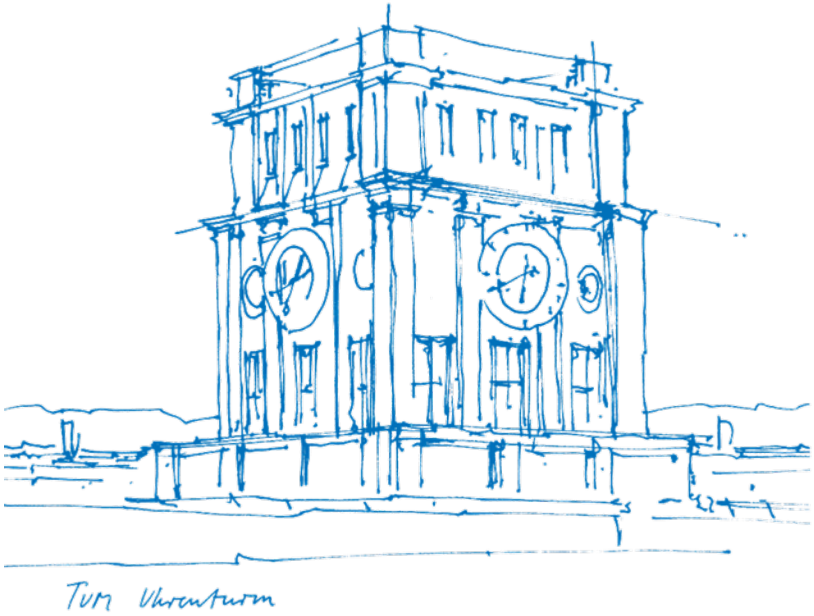
Can every analog system be simulated on a digital computer?

Holger Boche Volker Pohl

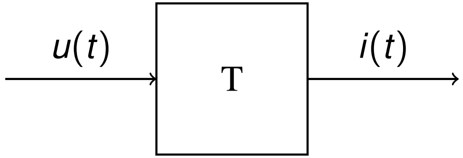
Technical University of Munich
Department of Electrical and Computer Engineering
Chair of Theoretical Information Technology

45th IEEE International Conference on
Acoustics, Speech, and Signal Processing (ICASSP)
May 8, 2020



Motivation – Simulation of Continuous LTI Systems

- Consider a linear *time-invariant (LTI) system* T mapping an input signal $u(t)$ onto an output signal $i(t)$.



- Input and output are described by *continuous* variables $u(t)$ and $i(t)$, respectively, which live in an *uncountable set* (e.g. $\mathbb{R}$, $\mathbb{C}$, $\mathbb{R}^N$). Often, these quantities are currents, voltages, potentials, etc.
- The input and output variables often depend on a *continuous* parameters t in *uncountable set* like. $\mathbb{R}$, $\mathbb{C}$, $\mathbb{R}^N$, etc. These parameters t often describes „time“ or „space“ (position).
- Assume there is a mathematical model for T describing the relation between input and output: $i(t) = (\mathrm{T}u)(t)$

Question:

Assume u is an arbitrary *admissible input* for T. Is it possible to calculate $i(t) = (\mathrm{T}u)(t)$ on a *digital* computer?

Problem: Digital computers can exactly solve only finite discrete problems.

Outline

1. Review of computability theory
   - computable numbers, computable functions, Turing machines, etc.

2. The most simple LTI system with non-computable output – Ideal capacitor

3. Classes of input signals with a computable output

4. Summary and outlook

Computability Analysis

Computable Rational Numbers

Definition: A sequence $\{r_n\}_{n\in\mathbb{N}} \subset \mathbb{Q}$ of rational numbers is said to be computable if there exist recursive functions $a, b, s : \mathbb{N} \to \mathbb{N}$ with $b(n) \neq 0$ and such that

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

A recursive function $a : \mathbb{N} \to \mathbb{N}$ is a mapping that is build form elementary computable functions and recursion and can be calculated on a *Turing machine*.

Turing machine

- can simulate any given algorithm and therewith provide a simple but very powerful model of computation.
- is a theoretical model describing the fundamental limits of any realizable digital computer.
- Most powerful programming languages are called Turing-complete (such as C, C++, Java, etc.).

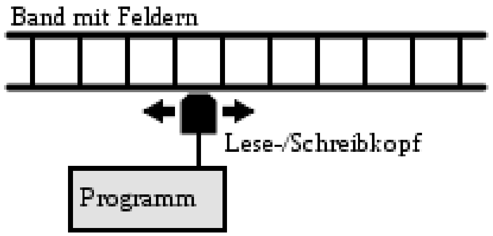


Figure taken from *Wikipedia*

A. M. Turing, "On computable numbers, with an application to the Entscheidungsproblem," *Proc. London Math. Soc.*, vol. s2-42, no. 1, 1937.

Computable Real Numbers

- ▷ Any real number $x \in \mathbb{R}$ is the limit of a sequence of rational numbers.
- ▷ For $x \in \mathbb{R}$ to be computable, the convergence has to be effective.

Definition (Computable number): A real number $x \in \mathbb{R}$ is said to be *computable* if there exists a computable sequence $\{r_n\}_{n\in\mathbb{N}} \subset \mathbb{Q}$ of rational numbers which *converges effectively* to x,
i.e. if there exists a recursive function $e : \mathbb{N} \to \mathbb{N}$ such that for all $N \in \mathbb{N}$

$$|x - r_n| \leq 2^{-N} \qquad \text{whenever} \quad n \geq e(N)\,.$$

$\Rightarrow$ $x \in \mathbb{R}$ is computable if a Turing machine can approximate it with exponentially vanishing error.

- $\mathbb{R}_c$ stand for the set of all *computable real numbers*.
- $\mathbb{C}_c = \{x + iy : x, y \in \mathbb{R}_c\}$ stands for the set of all *computable complex numbers*.
- Note that the set of computable numbers $\mathbb{R}_c \subsetneq \mathbb{R}$ is only countable.

Computable Functions

Definition: A function $f : \mathbb{T} \to \mathbb{R}$ on an interval $\mathbb{T} \subset \mathbb{R}$ is said to be computable if

(a) f is Banach–Mazur computable, i.e. if f maps computable sequences $\{t_n\}_{n\in\mathbb{N}} \subset \mathbb{R}_c$ onto computable sequences $\{f(t_n)\}_{n\in\mathbb{N}} \subset \mathbb{R}_c$.

(b) f is effective uniformly continuous, i.e. if there is a recursive function $d : \mathbb{N} \to \mathbb{N}$ such that for every $N \in \mathbb{N}$ and all $t_1, t_2 \in \mathbb{T}$ with $|t_1 - t_2| \leq 1/d(N)$ always $|f(t_1) - f(t_2)| \leq 2^{-N}$ is satisfied.

Lemma (equivalent definition of computability):
A function $f : \mathbb{T} \to \mathbb{R}$ is computable if and only if there exists a computable sequence of rational polynomials $\{p_m\}_{m\in\mathbb{N}}$ which *converges effectively* to f in the uniform norm, i.e. if there exists a recursive function $e : \mathbb{N} \to \mathbb{N}$ such that for all $t \in \mathbb{T}$ and every $N \in \mathbb{N}$

$$m \geq e(N) \qquad \text{implies} \qquad \left|f(t) - p_m(t)\right| \leq 2^{-N} .$$

Remark:

- There exist various notions of computability e.g. *Borel- or Markov computability*.
- Banach–Mazur computability is the weakest form of computability.
  $\Rightarrow$ If a function is not Banach–Mazur computable then it is not computable with respect to any other notion of computability.

Computable Functions in Banach Spaces

- We consider 2π-periodic functions on $\mathbb{R}$.
- We write $\mathbb{T} = \mathbb{R}/2\pi\mathbb{Z}$ for the additive quotient group of real numbers modulo 2π (think of $\mathbb{T} = [-\pi, \pi)$).
- Let $\mathscr{X}$ be a Banach space of functions on $\mathbb{T}$ with norm $\|f\|_{\mathscr{X}}$.

Definition: A function $f \in \mathscr{X}$ is said to be *$\mathscr{X}$–computable* if

(a) f is computable (i.e. effectively approximable by rational polynomials p_m).

(b) its norm $\|f\|_{\mathscr{X}}$ is computable $\Rightarrow \|f - p_m\|_{\mathscr{X}}$ converges to zero effectively as $m \to \infty$.

The set of all $\mathscr{X}$-computable functions is denoted by $\mathscr{X}_{\mathrm{c}}$.

For continuous functions $\mathscr{C}(\mathbb{T})$, computability implies $\mathscr{C}(\mathbb{T})$-computability.

Lemma:

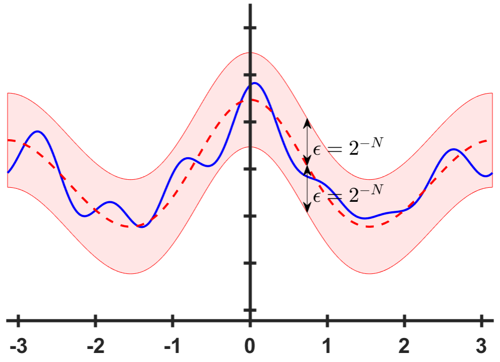
Let $f : \mathbb{T} \to \mathbb{R}$ be a computable function. Then f is computable as a continuous function, i.e. $f \in \mathscr{C}_{\mathrm{c}}(\mathbb{T})$.

J. Avigad and V. Brattka, "Computability and analysis: The legacy of Alan Turing," in *Turing's legacy: developments from Turing's ideas in logic*, ser. Lecture Notes in Logic, Bd. 42. New York: Cambridge University Press, 2014, pp. 1–47.

K. Weihrauch, *Computable Analysis*. Berlin: Springer-Verlag, 2000.

Computability of the Output– Intuition

- The output $i(t) = (\mathrm{T}u)(t)$ of T is usually not explicitly known.
- A function $i(t)$ is computable if it can be *approximated effectively* by a function $p_M(t)$ which can *perfectly be calculated* on a digital computer.
  - $p_M(t)$ might be a rational polynomial of appropriate degree M
  - effective approximation $\Rightarrow$ one can control the approximation error



Computability (an informal definition)

The output $i(t) = (\mathrm{T}u)(t)$ is computable if there exists an algorithm with the following properties

- It can be implemented on a digital computer (a Turing machine).
- It has two inputs: 1. the input $u(t)$ of T 2. an error bound $\varepsilon > 0$.
- It is able to determine in finitely many steps an approximation $p_M(t)$ of $i(t)$ such that the true $i(t)$ is guaranteed to be close to $p_M(t)$, i.e. such that

$$i \in \{f \in \mathscr{X} \ : \ \|f - p_M\|_{\mathscr{X}} < \varepsilon\}$$

where $\mathscr{X}$ is an appropriate Banach space with a corresponding norm $\|\cdot\|_{\mathscr{X}}$.

The Ideal Capacitor

Ideal Capacitor – Negative Result

- We consider the system T given by the voltage-current relation on an ideal capacitor with capacitance C.
- Applying a time-variant voltage $u(t)$ to a capacitor, the corresponding current $i(t)$ is known to be given by

$$i(t) = (\mathrm{T}u)(t) = C\frac{\mathrm{d}u}{\mathrm{d}t}(t) = C\,u'(t)\,, \qquad t \in \mathbb{R}\,. \tag{1}$$

- $C \in \mathbb{R}_{\mathrm{c}}$ is a computable real number $\Rightarrow$ the „system T is commutable“.
- Input signals: $u \in \mathscr{C}^1(\mathbb{T})$, i.e. 2π-periodic continuously differentiable functions on $\mathbb{T} \Rightarrow$ admissible.

Question: Let $u \in \mathscr{C}^1(\mathbb{T}) \cap \mathscr{C}_{\mathrm{c}}(\mathbb{T})$ be an arbitrary admissible input signal for the system T which is additionally computable. Is it true that also the output $i \in \mathscr{C}(\mathbb{T})$ is a computable continuous function?

Theorem: There exists an $u \in \mathscr{C}^1(\mathbb{T})$ with the following properties

1. $u \in \mathscr{C}_c(\mathbb{T})$, i.e. u is a computable continuous function.
2. $u' \in \mathscr{C}(\mathbb{T})$ is absolute continuous and u' has an absolute converging Fourier series.
3. $i(0) = C\frac{\mathrm{d}u}{\mathrm{d}t}(0) = Cu'(0) \notin \mathbb{R}_{\mathrm{c}}$, i.e. the value of the output current at $t = 0$ is not computable.

Ideal Capacitor – Remarks and Further Questions

- Proof: Explicit construction of $u \in \mathscr{C}^1(\mathbb{T})$ such that $u'(0)$ is not computable.
- Similar result for the ideal inductor: $u(t) = Li'(t)$.
- Every non-trivial circuit contains capacitors or inductors.

Previous result holds for a subset $\mathscr{S} \subset \mathscr{C}^1(\mathbb{T}) \cap \mathscr{C}_{\mathrm{c}}(\mathbb{T})$ of all admissible and computable inputs signals (namely for those u for which u' is additionally absolute continuous and possesses an absolute converging Fourier series).

Question: Can we find (large) subsets $\mathscr{B} \subset \mathscr{C}^1(\mathbb{T}) \cap \mathscr{C}_{\mathrm{c}}(\mathbb{T})$ of all admissible and computable input signals such that for every $u \in \mathscr{B}$ the output $i(t) = (\mathrm{T}u)(t)$ of the ideal capacitor is guarenteed to be computable?

We present two sharp characterizations of such subsets:

1. in terms of the second derivative u''
2. in terms of the smoothness of u (in the Sobolev scale)

Good Input Set – In Terms of Second Derivative

Theorem: Let u be the input signal for the ideal capacitor with the following properties

1. $u \in \mathscr{C}^1(\mathbb{T}) \cap \mathscr{C}_c(\mathbb{T})$.
2. u' is absolute continuous and $u'' \in L^1_c(\mathbb{T})$.

Then the output current $i(t) = (\mathrm{T}u)(t) = C\,u'(t)$ is a computable continuous function, i.e. $i \in \mathscr{C}_c(\mathbb{T})$.

Remark:

- If the second derivative of the input signal u belongs to $L^1_c(\mathbb{T})$, then the output i is computable.
- The statement is sharp with respect to the requirement $u'' \in L^1_c(\mathbb{T})$.
  If $u'' \notin L^1_c(\mathbb{T})$ then the output i might not be computable.

Good Input Set – In Terms of Smoothness

- Let $s \in \mathbb{R}_c$, $s \geq 0$ be a computable number.
- For every $f \in L^2(\mathbb{T})$ with its Fourier series $f(t) = a_0/2 + \sum_{k=1}^{\infty} a_k \cos(kt) + b_k \sin(kt)$, we define the seminorm

$$\|f\|_{s,2} = \left(\sum_{k=1}^{\infty} k^{2s} \left[|a_k|^2 + |b_k|^2 \right] \right)^{1/2} . \tag{2}$$

- The set $H^s(\mathbb{T}) = \left\{ f \in \mathscr{C}(\mathbb{T}) : \|f\|_{s,2} < \infty \right\}$ equipped with the norm $\|f\|_{H^s(\mathbb{T})} = \max\left(\|f\|_\infty, \|f\|_{s,2} \right)$ becomes a Banach space.
- Parameter s characterizes the smoothness of the functions in $H^s(\mathbb{T})$.

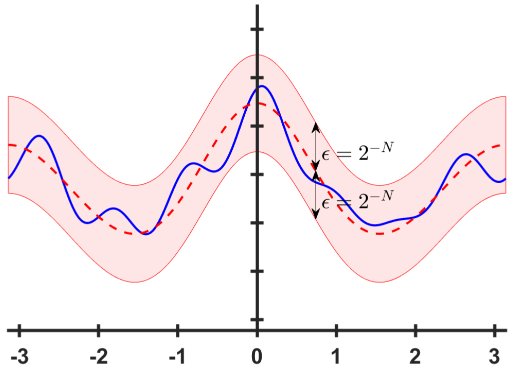
Theorem: Let $\mathrm{T} : u \mapsto i$ be the LTI system given by the ideal capacitor with a computable capacitance $C \in \mathbb{R}_c$, $C > 0$. Then for every $0 \leq s \leq 3/2$ there exists a computable input signal $u \in H^s_c(\mathbb{T})$ such that the output signal i is not computable.

Theorem: Let $s \in \mathbb{R}_c$, $s > 3/2$ and assume $u \in H^s_c(\mathbb{T})$, then $u' \in \mathscr{C}_c(\mathbb{T})$ and so $i = Cu' \in \mathscr{C}_c(\mathbb{T})$.

Summary and Outlook

- ▷ There is generally no closed form expression for the output $u(t) = (\mathrm{T}i)(t)$ of an LTI system.
  $\Rightarrow$ Numerically approximation methods (on digital computers) are applied to determine $u(t)$.

- ▷ Numerically approximation:
  Given input i and $\varepsilon > 0$, determine (in finite time) a confidence interval of width 2ε in which the (unknown) unknown output $u(t)$ lies. $\Rightarrow$ $u(t)$ is computable.



- ▷ Main result: For the ideal capacitor, there exist admissible and computable inputs $i(t)$ such that the corresponding output $u(t) = (\mathrm{T}i)(t)$ is not computable.

- ▷ Good input sets: We characterized sets $\mathscr{B}$ of admissible and computable inputs such that for every $i \in \mathscr{B}$ the corresponding output $u(t) = (\mathrm{T}i)(t)$ is computable.

- ▷ Outlook: Investigation of other systems, characterization of good input sets.