Contents

1 Week 7 Module 5:

1.1 Sampling and Quantization

Interpolation describes the process of building a continuous-time signal $\mathbf{x(t)}$ from a sequence of samples $\mathbf{x[n]}$. 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 $\mathbf{x(t)}$. 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 proove a version of the sampling theorem was Whittaker in England in 1915. Harry Nyquist at Bell Labs came up with the Nyqvist criterion, namely that a function that has a maximum frequency $F0F_{0F0}$ could be sampled at $2F02F_{02F0}$. 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 Continous-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 continous-time paradigm

   Two views of the world

   Table 1: Two views of the world 1

Digital World	Analog World
arithmetic	calculus
combinatorics	distributions
computer sience	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 $\Omega \in \mathbb{R}(rad/sec)$
DTFT: $\ell_2(\mathbb{Z}) \to L_2[-\pi, \pi]$	FT: $L_2(\mathbb{R}) \to L_2(\mathbb{R})$

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

3. Continuous-time signal processing

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

(a) Analog LTI filters

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

(b) Fourier analysis

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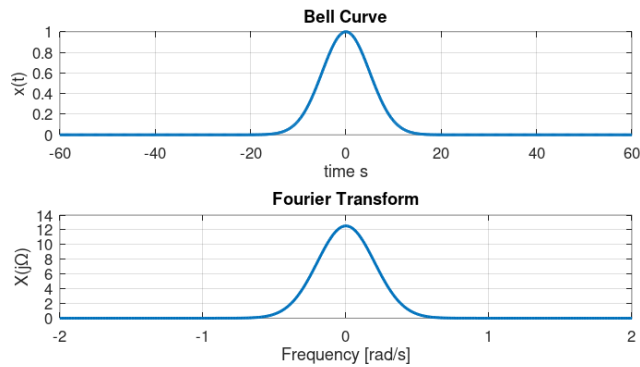- concept is the same:

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

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

(c) Real-world frequency

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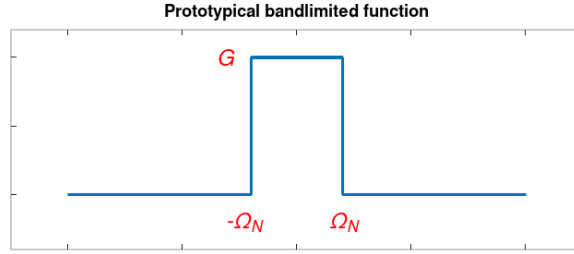- period $T = \dfrac{1}{F} = \dfrac{2\,\pi}{\Omega}$

(d) Example

4

(e) Convolution theorem

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

(f) Prototypical Bandlimited Functions

Prototypical bandlimited function

$$\Phi(j\,\Omega) = G\,rect(\frac{\Omega}{2\,\Omega_N})$$

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

$$\phi(t) = \frac{1}{2\,\pi}\int_{-\infty}^{\infty}\Phi(j\Omega)e^{j\Omega\,t}d\Omega$$
$$= G\frac{\Omega_N}{\pi}sinc(\frac{\Omega_N}{\pi}\,t)$$

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 2π in the Frequency Domain. Then the inverse continuous time Fourier Transform will have a maxmimum of 1 at the origin.

normalization $\qquad G = \dfrac{\pi}{\Omega_n}$

totoal bandwith $\qquad \Omega_B = 2\,\Omega_N$

define $\qquad T_s = \dfrac{2\,\pi}{\Omega_B} = \dfrac{\pi}{\Omega_N}$

This leads to the normalized prototypical bandlimted function:

5

> **Frequency Domain**
>
> $$\Phi(j\,\Omega) = \frac{\pi}{\Omega_N}\,rect\left(\frac{\Omega}{2\,\Omega_N}\right)$$

> **Time Domain**
>
> $$\phi(t) = sinc\left(\frac{t}{T_s}\right)$$

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

1.1.2 Interpolation

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

Gaps $\qquad\qquad$ 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)
   - $p(t) = a_0 + a_1 t + a_2 t^2 a... + a_{N-1} t^{(N-1)}$
   - "naive" approach

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

Without loss of generality:

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

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

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

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

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

The formula:

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

The Lagrange interpolation is the sought-after polynominal interpolation:

- polynominal 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   maximmally smooth (infinitely many continuous derivatives)

drawback   interpolation "bricks" depend on N

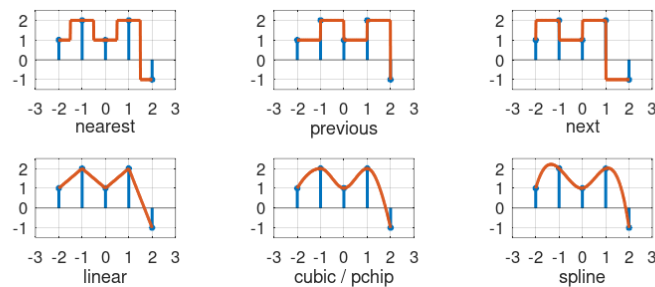5. Sinc interpolation formula

A remarkable result:

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

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

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

6. 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. The spectrum of interpolated signals

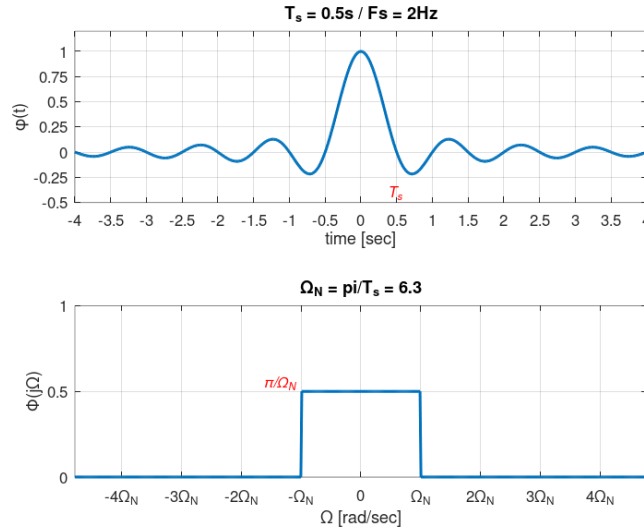   (a) Sinc interpolation the ingredients:

   - discrete-time signal $x[n]$, $n \in \mathbb{Z}$ (with DTFT $X(e^{j\omega})$)
   - interpolation interval T_s
   - the sinc function (properly scaled to have zero crossing at multiple of T_s the result
   - a smooth, contious-time signal $x(t)$, $t \in \mathbb{R}$

   What does the spectrum of x(t) look like?

   (b) Key Facts about the sinc

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

   $T_s = 0.5s$ / $Fs = 2Hz$

   $\Omega_N = pi/T_s = 6.3$

   (c) Sinc interpolation

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

   (d) Spectral representation (I)

9

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

$$= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x[n] sinc\left(\frac{t - nT_s}{T_s}\right) e^{-j\Omega t} dt$$

$$= \sum_{n=-\infty}^{\infty} x[n] \int_{-\infty}^{\infty} sinc\left(\boxed{\frac{t - nT_s}{T_s}}\right) e^{-j\Omega t} dt$$

$$= \sum_{n=-\infty}^{\infty} x[n] \left(\frac{\pi}{\Omega_N}\right) rect\left(\frac{\Omega}{2\Omega_N}\right) \boxed{e^{-jnT_s\Omega}}$$

(e) Spectral representation (II) Let's analyse the formula

$$X(j\Omega) = \sum_{n=-\infty}^{\infty} x[n] \left(\frac{\pi}{\Omega_N}\right) rect\left(\frac{\Omega}{2\Omega_N}\right) e^{-jnT_s\Omega}$$

$$= \left(\frac{\pi}{\Omega_N}\right) rect\left(\frac{\Omega}{2\Omega_N}\right) \sum_{n=-\infty}^{\infty} x[n]e^{-j(\pi/\Omega_N)\Omega\, n}$$

$$= \begin{cases} \left(\dfrac{\pi}{\Omega_N}\right) X(e^{j\pi(\Omega/\Omega_N)}) & |\Omega| \leq \Omega_N \\ 0 & otherwise \end{cases}$$

The DTFT is periodic and the periodic is

$$\sum_{n=-\infty}^{\infty} x[n]e^{-j(\pi/\Omega_N)\Omega\, n} = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}, \text{ with } \omega = 2 \cdot \Omega_N$$

> 📙 **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 Ω_N.

2. The space of bandlimited functions

   Claims:

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

   The space $\pi - BL$

   - is a 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
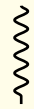
   > **Basis for $\pi - BL$**
   >
   > The sync function $sinc\left(\dfrac{t - nT_s}{T_s}\right)$ is an orthornormal basis for the $\pi - BL$ space.

   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

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

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

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

3. 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

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

Synthesis Formula

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

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

Corolllary

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 continous-time complex exponential

$$x(t) = e^{j\Omega_0 t}$$

- always periodic, period $T = \dfrac{2\pi}{\Omega_0}$
- all angualr speed are allowed
- $FT\left\{e^{j\Omega_0 t}\right\} = 2\pi\delta(\Omega - \Omega_0)$
- bandlimited to Ω_0

Table 3: Aliasing

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

2. Sinusoidal Aliasing

$$x(t) = cos(2\pi F_o t)$$
$$x[n] = x(nT_s) = cos(\omega_0 n)$$

with

$$F_s = \frac{1}{T_s}$$
$$\omega_o = 2\pi(\frac{F_0}{F_s})$$

(a) Aliasing: Sampling a Sinusoid

13
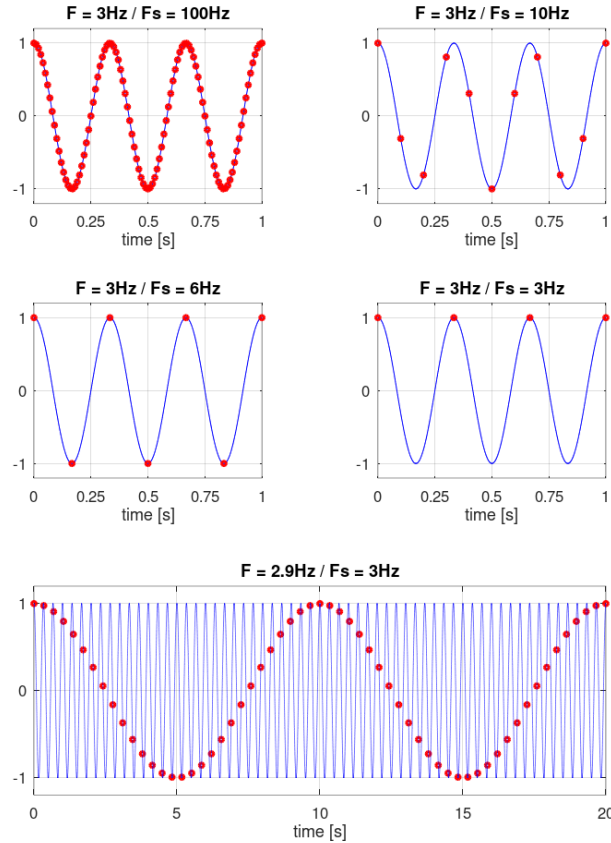
3. Aliasing for arbitrary spectra A contiuous time signal x_c sampled every T_s seconds gives a sequence x[n]. Which is equal to the contious time signals at multiples of the sampling intervals T_s.

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

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 genaral? 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 $\Omega_= < \Omega_N$

14

$$e^{j\Omega_0 t} \rightarrow e^{j\Omega_0 T_s n}$$

$$e^{j(\Omega_0 + 2\Omega_N)t} \rightarrow e^{j(\Omega_0 + 2\Omega_N)T_s n}, \text{ add } 2\Omega_N$$

$$e^{j(\Omega_0 + 2\Omega_N)t} \rightarrow e^{j(\Omega_0 T_s n + 2\Omega_N T_s n)}, \text{ expand this product}$$

$$e^{j(\Omega_0 + 2\Omega_N)t} \rightarrow e^{j(\Omega_0 T_s n + \frac{2\pi}{T_s} T_s n)}$$

$$e^{j(\Omega_0 + 2\Omega_N)t} \rightarrow e^{j(\Omega_0 T_s n + 2\pi n)}, \ e^{j2\pi n} \text{ is equal to one}$$

$$e^{j(\Omega_0 + 2\Omega_N)t} \rightarrow e^{j\Omega_0 T_s n}, \text{ the same discrete time sequence as before}$$

So we do not see the higher frequency complex exponential, it simply looks like the lower frequency exponential Ω_0.
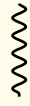
> **📙 Aliasing**
>
> So in general, if we have two frequencies sampled, the higher frequency is aliased back onto the lower frequency and we simply see the sum of these two.

(a) Spectrum of raw-sampled signals

- start with the inverse Fourier Transform

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

- 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

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

- so that

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

- set $\omega = \Omega T_s$

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{1}{T_s}\tilde{X}_c(j\frac{\omega}{T_s})e^{j\omega M}d\omega$$

$$= IDTFT\left\{\frac{1}{T_s}\tilde{X}_c(j\frac{\omega}{T_s})\right\}$$

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

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

(b) **TODO** Example: signal bandlimited to Ω_0 and $\Omega_N > \Omega_0$

(c) **TODO** Example: signal bandlimited to Ω_0 and $\Omega_N = \Omega_0$

(d) **TODO** Example: signal bandlimited to Ω_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 π/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 an incur aliasing
- aliasing sounds horrible, so usualle we choose to bandlimit in continuous time

(a) Sinc Sampling and Interpolation

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

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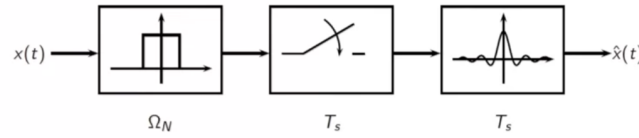
1.1.5 Quantization

1. Stochastic signal processing

   (a) Terminology (from W.Smith)

   Mean, Average

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

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

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

   $|x_i - \mu|$ describes how far the ith 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 termss 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

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

CV Coeffecent Variation

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

Probability Density Function PDF The probability density function is a measure of the likelihood of a particular value occuring in some function.

- PDF values are never negative $f(x) \geq 0$
- The sum of all the PDF values is one: $\int_{-\infty}^{\infty} f(x)dx = 1$
- Mean: $\mu_x = \int_{-\infty}^{\infty} x \cdot f(x)dx$

18

- Variance: $\mu_x^2 = \int_{-\infty}^{\infty} (x - \mu)^2 \cdot f(x) dx = \int_{-\infty}^{\infty} x^2 \cdot p(x) dx - \mu_x^2$

(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 independet 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
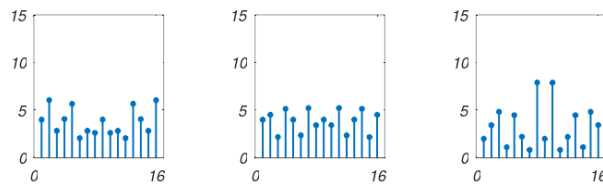
- every time it's a different
- try with more data
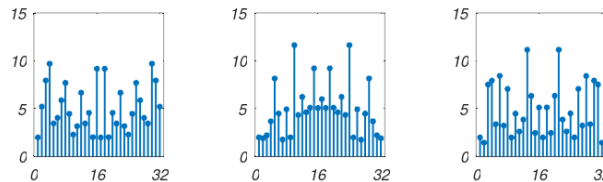
- no clear pattern

(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

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

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

(f) **TODO** Averaging the DFT

(g) Energy and power

- the coin-toss signal has infinite energy

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

- however it has finite power over any interval:

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

(h) 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

20
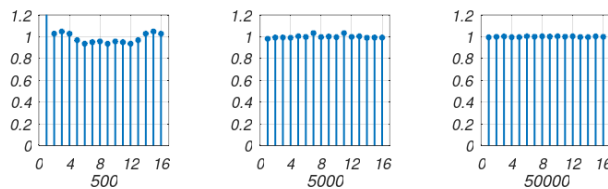
(i) Power spectral density

<div style="border:1px solid red;">

Power Spectral Density

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

</div>

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

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

(j) 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 "superfast"
- 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.

(k) Filtering a random process
- let's filter the random 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
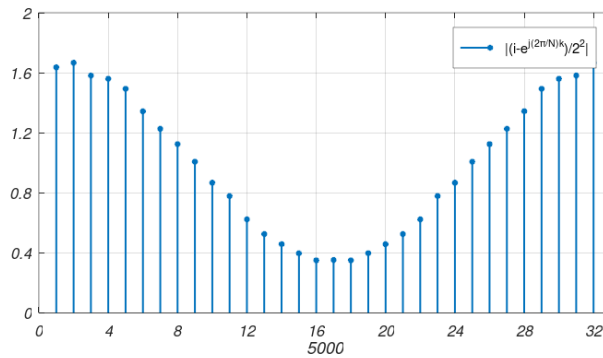- compute the DFT of each filtered realizations

- average their square magnitude divided by N
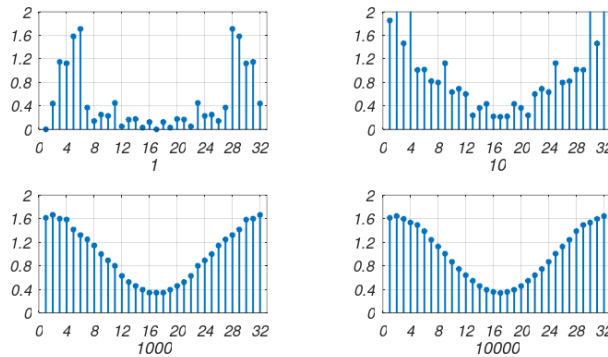
The frequency response of the moving average filter

$$H(e^{j\omega}) = \frac{1 - e^{j\omega}}{2}$$

wheraes the ploted shape is nothing but the square magnitude of the moving average filter evaluated at the DFT point $\dfrac{2\pi}{N}k$

(1) Stochaistic signal processing

- a stochastic process is characterized in frequency by its power spectral density (PSD)
- it can be shown (see text book) that the PSD is the DTFT of the autocorrelation of the process:

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

22

where each sample of the autocrrelation is obtained by taking the Expectation of the product of the stochastic signal times a delayed copie of itself:

$$r_x[n] = E\left[x[k]x[n+k]\right]$$

- for a filtered stochastic process $y[n] = H\{x[n]\}$ the general result is that the power spectral density (PSD) is equal to the PSD of the input times the frequency response in magnitude square.

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

General Result for a Stochaistic Process
The power spectral density of the output of a LTI system is equal to the power spectral density of the input times squared magnitude of the Fourier transform of the filter frequency response.

Key Results – filters designed for deterministic signals work (in magnitude) in the stochaistic case
- we lose the concept of phase since we don't know the shape of a realization in advance. All we know is the PSD in frequency

(m) Noise

Noise is everywhere • thermal noise
- sum of extraneous interferences
- quantization and numerical errors
- ...

we can model noise as a stochastic signal

the most important noise is white noise

(n) White noise
- white indicates uncorrelated samples
- $r_w[n] = \sigma^2\delta[n]$: The autocorrleation is zero except at zero where it will take the value of the variance

23

- $P_w(e^{j\omega} = \sigma^2$: The power spectral density is the constant σ^2 where σ is the variance of the stochastic signal.

Graphically the power spectral density of a white signal couldn't be any simpler.
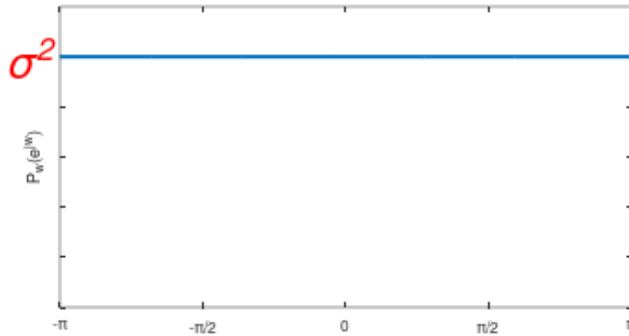
- the PSD is independent of the probability distribution of the **single** samples (depends only on the variance)
- distribution is important to estimate bounds for the signal
- very often a Gaussian distribution models the experimental data the best
- **AWN**: additive white Gaussian noise

> **Power Spectral Density**
> - The PSD of a random signal is the average squared magnitude of its Fourier transform. The average is taken across multiple realizations of the process.
> - The PSD is the Fourier transform of the autocorrelation function
> - The BSD of the output of a LEI system is the product of the BSD of the input and squared magnitude of the Fourier transform of the filter frequency response.

2. Quantization

   (a) 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

- irreverible loss of information $\longrightarrow$ Quantization Noise
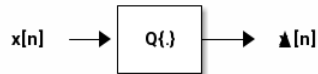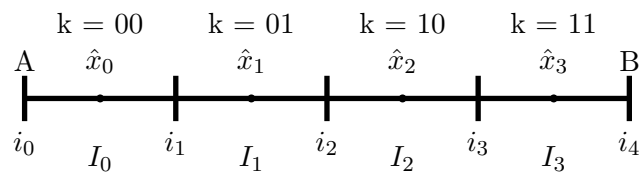

Several factors at play:

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

(b) 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

- what are the optimal interval boundries I_k ?
- what are the optimal quantization values $\hat{x}_k$?

(c) Quantization Error

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

- 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

i. Uniform quantization
- simple but very general case
- range is split into 2^R equal intervals of width $\Delta = (B - A)2^{-R}$

25

- $f_x(\tau)$: PDF of the input
- **With a Bit-Rate R** of 2 bits is a region split into 4 equally spaced intervals

A ┣━━━━╋━━━━╋━━━━╋━━━━┫ B

i-1   Δ   i-2   Δ   i-3   Δ   i-4   Δ   i-5
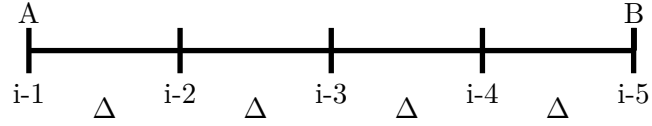
Mean Square Error is the variance of the error signal

$$\sigma_e^2 = E\left[|Q\{x[n]\} - x[n]|^2\right]$$

$$= \int_A^B f_x(\tau)(Q\{\tau\} - \tau)^2 d\tau$$

$$= \sum_{k=0}^{2^R-1} \int_{I_k} f_x(\tau)(\hat{x}_k - \tau)^2 d\tau$$
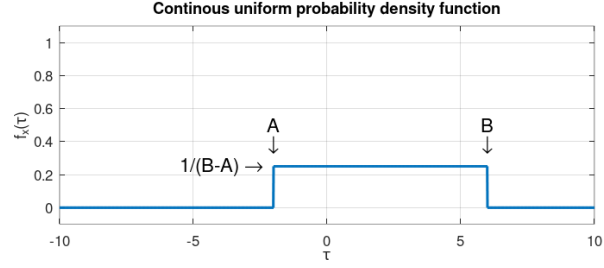
- $Q\{\tau\} - \tau$: Error function

error depends on the probability density of the input. The calculation is done in the following subsections here we already subsitutes the results.

$$\sigma_e^2 = \sum_{k=0}^{2^R-1} \int_{A+k\Delta}^{A+k\Delta+k} \frac{(A + k\Delta + \Delta/2 - \tau)^2}{B - A} d\tau$$

$$= 2^R \int_0^\Delta \frac{(\Delta/2 - \tau)^2}{B - A} d\tau$$

$$= \frac{\Delta^2}{12} \text{ with } \Delta = \frac{B - A}{2^R}$$

> **Quantization Error**
>
> $$\sigma_e^2 = \frac{\Delta^2}{12} \text{ with } \Delta = \frac{B - A}{2^R}$$

ii. Uniform quantization of uniform input

Continous uniform probability density function

iii. Uniform-input hypothesis

$$f_x(\tau) = \frac{1}{B - A}$$

$$\sigma^2 = \sum_{k=0}^{2^R - 1} \int_{I_k} \frac{(\hat{x} - \tau)^2}{B - A} d\tau$$

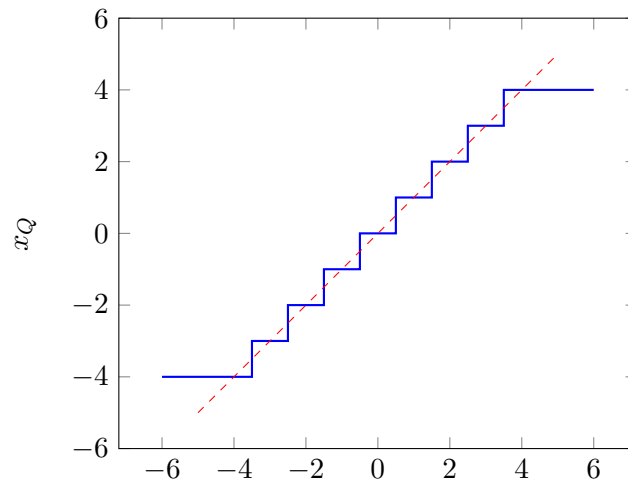iv. Find the optimal quantization point by minimazing the error

$$\frac{\partial \sigma_e^2}{\partial \hat{x}_m} = \frac{\partial}{\partial \hat{x}_m} \sum_{k=0}^{2^R - 1} \int_{I_k} \frac{(\hat{x}_k - \tau)^2}{B - A} d\tau$$

$$= \int_{I_m} \frac{2(\hat{x}_m - \tau)^2}{B - A} d\tau$$

$$= \frac{2(\hat{x}_m - \tau)^2}{B - A} \Big|_{A+m\Delta}^{A+m\Delta+\Delta}$$

Minimizing the error:

$$\frac{\partial \sigma_e^2}{\partial \hat{x}_m} = 0 \text{ for } \hat{x}_m = A + m\Delta + \frac{\Delta}{2}$$

optimal quantization point is the interval's midpoint, for all intervals

v. **TODO** Correct this PGF Plot Quantization Characteristic

27

The quantiziser associates each quantization interval to its midpoint.

(d) Error Analysis

Error Energy

$$\sigma_e^2 = \frac{\Delta^2}{12} \text{ with } \Delta = \frac{B-A}{2^R}$$

Signal Energy

$$\sigma_x^2 = \frac{(B-A)^2}{12}$$

Signal to Noise Ratio

$$SNR = 2^{2R}$$

Signal to Noise Ratio in db

$$SNR_{db} = 10 \cdot log_{10} 2^{2R} \approx 6R \ db$$

3. The 6db/bit rule of thumb

   - a compact disk has 16 bits/sample:

$$maxSNR_{db} =\approx 6R\ db = 6 \cdot 16\ db = 96\ db$$

- a DVD has 24 bits/sample:

$$maxSNR_{db} =\approx 6R\ db = 6 \cdot 24\ db = 144\ db$$

1.1.6 Notes and External Resources

1. **TODO** Clipping, saturation and interpolation

2. Practical interpolation and sampling

   (a) Time Domain to Discrete

   ideally   x(t) $\Rightarrow$ x[n]   in practice

   $$x(t) = \sum_{n=-\infty}^{\infty} x[n]sinc\,(fract - nT_sT_s)$$

   $$X(j\Omega) = \frac{\pi}{\Omega_N}X(e^{j\pi\Omega/\Omega_N})rect\left(\frac{\Omega}{2\Omega_N}\right)$$

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

   $$X(j\Omega) = \frac{\pi}{\Omega_N}I\left(j\pi\frac{\Omega}{\Omega_N}\right)X(e^{j\pi\frac{\Omega}{\Omega_N}})$$

   (b) Discrete to Time Domain

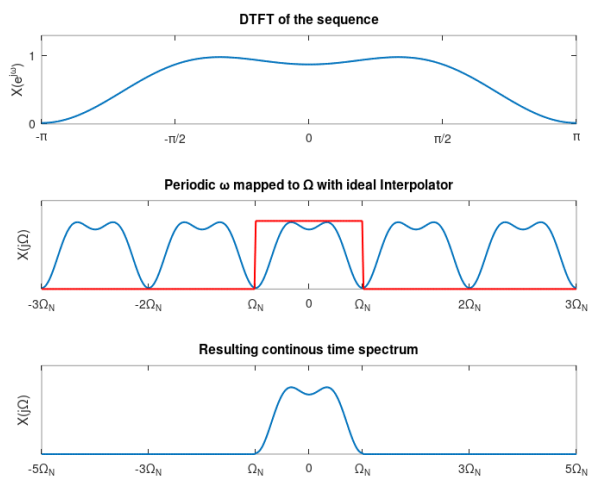   ideally   x[n] $\Rightarrow$ x(t)   in practice

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

   $$X(j\Omega) = \frac{\pi}{\Omega_N}X(e^{j\pi\Omega/\Omega_n}rect\left(\frac{\Omega}{2\Omega_N}\right))$$

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

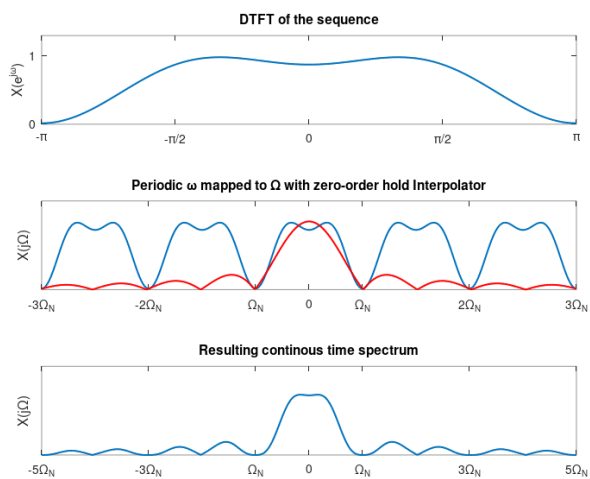 $$X(j\Omega) = \frac{\pi}{\Omega_N}I\left(j\pi\frac{\Omega}{\Omega_N}\right)X(e^{j\pi\frac{\Omega}{\Omega_N}})$$

   i. Ideal Interpolator
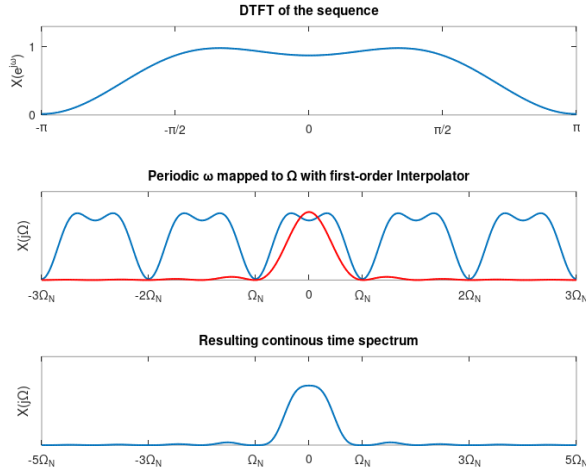   - Frequency Domain: Rect
      - Time Domain: sinc

DTFT of the sequence

Periodic ω mapped to Ω with ideal Interpolator

Resulting continous time spectrum

ii. Zero-Order Hold Interpolator

- Frequency Domain: sinc
- Time Domain: rect

DTFT of the sequence

Periodic ω mapped to Ω with zero-order hold Interpolator

Resulting continous time spectrum

iii. First-Order Interpolator

- Frequency Domain: $sinc^2$
- Time Domain: Triangle

DTFT of the sequence

Periodic ω mapped to Ω with first-order Interpolator

Resulting continous time spectrum

3. **TODO** Bandbass sampling

4. Multirate Signal Processing Book: Multirate Filtering for Digital Signal Processing Matlab Applications

   (a) Upsampling

   $$x_{NU}[n] = \begin{cases} x[k] \text{ for } n = kN, \, k \in \mathbb{Z} \\ 0 \text{ otherwise} \end{cases}$$

   $$x[n] \longrightarrow \boxed{N \uparrow} \longrightarrow x_{NU}[n]$$

   We create an upsampled sequence from any discrete time sequence simply by building a sequence where we output one sample of the original sequence followed by captial N minus 1 samples.
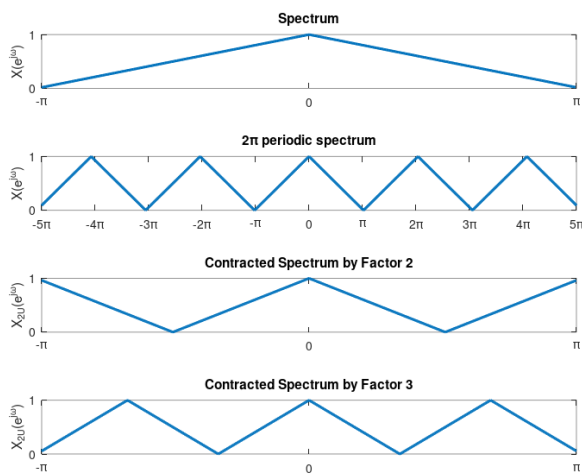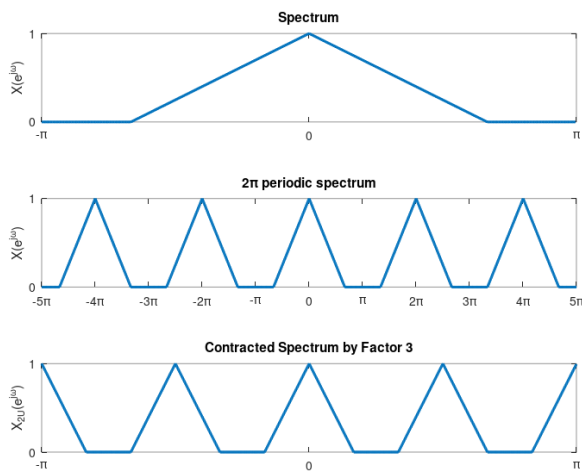
31
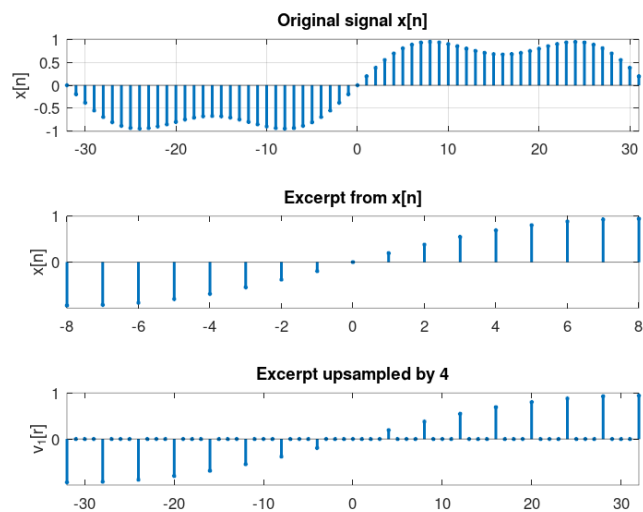
i. Spectral Representation

> **Spectral Representation**
>
> $$X_{NU}(e^{j\omega}) = X(e^{j\omega N})$$

The spectrum of the upsampled sequence is equal to the spectrum of the original sequence contracted by a factor N.

ii. Non fullband spectrum

iii. What we don't like

- in the time domain: zeors between nonzero samples are not "natural"
- in the frequency domain: extra replicas of the spectrum; can we get rid of them?
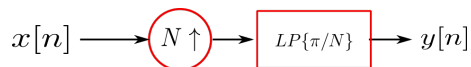 We need to interpolate the signals in the Frequency domain:
  - zero-order hold interpolator
  - first-order interpolator

iv. The ideal digital interpolator

$$H(e^{j\omega}) = rect\left(frac\omega N2\pi\right)$$
$$h[n] = \frac{1}{N}sinc\left(\frac{n}{N}\right)$$

This can not be achieved but as good as possible add an Low Pass with cut off frequency $\dfrac{pi}{N}$,

$$x[n] \longrightarrow \boxed{N\uparrow} \longrightarrow \boxed{LP\{\pi/N\}} \longrightarrow y[n]$$
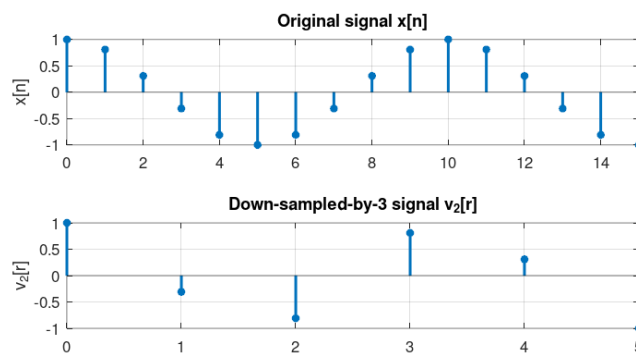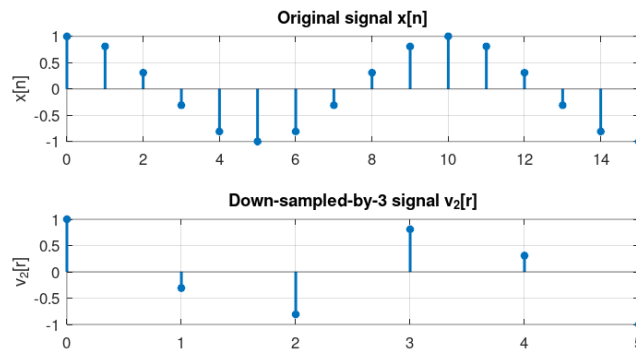
with a discrete time ideal low pass filter with cut off frequency π/N

(b) Downsampling

$$x_{ND}[n] = x[nN]$$

Every $N - 1$ input sample is discareded resp. only every N-th sample is used.

$$x[n] \longrightarrow \boxed{N \downarrow} \longrightarrow x_{ND}[n]$$

Original signal x[n]

Down-sampled-by-3 signal v₂[r]

[[

Original signal x[n]

Down-sampled-by-3 signal v₂[r]
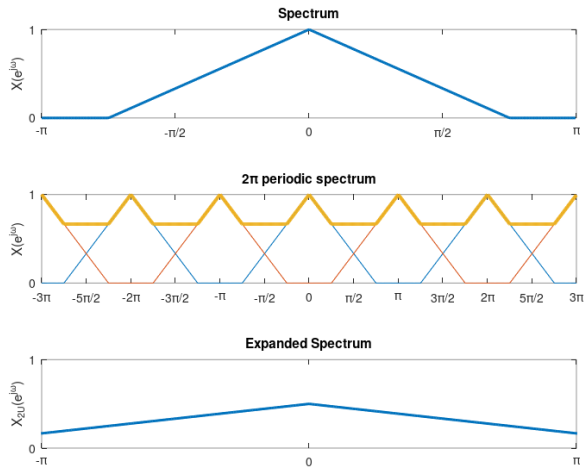
]

i. Spectral Representation

> **Spectral Representation**
>
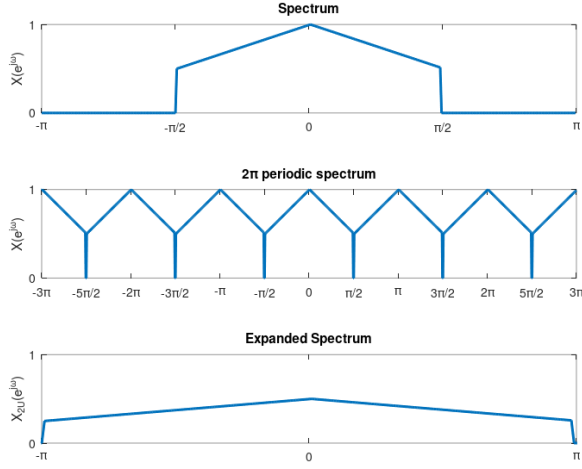> $$X_{ND}(e^{j\omega}) = \frac{1}{N} \sum_{m=0}^{N-1} X(e^{j\frac{\omega - 2\pi m}{N}})$$

34

There's 1/N normalization factor in front, that multiplies the sum of capital N copies of the original spectrum, where each copy has been shifted by a multiple of 2 pi/N, and where the frequency axis has been stretched out by a factor of capital N. So the interval -pi/N, pi/N becomes the interval -pi, pi.
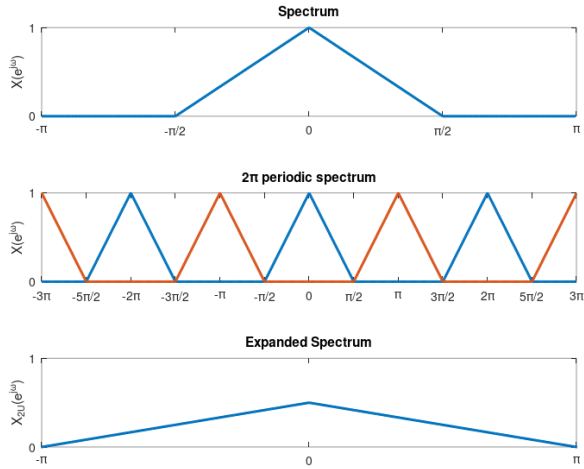

ii. Downsampling by 2 with aliasing

iii. Downsampling by 2 with antialiasing filter

Spectrum

$X(e^{j\omega})$

1

0

-π   -π/2   0   π/2   π

2π periodic spectrum

$X(e^{j\omega})$

1

0

-3π   -5π/2   -2π   -3π/2   -π   -π/2   0   π/2   π   3π/2   2π   5π/2   3π

Expanded Spectrum

$X_{2U}(e^{j\omega})$

1

0

-π   0   π

To avoid alisasing we have to filter the input sequence with a low-pass filter with cutoff frequency π/N, where N is the downsampling factor that will follow.

(c) Sampling Rate Change

Downsampling and upsampling change the implicit sampling rate by integer factor. But we can combine them so that the resulting sampling rate change will be an aribtrary fractional number. The combination always goes in the sense that we first upsample the signal, which is the operation that does not change the information content of the original sequence, and then we perfrom downsampling.

$$x[n] \longrightarrow \boxed{N\uparrow} \longrightarrow \boxed{LP\{min/\pi/N, \pi/M)\}} \longrightarrow \boxed{M\downarrow} \longrightarrow y[n]$$
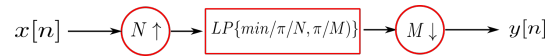
Example:

- CD: $F_s = 44100Hz$
- DVD: $F_s = 48000Hz$
- $\dfrac{N}{M} = \dfrac{160}{147}$
- in practice,we use time-varying local interpolation

5. **TODO** FIR-based sampling rate conversion

- Sampling rate change through time-varying local interpolation

(a) Subsample interpolation

36

(b) 2nd-order Lagrange interpolation polynomials

(c) Lagrange interpolation (N=1)

(d) CD to DVD revisted

6. **TODO** Analog-to-digital and digital-to-analog converters

7. **TODO** Oversampling

- Oversampled A/D -> reduce quantization error
- Oversampled D/A -> use cheaper hardware for interpolation