Vorlesung Signalprozessoren

Vortrag: Diskrete Fouriertransformation und Anwendungen

Einleitung

Mitte der siebziger Jahre begann in den USA die Entwicklung eines technischen Systems, das im weiteren Sinne zu den militärischen Frühwarnsystemen gerechnet werden kann. Aufgabe dieses Systems war es, den Funkverkehr der sowjetischen Luftwaffe abzuhören. Dies allein war aber noch nicht die eigentliche Aufgabe. Vielmehr sollten mit Hilfe der Computertechnik die Stimmen der sowjetischen Piloten identifiziert werden, so daß man jeden Piloten an Hand seines Stimmusters erkennen konnte. Der Sinn dieses Aufwandes war folgender: ein Großteil der sowjetischen Piloten war nicht in Osteuropa stationiert. Vor einem eventuellen Angriff auf Westeuropa - so die Theorie - wären aber verstärkt neue Piloten nach Osteuropa abkommandiert worden. Und diese neuen Piloten konnte man nur auf Grund ihrer Stimme eindeutig erkennen [HACK].

Glücklicherweise wurde dieses Systems nur angewendet, aber nie auf seine Effektivität überprüft. Das eigentlich erstaunliche aber daran war die Technik: wie kann man einen Menschen nur mit seiner - evtl. sogar verrauschten - Stimme erkennen?

Heutzutage ist diese Stimmerkennung keine Hexerei mehr. Mit der im größeren Verbreitung von digitalen Signalprozessoren gibt es inzwischen vielfältige Anwendungen dieser Technik:

- Soundkarten im PC, die auf die Benutzerstimme reagieren
- behindertengerechte Geräte, die eine Steuerung von Aktoren nur über die Stimme ermöglichen
- Sicherheitssysteme, die Türen nur bei bekannter Stimme öffnen
- selbst einem so simplen Gegenstand wie einem Lichtschalter kann über einen Mikrochip Leben eingehaucht werden: er reagiert auf die Kommandos "Licht an", "Licht aus"

Die Frage, die sich der Techniker stellt, liegt auf der Hand: wie geht das mit der Spracherkennung? Im wesentlichen beruht das Prinzip der Spracherkennung darauf, daß jeder Mensch in seiner Sprache bei jedem Laut bestimmte, für ihn charakteristische Frequenzen erzeugt.

Also ist die Laut- und Stimmerkennung im wesentlichen auf die Erkennung von Signalen bestimmter Frequenzen und Amplituden in einem bestimmten Zeitabschnitt reduziert.

Signalfrequenzen mit bestimmten Amplituden zu erkennen, ist natürlich eine Domäne der Fouriertransformation. Und genau da liegt das Problem: die Fouriertransformation beschäftigt sich mit kontinuierlichen (im mathematischen Sinn stetigen) Funktionen, und setzt unendlich lange Meßdauer voraus. Im vorliegenden Fall ändern sich die Signale ständig, so daß ein Signal in bestimmte kurze Zeitabschnitte unterteilt werden muß. Damit liegen aber keine stetige Funktionen vor, sondern Zahlenfolgen mit einer bestimmten Anzahl von Werten und einer festgelegten Zeitdauer, die von einem A/D-Umsetzer im Speicher einer Recheneinheit abgelegt wurden.

So bleibt uns nichts anderes übrig, als die Begriffe "Stetigkeit" und "Unendlichkeit" über Bord zu werfen, und eine Theorie zur Fouriertransformation endlicher Zahlenfolgen zu entwerfen.

Herleitung der Transformationsgleichungen

Bekanntlich ist die Fourierreihe der Sonderfall der Fouriertransformation genau dann, wenn die zu transformierende Funktion periodisch ist.

Es gilt also (z.B. nach [OPPE]):

Transformation aperiodischer Funktionen:

$$X(f) = \int_{-\infty}^{\infty} x(t) \cdot \mathrm{e}^{-j2\pi ft} \mathrm{d}t \qquad (1)$$

$$\dim X = \frac{\dim x}{Hz}$$

Transformation periodischer Funktionen:

$$c_m = \frac{1}{T} \int_0^T x(t) \cdot \mathrm{e}^{-jm2\pi ft} \mathrm{d}t \qquad (2)$$

$$\dim c = \dim x$$

T : Periodendauer von f

Im letzten Fall wurden die Koeffizienten der komplexen Fourierreihe zur Darstellung der Spektralanteile gewählt.

Betrachtet werden soll nun folgender Fall: es liegt eine Funktion vor, die durch ein Zeitfenster ausmaskiert wurde, d.h.

$$\tilde{x}(t) = \begin{cases} x(t) & ,0 \le t < T_F \\ 0 & ,sonst \end{cases} \qquad (3)$$

Hierbei ist T_F die Zeitdauer des Zeitfensters. Diese Funktion kann wie folgt aussehen:

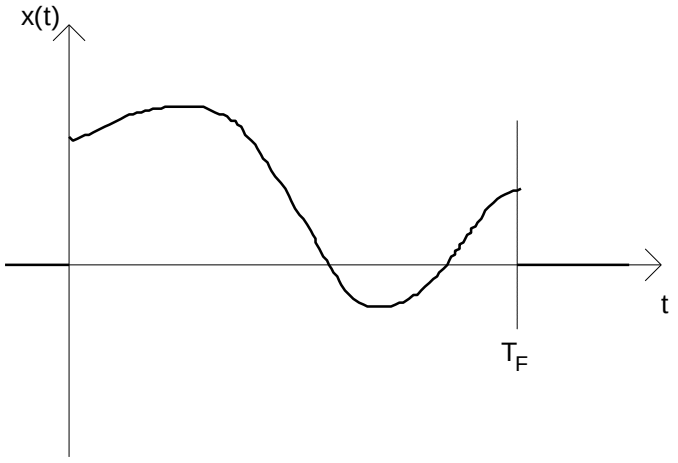


Bild 1: Funktion, die durch ein Fenster nur in einem Intervall $[0,T_F]$ existiert

Diese Funktion wird nun mit der Periode T_F periodisch fortgesetzt. Man erhält die folgende Funktion:

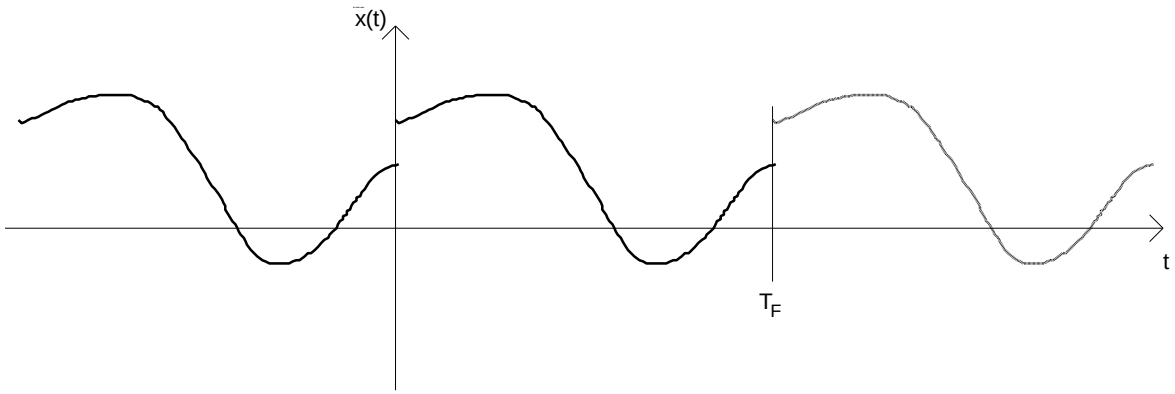


Bild 2: Periodische Fortsetzung einer Funktion

Eine periodische Funktion läßt sich bekanntlich in eine Fourierreihe entwickeln. Man macht nun mit (1) Ansatz für die komplexen Fourierkoeffizienten:

$$X_m = \frac{1}{T}\int_0^T \bar{x}(t)\cdot \mathrm{e}^{-j2\pi ft}\mathrm{d}t \qquad (4)$$

Da die DFT auf einem Rechner implementiert werden soll, muß man zur Berechnung dieses Integralausdrucks auf ein Näherungsverfahren ausweichen. Aus der Analysis ist folgendes einfaches Verfahren bekannt: man nähert das Integral durch seine Untersumme, d.h. die Integrantenfunktion wird in äquidistante Stützstellen aufgeteilt, und die Flächen der dadurch entstehenden Rechtecke werden aufsummiert. Dieses Vorgehen ist auch bei komplexwertigen Funktion ($\bar{x}(t)e^{-j2\pi ft}$ ist komplexwertig) erlaubt, nur kann man hier nicht mehr von Flächen und Rechtecken im geometrischen Sinn sprechen.

Die Anzahl der äquidistanten Stützstellen sei *N* , wodurch sich ihr Abstand zu $T = \frac{T_F}{N}$ ergibt.

Insgesamt erhält man für (4) den Ausdruck

$$\begin{aligned} X_m &= \frac{1}{T}\int_0^T \bar{x}(t)\cdot \mathrm{e}^{-j2\pi ft}\mathrm{d}t \\ &\approx \frac{1}{T}\cdot\sum_{n=0}^{N-1} T\cdot x(nT)\cdot \mathrm{e}^{-jm2\pi fnT} \qquad (5) \\ &= \sum_{n=0}^{N-1} x(nT)\cdot \mathrm{e}^{-jm2\pi fnT} \end{aligned}$$

Da T_F die Periodendauer ist, gilt: $f = \frac{1}{T_F}$, und somit auch $f = \frac{1}{N\cdot T}$

Damit wird die obige Gleichung zu:

$$X_m = \sum_{n=0}^{N-1} x(nT) \cdot \mathrm{e}^{-j2\pi\frac{mn}{N}} = \sum_{n=0}^{N-1} x_n \cdot \mathrm{e}^{-j2\pi\frac{mn}{N}} \quad \text{, mit } \; x_n := x(nT) \tag{6}$$

Um eine standardisierte Bezeichnung zu erhalten, werden die Folgen umbenannt (d.h. f_n statt x_n, und F_m statt X_m):

$$F_m = \sum_{n=0}^{N-1} f_n \cdot \mathrm{e}^{-j2\pi\frac{mn}{N}} \tag{7}$$

$$\dim F_m = \dim f_n$$

Die Bezeichnung F_m bezeichnet dabei jeweils die m-te Spektrallinie, mit $0 \le m \le N-1$.

Das erhaltene Spektrum ist also diskret, d.h. ein Linienspektrum. Es ist dabei zu beachten, daß sowohl f_n als auch F_m komplexwertig sein können!

Die Definitionsgleichung (7) wird als "Diskrete Fouriertransformierte (DFT) von f_n" bezeichnet (Zur Herleitung siehe zum Beispiel: [ENDE], [HESS], [SCHR], für die globale Einordnung kontinuierlicher und diskreter Verfahren auch [OPPE]).

In der Literatur, insbesondere der mathematischen, findet man vereinzelt auch die Schreibweise $F_m = \mathrm{F_N}\{f_n\}$ für die diskrete Fouriertransformierte einer Folge, wobei $\mathrm{F_N}$ dann "diskreter Fourieroperator" heißt. Allerdings ist diese Schreibweise nicht einheitlich, wie überhaupt in der Literatur die Definition von (5) teilweise stark voneinander abweicht. Eine verbreitete andere Definitionen ist die, daß die Gleichung (5) noch mit der Abtastzeit multipliziert wird, um die gleiche Dimension wie bei der kontinuierlichen Fouriertransformation zu erhalten. Dies ergibt prinzipiell keine Unterscheide zu der hier verwendeten Definition, allerdings sind gewisse Aussagen und Sätze in der Form etwas abweichend.

Beispiel:

Für eine 4-Punkte DFT (N=4) seien die folgenden Zahlenwerte gegeben:

$$f_n = (1,2,3,0)$$
$$f_0 = 1,\ f_1 = 2,\ f_2 = 3,\ f_3 = 0$$

Für diese Folge wird nun die DFT berechnet:

$$F_0 = f_0 \cdot e^{-j2\pi\frac{0\cdot 0}{4}} + f_1 \cdot e^{-j2\pi\frac{0\cdot 1}{4}} + f_0 \cdot e^{-j2\pi\frac{0\cdot 2}{4}} + f_0 \cdot e^{-j2\pi\frac{0\cdot 3}{4}}$$
$$= 1\cdot 1 + 2\cdot 1 + 3\cdot 1 + 0\cdot 1 = 6$$

$$F_1 = f_0 \cdot e^{-j2\pi\frac{1\cdot 0}{4}} + f_1 \cdot e^{-j2\pi\frac{1\cdot 1}{4}} + f_0 \cdot e^{-j2\pi\frac{1\cdot 2}{4}} + f_0 \cdot e^{-j2\pi\frac{1\cdot 3}{4}}$$
$$= 1\cdot 1 + 2\cdot(-j) + 3\cdot(-1) + 0\cdot j = -2 - j2$$

$$F_2 = f_0 \cdot e^{-j2\pi\frac{2\cdot 0}{4}} + f_1 \cdot e^{-j2\pi\frac{2\cdot 1}{4}} + f_0 \cdot e^{-j2\pi\frac{2\cdot 2}{4}} + f_0 \cdot e^{-j2\pi\frac{2\cdot 3}{4}}$$
$$= 1\cdot 1 + 2\cdot(-1) + 3\cdot 1 + 0\cdot(-1) = 2$$

$$F_3 = f_0 \cdot e^{-j2\pi\frac{3\cdot 0}{4}} + f_1 \cdot e^{-j2\pi\frac{3\cdot 1}{4}} + f_0 \cdot e^{-j2\pi\frac{3\cdot 2}{4}} + f_0 \cdot e^{-j2\pi\frac{3\cdot 3}{4}}$$
$$= 1\cdot 1 + 2\cdot j + 3\cdot(-1) + 0\cdot(-j) = -2 + j2$$

Also erhält man als Ergebnis:

$$F_m = \mathrm{F_N}\{f_n\} = (6, -2 - j2, 2, -2 + j2)$$

Bei dieser Rechnung zeigt sich ein erstaunlicher Effekt: die Exponenten innerhalb der Exponentialfunktion kommen mehrfach vor. Da die komplexe Exponentialfunktion außerdem periodisch (mit j2π) ist, sind auch Exponenten wie 6/4 und 2/4 identisch, d.h. sie ergeben den gleichen Wert. Diese Tatsache sollte man sich zunächst einmal im Hinterkopf behalten.

Zusammenfassende Darstellung über die diskreten Fourieroperatoren:

Diskreter Fourieroperator:
$$F_m = F_N\{f_n\} = \sum_{n=0}^{N-1} f_n \cdot e^{-j2\pi\frac{mn}{N}} \qquad (8)$$

Inverser Diskreter Fourieroperator:
$$f_n = F_N^{-1}\{F_m\} = \frac{1}{N} \cdot \sum_{m=0}^{N-1} F_m \cdot e^{j2\pi\frac{mn}{N}} \qquad (9)$$

Daß Gleichung (9) wirklich die Umkehrung von (8) ist, läßt sich leicht durch ineinander Einsetzen zeigen.

Für (9) gibt es noch eine alternative Rücktransformationsgleichung, die bei einer Rechnerimplementation oftmals von Vorteil ist. Sie lautet:

$$f_n = \frac{1}{N} \cdot \sum_{m=0}^{N-1} F_m \cdot e^{j2\pi\frac{mn}{N}} = \frac{1}{N} \cdot \left(\sum_{m=0}^{N-1} F_m \cdot e^{j2\pi\frac{mn}{N}} \right)^{**}$$
$$= \frac{1}{N} \cdot \left(\sum_{m=0}^{N-1} (F_m)^* \cdot e^{-j2\pi\frac{mn}{N}} \right)^* \qquad (10)$$

Der Vorteil in (10) ist der, daß für die eigentliche Transformation der gleiche Algorithmus verwendet werden kann. Nur die Eingangsfolge und die Ausgangsfolge müssen jeweils in ihr konjugiert Komplexes umgewandelt werden.

Wie man sieht, benötigt die Auswertung der Gleichungen (8) bis (10) Rechenoperationen, die aus einer Multiplikation und anschließender Addition bestehen (multiply-and-accumulate = MAC). Daher bietet es sich an, für die Auswertung der Rechnung einen Signalprozessor zu verwenden, der bekanntlich diese MAC-Operation sehr schnell durchführen kann.

Der Vollständigkeit halber sei noch erwähnt, daß die Darstellung nicht nur in komplexer Form möglich ist. Falls für eine Anwendung direkt Betrag und Phase von Interesse sind, so ist die übliche Umrechnung in der Form:

$$|F_m| = \sqrt{(\mathrm{Re}\, F_m)^2 + (\mathrm{Im}\, F_m)^2}$$

$$\arg F_m = \begin{cases} \frac{\pi}{2} & , \mathrm{Re}\, F_m = 0 \wedge \mathrm{Im}\, F_m > 0 \\ -\frac{\pi}{2} & , \mathrm{Re}\, F_m = 0 \wedge \mathrm{Im}\, F_m < 0 \\ \arctan \frac{\mathrm{Im}\, F_m}{\mathrm{Re}\, F_m} & , \mathrm{Re}\, F_m > 0 \\ \pi + \arctan \frac{\mathrm{Im}\, F_m}{\mathrm{Re}\, F_m} & , \mathrm{Re}\, F_m > 0 \end{cases} \tag{11}$$

jederzeit möglich.

Die diskrete Fouriertransformation und das Abtasttheorem

Bei der Herleitung der DFT-Transformationsgleichungen wird eine periodische Funktion im Zeitbereich unterstellt. Dies läßt sich jedoch mit Hilfe der Systemtheorie begründen. Tastet man eine Funktion im Zeitbereich ab, so entsteht nach Shannon eine periodische Funktion im Frequenzbereich. Entwickelt man eine periodische Funktion in eine Fourierreihe, so entstehen die diskreten Fourierkoeffizienten im Frequenzbereich.

Um dies noch einmal gegenüber zu stellen:

Abtastung im Zeitbereich (also Diskretisierung) => Periodizität im Frequenzbereich

Periodizität im Zeitbereich => Diskrete Koeffizienten im Frequenzbereich

Bei der diskreten Fouriertransformation hat man, wie bei jedem numerischen Verfahren, das Problem, daß man eine Funktion nur mit einer bestimmten Anzahl von Stützstellen beschreiben bzw. speichern kann. Daher liegen bei der DFT auch im Spektralbereich diskrete Koeffizienten vor, zugleich ist das Spektrum periodisch. Mit den obigen Sätzen der Systemtheorie folgt zwingend, daß dann die Zeitfunktion diskret (ist durch Abtastung erfüllt) und periodisch sein *muß*.

In allen zuvor verwendeten Gleichungen wird für den Index der Folge im Zeitbereich der Buchstabe *n*, für den Index der Folge im Frequenzbereich der Buchstabe *m* verwendet. Dies allein ist natürlich etwas unbefriedigend, da für eine physikalische Interpretation auch physikalische Größen benötigt werden.

Die Zuordnung des Index *n* zu einer bestimmten Zeit ist relativ naheliegend. Es gilt:

$$t_n = n \cdot T_{Ab} \quad ,0 \le n \le N-1 \qquad (12)$$

Für die Zuordnung im Frequenzbereich muß man zuerst kurz in die Theorie der Fourierreihen abschweifen. Für eine Funktion der Periode *T* weiß man, daß die Abstände der Spektrallinien den Wert $\Delta\omega = \frac{2\pi}{T}$ besitzen. Da bei der Herleitung der DFT-Transformationsgleichungen ein periodisches Zeitfenster der Periode T_F angenommen wurde, gilt somit:

$$\Delta\omega = \frac{2\pi}{T_F} = \frac{2\pi}{N \cdot T_{Ab}} \qquad (13)$$

Die Gleichung (13) wird auch als "spektrale Auflösung" der DFT bezeichnet, da sie angibt, mit welcher Genauigkeit die Darstellung im Frequenzbereich erfolgen kann.

Somit kann man analog zum Zeitbereich die folgende Gleichung aufstellen:

$$\omega_m = m \cdot \Delta\omega = m \cdot \frac{2\pi}{N \cdot T_{Ab}} = \frac{m}{N} \cdot 2\pi \cdot f_{Ab} \quad ,0 \le m \le N-1 \qquad (14)$$

bzw., wenn man sich nicht für die Kreisfrequenz, sondern für die Frequenz interessiert:

$$f_m = m \cdot \frac{\Delta\omega}{2\pi} = m \cdot \frac{1}{N \cdot T_{Ab}} = \frac{m}{N} \cdot f_{Ab} \quad ,0 \le m \le N-1 \qquad (15)$$

Da bei der DFT im Zeitbereich eine Abtastung vorliegt, muß für den Zeitausschnitt, der abgetastet wurde, selbstverständlich das Nyquist-Kriterium erfüllt sein, das besagt:

$$f_{\max} < \frac{1}{2} \cdot f_{Ab} \qquad (16)$$

Das Abtasttheorem besagt aber noch etwas: durch die Abtastung wiederholt sich das Spektrum periodisch mit der Abtastfrequenz (Aliasing-Effekt), und die Werte zwischen halber Abtastfrequenz und Abtastfrequenz sind eine Spiegelung der Wert zwischen 0 und der halben Abtastfrequenz.

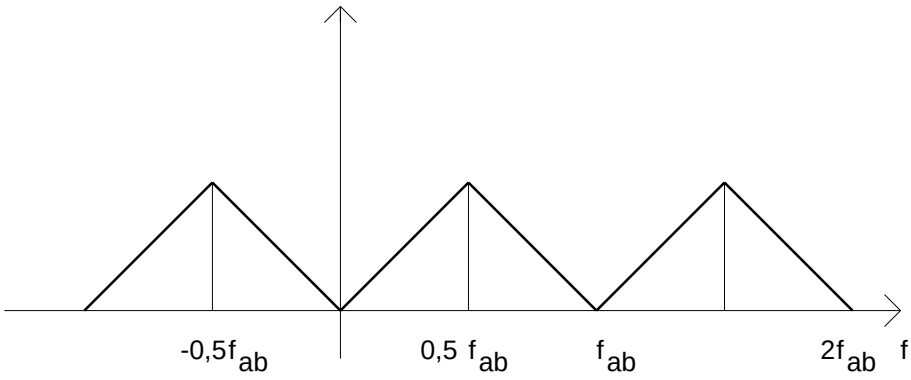


Bild 3: Periodisches Spektrum durch Abtastung im Zeitbereich

Betrachtet man nun aber Gleichung (15), so sieht man, daß für $m > \frac{N}{2}$ die Frequenzen oberhalb der halben Abtastfrequenz liegen. Was bedeutet dies für die DFT?

Wegen der Spiegelung an der halben Abtastfrequenz sind die DFT-Werte für $\frac{N}{2} + 1 \le m \le N - 1$ redundant, das heißt, bereits der halbe transformierte Wertesatz enthält die volle Information über das Signal im Zeitbereich. Wertet man also die transformierte Folge aus, oder stellt sie grafisch als Spektrum dar, so genügt es mit dem *N/2*-ten Wert aufzuhören. Man könnte nun natürlich auf die Idee kommen, generell auf diese redundanten Werte zu verzichten. Für die Rücktransformation aber werden alle *N* Werte benötigt, daher muß man diese redundanten Werte weiterhin berechnen und im Speicher halten, wird sich aber für Auswertungen auf den halben Datensatz beschränken.

Die Periodizität im Frequenzbereich hat auch eine Erweiterung der Gleichungen (14) und (15 zur Folge. Da sich die Frequenzwerte auch oberhalb der Abtastfrequenz wiederholen, kann die Einschränkung von *m* auf den Bereich *0..N-1* entfallen. Es gilt dann:

$$f_m = m \cdot \frac{\Delta\omega}{2\pi} = m \cdot \frac{1}{N \cdot T_{Ab}} = \frac{m}{N} \cdot f_{Ab} \qquad (16)$$

und das Spektrum besitzt an der allgemeinen Stelle *m* den Wert:

$$X\left(\frac{m}{N} \cdot f_{Ab}\right) = X_{(m \bmod N)} \qquad (16)$$

Abschließend noch einmal eine wichtige Warnung:
Alle Aussagen zur Transformation von Zeitfolgen gelten nur, wenn das Abtasttheorem tatsächlich erfüllt ist. Dies muß in der praktischen Anwendung unbedingt sichergestellt sein (Antialiasing-Filter!), da sich sonst gravierende Fehler einschleichen können. Denn der mathematische Algorithmus wird in jedem Fall ein Ergebnis liefern. Nur wenn das Nyquist-Kriterium erfüllt ist, stimmt das mathematische Ergebnis mit dem physikalischen Spektrum überein, sonst *nicht*.

Rechengesetze für den diskreten Fourieroperator

Allein die Definition der diskreten Fouriertransformation ist noch nicht ausreichend um festzustellen, welchen Wert diese Transformation hat. Denn für die Anwendbarkeit ist es wichtig zu wissen, welche der Sätze über die kontinuierliche Fouriertransformation auch im diskreten Fall erhalten bleiben. Denn nur dann, wenn ein Großteil der Rechengesetze aus dem zeitkontinuierlichen Fall auch im zeitdiskreten Fall erhalten bleibt, kann die DFT als rechnerbasierter Ersatz für die kontinuierliche Fouriertransformation angesehen werden.

Bevor jedoch auf diese Problematik eingegangen wird, soll am Anfang eine kurze Wiederholung für die Rechengesetze für Folgen stehen.

Exkurs: Rechnen mit (endlichen) Zahlenfolgen

Bei einer Zahlenfolge wird jeder natürlichen Zahl aus dem Definitionsbereich eine beliebige reelle oder komplexe Zahl zugeordnet. In der Regel geschieht dies über eine Berechnungsvorschrift, kann aber auch auf anderem Wege erfolgen.

Schreibweise: $f:\aleph \to \Re, \quad f_n := (a_n), a_n \in \Re$

Beispiele: Für Folgen, die durch Rechenvorschriften gegeben sind:

$$f_n := n \qquad f_n := \frac{n^2+1}{n}, n > 0$$

Für sprachliche Vorgaben:

$$f_n := \begin{cases} 0 & \text{, n ist eine gerade Zahl} \\ 1 & \text{, n ist eine ungerade Zahl} \end{cases}$$

Mit Folgen wird gerechnet, indem die Verknüpfungen auf jedes einzelne Element der Folge angewandt werden.

Addition: $f_n = a_n + b_n := (a_n) + (b_n)$

Multiplikation: $f_n = a_n \cdot b_n := (a_n) \cdot (b_n)$

Skalare Multiplikation: $f_n = \lambda \cdot a_n := \lambda \cdot (a_n) \quad , \lambda \in \mathrm{C}$

Faltung: $f_n = a_n * b_n := \sum_{r \in \aleph} a_r \cdot b_{n-r} = \sum_{r \in \aleph} a_{n-r} \cdot b_r = b_n * a_n$

Korrelation: $k(a_n, b_n) = \sum_{r \in \aleph} a_r \cdot b_{n+r}$

Beispiele: Sei $a_n = \frac{1}{n}, \; b_n = \frac{n^2}{n+1}$. Dann gilt:

$$f_n = a_n + b_n = \frac{1}{n} + \frac{n^2}{n+1} = \frac{1+n+n^3}{n(n+1)}$$

$$f_n = a_n \cdot b_n = \frac{1}{n} \cdot \frac{n^2}{n+1} = \frac{n}{n+1}$$

$$f_n = 3 \cdot a_n = \frac{3}{n}$$

$$f_n = (a_n * b_n) = (\frac{1}{n} * \frac{n^2}{n+1}) = \sum_{r \in \aleph} (\frac{1}{n-r} \cdot \frac{r^2}{r+1})$$

Wichtig ist insbesondere, daß Teile der Folge zu Null gesetzt werden können, die Rechengesetze (vor allem die Faltung) bleiben dann dennoch erhalten:

Beispiel: $f_n = \begin{cases} n^2 & , n < 24 \\ 0 & , n \geq 24 \end{cases}$ ist eine gültige Folge.

Dieser Typ Folgen, bei dem ab einem bestimmten Wert alle Folgewerte identisch 0 gesetzt werden, wird häufig auch als endliche Folge bezeichnet.

Nun geht es zurück zur eigentlichen Thematik: der DFT

Linearität:

Die Linearität ist eine der wichtigsten Eigenschaften, die eine Transformation erfüllen muß, da sonst keinerlei Parallelen beim Rechnen in Zeit- und Frequenzbereich bestehen. In anderen Worten: das Prinzip der linearen Superposition muß gelten.

$$\begin{aligned} \mathrm{F_N}\{ax_n + by_n\} &= \sum_{n=0}^{N-1} (ax_n + by_n) \mathrm{e}^{-j2\pi\frac{mn}{N}} \\ &= a\sum_{n=0}^{N-1} x_n \mathrm{e}^{-j2\pi\frac{mn}{N}} + b\sum_{n=0}^{N-1} y_n \mathrm{e}^{-j2\pi\frac{mn}{N}} \\ &= a\,\mathrm{F_N}\{x_n\} + b\,\mathrm{F_N}\{y_n\} \end{aligned}$$

Wegen der Linearität des Summenzeichens gilt dieses Gesetz wirklich, wodurch eine wesentliche Voraussetzung für die Verwendbarkeit geschaffen ist.

Verschiebungssätze:

In der kontinuierlichen Fouriertransformation sind auch die Verschiebungssätze für den Zeit- und Frequenzbereich wichtig, da hier aus der Verschiebung z.B. im Frequenzbereich Rückschlüsse auf die Veränderung des Zeitsignals möglich sind. Auch diese Verschiebungssätze sind erfüllt.

Verschiebung im Zeitbereich:

$$\begin{aligned} \mathrm{F_N}\{x_{n-\alpha}\} &= \sum_{n=0}^{N-1} x_{n-\alpha} \mathrm{e}^{-j2\pi\frac{mn}{N}} \\ &= \sum_{n=-\alpha}^{N-1-\alpha} x_n \mathrm{e}^{-j2\pi\frac{m}{N}(n+\alpha)} \\ &= \mathrm{e}^{-j2\pi\frac{m}{N}\alpha} \cdot \sum_{n=0}^{N-1} x_n \mathrm{e}^{-j2\pi\frac{mn}{N}} \\ &= \mathrm{e}^{-j2\pi\frac{m}{N}\alpha} \cdot \mathrm{F_N}\{x_n\} \end{aligned}$$

Verschiebung im Frequenzbereich:

$$\mathrm{F_N}\left\{\mathrm{e}^{j2\pi\frac{m\alpha}{N}} \cdot x_n\right\} = X_{m-\alpha}$$

$$\mathrm{F_N^{-1}}\{X_{m-\alpha}\} = \mathrm{e}^{j2\pi\frac{m\alpha}{N}} \cdot x_n$$

An dieser Stelle ist ein wichtiger Hinweis angebracht:

Die obigen Verschiebungssätze beziehen sich ja auf die Tatsache, daß die Zeitfunktion periodisch ist. Dies bedeutet aber, daß die Funktion zyklisch verschoben wird, was an dem einen Rand des Zeitfensters hinausgeschoben wird, erscheint am anderen Fensterrand wieder. Eine Grafik verdeutlicht diesen Zusammenhang.

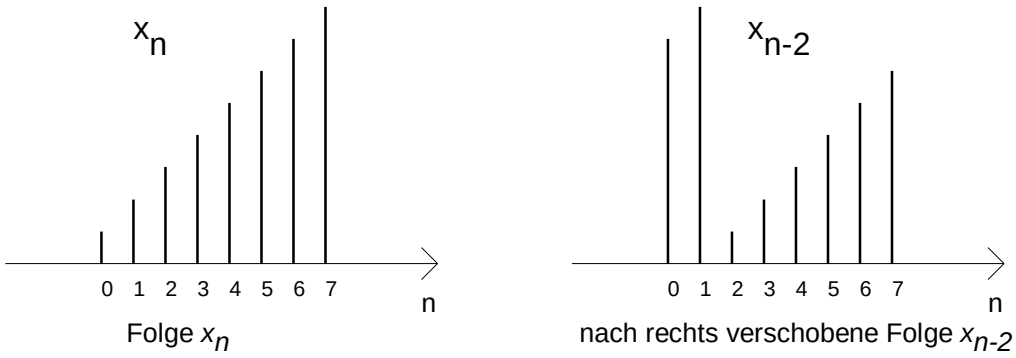


Bild 4: Zyklische Verschiebung von Folgen

Diese Tatsache zu verstehen fällt nicht schwer, da man mit periodischen Funktionen arbeitet. Häufig wird die zyklische Verschiebung dennoch falsch eingesetzt, da man von der Fourier- oder Laplacetransformation gewohnt ist, daß eine Zeitfunktion, die bei *t=0* beginnt, nach einer Rechtsverschiebung um t_0 eben erst bei t_0 beginnt. Bei der zyklischen Verschiebung gilt dies aber nicht - nach wie vor sind zum Zeitpunkt *t=0* Werte vorhanden.

Dem aufmerksamen Beobachter wird nicht entgangen sein, daß man die obige Rechtsverschiebung um 2 auch durch eine Linksverschiebung um 6 hätte darstellen können, also als x_{n+6}.

Differenzen-Satz:

Dieser Satz entsteht durch das Analogon zwischen der Differentiation im kontinuierlichen Fall und der Differenzenbildung im diskreten Fall. Für eine Ableitung existiert bekanntlich die Näherung:

$\frac{dx}{dt} \approx \frac{\Delta x}{\Delta t} = \frac{x_n - x_{n-1}}{\Delta t}$, womit die Entstehung dieses Satzes hinreichend erklärt sein dürfte. Die Verschiebung x_{n-1} ist wie zuvor als eine zyklische Verschiebung zu interpretieren.

$$
\begin{aligned}
F_N\{x_n - x_{n-1}\} &= \sum_{n=0}^{N-1}(x_n - x_{n-1})\mathrm{e}^{-j2\pi\frac{mn}{N}} \\
&= \sum_{n=0}^{N-1} x_n \mathrm{e}^{-j2\pi\frac{mn}{N}} - \sum_{n=0}^{N-1} x_{n-1}\mathrm{e}^{-j2\pi\frac{mn}{N}} \\
&= \sum_{n=0}^{N-1} x_n \mathrm{e}^{-j2\pi\frac{mn}{N}} - \sum_{n=-1}^{N-2} x_n \mathrm{e}^{-j2\pi\frac{m}{N}(n+1)} \\
&= (1-\mathrm{e}^{-j2\pi\frac{m}{N}}) \cdot \sum_{n=0}^{N-1} x_n \mathrm{e}^{-j2\pi\frac{mn}{N}} \\
&= (1-\mathrm{e}^{-j2\pi\frac{m}{N}}) \cdot F_N\{x_n\}
\end{aligned}
$$

Summationssatz:

Ein Summationssatz wäre das Analogon zum Integrationssatz bei der kontinuierlichen Fouriertransformation, allerdings gibt es bei der DFT keine sinnvolle Definition eines Summationssatzes. Zwar ist eine Summe über die Koeffizienten im Zeitbereich denkbar, es gibt allerdings keinen vernünftigen Zusammenhang zum Frequenzbereich.

Faltungssatz:

Um sich die Bedeutung des Faltungssatzes zu veranschaulichen, ein kurzer Rückblick auf die Systemtheorie.

Wird ein System mit einer Übertragungsfunktion $g(t)$ am Eingang durch eine Funktion $e(t)$ angeregt, so erscheint am Ausgang die Funktion $a(t) = e(t) * g(t)$. Da die Faltungsoperation, gerade bei der Verkettung von mehreren Übertragungsgliedern, nur noch sehr kompliziert ausgeführt werden kann, macht man den Schritt in den Frequenzbereich. Man ordnet der Übertragungsfunktion über die Fouriertransformation die Funktion $G(\omega)$, und der Eingangsgröße die Funktion $E(\omega)$ zu. Am Ausgang erscheint dann im Frequenzbereich die Antwort mit $A(\omega) = E(\omega) \cdot G(\omega)$, und dies ist durch die gewöhnliche Multiplikation leicht zu berechnen.

Die Erhaltung dieses Satzes bei der DFT ist ebenfalls unabdingbar, da hiermit erst eine rechnergestützte Systemanalyse möglich wird, und Systemantworten aus Zahlenfolgen berechenbar sind.

$$\begin{aligned}
\mathrm{F_N}\{x_n * y_n\} &= \mathrm{F_N}\left\{\sum_{r=0}^{N-1} x_r y_{n-r}\right\} = \sum_{n=0}^{N-1}\left(\sum_{r=0}^{N-1} x_r y_{n-r}\right)\cdot e^{-j2\pi\frac{mn}{N}} \\
&= \sum_{n=0}^{N-1}\left(\sum_{r=0}^{N-1} x_r y_{n-r}\cdot e^{-j2\pi\frac{mn}{N}}\right) = \sum_{r=0}^{N-1}\left(\sum_{n=0}^{N-1} x_r y_{n-r}\cdot e^{-j2\pi\frac{mn}{N}}\right) \\
&= \sum_{r=0}^{N-1} x_r\cdot e^{-j2\pi\frac{mr}{N}}\left(\sum_{n=0}^{N-1} y_{n-r}\cdot e^{-j2\pi\frac{m}{N}(n-r)}\right) \underbrace{=}_{\substack{\text{Periodizität von exp und } y_n}} \sum_{r=0}^{N-1} x_r\cdot e^{-j2\pi\frac{mr}{N}}\left(\sum_{n=0}^{N-1} y_n\cdot e^{-j2\pi\frac{mn}{N}}\right) \\
&= \mathrm{F_N}\{x_n\}\cdot\mathrm{F_N}\{y_n\}
\end{aligned}$$

Und umgekehrt gilt auch (Beweis verläuft analog):

$$\mathrm{F_N}\{x_n\cdot y_n\} = \frac{1}{N}\cdot\mathrm{F_N}\{x_n\} * \mathrm{F_N}\{y_n\}$$

Somit bleibt die Aussage, daß einer Faltung im Zeitbereich eine Multiplikation im Frequenzbereich entspricht, und umgekehrt, in der Tat richtig.

Somit kann eine Faltung im Zeitbereich mit folgenden Rechenvorschriften ausschließlich auf Multiplikationen von Folgen zurückgeführt werden (sogenannte schnelle Faltung):

$$x_n * y_n = \mathrm{F_N^{-1}}\{\mathrm{F_N}\{x_n\}\cdot\mathrm{F_N}\{y_n\}\}$$

Beispiel:

Zu diesem wichtigen Satz soll ein kurzes Beispiel die Richtigkeit der Aussagen verdeutlichen.

Es sind 2 Polynome gegeben: $p(x) := 1+2x \;, q(x) := 1-3x$

Diese 2 Polynome sind zu multiplizieren, und es ist das Produktpolynom zu ermitteln:

Lösungsmethode a:

Man multipliziert die beiden Polynome schriftlich aus und erhält:

$$\begin{aligned}
(pq)(x) &= p(x)\cdot q(x) = (1+2x)(1-3x) \\
&= 1-3x+2x-6x^2 \\
&= 1-x-6x^2
\end{aligned}$$

Nun gibt es in der Mathematik einen Satz, der in etwa besagt: 2 Polynome zu multiplizieren heißt, ihre Koeffizienten zu falten. Dieser Sachverhalt soll einmal unbewiesen akzeptiert werden. Hieraus ergibt sich die

Lösungsmethode b:

Die Faltung soll durch eine Multiplikation der transformierten Polynom-Koeffizienten erfolgen. Dazu schreibt man die Polynom-Koeffizienten als Folgen:

$p = (1,2,0,0) \quad , q = (1,-3,0,0)$

und wendet man DFT auf die beiden Folgen an, mit dem Ergebnis:

$P = (3,1-2j,-1,1+2j) \quad , Q = (-2,1+3j,4,1-3j)$

Multiplikation der transformierten Folgen:

$PQ = (-6,7+j,-4,7-j)$

Die Folge *PQ* wird nun mit der inversen DFT zurücktransformiert, und man erhält die Folge:

$pq = (1,-1,-6,0)$, bzw. in Polynomschreibweise $(pq)(x) = 1 - x - 6x^2$

Die Lösungsmethode b sieht auf den ersten Blick etwas wie Spielerei aus. Aber gerade bei Verschlüsselungsalgorithmen werden Polynommultiplikationen mit sehr hoher Ordnung der Polynome benötigt. Und dann ist das Verfahren b in der Tat schneller, vor allem wenn zur Durchführung der DFT der FFT-Algorithmus (dazu später mehr) eingesetzt wird ([LOCH]).

Korrelationsatz:

Die Korrelationsfunktion zweier Funktionen ist vor allem für die Meßtechnik von Bedeutung. Eine Vielzahl von Meßmethoden arbeitet mit dem Prinzip der Korrelation, daher liegt der Gedanke nahe, auch diesen Satz der kontinuierlichen Funktionen auf die zeitdiskreten Funktionen zu übertragen.

$$\begin{aligned}
\mathrm{F_N}\{k_n(x_n, y_n)\} &= \mathrm{F_N}\left\{\sum_{r=0}^{N-1} x_r y_{n+r}\right\} = \sum_{n=0}^{N-1}\left(\sum_{r=0}^{N-1} x_r y_{n+r}\right)\cdot e^{-j2\pi\frac{mn}{N}} \\
&= \sum_{n=0}^{N-1}\left(\sum_{r=0}^{N-1} x_r y_{n+r}\cdot e^{-j2\pi\frac{mn}{N}}\right) = \sum_{r=0}^{N-1}\left(\sum_{n=0}^{N-1} x_r y_{n+r}\cdot e^{-j2\pi\frac{mn}{N}}\right) \\
&= \sum_{r=0}^{N-1} x_r \cdot e^{j2\pi\frac{mr}{N}}\cdot\left(\sum_{n=0}^{N-1} y_{n+r}\cdot e^{-j2\pi\frac{m}{N}(n+r)}\right) = \left(\sum_{r=0}^{N-1} x_r \cdot e^{j2\pi\frac{mr}{N}}\right)^{**}\cdot\left(\sum_{n=0}^{N-1} y_{n+r}\cdot e^{-j2\pi\frac{m}{N}(n+r)}\right) \\
&= \left(\sum_{r=0}^{N-1}(x_r)^*\cdot e^{-j2\pi\frac{mr}{N}}\right)^*\cdot\left(\sum_{n=0}^{N-1} y_n\cdot e^{-j2\pi\frac{mn}{N}}\right) \\
&= \left(\mathrm{F_N}\left\{(x_n)^*\right\}\right)^*\cdot\mathrm{F_N}\{y_n\}
\end{aligned}$$

Falls die Zeitfolge x_n rein reell ist, vereinfacht sich der Zusammenhang noch etwas:

$$\mathrm{F_N}\{k_n(x_n, y_n)\} = \left(\mathrm{F_N}\{x_n\}\right)^*\cdot\mathrm{F_N}\{y_n\}$$

Parseval'sches Theorem:

Das Parseval'sche Theorem sagt anschaulich aus, daß die Energie eines Signals im Zeitbereich gleich seiner Energie im Frequenzbereich ist. Daß diese Aussage auch im Falle der diskreten Fouriertransformation gelten muß, ist klar: eine Energie kann sich nicht durch eine andere mathematische Darstellung plötzlich erhöhen oder vermindern.

Es gilt:

$$f_n = \frac{1}{N}\sum_{m=0}^{N-1} F_m \cdot e^{j2\pi\frac{mn}{N}} \Rightarrow (f_n)^* = \frac{1}{N}\sum_{m=0}^{N-1} (F_m)^* \cdot e^{-j2\pi\frac{mn}{N}}$$

Diese Aussage wird im Folgenden verwendet:

$$\begin{aligned}\sum_{n=0}^{N-1}|f_n|^2 &= \sum_{n=0}^{N-1} f_n \cdot (f_n)^* = \sum_{n=0}^{N-1} f_n \cdot \frac{1}{N}\sum_{m=0}^{N-1} F_m{}^* e^{-j2\pi\frac{mn}{N}} \\ &= \frac{1}{N}\cdot\sum_{m=0}^{N-1} F_m{}^* \cdot \sum_{n=0}^{N-1} f_n e^{-j2\pi\frac{mn}{N}} = \frac{1}{N}\cdot\sum_{m=0}^{N-1} F_m{}^* \cdot F_m \\ &= \frac{1}{N}\cdot\sum_{m=0}^{N-1}|F_m|^2\end{aligned}$$

Man hat also das Parseval'sche Theorem in der Form:

$$\sum_{n=0}^{N-1}|f_n|^2 = \frac{1}{N}\cdot\sum_{m=0}^{N-1}|F_m|^2$$

Symmetriesätze:

Aus der Theorie der Fouriertransformation und der Fourierreihen kennt man gewisse Symmetrien, die sich bei ungeraden oder geraden Funktionen im Frequenzbereich einstellen. Die beiden wesentlichen Symmetrien sind hierbei:

Zeitfunktion reell und gerade	⇒	Spektrum rein reell
Zeitfunktion reell und ungerade	⇒	Spektrum rein imaginär

Diese Symmetrien existieren natürlich auch bei der diskreten Fouriertransformation. Es gibt bei der Anwendung allerdings ein kleines Problem: es stellt sich nämlich die Frage, was eine gerade oder eine ungerade periodische Folge sein soll. Denn in der Mathematik sind diese Begriffe nicht definiert.

Daher wird folgende Vereinbarung getroffen:

Ein periodische Folge heiße ungerade (gerade), wenn ihre zugehörige Zeitfunktion, aus der sie entstanden ist, ebenfalls ungerade (gerade) ist.

Mit dieser Vereinbarung sind nun folgende Aussagen möglich:

Zeitfolge reell und gerade	⇒	diskretes Spektrum rein reell
Zeitfolge reell und ungerade	⇒	diskretes Spektrum rein imaginär

Dem aufmerksamen Leser ist wahrscheinlich aufgefallen, daß bisher für die Zeitfolgen keine Einschränkung auf reelle Folgen gemacht wurde. Dies hat den einfachen Grund, weil die DFT auch für komplexwertige Folgen definiert und anwendbar ist. In der Technik wird man aber fast immer

mit reellwertigen Folgen arbeiten. In diesem Fall vereinfachen sich einige Sätze etwas (z.B. Korrelation, Parseval'sches Theorem), sind dann allerdings nicht mehr allgemeingültig. Dem Autor war es aber wichtiger, eine vollständige Definition der DFT und ihrer Eigenschaften niederzulegen, und nicht die rechentechnischen Vereinfachungen ins Detail auszureizen.

Gegenüberstellung DFT und kontinuierliche Fouriertransformation

Kontinuierliche Fouriertransformation	Diskrete Fouriertransformation
Transformationsgleichung $F(\omega) := \mathcal{F}\{f(t)\} = \int_{-\infty}^{\infty} f(t) \cdot e^{-j\omega t} dt$	$\mathrm{F_N}\{f_n\} = \sum_{n=0}^{N-1} f_n \cdot e^{-j2\pi\frac{nm}{N}}$
Rücktransformationsgleichung $f(t) := \mathcal{F}^{-1}\{F(\omega)\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega) \cdot e^{j\omega t} d\omega$	$\mathrm{F_N}^{-1}\{F_m\} = \frac{1}{N} \cdot \sum_{m=0}^{N-1} F_m \cdot e^{j2\pi\frac{nm}{N}}$
Linearität $\mathcal{F}\{a \cdot f(t) + b \cdot g(t)\} = a \cdot F(\omega) + b \cdot G(\omega)$	$\mathrm{F_N}\{ax_n + by_n\} = a\,\mathrm{F_N}\{x_n\} + b\,\mathrm{F_N}\{y_n\}$
Verschiebung im Zeitbereich $\mathcal{F}\{f(t-a)\} = e^{-j\omega a} \cdot F(\omega)$	$\mathrm{F_N}\{x_{n-\alpha}\} = \mathrm{e}^{-j2\pi\frac{m}{N}\alpha} \cdot \mathrm{F_N}\{x_n\} \quad , \; a < N$
Verschiebung im Frequenzbereich $\mathcal{F}\{e^{jta} \cdot f(t)\} = F(\omega - a)$	$\mathrm{F_N}\left\{\mathrm{e}^{j2\pi\frac{m\alpha}{N}} \cdot x_n\right\} = X_{m-\alpha} \quad , \; a < N$
Differentiation im Zeitbereich $\mathcal{F}\left\{\frac{\mathrm{d}f(t)}{\mathrm{d}t}\right\} = j\omega \cdot F(\omega)$	$F_N\{x_n - x_{n-1}\} = (1 - \mathrm{e}^{-j2\pi\frac{m}{N}}) \cdot F_N\{x_n\}$
Integration im Zeitbereich $\mathcal{F}\left\{\int_{-\infty}^{t} f(\tau)\mathrm{d}\tau\right\} = \frac{1}{j\omega}F(\omega) + \pi \cdot S(0) \cdot \delta(\omega)$	keine sinnvolle Definition
Faltung im Zeitbereich $\mathcal{F}\{f(t) * g(t)\} = F(\omega) \cdot G(\omega)$	$\mathrm{F_N}\{x_n * y_n\} = \mathrm{F_N}\{x_n\} \cdot \mathrm{F_N}\{y_n\}$
Faltung im Frequenzbereich $\mathcal{F}\{f(t) \cdot g(t)\} = F(\omega) * G(\omega)$	$\mathrm{F_N}\{x_n \cdot y_n\} = \frac{1}{N} \cdot \mathrm{F_N}\{x_n\} * \mathrm{F_N}\{y_n\}$
Korrelation reeller Funktionen / Folgen $\mathcal{F}\{k_{f,g}\} = (F(\omega))^* \cdot G(\omega)$	$\mathrm{F_N}\{k_n(x_n, g_n)\} = (\mathrm{F_N}\{x_n\})^* \cdot \mathrm{F_N}\{y_n\}$
Parseval'sches Theorem $\int_{-\infty}^{\infty} \lvert f(t)\rvert^2 \mathrm{d}t = \int_{-\infty}^{\infty} \lvert F(\omega)\rvert^2 \mathrm{d}\omega$	$\sum_{n=0}^{N-1} \lvert f_n\rvert^2 = \frac{1}{N} \cdot \sum_{m=0}^{N-1} \lvert F_m\rvert^2$

(Für die obigen Sätze findet sich in der Literatur kaum eine einheitliche Darstellung. [ENDE] und [OPPE] stellen sehr viele Sätze inklusive Beweis vor, während [BRIG] ziemlich ausführlich auf die diskrete Faltung und Korrelation eingeht.)

Meßtechnische Probleme der DFT - die Wahl des Zeitfensters

Um die prinzipielle Problematik der meßtechnischen Anwendung der DFT aufzuzeigen, muß zu Beginn eine kurze Analyse der zu untersuchenden Signale stehen. Daß es überhaupt Probleme bei der Anwendung der DFT auf kontinuierliche Signale gibt, sollte leicht einzusehen sein. Schließlich wendet man ein numerisches Näherungsverfahren (also diskretes Verfahren) auf die (heile) analoge Welt an, und muß damit auch gewisse Fehler in Kauf nehmen.

Die im wesentlichen auftauchenden Signale lassen sich in 3 Gruppen einteilen:

a. periodische Signale mit einer solchen Frequenz, daß die Abtastfrequenz ein ganzzahliges Vielfaches der Signalfrequenz ist
b. ein beliebiges periodisches Signal oder periodisches Signalgemisch
c. ein transientes (d.h. nichtperiodisches) Signal

Ein Signal der Gruppe a. ist unproblematisch. Wendet man die DFT auf dieses Signal an, so erhält man in der Tat im Spektrum nur eine einzige Linie, deren Amplitude linear von der Amplitude des Signals abhängt. Allerdings ist dieser Fall eher akademischer Natur: in der Regel wendet man eine Frequenzanalyse ja auf Signale an, deren Frequenz unbekannt ist. Wie soll da die Forderung, daß die Abtastfrequenz ein Vielfaches der Signalfrequenz ist, überhaupt eingehalten werden?

Man kommt also sehr schnell zu dem Signaltyp b., einem beliebigen periodischen Signal. Und genau hier treten 2 Fehlerquellen der DFT auf.

Zum einen kann es vorkommen, daß ein Signal mit einer bestimmten Frequenz keine geeignet Linie im diskreten Spektrum hat. Zum Beispiel wurde die Auflösung so gewählt, daß bei 990Hz und 1010Hz jeweils eine diskrete Spektrallinie liegt. Analysiert man nun ein Signal mit einer Frequenz von 1000Hz, so gibt es zwangsläufig ein Problem: an welchen Stellen soll jetzt die Signalenergie im Frequenzbereich dargestellt werden? Führt man diese Rechnung durch, so wird man feststellen, daß sowohl bei 990Hz, als auch bei 1010Hz jeweils eine Spektrallinie erscheinen wird. Es ist daher offensichtlich, daß eine Spektrallinie nach Anwendung der DFT nicht automatisch bedeutet, daß ein Signal mit exakt dieser Frequenz im Zeitbereich vorliegt. Vielmehr ist es so, daß in dem Frequenzbereich zwischen 2 Spektrallinien eine bestimmte Signalenergie vorliegt, die sich in der Höhe der Spektrallinie darstellt.

Die folgenden Bilder verdeutlichen diese Auswirkungen.

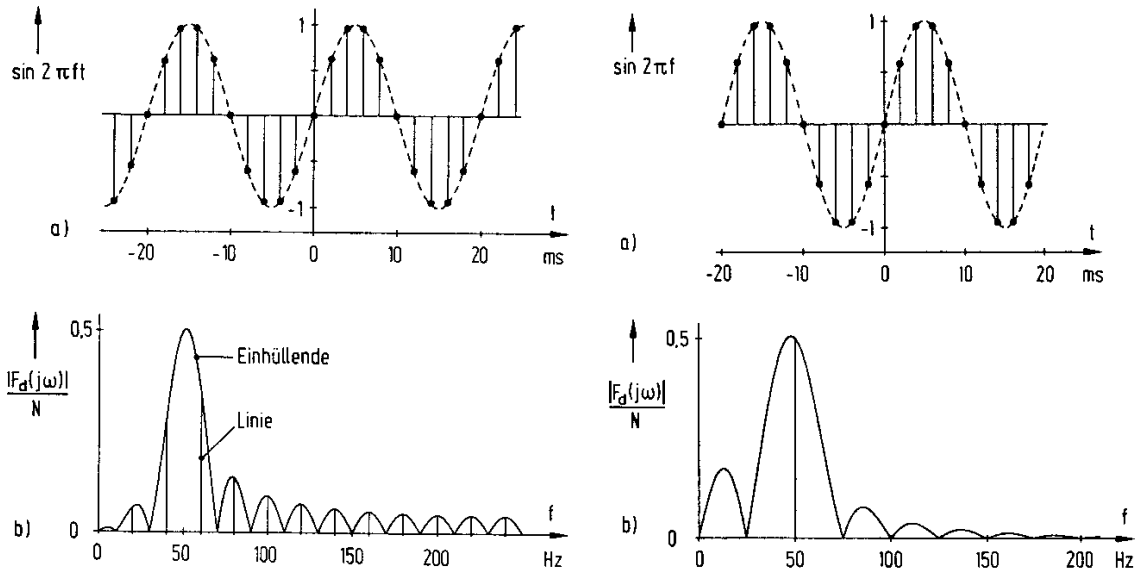


Bild 5: Ist die Abtastfrequenz kein Vielfaches der Signalfrequenz, so entstehen zusätzliche Linien im Spektrum (rechts); a) Zeitsignal, b) Spektrum mit Einhüllender

Der andere Fehler, der entsteht, ist wesentlich subtiler. Die DFT unterstellt bei ihrer mathematischen Herleitung ein periodisch fortgesetztes Zeitsignal. Es gibt aber durchaus einen Unterschied, wie das Signal periodisch fortgesetzt wird, das folgende Bild zeigt diesen Unterschied.

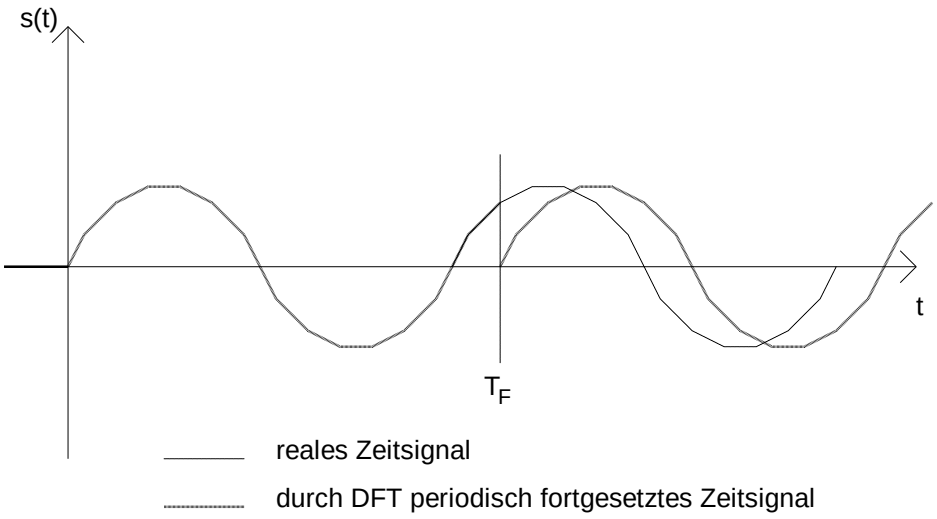


Bild 6: Periodische Fortsetzung durch DFT

Man sieht deutlich, daß zur Zeit T_F eine Sprungstelle auftaucht, die im physikalischen Signal noch nicht vorhanden war. Aus der Theorie der Fourier-Reihen weiß man aber auch, daß Sprungstellen zu Frequenzanteilen führen. Daher erzeugt die mathematische Berechnung der DFT des physikalischen Signals Frequenzanteile an Stellen, an denen eigentlich keine Frequenzanteile vorhanden sind. Dieser Effekt ist auch als Leck-Effekt (leakage) bekannt.

Der erste Lösungansatz, die Abtastfrequenz so zu wählen, daß die mathematische Periode mit der physikalischen Periode zusammenfällt, scheidet aus zuvor genannten Gründen aus. Daher muß man sich mit der Ursache der Sprünge befassen, und versuchen, die Sprunghöhe zu verkleinern.

Ein Sprung an der periodischen Fortsetzung wird im wesentlichen dadurch verursacht, daß die Abtastung schlagartig beginnt und schlagartig endet. Mathematisch ausgedrückt, multipliziert man die abzutastende Zeitfunktion mit einem Rechteckfenster, und wertet den entstehenden Ausdruck aus:

Sei *s(t)* die zu untersuchende Zeitfunktion. Als unterlegtes Zeitfenster verwendet man ein Rechteck, daß wie folgt definiert ist:

$$w(t) = \begin{cases} 1 & ,0 \le t < nT_{Ab} \\ 0 & ,sonst \end{cases}$$

Der Abtastvorgang, also quasi das Einschalten des AD-Umsetzers und das Ausschalten, läßt sich durch die Fensterfunktion *w(t)* beschreiben.

Die Fensterfunktion wird mit dem Eingangssignal multipliziert, und das entstehende Signal wird der DFT zugeführt:

$$s_0(t) = s(t) \cdot w(t) = \begin{cases} s(t) & ,0 \le t \le nT_{Ab} \\ 0 & ,sonst \end{cases}$$

Die Funktion *w(t)* weist an ihren Rändern Unstetigkeitstellen auf. Daher hatte man die Idee, Fensterfunktionen zu verwenden, die an den Rändern stetig gegen Null gehen, so daß die periodische Fortsetzung nun keine Sprungstellen mehr besitzt.

Eine Fensterfunktion wird wie folgt angewendet:

Sei *w(t)* eine Fensterfunktion, für die gilt:

$$w(t) = \begin{cases} f(t) & ,0 \le t < nT_{Ab} \\ 0 & ,sonst \end{cases}$$

Außerdem sei f_n die zeitdiskrete Wertefolge. Die "gefensterte" Ausgangsfolge berechnet sich dann zu:

$$g_n = f_n \cdot w(n \cdot T_{Ab}) \quad ,n = 0,...,N-1 \qquad (17)$$

Weiter unten sind einige gängige Zeitfensterfunktionen, sowie ihre grafische Darstellung, zusammengestellt.

Nun ist aber mit der alleinigen Anwendung eines Zeitfenster das Problem noch nicht vollständig abgehandelt, denn ein Zeitfenster beeinflußt selbstverständlich auch die Spektraldarstellung des Signals.

Die Multiplikation des Signals im Zeitbereich mit einer Fensterfunktion stellt im Frequenzbereich eine Faltung dar, d.h. das Spektrum des Signals wird mit dem Spektrum des Fenster gefaltet. Aus Zeit- und Platzgründen kann leider nicht ausführlich auf die konkrete Auswirkung eines Zeitfensters auf das Ausgangsspektrum eingegangen werden. Im wesentlichen sollte der Leser folgende Aussage behalten: im ursprünglichen Spektrum werden die physikalisch vorhandenen "Frequenzspitzen" etwas in die Breite gezogen, dafür aber die durch die periodische Fortsetzung entstandenen Frequenzen bedämpft.

Ausführlichere Darstellungen der Auswirkungen von Zeitfenstern finden sich in [SCHR], [BRIG] oder [ENDE], aber für ein umfassendes Verständnis ist es am besten, verschiedene Spektren mit Zeitfensterung zu berechnen und sich die Unterschiede zu betrachten.[#]

Im folgenden soll noch kurz ein Überblick über die Bewertungskriterien und die Definitionen häufig verwendeter Fensterfunktionen gegeben werden.

Das folgende Bild zeigt die Definition der Bewertungskriterien, aber auch das Spektrum eines Zeitfensters. Denn für alle Zeitfenster sind die Spektren von der Form her ähnlich.

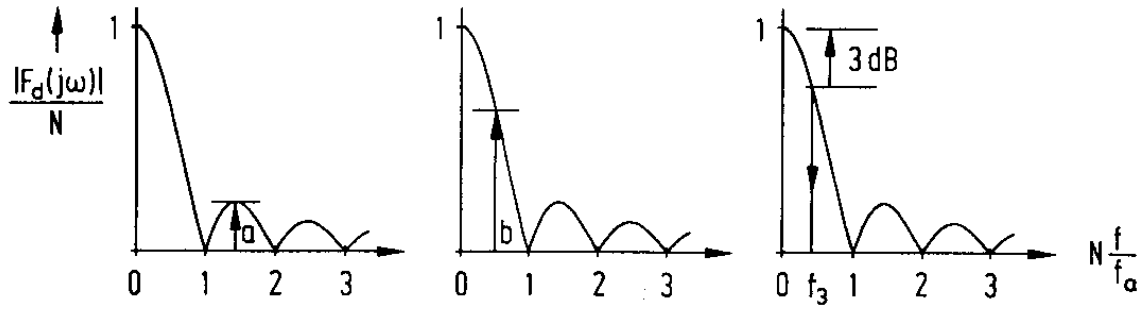


Bild 7: Kriterien zur Beurteilung eines Fensters

Verhältnis aus der Amplitude des höchsten Nebenzipfels und der Amplitude des Hauptzipfels

Die Fouriertransformation einer Fensterfunktion liefert bei $\omega=0$ die maximale Amplitude des Hauptzipfels. Die Amplituden der Nebenzipfel sind geringer. Das Verhältnis

$$a = \frac{\text{Amplitude des höchsten Nebenzipfels}}{\text{Amplitude des Hauptzipfels}}$$

wird für den Vergleich verschiedener Fensterfunktionen benutzt.

Maximale Abtastfehler

Die Spektrallinien einer abgetasteten Funktion fallen nicht notwendigerweise mit den Nullstellen der Fouriertransformierten eines Fensters zusammen. Die Spektrallinien des Fensters liegen im Abstand $\Delta\omega$. Die Amplitude des Hauptzipfels bei $w=0$ ist größer als die Amplitude bei der Frequenz $\omega = \frac{\Delta\omega}{2}$. Das Verhältnis

$$b = \frac{\text{Amplitude der Fenster - Fouriertransformierten bei } \omega = \frac{1}{2}\Delta\omega}{\text{Amplitude der Fenster - Fouriertransformierten bei } \omega = 0}$$

gibt an, umwieviel eine Amplitude höchstens falsch gemessen wird. Es wird als maximaler Abtastfehler bezeichnet.

[#] Dies ist übrigens mit dem ausgegebenen Demoprogramm in PASCAL möglich.

Breite des Hauptzipfels

Fensterfunktionen, bei denen die Nebenzipfel niedrig bleiben, haben einen besonders breiten Hauptzipfel. Dies ist ungünstig und führt zu einem Auseinanderlaufen der Spektrallinien. Zur Charakterisierung des Hauptzipfels dient die 3-dB-Grenzfrequenz. Dies ist die Frequenz, bei der die Amplitude des Hauptzipfels um 3dB abgefallen ist. Das Verhältnis

$$\frac{\text{Amplitude bei } \omega = 0}{\text{Amplitude bei } f_{-3dB}} \mathrel{\hat{=}} 3dB$$

definiert die Frequenz f_{-3dB}. Sie ist ein Maß für die Breite des Hauptzipfels. (aus [SCHR]).

Zusammenstellung wichtiger Fensterfunktionen und deren Daten

Die Fensterfunktionen sind bereits für den diskreten Fall angegeben, d.h. als Funktion von *n*, und nicht als Funktionen der Zeit *t*. Dies erleichtert die Rechnerumsetzung. Für Werte außerhalb des Bereichs *0,...,N-1* sind alle Fensterfunktionen identisch 0.

Rechteckfenster:

Zeitfunktion: $w_0(n) = 1 \ , n = 0,...,N-1$

Kontinuierliches Betragsspektrum: $|W_0(\omega)| = T_F |si(\pi f T_F)|$

Höchster Nebenzipfel: $a = -13dB \mathrel{\hat{=}} 0{,}224$

maximaler Abtastfehler: $b = 0{,}64$

Breite des Hauptzipfels: $c = 0{,}45 \cdot \Delta f$

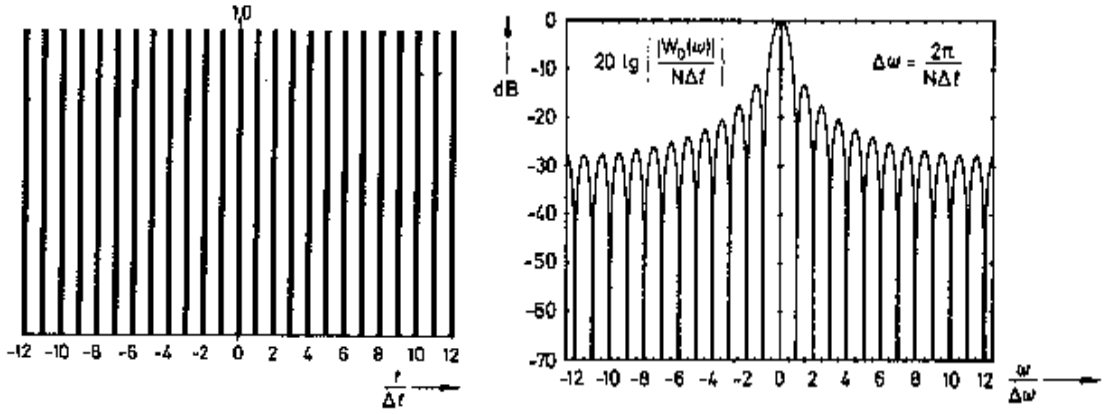


Bild 8: Abtastwerte und Spektralfunktion des Rechteckfensters

Dreieckfenster:

(auch Bartlett-Fenster genannt)

Zeitfunktion: $$w_1(n) = \begin{cases} \dfrac{n}{N/2} & ,n = 0,\ldots,\dfrac{N}{2} \\ w_1(N-n) & ,n = \dfrac{N}{2},\ldots,N-1 \end{cases}$$

Kontinuierliches Betragsspektrum: $$|W_1(\omega)| = \frac{T_F}{2}\left(si(\frac{\pi f T_F}{2})\right)^2$$

Höchster Nebenzipfel: $a = -27dB \mathrel{\hat{=}} 0{,}045$
maximaler Abtastfehler: $b = 0{,}81$
Breite des Hauptzipfels: $c = 0{,}64 \cdot \Delta f$

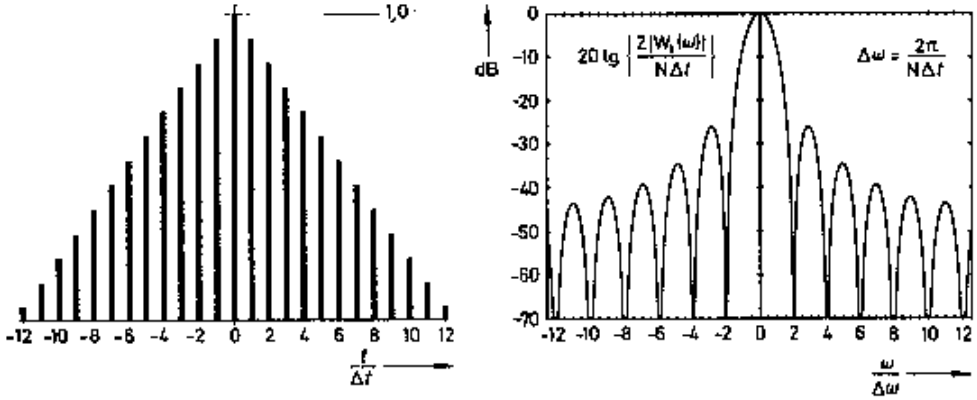


Bild 9: Abtastwerte und Spektralfunktion des Dreieckfensters

Von-Hann-Fenster:

(auch Hann-Fenster oder Hanning-Fenster genannt)

Zeitfunktion: $$w_2(n) = 0{,}5 - 0{,}5 \cdot \cos\frac{2\pi n}{N} \quad ,n = 0,\ldots,N-1$$

Kontinuierliches Betragsspektrum:

$$|W_2(\omega)| = \left|0{,}5 \cdot W_0(\omega) + 0{,}25 \cdot \left(W_0\left(\omega + \frac{2\pi}{T_F}\right) + W_0\left(\omega - \frac{2\pi}{T_F}\right)\right)\right|$$

Höchster Nebenzipfel: $a = -32dB \mathrel{\hat{=}} 0{,}025$
maximaler Abtastfehler: $b = 0{,}85$

Breite des Hauptzipfels: $c = 0{,}72 \cdot \Delta f$

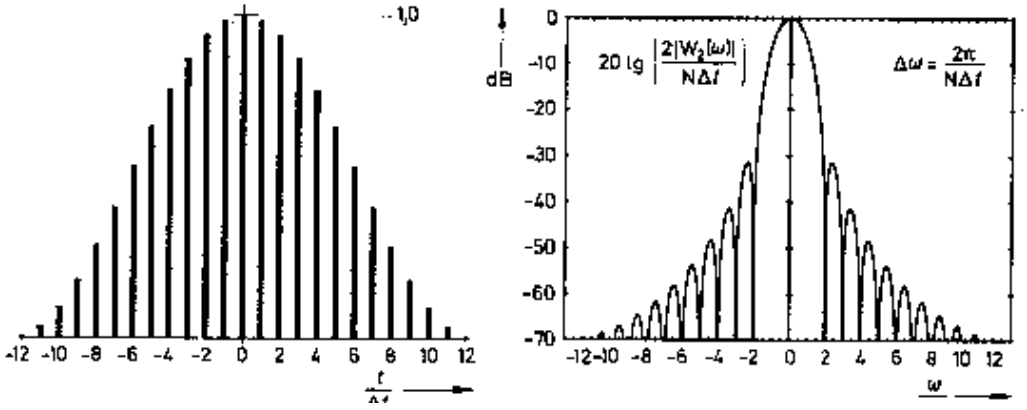


Bild 10: Abtastwerte und Spektralfunktion des Hann-Fensters

Hamming-Fenster:

Zeitfunktion: $w_3(n) = 0{,}54 - 0{,}46 \cdot \cos\frac{2\pi n}{N} \quad , n = 0,...,N-1$

Kontinuierliches Betragsspektrum:

$$|W_3(\omega)| = \left| 0{,}54 \cdot W_0(\omega) + 0{,}23 \cdot \left(W_0\left(\omega + \frac{2\pi}{T_F}\right) + W_0\left(\omega - \frac{2\pi}{T_F}\right)\right)\right|$$

Höchster Nebenzipfel: $a = -43dB \mathrel{\hat{=}} 0{,}007$
maximaler Abtastfehler: $b = 0{,}82$
Breite des Hauptzipfels: $c = 0{,}65 \cdot \Delta f$

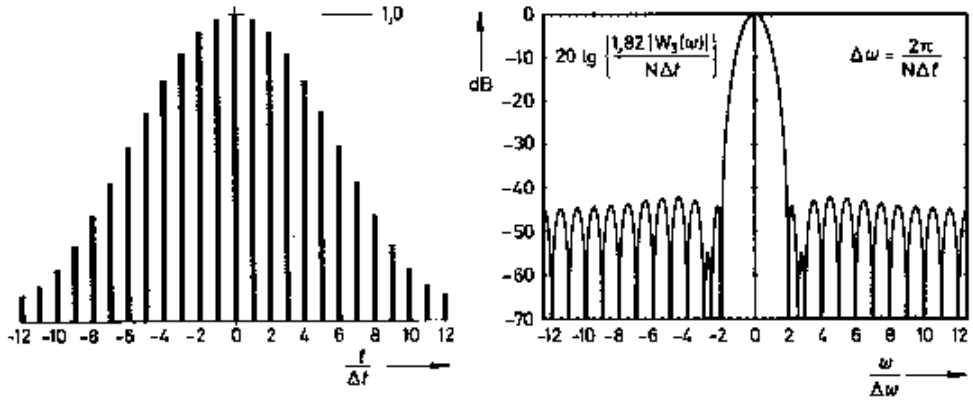


Bild 11: Abtastwerte und Spektralfunktion des Hamming-Fensters

Blackman-Fenster:

Zeitfunktion: $$w_4(n) = a_0 - a_1 \cdot \cos\frac{2\pi n}{N} + a_2 \cdot \cos\frac{4\pi n}{N} \quad , n = 0,\ldots,N-1$$

Kontinuierliches Betragsspektrum:

$$|W_4(\omega)| = \left| a_0 \cdot W_0(\omega) + \sum_{\nu=1}^{2} (-1)^{\nu} a_{\nu} \cdot \left(W_0\left(\omega + \frac{2\pi\nu}{T_F}\right) + W_0\left(\omega - \frac{2\pi\nu}{T_F}\right)\right)\right|$$

Mit den Koeffizienten:

$$a_0 = \frac{3969}{9304} \approx 0{,}42 \;, a_1 = \frac{1155}{4652} \approx 0{,}25 \;, a_2 = \frac{715}{18608} \approx 0{,}04$$

(Im allgemeinen werden zur praktischen Berechnung die Näherungswerte verwendet, und nicht die exakten Werte der Brüche)

Höchster Nebenzipfel: $a = -58dB \mathrel{\hat{=}} 0{,}001$

maximaler Abtastfehler: $b = 0{,}88$

Breite des Hauptzipfels: $c = 0{,}84 \cdot \Delta f$

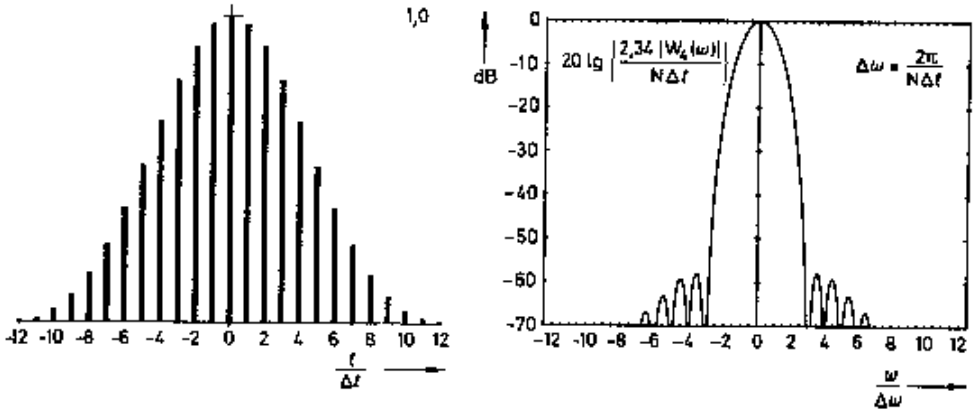


Bild 12: Abtastwerte und Spektralfunktion des Blackman-Fensters

(Die Übersicht wurde im wesentlichen aus [HESS] und [SCHR] zusammengestellt).

Nun fehlt (immer noch) der 3. Typ der Funktionen, nämlich die Gruppe der transienten Signale. Allerdings läßt sich die wesentliche Erkenntnis kurz zusammenfassen:

Zur Analyse transienter Signale und nur innerhalb der Meßdauer, d.h. innerhalb der für die Signalerfassung notwendigen Zeit definierter Signalvorgänge, dürfen mit Ausnahme des Rechteckfensters keine Fensterfunktionen verwendet werden.

Eine Verwendung einer Fensterfunktion würde hier auch keinen Sinn machen: da keine Funktionswerte außerhalb der Beobachtungszeit existieren, und es daher auch keine Randeffekte an den Analysegrenzen geben kann, wird der Vorgang grob verfälscht und wahrscheinlich unbrauchbar.

Daraus ergibt sich die zwingende Konsequenz, daß man die Analyse transienter Signale mit einer Dauer, die größer ist als die Erfassungszeit, mit der DFT vermeiden sollte, da die dabei entstehenden Fehler nicht mehr beseitigt werden können.

Das folgende Beispiel zeigt noch einmal einen transienten Vorgang, der fälschlicherweise mit einem Dreieckfenster gewichtet wurde [SCHR].

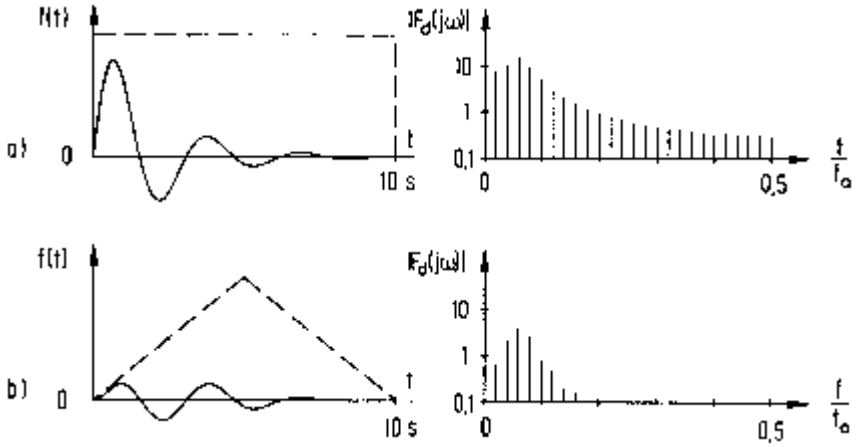


Bild 13: a) Originalfunktion b) gewichtet (und verfälscht) mit Dreieckfenster

Die Problematik dieses Abschnitts wird in der Literatur überwiegend ausführlich dargestellt, was wohl daran liegt, daß es sich um meßtechnische, und weniger um mathematische Probleme handelt. Dazu: [ELPR], [ENDE], [HESS], [SCHR].

Übersicht über Signaltransformationen

Den meisten Lesern werden die Begriffe Fouriertransformation, Laplacetransformation und z-Transformation geläufig sein. Im Folgenden soll gezeigt werden, daß die diskrete Fouriertransformation nicht etwa eine Untergruppe einer bestehenden Transformation ist, sondern als vierte Transformationsart das fehlende Glied der obigen Transformationen darstellt.

Ausgehend von der Laplace-Transformation gelangt man zur Fouriertransformation, indem man den Realteil des komplexen Operators *s* zu *0* setzt. In der Regelungstechnik wird die so entstehende Übertragungsfunktion auch als Frequenzgang bezeichnet. Während also die Laplace-Transformation das System allgemein beschreibt, gibt der Frequenzgang, also die Fouriertransformierte, Auskunft über das Verhalten des Systems für bestimmte Frequenzen.

Aus der Laplace-Transformation einer Zahlenfolge entsteht bei entsprechender Umbenennung die z-Transformation, die wie folgt definiert ist:

$$F(z) = \sum_{k=0}^{\infty} f_k z^{-k}$$

In der z-Transformation ist der Operator *z* die Abkürzung für $z = e^{sT}$. Macht man wieder den Übergang zum Frequenzgang, indem man $s = \underbrace{\sigma}_{0} + j\omega$ setzt, so ergibt sich: $z = e^{j\omega T}$. Und damit erhält man (fast) schon die Definition der diskreten Fouriertransformation, man muß aber noch die endliche Summation über eine Periode einführen:

$$F_N(f_k) = \sum_{k=0}^{\infty} f_k e^{-j\omega T}$$

Man sieht also, so wie die Fouriertransformation als Frequenzgang zur Laplacetransformation gehört, so läßt sich ebenfalls die diskrete Fouriertransformation aus der diskreten Laplacetransformation, = z-Transformation, erhalten.

Und damit wird auch die jeweilige Domäne von z-Transformation und DFT deutlich. Denn grundsätzlich läßt sich ein digitaler Filter sowohl mit der z-Transformation, als auch mit der DFT realisieren. Die z-Transformation geht dabei aber vom Entwurf dergestalt aus, daß der Entwurf nur einen Momentanwert (und die kurzfristige Vergangenheit) betrachtet. Ein digitaler Filter mit der DFT würde ganz anders aufgebaut: man tastet für einen festen Zeitraum eine Zeitfolge ab, manipuliert im Frequenzbereich alle Frequenzen simultan und führt anschließend eine inverse DFT aus, um wieder ein Zeitsignal zu erhalten.

Aus der umständlichen Vorgehensweise und des größeren Rechenaufwands läßt sich bereits ableiten, daß man in der Regel (es gibt Ausnahmen) die DFT nicht zum Entwurf von Echtzeitfiltern verwendet, sondern in diesem Fall den Entwurf mit Hilfe der z-Transformation durchführt. Aber immer dann, wenn man einen ganzen Abtastsatz vorliegen hat, der ausgewertet oder gefiltert werden soll, und genügend Zeit vorhanden ist, verwendet man die DFT ([ENDE], [OPPE]).

Die schnelle Diskrete-Fouriertransformation

Der wahrscheinlich bekannteste Begriff im Zusammenhang mit der Rechneranwendung von Fouriertransformationen ist die sogenannte "FFT", oder ausführlich Fast Fourier Transform, zu deutsch also schnelle Fouriertransformation. Dieser Begriff ist allerdings etwas unscharf gewählt, denn es wird eigentlich keine Fouriertransformation, sondern eine diskrete Fouriertransformation durchgeführt. Der FFT-Algorithmus ist keine eigenständige Transformation, vielmehr liefert er exakt das gleiche Ergebnis wie die DFT-Gleichungen, die zuvor besprochen wurden. Wenn also vom FFT-Algorithmus als ein "Näherungsverfahren" gesprochen wird, so bedeutet dies: die DFT ist eine Näherung zur kontinuierlichen Fouriertransformation, also ist auch der FFT-Algorithmus eine Näherung zur kontinuierlichen Fouriertransformation - dennoch ist das Ergebnis von FFT und DFT identisch, also keine weitere Näherung mehr.

Wie man am Anfang bei der beispielhaften Berechnung einer 4-Punkte-DFT sehen konnte, waren zur Berechnung $4^2=16$ komplexe Multiplikation notwendig. Dies gilt auch allgemein: eine N-Punkte DFT benötigt $O(n^2)$-komplexe Multiplikationen.

In der Informatik gibt es die große Gruppe der Divide-And-Conquer-Algorithmen (kurz DAC-Algorithmen). Bekannte Vertreter dieser Algorithmen sind u.A. auch die Sortierverfahren "Quicksort" und "Heapsort". Die Idee der DAC-Algorithmen soll am Beispiel von Heapsort kurz erläutert werden:

Zu sortieren seien 2^n-Zahlen (vereinfachende Annahme) in aufsteigender Reihenfolge. Man zerlegt nun die Zahlen in 2 gleich große Listen, und beginnt für beide Listen den Algorithmus von vorne, also mit 2 zu sortierenden Listen à 2^{n-1} Zahlen. Dies führt schließlich zu n Listen mit nur noch 2 Zahlen. Eine Sortierung von 2 Zahlen ist mit einem einzigen Vergleich erledigt. Also werden nun alle 2-elementigen Listen durch einen Vergleich der Größe nach sortiert. Nun werden immer 2 Listen wieder zu einer doppelt so großen sortierten Liste zusammengefügt. Schlußendlich erhält man eine Liste mit 2^n Zahlen in sortierter Reihenfolge.

Dies erklärt auch den Namen Divide-And-Conquer: Teilen und Erobern, man teilt also das Problem in klein "Häppchen", und die Hauptaufgabe läßt sich auf einem solch kleinen "Häppchen" trivial erledigen. Danach fügt man alles wieder zusammen, und das Problem ist gelöst. Kennzeichnend für die DAC-Algorithmen ist ihre Konvergenzklasse, also die Anzahl benötigter Operationen: diese ist $O(n \log n)$, also wesentlich unter einem Algorithmus mit quadratischer Laufzeit wie der DFT-Berechnung (Tieferen Einblick in die Thematik der DAC-Algorithmen gibt [GÜTI]).

Eine Schlüsselrolle bei der Ableitung des FFT-Algorithmus spielt einerseits die Linearität der DFT, andererseites die Periodizität der komplexen Exponentialfunktion. Dies zusammen ermöglicht die Zerlegung eines Abtastsatzes in kleinere Partialsätze, auf die ihrerseits die DFT angewendet werden kann. Das korrekte Ergebnis wird durch eine phasenrichtige Addition der transformierten Partialsätze gewonnen.

Am einfachsten sind die Verhältnisse, wenn man sich auf Abtastsätze mit Blocklängen beschränkt, die Potenzen von 2 sind, also wenn $N=2^k$. Dies hat zur Folge, daß man den Abtastsatz fortlaufend in 2 kleinere Blöcke zerlegen kann. Das Konzept basiert auf einem 1965 von *Cooley* und *Tukey* vorgestellten Verfahren, das als das klassische FFT-Verfahren anzusehen ist.

Zur Vereinfachung der Herleitung wird, wie in der Literatur üblich, folgende Abkürzung eingeführt:

$$w_N = e^{-j2\pi\frac{1}{N}}$$

Dieser Ausdruck wird als Drehfaktor (engl. twiddle factor) oder auch als Einheitswurzel bezeichnet.

Aus der Definition des Drehfaktors folgen mit der Periodizität der komplexen Exponentialfunktion 2 wichtige Eigenschaften:

$$w_N^k = w_N^{k \bmod N}$$

$$w_N^k = -w_N^{N/2+k}$$

D.h. es genügt für beliebiges k immer, nur den 0-ten bis ((N/2)-1)-ten Drehfaktor zu bestimmen, die anderen Exponenten lassen sich auf diese Basisdrehfaktoren zurückführen.

Die Definition der diskreten Fouriertransformation läßt sich unter Verwendung des Drehfaktors wie folgt umschreiben:

$$\begin{aligned}
F_m &= \sum_{n=0}^{N-1} f_n \cdot e^{-j2\pi\frac{mn}{N}} \\
&= \sum_{\substack{n=0,\\ gerade}}^{N-1} f_n \cdot e^{-j2\pi\frac{mn}{N}} + \sum_{\substack{n=0,\\ ungerade}}^{N-1} f_n \cdot e^{-j2\pi\frac{mn}{N}} \\
&= \sum_{n=0}^{\frac{N}{2}-1} f_{2n} \cdot e^{-j2\pi\frac{m(2n)}{N}} + \sum_{n=0}^{\frac{N}{2}-1} f_{2n+1} \cdot e^{-j2\pi\frac{m(2n+1)}{N}} \\
&= \sum_{n=0}^{\frac{N}{2}-1} f_{2n} \cdot e^{-j2\pi\frac{mn}{\frac{N}{2}}} + e^{-j2\pi\frac{m}{N}} \cdot \sum_{n=0}^{\frac{N}{2}-1} f_{2n+1} \cdot e^{-j2\pi\frac{mn}{\frac{N}{2}}} \\
&= \sum_{n=0}^{\frac{N}{2}-1} f_{2n} \cdot w_{N/2}^{mn} + w_N^m \cdot \sum_{n=0}^{\frac{N}{2}-1} f_{2n+1} \cdot w_{N/2}^{mn}
\end{aligned}$$

Also ist eine DFT der Länge *N* auch zu berechnen, indem man 2 DFTs der halben Länge berechnet, und anschließend (phasenrichtig) addiert. Dies setzt allerdings voraus, daß die Anzahl *N* geradzahlig ist. Die gezeigte Halbierung läßt sich sooft durchführen, bis am Ende nur noch DFTs der Länge 2 zu berechnen sind. Dies setzt aber voraus, daß die Anzahl der Werte immer eine Potenz von 2 ist, also:

$$N = 2^k \quad , k \in \aleph$$

Eine solche Minimal-DFT mit nur noch 2 Werten wird grafisch in der Regel als sogenannter "Schmetterling" oder "Butterfly" dargestellt. Ein FFT-Butterfly hat folgenden Signalflußplan:

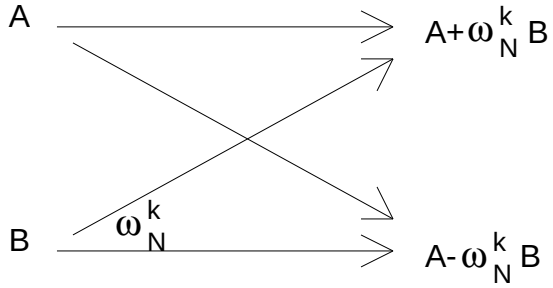


Bild 14: Basis-Butterfly-Operation

Noch eines ist bei der Zerlegung in Teilfolgen wichtig: wie zuvor gesehen, werden bei der Aufteilung in der einen DFT die geraden, in der anderen DFT die ungeraden Werte zur Berechnung herangezogen. Wendet man nun die Aufteilung sooft an, bis man am Ende nur noch Minimal-DFTs erhält, sieht man leicht ein, daß die Reihenfolge, in der man die zwei Werte zusammensetzt, nicht mehr der Größe nach sortiert ist. Es läßt sich in der Tat zeigen, daß die Zerlegung in Teilfolgen für die Indizes eine Bitumkehr darstellt.

An einem Beispiel sei dies verdeutlicht: der Index 6 läßt sich binär als (z.B. *N=8*) 110 darstellen. Nach einer Bitumkehr wird der Index zu einer 011, also dezimal 3. Konkret bedeutet das, daß der Wert, der in der Ausgangsfolge an der Stelle 6 steht, für die Berechnung an die Stelle 3 geschrieben werden muß.

Die Bitumkehr, engl. bit (reverse) shuffling, ist bei Darstellung in binärer Schreibweise die Spiegelung der Zahl in der Zahlmitte. Die folgende Tabelle zeigt die Bitumkehr für alle Indizes einer 8-Werte-FFT:

Index vor Bitumkehr	Binäre Darstellung	Spiegelung	Index nach Bitumkehr
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

Bild 15: Bitumkehroperation für 3 Bit

Das folgende Bild zeigt ein Signalflußgraphen für eine 8-Werte-FFT, in der die Abfolge der Butterflies und die Wahl der Drehfaktoren deutlich wird.

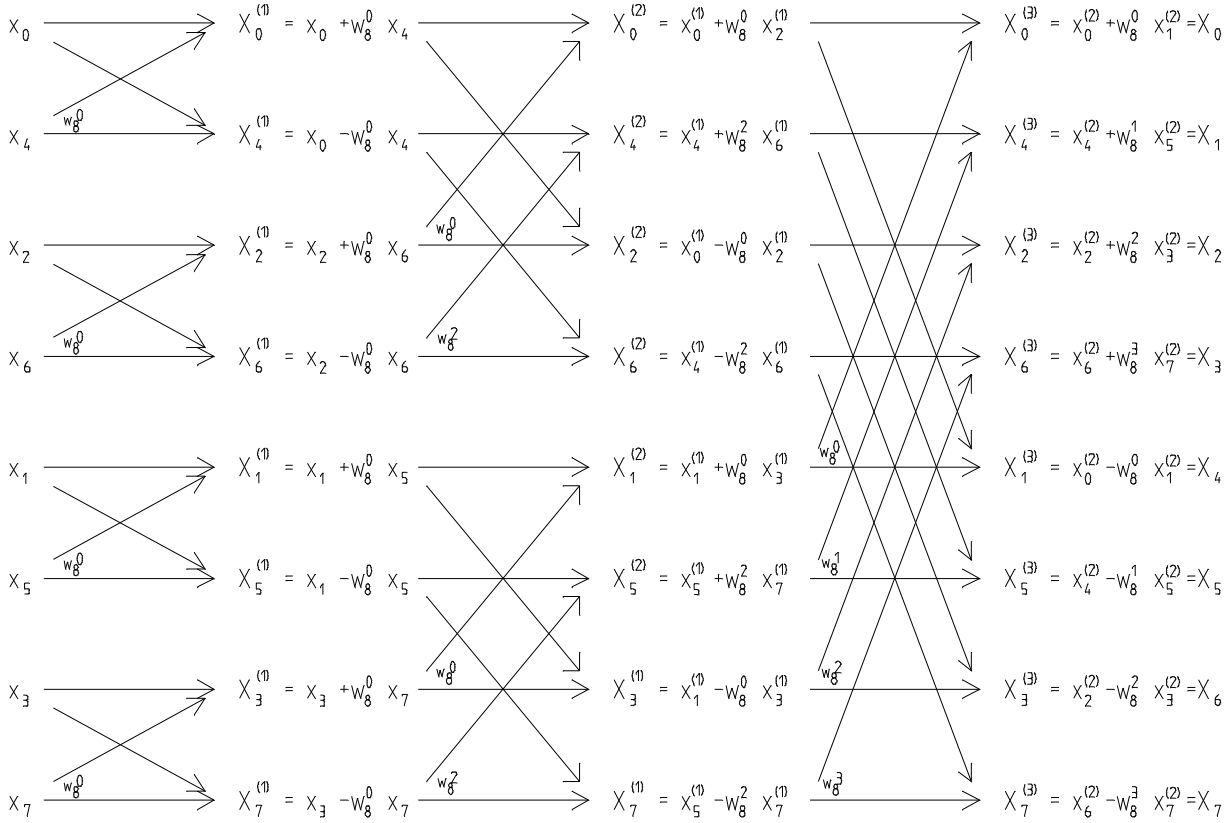


Bild 16: Signalflußplan für eine 8-Punkte FFT

Zur Überprüfung des Berechnungsschemas soll für einen bestimmten Wert nun einmal der Signalflußgraph durchlaufen werden, so daß man die Identität zwischen DFT und FFT sehen kann.

$$\begin{aligned} X_7 &= x_6^{(2)} - w_8^3 \cdot x_7^{(2)} \\ &= x_4^{(1)} - w_8^2 \cdot x_6^{(1)} - w_8^3 \cdot \left(x_5^{(1)} - w_8^2 \cdot x_7^{(1)} \right) \\ &= x_0 - w_8^0 \cdot x_4 - w_8^2 \cdot \left(x_2 - w_8^0 x_6 \right) - w_8^3 \cdot \left(x_1 - w_8^0 \cdot x_5 - w_8^2 \cdot \left(x_3 - w_8^0 \cdot x_7 \right) \right) \\ &= x_0 - w_8^3 \cdot x_1 - w_8^2 \cdot x_2 + w_8^5 \cdot x_3 - w_8^0 \cdot x_4 + w_8^3 \cdot x_5 + w_8^2 \cdot x_6 - w_8^5 \cdot x_7 \end{aligned}$$

Wird der Wert mit Hilfe der DFT-Definitionsgleichungen transformiert, so erhält man:

$$\begin{aligned} X_7 &= w_8^0 \cdot x_0 + w_8^7 \cdot x_1 + w_8^{14} \cdot x_2 + w_8^{21} \cdot x_3 + w_8^{28} \cdot x_4 + w_8^{35} \cdot x_5 + w_8^{42} \cdot x_6 + w_8^{49} \cdot x_7 \\ &= x_0 - w_8^3 \cdot x_1 - w_8^2 \cdot x_2 + w_8^5 \cdot x_3 - w_8^0 \cdot x_4 + w_8^3 \cdot x_5 + w_8^2 \cdot x_6 - w_8^5 \cdot x \end{aligned}$$

Die Übereinstimmung der beiden letzten Zeilen läßt sich mit Hilfe einer Darstellung der Drehzeiger in der komplexen Ebene leicht nachvollziehen:

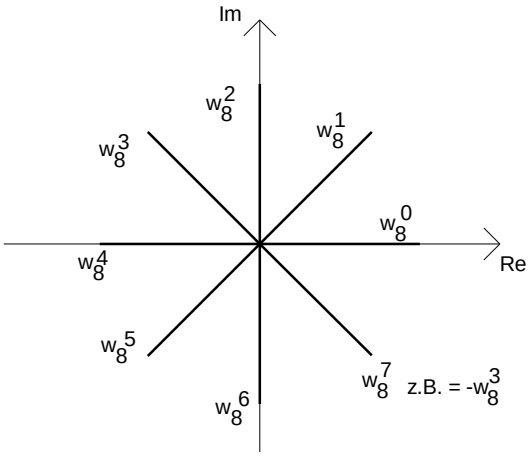


Bild 17: Alle möglichen Drehzeiger für *N=8* in der komplexen Ebene

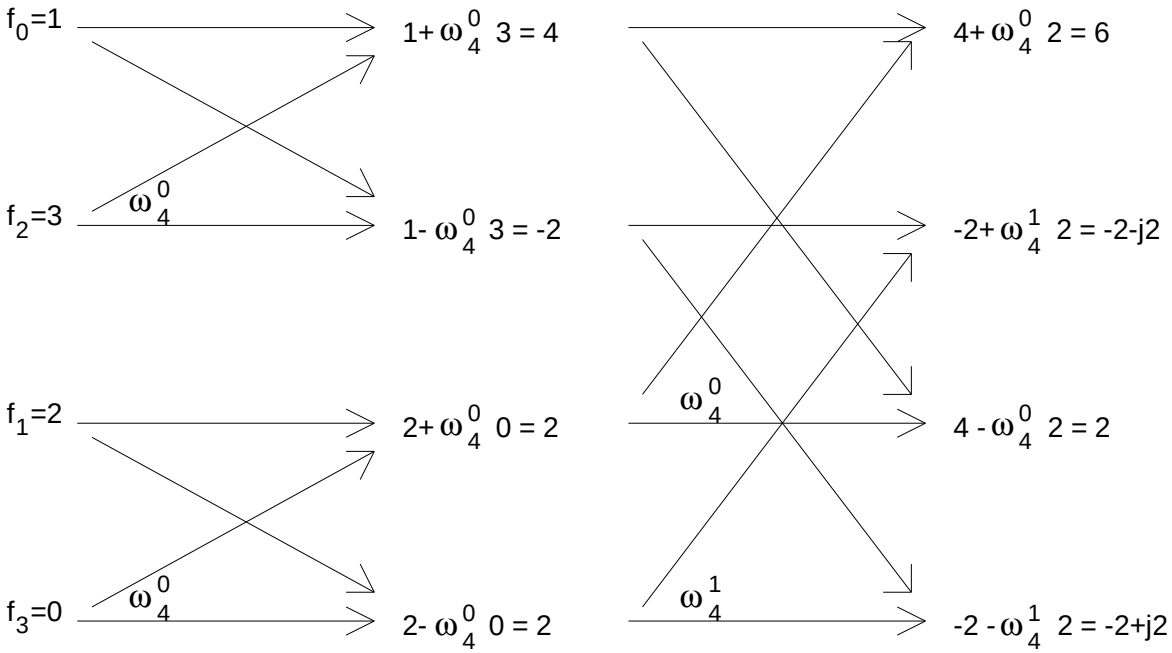
Natürlich kann diese Rechnung nur eine Plausibiltätskontrolle darstellen, reicht aber für die Anschauung aus, daß FFT und DFT identische Ergebnisse liefern.

Beispiel:

Für die bereits am Anfang betrachtete Zahlenfolge $f = (1,2,3,0)$ soll mit Hilfe des FFT-Algorithmus die diskrete Fouriertransformierte berechnet werden.

Zuerst erfolgt die Bitumkehr:
$0=(00)_2 \Rightarrow (00)_2=0$, $1=(01)_2 \Rightarrow (10)_2=2$, $2=(10)_2 \Rightarrow (01)_2=1$, $3=(11)_2 \Rightarrow (11)_2=3$

Danach erfogt die Berechnung mit dem Butterfly-Schema:



In der Tat ist das Ergebnis mit $F = (6,-2-j2,2,-2+j2)$ das gleiche wie zuvor mit der DFT berechnet wurde, aber es wurden wesentlich weniger komplexe Multiplikation benötigt.

Der folgende Programmausschnitt in PASCAL zeigt nun eine Implementation eines allgemeinen FFT-Algorithmus, der auch die Möglichkeit der inversen Transformation beeinhaltet. Für die inverse Transformation wird von der Tatsache Gebrauch gemacht, daß die iFFT als FFT der konjugiert komplexen Folge dargestellt werden kann (siehe unter "Herleitung der Transformationsgleichungen").

Der hier dargestellte Algorithmus basiert auf dem "radix-2-decimation in-place FFT"-Algorithmus, da er mit einer Basis-FFT mit 2 Werten beginnt (daher radix-2-decimation ≈ Verminderung auf 2er-Wurzeln), und die Umsortierung in-place vornimmt, d.h. führt die FFT-Berechnung mit einem einzigen komplexen Array durch. Das Ergebnis steht im gleichen Array wie zuvor die Eingabeparameter.

Der eingefleischte Programmierer erkennt beim oben gezeigten Verfahren sicherlich noch eine weitere Optimierungsmöglichkeit: die erste DFT-Stufe mit 2 Werten ist immer eine reine Addition, da der Exponent 0 ist, und der Drehfaktor hoch 0 ergibt 1. Somit läßt sich eine gesamte Multiplikationsstufe einsparen. Diese Änderung wurde im Programm ebenfalls bereits durchgeführt.

Zusätzlich greift der Algorithmus für die trigonometrischen Funktionen auf hinterlegte Sinus- und Cosinustabellen zu, so daß die Fließkommaberechnung immer nur für eine Tabelle erstellt werden muß. Hierfür wird allerdings relativ viel Speicherplatz benötigt (Stichwort "Trading Place for Time"), falls dieser nicht zur Verfügung steht, berechnet das Programm die Werte wieder automatisch mit den eingebauten Sinus- und Cosinusfunktionen.

```
TYPE
   float = single;
   tFolge = ARRAY [0..8191] OF float;
   tFolgePtr = ^tFolge;
   tFFTMode = (FM_Transformation,FM_InverseTransformation);

   CONST
      FFT  = FM_Transformation;
      iFFT = FM_InverseTransformation;


   FUNCTION BitReverse(inValue : WORD; inBitCnt : BYTE) : WORD;
      (* algorithm:
         FOR i := 1 TO inBitCnt DO
            Logical Shift Right inValue;
            carry := inValue MOD 2;
            inValue := inValue DIV 2;
            Logical Shift Left dest;
            dest := dest * 2 + carry;
         OD;
         BitReverse := dest;
         *)
      BEGIN
         ASM
               MOV   AX,inValue     (* inValue nach AX *)
               MOV   BL,inBitCnt    (* inBitCnt nach BL *)
               MOV   CX,0           (* outResult := 0 *)
            @BitLoop:
               RCR   AX,1           (* inValue DIV 2, CY=inValue.bit0 *)
               RCL   CX,1           (* CY nach outResult.bit0 *)
               SUB   BL,1           (* --inBitCnt *)
```

```
            JNZ   @BitLoop       (* until (inBitCnt=0) *)
            MOV   @Result,CX     (* MathBitReverse := CX *)
         END (* ASM *);
      END (* BitReverse *);


   PROCEDURE FFT(inPunkte : WORD;
                 inoutDatenReal, inoutDatenImag : POINTER;
                 inModus : tFFTMode);
      CONST
         cLN_ZWEI = 0.693147181;  (* Ln(2) *)
      VAR
         dataRe,
         dataIm      : tFolgePtr;
         fRe,
         fIm,
         fcos,
         fsin,
         freal,
         fimag,
         fArg         : float;
         wSpruenge,
         wSprung,
         wBflyDist,
         wSprunghoehe,
         wStufe,
         wElement1,
         wElement2,
         wExponent,
         wBflies       : WORD;
         bBits         : BYTE;
         wPos,
         wRevPos       : WORD;

      BEGIN
         dataRe := inoutDatenReal;
         dataIm := inoutDatenImag;
         ErstelleTabellen(inPunkte);
         bBits := ROUND( Ln(inPunkte) / cLN_ZWEI );

         (* Bitreverse-Shuffling *)
         FOR wPos := 0 TO (inPunkte - 1) DO BEGIN
            wRevPos := Bitreverse(wPos,bBits);
            IF wRevPos > wPos THEN BEGIN
               fRe := dataRe^[wPos];
               fIm := dataIm^[wPos];
               dataRe^[wPos] := dataRe^[wRevPos];
               dataIm^[wPos] := dataIm^[wRevPos];
               dataRe^[wRevPos] := fRe;
               dataIm^[wRevPos] := fIm;
            END (* IF *);
         END (* FOR *);

         (* Für den Fall der inversen FFT hier das konjugiert
            Komplexe der Folge bilden *)
         IF inModus = FM_InverseTransformation THEN
            FOR wElement1 := 0 TO (inPunkte - 1) DO
               dataIm^[wElement1] := - dataIm^[wElement1];
            (* END FOR *)
         (* END FOR *)

         (* Erste Stufe ist eine reine Addition zweier benachbarter
```

```
   Elemente, und kann daher vorgezogen werden *)
wElement1 := 0;
wElement2 := 1;
WHILE (wElement1 < inPunkte) DO BEGIN
   freal := dataRe^[wElement2];
   fimag := dataIm^[wElement2];
   fRe := dataRe^[wElement1];
   fIm := dataIm^[wElement1];
   (* Berechnung des Butterfly *)
   dataRe^[wElement2] := fRe - freal;
   dataIm^[wElement2] := fIm - fImag;
   dataRe^[wElement1] := fRe + freal;
   dataIm^[wElement1] := fIm + fimag;
   INC(wElement1,2);
   INC(wElement2,2);
END (* FOR *);

(* hier setzt der "gewöhnliche" FFT-Algorithmus ein *)
wSpruenge := inPunkte DIV 4;         (* 2 *)
wBflyDist := 2;                     (* 1 *)
wSprunghoehe := 4;                  (* 2 *)
FOR wStufe := 2 TO bBits DO BEGIN
   wElement1 := 0;
   wElement2 := wBflyDist;
   FOR wSprung := wSpruenge DOWNTO 1 DO BEGIN
        wExponent := 0;
      FOR wBflies := 1 TO wBflyDist DO BEGIN
         fcos := ptCos(wExponent);
         fsin := ptSin(wExponent);
         freal := dataRe^[wElement2];
         fimag := dataIm^[wElement2];
         (* Multiplikation des 2. Elementes mit
            dem Drehfaktor *)
         fRe := freal * fcos + fimag * fsin;
         fIm := fimag * fcos - freal * fsin;
         freal := dataRe^[wElement1];
         fimag := dataIm^[wElement1];
         (* Berechnung des Butterfly *)
         dataRe^[wElement2] := freal - fRe;
         dataIm^[wElement2] := fimag - fIm;
         dataRe^[wElement1] := freal + fRe;
         dataIm^[wElement1] := fimag + fIm;
         INC(wElement1);
         INC(wElement2);
         wExponent := wExponent + wSpruenge;
      END (* FOR *);
      wElement1 := wElement1 - wBflyDist + wSprunghoehe;
      wElement2 := wElement1 + wBflyDist;
   END (* FOR *);
   wSpruenge := wSpruenge DIV 2;
   wSprunghoehe := wSprunghoehe * 2;
   wBflyDist := wBflyDist * 2;
  END (* FOR *);

(* Für den Fall der inversen FFT hier wieder das konjugiert
   Komplexe der Folge bilden, und Real- und Imaginärteil
   durch die Anzahl der Punkte dividieren *)
IF inModus = FM_InverseTransformation THEN BEGIN
   fRe := - inPunkte;    (* Konversion WORD -> float *)
   fIm := inPunkte;      (* ist Schleifeninvariant *)
   FOR wElement1 := 0 TO (inPunkte - 1) DO BEGIN
      dataIm^[wElement1] := dataIm^[wElement1] / fRe;
```

```
            dataRe^[wElement1] := dataRe^[wElement1] / fIm;
          END (* FOR *);
        END  (* IF *);

    END (* FFT *);
```

Für die direkte Berechnung jeder dieser kleineren DFT's benötigt man eine Anzahl von Operationen der Größenordnung $(N/2)^2$, das macht zusammen $2(N/2)^2$. Zusätzlich sind noch $N/2$ Operationen für die Multiplikation mit dem Phasendrehfaktor notwendig. Für große N ist dies jedoch vernachlässigbar, so daß für jede Berechnungsstufe die Anzahl notwendiger komplexer Multiplikationen in der Größenordnung $O(2(N/2)^2)$ liegt. Da die Gesamtzahl der Punkte eine Potenz von 2 ist, läßt sich die Teilung genau $log_2\ N=w$-mal durchführen. Insgesamt besitzt daher der Algorithmus eine Komplexität der Klasse $O(N\ log_2\ N)$, ist also wesentlich weniger aufwendig als der DFT-Algorithmus mit der Komplexität $O(N^2)$. Die folgende Grafik zeigt eine Gegenüberstellung der Anzahl komplexer Multiplikation für die beiden Verfahren.

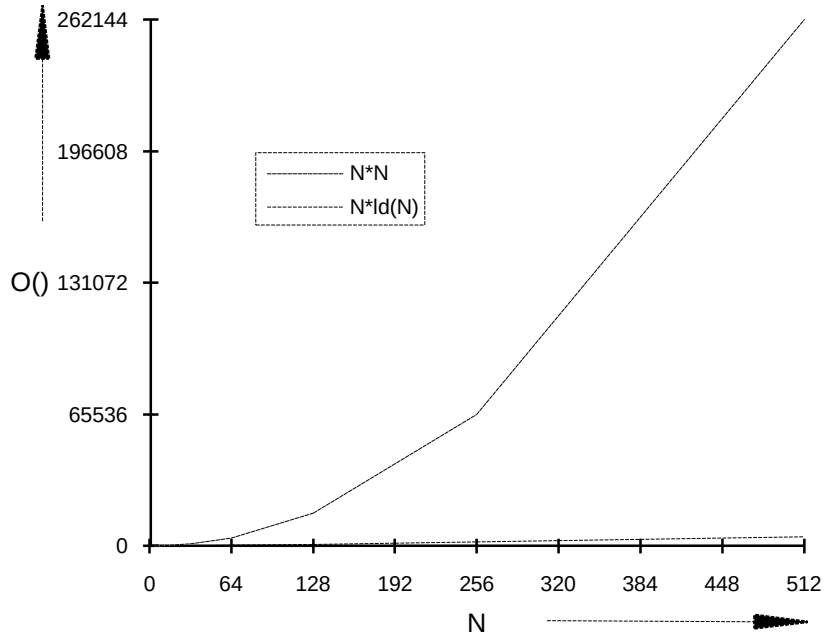


Bild 18: Komplexitätsvergleich DFT - FFT

Natürlich wird die Geschwindigkeitseinsparung nicht immer genauso stark sein, wie die obige Grafik zeigt. Dennoch wird deutlich, daß sich der Mehraufwand für die Implementierung einer FFT *immer* lohnt.

Prinzipiell gibt es auch noch andere Möglichkeiten, die Periodizität der DFT zu nutzen, um den Berechnungsaufwand zu vereinfachen. Auch ist es im Grunde egal, ob die Bitvertauschung vor oder nach der eigentlichen Berechnung durchgeführt wird. Ein Teil der Algorithmen arbeitet auch mit beliebiger Punktanzahl, oder auch mit Punktanzahlen, die Vielfache von 4 oder 16 sind. In [BRIG] ist ein Großteil der gängigen Verfahren vorgestellt, in [BETH] dagegen ist im wesentlichen die mathematisch - algebraische Darstellung der FFT-Theorie dargelegt. Für konkrete Implementationen auf einem Digitalen Signalprozessor der Reihen Motorola DSP56xxx oder DSP 96xxx gibt [MOTO] einen umfassenden Einblick in die Programmierung der FFT-Algorithmen in Assembler.

Ein kleiner Tip zur Implementierung des FFT-Algorithmus:

Der FFT-Algorithmus benötigt zu seiner Ausführung komplexe Multiplikationen. Bekanntlich berechnet sich das Produkt zweier komplexer Zahlen wie folgt:

Seien *a,b,c* komplexe Zahlen mit $a := a_0 + ja_1$, $b := b_0 + jb_1$, $c := c_0 + jc_1$.

Dann gilt:

$$c = a \cdot b \Leftrightarrow \begin{aligned} c_0 &= a_0 \cdot b_0 - a_1 \cdot b_1 \\ c_1 &= a_0 \cdot b_1 + a_1 \cdot b_0 \end{aligned}$$

Diese Berechnung erfordert 4 reelle Multiplikationen und 2 reelle Additionen, also benötigt die Berechnung insgesamt die Zeit $T_1 = 4T_{Mult} + 2T_{Add}$.

Es gibt allerdings folgenden Algorithmus zum Produkt komplexer Zahlen, der mit einer Multiplikation weniger auskommt:

$$\begin{aligned} m_0 &= a_0 \cdot b_0 \\ m_1 &= a_1 \cdot b_1 \\ m_2 &= \underbrace{(\overbrace{a_0 + a_1}^{Add}) \cdot (\overbrace{b_0 + b_1}^{Add})}_{Mult} \\ c_0 &= m_0 - m_1 \\ c_1 &= m_2 - m_1 - m_0 \end{aligned}$$

Wie man leicht nachprüft, kommt diese Berechnungsmethode mit 3 reellen Multiplikationen und 5 reellen Additionen aus, die notwendige Zeit liegt daher bei $T_2 = 3T_{Mult} + 5T_{Add}$.

Ein Vergleich von T_1 und T_2 zeigt, daß die zweite Berechnungsmethode dann schneller ist, wenn gilt: $T_{Mult} > 3 \cdot T_{Add}$. In diesem Vergleich ist allerdings der Mehraufwand für die Zwischenspeicherung der Ergebnisse nicht enthalten, man kann aber davon ausgehen, daß immer dann, wenn die Multiplikation 4-mal so lange dauert wie eine Addition, die zweite Methode sicher schneller ist.

Bei einem echten Signalprozessor kommt dieser Vorteil nicht so sehr zum Tragen, weil hier die Dauer einer Multiplikation kaum größer ist als die Dauer einer Addition. Aber für die Durchführung einer FFT steht nicht immer das Optimum zur Verfügung, oftmals müssen die Berechnungen auf einem konventionellen Mikroprozessor durchgeführt werden, vielfach sogar in einer Hochsprache. Gerade dann machen sich solche Einsparungen wieder bemerkbar, weil hier die Dauer einer Fließkomma-Multiplikation immer wesentlich größer ist als die Dauer einer Fließkomma-Addition.

Mischung von DFT und kontinuierlicher Fouriertransformation

Aus der Systemtheorie, die ja u.A. auf der *kontinuierlichen* Fouriertransformation basiert, sind umfangreiche Kenntnisse auf dem Gebiet der Übertragungsfunktionen, insbesondere von Filtern, vorhanden. Wie kann man diese Erkenntnisse auf den diskreten Fall übertragen?

Wie zuvor bereits gezeigt, existiert zwischen den Indizes der transformierten Folgenwerte und der zugehörigen Frequenz eine eindeutige Zuordnung. Will man nun ein diskretes Spektrum, das man mit Hilfe einer DFT gewonnen hat, durch eine kontinuierliche Übertragungsfunktion bewerten, so stellt sich die Aufgabe wie folgt dar: zu jedem Index läßt sich eine Frequenz berechnen. An dieser Frequenz läßt sich die Übertragungsfunktion auswerten, und dieses Ergebnis wird mit dem Wert der Folge multipliziert. Wendet man dieses "Kochrezept" allerdings ohne Nachzudenken auf alle Werte der transformierten Folge an, so wird sich das gewünschte Ergebnis nicht einstellen.

Der Grund dafür liegt wieder einmal in den Effekten, die die Abtastung im Frequenzbereich verursacht. Wie zuvor schon gesehen, enthält nur die erste Hälfte der transformierten Folge Frequenzen unterhalb der halben Abtastfrequenz. Oberhalb der halben Abtastfrequenz liegt eine Spiegelung der ersten Hälfte vor. In der Konsequenz bedeutet dies, daß man auch die kontinuierliche Übertragungsfunktion an der halben Abtastfrequenz spiegeln muß, damit man korrekte Ergebnisse erhält. Diese etwas verwirrende Aussage soll zuerst mit einer Skizze verdeutlicht werden, bevor auf die mathematische Seite der Angelegenheit eingegangen wird.

Ein diskretes Frequenzspektrum hat zum Beispiel folgende Einhüllende:

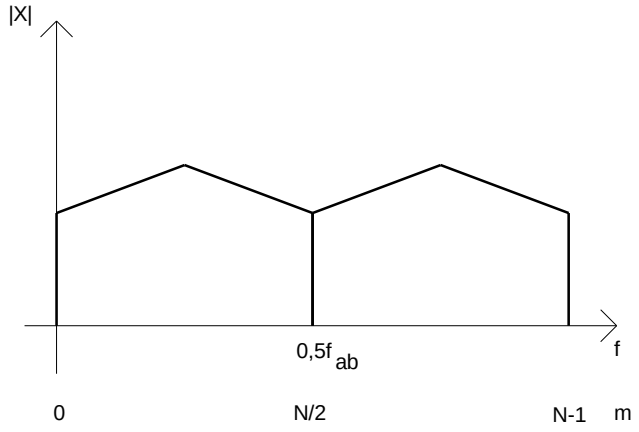


Bild 19: Einhüllende eines periodischen Spektrums

Als Übertragungsfunktion wird ein Tiefpaß gewählt: $G(\omega) = \frac{1}{1 + j\omega T_g}$, $f_g << f_{ab}$, dessen Übertragungsfunktion folgende Darstellung im Frequenzbereich hat:

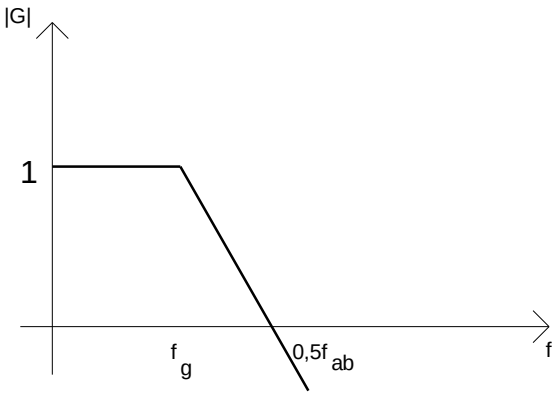


Bild 20: Übertragungsfunktion (Betrag) eines Tiefpaß 1. Ordnung

Um die gewünschte Tiefpaßfilterung zu erreichen, muß die Übertragungsfunktion des Tiefpasses ebenfalls an der halben Abtastfrequenz gespiegelt werden. Erst danach ist eine Multiplikation der Werte möglich und sinnvoll.

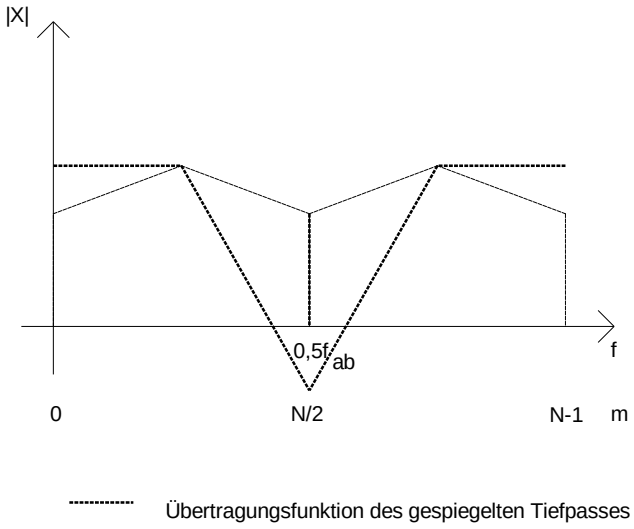


Bild 21: Gespiegelte Übertragungsfunktion Tiefpaß 1. Ordnung

Berechnet wird dies wie folgt:

Sei $G(\omega)$ eine kontinuierliche Übertragungsfunktion, X_m die diskrete Fouriertransformierte der Eingangsfolge. Dann gilt für die diskrete Fouriertransformierte der Ausgangsfolge Y_m:

$$0 \le m \le \frac{N}{2}: \qquad Y_m = X_m \cdot G\left(\frac{m}{N} \cdot 2\pi f_{ab}\right)$$

$$\frac{N}{2} + 1 \le m \le N - 1: \quad Y_m = X_m \cdot G\left(\frac{(N-m)}{N} \cdot 2\pi f_{ab}\right)$$

Wobei f_{ab} Abtastfrequenz, N die Anzahl der Abtastwerte. Die Anzahl der Abtastwerte muß für dieses Berechnungsschema geradzahlig sein, was aber keine Einschränkung darstellt: in der Regel wird man die DFT mit Hilfe des FFT-Algorithmus berechnen, so daß die Anzahl der Werte immer durch 2 teilbar sein wird (für ungerades N wird lassen sich die beiden Gleichungen natürlich ebenfalls aufstellen, nur sind dann die Fallunterscheidungen etwas anders; allerdings wird dies in der Praxis nicht benötigt).

Digitale Systemidentifikation

Unter dem Begriff Systemidentifikation versteht man die Aufgabe, den Frequenzgang *G(w)*, bzw. im Zeitbereich die Impulsantwort *g(t)*, eines Systems aus der Erregerfunktion *x(t)* und der Antwortfunktion *y(t)* zu berechnen.

Das Problem tritt bei der experimentellen Systemanalyse auf: *x(t)* und *y(t)* werden an einem unbekanntem System gemessen, dessen Systemeigenschaften man ermitteln möchte.

Diese Aufgabe ist genau die Umkehrung zur Bestimmung der Antwort eines Systems auf eine bestimmte Erregerfunktion (z.B. Sprung, Rampe, etc.):

$$Y(\omega) = G(\omega) \cdot X(\omega)$$

Hieraus folgt für die Systemidentifikation:

$$G(\omega) = \frac{X(\omega)}{Y(\omega)}$$

Wenn man nun entsprechend den Darlegungen der vergangenen Abschnitte alle Fouriertransformierten durch ihre diskreten Fouriertransformierten näherungsweise darstellt, so lautet eine numerische Bestimmungsgleichung:

$$G_m = \frac{X_m}{Y_m} = \frac{\mathrm{F_N}\{x_n\}}{\mathrm{F_N}\{y_n\}} \quad , 0 \le m \le \frac{N}{2}, Y_m \neq 0 \qquad (18)$$

(Übrigens: die Division in (18) ist eine komplexe Division!)

In (18) ist X_m die diskrete Fouriertransformierte der abgetasteten Erregerfunktion, und Y_m die diskrete Fouriertransformierte der abgetasteten Antwortfunktion. Die Division wird nur mit der ersten Hälfte der Folgenwerte durchgeführt, da die weiteren Werte wegen der Spiegelung im Frequenzbereich redundant sind.

Somit läßt sich unter Anwendung des Abtasttheorems auch genau die Grenzfrequenz dieses Verfahrens angeben: da man nur die erste Hälfte der Werte verwendet, ist die so bestimmte diskrete Übertragungsfunktion nur bis zur halben Abtastfrequenz eine Näherung zur kontinuierlichen Übertragungsfunktion:

$$G_m \approx G(\omega) \quad , \omega = m \cdot \frac{2\pi \cdot f_{Ab}}{N} \, , \omega < \frac{2\pi \cdot f_{Ab}}{2}$$

Die Abtastung muß natürlich für Erregerfunktion und Antwortfunktion mit derselben Anzahl von Abtastwerten und derselben Abtastzeit durchgeführt werden. Ein vorhandenes Antialiasing-Filter verursacht übrigens keinen Fehler in der Messung, seine Übertragungsfunktion würde herausgekürzt. (Natürlich nur, wenn man immer das gleiche Filter verwendet!) Vorsicht dagegen ist bei Verwendung von Fensterfunktionen geboten, da man hier eine Faltung im Frequenzbereich durchführt, so daß das Resultat der Division verändert wird. Es ist hier daher ratsam, außer dem Rechteckfenster keine anderen Fensterfunktionen einzusetzen.

Für die praktische Durchführung von (18) ist natürlich die Genauigkeit dieser Approximation wichtig. Aus der numerischen Mathematik weiß man, daß bei einer Division durch eine kleine Zahl

sich bereits sehr kleine Fehler dieser Zahl fatal auf das Ergebnis auswirken können. Denn bei einer konkreten Anwendung muß man sich über eines im Klaren sein: bei einer Abtastung ist durch die damit verbundene AD-Umsetzung immer (!) ein Quantisierungsfehler vorhanden. Und dieser Fehler ist bei einer kleinen Zahl prozentual sehr groß, daher wird der relative Fehler des Ergebnisses ebenfalls sehr groß.

Beispiel:

Sei der absolute Fehler von *Y* gleich $\Delta Y=1$. Dann gilt für ein großes *Y*, z.B. 128, daß man um 127, 128 oder 129 teilt. D.h., der Fehler wirkt sich nicht besonders gravierend aus. Ist dagegen $Y=1$, so teilt man entweder durch 0, 1 oder 2, d.h. das Ergebnis wird unbrauchbar, da man als Fehlerabschätzung Werte zwischen dem halben Wert und "unendlich" erhalten würde.

Besonders ungünstig fällt die Division aus, wenn der Nenner für einen Wert *m* verschwindet, d.h. gleich Null wird. In diesem Fall läßt sich über den Frequenzgang des Systems bei dieser Frequenz keine Aussage treffen. Da im Nenner von (18) die DFT der Erregerfunktion auftaucht, bedeutet dies für die praktische Wahl der Erregerfunktion, daß das diskrete Spektrum der Erregerfunktion möglichst keine Nullstellen besitzen darf.

Es ist daher bei der Systemidentifikation weniger das Problem, die Berechnungen des Frequenzganges durchzuführen, vielmehr ist die Hauptschwierigkeit eine Erregerfunktion zu finden, die keine Nullstellen im Spektrum besitzt, und dennoch real erzeugbar ist.

Im folgenden ist nocheinmal die Rechenvorschrift zur Systemidentifikation zusammengestellt:

1. Man tastet die Erregerfunktion *x(t)* ab, und bestimmt mit der DFT (oder FFT) die diskrete Fouriertransformierte X_m. Die Erregerfunktion sollte im Spektrum im interessierenden Frequenzbereich möglichst wenig Nullstellen besitzen.

2. Man tastet die Antwortfunktion *y(t)* ab, und bestimm mit der DFT (oder FFT) die diskrete Fouriertransformierte Y_m.

3. Man bestimmt den diskreten Frequenzgang des Systems durch die Division $G_m = \frac{X_m}{Y_m}$

für $0 \le m \le \frac{N}{2}$

4. Falls auch im Zeitbereich die Impulsantwort von Interesse ist, so läßt sich diese leicht aus einer inversen diskreten Fouriertransformation gewinnen: $g_n = \mathrm{F}_\mathrm{N}^{-1}\{G_m\}$

Näheres zu der Systemidentifikation und der Fehlerabschätzung ist in [LANG] zu finden, zu den möglichen Fehlerquellen in [ELPR].

Literaturangaben:

[BETH] Beth, Thomas: Verfahren der schnellen Fourier-Transformation. Algebraische Beschreibung, Komplexität und Implementierung. Stuttgart: Teubner, 1984

[BRIG] Brigham, E. Oran. FFT Schnelle Fourier-Transformation. München, Wien: Oldenbourg Verlag, 1987

[ELPR] Fachartikel Elektronik Praxis Nr. 22, 24. November 1994, "Trivial aber fatal - Typische Fehlerquellen beim Einsatz von FFT-Analysatoren"

[ENDE] van den Enden, Ad W. M.: Digitale Signalverarbeitung. Braunschweig, Wiesbaden: Friedrich Vieweg & Sohn, 1990

[GÜTI] Güting, Ralf Hartmut: Datenstrukturen und Algorithmen. Stuttgart: Teubner Verlag, 1992

[HACK] Sir Hackett, John: Die Welt in Flammen. Tagebuch des 3. Weltkrieges

[HESS] Hesselmann, Norbert: Digitale Signalverarbeitung: Rechnergestützte Erfassung, Analyse und Weiterverarbeitung analoger Signale. Würzburg: Vogel, 1983

[LANG] Lange, Dieter: Methoden der Signal- und Systemanalyse: Eine Einführung mit dem Personalcomputer. Braunschweig, Wiesbaden: Vieweg, 1985.

[LOCH] Locher, Franz: Mathematik III (numerische Mathematik), Studienbrief der Fernuniversität Hagen, 1989; erschienen im Hüthig-Verlag unter dem Titel Numerische Mathematik

[LÜKE] Lüke, Hans Dieter. Signalübertragung. Einführung in die Theorie der Nachrichtenübertragungstechnik. Berlin, Heidelberg, New York: Springer Verlag, 1979

[MOTO] Implementation of Fast Fourier Transforms on Motorola's Digital Signal Processors. Motorola, Digital Signal Processing Division, 1993. Bestellnummer APR4/D

[OPPE] Oppenheim, Alan: Signale und Systeme. Weinheim, Basel (CH), Cambridge, New York: VCH Verlagsgesellschaft, 1992

[SCHR]: Schrüfer, Elmar: Signalverarbeitung: Numerische Verarbeitung digitaler Signale. München, Wien: Hanser Verlage, 1990