algorithms

Article

Trustworthy Digital Representations of Analog Information—An Application-Guided Analysis of a Fundamental Theoretical Problem in Digital Twinning †

Holger Boche [1], Yannik N. Böck [1,*], Ullrich J. Mönich [1] and Frank H. P. Fitzek [2]

1 Chair of Theoretical Information Technology, Technical University of Munich, 80333 Munich, Germany; boche@tum.de (H.B.); moenich@tum.de (U.J.M.)
2 Deutsche Telekom Chair of Communication Networks, Technical University of Dresden, 01187 Dresden, Germany; frank.fitzek@tu-dresden.de
* Correspondence: yannik.boeck@tum.de

† This paper is an extended version of our paper published in Boche, H.; Mönich, U.J.; Böck, Y.N.; Fitzek, F.H.P. Optimization of Digital-Twin Representations of Analog Signals and Systems in Proceedings of the IEEE International Conference on Communications 2023, Rome, Italy, 28 May–1 June 2023; further, it will be discussed by Holger Boche and Frank H. P. Fitzek at the IEEE GLOBECOM within the tutorial Tactile Internet Support for the Metaverse.

Abstract: This article compares two methods of algorithmically processing bandlimited time-continuous signals in light of the general problem of finding "suitable" representations of analog information on digital hardware. Albeit abstract, we argue that this problem is fundamental in digital twinning, a signal-processing paradigm the upcoming 6G communication-technology standard relies on heavily. Using computable analysis, we formalize a general framework of machine-readable descriptions for representing analytic objects on Turing machines. Subsequently, we apply this framework to sampling and interpolation theory, providing a thoroughly formalized method for digitally processing the information carried by bandlimited analog signals. We investigate discrete-time descriptions, which form the implicit quasi-standard in digital signal processing, and establish continuous-time descriptions that take the signal's continuous-time behavior into account. Motivated by an exemplary application of digital twinning, we analyze a textbook model of digital communication systems accordingly. We show that technologically fundamental properties, such as a signal's (Banach-space) norm, can be computed from continuous-time, but *not* from discrete-time descriptions of the signal. Given the high trustworthiness requirements within 6G, e.g., employed software must satisfy assessment criteria in a provable manner, we conclude that the problem of "trustworthy" digital representations of analog information is indeed essential to near-future information technology.

Keywords: digital twinning; sampling; interpolation; Shannon series; bandlimited signals; Turing machines; computability; computable analysis; trustworthiness

check for updates

Citation: Boche, H.; Böck, Y.N.; Mönich, U.J.; Fitzek, F.H.P. Trustworthy Digital Representations of Analog Information—An Application-Guided Analysis of a Fundamental Theoretical Problem in Digital Twinning. *Algorithms* **2023**, *16*, 514. https://doi.org/10.3390/a16110514

Academic Editors: Ali Sadiq, Houbing Song, Ahmad Fadhil Yusof, Sushil Kumar and Omprakash Kaiwartya

Received: 30 September 2023
Revised: 20 October 2023
Accepted: 25 October 2023
Published: 9 November 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

State-of-the-art technological information processing happens mainly within the digital realm. Numerical values are quantized, and calculations are performed in discrete-time computational cycles. In contrast, the information carried by the physical world surrounding any technological device is analog and continuous. Twelve years after Alan Turing formalized the notion of digital computing [1], Claude Shannon established the theoretical foundation of sampling and interpolation theory [2]. Since then, scientists and engineers have extended Turing's and Shannon's theories and refined the relevant hardware, making the analog world increasingly accessible to digital information processing.

As follows from Shannon's sampling theorem, suitable (infinite) interpolation series uniquely restore any bandlimited continuous-time signal, provided that the signal's energy is finite and the samples are taken at least at the Nyquist rate. In its purest form, this result is present in signal processing and communications technology in the context of analog–digital/digital–analog conversion. In principle, however, any computational discretization method, such as finite-element algorithms, employs a similar paradigm: A set of discrete points, sufficiently dense for the in-between to become negligible, represents some continuous, physical object (c.f. [3] for a recent example in which the physical "object" is an electromagnetic field).

Another recent concept in the domain of digital information processing is known as digital twinning. According to the formalization established in [4], digital twinning commonly involves a physical entity to be twinned, a machine-readable description (in some machine-readable language) that represents the entity virtually on an appropriate hardware platform, and an interaction between the entity and its description through measurement and control. In this context, the machine-readable description is the entity's digital twin. More generally, if the particular type of computing hardware is not specified, we refer to the entity's virtual representation as a virtual twin. Furthermore, the term "entity" indicates that the virtual twin's physical counterpart does not necessarily have to be an actual object. In theory, any abstract formation—such as, for example, an entire communication network [5], or, as above, an electromagnetic field—qualifies for digital twinning, provided we can characterize it by a suitable mathematical model. Despite the different contexts, digital twinning resembles traditional Shannon Sampling and Interpolation (SSI) in some aspects. Both approaches represent a physical entity (in the context of SSI, a bandlimited signal) using digital data, digitalizing the relevant analog information through a sequential measurement process. However, classical SSI is geared towards completely restoring the physical entity, while digital twinning primarily aims to recover the entity's relevant properties. Within the employed mathematical model, relevant properties usually take the form of a (mathematical) function or relation with a particular interpretation on the practical level, such as the position of a material object at a given time or the total energy contained in an electromagnetic field.

Originally associated primarily with Industry 4.0 [6], digital twinning is attracting significant interest in many areas of modern technology. As part of the internet's anticipated evolution towards a unified metaverse, even more facets of the physical environment will connect to virtual space. In particular, information processing will increasingly incorporate the interaction between human multi-modalities (human senses) and the digital domain, c.f. [7]. In order to make human senses experienceable, the computational infrastructure will have to coordinate, process, and distribute the relevant information in real time. This requirement imposes engineers with unprecedented technological challenges regarding optimization, control, and decision-making. In this regard, research and development advocates digital twinning as one of several critical enablers. The novel technological applications that researchers envision in the context of digital twinning are just as ambitious. Medical research, for example, considers applications such as disease-trajectory estimation, optimization of medical-care timing, identification of biomarkers or elucidation of drug mechanisms, and patient-tailored prediction of treatment effects, employing digital twins of, e.g., a patient's immune system [8].

Given the potential for hazardous impacts of future digital-twinning applications on sensitive aspects of human well-being, the need to follow strict specifications on privacy, integrity, reliability, and safety is manifest. The upcoming 6G industry standard for communication technologies, which incorporates large parts of the technological infrastructure for the metaverse and other digital-twinning applications, summarizes such requirements by the term trustworthiness [9]. Depending on the potential hazards of an application, the physical entity's relevant properties must be reliably recoverable from the entity's digital twin. In practice, technological systems for critical applications must undergo technology assessment, which evaluates the implementation with regard to criteria of provable performance. When expressed in mathematical terms, e.g., by a margin of error, the recovered property must almost surely meet, such criteria entail "sufficient" and "insufficient" ways of representing a physical entity in virtual space. That is, the employed machine-readable language must satisfy specific structural characteristics, such that the relevant properties can be reliably computed from any of the entity's machine-readable descriptions. We summarize this observation in terms of the following fundamental problem statement, which we aim to elaborate on throughout this article.

> Given an application that requires the processing of analog information, find a sufficient way to represent the information on the chosen hardware platform.

The problem statement refers to general hardware platforms. As previously indicated, future virtual-twinning applications will not necessarily be limited to digital technology, c.f. [4]. However, in the scope of this article, we will only discuss traditional digital computing. Hence, in the problem statement, we may replace "the chosen hardware platform" with "digital hardware" in the context of the subsequent sections.

So far, our discussion on the fundamental problem statement and the associated concepts has been abstract, without a clear picture of how they translate to actual engineering problems. Throughout the subsequent sections, we aim to draw a precise picture of machine-readable languages, relevant properties, and proper representations for signal-processing and communications-engineering applications that involve traditional SSI. Aside from the conceptual similarities between digital twinning and SSI we discussed above, SSI has relevant direct applications in digital twinning. In the context of general virtual twinning, ref. [10] discussed an actual implementation of such an application, c.f. Figure 1.

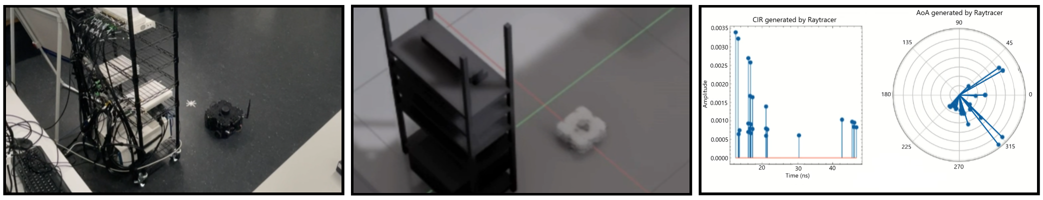

Figure 1. Application of digital twinning as discussed in [10]. (**Left**) A robot (physical entity) is moving on the floor of a laboratory environment. It is sequentially measuring its position relative to a fixed coordinate system and transmitting the relevant data through wireless communication to a receiving end. (**Middle**) The receiving end tracks the position and forms a virtual representation (digital twin) of the robot inside the room. It updates the virtual representation to match the physical robot whenever new information becomes available. The depicted image is thus a visualization of the robot's instantaneous machine-readable description. (**Right**) Using the robot's virtual representation, the receiving end computes the impulse response of the wireless communication link (the depicted implementation uses a ray-tracing approach), i.e., a sequence of samples representing a bandlimited signal. In essence, the impulse response forms a digital twin of the communication link.

Recall that traditional SSI aims at restoring the physical entity (i.e., a bandlimited signal) entirely. The relevant analytic result is known as (generalized) Plancherel–Pólya Theorem, c.f. Section 2 and Figure 2: The bandlimited signal uniquely determines the corresponding sequence of samples, and vice versa.

Accordingly, we expect that any property of the bandlimited signal should be recoverable from the sequence of sampling values. At this point, Turing's theory of digital computing enters the stage: A priori, the (generalized) Plancherel–Pólya Theorem is a purely analytic result. For it to hold on the algorithmic level, effectiveness in the sense of computable analysis is required. In this context, we will analyze two machine-readable languages emerging from the (generalized) Plancherel–Pólya Theorem for their structural properties. We employ the theory of Turing machines and effective analysis, classifying our results in terms of digital twinning and the article's fundamental problem statement. Particularly, we provide formal definitions of the terms machine-readable languages and machine-readable descriptions, and discuss formal examples of relevant properties. After the mathematical part of the article, we provide a brief subsumption and interpretation of our results, together with some prospects of how they affect near-future digital information-processing technology.

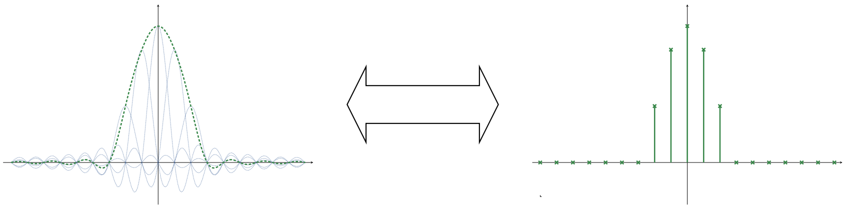

Figure 2. Generalized Plancherel–Pólya Theorem (c.f. Section 2). By evaluating a bandlimited, continuous-time signal (**left**) at all integer multiples of a suitable sampling interval T, we obtain the sequence of samples (**right**) corresponding to the function under consideration. We can restore the function through an (infinite) interpolation series, where each term is associated with one of the sample sequence's components. In particular, each term consists of a time-shifted interpolation function multiplied by the associated sample. This way, the bandlimited function and the sequence of sampling values uniquely determine each other.

The remainder of the article is structured as follows. In Section 2, we provide some mathematical background on SSI, introducing the signal spaces ℓ_0^∞, ℓ_0^1, $\mathcal{B}_{0,\pi}^\infty$, and $\mathcal{B}_\pi^1$, and formally establishing the (generalized) Plancherel–Pólya Theorem in terms of the Banach-space operators $\boldsymbol{S}_\star : \mathcal{B}(\star) \to \ell(\star)$ and $\boldsymbol{T}_\star := \boldsymbol{S}_\star^{-1}$. Applying the theory of Turing computability and effective analysis, we continue to develop a framework of machine-readable languages for $\mathcal{B}_{0,\pi}^\infty$ and $\mathcal{B}_\pi^1$. This framework formalizes the traditional theory of digital signal processing for communications engineering based on a mathematically rigorous notion of computability. Particularly, we define the machine-readable languages $\mathfrak{X}_1$ and $\mathfrak{X}_\infty$, which mirror the implicit quasi-standard in digital signal processing, and the machine-readable languages $\mathfrak{F}_1$ and $\mathfrak{F}_\infty$, which take the relevant signal's continuous-time behavior into account. In Section 3, we provide (for didactic purposes) a mathematical model of digital-twinning systems such as the one shown in Figure 1, marking the Banach-space norms $\|\cdot\|_\infty$ and $\|\cdot\|_1$ as a relevant property of signals $\boldsymbol{f} \in \mathcal{B}_{0,\pi}^\infty$, $\boldsymbol{f} \in \mathcal{B}_\pi^1$, respectively. Guided by the exemplary application case, we establish our main results: The (generalized) Plancherel–Pólya Theorem does *not* hold true on the algorithmic level. Depending on which of the established machine-readable languages we choose, we either can or cannot compute $\|\boldsymbol{f}\|_\infty$, $\|\boldsymbol{f}\|_1$, respectively, despite all languages determining the relevant signals uniquely in the (analytic) sense of the (generalized) Plancherel–Pólya Theorem. Finally, Section 4 discusses several other signal properties our theory can analyze and closes the article by interpreting our results as indicated above.

2. Materials and Methods

In the following, we provide a concise introduction to the mathematics of sampling and interpolation, which are primarily based on the theory of Banach spaces and linear operators. To this end, we introduce the Banach spaces ℓ_0^∞, ℓ^p, $\mathcal{B}_{0,\sigma}^\infty$, and $\mathcal{B}_\sigma^p$, $1 \leq p < \infty$, $0 < \sigma < \infty$. Commonly, $\mathcal{B}_{0,\sigma}^\infty$ and $\mathcal{B}_\sigma^p$ are referred to as Bernstein spaces. For a comprehensive introduction, we refer the reader to [11,12].

By ℓ_0^∞, we denote the set of all complex-valued sequences indexed by $\mathbb{Z}$ that vanish at infinity. That is, we have

$$\lim_{k\to\infty} \boldsymbol{x}[k] = \lim_{k\to-\infty} \boldsymbol{x}[k] = 0$$

for all $\boldsymbol{x} = (\boldsymbol{x}[k])_{k\in\mathbb{Z}} \in \ell_0^\infty$. Equipped with the uniform norm $\|\boldsymbol{x}\|_\infty := \sup_{k\in\mathbb{Z}}|\boldsymbol{x}[k]|$ the set ℓ_0^∞ becomes a Banach space. Further, by ℓ^p, $1 \le p < \infty$, we denote the Banach space of pth-power-summable sequences with the p-norm

$$\|\boldsymbol{x}\|_p := \left(\sum_{k=-\infty}^{\infty} |\boldsymbol{x}[k]|^p\right)^{1/p}.$$

A function $\boldsymbol{f} : \mathbb{C} \to \mathbb{C}, z \mapsto \boldsymbol{f}(z)$ is called entire if it is well-defined and holomorphic on all of $\mathbb{C}$. For entire functions that are (essentially) bounded on the real line, we define the essential-supremum norm $\|\boldsymbol{f}\|_\infty := \operatorname{ess\,sup}_{t\in\mathbb{R}}|\boldsymbol{f}(t)|$. The space $\mathcal{B}_{0,\sigma}^\infty$, $0 < \sigma < \infty$, consists of all entire functions $\boldsymbol{f}$ that satisfy the following conditions:

1. $\boldsymbol{f}$ is (essentially) bounded on the real line;
2. $\boldsymbol{f}$ of exponential type (c.f. [11], Lecture 1, p. 3ff) at most σ;
3. $\boldsymbol{f}(t)$ vanishes for $t \to \pm\infty$ on the real line.

Equipped with the essential supremum norm, the space $\mathcal{B}_{0,\sigma}^\infty$ becomes a Banach space. Further, the Banach spaces $\mathcal{B}_\sigma^p$, $1 \le p < \infty$, consists of all functions in $\mathcal{B}_{0,\sigma}^\infty$ that are pth-power integrable on the real line, equipped with the p-norm

$$\|\boldsymbol{f}\|_p := \left(\int_{t\in\mathbb{R}} |\boldsymbol{f}(t)|^p \, \mathrm{d}t\right)^{1/p}.$$

Pure mathematics studies all of the spaces ℓ_0^∞, ℓ^p, $\mathcal{B}_{0,\sigma}^\infty$, and $\mathcal{B}_\sigma^p$, $1 \le p < \infty, 0 < \sigma < \infty$. In contrast, only the spaces ℓ^1, ℓ^2, ℓ_0^∞ and $\mathcal{B}_\sigma^1$, $\mathcal{B}_\sigma^2$, $\mathcal{B}_{0,\sigma}^\infty$ occur frequently throughout signal processing and communications engineering, the arguably most "well-known" ones being ℓ^2 and $\mathcal{B}_\sigma^2$. They consist of discrete- and, respectively, bandlimited continuous-time signals with finite energy and form the mathematical basis for the seminal results in SSI, established before the relevant theory was extended to general Bernstein spaces. Fourier analysis provides a bijective isometry between ℓ^2 and $\mathcal{B}_\sigma^2$: defining $\boldsymbol{x}_{\boldsymbol{f}}[k] := \boldsymbol{f}(k\pi/\sigma)$, $k \in \mathbb{Z}$, we have

$$\begin{aligned}
\big[\widetilde{\mathcal{F}\boldsymbol{f}}\big](\xi) :&= \sqrt{\frac{\pi}{2\sigma^2}} \sum_{k=-\infty}^{\infty} \boldsymbol{x}_{\boldsymbol{f}}[k]\, \mathrm{e}^{-\jmath\pi k\xi/\sigma}, \\
\big[\mathcal{F}\boldsymbol{f}\big](\xi) :&= \frac{1}{\sqrt{2\pi}} \int_{t\in\mathbb{R}} \boldsymbol{f}(t)\, \mathrm{e}^{-\jmath\xi t} \, \mathrm{d}t = \begin{cases} \big[\widetilde{\mathcal{F}\boldsymbol{f}}\big](\xi), & \text{if } -\sigma \le \xi \le \sigma, \\ 0, & \text{otherwise,} \end{cases}
\end{aligned}$$

for all $\boldsymbol{f} \in \mathcal{B}_\sigma^2$, where $\mathcal{F}\boldsymbol{f}$ denotes the Fourier Transform of $\boldsymbol{f}$ on the real line. Through the definition of the sinc-function,

$$\operatorname{sinc}(z) := \begin{cases} \frac{\sin(\pi z)}{\pi z} & \text{if } z \neq 0, \\ 1 & \text{if } z = 0,, \end{cases}, \quad z \in \mathbb{C},$$

and the linearity of the Fourier Transform, the isometry provides Shannon's original sampling theorem for the spaces ℓ^2 and $\mathcal{B}_\sigma^2$, we have

$$\lim_{N\to\infty} \int_{\mathbb{R}} \left| \boldsymbol{f}(t) - \sum_{k=-N}^{N} \boldsymbol{x}_{\boldsymbol{f}}[k] \operatorname{sinc}\left(t - (k\pi/\sigma)\right) \right|^2 \mathrm{d}t = 0.$$

Since $\mathcal{F}\boldsymbol{f}$ is zero outside the interval $[-\sigma, \sigma]$, $\boldsymbol{f}$ is called bandlimited with bandwidth σ. The spaces $\mathcal{B}_\sigma^2$, $0 < \sigma < \infty$, thus correspond to the traditional notion of bandlimited signals. Through the definition of exponential types (Point 2 of the requirements above), this notion is generalized to a significantly larger class of functions. For $1 < p < \infty$ arbitrary, the Plancherel–Pólya theorem (Theorem 3, p. 152 in [11]) provides a nontrivial extension to the Shannon's sampling theorem.

Theorem 1 (Plancherel–Pólya)**.** *Let $1 < p < \infty$. For all sequences $\boldsymbol{x} \in \ell^p$, $1 < p < \infty$, there exists a unique function $\boldsymbol{f} \in \mathcal{B}_\sigma^p$, such that*

$$\lim_{N\to\infty} \int_{\mathbb{R}} \left| \boldsymbol{f}(t) - \sum_{k=-N}^{N} \boldsymbol{x}[k] \operatorname{sinc}\left(t - (k\pi/\sigma)\right) \right|^p \mathrm{d}t = 0$$

is satisfied. In particular, $\boldsymbol{f}$ is the unique solution to the interpolation problem $\boldsymbol{f}(k\pi/\sigma) = \boldsymbol{x}[k]$, $k \in \mathbb{Z}$. Conversely, for all signals $\boldsymbol{f} \in \mathcal{B}_\sigma^p$, $1 < p < \infty$, the sequence $(\boldsymbol{f}(k\pi/\sigma))_{k\in\mathbb{Z}}$ belongs to ℓ^p and there exist constants $C_L(p) > 0$ and $C_R(p) > 0$, independent of $\boldsymbol{f}$, such that

$$C_L(p) \sum_{k=-\infty}^{\infty} |\boldsymbol{f}(k\pi/\sigma)|^p \le \|\boldsymbol{f}\|_p^p \le C_R(p) \sum_{k=-\infty}^{\infty} |\boldsymbol{f}(k\pi/\sigma)|^p$$

holds true.

For the spaces ℓ^1, $\mathcal{B}_\sigma^1$ and ℓ_0^∞, $\mathcal{B}_{0,\sigma}^\infty$, $0 < \sigma < \infty$, Theorem 1 does *not* hold to its full extent. The (generalized) Plancherel–Pólya Theorem, which we will subsequently refer to as generalized Shannon equivalence, provides the following: For all $\boldsymbol{f} \in \mathcal{B}_\pi^1$, $\boldsymbol{f} \in \mathcal{B}_{0,\pi}^\infty$, respectively, and $(\boldsymbol{f}(k\pi/\sigma))_{k\in\mathbb{Z}} = \boldsymbol{x}_{\boldsymbol{f}}$, we have $\boldsymbol{f} \equiv 0$ if and only if we also have $\boldsymbol{x}_{\boldsymbol{f}} \equiv 0$. Furthermore, interpolation on the basis of sinc-functions provides uniform convergence on all compact subsets of $\mathbb{C}$, i.e., for all $C > 0$, we have

$$\lim_{N\to\infty} \operatorname*{ess\,sup}_{z\in\mathbb{C},|z|\le C} \left| \boldsymbol{f}(z) - \sum_{k=-N}^{N} \boldsymbol{x}_{\boldsymbol{f}}[k] \operatorname{sinc}\left(z - (k\pi/\sigma)\right) \right| = 0.$$

However, there exist sequences $\boldsymbol{x} \in \ell^1$, $\boldsymbol{x} \in \ell_0^\infty$, respectively, such that *no* function $\boldsymbol{f} \in \mathcal{B}_\sigma^1$, $\boldsymbol{f} \in \mathcal{B}_{0,\sigma}^\infty$, respectively, satisfies the interpolation condition $\boldsymbol{x}[k] = \boldsymbol{f}(k)$ for all $k \in \mathbb{Z}$. For ℓ^1, the Kronecker-delta family $\boldsymbol{\delta}_i \in \ell^1$, $i \in \mathbb{Z}$, defined by

$$\boldsymbol{\delta}_i[k] := \begin{cases} 1, & \text{if } i = k, \\ 0, & \text{otherwise}, \end{cases}, \quad k \in \mathbb{Z},$$

forms a simple example of such sequences. For an example in ℓ_0^∞, see (A2). In other words, the inclusions $\{\boldsymbol{x}_{\boldsymbol{f}} = (\boldsymbol{f}(k))_{k\in\mathbb{Z}} : \boldsymbol{f} \in \mathcal{B}_\pi^1\} \subset \ell^1$ and $\{\boldsymbol{x}_{\boldsymbol{f}} = (\boldsymbol{f}(k))_{k\in\mathbb{Z}} : \boldsymbol{f} \in \mathcal{B}_{0,\sigma}^\infty\} \subset \ell_0^\infty$ are proper. For further details regarding the generalized Shannon equivalence, we refer to [11,12] (Lecture 21, pp. 155–162; Chapter 6, pp. 48–66).

The results established in the present article hold true for all spaces $\mathcal{B}_\sigma^1$, $\mathcal{B}_{0,\sigma}^\infty$, $0 < \sigma < \infty$. In particular, the specific choice of σ is irrelevant. Therefore, without loss of generality, we will restrict ourselves to the case of $\sigma = \pi$ in the following, and denote

$$\operatorname{sinc}_k : \mathbb{C} \to \mathbb{C}, z \mapsto \operatorname{sinc}_k(z) := \operatorname{sinc}(z - k), \quad k \in \mathbb{Z}.$$

Further, most definitions and results hold analogously for both $\mathcal{B}_\pi^1$ and $\mathcal{B}_{0,\pi}^\infty$. For the sake of brevity, we will thus employ the symbol '$\star$' as a placeholder that may be (uniformly) replaced by '1' or '∞' within every appropriate scope (such as a definition, a lemma, a theorem, or a proof), and write

$$\ell(\star) := \begin{cases} \ell_0^\infty, & \text{if } \star = \infty, \\ \ell^1, & \text{otherwise}, \end{cases} \qquad \mathcal{B}(\star) := \begin{cases} \mathcal{B}_{0,\pi}^\infty, & \text{if } \star = \infty, \\ \mathcal{B}_\pi^1, & \text{otherwise}. \end{cases}$$

with some abuse of notation.

For notational convenience, we introduce the sampling operator $\boldsymbol{S}_\star : \mathcal{B}(\star) \to \ell(\star)$, $\boldsymbol{f} \mapsto \boldsymbol{S}_\star \boldsymbol{f} := (\boldsymbol{f}(k))_{k\in\mathbb{Z}}$ and its inverse $\boldsymbol{S}_\star^{-1} =: \boldsymbol{T}_\star$, which we refer to as interpolation operator. Subsequently, we will formalize the notion of computability for the spaces $\mathcal{B}(\star)$

and $\ell(\star)$. Then, the (informal) question of whether the generalized Shannon equivalence holds true on the algorithmic level corresponds to the (formal) question of whether the operators $\boldsymbol{S}_\star$ and $\boldsymbol{T}_\star$ are computable in the chosen machine-readable language, which we will address in Section 3. Observe that the sampling operator $\boldsymbol{S}_\star$ is bounded and injective. Thus, the interpolation operator $\boldsymbol{T}_\star$ is well-defined on a linear subspace

$$\mathrm{dom}\big(\boldsymbol{T}_\star\big) := \mathrm{img}\big(\boldsymbol{S}_\star\big) = \big\{\boldsymbol{x_f} = (\boldsymbol{f}(k))_{k\in\mathbb{Z}} : \boldsymbol{f} \in \mathcal{B}(\star)\big\}$$

of $\mathcal{B}(\star)$. However, $\boldsymbol{T}_\star$ is unbounded, and the subspace $\mathrm{dom}\big(\boldsymbol{T}_\star\big)$ is *not* closed, c.f. (A1) and (A4). Therefore, the set $\mathrm{dom}\big(\boldsymbol{T}_\star\big)$ is *not* a Banach space itself, which is essential in deriving the main results of our work.

Having established the analytic theory of sampling and interpolation, we will now turn to the formalization of its computable variant. To this end, we provide a concise introduction to the theory of Turing machines [1], μ-recursive functions [13], and computable analysis [14–17]. Although mature topics in the field of computer science, they have not yet received much attention within the signal-processing community.

Turing machines form an abstract mathematical model for digital computing. In fact, the widely accepted Church–Turing Thesis implies that they form a definitive and universal model of digital computing, i.e., any mathematical problem can (in principle) be solved through a real-world digital computer if and only if it can theoretically be solved by a Turing machine. Hence, if a certain algorithmic problem cannot be solved on a Turing machine, it can definitely not be solved on an actual digital hardware. The algorithms a Turing machine can compute is equivalent to the class of μ-recursive functions, c.f. [18] for the proof of equivalence.

By $\mathbb{N}$, we denote the set of natural numbers including zero. Throughout this article, it will occasionally be necessary to exclude zero from $\mathbb{N}$ to obtain a meaningful mathematical expression. As the reader may easily detect such a necessity from the relevant context, we avoid indicating them explicitly by a distinguished notation.

We call a mapping natural-number function if it is of the form $g : \mathbb{N}^n \supseteq\!\rightarrow \mathbb{N}$, $n \in \mathbb{N}$, where the symbol "$\supseteq\!\rightarrow$" denotes partiality. That is, we have $\mathrm{dom}(g) \subseteq \mathbb{N}^n$. A partial natural-number function is called total if the inclusion is improper, i.e., if we have $\mathrm{dom}(g) \subseteq \mathbb{N}^n$. Then, the set of μ-recursive functions $\mathcal{U}$ consists of all those natural-number functions that we can construct from the sucessor function, constant functions, and projection functions through application of composition, primitive recursion, and unbounded search (Definition 2.1, p. 8, Definition 2.2, p. 10 in [14]). By $\mathcal{U}(n)$, $n \in \mathbb{N}$, we denote the set of μ-recursive functions in n arguments, where $\mathcal{U}(0)$ can be understood as the set $\mathbb{N}$ itself, i.e., constant functions in zero arguments. Accordingly, we have $\mathcal{U}(0) \cup \mathcal{U}(1) \cup \mathcal{U}(2) \cup \ldots = \mathcal{U}$.

Observe that, generally speaking, μ-recursive functions are partial. When Turing machines are modeled as actual state-based machines that perform computations in sequential processing steps, the domain of the corresponding μ-recursive function equals the set of inputs for which the Turing machine halts its computation in finite time. A set $\Omega \subseteq \mathbb{N}$ is called recursively enumerable if it is either empty or the domain of a μ-recursive function. Consequently, a set $\Omega \subseteq \mathbb{N}$ is recursively enumerable if and only if it is either empty or the range of a total μ-recursive function. Furthermore, if Ω is recursively enumerable, there exists a (total) μ-recursive function $g_\Omega : \mathbb{N}^2 \rightarrow \{0, 1\}$ that satisfies the following for all $n \in \mathbb{N}$:

- There exists a number $m \in \mathbb{N}$ such that $g_\Omega(n, m) = 1$ is satisfied if and only if $n \in \Omega$ holds true;
- If $g_\Omega(n, m) = 1$ holds true for a number $m \in \mathbb{N}$, then $g_\Omega(n, k) = 1$ holds true for all $k \in \mathbb{N}$ that satisfy $k > m$.

We call such a function a runtime function for Ω. A set $\Omega \subset \mathbb{N}$ is called recursive if both Ω and $\mathbb{N} \setminus \Omega$ are recursively enumerable which, in turn, holds true if and only if the indicator function

$$\mathbb{1}_\Omega : \mathbb{N} \to \{0,1\},\ n \mapsto \begin{cases} 1 & \text{if } n \in \Omega, \\ 0 & \text{otherwise,} \end{cases}$$

of Ω is a (total) μ-recursive function.

Alan Turing introduced the concept of computable real numbers in [1]. Our definition of computable real numbers, and, subsequently, computable complex numbers, is based on computable sequences of rational numbers (p. 14 in [15]).

Definition 1. *A sequence of rational numbers, $(r_m)_{m\in\mathbb{N}}$, is called a computable sequence of rational numbers if there exist (total) μ-recursive functions $g, h_1, h_2 : \mathbb{N} \to \mathbb{N}$ such that*

$$r_m = \frac{(-1)^{g(m)} \cdot h_1(m)}{h_2(m) + 1}$$

is satisfied for all $m \in \mathbb{N}$. Analogously, for $n \in \mathbb{N}$, an n-fold computable sequence of rational numbers is defined through (total) μ-recursive functions $g, h_1, h_2 : \mathbb{N}^n \to \mathbb{N}$ in n arguments.

Definition 2. *A sequence of complex numbers, $(s_m)_{m\in\mathbb{N}}$, is called a computable sequence of rational-complex numbers if there exist a pair $((r_{1,m})_{m\in\mathbb{N}}, (r_{2,m})_{m\in\mathbb{N}})$ of computable sequences of rational numbers such that $s_m = r_{1,m} + \jmath r_{2,m}$ is satisfied for all $m \in \mathbb{N}$. Analogously, for $n \in \mathbb{N}$, an n-fold computable sequence of rational-complex numbers is defined through a pair of n-fold computable sequences of rational numbers.*

Definition 3. *A real number, x, is called computable if there exist a computable sequence of rational numbers $(r_n)_{n\in\mathbb{N}}$ and a (total) μ-recursive function $\xi\colon \mathbb{N} \to \mathbb{N}$ such that $|x - r_n| < 2^{-M}$ holds true for all $n, M \in \mathbb{N}$ that satisfy $n \geq \xi(M)$. For a triple $(x, (r_n)_{n\in\mathbb{N}}, \xi)$ of this kind, we write $[(r_n)_{n\in\mathbb{N}}, \xi]_{\mathfrak{R}} = x$. Further, we denote the set of computable real numbers by $\mathbb{R}_\mu$.*

Definition 4. *A complex number, z, is called computable if there exist a computable sequence of rational-complex numbers $(s_n)_{n\in\mathbb{N}}$ and a (total) μ-recursive function $\xi\colon \mathbb{N} \to \mathbb{N}$ such that $|z - s_n| < 2^{-M}$ holds true for all $n, M \in \mathbb{N}$ that satisfy $n \geq \xi(M)$. For a triple $(z, (s_n)_{n\in\mathbb{N}}, \xi)$ of this kind, we write $[(s_n)_{n\in\mathbb{N}}, \xi]_{\mathfrak{C}} = z$. Further, we denote the set of computable complex numbers by $\mathbb{C}_\mu$.*

In Definitions 3 and 4, the μ-recursive function ξ provides a computable way to control the approximation error $|x - r_n|$, $n \in \mathbb{N}$, $|z - s_n|$, $n \in \mathbb{N}$, respectively. In this case, the convergence of $(r_n)_n$ and $(s_n)_{n\in\mathbb{N}}$ to x and z, respectively, is referred to as effective, and the function ξ is called a corresponding effective modulus of convergence.

In the following, we will extend the established concepts of computability to the spaces $\ell(\star)$ and $\mathcal{B}(\star)$. To this end, we analogously write

$$\mathcal{C}\ell(\star) := \begin{cases} \mathcal{C}\ell_0^\infty, & \text{if } \star = \infty, \\ \mathcal{C}\ell^1, & \text{otherwise,} \end{cases} \qquad \mathcal{CB}(\star) := \begin{cases} \mathcal{CB}_{0,\pi}^\infty, & \text{if } \star = \infty, \\ \mathcal{CB}_\pi^1, & \text{otherwise,} \end{cases}$$

with the relevant formal definitions following below. Further, we employ an enumeration $\nu : \mathbb{N} \to \mathbb{Z}$, $n \mapsto \nu(n)$ of the integers $\mathbb{Z}$, defined through

$$\nu(n) := \frac{1}{2}(-1)^{(n \mod 2)}\big(n + (n \mod 2)\big).$$

Definition 5. *A sequence, $\boldsymbol{x} \in \ell(\star)$, is called computable in $\ell(\star)$ if there exist a computable double sequence of rational-complex numbers $(s_{n,m})_{n,m\in\mathbb{N}}$ and a (total) μ-recursive function $\xi\colon \mathbb{N} \to \mathbb{N}$, such that*

$$\left\| \boldsymbol{x} - \sum_{n=0}^{\xi(M)} s_{n,M} \cdot \boldsymbol{\delta}_{\nu(n)} \right\|_\star < \frac{1}{2^M} \tag{1}$$

is satisfied for all $M \in \mathbb{N}$. We denote the set of all such sequences by $\mathcal{C}\ell(\star)$. Further, if we have $\boldsymbol{f} = \boldsymbol{T}_\star \boldsymbol{x}$ for some $\boldsymbol{f} \in \mathcal{CB}(\star)$ (c.f. Definition 6), we write $[(s_{n,m})_{n,m\in\mathbb{N}}, \xi]^\star_{\mathfrak{X}} = \boldsymbol{f}$ for the triple $(\boldsymbol{f}, (s_{n,m})_{n,m\in\mathbb{N}}, \xi)$.

Definition 6. *A function, $\boldsymbol{f} \in \mathcal{B}(\star)$, is called computable in $\mathcal{B}(\star)$ if there exist a computable double sequence of rational-complex numbers $(s'_{n,m})_{n,m\in\mathbb{N}}$ and a (total) μ-recursive function $\xi'\colon \mathbb{N} \to \mathbb{N}$, such that*

$$\left\| \boldsymbol{f} - \sum_{n=0}^{\xi'(M)} s'_{n,M} \cdot \mathrm{sinc}_{\nu(n)} \right\|_\star < \frac{1}{2^M} \tag{2}$$

is satisfied for all $M \in \mathbb{N}$. We denote the set of all such functions by $\mathcal{CB}(\star)$. Further, we write $[(s'_{n,m})_{n,m\in\mathbb{N}}, \xi']^\star_{\mathfrak{F}} = \boldsymbol{f}$ for the triple $(\boldsymbol{f}, (s'_{n,m})_{n,m\in\mathbb{N}}, \xi')$.

Observe that, generally speaking, linear combinations of sinc-functions are *not* elements of $\mathcal{B}^1_\pi$. However, specific linear combinations of sinc-functions with rational-complex coefficients are, in fact, elements of $\mathcal{B}^1_\pi$, and the set of these linear combinations is dense in $\mathcal{B}^1_\pi$. For details, we refer the reader to Appendix B, p. 6363f in [19] (upon minor adjustments, the proof presented in the reference holds true for the restricted case of rational-complex coefficients).

A sequence $\boldsymbol{x} \in \mathcal{C}\ell(\star)$ is called an elementary computable if there exists a rational-complex $(2L+1)$-tuple $(z_k)_{k\in\mathcal{I}}, \mathcal{I} := \{0, \dots, 2L\}$, such that we have

$$\boldsymbol{x} := \sum_{k=0}^{2L} z_k \cdot \boldsymbol{\delta}_{\nu(k)}.$$

Analogously, a function $\boldsymbol{f} \in \mathcal{CB}(\star)$ is called an elementary computable if there exists an elementary computable sequence $\boldsymbol{x} \in \mathcal{C}\ell(\star)$ such that we have $\boldsymbol{f} = \boldsymbol{T}_\star \boldsymbol{x}$. Hence, elementary computable functions are exactly those functions that we can represented by a finite interpolation series with rational-complex coefficients $(z_k)_{k\in\mathcal{I}}$ in the sense of traditional SSI.

For $\boldsymbol{x} \in \mathcal{C}\ell(\star)$, a computable double sequence of rational-complex numbers $(s_{n,m})_{n,m\in\mathbb{N}}$, and a (total) μ-recursive function $\xi\colon \mathbb{N} \to \mathbb{N}$, let (1) be satisfied. Then, for all $M \in \mathbb{N}$, the sequence

$$\boldsymbol{x}_M := \sum_{n=0}^{\xi(M)} \underbrace{s_{n,M}}_{:=z_n} \cdot \boldsymbol{\delta}_{\nu(n)}$$

is an elementary computable with coefficients $z_n,\ n \in \mathbb{N}$. The sequence of sequences $(\boldsymbol{x}_M)_{M\in\mathbb{N}}$ is called a computable sequence of elementary computable sequences. Further, for all $M \in \mathbb{N}$, we have $\|\boldsymbol{x} - \boldsymbol{x}_M\|_\star < 2^{-M}$. In general, a sequence $\boldsymbol{x} \in \ell(\star)$ is computable in $\ell(\star)$ if and only if there exists a computable sequence of elementary computable sequences $(\boldsymbol{x}_M)_{M\in\mathbb{N}}$ that converges effectively towards $\boldsymbol{x}$, with respect to $\|\cdot\|_\star$ and a suitable effective modulus of convergence.

For $\boldsymbol{f} \in \mathcal{CB}(\star)$, a computable double sequence of rational-complex numbers $(s'_{n,m})_{n,m\in\mathbb{N}}$, and a (total) μ-recursive function $\xi' : \mathbb{N} \to \mathbb{N}$, let (2) be satisfied. Analogously, for all $M \in \mathbb{N}$, the function

$$\boldsymbol{f}_M := \boldsymbol{T}_\star \sum_{n=0}^{\xi'(M)} \underbrace{s'_{n,M}}_{:=z'_n} \cdot \boldsymbol{\delta}_{\nu(n)} = \sum_{n=0}^{\xi'(M)} z'_n \cdot \mathrm{sinc}_{\nu(n)} \tag{3}$$

is an elementary computable with coefficients z'_n, $n \in \mathbb{N}$. The sequence of functions $(\boldsymbol{f}_M)_{M\in\mathbb{N}}$ is called a computable sequence of elementary computable functions. Further, for all $M \in \mathbb{N}$, we have $\|\boldsymbol{f} - \boldsymbol{f}_M\|_\star < 2^{-M}$. In general, a function $\boldsymbol{f} \in \mathcal{B}(\star)$ is computable in $\mathcal{B}(\star)$ if and only if there exists a computable sequence of elementary computable functions $(\boldsymbol{f}_M)_{M\in\mathbb{N}}$ that converges effectively towards $\boldsymbol{f}$, with respect to $\|\cdot\|_\star$ and a suitable effective modulus of convergence.

Throughout the remainder of this article, we will prove the following: There exist $\boldsymbol{f} = \boldsymbol{T}_\star \boldsymbol{x} \in \mathcal{CB}(\star)$ and $(\boldsymbol{x}_M)_{M\in\mathbb{N}}$ as above, such that we have $\lim_{M\to\infty} \|\boldsymbol{f} - \boldsymbol{T}_\star \boldsymbol{x}_M\| \neq 0$, despite $(\boldsymbol{x}_M)_{M\in\mathbb{N}}$ converging effectively towards $\boldsymbol{x}$. In other words, even if $(\boldsymbol{x}_M)_{M\in\mathbb{N}}$ converges effectively towards $\boldsymbol{x} = \boldsymbol{S}_\star \boldsymbol{f}$, the computable sequence of elementary computable functions $(\boldsymbol{T}_\star \boldsymbol{x}_M)_{M\in\mathbb{N}}$ does *not* necessarily converge towards $\boldsymbol{T}_\star \boldsymbol{x}$. Section 3 will discuss the consequences of this observation extensively. To this end, we will now establish two preliminary lemmas, the proofs of which we provide in Appendix A.

Lemma 1. *Let $\Omega \subset \mathbb{N}$ be a recursively enumerable set. There exists a* (not *necessarily computable) sequence $(\boldsymbol{f}_m)_{m\in\mathbb{N}}$ of elementary computable functions in $\mathcal{B}^\infty_{0,\pi}$ that satisfies the following: the sequence $(\boldsymbol{x}_m)_{m\in\mathbb{N}} = (\boldsymbol{S}_\infty \boldsymbol{f}_m)_{m\in\mathbb{N}}$ is a computable sequence of sequences in $\mathcal{C}\ell^\infty_0$, and, for all $m \in \mathbb{N}$, we have*

$$\|\boldsymbol{f}_m\|_\infty \begin{cases} \geq 1, & \text{if } \mathbb{1}_\Omega(m) = 1, \\ = 0, & \text{if } \mathbb{1}_\Omega(m) = 0. \end{cases}$$

Lemma 2. *Let $\Omega \subset \mathbb{N}$ be a recursively enumerable set. There exists a* (not *necessarily computable) sequence $(\boldsymbol{f}_m)_{m\in\mathbb{N}}$ of elementary computable functions in $\mathcal{B}^1_\pi$ that satisfies the following: the sequence $(\boldsymbol{x}_m)_{m\in\mathbb{N}} = (\boldsymbol{S}_1 \boldsymbol{f}_m)_{m\in\mathbb{N}}$ is a computable sequence of sequences in $\mathcal{C}\ell^1$, and for all $m \in \mathbb{N}$, we have*

$$\|\boldsymbol{f}_m\|_1 \begin{cases} \geq 1, & \text{if } \mathbb{1}_\Omega(m) = 1, \\ = 0, & \text{if } \mathbb{1}_\Omega(m) = 0. \end{cases}$$

For the general definition of computable sequences of abstract objects (such as those used in Lemmas 1 and 2), see below.

Recall this article's fundamental problem statement from Section 1: Given an application that requires the processing of analog information, find a sufficient way to represent the information on digital hardware. In order to provide a solution, we require a general formalization of how to represent information on Turing machines, employing the natural numbers as their "atomic" numerical object. The authors advise readers that this formalization is somewhat abstract, but necessary for a mathematically rigorous treatment. After establishing the formalization in its abstract form, we will put it in the context of SSI, allowing for a less cumbersome and more intuitive treatment.

For two μ-recursive functions $g_1, g_2 : \mathbb{N}^n \supseteq\!\to \mathbb{N}$, we write $g_1 = g_2$ if $\mathrm{dom}(g_1) = \mathrm{dom}(g_2)$ is satisfied, and for all $(m_1, \ldots, m_n) \in \mathrm{dom}(g_1)$, we have $g_1(m_1, \ldots, m_n) = g_2(m_1, \ldots, m_n)$. Furthermore, for ease of notation, we will make use of anonymous mappings. In general, an explicit definition of a mapping is of the form $G : \mathcal{A} \supseteq\!\to \mathcal{B}, a \mapsto G(a) :=$ "EXPR(a)", where $\mathcal{A}$ and $\mathcal{B}$ are arbitrary sets, and "EXPR(a)" is the term defining G, such as, for example, "$a + a$", "a^2", "$\ln(a)$", and so on. If the context determines $\mathcal{A}$ and $\mathcal{B}$ without ambiguity, but does not require providing an explicit definition, we simply write $(a \mapsto \text{“EXPR}(a)\text{”})$ to denote the respective mapping.

The formalization of how to represent information on Turing machines builds upon the existence of universal μ-recursive functions $U : \mathbb{N} \times \mathbb{N} \supseteq\!\rightarrow \mathbb{N}$, an arbitrary one of which we fix for the remainder of this article. Then, for every μ-recursive function $g : \mathbb{N}^n \supseteq\!\rightarrow \mathbb{N}$, $n \in \mathbb{N}$, the following holds true: there exists a “program” $M \in \mathbb{N}$ such that the function $U^n_M : \mathbb{N}^n \supseteq\!\rightarrow \mathbb{N}$, $(m_1, \dots, m_n) \mapsto U^n_M(m_1, \dots, m_n)$, defined through

$$U^n_M(m_1, \dots, m_n) := U(\cdots U(U(M, m_1), m_2) \cdots, m_n)$$

satisfies $g = U^n_M$. Accordingly, for all $n \in \mathbb{N}$, the universal μ-recursive function U provides an equivalence relation on $\mathbb{N}$: for $M, K \in \mathbb{N}$, we have $M \equiv K$ if $U^n_M = U^n_K$. Evidently, the equivalence-relation's quotient set $\{\{K \in \mathbb{N} : K \equiv M\} : M \in \mathbb{N}\}$ is in one-to-one correspondence with the the set $\mathcal{U}(n)$. We denote

$$[\cdot]^n_{\mathfrak{U}} : \mathbb{N} \rightarrow \mathcal{U}(n), M \mapsto [M]^n_{\mathfrak{U}} := U^n_M,$$

which hints towards the usual notation for quotient sets in the context of equivalence relations: we have $[M]^n_{\mathfrak{U}} = [K]^n_{\mathfrak{U}}$ if and only if $M \equiv K$. We call the set

$$\mathfrak{U}(n) := \left\{ \left(g, \mathbb{N}, [\cdot]^n_{\mathfrak{U}}\right) : g \in \mathcal{U}(n) \right\}$$

a machine-readable language for the set $\mathcal{U}(n)$, and $M \in \mathbb{N}$ is a machine-readable description of $U^n_M \in \mathcal{U}(n)$. Furthermore, for $n, m \in \mathbb{N}$, a mapping $G : \mathfrak{U}(m) \supseteq\!\rightarrow \mathfrak{U}(n)$ is called computable if there exists a μ-recursive function $g : \mathbb{N} \supseteq\!\rightarrow \mathbb{N}$ such that, for all $M \in \mathbb{N}$ with $U^m_M \in \mathrm{dom}(G)$, we have

$$G\left(h, \mathbb{N}, [\cdot]^m_{\mathfrak{U}}\right) = \left(U^n_{g(M)}, \mathbb{N}, [\cdot]^n_{\mathfrak{U}}\right).$$

For $m_1, \dots, m_k \in \mathbb{N}$, we can extend the principle to analogously mappings of the form $G : \mathfrak{U}(m_1) \times \ldots \times \mathfrak{U}(m_k) \supseteq\!\rightarrow \mathfrak{U}(n)$. Observe that arithmetic operations on μ-recursive functions, such as

$$\begin{aligned}
&(g_1, g_2) \mapsto (n \mapsto g_1(n) + g_2(n)), \\
&(g_1, g_2) \mapsto (n \mapsto g_1(n) g_2(n)), \\
&(g_1, g_2) \mapsto (n \mapsto \max\{g_1(n), g_2(n)\}), \\
&(g_1, g_2) \mapsto (n \mapsto \min\{g_1(n), g_2(n)\}),
\end{aligned}$$

and so on, as well as composition, primitive recursion, and unbounded search (see above), when seen as operations on μ-recursive functions, provide computable mappings in the sense of the definition above. Throughout the remainder of the article, we will make implicit use of the computability of mappings of the form $G : \mathfrak{U}(m_1) \times \ldots \times \mathfrak{U}(m_k) \supseteq\!\rightarrow \mathfrak{U}(n)$ on many occasions. For details, we refer to the SMN-Theorem, c.f. Theorem 3.5, p. 16 in [14].

Following the principle of $\mathfrak{U}(n)$, $n \in \mathbb{N}$, we can now define general machine-readable languages and general computable mappings through an inductive scheme. For all $n \in \mathbb{N}$, the set of n-tuples of natural numbers, $\mathbb{N}^n$, is an atomic machine-readable language, and for all $n, k \in \mathbb{N}$, a mapping $G : \mathbb{N}^n \supseteq\!\rightarrow \mathbb{N}^k$, $(m_1, \dots, m_n) \mapsto G(m_1, \dots, m_n)$ is called atomically computable if there exist functions $g_1, \dots, g_k \in \mathcal{U}(n)$ such that $\mathrm{dom}(G)$ is a (possibly improper) subset of $\mathrm{dom}(g_1) \cap \ldots \cap \mathrm{dom}(g_k)$, and we have

$$G(m_1, \dots, m_n) = (g_1(m_1, \dots, m_n), \dots, g_k(m_1, \dots, m_n))$$

for all $(m_1, \ldots, m_n) \in \mathrm{dom}(G)$. A (*non*-atomic) machine-readable language for the (abstract) set $\mathcal{A}$ is of the form

$$\mathfrak{A} := \Big\{ \Big(a, \Lambda_{\mathfrak{A}}, [\cdot]_{\mathfrak{A}}\Big) : a \in \mathcal{A} \Big\},$$

where $\Lambda_{\mathfrak{A}}$ is a machine-readable language and $[\cdot]_{\mathfrak{A}} : \Lambda_{\mathfrak{A}} \supseteq\!\rightarrow \mathcal{A}$ is a partial surjective mapping. Further, $\lambda \in \mathrm{dom}([\cdot]_{\mathfrak{A}})$ is called a machine-readable description of $[\lambda]_{\mathfrak{A}} \in \mathcal{A}$. Again, "$[\cdot]_{\mathfrak{A}}$" hints towards the usual notation for quotient sets: for $\lambda_1, \lambda_2 \in \mathrm{dom}([\cdot]_{\mathfrak{A}})$, we have $\lambda_1 \equiv \lambda_2$ if and only if $[\lambda_1]_{\mathfrak{A}} = [\lambda_2]_{\mathfrak{A}}$, i.e., λ_1 and λ_2 are two machine-readable descriptions of the same abstract object. However, since $[\cdot]_{\mathfrak{A}}$ is generally partial, so is the induced equivalence relation. Finally, mapping $G_1 : \mathfrak{A} \supseteq\!\rightarrow \mathfrak{B}$, where $\mathfrak{A}$ and $\mathfrak{B}$ are machine-readable languages, is called (*non*-atomically) computable if there exists a computable mapping $G_2 : \Lambda_{\mathfrak{A}} \supseteq\!\rightarrow \Lambda_{\mathfrak{B}}$ such that, for all $\lambda \in \mathrm{dom}([\cdot]_{\mathfrak{A}})$ with $([\lambda]_{\mathfrak{A}}, \Lambda_{\mathfrak{A}}, [\cdot]_{\mathfrak{A}}) \in \mathrm{dom}(G_1)$, we have

$$G_1\Big([\lambda]_{\mathfrak{A}}, \Lambda_{\mathfrak{A}}, [\cdot]_{\mathfrak{A}}\Big) = \Big([G_2(\lambda)]_{\mathfrak{B}}, \Lambda_{\mathfrak{B}}, [\cdot]_{\mathfrak{B}}\Big).$$

Unless defined otherwise, a sequence $(\mathfrak{a}_n)_{n\in\mathbb{N}}$ of elements of $\mathfrak{A}$ is called computable if there exists a (total) computable mapping $(n \mapsto \mathfrak{a}_n)$. Observe that if $\mathfrak{A}$ and $\mathfrak{B}$ are machine-readable languages for arbitrary abstract sets $\mathcal{A}$ and $\mathcal{B}$, respectively, G_1 naturally induces a mapping $G_1 : \mathcal{A} \supseteq\!\rightarrow \mathcal{B}$ according to

$$G_1\Big(a, \Lambda_{\mathfrak{A}}, [\cdot]_{\mathfrak{A}}\Big) = \Big(b, \Lambda_{\mathfrak{B}}, [\cdot]_{\mathfrak{B}}\Big) \quad \Leftrightarrow \quad G_1(a) = b.$$

and vice versa. If, according to the specific context, there is no danger of ambiguity, we will not distinguish between $G_1 : \mathfrak{A} \supseteq\!\rightarrow \mathfrak{B}$ and $G_1 : \mathcal{A} \supseteq\!\rightarrow \mathcal{B}$.

In essence, a machine-readable language is a formal specification of how to represent abstract information on digital hardware, such that we can (in principle) trace this specification down to the level of tuples of natural numbers and fundamental operations thereon. Upon fixing a suitable μ-recursive pairing function (Chapter 1.4, p. 12 in [16]), i.e., a bijective mapping

$$(\langle\cdot\rangle_1, \langle\cdot\rangle_2) : \mathbb{N} \mapsto \mathbb{N}^2,\ m \mapsto (\langle m\rangle_1, \langle m\rangle_2)$$

with $\langle\cdot\rangle_1, \langle\cdot\rangle_2 \in \mathcal{U}(1)$, every machine-readable language $\mathfrak{A}$ exhibits a canonical numbering, i.e., computable surjective mapping $\varphi_{\mathfrak{A}} : \mathbb{N} \supseteq\!\rightarrow \mathfrak{A}$, defined in an inductive manner:

- If $\mathfrak{A}$ is an atomic machine-readable language, i.e., we have $\mathfrak{A} = \mathbb{N}^n$ for some number $n \in \mathbb{N}$, we define

$$\varphi_{\mathfrak{A}}(m) := \begin{cases} m, & \text{if } n = 1, \\ (\langle m\rangle_1, \langle m\rangle_2) & \text{if } n = 2, \\ (\langle m\rangle_1, \langle\langle m\rangle_2\rangle_1, \langle\langle m\rangle_2\rangle_2), & \text{if } n = 3, \\ (\langle m\rangle_1, \langle\langle m\rangle_2\rangle_1, \ldots, \langle\langle m\rangle_2^{n-2}\rangle_1, \langle\langle m\rangle_2^{n-2}\rangle_2), & \text{if } n \geq 4, \end{cases}$$

  where $\langle m\rangle_2^n$ denotes the n-fold successive application of $\langle\cdot\rangle_2$ to m;
- For a *non*-atomic machine-readable language $\mathfrak{A} = \{(a, \mathfrak{B}, [\cdot]_{\mathfrak{A}}) : a \in \mathcal{A}\}$ and a general machine-readable language $\mathfrak{B}$ with canonical numbering $\varphi_{\mathfrak{B}} : \mathbb{N} \supseteq\!\rightarrow \mathfrak{B}$, we define

$$\varphi_{\mathfrak{A}}(m) := (a, \mathfrak{B}, [\cdot]_{\mathfrak{A}}) \quad :\Leftrightarrow \quad [\varphi_{\mathfrak{B}}(m)]_{\mathfrak{A}} = a,$$

  with $\mathrm{dom}(\varphi_{\mathfrak{A}}) := \{m \in \mathrm{dom}(\varphi_{\mathfrak{B}}) : \varphi_{\mathfrak{B}}(m) \in \mathrm{dom}([\cdot]_{\mathfrak{A}})\}$ accordingly.

Among other things, and together with the relevant pairing function $(\langle\cdot\rangle_1, \langle\cdot\rangle_2)$, the canonical numbering facilitates the definition of machine-readable languages for tuples of the form $(a_1, a_2) \in \mathcal{A}_1 \times \mathcal{A}_2$, provided we have already defined machine-readable languages for the abstract sets $\mathcal{A}_1$ and $\mathcal{A}_2$.

Referring to this article's fundamental problem statement, if we want to represent abstract information on a digital machine in a sufficient way, we need to specify a machine-readable language for the relevant abstract set, and then investigate the language's structural properties. Albeit rarely explicit, this principle is used throughout the literature of computable analysis. In the context of Banach spaces, it is strongly related to the definitions of computability structures (Chapter 2.1, p. 80ff in [15]). Further, any canonical numbering $\varphi_{\mathfrak{A}}$ as defined above is essentially a numbering in the sense of a concept that is fundamental in computability theory (Chapter 1.4, p. 12 in [16]). As indicated before, formal approaches of this form are necessary for a mathematically rigorous theory of computable analysis. Yet, they are somewhat cumbersome in use. In Definition 3, for example, we have implicitly introduced a machine-readable language for the set of computable real numbers by defining the relation $[(r_n)_{n\in\mathbb{N}}, \xi]_{\mathfrak{R}} = x$. This convention is an abuse of notation regarding the just-established formalization of machine-readable languages. Strictly speaking, we first have to define machine-readable languages for the set of triples $(g, h_1, h_2) : g, h_1, h_2 \in \mathcal{U}(1)$. Then, we have to define a machine-readable language for the set of computable sequences of rational numbers. Finally, we have to define a machine-readable language for the set of pairs $((r_n)_{n\in\mathbb{N}}, \xi)$ as above, based on which we can define the machine-readable language for the set of computable real numbers in the sense of Definition 3. Intuitively, on the other hand, it is evident from Definition 3 that we describe a computable real number by a suitable pair $((r_n)_{n\in\mathbb{N}}, \xi)$, and we can implement computable mappings on computable real numbers by applying μ-recursive functions to the "programs" (with respect to the universal μ-recursive function U) of the underlying quadruple (g, h_1, h_2, ξ). Keeping the formal definition in mind, we will, with some abuse of nomenclature and notation, employ the following conventions for mathematical ease:

- A standard description of computable real number x consists of a pair $((r_n)_{n\in\mathbb{N}}, \xi)$ that characterizes x in the sense of Definition 3, and we write $x = [(r_n)_{n\in\mathbb{N}}, \xi]_{\mathfrak{R}}$. We denote the associated standard machine-readable language by $\mathfrak{R}$;
- A standard description of computable complex number z consists of a pair $((s_n)_{n\in\mathbb{N}}, \xi)$ that characterizes z in the sense of Definition 4, and we write $z = [(s_n)_{n\in\mathbb{N}}, \xi]_{\mathfrak{C}}$. We denote the associated standard machine-readable language by $\mathfrak{C}$.

For the set $\mathcal{CB}(\star)$, the same convention applies. However, based on the generalized form of Theorem 1, we have two different machine-readable languages available:

- A discrete-time description of $\boldsymbol{f} \in \mathcal{CB}(\star)$ consists of a pair $((s_{n,m})_{n,m\in\mathbb{N}}, \xi)$ that characterizes $\boldsymbol{f}$ in the sense of Definition 5, and we write $\boldsymbol{f} = [(s_{n,m})_{n,m\in\mathbb{N}}, \xi]^{\star}_{\mathfrak{X}}$. We denote the associated discrete-time machine-readable language by $\mathfrak{X}_\star$;
- A continuous-time description of $\boldsymbol{f} \in \mathcal{CB}(\star)$ consists of a pair $((s'_{n,m})_{n,m\in\mathbb{N}}, \xi')$ that characterizes $\boldsymbol{f}$ in the sense of Definition 6, and we write $\boldsymbol{f} = [(s'_{n,m})_{n,m\in\mathbb{N}}, \xi']^{\star}_{\mathfrak{F}}$. We denote the associated continuous-time machine-readable language by $\mathfrak{F}_\star$.

Returning to the abstract theory of machine-readable language once more, we can consider the general case of an abstract set $\mathcal{A}$ with more than one associated machine-readable language: in fact, even though any machine-readable language has necessarily only countably many elements, there exists an uncountable number of machine-readable languages for any nontrivial abstract set. Consider machine-readable languages $\mathfrak{A}_1$ and $\mathfrak{A}_2$ for the set $\mathcal{A}$, and define the corresponding identity mapping

$$\mathrm{Id}_{1,2} : \mathfrak{A}_1 \to \mathfrak{A}_2, \quad (a, \Lambda_1, [\cdot]_1) \mapsto \mathrm{Id}_{1,2}(a, \Lambda_1, [\cdot]_1) := (a, \Lambda_2, [\cdot]_2).$$

We can now define a partial quasiorder on the class of machine-readable languages for the set $\mathcal{A}$ as follows:

$$\begin{aligned}
\mathfrak{A}_1 \succeq \mathfrak{A}_2 \quad &:\Leftrightarrow \quad \mathrm{Id}_{1,2} \text{ is computable,} \\
\mathfrak{A}_1 \succ \mathfrak{A}_2 \quad &:\Leftrightarrow \quad \mathrm{Id}_{1,2} \text{ is computable, but } \mathrm{Id}_{2,1} \text{ is uncomputable,} \\
\mathfrak{A}_1 \simeq \mathfrak{A}_2 \quad &:\Leftrightarrow \quad \mathrm{Id}_{1,2} \text{ and } \mathrm{Id}_{2,1} \text{ are computable.}
\end{aligned}$$

Intuitively, if $\mathfrak{A}_1 \succeq \mathfrak{A}_2$, we can find an algorithm that transforms any description λ_1 of any object $a \in \mathfrak{A}$ in the language $\mathfrak{A}_1$ into a description λ_2 of the same object in the language $\mathfrak{A}_2$. For any computable mapping $G : \mathfrak{A}_2 \supseteq\!\!\rightarrow \mathfrak{B}$, where $\mathfrak{B}$ is an arbitrary machine-readable language, the composition $G \circ \mathrm{Id}_{1,2}$ is computable as well. Thus, any computational problem we can solve by means of the language $\mathfrak{A}_2$, we can also solve by means of the language $\mathfrak{A}_1$. In view of this article's fundamental problem statement, we can distinguish four cases:

1. If $\mathfrak{A}_1 \succ \mathfrak{A}_2$, descriptions in the language $\mathfrak{A}_1$ contain more information than descriptions in the language $\mathfrak{A}_2$;
2. If $\mathfrak{A}_1 \prec \mathfrak{A}_2$, descriptions in the language $\mathfrak{A}_1$ contain less information than descriptions in the language $\mathfrak{A}_2$;
3. If $\mathfrak{A}_1 \simeq \mathfrak{A}_2$, descriptions in the language $\mathfrak{A}_1$ contain the same information as descriptions in the language $\mathfrak{A}_2$;
4. If neither of the previous cases holds, descriptions in the language $\mathfrak{A}_1$ contain different information than descriptions in the language $\mathfrak{A}_2$.

The remainder of this article will address the relationship between the languages $[\cdot]^\star_{\mathfrak{X}}$ and $[\cdot]^\star_{\mathfrak{F}}$. The generalized Shannon equivalence motivates the engineering paradigm that processing any (bandlimited) analog signal can be entirely moved to the discrete-time domain, provided that we have a sequence of sampling values with sufficient quantization accuracy available. However, as stated before, the generalized Shannon equivalence is an abstract analytical concept, formalized in terms of the Banach-space operators $\boldsymbol{S}_\star$ and $\boldsymbol{T}_\star$. Previously in this section, we have stated that the (informal) question of whether the generalized Shannon equivalence also holds true on the algorithmic level corresponds to the (formal) question of whether the operators $\boldsymbol{S}_\star$ and $\boldsymbol{T}_\star$ are computable (in the machine-readable language under consideration). In Section 3, we will establish that the computability of $\boldsymbol{S}_\star$ and $\boldsymbol{T}_\star$ is essentially a rephrasing of the relationship between $[\cdot]^\star_{\mathfrak{X}}$ and $[\cdot]^\star_{\mathfrak{F}}$.

Before concluding the present section, observe that we have $\mathcal{CB}^1_\pi \subset \mathcal{CB}^\infty_{0,\pi}$. Further, for all $\boldsymbol{x} \in \ell^1$, we have $\|\boldsymbol{x}\|_1 \geq \|\boldsymbol{x}\|_\infty$, and for all $\boldsymbol{f} \in \mathcal{B}^1_\pi$, we have $\|\boldsymbol{f}\|_1 \geq \|\boldsymbol{f}\|_\infty$, implying

$$\mathfrak{X}_1 \succeq \mathfrak{X}_\infty|\mathcal{CB}^1_\pi \quad \text{and} \quad \mathfrak{F}_1 \succeq \mathfrak{F}_\infty|\mathcal{CB}^1_\pi$$

for the restrictions $\mathfrak{X}_\infty|\mathcal{CB}^1_\pi$ and $\mathfrak{F}_\infty|\mathcal{CB}^1_\pi$ of $\mathfrak{X}_\infty$ and $\mathfrak{F}_\infty$ to elements of $\mathcal{CB}^1_\pi$. We will briefly return to these inequalities in Section 3.

3. Results

In the scope of our theory, digital twinning involves an abstract set $\mathcal{A}$ and a corresponding machine-readable language $\mathfrak{A}$, both of which are results of how the relevant technological system is modeled from an engineering perspective. When the system is operated, it gives rise to a (*not* necessarily computable) sequence $(\mathfrak{a}_t)_{t\in\mathbb{N}}$ of elements of $\mathfrak{A}$, which ultimately emerges from a successive measurement process. In each instance $t \in \mathbb{N}$, $\mathfrak{a}_t$ should, in one way or another, correspond to the instantaneous state of the physical technological system. The details of this correspondence are, again, a result of modeling. For example, referring to Figure 1, denote the robot's instantaneous position at time $t \in \mathbb{N}$ by $\vec{\varkappa}_t \in \mathbb{R}^2$. Further, denote by

$$\mathfrak{K} := \left\{ \left(\vec{\varkappa}, \Lambda_{\mathfrak{K}}, [\cdot]_{\mathfrak{K}}\right) : \vec{\varkappa} \in \widetilde{\mathbb{R}}^2 \right\}$$

a suitable machine-readable language corresponding to a countable set $\widetilde{\mathbb{R}}^2 \subset \mathbb{R}^2$ of our choice, and define

$$d\Big(\vec{\varkappa}_t, (\vec{\varkappa}, \Lambda_{\mathfrak{K}}, [\cdot]_{\mathfrak{K}})\Big) := \|\vec{\varkappa}_t - \vec{\varkappa}\|_2, \qquad \vec{\varkappa} \in \widetilde{\mathbb{R}}^2$$

Then, we might require that for a suitable computable mapping $G : \mathfrak{A} \supseteq\!\!\rightarrow \mathfrak{K}, \mathfrak{a} \mapsto G(\mathfrak{a})$ and some $\epsilon > 0$, we have

$$d\Big(\vec{\varkappa}_t, G\big(\mathfrak{a}_t\big)\Big) < \epsilon$$

for all $t \in \mathbb{N}$. Intuitively, we consider the robot's instantaneous position a relevant property, and thus want to be able to recover it from the robot's virtual twin up to a certain error at any instance in time.

Recall that Figure 1 (Right) illustrates the instantaneous discrete-time impulse response of the transmission channel between the robot and the receiving end. Commonly, wireless transmission channels are assumed linear, i.e., their behavior is determined entirely by the relevant impulse response. Further, wireless communication systems are commonly restricted to a specific frequency range, i.e., the transmission is bandlimited. Accordingly, for all $t \in \mathbb{N}$, the instantaneous physical transmission channel uniquely corresponds to a bandlimited signal $f_{\mathrm{ph},t} \in \mathcal{B}^1_\sigma$, $0 < \sigma < \infty$. Without loss of generality, we again consider $\sigma = \pi$ in the following. For $(f, \Lambda_{\mathfrak{X}}, [\cdot]^1_{\mathfrak{X}}) \in \mathfrak{X}_1$ and $(f, \Lambda_{\mathfrak{F}}, [\cdot]^1_{\mathfrak{F}}) \in \mathfrak{F}_1$, define

$$d\Big(f_{\mathrm{ph},t}, (f, \Lambda_{\mathfrak{X}}, [\cdot]^1_{\mathfrak{X}})\Big) = d\Big(f_{\mathrm{ph},t}, (f, \Lambda_{\mathfrak{F}}, [\cdot]^1_{\mathfrak{F}})\Big) := \|f_{\mathrm{ph},t} - f\|_1.$$

Then, for some $\epsilon_{\mathrm{ph}} > 0$ and suitable computable mappings $G_{\mathrm{dt}} : \mathfrak{A} \supseteq\!\!\rightarrow \mathfrak{X}_1, \mathfrak{a} \mapsto G_{\mathrm{dt}}(\mathfrak{a})$ and $G_{\mathrm{ct}} : \mathfrak{A} \supseteq\!\!\rightarrow \mathfrak{F}_1, \mathfrak{a} \mapsto G_{\mathrm{ct}}(\mathfrak{a})$, we might again require

$$d\Big(f_{\mathrm{ph},t}, G_{\mathrm{dt}}\big(\mathfrak{a}_t\big)\Big) < \epsilon_{\mathrm{ph}}, \qquad d\Big(f_{\mathrm{ph},t}, G_{\mathrm{ct}}\big(\mathfrak{a}_t\big)\Big) < \epsilon_{\mathrm{ph}} \tag{4}$$

respectively, to hold for all $t \in \mathbb{N}$. Observe that (4) is a purely analytical relation, describing the requirement that $G_{\mathrm{dt}}(\mathfrak{a}_t)$ and $G_{\mathrm{ct}}(\mathfrak{a}_t)$ at each time $t \in \mathbb{N}$ provide a sufficiently accurate approximation to the instantaneous properties of the physical channel. The "true" sequence $(f_{\mathrm{ph},t})_{t\in\mathbb{N}}$ of channel characteristics does *not* need to consist of computable components. In the design process of a digital-twin system, such as that shown in Figure 1, the responsible engineer has to prove—based on mathematical modeling—that, during the system's operation, the sequence $(G_{\mathrm{dt}}(\mathfrak{a}_t))_{t\in\mathbb{N}}$, $(G_{\mathrm{ct}}(\mathfrak{a}_t))_{t\in\mathbb{N}}$, respectively, will satisfy (4). Recall that, by definition, both $\mathfrak{X}_1$ and $\mathfrak{F}_1$ are machine-readable languages for the set $\mathcal{CB}^1_\pi$. Hence, according to Section 2, G_{dt} and G_{ct} each induce a mapping $G_{\mathrm{dt}} : \mathcal{A} \supseteq\!\!\rightarrow \mathcal{CB}^1_\pi$, $G_{\mathrm{ct}} : \mathcal{A} \supseteq\!\!\rightarrow \mathcal{CB}^1_\pi$, respectively. In the following, we assume G_{dt} and G_{ct} to be the same, and we denote $G_{\mathrm{dt}}(a_t) = G_{\mathrm{ct}}(a_t) =: f_t$. The system's design process will then include the choice between implementing $(f_t)_{t\in\mathbb{N}}$ using $(G_{\mathrm{dt}}(\mathfrak{a}_t))_{t\in\mathbb{N}}$—i.e., approximating $(f_{\mathrm{ph}})_{t\in\mathbb{N}}$ through discrete-time descriptions—or using $(G_{\mathrm{ct}}(\mathfrak{a}_t))_{t\in\mathbb{N}}$—i.e., approximating $(f_{\mathrm{ph}})_{t\in\mathbb{N}}$ through continuous-time descriptions—the implications of which we will analyze subsequently.

Motivated by the generalized Shannon equivalence, the textbook approach considers discrete-time descriptions of bandlimited signals. As indicated before, the evident advantage of this paradigm consists of computational "convenience". In a simplified manner, the standard (abstract) engineering model—that is, without considering questions of computability, yet—for the wireless communication (sub)system from Figure 1 may look as follows. At time $t \in \mathbb{N}$, the robot aims to transmit one of $M \in \mathbb{N}$ messages to the receiving end, for which he employs an encoding scheme $\mathcal{E}_t : \{1, \ldots, M\} \to \mathrm{dom}(\boldsymbol{T}_\infty), m \mapsto \mathcal{E}_t(m) := \boldsymbol{y}_{t,m}$. For reasons of simplicity, we summarize processes such as encoding (in the sense of information theory), channel precoding, modulation, and pulse shaping in this step. Setting $\boldsymbol{x}_{\mathrm{ph},t} := \boldsymbol{S}_1 \boldsymbol{f}_{\mathrm{ph},t}$, the signal at the receiving end is of the form

$$\boldsymbol{y}_{\mathrm{re},t} := \boldsymbol{w} + \left(\boldsymbol{x}_{\mathrm{ph},t} * \boldsymbol{y}_{t,m}\right),$$

where $\boldsymbol{w} \in \ell_0^\infty$ is a sequence of additive noise-like disturbances, and $\boldsymbol{x}_{\mathrm{ph},t} * \boldsymbol{y}_{t,m}$ denotes the convolution of $\boldsymbol{x}_{\mathrm{ph},t}$ and $\boldsymbol{y}_{t,m}$. The receiving end then employs a decoding scheme $\mathcal{D}_t : \ell_0^\infty \to \{1, \ldots, M\}$, $\boldsymbol{y}_{\mathrm{re},t} \mapsto \mathcal{D}_t(\boldsymbol{y}_{\mathrm{re},t})$. Again, for reasons of simplicity, we summarize processes such as demodulation, filtering, and decoding (in the sense of information theory) in this step. Region-based decoding is a common way of implementing $\mathcal{D}_t$, in which case $\mathcal{D}_t$ is of the form

$$\mathcal{D}_t(\boldsymbol{y}_{\mathrm{re},t}) := D\left(\arg\min_{n \in \{1,\ldots,N\}} \left\|\boldsymbol{y}_{\mathrm{re},t} - \boldsymbol{y}_{\mathrm{de},t,n}\right\|_\infty\right)$$

where $\boldsymbol{y}_{\mathrm{de},t,1}, \ldots, \boldsymbol{y}_{\mathrm{de},t,N} \in \ell_0^\infty$, $N \in \mathbb{N}$, is a list of reference signals and $D : \{1, \ldots, N\} \to \{1, \ldots, M\}$ is a mapping that assigns each reference signal an inferred message.

In the setup depicted in Figure 1, the choice of the pair $(\mathcal{E}_t, \mathcal{D}_t)$ will generally be based on the robot's digital twin, i.e., (assuming both the receiving end and the robot itself have access to the sequence $(\mathfrak{a}_t)_{t\in\mathbb{N}}$), it will be implemented through computable mappings $\mathfrak{a}_t \mapsto \mathcal{E}_t$, $\mathfrak{a}_t \mapsto \mathcal{D}_t$, involving an optimization of some kind. For example, we may aim to choose $\boldsymbol{y}_{t,1}, \ldots, \boldsymbol{y}_{t,M}$ and $\boldsymbol{y}_{\mathrm{de},t,1}, \ldots, \boldsymbol{y}_{\mathrm{de},t,N}$ such that

$$m = D\left(\arg\min_{n \in \{1,\ldots,N\}} \left\|\boldsymbol{w} + \left(\boldsymbol{x}_{\mathrm{ph},t} * \boldsymbol{y}_{t,m}\right) - \boldsymbol{y}_{\mathrm{de},t,n}\right\|_\infty\right)$$

for all $m \in \{1, \ldots, M\}$ and all $\boldsymbol{w} \in \ell_0^\infty$ that satisfy $\|\boldsymbol{w}\|_\infty < \epsilon_{\boldsymbol{w}}$ for some $\epsilon_{\boldsymbol{w}} > 0$. Accordingly, upon implementing the communication system, we require $\boldsymbol{y}_{\mathrm{de},t,1}, \ldots, \boldsymbol{y}_{\mathrm{de},t,N} \in \mathcal{C}\ell_0^\infty$ and $\boldsymbol{y}_{t,1}, \ldots, \boldsymbol{y}_{t,M} \in \mathrm{dom}(\boldsymbol{T}_\infty) \cap \mathcal{C}\ell_0^\infty$. In theory, we can then entirely neglect the analog part of the real system, i.e., the transmission of the signals through the physical (analog) medium, in the design of our signal processing algorithms.

Again, recall that both $\mathfrak{X}_\star$ and $\mathfrak{F}_\star$ are machine-readable languages for the set $\mathcal{CB}(\star)$. In particular, per its definition, $\mathfrak{X}_\star$ does *not* include the entirety of $\mathcal{C}\ell(\star)$, which appears unnecessary following the discussion above. In fact, there does not seem to be an obvious reason to consider the space $\mathcal{CB}(\star)$ at all. Assuming we are able to prove that the implementation of our systems satisfies (4), we even can perform the optimization of $(\mathcal{E}_t, \mathcal{D}_t)$ within the discrete-time domain. Upon closer inspection, we find that despite the discussion above, there exists a variety of reasons why we cannot neglect the analog part of the communication system. We will discuss several of them in the following:

1. As mentioned above, any real implementation of our system will involve a steps of digital-to-analog conversion of the transmission signal $\boldsymbol{y}_{t,m}$. However, any real-world digital-to-analog converter will not be able to synthesize the signal $\boldsymbol{T}_\infty \boldsymbol{y}_{t,m}$ perfectly. More realistically, we will be able to synthesize some signal $\widetilde{\boldsymbol{f}}_{t,m}$, that exhibits distortion from effects such as quantization and imperfect filtering. Depending on the application, it may be necessary to compute the signal $\widetilde{\boldsymbol{f}}_{t,m}$ in the first place, or at least compute the error $\|\widetilde{\boldsymbol{f}}_{t,m} - \boldsymbol{T}_\infty \boldsymbol{y}_{t,m}\|_\infty$, in order to ensure the proper transmission of messages.
2. The generalized Shannon equivalence, which motivates the processing of signals in the digital domain, is applicable as longs as the entirety of the considered system is linear. In practical wireless communications systems, *non*-linear distortions are a common issue. In particular, the analog subsystems of both the transmitter and the receiving end have to operate within a certain dynamic range, which sets an upper limit to the $\|\cdot\|_\infty$-norm of the analog signals they can process properly. Accordingly, in order to avoid *non*-linear distortions, we need to be able to compute or at least upper-bound the values $\|\widetilde{\boldsymbol{f}}_{t,m}\|_\infty$ and

$$\left\|\boldsymbol{f}_{\mathrm{ph},t} * \widetilde{\boldsymbol{f}}_{t,m}\right\|_\infty \leq \left\|\boldsymbol{f}_{\mathrm{ph},t}\right\|_1 \left\|\widetilde{\boldsymbol{f}}_{t,m}\right\|_\infty \leq \left(\left\|\boldsymbol{f}_t\right\|_1 + \epsilon_{\mathrm{ph}}\right) \left\|\widetilde{\boldsymbol{f}}_{t,m}\right\|_\infty.$$

The details of this issue are investigated in the context of bounded-input-bounded-output (BIBO) stability analysis and the peak-to-average power ratio (PAPR) problem.

3. Analogous to the situation at the transmitter, we will not be able to measure the signal $\boldsymbol{y}_{\mathrm{re},t}$ perfectly at the receiving end. Due to finite quantization accuracy and imperfect filtering, we obtain an approximate signal $\widetilde{\boldsymbol{y}}_{\mathrm{re},t}$. Since the overall duration of sampling is finite, we can assume $\widetilde{\boldsymbol{y}}_{\mathrm{re},t}$ and $\boldsymbol{T}_\infty \widetilde{\boldsymbol{y}}_{\mathrm{re},t} =: \widetilde{f}_{\mathrm{re},t}$ are elementary computable. Choosing $\boldsymbol{y}_{\mathrm{de},t,n} \in \mathrm{dom}(\boldsymbol{T}_\infty) \cap \mathcal{C}\ell_0^\infty, n \in \{1,\dots,N\}$, and denoting $f_{\mathrm{de},t,n} := \boldsymbol{T}_\infty \boldsymbol{y}_{\mathrm{de},t,n}, n \in \{1,\dots,N\}$, we generally have $\|\widetilde{\boldsymbol{y}}_{\mathrm{re},t} - \boldsymbol{y}_{\mathrm{de},t,n}\|_\infty \neq \|\widetilde{f}_{\mathrm{re},t} - f_{\mathrm{de},t,n}\|_\infty$. Aside from computational convenience, there is no a priori reason to perform decoding based on $\widetilde{\boldsymbol{y}}_{\mathrm{re},t}$ and $\boldsymbol{y}_{\mathrm{de},t,1},\dots,\boldsymbol{y}_{\mathrm{de},t,N}$ rather than $\widetilde{f}_{\mathrm{re},t}$ and $f_{\mathrm{de},t,1},\dots,f_{\mathrm{de},t,N}$. In view of the mentioned limitations of real-world systems, this observation becomes even more relevant: since effects such as quantization are generally *non*-linear, information may actually be lost if the decoding is performed in the space $\mathcal{C}\ell_0^\infty$ rather than $\mathcal{CB}_{0,\pi}^\infty$. Unless proven otherwise for a specific case, the same argument holds true when we consider decoding schemes other than region-based ones.
4. From a model-based perspective, taking imperfect sampling at the receiving end into account raises another issue when considering the entire system within $\mathcal{C}\ell^1$. As indicated above, we may want to design the system with a specified margin-of-error, i.e., we want to guarantee that proper message transmission is possible as long as we have $\|\boldsymbol{w}\|_\infty < \epsilon_{\boldsymbol{w}}$. If we instead require $\|\boldsymbol{T}_\infty \boldsymbol{w}\|_\infty < \epsilon_{\boldsymbol{w}}$, we can provide the continuous-time-domain upper bound
$$\left\|\widetilde{f}_{\mathrm{re},t} - \left(\boldsymbol{T}_\infty \boldsymbol{w} + (f_{\mathrm{ph},t} * \widetilde{f}_{t,m})\right)\right\|_\infty \leq \epsilon_{\boldsymbol{w}} + \epsilon_{\mathrm{ph}} \left\|\widetilde{f}_{t,m}\right\|_\infty + \left\|\widetilde{f}_{\mathrm{re},t} - (f_t * \widetilde{f}_{t,m})\right\|_\infty$$
for the reconstruction error, which can then be transferred to the discrete-time domain. Requiring $\|\boldsymbol{w}\|_\infty < \epsilon_{\boldsymbol{w}}$ alone is insufficient, since $\boldsymbol{T}_\infty \boldsymbol{w}$ may be arbitrarily large in this case nevertheless. Thus, we can provide an estimate for the reconstruction error only if we consider the continuous-time domain of the system.

Consequently, we consider signals $f_{\mathrm{ph},t} \in \mathcal{B}_\pi^1$, $f_t \in \mathcal{CB}_\pi^1$, $\widetilde{f}_{t,m}, f_{\mathrm{de},t,n} \in \mathcal{CB}_{0,\pi}^\infty$, $t \in \mathbb{N}$, $m \in \{1,\dots,M\}, n \in \{1,\dots,N\}$, and a communications system of the form
$$\mathcal{E}_t(m) := f_{t,m}, \quad \mathcal{D}_t(\widetilde{f}_{\mathrm{re},t}, t) := D\left(\arg\min_{n \in \{1,\dots,N\}} \left\|\widetilde{f}_{\mathrm{re},t} - f_{\mathrm{de},t,n}\right\|_\infty\right), \tag{5}$$
$$\boldsymbol{T}_\infty \boldsymbol{y}_{\mathrm{re},t} := \boldsymbol{T}_\infty \boldsymbol{w} + \left(f_{\mathrm{ph},t} * \widetilde{f}_{t,m}\right), \quad \left\|f_{\mathrm{ph},t} - f_t\right\|_\infty < \epsilon_{\mathrm{ph}}, \tag{6}$$
in the following. Observe that we do not aim at actually implementing signal processing in the analog domain. We merely aim at finding find proper digital representatives for the continuous-time signals f_t, $\widetilde{f}_{t,1},\dots,\widetilde{f}_{t,M}$, $f_{\mathrm{de},t,1},\dots,f_{\mathrm{de},t,N}$, $\boldsymbol{x}_t, \widetilde{f}_{t,1},\dots,\widetilde{f}_{t,M}$, $f_{\mathrm{de},t,1},\dots,f_{\mathrm{de},t,N}$, since only considering their discrete-time counterparts is insufficient for the reasons we mentioned above. Nevertheless, $\mathfrak{X}_\star$ is, in principle, a valid way to represent f_t, $\widetilde{f}_{t,1},\dots,\widetilde{f}_{t,M}$, $f_{\mathrm{de},t,1},\dots,f_{\mathrm{de},t,N}$, $\boldsymbol{x}_t, \widetilde{f}_{t,1},\dots,\widetilde{f}_{t,M}$, $f_{\mathrm{de},t,1},\dots,f_{\mathrm{de},t,N}$, as follows from the generalized Shannon equivalence: for each $((s_{n,m})_{n,m\in\mathbb{N}}, \xi) \in \mathrm{dom}([\cdot]_{\mathfrak{X}}^\star)$, there exists exactly one $f \in \mathcal{CB}(\star)$ such that $[(s_{n,m})_{n,m\in\mathbb{N}}, \xi]_{\mathfrak{X}}^\star = f$ holds true, leaving us with the decision of whether to implement the signal processing for f_t, $\widetilde{f}_{t,1},\dots,\widetilde{f}_{t,M}$, $f_{\mathrm{de},t,1},\dots,f_{\mathrm{de},t,N}$ based on $\mathfrak{X}_1$ and $\mathfrak{X}_\infty$ or $\mathfrak{F}_1$ and $\mathfrak{F}_\infty$.

From Section 2, recall the inequalities $\mathfrak{X}_1 \succeq \mathfrak{X}_\infty|\mathcal{CB}_\pi^1$ and $\mathfrak{F}_1 \succeq \mathfrak{F}_\infty|\mathcal{CB}_\pi^1$. Hence, if possible, it may be beneficial to implement the signal processing for all of the signals $\widetilde{f}_{t,1},\dots,\widetilde{f}_{t,M}$, $f_{\mathrm{de},t,1},\dots,f_{\mathrm{de},t,N}$ based on $\mathfrak{X}_1$ or $\mathfrak{F}_1$ as well, since we can always recover corresponding signal descriptions in $\mathfrak{X}_\infty|\mathcal{CB}_\pi^1$, $\mathfrak{F}_\infty|\mathcal{CB}_\pi^1$, respectively, if needed. On the other hand, depending on the specific application, we may have to choose $\widetilde{f}_{t,1},\dots,\widetilde{f}_{t,M}$, $f_{\mathrm{de},t,1},\dots,f_{\mathrm{de}t,N} \in \mathcal{CB}_{0,\pi}^\infty \setminus \mathcal{CB}_\pi^1$, in which case we necessarily have to resort to using either $\mathfrak{X}_\infty$ or $\mathfrak{F}_\infty$. However, in any of the above cases, we must first and foremost be able to compute the relevant norms of the involved signals to implement the communication system. Before mathematical analysis, we summarize the relevant requirements as follows:

In a communication system of the form (5) and (6), we consider $\|f_t\|_1$, $\|\widetilde{f}_{t,m}\|_\infty$, $\|f_{\mathrm{de},t,n}\|_\infty$, $\|f_t * \widetilde{f}_{t,m}\|_\infty$, $\|\widetilde{f}_{\mathrm{re},t} - f_{\mathrm{de},t,n}\|_\infty$, and $\|\widetilde{f}_{\mathrm{re},t} - (f_t * \widetilde{f}_{t,m})\|_\infty$ relevant properties for all $t \in \mathbb{N}$, $m \in \{1,\ldots,M\}$, $n \in \{1,\ldots,N\}$. Thus, regarding any sufficient representation of signals $f \in \mathcal{CB}(\star)$ on digital hardware, we require to be able to recover $\|f\|_\star$. In other words, the mapping $\|\cdot\|_\star : \mathcal{CB}(\star) \to \mathbb{R}_\mu$, $f \mapsto \|f\|_\star$ has to be computable in the employed machine-readable language.

Theorem 2. *The mapping* $\|\cdot\|_\star : \mathfrak{F}_\star \to \mathfrak{R}$, $(f, \Lambda_{\mathfrak{F}}, [\cdot]^\star_{\mathfrak{F}}) \mapsto (\|f\|_\star, \Lambda_{\mathfrak{R}}, [\cdot]_{\mathfrak{R}})$ *is computable.*

Proof. Observe that for $f \in \mathcal{CB}(\star)$ elementary computable, the mapping $f \mapsto \|f\|_\star$ is computable. That is, for $N, M \in \mathbb{N}$ and a rational-complex $(M+1)$-tuple $\boldsymbol{z} := (z_m)_{m\in\mathcal{I}}$, $\mathcal{I} = \{0,\ldots M\}$, there exists a computable mapping $(N, \boldsymbol{z}) \mapsto G_\star(N, \boldsymbol{z}) \in \mathbb{Q}$, such that

$$\left| G_\star(N, \boldsymbol{z}) - \left\| \sum_{m=0}^{M} z_m \cdot \mathrm{sinc}_{\nu(m)} \right\|_\star \right| < \frac{1}{2^N}$$

holds true (provided the relevant norm exists). For $f = [(s'_{n,m})_{n,m\in\mathbb{N}}, \xi']^\star_{\mathfrak{F}}$ arbitrary, let the computable sequence $(f_M)_{M\in\mathbb{N}}$ of elementary computable sequences satisfy (3). Further, for $M \in \mathbb{N}$, define

$$r_M := G_\star\big(M, (s'_{n,M})_{n\in\mathcal{I}(M)}\big), \quad \mathcal{I}(M) := \{0,\ldots,\xi'(M)\}.$$

Then, $(r_M)_{M\in\mathbb{N}}$ is a computable sequence of rational numbers. Employing the triangle inequality, we obtain

$$\big|r_M - \|f\|_\star\big| \le \big|r_M - \|f_M\|_\star\big| + \big\|f - f_M\big\|_\star < \frac{1}{2^M} + \frac{1}{2^M} = \frac{1}{2^{(M-1)}}.$$

Thus, defining $\xi : \mathbb{N} \to \mathbb{N}, M \mapsto M+1$, we have $[(r_M)_{M\in\mathbb{N}}, \xi]_{\mathfrak{R}} = \|f\|_\star$. Further, it follows from the SMN-Theorem (c.f. Section 2) that the mapping $((s'_{n,m})_{n,m\in\mathbb{N}}, \xi') \mapsto ((r_M)_{M\in\mathbb{N}}, \xi)$ (with $((s'_{n,m})_{n,m\in\mathbb{N}}, \xi')$ and $((r_M)_{M\in\mathbb{N}}, \xi)$ as above) is computable, which concludes the proof. □

Theorem 3. *The mapping* $\|\cdot\|_\star : \mathfrak{X}_\star \to \mathfrak{R}$, $(f, \Lambda_{\mathfrak{X}}, [\cdot]^\star_{\mathfrak{X}}) \mapsto (\|f\|_\star, \Lambda_{\mathfrak{R}}, [\cdot]_{\mathfrak{R}})$ *is* not *computable.*

Proof. The statement follows by contradiction from Lemmas 1 and 2, respectively. To this end, assume the mapping $\|\cdot\|_\star : \mathfrak{X}_\star \to \mathfrak{R}$, $(f, \Lambda_{\mathfrak{X}}, [\cdot]^\star_{\mathfrak{X}}) \mapsto (\|f\|_\star, \Lambda_{\mathfrak{R}}, [\cdot]_{\mathfrak{R}})$ is computable and let $(x_k)_{k\in\mathbb{N}}$ be a sequence that satisfies Lemma 1, Lemma 2, respectively, for some recursively enumerable *nonrecursive* set $\Omega \subset \mathbb{N}$. Then, there exists a computable mapping

$$k \mapsto \big((s_{n,m}(k))_{n,m\in\mathbb{N}}, \xi_k\big)$$

such that for all $k \in \mathbb{N}$, the pair $((s_{n,m}(k))_{n,m\in\mathbb{N}}, \xi_k)$ determines x_k in the sense of Definition 5, and we have

$$[(s_{n,m}(k))_{n,m\in\mathbb{N}}, \xi_k]^\star_{\mathfrak{X}} = f_k.$$

If the mapping $\|\cdot\|_\star : \mathfrak{X}_\star \to \mathfrak{R}$, $(f, \Lambda_{\mathfrak{X}}, [\cdot]^\star_{\mathfrak{X}}) \mapsto (\|f\|_\star, \Lambda_{\mathfrak{R}}, [\cdot]_{\mathfrak{R}})$ is indeed computable, there must also exist a computable mapping $((s_{n,m}(k))_{n,m\in\mathbb{N}}, \xi_k) \mapsto ((r_m(k))_{m\in\mathbb{N}}, \xi'_k)$, $k \in \mathbb{N}$ such that, for all $k \in \mathbb{N}$, we have

$$[(r_m(k))_{m\in\mathbb{N}}, \xi'_k]_{\mathfrak{R}} = \|f_k\|_\star.$$

By concatenation, we conclude that $k \mapsto \|f_k\|_\star$, $k \in \mathbb{N}$ is computable as well. For all $k \in \mathbb{N}$, we define

$$r_<(k) := r_{\xi'_k(1)}(k) - \frac{1}{2}.$$

Then, $r_<(k)$ is a rational number, and the mapping $k \mapsto r_<(k)$, $k \in \mathbb{N}$, is computable. For all $k \in \mathbb{N}$, we further have $r_<(k) < \lim_{m\to\infty} r_m(k) < r_<(k) + 1$ by construction, and thus, $k \in \Omega \Leftrightarrow r_<(k) > 0$ by the requirements of Lemma 1, Lemma 2, respectively. We define

$$g : \mathbb{N} \to \mathbb{N}, k \mapsto g(k) := \begin{cases} 1, & \text{if } r_<(k) > 0, \\ 0, & \text{otherwise,} \end{cases}$$

and observe that, we have $g = \mathbb{1}_\Omega$. By the SMN-Theorem (c.f. Section 2), $(k \mapsto g(k))$ is computable, i.e., g is a μ-recursive function. Accordingly, Ω is recursive, which contradicts the prerequisite of Ω being *nonrecursive*. $\square$

Theorem 4. *We have $\mathfrak{F}_\star \succ \mathfrak{X}_\star$. In the sense of Section 2, the inequality is strict.*

Proof. Denote by $\mathrm{Id}^\star_{\mathfrak{X},\mathfrak{F}} : \mathfrak{X}_\star \to \mathfrak{F}_\star$ and $\mathrm{Id}^\star_{\mathfrak{F},\mathfrak{X}} : \mathfrak{F}_\star \to \mathfrak{X}_\star$ the relevant identity mappings in the sense of Section 2. That is, we have

$$\mathrm{Id}^\star_{\mathfrak{X},\mathfrak{F}}(f, \Lambda_\mathfrak{X}, [\cdot]^\star_\mathfrak{X}) = (f, \Lambda_\mathfrak{F}, [\cdot]^\star_\mathfrak{F}),$$
$$\mathrm{Id}^\star_{\mathfrak{F},\mathfrak{X}}(f, \Lambda_\mathfrak{F}, [\cdot]^\star_\mathfrak{F}) = (f, \Lambda_\mathfrak{X}, [\cdot]^\star_\mathfrak{X})$$

for all $f \in \mathcal{CB}(\star)$. We divide the proof in two parts: first, we prove that $\mathrm{Id}^\star_{\mathfrak{X},\mathfrak{F}}$ is *not* computable; second, we prove that $\mathrm{Id}^\star_{\mathfrak{F},\mathfrak{X}}$ is computable.

In essence, the first part is a corollary of Theorems 2 and 3, which follows by contradiction. Assume $\mathrm{Id}^\star_{\mathfrak{X},\mathfrak{F}}$ is computable. By Theorem 2, the mapping $\|\cdot\|_\star : \mathfrak{F}_\star \to \mathfrak{R}$, $(f, \Lambda_\mathfrak{F}, [\cdot]^\star_\mathfrak{F}) \mapsto (\|f\|_\star, \Lambda_\mathfrak{R}, [\cdot]_\mathfrak{R})$, is computable. Hence, by concatenation, we obtain the computable mapping

$$\left\|\mathrm{Id}^\star_{\mathfrak{X},\mathfrak{F}}(\cdot)\right\|_\star : \mathfrak{X}_\star \to \mathfrak{R},\ (f, \Lambda_\mathfrak{X}, [\cdot]^\star_\mathfrak{X}) \mapsto (\|f\|_\star, \Lambda_\mathfrak{R}, [\cdot]_\mathfrak{R}),$$

contradicting Theorem 3. Thus, $\mathrm{Id}^\star_{\mathfrak{X},\mathfrak{F}}$ cannot be computable.

The second part is a consequence of the continuity of the sampling operator $\boldsymbol{S}_\star : \mathcal{B}(\star) \to \ell(\star)$. Particularly, there exist constants $C_\star \in \{q \in \mathbb{Q} : \log_2(q) \in \mathbb{Z}\}$ such that $\|\boldsymbol{S}_\star f\|_\star < C_\star\|f\|_\star$ holds true for all $f \in \mathcal{B}(\star)$. Let $[(s'_{n,m})_{n,m\in\mathbb{N}}, \xi']^\star_\mathfrak{F} = \boldsymbol{T}_\star \boldsymbol{x}$ be arbitrary. For all $M \in \mathbb{N}$, we have

$$\frac{1}{2^M} > \left\| \boldsymbol{T}_\star \boldsymbol{x} - \sum_{n=0}^{\xi'(M)} s'_{n,M} \cdot \mathrm{sinc}_{\nu(n)} \right\|_\star \geq \frac{1}{C_\star} \left\| \boldsymbol{x} - \sum_{n=0}^{\xi'(M)} s'_{n,M} \cdot \boldsymbol{\delta}_{\nu(n)} \right\|_\star.$$

Define $K_\star := \log_2 C_\star$, $\xi : \mathbb{N} \to \mathbb{N}$, $M \mapsto \xi'(M + K_\star)$, and $(s_{n,m})_{n,m\in\mathbb{N}} := (s'_{n,m+K_\star})_{n,m\in\mathbb{N}}$, and observe that $(s_{n,m})_{n,m\in\mathbb{N}}$ is a computable double sequence of rational-complex numbers and ξ is a μ-recursive function. For all $M \in \mathbb{N}$, we have

$$\left\| \boldsymbol{x} - \sum_{n=0}^{\xi(M)} s_{n,M} \cdot \boldsymbol{\delta}_{\nu(n)} \right\|_\star = \left\| \boldsymbol{x} - \sum_{n=0}^{\xi'(M+K_\star)} s'_{n,M+K_\star} \cdot \boldsymbol{\delta}_{\nu(n)} \right\|_\star < \frac{C_\star}{2^{M+K_\star}} = \frac{C_\star}{C_\star} \frac{1}{2^M} = \frac{1}{2^M}.$$

Thus, the pair $((s_{n,m})_{n,m\in\mathbb{N}}, \xi)$ determines $\boldsymbol{x}$ in the sense of Definition 5, and we have $[(s_{n,m})_{n,m\in\mathbb{N}}, \xi]^\star_\mathfrak{X} = \boldsymbol{T}_\star \boldsymbol{x}$. Further, by the SMN-Theorem (c.f. Section 2), the mapping $((s'_{n,m})_{n,m\in\mathbb{N}}, \xi') \mapsto ((s_{n,m})_{n,m\in\mathbb{N}}, \xi)$ is computable, which concludes the proof. $\square$

For the languages $\mathfrak{F}_\star$ and $\mathfrak{X}_\star$, Theorem 4 corresponds to the Case 1 of the distinction made in Section 2: descriptions in the language $\mathfrak{F}_\star$ contain more information than descriptions in the language $\mathfrak{X}_\star$. In Section 2, we also indicated a link between the relationship of $\mathfrak{F}_\star$ and $\mathfrak{X}_\star$, i.e., the inequality $\mathfrak{F}_\star \succ \mathfrak{X}_\star$, and the computability of the operators $\boldsymbol{S}_\star$ and $\boldsymbol{T}_\star$. In turn, whether $\boldsymbol{S}_\star$ and $\boldsymbol{T}_\star$ are computable is the formal rephrasing of whether the generalized Shannon equivalence holds true on the algorithmic level. Consider the set

$$\boldsymbol{S}_\star(\mathcal{CB}(\star)) := \left\{ \boldsymbol{x} \in \mathcal{C}\ell(\star) : \boldsymbol{x} = \boldsymbol{S}_\star \boldsymbol{f} \text{ for some } \boldsymbol{f} \in \mathcal{CB}(\star) \right\} \subset \mathcal{C}\ell(\star),$$

i.e., $\boldsymbol{S}_\star(\mathcal{CB}(\star))$ consists of those sequences $\boldsymbol{x} \in \mathcal{C}\ell(\star)$ that equate to some computable signal $\boldsymbol{f} \in \mathcal{CB}(\star)$ under the action of $\boldsymbol{T}_\star$. Naturally, we can define a machine-readable language $\mathfrak{l}_\star$ for $\boldsymbol{S}_\star(\mathcal{CB}(\star))$ according to

$$[(s_{n,m})_{n,m\in\mathbb{N}}, \xi]^\star_{\mathfrak{l}} = \boldsymbol{x} \quad :\Leftrightarrow \quad [(s_{n,m})_{n,m\in\mathbb{N}}, \xi]^\star_{\mathfrak{X}} = \boldsymbol{T}_\star \boldsymbol{x},$$

in which case the identity mappings (in the sense of Section 2) $\mathrm{Id}^\star_{\mathfrak{X},\mathfrak{F}} : \mathfrak{X}_\star \to \mathfrak{F}_\star$ and $\mathrm{Id}^\star_{\mathfrak{F},\mathfrak{X}} : \mathfrak{F}_\star \to \mathfrak{X}_\star$ become the interpolation operator $\boldsymbol{T}_\star : \mathfrak{l}_\star \to \mathfrak{F}_\star$ and sampling operator $\boldsymbol{S}_\star : \mathfrak{F}_\star \to \mathfrak{l}_\star$, respectively. Then, according to Theorem 4, $\boldsymbol{S}_\star$ is computable, while $\boldsymbol{T}_\star$ is *not*. As indicated in Section 2, this is due to the discontinuity of $\boldsymbol{T}_\star$ with regard to the relevant norm. In other words, the generalized Shannon equivalence between $\mathcal{B}(\star)$ and $\ell(\star)$ does *not* hold true on an algorithmic level! Analytically, if

$$[(s_{n,m})_{n,m\in\mathbb{N}}, \xi]^\star_{\mathfrak{X}} = [(s'_{n,m})_{n,m\in\mathbb{N}}, \xi']^\star_{\mathfrak{F}} = \boldsymbol{f}$$

holds true, both $((s_{n,m})_{n,m\in\mathbb{N}}, \xi)$ and $((s'_{n,m})_{n,m\in\mathbb{N}}, \xi')$ uniquely determine all mathematically well-defined properties of $\boldsymbol{f}$, including $\|\boldsymbol{f}\|_\star$. Algorithmically, as Theorems 2–4 show, this is *not* the case. With respect to the requirements summarized above, we conclude our analysis as follows:

The generalized Shannon equivalence between $\mathcal{B}(\star)$ and $\ell(\star)$ does *not* hold true on an algorithmic level. In particular, we observe the following:

- Regarding $\|\boldsymbol{f}\|_\star$ as relevant property, discrete-time descriptions of signals $\boldsymbol{f} \in \mathcal{CB}(\star)$ are an insufficient representative of analog information on digital hardware;
- Regarding $\|\boldsymbol{f}\|_\star$ as relevant property, continuous-time descriptions of signals $\boldsymbol{f} \in \mathcal{CB}(\star)$ are a sufficient representative of analog information on digital hardware;
- Any computation on the basis of discrete-time descriptions can be processed on the basis of continuous-time descriptions as well, i.e., continuous-time descriptions capture more information (in the sense of the distinction made in Section 2) than discrete-time descriptions.

4. Discussion

In Section 2, we have established a formal framework of machine-readable languages for analog bandlimited signals $\boldsymbol{f} \in \mathcal{CB}(\star)$ (recall that we used the symbol '$\star$' as a placeholder that may be uniformly replaced by '1' or '∞'). In particular, we have introduced the languages $\mathfrak{X}_\star$ and $\mathfrak{F}_\star$, which formalize discrete-time descriptions and continuous-time descriptions for the elements of $\mathcal{CB}(\star)$, respectively. In Section 3, we have applied the established framework to a standard engineering model for wireless communication networks. Particularly, the model is relevant in the context of digital twinning, an exemplary application of which we have discussed in Section 1. Then, we have shown that discrete-time descriptions are, despite being the quasi-standard in digital signal processing, an insufficient representative of the elements of $\mathcal{CB}(\star)$: In contrast to continuous-time descriptions, they do *not* allow for computing the norm $\|\cdot\|_\star$, which we substantiated to be a relevant property within the considered model for wireless communication networks. Finally, we have shown that any computation on the basis of discrete-time descriptions can be processed on the basis of continuous-time descriptions equally, i.e., continuous-time descriptions capture "more" information than discrete-time descriptions.

Section 3 focuses on the $\|\cdot\|_\star$-norm as relevant property. Using the methods and techniques we established in Sections 2 and 3, we can extend the results to several other properties that can be considered relevant in particular applications. We will discuss several of them in the following.

- Time concentration. For $L \in \mathbb{R}_\mu$, denote by $\mathrm{rct}(\cdot, L) : \mathbb{R} \to \mathbb{R}$ the function that equals one for $-L \leq t \leq L$, zero otherwise. For a signal $\boldsymbol{f} \in \mathcal{CB}(\star)$, the function $\boldsymbol{f}_L : \mathbb{R} \to \mathbb{R}, t \mapsto \mathrm{rct}(t, L)\boldsymbol{f}(t)$ is an element of $\mathcal{L}^\star(\mathbb{R})$ (c.f. [12], Definition 2.1, p. 15). We refer to the mapping
$$\Gamma : \mathcal{CB}(\star) \times \mathbb{R}_\mu \to \mathbb{R}_\mu, (\boldsymbol{f}, L) \mapsto \|\boldsymbol{f}_L\|_\star$$
as time concentration. It is computable as a mapping $\Gamma : \mathfrak{F}_\star \times \mathfrak{R} \to \mathfrak{R}$, but *not* as a mapping $\Gamma : \mathfrak{X}_\star \times \mathfrak{R} \to \mathfrak{R}$. For details, we refer to [20];
- Time evaluation. We refer to the mapping $\Phi : \mathcal{CB}(\star) \times \mathbb{R}_\mu \to \mathbb{C}_\mu, (\boldsymbol{f}, t) \mapsto \boldsymbol{f}(t)$ as time evaluation. It is computable as a mapping $\Phi : \mathfrak{F}_1 \times \mathfrak{R} \to \mathfrak{C}$, $\Phi : \mathfrak{X}_1 \times \mathfrak{R} \to \mathfrak{C}$, and $\Phi : \mathfrak{F}_\infty \times \mathfrak{R} \to \mathfrak{C}$, but *not* as a mapping $\Phi : \mathfrak{X}_\infty \times \mathfrak{R} \to \mathfrak{C}$;
- Time derivative. Denote the time derivative of $\boldsymbol{f} \in \mathcal{CB}(\star)$ by ${}^{\mathrm{d}}/_{\mathrm{d}t}\boldsymbol{f}$ (among other things, computing $\boldsymbol{f} \mapsto {}^{\mathrm{d}}/_{\mathrm{d}t}\boldsymbol{f}$ is essential for estimating system dynamics by means of the mean-value theorem). We can directly compute the time derivative of elementary computable functions in $\mathcal{CB}(\star)$. Further, we have $\|{}^{\mathrm{d}}/_{\mathrm{d}t}\boldsymbol{f}\|_\star \leq \pi\|\boldsymbol{f}\|_\star$ for all $\boldsymbol{f} \in \mathcal{CB}(\star)$. Hence, ${}^{\mathrm{d}}/_{\mathrm{d}t}$ is computable as a mapping ${}^{\mathrm{d}}/_{\mathrm{d}t} : \mathfrak{F}_\star \to \mathfrak{F}_\star$. In contrast, ${}^{\mathrm{d}}/_{\mathrm{d}t}$ is *not* computable as a mapping ${}^{\mathrm{d}}/_{\mathrm{d}t} : \mathfrak{X}_\star \to \mathfrak{X}_\star$.

As indicated in Section 1, the upcoming 6G communication standard specifies large parts of the technological infrastructure required for applications of digital twinning. These applications span a broad range within what is called the critical infrastructure, including medicine and healthcare, transportation, critical industrial facilities, and energy supply. In this context, it is easy to imagine communication systems that must be designed fail-safe. From Section 3, recall the reasons for considering a communication system's analog part. Point 4 list a possible requirement for such a fail-safe system: provided the noise sequence $\|\boldsymbol{w}\|_\infty$ does not exceed a "maximum credible amplitude" $\epsilon_{\boldsymbol{w}}$, correct message transmission must be possible. Outside of communications engineering, such requirements are standard in control theory and correspond to a distinction of design-basis events from beyond-design-basis events. Traditionally, we find this distinction in engineering for safety-critical facilities, such as nuclear power plants. Based on mathematical modeling, the plant is designed to withstand all accidents whose occurrence is considered realistic. Consequently, a model-to-system relationship guides the design process.

The replacement of the design process's model-to-system relationship by a metamodel-to system relationship is a distinguishing feature of digital twinning: any machine-readable language necessarily incorporates a formal model of the physical entity it describes. In our case, this formal model consists of the sampling and interpolation theory, as introduced in Section 2. Once the relevant algorithms are implemented on the employed computing hardware, the model itself becomes part of the physical system and its dynamics. Accordingly, a conclusive generalization of the distinction between design-basis and beyond-design-basis events for such systems considers the resulting interaction between computing hardware, implemented algorithms, and physical agents.

The analysis of the structural properties of $\mathfrak{X}_\star$ and $\mathfrak{F}_\star$ in Section 3 follows this approach. In order to ensure that the system as a whole can withstand all design-basis events, such as any possible noise sequence not exceeding "maximum credible amplitude" $\epsilon_{\boldsymbol{w}}$, we need to prove that the employed algorithm can capture those characteristics of the physical entity that are relevant for design-basis events correctly. In other words, we require the employed machine-readable language to provide sufficient representatives of the analog world. This facet of trustworthiness is known as integrity [4,9]. To the best of the authors' knowledge, this work is the first to consider the problem of identifying trustworthy digital representations of analog systems. However, the problem is fundamental to digital twinning, and the authors believe that it will receive increased attention in the anticipated generation of communication technology.

Author Contributions: Conceptualization, H.B. and F.H.P.F.; methodology, H.B. and Y.N.B.; validation, H.B., Y.N.B. and U.J.M.; formal analysis, H.B. and Y.N.B.; investigation, H.B. and Y.N.B.; writing—original draft preparation, H.B., Y.N.B. and U.J.M.; writing—review and editing, H.B., Y.N.B., U.J.M. and F.H.P.F.; visualization, Y.N.B.; supervision, H.B. and F.H.P.F.; project administration, H.B. and F.H.P.F.; funding acquisition, H.B. and F.H.P.F. All authors have read and agreed to the published version of the manuscript.

Funding: The work of Holger Boche was supported in part by the German Federal Ministry of Education and Research (BMBF) within the National Initiative on 6G Communication Systems through the Research Hub 6G-life under Grant 16KISK002; Excellence Cluster Cyber Security in the Age of Large-Scale Adversaries (CASA), Ruhr University Bochum, 44801 Bochum. The work of Yannik N. Böck was supported in part by the BMBF within the national initiative on Post Shannon Communication (NewCom) under Grant 16KIS1004 and in part by the BMBF within 6G-life under Grant 16KISK002. The work of Ullrich J. Mönich was supported by the BMBF within NewCom under Grant 16KIS1003K. The work of Frank H. P. Fitzek was supported in part by the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as part of Germany's Excellence Strategy—EXC 2050/1—Project ID 390696704—Cluster of Excellence "Centre for Tactile Internet with Human-in-the-Loop" (CeTI) of Technische Universität Dresden and in part by the BMBF within the program "Souverän. Digital. Vernetzt." (joint project 6G-life, project identification number: 16KISK001K).

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: The authors would like to thank Vlad-Costin Andrei and Xinyang Li of the TUM ACES Lab for providing images and technological details of the exemplary digital-twinning application discussed within the article.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the collection, analysis, and interpretation of the established results, nor in the writing of the manuscript or the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

SSI	Shannon Sampling and Interpolation
BIBO	Bounded-Input, Bounded-Output
PAPR	Peak-to-Average Power Ratio

Appendix A. Proofs of Lemmas 1 and 2

This appendix provides detailed proofs of Lemmas 1 and 2 from Section 2, as well as a brief discussion of the discontinuity of the interpolation operator $\boldsymbol{T}_\star$. To this end, we establish two additional supplementary results, Lemmas A1 and A2.

Lemma A1. *There exists a computable sequence $(f_M)_{M\in\mathbb{N}}$ of elementary computable functions in $\mathcal{CB}^\infty_{0,\pi}$ such that $f_M(1/2) \geq 1$, $\|\boldsymbol{S}_\infty f_M\|_\infty < 1/M$, and $\|f_M\|_\infty \leq C$ are satisfied for all $M \in \mathbb{N}$ and some number $C > 0$.*

Observe that Lemma A1 implies the unboundedness (and hence discontinuity) of the interpolation operator $\boldsymbol{T}_\infty$. For $(f_M)_{M\in\mathbb{N}}$ as in Lemma A1,

$$\lim_{M\to\infty} \frac{\|\boldsymbol{T}_\infty \boldsymbol{S}_\infty f_M\|_\infty}{\|\boldsymbol{S}_\infty f_M\|_\infty} = \lim_{M\to\infty} \frac{\|f_M\|_\infty}{\|\boldsymbol{S}_\infty f_M\|_\infty} > \lim_{M\to\infty} \frac{1}{1/M} = \infty \tag{A1}$$

holds true. Further, it provides an example of a sequence $\boldsymbol{x} \in \ell_0^\infty \setminus \mathrm{dom}(\boldsymbol{T}_\infty)$: for all $N \in \mathbb{N}$, we have

$$\left\| \sum_{M=1}^{N} \frac{\boldsymbol{S}_\infty \boldsymbol{f}_M}{M} \right\|_\infty \leq \sum_{M=1}^{N} \frac{1}{M^2} = \frac{\pi^2}{6}, \quad \left\| \sum_{M=1}^{N} \frac{\boldsymbol{f}_M}{M} \right\|_\infty \geq \left| \sum_{M=1}^{N} \frac{\boldsymbol{f}_M(1/2)}{M} \right| \geq \ln(N), \tag{A2}$$

i.e., there exists $\boldsymbol{x} \in \ell_0^\infty$ with $\boldsymbol{x} = \boldsymbol{S}_\infty \boldsymbol{f}_1 + 1/2 \cdot \boldsymbol{S}_\infty \boldsymbol{f}_2 + 1/3 \cdot \boldsymbol{S}_\infty \boldsymbol{f}_3 + \ldots$, while the sequence $\boldsymbol{f}_1,\ \boldsymbol{f}_1 + 1/2 \cdot \boldsymbol{f}_2,\ \boldsymbol{f}_1 + 1/2 \cdot \boldsymbol{f}_2 + 1/3 \cdot \boldsymbol{f}_3, \ldots$ diverges with respect to $\|\cdot\|_\infty$.

Proof of Lemma A1. To begin with, consider a computable sequence of rational numbers $(r_m)_{m\in\mathbb{N}}$ that satisfies $r_m - 2^{-m} < \pi < r_m$ for all $m \in \mathbb{N}$. Further, define

$$s_{n,M} := \sum_{k=1}^{\infty} \frac{(-1)^k \delta_k[\nu(n)]}{\Delta(2^{8M})},\ \Delta(m) := \sum_{k=1}^{m} \frac{-1}{r_m(k - \frac{1}{2})},\ \xi(M) := 2^{8M+1}$$

for all $n, m, M \in \mathbb{N}$. Then, the pair $((s_{n,M})_{n,M\in\mathbb{N}}, \xi)$ determines a computable sequence $(\boldsymbol{f}_M)_{M\in\mathbb{N}}$ of elementary computable functions in the sense of Section 2. For $K = 2^{8M}$, $M \in \mathbb{N}$, we have

$$\boldsymbol{f}_M := \sum_{n=0}^{\xi(M)} s_{n,M} \cdot \mathrm{sinc}_{\nu(n)} = \frac{1}{\Delta(K)} \sum_{k=1}^{K} (-1)^k \,\mathrm{sinc}_k\,.$$

In the following, we will prove that $(\boldsymbol{f}_M)_{M\in\mathbb{N}}$ satisfies the requirements of the lemma. First, observe that

$$\mathrm{sinc}_M\left(\frac{1}{2}\right) = \frac{(-1)^M}{\pi(\frac{1}{2} - M)}$$

holds true for all $M \in \mathbb{N}$. Consequently, for all $K = 2^{8M}$, $M \in \mathbb{N}$, we have

$$\boldsymbol{f}_M(1/2) = \frac{\sum_{k=1}^{K} (-1)^k \frac{(-1)^k}{\pi(\frac{1}{2}-k)}}{-\frac{1}{r_M} \sum_{k=1}^{K} \frac{1}{k-\frac{1}{2}}} \geq -\frac{\sum_{k=1}^{K} \frac{1}{(k-\frac{1}{2})}}{\sum_{k=1}^{K} \frac{1}{\frac{1}{2}-k}} = 1.$$

Next, recall that we have $r_N - 2^{-N} < \pi < r_N$ and thus $r_N < 4$ for all $N \in \mathbb{N}$. Hence, the inequality

$$|\Delta(N)| = \frac{1}{r_N} \sum_{k=1}^{N} \frac{1}{k - \frac{1}{2}} > \frac{1}{4} \int_1^{N+1} \frac{1}{\tau - \frac{1}{2}}\,\mathrm{d}\tau = \frac{1}{4}\left(\ln\left(N + \frac{1}{2}\right) - \ln\left(\frac{1}{2}\right)\right) > \frac{\log_2(N)}{4}$$

is satisfied for all $N \in \mathbb{N}$. Furthermore, for all $K = 2^{8M}$, $M \in \mathbb{N}$, we have

$$\left\|\boldsymbol{S}_\infty \boldsymbol{f}_M\right\|_\infty = \sup_{m\in\mathbb{N}} \left| \frac{1}{\Delta(K)} \sum_{k=1}^{K} (-1)^k \delta_k[m] \right| \leq \frac{1}{|\Delta(K)|} \sup_{m\in\mathbb{N}} \sum_{k=1}^{K} \left|(-1)^k \delta_k[m]\right| = \frac{1}{|\Delta(K)|} < \frac{1}{M}.$$

It remains to be shown that the sequence $(\boldsymbol{f}_M)_{M\in\mathbb{N}}$ is uniformly bounded in the norm of its components, i.e., we have $\|\boldsymbol{f}_M\|_\infty < C$ for all $M \in \mathbb{N}$ and some number $C > 0$. For $t \in \mathbb{R}$ and $N \in \mathbb{N}$, we define

$$\begin{aligned}
k_1(t, N) &:= \max\left(\{0\} \cup \{n \in \mathbb{N} : n \leq N \text{ and } n + 1 < t\}\right), \\
k_2(t, N) &:= \min\left(\{n \in \mathbb{N} : n \leq N \text{ and } n - 1 > t\} \cup \{N + 1\}\right), \\
k_3(t, N) &:= \sum_{k=1}^{k_1(t,N)} \frac{1}{k_1(t, N) + 1 - k} + \sum_{k=k_2(t,N)}^{N} \frac{1}{k - k_2(t, N) + 1}.
\end{aligned}$$

For $t \in \mathbb{Z}$ and $M \in \mathbb{N}$, we have $|f_M(t)| \leq \|S_\infty f_M\|_\infty \leq 1$. Furthermore, for all $t \in \mathbb{R} \setminus \mathbb{Z}$, and $N \in \mathbb{N}$, observe that

$$\left|\sum_{k=1}^{N}(-1)^k \operatorname{sinc}_k(t)\right| \leq \sum_{k=1}^{N} \underbrace{|\operatorname{sinc}_k(t)|}_{\leq 1/(\pi t - \pi k)} < 2 + \frac{1}{\pi}k_3(t,N),$$

holds true, as well as

$$\frac{1}{\pi}k_3(t,N) = 2 + \frac{1}{\pi}\left(\sum_{k=1}^{k_1(t,N)}\frac{1}{k} + \sum_{k=1}^{N-k_2(t,N)+1}\frac{1}{k}\right) \leq 2 + \frac{2}{\pi}\sum_{k=1}^{N}\frac{1}{k} \overset{\text{(a)}}{<} 2 + \frac{2}{\pi} + \frac{2}{\pi}\log_2(N),$$

where, (a) follows from the inequality $\sum_{k=1}^{N}\frac{1}{k} < 1 + \ln(N) < 1 + \log_2(N)$. Accordingly, for all $t \in \mathbb{R} \setminus \mathbb{Z}$, and $K = 2^{8M}$, $M \in \mathbb{N}$, we have

$$|f_M(t)| = \left|\frac{1}{\Delta(K)}\sum_{k=1}^{K}(-1)^k \operatorname{sinc}_k(t)\right| \leq \frac{8 + \frac{8}{\pi} + \frac{8}{\pi}\log_2(K)}{\log_2(K)} \leq \frac{1}{M} + \frac{1}{\pi M} + \frac{8}{\pi} < 4,$$

Thus, setting $C := 4$ concludes the proof. □

Proof of Lemma 1. Let $\Omega \subset \mathbb{N}$ be a recursively enumerable set with runtime function $g_\Omega : \mathbb{N}^2 \to \{0,1\}$. Consider the function $h : \mathbb{N}^2 \to \mathbb{N}$ defined according to

$$h(m,k) := \sum_{l=0}^{L(k)}(1 - g_\Omega(m,l)), \quad L(k) := 2^{k+2}, \tag{A3}$$

for $m,k \in \mathbb{N}$. Further, let $(f_M)_{M\in\mathbb{N}}$ be a computable sequence of elementary computable functions as specified by Lemma A1 and define $f_{m,k} := f_{h(m,k)}$ for all $m,k \in \mathbb{N}$. Then, $(f_{m,k})_{m,k\in\mathbb{N}}$ is a computable double sequence of elementary computable functions. Moreover, the sequence $(x_{m,k})_{m,k\in\mathbb{N}} := (S_\infty f_{m,k})_{m,k\in\mathbb{N}}$ is a computable double sequence of elementary computable sequences.

For $m \in \Omega$, there exists $k \in \mathbb{N}$ such that for all $l \in \mathbb{N}$ that satisfy $l \geq k$, we have $f_{m,l} = f_{m,k}$, i.e., the limit value $\lim_{l\to\infty} f_{m,l}$ exists and is an elementary computable function. Furthermore, there exists an $M \in \mathbb{N}$ such that $f_M = \lim_{l\to\infty} f_{m,l}$ is satisfied. We define

$$f'_m := \begin{cases} \lim_{l\to\infty} f_{m,l} & \text{if } m \in \Omega, \\ 0 & \text{otherwise,} \end{cases}$$

for all $m \in \mathbb{N}$. Hence, $(f'_m)_{m\in\mathbb{N}}$ is a (*not* necessarily computable) sequence of elementary computable functions. Furthermore, the sequence $(x'_m)_{m\in\mathbb{N}} := (S_\infty f'_m)_{m\in\mathbb{N}}$ is a sequence of elementary computable sequences. In the following, we will prove by case distinction that for all $m \in \mathbb{N}$, the sequence $(x_{m,k})_{k\in\mathbb{N}}$ converges effectively towards x'_m in ℓ_0^∞, with respect to the μ-recursive modulus of convergence $\xi : \mathbb{N}^2 \to \mathbb{N}$, $(m,K) \mapsto K$.

First, assume that $m \in \Omega$ is satisfied. Then, there exists $k \in \mathbb{N}$ such that for all $K \in \mathbb{N}$ that satisfy $K \geq k$, we have $x_{m,K} = x_{m,k}$. Consider the smallest such $k \in \mathbb{N}$ and observe the following for all $K \in \mathbb{N}$:

- If $K \in \mathbb{N}$ satisfies $K \geq k$, we have $\|x'_m - x_{m,K}\|_\infty = \|x_{m,k} - x_{m,k}\|_\infty = 0 < 2^{-K}$.
- If $K \in \mathbb{N}$ satisfies $K < k$, we have, by the properties of the runtime function g_Ω and the construction of h as above, $h(m,k) \geq h(m,K) = L(K) + 1$ with $L(K)$ as in (A3). Accordingly, by $x'_m = x_{m,k}$ and application of the triangle inequality, we also have

$$\|x'_m - x_{m,K}\|_\infty \leq \|S_\infty f_{h(m,k)}\|_\infty + \|S_\infty f_{h(m,K)}\|_\infty \dots$$
$$\dots \leq \frac{1}{h(m,k)} + \frac{1}{L(K)+1} \leq \frac{1}{2^{K+2}} + \frac{1}{2^{K+2}} = \frac{2}{2^{K+2}} = \frac{1}{2^{K+1}} < 2^{-K}.$$

Second, assume that $m \in \mathbb{N} \setminus \Omega$ is satisfied. Then, we have $\boldsymbol{x}'_m \equiv 0$. Observe the following for all $K \in \mathbb{N}$:

- With $L(K)$ as in (A3), we have $\|\boldsymbol{x}'_m - \boldsymbol{x}_{m,K}\|_\infty = \|\boldsymbol{x}_{m,K}\|_\infty \le (L(K)+1)^{-1} < 2^{-K}$.

Following the preceding case distinction, we conclude that $\|\boldsymbol{x}'_m - \boldsymbol{x}_{m,K}\|_\infty < 2^{-K}$ is satisfied for all $m, K \in \mathbb{N}$. In other words, for all $m \in \mathbb{N}$, $(\boldsymbol{x}_{m,k})_{k\in\mathbb{N}}$ converges effectively towards $\boldsymbol{x}'_m$ in ℓ_0^∞, with respect to the μ-recursive modulus of convergence $\xi : \mathbb{N}^2 \to \mathbb{N}, (m, K) \mapsto K$. Consequently, $(\boldsymbol{x}'_m)_{m\in\mathbb{N}}$ is a computable sequence of sequences in $\mathcal{C}\ell_0^\infty$.

It remains to show that for all $m \in \Omega$, we have $\|f'_m\|_\infty \ge 1$, while for all $m \in \mathbb{N} \setminus \Omega$, we have $\|f'_m\|_\infty = 0$, which we again prove by case distinction.

First, recall that if $m \in \Omega$ is satisfied, f'_m is an elementary computable function such that for some $M \in \mathbb{N}$, we have $f_M = f'_m$. By assumption, $C \ge \|f_M\|_\infty \ge f_M(1/2) \ge 1$ is satisfied for some number $C > 0$ and all $M \in \mathbb{N}$. Hence, for all $m \in \Omega$ there exists an $M \in \mathbb{N}$ such that we have $C \ge \|f'_m\|_\infty = \|f_M\|_\infty \ge 1$.

Second, recall that if $m \in \mathbb{N} \setminus \Omega$ is satisfied, then f'_m is the trivial elementary computable function. That is, we have $f'_m \equiv 0$, in which case $\|f'_m\|_\infty = 0$ holds true. Hence, for all $m \in \mathbb{N} \setminus \Omega$, we have $\|f'_m\|_\infty = 0$.

Following the preceding case distinction, we conclude that for all $m \in \Omega$, we have $\|f'_m\|_\infty \ge 1$, while for all $m \in \mathbb{N} \setminus \Omega$, we have $\|f'_m\|_\infty = 0$. □

Lemma A2. *There exists a computable sequence $(f_M)_{M\in\mathbb{N}}$ of elementary computable functions in $\mathcal{CB}_\pi^1$ such that $\|f_M\|_1 \ge 1$, $\|\boldsymbol{S}_1 f_M\|_1 < 1/M$, and $\|f_M\|_1 \le C$ are satisfied for all $M \in \mathbb{N}$ and some number $C > 0$.*

Recall that Lemma A1 implies the discontinuity of $\boldsymbol{T}_\infty$, as follows from (A1). In the same manner, Lemma A2 implies the discontinuity of $\boldsymbol{T}_1$: For $(f_n)_{n\in\mathbb{N}}$ as in Lemma A2, we have

$$\lim_{n\to\infty} \frac{\|\boldsymbol{T}_1 \boldsymbol{S}_1 f_n\|_1}{\|\boldsymbol{S}_1 f_n\|_1} = \lim_{n\to\infty} \frac{\|f_n\|_1}{\|\boldsymbol{S}_1 f_n\|_1} > \lim_{n\to\infty} \frac{1}{1/n} = \infty \tag{A4}$$

Proof of Lemma A2. For all $M \in \mathbb{N}$ and with $k(m) := -2m$ for $m \in \mathbb{N}$, define the function f'_M according to

$$f'_M := \operatorname{sinc} - \frac{1}{M} \sum_{m=1}^{M} \operatorname{sinc}_{k(m)} .$$

First, observe that $\|f'_M\|_1 < 4 + (5/\pi)\ln(2M+1)$ is satisfied for all $M \in \mathbb{N}$. For a detailed proof, we refer directly to Equation (18) in [19], where the relevant inequality is explicitly derived. Since in addition, f'_M is a finite linear combination of sinc-functions, we consequently have $f'_M \in \mathcal{CB}_\pi^1$ for all $M \in \mathbb{N}$.

Second, observe that for all $M \in \mathbb{N}$, we have $(1/(6\pi))\ln(M/2) - (1/\pi) < \|f'_M\|_1$, which we can deduce from Equation (11) in [19] with a few additional steps. We define $\eta : \mathbb{R} \to \mathbb{R}$, $t \mapsto \eta(t)$,

$$\eta(t) := \begin{cases} 0 & \text{if } t < 0, \\ t\sin(\pi t) & \text{if } 0 \le t < 1, \\ \sin(\pi t) & \text{otherwise.} \end{cases}$$

Then, using Equation (11) in [19], we obtain the chain of inequalities

$$\frac{1}{6\pi}\ln\left(\frac{M}{2}\right) - \frac{1}{\pi} < \int_0^\infty f'_M(t)\eta(t)\,\mathrm{d}t \overset{(a)}{<} \int_{-\infty}^\infty |f'_M(t)||\eta(t)|\,\mathrm{d}t \overset{(b)}{\le} \int_{-\infty}^\infty |f'_M(t)|\,\mathrm{d}t = \|f'_M\|_1$$

for the function f'_M and all $M \in \mathbb{N}$, where (a) follows from the fact that $a(t) = 0$ for all $t < 0$ and (b) follows from the fact that $|a(t)| \le 1$ for all $t \in \mathbb{R}$.

Let $(r_m)_{m\in\mathbb{N}}$ be any computable sequence of rational numbers such that for all $M \in \mathbb{N}$, we have

$$\frac{1}{6\pi}\ln\left(\frac{M}{2}\right) - \frac{1}{\pi} < r_M \le \|f'_M\|_1.$$

Observe that by construction of $(f'_M)_{M\in\mathbb{N}}$ and since π is a computable number and $\ln(M/2)$ is a computable sequence of computable numbers, such a sequence exists. Using $k(m)$ as above and $K := 2^{96M+13}$, we define

$$s_{n,M} := \frac{1}{r_K}\left(\delta_0[\nu(n)] - \sum_{m=1}^{\infty}\frac{\delta_{k(m)}[\nu(n)]}{K}\right), \quad \xi(M) := 2^{96M+13+2}$$

for all $n, M \in \mathbb{N}$. Then, the pair $((s_{n,M})_{n,M\in\mathbb{N}}, \xi)$ determines a computable sequence $(f_M)_{M\in\mathbb{N}}$ of elementary computable functions in the sense of Section 2, and we have

$$f_M := \sum_{n=0}^{\xi(M)} s_{n,M}\cdot \mathrm{sinc}_{\nu(n)} = \frac{f'_K}{r_K} \tag{A5}$$

for all $M \in \mathbb{N}$. In the following, we will prove that $(f_M)_{M\in\mathbb{N}}$ satisfies the requirements of the lemma.

First, by construction of the sequence $(r_K)_{K\in\mathbb{N}}$, we have $0 < r_K \le f'_K$ for all $K \in \mathbb{N}$. Thus, following from (A5), we have $f_M \ge 1$ for all $M \in \mathbb{N}$.

Next, observe that for all $M \in \mathbb{N}$ and with $K = 2^{96M+13}$, we obtain

$$\|S_1 f_M\|_1 = \frac{\|S_1 f'_K\|_1}{r_K} < \frac{1 + 1/K\sum_{k=1}^{K} 1}{1/(6\pi)\ln(1/2\cdot 2^{96M+13}) - 1/\pi}\cdots$$
$$\cdots < \frac{2}{1/(6\pi)1/2\log_2(2^{96n+12}) - 1/\pi} = \frac{2\pi}{1/12(96M+12) - 1} = \frac{2\pi}{8M} < \frac{1}{M}$$

for the sequence $(f_M)_{M\in\mathbb{N}}$, proving the second requirement of the lemma.

It remains to be shown that the sequence $(f_M)_{M\in\mathbb{N}}$ is uniformly bounded in the norm of its components, i.e., we have $\|f_M\|_1 < C$ for all $M \in \mathbb{N}$ and some number $C > 0$. We have

$$\limsup_{K\to\infty}\frac{\|f'_K\|_1}{r_K} < \limsup_{K\to\infty}\frac{4 + (5/\pi)\ln(2M+1)}{(1/(6\pi))\ln(M/2) - (1/\pi)} = \limsup_{K\to\infty}\frac{(5/\pi)\ln(M)}{(1/(6\pi))\ln(M)} = 30.$$

Using (A5) and $K = 2^{96M+13}$ as above, it follows that $\|f_M\|_1 < C$ holds true for some number $C > 0$ and all $M \in \mathbb{N}$, which concludes the proof. □

Proof of Lemma 2. In large parts, the proof of the statement is analogous to the proof of Lemma 1. For the sake of completeness, we repeat the relevant steps nevertheless.

Let $\Omega \subset \mathbb{N}$ be a recursively enumerable set with runtime function $g_\Omega : \mathbb{N}^2 \to \{0,1\}$. Consider the function $h : \mathbb{N}^2 \to \mathbb{N}$ defined according to

$$h(m,k) := \sum_{l=0}^{L(k)}(1 - g_\Omega(m,l)), \quad L(k) := 2^{k+2}, \tag{A6}$$

for $m, k \in \mathbb{N}$. Further, let $(f_M)_{M\in\mathbb{N}}$ be a computable sequence of elementary computable functions as specified by Lemma A2 and define $f_{m,k} := f_{h(m,k)}$ for all $m, k \in \mathbb{N}$. Then, $(f_{m,k})_{m,k\in\mathbb{N}}$ is a computable double sequence of elementary computable functions. Moreover, the sequence $(x_{m,k})_{m,k\in\mathbb{N}} := (S_1 f_{m,k})_{m,k\in\mathbb{N}}$ is a computable double sequence of elementary computable sequences.

For $m \in \Omega$, there exists $k \in \mathbb{N}$ such that for all $l \in \mathbb{N}$ that satisfy $l \geq k$, we have $f_{m,l} = f_{m,k}$, i.e., the limit value $\lim_{l\to\infty} f_{m,l}$ exists and is an elementary computable function. Furthermore, there exists an $M \in \mathbb{N}$ such that $f_M = \lim_{l\to\infty} f_{m,l}$ is satisfied. We define

$$f'_m := \begin{cases} \lim_{l\to\infty} f_{m,l} & \text{if } m \in \Omega, \\ 0 & \text{otherwise,} \end{cases}$$

for all $m \in \mathbb{N}$. Hence, $(f'_m)_{m\in\mathbb{N}}$ is a (*not* necessarily computable) sequence of elementary computable functions. Furthermore, the sequence $(\boldsymbol{x}'_m)_{m\in\mathbb{N}} := (S_1 f'_m)_{m\in\mathbb{N}}$ is a sequence of elementary computable sequences. In the following, we will prove by case distinction that for all $m \in \mathbb{N}$, the sequence $(\boldsymbol{x}_{m,k})_{k\in\mathbb{N}}$ converges effectively towards $\boldsymbol{x}'_m$ in ℓ^1_0, with respect to the μ-recursive modulus of convergence $\xi : \mathbb{N}^2 \to \mathbb{N}$, $(m, K) \mapsto K$.

First, assume that $m \in \Omega$ is satisfied. Then, there exists $k \in \mathbb{N}$ such that for all $K \in \mathbb{N}$ that satisfy $K \geq k$, we have $\boldsymbol{x}_{m,K} = \boldsymbol{x}_{m,k}$. Consider the smallest such $k \in \mathbb{N}$ and observe the following for all $K \in \mathbb{N}$:

- If $K \in \mathbb{N}$ satisfies $K \geq k$, we have $\|\boldsymbol{x}'_m - \boldsymbol{x}_{m,K}\|_1 = \|\boldsymbol{x}_{m,k} - \boldsymbol{x}_{m,k}\|_1 = 0 < 2^{-K}$.
- If $K \in \mathbb{N}$ satisfies $K < k$, we have, by the properties of the runtime function g_Ω and the construction of h as above, $h(m,k) \geq h(m,K) = L(K) + 1$ with $L(K)$ as in (A6). Accordingly, by $\boldsymbol{x}'_m = \boldsymbol{x}_{m,k}$ and application of the triangle inequality, we also have

$$\begin{aligned} \left\|\boldsymbol{x}'_m - \boldsymbol{x}_{m,K}\right\|_1 &\leq \left\|S_1 f_{h(m,k)}\right\|_1 + \left\|S_1 f_{h(m,K)}\right\|_1 \dots \\ \dots &\leq \frac{1}{h(m,k)} + \frac{1}{L(K)+1} \leq \frac{1}{2^{K+2}} + \frac{1}{2^{K+2}} = \frac{2}{2^{K+2}} = \frac{1}{2^{K+1}} < 2^{-K}. \end{aligned}$$

Second, assume that $m \in \mathbb{N} \setminus \Omega$ is satisfied. Then, we have $\boldsymbol{x}'_m \equiv 0$. Observe the following for all $K \in \mathbb{N}$:

- With $L(K)$ as in (A6), we have $\|\boldsymbol{x}'_m - \boldsymbol{x}_{m,K}\|_1 = \|\boldsymbol{x}_{m,K}\|_1 \leq (L(K)+1)^{-1} < 2^{-K}$.

Following the preceding case distinction, we conclude that $\|\boldsymbol{x}'_m - \boldsymbol{x}_{m,K}\|_1 < 2^{-K}$ is satisfied for all $m, K \in \mathbb{N}$. In other words, for all $m \in \mathbb{N}$, $(\boldsymbol{x}_{m,k})_{k\in\mathbb{N}}$ converges effectively towards $\boldsymbol{x}'_m$ in ℓ^1_0, with respect to the μ-recursive modulus of convergence $\xi : \mathbb{N}^2 \to \mathbb{N}$, $(m, K) \mapsto K$. Consequently, $(\boldsymbol{x}'_m)_{m\in\mathbb{N}}$ is a computable sequence of sequences in $\mathcal{C}\ell^1_0$.

It remains to show that for all $m \in \Omega$, we have $\|f'_m\|_1 \geq 1$, while for all $m \in \mathbb{N} \setminus \Omega$, we have $\|f'_m\|_1 = 0$, which we again prove by case distinction.

First, recall that if $m \in \Omega$ is satisfied, f'_m is an elementary computable function such that for some $M \in \mathbb{N}$, we have $f_M = f'_m$. By assumption, $C \geq \|f_M\|_1 \geq 1$ is satisfied for some number $C > 0$ and all $M \in \mathbb{N}$. Hence, for all $m \in \Omega$ there exists an $M \in \mathbb{N}$ such that we have $C \geq \|f'_m\|_1 = \|f_M\|_1 \geq 1$.

Second, recall that if $m \in \mathbb{N} \setminus \Omega$ is satisfied, then f'_m is the trivial elementary computable function. That is, we have $f'_m \equiv 0$, in which case $\|f'_m\|_1 = 0$ holds true. Hence, for all $m \in \mathbb{N} \setminus \Omega$, we have $\|f'_m\|_1 = 0$.

Following the preceding case distinction, we conclude that for all $m \in \Omega$, we have $\|f'_m\|_1 \geq 1$, while for all $m \in \mathbb{N} \setminus \Omega$, we have $\|f'_m\|_1 = 0$. □

References

1. Turing, A.M. On computable numbers, with an application to the Entscheidungsproblem. *Proc. Lond. Math. Soc.* **1937**, *s2-42*, 230–265; Erratum in *Proc. Lond. Math. Soc.* **1938**, *s2-43*, 544–546. [CrossRef]
2. Shannon, C.E. Communication in the presence of noise. *Proc. IRE* **1949**, *37*, 10–21. [CrossRef]
3. Asadzadeh, M.; Beilina, L. Stability and Convergence Analysis of a Domain Decomposition FE/FD Method for Maxwell's Equations in the Time Domain. *Algorithms* **2022**, *15*, 337. [CrossRef]
4. Böck, Y.N.; Boche, H.; Schaefer, R.F.; Fitzek, F.H.P.; Poor, H.V. Virtual-Twin Technologies in Networking. *IEEE Commun. Mag.* **2023**, 1–7. [CrossRef]
5. Hong, H.; Wu, Q.; Dong, F.; Song, W.; Sun, R.; Han, T.; Zhou, C.; Yang, H. NetGraph: An intelligent operated digital twin platform for data center networks. In Proceedings of the ACM SIGCOMM 2021 Workshop on Network-Application Integration (NAI '21), virtually, 23–27 August 2021; pp. 26–32.

6. Tao, F.; Zhang, H.; Liu, A.; Nee, A.Y.C. Digital twin in industry: State-of-the-art. *IEEE Trans. Ind. Inform.* **2019**, *15*, 2405–2415. [CrossRef]
7. Fitzek, F.H.P.; Li, S.C.; Speidel, S.; Strufe, T.; Simsek, M.; Reisslein, M. (Eds.) *Tactile Internet with Human-in-the-Loop*; Academic Press: Cambridge, MA, USA, 2021.
8. Laubenbacher, R.; Niarakis, A.; Helikar, T.; An, G.; Shapiro, B.; Malik-Sheriff, R.S.; Sego, T.J.; Knapp, A.; Macklin, P.; Glazier, J.A. Building digital twins of the human immune system: Toward a roadmap. *NPJ Digit. Med.* **2022**, *5*, 64. [CrossRef] [PubMed]
9. Fettweis, G.P.; Boche, H. On 6G and trustworthiness. *Commun. ACM* **2022**, *65*, 48–49. [CrossRef]
10. Boche, H.; Böck, Y.; Deppe, C.; Fitzek, F.H.P. Networked Decision-Making through Virtual-Twin Technologies. *IEEE Trans. Autom. Control.* 2023, *Submitted for Review*.
11. Levin, B.Y. *Lectures on Entire Functions*; American Mathematical Society: Providence, RI, USA, 1996.
12. Higgins, J.R. *Sampling Theory in Fourier and Signal Analysis–Foundations*; Oxford University Press: Oxford, UK, 1996.
13. Kleene, S.C. General recursive functions of natural numbers. *Math. Ann.* **1936**, *112*, 727–742. [CrossRef]
14. Soare, R.I. *Recursively Enumerable Sets and Degrees*; Perspectives in Mathematical Logic; Springer: Berlin/Heidelberg, Germany, 1987.
15. Pour-El, M.B.; Richards, J.I. *Computability in Analysis and Physics*; Springer: Berlin/Heidelberg, Germany, 1989.
16. Weihrauch, K. *Computable Analysis: An Introduction*; Springer: Berlin/Heidelberg, Germany, 2000.
17. Boolos, G.S.; Burgess, J.P.; Jeffrey, R.C. *Computability and Logic*; Cambridge University Press: Cambridge, UK, 2002.
18. Turing, A.M. Computability and λ-definability. *J. Symb. Log.* **1937**, *2*, 153–163. [CrossRef]
19. Boche, H.; Mönich, U.J. Turing meets Shannon: Computable sampling type reconstruction with error control. *IEEE Trans. Signal Process.* **2020**, *68*, 6350–6365. [CrossRef]
20. Boche, H.; Mönich, U.J. Computable time concentration of bandlimited signals and systems. *IEEE Trans. Signal Process.* **2021**, *69*, 5523–5538. [CrossRef]

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