

Versuchsbeschreibungen
Grundlagenlabor Nachrichtentechnik
– WS 2020/2021 –

Dr. Dirk Wübben

NW1, Raum N2300, Tel.: 0421/218-62385

E-mail: wuebben@ant.uni-bremen.de



Universität Bremen, FB1
Institut für Hochfrequenz- und Nachrichtentechnik
Arbeitsbereich Nachrichtentechnik
Prof. Dr.-Ing. A. Dekorsy
Postfach 33 04 40
D-28334 Bremen

WWW-Server: <http://www.ant.uni-bremen.de>

4. November 2021

Vorbemerkungen

In dem *Grundlagenlabor Nachrichtentechnik* sollen die Inhalte der Vorlesung *Grundlagen der Nachrichtentechnik* anhand praktischer Beispiele, Übungen und Messungen vertieft werden. Begleitend zu den Vorlesungen werden hierzu insgesamt sechs Laborversuche durchgeführt, welche aufeinander aufbauen.

Zur Vorbereitung auf den jeweiligen Versuch ist das entsprechende Kapitel dieser Versuchsbeschreibung zu bearbeiten und der zugehörige Stoff der Vorlesung zu wiederholen. Verweise auf theoretische Grundlagen beziehen dabei auf die Folien zur Vorlesung „Grundlagen der Nachrichtentechnik“ (<http://www.ant.uni-bremen.de/courses/gnt/>) bzw. auf das Lehrbuch „*Nachrichtenübertragung*“ von K.D. Kammeyer, A. Dekorsy, 6. Auflage 2017, Springer Vieweg Verlag, 2017. Zu jedem Versuch sind Vorbereitungsaufgaben angegeben. Diese sind vor der Versuchsdurchführung von jedem Teilnehmer so vorzubereiten, dass diese im Plenum / Online-Meeting erläutert werden können.

Im Gegensatz zu der bisherigen Durchführung des Labors findet zu Beginn des Labors kein Kolloquium mehr statt und ebenso entfällt die Ausarbeitung eines Protokolls. Im Vordergrund steht vielmehr die praktische Anwendung der theoretischen Grundlagen und die selbständige Lösung von Aufgaben und Problemen. Dies setzt jedoch die gute Vorbereitung voraus, da nur so eine sinnvolle Durchführung der Versuche möglich ist. Im Rahmen des Praktikums sollen dann die unterschiedlichen Strategien zur Problemlösung vorgestellt und diskutiert werden.

Ein Schwerpunkt des Grundlagenlabors stellt das Programm Matlab (MATrix-LABoratory) dar. Es ist ein sehr vielseitiges und doch einfach zu erlernendes Werkzeug zur Erstellung mathematischer Berechnungen. Es zeichnet sich insbesondere durch eine sehr einfache Programm-Syntax und einer Vielzahl an Möglichkeiten zur grafischen Darstellung von Daten aus. Eine Alternative stellt Octave da, welches eine OpenSource Software mit Matlab-Syntax ist. Im Rahmen des Labors sind alle zur Verfügung gestellten Skripte ebenfalls in Octave ausführbar.

Im Rahmen der Vorbesprechung erfolgt eine Einführung in die wesentlichen Grundlagen von Matlab. Eine Einleitung zum Programmaufruf unter Linux, eine Übersicht zu den wichtigsten Befehlen und eine Auflistung von weiterführenden Einführungen finden sich in dem Anhang dieses Skriptes.

Kann ein Versuch wegen Krankheit nicht durchgeführt werden, so ist schnellstmöglich der entsprechende Betreuer zu kontaktieren und nach Möglichkeit der Versuch nachzuholen. Bei mehrfachem Fehlen oder wiederholt unzureichender Vorbereitung ist das gesamte Praktikum im nächsten Jahr zu wiederholen.

Inhaltsverzeichnis

1	Signale	1
1.1	Einführung und Vorbereitung	1
1.1.1	Simulation kontinuierlicher Signale auf einem PC	1
1.1.2	Zeitdiskrete Signale	2
1.1.3	Vorbereitungsaufgabe	4
1.2	Versuchsdurchführung	5
1.2.1	Erzeugung von Signalen	5
1.2.2	Der Fourier-Transformation	5
1.2.3	Zeitdiskrete Signale	6
2	Lineare Systeme	9
2.1	Einführung	9
2.2	Vorbereitung	9
2.3	Hilbert-Transformation	10
2.3.1	Vorbereitungsaufgabe	10
2.4	Analytische Signale	11
2.4.1	AUFGABE	11
2.5	Demodulation und Tiefpass-Filterung	12
2.6	Quadraturmischer	12
2.7	Äquivalentes Basisbandmodell	13
2.7.1	Direkte Übertragung	13
2.7.2	Äquivalentes Kanalmodell	14
3	Diskrete Signale und Systeme	16
3.1	Motivation	16
3.2	Elementare zeitdiskrete Signale	17

3.3	Transformationen zeitdiskreter Signale	17
3.3.1	Zeitdiskrete Fourier-Transformation	17
3.3.2	z-Transformation	17
3.4	Zeitdiskrete Systeme	19
3.4.1	Eigenschaften	19
3.4.2	Frequenzgang	21
3.4.3	Systemfunktion und Differenzgleichung	22
3.4.4	Übersicht	23
3.5	Zeitdiskrete Zufallsprozesse	24
3.5.1	Zufallsvariable, Verteilungsfunktion, Verteilungsdichtefunktion	24
3.5.2	Statistische Kenngrößen (Momente von Verteilungen)	26
3.5.3	Statistische Unabhängigkeit, Summation und zentraler Grenzwertsatz	27
3.5.4	Zufallsprozesse und ihre Kenngrößen	28
3.5.5	Spektrale Darstellung zeitdiskreter stochastischer Prozesse	31
3.6	Zeitdiskrete Zufallsprozesse und zeitdiskrete LTI-Systeme	31
3.6.1	Korrelationsfolgen und Leistungsdichtespektren des Ausgangsprozesses	31
3.6.2	Systemidentifikation	32
3.7	Vorbereitungsaufgaben	32
3.7.1	Vorbemerkungen	32
3.8	Versuchsdurchführung	33
4	Codierung	41
4.1	Einführung	41
4.1.1	Theoretische Grundlagen	41
4.2	Vorbereitungsaufgaben	45
4.2.1	Quellencodierung	45
4.2.2	Fehlerkorrektur	46
4.3	Versuchsdurchführung	46
4.3.1	Kanalcodierung	46
4.3.2	Decodierung und Fehlerkorrektur	47
4.3.3	Bitfehlerrate über BSC	47
4.3.4	Bitfehlerrate mit BPSK über AWGN Kanal	48
4.3.5	Bitfehlerrate mit QPSK über AWGN Kanal	48

A	Einleitung zu Matlab	50
B	Übersicht zu den Befehlen	52

Kapitel 1

Signale

Niklas Bulk

BITZ, Raum 0.16, Tel.: 0421/ 2208-115

E-Mail: bulk@ant.uni-bremen.de

1.1 Einführung und Vorbereitung

In diesem Versuch soll ein Überblick über die in der Vorlesung *Grundlagen der Nachrichtentechnik* verwendeten Elementarsignale im Zeit- und Frequenzbereich gegeben werden. Weiterhin sollen die Korrespondenzen der Fouriertransformation und die Verarbeitung kontinuierlicher Signale auf einem Digitalrechner erlernt werden.

Für die Durchführung dieses Versuchs wird die Kenntnis des in den Folien zur Vorlesung *Grundlagen der Nachrichtentechnik* behandelten Stoffes vorausgesetzt, die unter [Stud.IP](#) heruntergeladen werden können.

1.1.1 Simulation kontinuierlicher Signale auf einem PC

Bei Verwendung eines digitalen Systems, wie z.B. eines PC stellt sich zwangsläufig das Problem der Darstellung und Verarbeitung von kontinuierlichen Signalen im Zeit- oder Frequenzbereich. Daher soll im Folgenden kurz auf die Signalverarbeitung unter MATLAB/Octave eingegangen werden.

Bild 1.1 soll die Verarbeitung eines zeitkontinuierlichen Signals in MATLAB/Octave mittels Annäherung durch eine Rechteck-Treppenfunktion verdeutlichen. Der zeitkontinuierliche Sinus $s(t) = \sin(2 \cdot \pi \cdot 0.8\text{kHz} \cdot t)$ für $0 \leq t < 5\text{ms}$ geht also bei einer Abtastung mit 200 Werten pro Millisekunde (d.h. das Abtastintervall ΔT beträgt $1/200$ Millisekunden) über in die Folge $s = [0, 0.0251, 0.0502, 0.0753, 0.1004, \dots]^T$. Die ersten 20 Werte dieser Folge sind in Bild 1.1 (rechts) abzulesen. Es sei an dieser Stelle darauf hingewiesen, dass die Grundeinheiten im Weiteren in Millisekunden bzw. kHz angegeben werden. Sollte davon abgewichen werden, wird an gegebener Stelle gesondert darauf hingewiesen.

Für die Darstellung 'quasikontinuierlicher' Zeitsignale muss ein Zeitvektor (z.B. $t = 0 : \Delta T : t_{max} - \Delta T$ bzw. $\mathbf{t} = 0 : 1/200 : 5 - 1/200$; für einen Vektor der Länge 5 ms) mit sehr kleinem Abtastintervall erzeugt werden.

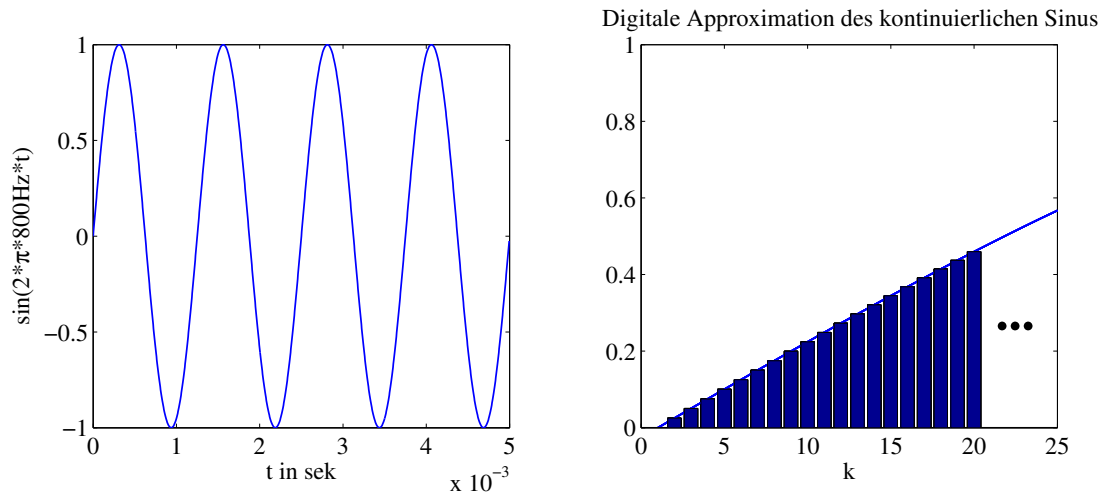


Abbildung 1.1: Abtastung eines kontinuierlichen Zeitsignals mit Stützstellen der Breite ΔT

1.1.2 Zeitdiskrete Signale

Elementare zeitdiskrete Signale

Zeitdiskrete Signale entstehen durch das Abtasten von zeitkontinuierlichen Signalen. Die Abtastung kann dabei zu äquidistanten oder auch zufälligen Zeitpunkten erfolgen. Für die hier betrachteten Signale wird grundsätzlich von einer äquidistanten Abtastung ausgegangen. Im weiteren wird angenommen, dass aus dem zeitdiskreten Signal das kontinuierliche Signal wieder eindeutig rekonstruierbar ist, d.h. dass das Abtasttheorem eingehalten wird.

Mathematisch werden zeitdiskrete Signale als Folgen von Zahlen dargestellt, wobei als Schreibweise

$$x_K(t)|_{t=kT_A} = x_K(kT_A) =: x(k) \quad (1.1)$$

verwendet wird. Dabei bezeichnet $x_K(t)$ das kontinuierliche Signal, $x(k)$ die Zahlenfolge, $T_A = 1/f_A =: T$ das Abtastintervall, f_A die Abtastfrequenz und k eine ganze Zahl im Bereich $-\infty < k < \infty$. Das Signal $x_K(t)$ bzw. die Zahlenfolge $x(k)$ kann dabei sowohl reellwertig als auch komplexwertig sein.

Bei der Besprechung der Theorie von zeitdiskreten Systemen und Signalen spielen einige grundlegende Zahlenfolgen eine wichtige Rolle.

Man unterscheidet u.a.

- die Impulsfolge

$$x(k) = \delta(k) := \begin{cases} 1 & k = 0 \\ 0 & k \neq 0, \end{cases} \quad (1.2)$$

- die Sprungfolge

$$x(k) = \varepsilon(k) := \begin{cases} 1 & k \geq 0 \\ 0 & k < 0, \end{cases} \quad (1.3)$$

- und die komplexe Exponentialfolge

$$x(k) = e^{j\omega T k} = e^{j\Omega k}. \quad (1.4)$$

Dabei beschreibt Ω die auf die Abtastfrequenz $f_A = 1/T$ normierte Kreisfrequenz, für die

$$\Omega = \frac{\omega}{f_A} = \omega T \quad (1.5)$$

gilt.

Weiterhin kann unter Verwendung der Ausblendeigenschaft der Impulsfolge $\delta(k)$ nach Gl.(1.2) jede Folge durch

$$x(k) = \sum_{i=-\infty}^{\infty} x(i) \cdot \delta(k-i) \quad (1.6)$$

beschrieben werden. Hierbei ist zu beachten, dass $x(i)$ einen einzigen Amplitudenwert und $x(k)$ das gesamte Zeitsignal beschreibt.

Zeitdiskrete Fourier-Transformation

Analog zur Fourier-Transformation zeitkontinuierlicher Signale kann auch eine Fourier-Transformation von Zahlenfolgen angegeben werden. Die so genannte zeitdiskrete Fourier-Transformation einer Folge $x(k)$ und die zugehörige Rücktransformation sind definiert als:

$$\begin{aligned} X(e^{j\Omega}) &:= \sum_{k=-\infty}^{\infty} x(k) e^{-j\Omega k} \\ x(k) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\Omega}) e^{j\Omega k} d\Omega. \end{aligned} \quad (1.7)$$

Unter Verwendung der Definitionen Gl.(1.7) können nun folgende elementare Aussagen getroffen werden:

1. $X(e^{j\Omega})$ ist eine frequenzkontinuierliche Funktion, also keine Zahlenfolge
2. Da die Eigenschaft

$$e^{-j(\Omega \pm 2\pi)k} = e^{-j\Omega k} \quad (1.8)$$

gilt und k ganzzahlig ist, ist die Fourier-Transformierte $X(e^{j\Omega})$ periodisch in 2π .

Für viele Anwendungen ist entscheidend, dass aus dem zeitdiskreten Signal das ursprüngliche kontinuierliche Signal wieder zurückgewonnen werden kann. Um dies zu ermöglichen muss die Abtastfrequenz so gewählt werden, dass das Abtasttheorem eingehalten wird. Im folgenden soll dieses für die digitale Signalverarbeitung entscheidende Theorem kurz erläutert werden.

Ausgehend von einem zeitkontinuierlichen Signal $x_K(t)$ und den Beziehungen der Fourier-Transformation

$$\begin{aligned} \mathcal{F}\{x_K(t)\} = X_K(j\omega) &= \int_{-\infty}^{\infty} x_K(t) e^{-j\omega t} dt \\ \mathcal{F}^{-1}\{X_K(j\omega)\} = x_K(t) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} X_K(j\omega) e^{j\omega t} d\omega \end{aligned} \quad (1.9)$$

erhält man durch Abtastung mit der Abtastfrequenz f_A die Zahlenfolge $x(k)$ und das durch Gl.(1.7) gegebene Transformationspaar. Es kann nun gezeigt werden dass die Spektren $X_K(j\omega)$ und $X(e^{j\Omega})$ durch die Beziehung

$$X(e^{j\Omega}) = \frac{1}{T} \sum_{i=-\infty}^{\infty} X_K(e^{j(\Omega+i2\pi)/T}) \quad (1.10)$$

verknüpft sind. Daraus folgt, dass sich das Spektrum diskreter Signale $X(e^{j\Omega})$ aus der Überlagerung der um 2π verschobenen und auf $1/T$ normierten Spektren des kontinuierlichen Signals ergibt. Ein Beispiel zeigt Bild 1.2.

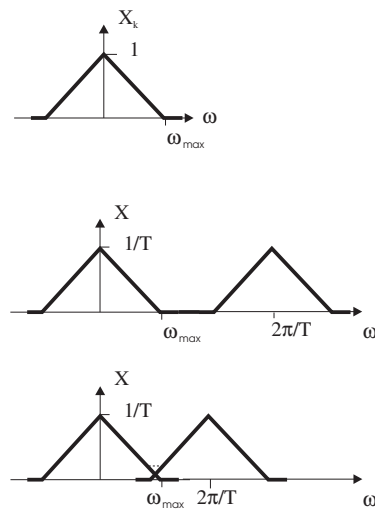


Abbildung 1.2: Spektren eines zeitkontinuierlichen Signals und zweier durch verschiedene Abtastfrequenzen daraus gewonnener zeitdiskreter Signale

Soll nun aus dem zeitdiskreten Signal das zeitkontinuierliche vollständig rekonstruierbar sein, so dürfen sich die Teilspektren $X_K(e^{j(\Omega+i2\pi)/T})$ nicht überlappen was andernfalls aus dem Englischen kommend als *aliasing* bezeichnet wird. Somit ergeben sich folgende Forderungen, die als Abtasttheorem bekannt sind:

- $X_K(j\omega)$ muss bandbegrenzt sein, also im Frequenzbereich $|\omega| \geq \omega_{max}$ identisch verschwinden
- die Abtastfrequenz $\omega_A = 2\pi f_A = 2\pi/T$ muss mindestens doppelt so groß wie die maximale Frequenz ω_{max} von $X_K(j\omega)$ gewählt werden.

Auf das Problem der Rekonstruktion des kontinuierlichen Signals soll im Rahmen dieses Versuchs nicht eingegangen werden. Hierzu sei auf die Literatur verwiesen.

1.1.3 Vorbereitungsaufgabe

Generieren Sie mit MATLAB/Octave ein Sinussignal und den zugehörigen Zeitvektor \mathbf{t} mit einem Abtastintervall $\Delta T = 1/200\text{ms}$. Speichern Sie den geschriebenen Code und ein Bild des generierten Signals in einer PDF-Datei und senden Sie diese per E-Mail an den zuständigen Betreuer. Der erzeugte Sinus soll folgende Anforderungen erfüllen:

- Frequenz 0.8 kHz
- Amplitude 2.5
- Länge 10 ms

Verwenden Sie bei Problemen die MATLAB/Octave-Hilfe (`help`) und das Internet. Sollten sich weitere Probleme ergeben, kontaktieren Sie **vor dem Versuchstermin** den zuständigen Betreuer.

Machen Sie sich weiterhin mit folgenden Befehlen vertraut, indem Sie die Hilfe lesen (`help <Befehl>`), da sie für die Durchführung dieses Versuches benötigt werden:

`help`, `plot`, `subplot`, `xlabel`, `ylabel`, `title`, `size`, `length`, `plot_zeitsignal`¹, `bar_spektrum`¹, `f_trafo`¹, `f_reihe`¹, `faltung`¹.

Für die Durchführung des Versuchsteils „Diskrete Signale“ werden zudem die Funktionen `erste_d`¹, `imp_per`¹, `cos_abt`¹ und `si`¹ benötigt.

1.2 Versuchsdurchführung

1.2.1 Erzeugung von Signalen

Erzeugen Sie

- einen Sinus $s_1(t_1) = \sin(2\pi f t_1)$
- einen Cosinus $s_2(t_1) = \cos(2\pi f t_1)$
- einen quadrierten Sinus $s_3(t_1) = \sin^2(2\pi f t_1)$
- eine Exponentialschwingung $s_4(t_1) = e^{j2\pi \cdot f t_1}$
- eine Rechteckfolge $s_5(t_2) = \begin{cases} 1 & \text{für } -T/2 \leq t \leq T/2 \\ 0 & \text{sonst} \end{cases}$
- einen Impuls $s_6(t_2) = \begin{cases} 1 & \text{für } t_2 = 0 \\ 0 & \text{sonst} \end{cases}$
- einen Si-Impuls $s_7(t_2) = \frac{\sin(2\pi f t_2)}{2\pi f t_2}$
- einen Gauß-Impuls $s_8(t_2) = e^{-|\alpha| t_2^2}$
- ein Multisinus-Signal $s_9(t_1) = 0.25 \cdot \sin(2\pi f_1 t_1) + 0.5 \cdot \sin(2\pi f_2 t_1) + 0.25 \cdot \sin(2\pi f_3 t_1)$ mit $f_1 = 0.8\text{kHz}$, $f_2 = 1.6\text{kHz}$ und $f_3 = 2.4\text{kHz}$
- ein Zufallssignal $s_{10}(t_1) = \text{randn}(\text{size}(t_1))$

Die ansonsten benötigten Parameter lauten: $f = 0.8\text{kHz}$, $0 \leq t_1 < 20\text{ms}$, $-20\text{ms} \leq t_2 < 20\text{ms}$, $T = 4\text{ms}$ und $\alpha = 1$. Dabei soll das Abtastintervall $\Delta T = 1/200\text{ms}$ betragen.

1.2.2 Der Fourier-Transformation

Berechnen Sie die Fourier-Transformierten der in Abschnitt 1.2.1 berechneten Zeitsignale $s_i(f) = \mathcal{F}\{s_i(t)\}$, $i = 1..10$ und stellen Sie sie mit einem geeigneten Befehl grafisch dar (Befehl `plot_zeitsignal` im Zeitbereich und `bar_spektrum` im Frequenzbereich). Benutzen Sie für die Fourier-Transformation periodischer Signale den Befehl `[f,S]=f_reihe(t,s)` und für nichtperiodische Signale den Befehl `[f,S]=f_trafo(t,s)`.

¹Diese Befehle gehören nicht zum Standardbefehlssatz von MATLAB/Octave, sondern sind ANT-spezifisch, d.h., sie sind nicht im MATLAB/Octave-Lieferumfang, sondern müssen vor Benutzung von Stud.IP heruntergeladen werden.

Linearität der Fourier-Transformation

Transformieren Sie die drei überlagerten Sinuskomponenten von $s_9(t_1)$ einzeln in den Frequenzbereich und addieren Sie die Spektren. Ergeben sich Unterschiede zum Spektrum $S_9(f)$?

Konjugiert gerade Spektren reeller Zeitsignale

Welche der dargestellten Spektren weisen eine konjugiert gerade Symmetrie auf, d.h. ihr Realteil hat eine Achsensymmetrie zur y-Achse und ihr Imaginärteil eine Punktsymmetrie zum Nullpunkt. Welche Eigenschaft besitzen all diese Signale im Zeitbereich?

Verzögerung im Zeitbereich und Verschiebung im Frequenzbereich (Modulation)

Multiplizieren Sie das erzeugte Multisinussignal $s_9(t_1)$ mit der Exponentialschwingung $s_4(t_1)$. Dieser Vorgang wird in der Nachrichtentechnik als Modulation bezeichnet. Welche Auswirkungen sind im Spektralbereich zu beobachten. Erhöhen Sie die Frequenz der Exponentialschwingung um einen Faktor 5 und wiederholen Sie die Multiplikation im Zeitbereich. Was beobachten Sie im Frequenzbereich?

Verschieben Sie den erzeugten Gaußimpuls derart, dass sein Mittelpunkt bei 5 ms liegt ($s_8(t_2 - 5ms)$) und stellen Sie ihn im Zeit und Frequenzbereich dar. Welche Auswirkungen sind zu beobachten?

Faltung und Multiplikation

Aus der Vorlesung ist bekannt, dass eine Faltung im Zeitbereich in eine Multiplikation im Frequenzbereich übergeht und umgekehrt. Zur Faltung in MATLAB/Octave steht die Funktion `[ty,y]=faltung(t1,x1,t2,x2)` zur Verfügung.

Falten Sie zwei Rechteckfunktionen (z.B. $s_5(t_2)$) und betrachten Sie das Ergebnis im Zeit- und Frequenzbereich.

Welche Aussage können Sie treffen?

1.2.3 Zeitdiskrete Signale

Impulsfolge, Sprungfolge Aus der Systemtheorievorlesung ist bekannt, dass im zeitkontinuierlichen Fall die Deltafunktion $\delta(t)$ und die Einheitssprungfunktion $\epsilon(t)$ über die Integration bzw. Differentiation zusammenhängen. Es gilt:

$$\delta(t) = \frac{\partial \epsilon(t)}{\partial t} \quad \text{und} \quad \epsilon(t) = \int_{-\infty}^t \delta(\tau) d\tau.$$

Für den hier betrachteten Fall zeitdiskreter Signale geht die Impulsfolge $\delta(k)$ durch die Bildung der ersten Differenz aus der Sprungfolge $\epsilon(k)$ hervor, d.h.

$$\delta(k) = \epsilon(k) - \epsilon(k - 1). \quad (1.11)$$

a) Gegeben ist eine Rechteckimpulsfolge der Länge $L = 6$:

$$x_0(k) = \begin{cases} 1 & 5 \leq k \leq 10 \\ 0 & \text{sonst.} \end{cases} \quad (1.12)$$

Berechnen Sie die erste Differenz $x_1(k) = x_0(k) - x_0(k - 1)$ für $k = 3 \dots 12$ und skizzieren Sie die Folgen $x_0(k)$, $x_0(k - 1)$ und $x_1(k)$. Stellen sie anschließend die Folgen mit Hilfe von MATLAB/Octave graphisch dar und vergleichen Sie die Ergebnisse mit den berechneten.

Befehl:

- `x0=[zeros(1,4) ones(1,6) zeros(1,3)];`
- `k=1:13;`
- `erste_d(k,x0);`

b) Gegeben ist der folgende Ausdruck für eine periodische Folge:

$$x_0(k) = \sum_{l=0}^{M-1} \delta(k - lP - 2). \quad (1.13)$$

Die Gesamtlänge der Folge ist MP , wobei P die Periode darstellt.

- Stellen sie eine periodische Folge nach Glg.(1.13) mit Hilfe von MATLAB/Octave graphisch dar.
(Befehl: `imp_per`)
- Geben Sie mit Hilfe von Glg.(1.13) einen Ausdruck für die dargestellte Folge an.
- Geben Sie den mathematischen Zusammenhang zwischen den Folgen $x_0(k)$ und $x_1(k)$ an.

Exponentialfolgen

a) Gegeben ist die zeitkontinuierlich komplexe Exponentialschwingung

$$x(t) = e^{j\omega t} \quad (1.14)$$

mit der Kreisfrequenz $\omega = 3\frac{1}{s}$. Die Exponentialschwingung wird mit der Frequenz $f_a = \frac{1}{\pi}$ Hz abgetastet.

- Geben sie die durch die Abtastung entstandene Folge $x(k)$ nach Glg.(1.4) an.
- Skizzieren Sie den Realteil und den Imaginärteil der komplexen Exponentialfolge.
- Geben Sie für die komplexe Exponentialfolge einen Ausdruck unter Verwendung der Impulsfolge $\delta_0(k)$ und Glg.(1.6) an.

b) Gegeben ist der Realteil einer komplexen Exponentialfolge

$$x(k) = \cos(k) \quad \forall \quad k = -\infty \dots \infty. \quad (1.15)$$

- Stellen Sie die Folge $x(k)$ im Bereich $k = 0 \dots 20$ graphisch dar.
(Befehl: `cos_abt`)
- Begründen Sie, warum $x(k)$ ($k \in [0 \dots 20]$) keine periodische Folge ist.

Abtastung, Aliasing Im folgenden wird ein *idealer Tiefpass* betrachtet, dessen Impulsantwort und Frequenzgang durch folgende Ausdrücke gegeben sind,

$$h_{TP}(t) = \frac{\omega_g \sin(\omega_g t)}{\pi \omega_g t} \quad \text{und} \quad H_{TP}(j\omega) = \begin{cases} 1 & |\omega| < \omega_g \\ 0.5 & |\omega| = \omega_g \\ 0 & \text{sonst.} \end{cases} \quad (1.16)$$

a) • Skizzieren Sie die Impulsantwort und den Frequenzgang des idealen Tiefpasses.

- Durch die Abtastung der zeitkontinuierlich Impulsantwort mit der Abtastfrequenz f_a wird die zeitdiskrete Folge

$$h_{TP}(k) = \frac{\omega_g}{\pi} \frac{\sin(\Omega_g k)}{\Omega_g k} \quad (1.17)$$

gewonnen. Der zugehörige Frequenzgang ist dabei durch die zeitdiskrete Fourier-Transformierte nach Glg.(1.7) gegeben.

- Welcher Zusammenhang besteht zwischen Ω_g , ω_g und der Abtastfrequenz f_a ?
- Berechnen Sie die maximale normierte Grenzfrequenz Ω_g für die kein Aliasing eintritt.

b) Im folgenden ist die Grenzfrequenz des idealen Tiefpasses gleich $f_g = 1$ kHz.

Stellen Sie die zeitdiskrete Impulsantwort und deren Frequenzgang für die drei Abtastfrequenzen

- 1) $f_a = 3$ kHz
- 2) $f_a = 1.5$ kHz
- 3) $f_a = 2$ kHz

graphisch dar.

(Befehl: `si(fg, fa)` z.B. `si(1,3)`)

- Interpretieren Sie die Ergebnisse für die Fälle 1) und 2).
- Fall 3:
 - Was für eine Folge stellt $h_{TP}(k)$ dar, Begründung?
 - Welche für die Nachrichtentechnik sehr wichtige Bedingung erfüllt dabei die Abtastung mit $f_a = 2$ kHz?

Kapitel 2

Lineare Systeme

Tobias Monsees

NW1, Raum N2370, Tel.: 0421/218-62407

E-Mail: tmonsees@ant.uni-bremen.de

2.1 Einführung

Der Versuch Lineare Systeme beschäftigt sich mit grundlegenden Methoden, um Signale zum Zwecke der Übertragung zu modulieren und mit Hilfe geeigneter Verfahren empfangen bzw. rekonstruieren zu können. Ferner wird aufgezeigt, wie sich für einen Bandpass-Kanal ein äquivalenter Basisband-Kanal herleiten lässt.

Für die Durchführung dieses Versuchs wird die Kenntnis des in den Folien zur Vorlesung Grundlagen der Nachrichtentechnik behandelten Stoffes vorausgesetzt, die unter <http://www.ant.uni-bremen.de/courses/gnt/> heruntergeladen werden können.

Dies bezieht sich im Besonderen auf die Themen

- Hilbert-Transformation
- Analytische Signale
- Äquivalente Tiefpass-Darstellung von Bandpass-Signalen
- Quadraturfilter

Die einzelnen Aufgaben zur Versuchsdurchführung finden sich jeweils in den Erläuterungen der Theorie.

2.2 Vorbereitung

Zur Vorbereitung des Versuchs ist die nachfolgend aufgeführte Theorie durchzuarbeiten. Ferner ist zu der in Kapitel 2.3 erläuterten Hilbert-Transformation eine MATLAB-Aufgabe zu bearbeiten. Der Code wird zur erfolgreichen Durchführung des Versuchs benötigt.

2.3 Hilbert-Transformation

Gegeben sei ein zeitkontinuierliches Cosinus-Signal der Frequenz $f_1 = 0,4\text{kHz}$

$$x_1(t) = \cos(2 \cdot \pi \cdot f_1 \cdot t) \quad (2.1)$$

Die korrespondierende Fourier-Transformierte ist

$$X_1(j\omega) = \text{Re} \{X_1(j\omega)\} + j \text{Im} \{X_1(j\omega)\} \quad (2.2)$$

Die Hilbert-Transformierte $\hat{X}_1(j\omega)$ zu $X_1(j\omega)$ ist im Spektralbereich definiert als

$$\hat{X}_1(j\omega) = \text{Im} \{X_1(j\omega)\} \cdot \text{sgn}(\omega) - j \text{Re} \{X_1(j\omega)\} \cdot \text{sgn}(\omega) \quad (2.3)$$

wobei $\text{sgn}(\omega)$ die Signumfunktion bezeichnet.

Fasst man die Hilbert-Transformation als eine Übertragungsfunktion $H_{\mathcal{H}}(j\omega)$ gemäß

$$\hat{X}_1(j\omega) = X_1(j\omega) \cdot H_{\mathcal{H}}(j\omega) \quad (2.4)$$

auf, so entspricht sie der folgenden Kennlinie.

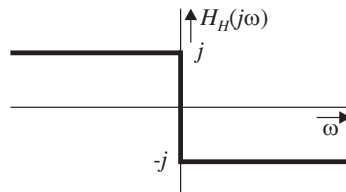


Abbildung 2.1: Übertragungsfunktion eines Hilbert-Transformators

Für ein Kosinus-Signal $x_1(t)$ (ohne Phasenverschiebung), welches ein reines reelles Spektrum besitzt, ist das Spektrum seiner Hilbert-Transformierten $\hat{x}_1(t)$ also rein imaginär.

2.3.1 Vorbereitungsaufgabe

AUFGABE

- Generieren Sie mit MATLAB einen Zeitvektor `t1` der Länge 20ms und mit dem Abtastintervall $\Delta T = \frac{1}{200}$ ms. Erzeugen Sie ein Cosinus-Signal $x_1(t)$ der Frequenz $f_1 = 0,4\text{kHz}$.
- Bestimmen Sie die Fouriertransformierte $X_1(j\omega)$ anhand der Funktion `f_reihe`.
- Stellen Sie das Zeitsignal mit `plot_zeitsignal` und seine Fouriertransformierte mit `bar_spektrum` dar.
- Bestimmen Sie die Hilbert-Transformierte $\hat{x}_1(t)$ bzw. `xh` mit Hilfe der Funktion `[th,xh]=hilbert_transformation(t1,x1)`.
- Bestimmen Sie die Fouriertransformierte $\hat{X}_1(j\omega)$ des Signal $\hat{x}_1(t)$ anhand der Funktion `f_reihe` und stellen Sie wiederum das Zeitsignal und das Spektrum dar. Was lässt sich über die Spektralanteile von $x_1(t)$ und $\hat{x}_1(t)$ aussagen ?

2.4 Analytische Signale

Die Spektren reeller Zeitsignale besitzen immer ein konjugiert gerades Spektrum, also einen linksseitigen und einen rechtsseitigen Spektralanteil.

Ein komplexes Zeitsignal, dessen Imaginärteil die Hilberttransformierte seines Realteils ist, besitzt stets ein Spektrum, das für negative Frequenzen verschwindet. Man nennt solche Signale **analytische Signale**.

Es soll nun gezeigt werden, wie anhand der Hilbert-Transformation ein reelles Zeitsignal $x_1(t)$ in ein analytisches Signal $x_1^+(t)$ überführt werden kann.

Man betrachte das Blockschaltbild in Bild 2.2.

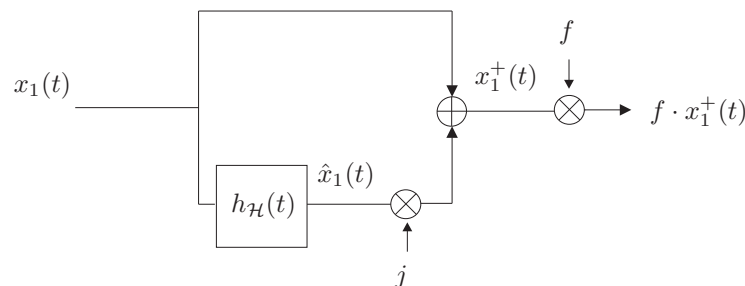


Abbildung 2.2: Blockschaltbild zur Erzeugung eines analytischen Signals

Die abschliessende Multiplikation mit dem Faktor f sei zunächst ausser acht gelassen.

Zu dem reellen Zeitsignal wird seine Hilbert-Transformierte gebildet, diese mit j multipliziert und beide Terme addiert, so dass das analytische Signal entsteht

$$x_1^+(t) = x_1(t) + j \hat{x}_1(t) \quad (2.5)$$

2.4.1 AUFGABE

- Erzeugen Sie für das Signal $x_1(t)$ das analytische Signal $x_1^+(t)$. Bestimmen Sie die Fouriertransformierten beider Signale. Stellen Sie beide Signale im Zeit- und Frequenzbereich dar.

Die Leistung eines Signals $x(t)$ ist wie folgt definiert:

$$E\{|x(t)|^2\} = \int_{-\infty}^{\infty} |X(j\omega)|^2 d\omega \quad (2.6)$$

- Bestimmen Sie die Leistung von $x_1(t)$.
- Bestimmen Sie die Leistung von $x_1^+(t)$.
- Welchen Wert muss der Faktor f annehmen, damit die Leistungen von $x_1(t)$ und $f \cdot x_1^+(t)$ identisch sind?

2.5 Demodulation und Tiefpass-Filterung

Prinzipiell stellt sich bei Signalen, welche zum Zwecke der Übertragung um eine Frequenz f_0 hochmoduliert wurden, die Aufgabe, das Signal am Empfänger wieder entsprechend zu rekonstruieren.

AUFGABE

- Laden Sie das Signal $x_{BP}(t)$ bzw. `xbp` und den dazugehörigen Zeitvektor `tbp`, indem Sie `load xbp` eingeben.
- Transformieren Sie das Signal in den Frequenzbereich und stellen Sie es im Intervall $-25\text{kHz} \leq f \leq 25\text{kHz}$ dar.

Die Vorgehensweise ist in Bild 2.3 dargestellt.

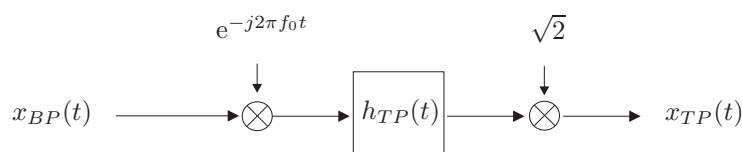


Abbildung 2.3: Demodulation und Tiefpassfilterung

Das Signal wird zunächst um die Frequenz f_0 demoduliert, damit das zuvor rechtsseitige Spektrum zentriert um die Frequenzachse liegt. Die anschließende Tiefpassfilterung mit entsprechender Grenzfrequenz führt dazu, dass das zuvor linksseitige Spektrum vollständig entfernt wird. Die abschliessende Normierung mit dem Faktor $\sqrt{2}$ wird aus dem Grund der Leistungserhaltung durchgeführt (siehe Abschnitt 2.4).

AUFGABE

- Demodulieren Sie das Signal $x_{BP}(t)$ mit einer Frequenz von $f_0 = 10\text{kHz}$.
- Stellen Sie das demodulierte Signal im Frequenzbereich bis $-25\text{kHz} \leq f \leq 25\text{kHz}$ dar.
- Führen Sie eine Filterung des demodulierten Zeitsignals mit einem Tiefpass-Filter der Grenzfrequenz 5kHz durch. Verwenden Sie hierzu den Befehl `x_tp = tiefpass(tbp, xbp, 5)`
- Normieren Sie das Signal mit dem Faktor $\sqrt{2}$.
- Transformieren Sie das erhaltene Signal in den Frequenzbereich. Stellen Sie es im Zeit- und Frequenzbereich dar.

2.6 Quadraturmischer

Eine Alternative zum obigen Konzept stellt der Quadraturmischer dar. Wiederum sei das Blockschaltbild betrachtet (Bild 2.4).

Es wird zunächst mit Hilfe der Hilbert-Transformierten das analytische Signal gebildet. Anschliessend wird eine Demodulation um die Frequenz f_0 durchgeführt und abschliessend mit dem Faktor $\frac{1}{\sqrt{2}}$ normiert.

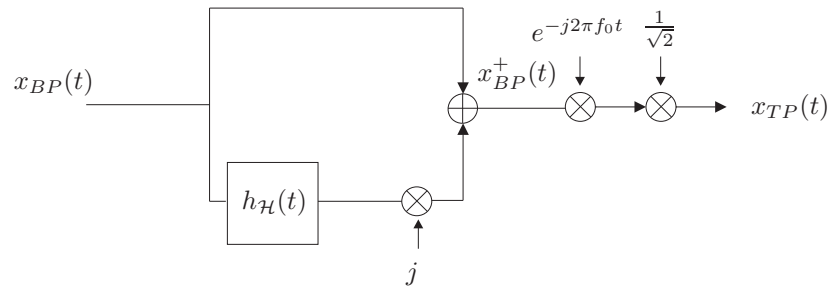


Abbildung 2.4: Quadraturmischer

AUFGABE

- Bilden Sie die Hilbert-Transformierte von $x_{BP}(t)$.
- Erzeugen Sie das analytische Signal $x_{BP}^+(t)$.
- Stellen Sie das analytische Signal $x_{BP}^+(t)$ im Frequenzbereich dar.
- Demodulieren Sie das analytische Signal um $f = 10\text{kHz}$ und normieren Sie es.
- Stellen Sie das Signal $x_{TP}(t)$ im Zeit- und Frequenzbereich dar.

2.7 Äquivalentes Basisbandmodell

2.7.1 Direkte Übertragung

Bild 2.5 zeigt eine Übertragungsstrecke.

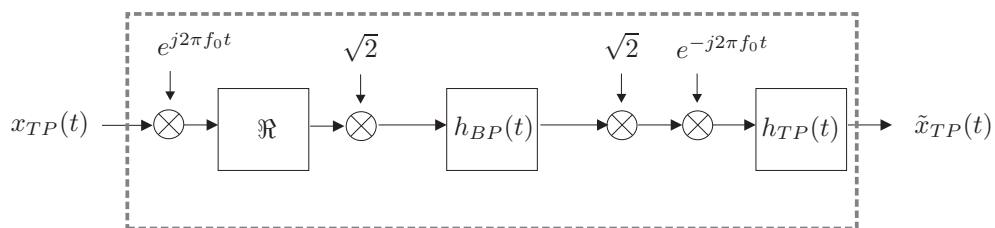


Abbildung 2.5: Übertragungsstrecke

Zunächst wird das Signal auf die Frequenz f_0 hochmoduliert. Hier sei $f_0 = 10\text{kHz}$. Anschliessend wird der Realteil des Signals genommen und mit $\sqrt{2}$ multipliziert, um die gleiche Leistung des Eingangssignals zu erhalten.

Der Übertragungskanal ist ein Bandpass-Kanal mit einer nicht konstanten Amplitude im Frequenzbereich.

Es folgt die Demodulation mit Leistungsnormierung und eine Bandbegrenzung mit Hilfe eines Tiefpass-Filters.

AUFGABE

- Laden Sie das Tiefpass-Signal $x_{TP}(t)$ mit `load xtp`
- Übertragen Sie das Signal bis zum dem Punkt, bevor es in den Übertragungskanal $h_{BP}(t)$ geht.
- Stellen Sie das Signal im Frequenzbereich dar.
- Laden Sie den Frequenzgang des Bandpass-Kanals $H_{BP}(j\omega)$ mit `load HBP`
- Stellen Sie den Frequenzgang des Kanals dar.
- Führen Sie die Übertragung über den Kanal als Multiplikation im Frequenzbereich durch.
- Stellen Sie das Signal im Frequenzbereich nach der Übertragung über den Kanal dar.
- Transformieren Sie das Signal zurück in den Zeitbereich und erzeugen Sie das Signal $\tilde{x}_{TP}(t)$. Verwenden Sie zur Tiefpassfilterung die Funktion `x_tp = tiefpass(t,x,f_g)`, wobei `f_g` die Grenzfrequenz mit $f_g = 5\text{kHz}$ sei.
- Stellen Sie $\tilde{x}_{TP}(t)$ im Frequenzbereich dar.

2.7.2 Äquivalentes Kanalmodell

An Stelle der Modulation des zu übertragenden Signal über die gesamte Übertragungstrecke ist es zu Simulationszwecken bei genauer Kenntnis bzw. Annahme des Kanals auch möglich, mit dem **äquivalenten Basisbandmodell** zu arbeiten.

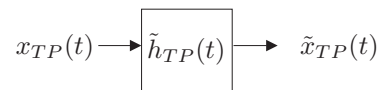


Abbildung 2.6: Äquivalentes Basisbandmodell

Hierbei wird die gesamte Übertragungstrecke einschliesslich des Kanals - wie in Bild 2.5 grau gestrichelt gekennzeichnet - als ein System $\tilde{h}_{TP}(t)$ aufgefasst.

Der Zusammenhang der beiden Systeme ist gegeben durch

$$\tilde{h}_{TP}(t) = \frac{1}{2} h_{BP}^+(t) \cdot e^{-j2\pi f_0 t} \quad (2.7)$$

AUFGABE

- Transformieren Sie den Frequenzgang des Bandpass-Kanals in den Zeitbereich, um $h_{BP}(t)$ zu erhalten.
- Erzeugen Sie den Äquivalenten Tiefpasskanal $\tilde{h}_{TP}(t)$.
- Transformieren Sie $\tilde{h}_{TP}(t)$ in den Frequenzbereich.
- Transformieren Sie $x_{TP}(t)$ in den Frequenzbereich.

- Führen Sie die Übertragung über den Kanal als Multiplikation im Frequenzbereich durch.
- Transformieren Sie das Signal in den Zeitbereich zurück.
- Stellen Sie das Signal $\tilde{x}_{TP}(t)$ im Zeit- und Frequenzbereich dar. Vergleichen Sie das Ergebnis mit der Übertragung über die gesamte Übertragungsstrecke.

Kapitel 3

Diskrete Signale und Systeme

Edgar Beck

NW1, Raum N2390, Tel.: 0421/218-62392

E-Mail: beck@ant.uni-bremen.de

3.1 Motivation

In den letzten Jahrzehnten gewann der Begriff "digital" eine immense Bedeutung in unserer Gesellschaft. Nach dem Einzug der digitalen Signalverarbeitung am Ende der 50er Jahre ist sie aus den heutigen Forschungs- und Entwicklungslabors nicht mehr wegzudenken. Technische Realisierungen wie z.B. der Computer, das Smartphone, der Mobilfunk oder SD-Karten sind nur einige Anwendungen, die die Bedeutung von digitalen Signalen und Systemen widerspiegeln.

Um dem Studenten im Rahmen seiner Ingenieurausbildung einen Einstieg in die Thematik der digitalen Signalverarbeitung zu ermöglichen, besteht die Aufgabe dieses Versuchs in der Einführung von zeitdiskreten Signalen und Systemen. Es sollen die wichtigsten Zusammenhänge und Kenngrößen erarbeitet und anhand der Simulationsumgebung MATLAB konkret untersucht und angewandt werden.

Hierzu gliedert sich der Versuchsaufbau in eine theoretische Einführung von zeitdiskreten Signalen und Systemen, welche in den Abschnitten 1.1.2 bis 3.4 erfolgt. Anschließend werden im Abschnitt 3.5 zeitdiskrete Zufallsprozesse behandelt. Diese spielen unter anderem in der Nachrichtenübertragung eine wesentliche Rolle, da sowohl Störungen wie z.B. das thermische Rauschen als auch das zu sendende Signal als zufällige Signale aufzufassen sind. Im letzten Abschnitt findet dann die Zusammenführung von Zufallsprozessen mit zeitdiskreten Systemen statt.

Da bei der Versuchsdurchführung die elementaren Zusammenhänge analysiert und diskutiert werden sollen, ist es notwendig, die theoretischen Grundlagen der vorangegangenen Abschnitte zu kennen. Diese werden bei Versuchsbeginn in Form eines Kolloquiums bzw. der Vorbereitungsaufgaben abgefragt: Bei mangelhafter Kenntnis bzw. Vorbereitung gilt der Versuch als nicht bestanden! Werden von einer Gruppe nicht alle Aufgaben innerhalb der vorgegebenen Zeit erfolgreich bearbeitet, so hat diese ein Protokoll des Versuchs abzugeben, welches die noch fehlenden Ergebnisse sowie deren Interpretationen zusammenfasst.

Weitere Anregungen zur Ausführung des Versuchs, wie auch zur Thematik selber, werden gerne vom zuständigen Betreuer entgegengenommen.

3.2 Elementare zeitdiskrete Signale

Die elementaren zeitdiskreten Signale sind bereits im ersten Versuch in 1.1.2 eingeführt worden. Für die Durchführung dieses Versuchs sollen dieselben Definitionen verwendet werden.

3.3 Transformationen zeitdiskreter Signale

3.3.1 Zeitdiskrete Fourier-Transformation

Die zeitdiskrete Fouriertransformation ist ebenfalls im ersten Versuch, im Abschnitt 1.1.2 eingeführt worden. Sie wird auch hier wieder benötigt.

3.3.2 z-Transformation

Einführung und Definition

Zur Analyse und Synthese von Systemen wird sehr häufig anstelle der Fourier-Transformation die Laplace-Transformation,

$$\mathcal{L}\{x_K(t)\} = X_K(s) = \int_{-\infty}^{\infty} x_K(t)e^{-st} dt. \quad (3.1)$$

verwendet. Die zugehörige Transformationsvariable $s = \sigma + j\omega$ umfasst dabei die gesamte komplexe Ebene.

Es stellt sich nun die Frage, ob es nicht auch für zeitdiskrete Signale sinnvoll ist, eine Erweiterung der Transformationsvariablen Ω auf die gesamte komplexe Ebene vorzunehmen.

Hierzu führt man die komplexe Transformationsvariable z ein, die durch die konforme Abbildung

$$z = e^{sT} = e^{(\sigma+j\omega)T} = e^{(\sigma T+j\Omega)} \quad (3.2)$$

mit der Laplace-Variablen s verknüpft ist. Diese konforme Abbildung bildet die gesamte s -Ebene eindeutig und nichtlinear in die z -Ebene ab, wobei sich folgende Korrespondenzen ergeben (siehe Tabelle 3.1).

s-Ebene	z-Ebene
linke komplexe Ebene	Inneres d. Einheitskreises
imaginäre Achse	Einheitskreis
rechte komplexe Ebene	äußeres d. Einheitskreises
$s = 0$	$z = 1 + j0$
$s = f_A/2$	$z = -1 + j0$

Tabelle 3.1: s-z Korrespondenzen

Die zugehörige z-Transformation ist dabei durch den Ausdruck

$$X(z) = \mathcal{Z}\{x(k)\} := \sum_{k=-\infty}^{\infty} x(k)z^{-k} \quad (3.3)$$

gegeben.

Die komplexe Variable z kann auch in der Form $z = r \cdot \exp(-j\omega T)$ mit $\omega T = \Omega$ geschrieben werden, so dass sich für die z -Transformation der Ausdruck

$$X(z) = \sum_{k=-\infty}^{\infty} x(k)r^{-k}e^{-j\Omega k} \quad (3.4)$$

ergibt. Es ist zu erkennen, dass für den Fall $|z| = r = 1$ die zeitdiskrete Fourier-Transformation Gl.(1.7) vorliegt. Mit anderen Worten, auf dem Einheitskreis sind zeitdiskrete Fourier-Transformation und z -Transformation identisch.

Fasst man die wichtigsten Aussagen dieses Abschnitts zusammen, so gilt:

- Zeitkontinuierliche Signale
 1. Aufgrund von Konvergenzproblemen (mehr dazu im nächsten Abschnitt) der Fourier-Integrale erweitert man die Transformation auf die gesamte komplexe Ebene und erhält die Laplace-Transformation.
 2. Auf der imaginären Achse sind Fourier-Transformation und Laplace-Transformation identisch.
- Zeitdiskrete Signale
 1. Es wird die konforme Abbildung $z = e^{sT}$ eingeführt und man erhält die z -Transformation, welche ein zeitdiskretes Signal in die komplexe z -Ebene transformiert.
 2. Auf dem Einheitskreis sind z -Transformation und zeitdiskrete Fourier-Transformation identisch.

Konvergenz

Die z -Transformierte existiert, wenn die Summe in Gl.(3.3) konvergiert, d.h. , wenn

$$\sum_{k=-\infty}^{\infty} |x(k)z^{-k}| < \infty \quad (3.5)$$

gilt. Schreibt man Gl.(3.3) in den Ausdruck

$$X(z) = \mathcal{Z} \{x(k)\} = \sum_{k=1}^{\infty} x(-k)z^k + \sum_{k=0}^{\infty} x(k) \left(\frac{1}{z}\right)^k \quad (3.6)$$

um, so stellen beide Teilsummen so genannte Taylorreihen über z bzw. $1/z$ dar. Es kann gezeigt werden, dass Taylorreihen innerhalb eines Kreises $|z| = r < R$ mit dem Radius R konvergieren. Angewandt auf die z -Transformierte bedeutet dies, dass das Konvergenzgebiet für $X(z)$ ein ringförmiges Gebiet darstellt.

Ein Beispiel einer Taylorreihe ist die „geometrische Reihe“

$$X(z) = \sum_{k=0}^{\infty} z^k = \frac{1}{1-z} \text{ fr } |z| < 1 \quad (3.7)$$

mit dem Konvergenzradius $R=1$.

Zur Darstellung der Konvergenzeigenschaften der z -Transformation wird im folgenden das Beispiel der Sprungfolge Gl.(1.3)

$$x_1(k) = \begin{cases} 1 & k \geq 0 \\ 0 & k < 0, \end{cases}$$

und der antikausalen Sprungfolge

$$x_2(k) = \begin{cases} 0 & k \geq 0 \\ -1 & k < 0, \end{cases} \quad (3.8)$$

d.h. $x_1(k) - x_2(k) = 1 \quad \forall k$ betrachtet. Mit Hilfe der „geometrischen Reihe“ Gl.(3.7) gilt,

$$\mathcal{Z}\{x_1(k)\} = \sum_{k=0}^{\infty} z^{-k} = \sum_{k=0}^{\infty} \left(\frac{1}{z}\right)^k = \frac{z}{z-1} \quad \forall |z| > 1 \quad (3.9)$$

und

$$\mathcal{Z}\{x_2(k)\} = \sum_{k=-\infty}^{-1} -z^{-k} = -\sum_{k=1}^{\infty} z^k = \frac{z}{z-1} \quad \forall |z| < 1. \quad (3.10)$$

Die erhaltenen Ergebnisse lassen nun folgende wichtige Rückschlüsse zu:

- Die Angabe einer z -Transformierten ist immer nur im Zusammenhang mit einem Konvergenzgebiet eindeutig
- Da sich die Konvergenzgebiete von $\mathcal{Z}\{x_1(k)\}$ und $\mathcal{Z}\{x_2(k)\}$ nicht überlappen, existiert die z -Transformierte der Zahlenfolge $x(k) = x_1(k) - x_2(k) = 1 \quad \forall k$ nicht.

3.4 Zeitdiskrete Systeme

3.4.1 Eigenschaften

Ein zeitdiskretes System ist mathematisch als eine Transformation oder als ein Operator $\mathcal{T}\{\cdot\}$ definiert, die bzw. der eine Eingangsfolge $x(k)$ in eine Ausgangsfolge $y(k)$ abbildet. Dies kann durch

$$y(k) = \mathcal{T}\{x(k)\} \quad (3.11)$$

oder durch Bild 3.1 ausgedrückt werden.

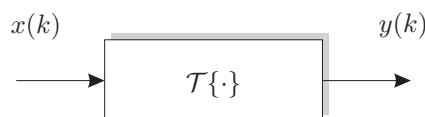


Abbildung 3.1: Zeitdiskretes System

In den folgenden Ausführungen werden ausschließlich lineare und zeitinvariante Systeme (LTI, *Linear Time Invariant*) betrachtet, d.h.

- **linear:** Es gilt das Superpositionsprinzip

$$\mathcal{T}\{a \cdot x_1(k) + b \cdot x_2(k)\} = a \cdot \mathcal{T}\{x_1(k)\} + b \cdot \mathcal{T}\{x_2(k)\}. \quad (3.12)$$

Eigenschaften der z-Transformation

Operation	Folge	Transformierte
	$x(k)$	$X(z)$
	$y(k)$	$Y(z)$
Linearität	$a \cdot x(k) + b \cdot y(k)$	$a \cdot X(z) + b \cdot Y(z)$
Verschiebung	$x(k - k_0)$	$z^{-k_0} X(z)$
Spiegelung	$x(-k)$	$X(1/z)$
konjugiert Komplexe	$x^*(k)$	$X^*(z^*)$
Realteil	$Re\{x(k)\}$	$\frac{1}{2}[X(z) + X^*(z^*)]$
Imaginärteil	$Im\{x(k)\}$	$\frac{1}{2j}[X(z) - X^*(z^*)]$
Faltung	$x(k) * y(k)$	$X(z) \cdot Y(z)$
Modulation	$(z_0)^k \cdot x(k)$	$X(z/z_0)$
lin. Gewichtung	$k \cdot x(k)$	$-z \frac{d}{dz} X(z)$
Anfangswert	$\lim_{k \rightarrow 0} x(k) = \lim_{z \rightarrow \infty} X(z),$ wenn $X(z)$ existiert	
Endwert	$\lim_{k \rightarrow \infty} x(k) = \lim_{z \rightarrow 1+0} (z-1)X(z),$ wenn $X(z)$ für $ z > 1$ existiert	

Tabelle 3.2: Eigenschaften der z-Transformation

- **zeitinvariant:** Die Systemantwort hängt nicht vom Zeitpunkt der Erregung ab

$$y(k - k_0) = \mathcal{T}\{x(k - k_0)\}, \quad k_0 \in \mathbb{N}. \quad (3.13)$$

Zur Ermittlung elementarer Eigenschaften von Systemen verwendet man in der Praxis bestimmte Eingangssignale bzw. Eingangsfolgen mit denen das jeweils zu analysierende System angeregt wird. Die wichtigsten Folgen,

- Impulsfolge $\delta(k)$
- Sprungfolge $\varepsilon(k)$
- komplexe Exponentialfolge,

wurden bereits in Abschnitt 1.1.2 eingeführt.

So wird z.B. für die Beschreibung des Zeitverhaltens eines Systems die Systemantwort auf die Erregung mit der Impulsfolge $\delta(k)$, d.h.

$$h(k) := \mathcal{T}\{\delta(k)\}.$$

betrachtet. Die Ausgangsfolge $h(k)$ wird *Impulsantwort* des Systems genannt.

Die allgemeine Systemantwort eines LTI-Systems kann mit Hilfe von Gl.(3.11) wie folgt berechnet werden:

$$\begin{aligned}
 y(k) &= \mathcal{T}\{x(k)\} = \mathcal{T}\left\{\sum_{i=-\infty}^{\infty} x(i)\delta(k-i)\right\} \\
 &= \sum_{i=-\infty}^{\infty} x(i)\mathcal{T}\{\delta(k-i)\} : \textit{linear} \\
 &= \sum_{i=-\infty}^{\infty} x(i)h(k-i) \\
 &= \sum_{i=-\infty}^{\infty} x(k-i)h(i) : \textit{zeitinvariant} \\
 &= x(k) * h(k). \tag{3.14}
 \end{aligned}$$

D.h. das Ausgangssignal ist als Faltungssumme von Eingangssignal und Impulsantwort darstellbar, wobei $*$ den Faltungsoperator bezeichnet.

3.4.2 Frequenzgang

Um die Wirkung eines LTI-Systems im Frequenzbereich zu untersuchen, betrachtet man die Antwort auf die Erregung mit der komplexen Exponentialfolge

$$x(k) = e^{j\omega T k} = e^{j\Omega k}$$

nach Gl.(1.4). Eingesetzt in die Faltungsoperation nach Gl.(3.14) ergibt sich als Ausgangsfolge $y(k)$ der Ausdruck

$$\begin{aligned}
 y(k) &= \mathcal{T}\{e^{j\Omega k}\} \\
 &= \sum_{i=-\infty}^{\infty} e^{j\Omega(k-i)}h(i) \\
 &= e^{j\Omega k} \sum_{i=-\infty}^{\infty} h(i)e^{-j\Omega i}. \tag{3.15}
 \end{aligned}$$

Der zweite Teil des Ausdrucks stellt dabei nach Gl.(1.7) die zeitdiskrete Fourier-Transformation der Impulsantwort $h(k)$ dar, welche als der *Frequenzgang* eines Systems bezeichnet wird,

$$H(e^{j\Omega}) = \sum_{k=-\infty}^{\infty} h(k)e^{-j\Omega k}.$$

Eingesetzt in Gl.(3.15) erhält man schließlich die Ausgangsfolge zu

$$y(k) = \mathcal{T}\{e^{j\Omega k}\} = e^{j\Omega k} H(e^{j\Omega}). \tag{3.16}$$

Dies bedeutet, dass die Ausgangsfolge wiederum eine komplexe Exponentialfolge derselben Frequenz darstellt, die lediglich in der Amplitude und Phase durch den Frequenzgang $H(e^{j\Omega})$ verändert wird. Mit anderen Worten gesagt, die komplexe Exponentialfolge stellt eine Eigenfunktion eines LTI-Systems dar. Dies ist eine typische Eigenschaft von linearen Systemen.

Der Frequenzgang $H(e^{j\Omega})$ ist dabei nach Abschnitt 1.1.2 periodisch in $\Omega = 2\pi$.

Der Betrag $|H(e^{j\Omega})|$ wird als *Amplitudengang* und die Phase $\phi(\Omega)$ als *Phasengang* des Systems mit

$$H(e^{j\Omega}) = |H(e^{j\Omega})| e^{-j\phi(\Omega)} \quad \text{bezeichnet.} \quad (3.17)$$

Um den Frequenzgang eines Systems zu messen, ist es nun möglich das System mit einer Exponentialschwingung einer Frequenz anzuregen und den Quotienten

$$H(e^{j\Omega}) = \frac{\mathcal{T}\{e^{j\Omega k}\}}{e^{j\Omega k}}, \quad (3.18)$$

zwischen Eingangs- und Ausgangssignal zu messen. Man erhält somit den Frequenzgang an der Frequenz des Eingangssignals. Durch das Variieren der Eingangsfrequenz ergibt sich somit der gesuchte Frequenzgang.

Analog zur Beziehung zwischen Eingangssignal und Ausgangssignal im Zeitbereich nach Gl.(3.14) kann durch Anwendung der zeitdiskreten Fourier-Transformation auf die Faltungssumme die Beziehung der Signale im Frequenzbereich angegeben werden. Es gilt:

$$Y(e^{j\Omega}) = X(e^{j\Omega}) \cdot H(e^{j\Omega}).$$

Für beliebige Eingangssignale ist damit der Frequenzgang durch den Ausdruck

$$H(e^{j\Omega}) = \frac{Y(e^{j\Omega})}{X(e^{j\Omega})} \quad (3.19)$$

gegeben.

3.4.3 Systemfunktion und Differenzgleichung

Der Frequenzgang $H(e^{j\Omega})$ besitzt den Nachteil, dass er lediglich das Verhalten des Systems auf dem Einheitskreis ($|e^{j\Omega}| = 1$) beschreibt. Aus systemtheoretischer Sicht ist es nun sinnvoll, ein System in der gesamten komplexen Ebene zu beschreiben. Hierzu bildet man die z-Transformierte nach Gl.(3.3) der Impulsantwort $h(k)$

$$H(z) = \sum_{k=-\infty}^{\infty} h(k)z^{-k}$$

und erhält die Systemfunktion $H(z)$. Wie bereits in Abschnitt(3.3.2) erläutert wurde, sind zeitdiskrete Fourier-Transformierte und z-Transformierte (Frequenzgang und Systemfunktion) für $z = e^{j\Omega}$ also auf dem Einheitskreis identisch.

Die Systemfunktion eines LTI-Systems ist immer als Quotient aus Polynomen in z in der Form

$$\begin{aligned} H(z) = \frac{Y(z)}{X(z)} &= \frac{\sum_{j=0}^m b_j z^{-j}}{\sum_{i=0}^n a_i z^{-i}} \\ &= \frac{\sum_{j=0}^m b_j z^{n-j}}{\sum_{i=0}^n a_i z^{n-i}} \end{aligned} \quad (3.20)$$

bzw. in der Form

$$H(z) = b_m \frac{\prod_{j=1}^m (z - z_{0_j})}{\prod_{i=1}^n (z - z_{\infty_i})} \quad (3.21)$$

darstellbar. Hierbei sind z_{0_j} und z_{∞_i} die Nullstellen bzw. Polstellen der Systemfunktion $H(z)$ und m der Zähler- bzw. n der Nennergrad.

Alternativ zur Impulsantwort $h(k)$ und der Systemfunktion $H(z)$ können zeitdiskrete Systeme auch durch eine Differenzgleichung mathematisch erfasst werden. Handelt es sich wie hier angenommen um ein LTI-System so besitzt die Differenzgleichung die Form

$$\sum_{i=0}^m a_i y(k - i + n) = \sum_{j=0}^n b_j x(k - j + n). \quad (3.22)$$

Die Koeffizienten a_i und b_j sind also zeitlich konstant und die Differenzgleichung linear. Vergleicht man die Differenzgleichung Gl.(3.22) mit der Systemfunktion nach Gl.(3.20), so ist zu erkennen, dass die Differenzgleichung und die Systemfunktion über die z -Transformation unter Anwendung des Verschiebungssatzes und der Linearität ineinander übergehen.

Wird z.B. die Systemfunktion

$$H(z) = \frac{z^2 + 1}{z^3} = z^{-1} + z^{-3}$$

betrachtet, so besitzt dieses System die Differenzgleichung

$$y(k) = x(k - 1) + x(k - 3).$$

Weiterhin stellt der Nenner von $H(z)$ den Anteil dar, welcher im System vom Ausgang zurückgekoppelt wird ($y(k - i + n) \quad \forall \quad i = 1 \dots n$). Dieser so genannte *rekursive* Anteil ist für die Stabilität des Systems verantwortlich. Der Zähler beschreibt dementsprechend den Vorwärtszweig, also den *nicht rekursiven* Anteil.

3.4.4 Übersicht

Fasst man die wesentlichen Eigenschaften diskreter Systeme zusammen und charakterisiert sie anhand der Impulsantwort, Systemfunktion oder Differenzgleichung, so gilt:

- **Linearität:** lineare Differenzgleichung
- **Zeitinvarianz:** Koeffizienten der Differenzgleichung und von $H(z)$ sind zeitinvariant
- **Kausalität,**

d.h., dass das Ausgangssignal $y(k)$ zu einem Zeitpunkt k_0 unabhängig ist von vorangegangenen Eingangssignalwerten $x(k_0 + 1)$, $x(k_0 + 2)$ etc.. Die notwendige und hinreichende Bedingung ist:

$$h(k) = 0 \quad \forall \quad k < 0. \quad (3.23)$$

– für $H(z)$ gilt:

$$H(z) = \sum_{k=0}^{\infty} h(k) z^{-k} \quad (3.24)$$

- * Es treten nur negative Potenzen von z auf, d.h. dass $n \geq m$ gilt.
- * $H(z)$ beschreibt eine Taylorreihe in $\frac{1}{z}$ und konvergiert damit außerhalb eines Kreises, siehe Abschnitt 3.3.2

- **BIBO-Stabilität,**

d.h. , dass für eine beschränkte Erregung $x(k)$ eine beschränkte Ausgangsfolge $y(k)$ vorliegt. Die notwendige und hinreichende Bedingung ist:

$$\sum_{k=-\infty}^{\infty} |h(k)| < \infty. \quad (3.25)$$

- Systeme mit endlicher Impulsantwort sind immer stabil
- Für $H(z)$ gilt:
 - * Pole z_{∞_i} für $i = 1..N$ liegen im Einheitskreis, $|z_{\infty_i}| < 1 \quad \forall \quad i = 1..N$
 - * Einheitskreis gehört zum Konvergenzgebiet von $H(z)$, siehe Gl.(3.5)

3.5 Zeitdiskrete Zufallsprozesse

In den vorangegangenen Abschnitten wurden Methoden vorgestellt, mit denen deterministische Signale beschrieben und die Übertragung solcher Signale über LTI-Systeme berechnet werden konnte. In den folgenden Abschnitten sollen diese Methoden auf nicht deterministische, d.h. zufällige Signale ausgedehnt werden. Zufallssignale können einerseits Nutzsignale sein (Sprachsignale, Musiksignale) oder auch Störsignale wie z.B. das thermische Rauschen eines Widerstands.

3.5.1 Zufallsvariable, Verteilungsfunktion, Verteilungsdichtefunktion

Zufallsvariable

Eine Variable X , die je nach dem Ausgang eines Versuchs, also in Abhängigkeit vom Zufall, verschiedene Werte annimmt, heißt eine Zufallsvariable (ZV).

Beispiele einer ZV sind

- der Amplitudenwert von thermischen Rauschen eines Widerstands
- die Augenzahl beim Würfeln.

Für die Praxis sind zwei Typen von Zufallsvariablen von besonderer Bedeutung, die kontinuierlichen (z.B. Rauschen) und die diskreten Zufallsvariablen (z.B. Augenzahl).

Verteilungsfunktion

Ausgehend von einer ZV kann nun für diese eine Wahrscheinlichkeit P angegeben werden, mit der sie einen gewissen Wert x_0 nicht überschreitet, d.h. $P(X \leq x_0)$. Als Verteilungsfunktion $F_X(x)$ der ZV X definiert man die Funktion

$$F_X(x) := P(X \leq x). \quad (3.26)$$

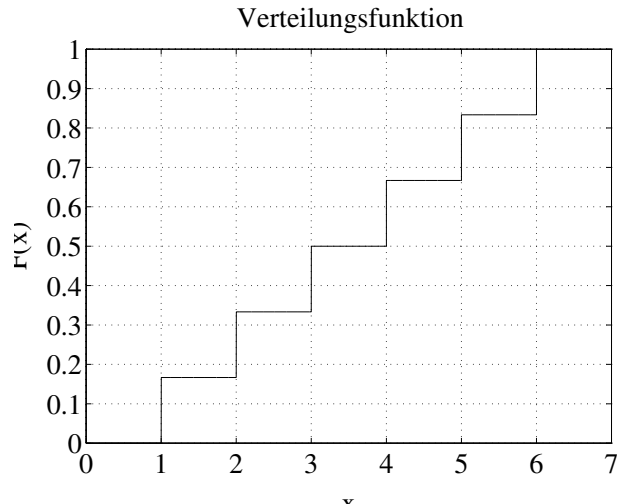


Abbildung 3.2: Verteilungsfunktion $F_X(x)$ für Augenzahl beim Würfeln

Man erhält für das Beispiel der Augenzahl unter anderem $P(X \leq 2) = 2/6$ oder $P(X \leq 5) = 5/6$. Die zugehörige Verteilungsfunktion ist in Bild 3.2 dargestellt.

Für die Verteilungsfunktion einer beliebigen ZV gilt immer:

$$F_X(\infty) = 1, \quad F_X(-\infty) = 0$$

Verteilungsdichtefunktion

Wird wiederum von einer ZV X ausgegangen und die Wahrscheinlichkeit $P(x \leq X < x + \Delta x) = F_X(x + \Delta x) - F_X(x)$ betrachtet, so ist die Verteilungsdichtefunktion (Wahrscheinlichkeitsdichtefunktion) definiert als

$$f_X(x) := \lim_{\Delta x \rightarrow 0} \frac{P(x \leq X < x + \Delta x)}{\Delta x} = \frac{\partial F_X(x)}{\partial x},$$

bzw.

$$F_X(x) = \int_{-\infty}^x f_X(\xi) d\xi \quad (3.27)$$

Bei gegebener Dichte $f_X(x)$ bzw. Verteilungsfunktion $F_X(x)$ ergibt sich die Wahrscheinlichkeit dafür, dass die ZV in ein vorgegebenes halboffenes Intervall fällt aus Gl.(3.27) zu

$$P(a \leq X < b) = F_X(b) - F_X(a) = \int_a^b f_X(\xi) d\xi. \quad (3.28)$$

Zur Ermittlung der Verteilungsdichtefunktion einer ZV, z.B. der Augenzahl beim Würfeln, führt man das Experiment Würfeln N mal durch (Stichprobe vom Umfang N) und berechnet die relativen Häufigkeiten, wie oft die Augenzahl z.B. kleiner als zwei ist. Die Ergebnisse lassen sich als Histogramm grafisch darstellen. Für den theoretischen Fall $N \rightarrow \infty$ ergibt sich aus dem Histogramm die zugehörige Verteilungsdichtefunktion.

Bild 3.3 zeigt die Verteilungsdichtefunktionen der beiden hier betrachteten Zufallsvariablen Augenzahl und Rauschen. Es ist deutlich der Unterschied einer diskreten und einer stetigen Zufallsvariablen zu erkennen.

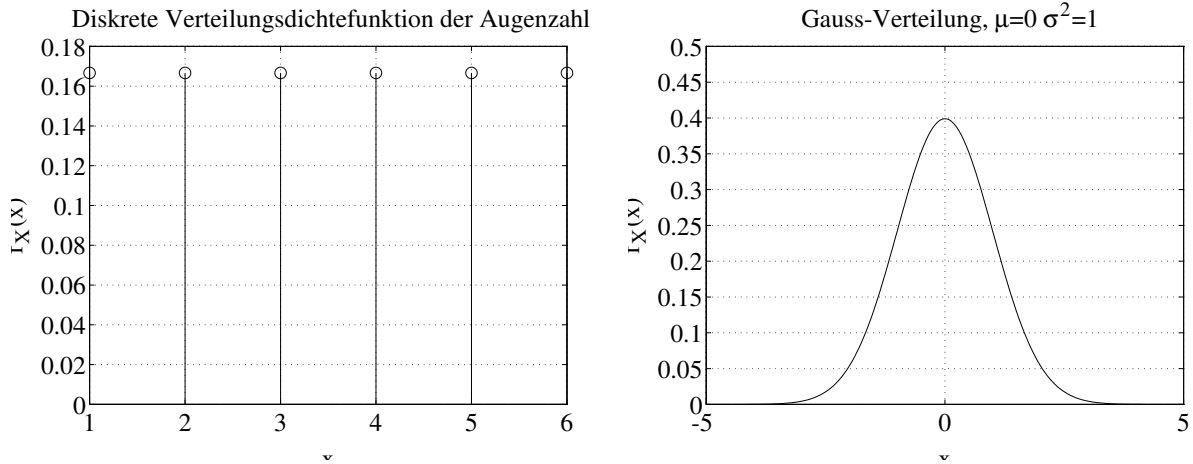


Abbildung 3.3: Dichtefunktionen

Die Verteilungsdichtefunktion des thermischen Rauschens stellt den in der Praxis sehr wichtigen Fall einer *Gauß-Verteilung* dar, deren Verteilungsdichtefunktion durch den Ausdruck

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3.29)$$

gegeben ist. Dabei bezeichnet μ den Erwartungswert und σ^2 die Varianz der Verteilung. Diese so genannten Momente werden im folgenden Abschnitt näher erläutert.

3.5.2 Statistische Kenngrößen (Momente von Verteilungen)

Ausgehend von der Verteilungsfunktion $F_X(x)$ oder der Verteilungsdichtefunktion $f_X(x)$ lassen sich so genannte Momente der Verteilung angeben. Man definiert als:

- i-tes Moment

$$\mu_i := \int_{-\infty}^{\infty} x^i f_X(x) dx \quad (3.30)$$

- und als i-tes zentrales Moment

$$\alpha_i := \int_{-\infty}^{\infty} (x - \mu_1)^i f_X(x) dx. \quad (3.31)$$

Hierbei ist nun zu beachten, dass diese Momente mit Hilfe der exakten Dichtefunktion $f_X(x)$ berechnet werden. In der Praxis liegt diese jedoch aufgrund der endlichen Anzahl an Stichproben N nicht vor. Aus diesem Grunde werden die einzelnen Momente unter Verwendung von Schätzfunktionen ermittelt.

Im folgenden sind einige der wichtigsten Momente und ihre Schätzfunktionen gegenübergestellt.

1. Erwartungswert/Mittelwert

- Erwartungswert:

$$E\{X\} = \mu_1 = \int_{-\infty}^{\infty} x f_X(x) dx \quad (3.32)$$

- Mittelwert (geschätzter Erwartungswert):

$$\hat{\mu}_1 = \frac{1}{N} \sum_{i=1}^N x(i) \quad (3.33)$$

2. Quadratischer Erwartungswert/Quadratischer Mittelwert

- Quadratischer Erwartungswert

$$E\{X^2\} = \mu_2 = \int_{-\infty}^{\infty} x^2 f_X(x) dx \quad (3.34)$$

- Quadratischer Mittelwert

$$\hat{\mu}_2 = \frac{1}{N} \sum_{i=1}^N x^2(i) \quad (3.35)$$

3. Varianz/geschätzte Varianz

- Varianz

$$\sigma^2 = \alpha_2 = \int_{-\infty}^{\infty} (x - \mu_1)^2 f_X(x) dx \quad (3.36)$$

- geschätzte Varianz

$$\hat{\alpha}_2 = \frac{1}{N-1} \sum_{i=1}^N (x(i) - \hat{\mu}_1)^2 \quad (3.37)$$

Hierbei bezeichnet $x(i)$ den Wert, den die ZV X für die i 'te Stichprobe annimmt, wenn insgesamt das Experiment N mal durchgeführt wird.

Zwischen den eingeführten Momenten einer Verteilung existiert der sehr wichtige Zusammenhang

$$\sigma^2 = E\{X^2\} - E\{X\}^2. \quad (3.38)$$

3.5.3 Statistische Unabhängigkeit, Summation und zentraler Grenzwertsatz

Statistische Unabhängigkeit

Betrachtet man n Zufallsvariablen, so besitzen diese Zufallsvariablen eine gemeinsame Verteilungsfunktion $F_{X_1, X_2, \dots, X_n}(x_1, x_2, \dots, x_n)$. Diese so genannte Verbundverteilungsfunktion stellt eine Funktion von n Parametern x_1, x_2, \dots, x_n dar.

Gegeben seien nun n Zufallsvariablen X_i für $i = 1 \dots n$ mit einer Verteilungsfunktion $F_{X_1, X_2, \dots, X_n}(x_1, x_2, \dots, x_n)$ bzw. Verbundverteilungsdichtefunktion $f_{X_1, X_2, \dots, X_n}(x_1, x_2, \dots, x_n)$. Die n Zufallsvariablen heißen statistisch unabhängig, wenn

$$F_{X_1, X_2, \dots, X_n}(x_1, x_2, \dots, x_n) = F_{X_1}(x_1) \cdot F_{X_2}(x_2) \cdot \dots \cdot F_{X_n}(x_n) \quad (3.39)$$

bzw.

$$f_{X_1, X_2, \dots, X_n}(x_1, x_2, \dots, x_n) = f_{X_1}(x_1) \cdot f_{X_2}(x_2) \cdot \dots \cdot f_{X_n}(x_n) \quad (3.40)$$

gilt.

Summe zweier statistisch unabhängiger Zufallsvariablen

Gegeben seien zwei statistisch unabhängige Zufallsvariablen X_i für $i = 1, 2$ mit der zugehörigen Verteilungsfunktion $F_{X_i}(x)$ für alle X_i . Die Zufallsvariable Y mit

$$Y = X_1 + X_2 \quad (3.41)$$

besitzt die Verteilungsdichtefunktion

$$f_Y(y) = f_{X_1}(x_1) * f_{X_2}(x_2). \quad (3.42)$$

D.h. die Verteilungsdichtefunktion der Summe zweier statistisch unabhängiger Zufallsvariablen ergibt sich aus der Faltung der beiden Verteilungsdichtefunktionen der Zufallsvariablen.

Zentraler Grenzwertsatz

Gegeben seien N statistisch unabhängige Zufallsvariablen X_i für $i = 1 \dots N$ mit der zugehörigen Verteilungsfunktion $F_{X_i}(x)$ für alle X_i . Weiterhin besitzen alle Verteilungen die Varianz $\sigma^2 = 1$ und den Erwartungswert $E\{X\} = 0$.

Es gilt [Pro95]:

Die Zufallsvariable Y mit

$$Y = \frac{1}{\sqrt{N}} \sum_{i=1}^N X_i \quad (3.43)$$

besitzt für $N \rightarrow \infty$ die Verteilungsdichtefunktion

$$f_Y(y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}}, \quad (3.44)$$

d.h., sie ist gaußverteilt mit $E\{Y\} = 0$ und $\sigma^2 = 1$, was auch als *normalverteilt* bezeichnet wird.

3.5.4 Zufallsprozesse und ihre Kenngrößen

Nach der Einführung von Zufallsvariablen soll nun noch zusätzlich die Zeit berücksichtigt werden. Mit anderen Worten, betrachtet man die Zeitachse und ordnet jedem diskreten Zeitpunkt k eine ZV X_k zu, so erhält man einen zeitdiskreten *stochastischen Prozess* $\{x(k)\}$. Somit liegt jedem Zeitpunkt eine Verteilungsdichtefunktion zu Grunde, aus der die jeweiligen Momente der ZV X_k berechnet werden können.

Ein Beispiel für einen diskreten Zufallsprozess ist das thermische Rauschen, bei dem zu jedem diskreten Zeitpunkt k eine Gauß-Verteilung nach Gl.(3.29) vorliegt.

Stochastische Prozesse können weiterhin komplexwertig oder reellwertig sein, wobei ein komplexwertiger Prozess als Summe zweier reellwertiger aufgefasst werden kann.

Stationarität

Sind alle Momente der Zufallsvariablen X_k für alle diskreten Zeitpunkte k dieselben, so ist der Prozess im strengen Sinne stationär (*SSS, Strong Sense Stationary*).

Wird lediglich gefordert, dass der Erwartungswert $E\{X_k\}$ zeitlich unabhängig ist und die Autokorrelationsfunktion, siehe Gl.(3.45), nur von der Zeitdifferenz κ abhängt, so wird dieser Prozess als stationär im weiteren Sinne (*WSS, Wide Sense Stationary*) bezeichnet.

Ist ein Prozess stationär, so kann er zusätzlich die Eigenschaft der Ergodizität aufweisen. Dies bedeutet, dass die mittels der Verteilungsdichtefunktion der Zufallsvariablen berechneten Momente auch über die Zeitvariable k berechnet werden können. Im Rahmen dieses Versuchs soll auf diese Eigenschaft jedoch nicht weiter eingegangen werden.

Für die folgenden Betrachtungen wird nun prinzipiell immer von *WSS*-Prozessen ausgegangen.

Autokorrelation- und Kreuzkorrelationsfolgen

Die Autokorrelationsfolge eines Zufallsprozesses $\{x(k)\}$ ist definiert als

$$r_{xx}(\kappa) := E\{X_k^* \cdot X_{k+\kappa}\}. \quad (3.45)$$

Für den Fall $\kappa = 0$ gilt mit Gl.(3.38)

$$r_{xx}(0) = E\{|X_k|^2\} = \sigma_x^2 + |\mu_x|^2, \quad (3.46)$$

mit $\mu_x = E\{X\}$. Da der Term $E\{|X_k|^2\}$ die mittlere Leistung eines Prozesses darstellt, ist diese somit durch den Wert der Autokorrelationsfolge für $\kappa = 0$ gegeben.

Weiterhin besitzt die Autokorrelationsfolge die Eigenschaft, dass sie *konjugiert gerade* ist, d.h.

$$r_{xx}(-\kappa) = r_{xx}^*(\kappa) \quad (3.47)$$

gilt.

Betrachtet man zwei stationäre Zufallsprozesse $\{x(k)\}$ und $\{y(k)\}$ so ist die Kreuzkorrelationsfolge definiert als

$$r_{xy}(\kappa) := E\{X_k^* \cdot Y_{k+\kappa}\}. \quad (3.48)$$

Bezüglich ihrer Symmetrieeigenschaften gilt:

$$r_{xy}(-\kappa) = r_{yx}^*(\kappa). \quad (3.49)$$

Autokovarianz- und Kreuzkovarianzfolgen

In vielen Anwendungsfällen interessiert man sich für die zentralen Momente. Auch hier lässt sich analog zur Autokorrelationsfolge eines Prozesses $\{x(k)\}$ die so genannte Autokovarianzfolge

$$\begin{aligned} c_{xx}(\kappa) &:= E\{(X_k^* - \mu_x^*) \cdot (X_{k+\kappa} - \mu_x)\} \\ &= r_{xx}(\kappa) - |\mu_x|^2 \end{aligned} \quad (3.50)$$

definieren. Entsprechend gilt für die Kreuzkovarianzfolge zweier Prozesse

$$\begin{aligned} c_{xy}(\kappa) &:= E\{(X_k^* - \mu_x^*) \cdot (Y_{k+\kappa} - \mu_y)\} \\ &= r_{xy}(\kappa) - \mu_x^* \mu_y. \end{aligned} \quad (3.51)$$

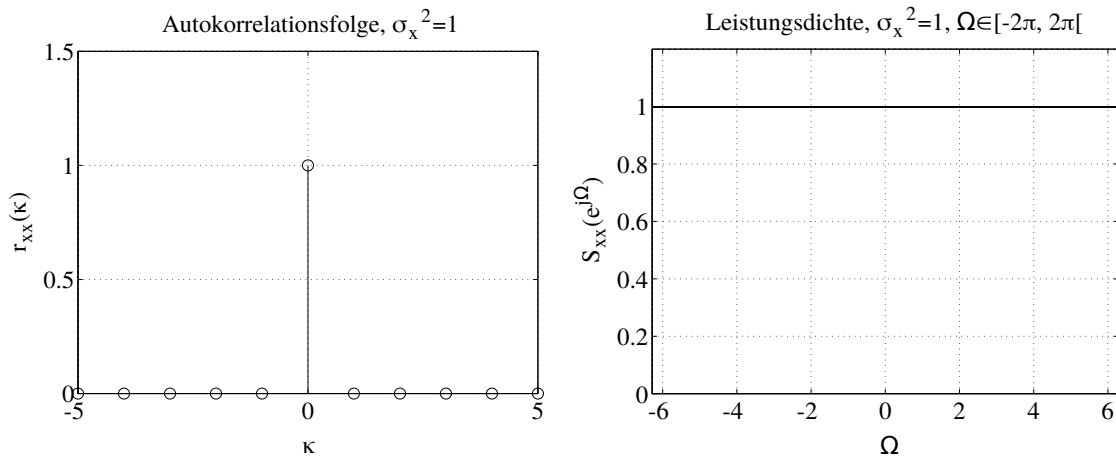


Abbildung 3.4: Autokorrelationsfolge und spektrale Leistungsdichte eines weißen Prozesses

Korrelation und Orthogonalität

In diesem Abschnitt sollen kurz zwei der wesentlichen Eigenschaften von Zufallsprozessen, die Korrelation und die Orthogonalität erörtert werden.

Ist die Kreuzkovarianzfolge $c_{xy}(\kappa)$ nach Gl.(3.51) zweier Prozesse identisch Null, d.h. gilt

$$c_{xy}(\kappa) = 0 \quad \forall \quad \kappa, \quad (3.52)$$

dann sind die Zufallsvariablen X_k und Y_k der Prozesse für alle k *unkorreliert*. Somit sind auch die beiden Prozesse $\{x(k)\}$ und $\{y(k)\}$ unkorreliert.

Für die Kreuzkorrelationsfolge ergibt sich für unkorrelierte Prozesse mit Gl.(3.51) die Beziehung

$$r_{xy}(\kappa) = \mu_x^* \mu_y. \quad (3.53)$$

Entsprechend kann man auch die Korrelationseigenschaften eines Prozesses anhand der Autokovarianzfolge nach Gl.(3.50) angeben. Besitzt $c_{xx}(\kappa)$ die Form

$$c_{xx}(\kappa) = \begin{cases} \sigma_x^2 & \kappa = 0 \\ 0 & \kappa \neq 0, \end{cases} \quad (3.54)$$

d.h. stellt die Autokovarianzfolge eine Impulsfolge der Höhe σ_x^2 dar, dann sind die Zufallsvariablen des zugrundeliegenden Prozesses unkorreliert. Ein solcher Prozess wird als *weiß* bezeichnet und seine Varianz beträgt σ_x^2 . Bild 3.4 zeigt die Autokovarianzfolge eines weißen (mittelwertfreien) Prozesses.

Weiterhin gelten folgende wichtige Zusammenhänge zwischen der Korrelation und der statistischen Unabhängigkeit:

- Sind zwei beliebige Prozesse statistisch unabhängig, so sind sie auch unkorreliert.
- Der Umkehrschluss gilt nur für gaußverteilte Prozesse, d.h. sind zwei gaußverteilte Prozesse unkorreliert, so sind sie auch statistisch unabhängig.

Schließlich nennt man zwei Prozesse *orthogonal* bei Gültigkeit von

$$r_{xy}(\kappa) = 0 \quad \forall \quad \kappa, \quad (3.55)$$

d.h., wenn die Kreuzkorrelationsfolge identisch Null ist. Mit Gl. (3.53) gilt, dass zwei unkorrelierte Prozesse orthogonal sind, wenn mindestens ein Mittelwert Null ist.

3.5.5 Spektrale Darstellung zeitdiskreter stochastischer Prozesse

Zeitdiskrete Signale bzw. Folgen $x(k)$ kann man, wie bereits ausgeführt mit Hilfe der zeitdiskreten Fourier-Transformation, in den Frequenzbereich transformieren. Es ist zu beachten, dass die Anwendung der Fourier-Transformation auf Zufallssignale auf Grund der Konvergenzvoraussetzung der Integrale nicht möglich ist. Um dennoch eine Aussage über die spektralen Eigenschaften von Zufallssignalen zu erhalten ist die Transformation der Autokorrelationsfolge $r_{xx}(\kappa)$ möglich (*Wiener-Khintchine-Theorem*). D.h.,

$$S_{xx}(e^{j\Omega}) = \sum_{\kappa=-\infty}^{\infty} r_{xx}(\kappa) e^{-j\Omega\kappa} \quad (3.56)$$

gilt. Dabei wird $S_{xx}(e^{j\Omega})$ als die spektrale *Autoleistungsdichte* des Prozesses $x\{k\}$ bezeichnet.

Unter der oben getroffenen Annahme stationärer Prozesse ist die Autokorrelationsfolge konjugiert gerade. Die spektrale Autoleistungsdichte ist stets eine reelle Funktion,

$$S_{xx}(e^{j\Omega}) \in \mathbb{R}. \quad (3.57)$$

Dies bedeutet, dass anhand der Autoleistungsdichte eines Prozesses keine Aussage über die Phase möglich ist. Für den Fall eines weißen Zufallsprozesses ist in Bild 3.4 die Autoleistungsdichte dargestellt.

In vielen Anwendungsfällen ist die Leistung eines Zufallsprozesses eine entscheidende Größe, z.B. zur Ermittlung eines Signal-Rauschabstands bei der Übertragung von Signalen. Hierbei kann die Berechnung zum einen im Zeitbereich oder im Frequenzbereich durchgeführt werden. Mit Gl.(3.46) und der inversen diskreten Fourier-Transformation von $S_{xx}(e^{j\Omega})$ gilt

$$E\{|X_k|^2\} = r_{xx}(0) = \frac{1}{2\pi} \int_{-\pi}^{\pi} S_{xx}(e^{j\Omega}) d\Omega. \quad (3.58)$$

3.6 Zeitdiskrete Zufallsprozesse und zeitdiskrete LTI-Systeme

3.6.1 Korrelationsfolgen und Leistungsdichtespektren des Ausgangsprozesses

Der letzte Abschnitt befasst sich mit der Übertragung eines stochastischen Prozesses über ein LTI-System, d.h der Zusammenführung von Systemen und Zufallsprozessen.

Ausgehend von der elementaren Eigenschaft eines LTI-Systems, dass die Ausgangsfolge $y(k)$ durch die Faltungssumme

$$y(k) = x(k) * h(k)$$

nach Gl.(3.14) zwischen Eingangsfolge $x(k)$ und Impulsantwort $h(k)$ darstellbar ist, können folgende Korrelationsfolgen und ihre spektralen Leistungsdichten berechnet werden.

Die Autokorrelationsfolge $r_{yy}(\kappa)$ des Ausgangsprozesses ist durch den Ausdruck

$$r_{yy}(\kappa) = r_{hh}^E(\kappa) * r_{xx}(\kappa) \quad (3.59)$$

gegeben. Da die Impulsantwort ein deterministisches Energiesignal darstellt bezeichnet man den Ausdruck

$$r_{hh}^E(\kappa) = \sum_{i=-\infty}^{\infty} h^*(i)h(i + \kappa) \quad (3.60)$$

als *Energiekorrelationsfolge*.

Man erhält also die Autokorrelationsfolge des Ausgangsprozesses durch Faltung der Autokorrelationsfolge des Eingangsprozesses mit der Energiekorrelationsfolge der Impulsantwort.

Durch Anwendung der zeitdiskreten Fourier-Transformation auf Gl.(3.59) folgt im Frequenzbereich der Zusammenhang

$$S_{yy}(e^{j\Omega}) = |H(e^{j\Omega})|^2 S_{xx}(e^{j\Omega}). \quad (3.61)$$

Analog zur Autokorrelationsfolge kann auch die Kreuzkorrelationsfolge zwischen dem Eingangsprozess und dem Ausgangsprozess berechnet werden. Es gilt:

$$r_{xy}(\kappa) = h(\kappa) * r_{xx}(\kappa) \quad (3.62)$$

im Zeitbereich und im Frequenzbereich

$$S_{xy}(e^{j\Omega}) = H(e^{j\Omega}) S_{xx}(e^{j\Omega}). \quad (3.63)$$

3.6.2 Systemidentifikation

Eine der bekanntesten Anwendungen der hier dargestellten Zusammenhänge ist die so genannte *Systemidentifikation*. So ist es z.B. bei der Übertragung von Daten über einen Kanal (z.B. Mobilfunkkanal beim D-Netz) von großem Vorteil, wenn die Impulsantwort des Kanals am Empfänger bekannt ist. Eine Möglichkeit besteht nun darin, dass als Eingangsprozess ein weißer und mittelwertfreier Prozess verwendet wird. Die Autokorrelationsfolge stellt somit nach Gl.(3.54) eine Impulsfolge

$$r_{xx}(\kappa) = \begin{cases} \sigma_x^2 & \kappa = 0 \\ 0 & \kappa \neq 0, \end{cases} \quad (3.64)$$

dar.

Unter der Annahme, dass es möglich ist, am Ausgang des Systems auf das Eingangssignal zurückzugreifen, kann die Impulsantwort mit Gl.(3.62) und Gl.(3.64) berechnet werden durch

$$h(\kappa) = \frac{r_{xy}(\kappa)}{\sigma_x^2}, \quad (3.65)$$

bzw. der Frequenzgang mit Hilfe von

$$H(e^{j\Omega}) = \frac{S_{xy}(e^{j\Omega})}{\sigma_x^2}. \quad (3.66)$$

Ist es hingegen nicht möglich den Eingangsprozess zu verwenden, so kann mit Hilfe der Autokorrelationsfolge des Ausgangs lediglich der Betrag des Frequenzgangs ermittelt werden, nicht aber die Phase. Es gilt:

$$|H(e^{j\Omega})|^2 = \frac{S_{yy}(e^{j\Omega})}{\sigma_x^2}. \quad (3.67)$$

3.7 Vorbereitungsaufgaben

3.7.1 Vorbemerkungen

- Geben Sie in MATLAB nach jeder Aufgabe die Befehlsfolge
close all
clear all
ein.

In den folgenden drei Aufgaben werden zeitdiskrete Systeme analysiert. Bei der Darstellung der Pole und Nullstellen in der komplexen z -Ebene gelten die folgenden üblichen Symbole:

- Pole $\hat{=}$ x
- Nullstellen $\hat{=}$ o

Aufgabe 1: Zeitdiskrete Systeme

Geben ist die Differenzgleichung eines LTI-Systems

$$y(k) = x(k) + x(k+1) + x(k+2) + x(k+3). \quad (3.68)$$

- a)
 - Berechnen sie mit Hilfe der Beziehungen nach Glg.(3.22) und Glg.(3.20) die Systemfunktion $H(z)$.
 - Geben Sie die Polstellen (POL) und Nullstellen (NST) von $H(z)$ an. (Matlab-Befehl: `roots`)
- b) Laden Sie das gegebene System in MATLAB (Befehl `imp_trans_pz_1`).
 - Ist das hier betrachtete System reellwertig? Begründung!
 - Welche Bedingung erfüllt die Übertragungsfunktion, wenn das System reellwertig ist?
 - Welchen typischen Verlauf besitzt der Amplitudengang $|H(e^{j\Omega})|$?
 - Ordnen Sie die NST'en von $|H(e^{j\Omega})|$ den NST'en im z -Diagramm zu.
 - Geben Sie eine Begründung an, warum das System nicht kausal ist.
- c) Im weiteren sollen nun Polstellen mit $z_{\infty} = 0$ zugefügt werden. (In MATLAB Aufruf `imp_trans_pz_1(n)`, wobei n die Anzahl der Pole ist.)
 - Wieviele Polstellen sind mindestens einzufügen, wenn das System kausal sein soll?
 - Wie lautet für diesen Fall die Systemfunktion $H(z)$?
 - Berechnen Sie für diesen Fall die zeitdiskrete Impulsantwort.
 - Warum ändert sich trotz geänderter Impulsantwort der Betrag der Übertragungsfunktion nicht?

3.8 Versuchsdurchführung

Aufgabe 1: Stabilität zeitdiskreter Systeme

Laden Sie durch den MATLAB-Aufruf `imp_pz_2` ein neues System.

- a) Ordnen Sie die Polstellen und NST'en den Frequenzen im Amplitudengang $|H(e^{j\Omega})|$ zu.
- b) Verschieben Sie den Pol $z_{\infty} = 0.48$ in Richtung des Einheitskreises und darüber hinaus. MATLAB-Aufrufe `imp_pz_2(0.8)`, `imp_pz_2(1.0)`, `imp_pz_2(1.1)`, usw.
 - Beobachten Sie bei der Verschiebung die Impulsantwort des Systems. Welche Eigenschaft des Systems verändert sich durch die Verschiebung?

c) Am Eingang des Systems liegt nun das zeitdiskrete Signal

$$x(k) = e^{j k \frac{\pi}{2}} + e^{j k \frac{\pi}{10}} + 1$$

an. Geben Sie mit Hilfe der Eigenschaft nach Glg.(3.16) das Ausgangssignal $y(k)$ an (Beträge und Phasen der Übertragungsfunktion können abgelesen werden).

Aufgabe 2: Spezielle zeitdiskrete Systeme

Laden Sie durch den MATLAB-Aufruf `pz_3a` ein neues System.

- a)
- Geben Sie die Polstellen und NST'en des Systems an (Vielfachheit Eins).
 - Berechnen Sie die Systemfunktion $H(z)$ nach Glg.(3.21) mit $b_M = 1$.
 - Für dieses System besteht zwischen den Polstellen und NST'en der Zusammenhang

$$z_{0\nu} = 1/z_{\infty\nu}^*.$$

Welche Eigenschaft besitzt der Frequenzgang $H(e^{j\Omega})$ für diese Pol-Nullstellenlagen und erklären Sie diese Eigenschaft anschaulich?

- Welche Anwendung findet dieses System in der Nachrichtentechnik?

b) Laden Sie durch den MATLAB-Aufruf `pz_3b` ein neues System.

Dieses System besitzt einen vierfachen Pol bei $z_\infty = 0$ und NST'en bei $z_{01} = 1 + j$, $z_{02} = 1 - j$, $z_{03} = 1/(1 - j)$ und $z_{04} = 1/(1 + j)$.

- Ist das System komplex- oder reellwertig?
- Ist das System kausal?
- Ist das System stabil?
- Für dieses System besteht zwischen den NST'en der Zusammenhang

$$z_{01} = 1/z_{03}^*, \quad z_{02} = 1/z_{04}^*$$

Welche Eigenschaft besitzt die Phase von $H(e^{j\Omega})$ für diese Nullstellenlagen und erklären Sie diese Eigenschaft anschaulich?

Wie müssen die Pole für diese Systemeigenschaft notwendigerweise liegen?

- Welche Anwendung findet dieses System in der Audiotechnik?

Aufgabe 3: Zufallsvariable, Verteilungsfunktion, Verteilungsdichtefunktion

Gegeben ist die Verteilungsdichtefunktion $f_{X_1}(x_1)$ der Zufallsvariable X_1 mit

$$f_{X_1}(x_1) = \frac{1}{\sqrt{2\pi}3} e^{-\frac{(x-4)^2}{18}} \quad (3.69)$$

- Geben Sie die Varianz, den Erwartungswert und den quadratischen Erwartungswert der Zufallsvariable an.
- Führen Sie die weiteren Schritte für die Fälle $N = 10^2, 10^3, 10^4$ durch:

- Erzeugen Sie N Werte einer gaußverteilten Zufallsvariablen X_2 .
(Befehl: `x2=randn(N,1)`)
- Geben Sie für jeden Fall den geschätzten Erwartungswert und die geschätzte Varianz an
(Tabelle).
(Befehle: `mu=mean(x2)`; `sigma_quadrat=var(x2)`;))
- Stellen Sie für jeden Fall das jeweilige Histogramm der Verteilung dar.
(Befehle: `figure`; `[d,x] = hist(x2,100)`; `bar(x,d)`; für alle drei Fälle)
- Geben Sie einen analytischen Ausdruck für die Verteilungsdichtefunktion $f_{X_2}(x_2)$ der Zufallsvariable X_2 an.
Wie nennt man diese spezielle Verteilung?
- Geben Sie den Zusammenhang an, wie aus der Zufallsvariablen X_2 die Zufallsvariable X_1 berechnet werden kann.

Aufgabe 4: Summation von Zufallsvariablen, Zentraler Grenzwertsatz

Im folgenden sind I voneinander statistisch unabhängige Zufallsvariablen $X_i \forall i = 0 \dots I - 1$ gegeben. Die Zufallsvariablen sind alle gleichverteilt im Intervall $[0,1[$ mit dem Erwartungswert $E\{X_i\} = 0.5$ und der Varianz $\sigma^2 = 1/12 \forall i$.

- a) Für die Summation zweier Zufallsvariablen

$$Y = X_1 + X_2 \quad (3.70)$$

gilt für die Verteilungsdichtefunktion

$$f_Y(y) = f_{X_1}(x_1) * f_{X_2}(x_2), \quad (3.71)$$

d.h. die Verteilungsdichtefunktion von Y ergibt sich aus der Faltungssumme der einzelnen Verteilungsdichtefunktionen von X_1 und X_2 .

- Skizzieren Sie die Verteilungsdichtefunktion $f_{X_i}(x_i)$.
- Skizzieren Sie die Verteilungsdichtefunktion $f_Y(y)$ mit Hilfe von Glg.(3.71).
- Für die folgenden Teilaufgaben werden $N = 10^4$ Stichproben verwendet
 - Erzeugen Sie zwei voneinander statistisch unabhängige Zufallsvariablen X_1 und X_2 und die Zufallsvariable $Y = X_1 + X_2$
Befehle:
`x1=rand(10000,1)`;
`x2=rand(10000,1)`;
`y=x1+x2`;
 - Stellen Sie die Histogramme der Zufallsvariablen X_1, X_2 und Y dar
Befehle:
`figure`; `[d,x] = hist(x1,100)`;
`bar(x,d)`;
`figure`; `[d,x] = hist(x2,100)`;
`bar(x,d)`;
`figure`; `[d,x] = hist(y,100)`;
`bar(x,d)`;

- Vergleichen Sie die Ergebnisse mit Ihren Skizzen

b) Sei nun

$$Y = \sum_{i=1}^I X_i \quad \text{for } I = 3, 10, 10^2$$

- Stellen Sie die Verteilungsdichtefunktion $f_{X_i}(x_i)$ und $f_Y(y)$ für alle Werte von I graphisch dar, und schreiben Sie in einer Tabelle die jeweiligen geschätzten Erwartungswerte und Varianzen auf.

(Befehl: `grenz(I, 'u')`)

- Welcher Zusammenhang besteht zwischen dem (geschätzten) Erwartungswert bzw. der (geschätzten) Varianz der Zufallsvariable X_i und denen der Zufallsvariable Y .

c) Sei nun

$$Y = \frac{1}{\sqrt{I}} \sum_{i=1}^I X_i \quad \text{for } I = 3, 10, 10^2$$

Gehen Sie wie in Teilaufgabe 5.b vor, wobei der (Befehl: `grenz(I, 'n')`) anzuwenden ist.

- Welcher Zusammenhang besteht nun zwischen den (geschätzten) statistischen Kenngrößen Erwartungswert bzw. Varianz der Zufallsvariablen Y , X_i ?

Aufgabe 5: Unkorrelierte/Korrelierte Prozesse

Gegeben ist ein stationärer Zufallsprozess $\{x(k)\}$ dessen Zufallsvariablen X_k folgende Eigenschaften besitzen:

- gaußverteilt mit $E\{X_k\} = 0$ und $\sigma^2 = 1$
- unkorreliert (da gaußverteilt: statistisch unabhängig)

- a)
- Skizzieren Sie die Autokovarianzfunktion $c_{xx}(\kappa)$.
 - Skizzieren Sie das Leistungsdichtespektrum $S_{xx}(e^{j\Omega})$ im Intervall $[-\pi, \pi[$.
 - Geben Sie drei Möglichkeiten zur Berechnung der Leistung an. Welche Leistung besitzt der hier vorgegebene Prozess?

b) Der Prozess $\{x(k)\}$ wird auf ein System gegeben, dessen Amplitudengang durch den Ausdruck

$$|H(e^{j\Omega})| = \begin{cases} 0.5 & \Omega_g/2 \leq |\Omega| < \Omega_g \\ 1 & |\Omega| < \Omega_g/2, \\ 0 & \text{sonst} \end{cases} \quad (3.72)$$

bestimmt ist.

- Skizzieren Sie den Amplitudengang des Systems.
- Skizzieren Sie das Leistungsdichtespektrum des Ausgangsprozesses $\{y(k)\}$ mit Hilfe von Glg.(3.61).

- Berechnen Sie die Autokorrelationsfunktion $r_{yy}(\kappa)$.

Hinweis:

$$\text{rect}\left(\frac{\Omega}{\Omega_g}\right) \bullet \text{---} \circ \frac{\Omega_g}{\pi} \text{si}(\Omega_g \kappa)$$

$$\text{rect}(x) = \begin{cases} 1 & |x| < 1 \\ 0 & \text{sonst} \end{cases}$$

c) Der Prozess $\{x(k)\}$ wird nun auf ein System gegeben, dessen Amplitudengang durch den Ausdruck

$$|H(e^{j\Omega})| = \begin{cases} 1 & |\Omega| < \Omega_g \\ 0.5 & |\Omega| = \Omega_g, \\ 0 & \text{sonst} \end{cases} \quad (3.73)$$

bestimmt ist.

- Skizzieren Sie den Amplitudengang des Systems.
- Skizzieren Sie das Leistungsdichtespektrum des Ausgangsprozesses $\{y(k)\}$ mit Hilfe von Glg.(3.61) und geben Sie die Leistung des Ausgangsprozesses unter Verwendung von Glg.(3.58) an.
- Für die folgenden Teilaufgaben werden $N = 10^4$ Werte eines Zufallsprozesses $\{x(k)\}$ erzeugt und auf das System nach Glg.(3.73) gegeben. Die normierte Grenzfrequenz des Systems soll dabei die Werte $\Omega_g = 0.5; 1.5; 2.5$ annehmen.

- Stellen Sie für die beiden Prozesse $\{x(k)\}$ und $\{y(k)\}$ jeweils die Autokovarianzfunktion und das Leistungsdichtespektrum sowie den Amplitudengang des Systems im Intervall $[-\pi \quad \pi[$ dar.

Befehl:

`noise(Ω_g);`

- Bestimmen Sie die jeweiligen geschätzten quadratischen Mittelwerte (Leistung) der Prozesse und vergleichen Sie die Ergebnisse mit den analytisch berechneten.
- Erklären Sie sich anschaulich den Zusammenhang zwischen den Leistungsdichtespektren und dem Amplitudengang des Systems.
- Betrachten Sie die Autokovarianzfolge des Ausgangsprozesses. Welchen typischen Verlauf besitzt die Folge? Erklären Sie anschaulich den Verlauf der Folge (Hinweis: Der Eingangsprozess ist mittelwertfrei und die Faltung zweier si-Folgen ist wieder eine si-Folge)
- Betrachten Sie wiederum die Autokovarianzfolgen der Prozesse. Welche Aussage kann dabei über das Korrelationsverhalten des Ausgangsprozesses bei Veränderung der Grenzfrequenz Ω_g getroffen werden?

Zusatzaufgabe 6*: Systemidentifikation

Hinweis: Wenn noch etwas Zeit ist, sollte diese Aufgabe bearbeitet werden.

Zur Identifikation eines LTI-Systems wird ein stationärer und mittelwertfreier Eingangsprozess $\{x(k)\}$ mit der Autokorrelationsfunktion

$$r_{xx}(\kappa) = \begin{cases} P_{xx} & \kappa = 0 \\ 0 & \text{sonst} \end{cases} \quad (3.74)$$

verwendet.

a) Mit Hilfe von $\{x(k)\}$ soll die Impulsantwort $h(k)$ des Systems ermittelt werden. Welche Möglichkeit besteht prinzipiell, und erfüllt der hier verwendete Eingangsprozess die notwendigen Voraussetzungen?

Geben Sie einen mathematischen Ausdruck zur Ermittlung der Impulsantwort an.

b) In der Folge soll das System betrachtet werden, welches mit dem Befehl `systemid(h)`; aufgerufen werden kann. Hinweise zu `systemid(h)`;: $\mathbf{h}=[h(1) \ h(2) \ h(3) \ \dots]$; für die Werte der Impulsantwort, z.B. $\mathbf{h}=[1 \ 1 \ 1]$. Sollen nur Einsen in der Impulsantwort enthalten sein, lässt sich auch `h=ones(1, N)`; verwenden.

- Bestimmen Sie die Leistung P_{xx} des verwendeten Eingangsprozesses. Bestimmen Sie außerdem den Bereich $[-\kappa_p \ \kappa_p]$, in dem die Autokorrelationsfunktion die Form eines Dirac-Impulses annimmt.
- Testen Sie die Möglichkeit zur Systemidentifikation mit Hilfe der dargestellten Autokorrelationsfunktion

$$r_{xx}(\kappa) = \begin{cases} P_{xx} & \kappa = 0 \\ 0 & -\kappa_p \leq \kappa \leq \kappa_p \\ \text{beliebig} & \text{sonst} \end{cases} \quad (3.75)$$

indem Sie unterschiedlich lange Impulsantworten (Vektoren \mathbf{h}) vorgeben.

- Welche Länge darf die Impulsantwort maximal besitzen, wenn der hier vorgegebene Eingangsprozess verwendet wird?

MATLAB-Funktionen

figure

Die Funktion `figure` öffnet ein neues Graphikfenster.

grenz

Die Funktion `grenz(I, aus)` berechnet die Summe von I statistisch unabhängigen Zufallsvariablen, die alle gleichverteilt sind und den Erwartungswert $E\{X_i\} = 0$ und die Varianz $\sigma_2 = 1/12 \ \forall \ i = 1 \dots I$ besitzen. Ausgegeben wird der Mittelwert und die geschätzte Varianz für $N = 10^4$ Stichproben. Weiterhin stellt sie die Verteilungsdichtefunktionen graphisch dar.

Eingabedaten:

- I : Anzahl der Zufallsvariablen
- `aus`:

– 'n':

$$Y = \frac{1}{\sqrt{I}} \sum_{i=1}^I X_i$$

– 'u':

$$Y = \sum_{i=1}^I X_i$$

hist

Die Funktion `hist(x,bin)` stellt das Histogramm für die Elemente in Vektor `x` dar. Dabei wird der Wertebereich in `bin`-Intervalle unterteilt.

- `x`: Datenvektor
- `bin`: Anzahl der Teilintervalle

noise

Die Funktion `noise(Ω_g)` berechnet die Autokovarianzfolge und das Leistungsdichtespektrum eines weißen Eingangs- und Ausgangsprozesses eines Tiefpasses der Grenzfrequenz Ω_g für $N = 10000$ Werte. Graphisch dargestellt werden die Autokovarianzfolgen und Spektren sowie der Frequenzgang des Systems. Weiterhin werden die Leistungen der Prozesse berechnet und ausgegeben.
Eingabedaten:

- Ω_g : Grenzfrequenz des TP-Filters

rand, randn

Die Funktion `rand(N)` berechnet eine $N \times N$ -Matrix deren Werte im Intervall $[0,1[$ gleichverteilt und mittelwertfrei sind.

Die Funktion `randn(N)` berechnet eine $N \times N$ -Matrix deren Werte normalverteilt sind.

- N : Skalar

roots

Die Funktion `roots(a)` berechnet die Wurzeln (Nullstellen) des Polynoms $a_1s^n + \dots + a_n s + a_{n+1}$.
Eingabedaten:

- a : Vektor $a = [a_1, a_2, \dots, a_{n+1}]$

systemid

Die Funktion `systemid(b)` ermittelt die Impulsantwort eines Systems mit Hilfe eines (annähernd) weißen Eingangsprozesses durch Bilden der Kreuzkorrelationsfolge zwischen Eingangsprozess und Ausgangsprozess. Als Eingangsprozess dient hier eine Datenfolge von 26 Werten (GSM-Datenfolge). Graphisch ausgegeben werden die Eingangskorrelationsfolge und die Kreuzkorrelationsfolge.
Eingabedaten:

- b : Vektor b dessen Elemente die Werte der Impulsantwort sind.

Anmerkungen

- Alle Funktionen werden mittels der Hilfsfunktion `help` näher erläutert und können unter Verwendung des Befehls `type` eingesehen werden.
- Nach jedem Aufruf eines MATLAB-Befehls ist der Befehl `clear all` durchzuführen.

Octave

Damit die Funktionen in Octave funktionieren, müssen folgende Pakete installiert werden:

- `pkg install -forge control`
- `pkg install -forge signal`

Genutzt werden diese über:

- `pkg load control`
- `pkg load signal`

Kapitel 4

Codierung

Matthias Hummert

NW1, Raum N2390, Tel.: 0421/218-62419

E-Mail: hummert@ant.uni-bremen.de

4.1 Einführung

In diesem Versuch sollen Beispiele für die in der Vorlesung “Grundlagen der Nachrichtentechnik” vorgestellten Verfahren der Quellen- und Kanalcodierung untersucht werden. Dazu soll eine einfache Übertragungsstrecke aufgebaut und die einzelnen Teile analysiert werden.

Für die Durchführung dieses Versuchs wird die Kenntnis des behandelten Stoffes der Vorlesung “Grundlagen der Nachrichtentechnik” vorausgesetzt. Die Folien können auf Stud.IP heruntergeladen werden. Einzelne für diesen Versuch wichtige Themen werden im Folgenden noch einmal kurz erläutert.

4.1.1 Theoretische Grundlagen

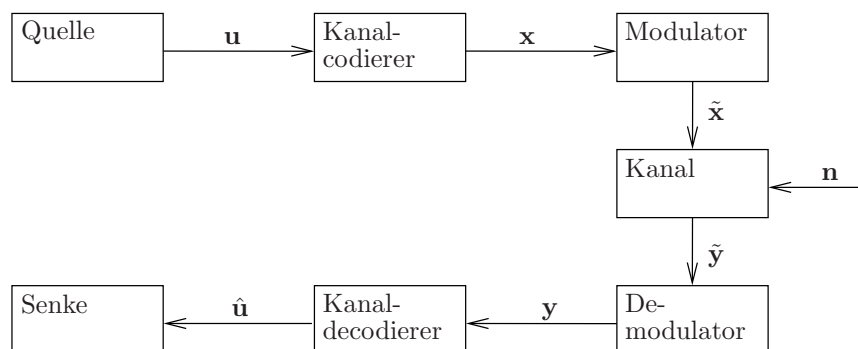


Abbildung 4.1: Übertragungsstrecke bestehend aus äquivalentem Basisbandmodell

In diesem Versuch soll exemplarisch eine einfache Übertragungsstrecke erstellt und simuliert werden. Hier wird als Model angenommen, dass die Übertragungsstrecke aus einzelnen Blöcken besteht, die jeweils das eingehende Signal verändern und das resultierende Signal an den nächsten Block weiter geben (siehe Bild

4.1). In diesem Model ist der Kanal durch die physikalischen Randbedingungen gegeben. Sein genaues Verhalten ist nicht bekannt und kann nur stochastisch modelliert werden. Bei den restlichen Blöcken ist das jeweilige deterministisches Verhalten vorher bekannt. Die wichtigen Grundlagen der einzelnen Blöcke der Übertragungstrecke werden nachfolgend kurz erläutert.

Quellencodierung

In der Nachrichtentechnik wird mit “Quelle” der Ausgangspunkt einer Übertragung und mit “Senke” der Endpunkt dieser Übertragung bezeichnet. Im Allgemeinen liegen die zu übertragenden Informationen als physikalische Signale (Sprache, Sensorwerte, etc) vor. Diese müssen vor der Übertragung erst in eine in Amplitude und Zeit diskrete Darstellung überführt werden (“Quantisierung” und “Abtastung”). Diesen quantisierten Werten wird dann jeweils eine binäre Datensequenz zugewiesen. Diesen Vorgang bezeichnet man auch als “Mapping”. Um die Ressourcen der Übertragungstrecke möglichst effizient zu nutzen “mappt” die Quellencodierung die gegebenen Informationen der Quelle auf möglichst wenigen Bits.

Ein Beispiel für eine Quellencodierung ist die “Huffman Codierung”. Ziel der Huffman Codierung ist es den oft auftretenden Signalwerten kurze Informationsworte und den selten auftretenden Signalwerten lange Informationsworte zuzuweisen, und somit im Mittel möglichst wenig Bits übertragen zu müssen. Hierfür werden die quantisierten Signalwerte entsprechend ihrer Wahrscheinlichkeit sortiert und schrittweise die beiden mit der geringsten Wahrscheinlichkeit zu einem neuen Symbol zusammengefasst. Dieses neue Symbol erhält dann als Wahrscheinlichkeit die Summe der beiden Einzelwahrscheinlichkeiten. Sobald alle Signalwerte zusammengefasst wurden, kann man das Mapping der einzelnen Signalwerte aus der Zuordnung zu den zusammengefassten Symbolen ablesen.

Für die anderen Blöcke der Übertragungstrecke wird im Nachfolgenden davon ausgegangen, dass die Quellencodierung die Daten in Form von Informationswörtern \mathbf{u} mit konstanter Länge k ausgibt.

Kanalcodierung

Das Ziel der Kanalcodierung ist es, die Informationswörter \mathbf{u} der Quelle gegen Fehler auf dem Kanal zu schützen. Dies wird durch Hinzufügen weiterer Symbole, welche Redundanz beinhalten, erreicht. Das verwendete Verfahren zur Kanalcodierung (auch “Code” genannt) legt fest, wie aus dem Informationswort \mathbf{u} das Codewort \mathbf{x} erzeugt wird. Bei einem sogenannten “systematischen Encoder” sind alle Symbole des Informationsworts \mathbf{u} direkt im Codewort \mathbf{x} enthalten, es wird hier nur Redundanz hinzugefügt und eventuell die Reihenfolge der Symbole geändert.

Eine einfache Form der Kanalcodierung ist der Lineare Blockcode. Ein Linearer Blockcode wird durch die Zuordnung der Informationsworte \mathbf{u} der Länge k zu den Codewörtern \mathbf{x} der Länge n definiert. Diese Zuordnung lässt sich mit der sogenannten “Generator Matrix” $\mathbf{G} \in \{0, 1\}^{k \times n}$ beschreiben, welche aus k verschiedenen gültigen Codewörtern \mathbf{x} besteht. Mit dieser Generator Matrix \mathbf{G} kann die Codierung eines Informationsworts $\mathbf{u} = [u_0, u_1, \dots, u_{k-1}]$, mit $u_i \in \{0, 1\}$, als Multiplikation dargestellt werden:

$$\mathbf{x} = \mathbf{u} \cdot \mathbf{G}_{\text{mod}2}. \quad (4.1)$$

Somit sind bei diesem Code alle gültigen Codeworte $\mathbf{x} = [x_0, x_1, \dots, x_{n-1}]$, mit $x_i \in \{0, 1\}$, entweder Zeilen der Generator Matrix \mathbf{G} oder ergeben sich aus der binären Addition von Zeilen der Generator Matrix \mathbf{G} .

Modulation

Während der Übertragung über einen Kanal tritt im Allgemeinen Rauschen auf, welches die gesendeten Werte verändert. Um die Detektion der gesendeten Werte am Empfänger zu verbessern, werden die binären Werte vor dem Senden moduliert.

Die einfachste Form der Modulation ist das sogenannte “Binary Phase-Shift Keying” (BPSK). Hierbei wird jedes binäre Symbol (oder Bit) auf das entsprechende Symbol im Signalraum abgebildet. Diese Symbole unterscheiden sich in ihrer Phase, aber nicht in ihrer Amplitude. Hierbei werden meist die Symbole “1” und “-1” verwendet. Am Empfänger müssen die verrauschten Empfangswerte \tilde{y} wieder in Bits umgewandelt oder demoduliert werden. Dies wird mit einer sogenannten “hard-decision” gemacht. Hierbei wird überprüft, ob der Empfangswert \tilde{y}_i über- oder unterhalb einer Entscheidungsschwelle liegt. Die ideale Entscheidungsschwelle im Sinne geringster Fehlerwahrscheinlichkeit hängt von der Auftretenswahrscheinlichkeit der einzelnen Werte ab. Im Falle einer BPSK und falls die Quelle gleichwahrscheinliche binäre Werte liefert, dann liegt die ideale Entscheidungsschwelle bei 0.

Ein Beispiel für ein höherstufiges Modulationsverfahren ist das “Quadrature Phase-Shift Keying” (QPSK). Höherstufige Modulationsverfahren zeichnen sich dadurch aus, dass sie mehrere Bits eines Codeworts \mathbf{x} auf ein Symbol im Signalraum abbilden. Im Fall der QPSK werden 2 Bits auf eines von 4 Symbolen im Signalraum abgebildet. Die verwendeten Symbole bei der QPSK unterscheiden sich wie bei der BPSK nur in ihrer Phase, aber nicht in ihrer Amplitude, daher muss hier die komplexe Raumrichtung genutzt werden. Die QPSK kann wie zwei überlagerte BPSK betrachtet werden, eine in reelle und eine in imaginäre Raumrichtung. Die Demodulation wird daher ähnlich wie bei der BPSK durchgeführt, wobei hier Real- und Imaginärteil getrennt betrachtet werden und somit eine hard-decision für jedes übertragene Bit getrennt durchgeführt wird.

Kanal

Als Modell für den Kanal wird oft der “Additive White Gaussian Noise” (AWGN) Kanal verwendet. Bei diesem Kanal wird zu den gesendeten Werten $\tilde{\mathbf{x}}$ der Rauschterm \mathbf{n} addiert. Der Rauschterm \mathbf{n} ist beim AWGN-Kanal mittelwertfrei und Gauss verteilt. Die Varianz des Rauschterms \mathbf{n} bestimmt, wie stark der Einfluss des Rauschens auf das empfangene Signal ist.

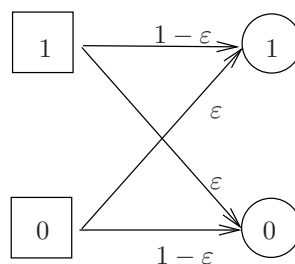


Abbildung 4.2: Binary Symmetric Channel

Aus Sicht des Kanaldecoders, können die Blöcke Modulator, Kanal und Demodulator zusammengefasst und zusammen als binärer Kanal betrachtet werden. Hierfür wird das Modell des “Binary Symmetric Channel” (BSC) verwendet (siehe Bild 4.2). Bei einem BSC wird jedes übertragene Bit einzeln mit einer Fehlerwahrscheinlichkeit von ε gestört und damit invertiert empfangen. Somit wird jedes Bit mit einer Wahrscheinlichkeit von $1 - \varepsilon$ korrekt empfangen.

Kanaldecodierung und Fehlerkorrektur

Bei der Übertragung über einen BSC wird das übertragene Codewort \mathbf{x} am Empfänger als Empfangswort $\mathbf{y} = \mathbf{x} + \mathbf{e}$ empfangen, wobei \mathbf{e} hier ein binärer Fehlervektor ist. Der Fehlervektor wird durch das Rauschen auf dem Kanal bestimmt und daher ist der Wert eines konkreten Fehlervektors im Allgemeinen nicht bekannt. Mit der sogenannten “Parity-Check Matrix” $\mathbf{H} \in \{0, 1\}^{(n-k) \times n}$ kann überprüft werden, ob ein ungestörtes Codewort vorliegt. Die Parity-Check Matrix \mathbf{H} ist definiert durch:

$$\mathbf{G} \cdot \mathbf{H}^T = \mathbf{0}. \quad (4.2)$$

Somit erfüllen alle gültigen Codeworte \mathbf{x} , die mit der Generator Matrix \mathbf{G} erzeugt wurden, folgende Gleichung:

$$\mathbf{x} \cdot \mathbf{H}^T = \mathbf{0}. \quad (4.3)$$

Falls für ein empfangenes Wort $\mathbf{y} \cdot \mathbf{H}^T = \mathbf{0}$ gilt, so handelt es sich bei \mathbf{y} um ein gültiges Codewort. Dieses Codewort \mathbf{y} muss nicht zwingend das gesendete Codewort \mathbf{x} sein, da der Fehlervektor \mathbf{e} selber ein gültiges Codewort sein kann und somit Empfangsvektor $\mathbf{y} = \mathbf{x} + \mathbf{e}$ ein anderes gültiges Codewort ist. Falls $\mathbf{y} \cdot \mathbf{H}^T \neq \mathbf{0}$ gilt, so ist das empfangene Wort mit Fehlervektor $\mathbf{e} \neq \mathbf{0}$ gestört und muss vor dem Decodieren erst korrigiert werden. Welcher Fehler im einzelnen Fall vorliegt wird über das “Syndrom” \mathbf{s} bestimmt, welches wie folgt definiert ist:

$$\mathbf{s} = \mathbf{y} \cdot \mathbf{H}^T \neq \mathbf{0} \text{ mit } \mathbf{s} \in \{0, 1\}^{n-k}. \quad (4.4)$$

Aufgrund der Linearität des Codes und Gleichung (4.3), folgt mit $\mathbf{y} = \mathbf{x} + \mathbf{e}$:

$$\mathbf{s} = \mathbf{y} \cdot \mathbf{H}^T = \underbrace{\mathbf{x} \cdot \mathbf{H}^T}_{= 0} + \mathbf{e} \cdot \mathbf{H}^T = \mathbf{e} \cdot \mathbf{H}^T. \quad (4.5)$$

Daher ist das Syndrom \mathbf{s} für Empfangsvektor \mathbf{y} alleine durch den Fehlervektor \mathbf{e} bestimmt und jedem Fehlervektor \mathbf{e} ist eindeutig ein Syndrom \mathbf{s} zugeordnet. Da im Allgemeinen deutlich mehr mögliche Fehlervektoren \mathbf{e} existieren, als es mögliche Syndrome \mathbf{s} gibt, kann von einem Syndrom \mathbf{s} nicht eindeutig auf den tatsächlich aufgetretenen Fehlervektor \mathbf{e} geschlossen werden. Wegen der nicht eindeutigen Zuordnung wird im Allgemeinen davon ausgegangen, dass bei einem gegebenen Syndrom \mathbf{s} der wahrscheinlichste passende Fehlervektor \mathbf{e} aufgetreten ist. Der wahrscheinlichste passende Fehlervektor ist der Fehlervektor mit dem geringsten Gewicht. Der so ermittelte Fehlervektor \mathbf{e} ist genau dann der tatsächliche Fehlervektor \mathbf{e} , wenn ein korrigierbarer Fehler aufgetreten ist. Ein Fehler ist dann korrigierbar, wenn gilt:

$$|\mathbf{e}| \leq \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor, \quad (4.6)$$

wobei d_{\min} die minimale “Hamming Distanz” des Codes ist. Mit dem so ermittelten Fehlervektor \mathbf{e} kann mit $\hat{\mathbf{x}} = \mathbf{y} - \mathbf{e}$ das korrigierte Codewort $\hat{\mathbf{x}}$ ermittelt werden.

Nachdem durch die Fehlerkorrektur sicher gestellt ist, dass ein gültiges Codewort vorliegt, kann dieses Codewort anschließend decodiert werden. Im Allgemeinen muss hierfür festgestellt werden, welche Linearkombination der Zeilen der Generator Matrix \mathbf{G} das Codewort \mathbf{x} ergibt. Dies ist bei jedem gültigen Codewort \mathbf{x} eindeutig. Falls ein sogenannter “systematischer Encoder” vorliegt, dann sind die Symbole des Informationsworts \mathbf{u} direkt im Codewort \mathbf{x} enthalten, und können daher direkt an den entsprechenden Stellen des Codeworts \mathbf{x} abgelesen werden.

Bitfehlerrate

Die ‘‘Bitfehlerrate’’ oder ‘‘Bit Error Rate’’ (BER) ist ein hufig verwendetes Ma zur Bewertung der Fehler-sicherheit einer bertragungsstrecke, insbesondere des darin verwendeten Kanalcodes. Die Bitfehlerrate gibt die mittlere Wahrscheinlichkeit an, dass ein Bit im Informationswort \mathbf{u} nach der bertragung ber den Kanal, der Demodulation und der Decodierung fehlerhaft ist.

Bei einem AWGN Kanal wird die Bitfehlerrate in Abhangigkeit des Rauschens auf dem Kanal gemessen. Das Ma E_S/N_0 gibt das Verhalt-nis der mittleren Energie pro bertragenem Symbol zur mittleren Energie des Rauschens auf dem Kanal an. Man spricht auch allgemeiner vom ‘‘Signal-to-Noise-Ratio’’ (SNR). Das E_S/N_0 wird meist in Dezibel (dB) angegeben. Hierbei gilt folgende Umrechnung:

$$E_{\text{dB}} = 10 \cdot \log_{10}(E_{\text{lin}}). \quad (4.7)$$

Bei einem AWGN Kanal gilt fur die mittlere Energie des Rauschens:

$$N_0 = \sigma_N^2. \quad (4.8)$$

Daher lasst sich fur ein gegebenes E_S/N_0 und E_S die entsprechende Varianz des Rauschens bestimmen.

In bestimmten Fallen wird statt dem E_S/N_0 das Ma E_b/N_0 verwendet, welches die Energie pro Bit anstatt der Energie pro Symbol im Verhalt-nis zur Rauschenergie misst. Dies wird meist in Fallen gemacht, bei denen ein Symbol mehrere Bits darstellt. Falls jedes Symbol jeweils ein Bit darstellt, so sind E_S/N_0 und E_b/N_0 identisch. Es gilt fur M -stufige Modulation der Zusammenhang:

$$\frac{E_b}{N_0} = \frac{1}{\text{ld}(M)} \cdot \frac{E_S}{N_0}, \quad (4.9)$$

wobei hier M die Anzahl Signalraumpunkte ist, die durch die Symbole der Modulation beschrieben werden.

Bei einem BSC wird die Bitfehlerrate in Abhangigkeit der Fehlerwahrscheinlichkeit ε des BSC gemessen. Diese Sichtweise ist aquivalent zu der vorherigen Betrachtung, da die Fehlerwahrscheinlichkeit ε des BSC von dem zugrundeliegenden physikalische Kanal abhangt. Die Bitfehlerrate ist fur einen AWGN Kanal und bei einer BPSK Modulation gegeben durch:

$$P_{b|BPSK} = \frac{1}{2} \text{erfc} \left(\sqrt{\frac{E_S}{N_0}} \right), \quad (4.10)$$

wobei erfc die ‘‘komplementare Fehlerfunktion’’ ist, gegeben durch:

$$\text{erfc}(x) = 1 - \text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt. \quad (4.11)$$

Desweiteren gilt fur die Bitfehlerrate einer QPSK uber einen AWGN Kanal:

$$P_{b|QPSK} = \frac{1}{2} \text{erfc} \left(\sqrt{\frac{E_b}{N_0}} \right). \quad (4.12)$$

4.2 Vorbereitungsarbeiten

4.2.1 Quellencodierung

Gegeben sei ein analoges Eingangssignal das mittels Puls-Code-Modulation in Zeit und Amplitude diskretisiert wird. Es werden hierbei sechs verschiedene Quantisierungsstufen q_1, \dots, q_6 fur die Amplituden

verwendet. Jede dieser Quantisierungsstufen tritt innerhalb des analogen Eingangssignals mit einer bekannten Wahrscheinlichkeit auf:

- $\Pr\{q_1\} = 0.09$
- $\Pr\{q_2\} = 0.11$
- $\Pr\{q_3\} = 0.26$
- $\Pr\{q_4\} = 0.3$
- $\Pr\{q_5\} = 0.14$
- $\Pr\{q_6\} = 0.1$

a) Führen Sie eine Quellencodierung nach Huffman (siehe Einleitung) entsprechend der gegebenen Wahrscheinlichkeiten $\Pr\{q_i\}$ durch.

b) Berechnen Sie für diesen Huffman Code die durchschnittliche Länge eines Codewortes und die Entropie der Quelle (siehe Skript).

4.2.2 Fehlerkorrektur

Gegeben sei ein systematischer $(7, 4, 3)_2$ -Hamming-Code, mit der folgenden Generator-Matrix:

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \quad (4.13)$$

und der dazu gehörigen Parity-Check Matrix:

$$\mathbf{H} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \quad (4.14)$$

a) Berechnen Sie für alle möglichen binären Fehlervektoren \mathbf{e} mit Gewicht 1 (dies wird im weiteren als 1-Bit-Fehler bezeichnet), d.h. $|\mathbf{e}| = 1$, das zugehörige Syndrom \mathbf{s} nach (4.4). Erstellen Sie die Syndromtabelle, welche jedem dieser Fehlervektoren \mathbf{e} das entsprechende Syndrom \mathbf{s} zuordnet.

b) Was fällt bei den Syndromen \mathbf{s} auf?

4.3 Versuchsdurchführung

4.3.1 Kanalcodierung

Gegeben sei eine binäre Quelle, welche Informationsworte \mathbf{u} der Länge $k = 4$ produziert. Die binären Werte "0" und "1" werden hier als gleichwahrscheinlich angenommen. Um ein Informationswort aus k Symbolen einer solchen Datenquelle in MATLAB zu erzeugen, nutzen Sie den folgenden Befehl: `round(rand(1, k))`.

Die Informationswörter der Quelle werden zum Schutz gegen Fehler mit dem in Abschnitt 4.2.2 angegebenen $(7, 4, 3)_2$ -Hamming-Code codiert.

a) Erstellen Sie in MATLAB eine Funktion, welche die Kanalcodierung eines Informationsworts \mathbf{u} mittels des gegebenen $(7, 4, 3)_2$ -Hamming-Codes durchführt. Erweitern Sie hierzu den gegebenen Funktionsrumpf `encoder`, der ein Informationswort \mathbf{u} als Eingang erhält und das zugehörige Codewort \mathbf{x} als Ausgang ausgibt.

Hinweis: Beachten Sie, dass die Codierung eine binäre Operation ist. Da in MATLAB eine Addition oder Multiplikation über reellen Zahlen berechnet wird, müssen Sie bei jeder Addition oder Multiplikation binärer Werte die Modulo-Operation auf das Ergebnis anwenden. Verwenden Sie hierfür in MATLAB den Befehl: `mod(x, 2)`.

b) Erweitern Sie das MATLAB-Skript `aufgabe_6_3_1` so, dass ein Informationswort \mathbf{u} erzeugt wird, welches anschliessend mit `encoder` codiert wird. Überprüfen Sie die so entstehenden Codeworte \mathbf{x} auf Korrektheit.

4.3.2 Decodierung und Fehlerkorrektur

Das Codewort \mathbf{x} wird nun über einen BSC übertragen und dabei mit einem binären Fehlervektor \mathbf{e} gestört. Zuerst soll angenommen werden, dass maximal nur ein einzelnes Bit innerhalb eines Codewortes \mathbf{x} gestört wird. Da der verwendete $(7, 4, 3)_2$ -Hamming-Code eine minimale Hamming-Distanz von $d_{\min} = 3$ hat, sind alle 1-Bit-Fehler korrigierbar.

a) Schreiben Sie eine Funktion in MATLAB, die zuerst das Syndrom \mathbf{s} für ein empfangenes Wort $\mathbf{y} = \mathbf{x} + \mathbf{e}$ bestimmt, dann mit Hilfe des Syndroms \mathbf{s} den Fehler korrigiert und anschließend das korrigierte Codewort $\hat{\mathbf{x}}$ decodiert. Erweitern Sie hierzu den gegebenen Funktionsrumpf `decoder`, der das empfangene Wort \mathbf{y} als Eingang erhält und das decodierte Informationswort $\hat{\mathbf{u}}$ zurück gibt. Beachten Sie, wie im vorherigen Abschnitt, die Modulo-Operation bei jeder Multiplikation oder Addition von binären Werten.

Hinweis: Beachten Sie bei der Decodierung, dass der verwendete Hamming-Code systematisch ist.

b) Erweitern Sie das MATLAB Skript `aufgabe_6_3_2` so, dass ein wie in `aufgabe_6_3_1` gegebenes Codewort \mathbf{x} mit einem 1-Bit-Fehlervektor \mathbf{e} gestört wird und anschliessend mit `decoder` decodiert wird. Zeigen Sie, dass alle 1-Bit Fehler korrekt decodiert werden.

c) Welches Verhalten des Decoder ist bei zwei oder mehr Bit-Fehlern zu erwarten?

4.3.3 Bitfehlerrate über BSC

Für eine vereinfachte Übertragungsstrecke bestehend aus einer Quelle, einem Kanalcoder, einem BSC, einem Kanaldecoder und einer Senke soll die Bitfehlerrate in Abhängigkeit von der Fehlerwahrscheinlichkeit ε des BSCs bestimmt werden.

a) Ermitteln Sie die Bitfehlerrate nach der Decodierung für die Übertragung eines Codewortes \mathbf{x} über einen BSC, wobei die Fehlerwahrscheinlichkeit ε im Bereich von 0 bis 0.5 liegt. Erstellen Sie dazu im MATLAB Skript `aufgabe_6_3_3` die Übertragungsstrecke und nutzen Sie für den BSC in MATLAB die gegebene Funktion `BSC`. Um die Ergebnisse mit MATLAB graphisch darzustellen, nutzen Sie den Befehl `semilogy(ε , BER)`.

b) Welche Bitfehlerrate ist zu erwarten, wenn ohne Kanalcodierung über den beschriebenen BSC übertragen wird? Zeigen Sie diese Bitfehlerrate im Vergleich zu der aus Teil a) in einer Grafik.

4.3.4 Bitfehlerrate mit BPSK über AWGN Kanal

Das Codewort \mathbf{x} wird nun mit einer BPSK moduliert und anschließend über einen AWGN-Kanal übertragen. Verwenden Sie folgende Zuordnung für die BPSK Modulation:

$$\begin{aligned}x_i = 0 &\rightarrow \tilde{x}_i = 1 \\x_i = 1 &\rightarrow \tilde{x}_i = -1\end{aligned}$$

a) Schreiben Sie eine Funktion in MATLAB, welche ein kanalcodiertes Codewort \mathbf{x} auf modulierte Symbole $\tilde{\mathbf{x}}$ abbildet. Erweitern Sie hierzu den gegebenen Funktionsrumpf `modulator_BPSK`.

Das modulierte Codewort $\tilde{\mathbf{x}}$ wird nun über einen AWGN-Kanal übertragen. Um den Rauschterm $\mathbf{n} \in \mathcal{N}(0, \sigma^2)$ mit m Elementen in MATLAB zu erzeugen, nutzen sie den Befehl: $\frac{\sigma}{\sqrt{2}} * \text{randn}(1, m)$.

b) Erweitern Sie in MATLAB den Funktionsrumpf `demodulator_BPSK` so, dass diese verrauschte Empfangssymbole $\tilde{\mathbf{y}}$ als Eingang erhält und das demodulierte binäre Empfangswort \mathbf{y} als Ausgang ausgibt. Nutzen Sie für die hard-decision in MATLAB die Signum Funktion: `sign(x)`.

c) Erweitern Sie das MATLAB Skript `aufgabe_6_3_4` um die Übertragungsstrecke bestehend aus einer binärer Quelle, einem Kanalencoder, einem Modulator, einem Kanal, einem Demodulator und einem Kanaldecoder (die Senke wird nicht explizit implementiert). Nutzen Sie hierfür die in Aufgabenteilen 4.3.1, 4.3.2 und 4.3.4 a), b) erstellten Funktionen und verwendeten Befehle.

d) Ermitteln Sie mit der in `aufgabe_6_3_4` erstellten Übertragungsstrecke die Bitfehlerrate nach der Decodierung für ein E_S/N_0 im Bereich von 0dB bis 10dB (Umrechnung von dB beachten!). Um die Ergebnisse mit MATLAB graphisch darzustellen, nutzen Sie den Befehl `semilogy(E_S/N_0, BER)`.

e) Messen Sie in demselben E_S/N_0 Bereich, wie in Teil d), die Bitfehlerrate für den Fall, dass keine Kanalcodierung und Kanaldecodierung durchgeführt wird, also uncodierte modulierte Wörter übertragen werden. Beachten Sie, dass sich hierdurch die Wortlänge am Eingang des Modulators ändert.

f) Vergleichen Sie die simulierten Ergebnisse der Bitfehlerrate für den codierten und uncodierten Fall mit der Bitfehlerrate für einen AWGN-Kanal (siehe Einführung). Nutzen Sie die MATLAB Funktion `erfc(x)` zur Berechnung der Bitfehlerrate des AWGN Kanals. Um mehrere Fehlerkurven in einem Graph darzustellen, nutzen Sie in MATLAB den Befehl `semilogy(E_S/N_0, BER_1, E_S/N_0, BER_2, ...)`.

4.3.5 Bitfehlerrate mit QPSK über AWGN Kanal

Anstatt mit einer BPSK Modulation wird nun mit einer QPSK moduliert. Für diesen Abschnitt wird folgende Zuordnung für die QPSK Modulation verwendet:

$$\begin{aligned}x_i = 0, x_{i+1} = 0 &\rightarrow \tilde{x}_i = \frac{1}{\sqrt{2}} \cdot (1 + j) \\x_i = 0, x_{i+1} = 1 &\rightarrow \tilde{x}_i = \frac{1}{\sqrt{2}} \cdot (1 - j) \\x_i = 1, x_{i+1} = 0 &\rightarrow \tilde{x}_i = \frac{1}{\sqrt{2}} \cdot (-1 + j) \\x_i = 1, x_{i+1} = 1 &\rightarrow \tilde{x}_i = \frac{1}{\sqrt{2}} \cdot (-1 - j)\end{aligned}$$

Da bei einer QPSK auch die komplexe Raumrichtung im Symbolraum genutzt wird, muss auch ein komplexer Rauschterm betrachtet werden. Bei komplexem weißem Gaußschem Rauschen wird davon

ausgegangen, dass das Rauschen aus einer reellen und komplexen Komponente mit gleicher Varianz besteht, die sich additiv überlagern. Nutzen Sie für den komplexen Rauschterm \mathbf{n} aus $\lceil \frac{n}{2} \rceil$ Elementen in MATLAB den Befehl $\frac{\sigma}{\sqrt{2}} * (\text{randn}(1, \lceil \frac{n}{2} \rceil) + 1i * \text{randn}(1, \lceil \frac{n}{2} \rceil))$.

Hinweis: Damit das Codewort \mathbf{x} mit QPSK moduliert werden kann, nehmen Sie an, dass auf das letzte Symbol in \mathbf{x} immer ein "0" Symbol folgt (zero-padding) und modulieren Sie dies zusammen mit dem letzten Symbol in \mathbf{x} .

a) Erweitern Sie, wie in Abschnitt 4.3.4, in MATLAB die gegebenen Funktionrümpfe `modulator_QPSK` und `demodulator_QPSK` für die Modulation, beziehungsweise Demodulation, mit einer QPSK. Nutzen Sie bei der Demodulation wieder die Signum Funktion für die hard-decision und führen Sie die hard-decision separat für den Real- und Imaginärteil durch. Den Realteil einer Variable erhalten Sie in MATLAB mit `real(ŷ)`, und den Imaginärteil mit `imag(ŷ)`.

b) Messen Sie, wie in Abschnitt 4.3.4 beschrieben, die Bitfehlerrate dieser Übertragungsstrecke (nutzen Sie hierfür das MATLAB Skript `aufgabe_6_3_5`). Hierbei sollen die Fälle mit und ohne Kanalcodierung, aber jeweils mit QPSK Modulation, gemessen werden. In beiden Fällen ist ein komplexer Rauschterm für den AWGN-Kanal zu verwenden.

c) Vergleichen Sie die Ergebnisse für die QPSK Modulation mit denen der BPSK Modulation aus Abschnitt 4.3.4 und der Bitfehlerrate für einen AWGN-Kanal (siehe Einführung) in einer Grafik. Was fällt hierbei auf und wie ist das zu erklären?

Anhang A

Einleitung zu Matlab

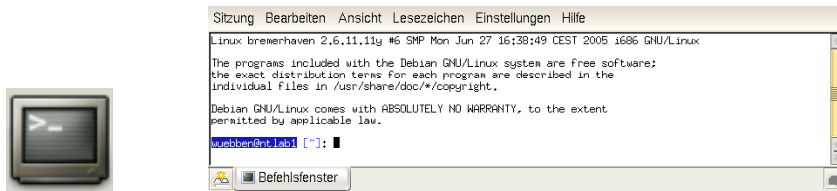
Das Programm Matlab (MATrix-LABoratory) ist ein vielseitiges und doch einfach zu erlernendes Werkzeug zur Erstellung mathematischer Berechnungen. Ausführliche Informationen finden sich unter www.mathworks.de bzw. www.mathworks.com. Weiterhin stehen eine Vielzahl an ausführlichen und detaillierten Einführungen im Internet zur Verfügung, z.B.

- T. Schubiger: Ausführliches Skript der ETH Zürich
www.imrt.ethz.ch/education/tutorials/matlab/grundlagen_P.pdf
- Einführung in Matlab von P. Arbenz, ETH Zürich
<http://people.inf.ethz.ch/arbenz/MatlabKurs/matlabintro.pdf>
- C. Moler: Numerical Computing with MATLAB (Buch mit umfassender Einführung)
www.mathworks.com/moler/
- MATLAB Primer, 3rd edition, Kermit Sigmon
<http://www.math.toronto.edu/mpugh/primer.pdf>

Da das Labor selber unter Linux ausgeführt wird (es ergeben sich dabei keine Unterschiede zwischen Linux- und Windows-Version von Matlab), soll im Folgenden kurz der Aufruf von Matlab vorgestellt werden.

Matlab starten:

Öffnen Sie einen Terminal (Shell) unter Linux durch Klicken auf dieses Symbol. In dem geöffneten Terminal geben Sie `matlab` zum Starten von Matlab ein.



Nach dem Start von Matlab wird ein Fenster wie im rechten Teil von Bild A.1 geöffnet. Das dargestellte Befehls-Fenster (*Command Window*) ist der Ort, an dem Sie Matlab-Befehle eingeben werden. Ein Befehl wird rechts vom Doppelpfeil eingetippt und mit der **Enter**-Taste abgeschlossen. Er wird dann von

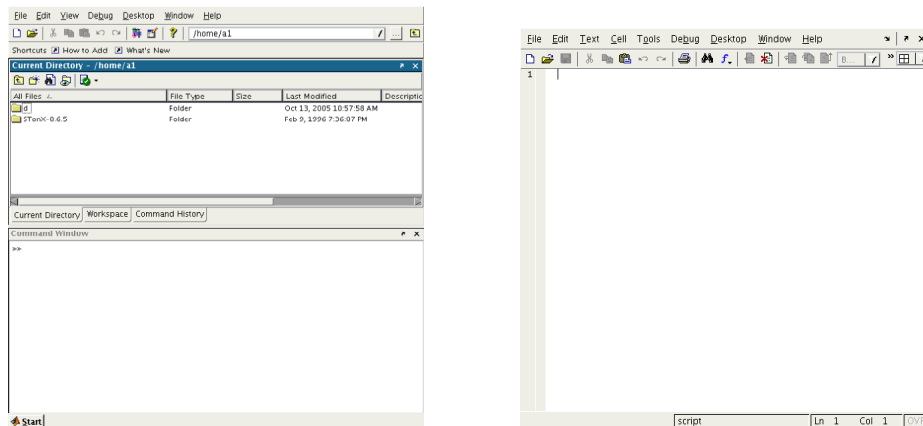


Abbildung A.1: Matlab Workspace und Matlab Editor

Matlab ausgeführt. Eine ganze Reihe von Befehlen können zusammen in einer Datei mit der Endung `.m` abgespeichert werden (siehe Öffnen eines Editors). Solche Dateien werden M-Files genannt. Die Datei wird im Befehls-Fenster mit ihrem Namen (ohne Endung) aufgerufen. Der Benutzer kann die Datei `prog1.m` also mit dem Befehl `prog1` starten. Dazu muss sich die Datei im aktuellen Verzeichnis (*current directory*) befinden, welches in der Befehlsleiste oben ausgewählt wird, oder in einem Verzeichnis des `matlabpath` befinden.

Im oberen Bereich des Fensters sind mehrere Teilfenster in Form von Tags dargestellt. Im *Command History* werden Befehle gespeichert, die Sie bereits ausgeführt haben. Durch Klick auf einen Befehl in diesem Fenster wird der Befehl ins Befehlsfenster kopiert und bei einem Doppelklick erneut ausgeführt. Im Teilfenster *Current Directory* werden die Files des gegenwärtige Verzeichnisses angezeigt und der Arbeitsspeicher von Matlab ist unter *Workspace* aufgeführt. Hierin befinden sich die Variablen, die sie angelegt haben. Durch einen Doppelklick auf den Variablennamen wird ein *Arrayeditor* geöffnet, mit welchem man die Matrixelemente editieren kann.

Matlab bietet vielfältige Hilfen an. Help an der oberen Befehlsleiste eröffnet ein neues Teilfenster, welches sämtliche Hilfsangebote von Matlab auflistet. Neben der Dokumentation einzelner Befehle finden sich hier auch ein Matlab- Einführungskurs (*Getting Started*), ein Benutzer-Handbuch (*User Guide*), Demos, pdf-Files der Dokumentation, und vieles mehr.

Ändern des Verzeichnis:

Geben Sie `cd glab` im Matlab Command Window ein, um in das Verzeichnis `glab` zu wechseln. Hier können Sie alle Dateien speichern, die Sie während der Laborversuche erstellen. Um sich zu vergewissern, in welchem Verzeichnis man sich befindet, kann man `pwd` im Command Window eingeben und erhält als Ergebnis den aktuellen Pfad.

Öffnen des Editors:

Geben Sie `edit` im Command Window ein, um den Editor mit einer unbenannten Datei zu öffnen. In dieser Datei können Matlab-Programme bearbeitet, gespeichert, etc. werden. Vor dem ersten Start Ihres Programms, muss dieses zunächst gespeichert werden, beispielsweise unter `prog1.m`. Danach kann das Programm durch Eingabe von `prog1.m` im Command Window gestartet werden. Dabei ist zu beachten, dass sich die Datei in dem aktuellen Pfad befindet, ansonsten findet Matlab die Datei nicht. Alternativ kann das Programm im Editor auch durch Betätigen der Taste F5 gestartet werden. Diese Option ist empfehlenswert, um das Programm im Debug-Modus laufen zu lassen.

Anhang B

Übersicht zu den Befehlen



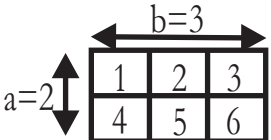
Zuordnung von Zahlen, Vektoren und Matrizen

<code>A = 1;</code>	Variable A wird der Wert 1 zugewiesen, Semikolon unterdrückt Bildschirmausgabe
<code>A = 1:20;</code>	Array (Zeilenvektor) A mit den Elementen 1 bis 20
<code>A = 2:5:25;</code>	Array (Zeilenvektor) A mit Werten von 2 bis 25 mit der Schrittweite 5: $A = [2, 7, 12, 17, 22]$
<code>A = [1,2,3];</code>	Array (Zeilenvektor) $A = (1\ 2\ 3)$
<code>A = [1;2;3];</code>	Array (Spaltenvektor) $A = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$
<code>A = [1,2,3;4,5,6;7,8,9]</code>	Matrix $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$
<code>B = A(:);</code>	A in Spaltenvektor wandeln (Spaltenweise von links nach rechts) $B = \begin{bmatrix} 1 \\ 4 \\ 7 \\ 2 \\ \vdots \end{bmatrix}$
<code>B = A(1);</code>	B wird das erste Element aus A zugewiesen (Index beginnt mit 1 nicht mit 0 wie bei C/C++)
<code>B = A(1,2);</code>	B wird das Element aus der 1. Zeile der 2. Spalte von A zugewiesen (also hier 2)
<code>B = A(2:3,1:3);</code>	B wird ist eine 2×3 Submatrix bestehend aus den Zeilen 2 bis 3 und den Spalten 1 bis 3: $B = \begin{bmatrix} 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$
<code>A = [a;b;c];</code>	Matrix $A = \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ mit den Zeilenvektoren $a = [1, 2, 3]$, $b = [4, 5, 6]$, $c = [7, 8, 9]$
<code>D = [A,B,C];</code>	Matrix $D = [A\ B\ C] = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ mit $A = \begin{bmatrix} 1 \\ 4 \\ 7 \end{bmatrix}$, $B = \begin{bmatrix} 2 \\ 5 \\ 8 \end{bmatrix}$, $C = \begin{bmatrix} 3 \\ 6 \\ 9 \end{bmatrix}$
<code>A = B + j*C</code>	Erzeugt komplexe Matrix mit Realteil B und Imaginärteil C
<code>A = zeros(y,x);</code>	Erzeugt Matrix mit y Zeilen und x Spalten initialisiert mit 0.
<code>A = zeros(z);</code>	Erzeugt $z \times z$ Matrix initialisiert mit 0.
<code>A = ones(y,x);</code>	Erzeugt Matrix analog zu <code>zeros()</code> initialisiert mit 1.

Operatoren

<code>+</code>	<code>-</code>	Addition, Subtraktion	
<code>*</code>	<code>/</code>	<code>^</code>	(Matrix)- Multiplikation, Division, Potenz
<code>.*</code>	<code>./</code>	<code>.^</code>	Elementweise Multiplikation, Division, Potenz
<code>'</code>	<code>.'</code>	Transjungtierte, Transponierte	

Grafik

<code>stem(x,y)</code>		y-Werte über x-Achse aufgetragen. <code>stem(y)</code> erzeugt eigene X-Achse mit der Länge von y
<code>plot(x,y,option)</code>		Analog zu <code>stem()</code> , mit zusätzlichen Optionen (Farbe, Linien-Form..)
<code>subplot(a,b,c)</code>		unterteilt das Bild und legt aktuelles Teilbild fest

Grafik

<code>figure</code>	Öffnet ein neues Figure.
<code>figure(x)</code>	Figure mit der Nummer x wird als aktuelles Figure gesetzt.
<code>hold on</code>	Aktuelles Bild wird nicht mehr überschrieben. Neue Plots werden eingefügt.
<code>hold off</code>	Aktuelles Bild wird freigegeben. Nur <code>hold</code> springt zw. freigegebenen und festen Zustand hin und her.
<code>grid</code>	Gitter an aus, analog zu <code>hold</code>
<code>axis([x_{min} x_{max} y_{min} y_{max}])</code>	Legt Achsenausschnitt fest
<code>xlabel('Text')</code>	X-Achsenbeschriftung, (<code>ylabel()</code> , <code>zlabel()</code> , <code>title()</code>) analog

wichtige Funktionen

<code>help < fkt ></code>	gibt Hilfstext zur Funktion <i>fkt</i> , z.B: <i>help imag</i>
<code>doc < fkt ></code>	gibt ausführliche Hilfe zur Funktion <i>fkt</i> , z.B: <i>doc imag</i>
<code>i, j, pi</code>	vorgegebene Konstanten (können überladen werden)
<code>real(x), imag(x)</code>	Real-Teil bzw. Imaginär-Teil von <i>x</i>
<code>exp</code>	Exponentialfunktion
<code>sin(x), cos(x)</code>	Sinus bzw. Kosinus (<i>x</i> in rad!)
<code>tan(x), cot(x)</code>	Tangens bzw. Kotangens (<i>x</i> in rad!)
<code>asin(x), acos(x)</code>	Arkus-Sinus bzw. Arkus-Kosinus
<code>atan(x), acot(x)</code>	Arkus-Tangens bzw. Arkus-Kotangens
<code>[a,b]=size(x)</code>	gibt die Größe von <i>x</i> aus (<i>a</i> = Anzahl der Zeilen, <i>b</i> = Anzahl der Spalten)
<code>length(x)</code>	Länge des Vektors <i>x</i> , entspricht <i>max(size(x))</i>
<code>sum(x)</code>	Summe des Vektors <i>x</i> bzw. Summe aller Spalten der Matrix <i>x</i>
<code>max(x)</code>	Bildet das Maximum, analog zu <i>sum(x)</i> bzw. <i>min(x)</i>

Steuerung des Programmablaufs

<code>for i=a:s:e</code> <statements> <code>end;</code>	for - Schleife mit Anfangswert <i>a</i> und Endwert <i>e</i> sowie der optionalen Stufigkeit <i>s</i> . Anstelle dessen sind auch Konstrukte der Form <i>i</i> = [<i>x</i> ₁ <i>x</i> ₂ .. <i>x</i> _{<i>n</i>}] zulässig. Schleife wird <i>n</i> -mal durchlaufen, mit den Indizes <i>x</i> ₁ <i>x</i> ₂ .. <i>x</i> _{<i>n</i>}
<code>if <A1>;</code> <statements1> <code>elseif <A2></code> <statements2> <code>else</code> <statements3> <code>end;</code>	if Konstrukt, verzweigt den Programmablauf. Ausdrücke (Bedingungen) <i>A1</i> und <i>A2</i> haben oftmals die Form: <i>a</i> > <i>b</i> , <i>a</i> == <i>b</i> , <i>a</i> <= <i>b</i> , ... Die <i>else</i> und <i>elseif</i> - Anweisungen sind optional. Siehe hierzu Bild B.1

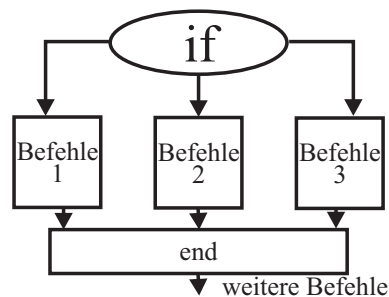
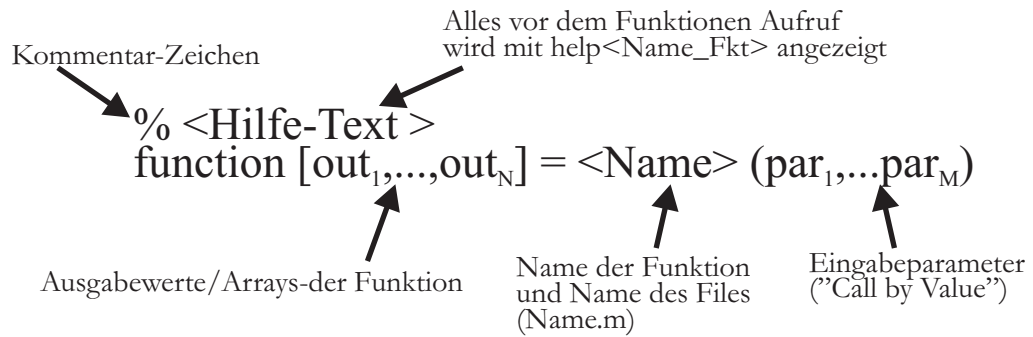


Abbildung B.1: *if*-Verzweigungen

Funktionen-Header



Im Programm erfolgt dann der Aufruf:

`[out1, ..., outN] = <Name> (par1, ..., parM)`