CHRISTOPH LAUER

CONTEMPORARY SIGNAL PROCESSING ALGORITHM

CHRISTOPH LAUER ENGINEERING

PUBLISHED BY CHRISTOPH LAUER ENGINEERING

http://www.christoph-lauer.de

First printing, April 2015

Contents

List of Figures

List of Tables

Dedicated to my family, my cat and friends.

Chapter 1- Time-Domain based Signal Processing

Time domain and analoge to digital signal processing

Signal processing in the time domain...

The Dirac Delta Function

The Dirac delta impulse function[1] has a special importance when it came to the description of signal processing systems. He is a model for a loud bang and bright flash of light, power bursts or a similarly strong but short signal. Due to its importance in theory and application of signal processing, the properties of the Dirac delta impulse are to be described. They will find use again and again in the later chapters. Dirac's delta function was introduced in 1927 from Dirac, but defined in 1950 from Schwartz in his theory of distributions. The Dirac Delta Function is somewhat abstractly defined as:

$$\delta(t) = 0 \quad \text{for } t \neq 0$$
$$\int_{-\infty}^{\infty} \delta(t)dt = 1$$

This can be thought of as a very "tall-and-thin" spike with unit area located at the origin, as shown in Figure.

The sifting property of the Dirac Impulse: We can use the fact that the delta function lose the argument at points unequal zero to "pull out" function values for a specific time.

$$\int_{-\infty}^{\infty} f(t)\delta(t)dt$$

To calculate this integral we use the definition of the delta function as scaled rectangle function.

$$\delta(t) = \lim_{\Delta \to 0} \frac{1}{\Delta t} rect(t/\Delta T)$$

[1] The Dirac impulse is also called, unit impulse or delta distribution.

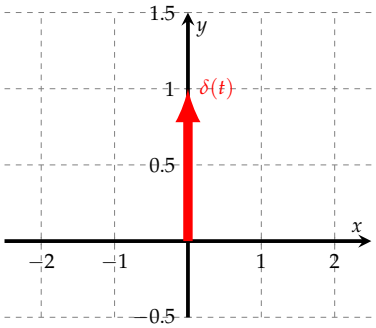


Figure 1: The Dirac delta function $\delta(x)$.

and thus

$$\begin{aligned}\int_{-\infty}^{\infty} f(t)\delta(t)dt &= \lim_{\Delta\to 0}\frac{1}{\Delta t}\int_{-\infty}^{\infty} rect(t/\Delta T)dt \\ &= \lim_{\Delta\to 0}\frac{1}{\Delta t}\int_{-\Delta/2}^{\Delta/2} f(t)dt \\ &= \lim_{\Delta\to 0}\frac{1}{\Delta t}F(t)\Big|_{\frac{-\Delta t}{2}}^{\frac{\Delta t}{2}} \\ &= \lim_{\Delta\to 0}\frac{F(\frac{\Delta t}{2})-F(-\frac{\Delta t}{2})}{\Delta t} \\ &= F'(0) \\ &= f(0)\end{aligned}$$

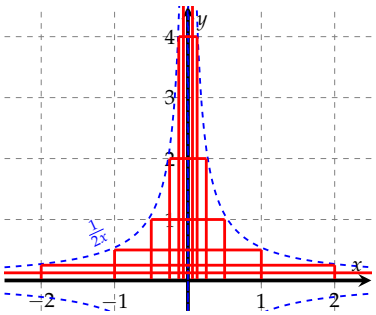


Figure 2: The delta function approximated from the scaled rectangle function. $\delta(x)$.

Then one has:

Theorem 1 (Sifting Property)

$$\int_{-\infty}^{\infty} f(t)\delta(t)dt = f(0)$$

This property can be generalized for arbitrary points t_0:

Theorem 2 (Generalised Sifting Property)

$$\int_{-\infty}^{\infty} f(t-t_0)\delta(t)dt = f(t_0)$$

Linear combinations of Dirac functions: A constant in front of the Delta function is called the weight. After the usual calculation rules for the integration we get

$$\int_{-\infty}^{\infty} f(t)a\delta(t)dt = a\int_{-\infty}^{\infty} f(t)\delta(t)dt = af(0)$$

In the same way we get for the linear combination of delta impulses

$$\begin{aligned}\int_{-\infty}^{\infty} f(t)(a\delta(t)+b\delta(t))dt &= a\int_{-\infty}^{\infty} f(t)a\delta(t)dt + b\int_{-\infty}^{\infty} f(t)b\delta(t)dt \\ &= (a+b)\int_{-\infty}^{\infty} f(t)\delta(t)dt\end{aligned}$$

or shorter:

Theorem 3 (Linear Combination of Dirac Functions)

$$a\delta(t) + a\delta(t) = (a+b)\delta(t)$$

Time scaling of the Dirac Impulse: Consider the integral

$$\int_{-\infty}^{\infty} \delta(at) f(t) dt$$

substitution with $\tau = at$ results for positive a

$$\int_{-\infty}^{\infty} \delta(\tau) f(\frac{\tau}{a}) \frac{d\tau}{a} = \frac{1}{a} f(0)$$

for negative a, since the integration limits must be substituted

$$\int_{\infty}^{-\infty} \delta(\tau) f(\frac{\tau}{a}) \frac{d\tau}{a} = -\frac{1}{a} \int_{-\infty}^{\infty} \delta(\tau) f(\frac{\tau}{a}) \frac{d\tau}{a} = -\frac{1}{a} f(0)$$

for positive and negative a

$$\int_{-\infty}^{\infty} \delta(\tau) f(\frac{\tau}{a}) \frac{d\tau}{a} = -\frac{1}{|a|} \int_{-\infty}^{\infty} \delta(\tau) f(\frac{\tau}{a}) \frac{d\tau}{a} = -\frac{1}{|a|} f(0)$$

Theorem 4 (Stretched Dirac Function)

$$\delta(at) = -\frac{1}{|a|} f(0)$$

The derivation of the Heaviside function: Although the Heaviside step function is not differentiable at the jump point, the Dirac function can useful defined as it derivation.

$$\int_{-\infty}^{t} \delta(\tau) d\tau = \int_{-\infty}^{t} \Theta(t-\tau)\delta(\tau) d\tau = \Theta(t)$$

Derivation of both sides results

Theorem 5 (Derivation of the Heaviside Function)

$$\delta(t) = \frac{d}{dt}\Theta(t)$$

[2]

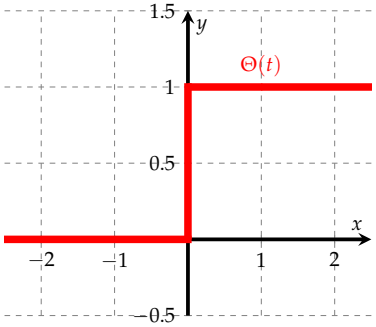


Figure 3: The derivation of the Heaviside function $\Theta(t)$ is the Dirac function $\delta(x)$.

[2] The Heaviside function is often called $u(t)$, unit step function or in the literature ϵ function.

The derivation of the Dirac function: For the determination of the derivative of the delta function we consider the integral

$$\int_{-\infty}^{\infty} \left(\frac{d}{dt}\delta(t)\right) f(t)dt = \delta(t)f(t)\Big|_{-\infty}^{\infty} - \int_{-\infty}^{\infty} \delta(t)\dot{f}(t)dt$$

we set $f(t) = t$ and derive on both sides

Theorem 6 (Derivation of the Dirac Function)

$$t\dot{\delta}(t) = -\delta(t)$$

Systems

Signals are not changeless in the real world. They often receive intended or accidentally changes when they are transferred over a physical medium. The transmission of an audio signal via an electrical line (for example with the telephone) can distort, damp or noise the signal, and the vocal of a chorus receives an echo in the church. Even the inner ear of the human transfers not all frequencies equally well and leads to changes of the signal. Mathematical a system is described through the transformation of an input signal $s(t)$ to an output signal $g(t)$:

$$(g(t) = Tr\{s(t)\}$$

Linear Time-Invariant Systems

A LTI system is an idealization of an real system with the advantage of simpler theoretical treatment. Many real systems approximate very good a LTI system. A system is linear if

$$Tr\left\{\sum_i a_i s_i(t)\right\} = \sum_i a_i \underbrace{Tr\{s_i(t)\}}_{g_i(t)} = \sum_i a_i g_i(t)$$

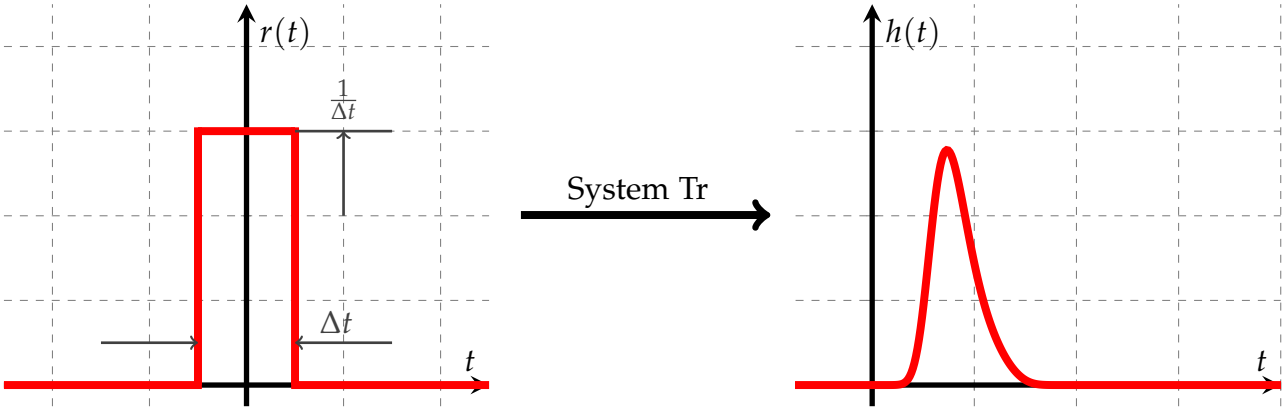
It responds to an input signal with the double amplitude with output signal with the double amplitude. An time invariant system is defined as:

$$Tr\{s_i(t - t_0)\} = g(t - t_0)$$

A system that satisfy both conditions is called a LTI system.

The Impulse Response

Beginning from the rectangle signal we can derive an general relation between an input and an output signal of a LTI system. Consider the



input signal

output signal

Figure 4: The answer $h(t)$ from a real LTI system of a rectangle input impulse $r(t)$ with the width Δt and the height $1/\Delta t$.

response of a LTI system to an rectangle input signal from the duration Δt and the height $1/\Delta t$:

$$r(t) = \frac{1}{\Delta t} rect(t/\Delta t)$$
$$Tr\{r(t)\} = h(t)$$

An arbitrary input signal can be decomposed to a linear combination of rectangular pulses:

$$s(t) \approx \sum_{i=-\infty}^{\infty} rect\left(\frac{t-\Delta T}{\Delta t}\right) s(i\Delta)$$

the concatenated and approximated signal (see Figure 5) can be send trough a LTI system and we get corresponding the linearity and time-invariance of the system:

$$\begin{aligned} g(t) &= Tr\left\{ \sum_{i=-\infty}^{\infty} rect\left(\frac{t-\Delta T}{\Delta t}\right) s(i\Delta) \right\} \\ &= \sum_{i=-\infty}^{\infty} s(i\Delta t) Tr\left\{ \underbrace{\frac{1}{\Delta t} rect\left(\frac{t-\Delta T}{\Delta t}\right)}_{r(t-i\Delta t)} \right\} \Delta t \\ &= \sum_{i=-\infty}^{\infty} s(i\Delta t) h(t-i\Delta t) \Delta t \end{aligned}$$

To approximate the signal more precisely we must decrease Δt. For the limit $\Delta t \to 0$ the sum results over in the integral

$$g(t) = \int_{-\infty}^{\infty} h(t-\tau) s(\tau) d\tau$$

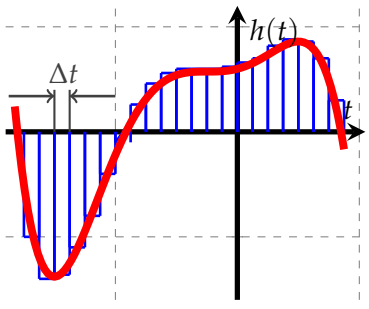


Figure 5: Linear decomposition of an arbitrary input signal into rectangle functions with the distance Δt.

Example: for $s(t) = \delta(t)$ we get:

$$g(t) = \int_{-\infty}^{\infty} h(t-\tau)\delta(\tau)d\tau = h(t)$$

We make in interesting observation:

Theorem 7 (Impulse Response of a LTI system)

$$g(t) = \int_{-\infty}^{\infty} h(t-\tau)\delta(\tau)d\tau$$

[3]

[3] The answer of a system to a Delta function is called impulse response $h(t)$ and completely describes a system.

Casusal Systems: The output of a causal system depends only of the present and past values of the input to the system (Event before Effect principle). Classically, nature or physical reality has been considered to be a causal system. [4]

[4] This requires that the impulse response of a causal system is o for t<o

Stability: A stable system came automatically back to his initial state and reacts to every bounded input signal with a bounded output signal. This is called BiBo stability and can be expressed with $\sum_{-\infty}^{\infty} |h(t)| < \infty$.

The Convolution

Convolution is the generalization of the above derivated integrals.

Theorem 8 (Convolution)
The integral

$$g(t) = \int_{-\infty}^{\infty} a(t-\tau)b(\tau)d\tau$$

is called Convolution and is abbreviated with the convolution product

$$a(t) * b(t)$$

short a(t) convoluted with b(t).

The convolution is a valid transformation equation for all LTI systems, because the convolution of an an arbitrary input signal with the impulse response always results the corresponding output signal. Corresponding theorem 8 we can also write for such a system

$$g(t) = h(t) * s(t)$$

where $s(t)$ and $g(t)$ describe the input and output signal, and h(t) the impulse response of the system. In principle the calculation rules for the convolution are the same as for the multiplication.

One Element:	$a(t) * \delta(t) = a(t)$
Commutativity	$a(t) * b(t) = b(t) * a(t)$
Associative Property	$[a(t) * b(t)] * c(t) = a(t) * [b(t) * c(t)]$
Distributive property	$a(t)[*b(t)] + c(t)] = a(t) * b(t) + a(t) * c(t)$

As example, the commutativity is proved by

$$a(t) * b(t) = \int_{-\infty}^{\infty} a(t-\tau)b(\tau)d\tau$$

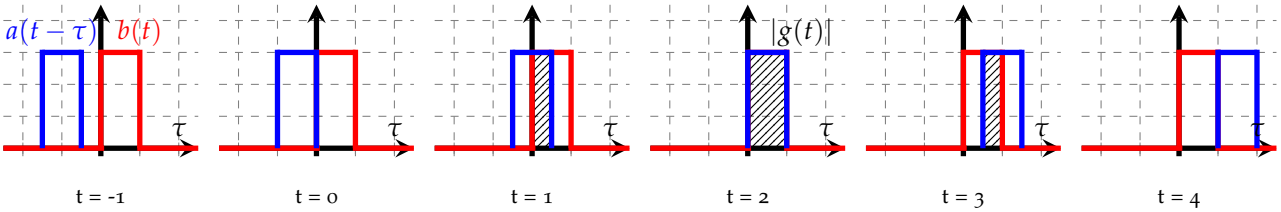
we substitute $\vartheta = t - \tau$

$$\int_{\infty}^{-\infty} a(\vartheta)b(t-\vartheta)[-d\vartheta] = \int_{-\infty}^{\infty} b(t-\vartheta)a(\vartheta)d\vartheta$$
$$= b(t) * a(t)$$

■

The Graphical Convolution

The graphical interpretation of the convolution can be easily deduced from the definition of the convolution integral via the graphical integration. Thus the convolution integration can be illustrated as shifting the flipped function a(t) over the function b(t). The convolution at the time t is the value of the intersection area of both.



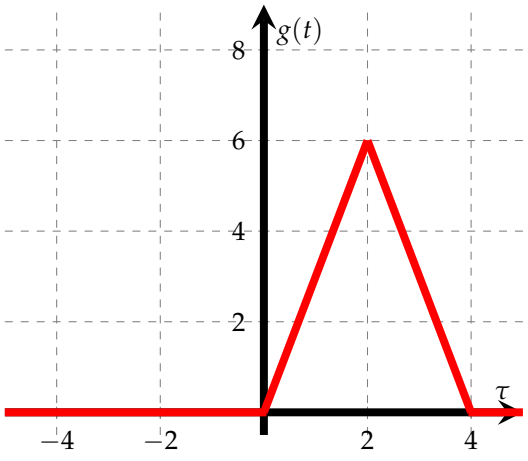


Figure 6: An example for a graphical convolution with two rectangle signals. The flipping of the blue rectangle can not be seen due the symmetry.

The Shannon-Nyquist Sampling Theorem

Introduction

Signals and waveforms are essential to the understanding of physical measurement systems. A signal can be discrete or continuously on the time axis or the range of values. A signal which can hold any value is called a value-continuous signal, otherwise its called a value-discrete signal. A signal is time-continuous if the knowledge of its value is required at any one time, otherwise it is called a time-discrete signal. A value-discrete signal is also called two-valued or binary. The conversion of a value-continuous signal into a value-discrete signal is called **quantization (=value discretization)**. The conversion of a time-continuous signal into a time-discrete signal is called **sampling (=time discretization)**.

Sampling in the Time Domain

Sampling means the recording of measured values at discrete time points. Thus, a time-continuous signal is converted into a time-discrete signal. In this context the sampling rate has a high importance, since it indicates the number of samples per second. The main principle of the sampling is the recording of values over long time period. These values are converted into digital measurement values in the following. Sampling is thus a repeated measurement of the signal for a short time (see figure 7).

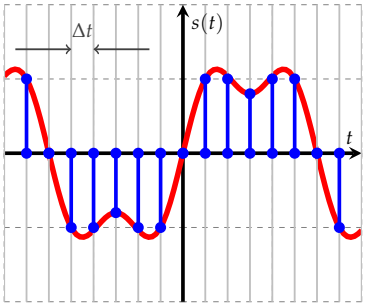


Figure 7: Time sampling with a sampling rate from $s_r = 1/\Delta t$.

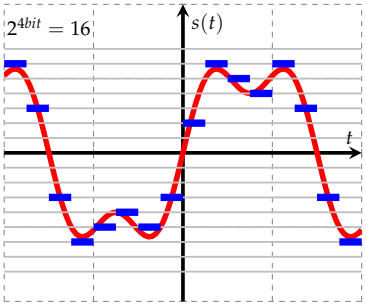


Figure 8: Value quantization with a typical resolution of $2^{n=4}$.

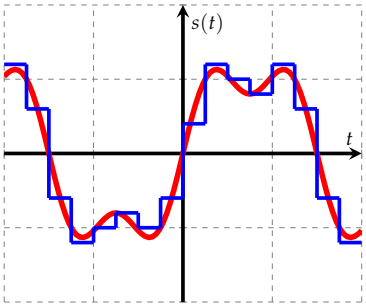


Figure 9: The sampled Signal.

The Ideal Sampler

The ideal sampler is defined as a detector where the signal is not detected over a long time period, but picked up exactly at the sampling time. Mathematically the ideal sampler can be described with a summarization of delta pulses.

$$s_a(t) = \sum_{-\infty}^{\infty} \delta(t - n\Delta T) \cdot s(t)$$

The Real Sampler

Of course, the ideal sampler is not feasible because we can not produce ideal Dirac pulses. The signal is rather captured to a time period around the actual sampling point. For the reconstruction of the signal we need further an ideal low pass. However such a low pass is practically not realizable. Of course there is a approach for both problems. With the *"Sample and Hold"* circuit the Dirac impulse can be corrected with a rectangle impulse. The difficulty with the low pass filter can be handled

with the increase of the sampling rate. The single spectra are thereby stretched further apart (see figure 10).

$$s_A(t) = \sum_{-\infty}^{\infty} rect\left(\frac{t - n\Delta T}{T_0}\right) \cdot s(t).$$

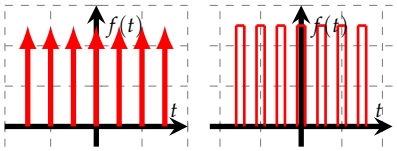


Figure 10: The real and Ideal Sampler.

Theorem 9 (Scha Function)

$$\text{Ш}(t) = \sum_{n=-\infty}^{\infty} \delta(t - n)d$$

[5]

[5] The name comes from the similarity with the Cyrillic letter Ш ("Scha"), the function is also called Dirac comb.

We build the Fourier transformation from the Ш function and obtain the following transform pair.

$$\text{Ш} \circ\!\!-\!\!\bullet\, \text{Ш}\left(\frac{\omega}{2\pi}\right)$$

We can present the ideal sampler with this formulation

$$s_A(t) = \frac{1}{\Delta T} \cdot \text{Ш}\left(\frac{\omega}{2\pi}\right) \cdot s(t)$$

We apply the scaling theorem and get

$$\frac{1}{\Delta T}\text{Ш}\left(\frac{t}{\Delta T}\right) \circ\!\!-\!\!\bullet\, \text{Ш}\left(\Delta T \cdot \frac{\omega}{2\pi}\right)$$

$$\Rightarrow S_a(\omega) = \text{Ш}\left(\Delta T \cdot \frac{\omega}{2\pi}\right) * S(\omega) = \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} \delta\left(\Delta T \cdot \frac{\omega'}{2\pi} - n\right) \cdot (\omega - \omega')d\omega'$$

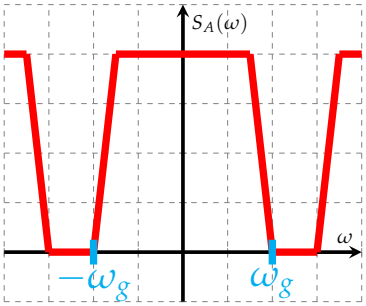


Figure 11: Spectrum of the real sampler.

We substitute $F = \Delta T \cdot \omega'/2\pi$ and rearrange

$$\begin{aligned} S_A(\omega) &= \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} \delta(F - n) \cdot S\left(\omega - \frac{2\pi F}{\Delta T}\right) dF \frac{2\pi}{\Delta T} \\ &= \sum_{n=-\infty}^{\infty} S\left(\omega - \frac{2\pi n}{\Delta T}\right) \cdot \frac{2\pi}{\Delta T} \end{aligned}$$

It is noticeable that $S_A(\omega)$ is periodic. The period length of the function is $2\pi/\Delta T$ Figue 11 illustrates this. The condition for the reconstruction of the signal is

$$\omega_g < \frac{2\pi}{\Delta T} - \omega_g \Rightarrow \Delta T < \frac{\pi}{\omega_g}.$$

Nyquist Theorem

The Nyquist frequency is defined as the cutoff frequency. It is the highest occurring frequency in the initial signal. The so called Nyquist rate refers to the frequency $2f$, which must be exceeded by the sampling rate.

Theorem 10 (Nyquist Rate)

$$\Delta T \leq \frac{\pi}{\omega} = \frac{1}{2f}$$

Undersampling

In signal processing we speak of subsampling or undersampling if the signal is discrete sampled with a sampling rate that is less than the Nyquist rate. The subsampling entail however several problems and makes the reconstruction of the signal more difficult. On the one hand, information is lost by the incomplete reconstruction of the signal components, on the other hand interference frequencies (mirror frequencies, aliasing) arise.

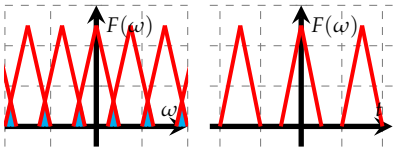


Figure 12: Under and Oversampling.

$$\Delta T > \frac{\pi}{\omega_g}$$

Oversampling

In signal processing, we speak of oversampling, if the signal is analyzed with a sampling rate higher than the Nquist rate. This sampling rate is greater than the signal bandwidth. This sampling shows, in contrast to the sub sampling some real world applications. It is used in SC filters, DA and AD converters.

$$\Delta T < \frac{\pi}{\omega_g}$$

Signal Reconstruction in the Frequency Domain

Sampling in the Frequency Domain

We need the transfer function for the reconstruction of the sampled signal

$$H(f) = \frac{1}{2\pi} rect\left(\frac{\omega}{2\omega_g}\right)$$

and get thus the signal

$$s(\omega) = \frac{\Delta T}{2\pi} S_a(\omega) \cdot rect\left(\frac{\omega}{2\omega_g}\right)$$

This applies only when the major condition $\Delta T < \pi/\omega_g$ is true.

$$s(t) = \frac{\Delta T}{2\pi} \cdot \left[\frac{1}{\Delta T} \text{Ш}\left(\frac{t}{\Delta T} \cdot s(t) \right) \right] * si(\omega_g t)$$

It should ne noted that $\Delta T = \frac{1}{2f} = \frac{\pi}{\omega_g}$.

$$\begin{aligned} s(t) &= \frac{1}{2\pi} \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} \delta(t' - n\Delta T) \cdot s(t') \cdot si(\omega_g(t - t'))dt' \\ &= \frac{1}{2\pi} \sum_{n=-\infty}^{\infty} s(n \cdot \Delta T) \cdot si(\omega_g(t - n \cdot \Delta T)) \\ &= \frac{1}{2\pi} \sum_{n=-\infty}^{\infty} s(n \cdot \Delta T) \cdot si\left(\frac{\pi}{\Delta T_g} (t - n \cdot \Delta T) \right) \end{aligned}$$

A band limited signal can be represented by a linear combination of si functions.

Sampling in the Frequency Domain

Like a continuous signal can be discretized in the time domain, the frequency function can also expressed via discrete frequency values. The application of the Fourier transformation results a similar therm like in the chapter before.

$$S_P(f) = S(f) \frac{1}{F} \text{Ш}\left(\frac{f}{F} \right)$$

time domain	sampled	periodic
frequency domain	periodic	sampled (linespectrum)

Table 1: Sampling properties in the time and frequency domain.

$$\mathcal{F}\{g(t)\} = \frac{1}{2\pi} \cdot T\pi \cdot rect\left(\frac{T\omega}{2} \right) * \left(T_0 \cdot si\left(\frac{T_0\omega}{2} \right) \cdot \frac{2}{T} \sum_{k=-\infty}^{\infty} \delta\left(\omega - k \cdot \frac{2}{T} \right) \right)$$

Quantization Error and Noise

When an Analog-Digital Converter (ADC) converts a continuous signal into a discrete digital representation, the transfer function is sort of a staircase. For each output code, there is a range of input values that produces the same output. That range is called quantum (Q) and is equivalent to the Least Significant Bit (LSB)[6]. Q can be calculated by

[6] The LSB is the lowest significant bit in the byte and the smallest possible level which can be discretized. The most significant bit is called MSB.
E.g. $\underbrace{1}_{MSB}0101010$, $0101010\underbrace{1}_{LSB}$.

dividing the range of the ADC by the number of steps in the staircase.

$$Q = \frac{V_{ref}}{2^n}$$

In the above equation, N is the number of bits of the ADC and the input range can be somewhere between 0 and V_{ref}. The difference between input and output is named the quantization error. Therefore, the quantization error can be between $-1/2Q$ and $+1/2Q$. This error can be considered a quantization noise with RMS:

$$v_{qn} = \frac{Q}{\sqrt{12}}$$

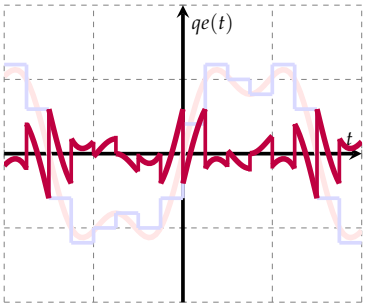


Figure 13: The Quantization Error.

The Spectrum of the Quantization Noise: We know the quantization noise power is v_{qn}^2, now where will it be concentrated or stretched in the frequency domain? The quantization error causes harmonics in the signal that go well above the Nyquist frequency. Through the sampling aspect of an ADC, those harmonics are folded to the Nyquist band, forcing the whole noise power into the Nyquist band and with an approximately white spectrum (similarly dispersed across all frequencies in the band). Particular converters work explicitly in oversampling (sampling much above the Nyquist frequency) to widen the noise over a wider band and then filter it digitally. Therefore, noise power can be lowered.

The Signal to Noise Ratio (SNR): Taking an input sinusoidal with peak-to-peak amplitude V_{ref}, at which V_{ref} is the reference voltage of an N-bit ADC (therefore, occupying the full-scale of the ADC), its RMS value is

$$V_{rms} = \frac{2^n Q}{2\sqrt{2}}$$

To determine the Signal-Noise Ratio, we divide the RMS of the input signal V_{rms} by the RMS of the quantization noise v_{qn}:

$$SNR = 20 \log \left(\frac{V_{rms}}{v_{qn}} \right) \quad [dB]$$

[7] Modifying equations v_{qn} and V_{RMS} into SNR will lead to

$$SNR = n \cdot 6.02 + 1.76 \quad [dB]$$

generalizes to any other system with a digital representation. Hence, a microprocessor which represents values with N bits would have a SNR defined by the above formula.

[7] Since the decibel is defined with respect to power, not amplitude, conversions of voltage ratios to decibels must square the amplitude, or use the factor of 20 instead of 10.

Increase the SNR by Oversampling: Oversampling the signal can be used to virtually increasing the SNR and the value resolution by **averaging multiple sample points in between**. It is important that the level of the signal noise is about the quantum (LSB), to archive true random values. In case of need the noise must be artificially generated and added to the signal. For example, if we oversample by a factor of $k = 4$, the SNR in terms of power improves by factor of 4 which corresponds to a factor of 2 improvement in terms of voltage. Figure 14 illustrates the SNR increasing.

$$SNR = n \cdot 6.02 + 1.76 + log_4(k) \cdot 6.02 \quad [dB]$$

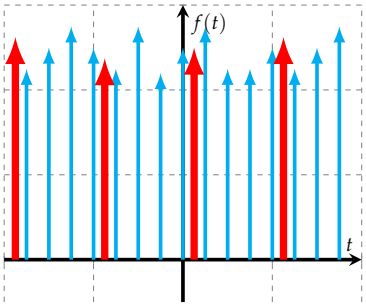


Figure 14: Increase the SNR and the resolution by oversampling and averaging. In this example is $k = 4$.

The Analog Digital Converter

Flash	Ultra-High Speed when power consumption not primary concern.
SAR	Medium to high resolution (8 to 16bit), 5Msps and under, low power, small size.
Delta Sigma	High resolution, low to medium speed, no precision external components, simultaneous 50/60Hz rejection, digital filter reduces anti-aliasing requirements.
Pipeline	High speeds, few Msps to 1+ Gsps, 8 bits to 16 bits, lower power consumption than flash.
Integrating	Monitoring DC signals, high resolution, low speed. low power consumption, good noise performance.

Table 2: ADC application comparison.

Analog-to-digital conversion is an electronic process in which a continuously variable (analog) signal is changed, without altering its essential content, into a multi-level (digital) signal. The input to an analog-to-digital converter (ADC) consists of a voltage that varies among a theoretically infinite number of values. Examples are sine waves, the waveforms representing human speech, and the signals from a conventional television camera. The output of the ADC, in contrast, has defined levels or states. The number of states is almost always a power of two – that is, 2, 4, 8, 16, etc. The simplest digital signals have only two states, and are called binary. All whole numbers can be represented in binary form as strings of ones and zeros. Digital signals propagate more efficiently than analog signals, largely because digital impulses, which are well-defined and orderly, are easier for electronic circuits to distinguish from noise, which is chaotic. This is the chief advantage of digital modes in communications. Computers "talk" and "think" in terms of binary digital data; while a microprocessor can analyze analog data, it must be converted into digital form for the computer

to make sense of it. A typical telephone modem makes use of an ADC to convert the incoming audio from a twisted-pair line into signals the computer can understand. In a digital signal processing system, an ADC is required if the signal input is analog. The image below shows a comparison the most common standard ADC types regarding the sample rate and the bit value resolution. Following we will discuss

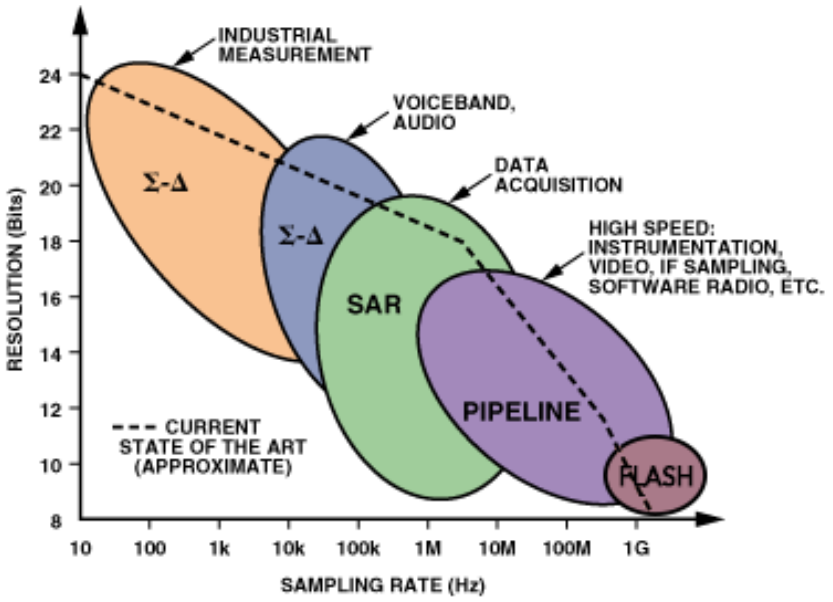


Figure 15: ADC performance comparison.

5 standard ADC types (Flash, Successive Approximation(SAR), Delta Sigma, Pipeline and Integrating) in detail.

The Flash ADC: Direct conversion is also known as a flash conversion. ADCs based on this architecture are extremely fast with a sampling rate of up to 5GHz. However, their resolution is limited because of the large number of comparators and reference voltages required. The input signal is fed simultaneously to all comparators. A priority encoder generates a digital output that corresponds with the highest activated comparator. The identity of comparators is important, any mismatch can cause a static error. Flash ADCs have a short aperture interval - the time when the comparators' outputs are latched.

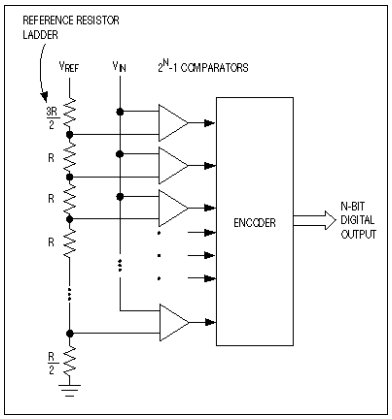


Figure 16: The Flash ADC.

The Successive Approximation ADC: The conversion technique based on a successive approximation register (SAR), also known as bit-weighing conversion, employs a comparator to weigh the applied input voltage against the output of an n-bit digital-to-analog converter (DAC). Using the DAC output as a reference, this process approaches the final result

as a sum of n weighting steps, in which each step is a single-bit conversion. Initially all bits of SAR are set to 0. Then, beginning with the most significant bit, each bit is set to 1 sequentially. If the DAC output does not exceed the input signal voltage, the bit is left as a 1. Otherwise it is set back to 0. It is kind of a binary search. For an n-bit ADC, n steps are required.

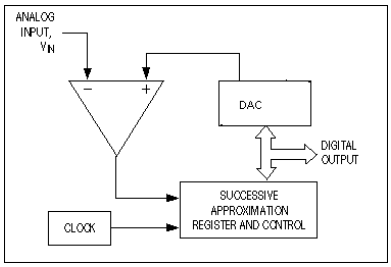


Figure 17: The Successive Approximation ADC.

The Delta Sigma ADC: Sigma Delta analog to digital converters (ΔΣ), also called oversampling converters or 1-bit converters, consist of 2 major blocks: modulator and digital filter. The modulator, whose architecture is similar to that of a integrating ADC, includes an integrator and a comparator with a feedback loop that contains a 1-bit DAC. The modulator is oversampling the input signal, transforming it to a serial bit stream with a frequency well above the required sampling rate. The output filter then converts the bit stream to a sequence of parallel digital words at the sampling rate. It is essentially a comparator with high order of oversampling which varies around the LSB. The level differences to the following are called Δ. 1 = higher Level, 0 = falling level. Higher resolutions can be archived trough higher sampling rates. For example 16-times oversampling for 16 bit. The delta-sigma converters perform high-speed, low resolution (1-bit) A/D conversions, and then remove the resulting high-level quantization noise by passing the signal through analog and digital filters. Features: high resolution, high accuracy, low noise, low cost. Good for applications with a bandwidth up to 1MHz, such as speech, audio.

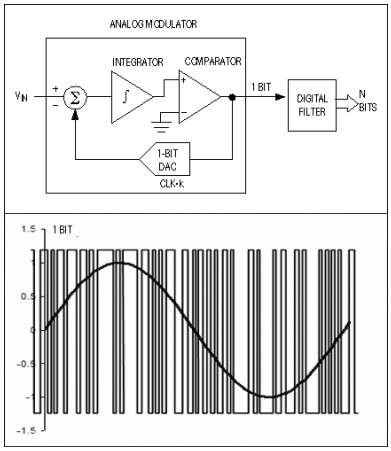


Figure 18: The Delta Sigma ADC.

The Pipeline ADC: A Pipeline ADC consists of a cascade of stages, each of which contains a low resolution ADC, DAC and amplifier, that successively convert the analog input into its digital representation, while processing the data in a pipe-lined manner. Pipeline ADCs are commonly used for power-efficient high-speed conversion of wide bandwidth input signals (e.g. 10 to 100 MHz). The ADC sampling frequency is usually the Nyquist frequency or lower using small OSRs (e.g. 2 or 4) and the ADC output code resolution is typically between 8 and 14-bit. The pipeline ADC is an open-loop architecture, having a small inherent latency (e.g. 4 to 6 clock cycles), and a direct relationship between the input signal and the output code. Pipeline ADCs are typically implemented in CMOS processes using Switched Capacitor (SC) discrete-time circuitry. A Pipeline ADC needs a non-trivial analog Anti-Aliasing filter, which consumes additional power and area.

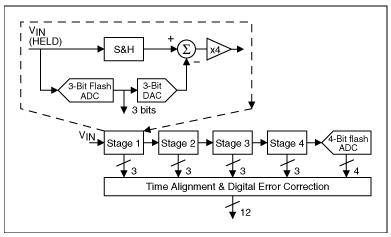


Figure 19: The Pipeline ADC.

The Integrating ADC: A classical dual-slope integrating converter (also called switched capacitor or SC ADC) is shown at the drawing. A current, proportional to the input voltage, charges a capacitor for a

fixed time interval T_{charge}. At the end of this interval the device resets its counter and applies an opposite-polarity (negative) reference voltage to the integrator input. With this opposite-polarity signal applied the cap is discharged by a constant current until the voltage at the output of the integrator reaches zero again. The time $T_{discharge}$ is proportional to the input voltage level and used to enable a counter. The final count provides the digital output, corresponding to the input level. Note that even the clock frequency does not have to have high stability, because both ramp-up and ramp down time are measured with the same clock.If the clock slows down 10%, the initial ramp will go 10% higher than normal, requiring 10% longer ramp down time resulting in the same final count. Only the discharge current produced by precise V_{ref} has to be of high stability. Integrating ADCs are extremely slow devices with low input bandwidths. But their ability to reject high-frequency noise and fixed low frequencies such as 50Hz or 60Hz makes them useful in noisy industrial environments and applications. Provide 10-18 bit resolution. A conversion time for a medium speed 12 bit integrating ADC is about 20mS. This type of ADC is most commonly used in multi-meters

The parameters of data converters may be expressed in several different ways: Q, parts per million of full scale (ppm), millivolts (mV), etc. Different devices (even from the same manufacturer) will be specified differently, so converter users must learn to translate between the different types of specification if they are to compare devices successfully.

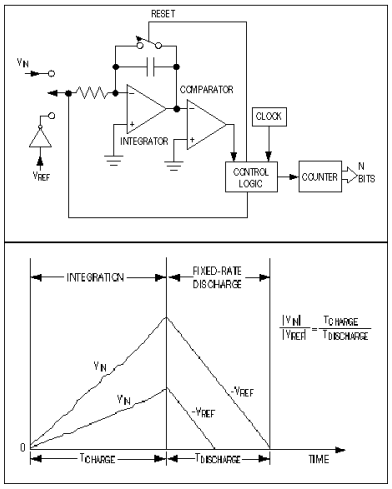


Figure 20: The Integrating ADC.

Resolution (n)	2^n	Q (V_{ref}=10V)	ppm	%	dB
2-bit	4	2.5V	250'000	25	-12
4-bit	16	625mV	62'500	6.25	-24
6-bit	64	156mV	15'625	1.56	-36
8-bit	256	39.1mV	3,906	0.39	-48
10-bit	1024	9,77mV	977	.098	-60
12-bit	4096	2.44mV	244	.024	-72
14-bit	16'384	610μV	61	.0061	-84
16-bit	65'536	153μV	15	.0015	-96
18-bit	262'144	38μV	4	.004	-108
20-bit	1'048'576	9.54μV	1	.0001	-120
22-bit	4'194'304	2.38μV	.24	.000024	-132
24-bit	16'777'216	596nV	.06	.000006	-144

Table 3: The Size of an LSB.

The Digital Analog Converter

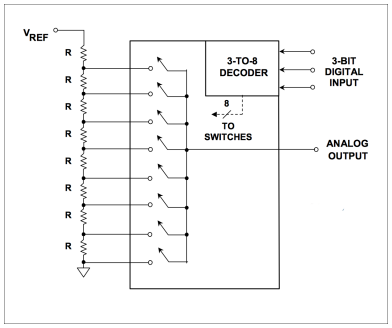


Figure 21: The String DAC.

The String DAC: The simplest DAC structure of all is the Kelvin divider, thermometer DAC or string DAC as shown in Figure 21. An n bit version of this DAC simply consists of $2n$ equal resistors in series and $2n$ switches (usually CMOS), one between each node of the chain and the output. The output is taken from the appropriate tap by closing just one of the switches. There is some slight digital complexity involved in decoding to 1 of $2n$ switches from N-bit data, but this digital circuitry is cheap.

The R / 2R DAC: One of the most common DAC building-block structures is the R-2R resistor ladder network shown in Figure 22. It uses resistors of only two different values, and their ratio is 2:1. An N-bit DAC requires 2N resistors, and they are quite easily trimmed. There are also relatively few resistors to trim.

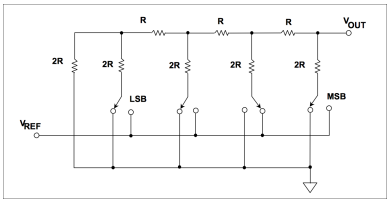


Figure 22: The R/2R DAC.

The Multiplying DAC: When we are required to design a DAC with a specific performance, it may well be that no single architecture is ideal. In such cases, two or more DACs may be combined in a single higher resolution DAC to give the required performance. These DACs may be of the same type or of different types and need not each have the same resolution. In all DACs, the output is the product of the reference voltage and the digital code, so in that sense, all DACs are multiplying DACs. But some DACs use an external reference voltage which may be varied over a wide range. These are "Multiplying DACs" or MDACs where the analog output is the product of the analog input and the digital code as shown in Figure **??**. They are extremely useful in many different applications. A strict definition of an MDAC is that it will continue to work correctly as its reference is reduced to zero, but sometimes the term is used less stringently for DACs which work with a reference range of 10:1 or even 6:1 a better name for devices of this type might be "semi-multiplying" DACs.

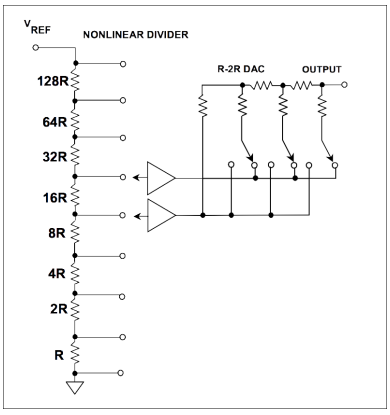


Figure 23: The Multiplying DAC.

The Delta Sigma DAC: The Delta Sigma digital to analog converters ($\Delta\Sigma$) are also called 1-bit converter. It is essentially a PWM signal whose arithmetic mean correspond to the reconstructed signal. Oversampled and interpolated so that the output value varies around the LSB with a comparator. Afterward this signal must be filtered with an aliasing filter. Delta-Sigma type DAC is another type of DAC that does not use the R2R ladder resistor network. Delta-Sigma type DAC is a DAC realizable only by software, which outputs PWM binary pulse waveform. To convert the PWM waveform to its corresponding analog waveform, a low-pass filter made of a RC network is all required. Following a Matlab/Octave listing for a Delta Sigma digital to analog converter which can be easily converted to any other language for example C or

assembler. A hardware low pass filter with the cutoff frequency of the half sample rate (for example a simple RC low pass) must be added to the output.

```
% initialize the variables and define the constants
close all;
clear all;
N = 64;   % the signal size
k = 8;    % the oversampling factor

% generate the digital sine signal
for i = 1:N
    y(i) = round( 32 * sin(4*pi*i/N) + 32);
end

% plot the sine signal
subplot(211);
plot(y,'linewidth',4,'color','r'); axis tight;
title('Sine wave')

% perform the DAC
Qtzer = 0;
Sigma = 0;
Out = [];
for j = 1:N
  DACin = y(j);
  for i = 1:k
    Delta = DACin - Qtzer;
    Sigma = Sigma + Delta;
    if(Sigma > 64)
      Qtzer = 64;
      Out = [Out, 1];
    else
      Qtzer = 0;
      Out = [Out, 0];
    end
  end
end

% plot the unfiltered oversampled analog signal
subplot(212);
stairs(Out,'-+b'); axis tight;
title('Unfiltered and oversampled DAC signal');
% save the output as image file
print -djpeg 'SinePWM.jpg'
```

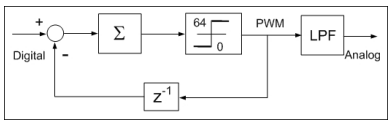


Figure 24: The ΔΣ DAC.

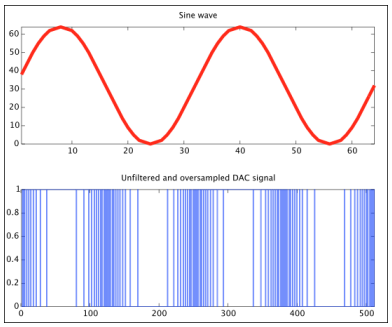


Figure 25: The output of the Matlab/Octave software ΔΣ DAC example program.

Correlation of Signals

One of the most interesting questions in the signal processing is: "How similar are two signals ?" This question arises whenever one wants to find a known pattern in a signal, is it to synchronize a wireless receiver with the carrier signal of a transmission, to recognize an object in image processing, or to find a word in voice recognition. The concept that every time is used for this purpose is called correlation.

Energy, Power, Correlation

The quadrature function is borrowed from the statistics where small differences are not noticed, however large differences emphasized disproportionately. In this sense the signal energy is defined as:

Theorem 11 (Signal Energy)

$$E_s = \int_{-t_1}^{t_2} |s(t)|^2 dt$$

8

[8] The absolute value is used in order that the formula remains valid for complex numbers. The limits are mostly $t_1 = -\infty$ and $t_2 = \infty$

This definition is for non periodic signals (for example a damped oscillation) to count the entire signal.
The observation of the entire signal is not meaningfully, especially not for the limits from $-\infty$ to ∞, because this limits can not be realized. Here it makes more sense to integrate over one period. In order to get always the same value, even for the integration over more periods, we normalize to the time. We define for periodic functions:

Theorem 12 (Signal Power)

$$P_s = \frac{1}{t_2 - t_1} \int_{-t_1}^{t_2} |s(t)|^2 dt$$

The therms "Energy" and "Power" came from the electrical engineering where the integrals are similar to those treated here. [9] For out further considerations we take the signal energy as base. The definitions and theorems apply also for signal powers and are derived analogously.

[9] The energy of a voltage signal in a circuit with a known resistance if $\int U(t)I(t)dt = 1/R \int U^2(t)dt$, it depend on the square of the voltage

, the signal energy is now to be taken as a basis. The definitions and theorems apply also to signal powers and are derived analogously.

Similarity Measure for signals

The mean square deviation of two signals $s(t)$ and $g(t)$ is

$$\begin{aligned} E_{sg}^A &= \int_{-\infty}^{\infty} |s(t) - g(t)|^2 dt < \\ &= \int_{-\infty}^{\infty} s^2(t)dt - \int_{-\infty}^{\infty} s(t)g(t) + \int_{-\infty}^{\infty} g^2(t)dt \\ &= E_s + E_g - 2\int_{-\infty}^{\infty} s(t)g(t)dt. \end{aligned}$$

The relationship to the shift theorem to calculate the variance in the statistics can be easily recognized.
The absolute energy is uninterestingly when two signals are compared, so they are not considered. Interestingly is only the last term

Theorem 13 (Cross-Correlation Coefficient)

$$\Phi_{sg} = \int_{-\infty}^{\infty} s^*(t)g(t)dt$$
$$= \int_{-\infty}^{\infty} s(t)g(t)dt \qquad \textit{(for real signals)}$$

The Cross-Correlation coefficient is exactly zero if there is no linear relationship between two signals. Such signals are called **orthogonal signals**. In the case of trivial linear functions the definition of orthogonality coincides with the definition from the geometry. This definition of the correlation is only useful if we can assume that the two signals have no temporal delay. If two signals occur shifted one to another, like in most technical systems, it makes sense to extend the function expression

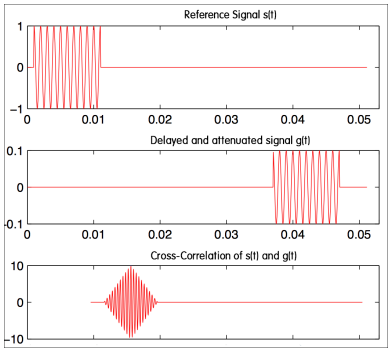


Figure 26: Cross-Correlation example.

Theorem 14 (Cross-Correlation)

$$\Phi_{sg}(\tau) = \int_{-\infty}^{\infty} s(t)g(t+\tau)dt$$

The Cross-Correlation coefficient is obtained from the Cross-Correlation function quite trivial with $\mathbf{\Phi}_{sg} = \Phi_{sg}(0)$. The Cross-Correlation is often used to measure time differences between similar signals like it shown in figure 26. Amazingly, one of the more common applications of the correlation definition is the comparison of a signal with itself, called self correlation or Auto-Correlation.

Theorem 15 (Auto-Correlation)

$$\Phi_{ss}(\tau) = \int_{-\infty}^{\infty} s(t)s(t+\tau)dt$$

Theorem 11 (signal energy) particular applies:

$$\Phi_{ss}(0) = \Phi_{ss} = E_s$$

The Auto-Correlation is often used is often used to detect periodicity in time series data. Autocorrelation measures the degree of association between numbers in a time series separated by different time lags. The Auto-Correlation is used to find repetives is signals. In this case the Auto-Correlation function has non-zero values on locations other than $\Phi_{ss}(0)$. On the other hand, only background noise has such a randomly distribution that periodicities do not occur and the noise never repeats.

Noise can be recognized as sharp peak the zero point but everywhere else disappears.

Correlation and Convolution

When we look at the definition of the correlation (theorem 13) we can see that there is a similarity between the correlation and the convolution. Actually the correlation can be expressed through the convolution.

$$\begin{aligned}\Phi_{sg}(\tau) &= \int_{-\infty}^{\infty} s(t)g(t+\tau) \\ &= \int_{-\infty}^{\infty} s(-\vartheta)g(\tau-\vartheta)dt' \qquad \text{(Substitution } \vartheta = -t\text{)} \\ &= s(-\tau) * g(\tau)\end{aligned}$$

From this relation you can also see what happens if the functions to compare are swapped

$$\Phi_{gs}(\tau) = g(-\tau) * s(\tau) \qquad = s(\tau) * g(-\tau) = \Phi_{sg}(-\tau)$$

Especially the Auto-Correlation function is now $\Phi(\tau) = \Phi(-\tau)$, the Auto-Correlation function os even.

Correlation relations in the Frequency Domain

We can now transform the above relations into the frequency domain and simplify the calculation from the convolution and correlation.

$$\begin{aligned}\Phi_{sg}(\omega) &= \mathcal{F}\{\Phi_{sg}(\tau)\} \\ &= \mathcal{F}\{s(-\tau) * g(\tau)\} \\ &= \mathcal{F}\{s(-\tau)\} \cdot \mathcal{F}\{g(\tau)\} \\ &= S*(\omega) \cdot G(\omega)\end{aligned}$$

Theorem 16 (Cross-Correlation in the frequency domain)

$$\Phi_{gs}(\tau) = S*(\omega) \cdot G(\omega)$$

we can now use the FFT for the fast computation of the Correlation.

The Wiener-Khinchin-Theorem: If we apply the last theorem for the Auto-Correlation we get:

Theorem 17 (Wiener-Khinchin relation)

$$\Phi_{ss}(\tau) = |S(\omega)|^2$$

The Wiener-Lee-Relation: If we want to estimate the correlation function of an signal which is transmitted over an LTI system we can make usage of the Wiener-Lee relation: Fot $g(\tau) = h(\tau) * s(\tau)$ we have

$$\begin{aligned}\Phi_{gg}(\tau) &= g(-\tau) * g(\tau) \\ &= h(-\tau) * s(-\tau) * h(\tau) * s(\tau) \\ &= h(-\tau) * h(\tau) * s(-\tau) * s(\tau)\end{aligned}$$

Theorem 18 (Wiener-Lee relation)

$$\Phi_{gg}(\tau) = \Phi_{hh}(\tau) * \Phi_{ss}(\tau)$$

Linear Predictive Coding

The LPC Error foo foo foo

The Digital Filters

foo foo foo

The Envelope Extraction

foo foo foo

The Resampling Methods

foo foo foo

The Zero-Order-Hold Resampler

foo foo foo

The Linear Interpolation Resampler

foo foo foo

The Sinc Bandlimited Interpolator Resampler

foo foo foo

The Impulse Response Extraction

foo foo foo

The Damping Rate Extraction

foo foo foo

The Shimmer Analysis

Shimmer is defined as as the average difference (in dB or percent) between two successive amplitude periods.

The Jitter Analysis

Jitter is defined as the variability of the periodic time (in percent) from period to period. foo foo foo

The Beamforming

foo foo foo

The Acoustic Camera

foo foo foo

Classical Methods

foo foo foo

The Zero Crossing Rate foo foo foo

The Polygonal Chain foo foo foo

Chapter 2 Frequency-Domain based Signal Processing

Integral Transformations

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

The Transfer Function

This section is used to derive the concept of eigenvectors and eigenfunctions of LTI systems and characterize them. The terms eigenfunction and eigenvalue derived from linear algebra.

The Eigenvectors:

The eigenvector of a mapping is defined as a vector which is not the zero vector and not change the direction of the mapping. But it takes a stretched by a factor, referred to as an eigenvalue. [10] Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be a matrix and $\mathbf{x}_E \in \mathbb{R}^n$ be a vector. Applies

$$A \cdot \mathbf{x}_E = \alpha \cdot \mathbf{x}_E$$

so is $\mathbf{x}_E$ called the eigenvector of the matrix $\mathbf{A}$. The variable α is this representation the eigenvalue. It should be noted that there are at most n linearly independent eigenvectors.

[10] The therms eigenvalue, eigenvector and eigenfunction are also called own-value, own-vector and own-function.

The Eigenfunctions:

It was already shown that a LTI system with an impulse response $h(t)$ answers to an input signal $s(t)$ with $g(t) = h(t) * s(t)$. Now there are functions $s_E(t)$ which show the following behavior if transmitted over a LTI system:

$$h(t) * s_E(t) = H \cdot s_E(t)$$

Such functions are not changed in their form, but only multiplied with a complex amplitude form factor H. The basic type of such a function is called the eigenfunction and is

$$s_E(t) = e^{j\omega t} = \cos(\omega t) + j\sin(\omega t).$$

for any frequency we obtain by inserting the special eigenfunction

$$h(t) * s_E(t) = h(t) * e^{j\omega t} = \int_{-\infty}^{\infty} e^{j\omega(t-\tau)} \cdot h(\tau) d\tau.$$

Extraction of the exponential function results in

$$\int_{-\infty}^{\infty} e^{j\omega(t-\tau)} \cdot h(\tau) d\tau = e^{j\omega\tau} \underbrace{\int_{-\infty}^{\infty} e^{-j\omega\tau} \cdot h(\tau) d\tau}_{H(\omega)} = e^{j\omega t} \cdot H(\omega).$$

$H(\omega)$ is called the transfer function or the frequency response.

Theorem 19 (Transfer Function)

$$H(\omega) = \int_{-\infty}^{\infty} h(\tau) \cdot e^{-j\omega\tau} d\tau$$

The Classical Continuous Fourier Transformation

After we met in the first chapter the properties of LTI systems, we now turn to the Fourier transform. It was derived from the Fourier series, which was developed by the French mathematician and physicist Jean Baptiste Joseph Fourier. The Fourier series decomposes a real-valued periodic function into a linear combination of sine and cosine oscillations. It should be noted that it is limited to periodic systems and aperiodic signals can not be evaluated.[11] The Fourier transformation converts a signal from the time domain into a signal in the frequency domain. The Fourier transformation results in the frequency spectrum of a time domain signal, and the frequency spectrum is defined as the sum of the frequencies which the signal contains. The time and frequency domains are alternative ways of representing signals. The Fourier transform is the mathematical relationship between these two representations. For this we look at the definition of the classical continuous Fourier transformation:

[11] Generally an aperiodic signal is understood as a function whose period runs to infinity.

$$H(\omega) = \int_{-\infty}^{\infty} h(\tau) \cdot e^{-j\omega\tau} d\tau$$

The integral shows the transfer function $H(\omega)$in the frequency domain of the one, and on the other hand the impulse response $h(\tau)$ of a system. Such relationship is referred to as transformation or a Fourier transform.

Theorem 20 (Fourier Transformation)

$$\mathcal{F}\{s(t)\} = S(\omega) = \int_{-\infty}^{\infty} s(t) \cdot e^{-j\omega t} dt$$

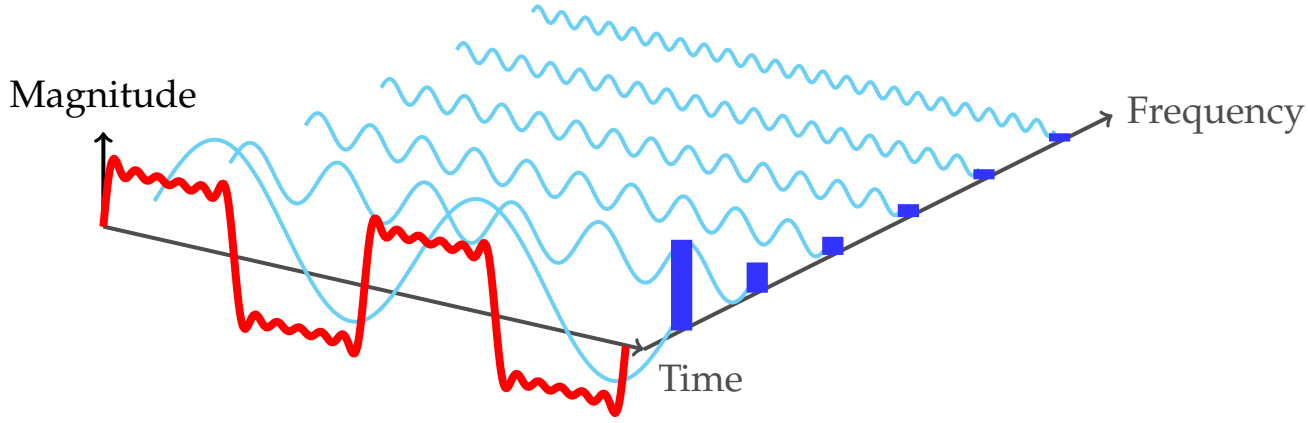


Figure 27: An illustration of the Fourier synthesis. A rectangle function summaruzed of single sine waves.

The inverse Fourier Transformation

We look again at the Formular

$$H(\omega) = \int_{-\infty}^{\infty} h(\tau) \cdot e^{-j\omega\tau} d\tau$$

Under certain circumstances it is possible to recover the time function $s(t)$ from its Fourier transform. This is accomplished exactly by the inverse Fourier transform. Converts a signal from the frequency domain into a signal in the time domain [12]

Theorem 21 (Inverse Fourier Transformation)

$$\mathcal{F}\{s(t)\}^{-1} = s(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} s(\omega) \cdot e^{j\omega t} d\omega$$

We prove now that this definition is actually the inverse function to the Fourier transformation.

Proof:

[12] The inverse Fourier transformation is also called the continuous Fourier synthesis, and the forward transformation is often called Fourier analysis.

We start with the general representation of the inverse Fourier transformation

$$s(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} s(\omega) \cdot e^{j\omega t} d\omega.$$

The approach is now to insert the definition of the inverse Fourier transformation into the Fourier Transformation:

$$S(\omega) = \int_{-\infty}^{\infty} s(t') \cdot e^{-j\omega t'} dt'$$

This results in the relationship

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega) \cdot e^{j\omega t} d\omega = \frac{1}{2\pi} \cdot \left(\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s(t') \cdot e^{-j\omega t'} dt' \cdot e^{j\omega t} d\omega \right).$$

Now you can simplify the presentation further by skillfully combining:

$$\frac{1}{2\pi} \cdot \left(\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s(t') \cdot e^{-j\omega t'} \cdot e^{j\omega t} dt' d\omega \right) = \frac{1}{2\pi} \cdot \left(\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s(t') \cdot e^{j\omega(t-t')} dt' d\omega \right)$$

After Fubini's theorem we can interchange the integration order and receive:

$$\frac{1}{2\pi} \cdot \left(\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s(t') \cdot e^{j\omega(t-t')} d\omega \, dt' \right).$$

In order to simplify the equation somewhat, we take a different mathematical description of the delta pulse $\delta(t)$:

$$\delta(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-j\omega t} d\omega$$

$$\delta(t' - t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-j\omega(t-t')} d\omega,$$

Since $s(t')$ is constant with respect to the integration to ω, one can extract the function from the second integral. Thus we finally obtain the expression

$$\frac{1}{2\pi} \cdot \left(\int_{-\infty}^{\infty} s(t') \int_{-\infty}^{\infty} e^{j\omega(t-t')} d\omega \, dt \right) = \int_{-\infty}^{\infty} s(t') \cdot \delta(t' - t) dt' = s(t).$$

■

Example:
To calculate is the Fourier transform of the shifted Delta pulse:

$$s(t) = \delta(t - t_0).$$

One put the given signal into the Fourier transformation and get the term

$$\mathcal{F}(st) = \int_{-\infty}^{\infty} s(t) \cdot e^{-j\omega t} dt = \int_{-\infty}^{\infty} \delta(t - t_0) \cdot \underbrace{e^{-j\omega t}}_{f(t)} dt.$$

with the sifting property

$$\int_{-\infty}^{\infty} f(t) \cdot \delta t dt = f(0)$$

and the relation of the shifted Delta pulse

$$\delta(t - t_0) = \begin{cases} \infty & t_0 = t \\ 0 & else \end{cases}$$

results

$$\mathcal{F}\{s(t)\} = e^{-j\omega t_0}$$

The sinc Function

To calculate is the Fourier transform of the rectangular pulse

$$s(t) = rect(t) = \begin{cases} 0 & |t| \geq 1/2 \\ 1 & |t| \leq 1/2 \end{cases}$$

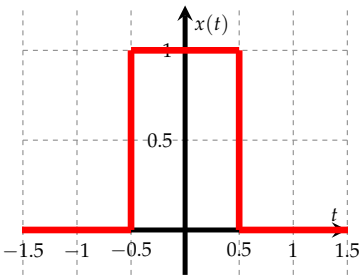


Figure 28: The *rect* function.

by use of the Fourier transformation

$$\mathcal{F}\{s(t)\} = \int_{-\infty}^{\infty} s(t) \cdot e^{-j\omega t} dt = \int_{-\infty}^{\infty} rect(t) \cdot e^{-j\omega t} dt$$
$$= \int_{-1/2}^{1/2} s(t) \cdot e^{-j\omega t} dt = \left[-\frac{1}{j\omega} \cdot e^{-j\omega t} \right]_{-1/2}^{1/2}$$

and the evaluation of the limits, we obtain

$$\mathcal{F}\{s(t)\} = \left[\frac{1}{\omega} \cdot e^{-j\omega t} \right]_{-1/2}^{1/2} = \frac{j}{\omega} \cdot \left(e^{-\frac{1}{2}j\omega} - e^{\frac{1}{2}j\omega} \right).$$

With the notation $e^{jz} = \cos(z) + j\sin(z)$, we arrive at

$$\mathcal{F}\{s(t)\} = \frac{1}{\omega} \left[\cos\left(\tfrac{1}{2}\omega\right) - j\sin\left(\tfrac{1}{2}\omega\right) - \cos\left(\tfrac{1}{2}\omega\right) - j\sin\left(\tfrac{1}{2}\omega\right) \right]$$

and finally obtain the desired result

$$\mathcal{F}\{s(t)\} = \frac{2}{\omega} \cdot \sin(\tfrac{1}{2}\omega)$$

This example leads us to the definition of the sinc function. [13]

Theorem 22 (The sinc Function)

$$sinc(x) = \frac{\sin(x)}{x}$$

In the information theory and signal processing the normalized sinc function is often used.

$$sinc(x) = \frac{\sin(\pi x)}{\pi x}$$

[13] The sinc function is also called cardinal sine function, or sinus cardinalis or spalt function.

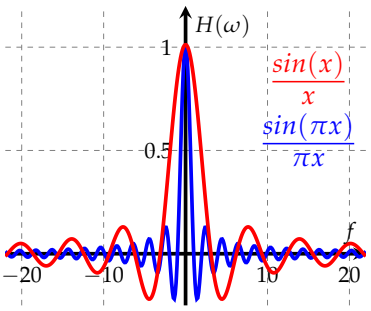


Figure 29: The normalized (red) and not normalized (blue) sinc function.

The Properties of the Fourier Transformation:

The Linearity Property:

A transformation is called linear if the following condition is satisfied:

$$Tr\{a \cdot s_1(t) + b \cdot s_2(t)\} = a \cdot Tr\{s_1(t)\} + b \cdot Tr\{s_2(t)\}$$

Here, s_1 and s_2 are two different signals. a and b are constants. Since integration is linear this propperty applies to the Fourier Transformation.

Theorem 23 (Linearity)

$$\mathcal{F}\{a \cdot s_1(t) + b \cdot s_2(t)\} = a \cdot \mathcal{F}\{s_1(t)\} + b \cdot \mathcal{F}\{s_2(t)\}$$

Example: How helpful the linearity can be in the evaluation of the Fourier transformation, shows the following example. Given are the two signals $s_1(t)$ and $s_2(t)$:

$$s_1(t) = \delta(t - t_0)$$
$$s_2(t) = \delta(t + t_0)$$

we want the Fourier transform of the this two shifted δ functions.

$$s_{ges}(t) = s_1(t) + s_2(t)$$

For this, we use the linearity property:

$$\mathcal{F}\{s_{ges}\} = \mathcal{F}\{\delta(t - t_0) + \delta(t + t_0)\} = \mathcal{F}\{\delta(t - t_0)\} + \mathcal{F}\{\delta(t + t_0)\}$$

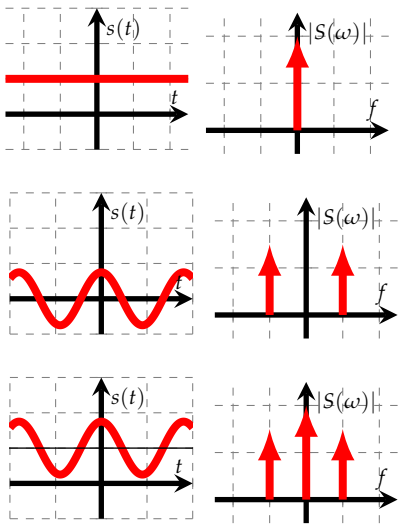


Figure 30: Example application of the linearity theorem by the addition of an constant signal with a cosine function.

By using the sifting property:

$$\int_{-\infty}^{\infty} f(t-t_0)\delta(t)dt = f(t_0)$$

we can specify the individual Fourier tranforms

$$\mathcal{F}\{s_{ges}\} = e^{-j\omega t} + e^{j\omega t}$$

With the Euler identity we get now

$$\mathcal{F}\{s_{ges}\} = \cos(-\omega t_0) + j\sin(-\omega t_0) + \cos(\omega t_0) + j\sin(\omega t_0).$$

Finally, we arrive at the desired result

$$\mathcal{F}\{s_{ges}\} = 2 \cdot \cos(\omega t_0)$$

The Time Scaling:

The multiplication of a given function with constant in the time scale results in:

Theorem 24 (Time Scaling)

$$s(bt) \circ\!\!-\!\!\bullet \frac{1}{|b|} \cdot S\left(\frac{\omega}{b}\right)$$

The scaling theorem states among other things that broad signals have narrow spectrums and peak signals have broad spectrums.

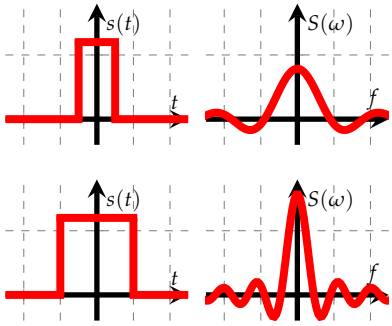


Figure 31: Time scaling.

Proof:
We assume that $b > 0$ is, and get

$$\mathcal{F}\{s(bt)\} = \int_{-\infty}^{\infty} s(bt) \cdot e^{-j\omega t} dt$$

Using the substitution $t' = bt$ results in

$$\begin{aligned}\mathcal{F}\{s(bt)\} &= \int_{-\infty}^{\infty} s(t') \cdot e^{-j\omega \frac{t'}{b}} \cdot \frac{dt'}{b} \\ &= \frac{1}{b} \int_{-\infty}^{\infty} s(t') \cdot e^{-} j\omega dt' \\ &= \frac{1}{b} \cdot S(\frac{\omega}{b})\end{aligned}$$

The proof is analogous for $b < 0$. ■

The Time Shifting:

The time shift of a signal causes a multiplication in the frequency domain.

Theorem 25 (Time Shifting)

$$s(t-t_0) \circ\!\!-\!\!\bullet\, S(\omega)\cdot e^{-j\omega t_0}$$

The time shifting of a function changes under no circumstances the absolute value of the associated spectrum, only the phase undergoes a change.

Proof:
Given is a shifted signal $s(t-t_0)$. We substitute $t' = t - t_0$ and get:

$$\mathcal{F}\{s(t-t_0)\} = \int_{-\infty}^{\infty} s(t-t_0)\cdot e^{-j\omega t}dt = \int_{-\infty}^{\infty} s(t')\cdot e^{-j\omega(t'+t_0)}dt'.$$

After exclude the e function

$$\mathcal{F}\{s(t-t_0)\} = \int_{-\infty}^{\infty} s(t')\cdot e^{-j\omega t_0}\cdot e^{-j\omega t'}dt'.$$

we extract $e^{-j\omega t}$ from the integral and get

$$\mathcal{F}\{s(t-t_0)\} = e^{-j\omega t_0}\cdot \int_{-\infty}^{\infty} s(t')\cdot e^{-j\omega t'}dt' = S(\omega)\cdot e^{-j\omega t_0}.$$

■

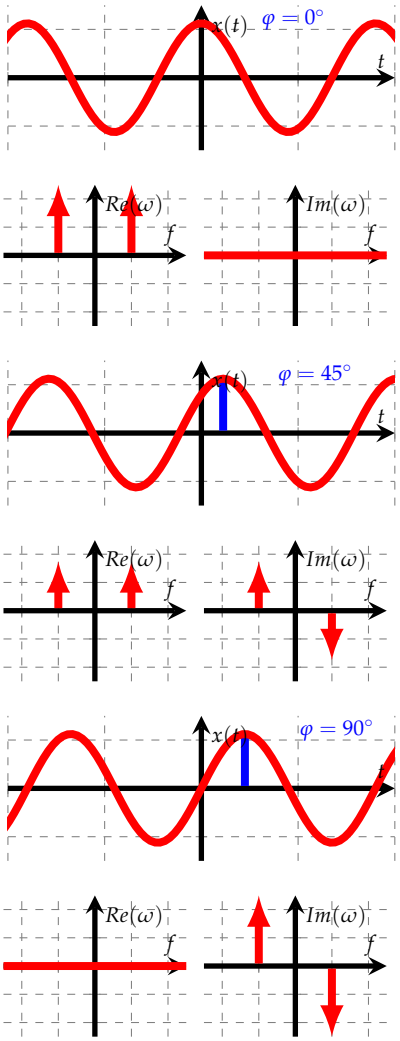


Figure 32: Phase change while Time shifting.

The Differentiation Property:

Another important property of the Fourier transform of the derivative theorem. He indicates that the differentiation in the time domain causes a scaling in the frequency range with $j\omega$:

Theorem 26 (Derivative Property)

$$\frac{d}{dt}s(t) \circ\!\!-\!\!\bullet\, j\omega\cdot S(\omega)$$

Proof:
Now we want to prove the derivation theorem. We are looking for the Fourier transform of a derived signal:

$$\mathcal{F}\left\{\frac{d}{dt}s(t)\right\} = \int_{-\infty}^{\infty}\frac{d}{dt}s(t)\cdot e^{-j\omega f}dt$$

By partial integration

$$\int_a^b f(t) \cdot g'(t)dt = [f(t) \cdot g(t)]_a^b - \int_{-\infty}^{\infty} f(t') \cdot g(t)dt$$

we get

$$\mathcal{F}\left\{ \frac{d}{dt}s(t) \right\} = \underbrace{\lim_{\alpha \to 0}[s(t) \cdot e^{-j\omega t}]_{-a}^{a}}_{=0} - \int_{-\infty}^{\infty} s(t) \cdot \frac{d}{dt}e^{-j\omega t}dt.$$

after the derivation of the e function

$$\mathcal{F}\left\{ \frac{d}{dt}s(t) \right\} = -\int_{-\infty}^{\infty} s(t) \cdot (-j\omega \cdot e^{-j\omega t})dt.$$

we get

$$\mathcal{F}\left\{ \frac{d}{dt}s(t) \right\} = -j\omega \cdot S(\omega)$$

■

Example:
Calculate the Fourier transform of the derived delta pulse $\frac{d}{dt}\delta(t)$.

$$\mathcal{F}\left\{ \frac{d}{dt}\delta(t) \right\} = j\omega \cdot \underbrace{\mathcal{F}\{\delta(t)\}}_{=1} = j\omega$$

The Duality:

Theorem 27 (Duality)

$$s(t) \circ\!\!-\!\!\bullet\, S(\omega) \quad \textit{and}$$
$$S(t) \circ\!\!-\!\!\bullet\, 2\pi \cdot s(-\omega)$$

Make two functions $s(t)$ and $S(\omega)$ is a transform pair, then $S(t)$ and $2\pi \cdot s(-\omega)$ is also a transform pair. This is called dual correspondence.

Proof:
The following calculation steps are intended to illustrate how we get to the understand that $s(t) \circ\!\!-\!\!\bullet\, S(\omega)$ and $S(t) \circ\!\!-\!\!\bullet\, 2\pi \cdot s(-\omega)$ applies. The Fourier transformation is defined as

$$S(\omega) = \int_{-\infty}^{\infty} s(t) \cdot e^{-j\omega t}dt$$

our inverse Fourier transformation

$$s(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} S(\omega) \cdot e^{j\omega t}dt$$

differs only minimal from the forward Fourier transformation. Substitution of $t \to -t$ results in

$$s(-t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega) \cdot e^{-j\omega t} dt$$

If one exchanges now the variables t and ω, then one arrives too

$$2\pi \cdot s(-\omega) = \int_{-\infty}^{\infty} S(t) \cdot e^{-j\omega t} dt.$$

■

The Symmetry Property:

Theorem 28 (Symmetry)
For all real signals does apply:

$$S(-\omega) = \overline{S(\omega)} \quad \forall \omega$$

This means that the real part of the Fourier transform of a real signal is always even and the imaginary part is always odd. This property is known as Hermite Symmetrie. Is additionally $s(t)$ is even, then the imaginary part of the Fourier transform disappears $S(\omega)$. Is $s(t)$ odd, then the real part disappears.

	Function		Transform	
	Complex	**Symmetry**	**Complex**	**Symmetry**
1	real	even	real	even
2	real	odd	imaginary	odd
3	imaginary	even	imaginary	even
4	imaginary	odd	real	odd
5		even		even
6		odd		odd
5	complex	even	complex	even
6	complex	odd	complex	odd
7	real		real even, imaginary odd	
8	imaginary		real odd, imaginary even	

The Convolution Theorem:

Now we consider the convolution theorem:

Theorem 29 (Convolution)

$$s_1(t) * s_2(t) \circ\!\!-\!\!\bullet\, S_1(\omega) * S_1(\omega)$$

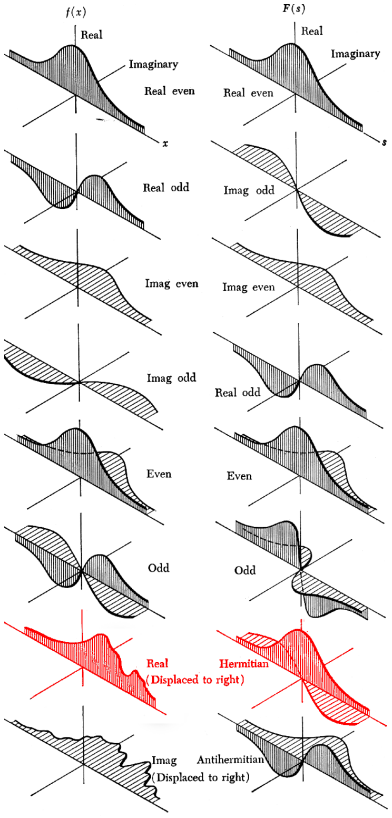


Figure 33: Fourier symmetry relations.

Table 4: The table of symmetry relations of the Fourier transformation can be used in both directions. The most common Hermitian symmetry has No. 7.

Folding algebra is transformed into a multiplication algebra. This leads to a significant simplification in the treatment of LTI systems. We get the representation from the duality of the Fourier transform.

Proof:
We now prove the convolution theorem. For this purpose, one recalls the definition of the convolution

$$\int_{-\infty}^{\infty} a(t-\tau)\cdot b(\tau)d\tau$$

and thus get

$$\begin{aligned}\mathcal{F}\{s_1(t)*s_2(t)\} &= \int_{-\infty}^{\infty}\left(\int_{-\infty}^{\infty} s_1(t-\tau)\cdot s_2(\tau)d\tau\right)\cdot e^{-j\omega t}dt \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s_1(t-\tau)\cdot s_2(\tau)\cdot e^{-j\omega t}d\tau\,dt\end{aligned}$$

we substitute $t' = t$ and get

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s_1(t')\cdot s_2(\tau)\cdot e^{-j\omega(t'+\tau)}d\tau\,dt'.$$

With $e^{j\omega(t'+\tau)} = e^{-j\omega t'}\cdot e^{-j\omega\tau}$ we can pull apart the integrals

$$= \int_{-\infty}^{\infty} s_1(t')e^{-j\omega t'}dt'\cdot\int_{-\infty}^{\infty} s_1(\tau)e^{-j\omega\tau}d\tau.$$

Thus we obtain

$$\mathcal{F}\{s_1(t)*s_2(t)\} = \mathcal{F}\{s_1(t)\}\mathcal{F}\{s_2(t)\}$$

■

The Multiplication Theorem:

The multiplication theorem [14] can be derived using the convolution theorem. With the addition of duality we obtain the following property:

Theorem 30 (Multiplication)

$$s_1(t)\cdot s_2(t) \circ\!\!-\!\!\bullet \frac{1}{2\pi}S_1(\omega)*S_1(\omega)$$

The convolution theorem and the multiplication theorem represent two fundamental relations in the system theory. A convolution in the time

[14] The multiplication property is also called (amplitude) modulation.

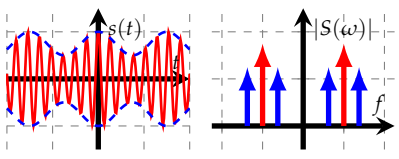


Figure 34: Example of the multiplication property with two sinus signals. The carrier is red and the signal is blue.

domain corresponds to multiplication in the frequency domain, and the multiplication in the time domain corresponds to the convolution in the frequency domain. Both theorems allow to pass the rather complicated folding variant in one domain to a simple multiplication variant in the other domain.

The Parseval Theorem:

The Parseval theorem reflects the similarity between the power of two signals in the time domain power, and the power of two signals in the frequency domain.

$$\int_{-\infty}^{\infty} s_1(t) \cdot s_2^* \frac{1}{2\pi} \int_{-\infty}^{\infty} S_1(\omega) \cdot S_2^*(\omega) d\omega$$

Are both signals $s_1(t)$ $s_2(t)$ equal, thus applies:

$$\int_{-\infty}^{\infty} |s(t)|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |S(\omega)|^2 d\omega$$

$|s(\omega)|^2$ is called energy spectrum.

Theorem 31 (Signal Energy)

$$E = \int_{-\infty}^{\infty} |s(t)|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |S(\omega)|^2 d\omega$$

While Parseval's relation is interesting from the physics it describes (conservation of energy), it has few practical uses in DSP.

Proof:
We prove now the Parseval theorem. The prove begins with the symmetry approach

$$s_2^*(t) \circ\!\!-\!\!\bullet\, S_2^*(-\omega)$$

one gets the signal energy

$$\int_{-\infty}^{\infty} s_1(t) \cdot s_2^*(t) \cdot e^{-j\omega' t} dt.$$

By usage of the multiplication theorem one get

$$\begin{aligned} \int_{-\infty}^{\infty} s_1(t) \cdot s_2^*(t) \cdot e^{-j\omega' t} dt &= \frac{1}{2\pi} \cdot S_1(\omega') * S_2^*(\omega') \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} S_2^*(-(\omega' - \omega)) \cdot S_1(\omega) d\omega. \end{aligned}$$

If we set $\omega' = 0$, the equation changes to

$$\int_{-\infty}^{\infty} s_1(t) \cdot s_2^*(t) dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_1(\omega) \cdot S_2^*(\omega) d\omega$$

■

Table of Transformation Pairs and Properies

Description	**Time Domain**	**Frequency Domain**
time reversal	$s(-t)$	$S(-\omega)$
conjugation	$\overline{s(t)}$	$\overline{S(-\omega)}$
delta pulse	$\delta(t)$	1
shifted delta pulse	$\delta(t - t_0)$	$e^{-j\omega t_0}$
heaviside unit jump	$\Theta(t)$	$\frac{1}{j\omega} + \pi\delta(\omega)$
triangle	$\Delta(t)$	$sinc^2(\omega/2\pi)$
time shift	$s(t - t_0)$	$F(\omega)e^{-j\omega t_0}$
frequency shift	$s(t)e^{j\omega_0 t}$	$S(\omega - \omega_0)$
time scaling	$s(\alpha t)$	$\frac{1}{\lvert\alpha\rvert} S\left(\frac{\omega}{\alpha}\right)$
duality	$S(t)$	$2\pi s(-\omega)$
time derivation	$\frac{d^n f(t)}{dt^n}$	$(j\omega)^n S(\omega)$
frequency derivation	$(-j\omega)^n s(t)$	$\frac{d^n S(\omega)}{d\omega^n}$
integration	$\int_{-\infty}^{t} s(\tau) d\tau$	$\frac{S(\omega)}{j\omega} + \pi F(0)\delta(\omega)$
modulation	$e^{j\omega_0 t}$	$2\pi\delta(\omega - \omega_0)$
signum	$sgn(t)$	$\frac{2}{j\omega}$
rectangle	$rect\left(\frac{t}{\tau}\right)$	$\tau sinc\left(\frac{\omega\tau}{2}\right)$
sinc	$\frac{B}{2\pi} sinc\left(\frac{Bt}{\tau}\right)$	$rect\left(\frac{\omega}{B}\right)$
sinus	$\cos(\omega_0 t)$	$\frac{\pi}{j}[\delta(\omega - \omega_0) - \delta(\omega + \omega_0)]$
cosinus	$\sin(\omega_0 t)$	$\pi[\delta(\omega - \omega_0) + \delta(\omega + \omega_0)]$
exponential	$e^{-\alpha\lvert t\rvert}$	$\frac{2\alpha}{\alpha^2 + \omega^2}$

Table 5: Table of Fourier transformation pairs and properties.
Note: The *sinc* function is defined as the normalized *sinc* function $sinc(t) = \frac{\sin(\pi t)}{\pi t}$. The rectangle function $rect(t)$ reach over the interval from $-1/2$ to $1/2$, and the triangle function $\Delta(t)$ is defined in the interval from -1 to 1.

Window Functions:

Sometimes signals can be stubbornly non-periodic. This can make digital signal processing so much more annoying. The supposedly simple act of plotting the spectrum ends up being a game of trial-and-error, or requires making a decision that involves prior knowledge of what one would actually like to know in the first place. We'll have a look at why this is so and what to do about it.

About the Leakage and the Dynamic Range: Basically, leakage is a consequence of looking at only a finite segment of a signal instead of the whole signal extending over all time, from negative to positive infinity. In the most simple case, the signal segment $s(t)$ is isolated by multiplying a signal $s(t)$ with a box function such that

$$s'(t) = s(t) \cdot w_{rect}(t) \quad with \quad w_{rect}(t) = \begin{cases} 1 & -T/2 \leq t \leq T/2 \\ 0 & else \end{cases}$$

This multiplication in the time domain corresponds to a convolution in the frequency domain, i.e. each frequency component contained in $s(t)$ is convolved with the Fourier transform of the box function, which is

$$I_{rect}(f) = T\ sinc(fT)$$

where the tilde denotes a Fourier-transformed quantity. Part of the energy contained in that frequency component is distributed over the entire frequency domain. This is referred to as leakage. One disadvantage of such strong leakage is that small signals in the vicinity of strong signals can be completely masked if the leakage energy at some frequency is larger than the signal energy at that frequency. This greatly reduces the dynamic range of the spectral analysis. The box window has a very low dynamic range.

Windows: As long as the segment $s(t)$ is to be of finite duration, leakage cannot be prevented. However, by using functions other than the box, the distribution of leakage energy in the spectrum can be shaped, since that is described by the Fourier transform of the window. To make things a bit simpler, we will normalize window functions so that $T = 1$. We call their Fourier transforms instrument functions. Figures 37 through 43 illustrate $w_{Rect}(t)$ and its instrument function, as well as the Hann and flat-top windows, two often used windows. The energy spectral density (ESD), shown in the figures, is defined as the squared magnitude of the instrument function. The Hann window is defined as

$$w_{Hann}(t) = \begin{cases} cos^2(\pi t) & -T/2 \leq t \leq T/2 \\ 0 & else \end{cases}$$

which is actually the product of a raised cosine function with the box window $w_{Rect}(t)$. No matter the function that describes the shape of the window, it must always be bounded in time by multiplication with the box - that is also the reason why there is always leakage: the instrument function of $w_{Hann}(t)$ is the convolution of the infinitely extended $I_{Rect}(f)$ with another function (most often it's a number

of Dirac delta peaks resulting from a sum of cosines). The Flat-Top window is often defined as

$$w_{Flat} = \begin{cases} \sum_{n=0}^{4} a_n \cos 2n\pi t & -T/2 \leq t \leq T/2 \\ 0 & else \end{cases}$$

$$with \qquad a_0 = 0.21557895 \qquad a_1 = 0.41663158$$

$$a_2 = 0.277263158 \qquad a_3 = 0.083578947 \qquad a_4 = 0.006947368$$

The side lobes of both, the Hann window and the Flat-Top window, are significantly lower than those of the Rectangular window, increasing the dynamic range in the vicinity of a strong peak. In the case of the Hann window they also fall off much more rapidly, their slope is steeper. The flat-top falls off more slowly, but it has a very large dynamic range over a broad spectral range.

Resolution: Comparing the three windows, we also notice that their main lobe width differs significantly. Main lobe width can be defined either as the 3dB (half-)width or the width to the first spectral zero. Within the main lobe the dynamic range is very low - if two spectral components of a signal are separated by less than their main lobe (3dB) width, they are virtually indistinguishable. The minimum separation for detectability determines the resolution of the window. The box window has the highest possible resolution (at the expense of low dynamic range). The Hann window is a middle-ground trade-off between resolution and dynamic range. The Flat-Top window has very low resolution owing to its significant main lobe width. Other windows usually weight resolution and dynamic range somewhere between the box and the Flat-Top.

Discrete Signals: Spectral leakage occurs for continuous and discrete signals alike, as long as they are not of infinite duration. However, there are some interesting peculiarities when working with discrete signals (such as the apparent absence of leakage in some signals), for which reason we'll look at these a bit more in detail. The discrete Fourier transform (DFT) takes a discrete sample sequence and transforms it into another discrete sample sequence. If the first sequence is interpreted as consecutive samples of a time signal (segment), the second sequence corresponds to the sampled Fourier spectrum of the time signal, with some caveats.2 When there are N time samples within the period T, the sample duration is

$$\Delta t = \frac{T}{N}$$

The resulting frequency range spanned by the discrete spectrum and spacing between frequency samples (or "bins") are

$$F = \frac{1}{\Delta t} = \frac{N}{T} \qquad \Delta f = \frac{1}{T}$$

Looking at the instrument function of the Rectangle window, we see that it has zeros at all non-zero integer multiples of $1/T = \Delta f$ - which is the same spacing as between the samples of the DFT frequency spectrum. This means that if a spectral component of $s(t)$ corresponds exactly to one of the frequencies sampled by the DFT spectrum, all other frequency samples coincide with zeros of the box window instrument function, and thus it appears as if there is no leakage at all! However, this is really just an artifact of the sampled nature of the DFT spectrum. The particular frequency components for which this occurs (those at integer multiples of Δf) are those corresponding to an integer number of full oscillations within T. Hence, any signal segment for which all frequency components (and thus the segment itself) are periodically extensible without jumps (for all its frequency components) is transformed leakage-free and thus exact. With real-world signals, the segment $s(t)$ will be properly periodic only in the rarest of cases, and almost always we will have to deal with leakage.

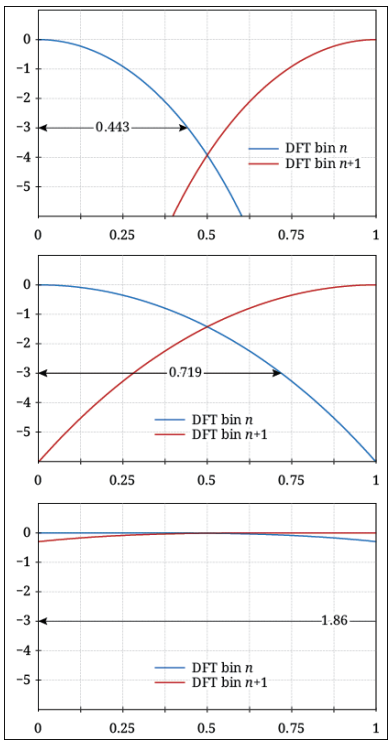


Figure 35: Comparisation of the Box, Hann, and Flat-Top window with respect to scalloping.
Shown is the energy detected in the neighboring DFT bins when the true frequency of the signal lies somewhere in between. Maximum scalloping occurs at a frequency offset of 0.5 bins (thereafter the frequency is closer to the next bin). Also shown is the 3dB half-width of the main lobe.

Scalloping: Another effect peculiar to discrete spectra is called scalloping. If a frequency component of $s(t)$ falls outside any frequency that is sampled by the DFT spectrum, then its convolution with the instrument function is not sampled at the (known) maximum of the resulting curve but slightly offset. The maximum possible offset is half the separation between two DFT samples, or $\frac{\Delta}{2}$. The maximum possible scalloping is thus the instrument function difference between $f = 0$ and $f = \frac{\Delta}{2}$. When scalloping is large then the obtained energy reading of a frequency component depends on the frequency of that spectral component relative to the DFT sampling points. This scalloping is largest for the box window âĂŞ its instrument function falls off rapidly, which also results in its excellent resolution. The Flat-Top window âĂŞ hence the name falls off very slowly, and scalloping is usually not an issue, though it results in poor resolution. Since by means of scalloping the energy in a frequency component is distributed into several DFT bins, it can be used to approximate the exact frequency of that component even if it does not coincide with a DFT bin, if the instrument function is known. When multiple frequency components are superposed in a signal, this is generally not possible, however, since their overlapping contributions to the DFT spectrum cannot be separated. Figure 35 shows the relative energy detected in neighboring DFT bins (and hence the scalloping) when the frequency of a single spectral component is

swept between the bin frequencies for the box, Hann, and Flat-Top windows. The 3dB half-width of the main lobe is also shown.

Agony of Choice: So which window is the best one box, Hann, Flat-Top, or any of the other windows profiled on the overview page or somewhere else? The answer is: that depends. It depends on the kind of signal we want to measure. If the exact energy of the frequency components in the signal is important, windows with very little scalloping such as the Flat-Top are usually the best choice. However, when multiple frequency components can occur close to each other, the low resolution of these windows will blur them into a single peak without usable information about the energy content of the single components. If resolution is important, we usually need to sacrifice dynamic range near strong peaks, and vice versa. The box window offers the highest resolution, but has very low dynamic range. The flat-top has a high dynamic range, but poor resolution. The Hann window balances resolution and dynamic range and is often a good starting point for spectral analysis. Figure 36 illustrates the advantages and disadvantages of our three window types. A time signal is composed of various frequency components (grey peaks in each graph) and analyzed used each window. The energy spectral density values were normalized so that $W(0)$ (the main lobe height) corresponds to one. This is necessary to obtain the correct energy information since each window has a deterministic correction factor that depends on the energy of the window function itself (window functions w(t) that have small values most of the time reduce the energy in the windowed signal) and the leakage properties (the portion of the instrument function energy that is contained in the main lobe and the width of the main lobe determine its height).

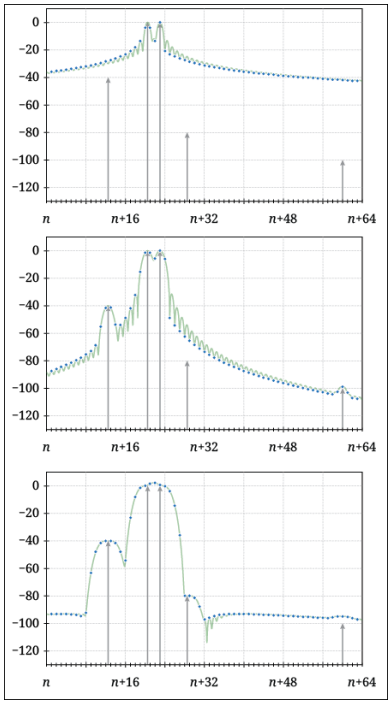


Figure 36: Comparisation of the Box, Hann and Flat-Top window with respect to signal detection ability.
The Grey arrows denote the frequency components in the time signal, the blue dots denote the obtained DFT spectrum, and the green lines illustrate the continuous shape of the spectrum of the windowed signal.

Window Function Overview:

The visual overviews contain some of the more important properties of each window function, such as window shape and total energy, instrument function shape, main lobe height and fraction of total energy E contained therein (E_0/E), suppression of highest side lobe, and maximum scalloping. Where possible, the apodization function is given. Since the window function usually itself is just a segment of an infinitely extended function, it must be bounded. This can be written as

$$w_(t) = \begin{cases} A(t) & -T/2 \leq t \leq T/2 \\ 0 & else \end{cases}$$

where A(t) describes the actual shape of the window and is given below. The apodization functions are normalized such that the window width

$T = 1$. The instrument function corresponding to the un-normalized apodization function $A(t/T)$ is then, according to Fourier transform rules, $T \cdot I(fT)$. The frequency axis is divided into DFT bins which are separated by $\Delta f = T^{-1}$. For the normalized functions, $\Delta f = 1$. Aliasing effects due to finite time-resolution sampling, which make the instrument function dependent on the number of time samples within the window, are not considered.

The Box/Rectangle Function:

$$A_{Rect}(t) = 1$$
$$I_{Rect}(f) = sinc(\pi f)$$

window energy	$0dB$
main lobe height	$0dB$
equivalent noise BW	$1\Delta f (0dB)$
main lobe FWHM	$0.884\Delta f$
first zero	$1\Delta f$
maximum scalloping	$3.92dB$
highest side lobe	$-13.3dB$ *at* $1.43\Delta f$
side lobe slope	$20dB/decade$
notes	This is the window when no window is used, i.e. when the segment is simply cut out of the time signal. It has the best resolution (narrowest main lobe), but highest spectral leakage and scalloping, and lowest dynamic range. Used for detecting strong signals that are spectrally close.

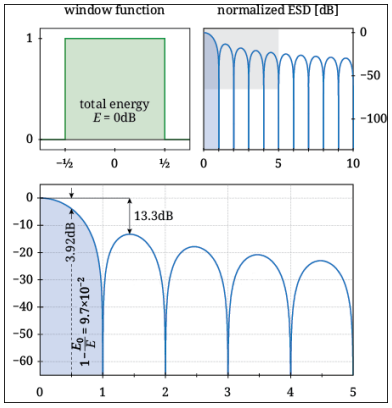


Figure 37: Some properties of the box window and its instrument function.
The window function (top left) is normalized to T = 1. The graph also shows the window energy relative to the box window. The Instrument function (top right) is shown for the spectral region covering the first 10 DFT bins. The close-up (bottom) shows maximum scalloping loss (see below), the relative height of first side lobe, and fraction of energy contained in main lobe, Eo, to total energy E.

The Hann Function:

$$A_{Rect}(t) = \cos^2(\pi t)$$
$$I_{Rect}(f) = \frac{1}{2}\frac{sinc(\pi f)}{1 - f^2}$$

window energy	$-4.62dB$
main lobe height	$-6.02dB$
equivalent noise BW	$1.5\Delta f (1.76dB)$
main lobe FWHM	$1.44\Delta f$
first zero	$2\Delta f$
maximum scalloping	$1.42dB$
highest side lobe	$-31.5dB$ *at* $2.36\Delta f$
side lobe slope	$60dB/decade$
notes	Also called the raised cosine or (wrongly) Hanning window, the Hann window is quite universal due to its balance between good resolution and dynamic range. The first side lobe is still somewhat high, but their slope is steep. The power-of-cosine window family is a generalization of the Hann window with arbitarily steep side lobe slope.

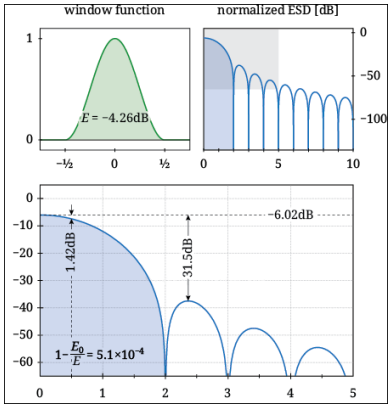


Figure 38: The Hann Window

The Flat-Top Window:

$$A_{Flat}(t) = \sum_{n=0}^{4} a_n \cos(2n\pi t)$$

$$with \qquad a_0 = 0.21557895 \qquad a_1 = 0.41663158$$

$$a_2 = 0.277263158 \qquad a_3 = 0.083578947 \qquad a_4 = 0.006947368$$

window energy	$-7.56dB$
main lobe height	$-13.3dB$
equivalent noise BW	$3.75\Delta f (5.74dB)$
main lobe FWHM	$3.72\Delta f$
first zero	$5\Delta f$
maximum scalloping	$0.00978dB$
highest side lobe	$-93.6dB$ *at* $7.47\Delta f$
side lobe slope	$20dB/decade$
notes	The Flat-Top window has very little scalloping and high dynamic range, but poor resolution, and is thus used to accurately determine the height of single spectral peaks or signals with some separation between peaks. Good for high sampling rates where Δf can be very small.

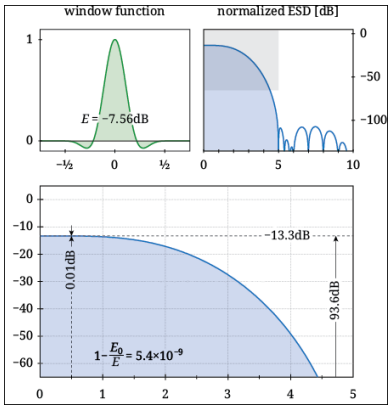


Figure 39: The Flat-Top Window

The Hamming Window:

$$A_{Hamming}(t) = \frac{27}{50} + \frac{23}{50}\cos(2\pi t)$$

$$I_{Hamming}(f) = \left(\frac{27}{50} - \frac{4}{50}f^2\right)\frac{sinc(\pi f)}{1-f^2}$$

window energy	$-4.01dB$
main lobe height	$-5.35dB$
equivalent noise BW	$1.36\Delta f (1.34dB)$
main lobe FWHM	$1.30\Delta f$
first zero	$2\Delta f$
maximum scalloping	$1.75dB$
highest side lobe	$-42.7dB$ *at* $4.50\Delta f$
side lobe slope	$20dB/decade$
notes	With a main lobe still narrower than the Hann window, the Hamming window was designed to minimize the height of the first sidelobe. The side lobe slope is significantly smaller than that of the Hann window. It has a medium dynamic range. The coefficients of the adoptization function may be slightly different depending on the author.

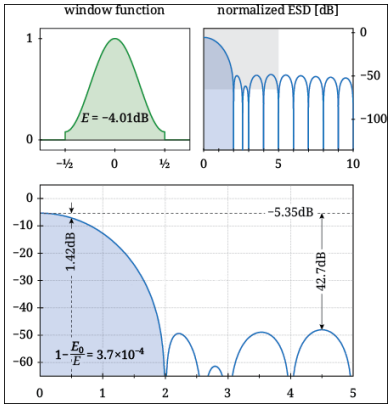


Figure 40: The Hamming Window

The Blackmann-Harris Window:

$$A_{BH}(t) = \sum_{n=0}^{3} a_n \cos(2n\pi t)$$

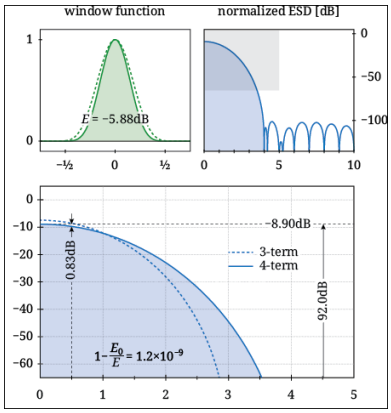


Figure 41: The Blackmann-Harris Window.

window energy	$-5.13dB(3-term), -5.88dB(4-term)$
main lobe height	$-7.35dB, -8.90dB$
equivalent noise BW	$1.70\Delta f(2.31dB), 2.00\Delta f(3.02dB)$
main lobe FWHM	$1.62\Delta f, 1.90\Delta f$
first zero	$3\Delta f, 4\Delta f$
maximum scalloping	$1.14dB, 0.826dB$
highest side lobe	$-71.5dB$ *at* $3.64\Delta f, -92.0dB$ *at* $4.52\Delta f$
side lobe slope	$20dB/decade, 20dB/decade$
notes	The Blackman-Harris windows come mainly in two flavors: a three-term and a four-term sum of cosines. Their coefficients are designed to minimized side lobe levels.

The Gaussian Window:

$$A_{Gauss}(t) = e^{(-\frac{t^2}{2\sigma^2})}$$

$$I_{Gauss}(f) = \sqrt{\frac{\pi}{2}}\sigma e^{(-2\pi^2\sigma^2 f^2)}\left[erf\left(\frac{\frac{1}{2}-2if\pi\sigma^2}{\sqrt{2}\sigma}\right)+erf\left(\frac{\frac{1}{2}-2if\pi\sigma^2}{\sqrt{2}\sigma}\right)\right]$$

window energy	$-4.51dB(\sigma=0.20), -3.56dB(\sigma=0.25), -2.82dB(\sigma=0.30)$
main lobe height	$-8.90dB, -4.46dB, -3.35dB$
equivalent noise BW	$2.75\Delta f(4.39dB), 1.23\Delta f(0.90dB), 1.13\Delta f(0.53dB)$
main lobe FWHM	$1.89\Delta f, 1.18\Delta f, 1.08\Delta f$
first zero	$3.2\Delta f, 1.67\Delta f, 1.38\Delta f$
maximum scalloping	$1.58dB, 2.13dB, 2.54dB$
highest side lobe	$-43.3dB$ *at* $3.65\Delta f, -31.9dB$ *at* $2.62\Delta f, -25.0dB$ *at* $1.70\Delta f$
side lobe slope	$20dB/decade, 20dB/decade$
notes	Better results may be obtained when the Gaussian window is truncated (multiplied) with another window instead of the box function.

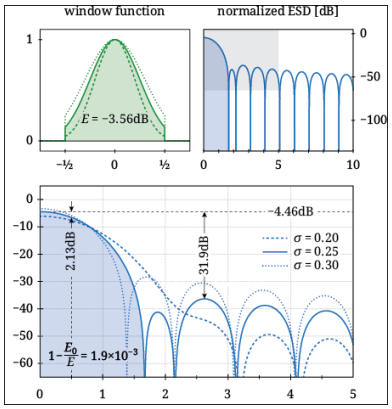


Figure 42: The Gaussian Window.

The Kaiser Window:

$$A_{Kaiser}(t) = \begin{cases} \frac{I_0\left(\pi\alpha\sqrt{1-\left(\frac{2n}{N-1}\right)^2}\right)}{I_0(\pi\alpha)} & 0 \le n \le N-1 \\ 0 & otherwise \end{cases}$$

No analytic closed-form apodization function is available. I_0 is the zeroth order modified Bessel function of the first kind. α is an arbitrary, non-negative real number that determines the shape of the window. In the frequency domain, it determines the trade-off between main-lobe width and side lobe level, which is a central decision in window design.

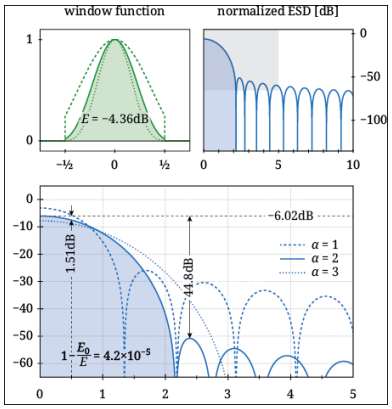


Figure 43: The Kaiser Window.

window energy	$-2.61dB(\alpha = 1), -4.36dB(\alpha = 2), -5.30dB(\alpha = 3)$
main lobe height	$-3.09dB, -6.02dB, -7.78dB$
equivalent noise BW	$1.12\Delta f(0.48dB), 1.47\Delta f(1.66dB), 1.77\Delta f(2.48dB)$
main lobe FWHM	$1.07\Delta f, 1.40\Delta f, 1.68\Delta f$
first zero	$1.35\Delta f, 2.17\Delta f, 3.11\Delta f$
maximum scalloping	$2.59dB, 1.51dB, 1.05dB$
highest side lobe	$-22.9dB$ *at* $1.69\Delta f, -44.8dB$ *at* $2.39\Delta f, -69.7dB$ *at* $3.26\Delta f$
side lobe slope	$20dB/decade$
notes	The discrete prolate spheroidal sequence window is optimized to have maximum energy in its main lobe for a given time-bandwidth product. The (real-valued) parameter α corresponds to half this time-bandwidth product and also gives the largest DFT bin number that is still inside the main lobe. The Kaiser window is an approximation to the DPSS window using Bessel functions. The continuous equivalent, the Slepian window, is described by the first prolate spheroidal wave function.

The Discrete Fourier Transformation

The Fast Fourier Transformation

The Inverse Fast Fourier Transformation can be expressed in therms of the forward FFT with the transformation of the scaled and conjugated $x(n)$:

$$x(n) = \mathcal{IFFT}\{X(f)\} = \mathcal{FFT}^{-1}\{X(f)\} = \frac{\mathcal{FFT}\{X^*(f)\}}{N}$$

How To Perform the FFT and iFFT:

In the next chapter i describe how the FFT algorithm practically is used and how to extract the frequency from the FFT result.

*The classical standard **forward** FFT is performed in the following steps:*

- Apply the window function to the time domain input signal.
- If the signal has not a length from a power of two (2^n for $n \in \mathbb{N}$), we have to *"zero pad"* the vector. This means that the length of the allocated vector must have a length from the next power of two, and the rest will be filled with zeros. We have to note the scaling factor $s_N = \frac{N}{next\,2^n}$ for later descaling.
- Allocate the vector for the imaginary part of the time domain signal with the same length (from a power of two) as for the real signal and fill it with zeros.

- Perform the FFT with the complex time domain input signal.
- Descale the output vector with s_N
- Due the symmetry property of the Fourier transformation, the resulting frequency domain spectrum is mirrored around the middle point $\frac{N}{2} - 1$. The frequency bins are arranged in the following order:(See the Figure 46 and 47 for details)

$$re = \{f_0, f_1, f_2, ..., f_{\frac{n}{2}-2}, f_{\frac{n}{2}-1}, f_{\frac{n}{2}-1}, f_{\frac{n}{2}-2}, ..., f_2, f_1, f_0\}$$
$$im = \{f_0, f_1, f_2, ..., f_{\frac{n}{2}-2}, f_{\frac{n}{2}-1}, -f_{\frac{n}{2}-1}, -f_{\frac{n}{2}-2}, ..., -f_2, -f_1, -f_0\}$$

- The FFT results a $re = \mathrm{Re}(X(f))$ and an $im = \mathrm{Im}(X(f))$ vector. We can now extract the magnitude spectrum element wise with $|X(f)| = \sqrt{re^2 + im^2}$, and the phase spectrum with $\varphi = arctan(\frac{im}{re})$. We need only the first half of the complex frequency domain vectors for this calculation.

$$mag = \{f_0, f_1, f_2, ..., f_{\frac{n}{2}-2}, f_{\frac{n}{2}-1}\}$$

15

*The classical standard **inverse** FFT is computed in the following steps:*

- We assume that the input length of the frequency domain input vectors has length from a power of two.
- Conjugate the frequency domain input vector by inversion of the imaginary part $\mathrm{Im}(X(f)) = -\mathrm{Im}(X(f))$.
- Perform the forward FFT with the conjugated input vectors and multiply each element with with $\frac{1}{N}$.
- Scale the time domain output vectors

[15] Note that the frequency resolution is *virtually* increased with the zero padding because more samples are used for the transformation. The additional frequency points are implicitly interpolated by the FFT and do not affect the real frequency resolution.

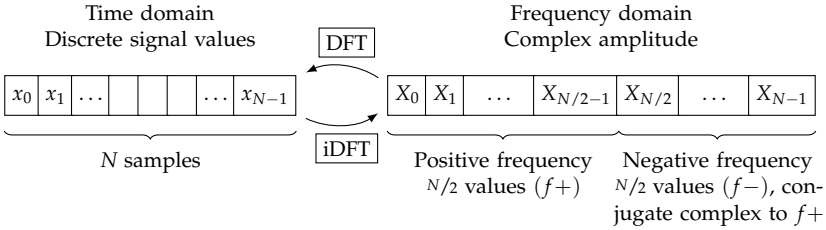


Figure 44: Time and frequency value bins of the discrete/fast Fourier transformation.

Common Pitfalls with the FFT:

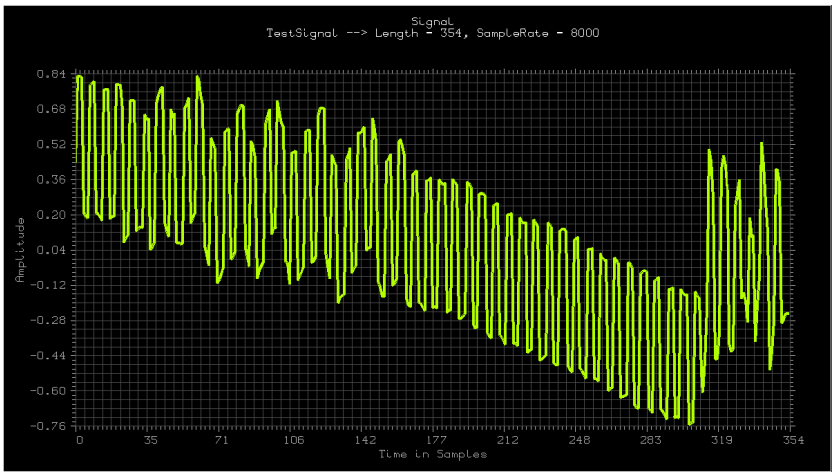


Figure 45: The time domain real input signal (The imaginary input signal vector is set to zero).

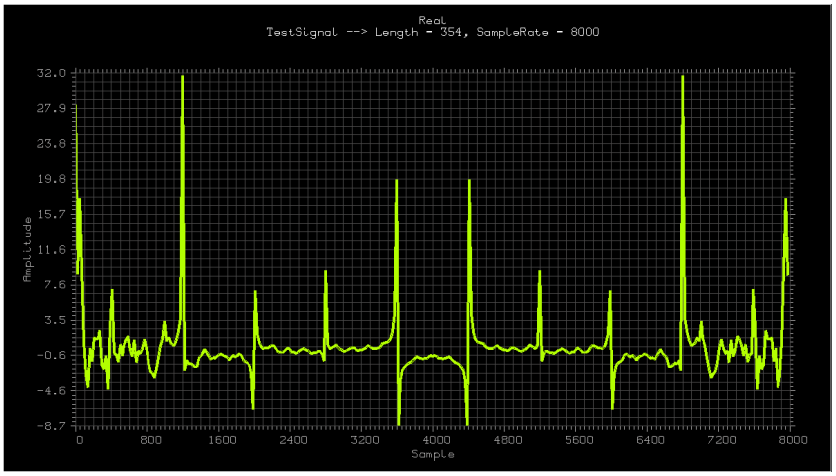


Figure 46: The real part of the complex FFT frequency domain signal.

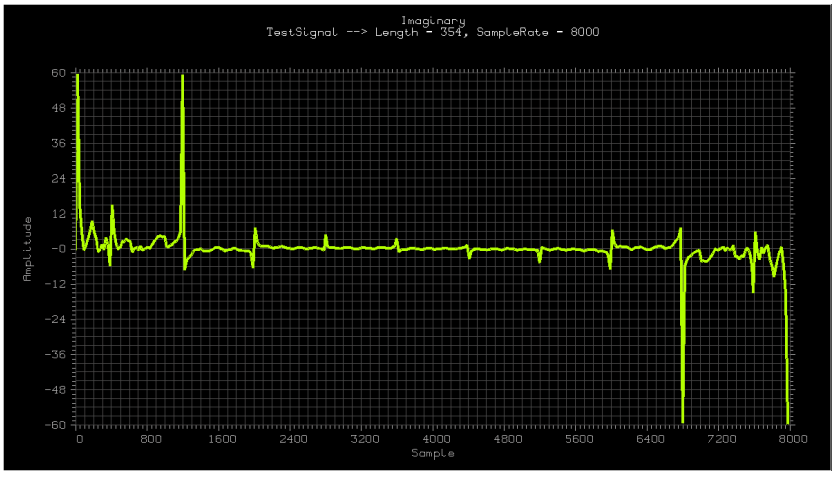


Figure 47: The imaginary part of the complex FFT frequency domain signal.

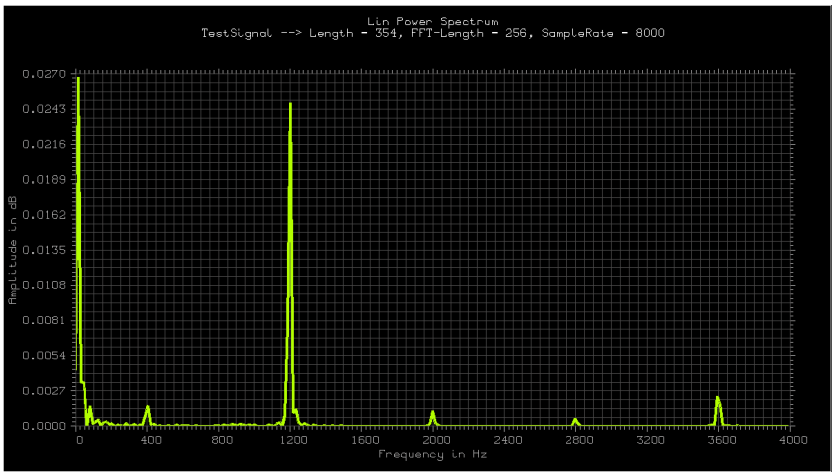


Figure 48: The magnitude spectrum in linear scale.

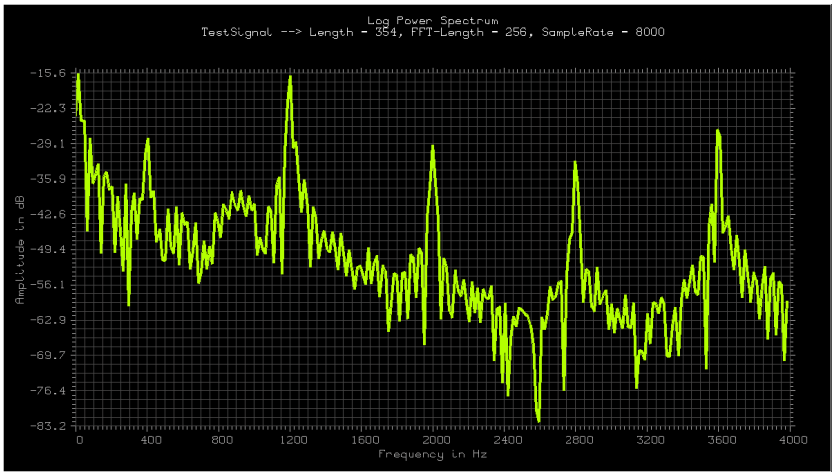


Figure 49: The magnitude spectrum in logarithmic scale. (Note that the magnituge spectrum has only the half of the size than the *re* and *im* parts of the complex spectrum.)

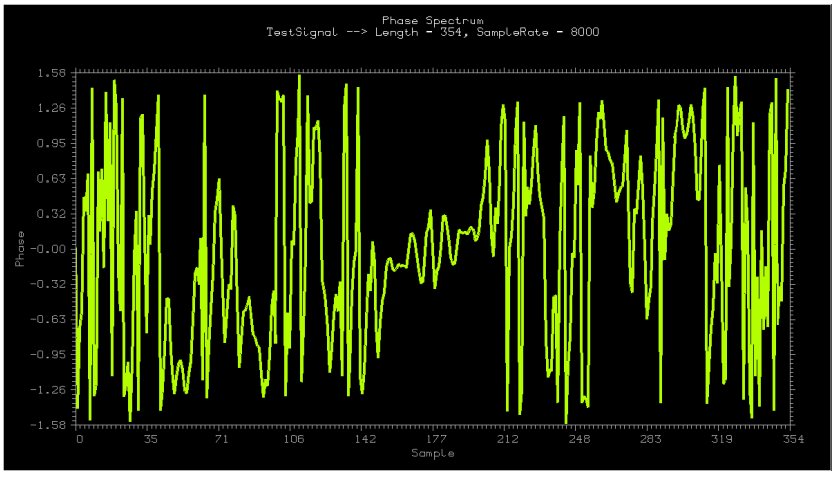


Figure 50: The phase spectrum.

The Chirp-Z Transformation: Bluestein-FFT-Algorithmus

The Magnitude Spectrum foo foo foo

The Phase Spectrum foo foo foo

The LPC Spectrum foo foo foo

The Cosine Transformation

foo foo foo

The Laplace Transformation

foo foo foo

The Z-Transformation

foo foo foo

The Hilbert Transformation

foo foo foo

Envelope extraction foo foo foo

Transformations using Integral Transformations

Cepstrum analysis

foo foo foo

Mel Frequency Cepstral Coefficients foo foo foo

The Harmonic Analyze

foo foo foo

Relations between frequency-domain and time-domain based methods

foo foo foo

Chapter 3 - The Time-Frequency Analysis

Section 3 - Fullwidth Environment Example

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Windowing

foo foo foo

The Short Time Fourier Analysis

The Order Analysis

foo foo foo

The Wavelet Transformation

foo foo foo

The Wavelet Package Decomposition

The Cohen's Class Distributions

The Pseudo Smoothed Wigner-Ville Transformation

foo foo foo

The Choi-Williams Transformation

foo foo foo

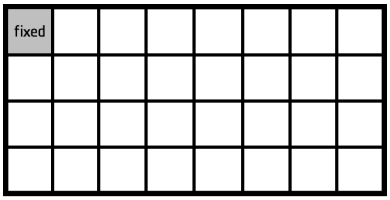


Figure 51: The short time fourier transformation.

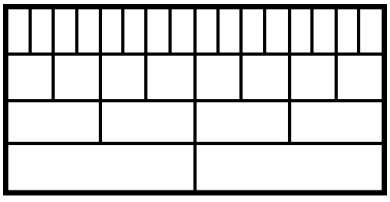


Figure 52: The Wavelet transformation.

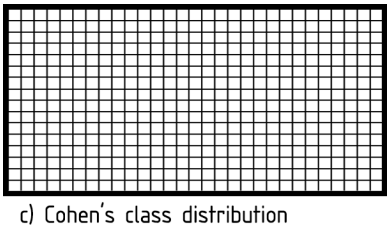


Figure 53: The Cohen's Class Distribution.

Chapter 4 - Signal Classification

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

The Kalman Filter

foo foo foo

Chapter 5 - Other Techniques

Pre-Emphasis

`Y[i] = X[i] - (X[i-1] * preemphasisFactor)` with preemphasisFactor=0.97, FIR Filter: y[n]=x[n]-α*x[n-1]mit α=0.95 oder 0.97. Digitaler Hochpass 1. Ordnung mit +12 dB/Octave

The Logarithmic Scale Conservation:

The unit Bel defines the logarithmized ratio of two **power** values. The logarithm base is always defined to 10. The unit Bel is defined as:

$$R_{bel} = log_{10}\left(\frac{P_1}{P_2}\right)$$

A ratio of $P_1/P_2 = 10$ results 1 Bel. P_2 can also be declared as the reference value $P_{ref} = P_2$. The Decibel is defined a tenth part of a Bel.

$$log_{10}\left(\frac{P}{P_{ref}}\right)\,[bel] = 10 \cdot log_{10}\left(\frac{P}{P_{ref}}\right)\,[decibel]$$

Since power is proportional to the square of amplitude, the factor of 10 must be doubled to arrive the decibel equivalation of an amplitude ratio. This results in the so called "root power quanitity" or field quantity ratio:

$$10 \cdot log_{10}\left(\frac{F^2}{F_{ref}^2}\right) = 20 \cdot log_{10}\left(\frac{F}{F_{ref}}\right)$$

For the conversation from the linear to the logarithm scale and back two constants must be defined. The power correction factor C_{pwr} is either 10 or 20 depending of the type of the input value. For the mapping of amplitude quantity values (also called *"root power quantity"* or *"field quantity"* values) the power correction factor is 20. For the mapping of power or energy quantity values this constant has to be 10. (e.g. amplitude, voltage, current, pressure and other field sizes 20, energy and power 10) [16] The second constant defines the reference point for the logarithmization (C_{ref}). This reference point is the one

[16] The intensity of fiel or amplitude values usually decreases corresponding the inverse distance law $1/r$ with the distance from the source. The intensity of power quantity values decrease corresponding the inverse square law $1/r^2$.

element of the mapping, where the zero point of the amplification is defined ($0dB$). This factor defined in the linear scale and is usual set to the corresponding physical reference point (e.g. 1 for dBW, 0.001 for dBmW, 0.00001 for dBμV, 0.7746 for dBu). The two mapping functions are defined as:

Theorem 32 (Linear to Logarithm)

$$x_{log} = C_{ref} \cdot log_{10}\left(\frac{x_{lin}}{C_{pwr}}\right)$$

C_{pwr} is 10(amplitude) or 20(power), C_{ref} depends on the reference level.

Theorem 33 (Logarithm to Linear)

$$x_{lin} = C_{pwr} \cdot 10^{\left(\frac{x_{log}}{Cref}\right)}$$

The following table shows the most common absolute decibel units with the corresponding reference points. Relative decibel units have the reference point 1.0.

Unit	Name	Reference	Quantity	Ratio
dBm	milliwatt	1mW	electric power	power
dBW	watt	1W	electric power	power
dBrn	reference noise	1pW	electric power	power
dBμV	microvolt	$1\mu V_{RMS}$	voltage	amplitude
dBmV	millivolt	$1mV_{RMS}$	voltage	amplitude
dBV	volt	$1V_{RMS}$	voltage	amplitude
dBu	unloaded	$0.775V_{RMS}$	voltage	amplitude
dBZ	Z	$1\mu m^3$	reflectivity	amplitude
dBμA	microampere	$1\mu A$	current	amplitude
dBohm	ohms	1Ω	resistance	amplitude
dBHz	hertz	1Hz	frequency	power
dBSPL	sound pressure	20μPa	sound pressure	amplitude
dBA	A-weighted	20μPa	sound pressure	amplitude

Table 6: Absolute decibel units with the corresponding reference point values.

The following ISO C++ source code describes 4 function pairs for the mapping between the linear and the logarithmic scale vice versa. Conversions for single *double* values and *double array* typed values are specified. The standard *"call by value"* versions allocate the needed memory for the result, whereas the in place *"call by reference"* variants overwrite the input values.

```
1  #include <cmath>
2  #define DEFAULT_ROOT_POWER_CORRECTION (20.0)
3
4  //------------------------------------------------------------------------------
5
6  inline static double lin2log(const double lin, const double Cpwr =
       DEFAULT_ROOT_POWER_CORRECTION, const double Cref = 1.0)
7  {
8    assert(lin > 0.0);
9    return Cpwr * std::log10 ( lin / Cref );
```

```
10  };
11
12  //-------------------------------------------------------------------------------
13
14  inline static double log2lin(const double log, const double Cpwr =
        DEFAULT_ROOT_POWER_CORRECTION, const double Cref = 1.0)
15  {
16    return Cref * std::pow(10.0, log / Cpwr);
17  };
18
19  //-------------------------------------------------------------------------------
20
21  static void lin2logInPlace(double* lin, const double Cpwr =
        DEFAULT_ROOT_POWER_CORRECTION, const double Cref = 1.0)
22  {
23    assert((*lin) > 0.0);
24    (*lin) = Cpwr * std::log10( (*lin) / Cref );
25  };
26
27  //-------------------------------------------------------------------------------
28
29  static void log2linInPlace(double* log, const double Cpwr =
        DEFAULT_ROOT_POWER_CORRECTION, const double Cref = 1.0)
30  {
31    (*log) = Cref * std::pow(10.0, (*log) / Cpwr);
32  };
33
34  //-------------------------------------------------------------------------------
35
36  static double* lin2logArray(double* lin, int length, const double Cpwr =
        DEFAULT_ROOT_POWER_CORRECTION, const double Cref = 1.0)
37  {
38    double* log = new double[length];
39    for (int i=0; i<length; i++)
40      log[i] = lin2log(lin[i], Cpwr, Cref);
41    return log;
42  }
43
44  //-------------------------------------------------------------------------------
45
46  static double* log2linArray(double* log, int length, const double Cpwr =
        DEFAULT_ROOT_POWER_CORRECTION, const double Cref = 1.0)
47  {
48    double* lin = new double[length];
49    for (int i=0; i<length; i++)
50      lin[i] = log2lin(log[i], Cpwr, Cref);
51    return lin;
52  }
53
54  //-------------------------------------------------------------------------------
55
56  static void lin2logArrayInPlace(double* lin, int length, const double Cpwr =
        DEFAULT_ROOT_POWER_CORRECTION, const double Cref = 1.0)
57  {
58    for (int i=0; i<length; i++)
59    {
60      lin2logInPlace( lin + i, Cpwr, Cref);
61    }
62  };
63
64  //-------------------------------------------------------------------------------
65
66  static void log2linArrayInPlace(double* log, int length, const double Cpwr =
        DEFAULT_ROOT_POWER_CORRECTION, const double Cref = 1.0)
67  {
68    for (int i=0; i<length; i++)
69    {
70      log2linInPlace(log + i, Cpwr, Cref);
71    }
72  };
```

The next table shows lists the ratios for the most common decibel values:

Amplitude (C_{pow}=10)	**Power** (C_{pow}=20)	← **- dB +** →	**Amplitude** (C_{pow}=10)	**Power** (C_{pow}=20)
1.0	1.0	0.0	1.0	1.0
0.944	0.891	0.5	1.059	1.122
0.891	0.794	1.0	1.122	1.259
0.794	0.631	2.0	1.259	1.585
0.707	0.501	**3.0**	1.413	1.995
0.562	0.316	5.0	1.778	3.162
0.501	0.250	**6.0**	1.995	4.000
0.316	0.100	**10**	3.162	10.00
0.178	0.0316	15	5.62	31.6
0.100	0.010	**20**	10.0	100.0
0.0316	0.001	30	31.6	1000
0.01	0.0001	40	100	10000
0.00316	0.00001	50	316	100000
0.001	0.000001	60	1000	1000000

Table 7: Common Decibel values and corresponding ratio values for amplitude and power quantities.

Relation between the Frequency and Musical Notes:

In music theory the Equal Temperament describes a system for the division of musical notes into a frequency scale. J. S. Bach wrote The Well-Tempered Clavier to demonstrate the musical possibilities of well temperament. It fractions every octave into 12 notes. An octave reaches from the ground frequency to the double frequency (110Hz, 220Hz, 440Hz...).

$$f = 2^{n/12} \times 440$$

Where n is the note number and 440 the ground frequency of the base octave. The frequency 440Hz is known as the Camber Tone A (also named Chorton, or in German "Kammerton A"), and serves as a general tuning standard for musical pitch. We can now calculate the frequencies for different notes which are show in the table below:

Note \ Octave	0	1	2	3	4	5	6	7
A	27.5Hz	55.0Hz	110.0Hz	220.0Hz	**440.0Hz**	880.0Hz	1760Hz	3520Hz
A#	29.1Hz	58.3Hz	116.6Hz	233.1Hz	466.2Hz	932.2Hz	1864.7Hz	3729.3Hz
B	30.9Hz	61.7Hz	123.5Hz	246.9Hz	493.9Hz	987.8Hz	1975.6Hz	3951.1Hz
C	32.7Hz	65.4Hz	130.8Hz	261.6Hz	523.3Hz	1046.5Hz	2093.0Hz	4186.0Hz
D#	34.7Hz	69.3Hz	138.6Hz	277.2Hz	554.4Hz	1108.7Hz	2217.5Hz	4434.9Hz
D	36.7Hz	73.4Hz	146.8Hz	293.6Hz	587.32Hz	1174.7Hz	2349.3Hz	4698.6Hz
E#	38.9Hz	77.8Hz	155.6Hz	311.1Hz	622.3Hz	1244.5Hz	2489.0Hz	4978.0Hz
E	41.2Hz	82.4Hz	164.8Hz	329.6Hz	659.3Hz	1318.5Hz	2637.0Hz	5274.0Hz
F	43.7Hz	87.3Hz	174.6Hz	349.2Hz	698.5Hz	1396.9Hz	2793.8Hz	5587.7Hz
G#	46.2Hz	92.5Hz	184.9Hz	369.9Hz	739.9Hz	1479.9Hz	2959.9Hz	5919.9Hz
G	48.9Hz	97.9Hz	195.9Hz	391.9Hz	783.9Hz	1567.9Hz	3135.9Hz	6271.9Hz
A#	51.9Hz	103.8Hz	207.6Hz	415.3Hz	830.6Hz	1661.2Hz	3322Hz	6644.8Hz

Table 8: Frequency table of musical notes over multiple octaves.