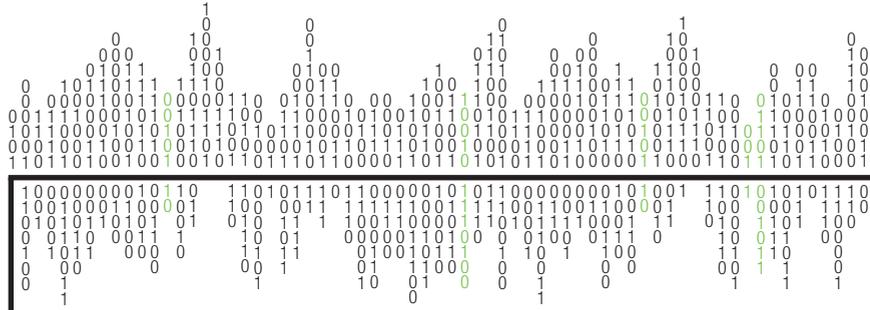


Fernando Puente León · Sebastian Bauer



PRAXIS DER DIGITALEN SIGNALVERARBEITUNG

2. AUFLAGE

Fernando Puente León, Sebastian Bauer

PRAXIS DER DIGITALEN SIGNALVERARBEITUNG

2., überarbeitete Auflage

Praxis der Digitalen Signalverarbeitung

von

Fernando Puente León, Sebastian Bauer

2., überarbeitete Auflage

Impressum



Karlsruher Institut für Technologie (KIT)
KIT Scientific Publishing
Straße am Forum 2
D-76131 Karlsruhe

KIT Scientific Publishing is a registered trademark of Karlsruhe
Institute of Technology. Reprint using the book cover is not allowed.

www.ksp.kit.edu



*This document – excluding the cover, pictures and graphs – is licensed
under the Creative Commons Attribution-Share Alike 4.0 International License
(CC BY-SA 4.0): <https://creativecommons.org/licenses/by-sa/4.0/deed.en>*



*The cover page is licensed under the Creative Commons
Attribution-No Derivatives 4.0 International License (CC BY-ND 4.0):
<https://creativecommons.org/licenses/by-nd/4.0/deed.en>*

Print on Demand 2017 – Gedruckt auf FSC-zertifiziertem Papier

ISBN 978-3-7315-0651-5

DOI 10.5445/KSP/1000067012

Vorwort zur 2. Auflage

Das vorliegende Lehrbuch wendet sich an Studierende der Ingenieurwissenschaften (insbesondere der Elektrotechnik, Informationstechnik, Mechatronik und des Maschinenbaus) und der Informatik sowie an Ingenieure in der Praxis. Es behandelt sowohl grundlegende als auch fortgeschrittene Aspekte des spannenden Gebiets der digitalen Signalverarbeitung. Es fußt auf dem Modul „Praktikum Digitale Signalverarbeitung“ der Autoren am Karlsruher Institut für Technologie (KIT, ehemals Universität Karlsruhe (TH)). Das Buch ist in acht Versuche gegliedert, in welchen die folgenden Inhalte vermittelt werden:

- Einführung in Matlab und LabVIEW,
- Korrelationsmesstechnik,
- digitale Messdatenerfassung,
- analoge und digitale Filterung,
- Akustik und Sprachsignalverarbeitung,
- Zustandsschätzung mittels Kalman-Filterung,
- Modalanalyse sowie
- Bildverarbeitung.

Vorausgesetzt werden Kenntnisse der gebräuchlichen Integraltransformationen und der Grundlagen der Wahrscheinlichkeitsrechnung. Das Buch ist so konzipiert, dass sich der Leser den Stoff kapitelweise leicht selbst aneignen und weiterführende Literatur erschließen kann. Zahlreiche Beispiele und Übungsaufgaben illustrieren die dargestellten Methoden und tragen somit zum besseren Verständnis und zur Vertiefung der Inhalte bei.

In der vorliegenden überarbeiteten und didaktisch verbesserten Auflage wurde die bewährte Struktur des Buches beibehalten. Dennoch wurden zahlreiche Textpassagen erweitert, neue Beispiele hinzugefügt und etliche Abbildungen überarbeitet, um eine bessere Verständlichkeit sowie eine konsistentere Verwendung der mathematischen Symbole zu erzielen. Bei dieser Gelegenheit wurde auf eine einheitliche Darstellung zeitdiskreter Signale geachtet.

Darüber hinaus konnten durch Hinweise von Lesern Fehler korrigiert und Erläuterungen verbessert werden; ihnen allen sei an dieser Stelle herzlich gedankt. Schließlich danken wir dem Verlag KIT Scientific Publishing für die wiederum ausgezeichnete Zusammenarbeit.

Karlsruhe, im Frühjahr 2017

Fernando Puente León
Sebastian Bauer

Inhaltsverzeichnis

1	Einführung in Matlab und LabVIEW	1
1.1	Einleitung	3
1.2	Einführung in Matlab	3
1.2.1	Bedienungselemente	4
1.2.2	Allgemeines und Dokumentation	5
1.2.3	Vektoren	5
1.2.4	Matrizen	8
1.2.5	Skripte und Funktionen	9
1.2.6	for-Schleife	10
1.2.7	Visualisierung von Daten	11
1.2.8	Operatoren und vordefinierte Konstanten	11
1.2.9	Befehlsübersicht	12
1.3	Einführung in Simulink	12
1.3.1	Allgemeines	12
1.3.2	Bedienung	14
1.4	Einführung in LabVIEW	16
1.4.1	Allgemeines zu LabVIEW-Programmen	16
1.4.2	Bedienung und Programmierung von VIs	17
1.4.3	Erstellen eines VIs	23
1.4.4	Weitere Programmstrukturen	28
1.5	Versuchsvorbereitende Aufgaben	32
1.6	Versuchsaufgaben	32
1.7	Literatur	42
2	Korrelationsmesstechnik	43
2.1	Momente und Korrelationsfunktionen	45
2.1.1	Korrelationsfunktionen stationärer Prozesse	47
2.1.2	Korrelationsfunktionen ergodischer Prozesse	47
2.2	Signalklassen	47
2.3	Messung von Korrelationsfunktionen	48
2.4	Korrelationsfunktion und Leistungsdichte	49
2.5	Korrelation abgetasteter Signale	50
2.6	Schätzung von AKF und KKF mittels der FFT	53
2.7	Polaritätskorrelation	55
2.8	Versuchsdurchführung	55

2.8.1	Erfassung physikalischer Größen in Matlab	56
2.8.2	Einstellungen der Programmoberfläche	56
2.9	Versuchsvorbereitende Aufgaben	59
2.10	Versuche an der Förderbandanlage	60
2.11	Literatur	65
3	Digitale Messdatenerfassung	67
3.1	Einleitung	69
3.2	Zeitdiskretisierung	69
3.2.1	Abtastung	69
3.2.2	Abtasttheorem	70
3.2.3	Anti-Aliasing-Filter	70
3.2.4	Mittelwertbildung bei endlicher Abtastdauer	71
3.2.5	Zeitliche Abtastfehler (Jitter)	73
3.2.6	Quantisierung analoger Signale	73
3.3	Diskrete Signale im Frequenzbereich	77
3.3.1	Die Fourier-Transformation kontinuierlicher Signale	77
3.3.2	Fourier-Transformation zeitdiskreter Signale	78
3.3.3	Diskrete Fourier-Transformation (DFT)	79
3.4	Verwendete Hardware und ihre Ansteuerung	83
3.4.1	Signalerfassung	84
3.4.2	Signalauswertung und Signalpräsentation	85
3.5	Versuchsvorbereitende Aufgaben	86
3.6	Versuchsaufgaben	87
3.7	Literatur	95
4	Analoge und digitale Filterung	97
4.1	Einleitung	99
4.2	Theorie	100
4.2.1	Grundlagen	100
4.2.2	Anwendungsgebiete von Filtern	106
4.2.3	Analoge Filter	108
4.2.4	Digitale Filter	111
4.2.5	Auswirkungen endlicher Rechengenauigkeit	120
4.3	Versuchsaufbau	125
4.3.1	Beschreibung des Versuchs	125
4.3.2	Versuchsvorbereitende Aufgaben	133
4.3.3	Versuche am Versuchstag	135
4.4	Literatur	138

5	Akustik und Sprachsignalverarbeitung	139
5.1	Einleitung	143
5.2	Grundlagen	143
5.2.1	Skalierung	144
5.2.2	Momente von Energiesignalen	144
5.3	Fourier-Transformation und Leistungsdichte	146
5.3.1	Diskrete Fourier-Transformation (DFT)	146
5.3.2	Kurzzeit-Fourier-Transformation	153
5.3.3	Zeitdiskretisierung	154
5.4	Wavelet-Transformation	156
5.4.1	Kontinuierliche Wavelet-Transformation	156
5.4.2	Diskrete Wavelet-Transformation	159
5.4.3	Wavelet-Rekonstruktion	161
5.5	Wigner-Ville-Verteilung	162
5.5.1	Ambiguitätsfunktion	162
5.5.2	Definition der Wigner-Ville-Verteilung	162
5.5.3	Eigenschaften der Wigner-Ville-Verteilung	164
5.5.4	Kreuzterme	166
5.5.5	Pseudo-Wigner-Ville-Verteilung	167
5.5.6	Geglättete Pseudo-Wigner-Ville-Verteilung	167
5.5.7	Zeitdiskrete Wigner-Ville-Verteilung	168
5.5.8	Zusammenhang zwischen Wigner-Ville-Verteilung und STFT	168
5.6	Sprachsignale	169
5.6.1	Sprachproduktion	169
5.6.2	Eigenschaften	170
5.6.3	Formanten	171
5.7	Schallausbreitung in begrenzten Räumen	171
5.7.1	Signalmodelle	171
5.7.2	Raumimpulsantworten	172
5.8	Laufzeit- und Richtungsschätzung	172
5.8.1	Laufzeitschätzung	173
5.8.2	Ermittlung der Schalleinfallrichtung	174
5.9	Versuchsaufbau und Matlab-Umgebung	175
5.9.1	Versuchsaufbau	175
5.9.2	Matlab-Umgebung	175
5.10	Versuchsvorbereitende Aufgaben	176
5.11	Versuche am Versuchstag	177
5.12	Literatur	184

6	Multisensorielle Lokalisierung eines Modellfahrzeuges mittels Kalman-Filterung	185
6.1	Einleitung	187
6.2	Lokalisierung des Modellfahrzeuges	189
6.2.1	Die Sensoren des Fahrzeugs	189
6.2.2	Die zeitdiskreten Positionsgleichungen	190
6.2.3	Auswertung der Inkrementalgeberdaten	193
6.2.4	Auswertung der Beschleunigungssensordaten	195
6.2.5	Auswertung der Gierratensensordaten	197
6.2.6	Fehlerursachen bei der Lokalisierung	197
6.3	Datenfusion durch Kalman-Filterung	200
6.3.1	Zeitdiskrete Beschreibung linearer Systeme	200
6.3.2	Das Kalman-Filter	203
6.3.3	Multisensor-Datenfusion mit dem Kalman-Filter	208
6.4	Die Versuchssoftware	212
6.4.1	Das Programm auf dem Fahrzeugrechner	213
6.4.2	Das Programm auf dem Praktikumsrechner	213
6.4.3	Vorgehensweise bei der Datenaufnahme	216
6.5	Versuchsvorbereitende Aufgaben	216
6.6	Versuche mit dem Fahrzeug	217
6.6.1	Eingangsbemerkungen	217
6.6.2	Messungen nur mit den Sensoren	218
6.7	Literatur	223
7	Modalanalyse	225
7.1	Einleitung	227
7.2	Theorie der Modalanalyse	228
7.2.1	Voraussetzungen für die Modaltransformation	228
7.2.2	Ausführung der Modaltransformation	229
7.2.3	Berechnung des Frequenzgangs	233
7.2.4	Einfluss verschiedener Dämpfungsansätze auf die Durchführung der Modaltransformation	236
7.3	Experimentelle Modalanalyse	237
7.3.1	Identifikation	237
7.3.2	Frequenzgang	237
7.3.3	Anregungssignale	238
7.3.4	Praktische Durchführung der Modalanalyse	244
7.3.5	Parameterschätzung	245
7.4	Versuchsdurchführung	250

- 7.4.1 Hardware 251
- 7.4.2 Messdatenaufnahme in LabVIEW 251
- 7.4.3 Eigentliche Modalanalyse 253
- 7.4.4 Nützliche Matlab-Befehle 256
- 7.5 Versuchsvorbereitende Aufgaben 257
- 7.6 Versuchsaufgaben 258
 - 7.6.1 Messanweisungen 258
 - 7.6.2 Messungen und Versuchsdurchführung 259
- 7.7 Literatur 262
- 8 Bildverarbeitung 263**
- 8.1 Einleitung 265
- 8.2 Grundlagen der Bildverarbeitung 265
 - 8.2.1 Einführung 266
 - 8.2.2 Bildgewinnung 266
 - 8.2.3 Bilddarstellung in Ortsbereich und Ortsfrequenzbereich 270
 - 8.2.4 Bildvorverarbeitung und Merkmalsextraktion 273
 - 8.2.5 Segmentierung und Klassifikation 287
- 8.3 Realisierung in LabVIEW 289
 - 8.3.1 Kurzbeschreibung NI-VISA für RS232 289
 - 8.3.2 Bildverarbeitung mit IMAQ Vision for LabVIEW 291
- 8.4 Versuchsaufbau 293
- 8.5 Versuchsvorbereitende Aufgaben 293
- 8.6 Aufgaben am Versuchstag 295
- 8.7 Literatur 300
- Index 303**

Versuch 1

Einführung in Matlab und LabVIEW

1	Einführung in Matlab und LabVIEW	
1.1	Einleitung	3
1.2	Einführung in Matlab	3
1.2.1	Bedienungselemente	4
1.2.2	Allgemeines und Dokumentation	5
1.2.3	Vektoren	5
1.2.4	Matrizen	8
1.2.5	Skripte und Funktionen	9
1.2.6	for-Schleife	10
1.2.7	Visualisierung von Daten	11
1.2.8	Operatoren und vordefinierte Konstanten	11
1.2.9	Befehlsübersicht	12
1.3	Einführung in Simulink	12
1.3.1	Allgemeines	12
1.3.2	Bedienung	14
1.3.2.1	Subsystem	15
1.3.2.2	Ausgabe von Daten an Matlab	15
1.4	Einführung in LabVIEW	16
1.4.1	Allgemeines zu LabVIEW-Programmen	16
1.4.2	Bedienung und Programmierung von VIs	17
1.4.2.1	Frontpanel	17
1.4.2.2	Blockdiagramm	18
1.4.2.3	Express-VI	18
1.4.2.4	Elemente- und Funktionenpalette	20
1.4.2.5	Werkzeugpalette	21
1.4.2.6	Symbolleiste	21
1.4.3	Erstellen eines VIs	23
1.4.3.1	Datenflussprogrammierung	27
1.4.3.2	Fehlersuche	27
1.4.4	Weitere Programmstrukturen	28
1.4.4.1	Schleifen	28
1.4.4.2	Diagramme und Graphen	30
1.5	Versuchsvorbereitende Aufgaben	32
1.6	Versuchsaufgaben	32
1.7	Literatur	42

1 Einführung in Matlab und LabVIEW

1.1 Einleitung

In diesem Versuch werden die Programmiersysteme *Matlab/Simulink* der Firma *MathWorks* und *LabVIEW* von *National Instruments* vorgestellt. Damit erhält der Praktikumssteilnehmer einen Überblick über die derzeit führenden Signalverarbeitungsprogramme, auf denen die nachfolgenden Praktikumsversuche basieren.

Im Abschn. 1.2 wird eine kurze Einführung in *Matlab* gegeben, die für diejenigen gedacht ist, die noch nie mit *Matlab* gearbeitet haben. Das Gleiche gilt für die Abschnitte 1.3 und 1.4, in welchen *Simulink* bzw. *LabVIEW* vorgestellt werden. Die Aufgaben in Abschn. 1.6 sind am ersten Praktikumstermin zu lösen und dienen der Einarbeitung in *Matlab*, *Simulink* und *LabVIEW*. Die versuchsvorbereitenden Aufgaben in Abschn. 1.5 müssen vor dem Versuch gelöst werden, um mit der zu bearbeitenden Thematik vertraut zu werden.

1.2 Einführung in Matlab

Der Name *Matlab* ist eine Abkürzung für *MATRIX LABORATORY* und bezeichnet eine Software für mathematisch-technische Anwendungen, die aus LINPACK/EISPACK-Programmibibliotheken¹ entstanden ist. *Matlab* ist eine interaktive und programmierbare Umgebung für numerische Berechnungen sowie zur Datenanalyse und -visualisierung. Die Bandbreite der Anwendungen umfasst u. a. die Bereiche der

- Signalverarbeitung,
- Kommunikationstechnik,
- Bild- und Videoverarbeitung,
- Steuerungs- und Regelungssysteme,
- Messtechnik,
- Finanzmathematik und
- Bioinformatik.

Vorteilhaft an *Matlab/Simulink* sind die vielfältigen Einsatzgebiete und Programmiermöglichkeiten. Diese „Mächtigkeit“ kann für den Einsteiger erdrückend wirken, ist jedoch für Experten hilfreich bei der Lösung zahlreicher Aufgaben. Einschränkungen ergeben sich u. a. bei der Messdatenerfassung über einen PC, da *Matlab/Simulink* vornehmlich zur numerischen Berechnung und zur Simulation von Modellen

¹LINPACK und EISPACK sind Open-Source-Programmibibliotheken zur Lösung linearer Gleichungssysteme bzw. zur Berechnung von Eigenwerten und Eigenvektoren von Matrizen.

entwickelt wurde. Abhilfe wird hier durch das Rapid-Prototyping geschaffen (vgl. Versuch 4).

1.2.1 Bedienungselemente

Nach dem Start von *Matlab* öffnet sich das Hauptfenster (Abb. 1.1). Im Folgenden werden die einzelnen Fensterbereiche kurz vorgestellt. Dabei wird von dem Standard-Layout ausgegangen.

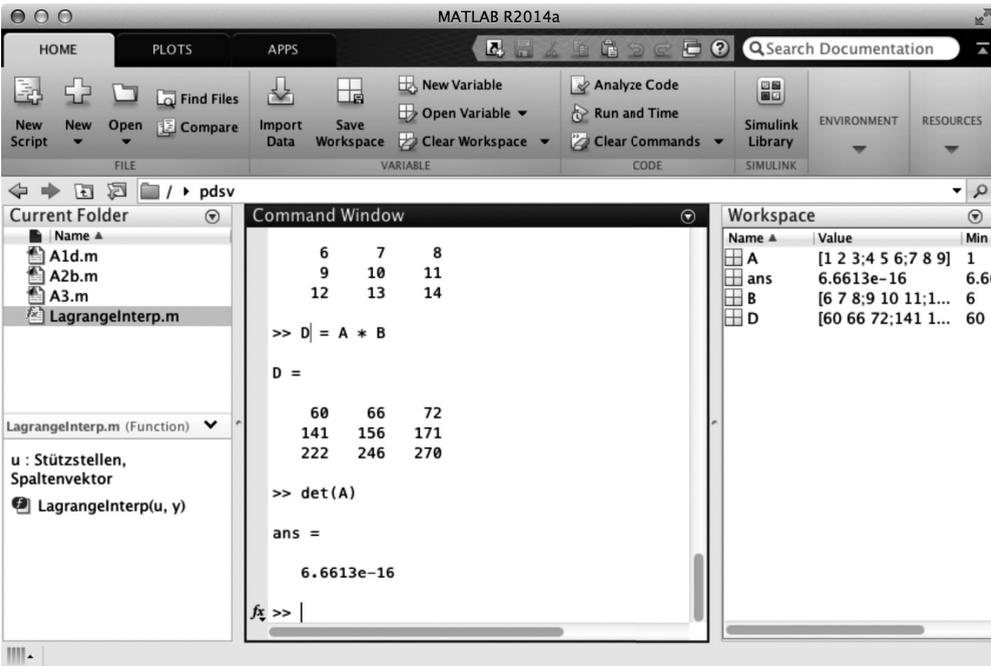


Abbildung 1.1. Matlab-Hauptfenster.

a) Command Window:

In der Mitte finden Sie das Command Window. Hier können *Matlab*-Befehle direkt eingegeben werden, die dann unmittelbar ausgeführt werden. Die Ergebnisse der durchgeführten Berechnungen werden im Workspace abgelegt und im Command Window ausgegeben. Der Prompt » signalisiert die Eingabebereitschaft. Jede Eingabe wird mit Enter abgeschlossen.

b) Workspace:

Im rechten Fenster werden alle im Workspace existierenden Variablen, Funktionen etc. mit ihrem Namen, Dimension, Größe in Bytes und dem Datentyp angezeigt. Zusätzlich lassen sich diese Daten speichern und Variablen mittels des Array-Editors einfach verändern.

c) **Current Folder:**

Mittels des Current Folder-Browsers lässt sich das aktuelle Arbeitsverzeichnis einsehen oder ändern, Dateien öffnen, Verzeichnisse erstellen und andere Datei-Manager-Funktionen ausführen.

d) **Command History:**

In der Command History (Pfeiltaste nach oben) werden alle im Command Window eingegebenen Befehle gespeichert und angezeigt. Durch Doppelklick können die Befehle wiederholt werden; darüber hinaus lassen sich einzelne oder mehrere Befehle ausschneiden, kopieren oder löschen.

e) **Editor:**

Mit dem Editor können Skripts, Funktionen und Klassen erstellt und bearbeitet werden. Der Editor kann entweder über die Schaltfläche New Script oder durch Eingabe von `edit` im Command Window gestartet werden.

1.2.2 Allgemeines und Dokumentation

Matlab verfügt über eine sehr umfangreiche Online-Dokumentation, die auch für erfahrene Anwender unerlässlich ist. Man findet dort neben der eigentlichen Dokumentation (Schnittstellen etc.) auch zusätzliche Informationen über *Matlab*-Befehle. Zur Dokumentation gelangen Sie entweder durch Eingabe von `doc` nach dem Prompt » oder durch Eingabe von `help FUNKTIONSNAME`. Letzteres liefert u. a. Informationen über die Schnittstellen der jeweiligen Funktion.

Der wichtigste Datentyp in *Matlab* ist die Matrix (eine Zahl ist eine 1×1 -Matrix, ein Vektor eine $1 \times n$ - oder $n \times 1$ -Matrix).

Achtung: Bei Variablenbezeichnungen wird zwischen Groß- und Kleinschreibung unterschieden: `M` und `m` sind zwei verschiedene Variablen!

1.2.3 Vektoren

Bei einer Eingabe von

```
>> a=[1 2 3 4 5 6 7]
```

zeigt *Matlab* nach dem Prompt » Folgendes an:

```
a =
    1  2  3  4  5  6  7
```

Bei Vektoren mit vielen Elementen will man die Ausgabe unterdrücken. Dies geschieht durch ein Semikolon am Ende der Zeile:

```
>> a=[1 2 3 4 5 6 7];
```

Eine Addition eines Vektors und eines Skalars

```
>> b = a + 5
```

ergibt:

```
b =
    6  7  8  9 10 11 12
```

Da die Vektoren *a* und *b* gleich groß sind, kann man sie addieren:

```
>> c = a + b
c =
    7  9 11 13 15 17 19
```

oder elementweise multiplizieren:

```
>> d = a .* b
d =
    6 14 24 36 50 66 84
```

Der Punkt vor dem Operationszeichen veranlasst *Matlab*, die angegebene Operation elementweise auszuführen.

Ein Vektor kann wie folgt transponiert² werden:

```
>> c = c'
c =
    7
    9
   11
   13
   15
   17
   19
```

Das Skalarprodukt zweier Vektoren erhält man durch:

```
>> a*c
ans =
   420
```

Die Variable *ans* (*answer*) wird vom Programm automatisch zugewiesen, wenn keine andere Variablenbezeichnung verwendet wurde, z. B.:

```
>> sin(pi/2)
ans =
    1
```

Mit *ans* kann weiter gerechnet werden:

```
>> ans*2
ans =
    2
```

Die Variablen werden im *Workspace* gehalten, bis sie mit *clear* gelöscht werden. Der Inhalt des *Workspace* kann mit *who* oder ausführlicher *whos* angesehen werden.

²Bei komplexen Zahlen erfolgt zusätzlich eine komplexe Konjugation.

Vektoren mit äquidistanten Elementen können wie folgt erzeugt werden:

```
>> f=0:10
f =
    0  1  2  3  4  5  6  7  8  9 10
```

oder mit einer vorgebbaren Schrittweite:

```
>> g=10:2:20
g =
   10  12  14  16  18  20
```

Einzelne Elemente eines Vektors werden über Indizes angesprochen:

```
>> f(5)
ans =
    4
```

und für Intervalle:

```
>> f(2:4)
ans =
    1  2  3
```

Einzelne Elemente können auch ersetzt oder gelöscht werden:

```
>> f(5)=100
f =
    0  1  2  3 100  5  6  7  8  9  1
```

oder

```
>> f(1:3)=[1 1 1]
f =
    1  1  1  3 100  5  6  7  8  9 10
```

Mit

```
>> f(4:10)=[];
```

können die Elemente 4 bis 10 entfernt werden:

```
f =
    1  1  1 10
```

Mittels eckiger Klammern [] können Vektoren aneinandergehängt bzw. zu Matrizen zusammengesetzt werden:

```
>> k=[1 2 3];
>> l=[k k k]
l =
    1  2  3  1  2  3  1  2  3
```

oder

```
>> M =[k; k; k]
```

```
M =
    1  2  3
    1  2  3
    1  2  3
```

1.2.4 Matrizen

Matlab bietet umfangreiche Möglichkeiten der Matrizenrechnung und -manipulation, die analog zu den Vektorfunktionen verwendet werden. Einzelne Elemente oder Submatrizen der Matrix *M* können wie folgt angesprochen werden:

```
>> M(2,2)
ans =
    2

>> M(2:3,2:3)
ans =
    2  3
    2  3
```

Ganze Zeilen oder Spalten werden mit Hilfe des Doppelpunkts `:` angesprochen:

```
>> M(2,:)
ans =
    1  2  3
```

Spezielle Funktionen für Matrizen sind z. B. die Diagonale:

```
>> diag(M)
ans =
    1
    2
    3
```

und die Inverse:

```
>> A=[1 2 3;5 7 6;1 4 6]
A =
    1  2  3
    5  7  6
    1  4  6

>> inv(A)
ans =
    2.0000  0.0000 -1.0000
   -2.6667  0.3333  1.0000
    1.4444 -0.2222 -0.3333
```

```
>> A*inv(A)
ans =
    1.0000    0.0000    0.0000
    0.0000    1.0000    0.0000
    0.0000    0.0000    1.0000
```

Eine 3×3 -Matrix mit standardnormalverteilten Zufallszahlen erhält man durch Eingabe von:

```
>> randn(3)
ans =
   -0.5033   -1.0711    0.5333
    0.0915   -0.1636    0.4024
   -1.0127   -0.4402   -1.3607
```

Im Folgenden wird auf die Darstellung der *Matlab*-Ausgabe verzichtet.

Matrizen lassen sich in bekannter Weise addieren, subtrahieren und multiplizieren, etwa:

```
>> C = A + B - A * B
```

Ferner kann man eine Matrix mit einem Skalar multiplizieren, ein Skalar zu jeder Komponente addieren/subtrahieren oder eine Matrix transponieren:

```
>> 2*A
...
>> A+3
...
>> A'
```

Auch bei Matrizen gibt es die **Punktoperationen**, die elementweise berechnet werden:

```
>> C = A .* B
```

Die Elemente einer Matrix können auch einzeln angesprochen werden:

```
>> A(1,1) = A(1,2) + 2
```

In *Matlab* sind zwei Arten der Division für Matrizen vorgesehen:

a) **Linksdivision:**

$X = A \setminus B$ stellt die Lösung der Matrixgleichung $A \cdot X = B$ dar. **A** und **B** müssen die gleiche Anzahl von Zeilen haben. **X** hat dann die gleiche Zahl von Spalten wie **B**.

b) **Rechtsdivision:**

$X=B/A$ ist die Lösung der Gleichung $X \cdot A = B$. **A** und **B** müssen die gleiche Spaltenanzahl haben. **X** hat die gleiche Zeilenanzahl wie **B**.

1.2.5 Skripte und Funktionen

Ein Grund für die weite Verbreitung von *Matlab* ist die sehr einfache Erstellung von *Matlab*-Programmen. Es gibt drei Arten:

- **Skripte** sind Textdateien, die eine Folge von *Matlab*-Befehlen enthalten. Sie können mit dem Editor (Eingabe von `edit name.m`) erstellt und mit einem Namen der Form `name.m` als *m-File* abgespeichert werden. Bei Eingabe des Dateinamens `name` (ohne `.m`) wird dann die Folge der *Matlab*-Befehle abgearbeitet. An Skripte lassen sich keine Parameter übergeben. Definierte Variablen werden im globalen Workspace abgelegt, auf welchen auch vom Skript aus zugegriffen werden kann. Der Inhalt eines *m-Files* kann mit `type name` angezeigt werden.
- **Funktionen** werden wie Skripte erzeugt, abgespeichert und aufgerufen. Es lassen sich aber Parameter übergeben sowie lokale und globale Variablen und lokale Unterprogramme erzeugen.
- **Klassen** erlauben eine objektorientierte Programmierung in *Matlab*, indem u. a. Daten und Operationen verknüpft werden.

1.2.6 for-Schleife

Die *for*-Schleife wird für eine feste Anzahl von Durchläufen von *Matlab*-Statements benutzt. Die Syntax lautet:

```
for index=Anfangswert:Schrittweite:Endwert
    Matlab-Statements
    ...
end
```

Die Schrittweite kann positiv oder negativ gewählt werden. Ihre Angabe ist optional. Wird sie nicht angegeben, verwendet *Matlab* die Schrittweite eins. Der Index kann in der Schleife verwendet werden, ist beim Start gleich dem Anfangswert und wird nach jedem Schritt um die Schrittweite verändert. Ist die Schrittweite positiv, wird die Schleife solange durchlaufen, bis der Index den Endwert übersteigt. Beispielweise wird bei folgender Eingabe:

```
x=1;
for i=1:1.1:3
    x=x+i;
end
```

die Schleife zweimal ausgeführt. Die Variable `x` tritt mit dem Wert 1 in die Schleife ein, wird dort im ersten Durchlauf auf 2 und nach dem zweiten auf 4,1 erhöht. Der Schleifenindex `i` hat dann den Wert 2,1.

Ist die Schrittweite negativ, wird die Schleife solange durchlaufen, bis der Index kleiner als der Endwert ist. In sämtlichen Fällen existiert der Schleifenindex auch nach Durchlauf der Schleife.

1.2.7 Visualisierung von Daten

In *Matlab* stehen umfangreiche Möglichkeiten zur Visualisierung von Daten zur Verfügung – angefangen von einfachen x,y -Plots bis hin zu Animationen. Eine typische Befehlsfolge lautet:

```
x=0:0.1:2*pi;           % Festlegen x-Wertebereich
y1=sin(x);              % Berechnen der sin-Werte
y2=cos(x);              % Berechnen der cos-Werte
plot(x,y1,x,y2,'--');   % Plot der x- und y-Werte,
                        % y2 gestrichelt zeichnen
title('Sinus und Cosinus'); % Einfügen eines Titels
xlabel('X');             % x-Achsenbeschriftung
ylabel('Y');            % y-Achsenbeschriftung
legend('sin','cos');    % Einfügen einer Legende
```

Das Ergebnis dieser Befehlsfolge sehen Sie in Abb. 1.2. Die Legende kann mit der Maus an eine beliebige Position verschoben werden.

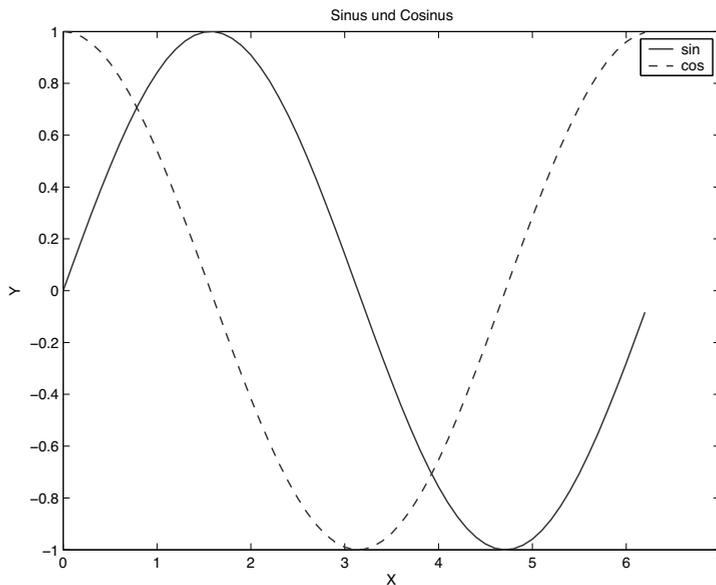


Abbildung 1.2. Beispielplot einer Sinus- und einer Cosinus-Funktion.

1.2.8 Operatoren und vordefinierte Konstanten

Matlab kennt die arithmetischen Standardoperatoren und deren Ausführungsregeln, die für Zahlen (Skalare) und Matrizen gelten (Tab. 1.1). Sollen die Operationen elementweise auf Matrizen angewendet werden, verwendet man die Punktoperatoren (Tab. 1.2). Schließlich zeigt Tab. 1.3 in *Matlab* vordefinierte Konstanten.

Tabelle 1.1. *Matlab*-Standardoperatoren.

Standardoperatoren	
+	Addition und Matrixaddition
-	Subtraktion und Matrixsubtraktion
*	Multiplikation und die nichtkommutative Matrixmultiplikation
/, \	Division und die Matrixrechts- bzw. Matrixlinksdivision
^	Exponentiation und Matrixexponentiation
'	Komplex konjugierte, transponierte Matrix

Tabelle 1.2. *Matlab*-Punktoperatoren.

Punktoperatoren	
.*	Elementweise Multiplikation zweier Matrizen
./, \	Elementweise Division zweier Matrizen
.^	Elementweise Exponentiation zweier Matrizen
.'	Nicht komplex konjugierte, transponierte Matrix

Tabelle 1.3. In *Matlab* vordefinierte Konstanten.

Vordefinierte Konstanten	
pi	3,14159265...
j, i	Imaginäre Einheit, $\sqrt{-1}$
eps	2^{-52} Abstand der nächsten Fließkommazahl von 1.0, wird von <i>Matlab</i> -Funktionen zur Toleranzermittlung benutzt. eps kann überschrieben (z. B. <code>eps=1.e-10</code>) und zurückgesetzt werden (<code>clear eps</code>).
inf	Unendlich, Ergebnis wenn eine von null verschiedene Zahl durch null geteilt wird
nan	Mathematisch nicht definierte Zahl, z. B. $0/0$

1.2.9 Befehlsübersicht

In Tab. 1.4 sind weitere Befehle abgedruckt, die für die Lösung der gestellten Aufgaben hilfreich sind. Für weitere Informationen nutzen Sie die *Matlab*-Hilfe.

1.3 Einführung in Simulink

1.3.1 Allgemeines

Simulink ist ein Zusatz-Softwarepaket zu *Matlab*, das die Modellierung, Simulation und Analyse dynamischer Systeme ermöglicht. Das zu untersuchende System kann

Tabelle 1.4. Matlab-Befehlsübersicht.

<code>clear</code>	Löscht den Workspace
<code>clock</code>	Gibt das aktuelle Datum und die aktuelle Uhrzeit aus
<code>close all</code>	Schließt alle Plots
<code>colormap(gray)</code>	Das aktuelle Bild wird mit Grauwerten dargestellt
<code>det(A)</code>	Berechnung der Determinante von A
<code>etime(t1,t2)</code>	Berechnet die verstrichene Zeit zwischen den Zeitpunkten t_1 und t_2
<code>eval(g)</code>	Führt einen <i>Matlab</i> -String aus, z. B. „ $m*x+b$ “, s. Aufgabe 1
<code>filter2(a,b,option)</code>	Filtert die Daten in b mit dem digitalen Filter a
<code>for</code>	Siehe Abschnitt 1.2.6 auf S. 10
<code>grid</code>	Erzeugt Gitterlinien in einem Plot
<code>hold</code>	Aktueller Plot wird nicht überschrieben
<code>image(F)</code>	Stellt das Bild F graphisch dar
<code>legend('a','b')</code>	Erzeugt eine Legende im aktuellen Plot
<code>length(x)</code>	Berechnet die Länge des Vektors x
<code>load name.mat</code>	Lädt die Datei <code>name.mat</code> in den Workspace
<code>ones(m,n)</code>	Erzeugt eine $m \times n$ -Matrix mit lauter Einsen
<code>plot(x,y)</code>	Erzeugt eine graphische Darstellung
<code>randi(range,m,n)</code>	Erzeugt eine $m \times n$ -Matrix mit Zufallszahlen, die im Bereich <code>range</code> liegen
<code>save name variable</code>	Speichert die Variable <code>variable</code> in der Datei <code>name.mat</code> ab
<code>sim('model')</code>	Simuliert ein <i>Simulink</i> -Modell
<code>subplot(m,n,k)</code>	Unterteilt das <code>figure</code> -Fenster in m Zeilen und n Spalten und fügt bei k den nachfolgenden Plot ein
<code>sum(x)</code>	Bildet die Summe der Elemente des Vektors x
<code>title('Titel')</code>	Fügt die Überschrift „Titel“ in ein Plot ein
<code>tic</code>	Startet den Stoppuhr-Timer
<code>toc</code>	Lesen des Stoppuhr-Timers und Ausgabe der seit dem Aufruf von <code>tic</code> verstrichenen Zeit
<code>xcorr(x,y,option)</code>	Schätzen der Kreuzkorrelationsfolge von den Vektoren x und y
<code>xlabel('Time')</code>	Beschriftet die x -Achse des aktuellen Plots mit „Time“
<code>ylabel('Value')</code>	Beschriftet die y -Achse des aktuellen Plots mit „Value“
<code>zeros(m,n)</code>	Erzeugt eine $m \times n$ -Matrix, bei der alle Elemente gleich null sind

mit Hilfe einer graphischen Oberfläche aufgebaut werden. Die Simulation kann wahlweise zeitkontinuierlich, zeitdiskret oder hybrid durchgeführt werden. Die Simulati-

onsergebnisse lassen sich über Anzeigen direkt in *Simulink* betrachten. Über Schnittstellen ist ebenfalls eine Auswertung in *Matlab* möglich.

Eine Modellierung komplexer Abläufe ist über sogenannte *s-Functions* möglich. Sie ermöglichen die Einbindung von Programmen in die Simulationsumgebung.

1.3.2 Bedienung

Wird *Simulink* über die Eingabe `simulink` am *Matlab*-Prompt gestartet, so erscheint der Library-Browser (siehe Abb. 1.3).

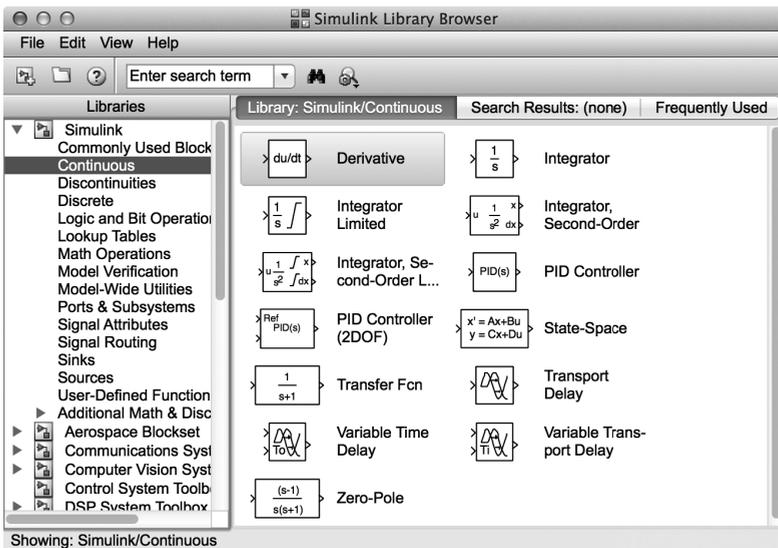


Abbildung 1.3. Simulink Library Browser.

Zunächst muss über das Menü `File` → `New` → `Model` ein neues Modell erzeugt werden. In dieses Modell können vorgefertigte Blöcke aus dem Library-Browser eingefügt werden, indem der gewünschte Block in das Modell gezogen wird. Die Blöcke sind in Kategorien eingeteilt. Ein Integrierer für zeitkontinuierliche Systeme ist beispielsweise unter `Simulink` → `Continuous` zu finden. Eine Suche in allen Kategorien ist über das Eingabefeld unter der Menüleiste möglich.

Mit der Maus können Verbindungen zwischen den Blöcken hergestellt werden. Durch Doppelklick auf einen Block öffnet sich eine Maske, in welche Parameter eingegeben werden können. Bevor eine Simulation gestartet wird, sollten die Parameter des Modells überprüft werden: Nach Aufruf des Menüs `Simulation` → `Model Configuration Parameters` öffnet sich ein Fenster, in das Parameter eingegeben werden können, die für das gesamte Modell gelten. Hier können Einstellungen zur `step size` der Simulation definiert werden. Zusätzlich wird hier die Simulationszeit festgelegt: Wird als Simulationsende statt `inf` ein Wert eingetragen, so stoppt die Simulation nach Ablauf dieser Zeit.

Gestartet wird die Simulation durch Klick auf den Run-Button. Treten Fehler oder Warnungen auf, so werden diese im *Matlab*-Prompt ausgegeben. Es empfiehlt sich deshalb, nach Anhalten der Simulation diesen auf Fehler zu überprüfen.

Die Betrachtung von Signalen ist beispielsweise über den Scope-Block möglich. Durch Doppelklick öffnet sich ein Koordinatensystem, in welchem das Signal am Eingang auch während der Simulation in Echtzeit dargestellt wird. Wird die Simulation angehalten, so kann über die entsprechenden Buttons gezoomt werden (siehe Abb. 1.4).

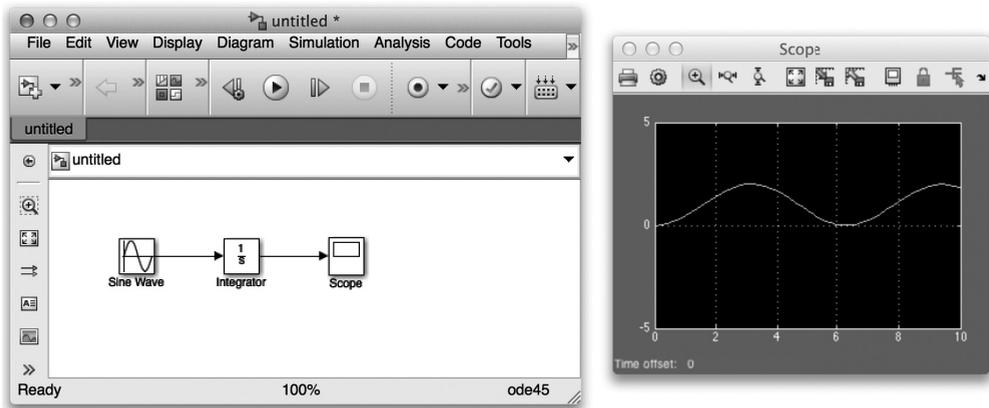


Abbildung 1.4. Modell (links) und Scope (rechts).

1.3.2.1 Subsystem

Zur Vereinfachung der Darstellung ist es oft sinnvoll, mehrere Blöcke zu einer Funktion zusammenzufassen. Hierzu werden mehrere Blöcke ausgewählt. Dies kann beispielsweise durch Drücken der Shift-Taste beim Markieren der Blöcke geschehen. Durch Auswählen des Kontextmenüs *Create Subsystem from Selection* werden diese zusammengefasst. Der Block kann nun durch Anklicken des Namens umbenannt werden. Ein Doppelklick auf den erstellten Block zeigt die enthaltene Struktur.

Wird solch ein System mehrfach benötigt, müssen diese Schritte nicht wiederholt werden. Es genügt, den Subsystem-Block zu kopieren.

1.3.2.2 Ausgabe von Daten an Matlab

Simulationsergebnisse können an *Matlab* über den Block *To Workspace* ausgegeben werden. Über einen Doppelklick auf den Block können die Parameter der Datenübergabe eingestellt werden. Wichtig sind hier die Einstellungen zur Art der Datenübergabe. Es kann zwischen *Structure*, *Structure With Time*, *Array* und *Timeseries* ausgewählt werden.

- Wird Array gewählt, so sind die Daten nach Ablauf der Simulation in zwei Variablen zu finden. Eine trägt den Namen des Übergabeblocks und enthält die Signal-daten, in der anderen (t_{out}) ist die Simulationszeit gespeichert.
- Wird als Format Structure gewählt, so werden alle Daten in einer einzigen `struct`-Variable gespeichert. Über `simout.signals.values` kann dann beispielsweise auf die Werte der einzelnen Signale zugegriffen werden.

1.4 Einführung in LabVIEW

Mit *LabVIEW* wurde Anfang der achtziger Jahre auf Basis einer graphischen Datenflussmaschine eine völlig neue Art der Mensch-Maschine-Kommunikation eingeführt. *LabVIEW* verknüpft zwei bewährte Programmiermethoden miteinander – den Datenfluss und die strukturierte Programmierung – und integriert sie in eine einzige graphische Programmierumgebung, die alle Elemente einer modernen Benutzeroberfläche bereitstellt.

Mess- und Automatisierungstechnik ohne Software ist heute kaum noch denkbar. Durch das Konzept der graphischen Programmierung, insbesondere durch die Express-Technologie, weist *LabVIEW* einen erheblichen Vorteil gegenüber textbasierten Programmiersprachen auf. Viele Standardaufgaben – wie z. B. Datenerfassung, Datenaufbereitung und Ergebnisdarstellung – können durch einfache Konfigurationswerkzeuge schnell und effizient gelöst werden, ohne dass eine langwierige Schnittstellenprogrammierung notwendig wäre. Dies erleichtert die Arbeit erheblich und bringt eine enorme Verkürzung der Entwicklungszeit mit sich. Allerdings können diese „vorgefertigten“ Werkzeuge auch eine Einschränkung bedeuten, falls z. B. sehr spezielle Probleme gelöst werden müssen und viele Blöcke bzw. Unterprogramme neu implementiert oder angepasst werden müssen.

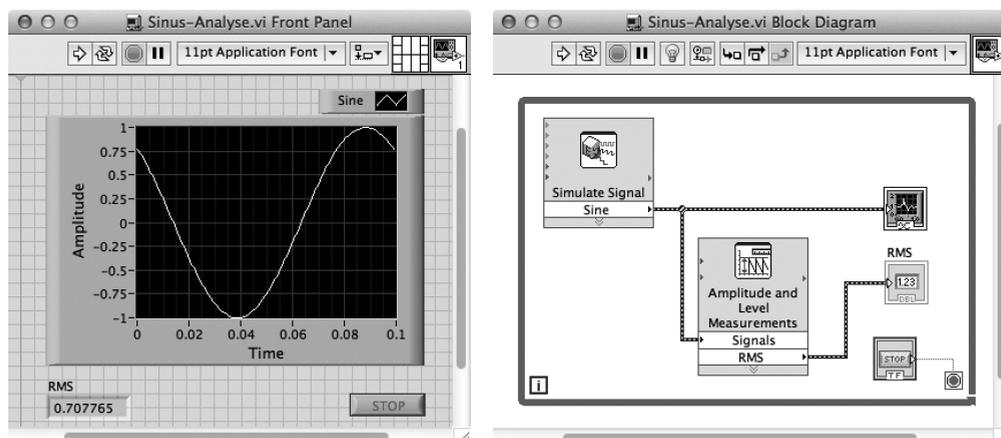
In diesem Abschnitt werden die Grundzüge von *LabVIEW* vorgestellt, damit Sie in der Lage sind, einfache Probleme damit zu lösen. Die anschließenden Übungsaufgaben dienen der Vertiefung.

1.4.1 Allgemeines zu LabVIEW-Programmen

LabVIEW-Programme werden als virtuelle Instrumente (**VI**s) bezeichnet. Die Bedienelemente in einem VI dienen zur Eingabe, die Anzeigeelemente hingegen zur Ausgabe von Daten. Jedes VI besteht aus drei Komponenten:

- einem **Frontpanel** (Abb. 1.5(a)), über welches man mit dem VI interagiert,
- einem **Blockdiagramm** (Abb. 1.5(b)), das den Programmcode des VIs darstellt,
- Anschlüssen, die zur Anbindung des VIs an weitere VIs dienen.

Das Frontpanel dient zur Benutzerinteraktion während der Programmausführung. Es ermöglicht die Steuerung des Programms und die Änderung von Eingabewerten durch den Anwender sowie die Online-Darstellung der Daten. Die Eingabe von Wer-



(a) Frontpanel

(b) Blockdiagramm

Abbildung 1.5. Komponenten von LabVIEW.

ten erfolgt über Bedienelemente. So stehen u. a. Schieberegler beispielsweise zum Einstellen eines Alarmwerts und Kippschalter zum Ein-/Ausschalten sowie Schaltflächen – z. B. Stopp zum Beenden der Programmausführung – zur Verfügung. Anzeigeelemente hingegen werden zur Ausgabe von Daten verwendet. Thermometer, LEDs und sonstige Anzeigeelemente zeigen die vom Programm gelieferten Werte an. Bei diesen Werten handelt es sich beispielsweise um Daten, Programmzustände und diverse andere Informationen. Jedes Bedien- bzw. Anzeigeelement auf dem Frontpanel verfügt über einen entsprechenden Anschluss im Blockdiagramm. Bei der Ausführung eines VIs fließen die über Bedienelemente eingegebenen Werte durch das Blockdiagramm; hier werden sie von Funktionen weiterverarbeitet und die Ergebnisse anschließend an weitere Funktionen und Anzeigeelemente übergeben.

1.4.2 Bedienung und Programmierung von VIs

1.4.2.1 Frontpanel

Das Frontpanel stellt die Benutzeroberfläche des VIs dar. Es kann Bedien- und Anzeigeelemente aufnehmen, die als interaktive Ein- bzw. Ausgabeelemente des VIs fungieren. Zu den Bedienelementen gehören Drehknöpfe, Schaltflächen, Drehregler und andere Eingabeelemente. Anzeigeelemente sind z. B. Graphen oder LEDs. Die Bedienelemente auf dem Frontpanel simulieren also entsprechende reale Eingabeeinheiten und ermöglichen die Übergabe von Werten an das Blockdiagramm (Steuerprogramm) des VIs. Anzeigeelemente ahmen ihrerseits physische Ausgabeeinheiten nach und zeigen die im Rahmen des Programms erfassten bzw. erzeugten Daten an.

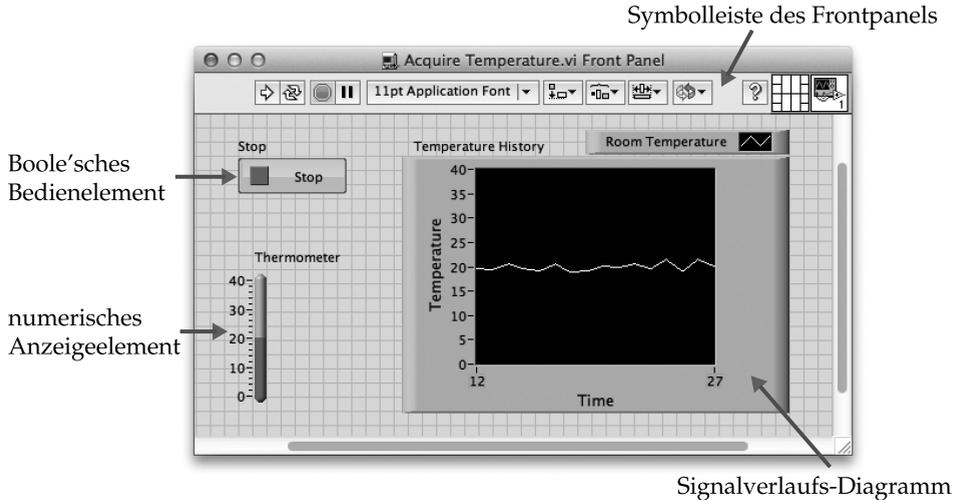


Abbildung 1.6. Frontpanel eines VIs.

Auf dem Frontpanel in Abb. 1.6 dient ein Boole'sches Bedienelement als Stoppschalter. Ein Boole'sches Element kann nur zwei mögliche Werte annehmen: TRUE oder FALSE. Klickt man mit der Maus auf den Schalter, so nimmt dieses Eingabeelement den Wert TRUE an.

Das mit Temperature History beschriftete Anzeigeelement ist ein Signalverlaufsdiagramm. Es ist in der Lage, Zahlenwerte im Zeitverlauf darzustellen. In diesem Fall wird die Temperatur in Grad Celsius gegen die Zeit in Sekunden abgetragen.

1.4.2.2 Blockdiagramm

Das Blockdiagramm enthält den graphischen Quellcode. Frontpanel-Objekte erscheinen im Blockdiagramm in Form entsprechender Anschlüsse. Darüber hinaus weist das Blockdiagramm Funktionen und Strukturen aus den integrierten *LabVIEW*-VI-Bibliotheken auf. Sämtliche Knoten im Blockdiagramm einschließlich der Anschlüsse von Funktionen, Strukturen, Bedien- und Anzeigeelementen sind mit anderen Blockdiagrammobjekten verbunden („verdrahtet“).

Im Blockdiagramm Abb. 1.7 ruft das Objekt Demo Temp ein SubVI auf, das einen von der Datenerfassungskarte aufgenommenen Temperaturwert einliest. Dieser Temperaturwert wird sodann im Signalverlaufsdiagramm Temperature History dargestellt. Beim Schalter Stop handelt es sich um ein Boole'sches Bedienelement auf dem Frontpanel, mit dessen Hilfe sich die Ausführung der While-Schleife beenden lässt.

1.4.2.3 Express-VI

Mit *LabVIEW 7.0* wurde eine neue Art von SubVI eingeführt: das sogenannte **Express-VI** (Abb. 1.8). Hierbei handelt es sich um interaktive VIs, deren Funktionalität

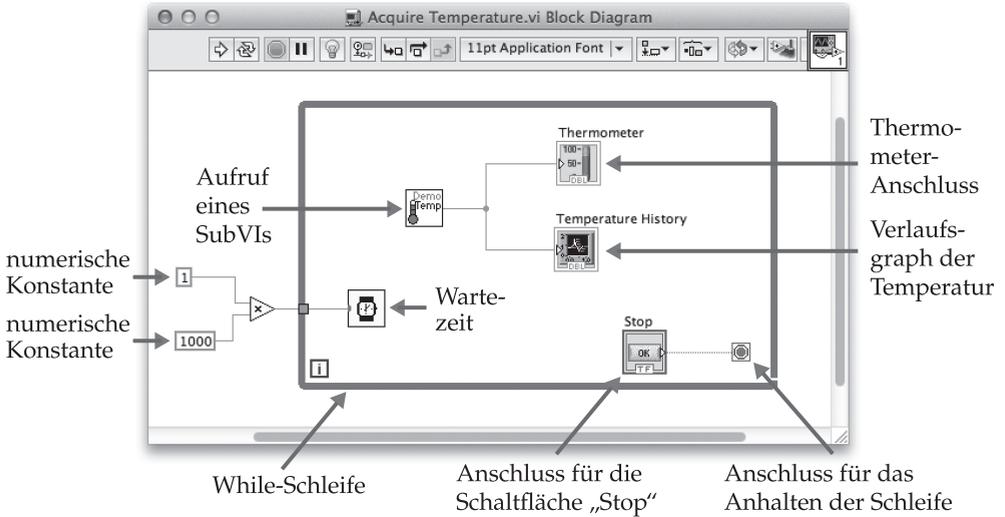


Abbildung 1.7. Blockdiagramm eines VIs.

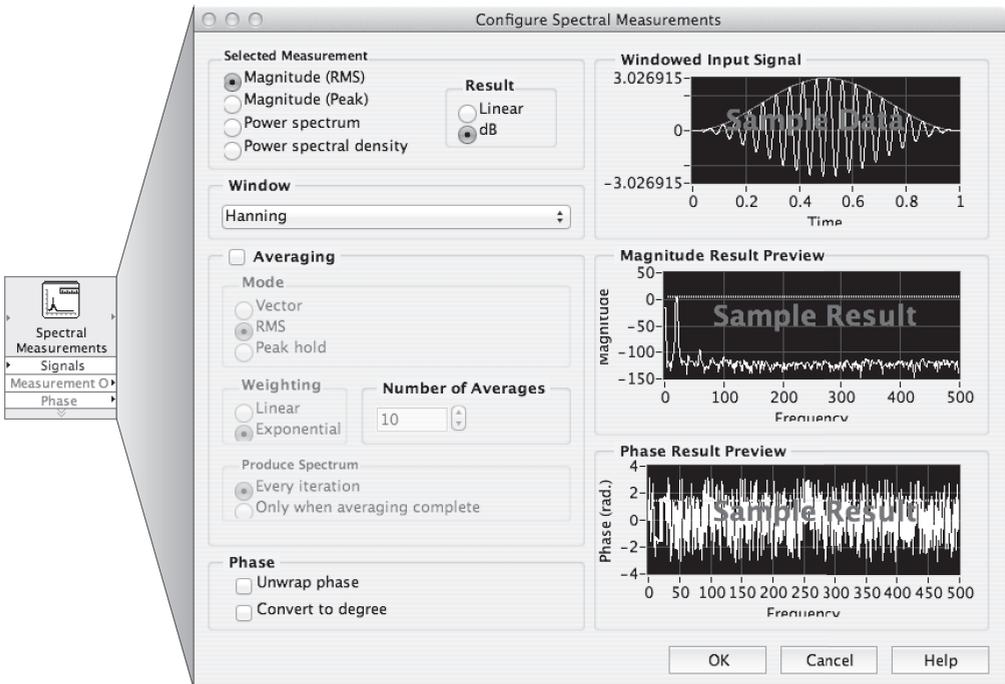


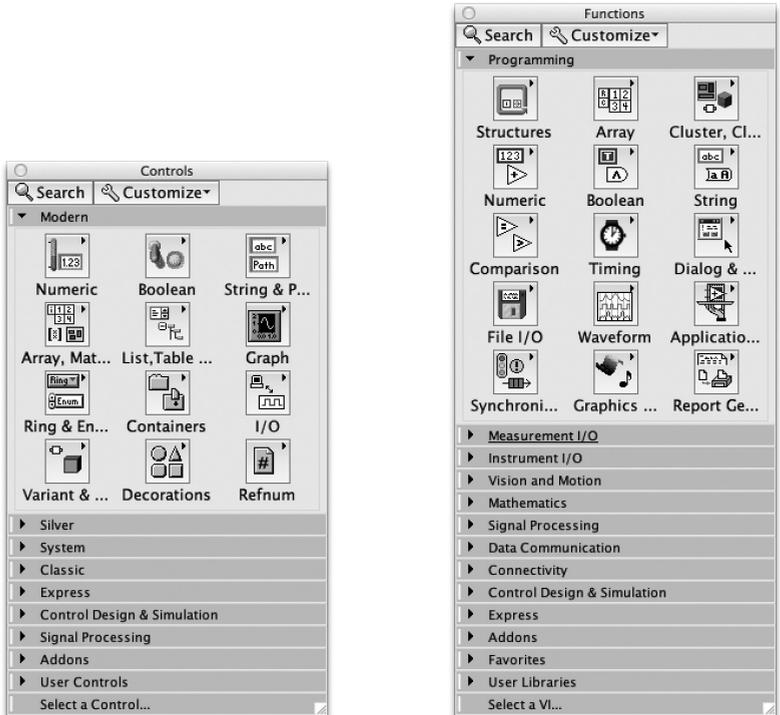
Abbildung 1.8. Express-VI Spectral Measurements.

über ein spezielles Dialogfenster entsprechend den Anforderungen des Anwenders konfiguriert werden kann. Anschließend erstellt *LabVIEW* gemäß den vorgenommenen Einstellungen ein SubVI. Ein Standard-VI hingegen ist ein aus Frontpanel und Blockdiagramm bestehendes VI, das innerhalb eines übergeordneten VIs zum Einsatz kommt. Funktionen sind die sämtlichen VIs zugrunde liegenden Bausteine. Sie verfügen weder über Frontpanel noch über Blockdiagramm.

1.4.2.4 Elemente- und Funktionenpalette

Mit Hilfe der Elementpalette (Abb. 1.9(a)) platzieren Sie Bedien- und Anzeigeelemente auf dem Frontpanel. Diese Palette steht nur für das Frontpanel zur Verfügung. Öffnen Sie die Elementpalette über den Menüpunkt Ansicht → Werkzeugpalette. Alternativ können Sie mit der rechten Maustaste an einer freien Stelle des Frontpanels klicken. Wenn Sie die Elementpalette auf dem Bildschirm verankern möchten, klicken Sie auf das Reißzweckensymbol in der linken oberen Ecke der Palette.

Die Funktionenpalette (Abb. 1.9(b)) hingegen dient zur Erstellung des Blockdiagramms. Sie steht ausschließlich für das Blockdiagramm zur Verfügung. Öffnen Sie



(a) Elementpalette (Frontpanel) (b) Funktionenpalette (Blockdiagramm)

Abbildung 1.9. Elemente- und Funktionenpalette.

die Funktionenpalette über den Menüpunkt Ansicht → Funktionenpalette. Alternativ können Sie mit der rechten Maustaste an einer freien Stelle im Blockdiagramm klicken. Wenn Sie die Funktionenpalette auf dem Bildschirm verankern möchten, klicken Sie auf das Reißzweckensymbol in der linken oberen Ecke der Palette.

1.4.2.5 Werkzeugpalette

Ist die automatische Werkzeugwahl aktiviert, wählt *LabVIEW* selbständig das jeweils geeignete Werkzeug aus der *Werkzeugpalette* (Abb. 1.10), sobald Sie den Mauszeiger über ein Frontpanel- oder Blockdiagramm-Objekt bewegen. Es empfiehlt sich, die automatische Werkzeugwahl dauerhaft beizubehalten. Zum Aktivieren bzw. Deaktivieren der automatischen Werkzeugwahl klicken Sie in der *Werkzeugpalette* auf die Schaltfläche *Automatische Werkzeugwahl*.

Das *Bedienwerkzeug* dient zum Ändern des Werts eines Bedienelements oder als Cursor zur Texteingabe. Mit dem *Positionierwerkzeug* können Objekte ausgewählt, bewegt und in ihrer Größe verändert werden. Dabei nimmt das *Positionierwerkzeug* die Form eines kleinen Doppelpfeils an, wenn der Mauszeiger beispielsweise auf die Ecken eines in der Größe veränderbaren Objekts bewegt wird. Das *Beschriftungswerkzeug* wiederum wird verwendet, um Text zu bearbeiten und an beliebigen Stellen im Frontpanel bzw. Blockdiagramm freie Beschriftungen anzubringen. Bei der Erstellung einer Beschriftung verwandelt sich das *Beschriftungswerkzeug* in einen Cursor. Das *Verbindungswerkzeug* dient zum Verbinden („Verdrahten“) von Blockdiagramm-Objekten.



Abbildung 1.10. Werkzeugpalette.

1.4.2.6 Symbolleiste

Abbildung 1.11 zeigt die *LabVIEW*-Symbolleiste.

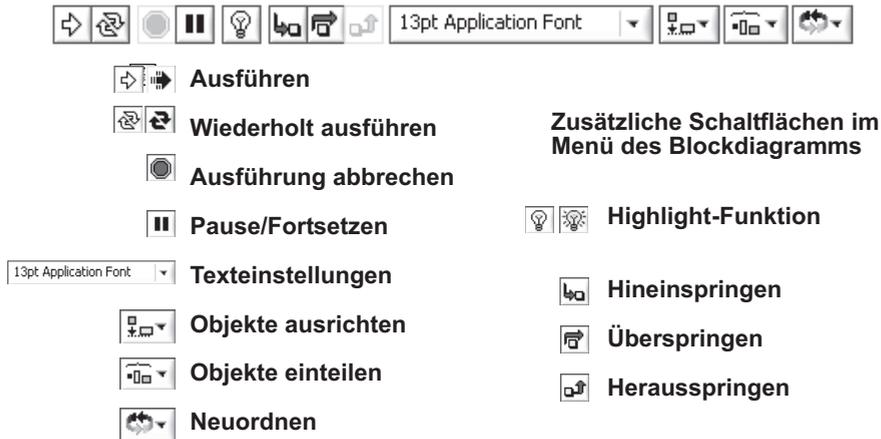


Abbildung 1.11. Symbolleiste.

- Klicken Sie auf die Schaltfläche **Ausführen**, um das VI auszuführen. Während der Ausführung erscheint auf der Schaltfläche ein schwarzer Pfeil, wenn es sich beim VI um eines der höchsten Ebene (*Top-Level VI*) handelt. Darunter versteht man ein VI, das seinerseits von keinem übergeordneten VI aufgerufen wird und somit kein SubVI ist.
- Klicken Sie auf die Schaltfläche **Wiederholt ausführen**, um das VI so lange auszuführen, bis Sie die Ausführung unterbrechen bzw. ganz beenden. Durch einen erneuten Klick auf diese Schaltfläche können Sie den Modus der wiederholten Ausführung deaktivieren.
- Während der Ausführung des VIs steht die Schaltfläche **Ausführung abbrechen** zur Verfügung. Klicken Sie auf diese Schaltfläche, um das VI ohne Verzögerung anzuhalten.
Hinweis: Vermeiden Sie möglichst die Verwendung dieser Schaltfläche, um VIs zu beenden. Lassen Sie stattdessen das VI bis zum Ende ausführen oder implementieren Sie eine Möglichkeit zur Beendigung der Ausführung. Hierdurch ist der Zustand des VIs zum Zeitpunkt der Beendigung bekannt. Fügen Sie hierzu auf dem Frontpanel eine Schaltfläche (Boole'sches Bedienelement) hinzu, die bei einem Klick die Ausführung des VIs beendet (vgl. Abb. 1.6).
- Klicken Sie auf die Schaltfläche **Pause**, um die Ausführung des VIs vorübergehend anzuhalten. Beim Klick auf diese Schaltfläche markiert *LabVIEW* im Blockdiagramm die Stelle, an der sich die Ausführung zum Zeitpunkt der Unterbrechung gerade befindet. Ein erneuter Klick auf die Schaltfläche **Pause** setzt die Ausführung des VIs fort.
- Im Menü **Anwendungsschriftart** können Sie die im VI verwendeten Schriftarten samt Größe, Formatierung und Farbe festlegen.

- Über das Menü **Objekte** ausrichten richten Sie Objekte an einer gemeinsamen Achse aus, beispielsweise vertikal zentriert, an der oberen oder linken Kante entlang.
- Verwenden Sie das Menü **Objekte anordnen**, um die Objekte gleichmäßig im Fenster zu verteilen, beispielsweise mit Abstand zueinander, dicht beisammen ohne Abstand usw.
- Verwenden Sie das Menü **Neu ordnen**, um die Darstellungsreihenfolge sich überlappender Objekte festzulegen. Markieren Sie hierzu das betreffende Objekt mit dem Positionierwerkzeug und wählen Sie dann einen der Menüpunkte **Um eins nach vorne**, **Um eins nach hinten**, **Nach vorne** oder **Nach hinten**.

Die folgenden Schaltflächen erscheinen nur auf der Symbolleiste des Blockdiagramms:

- Klicken Sie auf die Schaltfläche **Highlight-Funktion**, um den Fluss der Daten durch das Blockdiagramm mitzuverfolgen. Zum Deaktivieren der Highlight-Funktion klicken Sie erneut auf diese Schaltfläche.
- Klicken Sie auf die Schaltfläche **Einzelstrittausführung starten**, um sich im Einzelschrittmodus in eine Schleife, ein SubVI etc. hineinzubegeben. Der Einzelschrittmodus ermöglicht die Ausführung eines VIs Knoten für Knoten. Dabei blinkt ein Knoten, wenn er zur Ausführung bereit ist. Durch Hineinspringen in einen Knoten können Sie diesen im Einzelschrittmodus ausführen.
- Klicken Sie auf die Schaltfläche **Überspringen**, um eine Schleife, ein SubVI etc. zu überspringen. Durch Überspringen eines Knotens führen Sie diesen zwar aus, jedoch nicht im Einzelschrittmodus.
- Klicken Sie auf die Schaltfläche **Herausspringen**, um aus einer Schleife, einem SubVI etc. herauszuspringen. Durch Herausspringen aus einem Knoten beenden Sie die Einzelstrittausführung dieses Knotens und begeben sich zum nächsten Knoten.

1.4.3 Erstellen eines VIs

Wird auf dem Frontpanel ein Objekt hinzugefügt, so erscheint automatisch ein entsprechender Anschluss im Blockdiagramm. Diese Anschlüsse ermöglichen das Zugreifen auf die Frontpanel-Objekte aus dem Blockdiagramm heraus. Jeder Anschluss bietet nützliche Informationen über das Frontpanel-Objekt, welches er repräsentiert. Die Farben und Symbole etwa geben Aufschluss über den Datentyp. Fließkommawerte mit doppelter Genauigkeit werden durch orangefarbene Anschlüsse mit der Bezeichnung **DBL** dargestellt. Boole'sche Anschlüsse hingegen sind grün und tragen die Bezeichnung **TF**. Außerdem gibt es Anschlüsse für String-Konstanten (purpur) und Dateipfade (türkis).

Normalerweise sollten nur Anschlüsse derselben Farbe miteinander verbunden werden. Dies ist jedoch keine verbindliche Regel: Es ist auch möglich, blaue An-

schlüsse (Ganzzahlen) mit orangefarbenen Anschlüssen (Bruchzahlen) zu verbinden. In der Regel empfiehlt sich jedoch die Kombination gleichfarbiger Anschlüsse.

Zu Bedienelementen gehörende Anschlüsse weisen einen Pfeil auf der rechten Seite sowie eine stärkere Umrandung auf. Bei den Anschlüssen der Anzeigeelemente hingegen befindet sich der Pfeil links und die Umrandung ist dünner. Die Verbindung von Objekten erfolgt nach logischen Regeln: Jede Verbindung muss über genau eine Quelle (z. B. Bedienelementanschluss) sowie mindestens ein Ziel (z. B. Anzeigeelementanschluss) verfügen. Beim in Abb. 1.12 dargestellten Programm werden die Daten sowohl von A als auch von B jeweils an eine Additions- und eine Subtraktionsfunktion übergeben. Die Ergebnisausgabe erfolgt über die entsprechenden Anzeigeelemente.

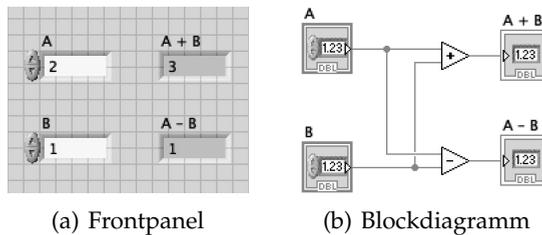


Abbildung 1.12. Frontpanel und Blockdiagramm.

Zusätzlich zu den Frontpanel-Anschlüssen enthält das Blockdiagramm auch Funktionen. Jede Funktion kann über mehrere Ein- und Ausgabeanchlüsse verfügen. Die richtige Verbindung dieser Anschlüsse ist ein wichtiger Aspekt bei der Programmierung in *LabVIEW*. Es folgen einige grundlegende Tipps:

- Zum Verbinden („Verdrahten“) der Funktionsknoten dient das Verbindungswerkzeug (Abb. 1.13). Beim Versuch, eine Verbindung mit dem Anschluss zu erstellen, sollte man mit dem abstehenden Drahtende des Verbindungswerkzeuges (Drahtspule, vgl. Abb. 1.16 oben links) auf den blinkenden Anschluss zielen. Dort wird die Verbindung angebracht.

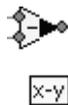


Abbildung 1.13. „Verdrahtung“.

- Beachten Sie die gelben Hinweisstreifen, wenn das Verbindungswerkzeug über einen der Anschlüsse einer Funktion bewegt wird. Dieser gibt Auskunft über die Art des anvisierten Anschlusses.
- Wird das Verbindungswerkzeug über einen Anschluss bewegt, so blinkt dieser. Dies erleichtert es, die Verbindung mit dem gewünschten Anschluss zu erstellen.

- Klicken Sie mit der rechten Maustaste auf eine Funktion und wählen Sie Sichtbare Objekte → Anschlüsse, wenn Sie weitere Hilfestellung zu den Anschlüssen benötigen (Abb. 1.14). Das Funktionssymbol tritt dann in den Hintergrund, die farblich abgesetzten Anschlüsse in den Vordergrund. Dabei entsprechen die verschiedenen Farben den jeweiligen Datentypen der Frontpanel-Anschlüsse.

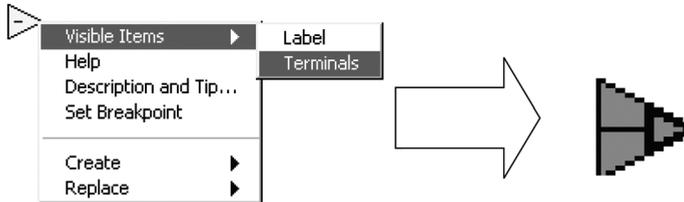


Abbildung 1.14. Kontextmenü einer LabVIEW-Funktion.

Weitergehende Details erhalten Sie über die Online-Hilfe (Abb. 1.15), die sich für jede Funktion über den Menüpunkt Hilfe des Kontextmenüs aufrufen lässt.

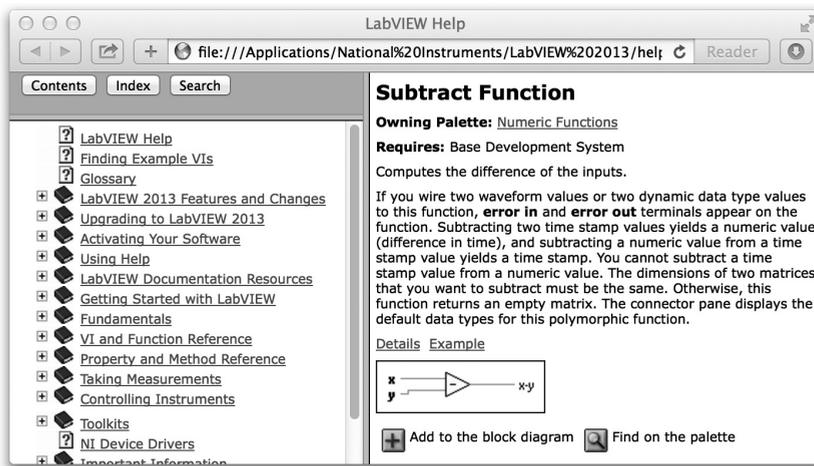


Abbildung 1.15. Online-Hilfe.

Sollte eine Verbindung unübersichtlich erscheinen, so klicken Sie mit der rechten Maustaste darauf und wählen Sie Verdrahtung bereinigen. Der Verlauf der Verbindung wird optimiert (Abb. 1.16).

Das Verbinden von Objekten wird in *LabVIEW* sehr flexibel gehandhabt. Nutzen Sie beim Erstellen von Verbindungen die kombinierte Tastatur- und Mausbedienung. Hier einige Beispiele:

- Durch einen, zwei oder drei Mausklicks auf eine Verbindung lassen sich diverse Abschnitte dieser Verbindung auswählen (Abb. 1.16). Die Auswahl kann sodann verschoben oder gelöscht werden.

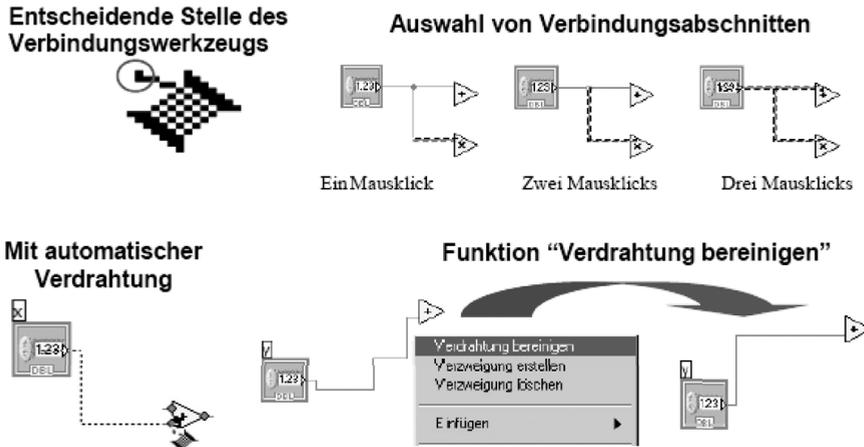


Abbildung 1.16. Verbinden von Objekten.

- Wenn Sie während der Verlegung einer Verbindung mit der Maus klicken, entsteht an dieser Stelle eine Abbiegung im Verbindungsverlauf.
- Über einen Klick mit der rechten Maustaste oder durch Drücken der Esc-Taste kann die Erstellung einer Verbindung abgebrochen werden.

Um die richtige Farbe der Verbindung brauchen Sie sich nicht zu kümmern, denn *LabVIEW* wählt automatisch die dem Datentyp entsprechende Farbe.

Durch Klick mit der rechten Maustaste auf einen Block-Eingang lässt sich im erscheinenden Menü durch Klick auf Konstante ein dem Eingangsdatentyp entsprechender Konstanten-Block erstellen.

Automatisches Verbinden von Objekten

LabVIEW ist in der Lage, bei der Platzierung von Objekten auf dem Frontpanel diese automatisch zu verbinden. Selbst Objekte, die sich bereits im Blockdiagramm befinden, können nachträglich von *LabVIEW* verbunden werden. Dabei werden zueinander passende Anschlüsse verbunden – solche jedoch, die zu keinem Anschluss passen, bleiben ohne Verbindung. Bewegen Sie im Blockdiagramm ein Objekt bei gedrückter Maustaste nahe an ein anderes Objekt heran, so erscheinen temporäre Drähte, die gültige Verbindungen darstellen. Wenn Sie die Maustaste loslassen und so das Objekt im Blockdiagramm platzieren, wandelt *LabVIEW* die temporäre Verdrahtung in eine permanente um. Sie können die automatische Verbindungserstellung (de-)aktivieren, indem Sie die Leertaste drücken, während Sie ein Objekt mithilfe des Positionierwerkzeugs verschieben. Zur standardmäßigen (De-)Aktivierung der automatischen Verdrahtung wählen Sie **Werkzeuge** → **Optionen**, im erscheinenden Dialog die Option **Blockdiagramm** und hier wiederum das entsprechende Kontrollkästchen **Automatische Verbindung** aktivieren.

1.4.3.1 Datenflussprogrammierung

In *LabVIEW* liegt der Ausführung von VIs das Datenflussmodell zugrunde. Ein Blockdiagrammknoten wird ausgeführt, sobald alle seine Eingänge belegt sind. Ist die Ausführung eines Knotens abgeschlossen, werden die Daten an die jeweiligen Ausgabeanschlüsse übergeben und die Ausgabedaten dann an den nächsten Knoten desselben Pfads im Datenflussdiagramm weitergeleitet. *C++*, *JAVA* und die meisten anderen textbasierten Programmiersprachen folgen bei der Programmausführung einem auf Steuerstrukturen basierendem Modell. Hierbei legt die Reihenfolge der einzelnen Programmelemente den Ablauf der Programmausführung fest.

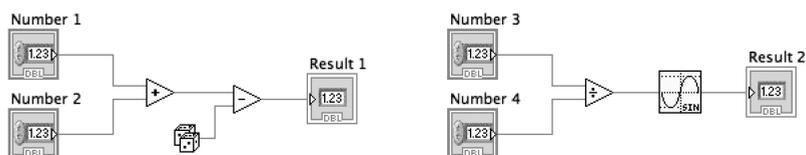


Abbildung 1.17. Datenflussprogrammierung.

Es sei nun das Blockdiagramm Abb. 1.17 betrachtet. Links werden zwei Zahlen addiert und von der Summe anschließend eine Zufallszahl subtrahiert. In diesem Fall wird das Blockdiagramm von links nach rechts ausgeführt. Grund hierfür ist jedoch nicht die Reihenfolge der Objekte, sondern die Tatsache, dass einer der Eingänge der Subtraktionsfunktion erst belegt werden kann, wenn die Additionsfunktion ausgeführt und das Ergebnis an die Subtraktionsfunktion weitergeleitet wurde. Bedenken Sie, dass ein Knoten nur dann ausgeführt werden kann, wenn an all seinen Eingängen Daten anliegen. Zudem erfolgt die Datenübergabe an die Ausgänge erst nach Beendigung der Ausführung.

Es sei das Blockdiagramm Abb. 1.17 erneut betrachtet: Welches der folgenden Codesegmente wird zuerst ausgeführt – die Funktion Addieren, Zufallszahl oder Dividieren? Diese Frage lässt sich so nicht beantworten, denn die Eingänge der Funktionen Addieren und Dividieren werden zeitgleich belegt, während die Funktion Zufallszahl über keinerlei Eingänge verfügt. Ist es erforderlich, dass von zwei Codesegmenten eines vor dem anderen ausgeführt wird und besteht keine Abhängigkeit zwischen beiden Segmenten hinsichtlich der Daten, so verwenden Sie eine Sequenzstruktur zur Erzwingung der erforderlichen Ausführungsreihenfolge (Funktionen → Programmierung → Strukturen → Flache Sequenzstruktur).

1.4.3.2 Fehlersuche

Sollte ein VI nicht ausführbar sein, so erscheint auf der Schaltfläche Ausführen ein gebrochener Pfeil (Abb. 1.18).

- **Fehlersuche:** Mit einem Klick auf die Schaltfläche mit dem gebrochenen Pfeil öffnet sich das Fenster Fehlerliste. Um das fehlerhafte Objekt im Blockdiagramm hervorzuheben, führen Sie einen Doppelklick auf eine Fehlermeldung aus.

- **Suchen nach Fehlern**



Klick auf gebrochene Schaltfläche Ausführen öffnet das Fenster "Fehlerliste"

- **Ausführung im Highlight-Modus**



Klick auf die Schaltfläche "Highlight-Funktion" bewirkt Animation des Datenflusses in Form von "Datenbällen"; Zwischenergebnisse werden direkt an den Verbindungen angezeigt

- **Sonde**



Klick mit rechter Maustaste auf Verbindung, um Sonde anzubringen; Sonde zeigt aktuellen Datenwert im jeweiligen Verbindungsstück



Alternativ: Werkzeug Probe-Daten von der Werkzeugpalette und Klick auf gewünschte Verbindung

Abbildung 1.18. Möglichkeiten der Fehlersuche.

- **Highlight-Funktion:** Sie ermöglicht es, durch die Animation des Blockdiagramms den Datenfluss mitzuverfolgen und Zwischenwerte einzusehen. Sie können diese Funktion über die Schaltfläche mit dem Glühbirnensymbol (de-)aktivieren.
- **Sonde:** Mit ihrer Hilfe lassen sich Werte gezielt einsehen, z. B. in Arrays und Clusters. Klicken Sie zum Anbringen einer Sonde mit dem Werkzeug Sondenwerte auf die gewünschte Verbindung.
- **Haltepunkt:** Dient zum gezielten Unterbrechen der VI-Ausführung an einer gewünschten Stelle im Blockdiagramm. Klicken Sie mit dem Haltepunkt-Werkzeug auf ein Verbindungsstück oder ein Objekt, um dort einen Haltepunkt zu setzen.

1.4.4 Weitere Programmstrukturen

1.4.4.1 Schleifen

Sowohl die While- als auch die For-Schleife sind auf der Palette Funktionen → Strukturen zu finden. Im Unterschied zur While-Schleife wird die For-Schleife nur so oft ausgeführt, wie vom Anwender spezifiziert. Eine While-Schleife hingegen stoppt die Ausführung des Unterdiagramms nur dann, wenn am Bedingungsanschluss ein entsprechender Wert vorliegt.

- **While-Schleife**

Ähnlich wie bei der Do- oder Repeat-Until-Schleife in textbasierten Programmiersprachen führt eine While-Schleife (Abb. 1.19(a)) ein Unterdiagramm so lange aus, bis eine bestimmte Bedingung erfüllt ist. Die While-Schleife führt den Code so lange aus, bis der Bedingungsanschluss (ein Eingangsanschluss) einen bestimmten Boole'schen Wert annimmt. Das Verhalten und das Erscheinungsbild des Be-

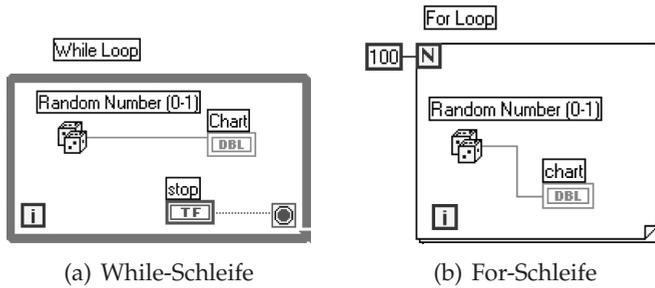


Abbildung 1.19. Schleifen in LabVIEW.

dingungsanschlusses (rechts unten in der While-Schleife) sind standardmäßig auf Weiter, wenn TRUE eingestellt ist. Ist der Bedingungsanschluss auf Weiter wenn TRUE gesetzt ist, so wird das Unterdiagramm der While-Schleife so lange ausgeführt, bis der Bedingungsanschluss einen FALSE-Wert annimmt. Der Iterationsanschluss (links unten in der While-Schleife), ein Ausgangsanschluss, enthält die Anzahl der abgeschlossenen Schleifendurchläufe. Die Zählung der Durchläufe beginnt stets bei null. Während des ersten Durchlaufs gibt der Iterationsanschluss null zurück.

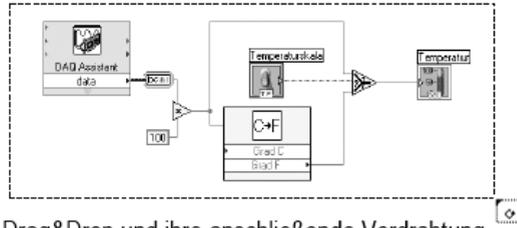
– For-Schleife

Eine For-Schleife (Abb. 1.19(b)) führt ein Unterdiagramm so oft aus wie vorgegeben. Der Wert N stellt einen Eingangsanschluss dar, der anzeigt, wie oft das Unterdiagramm wiederholt werden soll (Schleifenzähler). Der Iterationsanschluss (links unten in der For-Schleife), der auch hier ein Ausgabeanschluss darstellt, enthält die Anzahl der abgeschlossenen Schleifendurchläufe. Die Zählung der Durchläufe beginnt stets bei null. Während des ersten Durchlaufs gibt der Iterationsanschluss den Wert null zurück.

Programmieren von Schleifen

Die Schleifen befinden sich auf der Palette Funktionen → Strukturen und werden wie folgt im Blockdiagramm abgelegt (Abb. 1.20). Wurde eine der beiden Schleifen ausgewählt, so nimmt der Mauszeiger die Form einer Miniaturschleife an, mit deren Hilfe nun der zu wiederholende Codeabschnitt umrahmt werden kann. Klicken Sie hierzu mit der Maus zuerst an der Stelle, an der Sie die linke obere Ecke der Schleife wünschen, und ein zweites Mal dort, wo sich die untere rechte Ecke der Schleife befinden soll. Hierdurch haben Sie den zu wiederholenden Codeabschnitt im Rechteck der Schleife eingeschlossen. Legen Sie bei Bedarf weitere Knoten in der While- oder For-Schleife ab.

1. Wahl der Schleife
2. Einschließen des zu wiederholenden Programmcodes



3. Hinzufügen weiterer Knoten per Drag&Drop und ihre anschließende Verdrahtung

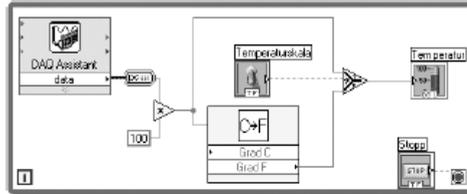
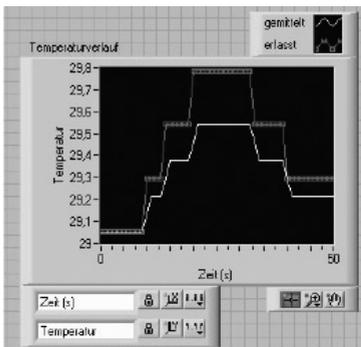


Abbildung 1.20. Schleifen erstellen.

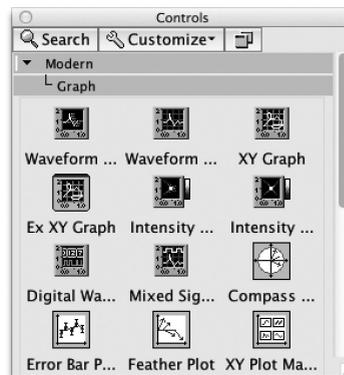
1.4.4.2 Diagramme und Graphen

Signalverlaufsdiagramme sind numerische Anzeigeelemente zur Darstellung von Kurvenzügen. Sie finden es in der Unterpalette Elemente → Modern → Graph (Abb. 1.21). Sie können einen oder mehrere Kurvenzüge anzeigen (vgl. Abb. 1.21(a)).

Die Minimal- und Maximalwerte der x- bzw. y-Achse lassen sich anpassen, indem Sie mit dem Beschriftungswerkzeug einen Doppelklick auf den betreffenden Skalenwert ausführen und den gewünschten Wert eingeben. Analog hierzu erfolgt auch die Achsenumbenennung. Wenn Sie den Linienstil, die Form oder die Farbe eines



(a) Diagramm



(b) Unterpalette Graph

Abbildung 1.21. Diagramme und Graphen.

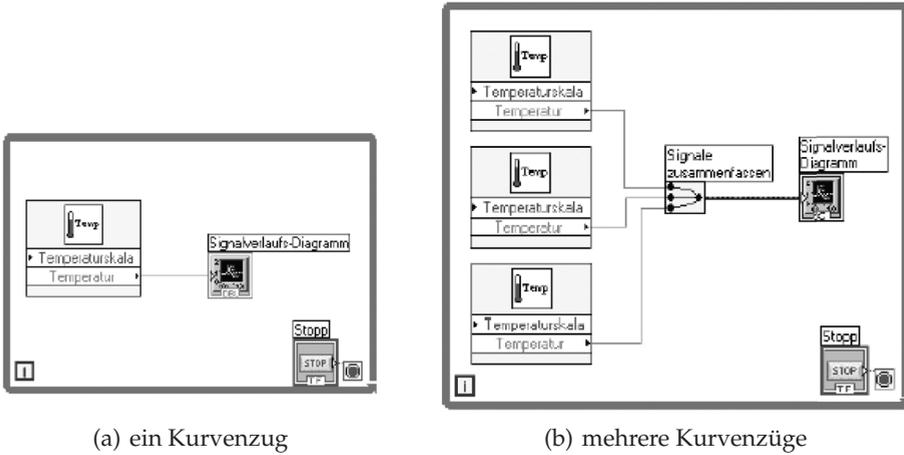


Abbildung 1.22. Diagramme mit Kurvenzügen.

Kurvenzugs ändern möchten, so klicken Sie mit der rechten Maustaste auf die Kurvenlegende, um auf die einzelnen Optionen zuzugreifen.

Sie können einen skalaren Ausgang direkt mit einem Signalverlaufsdiagramm verbinden und so die Daten in Form eines Kurvenzugs darstellen (Abb. 1.22). Zur Anzeige mehrerer Kurvenzüge in einem Diagramm verwenden Sie die Funktion Signale zusammenfassen, zu finden auf der Unterpalette Funktionen → Express → Signalverarbeitung. Diese Funktion leitet die Daten diverser Ausgänge in ein und dasselbe Signalverlaufsdiagramm. Zum Hinzufügen weiterer Kurvenzüge ziehen Sie einfach die Funktion Signale zusammenfassen mit dem Positionierwerkzeug vertikal auf. Die Online-Hilfe hält nützliche Informationen über die verschiedenen Möglichkeiten der Anbindung von Datenausgängen an Diagramme bereit.

Zoom-Funktion

In einem Signalverlaufsdiagramm stehen verschiedene Möglichkeiten zur Verfügung, um Signale genauer zu untersuchen. Klickt man im Frontpanel mit der rechten Maustaste auf den Graphen, so erscheint ein Kontextmenü mit verschiedenen Optionen. Wählt man Sichtbare Objekte → Graph-Palette, dann erhält man u. a. die Zoom-Funktion (Abb. 1.23). Über das mittlere Icon öffnet sich eine Palette, die mehrere Funktionen enthält.

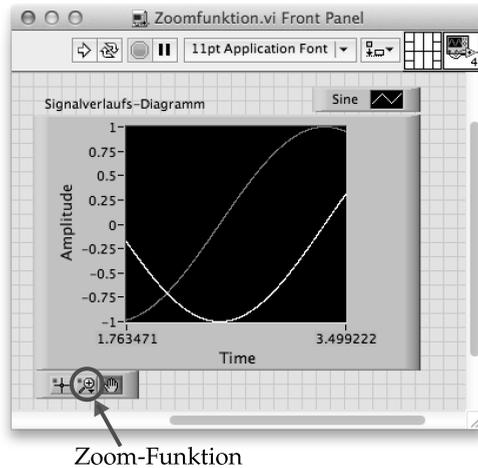


Abbildung 1.23. Zoom-Funktion in einem Signalverlaufsdiagramm.

1.5

1.5 Versuchsvorbereitende Aufgaben

Aufgabe 1.1: Grundlagen

Um die vorgestellten Programme anhand von realistischen Fragestellungen aus dem Alltag eines Ingenieurs kennenzulernen, sollen in dieser Aufgabe zunächst einige Grundlagen wiederholt werden.

- Machen Sie sich mit den Bedienelementen und mit den Besonderheiten der einzelnen Programme *Matlab*, *Simulink* und *LabVIEW* vertraut.
- Nach welchem Prinzip arbeitet ein Least-Squares-Schätzer? Beschreiben Sie einen Algorithmus, der eine LS-Schätzung durchführen kann.
- Definieren Sie das ARMA-Filter und geben Sie dessen Systemfunktion in allgemeiner Form an. Zeichnen Sie das zugehörige Strukturbild.
- Was ist stationäre Genauigkeit? Wie kann die Stabilität eines Systems überprüft werden?
- Wie ist das Leistungsdichtespektrum definiert und wie wird es berechnet?

1.6

1.6 Versuchsaufgaben

Erstellen Sie im Ordner `D:\PDSV\Gruppen` ein Verzeichnis für Ihre Gruppe. Legen Sie alle Dateien, die Sie im Laufe des Versuchs erstellen, nur in diesem Ordner ab!

Im Ordner `D:\PDSV\Versuch_1` finden Sie alle vorgefertigten VIs, die Sie für die einzelnen Aufgaben benötigen. Wenn Sie an diesen VIs Veränderungen vornehmen, dann überschreiben Sie die VIs in diesem Verzeichnis bitte nicht.

Aufgabe 1.2: Matrix-Operationen

In *Matlab* werden Zahlen und Werte in Matrix-Form verarbeitet. Diese Aufgabe zeigt einige Möglichkeiten auf, um Matrizen in *Matlab* zu verarbeiten.

a) Doppelpunkt-Operator

Ein wichtiger Operator in *Matlab* ist der Doppelpunkt, der als Platzhalter dient: Damit werden einzelne Zeilen oder Spalten einer Matrix angesprochen. Es soll die Determinante einer Matrix durch Umformung berechnet werden. Dabei wird ausgenutzt, dass sich die Determinante nicht ändert, wenn ein Vielfaches einer Zeile oder Spalte zu einer anderen Zeile bzw. Spalte addiert oder subtrahiert wird. Ziel ist es, möglichst viele Nulleinträge zu erzeugen. Gehen Sie wie folgt vor:

- Erzeugen Sie in *Matlab* die Matrix

$$\mathbf{G} = \begin{bmatrix} 7 & 5 & 3 \\ 12 & 14 & -3 \\ 8 & 8 & 6 \end{bmatrix}. \quad (1.1)$$

- Eliminieren Sie zunächst das rechte obere Element der Matrix \mathbf{G} unter Verwendung des Doppelpunkt-Operators.
- Aus der so entstandenen Matrix lässt sich nun das mittlere obere Element eliminieren.
- Berechnen Sie nun die Determinante als Produkt von drei Skalaren.
- Überprüfen Sie Ihr Ergebnis mit Hilfe des *Matlab*-Befehls `det`.

b) Punkt-Operator

Der Punkt-Operator ermöglicht die effiziente Berechnung einiger Operationen. Zur Veranschaulichung sollen Sie zwei Matrizen elementweise miteinander multiplizieren. Schreiben Sie hierzu ein *m-File*:

- Definieren Sie zwei Matrizen \mathbf{C} und \mathbf{D} der Größe 1000×1000 und füllen Sie diese mit Zufallszahlen aus dem Zahlenbereich zwischen 1 und 10 auf. Verwenden Sie hierzu den Befehl `randi`. Hinweise zu diesem Befehl erhalten Sie durch die Eingabe `help randi`.
- Verwenden Sie zuerst zwei `for`-Schleifen und multiplizieren Sie elementweise alle Elemente der beiden Matrizen miteinander.
- Realisieren Sie die gleiche Operation mit Hilfe des Punkt-Operators.
- Vergleichen Sie die Laufzeit beider Vorgänge und geben Sie die relative Zeitdifferenz an. Nutzen Sie dazu die Befehle `clock` und `etime`.

Wie unterscheiden sich die beiden Berechnungsverfahren hinsichtlich der benötigten Rechenzeit?

c) Lineare Gleichungssysteme

Diese Teilaufgabe soll zeigen, wie man Matrizen manipuliert und Determinanten berechnet. Dazu ist folgendes Gleichungssystem gegeben:

$$\begin{aligned} x_1 + 2x_2 + 3x_3 &= 4 \\ 2x_1 + 3x_2 + 4x_3 &= 5 \\ 4x_1 + 2x_2 + 5x_3 &= 1 \end{aligned} \quad \Leftrightarrow \quad \mathbf{A} \cdot \mathbf{x} = \mathbf{b}. \quad (1.2)$$

Dieses lineare Gleichungssystem soll auf unterschiedliche Weise gelöst werden.

1. Cramer'sche Regel (Determinantenverfahren)

Nach der Cramer'schen Regel lassen sich die unbekanntenen Größen x_1 , x_2 und x_3 mit der Formel:

$$x_i = \frac{\det(\mathbf{A}_i)}{\det(\mathbf{A})} \quad \text{mit} \quad i \in \{1, 2, 3\} \quad (1.3)$$

berechnen. Darin ergeben sich die Matrizen \mathbf{A}_i aus \mathbf{A} , indem die i -te Spalte von \mathbf{A} durch den Vektor \mathbf{b} ersetzt wird.

- Bestimmen Sie mittels der Funktion `det` die Determinante von \mathbf{A} .
- Ersetzen Sie in \mathbf{A} die i -te Spalte durch \mathbf{b} und berechnen Sie die Determinante der resultierenden Matrix \mathbf{A}_i .
- Die Lösung für x_i ergibt sich als Quotient der Determinante von \mathbf{A}_i und der Determinante von \mathbf{A} .

2. Matrizeninversion

Alternativ kann der Vektor \mathbf{x} über eine Matrizeninversion berechnet werden:

$$\mathbf{x} = \mathbf{A}^{-1} \cdot \mathbf{b}. \quad (1.4)$$

Verwenden Sie für die Inversion von \mathbf{A} den *Matlab*-Befehl `inv`.

3. Gauß'sches Eliminationsverfahren

Matlab bietet zusätzlich die Möglichkeit der *Matrixlinksdivision* `\`. In diesem Fall wird der Gauß-Algorithmus verwendet.

Vergleichen Sie die Ergebnisse der angewandten Verfahren.

d) Anwendungsbeispiel: Approximation

Bei der Kurvenanpassung wird eine mathematische Funktion gesucht, die einen gemessenen Kurvenverlauf möglichst exakt nachbildet. Die Messung eines Prozesses liefert immer eine endliche Anzahl an Messwerten. Je nachdem, wie groß diese Anzahl ist, verwendet man unterschiedliche Ansätze [PL15]:

- **Interpolation:** Liegen wenige Messwerte vor, so wird angestrebt, dass die gesuchte Funktion alle Messwerte durchläuft.
- **Approximation:** Bei einer großen Anzahl an Messwerten wird versucht, diese durch einfache Funktionen möglichst gut zu beschreiben.

Als Beispiel für die Kurvenanpassung soll der Least-Squares-Schätzer als Vertreter der Gruppe Approximationsverfahren näher untersucht werden. Das LS-Verfahren kann zur Schätzung linearer Systemmodelle eingesetzt werden und wird dann auch lineare Regression genannt. Als Messwerte liegen die Eingangsgrößen x_i und y_i mit $i = 1, \dots, n$ vor. Gesucht ist diejenige Gerade

$$\hat{y}_i = m \cdot x_i + b, \quad (1.5)$$

für welche die Abweichung zwischen y_i und \hat{y}_i möglichst klein ist. Hierzu wird als Gütemaß J der quadratische Fehler über alle Messwerte betrachtet:

$$J = \sum_{i=1}^n |y_i - \hat{y}_i|^2 = (\mathbf{y} - \hat{\mathbf{y}})^T \cdot (\mathbf{y} - \hat{\mathbf{y}}) \quad (1.6)$$

mit

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \text{und} \quad \hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix}. \quad (1.7)$$

Dieses Gütemaß soll nun minimiert werden. Dazu wird die Geradengleichung (1.5) zunächst auf Matrizenform gebracht:

$$\left. \begin{array}{l} \hat{y}_1 = m \cdot x_1 + b \\ \hat{y}_2 = m \cdot x_2 + b \\ \vdots \\ \hat{y}_n = m \cdot x_n + b \end{array} \right\} \Leftrightarrow \underbrace{\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix}}_{\hat{\mathbf{y}}} = \underbrace{\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix}}_{\mathbf{M}} \cdot \underbrace{\begin{bmatrix} m \\ b \end{bmatrix}}_{\mathbf{p}}. \quad (1.8)$$

Wird $\hat{\mathbf{y}} = \mathbf{M} \cdot \mathbf{p}$ nach (1.8) in (1.6) eingesetzt, ergibt sich das Gütemaß zu:

$$J = \mathbf{y}^T \mathbf{y} - 2 \mathbf{y}^T \mathbf{M} \mathbf{p} + \mathbf{p}^T \mathbf{M}^T \mathbf{M} \mathbf{p}. \quad (1.9)$$

Bekannte Größen sind darin der Messvektor \mathbf{y} und die Matrix \mathbf{M} . Der Parametervektor \mathbf{p} setzt sich aus der unbekanntenen Geradensteigung m und dem unbekanntenen y -Achsenabschnitt b zusammen. Dieser Vektor soll nun so bestimmt werden, dass das Gütemaß J minimal wird. Hierzu wird J nach \mathbf{p} abgeleitet und das Ergebnis zu null gesetzt:

$$\frac{\partial J}{\partial \mathbf{p}} = -2 \mathbf{M}^T \mathbf{y} + 2 \mathbf{M}^T \mathbf{M} \mathbf{p} \stackrel{!}{=} 0. \quad (1.10)$$

Auflösen nach \mathbf{p} ergibt den gesuchten Parametervektor:

$$\mathbf{p} = (\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T \mathbf{y}. \quad (1.11)$$

Gegeben seien nun folgende Messwerte x_i und y_i :

i	1	2	3	4	5	6	7	8	9	10	11
x_i	0	1	2	3	4	5	6	7	8	9	10
y_i	0,2	1,3	2,2	2,9	3,8	5,1	6,7	7,5	8,3	9,6	10,4

- Bestimmen Sie nach der Least-Squares-Methode diejenige Gerade $\hat{y} = c + mx$, die den kleinsten quadratischen Fehler liefert.
- Plotten Sie die Messpunkte und die Lösungsgerade in ein Schaubild – die Messpunkte als Kreise und die Gerade als durchgezogene Linie. Beschriften Sie die Achsen und fügen Sie eine Legende hinzu.

Aufgabe 1.3: ARMA-Modell in Simulink

Gegeben sei ein lineares, zeitkontinuierliches System, das durch folgende Differentialgleichung beschrieben ist:

$$\ddot{y}(t) = \dot{u}(t) - 4\dot{y}(t) - 4y(t) + u(t). \quad (1.12)$$

Darin ist $u(t)$ die Eingangsgröße und $y(t)$ die Ausgangsgröße. In Abb. 1.24 ist das System in ARMA-Struktur dargestellt [PJ15].

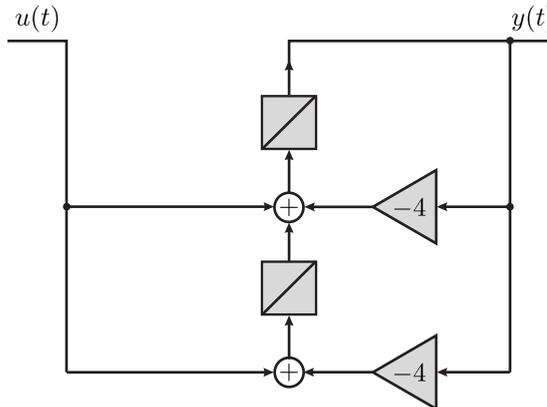


Abbildung 1.24. ARMA-Struktur des Systems.

- a) Dieses System soll nun unter *Simulink* aufgebaut werden. Gehen Sie dazu wie folgt vor:
1. Geben Sie `simulink` am *Matlab*-Prompt ein. Der Simulink Library Browser wird geöffnet. Über `File` → `New` → `Model` können Sie ein leeres *Simulink*-Modell erstellen.
 2. Bauen Sie das System in der ARMA-Struktur nach Abb. 1.24 auf. Die benötigten Blöcke finden Sie im Simulink Library Browser.

3. Wählen Sie als Eingangsgröße eine Sprungfunktion, die bei $t = 1$ s von 0 auf 1 springt. Verwenden Sie hierzu den *Simulink*-Block Step.
4. Fügen Sie zur Visualisierung den Block Scope in Ihr *Simulink*-Modell ein. Öffnen Sie den Scope durch Doppelklick. Klicken Sie auf das Symbol Parameters (zweites von links oben) und geben Sie unter General bei Number of axes 2 ein. Verbinden Sie den ersten Eingang des Scope mit dem Eingangssignal u und den zweiten mit dem Ausgangssignal y .
5. Starten Sie die Simulation (Simulation → Run).

b) Ist das System stabil? Wenn ja, welcher stationäre Endwert wird erreicht?

c) Alternativ kann die Übertragungsfunktion direkt als *Simulink*-Block eingebunden werden. Die Übertragungsfunktion des Systems lautet:

$$G(s) = \frac{Y(s)}{X(s)} = \frac{s + 1}{s^2 + 4s + 4}. \quad (1.13)$$

- Erweitern Sie Ihr *Simulink*-Modell um den Block Transfer Fcn mit der entsprechenden Übertragungsfunktion $G(s)$.
- Führen Sie den Ausgang der aufgebauten ARMA-Struktur und den der Übertragungsfunktion auf einen Mux und verbinden Sie diesen mit dem Scope.

Simulieren Sie noch einmal das Modell und vergleichen Sie die Ergebnisse der beiden Systemstrukturen.

d) Bisher wurde mit zeitkontinuierlichen Systemen gearbeitet. Nun soll die zeitdiskrete Übertragungsfunktion eingesetzt werden. *Matlab* bietet hierzu die Möglichkeit, zeitkontinuierliche Übertragungsfunktionen direkt in das zeitdiskrete Äquivalent umzuwandeln. Gehen Sie dazu wie folgt vor:

- Ergänzen Sie ihr *Simulink*-Modell um zwei To Workspace-Blöcke. Durch Doppelklick können Sie die Blöcke öffnen und die Eigenschaften ändern. Führen Sie die Eingangsgröße auf den ersten Block und ändern Sie den Namen auf u . Führen Sie analog die Ausgangsgröße der Übertragungsfunktion auf den zweiten Block und bezeichnen Sie ihn als y . Ändern Sie bei beiden Blöcken das Save format von Structure auf Structure With Time.
- Nehmen Sie unter Simulation → Model Configuration Parameters die folgenden Einstellungen vor:
 - Wechseln Sie unter Solver options von variable-step auf fixed-step und von discrete auf ode5.
 - Geben Sie unter Fixed-step size die Abtastzeit $1e - 3$ ein.

Starten Sie die Simulation.

- Ein- und Ausgangsgröße u bzw. y liegen jetzt im Workspace vor und können vom Command Window aus verwendet werden. Die beiden Größen wurden als struct abgelegt, der neben den Abtastwerten (`u.signals.values`) auch den zugehörigen Zeitvektor (`u.time`) enthält.

- Wechseln Sie in das Command Window. Erzeugen Sie noch einmal die Übertragungsfunktion $G(s)$ mit Hilfe der *Matlab*-Funktion `tf` (über `help tf` erhalten Sie Hinweise zur Benutzung).
- Über die Funktion `c2d` können Sie die erzeugte kontinuierliche Übertragungsfunktion diskretisieren. Wählen Sie die Abtastzeit $t_A = 1$ ms.
- Die zeitdiskrete Übertragungsfunktion können Sie nun auch direkt im Command Window mit der Funktion `y2 = filter(B,A,u)` simulieren. Dieser Funktion müssen Sie die Koeffizienten der Übertragungsfunktion übergeben. Haben Sie Ihre Übertragungsfunktion als G_z bezeichnet, dann ist das Zählerpolynom in dem Vektor `Gz.num{1}` abgelegt und das Nennerpolynom in `Gz.den{1}`. Übergeben Sie der Funktion `filter` die aus dem *Simulink*-Modell ausgelesene Eingangsgröße und berechnen Sie die Ausgangsgröße.
- Lassen Sie sich in einem Plot die Ausgangsgröße, die das *Simulink*-Modell liefert und die Ausgangsgröße, die Sie über die Funktion `filter` erhalten haben, anzeigen. Fügen Sie eine Legende ein und beschriften Sie die Achsen. Vergleichen Sie die beiden Ergebnisse.

Aufgabe 1.4: Öffnen und Ausführen von VIs in LabVIEW

a) Sehen Sie sich das VI *Signal Generation and Processing.vi* zunächst genauer an und führen Sie es sodann aus. Ändern Sie im Feld *Signal Generation* die Frequenz und Art der Eingangssignale; die entsprechenden Auswirkungen können Sie direkt im Graphen rechts mitverfolgen. Modifizieren Sie nun im Feld *Signal Processing* die Fensterungs- und Filtereinstellungen. Haben Sie das VI und seine Optionen unter die Lupe genommen, so beenden Sie seine Ausführung durch einen Klick auf die Schaltfläche *STOP*.

1. Starten Sie *LabVIEW*. Es öffnet sich das Dialogfenster.
2. Sie können das VI über die Schaltfläche *Öffnen* starten. Der Pfad lautet `D:\PDSV\Versuch_1`.
3. Klicken Sie im Frontpanel in der Symbolleiste auf die Schaltfläche *Ausführen*, um das VI zu starten. Dieses VI dient zur Filterung und Fensterung eines erzeugten Signals. Zudem ermittelt dieses Beispiel auch das Leistungsspektrum für das erzeugte Signal. Die Analyseergebnisse werden schließlich in den Graphen auf der rechten Seite des Frontpanels dargestellt.
4. Mithilfe des Bedienwerkzeugs können Sie die Einstellungen in den Sektionen *Signal Generation* und *Signal Processing* anpassen. Klicken Sie hierzu auf die Pfeilsymbole der betreffenden Bedienelemente. Zum Einstellen der Frequenzen ziehen Sie die Schieberegler mit der Maus nach oben oder unten.
5. Über die Schaltfläche *More Info* bzw. die Taste `F5` erhalten Sie mehr Informationen zu den Analysefunktionen.

6. Klicken Sie auf die Schaltfläche STOP oder drücken Sie die Taste F4, um die Ausführung des VIs zu beenden.
 7. Öffnen Sie über den Menüpunkt Fenster → Blockdiagramm anzeigen bzw. die Tastenkombination Strg+E das Blockdiagramm des VIs Signal Generation and Processing. Im Blockdiagrammfenster sehen Sie einige der elementaren Blockdiagrammobjekte wie etwa SubVIs, Funktionen und Strukturen.
 8. Kehren Sie über den Menüpunkt Fenster → Frontpanel anzeigen bzw. die Tastenkombination Strg+E zum Frontpanel zurück.
 9. Schließen Sie das VI, **ohne die Änderungen zu speichern**.
- b) Starten Sie das VI Debug Exercise (Main). Setzen Sie die einzelnen Werkzeuge und Optionen zur Fehlersuche nach Abschn. 1.4.3.2 ein und korrigieren Sie die Fehler.

Aufgabe 1.5: Berechnung des Flächeninhaltes eines Kreises

Diese Übung ist zwar nicht schwierig, sie wird allerdings einige Zeit in Anspruch nehmen. Es ist ratsam, sich genügend Zeit zu nehmen, um sich gut mit der *LabVIEW*-Umgebung vertraut zu machen.

- a) Erstellen Sie ein VI, das den Flächeninhalt eines Kreises berechnet. Verwenden Sie dazu die Formel $A = \pi d^2/4$, wobei A den Flächeninhalt und d den Durchmesser des Kreises bezeichnen.
1. Öffnen Sie ein neues, leeres VI.
 2. Erstellen Sie auf dem Frontpanel eine Eingabe für den Durchmesser d und beschriften Sie diesen: Rechtsklick im Frontpanel, Express → Numerische Bedienelemente → Numerisches Bedienelement.
 3. Erstellen Sie ebenfalls eine Ausgabe für den Flächeninhalt und beschriften Sie diese: Rechtsklick im Frontpanel, Express → Numerische Anzeigeelemente → Numerisches Anzeigeelement.
 4. Wechseln Sie ins Blockdiagramm: Strg+E oder Fenster → Blockdiagramm anzeigen.
 5. Erstellen Sie das Programm nach obiger Formel:
 1. Rechtsklick ins Blockdiagramm.
 2. Klicken Sie auf die im Fenster links oben befindliche **Pinnadel** (die Funktionspalette bleibt nun auf dem Bildschirm).
 3. Klicken sie als Nächstes auf Programmierung → Numerisch.

Hinweise:

- Erstellen Sie die Quadrierung mittels des Blocks Multiplizieren.
- Teilen Sie dieses Ergebnis durch 4 mittels des Blocks Dividieren. Um die Konstante 4 zu erzeugen, klicken Sie mit der rechten Maustaste auf den freien Eingang des Dividieren-Blockes. Wählen Sie Erstellen → Konstante aus und ändern Sie den Wert 0 in 4.

6. Multiplizieren Sie nun mit π .
Erstellen Sie dazu den Block Multiplizieren und verbinden das Ergebnis der Division mit einem Eingang des Multiplizierers. Fügen Sie als Nächstes die Konstante π ein, indem Sie unter Konstanten π auswählen.
 7. Verbinden Sie den Ausgang des Multiplizierers mit dem Eingang der numerischen Ausgabe.
 8. Wechseln Sie ins Frontpanel zurück und geben einen Wert für den Durchmesser ein.
 9. Starten Sie das VI und überprüfen Sie Ihr Ergebnis.
Für $d = 1$ muss $A = 0,785398$ herauskommen!
- b) Erweitern Sie Ihr VI um eine Schleife, sodass eine kontinuierliche Eingabe des Durchmessers möglich ist.
1. Wechseln Sie dazu ins Blockdiagramm.
 2. Suchen Sie die Funktionenpalette Strukturen.
 3. Wählen Sie die While-Schleife aus. Der Mauszeiger ändert seine Form.
 4. Klicken Sie auf die linke Maustaste, halten diese gedrückt und ziehen Sie damit einen Rahmen um Ihre Funktionsblöcke.
 5. Lassen Sie die Maustaste los – Ihre Funktionsblöcke sind nun in der Schleife.
 6. Klicken Sie mit der rechten Maustaste auf den roten Punkt in der rechten unteren Ecke.
 7. Wählen Sie Bedienelement erstellen.
 8. Wechseln Sie ins Frontpanel.
 9. Starten Sie das VI und geben Sie verschiedene Werte ein. Bestätigen Sie die Eingaben mit „Return“.
 10. Stoppen Sie das VI über die Schaltfläche Stop.
- c) Speichern Sie das VI in Ihrem Gruppenverzeichnis ab.
- d) Probieren Sie verschiedene Eingabe- und Ausgabemöglichkeiten aus (z.B. Verlaufsleiste, Schieber etc.).

Aufgabe 1.6: Vergleich von Matlab, Simulink und LabVIEW

Bei dieser Aufgabe sollen Sie den Umgang mit allen drei Programmen üben und die Unterschiede in der Programmierung näher kennenlernen. Die Programmieranleitungen sind bewusst kurz gehalten, damit Sie sich mit den Programmen intensiver auseinandersetzen.

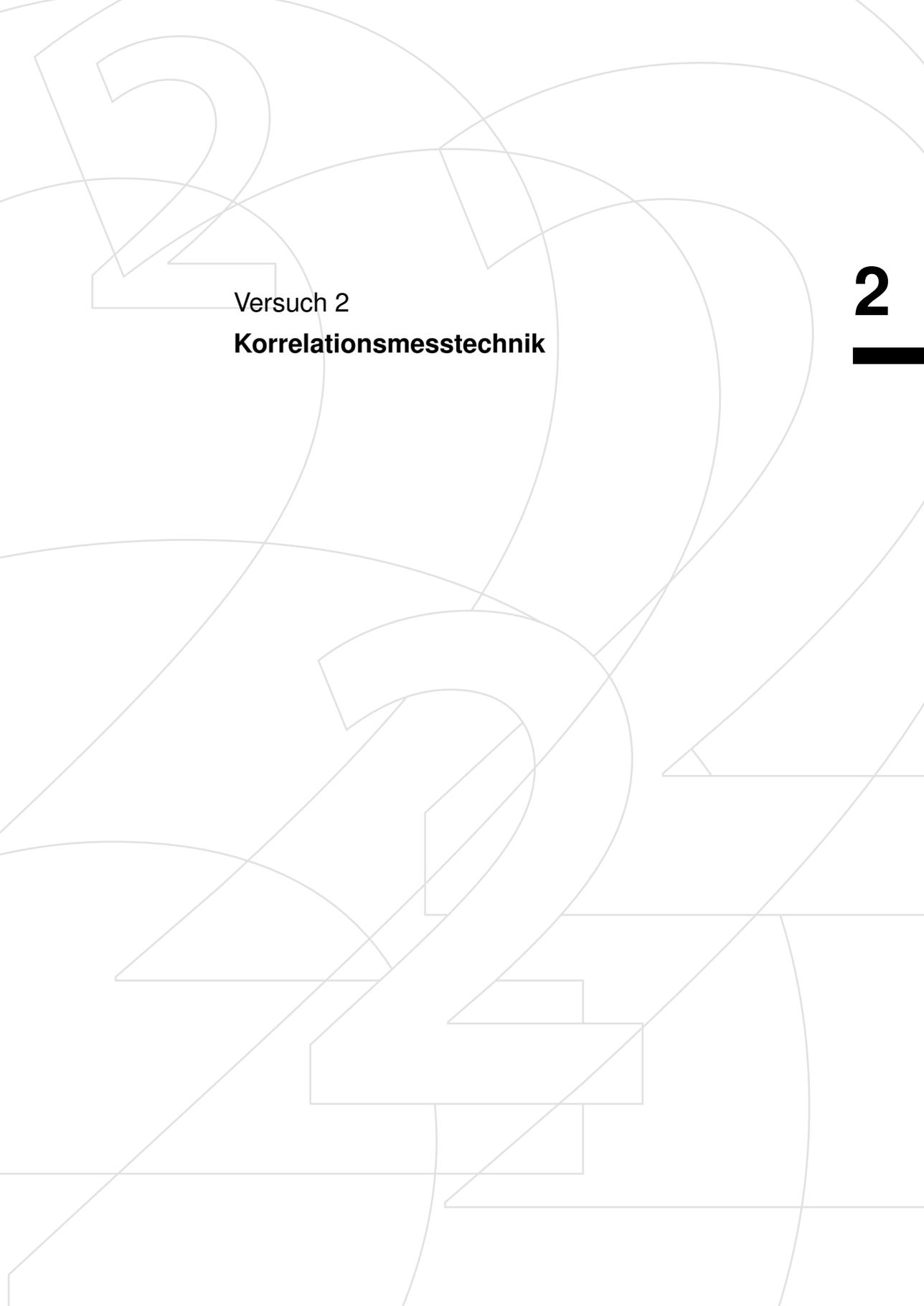
Erstellen Sie ein Programm, mit dessen Hilfe die spektrale Leistungsdichte eines Sinus-Signals mit 100 Hz angezeigt werden kann. Die Abtastrate sei 2000 Hz und die Amplitude des Signals sei 1.

- a) Schreiben Sie ein *m-File* (Matlab-Skript), das obige Aufgabe erfüllt.

1. Erstellen Sie einen Zeitvektor, der in Schritten von 0,5 ms Werte zwischen 0 s und 0,06 s beinhaltet. Die 0,5 ms-Schritte entsprechen der vorgegebenen Abtastrate von 2000 Hz.
 2. Erstellen Sie den Datenvektor für die Sinus-Funktion mit 100 Hz.
 3. Ein Graph soll das Signal darstellen. Wählen Sie eine Überschrift und beschriften Sie die Achsen.
 4. Berechnen Sie die FFT des Signals (FFT-Länge sei 512).
 5. Berechnen Sie die Leistungsdichte und normieren Sie diese mit 512.
 6. Erstellen Sie einen geeigneten Frequenzvektor für die Darstellung. Die Länge des Vektors sei 0 : 256. Überlegen Sie sich noch eine sinnvolle Normierung (Tipp: Frequenzauflösung des FFT-Spektrums beachten).
 7. Ein Diagramm soll die Leistungsdichte im positiven Frequenzbereich darstellen (Beschriftung).
Lassen Sie sich nur die Werte 1 : 257 der Leistungsdichte anzeigen.
 8. Abspeichern und ausführen.
- b) Erstellen Sie eine *Simulink*-Simulation, die obige Aufgabe erfüllt.
1. Erstellen Sie ein Modell mit den beiden Blöcken Sine Wave und Power Spectral Density.
 2. Konfigurieren Sie die Blöcke geeignet.
Hinweis zum Block Sine Wave: Frequenz und Abtastrate geeignet wählen.
Hinweis zum Block Power Spectral Density: Passen Sie die Abtastrate entsprechend an!
 3. Speichern.
 4. Simulieren Sie das Modell bis zu einer Stop Time von 3.
- c) Erstellen Sie ein *LabVIEW*-VI, das obige Aufgabe erfüllt.
1. Verwenden Sie die Blöcke Signal simulieren, Spektrum ermitteln und Signalverlaufsgraph.
 2. Verwenden Sie eine While-Schleife mit Stop-Knopf.
 3. Konfigurieren Sie die Blöcke.
Hinweis: Abtastung, Signalfrequenz und Art des Spektrums müssen noch angepasst werden.
 4. Speichern.
 5. Starten Sie das VI und benutzen Sie die Zoom-Funktion des Signalverlaufsgraphen, um das Signal detaillierter zu betrachten.
- d) Beschreiben Sie die unterschiedlichen Arten der Programmierung.
- e) Welche Vor-/Nachteile haben die einzelnen Programmierarten?
- f) Wo liegen Ihrer Meinung nach die Stärken und Schwächen der Programme?

1.7 Literatur

- [PJ15] FERNANDO PUENTE LEÓN und HOLGER JÄKEL: *Signale und Systeme*. De Gruyter Oldenbourg, Berlin, 6. Auflage, 2015.
- [PL15] FERNANDO PUENTE LEÓN: *Messtechnik – Systemtheorie für Ingenieure und Informatiker*. Springer, Berlin Heidelberg, 10. Auflage, 2015.



Versuch 2

Korrelationsmesstechnik

2

2

2 Korrelationsmesstechnik

2.1	Momente und Korrelationsfunktionen	45
2.1.1	Korrelationsfunktionen stationärer Prozesse	47
2.1.2	Korrelationsfunktionen ergodischer Prozesse	47
2.2	Signalklassen	47
2.3	Messung von Korrelationsfunktionen	48
2.4	Korrelationsfunktion und Leistungsdichte	49
2.5	Korrelation abgetasteter Signale	50
2.6	Schätzung von AKF und KKF mittels der FFT	53
2.7	Polaritätskorrelation	55
2.8	Versuchsdurchführung	55
2.8.1	Erfassung physikalischer Größen in Matlab	56
2.8.2	Einstellungen der Programmoberfläche	56
2.8.2.1	Datenerfassung	57
2.8.2.2	Einstellungen zur Signalverarbeitung	58
2.8.2.3	Mathematische Funktionen	58
2.8.2.4	Schaubilder	59
2.9	Versuchsvorbereitende Aufgaben	59
2.10	Versuche an der Förderbandanlage	60
2.11	Literatur	65

2 Korrelationsmesstechnik

Korrelationsverfahren lassen sich zur berührungslosen Laufzeitmessung nutzen. Dies wird anhand eines Förderbands gezeigt, das Schüttgut transportiert. Über dem Förderband sind zwei Sensoren angebracht, die zeitlich versetzt das vom Schüttgut reflektierte Licht erfassen. Durch eine Analyse der aufgenommenen Signale mit Hilfe von Korrelationsfunktionen wird auf die Laufzeit des Schüttguts zwischen den beiden Sensoren und damit auf die Geschwindigkeit des Förderbandes geschlossen.

Vor der Versuchsdurchführung, die Gegenstand von Abschn. 2.8 ist, wird auf die erforderliche Theorie zur Messung von Korrelationsfunktionen eingegangen. Im Fokus stehen dabei die unterschiedlichen Korrelationsfunktionen sowie ihre Berechnung und ihre Eigenschaften, wobei besonderes Augenmerk auf den statistischen Eigenschaften liegt. Die folgenden Ausführungen basieren auf [PL15].

In der Versuchsdurchführung werden die erläuterten Eigenschaften vertieft sowie auf praktische Aspekte der Nutzung von Korrelationsfunktionen zur Laufzeitbestimmung eingegangen.

2.1 Momente und Korrelationsfunktionen

Unter dem m -ten **Moment** eines stochastischen Prozesses $x(t)$ wird der folgende Erwartungswert verstanden [PL15]:

$$\mu_{x,m}(t) = E\{x^m(t)\} = \int_{-\infty}^{\infty} x^m f_{x(t)}(x) dx. \quad (2.1)$$

Dabei bezeichnet $f_{x(t)}(x)$ die Wahrscheinlichkeit, mit der ein Signalwert x zum Zeitpunkt t auftritt.

Der **Mittelwert** $\mu_x(t)$ eines stochastischen Prozesses $x(t)$ ist das erste Moment:

$$\mu_x(t) = E\{x(t)\} = \int_{-\infty}^{\infty} x f_{x(t)}(x) dx. \quad (2.2)$$

Die Momente der Statistik 2. Ordnung ergeben sich durch das „Festhalten“ der Signale zu zwei Zeitpunkten t_1 und t_2 . Wird dabei ein stochastischer Prozess zu zwei Zeitpunkten betrachtet, so führt dies auf die **Autokorrelationsfunktion** (AKF)

$$r_{xx}(t_1, t_2) = E\{x(t_1) x^*(t_2)\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_1 x_2^* f_{x(t_1)x(t_2)}(x_1, x_2) dx_1 dx_2 \quad (2.3)$$

als zweites Moment der Statistik 2. Ordnung und auf die **Autokovarianzfunktion** (AKV)

$$C_{xx}(t_1, t_2) = E\{(\mathbf{x}(t_1) - E\{\mathbf{x}(t_1)\})(\mathbf{x}(t_2) - E\{\mathbf{x}(t_2)\})^*\} \quad (2.4)$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x_1 - \mu_x(t_1))(x_2 - \mu_x(t_2))^* f_{\mathbf{x}(t_1)\mathbf{x}(t_2)}(x_1, x_2) dx_1 dx_2 \quad (2.5)$$

als zweites zentrales Moment der Statistik 2. Ordnung. Die Autokovarianzfunktion $C_{xx}(t_1, t_2)$ unterscheidet sich von der Autokorrelationsfunktion $r_{xx}(t_1, t_2)$ lediglich um das Produkt der Mittelwerte:

$$C_{xx}(t_1, t_2) = r_{xx}(t_1, t_2) - \mu_x(t_1)\mu_x^*(t_2). \quad (2.6)$$

Werden zwei verschiedene stochastische Prozesse zu den Zeitpunkten t_1 und t_2 betrachtet, so ergeben sich als zweite Momente der Statistik 2. Ordnung die **Kreuzkorrelationsfunktion** (KKF)

$$r_{xy}(t_1, t_2) = E\{\mathbf{x}(t_1)\mathbf{y}^*(t_2)\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x y^* f_{\mathbf{x}(t_1)\mathbf{y}(t_2)}(x, y) dx dy \quad (2.7)$$

und die **Kreuzkovarianzfunktion** (KKV):

$$C_{xy}(t_1, t_2) = E\{(\mathbf{x}(t_1) - E\{\mathbf{x}(t_1)\})(\mathbf{y}(t_2) - E\{\mathbf{y}(t_2)\})^*\} \quad (2.8)$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \mu_x(t_1))(y - \mu_y(t_2))^* f_{\mathbf{x}(t_1)\mathbf{y}(t_2)}(x, y) dx dy. \quad (2.9)$$

Auch die Kreuzkovarianzfunktion $C_{xy}(t_1, t_2)$ unterscheidet sich von der Kreuzkorrelationsfunktion $r_{xy}(t_1, t_2)$ nur um das Produkt der Mittelwerte:

$$C_{xy}(t_1, t_2) = r_{xy}(t_1, t_2) - \mu_x(t_1)\mu_y^*(t_2). \quad (2.10)$$

Durch die Betrachtung eines Prozesses zu verschiedenen Zeitpunkten ist es möglich, Aussagen über die „Zufälligkeit“ dieses Prozesses zu machen (Erhaltungstendenz). Durch die Betrachtung verschiedener Signale zu unterschiedlichen Zeiten kann dagegen die Ähnlichkeit der zugehörigen Prozesse beurteilt werden: Je ähnlicher die Prozesse zu den gewählten Zeitpunkten sind, desto größer wird die Kreuzkorrelation. Dies wird bei der Messung von Laufzeiten mittels Korrelationsverfahren ausgenutzt.

2.1.1 Korrelationsfunktionen stationärer Prozesse

Für **stationäre** Prozesse sind die Momente der Statistik 1. Ordnung unabhängig von der Zeit t . So gilt für den Erwartungswert:

$$\mu_x(t) = E\{x(t)\} = \mu_x = \text{const.} \quad \text{für beliebige Zeitpunkte } t. \quad (2.11)$$

Die Momente der Statistik 2. Ordnung hängen nur von der Zeitdifferenz $\tau = t_1 - t_2$ ab. Für die Kreuzkorrelationsfunktion gilt dann beispielsweise:

$$r_{xy}(t_1, t_2) = E\{x(t_1) y^*(t_2)\} = E\{x(t_1) y^*(t_1 - \tau)\} \quad (2.12)$$

$$= E\{x(t_1 + \tau) y^*(t_1)\} = r_{xy}(\tau). \quad (2.13)$$

2.1.2 Korrelationsfunktionen ergodischer Prozesse

Ein stationärer Zufallsprozess heißt (streng) **ergodisch**, wenn seine Erwartungswerte mit der Wahrscheinlichkeit eins mit den entsprechenden Zeitmittelwerten einer beliebigen Musterfunktion $x(t)$ übereinstimmen. Für die Kreuzkorrelationsfunktion zweier ergodischer Prozesse $x(t)$ und $y(t)$ mit den Musterfunktionen $x(t)$ bzw. $y(t)$ gilt entsprechend:

$$r_{xy}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T x(t + \tau) y^*(t) dt. \quad (2.14)$$

In der Praxis wird diese Tatsache ausgenutzt, um Erwartungswerte und Korrelationsfunktionen überhaupt berechnen zu können. Wird für einen Zufallsprozess $x(t)$ Ergodizität angenommen, dann lassen sich die verschiedenen Momente einfach über die zeitlichen Mittelwerte einer einzigen Musterfunktion $x(t)$ bestimmen.

In praktischen Anwendungen lassen sich die Korrelationsfunktionen allerdings auch unter der Annahme ergodischer Prozesse nicht exakt ermitteln. Zum einen stehen bei der Berechnung immer nur abgetastete Signale zur Verfügung und zum anderen sind die Messungen immer auf ein endliches Zeitintervall begrenzt. Daher lassen sich lediglich Schätzwerte für die Korrelationsfunktionen angeben. Die Vorgehensweise wird in Abschn. 2.5 beschrieben.

2.2 Signalklassen

Ausgehend von einem allgemeinen stochastischen Prozess, der beliebige Zeitfunktionen als Musterfunktionen beinhalten kann, lassen sich die Musterfunktionen in drei Klassen gliedern [PJ15, PL15]:

- **Energiesignale** sind beschränkte, stückweise stetige Signale, für welche die folgende Bedingung erfüllt ist:

$$\|x(t)\|^2 = \int_{-\infty}^{\infty} x(t) x^*(t) dt < \infty. \quad (2.15)$$

- **Leistungssignale** sind beschränkte, stückweise stetige Signale, für welche das Integral (2.15) divergiert, aber der folgende Grenzwert existiert:

$$\|x(t)\|^2 = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T x(t) x^*(t) dt < \infty. \quad (2.16)$$

- **Sonstige Signale** sind alle Signale, für welche die Integrale (2.15) und (2.16) nicht existieren oder welche nicht stückweise stetig oder unbeschränkt sind.

Für Leistungssignale ist die Kreuzkorrelationsfunktion über das Innenprodukt definiert:

$$r_{xy}(\tau) = \langle x(t+\tau), y(t) \rangle = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T x(t+\tau) y^*(t) dt. \quad (2.17)$$

Diese Definition ist identisch mit der Berechnung der Korrelation ergodischer Prozesse nach (2.14). Dies ist nicht verwunderlich, da Leistungssignale Musterfunktionen ergodischer Prozesse sein können.

Obwohl Energiesignale keine Musterfunktionen ergodischer Prozesse sein können, lässt sich auch für Energiesignale eine Korrelationsfunktion über das Innenprodukt von Energiesignalen angeben:

$$r_{xy}^E(\tau) = \langle x(t+\tau), y(t) \rangle = \int_{-\infty}^{\infty} x(t+\tau) y^*(t) dt < \infty. \quad (2.18)$$

Der Zusatz „E“ in $r_{xy}^E(\tau)$ kennzeichnet die abweichende Berechnungsvorschrift und die Tatsache, dass es sich hier nicht mehr um eine Korrelationsfunktion im stochastischen Sinn handelt. Die Korrelationsfunktion für Energiesignale $r_{xy}^E(\tau)$ konvergiert auch dann, wenn nur eines der beiden Signale ein Energiesignal ist [PL15]. Das liegt daran, dass das Energiesignal den Integranden für große Zeiten t zu null werden lässt.

2.3 Messung von Korrelationsfunktionen

Entsprechend Abschn. 2.1 sind Korrelationsfunktionen mathematisch – und nicht durch eine physikalische Messvorschrift – definiert. Durch Messungen lassen sich daher immer nur Schätzwerte bestimmen. Hierbei treten Schwierigkeiten auf, die in

dem Gegensatz zwischen der mathematischen Definition und den messtechnischen Möglichkeiten begründet liegen.

1. Aus der Schar der Musterfunktionen eines Zufallsprozesses ist in der Regel nur eine einzelne Musterfunktion $x(t)$ für eine Messung verfügbar.
2. Anstelle des Scharmittelwertes muss folglich ein Zeitmittelwert gebildet werden.
3. Bei einer Auswertung der Messsignale in Echtzeit liegen künftige Werte der Signale nicht vor. Entsprechend lassen sich die momentanen Signalwerte nur mit vergangenen Werten, nicht jedoch mit positiven Werten verknüpfen (Kausalität).
4. Die Messung ist auf ein endliches Zeitintervall beschränkt. Dies geht mit einer Fensterung der Signale einher.

Der erste Punkt dieser Einschränkung bereitet formal keine Schwierigkeiten, wenn der zugrunde liegende Prozess ergodisch ist. In diesem Fall kann der gesamte Prozess durch eine beliebige Musterfunktion vertreten werden. Darf für den Prozess keine Ergodizität angenommen werden, so schließt das nicht aus, dass einzelne Musterfunktionen – aber eben nicht alle möglichen Musterfunktionen – repräsentativ für den Prozess sind. In diesem Fall ist eine geeignete Auswahl notwendig.

Der zweite Punkt hängt eng mit dem ersten zusammen. Für einzelne Musterfunktionen lassen sich nur Zeitmittelwerte bestimmen, was bei ergodischen Prozessen keine Einschränkung darstellt. Es bedeutet aber, dass Korrelationsfunktionen nur für stationäre Prozesse gemessen werden können. Nur bei Energiesignalen lässt sich eine Korrelation nicht stationärer Signale mittels des Integrals (2.18) vornehmen.

Die Echtzeitmessung verlangt negative Zeitverschiebungen τ . Dies stellt insofern keine Einschränkung dar, weil die Symmetrieeigenschaften der Korrelationsfunktionen bekannt sind [PL15].

Der vierte Punkt stellt das gravierendste Problem bei der Messung aller statistischen Kenngrößen dar: Die Messung muss nach endlicher Zeit abgebrochen werden. Bei der Messung von Korrelationen bedeutet dies, dass nur Kurzzeitkorrelationsfunktionen gemessen werden können.

2.4 Korrelationsfunktion und Leistungsdichte

Für Musterfunktionen stationärer Prozesse existiert im Allgemeinen das Fourier-Integral nicht, da sie für $|t| \rightarrow \infty$ nicht abklingen. Um diese Prozesse dennoch im Frequenzbereich beschreiben zu können, wird die Fourier-Transformation der Korrelationsfunktion betrachtet. Hierbei werden die Eigenschaften des Prozesses zuerst durch Erwartungswertbildung im Zeitbereich ermittelt und dann durch Fourier-Transformation im Frequenzbereich dargestellt. Dadurch lässt sich beschreiben, wie stochastische Prozesse durch LTI-Systeme verändert werden.

Die Fourier-Transformierte der Autokorrelationsfunktion $r_{xx}(\tau)$ eines schwach stationären Prozesses wird als **Autoleistungsdichtespektrum** bezeichnet:

$$S_{xx}(f) = \int_{-\infty}^{\infty} r_{xx}(\tau) e^{-j2\pi f\tau} d\tau. \quad (2.19)$$

Die Fourier-Transformierte der Kreuzkorrelationsfunktion $r_{xy}(\tau)$ zweier schwach stationärer Zufallsprozesse heißt **Kreuzleistungsdichtespektrum**:

$$S_{xy}(f) = \int_{-\infty}^{\infty} r_{xy}(\tau) e^{-j2\pi f\tau} d\tau. \quad (2.20)$$

Die Umkehrung ergibt sich nach den Regeln der Fourier-Transformation:

$$r_{xx}(\tau) = \int_{-\infty}^{\infty} S_{xx}(f) e^{j2\pi f\tau} df, \quad (2.21)$$

$$r_{xy}(\tau) = \int_{-\infty}^{\infty} S_{xy}(f) e^{j2\pi f\tau} df. \quad (2.22)$$

Der Begriff **Leistungsdichte** leitet sich davon ab, dass $S_{xx}(f)$ die Leistungsverteilung über den Frequenzen f darstellt. Integriert man $S_{xx}(f)$ über einen Frequenzbereich $f_1 \leq f \leq f_2$, so ist das Ergebnis proportional zur Leistung des Prozesses in diesem Bereich. Damit ergibt sich die **Leistung** eines Zufallsprozesses $x(t)$ zu:

$$P_x = E\{x^2(t)\} = r_{xx}(0) = \int_{-\infty}^{\infty} S_{xx}(f) df. \quad (2.23)$$

Korrekterweise müsste noch ein **Energiedichtespektrum** für die Fourier-Transformierte der Korrelationsfunktion für Energiesignale eingeführt werden. Die Unterscheidung in Energie- und Leistungsdichtespektrum wird im weiteren Verlauf allerdings unterlassen, da im Spektralbereich nicht mehr festgestellt werden kann, ob das Spektrum von einem Energie- oder Leistungssignal stammt. Im Folgenden wird daher nur noch von Leistungsdichtespektren gesprochen.

2.5 Korrelation abgetasteter Signale

Heutzutage wird die Korrelation meist mit einem Rechner ermittelt. Die Signale stehen nur zu diskreten Zeitpunkten zur Verfügung. In Analogie zu (2.13) ist die Kreuzkorrelationsfunktion eines abgetasteten stationären Prozesses wie folgt definiert:

$$r_{xy}[k] = E\{x_{n+k} y_n^*\}. \quad (2.24)$$

Unter der Annahme ergodischer Prozesse berechnet sich die Kreuzkorrelation aus einzelnen Musterfunktionen x_n und y_n zu

$$r_{xy}[k] = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=-N/2}^{N/2-1} x_{n+k} y_n^* \quad (2.25)$$

Da Messungen immer nach endlicher Zeit abgebrochen werden, können nur **Kurzzeitkorrelationsfunktionen** gemessen werden. Es lässt sich deshalb nur eine Schätzung $\hat{r}_{xy}[k]$ der Korrelationsfunktion bestimmen, die selbst wieder zufällig ist. Dabei stehen nur N zeitdiskrete Abtastwerte der Signale zur Verfügung:

$$\hat{r}_{xy}[k] = \frac{1}{N} \sum_{n=0}^{N-1-k} x_{n+k} y_n^* \quad \text{mit} \quad 0 \leq k < N. \quad (2.26)$$

Um eine Schätzung der Kreuzkorrelationsfunktion für negative Verschiebungen zu erhalten, wird die folgende Symmetrieeigenschaft ausgenutzt:

$$r_{xy}[-k] = E\{x_{n-k} y_n^*\} = E\{x_n y_{n+k}^*\} = r_{yx}^*[k]. \quad (2.27)$$

Damit lässt sich die folgende Schätzung für negative Zeitverschiebungen $-k$ mit $0 \leq k < N$ berechnen:

$$\hat{r}_{xy}[-k] = \hat{r}_{yx}^*[k] = \frac{1}{N} \sum_{n=0}^{N-1-k} y_{n+k}^* x_n. \quad (2.28)$$

Zur Beurteilung der Güte der Schätzung untersucht man deren systematischen Schätzfehler (engl. *bias*) und deren Varianz für eine zunehmende Anzahl N von Abtastwerten. Hierbei interessieren zwei Forderungen [PL15]:

- **Erwartungstreue:** Ein Schätzer \hat{x} heißt **erwartungstreu** (engl. *unbiased*), wenn bei wiederholten Stichproben der wahre Wert $x = E\{x\}$ im Mittel richtig geschätzt wird:

$$E\{\hat{x}\} = E\{x\}. \quad (2.29)$$

- **Konsistenz:** Ein Schätzer \hat{x} heißt **konsistent**, wenn mit wachsendem Stichprobenumfang N der wahre Wert x mit Sicherheit ermittelt werden kann:

$$\lim_{N \rightarrow \infty} \hat{x} = E\{x\} \quad \text{und} \quad \lim_{N \rightarrow \infty} \sigma_{\hat{x}} = 0. \quad (2.30)$$

Der Erwartungswert des KKF-Schätzers (2.26) berechnet sich zu:

$$E\{\hat{r}_{xy}[k]\} = E\left\{\frac{1}{N} \sum_{n=0}^{N-1-k} x_{n+k} y_n^*\right\} \quad (2.31)$$

$$= \frac{1}{N} \sum_{n=0}^{N-1-k} \underbrace{E\{x_{n+k} y_n^*\}}_{= r_{xy}[k]} \quad (2.32)$$

$$= r_{xy}[k] \cdot \frac{1}{N} \sum_{n=0}^{N-1-k} 1 \quad (2.33)$$

$$= \frac{N-k}{N} \cdot r_{xy}[k] \quad \text{für} \quad 0 \leq k < N. \quad (2.34)$$

Der KKF-Schätzer für negative Zeitverschiebungen berechnet sich analog:

$$E\{\hat{r}_{xy}[k]\} = E\{\hat{r}_{yx}[-k]\} = \frac{N+k}{N} \cdot r_{yx}[-k] \quad (2.35)$$

$$= \frac{N+k}{N} \cdot r_{xy}[k] \quad \text{für} \quad -N < k \leq 0. \quad (2.36)$$

Insgesamt gilt somit für die Schätzung der Kreuzkorrelationsfunktion:

$$E\{\hat{r}_{xy}[k]\} = \frac{N-|k|}{N} \cdot r_{xy}[k] \quad \text{für} \quad -N < k < N. \quad (2.37)$$

Sie entspricht der wahren KKF $r_{xy}[k]$ gewichtet mit einem Dreieckfenster. Die Schätzung ist daher **nicht erwartungstreu**. Ihr systematischer Schätzfehler (*Bias*) beträgt:

$$B_k = E\{\hat{r}_{xy}[k]\} - E\{r_{xy}[k]\} = E\{\hat{r}_{xy}[k]\} - r_{xy}[k] = \frac{|k|}{N} r_{xy}[k]. \quad (2.38)$$

Besonders für große Zeitverschiebungen $|k| \rightarrow N-1$ weicht der Erwartungswert des Schätzers vom wahren Wert deutlich ab. Der *Bias* wird dagegen klein, wenn nur kleine Zeitverschiebungen $|k|$ betrachtet werden.

Für die Varianz des Schätzers kann gezeigt werden, dass sie für $N \rightarrow \infty$ gegen null strebt [KK12]. Die Beweisführung ist jedoch umfangreich und soll hier nicht aufgeführt werden. Zusammenfassend lässt sich sagen, dass für Zeitverschiebungen, die klein gegen die Werteanzahl sind, also $|k| \ll N$, sowohl der *Bias* nach (2.38) als auch die Varianz gegen null streben.

Die Schätzung der Kreuzkorrelation nach (2.26) und (2.28) ist also nur für $|k| \ll N$ konsistent. Nach [KK12] ist hierzu die Bedingung

$$|k| < 0,2 \cdot N \quad (2.39)$$

ausreichend. Diese Bedingung für $|k|$ kann in äquivalente Bedingungen für die Zeitverschiebung τ bzw. die Geschwindigkeit v umgeformt werden:

$$|k| < 0,2 \cdot N \quad \Leftrightarrow \quad \tau < 0,2 \cdot \frac{N}{f_A} \quad \Leftrightarrow \quad v > \frac{d \cdot f_A}{0,2 \cdot N}. \quad (2.40)$$

Darin bezeichnet f_A die Abtastfrequenz und d den Abstand der beiden Sensoren.

Eine **erwartungstreue** Schätzung der Kreuzkorrelation kann erzielt werden, wenn nicht auf die gesamte Werteanzahl N normiert wird, sondern auf die Anzahl $N - |k|$ der Werte, die zur Berechnung des aktuellen Wertes herangezogen werden:

$$\hat{r}_{xy}[k] = \begin{cases} \frac{1}{N-|k|} \sum_{n=0}^{N-1-k} x_{n+k} y_n^* & \text{für } 0 \leq k < N, \\ \frac{1}{N-|k|} \sum_{n=0}^{N-1-|k|} x_n y_{n+|k|}^* & \text{für } -N < k \leq 0. \end{cases} \quad (2.41)$$

Analog zur Herleitung von (2.37) folgt für den Erwartungswert von (2.41):

$$E\{\hat{r}_{xy}[k]\} = \frac{N - |k|}{N - |k|} \cdot r_{xy}[k] = r_{xy}[k] \quad \text{für } -N < k < N. \quad (2.42)$$

Diese Schätzung ist somit erwartungstreu.

Nach [KK12] geht die Varianz für eine wachsende Werteanzahl N aber nur dann gegen null, wenn die Zeitverschiebung $|k|$ klein gegenüber der Werteanzahl ist. Der Schätzer ist somit wie auch der Schätzer nach den Gleichungen (2.26) und (2.28) nur unter der Bedingung $|k| \ll N$ konsistent.

2.6 Schätzung von AKF und KKF mittels der FFT

Die Faltung zweier diskreter Funktionen f_n und g_n ist definiert als:

$$h_n = \sum_{k=-\infty}^{\infty} f_{n-k} g_k. \quad (2.43)$$

Diese diskrete Faltung wird auch als gewöhnliche, aperiodische Faltung bezeichnet. Es sei angenommen, dass f_n und g_n nur für $0 \leq n < N$ Werte ungleich null besitzen und ansonsten verschwinden. Es ist klar, dass dann $h_n = 0$ für $n < 0$ und für $n > 2N - 2$ ist. Die direkte Berechnung der Faltungssumme h_n erfordert N^2 Multiplikationen und $(N - 1)^2$ Additionen.

Alternativ kann h_n über den Frequenzbereich berechnet werden, indem zunächst über $2N$ -Punkte-DFTs die Spektren F_k und G_k von f_n bzw. g_n ermittelt werden. Dabei werden die Funktionen f_n und g_n implizit periodisch fortgesetzt. Die inverse DFT von $H_k = F_k \cdot G_k$ ergibt dann im Prinzip die gesuchte Funktion h_n an den Stellen $0 \leq n \leq 2N - 2$.

Diese Berechnungsweise der KKF erfordert die Berechnung dreier FFTs sowie weitere N Multiplikationen. Unter der Voraussetzung, dass N eine Potenz von 2 ist, lässt sich die Zahl der Multiplikationen angeben als:

$$3N \log_2 N + N. \quad (2.44)$$

Bei größeren Datensätzen ist also aus Gründen der Rechenzeiterparnis die indirekte Berechnung der Faltung im Frequenzbereich vorzuziehen.

Bei der Schätzung einer Kreuzkorrelationsfunktion nach (2.26)

$$\hat{r}_{xy}[k] = \frac{1}{N} \sum_{n=0}^{N-1-k} x_{n+k} y_n^* \quad (2.45)$$

$$= \frac{1}{N} \sum_{m=-N+1+k}^0 x_{-m+k} y_{-m}^* \quad (2.46)$$

$$= \frac{1}{N} \sum_{m=-N+1+k}^0 x_{k-m} y_{-m}^* = x_k * y_{-k}^* \quad (2.47)$$

handelt es sich um eine Faltung der Funktion x_k mit der Funktion y_{-k}^* . Die Faltung entspricht einer Multiplikation im Frequenzbereich:

$$\mathcal{F}\{x_k * y_{-k}^*\} = X_n \cdot Y_n^*. \quad (2.48)$$

Man erhält also $\hat{r}_{xy}[k]$ als inverse DFT des Produktes $X_n \cdot Y_n^*$.

Allerdings ist wegen der bei Anwendung des FFT-Algorithmus impliziten periodischen Fortsetzung der Funktionen Vorsicht geboten. Statt der gewünschten Korrelationsfunktion erhält man eine sogenannte **zyklische Korrelationsfunktion**, da aufgrund der periodischen Fortsetzung je nach Größe der Verschiebung k kleinere oder größere Teile der Funktionen korreliert werden, die eigentlich nichts miteinander zu tun haben. Der damit verbundene Fehler wird als **Randeffekt** bezeichnet.

Dieser Effekt ist klein bei Kreuzkorrelationsfunktionen, die schnell abfallen, und kann für

$$k < 0,2 \cdot N \quad (2.49)$$

meist vernachlässigt werden. Der Randeffekt kann auch vollständig umgangen werden, wenn die beiden Datensätze mit Nullen aufgefüllt werden, was als **Zero-Padding** bezeichnet wird: Werden an die Datensätze vom Umfang N jeweils $N - 1$ Nullen angehängt, so bewirken sie die vollständige Elimination von Beiträgen der periodischen Fortsetzung.

2.7 Polaritätskorrelation

Statt mit großer Auflösung zu quantisieren und dann zu korrelieren, kann der Rechenaufwand mit einem zweistufigen Quantisierer und dem sogenannten **Polaritätskorrelator** erheblich verringert werden. Bei mittelwertfreien Signalen ist es ausreichend, die Vorzeichen der beiden Signale zu untersuchen. Nach [PL15] bleibt dabei die Lage des Korrelationsmaximums erhalten. Dadurch reduziert sich die Multiplikation der Signale auf einfache Operationen, die mit Hilfe von logischen Gattern realisiert werden können. Die Polaritätskorrelationsfunktion für abgetastete Signale lautet:

$$\hat{r}_{xy}[k] = \frac{1}{N} \sum_{n=0}^{N-1-k} \text{sign}(x_{n+k}) \cdot \text{sign}(y_n) \quad \text{mit } 0 \leq |k| < N. \quad (2.50)$$

Sind die Signale x_n und y_n nicht mittelwertfrei, so kann statt der Signumfunktion eine Schaltschwelle verwendet werden, die den Umschlag von -1 nach $+1$ definiert.

Die normierte Kreuzkorrelation, welche die Ähnlichkeit zweier zeitlich gegeneinander verschobener Funktionen bewertet, heißt **Kreuzkorrelationskoeffizient**:

$$r_{xy,\text{norm}}[k] = \frac{r_{xy}[k] - \mu_x \mu_y^*}{\sqrt{(r_{xx}[0] - |\mu_x|^2)(r_{yy}[0] - |\mu_y|^2)}}. \quad (2.51)$$

Der Kreuzkorrelationskoeffizient gibt an, wie stark die beiden Signale korreliert sind. Es gilt:

$$-1 \leq r_{xy,\text{norm}}[k] \leq 1. \quad (2.52)$$

Bei mittelwertfreien, normalverteilten Signalen kann für den Zusammenhang von Kreuzkorrelationskoeffizient $\hat{r}_{xy,\text{norm}}[k]$ und Polaritätskorrelation $\hat{r}_{xy}[k]$ die folgende Beziehung hergeleitet werden:

$$\hat{r}_{xy,\text{norm}}[k] = \sin\left(\frac{\pi}{2} \cdot \hat{r}_{xy}[k]\right). \quad (2.53)$$

Die Multiplikation mit $\frac{\pi}{2}$ und die anschließende Berechnung des Sinus kann auf einem Rechner leicht durchgeführt werden, so dass über die Polaritätskorrelationsfunktion der Kreuzkorrelationskoeffizient berechnet werden kann.

2.8 Versuchsdurchführung

Das in Abb. 2.1 dargestellte Förderband transportiert Kies von einem Vorratsbehälter bis zur Abwurfstelle. Zwei optische Sensoren mit dem Abstand d sind in der Höhe h über dem Förderband montiert. Die Sensoren verfügen über eine Infrarotlichtquelle, die für eine gute Beleuchtung des Förderbandes im Bereich des Sensors sorgt. Die Sensoren messen das vom Förderband bzw. dem Schüttgut reflektierte Licht, das von der Beladung des Bandes und der Art des Schüttgutes abhängig ist.

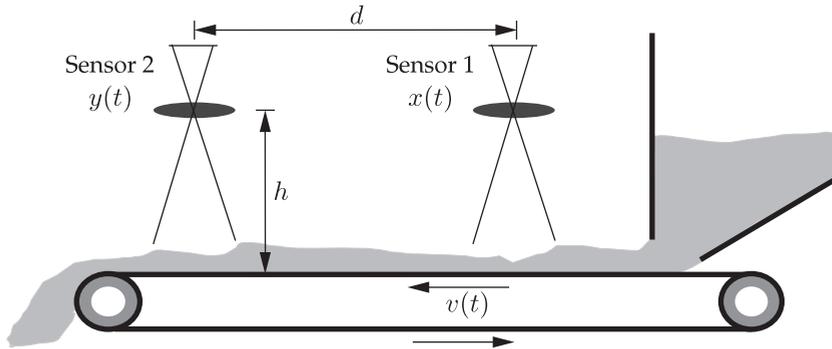


Abbildung 2.1. Förderbandanlage.

Mit Hilfe eines Korrelationsverfahrens soll die Laufzeit zwischen den beiden Sensoren ermittelt werden. Daraus kann dann bei bekanntem Abstand d zwischen den beiden Sensoren die Geschwindigkeit des Förderbandes berechnet werden. In den Versuchsaufgaben werden die einzelnen Korrelationsfunktionen sowie unterschiedliche Berechnungsweisen genauer analysiert. Außerdem wird deutlich gemacht, welche praktischen Einflüsse bei der Durchführung von Korrelationsmesstechnik zu berücksichtigen sind.

2.8.1 Erfassung physikalischer Größen in Matlab

Dieser Versuch zeigt, wie man in *Matlab* ohne zusätzliche A/D-Wandlerkarten physikalische Messgrößen erfassen und weiterverarbeiten kann. Die Datenerfassung erfolgt mit Hilfe einer Soundkarte, wie sie in herkömmlichen PCs zur Verfügung steht. Die beiden Sensorsignale werden der Soundkarte über ihren Stereoingang zugeführt.

Hinweis: Achten Sie darauf, dass das richtige Kabel eingesteckt ist! Nutzen Sie bitte nur den „Line-in“- (blau), nicht jedoch den Mikrofoneingang!

Die Erfassung der Audiosignale erfolgt mit Hilfe der *Matlab*-Funktion `audiorecorder`, die eine Datenstruktur liefert, in der die aufgenommenen Signale, die Abtastfrequenz und die Quantisierung abgelegt sind. Sie dient als Ausgangspunkt für die weitere Signalverarbeitung. Für die Erfassung und Weiterverarbeitung der Daten steht das GUI `Korrelation` zur Verfügung, das im Folgenden näher beschrieben wird.

2.8.2 Einstellungen der Programmoberfläche

Starten Sie das *Matlab*-GUI durch die Eingabe `Korrelation` am *Matlab*-Prompt. Die Programmoberfläche ist in Abb. 2.2 dargestellt.

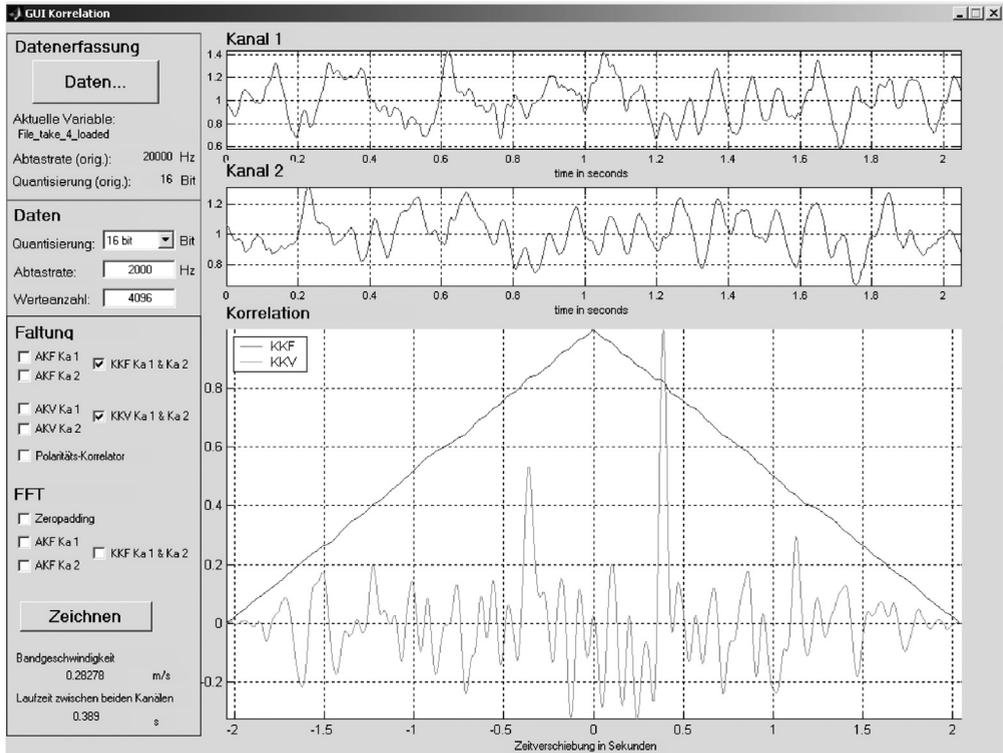


Abbildung 2.2. Benutzeroberfläche des Programms zur Korrelationsmessung.

2.8.2.1 Datenerfassung

Der Dialog zur Datenerfassung wird über die Schaltfläche Daten aufgerufen (Abb. 2.3). Fest vorgegeben sind die Abtastfrequenz $f_A = 20 \text{ kHz}$ und die Quantisierung 16 bit. Die Dauer der Datenerfassung können Sie frei vorgeben, minimal werden aber $2^{17} = 131\,072$ Werte aufgezeichnet.

- Über die Schaltfläche Daten aufzeichnen starten Sie die Messung. Sobald die Messwerte aufgezeichnet sind, schließt sich der Dialog und die Messwerte werden im GUI Korrelation dargestellt. Zusätzlich wird im aktuellen *Matlab*-Verzeichnis ein neuer Ordner Data angelegt, in dem die aufgezeichneten Messdaten als *.mat-File abgespeichert werden.
- Einmal aufgezeichnete Daten können jederzeit über die Schaltfläche Daten aus Datei laden neu geladen werden.
- Über Daten aus Workspace laden können außerdem Messdaten direkt aus dem *Matlab*-Workspace geladen werden.

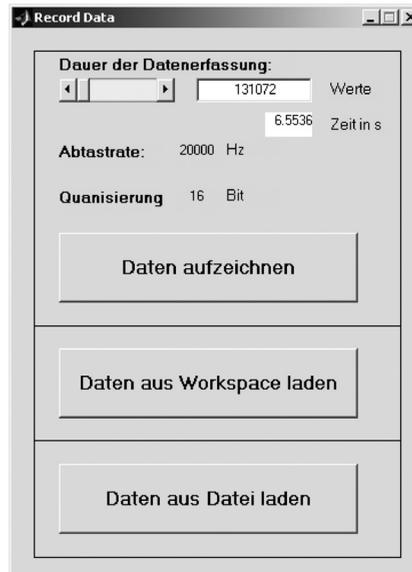


Abbildung 2.3. Dialog zur Datenerfassung.

2.8.2.2 Einstellungen zur Signalverarbeitung

Quantisierung, Abtastrate und Länge der aufgezeichneten Messdaten können direkt auf der Oberfläche angepasst werden. Eine erneute Datenerfassung ist nicht erforderlich.

- Als Quantisierungsstufen können Sie die Werte 1, 2, 4, 8, 16 vorgeben.
- Die Abtastfrequenz wird verändert, indem nur jeder n -te Abtastwert verwendet wird. Damit ergeben sich aus der ursprünglichen Abtastfrequenz $f_A = 20$ kHz als zulässige neue Abtastfrequenzen die Bruchteile $f_{A,\text{neu}} = \frac{20}{n}$ kHz.
- Für die Anzahl der Abtastwerte sind alle Zweierpotenzen zwischen $2^4 = 16$ und $2^{12} = 4096$ erlaubt.

Das Programm reagiert auf fehlerhafte Eingaben mit einer automatischen Korrektur der Werte. Dabei erfolgt keine Fehlermeldung.

2.8.2.3 Mathematische Funktionen

Auf der linken Seite können Sie auswählen, welche mathematischen Operationen durchgeführt werden sollen. Manche Funktionen lassen sich sowohl im Zeitbereich (Faltung) als auch im Frequenzbereich (FFT) berechnen. Links unten wird die Bandgeschwindigkeit des Förderbandes ausgegeben. Sie wird immer über die Kreuzkovarianz berechnet, auch wenn Sie andere Korrelationsfunktionen ausgewählt haben.

2.8.2.4 Schaubilder

Rechts werden die Ergebnisse der Berechnungen visualisiert. Die oberen Schaubilder zeigen die aufgenommenen Sensorsignale, das untere die Korrelationsfunktionen. Durch entsprechende Auswahl der Klickbuttons werden die gewünschten Korrelationsfunktionen berechnet und dargestellt.

2.9 Versuchsvorbereitende Aufgaben

Um ein Gefühl für die Berechnung von Korrelationsfunktionen und deren Eigenschaften zu bekommen, ist es notwendig, zunächst einige vorbereitende Rechnungen durchzuführen. Mit den damit erzielten Ergebnissen im Hinterkopf lassen sich die mit dem Versuchsaufbau gewonnenen Korrelationsfunktionen besser interpretieren.

Aufgabe 2.1: Korrelation stochastischer Signale

Ein mittelwertfreier weißer Gauß'scher Rauschprozess $x(t)$ wird mit einem idealen Tiefpassfilter der Bandbreite $B = \frac{1}{T}$ gefiltert.

- Wie lautet die Autokorrelationsfunktion des gefilterten Prozesses $y(t)$?
- Der gefilterte Prozess $y(t)$ wird jetzt mit $\tau = k \cdot t_A$ abgetastet. Wie ist t_A zu wählen, damit die Abtastwerte unkorreliert sind?

Aufgabe 2.2: Autokorrelation zusammengesetzter Funktionen

Gegeben sei ein mittelwertfreies, stationäres Zufallssignal $f(t)$, das von einem optischen Referenzsensor aufgenommen wurde und welches die Autokorrelationsfunktion $r_{ff}(\tau)$ besitzt. In der Produktion soll dieser Sensor durch einen preiswerteren ersetzt werden, von dem bekannt ist, dass er bzgl. der Referenz an der Stelle $f(0)$ einen Offset von B (ohne Einheit) aufweist. Darüber driftet er über dem Wertebereich von $f(t)$ linear. Diese Drift wurde aufgenommen und ihre Steigung zu A bestimmt.

- Stellen Sie den funktionalen Zusammenhang zwischen der Zeitfunktion des preiswerteren Sensors $g(t)$ und der des Referenzsensors $f(t)$ her und zeichnen Sie die Verhältnisse in einem Schaubild, dessen Abszisse von $f(t)$ gebildet wird.
- Bestimmen Sie die Autokorrelationsfunktion (AKF) $r_{gg}(\tau)$ des preiswerten Sensors in Abhängigkeit von der AKF $r_{ff}(\tau)$ des Referenzsensors.

Aufgabe 2.3: Laufzeitmessung

In Abb. 2.4 sind zwei Signale x_n und y_n einer Laufzeitmessung dargestellt, die mit dem Versuchsförderband aufgezeichnet wurden.

- Bestimmen Sie die Kreuzkorrelationsfolge $r_{xy}[k]$ von x_n und y_n .

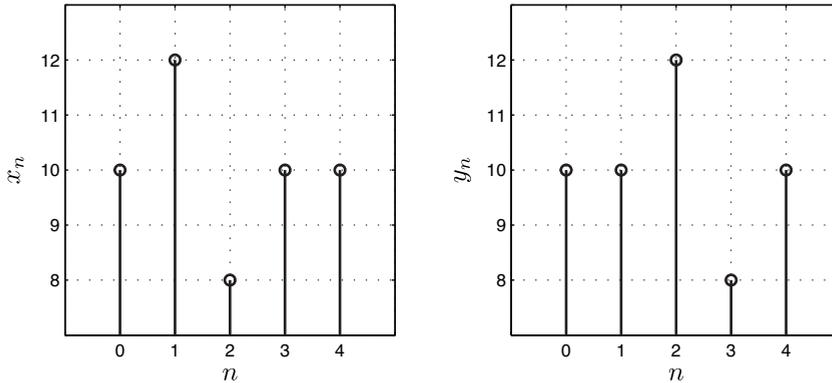


Abbildung 2.4. Signale x_n und y_n einer Laufzeitmessung.

- b) Lässt sich anhand des Ergebnisses aus Teilaufgabe (a) die Laufzeit auf einfache Weise ermitteln?
- Wenn ja: Anhand welchen Kriteriums wird die Laufzeit ermittelt?
 - Wenn nein: Welches Verfahren ist für die einfache Bestimmung der Laufzeit besser geeignet (Begründung)?
- c) Bei umfangreichen Datensätzen wird die Kreuzkorrelationsfolge häufig im Frequenzbereich berechnet.
- Wo liegen die Vorteile?
 - Wodurch wird das Ergebnis verfälscht? Wann liefert diese Vorgehensweise zuverlässige Ergebnisse (ohne Zero-Padding)?
 - Warum wird in diesem Fall Zero-Padding angewandt?
- d) Häufig kommt anstelle der Kreuzkorrelation die Polaritätskorrelation zum Einsatz.
- Berechnen sie die Polaritätskorrelation der Signale aus Abb. 2.4. Beachten Sie, dass x_n und y_n nicht mittelwertfrei sind. Wählen Sie daher für die 1-Bit-Quantisierung die Grenzen $x_n = 9$ und $y_n = 9$ als Schaltschwellen. Verwenden Sie die quantisierten Werte $\{-1, 1\}$.
 - Wie schätzen Sie den Rechenaufwand dieses Verfahrens im Vergleich zur Berechnung der Kreuzkorrelation $r_{xy}[k]$ ein?

2.10 Versuche an der Förderbandanlage

Nun soll die Geschwindigkeit eines Förderbandes entsprechend Abschn. 2.8 ermittelt werden. Für die Ermittlung der Korrelationsfunktionen steht Ihnen die *Matlab*-Oberfläche aus Abb. 2.2 zur Verfügung. Die Korrelationsfunktionen sind bereits implementiert. Um Ihnen dennoch einen Eindruck von der Berechnung der Korrelationsfunktionen unter *Matlab* zu geben, sollen Sie zunächst ein eigenes kleines Programm

schreiben, bevor Sie dann auf die bereits implementierten Funktionen zurückgreifen. Im weiteren Verlauf des Versuchs wird herausgearbeitet, welchen Einfluss gewisse Parameter wie Abtastzeit und Beobachtungsdauer auf die in der Praxis erzielten Ergebnisse haben.

Aufgabe 2.4: Berechnung der Kreuzkorrelation unter Matlab

In Aufgabe 2.3 haben Sie die Kreuzkorrelation zweier Signalfolgen x_n und y_n berechnet. Diese Berechnung soll nun mit Hilfe von *Matlab* erfolgen.

- Legen Sie im Verzeichnis `D:\PDSV\Gruppen` ein neues Verzeichnis für Ihre Gruppe an. Legen Sie alle Dateien, die Sie im Laufe des Versuchs erstellen, nur in diesem Ordner ab!
 - Starten Sie *Matlab* und wechseln Sie in das Verzeichnis Ihrer Gruppe.
- a) Geben Sie am *Matlab*-Prompt `edit` ein. Ein leeres `m`-File wird geöffnet. Speichern Sie es unter dem Namen `Kreuzkorrelation.m` ab. Zur Programmierung der Korrelationsfunktion gehen Sie folgendermaßen vor:
1. Erstellen Sie zwei Vektoren x und y für die Signalfolgen aus Aufgabe 2.3.
 2. Ermitteln Sie den Wertebereich von k . Legen Sie einen Ergebnisvektor `Rxy` in seiner zu erwartenden Dimension an.
 3. Berechnen Sie die Kreuzkorrelation $r_{xy}[k]$ zunächst für die nicht negativen Werte von k . Weisen Sie das Ergebnis dem Vektor `Rxy` an der entsprechenden Stelle zu. Beachten Sie dabei, dass ein *Matlab*-Vektor nur mit positiven Indizes beschrieben werden kann.
 4. Wiederholen Sie den letzten Schritt für negative Werte von k .
 5. Erstellen Sie einen Indexvektor, der bei der graphischen Ausgabe die richtige Skalierung der Ordinate ermöglicht.
 6. Geben Sie das Ergebnis der Berechnung über den `stem`-Befehl in einer `figure` aus.
- b) Für die Berechnung der Korrelationsfunktion bzw. der Kovarianzfunktion stehen unter *Matlab* auch die Befehle `xcorr` bzw. `xcov` zur Verfügung. Über die Eingabe `help xcorr/xcov` am *Matlab*-Prompt erhalten Sie nähere Informationen über die Anwendung der Befehle.
- Erweitern Sie Ihr `m`-File durch eine Schätzung der Kreuzkorrelationsfolge $\hat{r}_{xy}[k]$ und der Kreuzkovarianzfolge $\hat{C}_{xy}[k]$ anhand der Befehle `xcorr` und `xcov`.
 - Welche Optionen müssen `xcorr` übergeben werden, um das gleiche Ergebnis zu erhalten wie bei Ihrer Berechnung? Übergeben Sie `xcov` dieselben Optionen.
 - Geben Sie das Ergebnis bei Berechnung über `xcorr` und über `xcov` jeweils in einer zusätzlichen `figure` aus.
 - Aus welchem der beiden Graphen kann die Zeitverschiebung k der beiden Signale x_n und y_n direkt durch Ablesen ermittelt werden? Bestimmen Sie k .

Aufgabe 2.5: Vorbereitende Messungen

- a) Messen Sie mit einem Lineal den Sensorabstand d und überprüfen Sie, ob die Sensorhöhe h (Abstand Förderband–Gehäuseunterseite) für beide Sensoren gleich ist.
 - b) Schalten Sie das Förderband ein. Messen Sie die ungefähre Förderbandgeschwindigkeit anhand der Laufzeit τ einer Marke über die Förderbandlänge $l = 121,5$ cm.
-

Aufgabe 2.6: Autokorrelation und Autokovarianz

Nun wird das Förderband für Autokorrelationsmessungen eingesetzt.

- Geben Sie eine ausreichende Menge Kies in den Vorratsbehälter des Förderbandes.
 - Schalten Sie die Spannungsversorgung für die Sensoren und Lichtquellen an. Es müssen die roten LEDs am Verstärkergehäuse leuchten.
 - Geben Sie am *Matlab*-Prompt den Befehl `Korrelation` ein. Es öffnet sich eine *Matlab*-GUI – wie in Abschn. 2.8.2 beschrieben.
 - Durch Auswahl der Schaltfläche `Daten` öffnet sich das *Matlab*-GUI für die Datenerfassung (siehe Abschn. 2.8.2.1).
 - Starten Sie das Förderband bei beliebiger Geschwindigkeit und beginnen Sie mit der Messung, sobald der Kies den zweiten Sensor erreicht hat.
 - Nachdem die Messung beendet ist, nehmen Sie im Hauptfenster die folgenden Einstellungen vor:
 - Abtastfrequenz: $f_A = 500$ Hz,
 - Anzahl der Messwerte: $N = 2048$,
 - Quantisierung: 16 bit.
 - Lassen Sie sich die AKF der beiden Sensorsignale ($Ka\ 1$ und $Ka\ 2$) jeweils im Zeit- und im Frequenzbereich berechnen.
 - Ändern Sie die Abtastfrequenz auf $f_A = 1000$ Hz und wiederholen Sie den letzten Schritt.
- a) Warum stimmen die Messsignale der beiden Sensoren nicht vollkommen überein?
 - b) Vergleichen Sie die Autokorrelationsfunktionen der Signale für die Abtastfrequenzen $f_A = 500$ Hz und $f_A = 1000$ Hz. Welchen Einfluss hat die Abtastfrequenz auf die Autokorrelationsfunktionen?
-

Aufgabe 2.7: Kreuzkorrelation und Kreuzkovarianz

Die Signale der beiden Sensoren sollen nun kreuzkorreliert werden. Stellen Sie eine Quantisierung von 16 bit und eine Werteanzahl von $N = 2048$ ein.

- a) Welche Abtastfrequenz f_A aus der vorangegangenen Aufgabe erscheint Ihnen aufgrund der Bandgeschwindigkeit am günstigsten? Begründen Sie! Verwenden Sie diese Einstellung im Folgenden.

- b) Warum eignet sich im Zeitbereich die KKV zur Berechnung der Bandgeschwindigkeit besser als die KKF?
- c) Inwiefern stellt eine erwartungstreue Schätzung der KKF eine Verbesserung oder Verschlechterung gegenüber der nicht erwartungstreuen KKF und der KKV dar? Vergleichen sie dazu im GUI die entsprechenden Korrelationsgraphen.

Nun wird eine Geschwindigkeitsmessung durch Auswertung der Kreuzkovarianzfolge im Zeitbereich durchgeführt. Aus dem Sensorabstand d sowie dem Maximum der KKV ($\hat{=}$ Signallaufzeit zwischen den Sensoren) berechnet das *Matlab*-Programm die Förderbandgeschwindigkeit.

- d) Wie hoch ist die ermittelte Förderbandgeschwindigkeit? Stimmt sie mit dem erwarteten Wert überein?
- e) Welchen Einfluss hat die Justage der Sensoren auf die Berechnung der KKV und damit mittelbar auf die Ermittlung der Förderbandgeschwindigkeit? Verändern Sie dazu schrittweise die Ausrichtung der Sensorbrücke.

Aufgabe 2.8: Konsistenz der Korrelationsschätzung

Stellen Sie die Förderbandgeschwindigkeit mit Hilfe des Rädchens zur Geschwindigkeitseinstellung auf minimal ein und führen Sie anschließend eine Datenerfassung durch. Im Folgenden soll die Quantisierung 16 bit, die Wertezahl $N = 2048$ und die Abtastfrequenz $f_A = 10\,000$ Hz betragen.

- a) Ermitteln Sie mit der Stoppuhr die tatsächliche Geschwindigkeit v des Förderbandes sowie die Zeitverschiebung τ der Sensorsignale. Verwenden Sie dazu die Werte aus Aufgabe 2.5.
- b) Lassen Sie sich nun anhand der KKV die Zeitverschiebung τ und die Förderbandgeschwindigkeit v berechnen.
 - Vergleichen Sie das Ergebnis mit den Werten, die Sie mit der Stoppuhr ermittelt haben. Was stellen Sie fest? Erklären Sie Ihre Beobachtung.
Hinweis: Vergleichen Sie die Beobachtungsdauer der Messung mit der von Ihnen berechneten Laufzeit τ . Kann die Korrelation für diese Konstellation sinnvolle Ergebnisse liefern?
 - Welche Beobachtungsdauer T_B ist mindestens erforderlich, um die Laufzeit τ korrekt zu ermitteln? Wie muss dann die Abtastfrequenz f_A gewählt werden?
- c) Verändern Sie nun die Abtastfrequenz auf $f_A = 4000$ Hz.
 - Ist die Bedingung aus dem vorangegangenen Aufgabenteil erfüllt?
 - Lassen Sie sich wieder die Zeitverschiebung und die Förderbandgeschwindigkeit über die KKV berechnen. Stimmen die Ergebnisse jetzt mit Ihren Erwartungen überein? Warum reicht die Minimalbedingung aus Aufgabenteil b) im Normalfall nicht für eine korrekte Laufzeitbestimmung über die KKV aus?

- Wie muss die Bedingung aus dem letzten Aufgabenteil modifiziert werden, um die Konsistenz der Korrelationsschätzung gemäß (2.39) zu gewährleisten?

d) Ermitteln Sie mit Hilfe der Konsistenzbedingung aus dem letzten Aufgabenteil die maximal zulässige Abtastfrequenz $f_{A,\max}$. Stimmen nun die Stoppuhr-Ergebnisse für τ und v mit den zugehörigen KKV-Ergebnissen überein?

In den folgenden Teilaufgaben wird nun bei einer festen Werteanzahl $N = 2048$ und bei fester Abtastfrequenz $f_A = 4000$ Hz die Geschwindigkeit des Förderbandes variiert. Dabei soll die tatsächliche Grenze für eine konsistente Schätzung der Laufzeit ermittelt und mit der theoretischen Grenze verglichen werden.

e) Berechnen Sie mit Hilfe der Konsistenzbedingung nach (2.39) zunächst die *theoretisch* zulässige Minimalgeschwindigkeit $v_{\min,\text{theor}}$ bei der die Konsistenz der Schätzung noch gewährleistet ist.

f) Ermitteln Sie nun bei gleichbleibender Wertezahl N und gleicher Abtastfrequenz f_A die *reale* zulässige Minimalgeschwindigkeit $v_{\min,\text{real}}$ bei der die Berechnung anhand der KKV noch korrekt ist. Erhöhen Sie dazu schrittweise die Geschwindigkeit des Förderbandes und lassen Sie sich jeweils die Laufzeit und Förderbandgeschwindigkeit anhand der KKV berechnen. Als Referenz für die jeweilige Geschwindigkeitsmessung können Sie das Ergebnis verwenden, das sich für $N = 2048$ und $f_A = 500$ Hz ergibt (bei dieser Konstellation ist die Konsistenz für alle einstellbaren Bandgeschwindigkeiten immer gewährleistet). Ab welcher Geschwindigkeit $v_{\min,\text{real}}$ liefert die KKV das richtige Ergebnis?

Aufgabe 2.9: Ungenauigkeiten bei der Korrelationsschätzung

Stellen Sie die maximale Förderbandgeschwindigkeit ein und führen Sie eine Messung durch. Wählen Sie zunächst die Werteanzahl $N = 4096$ und die Abtastfrequenz $f_A = 1000$ Hz und lassen Sie sich die Geschwindigkeit berechnen. Halbieren Sie anschließend die Anzahl der Messwerte schrittweise von $N = 4096$ bis auf $N = 64$. Passen Sie dabei die Abtastfrequenz so an, dass die Beobachtungsdauer konstant bleibt.

Wie verändert sich die Genauigkeit der Messwerte mit der Abtastfrequenz? Betrachten Sie dazu die Position des Maximums der KKV. Können Sie sich Ihre Beobachtung erklären?

Aufgabe 2.10: Zero-Padding

a) Die Berechnung der KKF anhand eines beschränkten Datensatzes über die FFT (kurz: FFT-KKF) ist prinzipiell fehlerhaft. Für welche Argumente der FFT-KKF kann man von einem sehr kleinen Fehler ausgehen?

b) Lassen Sie sich bei identischer Abtastfrequenz $f_A = 500$ Hz und Geschwindigkeit des Förderbandes eine 2048-Punkte-KKF, eine 2048-Punkte-FFT-KKF ohne Zero-

Padding sowie eine 2048-Punkte-FFT-KKF mit Zero-Padding berechnen. Vergleichen Sie die Ergebnisse.

- c) Wiederholen Sie die vorherige Aufgabe, ersetzen Sie aber die KKF durch die KKV. Was stellen Sie fest?
- d) Wie viele Multiplikationen sind bei der direkten Berechnung der KKV bzw. KKF im Gegensatz zur Berechnung der FFT-KKV bzw. FFT-KKF mit $N = 2048$ notwendig? Welche Schlussfolgerung kann daraus gezogen werden?

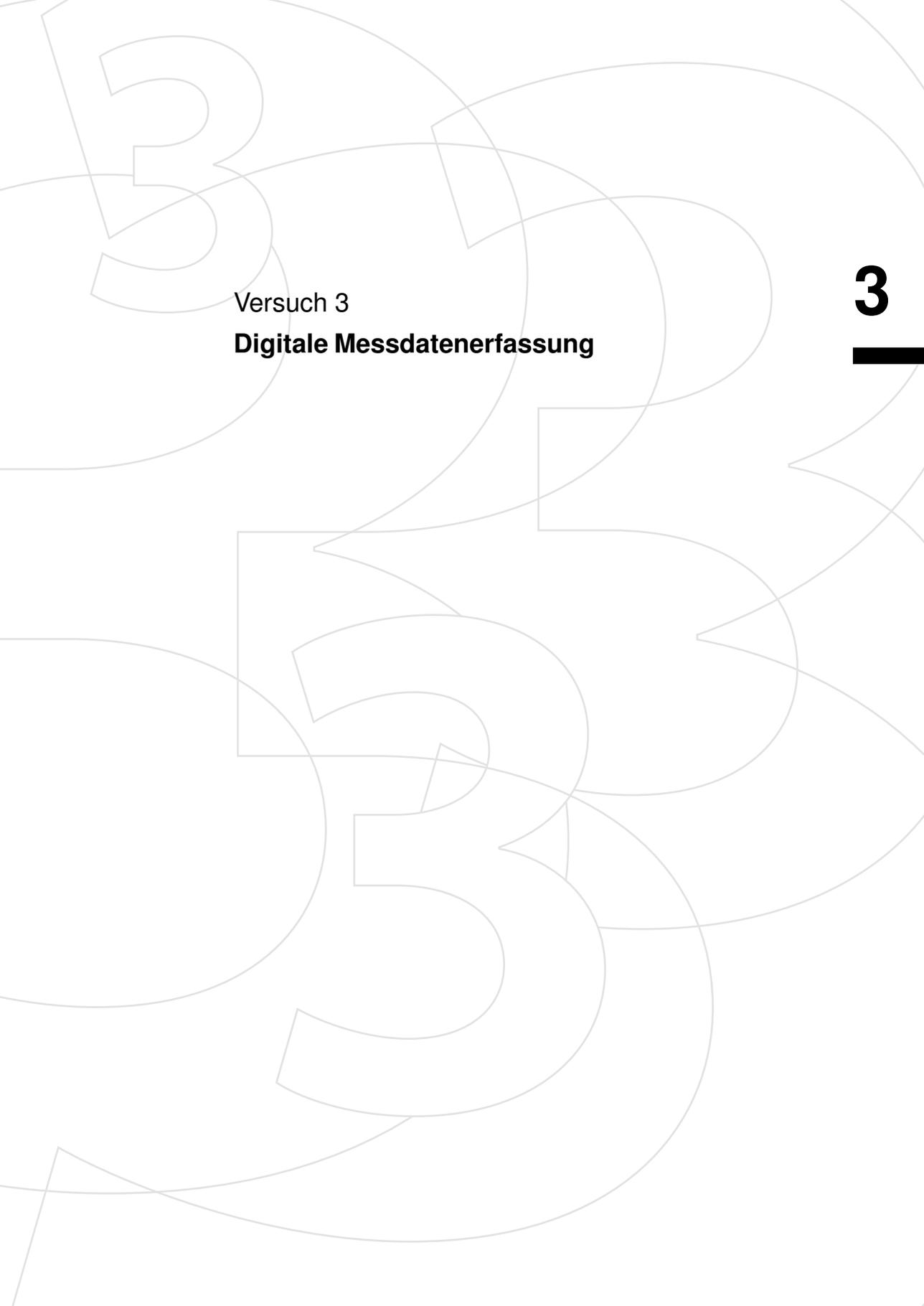
Aufgabe 2.11: Polaritätskorrelation

Stellen Sie die Abtastfrequenz $f_A = 1$ kHz und die Werteanzahl $N = 1024$ ein. Variieren Sie die Quantisierungsstufen von 8 bit bis 1 bit und vergleichen Sie die Ergebnisse für die KKV und Bandgeschwindigkeit. Plotten Sie Ihre Ergebnisse für zwei repräsentative Quantisierungen!

- a) Wie leistungsfähig ist der Polaritätskorrelator hinsichtlich seiner Genauigkeit bei niedrigen Quantisierungsstufen? Bewerten Sie den Rechenaufwand der Polaritätskorrelation, insbesondere bei der 1-Bit-Quantisierung.
- b) Überprüfen Sie, ob die Konsistenzbedingung $\tau < 0,2 \cdot \frac{N}{f_A}$ auch für die Polaritätskorrelation mit 1-Bit-Quantisierung gilt.

2.11 Literatur

- [KK12] KARL-DIRK KAMMEYER und KRISTIAN KROSCHER: *Digitale Signalverarbeitung – Filterung und Spektralanalyse mit MATLAB-Übungen*. Springer Vieweg, Wiesbaden, 8. Auflage, 2012.
- [PJ15] FERNANDO PUENTE LEÓN und HOLGER JÄKEL: *Signale und Systeme*. De Gruyter Oldenbourg, Berlin, 6. Auflage, 2015.
- [PL15] FERNANDO PUENTE LEÓN: *Messtechnik – Systemtheorie für Ingenieure und Informatiker*. Springer, Berlin Heidelberg, 10. Auflage, 2015.



Versuch 3

Digitale Messdatenerfassung

3

3	Digitale Messdatenerfassung	
3.1	Einleitung	69
3.2	Zeitdiskretisierung	69
3.2.1	Abtastung	69
3.2.2	Abtasttheorem	70
3.2.3	Anti-Aliasing-Filter	70
3.2.3.1	Butterworth-Filter	71
3.2.3.2	Tschebyscheff-Filter	71
3.2.3.3	Elliptisches Filter	71
3.2.3.4	Bessel-Filter	71
3.2.4	Mittelwertbildung bei endlicher Abtastdauer	71
3.2.5	Zeitliche Abtastfehler (Jitter)	73
3.2.6	Quantisierung analoger Signale	73
3.2.6.1	Wahrscheinlichkeitsdichte der Signalwerte	73
3.2.6.2	Quantisierungstheorem	74
3.2.6.3	Lineares Quantisierungsmodell	75
3.2.6.4	Dithering	76
3.3	Diskrete Signale im Frequenzbereich	77
3.3.1	Die Fourier-Transformation kontinuierlicher Signale	77
3.3.2	Fourier-Transformation zeitdiskreter Signale	78
3.3.3	Diskrete Fourier-Transformation (DFT)	79
3.3.3.1	Die schnelle Fourier-Transformation (FFT)	80
3.3.3.2	Leistungsdichtespektrum und Periodogramm	80
3.3.3.3	Leckeffekt	80
3.3.3.4	Fensterfunktionen	81
3.4	Verwendete Hardware und ihre Ansteuerung	83
3.4.1	Signalerfassung	84
3.4.2	Signalauswertung und Signalpräsentation	85
3.5	Versuchsvorbereitende Aufgaben	86
3.6	Versuchsaufgaben	87
3.7	Literatur	95

3 Digitale Messdatenerfassung

3.1 Einleitung

In der modernen Messtechnik werden Signale meist rechnergestützt ausgewertet. Dazu müssen die analogen Signale in eine für den Rechner geeignete Form gebracht werden. Dieser Prozess wird **Digitalisierung** genannt und umfasst sowohl eine Abtastung als auch eine Quantisierung (Amplitudendiskretisierung). Dabei werden die Signale implizit einer Signalverarbeitung unterzogen, der sich der Ingenieur oft nicht bewusst ist. Aus diesem Grund sollen die wesentlichen Einflüsse verdeutlicht werden. Der Ingenieur soll sich dessen bewusst sein, dass auch die digitalisierten Daten fehlerbehaftet sein können. Durch dieses Bewusstsein und sorgfältiges Design des Datenerfassungssystems soll vermieden werden, dass in der Praxis innerhalb der digitalen Verarbeitungskette Fehler auftreten, die aus ungeeignet gewählter Digitalisierung resultieren.

Ziel dieses Versuches ist es, den Konvertierungsvorgang von analogen Signalen in digitale Signale sowie die dabei auftretenden Effekte – Bandüberlappungsfehler (engl. *aliasing*), Leckeffekt, Quantisierungsrauschen und Jitter – zu untersuchen. Die Datenerfassung erfolgt mit dem Datenerfassungsmodul *USB-6008* von *National Instruments (NI)*. Für die Auswertung der digitalisierten Signale und die Untersuchung der Effekte, die mit der Abtastung verbunden sind, wird das Programm *LabVIEW* eingesetzt.

3.2 Zeitdiskretisierung

3.2.1 Abtastung

Bei der idealen Abtastung wird ein zeitkontinuierliches Signal $x(t)$ zu äquidistanten Zeitpunkten $n t_A$ erfasst und in das zeitdiskrete Signal $x_*(t)$ überführt:

$$x_*(t) = x(t) \cdot \sum_{n=-\infty}^{\infty} \delta(t - n t_A). \quad (3.1)$$

Die Fourier-Transformierte $X_*(f)$ des abgetasteten Signals $x_*(t)$ wird durch eine Faltung im Frequenzbereich beschrieben:

$$X_*(f) = X(f) * \frac{1}{t_A} \cdot \sum_{k=-\infty}^{\infty} \delta\left(f - \frac{k}{t_A}\right) = \frac{1}{t_A} \cdot \sum_{k=-\infty}^{\infty} X\left(f - \frac{k}{t_A}\right). \quad (3.2)$$

Das Spektrum $X(f)$ wird somit im Abstand $f_A = \frac{1}{t_A}$ periodisch wiederholt [PJ15].

3.2.2 Abtasttheorem

Ist das Spektrum $X(f)$ eines kontinuierlichen Signals bandbegrenzt auf die halbe Abtastfrequenz $\frac{1}{2}f_A$, so kann das ursprüngliche Signal $x(t)$ aus dem zeitdiskreten Signal $x_*(t)$ fehlerfrei rekonstruiert werden. Ist diese Bedingung nicht erfüllt, so treten im Spektrum des abgetasteten Signals $X_*(f)$ aufgrund der periodischen Wiederholungen Überlappungen (**Aliasing**) auf (siehe Abb. 3.1).

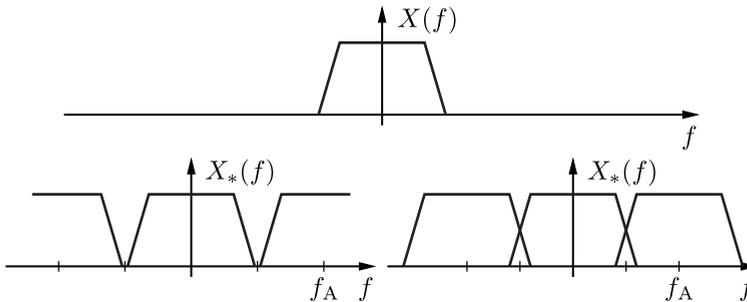


Abbildung 3.1. Spektrale Überlappung bei bandbegrenzten Signalen.

3.2.3 Anti-Aliasing-Filter

Ist das Spektrum des abzutastenden Signals nicht bandbegrenzt, müssen zur Erfüllung des Abtasttheorems vor der Abtastung alle Frequenzanteile, die größer-gleich der halben Abtastfrequenz sind, mit Hilfe eines Tiefpassfilters (sog. **Anti-Aliasing-Filter**) herausgefiltert werden. An ein Anti-Aliasing-Filter werden in der Praxis sehr hohe Anforderungen gestellt. Im Bereich bis zur halben Abtastfrequenz $f_A/2$ (Nyquist-Band) soll das Spektrum möglichst nicht gestört werden. Ab $f_A/2$ sollen alle Spektralanteile vollständig verschwinden. Solch ein idealer Tiefpass ist jedoch nicht zu verwirklichen (Abb. 3.2). In der Praxis muss man sich deshalb mit steilflankigen Filtern hoher Ordnung begnügen [PL15].

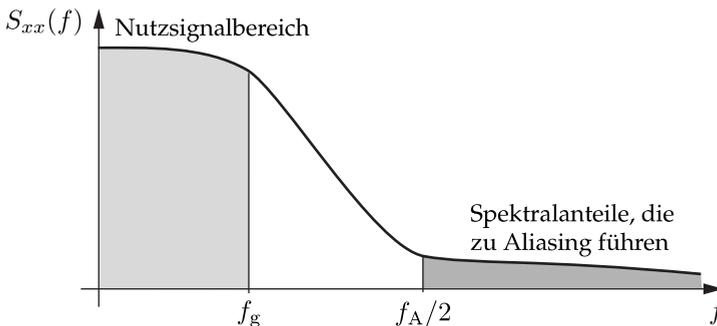


Abbildung 3.2. Aliasing-Fehler durch nicht ideale Filter.

Im Folgenden werden einige gängige Tiefpassfilter kurz vorgestellt. Da die Filtertheorie ausführlich im Versuch 4 „Analoge und digitale Filterung“ behandelt wird, wird hier lediglich auf die wesentlichen Unterschiede eingegangen.

3.2.3.1 Butterworth-Filter

Das **Butterworth-Filter** zeichnet sich durch einen besonders flachen Verlauf sowohl im Durchlass- als auch im Sperrbereich aus. Ein wesentlicher Nachteil des Butterworth-Filters ist der breite Übergang zwischen Durchlass- und Sperrbereich.

3.2.3.2 Tschebyscheff-Filter

Das **Tschebyscheff-Filter** weist bei gleicher Filterordnung einen deutlich steileren Übergang zwischen Durchlass- und Sperrbereich auf als das Butterworth-Filter. Dem steht der Nachteil gegenüber, dass der Verlauf des Filters nun eine Welligkeit aufweist:

- Beim Tschebyscheff-Filter Typ I liegt die Welligkeit gleichmäßig im Durchlassbereich, während der Amplitudengang im Sperrbereich monoton fällt.
- Beim Tschebyscheff-Filter Typ II (auch inverses Tschebyscheff-Filter genannt) liegt die Welligkeit im Sperrbereich und der Durchlassbereich ist flach.

3.2.3.3 Elliptisches Filter

Das **elliptische Filter** oder **Cauer-Filter** kombiniert die beiden Tschebyscheff-Filter. Es weist sowohl im Durchlass- als auch im Sperrbereich eine Welligkeit auf. Der entscheidende Vorteil dieses Filters ist seine große Flankensteilheit. Es benötigt im Vergleich zum Butterworth- und den Tschebyscheff-Filtern eine deutlich niedrigere Ordnung, um ein vorgegebenes Toleranzschema einzuhalten. Die steile Flanke des elliptischen Filters erkaufte man sich allerdings durch eine extreme Phasenverzerrung.

3.2.3.4 Bessel-Filter

Das **Bessel-Filter** weist wie das Butterworth-Filter einen sehr flachen Amplitudenverlauf sowohl im Durchlass- als auch im Sperrbereich auf. Zusätzlich ist der Phasenverlauf im Durchlassbereich linear. Im Vergleich zu Tschebyscheff- und elliptischem Filter ist der Übergang zwischen Durchlass- und Sperrbereich aber deutlich breiter.

3.2.4 Mittelwertbildung bei endlicher Abtastdauer

In der Praxis wird für die Abtastung eine gewisse Zeitspanne benötigt. Wird beispielsweise ein Kondensator eingesetzt, der über das kontinuierliche Eingangssignal $x(t)$ aufgeladen wird und den Spannungswert für den Abtastvorgang kurz speichert, ist das implizit mit einer Integration der Eingangsspannung über die Ladezeit des Kondensators verbunden [PL15].

Diese Integration entspricht einer Mittelwertbildung über ein endliches Zeitintervall T . Wird diese Mittelung zu jedem Zeitpunkt t durchgeführt, ergibt sich der gleiche Mittelwert

$$\bar{x}(t) = \frac{1}{T} \int_{t-\frac{T}{2}}^{t+\frac{T}{2}} x(\tau) d\tau. \quad (3.3)$$

Diese Mittelwertbildung entspricht einer Faltung des kontinuierlichen Signals $x(t)$

$$\bar{x}(t) = x(t) * g(t) = \int_{-\infty}^{\infty} x(\tau) g(t - \tau) d\tau \quad (3.4)$$

mit einem **Rechteckfenster**

$$g(t) = \frac{1}{T} \cdot r_T(t) \quad \text{mit} \quad r_T(t) = \begin{cases} 1 & \text{für } |t| \leq \frac{T}{2} \\ 0 & \text{sonst} \end{cases}. \quad (3.5)$$

Im Frequenzbereich korrespondiert die Faltung mit einer Multiplikation des Spektrums $X(f)$ mit einer Sinc-Funktion:

$$\bar{X}(f) = X(f) \cdot G(f) = X(f) \cdot \frac{\sin(\pi f T)}{\pi f T} = X(f) \cdot \text{sinc}(f T). \quad (3.6)$$

Das Eingangssignal wird also bereits vor der sich anschließenden idealen δ -Abtastung verfälscht. Die Filtercharakteristik der Mittelwertbildung ist in Abb. 3.3 dargestellt.

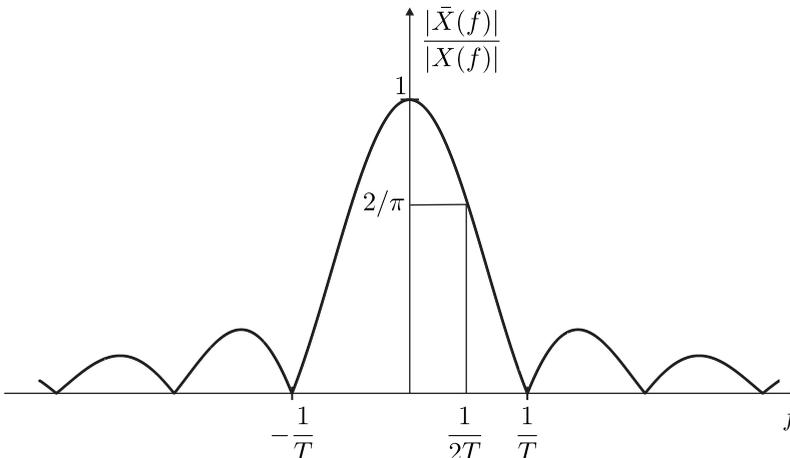


Abbildung 3.3. Filter-Charakteristik der Mittelwertbildung.

3.2.5 Zeitliche Abtastfehler (Jitter)

Bisher wurde davon ausgegangen, dass die Abtastung ohne zeitliche Fehler exakt zu den Zeitpunkten $n \cdot t_A$ erfolgt. In realen Systemen kann es aber zu einem zeitlichen Fehler bei der Abtastung kommen, der als **Jitter** bezeichnet wird. Dieser äußert sich in einer Signalverfälschung, die als additives Störsignal interpretiert werden kann.

Das Signal-Rausch-Verhältnis (SNR) für Jitter berechnet sich wie folgt [PL15]:

$$\text{SNR} = \frac{P_{\text{Sig}}}{P_{\text{Str}}} \approx \frac{2A_{\text{max}}^2 f_g^2}{4\pi^2 f_g^4 A_{\text{max}}^2 \sigma_\tau^2} = \frac{2}{(2\pi f_g \sigma_\tau)^2}. \quad (3.7)$$

Dabei bezeichnen σ_τ^2 die Varianz der Abtastzeit und f_g die maximale Signalfrequenz.

3.2.6 Quantisierung analoger Signale

Im Folgenden werden zunächst die grundlegenden mathematischen Zusammenhänge des Quantisierungstheorems erläutert. Darauf folgt eine qualitative Beschreibung der Wirkungsweise eines Dithersignals.

Bei der Quantisierung werden der kontinuierlichen Amplitude eines stetigen Eingangssignals $x(t)$ diskrete Amplitudenwerte $x_q(t)$ zugeordnet. Diese Zuordnung ist eine nichtlineare Operation. Befindet sich die Amplitude des Zeitsignals am Eingang eines Quantisierers innerhalb eines Quantisierungstreifens der Breite q , so nimmt das quantisierte Signal am Ausgang des Quantisierers den Amplitudenwert an, der der Mitte dieses Streifens entspricht. Dabei wird der Quantisierungsfehler

$$|e_q(t)| = |x_q(t) - x(t)| \leq \frac{q}{2} \quad (3.8)$$

in Kauf genommen. Aufgrund der Nichtlinearität der Quantisierungsfunktion weist das Fehlersignal $e_q(t)$ viele Sprünge und daher ein wesentlich breiteres Spektrum als das Eingangssignal $x(t)$ auf.

3.2.6.1 Wahrscheinlichkeitsdichte der Signalwerte

Grundlage für die weiterführenden Betrachtungen ist die Wahrscheinlichkeitsdichte $f_x(x)$ der Werte des kontinuierlichen Zeitsignals $x(t)$, die im Folgenden kurz als **Amplitudendichte** bezeichnet wird. Die Berechnung der Amplitudendichte lässt sich auf eine Transformation von Zufallsvariablen zurückführen [PL15, Abschn. 7.2.1]. An dieser Stelle interessiert jedoch vielmehr die qualitative Eigenschaft der Amplitudendichte $f_x(x)$: Sie gibt die **relative Häufigkeit** der in einem gegebenen Beobachtungsintervall auftretenden **Signalwerte** an.

3.2.6.2 Quantisierungstheorem

Die Zuordnung der kontinuierlichen Amplitudenwerte x zu den diskreten Amplitudenstufen

$$x_q = n \cdot q \quad \text{mit} \quad n \in \{-2^{N-1}, \dots, 2^{N-1} - 1\} \quad (3.9)$$

ist prinzipiell vergleichbar mit der Diskretisierung von Zeitsignalen. Dabei bezeichnet N die Auflösung des Quantisierers in Bit. Anstatt einer Diskretisierung entlang der Zeitachse findet bei der Quantisierung jedoch eine Diskretisierung der Signalwerte statt. Es wird keine Zeitfunktion abgetastet, sondern die Amplitudendichte. Aus der kontinuierlichen Amplitudendichte $f_x(x)$ des Eingangssignals $x(t)$ wird infolge der Quantisierung eine Reihe gewichteter Dirac-Impulse (siehe Abb. 3.4). Die Gewichtung der Impulse quantifiziert dabei die Auftretenswahrscheinlichkeit der Amplitudenwerte in der jeweiligen Quantisierungsstufe $n \cdot q$.

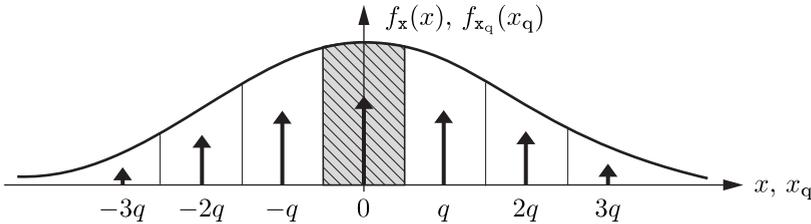


Abbildung 3.4. Wahrscheinlichkeitsdichten $f_x(x)$ des unquantisierten Signals und $f_{x_q}(x_q)$ des quantisierten Signals.

Damit ergibt sich für die Amplitudendichte des quantisierten Signals:

$$f_{x_q}(x_q) = P\left(x_q - \frac{q}{2} \leq x \leq x_q + \frac{q}{2}\right) \cdot \sum_{k=-\infty}^{\infty} \delta(x_q - k \cdot q) \quad (3.10)$$

$$= \int_{x_q - \frac{q}{2}}^{x_q + \frac{q}{2}} f_x(x) dx \cdot \sum_{k=-\infty}^{\infty} \delta(x_q - k \cdot q). \quad (3.11)$$

Das Integral in (3.11) entspricht einer Faltung der Amplitudendichte $f_x(x)$ mit einem Rechteckfenster $r_q(x)$:

$$\int_{x - \frac{q}{2}}^{x + \frac{q}{2}} f_x(u) du = f_x(x) * r_q(x), \quad r_q(x) = \begin{cases} 1 & \text{für } |x| \leq \frac{q}{2} \\ 0 & \text{sonst} \end{cases}. \quad (3.12)$$

Dies aber bedeutet für die charakteristische Funktion:

$$\Phi_{x_q}(f) = \mathcal{F}^{-1}\{f_{x_q}(x_q)\} = [\Phi_x(f) \cdot R_q(f)] * \frac{1}{q} \sum_{k=-\infty}^{\infty} \delta\left(f - \frac{k}{q}\right) \quad (3.13)$$

$$= \sum_{k=-\infty}^{\infty} \Phi_x\left(f - \frac{k}{q}\right) \cdot \frac{\sin\left(\pi q \left(f - \frac{k}{q}\right)\right)}{\pi q \left(f - \frac{k}{q}\right)}, \quad (3.14)$$

womit der Bezug zur Zeitdiskretisierung eines Signals hergestellt wäre. Auch hier hat eine Diskretisierung im Zeitbereich eine periodische Fortsetzung im Frequenzbereich zur Folge, bei der es zu Überlappungen (vgl. Aliasing) kommen kann.

Wenn die Amplitudendichte $f_x(x)$ eines kontinuierlichen Signals bandbegrenzt ist und für die charakteristische Funktion $\Phi_x(f)$ die Bedingung

$$\Phi_x(f) = 0 \quad \text{für} \quad |f| \geq \frac{1}{2q} \quad (3.15)$$

erfüllt ist, lässt sich aus der Amplitudendichte $f_{x_q}(x_q)$ des quantisierten Signals die ursprüngliche Amplitudendichte $f_x(x)$ des kontinuierlichen Signals fehlerfrei rekonstruieren. Die Bedingung (3.15) ist als **Quantisierungstheorem** bekannt und beantwortet die Frage, wie grob man Signale quantisieren darf, ohne wesentliche Signalinhalte zu verlieren.

Für die fehlerfreie Rekonstruktion der Momente der Amplitudendichte $f_x(x)$ genügt anstelle von (3.15) die schwächere Bedingung

$$\Phi_x(f) = 0 \quad \text{für} \quad |f| \geq \frac{1}{q}, \quad (3.16)$$

da diese für die Frequenz $f = 0$ berechnet werden [PL15].

In der Praxis ist das Quantisierungstheorem nie exakt erfüllbar. Es ist aber *näherungsweise* erfüllt, wenn die Amplitude des Eingangssignals innerhalb der Beobachtungszeit T_0 mehrere Quantisierungsstufen durchläuft.

3.2.6.3 Lineares Quantisierungsmodell

Bei Erfüllung der Quantisierungsbedingung (3.15) genügt die Betrachtung der charakteristischen Funktion im Nyquist-Band $-\frac{1}{2q} \leq f \leq \frac{1}{2q}$. Aus (3.13) ergibt sich mit $k = 0$:

$$\Phi_{x_q}(f) = \Phi_x(f) \cdot \frac{1}{q} R_q(f) = \Phi_x(f) \cdot \underbrace{\frac{\sin(\pi q f)}{(\pi q f)}}_{\Phi_{e_q}(f)}. \quad (3.17)$$

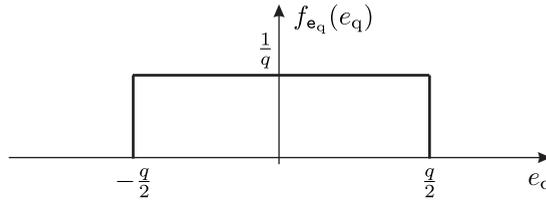


Abbildung 3.5. Wahrscheinlichkeitsdichte $f_{e_q}(e_q)$ des Quantisierungsfehlers.

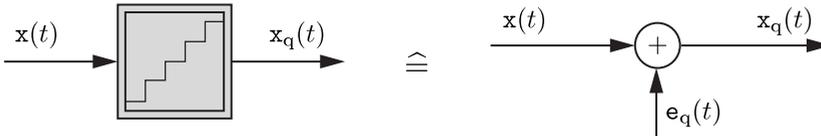


Abbildung 3.6. Lineares Quantisierungsmodell.

Es werden also zwei charakteristische Funktionen miteinander multipliziert. Dies entspricht einer Summation der zugehörigen Zufallsvariablen

$$x_q = x + e_q, \quad (3.18)$$

wobei der Quantisierungsfehler e_q eine Gleichverteilung innerhalb einer Quantisierungsstufe aufweist (siehe Abb. 3.5):

$$f_{e_q}(e_q) = \mathcal{F}\{\Phi_{e_q}(f)\} = \begin{cases} \frac{1}{q} & \text{für } |e_q| \leq \frac{q}{2} \\ 0 & \text{sonst.} \end{cases} \quad (3.19)$$

Aus (3.18) ergibt sich für die realen Signale

$$x_q(t) = x(t) + e_q(t) \quad (3.20)$$

und damit das lineare Signalmodell in Abb. 3.6 (rechts).

Das Signal-Rausch-Verhältnis infolge der Quantisierung errechnet sich für sinusförmige Signale unter Annahme des linearen Quantisierungsmodells gemäß

$$\text{SNR}|_{\text{dB}} = 10 \lg(1,5 \cdot 2^{2N}) = 6,02 \cdot N + 1,76, \quad (3.21)$$

wobei N der Auflösung des Quantisierers in Bit entspricht [PL15].

3.2.6.4 Dithering

Ist ein Signal innerhalb der Beobachtungszeit näherungsweise konstant,

$$x(t) \approx x_0 = \text{const} \quad \forall t \in [0, T_0], \quad (3.22)$$

so ergibt sich für seine Amplitudendichte ein Dirac-Impuls

$$f_x(x) = \delta(x - x_0) \quad (3.23)$$

bei dem entsprechenden Wert x_0 , was wegen

$$\Phi_x(f) = \mathcal{F}^{-1}\{\delta(x - x_0)\} = e^{j2\pi f x_0} \quad (3.24)$$

zu einer unendlich ausgedehnten charakteristischen Funktion führt. Infolge der Quantisierung kommt es zu Überlappungen der charakteristischen Funktion – das Quantisierungstheorem ist also nicht erfüllt.

Bei der Zeitdiskretisierung von Signalen vermeidet man Bandüberlappungen durch Anti-Aliasing-Filter. Bei der Quantisierung analoger Signale übernimmt die Funktion der Bandbegrenzung das sogenannte **Dithering**. Dazu wird dem Nutzsignal vor der eigentlichen Quantisierung ein **Dithersignal** $x_{\text{dith}}(t)$ mit folgenden Eigenschaften addiert:

- Die charakteristische Funktion der Amplitudendichte von $x_{\text{dith}}(t)$ muss bandbegrenzt sein.
- Vorzugsweise sollte $x_{\text{dith}}(t)$ hochfrequent und mittelwertfrei sein.

Das Dithersignal kann deterministisch oder stochastisch sein. Bei Verwendung eines periodischen Dithersignals muss eventuell bei der Zeitdiskretisierung die Abtastrate an die Wiederholungsfrequenz angepasst werden.

Die bandbegrenzende Wirkung des Dithersignals ergibt sich durch die Multiplikation der beiden charakteristischen Funktionen. Nach der Quantisierung wird das Dithersignal durch Mittelung bzw. Tiefpassfilterung unterdrückt. Abbildung 3.7 zeigt die Wirkungsweise des Ditherings.

3.3 Diskrete Signale im Frequenzbereich

Um auch den Frequenzgehalt diskreter Signale analysieren zu können, werden in diesem Abschnitt die dazu nötigen mathematischen Werkzeuge eingeführt. Außerdem wird verdeutlicht, worauf bei der Frequenzanalyse diskreter Signale geachtet werden muss, um unerwünschte Effekte wie zum Beispiel den Leckeffekt zu verhindern bzw. verringern.

3.3.1 Die Fourier-Transformation kontinuierlicher Signale

Periodische Signale lassen sich mittels der Fourier-Reihenentwicklung aus harmonischen Schwingungen der Grundfrequenz und ihren Oberschwingungen darstellen.

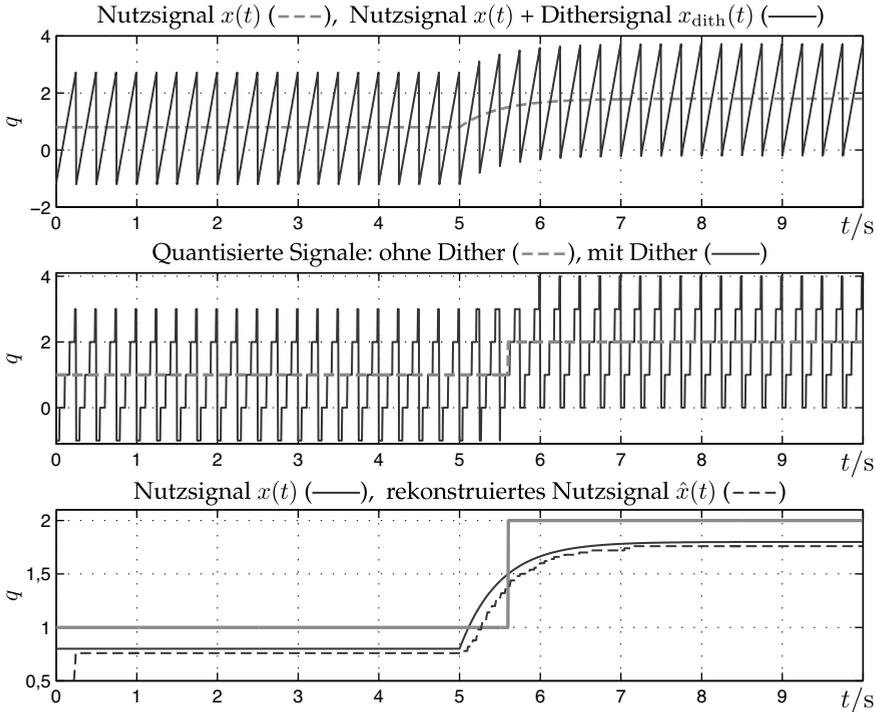


Abbildung 3.7. Nutzsinal mit und ohne Dithersignal (oben), quantisierte Signale (Mitte) und Ergebnis der Rekonstruktion durch Mittelwertbildung über eine Ditherperiode (unten).

Eine Erweiterung der Fourier-Reihe für aperiodische Signale ist die Fourier-Transformation:

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt. \quad (3.25)$$

Für die Rücktransformation der Fourier-Transformierten $X(f)$ aus dem Frequenzbereich in den Zeitbereich gilt:

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df. \quad (3.26)$$

3.3.2 Fourier-Transformation zeitdiskreter Signale

Nach (3.1) ergibt sich ein abgetastetes Signal $x_*(t)$ aus dem kontinuierlichen Signal $x(t)$ durch Multiplikation mit einer Impulsreihe. Die Fourier-Transformierte $X_*(f)$

des zeitdiskreten Signals ergibt sich damit zu:

$$X_*(f) = \int_{-\infty}^{\infty} x_*(t) e^{-j2\pi ft} dt \quad (3.27)$$

$$= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x(t) \delta(t - n t_A) e^{-j2\pi ft} dt \quad (3.28)$$

$$= \sum_{n=-\infty}^{\infty} x(n t_A) e^{-j2\pi f n t_A} . \quad (3.29)$$

3.3.3 Diskrete Fourier-Transformation (DFT)

Für eine rechnergestützte Spektralanalyse müssen neben den Signalen im Zeitbereich auch die Signale im Frequenzbereich diskretisiert werden. Zu diesem Zweck wird die **diskrete Fourier-Transformation** (DFT) eingeführt. Dabei sind zwei Punkte zu berücksichtigen:

- Jede Messung ist auf ein endliches Zeitintervall beschränkt. Daher stehen nur endlich viele Abtastwerte $x_n = x(n t_A)$ zur Verfügung.
- Neben der Zeit muss auch die Frequenz diskretisiert werden.

Werden im Rechner N Werte zu äquidistanten Zeitpunkten t_A erfasst, entspricht das einer **Beobachtungszeit** von

$$T_0 = N \cdot t_A \quad (3.30)$$

und einer **Beobachtungsfrequenz (Frequenzauflösung)** von

$$\Delta f = \frac{1}{T_0} = \frac{1}{N \cdot t_A} = \frac{f_A}{N} . \quad (3.31)$$

Da das Signal $x_*(t)$ diskret ist, ist das zugehörige Spektrum $X_*(f)$ periodisch. Daher genügt die Betrachtung einer Periode. Innerhalb dieser Periode besitzt das diskrete Spektrum endlich viele Spektrallinien, deren Abstand der Beobachtungsfrequenz Δf entspricht. Ferner induziert die Frequenzdiskretisierung eine periodische Fortsetzung des Zeitsignals $x_*(t)$.

Zur Berechnung der DFT an den diskreten Punkten $k \Delta f$ mit $0 \leq k < N$ wird die Summe (3.29) auf das Intervall $0 \leq n < N$ beschränkt:

$$X_k = X(k \Delta f) = \sum_{n=0}^{N-1} x_n e^{-j2\pi k \Delta f n t_A} = \sum_{n=0}^{N-1} x_n e^{-j2\pi \frac{k n}{N}} . \quad (3.32)$$

Die inverse diskrete Fourier-Transformation lautet:

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{j2\pi \frac{kn}{N}}. \quad (3.33)$$

3.3.3.1 Die schnelle Fourier-Transformation (FFT)

Für die Berechnung eines Wertes X_k der DFT werden N komplexe Multiplikationen und Additionen benötigt. Für N DFT-Koeffizienten ergeben sich damit insgesamt N^2 komplexe Multiplikationen und Additionen. Für große Datensätze ist die Berechnung der DFT daher sehr rechenintensiv.

Mit der **schnellen Fourier-Transformation** (engl. *fast Fourier transform*, FFT) existiert ein schneller Algorithmus, der die identische Rechnung der DFT durchführt, jedoch mit weniger Operationen. Der Rechenaufwand entspricht etwa $N \ln N$ komplexen Additionen und Multiplikationen [KK12].

3.3.3.2 Leistungsdichtespektrum und Periodogramm

Die DFT liefert eine komplexe Frequenzfunktion. Meist interessiert aber nur die Verteilung der Leistung des Signals auf die verschiedenen Frequenzen, während die Phaseninformation weniger wichtig erscheint. In diesen Fällen genügt es, das **Periodogramm** $|X_k|^2 = X_k X_k^*$ zu betrachten, welches eine Schätzung des **Leistungsdichtespektrums** ist. Oft wird das Periodogramm auf die Wertezahl N bezogen [KK12]:

$$S_k = \frac{1}{N} |X_k|^2. \quad (3.34)$$

3.3.3.3 Leckeffekt

Betrachtet wird eine komplexe harmonische Schwingung $x(t)$ der Frequenz f_s . Der **Leckeffekt** (engl. *leakage*) entsteht, wenn die Beobachtungszeit T_0 nicht einem ganzzahligen Vielfachen der Periodenlänge $1/f_s$ des Signals entspricht. Die Begrenzung der Beobachtungszeit lässt sich im Zeitbereich durch eine Multiplikation des Signals $x(t)$ mit einem Rechteckfenster $r_{T_0}(t)$ gemäß (3.5) beschreiben. Diese Multiplikation entspricht einer Faltung im Frequenzbereich. Die Fourier-Transformierte des Rechteckfensters $r_{T_0}(t)$ ist:

$$R_{T_0}(f) = \frac{T_0 \sin(\pi f T_0)}{\pi f T_0} = T_0 \operatorname{sinc}(f T_0), \quad (3.35)$$

d. h. infolge der Fensterung wird im Spektralbereich die Frequenz f_s nicht mehr durch einen Dirac-Impuls, sondern durch eine um die Frequenz f_s zentrierte Sinc-Funktion repräsentiert. Gilt nun für die Beobachtungszeit

$$T_0 = \frac{l}{f_s} \quad \text{mit} \quad l \in \mathbb{N}, \quad (3.36)$$

so tastet das DFT-Spektrum für $k = l$ die Sinc-Funktion im Maximum und für $k \neq l$ an den Nullstellen ab (Abb. 3.8 oben). Passt jedoch die Schwingfrequenz f_s nicht in das diskrete Raster der Beobachtungsfrequenz Δf ,

$$l = f_s T_0 = \frac{f_s}{\Delta f} \notin \mathbb{N}, \tag{3.37}$$

so werden Werte der Sinc-Funktion abgetastet, die das Spektrum verfälschen – dies entspricht dem **Leckeffekt** (Abb. 3.8 unten).

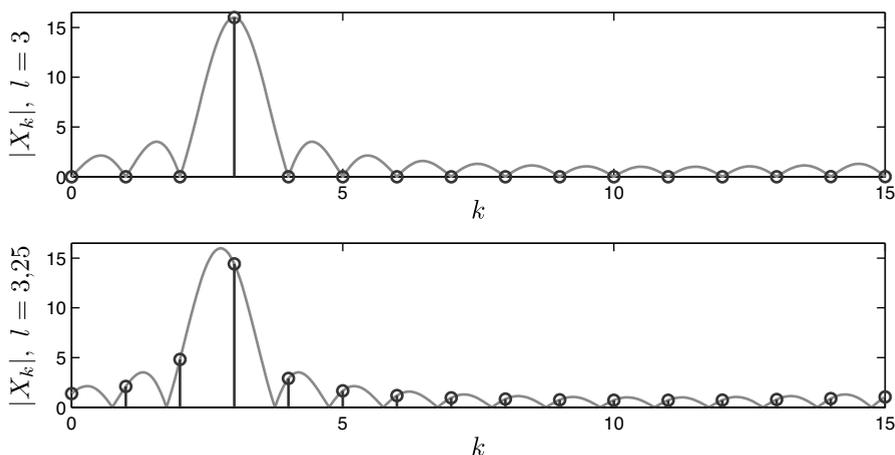


Abbildung 3.8. Verdeutlichung des Leckeffekts anhand der 16-Punkte-DFT einer Schwingung.

3.3.3.4 Fensterfunktionen

Entsprechend Abschn. 3.3.3.3 ist der Leckeffekt die Folge der Begrenzung einer Zeitfunktion durch eine Fensterfunktion. Für Anteile der Zeitfunktion $x(t)$, die nicht T_0 -periodisch sind, nähert die DFT nicht die Koeffizienten der Fourier-Reihe an, sondern die Fourier-Transformierte des Produkts von $x(t)$ mit einem rechteckigen Fenster $w(t)$ der Dauer T_0 . Die Verschmierung im DFT-Spektrum wird durch die Fourier-Transformierte des Fensters $w(t)$ festgelegt. Bei Verwendung von Fensterfunktionen mit schmalere Spektren lässt sich das Ausmaß des Leckeffekts verringern. Im Folgenden werden gebräuchliche Fenster vorgestellt.

Dreieckfenster

Das **Dreieckfenster** (auch **Bartlett-Fenster** genannt) ist wie folgt definiert:

$$w_n = \begin{cases} \frac{2n}{N} & \text{für } 0 \leq n \leq \frac{N}{2} \\ \frac{2(N-n)}{N} & \text{für } \frac{N}{2} < n \leq N \\ 0 & \text{sonst.} \end{cases} \tag{3.38}$$

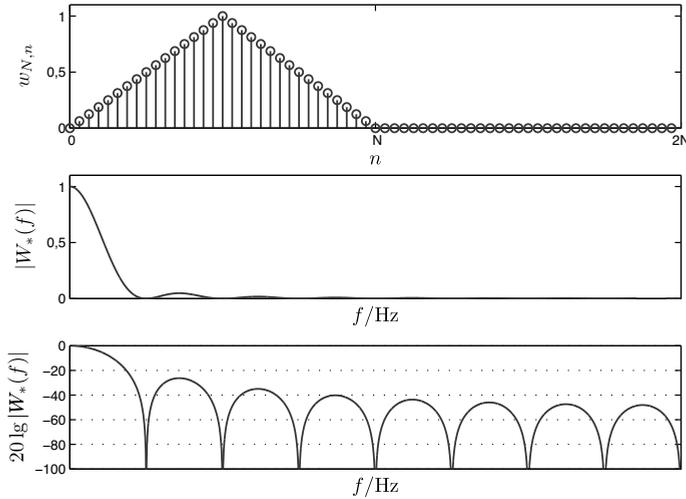


Abbildung 3.9. Dreieckfenster.

Seine Fourier-Transformierte berechnet sich zu:

$$W_*(f) = e^{-j\pi f t_A (N-1)} \cdot \left(\frac{\sin(2\pi f t_A \frac{N}{4})}{\sin(2\pi f t_A \frac{1}{2})} \right)^2. \quad (3.39)$$

Abbildung 3.9 zeigt die Fensterfunktion im Zeitbereich sowie den Betrag $|W_*(f)|$ der Fourier-Transformierten, wobei letzterer einmal linear und einmal logarithmisch in Dezibel aufgetragen wurde.

Hann-Fenster

Das **Hann-Fenster**

$$w_n = \begin{cases} \frac{1}{2} \left(1 - \cos\left(\frac{2\pi n}{N}\right) \right) & \text{für } 0 \leq n \leq N \\ 0 & \text{sonst} \end{cases} \quad (3.40)$$

besitzt die Fourier-Transformierte

$$W_*(f) = e^{-j\Omega \frac{N-1}{2}} \left(\frac{1}{2} \frac{\sin(\Omega \frac{N}{2})}{\sin(\Omega \frac{1}{2})} + \frac{1}{4} \frac{\sin(\Omega \frac{N}{2} - \frac{\pi N}{N-1})}{\sin(\Omega \frac{1}{2} - \frac{\pi}{N-1})} + \frac{1}{4} \frac{\sin(\Omega \frac{N}{2} + \frac{\pi N}{N-1})}{\sin(\Omega \frac{1}{2} + \frac{\pi}{N-1})} \right). \quad (3.41)$$

mit der normierten Kreisfrequenz $\Omega = 2\pi f t_A$. Es ist in Abb. 3.10 dargestellt.

Hamming-Fenster

Das **Hamming-Fenster** ist in Abb. 3.11 dargestellt:

$$w_n = \begin{cases} 0,54 - 0,46 \cdot \cos\left(\frac{2\pi n}{N}\right) & \text{für } 0 \leq n \leq N \\ 0 & \text{sonst} \end{cases}. \quad (3.42)$$

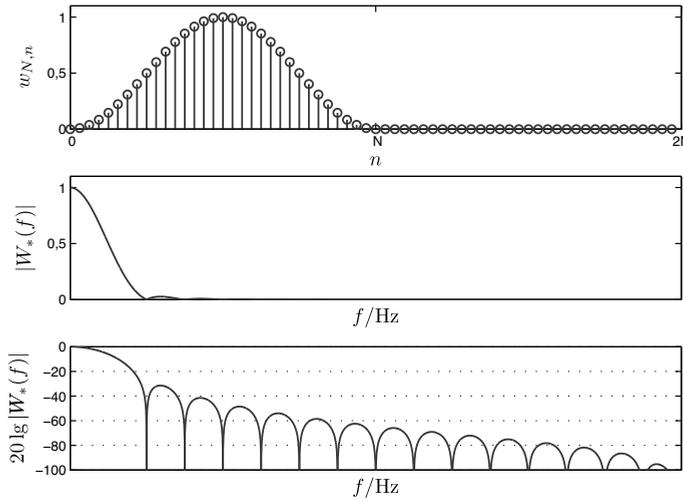


Abbildung 3.10. Hann-Fenster.

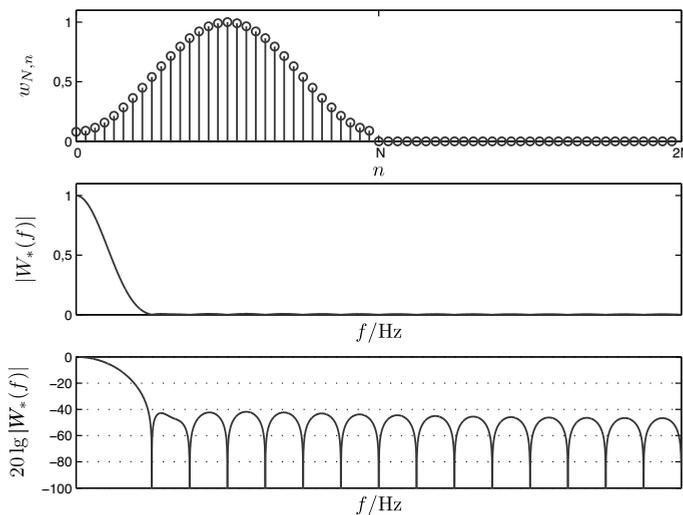


Abbildung 3.11. Hamming-Fenster.

3.4 Verwendete Hardware und ihre Ansteuerung

Mit der zunehmenden Leistungsfähigkeit von PCs zeichnet sich ein Trend weg von spezialisierten *Stand-alone*-Messgeräten hin zu flexibel konfigurierbaren PC-basierten Analyse-Tools ab. In diesem Versuch wird ein PC-gestütztes Komplettsystem zur digitalen Signalverarbeitung eingesetzt. Die Struktur dieses Systems sowie die jeweils verwendeten Schnittstellen sind Abb. 3.12 zu entnehmen. Die Ansteuerung erfolgt

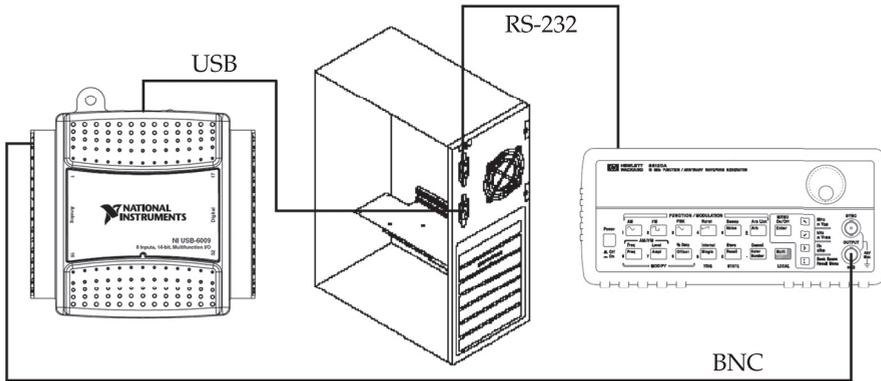


Abbildung 3.12. Hardware-Struktur.

über *LabVIEW*. In den folgenden Abschnitten werden die für die einzelnen Schritte verwendeten Hardware-Komponenten und deren Schnittstellen zum PC vorgestellt.

Als Signalquelle fungiert der Funktionsgenerator *HP 33120A* der Firma *Hewlett Packard*, der Signale nach dem Prinzip der direkten digitalen Synthese (DDS) erzeugt. Er wird bei den vorgefertigten VIs mittels der seriellen RS-232-Schnittstelle über den COM-Port des PCs angesteuert. In diesem Betriebsmodus ist der Funktionsgenerator nicht manuell bedienbar.

Die Steuerung des Funktionsgenerators erfolgt anhand von **SCPI**¹-Befehlen. Dafür werden Blöcke verwendet, die über die *LabVIEW*-Support-Seite www.ni.com zum Download zur Verfügung stehen und speziell auf diesen Funktionsgenerator abgestimmt wurden. Diese werden bei der Programmierung in *LabVIEW* ebenso angeordnet wie einzelne SCPI-Kommandos. Alternativ wäre die Steuerung über eine detaillierte Programmierung in *LabVIEW* oder über ein Terminal-Programm möglich.

3.4.1 Signalerfassung

Der Funktionsgenerator wird an das USB-Datenerfassungsmodul *USB-6008* angeschlossen, welches über acht Analogeingänge (*analog input, AI*), zwei Analogausgänge (*analog output, AO*) und zwölf digitale Ein-/Ausgänge (DIO) verfügt. Schließen Sie die Anschlusskabel an den Eingang *AI0* an, da die vorhandenen VIs ausschließlich Daten von diesem Eingang lesen. Eingangsspannungen dürfen ± 10 V nicht überschreiten!

Mit dem *USB-6008* können Eingangssignale sowohl differentiell (Spannungsdifferenz zwischen zwei Leitern) als auch in Bezug eines Leiters zu Masse (*reference single-ended, RSE*) erfasst werden. In diesem Versuch wird ausschließlich im differentiellen Modus gearbeitet. Dazu muss ein Kabel mit dem Eingang *AI0+* und eines mit *AI0-*

¹SCPI (*Standard Commands for Programmable Instruments*, sprich: „skippy“) ist eine auf dem ASCII-Zeichensatz basierende Sprache für Test- und Messgeräte.

verbunden werden. Tabelle 3.1 zeigt einen Überblick über die wichtigsten Leistungsmerkmale des *USB-6008*.

Tabelle 3.1. Eckdaten des Datenerfassungsmoduls *USB-6008*.

Anzahl analoger Eingänge	8
Anzahl analoger Ausgänge	2
Anzahl digitaler Ein-/Ausgänge	12
Auflösung der A/D- und D/A-Umsetzer	12 bit
Stabilität der Zeitbasis	50 ppm

Die zur Messwerterfassung und Gerätekommunikation von *LabVIEW* unterstützten Schnittstellen und die darüber angeschlossenen Hardwarekomponenten lassen sich über den Measurement and Automation Explorer (MAX) konfigurieren, testen und – soweit möglich – kalibrieren. Der MAX ist jedoch nicht Gegenstand dieses Versuchs.

Die Messwertaufnahme wird in *LabVIEW* mit DAQ (für *Data AcQuisition*) bezeichnet. In diesem Versuch werden folgende Blöcke verwendet:

- DAQ-Assistent (Express-VI): Erlaubt die Konfiguration des Datenerfassungsmoduls und den Zugriff auf sämtliche Funktionen und Parameter über ein Dialogfenster vor Ausführung des VIs.
- DAQmx Virtuellen Kanal erzeugen: Erstellt einen Kanal zur Spannungsmessung.
- DAQmx Timing: Bestimmt Parameter für die Abtastrate.
- DAQmx Lesen: Liest Daten von der Messkarte.
- DAQmx Task starten: Startet eine Messung mit den vorher definierten Parametern.
- DAQmx Warten: Wartet, bis die Messung abgeschlossen ist.
- DAQmx Task zurücksetzen: Löscht den erstellten Messkanal.

Die DAQmx-Blöcke erlauben dabei den dynamischen Zugriff auf die einzelnen DAQ-Funktionen während des laufenden VIs und werden in den bereits vorgefertigten VIs verwendet.

3.4.2 Signalauswertung und Signalpräsentation

Sowohl Signalauswertung als auch Signalpräsentation werden in diesem Versuch mittels *LabVIEW* ausgeführt. Somit wird der PC als vollständig konfigurierbares Messgerät und Display verwendet.

3.5 Versuchsvorbereitende Aufgaben

Aufgabe 3.1: Abtasttheorem

Gegeben sei das Signal:

$$x(t) = \sin(2\pi \cdot 250 \text{ Hz} \cdot t). \quad (3.43)$$

Zur rechnergestützten Weiterverarbeitung wird das Signal mit der Abtastfrequenz $f_A = 400 \text{ Hz}$ abgetastet.

- Ist diese Abtastfrequenz laut Abtasttheorem ausreichend? Begründung!
- Wie unterscheiden sich das ursprüngliche Spektrum $X(f)$ und das Spektrum $X_*(f)$ des abgetasteten Signals?
- Berechnen Sie die Fourier-Transformierte $X_*(f)$ des abgetasteten Signals. Geben Sie den Anteil von $X_*(f)$ im Nyquist-Band an!
- Ermitteln Sie das Zeitsignal durch Fourier-Rücktransformation von $X_*(f)$.
- Das Abtasttheorem liefert eine theoretische Aussage. Was ist bei Abtastvorgängen in realen Anwendungen zu beachten?

Aufgabe 3.2: Anti-Aliasing-Filter

Gegeben sei ein kontinuierliches Eingangssignal $x(t)$ mit konstanter Leistungsdichte (weißes Rauschen)

$$|S_{xx}(f)| = \text{const} = A. \quad (3.44)$$

Das Signal wird mit der Abtastfrequenz f_A abgetastet. Als Anti-Aliasing-Filter kommt ein Filter 1. Ordnung mit der Grenzfrequenz $f_g < f_A/2$ und folgender Übertragungsfunktion zum Einsatz:

$$G(f) = \frac{1}{1 + j \cdot \frac{f}{f_g}}. \quad (3.45)$$

- Wie groß ist das Signal-Rausch-Verhältnis, wenn der Nutzsinalbereich auf die Hälfte der maximalen Bandbreite $f_g = f_A/4$ festgelegt wird? Eine passende Formel für das Signal-Rausch-Verhältnis ist beispielsweise in [PL15] zu finden.
- Ist ein Filter 1. Ordnung als Anti-Aliasing-Filter geeignet?
- Wie groß ist das Signal-Rausch-Verhältnis für ein Anti-Aliasing-Filter der Ordnung $n = 4$ und einer genutzten Bandbreite von $f_g = f_A/10$?
- Würden Sie ein Anti-Aliasing Filter hardwaremäßig analog oder digital realisieren? Begründen Sie Ihre Aussage!

Aufgabe 3.3: Mittelwertbildung bei endlicher Abtastdauer

In der Praxis erfolgt die Abtastung nicht in unendlich kurzer Zeit, sondern über eine endliche Zeitspanne Δt . Dies impliziert eine Mittelwertbildung, die das ideale Abtastsignal verfälscht (siehe Abschn. 3.2.4). Die Mittelwertbildung wirkt sich je nach Signalfrequenz f_s unterschiedlich aus. Man unterscheidet dabei drei Bereiche:

$$1. |f_s| \leq \frac{1}{2\Delta t}, \quad (3.46)$$

$$2. |f_s| \approx \frac{1}{\Delta t}, \quad (3.47)$$

$$3. |f_s| > \frac{1}{\Delta t}. \quad (3.48)$$

- a) Wie wird das Nutzsinal in den drei Bereichen infolge der Mittelwertbildung verändert?
- b) In welchem Bereich sollte daher das Nutzsinal liegen?

Aufgabe 3.4: Quantisierungstheorem

- a) Aus der Varianz des Quantisierungsrauschens lässt sich, bei gegebener Auflösung des Quantisierers in Bit, ein SNR angeben. Berechnen Sie das SNR in dB für 1 bit bis 10 bit in Schritten von einem Bit.
- b) Voraussetzung für die approximative Erfüllung des Quantisierungstheorems ist, dass die Amplitude des Eingangssignals mehrere Quantisierungsstufen durchläuft. Dies setzt ein dynamisch veränderliches Signal voraus. Welches Problem tritt bei langsam veränderlichen Prozessen auf?

Aufgabe 3.5: Dithering

- a) Welche Signale sind als Dithersignale geeignet?
- b) Wie sollte das Spektrum des Dithersignals beschaffen sein?
- c) Wodurch wird der Genauigkeitsgewinn erkauft?

3.6 Versuchsaufgaben

Erstellen Sie im Ordner `D:\PDSV\Gruppen` ein Verzeichnis für Ihre Gruppe. Legen Sie alle Dateien, die Sie im Laufe des Versuchs erstellen, nur in diesem Ordner ab!

Im Ordner `D:\PDSV\Versuch_3` finden Sie alle vorgefertigten VIs, die Sie für die einzelnen Aufgaben benötigen. Wenn Sie an diesen VIs Veränderungen vornehmen, dann überschreiben Sie die VIs in diesem Verzeichnis bitte nicht, sondern kopieren Sie die bereitgestellten VIs in Ihren Gruppenordner.

Aufgabe 3.6: Messwertaufnahme mit LabVIEW

Programmieren Sie ein VI, das die Messwerte vom Funktionsgenerator über das Express-VI DAQ-Assistent in *LabVIEW* einliest und einer FFT unterzieht. Gehen Sie dazu folgendermaßen vor:

1. Schalten Sie den Funktionsgenerator ein und lassen Sie die Einstellungen unverändert (siehe Tab. 3.2).

Tabelle 3.2. Einstellungen Funktionsgenerator.

Funktion:	Sine
Frequenz:	10 Hz
Amplitude:	100 mVPP
Offset:	0 VDC

2. Starten Sie *LabVIEW* und öffnen Sie ein leeres VI. Speichern Sie das VI unter *Messung.vi* im Gruppenverzeichnis ab.
3. Wechseln Sie ins Fenster Blockdiagramm und rufen Sie durch einen Rechtsklick die Funktionenpalette auf. Wählen Sie unter *Express* → *Eingabe* das *Express-VI DAQ-Assistent* aus und platzieren Sie es im Blockdiagramm. Das Dialogfenster für den DAQ-Assistenten wird geöffnet.
4. Wählen Sie *Signale erfassen* → *Analoge Erfassung* → *Spannung* aus.
5. Der physikalische Kanal, von dem Daten erfasst werden sollen, ist *A10*. Wählen Sie diesen Kanal aus und klicken Sie auf *Beenden*.
6. Im sich nun öffnenden Fenster ändern Sie die folgenden Einstellungen:
 - Setzen Sie den Signaleingangsbereich auf ± 1 V.
 - Wählen Sie unter *Eigene Skalierungen* die Skalierung *Volt*.
 - Stellen Sie die Schaltungsart auf *Differentiell*.
 - Markieren Sie unter *Erfassungsmodus* die Auswahl *Endliche Anzahl*.
 - Setzen Sie *Zu lesende Samples* auf 100.
 - Setzen Sie *Rate (Hz)* auf 10 k und klicken Sie auf *OK*.

Der Block DAQ-Assistent wird nun initialisiert.

7. Führen Sie erneut einen Rechtsklick auf eine freie Stelle des Blockdiagramms aus, wählen Sie unter *Express* → *Analyse* den Block *Spektrum ermitteln* und legen Sie ihn im Blockdiagramm ab. Ein Konfigurationsdialog wird geöffnet. Wählen Sie unter *Ergebnis* die Einstellung *Linear* aus und unter Fenster *Kein*. Behalten Sie die übrigen Einstellungen bei. Verbinden Sie nun die Blöcke DAQ-Assistent und *Spektrum ermitteln* miteinander.
8. Mit einem Rechtsklick auf den Ausgang *FFT-(RMS)* des Blocks *Spektrum ermitteln* öffnen Sie das entsprechende Kontextmenü. Wählen Sie *Erstelle* → *Graphanzeige*.

9. Kopieren Sie den soeben erstellten Graphen durch Drücken der Strg-Taste und Linksklick auf den entsprechenden Graphen (Maustaste gedrückt halten). Ziehen Sie ihn an seine Zielposition und platzieren Sie ihn durch Loslassen der linken Maustaste.
10. Führen Sie einen Doppelklick auf die Beschriftung des kopierten Graphen aus und benennen Sie ihn in „Zeitsignal“ um. Verbinden Sie ihn mit den übrigen Blöcken in sinnvoller Weise.
11. Wechseln Sie in die Frontpanel-Ansicht und machen Sie bei beiden Graphen durch Rechtsklick auf den jeweiligen Graphen und Auswahl von Sichtbare Objekte → Graph-Palette die Graphenpalette sichtbar.
12. Starten Sie das VI durch Klick auf den entsprechenden Pfeil in der Symbolleiste oder Eingabe von Strg+R.

Hinweis: Durch Klick auf das Lupensymbol in der Graphenpalette erhalten Sie Zugang zu verschiedenen Zoom-Funktionen. Um jedoch wirklich in die Graphen hineinzoomen zu können, müssen Sie im Kontextmenü des jeweiligen Graphen die Funktion Auto-Skalierung deaktivieren.

- a) Liefert das VI eine sinnvolle Darstellung von Signal und Spektrum?
- b) Passen Sie nun die Beobachtungsdauer an. Wechseln Sie dazu in das Blockdiagramm und führen Sie einen Doppelklick auf den Block DAQ-Assistent aus. Tragen Sie nun unter Zu lesende Samples einen geeigneten Wert ein. Starten Sie das VI noch einmal. Liefert das VI jetzt den erwarteten Verlauf für Signal und Spektrum?

Aufgabe 3.7: Abtasttheorem

In der Aufgabe 3.6 haben Sie gelernt, wie sich mit *LabVIEW* Messwerte erfassen und weiterverarbeiten lassen. Dabei wurden die Einstellungen am Funktionsgenerator manuell vorgenommen. Nun soll dieser direkt aus *LabVIEW* angesteuert werden. Öffnen Sie hierzu das VI *Abtastung.vi*. Sie können sich im Blockdiagramm den Aufbau des VIs anschauen. Hier genügt kein einfaches Express-VI mehr wie in der vorangegangenen Aufgabe; das VI ist entsprechend umfangreicher geworden.

Wechseln Sie in das Frontpanel und starten Sie das VI. Geben Sie die folgenden Einstellungen für die Ansteuerung des Funktionsgenerators ein:

- Frequenz: 250 Hz,
- Funktion: Sine.

Die Amplitude beträgt 1 V (*Peak–Peak*). Klicken Sie den Button OK, um die Einstellungen an den Funktionsgenerator zu senden.

- a) Stellen Sie zunächst die Abtastfrequenz $f_A = 400$ Hz ein. Starten Sie die Messung durch Klick auf den Button Start. Ist das Abtasttheorem erfüllt? Vergleichen Sie das angezeigte Spektrum mit dem von Ihnen in der Aufgabe 3.1 errechneten Spektrum.

- b) Wie müssen Sie die Abtastzeit wählen, damit das Abtasttheorem erfüllt ist?
- c) Ist die von Ihnen bestimmte Abtastzeit ausreichend, um eine sinnvolle Rekonstruktion zu erhalten?
- d) Ab welcher Abtastzeit wird eine zufriedenstellende Signalrekonstruktion erreicht?

Aufgabe 3.8: Anti-Aliasing-Filter

Diese Aufgabe soll die Anforderungen an Anti-Aliasing-Filter veranschaulichen und die Leistungsfähigkeit der verschiedenen Filter deutlich machen. Öffnen Sie hierzu das VI `AntiAliasingFilter.vi`. Dieses stellt Ihnen ein Mischsignal aus einem Dreieckssignal und einem Sinussignal zur Verfügung. Amplitude und Frequenz beider Signale sind variabel einstellbar. Dabei werden das Dreieckssignal als Nutzsinal und das Sinussignal als Störsignal interpretiert. Mittels eines Anti-Aliasing-Filters soll das Störsignal möglichst unterdrückt werden, ohne das Nutzsinal zu beeinflussen. Die Art des Filters, die Filterordnung und die Knickfrequenz können Sie variieren.

- a) Nehmen Sie zunächst folgende Einstellungen vor:
 - Dreieck: Amplitude $A_D = 1$, Frequenz $f_D = 100$ Hz,
 - Sinus: Amplitude $A_S = 1$, Frequenz $f_S = 1000$ Hz,
 - Filtertyp: Butterworth, Filterordnung 1, Knickfrequenz $f_g = 100$ Hz.
 Starten Sie das VI und betrachten Sie das gefilterte Gesamtsignal sowie die Spektren vor und nach der Filterung.
 1. Ist die Filterordnung ausreichend, um das Störsignal vollständig zu unterdrücken? Sollte das nicht der Fall sein, welche Filterordnung ist dann mindestens erforderlich?
 2. Lässt das Filter das Nutzsinal ungestört passieren? Falls nicht, passen Sie das Filter entsprechend an, um sowohl das Störsignal zu eliminieren als auch das Nutzsinal möglichst wenig zu stören. Ermitteln Sie die minimale Filterordnung, die benötigt wird, und bestimmen Sie eine geeignete Knickfrequenz.
 3. Verändern Sie nun den Filtertyp. Vergleichen Sie die Filterwirkung der einzelnen Filtertypen. Geben Sie für jedes Filter wieder eine geeignete Knickfrequenz und Filterordnung an.
- b) Nun soll die Frequenz des Dreieckssignals auf $f_D = 200$ Hz erhöht werden. Wählen Sie wieder das Butterworth-Filter aus. Geben Sie die minimale Filterordnung und eine geeignete Knickfrequenz an, um das Dreieckssignal möglichst unverändert passieren zu lassen und dabei das Störsignal möglichst stark zu unterdrücken.
- c) Erhöhen Sie die Frequenz auf $f_D = 300$ Hz und abschließend auf $f_D = 400$ Hz. Lässt sich das Störsignal noch herausfiltern und gleichzeitig das Dreieckssignal erhalten? Falls ja, geben Sie geeignete Werte für die Filterordnung und die Knickfrequenz an. Sind die Anforderungen an das Filter noch realisierbar? Berücksichtigen Sie dazu, wie das Filter in praktischen Anwendungen realisiert werden muss.

Aufgabe 3.9: Leckeffekt

Nun soll der Leckeffekt anhand eines aus zwei Frequenzen bestehenden Summensignals untersucht werden. Gehen Sie dazu wie folgt vor:

- Erstellen Sie ein neues VI und speichern Sie es im Verzeichnis Ihrer Gruppe unter `Leakage.vi` ab.
- Durch Rechtsklick im Blockdiagramm öffnen Sie die Funktionenpalette. Wählen Sie unter `Express` → `Eingabe` den Block `Signal simulieren` und legen Sie ihn im Blockdiagramm ab. Nehmen Sie folgende Einstellungen vor:
 - Signaltyp: `Sinus`
 - Frequenz (Hz): `6,25`
 - Amplitude: `1`
 - Samples pro Sekunde (Hz): `100`
 - Sample Anzahl: `128`
- Kopieren Sie den Block und ändern Sie folgende Einstellungen im neuen Block:
 - Frequenz (Hz): `9,375`
 - Amplitude: `0,01`
- Wählen Sie aus der Funktionenpalette einen `Additionsblock` (`Programmierung` → `Numerisch` → `Addieren`) und platzieren Sie ihn im Blockdiagramm. Verbinden Sie die Ausgänge der beiden Signalblöcke mit dem `Additionsblock` und erzeugen Sie an dessen Ausgang einen `Graphen` (Rechtsklick auf den Ausgang → `Erstellen` → `Graphanzeige`).
- Ergänzen Sie das Blockdiagramm zusätzlich um einen Block zur Berechnung des Spektrums (Funktionenpalette: `Express` → `Analyse` → `Spektrum ermitteln`) und ändern Sie die folgenden Einstellungen:
 - Ergebnis: `Linear`
 - Fenster: `Kein`Verbinden Sie den Ausgang des `Additionsblocks` mit dem `Signale-Eingang` des `Spektrum ermitteln`-Blocks und erzeugen Sie am Ausgang (`FFT-(RMS)`) wieder einen `Graphen`.
- Wechseln Sie in die `Frontpanel-Ansicht`. Nehmen Sie an den `Graphen` folgende Einstellungen vor (Rechtsklick auf den `Graph` → `Eigenschaften`):
 - Plots: Lassen Sie sich die einzelnen Werte als Punkte plotten und mit einer gestrichelten Linie verbinden.
 - Skalen: Aktivieren Sie bei der `FFT` zusätzlich die `logarithmische Darstellung` (`Logarithmisch`) der `y-Achse`.Wenn Sie die `Graphen` anklicken, können Sie sie in die Länge ziehen und damit die `x-Achse` strecken, um eine bessere Darstellung zu erhalten.

a) Starten Sie die Simulation und betrachten Sie `Signal` und `Spektrum`. Tritt ein `Leckeffekt` auf? Begründen Sie anhand von `Beobachtungsdauer` und `Signalfrequenzen`!

- b) Wechseln Sie wieder in das Blockdiagramm und verändern Sie *bei beiden Signalen* die Anzahl N der Abtastwerte. Wählen Sie die folgenden Werte:

$$N \in \{121, 112, 104, 96, 64\}. \quad (3.49)$$

Für welche Werte N tritt ein besonders deutlicher Leckeffekt auf? Begründen Sie! Betrachten Sie hierzu die Beobachtungsdauer und die Signalfrequenzen.

- c) Der Leckeffekt lässt sich durch Einsatz eines geeigneten Fensters reduzieren. Verändern Sie hierzu das VI `Leakage.vi` wie folgt:
- Wechseln Sie in das Blockdiagramm. Stellen Sie die Anzahl N der Abtastwerte ein, für die sich der Leckeffekt am stärksten auswirkt.
 - Kopieren Sie den Block zur Berechnung des Spektrums einschließlich Anzeige-Graphen und geben Sie das Zeitsignal auf den neuen Spektrum ermitteln-Block. Öffnen Sie den Block durch Doppelklick und verändern Sie unter `Fenster` die Art des verwendeten Fensters.
 - Wechseln Sie wieder in das Frontpanel und starten Sie die Simulation. Vergleichen Sie die Spektren mit und ohne Fensterung.

Führen Sie diesen Vergleich für die verschiedenen verfügbaren Fenster durch. Mit welchem Fenster wird der Leckeffekt am besten unterdrückt?

Aufgabe 3.10: Jitter

Öffnen Sie das VI `Jitter.vi`, das einen Sinus mit variabler Frequenz erzeugt. Das Signal wird einmal zu äquidistanten Zeitpunkten $i \cdot t_A$ mit einer Abtastzeit von $t_A = 0,1$ ms abgetastet. Zusätzlich wird die Abtastzeit mit einem zufälligen Fehler mit der Standardabweichung $\sigma = 0,25 \cdot t_A$ beaufschlagt.

Im Frontpanel sind der zeitliche Fehler (Jitter), das äquidistant abgetastete Signal sowie das mit Jitter abgetastete Signal zu sehen. Außerdem werden die Abweichung zwischen unverfälschtem und verfälschtem Signal sowie das SNR in dB angezeigt.

- a) Stellen Sie die Signalfrequenz 100 Hz ein. In welchem Bereich liegt der Fehler?
- b) Erhöhen Sie die Signalfrequenz schrittweise um 100 Hz bis zu einer Frequenz von 1000 Hz. Wie verändert sich der Fehler? Begründen Sie Ihre Antwort.

Aufgabe 3.11: Mittelwertbildung bei endlicher Abtastdauer

In realen Anwendungen wird für die Abtastung eine gewisse Zeit benötigt, was einer Integration über eine endliche Zeitspanne Δt entspricht. Um die damit verbundene Mittelwertbildung zu veranschaulichen, steht Ihnen das VI `Mittelwertbildung.vi` zur Verfügung, welches einen Sinus mit variabler Frequenz liefert. Für die Berechnung

der Abtastwerte $\bar{y}(t_i)$ wird der Sinus über das Zeitintervall Δt gemittelt:

$$\bar{y}(t_i) = \frac{1}{\Delta t} \int_{\tau=t_i-\Delta t/2}^{t_i+\Delta t/2} y(\tau) d\tau. \quad (3.50)$$

Die Abtastfrequenz beträgt $f_A = 1000$ kHz.

- a) In der Aufgabe 3.3 wurden drei Frequenzbereiche betrachtet, in denen sich die Mittelung unterschiedlich auswirkt. Dies ist nun zu überprüfen.
 - Stellen Sie das Mittelungsintervall auf $\Delta t = 0,1$ ms ein.
 - Starten Sie das VI und wählen Sie für jeden der drei Bereiche eine geeignete Signalfrequenz aus. Stimmen Ihre Überlegungen aus der Versuchsvorbereitung mit den Simulationsergebnissen überein?
- b) Wechseln Sie in das Blockdiagramm und öffnen Sie den Block Signal simulieren. Ändern Sie den Signaltyp von Sinus auf ein Dreieck. Wechseln Sie danach wieder in das Frontpanel. Variieren Sie die Signalfrequenz zwischen 1000 Hz und 5000 Hz in Schritten von 1000 Hz. In welchem der drei Bereiche liegt das Signal für diese Frequenzen? Wie wirkt sich die Mittelwertbildung auf das Signal aus?

Aufgabe 3.12: Moving-Average-Filter als Tiefpassfilter

Mit dem **Moving-Average-Filter** (kurz: MA-Filter) kann eine einfache Tiefpassfilterung realisiert werden. Hierzu wird der gleitende Mittelwert über $N = 2m + 1$ Messwerte berechnet:

$$\bar{x}_k = \frac{1}{N} \sum_{i=k-m}^{k+m} x_i. \quad (3.51)$$

Öffnen Sie das VI MA-Filter.vi und wechseln Sie in das Frontpanel. Das VI erzeugt einen Sinus der Frequenz $f_s = 10$ Hz, der mit weißem Gauß'schem Rauschen überlagert ist. Ein MA-Filter soll das Rauschen unterdrücken. Dabei sind die halbe Fensterbreite m des Filters sowie die Standardabweichung σ des Rauschens variabel. Die Abtastfrequenz beträgt $f_A = 10$ kHz.

- a) Stellen Sie zunächst eine Standardabweichung von 0,1 ein. Wie groß sollte die halbe Fensterbreite m des Filters mindestens sein, um das Rauschen weitgehend zu unterdrücken?
- b) Erhöhen Sie nun schrittweise die Standardabweichung σ . Bis zu welchem Wert von σ lässt sich das Rauschen mit dem MA-Filter unterdrücken, ohne das Nutzsignal zu verfälschen? Welchen Wert haben Sie in diesem Fall für m gewählt?
- c) Welcher maximale Wert von m ist bei der gegebenen Signalfrequenz und Abtastfrequenz noch sinnvoll?

- d) Wechseln Sie in das Blockdiagramm. Öffnen Sie den Block Signal simulieren durch Doppelklick. Ändern Sie den Signaltyp auf ein Dreiecksignal; behalten Sie die restlichen Einstellungen bei. Schließen Sie den Block und wechseln Sie in das Frontpanel. Ändern Sie die Standardabweichung auf 0,1. Variieren Sie m von 10 bis 80. Wie wird das Nutzsignal in diesem Bereich durch das MA-Filter beeinflusst?

Aufgabe 3.13: Quantisierung

Mit dem VI Quantisierung.vi soll der Einfluss der Quantisierung auf das Signal und auf das SNR untersucht und die Gültigkeit des Quantisierungstheorems experimentell überprüft werden. Der Funktionsgenerator wird direkt vom VI angesteuert:

- Die Einstellungen werden mit dem OK-Button unter Einstellungen übernehmen an den Funktionsgenerator gesendet.
 - Mit dem Taster Messwertaufnahme starten werden die Daten vom DAQ-Gerät eingelesen (Dauer: 1 Sekunde) und das SNR in dB, der Quantisierungsfehler, die Breite einer Quantisierungsstufe in Volt sowie die Amplitudendichten des eingelesenen und quantisierten Signals berechnet und angezeigt.
 - Ferner können die Quantisierung in Bit, die maximale bzw. minimale Amplitude des Quantisierers sowie die Beobachtungsdauer für die Berechnung der Amplitudendichten variiert werden.
 - Durch Klicken auf den Stop-Button wird das VI gestoppt.
- a) Starten Sie das VI. Nehmen Sie die Einstellungen gemäß Tabelle 3.3 vor und starten Sie die Messwertaufnahme. Machen Sie sich die jeweiligen Verläufe und Zusammenhänge der dargestellten Funktionen klar.

Tabelle 3.3. Einstellungen von Quantisierung.vi.

Funktion:	Sine
Frequenz (Hz):	100
Amplitude (V _{pp}):	1
Beobachtungsdauer (s):	1
Maximale Amplitude ($\pm V$):	1
Quantisierung (Bit):	1

- b) Erstellen Sie eine Messwertreihe für das SNR in dB, wobei Sie die Quantisierung in Bit im Bereich $\{1, \dots, 10\}$ unter Beibehaltung der sonstigen Parameter in Schritten von einem Bit erhöhen.
- c) Wie ändert sich das SNR? Stimmen die Messwerte mit den Berechnungen aus Aufgabe 3.4 überein? Falls nicht, womit lässt sich dies begründen?

Aufgabe 3.14: Dithering

Ein langsam veränderliches Messsignal wird mit einer Auflösung von 2 Bit digitalisiert. Anschließend soll das Signal mit einer Auflösung von 16 Bit weiterverarbeitet werden. Durchläuft das Messsignal während der Messaufnahme nur wenige Quantisierungsstufen, ist seine charakteristische Funktion sehr weit ausgedehnt und das Quantisierungstheorem unter Umständen nicht erfüllt (Abschn. 3.2.6.4). Daher wird dem Messsignal vor der Quantisierung ein Dither überlagert. Nach der Quantisierung wird der Dither mit Hilfe eines Moving-Average-Filters entfernt.

- Öffnen Sie das VI Dither.vi.
 - Als Nutzsignal dient ein Sinus mit wählbarer Frequenz. Wählen Sie hierfür 0,8 Hz.
 - Wählen Sie eine Auflösung des Messgerätes von 2 Bit.
 - Als Dither dient ein Sinus, dessen Frequenz Sie frei vorgeben können. Die Abtastfrequenz beträgt $f_A = 100$ kHz.
- a) Wählen Sie eine geeignete Ditherfrequenz und starten Sie das VI. Vergleichen Sie das SNR, wenn das Messsignal direkt quantisiert wird, mit dem SNR, wenn vor der Quantisierung der Dither überlagert wird.
- b) Zur Rekonstruktion des Messsignals wird ein Butterworth-Filter 8. Ordnung mit einstellbarer Knickfrequenz verwendet. Der Tiefpass wird zum einen auf das direkt quantisierte Messsignal angewandt und zum anderen auf das quantisierte Signal, dem zuvor der Dither überlagert wurde. Wie müssen Sie die Knickfrequenz des Tiefpassfilters wählen, um das ursprüngliche Messsignal möglichst gut zu rekonstruieren? Vergleichen Sie für diese Filtereinstellung die Ergebnisse, die sich für das Signal mit Dither ergeben mit denen für das direkt quantisierte Signal.

3.7 Literatur

- [KK12] KARL-DIRK KAMMEYER und KRISTIAN KROSCHEL: *Digitale Signalverarbeitung – Filterung und Spektralanalyse mit MATLAB-Übungen*. Springer Vieweg, Wiesbaden, 8. Auflage, 2012.
- [PJ15] FERNANDO PUENTE LEÓN und HOLGER JÄKEL: *Signale und Systeme*. De Gruyter Oldenbourg, Berlin, 6. Auflage, 2015.
- [PL15] FERNANDO PUENTE LEÓN: *Messtechnik – Systemtheorie für Ingenieure und Informatiker*. Springer, Berlin Heidelberg, 10. Auflage, 2015.

Versuch 4

Analoge und digitale Filterung

4

4	Analoge und digitale Filterung	
4.1	Einleitung	99
4.2	Theorie	100
4.2.1	Grundlagen	100
4.2.1.1	Grundbegriffe der Systemtheorie	100
4.2.1.2	Zusammenhang zwischen Polstellen, Frequenzgang und Gruppenlaufzeit	102
4.2.1.3	Grundbegriffe der Filterung	106
4.2.2	Anwendungsgebiete von Filtern	106
4.2.3	Analoge Filter	108
4.2.3.1	Butterworth-Filter	108
4.2.3.2	Tschebyscheff-Filter	109
4.2.3.3	Cauer-Filter	110
4.2.4	Digitale Filter	111
4.2.4.1	FIR-Filter	111
4.2.4.2	IIR-Filter	113
4.2.5	Auswirkungen endlicher Rechengenauigkeit	120
4.2.5.1	Quantisierung der Koeffizienten	120
4.2.5.2	Quantisierungsrauschen	122
4.3	Versuchsaufbau	125
4.3.1	Beschreibung des Versuchs	125
4.3.1.1	Beschreibung der Hardware	127
4.3.1.2	Beschreibung der Programme	129
4.3.2	Versuchsvorbereitende Aufgaben	133
4.3.3	Versuche am Versuchstag	135
4.3.3.1	Versuche mit der Analog-Hardware	135
4.3.3.2	Versuche am PC	135
4.4	Literatur	138

4 Analoge und digitale Filterung

4.1 Einleitung

Lineare Filter ermöglichen es, den Frequenzgehalt von Signalen gezielt zu beeinflussen, um nicht erwünschte Signalanteile wie etwa hochfrequente Störungen zu unterdrücken.

Ursprünglich wurden *digitale* Filter lediglich zur Simulation von mess- und regelungstechnischen Anwendungen verwendet. Durch die Einführung der **Compact Disc** (CD) hat die digitale Filterung auch auf dem Massenmarkt zunehmende Verbreitung gefunden. So werden heute in immer mehr Anwendungsgebieten, die bislang analogen Filtern vorbehalten waren, digitale Filter eingesetzt.

Im Vergleich zu analogen Filtern liefern digitale Filter ein besseres Signal-Rausch-Verhältnis (SNR). Auch müssen keine Effekte wie Bauteiltoleranzen, -drift und -alterung berücksichtigt werden. Die Flexibilität einer einzigen Hardware-Realisierung ist weitaus höher als beim analogen Pendant, da durch Anpassung der Filterkoeffizienten sehr viele verschiedene Filtereigenschaften auf einer vorgegebenen Hardware implementiert werden können. Weiterhin ist es mit analogen Filtern beinahe unmöglich, steilflankige oder schmalbandige Frequenzverläufe zu erreichen und gleichzeitig den schaltungstechnischen Aufwand in einem erträglichen Rahmen zu halten. Als Nachteil digitaler gegenüber analogen Filtern wäre jedoch zu nennen, dass durch die Rechendauer für das Filter dessen Maximalfrequenz limitiert wird. Hierbei bedeutet Maximalfrequenz die Frequenz, mit der die Werte des gefilterten Signals zur Verfügung gestellt werden können.

Um der besonderen Bedeutung von Filtern gerecht zu werden, beschäftigt sich dieser Versuch mit der analogen, jedoch hauptsächlich mit der digitalen Filterung. Dabei werden Analogien zwischen den beiden Filtertypen aufgezeigt.

Aufgrund der steigenden Komplexität und der immer kürzer werdenden Entwicklungszeit von modernen Steuerungen und Reglern werden hohe Anforderungen an die Softwareentwicklung gestellt. Zwei leistungsfähige Werkzeuge hierzu sind **Rapid Control Prototyping** (RCP) und **Hardware in the Loop** (HiL). Rapid Control Prototyping ist eine effiziente Methode zur rechnergestützten Reglerentwicklung. Der Entwurf einer Automatisierungslösung erfolgt dabei, ausgehend von einer Aufgabenstellung, schrittweise und systematisch. Zunächst wird ein Modell des zu regelnden Systems und darauf basierend dann ein Regelungs- oder Steuerungsalgorithmus des vorliegenden Systems entwickelt. Als nächster Schritt folgt zunächst die Simulation der entworfenen Software und anschließend deren Implementierung. Zum Testen des auf diese Weise entwickelten Produkts kann das HiL-Verfahren eingesetzt werden. Dabei werden ein oder mehrere Elektronikkomponenten (z. B. Steuergeräte) an

einen HiL-Simulator angeschlossen und in Echtzeit getestet. Auf dem HiL-Simulator ist dabei ein Modell des betrachteten Systems implementiert. Er beaufsichtigt das zu testende reale System mit Eingangssignalen und erfasst die Antwort des Systems.

Ein führender Hersteller von solchen Entwicklungswerkzeugen ist die Firma *dSPACE*. Um einen Einblick in die o. g. Methoden und die dazu verwendete Hardware zu erhalten, wird im zweiten Teil des Versuchs die *dSPACE AutoBox* eingesetzt. Zunächst soll mit dem erworbenen theoretischen Wissen in *Matlab* ein Tiefpassfilter entworfen werden. Anschließend werden mittels der *dSPACE AutoBox* die Beschleunigungssignale eines Crashtest-Versuchsfahrzeugs aufgezeichnet und mit dem entworfenen Filter gefiltert.

4.2 Theorie

4.2.1 Grundlagen

In diesem Abschnitt werden die Grundlagen der Systemtheorie kurz wiederholt. Für eine ausführlichere Herleitung der einzelnen Zusammenhänge sei auf [PJ15] verwiesen.

4.2.1.1 Grundbegriffe der Systemtheorie

Wir werden uns auf lineare, zeitinvariante Systeme (**LTI-Systeme**, als Abkürzung von *linear, time-invariant*) beschränken. Ein zeitkontinuierliches LTI-System wird durch die **Impulsantwort** $g(t)$ vollständig beschrieben. Zwischen dem **Ausgangssignal** $y_a(t)$ und dem **Eingangssignal** $y_e(t)$ gilt dann

$$y_a(t) = g(t) * y_e(t) \quad (4.1)$$

mit dem Faltungsoperator (*). Die Faltung ist definiert als

$$x(t) * y(t) = \int_{-\infty}^{\infty} x(t - \tau) y(\tau) d\tau. \quad (4.2)$$

Durch Laplace-Transformation von (4.1) erhält man

$$Y_a(s) = G(s) \cdot Y_e(s), \quad (4.3)$$

wobei $G(s)$ als die **Systemfunktion** oder **Übertragungsfunktion** bezeichnet wird. Für ein LTI-System gibt es eine Vielzahl von Darstellungsmöglichkeiten, je nachdem, welche Eigenschaften des Systems zu untersuchen sind. Wenn $G(s)$ in ein Nennerpoly-

nom $N(s)$ und Zählerpolynom $Z(s)$ aufgeteilt wird, erhält man:

$$G(s) = \frac{Z(s)}{N(s)} = \frac{\sum_{j=0}^m b_j s^j}{\sum_{i=0}^n a_i s^i} = c \cdot \frac{\prod_{j=0}^m (s - s_{0j})}{\prod_{i=0}^n (s - s_{\infty i})}, \quad m \leq n. \quad (4.4)$$

Die Stellen s_{0j} sind die Nullstellen von $Z(s)$, auch als *Nullstellen der Übertragungsfunktion* bezeichnet. Die Nullstellen $s_{\infty i}$ von $N(s)$ werden *Polstellen der Übertragungsfunktion* genannt.

Die Fourier-Transformierte der Impulsantwort wird **Frequenzgang** $G(f)$ genannt:

$$\mathcal{F}\{g(t)\} = G(s = j2\pi f) = |G(f)| \cdot e^{j\psi(f)}. \quad (4.5)$$

Die Frequenzachse entspricht dabei der imaginären Achse der s -Ebene, da s im Allgemeinen komplex ist: $s = \delta + j2\pi f$. Der Frequenzgang setzt sich aus dem **Amplitudengang** $|G(f)|$ und dem **Phasengang** $\psi(f)$ zusammen. Neben dem Amplitudengang und dem Phasengang interessiert auch die **Gruppenlaufzeit** $\tau_g(f) = -\frac{1}{2\pi} \frac{d\psi(f)}{df}$, die die Phasenverzerrung beschreibt.

Bisher hat sich die Diskussion auf zeitkontinuierliche Signale und Systeme beschränkt. Sie lässt sich jedoch auch einfach auf zeitdiskrete Signale und Systeme übertragen. Zur Erinnerung: Ein Signal wird mit der **Abtastzeit** t_A abgetastet und als zeitdiskretes Signal als Folge $y_n = y(nt_A)$, $n \in \mathbb{Z}$, angegeben. Damit das stetige Zeitsignal $y(t)$ der Bandbreite B ($Y(f) = 0$ für $|f| \geq B$) vollständig rekonstruierbar ist, muss das **Abtasttheorem** erfüllt sein [PJ15]:

$$t_A \leq \frac{1}{2B}. \quad (4.6)$$

Die **Abtastfrequenz** f_A , in der englischen Literatur f_s (*sampling rate*) genannt¹, muss also größer oder gleich $2B$ sein. Durch die Abtastung im Zeitbereich (Multiplikation mit einem Impulskamm) entsteht im Frequenzbereich eine periodische Wiederholung des Signalspektrums durch die Faltung mit dem korrespondierenden Impulskamm (ein Impulskamm im Zeitbereich korrespondiert ebenfalls mit einem Impulskamm im Frequenzbereich). Wird das Abtasttheorem nicht erfüllt, wird das abgetastete Signal durch spektrale Überlappung verzerrt; dieser Effekt wird **Aliasing** genannt.

Systeme werden im zeitdiskreten Fall mit Differenzgleichungen statt Differentialgleichungen beschrieben. In diesem Fall wird die z -Transformation anstelle der Laplace-Transformation benutzt. Die Umwandlung erfolgt über die konforme Abbildung $z = e^{st_A}$. Eine wichtige Eigenschaft dieser Abbildung ist, dass die imaginäre Achse (Frequenzachse) der s -Ebene auf den Einheitskreis der z -Ebene abgebildet wird

¹Oft wird auch die Bezeichnung $\omega_A = 2\pi f_A$ verwendet.

(siehe auch Abschn. 4.2.1.2). Dadurch wird die linke Halbebene auf das Innere des Einheitskreises abgebildet. Weitere Folgen dieser Abbildung sind in [PJ15] nachzuschlagen.

4.2.1.2 Zusammenhang zwischen Polstellen, Frequenzgang und Gruppenlaufzeit

Die Systemfunktion $G(s)$ aus (4.4) beschreibt zeitkontinuierliche LTI-Systeme. Für zeitdiskrete LTI-Systeme ist die Übertragungsfunktion

$$G(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^q b_k z^{-k}}{1 + \sum_{k=1}^p a_k z^{-k}} = \frac{B(z)}{A(z)} = z^{p-q} \cdot b_0 \cdot \frac{\prod_{k=1}^q (z - z_{0k})}{\prod_{k=1}^p (z - z_{\infty k})} \quad (4.7)$$

definiert. Den Übergang vom Zeitkontinuierlichen zum Zeitdiskreten erhält man wie bereits oben erwähnt durch z-Transformation.

Den *zeitdiskreten Amplitudengang* erhält man durch Betragsbildung von (4.7) für $z = e^{j2\pi f t_A}$:

$$|G(e^{j2\pi f t_A})| = \frac{\left| \sum_{k=0}^q b_k e^{-j2\pi f t_A k} \right|}{\left| 1 + \sum_{k=1}^p a_k e^{-j2\pi f t_A k} \right|} = \frac{|B(e^{j2\pi f t_A})|}{|A(e^{j2\pi f t_A})|}. \quad (4.8)$$

Für den Phasengang gilt dann:

$$\psi(f) = \arg\{G(e^{j2\pi f t_A})\} = \arg\{B(e^{j2\pi f t_A})\} - \arg\{A(e^{j2\pi f t_A})\}. \quad (4.9)$$

Hierbei ist auf die Definition des arctan in den verschiedenen Quadranten zu achten. Hat ein System einen linearen Phasengang, so bedeutet dies, dass alle Signalfrequenzen beim Durchlaufen des Systems die gleiche Zeitverzögerung erfahren. Um einen linearen Phasengang zu erhalten, muss eine der folgenden zwei Bedingungen an das System erfüllt sein:

- Alle Pole müssen im Ursprung liegen und die Nullstellen befinden sich entweder auf dem Einheitskreis oder paarweise am Einheitskreis gespiegelt: $z_{0i} = \rho e^{j\alpha}$, $1/z_{0i}^* = 1/\rho e^{j\alpha}$ [KK12].
- Die Impulsantwort muss entweder symmetrisch ($g_n = g_{-n}$) oder antisymmetrisch ($g_n = -g_{-n}$) sein [PJ15].

Die Gruppenlaufzeit τ erhält man durch Ableitung von $\psi(f)$ nach f . Bei einem linearen Phasengang erhält man also eine konstante Gruppenlaufzeit.

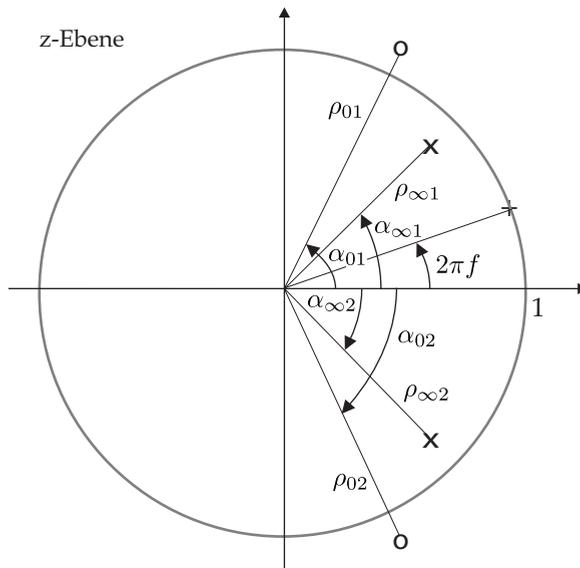


Abbildung 4.1. Pole und Nullstellen in der z-Ebene.

Stellt man den Frequenzgang durch Pole und Nullstellen dar, so erhält man aus (4.7) an der Stelle $z = e^{j2\pi ft_A}$

$$G(e^{j2\pi ft_A}) = e^{j(p-q)2\pi ft_A} \cdot b_0 \cdot \frac{\prod_{k=1}^q (e^{j2\pi ft_A} - \rho_{0k} e^{j\alpha_{0k}})}{\prod_{k=1}^p (e^{j2\pi ft_A} - \rho_{\infty k} e^{j\alpha_{\infty k}}} \quad (4.10)$$

Dabei werden Pol- und Nullstellen in (4.10) in Polarkoordinaten dargestellt (siehe auch Abb. 4.1):

$$z_{\infty k} = \rho_{\infty k} e^{j\alpha_{\infty k}} \quad (4.11)$$

$$z_{0k} = \rho_{0k} e^{j\alpha_{0k}} \quad (4.12)$$

Unter Verwendung der Beziehung

$$e^{j2\pi ft_A} = \cos(2\pi ft_A) + j \sin(2\pi ft_A) \quad (4.13)$$

erhält man schließlich nach Betragsbildung den Amplitudengang [KK12]

$$|G(e^{j2\pi ft_A})| = |b_0| \cdot \frac{\prod_{k=1}^q \sqrt{1 - 2\rho_{0k} \cos(2\pi ft_A - \alpha_{0k}) + \rho_{0k}^2}}{\prod_{k=1}^p \sqrt{1 - 2\rho_{\infty k} \cos(2\pi ft_A - \alpha_{\infty k}) + \rho_{\infty k}^2}} \quad (4.14)$$

Für den Phasengang resultiert demnach

$$\psi(f) = (p - q) 2\pi f t_A + \arg\{b_0\} + \sum_{k=1}^q \arctan \left\{ \frac{\sin 2\pi f t_A - \rho_{0k} \cdot \sin \alpha_{0k}}{\cos 2\pi f t_A - \rho_{0k} \cdot \cos \alpha_{0k}} \right\} - \sum_{k=1}^p \arctan \left\{ \frac{\sin 2\pi f t_A - \rho_{\infty k} \cdot \sin \alpha_{\infty k}}{\cos 2\pi f t_A - \rho_{\infty k} \cdot \cos \alpha_{\infty k}} \right\} \quad (4.15)$$

und die Gruppenlaufzeit wird zu

$$2\pi\tau_g(f) = (q - p) - \sum_{k=1}^q \frac{1 - \rho_{0k} \cos(2\pi f t_A - \alpha_{0k})}{1 - 2\rho_{0k} \cos(2\pi f t_A - \alpha_{0k}) + \rho_{0k}^2} + \sum_{k=1}^p \frac{1 - \rho_{\infty k} \cdot \cos(2\pi f t_A - \alpha_{\infty k})}{1 - 2\rho_{\infty k} \cos(2\pi f t_A - \alpha_{\infty k}) + \rho_{\infty k}^2}. \quad (4.16)$$

Anschauliche Interpretation

Der Einheitskreis bildet die $2\pi f t_A$ -Achse. Der Zähler in (4.14) beschreibt das Produkt aller Nullstellenabstände vom gegenwärtigen Punkt auf der $2\pi f t_A$ -Achse. Dies wird klar, wenn man das beliebige Dreieck in Abb. 4.2(a) näher betrachtet. Die Strecke ΔNS , die den Abstand der Nullstelle vom Punkt auf der $2\pi f t_A$ -Achse (Einheitskreis) beschreibt, errechnet sich nach dem Cosinussatz wie folgt [BS⁺12]:

$$(\Delta NS)^2 = 1^2 + \rho_0^2 - 2 \cdot 1 \cdot \rho_0 \cos(\delta) \quad (4.17)$$

$$= 1^2 + \rho_0^2 - 2 \cdot 1 \cdot \rho_0 \cos(\alpha - 2\pi f t_A) \quad (4.18)$$

$$= 1^2 + \rho_0^2 - 2 \cdot 1 \cdot \rho_0 \cos(-(2\pi f t_A - \alpha)), \quad (4.19)$$

$$\Delta NS = \sqrt{1 + \rho_0^2 - 2\rho_0 \cos(2\pi f t_A - \alpha)}. \quad (4.20)$$

Der Nenner von (4.14) beschreibt das Produkt der Polstellenabstände. Es wird nun klar, dass nahe am Einheitskreis liegende Polstellen ein kleines „Abstandsprodukt“ erzeugen und daher Amplitudenüberhöhungen im Frequenzgang bei den entsprechenden Frequenzen f auf der Frequenzachse, die nahe dieser Polstellen liegen, vorhanden sind.

Abbildung 4.2(b) soll den Phasengang (4.15) näher beleuchten. Der erste Summand $(p - q) 2\pi f t_A$ beschreibt die stationäre Phasenverschiebung für große Frequenzen. Der zweite Term $\arg\{b_0\}$ verschwindet für reelle Koeffizienten b_0 . Der Winkel γ_2 in Abb. 4.2(b) steht stellvertretend für den Winkel unter der zweiten Summe in (4.15). Er beschreibt den Winkel zwischen der reellen Achse und der Verbindungslinie von betrachtetem Pol zu gegenwärtigem Punkt auf der Frequenzachse. Entsprechende Überlegungen führen auf γ_1 , der den Winkel unter der ersten Summe in (4.15) repräsentiert. Die Differenz aus Pol- und Nullstellen-Winkelsummen geht also weiterhin in (4.15) ein.

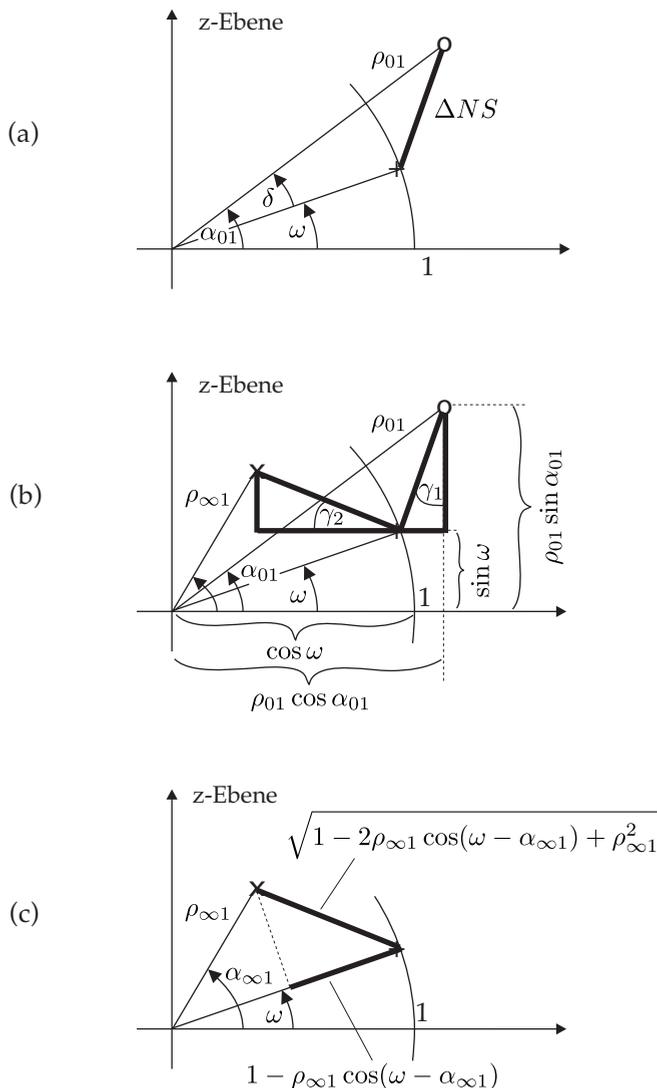


Abbildung 4.2. Veranschaulichung der Gleichungen (4.14) bis (4.16).

Abbildung 4.2(c) beschreibt den Zusammenhang (4.16), wobei hier die Anschauung schon schwerer fällt. Aus Übersichtlichkeitsgründen wurden die Überlegungen auch nur für eine Polstelle skizziert. Diese Ergebnisse sollen kurz zusammengefasst und ihre Bedeutung für die grobe Erstellung von Skizzen von Amplituden- und Phasengang dargestellt werden. Durch Abfahren des Einheitskreises (Frequenzachse) in mathematisch positiver Richtung beginnend bei $z = 1$ (entspricht der Frequenz $f = 0$ Hz) bis $z = -1$ ($f = f_A/2$) und Betrachten der Abstände der Pol- bzw. Nullstellen zu dem aktuell betrachteten Punkt auf dem Einheitskreis ist es also möglich,

den Frequenzgang des Systems bis auf einen konstanten Faktor im Amplitudengang bzw. einen konstanten Offset im Phasengang zu berechnen. Durch die positive Umlaufrichtung wird nur der positive Teil der Frequenzachse betrachtet. Da reale Systeme jedoch immer einen geraden Amplitudengang und einen ungeraden Phasengang besitzen (zur Erinnerung: in realen Systemen liegen Pole und Nullstellen entweder auf der reellen Achse oder kommen in einem konjugiert komplexen Paar vor, weshalb in der Impulsantwort keine komplexen Schwingungen vorhanden sind), ist bei diesen Systemen das Pol-Nullstellendiagramm immer symmetrisch zur reellen Achse. Dadurch charakterisiert diese Information das System vollständig und es braucht nur die halbe Frequenzachse von $z = 1$ bis $z = -1$ abgefahren werden.

Es ist so auch leicht ersichtlich, dass beispielsweise nahe am Einheitskreis liegende Pole Überhöhungen im Amplitudengang bewirken und dass nahe beisammen liegende Pole und Nullstellen steile Filterflanken hervorrufen können (vgl. das Notch-Filter in [PJ15]).

4.2.1.3 Grundbegriffe der Filterung

Ein Filter ist hier zu verstehen als *ein elektronisches System, das den Frequenzgehalt eines Signals modifiziert*. Mathematisch wird ein Filter durch eine Übertragungsfunktion nach (4.4) beschrieben. Der **Filterentwurf** umfasst dann die Wahl der Parameter b_j und a_i derart, dass bestimmte Eigenschaften bezüglich des Amplitudengangs und der Phase realisiert werden. Besteht der gewünschte Amplitudengang näherungsweise aus stückweise konstanten Abschnitten, so spricht man von **frequenzselektiven Filtern** – die Grundformen sind als **Tiefpass**, **Hochpass**, **Bandpass** und **Bandsperr** bekannt. Für den Entwurf dieser Filterformen existieren geschlossene Verfahren, z. B. solche, die sich vom Entwurf klassischer Analognetzwerke herleiten.

Idealerweise besitzen frequenzselektive Filter einen Amplitudengang, der im **Durchlassbereich** den Wert eins und im **Sperrbereich** den Wert null besitzt. Soll ein Filter als lineares, zeitinvariantes System realisiert werden, ist die Übertragungsfunktion $G(s)$ bzw. $G(z)$ wie schon erwähnt im Allgemeinen gebrochen rational. Oft kann damit der gewünschte Verlauf von Amplituden- und Phasengang aufgrund der begrenzten Zahl von Koeffizienten nur approximiert werden. Die Abweichungen vom gewünschten Verhalten werden in einem **Toleranzschema** zusammengestellt. Das Toleranzschema besteht aus einem definiertem **Übergangsbereich** zwischen Durchlass- und Sperrbereich, definiert durch die **Durchlassfrequenz** f_D und **Sperrfrequenz** f_S , sowie den maximal erlaubten Amplitudenabweichungen δ_D , δ_S in Durchlass- und Sperrbereich (siehe Abb. 4.3).

4.2.2 Anwendungsgebiete von Filtern

Filter kommen in den unterschiedlichsten Anwendungsgebieten zum Einsatz. Eine kleine Auswahl ausgewählter Themengebiete wird im Folgenden hier vorgestellt; die Aufzählung könnte jedoch beinahe beliebig fortgesetzt werden.

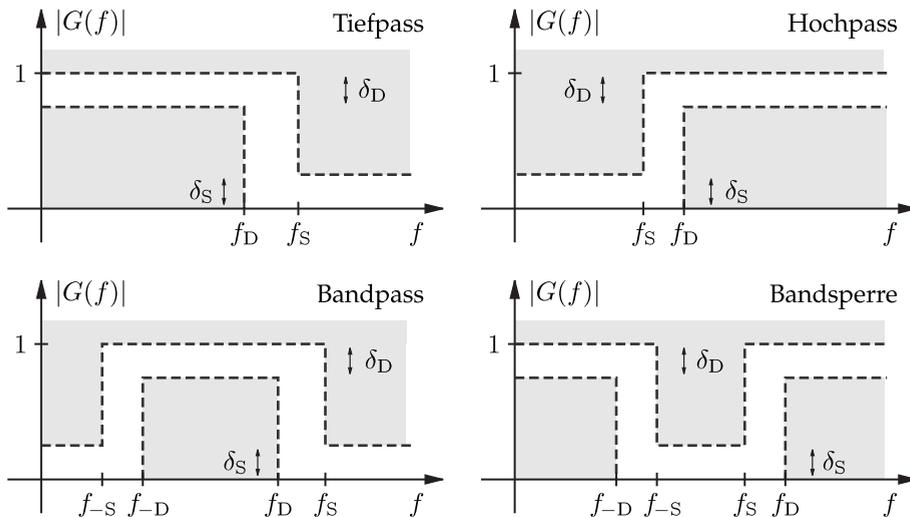


Abbildung 4.3. Toleranzschemata frequenzselektiver Filter. δ_D ist die erlaubte Amplitudenabweichung im Durchlassbereich und δ_S im Sperrbereich.

- **Nachrichtenübertragung:** Nachrichtenübertragung beschreibt den Austausch von Informationen. Unterschiedliche Übertragungsmedien haben unterschiedliche Charakteristika der Informations- oder Signalübertragung. Wegen externer Störquellen wie zum Beispiel dem Netzbrummen können frequenzselektive Störungen dem Nutzsignal während der Übertragung überlagert werden. Das Medium selbst besitzt auch eine bestimmte Bandbreite; normalerweise steigt die Dämpfung mit der Frequenz und somit fungiert das Medium selbst häufig wie ein Tiefpassfilter. Ein Filter auf der Empfängerseite kann anhand von bekannten Eigenschaften des Mediums Störungen beseitigen, um die Übertragung zu optimieren. Im Allgemeinen ist die Nutzung von Frequenzen gesetzlich geregelt. Es muss daher sichergestellt werden, dass ein Sender außerhalb des zugewiesenen Spektrums mit der Sendeleistung den gesetzlichen Grenzwert nicht überschreitet. Weiterhin muss in der Nachrichtenübertragung häufig auf die Phasenänderung über der Frequenz (Gruppenlaufzeit) geachtet werden, da eine Datenübertragung oft nur möglich ist, wenn die Phase des übertragenen Signals fast linear ist.
- **Audioverarbeitung:** In vielen HiFi-Geräten sind sogenannte Equalizer enthalten, welche eine Filterbank repräsentieren. Mit ihnen können bestimmte akustische Eigenschaften des beschallten Raumes erreicht, das heißt, ein gewünschter Frequenzgang realisiert werden. In der Audioverarbeitung ist eine Linearität der Phase der eingesetzten Filter besonders wichtig, da das menschliche Ohr empfindlich auf nicht konstante Gruppenlaufzeiten reagiert. Hier ist deshalb oft von einer *Phasenverzerrung* die Rede.

- **Bildverarbeitung:** Effekte, die Bilder schärfen oder unschärfer machen (*blur*), werden mit Hilfe zweidimensionaler Filter realisiert. Bei der Bildverarbeitung sieht man Verzerrungen besonders deutlich, wenn Bilder verkleinert oder vergrößert werden sollen. Ein entsprechendes Hoch- oder Tiefpassfilter kann diese Verzerrungen unterdrücken.

Je nach Einsatzzweck müssen Filter also bestimmte Eigenschaften in Bezug auf Phase, Gruppenlaufzeit oder Amplitudengang aufweisen.

4.2.3 Analoge Filter

Analoge Filter realisieren eine Approximation eines gewünschten Frequenzgangs $G(f)$ mit analogen Bauteilen. Die Bauteile können entweder *passiv* oder *aktiv* sein, man spricht dann von einem passiven oder aktiven Filter. Analoge Filter sind Schaltungen, die sich mittels einfacher Methoden wie z. B. Bode-Diagrammen [PJ15] und der Laplace-Transformation analysieren lassen. Im Folgenden werden einige analoge Tiefpassfilter beschrieben.

4.2.3.1 Butterworth-Filter

Der Frequenzgang eines **Butterworth-Filters** ist im Durchlassbereich maximal flach. Für ein Tiefpassfilter N -ter Ordnung bedeutet das, dass die ersten $(2N - 1)$ Ableitungen vom Quadrat des Frequenzgangs bei $f = 0$ null sind. Das Butterworth-Filter ist sowohl im Durchlass- als auch im Sperrbereich monoton fallend. Das Quadrat des Amplitudengangs wird verwendet, da es die Leistungsverteilung über der Frequenz repräsentiert. Für ein Butterworth-Tiefpassfilter ist es wie folgt definiert:

$$|G(f)|^2 = \frac{1}{1 + \varepsilon^2 (2\pi f')^{2N}} \quad (4.21)$$

mit der normierten Frequenz $f' = f/f_D$ und $\varepsilon > 0$. Mit höherer Ordnung N wird der Übergang vom Durchlass- zum Sperrbereich steiler. Für den Entwurf eines Butterworth-Filters hat man die folgenden Parameter zur Verfügung: f_D , f_S , δ_D und δ_S (siehe Abb. 4.3). Mit der folgenden Gleichung kann die benötigte Ordnung berechnet werden [KK12, PJ15]:

$$N = \frac{\log \left(\frac{(\frac{1}{\delta_S})^2 - 1}{(\frac{1}{1 - \delta_D})^2 - 1} \right)}{2 \log \left(\frac{f_S}{f_D} \right)}. \quad (4.22)$$

Die Polstellen können mittels

$$s_{\infty k} = \varepsilon^{-\frac{1}{N}} \cdot e^{\left(\frac{j\pi}{2N}(2k+N-1)\right)}, \quad k = 1, \dots, 2N, \quad (4.23)$$

aus dem Betragsquadrat des Amplitudengangs (4.21) berechnet werden. Um ein kausales und stabiles Filter zu erhalten, müssen nun nur noch diejenigen Pole gewählt

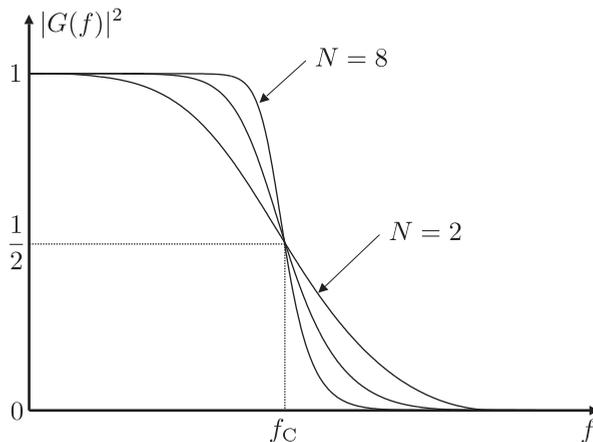


Abbildung 4.4. Amplitudengang des Butterworth-Tiefpassfilters. f_C ist die Knickfrequenz (Abschn. 4.2.4.1).

werden, die in der linken s -Halbebene liegen. Damit erspart man sich das aufwändige Aufspalten des Betragsquadrates $|G(s)|^2 = G(-s) \cdot G(s)$, um die Pole der (stabilen) Übertragungsfunktion $G(s)$ zu erhalten. Diese liegen auf einem Halbkreis in der s -Ebene.

4.2.3.2 Tschebyscheff-Filter

Beim Butterworth-Filter ist die maximale Amplitudenabweichung im Durchlass- oder Sperrbereich in nur einem Punkt gegeben, nämlich im Idealfall bei den Frequenzen f_S oder f_D . Wenn im Durchlassbereich mehrere maximale Amplitudenabweichungen erlaubt sind (vgl. Abb. 4.5), erhält man ein **Tschebyscheff-Filter**. Auf Grund der Aufteilung des Fehlers auf die Punkte maximaler Amplitudenabweichung ergibt das Tschebyscheff-Filter einen steileren Übergang zwischen Durchlass- und Sperrbereich.

Wenn der Frequenzgang Schwingungen im Durchlassbereich hat und monoton im Sperrbereich ist, handelt es sich dabei um ein Tschebyscheff-I-Filter. Der Frequenzgang von Tschebyscheff-II-Filtern ist monoton im Durchlassbereich und weist Schwingungen im Sperrbereich auf. Das Quadrat des Frequenzgangs für ein Tschebyscheff I-Tiefpassfilter lautet

$$|G(f)|^2 = \frac{1}{1 + \varepsilon^2 V_N^2(f')} \tag{4.24}$$

Dabei ist $V_N(x)$ das sogenannte **Tschebyscheff-Polynom**:

$$V_N(x) = \begin{cases} \cos(N \arccos(f')) & \text{für } |f'| \leq 1, \\ \cosh(N \operatorname{arccosh}(f')) & \text{für } |f'| > 1. \end{cases} \tag{4.25}$$

Die Schwingung, im Durchlassbereich begrenzt durch δ_D , schwingt zwischen 1 und $1/(1 + \varepsilon^2)$.

Die Pole eines Tschebyscheff-Filters liegen in der s -Ebene auf einer Ellipse. Die kleine und große Halbachse dieser Ellipse sind nur abhängig von der Höhe des Durchlasstoleranzbandes ε und der Filterordnung N . Die geometrische Bestimmung kann [OSB04] entnommen werden.

Das Quadrat des Frequenzgangs eines Tschebyscheff-II-Filters erhält man mit

$$|G(f)|^2 = \frac{1}{1 + \varepsilon^2 \frac{V_N^2(f'_S)}{V_N^2(f'_S/f')}}. \quad (4.26)$$

Hierbei gilt analog zur Definition von f' (s. o.) $f'_S = f_S/f_D$.

Das Typ-II-Filter erzeugt im Übergangsbereich zwischen Durchlass- und Sperrbereich die geringste Laufzeit und besitzt im Durchlassbereich das größte Intervall mit konstanter Gruppenlaufzeit.

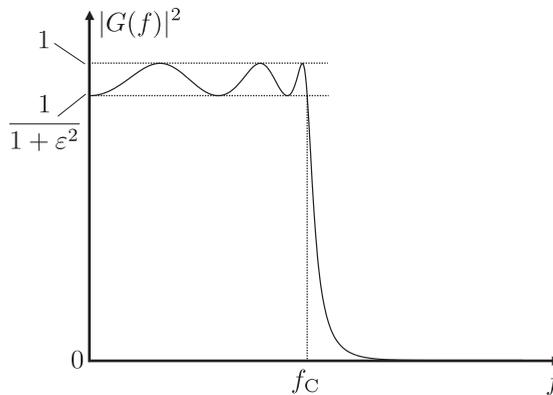


Abbildung 4.5. Amplitudengang des Tschebyscheff-I-Tiefpassfilters.

4.2.3.3 Cauer-Filter

Die elliptischen Filter oder **Cauer-Filter** stellen eine Kombination aus Tschebyscheff-Filtern der Typen I und II dar, das heißt, sie sind sowohl im Durchlass- als auch im Sperrbereich wellig. Das Betragsquadrat des Frequenzgangs eines Cauer-Filters lautet:

$$|G(f)|^2 = \frac{1}{1 + \varepsilon^2 U_N^2(f')}. \quad (4.27)$$

Die Funktion $U_N(f)$ ist dabei die sogenannte **Jacobi'sche Ellipsenfunktion**. Daher rührt auch der Name *elliptische Filter*.

Cauer-Filter zeichnen sich durch eine sehr große Flankensteilheit aus und benötigen eine weit geringere Ordnung als die anderen bereits erwähnten Filter, um ein vorgegebenes Toleranzband einzuhalten. Die steile Flanke des Cauer-Filters erkauft man sich allerdings durch eine extreme Phasenverzerrung. Insbesondere im Durchlassbe-

reich tritt ein Peak in der Gruppenlaufzeit auf, weshalb solche Filter zum Beispiel für Audioanwendungen ungeeignet sind. Auch für die Impulswiedergabe sind Cauer-Filter schlecht verwendbar.

4.2.4 Digitale Filter

Wie bereits erwähnt, bieten digitale Filter gegenüber einer analogen Realisierung entscheidende Vorteile wie einfache Parametrierbarkeit und eine geringere Abhängigkeit von Bauteiltoleranzen. Wo möglich, wird daher die digitale Filterung der analogen vorgezogen. Analoge Filter finden sich heute hauptsächlich in zwei großen Anwendungsgebieten, die der digitalen Filterung nicht zugänglich sind. Dies ist zum einen die Vorverarbeitung vor der Digitalisierung (Abtastung) von Messdaten, zum anderen der Bereich der Hochfrequenz, da die maximale Frequenz, bei der digitale Filter in Echtzeit eingesetzt werden können, durch die Schaltgeschwindigkeit von Transistoren beschränkt ist.

In Abschn. 4.2.1.1 wurde bereits gezeigt, dass durch die Abtastung im Zeitbereich im Frequenzbereich eine periodische Wiederholung mit der Abtastfrequenz entsteht. Daraus resultiert eine Redundanz im Frequenzgang, die nicht erwünscht ist. Bei digitalen Filtern wird deshalb oft die auf die Abtastfrequenz bezogene Frequenz $\omega = 2\pi f t_A$ als Variable verwendet. Für die Analyse realer Filter reicht die Betrachtung im Bereich $0 \leq \omega \leq \pi$ aus.

In diesem Abschnitt werden die zwei Arten von digitalen Filtern vorgestellt sowie jeweils anschließend der Entwurf von Filtern mit gewünschtem Übertragungsverhalten diskutiert.

4.2.4.1 FIR-Filter

Generell kann man digitale Filter in zwei Hauptklassen einteilen, je nachdem, ob Ausgangswerte zurückgekoppelt werden oder nicht. Ist eine Rückkopplung vorhanden, spricht man von *rekursiven* Filtern, sonst von *nichtrekursiven* Filtern. In diesem Kapitel soll auf die nichtrekursiven oder FIR-Filter eingegangen werden.

Algorithmus

Nichtrekursive Filter verwenden ausschließlich *Eingangswerte* zur Berechnung der Ausgangswerte, siehe auch Abb. 4.6. Dies führt zu folgender Impulsantwort:

$$g_n = \sum_{k=0}^m b_k \delta_{n-k}. \quad (4.28)$$

Die Impulsantwort des nichtrekursiven Systems weist nur in den ersten m Schritten ein von null verschiedenes Ausgangssignal auf. Da nach $m + 1$ Takten die Impulsantwort verschwindet, wird dieser Filtertyp auch FIR-Filter genannt („finite impulse response“).

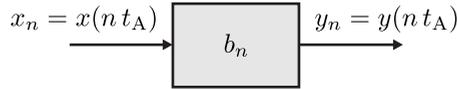


Abbildung 4.6. Blockschaltbild eines nichtrekursiven digitalen Filters.

Übertragungsfunktion

Die Abtastwerte der Impulsantwort eines FIR-Filters entsprechen gerade dessen Filterkoeffizienten:

$$g_n = \begin{cases} b_n & \text{für } 0 \leq n \leq m \\ 0 & \text{sonst.} \end{cases} \quad (4.29)$$

Für die *nicht-symmetrische* Form der Übertragungsfunktion gilt dann

$$G(f) = \sum_{k=0}^m b_k e^{-j2\pi f t_A k} \quad (4.30)$$

und für die *symmetrische* Darstellung für gerade m

$$G(f) = \sum_{k=-\frac{m}{2}}^{\frac{m}{2}-1} b_k e^{-j2\pi f t_A k} . \quad (4.31)$$

Die Darstellungsformen können im Allgemeinen durch eine Zeitverschiebung der Impulsantwort ineinander überführt werden.

Entwurf nichtrekursiver Filter

Der Filterentwurf nichtrekursiver Systeme besteht in der Vorgabe eines gewünschten Frequenzgangs $G_W(f)$. Die Impulsantwort des gesuchten Filters resultiert aus der inversen zeitdiskreten Fourier-Transformation:

$$g_n = t_A \int_{-\frac{f_A}{2}}^{\frac{f_A}{2}} G_W(f) \cdot e^{j2\pi f t_A n} df . \quad (4.32)$$

Anschaulich bedeutet dies also, dass man beispielsweise beim Entwurf eines Tiefpassfilters dessen Frequenzgang vorgibt und davon die *inverse* Fourier-Transformierte bildet, um die gesuchten Filterkoeffizienten zu erhalten. Somit erhält man die folgende Approximation für die Impulsantwort des gesuchten idealen Tiefpassfilters [PJ15]:

$$g_n = \begin{cases} \frac{2\pi f_C t_A}{\pi} \frac{\sin(2\pi f_C t_A (n - \frac{m}{2}))}{2\pi f_C t_A \cdot (n - \frac{m}{2})} & \text{für } 0 \leq n \leq m \\ 0 & \text{sonst.} \end{cases} \quad (4.33)$$

Die Frequenz f_C stellt hierbei die **Grenzfrequenz** oder auch **Knickfrequenz** dar, also die Frequenz, bei der die Amplitude im Vergleich zur Maximalamplitude um 3 dB

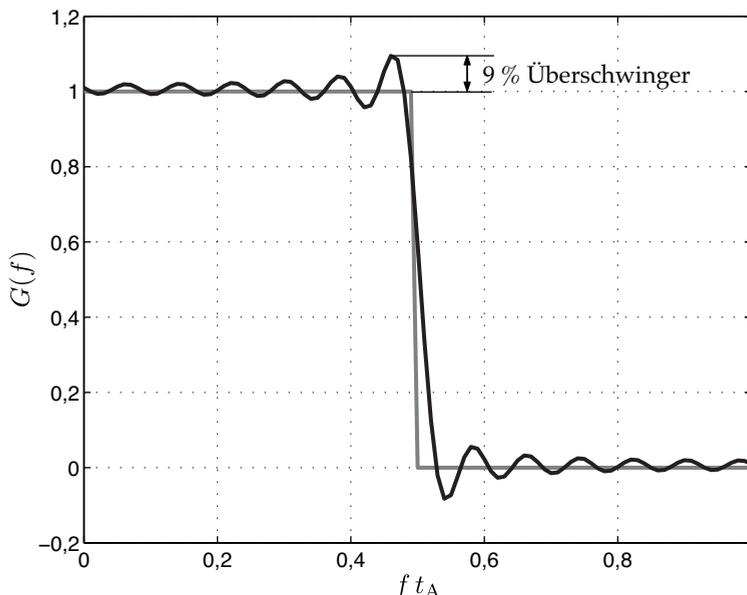


Abbildung 4.7. Fourier-Approximation eines idealen Tiefpassfilters.

gedämpft wird. Durch die endliche Länge der Filterantwort kommt es zu dem in Abb. 4.7 dargestellten Verlauf des Frequenzgangs. Sowohl im Durchlass- als auch im Sperrbereich kommt es zu Oszillationen, deren Höhe unabhängig von der Filterlänge ist (**Gibbs'sches Phänomen**). Allerdings steigt die Oszillationsfrequenz mit steigender Filterlänge. Die endliche Impulsantwort kann auch dargestellt werden als

$$g_n = \frac{2\pi f_C t_A}{\pi} \frac{\sin(2\pi f_C t_A (n - \frac{m}{2}))}{2\pi f_C t_A (n - \frac{m}{2})} \cdot r_m \left[n - \frac{m}{2} \right] \quad (4.34)$$

mit dem zeitdiskreten Rechteckfenster

$$r_m[n] = \begin{cases} 1 & \text{für } -\frac{m}{2} \leq n \leq \frac{m}{2} \\ 0 & \text{sonst.} \end{cases} \quad (4.35)$$

Um das Übertragungsverhalten des Filters in Abb. 4.7 zu verbessern, kann man sich anderer Fensterfunktionen (z. B. Hann-, Hamming-, Kaiser-Fenster usw.) bedienen. Allerdings nimmt man dadurch Verschlechterungen der Filtereigenschaften wie z. B. eine Verbreiterung des Hauptmaximums im Frequenzbereich in Kauf, siehe auch [KK12].

4.2.4.2 IIR-Filter

Die zweite Gruppe von zeitdiskreten Filtern neben den FIR-Filtern sind die IIR (*infinite impulse response*)-Filter. Sie verwenden im Gegensatz zu FIR-Filtern neben Eingangs-

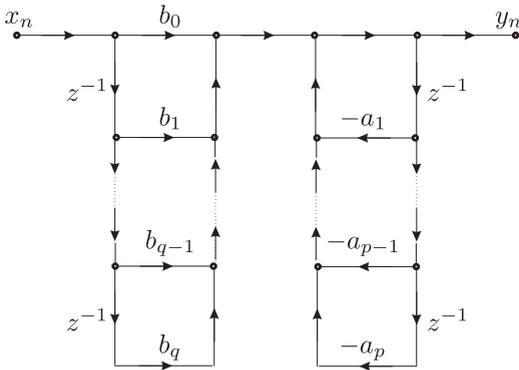


Abbildung 4.8. Realisierung eines IIR-Filters in direkter Form I.

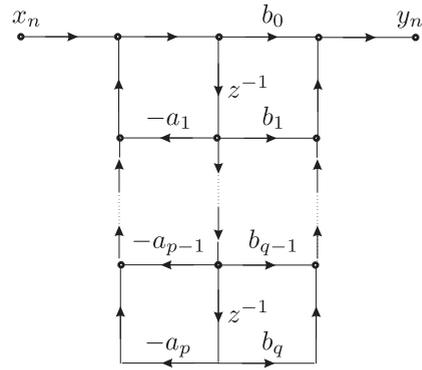


Abbildung 4.9. IIR-Filter in direkter Form II mit $p = q$.

werten auch gefilterte Ausgangswerte zur Berechnung des aktuellen Ausgangswertes. Die Umsetzung der Differenzgleichungen von IIR-Filtern ist mit verschiedenen Strukturen möglich. Auf die am häufigsten verwendeten dieser Strukturen wird hier eingegangen und auf ihre Vorteile hinsichtlich Speicherbedarf, Robustheit usw. hingewiesen.

Die vier Hauptstrukturen digitaler IIR-Filter

Der Eingang x_n und der Ausgang y_n eines kausalen IIR-Filters mit der Übertragungsfunktion

$$G(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^q b_k z^{-k}}{1 + \sum_{k=1}^p a_k z^{-k}} \tag{4.36}$$

hat die folgende Differenzgleichung:

$$y_n = \sum_{k=0}^q b_k x_{n-k} - \sum_{k=1}^p a_k y_{n-k} . \tag{4.37}$$

Wird die Differenzgleichung mit der Hilfsgröße w_n wie folgt geschrieben:

$$w_n = \sum_{k=0}^q b_k x_{n-k} , \tag{4.38}$$

$$y_n = w_n - \sum_{k=1}^p a_k y_{n-k} , \tag{4.39}$$

so handelt es sich um eine serielle (Kaskaden-)Schaltung von zwei Teilen:

$$Y(z) = \frac{1}{A(z)} \left(B(z) X(z) \right) \quad (4.40)$$

und wird *Struktur in direkter Form I* genannt [OSB04]. Die Rechenintensität und der Speicherbedarf dieser Struktur sind folgendermaßen:

- Anzahl Multiplikationen pro Element (Sample): $p + q + 1$,
- Anzahl Additionen pro Element: $p + q$,
- Anzahl Verzögerungsglieder: $p + q$.

Wenn die Reihenschaltung von $1/A(z)$ und $B(z)$ vertauscht ist, folgt

$$Y(z) = B(z) \left(\frac{1}{A(z)} X(z) \right) \quad (4.41)$$

und die Differenzgleichungen lauten:

$$w_n = x_n - \sum_{k=1}^p a_k w_{n-k}, \quad (4.42)$$

$$y_n = \sum_{k=0}^q b_k w_{n-k}. \quad (4.43)$$

Diese Schaltung wird *Direkte Form II* genannt und hat den Vorteil, dass die Verzögerungsglieder kombiniert werden können, wie in Abb. 4.9 dargestellt. Dabei wird der Speicherbedarf reduziert. Die Rechenintensität und der Speicherbedarf dieser Struktur ergeben sich zu:

- Anzahl Multiplikationen pro Element (Sample): $p + q + 1$,
- Anzahl Additionen pro Element: $p + q$,
- Anzahl Verzögerungsglieder: $\max(p, q)$.

Eine Alternative zu den einfachen Strukturen in Direktform I und II ist die Reihenschaltung von Teilfiltern zweiter Ordnung. Dies erfolgt über die Faktorisierung der Übertragungsfunktion $G(z)$:

$$G(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^q b_k z^{-k}}{1 + \sum_{k=1}^p a_k z^{-k}} = \prod_{n=1}^{N_S} \frac{b_{0,n} + b_{1,n} z^{-1} + b_{2,n} z^{-2}}{1 + a_{1,n} z^{-1} + a_{2,n} z^{-2}}. \quad (4.44)$$

Dabei ist $N_S = \lfloor (N + 1)/2 \rfloor$ die Zahl $(N + 1)/2$ nach unten gerundet. Die Teilfilter können in der Direktform II realisiert werden, um den benötigten Speicherplatz zu minimieren.

Alternativ zur Reihenschaltung ist darüber hinaus die Parallelschaltung von Systemen erster und zweiter Ordnung denkbar. Die z -Transformierten der einzelnen

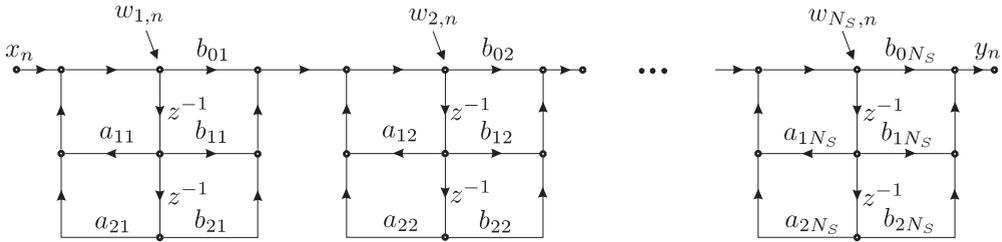


Abbildung 4.10. Reihenschaltung von Teilfiltern 2. Ordnung in direkter Form II.

Subsysteme werden in diesem Fall nicht wie bei der Serienschaltung multipliziert, sondern addiert. Man erhält die Übertragungsfunktion der einzelnen Subsysteme durch Partialbruchzerlegung der Gesamtübertragungsfunktion (siehe Abb. 4.10).

Die Freiheitsgrade beim Entwurf solcher Filterstrukturen werden durch verschiedene Kombinationen von Null- und Polstellen sowie durch die Reihenfolge der resultierenden Teilstrukturen repräsentiert. Untersuchungen haben gezeigt, dass mittels gewisser Entwurfsrichtlinien diese Struktur hinsichtlich Quantisierungsrauschen optimiert werden kann (siehe Abschn. 4.2.5).

Entwurf rekursiver Tiefpassfilter

Ziel des Filterentwurfs ist die Aufstellung der Übertragungsfunktion

$$G(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^q b_k z^{-k}}{1 + \sum_{k=1}^p a_k z^{-k}}. \quad (4.45)$$

Der Entwurf eines zeitdiskreten IIR-Filters basiert auf der Transformation eines zeitkontinuierlichen Systems mit dem gewünschten Frequenzgang in die z -Ebene. Dies geschieht typischerweise mittels der *bilinearen Transformation* [PJ15]:

$$s = \frac{2}{t_A} \cdot \frac{z-1}{z+1}. \quad (4.46)$$

Die Abbildung bildet die linke Hälfte der s -Ebene in das Innere des Einheitskreises ab, die imaginäre Achse wird auf den Einheitskreis transformiert:

$$\pi f = \frac{1}{t_A} \tan(\pi f t_A) = \frac{1}{t_A} \tan\left(\frac{\omega}{2}\right). \quad (4.47)$$

Es kommt also durch (4.47) zu einer nichtlinearen Vorverzerrung der Frequenzachse.

Gleichung (4.47) bildet den Bereich $-\infty \leq f \leq \infty$ der s -Ebene auf den Bereich $-\pi \leq \omega \leq \pi$ der z -Ebene ab. Die kritischen Frequenzen im Toleranzschema des zeitkontinuierlichen Systems müssen gemäß (4.47) vorverzerrt werden; dann nur erfüllt

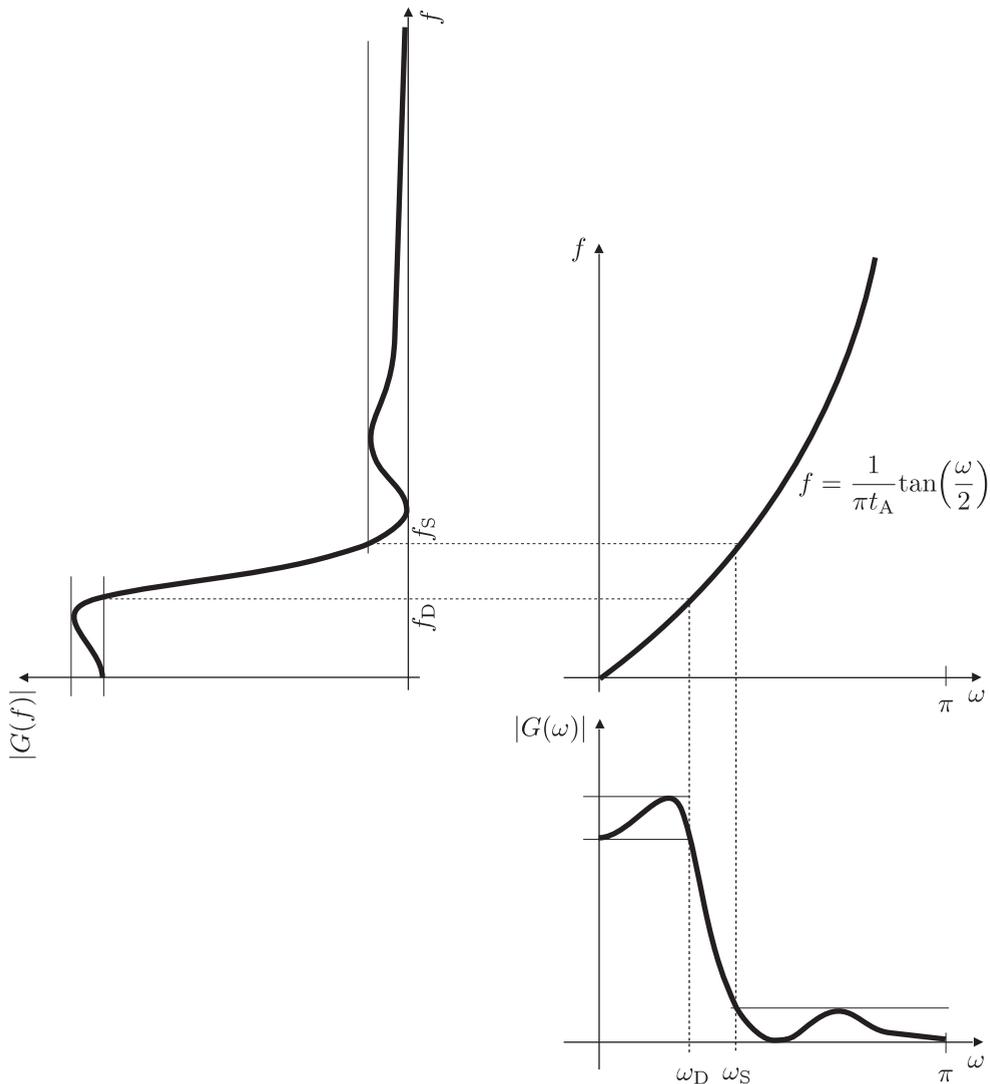


Abbildung 4.11. Frequenzverzerrung aufgrund der Anwendung der bilinearen Transformation.

das nach Transformation mit (4.46) erhaltene zeitdiskrete Filter die Spezifikation. Den Sachverhalt der Verzerrung veranschaulicht Abb. 4.11.

Ein weiterer Effekt der bilinearen Transformation betrifft die Phase. Durch die nichtlineare Abbildung der Frequenzachse kann beispielsweise der lineare Phasenverlauf eines zeitkontinuierlichen Filters für das korrespondierende zeitdiskrete Filter verloren gehen. Dies muss beim Filterentwurf unbedingt berücksichtigt werden.

Die Vorgehensweise beim Entwurf eines digitalen rekursiven Filters kann am einfachsten anhand eines Beispiels beschrieben werden.

Entwurf eines rekursiven digitalen Tiefpassfilters aus einem analogen Butterworth-Filter

In der folgenden Anleitung zum Entwurf eines Tiefpassfilters werden zunächst die Schritte genannt und anschließend Zahlenwerte vorgegeben. Diese sind willkürlich gewählt und sollen ausschließlich der Veranschaulichung dienen.

a) Durchlass- und Sperrbereich im Zeitdiskreten definieren (vgl. Abschn. 4.2.1.3), d. h. ω_D , ω_S , δ_D und δ_S festlegen:

$$\text{Durchlassbereich: } 0,89125 \leq |G_Z(\omega)| \leq 1, \quad 0 \leq \omega \leq 0,2\pi,$$

$$\text{Sperrbereich: } 0 \leq |G_Z(\omega)| \leq 0,17783, \quad 0,3\pi \leq \omega \leq \pi.$$

b) Abbildung der zeitdiskreten Grenzfrequenzen auf die entsprechenden zeitkontinuierlichen durch Vorverzerrung gemäß (4.47):

$$\text{Durchlassbereich: } 0,89125 \leq |G_S(f)| \leq 1, \quad 0 \leq f \leq \frac{1}{\pi t_A} \tan\left(\frac{0,2\pi}{2}\right),$$

$$\text{Sperrbereich: } 0 \leq |G_S(f)| \leq 0,17783, \quad \frac{1}{\pi t_A} \tan\left(\frac{0,3\pi}{2}\right) \leq f < \infty.$$

c) Einsetzen der Grenzfrequenzen in das Betragsquadrat des Frequenzgangs: Butterworth-Filter nach Abschn. 4.2.3.1, Annahme: $t_A = 1$:

$$|G_S(f)|^2 = \frac{1}{1 + (f/f_C)^{2N}}; \quad (4.48)$$

$$1 + \left(\frac{f_D}{f_C}\right)^{2N} \stackrel{!}{=} \left(\frac{1}{1 - \delta_D}\right)^2 \approx \left(\frac{1}{0,89}\right)^2 \Rightarrow 1 + \left(\frac{2 \tan(0,1\pi)}{2\pi f_C}\right)^{2N} \stackrel{!}{=} \left(\frac{1}{0,89}\right)^2, \quad (4.49)$$

$$1 + \left(\frac{f_S}{f_C}\right)^{2N} \stackrel{!}{=} \left(\frac{1}{\delta_S}\right)^2 \approx \left(\frac{1}{0,18}\right)^2 \Rightarrow 1 + \left(\frac{2 \tan(0,15\pi)}{2\pi f_C}\right)^{2N} \stackrel{!}{=} \left(\frac{1}{0,18}\right)^2. \quad (4.50)$$

d) Auflösen der beiden Gleichungen (4.49) und (4.50) liefert neben der (zeitkontinuierlichen) Grenzfrequenz f_C die Filterordnung N (siehe (4.22)):

$$\text{Ergebnis: } N = 5,26 \Rightarrow N = 6.$$

e) Anpassung von f_C mittels (4.50) wegen des Aufrundens in Schritt d):

$$\Rightarrow f_C = \frac{0,77}{2\pi} = 0,123.$$

f) Bestimmung der Pole des Butterworth-Filters nach (4.23):

$$s_{\infty k} = 2\pi f_C \cdot e^{(j \frac{\pi}{2N} (2k + N - 1))} \quad (4.51)$$

\Rightarrow Auswahl der Pole der linken Halbebene liefert Pole der Systemfunktion.

g) Aufstellen der zeitkontinuierlichen Systemfunktion mittels der Pole:

$$G_S(s) = \frac{0,20}{(s^2 + 0,40s + 0,59)(s^2 + 1,08s + 0,59)(s^2 + 1,48s + 0,59)}. \quad (4.52)$$

h) Anwendung der bilinearen Transformation (4.46) unter der Annahme $t_A = 1$ liefert schließlich die zeitdiskrete Übertragungsfunktion und damit die Pole in der z -Ebene:

$$G_Z(z) = \frac{7,378 \cdot 10^{-4} (1 + z^{-1})^6}{(1 - 1,27z^{-1} + 0,71z^{-2}) (1 - 1,01z^{-1} + 0,36z^{-2}) (1 - 0,90z^{-1} + 0,22z^{-2})} \cdot \quad (4.53)$$

Der digitale Filterentwurf für die anderen in Abschn. 4.2.3 beschriebenen analogen Filter vom Typ Tschebyscheff und Cauer verläuft auf entsprechende Art und Weise, resultiert jedoch in einem weitaus komplizierteren Rechengang.

Frequenztransformation von IIR-Tiefpassfiltern

Der oben beschriebene Weg zum Entwurf eines zeitdiskreten Tiefpassfilters im Zeitkontinuierlichen unter Anwendung der bilinearen Transformation kann auch für die anderen drei Filter-Grundtypen Hochpass, Bandpass und Bandsperr durchgeföhrt werden. Wünschenswert wäre jedoch ein einheitlicher Entwurfsprozess für alle Filter. Aus diesem Grund werden ausgehend von sogenannten *Tiefpassgrundfiltern* durch Anwendung spezieller Transformationen die gewünschten frequenzselektiven Filter gewonnen. Dies kann durchaus auch ein Tiefpassfilter mit einer anderen Grenzfrequenz sein.

Es wird eine Abbildung gesucht, welche die Variable Z^{-1} des Grundfilters in geeigneter Weise in die z -Ebene des zu entwerfenden Filters transformiert, sodass also gilt:

$$H(z) = H(Z) \Big|_{Z^{-1} = G(z^{-1})} \cdot \quad (4.54)$$

Um ein rationales, kausales und stabiles Grundsystem $H(Z)$ in ein Zielsystem $H(z)$ mit denselben Eigenschaften zu transformieren, müssen für $Z^{-1} = G(z^{-1})$ folgende Bedingungen erfüllt sein:

- $G(z^{-1})$ muss eine rationale Funktion in z^{-1} sein.
- Das Innere des Einheitskreises der Z -Ebene muss auch in das Innere des Einheitskreises in der z -Ebene abgebildet werden.
- Der Einheitskreis der Z -Ebene muss auf den Einheitskreis in der z -Ebene abgebildet werden.

Auf Basis dieser Kriterien kann eine allgemeine Beziehung für $G(z^{-1})$ hergeleitet werden (siehe auch [OSB04]).

Die Transformationsbedingungen werden schnell sehr kompliziert und können detailliert den einschlägigen Lehrbüchern [PJ15, OSB04] entnommen werden. An dieser Stelle sollen daher nur die Tiefpass-Tiefpass-Transformation sowie die Tiefpass-Hochpass-Transformation Erwähnung finden. Für erstere ergibt sich die Transforma-

tionsvorschrift

$$Z^{-1} = \frac{z^{-1} - \alpha_{\text{TP}}}{1 - \alpha_{\text{TP}} z^{-1}} \quad (4.55)$$

mit

$$\alpha_{\text{TP}} = \frac{\sin\left(\frac{\omega'_P - \omega_P}{2}\right)}{\sin\left(\frac{\omega'_P + \omega_P}{2}\right)}. \quad (4.56)$$

Hierbei stellt ω'_P die Grenzfrequenz des bereits bekannten Tiefpassfilters dar und ω_P repräsentiert die Grenzfrequenz des gesuchten Tiefpasses.

Für die Tiefpass-Hochpass-Transformation gilt folgende Abbildungsvorschrift:

$$Z^{-1} = \frac{z^{-1} + \alpha_{\text{HP}}}{1 + \alpha_{\text{HP}} z^{-1}} \quad (4.57)$$

mit

$$\alpha_{\text{HP}} = \frac{\cos\left(\frac{\omega'_P + \omega_P}{2}\right)}{\cos\left(\frac{\omega'_P - \omega_P}{2}\right)}. \quad (4.58)$$

Die Entwurfsformeln setzen voraus, dass sämtliche Grenzfrequenzen im Intervall $0 \leq \omega \leq \pi$ liegen.

Der Entwurf für Bandpässe und Bandsperren erfolgt analog, jedoch mit komplexeren Transformationen. Somit ist der Grundstein für den IIR-Filterentwurf gelegt.

4.2.5 Auswirkungen endlicher Rechengenauigkeit

4.2.5.1 Quantisierung der Koeffizienten

Durch Quantisierung des Wertebereichs kann in digitalen Systemen keine infinitesimale Genauigkeit erzielt werden. Das bedeutet, dass die Koeffizienten einer Differenzgleichung bzw. einer Übertragungsfunktion nur endlich genau vorliegen. Das kann bedeuten, dass Pole und Nullstellen in der z -Ebene verschoben werden und dass damit zwangsläufig eine Abweichung zwischen realisierbarer zu gewünschter Übertragungsfunktion auftritt. Im schlimmsten Fall kann daraus die Instabilität eines Systems resultieren, wenn die Pole durch Quantisierung den Einheitskreis verlassen.

Die realisierbare Übertragungsfunktion eines quantisierten Systems ergibt sich zu

$$\hat{G}(z) = \frac{\sum_{k=0}^q \hat{b}_k z^{-k}}{1 + \sum_{k=1}^p \hat{a}_k z^{-k}} \quad (4.59)$$

mit den quantisierten Koeffizienten $\hat{a}_k = a_k + \Delta a_k$ und $\hat{b}_k = b_k + \Delta b_k$. Die Delta-Größen repräsentieren hier den Quantisierungsfehler, das heißt die Abweichung vom idealen Koeffizienten.

Der Quantisierungsfehler *eines* Koeffizienten beeinflusst *alle* Pole $z_{\infty i}$, $i = 1, \dots, p$, der Systemfunktion. Ersichtlich wird diese Tatsache bei Ableitung eines Poles nach dem quantisierten Parameter, vgl. [OSB04]:

$$\frac{\partial z_{\infty i}}{\partial a_k} = \frac{z_{\infty i}^{p-k}}{\prod_{j=1, j \neq i}^p (z_{\infty i} - z_{\infty j})}. \quad (4.60)$$

Werden nun alle p Koeffizienten durchlaufen, um die resultierende Polverschiebung $\Delta z_{\infty i}$ zu erhalten, so folgt:

$$\Delta z_{\infty i} = \sum_{k=1}^p \frac{\partial z_{\infty i}}{\partial a_k} \Delta a_k. \quad (4.61)$$

Entsprechend wird durch Quantisierung der Koeffizienten natürlich auch der Frequenzgang des Filters beeinträchtigt.

In den 1960er-Jahren wurde gezeigt, dass bei Strukturen der direkten Form mit eng beisammen liegenden Polen oder Nullstellen selbst bei kleinen Ungenauigkeiten der Koeffizienten des Nenners bzw. Zählers große Verschiebungen der Pole und Nullstellen verursacht werden können.

Insbesondere schmalbandige Filter reagieren daher sehr empfindlich auf die Quantisierung der Koeffizienten. Für diese wird das Produkt der Abstände im Nenner von (4.60) sehr klein und folglich ist die Empfindlichkeit des i -ten Pols groß gegenüber einer Variation des k -ten Koeffizienten. Gleiches gilt entsprechend für die Nullstellen.

Da bei der Reihenschaltung von Systemen 2. Ordnung die konjugiert komplexen Polpaare und Nullstellen der einzelnen Teilsysteme unabhängig voneinander vorgegeben werden, wirken sich Quantisierungsfehler der Koeffizienten nicht auf *alle* Pole und Nullstellen des Systems aus. Somit können Pole und Nullstellen verschiedener Teilsysteme eng beieinander liegen, ohne sich gegenseitig zu beeinflussen. Die Struktur der Reihenschaltung ist somit sehr viel robuster gegenüber einer Variation der Koeffizienten ihrer Übertragungsfunktion.

Die Parallelschaltung ist hinsichtlich der Robustheit zwar der direkten Form vorzuziehen, ist jedoch aufgrund des Nullstellen-Entwurfs schlechter geeignet als die Reihenschaltung.

Die Standardausführungen der direkten Formen reagieren also anfälliger auf Quantisierungseffekte. Weil die modularen Strukturen hinsichtlich des Speicher- und Schaltungsaufwandes nur unwesentlich aufwändiger sind, werden die direkten Formen beinahe ausschließlich für die Implementierung der Teilmodelle 2. Ordnung ver-

wendet, welche dann ihrerseits in den Reihen- und Parallelschaltungen verwendet werden.

Eine Verbesserung der Quantisierungsrobustheit direkter Formen kann durch sogenannte gekoppelte Strukturen erreicht werden [KK12].

4.2.5.2 Quantisierungsrauschen

Die Ursache des Quantisierungsrauschens ist die endliche Wortbreite, die in digitalen Systemen eine beliebig genaue Berechnung der gewünschten Ergebnisse verhindert.

Quantisierung bewirkt durch die entstehenden Stufen im Signalverlauf, dass aus dem LTI-System ein nichtlineares System wird. Eine genaue Analyse nichtlinearer Systeme ist sehr schwierig, weshalb mit Hilfe von Simulationen die Effekte der Quantisierung ermittelt werden und überprüft wird, ob die geforderte Spezifikation noch erfüllt wird. Für erste Untersuchungen kann das Verhalten mit einem linearen Rauschmodell beschrieben werden, welches schon gute Ergebnisse liefert.

Lineares Rauschmodell

Das verwendete lineare Rauschmodell wird anhand der Struktur in Direktform I erklärt und kann auf die anderen Grundstrukturen übertragen werden.

Die Einschränkung *endliche Wortlänge* bedeutet anschaulich das Runden oder Abschneiden der letzten B Nachkommastellen einer Zahl. Die Differenz aus quantisiertem und ursprünglichem Signal

$$e_{k,n} = Q\{b_k x_n\} - b_k x_n \quad (4.62)$$

kann als Rauschen betrachtet werden (siehe hierzu auch Abb. 4.12). $Q\{\cdot\}$ bezeichnet dabei den Quantisierungsoperator, welcher die Rundung bzw. das Abschneiden von Stellen hervorruft. Für das modellierte Quantisierungsrauschen werden die folgenden Annahmen getroffen:

- a) e_n ist weißes Rauschen,
- b) die Amplitudenverteilung von e_n ist gleichverteilt über einem Quantisierungsintervall,
- c) die Quantisierungsrauschquelle ist unkorreliert mit ihrem Eingangssignal und mit den anderen Rauschquellen.

Obwohl Annahme c) streng genommen nicht gilt, da e_n natürlich von seinem Eingang abhängt, können damit gute Ergebnisse erzielt werden. Dies gilt besonders für breitbandige Signale wie Sprachsignale, da bei solchen Signalen ein schneller Wechsel zwischen unterschiedlichen Quantisierungsstufen erfolgt.

Aufgrund der Eigenschaften a) bis c) können die einzelnen Rauschquellen in Abb. 4.12 zusammengefasst werden, was für die direkte Form I zu der Darstellung in Abb. 4.13 führt. Diese Darstellung stellt die Grundlage der weiteren Betrachtungen dar.

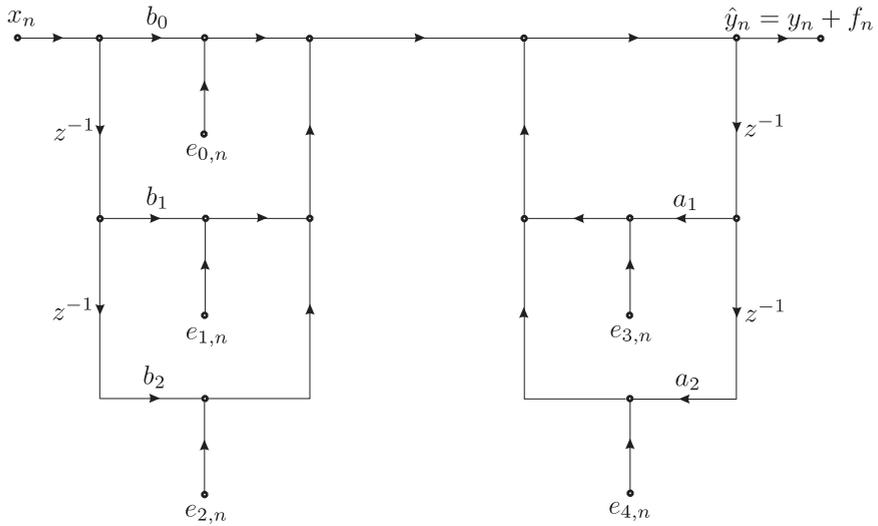


Abbildung 4.12. Lineares Rauschmodell eines Systems in der Grundstruktur Direktform I.

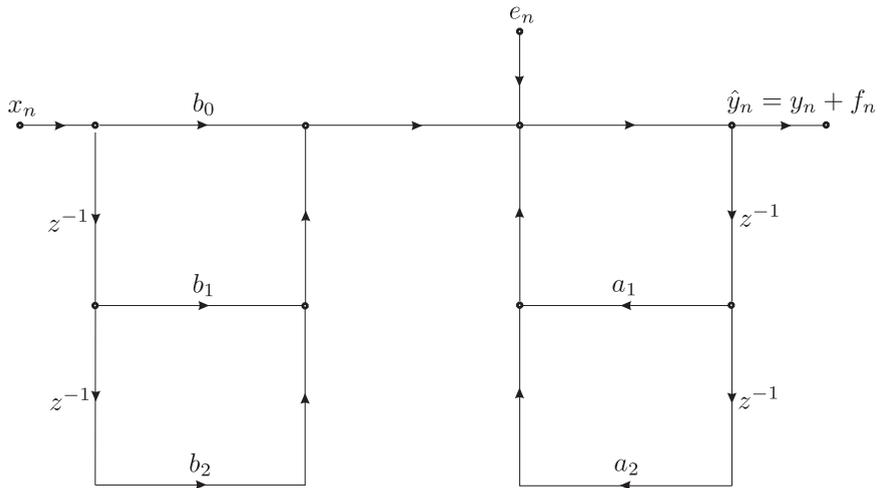


Abbildung 4.13. Vereinfachtes lineares Rauschmodell eines Systems in der Grundstruktur Direktform I.

Für die Gesamtrauschquelle gilt demnach:

$$e_n = \sum_{i=0}^{p+q} e_{i,n}, \tag{4.63}$$

$$\sigma_e^2 = \sum_{i=0}^{p+q} \sigma_{ei}^2 = \frac{2^{-2B}}{12} \cdot (q + 1 + p). \tag{4.64}$$

Die Funktion f_n aus Abb. 4.13 stellt hierbei die Reaktion des Systems auf die Eingangsfolge e_n dar. Auf Basis dieser Signale können nun der Erwartungswert und die Varianz des Ausgangsrauschens bestimmt werden [OSB04]. Je nachdem, ob die z -Übertragungsfunktion, der Frequenzgang oder die Impulsantwort des zeitdiskreten Filters bekannt ist, kann eine der drei folgenden Gleichungen zur Bestimmung von σ_f^2 verwendet werden:

$$\sigma_f^2 = \sigma_e^2 \cdot \int_{-\pi}^{+\pi} |G_{ef}(f)|^2 df, \quad (4.65)$$

$$\sigma_f^2 = \sigma_e^2 \cdot \sum_{n=-\infty}^{\infty} |g_{ef,n}|^2, \quad (4.66)$$

$$\sigma_f^2 = \sigma_e^2 \cdot \frac{1}{2\pi j} \oint_C G_{ef}(z) \cdot G_{ef}(z^{-1}) \cdot z^{-1} dz. \quad (4.67)$$

Das Integral (4.67) kann dabei mit dem Cauchy-Satz gelöst werden [BS⁺12].

In (4.65) bis (4.67) muss beachtet werden, dass die Übertragungsfunktion $G_{ef}(z)$ nur vom Eingang des Rauschens zum Ausgang, *nicht* jedoch diejenige durch das Gesamtsystem hindurch verwendet werden muss! Entsprechendes gilt für die Impulsantwort $g_{ef,n}$.

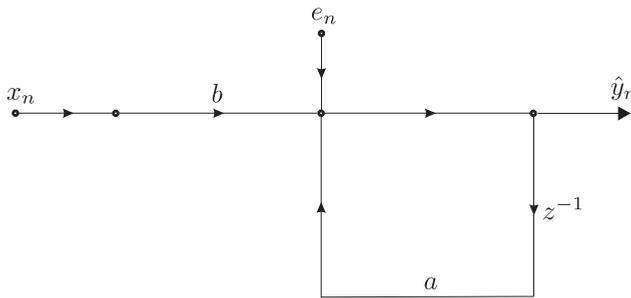


Abbildung 4.14. Rauschmodell für ein System 1. Ordnung in Direktform I.

Beispiel: Für ein System 1. Ordnung mit der Struktur in Abb. 4.14 gilt beispielsweise:

$$G_{ef} = \frac{1}{1 - az^{-1}}, \quad (4.68)$$

oder z -rücktransformiert innerhalb des Konvergenzgebietes

$$g_{ef,n} = a^n \cdot \sigma_n, \quad a < |z|. \quad (4.69)$$

Unter Verwendung von (4.64) und (4.66) erhält man also für $q = 0$ und $p = 1$

$$\sigma_f^2 = 2 \cdot \frac{2^{-2B}}{12} \sum_{n=0}^{\infty} a^{2n} = 2 \cdot \frac{2^{-2B}}{12} \cdot \frac{1}{1 - a^2}. \quad (4.70)$$

Man kann erkennen, dass die Varianz des Ausgangsrauschens für $|a| \rightarrow 1$ zunimmt und daher Vorsicht bezüglich der Systemstabilität geboten ist.

Das Rauschmodell, das für die Direktform I präsentiert wurde, kann – wie bereits erwähnt – auf die anderen Formen erweitert werden, allerdings werden die Betrachtungen dann schnell sehr kompliziert. Eine angemessene Behandlung führt für den Rahmen des Praktikums zu weit.

Allgemein kann man jedoch sagen, dass das Ziel der Betrachtungen des Quantisierungsrauschens stets die Auswirkung auf das Ausgangsrauschen ist, welches durch seine Varianz σ_f^2 charakterisiert wird. Hierbei ist natürlich wieder die Lage der Polstellen der Übertragungsfunktion von besonderem Interesse.

4.3 Versuchsaufbau

4.3.1 Beschreibung des Versuchs

Im ersten Teil des Versuchs arbeiten Sie mit einem Funktionsgenerator sowie mit einem Oszilloskop. Beide werden an ein Analogschaltungsboard angeschlossen, der Funktionsgenerator an den Eingang links, das Oszilloskop an den Ausgang der Schaltung rechts. Vergewissern Sie sich, bevor es losgeht, ob alle Geräte, insbesondere die Analogschaltung, an die Spannungsversorgung angeschlossen und eingeschaltet sind.

Der zweite Teil des Versuchs wird an dem Crashtest-Auto mit Hilfe der *AutoBox* sowie eines PCs durchgeführt. Dazu muss zunächst ein *Simulink*-Modell zum Abtasten der Sensorsignale erstellt und auf der *AutoBox* ausgeführt werden. Mit Hilfe dieses Modells werden Sie die Parameter wie Offset und Empfindlichkeit des Sensors ermitteln. Des Weiteren wird ein Filtermodell erstellt, mit dem die aufgezeichneten Daten eines Crashes offline weiterverarbeitet werden. Am Schluss werden dann Abtast- und Filtermodell zusammen auf die *AutoBox* geladen und nur das gefilterte Signal aufgezeichnet.

Schalten Sie beim Einschalten zwingend zuerst die *AutoBox* und dann den PC ein, beim Ausschalten zuerst den PC und dann die *AutoBox* aus!

Um Zeit und Geld zu sparen, benutzt man in der Elektrotechnik heutzutage unter anderem folgende zwei Verfahren zum Entwickeln und Testen von neuen Produkten:

- **Rapid Control Prototyping (RCP):** RCP bezeichnet ein rechnergestütztes Entwurfsverfahren zur Regelungs- und Steuerungsentwicklung. Bei diesem werden

ein Modell des zu beschreibenden Systems und ein auf diesem Modell basierendes Regelungsmodell entwickelt (Abb. 4.15). Der Test eines solchen Reglers kann z. B. mit HiL vorgenommen werden.

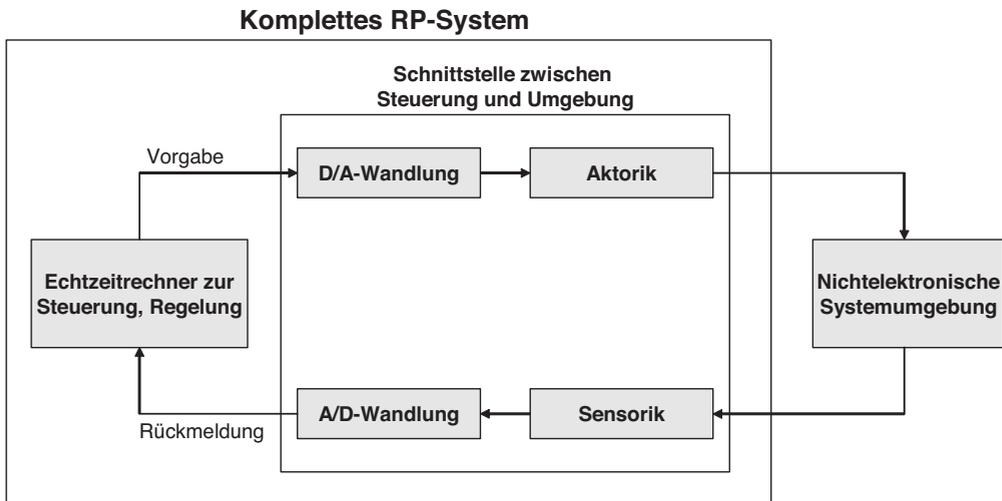


Abbildung 4.15. Rapid Prototyping.

- **Hardware in the Loop (HiL):** Eng mit RCP hängt das sogenannte HiL-Testen zusammen (Abb. 4.16). Dies bedeutet den Echtzeittest von Elektronikkomponenten wie zum Beispiel einem oder mehreren Steuergeräten (um auch die Interaktion zwischen diesen zu untersuchen), bei dem die Ausgangssignale des Steuergeräts auf vorgegebene Sensordaten (Eingangssignale) untersucht werden. Dabei werden die Eingangssignale vom sogenannten HiL-Simulator erzeugt und die Ausgangssignale von diesem erfasst. Diese Methode erlaubt es, in vielen Bereichen der Technik, wie z. B. im Automobilbau oder in der Luftfahrt, auf einfache Weise mit dem HiL-Simulator Simulationen durchzuführen, ohne das Steuergerät dabei im realen Einsatz testen zu müssen. Dies bietet unter anderem folgende Vorteile:
 - kein Simulieren eines kompletten Szenarios notwendig, sondern z. B. nur von Grenzsituationen,
 - standardisierte Tests können erstellt werden, bei denen immer die gleichen Eingangssignale auf das zu testende System gegeben werden und zum Beispiel die Reaktionen von verschiedenen Geräteversionen darauf ermittelt werden,
 - keine katastrophalen Auswirkungen von Fehlern wie Unfälle, Verletzungen etc.,
 - geringere Kosten.

Um einen kleinen Einblick in die Mittel moderner Entwurfs- und Testverfahren zu gewähren, wird in diesem Versuch mit der *dSPACE AutoBox* gearbeitet. Diese kann

unter anderem als Basis für HiL-Simulationen (als HiL-Simulator) eingesetzt werden. Sie kann im Fahrzeug selbst installiert werden und über verschiedene Schnittstellen (Ein-/Ausgabekarten, CAN-Schnittstelle) direkt in das Fahrverhalten des Fahrzeugs eingreifen. Es ist sogar möglich, das gesamte Fahrzeug ohne Fahrer zu steuern.

Um die Komplexität dieses Versuches nicht ausufern zu lassen, wird die *AutoBox* in diesem Versuch allerdings nur zum Abtasten von Signalen verwendet.

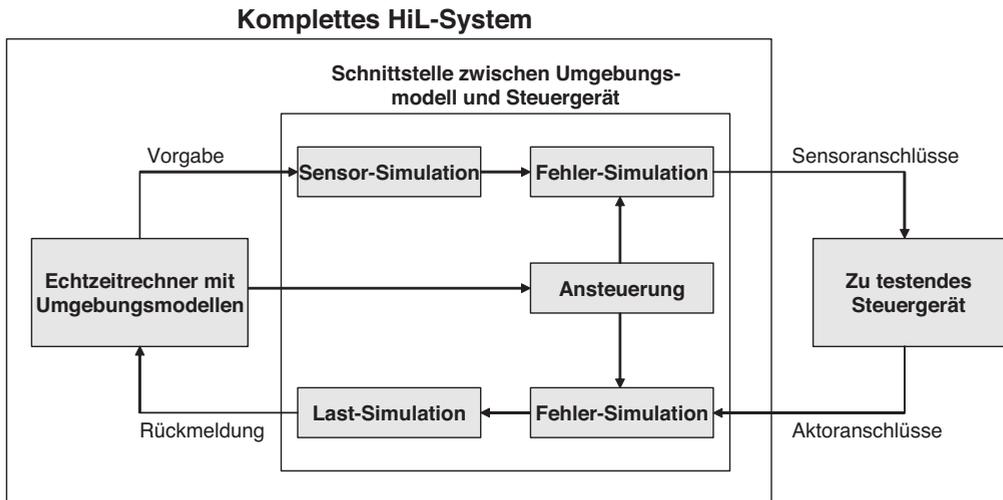


Abbildung 4.16. Hardware in the Loop.

4.3.1.1 Beschreibung der Hardware

Im zweiten Teil dieses Versuches werden Sie mit einer *AutoBox* arbeiten, in der folgende Hardwaremodule installiert sind:

- Prozessorkarte *DS1005*,
- Multi-Channel A/D-Karte *DS2003*,
- CAN Interface Board *DS4302*,
- Serial Interface Board *DS4201-S*.

Direkt an die A/D-Karte wird das *DS2002/DS2003 MUX A/D-Board* angeschlossen, das BNC-Anschlüsse für alle 32 aufzeichnbare Kanäle bietet. An die ersten 3 Kanäle werden die 3 Ausgänge der Verstärkerbox angeschlossen (Abb. 4.17). Die Ausgangskabel des Crashtest-Autos werden in die Sensor-Eingänge der Verstärkerbox gesteckt.

Der Crashtest-Schlitten, der in diesem Versuch verwendet wird, ist in Abb. 4.18 maßstabsgetreu skizziert. Im Versuch wird das Fahrzeug mit drei 50 g-Beschleunigungssensoren bestückt. Diese messen die Beschleunigungen in x -, y - und z -Richtung.

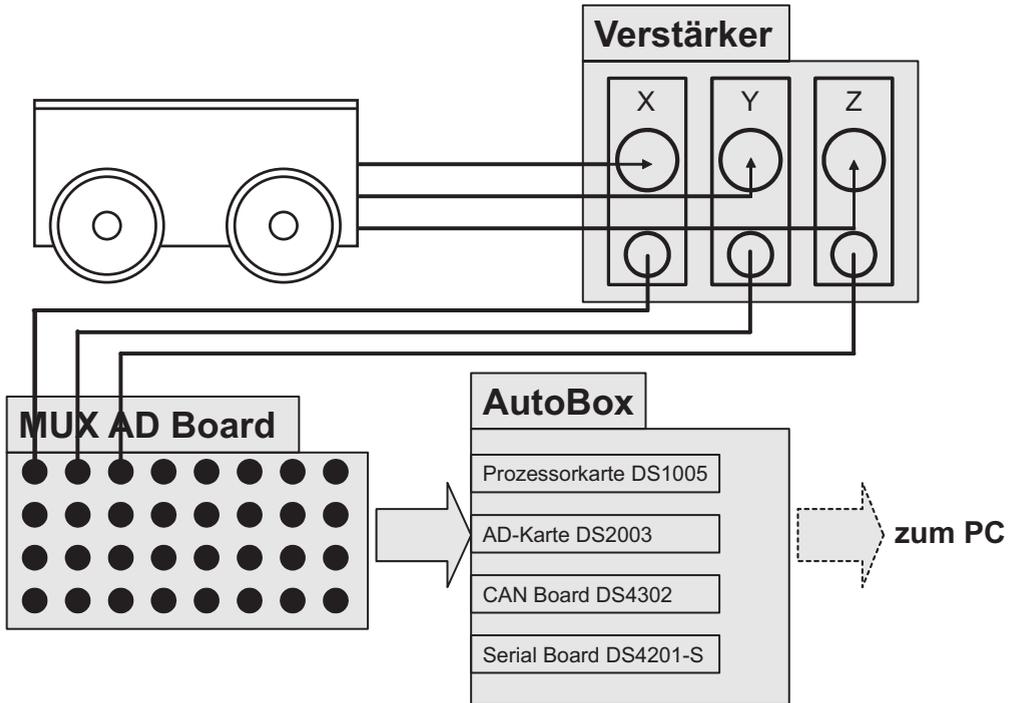


Abbildung 4.17. Verbindungen zwischen den Geräten.

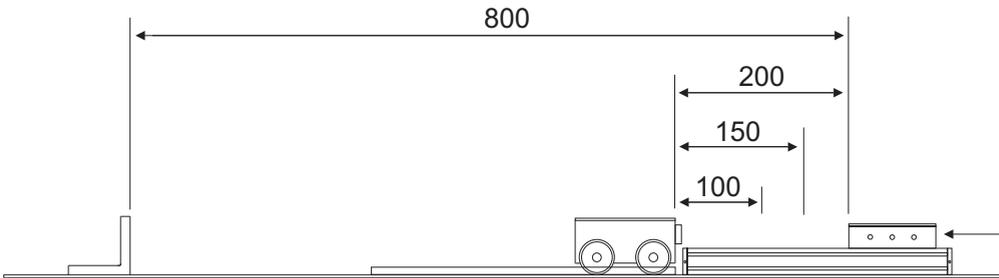


Abbildung 4.18. Maßstabsgetreue Skizze des Crashtest-Schlittens.

Der Schlitten, der das Fahrzeug in Bewegung versetzt, wird mit Hilfe des darunter liegenden Kolbens beschleunigt. Durch einströmende Druckluft in die rechte Öffnung des Kolbens wird der Schlitten gegen das Fahrzeug katapultiert. Hierbei besteht die Möglichkeit, den Weg, den der Schlitten bis zur Übergabe seines Impulses zurücklegt, mit Hilfe eines Bolzens im Kolben auf 100 mm, 150 mm oder 200 mm einzustellen.

Anfangs fährt das Modellfahrzeug in einer Führungsschiene, damit seine Bewegung stabilisiert wird. Diese endet jedoch vor dem Hindernis, damit das Fahrzeug beim Crash in allen Koordinatenachsen frei beweglich ist. Ansonsten würde nur der

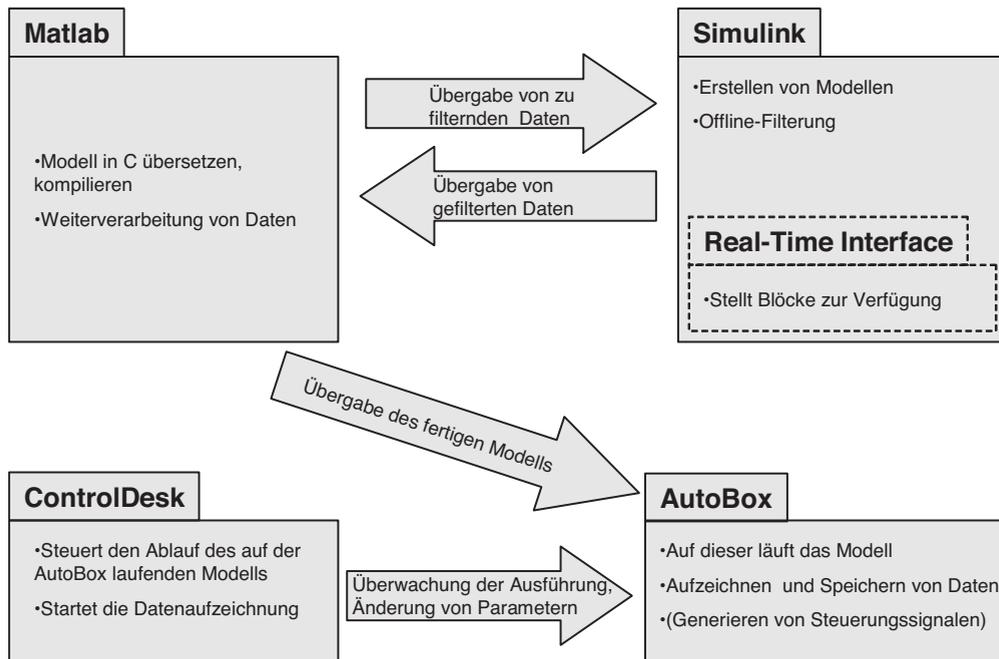


Abbildung 4.19. Interaktionen zwischen den Programmen.

Beschleunigungssensor in x -Richtung eine negative Beschleunigung bzw. einen Abbremsvorgang registrieren.

Bevor ein weiterer Crash durchgeführt werden kann, muss der Schlitten wieder in seine Ausgangslage gebracht werden. Dies geschieht mit Hilfe der linken Kolbenöffnung. Sobald Druckluft hindurch strömt, wird der Schlitten in Bewegung versetzt und kehrt zu seiner Ausgangsposition zurück.

4.3.1.2 Beschreibung der Programme

Zur Ausführung von bestimmten Aufgaben wird die *AutoBox* unter *Simulink* mit dem Real-Time Interface, einem echtzeitfähigen Matlab-Paket, angesteuert. Dies funktioniert folgendermaßen (Abb. 4.19): In *Simulink* wird ein Modell erstellt, das Blöcke des von *dSPACE* mitgelieferten Real-Time Interface verwendet. Diese *Simulink*-Modelle werden dann automatisch in die Programmiersprache C übersetzt, kompiliert, auf die *AutoBox* geladen und dort ausgeführt. Zusätzlich zu den Standard-*Simulink*-Bibliotheken gibt es zur Ansteuerung der *AutoBox* eigene Bibliotheken.

Zur Aufnahme von Daten erstellt man ein *Simulink*-Modell, das später auf der *AutoBox* ausgeführt wird. Für dieses Modell sind folgende Blöcke wichtig:

- der Block MUX_ADC aus dem Untermenü DS2003 der Bibliothek dSPACE rti1005. Dieser Multiplexer stellt die gemessenen Daten aller Kanäle zur Verfügung. Folgende Einstellungen sind wichtig:
 - Bei der Registerkarte `Unit` wählt man die Kanäle aus, deren Daten angezeigt werden sollen.
 - Unter `Range` definiert man den Eingangsspannungsbereich, für jeden Kanal in allen Versuchen ± 10 V.
 - Bei `Resolution`: 16 bits.
- Demultiplexer (zu finden in der Bibliothek Commonly Used Blocks): Wird direkt an den Multiplexer angeschlossen und gibt die aufgezeichneten Kanäle einzeln aus. Wichtig ist, dass man in den Einstellungen die korrekte Anzahl der Ausgänge angibt.
- 3 `Scopes` (ebenfalls in der Bibliothek Commonly Used Blocks) oder andere Anzeige- oder Ausgabeblöcke, die mit den Ausgängen des Demultiplexers verbunden werden. Die Signalverläufe im `Scope` braucht man sich nicht anzuschauen; diese Blöcke werden allerdings zur späteren Weiterverarbeitung der Daten in *ControlDesk* benötigt (bei Verwendung eines Terminators ist dies nicht möglich).



Abbildung 4.20. Demultiplexer.



Abbildung 4.21. Multiplexer.



Abbildung 4.22. Scope.

Zur Einstellung der Modellparameter wählt man im Menü *Simulation* den Unterpunkt *Model Configuration Parameters*. Unter den Einträgen in der Liste im linken Teil des Fensters gibt man unter *Solver* im Feld *Fixed-step size* die Abtastzeit in Sekunden an. Außerdem ist der *Solver* zu *Discrete* zu wählen. **Alle anderen Einstellungen dürfen nicht verändert werden!**

Um das Modell auszuführen, klickt man im Menü *Model Configuration Parameters* unter *Code Generation* auf *Build*. Damit wird das Modell in *Matlab* automatisch erstellt und in C-Code umgewandelt, der dann auf der *AutoBox* ausgeführt wird. Die Ausführung des *Simulink*-Modells lässt sich mit Hilfe des dSPACE-Programms *ControlDesk* in Echtzeit steuern, so kann man z. B. Simulationsparameter verändern und Signalverläufe aufzeichnen. Hierbei ist zu beachten, dass *ControlDesk* vor dem Ausführen des Modells geöffnet sein muss. *ControlDesk* ist ein eigenständiges Programm und kann über das Startmenü aufgerufen werden.

Zur Speicherung der aufgezeichneten Daten erstellt man in *ControlDesk* mit *File* → *New* → *Project+Experiment* ein neues Experiment. Dazu ist nach Eingabe von Projekt- und Experimentname bei *Add Platform/Device* das DS 1005 PPC Board auszuwählen. Unter *Select Variable Description* wählt man die *.sdf*-Datei des *Simulink*-

Modells aus. Das neue Projekt enthält dann eine neu erstellte grafische Benutzeroberfläche, ein sogenanntes Layout. Dieses ist in Abb. 4.23 dargestellt. Auf dem Layout platziert man so viele Plotter-Elemente, wie Variablen aufgezeichnet werden sollen. Ein einzelnes Plotter-Element wird dabei aus dem Instrument Selector rechts neben dem Layout auf dieses gezogen.

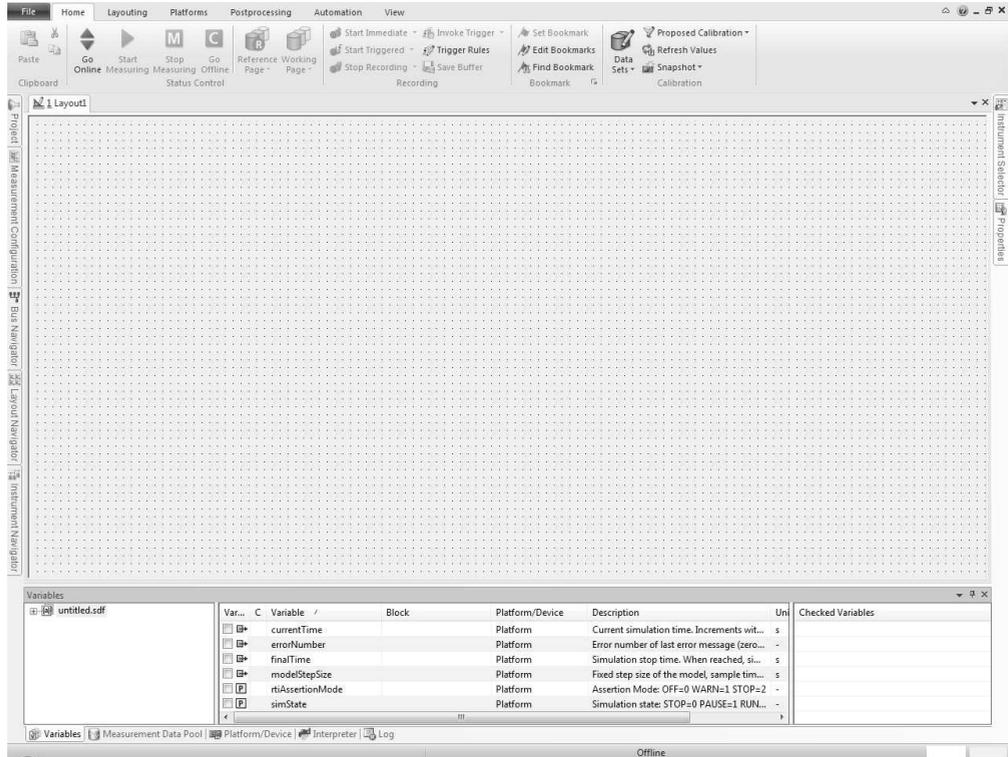


Abbildung 4.23. ControlDesk: Fenster mit neu erstelltem Layout.

- Um einem Plotter einen Signalverlauf zuzuordnen, muss im Variable Browser links unten im Control Desk-Fenster die entsprechende .sdf-Datei ausgewählt werden. Dann zieht man nach Auswahl der entsprechenden Simulink-Blöcke aus dem Fenster rechts neben dem Variable Browser die gewünschte Variable (für jeden Block jeweils In1) auf den Plotter.
- Um das Abspeichern der aufgezeichneten Daten vorzubereiten, klickt man auf die Schaltfläche Measurement Configuration links neben dem Layout (siehe Abb. 4.24). Nach Rechtsklick auf Recorder und Klick auf Properties wählt man im sich daraufhin öffnenden Fenster unter Stop Condition das Time limit zur gewünschten Dauer der Aufzeichnung. Unter Storage information aktiviert man die Checkbox Automatic export. Außerdem ist der Dateityp Automatic export: file type (MATLAB-files) sowie der Ordner anzugeben, in dem die Daten gespeichert werden sollen.

- Im Measurement Configuration-Menü muss unter Triggers → Duration Trigger die standardmäßig aktivierte Checkbox Duration Trigger 1 deaktiviert werden. Somit ist sichergestellt, dass das Signal über die gewünschte Dauer abgetastet wird.

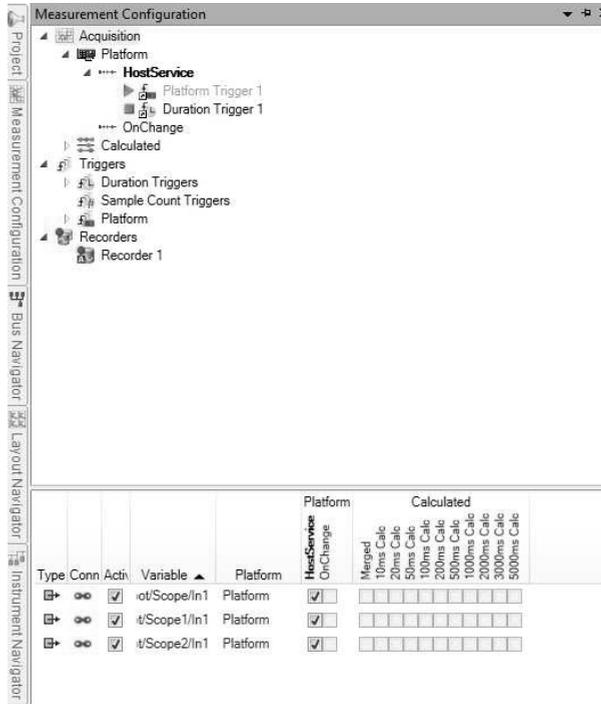


Abbildung 4.24. ControlDesk: Measurement Configuration.

Um die Aufnahme zu starten, klickt man im *ControlDesk*-Hauptfenster auf die Schaltfläche Start Triggered. Dadurch wird das Signal für die definierte Zeit aufgezeichnet und mit dem angegebenen Dateinamen gespeichert. Während des erneuten Ladens eines Modells auf die *AutoBox* muss diese im Offline-Modus betrieben werden (Go Offline), sonst kann das Modell in *Simulink* nicht erstellt werden und *Matlab* gibt eine Fehlermeldung.

Dringend beachten: Die *AutoBox* gibt die gemessenen Signale immer in Relation zum maximalen Eingang an. Wählt man zum Beispiel bei den Optionen des Blocks DS2003_B1 eine Eingangsspannung von ± 10 V aus, so entspricht ein von diesem Block ausgegebenes Signal mit der Amplitude 1 einem realen Signal von 10 V.

Die Signalverläufe können auch auf dem nichtflüchtigen Flash-Speicher der *AutoBox* gespeichert und nach Beendigung der Simulation ausgewertet werden (damit kann man z. B. die *AutoBox* im automatisierten Zustand ohne angeschlossenen PC betreiben). Allerdings ergibt sich bei unseren hohen Abtastraten eine so große Datenmenge, dass das Auslesen für unsere Zwecke zu lange dauern würde.

Für weiterführende Informationen zu *ControlDesk* bietet sich die umfangreiche Hilfe *HelpDesk*, zu finden unter Help → Help Topics, an.

4.3.2 Versuchsvorbereitende Aufgaben

In diesen versuchsvorbereitenden Aufgaben sollen die theoretischen Zusammenhänge dieses Versuchs vertieft werden, bevor sie dann im praktischen Teil auf eine reale Problemstellung, nämlich die Filterung von Beschleunigungssignalen eines Crashtestautos, angewendet werden.

Aufgabe 4.1: Dämpfung

Die Dämpfung eines Filters bei der Knickfrequenz beträgt ungefähr -3 dB. Welcher Leistung und Amplitude entspricht das? Wie lautet die Beziehung zwischen der dB-Skala, der Leistung und der Amplitude?

Aufgabe 4.2: Frequenzantwort verschiedener Filter

Was ist der Hauptunterschied zwischen den Frequenzantworten eines

- a) Butterworth-Filters?
- b) Tschebyscheff-Filters?
- c) Cauer-Filters?
- d) Welche Filterart würden Sie empfehlen, wenn die Breite des Übergangsbereiches minimiert werden soll?

Aufgabe 4.3: Systemfunktion und Frequenzantwort

Welchen Zusammenhang gibt es zwischen der Systemfunktion $G(z)$ und der Frequenzantwort $G(e^{j2\pi f t_A})$?

Aufgabe 4.4: Phasengang, Gruppenlaufzeit

- a) Wie lautet die Definition des Phasengangs eines zeitdiskreten Systems?
- b) Was gilt im Falle eines linearen Phasengangs?
- c) Was bezeichnet die Gruppenlaufzeit?

Aufgabe 4.5: FIR-Filter

Beantworten Sie die folgenden Fragen. Begründen Sie ihre Aussagen und finden Sie ggf. ein Beispiel oder Gegenbeispiel!

- Ist ein FIR-Filter immer stabil?
- Ist es immer kausal?
- Hat es immer linearen Phasengang?

Aufgabe 4.6: Pol-Nullstellen-Diagramme

In Abb. 4.6 sind vier Pol-Nullstellen-Diagramme gegeben. Nutzen Sie zur Beantwortung der folgenden Fragen die Ausführungen in Abschnitt 4.2.1.2. Fahren Sie die Frequenzachse, also den Einheitskreis, von 1 bis -1 ab und leiten Sie sich damit den Amplitudengang des jeweiligen Filters her. Begründen Sie Ihre Antworten!

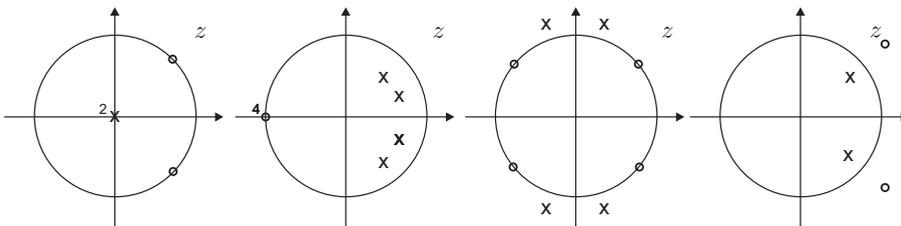


Abbildung 4.25. Pol-Nullstellen-Diagramme der Filter 1–4.

- Handelt es sich bei den durch sie repräsentierten Filtern um Tiefpass-, Hochpass-, Bandpass-, Bandstop- oder Allpass-Filter?
- Sind es FIR- oder IIR-Filter? Welcher Ordnung?
- Sind sie stabil?
- Ist der Phasengang linear?

Aufgabe 4.7: Serienschaltung von Filtern

Zwei Filter haben die Impulsantworten h_1 und h_2 .

- Wie lautet die Impulsantwort einer Serienschaltung der beiden Filter?
- Wie sieht die Systemfunktion aus?

Aufgabe 4.8: Differenzgleichung

Ein kausales, lineares, verschiebungsinvariantes System kann mittels der folgenden Differenzgleichung beschrieben werden:

$$y_n = \frac{1}{4} y_{n-1} + \frac{1}{8} y_{n-2} + x_n - x_{n-1}. \quad (4.71)$$

Bestimmen Sie die Übertragungsfunktion $H(z)$ und die Impulsantwort h_n .

4.3.3 Versuche am Versuchstag

4.3.3.1 Versuche mit der Analog-Hardware

Aufgabe 4.9: Analogschaltung

Arbeiten Sie in diesem Versuchsteil mit der Anlogschaltung 1. Diese Schaltung stellt ein lineares Filter unbekannter Charakteristik dar. Ziel ist es, durch Messungen mit Funktionsgenerator und Oszilloskop das Filter zu charakterisieren.

Stellen Sie am Funktionsgenerator eine Sinusspannung mit einer Amplitude von 1 V ein und überprüfen Sie bei Veränderung der Frequenz mittels des Oszilloskops die Ausgangsspannung!

- a) Bestimmen Sie die Knickfrequenz der Schaltung (± 3 dB)! Welcher Verstärkung entsprechen 3 dB? (Rechnung!)
- b) Um was für ein Filter handelt es sich (Tiefpass, Hochpass, Bandpass oder Bandsperr)? Begründung!
- c) Bestimmen Sie durch Wahl eines weiteren geeignet gewählten Messpunktes für die Frequenz die Steigung bzw. das Gefälle des Amplitudengangs! Geben Sie die Frequenz sowie die Verstärkung Ihres Messpunktes an! Welche Ordnung hat das Filter?
- d) Zeichnen Sie eine mögliche Realisierung der Anlogschaltung als passives Netzwerk!

4.3.3.2 Versuche am PC

In diesem Versuchsteil werden Sie die Beschleunigungssignale eines Crashtest-Autos aufzeichnen und anschließend einer Filterung unterziehen. Das in dieser Filterung verwendete Filter soll zunächst entworfen werden.

Verwenden Sie zum Bearbeiten der folgenden Aufgaben *Matlab*. Wählen Sie nach dem Start die Plattform RTI1005 und bestätigen Sie danach mit OK.

Erstellen Sie unter $D:\PDSV\Gruppen$ ein Verzeichnis für Ihre Gruppe. Legen Sie alle Dateien, die Sie im Laufe des Versuchs erstellen, nur in diesem Ordner ab!

Aufgabe 4.10: Filterung von Crashtest-Signalen mit Matlab

Entwerfen Sie in *Matlab* ein Butterworth-Tiefpassfilter, das folgende Spezifikationen erfüllt:

- $f_p = 1$ Hz, $f_s = 200$ Hz (Grenzfrequenzen Durchlass-/Sperrbereich),
- $\delta_D = 1$ dB, $\delta_S = 15$ dB (Durchlass-/Sperrbereichstoleranzen in dB),
- $f_A = 5$ kHz (Abtastfrequenz der zu filternden Signale).

Schreiben Sie für diese Aufgabe ein m-File, das den Filterentwurf automatisiert durchführt. Verwenden Sie dazu im ersten Schritt den *Matlab*-Befehl `butterord()`. Beachten Sie die Nyquist-Frequenz!

- a) Welche Ordnung und welche Knickfrequenz hat Ihr Filter?
- b) Geben Sie die Filterkoeffizienten mit Hilfe des Befehls `butter()` an!
- c) Geben Sie mit Hilfe von *Matlab* für das entworfene IIR-Filter den Pol-Nullstellen-Plan mit `pzmap()` sowie den Frequenzgang mit `freqz()` an! *Matlab* bietet mehrere Funktionen mit dem Namen `freqz()` an; beachten Sie, dass sie die Funktion mit den korrekten Argumenten (u. a. unter Verwendung von 128 diskreten Frequenzwerten) und ohne Rückgabewerte aufrufen. Weiterer Hinweis: Wie können Sie hier den Befehl `tf()` zur Erzeugung zeitdiskreter Übertragungsfunktionen aus gegebenen Filterkoeffizienten nutzen?

Aufgabe 4.11: Ermittlung der Sensorparameter und Filterung des Crashverlaufs

Wichtige Hinweise für die Arbeit mit der Autobox: Lesen Sie sich die Beschreibung in Abschn. 4.3.1.2 nochmals aufmerksam durch! Setzen Sie den aktuellen *Matlab*-Pfad auf den Gruppenordner (`D:\PDSV\Gruppen\Gruppenname`) und belassen Sie ihn für die Dauer des gesamten Versuchs dort! Arbeiten Sie im Laufe des Versuchs immer mit demselben *Simulink*-Modell mit gleichbleibendem Dateinamen (!) und modifizieren Sie es entsprechend jeder Aufgabe. Sie können nach jeder Änderung eine Kopie des Modells unter einem anderen Namen für Ihre Unterlagen abspeichern, aber der Name des Modells, das jeweils auf die *Autobox* geladen wird, darf nicht geändert werden. Bevor Sie das geänderte Modell mittels `Build` wieder auf die *AutoBox* laden, klicken Sie in *ControlDesk* auf `Go Offline`.

- a) Erstellen Sie mit Hilfe der Blöcke aus Abschn. 4.3.1.2 ein *Simulink*-Modell zur Aufzeichnung der 3 Sensorsignale. Belassen Sie die standardmäßig eingestellte `Simulation Time` bei `inf`.
 - Wählen Sie im *Simulink*-Modell als Abtastfrequenz 5 kHz und als Auflösung 16 bit.
 - Laden Sie das Modell wie in Abschn. 4.3.1.2 beschrieben auf die *AutoBox*.
 - Öffnen Sie *ControlDesk* und erstellen Sie eine Benutzeroberfläche zur Aufnahme der Messdaten. Gehen Sie dabei nach Abschn. 4.3.1.2 vor. Wählen Sie eine Aufnahmedauer im Bereich von 3 – 10 s.
 - Speichern Sie die Messdaten in *ControlDesk* als `.mat`-Dateien ab, damit Sie sie in *Matlab* weiterverarbeiten können.

Im nächsten Schritt sollen nun die Messdaten der Sensoren verwendet werden. Die Sensoren liefern einen Spannungswert, der in eine Beschleunigung umgerechnet

werden muss. Gehen Sie von dem linearen Sensormodell

$$a = \frac{1}{m} \cdot (U - k) \quad (4.72)$$

aus. Hierbei entspricht U der gemessenen Spannung (in V), a der aufgetretenen Beschleunigung (in g), m der Empfindlichkeit (in V/g) und k dem Offset (Einheit V). Wie können Sie die Parameter der Sensoren, also Empfindlichkeit und Offset, ermitteln? Stellen Sie zunächst folgende Überlegungen an: Welcher Spannung entsprechen $0g$, $-1g$, $1g$? Wie können Sie einen Sensor physikalisch mit exakt $1g$ belasten? Bestimmen Sie die Daten jedes Sensors einzeln und nutzen Sie dabei das bereits erstellte Modell. Gehen Sie am Rechner nach Bestimmung der Kalibrierungsdaten wie folgt weiter vor:

- b) Erweitern Sie Ihr *Simulink*-Modell so, dass das Signal offsetfrei wird und auf der Ordinate nicht die gemessene Spannung, sondern die tatsächliche Beschleunigung (in g) aufgetragen wird.
Beschriften Sie in allen Plots die Achsen!
- c) Führen Sie nun einen Crash mit Hilfe der Druckluft-Beschleunigungseinrichtung durch (Druck: 3 bar; Abtastfrequenz 5 kHz). Experimentieren Sie mit verschiedenen Drücken (< 5 bar).

Erstellen Sie für den folgenden Aufgabenteil ein neues *Simulink*-Modell, das sie allerdings nicht auf die *AutoBox* ziehen, da nun lediglich mit aufgezeichneten Daten gearbeitet wird.

- d) Bauen Sie in *Simulink* ein IIR-Filter in Direktform II (vgl. Abb. 4.9) mit den in Aufgabe 4.10 berechneten Koeffizienten auf. Gehen Sie dabei nach der folgenden Anleitung vor: Überlegen Sie sich, welche Blöcke aus der *Simulink*-Bibliothek benötigt werden. Belassen Sie in allen Blöcken die Abtastzeit bei -1 (vererbt aus vorherigem Block). Ergänzen Sie die Blöcke *From Workspace* und *To Workspace* und führen Sie eine Filterung der aufgezeichneten Daten des x -Sensors durch. Der Block *From Workspace* bietet die Möglichkeit, die Eingangsdaten auf unterschiedliche Weise zu importieren, beispielsweise im *timeseries*- oder *structure*-Format (siehe das Fenster mit den Blockparametern). Am einfachsten ist es, eine Matrix zu erstellen, die in der ersten Spalte den Zeitvektor und in den folgenden Spalten die Signalvektoren der einzelnen Kanäle enthält. Achten Sie darauf, die Werte tatsächlich spaltenweise und nicht zeilenweise einzutragen, da die Berechnung sonst äußerst zeitintensiv ist und keine sinnvollen Ergebnisse liefert. Stellen Sie im *To Workspace*-Block das *Save format* auf *Structure With Time*. Passen Sie die Simulationszeit des Modells an die Dauer der aufgezeichneten Signale an. Plotten Sie danach das Signal für verschiedene Crashes (je ein Plot).

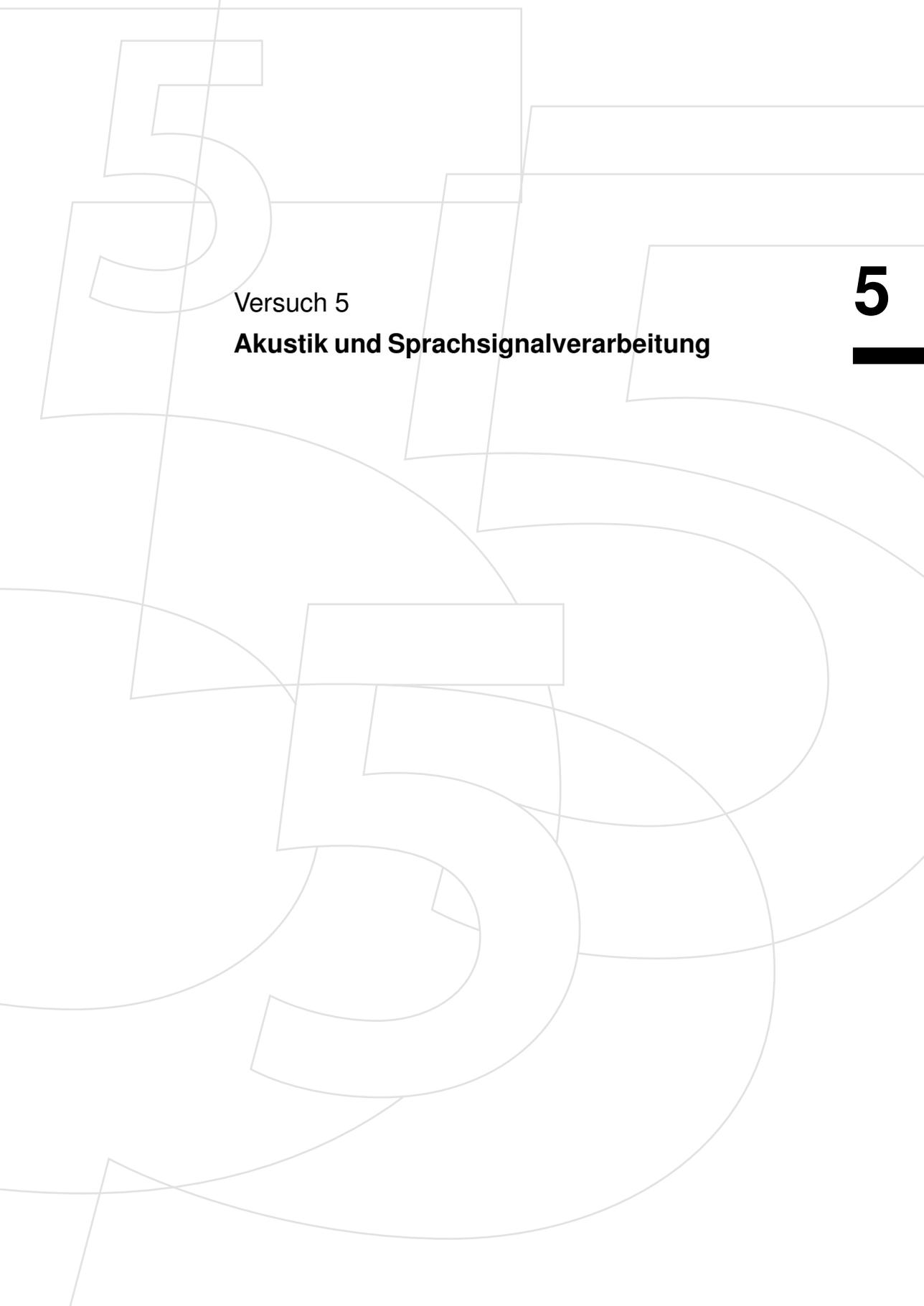
Aufgabe 4.12: Durchführung der Filterung auf der AutoBox

- a) Ihr *Simulink*-Aufzeichnungsmodell soll nun so erweitert werden, dass die Werte des x -Sensors direkt nach der Messung und vor dem Speichern der Daten mit dem Tiefpass aus Aufgabe 4.11 gefiltert werden. Ergänzen Sie dazu ihr *Simulink*-Aufzeichnungsmodell (nur x -Sensor) um die Filterstruktur aus Aufgabe 4.11, sodass die aufgezeichneten Daten das bereits gefilterte Signal wiedergeben. Bauen Sie sowohl für das ungefilterte als auch für das gefilterte Signal je ein Scope ein. Laden sie das Modell auf die *AutoBox* und testen Sie es.
- Plotten Sie das ungefilterte und das gefilterte Signal für verschiedene Crashes.
- b) Welche Crashedauer können Sie dem gefilterten Signal entnehmen (der Crash gelte dabei als beendet, wenn die Beschleunigung für 10 ms dauerhaft unter der Schwelle $a_x = 0,5 g$ liegt)?
- c) Bestimmen Sie die Maximalverzögerung während der Crashphase in g .
- d) Berechnen und plotten Sie die Momentangeschwindigkeit $v(t)$ in ms^{-1} in Matlab! Programmieren Sie im Command Window. Verwenden Sie ein ungefiltertes und Offset-befreites Messsignal. Nehmen Sie dabei an, dass das Fahrzeug am Anfang und am Ende in Ruhe ist! Was muss dann für die Beschleunigung gelten? Korrigieren Sie den Geschwindigkeitsverlauf entsprechend. Programmieren Sie auch hier im Command Window und nutzen Sie die Befehle `trapz()` und `cumtrapz()`. Woran liegt es, dass eine solche Korrektur nötig ist?

4.4

4.4 Literatur

- [BS⁺12] IL'JA N. BRONŠTEJN, KONSTANTIN A. SEMENDJAEV, GERHARD MUSIOL und HEINER MÜHLIG (Herausgeber): *Taschenbuch der Mathematik*. Verlag Harry Deutsch, Frankfurt am Main, 2012.
- [KK12] KARL-DIRK KAMMEYER und KRISTIAN KROSCHEL: *Digitale Signalverarbeitung – Filterung und Spektralanalyse mit MATLAB-Übungen*. Springer Vieweg, Wiesbaden, 8. Auflage, 2012.
- [OSB04] ALAN V. OPPENHEIM, RONALD W. SCHAFER und JOHN R. BUCK: *Zeitdiskrete Signalverarbeitung*. Pearson, München, 2. Auflage, 2004.
- [PJ15] FERNANDO PUENTE LEÓN und HOLGER JÄKEL: *Signale und Systeme*. De Gruyter Oldenbourg, Berlin, 6. Auflage, 2015.



Versuch 5

Akustik und Sprachsignalverarbeitung

5

5	Akustik und Sprachsignalverarbeitung	
5.1	Einleitung	143
5.2	Grundlagen	143
5.2.1	Skalierung	144
5.2.2	Momente von Energiesignalen	144
5.2.2.1	Signalenergie und Energiedichte	144
5.2.2.2	Mittlere Zeit und mittlere Frequenz	144
5.2.2.3	Zeitdauer und Bandbreite	145
5.2.2.4	Zeitdauer-Bandbreite-Produkt	145
5.3	Fourier-Transformation und Leistungsdichte	146
5.3.1	Diskrete Fourier-Transformation (DFT)	146
5.3.1.1	Graphische Veranschaulichung der DFT	146
5.3.1.2	Leckeffekt	150
5.3.1.3	Fensterung im Zeitbereich	151
5.3.2	Kurzzeit-Fourier-Transformation	153
5.3.2.1	Einführung	153
5.3.3	Zeitdiskretisierung	154
5.4	Wavelet-Transformation	156
5.4.1	Kontinuierliche Wavelet-Transformation	156
5.4.1.1	Definition	156
5.4.1.2	Wavelets	158
5.4.2	Diskrete Wavelet-Transformation	159
5.4.2.1	Einführung	159
5.4.2.2	Approximationen und Details	160
5.4.2.3	Multiraten-Filterung	160
5.4.3	Wavelet-Rekonstruktion	161
5.5	Wigner-Ville-Verteilung	162
5.5.1	Ambiguitätsfunktion	162
5.5.2	Definition der Wigner-Ville-Verteilung	162
5.5.3	Eigenschaften der Wigner-Ville-Verteilung	164
5.5.4	Kreuzterme	166
5.5.5	Pseudo-Wigner-Ville-Verteilung	167
5.5.6	Geglättete Pseudo-Wigner-Ville-Verteilung	167
5.5.7	Zeitdiskrete Wigner-Ville-Verteilung	168
5.5.8	Zusammenhang zwischen Wigner-Ville-Verteilung und STFT	168
5.6	Sprachsignale	169
5.6.1	Sprachproduktion	169
5.6.2	Eigenschaften	170
5.6.3	Formanten	171
5.7	Schallausbreitung in begrenzten Räumen	171

5.7.1	Signalmodelle	171
5.7.1.1	Freifeld	171
5.7.1.2	Reflexionsbehaftete Umgebung	172
5.7.2	Raumimpulsantworten	172
5.8	Laufzeit- und Richtungsschätzung	172
5.8.1	Laufzeitschätzung	173
5.8.1.1	Kreuzkorrelation	173
5.8.1.2	Verallgemeinerte Kreuzkorrelation	174
5.8.2	Ermittlung der Schalleinfallrichtung	174
5.9	Versuchsaufbau und Matlab-Umgebung	175
5.9.1	Versuchsaufbau	175
5.9.1.1	Mikrofone	175
5.9.1.2	Analogelektronik	175
5.9.2	Matlab-Umgebung	175
5.10	Versuchsvorbereitende Aufgaben	176
5.11	Versuche am Versuchstag	177
5.12	Literatur	184

5 Akustik und Sprachsignalverarbeitung

5.1 Einleitung

5.1

Der vorliegende Versuch soll einen Einblick in die Akustik und die Sprachsignalverarbeitung geben – zwei Themengebiete, mit denen wir alltäglich konfrontiert werden. Innerhalb der Akustik (Lehre vom Schall) liegt im Rahmen dieses Versuchs der Fokus auf der Schallausbreitung in begrenzten Räumen und der entsprechenden Wirkung auf Sprachsignale. Die Sprachsignalverarbeitung ist ebenfalls ein breitgefächertes Gebiet, das sich unter anderem mit Aspekten der Sprachentstehung, Spracherkennung und Sprecheridentifikation befasst. Aus dem Bereich der Sprachsignalverarbeitung soll auf die menschliche Sprachproduktion und die resultierenden Charakteristika eingegangen werden. Die menschliche Sprache besitzt einen zeitvarianten Frequenzgehalt. Eine Bestimmung der beteiligten Frequenzen darf somit nur über einen begrenzten Zeitraum erfolgen, in dem sich der Frequenzgehalt nicht ändert (quasistationärer Bereich). Zur Analyse der Signale sind aus diesem Grund Zeit-Frequenz-Darstellungen erforderlich. Es werden verschiedene Methoden der Zeit-Frequenz-Darstellung vorgestellt und ihre Eigenschaften sowie Vor- und Nachteile diskutiert. Diese mathematischen Darstellungen sind universelle Werkzeuge, die eine zeitabhängige Frequenzanalyse nicht nur von akustischen, sondern auch von beliebigen anderen Signalen erlauben.

Die Versuchsdokumentation kann in drei grundlegende Abschnitte eingeteilt werden. In den Abschnitten 5.2 bis 5.5 werden zunächst Grundlagen vermittelt und anschließend drei verschiedene Zeit-Frequenz-Darstellungen vorgestellt. In den Abschnitten 5.6 bis 5.8 werden spezifische Aspekte der Akustik und der Sprachsignalverarbeitung besprochen, welche die für die Durchführung der Versuche notwendigen Kenntnisse vermitteln. Abschließend erfolgt in den Abschnitten 5.9 und 5.10 die Beschreibung des Versuchsaufbaus und der Versuchsdurchführung.

5.2 Grundlagen

5.2

Zunächst werden einige allgemeine Signaleigenschaften vorgestellt, die für die danach folgende Zeit-Frequenz-Analyse benötigt werden.

5.2.1 Skalierung

Bei der Wavelet-Transformation (Abschn. 5.4) spielt die **Skalierung** von Signalen eine bedeutende Rolle. Wird die Zeit t mit dem Faktor $a > 1$ skaliert, so ist das Signal

$$x_a(t) = x\left(\frac{t}{a}\right) \quad (5.1)$$

gegenüber dem ursprünglichen Signal um den Faktor a gestreckt. Für die Fourier-Transformierte von $x_a(t)$ gilt dann [PJ15]:

$$X_a(f) = a X(af). \quad (5.2)$$

Das Spektrum $X_a(f)$ ist also gegenüber $X(f)$ um den Faktor a gestaucht. Die Amplituden von $X_a(f)$ sind um den Faktor a vergrößert. Für $a < 1$ wird das Signal gestaucht, wodurch das Spektrum $X_a(f)$ gestreckt und dessen Amplitude verkleinert wird.

5.2.2 Momente von Energiesignalen

Um zu beschreiben, in welchem Zeit- und Frequenzbereich sich die Energie eines Signals konzentriert, werden die ersten beiden Momente der normierten zeitlichen bzw. spektralen Energiedichte betrachtet.

5.2.2.1 Signalenergie und Energiedichte

Die **Energie** E_x eines Energiesignals $x(t)$ ist wie folgt definiert:

$$E_x = \langle x(t), x(t) \rangle = \int_{-\infty}^{\infty} |x(t)|^2 dt. \quad (5.3)$$

Aufgrund der Parseval'schen Beziehung gilt auch:

$$E_x = \langle X(f), X(f) \rangle = \int_{-\infty}^{\infty} |X(f)|^2 df. \quad (5.4)$$

Der Ausdruck $|x(t)|^2$ kann als **zeitliche Energiedichte** und der Ausdruck $|X(f)|^2$ als **spektrale Energiedichte** interpretiert werden. Von **normierten Energiedichten** wird gesprochen, wenn die Energiedichten auf die Signalenergie E_x bezogen werden.

5.2.2.2 Mittlere Zeit und mittlere Frequenz

Die **mittlere Zeit** t_x ist das erste Moment der normierten zeitlichen Energiedichte:

$$t_x = \frac{1}{E_x} \int_{-\infty}^{\infty} t |x(t)|^2 dt. \quad (5.5)$$

Analog ist die **mittlere Frequenz** f_x das erste Moment der normierten spektralen Energiedichte:

$$f_x = \frac{1}{E_x} \int_{-\infty}^{\infty} f |X(f)|^2 df. \quad (5.6)$$

Die mittlere Zeit (5.5) und die mittlere Frequenz (5.6) liefern ein Maß, um welche Zeit bzw. Frequenz die Signalenergie konzentriert ist.

5.2.2.3 Zeitdauer und Bandbreite

Die **Zeitdauer** Δ_t ist die Quadratwurzel des zweiten zentralen Moments der normierten zeitlichen Energiedichte:

$$\Delta_t^2 = \frac{1}{E_x} \int_{-\infty}^{\infty} (t - t_x)^2 |x(t)|^2 dt. \quad (5.7)$$

Analog ist die **Bandbreite** Δ_f als die Quadratwurzel des zweiten zentralen Moments der normierten spektralen Energiedichte definiert:

$$\Delta_f^2 = \frac{1}{E_x} \int_{-\infty}^{\infty} (f - f_x)^2 |X(f)|^2 df. \quad (5.8)$$

Anschaulich stellen die Momente (5.7) und (5.8) Maße für die Breite der Intervalle dar, in welchen sich ein wesentlicher Anteil der Energie des betrachteten Signals $x(t)$ im Zeitbereich bzw. im Frequenzbereich konzentriert.

5.2.2.4 Zeitdauer-Bandbreite-Produkt

Wird eine Funktion durch Skalierung mit einem Faktor $a < 1$ in der Zeit gestaucht, so wird die Zeitdauer kleiner. Gleichzeitig wird die Bandbreite entsprechend größer, da das Spektrum gestreckt wird (siehe Abschn. 5.2.1). Für die Zeitdauer $\Delta_{t,a}$ eines nach (5.1) skalierten Signals gilt

$$\Delta_{t,a}^2 = a^2 \Delta_t^2, \quad (5.9)$$

die Bandbreite ergibt sich zu

$$\Delta_{f,a}^2 = \frac{1}{a^2} \Delta_f^2. \quad (5.10)$$

Somit wird das **Zeitdauer-Bandbreite-Produkt** $\Delta_t \cdot \Delta_f$ durch die Skalierung nicht verändert.

Allgemein hängt das Zeitdauer-Bandbreite-Produkt ausschließlich von der Form der Funktion ab. Dabei gilt die Abschätzung [PJ15]:

$$\Delta_t \Delta_f \geq \frac{1}{4\pi}. \quad (5.11)$$

Gauß-Impuls

Das kleinste Zeitdauer-Bandbreite-Produkt wird erreicht, wenn in (5.11) das Gleichheitszeichen steht. Die Funktion, die diese Bedingung erfüllt, ist der **Gauß-Impuls**, der folgendermaßen angegeben werden kann [KSW08]:

$$g(t) = \frac{1}{\sqrt{\sqrt{2\pi}\Delta_t}} \exp\left(-\frac{t^2}{4\Delta_t^2}\right) \quad \circ \bullet \quad G(f) = \frac{1}{\sqrt{\sqrt{2\pi}\Delta_f}} \exp\left(-\frac{f^2}{4\Delta_f^2}\right). \quad (5.12)$$

5.3 Fourier-Transformation und Leistungsdichte

5.3.1 Diskrete Fourier-Transformation (DFT)

5.3.1.1 Graphische Veranschaulichung der DFT

In diesem Abschnitt sollen die Unterschiede zwischen der kontinuierlichen Fourier-Transformation und der DFT graphisch veranschaulicht werden [Bri95]. Die DFT wird dabei als bekannt vorausgesetzt [PJ15].

Die kontinuierliche **Fourier-Transformation**

$$X(f) = \int_{-\infty}^{\infty} x(t) \cdot e^{-j2\pi ft} dt \quad (5.13)$$

geht von drei idealisierenden Annahmen aus:

- 1) Die untersuchte Zeitfunktion $x(t)$ liegt kontinuierlich vor.
- 2) Das Signal $x(t)$ wird im Zeitbereich unendlich lange beobachtet.
- 3) Je nach Funktion $x(t)$ ist das zugehörige Spektrum $X(f)$ kontinuierlich oder diskret [PJ15].

In der Praxis wird die Fourier-Transformierte allerdings auf einem Digitalrechner berechnet, wodurch die oben genannten Annahmen nicht eingehalten werden können. Bei der Berechnung der DFT wird vielmehr wie folgt verfahren:

- 1) Das Zeitsignal $x(t)$ wird zur Weiterverarbeitung im Rechner abgetastet.
- 2) Die Zeitfunktion wird auf eine endliche Länge begrenzt, wodurch im Zeitbereich eine Fensterung vorgenommen wird.

- 3) Das Spektrum wird ebenfalls diskretisiert, da im Rechner nur diskrete Frequenzwerte dargestellt werden können. Dieser Schritt geht mit einer periodischen Fortsetzung des Zeitsignals einher.

Dies führt zur bekannten Definition der **diskreten Fourier-Transformation**:

$$X\left(\frac{n}{N t_A}\right) = \sum_{k=0}^{N-1} x(k t_A) \cdot e^{-j \frac{2 \pi k n}{N}}, \quad (5.14)$$

wobei t_A die Abtastzeit bezeichnet. Im Folgenden werden die einzelnen Schritte, die von der kontinuierlichen zur diskreten Fourier-Transformierten führen, graphisch veranschaulicht.

Schritt 1: Abtastung

Ausgegangen wird vom in Abb. 5.1 (a) dargestellten kontinuierlichen Transformationspaar. Für die Verarbeitung im Rechner wird die Zeitfunktion zunächst mit der Abtastfrequenz $f_A = 1/t_A$ abgetastet. Dabei wird das ursprüngliche Signal $x(t)$ wie folgt verändert:

$$x_1(t) := x(t) \cdot \underbrace{\sum_{k=-\infty}^{\infty} \delta(t - k t_A)}_{\Delta_0(t)} = \sum_{k=-\infty}^{\infty} x(k t_A) \delta(t - k t_A). \quad (5.15)$$

Dies führt im Frequenzbereich zu einer periodischen Fortsetzung des Spektrums, vgl. Abb. 5.1 (b)–(c).

Wird die Abtastfrequenz f_A unpassend gewählt, so kann es zu spektralen Überlappungen kommen. Ist die Abtastfrequenz kleiner oder gleich dem Doppelten der höchsten im Signal vorkommenden Frequenz, so kommt es zum **Bandüberlappungsfehler** (engl. *aliasing*) [PJ15]. Bei der Abtastung ist also darauf zu achten, dass das Signal bandbegrenzt ist und die Abtastfrequenz ausreichend hoch gewählt wird – ansonsten ist das Abtasttheorem verletzt und es entsteht Aliasing.

Schritt 2: Zeitbegrenzung

Die nächste Einschränkung besteht im zeitbegrenzten Beobachtungsfenster, welches man durch ein Rechteckfenster

$$r_{T_0}(t) = \begin{cases} 1 & \text{für } -T_0/2 \leq t < T_0/2 \\ 0 & \text{sonst} \end{cases} \quad (5.16)$$

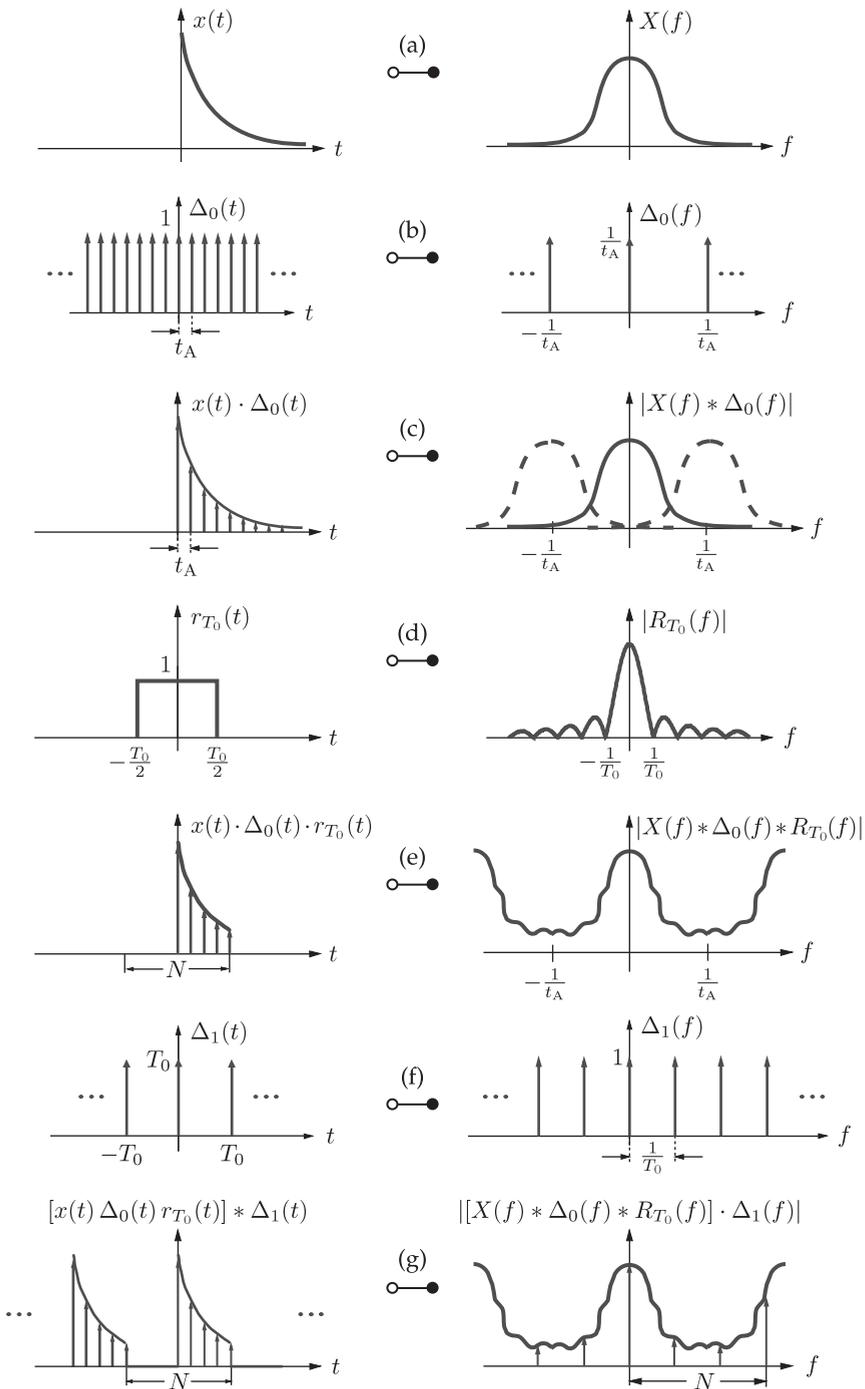


Abbildung 5.1. Anschauliche Herleitung eines diskreten Fourier-Paares [Bri95].

darstellen kann, vgl. Abb. 5.1 (d). Für die gefensterter Zeitfunktion gilt damit:

$$x_2(t) := \left(x(t) \cdot \sum_{k=-\infty}^{\infty} \delta(t - k t_A) \right) \cdot r_{T_0}(t) \quad (5.17)$$

$$= \sum_{k=0}^{N-1} x(k t_A) \delta(t - k t_A). \quad (5.18)$$

Die Summe wird endlich, da angenommen wird, dass in T_0 genau N äquidistante Abtast Schritte erfolgen. Es gilt dann für N :

$$N = \frac{T_0}{t_A} = \frac{\text{Beobachtungsdauer}}{\text{Abtastzeit}}. \quad (5.19)$$

Die Rechteckfensterung im Zeitbereich bleibt jedoch auch im Frequenzbereich nicht folgenlos. Sie bewirkt dort eine Faltung der Fourier-Transformierten mit einer Sinc-Funktion, was zu einer Verschmierung des Spektrums führt, vgl. Abb. 5.1 (d) und (e).

Schritt 3: Frequenzdiskretisierung

Die letzte Einschränkung der idealisierten Annahmen besteht in der Diskretisierung des Spektrums. Im Frequenzbereich entspricht dies einer Multiplikation mit einem Dirac-Impulskamm, im Zeitbereich erfolgt eine Faltung mit

$$\Delta_1(t) = T_0 \cdot \sum_{l=-\infty}^{\infty} \delta(t - l T_0). \quad (5.20)$$

Insgesamt wird das Zeitsignal also periodisch fortgesetzt, vgl. Abb. 5.1 (f) und (g). Mit (5.18) ergibt sich folgende Darstellung:

$$x_3(t) = [x(t) \Delta_0(t) r_{T_0}(t)] * \Delta_1(t) \quad (5.21)$$

$$= T_0 \cdot \sum_{l=-\infty}^{\infty} \left[\sum_{k=0}^{N-1} x(k t_A) \delta(t - k t_A - l T_0) \right]. \quad (5.22)$$

Dabei stellt k den Zählindex der Abtastzeitpunkte einer Periode dar; l steht für die unendlich vielen Perioden, die durch die Abtastung im Frequenzbereich entstehen.

Die Fourier-Transformierte des Signals in (5.21) sieht wie folgt aus:

$$X_3(f) = [X(f) * \Delta_0(f) * R_{T_0}(f)] \cdot \Delta_1(f). \quad (5.23)$$

Vergleicht man nun das ursprüngliche Transformationspaar $x(t) \circ \bullet X(f)$ mit den Gleichungen (5.21) und (5.23), so wird bewusst, wie stark die rechnergestützte Verarbeitung schon von vornherein Verfälschungen bewirkt. Der **Leckeffekt**, der durch die Fensterung im Zeitbereich entsteht, sowie der **Bandüberlappungsfehler** durch ei-

ne zu geringe Abtastfrequenz müssen durch entsprechende Maßnahmen minimiert bzw. verhindert werden.

5.3.1.2 Leckeffekt

Allgemein versteht man unter dem **Leckeffekt** die spektrale Verfälschung eines Signals aufgrund einer Fensterung im Zeitbereich. Wegen des Zusammenhangs

$$x(t) \cdot w(t) \quad \longleftrightarrow \quad X(f) * W(f) \quad (5.24)$$

entspricht die Fensterung im Zeitbereich einer Faltung mit dem Spektrum $W(f)$ der Fensterfunktion. So wird beispielsweise aus einem Dirac-Impuls $X(f) = \delta(f - f_0)$ das frequenzverschobene Spektrum des Fensters: $X(f) * W(f) = W(f - f_0)$.

Bei der DFT entsteht neben dem Leckeffekt infolge der endlichen Beobachtungsdauer eine periodische Wiederholung des Signals im Zeitbereich aufgrund der Abtastung des Spektrums. Prinzipiell gibt es dafür zwei Interpretationsmöglichkeiten:

- Die erste betrachtet das Signal im Frequenzbereich: Wird die als kontinuierlich betrachtete Hüllkurve des durch die endliche Beobachtungszeit mit $\sin(x)/x$ gewichteten Spektrums nicht genau im Maximum bzw. in den Nulldurchgängen abgetastet, sondern etwas daneben, dann entsteht ein spektraler Fehler α (Abb. 5.2 links).
- Im Zeitbereich lautet die Argumentation (anhand einer periodischen Funktion) wie folgt: Durch die Abtastung im Frequenzbereich wird die Funktion im Zeitbereich periodisch fortgesetzt. Beträgt nun die Beobachtungsdauer kein ganzzahliges Vielfaches der Periodendauer des kontinuierlichen, periodischen Signals, so kommt es am Fensterrand zu Unstetigkeitsstellen (Abb. 5.2 rechts). Bei der Fourier-Analyse bewirken diese Unstetigkeitsstellen ein Auftreten von Störanteilen, die das Spektrum verfälschen.

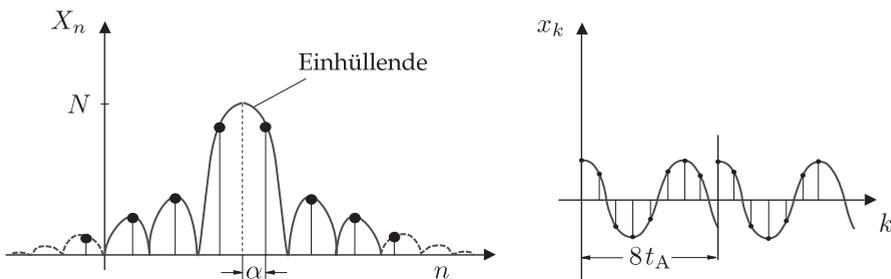


Abbildung 5.2. Leckeffekt: (links) Interpretation 1: die Einhüllende des Spektrums wird nicht im Maximum und in den Nulldurchgängen abgetastet; (rechts) Interpretation 2: periodische Fortsetzung im Zeitbereich durch Abtastung im Frequenzbereich ist unstetig.

5.3.1.3 Fensterung im Zeitbereich

Durch geschickte Wahl der Fensterfunktion zur Begrenzung der Beobachtungszeit kann der Leckeffekt verringert werden. Für die Anschauung kann man sich der Interpretation 2 aus dem vorigen Abschnitt bedienen. Durch ein Fenster, das an den Rändern „sanfter“ abfällt, werden die Unstetigkeitsstellen an den Fensterrändern der periodischen Fortsetzung gedämpft. Dadurch entstehen weniger hochfrequente Anteile im Spektrum des gefensternten Signals. Abbildung 5.3 zeigt einen Überblick über drei häufig verwendete Fensterfunktionen.

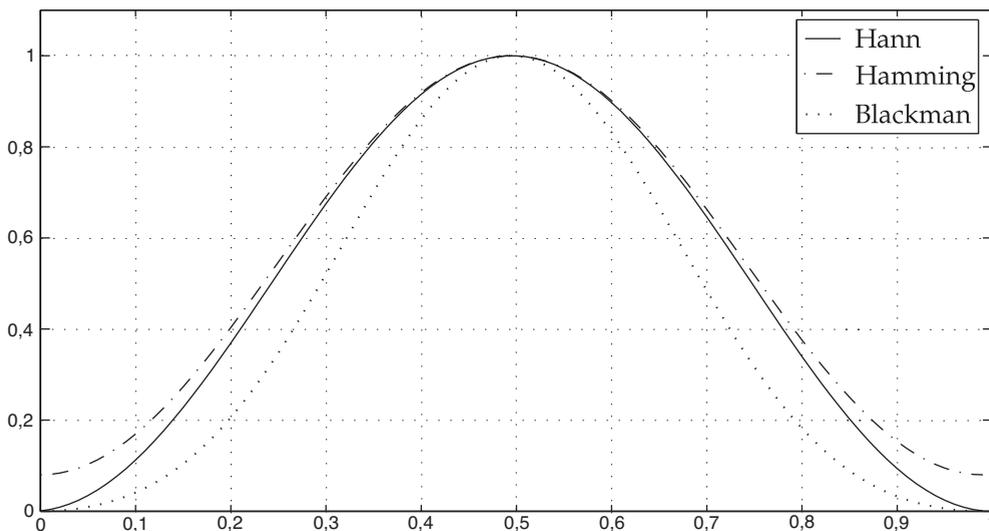


Abbildung 5.3. Geläufige Fensterfunktionen, Fensterbreite jeweils 256 Abtastwerte.

Hann-Fenster

Das **Hann-Fenster** ist wie folgt definiert:

$$w_n = \begin{cases} \frac{1}{2} \left(1 - \cos\left(\frac{2\pi n}{N}\right) \right) & \text{für } 0 \leq n \leq N \\ 0 & \text{sonst.} \end{cases} \quad (5.25)$$

Hamming-Fenster

Das **Hamming-Fenster** wurde zur Minimierung des ersten Nebenmaximums optimiert. Es fällt deshalb an den Fensterrändern nicht ganz auf null ab:

$$w_n = \begin{cases} 0,54 - 0,46 \cdot \cos\left(\frac{2\pi n}{N}\right) & \text{für } 0 \leq n \leq N \\ 0 & \text{sonst.} \end{cases} \quad (5.26)$$

Blackman-Fenster

Ein weiteres häufig verwendetes Fenster stellt das **Blackman-Fenster** dar. Es hat eine größere Sperrdämpfung als die oben betrachteten Fenster und ist wie folgt definiert:

$$w_n = \begin{cases} 0,42 - 0,5 \cos\left(2\pi \frac{n}{N}\right) + 0,08 \cos\left(4\pi \frac{n}{N}\right) & \text{für } 0 \leq n \leq N \\ 0 & \text{sonst.} \end{cases} \quad (5.27)$$

Kaiser-Fenster

Das **Kaiser-Fenster** hat gegenüber den bisher betrachteten Fenstern den Vorteil, dass die Fensterfunktion einen wählbaren Parameter enthält, der es ermöglicht, unterschiedliche Flankensteilheiten oder Sperrdämpfungen einzustellen. Das Fenster

$$w_n = \begin{cases} \frac{I_0\left(\beta \sqrt{1 - \left(1 - \frac{2}{N}n\right)^2}\right)}{I_0(\beta)} & \text{für } 0 \leq n \leq N \\ 0 & \text{sonst.} \end{cases} \quad (5.28)$$

enthält den Parameter β , mit dessen Hilfe dem Fenster die gewünschten Eigenschaften verliehen werden können. Die Funktion I_0 stellt dabei die Bessel-Funktion erster Ordnung dar. Abbildung 5.4 zeigt das Kaiser-Fenster für verschiedene β . Typische Werte liegen im Bereich zwischen 4 und 9.

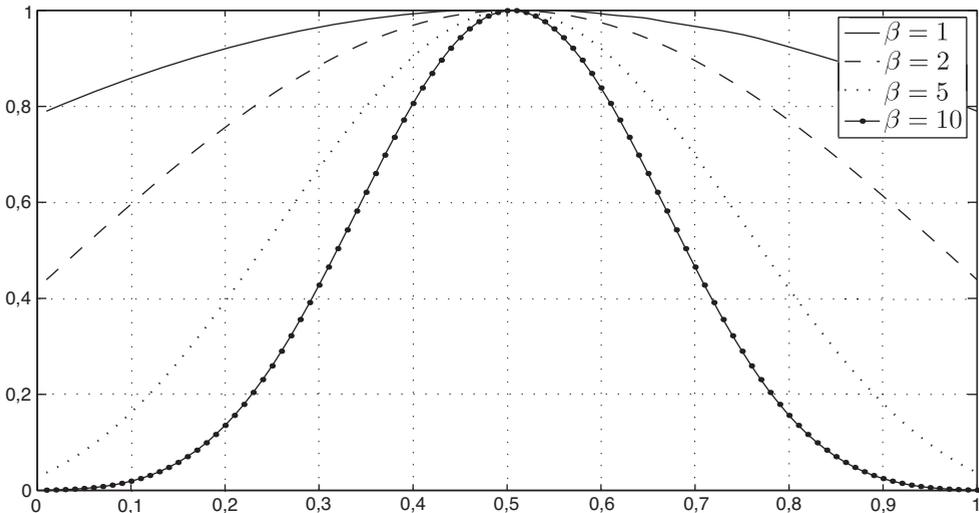


Abbildung 5.4. Kaiser-Fenster mit 100 Punkten für verschiedene Parameter β .

Für sehr genaue DFTs bedient man sich heute des **Dolph-Tschebyscheff-Fensters**. Die Dämpfung der Nebenmaxima im Spektrum wird hier durch ein breiteres Hauptmaximum als beim Rechteckfenster erkauft. Eine Vergrößerung der Fensterbreite bewirkt jedoch eine Verschmälerung des Hauptpeaks und stellt somit häufig

einen tragbaren Kompromiss dar. Dabei muss allerdings immer überprüft werden, ob die Randbedingungen, die durch die Zeitvarianz vorgegeben werden, noch erfüllt werden.

Für nähere Informationen zu diesem und auch zu anderen Fenstern sei auf [KK12] verwiesen.

5.3.2 Kurzzeit-Fourier-Transformation

5.3.2.1 Einführung

In vielen Anwendungsfällen ist man an einer gemischten Zeit- und Frequenzdarstellung eines Signals interessiert. Dies trifft besonders dann zu, wenn das untersuchte Signal nicht stationär ist, also einen zeitabhängigen Frequenzgehalt besitzt. Im vorliegenden Anwendungsfall trifft dies zu, da die menschliche Sprache ein stark instationäres Verhalten aufweist.

Um eine Darstellung des Spektrums über der Zeit zu bekommen, werden mehrere kurze Zeitfenster untersucht. Innerhalb eines Beobachtungsfensters wird das zu untersuchende Signal als stationär betrachtet und die Fourier-Transformation kann darauf angewendet werden.

Dies wird allerdings dadurch erkauft, dass sich bei einer kurzen Beobachtungszeit die Frequenzauflösung verschlechtert. Wählt man ein zu langes Beobachtungsintervall, um die Auflösung zu verbessern, so kann das beobachtete Signal im betrachteten Intervall nicht mehr als stationär betrachtet werden. Begründen lässt sich dieser Sachverhalt über das Zeitdauer-Bandbreite-Produkt (siehe Abschn. 5.2.2.4).

Bei der **Kurzzeit-Fourier-Transformation** (engl. *short-time Fourier transform*, STFT) versucht man dem Problem der Zeitvarianz dadurch zu begegnen, dass man das Signal zeitlich so fenstert, dass während der betrachteten Zeitintervalle Stationarität angenommen werden kann, was wiederum Grundlage für die Anwendung der FFT ist.

Im Hinblick auf den in Abschn. 5.3.1.2 angesprochenen Leckeffekt spielt die Wahl eines geeigneten Fensters eine wichtige Rolle. Das Gauß-Fenster hat dabei eine besondere Stellung, weshalb eine STFT auf Basis des Gauß-Fensters die eigene Bezeichnung **Gabor-Transformation** hat.

Definition

Die angesprochene zeitliche Fensterung des Signals lässt sich als Multiplikation der Funktion $x(t)$ mit dem Fenster $\gamma(t - \tau)$ darstellen:

$$x_{\gamma}(t, \tau) = x(t) \cdot \gamma^*(t - \tau). \quad (5.29)$$

Die Funktion ist abhängig von zwei Zeitvariablen t und τ . Die Variable τ gibt dabei an, an welcher Stelle sich das Zeitfenster befindet; t stellt die unabhängige Zeitvariable des Signals x dar.

Die Definition der STFT entspricht der Fourier-Transformation des gefensternten Signals (5.29):

$$F_x^\gamma(\tau, f) = \int_{-\infty}^{\infty} x_\gamma(t, \tau) e^{-j2\pi ft} dt = \int_{-\infty}^{\infty} x(t) \gamma^*(t - \tau) e^{-j2\pi ft} dt. \quad (5.30)$$

Kurzzeitspektrum und Spektrogramm

Das **Kurzzeitspektrum** ist eine zweidimensionale Funktion von Frequenz und Zeitfensterposition. Da das Spektrum komplex ist, benötigt man zur graphischen Darstellung für Betrag und Phase jeweils ein dreidimensionales Bild.

Häufig beschränkt man sich dabei jedoch auf den Betrag des Spektrums oder auf dessen Quadrat. Das **Spektrogramm** stellt das Betragsquadrat der Kurzzeit-Fourier-Transformierten dar:

$$S_x^\gamma(\tau, f) = |F_x^\gamma(\tau, f)|^2 = \left| \int_{-\infty}^{\infty} x(t) \gamma^*(t - \tau) e^{-j2\pi ft} dt \right|^2. \quad (5.31)$$

Eigenschaften der STFT

– **Unschärfepinzipp:**

Die STFT unterliegt dem Unschärfepinzipp, es muss also ein Kompromiss bezüglich der Auflösung der Zeit und Frequenz eingegangen werden (siehe Abb. 5.5).

– **Abhängigkeit von der Anzahl der Abtastpunkte:**

Frequenz- und Zeitauflösung sind abhängig von der Anzahl der Abtastpunkte N , welche häufig durch Randbedingungen wie Stationarität festgelegt werden.

– **Verschiebungsinvarianz:**

– Zeitverschiebung:

Eine Verschiebung des Zeitsignals mittels $x(t) \rightarrow x(t - t_0)$ führt zu einer Verschiebung der Kurzzeit-Fourier-Transformierten um t_0 . Die Transformierte bleibt bis auf einen Modulationsfaktor gleich.

– Modulation:

Eine Modulation des Signals durch $x(t) \rightarrow x(t) e^{j2\pi f_0 t}$ führt in Richtung der Frequenzachse zu einer Verschiebung der Kurzzeit-Fourier-Transformierten um f_0 .

5.3.3 Zeitdiskretisierung

Bei der rechnergestützten Datenverarbeitung sind die Kurzzeit-Fourier-Transformation und das Spektrogramm in der zeitkontinuierlichen Form nicht berechenbar. Der

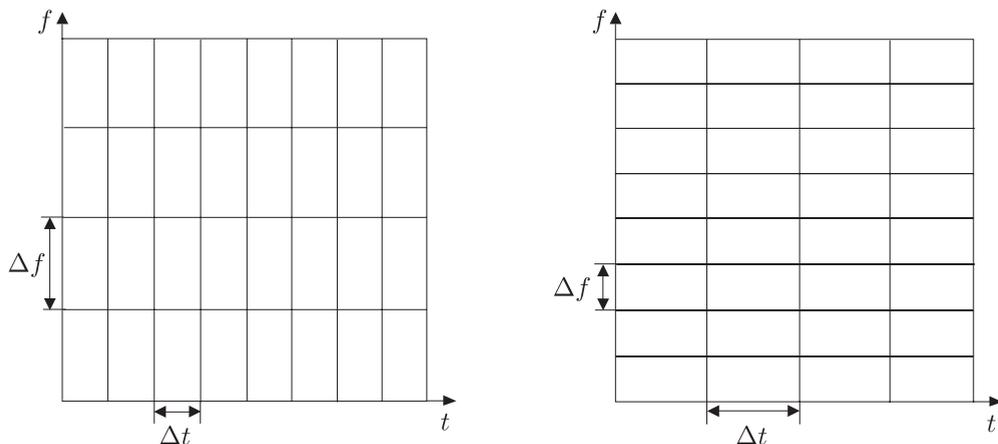


Abbildung 5.5. Kompromiss der Zeit-Frequenz-Auflösung: Wird das im linken Bildteil vorgegebene Frequenzraster verfeinert, so vergrößert sich entsprechend das Zeitraster.

Übergang zur zeitdiskreten Darstellung ergibt:

$$F_x^\gamma[m, k] = \text{DFT}_n \{ x_n \gamma_{n-m}^* \} \tag{5.32}$$

$$= \sum_{n=m-N/2}^{m+N/2-1} x_n \gamma_{n-m}^* e^{-j2\pi kn/N} . \tag{5.33}$$

Die Beschränkung auf diskrete Frequenzen entspricht periodischen Wiederholungen des Signals im Zeitbereich. Damit dies keine Überlappungen verursacht (analoges Phänomen zu Aliasing durch Abtastung im Zeitbereich), muss die Fensterfunktion zeitlich begrenzt sein:

$$\gamma_{n-m} = 0, \quad \text{falls } n - m > N/2 - 1 \quad \text{oder} \quad n - m < -N/2. \tag{5.34}$$

Aus der diskreten Kurzzeit-Fourier-Transformierten kann das zeitdiskrete Spektrogramm

$$S_x^\gamma[m, k] = \frac{1}{N} |F_x^\gamma[m, k]|^2 \tag{5.35}$$

berechnet werden.

5.4 Wavelet-Transformation

5.4.1 Kontinuierliche Wavelet-Transformation

Die Geschichte der **Wavelet-Transformation** ist noch relativ jung. Erst seit Beginn der 80er-Jahre wird das Verfahren für die Signalverarbeitung verwendet und hat in der Audio- und Bildcodierung, der numerischen Integration und in der Mustererkennung Einsatzmöglichkeiten gefunden [Mer13]. Die kontinuierliche Wavelet-Transformation kann aus der Idee der Kurzzeit-Fourier-Transformation (STFT) entwickelt werden.

5.4.1.1 Definition

Bisher wurde die STFT als Fourier-Transformierte des gefensterten Signals gedeutet. Mit Hilfe der Innenproduktschreibweise ist jedoch auch eine andere Interpretation möglich:

$$F_x^\gamma(\tau, f) = \int_{-\infty}^{\infty} x(t) \gamma^*(t - \tau) e^{-j2\pi ft} dt = \langle x(t), \gamma(t - \tau) e^{j2\pi ft} \rangle. \quad (5.36)$$

Bei der STFT wird also das Signal $x(t)$ mit dem zeit- und frequenzverschobenen Fenster $\gamma(t - \tau) e^{j2\pi ft}$ verglichen.

Das Analysefenster $\gamma(t)$ hat die mittlere Zeit 0, die mittlere Frequenz 0, die Zeitdauer Δ_t und die Bandbreite Δ_f . Das bedeutet, dass sich die meiste Signalenergie im Zeitintervall $[-\Delta_t, \Delta_t]$ bzw. im Frequenzintervall $[-\Delta_f, \Delta_f]$ konzentriert. In der Zeit-Frequenz-Ebene lässt sich dieser Bereich in Form eines Rechtecks darstellen. Durch die Zeit- und Frequenzverschiebung verschiebt sich der Mittelpunkt dieses Rechtecks vom Ursprung zu den Koordinaten (τ, f) .

Aus diesen Überlegungen folgt, dass sich die Signalenergie des zeit- und frequenzverschobenen Fensters um die mittlere Zeit τ und die mittlere Frequenz f konzentriert, mit der Zeitdauer Δ_t und der Bandbreite Δ_f . Das Ergebnis des Vergleichs mit dem Signal $x(t)$ ist daher eine Maßzahl dafür, wie viel Energie das Signal $x(t)$ in diesem Bereich enthält. Das bedeutet, dass die Zeitdauer und Bandbreite des Analysefensters $\gamma(t)$ eine Unschärfe des STFT-Spektrums bewirkt. Anders formuliert: Aus $F_x^\gamma(\tau, f) \neq 0$ kann nicht geschlossen werden, dass das Signal $x(t)$ exakt zum Zeitpunkt τ die Frequenz f enthält. Diese Tatsache gilt bei der Wavelet-Transformation ebenfalls.

Bei der Wavelet-Transformation wird das Analysefenster $\psi(t)$ (die Wavelet-Funktion) zeitverschoben und *skaliert*:

$$W_x^\psi(a, b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} x(t) \psi^*\left(\frac{t-b}{a}\right) dt = \left\langle x(t), \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right) \right\rangle. \quad (5.37)$$

Hierbei wird vorausgesetzt, dass die Wavelet-Funktion $\psi(t)$ ein Bandpasssignal ist, d. h. sie besitzt eine von null verschiedene mittlere Frequenz f_ψ .

Durch die Skalierung a und die Zeitverschiebung b ändern sich sowohl die mittlere Zeit und die mittlere Frequenz des Wavelets als auch die Zeitdauer und die Bandbreite. Hierbei gelten folgende Zusammenhänge, wobei jeweils die Größen auf der rechten Seite der Gleichung die unskalierten und nicht zeitverschobenen Wavelets $\psi(t)$ bezeichnen:

$$\text{mittlere Zeit:} \quad t_{\psi a,b} = at_\psi + b,$$

$$\text{mittlere Frequenz:} \quad f_{\psi a,b} = \frac{f_\psi}{a},$$

$$\text{Zeitdauer:} \quad \Delta_{t_{\psi a,b}}^2 = a^2 \Delta_{t_\psi}^2,$$

$$\text{Bandbreite:} \quad \Delta_{f_{\psi a,b}}^2 = \frac{1}{a^2} \Delta_{f_\psi}^2.$$

Somit ist $W_x^\psi(a, b)$ eine Maßzahl dafür, wie viel Energie das Signal $x(t)$ im Bereich der Zeit $t_{\psi a,b}$ und der Frequenz $f_{\psi a,b}$ enthält. Die Unschärfe wird durch die Zeitdauer $\Delta_{t_{a,b}}$ und die Bandbreite $\Delta_{f_{a,b}}$ bestimmt.

Die Darstellung in der Zeit-Skalierungs-Ebene ist nicht besonders anschaulich. Um ein zeitabhängiges Spektrum zu erhalten, das mit dem STFT-Spektrum vergleichbar ist, kann die Wavelet-Transformierte mit Hilfe der Definitionen $\tau := t_{\psi a,b}$ und $f := f_{\psi a,b}$ in eine Zeit-Frequenz-Darstellung überführt werden. Mit den Zusammenhängen

$$a = \frac{f_\psi}{f}, \quad b = \tau - \frac{f_\psi}{f} \cdot t_\psi, \quad \frac{t-b}{a} = \frac{f}{f_\psi}(t-\tau) + t_\psi \quad (5.38)$$

folgt

$$W_x^\psi(\tau, f) = \left\langle x(t), \sqrt{\left| \frac{f}{f_\psi} \right|} \cdot \psi\left(\frac{f}{f_\psi}(t-\tau) + t_\psi\right) \right\rangle \quad (5.39)$$

$$= \sqrt{\left| \frac{f}{f_\psi} \right|} \int_{-\infty}^{\infty} x(t) \cdot \psi^*\left(\frac{f}{f_\psi}(t-\tau) + t_\psi\right) dt. \quad (5.40)$$

Der wesentliche Unterschied zur STFT besteht nun darin, dass die Zeitdauer und Bandbreite (und damit die Unschärfe des Spektrums) von der Frequenz abhängen:

$$\Delta_t^2(f) = \left(\frac{f_\psi}{f}\right)^2 \Delta_t^2, \quad \Delta_f^2(f) = \left(\frac{f}{f_\psi}\right)^2 \Delta_f^2. \quad (5.41)$$

Je höher die untersuchte Frequenz, desto besser ist die Zeitauflösung und desto schlechter die Frequenzauflösung (siehe Abb. 5.6). Dies ist eine sinnvolle Anpassung, wie folgendes Beispiel zeigt.

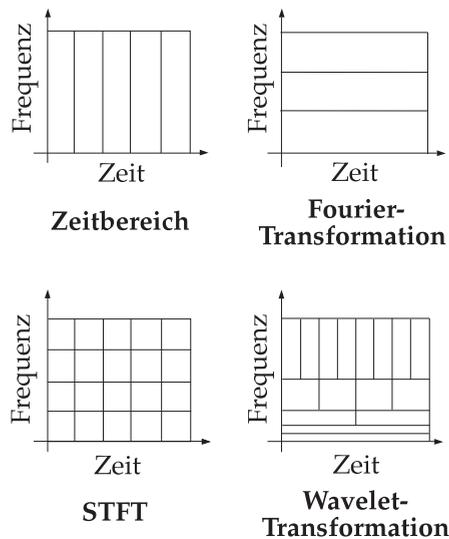


Abbildung 5.6. Gegenüberstellung von Zeitbereich, Frequenzbereich, Zeit-Frequenz-Bereich und Zeitverschiebungs-Skalierungsbereich.

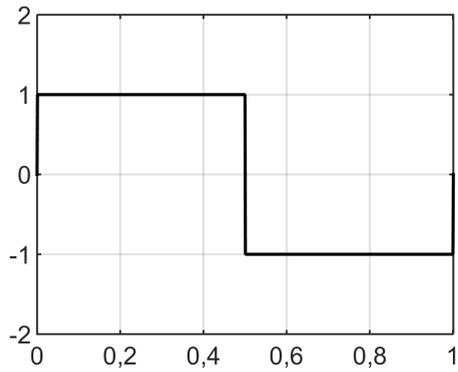
Angenommen, es wird ein Signal untersucht, das Frequenzen der Größenordnung 1 Hz und 1 kHz enthält. Das Zeit-Frequenz-Spektrum sollte in der Lage sein, eine Veränderung des niederfrequenten Anteils um beispielsweise 0,5 Hz aufzulösen. Verändert sich dagegen der hochfrequente Anteil um diese Frequenz, ist diese Information weniger relevant. Dafür müssen für hohe Frequenzen zeitliche Veränderungen besser aufgelöst werden, da hier viel kürzere Periodendauern vorliegen.

Ein weiterer Vorteil der hohen Zeitauflösung bei hohen Frequenzen besteht darin, dass Sprünge im Signal zeitlich besser lokalisiert werden können.

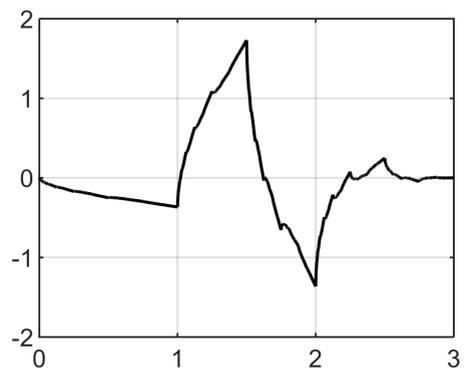
5.4.1.2 Wavelets

Wavelets besitzen eine irreguläre und meist asymmetrische Wellenform von begrenzter Zeitdauer, ihr zeitlicher Mittelwert ist 0 und ihre Energie auf 1 normiert. Sie lassen sich in Wavelet-Familien gliedern, die im Folgenden anhand einiger Beispiele vorgestellt werden.

- **Daubechies-Wavelets:** Diese zehn Wavelets umfassende Familie wurde von Ingrid Daubechies entwickelt (Abb. 5.7). Ein spezielles Daubechies-Wavelet ist das in Abb. 5.7(a) gezeigte Daubechies(1)- oder auch Haar-Wavelet.
- **Biorthogonale Wavelets:** Diese Familie besteht aus Wavelet-Paaren, wobei für Analyse (Dekomposition) ein Wavelet und für die Synthese (Rekonstruktion) ein anderes Wavelet verwendet wird.



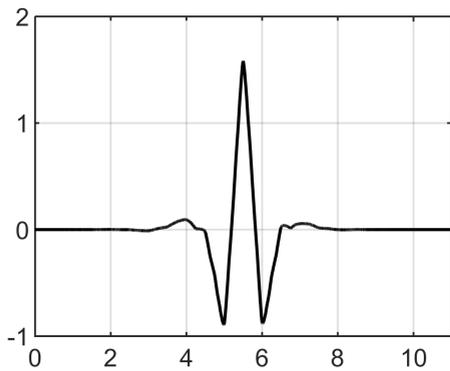
(a) Daubechies(1)-Wavelet (Haar-Wavelet)



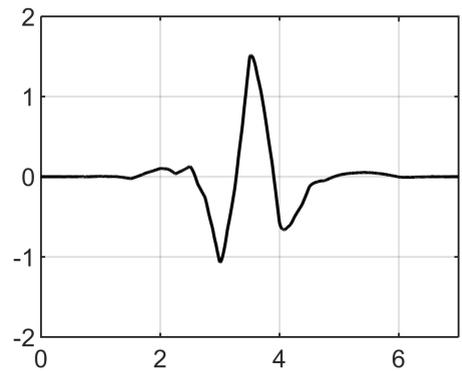
(b) Daubechies(2)-Wavelet

Abbildung 5.7. Daubechies-Wavelets.

– **Coiflets und Symlets:** Beispiele für Wavelets dieser Familie sind in Abbildung 5.8 gezeigt.



(a) Coiflet



(b) Symlet

Abbildung 5.8. Weitere Wavelets.

5.4.2 Diskrete Wavelet-Transformation

5.4.2.1 Einführung

Auch bei der Wavelet-Transformation können natürlich nicht alle unendlich vielen Wavelet-Koeffizienten für beliebig genaue Skalierungen und Zeitverschiebungen berechnet werden. Man behilft sich deshalb damit, dass man Skalierungsfaktoren und Zeitverschiebungen wählt, die auf Potenzen von zwei beruhen und nennt diese *dyadische* Skalierungen und Positionen. Diese Methode ist weit effizienter, jedoch trotzdem

sehr genau. Mallat implementierte 1988 ein darauf beruhendes Filterkonzept, das zur so genannten *fast Wavelet transform* geführt hat.

5.4.2.2 Approximationen und Details

Häufig ist der niederfrequente Anteil eines Signals der wichtigere, der hochfrequente liefert nur noch Nuancen (Beispiel: Sprache).

Nimmt man ein bestehendes Signal und führt nacheinander eine Tief- und eine Bandpassfilterung durch, so erhält man zum einen die so genannten *Approximationen*, welche den für ein großes a gewonnenen niederfrequenten Anteil des Signals repräsentieren, zum anderen bekommt man die *Details*, die den hochfrequenten Signalanteil für kleine a darstellen (Abb. 5.9).

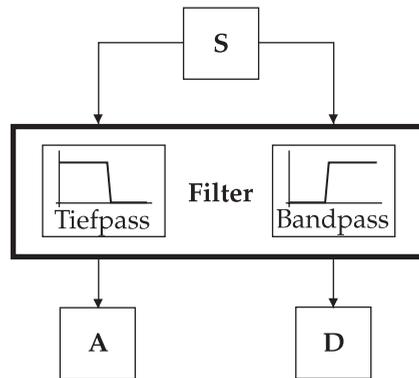


Abbildung 5.9. Aufspaltung des ursprünglichen Signals in Approximation und Details durch Filterung.

Durch die beiden Filterungen erhält man zwei Signale, die die gleiche Anzahl an Datenelementen enthalten wie das ursprüngliche Signal. Um die daraus folgende Datenexplosion – besonders bei mehrmaligem Anwenden dieser Strategie – zu verhindern, werden Approximationen und Details um den Faktor 2 unterabgetastet. Es wird also jeder zweite Abtastwert verworfen.

5.4.2.3 Multiraten-Filterung

Wie bereits im vorigen Abschnitt angedeutet, wird die fortlaufende Aufspaltung eines Signals in Tief- und Bandpasssignal wenn nötig weiter fortgesetzt. Daraus resultiert ein Baum mit einem langen nach links unten verlaufenden Ast mit Approximationen und Details der niedrigen Frequenzen (Abb. 5.10).

Da der Analyseprozess iterativ verläuft, bedarf es einer Abbruchbedingung zum Beispiel durch eine vorgegebene endliche Anzahl von Durchläufen oder wenn die erhaltenen Daten sich von denen der vorigen Ebene nicht mehr ausreichend unterscheiden (ϵ -Kriterium).

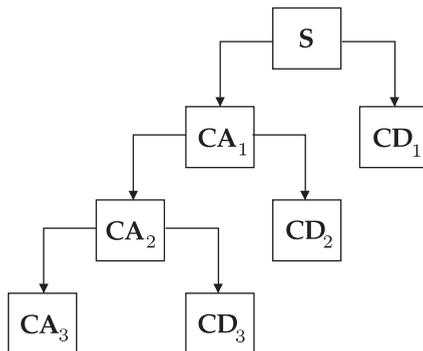


Abbildung 5.10. Wavelet-Dekompositionsbaum bei mehrfacher Aufspaltung in Approximationen und Details.

5.4.3 Wavelet-Rekonstruktion

Bei der Rekonstruktion findet der zur Analyse umgekehrte Prozess statt. Aus einem Satz von Wavelet-Koeffizienten soll das ursprüngliche Signal synthetisiert werden, also eine *inverse diskrete Wavelet-Transformation* durchgeführt werden. Problematisch für die Rekonstruktion ist das bei der Multiraten-Filterung angewendete *Downsampling*. Hier treten Aliasing-Effekte auf, also spektrale Verfälschungen. Es müssen sehr leistungsfähige und anspruchsvolle Filter angewendet werden, um das Signal wieder korrekt herstellen zu können (Abb. 5.11). Die Koeffizienten dieser sogenannten **Zwei-Kanal-Filterbank** werden dabei so gewählt, dass die daraus bestimmten Wavelets und Skalierungsfunktionen die gewünschten Eigenschaften besitzen [Mer13].

Durch iteratives Upsampeln und Falten des Bandpassfilters mit sich selbst erhält man die entsprechende **Wavelet-Funktion**. Durch iteratives Upsampeln und Falten des Tiefpassfilters mit sich selbst bekommt man die **Skalierungsfunktionen**. Dies sind die Basisfunktionen, mit deren Hilfe ein Signal ähnlich wie mit den harmonischen Funktionen bei Fourier-Reihen wiederhergestellt werden kann [KSW08].

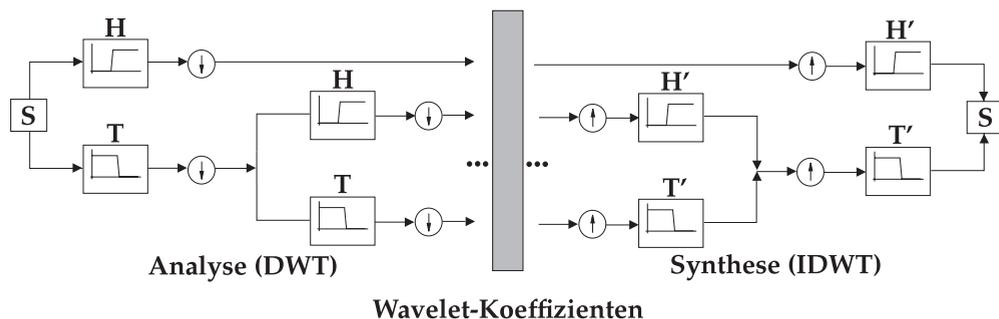


Abbildung 5.11. Wavelet-Dekompositionsbaum bei mehrfacher Aufspaltung in Approximationen und Details.

5.5 Wigner-Ville-Verteilung

Die STFT und die Wavelet-Transformation zur Analyse eines nichtstationären Signals gehen von der Annahme aus, dass sich der Frequenzgehalt eines Signals während des betrachteten kurzen Zeitintervalls nicht oder nur unwesentlich ändert.

Die Breite des Signalfensters ist jedoch durch die zeitliche Variabilität des Signals festgelegt und beeinflusst damit die Frequenzauflösung.

Die **Wigner-Ville-Verteilung** (WVV) stellt eine Zeit-Frequenz-Verteilung dar, die diesen Nachteil nicht aufweist. Die im nächsten Abschnitt vorgestellte **Ambiguitätsfunktion** und die Wigner-Ville-Verteilung werden nicht wie STFT und Wavelet-Transformation aus einer linearen Filterung gewonnen und sind in ihrer Auflösung nicht durch die Unschärferelation eingeschränkt [Mer13].

5.5.1 Ambiguitätsfunktion

Die Ambiguitätsfunktion wird hier vorgestellt, weil aus ihr recht anschaulich die später besprochene Interpretation der Wigner-Ville-Verteilung als Zeit-Frequenz-Energieverteilung abgeleitet werden kann. Die Ambiguitätsfunktion beschreibt die Ähnlichkeit eines zeit- und frequenzverschobenen Signals mit dem ursprünglichen Signal, stellt also eine Zeit-Frequenz-Autokorrelationsfunktion dar:

$$A_{xx}(\tau, \vartheta) = \int_{-\infty}^{\infty} x\left(t + \frac{\tau}{2}\right) x^*\left(t - \frac{\tau}{2}\right) e^{j2\pi\vartheta t} dt \quad (5.42)$$

$$= \int_{-\infty}^{\infty} X\left(f - \frac{\vartheta}{2}\right) X^*\left(f + \frac{\vartheta}{2}\right) e^{j2\pi f\tau} df. \quad (5.43)$$

Hierbei beschreibt der Parameter τ die Zeitverschiebung und ϑ die Frequenzverschiebung. Im Gegensatz zur eindimensionalen Autokorrelationsfunktion (AKF) bezüglich der Zeit ist mit der Ambiguitätsfunktion eine Laufzeit- und damit Abstandsbestimmung bei gleichzeitiger Geschwindigkeitsbestimmung (Frequenzverschiebung) eines Objektes möglich, z. B. in der Radartechnik.

Anders als bei der eindimensionalen AKF bezüglich der Zeit ist es hier üblich, die Zeit- und Frequenzverschiebungen auf beide Signalanteile $x(t)$ und $x^*(t)$ gleichmäßig zu verteilen. Somit wird zum Beispiel die Zeitverschiebung auf beide Signalanteile aufgeteilt, wodurch $x(t)$ und $x^*(t)$ nur noch um $\tau/2$ verschoben werden.

5.5.2 Definition der Wigner-Ville-Verteilung

Aus der Ambiguitätsfunktion erhält man die bekannte zeitliche AKF, indem man die Frequenzverschiebung zu null setzt:

$$r_{xx}^E(\tau) = A_{xx}(\tau, 0). \quad (5.44)$$

Die Fourier-Transformierte der AKF liefert dabei das Energiedichtespektrum:

$$S_{xx}^E(f) = \int_{-\infty}^{\infty} r_{xx}^E(\tau) e^{-j2\pi f\tau} d\tau = \int_{-\infty}^{\infty} A_{xx}(\tau, 0) e^{-j2\pi f\tau} d\tau. \quad (5.45)$$

Gleichermaßen lässt sich die zeitliche Energiedichte als die Fourier-Transformierte der AKF des Spektrums $X(f)$ aus $A_{xx}(0, \vartheta)$ bestimmen über

$$s_{xx}^E(t) = \int_{-\infty}^{\infty} A_{xx}(0, \vartheta) e^{-j2\pi\vartheta t} d\vartheta. \quad (5.46)$$

Möchte man keinen der beiden Parameter festhalten und eine allgemeine Verteilung für verschiedene Zeit- und Frequenzverschiebungen erhalten, so muss die Ambiguitätsfunktion gleichzeitig bezüglich der Parameter Frequenzverschiebung ϑ und Zeitverschiebung τ Fourier-transformiert werden:

$$W_{xx}(t, f) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} A_{xx}(\tau, \vartheta) e^{-j2\pi\vartheta t} e^{-j2\pi f\tau} d\vartheta d\tau. \quad (5.47)$$

Dies ist die Definition der Wigner-Ville-Verteilung anhand der Ambiguitätsfunktion. Aufgrund ihrer Definition als *zweifache Fourier-Transformierte* der Zeit-Frequenz-AKF wird der Charakter der Wigner-Ville-Verteilung als Zeit-Frequenz-Energieverteilung deutlich.

In der Literatur findet man häufig andere Definitionen für die Wigner-Ville-Verteilung. Führt man zunächst je eine der beiden Fourier-Transformationen aus (5.47) durch, so entstehen als Zwischenfunktionen nach der Fourier-Transformation bezüglich θ die **temporäre AKF des Zeitsignals** $\phi_{xx}(t, \tau)$,

$$\phi_{xx}(t, \tau) = \int_{-\infty}^{\infty} A_{xx}(\tau, \vartheta) e^{-j2\pi\vartheta t} d\vartheta = x\left(t + \frac{\tau}{2}\right) x^*\left(t - \frac{\tau}{2}\right), \quad (5.48)$$

nach der Fourier-Transformation bezüglich τ die **temporäre AKF des Spektrums** $\Phi_{xx}(\vartheta, f)$

$$\Phi_{xx}(\vartheta, f) = \int_{-\infty}^{\infty} A_{xx}(\tau, \vartheta) e^{-j2\pi f\tau} d\tau = X\left(f - \frac{\vartheta}{2}\right) X^*\left(f + \frac{\vartheta}{2}\right). \quad (5.49)$$

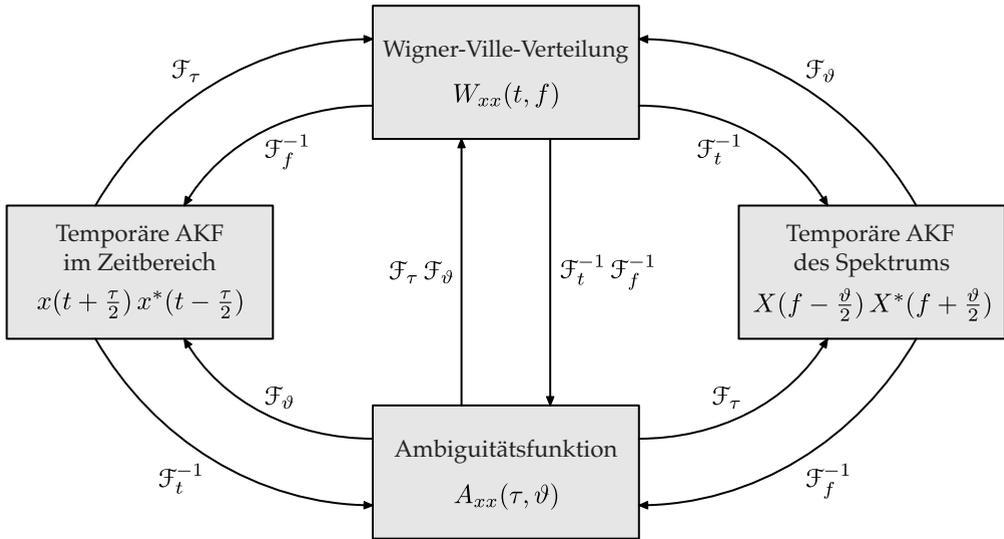


Abbildung 5.12. Zusammenhang zwischen Ambiguitätsfunktion, Wigner-Ville-Verteilung und den beiden temporären AKFs.

Die Fourier-Transformierte bezüglich des jeweils anderen Parameters liefert schließlich die in der Literatur gängige Definition der Wigner-Ville-Verteilung:

$$W_{xx}(t, f) = \int_{-\infty}^{\infty} x\left(t + \frac{\tau}{2}\right) x^*\left(t - \frac{\tau}{2}\right) e^{-j2\pi f\tau} d\tau \tag{5.50}$$

$$= \int_{-\infty}^{\infty} X\left(f - \frac{\vartheta}{2}\right) X^*\left(f + \frac{\vartheta}{2}\right) e^{-j2\pi\vartheta t} d\vartheta. \tag{5.51}$$

Abbildung 5.12 veranschaulicht die Zusammenhänge.

5.5.3 Eigenschaften der Wigner-Ville-Verteilung

a) Die Integration der WVV über alle τ liefert das zeitunabhängige Energiedichtespektrum:

$$S_{xx}^E(f) = \int_{-\infty}^{\infty} W_{xx}(t, f) dt = |X(f)|^2. \tag{5.52}$$

b) Die Integration der WVV über alle f liefert die momentane Energiedichte:

$$s_{xx}^E(t) = \int_{-\infty}^{\infty} W_{xx}(t, f) df = |x(t)|^2. \tag{5.53}$$

- c) Wie bereits im Zusammenhang mit der Ambiguitätsfunktion gezeigt, liefert die Integration über τ und f die gesamte Signalenergie:

$$E_x = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W_{xx}(t, f) dt df. \quad (5.54)$$

- d) Die Wigner-Ville-Verteilung stellt kein Energiedichtespektrum dar, da sie auch negative Werte enthalten kann.
- e) Eine Rekonstruktion des Zeitsignals aus der Wigner-Ville-Verteilung ist nur bis auf einen konstanten Modulationsfaktor $e^{j\varphi}$, $0 \leq \varphi < 2\pi$, möglich.
- f) Die WVV liefert einen einfachen Schätzer für die Momentanfrequenz eines Signals:

$$f_x(t) = \frac{\int_{-\infty}^{\infty} f \cdot W_{xx}(t, f) df}{\int_{-\infty}^{\infty} W_{xx}(t, f) df}. \quad (5.55)$$

- g) Eine Zeit- und eine Frequenzverschiebung des Zeitsignals bewirkt eine gleichzeitige Zeit- und Frequenzverschiebung der entsprechenden Wigner-Ville-Verteilung. Bei gleichzeitiger Zeitverschiebung und Modulation ergibt sich also:

$$\tilde{x}(t) = x(t - t_0) e^{j2\pi f_0 t} \Rightarrow W_{\tilde{x}\tilde{x}}(t, f) = W_{xx}(t - t_0, f - f_0). \quad (5.56)$$

- h) Die Auto-Wigner-Ville-Verteilung ist immer reell.

- i) **Multiplikation im Zeitbereich:** Bei gegebenem $\tilde{x}(t) = x(t) h(t)$ gilt für $W_{\tilde{x}\tilde{x}}(t, f)$:

$$W_{\tilde{x}\tilde{x}}(t, f) = \int_{-\infty}^{\infty} \phi_{xx}(t, \tau) \phi_{hh}(t, \tau) e^{-j2\pi f\tau} d\tau. \quad (5.57)$$

Führt man statt der Multiplikation im Zeitbereich eine Faltung im Frequenzbereich durch, so gilt:

$$W_{\tilde{x}\tilde{x}}(t, f) = W_{xx}(t, f) *_f W_{hh}(t, f). \quad (5.58)$$

Eine Multiplikation im Zeitbereich entspricht also einer Faltung der Wigner-Ville-Verteilungen bezüglich f .

- j) **Filterung eines Signals:** Die Filterung eines Signals $x(t)$ im Zeitbereich mit einem linearen Filter $h(t)$ wird durch die Faltung realisiert ($\tilde{x}(t) = x(t) * h(t)$), was eine Multiplikation der Spektren $X(f)$ und $H(f)$ bewirkt. Dies führt für die Wigner-Ville-Verteilungen zu einer Faltung bezüglich der Zeit t :

$$W_{\tilde{x}\tilde{x}}(t, f) = W_{xx}(t, f) *_t W_{hh}(t, f). \quad (5.59)$$

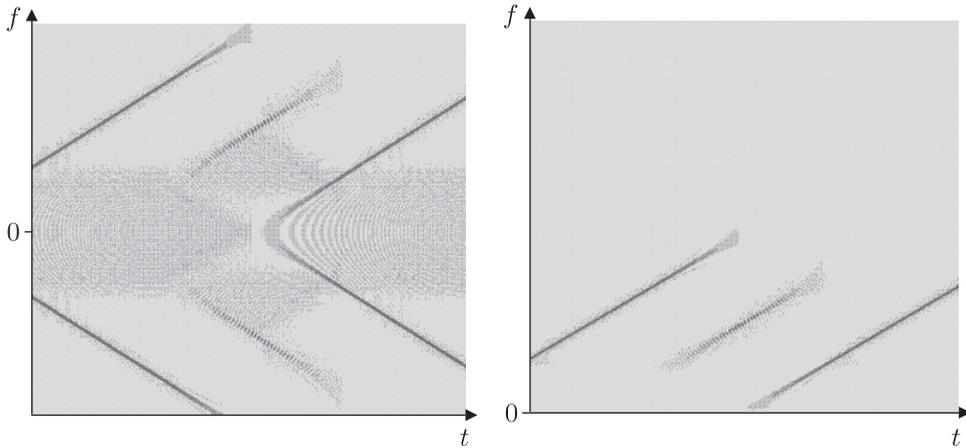


Abbildung 5.13. Wigner-Ville-Spektrum eines Sweep-Signals: (links) Berechnung mit dem reellen Zeitsignal; (rechts) Berechnung mit dessen analytischem Signal.

5.5.4 Kreuzterme

Im Gegensatz zur Kurzzeit-Fourier-Transformation und zur Wavelet-Transformation ist die Wigner-Ville-Verteilung keine lineare, sondern eine quadratische Integraltransformation. Entsprechend ergibt die Wigner-Ville-Verteilung der Summe zweier Signale nicht einfach die Summe der Wigner-Ville-Verteilungen der einzelnen Signale. Vielmehr gilt für ein gegebenes Signal $x(t) = x_1(t) + x_2(t)$:

$$W_{xx}(t, f) = W_{x_1x_1}(t, f) + W_{x_2x_2}(t, f) + 2 \operatorname{Re}\{W_{x_1x_2}(t, f)\}. \quad (5.60)$$

Der letzte Summand in (5.60) wird **Kreuzterm** oder **Interferenzterm** genannt.

Reelle Signale besitzen ein symmetrisches Spektrum. Hier entstehen grundsätzlich Kreuzterme zwischen den positiven und negativen Frequenzanteilen. Dies wird anhand von Abb. 5.13 deutlich. Das linke Bild zeigt die Wigner-Ville-Verteilung eines reellen Chirp-Signals. Das Chirp-Signal ist ein linear frequenzmoduliertes Signal, d. h. seine Frequenz steigt linear mit der Zeit. Die Kreuzterme sind als Störungen im niederfrequenten Bereich zu erkennen.

Eine Möglichkeit zur Reduzierung der Störungen besteht in der Berechnung der Wigner-Ville-Verteilung des **analytischen Signals**. Das analytische Signal zeichnet sich dadurch aus, dass es keine negativen Frequenzanteile enthält. Man erhält das analytische Signal, indem man zum ursprünglichen Signal $x(t)$ dessen **Hilbert-Transformierte** als Imaginärteil addiert:

$$x_a(t) = x(t) + j \mathcal{H}\{x(t)\}. \quad (5.61)$$

Näheres zur Hilbert-Transformation kann in [PJ15] nachgelesen werden.

In Abb. 5.13 rechts sind keine negativen Frequenzanteile enthalten. Hierbei handelt es sich um die Wigner-Ville-Verteilung des zum Sweep-Signal gehörenden analytischen Signals. Dadurch werden Kreuzterme deutlich reduziert. Kreuzterme zwischen Signalanteilen bei unterschiedlichen Zeitpunkten bleiben jedoch unbeeinflusst.

5.5.5 Pseudo-Wigner-Ville-Verteilung

Die Definitionsgleichung (5.50) setzt voraus, dass das zu analysierende Signal unendlich lange beobachtet wird. Auch in diesem Fall wird das Signal jedoch nur in einem beschränkten Intervall gemessen werden. Wie im Zusammenhang mit der DFT wird dieser Tatsache durch eine Fensterung im Zeitbereich mit $h(t)$ Rechnung getragen:

$$W_{xx}^{\text{PW}}(t, f) = \int_{-\infty}^{\infty} h(\tau) x\left(t + \frac{\tau}{2}\right) x^*\left(t - \frac{\tau}{2}\right) e^{-j2\pi f\tau} d\tau. \quad (5.62)$$

Es kann gezeigt werden, dass dies einer Faltung bezüglich der Frequenz mit dem Spektrum der Fensterfunktion entspricht:

$$W_{xx}^{\text{PW}}(t, f) = W_{xx}(t, f) *_f H(f) = \int_{-\infty}^{\infty} W_{xx}(t, f') H(f - f') df'. \quad (5.63)$$

Aufgrund der Faltung mit der Fourier-transformierten Fensterfunktion entspricht die Pseudo-Wigner-Ville-Verteilung nur noch näherungsweise der des ursprünglichen Signals. Es entsteht aufgrund der Fensterung ein Leckeffekt.

5.5.6 Geglättete Pseudo-Wigner-Ville-Verteilung

Die geglättete Pseudo-Wigner-Ville-Verteilung entsteht aus der Pseudo-Wigner-Ville-Verteilung durch Faltung mit einem eindimensionalen Glättungsfilter $g(t)$:

$$W_{xx}^{\text{SPW}}(t, f) = W_{xx}^{\text{PW}}(t, f) *_t g(t) \quad (5.64)$$

$$= \int_{-\infty}^{\infty} g(t - t') W_{xx}^{\text{PW}}(t', f) dt' \quad (5.65)$$

$$= \int_{-\infty}^{\infty} g(t - t') \int_{-\infty}^{\infty} h(\tau) x\left(t' + \frac{\tau}{2}\right) x^*\left(t' - \frac{\tau}{2}\right) e^{-j2\pi f\tau} d\tau dt'. \quad (5.66)$$

Die zusätzliche Glättung bewirkt eine Unterdrückung der Kreuzterme [KSW08].

5.5.7 Zeitdiskrete Wigner-Ville-Verteilung

Die Wigner-Ville-Verteilung in zeitdiskreter Form lässt sich als diskrete Fourier-Transformierte der temporären AKF des Zeitsignals darstellen:

$$W_{xx}[n, k] = 2 \sum_{m=-(N-1)}^{N-1} x_{n+m} x_{n-m}^* e^{-j4\pi km/N} \quad (5.67)$$

$$= 2 \cdot \text{DFT}_{2k} \{x_{n+m} x_{n-m}^*\} . \quad (5.68)$$

Die Implementierung der DFT erfolgt dabei wieder durch die FFT.

Um Aliasing-Effekte zu vermeiden, muss ein weitaus schärferes Abtasttheorem eingehalten werden als sonst üblich. Für die Abtastfrequenz gilt hier:

$$f_A \geq 4f_{\max} . \quad (5.69)$$

Da die Wigner-Ville-Verteilung prinzipiell zu jedem Zeitpunkt eine FFT verlangt, ist der Berechnungsaufwand entsprechend hoch. Nähere Informationen zur zeitdiskreten Wigner-Ville-Verteilung findet man in [CM80, Teil II].

5.5.8 Zusammenhang zwischen Wigner-Ville-Verteilung und STFT

Es besteht ein interessanter Zusammenhang zwischen dem Spektrogramm der STFT und der Wigner-Ville-Verteilung. Beide Transformationen sind verschiebungsinvariant und gehören zur sogenannten **Cohen-Klasse**.

Für das Spektrogramm gilt:

$$S_x^\gamma(\tau, f) = |F_x^\gamma(\tau, f)|^2 = \left| \int_{-\infty}^{\infty} x(t) \gamma^*(t - \tau) e^{-j2\pi ft} dt \right|^2 . \quad (5.70)$$

Mit $x_\tau(t) = x(t) \gamma^*(t - \tau)$ kann man also schreiben:

$$S_x^\gamma(\tau, f) = |X_\tau(f)|^2 . \quad (5.71)$$

Andererseits gilt die Eigenschaft der Wigner-Ville-Verteilung, dass bei Integration über t das Energiedichtespektrum resultiert. Es gilt also:

$$S_x^\gamma(\tau, f) = |X_\tau(f)|^2 = \int_{-\infty}^{\infty} W_{x_\tau x_\tau}(t, f) dt . \quad (5.72)$$

Wendet man die Modulations- und Zeitverschiebungseigenschaft (5.56) in Verbindung mit der Multiplikationseigenschaft (5.58) auf (5.70) an, so erhält man

$$S_x^\gamma(\tau, f) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W_{xx}(t', f') W_{\gamma\gamma}(t' - t, f' - f) dt df \quad (5.73)$$

oder

$$S_x(\tau, f) = W_{xx}(t, f) \underset{t, f}{**} W_{\gamma\gamma}(t, f). \quad (5.74)$$

In ähnlicher Weise kann übrigens auch eine Verbindung zwischen der Wigner-Ville-Verteilung und der Wavelet-Transformation hergestellt werden. Näheres dazu kann in [Mer13] nachgelesen werden.

5.6 Sprachsignale

Innerhalb dieses Abschnittes sollen nur einige Aspekte der Produktion menschlicher Sprache und grundlegende Eigenschaften diskutiert werden. Für weiterführende Informationen wird auf das Buch „Sprachverarbeitung“ [PK08] verwiesen.

5.6.1 Sprachproduktion

Die Produktion der Sprache lässt sich in zwei Bestandteile gliedern: Schallerzeugung und Klangformung. Die Schallerzeugung findet hierbei prinzipiell mit der aus der Lunge strömenden Luft und den Stimmbändern statt. Die aus der Lunge strömende Luft wird durch einen Spalt zwischen den beiden Stimmbändern geführt. Dieser Spalt wird als Stimmritze oder Glottis bezeichnet. Abhängig von dem Grad der Anspannung der Stimmbänder ergeben sich zwei unterschiedliche Arten der Schallanregung: Die stimmlose und die stimmhafte Schallanregung.

Bei der stimmlosen Anregung sind die Stimmbänder entspannt. Die Luft gelangt somit ohne Hindernis durch die Glottis. Das dabei mit der Luftströmung verbundene Geräusch wird als Rauschen wahrgenommen. Die stimmlose Anregung erfolgt für Konsonanten. Im Gegensatz dazu sind bei der stimmhaften Anregung die Stimmbänder angespannt. Die aus der Lunge ausgestoßene Luft regt nun die Stimmbänder zum Schwingen an. Dadurch öffnet und schließt sich die Glottis. Der Luftstrom wird somit periodisch unterbrochen. Die Schallanregung besitzt nun einen impulsartigen Charakter. Die Schwingfrequenz dieses Schallsignals variiert während des Sprechens in einem Umfang von ca. einer Oktave. Möglich ist dies durch das unterschiedliche Maß an Anspannung der Stimmbänder. Für eine Frau beträgt die mittlere Anregungsfrequenz etwa $f = 220$ Hz und bei einem Mann ungefähr $f = 120$ Hz. Insbesondere Vokale resultieren aus der stimmhaften Anregung [PK08].

Die Klangformung findet im Rachen-, Mund- und Nasenraum statt. Dorthin gelangt der Luftstrom, nachdem er die Stimmbänder passiert hat. Dieser Raum wird auch als Vokaltrakt bezeichnet. Die Klangformung erfolgt nun durch das Verändern der Stellung von Gaumensegel, Zunge, Zähne und Lippen. Somit ergeben sich unterschiedliche Hohlräume mit variierenden Resonanzfrequenzen, die das Anregungssignal in seiner Form verändern. Der Vokaltrakt stellt somit ein zeitlich variables Filter dar. Über die Öffnungen von Mund und Nase gelangt das modulierte Anregungs-

signal als akustisches Sprachsignal in die Umwelt. In Abb. 5.14 ist ein vereinfachtes technisches Modell der Sprachproduktion dargestellt [Wen04].

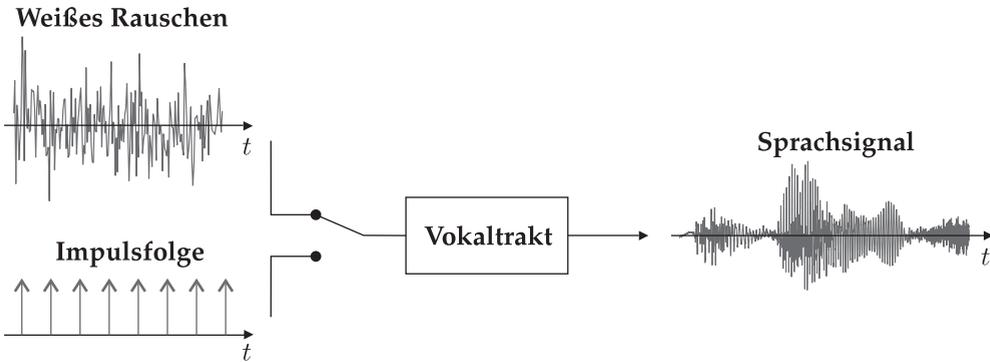


Abbildung 5.14. Technisches Modell der menschlichen Sprachproduktion.

5.6.2 Eigenschaften

Im vorherigen Abschnitt wurde die Sprachproduktion erläutert und dabei festgestellt, dass die Art der Schallanregung und die Klangformung im Laufe der Zeit variieren. Das resultierende Sprachsignal verändert daher über die Zeit seine Eigenschaften. Des Weiteren ist das Sprachsignal stark von den anatomischen Gegebenheiten des jeweiligen Sprechers abhängig und somit von Sprecher zu Sprecher unterschiedlich. Folglich ergeben sich auch für den Frequenzbereich eines Sprachsignals Werte, die vom Sprecher abhängig sind. Es kann jedoch eine maximale Bandbreite für Sprachsignale angegeben werden. Diese liegt für weibliche Stimmen zwischen $170 \text{ Hz} \leq f \leq 10 \text{ kHz}$ und für männliche Stimmen zwischen $100 \text{ Hz} \leq f \leq 9 \text{ kHz}$ [PK08].

Für die Analyse von Sprachsignalen ist es von Bedeutung, die Zeitdauer zu kennen, in welcher ein Sprachsignal als quasi-stationär betrachtet werden kann. Dieser Zeitraum liegt im Allgemeinen zwischen einigen Millisekunden und kann sich bis auf mehrere hundert Millisekunden ausdehnen. Die Mindestgröße eines solchen Analyseintervalls wird im Wesentlichen von der unteren Grenzfrequenz bestimmt, da, um diese auszuwerten, mindestens eine Schwingungsperiode betrachtet werden muss. Für Sprachsignale liegt daher das minimale Analyseintervall bei $T = \frac{1}{f_{\text{Grenz}}} = 10 \text{ ms}$. Dabei wird in der Regel eine Betrachtungsdauer von ca. 25 ms verwendet, damit auch für die tiefste Frequenz eine Mittelung über wenige Perioden möglich ist und somit die Auswertung robuster wird [PK08].

5.6.3 Formanten

Die Resonanzen des Vokaltrakts nennt man in der Akustik oder Phonetik **Formanten**. In der Zeit-Frequenz-Ebene erscheinen sie oftmals als Regionen mit hoher Signalenergie. Betrachtet man die bei Sprachsignalen auftretenden Zeitkonstanten, verändern sich Formanten nur relativ langsam über der Zeit. Formanten stellen ein hilfreiches Werkzeug zur Codierung, Erkennung und Qualitätsverbesserung von Sprachsignalen dar, da sie sich durch wenige Parameter einfach beschreiben lassen [BH08].

5.7 Schallausbreitung in begrenzten Räumen

Ausgehend vom Erzeuger breitet sich Schall normalerweise radial aus. Besteht zwischen der Quelle $s_i(t)$ und dem Sensor $x_j(t)$ eine direkte Sichtverbindung, wird der sogenannte Direktschall als Erstes am Mikrofon detektiert. In einer begrenzten Umgebung treten neben dem Direktschall auch Reflexionen auf. Die Anzahl der reflektierenden Flächen gibt die Ordnung an. In Abb. 5.15 sind neben dem direkten Ausbreitungsweg auch jeweils eine Reflexion erster und zweiter Ordnung eingezeichnet.

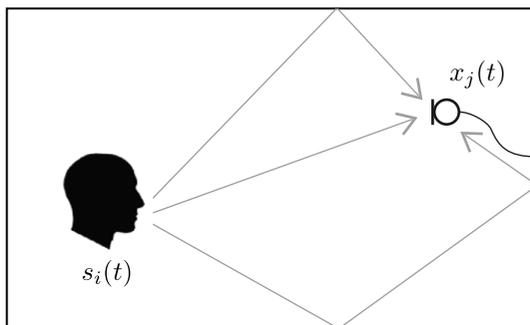


Abbildung 5.15. Schallausbreitung von Quelle $s_i(t)$ zu Sensor $x_j(t)$.

5.7.1 Signalmodelle

Für die Analyse der Signale ist es sinnvoll, Modelle für die Schallausbreitung zu definieren. Im Folgenden werden die beiden Modelle zur Schallausbreitung im Freifeld (ohne Reflexionen) und in begrenzten Räumen vorgestellt.

5.7.1.1 Freifeld

Bei der Ausbreitung im freien Raum kommt an den Sensoren nur der Direktschall an. Das Signal im j -ten Sensor

$$x_j(t) = a_{ji} s_i(t - t_{ji}) \quad (5.75)$$

kann dementsprechend als gedämpfte und zeitverzögerte Version eines Quellsignals $s_i(t)$ dargestellt werden. Die Dämpfung a_{ji} und die Laufzeit t_{ji} sind vom Abstand d_{ji} zwischen Quelle und Sensor abhängig. Die Laufzeit

$$t_{ji} = \frac{d_{ji}}{c_0} \quad (5.76)$$

ist über die Schallgeschwindigkeit c_0 mit dem Abstand verknüpft.

5.7.1.2 Reflexionsbehaftete Umgebung

Befinden sich Schallquellen und Sensoren in einem begrenzten Raum, treten, wie in Abb. 5.15 dargestellt, Reflexionen an Wänden und Objekten auf. Für eine exakte Berechnung der Schallausbreitung müssen theoretisch die Wellengleichungen unter Berücksichtigung von Nebenbedingungen (Begrenzungen durch Wände etc.) gelöst werden. Im Folgenden kann vereinfacht angenommen werden, dass die Ausbreitung des Schalls mit der geometrischen Akustik beschrieben werden kann, d. h. der Schall bewegt sich entlang einer Geraden und es gilt das Reflexionsgesetz (Einfallswinkel = Austrittswinkel).

Zum Sensorsignal liefern alle Ausbreitungspfade einen Beitrag, die nach einer beliebigen Anzahl von Reflexionen am Sensor ankommen. Das Sensorsignal kann bei Mehrwegeausbreitung (L Ausbreitungswege) durch

$$x_j(t) = \sum_{k=1}^L a_{ji}^k s_i(t - t_{ji}^k) = a_{ji}(t) * s_i(t) \quad (5.77)$$

dargestellt werden. Jeder Pfad hat eine spezifische Dämpfung und Zeitverzögerung. Alternativ kann der Einfluss der Umgebungsbedingungen in der Raumimpulsantwort $a_{ji}(t)$ zusammengefasst werden.

5.7.2 Raumimpulsantworten

Die Raumimpulsantworten beschreiben die Änderung eines Signals, das sich in einem geschlossenen Raum ausbreitet. Die Impulsantworten sind nicht nur von der Struktur des Raumes abhängig, sondern auch von der Lage der Quellen und Sensoren. Ein Beispiel für eine Raumimpulsantwort ist in Abb. 5.16 skizziert. Der erste Peak beschreibt den direkten Ausbreitungspfad. Die Amplitude ist tendenziell deutlich größer als die Amplituden der Reflexionen. Die folgenden Werte beschreiben die Reflexionen. Vereinzelt sind höhere Werte im Signal erkennbar.

5.8 Laufzeit- und Richtungsschätzung

Durch die Verwendung mehrerer Sensoren kann die Position akustischer Quellen ermittelt werden. Die Schätzung besteht aus zwei Teilen: Der Berechnung der Laufzeit-

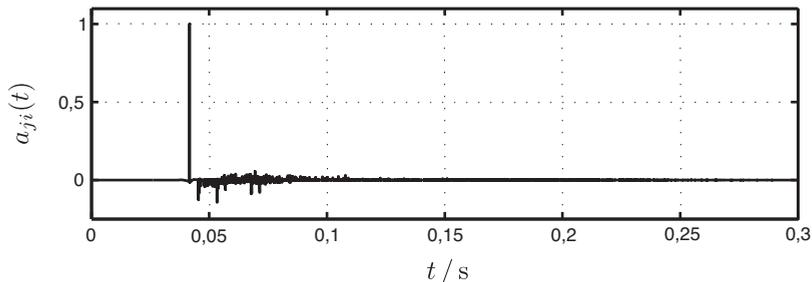


Abbildung 5.16. Beispiel für eine Raumimpulsantwort.

differenz zwischen beiden Sensoren und der Bestimmung der Einfallsrichtung aus der Laufzeit und der Sensorkonfiguration.

5.8.1 Laufzeitschätzung

Die Korrelation ist ein Maß für die Ähnlichkeit zweier Signale. Diese Kenngröße bietet sich für die Bestimmung der Laufzeit an, da Schallsignale, die von einer Quelle emittiert werden, auch nach der Ankunft an verschiedenen Sensoren eine gewisse Ähnlichkeit aufweisen. Für grundlegende Informationen zur Korrelationsmesstechnik sei z. B. auf [PL15] verwiesen.

5.8.1.1 Kreuzkorrelation

Die Kreuzkorrelation berechnet das Ähnlichkeitsmaß $r_{x_1x_2}(\tau)$ zweier reeller Signale $x_1(t)$ und $x_2(t)$ für beliebige Zeitverschiebungen τ durch die Integration über das Produkt der beiden Signale, wobei eines der Signale zeitverschoben eingeht:

$$r_{x_1x_2}(\tau) = \int_{-\infty}^{\infty} x_1(t) x_2(t + \tau) dt. \quad (5.78)$$

Eine effiziente Bestimmung im Frequenzbereich ist mit Hilfe der spektralen Kreuzleistungsdichte $S_{x_1x_2}(f) = X_1(f) \cdot X_2^*(f)$ möglich. Die Werte

$$r_{x_1x_2}(\tau) = \int_{-\infty}^{\infty} S_{x_1x_2}(f) e^{j2\pi f\tau} df \quad (5.79)$$

erhält man durch Rücktransformation in den Zeitbereich. Bei der diskreten Realisierung ist die Beobachtungszeit beschränkt (N Abtastwerte) und das Integral geht in eine Summation über.

5.8.1.2 Verallgemeinerte Kreuzkorrelation

Die **verallgemeinerte Kreuzkorrelation** (engl. *generalized cross correlation*, GCC) stellt eine verbesserte Implementierung der Kreuzkorrelation dar. Die Methode wurde 1976 von Knapp und Carter [KC76] veröffentlicht und ist eine der bekanntesten Methoden zur Laufzeitschätzung. Die Berechnung erfolgt im Frequenzbereich:

$$r_{x_1x_2}^{\text{GCC}}(\tau) = \int_{-\infty}^{\infty} W(f) S_{x_1x_2}(f) e^{j2\pi f\tau} df. \quad (5.80)$$

Im Vergleich zur Kreuzkorrelation unterscheidet sich die GCC nur um den Vorfaktor $W(f)$. Für die Wahl des Faktors wurden bereits in [KC76] unterschiedliche, anwendungsspezifische Funktionen vorgestellt. Für die Laufzeitschätzung in reflexionsbehafteter Umgebung eignet sich insbesondere der *Phase-Transform*-Vorfaktor

$$W_{\text{Phat}}(f) = \frac{1}{|S_{x_1x_2}(f)|}, \quad (5.81)$$

durch die der Einfluss der Sprechereigenschaften unterdrückt wird. Sie wird im weiteren Verlauf als Vorfaktor verwendet. Die allgemeine Gleichung kann man dementsprechend zu

$$r_{x_1x_2}^{\text{GCC}}(\tau) = \int_{-\infty}^{\infty} \frac{1}{|S_{x_1x_2}(f)|} S_{x_1x_2}(f) e^{j2\pi f\tau} df \quad (5.82)$$

$$= \int_{-\infty}^{\infty} e^{j(\arg[S_{x_1x_2}(f)])} e^{j2\pi f\tau} df = \int_{-\infty}^{\infty} e^{j\Delta\varphi(f)} e^{j2\pi f\tau} df \quad (5.83)$$

vereinfachen. Es wird nur noch die Phasendifferenz zwischen den Sensoren transformiert. Für die diskrete Realisierung gilt:

$$r_{x_1x_2}^{\text{GCC}}(n) = \sum_{k=0}^N e^{j\Delta\varphi(f_k)} e^{j2\pi n \frac{k}{N}}. \quad (5.84)$$

Die Transformation in den Frequenzbereich erfolgt mit einer N -Punkte-DFT.

5.8.2 Ermittlung der Schalleinfallrichtung

Die Bestimmung der Position der Schallquelle kann nur relativ zum Sensoraufbau erfolgen. In Abb. 5.17 (links) sind Quelle und Sensoren eingezeichnet. Aus den unterschiedlichen Abständen des Sprechers zu den beiden Sensoren resultiert ein spezifischer Weg- bzw. Laufzeitunterschied. Ist der Abstand der Quelle groß gegenüber dem Sensorabstand, gilt die Fernfeldnäherung. Die einfallende Schallwelle kann als ebene Welle betrachtet werden, wie im rechten Bild eingezeichnet. Zwischen der An-

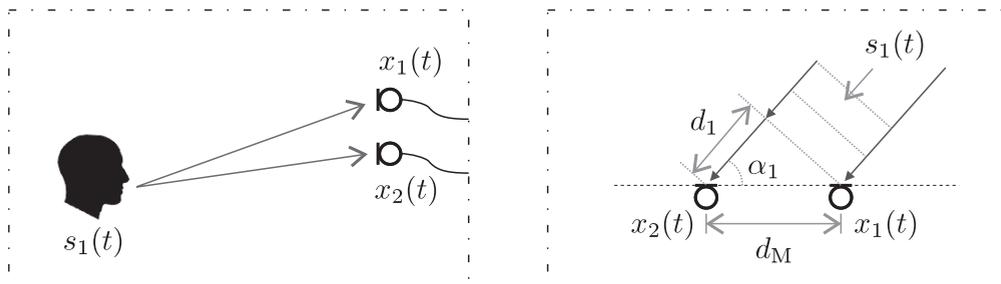


Abbildung 5.17. Bestimmung der Schalleinfallrichtung aus der Laufzeit.

kunft der Wellenfront am ersten und zweiten Sensor muss die Schallwelle den Weg d_1 zurücklegen.

Nach der Bestimmung der Laufzeitdifferenz Δt_1 zwischen den beiden Sensoren kann der Einfallswinkel α_1 berechnet werden:

$$\cos \alpha_1 = \frac{d_1}{d_M} = \frac{c_0 \cdot \Delta t_1}{d_M} . \quad (5.85)$$

Der Sensorabstand d_M muss bekannt sein.

5.9 Versuchsaufbau und Matlab-Umgebung

5.9.1 Versuchsaufbau

5.9.1.1 Mikrofone

Als Mikrofone werden zwei Beyerdynamik MCE 60 verwendet.

5.9.1.2 Analogelektronik

Bei der Aufnahme von Stereosignalen mit einem herkömmlichen Rechner muss normalerweise der *LINE IN*-Eingang genutzt werden. Das Mikrofonsignal muss aus diesem Grund verstärkt werden. Als Verstärkerbaustein werden zwei *LM 386 Low Voltage Audio Power Amplifier* von *National Semiconductor* verwendet.

5.9.2 Matlab-Umgebung

Alle Versuche werden in *Matlab* oder *Simulink* durchgeführt. Eine Einführung erfolgte bereits im ersten Versuch. Der folgende Kommentar soll die Handhabung des Blockes erleichtern.

Buffer- bzw. Unbuffer-Block:

Mit dem Buffer-Block ist es durch den Parameter *overlap* möglich, eine Überlappung der Daten herzustellen. Es wird also eine bestimmte Folge an Daten ausgegeben

und anschließend der nächste Datensatz mit einem gewünschten Overlap versehen. In diesem Versuch ist es sinnvoll, eine Überlappung von ungefähr 60–70 % der Vektorlänge zu verwenden (also ca. 150–180 Werte), d. h., es werden nur 30–40 % der Daten eines Ausgangsvektors verändert.

Der Unbuffer-Block ermöglicht es, ein gepuffertes Signal wieder zu entpuffern. Das Ausgangssignal dieses Blocks ist stets eindimensional.

5.10 Versuchsvorbereitende Aufgaben

Aufgabe 5.1: Kurzzeit-Fourier-Transformation

- a) Beschreiben Sie zwei verschiedene Interpretationen der Kurzzeit-Fourier-Transformation.
 - b) Welchen Einfluss hat die Länge des Analysefensters?
 - c) Welches Fenster halten Sie für die Zeit-Frequenz-Analyse für besonders geeignet? Begründen Sie.
-

Aufgabe 5.2: Wavelet-Transformation

Wo liegen die Unterschiede zwischen der Kurzzeit-Fourier-Transformation und der Wavelet-Transformation. . .

- a) bezüglich der Berechnung?
 - b) bezüglich der Zeit- und Frequenzauflösung des Spektrums?
-

Aufgabe 5.3: Wigner-Ville-Verteilung

- a) Nennen Sie den größten Vorteil und den größten Nachteil der Wigner-Ville-Verteilung gegenüber der STFT und der Wavelet-Transformation.
 - b) Warum sollte das analytische Signal anstelle des reellen Signals transformiert werden?
-

Aufgabe 5.4: Sprachsignalverarbeitung

- a) Was sind Formanten?
- b) Sie sollen die Kurzzeit-Fourier-Transformation eigenständig programmieren (nur die Transformation in den Zeit-Frequenz-Bereich). Auf der Internetseite zum Praktikum können Sie eine gezippte Datei herunterladen, die bereits eine Vorlage für das Programm liefert. Setzen Sie die einzelnen Schritte um, wobei die Fensterlänge und die Überlappung der einzelnen Fenster frei wählbar sein sollen.

- c) Was fällt Ihnen auf, wenn Sie das Sprachsignal mit der STFT analysieren (Ausführen von `test.m`)? Beschreiben Sie Ihre Beobachtungen.
- d) Bestimmen Sie die Grundfrequenz des Vokals 'u' (Verwenden Sie hierzu `fft()` und gegebenenfalls die Dokumentation dieses Befehls).

Aufgabe 5.5: Raumimpulsantwort

Um die Auswirkungen der Impulsantworten auf ein Sprachsignal zu testen, steht auch für die fünfte Aufgabe eine gezippte Datei zur Verfügung. Laden Sie sich diese Datei herunter.

- a) Öffnen Sie das Programm `convRIR.m` und gehen Sie das Programm Schritt für Schritt durch.
- b) Führen Sie das Skript aus und hören Sie sich die unterschiedlichen Signale an. Was fällt Ihnen auf?

5.11 Versuche am Versuchstag

Im praktischen Teil dieses Versuchs werden die Charakteristika von Sprachsignalen mittels der STFT festgestellt. Außerdem wird der Einfluss der Umgebungsbedingungen auf die Aufzeichnung von Sprachsignalen untersucht sowie eine Richtungs-schätzung der aufgenommenen Signale vorgenommen. Der Versuch schließt mit dem Vergleich der Zeit-Frequenz-Analyse, die mittels STFT, Wavelet-Transformation und Wigner-Ville-Verteilung ermittelt wurde. Zunächst soll jedoch die Aufzeichnung von Sprachsignalen mit *Simulink* implementiert werden.

Achten Sie darauf, den Mikrofon-Vorverstärker bei der Datenaufnahme eingeschaltet zu haben. Erstellen Sie unter `D:/PDSV/Gruppen` einen Arbeitsordner für Ihre Gruppen. Legen Sie alle Dateien, die Sie im Laufe des Versuchs erstellen, nur in diesem Ordner ab! Im Ordner `D:/PDSV/Versuch_5` finden Sie alle benötigten Dateien. Wenn Sie an diesen Dateien Veränderungen vornehmen, dann überschreiben Sie diese bitte nicht.

Aufgabe 5.6: Datenaufnahme

Simulink kann zur Live-Analyse und Datenaufnahme von Signalen verwendet werden. Hierzu enthält das *Signal Processing Blockset* die Blöcke *From Audio Device* und *Signal From Workspace* zum Datenimport sowie die Blöcke *To Audio Device* und *Signal To Workspace* zum Datenexport. Sind die Daten erst einmal in den *Workspace* exportiert, können sie dort mit den üblichen *Matlab*-Funktionen verarbeitet werden. Zunächst sollen diese Grundfunktionen erprobt werden.

- a) Generieren Sie hierzu drei neue Modelle, die die oben genannten Blöcke enthalten und Folgendes ermöglichen:
- Modell 1: Signalaufnahme in den Workspace mit einer Dauer von $t = 10$ s und einer Abtastrate von $f = 16$ kHz – stereo.
 - Modell 2: Wiedergabemöglichkeit der zuvor aufgezeichneten Variable.
 - Modell 3: Parallele Ausgabe und Aufnahme der Daten (beliebige Variable ausgeben und das Signal der Mikrofone gleichzeitig in einer anderen Variable abspeichern).
- Speichern Sie die Werte als `double` ab. Den Datentyp können Sie in der Konfiguration des Aufnahmeblocks einstellen.
- b) Erproben Sie Ihre Modelle, indem Sie eine Sprachaufzeichnung durchführen und diese anschließend wiedergeben. Verwenden Sie zum Testen des dritten Modells ein beliebiges Sprachsignal aus dem Workspace. Speichern Sie die Modelle in ihrem Projektverzeichnis ab. Sie werden für spätere Aufgaben benötigt.

Hinweis: Achten Sie auf die Aufnahmeeinstellungen der Soundkarte, um zu garantieren, dass die Signale korrekt aufgenommen werden. Probleme ergeben sich insbesondere bei zu hoher Verstärkung, da das Signal in diesem Fall abgeschnitten wird (Clipping!).

Aufgabe 5.7: Sprachcharakteristik

Im Folgenden soll das Frequenzspektrum menschlicher Sprache analysiert werden. Hierzu wird ein Modul zum Berechnen der Kurzzeit-Fourier-Transformation (STFT) benötigt.

- a) Fügen Sie die nötigen Blöcke zur Berechnung der STFT des Signals der Soundkarte in die Vorlage `spektrum_anzeige.mdl` ein. Die Darstellung der Ergebnisse übernimmt der bereits vorhandene Teil. Verwenden Sie hierzu die Blöcke `Magnitude FFT`, `Buffer`, `From Audio Device` und `Window Function` aus dem `Signal Processing Blockset`. Verwenden Sie jeweils 256 Werte (mono) bei einer Abtastrate von 16 kHz zur FFT-Berechnung mit einer Überlappung von 128 Werten.
- b) Was bewirkt der U - Y - Block (im Subsystem `Spektrum Anzeige`) in diesem Fall?
- c) Testen Sie ihr Modell, indem Sie über den Frequenzgenerator Signale ausgeben und die Ergebnisse der Transformation bewerten. Verwenden Sie hierbei die Signalformen `Sinus`, `Dreieck` und `Rechteck`.
- d) Wie kann hier Aliasing sichtbar gemacht werden? Machen Sie einen Screenshot davon.
- e) Analysieren Sie online ein Sprachsignal. Was fällt Ihnen im Signal auf? Machen Sie einen Screenshot, anhand dessen Sie die Sprachcharakteristik zeigen können.

Abschließend sollen noch charakteristische Eigenschaften der Sprache betrachtet werden. Zu diesem Zweck sollen die Vokale in dem Ordner `Vokale` bei 11 kHz analysiert werden.

- f) Berechnen Sie für jeden Vokal die Fourier-Transformierte (in *Matlab*, nicht in *Simulink*) und ermitteln Sie die ersten beiden Formanten. Zum Einlesen bietet sich im Command Window der Befehl `audioread()` an. Gleichen Sie die Resultate mit der Formantenkarte ab. Orientieren Sie sich bei der Umsetzung gegebenenfalls an der Dokumentation der Fourier-Transformation (`fft`) in der *Matlab*-Hilfe.

Aufgabe 5.8: Einfluss der Umgebungsbedingungen

Die **Raumimpulsantwort** (engl. *room impulse response*, RIR) stellt die Klangcharakteristik eines Raumes dar. Im Folgenden soll die RIR des Praktikumsraumes vermessen werden.

- a) Erweitern Sie `trigger.mdl` so, dass die Bedingung des IF-Blocks bei Aufzeichnung eines lauten Geräusches 'TRUE' wird und die Aufnahme dementsprechend gestoppt wird. Als Merkmal kann die Signalenergie verwendet werden. Ergänzen Sie das Simulink-Modell und bestimmen Sie experimentell einen geeigneten Schwellwert für die Energie. Zur Ergänzung des Modells sind die Blöcke `Abs`, `Matrix Sum` und `Unbuffer` hilfreich. Überlegen Sie sich, wie Sie diese sinnvoll einsetzen.

Verwenden Sie zur Realisierung den IF-Block, der den Block `Stop after Delay` aus dem Praktikums-Blockset betätigt. Bestimmen Sie experimentell einen Schwellwert.

- b) Zeichnen Sie eine Raumimpulsantwort auf, indem Sie ein möglichst kurzes, lautes Geräusch erzeugen, z. B. durch Klatschen.
- c) Analysieren Sie das gespeicherte Signal in *Matlab* (`plot`, `soundsc`).
- d) Falten Sie das gewonnene Signal mit der beiliegenden Sprachaufzeichnung `sprache_ohne_hall.wav` und geben Sie das Resultat mittels der Funktion `soundsc` aus. Dokumentieren Sie Ihre Ergebnisse:
- Zeichnen Sie beide Signale unter Verwendung von `subplot` in eine gemeinsame Darstellung.
 - Berechnen Sie die STFT der beiden Signale mithilfe der in den Vorbereitungsaufgaben implementierten STFT und visualisieren Sie die Ergebnisse in der gleichen Art wie in der vorhergehenden Aufgabe.
 - Beschreiben Sie Ihre Erkenntnisse.
- e) Als Nächstes soll das Signal mittels einer quasi-analogen Faltung mit dem Raumklang beaufschlagt werden. Geben Sie dazu die Sprachdatei über den Lautsprecher aus und nehmen Sie das Signal simultan auf. Zeichnen Sie das resultierende Signal und hören Sie sich auch das Signal an. Dokumentieren Sie die Resultate analog zu Aufgabe c). Woher kommt der Unterschied?

Aufgabe 5.9: Laufzeitschätzung mit der Kreuzkorrelation und der GCC

Um die Laufzeit von Signalen zu bestimmen, wird im einfachsten Fall eine Kreuzkorrelation durchgeführt. Diese hat ihr Maximum an der Position, die der Zeitverschiebung beider Signale entspricht. Die Laufzeit kann, wie in den Grundlagen beschrieben, in einen Winkel umgerechnet werden, welcher der Richtung der Schallquelle entspricht.

- a) Es liegen diskrete Samples vor. Bestimmen Sie allgemein die maximale Anzahl der Samples, die zwischen den Mikrofonen auftreten können – in Abhängigkeit der Abtastrate sowie des Mikrofonabstands.

Hinweis: Betrachten Sie Abb. 5.17 aus dem Theorieteil.

Diese maximale Anzahl multipliziert mit 2 und addiert 1 ergibt die Anzahl der maximal detektierbaren Winkel. Machen Sie sich diesen Zusammenhang klar.

- b) Der bereits vorbereitete Code in der Datei `cc_own.m` stellt eine Funktion zur Verfügung, die zwei Eingangssignale in mehrere Blöcke zerlegt (ohne Überlappung) und für die korrespondierenden Signale jeweils die Kreuzkorrelation (`xcorr`) berechnet. Testen Sie die Funktion mit der Datei `korrelation_testsignal_1/2.wav` (Sensorabstand 1 m). Hierfür müssen Sie die Funktion lediglich in das Testskript `test.m` im entsprechenden Ordner einfügen.
- c) Neben dieser Implementierung der Kreuzkorrelation steht noch eine Realisierung der GCC (`gcc_complete.m`) zur Verfügung. Erweitern Sie das vorherige Skript um diese Funktion und vergleichen Sie die Ergebnisse. Was fällt auf?
- d) Nehmen Sie mit der Hardware Sprachdaten (laut und deutlich sprechen) auf und notieren Sie sich die Position (also den Winkel) der Quelle relativ zu den Sensoren. Verwenden Sie die Funktion `gcc_complete.m`, um die Position der Quelle zu schätzen und dokumentieren Sie Ihre Ergebnisse.

Aufgabe 5.10: Wavelet-Analyse (Offline)

Die Wavelet-Transformation stellt eine alternative Vorgehensweise zur frequenzselektiven Zerlegung der Signale dar. In dieser Aufgabe soll das Verfahren zur Analyse von gesprochenen Lauten verwendet und die Eignung zur Datenkompression betrachtet werden. Hierzu wird das graphische *Matlab*-Tool `wavemenu` verwendet.

a) **Approximationen und Details**

1. Starten Sie `wavemenu` und wählen Sie in der GUI die eindimensionale Analyse Wavelet 1-D aus.
2. Über File → Load Signal können Sie Daten importieren. Es stehen `.wav`-Files für vier Vokale zur Verfügung (im Ordner `Vokale`). Entscheiden Sie sich für ein Signal mit 11 kHz Abtastrate.
3. Zur Zerlegung des Signals stehen verschiedene Wavelets zur Verfügung. Analysieren Sie das Signal mit den Wavelets

- Haar
- Daubechies 4 (dB)

Das Level (Tiefe der Filterbank) wird zu 5 gewählt.

Hinweis zu Wavelets: Für mehr Informationen zu den Wavelets suchen Sie in der Hilfefunktion von *Matlab* nach „*Wavelet Families: Additional Discussion*“ oder nach „*Introduction to the Wavelet Families*“.

4. Nutzen Sie unter Display Mode den Tree Mode und die Möglichkeiten zur statistischen Analyse. Welche Wavelets eignen sich Ihrer Meinung am besten zur Analyse der Signale? Berücksichtigen Sie dabei vor allem die praktische Umsetzbarkeit der Wavelets.

b) Kompression

An einem einfachen Beispiel soll der Einfluss der Detailkoeffizienten auf das Signal gezeigt werden. Um die Unterschiede aufzeigen zu können, wird ein Testsignal komprimiert und anschließend mit dem Originalsignal verglichen.

1. Laden Sie den Vokal ‘u’ in der GUI zur Wavelet 1-D Analyse.
2. Analysieren Sie das Signal mit dem im vorhergehenden Aufgabenteil gewählten Wavelet.
3. Um den Einfluss der Detailsignale auf das Signal abschätzen zu können, soll das Signal komprimiert werden. Über *Compress* kommen Sie in ein Menü, in dem die Komprimierung des Signals durch die Festlegung eines Schwellwertes für die Detailsignale erfolgen kann. Probieren Sie die Funktionen aus.
4. Rekonstruieren Sie das Signal nach der Definition eines beliebigen Schwellwertes und speichern Sie das Ergebnis.
5. Vergleichen Sie das ursprüngliche und rekonstruierte Signal (Hörprobe) und beschreiben Sie die Unterschiede. Welchen Einfluss haben die Detailkoeffizienten?

Aufgabe 5.11: Wigner-Ville-Analyse (Offline)

In dieser Aufgabe soll das Hauptaugenmerk auf die Wigner-Ville-Verteilung und ihre Varianten (Pseudo-Wigner-Ville-Verteilung, geglättete Pseudo-Wigner-Ville-Verteilung) gerichtet werden. Außerdem soll untersucht werden, welche Auswirkungen die Hilbert-Transformation (analytisches Signal) und Upsampling auf die Wigner-Ville-Verteilung des zu analysierenden Signals haben. Vorher soll jedoch kurz die in dieser Aufgabe verwendete Software vorgestellt werden.

Signalanalyse mit Hilfe des Programms *JTF Demo*

Unter D:\PDSV_Versuch5\Versuch5_Sprachsignalverarbeitung_Vorlage_Backup\Aufgabe_11_Wigner_Ville\ befindet sich ein GUI, welches über den Befehl *JTF_Demo* im Command Window gestartet werden kann.

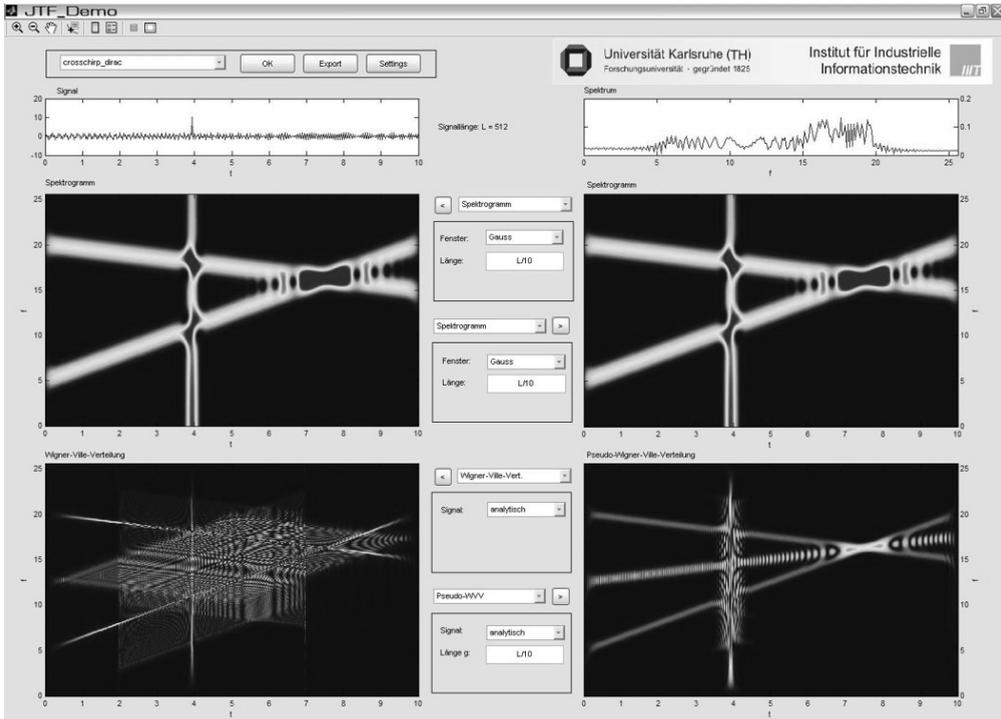


Abbildung 5.18. GUI zur Signalanalyse mittels entsprechender Transformationen.

ACHTUNG! Führen Sie zuvor unbedingt das m-File `files_setPath.m` im darüberliegenden Verzeichnis aus (über den Befehl `files_setPath` im Command Window).

Mit diesem Programm (vgl. Abb. 5.18) wird es dem Benutzer sehr anschaulich ermöglicht, zeitvariante Signale mit Hilfe einer Reihe von Transformationen zu analysieren. Hierzu stellt *JTF_Demo* vier separate Plots zur Verfügung, in denen verschiedene Transformationen des zu analysierenden Signals dargestellt werden können. Das abgebildete Frequenzband umfasst dabei das Intervall $[0, f_A/2]$.

Auch die Möglichkeit einer Parameteränderung ist vorgesehen. So können u. a. die Fensterformen sowie deren Längen variiert werden. *JTF_Demo* gibt dem Benutzer neben der Analyse vorgegebener Signale auch die Möglichkeit, eigene Signale zu analysieren. Für Letzteres muss im entsprechenden Drop-Down-Menü in der linken oberen Ecke der GUI der Punkt DURCHSUCHEN ausgewählt werden.

Zur Verbesserung der Darstellung der Plots der verschiedenen Transformationen kann man über das GUI-Menü *Settings* den minimalen und maximalen Schwellwert, bei dem ein Wert im Plot noch angezeigt wird, verändern.

Hinweis: Skalogramme sind sehr rechenintensiv! Die Berechnung dauert bei 5000 Signalwerten länger als 10 min. Daher sollte auf diese Darstellung verzichtet werden.

Künstliches Signal

Zunächst soll ein Crosschirp-Signal analysiert werden. Dabei sollen das Spektrogramm und verschiedene Varianten der Wigner-Ville-Verteilung verglichen werden.

Nehmen sie dazu folgende Einstellungen in der GUI vor:

Signal:	Crosschirp		
Spektrogramm:	Fenster: Hamming	Länge: $L/10$	
WVV:	Signal: analytisch		
Pseudo-WVV:	Signal: analytisch	Länge g: $L/4$	
Geglättete Pseudo-WVV:	Signal: analytisch	Länge g: $L/4$	Länge h: $L/10$

Hinweis: Der Parameter L bezeichnet die Signallänge.

- Starten sie das m-File. Vergleichen Sie die verschiedenen Transformationen bezüglich ihrer Auflösung und des Auftretens von Kreuztermen.
- Können die Kreuzterme identifiziert werden?

Nehmen Sie in der Oberfläche nun folgende Einstellungen vor:

Signal:	Crosschirp		
Spektrogramm:	Fenster: Hamming	Länge: $L/10$	
WVV:	Signal: analytisch		
WVV:	Signal: upsampled		
WVV:	Signal: reell		

Hinweis: Die Einstellung `upsampled` bewirkt, dass das reelle Signal um den Faktor 2 überabgetastet wird, d. h. es werden Nullen als Zwischenwerte eingefügt, anschließend folgt eine Tiefpassfilterung.

- Vergleichen Sie die WVV des reellen, des analytischen und des überabgetasteten Crosschirp-Signals. Wie erklären Sie sich ihre Beobachtungen?

Gemessenes Signal

Auch diese Aufgabe soll offline bearbeitet werden. In dem Auswahlmeneü für die Signale stehen vier Vokale zur Verfügung. Wählen Sie einen davon aus.

Settings:	min. Threshold: 0	max. Threshold: 10	
Spektrogramm:	Fenster: Hamming	Länge: ?	
WVV:	Signal: analytisch		
Pseudo-WVV:	Signal: analytisch	Länge g: ?	
GeglättetePseudo-WVV:	Signal: analytisch	Länge g: ?	Länge h: ?

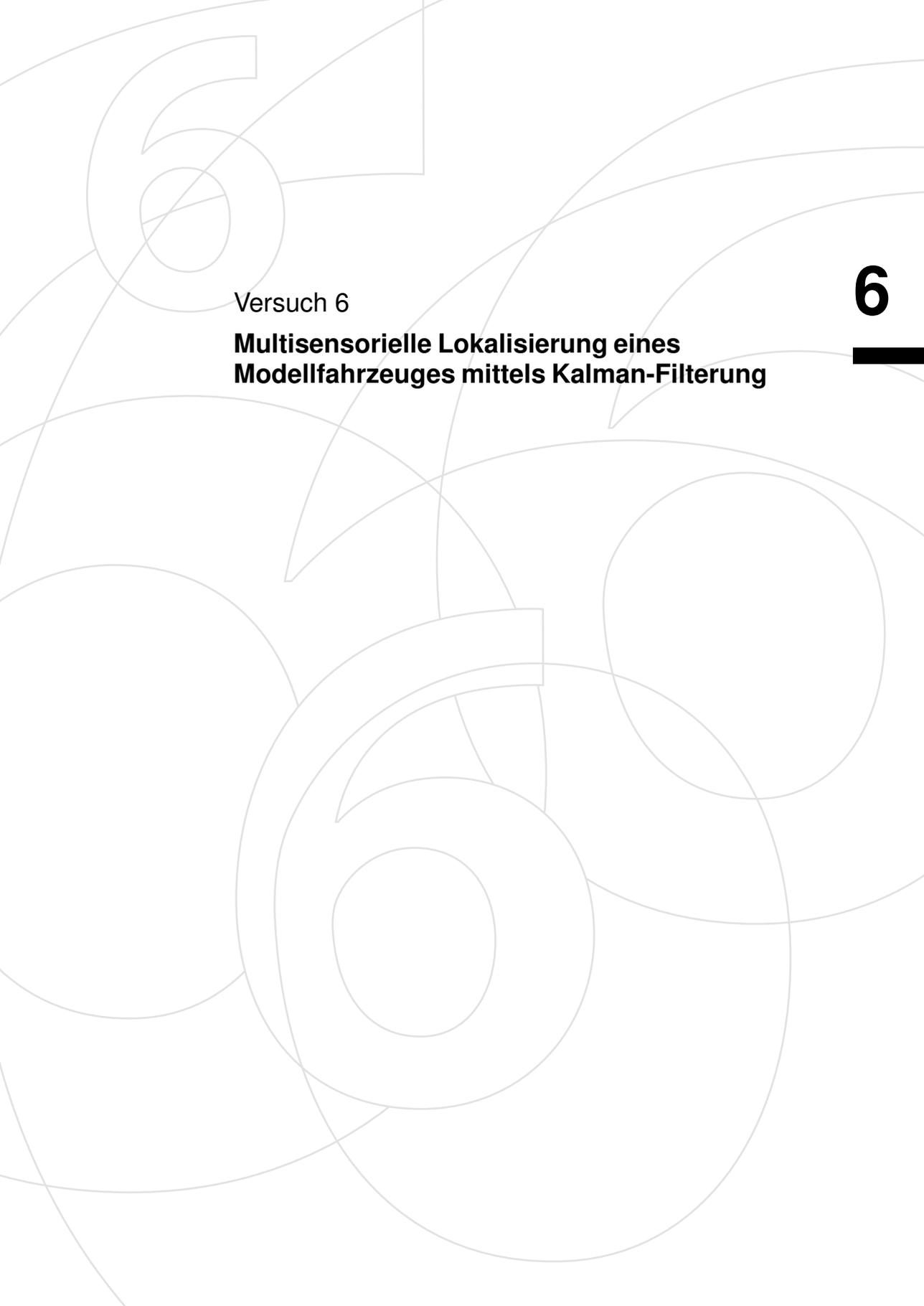
- Wie würden Sie die Fensterlängen wählen? Begründen Sie Ihre Wahl.

e) Welches Verfahren würden Sie zur Repräsentation der Signale verwenden?

5.12

5.12 Literatur

- [BH08] J. BENESTY und Y. HUANG: *Springer Handbook of Speech Processing*. Springer Handbook of Speech Processing. Springer, 2008.
- [Bri95] E. ORAN BRIGHAM: *FFT – Schnelle Fourier-Transformation*. Oldenbourg, München, 6. Auflage, 1995.
- [CM80] T. A. C. M. CLAASEN und W. F. G. MECKLENBRAUKER: *The Wigner distribution: a tool for time frequency signal analysis*. Philips Journal of Research, 35(3):217–250, 1980.
- [KC76] C. KNAPP und G. CARTER: *The generalized correlation method for estimation of time delay*. IEEE Transactions on Acoustics, Speech and Signal Processing, 24(4):320–327, 1976.
- [KK12] KARL-DIRK KAMMEYER und KRISTIAN KROSCHER: *Digitale Signalverarbeitung – Filterung und Spektralanalyse mit MATLAB-Übungen*. Springer Vieweg, Wiesbaden, 8. Auflage, 2012.
- [KSW08] UWE KIENCKE, MICHAEL SCHWARZ und THOMAS WEICKERT: *Signalverarbeitung – Zeit-Frequenz-Analyse und Schätzverfahren*. Oldenbourg, München, 2008.
- [Mer13] ALFRED MERTINS (Herausgeber): *Signaltheorie – Grundlagen der Signalbeschreibung, Filterbänke, Wavelets, Zeit-Frequenz-Analyse, Parameter- und Signalschätzung*. Springer Vieweg, Wiesbaden, 3. Auflage, 2013.
- [PJ15] FERNANDO PUENTE LEÓN und HOLGER JÄKEL: *Signale und Systeme*. De Gruyter Oldenbourg, Berlin, 6. Auflage, 2015.
- [PK08] BEAT PFISTER und TOBIAS KAUFMANN: *Sprachverarbeitung – Grundlagen und Methoden der Sprachsynthese und Spracherkennung*. Springer, Berlin Heidelberg, 2008.
- [PL15] FERNANDO PUENTE LEÓN: *Messtechnik – Systemtheorie für Ingenieure und Informatiker*. Springer, Berlin Heidelberg, 10. Auflage, 2015.
- [Wen04] ANDREAS WENDEMUTH (Herausgeber): *Grundlagen der stochastischen Sprachverarbeitung*. Oldenbourg, München, 2004.



Versuch 6

**Multisensorielle Lokalisierung eines
Modellfahrzeuges mittels Kalman-Filterung**

6

6 Multisensorielle Lokalisierung eines Modellfahrzeuges mittels Kalman-Filterung

6.1	Einleitung	187
6.2	Lokalisierung des Modellfahrzeuges	189
6.2.1	Die Sensoren des Fahrzeugs	189
6.2.2	Die zeitdiskreten Positionsgleichungen	190
6.2.3	Auswertung der Inkrementalgeberdaten	193
6.2.3.1	Berechnung der Wegänderung	193
6.2.3.2	Berechnung der Winkeländerung	194
6.2.4	Auswertung der Beschleunigungssensordaten	195
6.2.5	Auswertung der Gierratensensordaten	197
6.2.6	Fehlerursachen bei der Lokalisierung	197
6.2.6.1	Systematische Fehler	198
6.2.6.2	Zufällige Fehler	199
6.3	Datenfusion durch Kalman-Filterung	200
6.3.1	Zeitdiskrete Beschreibung linearer Systeme	200
6.3.1.1	Das zeitdiskrete System im Zustandsraum	200
6.3.1.2	Zustandsschätzverfahren	201
6.3.2	Das Kalman-Filter	203
6.3.2.1	Theorie des Kalman-Filters	203
6.3.2.2	Das realisierte Kalman-Filter	207
6.3.3	Multisensor-Datenfusion mit dem Kalman-Filter	208
6.4	Die Versuchssoftware	212
6.4.1	Das Programm auf dem Fahrzeugrechner	213
6.4.2	Das Programm auf dem Praktikumsrechner	213
6.4.3	Vorgehensweise bei der Datenaufnahme	216
6.5	Versuchsvorbereitende Aufgaben	216
6.6	Versuche mit dem Fahrzeug	217
6.6.1	Eingangsbemerkungen	217
6.6.2	Messungen nur mit den Sensoren	218
6.6.2.1	Manuelle Varianzenvorgabe	220
6.6.2.2	Abschließender Vergleich	222
6.7	Literatur	223

6 Multisensorielle Lokalisierung eines Modellfahrzeuges mittels Kalman-Filterung

6.1 Einleitung

In diesem Versuch geht es um die Lokalisierung – also die Positionsbestimmung – eines Modellfahrzeuges mit Hilfe des Kalman-Filters. Das Kalman-Filter wird hierbei zur sogenannten Datenfusion, also der Verschmelzung von Daten unterschiedlicher Sensoren, benutzt. Allgemein ist das Kalman-Filter ein Instrument zur Signal- bzw. Zustandsschätzung zeitabhängiger Signale bei instationären Zufallsprozessen, bei welchen die Momente – also auch Mittelwert und Varianz – zeitabhängig sind [PL15]. Für das Kalman-Filter wird ein Prozessmodell im Zustandsraum verwendet [KRS11]. Es stellt eine Verallgemeinerung des Luenberger-Beobachters dar, indem es die stochastischen Störgrößen und stochastischen Messfehler explizit berücksichtigt. Mit Hilfe dieser Filtertechnik kann der Zustand eines dynamischen Systems aus veräuschten Messungen optimal bezüglich der Varianz des Schätzfehlers bestimmt werden. In diesem Zusammenhang sind dynamische Systeme technische, wirtschaftliche oder biologische Prozesse, deren zeitliches Verhalten bei zeitkontinuierlichen Systemen durch gewöhnliche Differentialgleichungen oder bei zeitdiskreten Systemen durch Differenzgleichungen hinreichend genau beschrieben werden kann [Sch77]. Die optimale Zustandsschätzung erfolgt bei gewöhnlichen Differentialgleichungen, also im zeitkontinuierlichen Fall, mittels Kalman-Bucy-Filterung und bei Differenzgleichungen, also im zeitdiskreten Fall, mittels Kalman-Filterung. Die optimale Zustandsschätzung nutzt alle verfügbaren Informationen, um die bestmögliche Aussage über den Zustand des Systems zu einem beliebigen Zeitpunkt zu machen. Die nötigen Informationen stammen aus der Differentialgleichung des Systems (diese wird auch als A-priori-Information oder Strukturwissen bezeichnet) sowie aus den verfügbaren Messwerten und den statistischen Eigenschaften von Messunsicherheiten und Systemstörungen [Sch77].

Entwicklung der Bedeutung des Kalman-Filters

Zu Beginn der 60er-Jahre stieg das Interesse in der Wissenschaft und Technik an der Beobachtung von Systemgrößen mit Hilfe von Beobachtern – unter anderem auch dem Kalman-Filter – an. Einerseits ist dies darauf zurückzuführen, dass der rekursive Algorithmus des Kalman-Filters auf Digitalrechnern einfach in Echtzeit ausführbar ist, andererseits konnte vor allem in der aufkommenden Raumfahrt die Aufgabe der Positionsbestimmung und Navigation von Raumfahrzeugen besser bewältigt wer-

den. Gerade bei Positionsmessungen in der Raumfahrt ist das Verhältnis von Nutz- und Störsignal meist nicht konstant, sondern starken Schwankungen unterworfen. Das Kalman-Filter gewichtet weniger verrauschte Messwerte stärker als solche, die stark verrauscht sind. Stehen in einem Zeitabschnitt nur ungenaue Messungen eines Sensors zur Verfügung, so greift das Kalman-Filter auf vergangene, bessere Messwerte zurück oder berücksichtigt Messwerte anderer Sensoren stärker. Das Kalman-Filter versucht, durch Schätzung möglichst nahe an die wahren Werte zu kommen. Je stärker die Messungen verrauscht sind, desto schlechter ist selbstverständlich die Schätzung der interessierenden Größen. Der Filteralgorithmus ist optimal bezüglich der als bekannt vorausgesetzten Fehlervarianz und liefert deshalb die bestmögliche Schätzung. Es gibt zahlreiche Anwendungsbeispiele des Kalman-Filters. Ein breites Anwendungsfeld der Kalman-Filtertheorie findet sich, wie bereits erwähnt, in der Verfolgung von (Flug-)Zielen. Die Aufgabe eines solchen Filters besteht darin, aus verrauschten Messdaten den momentanen und künftigen Bewegungszustand (Position, Geschwindigkeit, Beschleunigung) des Ziels zu schätzen. Ein anderes typisches Beispiel für derartige Zustandsbestimmungen ist etwa die Temperaturmessung im Inneren von Reaktoren oder Verbrennungsmotoren, wobei Temperaturlaufnehmer nur außen an der Gehäusewand befestigt werden können und die Innentemperatur dann geschätzt werden muss. Kalman-Filter werden auch zur Echokompensation bei der Sprachübertragung [KRS11] und zur Fehlerüberwachung in Systemen und Netzen eingesetzt.

Aufgrund seiner Bedeutung hat das Kalman-Filter auch Einzug in die Lehre an den Hochschulen gefunden. Die Vorteile des Kalman-Filters lassen sich allerdings anhand der relativ komplizierten Herleitungen des Algorithmus schwer verdeutlichen. Nützlich ist eine praktische, anwendungsbezogene Vorführung, die das Ergebnis einer Kalman-Filterung anschaulich darstellt. Eine solche Vorführung ist das Ziel dieses Laborversuchs. Das verwendete Modellauto ist mit verschiedenen Sensoren zur Messung seiner Bewegung bestückt. Während der Versuchsfahrt werden die Sensoren zeitäquidistant abgetastet und die gemessenen Werte per WLAN an den Praktikumsrechner übertragen. Nach der Versuchsfahrt werden die gesammelten Messwerte ausgewertet: Der Fahrtverlauf wird rekonstruiert und graphisch dargestellt. Mit Hilfe der Kalman-Filterung der Sensorwerte wird das Ergebnis der Positionsbestimmung optimiert.

Im Folgenden werden zunächst das Modellfahrzeug und seine Sensoren erläutert. Danach werden das Kalman-Filter und seine Funktionsweise allgemein sowie die im Modellfahrzeug verwendeten konkreten Umsetzungen vorgestellt. Im praktischen Teil werden verschiedene Messfahrten durchgeführt, um die Messergebnisse der Einzelsensoren sowie die anschließende Sensorfusion zu bewerten.

6.2 Lokalisierung des Modellfahrzeugs

Ziel der Lokalisierung ist die Bestimmung der Position eines Objektes. Zu einer direkten Positionsbestimmung können bekannte feste Punkte, z. B. hohe Berge oder Sterne, dienen, indem mehrere solcher Punkte kreuzweise angepeilt werden. Man benutzt zur Lokalisierung auch Satelliten, beispielsweise das vom amerikanischen *Department of Defense* entwickelte **Global Positioning System** (GPS). Moderne Navigationssysteme für das Auto zeigen dem Fahrer seine aktuelle Position und nach dessen Eingabe auch das Fahrtziel sowie den Weg dorthin an. Daraus entstand auch die Idee dieses Versuchs: Ein Auto bekommt seine absolute Position nur an bestimmten Punkten, z. B. Kreuzungen, mitgeteilt und muss seine aktuelle Position dazwischen selbst bestimmen können. Wir wollen die Position eines Modellfahrzeugs durch laufende Beobachtung der Bewegung bestimmen, indem wir mehrere durch verschiedene Sensoren gemessene Größen heranziehen. Wir werden die Berechnungen für die Lokalisierung *offline*, also nach einer kompletten Messfahrt, am PC durchführen, wobei die Daten in Echtzeit an den Versuchsrechner übertragen werden. Grundsätzlich ist auch eine Echtzeit-Positionsschätzung mit den hier behandelten Methoden realisierbar.

6.2.1 Die Sensoren des Fahrzeugs

Allgemein wird zwischen zwei Sensorarten unterschieden: Sensoren, die eine differentielle Information über die gewünschte physikalische Größe liefern, werden als propriozeptive Sensoren bezeichnet und Sensoren, die die gewünschte physikalische Größe messen, als exterozeptive Sensoren. Da dieses Auto nur mit propriozeptiven Sensoren ausgestattet ist, kann hier ein einmal gemachter Fehler bei der Lokalisierung nicht mehr ausgeglichen werden.

Zur Beobachtung der Bewegung, d. h. zur Messung von Weg und Winkel, dienen die im Folgenden beschriebenen Sensoren. An jedem Rad befindet sich ein Inkrementalgeber. Dieser Inkrementalgeber besteht im Wesentlichen aus einer durchsichtigen Scheibe mit einem Strichraster, die an der Achse befestigt ist. Die Scheibe dreht sich durch eine Lichtschranke, die daraus elektrische Impulse erzeugt. Die Anzahl der aufgetretenen Impulse ergibt eine Information über den zurückgelegten Weg. Durch eine zweite, versetzt angeordnete Lichtschranke wird die Information über die Bewegungsrichtung geliefert. Eine genauere Beschreibung folgt in Abschn. 6.2.3.

Als weitere Wegsensoren befinden sich zwei Beschleunigungssensoren an einem Metallblock, der fest mit dem Fahrzeugchassis verschraubt ist. Jeder Beschleunigungssensor misst die Beschleunigung in Richtung seiner Achse eines fahrzeugfesten Koordinatensystems. Es werden also Längs- und Querschleunigung erfasst. Eine genauere Beschreibung folgt in Abschn. 6.2.4. Außerdem befindet sich an dem Metallblock noch ein Gierratensensor. Dieser erfasst die **Gierrate**, also die Winkelgeschwindigkeit des Fahrzeugs um seine Hochachse. Eine genauere Beschreibung folgt in Abschn. 6.2.5.

Alle Sensoren werden periodisch mit der Abtastzeit $t_A = 50 \text{ ms}$ ($f_A = 20 \text{ Hz}$) abgetastet. Daraus ergibt sich die aktuelle Zeit $t_n = n t_A$, wobei n für den n -ten Abtastzeitpunkt steht.

6.2.2 Die zeitdiskreten Positionsgleichungen

Mit Hilfe der zeitdiskreten Positionsgleichungen lässt sich rekursiv die Position des Fahrzeugs bestimmen, das heißt, es wird aus der alten Position und Ausrichtung fortlaufend mit Hilfe der gemessenen Weg- und Winkeländerung die neue Position bestimmt. Folgende Punkte sind zu beachten:

- Das Auto bewegt sich in der Ebene; eine kreisförmige Bewegung über zwei Abtastperioden stellt Abb. 6.1 dar.
- Die gemittelte Wegänderung Δs ist die Wegänderung, die der Mittelpunkt der Hinterachse zurücklegt.
- Der Ausrichtungswinkel Θ_n des Fahrzeugs wird bezüglich der x -Achse im mathematisch positiven Sinn dargestellt. Bei einer Linkskurve nimmt der Winkel dementsprechend zu.
- Die Änderung des Ausrichtungswinkels $\Delta\Theta$ während einer Abtastperiode entspricht genau dem Winkel des Kreissektors, was im Folgenden noch genauer gezeigt wird.

Zum Zeitpunkt $t_n = n \cdot t_A$ wird die Position des Fahrzeugs durch die x - und die y -Koordinate und die Ausrichtung durch den Ausrichtungswinkel Θ beschrieben. Nun soll die Position des Fahrzeugs zum neuen Zeitpunkt $t_{n+1} = (n + 1) \cdot t_A$ bestimmt werden. Dazu approximieren wir den Kreisbogen durch ein Geradenstück. Der dabei entstehende Fehler ist klein, wenn die Abtastzeit klein und dadurch der Kreisbogen kurz ist. Die geometrischen Beziehungen zur Berechnung der neuen Koordinaten zeigt das dick umrandete Dreieck in Abb. 6.2.

Mit Hilfe der Variablen $\Delta\Theta_n$ und Δs_n und den Zustandsvariablen x_n, y_n, Θ_n können wir die neue Position berechnen. Die Positionsgleichungen für die Koordinaten lauten:

$$x_{n+1} = x_n + \Delta s_n \cdot \cos\left(\Theta_n + \frac{\Delta\Theta_n}{2}\right), \quad (6.1)$$

$$y_{n+1} = y_n + \Delta s_n \cdot \sin\left(\Theta_n + \frac{\Delta\Theta_n}{2}\right). \quad (6.2)$$

Die Positionsgleichung für den Ausrichtungswinkel lautet

$$\Theta_{n+1} = \Theta_n + \Delta\Theta_n. \quad (6.3)$$

Diese drei Gleichungen sind also die zeitdiskreten Positionsgleichungen zur Bestimmung der neuen Position.

Man erkennt, dass der Ausrichtungswinkel Θ_n und die Änderung des Ausrichtungswinkels $\Delta\Theta_n$ in beiden Positionsgleichungen für die Koordinaten (6.1) und (6.2) nichtlinear mit der gemittelten Wegänderung Δs_n verknüpft werden. Daher ist es besonders wichtig, die beiden Winkelgrößen Θ_n und $\Delta\Theta_n$ so exakt wie möglich zu bestimmen.

Bemerkung zu den Winkeln

Zur Verdeutlichung von Abb. 6.2, bei der nicht unbedingt erkennbar ist, dass der Winkel $\frac{\Delta\Theta}{2}$ im Kreissektor gleich groß ist wie die halbe Winkelzunahme des Ausrichtungswinkels Θ , dient Abb. 6.3.

Die Gerade τ ist die Tangente an den Kreis und symbolisiert die Ausrichtung des Autos. Die beiden Dreiecke I und II sind ähnliche Dreiecke und es gilt daher:

$$\gamma = \frac{\Delta\Theta}{2} . \tag{6.4}$$

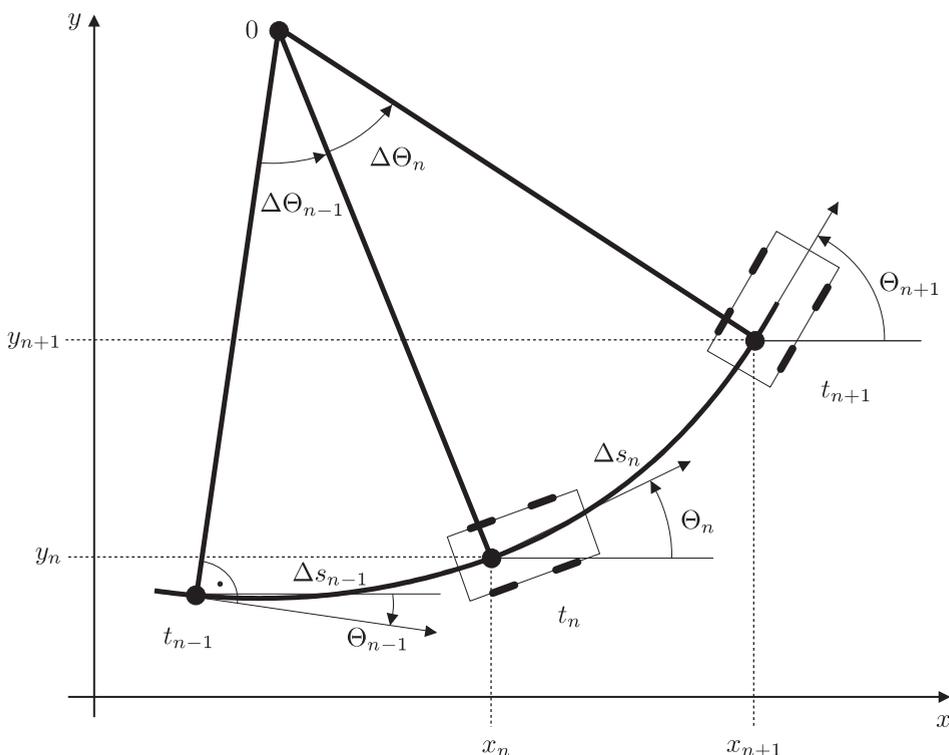


Abbildung 6.1. Bewegung während zwei Abtastperioden.

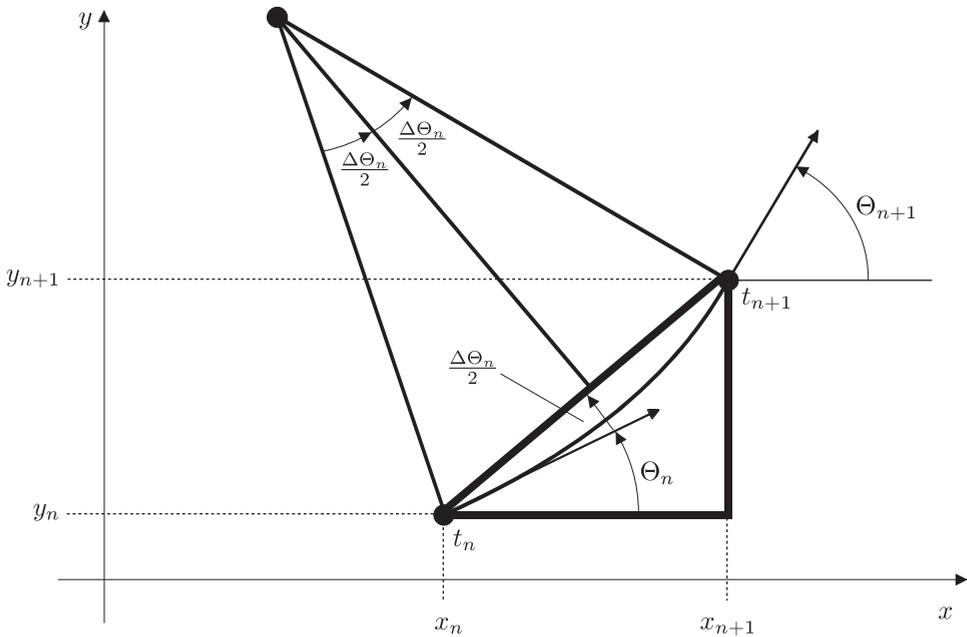


Abbildung 6.2. Das dick umrandete Dreieck als Grundlage zur Berechnung der neuen Position.

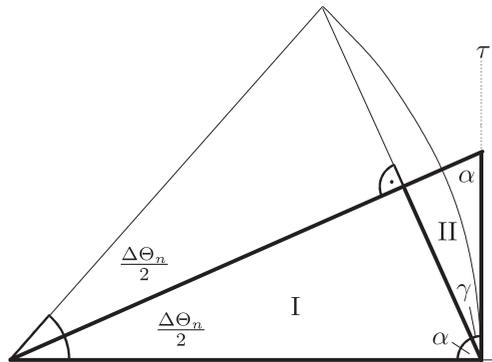


Abbildung 6.3. Darstellung der Verhältnisse bei der Winkeländerung.

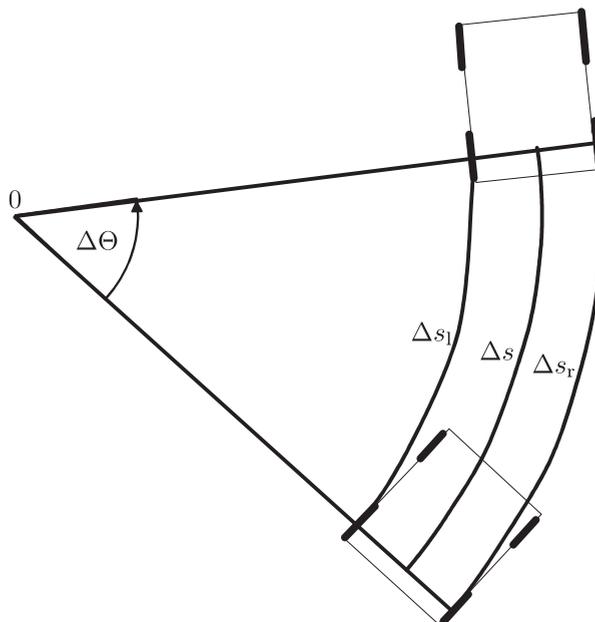


Abbildung 6.4. Darstellung der mittleren Weglänge.

6.2.3 Auswertung der Inkrementalgeberdaten

6.2.3.1 Berechnung der Wegänderung

Die Inkrementalgeber messen die Wegdifferenz zwischen zwei Abtastzeitpunkten. Dazu werden die Impulse des Inkrementalgebers während einer Abtastperiode gezählt. Die Wegdifferenz ergibt sich aus dem Wegelement pro Impuls und der Anzahl der gezählten Impulse. Im Inkrementalgeber dreht sich eine Scheibe mit 512 Strichen, durch eine eingebaute Übersetzung (Verhältnis 1:3) werden also 1536 Impulse pro Umdrehung gesendet. Ein Wegelement pro Impuls E wird mit dem Durchmesser D des Rades folgendermaßen berechnet:

$$E = \frac{\text{Radumfang}}{\text{Gesamtzahl Impulse}} = \frac{D \cdot \pi}{1536} \tag{6.5}$$

Mit der gemessenen Anzahl k der Impulse ergibt sich daraus der zwischen zwei Abtastperioden zurückgelegte Weg eines der Räder:

$$\Delta s_{\text{Rad}} = E \cdot k_{\text{Rad}} \tag{6.6}$$

Abbildung 6.4 zeigt die Verhältnisse bei einer Kreisfahrt um den Punkt 0, der Momentanpol genannt wird, während einer Abtastperiode. Dabei gilt:

Δs_{v1} : Wegänderung des linken Vorderrades,

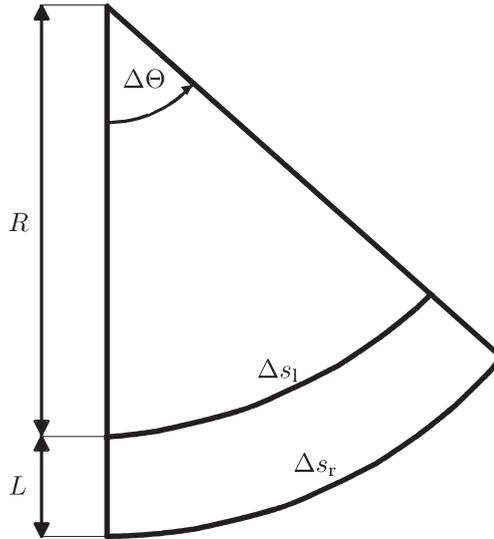


Abbildung 6.5. Darstellung der Winkeländerung.

- Δs_{vr} : Wegänderung des rechten Vorderrades,
- Δs_{hl} : Wegänderung des linken Hinterrades,
- Δs_{hr} : Wegänderung des rechten Hinterrades,
- Δs : mittlere Wegänderung.

Bei einer Kurvenfahrt sind die Wegänderungen der Räder unterschiedlich groß. Die mittlere Wegänderung ist der Mittelwert aller zurückgelegten Einzeldistanzen der vier Inkrementalgeber:

$$\Delta s = \frac{\Delta s_{vl} + \Delta s_{hr} + \Delta s_{vr} + \Delta s_{hl}}{4}. \quad (6.7)$$

6.2.3.2 Berechnung der Winkeländerung

Wenn der linke und rechte Inkrementalgeber gleich viele Impulse zählen, fährt das Auto geradeaus. Sind die Impulszahlen unterschiedlich, fährt es eine Kurve, die Ausrichtung des Fahrzeuges ändert sich. Dreht sich das rechte Rad schneller, fährt das Fahrzeug eine Linkskurve. Die Änderung des Ausrichtungswinkels ergibt sich aus der Differenz des rechten und linken Kreisbogens $\Delta s_r - \Delta s_l$ und der Spurbreite L :

$$\Delta \Theta = \frac{\Delta s_r - \Delta s_l}{L} \quad (6.8)$$

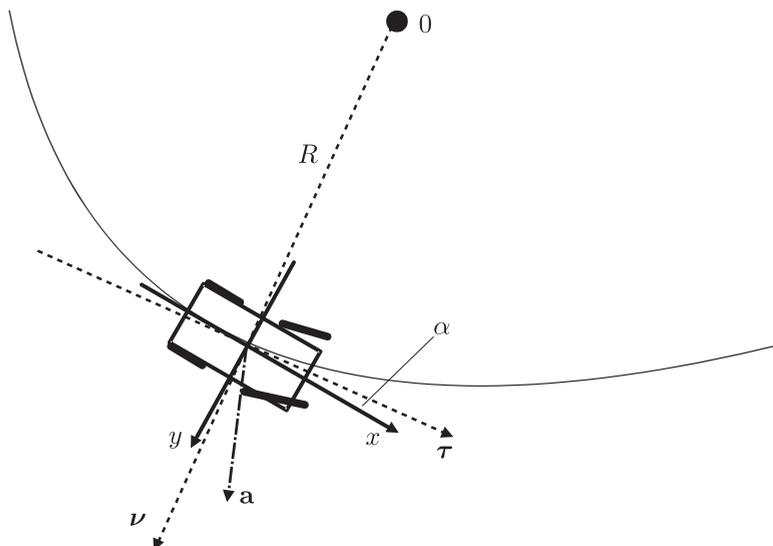


Abbildung 6.6. Resultierende Beschleunigung bei Kurvenfahrt.

mit $\Delta s_r = \frac{1}{2} (\Delta s_{vr} + \Delta s_{hr})$ und $\Delta s_l = \frac{1}{2} (\Delta s_{vl} + \Delta s_{hl})$. Diese Beziehung ergibt sich aus den beiden Gleichungen

$$\Delta s_l = \frac{\Delta \Theta}{2\pi} \cdot 2\pi R, \quad (6.9)$$

$$\Delta s_r = \frac{\Delta \Theta}{2\pi} \cdot 2\pi(R + L). \quad (6.10)$$

Durch Aufsummieren erhält man die momentane Ausrichtung:

$$\Theta_{n+1} = \Theta_n + \Delta \Theta_n. \quad (6.11)$$

6.2.4 Auswertung der Beschleunigungssensordaten

Die Beschleunigung ist eine vektorielle Größe. Nur bei Geradeausfahrt hat der Beschleunigungsvektor die gleiche Richtung wie die Fahrzeuglängsachse. Bei einer Kurvenfahrt stimmt die Richtung der resultierenden Beschleunigung nicht mehr mit der Fahrtrichtung überein. Den genaueren Sachverhalt verdeutlicht Abb. 6.6. Hierbei stehen die Variablen für folgende Größen:

- τ : Tangenten-Einheitsvektor – liegt tangential an der Bahnkurve,
- ν : Hauptnormalen-Einheitsvektor – liegt normal zur Bahnkurve,
- x : Längsachse des Fahrzeugs,
- a : resultierende Beschleunigung,
- α : Schwimmwinkel.

Die Geschwindigkeit ist tangential zur Bahnkurve – in Richtung τ – ausgerichtet. Die Beschleunigung setzt sich aus zwei Komponenten zusammen: aus einer Komponente tangential zur Bahnkurve – in Richtung τ – und aus einer Komponente normal zur Bahnkurve – in Richtung ν .

Die Beschleunigung in Tangentenrichtung heißt Tangentialbeschleunigung und hat den Betrag \dot{v} , wobei v die Bahngeschwindigkeit ist. Die Beschleunigung in Normalenrichtung ist die Zentripetalbeschleunigung. Sie hat den Betrag v^2/R . Aus beiden Komponenten ergibt sich durch vektorielle Addition die resultierende Beschleunigung:

$$\mathbf{a} = \dot{v} \boldsymbol{\tau} + \frac{v^2}{R} \boldsymbol{\nu}. \quad (6.12)$$

Die bisherigen Betrachtungen zur Beschleunigung gelten für Massenpunkte. Nun ist das Auto aber kein bewegter Massepunkt, sondern es hat geometrische Abmessungen. Die Mittelpunkte der beiden Fahrzeugachsen können sich in bestimmten Fahrsituationen auf unterschiedlich großen Kurvenradien bewegen. Dadurch liegt die Fahrzeuglängsachse nicht parallel zur Kurventangente τ und dazwischen entsteht ein Winkel, der als Schwimmwinkel α bezeichnet wird [MW04]. Die Beschleunigungssensoren nehmen die Beschleunigung in x - und y -Richtung – also in Richtung der Fahrzeugachsen – und nicht in Richtung der Bahntangente bzw. orthogonal dazu auf. Dadurch entsteht ein systematischer Messfehler. Der Fehler wird bei der Auswertung mit dem Kalman-Filter nicht berücksichtigt. Wir betrachten im Folgenden den Beschleunigungssensor in Längsrichtung als Sensor für die Tangentialbeschleunigung und den Sensor in Querrichtung als Aufnehmer für die Zentripetalbeschleunigung, machen also die Näherung:

$$a_x \approx \dot{v}, \quad a_y \approx \frac{v^2}{R}. \quad (6.13)$$

Es ergibt sich ein Fehler durch die Näherung, da der Schwimmwinkel α nicht berücksichtigt wird. Dieser Fehler kann nur korrigiert werden, wenn man zusätzlich die Messwerte der anderen Sensoren hinzuzieht.

Aus der Beschleunigung a_x in x -Richtung folgt die Geschwindigkeitsänderung

$$\Delta v = a_x t_A. \quad (6.14)$$

Durch Aufsummieren erhält man die Momentangeschwindigkeit:

$$v_{n+1} = v_n + \Delta v_n = v_n + a_{x,n} t_A. \quad (6.15)$$

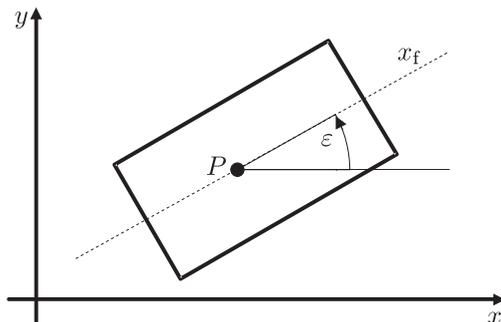


Abbildung 6.7. Darstellung des Gierwinkels.

6.2.5 Auswertung der Gierratensensordaten

Der Gierratensensor beruht auf dem piezoelektrischen Effekt. Mittels einer drehbar gelagerten Piezokeramik wandelt er dabei die Coriolis-Kraft in eine elektrische Spannung um, aus der die Winkeländerung berechnet wird. Bei einer ungeradlinigen Fahrt ändert sich die Ausrichtung des Fahrzeugs gegenüber einem ortsfesten Koordinatensystem.

Man denke sich das Auto drehbar um den Punkt P . Der **Gierwinkel** ε beschreibt die Schwenkung der fahrzeugfesten Koordinatenachse x_f gegenüber der raumfesten Koordinatenachse x . Die Ableitung des Gierwinkels nach der Zeit ist die Gierrate $\dot{\varepsilon}$. Bei Vernachlässigung des Schwimmwinkels entspricht die Gierrate physikalisch der Winkelgeschwindigkeit:

$$\dot{\varepsilon} \approx \omega = \frac{v}{R}. \quad (6.16)$$

Aus der Winkelgeschwindigkeit ω und der Abtastzeit t_A ergibt sich für die Winkeländerung während einer Abtastperiode:

$$\Delta\Theta = \omega t_A. \quad (6.17)$$

Durch Aufsummieren erhält man die momentane Ausrichtung:

$$\Theta_{n+1} = \Theta_n + \Delta\Theta_n. \quad (6.18)$$

6.2.6 Fehlerursachen bei der Lokalisierung

Es wurde die Herleitung der Positionsgleichungen bei optimalen Bedingungen beschrieben. Durch Unregelmäßigkeiten bei der Versuchsfahrt, wie z. B. Schlupf der Räder oder durch ungenaue Bestimmung der Raddurchmesser D und Achslänge L , können die Messergebnisse verfälscht werden. Um nach einer Versuchsfahrt richtige Schlüsse aus eventuellen Abweichungen bei der Positionsbestimmung ziehen zu können, werden wir die möglichen Fehlerursachen und ihre Auswirkungen in die-

sem Kapitel untersuchen. Man kann die auftretenden Fehler in systematische und zufällige Fehler aufteilen.

6.2.6.1 Systematische Fehler

Fehler durch Ungenauigkeiten in der Geometrie

Für eine gute Schätzung der Position ist es erforderlich, die Durchmesser D der Räder und die Spurweite L (d. h. den Abstand der Kontaktpunkte der Räder mit dem Untergrund) mit einer hohen Genauigkeit zu kennen. Da diese geometrischen Parameter den Berechnungen aus den Inkrementalgeberdaten zugrunde liegen (siehe (6.7) und (6.8)), führen schon kleine Ungenauigkeiten zu nicht mehr vernachlässigbaren Fehlern bei der Schätzung von Standort und Ausrichtung.

Fehler durch die Zeitdiskretisierung

Durch die zeitliche Diskretisierung der Messungen entstehen Fehler bei der Annäherung der real zurückgelegten Bahnkurven zwischen zwei Abtastpunkten durch Geradenstücke (siehe Abb. 6.2). Dieser Fehler wird groß bei schneller Fahrt mit engem Kurvenradius. Durch Erhöhung der Abtastfrequenz könnte der Fehler verkleinert werden.

Fehler durch Vernachlässigung des Schwimmwinkels

Bei Kurvenfahrt entspricht die Beschleunigung in x -Richtung nicht der tatsächlichen Ableitung der Bahngeschwindigkeit \dot{v} (siehe Abb. 6.6). Der Fehler ist bei einer engen Kurve (kleiner Kurvenradius) größer. Bei Geradeausfahrt verschwindet der Fehler.

Fehler durch die Offsetdrift der Sensoren

Wenn sich während der Messfahrt der Offset der Sensoren oder der Filter ändert, entsteht ein Fehler. Die ersten Messwerte jedes Sensors werden im Ruhezustand aufgenommen und dienen als Referenzwerte. Alle Messwerte werden mit dem Mittelwert der Referenzwerte verglichen. Eine Drift des Offsets während der Versuchsfahrt wird vom System allerdings nicht erkannt.

Fehler durch Verlagerung der Kontaktpunkte bei Kurvenfahrt

Bei der Kurvenfahrt verlagern sich die Kontaktpunkte zwischen Rad und Untergrund, dadurch ändert sich der Parameter L (Spurweite = Abstand der Kontaktpunkte der Hinterräder auf dem Untergrund).

6.2.6.2 Zufällige Fehler

Fehler durch das Auflösungsvermögen der Inkrementalgeber

Die kleinste Ortsveränderung, die gerade noch detektiert werden kann, entspricht dem Wegstück, das während eines Impulses zurückgelegt wird, also dem schon bekannten Wegelement E pro Impuls. Die verwendeten Inkrementalgeber geben mit der Übersetzung 1536 Impulse pro Umdrehung. Bei einem Durchmesser der Räder von 250 mm ist also der maximale absolute Fehler pro Abtastperiode:

$$F_{\text{abs}} = \frac{250 \text{ mm} \cdot \pi}{1536} = 0,5113 \text{ mm} . \quad (6.19)$$

Der relative Fehler ist geschwindigkeitsabhängig. Einer Anzahl k von Impulsen entspricht eine Geschwindigkeit v . Der relative Fehler bei dieser Geschwindigkeit beträgt

$$F_{\text{rel}} = \frac{1}{k} . \quad (6.20)$$

Quantisierung bei der A/D-Umsetzung

Die Ausgangssignale der Beschleunigungssensoren und des Gierratensensors werden vom A/D-Umsetzer digitalisiert. Dabei treten Quantisierungsfehler auf. Der maximale absolute Quantisierungsfehler ist so groß wie die halbe Auflösung U_{LSB} (kleinste darstellbare Spannung) des A/D-Umsetzers. Der relative Fehler ist abhängig von der Austeuerung des A/D-Umsetzers. Bei Vollaussteuerung ist er am kleinsten.

Fehler durch Unebenheiten des Geländes

Das Prinzip der Wegmessung ist für Messungen in der Ebene konzipiert. Wenn also Unebenheiten im Gelände auftreten (Bodenwellen, Vertiefungen, Türschwellen...), werden die Messergebnisse verfälscht, denn die zurückgelegte Distanz wird dann zu groß geschätzt. Im Versuchsraum befinden sich kleinere Unebenheiten, weshalb dieser Fehler tatsächlich auftreten kann.

Fehler durch Spiel zwischen Rad und Achse

Da das Rad vom Hersteller nicht ohne Spiel auf die Achse gesteckt werden kann, kommt es während der Messfahrt zu einem Wackeln aller vier Räder. Bemerkbar macht sich dieses Spiel durch ein stark verrauschtes Signal des y -Beschleunigungssensors, der durch die dabei entstehenden Vibrationen anspricht. Da man nicht genau sagen kann, wie welches Rad zum Fehlerzeitpunkt steht, handelt es sich um einen zufälligen Fehler.

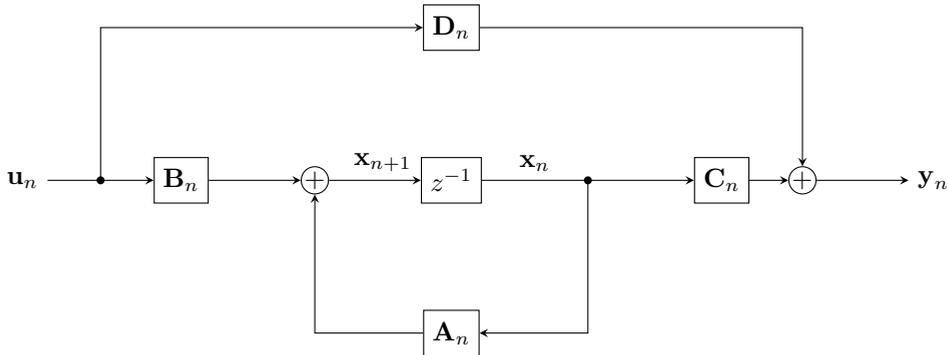


Abbildung 6.8. Zeitdiskretes System in Zustandsraumdarstellung [Bü91, BS94].

Fehler durch Schlupf der Räder

Bei starkem Beschleunigen und Abbremsen rutschen die Räder. Die Größe des Fehlers durch Schlupf hängt stark von der Durchführung der Messfahrt ab. Das Auto soll mit Gefühl so beschleunigt und um Kurven gelenkt werden, dass kein Schlupf auftritt und dieser Fehler somit verschwindet.

6.3 Datenfusion durch Kalman-Filterung

6.3.1 Zeitdiskrete Beschreibung linearer Systeme

6.3.1.1 Das zeitdiskrete System im Zustandsraum

Das Kalman-Filter geht, wie bereits erwähnt, von der Prozess- und Signaldarstellung im Zustandsraum aus.

In allgemeiner Form kann ein dynamisches System im Zustandsraum in vier funktionale Blöcke aufgegliedert werden. Der Zusammenhang zwischen Ausgangsvektor y_n und Eingangsvektor u_n wird dabei durch einen rekursiven und einen nichtrekursiven Block dargestellt, wie in Abb. 6.8 ersichtlich ist. Die Matrizen haben folgende Bezeichnung: Eingangsmatrix B , System- oder Dynamikmatrix A , Ausgangsmatrix C , Durchgangsmatrix D . Die Durchgangsmatrix D ist nur bei sprungfähigen Systemen vorhanden. Zur Beschreibung des Systems gehören neben dem Systemverhalten, das in den Matrizen enthalten ist, die Systemvariablen. Dieses sind Eingangsvektor u_n und Ausgangsvektor y_n sowie Zustandsvektor x_n , welcher den Zustand des Systems beschreibt. Zeitdiskrete dynamische Systeme im Zustandsraum werden durch die Zustandsgleichungen (eine Differenzgleichung und eine Ausgangsgleichung) beschrieben, die sich direkt aus dem obigen Zustandsdiagramm ablesen lassen. Die Differenzgleichung lautet

$$\mathbf{x}_{n+1} = \mathbf{A} \cdot \mathbf{x}_n + \mathbf{B} \cdot \mathbf{u}_n, \tag{6.21}$$

die Ausgangsgleichung

$$\mathbf{y}_n = \mathbf{C} \cdot \mathbf{x}_n + \mathbf{D} \cdot \mathbf{u}_n. \tag{6.22}$$

Das Ziel der Modellbildung im Zustandsraum ist es, quantitative Aussagen über das Verhalten des betrachteten Systems zu gewinnen. Diese Aussagen dienen dann der Beschreibung eines in der Gegenwart ablaufenden Vorgangs oder zur Vorhersage zukünftiger Ereignisse [BS94].

6.3.1.2 Zustandsschätzverfahren

Es besteht das Problem, aus dem gemessenen Ausgangsvektor \mathbf{y}_n den Zustandsvektor \mathbf{x}_n des Systems zu bestimmen. Leider ist aber der Zustandsvektor messtechnisch nicht zugänglich. Er muss deshalb aus den messbaren Größen des Systems berechnet werden, was als Zustandsschätzung bezeichnet wird. Abbildung 6.9 zeigt die Vorgehensweise bei einer Zustandsschätzung.

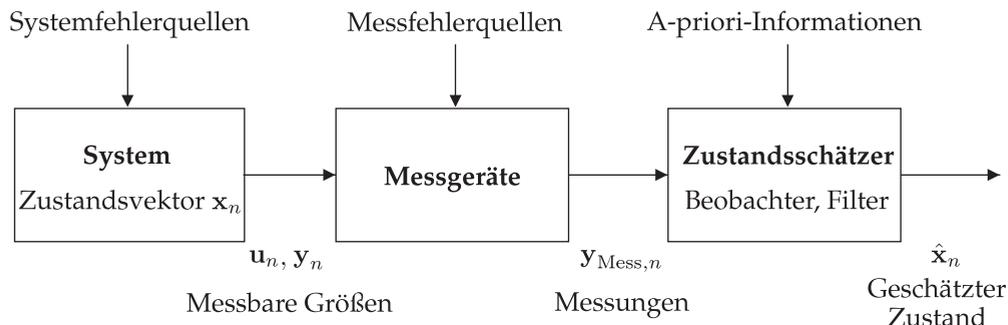


Abbildung 6.9. Zustandsdarstellung eines gestörten linearen dynamischen Systems [Bü91].

Zur Schätzung des Zustandsvektors wird ein Modell der Strecke eingesetzt. Im Modell der Strecke sind alle inneren Systemzustände abgreifbar. Abbildung 6.10 zeigt diese Anordnung.



Abbildung 6.10. Strecke mit Beobachter/Filter.

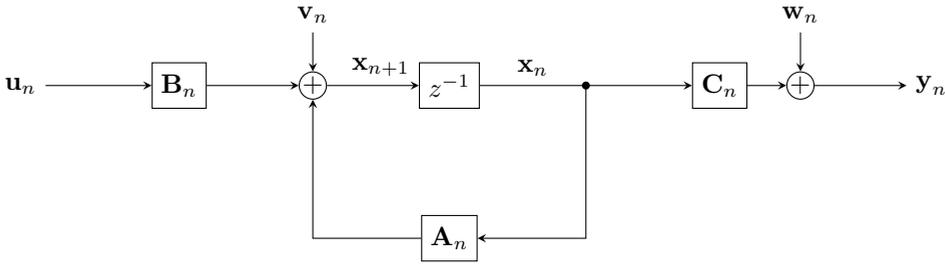


Abbildung 6.11. Signalprozess mit stochastischen Störgrößen.

Zustandsbeobachter und Zustandsfilter

Sind die Störgrößen des Systems so geringfügig, dass sie beim Entwurf des Zustandschätzers keine Rolle spielen, dann handelt es sich um einen Beobachter, andernfalls um ein Filter.

Das Filter hat die gleiche Struktur wie der entsprechende Beobachter [Fö08]. Der Unterschied besteht darin, dass die Verstärkungsgrade des Filters optimal bezüglich der gegebenen statistischen Eigenschaften der stochastischen Störgrößen sind, während die Verstärkungsgrade des Beobachters nach anderen Gesichtspunkten ausgewählt werden können [BS94]. Die Filteraufgabe ist also eine Verallgemeinerung der Beobachteraufgabe, bei der die stochastischen Störgrößen und Messfehler explizit in der Verstärkungsmatrix \mathbf{K} (siehe Abschn. 6.3.2.1) berücksichtigt werden. Wir werden im Folgenden auf die Filter eingehen, da das Kalman-Filter zur Klasse der Zustandsfilter gehört.

Werden also die stochastischen Störgrößen \mathbf{v}_n und \mathbf{w}_n berücksichtigt und die Durchgangsmatrix \mathbf{D} zu null angenommen, so verändert sich das System wie in Abb. 6.11 gezeigt. Die Störgröße \mathbf{v}_n am Eingang des Systems und der Messfehler \mathbf{w}_n am Ausgang des Systems seien mittelwertfreie, vektorielle weiße Gauß'sche Rauschprozesse in diskreter Zeit, mit gegebenen symmetrischen, positiv semidefiniten Kovarianzmatrizen:

$$E\{\mathbf{v}_n\} \equiv \mathbf{0}, \quad E\{\mathbf{w}_n\} \equiv \mathbf{0}. \quad (6.23)$$

Der Anfangszustand, der Stör- und der Messprozess seien gegenseitig unkorreliert:

$$\begin{aligned} E\{\mathbf{v}_n \cdot \mathbf{w}_m^T\} &\equiv \mathbf{0} && \forall m, n, \\ E\{\mathbf{v}_n \cdot \mathbf{x}_{n_0}^T\} &\equiv \mathbf{0} && \forall n, \\ E\{\mathbf{w}_n \cdot \mathbf{x}_{n_0}^T\} &\equiv \mathbf{0} && \forall n. \end{aligned} \quad (6.24)$$

6.3.2 Das Kalman-Filter

6.3.2.1 Theorie des Kalman-Filters

Das Kalman-Filter bildet einen Prozess im Zustandsraum nach [KSW08, Kro91, BS94, KRS11]. Es soll der unzugängliche Zustandsvektor \mathbf{x}_n eines dynamischen Systems auf Grund von Messungen der messbaren Ein- und Ausgangsvektoren geschätzt werden. Wegen der Messfehler liefert diese Berechnung keinen exakten, sondern einen Näherungswert. Dieser Wert ist der Schätzwert $\hat{\mathbf{x}}_n$ für den Zustandsvektor.

Ist etwas Näheres über den Messfehler bekannt, z. B. dessen Mittelwert, Streuung oder Varianz, dann lassen sich mit den Methoden der Wahrscheinlichkeitsrechnung bessere Schätzwerte erzielen [BS94]. Der Algorithmus des Kalman-Filters nutzt diese Kenntnisse über die Messfehler für seine Signalschätzung.

Es können – wie auch sonst bei den informationsverarbeitenden Operationen – drei grundlegende Arten der Kalman-Filterung unterschieden werden [KRS11]:

- **Kalman-Filterung (*filtering*)**: Der Ausgangsvektor \mathbf{y}_n hängt vom Eingangsvektor \mathbf{u}_k im Intervall $-\infty < k \leq n$ ab. Das bedeutet, dass die interessierende Information zum Zeitpunkt t ermittelt wird, indem Daten benutzt werden, die bis einschließlich t gemessen wurden. Das Filter ist kausal.
- **Kalman-Interpolation (*smoothing*)**: Der Ausgangsvektor \mathbf{y}_n hängt vom Eingangsvektor \mathbf{u}_k im Intervall $-\infty < k \leq n - \delta$, $\delta > 0$, ab. Das Filter ist kausal.
- **Kalman-Prädiktion oder -Extrapolation (*prediction*)**: Der Ausgangsvektor \mathbf{y}_n hängt vom Eingangsvektor \mathbf{u}_k im Intervall $-\infty < k \leq n + \delta$, $\delta > 0$, ab. Das Filter ist nicht kausal.

Die im Rahmen dieser Arbeit verwirklichten Kalman-Filter arbeiten nach dem Prinzip der Filterung.

Systemmodell und Funktionsweise

Zur Berechnung eines Systems mit messbaren Eingangsgrößen und stochastischen Störgrößen (nach Abb. 6.11) werden die Zustandsgleichungen (6.21) und (6.22) um die Störgrößen \mathbf{v}_n und \mathbf{w}_n erweitert:

$$\mathbf{x}_{n+1} = \mathbf{A} \cdot \mathbf{x}_n + \mathbf{B} \cdot \mathbf{u}_n + \mathbf{v}_n, \quad n \geq n_0, \quad (6.25)$$

$$\mathbf{y}_n = \mathbf{C} \cdot \mathbf{x}_n + \mathbf{w}_n. \quad (6.26)$$

Die Anfangsbedingung selbst oder ihr Erwartungswert sei gegeben:

$$E\{\mathbf{x}\}_{n_0} = \boldsymbol{\xi}, \quad (6.27)$$

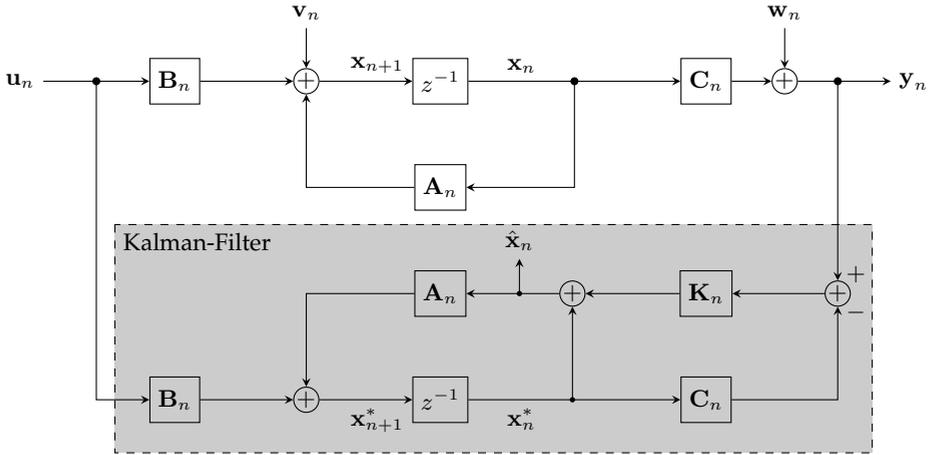


Abbildung 6.12. Beobachtetes System mit messbaren Eingangsgrößen und Kalman-Filter.

wobei ξ beliebige endliche Werte annehmen darf. Für die Störsignale v_n und w_n gilt:

$$E\{v_n v_m^T\} \equiv Q_n \cdot \delta_{n-m}, \tag{6.28}$$

$$E\{w_n w_m^T\} \equiv R_n \cdot \delta_{n-m}. \tag{6.29}$$

Im Fall eines eindimensionalen Filters bezeichnen Q_n die Varianz der Systemeingangsgröße und R_n die Varianz der Systemausgangsgröße.

Der Kalman-Schätzalgorithmus kann durch verschiedene Methoden hergeleitet werden [Bü91]:

- Orthogonalitätsprinzip von Schätzfehler und Messfehler,
- rekursive Methode der kleinsten Quadrate (RLS),
- Minimalschätzung – kleinste Varianz,
- Bayes-Methode.

Auf eine exakte Herleitung wird hier verzichtet.

Das Kalman-Filter arbeitet in zwei Schritten. Im **Prädiktionsschritt** wird mit Hilfe des letzten Schätzwertes \hat{x}_n , dessen Kovarianz \hat{P}_n und des Systemmodells die **A-priori-Dichte**

$$f(x_{n+1} | y_j, j = 1, \dots, n) \tag{6.30}$$

berechnet. Diese Wahrscheinlichkeitsdichte ist aufgrund der getroffenen Annahmen normalverteilt und kann durch ihre ersten beiden Momente vollständig beschrieben werden.

Im folgenden **Filterschritt** wird unter Berücksichtigung des neuen Messvektors y_{n+1} und der Informationen über die stochastischen Eigenschaften der Störeinflüsse

die **A-posteriori-Dichte**

$$f(\mathbf{x}_{n+1} \mid \mathbf{y}_j, j = 1, \dots, n+1) \quad (6.31)$$

bestimmt. Auch diese Wahrscheinlichkeitsdichte ist normalverteilt, sodass die Kenntnis ihrer ersten beiden Momente ausreicht.

Der Algorithmus wird nachfolgend angegeben. Der Schätzalgorithmus nach Kalman läuft in fünf Arbeitsschritten ab [BS94, BH92, KSW08], die für jeden Abtastwert wiederholt werden:

a) **Prädiktionsschätzwert (Erwartungswert der A-priori-Dichte)**

Der Schätzwert des alten Systemzustandes $\hat{\mathbf{x}}_n$ erfolgte unter Berücksichtigung der n vorhergehenden Messwerte (siehe die Berechnung des Filterschätzwerts im vierten Schritt):

$$\hat{\mathbf{x}}_n = E\{\mathbf{x}_n \mid \mathbf{y}_j, j = 1, \dots, n\}. \quad (6.32)$$

Damit wird jeweils zu jedem Abtastzeitpunkt eine Prädiktion um einen Schritt durchgeführt:

$$\mathbf{x}_{n+1}^* = \mathbf{A} \hat{\mathbf{x}}_n + \mathbf{B} \mathbf{u}_n, \quad \hat{\mathbf{x}}_{n_0} = \boldsymbol{\xi}, \quad (6.33)$$

wobei $\boldsymbol{\xi}$ beliebige endliche Werte annehmen darf.

b) **Prädiktionskovarianzmatrix (Kovarianzmatrix der A-priori-Dichte)**

Die Kovarianzmatrix von \mathbf{x}_{n+1}^* wird mit \mathbf{P}_{n+1}^* bezeichnet:

$$\mathbf{P}_{n+1}^* = \mathbf{A} \hat{\mathbf{P}}_n \mathbf{A}^T + \mathbf{Q}_n. \quad (6.34)$$

Sie stellt ein Maß für die Unsicherheit der Prädiktion dar.

c) **Verstärkungsmatrix**

Die Verstärkungsmatrix \mathbf{K}_{n+1} wird gestützt auf die Prädiktionskovarianzmatrix \mathbf{P}_{n+1}^* und die Varianz der Messstörung \mathbf{R}_{n+1} :

$$\mathbf{K}_{n+1} = [\mathbf{P}_{n+1}^* \mathbf{C}^T] [\mathbf{C} \mathbf{P}_{n+1}^* \mathbf{C}^T + \mathbf{R}_{n+1}]^{-1}. \quad (6.35)$$

d) **Filterschätzwert (Erwartungswert der A-posteriori-Dichte)**

Der Systemzustand $\hat{\mathbf{x}}_{n+1}$ wird zum Zeittakt $n+1$ geschätzt. Der geschätzte Zustand setzt sich aus dem vorhergesagten Zustand \mathbf{x}_{n+1}^* und der mit \mathbf{K}_{n+1} gewichteten Abweichung zwischen Messwert \mathbf{y}_{n+1} und dem vorhergesagten Systemausgang $\mathbf{C} \mathbf{x}_{n+1}^*$ zusammen:

$$\hat{\mathbf{x}}_{n+1} = \mathbf{x}_{n+1}^* + \mathbf{K}_{n+1} (\mathbf{y}_{n+1} - \mathbf{C} \mathbf{x}_{n+1}^*). \quad (6.36)$$

e) **Filterkovarianzmatrix (Kovarianzmatrix der A-posteriori-Dichte)**

Die Varianz des geschätzten Systemzustandes \hat{x}_n lautet:

$$\hat{P}_{n+1} = P_{n+1}^* - K_{n+1} C P_{n+1}^* \quad (6.37)$$

Tabelle 6.1. Bedeutung der verwendeten Größen.

K	Verstärkungsmatrix
x^*	vorhergesagter Systemzustand
\hat{x}	geschätzter Systemzustand
u	System-Eingangsgröße
y	System-Ausgangsgröße
P^*	Varianz des vorhergesagten Systemzustands
\hat{P}	Varianz des geschätzten Systemzustands
Q	Varianz des Systemrauschens
R	Varianz des Messrauschens

Die Verstärkungsmatrix ist lediglich eine Kurzschreibweise, welche die Darstellung der Momente der A-posteriori-Dichte vereinfacht. Anschaulich betrachtet bestimmt die Matrix, wie stark der Messvektor rückgekoppelt wird. Falls das Rauschen, das auf den Eingang des Systems wirkt, groß ist, kann das Systemmodell des Kalman-Filters keine zuverlässige Zustandsschätzung liefern. In diesem Fall muss der Messvektor verstärkt in die Schätzung einfließen. Umgekehrt muss die Rückkopplung geschwächt werden, wenn ein starkes Messrauschen vorliegt, da die Messdaten dann unzuverlässig sind. Dieser Zusammenhang wird im folgenden Abschnitt noch einmal für den skalaren Fall veranschaulicht.

Prinzip der minimalen Varianz

Die Verstärkungsmatrix K_n ist hinsichtlich der minimalen Varianz des geschätzten Zustandes optimal angelegt [Bü91]. Dies kann plausibel gemacht werden, wenn von der vektoriellen Darstellung aus (6.35) auf eine skalare Darstellung übergegangen und anschließend eine Grenzwertbetrachtung durchgeführt wird:

$$K_{n+1} = \frac{P_{n+1}^* C}{C P_{n+1}^* C + R_{n+1}} \quad (6.38)$$

mit

$$P_{n+1}^* = A^2 \hat{P}_{n+1} + Q_{n+1}. \quad (6.39)$$

- **Fall 1:** Ist das Messrauschen w_{n+1} gering im Vergleich zum Eingangsrauschen, d. h. $R_{n+1} \ll Q_{n+1}$, bedeutet dies im Grenzfalle, dass $K_{n+1} = 1/C$ ist. Wird also

$K_{n+1} = 1/C$ in die Schätzgleichung eingetragen, so gilt $\hat{x}_{n+1} = y_{n+1}/C$. Das Filter verlässt sich stark auf den Messwert, da dieser eine kleine Varianz besitzt.

- **Fall 2:** Das Messsignal ist im Verhältnis zum Eingangssignal mit sehr hohem Rauschen w_{n+1} behaftet, d. h. $R_{n+1} \gg Q_{n+1}$ ist sehr groß. Im Grenzfall bedeutet dies, dass $K_{n+1} = 0$ ist. Wird $K_{n+1} = 0$ in die Schätzgleichung eingetragen, so ist \hat{x}_{n+1} gleich dem vom Modell vorhergesagten Wert, also:

$$\hat{x}_{n+1} = x_{n+1}^* = A x_n + B u_n. \quad (6.40)$$

Die Messung wird nicht berücksichtigt.

Die Kernaussage des Kalman-Filters lautet also: In Abhängigkeit der Varianzen wird die Verstärkungsmatrix automatisch optimal eingestellt.

6.3.2.2 Das realisierte Kalman-Filter

Ohne auf das im nächsten Abschnitt zu behandelnde Datenfusionsmodell vorgreifen zu wollen, sei hier die Filterung der Winkelkomponente stellvertretend für die beiden gleich aufgebauten und im Programm verwirklichten Kalman-Filterstrukturen dargestellt. Zur Schätzung der Winkelkomponente stehen die Messdaten des Gierratensensors und des Inkrementalgebers zur Verfügung. Beide Sensoren liefern zunächst differentielle Informationen; allerdings wird das Signal des Gierratensensors bereits *vor* der Datenfusion aufintegriert, sodass eine absolute Größe zur Verfügung steht.

Das betrachtete dynamische System summiert zeitdiskrete Winkeländerungen auf:

$$\Theta_{n+1} = \Theta_n + \Delta\Theta_n, \quad n \geq n_0. \quad (6.41)$$

Die Zustandsgröße Θ des Systems ist der wahre Winkel. Als Eingangsgröße dient die aus den Inkrementalgebern bestimmte Winkeländerung $\Delta\Theta_{IG}$. Sie entspricht nicht der tatsächlichen Winkeländerung, sondern ist im Allgemeinen fehlerbehaftet:

$$\Delta\Theta_{IG,n} = \Delta\Theta_n + \tilde{v}_n. \quad (6.42)$$

Aufgelöst nach $\Delta\Theta$ ergibt dies:

$$\Delta\Theta_n = \Delta\Theta_{IG,n} - \tilde{v}_n = \Delta\Theta_{IG,n} + v_n, \quad n \geq n_0. \quad (6.43)$$

Dabei ist $v_n := -\tilde{v}_n$ ebenso wie \tilde{v}_n ein mittelwertfreier Gauß'scher weißer Rauschprozess.

Als Ausgangssignal steht nicht der tatsächliche absolute Winkel Θ zur Verfügung, sondern lediglich das gestörte integrierte Signal des Gierratensensors:

$$\Theta_{GR,n} = \Theta_n + w_n. \quad (6.44)$$

Damit ergeben sich zusammengefasst die folgenden vereinfachten Zustandsgleichungen (vergleiche (6.25) und (6.26)):

$$\Theta_{n+1} = \Theta_n + \Delta\Theta_{IG,n} + v_n, \quad n \geq n_0, \quad (6.45)$$

$$\Theta_{GR,n} = \Theta_n + w_n. \quad (6.46)$$

Das resultierende System für die Winkelkomponente mit Nachbildung durch das Kalman-Filter zeigt Abb. 6.13. Es stehen zwei Sensormesswerte zur Verfügung, und zwar die Winkeländerung und der daraus resultierende Absolutwinkel. Die Zustandsschätzung (6.32 bis 6.37) vereinfacht sich folgendermaßen:

a) **Prädiktionsschätzwert:**

$$\Theta_{n+1}^* = \hat{\Theta}_n + \Delta\Theta_{IG,n}, \quad \hat{\Theta}_{n_0} = \xi, \quad (6.47)$$

wobei ξ beliebige endliche Werte annehmen darf.

b) **Prädiktionsvarianz:**

$$P_{n+1}^* = \hat{P}_n + \sigma^2 (\Delta\Theta_{IG,n}). \quad (6.48)$$

c) **Verstärkungsfaktor:**

$$K_{n+1} = \frac{P_{n+1}^*}{P_{n+1}^* + \sigma^2 (\Theta_{GR,n+1})}. \quad (6.49)$$

d) **Filterschätzwert:**

$$\hat{\Theta}_{n+1} = \Theta_{n+1}^* + K_{n+1} \cdot (\Theta_{GR,n+1} - \Theta_{n+1}^*). \quad (6.50)$$

e) **Filtervarianz:**

$$\hat{P}_{n+1} = (1 - K_{n+1}) \cdot P_{n+1}^*. \quad (6.51)$$

6.3.3 Multisensor-Datenfusion mit dem Kalman-Filter

Die Multisensor-Datenfusion [AG92] ist eine auch in der Natur vorkommende Erscheinung. In zahlreichen biologischen Systemen werden Situationen durch eine Vielzahl von *Sensoren* ermittelt und gesteuert. So hat z. B. der Mensch die Fähigkeit, durch Kombination seiner Sinne (Sehen, Hören, Riechen, Fühlen), die er mit Hilfe seiner *Körpersensoren* (Augen, Ohren, Nase, Finger) erfasst, sich ein Bild von der Umgebung zu machen. Ziel dieser Datenfusionsmodelle ist es nun, die fehlerhaft gemessenen, aber teilweise redundanten physikalischen Größen so zu verknüpfen, dass die Position des Modellfahrzeuges so genau wie möglich ermittelt wird. Bei einer geschickten Kombination ist es hierbei möglich, durch Verknüpfung der Sensordaten mehr Informationen zu gewinnen, als in den einzelnen Daten schon enthalten sind.

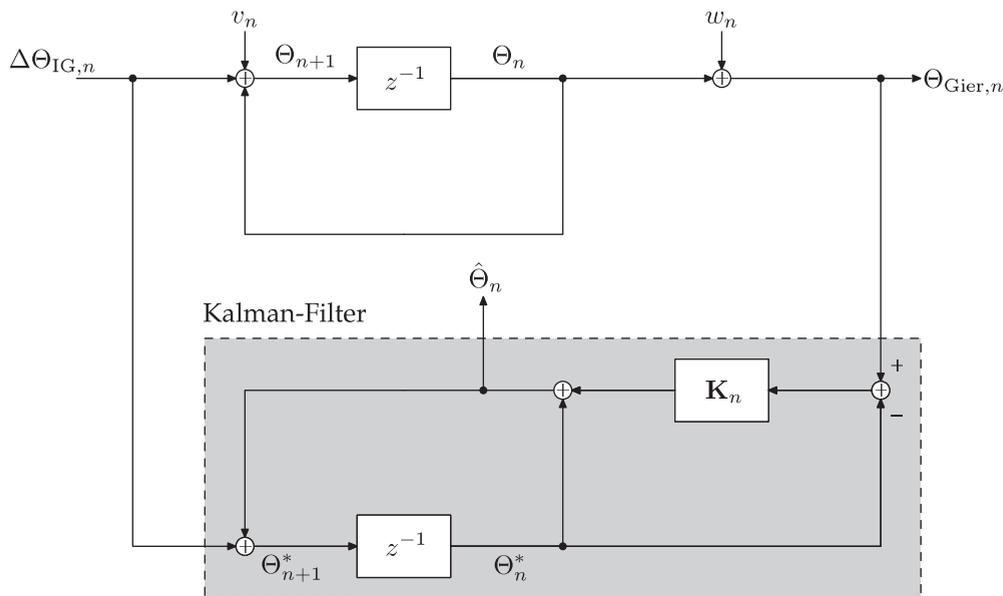


Abbildung 6.13. Realisiertes System zur Kalman-Filterung der Winkelkomponente.

Wie bereits in Abschn. 6.2 erläutert, sind auf dem Modellfahrzeug drei Sensoren (Inkrementalgeber, Gierratensensor und Beschleunigungssensoren) montiert. Die Messdaten aller Sensoren werden in zeitäquidistanten Abständen abgefragt und aufgezeichnet und stehen somit zur Positionsbestimmung zur Verfügung.

Wie ebenfalls in Abschn. 6.2 erläutert, erfolgt die Positionsbestimmung aufgrund der Positionsgleichungen (6.1) und (6.2). Die Sensoren beschreiben die neue Position durch eine Winkel- und eine Wegänderung des Fahrzeugs. Diese beiden Größen, die in die Positionsgleichung eingesetzt werden müssen, sollen mittels Kalman-Filterung optimiert werden. Durch die große Anzahl von Sensoren stehen nach geeigneter Umrechnung auch eine große Anzahl von Größen für die Positionsbestimmung zur Verfügung. Die jeweils optimale Kombination herauszufinden und zu verwenden ist Aufgabe der Datenfusion, wie in Abb. 6.14 dargestellt ist. Eine optimale Datenfusion wird durch das Kalman-Filter erreicht.

Nicht alle Messgrößen der Sensoren, die bestimmt werden können, sind sinnvoll und brauchbar. Es stehen in unserem Fall für ein Datenfusionsmodell folgende sinnvolle Sensormesswerte zur Verfügung:

- Winkeländerung aus Inkrementalgeber,
- Winkeländerung aus Gierratensensor,
- Geschwindigkeitsänderung aus Beschleunigungssensor,
- Wegänderung aus Inkrementalgeber.

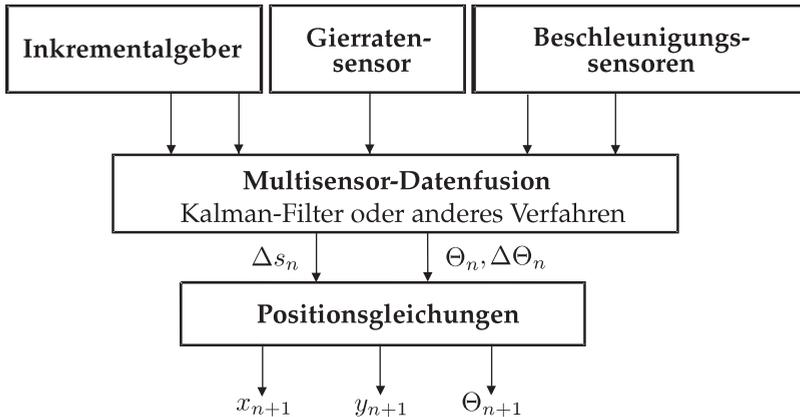


Abbildung 6.14. Positionsbestimmung mit Datenfusion.

Diese Sensormesswerte lassen sich nun für eine Kalman-Filterung kombinieren. Abbildung 6.15 zeigt die realisierte Datenfusion. In diesem realisierten Modell erfolgt die Positionsbestimmung mittels Kalman-Filterung über die Geschwindigkeits- und die Winkelmessung. Es wird von je einem System für Geschwindigkeit und Winkel ausgegangen. Die Messwerte der Sensoren müssen jeweils auf die physikalischen Größen Geschwindigkeit und Winkel gebracht werden. Messbarer Systemeingang ist jeweils die Deltagröße, messbarer Systemausgang die absolute Größe.

Neben den Messwerten benötigt das Kalman-Filter noch die Zuverlässigkeit der Messwerte in Form von Fehlervarianzen. Diese ergeben sich aus den Sensormessfehlern und müssen für jeden Takt neu bestimmt werden. Sie können auf verschiedene Arten bestimmt werden [KSW08]. Je nach Anwendungsfall können die Varianzen als konstant angenommen werden oder deren Zeitabhängigkeit berücksichtigt werden. Für einzelne Sensoren kann die Varianz im Datenblatt angegeben sein oder mittels statistischer Analysen selbst bestimmt werden. Für die Schätzung der Kovarianzmatrix des Messrauschens lässt sich folgende Formel verwenden [KSW08]:

$$\mathbf{R}_n \approx \frac{1}{N-1} \sum_{i=0}^{N-1} (\mathbf{y}_{n-1} - \bar{\mathbf{y}}_n) \cdot (\mathbf{y}_{n-1} - \bar{\mathbf{y}}_n)^T. \quad (6.52)$$

Hierbei steht $\bar{\mathbf{y}}_n$ für den gleitenden Mittelwert:

$$\bar{\mathbf{y}}_n = \frac{1}{N} \sum_{i=0}^{N-1} \mathbf{y}_{n-i}. \quad (6.53)$$

Im Versuch gilt für beide Systeme, also Weg und Winkel, dass die Varianz der direkten Größen Winkel und Geschwindigkeit mittels (6.52) und (6.53) aus den vergangenen 7 Messwerten bestimmt wird. Die Varianzen der Deltagrößen $\Delta v_{BE,n}$ und $\Delta \Theta_{IG,n}$ können im Programm entweder manuell gesetzt oder ebenfalls automatisch berechnet

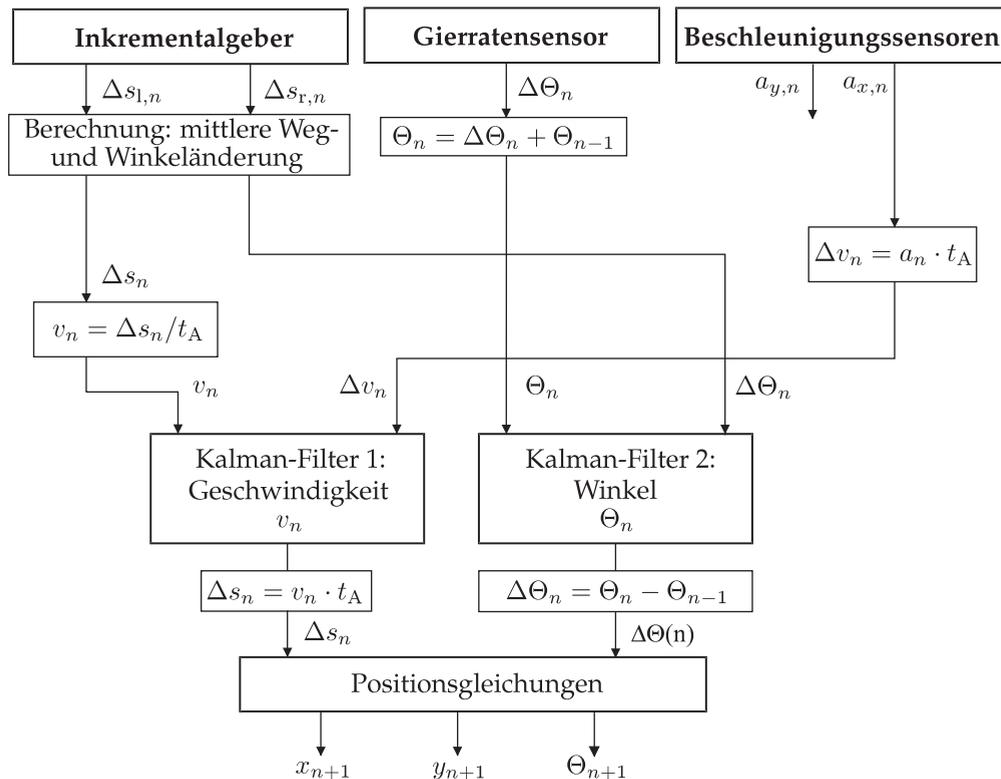


Abbildung 6.15. Datenfusionsmodell.

werden. Je nach Fahrweise und äußeren Rahmenbedingungen wie Untergrund etc. kann die eine oder andere Wahl sinnvoller sein.

Als Beispiel soll hier die Varianz des Beschleunigungssensors zum aktuellen Messwert n nach (6.52) und (6.53) berechnet werden. Der zu dieser Varianzschätzung benötigte Mittelwert aus den sieben vergangenen Messwerten errechnet sich folgendermaßen:

$$\overline{\Delta v_{BE,n}} = \frac{1}{7} \cdot \sum_{i=0}^6 \Delta v_{BE,n-i} \tag{6.54}$$

Mit Hilfe dieses Mittelwertes errechnet sich die Schätzung der Varianz zu:

$$\sigma^2(\Delta v_{BE,n}) = \frac{1}{6} \cdot \sum_{k=0}^6 (\Delta v_{BE,n-k} - \overline{\Delta v_{BE,n}})^2 \tag{6.55}$$

Eine andere Möglichkeit ist es, die drei vorhergehenden und die drei nachfolgenden Messwerte zu verwenden.

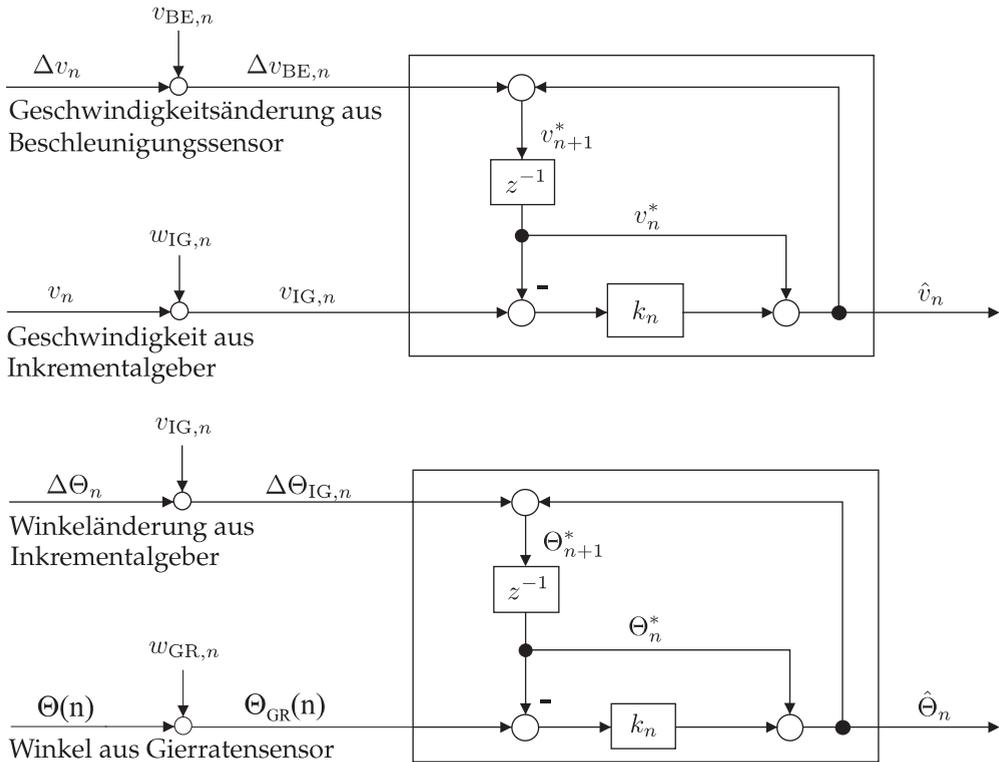


Abbildung 6.16. Die realisierten Kalman-Filterstrukturen.

Abbildung 6.16 zeigt schließlich das Blockdiagramm des ersten Kalman-Filters für die Winkelkomponente und des zweiten Kalman-Filters für die Geschwindigkeitskomponente.

6.4 Die Versuchssoftware

Um die Kalman-Filterung anwenden zu können, müssen die mit dem Fahrzeug aufgenommenen Daten dem Versuchsrechner zur Verfügung gestellt werden. Die Datenübertragung erfolgt dabei drahtlos. Auf dem Versuchsfahrzeug befindet sich ein Notebook, das mit USB-Datenerfassungseinheiten die Daten der vier Inkrementalgeber, der beiden Beschleunigungssensoren und des Gierratensensors erfasst. Mittels *Matlab* werden die Einheiten ausgelesen und per WLAN an den Praktikumsrechner gesendet, auf welchem die weitere Verarbeitung stattfindet. Die Programme auf dem Notebook und dem Versuchsrechner werden im Folgenden kurz vorgestellt, anschließend folgt eine Anleitung zur zeitlichen Abfolge der Bedienschritte (Abschn. 6.4.3).

6.4.1 Das Programm auf dem Fahrzeugrechner

Das Programm auf dem Fahrzeugrechner ist einfach gehalten und bietet lediglich die Möglichkeit, eine Messung zu starten und wieder zu beenden. Zum Starten des Programms geben Sie im Command Window `kalstart` ein. Es öffnet sich das Programm `kalstart`, siehe Abb. 6.17.

6.4.2 Das Programm auf dem Praktikumsrechner

Das Programm auf dem Versuchsrechner wird ebenfalls durch die Eingabe von `kalstart` im Command Window gestartet. Im oberen Teil des sich öffnenden Fensters (siehe Abb. 6.18) können die vier für die Positionsbestimmung wichtigen Größen Raddurchmesser D und Spurbreite L sowie die Varianzen für die Kalman-Filter-Eingangsgroßen $\Delta v_{BE,n}$ sowie $\Delta \Theta_{IG,n}$ individuell und unabhängig voneinander verändert werden. Bei den Varianzen kann zwischen festem Wert und automatischer Berechnung (siehe Abschn. 6.3.3) gewählt werden. Die Änderung der Werte wird erst nach Klick auf *Trajektorie neu berechnen und anzeigen*, *Datenübertragung starten* oder *Gespeicherte Datei plotten* wirksam.

Die Radio-Buttons im Teil *Wegberechnung* der Oberfläche gestatten die Auswahl des Berechnungsweges der Position des Messfahrzeuges. Da in den Positionsgleichungen Weg und Winkel benötigt werden, kann man für beide Größen jeweils genau eine Berechnungsart wählen. Die Funktionalität der Schaltfläche *Datenübertragung starten* wird in Abschn. 6.4.3 noch genauer erläutert werden. Nach Start der Datenübertragung öffnet sich ein neues *figure*-Fenster. In diesem wird die mittels der definierten Berechnungsarten für Weg und Winkel aus den aufgenommenen Messdaten ermittelte Position geplottet.

Unter dem *Datenübertragung starten*-Button befindet sich die Statusanzeige für die Messwertaufnahme. Unter dieser sind Buttons für das Speichern und erneute Laden von gespeicherten Messdaten platziert. Mit Klick auf *Trajektorie neu berechnen*

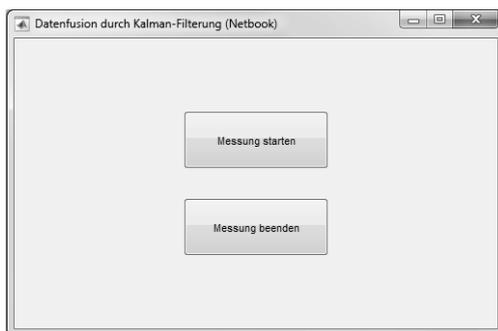


Abbildung 6.17. Bedienoberfläche des Programms auf dem Fahrzeug-Notebook.

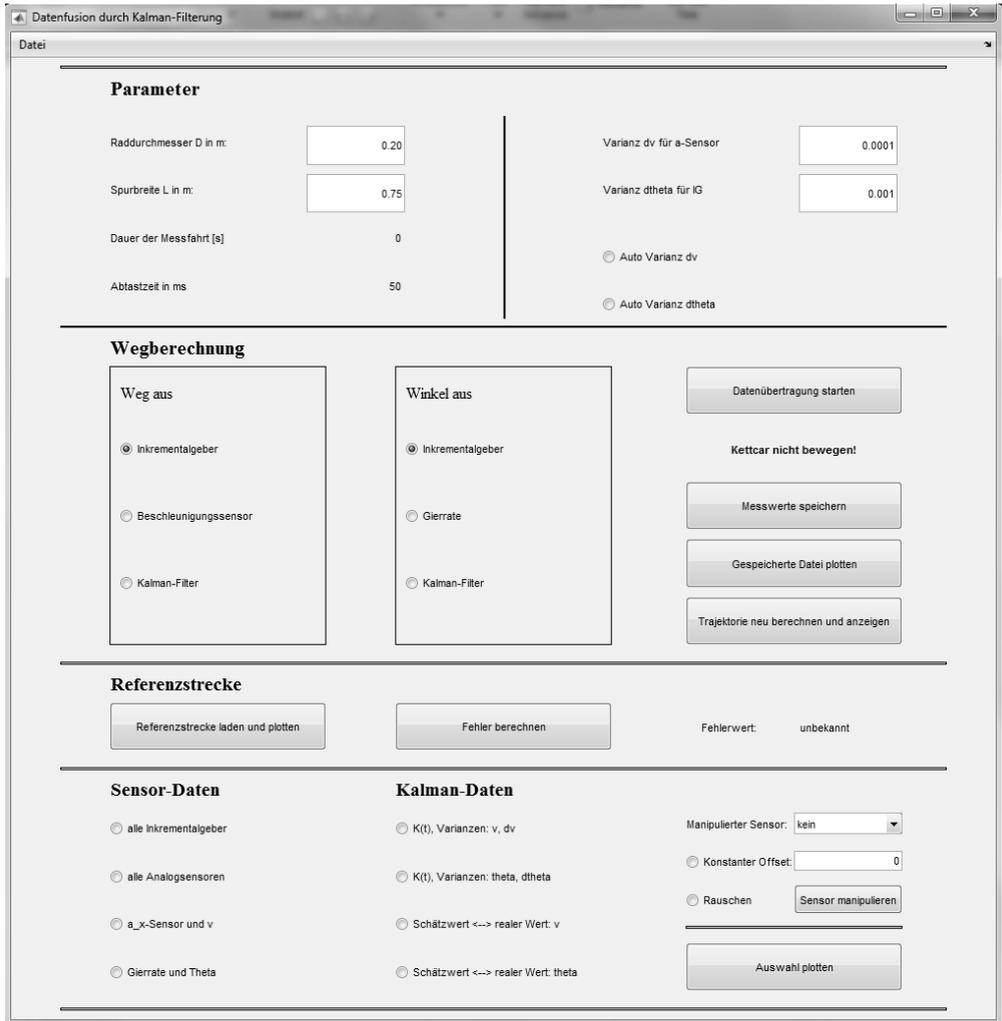


Abbildung 6.18. Bedienoberfläche des Programms auf dem Versuchsrechner.

und anzeigen wird nach einer Änderung der Berechnungsarten für Weg und Winkel die Trajektorie erneut geplottet.

Hinweis: Nach dem Einlesen aller Daten werden noch interne Korrekturen der Sensorwerte durchgeführt, die alle Werte und nicht nur die bereits eingelesenen benötigen. Dies bewirkt eine Verbesserung der Positionsschätzung und hat zur Folge, dass im Anzeigefenster eine zweite, korrigierte Kurve angezeigt wird.

Unter dem Titel *Referenzstrecke* befinden sich Buttons für das Laden einer gespeicherten Referenzstrecke und das Berechnen des Fehlers zwischen Referenzstrecke und tatsächlich gefahrener Strecke. Die Referenzstrecke wird nach dem Laden au-

tomatisch in das Positionsfenster eingezeichnet. Aufgrund der Tatsache, dass die Berechnung des Fehlers nicht über eine rein geometrische Betrachtung erfolgen kann, da bei dieser das Zeitverhalten der Fahrt nicht berücksichtigt würde, wird ein eigenes Fehlermaß verwendet. Bei rein geometrischer Betrachtung berechnete man lediglich die Abstände zwischen Punkten auf der Referenz- und der tatsächlich gefahrenen Strecke, beispielsweise die Abstände zwischen den zwei am nächsten liegenden Punkten von Referenz- und tatsächlicher Strecke. Die Abstände aller so ermittelten Punktpaare müssten aufsummiert und durch die Gesamtzahl der Punkte geteilt werden. Ein solches Fehlermaß könnte man täuschen, indem man das Fahrzeug sehr lange Zeit am Anfangspunkt stehen lässt, womit sich für sehr viele Punktpaare ein geringer Fehlerwert ergibt, der nach der Normierung auf die Gesamtzahl der Punkte den Gesamtwert künstlich nach unten drückt. Aus diesem Grund wird das Zeitverhalten der Fahrt berücksichtigt. Die Berechnung des Fehlermaßes beinhaltet die folgenden Schritte:

- a) Zunächst werden die gemessenen Daten durch Bildung des gleitenden Mittelwerts gefiltert.
- b) Aus der gemessenen Strecke wird über numerische Differentiation die Geschwindigkeit berechnet.
- c) Datenpunkte, an denen die Geschwindigkeit einen festgelegten Schwellwert unterschreitet, werden gelöscht. Dadurch werden Datenpunkte, an denen das Fahrzeug kaum bewegt wurde, nicht mehr berücksichtigt.
- d) Danach wird ein neuer Vektor erstellt, der die verbleibenden Datenpunkte enthält, wobei jeder Messpunkt im Verhältnis zu seiner Geschwindigkeit wiederholt wird. Betrug die Geschwindigkeit einen hohen Wert, wird der Punkt oft wiederholt, war sie niedrig, wird er nur wenige Male wiederholt.
- e) Anschließend werden aus dem neuen Vektor in gleichen Index-Abständen genauso viele Punkte, wie die Referenzstrecke enthält, verwendet. Zwischen diesen Punkten und den Referenzwerten wird der quadratische Fehler ermittelt, über alle Werte aufsummiert, anschließend die Wurzel gezogen und auf die Anzahl der Streckenpunkte normiert.

Dieses Verfahren stellt sicher, dass der berechnete Fehlerwert kaum von der gefahrenen Geschwindigkeit abhängt.

Im untersten Teil des Fensters können unter den Punkten *Sensor-Daten* und *Kalman-Daten* die Daten der einzelnen Sensoren sowie der Kalman-Größen wie Verstärkung etc. einzeln ausgewählt und geplottet werden (Klick auf *Auswahl plotten*). Rechts daneben befinden sich Schaltflächen zum künstlichen Verfälschen von Sensordaten, wobei der gewünschte Sensor durch ein Drop-Down-Menü ausgewählt wird. Zum Verfälschen stehen zwei Möglichkeiten zur Verfügung: Zum einen können die Sensordaten nachträglich verrauscht werden (SNR von 20 dB), zum anderen können die zeitabhängigen Sensordaten durch einen konstanten Wert (in der jeweiligen Ein-

heit des Sensors, also in m bei den Inkrementalgebern oder m/s bei den Beschleunigungssensoren) ersetzt werden. Werden beide Möglichkeiten ausgewählt, werden die Daten lediglich verrauscht. Sie können allerdings die Daten zunächst durch einen Offset ersetzen und in einem zweiten Schritt durch Wiederholen des Verfälschen-Prozesses verrauschen. Die jeweilige Manipulation geschieht durch Klick auf Sensor manipulieren.

Hinweis: Die Änderung der Sensordaten wird schon beim Klick auf Sensor manipulieren vorgenommen. Allerdings müssen aus diesen erst noch die Verläufe von Position und Winkel berechnet werden. Dies erfolgt erst nach einem erneuten Klick auf Trajektorie neu berechnen und anzeigen. Erst nach diesem lassen sich sowohl im Positionsfenster als auch in den einzelnen Sensorverläufen die Auswirkungen der Sensormanipulation beobachten.

6.4.3 Vorgehensweise bei der Datenaufnahme

Um eine Messfahrt mit dem Versuchsfahrzeug aufzunehmen, starten Sie sowohl auf dem Notebook als auch auf dem Versuchsrechner jeweils das Programm kalstart. Klicken Sie danach auf dem Notebook im sich öffnenden Fenster auf Messung starten und kurz danach auf dem Versuchsrechner auf Datenübertragung starten. Hierbei wird das Programmfenster auf dem Versuchsrechner inaktiv. Nach einigen Sekunden, in denen die Verbindung aufgebaut wird und Ruhewerte zur Offset-Korrektur der Analogsensoren aufgezeichnet werden, beginnt im Programmfenster auf dem Versuchsrechner der Zähler Dauer der Messfahrt zu laufen. Sobald sich das Fenster mit der Positionsanzeige geöffnet hat und in diesem die weiße Zeichenfläche zu sehen ist, kann das Kettcar bewegt werden. Nach Abschluss der Messung ist auf dem Notebook Messung beenden anzuklicken. Danach kann das Programmfenster auf dem Versuchsrechner wieder verwendet werden. Nach erfolgter Messung kann auf gleiche Weise eine neue Messung durchgeführt werden; die Programmfenster müssen nicht geschlossen und erneut geöffnet werden.

6.5 Versuchsvorbereitende Aufgaben

Aufgabe 6.1: Gierratensensor

- Spielt die Einbauposition des Gierratensensors in einem Fahrzeug eine Rolle?
- An welcher Stelle muss er eingebaut werden?

Aufgabe 6.2: Kalman-Filter vs. Luenberger-Beobachter

Nennen Sie die Unterschiede und die Gemeinsamkeiten von Kalman-Filter und Luenberger-Beobachter.

Aufgabe 6.3: Sensoraussetzer

Wie sieht das Ausgangssignal des Kalman-Datenfusionsfilters aus, wenn ein Sensorsignal ganz ausfällt?

- a) Annahme: Der Sensor liefert konstant den Wert null.
- b) Annahme: Der Sensor liefert ein Rauschsignal.

Aufgabe 6.4: Sensordrift

- a) Warum driftet der geschätzte absolute Winkel bei längerer Messfahrt vom tatsächlichen absoluten Winkel ab?
- b) Warum würde z. B. ein Erdmagnetfeldsensor helfen?

Aufgabe 6.5: Algorithmus des Kalman-Filters

Zur besseren Verdeutlichung kann der Algorithmus des Kalman-Filters wie in Abb. 6.19 graphisch veranschaulicht werden. Ergänzen Sie das Bild!

6.6 Versuche mit dem Fahrzeug

Im praktischen Teil werden nun verschiedene Messfahrten durchgeführt, um die Güte der einzelnen Sensormesswerte einschätzen zu können. Es folgt die Anwendung des Kalman-Filters auf die Messdaten zur Sensorfusion. Dabei wird untersucht, wie die verwendeten Sensorvarianzen (automatisch ermittelt, manuell konfiguriert etc.) das Fusionsergebnis beeinflussen. Somit wird eine umfassende Analyse des Verhaltens des Kalman-Filters vorgenommen. Dazu können die Sensordaten noch zusätzlich von Hand manipuliert werden.

6.6.1 Eingangsbemerkungen

Sollten Ihre Messungen unerwartete Ergebnisse liefern oder die Position fehlerhaft bestimmt worden sein, so begründen Sie Ihre Ergebnisse bitte anhand der Sensordaten, die Sie sich durch Klicken auf den gleichnamigen Button im Programm kalstart anzeigen lassen können.

Erstellen Sie unter `D:\PDSV\Gruppen` ein Verzeichnis für Ihre Gruppe. Legen Sie alle Dateien, die Sie im Laufe des Versuchs erstellen, nur in diesem Ordner ab!

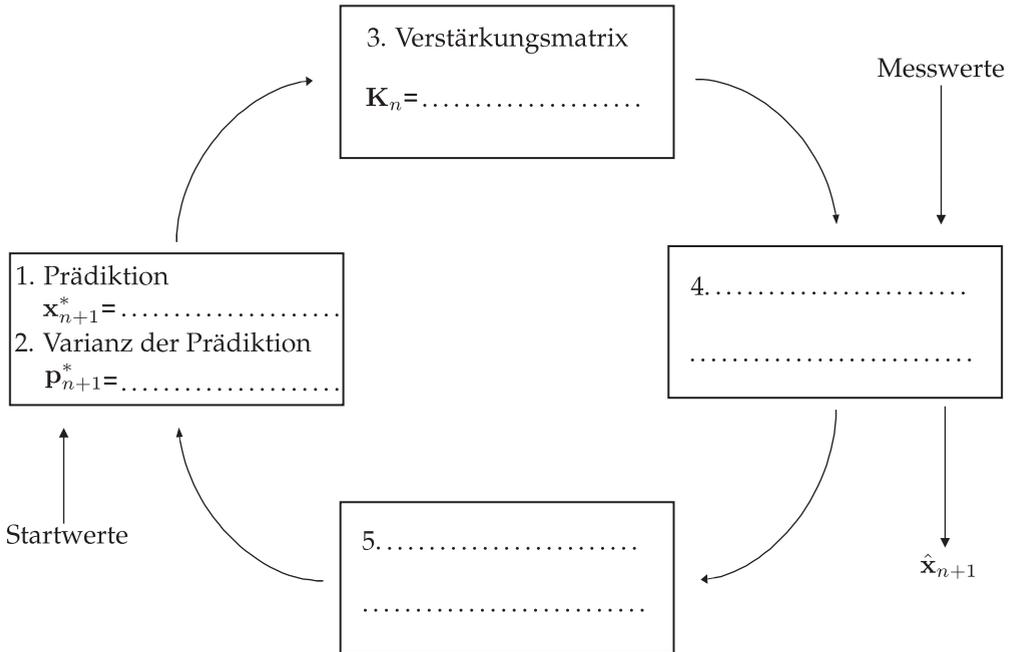


Abbildung 6.19. Graphische Veranschaulichung des Kalman-Algorithmus.

6.6.2 Messungen nur mit den Sensoren

Kalibrierung

Zuerst erfolgt eine bestmögliche Bestimmung, also Kalibrierung, der Auto-spezifischen Kennwerte Spurweite L und Raddurchmesser D .

Aufgabe 6.6: Messen der Parameter

Messen Sie mit dem Maßband die folgenden Parameter:

- Raddurchmesser D .
- gemessene Spurweite L .

Kreisfahrt

Unternehmen Sie mit dem Fahrzeug eine Kreisfahrt, wobei Start- und Endpunkt identisch sein sollen. Man kann so erkennen, wie groß die Abweichung der Schätzung der Endposition von der tatsächlichen Endposition ist.

Hinweis: Da der Offset der Beschleunigungssensoren korrigiert werden muss und deren Positionsschätzung mit zunehmender Messdauer zunehmend ungenau wird, empfiehlt es sich, die Fahrten mit dem Kettcar generell relativ zügig durchzuführen.

Dazu gehört auch, lediglich ungefähr ein bis zwei Sekunden, nachdem die Messung begonnen hat, loszufahren und nur ungefähr ein bis zwei Sekunden nach dem Anhalten die Messung zu beenden. Während der Ruhephase werden Ruhewerte zur Sensorkorrektur aufgezeichnet.

Aufgabe 6.7: Inkrementalgeber

Lassen Sie sich die Sensordaten der Inkrementalgeber anzeigen (Klick auf alle Inkrementalgeber im Bereich Sensordaten).

- a) Was fällt auf?
 - b) Wie lassen sich die Ergebnisse begründen?
-

Aufgabe 6.8: Schnelle Kurvenfahrt

Warum erhalten Sie schlechtere Ergebnisse, wenn Sie die Kurve schneller fahren?

Aufgabe 6.9: Beschleunigungssensoren

Warum liefern die Beschleunigungssensoren bei einer Kreisfahrt in der Regel das schlechtere Ergebnis bei der Berechnung des Weges?

Vergleich der gemessenen Daten

Aufgabe 6.10: Variation von D

Wie wirkt sich eine Änderung von D bei der Kreisfahrt aus?

Aufgabe 6.11: Variation von L

Wie wirkt sich eine Änderung von L aus?

Aufgabe 6.12: Messergebnisse

Welche optimalen Größen ergeben sich bei Messungen nur mit den Inkrementalgebern für:

- a) den Raddurchmesser D ,
 - b) die Spurweite L ,
- wenn beide Parameter variiert werden?

Hinweis: Variieren Sie dazu D und L so, dass im Positionsplot Anfangs- und Endpunkt der Kreisfahrt möglichst gut zusammenfallen.

Aufgabe 6.13: Messstörungen

Warum sind die anfangs von Hand gemessenen und die hier ermittelten Werte verschieden?

Aufgabe 6.14: Parameteränderungen

Können sich die „optimalen“ Parameter D und L auch zwischen einer Geradeausfahrt und einer Kreisfahrt ändern? Wie und warum?

Geradeausfahrt

Unternehmen Sie nun eine Geradeausfahrt, zunächst jedoch noch nicht entlang einer Bodenlinie!

Aufgabe 6.15: Parameteränderungen

Überprüfen Sie die Werte für den Raddurchmesser D aus dem vorigen Abschnitt, indem Sie die zurückgelegte Strecke von Hand messen und mit der geschätzten Weglänge vergleichen.

Aufgabe 6.16: Positionsbestimmung

Lassen Sie sich den zurückgelegten Weg nach Berechnung von Weg und Winkel durch die Inkrementalgeber anzeigen! Erhalten Sie eine exakte Gerade? Wenn nein, so begründen Sie, wie eine etwaige Abweichung zustande kommt!

Führen Sie nun eine Geradeausfahrt entlang der Bodenlinie durch!

Aufgabe 6.17: Positionsbestimmung mit IG

Plotten Sie auch noch einmal die über die Inkrementalgeber ermittelte Position! Haben sich die Ergebnisse gegenüber der anderen Geradeausfahrt verbessert? Wenn ja, woran könnte das liegen?

Messungen unter Zuhilfenahme des Kalman-Filters

Durch Variation der Fehlervarianzen sollen deren Einflüsse auf das Ergebnis untersucht werden. Diese Werte sind die Voraussetzung für eine optimale Positionsbestimmung mittels Kalman-Filterung.

6.6.2.1 Manuelle Varianzenvorgabe

Unternehmen Sie eine Messfahrt unter möglichst guter Raumausnutzung. Wählen Sie Anfangs- und Endposition wieder gleich.

Geben Sie nun die Varianzen für Δv_{BE} des Beschleunigungssensors und für $\Delta \theta_{IG}$ der Inkrementalgeber vor.

Hinweis: Es ergeben sich sehr kleine Werte für die Varianzen, scheuen Sie sich also nicht, diese im Bereich von 10^{-3} oder noch kleiner anzusetzen.

Lassen Sie sich wieder die Position anzeigen!

Aufgabe 6.18: Trajektorienberechnung

Lassen Sie sich den Weg über das Kalman-Filter und den Winkel über den Inkrementalgeber anzeigen. Anschließend vergleichen Sie den Verlauf des vom Kalman-Filter geschätzten Wertes v mit dem von den Inkrementalgebern gemessenen Verlauf von v_{IG} (Plot: Schätzwert \leftrightarrow IG Wert: v).

Wie muss die Varianz für Δv_{BE} gewählt werden, damit sich die mit Hilfe des Kalman Filters entstandene Kurve mit derjenigen nur aus den Inkrementalgeberwerten nahezu deckt? Warum muss $\sigma_{\Delta v, BE}^2$ so gewählt werden und was gilt dann für die Verstärkung K_n ?

Aufgabe 6.19: Varianzenvorgabe I

- Wie müssen Sie $\sigma_{\Delta \theta, IG}^2$ wählen, damit bei Berechnung des Weges über die Inkrementalgeber und des Winkels über das Kalman-Filter dasselbe Ergebnis erzielt wird wie bei Wahl von IG/Gierrate? Begründen Sie!
- Was gilt hier für K_n und warum?

Aufgabe 6.20: Varianzenvorgabe II

Wenn Sie beide Varianzen ($\sigma_{\Delta v, BE}^2$ und $\sigma_{\Delta \theta, IG}^2$) gleich 1 wählen, nach welchen Sensoren richtet sich das Kalman-Filter dann primär? Begründen Sie!

Aufgabe 6.21: Varianzenvorgabe III

Wie wirken sich Veränderungen der Fehlervarianzen allgemein auf das Verhalten des Kalman-Filters aus?

Aufgabe 6.22: Varianzenvorgabe IV

- Wenn Sie sowohl Weg als auch Winkel über das Kalman-Filter berechnen lassen, für welche Werte der Varianzen bekommen Sie dann die besten Ergebnisse?
- Plotten Sie für diese Werte die Verläufe von K_n und kommentieren Sie diese!

Automatische Varianzberechnung

In diesem Versuchsteil werden die Varianzen für Δv_{BE} und $\Delta \theta_{IG}$ automatisch bestimmt.

Aufgabe 6.23: Varianzvorgabe allgemein

- Vergleichen Sie die Auswirkungen der manuellen Varianzvorgabe mit der Online-Schätzung.
- Welches ist das bessere Verfahren?

Verwendung der Referenzstrecke

Aufgabe 6.24: Referenzstrecke

Auf dem Boden im Praktikumsraum ist eine Referenzstrecke markiert.

- Stellen Sie das Kettcar so auf die Strecke, dass sich der Markierstab über dem gekennzeichneten Anfangspunkt befindet und möglichst genau in Richtung der Strecke zeigt. Führen Sie eine Messfahrt entlang der Referenzstrecke durch, sodass sich der Markierstab an der Hinterachse möglichst exakt über der Bodenmarkierung befindet.
 - Werten Sie die Daten aus. Experimentieren Sie dabei mit verschiedenen Berechnungsverfahren (mit oder ohne Kalman-Filter) und Parameterwerten. Für welche Einstellungen erhalten Sie die genauesten Ergebnisse?
-

Aufgabe 6.25: Manipulation der Sensordaten

Nutzen Sie die im Programm vorgesehenen Schaltflächen zur Manipulation der Sensordaten.

- Wie ändert sich die Schätzung, wenn Sie für jeweils einen Sensor seinen Messwert als konstant gleich 0 einstellen? Probieren Sie alle Sensoren der Reihe nach aus! Speichern Sie zunächst die aufgenommenen Daten und manipulieren Sie den jeweiligen Sensor. Danach müssen die Daten erneut geladen werden, um wieder die Originaldaten zu bekommen und dann den nächsten Sensor verrauschen zu können.
- Können die anderen drei Inkrementalgeber den Ausfall des vierten kompensieren?
- Was ändert sich, wenn Sie, ebenfalls der Reihe nach, jeweils die Werte eines Sensors mit Rauschen versehen?

6.6.2.2 Abschließender Vergleich

Führen Sie nochmals eine Messfahrt entlang der Referenzstrecke durch!

Aufgabe 6.26: Vergleich

Vergleichen Sie folgende Verfahren der Positionsbestimmung und definieren Sie eine Reihenfolge der Güte:

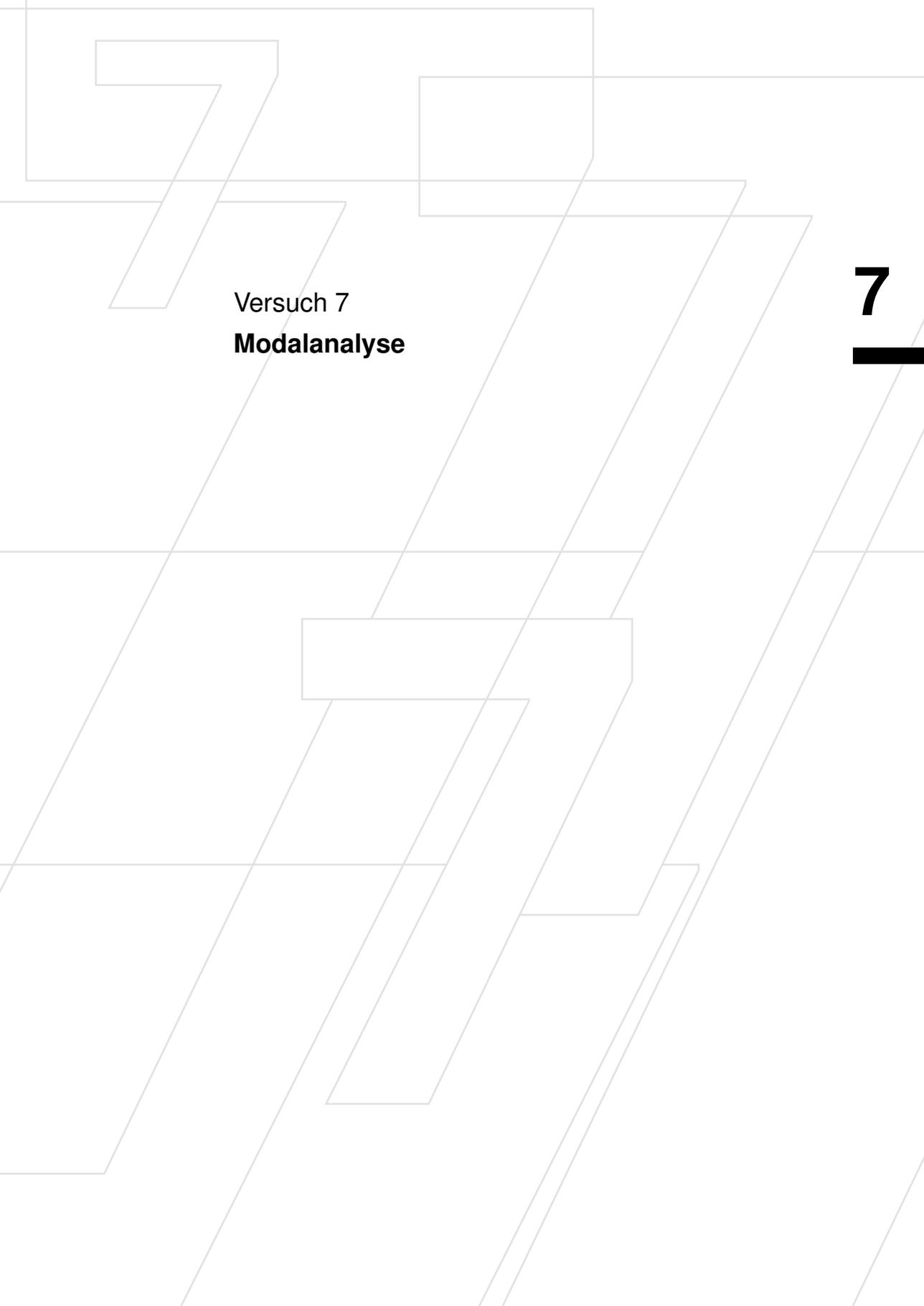
- Inkrementalgeberdaten,
- Position aus Beschleunigungs- und Gierratendaten,
- Kalman-Filterung mit manueller Varianzvorgabe,
- Kalman-Filterung mit automatischer Varianzvorgabe.

Aufgabe 6.27: Datenfusion

Weshalb kann bei Datenfusion mit dem Kalman-Filter das Ergebnis der Positionsbestimmung trotzdem schlechter ausfallen als bei Verwendung nur einer einfachen Sensorkombination, die allerdings gute Ergebnisse liefert?

6.7 Literatur

- [AG92] MONGI A. ABIDI und RAFAEL C. GONZALEZ (Herausgeber): *Data fusion in robotics and machine intelligence*. Academic Press, Boston, 1992.
- [BH92] ROBERT BROWN und PATRICK HWANG: *Introduction to random signals and applied Kalman Filtering*. John Wiley & Sons, New York, 2. Auflage, 1992.
- [BS94] KARL BRAMMER und GERHARD SIFFLING: *Kalman-Bucy-Filter – Deterministische Beobachtung und stochastische Filterung*. Oldenbourg, München, 4. Auflage, 1994.
- [Bü91] WINFRIED BÜTTNER: *Digitale Regelungssysteme – Grundlagen zum Systementwurf*. Vieweg, Braunschweig, 2. Auflage, 1991.
- [Fö08] OTTO FÖLLINGER: *Regelungstechnik: Einführung in die Methoden und ihre Anwendung*. Hüthig, Heidelberg, 10. Auflage, 2008.
- [Kro91] HEINZ KRONMÜLLER: *Digitale Signalverarbeitung*. Springer, Berlin Heidelberg, 1991.
- [KRS11] KRISTIAN KROSCHER, GERHARD RIGOLL und BJÖRN SCHULLER: *Statistische Informationstechnik – Signal- und Mustererkennung, Parameter- und Signalschätzung*. Springer, Berlin Heidelberg, 5. Auflage, 2011.
- [KSW08] UWE KIENCKE, MICHAEL SCHWARZ und THOMAS WEICKERT: *Signalverarbeitung – Zeit-Frequenz-Analyse und Schätzverfahren*. Oldenbourg, München, 2008.
- [MW04] MANFRED MITSCHKE und HENNING WALLENTOWITZ: *Dynamik der Kraftfahrzeuge*. Springer, Berlin Heidelberg, 4. Auflage, 2004.
- [PL15] FERNANDO PUENTE LEÓN: *Messtechnik – Systemtheorie für Ingenieure und Informatiker*. Springer, Berlin Heidelberg, 10. Auflage, 2015.
- [Sch77] KARL-WILHELM SCHRICK (Herausgeber): *Anwendungen der Kalman-Filter-Technik – Anleitung und Beispiele*. Oldenbourg, München, 1977.



Versuch 7
Modalanalyse

7

7	Modalanalyse	
7.1	Einleitung	227
7.2	Theorie der Modalanalyse	228
7.2.1	Voraussetzungen für die Modaltransformation	228
7.2.2	Ausführung der Modaltransformation	229
7.2.2.1	Entkopplung des gekoppelten Systems	229
7.2.2.2	Eigenschwingungen	232
7.2.3	Berechnung des Frequenzgangs	233
7.2.4	Einfluss verschiedener Dämpfungsansätze auf die Durchführung der Modaltransformation	236
7.3	Experimentelle Modalanalyse	237
7.3.1	Identifikation	237
7.3.2	Frequenzgang	237
7.3.3	Anregungssignale	238
7.3.3.1	Sinusförmige Anregung	239
7.3.3.2	Anregung mit Chirp-Signalen	239
7.3.3.3	Anregung mit transienten Signalen	240
7.3.3.4	Fast frequency sweep	243
7.3.3.5	Sprunganregung	243
7.3.3.6	Stochastische Anregung	243
7.3.3.7	Vor- und Nachteile der Anregungssignale	244
7.3.4	Praktische Durchführung der Modalanalyse	244
7.3.5	Parameterschätzung	245
7.4	Versuchsdurchführung	250
7.4.1	Hardware	251
7.4.2	Messdatenaufnahme in LabVIEW	251
7.4.3	Eigentliche Modalanalyse	253
7.4.3.1	Einlesen der Frequenzgänge	253
7.4.3.2	Modenbestimmung und -animation	254
7.4.4	Nützliche Matlab-Befehle	256
7.5	Versuchsvorbereitende Aufgaben	257
7.6	Versuchsaufgaben	258
7.6.1	Messanweisungen	258
7.6.2	Messungen und Versuchsdurchführung	259
7.6.2.1	Messdatenerfassung	259
7.6.2.2	Eigentliche Modalanalyse	260
7.7	Literatur	262

7 Modalanalyse

7.1 Einleitung

Mit dem zunehmenden Einfluss des Leichtbaus, dem Einsatz neuer Werkstoffe und den ständig wachsenden Anforderungen an Konstruktionen sind zu deren mathematischer Beschreibung immer komplexere Modelle erforderlich. Da schlanke Bauteile weniger Masse besitzen, sind sie anfälliger für Schwingungen, was sich etwa in einem erhöhten Geräuschpegel äußert. Damit ist man mit dem Problem einer möglichst präzisen Schätzung der Modellparameter (u. a. Massen, Dämpfungen und Steifigkeiten) konfrontiert.

Die **Modalanalyse** erlaubt eine Charakterisierung der dynamischen Eigenschaften schwingungsfähiger mechanischer Systeme. Im Rahmen dieser auch als **Eigen-schwingungsanalyse** bezeichneten Methode lassen sich die Modalparameter des Systems – **Eigenfrequenz**, **Dämpfung** und **Modenform** – experimentell ermitteln. Die Modalanalyse beruht auf der Tatsache, dass jede (erzwungene) dynamische Verformung einer Struktur als eine gewichtete Summe ihrer Modenformen darstellbar ist. Jeder Mode ist dabei durch die Schwingungsparameter einer (Sub-)Struktur mit genau einem Freiheitsgrad gegeben.

Der Begriff **Mode** ist der englischen Formulierung „*normal modes of vibration*“ entlehnt. Sie bezeichnet eine spezielle Schwingungsform der untersuchten Struktur, welche von anderen Formen entkoppelt werden kann und somit unabhängig ist. Für jeweils eine Mode schwingt jedes Strukturelement mit gleicher Frequenz und Phase. Bei Anregung einer Struktur durch eine Funktion gehen alle modalen Antworten anteilig in die gesamte Strukturbewegung ein. Durch gewichtete Summation dieser Antworten lässt sich die Schwingung der Struktur beschreiben. Somit charakterisieren die Modalparameter das dynamische Systemverhalten. Darüber hinaus stellt die Kenntnis dieser Parameter und des Frequenzgangs eine wichtige Grundlage für numerische Untersuchungen zur Modellbildung dar.

In der Praxis wird die untersuchte Struktur durch mechanische Krafteinkopplung zu dreidimensionalen Schwingungen angeregt, die z. B. über Beschleunigungssensoren gemessen werden. Nach der Signalverarbeitung können Eigenfrequenz, Dämpfung und Modenform aus der Beschleunigung berechnet werden. Im Karosseriebau kann mit diesem Wissen eine Aussage über Vibrationen getroffen werden.

Der vorliegende Versuch soll deutlich machen, wie die praktische Anwendung von Methoden der Signalverarbeitung zur Schwingungsanalyse einer Metallplatte verwendet werden kann. Im Wesentlichen stützt sich die Analyse auf den aus der die Fourier-Transformation abgeleiteten Frequenzgang. Zunächst wird in Abschn. 7.2 die physikalische Entstehung von Bauteilschwingungen erläutert. Anschließend befasst

sich Abschn. 7.3 mit der Theorie der Modalanalyse, wobei im ersten Teil die verwendeten Methoden zur Datenanalyse behandelt werden. Im zweiten Teil wird auf praktische Aspekte wie die geeignete Wahl der Anregungsform eingegangen, während der dritte Teil den Kreis schließt und erläutert, wie die Messsignale mit der physikalischen Schwingungstheorie in Einklang gebracht werden und aus diesen somit die Modalparameter bestimmt werden. Abschnitt 7.4 enthält die praktischen Versuchsdurchführung. Es werden zuerst Anregungssignale untersucht, die mit einem Impulshammer erzeugt werden. Anschließend werden die verschiedenen Moden einer dünnen Blechplatte ermittelt. Zuletzt erfolgt die Untersuchung des Schwingungsverhaltens eines Motorgehäuses.

7.2 Theorie der Modalanalyse

7.2.1 Voraussetzungen für die Modaltransformation

Die Modalanalyse ist prinzipbedingt nur bei mechanischen Schwingungssystemen möglich, die folgende Voraussetzungen erfüllen:

- linearisierbares Verhalten (linearisierte Bewegungsgleichung),
- zeitinvariantes Verhalten (zeitunabhängige Systemmatrizen),
- nur konservative Lagekräfte (keine schiefsymmetrischen Systemmatrizen oder gyroscopische Kräfte vorhanden).

Aus der Physik sind Bewegungsdifferentialgleichungen dynamischer mechanischer Systeme wie z. B. eines Feder-Masse-Dämpfer-Systems bekannt:

$$m\ddot{y} + d\dot{y} + cy = F. \quad (7.1)$$

Bislang wurden solche Gleichungen nur in einer Dimension betrachtet, d. h. es wurde eine Beschränkung auf Systeme mit *einem* Freiheitsgrad vorgenommen. Im Folgenden wird nun der Schritt auf Systeme mit n Freiheitsgraden vollzogen. Bei der Modalanalyse wird der untersuchte Körper in M Massenpunkte (sog. Strukturpunkte) diskretisiert. Jeder dieser Punkte kann Bewegungen in alle 3 Raumrichtungen ausführen, d. h. das Gesamtsystem besitzt $3M$ Freiheitsgrade. Durch sogenannte holonome Zwangsbedingungen (die z. B. vom Material abhängen oder auch willkürlich eingeführt werden können, um die Komplexität des Modells zu reduzieren), verringert sich die Anzahl der Freiheitsgrade zu n . Zum Beispiel hat ein System bestehend aus 2 Massenpunkten, die sich nur auf einer Geraden x bewegen können, 2 Freiheitsgrade, nämlich die Positionen x_1 und x_2 der Massenpunkte. Durch das Einfügen einer starren Kopplung zwischen den beiden Massenpunkten gilt $x_1 = x_2 + c$, wobei c der durch die Bindung vorgegebene Abstand zwischen beiden Massenpunkten ist. Somit sind die beiden ursprünglichen Freiheitsgrade nicht mehr unabhängig; die Zahl der Freiheitsgrade des Gesamtsystems reduziert sich zu 1.

Die Bewegungsgleichung eines Systems mit n Freiheitsgraden lässt sich allgemein in folgender Form angeben, wobei gyroskopische Kräfte und nichtkonservative Lagekräfte vernachlässigt wurden:

$$\mathbf{M} \ddot{\mathbf{y}}(t) + \mathbf{D} \dot{\mathbf{y}}(t) + \mathbf{K} \mathbf{y}(t) = \mathbf{f}(t). \quad (7.2)$$

Analog zum eindimensionalen Fall gilt in (7.2):

- $\mathbf{y}(t)$ stellt den n -dimensionalen Koordinatenvektor dar. Seine Elemente $y_i(t)$ beschreiben die Auslenkung der betrachteten Strukturpunkte aus der jeweiligen Ruhelage und somit die Lagekoordinaten im Konfigurationsraum. Durch zeitliche Ableitung des Koordinatenvektors lassen sich in dem hier vorausgesetzten Sonderfall der holonomen Bindungen die sogenannte Minimalgeschwindigkeit $\dot{\mathbf{y}}(t)$ und die Beschleunigung $\ddot{\mathbf{y}}(t)$ bestimmen.
- $\mathbf{f}(t)$ stellt den verallgemeinerten Kraftvektor dar. Die i -te Kraftkomponente $f_i(t)$ muss jeweils genau in Richtung der Koordinate y_i weisen.
- Die Massenmatrix $\mathbf{M} = \mathbf{M}^T$ ist stets symmetrisch und positiv definit.
- Die Dämpfungsmatrix $\mathbf{D} = \mathbf{D}^T$ ist symmetrisch.
- $\mathbf{K} = \mathbf{K}^T$ ist die Matrix der konservativen Lagekräfte oder auch die Steifigkeitsmatrix. Sie ist stets symmetrisch und positiv definit.

Gleichung (7.2) stellt ein System von n gekoppelten gewöhnlichen Differentialgleichungen dar. Eine anschauliche Interpretation eines gekoppelten Differentialgleichungssystems ist nicht möglich, auch die ingenieurmäßige Deutung ist bei einem entkoppelten System bedeutend einfacher. Die notwendige Entkopplung geschieht durch die in Abschn. 7.2.2 dargelegte Modaltransformation. Daran schließt sich die Berechnung von Frequenzgängen mit Hilfe der durch die Modaltransformation gewonnenen Größen an.

7.2.2 Ausführung der Modaltransformation

Die Theorie der Bewegungsdifferentialgleichungen mechanischer Schwingungssysteme wird hier der Einfachheit halber auf konservative Systeme beschränkt – also auf Systeme, in denen gyroskopische Kräfte, nichtkonservative Lagekräfte und Dämpfungen nicht berücksichtigt werden. Der Einfluss von Dämpfungstermen wird in Abschn. 7.2.4 kurz beschrieben. Das Vorgehen für Strukturen mit Dämpfung wird in [Bre88] behandelt.

7.2.2.1 Entkopplung des gekoppelten Systems

Ein konservatives Schwingungssystem mit n Freiheitsgraden wird durch die folgende Bewegungsgleichung in Matrixschreibweise beschrieben:

$$\mathbf{M} \ddot{\mathbf{y}}(t) + \mathbf{K} \mathbf{y}(t) = \mathbf{f}(t). \quad (7.3)$$

Ziel der Modaltransformation ist es, das gekoppelte System (7.3) zu entkoppeln. Für die Lösung des homogenen Differentialgleichungssystems

$$\mathbf{M} \ddot{\mathbf{y}}(t) + \mathbf{K} \mathbf{y}(t) = \mathbf{0} \quad (7.4)$$

wird der Ansatz

$$\mathbf{y}(t) = \mathbf{u} e^{\lambda t} \quad (7.5)$$

mit

$$\lambda = \underbrace{\delta}_{=0} + j\omega = j\omega \quad (7.6)$$

verwendet, da man für ungedämpfte Systeme nur Eigenfrequenzen ω und keine Dämpfungszahlen δ erhält. Durch Einsetzen von (7.5) in (7.4) folgt mit $e^{\lambda t} \neq 0 \forall t$ das allgemeine Eigenwertproblem:

$$(\lambda^2 \mathbf{M} + \mathbf{K}) \mathbf{u} = \mathbf{0} \quad \text{bzw.} \quad (-\omega^2 \mathbf{M} + \mathbf{K}) \mathbf{u} = \mathbf{0}. \quad (7.7)$$

Aus

$$\det(\lambda^2 \mathbf{M} + \mathbf{K}) = 0 \quad (7.8)$$

lassen sich die rein imaginären Eigenwerte λ_i mit $i = 1, 2, \dots, n$ bestimmen, aus denen sich wiederum die reellen Eigenfrequenzen ergeben. Für die Eigenwerte sind nur nichttriviale Lösungen der Gleichung zu berücksichtigen. Sofern alle Eigenwerte verschieden sind, gibt es zu jedem Eigenwert λ_i einen zugehörigen Eigenvektor \mathbf{u}_i , der das folgende lineare Gleichungssystem erfüllt:

$$(\lambda_i^2 \mathbf{M} + \mathbf{K}) \mathbf{u}_i = \mathbf{0}. \quad (7.9)$$

Da die Systemmatrizen als symmetrisch vorausgesetzt werden, sind ihre Eigenvektoren reell. Sie erfüllen die verallgemeinerten Orthogonalitätsbedingungen [Nat92]:

$$\mathbf{u}_i^T \mathbf{M} \mathbf{u}_j = m_i \delta_{ij}, \quad (7.10)$$

$$\mathbf{u}_i^T \mathbf{K} \mathbf{u}_j = k_i \delta_{ij} \quad (7.11)$$

mit dem **Kronecker-Delta**

$$\delta_{ij} = \begin{cases} 1 & \text{für } i = j \\ 0 & \text{für } i \neq j \end{cases}. \quad (7.12)$$

Für die weitere Rechnung werden zum einen die Modalmatrix

$$\mathbf{U} := [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n] \quad (7.13)$$

mit den Eigenvektoren als Spaltenvektoren und zum anderen die Diagonalmatrix der quadrierten Eigenwerte definiert:

$$\mathbf{\Lambda} := \text{diag}(\lambda_1^2, \dots, \lambda_n^2). \quad (7.14)$$

Damit lässt sich das Eigenwertproblem (7.7) nun als Matrixgleichung

$$\mathbf{M} \mathbf{U} \mathbf{\Lambda} + \mathbf{K} \mathbf{U} = \mathbf{0} \quad (7.15)$$

schreiben, womit sich für die verallgemeinerten Orthogonalitätsbedingungen folgende Beziehungen ergeben:

$$\tilde{\mathbf{M}} = \mathbf{U}^T \mathbf{M} \mathbf{U} = \text{diag}(m_i), \quad (7.16)$$

$$\tilde{\mathbf{K}} = \mathbf{U}^T \mathbf{K} \mathbf{U} = \text{diag}(k_i). \quad (7.17)$$

Dies bildet die Grundlage für die weiter unten durchgeführte Entkopplung der Differentialgleichungen.

Die Eigenvektoren sind nur bis auf einen Faktor bestimmbar. Dies eröffnet die Möglichkeit, sie beliebig zu normieren. So bietet es sich für Anwendungen beispielsweise an, die betragsmäßig größte Komponente des Eigenvektors $\equiv 1$ zu wählen:

$$\mathbf{u}_i^{\text{N1}} := \frac{\mathbf{u}_i}{\max_k (|\mathbf{u}_{i,k}|)}. \quad (7.18)$$

Für theoretische Untersuchungen ist z. B. die Normierung

$$\mathbf{u}_i^{\text{N2}} := \frac{\mathbf{u}_i}{\sqrt{m_i}} \quad (7.19)$$

praktisch, da sie auf die Orthonormierung

$$\tilde{\mathbf{M}} = \mathbf{I}, \quad \tilde{\mathbf{K}} = -\mathbf{\Lambda} \quad (7.20)$$

führt. Da sie von der jeweiligen Normierung abhängen, sind alle Größen $\tilde{\mathbf{M}}$, $\tilde{\mathbf{K}}$, m_i und k_i als generalisierte Größen zu verstehen.

Im Folgenden wird ebenso aufgrund der Normierung der generalisierte Koordinatenvektor $\mathbf{z}(t) = [z_1(t), z_2(t), \dots, z_n(t)]^T$ anstelle von $\mathbf{y}(t)$ weitergeführt.

Um schließlich die Entkopplung von (7.3) zu erreichen, wird zunächst eine Linksmultiplikation der Gleichung mit der transponierten Modalmatrix \mathbf{U}^T durchgeführt. Des Weiteren erfolgt eine Kongruenztransformation gemäß

$$\mathbf{y}(t) = \mathbf{U} \mathbf{z}(t). \quad (7.21)$$

Dieser Term wird in die durch Linksmultiplikation von (7.3) mit der Modalmatrix entstandene Gleichung eingesetzt, was faktisch einer Hauptachsentransformation ent-

spricht:

$$\mathbf{U}^T \mathbf{M} \mathbf{U} \ddot{\mathbf{z}}(t) + \mathbf{U}^T \mathbf{K} \mathbf{U} \mathbf{z}(t) = \mathbf{U}^T \mathbf{f}(t). \quad (7.22)$$

Der Ausdruck

$$\mathbf{h}(t) = \mathbf{U}^T \mathbf{f}(t) \quad (7.23)$$

stellt dabei einen Vektor gewichteter Kräfte dar.

Damit ist die Entkopplung vollzogen: Das System *gekoppelter* Differentialgleichungen 2. Ordnung aus (7.3) ist in n voneinander *unabhängige* Differentialgleichungen 2. Ordnung in den generalisierten (Normal-)Koordinaten $\mathbf{z}(t)$ überführt worden:

$$m_i \ddot{z}_i(t) + k_i z_i(t) = h_i(t) \quad \text{mit} \quad i = 1, 2, \dots, n. \quad (7.24)$$

Die Komponenten des Vektors $\mathbf{h}(t)$ ergeben sich zu

$$h_i(t) = \mathbf{u}_i^T \mathbf{f}(t) = \sum_{k=1}^n u_{i,k} f_k \quad \text{mit} \quad i = 1, 2, \dots, n. \quad (7.25)$$

Als Ergebnis erhält man also n unabhängige Einfreiheitsgradsysteme. In die Erregung $h_i(t)$ der entkoppelten Modalkoordinaten $z_i(t)$ gehen alle an der realen Struktur angreifenden Erregerkräfte $\mathbf{f}(t)$ ein. Jede Modalkoordinate $z_i(t)$ mit Ausnahme von Schwingungsknoten mit $u_{i,k} = 0$ kann also durch eine Kraft $f_k(t)$ am k -ten Strukturpunkt angeregt werden.

Die entkoppelten Differentialgleichungen (7.22) für $z_i(t)$ sind für sich alleine lösbar. Die Systemantwort $\mathbf{y}(t)$ im Konfigurationsraum erhält man durch Anwendung von (7.21).

Im folgenden Abschnitt werden die gewonnenen Ergebnisse physikalisch interpretiert.

7.2.2.2 Eigenschwingungen

Nimmt man den Kraftvektor $\mathbf{f}(t)$ zu null an, so beschränkt man sich auf die Lösung homogener Differentialgleichungen. Physikalisch bedeutet dies, dass das System bei beliebigen Anfangsbedingungen sich selbst überlassen wird. Es werden also die Eigenschwingungen des Systems berechnet. Jede der entkoppelten homogenen Differentialgleichungen aus (7.24)

$$m_i \ddot{z}_i(t) + k_i z_i(t) = 0 \quad \text{mit} \quad i = 1, 2, \dots, n \quad (7.26)$$

besitzt die allgemeine Lösung

$$z_i(t) = C_{1i} \sin(\omega_i t) + C_{2i} \cos(\omega_i t) \quad (7.27)$$

bzw. äquivalent dazu

$$z_i(t) = C_i \sin(\omega_i t + \alpha_i) \quad (7.28)$$

mit

$$C_i = \sqrt{C_{1i}^2 + C_{2i}^2}, \quad \alpha_i = \arctan \frac{C_{1i}}{C_{2i}}, \quad \omega_i = \sqrt{\frac{k_i}{m_i}}. \quad (7.29)$$

Aus jeder der n Gleichungen ergibt sich also eine Frequenz sowie ein Amplitudenfaktor und eine Phasenverschiebung. Die Frequenz definiert die Eigenschwingungsfrequenz eines Modes. Setzt man bis auf das i -te Element (also die i -te Eigenfrequenz) des Vektors \mathbf{z} alle anderen Elemente zu null, ergibt sich nur eine einzelne Eigenfrequenz. Einsetzen dieses Vektors

$$\mathbf{z} = [0, \dots, 0, z_i, 0, \dots, 0]^T \quad (7.30)$$

in (7.21) liefert:

$$\mathbf{y}_i = \mathbf{u}_i z_i(t) = \mathbf{u}_i C_i \sin(\omega_i t + \alpha_i). \quad (7.31)$$

Diese Gleichung zeigt die Bedeutung der Eigenvektoren \mathbf{u}_i : Während z_i die Eigenfrequenz der Eigenschwingung definiert, bestimmt der Eigenvektor (Modalvektor) die Eigenschwingungsform (= Mode) im Konfigurationsraum, d. h. die Maximalamplituden der harmonischen Schwingungen der jeweiligen Strukturpunkte entlang der freien Achse.

Dies soll an einem Beispiel erläutert werden: Ohne Beschränkung der Allgemeinheit sei hier α_i für alle i zu 0 angenommen. Für den Fall einer zweidimensionalen Platte in x - und y -Richtung und unter der Annahme, dass die Abstände der Strukturpunkte in der x, y -Ebene konstant bleiben (holonom gebunden), bleibt für jeden Strukturpunkt nur ein Freiheitsgrad in z -Richtung. Für den Zeitpunkt $\omega_i t = \pi/2$ nimmt jeder Strukturpunkt seine jeweilige Maximalamplitude in z -Richtung, definiert durch das jeweilige Element des Modalvektors, ein. Abbildung 7.1 zeigt eine solche Modenform. Für fortschreitende Zeit führt jeder Strukturpunkt für sich eine Schwingung mit der Frequenz ω_i aus. Zum Zeitpunkt $\omega_i t = 3\pi/2$ nimmt die Struktur die an der x, y -Ebene gespiegelte Modenform ein.

Im Allgemeinen setzt sich die Bewegung einer aus der Ruhelage ausgelenkten Struktur nach (7.21) aus der gewichteten Summe der einzelnen Eigenschwingungen zusammen.

7.2.3 Berechnung des Frequenzgangs

Nachdem die bisherigen Betrachtungen im Zeitbereich stattgefunden haben, soll nun beleuchtet werden, wo die Massen und Steifigkeiten bei Berechnungen im Frequenzbereich wieder auftauchen.

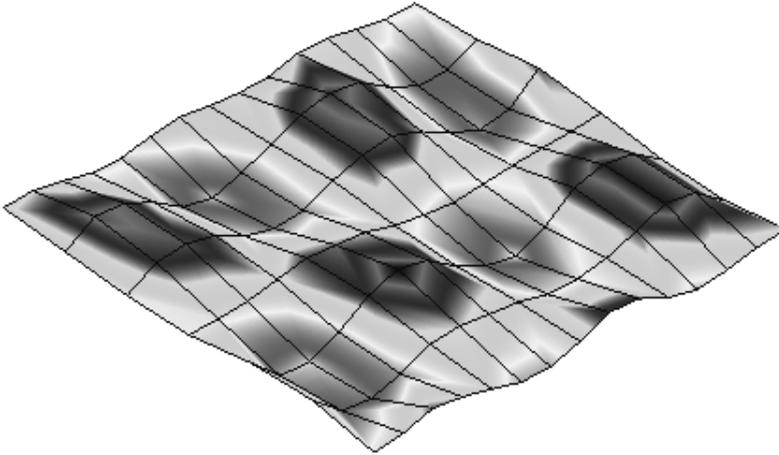


Abbildung 7.1. Modenform der verwendeten Metallplatte zur Eigenfrequenz von 1940 Hz.

Bei harmonischer Anregung lässt sich der Frequenzgang sehr einfach aus der partikulären Lösung der Bewegungsdifferentialgleichung berechnen. Die Lösungen der entkoppelten Differentialgleichung (7.22) im Modalraum liefern die Schwingungsantwort $\mathbf{z}(t)$. Bei einer harmonischen Anregung $\mathbf{f}(t) = \hat{\mathbf{f}} e^{j\Omega t}$ der realen Konfiguration entsteht im Modalraum entsprechend (7.23) die Erregerfunktion

$$\mathbf{h}(t) = \mathbf{U}^T \mathbf{f}(t) = \mathbf{U}^T \hat{\mathbf{f}} e^{j\Omega t} = \hat{\mathbf{h}} e^{j\Omega t} \quad (7.32)$$

mit

$$\hat{h}_i = \mathbf{u}_i^T \hat{\mathbf{f}} = \sum_{k=1}^n u_{i,k} \hat{f}_k \quad \text{und} \quad i = 1, 2, \dots, n. \quad (7.33)$$

Für die Schwingungsantwort $\mathbf{z}(t)$ wird ein Ansatz

$$\mathbf{z}(t) = \hat{\mathbf{z}} e^{j\Omega t} \quad (7.34)$$

wie bei der Anregung $\mathbf{f}(t)$ verwendet. Durch Einsetzen von (7.32) und (7.34) in das Differentialgleichungssystem (7.22) erhält man

$$-\text{diag}(m_i) \Omega^2 \hat{\mathbf{z}} e^{j\Omega t} + \text{diag}(k_i) \hat{\mathbf{z}} e^{j\Omega t} = \hat{\mathbf{h}} e^{j\Omega t}, \quad (7.35)$$

woraus sich mit $[\text{diag}(-\Omega^2 m_i + k_i)]^{-1} = \text{diag}\left(\frac{1}{-\Omega^2 m_i + k_i}\right)$ für die Amplituden $\hat{\mathbf{z}}$ der modalen Koordinaten

$$\hat{\mathbf{z}} = \text{diag}\left(\frac{1}{-\Omega^2 m_i + k_i}\right) \hat{\mathbf{h}} \quad (7.36)$$

ergibt. Geht man mittels der Modaltransformation $\hat{\mathbf{y}}(t) = \mathbf{U} \hat{\mathbf{z}}(t)$ wieder auf die ursprünglichen Konfigurationsraumkoordinaten zurück, so ergibt sich:

$$\hat{\mathbf{y}} = \mathbf{U} \operatorname{diag} \left(\frac{1}{-\Omega^2 m_i + k_i} \right) \hat{\mathbf{h}}. \quad (7.37)$$

Nach Einführung der Eigenwerte λ_i mit der Beziehung

$$\lambda_i^2 = -\frac{k_i}{m_i} \quad (7.38)$$

folgt aus (7.37) für die Systemantwort $\mathbf{y}(t)$ im Konfigurationsraum:

$$\hat{\mathbf{y}} = \mathbf{U} \operatorname{diag} \left(\frac{1}{-m_i(\Omega^2 + \lambda_i^2)} \right) \mathbf{U}^T \hat{\mathbf{f}}. \quad (7.39)$$

Mit (7.33) erhält man daraus für die m -te Amplitude ($m = 1, 2, \dots, n$) der Systemantwort im Konfigurationsraum

$$\hat{y}_m = [u_{1,m}, u_{2,m}, \dots, u_{n,m}] \begin{bmatrix} \vdots \\ \sum_{k=1}^n \frac{u_{i,k} \hat{f}_k}{-m_i(\Omega^2 + \lambda_i^2)} \\ \vdots \end{bmatrix} = \sum_{i=1}^n \left(\sum_{k=1}^n \frac{u_{i,m} u_{i,k} \hat{f}_k}{-m_i(\Omega^2 + \lambda_i^2)} \right). \quad (7.40)$$

Regt man nur an der Stelle k der Konfiguration an, entfällt in (7.40) die Summation über k , denn für die Amplitude der verallgemeinerten Anregung gilt

$$\hat{\mathbf{f}} = [0, \dots, 0, \hat{f}_k, 0, \dots, 0]^T. \quad (7.41)$$

Betrachtet man ferner nur eine Messstelle m , so ist der Frequenzgang von k nach m

$$G_{km} = \frac{\hat{y}_m}{\hat{f}_k} = \sum_{i=1}^n \frac{u_{i,m} u_{i,k}}{-m_i(\Omega^2 + \lambda_i^2)}. \quad (7.42)$$

Der Frequenzgang G_{km} ist immer eine Funktion der Erreger(kreis)frequenz Ω und enthält die modalen Parameter:

- Komponenten $u_{i,m}$ und $u_{i,k}$ der Eigenvektoren,
- modale Massen m_i (Elemente der diagonalisierten Massenmatrix $\tilde{\mathbf{M}}$),
- Eigenwerte λ_i (enthalten modale Massen und Steifigkeiten).

Der Frequenzgang (7.42) setzt sich somit aus Summanden zusammen, die jeweils die Systemantwort einer spezifischen Eigenschwingform darstellen (siehe Abb. 7.2). Wählt man die Normierung der Eigenvektoren nach (7.19), so ergeben sich die modalen Massen m_i jeweils zu 1. Die Eigenwerte λ_i entsprechen den Eigen(kreis)frequenzen der Modes. Somit sind in der Identifikation in Abschn. 7.3.1 nur noch die Komponenten $u_{i,m}$ und $u_{i,k}$ der (normierten) Eigenvektoren sowie die Eigenfrequenzen zu bestimmen.

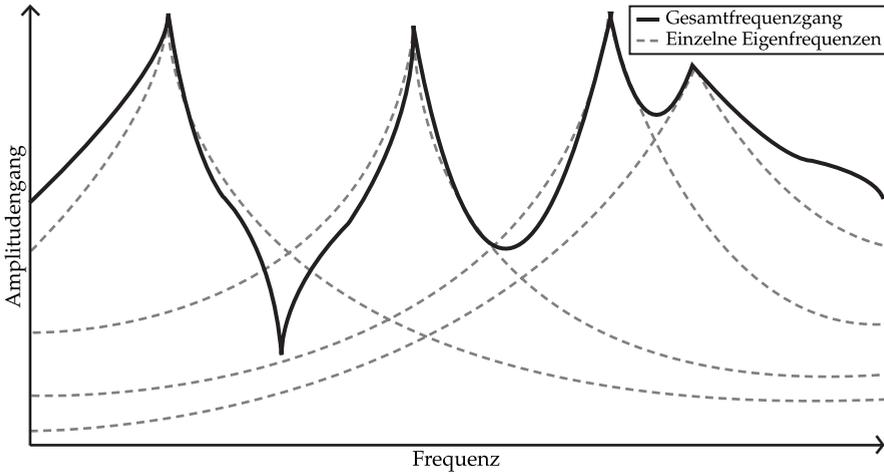


Abbildung 7.2. Der Gesamtfrequenzgang zwischen zwei Punkten einer Struktur setzt sich aus den Frequenzgängen der Eigenfrequenzen zusammen.

7.2.4 Einfluss verschiedener Dämpfungsansätze auf die Durchführung der Modaltransformation

Bei realen Strukturen treten stets Dämpfungen auf. Dabei gibt es:

- schwach (proportional) gedämpfte Systeme mit symmetrischer Dämpfungsmatrix,
- stark gedämpfte Systeme mit symmetrischer Dämpfungsmatrix,
- allgemeine Systeme mit schiefsymmetrischer Dämpfungsmatrix.

Elastische Strukturen ohne diskrete Dämpfungselemente erfahren oft nur eine geringe Materialdämpfung. Die im Abschn. 7.2.3 beschriebene Modaltransformation konservativer Systeme ist hier vollständig anwendbar, wenn die Dämpfungsmatrix proportional zur Massenmatrix oder zur Steifigkeitsmatrix oder gar zu beiden Matrizen ist. Im letztgenannten Fall gilt:

$$\mathbf{D} = \alpha \mathbf{M} + \beta \mathbf{K} \quad \text{mit} \quad \alpha, \beta = \text{const.} \quad (7.43)$$

Liegt Proportionaldämpfung vor, so bleiben die Eigenvektoren reell (klassische Eigenvektoren, wie beim konservativen Schwingungssystem), während die Eigenwerte konjugiert komplex werden. Auch die Dämpfungsmatrix erhält durch die Modaltransformation mit der Modalmatrix \mathbf{U} des konservativen Systems Diagonalgestalt:

$$\tilde{\mathbf{D}} = \mathbf{U}^T \mathbf{D} \mathbf{U} = \text{diag}(d_i). \quad (7.44)$$

Bei stark gedämpften Strukturen mit symmetrischer Dämpfungsmatrix reichen die reellen Eigenvektoren \mathbf{u}_i des konservativen Systems nicht mehr aus. Das Differentialgleichungssystem (7.2) wird im Zustandsraum mit der doppelten Dimension $2n$

dargestellt. Dort werden die komplexen Eigenwerte und Eigenvektoren für die Modalmatrix bestimmt.

Allgemeine Schwingungssysteme mit einer schief-symmetrischen Dämpfungsmatrix werden mit Hilfe der bimodalen Zerlegung (zwei Modalmatrizen bestehend aus Linkseigenvektoren und Rechtseigenvektoren) berechnet. Dieses Verfahren ist in [GK87] ausführlich beschrieben.

7.3 Experimentelle Modalanalyse

7.3.1 Identifikation

Zur Beschreibung eines dynamischen Systems werden im Allgemeinen dessen Differentialgleichungen aufgestellt. Häufig gelingt es jedoch nicht, eine geschlossene mathematische Systemdarstellung zu finden. Um dennoch zum Beispiel Systemparameter ermitteln zu können, bedient man sich Verfahren zur **Identifikation** dieser Systeme. Man unterscheidet prinzipiell zwischen parametrischer und nichtparametrischer Identifikation.

- Bei der **parametrischen Identifikation** steht die grobe Systemstruktur fest, es existiert also Vorwissen über das System. Daher kann mit einer kleinen Anzahl an Parametern das Verhalten beschrieben werden.
- Demgegenüber liegt bei der **nichtparametrischen Identifikation** kein Wissen über die Systemstruktur vor. Dies trifft bei der Analyse des Schwingverhaltens komplexer geometrischer Formen meist zu. Insbesondere zählt die Ermittlung des Frequenzgangs zur nichtparametrischen Identifikation.

Zunächst wird das Systemverhalten durch Auswertung der Antwort auf die Anregungssignale mittels Fourier-, Korrelations- oder Spektralanalyse analysiert. Bei der Fourier-Analyse des Schwingverhaltens eines Systems spielt insbesondere der Frequenzgang eine herausragende Rolle. Dazu werden in diesem Abschnitt auch verschiedene Anregungssignale sowie praktische Besonderheiten, die bei der Durchführung der Modalanalyse beachtet werden müssen, vorgestellt. Im nächsten Schritt wird das physikalische Modell nach Abschnitt 7.2 als gültig angenommen und dessen Parameter, nämlich die Modalparameter, im Rahmen einer parametrischen Identifikation bestimmt. Die Vorgehensweise dabei wird im Abschnitt über Parameterschätzung (7.3.5) behandelt.

7.3.2 Frequenzgang

Ein wichtiges Hilfsmittel zur Analyse linearer Systeme ist der **Frequenzgang** (engl. *frequency response function*, FRF). Aus der komplexen Wechselstromrechnung ist der

Frequenzgang bereits bekannt als

$$G(j\omega) = \frac{\underline{y}(\omega t)}{\underline{u}(\omega t)} = \frac{y_0(\omega) e^{j(\omega t + \varphi(\omega))}}{u_0(\omega) e^{j\omega t}} = \frac{y_0(\omega)}{u_0(\omega)} e^{j\varphi(\omega)}. \quad (7.45)$$

Diese Definition ist jedoch nur für harmonische Signale gültig. Die Ausweitung auf beliebige Signale ist mit Hilfe der Fourier-Transformation möglich. Betrachtet man bei linearen Systemen den Quotienten der Fourier-Transformierten aus Ausgangs- und Eingangssignal, so folgt:

$$G(j\omega) = \frac{\mathcal{F}\{y(t)\}}{\mathcal{F}\{u(t)\}} = \frac{Y(j\omega)}{U(j\omega)}. \quad (7.46)$$

Mit der Fourier-Transformierten des Dirac-Impulses $\delta(t)$,

$$\mathcal{F}\{\delta(t)\} = 1, \quad (7.47)$$

und der Impulsantwort $g(t)$ folgt [PJ15]:

$$G(j\omega) = \frac{\mathcal{F}\{g(t)\}}{\mathcal{F}\{\delta(t)\}} = \mathcal{F}\{g(t)\} = \int_{-\infty}^{\infty} g(t) e^{-j\omega t} dt. \quad (7.48)$$

Der Frequenzgang ist also die Fourier-Transformierte der Impulsantwort $g(t)$ und eignet sich somit hervorragend zur Beschreibung von Systemen im Frequenzbereich. Das Fourier-Integral eines Signals $x(t)$ lässt sich jedoch nur dann berechnen, wenn die folgenden Bedingungen erfüllt sind [Ise88]:

- die zu transformierende Zeitfunktion $x(t)$ ist stückweise stetig, monoton und an Unstetigkeitsstellen definiert;
- es gilt die Konvergenzbedingung:

$$\int_{-\infty}^{\infty} |x(t)| dt < \infty. \quad (7.49)$$

Der Frequenzgang ist eine komplexe Funktion. Üblicherweise spaltet man diesen für die Analyse in den Amplituden- und den Phasengang auf. Ersterer beschreibt die Verstärkung zwischen Ein- und Ausgangssignal über der Frequenz, wohingegen letzterer eine Aussage über die Phasenbeziehung zulässt. Üblicherweise wird der Amplitudengang doppelt logarithmisch und der Phasengang einfach logarithmisch aufgetragen. Diese Darstellung ist als **Bode-Diagramm** bekannt.

7.3.3 Anregungssignale

Bei der nichtparametrischen Identifikation wird die Systemantwort auf Anregungssignale untersucht. Zur Anregung eines Systems existieren zahlreiche Methoden. Ihre Anwendung ist abhängig von der zur Verfügung stehenden Messzeit, dem Equip-

ment zur Erzeugung der Testsignale oder von der Struktur des zu untersuchenden Systems. Im Folgenden werden einige dieser Verfahren vorgestellt und deren Vor- und Nachteile aufgezeigt.

7.3.3.1 Sinusförmige Anregung

Eine sinusförmige Erregung

$$p(t) = p_0 \sin \Omega(t) \quad (7.50)$$

ist mit geringem Aufwand realisierbar. Dabei müssen allerdings meist viele einzelne Frequenzen herangezogen werden, um das Systemverhalten im benötigten Frequenzbereich zu ermitteln. Deshalb ist diese Methode mit einem großen Zeitaufwand verbunden. Sie eignet sich aber dann, wenn die Resonanzfrequenzen aufgrund der Systemgeometrie grob bekannt sind und damit das Frequenzband bereits im Vorfeld eingeschränkt werden kann.

7.3.3.2 Anregung mit Chirp-Signalen

Wird die sinusförmige Erregung mit einer langsamen Gleitfrequenz durchgeführt, so erhält man ein **Chirp-Signal**, mit dem der Nachteil des zeitaufwändigen Messens (vgl. Abschn. 7.3.3.1) kompensiert werden kann:

$$p(t) = p_0 \sin \left(\left(\Omega_a + \frac{\Omega_b - \Omega_a}{2T} t \right) t \right), \quad (7.51)$$

wobei Ω_a die Startfrequenz, Ω_b die Endfrequenz und T die Periodendauer des Chirps sind. Die Frequenzerhöhung muss so langsam erfolgen, dass genügend Zeit zum Abklingen freier Schwingungen vorhanden ist.

Bei der Fourier-Transformierten des Chirp-Signals (Abb. 7.3) bewirkt die kontinuierliche Frequenzerhöhung, dass die Amplitude nicht über das komplette Anregungsfrequenzband konstant ist (Abb. 7.3 unten). Um diesen Fehler zu minimieren, sollten periodische Wiederholungen des Chirp-Durchlaufs mit anschließender Mittelwertbildung durchgeführt werden.

Beurteilung

Vorteilhaft an der harmonischen Anregung ist, dass die Signalenergie auf eine Frequenz konzentriert ist. Darüber hinaus ist die Frequenz gut steuer- und messbar. Weiterhin ist ein großes Signal-Rausch-Verhältnis vorhanden und Systemnichtlinearitäten (bei der Modalanalyse z. B. durch eine lokale Verletzung des Hooke'schen Gesetzes) können leicht aufgespürt werden. Der Hauptnachteil der Methode besteht im zeitaufwändigen Durchlaufen der einzelnen Frequenzen.

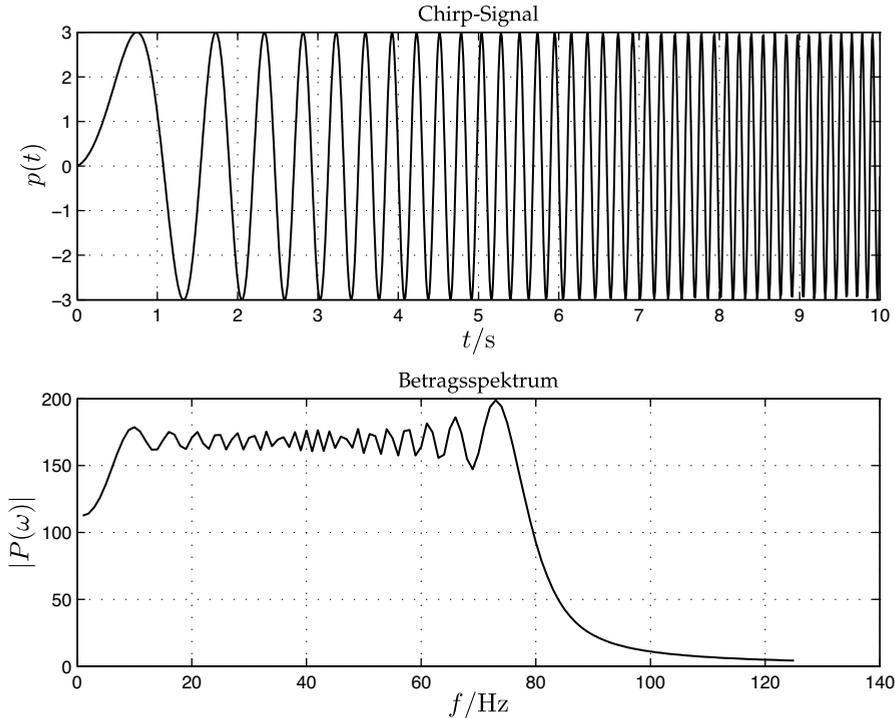


Abbildung 7.3. Chirp-Signal (oben) und zugehörige Fourier-Transformierte (unten).

7.3.3.3 Anregung mit transienten Signalen

Die Impulsanregung und die im Folgenden besprochenen weiteren Testsignale gehören zu den **transienten Signalen** (transient = vorübergehend), die im Gegensatz zu den harmonischen Signalen nicht periodisch sind.

Nach (7.47) ist der Dirac-Impuls ein ideales Anregungssignal, da mit ihm alle Frequenzen mit konstanter Amplitude im wahrsten Sinne des Wortes „auf einen Schlag“ angeregt werden. Leider ist es in der Praxis nicht möglich, einen idealen Dirac-Impuls zu erzeugen, weshalb dieser immer als ein mehr oder weniger breites (Zeit-)Fenster betrachtet werden muss.

Abbildung 7.4 zeigt eine Auswahl realisierbarer transienter Anregungssignale. In Abb. 7.5 sind die zugehörigen normierten Amplitudenspektren dargestellt. Man erkennt deutlich, dass das Rechtecksignal im unteren Frequenzbereich hohe Spektralwerte besitzt, diese jedoch schnell abfallen. Ferner sieht man, dass die anderen Signale höhere Frequenzen besser anzuregen vermögen als das Rechtecksignal. Auffällig ist zudem die Fähigkeit der Sprungfunktion, sehr niedrige Frequenzen stark anzuregen. Vergrößert man also im Zeitbereich die Breite T des Rechtecks, so wird das Amplitudenspektrum bei kleinen Frequenzen wegen der größeren Rechteckfläche ebenfalls größer. Danach fällt das Amplitudenspektrum mit zunehmender Frequenz steiler ab,

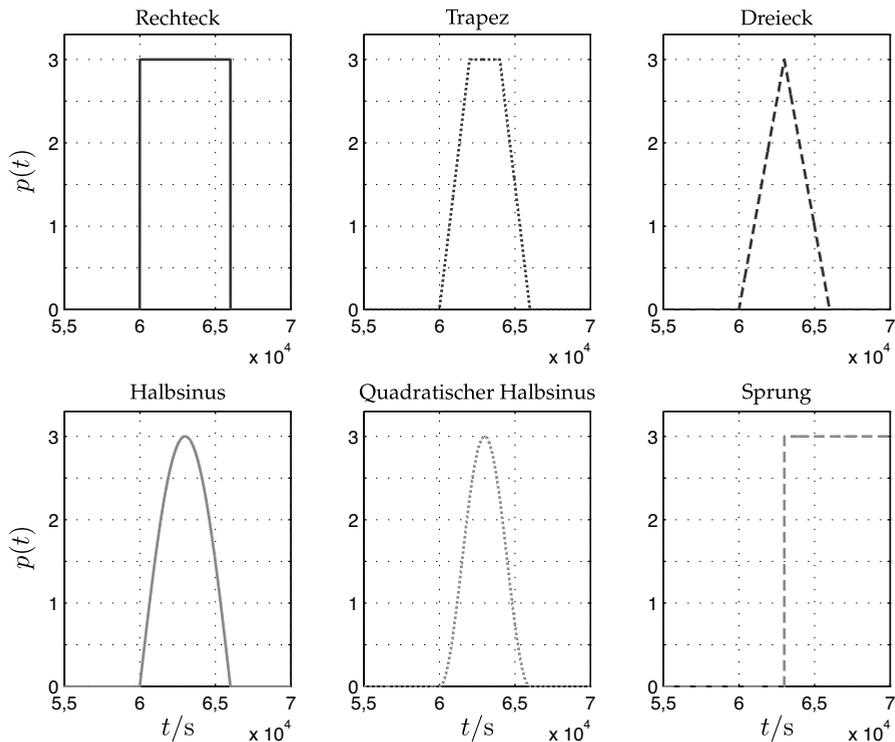


Abbildung 7.4. Auswahl transientser Anregungssignale.

da sich die Nullstellen zu kleineren Frequenzen hin verschieben [Ise88]. Für $T \rightarrow \infty$ konvergiert das Amplitudenspektrum gegen den Dirac-Impuls (vgl. Abb. 7.5).

Impulshammer

Üblicherweise werden Impulse bei der Modalanalyse durch einen Impulshammer erzeugt. Durch Zusatzgewichte und unterschiedliche Schlagoberflächen können die Impulshöhe und -breite beeinflusst werden. Weichere Kalotten bewirken breitere Impulse; durch ein Zusatzgewicht wird die Impulsamplitude vergrößert. Ein Problem bei der Impulsanregung ist die mangelhafte Reproduzierbarkeit des Anregungssignals. Einerseits fällt es schwer, immer dieselbe Stelle auf der untersuchten Struktur zu treffen. Andererseits erfolgt bei wiederholten Versuchen die Erregung durch unterschiedliche Stellen der Hammeroberfläche. Schließlich wird es nie gelingen, exakt die gleiche Krafrichtung zu reproduzieren. Aus diesem Grund müssen mehrere Messungen durchgeführt und die Ergebnisse gemittelt werden.

Ferner muss bei der experimentellen Analyse mit Impulshämmern darauf geachtet werden, dass keine durch Nachprellen bedingten Doppelschläge erfolgen, da diese das Anregungsspektrum signifikant verschlechtern. Werden zur Impulsanregung

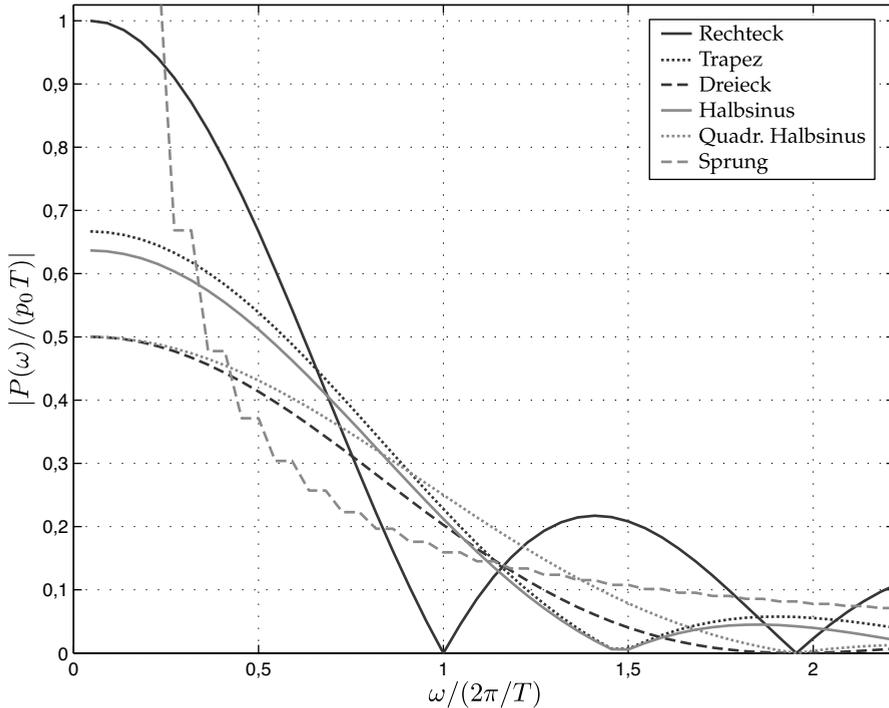


Abbildung 7.5. Normierte Amplitudenspektren transientser Testsignale.

Fallgewichte verwendet, so muss ebenfalls darauf geachtet werden, dass das vom Untersuchungsobjekt abfedernde Fallgewicht abgefangen wird, damit es nicht zu Doppelschlägen kommt.

Beurteilung der Impulsanregung

Impulsversuche sind schnell und mit geringem gerätetechnischen Aufwand möglich. Außerdem können sie eingeschränkt sogar während des Betriebs durchgeführt werden. Die Anregung niedriger Eigenfrequenzen ist jedoch verglichen mit der in Abschn. 7.3.3.5 besprochenen Sprunganregung nur beschränkt möglich. Die Probleme bei der Kraftsteuerung wurden im vorigen Abschnitt bereits diskutiert. Verglichen mit der harmonischen Anregung ist das SNR bei impulsförmiger Anregung wesentlich schlechter. Außerdem ist die Auswertung aufwändiger als bei harmonischer Erregung. Da jedoch die Vorteile dieser Anregungsart für die in diesem Versuch zu lösenden Aufgaben überwiegen, kommt ein Impulshammer zum Einsatz.

7.3.3.4 Fast frequency sweep

Wird ein Chirp-Signal mit einer hohen Gleitfrequenz gewählt, so kann man auch von einer transienten Anregung sprechen. Ein Beispiel hierzu ist:

$$p(t) = p_0 \sin\left(-\frac{2T}{\Omega_b - \Omega_a} \cdot \ln\left(\Omega_a - \frac{\Omega_b - \Omega_a}{2T} t\right)\right). \quad (7.52)$$

Verwendet werden kurze Signale $p(t)$ (z. B. der Dauer $T = 5$ s), die nicht periodisch wiederholt werden, was den Nachteil harmonischer Testsignale reduziert, dass sie viel Zeit in Anspruch nehmen.

7.3.3.5 Sprunganregung

Gemäß Abb. 7.5 sind Sprungfunktionen geeignet, um niedrige Frequenzen anzuregen. Solche Funktionen werden entweder auf sehr leichte oder sehr schwere Systeme angewendet. Erstere könnten durch einen Hammerschlag verformt oder zerstört werden, bei letzteren besteht die Gefahr, dass der Hammer zu leicht ist und damit die Anregungsamplitude nicht ausreicht.

Praktisch ist ein positiver Sprung jedoch schwer zu realisieren, denn die Struktur müsste schlagartig ausgelenkt werden und in der Endlage verharren. Dadurch würde aber die Auslenkeinrichtung mit der Struktur in Verbindung bleiben und diese dämpfen oder bei einer zusätzlichen eigenen Frequenz mitschwingen. Das Resultat wären verfälschte Moden.

Stattdessen lenkt man die Struktur zunächst mit einem Vorgewicht aus, z. B. mit einem Kabel oder Seil. Durch das plötzliche Entlasten durch Kappen des Seiles kommt es zu einem (negativen) Sprung, der sich gut reproduzieren lässt. Bedacht werden muss allerdings, dass das Seil an der Struktur befestigt werden muss und dass der dafür erforderliche Aufhängehaken das Schwingungsverhalten des Objekts verändern kann.

Nachteilig an der Sprunganregung ist, dass sie bei kompakten Systemen – z. B. einem Motorblock – praktisch nicht realisierbar ist.

7.3.3.6 Stochastische Anregung

Eine alternative Systemanregung ist die Verwendung von Rauschen. Die Autokorrelationsfunktion von weißem Rauschen $p(t)$ ist impulsförmig:

$$r_{pp}(\tau) = \sigma_p^2 \delta(\tau). \quad (7.53)$$

Durch Fourier-Transformation von (7.53) erhält man die Leistungsdichte des Rauschens, die konstant ist und nur von der Varianz von $p(t)$ abhängt:

$$S_{pp}(\omega) = \mathcal{F}\{r_{pp}(\tau)\} = \sigma_p^2. \quad (7.54)$$

Ideale weiße Rauschsignale sind nicht realisierbar, weshalb man sich in der Praxis mit farbigem (bandbegrenztem) Rauschen begnügen muss [PL15].

Das Autoleistungsdichtespektrum $S_{pp}(\omega)$ ist eine gerade Funktion. Deshalb müssen nur positive Frequenzen betrachtet werden; die negativen sind dadurch implizit berücksichtigt. Für einen Tiefpassprozess kann man unter diesen Annahmen bis zu einer oberen Eckfrequenz ω_b eine konstante Leistungsdichte annehmen und diese wie folgt darstellen:

$$S_{pp}(\omega) = \begin{cases} S_0 & \text{für } 0 \leq \omega \leq \omega_b \\ 0 & \text{für } \omega > \omega_b. \end{cases} \quad (7.55)$$

Durch inverse Fourier-Transformation erhält man:

$$r_{pp}(\tau) = S_0 \frac{\sin(\omega_b \tau)}{\pi \tau}. \quad (7.56)$$

Die zu (7.56) gehörende Zeitfunktion ist ebenfalls eine Sinc-Funktion:

$$p(t) = p_0 \frac{\sin(\omega_b t)}{\pi t}. \quad (7.57)$$

Für die stochastische Anregung ist ein ausreichend großes Zeitintervall notwendig. Beim Vergleich von (7.50) und (7.57) wird klar, dass die harmonische Anregung mit konstanter Kraftamplitude erfolgt. Demgegenüber besitzt die stochastische Anregung einen amplitudenmodulierten Anteil.

7.3.3.7 Vor- und Nachteile der Anregungssignale

Tabelle 7.1 zeigt einen Vergleich der unterschiedlichen Anregungssignale.

Tabelle 7.1. Eigenschaften der unterschiedlichen Anregungssignale.

	sinusförmig	transient	stochastisch
gute Steuerbarkeit	×		×
gutes SNR	×	(×)	×
günstige Erregung aller Freiheitsgrade im Frequenzintervall	×		×
Berücksichtigung von Nichtlinearitäten	×		×
kurze Versuchsdauer		×	(×)
kurzer Versuchsaufbau		×	
geringe Gerätekosten		(×)	

7.3.4 Praktische Durchführung der Modalanalyse

Für die praktische Modalanalyse sind folgende Punkte zu beachten:

- Es ist wichtig, die Objekte im Einbauzustand (befestigt, aufgehängt o. ä.) zu untersuchen, da sich sonst abweichende Modenformen ergeben.
- Nicht alle Punkte einer Struktur sind als Anregungs- oder Messpunkte geeignet. Durch Anregung an einem Punkt, an dem ein Mode eine Nullstelle besitzt (d. h. dieser Punkt bewegt sich bei diesem Mode nicht), kann dieser Mode nicht angeregt werden. Gleichzeitig kann durch Messung an diesem Punkt der Mode auch nicht detektiert werden. Daher ist es wichtig, die Anregungs- und Messpunkte sorgfältig auszuwählen.

7.3.5 Parameterschätzung

Während in Abschn. 7.2 das mathematische Modell einer schwingungsfähigen Struktur aufgestellt wurde und aus den als bekannt angenommenen Massen und Federkonstanten die Eigenfrequenzen und Modenformen in allgemeiner Form berechnet wurden, muss bei der praktischen Modalanalyse umgekehrt aus den gemessenen Signalen auf die unbekanntenen Massen und Federkonstanten geschlossen werden. Um aus den gemessenen Frequenzgängen die Modalparameter eines einzelnen Modes zu bestimmen, ist die Anwendung von Parameterschätzverfahren notwendig. Sogenannte SDOF-Verfahren (*single degree of freedom*) extrahieren aus einem einzelnen Frequenzgang die Modalparameter eines Modes, indem sie annehmen, dass in einem engen Bereich um eine Resonanzfrequenz das Systemverhalten ausschließlich von dem zu dieser Frequenz gehörenden Mode bestimmt wird. Es werden also in (7.42) alle Summanden vernachlässigt, die nicht zu der betrachteten Resonanzfrequenz gehören. Demgegenüber bestimmen MDOF-Verfahren (*multiple degrees of freedom*) aus einem Frequenzgang mehrere Moden gleichzeitig.

Ein Beispiel für ein solches MDOF-Verfahren ist das „Least-squares complex exponential“-Verfahren (LSCE), welches im Folgenden kurz erläutert wird [HF01]. Dabei wird lediglich eine einzige Übertragungsfunktion betrachtet, um die Komplexität nicht ausufern zu lassen. Im Unterschied zu (7.2.3) wird nicht der Quotient der Fourier-Transformierten, sondern der Laplace-Transformierten von Aus- und Eingangssignal gebildet. Dies stellt lediglich eine Verallgemeinerung dar, da die Laplace-Übertragungsfunktion für $s = j\omega$ in den Frequenzgang übergeht. Somit sind hier nun auch gedämpfte Systeme erlaubt. Die folgenden Betrachtungen sind relativ kompliziert und müssen nicht im Detail nachvollzogen werden. Es soll lediglich deutlich gemacht werden, wie vorgegangen werden kann, um die Eigenfrequenzen und Modenformen aus experimentell ermittelten Frequenzgängen zu bestimmen.

Beim LSCE-Verfahren wird allgemein eine Übertragungsfunktion zwischen den Punkten i und j eines MDOF-Systems betrachtet, die sich als Summe der einzelnen Moden in der Form

$$H_{ij}(s) = \sum_{r=1}^N \left(\frac{rA_{ij}}{s - s_r} + \frac{rA_{ij}^*}{s - s_r^*} \right) \quad (7.58)$$

schreiben lässt bzw. als

$$H_{ij}(s) = \sum_{r=1}^{2N} \frac{{}_r A_{ij}}{s - s_r}, \quad (7.59)$$

wobei je N Pole s_r und N Residuen ${}_r A_{ij}$ auch konjugiert komplex auftreten. Dieser Ansatz entspringt der Beobachtung aus (7.27), wonach sich Schwingungen ergeben; zwei Summanden mit konjugiert komplexer Polstelle und konjugiert komplexem Vorfaktor entsprechen einer Cosinusschwingung:

$$\begin{aligned} A e^{a_1 t} + A^* e^{a_1^* t} &= |A| e^{\operatorname{Re}(a_1)t} \left(e^{j(\operatorname{Arg}(A) + \operatorname{Im}(a_1)t)} + e^{-j(\operatorname{Arg}(A) + \operatorname{Im}(a_1)t)} \right) \\ &= 2|A| e^{\operatorname{Re}(a_1)t} \cos(\operatorname{Arg}(A) + \operatorname{Im}(a_1)t). \end{aligned} \quad (7.60)$$

Nach Laplace-Rücktransformation von (7.59) ergibt sich die Impulsantwort (engl. *impulse response function*, IRF):

$$h_{ij}(t) = \sum_{r=1}^{2N} {}_r A_{ij} e^{s_r t}. \quad (7.61)$$

Nach Abtasten der Impulsantwort mit der Abtastzeit Δt lässt sich diese als

$$h_{ij}(k\Delta t) = \sum_{r=1}^{2N} {}_r A_{ij} e^{s_r k\Delta t}, \quad k = 0, 1, \dots, 2N, \quad (7.62)$$

schreiben, wobei die $2N + 1$ Abtastwerte benachbart sein müssen. Mit $z_r^k := e^{s_r k\Delta t}$ folgt:

$$h_{ij,k} = \sum_{r=1}^{2N} {}_r A_{ij} z_r^k, \quad k = 0, 1, \dots, 2N, \quad (7.63)$$

d. h. der Abtastwert $h_{ij,k}$ der betrachteten Impulsantwort ist die Überlagerung der Abtastwerte der Moden zum selben Zeitpunkt. Da komplexe Pole und Residuen immer auch konjugiert komplex auftauchen, ist die abgetastete Impulsantwort reellwertig. Nun lässt sich ein Polynom als Produkt von Linearfaktoren bilden, dessen Nullstellen die z_r bilden, die noch zu berechnen sind:

$$\prod_{i=1}^{2N} (z - z_i) = (z - z_1) \cdot (z - z_2) \cdot \dots \cdot (z - z_{2N}) = 0. \quad (7.64)$$

Da die Pole immer komplex konjugiert auftreten, lässt sich dieses Produkt als ein Polynom mit reellen Koeffizienten umschreiben:

$$\beta_0 + \beta_1 z + \beta_2 z^2 + \dots + \beta_{2N-1} z^{2N-1} + \beta_{2N} z^{2N} = 0, \quad \beta_i \in \mathbb{R}. \quad (7.65)$$

Umgekehrt lassen sich bei bekannten Koeffizienten β_i , $i = 0, 1, \dots, 2N$, die z_r als Nullstellen des Polynoms berechnen.

Hinter (7.63) stecken $2N + 1$ Entsprechungen; daher ist es möglich, jedes h_k mit dem entsprechenden Koeffizienten β_k zu multiplizieren und über alle Produkte zu summieren:

$$\sum_{k=0}^{2N} \beta_k h_{ij,k} = \sum_{k=0}^{2N} \beta_k \sum_{r=1}^{2N} r A_{ij} z_r^k \quad (7.66)$$

bzw.

$$\sum_{k=0}^{2N} \beta_k h_{ij,k} = \sum_{r=1}^{2N} r A_{ij} \sum_{k=0}^{2N} \beta_k z_r^k. \quad (7.67)$$

Gleichung (7.65) wird zu null, falls z_r Nullstelle von ihr ist. Es resultiert eine ganz einfache Gleichung:

$$\sum_{k=0}^{2N} \beta_k h_{ij,k} = 0. \quad (7.68)$$

Im Umkehrschluss ist es möglich (und nötig, da die Nullstellen unbekannt sind), aus (7.68) die Koeffizienten β_i ($i = 0, 1, \dots, 2N$) zu berechnen (β_{2N} wird zu 1 gesetzt). Dazu werden $2N \cdot 2N$ Abtastpunkte der Impulsantwort verwendet:

$$\begin{bmatrix} h_{ij,0} & h_{ij,1} & h_{ij,2} & \cdots & h_{ij,2N-1} \\ h_{ij,1} & h_{ij,2} & h_{ij,3} & \cdots & h_{ij,2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{ij,2N-1} & h_{ij,2N} & h_{ij,2N+1} & \cdots & h_{ij,4N-2} \end{bmatrix} \cdot \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{2N-1} \end{bmatrix} = \begin{bmatrix} -h_{ij,2N} \\ -h_{ij,2N+1} \\ \vdots \\ -h_{ij,4N-1} \end{bmatrix}. \quad (7.69)$$

Werden mehr als $2N$ Zeilen verwendet, kann ein Least-Squares-Ansatz angewendet werden, beispielsweise kann die Pseudoinverse der linken Matrix mit h -Elementen gebildet werden. Mit den Koeffizienten β_i lassen sich die Nullstellen z_r des Polynoms (7.65) berechnen. Aus den Polen s_r der Übertragungsfunktion ergeben sich Eigenfrequenz ω_r und Dämpfung ξ_r des r -ten Modes:

$$s_r = -\xi_r \omega_r + j\sqrt{1 - \xi_r^2} \omega_r, \quad (7.70)$$

$$s_r^* = -\xi_r \omega_r - j\sqrt{1 - \xi_r^2} \omega_r \quad (7.71)$$

$$\Rightarrow \omega_r = \frac{1}{\Delta t} \sqrt{\ln(z_r) \ln(z_r^*)}, \quad (7.72)$$

$$\xi_r = -\frac{\ln(z_r z_r^*)}{2\omega_r \Delta t}. \quad (7.73)$$

Da die z_r nun bekannt sind, lässt sich (7.63) wie folgt umschreiben:

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ z_1^{2N-1} & z_2^{2N-1} & \cdots & z_{2N}^{2N-1} \end{bmatrix} \cdot \begin{bmatrix} 1A_{ij} \\ 2A_{ij} \\ \vdots \\ 2NA_{ij} \end{bmatrix} = \begin{bmatrix} h_{ij,0} \\ h_{ij,1} \\ \vdots \\ h_{ij,2N-1} \end{bmatrix}. \quad (7.74)$$

Aus dieser Gleichung lassen sich nun die Residuen rA_{ij} bestimmen.

Interpretation

Um nun nach der Bestimmung der Eigenfrequenzen, Dämpfungsfaktoren und Residuen mittels des LSCE-Verfahrens den Bezug der so bestimmten Parameter zur Theorie, im Wesentlichen zur Identifikation der Eigenvektoren, herzustellen, wird (7.22) unter Vernachlässigung der Anfangsbedingungen Laplace-transformiert:

$$s^2 \tilde{\mathbf{M}} \mathbf{Z}(s) + \tilde{\mathbf{K}} \mathbf{Z}(s) = \mathbf{U}^T \mathbf{F}(s), \quad (7.75)$$

$$(s^2 \tilde{\mathbf{M}} + \tilde{\mathbf{K}}) \mathbf{Z}(s) = \mathbf{U}^T \mathbf{F}(s), \quad (7.76)$$

$$\mathbf{Z}(s) = (s^2 \tilde{\mathbf{M}} + \tilde{\mathbf{K}})^{-1} \mathbf{U}^T \mathbf{F}(s) \quad (7.77)$$

$$\Rightarrow \mathbf{Y}(s) = \mathbf{U} (s^2 \tilde{\mathbf{M}} + \tilde{\mathbf{K}})^{-1} \mathbf{U}^T \mathbf{F}(s), \quad (7.78)$$

$$\mathbf{Y}(s) = \mathbf{U} \operatorname{diag} \left(\frac{1}{s^2 m_1 + k_1}, \dots, \frac{1}{s^2 m_N + k_N} \right) \mathbf{U}^T \mathbf{F}(s). \quad (7.79)$$

Mit der Normierung laut (7.19) ergibt sich m_i zu 1 und k_i zu $-\lambda_i^2$. Damit gilt für die Übertragungsmatrix $\mathbf{G}(s)$:

$$\mathbf{G}(s) = \mathbf{U} \operatorname{diag} \left(\frac{1}{s^2 - \lambda_1^2}, \dots, \frac{1}{s^2 - \lambda_N^2} \right) \mathbf{U}^T \quad (7.80)$$

$$= \begin{bmatrix} \sum_{i=1}^N u_{i1} u_{i1} \frac{1}{s^2 - \lambda_i^2} & \sum_{i=1}^N u_{i1} u_{i2} \frac{1}{s^2 - \lambda_i^2} & \cdots & \sum_{i=1}^N u_{i1} u_{iN} \frac{1}{s^2 - \lambda_i^2} \\ \sum_{i=1}^N u_{i2} u_{i1} \frac{1}{s^2 - \lambda_i^2} & \sum_{i=1}^N u_{i2} u_{i2} \frac{1}{s^2 - \lambda_i^2} & \cdots & \sum_{i=1}^N u_{i2} u_{iN} \frac{1}{s^2 - \lambda_i^2} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^N u_{iN} u_{i1} \frac{1}{s^2 - \lambda_i^2} & \sum_{i=1}^N u_{iN} u_{i2} \frac{1}{s^2 - \lambda_i^2} & \cdots & \sum_{i=1}^N u_{iN} u_{iN} \frac{1}{s^2 - \lambda_i^2} \end{bmatrix}. \quad (7.81)$$

Bei vernachlässigter Dämpfung $s_r = 0 + j\omega_r$ und für rein imaginäre ${}_r A_{ij} = -j {}_r B_{ij}$ ergibt sich (7.58) aus dem LSCE-Verfahren zu

$$H_{ij}(s) = \sum_{r=1}^N \left(\frac{{}_r A_{ij}}{s - s_r} + \frac{{}_r A_{ij}^*}{s - s_r^*} \right) = \sum_{r=1}^N \left(\frac{-j {}_r B_{ij}}{s - s_r} + \frac{j {}_r B_{ij}}{s - s_r^*} \right) \quad (7.82)$$

$$= \sum_{r=1}^N \frac{2 {}_r B_{ij} \omega_r}{s^2 + \omega_r^2}. \quad (7.83)$$

Schreibt man sämtliche $H_{ij}(s)$ in Matrixform, erhält man mit (7.83) die Übertragungsmatrix $\mathbf{H}(s)$ des LSCE-Verfahrens zu

$$\mathbf{H}(s) = \begin{bmatrix} \sum_{r=1}^N \frac{2 {}_r B_{11} \omega_r}{s^2 + \omega_r^2} & \sum_{r=1}^N \frac{2 {}_r B_{12} \omega_r}{s^2 + \omega_r^2} & \cdots & \sum_{r=1}^N \frac{2 {}_r B_{1N} \omega_r}{s^2 + \omega_r^2} \\ \sum_{r=1}^N \frac{2 {}_r B_{21} \omega_r}{s^2 + \omega_r^2} & \sum_{r=1}^N \frac{2 {}_r B_{22} \omega_r}{s^2 + \omega_r^2} & \cdots & \sum_{r=1}^N \frac{2 {}_r B_{2N} \omega_r}{s^2 + \omega_r^2} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{r=1}^N \frac{2 {}_r B_{N1} \omega_r}{s^2 + \omega_r^2} & \sum_{r=1}^N \frac{2 {}_r B_{N2} \omega_r}{s^2 + \omega_r^2} & \cdots & \sum_{r=1}^N \frac{2 {}_r B_{NN} \omega_r}{s^2 + \omega_r^2} \end{bmatrix}. \quad (7.84)$$

Durch Koeffizientenvergleich zwischen (7.81) und (7.84) lassen sich die λ_i und die Eigenvektorelemente u_{ij} ermitteln. Zum Beispiel folgt aus dem Vergleich der Matrixelemente $[N, N]$:

$$\sum_{i=1}^N u_{iN} u_{iN} \frac{1}{s^2 - \lambda_i^2} \stackrel{!}{=} \sum_{i=1}^N \frac{2 {}_i B_{NN} \omega_i}{s^2 + \omega_i^2} \quad (7.85)$$

$$\Rightarrow \quad j \lambda_i = \omega_i, \quad u_{Ni}^2 = 2 {}_i B_{NN} \omega_i. \quad (7.86)$$

Mit den durch Least-Squares-Schätzung für ω_i und ${}_r A_{ij}$ bzw. ${}_r B_{ij}$ ermittelten Werten lassen sich die Eigenfrequenzen und Eigenvektorelemente, sprich Modenformen der untersuchten Struktur, siehe Abschn. 7.2.3, bestimmen.

Aufgrund der Reziprozität [KSA06] ist es egal, ob man am Punkt i anregt und am Punkt j misst oder umgekehrt. Durch diese Eigenschaft ist die Übertragungsmatrix $\mathbf{G}(s)$ (vgl. (7.81)) symmetrisch. Regt man also einen Punkt an und misst an allen anderen Punkten, oder misst man an einem Punkt und regt an allen anderen Punkten an, erhält man eine Zeile bzw. Spalte der Übertragungsmatrix $\mathbf{G}(s)$. Aus dieser Zeile bzw. Spalte kann man die Schwingungsformen und Eigenfrequenzen aller betrachteten Moden an allen untersuchten Strukturpunkten bestimmen.

Mit der Parameterschätzung ist der Kreis geschlossen und der Bezug des physikalischen Modells zu den gemessenen Frequenzgängen eines realen Systems hergestellt. Aus den Frequenzgängen werden die Modalparameter extrahiert und somit experimentell die Eigenfrequenzen und die Modenformen bestimmt.



Abbildung 7.6. Analogeingangsmodul NI9234.



Abbildung 7.7. Kistler-Impulshammer mit integriertem Kraftsensor.

7.4 Versuchsdurchführung

Die Messdatenaufnahme erfolgt mit *LabVIEW*. Die Auswertung der aufgenommenen Messdaten wird teilweise durch Datenimport in *Matlab* durchgeführt. Im Folgenden erfolgt zunächst eine Beschreibung der Versuchsumgebung sowie anschließend eine kurze Erläuterung der einzelnen Schritte.

Zur Messdatenaufnahme wird das Analogeingangsmodul NI9234 von *National Instruments* verwendet (Abb. 7.6), welches über USB mit dem PC verbunden wird. Die vier Kanäle des Moduls können zeitsynchron ausgelesen werden. Jeder der vier Kanäle kann mit einem Erregerstrom (*Integrated Electronics Piezo-Electric, IEPE*) betrieben werden. Dieser Erregerstrom ist für die Verwendung von piezoelektrischen Sensoren, wie Kraft-, Beschleunigungs- oder Drucksensoren, erforderlich.

Die Versuchsdurchführung erfolgt anhand eines Impulshammers (vgl. Abb. 7.7) und eines piezoelektrischen Beschleunigungsaufnehmers. Für die Hammerspitze gibt es als Aufsätze Kalotten unterschiedlicher Härte. Es hängt von der untersuchten Struktur ab, welche Kalotte verwendet wird. Dabei regen härtere Kalotten höhere Frequenzen an als weichere.

Der Beschleunigungsaufnehmer (Abb. 7.8) ist ein einachsiger Piezosensor, der auf die untersuchte Struktur aufgeklebt wird. Wie auch der Kraftsensor des Impulshammers liefert der Beschleunigungssensor eine Spannung. Beim Ankleben des Beschleunigungssensors darf das Kabel nicht geknickt werden. Bitte halten Sie den Sensor am Korpus und nicht am Kabel!

Im Folgenden werden die verwendete Hardware und die Messdatenaufnahme in *LabVIEW* genauer beschrieben.



Abbildung 7.8. Einachsiger Beschleunigungsaufnehmer von PCB.

7.4.1 Hardware

Das Analogeingangsmodule *NI9234* kann vier 16-Bit-Kanäle zeitsynchron mit bis zu 51,2 kS/s abtasten, was einer maximalen Abtastfrequenz von $f_{A,max} = 51,2 \text{ kHz}$ entspricht. Die einzelnen Kanäle inklusive Abtastfrequenz lassen sich bequem in *LabVIEW* konfigurieren.

Der verwendete Impulshammer *Kistler 9722A500* ist mit unterschiedlich harten Kalotten ausgestattet. Die Kalibrierungsdaten für den integrierten Kraftsensor entnehmen Sie dem Auszug aus dem Datenblatt in Tab. 7.2.

Tabelle 7.2. Auszug aus dem Datenblatt des Kistler-Impulshammers.

Configuration	Tip Extender	9906(BLACK) NONE	9906(BLACK) 9922	9912(GREY) NONE	9912(GREY) 9922
Sensitivity	mV/N	11.04	11.73	8.88	9.66
at 100 Hz	mV/lbf	49.13	52.20	39.50	42.99
Sensitivity	N/g●	0.93	0.88	1.16	1.07
Ratio at 100 Hz	lbf/g●	0.21	0.20	0.26	0.24

Der einachsige Beschleunigungsaufnehmer *PCB M353B16* hat die Empfindlichkeit 10,88 mV/g. Beim Aufkleben des Sensors muss darauf geachtet werden, dass dieser mit der ganzen Fläche fest aufliegt. Das Anschlusskabel ist sehr empfindlich und sollte nicht geknickt werden. Zum Ankleben des Sensors halten Sie diesen daher am Korpus und drücken ihn fest an.

7.4.2 Messdatenaufnahme in LabVIEW

Zur Messdatenaufnahme wird der DAQ-Assistent verwendet. Dieser findet sich im Blockdiagramm unter Express → Eingabe → DAQ-Assistent (Abb. 7.9). Wählt man

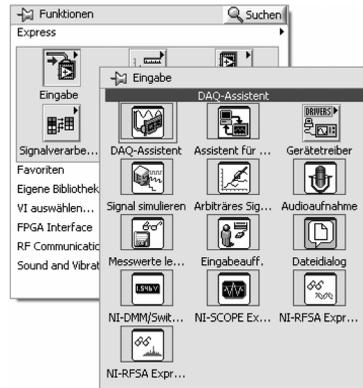


Abbildung 7.9. DAQ-Assistent.

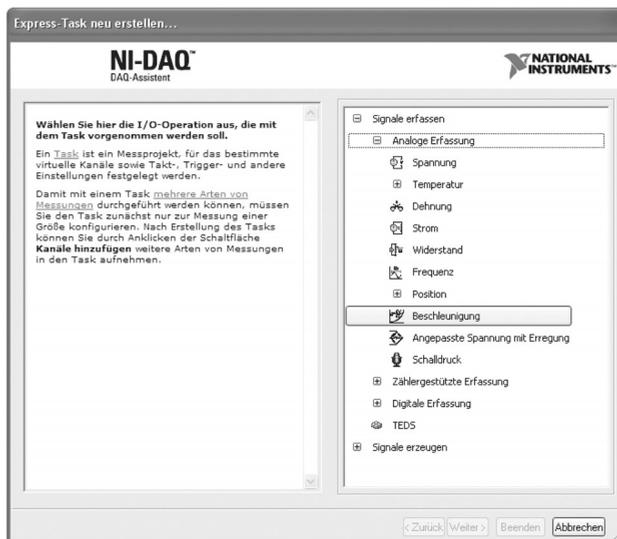


Abbildung 7.10. DAQ-Assistent: Wahl der Signalart.

den DAQ-Block aus und zieht diesen ins Blockdiagramm, so öffnet sich automatisch der DAQ-Assistent (Abb. 7.10).

Zunächst wird die Art des Signals ausgewählt, das ausgegeben oder gemessen werden soll. Im darauf folgenden Dialog wird nach dem Kanal gefragt, der konfiguriert werden soll (Abb. 7.11). Mit Beenden des Dialogs wird der Assistent zur Konfiguration des ausgewählten Kanals aufgerufen (Abb. 7.12). Darin kann eingestellt werden, ob der ausgewählte Kanal kontinuierlich ausgelesen werden soll, oder nur N Messwerte aufgezeichnet werden sollen. Zudem kann die Abtastrate f_A in der Einheit „kilosamples per second“ kS/s eingestellt werden. Über die Empfindlichkeit wird die Sensorkalibrierung durchgeführt. Nach Beenden des DAQ-Assistenten erscheint

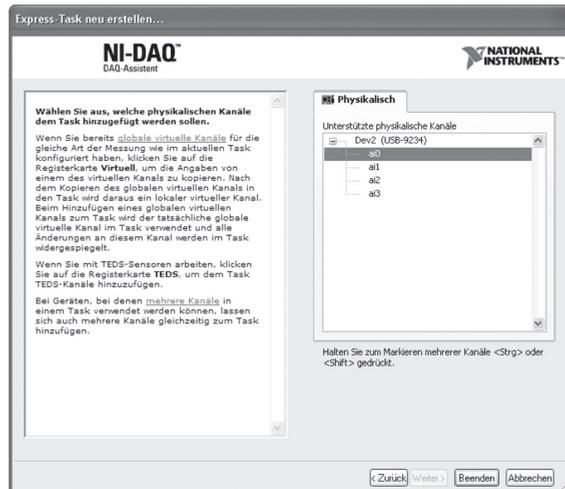


Abbildung 7.11. DAQ-Assistent: Wahl des Kanals.

der Block im Blockdiagramm und ist einsatzfähig. Anhand des DAQ-Blocks ist es nun möglich, die aufgenommenen Messdaten direkt weiter zu verarbeiten. Dazu wird der Anschluss Daten direkt mit weiteren Funktionsblöcken verbunden, z. B. mit einem Signalverlaufgraphen. Es besteht auch die Möglichkeit, die aufgenommenen Daten abzuspeichern, um beispielsweise eine Weiterverarbeitung in Matlab durchzuführen. Hierzu verwenden Sie den Block In Tabellenkalkulationsdatei schreiben. Sie finden diesen Block unter Programmierung → Datei I/O → In Tabellenkalkulationsdatei schreiben. Abbildung 7.13 zeigt die zu verwendende externe Beschaltung des Blocks. Nutzen Sie zur Beschaltung die Möglichkeit, mit Klick der rechten Maustaste auf die entsprechenden Blockeingänge automatisch geeignete Konstanten zu erstellen (Rechte Maustaste → Erstellen → Konstante). Durch diese Beschaltung werden die eingehenden Messdaten in dem in Dateipfad angegebenen Ort abgespeichert. Zudem werden in der abgespeicherten Datei alle von *LabVIEW* standardmäßig gesetzten Kommas durch Punkte ersetzt.

7.4.3 Eigentliche Modalanalyse

Für die Analyse der Metallplatte und des Motorgehäuses sind viele Frequenzgänge (FRF) aufzunehmen und zu analysieren; zu diesem Zweck sind zwei vorbereitete VIs auf dem Versuchsrechner abgelegt.

7.4.3.1 Einlesen der Frequenzgänge

Zum Einlesen mehrerer Frequenzgänge steht das VI FRFgenerierung zur Verfügung (Abb. 7.14). Zunächst muss die Gittergröße des Abtastrasters definiert werden; hierbei ist zu beachten, dass jeweils der erste Punkt einer Achse den Index 0 trägt. Zum

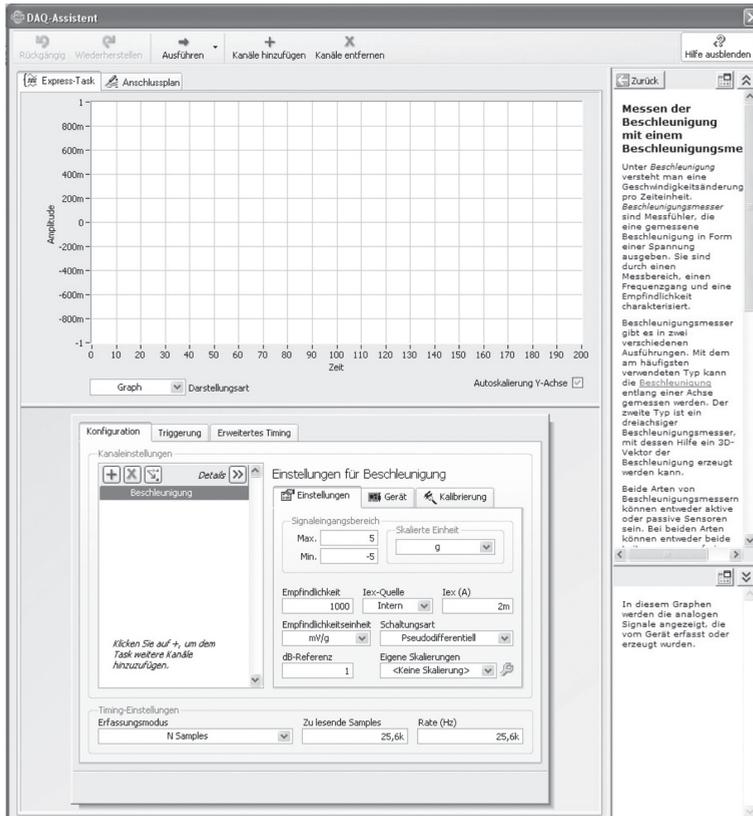


Abbildung 7.12. DAQ-Assistent: Konfiguration des Kanals.

Einlesen eines 2×2 -Gitters muss also für Gittergröße x und Gittergröße y jeweils eine 1 eingetragen werden. Die Abtastrate und die Anzahl der zu lesenden Samples ist bereits festgelegt. Mit Klick auf OK werden die nötigen Schaltflächen zum Einlesen und Anzeigen der Frequenzgänge aktiviert. Nach Klick auf Einzelne FRF einlesen werden für 2 Sekunden Daten aufgezeichnet. In dieser Zeit muss der Schlag mit dem Impulshammer erfolgen. Im rechten Teil des VIs können die aufgezeichneten Frequenzgänge nochmals angezeigt und gegebenenfalls gelöscht werden. Es ist auch möglich, mehrere Frequenzgänge für eine Gitterposition einzulesen. In diesem Fall erfolgt eine Mittelung über alle für eine Position eingelesenen Frequenzgänge.

7.4.3.2 Modenbestimmung und -animation

Zur Berechnung und Animation der Moden dient das VI Modalanalyse (Abb. 7.15). Mit Klick auf Load FRF data werden die gemessenen Frequenzgänge eingelesen. Danach können mit Identify mit dem unter Settings spezifizierten Parameterschätzverfahren die Modalparameter bestimmt werden. Dabei kann der jeweilige Frequenz-

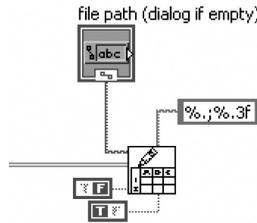


Abbildung 7.13. Speichern von Messdaten.

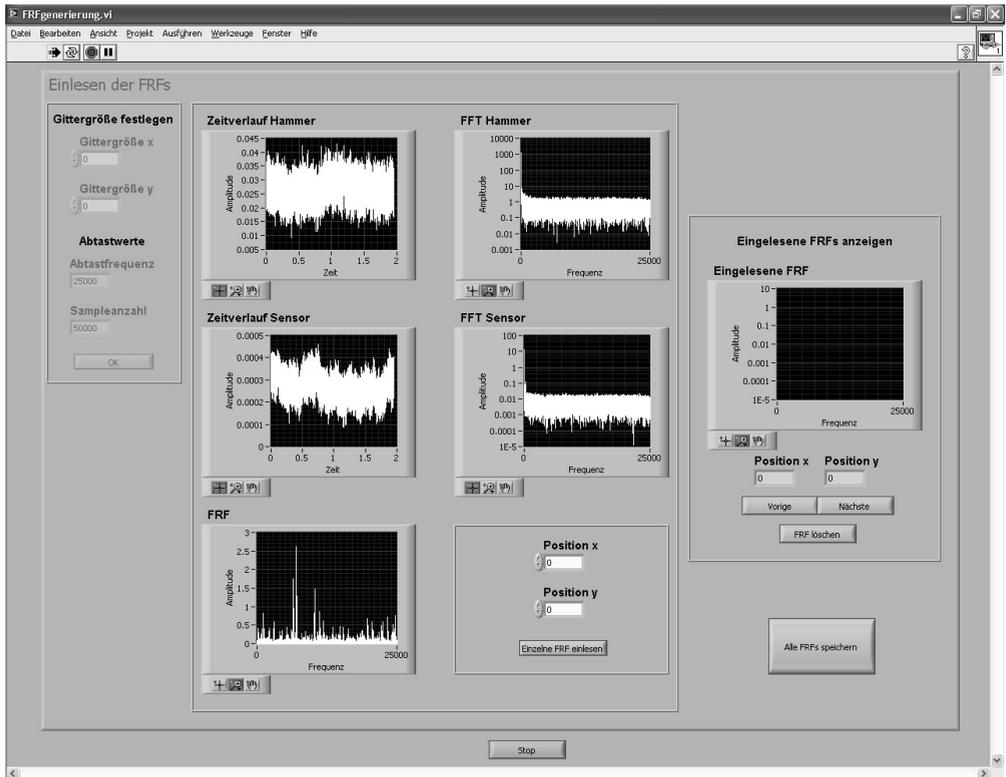


Abbildung 7.14. VI zum Einlesen von Frequenzgängen.

gang im Graph-Fenster angesehen werden und mit Hilfe der beiden gelben Balken der zu untersuchende Frequenzbereich bestimmt werden. Um die gelben Balken verschieben zu können, muss in der Graph-Palette das Kreuz (linker Button) markiert sein. Durch Auswählen im Feld Measure # werden die einzelnen Frequenzgänge angezeigt. Beachten Sie, dass die Berechnung der Modalparameter alle geladenen Frequenzgänge verwendet und nicht nur die momentan angezeigte. Nach der Identifikation können die einzelnen Resonanzfrequenzen (Identified Modes) ausgewählt werden, die im nächsten Schritt animiert werden sollen (Selected Modes). Nach Aus-



Abbildung 7.15. Modalanalyse-VI.

wahl der Resonanzfrequenzen wird mit Klick auf Next das Animationsfenster eingeblendet. In diesem kann nun jeweils ein einzelner Mode ausgewählt und animiert werden.

7.4.4 Nützliche Matlab-Befehle

In diesem Abschnitt werden einige nützliche *Matlab*-Befehle für die Durchführung des Versuchs erläutert. Sie erhalten eine ausführliche Hilfe über den Befehl `help`. Alternativ können Sie auch das Helpdesk durch den Befehl `helpdesk` starten.

- `fft(x)` berechnet die DFT des Vektors x . Der zugehörige Frequenzvektor lautet:

$$f = \frac{f_{\text{Abtast}} * \left(-\frac{n}{2} : \frac{n}{2}\right)}{n}, \quad (7.87)$$

wobei n die Länge von x und $\left(-\frac{n}{2} : \frac{n}{2}\right)$ ein Vektor in *Matlab*-Schreibweise sind.

- `ifft(x)` berechnet die inverse DFT von x .
- `fftshift(x)` ordnet das DFT-Spektrum x symmetrisch zum Nullpunkt an.

- `filter(B, A, x)` filtert den Vektor x mit einem digitalen Filter mit den Koeffizienten B (Zähler) und A (Nenner).
- `real` gibt den Realteil einer Zahl oder eines Vektors an.
- `imag` gibt den Imaginärteil einer Zahl oder eines Vektors an.
- `angle` gibt den Phasenwinkel einer Zahl oder eines Vektors an.
- `abs` gibt den Betrag einer Zahl oder eines Vektors an.
- `x .* y` multipliziert zwei Vektoren x und y gleicher Dimension elementweise.
- `length(x)` bestimmt die Länge des Vektors x .
- Plotoptionen:
 - `semilogx`, `semilogy`, `loglog` erstellen Plots mit logarithmischen Achsen.
 - `subplot(n, m, z)` erstellt mehrere Achsenkreuze in einem Plot. n gibt die Anzahl der Achsen in einer Zeile, m die Anzahl in den Spalten. z ist ein Zähler, der von oben links nach unten rechts durchläuft.
 - `grid on/off` blendet Hilfslinien in den Plot ein/aus.
 - `zoom on/off` erlaubt es, in den Plot zu zoomen.
 - `hold on/off` erlaubt es, mehrere Datenreihen in einen Plot einzutragen.
 - `title(x)` fügt den String x als Titel in den Plot ein.
 - `legend(x, y, ...)` fügt die Strings x und y als Legende entsprechender Datenreihen in den Plot ein.

7.5 Versuchsvorbereitende Aufgaben

Aufgabe 7.1

- a) Sie sollen eine flexible Metallplatte zum Schwingen anregen. Für welches transiente Anregungssignal würden Sie sich entscheiden? Begründung!
- b) Bei einer Modalanalyse mittels eines Impulshammers können vorhandene Oberschwingungen nicht untersucht werden. Welche Änderungen am Impulshammer können Sie vornehmen, um diese Oberschwingungen identifizieren zu können?
- c) Was zeigen die Eigenvektoren *anschaulich* an?
- d) Überlegen Sie sich einen Versuchsaufbau, bei dem die beschriebenen systematischen Fehler (unterschiedliche Krafrichtung und Anregungsamplitude...) durch Anregung mit einem Impulshammer vermieden/verringert werden können. Erstellen Sie dazu eine Skizze und beschreiben Sie den Versuchsaufbau und die daraus resultierenden Verbesserungen!
- e) Gegeben sei ein Anregungsmechanismus, der einen Rechteckimpuls auf das zu untersuchende Objekt übertragen soll. In einem Versuch wurde die in Abb. 7.16

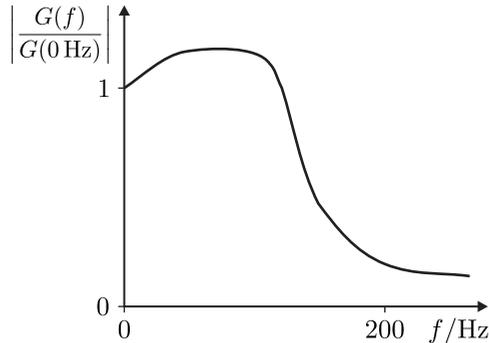


Abbildung 7.16. Betragsspektrum der übertragenen Kraft.

gezeigte Fourier-Transformierte der übertragenen Kraft ermittelt. Ist der Mechanismus korrekt justiert? Falls nicht, grenzen Sie den Fehler ein.

f) Gegeben sind die Matrizen

$$\mathbf{M} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} 4 & 4 \\ 4 & 20 \end{bmatrix}. \quad (7.88)$$

Bestimmen Sie die allgemeinen Eigenvektoren (mit variablem Vorfaktor) und die damit diagonalisierten Matrizen $\tilde{\mathbf{M}}$ und $\tilde{\mathbf{K}}$. Normieren Sie dann die Eigenvektoren nach (7.19) und berechnen Sie die mit diesen diagonalisierten Matrizen.

7.6 Versuchsaufgaben

7.6.1 Messanweisungen

Erstellen Sie unter $D:\backslash\text{PDSV}\backslash\text{Gruppen}$ ein Verzeichnis für Ihre Gruppe. Legen Sie alle Dateien, die Sie im Laufe des Versuchs erstellen, nur in diesem Ordner ab! Im Ordner $D:\backslash\text{PDSV}\backslash\text{Versuch}_7$ finden Sie alle vorgefertigten VIs, die Sie für die einzelnen Aufgaben benötigen. Wenn Sie an diesen VIs Veränderungen vornehmen, dann überschreiben Sie die VIs in diesem Verzeichnis bitte nicht.

Achten Sie darauf, beim Schlagen die Messpunkte möglichst senkrecht zu treffen, um genaue Messergebnisse zu erhalten. Verhindern Sie Doppelschläge. **Begrenzen Sie die Schlagamplituden auf max. 300 g, um plastische Verformungen am Metallstreifen zu vermeiden und um den teuren Impulshammer nicht zu zerstören.** Sie können Ihre Schlagamplitude direkt in *LabVIEW* mit dem DAQ-Assistenten betrachten.

7.6.2 Messungen und Versuchsdurchführung

7.6.2.1 Messdatenerfassung

Aufgabe 7.2: Hammersignal in LabVIEW einlesen

In dieser Aufgabe implementieren Sie die Aufzeichnung der Impulshammersignale in *LabVIEW*. Verwenden Sie zur Durchführung der Versuche den Blechstab. Schlagen Sie den auf einem Stück Holz aufliegenden Blechstab an.

- a) Zunächst soll der Impulshammer über den DAQ-Assistenten in *LabVIEW* eingebunden werden. Gehen Sie dazu wie in Abschn. 7.4 beschrieben vor. Verwenden Sie den DAQ-Assistenten zur Konfiguration des Eingangskanals und kalibrieren Sie den Impulshammer: Verwenden Sie dazu eine Abtastrate von $f_A = 25$ kHz und definieren Sie die Aufnahmedauer zu $T = 2$ s. Kalibrieren Sie den Impulshammer entsprechend den Angaben aus Tab. 7.2 für den Fall 9906 (BLACK) ohne Extender in der Einheit mV/g durch Eingabe des ermittelten Werts in das Fenster „Empfindlichkeit“. Alle anderen Angaben, wie der Erregerstrom i_{ex} oder die Schaltungsart, sind unverändert zu übernehmen. Visualisieren Sie das Zeitsignal anhand eines Signalverlaufsgraphen. Achten Sie darauf, dass der Hammerschlag während der Aufnahmedauer vollständig erfasst wird! In diesem Aufgabenteil werden folgende Blöcke benötigt:
- Express → Eingabe → DAQ-Assistent (im Blockdiagramm)
 - Express → Graph-Anzeigeelemente → Signalverlaufsgraph (im Frontpanel)
- b) Lassen Sie in einem zweiten Signalverlaufsgraphen den Betrag der FFT des Hammerschlags im Frequenzbereich 0–1500 Hz darstellen. Hierzu müssen Sie folgende Blöcke hinzufügen:
- Signalverarbeitung → Transformationen → FFT (im Blockdiagramm)
 - Express → Arithmetik und Vergleich → Num (Express) → Absoluter Wert (im Blockdiagramm)
 - Express → Graph-Anzeigeelemente → Signalverlaufsgraph (im Frontpanel)
- Achten Sie auf die Achsenbeschriftung und die Aufteilung der Frequenzachse!
- c) Speichern Sie das Ergebnis mit dem Block Programmierung → Datei-I/O → In Tabellenkalkulationsdatei schreiben (im Blockdiagramm) in eine .txt-Datei (vgl. Abschn. 7.4). Beschalten Sie den Block entsprechend Abb. 7.13.

Aufgabe 7.3: Hammeroberflächen

Diese Aufgabe dient zur Verifikation der Antwort aus Aufgabe 7.1 b). Dazu sollen Sie das Spektrum für verschiedene Hammeroberflächen analysieren.

- a) Beim Impulshammer wurden drei zusätzliche Kalotten aus unterschiedlichem Material und unterschiedlicher Härte mitgeliefert (rot, grün, grau, weiß). Führen Sie

für jede dieser Kalotten eine Messung mit dem in der vorherigen Aufgabe erstellten VI durch. Achten Sie darauf, dass die Anregungssignale immer die gleiche Gestalt haben – insbesondere, dass keine Mehrfachschläge auftreten. Schlagen Sie den Blechstreifen am eingespannten Rand an, wo keine signifikanten Nachschwingungen auftreten. Die Stärke der Anregung sollte so gewählt werden, dass die Impulse gut als solche erkennbar sind. Speichern Sie die Messungen in jeweils einer .txt-Datei ab und benennen Sie diese entsprechend.

- b) Sind die Messungen beendet, so schreiben Sie unter Zuhilfenahme der angegebenen Befehle ein Matlab-Programm zur Berechnung der Betragsspektren der einzelnen Anregungsmessungen. Importieren Sie dazu die einzelnen Messungen in *Matlab*. Bestimmen Sie die FFT und plotten Sie die einzelnen Betragsspektren mit einer Legende für die einzelnen Hammeroberflächen. Erstellen Sie zusätzlich einen Plot mit den Anregungssignalen im Zeitbereich, um zu überprüfen, dass die Schlagamplitude jeweils gleich groß ist.
- c) Wozu ist es nötig, die Schlagamplituden jeweils gleich groß zu halten? Bestätigen die Messergebnisse Ihre theoretischen Überlegungen? Beschreiben Sie Ihre Ergebnisse! Wie korrespondieren der Spektralverlauf und das Zeitsignal der Anregung?
- d) Ordnen Sie basierend auf Ihren Messergebnissen die Kalotten von weich nach hart.

Aufgabe 7.4: Beschleunigungssignal in LabVIEW einlesen

Nachdem nun das Hammersignal in *LabVIEW* eingelesen wird, soll nun auch das Signal des Beschleunigungssensors erfasst werden. Verwenden Sie zur Durchführung der Versuche die kleine weiße Kalotte.

- a) Fügen Sie im DAQ-Assistenten einen weiteren Kanal für den Beschleunigungssensor hinzu. Die Abtastung soll synchron zu der des Hammersignals erfolgen. Geben Sie daher für N und f_A entsprechende Werte ein. Kalibrieren Sie den Sensor entsprechend (siehe Abschnitt 7.4.1). Schließen Sie den DAQ-Assistenten und fügen Sie folgenden Block Ihrem Projekt hinzu:
 - Express → Signalverarbeitung → Signale trennen

Wie auch beim Impulshammer wird ein Signalverlaufgraph für das Zeitsignal und den Betrag der FFT benötigt. Erweitern Sie entsprechend Ihr *LabVIEW*-VI.

- b) Bilden Sie aus den gemessenen Signalen von Impulshammer und Beschleunigungssensor den Frequenzgang und lassen Sie diesen ebenfalls in *LabVIEW* plotten.

7.6.2.2 Eigentliche Modalanalyse

Für alle folgenden Aufgaben ist die weiße Kalotte zu verwenden.

Aufgabe 7.5: Analyse einer Metallplatte

In dieser Aufgabe sollen nun die Modenformen einer frei schwingenden Metallplatte untersucht werden. Lesen Sie dazu mit Hilfe des VIs FRFgenerierung die Frequenzgänge eines 3×7 -Punktrasters ein (Initialisierung der Gittergröße im VI beachten, siehe Theorieteil! Die jeweiligen Koordinaten der Gitterpunkte sind auf der Metallplatte angezeichnet und können so übernommen werden). Hierbei ist es sinnvoll, den Beschleunigungssensor an der Stelle (0,0) zu fixieren und direkt neben dieser sowie an allen anderen Punkten mit dem Impulshammer anzuregen. Schlagen Sie jeden Punkt dreimal an, um die Messgenauigkeit zu verbessern. Speichern Sie die Frequenzgänge. Falls es nötig ist, die Messungen zu wiederholen, stoppen und starten Sie das VI nicht erneut, sondern schließen und öffnen Sie es nochmals.

- a) Öffnen Sie das VI Modalanalyse und bestimmen Sie die Resonanzfrequenzen im Bereich von 0 Hz bis 600 Hz mit dem LSCE-Verfahren. Durch die Approximation wird in dem mit dem gelben Balken markierten Bereich eine bestimmte Anzahl von Resonanzfrequenzen (Fenster „mode num“) approximiert. Markieren Sie einen interessanten Bereich und überlegen Sie sich, wie viele Eigenfrequenzen in diesem zu bestimmen sind. Hierbei ist es wichtig, dass jeder Peak genau angenähert wird. Falls das Schätzverfahren Peaks liefert, die in den aufgezeichneten Daten nicht vorhanden sind, wird mit diesen auch keine real vorhandene Resonanzfrequenz dargestellt. Falls das Verfahren nicht alle Peaks gleichzeitig approximiert, bestimmen Sie im Zweifelsfall jeweils eine einzelne Resonanzfrequenz in einem engeren Frequenzbereich und fügen Sie die Frequenzen einzeln zu den ausgewählten Frequenzen (Fenster „Selected Modes“) hinzu.
- b) Visualisieren Sie die Modenformen mit dem VI. Fertigen Sie Screenshots sämtlicher Modenformen für das Protokoll an.

Aufgabe 7.6: Analyse eines Motorgehäuses

Für ein Motorgehäuse sollen die Resonanzfrequenzen und Modenformen bestimmt werden, um später die bei bestimmten Frequenzen besonders lästige Schallabstrahlung minimieren zu können. Lesen Sie für alle gekennzeichneten Punkte die Frequenzgänge dreimal ein! Beachten Sie hierbei, dass die Struktur stark zurückfedert und dabei den Impulshammer nicht berühren darf. Es empfiehlt sich also, eher mit kleinerer Amplitude anzuschlagen und den Hammer rechtzeitig zurückzuziehen.

- a) Bestimmen Sie mit dem VI Modalanalyse_Gehaeuse die Eigenfrequenzen und Modenformen bis ca. 250 Hz.
- b) Fahren Sie den Motor hoch und messen Sie mit dem Drehzahlmessgerät die Drehzahl der Welle. Das Drehzahlmessgerät ist im Abstand von ca. 2 cm waagrecht vor dem schwarz umrandeten Loch zu halten. Bei welchen Drehzahlen/Frequenzen

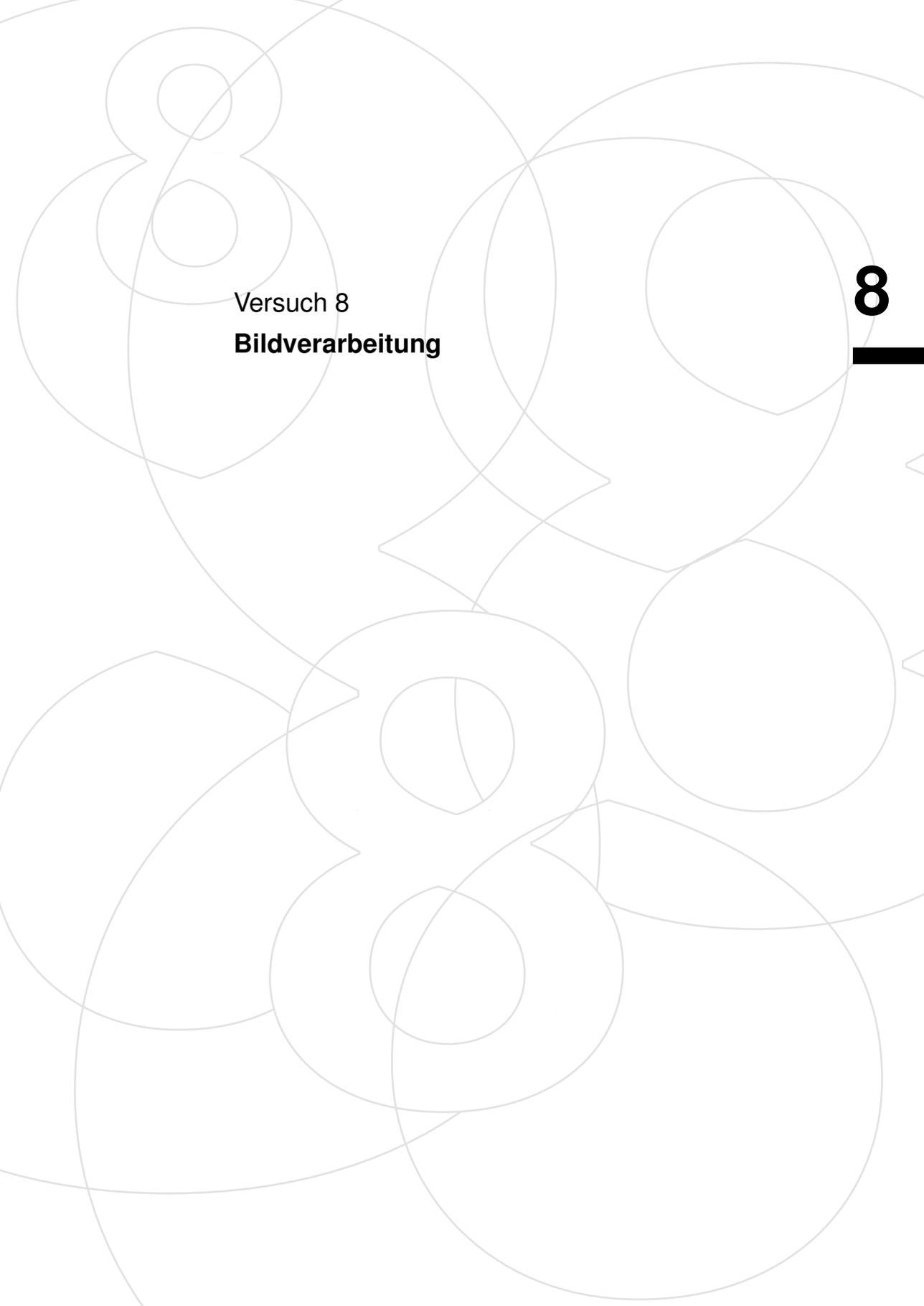
treten besonders stark hörbare Resonanzen auf? Stimmen diese mit den mit Hilfe der Modalanalyse bestimmten Drehzahlen überein?

- c) Welche Änderungen können an der Struktur des Gehäuses vorgenommen werden, um die Schallabstrahlung zu minimieren?

7.7

7.7 Literatur

- [Bre88] HARTMUT BREMER: *Dynamik und Regelung mechanischer Systeme*. Teubner, Stuttgart, 1988.
- [GK87] ROBERT GASCH und KLAUS KNOTHE: *Strukturodynamik, Band 1: Diskrete Systeme*. Springer, Berlin Heidelberg, 1987.
- [HF01] JIMIN HE und ZHI-FANG FU: *Modal Analysis*. Butterworth-Heinemann, Oxford, 2001.
- [Ise88] ROLF ISERMANN: *Identifikation dynamischer Systeme*. Springer, Berlin Heidelberg, 1988.
- [KSA06] FRANZ GUSTAV KOLLMANN, THOMAS FRANZ SCHÖSSER und ROLAND ANGERT: *Praktische Maschinenakustik*. Springer, Berlin Heidelberg, 2006.
- [Nat92] HANS G. NATKE: *Einführung in Theorie und Praxis der Zeitreihen- und Modalanalyse – Identifikation schwingungsfähiger elastomechanischer Systeme*. Vieweg, Braunschweig Wiesbaden, 3. Auflage, 1992.
- [PJ15] FERNANDO PUENTE LEÓN und HOLGER JÄKEL: *Signale und Systeme*. De Gruyter Oldenbourg, Berlin, 6. Auflage, 2015.
- [PL15] FERNANDO PUENTE LEÓN: *Messtechnik – Systemtheorie für Ingenieure und Informatiker*. Springer, Berlin Heidelberg, 10. Auflage, 2015.

The background of the page is a complex, abstract composition of thin, light gray lines. These lines form a series of overlapping circles and arcs of various sizes, creating a sense of depth and movement. Some circles are partially obscured by others, while some lines cross each other at different angles. The overall effect is a clean, modern, and somewhat organic geometric pattern.

Versuch 8
Bildverarbeitung

8 Bildverarbeitung

8.1	Einleitung	265
8.2	Grundlagen der Bildverarbeitung	265
8.2.1	Einführung	266
8.2.2	Bildgewinnung	266
8.2.2.1	CCD- und CMOS-Sensoren	266
8.2.2.2	Farbkameras	267
8.2.2.3	Helligkeitsdynamik	268
8.2.2.4	Digitalisierung	269
8.2.2.5	Übertragung zum Rechner	270
8.2.3	Bilddarstellung in Ortsbereich und Ortsfrequenzbereich	270
8.2.3.1	Zweidimensionale DFT	270
8.2.3.2	Eigenschaften der DFT	271
8.2.3.3	Zweidimensionale Faltung	272
8.2.4	Bildvorverarbeitung und Merkmalsextraktion	273
8.2.4.1	Allgemeine Eigenschaften von Operatoren	273
8.2.4.2	Punktoperatoren	274
8.2.4.3	Lokale Operatoren	276
8.2.4.4	Glättungsfiler	278
8.2.4.5	Kantendetektion	282
8.2.4.6	Einfache Kantendetektoren	286
8.2.5	Segmentierung und Klassifikation	287
8.2.5.1	Binärsegmentierung	287
8.2.5.2	Kantenorientierte Segmentierung	287
8.2.5.3	Template-Matching	288
8.3	Realisierung in LabVIEW	289
8.3.1	Kurzbeschreibung NI-VISA für RS232	289
8.3.2	Bildverarbeitung mit IMAQ Vision for LabVIEW	291
8.3.2.1	Arbeiten mit Bildern	291
8.3.2.2	Zugriff auf die FireWire-Kamera	292
8.4	Versuchsaufbau	293
8.5	Versuchsvorbereitende Aufgaben	293
8.6	Aufgaben am Versuchstag	295
8.7	Literatur	300

8 Bildverarbeitung

8.1 Einleitung

8.1

Digitale Bildverarbeitung ist in mehrfacher Hinsicht ein faszinierendes Gebiet. Während Menschen den größten Teil ihrer Information über die äußere Wirklichkeit mit Hilfe ihres natürlichen Sehvermögens aufnehmen, geht die Technik über das reine fotografische Festhalten von Bildern hinaus. Sichtbares maschinell wahrzunehmen und quantitativ auszuwerten stellt den Beginn einer weiteren tiefgreifenden Umwälzung im Umgang mit Bildern dar. Der enorme Fortschritt der Computertechnologie erlaubt einen breiten Einsatz von Bildverarbeitungsmethoden zur Auswertung komplexer zwei- und dreidimensionaler Vorgänge in allen Bereichen der Natur- und Ingenieurwissenschaften und darüber hinaus [BPF16, Jä12].

Zu Beginn der Bildverarbeitungskette steht die Bildaufnahme. Hierbei ist ebenso wie bei eindimensionalen Signalen auf die Einhaltung des Abtast- und Quantisierungstheorems zu achten. Andernfalls ergeben sich Moiré-Effekte oder Bilder mit schlechter Auflösung. Ebenso ist die zweidimensionale Filterung essentiell für die Bildvorverarbeitung bzw. die Merkmalsextraktion aus einem Bild. Teile der Bildverarbeitung können also als Erweiterung der eindimensionalen auf eine höherdimensionale Signalverarbeitung verstanden werden. Darüber hinaus sind die Bildsegmentierung, Bildanalyse und Bildklassifikation weitere Schritte der Bildverarbeitung.

Im Versuch werden die Grundlagen der Bildverarbeitung behandelt und Schwerpunktthemen – wie die Kantendetektion und Filteroperationen – näher beleuchtet. Danach wird auf die Realisierung von Bildverarbeitungsoperationen mit *LabVIEW* eingegangen und der Versuchsaufbau erklärt.

Die Versuchsaufgaben adressieren u. a. die Ansteuerung der Hardware des Aufbaus und die Bildaufnahme. Des Weiteren wird auf die Filterung von Bildern und die Kantendetektion eingegangen. Ferner werden die Korrelation in der Bildverarbeitung und das Template-Matching-Verfahren untersucht. Als Abschluss ist eine Demonstration aus dem Bereich der automatischen Sichtprüfung von Platinen vorgesehen.

8.2 Grundlagen der Bildverarbeitung

8.2

Dieser Abschnitt vermittelt Grundlagen der digitalen Bildverarbeitung. Der Schwerpunkt liegt dabei auf der Bildfilterung. Zuerst werden die Bildgewinnung und -repräsentation vorgestellt. Anschließend wird auf einfache Operationen zur Bildvorverarbeitung und Merkmalsextraktion eingegangen. Im letzten Abschnitt sind Verfahren zur Bildsegmentierung aufgeführt.

8.2.1 Einführung

Die Bildverarbeitung lässt sich in drei Stufen gliedern (vgl. Abb. 8.1).

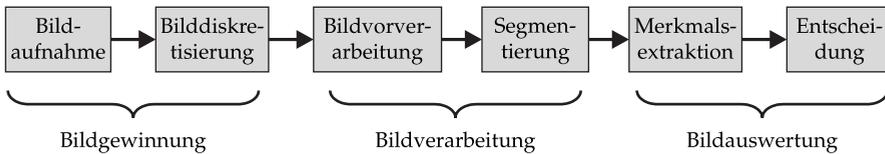


Abbildung 8.1. Stufen der Bildverarbeitung.

- Die erste Stufe stellt die **Bildgewinnung** dar. Meist werden dafür Kamerasysteme verwendet.
- Die eigentliche **Bildverarbeitung** lässt sich in zwei Schritte unterteilen: die Bildvorverarbeitung und die Segmentierung. Die Bildvorverarbeitung zielt darauf ab, störende oder irrelevante Signalanteile zu unterdrücken, beispielsweise durch eine Bildglättung. Die Segmentierung hat zur Aufgabe, das Bild in bedeutungstragende Regionen – z. B. Objekte und Hintergrund – aufzuteilen, um die nachfolgende Merkmalsextraktion zu erleichtern.
- In der **Bildauswertung** werden aufgabenrelevante Merkmale extrahiert, um darauf basierend eine Entscheidung (Detektion, Klassifikation oder Interpretation des Bildinhaltes) zu treffen.

8.2.2 Bildgewinnung

Der erste Schritt in der Bildverarbeitung ist die Bildgewinnung, die meist mit Hilfe einer Kamera oder eines Scanners erfolgt. Im Folgenden wird kurz auf die Funktionsweise der häufig eingesetzten CCD- und CMOS-Sensoren eingegangen sowie die weiteren notwendigen Verarbeitungsschritte bis zum digitalen Bild dargestellt.

8.2.2.1 CCD- und CMOS-Sensoren

Zur digitalen Erfassung von Bildern werden meist Halbleitersensoren eingesetzt. Die verbreitetsten Typen sind **CCD-** und **CMOS-Sensoren**. CCD steht für *charge-coupled device*, CMOS für *complementary metal oxide semiconductor*. Beiden gemeinsam ist der grundsätzliche Aufbau: Die aufzunehmende Szene wird durch eine Optik auf eine Matrix aus Photoelementen abgebildet, die als **Pixel** (Abkürzung für *picture element*) bezeichnet werden. In diesen bilden sich proportional zur Intensität des einfallenden Lichtes und zur Belichtungszeit unterschiedlich große elektrische Ladungen aus, die anschließend durch eine Auswerteelektronik ausgelesen werden.

Bei CCD-Sensoren werden die Ladungen während des Auslesens entlang der einzelnen Spalten nacheinander von Zelle zu Zelle verschoben [BPF16]. In jedem Schritt wird dabei eine komplette Bildzeile aus dem Sensor herausgeschoben und in einem horizontalen Schieberegister zwischengespeichert. Hieraus können schließlich

die einzelnen Pixelwerte nacheinander zur seriellen Übertragung ausgelesen werden. Diese horizontalen und vertikalen Speicherzellen sind als analoge Speicher ausgeführt, die Digitalisierung erfolgt pixelweise nach dem Herausschieben aus dem horizontalen Schieberegister mittels A/D-Umsetzer.

CMOS-Sensoren spielen erst seit wenigen Jahren eine große Rolle in der Bildfassung, da sie lange Zeit den CCD-Sensoren in der Bildqualität deutlich unterlegen waren. Da die heute gebräuchlichen digitalen ICs wie Mikroprozessoren, DSPs usw. auch in CMOS-Technik aufgebaut sind, können die Bildsensoren und die (Vor-)Verarbeitungselektronik auf einem einzigen Chip integriert werden, was eine kostengünstige Fertigung ermöglicht. Die eigentlichen lichtempfindlichen Zellen sind ähnlich aufgebaut wie bei den CCD-Sensoren. Allerdings müssen CMOS-Sensoren nicht zwangsläufig sequentiell ausgelesen werden, da die Photoelemente einer Spalte nicht direkt miteinander verbunden sind. Stattdessen kann jede Photozelle einzeln über Spaltenbusse adressiert werden, worauf sie einen der Beleuchtungsintensität proportionalen Spannungswert über den (analogen) Bus an den A/D-Umsetzer liefert. Pro Spalte ist ein eigener A/D-Umsetzer vorhanden, wodurch das Auslesen des kompletten Bildes deutlich schneller erfolgen kann als bei CCD-Sensoren. Darüber hinaus ist es möglich, nur einen bestimmten Bildausschnitt wiederholt auszulesen und somit höhere Bildraten – wenn auch bei reduzierter Auflösung – zu erreichen.

Um trotz des sequentiellen Auslesens eine über alle Bildpunkte konstante Belichtungszeit zu erreichen, können die lichtempfindlichen Zellen gemeinsam entladen werden. Dies legt den Beginn der Belichtung fest. Nach Ablauf der Belichtungszeit werden die Ladungen jedes Pixels wiederum gleichzeitig in je einen lichtunempfindlichen Bereich verschoben, woraus sie dann nacheinander ausgelesen werden.

8.2.2.2 Farbkameras

Übliche Farbkameras liefern für jedes Pixel die drei Farbkanäle Rot, Grün und Blau. Die Bildfassung erfolgt meist mit Hilfe von Farbfiltern, die vor die Bildsensoren in den Strahlengang montiert werden. In der Praxis sind zwei verschiedene Methoden verbreitet: Ein-Chip-Systeme und Drei-Chip-Systeme [BPF16]. Während bei letzteren der durch das Objektiv abgebildete Strahlengang durch Prismenspiegel aufgesplittet wird und durch drei großflächige Farbfilter auf je einen Sensorchip gelangt, kommt bei den Ein-Chip-Systemen nur ein einziger Bildsensor zum Einsatz. Hierbei werden je vier nebeneinanderliegende Sensorpixel zu einem Makropixel zusammengefasst und mit je einem kleinen Farbfilter einer Grundfarbe versehen. Dadurch reduziert sich zwar die Auflösung des Bildes auf ein Drittel gegenüber der Aufnahme eines Grauwertbildes mit dem gleichen Sensor, allerdings kann das gesamte optische System wesentlich kompakter und günstiger aufgebaut werden.

Im Folgenden wird stets von monochromen Bildern ausgegangen, deren Pixel ausschließlich unterschiedliche Grauwerte annehmen.

8.2.2.3 Helligkeitsdynamik

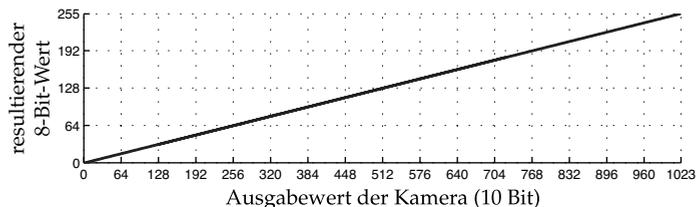
Aufgrund der physikalischen Eigenschaften der Photozellen gibt es sowohl eine untere Grenze der Beleuchtungsstärke, ab welcher überhaupt eine Ladung aufgebaut wird, als auch eine obere Grenze (Sättigungsgrenze), ab welcher auch bei weiterer Bestrahlung keine zusätzliche Ladung entsteht. Das Verhältnis beider Grenzen wird als Helligkeitsdynamik bezeichnet. Da die Dynamik real verfügbarer Sensoren im Vergleich zu den in der Natur vorkommenden Helligkeitsunterschieden gering ist, kommt es in der Praxis häufig zu Unter- und Überbelichtung. Um diese Effekte auszugleichen, gibt es unterschiedliche Ansätze:

- Beeinflussen der Belichtungszeit;
- Einsatz einer Irisblende oder von Graufiltern bei zu hoher Beleuchtungsintensität;
- künstliche Beleuchtung der Szene bei zu geringer Umgebungshelligkeit.

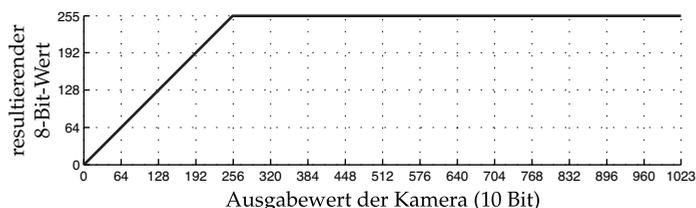
In vielen Fällen wird eine Kombination dieser Maßnahmen eingesetzt, um ein optimales Ergebnis zu erreichen. Darüber hinaus erfolgt bei der A/D-Umsetzung eine Quantisierung der einzelnen Pixelwerte.

Um Speicherplatz zu sparen und den Verarbeitungsaufwand zu verringern, wird häufig die vom A/D-Umsetzer gelieferte Helligkeitsauflösung reduziert: Die im Versuch eingesetzte CMOS-Kamera *Basler A601f* beispielsweise digitalisiert die Bilder mit 10 Bit pro Pixel; für die Weiterverarbeitung kann die Auflösung vor der Übertragung zum Rechner auf 8 Bit pro Pixel reduziert werden. Die Umrechnung erfolgt durch eine lineare Funktion, die den Eingangswertebereich $[0; 1023]$ auf den reduzierten Wertebereich $[0; 255]$ abbildet. Diese Funktion besitzt zwei vom Benutzer einstellbare Parameter: Brightness (Helligkeit) und Gain (Verstärkung) [Bas04].

- **Gain (Wertebereich: 0–255):** Durch diesen Parameter wird die Steigung der Umrechnungskennlinie festgelegt. Bei einem Gain von null wird der komplette Eingangswertebereich auf den Ausgang abgebildet, die Quantisierungsauflösung sinkt dabei auf ein Viertel (Abb. 8.2(a)). Der maximale Gain von 255 hat eine Verstärkung um 12 dB zur Folge, d. h. für Eingangswerte bis 255 entspricht der 8-Bit-Wert dem 10-Bit-Wert, für alle größeren Eingangswerte wird 255 ausgegeben (Abb. 8.2(b)). Durch Vergrößern des Gain kann also die Quantisierungsauflösung verbessert werden, allerdings nimmt die nutzbare Helligkeitsdynamik um bis zu 75 % ab. Dies ist meist nur dann sinnvoll, wenn die Kamera nicht voll angesteuert ist. Der Zusammenhang zwischen dem Gain-Wert und der resultierenden Verstärkung ist in Tab. 8.1 für einige typische Werte dargestellt.
- **Brightness (Wertebereich: 0–1023):** Dieser Parameter stellt den Achsenabschnitt der Umrechnungskennlinie dar. Beim Standardwert von ca. 725 (kann je nach Kamera leicht variieren) geht die Gerade durch den Ursprung; wird der Wert erhöht, wird die Gerade parallel nach oben verschoben, bei geringeren Werten wandert sie nach unten. Eine Veränderung dieses Parameters ist nur dann sinnvoll, wenn



(a) Gain = 0 (0 dB)



(b) Gain = 255 (12 dB)

Abbildung 8.2. Gain-Einstellungen der Kamera Basler A601f.

Tabelle 8.1. Zusammenhang zwischen Gain-Wert und Verstärkung.

Gain-Wert	dB	Verstärkung
0	0,0	1,0
28	2,5	1,3
43	3,5	1,5
85	6,0	2,0
128	8,0	2,5
170	9,5	3,0
213	10,9	3,5
255	12,0	4,0

Gain auf Werte größer null eingestellt ist, da bei einem Gain von null bereits der komplette Wertebereich des Sensors auf die 8 Bit 0–255 abgebildet wird.

8.2.2.4 Digitalisierung

Die Bilddigitalisierung umfasst sowohl eine örtliche Abtastung als auch eine Quantisierung der einzelnen Messwerte. Die Abtastung erfolgt durch den Bildsensor selbst: Jede Photozelle mittelt die auf ihrer aktiven Oberfläche auftreffende Strahlungsintensität und wirkt damit ähnlich wie das bei Zeitsignalen eingesetzte Abtast-Halte-Glied. Der A/D-Umsetzer führt anschließend die Quantisierung durch. Üblicherweise werden hierfür zwischen 8 Bit und 10 Bit verwendet. In Spezialfällen kann jedoch auch der Einsatz einer höheren Quantisierungsaufösung sinnvoll sein.

8.2.2.5 Übertragung zum Rechner

Für die Übertragung der Bilddaten zum Rechner stehen unterschiedliche Bussysteme zur Verfügung:

- IEEE 1394 (FireWire),
- USB 2.0/3.0,
- CameraLink,
- analoge Übertragung mit anschließendem **Frame-Grabber** (A/D-Umsetzer für analoge Bildsignale).

Dabei ist zu beachten, dass keine verlustbehaftete Kompression wie JPEG, MPEG o. ä. zum Einsatz kommen sollte, da die hierbei entstehenden Artefakte die Weiterverarbeitung erheblich stören können.

8.2.3 Bilddarstellung in Ortsbereich und Ortsfrequenzbereich

Neben der Darstellung als Matrix können digitale Bilder auch durch eine Funktion mit zwei (diskreten) Parametern m und n repräsentiert werden. Um der Indizierung bei der Matrixnotation zu entsprechen, wird der linke obere Bildpunkt (das Matrixelement $g_{1,1}$) durch die Koordinaten $m = n = 1$ angegeben. Die Koordinate m entspricht dabei dem Zeilenindex der Matrix und wird entgegen der kartesischen y -Richtung gezählt (also von oben nach unten aufsteigend), die Zählrichtung für n stimmt mit der kartesischen x -Koordinate überein (siehe Abb. 8.3). Hierbei spricht man von einer Bilddarstellung im **Ortsbereich**. Mittels einer zweidimensionalen diskreten Fourier-Transformation (2D-DFT) lässt sich das Bild in den **Ortsfrequenzbereich** überführen.

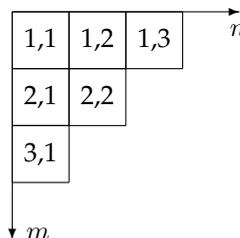


Abbildung 8.3. Koordinatenachsen in Ortsbereich.

8.2.3.1 Zweidimensionale DFT

Es sei die eindimensionale Fourier-Transformation eines Ortssignals $g(x)$ betrachtet:

$$G(f) = \int_{-\infty}^{\infty} g(x) e^{-j2\pi f x} dx. \quad (8.1)$$

Beim Übergang von den kontinuierlichen, unendlich ausgedehnten Signalen $g(x)$ auf die diskreten und endlichen Signale $g_n = g(n \Delta x)$ der Länge N resultiert aus (8.1) die

diskrete Fourier-Transformation (DFT), vgl. Abschn. 3.3.3 und 5.3.1 [Bri95]:

$$G_k = G(k \Delta f) = \sum_{n=0}^{N-1} g_n e^{-j2\pi \frac{kn}{N}}. \quad (8.2)$$

Hierbei bezeichnen k den **Frequenzindex** oder **Wellenzahl** und Δf die **Frequenzauflösung**. Aufgrund der Abtastung im Ortsbereich ist das DFT-Spektrum G_k periodisch; es reicht daher aus, eine einzige Periode zu betrachten ($0 \leq k < N$). Auch die Rücktransformierte

$$g_n = g(n \Delta x) = \frac{1}{N} \sum_{k=0}^{N-1} G_k e^{j2\pi \frac{kn}{N}} \quad (8.3)$$

ist wegen der Diskretisierung im Frequenzbereich periodisch, d. h. man kann sich auf den Bereich $0 \leq n < N$ beschränken.

Aufgrund des faktorisierbaren Kerns der DFT kann die Erweiterung auf zweidimensionale Signale g_{mn} der Größe $M \times N$ dadurch erfolgen, dass die **zweidimensionale DFT** (2D-DFT) durch zwei eindimensionale DFTs realisiert wird [BPF16]:

$$G_{kl} = \sum_{m=0}^{M-1} \left(\sum_{n=0}^{N-1} g_{mn} e^{-j2\pi \frac{nl}{N}} \right) e^{-j2\pi \frac{mk}{M}} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} g_{mn} e^{-j2\pi \left(\frac{mk}{M} + \frac{nl}{N} \right)}. \quad (8.4)$$

Für die inverse 2D-DFT erhält man:

$$g_{mn} = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} G_{kl} e^{j2\pi \left(\frac{mk}{M} + \frac{nl}{N} \right)}. \quad (8.5)$$

Zur Berechnung benutzt man den FFT-Algorithmus (*fast Fourier transform*). Im eindimensionalen Fall wird dadurch der Rechenaufwand für die Transformation eines Vektors der Länge $N = 2^k$ ($k \in \mathbb{N}$) von $\mathcal{O}(N^2)$ auf $\mathcal{O}(N \lg N)$ verringert, bei zweidimensionalen Signalen von $\mathcal{O}(M^2 N^2)$ auf $\mathcal{O}(MN (\lg M + \lg N))$ [Bri95].

8.2.3.2 Eigenschaften der DFT

Im Folgenden wird auf Eigenschaften der DFT eingegangen, die für die Bildverarbeitung besonders wichtig sind.

Periodizität

Wegen der Abtastung im Ortsbereich ist das DFT-Spektrum G_k periodisch. Darüber hinaus bewirkt die Frequenzdiskretisierung eine periodische Fortsetzung des Ortssignals g_n : Nach der Rücktransformation in den Ortsbereich gilt $g_n = g_{n+iN}$ mit $i \in \mathbb{Z}$, die Elemente g_0 und g_{N-1} sind also Nachbarn. Geometrisch lässt sich diese Periodizität veranschaulichen, indem man diese Randelemente zu einem Kreis verbindet. Bei Bildern erfolgt die periodische Fortsetzung in beide Richtungen. Durch das Verbinden der vertikalen und der horizontalen Bildränder miteinander resultiert ein Torus.

Entsprechend beschreibt das DFT-Spektrum eines eindimensionalen Signals das Signal ausschließlich im sogenannten Nyquist-Band $[-f_A/2, f_A/2]$, wobei f_A die Abtastfrequenz bezeichnet. Die höchste Ortsfrequenz, die in einer Bildspalte oder -zeile dargestellt werden kann, ergibt sich, wenn zwei benachbarte Pixel jeweils einem Minimum und dem benachbarten Maximum der zugehörigen Schwingung entsprechen. Abbildung 8.4 stellt eine solche Bildzeile der Länge $N = 10$ dar. Die größte darstellbare Ortsfrequenz hat den Frequenzindex $k = N/2 = 5$.

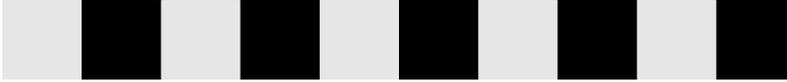


Abbildung 8.4. Höchste darstellbare Ortsfrequenz in einem Bild.

Symmetrie bei reellen Signalen

Bei reellen Signalen $g_n = g_n^*$ erhält man durch komplexe Konjugation von (8.2):

$$G_k^* = \left(\sum_{n=0}^{N-1} g_n e^{-j2\pi \frac{kn}{N}} \right)^* = \sum_{n=0}^{N-1} g_n^* e^{j2\pi \frac{kn}{N}} = \sum_{n=0}^{N-1} g_n e^{-j2\pi \frac{(-k)n}{N}} = G_{-k}, \quad (8.6)$$

woraus die Symmetrie des Betrags und die Schiefsymmetrie der Phase resultieren. Das DFT-Spektrum eines reellen Signals weist also eine Hermite'sche Symmetrie auf.

Im Beispiel aus Abb. 8.4 liefert die DFT ein Spektrum der Länge $N = 10$. Aufgrund der Symmetrieeigenschaften des DFT-Spektrums reichen zu dessen vollständiger Beschreibung bei geradem N die Werte G_k mit $k \in \{0, \dots, N/2\}$ aus. Bei zweidimensionalen reellen Signalen benötigt man entsprechend lediglich die Werte G_{kl} mit $k \in \{0, \dots, M/2\}$ und $l \in \{0, \dots, N/2\}$.

8.2.3.3 Zweidimensionale Faltung

Die eindimensionale **diskrete Faltung** $g_n * h_n$ ist wie folgt definiert:

$$k_n = g_n * h_n = \sum_{i=-\infty}^{\infty} g_i h_{n-i} = \sum_{i=-\infty}^{\infty} h_i g_{n-i}. \quad (8.7)$$

Analog dazu wird die diskrete Faltung zweidimensionaler Signale definiert:

$$k_{mn} = g_{mn} ** h_{mn} = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} g_{ij} h_{m-i, n-j} = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h_{ij} g_{m-i, n-j}. \quad (8.8)$$

Die Faltung zweier eindimensionaler kontinuierlicher Signale korrespondiert mit der Multiplikation ihrer Fourier-Transformierten im Frequenzbereich [PJ15]:

$$g(x) * h(x) \quad \circ \bullet \quad G(f) H(f). \quad (8.9)$$

Wenn endlich ausgedehnte diskrete Signale g_n und h_n und ihre DFT-Spektren G_k bzw. H_k betrachtet werden, muss beachtet werden, dass die DFT sowohl die Signale g_n und h_n als auch das Faltungsergebnis k_n periodisch fortsetzt; eine Multiplikation im Frequenzbereich korrespondiert daher mit einer **zyklischen Faltung** [BPF16]:

$$\begin{array}{l} \text{zyklisch} \\ \downarrow \\ g_n * h_n = \sum_{i=0}^{N-1} g_i h_{(n-i) \bmod N} \quad \circ \bullet \quad G_k H_k. \end{array} \quad (8.10)$$

Der Index $n-i$ in der Faltungssumme (8.10) ist modulo N zu verstehen. Um sicherzustellen, dass die Berechnung der Faltung im Frequenzbereich zum gleichen Ergebnis wie (8.7) führt, müssen die Signale vor der Berechnung der DFT durch Hinzufügen von mindestens $N-1$ Nullen (engl. *zero padding*) verlängert werden. Der Fehler infolge der Nichtbeachtung dieser Regel wird **Randeffekt** genannt.

Für zweidimensionale, endlich ausgedehnte Signale g_{mn} und h_{mn} und deren DFT-Spektren G_{kl} bzw. H_{kl} ergibt sich analog der Zusammenhang:

$$\begin{array}{l} \text{zyklisch} \\ \downarrow \\ g_{mn} ** h_{mn} = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} g_{ij} h_{(m-i) \bmod M, (n-j) \bmod N} \quad \circ \bullet \quad G_{kl} H_{kl}. \end{array} \quad (8.11)$$

Um den Randeffekt zu vermeiden, muss das Bild durch Hinzufügen von Nullen in beiden Dimensionen vergrößert werden [BPF16].

Der Zusammenhang zwischen Faltung im Ortsbereich und Multiplikation im Ortsfrequenzbereich ist besonders für die linearen Nachbarschaftsoperatoren (Abschn. 8.2.4.3) und die Kreuzkorrelation (Abschn. 8.2.5.3) interessant, da sich Faltungen über den FFT-Algorithmus effizient berechnen lassen. Schon bei Faltungskernen der Größe 5×5 ist die Berechnung im Ortsfrequenzbereich schneller als im Ortsbereich.

8.2.4 Bildvorverarbeitung und Merkmalsextraktion

Die hier behandelten Bildverarbeitungsoperatoren lassen sich in zwei Kategorien einteilen: **Punktoperatoren** und **lokale Operatoren**. Beiden gemeinsam ist, dass dadurch ein digitales Bild in ein Ergebnisbild meist gleicher Größe transformiert wird. Durch gezielten Einsatz von Operatoren können beispielsweise unerwünschte Signalanteile unterdrückt oder gewünschte Informationen in Bildern verstärkt werden.

8.2.4.1 Allgemeine Eigenschaften von Operatoren

Im Folgenden werden einige Eigenschaften vorgestellt, die im Zusammenhang mit Bildverarbeitungsoperatoren relevant sind. In der Praxis wird man zunächst festlegen, welche dieser Eigenschaften für die jeweilige Anwendung erforderlich sind und dann einen Operator auswählen, der diese aufweist.

Verschiebungsfreiheit

Ein **verschiebungsfreier** Operator verändert die Position von Objekten im Bild nicht. Dies entspricht einem konstanten Phasengang von 0° (*nullphasiges Filter*) [Jä12].

Richtungsunabhängigkeit (Isotropie)

Glättungsoperatoren sollten Bildinhalte richtungsunabhängig unterdrücken. Um dies zu gewährleisten, sollte die Filtermaske **isotrop** sein (vgl. Abschn. 8.2.4.3).

Verschiebungsinvarianz

Bei **verschiebungsinvarianten** Filtern wird auf alle Pixel des Originalbildes unabhängig von ihrer Position derselbe Operator angewendet. Bei Zeitsignalen heißt die zugehörige Eigenschaft **Zeitinvarianz**.

Linearität

Lineare Operatoren \mathcal{S} erfüllen das **Superpositionsprinzip**:

$$\mathcal{S}\{a g_{mn} + b h_{mn}\} = a \mathcal{S}\{g_{mn}\} + b \mathcal{S}\{h_{mn}\} \quad \text{mit} \quad a, b \in \mathbb{C}. \quad (8.12)$$

Lineare Operatoren erzeugen die Werte des Ergebnisbildes durch eine reine Linearkombination der Bildwerte des Originalbildes. Bei nichtlinearen Filtern werden auch nichtlineare Funktionen – z. B. Maximums- oder Betragsbildung – herangezogen.

8.2.4.2 Punktoperatoren

Bei den Punktoperatoren wird jedes Pixel des Originalbildes in genau ein Ergebnispixel transformiert (Abb. 8.5):

$$k_{mn} = p_{mn}(g_{mn}), \quad (8.13)$$

wobei die i. Allg. ortsabhängige Funktion p_{mn} die Pixeltransformation beschreibt. Die Indizes dieser Funktion erinnern an die Abhängigkeit der Operation vom Ort.

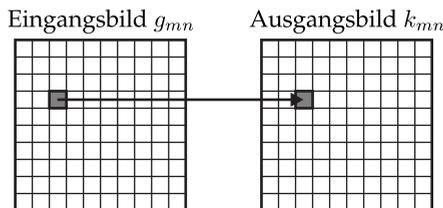


Abbildung 8.5. Punktoperatoren.

Es ist unmittelbar ersichtlich, dass Punktoperatoren verschiebungsfrei und isotrop sind. Sie können allerdings nichtlinear und verschiebungsvariant sein. Ein wichti-

ger Vertreter ist der **Schwelldwertoperator**, der u. a. zur **Bildsegmentierung** eingesetzt wird (Abschn. 8.2.5.1).

Schwelldwertoperator

Der **Schwelldwertoperator** überführt ein Grauwertbild durch Vergleich mit einem Schwellwert γ in ein **Binärbild** mit dem diskreten Wertebereich $\{0, 1\}$:

$$k_{mn} = \begin{cases} 0 & \text{für } g_{mn} < \gamma \\ 1 & \text{für } g_{mn} \geq \gamma \end{cases} \quad (8.14)$$

Zur Darstellung von Binärbildern wird vereinbart, dass „schwarz“ dem Wert 0 und „weiß“ dem Wert 1 entspricht.

Manchmal wird eine etwas abgewandelte Form verwendet, bei der alle Werte kleiner γ zu null gesetzt werden und die restlichen unverändert bleiben:

$$k_{mn} = \begin{cases} 0 & \text{für } g_{mn} < \gamma \\ g_{mn} & \text{für } g_{mn} \geq \gamma \end{cases} \quad (8.15)$$

Der Schwelldwertoperator ist nichtlinear. Wird ein ortsabhängiger Schwellwert verwendet, so ist der Operator verschiebungsvariant.

Grauwertskalierung

Gängige Digitalkameras und Anzeigegeräte stellen Grauwerte als vorzeichenlose Integer-Zahlen mit meist 8 Bits dar. Viele Bildverarbeitungsoperatoren erzeugen jedoch Werte außerhalb des Wertebereiches des Eingangsbildes. Exemplarisch seien die Differenzbildung zweier Bilder und die Differentiation (Abschn. 8.2.4.5) genannt, die i. Allg. auch negative Ausgangswerte liefern. Um solche Ergebnisbilder visualisieren oder weiterverarbeiten zu können, muss eine Skalierung auf den zulässigen Wertebereich des Anzeigegerätes oder der nachfolgenden Filter durchgeführt werden:

- **Clipping:** Beim Clipping werden die negativen Grauwerte zu null gesetzt. Zu hohe Werte werden auf das erlaubte Maximum (meist 255) reduziert. Hierbei ist zu beachten, dass u. U. wichtige Informationen verloren gehen können.
- **Betragsbildung:** Diese Methode ersetzt jeden Grauwert durch seinen Betrag. Negative Werte werden so in den positiven Bereich abgebildet. Dies ist besonders bei der Kantendetektion mittels Differentiationsfiltern interessant.
- **Lineare Skalierung:** Bei dieser Methode wird das Werteintervall des zu skalierenden Bildes linear auf das gewünschte Intervall abgebildet. Als Beispiel sei die Subtraktion zweier Bilder g_{mn} und h_{mn} mit dem Wertebereich $[0, 255]$ betrachtet, bei der sich Werte im Intervall $[-255, 255]$ ergeben können. Mit der Skalierung

$$k_{mn} = \frac{1}{2} (255 + g_{mn} - h_{mn}) \quad (8.16)$$

werden Bildpunkte mit einer Differenz von 0 („schwarz“) auf graue Pixel mit mittlerer Helligkeit abgebildet. Negative Differenzen werden zu dunkleren Grautönen oder schwarz, positive Differenzen zu helleren Grautönen oder weiß (255).

8.2.4.3 Lokale Operatoren

Im Unterschied zu den Punktoperatoren fließen bei den lokalen Operatoren auch die Werte von Bildpunkten der Umgebung („Nachbarschaft“) in das Ergebnispixel ein – es handelt sich also um dynamische, gedächtnisbehaftete Systeme. Die Größe und die Anordnung dieser Nachbarschaft relativ zum Zielpixel wird dabei je nach gewünschtem Effekt unterschiedlich festgelegt. Im einfachsten Fall wird sie als Quadrat definiert. Um die anschließende Berechnung zu vereinfachen, ist es ratsam, immer genau das mittlere Pixel dieser Nachbarschaft als Zielpixel zu verwenden (Abb. 8.6). Dies impliziert, dass die Kantenlänge dieses Quadrates eine ungerade Zahl von Bildpunkten umfassen muss. Im Folgenden wird immer von quadratischen Nachbarschaften ungerader Dimension ausgegangen.

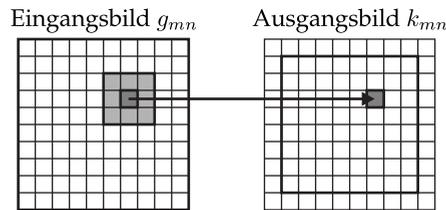


Abbildung 8.6. Lokale Operatoren: Nach der Bearbeitung des Eingangsbildes mit einem lokalen Operator der Größe 3×3 sind die Ränder des Ausgangsbildes nicht definiert.

Aufgrund ihrer vorrangigen Verwendung zur Bildfilterung bezeichnet man die Nachbarschaft auch als **Filtermaske**. Bei den lokalen Operatoren wird diese pixelweise über das Originalbild geschoben und somit die für das jeweilige Zielpixel relevanten Quellbildpunkte „ausmaskiert“. An den Bildrändern treten jedoch Probleme auf: Lässt man die Filtermaske nur bis zum Bildrand laufen, so sind die Ränder des gefilterten Bildes nicht definiert (Abb. 8.6). Um die Bildgröße unverändert zu lassen, muss das Zentrum der Filtermaske bis an die Ränder des Originalbildes laufen. Hierbei muss jedoch der Rand des Ursprungsbildes erweitert werden, damit genügend Werte unter der Filtermaske zur Verfügung stehen. Mögliche Ansätze sind:

- **Zero-Padding:** die äußeren Bereiche werden mit Nullen aufgefüllt;
- **periodische Fortsetzung:** das Bild wird in alle Richtungen periodisch fortgesetzt;
- **symmetrische Fortsetzung:** die Randpixel des Originalbildes werden an den Bildkanten gespiegelt. Bei dieser Methode ist der Randeffekt am geringsten.

Die Ergänzung der Ränder führt jedoch immer zu Verfälschungen im Randbereich des Ausgangsbildes (sog. **Randeffekt**). Deswegen sollte in der Praxis dafür gesorgt werden, dass die wichtigen Bildteile ausreichend weit von den Rändern entfernt sind.

Lineare Filter und diskrete Faltung

Bei den linearen Operatoren ergibt sich der Wert des Ergebnispixels aus einer Linearkombination der durch die Filtermaske ausgewählten Originalpixel. Man kann sich die Filtermaske als eine $(2R + 1) \times (2S + 1)$ -Matrix vorstellen, in welcher die Wichtungsfaktoren für die Linearkombination eingetragen sind:

$$\hat{\mathbf{h}} = \begin{bmatrix} \hat{h}_{-R,-S} & \cdots & \hat{h}_{-R,0} & \cdots & \hat{h}_{-R,S} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \hat{h}_{0,-S} & \cdots & \hat{h}_{0,0} & \cdots & \hat{h}_{0,S} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \hat{h}_{R,-S} & \cdots & \hat{h}_{R,0} & \cdots & \hat{h}_{R,S} \end{bmatrix}. \quad (8.17)$$

Analog zur in Abschn. 8.2.3 beschriebenen Interpretation der Bildmatrix als zweidimensionale Funktion kann auch die Filtermaske in dieser Weise dargestellt werden. Dabei wird das zentrale Ergebnispixel mit den Koordinaten $x = y = 0$ versehen und zur besseren Übersicht im Folgenden fett dargestellt.

Die Berechnungsvorschrift für lineare Operatoren wird damit zu:

$$k_{mn} = \sum_{i=-R}^R \sum_{j=-S}^S \hat{h}_{-i,-j} g_{m-i,n-j}. \quad (8.18)$$

Hierbei bezeichnen m und n die Koordinaten des Zielpixels. R und S ergeben sich aus den Dimensionen der Filtermatrix.

Es sei noch einmal die zweidimensionale Faltung (8.8) betrachtet:

$$k_{mn} = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h_{ij} g_{m-i,n-j}. \quad (8.19)$$

Ersetzt man nun in (8.18) die Filtermaske $\hat{\mathbf{h}}$ durch die an ihrem Symmetriezentrum gespiegelte Maske \mathbf{h} mit $h_{mn} = \hat{h}_{-m,-n}$, so fällt auf, dass es sich bei dieser Transformation um eine Faltung des Bildes g_{mn} mit der gespiegelten Filtermaske \mathbf{h} handelt, die der Impulsantwort h_{mn} des Filters entspricht. Die Filtermaske wird daher auch als **Faltungskern** (engl.: *convolution kernel*) bezeichnet.

Anmerkung: Aufgrund der Voraussetzung von Filtermasken mit ungerader Dimension und in der Mitte liegendem Symmetriezentrum entspricht die Spiegelung am Symmetriezentrum einer Rotation der Maske um 180° .

Nichtlineare Filter

Obwohl lineare Filter in vielen Fällen zufriedenstellende Ergebnisse liefern, haben sie einige prinzipbedingte Nachteile. Insbesondere sind sie nicht in der Lage, impulsförmige Störungen wirksam zu unterdrücken. Darüber hinaus bewirken sie eine oft unerwünschte Verschmierung von Kanten [BPF16]. Durch nichtlineare Funktionen

lassen sich diese Nachteile umgehen. Die resultierenden Filter sind dann allerdings nicht mehr mit Faltungskernen darstellbar.

8.2.4.4 Glättungsfilter

Glättungsfilter sind in der Lage, feine Strukturen wie scharfe Kanten und Rauschen zu unterdrücken und dabei grobe Strukturen zu erhalten. Meist wird gefordert, dass Glättungsfilter sowohl **verschiebungsfrei** als auch **isotrop** sind. Bei linearen Filtern ist die Verschiebungsfreiheit gewährleistet, wenn die Übertragungsfunktion reell ist [Jä12], was wiederum eine symmetrische Impulsantwort impliziert:

$$h_{-m,n} = h_{m,n} \quad \text{sowie} \quad h_{m,-n} = h_{m,n}. \quad (8.20)$$

Im Folgenden werden die wichtigsten Glättungsfilter vorgestellt.

Rechteckfilter

Das **Rechteckfilter** ist ein einfaches Filter, welches eine arithmetische Mittelung der Bildpunkte innerhalb einer Nachbarschaft bewirkt. Im eindimensionalen Fall wird das Rechteckfilter durch eine diskrete Rechteckfunktion dargestellt. Das zweidimensionale Rechteckfilter resultiert aus der Faltung eines horizontalen mit einem vertikalen 1D-Rechteckfilter. Für ein Filter der Größe $L \times L$ ergibt sich:

$$h_{mn,L} = \underbrace{\frac{1}{L^2} \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & & \vdots \\ 1 & \cdots & 1 \end{pmatrix}}_L \Bigg\}_L = \underbrace{\frac{1}{L} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}}_{h_{n,L}} ** \underbrace{\frac{1}{L} (1 \cdots 1)}_{h_{m,L}}, \quad (8.21)$$

Für die DFT-Übertragungsfunktion eines eindimensionalen Rechteckfilters der Länge $L = 3$ erhält man [BPF16]:

$$H_{k,3} = \frac{1}{3} + \frac{2}{3} \cos\left(2\pi \frac{k}{N}\right), \quad (8.22)$$

wobei N die Punkteanzahl der DFT bezeichnet. Abbildung 8.7 (links) zeigt den Verlauf der DFT-Übertragungsfunktion des eindimensionalen Rechteckfilters für die Filterlängen $L \in \{3, 5, 7\}$. Aufgrund der nur langsam und zudem schwingend abfallenden DFT-Übertragungsfunktion ist das Rechteckfilter kein besonders guter Tiefpass. Während Wellenlängen von L Pixeln komplett unterdrückt werden, werden höhere Wellenlängen – wenn auch gedämpft – wieder durchgelassen. Die Dämpfung nimmt nicht monoton mit dem Frequenzindex zu.

Aufgrund der Separierbarkeit des Rechteckfilters ergibt sich die DFT-Übertragungsfunktion des zweidimensionalen Rechteckfilters aus der Multiplikation der ein-

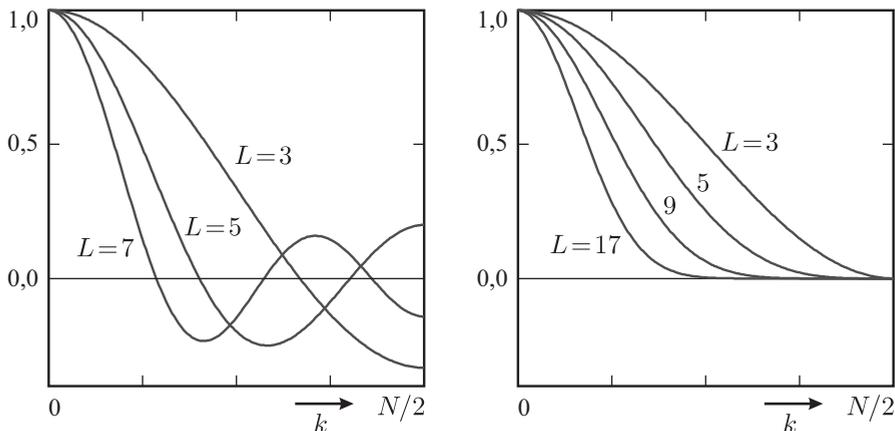


Abbildung 8.7. DFT-Übertragungsfunktionen H_k eindimensionaler Rechteckfilter (links) und eindimensionalen Binomialfilter (rechts) verschiedener Längen L . Der Frequenzindex $N/2$ entspricht der höchsten darstellbaren Signalfrequenz.

dimensionalen Übertragungsfunktionen. Für ein Filter der Größe 3×3 lautet sie:

$$H_{kl,3} = H_{k,3} H_{l,3} = \frac{1}{9} \left(1 + 2 \cos \left(2\pi \frac{k}{N} \right) \right) \left(1 + 2 \cos \left(2\pi \frac{l}{N} \right) \right). \tag{8.23}$$

Anhand des Verlaufes der DFT-Übertragungsfunktion lässt sich die ausgeprägte Anisotropie des Filters erkennen.

Binomialfilter

Das **Binomialfilter** beruht auf der wiederholten Anwendung des elementaren Filters

$$h_{n,2} = \frac{1}{2} \begin{pmatrix} 1 & 1 \end{pmatrix}. \tag{8.24}$$

Die wiederholte Anwendung eines Filters entspricht im Fourier-Bereich einer mehrfachen Multiplikation der Übertragungsfunktion mit sich selbst, was zu einer Faltung der Impulsantworten führt. Damit erhält man für Filterlängen $L > 2$:

$$h_{n,3} = h_{n,2} * h_{n,2} = \frac{1}{4} \begin{pmatrix} 1 & 2 & 1 \end{pmatrix}, \tag{8.25}$$

$$h_{n,4} = h_{n,3} * h_{n,2} = \frac{1}{8} \begin{pmatrix} 1 & 3 & 3 & 1 \end{pmatrix}, \tag{8.26}$$

$$h_{n,L+2} = h_{n,L} * h_{n,3}. \tag{8.27}$$

Analog setzen sich die Filtermasken des zweidimensionalen Binomialfilters aus zwei eindimensionalen Binomialfiltern der Länge L in den beiden Raumrichtungen zusam-

men. Für das 2D-Binomialfilter der Größe 3×3 erhält man beispielsweise:

$$h_{mn,3} = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} = \underbrace{\frac{1}{4} \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}}_{h_{n,3}} \cdot \underbrace{\frac{1}{4} \begin{pmatrix} 1 & 2 & 1 \end{pmatrix}}_{h_{m,3}}. \quad (8.28)$$

Um Verschiebungsfreiheit zu erzielen, werden in der Praxis die ungeradzahligen Kerne $h_{mn,3}, h_{mn,5}, \dots$ bevorzugt.

Die DFT-Übertragungsfunktion des eindimensionalen Binomialfilters der Länge $L = 3$ lautet [BPF16]:

$$H_k = \frac{1}{2} + \frac{1}{2} \cos\left(2\pi \frac{k}{N}\right). \quad (8.29)$$

In Abb. 8.7 (rechts) ist die DFT-Übertragungsfunktion des Binomialfilters für verschiedene Filterlängen dargestellt [BPF16]. Da in sämtlichen Fällen die Übertragungsfunktion monoton abfällt und für $k = N/2$ den Wert null erreicht, handelt es sich beim Binomialfilter um einen besseren Tiefpass als das Rechteckfilter. Aufgrund des zentralen Grenzwertsatzes konvergiert das Binomialfilter für $L \rightarrow \infty$ gegen das kontinuierliche Gauß-Filter. Die Zusammensetzung des Filterkerns aus zwei eindimensionalen Filterkernen, die ihrerseits wieder aus der Faltung eines elementaren Filterkerns mit sich selbst hervorgehen, ermöglicht eine effiziente rekursive Implementierung.

Medianfilter

Binomial- und Rechteckfilter werden zur Verringerung von mittelwertfreiem, normalverteiltem Rauschen eingesetzt, wobei allerdings die Schärfe des Ursprungsbildes abnimmt. Bei sogenanntem **Impulsrauschen**, also bei einzelnen fehlerhaften Pixeln, sind lineare Filter weniger gut geeignet. Der stark abweichende Grauwert einzelner Pixel wird zwar vermindert, allerdings wirkt sich ein solcher „Ausreißer“ durch die Mittelung auch auf seine Nachbarn aus [BPF16].

Das **Medianfilter** dagegen liefert bei impulsförmigen Störungen sehr gute Ergebnisse. Als nichtlineares Filter kann es nicht durch eine Faltung implementiert werden. Stattdessen werden die Grauwerte innerhalb der Filtermaske in eine Liste geschrieben und der Wertigkeit entsprechend sortiert. Das Element in der Mitte dieser Liste ist der **Median** oder Zentralwert, der als Filterergebnis in die entsprechende Stelle des Ausgangsbildes eingetragen wird (Abb. 8.8). Da der Grauwert eines fehlerhaften Pixels sich in der Regel sehr stark von den Grauwerten seiner Umgebung unterscheidet, ist es unwahrscheinlich, dass dieser nicht unterdrückt wird. Hingegen bleiben andere Strukturen, die eine Veränderung der Grauwerte von Pixel zu Pixel beinhalten, erhalten. Dies ist z. B. bei Kanten oder monotonen Grauwertänderungen, die sich über mehrere Pixel erstrecken, der Fall. Die Schärfe des Bildes bleibt nahezu unverändert.

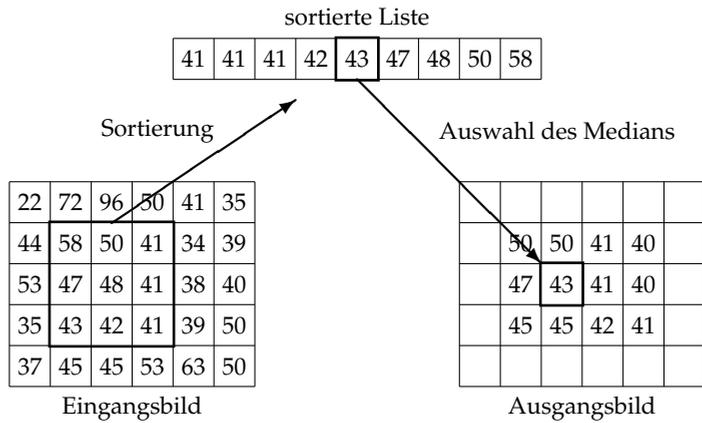
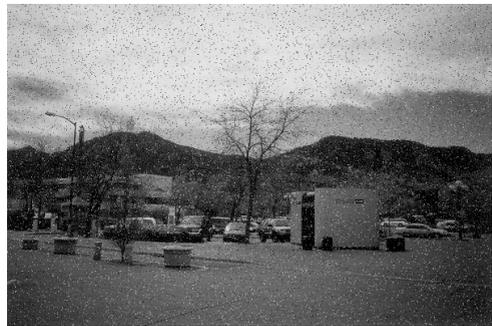


Abbildung 8.8. Illustration des Medianfilters der Größe $L \times L = 3 \times 3$ (nach [Jä12]).



(a) Originalbild



(b) Impulsverraushtes Bild



(c) Binomialgefiltertes Bild



(d) Mediangefiltertes Bild

Abbildung 8.9. Vergleich von Binomial- und Medianfilter bei impulsförmigem Rauschen.

Abschließend zeigt Abb. 8.9 das Ergebnis der Glättung eines impulsförmig verrauschten Bildes mit einem Binomialfilter und einem Medianfilter.

Rangordnungsfilter

Bei diesem Filtertyp handelt es sich um eine Verallgemeinerung des Medianfilters: Anstatt des mittleren Wertes aus der sortierten Liste der Grauwerte wird hier der n -te Wert als Filterwert in das Ergebnispixel geschrieben. Es muss somit gelten: $0 \leq n < L^2$, wobei L die Kantenlänge der Filtermaske bezeichnet und die Elemente der Liste von 0 bis $L^2 - 1$ durchnummeriert sind. Je nach Einsatzzweck wird der Parameter n als Konstante festgelegt oder variabel in Abhängigkeit vom Ort gewählt.

8.2.4.5 Kantendetektion

Eine **Kante** ist ein (im Idealfall sprunghafter) Übergang zwischen zwei Bereichen mit ungefähr konstanten Grauwerten. Aufgrund dieser Definition werden primär zwei Anforderungen an Kantendetektoren gestellt:

- Erkennung von Grauwertänderungen,
- Unterdrückung von Bereichen mit konstantem Grauwert.

Kantendetektoren müssen somit örtlich schnell veränderliche Grauwertänderungen hervorheben, wohingegen lineare Glättungsfiler gerade solche Änderungen abschwächen. Dagegen sollen örtlich konstante oder langsam veränderliche Werte unterdrückt werden. Des Weiteren müssen Kantendetektoren verschiebungsfrei sein, damit sie die Position einer Kante nicht beeinflussen. Am häufigsten werden dafür differenzierende Verfahren verwendet [BPF16].

Die Wirkung der ersten und zweiten Ableitung auf eine Kante lässt sich Abb. 8.10 entnehmen. Gebiete konstanten Grauwertes werden in den Ableitungen unterdrückt, wohingegen an den Positionen der Kanten in der ersten Ableitung **Extremwerte** resultieren. In der zweiten Ableitung zeigen sich Kanten als **Nulldurchgänge**, die von Peaks umgeben sind. Diese entstehen durch die Wendepunkte der ersten Ableitung.

Gradientenfilter

Da die Differentiation für diskrete Signale nicht definiert ist, wird diese meist durch diskrete lineare Filter approximiert. Einfache Approximationen für den eindimensionalen Fall sind der unsymmetrische und der symmetrische Differenzenquotient:

$$\frac{\partial g(x)}{\partial x} \approx \frac{g(x) - g(x - \Delta x)}{\Delta x} \quad (\text{unsymmetrisch, rückwärts}), \quad (8.30)$$

$$\frac{\partial g(x)}{\partial x} \approx \frac{g(x + \Delta x) - g(x)}{\Delta x} \quad (\text{unsymmetrisch, vorwärts}), \quad (8.31)$$

$$\frac{\partial g(x)}{\partial x} \approx \frac{g(x + \Delta x) - g(x - \Delta x)}{2 \Delta x} \quad (\text{symmetrisch}). \quad (8.32)$$

Die zugehörigen Impulsantworten sind in Tab. 8.2 zusammengefasst.

Bei zweidimensionalen Bildsignalen erfolgt die Differentiation üblicherweise in x - und in y -Richtung. Die entsprechenden partiellen Ableitungen sind die Komponen-

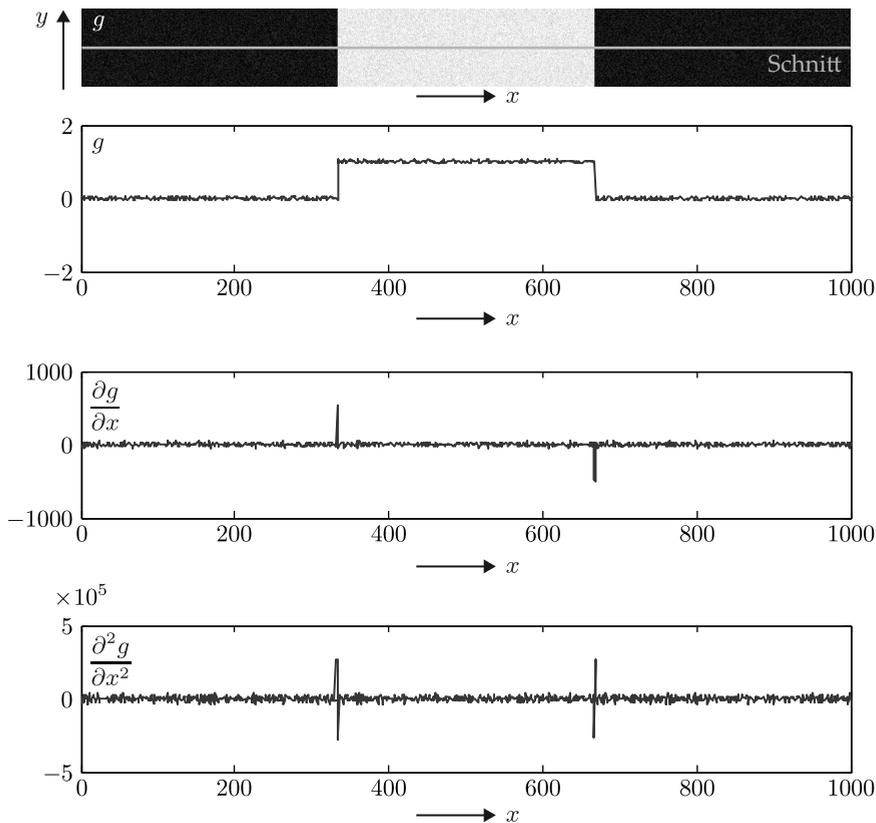


Abbildung 8.10. Auswirkungen von Kanten auf die erste und zweite Ableitung des Bildsignals (nach [BPF16]). Die Diagramme zeigen die Bildwerte sowie die erste und zweite Ableitung entlang des Schnitts.

Tabelle 8.2. Impulsantworten d_n einfacher linearer Ableitungsfiler:

diskreter Ort n	-2	-1	0	1	2
unsymmetrischer Differenzenquotient (8.30)	0	0	1	-1	0
unsymmetrischer Differenzenquotient (8.31)	0	1	-1	0	0
symmetrischer Differenzenquotient (8.32)	0	$\frac{1}{2}$	0	$-\frac{1}{2}$	0

ten des **Bildgradienten**:

$$\text{grad } g(x, y) = \left(\frac{\partial g(x, y)}{\partial x}, \frac{\partial g(x, y)}{\partial y} \right)^T. \tag{8.33}$$

Liegt nun am Ort x, y eine Kante vor, so ist der Gradientenbetrag $\|\text{grad } g(x, y)\|$ ein Maß für die Ausprägung der Kante.

Im Diskreten werden die partiellen Ableitungen durch den symmetrischen Differenzenquotienten approximiert:

$$\frac{\partial g(x, y)}{\partial x} \approx \frac{g(x + \Delta x, y) - g(x - \Delta x, y)}{2\Delta x}, \quad (8.34)$$

$$\frac{\partial g(x, y)}{\partial y} \approx \frac{g(x, y + \Delta y) - g(x, y - \Delta y)}{2\Delta y}. \quad (8.35)$$

Abbildung 8.11 zeigt die damit berechnete Gradientenapproximation an einem Beispielbild [BPF16]. Da die partiellen Ableitungen in x - und y -Richtung i. Allg. auch negative Werte annehmen, wurden die resultierenden Werte zur ihrer Darstellung linear auf das Intervall $[0, 255]$ skaliert.

Schließlich sei angemerkt, dass obwohl die Berechnung der partiellen Ableitungen durch lineare Filter erfolgt, die Berechnung des Gradientenbetrags keine lineare Operation mehr darstellt. Insofern ist der vorgestellte Kantendetektor nichtlinear.

Prewitt-Operator

Ein Problem von Differentiationsfiltern ist der mit der Frequenz ansteigende Betrag der Übertragungsfunktion. Dadurch werden auch hochfrequente Störungen verstärkt, was das Ergebnis der Kantendetektion negativ beeinflussen kann. Zur Beherrschung dieses Problems haben sich sogenannte **regularisierte Kantendetektoren** etabliert, die sich aus einem Ableitungsfiler und einem orthogonal dazu stehenden Glättungsfiler zusammensetzen. Derartige Filter sind allerdings nicht isotrop: Sie erkennen nur Kanten, die in bestimmten Raumrichtungen verlaufen. Ein einfaches Beispiel ist der **Prewitt-Operator**, der auf einem symmetrischen Differenzenquotienten und einem Rechteckfilter basiert, wobei auf eine Normierung verzichtet wird:

$$p_{mn}^x = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} ** \begin{pmatrix} 1 & 0 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}, \quad p_{mn}^y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}. \quad (8.36)$$

Das hochgestellte Symbol x bzw. y deutet dabei die Richtung der Differentiation an.

Sobel-Operator

Der **Sobel-Operator** ist dem Prewitt-Operator sehr ähnlich – allerdings wird hier zur Glättung ein Binomialfilter eingesetzt:

$$s_{mn}^x = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} ** \begin{pmatrix} 1 & 0 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}, \quad s_{mn}^y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}. \quad (8.37)$$

Dadurch weist dieses Filter bessere Glättungs- und Isotropieeigenschaften auf.

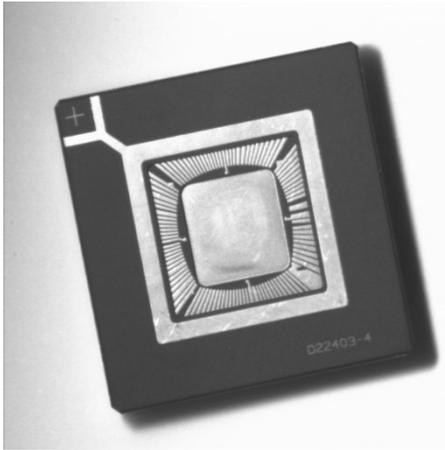
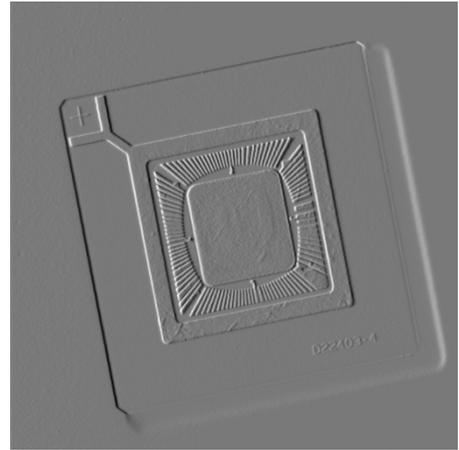
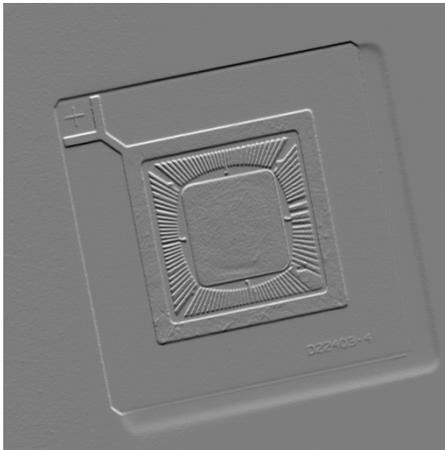
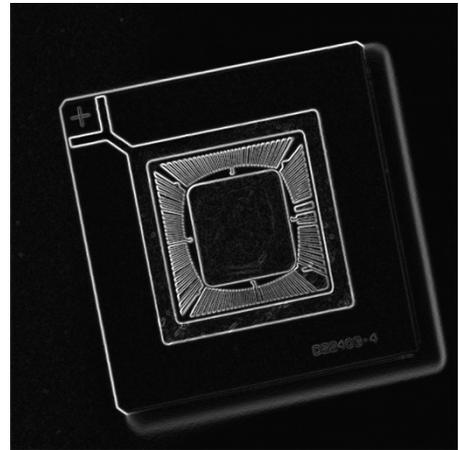
(a) $g(x, y)$ (b) $\frac{\partial g(x, y)}{\partial x}$ (c) $\frac{\partial g(x, y)}{\partial y}$ (d) $\|\text{grad } g(x, y)\|$

Abbildung 8.11. Kantendetektion mit dem symmetrischen Differenzenquotienten: (a) Originalbild; (b) symmetrischer Differenzenquotient in x -Richtung (schwarz $\hat{=}$ negative Werte, grau $\hat{=}$ 0, weiß $\hat{=}$ positive Werte); (c) symmetrischer Differenzenquotient in y -Richtung; (d) Gradientenbetrag (schwarz $\hat{=}$ 0) [BPF16].

Laplace-Filter

Durch zweimalige Anwendung des eindimensionalen unsymmetrischen Differenzenquotienten erhält man die zweite Ableitung eines Signals. Die Faltung der zugehörigen Impulsantwort d_n mit sich selbst beschreibt einen Operator, welcher die zweite Ableitung in einem einzigen Schritt erzeugt:

$$l_n = (1 \ -1) * (1 \ -1) = (1 \ -2 \ 1). \quad (8.38)$$

Der **Laplace-Operator** ist definiert als die Summe der zweiten Ableitungen einer mehrdimensionalen Funktion nach jedem ihrer Parameter [Mer10]. Mit der oben hergeleiteten diskreten Approximation der zweiten Ableitung kann man den zweidimensionalen diskreten Laplace-Operator wie folgt angeben:

$$l_{mn} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}. \quad (8.39)$$

Für die zweidimensionale Übertragungsfunktion ergibt sich:

$$L_{kl} = -4 \sin^2\left(\frac{\pi k}{2}\right) - 4 \sin^2\left(\frac{\pi l}{2}\right). \quad (8.40)$$

Dieses Ergebnis weicht für große Frequenzen stark von dem idealen Laplace-Operator ab. Ein weiteres Problem ist die Rauschverstärkung, die zwangsläufig beim Ableiten in Kauf zu nehmen ist und bei der zweiten Ableitung noch stärker auftritt als bei der ersten. Durch Vorglättung mit einem Tiefpassfilter können zumindest die durch hochfrequentes Rauschen verursachten Störungen verringert werden.

8.2.4.6 Einfache Kantendetektoren

Im Folgenden wird eine Auswahl einfacher Kantendetektoren vorgestellt, die in *LabVIEW* im VI IMAQ EdgeDetection vordefiniert sind [Nat03].

Nichtlineare Prewitt- und Sobel-Filter

Diese Filter kombinieren zwei zueinander orthogonale Prewitt- bzw. Sobel-Operatoren. Um die aufwendigere Berechnung des Gradientenbetrags zu vermeiden, erfolgt ihre Kombination durch Maximumbildung der Beträge der partiellen Ableitungen:

$$k_{mn} = \max(|g_{mn} ** p_{mn}^x|, |g_{mn} ** p_{mn}^y|). \quad (8.41)$$

Nichtlineares Gradientenfilter

Für jedes Pixel wird die Differenz zum darüberliegenden Nachbarpixel sowie die Differenz zwischen linkem Nachbarn und dem linken oberen Nachbarn betragsmäßig gebildet. Das Maximum dieser beiden Werte wird in das Zielpixel geschrieben:

$$k_{mn} = \max(|g_{m,n-1} - g_{m,n}|, |g_{m-1,n-1} - g_{m-1,n}|). \quad (8.42)$$

Roberts-Operator

Der **Roberts-Operator** ist ähnlich dem nichtlinearen Gradientenfilter, allerdings werden die Ableitungen hier „über Kreuz“ gebildet:

$$k_{mn} = \max(|g_{m-1,n-1} - g_{m,n}|, |g_{m,n-1} - g_{m-1,n}|). \quad (8.43)$$

Differentiationsfilter

Der neue Pixelwert ergibt sich aus dem größten Betrag der Differenzen eines Pixels zu seinen drei linken oberen Nachbarn:

$$k_{mn} = \max(|g_{m-1,n} - g_{m,n}|, |g_{m-1,n-1} - g_{m,n}|, |g_{m,n-1} - g_{m,n}|). \quad (8.44)$$

8.2.5 Segmentierung und Klassifikation

Unter **Segmentierung** versteht man die Zerlegung eines Bildes in getrennte bedeutungsvolle Bereiche – z. B. Hintergrund und Objekte. Dies stellt einen wichtigen Schritt auf dem Weg zur automatischen Bildauswertung dar. Durch die Segmentierung wird die Lage und Ausrichtung einzelner Objekte im Bild ermittelt, was eine Vermessung oder Anwesenheitsprüfung überhaupt erst ermöglicht. In diesem Abschnitt werden lediglich grundlegende Verfahren vorgestellt, weitere finden Sie unter anderem in [BPF16, Jä12].

8.2.5.1 Binärsegmentierung

Die Binärsegmentierung beruht auf der Annahme, dass sich die Grauwerte der zu erkennenden Objekte signifikant von denjenigen des Hintergrundes unterscheiden. Trifft diese Annahme auf die zu untersuchende Szene zu, so kann die Segmentierung einfach mittels eines Schwellwertoperators durchgeführt werden. Falls es gelingt, eine gleichmäßige Ausleuchtung des gesamten Bildes zu erreichen, kann ein globaler Schwellwert festgelegt werden, anderenfalls ist u. U. ein ortsabhängiger Schwellwert erforderlich.

8.2.5.2 Kantenorientierte Segmentierung

Kantenantastung

Dieses wenig rechenaufwändige Verfahren eignet sich besonders zur Ermittlung der exakten Lage und Ausrichtung eines Objektes vor möglichst homogenem Hintergrund. Dazu wird beim Systementwurf ein meist rechteckiger Bereich des Bildes definiert, in welchem eine Kante des Objektes erwartet wird. Dieser Bereich wird mit parallelen Geraden ausgefüllt, wobei diese Geraden nicht parallel zur erwarteten Objektkante liegen dürfen (Abb. 8.12). Das System approximiert dann mittels Differenzenquotienten die Ableitung des Bildes entlang dieser Geraden und sucht die Position des Maximums der Ableitung auf jeder Geraden. Bei genügend großem Kontrast

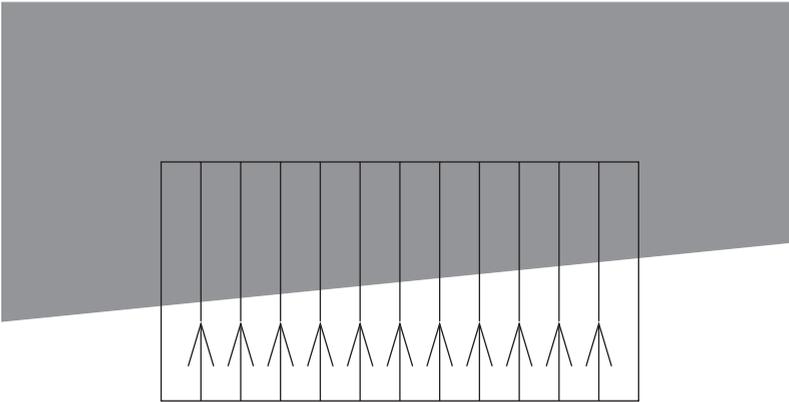


Abbildung 8.12. Funktionsweise der Kantenantastung.

zwischen Objekt und Hintergrund liegen diese Maxima genau auf der Objektkante. Bei geraden Objektkanten muss nur noch eine Gerade durch die gefundenen Punkte gelegt werden (z. B. mittels einer LS-Schätzung), um die Kante zu lokalisieren.

Im praktischen Einsatz werden normalerweise zwei solcher Bereiche definiert, in denen zwei zueinander orthogonale Objektgrenzen gesucht werden. Auf diese Weise kann sehr schnell ein Koordinatensystem festgelegt werden, welches als Bezugssystem zur Vermessung des Objektes eingesetzt werden kann. Das Verfahren ist auch bei Objekten ohne gerade orthogonale Kanten anwendbar, jedoch sind dann je nach Kantenverlauf andere Algorithmen erforderlich, um aus den ermittelten Punkten die Kante zu approximieren.

8.2.5.3 Template-Matching

Template-Matching ist ein mächtiges Verfahren, welches auch zur Merkmalsextraktion und zur Klassifikation eingesetzt wird. Es beruht darauf, ein oder mehrere vorher definierte Muster oder Vorlagen (englisch: *templates*) im Bild zu lokalisieren. Dadurch lässt sich beispielsweise untersuchen, ob Objekte mit einem bestimmten Aussehen im Bild auftauchen, und es kann ihre Position ermittelt werden.

Zweidimensionale Kreuzkorrelation

Eine einfache Möglichkeit dafür ist es, die (zweidimensionale) Kreuzkorrelationsfunktion des Musters mit dem Bild zu berechnen. Die KKF ist ein Maß für die Ähnlichkeit zweier – ggf. gegeneinander verschobener – Signale [BPF16]:

$$\hat{r}_{mn} = \sum_{i_1=0}^{T_1-1} \sum_{i_2=0}^{T_2-1} h_{i_1 i_2} \cdot g_{m+i_1, n+i_2} \quad (8.45)$$

Hierbei bezeichnen T_1 und T_2 die Höhe und Breite des Templates in Pixeln; im Folgenden sei $T = T_1 = T_2$. Für jedes Element der KKF (8.45) sind T^2 Multiplikationen erforderlich. Da Additionen meist weniger Zeit als Multiplikationen benötigen, werden diese vernachlässigt. Für die gesamte KKF ergibt sich damit ein Rechenaufwand von $M \cdot N \cdot T^2$ Multiplikationen. Nimmt man die Bildgröße als konstant an, so kann man den Aufwand in Abhängigkeit der Template-Größe durch $\mathcal{O}(T^2)$ beschreiben.

Ähnlich wie die zweidimensionale Faltung kann auch die 2D-KKF mit Hilfe der FFT berechnet werden. Dazu wird das Template durch Zero-Padding auf die Größe des Bildes gebracht. Es sind dann zwei Fourier-Transformierte (FT) und eine inverse FT zu berechnen. Bei quadratischen Bildern ($M = N$) beträgt der Aufwand damit $(3N^2 \cdot \ln N + N^2)$ Multiplikationen unabhängig von der Template-Größe.

Anwendung der KKF beim Template-Matching

Die Stellen, an welchen die KKF einen bestimmten Schwellwert überschreitet, können als Objektpositionen angenommen werden. Allerdings ist dieses Verfahren anfällig gegenüber Beleuchtungsschwankungen. Auch können gegenüber dem Muster gedrehte bzw. vergrößerte oder verkleinerte Objekte nicht gefunden werden. Diesem Problem lässt sich zwar abhelfen, indem man Templates verschiedener Größen und Rotationswinkel anlegt und für alle einzeln die KKF berechnet, allerdings wird das Verfahren dann schnell sehr rechenaufwändig. In der Praxis werden daher meist weiterentwickelte Algorithmen eingesetzt, die robust gegenüber Rotation, Skalierung oder auch Beleuchtungsschwankungen sind [BPF16, Nat03].

8.3 Realisierung in LabVIEW

Die grundlegende Bedienung von *LabVIEW* wird als bekannt vorausgesetzt. Hier soll lediglich auf die VIs zur Steuerung externer Hardware über die RS232-Schnittstelle (NI-VISA) sowie die Bildverarbeitung mit IMAQ Vision eingegangen werden.

8.3.1 Kurzbeschreibung NI-VISA für RS232

Die VISA (*Virtual Instrument System Architecture*)-Architektur ist ein herstellerübergreifender Standard, der die Kommunikation einzelner Systemkomponenten über unterschiedliche Schnittstellen (z. B. GPIB, RS232, VXI) ermöglicht. Es gibt dabei einerseits geräteabhängige Funktionen, die die Steuerung eines Gerätes unabhängig von der verwendeten physikalischen Schnittstelle erlauben. Darüber hinaus stehen noch schnittstellenabhängige Funktionen bereit, die das Ausnutzen aller Besonderheiten eines bestimmten Schnittstellentyps ermöglichen.

Für diesen Versuch sind lediglich einige der spezifischen Funktionen der RS232-Schnittstelle erforderlich. Die hierfür relevanten VIs finden Sie in der Funktionenpalette unter Funktionen → Instrumenten-I/O → Seriell.

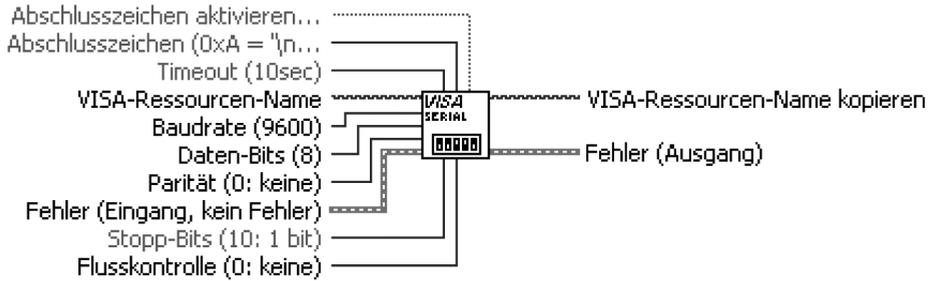


Abbildung 8.13. VISA: Seriellen Port konfigurieren.



Abbildung 8.14. VISA: Schreiben.

VISA: Seriellen Port konfigurieren (VISA Configure Serial Port)

Dieses VI ist vor jeder Benutzung der seriellen Schnittstelle aufzurufen, um die Schnittstellenparameter zu konfigurieren (Abb. 8.13). Für die Ansteuerung der Versuchshardware sind zwei der Eingänge mit Konstanten zu beschalten (rechte Maustaste auf den Eingangspin → Erstellen → Konstante):

- **VISA-Ressourcenname:** Es wird ein Drop-Down-Feld zur Festlegung der Konstante angelegt. Wählen Sie dort die Schnittstelle *COM1*.
- **Baudrate:** Ändern Sie den voreingestellten Wert (9600) auf den Wert 2400 ab.

Alle weiteren Eingänge dieses VIs sollten unbeschaltet bleiben, damit die jeweiligen Standardeinstellungen benutzt werden.

VISA: Schreiben (VISA Write)

Mit diesem VI können Zeichen(folgen) auf der seriellen Schnittstelle gesendet werden (Abb. 8.14). Folgende Eingänge sind zu verdrahten:

- **VISA-Ressourcenname:** Verbinden Sie diesen Eingang mit dem Ausgang „VISA-Ressourcenname (Ausgang)“ des vorgeschalteten Konfigurations-VIs.
- **Schreibpuffer:** An diesem Eingang muss eine Zeichenfolge übergeben werden, welche dann auf der Schnittstelle ausgesendet wird.

Die Kommandos der Versuchshardware bestehen aus einzelnen ASCII-Zeichen. Sie sind in Tab. 8.3 aufgelistet.

VISA: Schließen (VISA Close)

Dieses VI ist stets nach Beendigung aller Schnittstellenzugriffe aufzurufen, um diese wieder für andere Anwendungsprogramme freizugeben (Abb. 8.15). Dazu ist der

Tabelle 8.3. Kommandos der Versuchshardware.

Befehlscode	Bedeutung
F	„forward“ – drehe den Teller gegen den Uhrzeigersinn
R	„reverse“ – drehe den Teller im Uhrzeigersinn
S	„stop“ – stoppe die Drehung des Tellers
1	Drehgeschwindigkeit: langsam
2	Drehgeschwindigkeit: normal (Standardwert beim Einschalten)
3	Drehgeschwindigkeit: schnell
L	Schalte die Beleuchtung ein (<i>Befehlscode = großes L</i>)
l	Schalte die Beleuchtung aus (<i>Befehlscode = kleines L</i>)

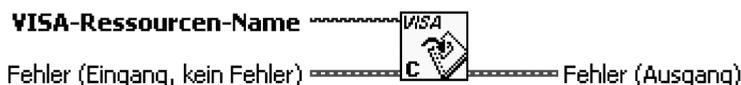


Abbildung 8.15. VISA: Schließen.

Eingang VISA-Ressourcenname mit dem Ausgang des zuletzt aufgerufenen VISA: schreiben-Blockes zu verbinden.

8.3.2 Bildverarbeitung mit IMAQ Vision for LabVIEW

Das Toolkit *Vision* von NI bietet eine umfangreiche Palette vorgefertigter Funktionen zur Bildverarbeitung. Einige dieser Werkzeuge sollen hier vorgestellt werden.

Die VIs für IMAQ Vision finden Sie in der Funktionenpalette unter Bilderkennung und Motorsteuerung → Vision Utilities → Image Management.

8.3.2.1 Arbeiten mit Bildern

IMAQ Create

Mit diesem VI wird ein Speicherbereich für ein Einzelbild reserviert. Über den Parameter *Image Name* muss ein eindeutiger Name für das Bild vergeben werden. Vorsicht: Wird dieses VI zweimal mit dem gleichen Namen aufgerufen, so meldet LabVIEW zwar keinen Fehler, es wird aber insgesamt nur *ein* Speicherbereich angelegt!

Soll das Bild im weiteren Verlauf mit einem der auf Filtermasken basierenden Operatoren verarbeitet werden, so ist ein entsprechender Randbereich festzulegen. Dies kann über den Parameter *Border Size* erfolgen: Um das Bild herum wird anschließend automatisch ein Bereich dieser Breite gelegt, der mit Nullen gefüllt wird. Auch bei einer Änderung der Bildgröße wird dieser Bereich automatisch mit angepasst. Der Standardwert für *Border Size* beträgt 3, d. h. es können Filter mit einer maximalen Kerngröße von 7×7 verwendet werden.

Der Parameter *Image Type* kann in diesem Versuch stets mit einer Konstante des Wertes *Grayscale (U8)* beschaltet werden.

IMAQ Dispose

Dieses VI ist unbedingt vor Beendigung des Haupt-VIs für jedes erzeugte Bild aufzurufen, um den reservierten Speicher wieder freizugeben.

Darstellung von Bildern auf dem Bildschirm

Das zum Anzeigen von Bildern benötigte VI ist in der Elementepalette **im Frontpanel-Editor** unter *Vision* → *Image Display* zu finden. Sobald es im Frontpanel eingefügt wird, ist im Blockdiagramm ein entsprechendes Symbol mit nur einem Eingang zu finden. Entsprechend der gewünschten Datenflussreihenfolge ist hier der Ausgang der letzten Stufe der Bildverarbeitung anzuschließen.

8.3.2.2 Zugriff auf die FireWire-Kamera

Die für FireWire benötigten VIs sind unter *Bilderkennung und Motorsteuerung* → *NI-IMAQdx* zu finden. In diesem Versuch sind folgende Blöcke zu verwenden:

IMAQdx Open.vi

Um überhaupt Bilder von der Kamera geliefert zu bekommen, ist zunächst dieses VI aufzurufen. Es können die Standardwerte der Eingänge verwendet werden: Rechtsklick auf Eingang → Erstellen → Konstante. Der Ausgang *Session Out* ist mit dem entsprechenden Eingang der nachfolgend beschriebenen VIs zu verbinden (*Session In*). Durch Verketteten der verschiedenen Bildverarbeitungs-VIs mittels der *Session-Anschlüsse* kann die gewünschte Ausführungsreihenfolge vorgegeben werden (Datenflussprogrammierung).

IMAQdx Close.vi

Vor Beendigung des Haupt-VIs ist unbedingt die Kamera mit Hilfe dieses VIs wieder freizugeben. Dabei ist der Eingang *Session In* wie oben beschrieben zu verdrahten.

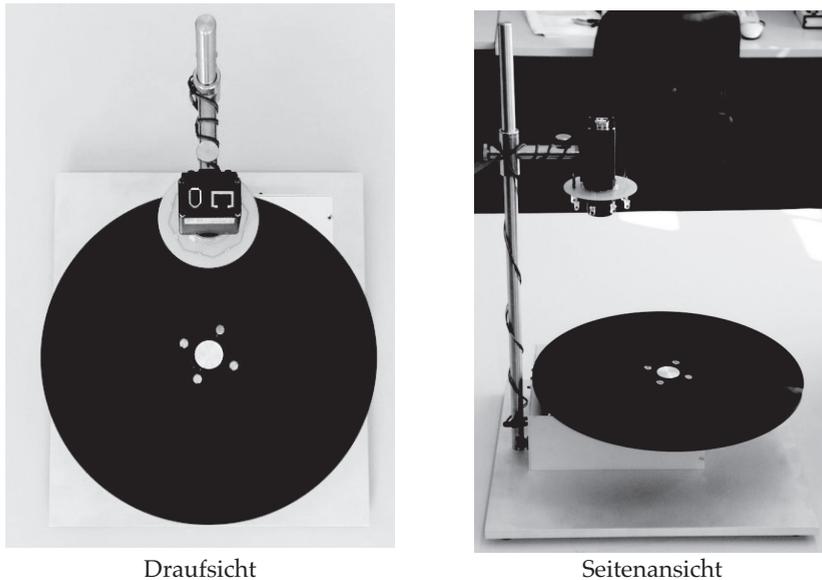
IMAQdx Configure Grab.vi und IMAQdx Grab.vi

Mit dem VI *IMAQdx Configure Grab.vi* wird die Aufnahme kontinuierlicher Bildsequenzen gestartet: Das System empfängt daraufhin kontinuierlich Einzelbilder von der Kamera. Die Bildwiederholrate wird dabei von der Kamera vorgegeben, sie beträgt bei der hier verwendeten Kamera maximal 60 Bilder pro Sekunde. Um diese Bilder weiterverarbeiten zu können, ist das VI *IMAQdx Grab.vi* einzusetzen: hier muss über den Eingang *Image In* ein mittels *IMAQ Create* angelegter Image-Speicherbereich angegeben werden, in welchen das zuletzt von der Kamera gelieferte Einzelbild kopiert wird. Um Bildsequenzen zu verarbeiten, ist *IMAQdx Grab.vi* z. B. mit einer *While-Schleife* zyklisch aufzurufen; bei jedem Aufruf wird das jeweils aktuelle Bild aus dem Empfangspuffer übernommen.

Achten Sie darauf, auch bei diesen VIs die Anschlüsse *Session In* und *-out* in der korrekten Datenflussreihenfolge zu verdrahten.

8.4 Versuchsaufbau

Der Versuchsaufbau (Abb. 8.16) besteht aus einem Kamerasystem (Kamera, Objektiv, Datenübertragung, Beleuchtung) und einem Drehteller für Testplatinen. Der Drehteller kann über einen Schrittmotor in Bewegung gesetzt werden. Die Ansteuerung des Motors und der Beleuchtung erfolgt über ein VI, das über die serielle Schnittstelle mit dem Versuchsaufbau kommunizieren kann. Die Bedienelemente des Versuchsaufbaus sehen Sie in Abb. 8.17.



Draufsicht

Seitenansicht

Abbildung 8.16. Versuchsaufbau: Ansichten.

8.5 Versuchsvorbereitende Aufgaben

Aufgabe 8.1: Bildaufnahme

- Beschreiben Sie die Unterschiede im Aufbau und die wichtigsten Eigenschaften von CCD- und CMOS-Bildsensoren.
- Geben Sie den Dezimalwert an, der bei der Basler-Kamera als Gain eingestellt werden muss, wenn eine Verstärkung um 6 dB erreicht werden soll (Brightness=0). Zeichnen Sie die resultierende Zuordnungsfunktion (Achsenbeschriftungen wie in Abb. 8.2).
- Nennen Sie drei Maßnahmen, um Unter- oder Überbelichtung bei der Aufnahme mit einer Kamera auszugleichen.

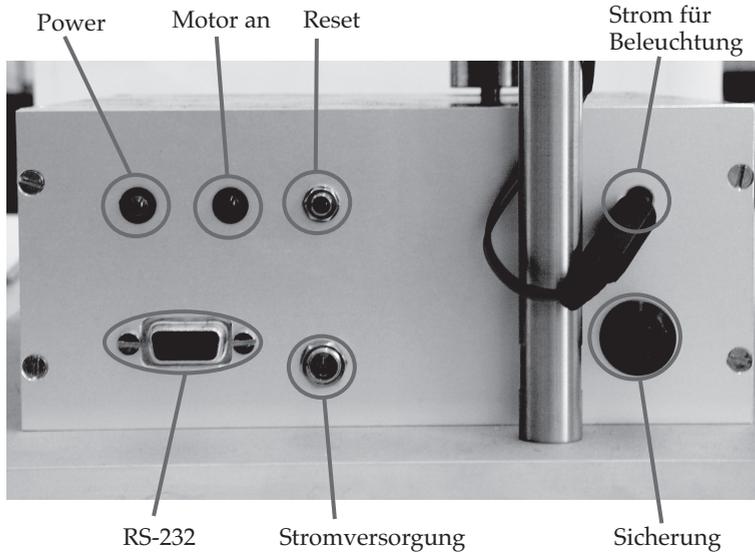


Abbildung 8.17. Versuchsaufbau: Bedienelemente.

Aufgabe 8.2: Bildvorverarbeitung

- a) Beschreiben Sie qualitativ die Effekte, die die folgenden Impulsantworten bei der Faltung mit Bildern hervorrufen (Stichworte):

$$h_{1,mn} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad h_{2,mn} = \frac{1}{7} \begin{bmatrix} 1 & 0 & -1 \\ 1 & 1 & -1 \\ 1 & 0 & -1 \end{bmatrix}, \quad h_{3,mn} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}. \quad (8.46)$$

Hinweis: Die Vorfaktoren brauchen nicht berücksichtigt zu werden.

- b) Welchen prinzipiellen Nachteil haben lineare Glättungsfilter gegenüber nichtlinearen Filtern?
- c) Nennen Sie drei Möglichkeiten, wie die Randproblematik bei der Filterung digitaler Bilder umgangen werden kann unter der Bedingung, dass das gefilterte Bild die gleiche Größe hat wie das Ursprungsbild.
- d) Berechnen Sie den zweidimensionalen Filterkern eines Binomialfilters der Größe 5×5 . Geben Sie des Weiteren einen 3×3 -Prewitt-Filterkern an, welcher horizontale Kanten erkennt.

Aufgabe 8.3: Bilderkennung

- a) Nennen Sie die zwei grundlegenden Typen von Kantendetektionsfiltern. Wie werden Kanten durch diese beiden Typen dargestellt?
- b) Was bedeutet Isotropie? Warum ist sie insbesondere bei der Kantendetektion so wichtig?
- c) Überlegen Sie, wie die hier vorgestellten *LabVIEW*-VISA-Blöcke beschaltet werden müssen, damit eine fest vorgegebene Zeichenfolge auf der seriellen Schnittstelle ausgegeben wird. Skizzieren Sie dazu das Blockdiagramm.

8.6 Aufgaben am Versuchstag

Im praktischen Teil des Versuchs werden Sie zu Beginn die Bildaufnahme in *LabVIEW* implementieren und anschließend die Effekte verschiedener Bildverarbeitungsalgorithmen auf die gewonnenen Bilder studieren. Zuletzt wird eine praktische Anwendung von Bildverarbeitung demonstriert.

Erstellen Sie zunächst im Ordner `D:\PDSV\Gruppen\` ein Verzeichnis für Ihre Gruppe. Legen Sie alle Dateien, die Sie im Laufe des Versuchs erstellen, nur in diesem Ordner ab!

Im Ordner `D:\PDSV\Versuch_8` finden Sie alle vorgefertigten VIs, die Sie für die einzelnen Aufgaben benötigen. Wenn Sie an diesen VIs Veränderungen vornehmen, dann überschreiben Sie die VIs in diesem Verzeichnis bitte nicht.

Aufgabe 8.4: Ansteuerung der Versuchshardware aus LabVIEW

- a) Erstellen Sie ein *LabVIEW*-VI, welches die Kamerabeleuchtung ein- oder ausschaltet. Gehen Sie dazu in folgenden Schritten vor:
 1. Starten Sie *LabVIEW*. Wählen Sie im *LabVIEW*-Dialogfenster „Neu...“ und im darauffolgenden Dialog auf „OK“, um ein leeres VI zu erstellen.
 2. Wechseln Sie in das Fenster „Blockdiagramm“ und öffnen Sie die Funktionenpalette (über Rechtsklick, Funktionenpalette).
 3. Fügen Sie folgende VIs ein: VISA: Seriellen Port konfigurieren, VISA: Schreiben, VISA: Schließen und verbinden Sie sie so, dass beim Starten des VIs zunächst die Schnittstelle initialisiert wird, dann ein fester Wert an die Hardware gesendet und die Schnittstelle gleich wieder freigegeben wird. Der Eingang Schreibpuffer des Schreiben-VIs soll mit einer Konstante beschaltet werden (rechte Maustaste → Erstellen → Konstante).
 4. Geben Sie nacheinander verschiedene Konstanten aus Tab. 8.3 ein und starten Sie jeweils Ihr VI, um das gewählte Kommando an die Hardware zu schicken.
- b) Nun soll ein Druckknopf-Steurelement zur Betätigung hinzugefügt werden (Knopf gedrückt – Beleuchtung ein, Knopf nicht gedrückt – Beleuchtung aus).

1. Duplizieren Sie zunächst die an VISA: Schreiben angeschlossene String-Konstante und stellen Sie eine auf „I“ und die andere auf „L“ ein.
2. Um zwischen den zwei erforderlichen Konstanten umschalten zu können, kann eine CASE-Struktur eingesetzt werden (Funktionen → Programmierung → Strukturen). Am linken Rand dieser Struktur finden Sie den Case-Selektor. Klicken Sie mit der rechten Maustaste auf den **äußeren Anschluss** des Case-Selektors und wählen Sie **Bedienelement erzeugen**.
3. Am oberen Rand der Case-Struktur kann die Entwurfsansicht zwischen den beiden Fällen *TRUE* (Knopf gedrückt) und *FALSE* (Knopf losgelassen) umgeschaltet werden. Verschieben Sie die beiden Konstanten in die Case-Struktur unter den entsprechenden Fällen.
4. Verbinden Sie anschließend zunächst eine der beiden Konstanten mit dem **Schreibpuffer-Eingang**. Dadurch entsteht am Rand der Case-Struktur ein sog. **Ausgangstunnel**, der mit der anderen Konstanten zu verbinden ist.
5. Damit das VI kontinuierlich durchläuft, ist eine *While-Schleife* entsprechend einzufügen.
6. Dokumentieren Sie Ihr funktionierendes Blockdiagramm per Screenshot im Versuchsprotokoll.

Im weiteren Verlauf können Sie das vordefinierte VI `4_Steuerung.vi` benutzen, um den Versuchsaufbau manuell zu steuern.

Aufgabe 8.5: Kamerabild auf Monitor darstellen

In dieser Aufgabe soll ein LabVIEW-VI erstellt werden, welches das Livebild von der Kamera empfängt und auf dem Bildschirm anzeigt.

- a) Erstellen Sie ein neues, leeres VI.
- b) Fügen Sie folgende Sub-VIs ein:
 - IMAQ Create
 - IMAQ Dispose
 - IMAQdx Open.vi
 - IMAQdx Close.vi
 - IMAQdx Configure Grab.vi
 - IMAQdx Grab.vi.
- c) Wechseln Sie in das Frontpanel und fügen ein Anzeigeelement für das Bild (Image Display) ein.
- d) Verwenden Sie hier ebenfalls eine While-Schleife. Überlegen Sie sich, welcher Block sich in der Schleife befinden darf. Achten Sie darauf, dass insbesondere die Configure Grab- und IMAQ Create-Blöcke auf keinen Fall in der Schleife enthalten sein dürfen.

- e) Zum Beenden des VIs ist unbedingt der Stop-Knopf der While-Schleife zu benutzen und nicht der Stop-Knopf in der *LabVIEW*-Menüleiste, damit die belegten Systemressourcen wieder freigegeben werden. Dies gilt auch für die in den folgenden Aufgaben eingesetzten vordefinierten VIs!
- f) Dokumentieren Sie auch dieses Blockdiagramm im Versuchsprotokoll.

Aufgabe 8.6: Glättungsfilter

Öffnen Sie das VI *6_Glättungsfilter.vi* und starten Sie es. Sie sehen links oben das Live-Kamerabild, rechts daneben das Bild mit künstlichem Rauschen überlagert (weißes Rauschen – Mittelwert und Varianz sind einstellbar – oder binäres Rauschen – die Dichte ist einstellbar zwischen 0 % und 100 %) und unten rechts nach Filterung mit dem ausgewählten Glättungsfilter.

Es kann entweder ein lineares Glättungsfilter oder ein Rangordnungsfilter eingesetzt werden. Beim linearen Filter ist der Filterkern einzugeben (bis zur Größe 7×7 , ohne Vorfaktor, kleinere Filterkerne sind zentriert in der Matrix einzutragen); beim Rangordnungsfilter können die Größe der Filtermaske sowie die gewünschte Rangordnung eingestellt werden.

Legen Sie z. B. eine der Platinen oder auch ein anderes Objekt unter die Kamera und stellen Sie **vorsichtig** den Fokus ein.

- a) Überlagern Sie das Bild mit weißem Rauschen. Welche Filtertypen würden Sie zur Reduktion dieses Rauschens verwenden?
Beobachten Sie die Veränderungen der Filterwirkung bei verschiedenen Werten für Mittelwert und Standardabweichung (Varianz) des weißen Gauß'schen Rauschens. Speichern Sie verschiedene charakteristische Bilder für Ihre Dokumentationen. Verwenden Sie u. a. das 5×5 -Binomialfilter aus Aufgabe 8.2 d).
- b) Beschreiben Sie die Wirkung des Mittelwert-Parameters (beim weißen Rauschen) auf das Rauschsignal und das verrauschte Bild.
Hinweis: das Rauschsignal und die Bilder werden intern als vorzeichenlose 8-Bit-Grauwerte dargestellt.
- c) Überlagern Sie das Bild nun mit binärem Rauschen. Wählen Sie das für diesen Rauschtyp am besten geeignete Filter und stellen Sie dessen Parameter ein. Beobachten Sie die Filterwirkung bei verschiedenen Rauschdichten und Filtergrößen. Vergleichen Sie auch die Filterwirkung des ungeeigneteren Filtertyps. Dokumentieren Sie charakteristische Bilder und geben Sie jeweils die verwendeten Filterparameter an.

Aufgabe 8.7: Kantendetektion

In dieser Aufgabe soll das VI 7_Kantendetektion.vi verwendet werden. Im linken oberen Fenster sehen Sie wie in der vorangegangenen Aufgabe das Originalbild. Dabei handelt es sich entweder um das Livebild der Kamera – wenn der entsprechende Button gedrückt ist – oder um ein Standbild, das aus einer *.bmp-Datei geladen wird. Dieses Bild wird durch die folgende Signalverarbeitungskette weiterverarbeitet:

1. N -maliges Glätten mit dem LabVIEW-Tiefpassfilter (IMAQ LowPass).
Der Parameter N kann im Eingabefeld Anzahl Vorglättungen eingestellt werden. Das geglättete Bild kann im rechten oberen Fenster dargestellt werden, indem das Dropdownfeld darüber entsprechend eingestellt wird.
2. Kantendetektion mit einem linearen Filter (Eingabe wie in Aufgabe 8.6) oder einem der in LabVIEW vordefinierten nichtlinearen Kantenfilter (siehe Abschn. 8.2.4.6).
Bei Einsatz eines linearen Filters können negative „Grauwerte“ entstehen. Durch den Button Absolutbetrag können Sie wählen, ob negative Werte zu null gesetzt werden sollen (schwarze Pixel) oder ob ihr Absolutbetrag als Grauwert dargestellt werden soll. Das hiermit gefilterte Bild kann ebenfalls im rechten oberen Fenster dargestellt werden.
3. Umwandlung in ein Binärbild mittels Schwellwertoperator.
Der globale Schwellwert kann im entsprechenden Eingabefeld frei gewählt werden. Das Binärbild wird im rechten unteren Fenster angezeigt.

Öffnen Sie das VI 7_Kantendetektion.vi und starten Sie es.

- a) Laden Sie zunächst das Standbild Oktagon.bmp. Verwenden Sie zunächst **lineare** Prewitt- und Sobel-Filter, die Sie selbst eingeben müssen, mit verschiedenen Wirkungsrichtungen und beobachten Sie den Einfluss einer Vorglättung sowie der Absolutbetrag-Einstellung. Dokumentieren Sie charakteristische Ergebnisbilder und die verwendeten Kerne. Wählen Sie den Schwellwert jeweils so, dass die Filterwirkung gut veranschaulicht wird.
- b) Verwenden Sie nun einen 3×3 -Laplace-Filterkern. Beobachten Sie ebenfalls den Einfluss der Vorglättung und des Absolutbetrags. Warum ergibt sich – bei Einsatz einmaliger Vorglättung – eine doppelte Kontur, wenn die Absolutbetrag-Anzeige aktiviert ist bzw. ohne Absolutbetrag eine einfache Kontur?
- c) Untersuchen Sie schließlich die Wirkungen der nichtlinearen Kantenfilter (Registerkarte IMAQ EdgeDetection). Welche dieser Filter sind isotrop, welche nicht?
Hinweis: Stellen Sie die Anzahl der Vorglättungen auf „0“.
- d) Schalten Sie nun die Aufnahme des Livebildes ein. Untersuchen Sie wiederum die Wirkung unterschiedlicher **linearer** Kantenfilterkerne mit und ohne Absolutbetrag-Darstellung und Vorglättung.
- e) Betrachten Sie nun die nichtlinearen Kantenfilter. Wo liegen die Unterschiede zu den linearen Filtern aus Teilaufgabe d)?

- f) Untersuchen Sie auch den Einfluss der Kamera-Beleuchtung auf die Kantendetektion mit den unterschiedlichen Filtern.

Aufgabe 8.8: Korrelation

Öffnen Sie das VI 8_Korrelation.vi und starten Sie es.

- a) Laden Sie in das linke obere Fenster die Datei Oktagon.bmp und in das linke untere Fenster als Template die Datei 11x11-weiss.bmp. Lassen Sie die Korrelation der beiden Bilder berechnen. Beschreiben Sie das Ergebnisbild und begründen Sie es. *Hinweis: Beachten Sie die Effekte an den Objektkanten. Stellen Sie den Schwellwert auf „0“ ein.*
- b) Ermittlung der Position eines Objektes mittels Template-Matching.
1. Laden Sie die Bilddatei Platine.bmp sowie das Template IC.bmp und lassen Sie die Korrelation durchführen.
 2. Ermitteln Sie aus dem Korrelationsergebnis die Koordinaten des IC-Mittelpunktes im Originalbild der Platine. *Hinweis: Benutzen Sie die Einstellmöglichkeit eines Schwellwertes im Korrelationsbild. In der Statuszeile unterhalb dieses Bildes können Sie den Grauwert und die x - und y -Koordinate des unter dem Mauszeiger befindlichen Pixels ablesen. Es wird eine normalisierte Korrelation berechnet, die Ausgangswerte im Intervall $[0, 255]$ erzeugt.*

Aufgabe 8.9: Template-Matching

Diese Aufgabe soll Ihnen die Möglichkeiten des Template-Matchings aufzeigen. Im VI IMAQ Find Pattern 2 ist ein leistungsfähiger Algorithmus implementiert, der über eine reine Kreuzkorrelation hinausgeht [Nat03]: Durch verschiedene Verfahren wie z. B. Kantendetektion werden weitergehende Informationen über die *Struktur* des Bildes und des Templates gewonnen, wodurch der Rechenaufwand reduziert und die Erkennungsleistung erhöht werden können (insbesondere bei veränderlichen Beleuchtungsbedingungen oder auch bei Skalierung und Rotation des Templates).

Ein Anwendungsbeispiel ist der Einsatz in der visuellen Qualitätssicherung in der Industrie. Hierbei werden die zu prüfenden Produkte mit einer Kamera erfasst und durch ein Bildverarbeitungssystem auf Einhaltung der Vorgaben untersucht, um Ausschuss aussortieren zu können.

Ziel dieser Aufgabe ist ein Bildverarbeitungssystem, welches bestückte Platinen automatisiert auf fehlende Bauteile untersucht. Am Ende werden dann in Aufgabenteil e) die zu untersuchenden Platinen nacheinander manuell auf den Drehteller gelegt. Der Rechner analysiert kontinuierlich das von der Kamera gelieferte Bild und stoppt den Drehteller, sobald eine korrekt bestückte Platine erkannt wird. Diese soll dann wiederum manuell entfernt und der Drehteller wieder gestartet werden.

- a) Öffnen Sie das VI 9a_TemplateMatching.vi. Wechseln Sie zum Blockdiagramm und analysieren Sie das VI. Berücksichtigen Sie bei der Event-Struktur (gelb-grau gestreifter Rahmen) lediglich den Case Timeout.
Starten Sie nun das VI und legen Sie eine der Platinen unter die Kamera. Sie können mit der Maus einen Bildausschnitt auswählen (z. B. ein IC) und diesen durch Get Template als Template festlegen. Mit dem Schalter Do Template Matching wird die kontinuierliche Template-Suche gestartet und es werden gefundene Muster im Bild mit einem roten Rechteck umrahmt. Vor dem Beenden des VIs empfiehlt es sich, das Template-Matching wieder zu deaktivieren.
- b) Fügen Sie ein numerisches Anzeigeelement hinzu, welches laufend die Anzahl der aktuell im Bild gefundenen Muster ausgibt. Dokumentieren Sie das Blockdiagramm.
- c) In welchem Fall werden Bauteile zuverlässiger gefunden: Wenn nur das Bauteil selbst ausgewählt wird oder wenn das Template noch einen Bereich um das Bauteil herum enthält?
- d) Untersuchen Sie den Einfluss der Lage eines dem Template entsprechenden Objektes (Rotation) auf das Suchergebnis.
- e) Laden Sie nun das VI 9b_QS.vi und starten Sie es. Sie können ebenfalls wie im vorangegangenen Aufgabenteil das Template festlegen. Jedoch ist die Suche hier anders implementiert: Es wird zusätzlich eine Kantenantastung durchgeführt, um die Lage der Kanten der Platine zu ermitteln (die Kanten werden innerhalb der beiden grünen Rechtecke gesucht). Das Template-Matching wird dann nur in einem kleinen Bereich relativ zu dem so festgelegten Koordinatensystem durchgeführt. Auf diese Weise kann das System nicht nur ermitteln, ob das Template im Bild vorhanden ist, sondern auch, ob es sich an der richtigen Position befindet. Untersuchen Sie, wie zuverlässig die oben beschriebene Prüfaufgabe mit diesem System durchgeführt werden kann.

8.7 Literatur

- [Bas04] BASLER VISION TECHNOLOGIES: *User's Manual Basler A600f*, 2004.
- [BPF16] JÜRGEN BEYERER, FERNANDO PUENTE LEÓN und CHRISTIAN FRESE: *Automatische Sichtprüfung – Grundlagen, Methoden und Praxis der Bildgewinnung und Bildauswertung*. Springer, Berlin Heidelberg, 2. Auflage, 2016.
- [Bri95] E. ORAN BRIGHAM: *FFT – Schnelle Fourier-Transformation*. Oldenbourg, München, 6. Auflage, 1995.
- [Jä12] BERND JÄHNE: *Digitale Bildverarbeitung*. Springer, Berlin Heidelberg, 7. Auflage, 2012.
- [Mer10] GERHARD MERZIGER (Herausgeber): *Formeln + Hilfen zur höheren Mathematik*. Bino-mi, Barsinghausen, 6. Auflage, 2010.

- [Nat03] NATIONAL INSTRUMENTS: *IMAQ Vision Concepts Manual*, 2003.
- [PJ15] FERNANDO PUENTE LEÓN und HOLGER JÄKEL: *Signale und Systeme*. De Gruyter Oldenbourg, Berlin, 6. Auflage, 2015.

Index

A

A-posteriori-Dichte 205
 A-priori-Dichte 204
 Abtastfrequenz 101
 Abtasttheorem 101
 Abtastung 69
 Abtastzeit 101
 AKF *siehe* Autokorrelationsfunktion
 Aliasing 70, 101, *siehe*
 Bandüberlappungsfehler
 Ambiguitätsfunktion 162
 Amplitudendichte 73
 Amplitudengang 101
 analytisches Signal 166
 Anti-Aliasing-Filter 70
 Ausgangssignal 100
 Autokorrelationsfunktion 45
 Autokovarianzfunktion 46
 Autoleistungsdichtespektrum 50

B

Bandbreite 145
 Bandpass 106
 Bandsperre 106
 Bandüberlappungsfehler 147, 149
 Bartlett-Fenster 81
 Beobachtungsfrequenz 79
 Beobachtungszeit 79
 Bessel-Filter 71
 Betragsbildung 275
 Bildaufnahme 265, 293, 295
 Bildauswertung 266
 Bildgewinnung 266
 Bildgradient 283
 Bildsegmentierung 266, 275
 Bildverarbeitung 266
 Bildvorverarbeitung 266, 273
 Binärbild 275
 Binomialfilter 279
 biorthogonales Wavelet 158
 Blackman-Fenster 152
 Bode-Diagramm 238
 Butterworth-Filter 71, 108

C

Cauer-Filter 71, 110
 CCD-Sensor 266
 Chirp-Signal 239
 Clipping 275
 CMOS-Sensor 266
 Cohen-Klasse 168
 Coiflet 159
 Compact Disc 99

D

Dämpfung 227
 Dämpfungsmatrix 236
 Daubechies-Wavelet 158
 DFT *siehe* diskrete Fourier-Transformation
 zweidimensionale ~ 271
 Digitalisierung 69
 diskrete Faltung 272
 diskrete Fourier-Transformation .. 79, 147, 271
 Dithering 77
 Dithersignal 77
 Dolph-Tschebyscheff-Fenster 152
 Dreieckfenster 81
 Durchlassbereich 106
 Durchlassfrequenz 106

E

Eigenfrequenz 227
 Eigenschwingung 232
 Eigenschwingungsanalyse 227
 Eingangssignal 100
 elliptisches Filter 71
 Energie 144
 Energiedichte 164
 normierte ~ 144
 spektrale ~ 144
 zeitliche ~ 144
 Energiedichtespektrum 50, 164
 Energiesignal 48
 Ergodizität 47
 Erwartungstreue 51
 Express-VI 18
 Extremwert 282

F

Faltung	
diskrete ~	272
zweidimensionale ~	272
zyklische ~	273
Faltungskern	277
Farbkamera	267
Feder-Masse-Dämpfer-System	228
Filterentwurf	106
Filterkovarianzmatrix	206
Filtermaske	276
Filterschätzwert	205
Filterschritt	204
Formanten	171
Fourier-Transformation	78, 146, 270
~ zeitdiskreter Signale	79
diskrete ~	79
schnelle ~	80
Frame-Grabber	270
Frequenzauflösung	79, 271
Frequenzgang	101, 237
Frequenzindex	271
frequenzselektives Filter	106
FRF	<i>siehe</i> Frequenzgang
Funktion	
Matlab~	10

G

Gabor-Transformation	153
Gauß-Impuls	146
GCC	<i>siehe</i> verallgemeinerte Kreuzkorrelation
Gibbs'sches Phänomen	113
Gierrate	189
Gierwinkel	197
Global Positioning System	189
Grauwertskalierung	275
Grenzfrequenz	112
Gruppenlaufzeit	101

H

Hamming-Fenster	82, 151
Hann-Fenster	82, 151
Hardware in the Loop	99
Helligkeitsdynamik	268
Hilbert-Transformation	166
Hochpass	106

I

Identifikation	237
nichtparametrische ~	237
parametrische ~	237
Impulsantwort	100
Impulsrauschen	280
Interferenzterm	166
Isotropie	274, 278

J

Jacobi'sche Ellipsenfunktion	110
Jitter	73

K

Kaiser-Fenster	152
Kalman-Filter	187, 188, 200, 203
Kalman-Filterschritt	204
Kalman-Prädiktionsschritt	204
Kalman-Verstärkungsmatrix	205
Kante	282
Kantendetektor	
regularisierter ~	284
KKF	<i>siehe</i> Kreuzkorrelationsfunktion
KKV	<i>siehe</i> Kreuzkovarianzfunktion
Klasse	
Matlab~	10
Knickfrequenz	112
Kompression	270
Konsistenz	51
Kreuzkorrelation	173
Kreuzkorrelationsfunktion	46
Kreuzkorrelationskoeffizient	55
Kreuzkovarianzfunktion	46
Kreuzleistungsdichtespektrum	50
Kreuzterm	166
Kronecker-Delta	230
Kurzzeit-Fourier-Transformation	153
Kurzzeitkorrelationsfunktion	51
Kurzzeitspektrum	154

L

Laplace-Operator	286
Laufzeit	172, 173
Leckeffekt	80, 81, 149, 150
Leistung	50
Leistungsdichte	50

Leistungsdichtespektrum 80
 Leistungssignal 48
 lineare Skalierung 275
 lineares Quantisierungsmodell 76
 Linksdivision 9
 lokaler Operator 273
 LTI-System 100

M

Median 280
 Medianfilter 280
 Mittelwert 45
 mittlere Frequenz 145
 mittlere Zeit 144
 Modalanalyse 227
 Modaltransformation 229, 230, 235
 Mode 227
 Modenform 227
 Moment 45
 Moving-Average-Filter 93

N

normierte Energiedichte 144
 Nulldurchgang 282

O

Ortsbereich 270
 Ortsfrequenzbereich 270

P

Periodogramm 80
 Phasengang 101
 Pixel 266
 Polaritätskorrelation 55
 Prädiktionsschritt 204
 Prädiktionskovarianzmatrix 205
 Prädiktionschätzwert 205
 Prewitt-Operator 284
 Punktoperation
 Matlab~ 9
 Punktoperator 273

Q

Quantisierung 69, 73
 Quantisierungsfehler 73
 Quantisierungsmodell

 lineares ~ 76
 Quantisierungstheorem 75

R

Randeffekt 54, 273, 276
 Rapid Control Prototyping 99
 Raumimpulsantwort 179
 Rechteckfenster 72
 Rechteckfilter 278
 Rechtsdivision 9
 Roberts-Operator 287

S

schnelle Fourier-Transformation 80
 Schwellwertoperator 275
 SCPI 84
 Segmentierung 266, 287
 Signal
 Energie~ 48
 Leistungs~ 48
 sonstiges ~ 48
 Skalierung 144
 Skalierungsfunktion 161
 Skalogramm 182
 Skript
 Matlab~ 10
 Sobel-Operator 284
 spektrale Energiedichte 144
 Spektrogramm 154
 Sperrbereich 106
 Sperrfrequenz 106
 Stationarität 47
 STFT . . siehe Kurzzeit-Fourier-Transformation
 Superpositionsprinzip 274
 Symlet 159
 Systemfunktion 100

T

temporäre AKF des Spektrums 163
 temporäre AKF des Zeitsignals 163
 Tiefpass 106
 Toleranzschema 106
 transientes Signal 240
 Tschebyscheff-Filter 71, 109
 Tschebyscheff-Polynom 109

U

Übergangsbereich	106
Übertragungsfunktion	100
~ des Binomialfilters	280
~ des Rechteckfilters	278
DFT~	278, 280
Unschärfeprinzip	154

V

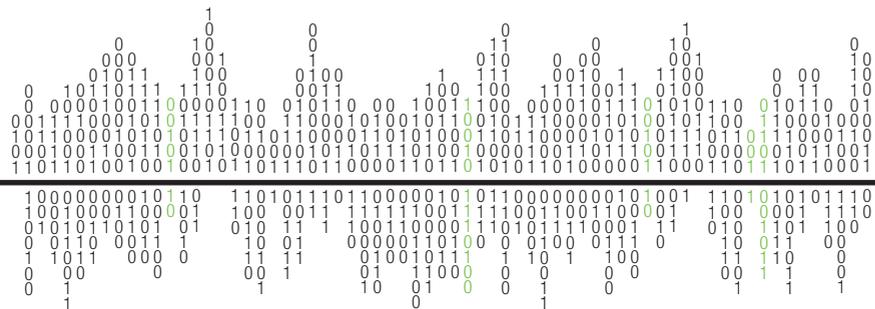
verallgemeinerte Kreuzkorrelation	174
Verschiebungsfreiheit	274, 278
Verschiebungsinvarianz	274
Verstärkung	135, 178, 268
Verstärkungsmatrix	202, 205
VI	16
VISA	289

W

Wavelet-Funktion	161
Wavelet-Transformation	156
Wellenzahl	271
Wigner-Ville-Verteilung	162
zeitdiskrete ~	168

Z

Zeitdauer	145
Zeitdauer-Bandbreite-Produkt	145
Zeitinvarianz	274
zeitliche Energiedichte	144
Zero-Padding	54
Zwei-Kanal-Filterbank	161
zyklische Korrelationsfunktion	54



ISBN 978-3-7315-0651-5



9 783731 506515 >

Dieses Lehrbuch richtet sich an Studenten der Ingenieurwissenschaften und der Informatik. Die Grundlagen der digitalen Signalverarbeitung werden im Rahmen von acht Versuchen vorgestellt. Als Experimentierwerkzeuge dienen hauptsächlich die etablierten Programmierumgebungen MATLAB/Simulink und LabVIEW. Die Bandbreite der Anwendungen reicht von einem Versuchsfahrzeug zur Sensordatenfusion über die Schwingungsanalyse von Gegenständen bis hin zu Akustik und Sprachsignalverarbeitung sowie Bildverarbeitung. Die vermittelten Inhalte sollen die Leser mit der praktischen Anwendung von modernen Methoden der Signalverarbeitung vertraut machen und ihnen hilfreiche Werkzeuge an die Hand geben, deren Beherrschung im Berufsalltag oftmals verlangt wird.

Gedruckt auf FSC-zertifiziertem Papier