Digital Signal Processing

1 Applications

- Input: in most cases sensory data from a physical environment: seismic vibrations, visual images, sound waves, etc.

Example Applications:

- Telecommunications
  - Multiplexing. Example: telephone standard *T-Carrier System* for simultaneous transmission of 24 voice signals. Each voice signal is represented as 64 KBit/sec, all 24 channels being contained in 1.544 MBit/sec (8000 samples/sec, 8bit DAC).
  - Compression of digitized voice signals.
  - Echo control. Telephone, anti noise.
- Audio Processing
  - Music. Typically a musical piece is recorded on multiple channels or tracks to allow greater flexibility in creating the final product. The process of combining the individual tracks into a final product is called *mix down*. DSP provides important functions during mix down including filtering, signal addition and subtraction. Examples: artificial reverberation, artificial echoes, etc.
  - Speech generation: digital recording vs. vocal tract simulation.

- Echo Location
  - Radar (RAdio Detection And Ranging) and sonar: impulse generation, impulse compression, filtering.
  - Reflection seismology
- Image Processing
  - Computed Tomography (CT)
  - Magnetic Resonance Imaging (MRI).
- Commercial imaging, especially image compression for digital TV, video telephones, etc.

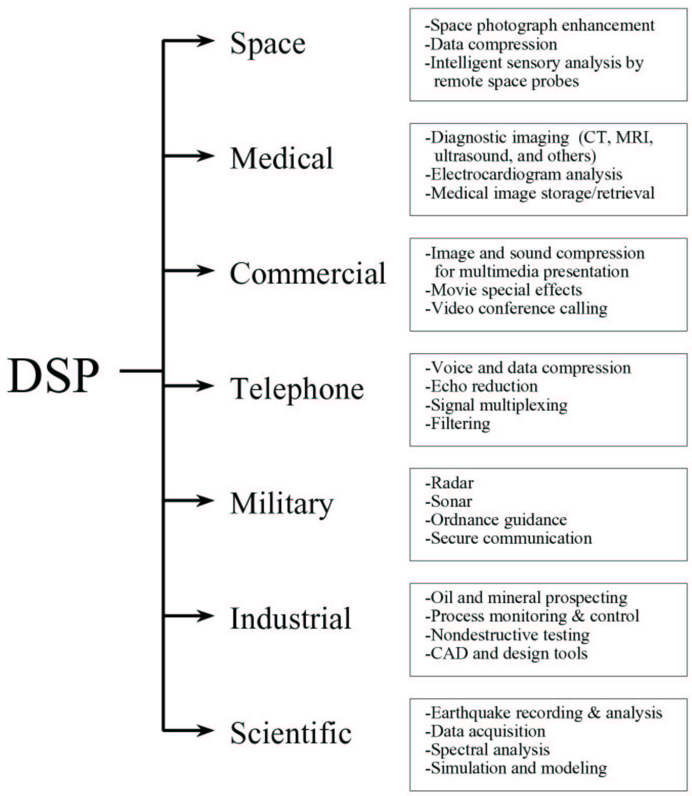

Figure 1: Applications of Digital Signal Processing.

2 AD-Conversion and DA-Conversion

- Signal: function of one or more variables with information about the behavior or the nature of physical processes.
- x-axis: *independent* variable; y-axis: *dependent* variable.
- A signal using time as independent variable is said to be in the *time domain.*
- A signal using frequency as independent variable is said to be in the *frequency domain.*

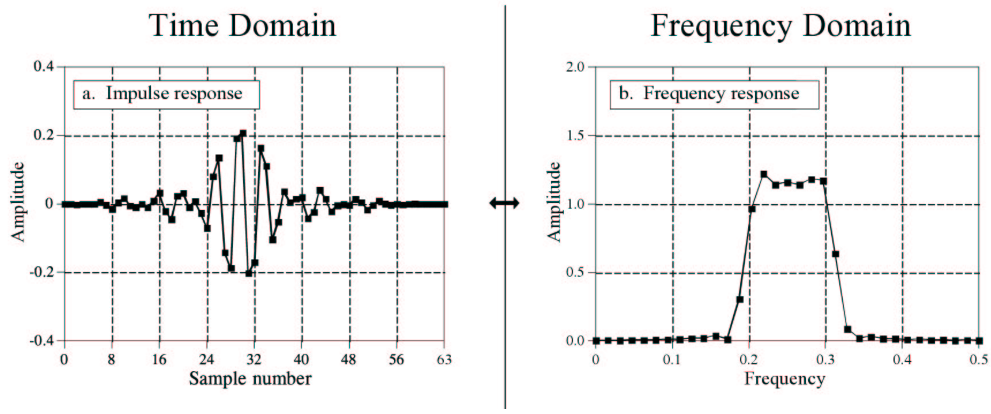

Figure 2: Time and Frequency Domain.

- Digitization of continuous signals: AD-Conversion (Analog-to-Digital Conversion, ADC).
- Inverse: DA-Conversion (Digital-to-Analog Conversion, DAC).
- Digitization:
  1. Sampling (Sample-and-Hold, S/H): Input signal is measured (sampled) with a given rate (sampling rate). $\Rightarrow$ Sampling converts the independent variable from continuous to discrete.

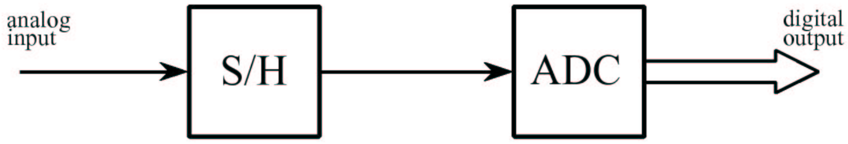

Figure 3: Digitization.

2. Quantization: the sampled value is converted to the nearest integer number. $\Rightarrow$ Quantization converts the dependent variable from continuous to discrete.

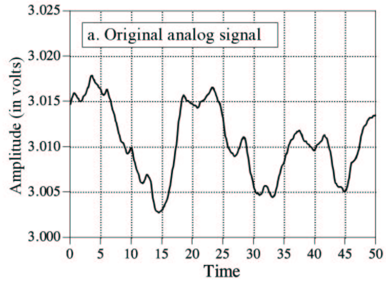

Figure 4: Input Signal.

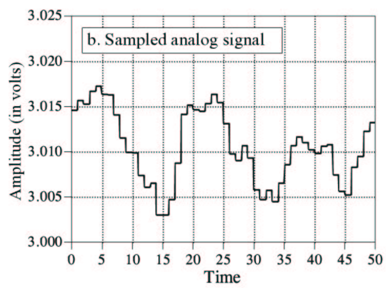

Figure 5: Sampled Input Signal.

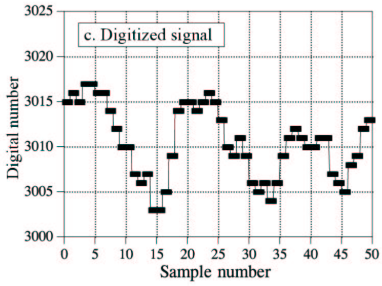

Figure 6: Digitized Signal.

3 Statistics

- Let N values $x_0, \ldots, x_{N-1}$ be given. The *mean* is defined as $\mu = \frac{1}{N}\sum_{i=0}^{N-1} x_i$. In electronics, the mean is commonly called the *direct current* (DC) value. Likewise, AC (alternating current) refers to how the signal fluctuates around the mean value.

- The *variance* σ^2 is defined as $\sigma^2 = \frac{1}{N-1}\sum_{i=0}^{N-1}(x_i - \mu)^2$; the square root of the variance is called *standard deviation*. The standard deviation is a measure of how far the signal fluctuates from the mean.

- The above definition of σ requires that all of the samples are involved in each new calculation, ie if new samples are acquired and added to the signal.
  Thus: $\sigma^2 = \frac{1}{N-1}(\sum_{i=0}^{N-1} x_i^2 - (\frac{1}{N}(\sum_{i=0}^{N-1} x_i)^2)$.

- The ratio $SNR = \frac{\mu}{\sigma}$ is called *signal-to-noise ratio*.

- The histogram H displays for each possible value the number of samples having this value. Let M be the number of possible values, then $N = \sum_{i=0}^{M-1} H_i$.

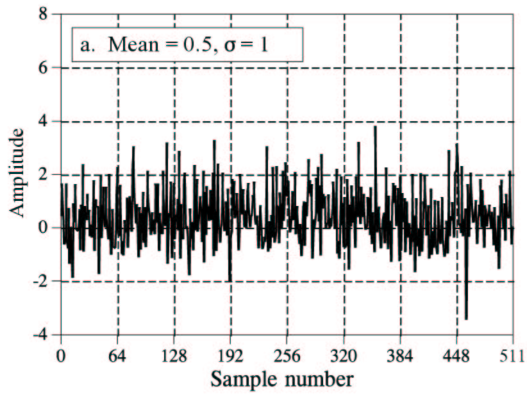

Figure 7: Mean and Standard Deviation.

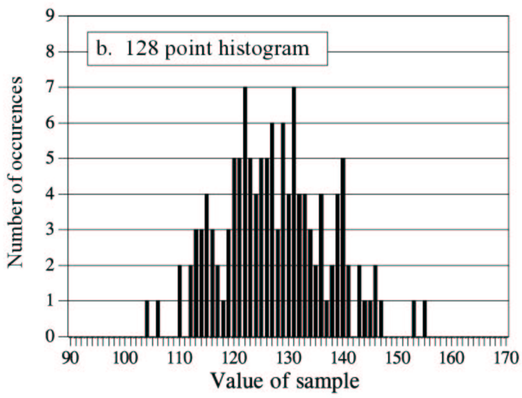

Figure 8: Histogram.

4 AD-Conversion and DA-Conversion (c'ed)

Quantization Error

- Storing a variable with a given number representation $\Rightarrow$ quantization error.
- n-bit Unsigned Integers $\Rightarrow$ Deviation between $+-\frac{1}{2}2^{-n}$.
- Value of digitized signal: continuous input plus quantization error.
- The bit width determines the *precision*.

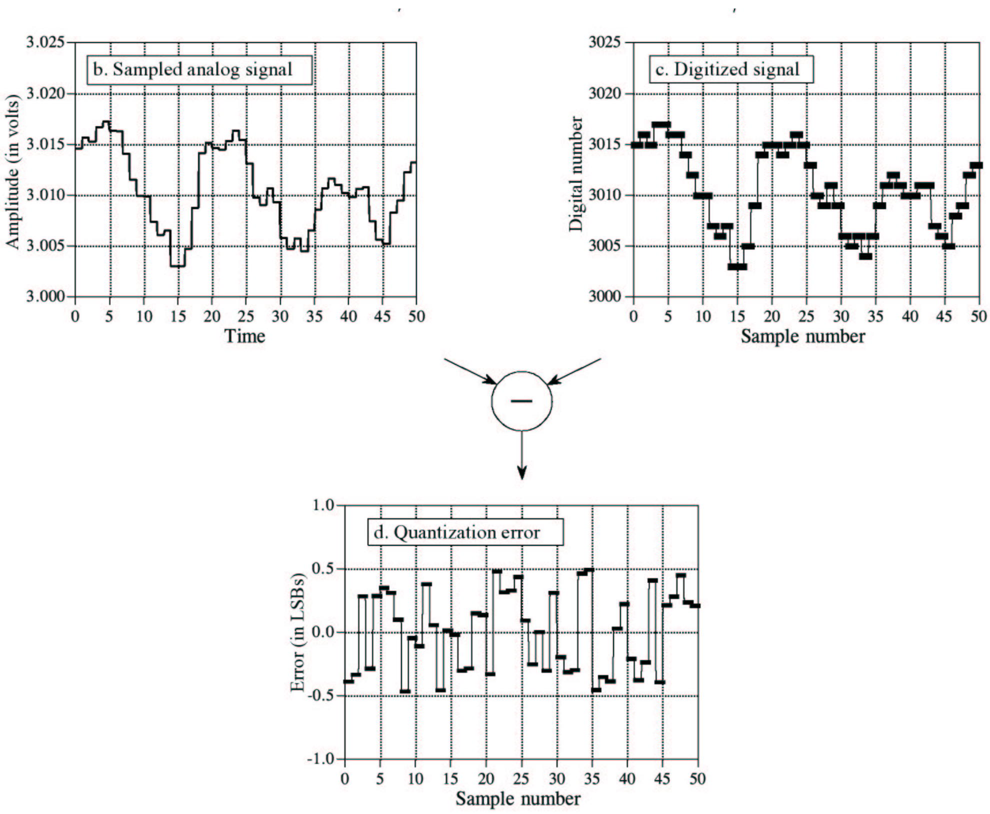

Figure 9: Quantization Error.

- Effects: Addition of *random noise* to the signal.
- Random noise: uniformly distributed between $+-\frac{1}{2}LSB$, with a mean of 0 and a standard deviation of $\frac{1}{\sqrt{12}}LSB$.

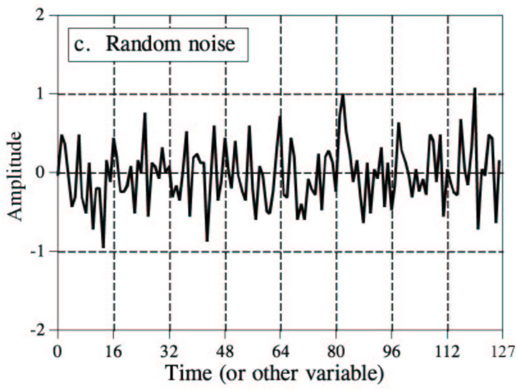

Figure 10: Random Noise.

Fourier Analysis

- Fourier Analysis: Description of a physical process depending on the variation of a quantity h in time ($h(t)$) by a sum of trigonometric functions characterized by their amplitudes $H(f)$ depending on the frequency.
- $h(t)$ and $H(f)$ are two representations of the same function.
- The fourier transform can be used to switch between these two representations.
- If the frequency is depicted on the x-axis and the corresponding amplitudes on the y-axis, the *frequency spectrum* results.

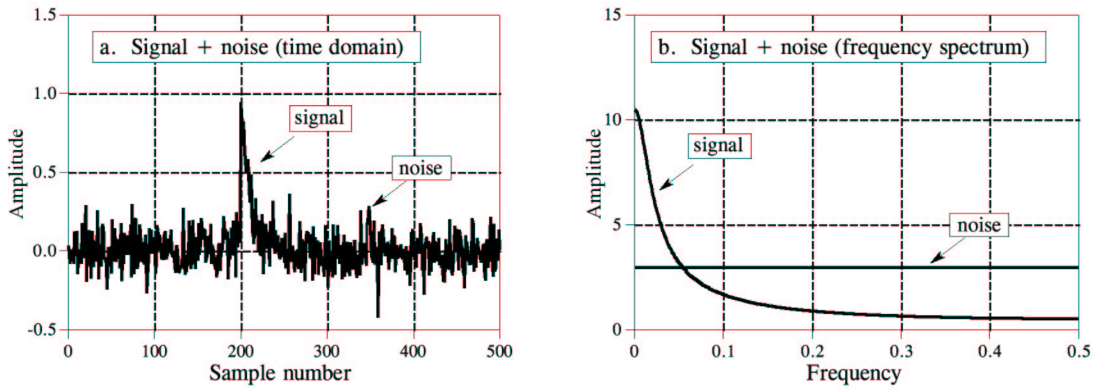

Figure 11: Frequency Spectrum.

Proper Sampling

- Sampling is called *proper* if the analog signal can be exactly reconstructed from the samples.

- *Nyquist Sampling Theorem*: If a function $h(t)$ is measured at time intervals Δt, there is a special frequency f_c, the so-called *Nyquist frequency*
$$f_c = \frac{1}{2\Delta t}$$
which represents the upper bound of the frequencies that can be represented by the fourier transform.

- A continuous signal can be properly sampled only if it does not contain frequency components above one-half of the sampling rate, ie above the Nyquist frequency.

- If a signal is transformed that contains frequencies above the Nyquist frequency, those are mapped to frequencies below the Nyquist frequency. This leads to *aliasing* corrupting the original signal.

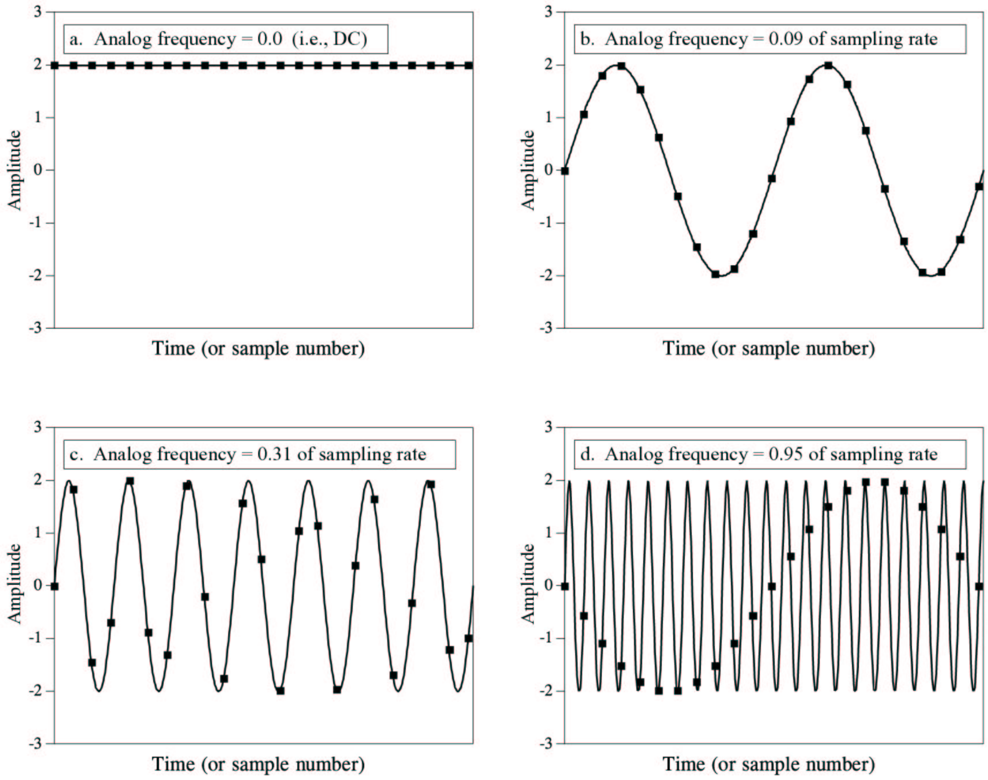

Figure 12: Sampling und Aliasing.

- The amount of information carried in a digital signal is limited in two ways:
  - The number of bits per sample limits the resolution of the dependent variable. Small changes in the signal's amplitude may be lost in the quantization noise.
  - The sampling rate limits the resolution of the independent variable, ie closely spaced events in the analog signal may be lost between the samples.
- DAC: similar to ADC. Usually, DACs operate by holding the last value until another sample is received (zeroth-order hold). Subsequently: filter reconstructing the continuous signal from the zeroth order hold.

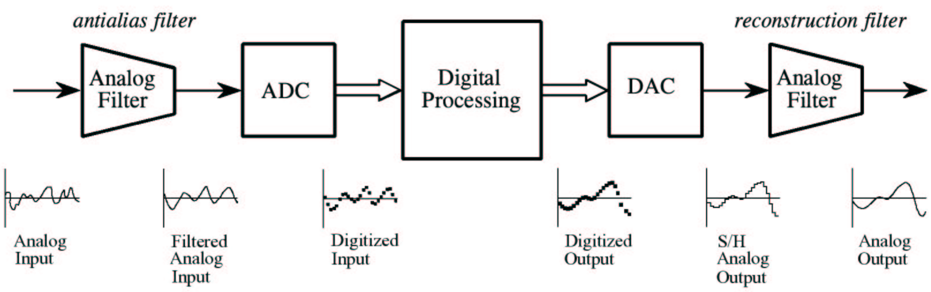

Figure 13: Simple DSP System.

Simple DSP system:

- Before encountering the ADC the input signal is processed with an electronic low-pass filter to remove all frequencies above the Nyquist frequency. This is done to prevent aliasing during sampling (antialias filter).

- On the other end, the digitized signal is passed through a DAC and another low-pass filter set to the Nyquist frequency (reconstruction filter).

- Trend: Replacement of analog circuitry by digital algorithms.

- Reason: cost and performance. E.g. due to physical limitations of analog circuits digital algorithms can yield significantly higher precision.

5 Linear Systems

- Divide-and-Conquer Strategy called *superposition*:
  - Signal being processed is broken into simple components each of which is processed individually.
  - Results are reunited.
  - Presupposition: system is linear.

- Input-Output system S: Process converting an input signal into an output signal.

- Input and output signals continuous: continous system.

- Input and output signals discrete: discrete system.

- A system is called *linear* if it has the following properties:
  - Homogeneity: An amplitude change in the input results in an identical amplitude change in the output. That is, if $S(x) = y, x'[n] = kx[n]\ \forall n$, then $S(x') = y'$, where $y'[m] = ky[m]\ \forall m$.

- Additivity: A system is called additive if added signals pass through it without interacting. Formally: if $S(x_1) = y_1$ and $S(x_2) = y_2$,then $S(x_1 + x_2) = y_1 + y_2$.

- Not a strict requirement for linearity, but a mandatory property for most DSP techniques is the *shift invariance*: A system is said to be shift invariant if a shift in the input signal causes an identical shift in the output signal. If $S(x) = y$ and $x'[n+s] = x[n]\ \forall n$, then $S(x') = y'$ where $y'[m + s] = y[m]\ \forall m$. Shift invariance means that the characteristics of the system do not change with time.

- Synthesis: Combining signals through scaling and addition.

- Decomposition: A single signal is broken into two or more additive components.

Fundamental Concept of DSP

- Any signal x can be decomposed into a group of additive components $x_1, x_2, \ldots, x_k$. Passing these components through a linear system produces the signals y_1, y_2, ..., y_k. The synthesis (addition) of these output signals forms y, the same signal produced when x is passed directly through the system. Input and output signals are viewed as a superposition (sum) of simpler waveforms.

Decomposition Techniques

- Impulse decomposition. Impulse: signal composed of all zeros except a single nonzero point. By knowing how a system responds to an impulse the system's output can be calculated for any given input. This technique is called *convolution*.

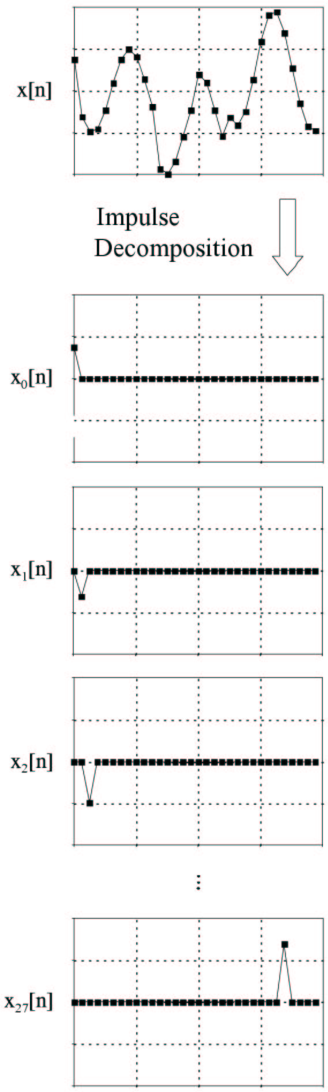

Figure 14: Impulse Decomposition.

- Interlaced decomposition. The signal is broken into two component signals, the even sample signal and the odd sample signal. Interlaced decomposition is the basis for the Fast Fourier Transformation.

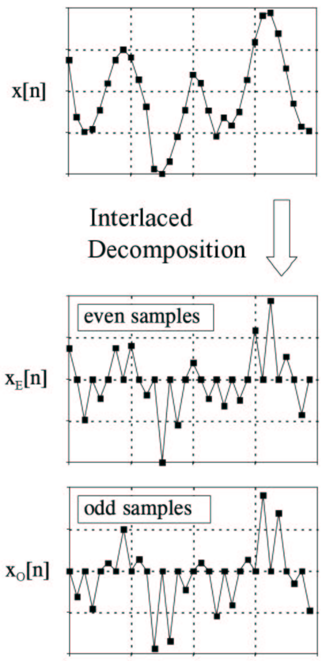

Figure 15: Interlaced Decomposition.

- Fourier decomposition.

6 Convolution

- The *unit impulse* (*delta function*) δ is a normalized impulse: Sample number zero has a value of one, while all other samples have a value of zero. Any impulse can be represented as a shifted and scaled delta function.

- *Impulse response* h: The signal that exits a system when a unit impulse is the input.

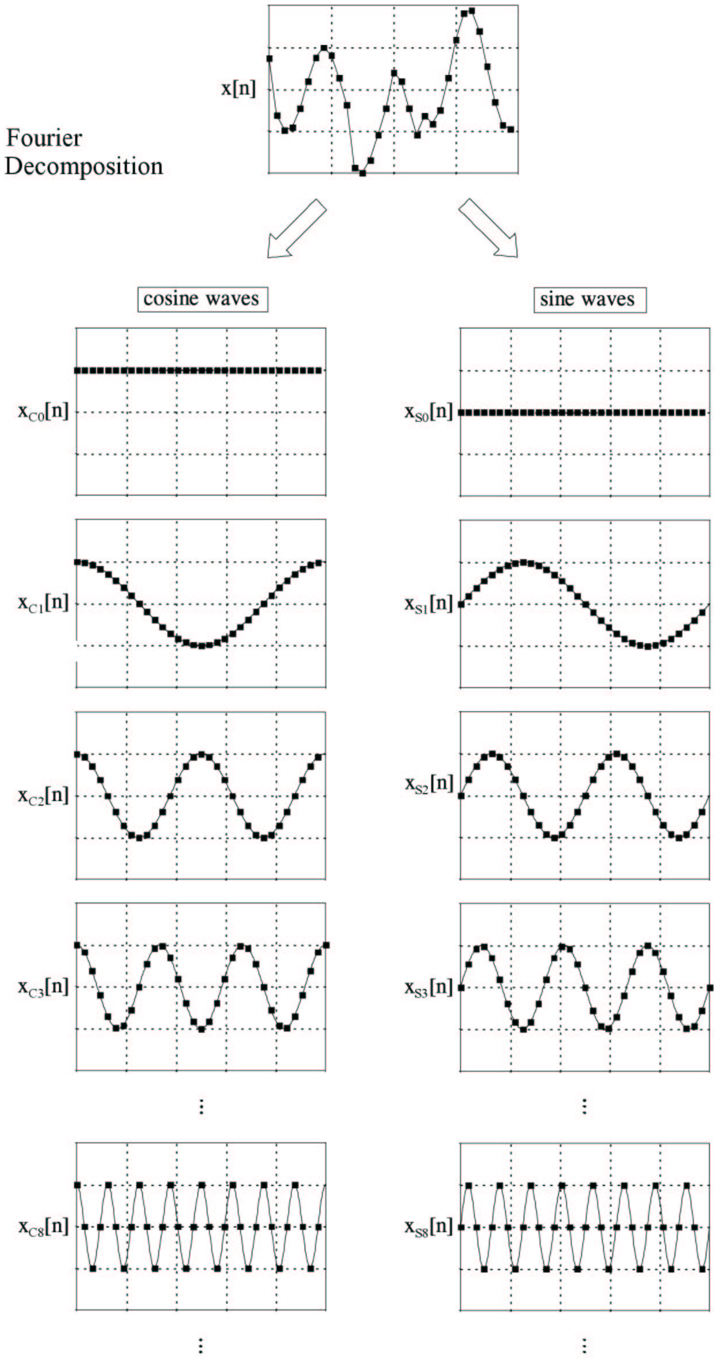

Figure 16: Fourier Decomposition.

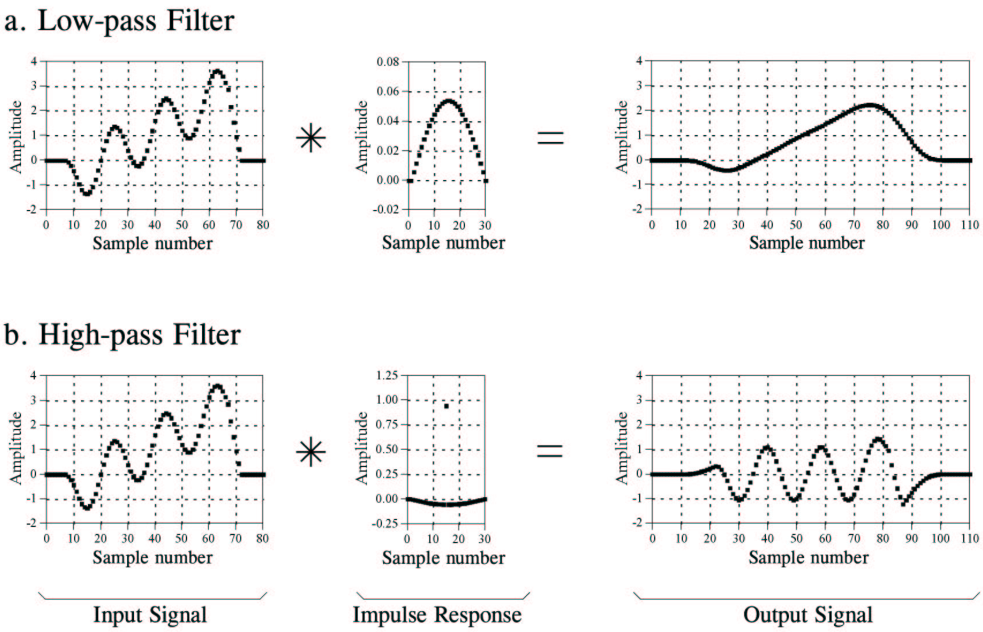

Figure 17: Convolution.

Definition

Let x be an N-point input signal $(x[0], \ldots, x[N-1])$ and h an M-point signal $(h[0], \ldots, h[M-1])$. Then the convolution y of h and x is the following $N+M-1$-point signal $(y[0], \ldots, y[N+M-1]$:

$$y[i] = \sum_{j=0}^{M-1} h[j]x[i-j]$$

Short notation: $y = h * x$.

- Padding / End effect problems.

Properties of Convolution

- Commutativity: $a * b = b * a$
- Associativity: $(a * b) * c = a * (b * c)$

- Distributivity: $a * b + a * c = a * (b + c)$
- $x * \delta = y$ where $y[j] = x[j] \; \forall \; j = 0, \ldots, N-1$
- $x * k\delta = y$ where $y[j] = kx[j] \; \forall \; j = 0, \ldots, N-1$
- Be $\delta'[j+s] = \delta[j]$, $\delta'[z] = 0$ for $z = 0, \ldots, s-1$, $s \in \mathbb{N}$. Then $x * \delta' = y$ where $y[j+s] = x[j] \; \forall \; j = 0, \ldots, N-1$.

7 Digital Filters

- Digital filters are created by *designing* an appropriate impulse response. Examples: radar detection, echo suppression, etc.
- Low-pass and high-pass filters.
- Radar systems: Given a signal of some known shape, what is the best way to determine where (or if) the signal occurs in *another* signal?

Definition

Let t be the target signal that has to be recognized, a the input signal; then the *cross correlation* of a and t is defined as

$$r[i] = \frac{1}{N} \sum_{j=0}^{M-1} a[j]t[i+j]$$

- Value of a sample in the cross correlation is a measure of how much the received signal resembles the target signal at that location.

- The value of the cross correlation is maximized when the target signal is aligned with the same features in the received signal.
- Correlation is the optimal technique for detecting a known waveform in random noise. That is, the peak is higher above the noise using correlation than can be produced by any other linear system. Using correlation to detect a known waveform is often called *matched filtering*.
- The cross correlation can be computed by convolution. This requires preflipping one of the two signals being correlated.

Types of Digital Filters

- *Finite Impulse Response Filter* (FIR): Convolution of input signal with the impulse response of the digital filter (scaling + addition). In this context, the impulse response is also called *filter kernel*.
- Recursive filter / *Infinite Impulse Response Filter* (IIR): Recursive filters are an extension of FIR filters, using previously calculated values from the output, besides points from the input. Instead of using a filter kernel, recursive filters are defined by a set of *recursion coefficients*. Typically the impulse responses of IIR filters are composed of sinusoids that exponentially decay in amplitude. Since this impulse response is infinitely long, recursive filters are often called *infinite impulse response* filters. In comparison, filters carried out by convolution have a finite impulse response.

Definition of Recursive Filters

$$y[n] = \sum_{i=0}^{L} a_i x[n-i] + \sum_{i=1}^{L} b_i y[n-i]$$

- a, b: Coefficients
- x: Input signal
- y: Output signal
- L: Number of poles

Finite vs. Infinite Impulse Response Filters

- Advantage of recursive filters: They are an efficient way of achieving a long impulse response without having to perform a long convolutions. Fast computation.
- Disadvantage of recursive filters: Less performance and flexibility (rounding errors e.g. due to number representation of coefficients).

8 Implementation

- Simple algorithm is straightforward. Problem: large number of additions and multiplications. Consequence: virtually all Digital Signal Processors provide Multiply-Accumulate units (MAC units) performing a multiplication and an addition of the result to an accumulator in one clock cycle.

- FFT convolution produces exactly the same results as the standard convolution but the execution time can be reduced drastically.
- Convolution is the method of choice if the DFT has less than 32 points; otherwise the FFT is used.

9 Discrete Fourier Transform

Given a piecewise continuous function $f : \mathbb{R} \to \mathbb{R}$ with period 2π, $f(x+2\pi) = f(x)\ \forall x \in \mathbb{R}$. f may have jump discontinuities; for a point of discontinuity x_0 there exist the limits y_0^- and y_0^+ with $y_0^- = \lim_{h\to 0^+} f(x_0 - h)$ and $y_0^+ = \lim_{h\to 0^+} f(x_0 + h$ and they are finite. Decompose $[0, 2\pi]$ in N subintervals with step $h = \frac{2\pi}{N}$ and consider the points $x_j = h_j = \frac{2\pi j}{N}$, $j = 0, 1, \ldots, N$. Define

$$\begin{aligned} a_k^* &= \frac{2}{N}\sum_{j=1}^{N} f(x_j)\cos(kx_j),\ \ k = 0, 1, \ldots \\ b_k^* &= \frac{2}{N}\sum_{j=1}^{N} f(x_j)\sin(kx_j),\ \ k = 1, 2, \ldots \end{aligned}$$

and be $n = \frac{N}{2} \in \mathbb{N}$. Then

$$g_n^*(x) = \frac{1}{2}a_0^* + \sum_{k=1}^{n-1}(a_k^*\cos(kx) + b_k^*\sin(kx)) + \frac{1}{2}a_n^*\cos(nx)$$

is the unique Fourier polynomial for the data points x_j with values $f(x_j)$, $j = 1, \ldots, N$ (data points correspond to sampling times and their values to the value of the input signal at the sampling times).

Thus:

$$\begin{aligned} a_k^* &= \frac{2}{N}\sum_{j=1}^{N} x[j]\cos(\frac{2\pi kj}{N}),\ k = 0, 1, \ldots, n \\ b_k^* &= \frac{2}{N}\sum_{j=1}^{N} x[j]\sin(\frac{2\pi kj}{N}),\ k = 0, 1, \ldots, n \end{aligned}$$

a_k^* and b_k^* are the $n + 1 = \frac{N}{2} + 1$ coefficients that are required to calculate g_n^*. Define further

$$\begin{aligned} a_k &:= \frac{N}{2} a_k^* \\ b_k &:= \frac{N}{2} b_k^* \end{aligned}$$

DFT basis functions:

- $c_k[i] = \cos(\frac{2\pi ki}{N}), i \in \{0, \ldots, N-1\}$
- $s_k[i] = \sin(\frac{2\pi ki}{N}), i \in \{0, \ldots, N-1\}$

k: Frequency of the basis functions, ie number of complete periods over the N points of the signal ($f = \frac{k}{N}$).

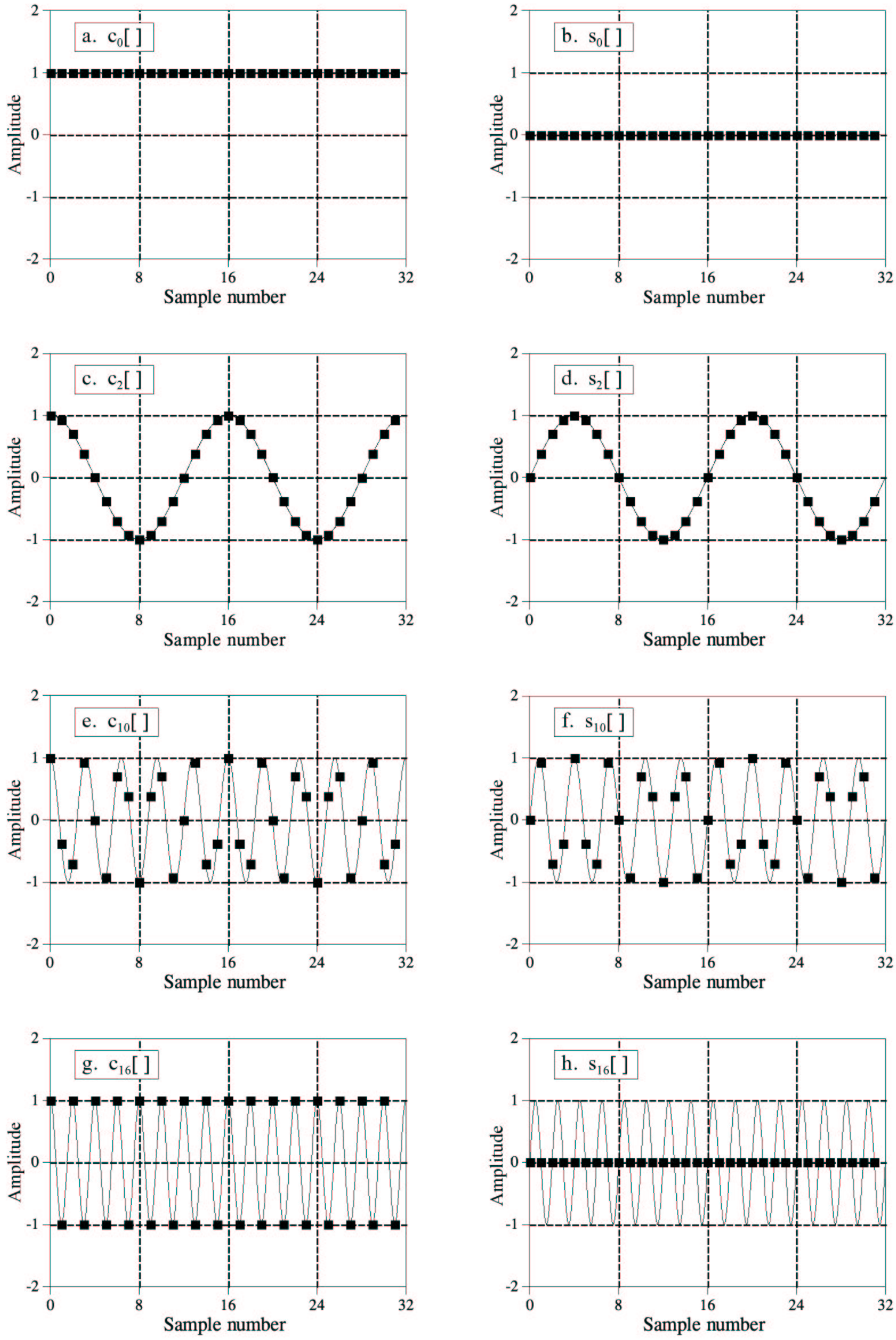

Figure 18: DFT Basis Functions.

Discrete Fourier Transform

- transforms an N-point input signal into two $N/2+1$-point output signals.
- Input signal: signal to be decomposed.
- Output signals: define the amplitudes of the DFT basis functions.
- The input signal is in the time domain.
- The number and frequency of the DFT basis functions is fixed: The output signal is in the frequency domain (depends on the frequency).
- Given the time domain signal, the *forward DFT* calculates the corresponding frequency domain.

  a_k^*: amplitudes of the cosine functions, b_k^*: amplitudes of the sinus functions.
- a_k and b_k correspond to the real part resp. the imaginary part of the frequency domain.

Inverse DFT (Synthesis Equation)

$$x[j] = \frac{1}{2}a_0 + \sum_{k=1}^{n} a_k \cos(\frac{2\pi kj}{N}) + \sum_{k=0}^{n} b_k \sin(\frac{2\pi kj}{N}) + \frac{1}{2}\cos(\pi k),\ k = 0, \ldots, n$$

Computing the DFT (Analysis)

- Solve a set of simultaneous equations where N values from the time domain are given and the N values of the frequency

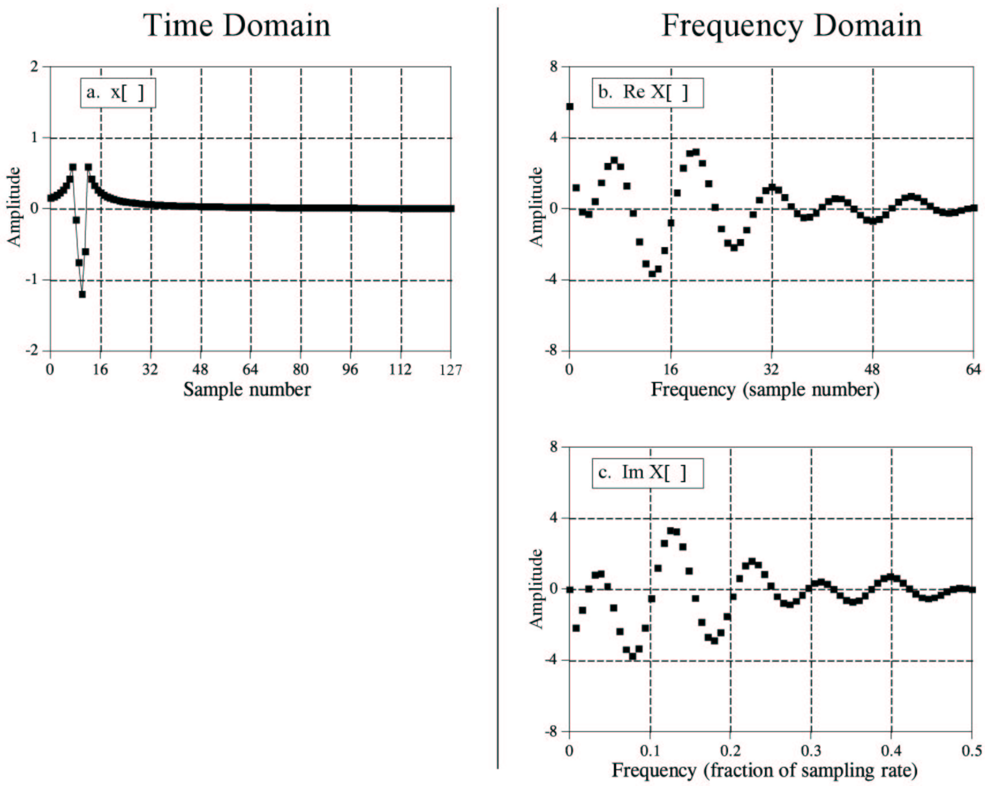

Figure 19: Time and Frequency Domain.

domain have to be computed. This requires N linearly independent equations $\Rightarrow$ inefficient.

- Correlation: detect a known waveform contained in another signal. The a_k and b_k are directly computed by multiplications and additions. This way, the input signal is correlated with each of the basis functions.
- Fast Fourier Transform: Decompose a DFT with N points in N DFTs, containing one point each.

Frequency Response

- Frequency response of a system: Fourier Transform of its impulse response.
- It describes how a system modifies amplitude and phase of cosine waves passing through it.
- Characterizes the system completely.

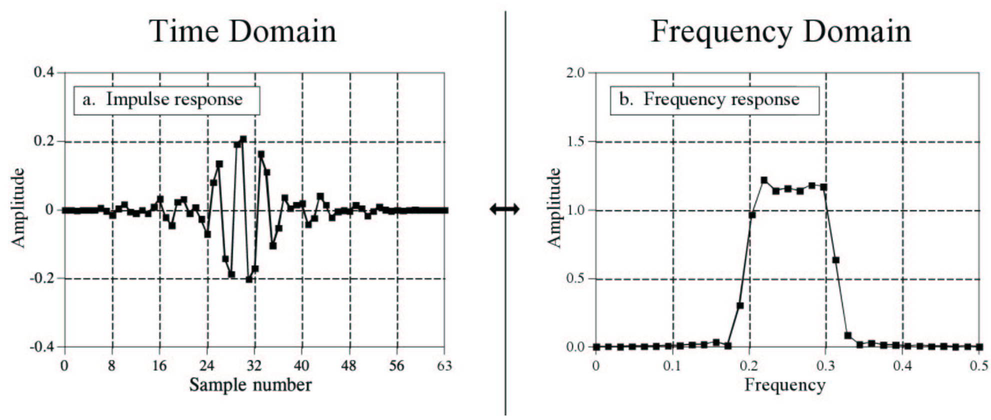

Figure 20: Impulse and Frequency Response.

Time and Frequency Domain

- Time domain: Convolution of the input signal with the impulse response ($x[n] * h[n] = y[n]$).
- Frequency domain: Multiplication of the input signal with the frequency response.
- DFT and IDFT relate the signals in the two domains.

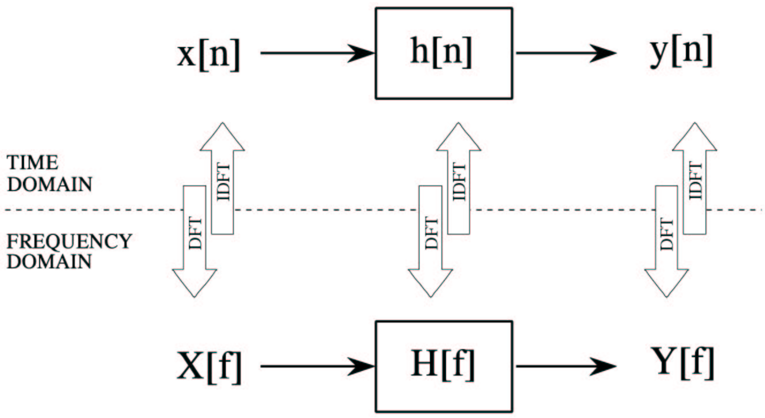

Figure 21: DCT und IDCT.

Alternative Computation

- Given: Input signal and impulse response.
- Transformation in the frequency domain.
- Multiplication
- Retransformation in the time domain.
- No gain in execution speed if standard convolution is performed – only when FFT is used.

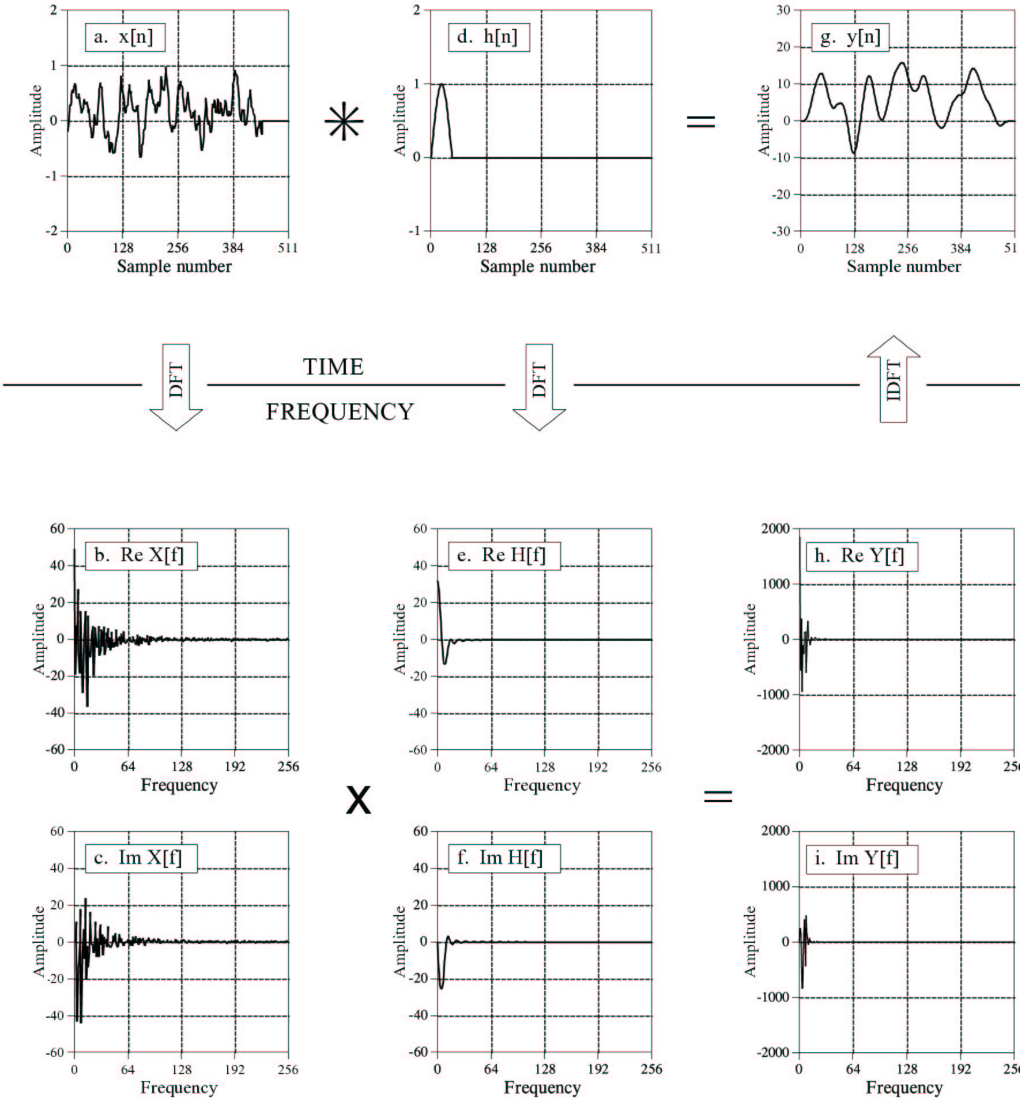

Figure 22: Convolution in the Frequency Domain.

10 FFT – Fast Fourier Transformation

- The FFT is an algorithm for calculating the complex DFT. Thus: DFT data have to be converted into complex format.
- If an N point signal is given, all its points are moved into the real part of the complex DFT's time domain, and all samples of the imaginary part are set to 0.
- Calculation of the complex DFT results in a real and an inaginary signal in the frequency domain, each composed of N points. Samples 0 through $\frac{N}{2}$ of the real part and the imaginary part of this signal correspond to the real DFT's spectrum (ie to the values a_k und b_k).

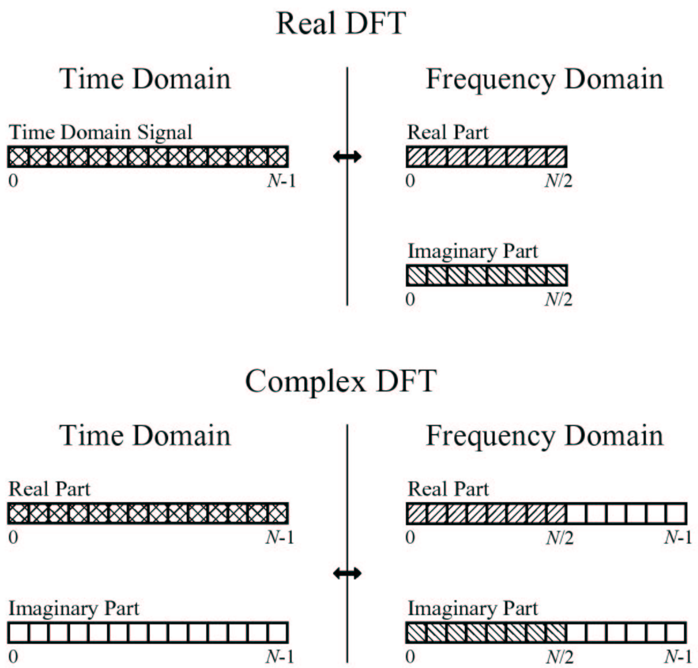

Figure 23: Real and complex DFT.

- The discrete complex Fourier transform is defined as follows:

$$X[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] \left(\cos(\frac{2\pi kn}{N}) - i \sin(\frac{2\pi kn}{N}) \right)$$

FFT

- Decompose the N-point time domain signal into N time domain signals each composed of a single point.
- Calculate the N frequency spectra corresponding to these N time domain signals.
- The frequency spectrum of a 1 point signal is equal to itself.
- Synthesize the N spectra into a single frequency spectrum.

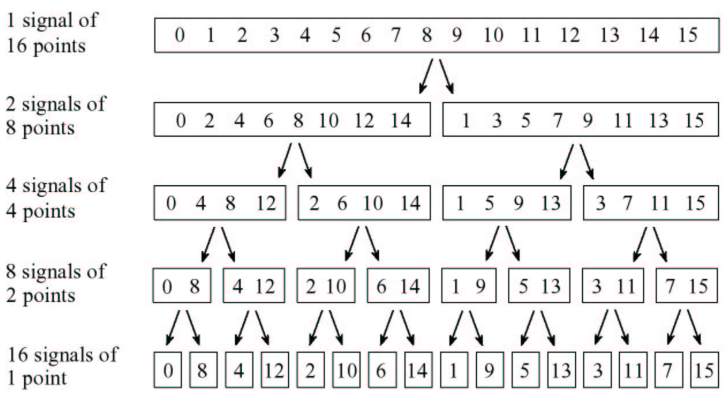

Figure 24: FFT Decomposition.

Interlacing

- Each stage of the FFT decomposition uses an interlace decomposition, separating the even and odd numbered samples.

• ⇒ Sort the samples in reverse bit order.

Sample numbers in normal order		Sample numbers after bit reversal	
Decimal	*Binary*	*Decimal*	*Binary*
0	0000	0	0000
1	0001	8	1000
2	0010	4	0100
3	0011	12	1100
4	0100	2	0010
5	0101	10	1010
6	0110	6	0100
7	0111	14	1110
8	1000	1	0001
9	1001	9	1001
10	1010	5	0101
11	1011	13	1101
12	1100	3	0011
13	1101	11	1011
14	1110	7	0111
15	1111	15	1111

Figure 25: Bit-reverse Sorting.

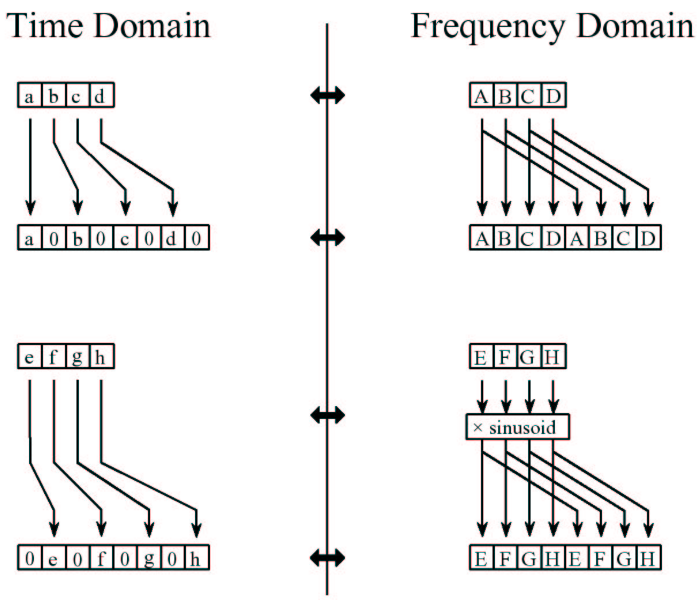

Figure 26: FFT Synthesis. Interlacing.

Synthesis

- Combine the N frequency spectra in the exact reverse order that the time domain decomposition took place.
- Let two frequency spectra be given, each composed of 4 points which are combined into a single frequency spectrum of 8 points. The synthesis must undo the interlaced decomposition done in the time domain.
- When a time domain signal is diluted with zeros, the frequency domain is duplicated.
- If the time domain signal is also shifted by one sample during the dilution, the spectrum will additionally be multiplied by a sinusoid.
- Time domain shift is equivalent to convolving the signal with

a shifted delta function. This multiplies the signal's spectrum with the spectrum of the shifted delta function. The spectrum of a shifted delta function is a sinusoid.

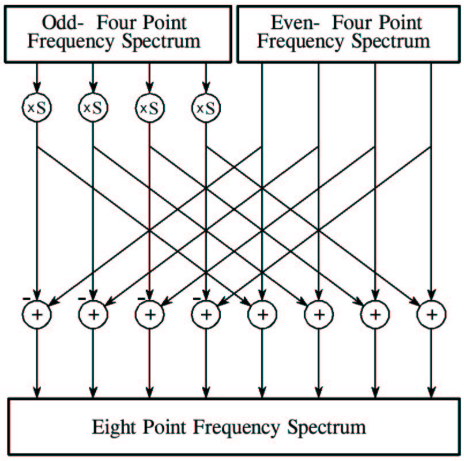

Figure 27: FFT synthesis.

FFT Butterfly

- Input: two complex points.
- Conversion into two other complex points.

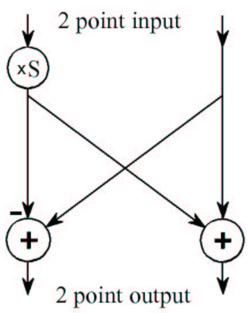

Figure 28: FFT Butterfly.

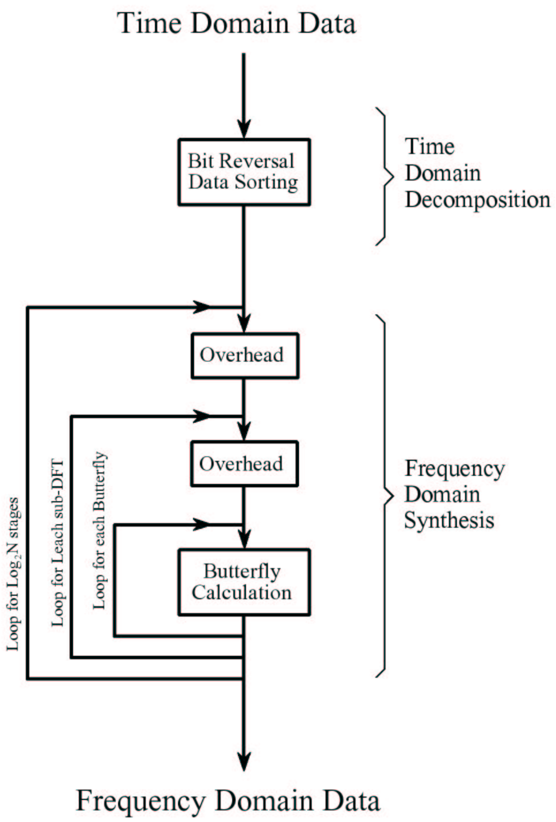

Figure 29: FFT Algorithm.

Bit-Reverse Addressing

- The bit-reverse addressing mode has been developed especially for the FFT and is implemented in most DSP.
- Example Infineon TriCore: Using two successive address registers, a circular buffer is implemented.
- Even numbered register: base address (beginning of buffer)
- Odd numbered register: element size and index.
- Example: Let an array with 8 elements be given:

  $A = 0, 1, 2, 3, 4, 5, 6, 7$, ie $A[i] = i$. The access sequence inside the FFT is $0, 4, 2, 6, 1, 5, 3, 7$.

```
mov.a  a10,#8             ; iteration count
movh.a a3,#16             ; size=16 (bit), index=0
lea    a2, val            ; base address = adr(val) = 0x00080000
_L: ld.h   d0,[a2/a3+r]   ; a3 = 0x00000000, 0x00000008,
                          ; 0x00000004, 0x0000000c,
                          ; 0x00000002, ...
...
loop a10, _L
```

References

- Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing.* 2nd Edition. California Technical Publishing.
- Schwarz. Numerische Mathematik. Teubner, 1993.