Dr. Lothar Wenzel

Digitale Signalverarbeitung ist keine Hexerei

Teil 2: Fouriertransformationen sowie einige Verallgemeinerungen

Zur Überführung von Signalen in andere bedient man sich typischerweise verschiedener Transformationstechniken. Dieser Teil der Serie berichtet von der wohl bedeutendsten Signaltransformation, und zwar der nach Fourier. Ferner werden einige wichtige Verallgemeinerungen vorgestellt.

Transformationen spielen in der Signaltheorie eine große Rolle. Generell geht es hierbei um die Überführung eines Signals in ein anderes. Manchmal bleibt man faktisch im selben Raum, z.B. dann, wenn es um die Filterung von vorgegebenen Signalen geht. Oftmals ändert sich der Signalcharakter während der Transformation aber auch grundlegend. Zuweilen, aber nicht in allen Fällen, gelingt eine aus physikalischer Sicht sinnvolle Interpretation des neuen Signals. Hier geht es um die wohl wichtigste Signaltransformation, nämlich die Fouriertransformation.

„Touch-Tone“-Telefonie durch die Fourier-„Brille“

Eine interessante Anwendung der Fouriertransformation ist die „Touch-Tone“-Telefonie. Wenn man eine Taste eines Telefons drückt, so hört man in der Regel zweierlei. Zuerst wird ein Ton (Piepser) wiedergegeben, der nur als Kontrolle für den Anrufer dient. Unabhängig von der gewählten Taste wird stets derselbe Ton erzeugt. Kurz danach kann man aber die eigentliche Information hören. Diese muß in irgendeiner Weise eine Codierung der Taste darstellen. Und gerade an dieser Stelle setzt die Fouriertransformation an. Auch ohne einen tieferen Einblick in die Theorie dieser Transformation zu besitzen, kann man sicherlich die verschiedenen Teile von *Bild 1* ausdeuten. Gezeigt ist ein sogenanntes Leistungsspektrum, dessen Zusammenhang mit der Fouriertransformation später noch erläutert wird.

Bei der Touch-Tone-Telefonie lassen die Spektren deutlich jeweils zwei Spitzen erkennen, wobei die Lage der Spitzen auch nicht ganz ohne Ordnung ist. In unterschiedlicher Kombination werden eigentlich immer dieselben Frequenzanteile in Verbindung gebracht. Diese Kombinationen sind gerade der Schlüssel zu einer erfolgreichen Codierung bzw. Decodierung bei der Touch-Tone-Telefonie. Deutet man die Muster der Leistungsspektren richtig aus und trägt die gewonnenen Resultate in Matrixform auf, zeigt sich das in *Bild 2* dargestellte Schema.

Aus der spektralen Verteilung läßt sich noch mehr herauslesen. Es ist so, daß die zu erkennenden Peaks, resultierend aus dem Trägerton und den beiden zur Decodierung der Zifferntasten erforderlichen Frequenzen (Bild 2), nicht alle gleichzeitig auftreten. Fährt man mit einem beweglichen Fenster mit einer Breite von etwa 0,05 Sekunden über das Signal hinweg, zeigen sich je nach momentaner Position verschiedene spektrale Muster. Anhand der Tastenauslösung der Ziffer „2“ läßt sich das nachvollziehen: Unmittelbar vor der Tastenbetätigung liegt nur der Trägerton vor, gleich nach dem Drücken der Taste „2“ erscheint neben der Trägerfrequenz auch die in der ersten Decodierungsphase erforderliche Frequenz. Es schließt sich das Paar der Trägerfrequenz, gekoppelt mit der zweiten notwendigen Decodierfrequenz an.

In der Praxis wird die hier genutzte Fouriertransformation bei dieser Problemstellung jedoch nicht eingesetzt. Hierzu ist der Rechnenaufwand zu hoch. Hinzu kommt, daß man sehr genaue Informationen über alle zugrundeliegenden Frequenzen besitzt.

In solchen Fällen kann man auf die viel wirkungsvolleren digitalen Filter zurückgreifen, die bekanntlich im Zeitbereich agieren und nur relativ wenige Rechenoperationen

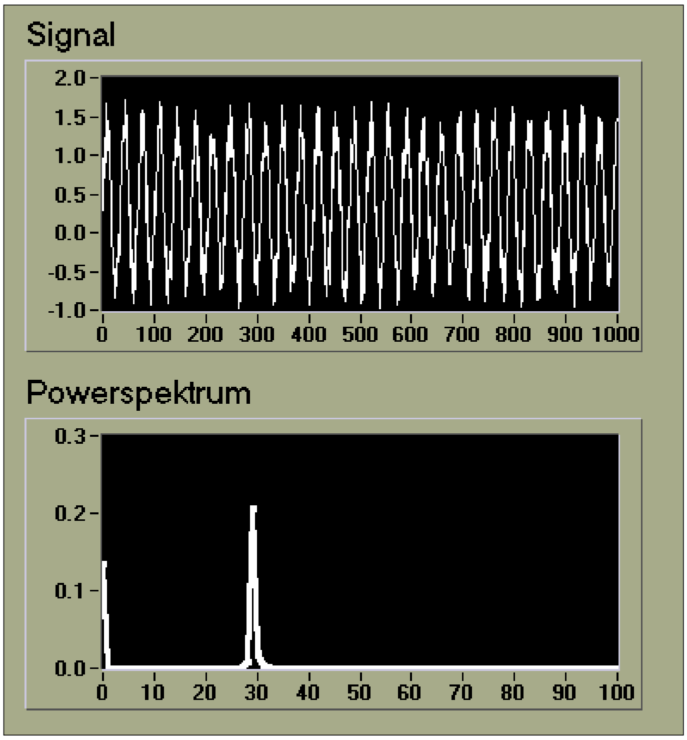

Bild 1. Powersprekten sind ein überaus wichtiges Mittel, um Signale zu charakterisieren. Peaks des Powersprektums eines gegebenen Signals lassen sich direkt als Energie-Inhalte deuten. Das dargestellte Signal basiert im wesentlichen auf einer Sinusschwingung. Im Powersprektrum bildet sich der entsprechende Energie-Inhalt deutlich ab. Ferner ist im Beispiel ein Gleichanteil vorhanden.

erfordern. Derartige Filter können so ausgelegt werden, daß sie spezielle Frequenzen und nur diese detektieren. Aus der Reihenfolge, in der diese Filter ansprechen, kann die gesamte gewählte Ziffernfolge eindeutig rekonstruiert werden.

Kontaktaufnahme

Das eben geschilderte Prinzip läßt sich interessanterweise auch umkehren. Hierzu muß man eigentlich nur in der Lage sein, Sinusschwingungen gegebener Frequenz, Amplitude und Dauer passend übereinander zu legen. Eine dieser Sinusschwingungen ist hierbei immer der bereits erwähnte Trägerton. Die Länge und die zeitliche Trennung der beiden Codierfrequenzen müssen gewissen Forderungen genügen.

Die bisher erläuterten Einzelheiten lassen sich ganz gut programmtechnisch umsetzen. Wichtig sind hierbei insbesondere einige vorbereitende Komponenten, die zunächst erläutert werden müssen.

Hörbares

Wie im ersten Teil der Artikelserie [1] erwähnt, macht die nicht allgemein voraussetzbare Verfügbarkeit von Datenerfassungskarten einen anderen Weg erforderlich. Die einzig gangbare Vorgehensweise stützt sich auf die Audio-Fähigkeiten der modernen PCs. Mit anderen Worten, es wird ein entsprechendes Hardware-Feature im Rechner des Nutzers erwartet. Dieses muß um ein Mikrofon (manchmal können auch andere Quellen genutzt werden) und um Kopfhörer oder Lautsprecher ergänzt werden. Allerdings sind die standardmäßig solchen Zusatzkarten beigefügten Software-Instrumente nicht geeignet, um signaltheoretische Fragestellungen zu behandeln.

Insofern besteht die erste Aufgabe darin, ein funktionsfähiges Audio-Instrumentarium seitens der Software aufzubauen. Zu diesem Zweck soll mit der Aufnahme, Ausgabe und Speicherung von Tonsequenzen begonnen werden. Zur konkreten Nutzung ist die Kenntnis einiger Parameter unerläßlich. Hierunter fallen insbesondere:

- Taktrate der Aufnahme (11025, 22050 oder 44100 Punkte je Sekunde)
- Mono/Stereo
- 8/16 bit Genauigkeit

Diese Parameter decken bereits ein breites Spektrum an Einsatzmöglichkeiten ab. Am unteren Ende mit 11 025 Werten je Sekunde/Mono/8 bit sind Anwendungen wie Telefonie, Spracherkennung oder Filter plaziert. Das andere Extrem, 44 100 Werte je Sekunde/Stereo/16 bit, entspricht qualitätsmäßig bereits CD-Aufnahmen. Grob gesprochen: Es besteht prinzipiell kein Unterschied zwischen beiden Varianten. Der rechnerische Aufwand ist jedoch im zweiten Fall etwa 16mal höher.

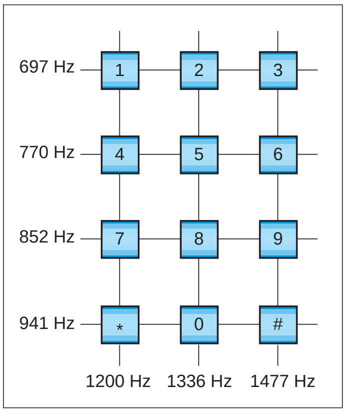

Bild 2. Die Touch-Tone-Anwahl basiert auf einigen wenigen Frequenzen. Dargestellt ist das Codierungsschema der verschiedenen Tasten eines Apparates.

Die Ausgabe von Audiosignalen folgt eben diesem Muster. Hierbei ist aber folgendes zu beachten: Das Abspielen einer qualitativ höheren Aufnahme ist auf einem schlechteren Niveau jederzeit möglich, die umgekehrte Richtung ist nicht realisierbar, es sei denn, man denkt an extrem ausgefeilte Spezialroutinen. Typischerweise werden Audioaufnahmen als Dateien abgelegt und von dort auch wieder eingelesen.

„INOUT“ ist ein erstes Beispiel für diese Art der Verarbeitung. Es geht hierbei nur um das simple Abspielen von Aufnahmen mit veränderbarer Qualität. Die Unterschiede müßten deutlich hörbar sein.

Besetzt-Zeichen

Nach diesen Vorbereitungen kann man auf die Touch-Tone-Telefonie zurückkommen. „TOUCHIN“ erlaubt die Aufnahme und Analyse einzelner spezieller Tasten eines Touch-Tone-Telefons. Man nehme hierzu einige Sekunden direkt mit einem Mikrofon auf, wobei man passende Tasten betätigen sollte. Nach der Aufnahme kann man sich mit Hilfe des Programms mit einem Zeitfenster der Breite 0,1 s über das Gesamtsignal hinwegbewegen und parallel dazu ein Leistungsspektrum produzieren. Im Spektrum zeigen sich jenseits der 400 Hz je nach Zeitpunkt unterschiedliche Peaks, die dem bereits genannten Codierungsschema gehorchen. Je nach Aufnahmecharakteristik kann auch ein mehr oder wenig hoher Anteil bei 0 Hz vorliegen (Gleichanteil). Diese Komponente hängt nur von den Aufnahmebedingungen ab und ist hier vollkommen ohne Belang.

Funktionen im Baukastensystem – die Fouriertransformation

Hinter der Fouriertransformation steckt die Idee, Funktionen jeglicher Komplexität aus sehr einfachen Basiselementen aufzubauen. Diese Vorgehensweise ist in der angewandten Mathematik keinesfalls selten. So wird bekanntlich versucht – mit Hilfe von Potenzfunktionen und Linearkombinationen dieser Bausteine –, buchstäblich jede stetige Funktion zu approximieren. Letzteres gelingt auf endlichen Intervallen perfekt. Bei Fouriertransformationen besteht die Hoffnung, daß diese Annäherung sogar für möglichst viele weitere Funktionen auf endlichen Intervallen perfekt ist. Für eine äußerst umfangreiche Klasse trifft das auch tatsächlich zu, wobei die Basiselemente aus Sinus- bzw. Cosinusfunktionen mit abgestuften Frequenzen bestehen.

Allerdings muß man hierbei den Begriff der Annäherung aufweichen. Wie *Bild 3* belegt, liegt die Konvergenz keinesfalls punktweise vor, sondern vielmehr nur im sogenannten quadratischen Mittel.

Die formale Definition der Fouriertransformation eines diskreten Signals x[0], x[1], ..., x[n] sieht wie folgt aus, wobei mit y[0], y[1], ..., y[n] das jeweilige Ergebnis dieser Umwandlung bezeichnet sei:

$$\begin{aligned} y[k] = 1/\{n+1\} \times \{&x[0]\exp(2\pi j0k/(n+1))+\\ &+x[1]\exp(2\pi j1k/(n+1))+\ldots+\\ &+x[n]\exp(2\pi jnk/(n+1))\} \end{aligned} \tag{1}$$

Hierbei steht j für die imaginäre Einheit, die Exponentialfunktion einer rein imaginären Größe $j\omega$ läßt sich gemäß nachfolgender Formel berechnen:

$$\exp(j\omega) = \cos\omega + j\sin\omega \tag{2}$$

In (2) finden sich die bereits erwähnten Sinus- und Cosinusfunktionen wieder. Gleichzeitig belegt (2) in Verbindung mit (1), daß das Signal y i.allg. komplexe Werte annimmt. Diese Eigenschaft ist durchaus charakteristisch für Signaltransformationen. Oftmals werden künstliche Signale produziert, die nicht in jedem Fall eine Verbindung zur realen Welt aufweisen. Für die Fouriertransformation ist das übrigens nur in einem geringen Maße zutreffend, hier gelingt solch eine physikalisch relevante Interpretation ohne Schwierigkeiten. Es zeigt sich, daß $|y[k]|^2$ sich als Energie-Inhalt des zugeordneten Frequenzanteils ausdeuten läßt. In diesem Zusammenhang ist das Parseval'sche Theorem von Bedeutung, das im wesentlichen die Energie-Inhalte des originalen und des transformierten Signals als identisch nachweist.

Mit der Fouriertransformation ist eine Reihe weiterer fundamentaler Eigenschaften und Definitionen verbunden. Hierunter fällt zunächst die inverse Fouriertransformation, deren Aufbau größte Ähnlichkeit mit der Ursprungsdefinition aufweist. Von Interesse ist ferner das sogenannte Faltungsgesetz, das die Faltungsoperation mit der Fouriertransformation verbindet.

Das Programm „FOURIER“ gestattet ein relativ freies Experimentieren mit dem Gegenstand der Fouriertransformation. Um ausreichend viel Ausgangsmaterial zur Verfügung zu haben, ist ein einfacher Trick möglich. Signale werden mittels symbolischer Formeleingabe generiert und durch Vorgabe einer Elemente-Anzahl diskretisiert. Die zulässigen symbolischen Komponenten sind in *Tabelle 1* aufgelistet, wobei eine fast unbeschränkte Kombinationsvielfalt besteht. Insbesondere lassen sich mit FOURIER die eingangs erwähnte Approximationseigenschaft und das Parseval'sche Theorem anhand von Beispielen veranschaulichen.

abs(x)	absoluter Wert von x
acos(x)	Umkehrung der Cosinus-Funktion
acosh(x)	Umkehrung des Cosinus Hyperbolicus
asin(x)	Umkehrung der Sinus-Funktion
asinh(x)	Umkehrung des Sinus Hyperbolicus
atan(x)	Umkehrung der Tangens-Funktion
atanh(x)	Umkehrung des Tangens Hyperbolicus
ci(x)	Cosinus-Integral
ceil(x)	rundet nach oben
cos(x)	Cosinus-Funktion
cosh(x)	Cosinus Hyperbolicus
cot(x)	1/tan(x)
csc(x)	1/sin(x)
exp(x)	Exponential e^x
floor(x)	rundet nach unten
gamma(x)	Gamma-Funktion
int(x)	übliche Rundung
ln(x)	Logarithmus zur Basis e
log(x)	Logarithmus zur Basis 10
log2(x)	Logarithmus zur Basis 2
pi(x)	3,14159...
rand(x)	Zufallszahl zwischen 0 und 1
sec(x)	1/cos(x)
si(x)	Sinus-Integral
sign(x)	Vorzeichen von x
sin(x)	Sinus-Funktion
sinc(x)	sin(x)/x
sinh(x)	Sinus Hyperbolicus
spike(x)	Spike-Funktion
sqrt(x)	Wurzel-Funktion
square(x)	Square-Funktion
step(x)	Step-Funktion
tan(x)	Tangens-Funktion
tanh(x)	Tangens Hyperbolicus

Tabelle 1. Eine Liste der zulässigen elementaren Funktionen. Jegliche Kombinationen der einzelnen Komponenten sind möglich.

Rechnerische Grundlagen

Die Formel (1) besticht durch ihre relative Einfachheit. Letzteres steht in keinem Verhältnis zur enormen Bedeutung der Fouriertransformation in Theorie und Praxis. Die am Anfang des Artikels behandelte Problematik der Touch-Tone-Telefonie könnte prinzipiell bereits auf Grundlage der Formel (1) gelöst werden. Es gibt nur einen einzigen Grund, der die Berechnungsvorschrift aus praktischer Sicht faktisch unattraktiv macht – der Zeitaufwand. Dazu genügt ein einfacher Kalkulationsüberschlag. Für eine Signallänge von der Größenordnung n benötigt man für die Berechnung eines einzigen Wertes y[k] etwa n Multiplikationen und n Additionen. Hierbei ist die Ermittlung der Werte für den exponentiellen Anteil nicht einmal mit erfaßt (liegen bei Spezialanwendungen tabelliert vor). Zur Ermittlung aller Elemente des resultierenden Signals y sind somit Operationen in der Größenordnung von $2 \times n^2$ nötig.

Diese Zahl sieht harmlos aus, ist es aber keinesfalls, wenn man bedenkt, daß eine mit 44 100 16-bit-Werten je Sekunde und zweikanalig aufgenommene Stereoaufnahme bereits fast n = 100 000 innerhalb einer einzigen Sekunde bedeutet. Ein Rechenaufwand von $2 \times n^2 = 2 \times 10^{10}$ Operationen je Sekunde wäre gegenwärtig nur von Supercomputern zu bewältigen, liegt also weit jenseits realer Anwendungen des Tagesgeschehens.

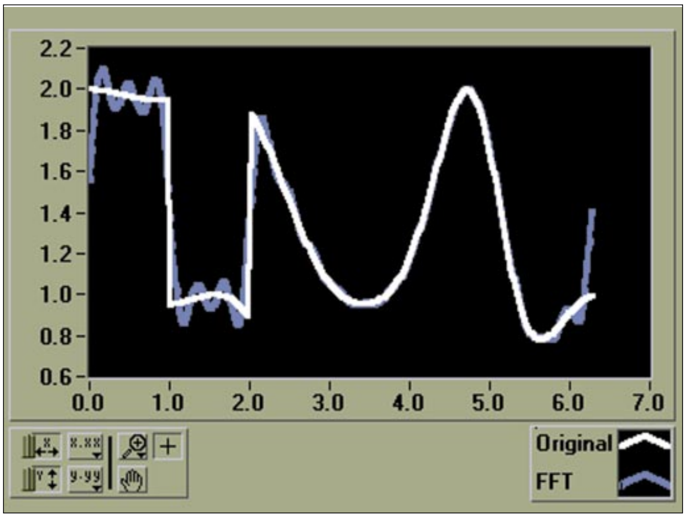

Bild 3. Die Klasse der sogenannten quadratisch integrierbaren Funktionen erlaubt eine perfekte Annäherung, basierend allein auf Sinus- und Cosinusfunktionen. Allerdings ist die Approximation nicht mehr punktweise oder sogar uniform für alle Punkte gleichzeitig zu verstehen. Vielmehr geht es genauer um die Konvergenz im quadratischen Mittel. Diese ist deutlich schwächer als etwa die uniforme Konvergenz. Historisch gesehen vergingen viele Jahre, bis die Forscher diesen Sachverhalt genau erkannten. Man erkennt im Bild die charakteristischen Unterschiede zwischen Original und Approximation.

Diese Werte sind so deprimierend, daß es trotz der vielen positiven Eigenschaften für die Fouriertransformation in der Praxis eigentlich nichts zu tun gäbe. Zum Glück wurde aber Mitte der 60er Jahre von Cooley und Tukey in einer berühmten Arbeit gezeigt, daß der Rechenaufwand der Größenordnung n^2 auf $n \times \log_2 n$ heruntergedrückt werden kann. Der Term $\log_2$ steht hierbei für den Logarithmus zur

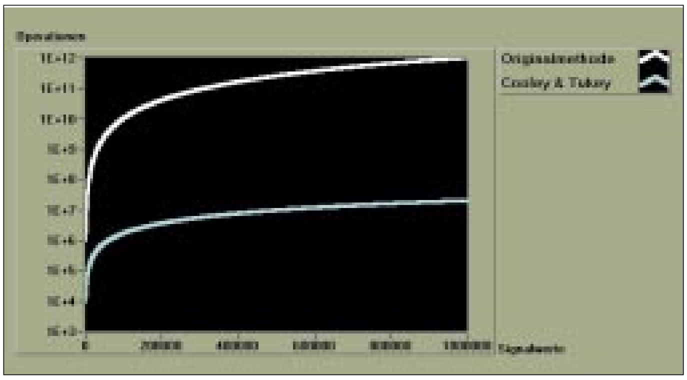

Bild 4. Der schnelle Algorithmus zur Berechnung der Fouriertransformation müßte eigentlich den Namen „superschnell“ tragen. Das Beispiel demonstriert die enormen Unterschiede zwischen der ursprünglichen Fassung und der verbesserten Version.

Basis 2. Die dadurch erzielte Rechenzeitersparnis ist außerordentlich hoch. *Bild 4* belegt diesen Sachverhalt anhand eines Demonstrationsbeispiels. Als Fazit bleibt, es liegen ganze Welten zwischen einem Rechenschema gemäß Formel (1) und der auf den Resultaten von Cooley und Tukey beruhenden Methode.

Der Schmetterlings-Algorithmus

Das von Cooley und Tukey genutzte Prinzip war im Grunde genommen bereits viel länger bekannt. Die Idee läßt sich bis zum großen deutschen Mathematiker Gauß zurückverfolgen. Im Unterschied zu den Jahrzehnten davor waren die 60er Jahre unseres Jahrhunderts aber von einem unmittelbaren Bedarf an schnellen Algorithmen für die Fouriertransformation geprägt. Sowohl die Rechentechnik als auch die Meßdatenerfassung hatten einen Stand erreicht, der einen derartigen Wunsch rechtfertigte.

Cooleys und Tukeys Algorithmus wird heute zumeist abkürzend FFT genannt, was als Fast Fourier Transform zu lesen ist. Die zugrundeliegende Idee läßt sich bestens in grafischer Form präsentieren (*Bild 5*), wodurch der ebenfalls gebräuchliche Begriff des Schmetterlings-Algorithmus etwas klarer wird. Gezeigt ist die elementare Operation, der Gesamtalgorithmus setzt sich aus zahlreichen Einzelkomponenten der gezeigten Art zusammen. Die Signallängen müssen Potenzen von 2 sein. Das schien anfangs keine starke Einschränkung zu sein, da eine entsprechende Anpassung des Datenmaterials meistens unproblematisch ist; das bedeutet jedoch nicht, daß es immer möglich ist. Insofern müssen auch die Fälle behandelbar sein, bei denen die Signallängen keine Potenzen von 2 sind. Als hocheffizientes Verfahren hat sich in diesem Zusammenhang der „Chirp-z“-Algorithmus erwiesen, der heutzutage anderen Varianten vorgezogen wird und mit jeder vorgegebenen Signallänge umgehen kann.

Fouriertransformation als Kompressor

Der weiter oben erläuterte Approximationseffekt (man vergleiche hierzu auch insbesondere das Programm FOURIER), der bei Vernachlässigung höherer Frequenzanteile auftritt, kann wirkungsvoll zur Komprimierung von Daten eingesetzt werden. Aus praktischer Sicht ist diese Vorgehensweise durchaus angebracht, u.a. dann, wenn es nicht um Tonsequenzen sondern um zweidimensionale Signale in Form von Bildern geht. Genauer wird dieser Gegenstand in einer späteren Folge der Serie bei der Vorstellung des digitalen Fernsehens erläutert. Die Methodik ist aber in all diesen Fällen sehr ähnlich.

Ein gegebenes und vollständig vorliegendes Signal wird mittels der Fouriertransformation in Frequenzkomponenten zerlegt. Ab einer gewissen Frequenz setzt man diese Anteile einfach auf 0 zurück. Es können aber auch nur die Frequenzanteile aufbewahrt werden, die eine gewisse Intensität aufweisen. Jedenfalls wird das Signal hierbei auf einige wenige Werte reduziert, d.h. stark komprimiert. Wendet man auf dieses künstlich beschnittene Signal nun die inverse Fouriertransformation an, ergibt sich ein weiteres Signal, das oftmals dem ursprünglich gegebenen sehr ähnlich ist. Allerdings dürfte im Regelfall das Resultat all dieser Manipulationen eine Sequenz sein, die nicht nur einen Real- sondern auch einen physikalisch nur schwer deutbaren Imaginärteil besitzt. Daher wird bei diesem Prinzip der Kompression auch nicht wirklich die Fouriertransformation eingesetzt, sondern vielmehr ein Ableger von dieser – die Cosinustransformation. Diese Transformation agiert vollständig im Raum reeller Zahlen.

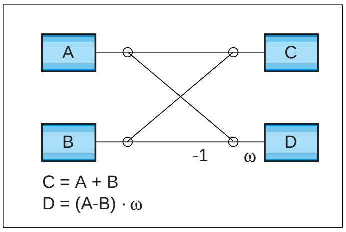

Bild 5. Die FFT wird auch als Schmetterlings-Algorithmus bezeichnet.

Den Effekt bei dieser Kompressionsart kann man sich anhand von „KOMPRESS“ anhören. Einige freie Parameter stehen zur Verfügung. Zu beachten sind hierbei auch die erzielten Kompressionsraten; alles hängt von einem annehmbaren Kompromiß ab. Beim praktischen Einsatz werden die wenigen ausgewählten Koeffizienten der zugrundeliegenden Transformation über sichere digitale Kanäle überspielt, und beim Empfang wird daraus ein möglichst genaues Duplikat des Originalsignals angefertigt. Diese Vorgehensweise ist oftmals qualitativ weitaus besser als die störanfälligen Transfers großer Mengen rein analoger Daten.

Eine ungewöhnliche Anwendung: Multiplikation großer Zahlen

Es wird eine Fragestellung aufgegriffen, die zumeist im Zusammenhang mit Computer-Algebra-Systemen auftaucht. Zuweilen spielen diese Probleme auch eine größere

Rolle bei komplizierten Codierungen bzw. Decodierungen. Genauer ist die Multiplikationen zweier äußerst großer natürlicher Zahlen gemeint. Um eine konkrete Aufgabenstellung zu wählen, sollen zwei Dezimalzahlen A und B von jeweils 1000 Stellen gegeben sein. Gesucht ist das Produkt C, das bis zu 2000 Stellen aufweisen kann.

Gespeichert werden solche Zahlen nicht mehr mittels interner Computerformate, vielmehr liegen sie als Felder von Dezimalziffern vor. Aus der Sicht der Signaltheorie sind solche Arrays aber nichts anderes als simple Signale. Es handelt sich hierbei um Korrelationsoperatoren, die allerdings nicht zyklisch zu verstehen sind. Insofern ist eine künstliche Verlängerung der gegebenen Felder erforderlich, was auf der anderen Seite aber keine dramatischen Konsequenzen in Hinblick auf das Rechenzeitverhalten hat.

Nutzt man nun noch das sogenannte Faltungstheorem, das die Berechnung der Korrelation mit Hilfe dreier Fouriertransformationen ermöglicht, so hat man insgesamt gesehen ein äußerst effizientes Multiplikationsschema für Zahlen mit sehr hoher Stellenzahl konstruiert. Einige Details des Algorithmus erfordern jedoch noch eine Erläuterung.

Unsere künstlichen Signale bestehen aus einfachen Ziffern, d.h. nur die Zahlen 0, 1,...9 sind mögliche Informationsträger. Dasselbe erwarten wir natürlich auch von den Ergebnissen. Da aber mehrfach die Fouriertransformation zum Einsatz kommt, agiert man vollständig im Bereich der reellen Zahl (genaugenommen sind es eigentlich sogar komplexe Zahlen). Das zwingt im Anschluß an die Rechnung zu einer Rundung. Im angegebenen Beispiel von 1000 Ziffern ist das unproblematisch, bei astronomisch vielen Stellen sollte man jedoch besser die langsamere aber sichere Schulmethode einsetzen. Im übrigen gibt es auch andere zahlentheoretische Transformationen, die das Faltungstheorem erfüllen. Diese sind sehr gut geeignet für die hier geschilderte Problematik.

Ferner liefert die Korrelation in aller Regel Überträge, die man passend verarbeiten muß. So liegt das Produkt C bei Vorlage der beiden Zahlen

$A = 7 \times 10^1 + 9 \times 10^0$ sowie $B = 8 \times 10^1 + 4 \times 10^0$
zunächst in der Form
$C = 56 \times 10^2 + 100 \times 10^1 + 36 \times 10^0$

vor. Das läßt sich aber schrittweise und von rechts beginnend in die Formen

$C = 56 \times 10^2 + 103 \times 10^1 + 6 \times 10^0$,
$C = 66 \times 10^2 + 3 \times 10^1 + 6 \times 10^0$ sowie
$C = 6 \times 10^3 + 6 \times 10^2 + 3 \times 10^1 + 6 \times 10^0$

transformieren. Diese Umwandlungen sind von einem vernachlässigbaren Aufwand im Verhältnis zur Gesamtrechnung.

Weitere Transformationsverfahren

Der nebenstehende *Kasten* gibt einen (sicher nicht ganz vollständigen) Überblick über andere Transformationen, die eng mit der Fouriertransformation verwandt sind. Der tiefere Grund für diese Vielfalt liegt in Spezialaufgaben, bei denen die einzelnen signaltheoretischen Abbildungen ihre jeweiligen Stärken ausspielen können. Insbesondere fällt hierunter die Eigenschaft, daß reellwertige Signale in ebensolche überführt werden. Das Entstehen eines Problems, wie das weiter oben geschilderte des Auftretens imaginärer Anteile im resultierenden Signal, wird somit von vornherein verhindert. Für die Cosinus-, Hartley- und Hilbert-Transformation wurden schnell arbeitende Algorithmen entwickelt.

Verwandte Transformationsverfahren

Fouriertransformation:

- Verfahren nach Goertzel
- Verschiedene Radix-Verfahren
- Verfahren, die auf Primzahlen beruhen
- Winograd-Methode (kurze Signale)
- Fractional Fourier Transform
- Reellwertige Transformationen

Zahlentheoretische Transformationen:

- Hartley-Transformation
- Hilbert-Transformation
- Cosinus- / Sinus-Transformation
- Hadamard / Walsh-Transformation

Chirp-Signale

Von großer Tragweite in der digitalen Signalverarbeitung sind Chirp-Signale. Diese werden gemäß dem einfachen Schema sin(at^2) generiert, wobei a eine Konstante und t die Zeit darstellen. Anhand des im ersten Teil der Serie vorgestellten Programms SIGNAL läßt sich der charakteristische Verlauf mit stetig anschwellender Frequenz hörbar machen. Das Leistungsspektrum zeigt, daß die Frequenzanteile etwa gleich stark vertreten sind. Die Diskretisierung führt zu einigen unbedeutenden Seiteneffekten, so daß zwischen Ideal und Wirklichkeit kleinere Differenzen auftreten. Die charakteristische Eigenschaft des Chirp-Signals, nämlich auf allen Frequenzbändern gleichstark vertreten zu sein, ist z.B. die Basis für die Systemerkennung bei unbekannter Übertragungsfunktion. Während eine einfache Sinus-Schwingung sozusagen nur punktuelle Informationen gewinnen kann, ist der Chirp ein Generalist. Wir kommen in einer späteren Folge noch darauf zurück.

Hersteller	DSP	Daten / Befehl	MIPS
Analog Devices	ADSP-21xx, Festkomma	16 / 24 bit	50
	ADSP-2106x, Gleitkomma	32 / 48 bit	40
Lucent Technologies	DSP16xx, Festkomma	16 / 16 bit	120
Motorola	DSP560xx, Festkomma	24 / 24 bit	47,5
	DSP563xx, Festkomma	24 / 24 bit	100
	DSP566xx, Festkomma	16 / 24 bit	80
	DSP568xx, Festkomma	16 / 16 bit	20
NEC	µPD7701x, Festkomma	16 / 32 bit	33
STMicroelectronics	D950-CORE, Festkommakern	16 / 16 bit	40
Texas Instruments	TMS320C1x, Festkomma	16 / 16 bit	8,8
	TMS320C2x, Festkomma	16 / 16 bit	12,5
	TMS320C2xx, Festkomma	16 / 16 bit	40
	TMS320C3x, Gleitkomma	32 / 32 bit	30
	TMS320C4x, Gleitkomma	32 / 32 bit	30
	TMS320C5x, Festkomma	16 / 16 bit	50
	TMS320C54x, Festkomma	16 / 16 bit	100
	TMS320C62xx, Festkomma	16 / 32 bit	200 bis 1600
	TMS320C8x, Fest- / Gleitkomma	8 / 16 bit u. 32 / 64 bit	3 oder 5 × 60
3Soft Corporation	M320C25, Festkommakern	16 / 16 bit	15
Zoran	ZR38xxx, Festkomma	20 / 32 bit	40

Tabelle 2. Leistungsparameter einiger bekannter DSP-Chips. Computer kommen diesen Werten sehr nahe, allerdings sprechen Preis, geringe Leistungaufnahme und Größe für die DSPs. FPGAs könnten schon bald ernsthafte Gegner der DSPs werden.

Die „Number-Cruncher“ – DSP-Chips

Trotz der enormen Leistungskraft herkömmlicher PCs sind DSP-Chips weiter präsent, und viele Experten glauben sogar, daß der ganz große Durchbruch noch gar nicht passiert sei. Schon von der Größe und den Leistungsaufnahmen her sind sie auf vielen Gebieten konkurrenzlos. Die Fouriertransformation gehört zum Standartrepertoire der meisten DSPs. Allerdings sind zumeist einige Besonderheiten zu beachten: Die Signallängen sind oftmals fixiert und relativ gering. Sind variable Signallängen zugelassen, so sind es im Regelfall nur einige wenige Zweierpotenzen. Außerdem werden nicht immer Gleitkomma-Formate unterstützt. Festkomma-Werte erfordern jedoch zusätzliche Überlegungen bezüglich der Rundungseffekte. Auf der anderen Seite ist man bei DSP-basierten Algorithmen sehr hardwarenah und schon deshalb schnell in der Ausführung. Die *Tabelle 2* stellt einige Leistungsparameter bekannter DSP-Chips vor. Zusätzlich sind einige weitere Daten zur besseren Orientierung mit aufgeführt. In jüngster Zeit deutet sich ferner an, daß FPGAs die Arena des DSP-Marktes betreten haben. Die freie Konfigurierbarkeit und immanente Parallelität der FPGAs könnten definitive Pluspunkte bei der sich andeutenden Auseinandersetzung sein. *gs*

Der nächste Teil der Serie befaßt sich mit einfacheren digitalen Filtern.

Anmerkung: Alle Programme wurden in National Instruments grafischer Programmiersprache LabVIEW geschrieben.

Literatur

[1] *Wenzel, L.*: Digitale Signalverarbeitung ist keine Hexerei. Teil 1, *Elektronik* 1998, H. 23, S. 96 bis 104.