Optimal Sampling Rate and Bandwidth of Bandlimited Signals

—

An Algorithmic Perspective

Holger Boche\textsuperscript{1,2} and Ullrich J. Mönich\textsuperscript{1}

\textsuperscript{1}Technical University of Munich, Chair of Theoretical Information Technology
\textsuperscript{2}Munich Center for Quantum Science and Technology (MCQST)

Motivation

• The actual bandwidth \( B(f) \) of a bandlimited signal is a key quantity. Relevant in numerous applications (e.g. wireless communications).

• The bandwidth determines the minimum sampling rate (Nyquist rate) that is necessary to reconstruct a bandlimited signal from its samples (Shannon sampling series).

\[
B(f) \to r_{\text{min}} = \frac{B(f)}{\pi} \to \{f(k/r_{\text{min}})\}_{k \in \mathbb{Z}}
\]

• The sequence of samples \( \{f(k\pi/B(f))\}_{k \in \mathbb{Z}} \), taken at Nyquist rate, can therefore be seen as a minimum representation of the signal \( f \) (no loss of information).

Can we determine the actual bandwidth \( B(f) \) of a bandlimited signal \( f \), i.e., the smallest number \( \sigma \) such that \( f \) is bandlimited with bandwidth \( \sigma \), on a digital computer?
Turing Machine I

**Turing Machine:**
Abstract device that manipulates symbols on a strip of tape according to certain rules.

- Turing machines are an *idealized computing model*.
- No limitations on *computing time* or *memory*, no computation errors.
- Although the concept is *very simple*, Turing machines are capable of simulating *any given algorithm*.

Turing machines are suited to study the *limitations in performance of a digital computer*:

> Anything that is not Turing computable cannot be computed on a real digital computer, regardless how powerful it may be.

---


Turing Machine II

- There exist problems that cannot be solved on a digital computer.
- For example, computation of Fourier transform or spectral factorization for certain signals.
- The computer cannot produce, for any desired error $\varepsilon$, a result that $\varepsilon$-close to the true value. 
  $\rightarrow$ the approximation error cannot be controlled.
Questions

Questions:

1. Is $B(f)$ computable?
2. Can we compute a lower bound for $B(f)$?
3. Can we compute an upper bound for $B(f)$?
Questions:

1. Is $B(f)$ computable?
2. Can we compute a lower bound for $B(f)$?
3. Can we compute an upper bound for $B(f)$?

In terms of sampling rate, those questions read:

1. Is the optimal, i.e. minimum required sampling rate computable?
2. Can we compute a lower bound for the minimum required sampling rate?
3. Can we compute an upper bound for the minimum required sampling rate?
**Overview of the Results**

<table>
<thead>
<tr>
<th>Actual bandwidth $B(f)$ computable?</th>
<th>Determine if $\sigma$ is a lower bound for $B(f)$?</th>
<th>Determine if $\sigma$ is an upper bound for $B(f)$?</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
Notation

- \( L^p(\Omega), 1 \leq p < \infty \): space of all measurable, \( p \)-th-power Lebesgue integrable functions on \( \Omega \)
  Norm: \( \|f\|_p = \left( \int_{\Omega} |f(t)|^p \, dt \right)^{1/p} \)
- \( L^\infty(\Omega) \): space of all functions for which the essential supremum norm \( \| \cdot \|_\infty \) is finite
Bandlimited Functions

**Definition (Bernstein Space)**

Let $\mathcal{B}_\sigma$ be the set of all entire functions $f$ with the property that for all $\epsilon > 0$ there exists a constant $C(\epsilon)$ with $|f(z)| \leq C(\epsilon) \exp((\sigma + \epsilon)|z|)$ for all $z \in \mathbb{C}$.

The Bernstein space $\mathcal{B}^p_\sigma$ consists of all functions in $\mathcal{B}_\sigma$, whose restriction to the real line is in $L^p(\mathbb{R})$, $1 \leq p \leq \infty$. The norm for $\mathcal{B}^p_\sigma$ is given by the $L^p$-norm on the real line.

- A function in $\mathcal{B}^p_\sigma$ is called **bandlimited** to $\sigma$.
- $\mathcal{B}(f)$: actual bandwidth of the function $\mathcal{B}(f) = \min\{\sigma \in \mathbb{R} : f \in \mathcal{B}_\sigma\}$
- We have $\mathcal{B}^p_\sigma \subset \mathcal{B}^r_\sigma$ for all $1 \leq p \leq r \leq \infty$.
- $\mathcal{B}^2_\sigma$: space of bandlimited functions with **finite energy**.
Actual Bandwidth for $\mathcal{B}_\pi^2$

For $f \in \mathcal{B}_\pi^2$ we have a simple characterization of the actual bandwidth.

• $B(f)$ is the smallest number $\sigma > 0$ such that
\[
\int_{-\infty}^{\infty} |f(t)|^2 \, dt = \frac{1}{2\pi} \int_{-\sigma}^{\sigma} |\hat{f}(\omega)|^2 \, d\omega.
\]

• $B(f)$ is the smallest $\sigma > 0$ such that
\[
f(t) = \frac{1}{2\pi} \int_{-\sigma}^{\sigma} \hat{f}(\omega) e^{i\omega t} \, d\omega
\]

for all $t \in \mathbb{R}$. 
A sequence of rational numbers \( \{r_n\}_{n \in \mathbb{N}} \) is called computable sequence if there exist recursive functions \( a, b, s \) from \( \mathbb{N} \) to \( \mathbb{N} \) such that \( b(n) \neq 0 \) for all \( n \in \mathbb{N} \) and

\[
    r_n = (-1)^{s(n)} \frac{a(n)}{b(n)}, \quad n \in \mathbb{N}.
\]

- A recursive function is a function, mapping natural numbers into natural numbers, that is built of simple computable functions and recursions. Recursive functions are computable by a Turing machine.
Computable Real Numbers

First example of an effective approximation

A real number $x$ is said to be **computable** if there exists a computable sequence of rational numbers $\{r_n\}_{n \in \mathbb{N}}$ and a recursive function $\xi : \mathbb{N} \rightarrow \mathbb{N}$ such that for all $M \in \mathbb{N}$ we have

$$|x - r_n| \leq 2^{-M}$$

for all $n \geq \xi(M)$.

- $\mathbb{R}_c$: set of computable real numbers
- $\mathbb{R}_c$ is a field, i.e., finite sums, differences, products, and quotients of computable numbers are computable.
- Commonly used constants like $e$ and $\pi$ are computable.
We call a function \( f \) elementary computable if there exists a natural number \( L \) and a sequence of computable numbers \( \{\alpha_k\}_{k=-L}^L \) such that

\[
f(t) = \sum_{k=-L}^L \alpha_k \frac{\sin(\pi(t - k))}{\pi(t - k)}.
\]

- Every elementary computable function is Turing computable.
- For every elementary computable function \( f \), the norm \( \| f \|_{\mathcal{B}_\pi^p} \) is computable.
A function in $f \in B^p_{\pi}$ is called computable in $B^p_{\pi}$ if there exists a computable sequence of elementary computable functions $\{f_n\}_{n \in \mathbb{N}}$ and a recursive function $\xi: \mathbb{N} \rightarrow \mathbb{N}$ such that for all $M \in \mathbb{N}$ we have

$$\|f - f_n\|_p \leq 2^{-M}$$

for all $n \geq \xi(M)$.

- $CB^p_{\pi}$: set of all functions that are computable in $B^p_{\pi}$. 
A function in $f \in B^p_{\pi}$ is called **computable in $B^p_{\pi}$** if there exists a computable sequence of elementary computable functions $\{f_n\}_{n \in \mathbb{N}}$ and a recursive function $\xi : \mathbb{N} \rightarrow \mathbb{N}$ such that for all $M \in \mathbb{N}$ we have

$$\|f - f_n\|_p \leq 2^{-M}$$

for all $n \geq \xi(M)$.

- $\mathcal{CB}^p_{\pi}$: set of all functions that are computable in $B^p_{\pi}$. 
Question:
Does there exist an algorithm that, for every computable signal $f \in CB_1^\pi$ (or $f \in CB_2^\pi$), is able to compute $B(f)$?

Necessary condition: $B$ maps computable functions into computable numbers ($B: CB_1^\pi \rightarrow R_c$).

Weaker question: Do we have $B(f) \in R_c$ for all $f \in CB_1^\pi$ (or $f \in CB_2^\pi$)?

Theorem
There exists a signal $f_1 \in CB_1^\pi$ (and $CB_2^\pi$) such that $B(f_1) \not\in R_c$, i.e., $B(f_1)$ is not Turing computable.

The actual bandwidth is not always computable.
Computability of the Actual Bandwidth

**Question:**
Does there exist an algorithm that, for every computable signal $f \in CB_\pi^1$ (or $f \in CB_\pi^2$), is able to compute $B(f)$?

- **Necessary condition:** $B$ maps computable functions into computable numbers ($B : CB_\pi^1 \to \mathbb{R}_c$).

**Weaker question:**
Do we have $B(f) \in \mathbb{R}_c$ for all $f \in CB_\pi^1$ (or $f \in CB_\pi^2$)?

Theorem
There exists a signal $f_1 \in CB_\pi^1$ (and $CB_\pi^2$) such that $B(f_1) \not\in \mathbb{R}_c$, i.e., $B(f_1)$ is not Turing computable.

The actual bandwidth is not always computable.
Computability of the Actual Bandwidth

**Question:**
Does there exist an algorithm that, for every computable signal $f \in \mathcal{CB}_1^\pi$ (or $f \in \mathcal{CB}_2^\pi$), is able to compute $B(f)$?

- Necessary condition: $B$ maps computable functions into computable numbers ($B: \mathcal{CB}_1^\pi \rightarrow \mathbb{R}_c$).

**Weaker question:**
Do we have $B(f) \in \mathbb{R}_c$ for all $f \in \mathcal{CB}_1^\pi$ (or $f \in \mathcal{CB}_2^\pi$)?

**Theorem**

There exists a signal $f_1 \in \mathcal{CB}_1^\pi$ (and $\mathcal{CB}_2^\pi$) such that $B(f_1) \not\in \mathbb{R}_c$, i.e., $B(f_1)$ is not Turing computable.

- The actual bandwidth is not always computable.
• There are **problematic signals** $f$, for which we cannot compute $B(f)$.

• Can we at least algorithmically determine whether, for a given signal $f$, we can compute $B(f)$, or not?

• Would be helpful to avoid problematic signals, e.g., in automated computer aided design (CAD).
Semi Decidability I

- There are problematic signals $f$, for which we cannot compute $B(f)$.
- Can we at least algorithmically determine whether, for a given signal $f$, we can compute $B(f)$, or not?
- Would be helpful to avoid problematic signals, e.g., in automated computer aided design (CAD).

We call a set $\mathcal{M} \subset CB^1_\pi$ semi-decidable if there exists a Turing machine

$$TM: CB^1_\pi \rightarrow \{TM \text{ stops, } TM \text{ runs forever}\}$$

that, given an input $f \in CB^1_\pi$, stops if and only if $f \in \mathcal{M}$.
Semi Decidability II

Set of all signals in $\mathcal{CB}^1_\pi$ for which $B(f)$ can be computed algorithmically:

\[ \mathcal{C}^1_{BW} = \{ f \in \mathcal{CB}^1_\pi : B(f) \in \mathbb{R}_c \} \]

Set of all signals in $\mathcal{CB}^1_\pi$, for which $B(f)$ cannot be computed algorithmically:

\[ \mathcal{NC}^1_{BW} = \mathcal{CB}^1_\pi \setminus \mathcal{C}^1_{BW} = \{ f \in \mathcal{CB}^1_\pi : B(f) \not\in \mathbb{R}_c \} \]
Semi Decidability II

Set of all signals in $\mathcal{C}B^1_\pi$ for which $B(f)$ can be **computed algorithmically**:

$$\mathcal{C}^1_{BW} = \{ f \in \mathcal{C}B^1_\pi : B(f) \in \mathbb{R}_c \}$$

Set of all signals in $\mathcal{C}B^1_\pi$, for which $B(f)$ **cannot** be computed algorithmically:

$$\mathcal{NC}^1_{BW} = \mathcal{C}B^1_\pi \setminus \mathcal{C}^1_{BW} = \{ f \in \mathcal{C}B^1_\pi : B(f) \notin \mathbb{R}_c \}$$

Can we determine algorithmically whether $f \in \mathcal{NC}^1_{BW}$?
Semi Decidability II

Set of all signals in $\mathcal{CB}_\pi^1$ for which $B(f)$ can be computed algorithmically:

$$C_{BW}^1 = \{ f \in \mathcal{CB}_\pi^1 : B(f) \in \mathbb{R}_c \}$$

Set of all signals in $\mathcal{CB}_\pi^1$, for which $B(f)$ cannot be computed algorithmically:

$$\mathcal{NC}_{BW}^1 = \mathcal{CB}_\pi^1 \setminus C_{BW}^1 = \{ f \in \mathcal{CB}_\pi^1 : B(f) \notin \mathbb{R}_c \}$$

Can we determine algorithmically whether $f \in \mathcal{NC}_{BW}^1$?

Theorem

Neither $C_{BW}^1$ nor $\mathcal{NC}_{BW}^1$ is semi-decidable.

Optimal Sampling Rate and Bandwidth of Bandlimited Signals—An Algorithmic Perspective

16
For $\sigma > 0$, we want to algorithmically determine if $B(f) > \sigma$. 
For $\sigma > 0$, we want to algorithmically determine if $B(f) > \sigma$.

**Theorem**

For all $\sigma \in (0, \pi) \cap \mathbb{R}_c$ the set

$$C_1^1(\sigma) = \{ f \in CB^1_\pi : B(f) > \sigma \}$$

is semi-decidable.

- There exists an algorithm that stops if and only if $B(f) > \sigma$. 
For $\sigma > 0$, we want to algorithmically determine if $B(f) > \sigma$.

**Theorem**

For all $\sigma \in (0, \pi) \cap \mathbb{R}_c$ the set

$$C^1_{\geq}(\sigma) = \{ f \in C^B_\pi: B(f) > \sigma \}$$

is semi-decidable.

- There exists an algorithm that stops if and only if $B(f) > \sigma$.
- Does not allow us to determine an effective upper bound for $B(f)$, because this Turing machine does not stop if $B(f) \leq \sigma$. 

Approximate Bandwidth I
Theorem

For all $\sigma \in (0, \pi) \cap \mathbb{R}_c$ the set

$$C^1_{\leq}(\sigma) = \{ f \in CB^1_{\pi}: B(f) \leq \sigma \}$$

is not semi-decidable.
### Theorem

For all \( \sigma \in (0, \pi) \cap \mathbb{R}_c \) the set

\[
\mathcal{C}^1_{\leq}(\sigma) = \{ f \in \mathcal{CB}^1_{\pi} : B(f) \leq \sigma \}
\]

is not semi-decidable.

### Consequence:

- For a given \( \sigma \in (0, \pi) \cap \mathbb{R}_c \), we **cannot determine algorithmically** for all \( f \in \mathcal{CB}^1_{\pi} \) whether \( f \) is uniquely determined by the samples \( \{ f(k\pi/\sigma) \}_{k \in \mathbb{Z}} \).
Conclusions

- We studied if it is possible to algorithmically determine the actual bandwidth of a bandlimited signal.
- We proved that this is not possible in general.
- The minimal sampling rate cannot be determined algorithmically in general.
Thank you!