

Lesson 1.1: Introduction to DSP

- analysis: understand the information carried
- synthesis: create a signal to contain the given information

Signal processing operates on an abstract representation of a physical quantity and not the quantity itself.

A signal is:

- descripton of the evolution of a physical phenomenon
- equations
- definition: $f : \mathbb{R} \rightarrow \mathbb{R}$
- some function cannot be described as equations, or it is hard to do

The Digital Paradigm

discrete time:

it is very practical, for example finding the mean of a function is just a sum of values, on the other hand, the analog signal we must integrate. Thus, $f : Z \rightarrow V$, Z is the integers, and V is the real values. Simple math replaces calculus.

The equivalence between the discrete and continuous representations only holds for signals which are sufficiently “slow” with respect to how fast we sample them.

- Sampling theorem and Nyquist theorem
- Fourier analysis, the Fourier transform will tell us how fast a signal moves, and once we know that we may choose a sampling interval that allow us to reconstruct the signal.

discrete amplitude

Each sample may take values based on the numbers of levels, this is called **quantization**. Some beneficts: Storage (something that can store bits), processing (we just need a CPU) and transmtion (we might get high SNR).

Communication

Noise is everywhere, and it is important to know how to make the communication more robust.

Digital signal processing and information theory have been able to join forces so

successfully because they share a common data model (the integer) and therefore they share the same architecture (the processor). Computer code written to implement a digital filter can dovetail seamlessly with code written to implement error correction; linear processing and nonlinear flow control coexist naturally.

Lesson 1.2: Discrete-time signals

A sequence of **complex** numbers

- notation $x[n]$
- two-sided sequences: $x : \mathbb{Z} \rightarrow \mathbb{C}$
- n is a-dimensional *time*
- analysis: periodic measurements
- synthesis: stream of generated samples

the sequence index n is best thought of as a measure of *dimensionless time*.

The important point to make here is that, once a real-world signal is converted to a discrete-time representation, the underlying notion of “time between measurements” becomes completely abstract. All we are left with is a sequence of numbers, and all signal processing manipulations, with their intended results, are independent of the way the discrete-time signal is obtained.

Types:

- finite-length:
 - sequence notation: $x[n], n = 0, 1, \dots, N - 1$
 - vector notation: $x = [x_0 \dots x_{N-1}]^T$
 - practical entities, good for numerical packages (numpy)
 - Although we use the notation $x[n]$, $x[n]$ is not defined for values outside its support, i.e. for $n < 0$ or for $n \geq N$.
- infinite-length
 - sequence notation: $x[n], n \in \mathbb{Z}$
 - abstraction, theorems
- periodic
 - N-periodic sequence: $x[n] = x[n + kN], n, k, N \in \mathbb{Z}$
 - same information as finite-length of length N
 - “natural” bridge between finite and infinite length
 - Periodic extensions: $x[n] = x[n \bmod N], n \in \mathbb{Z}$
- finite-support
 - infinite length sequence, with only finite number of non-zero samples
 - same information as finite-length of length N
 - another bridge between finite and infinite lengths

Elementary operators

- Scaling: $y[n] = \alpha x[n]$
- Sum
- Product
- Integration
- Differentiation
- Shift by k (delay): $y[n] = x[n - k]$
 - if the signal is finite-lengths, we can either embed it into a finite-support signal or use a periodic signal (circular shift)

The reproducing formula

Any signal can be expressed as a linear combination of suitably weighed and shifted impulses. In this case, the weights are nothing but the signal values themselves.

$$x[n] = \sum_{k=-\infty}^{\infty} x[k] \delta[n - k]$$

Energy and Power

- Energy:

$$E_x = \sum_{n=-\infty}^{\infty} \|x[n]\|^2$$

This definition is consistent with the idea that, if the values of the sequence represent a time-varying voltage, the above sum would express the total energy (in joules) dissipated over a 1Ω -resistor.

- Power: it is the local energy computed over a window of size $2N + 1$ divided by the size of the window, in other words, it measures the energy of a signal per unit of time.

$$P_x = \lim_{N \rightarrow \infty} \frac{1}{2N + 1} \sum_{n=-N}^N \|x[n]\|^2$$

Signals with finite energy has zero power.

For periodic signal, the energy is ∞ and power is just defined over one period:

$$P_x = \frac{1}{N} \sum_{n=0}^{N-1} \|x[n]\|^2$$

Lesson 1.3: Basic signal processing

Discrete vs physical frequency

- Discrete time:
 - n: no physical dimension
 - periodicity: how many samples before the pattern repeats
- Physical world:
 - periodicity: how many seconds before the pattern repeats
 - frequency measures in Hertz
- set T_s time in seconds between samples and periodicity of M samples, so the real world frequency is

$$f = \frac{1}{MT_s} Hz$$

- zero initial conditions

Lesson 1.4: Complex exponentials

Oscillations are everywhere.

- Sustainable dynamic systems exhibit oscillatory behavior
- Things that don't move in circles can't last

Trigonometric functions describe oscillatory behaviors

In discrete time oscillatory system, there are some ingredients:

- a frequency ω , in radians
- an initial phase ϕ , in radians
- an amplitude A

Since every oscillatory system can be written as sines and cosine, it makes sense to use exponentials, also the math is easier.

Thus, a signal can be written as $x[n] = \exp(j(\omega n + \phi))$

In discrete time **not every sinusoid is periodic**. It is only periodic if $\exp(j\omega n)$ is periodic in n , which means $\omega = \frac{2\pi M}{N}$

For example $\exp(j\pi n/20)$ has a period of 40 samples.

$$\begin{aligned}x[n] &= x[n + N] \\ \exp(j\omega N) &= 1 \\ \omega N &= 2M\pi, M \in \mathbb{Z}\end{aligned}$$

Aliasing:

When one point have multiple names. For instance, $\exp(j\alpha)$ is in the same position as $\exp(j(2\pi + \alpha))$.

This set an upper limit of how fast we can go.