BOOCs EPFL

Digital Signal Processing
Paolo Prandoni
and Martin Vetterli

DIGITAL SIGNAL PROCESSING

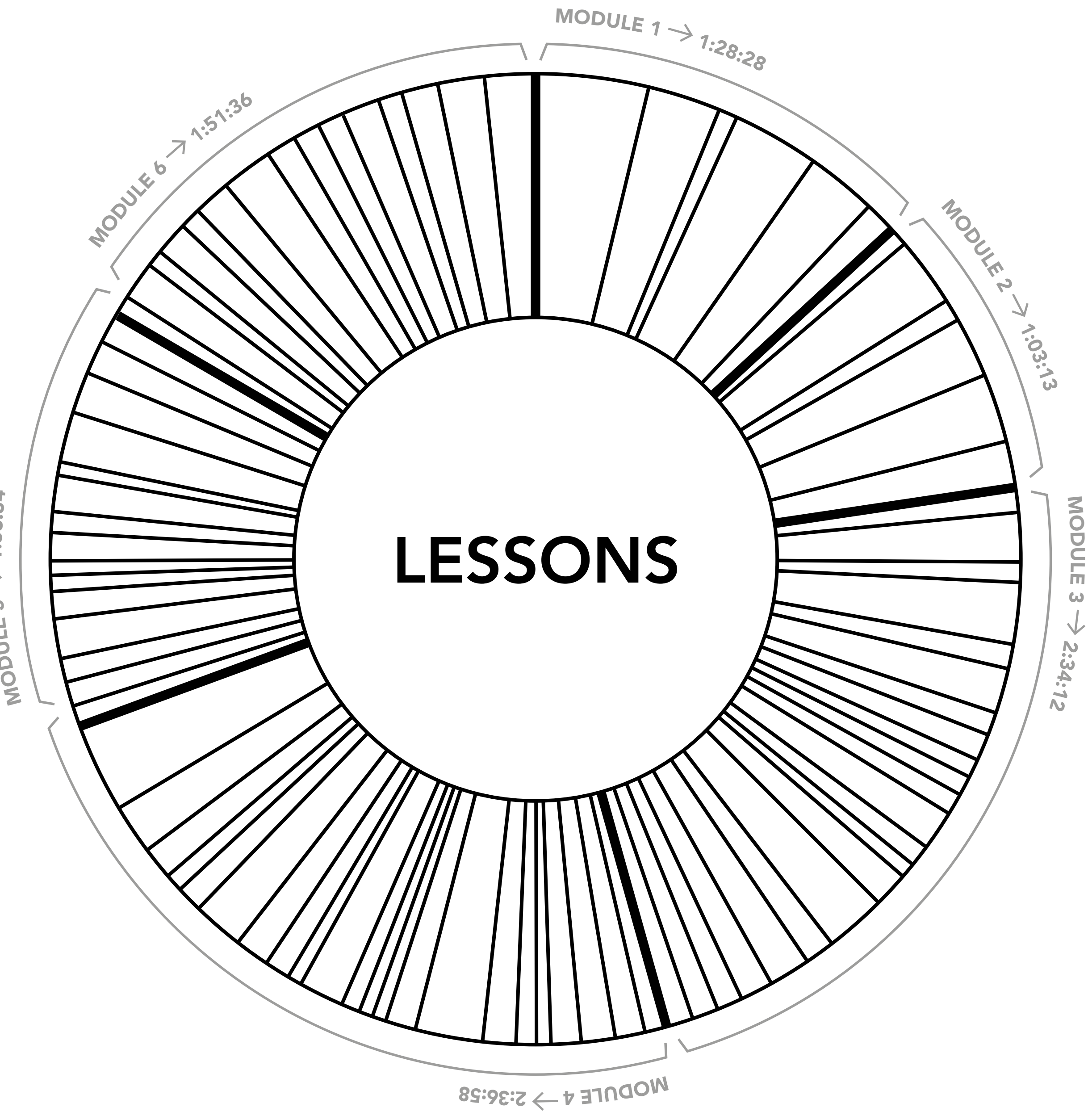
Paolo Prandoni
and Martin Vetterli

DOWNLOAD THE EBOOK

LEARN FASTER, LEARN BETTER!

BOOCs EPFL

Digital Signal Processing
Paolo Prandoni and Martin Vetterli



CONTENT

MODULE 4

MODULE 5

MODULE 6

1.1 INTRODUCTION TO DIGITAL SIGNAL PROCESSING

Let us first define the main concepts of this course. A signal is a description of the evolution of a physical phenomenon. Processing is where we make sense of the information that has been described by the signal. There are two ways we can process a signal:

- We can analyze it: understand the information carried by the signal and perhaps extract a more high level description.
- We can synthesize it: create a physical phenomenon that contains a certain amount of information that we want to put out in the world.

The digital paradigm is composed of two fundamental ingredients: discrete time and discrete amplitude.

Discrete models are extremely easy to use computationally speaking. Mathematically these are mappings from a set of integers to a set of values, *V*, which could be a set of real numbers. We will indicate a discrete-time signal as $x[n]$. The index n does not have a physical dimension, it is just an ordinal number that orders the samples one after another.

The continuous-time representation and the discrete-time representation are equivalent. Mathematically, the result is known as a sampling theorem, and it has a very simple statement:

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

which we will study in more detail in later classes.

Discrete amplitudes have very important consequences in three domains, as it is easier to deal with a finite set of values:

- storage
- processing
- transmission

1.2 DISCRETE TIME SIGNALS

Now we will consider discrete time signals and operators. First, we will define what discrete time signals are. A discrete time signal is a sequence of complex numbers. For now, most of the signals we will study will have one dimension, which typically will be time. Notation is particularly important: indices of a signal named x are integers such that the nth term of the signal will be denoted as $x[n]$. Note that the indices are dimensionless; the analysis of the signal implies the periodicity of measurements, also known as sampling. Typically, we have two-sided sequences x: $\mathbb{Z} \to \mathbb{C}$ meaning they could go from –infinity to +infinity.

Among the formal signals used in this class, there is the delta signal, $x[n] = \partial[n]$ (fig. 1). This is the simplest signal you will encounter in this class, it is equal to 1 whenever n = 0 and 0 otherwise.

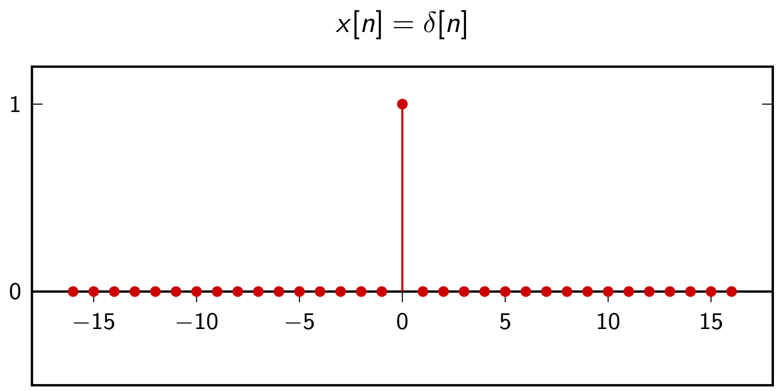


FIGURE 1

5:15 16:29

The delta signal

The next simple signal you will encounter is the unit step, $x[n] = u[n]$, which is a sequence equal to 0 from $-\infty$ until the origin and then flips to 1 at the origin to $+\infty$.

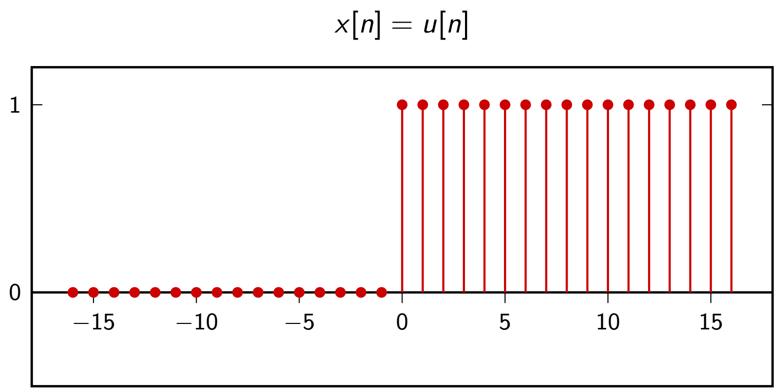


FIGURE 2

6:26 16:29

The unit step signal.

The third signal we will study in this lecture is exponential decay. $x[n] = |a|^n u[n]$ where $|a| < 1$, which is a combination of the unit step, meaning it is equal to 0 before the origin and then is positive. At that point, we apply an exponential where the root is smaller than 1. As it is taken to the nth power, it comes down as an exponential.

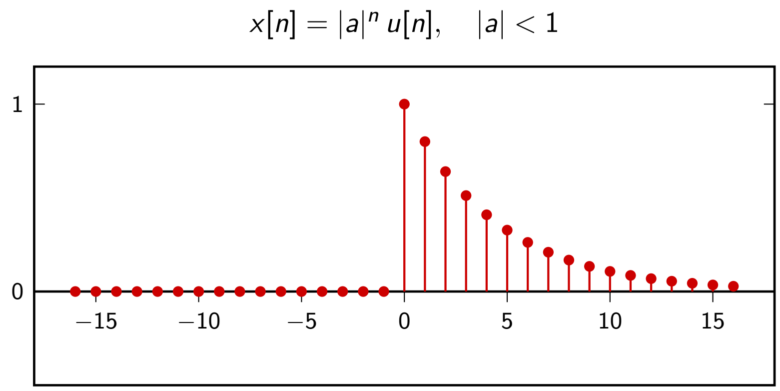


FIGURE 3

6:55 16:29

The exponential decay signal

Nowadays, most of the signals we encounter in real life are oscillations, a combination of sinusoids. A sinusoid is a signal such that $x[n] = \sin(\omega_0 n + ø)$.

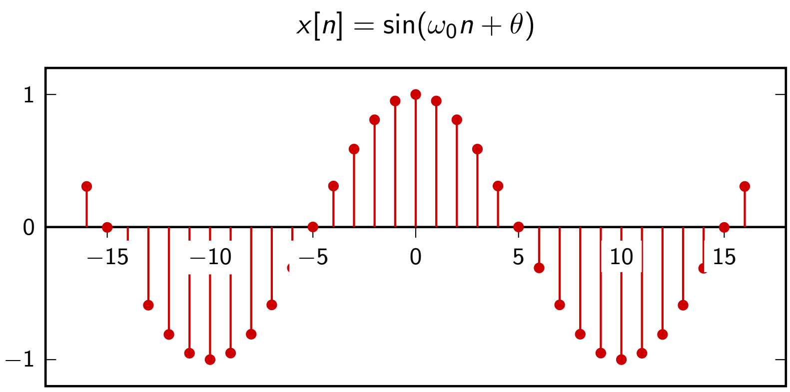


FIGURE 4

13:11 16:29

The sinusoid

There are four classes of signals and each is used in a different case:

– The finite length signal: can either be written as a sequence notation, $x[n]$ where $n = 0, \dots, N-1$, or as a vector notation $x = [x_0\ x_1 \dots x_{N-1}]^T$. They are very useful for numerical entities (e.g., Matlab).
– The infinite length signal: can also be written as a sequence notation, $x[n]$ where $n \in \mathbb{Z}$. Very good for mathematical abstraction and theorems.
– The periodic signal: for an N-periodic sequence $\tilde{x}[n] = \tilde{x}[n + kN]$ where $n, k, N \in \mathbb{Z}$ and one of these periods has all the information needed. It is thus a bridge between finite and infinite length sequences.
– The finite support sequence: $\bar{x}[n] = x[n]$ for $0 \leq n \leq N-1$ and 0 otherwise it has the same information as a finite length signal of length N. It is another possible bridge between finite and infinite length signals.

Let's now look at the elementary operators that we can apply to these elementary signals:

– scaling: $y[n] = \alpha x[n]$, where α is a scalar
– sum: $y[n] = x[n] + z[n]$
– product: $y[n] = x[n] \cdot z[n]$
– shift by k (delay): $y[n] = x[n-k]$

Two conceptual characteristics of a signal are its power and energy. The energy is the sum of the squares of the samples. It may or may not be finite. The notion of power is simply the energy of one period divided by the length of the period.

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

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

For periodic signals, energy will be infinite and power will be the same formula as stated above, but applied to a single period.

$$E_{\tilde{x}} = \infty$$

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

1.3.a HOW DOES YOUR PC PLAY DISCRETE-TIME SOUND?

A computer has interfaces that allow you to visualize the signals that you create by writing a few lines of code. You can visualize them as plots on the screen, and more importantly, you can materialize the signals as audio signals that you can hear. However, an interface is needed, as our ears are analog devices and the PC is a digital device.

In discrete time, *n* has no physical dimension, and periodicity is determined by how many samples we have to observe before the pattern repeats. In the physical world, periodicity is measured in how many seconds we have to wait before the pattern repeats, and frequency is measured in Hertz (s^{-1}). Figure 1 shows how the PC bridges this gap with a sound card.

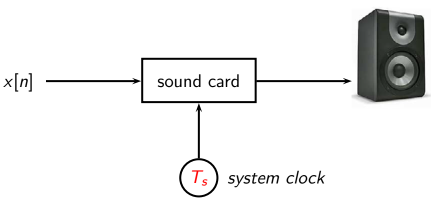


FIGURE 1

2:05 3:33

How your PC plays sound

T_s is the time we wait before we take a new sample from a discrete time sequence and feed it into the sound card. Hence a periodicity of *M* samples becomes a periodicity of MT_s seconds, so that the real-world frequency is thus: $f = 1/MT_s$ Hz.

1.3.b THE KARPLUS-STRONG ALGORITHM

This is an example of what signal processing can be applied to. We will see that there are some simple primitives that can be used to build signal processing devices. We will first consider digital signal processing as Lego blocks. Lego blocks have various colors and shapes but they all fit together. A block diagram, as you can see in figure 1, takes a sequence $x[n]$ as an input and outputs another sequence $y[n]$ through multipliers, adders, or delays.

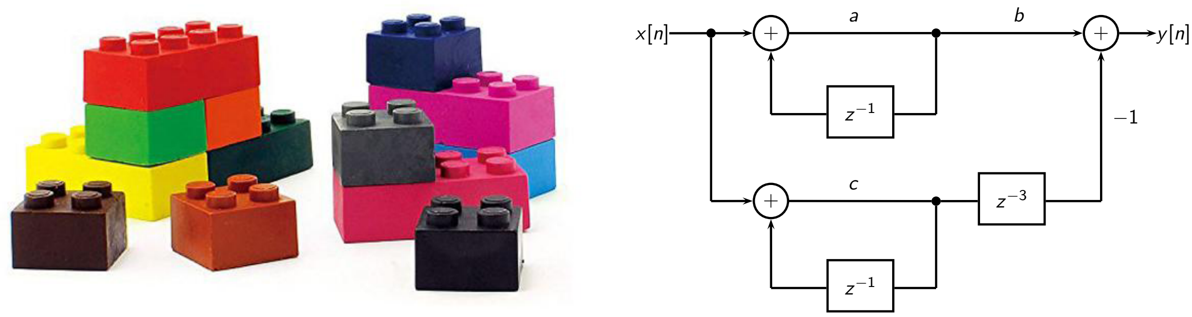


FIGURE 1

0:10 19:42

Lego versus block diagram

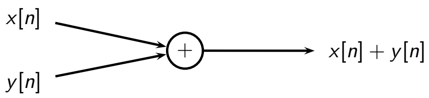


FIGURE 2

0:40 19:42

Adder

An adder takes two signals and outputs their sum (fig. 2).

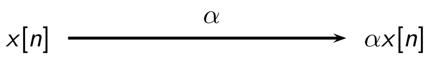


FIGURE 3

1:03 19:42

Multiplier

A multiplier is denoted by a variable placed next to an arrow, so that x gets multiplied by this variable (fig. 3).

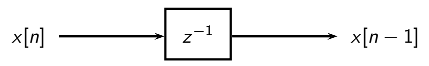


FIGURE 4

1:21 19:42

Delay

The delay is denoted by z^{-N} and the output of a delay is the same sequence shifted by an integer N to the right (fig. 4).

Let's study an operator: the moving average. The 2-point moving average is a simple average:

$$m = \frac{a+b}{2}$$

Its output is thus a "local" average:

$$y[n] = \frac{x[n-1] + x[n]}{2}$$

We can now use blocks to build a diagram for the 2-point moving average in figure 5.

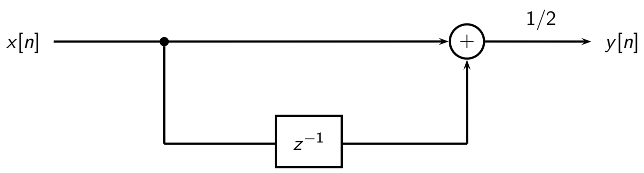


FIGURE 5

4:00 19:42

Moving average into diagram blocks

In figure 6, we can observe that the arrow of the previous block diagram has been reversed. That leads to a very different type of output. A single input other than 0 will, in general, generate a infinite output.

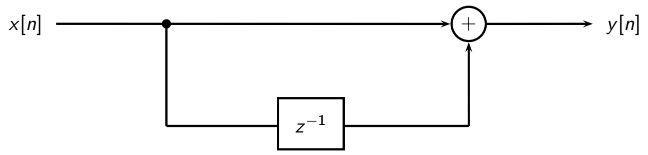


FIGURE 6

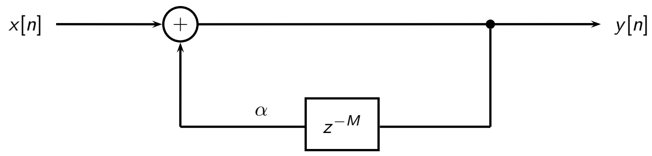
7:20 19:42

Reversed loop of moving average block diagram

In figure 7, we can see a generalized version of that reversed loop. If we fix α to 1, then we can observe that a delay of M implies an M-sample periodicity, which could be very useful in playing sounds. Going further into that generalized version:

– M controls the frequency (pitch).
– α controls the envelope (decay whenever $\alpha < 1$).

The Karplus-Strong algorithm is a range invented to simulate guitar sounds. It is initialized with a sequence of random numbers. We apply the generalized version of the reversed loop to it, and we can listen to the output that sounds surprisingly like a guitar. We encourage you to listen to it on the video (15:50/19:42).



$$y[n] = \alpha\, y[n-M] + x[n]$$

FIGURE 7

11:30 19:42

Generalized version of a reversed loop

GOETHE'S TEMPERATURE MEASUREMENT

Signal of the Day

The aim of the "signal of the day" video type is to show you some real-life applications of signal processing, which will be related to the theory taught in other videos. This video consists in investigating the longest recorded signal in daily measurements: a series of daily mean temperatures recorded in Jena (Germany). For convenience, we computed the annual mean daily temperature. We then applied a moving average over this obtained signal with a window of 25 years: we can observe that the resulting function increases over time which provides a perfect illustration of the effect of global warming.

1.4 COMPLEX EXPONENTIALS

$$e^{j\alpha} = \cos\alpha + j\sin\alpha$$

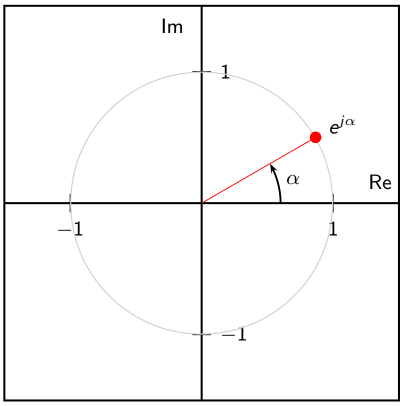


FIGURE 1

4:10 14:45

Spatial representation of a complex exponential

Most elementary and fundamental discrete time signals are oscillations. For an oscillation to be mathematically defined, we need to have certain information such as:

– a frequency ω (*omega*) (units in radians)
– an initial phase ø (units in radians)
– an amplitude A (units depending on underlying measurement)
– a trigonometric function

Complex numbers are essential for simplifying some complex expressions. A trigonometric expression can be reduced to sines and cosines using Euler's formula:

$$x[n] = Ae^{j(\omega n + ø)} = A[\cos(\omega n + ø) + j\sin(\omega n + ø)]$$

Mathematically, it is simpler to use a complex exponential than sines and cosines. The complex exponential relies on the unit circle whenever $A = 1$, as in figure 1.

rotation: $\mathbf{z}' = \mathbf{z}\, e^{j\alpha}$

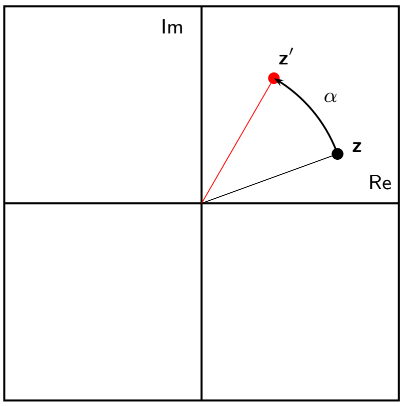


FIGURE 2

4:50 14:45

Rotation on the complex plane

Let z be a point on the complex plane, and we want to rotate it by α. We can simply multiply it by a complex number z' of argument α.

An example of the application of this complex exponential would be the complex exponential machine: $x[n] = e^{jwn}\, x[n+1] = e^{jwn}x[n]$. This kind of application leads to an illustration of aliasing that we can observe in the wagon wheel effect: a wheel going backwards in our perception instead of going forward. This is because the chosen frequency is too large: if the frequency is between π and 2π, we are most likely to think of it as a negative frequency. This is where the main differences between discrete time (no physical dimension, how many samples before pattern repeats) and what we call "real world" (frequency measured in Hz and notion of time between the repetition of patterns) appear. We have to find a way to relate them to one another, and this will be done by interpolation: let T_s be the time in seconds between two samples, and M be the periodicity, the "real-world" periodicity will thus be $f = 1/MT_s$ [Hz].

2.1 SIGNAL PROCESSING AND VECTOR SPACES

As linear algebra is a prerequisite for this course, it is important that you understand how a matrix-vector multiplication is carried out:

$$\begin{bmatrix} a_{0,0} & a_{0,1} & \dots & a_{0,N} \\ a_{1,0} & a_{1,1} & \dots & a_{1,N} \\ \vdots & \vdots & \vdots & \vdots \\ a_{M-1,0} & a_{M-1,1} & \dots & a_{M-1,N} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{M-1} \end{bmatrix} = \mathbf{A}\mathbf{x}$$

As we will be dealing with vectors, an important rule is the vector addition rule that you can observe in figure 1.

$$\mathbf{x} + \mathbf{y} = \begin{bmatrix} x_0 + y_0 & x_1 + y_1 \end{bmatrix}^T$$

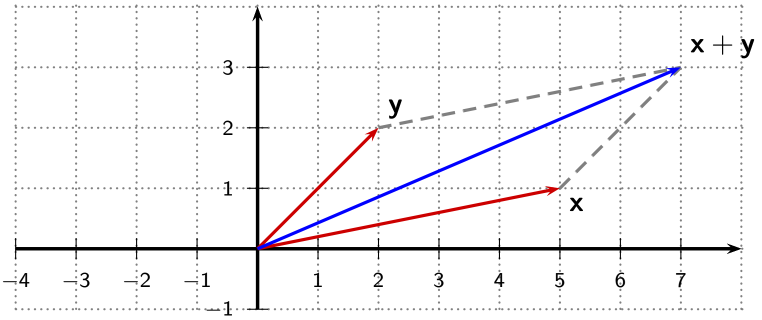


FIGURE 1

0:30 4:03

Parallelogram rule for vector addition, prerequisite

As we have four types of signals, it would very complicated to specialize the theory to each of the cases explained, so we need a common framework: vector space. It is very convenient as it provides the same framework for different classes of signals and also for continuous-time signals. Vector spaces are very general objects; they are defined by their properties (not by the shape of the vectors they contain), and once these are satisfied, we can use all the tools of the space.

2.2.a VECTOR SPACE

You should be familiar with some spaces such as:
– $\mathbb{R}^2$, $\mathbb{R}^3$: Euclidean space
– $\mathbb{R}^N$, $\mathbb{C}^N$: extension of Euclidean space

Here are two spaces that you might not know:
– $l_2(\mathbb{Z})$: space of square-summable infinite sequences
– $L_2([a,b])$: space of square-integrable functions over an interval

Here, the key point is that vectors can be arbitrarily complex entities, such as functions. And this will be very useful in unifying our approach to signal processing.

Note that some spaces can be represented graphically ($\mathbb{R}^2$, $\mathbb{R}^3$) and others cannot ($\mathbb{R}^N$ for $\mathbb{N} > 3$). What all these vectors spaces have in common is that they obey a set of axioms that define the properties of the vectors and what we can do with these vectors.

For $x, y, z \in \mathbb{V}$ and $\alpha, \beta \in \mathbb{C}$:

- $x + y = y + x$ (Commutativity)
- $(x + y) + z = x + (y + z)$ (Distributivity)
- $\alpha(x + y) = \alpha x + \alpha y$ (Distributive scalar multiplication w.r.t. sum of vectors)
- $(\alpha+\beta)x = \alpha x+\beta x$ (distributive scalar multiplication w.r.t. sum of scalars)
- $\alpha(\beta x) = \alpha\beta x$ (associative scalar multiplication)
- $\exists 0 \in \mathbb{V} | x + 0 = 0 + x = x$ (existence of null element)
- $\forall x \in \mathbb{V} \exists (-x) | x + (-x) = 0$ (existence of the inverse)

The inner product (dot product) provides an additional operation to measure and compare vectors. It takes a couple of vectors and returns a scalar:

$$< -, - >= \mathbb{V} \times \mathbb{V} \to \mathbb{C}$$

It measures the similarity between two vectors; if the inner product is zero, then the vectors are orthogonal (maximally different). It is also defined axiomatically:

For $x, y, z \in \mathbb{V}$ and $\alpha, \beta \in \mathbb{C}$:

- $< x + y, z >=< x, z > + < y, z >$ (Distributive w.r.t. vector addition)
- $< x, y >=< y, x >^*$ (Commutative with conjugation)
- $< \alpha x, y >= \alpha^* < x, y >, < x, \alpha y >= \alpha < x, y >$ (Multiplication, conjugate scalar if applied to first element)
- $< x, y >\geq 0$
- $< x, x >= 0 \Leftrightarrow x = 0$
- if $< x, y >= 0$ and $x, y \neq 0$ then x and y are called orthogonal

The formula to define the inner product in $\mathbb{R}^2$ is:

$$< x, y >= x_0 y_0 + x_1 y_1 = ||x|| \cdot ||y|| cos\alpha$$

which can be extended to $\mathbb{R}^N$.

Now, for $L_2([-1,1])$:

$$< x, y >= \int_{-1}^{1} x(t)y(t)dt$$

Also, note that:

- inner product defines a norm: $||x|| = \sqrt{< x, x >}$
- norm defines a distance $d(x, y) = ||x - y||$

2.2.b SIGNAL SPACE

We know that finite-length and periodic signals live in $\mathbb{C}^N$, a vector $\boldsymbol{x}$ is denoted by $[x_0 \ x_1 \ \dots \ x_{N-1}]^T$, all operations are well defined, and the space of N-periodic signals is sometimes indicated by $\widetilde{\mathbb{C}}^N$.

In $\mathbb{C}^N$, the inner product is defined as follows:

$$< x, y >= \sum_{n=0}^{N-1} x^*[n]y[n]$$

For infinite-length sequences, we will define the vector space of square-summable sequences $l_2(\mathbb{Z})$ to avoid the “explosion” of the sum:

$$< x, y >= \sum_{n=-\infty}^{\infty} x^*[n]y[n]$$

We require sequences to be square-summable $\sum |x[n]|^2 < \infty$.

However, many interesting signals are not in $l_2(\mathbb{Z})$.

Now, let us talk about completeness: if a vector space is closed under the limiting operation, we say that the vector space is complete, which will be useful to prove some fundamental results such as the sampling theorem. When a vector space equipped with an inner product is also complete, we call the vector space a Hilbert space.

2.3 BASES

The axioms of a vector space tell us how to combine vectors together. A linear combination of vectors is the basic operation that we perform in vector spaces:

$$g = \alpha x + \beta y$$

When we combine vectors together to obtain new vectors in the space, one usual question is whether we can find a minimal set of vectors $\{\mathbf{w}^{(k)}\}$ so that we can express any vector in the space as a linear combination of this base factor. This set of building blocks will be called the bases. Here is an example: the canonical basis for the Euclidean plane (canonical R^2 basis):

$$e^{(0)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$
$$e^{(1)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
$$\begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = x_0 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + x_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

It is possible to find a basis for infinite-dimensional spaces and for function vector spaces. An infinite number of bases exists for these spaces. For functions, one of the most famous is the Fourier basis, where for [–1,1]:

$$\frac{1}{\sqrt{2}}, cos(\pi t), sin(\pi t), cos(2\pi t), sin(2\pi t), cos(3\pi t), sin(3\pi t), ...$$

Now let us give you a formal definition of a basis for a vector space *H*: let *W* be a set of *K* vectors such that $W = \{\mathbf{w}^{(k)}\}_{k=0,1,...,K-1}$, *W* is a basis for *H* if:

1. if we can write for all $\mathbf{x} \in$ H: $\mathbf{x} = \sum_{k=0}^{K-1} {}_k\mathbf{w}^{(k)}$, $\alpha_k \in \mathbb{C}$

2. the coefficients α_k are unique

Uniqueness of representation is equivalent to linear independence:

$$\sum_{k=0}^{K-1} \alpha_k w^{(k)} = 0 \Leftrightarrow \alpha_k = 0, k = 0, 1, ..., K-1$$

In an orthogonal basis, the basis vectors are mutually orthogonal:

$$< w^{(k)}, w^{(n)} >= 0, \text{ for } k \neq n$$

In an orthonormal basis, the basis vectors are mutually orthonormal:

$$< w^{(k)}, w^{(n)} >= \delta[n-k]$$

A frequent question is how to find the coefficients: if the basis is orthonormal, this is the direct computation:

$$\alpha_k = < w^{(k)}, x >$$

Also, this is how to find the new coefficients when a change of basis is made:

$$x = \sum_{k=0}^{K-1} \alpha_k w^{(k)} = \sum_{k=0}^{K-1} \beta_k v^{(k)}$$

if $v^{(k)}$ is orthonormal:

$$\beta_h = < v^{(h)}, x > \; = < v^{(h)}, \sum_{k=0}^{K-1} \alpha_k w^{(k)} > \; = \sum_{k=0}^{K-1} \alpha_k < v^{(h)}, w^{(k)} >$$

2.4 SUBSPACE-BASED APPROXIMATION

A vector subspace is a subset of vectors closed under addition and scalar multiplication. We can extend the concept of subspace to more complicated vector spaces such as function vector spaces. Subspaces have their own bases.

Problem:

- vector $\mathbf{x} \in V$
- subspace $S \subseteq V$
- approximate $\mathbf{x}$ with $\hat{\mathbf{x}} \in S$

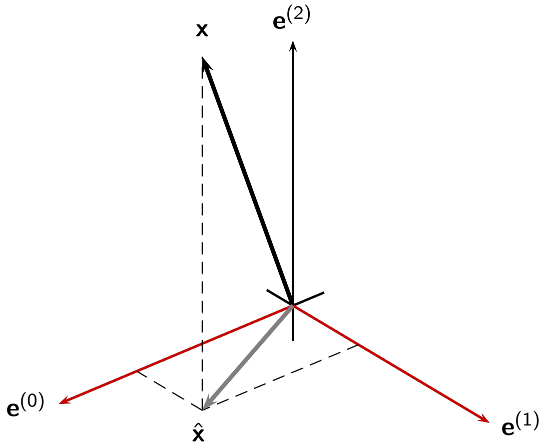


FIGURE 1

2:18 16:18

Approximation of a vector *x* in *V* in subspace *S*

Consider an orthonormal basis $\{\mathbf{s}^{(k)}\}_{k=0,1,\ldots,K-1}$, for the subspace *S*. The orthogonal projection of a vector *x* onto this subspace is defined by:

$$\hat{x} = \sum_{k=0}^{K-1} < s^{(k)}, x > s^{(k)}$$

The orthogonal projection is the best approximation for a given vector *x* onto the given subspace *S*. It has two fundamental properties:
– The orthogonal projection has minimum-norm error: $argmin_{y \in S} ||x - y|| = \hat{x}$
– Its error is orthogonal to the approximation: $< x - \hat{x}, \hat{x} >= 0$

We can build an orthonormal basis using the Gram-Schmidt orthonormalization procedure:
$\{\mathbf{s}^{(k)}\}$ (original set) → $\{\mathbf{u}^{(k)}\}$ (orthonormal set). These are the basic steps:

Algorithmic procedure: at each step k

$$p^{(k)} = s^{(k)} - \sum_{n=0}^{k-1} < u^{(n)}, s^{(k)} > u^{(n)}$$

$$u^{(k)} = p^{(k)} / ||p^{(k)}||$$

If we apply the Gram-Schmidt procedure to the set of polynomials $P_N([-1,1])$ we obtain a set of orthonormal polynomials called the Legendre polynomials, which are very useful for function approximations.

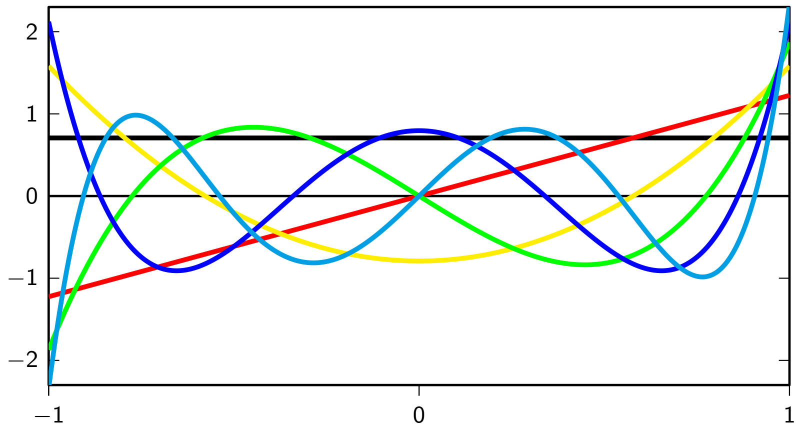


FIGURE 2

12:50

16:18

Legendre polynomials

Signal of the Day

EXOPLANET HUNTING

Looking at an image of the universe taken by the space telescope Hubble, the universe contains an estimated 100 billion galaxies, each containing hundreds of billions of stars. Thus there is a very high likelihood that another star planet system exists, comparable to our solar system. We are going to study a method for finding exoplanets: the transit method. As the planet transits in front of its star, the observer measures the reduction in the flux of light. It can be shown that the relative change in the flux is equal to the ratio between the radius of the planet and the radius of the star to the square: it is called the transit depth. We can adopt another point of view and use a CCD camera to measure the amount of light (technically the amount of photons) from the star that reaches the observer. This is where Digital Signal Processing comes into play. For further information, you will have to follow a course on statistical signal processing.

3.1.a THE FREQUENCY DOMAIN

The goal of the discrete Fourier transform (DFT) is to express a signal in terms of its sinusoidal components. Every sustainable dynamic system exhibits an oscillatory behavior, which is easy to describe and parameterize. From the analysis point of view, the Fourier transform basically changes from the time domain to the frequency domain. A signal is decomposed into different frequencies, which reveals hidden signal properties. The converse of this process is to revert from the frequency domain to the time domain and synthesize a signal from its frequency components. We can use that to generate signals that have a known frequency content, which can in turn be used in signal processing to fit signals to specific frequency regions.

3.1.b THE DFT AS A CHANGE OF BASIS

For a finite-length signal, the Fourier analysis is simply a change of basis. The Fourier basis for C^N is the following:

Claim: the set of N signals in $\mathbb{C}^N$ $w_k[n] = e^{j\frac{2\pi}{N}nk}$ n, k = 0,1,..., N is an orthogonal basis in $\mathbb{C}^N$.
In vector notation: $w^{(k)}{}_{k=0,1,\ldots,N-1}$ with $w_n^{(k)} = e^{j\frac{2\pi}{N}nk}$.

As an example, we will study $\boldsymbol{w}^{(3)}$; the fundamental frequency of this vector will be $2\pi/3$, so we have three periods both on the real and imaginary axes.

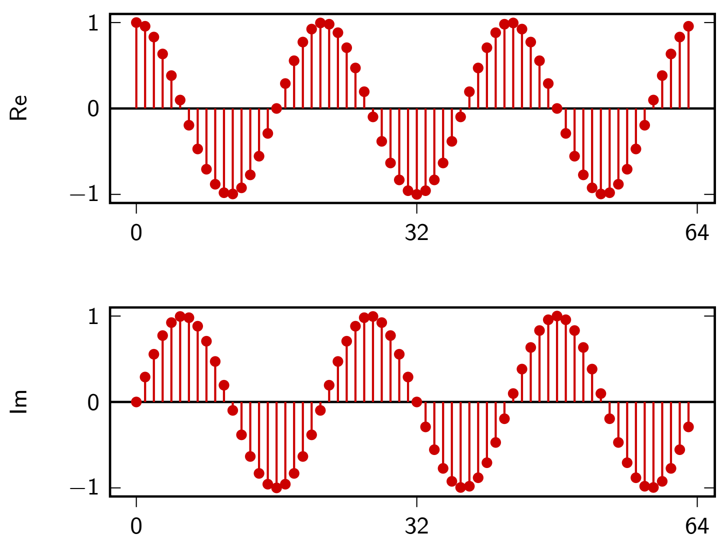


FIGURE 1

5:29

10:49

Example of $\boldsymbol{w}^{(3)}$

As we go up with the basis index, the vectors start moving faster. As we increase the index of the Fourier vector, the underlying frequency increases as well. Because of the discrete nature of the signal, it sometimes might be hard to understand the signs of the shape of the signal. A tell-tale sign of high frequency is a sign alternation. This constitutes a set of orthogonal vectors in $\mathbb{C}^n$ and can be proved by simple calculus: the inner product.

$$< w^{(k)}, w^{(h)} >= \sum_{n=0}^{N-1} (e^{j\frac{2\pi}{N}nk})^* e^{j\frac{2\pi}{N}nh} = \sum_{n=0}^{N-1} e^{j\frac{2\pi}{N}(h-k)n} = \begin{cases} N \text{ for } h = k \\ \dfrac{1-e^{j2\pi(h-k)}}{1-e^{j\frac{2\pi}{N}(h-k)}} = 0 \text{ otherwise} \end{cases}$$

How do we solve the last sum? When $h = k$, we take the inner product of a vector with itself. All the elements in the sum will be equal to 1, so the sum will converge to N. When $h \neq k$, then it is a type a^N series. Note that they are not orthonormal, we will have to normalize the vectors, dividing them by $\sqrt{N}$.

3.2.a THE DEFINITION OF DFT

Given the arbitrary element of $\mathbb{C}^n$, the analysis formula will give us *N* new coefficients for the vector in the new basis, each of which will be denoted by *X*[*k*]; the synthesis formula will allow us to retrieve the original vector in the canonical basis.

Analysis formula:

$$X_k = < w^{(k)}, x >$$

Synthesis formula:

$$x = \frac{1}{N} \sum_{k=0}^{N-1} X_k w^{(k)}$$

We can express the change of basis in matrix notation. The Hermitian operator is a combination of transposition and conjugation of each element of the matrix.

Define $W_N = e^{-j\frac{2\pi}{N}}$ (or simply W when N is evident from the context). Change of basis matrix W with $W[n,m] = W_N^{nm}$ where

$$W = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 & \\ 1 & W^1 & W^2 & W^3 & \cdots & W^{N-1} \\ 1 & W^2 & W^4 & W^6 & \cdots & W^{2(N-1)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & W^{N-1} & W^{2(N-1)} & W^{3(N-1)} & \cdots & W^{(N-1)^2} \end{bmatrix}$$

Analysis formula:

$$X = Wx$$

Synthesis formula:

$$x = \frac{1}{N} W^H X$$

A third way of looking at the DFT is to explicitly consider the operations involved as a sum:

Analysis formula:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}nk}$$

Synthesis formula:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}nk}$$

3.2.b EXAMPLES OF DFT CALCULATION

The DFT is a linear operator. DFT of the delta signal: the delta signal will isolate the 0-component of each basis vector, so the product will be 1 for all the values of the index *k*. So the Fourier transform of discrete time delta is constant 1.

DFT of $x[n] = \delta[n], \quad x[n] \in \mathbb{C}^N$

$$X[k] = \sum_{n=0}^{N-1} \delta[n]\, e^{-j\frac{2\pi}{N}nk}$$
$$= 1$$

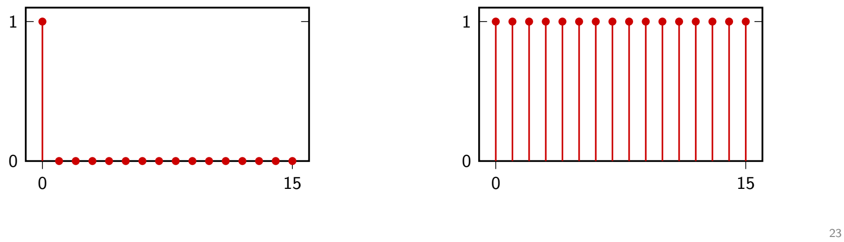


FIGURE 1

0:30 13:37

DFT of a delta signal

DFT of the unit function: corresponds to a delta (fig. 2).

DFT of $x[n] = 1, \quad x[n] \in \mathbb{C}^N$

$$X[k] = \sum_{n=0}^{N-1} e^{-j\frac{2\pi}{N}nk}$$
$$= N\delta[k]$$

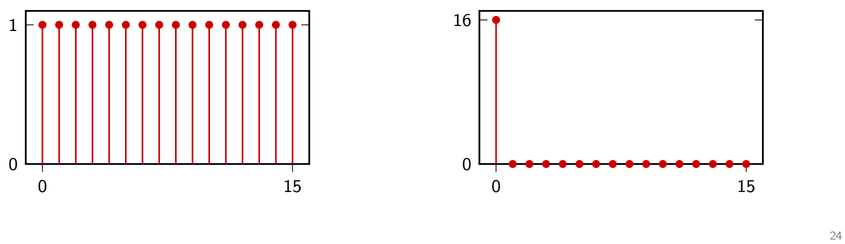


FIGURE 2

1:00 13:37

DFT of a delta signal

DFT of a sinusoidal function of frequency multiple of $2\pi/N$: one trick is to decompose the sines and cosines into exponentials (using Euler's formula), which then will correspond to a vector of our basis. Here is an example:

$$\text{DFT of } x[n] = 3cos(2\pi/16n + \pi/3),\ x \in \mathbb{C}^{64}:$$

$$\begin{aligned} x[n] &= 3cos(\frac{2\pi}{16}n + \frac{\pi}{3}) \\ &= 3cos(\frac{2\pi}{64}4n + \frac{\pi}{3}) \\ &= \frac{3}{2}[e^{j\frac{2\pi}{64}4n}e^{j\frac{\pi}{3}} + e^{-j\frac{2\pi}{64}4n}e^{-j\frac{\pi}{3}}] \\ &= \frac{3}{2}(e^{j\frac{\pi}{3}}w_4[n] + e^{-j\frac{\pi}{3}}w_{60}[n]) \end{aligned}$$

$$X[k] = \begin{cases} 96e^{j\frac{\pi}{3}} fork = 4 \\ 96e^{-j\frac{\pi}{3}} fork = 60 \\ 0 otherwise \end{cases}$$

A clear way to plot a Fourier transform is to plot both the magnitude and phase of the Fourier coefficients.

DFT of the step vector: from the following calculus, we can see that $X[0] = M$, $W[k] = 0$ is Mk/N is an integer and that the phase of $X[k]$ is linear (except at time changes), leading to the plot on the following figure 3.

$$\begin{aligned} X[k] &= \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{N}nk} \\ &= \sum_{n=0}^{M-1} e^{-j\frac{2\pi}{N}nk} \\ &= \frac{1 - e^{-j\frac{2\pi}{N}kM}}{1 - e^{-j\frac{2\pi}{N}k}} \\ &= \frac{e^{-j\frac{\pi}{N}kM}[e^{j\frac{\pi}{N}kM} - e^{-j\frac{\pi}{N}kM}]}{e^{-j\frac{\pi}{N}k}[e^{j\frac{\pi}{N}k} - e^{-j\frac{\pi}{N}k}]} \\ &= \frac{sin(\frac{\pi}{N}Mk)}{sin(\frac{\pi}{N}k)}e^{j\frac{\pi}{N}(M-1)k} \end{aligned}$$

DFT of length-4 step in $\mathbb{C}^{64}$ | DFT of length-4 step in $\mathbb{C}^{64}$ (phase wrapped)

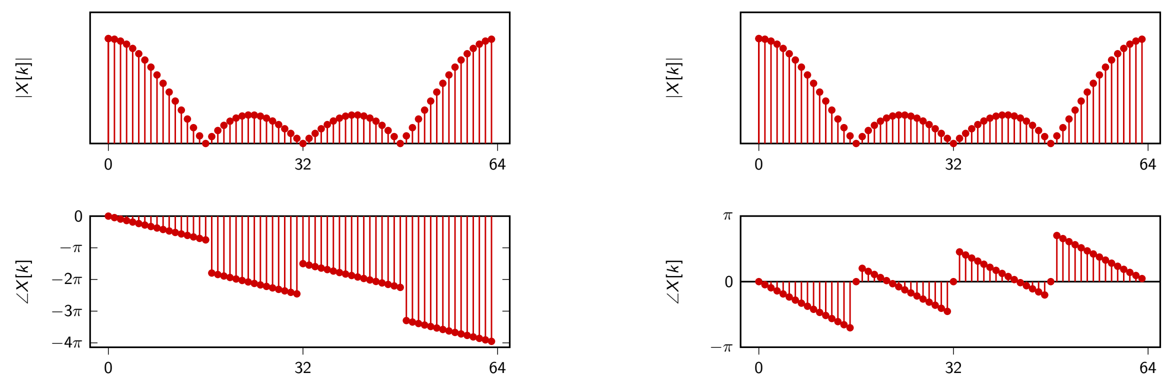


FIGURE 3

12:20 13:37

DFT of a step vector

Often, the phase displayed is wrapped over $[-\pi, \pi]$, adding or subtracting 2π.

3.2.c INTERPRETING A DFT PLOT

We have frequency coefficients from 0 to $N-1$: for example, the first $N/2$ coefficients correspond to frequencies less than π – counter-clockwise movement, the other $N/2$ coefficients correspond to frequencies more than π – clockwise movement. Low frequencies are around 0 and $N-1$, whereas high frequencies are around $N/2$.

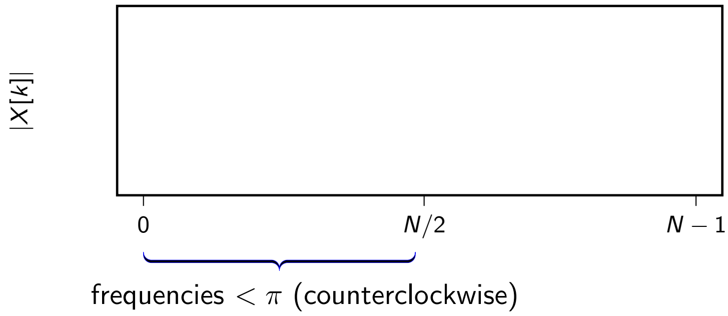


FIGURE 1a

0:20 4:58

Interpreting a DFT plot

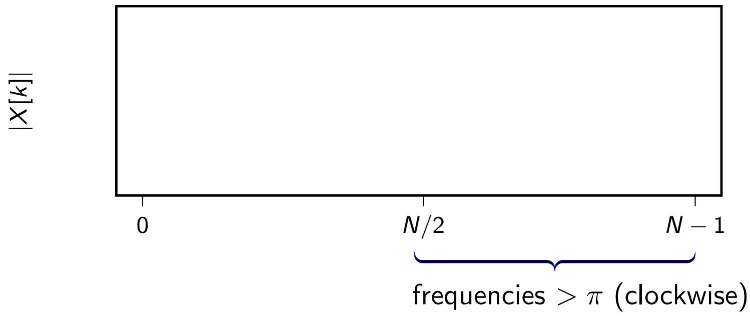


FIGURE 1b

0:30 4:58

Interpreting a DFT plot

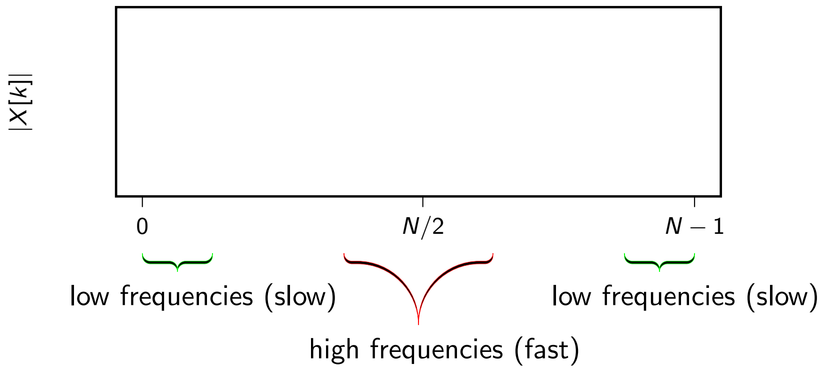


FIGURE 1c

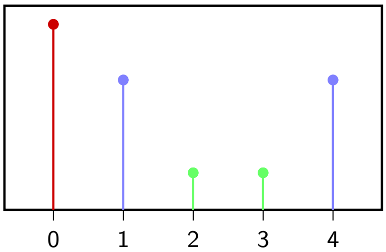
0:45 4:58

Interpreting a DFT plot

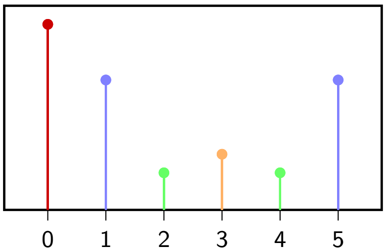
The DFT of real signals is symmetric in magnitude. In discrete-time, we have two situations in function of the parity of the signal. We say that the magnitude of the kth coefficient will be equal to the magnitude of the $(N - k)$th coefficient.

For real signals the DFT is "symmetric" in magnitude:

$$|X[k]| = |X[N-k]| \text{ for } k = 1, 2, \dots, \lfloor N/2 \rfloor$$



$N = 5$, odd length



$N = 6$, even length

FIGURE 2

4:00

4:58

3.3.a DFT ANALYSIS

The presence of spikes on the DFT plot indicates a strong sinusoidal component. More importantly, the sinusoidal component has a frequency that is a multiple of the basic frequency for the space that the signal lives in.

The DFT coefficient for $k = 0$ is a non-normalized average of all the data points. The period of the signal can be found by dividing the number of points N by the index corresponding to the main peak. The fastest positive frequency is π. So, two samples are required to complete a full revolution. Now, the clock T_s can also be expressed as $1/F_s$, where F_s is the frequency of the system. This is the standard relationship between period and frequency for any system. So, if the real world for the fastest sinusoid in a digital system is $2T_s$ measured in seconds, the real world frequency for the fastest sinusoid is $F_s/2$. Therefore the maximum frequency once the period between samples has been determined is $F_s/2$ or equivalently $1/2T_s$.

3.3.b DFT EXAMPLE – ANALYSIS OF MUSICAL INSTRUMENTS

Harmonics are what actually give the timbre of the instrument. For a signal of frequency w, we say that the harmonics are the multiples of w. We will look at the Fourier spectra of three instruments: the cello, saxophone, and violin (fig.1). A mathematical method to help us differentiate those three sounds from one another and guess at the frequency is to use the DFT over a few periods (fig. 2). The spectra look different but the peaks are at the same places, because the frequency played is 220 Hz and the harmonics are multiples of 220.

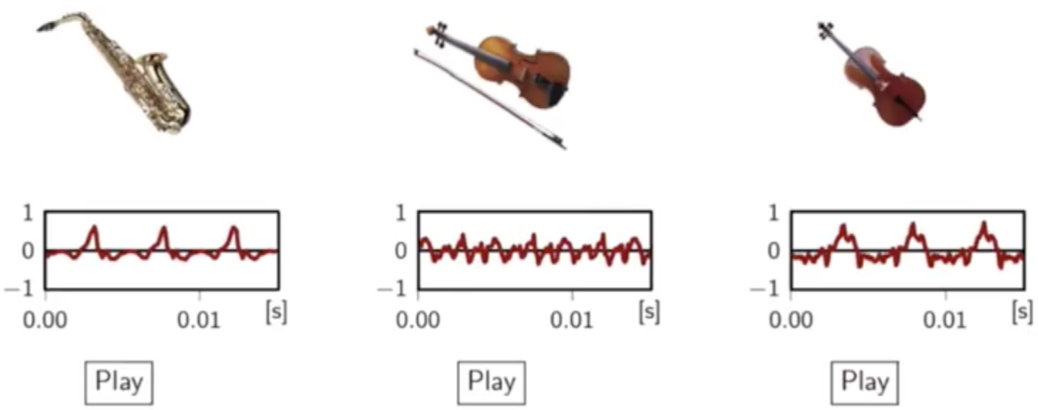


FIGURE 1

2:02 4:41

Comparison between the sound signals of the three instruments

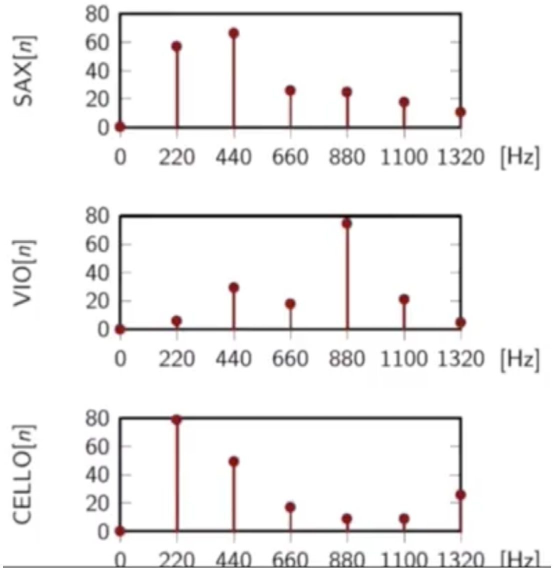


FIGURE 2
2:50 4:41

Comparison between the three DFTs

3.3.c DFT SYNTHESIS

Now let's look at the sinusoidal generator. We have a complex exponential with a frequency, which is a multiple of $2\pi/N$ and has an initial phase. The way this generator works is by successively generating points on the unit circle starting at the phase and proceeding in increments of $2\pi/N$.

Synthesis: the sinusoidal generator

$$w_k[n] = e^{j(\frac{2\pi}{N}kn+\phi_k)}$$

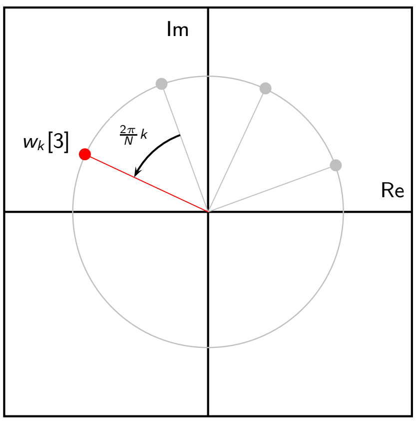
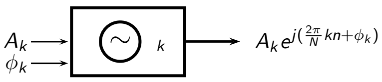


FIGURE 1
0:20 5:46

Sinusoidal generator

The DFT synthesis formula can be imagined as a machine composed of N sinusoidal generators, each of which is initialized with a gain and phase factor. All their outputs will be summed together, which will give us the original signal back.

Synthesis: the sinusoidal generator



DFT synthesis formula

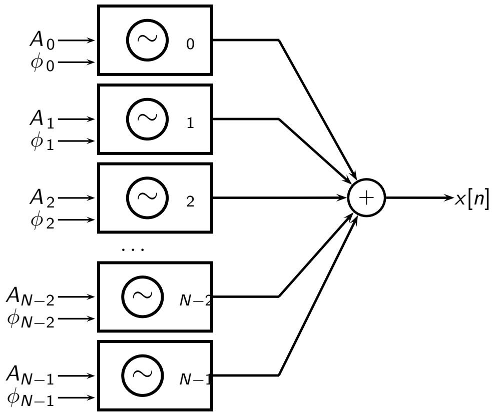


FIGURE 2

1:14 5:46

DFT synthesis "machine"

What happens if we turn the crank of the machine more than N times? The output will become periodic.

$$x[n + N] = x[n]$$

This is actually quite apparent from the structure of the synthesis and analysis formulae of the DFT. Let's start with the synthesis formula. The complex exponential is 2π periodic, so if you push n over $N - 1$, this will simply cycle over, as if N was looping over the 0 to $N - 1$ range. So in the end, you can actually safely take N to be from the set of integers and the output will be an N-periodic signal in a time domain. The same holds for the analysis formula. You can let the index, k, roam over the entire set of integers. And since it appears only in this complex exponential here, again, it will be as if k was looping over the 0 to $N - 1$ range. So what really happens is that you can consider the sequence of DFT coefficients as an N-periodic signal in the frequency domain.

3.3.d DFT EXAMPLE – TIDE PREDICTION IN VENICE

Tides are periodical phenomena. We will try to find out whether we can predict tides using Fourier transforms. Using Fourier coefficients we can recover a signal very close to the original one and thus continue further than the original data set, predicting what is going to happen in the future.

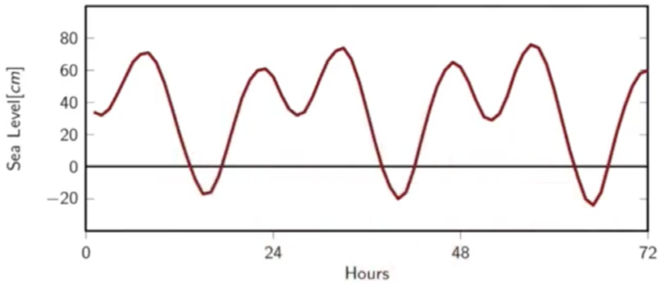


FIGURE 1

1:18 4:10

Data set

Look at the Fourier domain of a year of measurements:

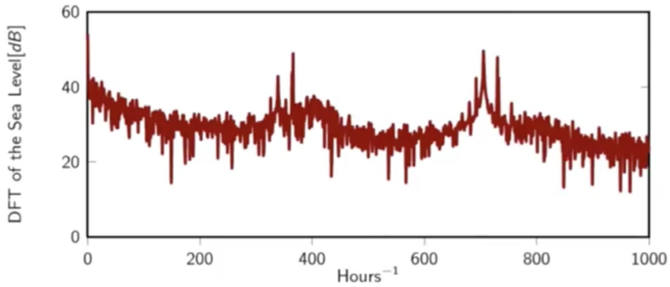


FIGURE 2

1:41 4:10

Fourier transform of data set log-magnitude

- We consider only $K = 1, 3, 5, 11$ Fourier coefficients
- Darker gray represents approximation with more coefficients.
- The approximations are very precise with a limited number of coefficient.

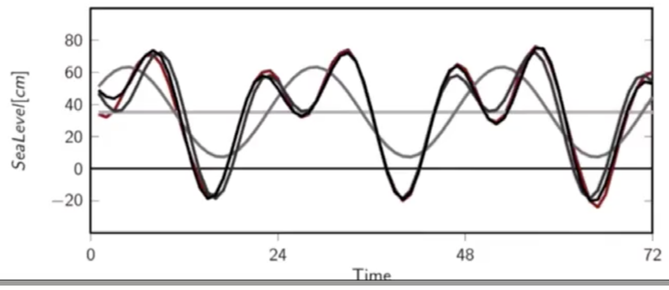


FIGURE 3

4:00 4:10

Approximation of original data set

3.3.e DFT EXAMPLE – MP3 COMPRESSION

Compression introduces noise and loss, so why can't we hear it on an MP3 record? Because this noise has been shaped to fool the human hearing system. One of the key elements of the compression algorithm is to shape errors adaptively as the Fourier transform spectrum varies, constantly adapting using the short time Fourier transform domain. The idea is essentially a local DFT.

Signal of the Day

THE FIRST MAN-MADE SIGNAL FROM OUTER SPACE

Sputnik was the first artificial Earth satellite, launched by the Soviet Union on October 4, 1957. Its main spherical body was surrounded by four external radio antennae, which transmitted a signal. This signal was thus the first man-made signal sent from outer space. Amateur radio operators could detect the signal from all over the world. The transmitted signal was just a sequence of beeps. If we plot the magnitude of the Fourier transform of the signal transmitted by Sputnik, there is a large component at $\omega = 0$. We can also observe two small peaks, which correspond to the frequency of the transmitted beeps. We see that the two peaks now appear at 1653 Hertz and –1653 Hertz.

3.4.a THE SHORT-TIME FOURIER TRANSFORM

Dual-Tone Multi Frequency dialing signaling is a method used in analog telephones. For each button press, we generate a sinusoid composed of two frequencies taken from a specific matrix. This is done to minimize the error whenever someone tries to recover the sequence of numbers dialed on an analog phone. When we take the DFT of these signals, the frequencies associated with each number will appear, but we cannot tell in which order the numbers have been pressed. The idea behind the short-time Fourier transform is to take small pieces of the original signal in order to localize in time the results obtained on the Fourier transform. We move the analysis window.

Take a small signal pieces of lenght L and look at the DFT of each piece: $X[m;k] = \sum_{n=0}^{L-1} x[m+n]e^{-l\frac{2\pi}{L}nk}$.

3.4.b THE SPECTROGRAM

The spectrogram is a way of showing time-varying spectral information in one single plot. We color-code the magnitudes: dark hues correspond to small values and light hues correspond to high values. We use the logarithm of the magnitude in order to compress the range of values. We put the spectral slices one after another in order to get an image like the following image (fig. 1). On the vertical axis, we place the DFT coefficients.

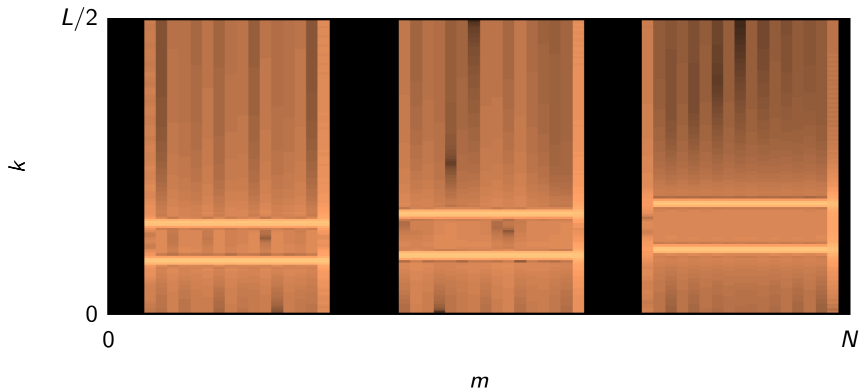


FIGURE 1

1:17 7:41

DTMF spectrogram (analog telephone example, 3.4.a)

If we know the "system clock" $F_s = 1/T_s$ we can label the axis:
– highest positive frequency $F_s/2$ Hz
– frequency resolution F_s/L Hz
– width of time slices: LT_s seconds

3.4.c TIME-FREQUENCY TILING

Speech is a particularly difficult signal to analyze. The idea is to split the signals.

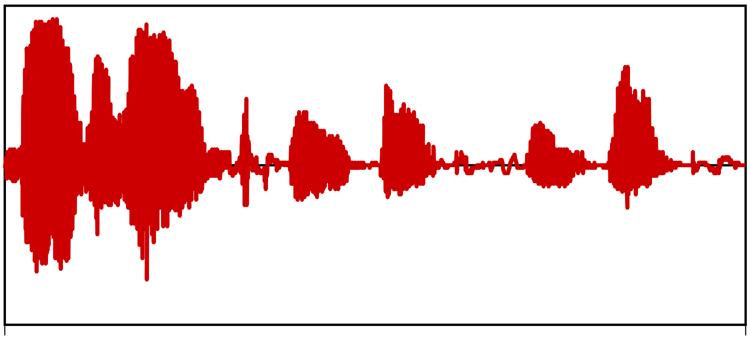


FIGURE 1

0:55

5:31

"There is a lag between thoughts and acts"

Spectrograms can be useful for getting information: narrow-band spectrograms give us information on the harmonic parts of speech and wide-band spectrograms give us information on the pulse-like and noise-like consonant sounds in speech.

The short time Fourier transform determines a tiling of the time-frequency plane, where the size of each tile is specified by the time and frequency resolution of the STFD.

3.5.a DISCRETE FOURIER SERIES

When we talk about the periodicity of the Fourier transform, we refer to the Discrete Fourier Series (DFS). The DFS is basically the DFT with periodicity explicit. The DFS maps an *N*-periodic signal onto an *N*-periodic sequence of Fourier coefficients, and the inverse DFS maps an *N*-period sequence of Fourier coefficients onto an *N*-periodic signal. If we take the DFS of this periodic sequence with a shift, we can easily work out that the DFT coefficient for index *k* is equal to the DFT coefficient for index *k* of the original sequence without the shift. The DFS helps us understand how to define time shifts for finite-length signals.

For an N-periodic sequence $\tilde{x}[n]$:

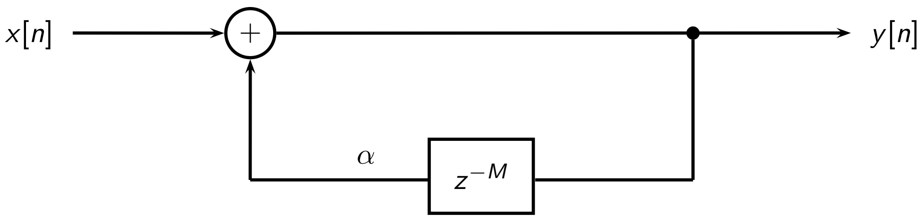
1. $\tilde{x}[n-M]$ for all $M \in \mathbb{N}$
2. DFS $\tilde{x}[n-M] = e^{-j\frac{2\pi}{N}Mk}\tilde{X}[k]$
3. IDFS $e^{-j\frac{2\pi}{N}Mk}\tilde{X}[k] = \tilde{x}[n-M]$.

For an N-point signal x[n]:

1. x[n-M] is not well-defined
2. build $\tilde{x}[n] = x[n \bmod N] \Rightarrow \tilde{X}[k] = X[k]$
3. IDFT $e^{-j\frac{2\pi}{N}Mk}X[k] =$ IDFS $e^{-j\frac{2\pi}{N}Mk}\tilde{X}[k] = \tilde{x}[n-M] = x[(n-M) \bmod N]$
4. shifts for finite length signals are naturally circular.

3.5.b KARPLUS-STRONG REVISITED AND DFS

An N-periodic sequence has only N degrees of freedom. The DFS provides us with a sequence that only has N distinct Fourier coefficients.



$$y[n] = \alpha\, y[n-M] + x[n]$$

FIGURE 1

0:22 7:54

Karplus-Strong circuit

Consider a finite-length signal of length M. What happens if we take the DFT of two periods of a finite-length signal? The DFT is the same as if it were one period with extra 0-valued coefficients. Let's generalize this case by taking L periods, and let y be that signal. Let's calculate its DFT:

$$\begin{aligned}
X_L[k] &= \sum_{n=0}^{LM-1} y[n]e^{-j\frac{2\pi}{LM}nk}, k = 0, 1, 2, ..., LM-1 \\
&= \sum_{p=0}^{L-1}\sum_{n=0}^{M-1} y[n-pM]e^{-j\frac{2\pi}{LM}(n-pM)k} \\
&= \sum_{p=0}^{L-1}\sum_{n=0}^{M-1} y[n]e^{-j\frac{2\pi}{LM}nk}e^{-j\frac{2\pi}{L}pk} \\
&= (\sum_{p=0}^{L-1} e^{-j\frac{2\pi}{L}pk}) \sum_{n=0}^{M-1} \tilde{x}[n]e^{-j\frac{2\pi}{LM}nk}
\end{aligned}$$

$$\sum_{p=0}^{L-1} e^{-j\frac{2\pi}{L}pk} = \begin{cases} L, \text{if } k \text{ multiple of } L \\ 0, \text{otherwise} \end{cases}$$

Moreover, these nonzero coefficients are just scaled versions of the DFT coefficients of the original finite-length sequence. Therefore, all the spectral information of an N-periodic sequence is entirely captured by the DFT coefficients of one period.

3.6.a KARPLUS-STRONG REVISITED AND DTFT

How do we calculate the Fourier transform of infinite non-periodic signals? We can use the Karplus-Strong algorithm simply by putting the factor alpha to less than 1 and then compute its DFT. As N goes to infinity, the frequency becomes smaller, and the set of frequencies in the $[0, 2\pi]$ range becomes denser. In the limit, the set of multiples of the fundamental frequency $2\pi/N$ will become so dense that we will try to replace this by a real valued variable frequency. And this real valued variable will last for the $[0, 2\pi]$ interval. So if we replace that in the formulation for the DFT, we now get a sum over all the points in the signal.

Here is a definition of the Discrete-Time Fourier Transform: the signal needs to be square summable (finite energy).

- $x[n] \in l_2(\mathbb{Z})$
- define the function of $\omega \in \mathbb{R}$ $F(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$
- inversion (when $F(\omega)$ exists):
  $x[n] = \frac{1}{2\pi}\int_{-\pi}^{pi} F(\omega)e^{j\omega n}d\omega,\ n \in \mathbb{Z}$

$F(\omega)$ is 2π-periodic. To stress its periodicity, we will write $F(\omega)$ as $X(e^{j\omega})$, reminding us that it is 2π-periodic. We choose $[-\pi, \pi]$ interval as the representative interval.

$$
\begin{aligned}
Y(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} y[n]e^{-j\omega} \\
&= \sum_{p=0}^{\infty}\sum_{n=0}^{M-1} \alpha^p \bar{x}[n]e^{-j\omega(pM+n)} \\
&= \sum_{p=0}^{\infty} \alpha^p e^{-j} \sum_{n=0}^{M-1} \bar{x}[n]e^{-j\omega n} \\
&= A(e^{j\omega M})\bar{X}(e^{j\omega M})
\end{aligned}
$$

The factor M in the exponent simply implies that there is scaling of the frequency axis.

So here we have derived the DTFT, namely the spectrum of an infinite non periodic signal that is not a trivial signal. Now so far we have treated the DTFT as a formal operator. And in the next module we will see how the DTFT relates to the concept of change of basis in an appropriate Hilbert space.

3.6.b EXISTENCE AND PROPERTIES OF THE DTFT

Existence means that the sum defining the DTFT does not explode. Our initial hypothesis is that our signal is absolutely summable.

$$
\begin{aligned}
|X(e^{j\omega})| &= |\sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}| \\
&\leq \sum_{n=-\infty}^{\infty} |x[n]e^{-j\omega n}| \\
&= \sum_{n=-\infty}^{\infty} |x[n]| \qquad \infty
\end{aligned}
$$

The inversion of the integral and the sum is possible because x is absolutely summable.

$$
\frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})e^{j\omega n}d\omega = \frac{1}{2\pi}\int_{-\pi} \pi(\sum_{k=-\infty}^{\infty} x[k]e^{-j\omega k})e^{j\omega n}d\omega \quad = \sum_{k=-\infty}^{\infty} x[k]\int_{-\pi}^{\pi} \frac{e^{j\omega(n-k)}}{2\pi}d\omega = x[n]
$$

Formally, the DTFT is a change of basis over an infinite, uncountable basis. The DTFT exists for all square-summable sequences.

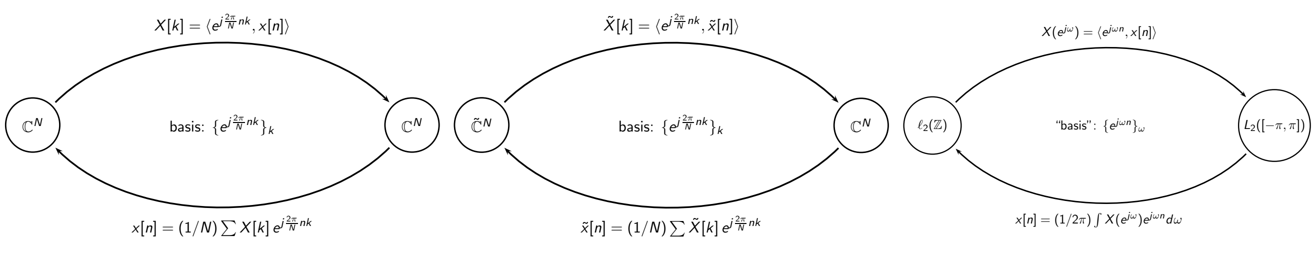


FIGURE 1

4:10 7:10

DFT, DFS, DTFT

Here are the DTFT properties:

- linearity $\text{DTFT}\{\alpha x[n] + \beta y[n]\} = \alpha X(e^{j\omega}) + \beta Y(e^{j\omega})$
- time shift $\text{DTFT}\{x[n-M]\} = e^{-j\omega M}X(e^{j\omega})$
- modulation (dual) $\text{DTFT}\{e^{j\omega_0 n}\} = X(e^{j(\omega-\omega_0)})$
- time reversal $\text{DTFT}\{x[-n]\} = X(e^{j\omega})$
- conjugation $\text{DTFT}\{x^*[n]\} = X^*(e^{-j\omega})$
- if $x[n]$ is symmetric, the DTFT is symmetric: $x[n] = x[-n] \Leftrightarrow \text{X}(e^{j\omega}) = X(e^{j\omega})$
- if $x[n]$ is real, the DTFT is Hermitian-symmetric: $x[n] = x^*[n] \Leftrightarrow X(e^{j\omega}) = X^*(e^{j\omega})$
- special case: if $x[n]$ is real, the magnitude of the DTFT is symmetric:$x[n] \in \mathbb{R}|X(e^{j\omega})| = |X(e^{-j\omega})|$
- more special case: if $x[n]$ is real and symmetric, $X(e^{j\omega})$ is also real and symmetric

3.6.c THE DTFT AS A CHANGE OF BASIS

The DFT of the constant 1 is very well defined, and is equal to N times the delta function in frequency. But the DTFT of 1 is, by definition, the sum from n goes from $-\infty$ to $+\infty$ of e^{-jwn}, at $w = 0$, and that sum diverges. This is because the unit sequence is not square summable, as are many other interesting signals. We will thus introduce the Dirac delta function.

Defined by the "sifting" property: $\int_{-\infty}^{\infty} \delta(t-s)f(t)dt = f(s)$ for all functions of $t \in \mathbb{R}$.

The area under this curve is equal to 1, as it is basically the function $\lim_{k\to inf} k \cdot rect(kt)$. We will be using the Dirac delta function in the frequency domain; as all DTFT spectra are 2π periodic, and so to use this tool in the frequency domain, we have to periodize it: its 2π-periodic version is called a pulse train.

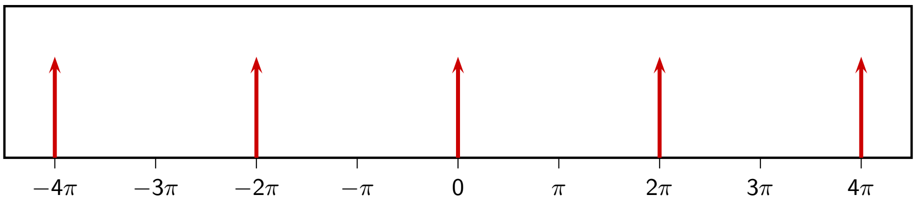


FIGURE 1

6:04 9:48

Graphic representation of the pulse train

$$\text{IDTFT}\{\tilde{\delta}(\omega)\} = \frac{1}{2\pi}\int_{-\pi}^{\pi} \tilde{\delta}(\omega)e^{j\omega n}d\omega = \int_{-\pi}^{\pi} \delta(\omega)e^{j\omega n}d\omega = e^{j\omega}|_{\omega=0} = 1$$

$\text{IDFT}\{\tilde{\delta}(\omega-\omega_0)\} = e^{j\omega_0 n}$. so:

- $\text{DTFT}\{1\} = \tilde{\delta}(\omega)$
- $\text{DTFT}\{e^{j\omega_0 n}\} = \tilde{\delta}(\omega-\omega_0)$
- $\text{DTFT}\{cos(\omega_0 n)\} = [\tilde{\delta}(\omega-\omega_0) + \tilde{\delta}(\omega+\omega_0)]/2$
- $\text{DTFT}\{sin(\omega_0 n)\} = -j[\tilde{\delta}(\omega-\omega_0) - \tilde{\delta}(\omega+\omega_0)]/2$

3.7.a SINUSOIDAL MODULATION

There are three broad categories of signals, in both discrete and continuous times, according to where the spectral energy mostly concentrates:

- Lowpass signals (a.k.a. "baseband" signals): the energy is mostly concentrated around 0 and there is no energy outside.
- Highpass signals: the energy is mostly concentrated around pi and $-\pi$ and there is no energy around 0.
- Bandpass signals: the energy is mostly concentrated around $-\pi/2$ and $\pi/2$.

Let's now consider a sinusoidal modulation, which is done by multiplying a signal by a cosine:

$$\text{DTFT}\{x[n]cos(\omega_c n)\} = \text{DTFT}\left\{\frac{1}{2}e^{j\omega_c n}x[n] + \frac{1}{2}e^{-j\omega_c n}x[n]\right\} = \frac{1}{2}[X(e^{j(\omega-\omega_c)}) + X(e^{j(\omega+\omega_c)})]$$

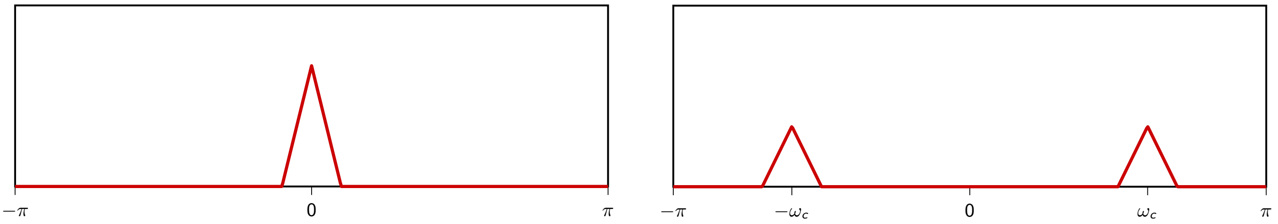


FIGURE 1

2:45 7:06

Example of demodulation

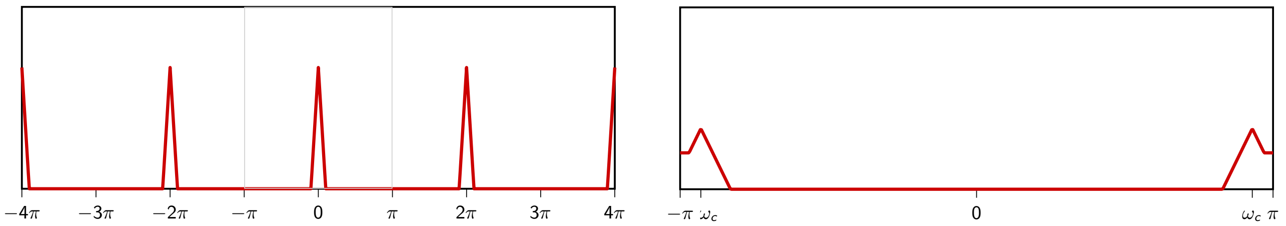


FIGURE 2

3:32 7:06

Example of demodulation with carrier frequency that is too high

3.7.b TUNING A GUITAR

Now we are going to see a real application: tuning your guitar. The abstraction of the problem is that you have reference sinusoid at a frequency ω_0. You have a tunable sinusoid of frequency ω. And we would like to make ω and ω_0 as close as possible, simply by listening to it. What we are going to do here is to beat between these two frequencies once they are close enough. And then by tuning, we can bring this beating to frequency zero: $\omega = \omega_0$. And we have tuned our guitar string with respect to a reference frequency.

First, we bring ω close to ω_0. When these two frequencies are close, we play both sinusoids together. We write $x[n]$ as a sum of both $\cos(\omega_0 n)$ and $\cos(\omega n)$. An error signal will appear: the $\cos(\partial[\omega n])$. When ω is close to ω_0, the error signal occurs at very low frequency. Because it's at such a low frequency, we cannot really hear it. So modulation will it bring up to hearing range. And we'll actually be able to hear it as an oscillation of the carrier frequency.

An electric bass has an E-string frequency of 41.2 Hertz and an A-string frequency of 55 Hertz. We usually use these two harmonics for tuning.

TRISTAN CHORD

Signal of the Day

The Tristan Chord is the first chord in Wagner's opera *Tristan und Isolde*. It first appears in the introductory bars of the opera's prelude. It is dissonant and does not sound pleasing to the ear. We will apply the fast Fourier transform to this chord to analyze its frequency components. From this analysis we see that there are many components: there is an F3, a B3, a D4 sharp, and a G4 sharp, and there appears to be an F4, which is a regular octave, but also B4. Now that we know the frequency components of this chord we can try to synthesize it, using, for example, the Karplus-Strong algorithm we have studied during lectures.

4.1.a LINEAR TIME-INVARIANT FILTERS

Let $y[n] = H\{x[n]\}$, where H is a processing device.

We say that H is linear if the output of a linear combination of inputs is equal to the linear combination of the outputs:

$$H\{ax_1[n] + bx_2[n]\} = aH\{x_1[n]\} + bH\{x_2[n]\}$$

We say that H is time-invariant if the system will behave in exactly the same way independently of when it is switched on:

$$H\{x[n - n_0]\} = y[n - n_0]$$

The system H is causal if it can only have access to input and output values from the past: the output is a linear functional of past values of the input and past values of the output.

4.1.b CONVOLUTION

The impulse response $h[n]$ fully characterizes an LTI system: $h[n] = H\{∂[n]\}$.

As in Module 3.2, we can write:

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

by linearity and an time invariance:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] \quad = x[n] \star h[n]$$

First, we time-reverse the impulse response. And at each step, from –infinity to +infinity, we center the time-reversed impulse response in the current sample n, so as to shift the time-reversed equal response by minus n. And then we compute the inner product between this shifted replica of the impulse response and the input sequence.

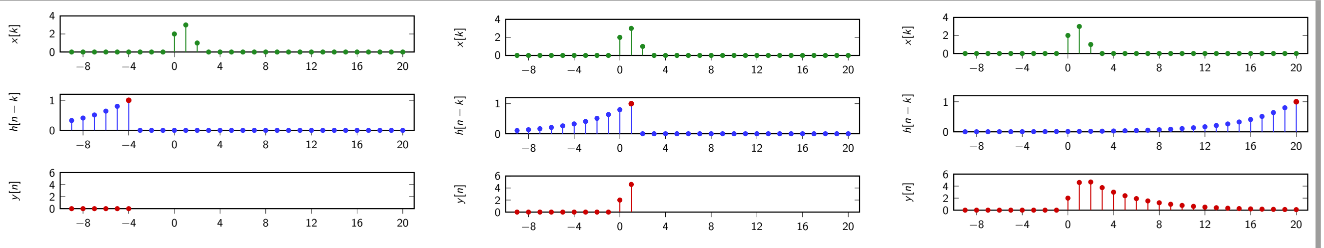


FIGURE 1

4:30 6:40

Example of convolution

Convolution properties:

- linearity and time invariance (by definition)
- commutativity: $(x \star h)[n] = (h \star x)[n]$
- associativity for absolutely- and square-summable sequences: $((x \star h) \star w)[n] = (x \star (h \star w))[n]$

Signal of the Day

CAMERA RESOLUTION AND SPACE EXPLORATION

The Rosetta spacecraft was launched about ten years ago by the European Space Agency with the mission of reaching the comet Churyumov–Gerasimenko. This comet is also known with the more technical name 67P. Rosetta managed to send back to Earth increasingly detailed pictures of the comet as it got progressively closer to it. As Rosetta approached the comet, the quality of the pictures progressively improved. From that example we can see that resolution is not limited by pixel density: resolution is limited by diffraction. You might be surprised to learn that the resolution of Rosetta's camera is only about four megapixels.

4.2.a THE MOVING AVERAGE FILTER

The following equation defines an average:

$$y[n] = \frac{1}{M}\sum_{k=0}^{M-1} x[n-k]$$

This formula can helps us to average a noisy output. The moving average filter is based on this, as $h[n] = H\{\partial[n]\}$:

$$h[n] = \frac{1}{M}\sum_{k=0}^{M-1} \delta[n-k] \quad = \begin{cases} 1/M, & for\ 0 \leq n < M \\ 0, & otherwise \end{cases}$$

The smoothing effect of the filter *i* proportional to the length of the impulse response, *M*. As a consequence, the number of operations and the storage are proportional to *M*.

$$y_M[n] = \frac{1}{M}\sum_{k=0}^{M-1} x[n-k]$$

$$y_{M-1}[n] = \frac{1}{M-1}\sum_{k=0}^{M-1} x[n-k]$$

$$\sum_{k=0}^{M-1} x[n-k] = x[n] + \sum_{k=1}^{M-1} x[n-k]$$

$$My_M[n] = x[n] + (M-1)y_{M-1}[n-1]$$

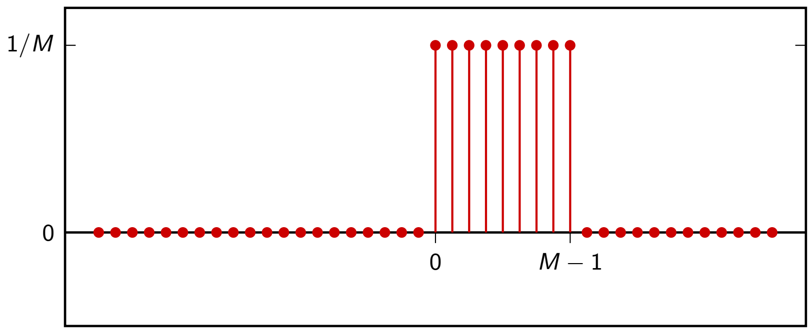


FIGURE 1

3:34

8:05

Plot of MA impulse response

4.2.b THE LEAKY INTEGRATOR

As we saw earlier, the leaky integrator is defined as follows:

$$y[n] = \lambda \cdot y[n-1] + (1-\lambda)x[n]$$

Let's compute its impulse response:

$$h[n] = \lambda \cdot h[n-1] + (1-\lambda)\partial[n] \Rightarrow h[n] = \lambda^n \cdot (1-\lambda) \cdot u[n]$$

To prevent explosion, we always choose $\lambda < 1$.

$$h[n] = (1-\lambda)\lambda^n\, u[n]$$

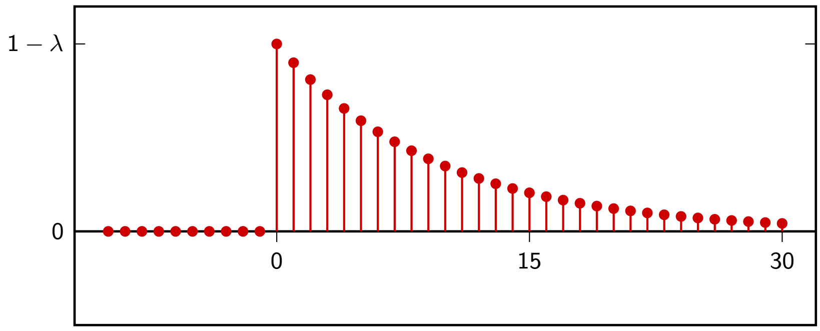


FIGURE 1

3:46 5:19

Plot of leaky integrator impulse response

4.3.a FILTER CLASSIFICATION IN THE TIME DOMAIN

According to the shape of its impulse response, we can label a filter as belonging to one of the following categories:

– Finite Impulse Responses (FIR): their impulse response has a finite support, only a finite number of samples are involved in the computation of each output sample, e.g., moving average filter.
– Infinite Impulse Responses (IIR): their impulse response has an infinite support, hence potentially an infinite number of samples involved in the computation of each output sample, e.g., leaky integrator filter.
– Causal: their impulse response is zero for $n < 0$, only past samples are involved in the computation, e.g., moving average filter.
– Noncausal: their impulse response is nonzero for some $n < 0$, e.g., zero-centered moving average filter.

4.3.b FILTER STABILITY

Stability guarantees that the system will not behave unexpectedly if the input of the system is well-behaved, i.e., a bounded input. The concept of Bounded-Input Bounded-Output (BIBO) Stability requires that a system produces a bounded output when the input is bounded.

FIR filters are always stable, because their impulse response only contains a finite number of nonzero values, and therefore the sum of their absolute values will always be finite. On the other hand, when it comes to IIR filters, we have to explicitly check for stability.

4.4.a THE CONVOLUTION THEOREM

Let *H* be an LTI system:

$$\begin{aligned} y[n] &= e^{j\omega_0 n} \star h[n] \\ &= h[n] \star e^{j\omega_0 n} \\ &= \sum_{k=-\infty}^{\infty} h[k] e^{j\omega_0 (n-k)} \\ &= e^{j\omega_0 n} \sum_{k=-\infty}^{\infty} h[k] e^{-j\omega_0 k} \\ &= H(e^{j\omega_0}) e^{j\omega_0 n} \end{aligned}$$

LTI filters cannot change the frequency of sinusoids. The DTFT of the impulse response fully determines the frequency characteristic of a filter.

The convolution theorem is defined as follows:

$$\begin{aligned} DTFT\{x[n] \star h[n]\} &= \sum_{n=-\infty}^{\infty} (x \star h)[n] e^{-j\omega n} \\ &= \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x[k] h[n-k] e^{-j\omega n} \\ &= \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x[k] h[n-k] e^{-j\omega (n-k)} e^{-j\omega k} \\ &= \sum_{k=-\infty}^{\infty} x[k] e^{-j\omega k} \sum_{n=-\infty}^{\infty} h[n-k] e^{-j\omega (n-k)} \\ &= H(e^{j\omega}) X(e^{j\omega}) \end{aligned} \tag{9}$$

The frequency response $H(e^{j\omega}) = DTFT\{h[n]\}$ has two effects:

- magnitude: amplification ($|H(e^{j\omega})| > 1$) or attenuation ($|H(e^{j\omega})| < 1$) of input frequencies
- phase: overall delay and shape changes

4.4.b EXAMPLES OF FREQUENCY RESPONSE

In general, if $H(e^{j\omega}) = A(e^{j\omega})e^{-j\omega d}$, it means that the filter operates by combining the action of a zero phase. Therefore a zero delay component only affects the magnitude of the input, followed by a delay of these samples.

$$|H(e^{j\omega})| = \frac{1}{M}\left|\frac{\sin(\frac{\omega}{2}M)}{\sin(\frac{\omega}{2})}\right|$$

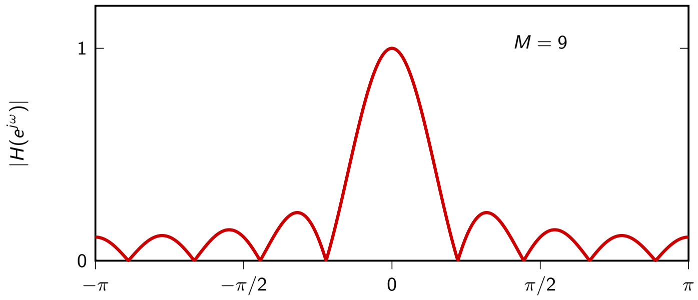
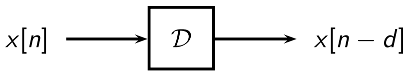


FIGURE 1

0:26 14:13

Magnitude of MA DTFT



- $y[n] = x[n-d]$
- $Y(e^{j\omega}) = e^{-j\omega d}X(e^{j\omega})$
- $H(e^{j\omega}) = e^{-j\omega d}$
- linear phase term

FIGURE 2

6:30 14:13

Linear-phase term

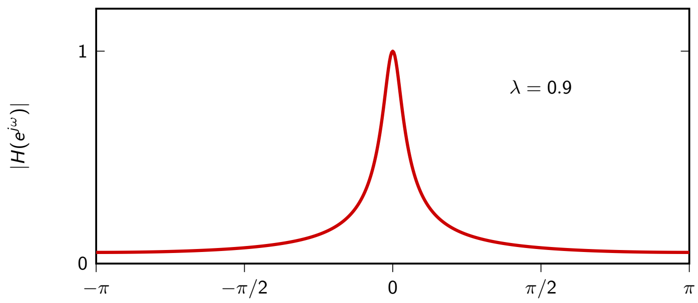


FIGURE 3

9:49 14:13

Magnitude response of the leaky integrator

4.5.a FILTER CLASSIFICATION IN THE FREQUENCY DOMAIN

We can classify filters according to four broad categories based on the shape of their magnitude response:
- Lowpass filters are filters that let the low frequencies live and kill everything else.
- Highpass filters do the opposite. They let high frequencies go through and they kill low frequencies, in particular frequencies around zero.
- Bandpass filters only let a band of frequencies go through in the middle of the frequency band.
- Allpass filters are filters for which the magnitude is a constant over the entire frequency band.

Filters can also be classified by their phase.

4.5.b THE IDEAL LOWPASS FILTER

The magnitude of the ideal lowpass filter would be 1 over the passband, a perfectly flat passband, and identically 0 over the stopband and an infinite attenuation over the stopband. To make matters even better we would require the magnitude response to be a real function so that the filter has zero phase and therefore introduces no delay.

$$\begin{aligned} h[n] &= IDFT\{H(e^{j\omega})\} \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{j\omega})e^{j\omega n} \\ &= \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} e^{j\omega n} \\ &= \frac{1}{\pi n}\frac{e^{j\omega_c n} - e^{-j\omega_c n}}{2j} \\ &= \frac{sin\ \omega_c n}{\pi n} \end{aligned}$$

$$H(e^{j\omega}) = \begin{cases} 1 \ for|\omega| < \omega_c \\ 0 \ otherwise \end{cases}$$

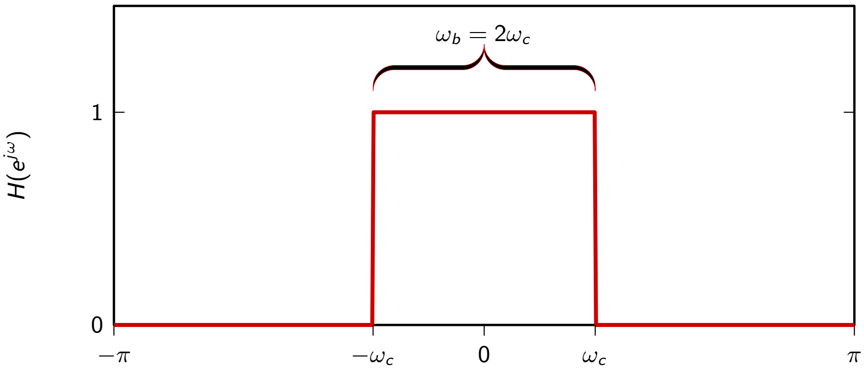


FIGURE 1

0:55 6:50

Ideal lowpass filter (0:55/6:50) (ω_b bandwidth and ω_c cutoff frequency)

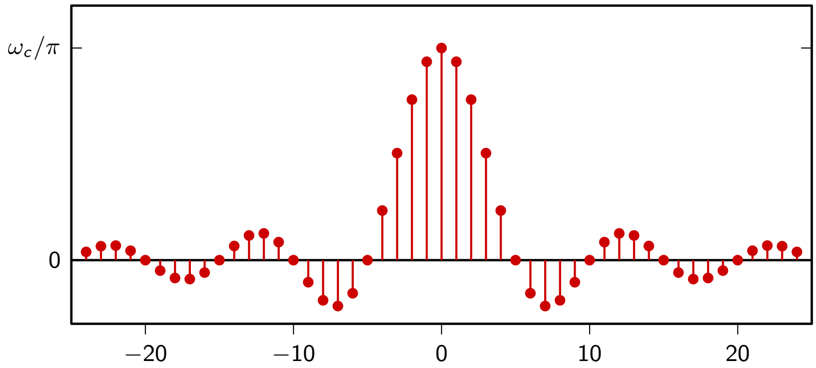


FIGURE 2

2:56 6:50

Ideal impulse response

This impulse response results in a problem, as it has an infinite support in both directions, which means that it cannot be implemented in practice. Also, as it decays very slowly in time, it requires too many samples.

NB: It is important to be familiar with the rect-sinc pair:

The sinc-rect pair:

$$rect(x) = \begin{cases} 1, & |x| \leq 1/2 \\ 0, & |x| > 1/2 \end{cases}$$

$$sinc(x) = \begin{cases} \frac{\sin(\pi x)}{\pi x}, & x \neq 0 \\ 1, & x = 0 \end{cases}$$

Note that $sinc(x) = 0$ when x is a nonzero integer.

The sinc function is not absolutely summable, which means that the ideal filter is not BIBO stable.

4.5.c IDEAL FILTERS DERIVED FROM THE IDEAL LOWPASS FILTER

The highpass filter is defined as follows:

$$H_{hp}(e^{j\omega}) = \begin{cases} 1 \ for \ \pi \geq |\omega| \geq \omega_c \\ 0 \ otherwise, \ 2\pi - periodicity \ implicit \end{cases}$$

$$H_{hp}(e^{j\omega}) = 1 - H_{lp}(e^{j\omega})$$

$$h_{hp}[n] = \delta[n] - \frac{\omega_c}{\pi} sinc(\frac{\omega_c}{\pi} n)$$

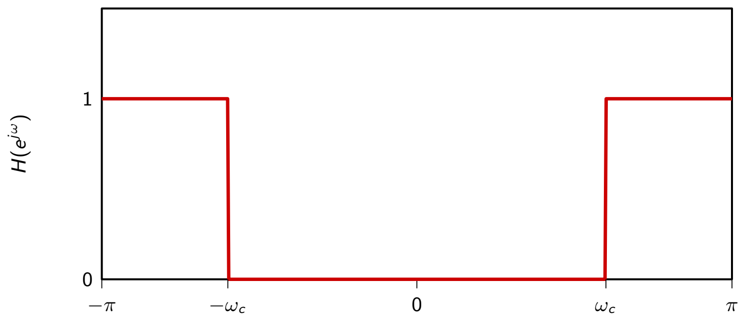


FIGURE 1

0:15 2:34

The ideal highpass filter of cutoff frequency ω_c

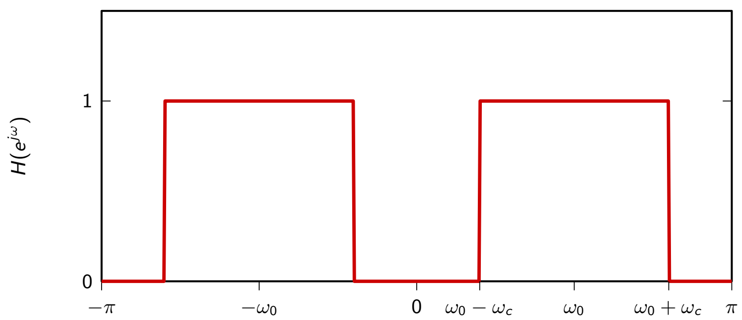


FIGURE 2

1:15 2:34

Bandpass filter of carrier frequency ω_0

The bandpass filter is actually a demodulation of the lowpass filter seen previously, meaning the following:

$$H_{bp}(e^{j\omega}) = \begin{cases} 1 \ for \ |\omega \pm \omega_0| \leq \omega_c \\ 0 \ otherwise, \ (2\pi - periodicity \ implicit) \end{cases}$$

$$h_{bp}[n] = 2cos(\omega_0 n) \frac{\omega_c}{\pi} sinc(\frac{\omega_c}{\pi} n)$$

4.5.d DEMODULATION REVISITED

In the previous module, we applied sinusoidal modulation to an input signal $x[n]$ to obtain $y[n]$, which is equal to $x[n]\cdot\cos(w_0 n)$. And we tried to demodulate the modulated signal by multiplying the modulator signal, again by the carrier, $\cos(w_0 n)$. And we found that the demodulated signal contained unwanted high frequency components that, at the time, we did not know how to remove.

Now we can modulate it, obtain two copies at half the amplitude, multiply it again by $\cos(w_0 n)$, and apply a lowpass filter as follows in figure 1.

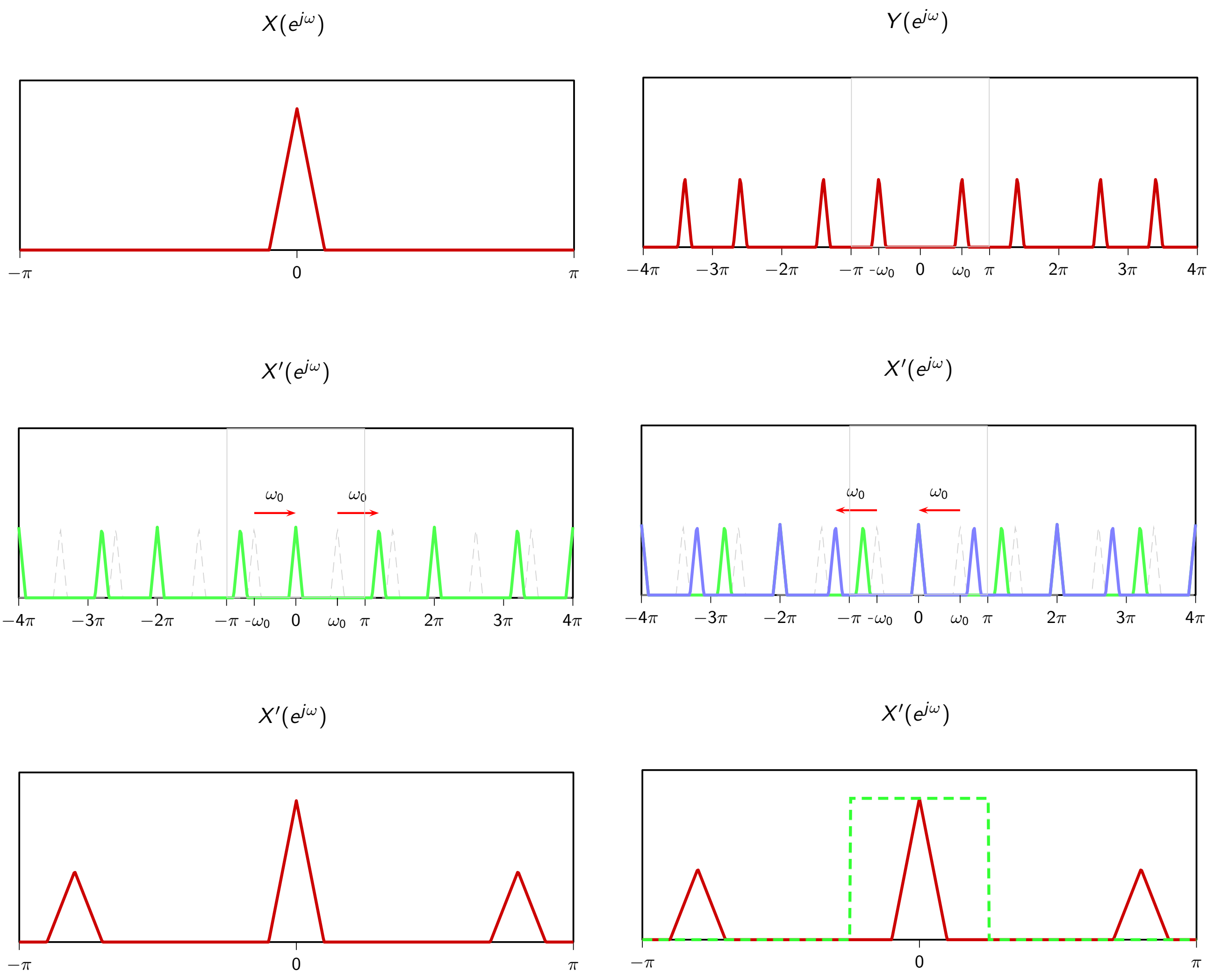


FIGURE 1

0:10 2:00 2:16

Revisited demodulation

4.6.a IMPULSE TRUNCATION AND GIBBS PHENOMENON

he idea is that we could approximate an ideal lowpass filter by truncating the impulse response to make it a finite support impulse response.

FIR approximation of lenght M = 2N+1:

$$\hat{h}[n] = \begin{cases} \frac{\omega_c}{\pi} sinc(\frac{\omega_c}{\pi} n), & |n| \leq N \\ 0, \text{otherwise} \end{cases}$$

It could seem to be a good idea as the mean square error is minimized:

$$\begin{aligned} MSE &= \frac{1}{2\pi} \int_{-\pi}^{\pi} |H(e^{j\omega}) - \hat{H}(e^{j\omega})|^2 d\omega \\ &= ||H(e^{j\omega}) - \hat{H}(e^{j\omega})||^2 \\ &= ||h[n] - \hat{h}[n]||^2 \\ &= \sum_{n=-\infty}^{\infty} |h[n] - \hat{h}[n]|^2 \end{aligned}$$

But the reason why this would not be a good idea is because it looks as though the maximum error near the transition point for the frequency response never really goes down in spite of the number of points that we used for the approximation. We can prove mathematically that the maximum error around the cutoff frequency is about 9% of the height of the jump, regardless of the number of points. This is known as the Gibbs phenomenon.

IS IT POSSIBLE TO HEAR THE SHAPE OF A ROOM?

Signal of the Day

We will investigate how sound propagates in a room between a sender, or a loud speaker, *s*, and the receiver, or a microphone, *r*. Every time a sound hits the wall, part of the sound is reflected in such a way that the incident angle equals the reflection angle. These angles are defined with respect to the normal of the walls. Sound propagation is a fairly complex phenomenon involving the superposition of a signal and its different echoes, and means that the derivation of a model might not be very simple. Nevertheless, in room acoustics, sound levels are low, and so the linear model is a good approximation. The question is whether we can reconstruct the shape of a room using only echoes. We experimented with this in our lab for two different shapes of rooms and obtained quite accurate results.

4.6.b WINDOW METHOD

The impulse truncation can be interpreted as the product of the ideal filter response and a rectangular window of *N* points. From the modulation theorem, the DTFT of the product of two signals is equivalent to the convolution of their DTFTs:

$$\begin{aligned}
\text{IDTFT}\{(X \star Y)(e^{j\omega})\} &= \frac{1}{2\pi}\int_{-\pi}^{\pi}(X \star Y)(e^{j\omega})e^{j\omega n}d\omega \\
&= \frac{1}{(2\pi)^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}X(e^{j\sigma})Y(e^{j(\omega-\sigma)})e^{j\omega n}d\sigma d\omega \\
&= \frac{1}{(2\pi)^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}X(e^{j\sigma})Y(e^{j(\omega-\sigma)})e^{j\sigma n}e^{j(\omega-\sigma)n}d\sigma d\omega \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi}X(e^{j\sigma})e^{j\sigma n}d\sigma\frac{1}{2\pi}\int_{-\pi}^{\pi}Y(e^{j(\omega-\sigma)})e^{j(\omega-\sigma)n}d\omega \\
&= x[n]y[n]
\end{aligned}$$

Hence, the choice of window influences the quality of the approximation results. The window method is just a generalization of the impulse truncation method, where we use a different window shape. For example, by using a triangular window, we reduce the Gibbs error at the price of a longer transition.

4.6.c FREQUENCY SAMPLING

The idea here is to say, what if we draw the desired frequency response in the frequency domain. Then we sample this frequency response at regularly spaced intervals and calculate the IDFT of these values – the inverse DFT. We can always do this for a finite set of frequency points and use the result as an *M*-tap impulse response hat *h*[*n*].

The interpolator turns out to be, once again, the transform of an *N*-tap rectangular window. So we're not really escaping from the indicator function that we used in the impulse truncation method. And, because of that, we have no control over mainlobe and sidelobes of the interpolator.

These are good methods to be familiar with in order to quickly try something out when we are faced with a filtering problem. But they are definitely not optimal and they leave a lot be desired in terms of the fine control that we can have over the maximum error.

4.7.a THE Z-TRANSFORM

As ideal filters cannot be implemented, we want to figure out what is the most realizable LTI transformation. How do we compute the frequency response of such an equation?

$$\sum_{k=0}^{N-1} a_k y[n-k] = \sum_{k=0}^{M-1} b_k x[n-k]$$

The tool we will use is called the *z*-transform. *z*-transform is a formal operator that maps the discrete type sequence *x*(*n*) onto a function of the complex variable *z*, defined as follows:

$$X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n}, \ z \in \mathbb{C}$$

Whenever we evaluate the *z*-transform in $z = e^{jw}$ it becomes the DTFT. Two key properties of the *z*-transform are linearity and time-shifting:

1. linearity: $Z\{\alpha x[n] + \beta y[n]\} = \alpha X(z) + \beta Y(z)$
2. time-shift: $Z\{x[n-N]\} = z^{-N} X(z)$

Also:

$$\sum_{k=0}^{N-1} a_k y[n-k] = \sum_{k=0}^{M-1} b_k x[n-k]$$

$$Y(z) \sum_{k=0}^{N-1} a_k z^{-k} = X(z) \sum_{k=0}^{M-1} b_k z^{-k}$$

$$Y(z) = H(z) X(z)$$

4.7.b REGION OF CONVERGENCE AND STABILITY

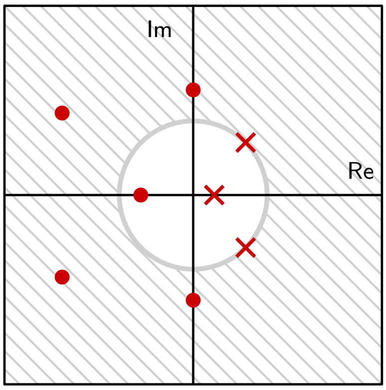


FIGURE 1
3:20 7:57

Example of a ROC (causal system)

As the *z*-transform is a power-series, convergence is always absolute. The region of convergence (ROC) of a *z*-transform is the set of points on the complex plane for which it exists. The ROC is the whole complex plane for finite support sequences; it has circular symmetry, cannot include poles and causal sequences extend from a circle to infinity.

It will be defined by the transfer function:

Consider the transfer function for an LTI system:

$$H(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b_{M-1} z^{-(M-1)}}{a_0 + a_1 z^{-1} + \ldots + a_{N-1} z^{-(N-1)}}$$

It can always be factored by:

$$H(z) = C \frac{\Pi_{n=1}^{M-1}(1 - z_n z^{-1})}{\Pi_{n=1}^{M-1}(1 - p_n z^{-1})}$$

From the definition of the transfer function we can obtain the following:
– The zeros (resp. the poles) are the roots of the numerator (resp. denominator) of the rational transfer function.
– The region of convergence is only determined by the magnitude of the poles.
– The *z*-transform of a causal LTI system extends outwards from the largest magnitude pole.
– An LTI system is stable if its region of convergence includes the unit circle.

This last condition in particular offers a simple method for studying the stability of LTI systems.

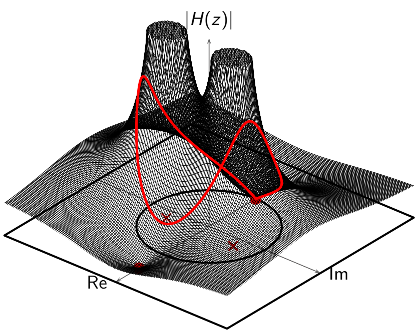


FIGURE 2
5:31 7:57

Estimating frequency response from pole-zero plot (glue zeros, push poles)

Consider a filter with impulse response *h*[*n*]:

- BIBO stability $\Leftrightarrow \sum_{n=-\infty}^{\infty} |h[n]| < \infty$ (Module 5.3)
- $H(z)|_{|z|=1} < \infty \Leftrightarrow \sum_{n=-\infty}^{\infty} |h[n]| < \infty$ (absolute convergence of z-transform)
  system is stable if and only if ROC includes the unit circle

The shape of the frequency response will be the level curve computed around the unit circle of the rubber sheet.

4.8 INTUITIVE IIR DESIGN

We can derive simple filters from known structures. In general, with low order systems, it is possible to analyze and predict their characteristics from their pole zero plot. Furthermore, to obtain a real filter, poles and zeros are added in complex conjugate pairs. More especially, we have studied the following filters.

- The resonator that selects one specific frequency $\omega 0$. It is derived by shifting the passband of a leaky integrator to ω_0 (fig. 1).
- The DC (Direct Current) notch that removes the direct current component. It contains a zero at z = 1 (i.e. $\omega = 0$) and a pole at $\lambda < 1$. DTFT equal to 0 at $\omega = 0$ (fig. 2).
- The hum removal that removes a specific frequency ω_0. It is a DC notch whose stopband is shifted to a particular frequency ω_0 (fig. 3).

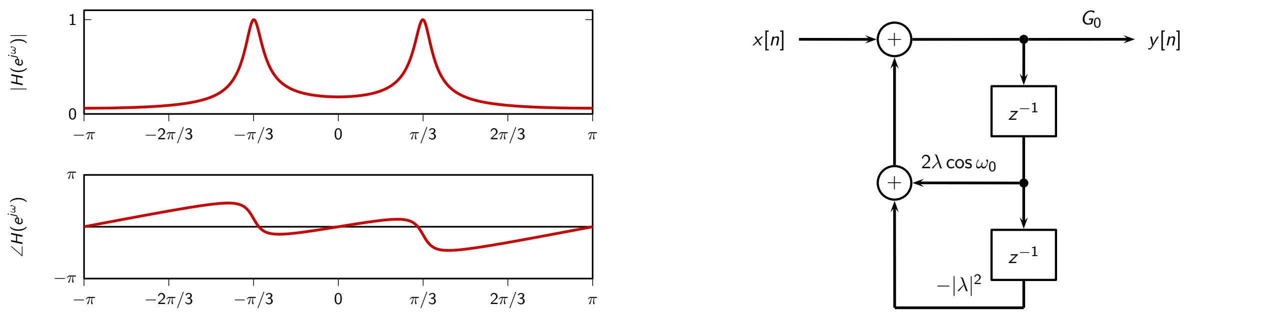


FIGURE 1

3:55 10:03

Frequency response of a resonator and filter structure

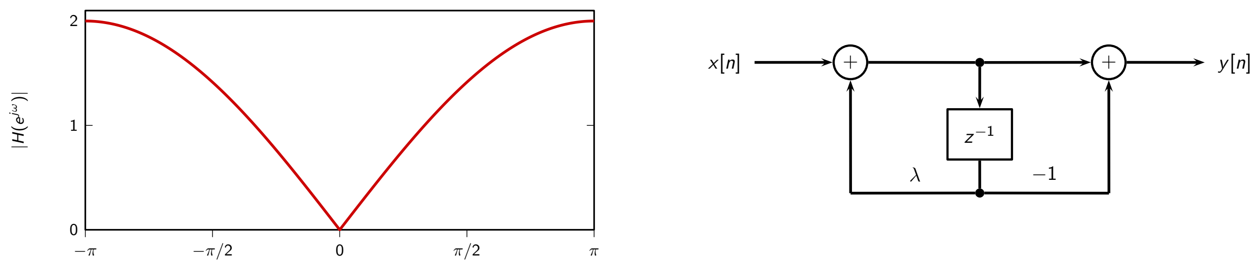


FIGURE 2

6:30 10:03

Frequency response and circuit of DC notch

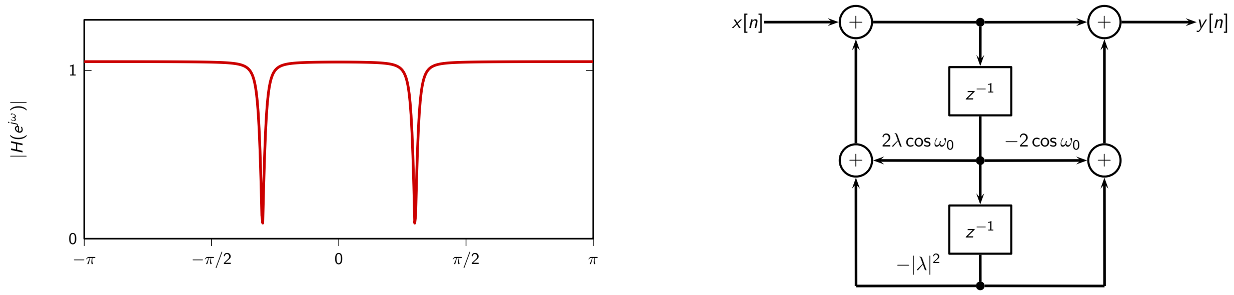


FIGURE 3

9:30 10:03

Hum removal frequency response and circuit

4.9.a FILTER SPECIFICATION

A filter design problem starts from a list of filter requirements:
– What is the desired frequency response of the filter, what are the passband(s) and stopband(s)?
– What are the desired phase characteristics?
– What are the limits in terms of computation and/or precision?

We are interested in realizable filters, therefore our final design will be expressed in the form of a rational transfer function.

$$H(z) = \frac{b_0 + b_1 z^{-1} + ... + b_{M-1} z^{-(M-1)}}{a_0 + a_1 z^{-1} + ... + a_{N-1} z^{-(N-1)}}$$

Our problem is to find the degree of the numerator and of the denominator as well as the coefficients of the polynomials involved in order to best fulfill the requirements. In general, if we want very small transmission bands, we need to use a filter of high order. And similarly, if we want small error tolerances, we need a high order filter. We also must take into account the fact that a high filter order implies more computation and hence a larger delay.

To depart from ideal filter characteristics we must also specify:
– The transition band: range of frequency between the passband and the stopband where the filter impulse response transitions from the passband to the stopband.
– The tolerance band for the error in the passband and stopband. An equiripple filter is where the error oscillates within the transition bands equally.

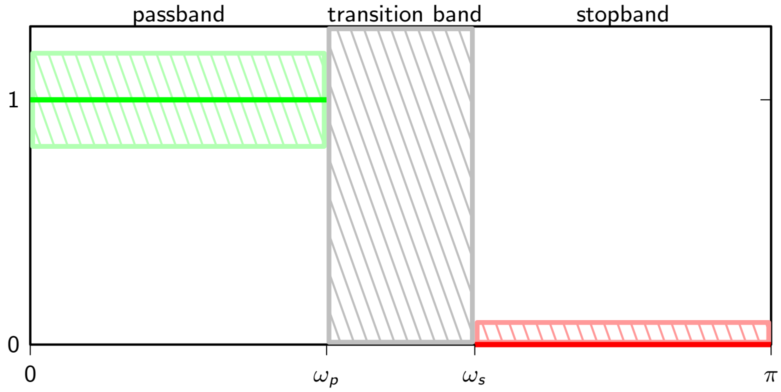


FIGURE 1

2:40

6:24

Example of realistic lowpass tolerance

An important case is what we call the equiripple error, where the error, in this case in the passband, oscillates between a maximum and a minimum, and the local extrema of the frequency response coincides with the upper and lower limit of the tolerance region.

4.9.b IIR DESIGN

IIR design are translations from known analog design. The routines for calculating them are available in most DSP numerical packages. From the specifications, a first filter is computed with a certain order N. Then the filter is tested to verify that it meets the desired requirements. If it is not the case, the procedure is run again with a higher value of the order 4.9.c FIR design.

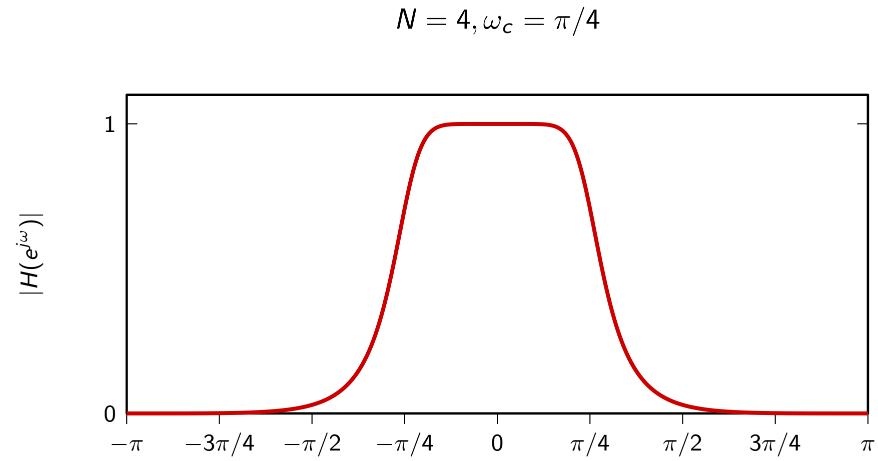


FIGURE 1

1:55 3:53

Butterworth lowpass

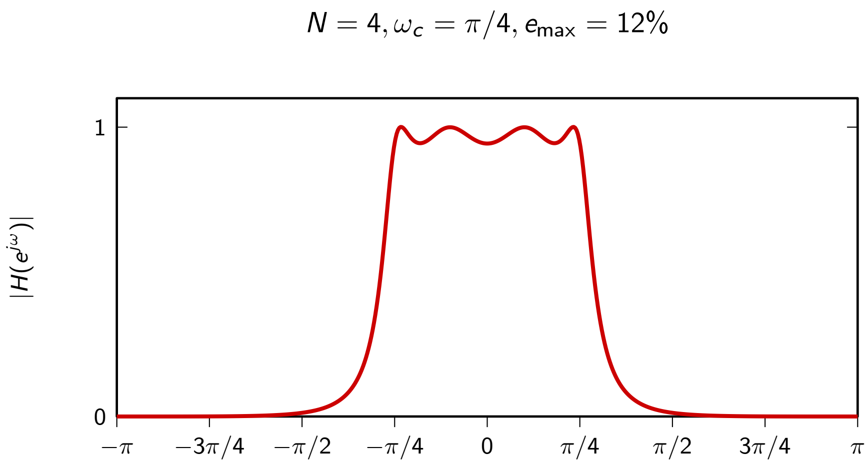


FIGURE 2

2:37 3:53

Chebychev lowpass

4.9.c FIR DESIGN

FIR design only exists in discrete-time. In FIR filter design, we only have to determine the coefficients of one polynomial. In the 1970s, Parks and McClellan devised a procedure to compute an optimal linear phase and equiripple error both in stopband and passband. The algorithm proceeds by minimizing the maximum error in the stopband and the passband. FIR are calculated from a list of specifications (frequency response and ratio of maximum equiripple error between the passband and the stopband). Linearity of the phase is achieved by designing an impulse response, which is either symmetric or antisymmetric. This procedure is optimal in the minimax sense as it minimizes the maximum error.

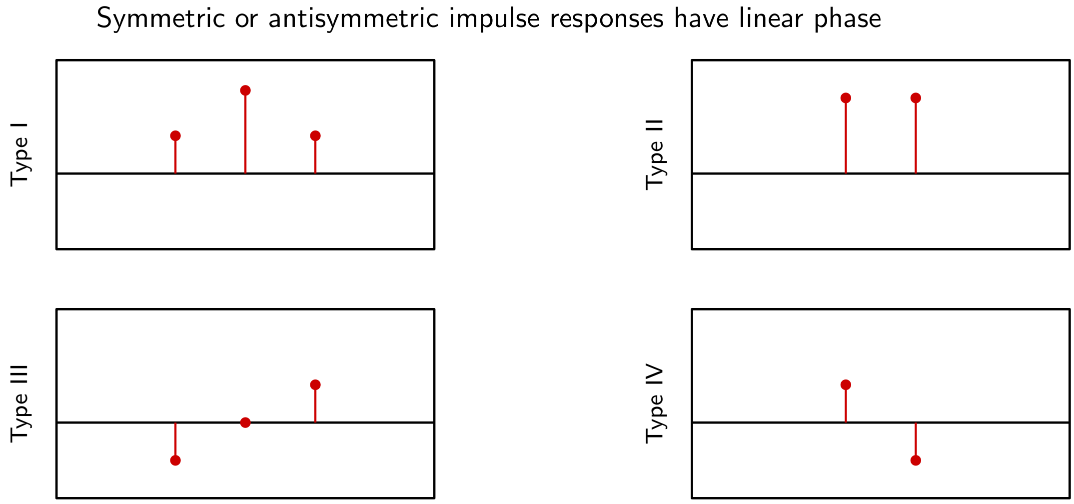


FIGURE 1

0:56 7:56

Four types of filters (odd/even length, symmetric/antisymmetric)

A Type I filter is an odd length symmetric FIR, where the taps are symmetric around an index. Type III filters have an odd number of taps; they are antisymmetric around the center tap, which imposes, of course, the zero center type. Type II and Type IV filters are symmetric and antisymmetric filters, respectively, both of which have an even number of taps.

A good way to compare the performance of different filters is to express their magnitude in decibels. Let G be the maximum magnitude in the passband, the attenuation of the filter expressed in decibels is: $A_{dB} = 20 \log_{10} (|H(e^{jw})|/G)$

Any of these designs can be used to obtain lowpass, bandpass, and highpass filters. Optimal FIR bandpass and highpass filters can be designed directly with the Parks-McClellan algorithm. You can also design optimal FIR with piecewise linear magnitude responses.

5.1.a THE CONTINUOUS-TIME PARADIGM

During this course, we look at two different views: the analog and digital view. Some examples of applications in digital worldview are arithmetics, combinatorics, computer science, and digital signal processing, while some examples of applications of the analog worldview are calculus, distributions, system theory, and continuous-time electronics. Related to our course we have the following differences between the two:

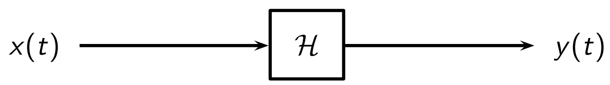
DIGITAL WORLDVIEW	ANALOG WORLDVIEW
countable integer index n	real-valued time t (sec)
sequences $x[n] \in l_2(\mathbb{Z})$	functions $x(t) \in L_2(\mathbb{R})$
frequency $\omega \in [-\pi, \pi]$	frequency $\Omega \in \mathbb{R}$ (rad/sec)
DTFT: $l_2(\mathbb{Z}) \rightarrow L_2[-\pi, \pi]$	FT: $L_2(\mathbb{R}) \rightarrow L_2(\mathbb{R})$

Digital worldview versus analog worldview

How do we bridge the gap between those two worlds? The relationship between going from continuous time $x(t)$ to x as a sequence is called *sampling*. From x the sequence to $x(t)$ is called *interpolation*. And we have to be able to clearly understand at what point we can go from one to the other, when these two are tightly related, and when they are not faithful images of each other.

5.1.b CONTINUOUS-TIME SIGNAL PROCESSING

We would like $x(t)$, our signal, to be a complex function of a real variable and of finite energy ($\in L_2(\mathbb{R})$). Let us define the inner product of two complex signals: $\langle x(t), y(t) \rangle = \int x^*(t)y(t)dt$, hence the energy of a signal: $\|x(t)\|^2 = \langle x(t), x(t) \rangle$. As discrete-time filters exist, we will now study continuous-time filters.



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

FIGURE 1

0:59 5:50

Analog LTI filters

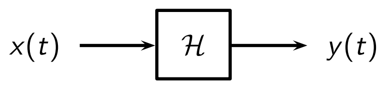
Note the $-\tau$ which corresponds to a time-reversal when computing the integral. Regarding Fourier analysis, there is no maximum frequency Ω. The corresponding formula is thus given by:

$$X(j\Omega) = \int_{-\infty}^{\infty} x(t)e^{-j\Omega t}dt \leftarrow \text{ not periodic!}$$

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

This real-world frequency Ω is important: it is expressed in rad/sec which we can convert to Hertz using the frequency F computed as $F = \Omega/2\pi$, the period of the signal will thus be given by $T = 1/F = 2\pi/\Omega$.

The convolution theorem in discrete time also holds for continuous time: the convolution of two signals in the time domain corresponds to their multiplication in the Fourier domain.



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

FIGURE 2

3:30 5:50

Convolution theorem for continuous-time signals

Let us now introduce a new concept: band-limitedness. It means that the Fourier transform of a function x that is band-limited is such that: $X(j\Omega) = 0$, for $|\Omega| > \Omega_N$.

5.2.a POLYNOMIAL INTERPOLATION

The interpolation problem seeks to fill the gaps between two samples: how should we do this? First of all, we have several requirements for interpolation: we have to decide on the value T_s which is the space between two samples, make sure that $x(nT_s) = x[n]$, and make sure that $x(t)$ is smooth. In general, we would like interpolators to be infinitely differentiable for them to make sense physically. A natural solution for this would thus be polynomial interpolation. For N point, we will obtain a polynomial of degree N–1. Let $I_N = [-N, \ldots, N]$ and P_N be the space of degree-2N polynomials over I_N. A basis for P_N is the family of $2N + 1$ Lagrange polynomials *s.t.*:

$$L_n^{(N)}(t) = \Pi_{k=-N, \neq n}^{N} \frac{t-k}{n-k}, \; n = -N, \cdots, \; N$$

Now we can express the Lagrange interpolation to be:

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

and the Lagrangian interpolator satisfies $p(n) = x[n]$ for $-N \leq n \leq N$ since

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

Signal of the Day

FUKUSHIMA

On March 11, 2011, after an earthquake and several tsunamis had devastated the northeast coast of Japan, a multiple core meltdown happened at the Fukushima Daiichi Nuclear Power plant, leading to a substantial release of radioactive material into the environment.

Due to a general lack of detailed radiation measurement, SafeCast, a citizen science group based in Japan, started building instruments for geo-localized measurements of radioactivity. The idea behind the project was to reconstruct a map of radioactivity from these measurements. Instruments were loaned to volunteers in the field and measurements were carried out by attaching the sensors to cars. The sensors would then take a measurement of the radiation every five seconds. Therefore, sampling was generally limited to areas accessible by car. Fairly high measurement levels were observed in some places. This is a typical example where the location of measurements is constrained by available infrastructure. Since the measurements were done by car, it was only possible to measure on roads. It would then be necessary to apply interpolation to get an estimate of the intensity of radiation in areas where no samples were available, such as in forests and fields. The function so obtained is then resampled on the regular grid and displayed. A decent estimate of the total radiation field is obtained in this way.

5.2.b LOCAL INTERPOLATION

Now, let us assume that infinite differentiability is no longer a requirement. The zero-order interpolation is the approximation by rect-function:

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

The interpolation kernel is thus rect(t). Even if it holds, there are discontinuities at some points.

The next type of interpolation is the first-order piece-wise linear interpolation that consists in drawing straight lines between the samples. It is called the "connect the dots" strategy:

$$x(t) = \sum_{n=-N}^{N} x[n]i_1(t-n)$$

where the interpolator kernel is:

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

This interpolation is continuous but its derivative is not.

The third-order interpolator is more interesting as its kernel is composed of two cubic polynomials making it continuous up to its second derivative.

So basically, a common scheme appears in local interpolation based on its kernel i_c:

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

One of the key properties of local interpolation is that the same interpolating function is used independently of N but the main drawback is the lack of smoothness.

In the limit, local and global interpolation are actually the same as:

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

A very elegant and powerful formula is thus (the sinc interpolation formula):

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

5.3.a THE SPECTRUM OF INTERPOLATED SIGNALS

The ingredients of a sinc interpolation are a discrete-time signal $x[n]$ with DTFT $X(e^{j\omega})$, an interpolation interval T_s, and the sinc function properly scaled to have zero crossings at multiples of T_s. The result will be a smooth, continuous signal $x(t)$. What is its spectrum? Here are some key facts about the sinc:

$$\phi(t) = sinc(\frac{t}{T_s}) \leftrightarrow \Phi(j\Omega) = \frac{\pi}{\Omega_N} rect(\frac{\Omega}{2\Omega_N}) \quad T_s = \frac{\pi}{\Omega_N}$$

We can derive the spectrum of x as follows:

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

If the samples spread apart, the function becomes slower. If the samples are closer together, the signal becomes faster. So when you pick the interpolation period T_s, the interpolated signal has only got a band-limited spectrum where Ω_N is π/T_s.

5.3.b THE SPACE OF BAND-LIMITED FUNCTIONS

General case derivations can be found in the book, but for now, let us assume that we will proceed with $T_s = 1$ and $\Omega_N = \pi$. The space of π-band-limited functions is a Hilbert space. The set of functions $\phi^{(n)}(t) = \text{sinc}(t - n)$ forms a basis for the space. If $x(t)$ is π-band-limited, then the sequence $x[n] = x(n)$ is a sufficient representation, i.e., if we have the samples $x[n]$, we can perfectly reconstruct $x(t)$, which is the essence of the sampling theorem. Let us now prove that the sinc function and its shifts by integer are the orthonormal basis for the π-band-limited space:

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

$$\begin{aligned}
< \phi^{(n)}(t), \phi^{(m)}(t) > &=< \phi^{(0)}(t-n), \phi^{(0)}(t-m) > \\
&=< \phi^{(0)}(t-n), \phi^{(0)}(m-t) > \\
&= \int_{-\infty}^{\infty} sinc(t-n) sinc(m-t) dt \\
&= \int_{-\infty}^{\infty} sinc(\tau) sinc((m-n)-\tau) d\tau \\
&= (sinc \star sinc)(m-n)
\end{aligned}$$

Now, using the convolution theorem we get:

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

$$\begin{aligned}
(sinc \star sinc)(m-n) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} [rect(\frac{\Omega}{2\pi})]^2 e^{j\Omega(m-n)} d\Omega \\
&= \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{j\Omega(m-n)} d\Omega \\
&= \begin{cases} 1 \ for \ m = n \\ 0 \ otherwise \end{cases}
\end{aligned}$$

5.3.c THE SAMPLING THEOREM

To see sampling as a basis expansion:

$$\text{for any } x(t) \in \pi - BL$$

$$\begin{aligned} < \phi^{(n)}(t), x(t) > &= < sinc(n-t), x(t) > \\ &= (sinc \star x)(n) \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} rect(\frac{\Omega}{2\pi})X(j\Omega)e^{j\Omega n}d\Omega \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} rect(\frac{\Omega}{2\pi})X(j\Omega)e^{j\Omega n}d\Omega \\ &= x(n) \end{aligned}$$

We have now the following formulae:

Analysis formula:

$$x[n] = < sinc(t-n), x(t) >$$

Synthesis formula:

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

Not all band-limited functions are precisely band-limited to [–π,π] – what happens to Ω_N band-limited functions? We will need to rescale those functions by T_s:

Analysis formula:

$$x[n] = < sinc(\frac{t-nT_s}{T_s}), x(t) > = T_s x(nT_s)$$

Synthesis formula:

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

We can now conclude on the sampling theorem: for any Ω_N-band-limited function *x*(*t*), we can define a sequence *x*[*n*] by taking equally-spaced samples every T_s seconds that unequivocally represents this sequence. In the continuous-time domain, if a signal has a maximum frequency of F_N Hertz, it is a sufficient condition to take samples of at least $2F_N$ H to faithfully represent this function.

5.4.a RAW SAMPLING

Raw sampling is almost the same, the difference here is that we do not need to worry about taking the inner product before sampling, we just take $x(t)$ every T_s seconds.

$$x[n] = (\text{sinc}_{T_s} * x)(nT_s)$$

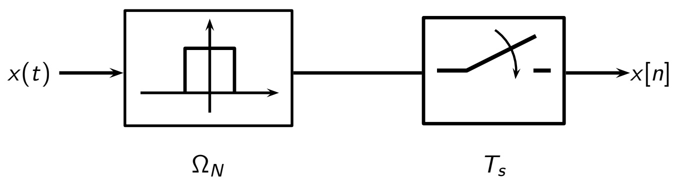


FIGURE 1
0:30 5:47
Sinc sampling block diagram

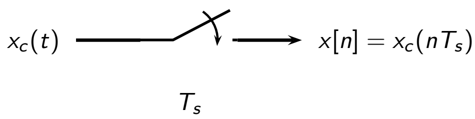
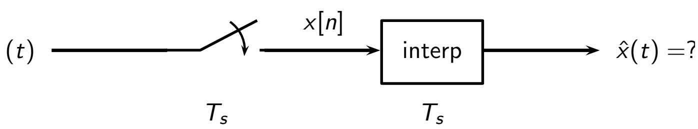


FIGURE 2
0:45 5:47
"Raw" sampling

Now let us analyze the wagon-wheel effect. The continuous-time complex exponential $x(t) = e^{j\Omega 0t}$ is always periodic with $T = 2\pi/\Omega_0$, any angular speed is allowed, and $FT\{e^{j\Omega 0t}\} = 2\pi\partial(\Omega - \Omega_0)$, which is band-limited to Ω_0. Taking snapshots at regular intervals of a rotating point is raw sampling. The resulting digital frequency is $\omega_0 = \Omega_0 T_s\, n$. When $T_s < \pi/\Omega_0$ or $\omega_0 < \pi$, then the rotating point goes the "right"/positive way. When $\pi/\Omega_0 < T_s < 2\pi/\Omega_0$ or $\pi < \omega_0 < 2\pi$, then the phaser advances in big steps, and it seems like it is going the "wrong"/negative way. Finally, when $T_s > 2\pi/\Omega_0$, then the phaser has already done a full rotation added to some other displacement. This effect is called aliasing.



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

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

FIGURE 3
3:45 5:47
Aliasing

5.4.b SINUSOIDAL ALIASING

Now, let us sample a simple sinusoid:

$$x(t) = cos(2_0 t)$$

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

$$F_s = 1/T_s$$

$$\omega_0 = 2\pi(F_0/F_s)$$

If we want to avoid aliasing we will have to make sure the following is satisfied:

sampling frequency	digital frequency	interpolation
$F_s > 2F_0$	$0 < \omega_0 < \pi$	OK: $\hat{F}_0 = F_0$
$F_s = 2F_0$	$\omega_0 = \pi$	OK (max frequency $\hat{F}_0 = F_s$)
$F_0 < F_s < 2F_0$	$\pi < \omega_0 < 2\pi$	negative frequency: $\hat{F}_0 = F_0 - F_s$
$F_s < F_0$	$\omega_0 > 2\pi$	full aliasing: $\hat{F}_0 = F_0 \mod F_s$

FIGURE 1

0:58 4:03

Sampling a sinusoid

5.4.c ALIASING FOR ARBITRARY SPECTRA

What is the spectrum of the raw sampled signals? Let us start by the inverse Fourier transform:

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

Frequencies that are $2\Omega_N$ apart will be aliased:

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

hence with a change of variables:

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

Now, we will define the periodized spectrum:

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

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

Set $\omega = \Omega T_s$:

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

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

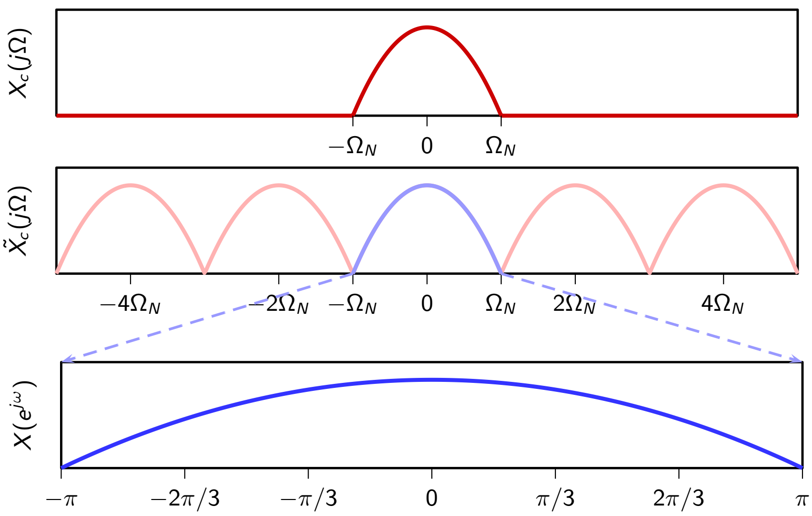


FIGURE 1

7:00 8:25

Example of a band-limited signal to Ω_0 and $\Omega_0 = \Omega_N$

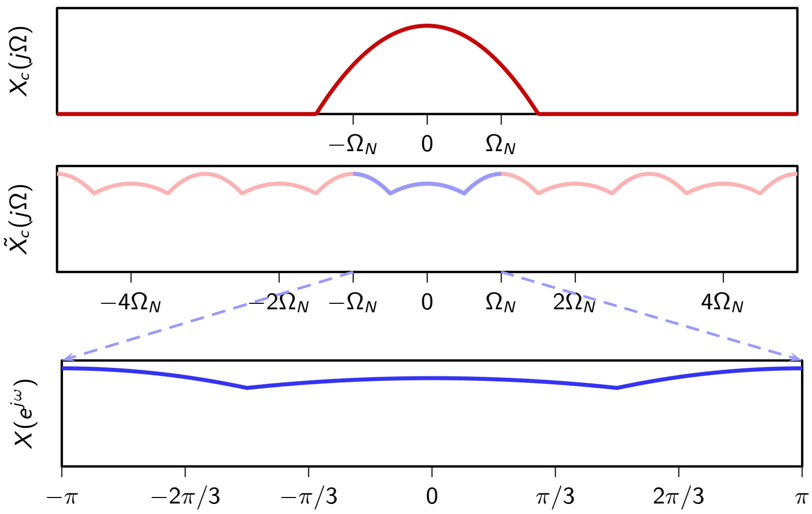


FIGURE 2

7:30 8:25

Example of a band-limited signal to Ω_0 and $\Omega_0 < \Omega_N$ (aliasing)

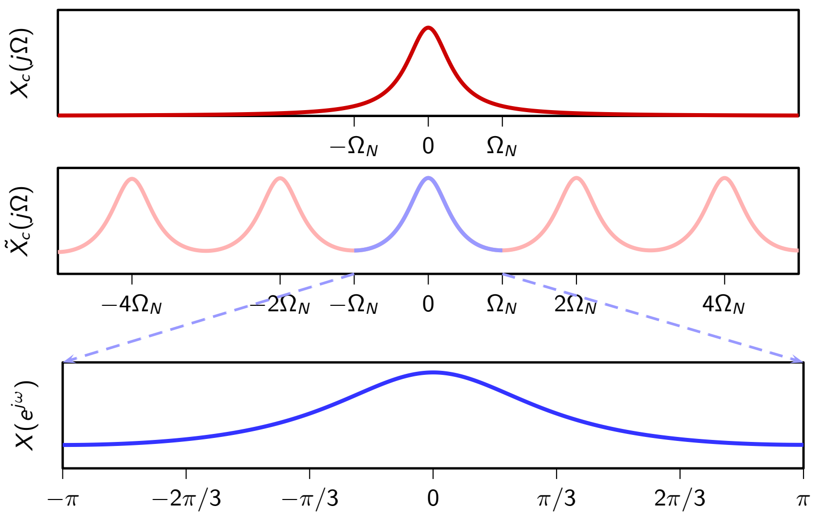


FIGURE 3

8:05 8:25

Example of a non-band-limited signal

5.4.d SAMPLING STRATEGIES

Given a sampling frequency T_s, if the signal is band-limited to π/T_s (or less), then raw sampling is fine, otherwise, we have two choices:

– Band-limit using a lowpass filter in the continuous-time domain before sampling (fig. 1).

– Raw sample the signal and incur aliasing (which gives a poor-quality sound, so we usually go for the first choice).

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

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

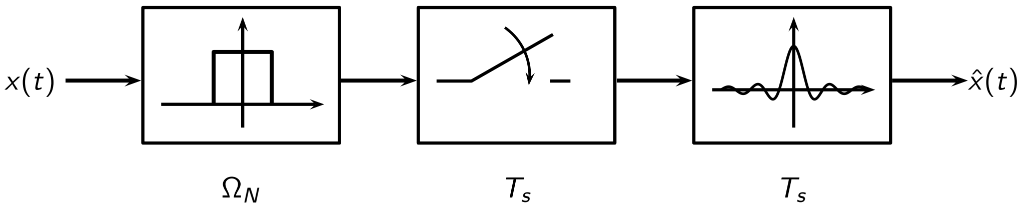


FIGURE 1

1:40 4:03

Sinc sampling and interpolation

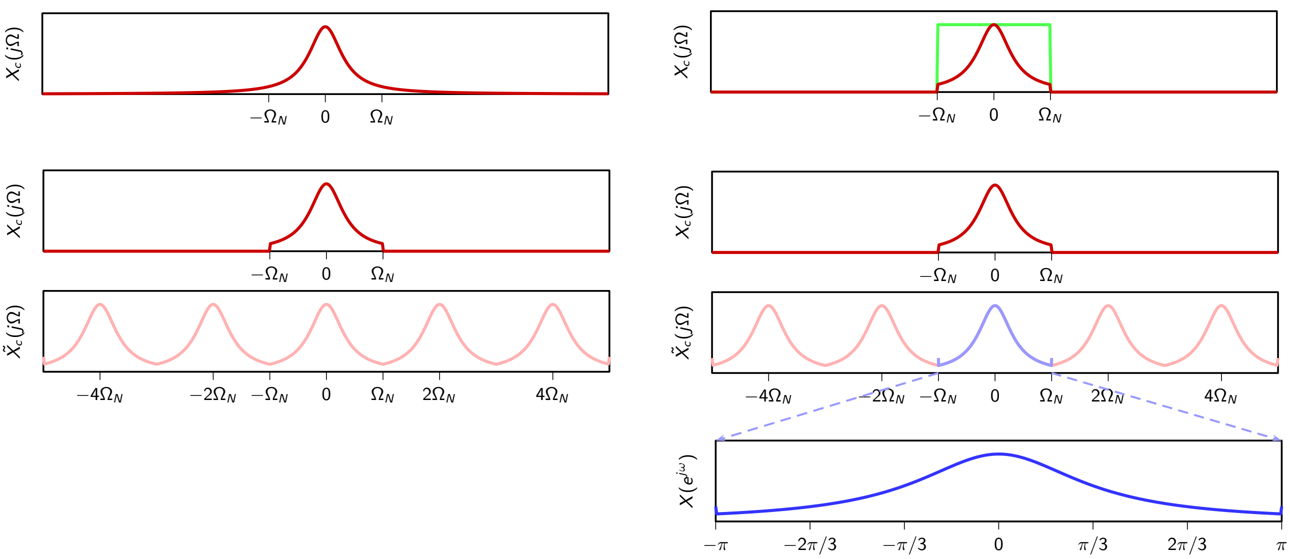


FIGURE 2

2:50 4:03

Concrete application of sinc sampling and interpolation

5.5.a STOCHASTIC SIGNAL PROCESSING

Stochastic signals can be described in terms of a probabilistic model. Is it possible to process random signals? Yes, it is, and we will give an indication of how to deal with such signals as noise.

The frequency-domain representation for stochastic processes is the power spectral density:

$$P[k] = E[|X_n[k]|^2/N]$$

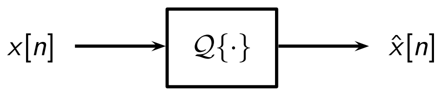
A stochastic process is characterized by its power spectral density (PSD). It can be shown that the PSD is $_x(e^{j\omega})$ = DTFT{$r_x[n]$} where $r_x[n] = E[x[k]x[n+k]]$ is the autocorrelation of the process x. For a filtered stochastic process, $y[n] = H\{x[n]\}$: $P_x(e^{j\omega}) = |H(e^{j\omega})|^2 P_x(e^{j\omega})$.

We can model noise as a stochastic signal, where the most important type of noise is white noise. "White" indicates the independencies of the samples, hence: $r_W[n] = \sigma^2 \partial[n]$ and $P_W(e^{j\omega}) = \sigma^2$ where σ^2 is the variance of the noise.

Very often we use Gaussian distribution to model the underlying probability distribution function for the sample. The reason is that Gaussian distribution is the model of choice when we want to represent the effect of many unknown superimposed sources, as is the case for noise. In this case we call the noise additive white Gaussian noise, or AWGN for short.

5.5.b QUANTIZATION

Digital devices can only deal with integer values, so we need to map the numeric range of a signal onto a finite set of values, which leads to an irreversible loss of information. Samples are stored and processed as integers. This operation is called quantization. It maps the numeric range of the discrete samples onto a finite set of integers.



Several factors at play:

- storage budget (bits per sample)
- storage scheme (fixed point, floating point)
- properties of the input
  - range
  - probability distribution

FIGURE 1

0:37 9:03

Quantization scheme $x \in \mathbb{C}$ and $\hat{x} \in \mathbb{R}$

Because of the difference between the actual value and the represented integer, the quantizer introduces noise into the original signal. We focus on scalar quantizers, that is, ones where samples are quantized individually and independently.

If the signal is bounded on an interval [A,B], the optimal quantizer is a uniform quantizer where the interval is divided into 2^R intervals of equal size. Moreover, if the input signal is uniformly distributed on this interval, the optimal representation value for each subinterval in terms of minimizing the mean-squared error is the subinterval midpoint. In this case, the error energy is $\Delta^2/12$, $\Delta = (B–A)/2^R$, the signal-to-noise ratio (ratio between the power of the signal and power of the noise) of a uniform quantizer is equal 2^{2R}, which in decibels corresponds to $6R$. Of course, we are analyzing a very simple type of quantization over a very simple class of signals, and quantizers can be designed to be much more complicated than that.

LEHMAN BROTHERS

Signal of the Day

We are going to look at an application of signal processing to financial data, more specifically, the time series of stock prices of Lehman Brothers around 2008. Lehman Brothers was a global financial services firm, and one of the largest investment banks in the US. On September 15, 2008, after more than 150 years of existence, the firm filed for bankruptcy.

The bankruptcy was just the final act in a series of events that ultimately led up to it. Lehman Brothers was highly exposed to the US housing market, in particular to a type of loan known as a subprime mortgage. These loans were typically granted to clients who were not able to afford standard loans due to their limited financial capabilities. With the burst of the real estate bubble in the US, these clients were the first to default on their mortgages. This led to huge financial losses for the banks that were exposed to this market. If we take a look at Lehman Brothers' stock price at the time around this event (2004–2014), we can observe a drastic drop at the time of bankruptcy. Evolution of the stock does not exhibit clear patterns. But the underlying statistical characteristic might be more stable over time. With random signals, certain tools that we have developed are still valid, such as filtering. But we also need to introduce new tools, which is the subject of another class on statistical signal processing.

6.1.a THE SUCCESS FACTORS FOR DIGITAL COMMUNICATION

What are the reasons behind the success of digital communication?

The first one is the DSP paradigm: the fact that DSP works with integers makes signals easy to regenerate, digital filters allow us to implement a very precise phase control, and we can also seamlessly integrate adaptive algorithms into a DSP system.

The second success factor for digital communication comes from the algorithmic nature of DSP techniques: for example, JPEG in image coding where signal processing techniques such as discrete cosine transformation can be matched to information theory techniques (which involve compression of bit streams).

Finally, the third success factor is related to hardware advancements. Communication devices have become very power efficient, so we can have large data centers or central offices that process an enormous number of communication channels in parallel.

6.1.b CONSTRAINTS OF THE ANALOG CHANNEL

The capacity of a channel is the maximum amount of information that can reliably be delivered along it (usually expressed in bits/seconds). Every analog channel has two inescapable limits: bandwidth and power constraints, both affecting the final capacity of the channel.

Let us focus on the relationship between bandwidth and capacity. Suppose we are going to transmit information encoded over a continuous-time channel: we take the samples, interpolate them with a sampling period T_s, if T_s is very small, then we can send more samples per second but the bandwidth will grow to $1/T_s$. So we see that capacity and the amount of information that we can send per second are related.

Similarly, the relationship between the power constraint and capacity can be appreciated because we can never do away with noise: as the receiver, we have to guess what has been sent after it has been corrupted by noise. It is rather intuitive that a signal with a wider range will have more power.

6.1.c THE DESIGN PROBLEM

The all-digital paradigm is basically keeping everything digital until we hit the physical channel (using a Digital-to-Analog converter).

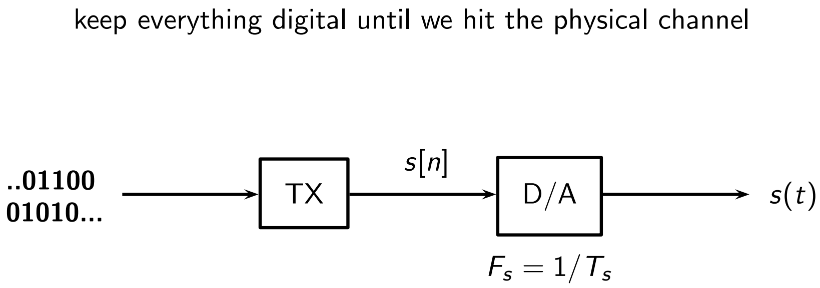


FIGURE 1

0:10 3:53

All-digital paradigm in a block diagram

The channel constraints look like a filter design problem: we have a bandwidth that is specified in term of a maximum and minimum frequency, so we can only operate over this band, and then we have a power constraint that restricts the power associated to the signal that we produce.

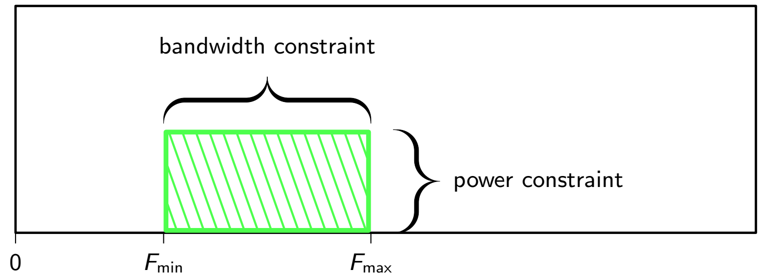


FIGURE 2

0:49 3:53

Channel constraints

Here are some working hypotheses that are common to most transmission systems you will come across. We start from a bitstream, and we will convert this bitstream into a sequence of symbols or samples $a[n]$ via a mapper. The mapper associates a groups of bits to a specific symbol. Now, we have to figure out what to do before converting those symbols into an analog signal: we have to fulfill both the bandwidth and power constraints. If we assume that the data is randomized (by a scrambler) and therefore the symbol sequence is a white sequence, we know that the power spectral density is simply equal to the variance, which implies that the power will be constant over the entire frequency band. To satisfy the bandwidth frequency we will have to introduce a new concept: upsampling.

6.2.a UPSAMPLING

We need to be able to "shrink" the support to a full-band signal so that it fits on the band allowed by the channel. We will thus use multirate techniques. In multirate, the main goal is to increase or decrease the number of samples in discrete-time signal. One way of doing this is to interpolate the digital signal we are given and resample it at a different sampling rate. However, we want to avoid the transition to discrete time and we want to perform this artificial change of sampling rate entirely in the digital domain. Let us then consider the upsampling operation.

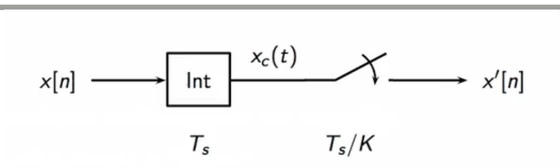


FIGURE 1

0:48 7:46

Upsampling via continuous time (interpolating at T_s and resampling at T_s/K)

Let $T_s = 1$,

$$x_c(t) = \sum_{m=-\infty}^{\infty} x[m]sinc(t-m)$$

$$x'[n] = x_c(n/K)$$

$$= \sum_{m=-\infty}^{\infty} x[m]sinc(\frac{n}{K}-m)$$

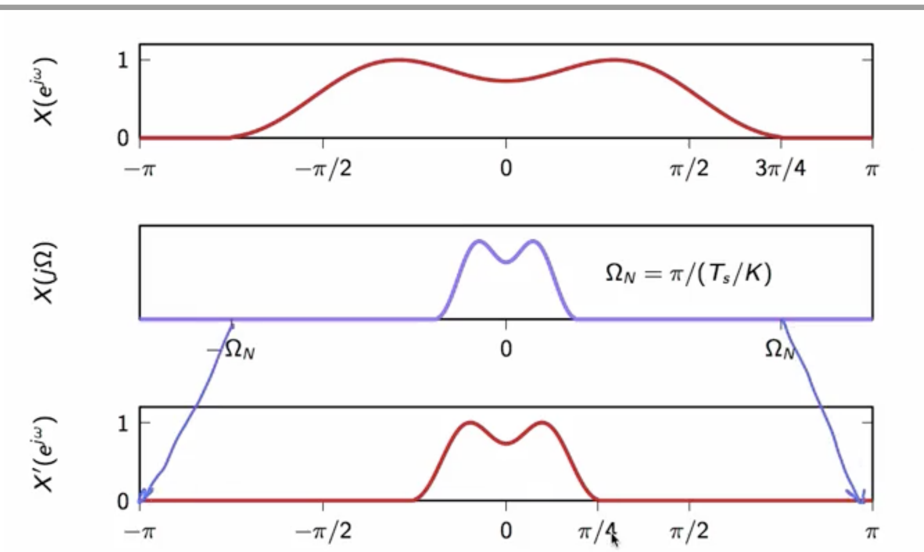


FIGURE 2

2:45 7:46

Example of upsampling application in the frequency domain

How can we achieve this in the digital domain only? We need to increase the number of samples by K, obviously the new signal will be built such that $x_u[m] = x[n]$ for m being a multiple of K, and insert $K-1$ zeros after each sample. Then, moving to the frequency domain, we will apply an ideal lowpass $\omega_c = \pi/K$, the result in sequence is thus:

$$\begin{aligned} x'[n] &= x_u(n) \star sinc(n/K) \\ &= \sum_{i=-\infty}^{\infty} x_u[i] sinc(\frac{n-i}{K}) \\ &= \sum_{m=-\infty}^{\infty} x[m] sinc(\frac{n}{K} - m) \end{aligned}$$

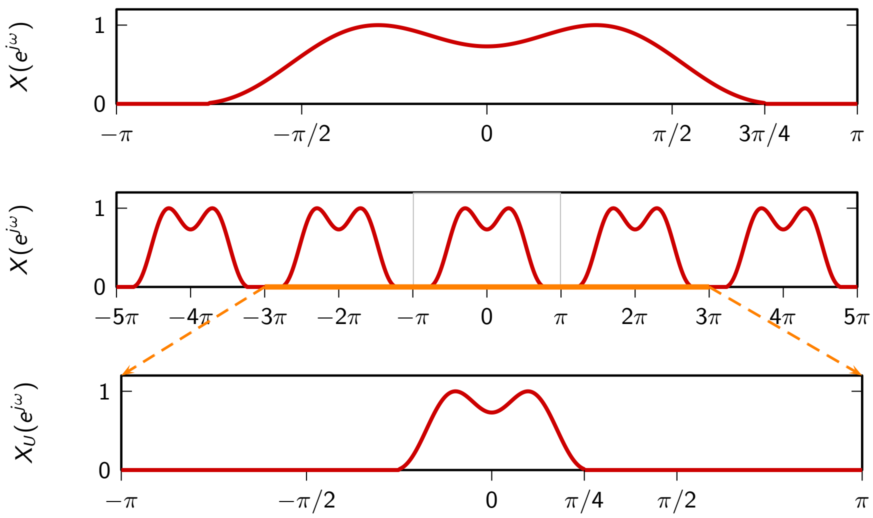


FIGURE 3

5:00 7:46

Example of upsampling with lowpass filter in the frequency domain

In general, if we have an upsampled sequence, we can always recover the original sequence by downsampling. Downsampling is a more complex operation than upsampling, just as sampling is more complicated than interpolation. We encourage you to read about multirate signal processing in the book.

6.2.b FITTING THE TRANSMITTER SPECTRUM

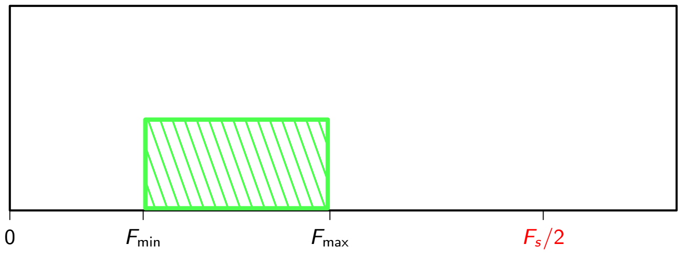


FIGURE 1

0:00 4:51

Bandwidth constraints

Let $W = F_{max} - F_{min}$ then pick $F_s < 2F_{max}$ to avoid aliasing, $F_s = KW$, where K is a positive integer. In the digital domain, we can simply upsample the symbol sequence by K so that its bandwidth in the digital domain will be $\omega_{max} - \omega_{min} = 2\pi W/F_s = 2\pi/K$.

Upsampling does not change the data rate, we produce W symbols per second. W is the fundamental data rate of the system, it is sometimes called the Baud rate of the system and it is equal to the available bandwidth.

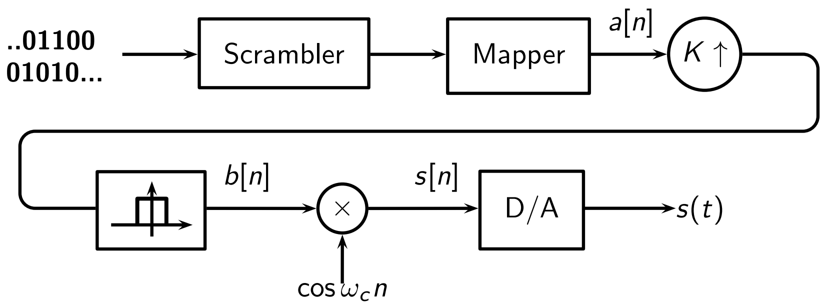


FIGURE 2

2:07 4:51

Revised block diagram for the transmitter

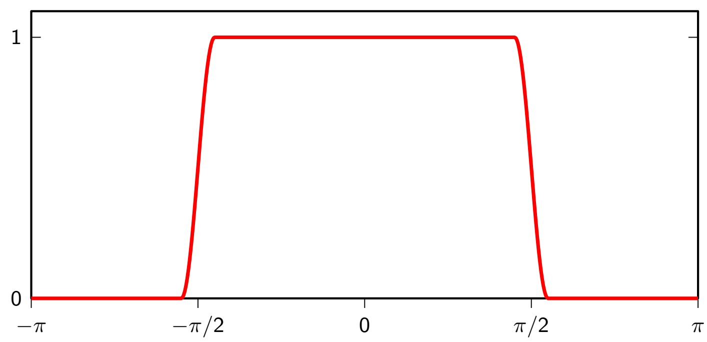


FIGURE 3

4:00 4:51

A very good filter: the raised cosine (frequency response)

6.3.a NOISE AND PROBABILITY OF ERROR

The transmitter sends a sequence a[n] and the receiver obtains a sequence â[*n*], but even if there is no distortion, we cannot avoid â[*n*] = *a*[*n*] + η[*n*], where η is the noise that corrupts the original sequence. When the noise is very large, our estimate of the original sequence will be off and we will make a decoding error. The probability of this error will depend on the power of the noise with respect to the power of the signal and on the decoding strategy. One way of maximizing our chances at correctly guessing the transmitted symbols is to use a suitable alphabet of transmission symbols.

Let's assume we have a randomized bitstream coming in, and we want to send upsampled and interpolated samples over the channel before transmitting them. How do we go from the bitstream to the samples, i.e., how does the mapper work? It splits incoming bitstreams into chunks and assigns a symbol *a*[*n*] from a finite alphabet A to each of them. To undo the mapping operation, and recover the bitstream, the receiver performs a slicing operation which consists in deciding which symbol is "closest" to â[*n*] and then piecing back together the corresponding bitstream.

Let's assume that the noise and the signal are independent, and that the noise is AWGN with zero mean and σ_0 variance. If we consider the example of a binary sequence sent mapped to {+*G*,–*G*}, the two-level signaling example, the probability of error will be:

$$
\begin{aligned}
P_{err} &= P[\nu[n] < -G | n-th \text{ bit is } 1] + P[\nu[n] > G | n-th \text{ bit is } 0] \\
&= (P[\nu[n] < -G] + P[\nu[n] > G])/2 \\
&= P[\nu[n] > G] \\
&= \int_G^{\infty} \frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-\frac{x^2}{2\sigma_0^2}} d\tau \\
&= erfc(G/\sigma_0)
\end{aligned}
$$

and the transmitted power will be defined as:

$$
\sigma_s^2 = G^2 P[n-th \text{ bit is } 1] + G^2 P[n-th \text{ bit is } 0] = G^2
$$

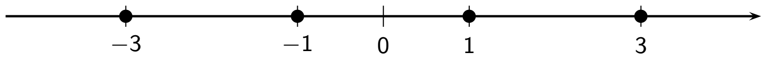
$$
Pe_{err} = erfc(\sigma_s/\sigma_0) = erfc(\sqrt{\text{SNR}})
$$

From that example, we can see that in order to reduce the probability of error, we should increase the amplitude of the signal *G*, which increases the power of the signal. But we also know that we cannot go above the channel's power constraint.

To increase the throughput, we can use multilevel signaling.

6.3.b PAM AND QAM

Pulse Amplitude Modulation (PAM): the mapper splits the incoming bitstream into chunks of M bits so that each chunk corresponds to an integer $k[n] \in \{1, 2, \dots, 2^M - 1\}$. This sequence is then mapped onto a sequence $a[n] = G((-2^M + 1) + 2k[n])$, G being the gain factor.



- distance between points is $2G$
- using odd integers creates a zero-mean sequence

FIGURE 1

1:05 11:33

PAM for $M = 2$, $G = 1$

The error analysis for PAM is very similar to what we carried out for bilevel signaling (which is basically a PAM where $M = 1$) and the result is again an exponentially decaying function of the SNR. If we want to increase the throughput even further, we can use complex numbers and build a complex valued transmission system. The name of this complex valued map and scheme is Quadratic Amplitude Modulation (QAM). The mapper takes the incumbent bitstream and splits it into chunks of M bits with M even. Then it uses half of the bits to define a PAM sequence, $a_r[n]$, and the remaining $M/2$ bits to define another independent PAM sequence, $a_i[n]$. The final symbol sequence is a sequence of complex numbers $a[n]= G(a_r[n] + ja_i[n])$. At the receiver, the slicer works by finding the symbol in the alphabet that is closest in Euclidian distance to the received symbol.

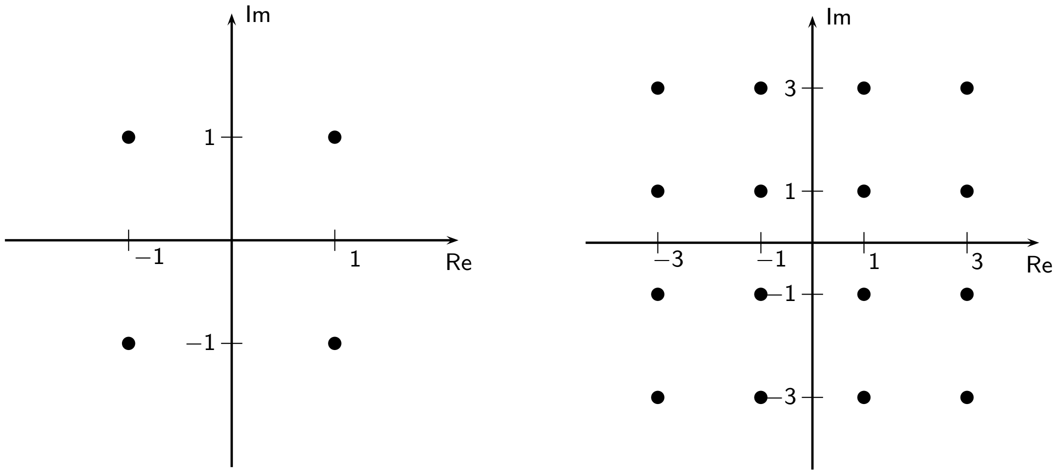


FIGURE 2

3:17 11:33

QAM for $M = 2, 4$, $G = 1$

The way the slicer works is by defining decision regions around each point, which is a square of side $2G$ centered around the point. As long as the received point is received within the decision region, we will not make an error; it will be decoded correctly. To quantify the probability of error, again we assume $\hat{a}[n] = a[n] + \eta[n]$, where the noise is a complex-valued Gaussian variable with variances in the real and imaginary axes of $\sigma_0/2$:

$$\begin{aligned} P_{err} &= P[|Re(\nu[n])| > G] + P[|Im(\nu[n])| > G] \\ &= 1 - P[|Re(\nu[n])| < G\hat{|}Im(\nu[n])| > G] \\ &= 1 - \int_D f_\nu(z)dz \\ &\simeq e^{-\frac{G^2}{\sigma_0^2}} \end{aligned}$$

Transmitter power (all symbols equiprobable and independent):

$$\begin{aligned} \sigma_s^2 &= G^2 \frac{1}{2^M} \sum_{a \in A} |a|^2 \\ &= G^2 \frac{2}{3}(2^M - 1) \end{aligned}$$

$$P_{err} \simeq e^{-\frac{G^2}{\sigma_0^2}} \simeq e^{3 - 2^{-(M-1)SNR}}$$

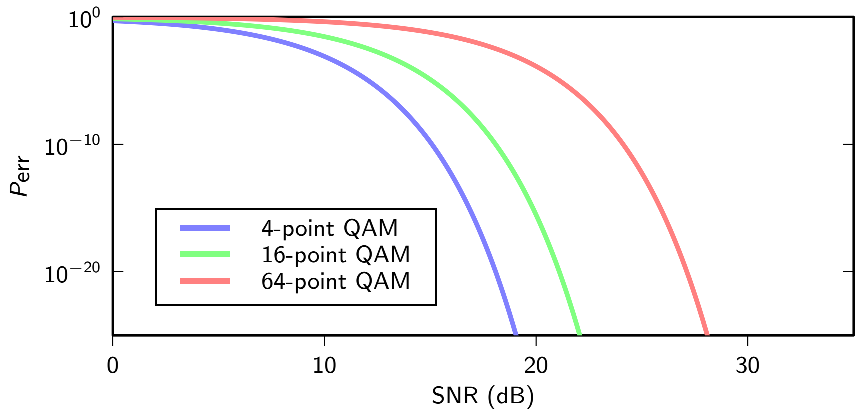


FIGURE 3

9:00 11:33

Probability error of a QAM constellation

6.4.a MODULATION AND DEMODULATION

Let $b[n] = b_r[n] + jb_i[n]$ be a complex-valued baseband signal. We can create the following real-valued passband signal:

$$
\begin{aligned}
s[n] &= Re\{b[n]e^{j\omega_c n}\} \\
&= Re\{(b_r[n] + jb_i[n])(cos\omega_c n + jsin\omega_c n)\} \\
&= b_r[n]cos\omega_c n - b_i[n]sin\omega_c n
\end{aligned}
$$

If we receive this as the receiver, we will be able to recover the original signal.

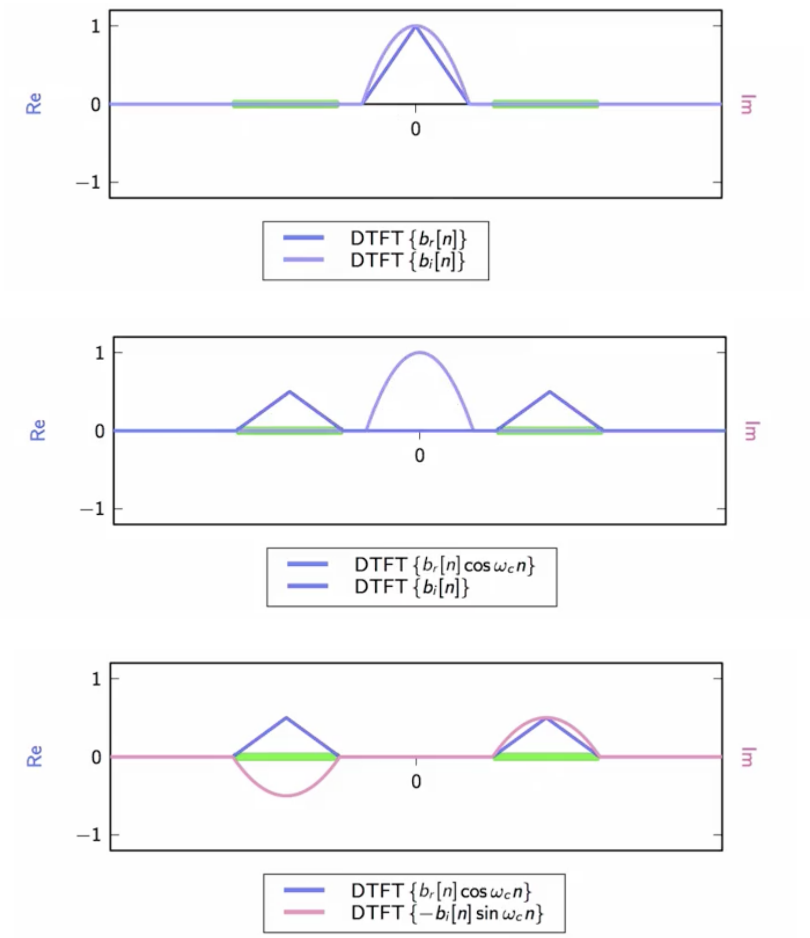


FIGURE 1

3:30 7:41

Illustration of the modulation process

To recover the baseband signal, we have to multiply by the carrier at the receiver as follows:

$$\begin{aligned} s[n]cos\omega_c n &= b_r[n]cos^2\omega_c n - b_i[n]sin\omega_c ncos\omega_c n \\ &= b_r[n]\frac{1+cos2\omega_c n}{2} - b_i[n]\frac{sin2\omega_c n}{2} \\ &= \frac{1}{2}b_r[n] + \frac{1}{2}(b_r[n]cos2\omega_c n - b_i[n]sin2\omega_c n) \end{aligned}$$

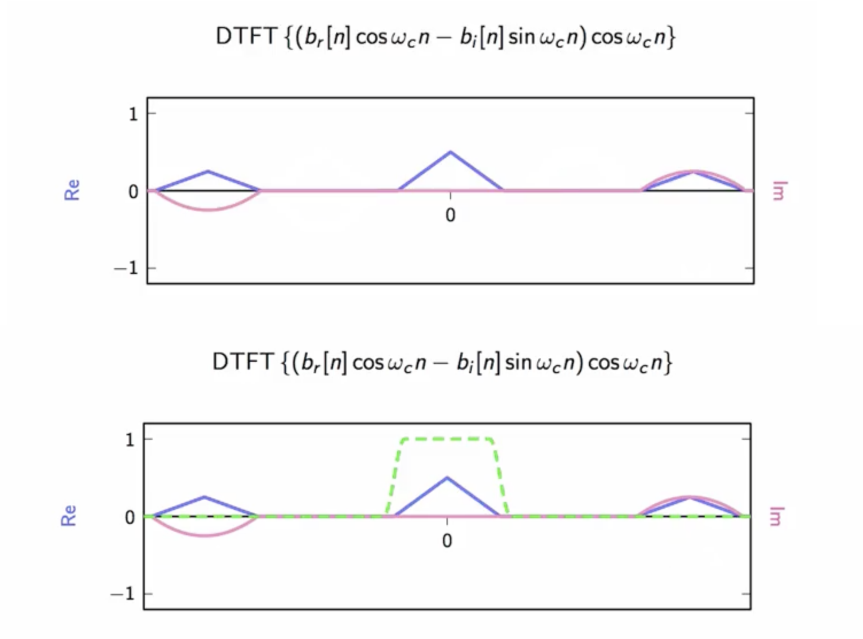


FIGURE 2

4:19 7:41

Recovering the in-phase component (filtering out the unnecessary components with a lowpass filter)

Similarly for the quadrature part of the baseband signal, we can multiply $s[n]$:

$$\begin{aligned} s[n]sin\omega_c n &= b_r[n]cos\omega_c nsin\omega_c n - b_i[n]sin^2\omega_c n \\ &= -\frac{1}{2}b_i[n] + \frac{1}{2}(b_r[n]sin2\omega_c n - b_i[n]cos2\omega_c n) \end{aligned}$$

then filter out the unnecessary values with a low pass filter.

6.4.b DESIGN EXAMPLE

Let us now see how we can put everything we have learned so far and design a practical system to send data over the telephone channel. Suppose that the bandwidth constraint for the telephone channel stipulates that we can only transmit data from F_{min} = 450Hz to F_{max} = 2850Hz, with the center frequency F_c = 1650Hz. This gives us a usable bandwidth of W = 2400Hz. As we have to pick a sample frequency that is at least twice the highest frequency, let us choose F_s = 3·2400 = 72000Hz. When we translate this back to the original domain, the modulating frequency is ω_c = 0.458π. Let us also assume that the maximum SNR is 22 dB. Let us pick P_{err} = 10^{-6}. Plugging those values in the formula, using QAM:

$$M = log_2(1 - \frac{3}{2}\frac{10^{22/10}}{ln(10^{-6})}) \tilde{=} 4.1865$$

we pick M = 4 and use a 16-point constellation. The final data rate is WM = 9600 bits/second.

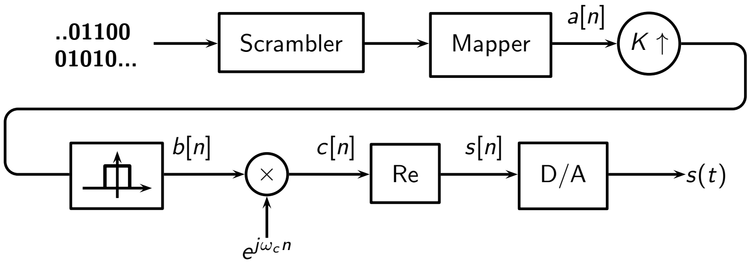


FIGURE 1

0:10 5:19

Final design for the transmitter

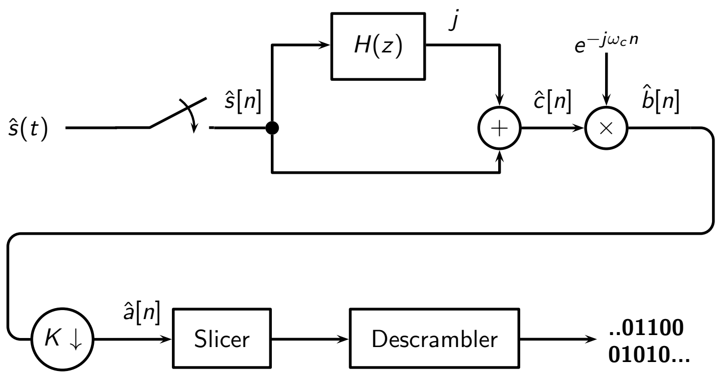


FIGURE 2

0:20 5:19

Final design for the receiver

Shannon's capacity formula is the upper bound on the amount of information that we can send C = W·$\log_2$(1+SNR) [bits/second], which in our case gives C ~17500bps. With our design we are basically hitting half of the capacity of the channel, but that gap can be narrowed by more sophisticated coding techniques.

6.5.a RECEIVER DESIGN

As this video treats an example around an audio file we encourage you to watch it.

You may have recognized the sound as the obligatory soundtrack every time you used to connect to the Internet. This is the sound made by a V34 modem. And you probably wondered what was going on. The receiver has to cope with four potential sources of problem: interference, the propagation delay (the delay introduced by the channel), the linear distortion (introduced by the channel), and drifts in the internal clocks between the digital system inside the transmitter and the digital system inside the receiver. When it comes to interference the handshake procedure and line probing pilot tones are used in clever ways to circumvent the major sources of interference. Propagation delay is tackled with a delay estimation procedure. Distortion for displaying the channel is compensated for by using adaptive equalization techniques. Clock drifts are tackled by using timing recovering techniques.

6.5.b DELAY COMPENSATION

To simplify the analysis we will assume that the clocks at the transmitter and receiver are synchronized and synchronous. Assume the channel acts as a simple delay: $\hat{s}(t) = s(t - d)$ which implies that the frequency response of the channel is $D(j\Omega)= e^{-j\Omega d}$. It introduces a delay of d seconds, which we can write as $d = (b + \tau) T_s$, where b is a positive integer and $|\tau| < 1/2$, b is called the bulk delay, and τ is called the fractional delay.

It is relatively simple to compensate for the bulk delay; imagine the transmitter begins transmission by sending just one impulse over the channel. The discrete time signal is just a delta in zero. It gets to the receiver after a delay, D, that we can estimate, for instance, by looking at the placement of the peak of the intervalation function. So for the receiver to offset the bulk delay we will just set the nominal $n = 0$, to coincide with the location of the maximum value of the sample sequence.

To compensate for the fractional delay, the transmitter sends an initial known sequence from which the receiver derives an estimate of the fractional delay. Given this estimation, the receiver implements a Lagrange interpolation (as in module 6.2) that uses a neighborhood of points around the current observations to compensate for the fractional delay. We then obtain $2N + 1$ Lagrangian coefficients. Then we filter with the resulting FIR. The latter can be implemented as a simple FIR filter, a great advantage in practice.

6.5.c ADAPTIVE EQUALIZATION

In figures 1 and 2, we would like to undo the effects of the channel on the transmitted signal.

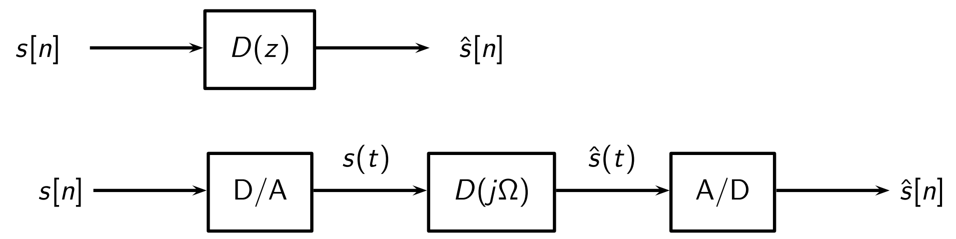


FIGURE 1

0:10 5:17

Scheme of effects on the signal sent through the channel

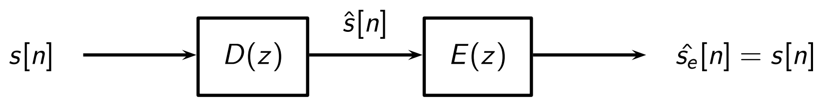


FIGURE 2

0:50 5:17

Undo the effect of the channel

The target is for the output to give us a signal that is equal to the transmitted signal. In theory, we could say that $E(z) = 1/D(z)$, but we might not know $D(z)$ in advance, and it also may change over time. We then need to use adaptive equalization.

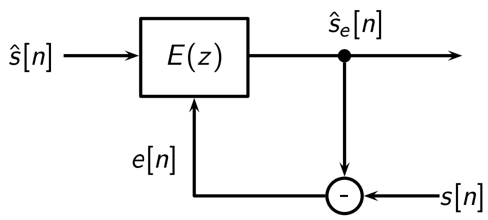


FIGURE 3

1:55 5:17

Adaptive equalization

How do we get the exact transmitting signal at the receiver's end? Using two tricks: the first one is bootstrapping (the transmitter will send a prearranged sequence of symbols to the receiver), and the second one is the online mode. Both are explained in figures 4 and 5 via block diagrams.

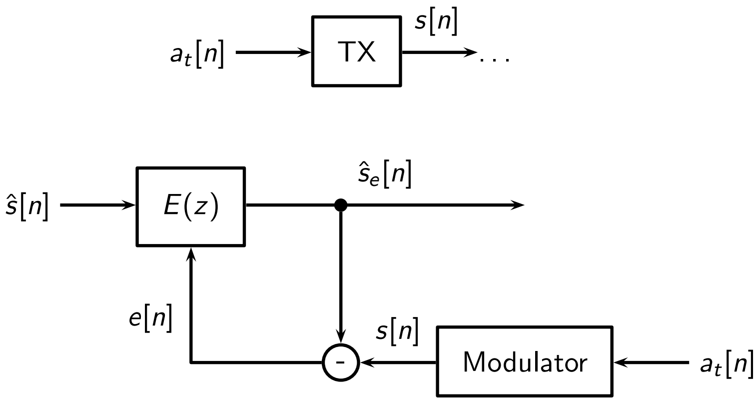


FIGURE 4

3:00 5:17

Bootstrapping via a training sequence

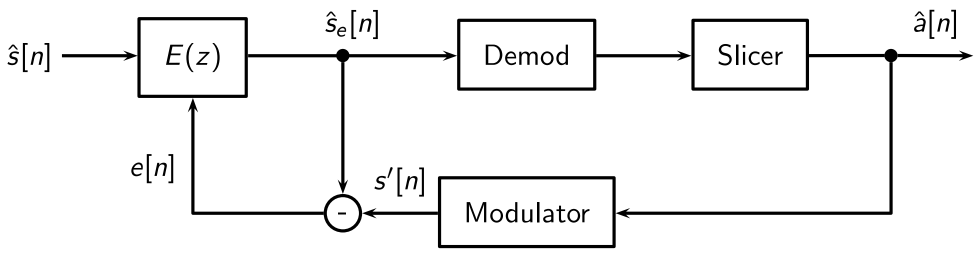


FIGURE 5

3:30 5:17

Online mode

There are still so many questions that we would have to answer to be thorough. For instance, how do we carry out the adaptation of the coefficients in the equalizer? How do we compensate for different clock rates in geographically diverse receivers and transmitters? How do we recover from the interference from other transmission devices and how do we improve noise resilience? The answers to all those questions require a much deeper understanding of adaptive signal processing. And hopefully, that will be the topic of your next signal processing class.

6.6.a ADSL DESIGN

When you talk on the phone, voice communication is sent to the voice network and then relayed to what is called the plain old telephone system, POTS.

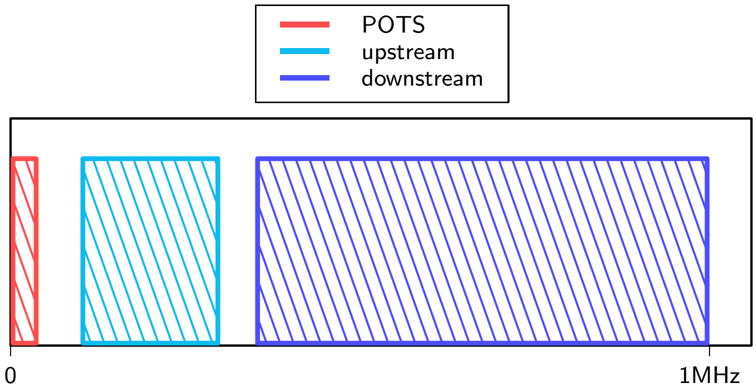


FIGURE 1

1:30

8:05

The ADSL channel organization for positive frequencies

Copper wire has both a very large bandwidth and can experience many different sources of disturbances, such as radio interferences. Therefore, it is hard to design a global solution for the entire channel. In ADSL the channel is divided into equally-spaced subchannels. The low frequency ones are reserved for voice communication. Then a portion is used for the upstream and a bigger portion for the downstream. It is this asymmetry in the differing sizes that explains the A in the ADSL acronym. ADSL stands for asymmetric digital subscriber line. Each subchannel is treated independently and transmits data according to its own SNR. Some might not even be used if the interferences are too high, for example. In ADSL, each subchannel implements an indecent QAM based on the local SNR. There is a simple implementation of this bank of independent signaling schemes as an inverse FFT. This ease of implementation is certainly an important reason behind the huge commercial success of ADSL.

6.6.b DISCRETE MULTITONE MODULATION

ADSL transmission can be efficiently implemented with simply an inverse FFT. We will look at a very efficient implementation of that signaling strategy that goes under the name of discrete multitone modulation.

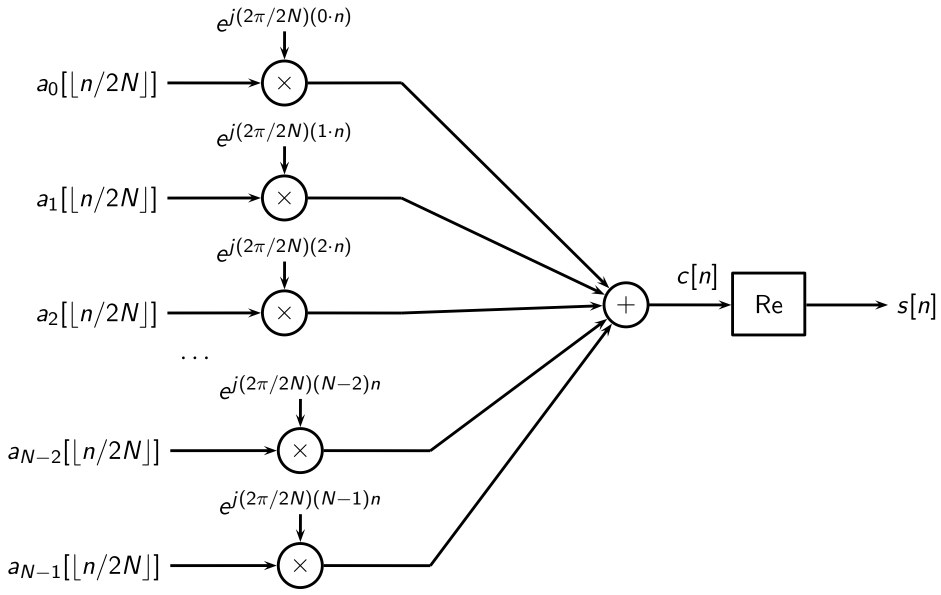


FIGURE 1

4:38 9:37

Bank of modems revisited

The complex output signal of such a scheme is:

$$
\begin{aligned}
c[n] &= \sum_{k=0}^{N-1} a_k[n/2N] e^{j\frac{2\pi}{2N}nk} \\
&= 2N\ IDFT_{2N}\{[a_0[m]\ a_1[m] \cdots a_{n-1}[m]\ 0\ 0 \cdots 0]\}[n] \\
m &= n/2N
\end{aligned}
$$

We are interested in:

$$s[n] = Re\{c[n]\} = (c[n] + c^*[n])/2$$

therefore

$$s[n] = N\ IDFT\{[2a_0[m]\ a_1[m] \cdots a_{N-1}[m]\ a^*_{N-1}[m]\ a^*_{N-2}[m]\ \cdots a^*_1[m]]\}[n]$$

Schematically, we can draw up the ADSL transmitter as one big inverse FFT. And the inputs to this FFT are twice the baseband symbol, followed by the symbols for the subchannels from 1 to $N-1$.

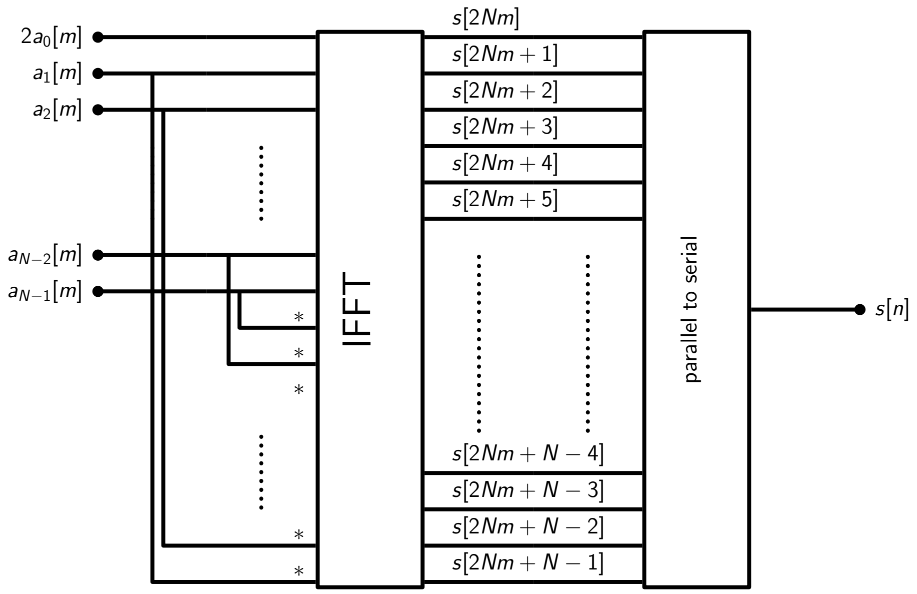


FIGURE 2

8:26 9:37

ADSL transmitter

Here are the ADSL specs:

- $F_{max} = 1104$ KHz
- $N = 256$
- each QAM can send from 0 to 15 bots per symbol
- forbidden channels: 0 to 7 (voice)
- channels 7 to 31: upstream data
- max theoretical throughput: 14.9 Mbps (downstream)

MOIRÉ PATTERNS

Signal of the Day

Moiré is a French word that indicates a form of textile with a lot of surface decoration. If you ever get called in for a TV interview, do not wear a striped shirt. A moiré pattern will appear and give your shirt a different aspect.

The simplest way to generate moiré patterns is to take a regular pattern, such as a set of vertical lines, and then make a copy of the same pattern, rotate it a little bit differently, and then superpose them. There are other types of moiré patterns of other dimensions and it can even be noticed in music. They are caused by aliasing.

Let's look at a practical application of moiré patterns, which comes in the form of secure printing. Take a 20 euro banknote. The banknote is embedded with very, very fine lines, too fine for most scanners or camera to capture correctly. So if you take this banknote and try to make more money by photocopying it or scanning it, the moiré patterns appear in the secure zone of the banknote, and prevents you from making an accurate copy of it.

IMPRESSUM

Layout:
Emphase Sàrl, Lausanne

Summary: Brune Bastide

Developed by the EPFL Press, BOOCs (Books and Open Online Courses) provide the accompanying text for MOOCs at the Swiss Federal Institute of Technology in Lausanne (EPFL).
They create a real learning advantage and bring added value to the MOOC, summarizing the main points to remember to help obtain the qualification.
Learn faster, learn better. Happy studying!

ISBN 978-2-88914-425-9