

Dr. Lothar Wenzel

Digitale Signalverarbeitung ist keine Hexerei

Teil 1: Grundlagen der digitalen Signalverarbeitung

In dieser Serie steht die anschauliche Darstellung der digitalen Signalverarbeitung anhand praktischer Beispiele im Mittelpunkt. Die theoretischen Grundlagen bleiben auf ein Minimum beschränkt, vielmehr zeigt diese Artikelserie, wie man durch den Einsatz einfacher Software-Werkzeuge tiefer in die Niederungen der digitalen Signalverarbeitung hinabsteigen kann.

Ziel dieser Serie ist es, ein tieferes Verständnis für das äußerst umfangreiche Gebiet der digitalen Signalverarbeitung aufzubauen. Es gibt mit Sicherheit eine große Zahl an Fachartikeln und zahlreiche Bücher, die dieses Gebiet von allen denkbaren Seiten aus beleuchten. Was jedoch weitgehend fehlt, ist die Kombination von Theorie mit sofort einsetzbaren Softwaretools.

Damit ist weniger die Bereitstellung von Listings in verschiedenen Programmiersprachen gemeint. Vielmehr geht es um lauffähige Applikationen, die viel Spielraum für eigene Experimente lassen. Selbstverständlich müssen Details der zugrundeliegenden Algorithmen dem Leser mitgeteilt werden, jedoch sind letztere im Regelfall weniger an Interna der konkreten Implementierung interessiert. Im Mittelpunkt hat die Idee hinter einer neu einzuführenden Methodik und deren Funktionalität zu stehen.

Gewissensfragen, die zu beantworten sind

Zunächst hat man es beim Start eines derartigen Projektes mit zahlreichen offenen Fragen zu tun. Die wichtigsten davon sind:

- Welche Betriebssysteme sollten im Mittelpunkt stehen?
- Auf welcher Basis kann man digitale Signalverarbeitung studieren, falls entsprechende Zusatzboards nicht zur Verfügung stehen?
- Welche Themen sollen im Rahmen der Artikelserie behandelt werden?

Natürlich ist bei der Beantwortung dieser drei Fragen allerhand Verhandlungsspielraum möglich. Vielleicht ist der nachfolgend vorgestellte Plan ein ganz guter Kompromiß:

Als Betriebssystem wird all das zugelassen, was gegenwärtig von Microsoft für eine breite Anwenderschaft angeboten wird. Demnach bieten sich Windows 3.x, Windows 95 und Windows NT an. Somit dürfte so gut wie jeder Leser die in der Serie angebotenen Programme auch tatsächlich nutzen können. Aber so ganz selbstverständlich ist diese Wahl nicht. Hiermit ist weniger das Fehlen solcher Betriebssysteme wie MacOS gemeint. Vielmehr, und das zeichnet Routinen der digitalen Signalverarbeitung gegenüber anderen Zweigen aus, sind in der Praxis sehr wohl andere Plattformen im Einsatz. Gemeint sind damit in erster Linie Echtzeit-Betriebssysteme, DSP-Boards und verschiedene Arten von Multiprozessoranlagen. Diese Vielfalt ist in den oftmals zu stellenden Randbedingungen begründet. Diese verlangen mit Regelmäßigkeit von den Signalverarbeitungsroutinen die Einhaltung eines strengen Daten- und Zeitregimes, das basierend auf der Windows-Welt nicht zu garantieren ist. Auf der anderen Seite sind solche Applikationen zumeist Adaptionen von Standardalgorithmen.

men. Zwar sind diese Anpassungen nicht immer leicht oder gar automatisch auszuführen, aber das Wesen der zugrundeliegenden Algorithmen wird hiervon kaum beeinflusst. Erstes Fazit demnach: Die Windows-Familie ist eine geeignete Plattform für eine Vorstellung des Gebietes der digitalen Signalverarbeitung.

Datenverkehr in unterschiedlichen Formen

Die Beantwortung der Frage nach einer geeigneten Datenbasis für die geplanten Experimente sieht auf den ersten Blick nicht ganz trivial aus. Es kann nicht davon ausgegangen werden, daß in jedem Rechner eine Zusatzkarte eingebaut ist, die mit digitalen und analogen Daten sowohl ein- als auch ausgabeseitig umgehen kann. Insofern sind Alternativen zu suchen, die einen möglichst geringen Einfluß auf das Ziel dieser Serie nehmen, nämlich die Erläuterung signaltheoretischer Operationen.

Aus diesem Blickwinkel bieten sich zumindest drei verschiedene Vorgehensweisen an, die je nach Eignung später auch immer wieder zum Einsatz kommen werden. Die naheliegendste Methodik stützt sich auf vorgefertigte Datenfiles (ASCII-Notation).

Solche Dateien werden immer dann genutzt, wenn spezielle Effekte erläutert werden sollen, für die die noch zu erläuternden zwei weiteren Vorgehensweisen nur wenig Eignung mitbringen. Ein Beispiel ist die Systemidentifikation, die sich am besten anhand wirklich aufgenommener und sodann abgespeicherter Werte erläutern läßt.

Die zweite Methodik stützt sich auf Formeln. Damit ist bereits angedeutet, daß Formelmanipulationsmöglichkeiten integraler Bestandteil fast aller der im weiteren vorzustellenden Programme sein werden. Das ist schon deshalb bemerkenswert, weil eine Reihe der für die Signalverarbeitung wichtigen Funktionen eine sehr einfache Formelnotation haben. Einige der wichtigsten Vertreter sind die Step-, Spike- und die Square-Funktionen. Die Beschreibung der vollständigen Syntax des Formelmechanismus folgt an späterer Stelle ausführlich.

Aus anwendungstechnischer Sicht ist vielleicht die dritte Möglichkeit der Datenbehandlung die wichtigste. Gemeint ist die Verarbeitung von Meßwerten der realen Welt, wovon sowohl die Ein- als auch die Ausgabevorgänge betroffen sind. Hat man einen modernen PC mit einem der Windows-Betriebssysteme im Auge, so gibt es eigentlich nur einen einzigen Weg, der aus der Sicht der Signalverarbeitung zu interessanten Daten führt – man muß die Audiomöglichkeiten des Rechners nutzen. Konzepte wie Feedback, Filterung oder Signaltransformation lassen sich auf diese Art und Weise in anschaulicher (bzw. anhörbarer) Form prä-



Bild 1. Einer der möglichen Datentransporte zwischen verschiedenen Programmen basiert auf einfachen Dateien. Die Übergabe der zumeist als Feld (Array) vorliegenden Werte wird vom Anwender organisiert.

sentieren. Als Quellen bieten sich hier neben CD-ROMs auch in ein Mikrofon gesprochene Worte an. Das bedeutet gleichzeitig auch allerhand Spielraum für eigene Experimente. Softwareseitig wird eine geeignete Audioanbindung über Treiber hergestellt. Letztere sind Bestandteile der ladbaren Programme.

Bleibt die dritte Frage zu beantworten: Welche Themen sind zu behandeln? Das Gebiet der digitalen Signalverarbeitung ist von einer enormen Vielfalt geprägt. Zusätzlich bestehen mannigfaltige Querverbindungen zwischen fast allen

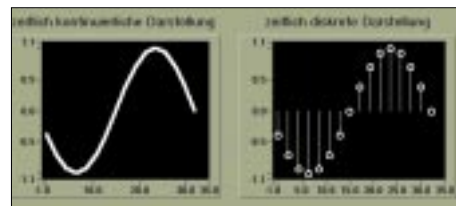


Bild 2. Signale können grundsätzlich als kontinuierlich oder als diskret betrachtet werden. Hierdurch wird die zeitliche Komponente betont.

Teilen dieses Theoriegebäudes. Wenn auch die nachfolgende Auswahl unbestritten subjektive Elemente enthält, so wird doch eine möglichst breite Palette abgearbeitet. Wie bereits gesagt, es werden alle Konzepte anhand von Beispielen erläutert. Zusätzlich befassen sich einige Teile der Serie ausschließlich mit Anwendungen. Versucht wird dabei immer, möglichst viel der bis zum entsprechenden Zeitpunkt verfügbaren Theorie zu kombinieren. Eine Konsequenz daraus ist, daß die gewählten Beispiele durchaus einen praxisnahen Hintergrund haben.

Der erste Teil befaßt sich mit einer Einführung sowie mit Signalen und deren Darstellung. Die weiteren Artikel werden sich in der genannten Reihenfolge mit folgenden Themen beschäftigen:

Grundlagen

2. Fouriertransformationen sowie einige Verallgemeinerungen
3. FIR-Filter
4. Digitale Filter mit Rückkopplung
5. Anwendungen des ersten Teils der Serie (Grundlagen)

Systeme

6. Systeme, Rückkopplung und PID-Regler
7. Abtastung, Laplace-Transformation und Signalrekonstruktion
8. Datenfitting und Vorhersage
9. Anwendungen des zweiten Teils der Serie (Systeme)

Fortgeschrittene Techniken

10. Wavelets und Joint-Time-Frequency-Transformationen
11. 2D-Signalverarbeitung
12. Anwendungen des dritten Teils der Serie (Fortgeschrittene Techniken)
13. Fallstudien

Die abschließenden Fallstudien beschreiben reale Problemstellung und Ansätze zu deren Lösung. Wie oft bei Vorlage praktisch relevanter Aufgabenstellungen gibt es erstens zu meist nicht einen einzigen Weg, der zum Erfolg führt, und zweitens ist eine Lösung des öfteren auch ein Kompromiß zwischen der Exaktheit der Resultate und deren effizienter Umsetzung. Letztlich ist ein noch so perfekter Algorithmus so gut wie wertlos, wenn z.B. das Laufzeitverhalten eine Echtzeit-Verarbeitung unmöglich macht.

Darüber hinaus erfordert die Bewältigung von Fallstudien mit Regelmäßigkeit die Koordination unterschiedlichster Werkzeuge. Deshalb müssen die in unterschiedlichen Teilen der Serie entwickelten und somit unabhängigen Tools zusammengeführt werden. Man könnte das mit Hilfe eines gemeinsamen Datencontainers für alle Programme realisieren. Allerdings ist dafür eine ganze Menge an Verwaltungsarbeit erforderlich. Kaum weniger effizient ist der Weg über Datenfiles (Bild 1). Hierbei gibt ein Programm A Resultate als Datei aus, die von einem zweiten Programm B direkt als Eingabe genutzt werden können. Einziger Nachteil, der Nutzer muß sich etwas einfallen lassen, um diese Vorgänge zu überwachen.

Warum eigentlich digital arbeiten?

Gute Frage! Für viele Teile der digitalen Signalverarbeitung gibt es Entsprechungen in der analogen Welt. Einige Jahrzehnte zurück gab es bekanntlich auch gar keinen anderen Weg. Computer sind nicht primär für die Belange der Signalverarbeitung im engeren Sinne entwickelt worden. Trotzdem dominiert die digitale Signalverarbeitung in dieser oder jener Form heutzutage unbestritten. Liegt es an der enormen Geschwindigkeit der Rechner?

Die Antwort darauf ist ganz eindeutig, daß die Geschwindigkeit der Computer zwar wichtig, aber nicht direkt verantwortlich für den enormen Erfolg der digitalen Signalverarbeitung ist. Genauer: Analoge Systeme sind um Größenordnungen schneller als deren digitale Entsprechungen. Wenn man z.B. einen simplen analogen Schaltkreis nimmt, so ist doch eines klar, die „Datenverarbeitung“ findet in diesem System im Bereich der Lichtgeschwindigkeit statt. Simulationen in der digitalen Welt müssen zwangsläufig drastische Verlangsamungen in Kauf nehmen. Man kann die relative Trägheit digitaler Systeme auch aus einem anderen Blickwinkel erläutern.

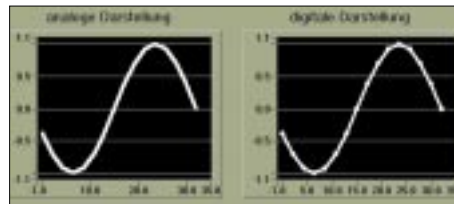


Bild 3. Die andere Charakterisierung beruht auf der eigentlichen Wertdarstellung. Denkbar sind analoge oder digitale Daten. Rechner müssen mit der zweiten Variante die erste nachbilden.

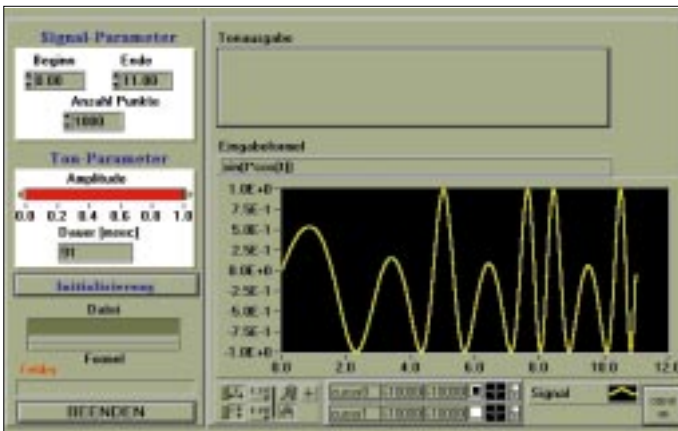


Bild 4. Das Programm SIGNAL bietet zahlreiche Möglichkeiten zur Signaldarstellung. Welche zu bevorzugen ist, hängt stark von der jeweiligen Anwendung ab.

Ein Rechner tut wirklich alles, um die zwei genutzten Zustände „0“ und „1“ rein zu halten. Aus analoger Sicht entspricht der „0“ nämlich ein ganzes Spektrum an Werten (dasselbe trifft natürlich für die „1“ zu). Die ganze Kunst liegt darin, daß diese beiden Einflußsphären perfekt auseinandergehalten werden oder, anders gesagt, analoge Entsprechungen der „0“ oder „1“ dürfen nicht die gering-

sten Überlappungen aufweisen. Wenn demnach für ein einziges Bit an Information Teile der analogen Skala benutzt werden müssen, dürfte verständlich sein, daß aus Sicht der Geschwindigkeit die digitalen Vertreter keine echte Konkurrenz darstellen.

Digital und analog

Aber gerade dieser enorme Aufwand für die Realisierung zweier Zustände macht auf der anderen Seite das digitale Konzept so leistungsstark. Ein erstes Argument betrifft die Stabilität. Analoge Systeme driften z.B. aufgrund von Temperaturschwankungen. Vom Computer erwartet man mit Recht, daß solche äußeren Einflüsse ohne Wirkung bleiben. Neben der Stabilität der Lösung resultiert daraus auch eine exakte Reproduzierbarkeit. Jede Kopie wird sich exakt so verhalten, wie es geplant und getestet wurde.

Ein weiterer sehr wichtiger Vorteil digitaler Systeme liegt darin, daß man Objekte fast jeder gewünschten Komplexität aufbauen kann. Will man im Bild der analogen Schaltkreise bleiben, so erfordert bereits das Hinzufügen einzelner Widerstände einen beträchtlichen Aufwand. Digitale Signalverarbeitung kennt solche Beschränkungen kaum. Eine Erhöhung der Komplexität erfordert zumeist nur eine Änderung der zugrundeliegenden Modellparameter – und das ist für einen Computer meist ohne Belang.

Der letzte Abschnitt betrifft gleichzeitig eine weitere Eigenschaft digitaler Systeme. Fast in allen Fällen ist eine Optimierung von gefundenen Lösungen durchführbar. Hierbei sind sogar zwei Stufen von Bedeutung. Durch Parameteranpassung läßt sich zunächst eine maßgeschneiderte Realisierung eines gegebenen Problems aufbauen. Ändern sich im Verlaufe des Einsatzes bestimmte Randbedingungen, so läßt sich in zweiter Instanz basierend auf digitalen Ansätzen auch viel für eine geeignete Adaption tun.

Ein sehr starkes Argument für die Überlegenheit digitaler Ansätze kommt aus der Ecke der Entwickler. Diese haben einen nicht immer ganz einfachen theoretischen Hintergrund letztlich in Hardwareform zu bringen. Die Rolle des Mittlers wird hierbei von Programmiersprachen übernommen, die sozusagen digitale Entwurfssprachen darstellen. Nun ist die Distanz zwischen mathematisch orientierter Theorie und Programmiersprachen oftmals gering, d.h., diese Art der Umsetzung wird stark begünstigt. Das kann auf der anderen Seite von analogen Systemen keinesfalls behauptet werden. Anders gesagt, der Nutzung eines überaus reichhaltigen Reservoirs an Lösungen von Signalverarbeitungsproblemen steht aus Sicht der digitalen Welt nichts entgegen.

Es gibt weitere Vorteile des digitalen Ansatzes: So kann man ggf. präzise arbeitende Speicherelemente in fast jeder gewünschten Größenordnung hinzufügen. Ferner gibt es auch die Möglichkeit, höherdimensionale Signale mit denselben Werkzeugen zu bearbeiten, digitale Systeme sind vielen Leuten vertraut, ein herkömmlicher Computer kann genutzt werden, die Ausbildung wird erleichtert und vieles mehr. Eine vollständige Liste ist mit Sicherheit sehr lang. Als Fazit bleibt: Die Bedeutung der digitalen Signalverarbeitung setzt sich aus zahlreichen Einzelkomponenten zusammen.

Was man weiß, was man wissen sollte!

Ein Leser, der an allen hier vorzustellenden Details interessiert ist, muß über ein gewisses Basiswissen verfügen. Dazu zählen insbesondere folgende Themen:

- Polynome und deren Nullstellen
- Komplexe Zahlen
- Integration und Differentiation von Funktionen
- Einfachste Fitting-Strategien

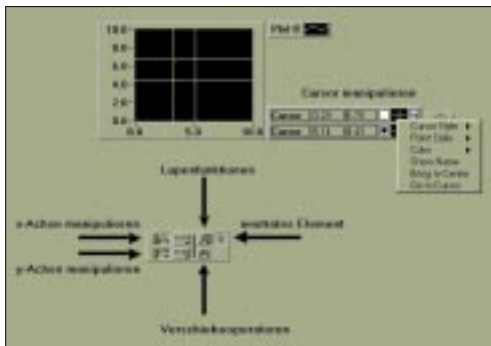


Bild 5. Einige der Hilfsmittel, um die Signaldarstellung zu verbessern. Einen vollständigen Überblick verschafft man sich durch Übung.

Alle weiteren zusätzlich erforderlichen Begriffe werden – so weit nötig – an passender Stelle erläutert.

Werkzeuge erleichtern das Ingenieurleben

Standardwerkzeuge zur Behandlung von Problemen der digitalen Signalverarbeitung gibt es reichlich. Ist man mehr an den theoretischen Hintergründen interessiert, sollte man auf ein Programm wie Mathematica von Wolfram Research zurückgreifen. Zwar fehlen Möglichkeiten zu einem direkten Kontakt mit der realen Welt in Form von Datenerfassungs- und Ausgabeboards, und die Geschwindigkeit der Abarbeitung ist auch nicht gerade hoch, doch die symbolischen Fähigkeiten von Mathematica machen vieles wieder wett. Für Ausbildungszwecke und bei der Entwicklung neuer Verfahren ist dieses Paket definitiv erste Wahl.

Stehen numerische Aspekte im Mittelpunkt, sollte man an ein Programm wie Matlab von The MathWorks denken. Zahlreiche Routinen der digitalen Signalverarbeitung sind integraler Bestandteil von Matlab. Zusätzlich stehen mit Simulink und weiteren Toolboxes Unterstützungen auf so gut wie allen Gebieten der Signalverarbeitung bereit. Einzige Schwachstelle ist die Meßdatenanbindung. Auf diesem Gebiet scheint sich allerdings allerhand zu tun. Meßwertverarbeitungsprogramme wie National Instruments LabVIEW weisen den letztgenannten Mangel nicht auf. Darüber hinaus stehen leistungsstarke Bibliotheken der digitalen Signalverarbeitung bereit, die weite Teile der momentan eingesetzten Theorie zur Verfügung stellen. Derartige Pakete sind sowohl für die Neuentwicklung von Systemen als auch für deren direkte Umsetzung in der Praxis geeignet. Natürlich muß man ggf. noch Anpassungen vornehmen, weil die eigentliche Zielhardware z.B. spezifische DSP-Boards sein können.

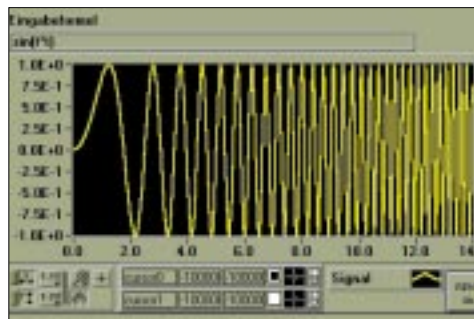


Bild 6. So bereitet man die „Anhörung“ eines Chirp-Signales vor. Die eingesetzten Parameter prägen deutlich den entstehenden Signalverlauf.

Reichhaltiges Literaturangebot

Bücher und Artikel über digitale Signalverarbeitung gibt es reichlich. Man sollte auf eine gute Mischung aus Theorie und Anwendung achten. Diese Mixtur erleichtert wesentlich das Verständnis und die Geschwindigkeit, mit der man sich neue Fertigkeiten aneignet.

Unvermeidlich: Signale und Konventionen

Soweit zu den Vorbereitungen. Was wir jetzt benötigen, ist eine Erläuterung des Signalkonzeptes an sich sowie eine passende Notation, die im Laufe der Serie auch beibehalten werden kann. Die grundlegende Charakterisierung der Signale basiert auf dem Unterschied zwischen diskret und kontinuierlich (*Bild 2*) sowie zwischen analog und digital (*Bild 3*). Die erste Klassifizierung bezieht sich auf den zeitlichen Verlauf der Abtastung. Und hiermit fangen bereits die ersten Konventionen an. Es ist keineswegs immer die Zeit, die als Maßstab dieser Abtastung eingesetzt werden muß. Das kann schon deshalb nicht sein, weil in diesem Fall Strukturen wie 2D-Bilder keinen Platz im Reich der Signalverarbeitung hätten. Zumindest im 1D-Fall ist die Vereinbarung, daß man es mit zeitabhängigen Signalen zu tun hat, sehr bequem und vereinfacht viele Betrachtungen.

In diesem Sinne ist ein kontinuierliches Signal dadurch geprägt, daß zu jedem Zeitpunkt ein Wert in Form einer (zumeist reellen) Zahl vorliegt. Diese Eigenschaft verbietet die direkte Behandlung basierend auf einem Computer. Zu diesem Zweck muß, zeitlich gesehen, diskret gearbeitet werden, d.h., zu fixierten Zeitpunkten werden Signalwerte aufgenommen. Hierbei ist sowohl eine gleichmäßige Tastung als auch eine, zeitlich gesehen, mehr oder weniger regellose Verteilung denkbar.

Ein Beispiel für die erste Situation liegt dann vor, wenn man Audiosignale von einer CD-ROM aufnimmt. In diesem Fall muß man mit einer Abtastfrequenz von 44,1 kHz arbeiten. Börsenkurse unterschiedlichster Art können zuweilen in sogenannter Ticker-Rate anfallen, was durchaus im Sekundenbereich liegen kann. Auf der Gegenseite stehen Zeiten wie nachts, Wochenenden oder Feiertage ohne jegliche Notierung. Verständlicherweise erschwert das die nachfolgenden Verarbeitungsroutinen deutlich. Das hat u.a. dazu geführt, daß für zahlreiche Signalverarbeitungsroutinen jeweils zwei Varianten existieren, eine für äquidistante Abtastung, die andere für ungleichmäßig verteilte Datenaufnahme.

Die zweite Art der Unterscheidung bezieht sich auf charakteristische Eigenschaften der Signalwerte. Aus rein theoretischer Sicht sind analoge Werte erfaßbar. Gemeint ist damit zumeist, daß ein konkretes Signal prinzipiell jeden reellen Wert annehmen kann, d.h., die Genauigkeit ist

beliebig hoch. Technisch gesehen geht das nicht. Man muß mit mehr oder weniger vielen Stellen (Digits) leben, d.h., man hat es mit digitalen Werten zu tun. Auch die Computer selbst sind von diesem Dilemma betroffen. So werden von Programmiersprachen oder anderen Entwicklungssystemen sehr oft „Real“-Formate verschiedener Präzision angeboten, jedoch wird intern das alles auf digitale und endliche Entsprechungen reduziert.

Das bereits zitierte Beispiel der CD-ROM erläutert ganz gut das Wesen der digital orientierten Signale. Standardmäßig wird mit 16 bit gearbeitet, d.h., der gesamte hörbare Intensitätsbereich wird dort untergebracht. Man wird zugeben müssen, daß dieses Ziel durchaus erfolgreich erreicht wurde.

Signalgeneratoren helfen bei der Simulation unterschiedlicher Quellen

Zum Abschluß des ersten Teils der Serie soll ein Werkzeug vorgestellt werden, mit dem man sich Signale aus unterschiedlichen Quellen besorgen kann. Entscheidend ist neben der eigentlichen Datengenerierung deren Darstellung. Das Programm SIGNAL ([1], geschrieben in National Instruments grafischer Sprache LabVIEW) bietet hierzu unterschiedliche Werkzeuge an (*Bild 4*), die in Abhängigkeit vom Verwendungszweck frei gewählt werden können. Zusätzlich sind Hilfsmittel integriert (*Bild 5*), die das Navigieren innerhalb größerer Datenbestände erleichtern. Dazu zählen insbesondere Lupenfunktionen. Die verwendete Symbolik ist weitgehend selbsterklärend, so daß man nur ein wenig Übung benötigt, um damit sicher umgehen zu können.

Als Datenquellen dienen die bereits genannten Verfahren. Die ASCII-Schnittstelle ist hierbei die einfachste. Ein allgemeines Signal muß die in *Tabelle 1* gezeigte Konvention einhalten. Erlaubt ist auch ein Wechsel der Darstellung (E-Formate). Jeder einzelne Wert wird paarweise (Zeitpunkt und eigentlicher Wert) mit einem Zeilenvorschub abgeschlossen. Die Anzahl der Datenpunkte muß dabei nicht vorgegeben werden, da das Programm SIGNAL diesen Wert automatisch bestimmt. Die ASCII-Schnittstelle ist z.B. immer dann vorteilhaft, wenn man Daten aus Programmen wie Excel importieren will. Exporte von Werten mittels der ASCII-Formate sind auch möglich.

Für eine schnelle Signalgenerierung eignet sich am besten die Formeleingabe. *Tabelle 2* gibt einen Überblick

über die einsetzbaren Funktionen. Man beachte, daß jegliche Kombination von Formeln erlaubt ist. So lassen sich aus signaltheoretischer Sicht wichtige Funktionen durch passende Verschachtelung von Einzelkomponenten definieren. Die korrekte Vorgabe von Signalen benötigt neben den zugrundeliegenden Formeln auch Zusatzangaben, die die Anzahl der zu generierenden Werte und das zur Debatte stehende Intervall betreffen. Bei Bedarf ließen sich mit Hilfe des Signalgenerators übrigens auch mehrdimensionale Datensätze generieren.

Schließlich, und das ist ein kaum zu unterschätzender Vorteil, wird eine reale Datenschnittstelle in Form einer Audioanbindung angeboten. Eine der vielen möglichen interessanten Anwendungsfälle ist die Generierung von beispielsweise Sinus- oder Chirp-Signalen. Kombinationen dieser Signaltypen sind selbstverständlich auch möglich. Hat man sich z.B. erst einmal ein Chirp-Signal angehört (*Bild 6* zeigt die dazu erforderliche Grundeinstellung im Programm SIGNAL, [1]), wird man später den Effekt der anschwellenden Frequenzen nicht mehr vergessen. Eine Datenaufnahme über eine standardmäßige Soundblaster-Karte ist ebenfalls möglich. Der Experimentierfreudigkeit sind dann kaum noch Grenzen gesetzt. Das alles fördert deutlich das Verständnis für die einzuführenden Konzepte. Ein Anhören der Resultate kommt dann oft dem Begreifen des Gegenstandes sehr nahe. gs

Der zweite Teil der Serie behandelt die Fouriertransformation.

Literatur

[1] Das Programm SIGNAL ist downloadbar unter: www.natinst.com/germany

0,0	1,234
0,1	6,456E2
0,2	-1,2E-3
0,3	1,11111
0,4	-2,345
0,5	6E+1

Tabelle 1. Die von der ASCII-Schnittstelle genutzte Konvention ist denkbar einfach. Man vermeide ein RETURN nach allen Wertepaaren, da das alle Paar (0,0) interpretiert wird.

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

Dr. Lothar Wenzel ist gebürtiger Berliner und hat Mathematik und Informatik in Greifswald und Dresden studiert. Nach Tätigkeiten in einem Kernkraftwerk und bei der BASF AG, Ludwigshafen, beschäftigt er sich z.Zt. an der University of Texas mit Fragen der grafischen Programmierung von komplexen Meßdatenverarbeitungs- und Simulationsproblemen.

