Digitale Signalverarbeitung

1 Anwendungen

- Eingabe: meist Sensordaten aus einer physikalischen Umgebung, z.B. Erdbebenwellen, Bilder, Schallwellen, etc.

Anwendungsbeispiele:

- Telekommunikation
  - Multiplexing. Beispiel: Telefonstandard T-Carrier System zur gleichzeitigen Übertragung von 24 Stimmsignalen über eine einzige Leitung. Übertragungsrate: 1.544 MBit/sec; 64KBit/sec für jedes Sprachsignal (8000 Samples pro Sek., 8-bit ADC).
  - Kompression von digitalisierten Sprachsignalen.
  - Echo-Kontrolle. Telefon, Antilärm.
- Audio Verarbeitung
  - Musik. Nachbearbeitung beim Mix-Down. Beispiel: künstliche Halleffekte und Echos, um verschiedene Hörumgebungen zu simulieren.
  - Spracherzeugung: digitale Aufzeichnung vs. Vocal Tract Simulators.
- Echo-Lokalisierung
  - Radar und Sonar (Impulserzeugung, Impulskompression, Filtern).
  - Reflection Seismology

- Bildverarbeitung
  - Computertomographie
  - Kernspintomographie (Magnetic Resonance Imaging, MRI).
- Kommerzielle Bildverarbeitung, insbesondere Bildkompression für digitales Fernsehen, Videotelephone, etc.

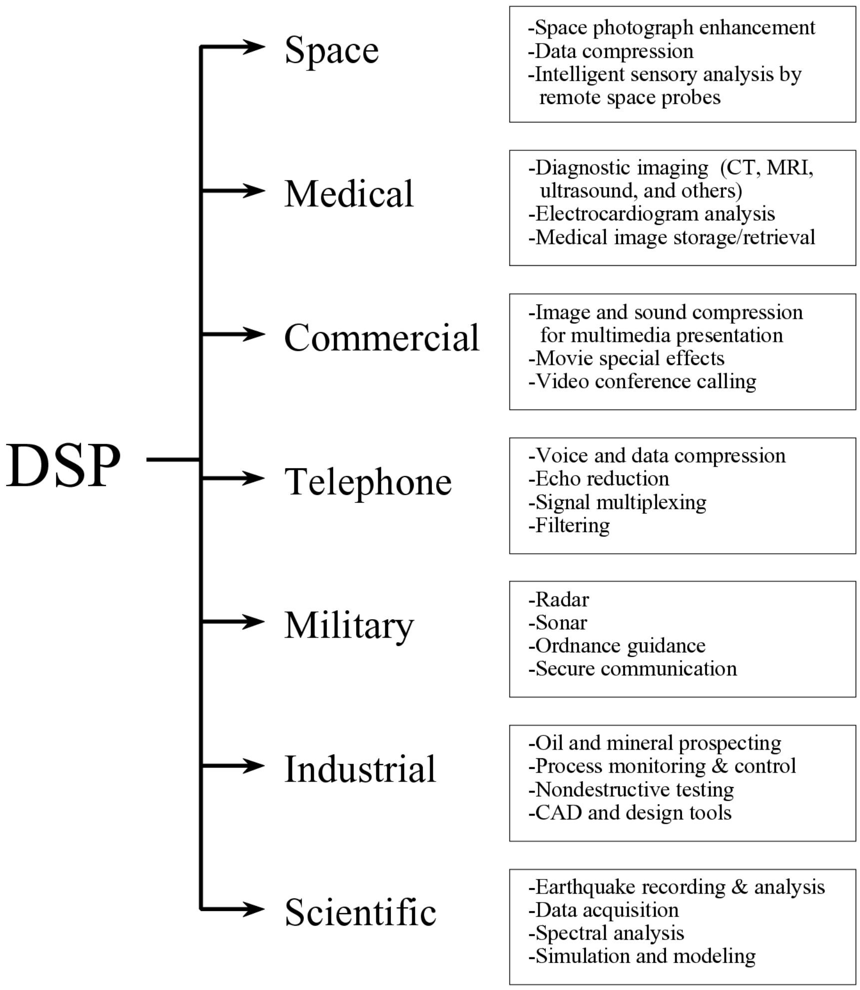

Figure 1: Anwendungen der digitalen Signalverarbeitung.

2 AD-Wandlung und DA-Wandlung

- Signal: Funktion einer oder mehrerer Variablen mit Informationen über das Verhalten oder die Natur physikalischer Erscheinungen.
- x-Achse: *unabhängige* Variable; y-Achse: *abhängige* Variable.
- Signal mit Zeit als unabhängige Variable: Zeitbereich.
- Signal mit Frequenz als unabhängiger Variable: Frequenzbereich.

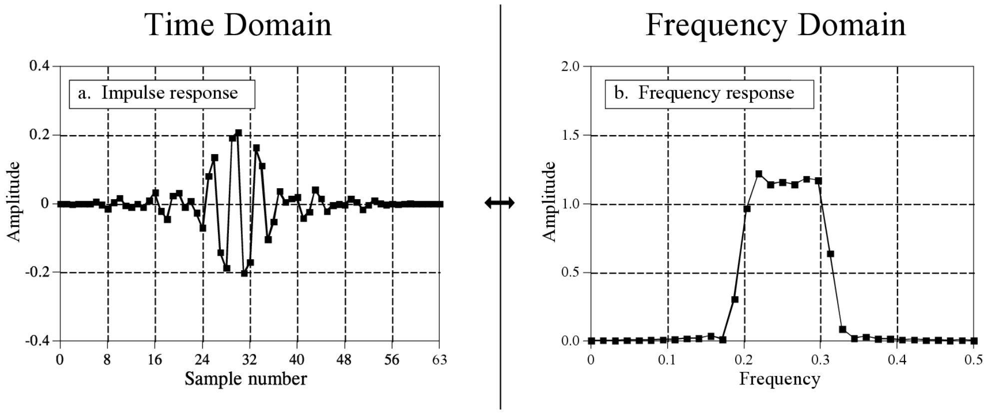

Figure 2: Time and Frequency Domain.

- Digitalisierung kontinuierlicher Signale: AD-Wandlung (Analog-to-Digital Conversion, ADC).
- Umkehrung: DA-Wandlung (Digital-to-Analog Conversion, DAC).

- Digitalisierung:
  1. Abtastphase (Sample-and-Hold, S/H): Eingangssignal wird in regelmäßigen Zeitabständen gemessen (gesampled). $\Rightarrow$

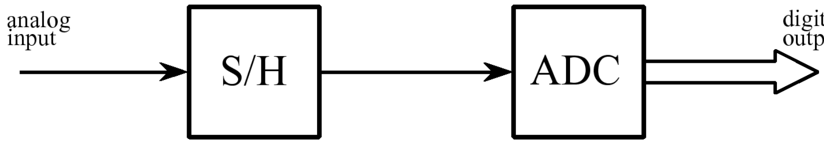

Figure 3: Digitalisierung.

Konvertierung der unabhängigen Variable von kontinierlich nach diskret.

2. Quantisierung: Rundung der abhängigen Variable auf eine ganze Zahl. $\Rightarrow$ Konvertierung der abhängigen Variable von kontinierlich nach diskret.

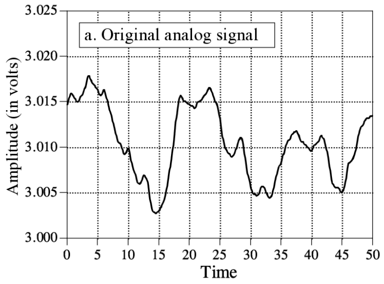

Figure 4: Eingangssignal.

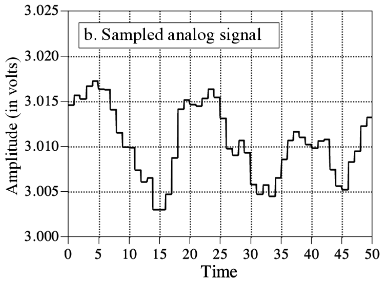

Figure 5: Gesampletes Eingangssignal.

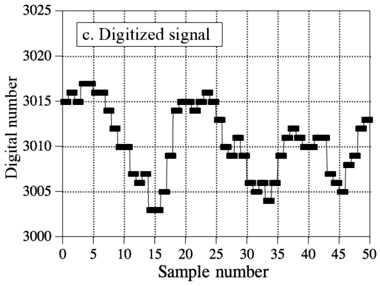

Figure 6: Digitalisiertes Signal.

3 Statistische Begriffe

- Seien N Werte $x_0, \ldots, x_{N-1}$ gegeben. Der Mittelwert ist definiert als $\mu = \frac{1}{N}\sum_{i=0}^{N-1} x_i$. In der Elektronik wird der Mittelwert üblicherweise als Direct Current (DC) bezeichnet, und AC (alternating current) beschreibt, wie ein Signal um den Mittelwert fluktuiert.

- Die Varianz σ^2 ist definiert als $\sigma^2 = \frac{1}{N-1}\sum_{i=0}^{N-1}(x_i - \mu)^2$; die Wurzel aus der Varianz wird als Standardabweichung bezeichnet. Die Standardabweichung beschreibt, wie weit ein Signal um den Mittelwert fluktuiert.

- Bei der obigen Definition von σ müssen alle Samples berücksichtigt werden, wenn neue Samples aufgenommen, dem Signal hinzugefügt werden und die neue Standardabweichung berechnet wird. Daher $\sigma^2 = \frac{1}{N-1}(\sum_{i=0}^{N-1} x_i^2 - (\frac{1}{N}(\sum_{i=0}^{N-1} x_i)^2)$.

- Der Quotient $SNR = \frac{\mu}{\sigma}$ wird als Rauschabstand (Signal-to-Noise Ratio) bezeichnet.

- Das Histogramm H stellt für jeden möglichen Wert die Anzahl der Samples dar, die diesen Wert haben. Ist M die Anzahl der möglichen Werte, dann gilt $N = \sum_{i=0}^{M-1} H_i$.

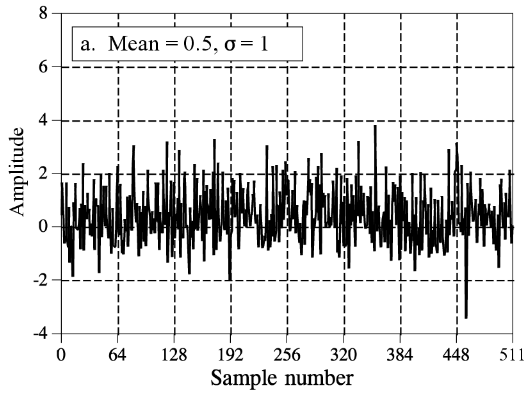

Figure 7: Mittelwert und Standardabweichung.

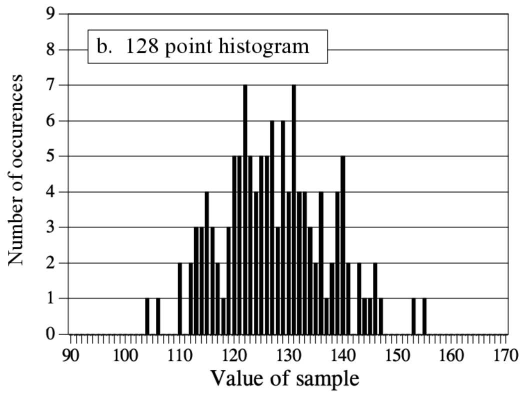

Figure 8: Histogramm.

4 AD-Wandlung und DA-Wandlung (weiter)

Quantisierungsfehler

- Speichern der Variablen in einer gegebenen Zahldarstellung $\Rightarrow$ Quantisierungsfehler.
- n-bit Unsigned Integers $\Rightarrow$ Abweichung zwischen $+-\frac{1}{2}2^{-n}$.

- Wert des digitalen Signals: Wert des kontinierlichen Signals plus Quantisierungsfehler.
- Bitbreite bestimmt Präzision.

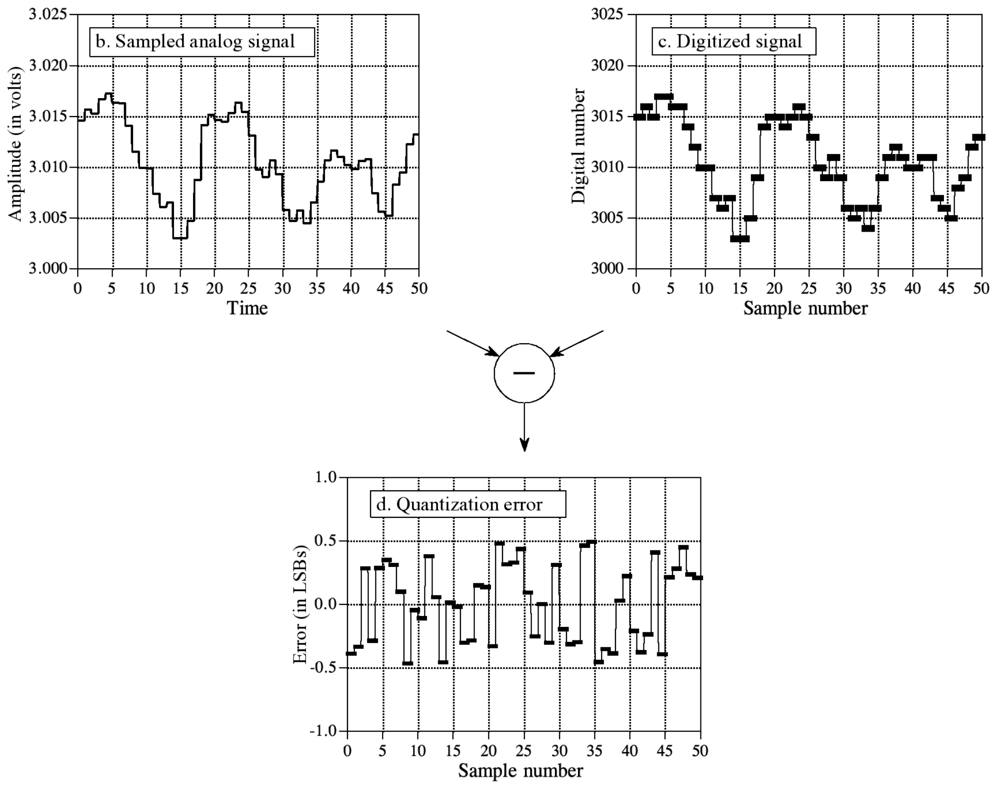

Figure 9: Quantisierungsfehler.

- Auswirkungen: Hinzufügung von weißem Rauschen zum Signal.
- Weißes Rauschen: gleichverteilt zwischen $+-\frac{1}{2}LSB$, Mittelwert von 0 und Standardabweichung von $\frac{1}{\sqrt{12}}LSB$.

Fourieranalyse

- Fourieranalyse: Beschreibung eines physikalischen Prozesses in Abhängigkeit der zeitlichen Änderung einer Größe $h(t)$ durch

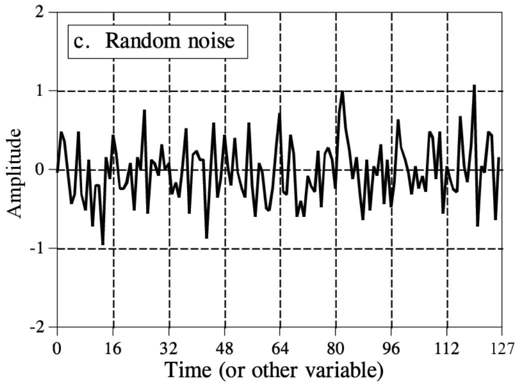

Figure 10: Weißes Rauschen.

eine Summe von trigonometrischen Funktionen, die charakterisiert sind durch ihre Amplituden $H(f)$ in Abhängigkeit der Frequenz.

- $h(t)$ und $H(f)$ sind zwei verschiedene Repräsentationen der gleichen Funktion.
- Mit der Fouriertransformation kann man zwischen den beiden Darstellungen wechseln.
- Trägt man die Frequenzen auf der x-Achse und die zugehörigen Amplituden auf der y-Achse auf, erhält man das *Frequenzspektrum.*

Rekonstruierbares Sampling (Proper Sampling)

- Das Sampling ist rekonstruierbar, wenn das analoge Signal aus den Samples rekonstruiert werden kann.
- *Nyquist Sampling Theorem*: Mißt man eine Funktion $h(t)$ in den Zeitabständen Δt, so gibt es eine spezielle Frequenz f_c, die

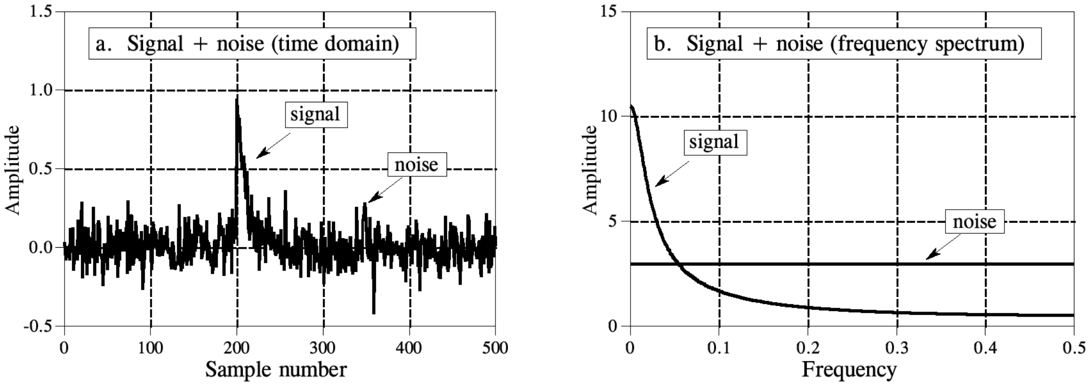

Figure 11: Frequenzspektrum.

sogenannte *Nyquist-Frequenz*

$$f_c = \frac{1}{2\Delta t}$$

welche die Obergrenze der durch die Fourier-Transformation noch erfaßbaren Frequenzen darstellt.

- Ein kontinierliches Signal kann nur rekonstruierbar gesampled werden, wenn es keine Frequenzen über der Nyquist-Frequenz enthält (Nyquist-Rate = halbe Sampling-Rate).
- Transformiert man ein Signal, das Frequenzen oberhalb der Nyquist-Frequenz enthält, werden diese auf die Frequenzen unterhalb der Nyquist-Frequenz abgebildet. Sie erzeugen dabei sogenannte Spiegelfrequenzen (aliasing), die das eigentliche Signal verfälschen.

- Die Informationsmenge im digitalen Signal ist zweifach beschränkt:
  - Die Anzahl der Bits pro Sample begrenzt die Auflösung der abhängigen Variablen. Kleine Änderungen in der Ampli-

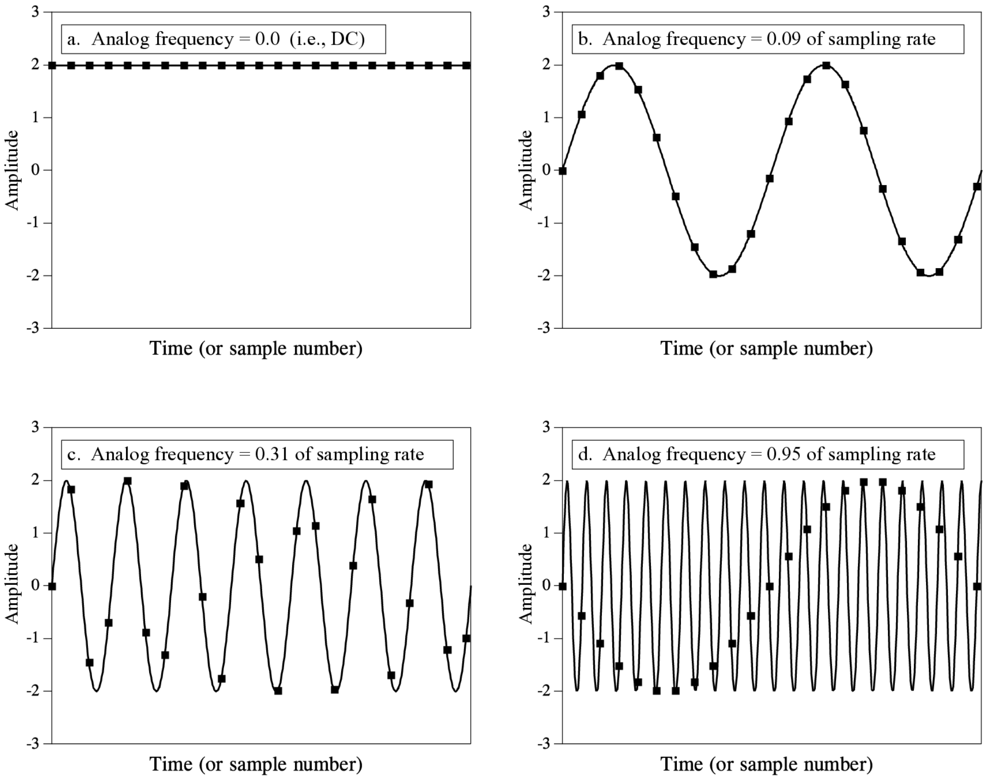

Figure 12: Sampling und Aliasing.

tude des Eingangssignals können also im Quantisierungsrauschen verloren gehen.

– Die Sampling-Rate limitiert die Auflösung der unabhängigen Variable, d.h. eng beieinanderliegende Änderungen im analogen Signal zwischen den Stichproben (Samples) können verloren gehen.

- DAC: ähnlich wie ADC. In der Regel halten DACs den zuletzt empfangenen Wert, bis ein neues Sample eintrifft (zeroth-order hold). Anschließend: Filter, der aus dem zeroth-order hold das kontinierliche Ausgangssignal erzeugt.

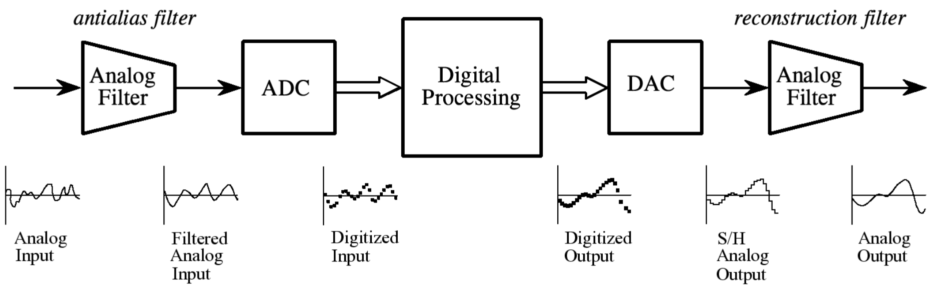

Figure 13: Einfaches DSP-System.

Einfaches DSP-System:

- Vor Eintritt in den ADC: elektronischer Tiefpaß Filter, der Frequenzen über der Nyquist-Frequenz entfernt (Antialias-Filter).
- Am anderen Ende vor DAC: weiterer Low-pass Filter, der auf die Nyquist-Frequenz gesetzt ist (Reconstruction Filter).
- Trend: Ersetzung analoger Schaltkreise durch digitale Algorithmen.

- Grund: Kosten u. Leistung: Aufgrund physikalischer Beschränkungen bei analogen Schaltkreisen können digitale Algorithmen erheblich höhere Präzision liefern.

5 Lineare Systeme

- Divide-and-Conquer Strategie *Superposition*:
  - Aufbrechen des Signals in Komponenten, die einzeln bearbeitet werden.
  - Vereinigung der Resultate.
  - Voraussetzung: System ist linear
- Input-Output-System S (kurz: System): Prozess, der ein Eingangssignal in ein Ausgangssignal konvertiert.
- Eingangs- und Ausgangssignal kontinierlich: kontinierliches System
- Eingangs- und Ausgangssignal diskret: diskretes System.
- Ein System ist linear, wenn es die folgenden Eigenschaften besitzt:
  - Homogenität: Eine Skalierung des Eingangssignals durch einen Faktor k führt auch zur Skalierung des Ausgangssignals durch k. Gilt $S(x) = y, x'[n] = kx[n] \; \forall n$, dann gilt $S'(x) = y'$, wobei $y'[m] = ky[m] \; \forall m$.
  - Additivität: Gilt $S(x_1) = y_1$ und $S(x_2) = y_2$, dann gilt auch $S(x_1 + x_2) = y_1 + y_2$.

- Weiterhin oft erforderlich: *Verschiebungsinvarianz*: Eine Verschiebung im Eingangssignal führt zu einer identischen Verschiebung im Ausgangssignal. Gilt $S(x) = y$ und $x'[n+s] = x[n]\ \forall n$, dann gilt auch $S(x') = y'$ mit $y'[m+s] = y[m]\ \forall m$. (Die Eigenschaften des Systems ändern sich nicht mit der Zeit.)
- Synthese: Komposition eines Signals aus verschiedenen Teilsignalen durch Skalierung und Addition.
- Dekomposition: Zerlegung eines Signals in zwei oder mehrere additive Komponenten.

Fundamentales Konzept der DSP

- Jedes Signal x kann in ein Gruppe additiver Komponenten $x_1, x_2, \ldots, x_k$ zerlegt werden. Werden diese Komponenten durch ein lineares System geführt, ergeben sich Ausgangssignale y_1, y_2, ..., y_k. Durch Synthese dieser Ausgangssignale ergibt sich dasselbe Signal y, das ausgegeben wird, wenn x direkt durch das System gegeben wird. Ein- und Ausgangssignal werden also als Überlagerung einfacherer Signale aufgefaßt.

Dekompositionsmethoden

- Impulsdekomposition. Impuls: Signal mit einem einzigen von Null verschiedenen Wert. Ist bekannt, wie ein System auf einen Impuls reagiert, kann die Ausgabe des Systems für jede Eingabe berechnet werden ($\rightarrow$ Konvolution).
- Interlaced Decomposition. Basis für Fast Fourier Transformation.

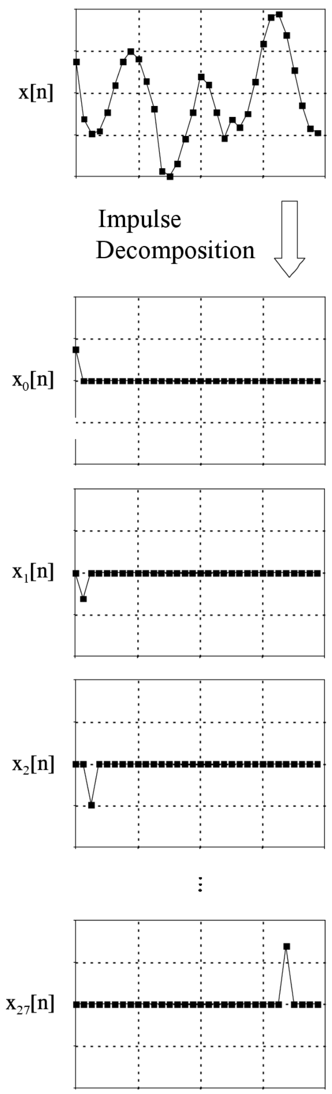

Figure 14: Impulse Decomposition.

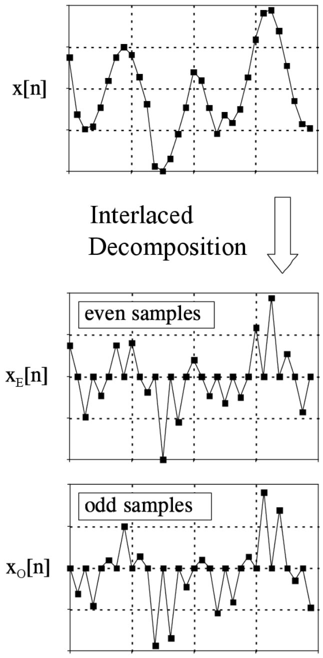

Figure 15: Interlaced Decomposition.

- Fourier Dekomposition.

6 Konvolution

- Einheitsimpuls (Delta-Funktion) δ ist ein normalisierter Impuls. Sample 0 hat den Wert 1, alle anderen Samples den Wert 0.
- Impulsantwort h (impulse response): Ausgangssignal des Systems für eine Delta-Funktion als Eingabe.

Definition

Sei x ein N-Punkte Eingangssignal $(x[0], \ldots, x[N-1])$ und h ein M-Punkte Signal $(h[0], \ldots, h[M-1])$. Dann ist die Konvolution y

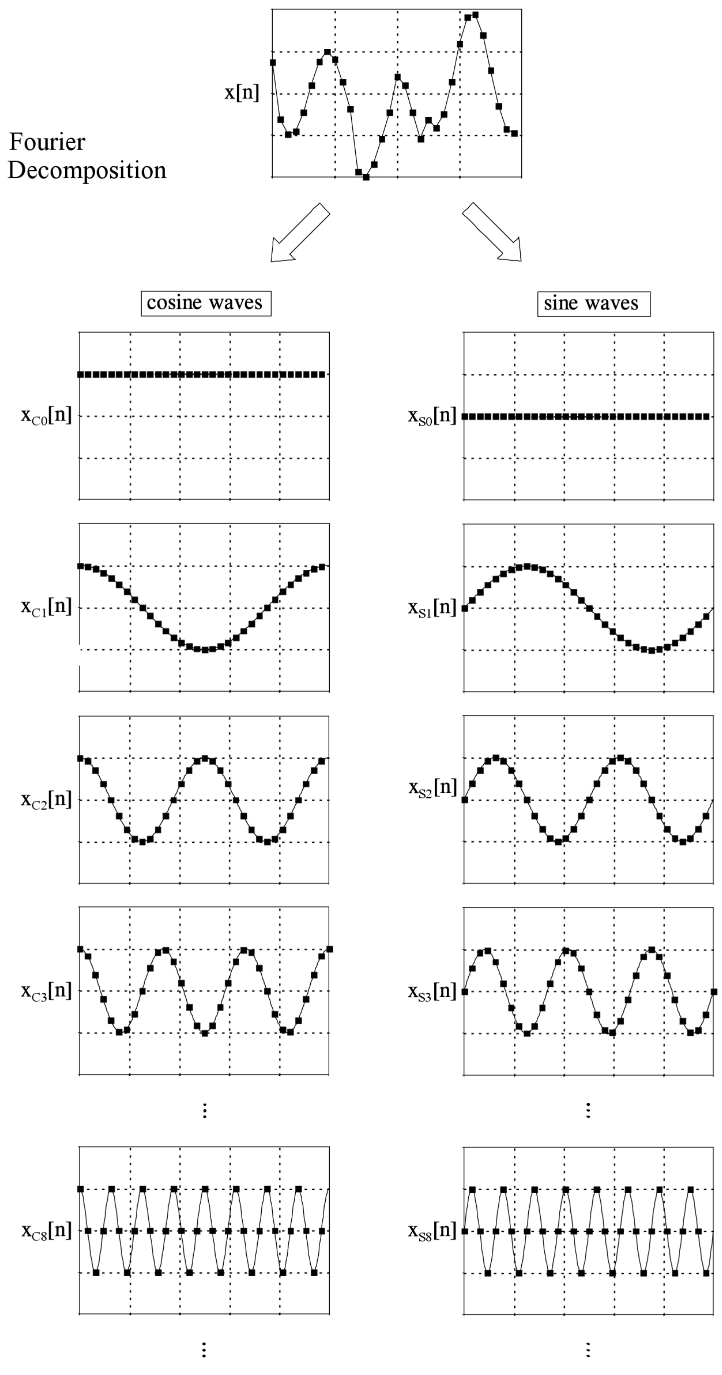

Figure 16: Fourier Decomposition.

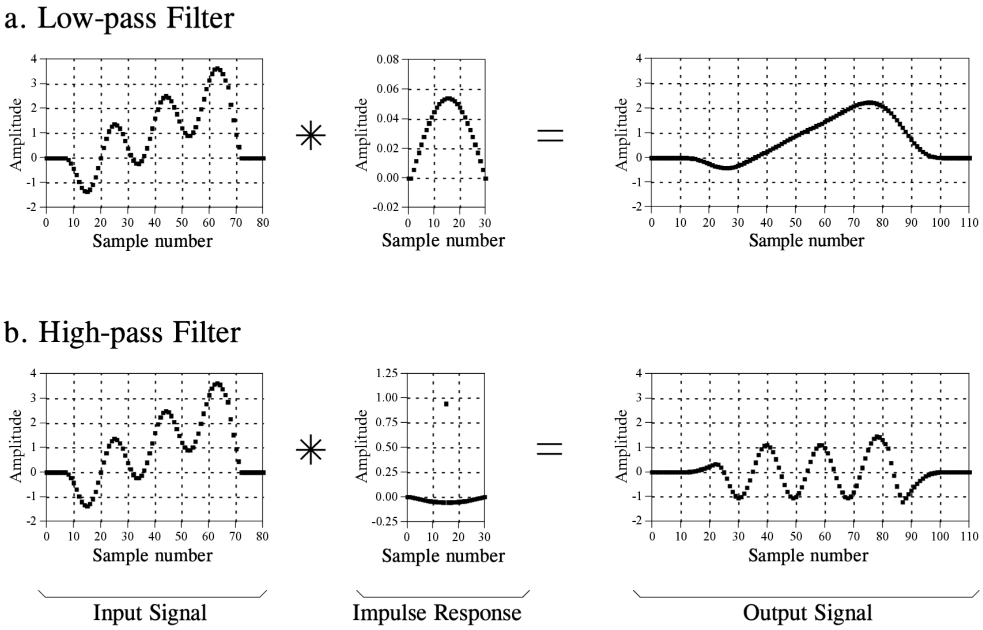

Figure 17: Konvolution.

von h und x das folgende $N+M-1$-Punkte Signal $(y[0], \ldots, y[N+M-1]$:

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

Kurzschreibweise: $y = h * x$.

- Padding / End effect problems.

Eigenschaften der Konvolution

- Kommutativität: $a * b = b * a$
- Assoziativität: $(a * b) * c = a * (b * c)$
- Distributivität: $a * b + a * c = a * (b + c)$
- $x * \delta = y$ mit $y[j] = x[j] \; \forall \; j = 0, \ldots, N-1$

- $x * k\delta = y$ mit $y[j] = kx[j] \; \forall \; j = 0, \ldots, N-1$
- Sei $\delta'[j+s] = \delta[j]$, $\delta'[z] = 0$ für $z = 0, \ldots, s-1$, $s \in \mathbb{N}$. Dann gilt $x * \delta' = y$ mit $y[j] = x[j+s] \; \forall \; j = 0, \ldots, N-1$x

7 Digitale Filter

- Entwurf digitaler Filter durch Entwurf einer geeigneten Impulsantwort. Beispiele: Radarerkennung, Echounterdrückung.
- Tiefpaß- und Hochpaß-Filter.
- Radarsysteme: Gegeben ein Signal einer bestimmten Form, wie kann man am besten herausfinden, wo (oder ob) das Signal in einem anderen Signal auftritt.

Definition

Sei t das zu erkennende Signal, a das Eingangssignal; dann ist die Kreuzkorrelation von a und t definiert als

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

- Wert eines Samples in der Kreuzkorrelation: Maß, wie sehr das empfangene Signal an dieser Stelle dem Zielsignal ähnelt.
- Der Wert der Kreuzkorellation wird maximiert, wenn das Zielsignal dieselben Eigenschaften wie das empfangene Signal hat.
- Korrelation wird dazu verwendet, ein bekanntes Signal in weißem Rauschen zu erkennen (matched filtering).

- Berechnung der Kreuzkorrelation durch Konvolution. Dazu: Umdrehen eines der beiden zu korrelierenden Signale.

Arten digitaler Filter

- Endlicher Impulsantwortfilter (Finite Impulse Response Filter, FIR): Konvolution des Eingangssignals mit der Impulsantwort des digitalen Filters (Skalierung + Addition).
- Rekursiver Filter / Unendlicher Impulsantwortfilter (Infinite Impulse Response Filter, IIR): Rekursive Filter verwenden zur Berechnung neuer Punkte zuvor berechnete Werte der Ausgabe zusammen mit Punkten der Eingabe. Typische Impulsantwort: Sinusfunktionen mit exponentiell abnehmenden Amplituden. Dadurch: unendliche Impulsantwort.

Definition Rekursiver Filter

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

- a, b: Koeffizienten
- x: Eingangssignal
- y: Ausgangssignal
- L: Polanzahl.

Finite vs. infinite impulse response filters

- Vorteil rekursiver Filter: Vermeidung langer Konvolutionen. Schnelle Berechnung.

- Nachteil rekursiver Filter: Beschränkte Leistung (Rundungsfehler z.B. bei Repräsentation der Koeffizienten) und geringere Flexibilität.

8 Implementierung

- Einfacher Algorithmus möglich. Problem: große Anzahl an Additionen und Multiplikationen. Daher: in fast allen DSPs Multiply-Accumulate Unit (Multiplikation + Addition zu Akkumulator in 1 Taktzyklus).
- FFT Konvolution produziert exakt die gleichen Ergebnisse wie die Standardkonvolution, aber die Ausführungszeit kann deutlich gesenkt werden.
- Konvolution ist die bevorzugte Technik, wenn die DFT weniger als etwa 32 Punkte hat, sonst wird die FFT verwendet.

9 Diskrete Fouriertransformation

Gegeben: Stückweise stetige Funktion $f : \mathbb{R} \to \mathbb{R}$ mit Periode 2π, $f(x + 2\pi) = f(x)\ \forall x \in \mathbb{R}$. f darf Sprungstellen aufweisen; für eine Unstetigkeitsstelle x_0 existieren die Grenzwerte y_0^- und y_0^+ mit $y_0^- = \lim_{h\to 0^+} f(x_0 - h)$ und $y_0^+ = \lim_{h\to 0^+} f(x_0 + h$ und sind endlich. Zerlege $[0, 2\pi]$ in N Teilintervalle mit Schrittweite $h = \frac{2\pi}{N}$ und betrachte die Stützstellen $x_j = h_j = \frac{2\pi j}{N}$, $j = 0, 1, \ldots, N$. Definiere

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

und sei $n = \frac{N}{2} \in \mathbb{N}$. Dann ist

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

das eindeutig bestimmte Fourierpolynom zu den Stützstellen x_j mit den Stützwerten $f(x_j)$, $j = 1, \ldots, N$. (Stützstellen: Abtastzeitpunkten (Samplingzeitpunkten), Stützwerte: Werte des Eingangssignals)

Also:

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

a_k^* und b_k^* sind die $n+1 = \frac{N}{2}+1$ Koeffizienten, die zur Bildung von g_n^* benötigt werden. Definiere ferner

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

DFT-Basisfunktionen:

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

k: Frequenz der Basisfunktionen, d.h. Anzahl der vollständigen Perioden über die N Punkte des Signals ($f = \frac{k}{N}$).

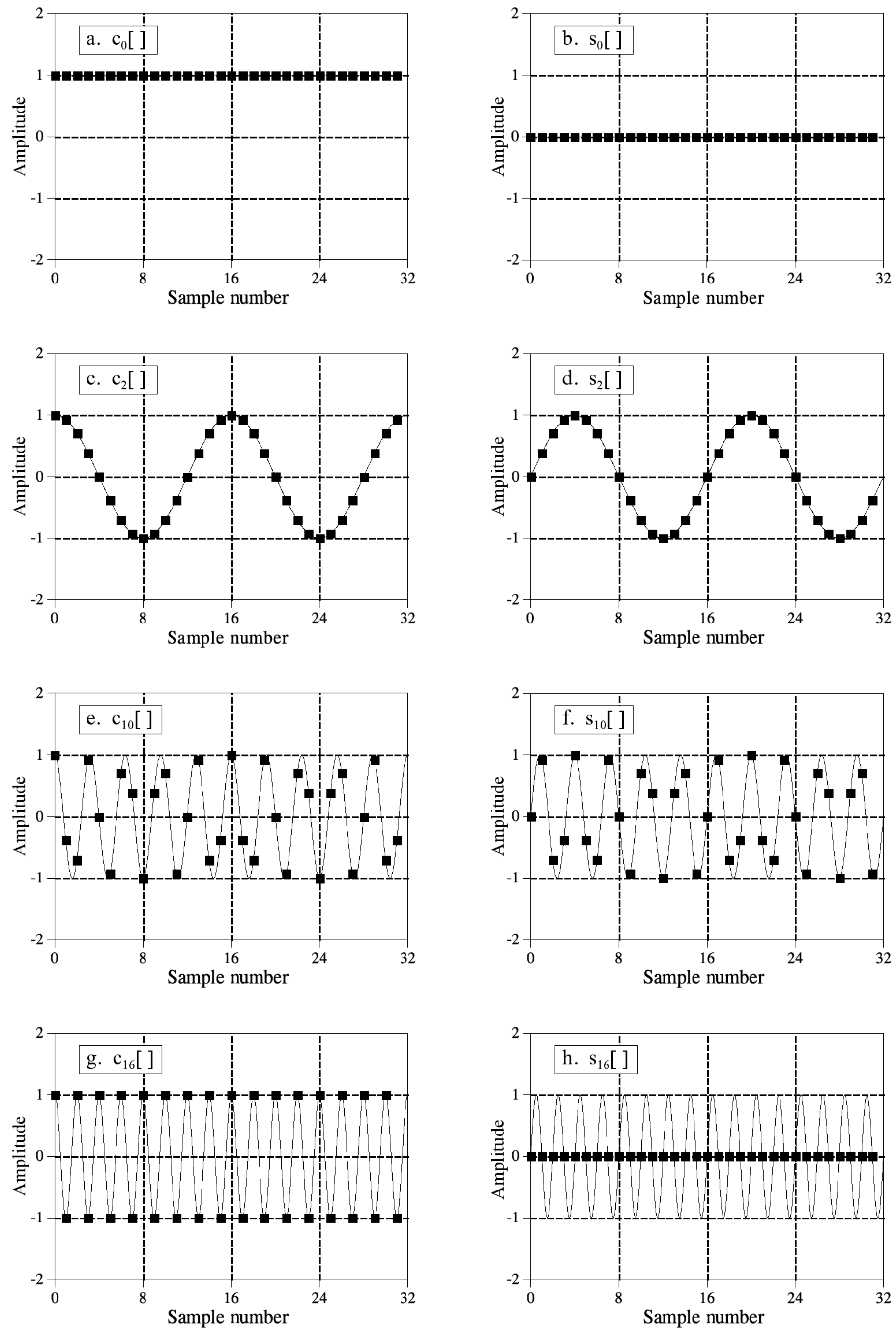

Figure 18: DFT-Basisfunktionen.

Diskrete Fouriertransformation

- transformiert ein N-Punkte Eingangssignal in zwei $N/2+1$-Punkte Ausgangssignale.
- Eingangssignal: zu zerlegendes Signal.
- Ausgangssignale: definieren Amplituden der DFT-Basisfunktionen.
- Eingangssignal im Zeitbereich.
- Anzahl und Frequenz der DFT-Basisfunktionen ist fixiert: Ausgangssignal in Abhängigkeit der Frequenz (im Frequenzbereich).
- Die DFT (Forward-DFT) berechnet somit für ein gegebenes zeitabhängiges Signal (im Zeitbereich) die korrespondierende frequenzabhängige Darstellung (Frequenzbereich). a_k^*: Amplituden der Cosinusfunktionen, b_k^*: Amplituden der Sinusfunktionen.
- a_k und b_k entsprechen dem Realteil bzw. dem Imaginärteil des Frequenzbereichs.

Inverse DFT (Synthesegleichung)

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

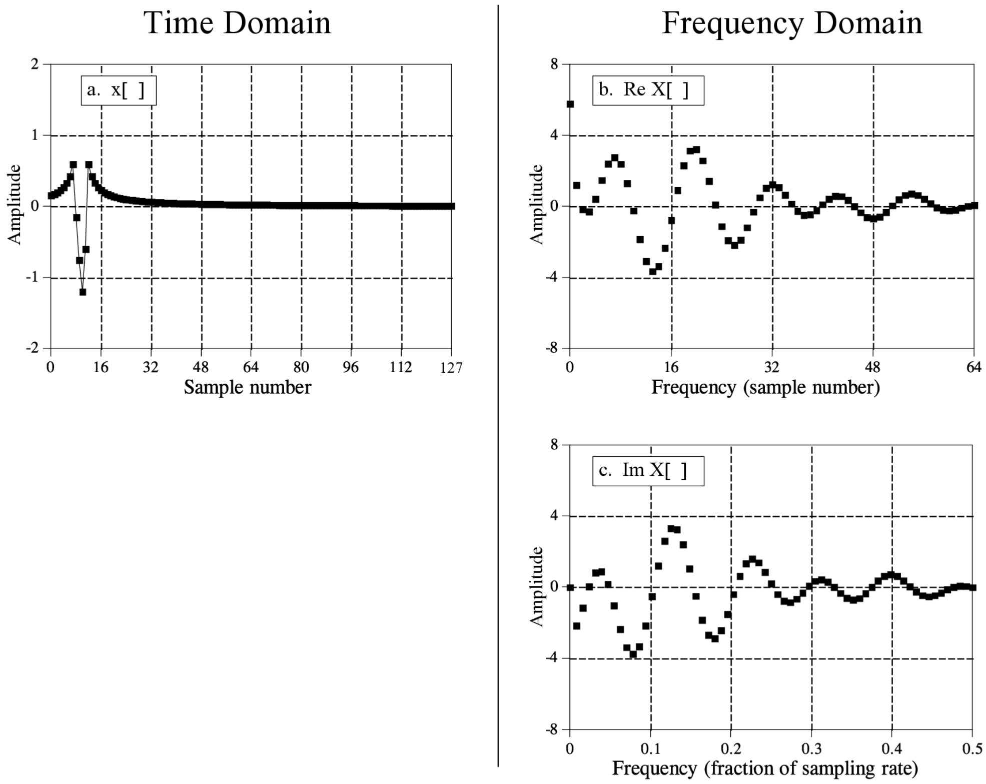

Figure 19: Zeit- und Frequenzbereich.

Berechnung der DFT (Analyse)

- Lösung eines Gleichungssystems, bei dem N Variablen des Zeitbereichs gegeben sind, und die N Werte des Frequenzbereichs zu berechnen sind. Benötigt werden somit N linear unabhängige Gleichungen $\Rightarrow$ Ineffizient.
- Korrelation: Entdeckung eines bekannten Signals in einem anderen Signal. Die a_k und b_k werden direkt durch Multiplikationen und Additionen berechnet. Dadurch wird das Eingangssignal mit der jeder Basisfunktion korreliert.
- Fast Fourier Transformation: Zerlegung einer DFT mit N Punkten in N DFTs, die jeweils einen Punkt enthalten.

Frequenzantwort (frequency response)

- Frequenzantwort eines Systems: Fouriertransformation der Impulsantwort.
- Sie beschreibt, wie ein System die Amplitude und Phase von hindurchlaufenden Cosinusfunktionen ändert.
- Charakterisiert das System vollständig.

Zeitbereich und Frequenzbereich

- Zeitbereich: Konvolution des Eingangssignals mit Impulsantwort ($x[n] * h[n] = y[n]$).
- Frequenzbereich: Multiplikation des Eingangsspektrums mit Frequenzantwort.

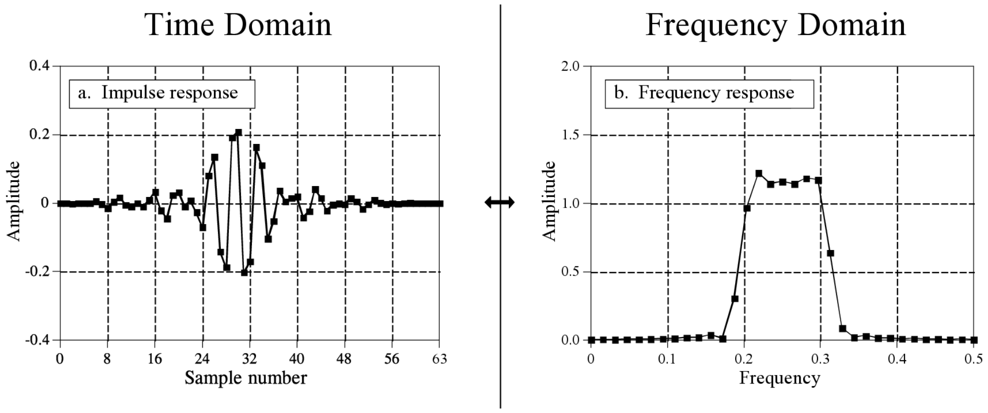

Figure 20: Impuls- und Frequenzantwort.

- DFT und IDFT definieren die Beziehung zwischen den Signalen beider Bereiche.

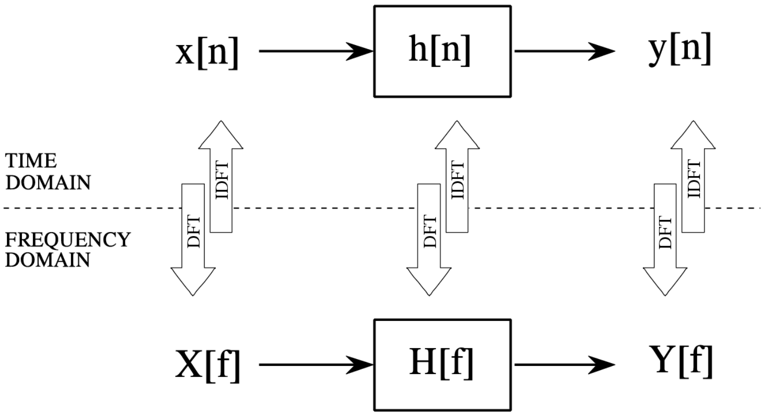

Figure 21: DCT und IDCT.

Alternative Berechnung

- Gegeben: Eingangssignal und Impulsantwort
- Transformation in Frequenzbereich
- Multiplikation

- Rücktransformation in Zeitbereich.
- Kein Geschwindigkeitsgewinn bei Standardkonvolution, erst mit FFT.

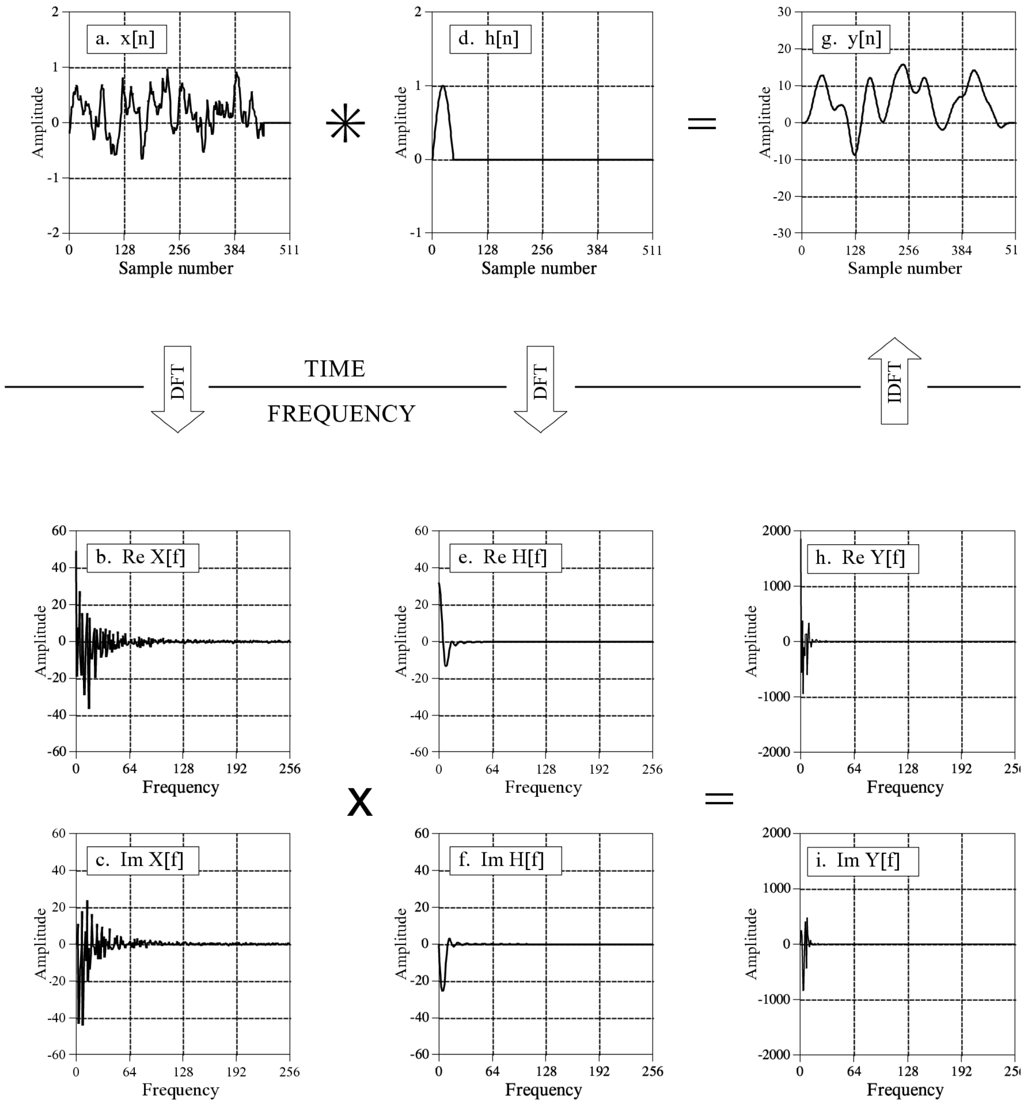

Figure 22: Konvolution im Frequenzbereich.

10 FFT – Fast Fourier Transformation

- Die FFT ist ein Algorithmus zur Berechnung der komplexen DFT. Daher: DFT-Datem müssen in komplexes Format konvertiert werden.

- Ist ein N Punkte Signal gegeben, werden dessen Punkte in den Realteil des Zeitbereichs der komplexen DFT geschrieben, und alle Samples im Imaginärteil auf 0 gesetzt.

- Die Berechnung der komplexen DFT ergibt ein komplexes Signal X des Frequenzbereichs. Die Samples von 0 bis $\frac{N}{2}$ des Realteils und des Imaginärteils dieses Signals entsprechen dem Spektrum der realen DFT (d.h. den Werten a_k und b_k).

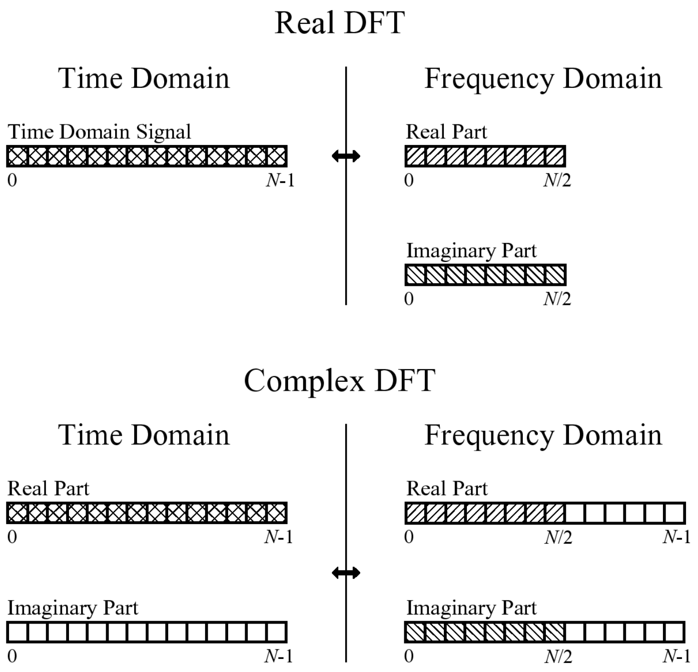

Figure 23: Reale und komplexe DFT.

- Die diskrete komplexe Fouriertransformation ist wie folgt definiert:

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

FFT

- Dekomposition eines N-Punkt Signals des Zeitbereichs in N 1-Punkt Signale aus dem Zeitbereich.
- Berechnung der N Frequenzspektra, die den N Signalen des Frequenzbereichs entsprechen.
- Das Frequenzspektrum eines 1-Punkt Signal ist das Signal selbst, da eine Fouriertransformation der Ordnung 1 mit den zu transformierenden Werten übereinstimmt.
- Synthese der N Spektra in ein einziges Frequenzspektrum.

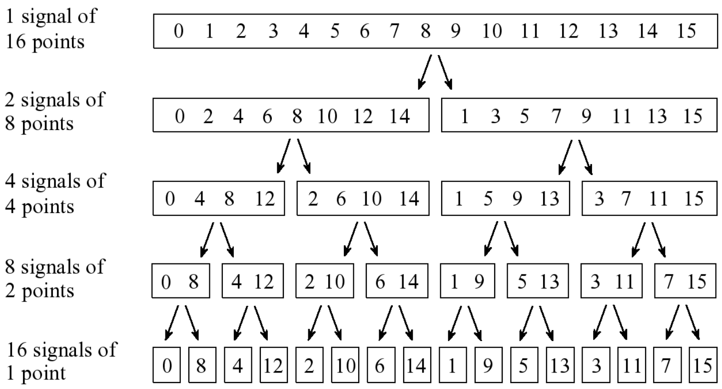

Figure 24: FFT decomposition.

Interlacing

- In jeder Stufe der FFT-Zerlegung: Trennung der gerade und ungerade numerierten Samples (Interlacing).
- $\Rightarrow$ Sortieren der Samples in umgekehrter Bitordnung (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.

Synthese

- Kombination der N Frequenzspektra in umgekehrter Reihenfolge ihrer Zerlegung.
- Seien zwei Frequenzspektra aus 4 Punkten gegeben, die in ein einzelnes Frequenzspektrum aus 8 Punkten kombiniert werden.
- Das Auffüllen des Timedomains mit Nullen entspricht einer Duplikation des Frequenzspektrums.

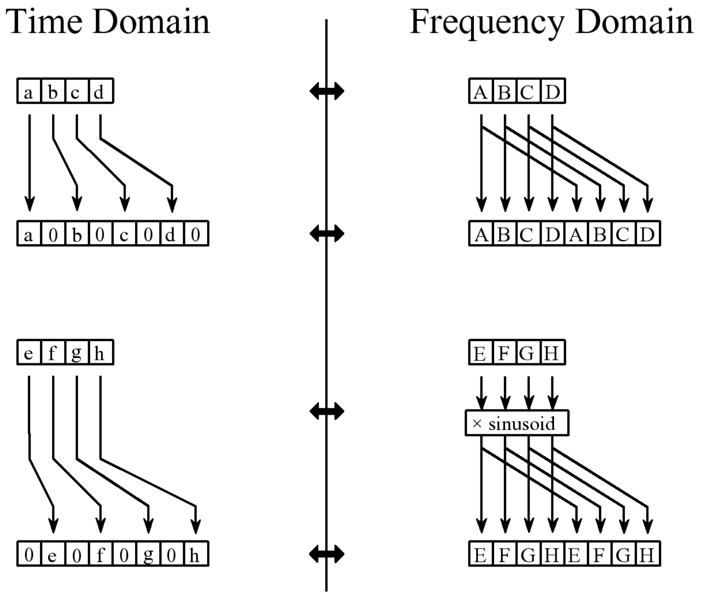

Figure 26: FFT Synthesis. Interlacing.

- Einsetzen von Nullen entspricht Addition nach Verschiebung eines Signals um ein Sample nach rechts. Diese Timedomainverschiebung entspricht der Multiplikation des Spektrums mit einer Sinusfunktion.
- Verschiebung im Zeitbereich: Konvolution des Signals mit verschobener Delta-Funktion. Dadurch: Multiplikation des Spektrums des Signals mit dem Spektrum der verschobenen Delta-Funktion. Das Spektrum einer verschobenen Delta-Funktion ist eine Sinusfunktion.

FFT-Schmetterling

- Eingabe: zwei komplexe Punkte.
- Konvertierung in zwei andere komplexe Punkte.

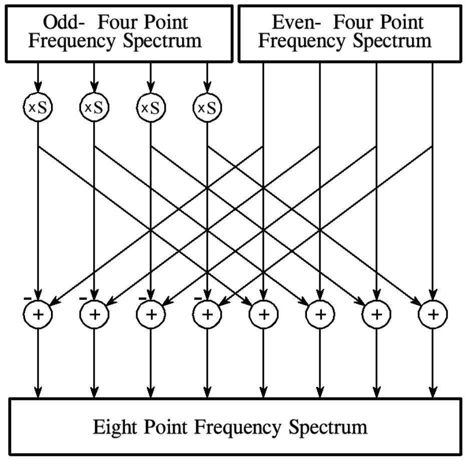

Figure 27: FFT synthesis.

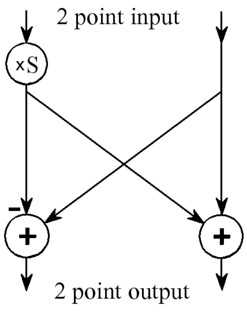

Figure 28: FFT butterfly.

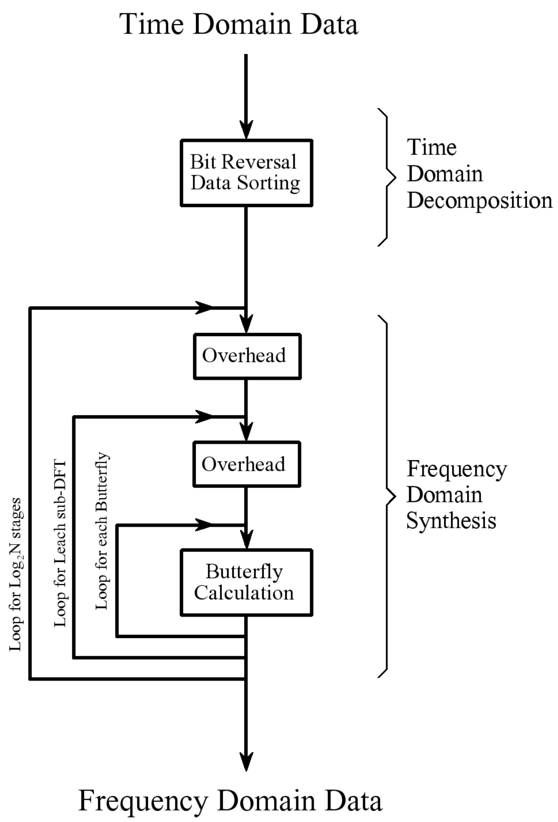

Figure 29: FFT Algorithmus.

Bit-Reverse Addressing

- Bit-reverse addressing mode wurde speziell für die FFT entwickelt und ist in vielen DSPs implementiert.
- Beispiel TriCore: Durch zwei aufeinanderfolgende Adreßregister wird zirkulärer Puffers implementiert.
- Gerade numeriertes Register: Basisadresse (Anfang des Puffers)
- Ungerade numeriertes Register: Elementgröße und Index.
- Beispiel: Sei gegeben ein 8-elementiges Array $A = 0, 1, 2, 3, 4, 5, 6, 7$, dh $A[i] = i$. Die Zugriffsreihenfolge innerhalb der FFT ist $0, 4, 2, 6, 1, 5, 3, 7$.

```
mov.a  a10,#8             ; Iterationsanzahl
movh.a a3,#16              ; Größe=16 (bit), index=0
lea    a2, val            ; base address = adr(val) = 0x00080000
_L: ld.h   d0,[a2/a3+r]   ; a3 = 0x00080008, 0x00080004,
                          ; 0x0008000c, 0x00080002, ...
...
loop a10, _L
```