

## Contents

<b>1 Week 7 Module 5:</b>	<b>1</b>
1.1 Sampling and Quantization . . . . .	1
1.1.1 The Continous-Time World . . . . .	2
1.1.2 Interpolation . . . . .	6
1.1.3 Sampling of bandlimited functions . . . . .	10
1.1.4 Sampling of nonbandlimited functions . . . . .	14
1.1.5 Quantization . . . . .	18

## 1 Week 7 Module 5:

### 1.1 Sampling and Quantization

Interpolation describes the process of building a continuous-time signal  $\mathbf{x}(\mathbf{t})$  from a sequence of samples  $\mathbf{x}[\mathbf{n}]$ . In other words, interpolation allows moving from the discrete-time world to the continuous-time world. Interpolation raises two interesting questions:

The first one is how to interpolate between samples?

- In the case, of two samples, this is simple enough and there is there is a straight line that goes between these two samples.
- In the case of three samples, similarly, you have a parabola that goes through these 3 samples.
- If you have many samples, you can try to do the same and go through all samples but you see this is a trickier issue compared to what we have done with two or three samples.

The second question is:

- is there a minimum set of values you need to measure the function at so that you can perfectly reconstruct it.

Later on in the module, we are going to study sampling, i.e. the process of moving from a continuous-time signal to a sequence of samples. In other words, sampling allows moving from the continuous-time world to the discrete-time world. Suppose we take equally-spaced samples of a function  $\mathbf{x}(\mathbf{t})$ . The question is when is there a one-to-one relationship between the continuous-time function and its samples, i.e. when do the samples form a

unique representation of the continuous-time function? To answer this question, we are going to use all the tools in the toolbox that we have looked at so far:

- Hilbert spaces
- projections
- filtering
- sinc functions
- and so on.

Everything comes together in this module to develop a profound and very useful result, the **sampling theorem**.

Before moving to the heart of the topic, let us briefly review its history. The Shannon sampling theorem has a very interesting history which goes back well before Shannon. Numerical analysts were concerned about interpolating tables of functions and the first one to prove a version of the sampling theorem was Whittaker in England in 1915. Harry Nyquist at Bell Labs came up with the Nyquist criterion, namely that a function that has a maximum frequency  $F_0$  could be sampled at  $2F_0$ . In the Soviet Union, Kotelnikov proved a sampling theorem. The son of the first Whittaker further proved results on the sampling theorem. Then Herbert Raabe in Berlin wrote his PhD thesis about a sampling theorem that, wrong time wrong city, he got zero credit for. Denis Gabor worked on a version of the sampling theorem in the mid 1940s. Then Claude Shannon, the inventor of information theory, wrote a beautiful paper that is in the further reading for this class where the Shannon sampling theorem appears in the form that we use today. Last but not least, in 1949 Someya in Japan also proved the sampling theorem. You can see that it's a very varied history, it's a fundamental result where many people independently came up with this result.

### 1.1.1 The Continuous-Time World

1. Introduction The continuous-time world is the world we live in, the physical reality of the world, in contrast with the discrete-time world, the world inside a computer. We are first going to look at models of the world and compare digital with analog views of the world. Then we are going to study continuous-time signal processing in greater details.

Furthermore, we will introduce the last form of Fourier transform we have not yet encountered in this class, the continuous-time Fourier transform.

## 2. The continuous-time paradigm

Two views of the world

Table 1: Two views of the world 1

Digital World	Analog World
arithmetic	calculus
combinatorics	distributions
computer science	system theory
DSP	electronics

Table 2: Two views of the world 2

Digital World	Analog World
countable integer index $M$	real-valued time $t$ [sec]
sequences $x[n] \in \ell_2(\mathbb{Z})$	function $x(t) \in L_2(\mathbb{R})$
frequency $\omega \in [-\pi, \pi]$	frequency $\Omega \in \mathbb{R}(\text{rad/sec})$
DTFT: $\ell_2(\mathbb{Z}) \rightarrow L_2[-\pi, \pi]$	FT: $L_2(\mathbb{R}) \rightarrow L_2(\mathbb{R})$

$\ell_2(\mathbb{Z})$	Square-Summable infinite sequences
$L_2([a, b])$	Square-integrable functions over an interval
<b>Sampling</b>	$x(t) \rightarrow x[n]$
<b>Interpolation</b>	$x[n] \rightarrow x(t)$

## 3. Continuous-time signal processing

<b>time</b>	real variable $t$
<b>signal <math>x(t)</math></b>	complex function of real variable
<b>finite energy</b>	$x(t) \in L_2(\mathbb{R})$
<b>inner product in <math>L_2(\mathbb{R})</math></b>	$\langle x(t), y(t) \rangle = \int_{-\infty}^{\infty} x^*(t) y(t)$
<b>energy</b>	$\ x(t)\ ^2 = \langle x(t), x(t) \rangle$

(a) Analog LTI filters

$$\begin{aligned} y(t) &= (x * h)t \\ &= \langle h^*(t - \tau), x(\tau) \rangle \\ &= \int_{-\infty}^{\infty} x(\tau) h(t - \tau) d\tau \end{aligned}$$

(b) Fourier analysis

- in discrete time max angular frequency is  $\pm\pi$
- in continuous time no max frequency:  $\Omega \in \mathbb{R}$
- concept is the same:

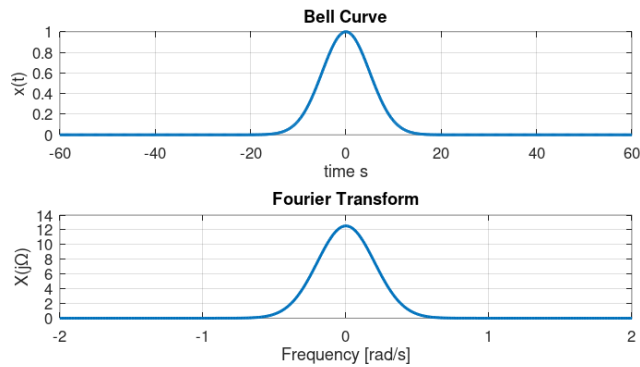
$$X(j\Omega) = \int_{-\infty}^{\infty} e^{-j\Omega t} dt$$

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\Omega) e^{j\Omega t} dt$$

(c) Real-world frequency

- $\Omega$  expressed in rad/s
- $F = \frac{\Omega}{2\pi}$ , expressed in Hertz (1/s)
- period  $T = \frac{1}{F} = \frac{2\pi}{\Omega}$

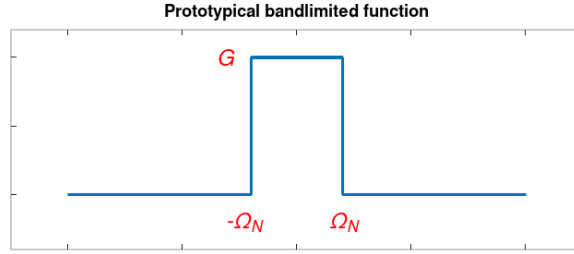
(d) Example



(e) Convolution theorem

$$Y(j\Omega) = X(j\Omega) H(j\Omega)$$

(f) Prototypical Bandlimited Functions



$$\Phi(j\Omega) = G \operatorname{rect}\left(\frac{\Omega}{2\Omega_N}\right)$$

The time domain function can be determined by means of its **Inverse Fourier Transform**

$$\begin{aligned}\phi(t) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \Phi(j\Omega) e^{j\Omega t} d\Omega \\ &= G \frac{\Omega_N}{\pi} \operatorname{sinc}\left(\frac{\Omega_N}{\pi} t\right)\end{aligned}$$

The time domain function is up to a scaling, one of these sinc functions. We will normalize this sinc function, so that the area is equal to  $2\pi$  in the Frequency Domain. Then the inverse continuous time Fourier Transform will have a maximum of 1 at the origin.

<b>normalization</b>	$G = \frac{\pi}{\Omega_N}$
<b>total bandwidth</b>	$\Omega_B = 2\Omega_N$
<b>define</b>	$T_s = \frac{2\pi}{\Omega_B} = \frac{\pi}{\Omega_N}$

This leads to the normalized prototypical bandlimited function:

#### Frequency Domain

$$\Phi(j\Omega) = \frac{\pi}{\Omega_N} \operatorname{rect}\left(\frac{\Omega}{2\Omega_N}\right)$$

#### Time Domain

$$\phi(t) = \operatorname{sinc}\left(\frac{t}{T_s}\right)$$

4. **TODO** Plot Normalized prototypical bandlimited function

### 1.1.2 Interpolation

**Main Task**  $x[n] \Rightarrow x(t)$

**Gaps** fill the gaps between samples

1. Interpolation requirements

- decide on  $T_s$
- make sure  $x(nT_s) = x[n]$
- make sure  $x(t)$  is smooth

2. Why smoothness

- jumps (1st order discontinuities) would require the signal to move "faster than light"
- 2nd order discontinuities would require infinite acceleration
- the interpolation should be infinitely differentiable
- "natural" solution: polynomial interpolation

3. Polynomial interpolation

- $N$  points  $\Rightarrow$  polynomial of degree  $(N-1)$
- $p(t) = a_0 + a_1t + a_2t^2 + \dots + a_{N-1}t^{(N-1)}$
- "naive" approach

$$\begin{cases} p(0) & = x[0] \\ p(T_s) & = x[1] \\ p(2T_s) & = x[2] \\ \dots\dots & \\ p((N-1)T_s) & = x[N-1] \end{cases}$$

Without loss of generality:

- consider a symmetric interval  $I_N = [-N \dots N]$
- set  $T_s = 1$

$$\begin{cases} p(-N) & = x[-N] \\ p(-N+1) & = x[-N+1] \\ \dots\dots & \\ p(0) & = x[0] \\ p(N) & = x[N] \end{cases}$$

4. Lagrange interpolation The natural solution to this interpolation problem is given by Lagrange interpolation

- $P_N$  : space of degree- $2N$  polynomials over  $I_N$
- a basis for  $P_N$  is the family of  $2N+1$  Lagrange polynomials

$$L_n^{(N)}(t) = \prod_{k=-N}^N \frac{t-k}{n-k} \text{ for } n = -N, \dots, N$$

The formula:

$$p(t) = \sum_{n=-N}^N x[n] L_n^{(N)}(t)$$

The Lagrange interpolation is the sought-after polynomial interpolation:

- polynomial of degree  $2N$  through  $2N+1$  points is unique

- the Lagrangian interpolator satisfies

$$p(N) = x[N] \text{ for } -N \leq M \leq N$$

since

$$L_n^{(N)}(N) = \begin{cases} 1 & \text{if } M = N \\ 0 & \text{if } M \neq N \end{cases} \quad -N \leq M, N \leq N$$

**key property**

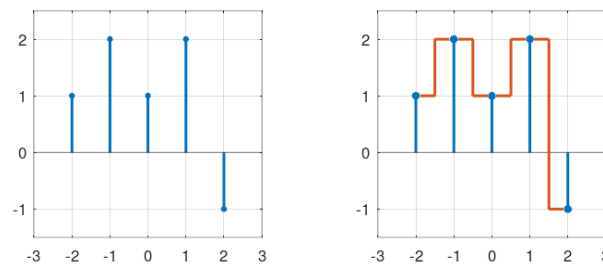
maximally smooth (infinitely many continuous derivatives)

**drawback**

interpolation "bricks" depend on N

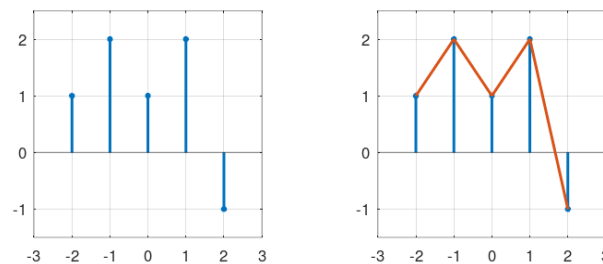
## 5. Local Interpolation

(a) Zero-order hold "Box Function"



- $x(t) = x[t + 0.5]$  for  $-N \leq t \leq N$
- $x(t) = \sum_{n=-N}^N x[n] \text{rect}(t - n)$
- interpolation kernel:  $i_0(t) = \text{rect}(t)$
- $i_0(t)$ : zero-order hold
- interpolation support is 1
- interpolation is not even continuous

(b) First-order piece-wise linear "Hat Function"



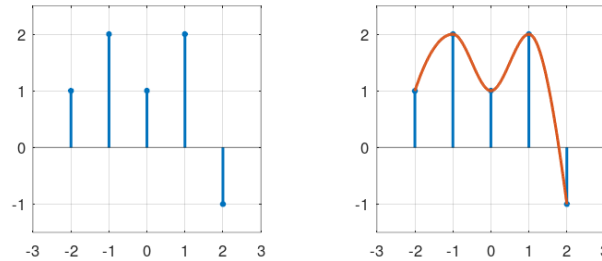


- connect the dots strategy
- $x(t) = \sum_{n=-N}^N x[n] i_1(t - n)$
- interpolation kernel:

$$i_1(t) = \begin{cases} 1 - |t| & |t| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

- interpolation support is 2
- interpolation is continuous but derivative is not

(c) Third-order interpolation



- $x(t) = \sum_{n=-N}^N x[n] i_3(t - n)$
- interpolation kernel obtained by splicing two cubic polynomials
- interpolation support is 4
- interpolation is continuous up to second derivative

(d) Local Interpolation schemes

$$x(t) = \sum_{n=-N}^N x[n] i_c(t - n)$$

Interpolator's requirements:

- $i_c$ : interpolation kernel
- $i_c(0) = 1$
- $i_c(t) = 0$

**Key property**

same interpolating function independently of N and of location

**drawback**

lack of smoothness

## 6. Sinc interpolation formula

A remarkable result:

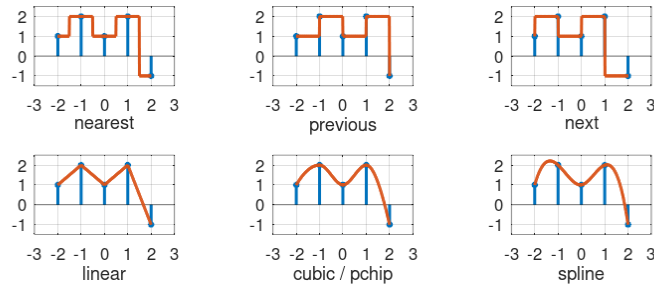
$$\lim_{N \rightarrow \infty} L_n^{(N)}(t) = \text{sinc}(t - n)$$

In the limit, local and global interpolation are the same!

$$x(t) = \sum_{n=-N}^N x[n] \text{sinc}\left(\frac{t - nT_s}{T_s}\right)$$

## 7. Octave Interpolation Overview

**Octave manual** Chapter 29.1 One-dimensional Interpolation

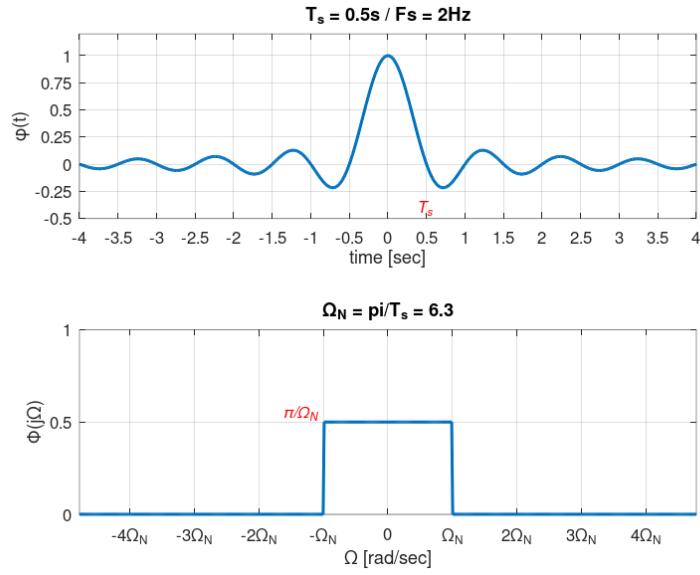


$$x(t) = \sum_{n=-N}^N x[n] i_c(t - n)$$

## 1.1.3 Sampling of bandlimited functions

## 1. Key Facts about the sinc

$$\begin{aligned} \phi(t) = \text{sinc}\left(\frac{t}{T_s}\right) &\longleftrightarrow \Phi(j\Omega) = \frac{\pi}{\Omega_N} \text{rect}\left(\frac{\Omega}{2\Omega_N}\right) \\ T_s = \frac{\pi}{\Omega_n} &\quad \Omega_N = \frac{\pi}{T_s} \end{aligned}$$



2. The spectrum of interpolated signals What is the spectrum of the

$$x(t) = \sum_{n=-\infty}^{\infty} x[n] \text{sinc}\left(\frac{t - nT_s}{T_s}\right)$$

$$\begin{aligned}
X(j\Omega) &= \int_{-\infty}^{\infty} x(t) e^{-j\Omega t} dt \\
&= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x[n] \text{sinc}\left(\frac{t - nT_s}{T_s}\right) e^{-j\Omega t} dt \\
&= \sum_{n=-\infty}^{\infty} x[n] \int_{-\infty}^{\infty} \text{sinc}\left(\frac{t - nT_s}{T_s}\right) e^{-j\Omega t} dt \\
&= \sum_{n=-\infty}^{\infty} x[n] \left(\frac{\pi}{\Omega_N}\right) \text{rect}\left(\frac{\Omega}{2\Omega_N}\right) e^{-jnT_s\Omega} \\
&= \left(\frac{\pi}{\Omega_N}\right) \text{rect}\left(\frac{\Omega}{2\Omega_N}\right) \sum_{n=-\infty}^{\infty} x[n] e^{-j(\pi/\Omega_N)\Omega n} \\
&= \begin{cases} \left(\frac{\pi}{\Omega_N}\right) X(e^{j\pi(\Omega/\Omega_N)}) & |\Omega| \leq \Omega_N \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

#### Spectrum of Sinc-Sampling

The spectrum of  $x(t)$  is equal to the scaled version of the DTFT of the sequence between  $-\Omega_N$  and  $\Omega_N$ .

Pick interpolation period  $T_s$ :

- $X(j\Omega)$  is  $\Omega_N$  - *bandlimited*, with  $\Omega_N = \pi/T_s$
- fast interpolation ( $T_s$  small)  $\Rightarrow$  wider spectrum
- slow interpolation ( $T_s$  large)  $\Rightarrow$  narrower spectrum

#### 3. The space of bandlimited functions

##### Claims:

- the space of  $\Omega_N$  - *bandlimited* functions is a Hilbert space
- the functions  $\phi^{(n)}(t) = \text{sinc}((t - n)/T_s)$ , with  $n \in \mathbb{Z}$  form a basis for the space
- if  $x(t)$  is  $\pi$  - *BL* the sequence  $x[n] = x(nT_s)$ , with  $n \in \mathbb{Z}$ , is a sufficient representation, i.e. we can reconstruct  $x(t)$  from  $x[n]$

The space  $\pi - BL$

- is a vector space because  $\pi - BL \subset L_2(\mathbb{R})$
- inner product is standard inner product in  $L_2(\mathbb{R})$
- completeness... that's more delicate



### Basis for $\pi - BL$

↗ The sinc function is an orthonormal basis for the  $\pi - BL$  space.

Inner product:

$$\langle x(t), y(t) \rangle = \int_{-\infty}^{\infty} x^*(t)y(t)dt$$

Convolution:

$$(x * y)(t) = \langle x^*(\tau), y(t - \tau) \rangle$$

A basis for the  $\pi - BL$  space

$$\phi^{(M)}(t) = \text{sinc}(t - n), \text{ for } M \in \mathbb{Z}$$

$$FT \text{sinc}(t) = \text{rect}\left(\frac{\Omega}{2\pi}\right)$$

$$(\text{sinc} * \text{sinc})(m - n) = \begin{cases} 1 & \text{for } m = n \\ 0 & \text{otherwise} \end{cases}$$

## 4. The sampling Theorem

- (a) Sampling as a basis expansion To see sampling as an orthonormal expansion, we take our sample of orthonormal vectors  $\phi^{(n)}$ , taking a product with  $x$  and we look what comes out.

### Analysis Formula

$$x[n] = \langle \text{sinc}\left(\frac{t - nT_s}{T_s}\right), x(t) \rangle = T_s x(nT_s)$$

### Synthesis Formula

$$x(t) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} x[n] \text{sinc}\left(\frac{t - nT_s}{T_s}\right)$$

- the space of  $\Omega_N$  – *bandlimited* functions is a Hilbert space
- set  $T_s = \pi/\Omega_N$
- the functions  $\phi^{(n)}(t) = \text{sinc}((t - nT_s)/T_s)$  form a basis for the space
- for any  $x(t) \in \Omega_N - BL$  the coefficients in the sinc basis are the (scaled) samples  $T_s x(nT_s)$

### Corollary

for any  $x(t) \in \Omega_N - BL$ , a sufficient representation is the sequence  $x[n] = x(nT_s)$

### The sampling theorem in Hertz

Any signal  $x(t)$  bandlimited to  $F_N$  Hz can be sampled with no loss of information using a sampling frequency  $F_s \geq 2F_N$  (i.e. sampling period  $T_s \leq 1/2 F_N$ )

#### 1.1.4 Sampling of nonbandlimited functions

1. Raw Sampling Raw sampling is when we don't care about first taking the inner product with the sinc function. So we just take  $x(t)$  and every  $T_s$  seconds, we take a sample.

The continuous-time complex exponential

Table 3: Aliasing

sampling period	digital frequency	$\{\hat{x}\}$
$T_s < \pi/\Omega_0$	$0 < \omega_0 < \pi$	$e^{j\Omega_0}$
$\pi/\Omega_0 < T_s < 2\pi/\Omega_0$	$\pi < \omega_0 < 2\pi$	$e^{j\Omega_1}: \Omega_1 = \Omega_0 - 2\pi/T_s$
$T_s > 2\pi/\Omega_0$	$\omega_0 > 2\pi$	$e^{j\Omega_2}: \Omega_2 = \Omega_0 \bmod(2\pi/T_s)$

## 2. Sinusoidal Aliasing

$$x(t) = \cos(2\pi f_0 t)$$

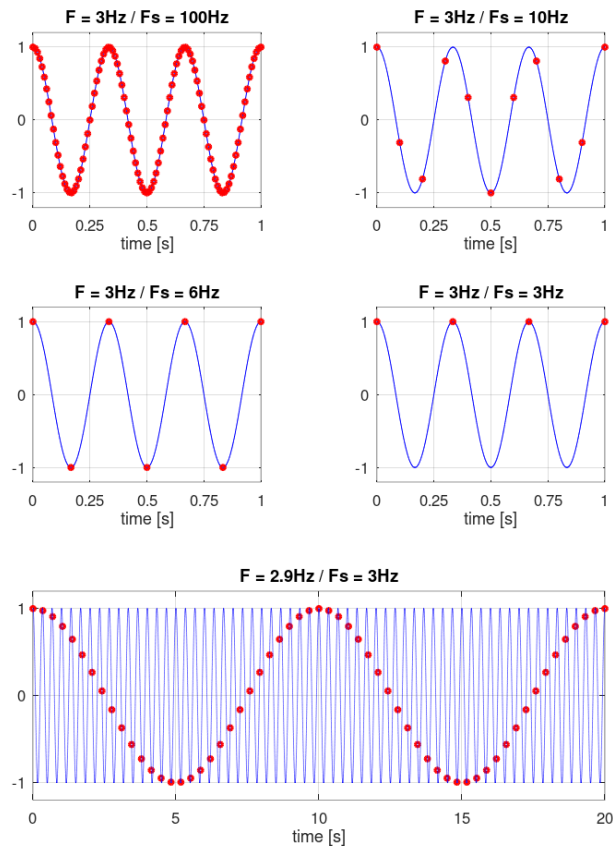
$$x[n] = x(nT_s) = \cos(\omega_0 n)$$

with

$$F_s = \frac{1}{T_s}$$

$$\omega_0 = 2\pi \left( \frac{F_0}{F_s} \right)$$

(a) Aliasing: Sampling a Sinusoid



3. Aliasing for arbitrary spectra A continuous time signal  $x_c$  sampled every  $T_s$  seconds gives a sequence  $x[n]$ . Which is equal to the continuous time signals at multiples of the sampling intervals  $T_s$ .

- $x_c(t) \Rightarrow x[n] = x_c(nT_s)$

In Fourier Transform domain we have a spectra of the continuous time signal  $X_c(j\Omega)$ . And at the output we have a discrete time Fourier Transform of the sequence  $X(j\omega)$ . What is that going to be in general? And how is it going to be related to the input spectrum?

- $X(j\Omega) \Rightarrow X(j\omega) = ?$

The key idea:

- pick  $T_s$  and set  $\Omega_N = \pi/T_s$
- pick  $\Omega_- < \Omega_N$

$$\begin{aligned}
 e^{j\Omega_0 t} &\rightarrow e^{j\Omega_0 T_s n} \\
 e^{j(\Omega_0 + 2\Omega_N)t} &\rightarrow e^{j(\Omega_0 + 2\Omega_N)T_s n}, \text{ add } 2\Omega_N \\
 e^{j(\Omega_0 + 2\Omega_N)t} &\rightarrow e^{j(\Omega_0 T_s n + 2\Omega_N T_s n)}, \text{ expand this product} \\
 e^{j(\Omega_0 + 2\Omega_N)t} &\rightarrow e^{j(\Omega_0 T_s n + \frac{2\pi}{T_s} T_s n)} \\
 e^{j(\Omega_0 + 2\Omega_N)t} &\rightarrow e^{j(\Omega_0 T_s n + 2)}, e^{j2} \text{ is equal to one} \\
 e^{j(\Omega_0 + 2\Omega_N)t} &\rightarrow e^{j\Omega_0 T_s n}, \text{ the same discrete time sequence as before}
 \end{aligned}$$

So we do not see the higher frequency complex exponential, it simply looks like the lower frequency exponential  $\Omega_0$ .

So in general, if we have two frequencies sampled, the higher frequency is aliased back onto the lower frequency and we simply see the sum of these two.

(a) Spectrum of raw-sampled signals

- start with the inverse Fourier Transform

$$x[n] = x_c(nT_s) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X_c(j\Omega) e^{j\Omega n T_s} d\Omega$$



- frequencies  $2\Omega_N$  apart will be aliased, so split the integration interval

$$x[n] = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \int_{(2k-1)\Omega_N}^{(2k+1)\Omega_N} X_c(j\Omega) e^{j\Omega MT_s} d\Omega$$

- with a change of variable and using  $e^{j(\Omega+2k\Omega_N)T_s M} = e^{j\Omega T_s M}$

$$x[n] = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \int_{-\Omega T_s M}^{\Omega T_s M} X_c(j(\Omega - 2k\Omega_N)) e^{j\Omega MT_s} d\Omega$$

- interchange summation and integral

$$x[n] = \frac{1}{2\pi} \int_{-\Omega T_s M}^{\Omega T_s M} \left[ \sum_{k=-\infty}^{\infty} X_c(j(\Omega - 2k\Omega_N)) \right] e^{j\Omega MT_s} d\Omega$$

- periodization of the spectrum; define

$$X_c(j\Omega) = \sum_{k=-\infty}^{\infty} X_c(j(\Omega - 2k\Omega_N))$$

- so that

$$x[n] = \frac{1}{2\pi} \int_{-\Omega T_s M}^{\Omega T_s M} X_c(j\Omega) e^{j\Omega MT_s} d\Omega$$

- set  $\omega = \Omega T_s$

$$\begin{aligned} x[n] &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{1}{T_s} X_c(j\frac{\omega}{T_s}) e^{j\omega M} d\omega \\ &= IDTFT \frac{1}{T_s} X_c(j\frac{\omega}{T_s}) \\ X(e^{j\omega}) &= \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X_c\left(j\frac{\omega}{T_s} - j\frac{2\pi k}{T_s}\right) \end{aligned}$$

$$X(e^{j\omega}) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X_c\left(j\frac{\omega}{T_s} - j\frac{2\pi k}{T_s}\right)$$

- (b) **TODO** Example: signal bandlimited to  $\Omega_0$  and  $\Omega_N > \Omega_0$
- (c) **TODO** Example: signal bandlimited to  $\Omega_0$  and  $\Omega_N = \Omega_0$
- (d) **TODO** Example: signal bandlimited to  $\Omega_0$  and  $\Omega_N < \Omega_0$

(e) **TODO** Example: non-bandlimited signal

#### 4. Sampling strategies

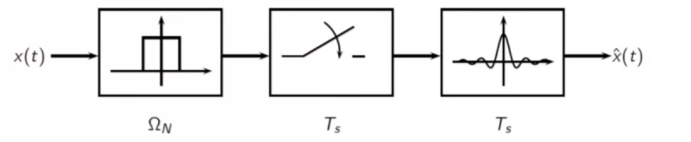
given a sampling period  $T_s$

- if the signal is bandlimited to  $\pi/T_s$  or less, raw sampling is fine i.e. equivalent to sinc sampling up to scaling factor  $T_s$ .
- if the signal is not bandlimited, two choices:
  - bandlimit via lowpass filter in the *continuous-time domain* before sampling i.e. sinc sampling
  - or raw sample the signal and incur aliasing
- aliasing sounds horrible, so usually we choose to bandlimit in continuous time

(a) Sinc Sampling and Interpolation

$$\hat{X}[n] = \left\langle \text{sinc} \left( \frac{t - nT_s}{T_s} \right), x(t) \right\rangle = (\text{sinc}_{T_s} * x)(nT_s)$$

$$\hat{X}[n] = \sum_n x[n] \text{sinc} \left( \frac{t - nT_s}{T_s} \right)$$



#### 1.1.5 Quantization

##### 1. Stochastic signal processing

(a) Terminology ( from W.Smith )

**Mean**

$$\mu = \frac{1}{N} \sum_{i=0}^{N-1} x_i = (x_0 + x_1 + x_2 + \dots + x_{N-1})/N$$

In electronics, the mean is commonly called the **DC** (direct current) value. Likewise, **AC**

(alternating current) refers to how the signal fluctuates around the mean value. For simple repetitive waveform, its excursion can be described by its peak-to-peak value. If the signal has a random nature, a more generalized method must be used.

### Standard Deviation

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=0}^{N-1} (x_i - \mu)^2} = \sqrt{(x_0 - \mu)^2 + (x_1 - \mu)^2 + \dots + (x_{N-1} - \mu)^2 / (N-1)}$$

$|x_i - \mu|$  describes how far the  $i^{th}$  sample **deviates** (differs) from the mean. The **average deviation** of a signal is found by summing the deviations of all the individual samples, and then dividing by the number of samples N. We take the absolute value of each deviation before summation; otherwise the positive and the negative terms would average to zero. The **standard deviation** is similar to the average deviation, except the averaging is done with power instead of amplitude. The standard deviation is a measure of how far the signal fluctuates from the mean.

### Variance

$$\sigma^2 = \frac{1}{N-1} \sum_{i=0}^{N-1} (x_i - \mu)^2$$

The variance represents the power of signal fluctuation from the mean.

**RMS Root Mean Square** The standard deviation measures only the AC portion of a signal, while rms value measures both the AC and DC components. If a signal has no DC component, its rms value is identical to its standard deviation.

### SNR Signal to Noise Ratio

$$snr = \frac{\text{mean}}{\text{standard deviation}} = \frac{\mu}{\sigma} = \frac{\frac{1}{N} \sum_{i=0}^{N-1} x_i}{\sqrt{\frac{1}{N-1} \sum_{i=0}^{N-1} (x_i - \mu)^2}}$$

### CV Coefficient Variation

$$CV = \frac{\text{standard deviation}}{\text{mean}} \times 100 = \frac{\sqrt{\frac{1}{N-1} \sum_{i=0}^{N-1} (x_i - \mu)^2}}{\frac{1}{N} \sum_{i=0}^{N-1} x_i} \times 100$$

- (b) **TODO** Deterministic vs. stochastic
- (c) A simple discrete-time random signal generator

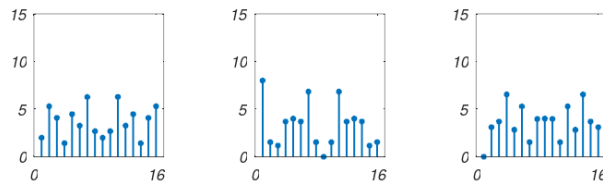
For each new sample, toss a fair coin:

$$x[n] = \begin{cases} +1 & \text{if the outcome of the n-th toss is head} \\ -1 & \text{if the outcome of the n-th toss is tail} \end{cases}$$

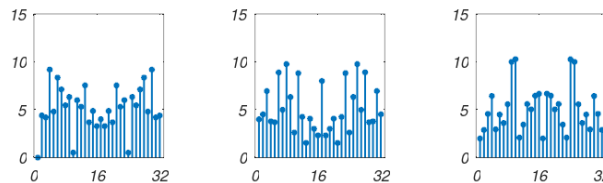
- each sample is independent from all others
- each sample value has 50% probability
- every time we turn on the generator we obtain a different *realization* of the signal
- we know the "mechanics" behind each instance
- but how can we analyze a random signal?

- (d) Spectral Properties

- let's try with the DFT of a finite set of random samples



- every time it's a different
- try with more data



- no clear pattern

- (e) Averaging

- when faced with random data an intuitive response is to take "averages"
- in probability theory the average is across realizations and it's called [Expectation](#)

- Expectation for the coin-toss signal

$$E[x[n]] = -1 \times P[\text{n-th toss is tail}] + 1 \times P[\text{n-th toss is head}] = 0$$

- (f) so the average value for each sample is zero....
- (g) as a consequence, averaging the DFT will not work
- (h)  $E[x[n]] = 0$
- (i) however the signal "moves", so its energy over power must be nonzero

## 2. **TODO** Averaging the DFT

## 3. Energy and power

- the coin-toss signal has infinite energy

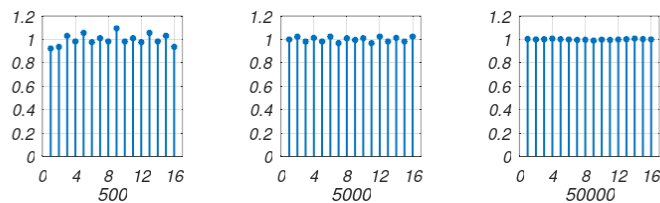
$$E_x = \sum_{k=-\infty}^{\infty} |x[n]|^2 = \lim_{N \rightarrow \infty} = \infty$$

- however it has finite power over any interval:

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

## 4. Averaging the DFT's square magnitude, normalized

- pick an interval length N
- pick an number of iterations M
- run the signal generator M times and obtain M N-point realizations
- compute the DFT of each realizations
- average their square magnitude divided by N



## 5. Power spectral density

$$P[k] = E \left[ |X_N[k]|^2 / N \right]$$

- it looks very much as if  $P[k] = 1$
- if  $|X_N[k]|^2$  tends to the *energy* distribution in frequency....
- ...  $|X_N[k]|^2 / N$  tends to the *power* distribution (aka **density**) in frequency



### PSD

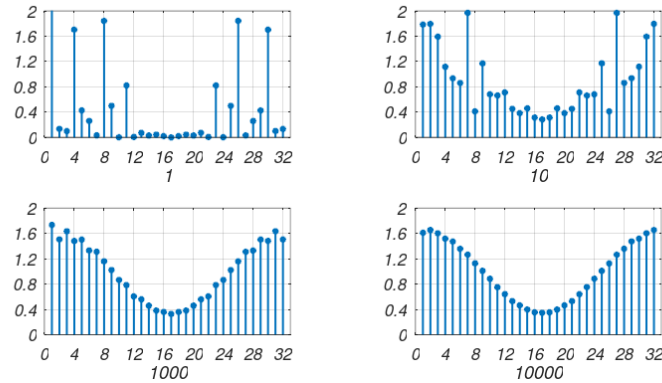
The frequency-domain representation for stochastic processes is the power spectral density:  $P[k] = \frac{1}{N} |X_N[k]|^2$

## 6. Power spectral density: Intuition

- $P[k] = 1$  means that the power is equally distributed over all frequencies
- i.e. we cannot predict the signal moves "slowly" or "super-fast"
- this is because each sample is independent of each other: we could have a realization of all ones or a realization in which the sign changes every other sample or anything in between.

## 7. Filtering a random process

- let's filter the process with a 2-point Moving Average filter
- $y[n] = (x[n] + x[n-1])/2$
- what is the power spectral density
- pick an interval length  $N$
- pick an number of iterations  $M$
- run the signal generator  $M$  times and obtain  $M$   $N$ -point realizations
- filter all  $M$ -realization
- compute the DFT of each filtered realizations
- average their square magnitude divided by  $N$

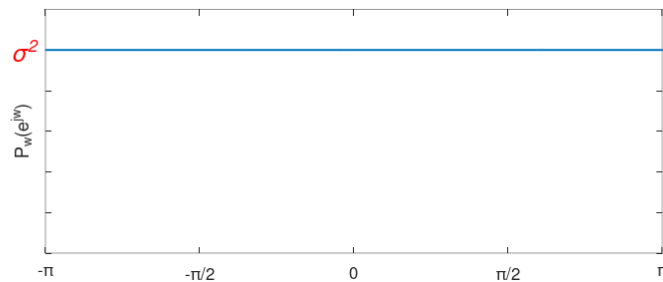


## 8. **TODO** Noise

### 9. White noise

- white indicates uncorrelated samples
- $r_w[n] = \sigma^2 \delta[n]$ : The autocorrelation is zero except at zero where it will take the value of the variance
- $P_w(e^{j\omega}) = \sigma^2$ : The power spectral density is the constant  $\sigma^2$  where  $\sigma$  is the variance of the stochastic signal.

Graphically the power spectral density of a white signal couldn't be any simpler.



- the PSD is independent of the probability distribution of the single samples (depends only on the variance)
- distribution is important to estimate bounds for the signal
- very often a Gaussian distribution models the experimental data the best
- **AWGN**: additive white Gaussian noise

## 10. Summary

- a stochastic process is characterized by its power spectral density (PSD)
- it can be shown (see text book) that the PSD is

$$P_x(e^{j\omega}) = DTFT r_x[n]$$

where

$$r_x[n] = E[x[k]x[n+k]]$$

is the [autocorrelation](#) of the process

- for a filtered stochastic process  $y[n] = \mathfrak{H}x[n]$ , it is:

$$P_x(e^{j\omega}) = |H(e^{j\omega})|^2 P_x(e^{j\omega})$$



### In Words

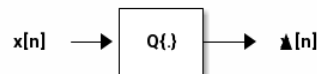


The power spectral density of the output is equal to the power spectral density of the input times the frequency response in magnitude square.

## Quantization

### 1. Quantization schemes

- digital devices can only deal with integers (b bits per sample)
- we need to map the range of a signal onto a finite set of values
- irreversible loss of information → [Quantization Noise](#)



Several factors at play:

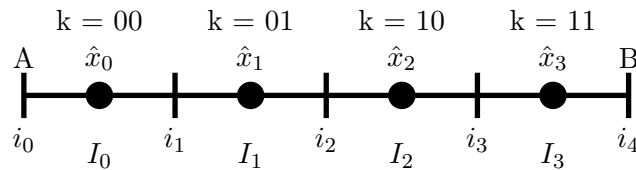


- storage budget (bits per sample)
- storage scheme (fixed point, floating point)
- properties of the input (input  $\in \mathbb{C} \rightarrow$  output  $\in \mathbb{N}$ )

## 2. Scalar quantization

The simplest quantizer:

- each sample is encoded individually (hence scalar)
- each sample is quantized independently (memoryless quantization)
- each sample is encoded using  $R$  bits



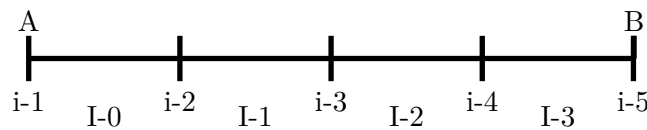
## 3. Quantization Error

$$e[n] = Q\{x[n]\} - x[n] = \hat{x} - x[n]$$

- model  $x[n]$  as a stochastic process
- model error as a white noise sequence
  - error samples are uncorrelated
  - all error samples have the same distribution
- we need statistics of the input to study the error

## 4. Uniform quantization

- simple but very general case
- range is split into  $2^R$  equal intervals of width  $\Delta = (B - A)2^{-R}$
- **With a Bit-Rate  $R$**  of 2 bits is a region split into 4 equally spaced intervals



- filters designed for deterministic signals work (in magnitude) in the stochastic case
- we lose the concept of phase since we don't know the shape of a realization in advance.