

# Theory and Algorithms for Pulse Signal Processing

Gabriel Nallathambi and Jose C. Principe

**Abstract**—The integrate and fire converter (IFC) transforms an analog signal into a train of biphasic pulses. The pulse train has information encoded in the timing and polarity of pulses. While it has been shown that any finite bandwidth analog signal can be reconstructed from these pulse trains with an error as small as desired, there is a need for fundamental signal processing techniques to operate directly on pulse trains without signal reconstruction. In this paper, the feasibility of performing online the operations of addition, multiplication, and convolution of analog signals using their pulses train representations is presented. The theoretical framework to perform signal processing with IFC pulse trains imposing minimal restrictions is derived, and algorithms for online implementation of the operators is developed. The performance of the proposed algorithms is studied by quantifying the variations in instantaneous occurrence of pulses. Comparisons are performed with digital processing of reconstructed pulse trains. Moreover, an application of noise subtraction and representation of relevant features of interest in electrocardiogram signal is demonstrated with a sparse data rate of less than 20 IFC pulses per second, and an absolute error in heart rate of  $0.16 \pm 0.18$  bpm.

**Index Terms**—Analog to pulse converter, biphasic pulse trains, convolution, pulse signal processing, semantic information.

## I. INTRODUCTION

One of the central principles in digital signal processing is the Whittaker–Shannon–Nyquist sampling theorem, which states that there is no loss of information between bandlimited analog signals and digital representations if the sampling rate is at least twice the maximum frequency present in the analog signal of interest [1]–[3]. Driven by sampling theory, programming flexibility and transistor scaling, nearly all data acquisition, processing and communication has progressed from continuous domain to the digital domain [4]. These advances along with the availability of high fidelity, low cost analog to digital converters (ADC) and digital signal processors (DSP) have led to an exponential increase in the digitalization of information processed from analog world sources [5]. The sampling theorem is a worst–case theorem, because it assumes that the highest frequency of input signal is always present, which is not always the case. Therefore, conventional Nyquist

sampling results in redundant sample representations that can overwhelm bandwidth in communications, and DSPs in real–time portable applications [5]. Recent developments in alternative sampling schemes such as compressive sensing [6], finite rate of innovation [7], and signal–dependent time–based samplers [8]–[10] show the ability to reconstruct signals well below the Nyquist rate. These approaches normally use conventional ADC followed by a compression step recognizing that useful information in real world signals is sparser than the raw data generated by sensors. The focus of this paper is to present an alternative to achieve lower data rates and the potential for ultra–low power computation by combining sensing and compression in a single step. We focus on pulse trains created by a special type of analog to pulse converter named integrate and fire converter (IFC), which converts an analog signal of finite bandwidth into a train of pulses where a given area under the curve of the analog signal constrains the time difference between pulses [10].

The IFC is inspired by the leaky integrator and fire neuron model [11], which is a reasonable approximation for how neurons in our brain processes information. It takes advantage of the local time structure of the input, enabling users to select the IFC parameters according to the specified accuracy required by the application; conceptually, it provides a compressed representation of the analog signal, using the physical charge time of the capacitor as the sparseness constraint [12]–[14]. Rastogi et al. [10] studied the hardware implementation of the IFC and showed that the power consumption and area required is smaller than most of the ADCs available: a single channel IFC has  $\sim 30$  transistors with a figure of merit of 0.6 pJ/conv for an 8–bit converter, implemented using CMOS 0.6  $\mu\text{m}$  technology in a layout box of 100  $\mu\text{m}$  X 100  $\mu\text{m}$ . The authors in [15] proved mathematically the conditions to approximately reconstruct a finite bandwidth analog signal from the train of IFC pulses with an error as small as desired. Unfortunately, the simplicity in IFC sampling produces complex reconstruction algorithms at the backend.

One of the interesting features of the IFC is that the information is contained in the time of the occurrence of pulses. When there are no pulses the processor is idle, which is totally different from the design of current DSP algorithms that are

---

G. Nallathambi is with Vital Connect Inc., San Jose, CA 95110, USA (e-mail: gabriel\_n.@gmail.com).

J. C. Principe is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA (e-mail: principe@cnel.ufl.edu).

The work of the authors is funded via the grant DARPA N66001–15–1–4054 and NSF EAGER 1723366.

implemented in computers. Since the pulses in the IFC are zero or one, this converter can also lead to digital implementation, but they can be asynchronous, potentially saving lots of power and leading to new compromises between power and computation as required for Internet of Things (IoT) applications.

Various processing schemes have been proposed in the literature to reconstruct pulse trains generated by the IFC. The simplest technique counts pulses in time bins to create a coarse time structure of the pulse train and apply standard algorithms on the vector space representation. Alvarado et al. [12] used this approach to solve the heartbeat classification problem with linear discriminant classifiers and binned pulses as features. McCormick [16] proposed asynchronous finite state machines to perform piecewise linear operations and reconstruct binary codes from input pulses. Signal processing is performed on the binary code followed by conversion back to pulses. In order to avoid the reconstruction, Nallathambi and Principe [13] applied attribute grammars and automata directly to the pulse timing for performing non-numeric processing of pulse trains and identification of QRS complexes in the electrocardiogram (ECG) signal with high accuracy. In the neuroscience literature, the pulse trains created by neurons are modeled as stochastic point processes [17], and many machine learning techniques are used to compute with pulses [18], [19]. Another alternative, which is the focus of this paper, is to avoid the reconstruction by directly applying arithmetic operators to the IFC pulses under a deterministic framework, i.e., assuming the signal is created from a deterministic source and the conversion is also deterministic, as used in sampling theory. In principle, functional algebras are necessary for computation as the pulse trains are time functions. However, because of the special characteristic of IFC pulse trains (difference between pulses is constrained by a constant area in the analog signal), we define much simpler algorithms that operate online and produce a range of accuracies (30 to 70 dB of SNR) compatible with many IoT applications.

The main contributions of this paper are as follows. First, a theoretical framework for performing the basic signal processing operations of addition, multiplication, and convolution is derived. Secondly, algorithms for online implementation of pulse-based arithmetic and convolution is proposed, and tested.

The rest of the paper is organized as follows: Section II describes the IFC in detail and presents the related works on pulse-based signal processing. Section III derives the theoretical framework for operating with pulse trains to perform addition, multiplication, and convolution. Section IV proposes algorithms for online implementation of the theoretical framework. Section V describes the datasets and performance metrics used for validation. Section VI quantifies the performance of the algorithms using synthetic and natural data. Section VII discusses the possibilities offered by the present work. Detailed derivations for the theorems are presented in the Appendix. MATLAB scripts for the algorithms are made available in [20].

## II. INTEGRATE AND FIRE CONVERTER

Pulse trains are waveforms where the information is contained in the timing of pulses instead on their amplitude. The use of pulses for signal processing is not a new idea. Early efforts include works on arithmetic using pulse encoding methods such as pulse-based rate, width, edge, burst, phase, delay, and amplitude [21]–[24].

Since its inception, many studies such as pulse-based population encoding for single or multiple sensors in video processing [8], [9], [25], time-embedding based on the inter-pulse intervals [26], learned input-output mappings based on a stochastic model for the events [18], [19], stochastic point process models [17], projections into reproducing kernel Hilbert spaces [27], and others [28] based on pulse streams have been proposed. Based on these works, various implementation schemes for pulse signal processing are proposed using magnetic cores [29], reconfigurable analog systems [30], fourth order Palmo filter [31], etc. The trends in silicon technology with a decrease in voltage and an increase in speed are making pulse-based representations more appealing.

In this paper, we focus our discussion on the biphasic IFC, which converts real world analog signals to analog time between pulses. The IFC output encodes information on both the timing of the pulses (analog) and polarity of pulses (digital). The methodology developed in this work can be easily applied to single polarity pulse trains as well.

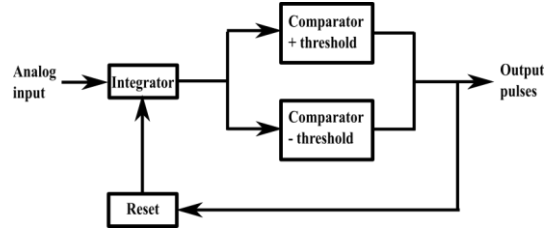


Fig. 1. Block diagram of the biphasic integrate and fire analog to pulse converter.

The IFC block diagram used in this paper is shown in Fig. 1. The analog input  $x(t)$  is integrated, and the result is compared against two thresholds. When either the positive or negative threshold  $\theta$  is reached, a pulse is generated at time  $t_k$  with positive or negative polarity  $p_k$  respectively. Unlike the integrate and fire neuron model, two thresholds are used to substantially reduce the mean data rate [32]. Fundamentally, each pulse interval satisfies the condition

$$\theta = \int_{t_k}^{t_{k+1}} x(t) e^{-\alpha(t_{k+1}-t)} dt \quad (1)$$

where  $\alpha$ , the rate of decay, models the leakage of the integrator in practical implementations [15], [32]. The pulse timings, the threshold and the rate of decay completely define the IFC pulse train output.

The IFC pulse train representation is rather different from discrete time representations. Pulses occur asynchronously in time, controlled by the amplitude of the analog signal, and the values of  $\theta$  and  $\alpha$ . There are several regimes possible for the IFC. Since the concept is very similar to the Asynchronous Delta Modulator, when the threshold is set very low, it produces a high-density pulse train that approximates the analog signal

when integrated. This regime simplifies the reconstruction (low pass filter similar to the reconstruction filter in digital to analog converter), but it produces a very high data rate when compared with the Nyquist frequency. We seek another point in the computation/data rate domain where the selection of the threshold is such that the data rates are comparable, or inferior to the Nyquist rate, which saves power but complicates the reconstruction. The dynamic range of the density of pulses depends upon the local structure of the input, with more pulses occurring in the large amplitude region of the analog signal, and fewer pulses appearing in the low amplitude portions of the analog signal. This creates a fundamental constraint for reconstruction and processing of pulse trains. Feichtinger et al. [15] studied the reconstruction of the analog signal from the pulses using frame theory and showed that it is possible to approximately reconstruct a bandlimited signal in  $L^\infty$  norm with an error proportional to the threshold  $\theta$ . In [15] a simpler procedure employing finite bandlimited spaces is presented based on least squares using splines or Fourier bases such that  $\hat{x}(t) = \sum_{k=1}^M a_k \phi_k(t)$ , where  $a_k$  is given by the linear regression  $\hat{\theta} = S\hat{a}$ ,  $S$  is obtained by integrating the basis set over the reconstruction interval, and  $\|x(t) - \hat{x}(t)\|_\infty \leq C\theta$  where  $C$  is a constant solely dependent on the window of analysis and the choice of the bases functions.

We explain next a theoretical framework for performing basic signal processing operations such as arithmetic and convolution directly on pulse trains. Moreover, algorithms to implement these operators are also proposed, where the processing of information is online and entirely in the time domain as the inputs and output of the system are pulse trains.

### III. THEORY OF PULSE SIGNAL PROCESSING FOR IFC

#### A. Setting up the problem for online operation.

IFC maps a continuous time, continuous amplitude signal into the structure of train of pulses in analog time such that the distance between any consecutive pulses  $t_k$  and  $t_{k+1}$  is fundamentally constrained by the threshold  $\theta$ , which controls the density of pulses; therefore, any arithmetic operation on pulse trains (addition or multiplication of pulses) also must be constrained by  $\theta$ . From eqn. (1), it is observed that  $\theta$  is equal to the leaky area under  $x(t)$  between  $t_k$  and  $t_{k+1}$  where the rate of decay is given by  $\alpha$ . Hence, any operation on pulse trains corresponds to equivalent operations on underlying areas.

Intuitively, it is straight-forward to determine from eqn. (1), a relation between areas and time as  $x(t_k) = \frac{\theta}{t_{k+1} - t_k}$  under the assumption that  $x(t)$  is constant between  $t_k$  and  $t_{k+1}$ , with a rate of decay of zero. We will explain first the main ideas for addition in this simplified framework, assuming that all the pulses are positive and the rate of decay is zero. Subsequently, this framework will be extended to bipolar pulses and non-zero rate of decay.

The goal is to perform arithmetic operations between pairs of continuous time signals  $x(t)$  and  $y(t)$ , i.e.,  $s(t) = x(t) + y(t)$ , using the corresponding pulse trains. Let us denote the polarity of the pulse train created from the signal  $x(t)$  as  $X = \{p_{x_k}, k = 1, 2, \dots\}$ . The timing of the pulses in  $X$  is very important, so we will also refer to their timings as  $t_{x_k}$ . Obviously, the addition

can only be carried out when a new pulse occurs in either of the asynchronous pulse-based representations of each pulse train. The general procedure is to estimate the integral between the *two most recent pulses* in a pair of signals, which will be denoted by the *current interval*  $(t_a, t_b)$ . In this framework,  $t_a$  and  $t_b$  may belong to the same pulse train, i.e.,  $t_a = t_{x_k}, t_b = t_{x_{k+1}}$  or alternatively between both pulse trains, i.e.,  $t_a = t_{x_k}, t_b = t_{y_k}$  depending on the local time structure of the signals. The consecutive pulses that compose the sum pulse train  $S$  still have to obey the IFC threshold  $\theta$ , which means that, very likely, somewhere in the interval  $(t_a, t_b)$  a new pulse for  $t_{s_k}$  should be created at a time when the sum of the areas reach  $\theta$ . This methodology has two implications: first, areas must be converted online with high precision into pulse timing; second, because in general the pulse at  $t_{s_k}$  does not coincide with any of the pulses in the addend or augend, there will be an *excess area (EA)* that needs to be stored and taken into consideration when evaluating the area for the next pair of pulses (carry over).

Suppose that  $t_{s_m}$  denotes the pulse time of the resultant sum of augend pulse train and addend pulse train, and  $0 = t_{x_0} = t_{y_0} < t_{x_1} < t_{x_2} \dots < t_{x_m} < t_{y_1}$  (Fig. 2). Let us define  $A_i = t_{x_i} - t_{x_{i-1}}$  and  $B_i = t_{y_i} - t_{y_{i-1}}$  for the augend and addend pulse trains respectively. The sequence of computing the rate of areas in each current interval  $(t_a, t_b)$  is illustrated in Table I. The *resultant area (RA)* due to addition at the end of each interval is given by the sum of rate of area per unit time in the augend and addend. To compute the exact area since the last pulse in the sum, the *EA* from the previous interval must be added to *RA* in current interval to obtain the *total area (TA)* due to addition. The exact timing to establish the location of  $t_{s_k}$  is obtained when *TA* exceeds one constant area.

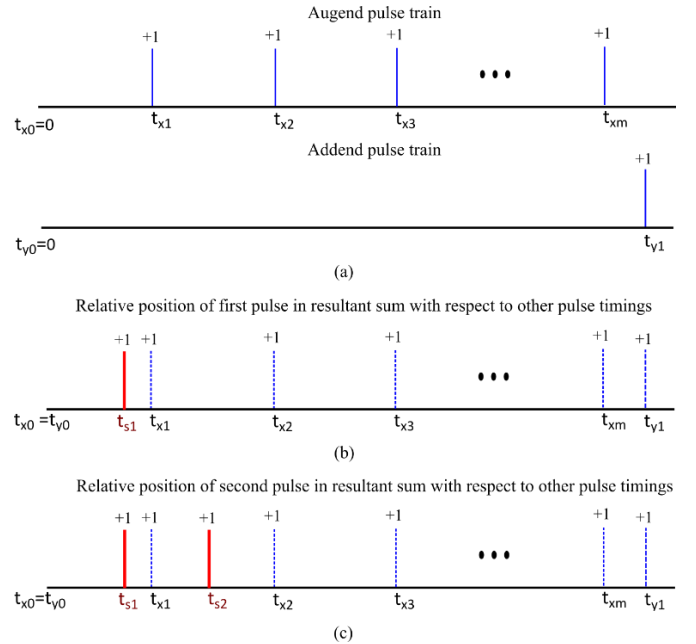


Fig. 2. Illustration of addition of pulse trains with positive polarities.

TABLE I  
ILLUSTRATION OF AREA CALCULATIONS IN EACH INTERVAL

| $(t_a, t_b)$         | RA                    | TA                                 | EA                             |
|----------------------|-----------------------|------------------------------------|--------------------------------|
| $(t_{x_0}, t_{x_1})$ | $1 + \frac{A_1}{B_1}$ | $1 + \frac{A_1}{B_1}$              | $\frac{A_1}{B_1}$              |
| $(t_{x_1}, t_{x_2})$ | $1 + \frac{A_2}{B_1}$ | $1 + \frac{\sum_{i=1}^2 A_i}{B_1}$ | $\frac{\sum_{i=1}^2 A_i}{B_1}$ |
| $(t_{x_2}, t_{x_3})$ | $1 + \frac{A_3}{B_1}$ | $1 + \frac{\sum_{i=1}^3 A_i}{B_1}$ | $\frac{\sum_{i=1}^3 A_i}{B_1}$ |

### B. General case for a single pair of pulses

Let us now address the general case of a pair of bipolar pulses with non-zero decay rates in the IFC for addition, multiplication and convolution. In this section, the RA due to an operation for a *single pair of pulses* is treated first for simplicity (no carryover). In section IV, we propose algorithms for updating recursively areas with carryovers by computing EA and TA, and discuss the choice of the overlapping time interval  $(t_a, t_b)$  for online implementation.

Suppose  $A_1(t_j, t_{j+1}) \circ A_2(t_j, t_{j+1})$ , where  $\circ$  represents the operator (+ or \*) on pulse trains,  $t_j$  and  $t_{j+1}$  are the most recent pulse times from the original pulse trains, and  $A_i(t_j, t_{j+1})$  is the underlying area of the  $i^{th}$  pulse train ( $i = 1, 2$ ) during  $(t_j, t_{j+1})$ . For the single pair of pulses case,  $(t_a, t_b)$  reduces to  $(t_j, t_{j+1})$  because this is always the most recent pair of pulses. The framework solves  $A_1(t_a, t_b) \circ A_2(t_a, t_b) = \eta\theta$  for  $\eta$  and  $t_k$ , where  $\eta$  is the RA and the pulse timing  $t_k$  due to the operation occurs when  $\eta = 1$ .

To solve deterministically and online the above equation, it is assumed that the input signal is constant between  $t_k$  and  $t_{k+1}$ . This simple signal model is an approximation controlled by the threshold  $\theta$  and depends upon the local structure of the pair of pulses, but it was surprisingly shown sufficient for many IoT applications such as continuous health monitoring. For completeness, the corresponding error bounds due to this assumption are also studied.

**Observation 1:** If  $\theta = \int_{t_k}^{t_{k+1}} x(t)e^{-\alpha(t_{k+1}-t)}dt$  and  $x(t)$  is constant between  $t_k$  and  $t_{k+1}$ , then  $\xi\theta = \int_{t_a}^{t_b} x(t)e^{-\alpha(t_{k+1}-t)}dt$ , where  $\xi = \frac{g(t_{k+1}-t_a)-g(t_{k+1}-t_b)}{g(t_{k+1}-t_k)}$ ,  $t_k \leq t_a < t_b \leq t_{k+1}$ , and  $g(t) = 1 - e^{-\alpha t}$  for  $\alpha > 0$ . This observation is critical in the derivation of theorems for arithmetic as it enables generalization of the results in any interval  $(t_a, t_b)$  between two consecutive pulses.

Derivations of the subsequent theorems is provided in the Appendix. The only assumption in this framework is that the input signal is constant between two consecutive pulses. The errors in pulse train arithmetic due to this assumption are also derived in the Appendix, and studied in section VI.

**Theorem 1 – Addition of single pair of pulses:** Consider two analog signals (continuous amplitude and time)  $x(t)$  and  $y(t)$  corresponding to augend and addend pulse trains  $X$  and  $Y$  respectively. Suppose the  $X$  pulses occur at  $t_{x_j}$  with polarity  $p_{x_j}$ ,  $Y$  pulses occur at  $t_{y_j}$  with polarity  $p_{y_j}$ , and the pulses of the sum pulse train  $S$  occur at  $t_{s_j}$  with polarity  $p_{s_j}$  such that

$t_{x_n}, t_{y_n} \leq t_a < t_b \leq t_{x_{n+1}}, t_{y_{n+1}}$ , then the distance between consecutive pulses of  $S$  is  $t_{s_{n+1}} - t_{s_n} = \frac{-1}{\alpha} \ln\{1 - K\gamma_S\}$  with  $K = \frac{sgn(\eta_S)p_{x_{n+1}}}{p_{y_{n+1}}}$  and  $\gamma_S = \frac{g(t_{x_{n+1}}-t_{x_n})g(t_{y_{n+1}}-t_{y_n})}{p_{x_{n+1}}g(t_{x_{n+1}}-t_{x_n})+p_{y_{n+1}}g(t_{y_{n+1}}-t_{y_n})}$ . Furthermore, the polarity of the pulses of  $S$ ,  $p_{s_{n+1}} = sgn(\eta_S)$  where  $sgn(\cdot)$  is the signum function, and

$$\eta_S = \frac{p_{x_{n+1}}[g(t_{x_{n+1}}-t_a)-g(t_{x_{n+1}}-t_b)]}{g(t_{x_{n+1}}-t_{x_n})} + \frac{p_{y_{n+1}}[g(t_{y_{n+1}}-t_a)-g(t_{y_{n+1}}-t_b)]}{g(t_{y_{n+1}}-t_{y_n})}$$

**Theorem 2 – Multiplication of single pair of pulses:** Consider two analog signals  $x(t)$  and  $y(t)$  corresponding to multiplicand and multiplier pulse trains  $X$  and  $Y$  respectively and let  $r(t) = 1$  correspond to the identity (reference) pulse train  $R$ . Then the distance between consecutive pulses in the product pulse train  $P$  is  $t_{p_{n+1}} - t_{p_n} = \frac{-1}{\alpha} \ln\{1 - \gamma_P\}$  where

$$\gamma_P = \frac{g(t_{x_{n+1}}-t_{x_n})g(t_{y_{n+1}}-t_{y_n})}{g(t_{r_{n+1}}-t_{r_n})}. \text{ The polarity of the pulses of } P \text{ is given by } p_{p_{n+1}} = sgn(\eta_P) \text{ where } \eta_P = \left[ \frac{p_{x_{n+1}}p_{y_{n+1}}g(t_{r_{n+1}}-t_{r_n})}{g(t_{r_{n+1}}-t_a)-g(t_{r_{n+1}}-t_b)} \right] \cdot \left[ \frac{g(t_{x_{n+1}}-t_a)-g(t_{x_{n+1}}-t_b)}{g(t_{x_{n+1}}-t_{x_n})} \right] \cdot \left[ \frac{g(t_{y_{n+1}}-t_a)-g(t_{y_{n+1}}-t_b)}{g(t_{y_{n+1}}-t_{y_n})} \right].$$

**Theorem 3 – Convolution of a single pair of pulse train segments:** Consider two analog signals  $x(t)$  and  $y(t)$  corresponding to pulse train  $X$  and pulse train  $Y$  respectively and let  $r(t) = 1$  correspond to the identity (reference) pulse train  $R$ . Then, the distance between the pulses in the convolution pulse train  $C$  is  $t_{c_{n+1}} - t_{c_n} = \frac{-1}{\alpha} \ln\{1 - \gamma_C\}$  with

$$\gamma_C = \frac{g(t_{x_{n+1}}-t_{x_n})g(t_{y_{n+1}}-t_{y_n})}{g(t_{r_{n+1}}-t_{r_n})(\beta_2-\beta_1)}, \beta = \beta_2 - \beta_1 \text{ is the period of intersection of the two pulse trains, and the polarity of the pulses of } C \text{ is } p_{c_{n+1}} = sgn(\eta_C), \text{ where } \eta_C = \frac{-1}{\alpha} \left[ \frac{p_{x_{n+1}}p_{y_{n+1}}g(t_{r_{n+1}}-t_{r_n})}{g(t_{r_{n+1}}-t_a)-g(t_{r_{n+1}}-t_b)} \right] \cdot \left[ \frac{g(t_{x_{n+1}}-t_a)-g(t_{x_{n+1}}-t_b)}{g(t_{x_{n+1}}-t_{x_n})} \right] \cdot \left[ \frac{g(t_{y_{n+1}}-t_a+\lambda_2)-g(t_{y_{n+1}}-t_b+\lambda_2)-g(t_{y_{n+1}}-t_a+\lambda_1)+g(t_{y_{n+1}}-t_b+\lambda_1)}{g(t_{y_{n+1}}-t_{y_n})} \right],$$

and  $\lambda = \lambda_2 - \lambda_1$  is the time offset (shift).

This framework computes one instance of the output pulse  $t_k$  resulting from RA due to addition, multiplication and convolution. Notice that the formulas have the same general form, only differing on the specifics of the operation regarding the area and rate of area calculations. In the next section, online algorithms for pulse trains extend this theoretical framework, where the focus is on the recursive computation of areas using carryovers when there are multiple pulse instances.

## IV. ALGORITHMS FOR PULSE TRAIN ARITHMETIC

### A. Online pulse train arithmetic algorithms

In Table II, the algorithm for computing online arithmetic of two pulse trains is presented. The time interval  $(t_a, t_b)$  may be selected to shift forward in fixed or variable contiguous windows across time. Sliding  $t_a$  and  $t_b$  in fixed intervals requires prior knowledge of the minimum value of the inter-pulse interval to ensure the observation window lies within

$(t_k, t_{k+1})$  as per Observation 1. In this paper, to ensure online implementation, contiguous sliding windows are used where the interval shifts forward at the arrival of every new pulse in the operands, i.e., variable window lengths, as described in Section III A. We could also use shifting based on the pulse timing of only one of the pulse trains, but this would compromise the accuracy of the result, since fewer updates to the areas would have been calculated, so we recommend the former.

The only algorithmic difference to recursively update the area for online operation, when compared with the pair of pulses case presented in section III, is the need to consider carryovers in both timing and area, namely excess time ( $t_{ex}$ ) and excess area ( $\eta_{ex}$ ) as demonstrated in the beginning example. As the online algorithm is based on an integration window defined by the original pulse trains, the pulses of the output pulse train for the arithmetic operation may occur before the end of the current integration window. As such the time difference between the upper time limit of the integration window and the occurrence of the output pulse (i.e.,  $t_{ex}$ ) should be added to the evaluation of the pulse timing for the next interval. Likewise, the area corresponding to the excess time ( $\eta_{ex}$ ) should be added to resultant area ( $\eta$ ) to yield  $TA$ . In order to perform addition, multiplication and convolution, excess time and  $EA$  are included in the calculation of  $TA$  and new pulse timing at every computation as shown in Table II. The carryovers ensure higher precision in the calculation and guarantee that the output pulse timings always fall within the current integration window ( $t_a, t_b$ ).

TABLE II  
ALGORITHM FOR PULSE TRAIN ARITHMETIC

|    |                                                                                                                                                                  |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | Select computation time points: $t_a, t_b$                                                                                                                       |
| 2. | Obtain pulse intervals corresponding to computation time points: $t_{x_{n+1}}, t_{x_n}, t_{y_{n+1}}, t_{y_n}$ ; set $\eta_{ex}, t_{ex} \rightarrow 0$ initially. |
| 3. | Calculate $\eta$ , where $\eta = \eta_s$ for addition and $\eta = \eta_p$ for multiplication.                                                                    |
| 4. | Compute output pulse timing and polarity: $T_k, p_k$                                                                                                             |

Calculate  $g(t_{k_{n+1}} - t_{k_n})$ , where  $g(t_{k_{n+1}} - t_{k_n}) = K\gamma_s$  for addition and  $g(t_{k_{n+1}} - t_{k_n}) = \gamma_p$  for multiplication.

**while**  $|\eta + \eta_{ex}| \geq 1$   
 $T_k = \frac{-1}{\alpha} \ln\{1 - g(t_{k_{n+1}} - t_{k_n})|sgn(\eta) - \eta_{ex}|\} + T_{k-1} + t_{ex}$   
 $p_k = sgn(\eta)$   
 Update:  
 $t_a = T_k$   
 Calculate  $\eta$   
 $\eta_{ex}, t_{ex} \rightarrow 0, k \rightarrow k + 1$   
**end**

$\eta_{ex} \leftarrow \eta + \eta_{ex}$   
 $t_{ex} \leftarrow t_b - T_k$

Thus the process involves selection of the observation window and its associated pulse intervals, and calculation of the  $TA$  (i.e.,  $\eta + \eta_{ex}$ ) as per Table II. When  $TA$  exceeds +1 or -1, the pulse occurs at that time instant  $T_k$  with corresponding polarity  $p_k$ .

## B. Approximations

As the values of inter-pulse intervals are inversely proportional to signal amplitude, in high inter-pulse intervals compared to noise floor,  $g(m)$  is well approximated by  $\alpha m$  without degrading performance. With this approximation, it is straight-forward to show from first principles that  $\eta$  reduces to  $\eta_a = \frac{p_{x_{n+1}}(t_b - t_a)}{t_{x_{n+1}} - t_{x_n}} + \frac{p_{y_{n+1}}(t_b - t_a)}{t_{y_{n+1}} - t_{y_n}}$  and  $\eta_a = \left[ \frac{p_{x_{n+1}}p_{y_{n+1}}(t_b - t_a)(t_{r_{n+1}} - t_{r_n})}{(t_{x_{n+1}} - t_{x_n})(t_{y_{n+1}} - t_{y_n})} \right]$  for addition and multiplication respectively. Likewise,  $t_{k_{n+1}} - t_{k_n}$  reduces to  $t_{ka_{n+1}} - t_{ka_n} = \frac{t_b - t_a}{|\eta_a|}$  for both addition and multiplication. It is to be noted that  $\eta = \eta_a$  when the rate of decay is zero. Unlike Table II, the simplified equations rely directly on the inter-pulse intervals; therefore, the real-time hardware implementation of pulse-based systems is straight-forward. Moreover, the update step does not require recalculation of  $\eta_a$  as shown in Table III, which will speed up the computation in hardware. In section VI, we compare performance of the more accurate algorithm in Table II with this approximate algorithm.

TABLE III  
SIMPLIFIED ALGORITHM FOR PULSE TRAIN ARITHMETIC

|    |                                                                                                                                                                  |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | Select computation time points: $t_a, t_b$ .                                                                                                                     |
| 2. | Obtain pulse intervals corresponding to computation time points: $t_{x_{n+1}}, t_{x_n}, t_{y_{n+1}}, t_{y_n}$ ; set $\eta_{ex}, t_{ex} \rightarrow 0$ initially. |
| 3. | Calculate $\eta_a$ .                                                                                                                                             |
| 4. | Compute output pulse timing and polarity: $T_{ka}, p_{ka}$ .                                                                                                     |

Calculate  $t_{ka_{n+1}} - t_{ka_n}$ .

**while**  $|\eta_a + \eta_{ex}| \geq 1$   
 $T_{ka} = (t_{ka_{n+1}} - t_{ka_n})|sgn(\eta_a) - \eta_{ex}| + T_{ka-1} + t_{ex}$   
 $p_{ka} = sgn(\eta_a)$   
 Update:  
 $\eta_a \rightarrow \eta_a - (sgn(\eta_a) - \eta_{ex})$   
 $\eta_{ex}, t_{ex} \rightarrow 0, k \rightarrow k + 1$   
**end**

$\eta_{ex} \leftarrow \eta_a + \eta_{ex}$   
 $t_{ex} \leftarrow t_b - T_{ka}$

## C. Pulse convolution algorithm

The algorithm for computing the convolution of two pulse trains  $X$  and  $Y$  is presented in Table IV. The pulse timings of  $Y$  are reversed and shifted by  $\lambda$ , which is given by the minimum distance required for pulses of  $Y$  to reach one of the pulses of  $X$  upon shifting. The overlap  $\beta$  between the two pulse trains after shifting is computed; so, unlike pulse arithmetic, there is a vector of computation time points  $t_{a_i}, t_{b_i}$  and an associated vector of pulse intervals corresponding to all pulses in the region of overlap. Each element of these vectors results in an area  $\eta_i$  and  $TA$  resulting during a shift operation is given by the sum of  $\eta_i$ 's. When  $TA$  exceeds +1 or -1, the component of  $\eta$  at which this occurs determines the timing  $T_k$  and polarity  $p_k$  of the pulse resulting from the convolution of pulse trains. Similar to Table III, the equations for convolution is approximated by  $\eta_a = \left[ \frac{p_{x_{n+1}}p_{y_{n+1}}(t_b - t_a)(t_{r_{n+1}} - t_{r_n})(\lambda_2 - \lambda_1)}{(t_{x_{n+1}} - t_{x_n})(t_{y_{n+1}} - t_{y_n})} \right]$  and  $t_{ka_{n+1}} - t_{ka_n} = \frac{(t_{x_{n+1}} - t_{x_n})(t_{y_{n+1}} - t_{y_n})}{(\beta_2 - \beta_1)(t_{r_{n+1}} - t_{r_n})}$ .

## V. PERFORMANCE VALIDATION

### A. Performance measures

**Measures based on inter-pulse intervals:** The quality of performance is evaluated in terms of measures of accuracy between instantaneous amplitude values  $\hat{z}$  and  $z$  calculated from the inter-pulse intervals of the algorithmic pulse train output  $\hat{Z}$  and true pulse train output  $Z$  respectively, which is calculated in simulations for the analysis. The instantaneous amplitude  $\hat{z}$  of a pulse train with consecutive pulses  $t_k$  and  $t_{k+1}$  is given by  $\hat{z} = \frac{\theta}{t_{k+1}-t_k}$  and  $\hat{z} = \frac{\theta\alpha}{1-e^{-\alpha(t_{k+1}-t_k)}}$  for the IFC with zero and non-zero rate of decay respectively. Peak signal to noise ratio,  $PSNR = -10\log\left(\frac{\sum_{i=0}^N(\hat{z}_i - z_i)^2}{N(\hat{z}_{max} - \hat{z}_{min})^2}\right)$ , and correlation coefficient,  $r = \frac{\sum_{i=1}^N(\hat{z}_i - m_{\hat{z}})(z_i - m_z)}{\sqrt{\sum_{i=1}^N(\hat{z}_i - m_{\hat{z}})^2} \sqrt{\sum_{i=1}^N(z_i - m_z)^2}}$ , are the measures used to quantify the accuracy between instantaneous amplitude values of the algorithmic and true outputs, where  $m_{\hat{z}}$  and  $m_z$  are the sample mean of  $\hat{z}$  and  $z$  respectively.

TABLE IV  
ALGORITHM FOR PULSE TRAIN CONVOLUTION

1. Choose time offset  $\lambda = \lambda_2 - \lambda_1$ ; set  $\eta_{ex}, t_{ex} \rightarrow 0$  initially.
2. Shift the pulse timings of the reversed pulse train by  $\lambda$ .
3. Find the overlap  $\beta = \beta_2 - \beta_1$  between the pulse trains after shifting.
4. Select computation time points  $t_{a_i}, t_{b_i}$  for all pulses in the region of intersection.
5. Obtain pulse intervals corresponding to all computation time points in the region of intersection:  $t_{x_{n+1_i}}, t_{x_{n_i}}, t_{y_{n+1_i}}, t_{y_{n_i}}$
6. Calculate  $\eta_i$  for all computation time points  $t_{a_i}, t_{b_i}$
7. Compute output pulse timing and polarity:  $T_k, p_k$

Calculate  $t_{k_i} = \frac{-1}{\alpha} \ln \left\{ 1 - \frac{g(t_{x_{n+1_i}} - t_{x_{n_i}})g(t_{y_{n+1_i}} - t_{y_{n_i}})}{(\beta_2 - \beta_1)g(t_{r_{n+1_i}} - t_{r_{n_i}})} \right\}$  for all computation time points.

Calculate  $\eta = \text{sum}(\eta_1, \eta_2, \dots, \eta_n)$

**while**  $|\eta + \eta_{ex}| \geq 1$

Find first  $j$  at which  $|\eta_{ex} + \text{sum}(\eta_1, \eta_2, \dots, \eta_j)| \geq 1$

$\sigma = \text{sgn}(\eta_j) - \eta_{ex} - \text{sum}(\eta_1, \eta_2, \dots, \eta_{j-1})$

$T_k = \{|\eta_1 t_{k_1}| + \dots + |\eta_{j-1} t_{k_{j-1}}| + |\sigma t_{k_j}|\} + T_{k-1} + t_{ex}$

$p_k = \text{sgn}(\sigma)$

Update:

$\eta_j \rightarrow \eta_j - \sigma$

$\eta_1, \eta_2, \dots, \eta_{j-1} \rightarrow 0$

$\eta = \text{sum}(\eta_j, \eta_{j+1}, \dots, \eta_n)$

$\eta_{ex}, t_{ex} \rightarrow 0, k \rightarrow k + 1$

**end**

$\eta_{ex} \leftarrow \eta + \eta_{ex}$

$t_{ex} \leftarrow \lambda_2 - T_k$

**Region of analysis:** As the pulse representation is dependent upon the structure of the input, the analysis window is subdivided into four regions namely  $A, B, C$ , and  $D$  based on amplitude quartiles. Unlike conventional DSP, pulse-based computation has relatively lower incidence of pulses near the noise floor (region  $A$ ) and high pulse density in the other regions of interest. Hence it is necessary to quantify the performance in the individual regions and the overall

performance is reported in terms of mean  $\pm$  standard deviation of all regions.

**Comparative studies:** The focus of the paper is on processing the semantic information directly with pulse trains without signal reconstruction. However, to ensure completeness, the algorithmic pulse train output  $\hat{Z}$  is reconstructed to get  $\hat{z}_r(n)$ , and then compared with  $z_r(n)$  obtained using digital arithmetic of  $x_r(n)$  and  $y_r(n)$ , which are reconstructed from input pulse trains  $X$  and  $Y$  respectively as per [15]. The performance is assessed by computing  $PSNR$  and  $r$  between  $\hat{z}_r(n)$  and  $z_r(n)$ . Note that the  $PSNR$  as calculated here is a pessimistic estimate because the reconstruction algorithm is not ideal [37] as the inverse needs to be regularized. Nevertheless, it is useful to understand the overall peak performance in practical applications to compare with ADC converters.

### B. Data analysis

**Synthetic data:** Aperiodic pulse trains  $X$  and  $Y$  generated from 1V, 1Hz sinusoidal signals  $x(t)$  and  $y(t)$  respectively are selected to demonstrate the performance of the algorithm. Note that the frequency of the signal should not be considered in absolute terms, because what matters is the ratio of the frequency of the signal and the threshold in the IFC, i.e., the density of pulses. As a result, the performance of the algorithm is quantified for variations in the IFC parameters, and we directly show the number of pulses/sec in each part of the dynamic range of the signal, because equivalent  $PSNR$  can be expected in practical applications provided that this ratio is similar.

A two-sample t-test at 5% significance level is used to study the significant differences in the mean  $PSNR$  and data rate of the algorithms with and without approximations from Table II and Table III respectively. Comparative studies are performed using the synthetic data in terms of the aforementioned performance measures.

**Real ECG data:** An application of subtraction of baseline wander from ECG signal is used to demonstrate the feasibility of the proposed pulse-based algorithm for semantic information processing in continuous patient monitoring systems. Baseline wander is a very low frequency noise related to amplifier and electrode drift. In practice, a high pass filter (HPF) preprocessor with a cutoff frequency of 0.05 to 0.67 Hz is applied to the signal to attenuate the drift [33]. Here, we want to demonstrate that a pulse-based preprocessor can be applied instead with the same basic goal of attenuating the drift.

Input ECG pulse train  $X$  is obtained by corrupting an ECG signal of 1-minute duration from MIT-BIH arrhythmia database [34] (dataset 100) with 300 $\mu$ V, 0.2Hz sinusoidal baseline wander, and converting to pulses with the IFC parameters chosen as in [13]. Pulse subtraction of pulse train  $Y$  corresponding to sinusoidal baseline wander from pulse train  $X$  is used to obtain noise-free pulse train  $\hat{Z}$ . Further, to evaluate the accuracy of the noise removal and demonstrate the semantic information processing of the features in ECG, the baseline corrupted ECG signal is filtered using a digital high pass bidirectional IIR filter [33] of order 2 and cut-off frequency 0.5Hz, and converted to pulse train  $Z$  with the same IFC parameter values. Then, QRS complexes are detected from both the pulse trains  $Z$  and  $\hat{Z}$  using the automaton method [13], and

the performance of detection is assessed in terms of sensitivity and positive predictive value. Finally, the heart rate (HR) is calculated from the beat-to-beat R-to-R intervals, and the absolute error between the HR computed from pulse trains  $Z$  and  $\hat{Z}$  is studied to demonstrate that the cardiac events are well represented in  $\hat{Z}$  after pulse-based processing of the ECG signal.

## VI. RESULTS

### A. Algorithmic performance in regions of analysis

The algorithmic outputs of synthetic data for pulse train addition, multiplication, and convolution with  $\theta = 0.01$  and rate of decay of zero is studied. Performance analysis of pulse train addition shows overall PSNR,  $r$ , and data rate of  $32.58 \pm 18.35$  dB,  $0.92 \pm 0.14$ , and  $31.0 \pm 36.57$  pulses per second (p/s) respectively. The performance of the algorithm is dependent upon the regions of activity as demonstrated in Table V for pulse train addition where PSNR is greater than 40dB with  $r = 1$  in regions C and D, and less than 25dB with reduced data rate in regions A and B. Likewise, pulse train multiplication has PSNR,  $r$ , and data rate of  $31.45 \pm 13.7$  dB,  $0.95 \pm 0.07$ , and  $12.25 \pm 12.75$  p/s respectively, while pulse train convolution has PSNR,  $r$ , and data rate of  $29.61 \pm 4.73$  dB,  $0.98 \pm 0.03$ , and  $12.38 \pm 11.49$  p/s respectively, which means that precision in multiplication and addition are similar.

TABLE V  
PERFORMANCE IN THE REGIONS OF ANALYSIS (ADDITION)

| Region   | PSNR, dB | $r$  | Data rate, p/s |
|----------|----------|------|----------------|
| Region A | 14.10    | 0.71 | 3.33           |
| Region B | 20.60    | 0.97 | 10.67          |
| Region C | 42.36    | 1.0  | 26             |
| Region D | 53.47    | 1.0  | 84             |

These results demonstrate that the proposed algorithm is capable of producing accurate results in portions of the input that yield modest concentration of pulses (42 dB reconstruction for 26 pulses/sec), which is similar to 8-bit resolution with ADC. More pulses per sec increase the accuracy, but regions A and B where the data rate is very low, the errors are severe. This is a direct consequence of the assumption that the input signal is constant between  $t_k$  and  $t_{k+1}$ . The error  $\delta_{x+y}$  due to this assumption shown in Fig. 3 indicates the simple signal model results in higher errors when the density of pulses is extremely sparse. But, this does not incur a huge penalty in many real world applications as the low amplitude portions of signals have poor SNR (i.e., additive noise affects the signal fidelity substantially), and it is wasteful to use an ADC with large output wordlength to represent such regions. On the other hand, in the proposed methodology, the overall performance is constrained by the low density regions; hence, in practice, the user should set the minimum accuracy requirement for such regions (given the specified SNR for the application). However, when better accuracy is needed in the low amplitude regions of the signal near the noise margin, then the compromise is the increase in computation load (more pulses), which in turn yields a large data rate for transmission.

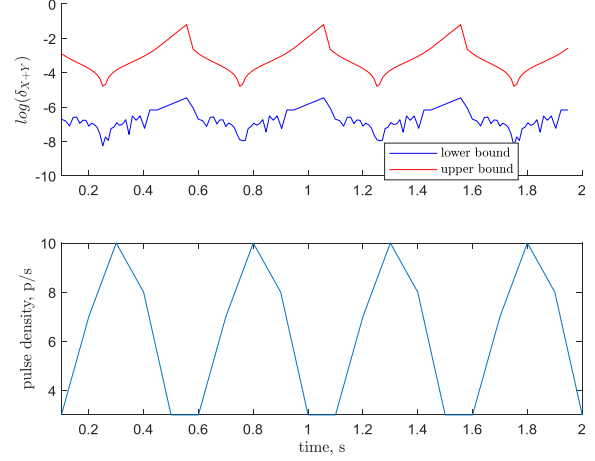


Fig. 3. Illustration of error bound in addition of pulse trains due to the simple signal model. The density of pulses is calculated in non-overlapping bins of width 100ms.

### B. Effect of IFC parameters

The effect of IFC threshold on the performance is studied across the regions of analysis. While the PSNR in regions C and D decreases gradually as the IFC threshold increases, the PSNR near the noise floor is consistently less than 20dB. Moreover, as the IFC threshold increases, the mean data rate decreases exponentially. From Fig. 4, it is evident that proper selection of IFC threshold guarantees lower data rates without degrading performance.

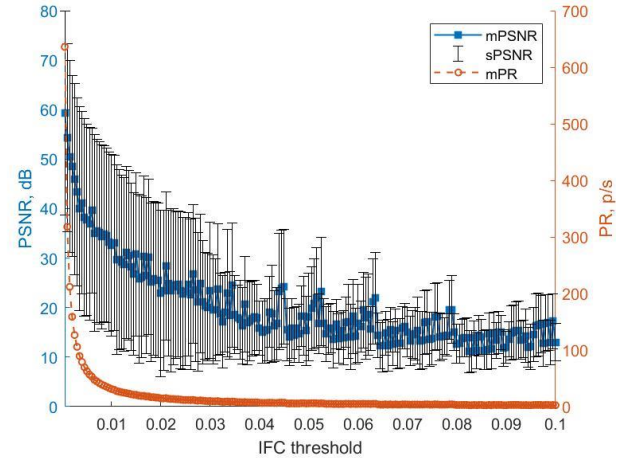


Fig. 4. Effect of IFC threshold on performance.  $mPSNR$  and  $sPSNR$  denote the mean and standard deviation of PSNR respectively, and  $mPR$  denotes the mean of data (pulse) rate.

The increase in PSNR variability for smaller IFC thresholds is expected. There are two basic components: first, the high variability in the data rate can be explained by the very different number of pulses in each of the percentiles of the amplitude dynamic range as shown in Table V. Second, because of the local nature of the sampling for the small time constants, which make measurements very local in time, we can expect larger variability (and lower repeatability across runs) for smaller thresholds.

The effect of IFC rate of decay and the zero decay approximation to the algorithm is presented in Fig. 5.

Comparison of performance of algorithm with and without approximation reveals PSNR at regions *A* and *B* to be similar in both cases while PSNR at regions *C* and *D* are significantly different ( $p < 0.05$ ) with the approximated algorithm having lower mean PSNR as shown in Fig. 5a. Moreover, the mean data rate of the algorithm with and without approximation across variations in IFC rate of decay is significantly different ( $p < 0.05$ ) as shown in Fig. 5b, where the approximated solution displays lower data rate. While the approximated algorithm in Table III offers simpler implementation dependent only on pulse intervals and provides sparse representation at higher rates of decay, there is a trade-off in PSNR at high amplitude regions when compared with the algorithm in Table II.

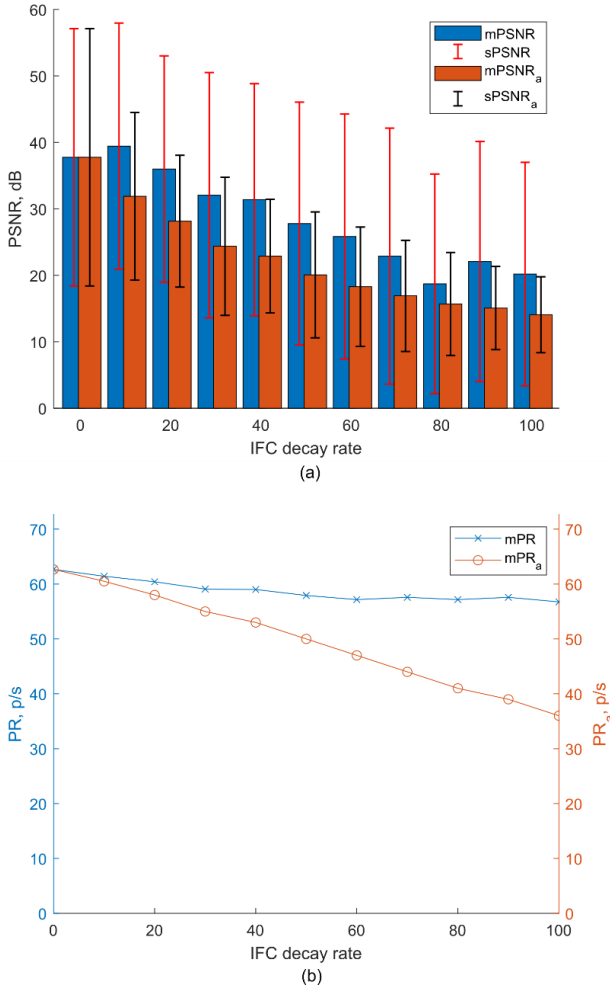


Fig. 5. Effect of IFC rate of decay and approximations on performance. The subscript 'a' in legend denotes the approximated algorithms.

### C. Comparison with digital processing

In Fig. 6, the performance curves for the comparison of the proposed algorithm with digital processing is presented. The performance characteristics are similar to Fig. 4 with PSNR being proportional to the signal amplitude for a given IFC threshold. It is to be noted that exact reconstruction for IFC is impossible and these results are obtained with the approximate method developed by Feichtinger et al. [15], which suffer from

the need to regularize the solution, decreasing the accuracy. Moreover, the size of the matrix for the reconstruction algorithm increases with the decrease in IFC threshold (more pulses are created). Therefore, the error in the reconstruction algorithm is not constant across different IFC thresholds, because the regularization constant to invert the sampling matrix increases towards the low thresholds. Due to this confounding problem, we can expect an increase in error in the reconstruction algorithm. As a result, the plotted result in Fig. 6 is a worst case bound combining two errors, but we could not find other ways of evaluating the comparative results.

While processing of reconstructed signals from pulse trains is not the focus of the paper, substantially higher data rates than corresponding Nyquist rates may be required for applications that necessitates high fidelity after signal reconstruction.

### D. Semantic information processing in ECG

The processing and representation of relevant information in ECG signal using pulse trains is demonstrated in Fig. 7. The top panel of Fig. 7 shows an ECG signal with (corrupted data) and without baseline wander (digital high pass filtered). During baseline deviations, pulses corresponding to cardiac events are obfuscated by higher pulse density due to the shift in baseline as shown in pulse train *X* of Fig. 7. Subsequently, after pulse-based processing to remove baseline wander, the boundaries of relevant ECG features of interest such as P wave, QRS complex, and T wave are clearly delineated in the pulse representation of  $\hat{Z}$ .

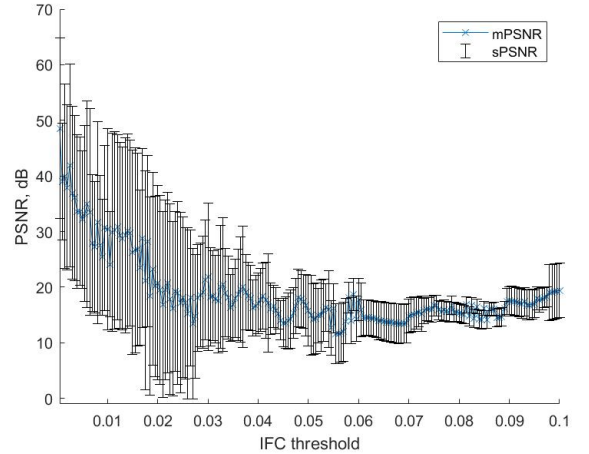


Fig. 6. Comparison of proposed algorithm with digital processing after reconstruction of operand pulse trains

Performance testing determined the sensitivity and positive predictive value of QRS peak detection is 100%, and the absolute error in HR is  $0.16 \pm 0.18$  bpm. Moreover, the data rate of ECG signal is reduced substantially from  $61.31 \pm 100.73$  IFC p/s to  $14.99 \pm 10.14$  IFC p/s after processing. In our previous article [13], we have shown through extensive testing that the accuracy of QRS detection from pulse trains is as good as traditional techniques in the absence of baseline noise, and these results indicate that the performance can be generalized to an end-to-end pulse computation system with a pulse-based preprocessor. Thus,

the proposed pulse-based signal processing can be used in continuous vital sign monitoring systems to represent the semantic information content in ECG signals with a sparse representation.

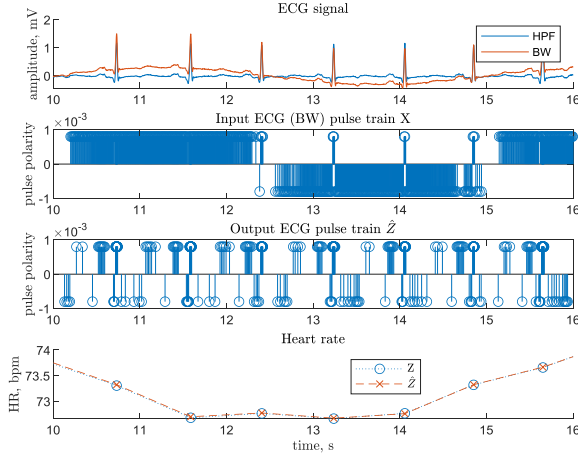


Fig. 7. Semantic processing of ECG signal using pulse trains. 'BW' denotes baseline wander in the signal.

## VII. DISCUSSION

Representation and processing of semantic information in signals is critical for mobile wireless sensor networks and IoT applications [35]. Prior research has shown that the IFC pulse conversion enable sparse representation of features of interest in physiological signals such as ECG [12], [13], neural data [32], and photoplethysmogram [14]. This article presented algorithms for processing the pulse trains created by IFC directly without signal reconstruction, and demonstrated processing of the semantic information in ECG signal.

Simulations with synthetic data show that pulse-based signal processing has PSNR proportional to signal amplitude, with limited pulse representation accuracy near the noise floor. Precision of the pulse-based operations is not uniform over the dynamic range of the signal amplitude, which is desirable in long term monitoring applications to represent semantic information with high precision while not representing with high resolution background noise, which saves data rates.

In DSP, to meet desired performance specifications, the number of samples to which the digital data must be interpolated is determined, and then the operations are performed sample-by-sample. However, in pulse signal processing, the IFC threshold that satisfies the required performance criteria should be determined, by specifying the minimum accuracy in the low amplitude portions of the signal of interest. If this yields high data rates for transmission, then a compromise should be established to meet approximately both specification. Feichtinger et al. [15] showed that bandlimited functions are not completely determined by the IFC and the same results hold true for processing of multiple operands i.e., there are non-zero bandlimited signals that will never produce pulses at the output even though the input operands have pulses. The trade-offs in terms of accuracy for the IFC parameters is studied in [32] and operating ranges are selected based on

specifications to provide the right balance between performance and sparseness.

Hardware implementations of the proposed algorithms require a time marker such as clock signals. Moreover, implementing pulse signal processing in hardware also requires new approaches as operations need to be performed on areas between pulses that occur non-uniformly. Recently, Nallathambi et al. [36], implemented a 16-bit pulse adder based on the approximated algorithms in Table III. Their system, synthesized in SMIC 0.18 $\mu$ m (100MHz) CMOS process, utilizes a clock and time counters as building blocks for a pulse-based ALU. The preliminary results demonstrate the feasibility of signal processing with pulse trains in hardware; however, the digital adder for the same SNR is still much better in terms of area and power, while the pulse multiplier is worse but in the same range of performance as the digital multiplier [41].

The alternative approach to the present work is to reconstruct signals from pulses, perform digital processing and convert the processed signals to pulses. In such scenarios, the accuracy of the operations will be limited by the approximate reconstruction procedure [15], and increasing the fidelity requires reduction of IFC threshold substantially, thereby impacting the data rates. By processing the pulses directly with the proposed algorithm, we circumvent both the complexity of the signal reconstruction algorithm and the subsequent process of IFC conversion. With the proposed algorithms, the focus is on applications where reconstruction is not necessarily the goal but tasks such as classification and anomaly detection in continuous monitoring systems that require representation of semantic features in signal.

In this article, an example of processing semantic information in ECG signals using pulse trains is presented. The data rate of the ECG is less than 20 IFC p/s, which is drastically lower than existing IoT-based cardiac patient monitors that require at least 125 samples per second [37]. Additionally, the accuracy in HR computations typifies the capability of the pulse-based methods in continuous monitoring of real world signals using sparse representations. In general, pulse trains are well suited for processing semantic information in transient data such as biosignals, seismological recordings, radar and others, where the crucial information is in the transients embedded in noisy backgrounds.

## VIII. CONCLUSION

The present work provides an alternative to conventional DSP techniques for performing arithmetic operations on continuous time/amplitude signals using pulse trains generated from an IFC. The IFC is conceptually similar to an asynchronous sigma delta converter, however it does not use oversampling to create an accurate representation in the sample domain. Nevertheless, its output is still in continuous time and creates far fewer pulses that represent the input signal sufficiently to describe properties and perform equivalent signal processing operations in the analog input. This newly discovered region in the precision/data rate plane could not have been anticipated by the theory of sigma delta converters.

We show in this paper for the first time a methodology to perform online arithmetic operations using pulse trains. The results with synthetic and natural data demonstrate the

capability of the algorithms to manipulate signal properties using arithmetic operators. This opens the door to implement linear models in the pulse domain because the convolution operator in the pulse domain was presented.

The sparse representation achieved with the IFC sampler, implies that the arithmetic operations have limited accuracy near the low amplitude regions (noise floor), so we present a methodology to allow users to properly meet design specification. In general, more precision requires a decrease of the IFC threshold  $\theta$  to create more pulses and smaller time between pulses, at the expenses of more computation (more pulses) and higher data rates. This presents a continuum of choices unlike the conventional ADC where the computation load (number of operations per second) is dictated by the algorithm and sampling frequency. In this methodology there is no sampling frequency, so what matters is the time between the pulses for the computation complexity, which is controlled by the threshold and depends also on the local structure of the signals.

The ability of selectively capturing and processing the semantic information in the signal is important in many continuous and event monitoring applications for the IoT and mobile wireless sensor networks [35]. Applications where the goal is detection or classification of vital events and not necessarily signal reconstruction, are ideal for the proposed pulse-based algorithms, which represent the features of interest in the signal while suppressing the background noise. An application of noise reduction in ECG signal is demonstrated with sparse pulse representation while preserving the semantic features of interest.

There are many aspects that require further study, since this is a new field. First, the assumption that the input signal is constant between pulses is an approximation that is only valid for signals that vary slowly within the charge time of the capacitor until the threshold is reached. While this is satisfied in the high amplitude portions of the signal, it is not met in the low amplitude portions of the signal. Therefore, ways to improve this aspect should be pursued. The first step is to define a constraint that the amplitude during the charge time of the capacitor increases linearly instead of being constant. This would improve the accuracy of the overall method, but the implications to the theory and the computational cost in the algorithm is unknown. Alternatively, an equalization preprocessor (such as a log amplifier) may mitigate some of the issues. But, an exponential transformation in the pulse domain would be needed to recover the original signal features. We don't know yet the practicality of creating an accurate exponential warping in time.

#### APPENDIX

*Proof of Theorem 1:* Based on the proposed framework, the addition operation on pulse trains is performed by solving eqn. (2) for  $\mu_S$  and  $t_{s_{n+1}}$ . Assuming  $x(t)$  and  $y(t)$  to be constant between consecutive pulses, we have

$$\begin{aligned} & \int_{t_a}^{t_b} x e^{-\alpha(t_{x_{n+1}}-t)} dt + \int_{t_a}^{t_b} y e^{-\alpha(t_{y_{n+1}}-t)} dt \\ &= \mu_S \int_{t_{s_n}}^{t_{s_{n+1}}} [x+y] e^{-\alpha(t_{s_{n+1}}-t)} dt \end{aligned} \quad (2)$$

Using Observation 1, eqn. (2) is written as  $\eta_S = p_{x_{n+1}} u + p_{y_{n+1}} d$ , where  $u = \frac{g(t_{x_{n+1}}-t_a)-g(t_{x_{n+1}}-t_b)}{g(t_{x_{n+1}}-t_{x_n})}$ ,  $d = \frac{g(t_{y_{n+1}}-t_a)-g(t_{y_{n+1}}-t_b)}{g(t_{y_{n+1}}-t_{y_n})}$ ,  $\mu_S = |\eta_S|$ , and  $p_{s_{n+1}} = \text{sgn}(\eta_S)$ . Moreover,  $y$  is expressed in terms of  $x$  as  $y = \frac{p_{x_{n+1}} x g(t_{x_{n+1}}-t_{x_n})}{p_{y_{n+1}} g(t_{y_{n+1}}-t_{y_n})}$  since  $p_{x_{n+1}} \int_{t_{x_n}}^{t_{x_{n+1}}} x e^{-\alpha(t_{x_{n+1}}-t)} dt = p_{y_{n+1}} \int_{t_{y_n}}^{t_{y_{n+1}}} y e^{-\alpha(t_{y_{n+1}}-t)} dt$ , because the same IFC is used and the integration is over a pulse interval. By substituting  $y$  and  $\mu_S$  in eqn. (2), we obtain

$$g(t_{s_{n+1}} - t_{s_n}) = \frac{K g(t_{x_{n+1}} - t_{x_n}) g(t_{y_{n+1}} - t_{y_n})}{p_{x_{n+1}} g(t_{x_{n+1}} - t_{x_n}) + p_{y_{n+1}} g(t_{y_{n+1}} - t_{y_n})} \quad (3)$$

where  $K = \frac{\text{sgn}(\eta_S) p_{x_{n+1}}}{p_{y_{n+1}}}$ ,  $K \in (1, -1)$ .

Thus, from eqn. (3), the polarity and timing of the sum of the pulse trains is given by  $p_{s_{n+1}} = \text{sgn}(\eta_S)$  and  $t_{s_{n+1}} - t_{s_n} = \frac{-1}{\alpha} \ln \left\{ 1 - \frac{K g(t_{x_{n+1}}-t_{x_n}) g(t_{y_{n+1}}-t_{y_n})}{p_{x_{n+1}} g(t_{x_{n+1}}-t_{x_n}) + p_{y_{n+1}} g(t_{y_{n+1}}-t_{y_n})} \right\}$  respectively.

*Proof of Theorem 2:* Multiplication of pulse trains requires normalization by the identity pulse train  $R$ , which is a periodic pulse train corresponding to  $r(t) = 1$ . Hence, by assuming  $x(t)$  and  $y(t)$  to be constant between consecutive pulses, we have

$$\begin{aligned} & \frac{\int_{t_a}^{t_b} x e^{-\alpha(t_{x_{n+1}}-t)} dt \int_{t_a}^{t_b} y e^{-\alpha(t_{y_{n+1}}-t)} dt}{\int_{t_a}^{t_b} e^{-\alpha(t_{r_{n+1}}-t)} dt} \\ &= \mu_P \int_{t_{p_n}}^{t_{p_{n+1}}} [xy] e^{-\alpha(t_{p_{n+1}}-t)} dt \end{aligned} \quad (4)$$

Using Observation 1, eqn. (4) is written as  $\eta_P = \frac{(p_{x_{n+1}} u)(p_{y_{n+1}} d)}{p_{r_{n+1}} r}$ , where  $u = \frac{g(t_{x_{n+1}}-t_a)-g(t_{x_{n+1}}-t_b)}{g(t_{x_{n+1}}-t_{x_n})}$ ,  $d = \frac{g(t_{y_{n+1}}-t_a)-g(t_{y_{n+1}}-t_b)}{g(t_{y_{n+1}}-t_{y_n})}$ ,  $r = \frac{g(t_{r_{n+1}}-t_a)-g(t_{r_{n+1}}-t_b)}{g(t_{r_{n+1}}-t_{r_n})}$ ,  $\mu_P = |\eta_P|$  and  $p_{p_{n+1}} = \text{sgn}(\eta_P)$ .

Substituting  $\mu_P$  in eqn. (4), we obtain

$$g(t_{p_{n+1}} - t_{p_n}) = \frac{g(t_{x_{n+1}} - t_{x_n}) g(t_{y_{n+1}} - t_{y_n})}{g(t_{r_{n+1}} - t_{r_n})} \quad (5)$$

Thus, from eqn. (5), the polarity and timing of the product of the pulse trains is given by  $p_{p_{n+1}} = \text{sgn}(\eta_P)$  and  $t_{p_{n+1}} - t_{p_n} = \frac{-1}{\alpha} \ln \left\{ 1 - \frac{g(t_{x_{n+1}}-t_{x_n}) g(t_{y_{n+1}}-t_{y_n})}{g(t_{r_{n+1}}-t_{r_n})} \right\}$  respectively.

*Proof of Theorem 3:* Convolution of pulse trains requires convolution of underlying areas and hence, we have

$$\begin{aligned} & \frac{\int_{t_a}^{t_b} x e^{-\alpha(t_{x_{n+1}}-t)} dt \otimes \int_{t_a}^{t_b} y e^{-\alpha(t_{y_{n+1}}-t)} dt}{\int_{t_a}^{t_b} e^{-\alpha(t_{r_{n+1}}-t)} dt} \\ &= \mu_C \int_{t_{c_n}}^{t_{c_{n+1}}} [x \otimes y] e^{-\alpha(t_{c_{n+1}}-t)} dt \end{aligned} \quad (6)$$

Eqn. (6) is expressed as follows:

$$\begin{aligned}
& \frac{\int_{t_a}^{t_b} e^{-\alpha(t_{x_{n+1}}-\tau)} d\tau}{\int_{t_a}^{t_b} e^{-\alpha(t_{r_{n+1}}-\tau)} d\tau} \left[ \frac{1}{\alpha} \int_{\lambda=\lambda_1}^{\lambda_2} \left( e^{-\alpha(t_{y_{n+1}}-t_b+\lambda)} \right. \right. \\
& \left. \left. - e^{-\alpha(t_{y_{n+1}}-t_b+\lambda)} \right) d\lambda \right] \\
& = \mu_C \int_{t_{c_n}}^{t_{c_{n+1}}} \left[ \int_{\beta=\beta_1}^{\beta_2} d\beta \right] e^{-\alpha(t_{c_{n+1}}-t)} dt
\end{aligned} \quad (7)$$

Using Observation 1, the above equation is written as  $\eta_C = \frac{(p_{x_{n+1}}u)(p_{y_{n+1}}d)}{\alpha p_{r_{n+1}}r}$ , where  $u = \frac{g(t_{x_{n+1}}-t_a)-g(t_{x_{n+1}}-t_b)}{g(t_{x_{n+1}}-t_{x_n})}$ ,  $r = \frac{g(t_{r_{n+1}}-t_a)-g(t_{r_{n+1}}-t_b)}{g(t_{r_{n+1}}-t_{r_n})}$ ,  $\mu_C = |\eta_C|$ ,  $p_{c_{n+1}} = \text{sgn}(\eta_C)$ ,  $d = \frac{g(t_{y_{n+1}}-t_a+\lambda_2)-g(t_{y_{n+1}}-t_b+\lambda_2)-g(t_{y_{n+1}}-t_a+\lambda_1)+g(t_{y_{n+1}}-t_b+\lambda_1)}{g(t_{y_{n+1}}-t_{y_n})}$ .

By substituting  $\mu_C$  in eqn. (7), we obtain

$$g(t_{c_{n+1}} - t_{c_n}) = \frac{g(t_{x_{n+1}} - t_{x_n})g(t_{y_{n+1}} - t_{y_n})}{(\beta_2 - \beta_1)g(t_{r_{n+1}} - t_{r_n})} \quad (8)$$

Thus, from eqn. (8), the polarity and timing of the product of the pulse trains is given by  $p_{c_{n+1}} = \text{sgn}(\eta_C)$  and  $t_{c_{n+1}} -$

$$t_{c_n} = \frac{-1}{\alpha} \ln \left\{ 1 - \frac{g(t_{x_{n+1}}-t_{x_n})g(t_{y_{n+1}}-t_{y_n})}{(\beta_2-\beta_1)g(t_{r_{n+1}}-t_{r_n})} \right\} \text{ respectively.}$$

**Error bounds for pulse arithmetic:** The assumption that  $x(t)$  is a constant  $x$  between  $t_a$  and  $t_b$  in eqns. (2), (4), and (6) yields an error  $\delta_X$  in the area of the pulse train  $X$  generated by IFC, where  $\delta_X = \int_{t_a}^{t_b} [x(t) - x] e^{-\alpha(t_{x_{n+1}}-t)} dt$ . From the mean value theorem [38], we have  $\delta_X = [x(z_1) - x] e^{-\alpha(t_{x_{n+1}}-z_1)} (t_b - t_a)$  where  $t_a \leq z_1 \leq t_b$ . This quantity of error is bounded by  $|t_b - t_a| \left| \frac{x(z_1)-x}{e^{\alpha(t_{x_{n+1}}-t_a)}} \right| \leq |\delta_X| = |t_b - t_a| \left| \frac{x(z_1)-x}{e^{\alpha(t_{x_{n+1}}-z_1)}} \right| \leq |t_b - t_a| \left| \frac{x(z_1)-x}{e^{\alpha(t_{x_{n+1}}-t_b)}} \right|$  since  $e^{-t}$  is a monotonically decreasing

function for  $t > 0$ . So, we have  $\frac{\min_{z \in [t_a, t_b]} |x(z)-x|}{e^{\alpha(t_{k_{n+1}}-t_a)}} \leq \frac{|\delta_X|}{(t_b-t_a)} \leq \frac{\max_{z \in [t_a, t_b]} |x(z)-x|}{e^{\alpha(t_{k_{n+1}}-t_b)}}$ . Hence, the error in each interval depends on the length of the interval  $(t_b - t_a)$  and the deviation of  $x(t)$  from  $x$  in  $[t_a, t_b]$ .

The error  $\delta_{X+Y}$  resulting from the addition of two pulses, with the assumption of  $x(t) = x$  and  $y(t) = y$  between consecutive pulses, is given by the quadrature formula [39],  $\delta_{X+Y} = \{(\delta_X)^2 + (\delta_Y)^2\}^{\frac{1}{2}}$  and bounded by  $\frac{\min_{z \in [t_a, t_b]} (x(z)-x)^2}{e^{2\alpha(t_{x_{n+1}}-t_a)}} + \frac{\min_{z \in [t_a, t_b]} (y(z)-y)^2}{e^{2\alpha(t_{y_{n+1}}-t_a)}} \leq \frac{\delta_{X+Y}^2}{(t_b-t_a)^2} \leq \frac{\max_{z \in [t_a, t_b]} (x(z)-x)^2}{e^{2\alpha(t_{x_{n+1}}-t_b)}} + \frac{\max_{z \in [t_a, t_b]} (y(z)-y)^2}{e^{2\alpha(t_{y_{n+1}}-t_b)}}$ .

$$\frac{\min_{z \in [t_a, t_b]} (x(z)-x)^2}{e^{2\alpha(t_{x_{n+1}}-t_a)}} + \frac{\min_{z \in [t_a, t_b]} (y(z)-y)^2}{e^{2\alpha(t_{y_{n+1}}-t_a)}} \leq \frac{\delta_{X+Y}^2}{(t_b-t_a)^2} \leq \frac{\max_{z \in [t_a, t_b]} (x(z)-x)^2}{e^{2\alpha(t_{x_{n+1}}-t_b)}} + \frac{\max_{z \in [t_a, t_b]} (y(z)-y)^2}{e^{2\alpha(t_{y_{n+1}}-t_b)}}.$$

The error  $\delta_{XY}$  due to the product of two pulses, with the assumption of  $x(t) = x$  and  $y(t) = y$  between consecutive pulses, is given by the formula [39],  $\delta_{XY} = \left| \frac{K_1 K_2 \theta}{K_3} \right| \left[ \left( \frac{\delta_X}{K_1 \theta} \right)^2 + \left( \frac{\delta_Y}{K_2 \theta} \right)^2 \right]^{\frac{1}{2}}$  where  $\int_{t_a}^{t_b} x(t) e^{-\alpha(t_{x_{n+1}}-t)} dt = K_1 \theta$ ,  $\int_{t_a}^{t_b} y(t) e^{-\alpha(t_{y_{n+1}}-t)} dt = K_2 \theta$ ,  $\int_{t_a}^{t_b} e^{-\alpha(t_{r_{n+1}}-t)} dt = K_3 \theta$ , and

$$\begin{aligned}
& \text{bounded by } \frac{K_2^2 \min_{z \in [t_a, t_b]} (x(z)-x)^2}{e^{2\alpha(t_{x_{n+1}}-t_a)}} + \frac{K_1^2 \min_{z \in [t_a, t_b]} (y(z)-y)^2}{e^{2\alpha(t_{y_{n+1}}-t_a)}} \leq \\
& \left( \frac{K_3}{t_b-t_a} \right)^2 \delta_{XY}^2 \leq \frac{K_2^2 \max_{z \in [t_a, t_b]} (x(z)-x)^2}{e^{2\alpha(t_{x_{n+1}}-t_b)}} + \frac{K_1^2 \max_{z \in [t_a, t_b]} (y(z)-y)^2}{e^{2\alpha(t_{y_{n+1}}-t_b)}}.
\end{aligned}$$

Likewise, it is straight-forward to show that error  $\delta_{X \otimes Y}$  due to the convolution of two overlapping pulse segments, with the assumption of  $x(t) = x$  and  $y(t) = y$  between consecutive

$$\begin{aligned}
& \text{pulses, is bounded by } \frac{K_2^2 \min_{z \in [t_a, t_b]} (x(z)-x)^2}{e^{2\alpha(t_{x_{n+1}}-t_a)}} + \\
& K_1^2 \frac{\min_{z' \in [\lambda_1-t_b, \lambda_2-t_a]} (y(z')-y)^2}{e^{2\alpha(t_{y_{n+1}}-(\lambda_1-t_b))}} \leq \left( \frac{K_3}{(t_b-t_a)(\lambda_2-\lambda_1)} \right)^2 \delta_{X \otimes Y}^2 \leq \\
& K_2^2 \frac{\max_{z \in [t_a, t_b]} (x(z)-x)^2}{e^{2\alpha(t_{x_{n+1}}-t_b)}} + K_1^2 \frac{\max_{z' \in [\lambda_1-t_b, \lambda_2-t_a]} (y(z')-y)^2}{e^{2\alpha(t_{y_{n+1}}-(\lambda_2-t_a))}}.
\end{aligned}$$

Notice that these errors refers to a local error within a pulse interval. The global mean error during a segment of a signal is the sum of local errors divided by the number of pulse intervals in that segment. However, this number depends upon the structure of the input signal, and the values of IFC parameters.

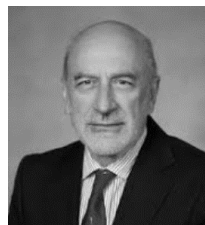
## REFERENCES

- [1] E. T. Whittaker, "On the functions which are represented by the expansion of the interpolation-theory," *Proc. Edinburgh Math. Soc.*, vol. 35, no. Sect. A, pp. 181–194, 1915.
- [2] C. E. Shannon, "Communication in the presence of noise," *Proc. IRE*, vol. 37, no. 2, pp. 10–21, 1949.
- [3] H. Nyquist, "Certain topics in telegraph transmission theory," *Proc. IEEE*, vol. 90, no. 2, pp. 280–305, 2002.
- [4] M. Unser, "Sampling-50 years after Shannon," *Proc. IEEE*, vol. 88, no. 4, pp. 569–587, Apr. 2000.
- [5] R. G. Baraniuk, "More Is Less: Signal Processing and the Data Deluge," *Science (80-. )*, vol. 331, no. 6018, 2011.
- [6] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, 2006.
- [7] M. Vetterli, P. Marziliano, and T. Blu, "Sampling signals with finite rate of innovation," *IEEE Trans. Signal Process.*, vol. 50, no. 6, pp. 1417–1428, Jun. 2002.
- [8] A. A. Lazar and L. T. Toth, "Perfect Recovery and Sensitivity Analysis of Time Encoded Bandlimited Signals," *IEEE Trans. Circuits Syst. Regul. Pap.*, vol. 51, no. 10, pp. 2060–2073, 2004.
- [9] A. A. Lazar, "Time Encoding with an Integrate-and-Fire Neuron with a Refractory Period," *Neurocomputing*, vol. 58–60, pp. 53–58, 2004.
- [10] M. Rastogi, A. Singh Alvarado, J. G. Harris, and J. C. Principe, "Integrate and fire circuit as an ADC replacement," in *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, 2011, no. 1, pp. 2421–2424.
- [11] L. . Abbott, "Lapicque's introduction of the integrate-and-fire model neuron (1907)," *Brain Res. Bull.*, vol. 50, no. 5–6, pp. 303–304, Nov. 1999.
- [12] A. Singh Alvarado, C. Lakshminarayan, and J. C. Principe, "Time-based compression and classification of heartbeats," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 6, pp. 1641–8, Jun. 2012.
- [13] G. Nallathambi and J. C. Principe, "Integrate and fire pulse train automaton for QRS detection," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 2, pp. 317–26, Feb. 2014.

- [14] G. Nallathambi, J. Principe, and N. Euliano, "Pulse based signal processing for systolic peak recognition," in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2015, pp. 1–6.
- [15] H. G. Feichtinger, J. C. Principe, J. L. Romero, A. Singh Alvarado, and G. A. Velasco, "Approximate reconstruction of bandlimited functions for the integrate and fire sampler," *Adv. Comput. Math.*, vol. 36, no. 1, pp. 67–78, Sep. 2011.
- [16] M. McCormick, "Digital pulse processing," MIT, 2012.
- [17] D. J. Daley and D. Vere-Jones, *An introduction to the theory of Point Processes*. 2003.
- [18] S. Mitra, S. Fusi, and I. Giacomo, "Real-time classification of complex patterns using spike-based learning in neuromorphic VLSI," *IEEE Trans. Biomed. Circuits Syst.*, vol. 3, no. 1, pp. 32–42, 2009.
- [19] W. Maass, "Computing with spikes," *Special Issue on Foundations of Information Processing of TELEMATIK 8.1*, pp. 32–36, 2002.
- [20] G. Nallathambi, "Pulse signal processing," 2017. [Online]. Available: <https://github.com/gabrielnallathambi>. [Accessed: 18-Sep-2017].
- [21] A. F. Murray, "Pulse arithmetic in VLSI neural networks," *IEEE Micro*, vol. 9, no. 6, pp. 64–74, 1989.
- [22] A. Murray, "Pulse techniques in neural VLSI: a review," *[Proceedings] 1992 IEEE Int. Symp. Circuits Syst.*, vol. 5, pp. 2204–2207, 1992.
- [23] A. F. Murray, D. Del Corso, and L. Tarassenko, "Pulse-Stream VLSI Neural Networks Mixing Analog and Digital Techniques," *IEEE Trans. Neural Networks*, vol. 2, no. 2, pp. 193–204, 1991.
- [24] S. Churcher, D. J. Baxter, A. Hamilton, A. F. Murray, and H. M. Reekie, "Generic Analog Neural Computation - The Epsilon Chip," *Adv. Neural Inf. Process. Syst.* 5, pp. 773–780, 1993.
- [25] A. A. Lazar, "Multichannel time encoding with integrate-and-fire neurons," *Neurocomputing*, vol. 65–66, pp. 401–407, 2005.
- [26] Tim Sauer, "System identification for chaotic integrate-and-fire dynamics," *Int. J. Intell. Syst.*, vol. 12, no. 4, pp. 255–265, 1997.
- [27] A. R. C. Paiva, I. Park, and J. C. Principe, "A reproducing kernel Hilbert space framework for spike train signal processing," *Neural Comput.*, vol. 21, no. 2, pp. 424–449, 2009.
- [28] F. Rieke, D. Warland, R. Steveninck, and W. Bialek, *Spikes: exploring the neural code*. Cambridge, MA, USA: MIT Press, 1999.
- [29] M. Miura, M. Goishi, N. Chiba, and J. Shida, "A magnetic neural network utilizing universal arithmetic modules for pulse-train signal processing," *[Proceedings 1992] IEEE Int. Conf. Syst. Eng.*, pp. 499–502, 1992.
- [30] A. Hamilton and K. Papathanasiou, "Reconfigurable analogue systems: The pulse-based approach," *IEE Proc. - Comput. Digit. Tech.*, vol. 147, no. 3, p. 203, 2000.
- [31] K. Papathanasiou and A. Hamilton, "Pulse based signal processing: VLSI implementation of a Palmo filter," in *1996 IEEE International Symposium on Circuits and Systems. Circuits and Systems Connecting the World. ISCAS 96*, 1996, vol. 1, pp. 270–273.
- [32] A. S. Alvarado, "Time encoded compression and classification using the integrate and fire sampler," University of Florida, 2012.
- [33] R. E. Gregg, S. H. Zhou, J. M. Lindauer, E. D. Helfenbein, and K. K. Giuliano, "What is inside the electrocardiograph?," *J. Electrocardiol.*, vol. 41, no. 1, pp. 8–14, 2008.
- [34] A. L. Goldberger *et al.*, "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. E215–E220, 2000.
- [35] K. Majumdar and S. Jayachandran, "Semantic Information Encoding in One Dimensional Time Domain Signals," pp. 1–24, Nov. 2016.
- [36] G. Nallathambi, L. Rao, and J. Principe, "Hardware implementation of adder for pulse signal processing," in *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2017, pp. 1330–1334.
- [37] A. M. Chan, N. Selvaraj, N. Ferdosi, and R. Narasimhan, "Wireless patch sensor for remote monitoring of heart rate, respiration, activity, and falls," *Conf. Proc. ... Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE Eng. Med. Biol. Soc. Annu. Conf.*, vol. 2013, pp. 6115–6118, 2013.
- [38] M. Comenetz, *Calculus*. WORLD SCIENTIFIC, 2002.
- [39] H. H. Ku, "Notes on the use of propagation of error formulas," *J. Res. Natl. Bur. Stand. Sect. C Eng. Instrum.*, vol. 70C, no. 4, p. 263, 1966.



**Gabriel Nallathambi** received the Ph.D. degree in electrical engineering from the University of Florida in 2015. He is currently the Senior Biomedical Engineer at VitalConnect Inc., California. His research interests include sensor design, signal processing, and machine learning with application to physiological monitoring.



**Jose C. Principe** (M'83-SM'90-F'00) is the Bell-South and Distinguished Professor of Electrical and Biomedical Engineering at the University of Florida, and the Founding Director of the Computational NeuroEngineering Laboratory (CNEL). His primary research interests are in advanced signal processing with information theoretic criteria and adaptive models in reproducing kernel Hilbert spaces (RKHS), with application to brain-machine interfaces (BMIs). Dr. Principe is a Fellow of the IEEE, ABME, and AIBME. He is the past Editor in Chief of the IEEE Transactions on Biomedical Engineering, past Chair of the Technical Committee on Neural Networks of the IEEE Signal Processing Society, Past-President of the International Neural Network Society, and a recipient of the IEEE EMBS Career Award and the IEEE Neural Network Pioneer Award.