This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TSP.2020.3042005

On the Solvability of the Peak Value Problem for Bandlimited Signals With Applications

Holger Boche, *Fellow, IEEE,* and Ullrich J. Mönich, *Senior Member, IEEE*

***Abstract*—In this paper we study from an algorithmic perspective the problem of finding the peak value of a bandlimited signal. This problem plays an important role in the design and optimization of communication systems. We show that the peak value problem, i.e., computing the peak value of a bandlimited signal from its samples, can be solved algorithmically if oversampling is used. Without oversampling this is not possible. There exist bandlimited signals, for which the sequence of samples is computable, but the signal itself is not. This problem is directly related to the question whether there is a link between computability in the digital domain and the analog domain, and hence to a fundamental signal processing problem. We show that there is an asymmetry between continuous-time and discrete-time computability. Further, we study the decay behavior of computable bandlimited signals, which describes the concentration of the signals in the time domain, and, for locally computable bandlimited signals, we analyze if it is always possible to decide algorithmically whether the peak value is smaller than a given threshold.**

***Index Terms*—Peak value, decay behavior, effective approximation, algorithm, computability**

I. Introduction

The peak value of a signal is a distinguished quantity, with relevance for many applications. For example, in communication systems that employ orthogonal frequency division multiplexing (OFDM) large peak-to-average power ratios (PAPRs), and hence large peak values, are problematic, because they can overload amplifiers, which in turn leads to undesired out-of-band radiation and distorted signals [2]–[4]. Thus, a control of the peak value is essential. Numerous papers analyzed the PAPR [5], [6], and several methods to reduce the PAPR have been proposed [7]–[12].

The PAPR control is important not only for power amplifiers in OFDM systems, but also for base stations and terminals in other modern communication systems. They key problem is that the power amplifiers have only a limited linear range, and the goal is to optimally utilize this range. For signals with constant envelop this can be easily achieved. However, for broadband signals the situation is more complicated. Those signals can have large PAPRs, which lead to out-of-band radiation that has to be suppressed by expensive analog filters.

In modern signal processing applications, often the signals are directly created in the digital domain and later converted into the analog domain. In the communications example discussed above, this would be the digital baseband signal that later is converted into an analog signal for the actual transmission over an antenna. Bandlimited signals are a suitable model for transmit signals in communication systems. In order to avoid large PAPRs, it is necessary to control the peak value of the continuous-time signal. One approach is to compute the peak value of the continuous-time signal, and, if it is too high, suitable correction algorithms are applied. Often, the correction algorithms are implemented in the digital domain. In order to make them work and to assess their effectiveness, it is essential that the peak value of the continuous-time signal can be determined from the discrete-time signal, i.e., from the samples of the signal. In this paper we study if this can be done algorithmically on a digital computer. To study this question, we employ the concept of Turing computability [13], [14]. We will come back to the peak value problem in Section IV, where we also discuss existing results. However, to the best of our knowledge, none of the results consider questions of computability.

A Turing machine is an abstract device that manipulates symbols on a strip of tape according to certain rules. Although the concept is very simple, a Turing machine is capable of simulating any given algorithm [15], [16]. Turing machines have no limitations in terms of memory or computing time, and hence provide a theoretical model that describes the fundamental limits of any practically realizable digital computer.

Computability is a mature topic in computer sciences [15]–[18], and one of the key concepts of this theory is the effective, i.e., algorithmic control of the approximation error. In the signal processing literature, however, this aspect has not gotten much attention so far. Recently, some observations about signal processing operations have been made, where computability problems can occur [19]–[23].

We show that with oversampling, the peak value of a bandlimited signal can be computed algorithmically from the samples of the signal, and give an algorithm for the computation (Theorem 1). In contrast, without oversampling the peak value of a bandlimited signal is not always computable (Theorem 5). As a consequence, the peak value of these critical signals cannot be determined using digital hardware such as DSPs, FPGAs, or CPUs. Since our signal model is very general—we assume only bandlimitedness—it does not

This work was supported by the Gottfried Wilhelm Leibniz Programme of the German Research Foundation (DFG) under grant BO 1734/20-1, the DFG under Germany's Excellence Strategy – EXC-2111 – 390814868, and the German Ministry of Education and Research (BMBF) within the project NewCom under grant 16KIS1003K.

Parts of this paper were presented at the 2020 IEEE International Conference on Acoustics, Speech, and Signal Processing [1].

Holger Boche is with the Technische Universität München, Lehrstuhl für Theoretische Informationstechnik, Germany, and the Munich Center for Quantum Science and Technology (MCQST), Schellingstr. 4, Germany. e-mail: boche@tum.de. Ullrich J. Mönich is with the Technische Universität München, Lehrstuhl für Theoretische Informationstechnik, Germany. e-mail: moenich@tum.de.

only apply to OFDM signals, but also to other more general waveforms, for example, those which have been proposed for 5G wireless systems [24].

Our analyses address a general question: when is it possible to infer properties of the continuous-time signal, such as the peak value, from the properties of the corresponding discrete-time signal? Such a relation would be important for all signal processing applications that link both domains. We will see that, in contrast to other signal properties, such as the signal energy, there is a difference between both domains with respect to the computability of the peak value (Theorem 3). A further property that we study is the signal's decay behavior. For finite energy signals it is a typical and justified assumption that the signal energy is concentrated in a certain time interval, and that the energy outside this interval is negligible for applications. Further, for finite energy signals, it is possible to algorithmically determine this interval. We analyze if the same is true for the peak value of the signal, i.e., if it is always possible to compute a time interval such that outside of this interval the peak value of the signal is below some threshold (Corollary 1 and Theorem 6).

The structure of this paper is as follows. After the introduction of the necessary notation in Section II, we present the basic definitions and concepts of computability in Section III. The peak value problem is further motivated in Section IV. In Section V we prove our first result, the computability of the peak value of a computable bandlimited signal if oversampling is used. The decay behavior for the oversampling case is studied in Section VI. Then, in Section VII we analyze critical sampling at Nyquist rate and show that in this case the peak value cannot always be computed. The decay behavior for this case is studied in Section VIII. Further, in Section IX the special case of locally computable signals is investigated. Semi-decidability for certain relevant signal sets and the connection to exit flags is analyzed in Section X, before we conclude the paper in Section XI.

II. Notation

By c_0 we denote the set of all sequences that vanish at infinity. For $\Omega \subset \mathbb{R}$, let $L^p(\Omega)$, $1 \leq p < \infty$, be the space of all measurable, pth-power Lebesgue integrable functions on Ω, with the usual norm $\|\cdot\|_p$, and $L^\infty(\Omega)$ the space of all functions for which the essential supremum norm $\|\cdot\|_\infty$ is finite. The Bernstein space $\mathcal{B}_\sigma^p$, $\sigma > 0$, $1 \leq p \leq \infty$, consists of all entire functions of exponential type at most σ, whose restriction to the real line is in $L^p(\mathbb{R})$ [25, p. 49]. The norm for $\mathcal{B}_\sigma^p$ is given by the L^p-norm on the real line. A function in $\mathcal{B}_\sigma^p$ is called bandlimited to σ. $\mathcal{B}_{\sigma,0}^\infty$ denotes the space of all functions in $\mathcal{B}_\sigma^\infty$ that vanish on the real line at infinity. For a function f and $a > 0$, we set $\mathbb{Z}/a = \{k/a\}_{k\in\mathbb{Z}}$ and denote by $f|_{\mathbb{Z}/a}$ the sequence $\{f(k/a)\}_{k\in\mathbb{Z}}$, which is the restriction of f to the set $\mathbb{Z}/a$.

III. Computability

The theory of computability is a well-established field in computer sciences [13]–[18]. Alan Turing introduced the concept of a computable real number in [13], [14]. A sequence of rational numbers $\{r_n\}_{n\in\mathbb{N}}$ is called computable sequence if there exist recursive functions a, b, s from $\mathbb{N}$ to $\mathbb{N}$ such that $b(n) \neq 0$ for all $n \in \mathbb{N}$ and $r_n = (-1)^{s(n)} a(n)/b(n)$, $n \in \mathbb{N}$. A recursive function is a function, mapping natural numbers into natural numbers, that is built of simple computable functions and recursions [26]. Recursive functions are computable by a Turing machine.

A set $A \subset \mathbb{N}$ is called recursively enumerable if $A = \emptyset$ or A is the range of a recursive function. A set $A \subset \mathbb{N}$ is called recursive if both A and $\mathbb{N}\setminus A$ are recursively enumerable. The fact that there exist sets which are recursively enumerable but not recursive will be important for us [17, p. 7, Proposition A], [26, p. 18].

A real number x is said to be computable if there exists a computable sequence of rational numbers $\{r_n\}_{n\in\mathbb{N}}$ and a recursive function $\xi\colon \mathbb{N} \to \mathbb{N}$ such that, for all $M \in \mathbb{N}$, we have $|x - r_n| \leq 2^{-M}$ for all $n \geq \xi(M)$. This form of convergence with a computable control of the approximation error is called effective convergence.

Example 1. We give an example that illustrates that this kind of an effective control of the approximation error is essential for the computation of even very simple real-world problems. Consider the discharge behavior of an RC circuit, which is mathematically described by

$$u(t) = \mathrm{e}^{-\frac{t}{RC}}, \quad t > 0, \quad u(0) = 1, \tag{1}$$

where R is the resistance, C the capacitance, and $u(t)$ is the voltage at the capacitor. A digital computer can only handle rational numbers exactly. Hence, we assume that R and C are rationals. According to the Lindemann–Weierstrass theorem, e^x is a transcendental number for every rational x. Hence, if we want to compute the voltage $u(t)$ at a rational time instant t then the result will be a transcendental number which has to be approximated by the digital computer. This computed approximation is only meaningful if the approximation error can be effectively controlled.

Note that the exponential function in (1) is an entire function of exponential type and therefore a bandlimited function. Even for this simple function a Turing machine is not able to compute the function values for $t \neq 0$ exactly, but only approximations. This behavior is generic for entire functions that are no polynomials.

We call a sequence of real numbers $\{x_n\}_{n\in\mathbb{N}} \subset \mathbb{R}$, a computable sequence if there exists a computable double sequence of rationals $\{r_{n,m}\}_{n,m\in\mathbb{N}}$ and a recursive function $\xi\colon \mathbb{N}\times\mathbb{N} \to \mathbb{N}$ such that, for all $M \in \mathbb{N}$ and $n \in \mathbb{N}$, we have $|x_n - r_{n,m}| \leq 2^{-M}$ for all $m \geq \xi(M,n)$. Note that if a computable sequence of real numbers $\{x_n\}_{n\in\mathbb{N}}$ converges effectively to a limit x, then x is a computable real number [17, p. 20, Corollary 2a]. A non-computable real number was for example constructed in [27]. By $\mathbb{R}_c$ we denote the set of computable real numbers and by $\mathbb{C}_c = \mathbb{R}_c + i\mathbb{R}_c$ the set of computable complex numbers.

A sequence $\{x(k)\}_{k\in\mathbb{Z}}$ in c_0 is called computable in c_0 if every number $x(k)$, $k \in \mathbb{Z}$, is computable and there exist a computable sequence $\{x_n\}_{n\in\mathbb{N}} \subset c_0$, where each x_n has only finitely many non-zero elements, and a recursive function

$\xi\colon \mathbb{N} \to \mathbb{N}$, such that for all $M \in \mathbb{N}$ we have $\|x - x_n\|_{\ell^\infty} \leq 2^{-M}$ for all $n \geq \xi(M)$. By $\mathcal{C}c_0$ we denote the set of all sequences that are computable in c_0.

There are several—not equivalent—definitions of computable functions, most notably, computable continuous functions, Turing computable functions, Markov computable functions, and Banach–Mazur computable functions [18]. A function that is computable with respect to any of the above definitions, has the property that it maps computable numbers into computable numbers.

We now give the definition of a computable continuous function [17, p. 25, Definition A(ii)]. Let $I \subset \mathbb{R}$ be an interval, where the endpoints are computable real numbers. A function $f\colon I \to \mathbb{R}$ is called a computable continuous function if

1) f maps every computable sequence $\{t_n\}_{n\in\mathbb{N}} \subset I$ into a computable sequence $\{f(t_n)\}_{n\in\mathbb{N}}$ of real numbers.
2) there exists a recursive function $d\colon \mathbb{N} \to \mathbb{N}$ such that for all $t_1, t_2 \in I$ and all $M \in \mathbb{N}$ we have: $|t_1 - t_2| \leq 1/d(M)$ implies $|f(t_1) - f(t_2)| \leq 2^{-M}$.

Next, we extend this definition to functions defined on $\mathbb{R}$. A function $f\colon \mathbb{R} \to \mathbb{R}$ is a called computable continuous function if

1) f maps every computable sequence $\{t_n\}_{n\in\mathbb{N}} \subset \mathbb{R}$ into a computable sequence $\{f(t_n)\}_{n\in\mathbb{N}}$ of real numbers.
2) there exists a recursive function $d\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that for all $L, M \in \mathbb{N}$ we have: $|t_1 - t_2| \leq 1/d(L, M)$ implies $|f(t_1) - f(t_2)| \leq 2^{-M}$ for all $t_1, t_2 \in [-L, L]$.

A weaker form of computability is Banach–Mazur computability. A function $f\colon \mathbb{R} \to \mathbb{R}$ is called Banach–Mazur computable if it maps every computable sequence $\{t_n\}_{n\in\mathbb{N}} \subset \mathbb{R}$ into a computable sequence $\{f(t_n)\}_{n\in\mathbb{N}}$ of real numbers, i.e., if it satisfies condition 1) of the definition of a computable continuous function. We can generalize the definition of Banach–Mazur computability to more general mappings. Let $\mathcal{M}$ be some set of computable functions. We call a mapping $\psi\colon \mathcal{M} \to \mathbb{R}$ Banach–Mazur computable if it maps every computable sequence $\{f_n\}_{n\in\mathbb{N}} \subset \mathcal{M}$ into a computable sequence $\{\psi(f_n)\}_{n\in\mathbb{N}}$ of real numbers.

In addition to the above mentioned definitions of computability, we introduce a definition for computable functions in Banach spaces, which is based on effective convergence. We call a function f elementary computable if there exists a natural number L and a sequence of computable numbers $\{\alpha_k\}_{k=-L}^{L}$ such that

$$f(t) = \sum_{k=-L}^{L} \alpha_k \frac{\sin(\pi(t-k))}{\pi(t-k)}. \tag{2}$$

Note that every elementary computable function f is a finite sum of computable functions and hence computable. As a consequence, for every $t \in \mathbb{R}_c$ the number $f(t)$ is computable. Further, the sum of finitely many elementary computable functions is computable, as well as the product of an elementary computable function with a computable number.

A signal $f \in \mathcal{B}_{\pi,0}^\infty$ is called computable in $\mathcal{B}_{\pi,0}^\infty$, if there exist a computable sequence of elementary computable functions $\{f_N\}_{N\in\mathbb{N}}$ and a recursive function $\xi\colon \mathbb{N} \to \mathbb{N}$ such that, for all $M \in \mathbb{N}$, we have $\|f - f_N\|_\infty \leq 2^{-M}$ for all $N \geq \xi(M)$. By $\mathcal{CB}_{\pi,0}^\infty$ we denote the set of all functions in $\mathcal{B}_{\pi,0}^\infty$ that are computable in $\mathcal{B}_{\pi,0}^\infty$. According to this definition, we can approximate any signal $f \in \mathcal{CB}_{\pi,0}^\infty$ by an elementary computable function, where we have an "effective" control of the approximation error, as illustrated in Fig 1.

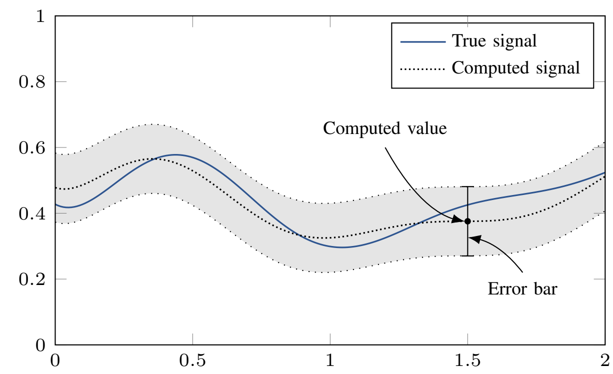

Fig. 1. For a computable signal we can always determine an error bar and then can be sure that the true value lies within the specified error range.

Remark 1. Since for every elementary computable function f_N the norm $\|f_N\|_\infty$ is computable, it follows from the inequality $|\|f\|_\infty - \|f_N\|_\infty| \leq \|f - f_N\|_\infty$ that the norm $\|f\|_\infty$, i.e., the maximum of f, is computable for all $f \in \mathcal{CB}_{\pi,0}^\infty$.

Remark 2. If $f \in \mathcal{CB}_{\pi,0}^\infty$ then f is also a computable continuous function according to the definition using the effective uniform continuity, because $|f(t_1) - f(t_2)| \leq \|f'\|_\infty |t_1 - t_2| \leq \pi \|f\|_\infty |t_1 - t_2|$, and $\|f\|_\infty$ is computable.

IV. The Peak Value Problem

The peak value problem that we study in this paper can be summarized as follows. Given a continuous-time bandlimited signal $f \in \mathcal{B}_{\pi,0}^\infty$, if we know f on a discrete set such as $\mathbb{Z}/a$, $a \geq 1$, i.e., if we know $f|_{\mathbb{Z}/a}$, can we determine the peak value of f, i.e., $\|f\|_\infty$, or can we at least find an upper bound for $\|f\|_\infty$?

The peak value problem is relevant, because in many applications the continuous-time signal f is not known, but only the discrete-time samples $f|_{\mathbb{Z}/a}$. This is, for example, the case in mobile communications, where we create a discrete-time complex baseband signals, and need to control the peak value of the corresponding continuous-time signal.

Important questions related to the peak value problem are:

1) Can we determine the peak value (or an upper bound) of the continuous-time signal from the discrete-time signal?
2) Can we compute the peak value on a digital computer?
3) Which role plays the oversampling factor a?

The peak value problem for equidistant sampling was studied, for example, in [28], [29], and for non-equidistant sampling in [30]. Further, results for the special cases of OFDM and CDMA signals were obtained in [31]–[33]. Question 1 has partly been answered in [28], [29], where it was shown that, for all $f \in \mathcal{B}_{\pi,0}^\infty$ and $a > 1$, we have

$$\|f\|_\infty \leq \frac{1}{\cos(\frac{\pi}{2a})} \|f|_{\mathbb{Z}/a}\|_{\ell^\infty}. \tag{3}$$

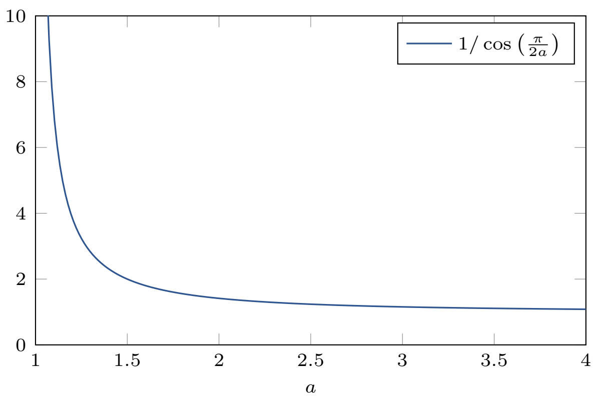

Fig. 2. Plot of the function $1/\cos(\pi/(2a))$. The unbounded increase when a tends to 1 is clearly visible.

The behavior of the upper bound on the right-hand side of (3) is plotted in Fig. 2. Inequality (3) shows that, with oversampling, we can bound the peak value of f from above by an expression that uses only the peak value of the samples $f|_{\mathbb{Z}/a}$ and the oversampling factor a. Inequality (3) can also be used to study the influence of the oversampling factor $a > 1$ on the computability and approximability of the peak value. If the oversampling factor a is increased, then the approximation error can be better controlled. This better control of the approximation error does not come for free, because the signal has to be known on the oversampling set.

We will study Questions 2 and 3 in the rest of this paper.

V. Computation of the Peak Values With Oversampling

The following theorem is our main result for the oversampling case. If we use oversampling, i.e., if we know f on an oversampling set $\mathbb{Z}/a$, $a > 0$, and if the sequence of samples $f|_{\mathbb{Z}/a}$ is computable, then we can compute $\|f\|_\infty$ from the samples $f|_{\mathbb{Z}/a}$. We prove this fact by providing an explicit algorithm for the computation of $\|f\|_\infty$. We will see in Section VII that without oversampling, i.e., for $a = 1$, the problem of computing the peak value cannot always be solved algorithmically.

Theorem 1. *Let $f \in \mathcal{B}^\infty_{\pi,0}$ and $a > 1$, $a \in \mathbb{R}_c$. If $f|_{\mathbb{Z}/a} \in \mathcal{C}c_0$ then we have $f(t) \in \mathbb{C}_c$ for all $t \in \mathbb{R}_c$, and we can compute $\|f\|_\infty \in \mathbb{R}_c$ algorithmically.*

Proof. Let $f \in \mathcal{B}^\infty_{\pi,0}$ and $a > 1$, $a \in \mathbb{R}_c$, be arbitrary but fixed. Further, let $\kappa \in \mathcal{CB}^1_{a\pi}$ be defined in the frequency domain by

$$\hat{\kappa}(\omega) = \begin{cases} \frac{1}{a}, & |\omega| \leq \pi, \\ \frac{|\omega| - a\pi}{a\pi(1-a)}, & \pi < |\omega| < a\pi, \\ 0, & |\omega| \geq a\pi. \end{cases}$$

Then we have

$$f(t) = \sum_{k=-\infty}^{\infty} f\left(\frac{k}{a}\right) \kappa\left(t - \frac{k}{a}\right), \quad t \in \mathbb{R}, \tag{4}$$

and the series in (4) converges absolutely. According to our assumption, we have $f|_{\mathbb{Z}/a} \in \mathcal{C}c_0$. Thus, it follows that there exists a recursive function η such that for all $M \in \mathbb{N}$ we have

$$\left| f\left(\frac{k}{a}\right) \right| \leq \frac{1}{2^M} \tag{5}$$

for all $|k| \geq \eta(M)$. We will prove this fact in Theorem 7. Let $M \in \mathbb{N}$ be arbitrary but fixed, and let

$$f_N(t) = \sum_{k=-N}^{N} f\left(\frac{k}{a}\right) \kappa\left(t - \frac{k}{a}\right), \quad t \in \mathbb{R}. \tag{6}$$

Then, for $N \geq \eta(M)$, we have

$$\begin{aligned} |f(t) - f_N(t)| &= \left| \sum_{|k|>N} f\left(\frac{k}{a}\right) \kappa\left(t - \frac{k}{a}\right) \right| \\ &\leq \max_{|k|>N} \left| f\left(\frac{k}{a}\right) \right| \sum_{k=-\infty}^{\infty} \left| \kappa\left(t - \frac{k}{a}\right) \right| \\ &\leq \frac{a(1+\pi)\|\kappa\|_1}{2^M} \end{aligned}$$

for all $t \in \mathbb{R}$ and $N \geq \eta(M)$, where we used Nikol'skiĭ's inequality [25, p. 49] as well as inequality (5) in the third line. Taking the supremum on both sides, it follows that

$$\|f - f_N\|_\infty \leq \frac{a(1+\pi)\|\kappa\|_1}{2^M} \tag{7}$$

for all $N \geq \eta(M)$. Since $\|\kappa\|_1 \in \mathbb{R}_c$, we see that $\{f_N\}_{N\in\mathbb{N}}$ converges effectively to f in the L^∞-norm. Thus, we have $f \in \mathcal{CB}^\infty_{a\pi,0}$, and, as a consequence, $f(t) \in \mathbb{C}_c$ for all $t \in \mathbb{R}_c$. Using the inverse triangle inequality, we further obtain

$$|\|f\|_\infty - \|f_N\|_\infty| \leq \|f - f_N\|_\infty \leq \frac{a(1+\pi)\|\kappa\|_1}{2^M}$$

for all $N \geq \eta(M)$, which shows that the computable sequence $\{\|f_N\|_\infty\}_{N\in\mathbb{N}}$ of computable numbers converges effectively to $\|f\|_\infty$. Hence, it follows that $\|f\|_\infty \in \mathbb{R}_c$. □

Remark 3. Note that the proof of Theorem 1 already gives us an algorithm how to compute the peak value $\|f\|_\infty$. We first specify the desired approximation accuracy ϵ, where ϵ has to be a computable number. Then we compute the corresponding M that achieves this accuracy, i.e., the smallest integer M such that

$$\frac{a(1+\pi)\|\kappa\|_1}{2^M} \leq \epsilon.$$

In the next step, we compute $N = \eta(M)$. This gives us the number of summands we need to use in (6). In the last step, we compute $\|f_N\|_\infty$, which is possible because f_N is the finite sum of computable functions. This number is the desired result that is guaranteed to be ϵ-close to $\|f\|_\infty$.

Remark 4. As we have seen in the proof of Theorem 1, for $t \in \mathbb{R}_c$, $f_N(t)$ is always a computable number and $\{f_N(t)\}_{N\in\mathbb{N}}$ is a computable sequence of computable numbers. Because of (7) it follows that $f(t) \in \mathbb{C}_c$ for all $t \in \mathbb{R}_c$. For $a = 0$ however, i.e., if no oversampling is used, we will see in Theorem 3 that this is no longer true in general.

VI. Decay Behavior With Oversampling

Even though bandlimited signals have always an infinite time duration, the assumption that they are essentially time-limited is often made in practical applications. For finite energy signals this is justified to the fact that the signal energy is concentrated in a certain time interval, and therefore the energy, and consequently the signal's amplitude, outside this interval is negligible.

For finite energy signals, it is possible to algorithmically determine a finite time interval that contains a certain prescribed amount of energy. We analyze if the same is true for the peak value of the signal, more specifically, we ask if it is always possible to compute a time interval such that outside of this interval the peak value of the signal is below some threshold.

In the following, we study this question for the oversampling case and under the assumption that the sequence of samples $f|_{\mathbb{Z}/a}$ is computable. Again, the answer is positive, i.e., we can compute the signal's concentration in the time domain from the discrete-time signal by using digital signal processing methods. This is expressed by the next result, which is a direct corollary of Theorem 1.

Corollary 1. *Let $f \in \mathcal{B}_{\pi,0}^{\infty}$ and $a > 1$, $a \in \mathbb{R}_c$. If $f|_{\mathbb{Z}/a} \in \mathcal{C}c_0$ then there exists a recursive function $\eta \colon \mathbb{N} \to \mathbb{N}$, such that, for all $M \in \mathbb{N}$, we have*

$$|f(t)| \leq \frac{1}{2^M} \tag{8}$$

for all $|t| \geq \eta(M)$.

Corollary 1 shows that if $f|_{\mathbb{Z}/a} \in \mathcal{C}c_0$ then for any threshold 2^{-M} we can compute a time instant $T_0 = \eta(M)$ such that the absolute value of the signal $|f(t)|$ is smaller than 2^{-M} outside the interval $[-T_0, T_0]$. Hence, it is possible to algorithmically determine an interval, on which the signal is "essentially" concentrated with respect to the amplitude. This behavior is illustrated in Fig. 3.

We will see in Theorem 6, Section VIII that all functions in $\mathcal{CB}_{\pi,0}^{\infty}$ have the property that the interval on which they are essentially concentrated, as described by (8), can be algorithmically determined. However, if only the samples are known to be computable, such as in Corollary 1, then oversampling is essential. In Corollary 2, Section VIII, we will prove that without oversampling, the time concentration cannot always be algorithmically controlled as in (8).

Proof of Corollary 1. Let $f \in \mathcal{B}_{\pi,0}^{\infty}$, $a > 1$, $a \in \mathbb{R}_c$, such that $f|_{\mathbb{Z}/a} \in \mathcal{C}c_0$. Inequality (7) in the proof of Theorem 1 shows that there exists a computable sequence $\{f_N\}_{N\in\mathbb{N}}$ of functions, having the shape (6), and a recursive function $\eta_1 : \mathbb{N} \to \mathbb{N}$, such that, for all $M \in \mathbb{N}$, we have

$$|f(t) - f_N(t)| \leq \frac{1}{2^{M+1}}$$

for all $N \geq \eta_1(M)$. Since each f_N is the finite sum of rapidly decreasing functions, there exists a recursive function $\eta_2 : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, such that, for all $M \in \mathbb{N}$, we have

$$|f_N(t)| \leq \frac{1}{2^{M+1}}$$

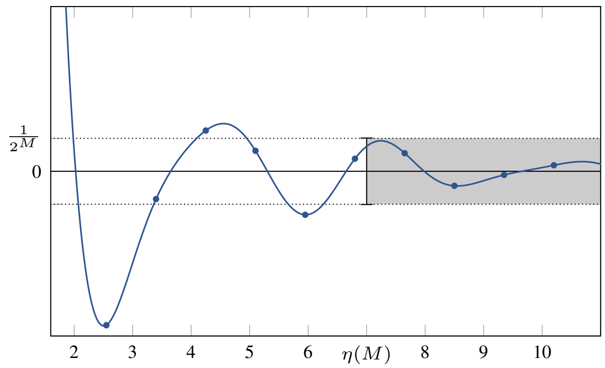

Fig. 3. Illustration of the decay behavior. We have $|f(t)| \leq 2^{-M}$ for all $|t| \geq T_0 = \eta(M)$. If the samples of f, taken at a rate above the Nyquist rate (oversampling), are computable, then η is computable.

for all $|t| \geq \eta_2(M, N)$. It follows that we have

$$\begin{aligned} |f(t)| &= |f(t) - f_{\eta_1(M)}(t) + f_{\eta_1(M)}(t)| \\ &\leq |f(t) - f_{\eta_1(M)}(t)| + |f_{\eta_1(M)}(t)| \\ &\leq \frac{1}{2^M} \end{aligned}$$

for all $|t| \geq \eta_2(M, \eta_1(M))$. □

VII. Computation of the Peak Values Without Oversampling

In this section we study the situation when no oversampling is used. Here, we will obtain a negative result and see that oversampling is indeed necessary to obtain the results from the last section.

Our first theorem is not related to computability, and shows that the peak value of a signal $f \in \mathcal{CB}_{\pi,0}^{\infty}$ cannot be inferred from the norm of its samples $\|f|_{\mathbb{Z}}\|_{\ell^\infty}$. Hence, a simple upper bound such as (3) cannot exist.

Theorem 2. *For all $M \in \mathbb{N}$, there exists a signal $f_M \in \mathcal{CB}_{\pi,0}^{\infty}$ such that $\|f_M|_{\mathbb{Z}}\|_{\ell^\infty} \leq 1$ and $\|f_M\|_\infty > M$.*

Proof. Let $M \in \mathbb{N}$ be arbitrary but fixed. For $N \in \mathbb{N}$, let

$$f(t, N) = \sum_{k=1}^{N} (-1)^k \frac{\sin(\pi(t-k))}{\pi(t-k)}, \quad t \in \mathbb{R}.$$

Then $f(\,\cdot\,, N)$ is computable, and we have $\|f(\,\cdot\,, N)|_{\mathbb{Z}}\|_{\ell^\infty} = 1$. For $t = N + 1/2$ we have

$$\begin{aligned} |f(N + \tfrac{1}{2}, N)| &= \frac{1}{\pi} \sum_{k=1}^{N} \frac{1}{N + \frac{1}{2} - k} \\ &= \frac{1}{\pi} \sum_{k=0}^{N-1} \frac{1}{k + \frac{1}{2}} \\ &> \frac{1}{\pi} \sum_{k=0}^{N-1} \int_k^{k+1} \frac{1}{\tau + \frac{1}{2}} \, \mathrm{d}\tau \\ &= \frac{1}{\pi} \int_0^N \frac{1}{\tau + \frac{1}{2}} \, \mathrm{d}\tau \\ &= \frac{1}{\pi} \left[\log\left(N + \frac{1}{2}\right) - \log\left(\frac{1}{2}\right) \right] \end{aligned}$$

$$= \frac{1}{\pi}\log(2N+1).$$

Hence, for $N \in \mathbb{N}$ with

$$N \geq \frac{\mathrm{e}^{\pi M} - 1}{2},$$

we have

$$\|f(\,\cdot\,,N)\|_\infty \geq |f(N+\tfrac{1}{2},N)| > M.$$

Choosing $f_M = f(\,\cdot\,,\overline{N})$ with $\overline{N} = \lceil(\mathrm{e}^{\pi M} - 1)/2\rceil$ completes the proof. □

In Theorem 2 we have seen that the knowledge of the peak value of the samples $\|f|_{\mathbb{Z}}\|_{\ell^\infty}$ is not enough to infer any information about the peak value of the continuous-time signal $\|f\|_\infty$. Next, we study from a computational point of view what happens if the entire sequence of samples $f|_{\mathbb{Z}}$ is available.

Theorem 1 has shown that if, for $f \in \mathcal{B}_{\pi,0}^\infty$, we have $f|_{\mathbb{Z}/a} \in \mathcal{C}c_0$ for some $a > 1$, $a \in \mathbb{R}_c$, then the signal values $f(t)$ are computable, i.e., we have $f(t) \in \mathbb{C}_c$ for all $t \in \mathbb{R}_c$. Next, we will prove that this is not guaranteed if $a = 1$, i.e., if no oversampling is used. For $a = 1$ it is not possible to infer the computability of the continuous time signal f from the computability of the discrete time signal $f|_{\mathbb{Z}}$. This highlights the importance of oversampling in Theorem 1.

Theorem 3. *There exists an $f_1 \in \mathcal{B}_{\pi,0}^\infty$ such that $f_1|_{\mathbb{Z}} \in \mathcal{C}c_0$ and $f_1(t) \notin \mathbb{C}_c$ for all $t \in \mathbb{R}_c \setminus \mathbb{Z}$. Hence, we have $f_1 \notin \mathcal{C}\mathcal{B}_{\pi,0}^\infty$.*

For the proof of Theorem 3, we need two auxiliary results. The first one is the Valiron sampling series, which is also sometimes called Tschakaloff's series. For a proof, see for example [34, p. 12] or [25, p. 60].

Lemma 1 (Valiron sampling series)**.** *For all $f \in \mathcal{B}_\pi^\infty$, we have*

$$f(t) = f(0)\frac{\sin(\pi t)}{\pi t} + f'(0)\frac{\sin(\pi t)}{\pi} + t\sum_{\substack{k=-\infty\\k\neq 0}}^{\infty} \frac{f(k)}{k}\frac{\sin(\pi(t-k))}{\pi(t-k)}, \quad t \in \mathbb{R}.$$

For fixed $t \in \mathbb{R}$, the series converges absolutely.

The second one is a statement about the computability of the last term in the Valiron expansion. Lemma 2 was proved in [21, p. 6433].

Lemma 2. *Let $f \in \mathcal{C}\mathcal{B}_{\pi,0}^\infty$ and $t \in \mathbb{R}_c$. Then we have*

$$t\sum_{\substack{k=-\infty\\k\neq 0}}^{\infty} \frac{f(k)}{k}\frac{\sin(\pi(t-k))}{\pi(t-k)} \in \mathbb{R}_c.$$

The proof of Theorem 3 follows the same line of ideas as the proof of Theorem 3 in [21], and parts are identical. However, since important details are different, and for the sake of completeness, we include the proof here.

Proof of Theorem 3. For $N \in \mathbb{N}$, let

$$p_N(t) = -\sum_{k=1}^{N}(-1)^k\frac{\sin(\pi(t-k))}{\pi(t-k)}, \quad t \in \mathbb{R}.$$

Since p_N is a finite sum of computable functions in $\mathcal{B}_{\pi,0}^\infty$, we see that $p_N \in \mathcal{C}\mathcal{B}_{\pi,0}^\infty$. For $t = 1/2$, we have

$$p_N\left(\frac{1}{2}\right) = \frac{1}{\pi}\sum_{k=1}^{N}\frac{1}{k-\frac{1}{2}}.$$

Note that $p_N(1/2)$ is a computable real number. Since

$$\frac{1}{k-\frac{1}{2}} > \int_k^{k+1}\frac{1}{\tau-\frac{1}{2}}\,\mathrm{d}\tau, \quad k \geq 1,$$

it follows that

$$p_N\left(\frac{1}{2}\right) = \frac{1}{\pi}\int_1^{N+1}\frac{1}{\tau-\frac{1}{2}}\,\mathrm{d}\tau > \frac{1}{\pi}\log(2N+1). \tag{9}$$

For $t \in \mathbb{Z}$, we have $|p_N(t)| \leq 1$. Further, for $t \in \mathbb{R}\setminus\mathbb{Z}$, we have

$$\begin{aligned}|p_N(t)| &\leq \sum_{k=1}^{N}\left|\frac{\sin(\pi(t-k))}{\pi(t-k)}\right| \\ &< 2 + \frac{1}{\pi}\sum_{k=1}^{k_1(t)}\frac{1}{t-k} + \frac{1}{\pi}\sum_{k=k_2(t)}^{N}\frac{1}{k-t} \\ &< 2 + \frac{1}{\pi}\sum_{k=1}^{k_1(t)}\frac{1}{k_1(t)+1-k} + \frac{1}{\pi}\sum_{k=k_2(t)}^{N}\frac{1}{k-k_2(t)+1} \\ &= 2 + \frac{1}{\pi}\sum_{k=1}^{k_1(t)}\frac{1}{k} + \frac{1}{\pi}\sum_{k=1}^{N-k_2(t)+1}\frac{1}{k} \\ &\leq 2 + \frac{2}{\pi}\sum_{k=1}^{N}\frac{1}{k} \\ &< 2 + \frac{2}{\pi} + \frac{2}{\pi}\log(N),\end{aligned}$$

where $k_1(t)$ is the largest natural number that is smaller than or equal to N and satisfies $k_1(t)+1 < t$. Further, $k_2(t)$ is the smallest natural number such that $k_2(t)-1 > t$. If $k_2(t) > N$ then the above sums involving $k_2(t)$ are the empty sums. We also used the inequality

$$\sum_{k=1}^{N}\frac{1}{k} < 1 + \sum_{k=2}^{N}\int_{k-1}^{k}\frac{1}{\tau}\,\mathrm{d}\tau = 1 + \int_1^N\frac{1}{\tau}\,\mathrm{d}\tau = 1 + \log(N)$$

in the last line. Hence, we have

$$\|p_N\|_\infty \leq 2 + \frac{2}{\pi} + \frac{2}{\pi}\log(N). \tag{10}$$

Let

$$g_N(t) = \frac{1}{p_N(\frac{1}{2})}p_N(t), \quad t \in \mathbb{R}.$$

We have

$$g_N(k) = \begin{cases} -\frac{(-1)^k}{p_N(\frac{1}{2})}, & 1 \leq k \leq N, \\ 0, & \text{otherwise.}\end{cases}$$

Further, for $N \in \mathbb{N}$, we have

$$\begin{aligned}\|g_N\|_\infty &= \frac{1}{|p_N(\frac{1}{2})|}\|p_N\|_\infty \\ &< \frac{2\pi}{\log(2N+1)} + \frac{2}{\log(2N+1)} + \frac{2\log(N)}{\log(2N+1)} \\ &< 2\pi + 4,\end{aligned}$$

where we used (10) and (9) in the first inequality.

Let $A \subset \mathbb{N}$ be an arbitrary recursively enumerable nonrecursive set, and let $\phi_A \colon \mathbb{N} \to \mathbb{N}$ be a recursive enumeration of the elements of A, where ϕ_A is an injective function, i.e., for every element $k \in A$ there exists exactly one $N_k \in \mathbb{N}$ with $\phi_A(N_k) = k$. We consider the function

$$f_1(t) = \sum_{N=1}^{\infty} \frac{1}{2^{\phi_A(N)}} g_N(t), \quad t \in \mathbb{R}. \tag{11}$$

Since

$$\begin{aligned}\sum_{N=1}^{\infty} \left\| \frac{1}{2^{\phi_A(N)}} g_N \right\|_\infty &\leq \sum_{N=1}^{\infty} \frac{1}{2^{\phi_A(N)}} \|g_N\|_\infty \\ &< \sum_{N=1}^{\infty} \frac{1}{2^N}(2\pi + 4) \\ &< 2\pi + 4,\end{aligned}$$

it follows that the series in (11) is absolutely convergent and that $f_1 \in \mathcal{B}_{\pi,0}^\infty$.

We further consider the sequence $f_1|_\mathbb{Z} = \{f_1(k)\}_{k\in\mathbb{Z}}$, defined by

$$f_1(k) = \sum_{N=1}^{\infty} \frac{1}{2^{\phi_A(N)}} g_N(k), \quad k \in \mathbb{Z}. \tag{12}$$

Since

$$\begin{aligned}\|f_1|_\mathbb{Z}\|_{\ell^\infty} &\leq \sum_{N=1}^{\infty} \frac{1}{2^{\phi_A(N)}} \|g_N|_\mathbb{Z}\|_{\ell^\infty} < \sum_{N=1}^{\infty} \frac{1}{2^N} \frac{1}{p_N(\frac{1}{2})} \\ &< \frac{\pi}{\log(2N+1)} < \pi,\end{aligned}$$

we see that the series in (12) is absolutely convergent and that $f_1|_\mathbb{Z} \in c_0$. We next prove that $f_1|_\mathbb{Z} \in \mathcal{C}c_0$. For $M \in \mathbb{N}$, the sequence

$$\left\{ \sum_{N=1}^{M} \frac{1}{2^{\phi_A(N)}} g_N(k) \right\}_{k\in\mathbb{N}} \tag{13}$$

is, as a finite linear combination of sequences in c_0 with only finitely many non-zero elements, a sequence in c_0 with only finitely many non-zero elements. Since $p_{N+1}(1/2) > p_N(1/2)$ for all $N \in \mathbb{N}$, we obtain, for $M \in \mathbb{N}$,

$$\begin{aligned}\left\| f_1|_\mathbb{Z} - \sum_{N=1}^{M} \frac{1}{2^{\phi_A(N)}} g_N|_\mathbb{Z} \right\|_{\ell^\infty} &\leq \sum_{N=M+1}^{\infty} \frac{1}{2^{\phi_A(N)}} \|g_N|_\mathbb{Z}\|_{\ell^\infty} \\ &< \frac{1}{p_{M+1}(\frac{1}{2})} \sum_{N=M+1}^{\infty} \frac{1}{2^N} \\ &< \frac{1}{p_{M+1}\left(\frac{1}{2}\right)} \\ &\leq \frac{\pi}{\log(2M+3)}.\end{aligned}$$

Thus, (13) converges effectively to $f_1|_\mathbb{Z}$ as M tends to infinity, which implies that $x_* \in \mathcal{C}c_0$. Since

$$\sum_{N=1}^{\infty} \frac{1}{2^{\phi_A(N)}}$$

is not computable [17], [27], it follows that

$$f_1\left(\frac{1}{2}\right) = \sum_{N=1}^{\infty} \frac{1}{2^{\phi_A(N)}} g_N\left(\frac{1}{2}\right) = \sum_{N=1}^{\infty} \frac{1}{2^{\phi_A(N)}}$$

is not computable, i.e., we have $f_1(1/2) \notin \mathbb{R}_c$.

According to Lemma 1, we have, using that $f_1(0) = 0$, that

$$f_1(t) = f_1'(0)\frac{\sin(\pi t)}{\pi} + \underbrace{t \sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} \frac{f_1(k)}{k} \frac{\sin(\pi(t-k))}{\pi(t-k)}}_{=:B_2(t)}. \tag{14}$$

According to Lemma 2 we have $B_2(t) \in \mathbb{R}_c$ for all $t \in \mathbb{R}_c$, and for $t = 1/2$ we know that $f_1(1/2) \notin \mathbb{R}_c$. Thus, the left-hand side and, consequently, the right-hand side of (14) are not computable for $t = 1/2$. It follows that $f_1'(0) \notin \mathbb{R}_c$. This implies that $f_1'(0)\sin(\pi t)/\pi \notin \mathbb{R}_c$, and, consequently, that $f_1(t) \notin \mathbb{R}_c$ for all $t \in \mathbb{R}_c \setminus \mathbb{Z}$. □

We have seen in Theorem 3 that the computability of the discrete time signal $f|_\mathbb{Z}$ does not always imply the computability of the continuous time signal f. However, if $f(t)$ is computable for at least one computable non-integer time instant t then $f(t)$ is computable for all $t \in \mathbb{R}_c$, as the next theorem shows. Note, however, that the computability of $f(t)$ for all $t \in \mathbb{R}_c$ does not imply that $f \in \mathcal{CB}_{\pi,0}^\infty$, as we will see in Section IX.

Theorem 4. *Let $f \in \mathcal{B}_{\pi,0}^\infty$ and $f|_\mathbb{Z} \in \mathcal{C}c_0$. We have $f(t) \in \mathbb{C}_c$ for all $t \in \mathbb{R}_c$ if and only if there exists a $t_1 \in \mathbb{R}_c \setminus \mathbb{Z}$ such that $f(t_1) \in \mathbb{C}_c$.*

Proof. "⇒": This direction is obvious. "⇐": Let $t_1 \in \mathbb{R}_c$ such that $f(t_1) \in \mathbb{C}_c$, and let

$$g(t) = \frac{f(t) - f(t_1)}{t - t_1}, \quad t \in \mathbb{R}.$$

Since $g \in \mathcal{B}_\pi^2$, we have

$$\begin{aligned}\frac{f(t) - f(t_1)}{t - t_1} = g(t) &= \sum_{k=-\infty}^{\infty} g(k) \frac{\sin(\pi(t-k))}{\pi(t-k)} \\ &= \sum_{k=-\infty}^{\infty} \frac{f(k) - f(t_1)}{k - t_1} \frac{\sin(\pi(t-k))}{\pi(t-k)}\end{aligned}$$

for all $t \in \mathbb{R}_c$. It follows that

$$\begin{aligned}f(t) = f(t_1) + (t - t_1) &\sum_{k=-\infty}^{\infty} \frac{f(k)}{k - t_1} \frac{\sin(\pi(t-k))}{\pi(t-k)} \\ - f(t_1)(t - t_1) &\sum_{k=-\infty}^{\infty} \frac{1}{k - t_1} \frac{\sin(\pi(t-k))}{\pi(t-k)}.\end{aligned} \tag{15}$$

It can be shown that, for all $t \in \mathbb{R}_c$, the first series in (15) converges effectively, because $f|_\mathbb{Z} \in \mathcal{C}c_0$. Hence, the limit is a number in $\mathbb{C}_c$. Similarly, for $t \in \mathbb{R}_c$, the limit of the second series is in $\mathbb{C}_c$. Both calculations, which use the effective convergence of $\sum_{k=1}^{\infty} 1/k^2$, are elementary but lengthy and therefore omitted. Since $f(t_1) \in \mathbb{C}_c$, it follows that $f(t) \in \mathbb{C}_c$ for all $t \in \mathbb{R}_c$, □

The next theorem, our main result about the non-computability of the peak value in the case where no oversampling is employed, shows that without oversampling there exist signals for which either the peak value is not computable or the maximum is attained at a non-computable time instant.

Theorem 5. *There exists a real-valued signal $f_1 \in \mathcal{B}_{\pi,0}^{\infty}$ with $f_1|_{\mathbb{Z}} \in \mathcal{C}c_0$ such that $\|f_1\|_\infty \notin \mathbb{R}_c$ or $\arg\max_{t\in\mathbb{R}} f_1(t) \notin \mathbb{R}_c$.*

Proof. We use the function $f_1 \in \mathcal{B}_{\pi,0}^{\infty}$ from Theorem 3. We have $f_1|_{\mathbb{Z}} \in \mathcal{C}c_0$ and $f_1(t) \notin \mathbb{R}_c$ for all $t \in \mathbb{R}_c \setminus \mathbb{Z}$. It can be shown that f_1 takes its maximum on the set $\mathbb{R} \setminus \mathbb{Z}$. Let $t_0 = \arg\max_{t\in\mathbb{R}} f_1(t)$. We do a proof by contradiction and assume that both $f_1(t_0) = \max_{t\in\mathbb{R}} f_1(t) \in \mathbb{R}_c$ and $t_0 \in \mathbb{R}_c$, and show that this assumption leads to a contradiction. If $t_0 \in \mathbb{R}_c$, then we know from Theorem 3 that $f_1(t_0) \notin \mathbb{R}_c$, which is a contradiction to our assumption. □

Theorems 2 and 5 show that the results for $a > 1$ are sharp in the sense that for $a = 1$ no algorithmic control of the peak value is possible, even for $f \in \mathcal{B}_{\pi,0}^{\infty}$.

Remark 5. The computability of the continuous time signal f in not a necessary condition for the peak value $\|f\|_\infty$ to be a computable number. Note that $\|f\|_\infty \in \mathbb{R}_c$ does not imply that we actually have a procedure to compute $\|f\|_\infty$ from f, it only means that $\|f\|_\infty$ is a computable number. We will come back to this topic in Theorem 13 of Section IX.

VIII. Decay Behavior Without Oversampling

In this section we derive three results about the decay behavior of computable signals and computable sequences, respectively.

If f is computable in $\mathcal{B}_{\pi,0}^{\infty}$ then, for every threshold, we can compute a time instant T_0 from which on the signal stays below the threshold.

Theorem 6. *Let $f \in \mathcal{CB}_{\pi,0}^{\infty}$. Then there exists a recursive function $\eta\colon \mathbb{N} \to \mathbb{N}$, such that, for all $M \in \mathbb{N}$, we have*

$$|f(t)| \le \frac{1}{2^M} \tag{16}$$

for all $|t| \ge \eta(M)$.

According to Theorem 6, for $f \in \mathcal{CB}_{\pi,0}^{\infty}$, it is possible to algorithmically compute the interval on which the signal f is essentially concentrated, as described by (16). Hence, signals $f \in \mathcal{CB}_{\pi,0}^{\infty}$ possess the same behavior as signals $f \in \mathcal{B}_{\pi,0}^{\infty}$ that additionally satisfy $f|_{\mathbb{Z}/a} \in \mathcal{C}c_0$ for some $a > 1$, $a \in \mathbb{R}_c$ (see Corollary 1). After the proof of Theorem 6, we will see in Theorem 7 that discrete-time signals $s \in \mathcal{C}c_0$ exhibit the same behavior. However, for general signals in $f \in \mathcal{B}_{\pi,0}^{\infty}$ that satisfy $f|_{\mathbb{Z}} \in \mathcal{C}c_0$, we do not have this algorithmic control in general, as Corollary 2 will show.

Proof of Theorem 6. Since $f \in \mathcal{CB}_{\pi,0}^{\infty}$, there exists a recursive function ξ and a computable sequence of elementary computable functions $\{f_N\}_{N\in\mathbb{N}}$ such that for all $M \in \mathbb{N}$ we have

$$\|f - f_N\|_\infty \le \frac{1}{2^M}$$

for all $N \ge \xi(M)$. Let $M \in \mathbb{N}$ be arbitrary but fixed. For $N \ge \xi(M+1)$ we have

$$|f(t) - f_N(t)| \le \frac{1}{2^{M+1}}$$

for all $t \in \mathbb{R}$. Each f_N is an elementary computable function, having the shape

$$f_N(t) = \sum_{k=-K(N)}^{K(N)} c_k(N) \frac{\sin(\pi(t-k))}{\pi(t-k)},$$

where $K(N)$ and $c_k(N)$, $k = -K(N), \dots, K(N)$, are recursive functions. For $|t| > K(N)$ we further have

$$\begin{aligned} |f_N(t)| &\le \sum_{k=-K(N)}^{K(N)} |c_k(N)| \left| \frac{\sin(\pi(t-k))}{\pi(t-k)} \right| \\ &\le \frac{1}{\pi(|t| - K(N))} \sum_{k=-K(N)}^{K(N)} |c_k(N)|. \end{aligned}$$

Since the numbers $c_k(N)$, $k = -K(N), \dots, K(N)$, are computable, it follows that

$$C_1(N) = \sum_{k=-K(N)}^{K(N)} |c_k(N)|$$

is a computable number. Next, we compute $l_0(N) \in \mathbb{N}$ such that

$$\frac{C_1(N)}{\pi l_0(N)} \le \frac{1}{2^{M+1}}.$$

Then, for $|t| \ge K(N) + l_0(N)$, we have

$$\begin{aligned} |f_N(t)| &\le \frac{C_1(N)}{\pi(|t| - K(N))} \\ &\le \frac{C_1(N)}{\pi(K(N) + l_0(N) - K(N))} \\ &= \frac{C_1(N)}{\pi l_0(N)} \\ &\le \frac{1}{2^{M+1}}, \end{aligned}$$

and it follows that

$$\begin{aligned} f(t) &= |f(t) - f_N(t) + f_N(t)| \\ &\le |f(t) - f_N(t)| + |f_N(t)| \\ &\le \frac{1}{2^{M+1}} + \frac{1}{2^{M+1}} = \frac{1}{2^M} \end{aligned}$$

for all $N \ge \xi(M+1)$ and $|t| \ge K(N) + l_0(N)$. Choosing $\eta(M) = K(\xi(M+1)) + l_0(\xi(M+1))$ completes the proof. □

Using the same arguments as in the proof of Theorem 6, we can derive an analogous result for computable sequences in $\mathcal{C}c_0$.

Theorem 7. *Let $\{s(n)\}_{n\in\mathbb{N}} \in \mathcal{C}c_0$. Then there exists a recursive function $\eta\colon \mathbb{N} \to \mathbb{N}$, such that, for all $M \in \mathbb{N}$ we, have*

$$|s(n)| \le \frac{1}{2^M}$$

for all $n \ge \eta(M)$.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TSP.2020.3042005

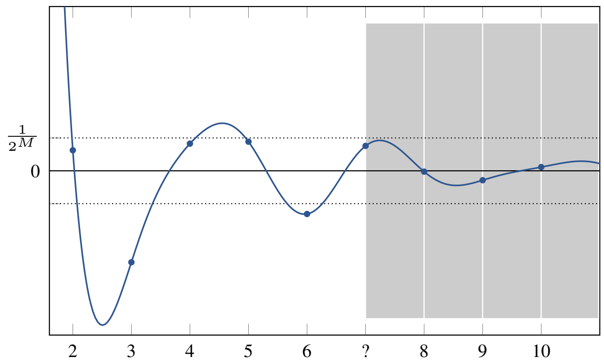

Fig. 4. Illustration of the decay behavior. The samples of f are taken at Nyquist rate (no oversampling) and are computable. However, it is not always possible to compute a time instant $T_0 = \eta(M)$, such that $|f(t)| \leq 2^{-M}$ for all $|t| \geq T_0 = \eta(M)$.

However, if we have a continuous-time signal $f \in \mathcal{B}_{\pi,0}^{\infty}$ and only know that the sequence of samples $f|_{\mathbb{Z}}$ is computable, then we cannot always compute such a time instant $T_0 = \eta(M)$ such that $|f(t)| \leq 2^{-M}$ outside the interval $[-T_0, T_0]$, as the next corollary shows. This is in contrast to the situation with oversampling that was discussed in Section VI, and, hence, here the question whether we can control the decay behavior of the continuous-time signal algorithmically has to answered in the negative. The problematic behavior is illustrated in Fig. 4.

Corollary 2. *There exists a signal $f_3 \in \mathcal{B}_{\pi,0}^{\infty}$ with $f(t) \in \mathbb{R}_c$ for all $t \in \mathbb{R}_c$ and $f|_{\mathbb{Z}} \in \mathcal{C}c_0$, such that there exists no recursive function $\eta\colon \mathbb{N} \to \mathbb{N}$ such that, for all $M \in \mathbb{N}$, we have*

$$|f_3(t)| \leq \frac{1}{2^M}$$

for all $|t| \geq \eta(M)$.

Remark 6. Note that f_3 in Corollary 2 cannot be in $\mathcal{CB}_{\pi,0}^{\infty}$, because otherwise the decay behavior would be computable according to Theorem 6.

Proof. For $n \in \mathbb{N}$, let

$$g_n(k) = \begin{cases} (-1)^k, & 0 \leq k \leq 2n, \\ 0, & \text{otherwise}, \end{cases}$$

and

$$g_n(t) = \sum_{k=0}^{2n} g_n(k) \frac{\sin(\pi(t-k))}{\pi(t-k)}, \quad t \in \mathbb{R}.$$

For $t_n = 2n + 1/2$, we have

$$g_n(t_n) = \frac{1}{\pi} \sum_{k=0}^{2n} \frac{1}{2n + \frac{1}{2} - k} =: C(n).$$

A simple calculation shows that

$$C(n) > \frac{\log(4n+1)}{\pi}.$$

We set

$$q_1(t) = \frac{1}{2^{\phi_A(1)}} \frac{g_1(t)}{C(1)}, \quad t \in \mathbb{R},$$

$\bar{t}_1 = t_1$, and $N_1 = 0$. For even N^* with $N^* > \bar{t}_1$ we have

$$\begin{aligned} \frac{g_2(\bar{t}_1 - N^*)}{C(2)} &= \frac{1}{C(2)} \sum_{k=N^*}^{N^*+4} (-1)^k \frac{\sin(\pi(\bar{t}_1 - k))}{\pi(\bar{t}_1 - k)} \\ &= \frac{\sin(\pi \bar{t}_1)}{\pi C(2)} \sum_{k=N^*}^{N^*+4} \frac{1}{\bar{t}_1 - k} \\ &= \frac{-1}{\pi C(2)} \sum_{k=N^*}^{N^*+4} \frac{1}{k - \bar{t}_1} \\ &> \frac{-1}{\pi C(2)} \frac{5}{N^* - \bar{t}_1}. \end{aligned}$$

Let $A \subset \mathbb{N}$ be an arbitrary recursively enumerable nonrecursive set, and let $\phi_A\colon \mathbb{N} \to \mathbb{N}$ be a recursive enumeration of the elements of A, where ϕ_A is an injective function. We choose $N_2 > \bar{t}_1$ as the smallest even number such that

$$-\frac{1}{2} \frac{1}{2^{\phi_A(1)}} < \frac{g_2(\bar{t}_1 - N_2)}{C(2)} < 0.$$

Further, we set

$$q_2(t) = q_1(t) + \frac{1}{2^{\phi_A(2)}} \frac{g_2(t - N_2)}{C(2)}, \quad t \in \mathbb{R},$$

and $\bar{t}_2 = N_2 + t_2$. For even N^* with $N^* > \bar{t}_2$ and $t = \bar{t}_1, \bar{t}_2$, we consider

$$\begin{aligned} \frac{g_3(t - N^*)}{C(3)} &= \frac{1}{C(3)} \sum_{k=N^*}^{N^*+6} (-1)^k \frac{\sin(\pi(t-k))}{\pi(t-k)} \\ &= \frac{\sin(\pi t)}{\pi C(3)} \sum_{k=N^*}^{N^*+6} \frac{1}{t-k} = \frac{-1}{\pi C(3)} \sum_{k=N^*}^{N^*+6} \frac{1}{k-t} \\ &> \frac{-1}{\pi C(3)} \frac{7}{N^* - t}. \end{aligned}$$

and chose $N_3 > \bar{t}_2$ as the smallest even number such that

$$-\frac{1}{4} \frac{1}{2^{\phi_A(1)}} < -\frac{7}{\pi C(3)} \frac{1}{N_3 - \bar{t}_1}$$

and

$$-\frac{1}{2} \frac{1}{2^{\phi_A(2)}} < -\frac{7}{\pi C(3)} \frac{1}{N_3 - \bar{t}_2}.$$

Then we have

$$-\frac{1}{4} \frac{1}{2^{\phi_A(1)}} < \frac{g_3(\bar{t}_1 - N_3)}{C(3)} < 0$$

and

$$-\frac{1}{2} \frac{1}{2^{\phi_A(2)}} < \frac{g_3(\bar{t}_2 - N_3)}{C(3)} < 0.$$

We set

$$q_3(t) = q_2(t) + \frac{1}{2^{\phi_A(3)}} \frac{g_3(t - N_3)}{C(3)}, \quad t \in \mathbb{R},$$

and $\bar{t}_3 = N_3 + t_3$. Suppose we have already defined N_k, q_k, and $\bar{t}_1, \ldots, \bar{t}_k$. Then, for even N^* with $N^* > \bar{t}_k$ and $t = \bar{t}_1, \ldots, \bar{t}_k$, we consider

$$\frac{g_{k+1}(t - N^*)}{C(k+1)}$$

and chose $N_{k+1} > \bar{t}_k$ as the smallest even number such that

$$-\frac{1}{2^{k-l+1}}\frac{1}{2^{\phi_A(l)}} < -\frac{t_{k+1}}{\pi C(k+1)}\frac{1}{N_{k+1}-\bar{t}_l}$$

for all $l = 1, \dots, k$. Then we have

$$-\frac{1}{2^{k-l+1}}\frac{1}{2^{\phi_A(l)}} < \frac{g_{k+1}(\bar{t}_l - N_{k+1})}{C(k+1)} < 0$$

for all $l = 1, \dots, k$. We set

$$q_{k+1}(t) = q_k(t) + \frac{1}{2^{\phi_A(k+1)}}\frac{g_{k+1}(t-N_{k+1})}{C(k+1)}, \quad t \in \mathbb{R},$$

and $\bar{t}_{k+1} = N_{k+1} + t_{k+1}$. Following this iterative procedure, we have constructed a sequence of functions $\{q_k\}_{k\in\mathbb{N}} \subset \mathcal{B}_{\pi,0}^{\infty}$, where

$$q_k(t) = \sum_{n=1}^{k}\frac{1}{2^{\phi_A(n)}}\frac{g_n(t-N_n)}{C(n)}.$$

Since

$$\|g_n\|_\infty \leq 2C(n), \quad n \in \mathbb{N},$$

and, for $M > N$,

$$\|q_M - q_N\|_\infty \leq \sum_{n=N+1}^{M}\frac{1}{2^{\phi_A(n)}}\frac{\|g_n\|_\infty}{C(n)} \leq 2\sum_{n=N+1}^{\infty}\frac{1}{2^{\phi_A(n)}},$$

we see that $\{q_k\}_{k\in\mathbb{N}}$ is a Cauchy sequence in $\mathcal{B}_{\pi,0}^{\infty}$. The unique limit $f_3 \in \mathcal{B}_{\pi,0}^{\infty}$ is our desired signal.

For $l \in \mathbb{N}$, we have

$$f_3(l) = \sum_{n=1}^{\infty}\frac{1}{2^{\phi_A(n)}}\frac{g_n(l-N_n)}{C(n)},$$

where the sum has only finitely many summands, and consequently each $f_3(l)$, $l \in \mathbb{N}$, is a computable real number. Next, we consider the computable sequence

$$\left\{\sum_{n=1}^{N}\frac{1}{2^{\phi_A(n)}}\frac{g_n(\,\cdot\, - N_n)}{C(n)}\right\}_{N\in\mathbb{N}}. \tag{17}$$

Note that each element of (17) is a sequence with only finitely many non-zero elements. We further have

$$\left\| f_3|_{\mathbb{Z}} - \sum_{n=1}^{N}\frac{1}{2^{\phi_A(n)}}\frac{g_n(\,\cdot\, - N_n)}{C(n)}\right\|_{\ell^\infty} \leq \sum_{n=N+1}^{\infty}\frac{1}{2^{\phi_A(n)}}\frac{1}{C(n)} < \frac{1}{C(N+1)}\sum_{n=N+1}^{\infty}\frac{1}{2^{\phi_A(n)}} < \frac{1}{C(N+1)} < \frac{\pi}{\log(4N+5)},$$

where we used that $\sup_{l\in\mathbb{Z}}|g_n(l)| = 1$ for all $n \in \mathbb{N}$. This shows that (17) converges effectively to $f_3|_{\mathbb{Z}}$. Hence. we have $f_3|_{\mathbb{Z}} \in \mathcal{C}c_0$.

We do the rest of the proof indirectly and assume that there exists a recursive function $\eta \colon \mathbb{N} \to \mathbb{N}$ such that, for all $M \in \mathbb{N}$, we have

$$|f_3(t)| \leq \frac{1}{2^M}$$

for all $|t| \geq \eta(M)$. Let $r \in \mathbb{N}$ be arbitrary. We have

$$\begin{aligned} f_3(\bar{t}_r) &= \sum_{n=1}^{\infty}\frac{1}{2^{\phi_A(n)}}\frac{g_n(\bar{t}_r - N_n)}{C(n)} \\ &= \sum_{n=1}^{r-1}\frac{1}{2^{\phi_A(n)}}\frac{g_n(\bar{t}_r - N_n)}{C(n)} + \frac{1}{2^{\phi_A(r)}}\frac{g_r(\bar{t}_r - N_r)}{C(n)} \\ &\quad + \sum_{n=r+1}^{\infty}\frac{1}{2^{\phi_A(n)}}\frac{g_n(\bar{t}_r - N_n)}{C(n)}. \end{aligned}$$

For $1 \leq n \leq r$ we have $\bar{t}_r - N_n > 2n$, and consequently $g_n(\bar{t}_r - N_n) \geq 0$. For $n \geq r+1$ we have

$$-\frac{1}{2^{n-r}}\frac{1}{2^{\phi_A(r)}} < \frac{g_n(\bar{t}_r - N_n)}{C(n)} < 0.$$

Hence, we obtain

$$\begin{aligned} f_3(\bar{t}_r) &> \frac{1}{2^{\phi_A(r)}}\frac{g_r(\bar{t}_r - N_r)}{C(n)} - \sum_{n=r+1}^{\infty}\frac{1}{2^{\phi_A(n)}}\frac{1}{2^{n-r}}\frac{1}{2^{\phi_A(r)}} \\ &> \frac{1}{2^{\phi_A(r)}}\frac{g_r(t_r)}{C(n)} - \frac{1}{2^{\phi_A(r)}}\sum_{n=r+1}^{\infty}\frac{1}{2}\frac{1}{2^{n-r}} \\ &= \frac{1}{2^{\phi_A(r)}} - \frac{1}{2}\frac{1}{2^{\phi_A(r)}} \\ &= \frac{1}{2}\frac{1}{2^{\phi_A(r)}}. \end{aligned}$$

Let $M \in \mathbb{N}$ be arbitrary, and let r_0 be the smallest number such that $\bar{t}_{r_0} \geq \phi(M)$. Then, for all $r \geq r_0$, we have

$$\frac{1}{2}\frac{1}{2^{\phi_A(r)}} < f_3(\bar{t}_r) \leq \frac{1}{2^M}.$$

It follows that $\phi_A(r) > M-1$ for all $r \geq r_0$, and, consequently, that $\phi_A(r) \notin [1, M-1]$ for all $r \geq r_0$. We consider the sets

$$A_{r_0} = \{\phi_A(1), \dots, \phi_A(r_0 - 1)\} \subset A$$

and

$$A_{r_0}^M = A_{r_0} \cap [1, \dots, M-1].$$

$A_{r_0}^M$ is the set of all $k \in A$ with $k \in [1, M-1]$, because for $r \geq r_0$ we have $\phi_A(r) > M-1$. Hence, for $k \in [1, \dots, M-1] \setminus A_{r_0}^M$, we have $k \notin A$. Since $M \in \mathbb{N}$ was arbitrary, we have an algorithm that can decide for each $k \in \mathbb{N}$, whether $k \in A$ or $k \notin A$. This implies that A is a recursive set, which is a contradiction. □

IX. Upper and Lower Bounds for Locally Computable Signals

We call a signal $f \in \mathcal{B}_{\pi,0}^{\infty}$ locally computable, if there exist a computable double sequence of elementary computable functions $\{f_{N,K}\}_{N\in\mathbb{N},K\in\mathbb{N}}$ and a recursive function $\xi \colon \mathbb{N}\times\mathbb{N} \to \mathbb{N}$ such that, for all $K \in \mathbb{N}$ and all $M \in \mathbb{N}$, we have

$$\max_{|t|\leq K}|f(t) - f_{N,K}(t)| \leq \frac{1}{2^M}$$

for all $N \geq \xi(M, K)$. The set of all locally computable signals $f \in \mathcal{B}_{\pi,0}^\infty$ is denoted by $\mathcal{LCB}_{\pi,0}^\infty$. Local computability in $\mathcal{B}_{\pi,0}^\infty$ is a weaker requirement than computability in $\mathcal{B}_{\pi,0}^\infty$. Hence, we have $\mathcal{LCB}_{\pi,0}^\infty \supset \mathcal{CB}_{\pi,0}^\infty$.

Theorem 8. *Let $f \in \mathcal{LCB}_{\pi,0}^\infty$. Then we have $\|f\|_\infty \in \mathbb{R}_c$. Further, there exists a computable sequence $\{TM_N\}_{N\in\mathbb{N}}$ of Turing machines $TM_N \colon \mathcal{LCB}_{\pi,0}^\infty \to \mathbb{R}_c$ such that*

1) for all $f \in \mathcal{LCB}_{\pi,0}^\infty$ and all $N \in \mathbb{N}$ we have

$$TM_{N+1}(f) \geq TM_N(f),$$

2) for all $f \in \mathcal{LCB}_{\pi,0}^\infty$ we have

$$\lim_{N\to\infty} TM_N(f) = \|f\|_\infty, \tag{18}$$

3) for each $f \in \mathcal{LCB}_{\pi,0}^\infty$ the convergence in (18) is effective, i.e., for each $f \in \mathcal{LCB}_{\pi,0}^\infty$ there exists a recursive function $\xi_f \colon \mathbb{N} \to \mathbb{N}$ such that for all $M \in \mathbb{N}$ we have

$$|TM_N(f) - \|f\|_\infty| \leq \frac{1}{2^M}$$

for all $N \geq \xi_f(M)$.

Remark 7. Theorem 8 does not imply that we have an algorithm that, for every $f \in \mathcal{LCB}_{\pi,0}^\infty$, can compute $\|f\|_\infty$. For $f \in \mathcal{CB}_{\pi,0}^\infty$, we know from Remark 1 that $\|f\|_\infty$ is algorithmically computable, i.e., there exists a Turing machine that, for every input $f \in \mathcal{CB}_{\pi,0}^\infty$ can determine $\|f\|_\infty$. In Theorem 8, i.e., for $f \in \mathcal{LCB}_{\pi,0}^\infty$, the situation is different. Even though $\|f\|_\infty \in \mathbb{R}_c$, there exists no algorithm that, for every $f \in \mathcal{LCB}_{\pi,0}^\infty$, can compute $\|f\|_\infty$. The reason is that the function ξ_f does not recursively depend on f, as we will see in Theorem 10.

Proof of Theorem 8. Let $f \in \mathcal{LCB}_{\pi,0}^\infty$ be arbitrary but fixed. Since $f \in \mathcal{LCB}_{\pi,0}^\infty$, there exist a computable double sequence of elementary computable functions $\{f_{N,K}\}_{N\in\mathbb{N},K\in\mathbb{N}}$ and a recursive function $\xi \colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that, for all $K \in \mathbb{N}$ and all $M \in \mathbb{N}$, we have $\max_{|t|\leq K} |f(t) - f_{N,K}(t)| < 2^{-M}$ for all $N \geq \xi(M, K)$. We set $f_M := f_{\xi(M,M),M}$. Clearly, $\{f_M\}_{M\in\mathbb{N}}$ is a computable sequence of computable functions, and we have

$$\max_{|t|\leq M} |f(t) - f_M(t)| \leq \frac{1}{2^M}.$$

Thus, it follows that

$$\max_{|t|\leq M} |f_M(t)| - \frac{1}{2^M} \leq \max_{|t|\leq M} |f(t)| \leq \max_{|t|\leq M} |f_M(t)| + \frac{1}{2^M}.$$

Since

$$\lim_{|t|\to\infty} f(t) = 0,$$

there exists a number $K_0 \in \mathbb{N}$ such that

$$\max_{|t|\leq K_0} |f(t)| = \|f\|_\infty.$$

Further, since f is a locally computable signal, and consequently a computable continuous function, we see from [17, p. 40, Theorem 7] that $\|f\|_\infty$ is a computable number, i.e., that $\|f\|_\infty \in \mathbb{R}_c$. For each $M \in \mathbb{N}$ there exists a Turing machine TM_M that computes

$$\mathrm{TM}_M f := \max_{|t|\leq M} |f_M(t)| - \frac{1}{2^M}.$$

Note that we have $\mathrm{TM}_{M+1}(f) > \mathrm{TM}_M(f)$, $M \in \mathbb{N}$, as well as

$$|\mathrm{TM}_M(f) - \|f\|_\infty| \leq \frac{2}{2^M} \tag{19}$$

for all $M \geq K_0$. Hence, we see that $\lim_{M\to\infty} \mathrm{TM}_M(f) = \|f\|_\infty$. For fixed $f \in \mathcal{LCB}_{\pi,0}^\infty$, the convergence of the sequence $\{\mathrm{TM}_M(f)\}_{M\in\mathbb{N}}$ to $\|f\|_\infty$ is effective according to (19). □

Based on the previous theorem, we can obtain a further interesting result. In order to state this result, we introduce the concept of semi-decidability. A set $\mathcal{S} \subset \mathcal{LCB}_{\pi,0}^\infty$ is called semi-decidable if there exists a Turing machine

$$\mathrm{TM}_\mathcal{S} \colon \mathcal{LCB}_{\pi,0}^\infty \to \{\mathrm{TM}_\mathcal{S} \text{ stops}, \mathrm{TM}_\mathcal{S} \text{ runs forever}\}$$

that, given an input $f \in \mathcal{LCB}_{\pi,0}^\infty$ stops if and only if $f \in \mathcal{S}$.

The next theorem shows that it is possible, for a given $\lambda \in \mathbb{R}_c$, $\lambda > 0$, to algorithmically detect the signals $f \in \mathcal{LCB}_{\pi,0}^\infty$ for which $\|f\|_\infty > \lambda$, because the corresponding set of signals

$$\mathcal{S}^> = \{f \in \mathcal{LCB}_{\pi,0}^\infty \colon \|f\|_\infty > \lambda\}$$

is semi-decidable. That is, there exists a Turing machine that takes $f \in \mathcal{LCB}_{\pi,0}^\infty$ as an input and stops if and only if $\|f\|_\infty > \lambda$. However, if $\|f\|_\infty \leq \lambda$ this Turing machine runs forever. Hence, if the machine has not stopped after a certain amount of time, we cannot say if $\|f\|_\infty \leq \lambda$, or if $\|f\|_\infty > \lambda$ but the machine has not yet finished. The other practical relevant set

$$\mathcal{S}^< = \{f \in \mathcal{LCB}_{\pi,0}^\infty \colon \|f\|_\infty < \lambda\}$$

is not semi-decidable, as we will see in Theorem 11. Hence, if $\|f\|_\infty < \lambda$, we have no way of verifying this algorithmically, the Turing machine may run forever. Note that this theorem is only concerned about continuous-time signals and makes no assertion about the connection to the corresponding discrete-time signals.

Theorem 9. *For all $\lambda \in \mathbb{R}_c$, $\lambda > 0$, the set*

$$\mathcal{S}^> = \{f \in \mathcal{LCB}_{\pi,0}^\infty \colon \|f\|_\infty > \lambda\}$$

is semi-decidable.

Proof. For all $\lambda \in \mathbb{R}_c$ there exists a Turing machine $\mathrm{TM}_\lambda^> \colon \mathbb{R}_c \to \{\mathrm{TM}_\lambda^> \text{ stops}, \mathrm{TM}_\lambda^> \text{ runs forever}\}$ such that, for each input $x \in \mathbb{R}_c$, $\mathrm{TM}_\lambda^>(x)$ stops if and only if $x > \lambda$ [17, p. 14, Proposition 0]. We also use the Turing machines TM_M that we defined in the proof of Theorem 8, given by

$$\mathrm{TM}_M f = \max_{|t|\leq M} |f_M(t)| - \frac{1}{2^M}.$$

We now describe an algorithm that defines a Turing machine $\mathrm{TM}_{\mathcal{S}^>}$ that, for $f \in \mathcal{LCB}_{\pi,0}^\infty$ stops if and only if $f \in \mathcal{S}^>$. The existence of this Turing machine proves that $\mathcal{S}^>$ is semi-decidable. Let $f \in \mathcal{LCB}_{\pi,0}^\infty$ and $\lambda \in \mathbb{R}_c$, $\lambda > 0$, be arbitrary but fixed. We compute $\mathrm{TM}_1(f)$ and start the Turing machine $\mathrm{TM}_\lambda^>(\mathrm{TM}_1(f))$. After the first instruction step, we check if

$\mathrm{TM}_\lambda^>(\mathrm{TM}_1(f))$ has stopped. If yes, then $\mathrm{TM}_1(f) > \lambda$, and we have

$$\|f\|_\infty \geq \max_{|r|\leq 1}|f_1(t)| - \frac{1}{2} > \lambda.$$

Further, we stop the algorithm, i.e., the Turing machine $\mathrm{TM}_{\mathcal{S}^>}$. If $\mathrm{TM}_\lambda^>(\mathrm{TM}_1(f))$ has not stopped then we compute $\mathrm{TM}_2(f)$ and start another Turing machine $\mathrm{TM}_\lambda^>(\mathrm{TM}_2(f))$. We execute the first instruction step of $\mathrm{TM}_\lambda^>(\mathrm{TM}_2(f))$ and the second instruction step of $\mathrm{TM}_\lambda^>(\mathrm{TM}_1(f))$. If one of these Turing machines has stopped, then we have, using the same reasoning as above, $\|f\|_\infty > \lambda$, and we stop the algorithm, i.e., the Turing machine $\mathrm{TM}_{\mathcal{S}^>}$. If neither $\mathrm{TM}_\lambda^>(\mathrm{TM}_2(f))$ nor $\mathrm{TM}_\lambda^>(\mathrm{TM}_1(f))$ has stopped, then we start the next iteration step. Note that if $\|f\|_\infty \leq \lambda$ then the algorithm, i.e., the Turing machine $\mathrm{TM}_{\mathcal{S}^>}(f)$ will run forever, because $\mathrm{TM}_M(f) < \|f\|_\infty \leq \lambda$ for all $M \in \mathbb{N}$.

Using this algorithm, we obtain a Turing machine $\mathrm{TM}_{\mathcal{S}^>}$ such that, for $f \in \mathcal{LCB}_{\pi,0}^\infty$, $\mathrm{TM}_{\mathcal{S}^>}(f)$ stops if and only if $\|f\|_\infty > \lambda$. Hence, we see that $\mathcal{S}^>$ is semi-decidable. □

Next, we prove that there exists no Turing machine that, for every $f \in \mathcal{LCB}_{\pi,0}^\infty$ can compute the peak value $\|f\|_\infty$. To this end, we consider the mapping $\psi\colon \mathcal{LCB}_{\pi,0}^\infty \to \mathbb{R}_c, f \mapsto \|f\|_\infty$, and prove that this mapping is not Banach–Mazur computable.

Theorem 10. *The mapping $\psi\colon \mathcal{LCB}_{\pi,0}^\infty \to \mathbb{R}_c$ is not Banach–Mazur computable.*

Proof. We need to find a computable sequence $\{g_n^*\}_{n\in\mathbb{N}} \subset \mathcal{LCB}_{\pi,0}^\infty$ such that the sequence of numbers $\{\psi(g_n^*)\}_{n\in\mathbb{N}} \subset \mathbb{R}_c$ is a non-computable sequence of computable reals. In particular, we will construct a computable sequence $\{g_n^*\}_{n\in\mathbb{N}}$ of functions in $\mathcal{LCB}_{\pi,0}^\infty$, such that $\psi(g_n^*) \in \{0,1\}$ for all $n \in \mathbb{N}$, but the sequence $\{\psi(g_n^*)\}_{n\in\mathbb{N}}$ is not computable, which means there exists no Turing machine $\mathrm{TM}\colon \mathbb{N} \to \{0,1\}$ with $\mathrm{TM}(n) = \psi(g_n^*)$, $n \in \mathbb{N}$.

Let $A \subset \mathbb{N}$ be an arbitrary recursively enumerable non-recursive set and $\phi_A\colon \mathbb{N} \to A$ a recursive enumeration of A, where ϕ_A is an injective function. Further, let

$$f_n(t) = \frac{\sin(\pi(t-n))}{\pi(t-n)}, \quad n \in \mathbb{N}. \tag{20}$$

$\{f_n\}_{n\in\mathbb{N}}$ is a computable sequence in $\mathcal{LCB}_{\pi,0}^\infty$. We set

$$g_{m,n} = \begin{cases} f_{\hat{m}}, & n \in \{\phi_A(1),\dots,\phi_A(m)\} \text{ with } \phi_A(\hat{m}) = n, \\ f_m, & n \notin \{\phi_A(1),\dots,\phi_A(m)\}. \end{cases}$$

$\{g_{m,n}\}_{m\in\mathbb{N},n\in\mathbb{N}}$ is a computable double sequence of functions in $\mathcal{LCB}_{\pi,0}^\infty$. Note that, if $n \in \{\phi_A(1),\dots,\phi_A(m)\}$ then there exists exactly one $\hat{m}$ with $\phi_A(\hat{m}) = n$. Further, we define the function $\kappa\colon \mathbb{N}\times\mathbb{N} \to \mathbb{N}$ by

$$\kappa(M,K) = K + 1 + 2^M.$$

Let $n \in \mathbb{N}$ be arbitrary. We first analyze the case $n \notin A$. Then we have $g_{m,n} = f_m$ for all $m \in \mathbb{N}$. According to the definition of f_m, we see that on each interval $[-K,K]$, the sequence $\{g_{m,n}\}_{m\in\mathbb{N}}$ of elementary computable functions converges uniformly to the zero function. For $K \in \mathbb{N}$ and $M \in \mathbb{N}$, we have for $m \geq \kappa(M,K)$ that

$$\begin{aligned} \max_{|t|\leq K}|g_{m,n}(t) - 0| &= \max_{|t|\leq K}|f_m(t)| < \frac{1}{\pi(m-K)} \\ &< \frac{1}{m-K} \leq \frac{1}{K+1+2^M-K} \\ &< \frac{1}{2^M}. \end{aligned} \tag{21}$$

Thus, for each interval $[-K,K]$, the sequence $\{g_{m,n}\}_{m\in\mathbb{N}}$ of elementary computable functions converges effectively to the zero function. We denote this limit function by $g_n^* \in \mathcal{LCB}_{\pi,0}^\infty$. Note that the convergence is effective in K.

Now, let $n \in A$. Let $\hat{m}$ denote the natural number for which $\phi_A(\hat{m}) = n$. Further, let $K \in \mathbb{N}$ and $M \in \mathbb{N}$, as well as $m \geq \kappa(M,K)$. If $n \in \{\phi_A(1),\dots,\phi_A(m)\}$ then we have $g_{m,n} = f_{\hat{m}}$. It follows that

$$\max_{|t|\leq K}|g_{m,n}(t) - f_m(t)| = 0 < \frac{1}{2^M}.$$

If $n \notin \{\phi_A(1),\dots,\phi_A(m)\}$, i.e., if $m < \hat{m}$, then we have $g_{m,n} = f_m$. It follows that

$$\begin{aligned} \max_{|t|\leq K}|g_{m,n}(t) - f_{\hat{m}}(t)| &= \max_{|t|\leq K}|f_m(t) - f_{\hat{m}}(t)| \\ &< \frac{1}{\pi(m-K)} + \frac{1}{\pi(\hat{m}-K)} \\ &\leq \frac{2}{\pi(m-K)} < \frac{1}{m-K} \\ &\leq \frac{1}{K+1+2^M-K} \\ &< \frac{1}{2^M}. \end{aligned} \tag{22}$$

Thus, for each interval $[-K,K]$, the computable double sequence $\{g_{m,n}\}_{m\in\mathbb{N}}$ of elementary computable functions converges effectively to $f_{\hat{m}}$. Again, we denote this limit function by $g_n^* \in \mathcal{LCB}_{\pi,0}^\infty$. Note that, as before, the convergence is effective in K.

Since (21) and (22) do not depend on n, the convergence of $\{g_{m,n}\}_{m\in\mathbb{N}}$ is also effective in n. It follows that $\{g_n^*\}_{n\in\mathbb{N}}$ is a computable sequence of functions in $\mathcal{LCB}_{\pi,0}^\infty$.

The sequence $\{\psi(g_n^*)\}_{n\in\mathbb{N}}$ is a sequence of numbers that satisfies $\psi(g_n^*) \in \{0,1\}$ for all $n \in \mathbb{N}$. We do a proof by contradiction and assume that $\{\psi(g_n^*)\}_{n\in\mathbb{N}}$ is a computable sequence. Let $n \in \mathbb{N}$ be arbitrary. If $n \in A$ then we have $g_n^* = f_{\hat{m}}$ and it follows that $\|g_n^*\|_\infty = \|f_{\hat{m}}\|_\infty = 1$. If $n \notin A$ then we have $g_n^* \equiv 0$ and therefore $\|g_n^*\|_\infty = 0$. For every $n \in \mathbb{N}$, we can algorithmically check whether the computable number $\psi(g_n^*)$ is larger or smaller than $1/2$. If $\psi(g_n^*) > 1/2$ then we have $n \in A$. And if $\psi(g_n^*) < 1/2$ then we have $n \notin A$. Hence, we have an algorithm that can determine whether $n \in A$ or $n \notin A$, which implies that A is a recursive set. This is a contradiction. Hence, our assumption was wrong, and it follows that $\{\psi(g_n^*)\}_{n\in\mathbb{N}}$ is not a computable sequence, which in turn implies that $\psi\colon \mathcal{LCB}_{\pi,0}^\infty \to \mathbb{R}_c$ is not Banach–Mazur computable. □

The proof of Theorem 10 further leads us to the next theorem, the counterpart of Theorem 9, where we showed that $\mathcal{S}^> = \{f \in \mathcal{LCB}_{\pi,0}^\infty\colon \|f\|_\infty > \lambda\}$ is semi-decidable.

Theorem 11. *For all $\lambda \in \mathbb{R}_c$, $\lambda > 0$, the set*

$$\mathcal{S}^< = \{f \in \mathcal{LCB}^\infty_{\pi,0} \colon \|f\|_\infty < \lambda\}$$

is not semi-decidable.

Proof. We do a proof by contradiction and assume that there exists a $\hat{\lambda} \in \mathbb{R}_c$, $\hat{\lambda} > 0$ such that the set

$$\{f \in \mathcal{LCB}^\infty_{\pi,0} \colon \|f\|_\infty < \hat{\lambda}\}$$

is semi-decidable. We use the same functions f_n, $n \in \mathbb{N}$ that were defined in (20) in the proof of Theorem 10, and consider the sequence $\{2\hat{\lambda} f_n\}_{n\in\mathbb{N}}$, which is a computable sequence of functions in $\mathcal{LCB}^\infty_{\pi,0}$. Let $A \subset \mathbb{N}$ be an arbitrary recursively enumerable non-recursive set and $\phi_A \colon \mathbb{N} \to A$ a recursive enumeration of A, where ϕ_A is an injective function. We use the same functions g^*_n, $n \in \mathbb{N}$ that were defined in the proof of Theorem 10, and consider the sequence $\{\hat{\lambda} g^*_n\}_{n\in\mathbb{N}}$, which is a computable sequence of functions in $\mathcal{LCB}^\infty_{\pi,0}$.

We define a Turing machine TM_A by starting two Turing machines in parallel. We start a Turing machine $\text{TM}_1(n)$ that checks for n if there exists a $\hat{m}$ such that $n = \phi_A(\hat{m})$. That is, we compute, for $m = 1, 2, \dots$, the sets $A_m = \{\phi_A(1), \dots, \phi_A(m)\}$ and check if $n \in A_m$. If so, we stop the Turing machine TM_1. If not, we let the Turing machine run. This Turing machine stops if and only if $n \in A$. According to our assumption there exists a Turing machine $\text{TM}_2 \colon \mathcal{LCB}^\infty_{\pi,0} \to \{\text{stops}, \text{runs forever}\}$ that stops if and only if $f \in \{f \in \mathcal{LCB}^\infty_{\pi,0} \colon \|f\|_\infty < \hat{\lambda}\}$. Parallel to TM_1, we start the Turing machine $\text{TM}_2(\hat{\lambda} g^*_n)$. This Turing machine stops if and only if $\|\hat{\lambda} g^*_n\|_\infty < \hat{\lambda}$, or, according to the definition of g^*_n, if and only if $n \notin A$. We see that eventually either TM_1 or TM_2 will stop. If TM_1 stops then we have $n \in A$, and if TM_2 stops we have $n \notin A$. The result is reported by the Turing machine TM_A. Hence, the Turing machine TM_A can determine for arbitrary $n \in \mathbb{N}$ whether $n \in A$ or $n \notin A$. This implies that A is a recursive set, which is a contradiction. □

Next, we study the domain of the signals and ask whether for all locally computable signals it is possible to algorithmically determine an interval on which the maximum of the signal is attained. For all $f \in \mathcal{B}^\infty_{\pi,0}$ there exists a natural number K_0 such that

$$\max_{|t|\le K_0} |f(t)| = \|f\|_\infty.$$

Further, for $f \in \mathcal{LCB}^\infty_{\pi,0}$, there always exists a $\hat{t} \in \mathbb{R}_c$, such that

$$\|f\|_\infty = |f(\hat{t})|. \tag{23}$$

For the existence of such a $\hat{t} \in \mathbb{R}_c$, see for example [17]. The question that we ask now is: Can we construct a Turing machine $\text{TM}_{\text{up}} \colon \mathcal{LCB}^\infty_{\pi,0} \to \mathbb{R}_c$ that, for each $f \in \mathcal{LCB}^\infty_{\pi,0}$, computes an upper bound for $|\hat{t}|$, i.e., a number $\bar{t}$ such that $|\hat{t}| \le \bar{t}$, where $\hat{t}$ is a number satisfying (23). The next theorem answers this question in the negative.

Theorem 12. *There exists no Turing machine $TM_{up} \colon \mathcal{LCB}^\infty_{\pi,0} \to \mathbb{R}_c$ that, for every $f \in \mathcal{LCB}^\infty_{\pi,0}$, computes a $\bar{t} = TM_{up}(f)$, such that $\|f\|_\infty = \max_{|t|\le\bar{t}} |f(t)|$.*

Remark 8. It is interesting that such a Turing machine TM_{up} does not exist. The goal to compute an upper bound for the time instant where the given signal attains its maximum, is weaker than the goal to compute the peak value itself or the time instant where the maximum is attained. In the following proof we will see that if we could find a Turing machine TM_{up} that solves the above problem, then there would exist a Turing machine that could compute $\|f\|_\infty$ for every input $f \in \mathcal{LCB}^\infty_{\pi,0}$.

Proof of Theorem 12. We do a proof by contradiction and assume that such a Turing machine TM_{up} exists. Using this Turing machine, we can compute $\bar{t} = \text{TM}_{\text{up}}(f)$. We have $\|f\|_\infty = \max_{|t|\le\bar{t}} |f(t)|$, and it follows that there exists a Turing machine $\text{TM}_1 \colon \mathcal{LCB}^\infty_{\pi,0} \to \mathbb{R}_c$ to compute $\|f\|_\infty$. The Turing machine TM_1 uses TM_{up} as a subroutine that computes $\bar{t} = \text{TM}_{\text{up}}(f)$. The remaining task for TM_1 is to compute and output the maximum of f on the interval $[-\bar{t}, \bar{t}]$, which is equal to $\|f\|_\infty$. However, such a Turing machine TM_1 cannot exist according to Theorem 10. □

As already mentioned in the introduction, for many problems in information and signal processing it has recently been shown that they cannot always be solved algorithmically on a digital computer. Examples are the computation of the Fourier transform [20], the bandlimited interpolation [21], the Wiener filter [23], and even the bandwidth of computable bandlimited signals [35]. In all these examples, it turned out that the key quantities themselves, such as the bandwidth, are not computable.

The question arose, whether this is a general phenomenon that holds for all signal processing problems or whether there exists interesting signal processing problems, where the key quantity is a computable number, but this number cannot be algorithmically derived from the signal. The above problems of computing the peak value for signals in $\mathcal{LCB}^\infty_{\pi,0}$ and of finding upper bounds have exactly this property. For each signal $f \in \mathcal{LCB}^\infty_{\pi,0}$ it is possible to find an algorithm, i.e., a Turing machine TM_f that computes $\|f\|_\infty$. This is possible because $\|f\|_\infty \in \mathbb{R}_c$, and hence, according to the definition of a computable number, an algorithm has to exist for the computation of $\|f\|_\infty$. However, this algorithm does not depend recursively on f, i.e., there exists no universal Turing machine that can compute $\|f\|_\infty$ for every signal $f \in \mathcal{LCB}^\infty_{\pi,0}$.

In Theorem 5, Section VII, we constructed a signal $f_1 \in \mathcal{B}^\infty_{\pi,0}$ with $f_1|_{\mathbb{Z}} \in \mathcal{C}c_0$ such that $\|f_1\|_\infty \notin \mathbb{R}_c$ or $\arg\max_{t\in\mathbb{R}} f_1(t) \notin \mathbb{R}_c$. This implies that $f_1 \notin \mathcal{CB}^\infty_{\pi,0}$. We next show that, for $f \in \mathcal{B}^\infty_{\pi,0}$ with $f|_{\mathbb{Z}} \in \mathcal{C}c_0$ and $\|f\|_\infty \in \mathbb{R}_c$, we do not necessarily have $f \in \mathcal{CB}^\infty_{\pi,0}$, i.e, $f \in \mathcal{CB}^\infty_{\pi,0}$ is not a necessary condition for the computability of the peak value $\|f\|_\infty$.

Theorem 13. *There exists a signal $f_2 \in \mathcal{B}^\infty_{\pi,0}$ such that*

1) $f_2|_{\mathbb{Z}} \in \mathcal{C}c_0$,
2) $f_2 \notin \mathcal{CB}^\infty_{\pi,0}$,
3) $\|f_2\|_\infty \in \mathbb{R}_c$.

Remark 9. The signal f_2 in Theorem 13, which will be constructed in the proof, is an explicit example of a signal that is in $\mathcal{LCB}^\infty_{\pi,0}$ but not in $\mathcal{CB}^\infty_{\pi,0}$.

Proof of Theorem 13. We use the function f_3 from Corollary 2, for which we already know from Remark 6 that $f_3 \notin \mathcal{CB}^\infty_{\pi,0}$. Further, we know that $f_3|_\mathbb{Z} \in \mathcal{C}c_0$. It remains to show that $\|f_3\|_\infty \in \mathbb{R}_c$. To this end, we use the representation

$$f_3(t) = \sum_{n=1}^{\infty} \frac{1}{2^{\phi_A(n)}} \frac{g_n(t-N_n)}{C(n)}, \quad t \in \mathbb{R}, \tag{24}$$

from the proof of Corollary 2. The series on the right-hand side of (24) converges effectively to f_3 on all intervals $[-M, M]$, $M \in \mathbb{N}$. Since

$$\left\{ \sum_{n=1}^{N} \frac{1}{2^{\phi_A(n)}} \frac{g_n(\,\cdot - N_n)}{C(n)} \right\}_{N\in\mathbb{N}}$$

is a computable sequence of functions in $\mathcal{CB}^\infty_{\pi,0}$, it follows that f_3 is a locally computable function in $\mathcal{LCB}^\infty_{\pi,0}$. Since $f_3 \in \mathcal{LCB}^\infty_{\pi,0}$, we obtain from Theorem 8 that $\|f_3\|_\infty \in \mathbb{R}_c$. □

X. Semi-Decidability of the Peak Value Problem on a Nyquist Set

Nowadays, the simulation of physical models and technical systems on digital computers is a standard method in research and development. However, as we have seen there exist signals and operations that cannot be computed algorithmically on a digital computer. In these cases, simulations cannot be used or give meaningless results, because it is impossible to assess how close the simulation output is to the real output. Modern simulations software usually contains functionality to assess the quality of the simulation. For example, tests can be executed to analyze the quality of the input data, the behavior of critical parameters during the simulation, and the confidence of the computed result. In some software packages such as MATLAB these kinds of checks are implemented in the form of exit flags, which indicate possible problems during the computation.

As for the peak value computation, we have seen that there exist signals, for which it is not possible to compute the peak value. In this section we study whether it is possible to determine the problematic signals algorithmically, i.e., to implement an exit flag for these input signals. To this end, we consider certain subsets of $\mathcal{C}c_0$, and use the concept of semi-decidability. The computation of an exit flag for the computability of the peak value $\|f\|_\infty$ is illustrated in Fig. 5.

For a sequence $x \in c_0$, let $f_x \in \mathcal{B}^\infty_{\pi,0}$ denote the bandlimited interpolation of x, i.e., the signal $f \in \mathcal{B}^\infty_{\pi,0}$ that satisfies $f_x(k) = x(k)$ for all $k \in \mathbb{Z}$, if it exists. For us the three sets

$$\mathcal{M}_1 = \{x \in \mathcal{C}c_0 \colon f_x \in \mathcal{B}^\infty_{\pi,0} \text{ and } f_x(\tfrac{1}{2}) \in \mathbb{C}_c\},$$

$$\mathcal{M}_2 = \{x \in \mathcal{C}c_0 \colon f_x \in \mathcal{B}^\infty_{\pi,0} \text{ and } f_x(\tfrac{1}{2}) \notin \mathbb{C}_c\},$$

and

$$\mathcal{M}_3 = \{x \in \mathcal{C}c_0 \colon f_x \in \mathcal{CB}^\infty_{\pi,0}\}$$

are interesting. We immediately see that $\mathcal{M}_1 = \mathcal{C}c_0 \setminus \mathcal{M}_2$, $\mathcal{M}_2 \cap \mathcal{M}_2 = \emptyset$, and $\mathcal{M}_3 \subset \mathcal{M}_1$.

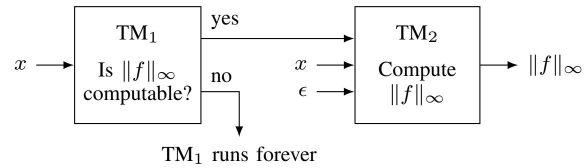

Fig. 5. Exit flag for the computability of the peak value $\|f\|_\infty$.

A set $\mathcal{M} \subseteq \mathcal{C}c_0$ is called semi-decidable if there exists a Turing machine

$$\text{TM} \colon \mathcal{C}c_0 \to \{\text{TM stops, TM runs forever}\}$$

that, given an input $x \in \mathcal{C}c_0$, stops if and only if $x \in \mathcal{M}$.

Theorem 14. *The subsets $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$ of $\mathcal{C}c_0$ are not semi-decidable.*

Before we can prove Theorem 14, we need to introduce the concept of a computable family of sequences in $\mathcal{C}c_0$. We call a family $\{x_\lambda\}_{\lambda\in[0,1]\cap\mathbb{R}_c}$ of sequences in $\mathcal{C}c_0$ a computable family of sequences in $\mathcal{C}c_0$ if there exist a computable sequence $\{\psi_k\}_{k\in\mathbb{Z}}$ of computable continuous functions $\psi_k \colon [0,1] \to \mathbb{C}_c$ and a recursive function $\xi \colon \mathbb{N} \to \mathbb{N}$, such that, for all $M \in \mathbb{N}$, we have

$$\|x_\lambda - x_\lambda^N\|_{\ell^\infty} \le \frac{1}{2^M}$$

for all $N \ge \xi(M)$, where

$$x_\lambda^N(k) = \begin{cases} \psi_k(\lambda), & |k| \le N, \\ 0, & |k| > N. \end{cases}$$

Remark 10. A family $\{x_\lambda\}_{\lambda\in[0,1]\cap\mathbb{R}_c} \subset \mathcal{C}c_0$ is a computable family if we can find an algorithm that obtains $M \in \mathbb{N}$ and $\lambda \in [0,1] \cap \mathbb{R}_c$ as inputs and then generates as output a $y_{M,\lambda} \in c_0$ with only finitely many non-zero elements such that $\|x_\lambda - y_{M,\lambda}\|_{\ell^\infty} \le 2^{-M}$. The so constructed output depends effectively on M and $\lambda \in [0,1] \cap \mathbb{R}_c$. This can be interpreted as follows. If $\{x_\lambda\}_{\lambda\in[0,1]\cap\mathbb{R}_c} \subset \mathcal{C}c_0$ is a computable family of sequences in $\mathcal{C}c_0$ then there exists a Turing machine $\text{TM} \colon [0,1] \cap \mathbb{R}_c \to \mathcal{C}c_0$ with $\text{TM}(\lambda) = x_\lambda$, i.e., TM generates a description of x_λ from a description of λ.

Proof of Theorem 14. We first show that $\mathcal{M}_3$ is not semi-decidable. From Theorem 3 we know that $\mathcal{M}_2$ is non-empty. Let $x_1 \in \mathcal{M}_2$ be arbitrary. Further, let $x_0 \in \mathcal{C}c_0$ be the sequence $x_0(k) = 0$ for all $k \in \mathbb{Z}$. We have $f_{x_0}(t) = 0$, $t \in \mathbb{R}$, and thus $x_0 \in \mathcal{M}_3$. For $\lambda \in [0,1] \cap \mathbb{R}_c$, we consider

$$\begin{aligned} x_\lambda(k) &= (1-\lambda)x_0(k) + \lambda x_1(k) \\ &= \lambda x_1(k), \quad k \in \mathbb{Z}. \end{aligned}$$

Since $x_1 \in \mathcal{C}c_0$, there exists a computable sequence $\{x_{1,n}\}_{n\in\mathbb{N}} \subset c_0$ where each $x_{1,n}$ has only finitely-many non-zero elements, and a recursive function ξ such that for all $M \in \mathbb{N}$ we have

$$\|x_1 - x_{1,n}\|_{\ell^\infty} \le 2^{-M}$$

for all $n \ge \xi(M)$. Let

$$x_{\lambda,n}(k) = \lambda x_{1,n}(k), \quad k \in \mathbb{Z}.$$

Then, for all $\lambda \in [0,1] \cap \mathbb{R}_c$ and all $M \in \mathbb{N}$, we have

$$\|x_\lambda - x_{\lambda,n}\|_{\ell^\infty} = \lambda \|x_1 - x_{1,n}\|_{\ell^\infty} \leq \frac{1}{2^M}$$

for all $n \geq \xi(M)$. Hence, we see that $\{x_\lambda\}_{\lambda \in [0,1]\cap\mathbb{R}_c}$ is a computable family of sequences in $\mathcal{C}c_0$.

Now we prove that $\mathcal{M}_3$ is not semi-decidable using an indirect proof. We assume that $\mathcal{M}_3$ is semi-decidable, i.e., that there exists a Turing machine $\mathrm{TM}_{\mathcal{M}_3} \colon \mathcal{C}c_0 \to \{\mathrm{TM}_{\mathcal{M}_3} \text{ stops}, \mathrm{TM}_{\mathcal{M}_3} \text{ runs forever}\}$ that, given an input $x \in \mathcal{C}c_0$, stops if and only if $x \in \mathcal{M}_3$, and show that this assumption leads to a contradiction. There exists a Turing machine $\mathrm{TM}_0^{>} : \mathbb{R}_c \to \{\text{stops}, \text{runs forever}\}$ such that, for each input $\lambda \in \mathbb{R}_c$, $\mathrm{TM}_0^{>}(\lambda)$ stops if and only if $\lambda > 0$ [17, p. 14, Proposition 0]. Now we construct a Turing machine $\mathrm{TM}(\lambda)$ with input $\lambda \in [0,1] \cap \mathbb{R}_c$ as follows. First, TM computes x_λ, which is possible, since $\{x_\lambda\}_{\lambda \in [0,1]\cap\mathbb{R}_c}$ is a computable family of sequences in $\mathcal{C}c_0$. Second, TM starts the two Turing machines $\mathrm{TM}_0^{>}(\lambda)$ and $\mathrm{TM}_{\mathcal{M}_3}(x_\lambda)$ in parallel. Exactly one of the two Turing machines will stop. $\mathrm{TM}_0^{>}(\lambda)$ will stop if and only if $\lambda > 0$, and $\mathrm{TM}_{\mathcal{M}_3}(x_\lambda)$ will stop if and only if $\lambda = 0$, because $x_\lambda \in \mathcal{M}_3$ if and only if $\lambda = 0$. The output of TM is 0 if $\mathrm{TM}_{\mathcal{M}_3}(x_\lambda)$ stops, and 1 if $\mathrm{TM}_0^{>}(\lambda)$ stops. Hence, TM is a Turing machine that can decide whether the input $\lambda \in [0,1] \cap \mathbb{R}_c$ is equal to zero or larger than zero. This is a contradiction because such a Turing machine cannot exist [17, p. 14, Proposition 0]. Thus, $\mathcal{M}_3$ is not semi-decidable.

The proof for $\mathcal{M}_1$ is done analogously as the proof for the set $\mathcal{M}_3$, because for x_0 we also have $x_0 \in \mathcal{M}_1$.

Next, we show that $\mathcal{M}_2$ is not semi-decidable. Again, we use an indirect proof and assume that $\mathcal{M}_2$ is semi-decidable, i.e., that there exists a Turing machine $\mathrm{TM}_{\mathcal{M}_2} \colon \mathcal{C}c_0 \to \{\mathrm{TM}_{\mathcal{M}_2} \text{ stops}, \mathrm{TM}_{\mathcal{M}_3} \text{ runs forever}\}$ that, given an input $x \in \mathcal{C}c_0$, stops if and only if $x \in \mathcal{M}_2$. Let $x_1 \in \mathcal{M}_2$ be arbitrary and consider, for $\epsilon \in [0,1] \cap \mathbb{R}_c$,

$$x_\epsilon(k) = x_1(k)\epsilon^{|k|}.$$

We have $x_\epsilon(k) \in \mathbb{C}_c$ for all $k \in \mathbb{Z}$. For $\epsilon = 1$, we have

$$x_\epsilon(k) = x_1(k), \quad k \in \mathbb{Z},$$

i.e., $x_\epsilon \in \mathcal{M}_2$. For $\epsilon \in [0,1) \cap \mathbb{R}_c$, we have

$$\begin{aligned}
&\left| f_{x_\epsilon}(\tfrac{1}{2}) - \sum_{k=-N}^{N} x_1(k)\epsilon^{|k|} \frac{\sin(\pi(\frac{1}{2}-k))}{\pi(\frac{1}{2}-k)} \right| \\
&= \left| \sum_{|k|>N} x_1(k)\epsilon^{|k|} \frac{\sin(\pi(\frac{1}{2}-k))}{\pi(\frac{1}{2}-k)} \right| \\
&\leq \|x_1\|_{\ell^\infty} \sum_{|k|>N} \epsilon^{|k|} \\
&= 2\|x_1\|_{\ell^\infty} \frac{\epsilon^{N+1}}{1-\epsilon}.
\end{aligned}$$

Since

$$\left\{ \sum_{k=-N}^{N} x_1(k)\epsilon^{|k|} \frac{\sin(\pi(\frac{1}{2}-k))}{\pi(\frac{1}{2}-k)} \right\}_{N\in\mathbb{N}}$$

is a computable sequence of computable numbers, we see that $f_{x_\epsilon}(1/2) \in \mathbb{C}_c$, and consequently that $x_\epsilon \in \mathcal{M}_1$. We now show that $\{x_\epsilon\}_{\epsilon\in[0,1]\cap\mathbb{R}_c}$ is a computable family in $\mathcal{C}c_0$. To this end, we consider

$$\psi_k(\epsilon) = x_1(k)\epsilon^{|k|}.$$

$\{\psi_k\}_{k\in\mathbb{Z}}$ is a computable sequence of computable continuous functions $\psi_k \colon [0,1] \to \mathbb{C}_c$. We choose

$$x_\epsilon^N(k) = \begin{cases} \psi_k(\epsilon), & |k| \leq N, \\ 0, & |k| > N. \end{cases}$$

Since $x_1 \in \mathcal{C}c_0$, there exist a computable sequence $\{x_{1,n}\}_{n\in\mathbb{N}} \subset c_0$, where each $x_{1,n}$ has only finitely many non-zero elements, and a recursive function $\xi \colon \mathbb{N} \to \mathbb{N}$, such that for all $M \in N$ we have $\|x_1 - x_{1,n}\|_{\ell^\infty} \leq 2^{-M}$ for all $n \geq \xi(M)$. For all $M \in \mathbb{N}$ and all $\epsilon \in [0,1] \cap \mathbb{R}_c$ we have

$$\begin{aligned}
|x_\epsilon(k) - x_\epsilon^N(k)| &= |x_1(k)\epsilon^{|k|} - x_1^N(k)\epsilon^{|k|}| \\
&= \epsilon^{|k|}|x_1(k) - x_1^N(k)| \\
&\leq |x_1(k) - x_1^N(k)| \\
&\leq \|x_1 - x_1^N\|_{\ell^\infty} \\
&\leq \frac{1}{2^M}
\end{aligned}$$

for all $k \in \mathbb{Z}$ and all $N \geq \xi(M)$. This shows that $\{x_\epsilon\}_{\epsilon\in[0,1]\cap\mathbb{R}_c}$ is a computable family in $\mathcal{C}c_0$. We already have shown that, for all $\epsilon \in [0,1) \cap \mathbb{R}_c$, we have $x_\epsilon \in \mathcal{M}_1$ and, for $\epsilon = 1$, $x_\epsilon = x_1 \in \mathcal{M}_2$.

Next, we construct a Turing machine $\mathrm{TM}(\epsilon)$ with input $\epsilon \in [0,1]\cap\mathbb{R}_c$ as follows. First, TM computes x_ϵ, which is possible because $\{x_\epsilon\}_{\epsilon\in[0,1]\cap\mathbb{R}_c}$ is a computable family in $\mathcal{C}c_0$. Second, TM starts the two Turing machines $\mathrm{TM}_1^{<}(\epsilon)$ and $\mathrm{TM}_{\mathcal{M}_2}(x_\epsilon)$ in parallel. Exactly one of the Turing machines will stop. $\mathrm{TM}_1^{<}(\epsilon)$ will stop if and only if $\epsilon < 1$, and $\mathrm{TM}_{\mathcal{M}_2}(x_\epsilon)$ will stop if and only if $\epsilon = 1$, because $x_\epsilon \in \mathcal{M}_2$ if and only if $\epsilon = 1$. However, such a Turing machine cannot exist [17, p. 14, Proposition 0]. This shows that $\mathcal{M}_2$ is not semi-decidable. □

XI. Conclusion

The peak value problem is an important problem, especially in communications, where we have to control the peak value of the transmit signal. In this work we analyzed whether certain questions regarding the peak value problem and the decay behavior can be answered algorithmically, i.e., on a digital computer. In the case of oversampling it is possible to compute the peak value of a bandlimited signal from its samples. However, if no oversampling is used, this is no longer possible in general. This shows that there are limitations in what can be computed on digital machine. For a more relaxed concept of computability, local computability, it is not even possible to always decide algorithmically whether the peak value of a signal is below a certain threshold.

The question of computability and the limitations of digital machines are usually not addressed in signal processing books and publications. If a certain value is not computable, then we cannot algorithmically control the error that is made in the approximation. As we have shown, this can be the case in the peak value computation of a continuous-time signal.

We also have seen that, for locally computable signals f, we always have $\|f\|_\infty \in \mathbb{R}_c$, but there exists no single Turing machine that can compute $\|f\|_\infty$ for every locally computable signal f as input to the Turing machine. This is an interesting example of a problem where the key quantity is a computable number, but where there exists no universal Turing machine that is capable of computing this number for all inputs.

It would be desirable to algorithmically identify the critical signals, for which computability problems exist, in order to filter them out before the actual computation begins. Unfortunately, such an exit flag functionality for detecting computability problems cannot exist, because the corresponding signal sets are not semi-decidable.

References

[1] H. Boche and U. J. Mönich, "Computability of the peak value of bandlimited signals," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '20)*, May 2020, pp. 5280–5284.

[2] F. Raab, P. Asbeck, S. Cripps, P. Kenington, Z. Popovic, N. Pothecary, J. Sevic, and N. Sokal, "Power amplifiers and transmitters for RF and microwave," *IEEE Transactions on Microwave Theory and Techniques*, vol. 50, no. 3, pp. 814–826, Mar. 2002.

[3] S. Litsyn, *Peak Power Control in Multicarrier Communications*. Cambridge University Press, 2012.

[4] G. Wunder, R. F. Fischer, H. Boche, S. Litsyn, and J.-S. No, "The PAPR problem in OFDM transmission: New directions for a long-lasting problem," *IEEE Signal Processing Magazine*, vol. 30, no. 6, pp. 130–144, Nov. 2013.

[5] T. Jiang, M. Guizani, H.-H. Chen, W. Xiang, and Y. Wu, "Derivation of PAPR distribution for OFDM wireless systems based on extreme value theory," *IEEE Transactions on Wireless Communications*, vol. 7, no. 4, pp. 1298–1305, Apr. 2008.

[6] C. Siegl and R. F. H. Fischer, "Asymptotic performance analysis and successive selected mapping for PAR reduction in OFDM," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3228–3237, Jun. 2010.

[7] B. S. Krongold and D. L. Jones, "An active-set approach for OFDM PAR reduction via tone reservation," *IEEE Transactions on Signal Processing*, vol. 52, no. 2, pp. 495–509, Feb. 2004.

[8] A. Aggarwal and T. H. Meng, "Minimizing the peak-to-average power ratio of OFDM signals using convex optimization," *IEEE Transactions on Signal Processing*, vol. 54, no. 8, pp. 3099–3110, Aug. 2006.

[9] N. Chen and G. T. Zhou, "Superimposed training for OFDM: a peak-to-average power ratio analysis," *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 2277–2287, Jun. 2006.

[10] T. Jiang and Y. Wu, "An overview: Peak-to-average power ratio reduction techniques for OFDM signals," *IEEE Transactions on Broadcasting*, vol. 54, no. 2, pp. 257–268, Jun. 2008.

[11] C. Li, T. Jiang, Y. Zhou, and H. Li, "A novel constellation reshaping method for PAPR reduction of OFDM signals," *IEEE Transactions on Signal Processing*, vol. 59, no. 6, pp. 2710–2719, Jun. 2011.

[12] S.-H. Wang, C.-P. Li, K.-C. Lee, and H.-J. Su, "A novel low-complexity precoded OFDM system with reduced PAPR," *IEEE Transactions on Signal Processing*, vol. 63, no. 6, pp. 1366–1376, Mar. 2015.

[13] A. M. Turing, "On computable numbers, with an application to the Entscheidungsproblem," *Proceedings of the London Mathematical Society*, vol. s2-42, no. 1, pp. 230–265, Nov. 1936.

[14] ——, "On computable numbers, with an application to the Entscheidungsproblem. A correction," *Proceedings of the London Mathematical Society*, vol. s2-43, no. 1, pp. 544–546, Jan. 1937.

[15] K. Weihrauch, *Computable Analysis: An Introduction*. Springer-Verlag, 2000.

[16] G. S. Boolos, J. P. Burgess, and R. C. Jeffrey, *Computability and Logic*. Cambridge University Press, 2002.

[17] M. B. Pour-El and J. I. Richards, *Computability in Analysis and Physics*. Springer-Verlag, 1989.

[18] J. Avigad and V. Brattka, "Computability and analysis: the legacy of Alan Turing," in *Turing's Legacy: Developments from Turing's Ideas in Logic*, R. Downey, Ed. Cambridge University Press, 2014.

[19] H. Boche and U. J. Mönich, "On the Fourier representation of computable continuous signals," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '19)*, May 2019, pp. 5013–5017.

[20] ——, "Turing computability of Fourier transforms of bandlimited and discrete signals," *IEEE Transactions on Signal Processing*, vol. 68, pp. 532–547, Jan. 2020.

[21] ——, "Downsampling of bounded bandlimited signals and the bandlimited interpolation: Analytic properties and computability," *IEEE Transactions on Signal Processing*, vol. 67, no. 24, pp. 6424–6439, Dec. 2019.

[22] ——, "Effective approximation of bandlimited signals and their samples," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '20)*, May 2020, pp. 5590–5594.

[23] H. Boche and V. Pohl, "On the algorithmic solvability of spectral factorization and applications," *IEEE Transactions on Information Theory*, 2020, in press.

[24] V. W. S. Wong, R. Schober, D. W. K. Ng, and L.-C. Wang, Eds., *Key Technologies for 5G Wireless Systems*. Cambridge University Press, 2017.

[25] J. R. Higgins, *Sampling Theory in Fourier and Signal Analysis – Foundations*. Oxford University Press, 1996.

[26] R. I. Soare, *Recursively Enumerable Sets and Degrees*, ser. Perspectives in Mathematical Logic. Springer-Verlag Berlin Heidelberg, 1987.

[27] E. Specker, "Nicht konstruktiv beweisbare Sätze der Analysis," *The Journal of Symbolic Logic*, vol. 14, no. 3, pp. 145–158, Sep. 1949.

[28] H. Boche and G. Wunder, "Über eine Verallgemeinerung eines Resultats von Riesz über trigonometrische Polynome auf allgemeine bandbegrenzte Funktionen," *Zeitschrift für angewandte Mathematik und Mechanik (ZAMM)*, vol. 82, no. 5, pp. 347–351, May 2002.

[29] G. Wunder and H. Boche, "Peak value estimation of bandlimited signals from their samples, noise enhancement, and a local characterization in the neighborhood of an extremum," *IEEE Transactions on Signal Processing*, vol. 51, no. 3, pp. 771–780, Mar. 2003.

[30] H. Al-Hammali and A. Faridani, "The zeros of a sine-type function and the peak value problem," *Signal Processing*, vol. 167, p. 107274, Feb. 2020.

[31] H. Boche and U. J. Mönich, "Peak to average power control via tone reservation in general orthonormal transmission systems," *IEEE Transactions on Signal Processing*, vol. 66, no. 13, pp. 3520–3528, Jul. 2018.

[32] ——, "Optimal tone reservation for CDMA systems," *IEEE Transactions on Signal Processing*, vol. 66, no. 23, pp. 6216–6227, Dec. 2018.

[33] J. Ilic and T. Strohmer, "PAPR reduction in OFDM using Kashin's representation," in *Proceedings of the 2009 IEEE 10th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Jun. 2009, pp. 444–448.

[34] P. L. Butzer, W. Splettstößer, and R. L. Stens, "The sampling theorem and linear prediction in signal analysis," *Jahresbericht der Deutschen Mathematiker-Vereinigung*, vol. 90, no. 1, pp. 1–70, Jan. 1988.

[35] H. Boche and U. J. Mönich, "Optimal sampling rate and bandwidth of bandlimited signals—an algorithmic perspective," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '20)*, May 2020, pp. 5905–5909.