.1, b = 6$. The input and output signals, as well as the output recovered with both methods, are presented in Figure 4a,b. The proposed prediction method does not need any knowledge of the input bandwidth. As an alternative, we reconstruct the input with the conventional indirect method, where the recovery bandwidth is $\Omega = 100$ rad/s. The value of Ω is well above the cutoff frequency of filter ω_c , which leads to accurate recovery of the filter output. However, random input $u(t)$ cannot be recovered for any $\Omega > 0$.

The output spike train prediction with both methods is presented in Figure 5a. The corresponding errors are $ERR_1^t = 0.6\%$, $ERR_2^t = 1.06\%$, $ERR_1^y = 25.22\%$, and $ERR_2^y = 26.68\%$, also summarized in Table 1. The computing times are 0.32 s for the proposed method and 0.81 s for the indirect method.

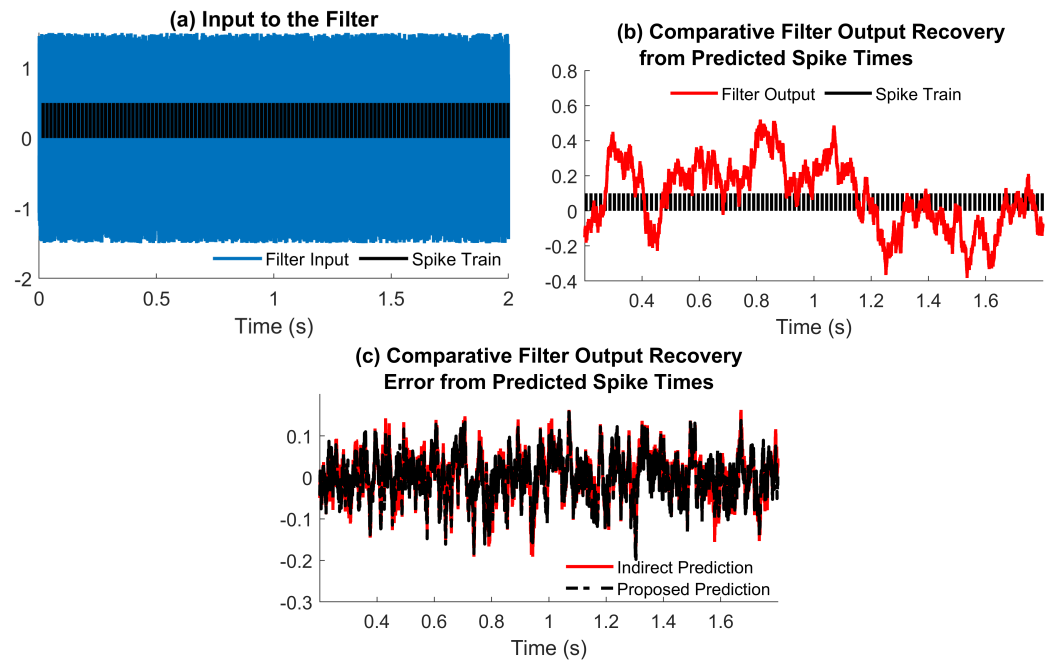


Figure 4. Output spike train prediction for uniform random noise. (a) Filter input and corresponding spike train. (b) Filter output and corresponding spike train. (c) Recovery errors $e_2^y(t)$ and $e_1^y(t)$ with the indirect and proposed methods, respectively.

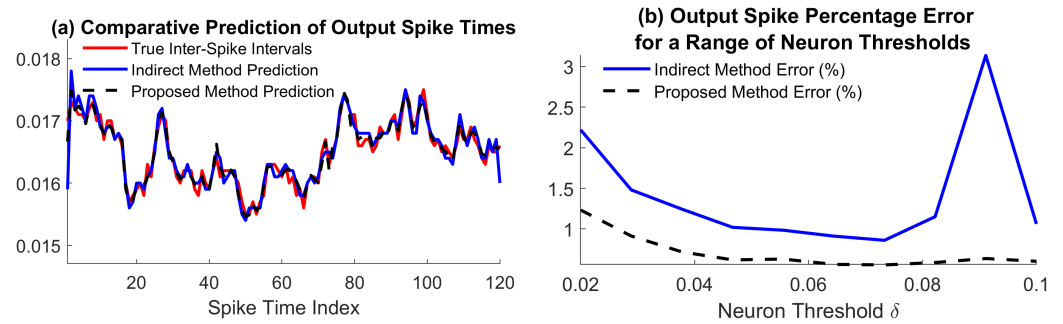


Figure 5. (a) Spike train prediction comparative performance: the true inter-spike intervals, interposed on the predicted inter-spike intervals with the indirect and proposed methods, respectively. (b) Spike train prediction comparative performance for 10 values of δ uniformly spaced in $[0.02, 0.1]$.

5.3. The Effect of Spike Density on Performance

During our experiments, we noticed that neuron threshold δ has an important role in deciding the computation time of both prediction routines. To illustrate this, we repeated the experiment in Section 5.2 using the same settings where δ was iteratively assigned 10 values uniformly spaced in interval $[0.02, 0.1]$. As before, we measured the error on the predicted spike train, the reconstructed filter output and computation time for each method, illustrated in Figures 5b and 6a,b, respectively.

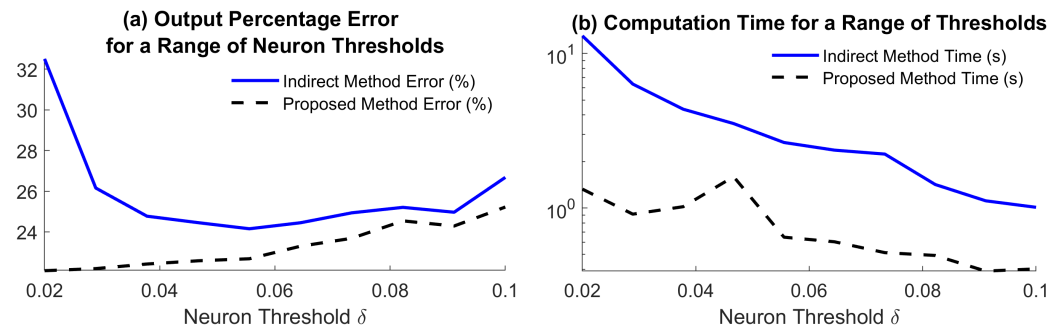


Figure 6. Filter output computation for 10 values of δ uniformly spaced in $[0.02, 0.1]$. (a) Analog output error. (b) Computation time.

5.4. Output Spike Prediction for Neuron Sampling below the Nyquist Rate

In the examples presented so far, the filter output is sampled at a rate that allows input reconstruction. In this example, we use a uniform white noise input bounded in $[-1.5, 1.5]$. The input is filtered using a bandpass filter computed employing the Daubechies orthogonal wavelet 30, depicted in Figure 7. We note that the effective bandwidth is around 1200 rad/s. When sampled with an IF with parameters $\delta = 0.04, b = 6$, the filter output samples satisfy $\max(t_{k+1}^y - t_k^y) = 0.0074 > \frac{\pi}{\Omega} = 0.0026$, which violates the condition for perfect recovery in (3). Therefore, in this example, we only evaluate the error using ERR_j^t , as ERR_j^y involves the reconstruction of the filter output from IF measurements.

We note that, as in Sections 5.2 and 5.3, here, the input is not bandlimited, and its recovery cannot be guaranteed by (3). In this case, the choice of the bandwidth parameter Ω used in recovery is not determined by any existing theoretical result. We run the conventional method by changing the recovery bandwidth across 11 uniformly distributed values in interval $[500 \text{ rad/s}, 1500 \text{ rad/s}]$. The comparative results are depicted in Figure 8a,b. We note that the proposed direct method does not have the bandwidth as a parameter, and therefore its performance is the same for all choices of Ω . The results show that the error introduced by the conventional method ERR_2^t changes quite significantly with Ω . An error around 15% higher than the proposed method is achieved for $\Omega \in [600, 1300]$, but the difference becomes significantly larger for Ω outside this interval. Meanwhile, the

computation time of the proposed method is more than one order of magnitude lower than that of the conventional method (see Figure 8b) for all values of Ω .

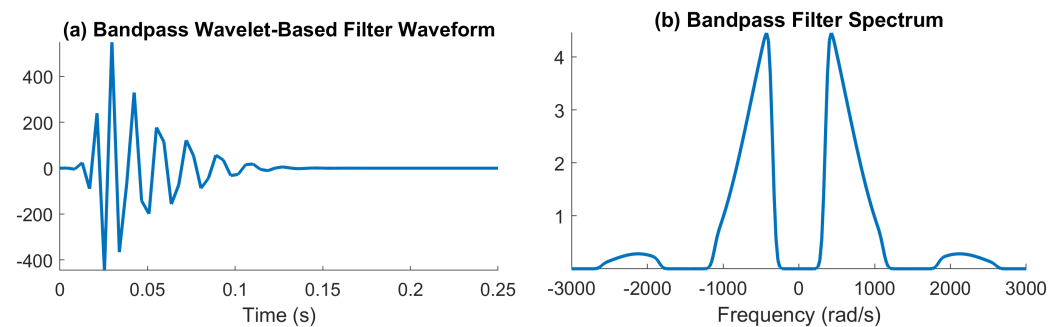


Figure 7. Bandpass filter computed via the Daubechies orthogonal wavelet 30.

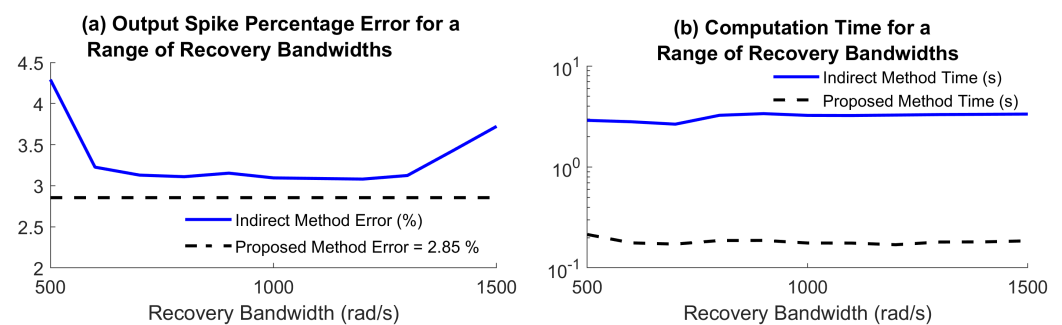


Figure 8. Filter output computation for 11 values of the reconstruction bandwidth uniformly spaced in [500 rad/s, 1500 rad/s]. (a) Spike train output error. (b) Computation time.

6. Conclusions

In this work, we introduced a new method to filter an analog signal via its neuromorphic measurements. Unlike existing approaches, the method does not require imposing smoothness type assumptions on the analog input and filter output, such as a limited bandwidth. We introduced recovery guarantees, showing that it is possible to approximate the output spike train with arbitrary accuracy for an appropriate choice of the sampling model. We compared the proposed method numerically against the conventional solution to this problem, which involves reconstruction of the analog signal. The results show the accuracy of the proposed method is comparable to that of the conventional approach. However, the computing time was smaller for the proposed method in all examples, ranging from 2–3 times up to more than one order of magnitude smaller.

Conceptually, the proposed method has the advantage of not depending on the characteristics of the analog signal, and therefore it is not restricted to satisfy any reconstruction guarantees. As demonstrated numerically, the method works well in the case of random inputs, as well as when the input and output of the filter are sampled below Nyquist. Moreover, given the fact that it bypasses input reconstruction, the proposed method is not affected by known artefacts of recovery methods such as boundary errors.

This work can be extended in several directions. First, the theoretical results can be extended to work with higher-order interpolation rather than a piecewise constant, which may lead to better error bounds. Second, the results can be extended for the more general scenarios of multichannel or nonlinear filters. Third, the proposed algorithm could be implemented in hardware and tested in practical communication scenarios. This work has the potential to lead to the development of neuromorphic filters that would facilitate a faster transition towards a power-efficient computing infrastructure.

Author Contributions: All the authors have contributed substantially to the paper. D.F. conceptualized the methodology, performed numerical studies and prepared the draft paper. D.C. contributed

in the writing phase through review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding. The APC is funded by the Imperial College London Open Access Fund.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

IF Integrate and fire
TEM Time encoding machine

References

1. Rabiner, L.R.; Gold, B. *Theory and Application of Digital Signal Processing*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1975.
2. Lazar, A.A.; Tóth, L.T. Perfect recovery and sensitivity analysis of time encoded bandlimited signals. *IEEE Trans. Circuits Syst. I* **2004**, *51*, 2060–2073.
3. Roza, E. Analog-to-digital conversion via duty-cycle modulation. *IEEE Trans. Circuits Syst. II* **1997**, *44*, 907–914.
4. Chen, J.; Skatchkovsky, N.; Simeone, O. Neuromorphic integrated sensing and communications. *IEEE Wirel. Commun. Lett.* **2022**, *12*, 476–480.
5. Skatchkovsky, N.; Jang, H.; Simeone, O. Spiking neural networks—Part III: Neuromorphic communications. *IEEE Commun. Lett.* **2021**, *25*, 1746–1750.
6. Abdul-Kreem, L.I.; Neumann, H. Estimating visual motion using an event-based artificial retina. In *Computer Vision, Imaging and Computer Graphics Theory and Applications, Proceedings of the 10th International Joint Conference, VISIGRAPP 2015, Berlin, Germany, 11–14 March 2015*; Revised Selected Papers 10; Springer: Cham, Switzerland, 2016; pp. 396–415.
7. Gallego, G.; Delbrück, T.; Orchard, G.; Bartolozzi, C.; Tabata, B.; Censi, A.; Leutenegger, S.; Davison, A.J.; Conradt, J.; Daniilidis, K.; et al. Event-based vision: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *44*, 154–180.
8. Kowalczyk, M.; Kryjak, T. Interpolation-Based Event Visual Data Filtering Algorithms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Vancouver, BC, Canada, 17–24 June 2023; pp. 4055–4063.
9. Scheerlinck, C.; Barnes, N.; Mahony, R. Asynchronous spatial image convolutions for event cameras. *IEEE Robot. Autom. Lett.* **2019**, *4*, 816–822.
10. Lazar, A.A. A simple model of spike processing. *Neurocomputing* **2006**, *69*, 1081–1085.
11. Florescu, D.; Coca, D. Implementation of linear filters in the spike domain. In *Proceedings of the 2015 European Control Conference (ECC), Linz, Austria, 15–17 July 2015*; pp. 2298–2302.
12. Lazar, A.A. Time encoding with an integrate-and-fire neuron with a refractory period. *Neurocomputing* **2004**, *58*, 53–58.
13. Florescu, D.; Coca, D. A novel reconstruction framework for time-encoded signals with integrate-and-fire neurons. *Neural Comput.* **2015**, *27*, 1872–1898.
14. Paninski, L.; Pillow, J.W.; Simoncelli, E.P. Maximum likelihood estimation of a stochastic integrate-and-fire neural encoding model. *Neural Comput.* **2004**, *16*, 2533–2561.
15. Florescu, D.; Coca, D. Identification of linear and nonlinear sensory processing circuits from spiking neuron data. *Neural Comput.* **2018**, *30*, 670–707.
16. Maass, W.; Natschläger, T.; Markram, H. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Comput.* **2002**, *14*, 2531–2560.
17. Florescu, D.; Coca, D. Learning with precise spike times: A new decoding algorithm for liquid state machines. *Neural Comput.* **2019**, *31*, 1825–1852.
18. Florescu, D. A Generalized Approach for Recovering Time Encoded Signals with Finite Rate of Innovation. *arXiv* **2023**, arXiv:2309.10223.
19. Gontier, D.; Vetterli, M. Sampling based on timing: Time encoding machines on shift-invariant subspaces. *Appl. Comput. Harmon. Anal.* **2014**, *36*, 63–78.
20. Alexandru, R.; Dragotti, P.L. Reconstructing classes of non-bandlimited signals from time encoded information. *IEEE Trans. Signal Process.* **2019**, *68*, 747–763.
21. Hilton, M.; Dragotti, P.L. Sparse Asynchronous Samples from Networks of TEMs for Reconstruction of Classes of Non-Bandlimited Signals. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, 4–10 June 2023.
22. Lazar, A.A.; Pnevmatikakis, E.A. Faithful representation of stimuli with a population of integrate-and-fire neurons. *Neural Comput.* **2008**, *20*, 2715–2744.
23. Shannon, C.E. Communication in the presence of noise. *Proc. IRE* **1949**, *37*, 10–21.

24. Feichtinger, H.G.; Gröchenig, K. Theory and practice of irregular sampling. *Wavelets Math. Appl.* **1994**, 305–363.
25. Gröchenig, K. Reconstruction algorithms in irregular sampling. *Math. Comput.* **1992**, 59, 181–194.
26. Aldroubi, A.; Gröchenig, K. Nonuniform sampling and reconstruction in shift-invariant spaces. *SIAM Rev.* **2001**, 43, 585–620.
27. Christensen, O. *An Introduction to Frames and Riesz Bases*; Birkhauser: Boston, NJ, USA, 2003.
28. Lazar, A.A.; Pnevmatikakis, E.A. Consistent Recovery of Sensory Stimuli Encoded with MIMO Neural Circuits. *Comput. Intell. Neurosci.* **2010**, 2010, 469658.
29. Lazar, A.A.; Pnevmatikakis, E.A. Reconstruction of sensory stimuli encoded with integrate-and-fire neurons with random thresholds. *EURASIP J. Adv. Signal Process.* **2009**, 2009, 682930.
30. Lazar, A.A.; Slutskiy, Y.B. Channel identification machines. *Comput. Intell. Neurosci.* **2012**, 2012, 209590.
31. Lazar, A.A.; Zhou, Y. Identifying multisensory dendritic stimulus processors. *IEEE Trans. Mol. Biol. Multi-Scale Commun.* **2016**, 2, 183–198.
32. Lazar, A.A.; Zhou, Y. Volterra dendritic stimulus processors and biophysical spike generators with intrinsic noise sources. *Front. Comput. Neurosci.* **2014**, 8, 95.
33. Lazar, A.A.; Simonyi, E.K.; Tóth, L.T. An overcomplete stitching algorithm for time decoding machines. *IEEE Trans. Circuits Syst. I* **2008**, 55, 2619–2630.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.