



Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Einführung in die Zeitreihenanalyse Teil 1

N. Deichmann

Institut für Geophysik
ETH-Zürich
März 2002

Inhalt

1	Zeitreihen	1
2	Diskrete Systeme	9
3	Die Fourier Transformation	17
4	Das Leistungsspektrum	40
5	Korrelation	44
6	Stochastische Zeitreihen	51

Kapitel 1

Zeitreihen

Die Diskretisierung des Kontinuums

Einführung

Im Allgemeinen gehen wir davon aus, dass Veränderungen unserer makroskopischen Umwelt kontinuierlich ablaufen. Messtechnisch lässt sich das zum Beispiel für die Änderungen der Lufttemperatur an Hand der traditionellen Thermographen veranschaulichen: Die gemessene Temperatur als Funktion der Zeit wird kontinuierlich mit einer Tintenfeder auf eine sich langsam drehende Trommel aufgezeichnet. Das Resultat ist eine ohne Unterbruch durchgezogene Kurve. Obwohl eine solche Kurve eine vollständige Wiedergabe der beobachteten Temperatur darstellt, lässt sie keine weitere mathematische Bearbeitung zu. Dafür müssen die Messungen als Zahlenwerte zu diskreten Zeitpunkten vorliegen. Eine Folge solcher diskreter Messwerte nennt man eine Zeitreihe oder ein diskretes Signal. Bei der elektronischen Datenerfassung werden die Analogsignale (z. Bsp. eine kontinuierliche Spannungsänderung am Ausgang eines Messgerätes) durch Digitalisierung in digitale Signale umgewandelt. Ein Beispiel hiervon sind die digitalen Seismogramme, welche die durch ein Erdbeben oder eine Sprengung verursachte Bodenbewegung darstellen.

Zusätzlich zu den diskreten Beobachtungen kontinuierlicher Prozesse als Funktion der Zeit gibt es aber auch Zeitreihen, die einfach aus dem Zählen diskreter Ereignisse entstehen. Beispiele solcher Zeitreihen sind die zeitliche Abfolge von Erdbeben in einem bestimmten Gebiet oder die Anzahl Autos, die zu unterschiedlichen Zeiten eine gegebene Strasse befahren.

Mit der Entwicklung des Computers, der mit grosser Geschwindigkeit riesige Zahlenmengen verarbeiten kann, hat sich aus der Mathematik des Kontinuums eine Mathematik des Diskreten entwickelt. Die Zeitreihenanalyse befasst sich mit den mathematischen Methoden zur quantitativen Erfassung und Interpretation der Eigenschaften diskreter Daten.

Einige Definitionen

Abtastperiode und Abtastrate

Bei der praktischen Verarbeitung von diskreten Messwerten stellt sich das Problem, eine kontinuierliche Zeitfunktion $x(t)$ zu diskreten Zeiten t_k zu betrachten. Die durch die *Abtastung* (englisch: *sampling*) entstandene *Folge* von Zahlen (*Stützwerte*, Engl. *samples*) nennt man *Zeitreihe* (Engl. *time series*)

Beispiel: $\{x_k\} = x_0, x_1, x_2, x_3, x_4, x_5, x_6, \dots = -23, -4, -19, -7, 5, 9, \dots$

Die zeitliche Differenz zwischen zwei aufeinanderfolgende Stützwerte $\Delta t_k = t_{k+1} - t_k$ heisst *Abtastintervall* oder *Abtastperiode* (Engl. *sampling period* oder *sampling interval*). Wenn die Abtastung zu äquidistanten Zeitpunkten vorgenommen wird, dann ist das Abtastintervall $\Delta t_k = \Delta t$ eine konstante Grösse. Oft wird das Abtastintervall auch nur mit Δ oder mit T bezeichnet. Folgende Schreibweisen sind je nach verwendeter Notation als äquivalent zu betrachten:

$$\{x_k\} = x_k = x(k) = x(kT) = x(k\Delta) = x(k\Delta t) = x(t)|_{t=k\Delta t}.$$

Der diskrete Index wird im Folgenden auch mit n oder i statt k bezeichnet.

Die *Abtastrate*, *Abtastfrequenz* oder *Digitalisierfrequenz* (Engl. *sampling rate*, *sampling frequency* oder *digitizing frequency*) ist die reziproke Abtastperiode: $f_s = 1/\Delta t$. Es ist intuitiv schon klar, dass eine kleine Abtastperiode oder grosse Abtastrate eine feinere Wiedergabe eines gegebenen kontinuierlichen Signals erlaubt, dies aber auf Kosten einer grösseren Datenmenge und entsprechendem Verarbeitungsaufwand geht. Tatsächlich zeigt es sich, dass eine zu kleine Abtastrate nicht nur eine ungenügende Auflösung zur Folge hat, sondern dass sie auch eine nicht rückgängig zu machende Verfälschung des digitalen Signals bewirkt. Somit stellt sich die grundsätzliche Frage nach einem Kriterium für die minimal notwendige Abtastrate für ein gegebenes Signal. Die Antwort ist durch das sogenannte *Abtasttheorem* gegeben, welches wir im Anschluss an eine Einführung in die Fourier Analyse behandeln werden.

Mittelwert eines Signals

$$\bar{x}_k = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^N x_n$$

Der Mittelwert wird auch oft als *offset* bezeichnet.

Energie eines Signals

$$E = \sum_{n=-\infty}^{\infty} |x_n|^2$$

Mittlere Leistung

$$P = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^N |x_n|^2.$$

Im Englischen wird die Leistung als *power* bezeichnet.

Klassifizierung von Zeitreihen

Im Folgenden sollen diskrete Signale nach ihren Eigenschaften klassifiziert werden.¹

Einseitige und zweiseitige Signale: wir unterscheiden,

- einseitig in die Zukunft (nach rechts): $x_n = 0$ für $n < \alpha$
- einseitig in die Vergangenheit (nach links): $x_n = 0$ für $n > \beta$
- zweiseitige Signale
- Signale endlicher Länge: $x_n = 0$ für $n < \alpha$ und für $n > \beta$

Kausale und nichtkausale Signale: wir unterscheiden,

- kausal: $x_n = 0$ für $n < 0$
- nicht-kausal: einige oder alle $x_n \neq 0$ für $n < 0$
- anti-kausal oder streng nicht-kausal: $x_n = 0$ für $n > 0$

(Man stelle sich vor, der Index $n = 0$ entspricht der Ankunftszeit des Signals.)

Stabile und nichtstabile Signale: es bestehen zwei unterschiedliche Kriterien für die Definition von Stabilität:

1. absolut summierbar: $\sum_{-\infty}^{\infty} |x_n| < \infty$
2. endliche Energie: $\sum_{-\infty}^{\infty} |x_n|^2 < \infty$

Wie man aus $\sum |x|^2 \leq [\sum |x|]^2$ sieht, ist die Definition 1. restriktiver als 2.

¹Eine Uebersetzung und Kurzfassung aus *Robinson, E. A., Durrani, T. S., und Pearson, L. G.: Geophysical Signal Processing. Prentice Hall International, London, 1986.*

Beispiel: eine Folge, die nicht absolut summierbar ist, aber endliche Energie aufweist:

$$x_n = \begin{cases} 1/n & n > 0 \\ 0 & n \leq 0 \end{cases}$$

$$\sum_{n=1}^{\infty} |1/n| = \infty,$$

$$\sum_{n=1}^{\infty} |1/n|^2 = \pi^2/6 < \infty.$$

Energie- oder Leistungssignale: wir unterscheiden,

- Energiesignal (die totale Energie ist endlich):

$$E = \sum_{-\infty}^{\infty} |x_n|^2 < \infty$$

- Leistungssignal (die über alle n gemittelte Leistung ist endlich):

$$0 < \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^{n=+N} |x_n|^2 < \infty$$

Ein Energiesignal hat null Leistung und ein Leistungssignal hat unendliche Energie. Ausserdem gibt es Signale die weder das eine noch das andere sind.

Periodische und nichtperiodische Signale: wir unterscheiden,

- Periodisch: es gibt eine ganze Zahl p , so dass $\{x_n\} = \{x_{n+p}\}$ für alle n .
- Nichtperiodisch: wenn obige Bedingung nicht erfüllt ist.

Deterministische und stochastische Signale: wir unterscheiden,

- Deterministisch: wenn schon vor dem Auftreten des Signals keine Unsicherheit über die Form des Signals besteht, d.h. wenn zumindest im Prinzip eine mathematische Relation existiert, welche das Signal für alle Zeiten beschreibt,
- Stochastisch: wenn auch nach der Beobachtung von vergangenen Werten des Signals, eine Unsicherheit über sein zukünftiges Verhalten besteht; in diesem Fall werden zur Beschreibung des Signals statistische Kriterien verwendet.

Die Unterscheidung zwischen deterministisch und stochastisch ist nicht unbedingt naturgegeben, sondern ist abhängig von der Modellierbarkeit eines Phänomens, d.h. ist ein Merkmal unserer Kenntnisse über das Signal.

Einige elementare diskrete Signale

Der diskrete δ -Impuls

Als ein diskretes Einheitssignal kann man die diskrete δ -Funktion oder *Einheitsimpuls* auffassen:

$$\delta_k = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases}$$

In der Praxis ist es zweckmässig, den Einheitsimpuls so zu definieren, dass seine Amplitude (Höhe) gleich der reziproken Abtastperiode ist:

$$\delta_k = \begin{cases} 1/\Delta t & k = 0 \\ 0 & k \neq 0 \end{cases}$$

Letztere Definition bewirkt, dass die Fläche unter dem diskreten Einheitsimpuls

$$\Delta t \sum_{k=-\infty}^{\infty} \delta_k = 1$$

ist, was äquivalent zum Integral über den δ -Impuls im Kontinuum ist. Im Kontinuum ist der δ -Impuls auch als *Dirac-Impuls* bekannt (s. später). Im Gegensatz zum kontinuierlichen Fall gibt es aber im Diskreten keine theoretischen Schwierigkeiten bei der Definition des δ -Impulses.

Die diskrete Sprungfunktion

Die diskrete Sprungfunktion, die oft entweder mit u_k oder h_k bezeichnet wird, ist wie folgt definiert:

$$u_k = \begin{cases} 1 & k \geq 0 \\ 0 & k < 0 \end{cases}$$

Die Entsprechende Funktion im Kontinuum wird auch als *Heaviside Funktion* bezeichnet.

Beziehung zwischen δ -Impuls und Sprungfunktion

Aus der in der elementaren Analysis üblichen Definition der Ableitung einer Funktion ist sofort ersichtlich, dass die diskrete Ableitung als ein Differenzenquotient angenähert werden kann:

$$\frac{dx(t)}{dt} \approx \frac{x_k - x_{k-1}}{\Delta t}$$

Angewendet auf die Sprungfunktion sehen wir, dass der δ -Impuls die diskrete Ableitung der Sprungfunktion ist:

$$\delta_k = \frac{u_k - u_{k-1}}{\Delta t}$$

Demzufolge kann die *Sprungfunktion als Integral des δ -Impulses* betrachtet werden, was im Diskreten zu einer Summe führt:

$$u_k = \Delta t \sum_{n=-\infty}^k \delta_n.$$

Deterministische und Stochastische Prozesse

Stochastische Prozesse können als zufällige Zeitfunktionen aufgefasst werden. Die Anzahl Anrufe an einer Telephonzentrale pro Zeiteinheit kann als ein solcher Prozess betrachtet werden.

Ein stochastischer Prozess kann aber auch als eine Schar von Zufallsgrößen, die von der Zeit t abhängen, betrachtet werden.

Diese Betrachtungsweise erlaubt eine einfache Definition wichtiger Begriffe wie *Stationarität* und *Ergodizität*.

In dieser Vorlesung werden wir einen stochastischen Prozess $\xi(t)$ als eine *Schar von reellen Zufallsgrößen*, die von der Zeit $t \in T$ abhängen, betrachten. Dabei ist T eine geordnete Menge (z.B. die Zeitachse). Die Schar ist so zu verstehen, dass für jedes feste $t_i \in T$, $\xi(t_i)$ eine gewöhnliche Zufallsgrösse darstellt. Wenn t die Menge T durchläuft, entsteht eine Realisierung des stochastischen Prozesses $\xi(t)$.

Intuitiv wird der Unterschied zwischen einer deterministischen und einer stochastischen Betrachtungsweise anhand des folgenden Beispiels verständlich.

Beispiel: Betrachten wir den freien Fall eines Körpers im Raum. Die Gesetze der Mechanik erlauben, falls die Anfangsbedingungen und der Wert von g exakt bekannt sind, die Trajektorie zu bestimmen. Das heisst, falls die *Ursachen* bekannt sind (hier : Anfangsbedingungen und der Wert von g), dann können die *Effekte* (hier: Trajektorie) bestimmt werden.

Eine sorgfältigere Analyse des Problems zeigt aber, dass im Falle, wo wir auf eine extrem genaue Bestimmung der Trajektorie angewiesen sind, unserer Kenntnis Grenzen gesetzt sind. Diese Grenzen kommen einerseits aus den Anfangsbedingungen und aus dem Wert von g , welche fehlerbehaftet sind und aus den Einflüssen der Luftreibung und den aerodynamischen Kräften, die im allgemeinen nur begrenzt bekannt sind.

Unter diesen Umständen ist die Trajektorie des Körpers nicht exakt berechenbar und das oben betrachtete *deterministische* Modell kann grundsätzlich nur Hinweise zur "exakten" Trajektorie liefern.

Unvorhergesehene Umstände können sogar dazu führen, dass wiederholte Versuche verschiedene Trajektorien als Resultat haben. Eine

sorgfältige Analyse der Resultate zeigt im allgemeinen eine gewisse statistische Regelmässigkeit in dem Sinne, dass eine nach gewissen Kriterien noch zu definierende mittlere Trajektorie die Tendenz zeigt, mit steigender Anzahl von Versuchen, sich zu stabilisieren.

Aus diesem Beispiel können wir jetzt folgende intuitive Definition eines stochastischen Prozesses betrachten:

Ein stochastischer Prozess ist ein Phänomen, dessen zeitliche Entwicklung durch die Komplexität der zugrundeliegenden Gesetze und durch die Unwägbarkeit verschiedener Faktoren nicht exakt, sondern nur innerhalb gewisser Grenzen bestimmt werden kann.

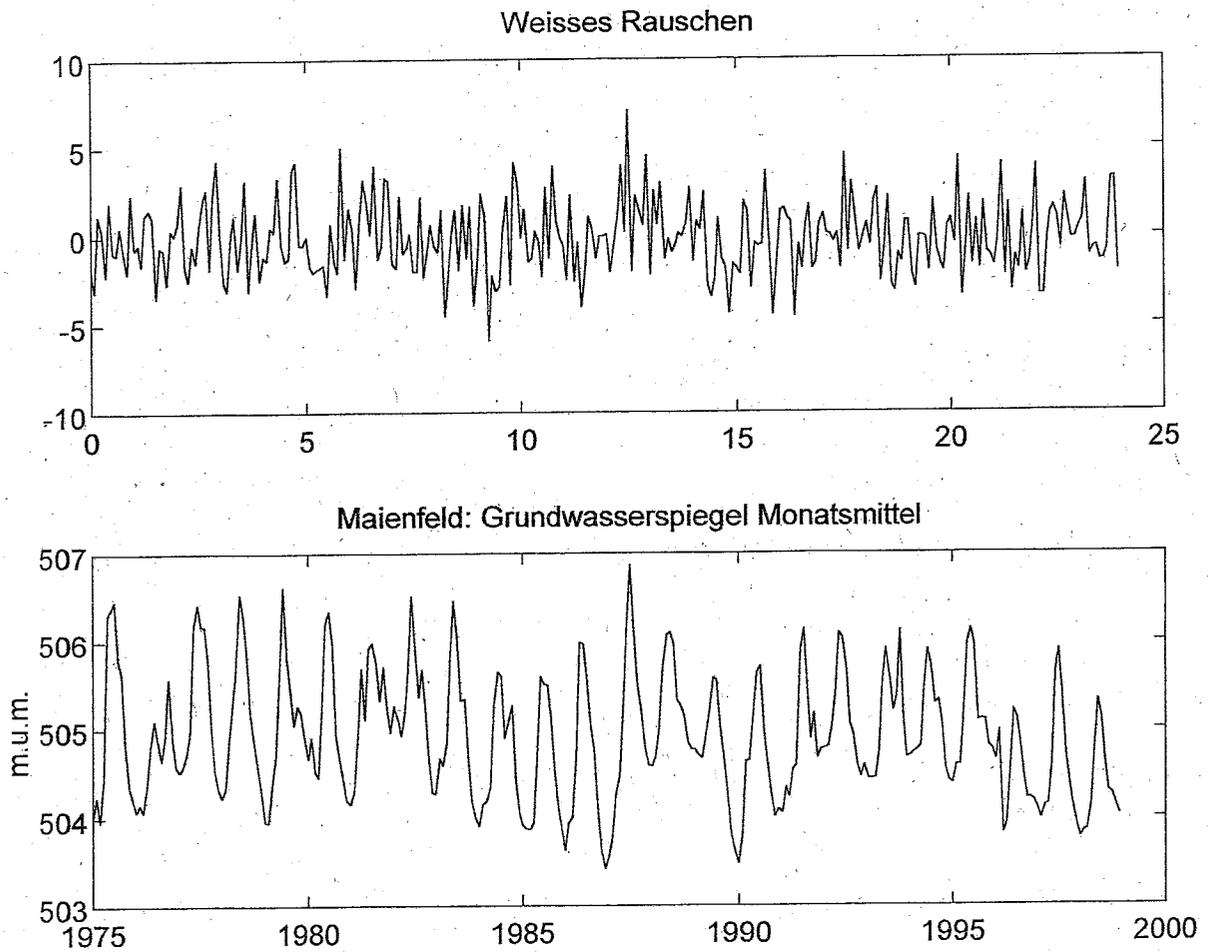
Abbildung 1.1 zeigt zwei Beispiele von stochastischen Zeitreihen: Weisses Rauschen, *white noise*, das aus einer Folge von reinen Zufallszahlen besteht, und eine Messreihe des Grundwasserspiegels bei Maienfeld. Während das weisse Rauschen sich durch maximale Regellosigkeit auszeichnet, sieht man in der Zeitreihe des Grundwasserspiegels Anzeichen einer periodischen Komponente. Die Bedeutung des Attributes *weiss* im weissen Rauschen, wird im Zusammenhang mit der Fourier Analyse klar. Weisses Rauschen hat in der Theorie der stochastischen Zeitreihen die gleiche Rolle wie der Einheitsimpuls in der Theorie der deterministischen Signale. Die Untersuchung stochastischer Prozesse erfolgt mit Methoden der Wahrscheinlichkeitstheorie, der Statistik und der mathematischen Systemtheorie.

Ueber den Begriff von Noise

Bei *Noise* (eine Störung oder das Unerklärliche) werden wir in Zukunft je nach Problem folgendes zu verstehen haben: jeder kausale oder zufällige Einfluss, der nicht modelliert worden ist, oder nicht modelliert werden kann, bei welchem keine zusätzliche Information gewonnen werden kann, der nicht reproduzierbar ist, u.s.w.

In Zusammenhang mit unserer Vorlesung werden wir das Wort Noise verwenden als allgemeiner Begriff für:

NOISE { Messfehler,
numerische Fehler,
Auswirkungen, die nicht modelliert worden sind,
Auswirkungen, die nicht modellierbar sind,
nicht reproduzierbare Störungen.



Figur 1.1: Zwei Beispiele stochastischer Zeitreihen: Weisses Rauschen (oben) und eine Messreihe der Monatsmittel des Grundwasserspiegels bei Maienfeld (unten).

Kapitel 2

Diskrete Systeme

Einleitung

In der Zeitreihenanalyse, bzw. digitalen Signalanalyse geht es häufig darum, Prozesse zu untersuchen, die eine Zeitreihe x_n in eine zweite Zeitreihe y_n verwandeln. Dabei kann es sowohl darum gehen, aus den beobachteten oder angenommenen Zeitreihen x_n und y_n auf den unbekanntem Prozess zu schliessen als auch die Wirkung eines bekannten Prozesses zu simulieren, um die Eigenschaften der Zeitreihe x_n oder y_n zu untersuchen. Solche Prozesse werden in der Analyse von Zeitreihen und digitalen Signalen als *diskrete Systeme* bezeichnet.

Als Beispiel aus der Seismologie können sowohl der Ausbreitungsweg in der Erde als auch jedes Aufzeichnungsgerät ganz allgemein als Systeme betrachtet werden, welche in einer bestimmten Art die Form der beobachteten Signale bedingen. Eines der Hauptaufgaben der digitalen Signalverarbeitung ist dann die rechnerische Korrektur der dadurch bedingten Verzerrungen sowie die Unterdrückung von unerwünschten Eigenschaften im beobachteten Signal.

Einerseits geht es also darum, physikalische Systeme durch rechnerische Systeme zu simulieren, und andererseits darum, rechnerische Systeme zu entwickeln mit denen ein gegebenes Eingangssignal x_n in ein Ausgangssignal y_n mit gewünschten Eigenschaften verwandelt werden kann. Die Wirkung eines solchen Systems wird symbolisch oft wie folgt dargestellt:

$$x_n \longrightarrow y_n.$$

Ein einfaches aber auch typisches Beispiel sei die folgende Aufgabe. Gegeben ist ein langperiodisches Signal, welches durch ein zweites kurzperiodisches Signal überlagert ist. Indem wir die kurzperiodischen Signalanteile unterdrücken, wollen wir nun dieses Eingangssignal x_n in ein geglättetes Ausgangssignal y_n verwandeln. Ein mögliches Vorgehen ist die Berechnung

eines gleitenden Mittelwertes (Engl. *moving average*)¹ über drei benachbarte Punkte:

$$y_n = \frac{x_n + x_{n-1} + x_{n-2}}{3} = \frac{1}{3}x_n + \frac{1}{3}x_{n-1} + \frac{1}{3}x_{n-2}.$$

Beispiel: sei $x_n = \{1, 3, -2, 5, 3, -4, -1, 4, 1, -5, 2, \dots\}$, dann ist

$$y_n = \{0.33, 1.33, 0.67, 2, 2, 1.33, -0.67, -0.33, 1.33, 0, -0.67, \dots\},$$

wobei wir hier angenommen haben, dass $x_n = 0$ für $n < 0$.

Definieren wir nun eine Zahlenfolge

$$h_n = \{h_0, h_1, h_2\} = \{1/3, 1/3, 1/3\}$$

dann können wir obige Rechenvorschrift auch wie folgt ausdrücken:

$$y_n = h_0x_n + h_1x_{n-1} + h_2x_{n-2} = \sum_{m=0}^2 h_m x_{n-m}.$$

Man sagt dann, dass y_n das Resultat der diskreten Konvolution von h_n und x_n ist. Die Konvolution wird auch als Faltung bezeichnet und kompakt wie folgt geschrieben:

$$y_n = h_n * x_n.$$

Betrachten wir nun den Fall des Einheitsimpulses als Eingangssignal: $x_n = \delta_n$. Aus der Eigenschaft des Einheitsimpulses folgt dann

$$y_n = h_n * \delta_n = h_n$$

und h_n wird als die *Impulsantwort* des Systems bezeichnet. In obigem Beispiel ist der Glättungsoperator ein sogenanntes *lineares verschiebungsinvariantes System* und die Zahlenfolge $\{1/3, 1/3, 1/3\}$ ist die entsprechende Impulsantwort. Wie wir im Folgenden zeigen werden, lässt sich dieses Beispiel zu der Aussage verallgemeinern

*Das Ausgangssignal y_n eines linearen verschiebungsinvarianten Systems als Folge eines beliebigen Eingangssignals x_n ist das Resultat der Faltung von x_n mit der Impulsantwort h_n des Systems: $y_n = h_n * x_n$.*

Einige Definitionen

Im Folgenden wollen wir den soeben beschriebenen Sachverhalt formal begründen. Ein diskretes System ist eine Relation, welche ein diskretes Eingangssignal $\{x_n\}$ in ein diskretes Ausgangssignal $\{y_n\}$ überführt:

$$\{x_n\} \longrightarrow \{y_n\}.$$

¹Nicht zu verwechseln mit *running average*!

Einfachheitshalber schreiben wir im Folgenden jeweils nur x_n und y_n . Ob damit die gesamte Zahlenfolge oder ein einziges Element gemeint ist, ergibt sich meistens aus dem Zusammenhang. Die Zahlenfolge x_n bzw. y_n wird auch als Zeitreihe (Engl. *time series*) bezeichnet, was aber mit einer formalen Reihe nichts zu tun hat.

Ist das Eingangssignal der Einheitsimpuls $x_n = \delta_n$, dann bezeichnet man $y_n = h_n$ als die *Impulsantwort* des Systems:

$$\delta_n \longrightarrow h_n.$$

Diskrete Systeme können nach folgenden Kriterien klassifiziert werden:

Linearität Ein System (Operator oder Filter) wird als *linear* bezeichnet wenn es folgende Bedingung erfüllt:

Wenn $x_n \longrightarrow y_n$ und $q_n \longrightarrow r_n$, dann muss

$$\alpha x_n + \beta q_n \longrightarrow \alpha y_n + \beta r_n,$$

wobei α und β beliebige (im Allgemeinen komplexe) Konstanten sind.

Alle anderen Systeme werden als *nichtlinear* bezeichnet.

Die Bedingung der Linearität besagt folgendes:

Sei

$$y = f(x) \text{ und } r = f(q)$$

dann ist die Funktion f linear wenn

$$f(\alpha x + \beta q) = \alpha f(x) + \beta f(q) = \alpha y + \beta r$$

Die Linearität ist eine besonders wichtige Eigenschaft, weil sie es erlaubt eine oft komplizierte Beziehung als Ueberlagerung (*Superposition*) von einfacheren Beziehungen auszudrücken.

Beispiel 1: Zeigen Sie durch anwenden obiger Definition, dass die Funktion $f(x) = mx$ linear ist.

Sei

$$y = mx \text{ und } r = mq$$

dann ist

$$f(\alpha x + \beta q) = m(\alpha x + \beta q) = \alpha mx + \beta mq$$

und

$$\alpha y + \beta r = \alpha mx + \beta mq$$

Somit ist die Funktion $f(x) = mx$ linear.

Beispiel 2: Ist die Funktion $f(x) = x^2$ linear?

Sei

$$y = x^2 \text{ und } r = q^2$$

dann ist

$$f(\alpha x + \beta q) = (\alpha x + \beta q)^2 = \alpha^2 x^2 + 2\alpha x \beta q + \beta^2 q^2$$

und

$$\alpha y + \beta r = \alpha x^2 + \beta q^2$$

Somit ist

$$f(\alpha x + \beta q) \neq \alpha y + \beta r$$

und die Funktion $f(x) = x^2$ ist, wie erwartet, nicht linear.

Beispiel 3: Ist die Funktion

$$f(x) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt$$

linear? (Diese Funktion ist das Fourier Integral)

Sei

$$y = \int_{-\infty}^{\infty} x e^{-j\omega t} dt \text{ und } r = \int_{-\infty}^{\infty} q e^{-j\omega t} dt$$

dann ist

$$f(\alpha x + \beta q) = \int (\alpha x + \beta q) e^{-j\omega t} dt = \int \alpha x e^{-j\omega t} dt + \int \beta q e^{-j\omega t} dt$$

und

$$\alpha y + \beta r = \alpha \int x e^{-j\omega t} dt + \beta \int q e^{-j\omega t} dt = \int \alpha x e^{-j\omega t} dt + \int \beta q e^{-j\omega t} dt$$

Somit ist (wie wahrscheinlich nicht erwartet) das Fourier Integral eine lineare Operation.

Frage: Ist die allgemeine Form der Geradengleichung $y = mx + n$ eine lineare Beziehung?

Invarianz gegenüber Verschiebung Ein System wird als *verschiebungsinvariant* (Englisch: *shift-invariant*) oder, falls der Index n die Zeit darstellt, als *zeit-invariant* bezeichnet, wenn die Beziehung zwischen Ein- und Ausgang unabhängig von n ist, d.h. wenn aus $x_n \rightarrow y_n$, $x_{n-k} \rightarrow y_{n-k}$ folgt.

Kausalität Ein System wird als *kausal* bezeichnet, wenn der Ausgang y_n zum Zeitpunkt n nur von gegenwärtigen und allenfalls vergangenen Eingangswerten $x_n, x_{n-1}, x_{n-2}, \dots$ und möglicherweise von den früheren

Ausgangswerten $y_{n-1}, y_{n-2}, y_{n-3}, \dots$ abhängt. Falls y_n durch eine Relation gegeben ist, welche auch spätere (zukünftige) Werte x_{n+k} miteinbezieht, dann wird das System als *nichtkausal* oder *akausal* bezeichnet. Physikalisch ist natürlich nur ersteres realisierbar. Rechnerisch hingegen kann man auch nichtkausale Systeme realisieren, wie folgendes Beispiel zeigt.

Beispiel: Der oben schon eingeführte Glättungsoperator mit drei Koeffizienten (gleitender Mittelwert) kann auch wie folgt als akausales System definiert werden:

$$y_n = \sum_{k=-1}^1 \frac{1}{3} x_{n-k} = \frac{1}{3} (x_{n+1} + x_n + x_{n-1}).$$

In diesem Fall ist der Ausgangswert eine Linearkombination vom gegenwärtigen Eingangswert mit einem vergangenen und einem zukünftigen Eingangswert. Um dieses System kausal zu machen, genügt es den Ausgang um einen Wert zu verzögern:

$$y_n = \sum_{k=0}^2 \frac{1}{3} x_{n-k} = \frac{1}{3} (x_n + x_{n-1} + x_{n-2}).$$

Stabilität Ein System wird als *stabil* bezeichnet, wenn ein beschränktes Eingangssignal auch ein beschränktes Ausgangssignal zur Folge hat. Insbesondere ist die Impulsantwort eines stabilen Systems eine beschränkte Zeitreihe. Analog zur Definition der Stabilität von diskreten Signalen, gibt es auch für diskrete Systeme zwei unterschiedliche Stabilitätskriterien, bzw. Definitionen von Beschränktheit:

1. Ein System ist stabil wenn die Impulsantwort absolut summierbar ist:

$$\sum_{n=-\infty}^{\infty} |h_n| \leq C_1 < \infty$$

2. Ein System ist stabil wenn die Impulsantwort endliche Energie aufweist:

$$\sum_{n=-\infty}^{\infty} |h_n|^2 \leq C_2 < \infty.$$

Diskrete Konvolution

Eines der Hauptprobleme der digitalen Signalverarbeitung ist die Bestimmung des Ausgangssignal eines Systems als Folge eines beliebigen Eingangssignals. Im Falle von linearen und verschiebungsinvarianten Systemen

lässt sich das Superpositionsprinzip anwenden, wobei der Ausgang als Ueberlagerung der Antworten des Systems auf einzelne elementare Eingangssignale dargestellt wird. Die entsprechende Operation ist die *Konvolution* oder *Faltung*.

Das Einheitssignal ist der diskrete Einheitsimpuls δ_n und der entsprechende Ausgang des Systems ist die Impulsantwort h_n . Wir wollen die Wirkung des Systems durch einen Pfeil symbolisieren und schreiben somit $\delta_n \rightarrow h_n$. Da das System verschiebungsinvariant ist, gilt auch $\delta_{n-k} \rightarrow h_{n-k}$. Wir suchen nun einen Ausdruck, der ein beliebiges Eingangssignal x_n in ein entsprechendes Ausgangssignal y_n überführt: $x_n \rightarrow y_n$.

Wir können jede beliebige Zeitreihe als Summe von skalierten und verschobenen Einheitsimpulsen darstellen:

$$x_n = \sum_{k=-\infty}^{\infty} x_k \delta_{n-k}.$$

Hier ist nur für $k = n$ der Summand $\neq 0$, so dass sich für jeden Wert von n die Summe auf ein einziges Element reduziert und wir die Identität $x_n = x_n$ haben. Diese scheinbar komplizierte Darstellung eines banalen Resultates wird uns gleich recht nützlich sein.

Aus der Eigenschaft der Linearität unseres Systems folgt erstens,

$$x_k \delta_{n-k} \rightarrow x_k h_{n-k}$$

und zweitens,

$$\sum_{k=-\infty}^{\infty} x_k \delta_{n-k} \rightarrow \sum_{k=-\infty}^{\infty} x_k h_{n-k}.$$

Auf der linken Seite dieser Korrespondenz steht also ein beliebiges Eingangssignal x_n . Somit entspricht die rechte Seite dem gesuchten Ausgangssignal y_n mit der Beziehung $x_n \rightarrow y_n$. Der Ausdruck

$$y_n = \sum_{k=-\infty}^{\infty} x_k h_{n-k} = x_n * h_n$$

wird als diskrete Konvolution oder Faltung bezeichnet.

Dies besagt, dass der Ausgang y_n eines linearen und verschiebungsinvarianten Systems als Folge eines beliebigen Eingangssignals x_n die Faltung dieses Eingangssignals mit der Impulsantwort des Systems ist.

Eigenschaften der Konvolution

Ersetzt man $m = n - k$, dann erhält man für die Konvolution:

$$y_n = \sum_{m=-\infty}^{\infty} h_m x_{n-m} = h_n * x_n.$$

Die Konvolution ist also *kommutativ*.

Im weiteren ist die Konvolution auch *distributiv*,

$$(x_n + q_n) * h_n = x_n * h_n + q_n * h_n$$

und *assoziativ*,

$$(x_n * q_n) * h_n = x_n * (q_n * h_n) = x_n * q_n * h_n.$$

Ausführung der Konvolution

Zur Veranschaulichung des praktischen Vorgehens bei der Konvolution betrachten wir die zwei Folgen:

$$x_0, x_1, x_2, x_3, x_4, x_5$$

und

$$h_0, h_1, h_2.$$

Die Konvolution ist dann,

$$y_n = x_n * h_n = \sum_{k=0}^n x_k h_{n-k},$$

mit dem Resultat,

$$\begin{aligned} y_0 &= x_0 h_0 \\ y_1 &= x_0 h_1 + x_1 h_0 \\ y_2 &= x_0 h_2 + x_1 h_1 + x_2 h_0 \\ y_3 &= x_1 h_2 + x_2 h_1 + x_3 h_0 \\ y_4 &= x_2 h_2 + x_3 h_1 + x_4 h_0 \\ y_5 &= x_3 h_2 + x_4 h_1 + x_5 h_0 \\ y_6 &= x_4 h_2 + x_5 h_1 \\ y_7 &= x_5 h_2 \end{aligned}$$

Wie wir im Folgenden sehen werden, entspricht die Konvolution von zwei Zeitreihen einer Korrelation, bei der die Reihenfolge der zweiten Folge invertiert wird. Im Allgemeinen wird das Resultat der Konvolution von zwei endlich langen Folgen mit N bzw. M Elementen die Länge $N + M - 1$ haben. Mit $N \geq M$ wird in der Praxis jedoch y_n meistens auf die Länge N beschränkt. Wegen der endlichen Länge der zwei Folgen, erstreckt sich tatsächlich für ein bestimmtes y_n die Summe über k jeweils vom grösseren der zwei Werte $n - (M - 1)$ und 0 bis zum kleineren der zwei Werte n und $N - 1$. Das heisst, dass für die ersten und letzten $M - 1$ Werte des Outputs die Summe nicht über die volle Länge M der kürzeren Folge geht. Sind h_n

die Impulsantwort eines Filters der Länge M und x_n die N Abtastwerte eines gegebenen Signals, dann entsprechen die ersten $M - 1$ Werte des Ausgangs der Einschwingzeit des Filters.

Die diskrete Konvolution kann auch in Matrix-Schreibweise als lineares Problem formuliert werden. Die Zahlenfolgen h_n , x_n und y_n können auch als Vektoren betrachtet werden und dann kann man eine Matrix H bilden, so dass die Faltung als Multiplikation einer Matrix mit einem Vektor dargestellt werden kann. Für das obige Beispiel sieht das wie folgt aus:

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix} = \begin{pmatrix} h_0 & 0 & 0 & 0 & 0 & 0 \\ h_1 & h_0 & 0 & 0 & 0 & 0 \\ h_2 & h_1 & h_0 & 0 & 0 & 0 \\ 0 & h_2 & h_1 & h_0 & 0 & 0 \\ 0 & 0 & h_2 & h_1 & h_0 & 0 \\ 0 & 0 & 0 & h_2 & h_1 & h_0 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix}$$

Hier haben wir die Länge von y_n , wie oben erwähnt, auf N beschränkt. In Kurzschreibweise kann man die Faltungsoperation also schreiben:

$$\mathbf{y} = \mathbf{H}\mathbf{x}.$$

Diese Art die Faltung auszuführen, erlaubt dann eine sehr effiziente Formulierung der später zu besprechenden Dekonvolution im Zeitbereich als lineares Inversionsproblem, zu dessen Lösung uns eine Vielzahl von Methoden aus der numerischen linearen Algebra zur Verfügung stehen².

Das Faltungsintegral

Im Kontinuum wird die Faltung sinngemäss als Integral formuliert:

$$g(t) = f(t) * h(t) = \int_{-\infty}^{\infty} f(\tau)h(t - \tau)d\tau.$$

In dieser Form ist die Operation wesentlich schwieriger nachzuvollziehen als im Diskreten. Eine anschauliche Darstellung findet sich aber im Buch von Scherbaum³.

²Gurrola, H., Baker, G.E., Minster, J. B. (1995). Simultaneous time-domain deconvolution with application to the computation of receiver functions. *Geophys. J. Int.*, 120, 537-543.

O'Dowd, R. J. (1990). Ill-conditioning and pre-whitening in seismic deconvolution. *Geophys. J. Int.*, 101, 489-491.

Oldenburg, D. W. (1981). A comprehensive solution to the linear deconvolution problem. *Geophys. J. R. astr. Soc.*, 65, 331-357.

Sipkin, A. A., Lerner-Lam, A. L. (1992). Pulse-shape distortion introduced by broadband deconvolution. *Bull. seism. Soc. Am.*, 82, 238-258.

³Scherbaum, F. (1996). *Of Poles and Zeros, Fundamentals of Digital Seismology*. Kluwer Academic Publishers, Fig. 2.4, p. 17.

Kapitel 3

Die Fourier Transformation

Einleitung

Die Fourier Transformation ist immer noch eines der wichtigsten Instrumente der Signalanalyse. Einerseits erlaubt sie es, Gesetzmässigkeiten eines Signals zu erkennen, die aus den Daten in ihrer ursprünglichen Form als Funktion z.Bsp. der Zeit oder des Raumes nicht direkt ersichtlich sind. Andererseits lassen sich gewisse mathematische Probleme (z.Bsp. Differentialgleichung) mittels Fourier Transformation einfacher und eleganter lösen und gewisse mathematische Operationen lassen sich mit deutlich geringerem Aufwand ausführen.

Die Fourier Analyse beruht auf der Tatsache, dass sich eine mathematische Funktion oder ein beobachtetes Signal als eine Summe (Ueberlagerung) von harmonischen Schwingungen darstellen lässt. Eine harmonische Schwingung $h_\omega(t)$ lässt sich mathematisch ganz allgemein durch

$$h_\omega(t) = C_\omega \cos(\omega t - \phi_\omega)$$

beschreiben. Sie ist eindeutig bestimmt durch ihre Amplitude C_ω , durch ihre Phase ϕ_ω und durch ihre Kreisfrequenz ω oder Frequenz f , wobei $\omega = 2\pi f$ ist. Unter Anwendung der Summationsformel für den Cosinus, erhält man

$$h_\omega(t) = C_\omega [\cos(\omega t) \cos \phi_\omega + \sin \phi_\omega \sin(\omega t)].$$

$C_\omega \cos \phi_\omega$ und $C_\omega \sin \phi_\omega$ können wir für eine gegebene Schwingung der Kreisfrequenz ω als zwei Konstanten a_ω und b_ω betrachten. Dann kann man die allgemeine harmonische Schwingung $h_\omega(t)$ auch als Linearkombination einer Cosinus- und einer Sinus-Schwingung darstellen:

$$h_\omega(t) = a_\omega \cos(\omega t) + b_\omega \sin(\omega t).$$

Die zugehörige Phase ergibt sich dann einfach aus

$$\tan \phi_\omega = \frac{\sin \phi_\omega}{\cos \phi_\omega} = \frac{b_\omega}{a_\omega}.$$

Da bekanntlich $\sin^2 \phi_\omega + \cos^2 \phi_\omega = 1$ ist, erhält man wieder die Amplitude von $h_\omega(t)$ als

$$C_\omega = \sqrt{a_\omega^2 + b_\omega^2}.$$

Wie man sich durch anwenden der obigen Summationsformel leicht vergewissern kann, entspricht $\phi_\omega = 0$ einer reinen Cosinus-Schwingung und $\phi_\omega = \pi/2$ einer reinen Sinus-Schwingung. Man sagt, dass zwischen dem Sinus und dem Cosinus eine Phasenverschiebung oder Phasendrehung von $\phi = \pi/2$ besteht.

Die nebenstehende Abbildung soll nun an einem Beispiel veranschaulichen, wie die Ueberlagerung von einzelnen harmonischen Schwingungen ein Signal oder eine Funktion entstehen lässt, dessen Form weit entfernt von einer harmonischen Schwingung zu sein scheint. Links oben ist das knapp 1 Sekunde lange, aus 32 diskreten Punkten bestehende Signal abgebildet. Rechts oben ist das sogenannte Amplitudenspektrum, welches mittels der diskreten Fourier Transformation aus dem Signal berechnet wurde. Es stellt die Amplitude C als Funktion der Frequenz f dar. Für die vollständige Beschreibung der Fourier Transformation fehlt noch die Darstellung der Phase ϕ als Funktion der Frequenz.

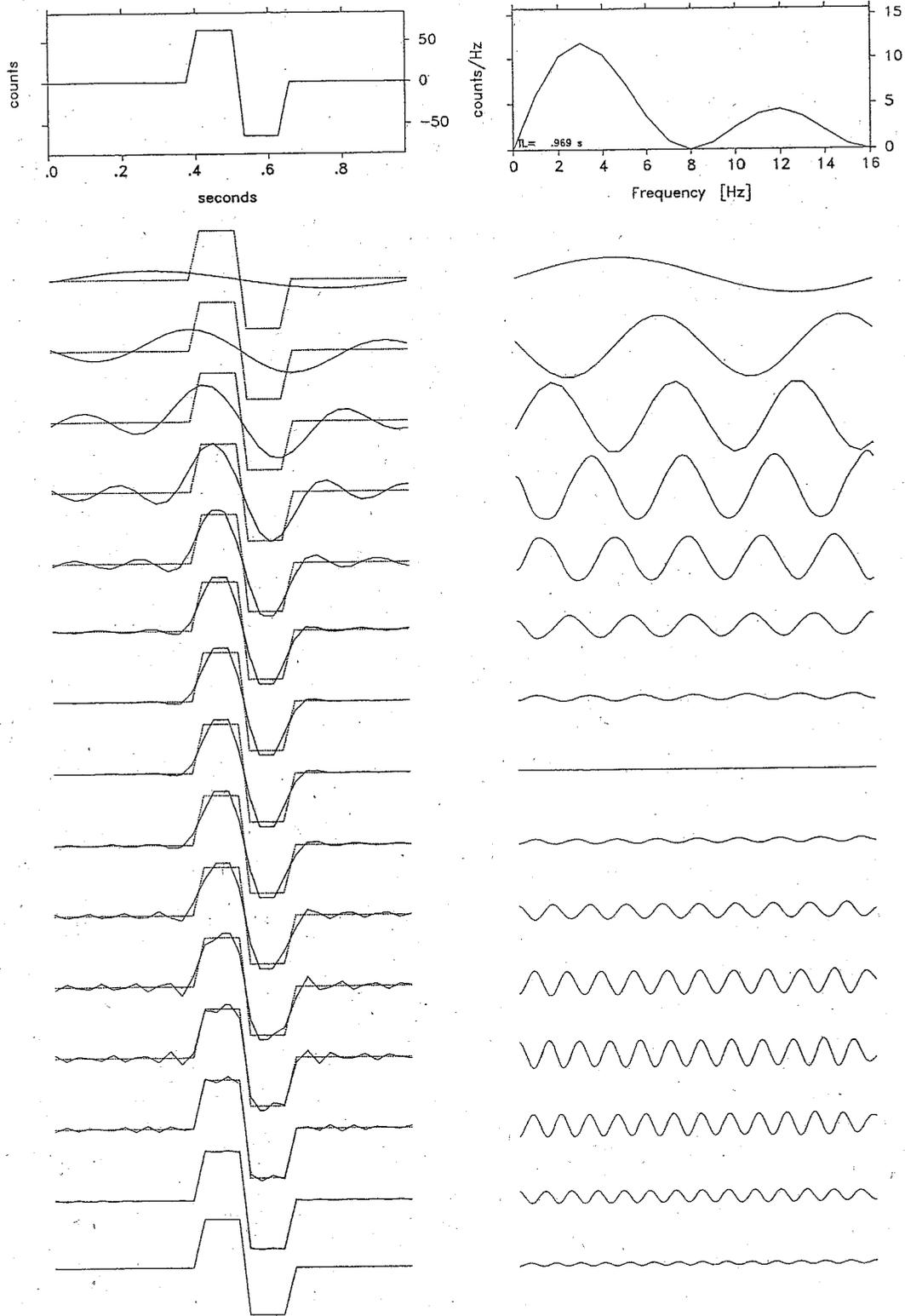
Stattdessen sind unter dem Amplitudenspektrum die einzelnen harmonischen Schwingungen mit den Frequenzen 1 bis 15 Hz dargestellt. Da die Dauer jeder Schwingung knapp 1 Sekunde beträgt, kann man durch abzählen der Maxima oder Minima die Frequenzangaben leicht verifizieren. Ausserdem entsprechen die Amplitudenverhältnisse zwischen den einzelnen Schwingungen genau dem Verlauf des abgebildeten Amplitudenspektrums. Die Phase als Funktion der Frequenz wirkt sich darin aus, dass jede einzelne Schwingung an einem unterschiedlichen Punkt relativ zum jeweiligen ersten Maximum oder Minimum anfängt. Im linken Teil der Abbildung ist das Resultat der laufenden Aufsummierung der nebenstehenden harmonischen Schwingung dargestellt. Mit jeder zusätzlichen Frequenz gleicht sich die Summe der Schwingungen mehr und mehr dem ursprünglichen Signal an. Da die Amplitude bei der Frequenz von 16 Hz in diesem Fall gleich 0 ist, ist die Uebereinstimmung mit dem gegebenen Signal schon bei 15 Hz perfekt.

Mathematisch ausgedrückt haben wir nun das diskrete Signal x_n , ($n = 0, \dots, 31$) als eine Summe

$$x_n = \sum_{k=0}^{N-1} a_k \cos\left(\frac{2\pi kn}{N}\right) + b_k \sin\left(\frac{2\pi kn}{N}\right)$$

erzeugt, wobei in diesem Fall $N = 16$. Der Witz der Fourier Transformation ist nun der, dass sie uns ermöglicht in eindeutiger Weise die Koeffizienten a_k und b_k aus den gegebenen Werten von x_n zu berechnen.

Vor der Behandlung der diskreten Fourier Transformation, die in der digitalen Signalverarbeitung Anwendung findet, ist es sinnvoll sich mit der Theorie der Fourier Reihe und des Fourier Integrals im Kontinuum vertraut zu machen.



Figur 3.1: Beispiel einer Signalsynthese aus harmonischen Schwingungen.

Die Fourier Reihe

Sei $h(t)$ eine kontinuierliche Funktion von t , welche periodisch oder periodisch fortgesetzt ist mit Periode T und für $0 < t \leq T$ stückweise stetig sowie von begrenzter Schwankung ist. Dann lässt sich $h(t)$ als eine unendliche Reihe von harmonischen Schwingungen mit diskreten Frequenzen darstellen:

$$h(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos\left(\frac{2\pi kt}{T}\right) + b_k \sin\left(\frac{2\pi kt}{T}\right).$$

Wie lassen sich nun die Koeffizienten a_0 , a_k und b_k , ($k = 1, \dots, \infty$) in eindeutiger Weise aus der Funktion $h(t)$ bestimmen?

Um die Notation zu vereinfachen, schreiben wir obige Gleichung für $h(t)$ wie folgt um:

$$h(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(\omega_n t) + b_n \sin(\omega_n t),$$

wobei wir $2\pi k/T = \omega_n$ mit $k = n$ gesetzt haben. Multiplizieren wir nun $h(t)$ mit $\cos(\omega_m t)$, wobei $\omega_m = 2\pi m/T$, und integrieren wir von 0 bis T , dann erhalten wir

$$\begin{aligned} \int_0^T h(t) \cos(\omega_m t) dt &= \frac{a_0}{2} \int_0^T \cos(\omega_m t) dt + \\ &\sum_{n=1}^{\infty} a_n \int_0^T \cos(\omega_n t) \cos(\omega_m t) dt + \\ &\sum_{n=1}^{\infty} b_n \int_0^T \sin(\omega_n t) \cos(\omega_m t) dt. \end{aligned}$$

Auf Grund der sogenannten Orthogonalitätsbeziehung der Sinus und Cosinus Funktionen (durch Anwendung von elementaren trigonometrischen Beziehungen und einfachen Integrationsregeln zu beweisen) erhalten wir

$$\int_0^T h(t) \cos(\omega_m t) dt = \begin{cases} \frac{a_m T}{2} & (m = n) \\ 0 & (m \neq n) \end{cases}$$

Durch analoges Vorgehen wie oben aber mit $\sin(\omega_m t)$ an Stelle von $\cos(\omega_m t)$ erhalten wir

$$\int_0^T h(t) \sin(\omega_m t) dt = \begin{cases} \frac{b_m T}{2} & (m = n) \\ 0 & (m \neq n) \end{cases}$$

Da also nur die Terme mit $m = n$ etwas zu den Integralen beitragen, können wir wieder m und n durch k ersetzen und mit $\omega_k = 2\pi k/T$ erhalten wir die gesuchten Beziehungen für a_k und b_k :

$$a_k = \frac{2}{T} \int_0^T h(t) \cos\left(\frac{2\pi kt}{T}\right) dt \quad \text{und} \quad b_k = \frac{2}{T} \int_0^T h(t) \sin\left(\frac{2\pi kt}{T}\right) dt.$$

In komplexer Schreibweise lässt sich die Fourier Reihe auch wie folgt schreiben:

$$h(t) = \sum_{k=-\infty}^{\infty} c_k e^{j\frac{2\pi kt}{T}},$$

$$c_k = \frac{1}{T} \int_0^T h(t) e^{-j\frac{2\pi kt}{T}} dt.$$

Die komplexen Koeffizienten c_k ergeben sich aus den reellen Koeffizienten a_k und b_k durch Anwendung der Euler'schen Gleichungen für Sinus und Cosinus:

$$c_0 = \frac{a_0}{2}; \quad c_k = \frac{a_k - jb_k}{2}; \quad c_{-k} = \frac{a_k + jb_k}{2}, \quad k > 0.$$

Keht man bei der Herleitung der komplexen Darstellung die Zuordnung der Koeffizienten um,

$$c'_k = \frac{a_k + jb_k}{2}; \quad c'_{-k} = \frac{a_k - jb_k}{2}, \quad k > 0,$$

dann erhält man für $h(t)$ eine Reihenentwicklung mit negativem Exponenten,

$$h(t) = \sum_{k=-\infty}^{\infty} c'_k e^{-j\frac{2\pi kt}{T}},$$

mit einem entsprechend modifizierten Ausdruck für c'_k . Die Wahl zwischen der einen oder anderen Formulierung ist absolut willkürlich und hat lediglich auf die Richtung der Phasendrehung einen Einfluss.

Betrachten wir noch einmal die erste Formulierung

$$h(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos\left(\frac{2\pi kt}{T}\right) + b_k \sin\left(\frac{2\pi kt}{T}\right)$$

und bezeichnen die einzelnen Summanden dieser Reihenentwicklung mit h_k . Aus der Beziehung

$$a_k = \frac{2}{T} \int_0^T h(t) \cos\left(\frac{2\pi kt}{T}\right) dt$$

sehen wir, dass

$$a_0 = \frac{2}{T} \int_0^T h(t) dt$$

und somit, dass der erste Term der Reihenentwicklung für $h(t)$ dem Mittelwert von $h(t)$ entspricht:

$$h_0(t) = \frac{a_0}{2} = \overline{h(t)}.$$

Für $k = 1$ ist

$$h_1(t) = a_1 \cos\left(\frac{2\pi t}{T}\right) + b_1 \sin\left(\frac{2\pi t}{T}\right).$$

Aus der Tatsache, dass $0 \leq t \leq T$, folgt in diesem Fall, dass das Argument der trigonometrischen Funktionen von 0 bis 2π geht. Somit entspricht der Term $h_1(t)$ genau einer harmonischen Schwingung mit der Periode $T_1 = T$, bzw. der Frequenz $f_1 = 1/T$. In gleicher Weise kann man zeigen, dass der Term $h_2(t)$ genau zwei Schwingungen mit der Periode $T/2$, bzw. der Frequenz $2/T$ entspricht. Daraus folgt, dass die anfangs vorausgesetzte Periodizität bzw. die Grundperiode T sowohl die kleinste im Spektrum enthaltene Frequenz f_1 , als auch den Abstand zwischen zwei benachbarten Spektralwerten, also die Auflösung Δf , bedingt:

$$f_1 = \Delta f = f_{i+1} - f_i = 1/T.$$

Wenn T gegen ∞ geht, dann geht f_1 gegen 0 und Δf geht in ein infinitesimales df über. Dies führt uns zwanglos auf das Fourier Integral.

Das Fourier Integral

Definition

Lässt man die Periode T gegen unendlich gehen, dann wird aus der Fourier Reihe ein Integral über ein kontinuierliches Frequenzspektrum:

$$h(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H(\omega) e^{j\omega t} d\omega,$$

$$H(\omega) = \int_{-\infty}^{\infty} h(t) e^{-j\omega t} dt.$$

Eine hinreichende und in der Praxis sehr brauchbare Bedingung für die Existenz des kontinuierlichen Fourier Integrals ist, dass die Funktion $h(t)$ eine endliche Energie aufweist:

$$\int_{-\infty}^{\infty} |h(t)|^2 dt < \infty.$$

Die zwei Integrale für $h(t)$ und $H(\omega)$ werden als kontinuierliche Fourier Transformation bezeichnet: eine Funktion der Zeit (oder einer beliebigen anderen Variable, z.Bsp. des Raumes) wird in eine Funktion der Frequenz übertragen und umgekehrt. Diese Beziehung wird oft durch die Korrespondenz $h(t) \leftrightarrow H(\omega)$ dargestellt.

Wenn t die Zeit ist, dann ist in obiger Formulierung $\omega = 2\pi f$ die Kreisfrequenz (Bogenmass/sekunden, engl. radians/s). Einfacher ist es, wenn man direkt über die Frequenz f (Hz) integriert, weil dann der Faktor $1/2\pi$ vor dem Integral wegfällt. Dann lautet die Fourier-(Hin)Transformation,

$$\mathcal{F}\{h(t)\} = H(f) = \int_{-\infty}^{\infty} h(t) e^{-j2\pi ft} dt,$$

und die Rücktransformation,

$$\mathcal{F}^{-1}\{H(f)\} = h(t) = \int_{-\infty}^{\infty} H(f)e^{+j2\pi ft} df.$$

Die Fourier Transformierte $H(f)$ ist im Allgemeinen eine komplexe Grösse, welche aus der Summe eines Real- und eines Imaginärteils besteht:

$$H(f) = \Re\{H(f)\} + j\Im\{H(f)\} = \int_{-\infty}^{\infty} h(t) \cos(2\pi ft) dt - j \int_{-\infty}^{\infty} h(t) \sin(2\pi ft) dt.$$

Also kann sie auch als ein Produkt des Betrages und des Argumentes dargestellt werden:

$$H(f) = |H(f)|e^{j\phi(f)} = |H(f)|[\cos \phi(f) + j \sin \phi(f)].$$

Der Betrag (engl. modulus) wird als Amplituden-Spektrum (oder Amplitudendichte) bezeichnet,

$$|H(f)| = \sqrt{\Re\{H(f)\}^2 + \Im\{H(f)\}^2},$$

und das Argument ist das Phasen-Spektrum,

$$\phi(f) = \arctan \frac{\Im\{H(f)\}}{\Re\{H(f)\}}.$$

In der Literatur findet man ebenso oft auch die Definition der Fourier Transformation mit vertauschten Vorzeichen im Exponent. Wie im Zusammenhang mit der komplexen Darstellung der Fourier Reihe schon gesagt, sind beide Formulierungen absolut gleichwertig. Es ist einzig darauf zu achten, dass sich dann auch in den Beziehungen, welche die Phase betreffen, andere Vorzeichen ergeben.

Wichtige Eigenschaften der FT

Sei im Folgenden $H(f) = \mathcal{F}\{h(t)\}$ die Fourier Transformation von $h(t)$. Dann lassen sich durch Anwendung der Definition der Fourier Transformation, bzw. ihrer Rücktransformation, sehr einfach folgende Beziehungen beweisen:

Reziprozität

$$\mathcal{F}\{H(t)\} = h(-f),$$

was bedeutet, dass $h(-f)$ die Fourier Transformierte von $H(t)$ ist. Verwendet man die Kreisfrequenz ω anstelle von f , dann lautet letztere Beziehung,

$$\mathcal{F}\{H(t)\} = 2\pi h(-\omega).$$

Linearität

$$\mathcal{F}\{c_1 h(t) + c_2 g(t)\} = c_1 H(f) + c_2 G(f)$$

Nullpunktverschiebung in der Zeit

$$\mathcal{F}\{h(t + t_0)\} = e^{+j2\pi t_0 f} H(f)$$

Nullpunktverschiebung in der Frequenz

$$\mathcal{F}\{e^{+j2\pi f_0 t} h(t)\} = H(f - f_0)$$

Skalenänderung

$$\mathcal{F}\{h(ct)\} = \frac{1}{c} H\left(\frac{f}{c}\right)$$

Ableitung

$$\mathcal{F}\{h'(t)\} = j2\pi f H(f)$$

Integral

$$\mathcal{F}\left\{\int h(t) dt\right\} = \frac{1}{j2\pi f} H(f)$$

Man beachte, dass wenn die FT mit positivem Vorzeichen im komplexen Exponenten definiert ist (und die Rücktransformation mit negativem Vorzeichen), dann bei der Nullpunktverschiebung, der Ableitung und dem Integral das j zu einem $-j$ wird (dies sind eben Beziehungen, welche eine Phasenverschiebung verursachen).

Faltung Die Faltung (Engl. *convolution*) von zwei Funktionen $h(t)$ und $g(t)$ ist definiert als,

$$h(t) * g(t) = \int_{-\infty}^{\infty} h(\tau) g(t - \tau) d\tau.$$

Es gilt dann die folgende Beziehung:

$$\mathcal{F}\{h(t) * g(t)\} = H(f)G(f).$$

Der Beweis dieses Satzes folgt aus der Definition der FT, durch Vertauschen der Reihenfolge der Integration und Anwendung des Verschiebungssatzes. Es gilt auch die Umkehrung dieses Satzes:

$$\mathcal{F}\{h(t)g(t)\} = H(f) * G(f).$$

Das heisst, die Fourier Transformation des Produktes von zwei Funktionen der Zeit ist gleich der Faltung der einzelnen Fourier transformierten:

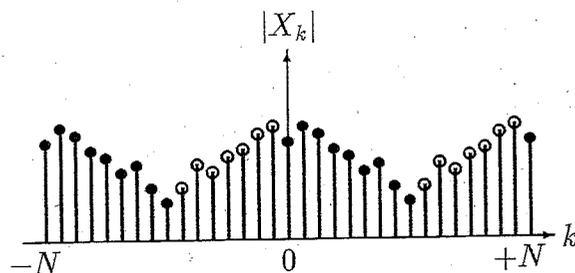
$$H(f) * G(f) = \int_{-\infty}^{\infty} H(\sigma)G(f - \sigma)d\sigma.$$

Die Faltung ist eine der wichtigsten Operationen in der Signalverarbeitung. Zum Beispiel, ergibt sich das Ausgangssignal eines Filters aus der Faltung des Eingangssignals mit der Filterfunktion, und ein Seismogramm kann als Faltung der Quellfunktion mit der Green'schen Funktion des Uebertragungsmediums und mit der Impulsantwort des Seismometers dargestellt werden. Was im Zeitbereich eine komplizierte und rechenintensive Operation ist, reduziert sich im Frequenzbereich auf eine einfache Multiplikation der entsprechenden Spektren. Selbstverständlich gilt auch die Umkehrung des Faltungssatzes: einer Multiplikation im Zeitbereich entspricht eine Faltung im Frequenzbereich.

Symmetrie-Eigenschaften

Eine Funktion $h(t)$ wird als *gerade* bezeichnet wenn $h(-t) = h(t)$ und als *ungerade* wenn $h(-t) = -h(t)$. Somit ist $\cos \omega t$ eine gerade Funktion und $\sin \omega t$ ungerade. Wenn $h(t)$ reell ist, dann gilt $H(-f) = H^*(f)$ ($*$ = konjugiert komplex). Folgende Symmetrie-Eigenschaften erleichtern oft die Berechnung der Fourier Transformation:

$h(t)$	$H(f)$
gerade	gerade
ungerade	ungerade
reell und gerade	reell und gerade
reell und ungerade	imaginär und ungerade



Figur 3.2: Betrag des diskreten Fourier Spektrums einer reellen diskreten Funktion mit $N = 16$. Die leeren Kreise entsprechen den konjugiert komplexen Werten der vollen Kreise.

Die diskrete Fourier Transformation

Betrachten wir nun eine Funktion (in unserem Fall ein Signal als eine Funktion der Zeit) dessen Werte x_n nur über ein beschränktes Intervall T und zu diskreten Zeitpunkten $n\Delta t$ gegeben sind (Δt ist das Abtastintervall und $n = 0, \dots, N - 1$). Die Abtastrate ist dann $f_s = 1/\Delta t$. Die Diskretisierung der Integrale in der kontinuierlichen Fourier Transformation führt auf die *Zeitdiskrete Fourier-Reihe*, welche den Uebergang zwischen diskretem Signal und diskretem Spektrum darstellt:

$$\begin{aligned} dt &\longrightarrow \Delta t \\ t &\longrightarrow t_n = n\Delta t \\ h(t) &\longrightarrow h(t_n) = x_n \\ df &\longrightarrow \Delta f = \frac{1}{N\Delta t} = \frac{1}{T} \\ f &\longrightarrow f_k = k\Delta f = \frac{k}{N\Delta t} \\ H(f) &\longrightarrow H(f_k) = X_k \end{aligned}$$

$$H(f) = \int_{-\infty}^{\infty} h(t) e^{-j2\pi ft} dt \longrightarrow X_k = \Delta t \sum_{n=0}^{N-1} x_n e^{-j\frac{2\pi kn}{N}}$$

$$h(t) = \int_{-\infty}^{\infty} H(f) e^{+j2\pi ft} df \longrightarrow x_n = \frac{1}{N\Delta t} \sum_{k=0}^{N-1} X_k e^{+j\frac{2\pi kn}{N}}$$

Dabei ist N die Anzahl Punkte in der Zeitserie x_n und Δf ist nicht die Abtastrate f_s sondern das Frequenzintervall zwischen zwei benachbarten Frequenzwerten im Spektrum.

Man beachte zweierlei:

- Die komplexe Folge $\{X_k\}$ ist periodisch mit der Periode N .
- Für reelle Signale $\{x_n\}$ gilt $X_{-k} = X_k^*$ für $(0 \leq k \leq N - 1)$.

Aus diesen zwei Tatsachen folgt, dass es für reelle Signale auch nur $N/2 + 1$ unabhängige Spektralkoeffizienten gibt (siehe Abbildung 3.2). Die höchste

im diskreten Spektrum enthaltene Frequenz folgt aus $f_k = k/(N\Delta t)$ mit $k = N/2$ und wird als *Nyquist Frequenz*, f_{Nyq} bezeichnet:

$$f_{Nyq} = \frac{N}{2N\Delta t} = \frac{1}{2\Delta t} = \frac{f_s}{2}.$$

Dies führt zu einem der wichtigsten Ergebnisse der digitalen Signalanalyse: *Die höchste in einem diskreten Signal enthaltene Frequenz ist gleich der halben Abtastrate.*

Wir sprechen in diesem Fall von einem Signal mit beschränktem Frequenzumfang (engl. band-limited signal). Wegen der Beschränkung des Signals sowohl in der Zeit als auch in der Frequenz, entspricht einer diskreten Zeitserie endlicher Länge auch ein diskretes Frequenzspektrum endlicher Länge.

Normierung der DFT

In der Literatur wird die diskrete Fourier Transformation, DFT, oft ohne den Faktor Δt angegeben. Tatsächlich ist die DFT auch gültig ohne diesen Skalierungsfaktor, dieser ist aber unerlässlich falls es auf die richtigen Einheiten ankommt. Nur dann entspricht X_n dem kontinuierlichen Spektrum und hat $|X_n|$ wirklich die Einheit einer spektralen Amplituden-Dichte (z.Bsp. m/Hz).

Beziehung zwischen diskreter Fourier Transformation, Fourier Reihe und Fourier Integral

Ausgehend von der diskreten Fourier Transformation, welche auch als *zeitdiskrete Fourier Reihe* bezeichnet wird, lässt man Δt gegen 0 gehen, dann geht f_N gegen ∞ und man erhält die Fourier Reihe (genauer, die *zeitkontinuierliche Fourier Reihe*). Lässt man nun die Signallänge T gegen ∞ gehen, dann geht Δf gegen 0, bzw. gegen df , und man erhält das Fourier Integral (oder genauer, das *zeitkontinuierliche Fourier Integral*).

Wie man sich leicht überlegen kann, kann man den Uebergang von der diskreten Fourier Transformation zum Fourier Integral auch über das sogenannte *zeitdiskrete Fourier Integral* ausführen, indem man das endliche Δt beibehält aber die Signallänge gegen ∞ und somit das endliche Δf gegen das infinitesimale df gehen lässt.

Wichtig ist die Erkenntnis, dass die Diskretisierung der Zeit mit der Abtastperiode Δt zu einer Beschränkung der maximalen Frequenz $f_{max} = f_{Nyq} = 1/(2\Delta t)$ führt und dass die Begrenzung der Zeitdauer T eine Diskretisierung der Frequenz mit einer Auflösung von $\Delta f = 1/T$ bedingt.

	Diskrete FT	Fourier Reihe	Fourier Integral
Auflösung Δt	Diskret (x_n) Δt	Kontinuierlich dt	Kontinuierlich dt
Auflösung Δf	Diskret (c_k) $\frac{1}{N\Delta t}$	Diskret (c_k) $\frac{1}{T}$	Kontinuierlich df
Signallänge	Endlich Periodizität = $N\Delta t$	Endlich Periodizität = T	Unendlich
Max. Frequenz	Endlich $f_{Nyq} = \frac{1}{2\Delta t}$	Unendlich	Unendlich

Die FFT

Die schnelle Fourier Transformation (engl. fast Fourier transform, FFT) ist nicht eine weitere Transformation, sondern nur ein Algorithmus um die diskrete Fourier Transformation mit relativ geringem Rechenaufwand durchzuführen. Um die gewünschte hohe Rechengeschwindigkeit zu erreichen, arbeiten die meisten FFT Routinen mit einer Signallänge von $N = 2^n$ Punkten. Dabei wird das gegebene Signal mit Nullen bis zur nächsten Zweierpotenz verlängert (zero padding). Dadurch wird der Informationsgehalt des resultierenden Spektrums nicht erhöht. Das Spektrum wird zwar feiner abgetastet, aber die Erhöhung der Auflösung ist nur optisch. Insbesondere sind die Amplitudenwerte des Spektrums nur von der Dauer des tatsächlich für die Transformation ausgewählten Signals abhängig und nicht von der Anzahl angehängter Nullen.

Sei Δt das Abtastintervall, dann ist $1/\Delta t$ die Abtastrate. Unabhängig von der Signallänge ist der Frequenzumfang der FFT nur von der Abtastrate abhängig und umfasst immer den Bereich von der Frequenz 0 bis zur Nyquist-Frequenz, $f_{Nyq} = 1/(2\Delta t)$. Bei einer Signallänge von $N = 2^n$ diskreten Abtastwerten besteht das Spektrum aus $N/2 + 1 = 2^{n-1} + 1$ diskreten Spektralwerten. Das heisst, der Frequenzbereich zwischen 0 und der Nyquist-Frequenz ist in $N/2$ gleichlange Frequenzintervalle Δf eingeteilt. Dabei gilt:

$$\Delta f = f_{Nyq}/2^{n-1} = \frac{f_{Nyq}}{N/2} = \frac{1}{N\Delta t}$$

Man beachte, dass, sowohl hier als auch in der folgenden Tabelle, N die Anzahl Datenpunkte nach Verlängerung mit Nullen bis zur nächsten Zweierpotenz ist!

Wegen der Nichtexistenz des $\log 0$ ist in einer logarithmischen Darstellung der Frequenzachse die kleinste Frequenz gleich Δf . Ausserdem ist der Bereich tiefer Frequenzen stark gedehnt (geringe Punktdichte) während der Bereich hoher Frequenzen stark gedrängt ist (hohe Punktdichte). Das Spektrum kann natürlich auch als Funktion der Periode ($1/f$) dargestellt werden, wobei dann auch wieder die Nichtlinearität der Skala beachtet werden muss.

Führt man die FFT für eine reelle Zeitserie durch, besteht das Resultat der komplexen FFT aus Paaren von Cosinus- und Sinus-Termen, die jeweils paarweise einem Frequenzwert entsprechen (Cosinus = reell, Sinus = imaginär). Die Sinus-Terme zur Frequenz 0 und zur Nyquist-Frequenz sind immer gleich 0, so dass zur Speicherplatz-Optimierung oft der Cosinus-Term der Nyquist-Frequenz an Stelle des Sinus-Terms der Frequenz 0 abgespeichert wird.

Die FFT in MATLAB

Die FFT einer Zeitreihe x_n , die N Werte lang ist, wird in MATLAB mit dem Befehl `fft(x,M)` ausgeführt. Wenn $M < N$ ist, dann wird die Zeitreihe gekürzt, wenn $M > N$ ist, dann werden entsprechend viele 0 angehängt. Ohne Angabe von M , also mit `fft(x)`, wird die FFT mit N Werten ausgeführt. Weder N noch M müssen einer Zweierpotenz entsprechen, aber für den Rechenaufwand ist es von Vorteil: Im ungünstigsten Fall wird aus der FFT eine langsame DFT. Sei im Folgenden $M = N$, dann ist

$$\text{fft}(x,N) = \sum_{n=0}^{N-1} x_n e^{-j \frac{2\pi kn}{N}}$$

Wird ein Spektrum mit korrekten Einheiten gewünscht, muss das Resultat noch mit der Länge des Abtastintervalls Δt multipliziert werden.

Mit $\text{Delta} = \Delta t$ ist dann

$$X = \text{Delta} * \text{fft}(x,N) = \Delta t \sum_{n=0}^{N-1} x_n e^{-j \frac{2\pi kn}{N}} = X_k$$

Der komplexe Vektor X hat N Werte. Zur Darstellung des vollständigen Spektrums von der Frequenz 0 bis zur Nyquist-Frequenz genügen aber die ersten $N/2 + 1$ Werte. Mit der Abtastrate f_s , sei $F_{\text{Nyq}} = F_{N/2} = f_s/2 = 1/(2\Delta t)$ die Nyquist Frequenz und $N_f = N/2 + 1$ die Anzahl Frequenzwerte von 0 bis F_{Nyq} , dann kann man zur Darstellung des Resultates der FFT den Vektor mit den entsprechenden Frequenzwerten wie folgt erstellen:

$$f = \text{linspace}(0, F_{\text{Nyq}}, N_f);$$

oder mit dem Frequenzintervall $df = \Delta f = 1/(N\Delta t)$,

$$f = 0:df:F_{\text{Nyq}};$$

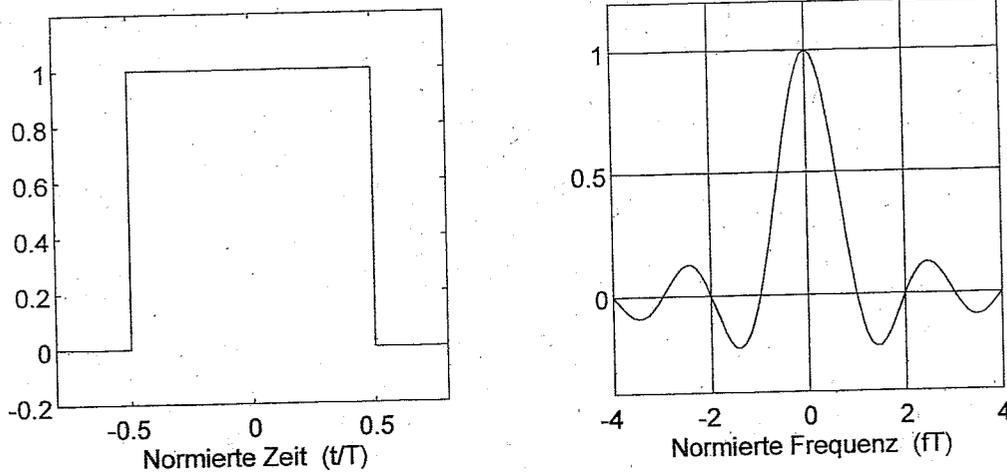
Das Amplitudenspektrum kann man dann darstellen mit

$$\text{plot}(f, \text{abs}(X(1:N_f)))$$

Die inverse FFT wird mit `ifft(X)` ausgeführt, wobei

$$y = \text{ifft}(X) = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{+j \frac{2\pi kn}{N}}$$

Das Resultat y ist ein komplexer Vektor und die ursprüngliche reelle Zeitreihe x_n erhält man mit $x = \text{real}(y)$, oder mit $x = \text{real}(y) / \text{Delta}$, falls X korrekt mit Δt normiert worden ist. Falls X_k das Resultat von $X = \text{fft}(x,M)$ und $M \neq N$ war, ist die Länge von x_n als Resultat von `ifft(X)` nun aber gleich M statt N .



Figur 3.3: Die Rechteckfunktion und der Realteil ihrer Fourier-Transformierten. Da die so definierte Rechteckfunktion eine reelle und gerade Funktion ist, verschwindet der Imaginärteil der FT für alle Frequenzen und somit stellt der Realteil die vollständige Transformierte dar.

Beispiele

FT einer Rechteckfunktion

Eine Rechteckfunktion der Höhe 1 und der Breite $T = 2\tau$ sei wie folgt definiert:

$$h(t) = \begin{cases} 1 & |t| < \tau = T/2 \\ 0 & |t| > \tau = T/2 \end{cases}$$

Die Fourier Transformation ist dann:

$$H(\omega) = \int_{-\infty}^{\infty} h(t)e^{-j\omega t} dt = \int_{-\tau}^{\tau} 1e^{-j\omega t} dt,$$

und da h reell und gerade ist folgt,

$$H(\omega) = \int_{-\tau}^{\tau} \cos \omega t dt = \frac{1}{\omega} \sin \omega t \Big|_{-\tau}^{\tau} = 2 \frac{\sin \omega \tau}{\omega}.$$

Durch Multiplikation mit τ/τ und mit den Substitutionen $\tau = T/2$ und $\omega = 2\pi f$ erhält man das Ergebnis in Form einer sogenannten *sinc*-Funktion:

$$H(f) = T \frac{\sin(\pi T f)}{\pi T f} = T \operatorname{sinc}(\pi T f).$$

Unter Anwendung der Regel von L'Hospital erhält man auch den Wert für $f = 0$:

$$H(0) = \lim_{f \rightarrow 0} T \frac{\sin(\pi T f)}{\pi T f} = \lim_{f \rightarrow 0} T \frac{\pi T \cos(\pi T f)}{\pi T} = T.$$

Beachte, dass $H(0)$ gleich der Fläche des Rechteckes (also gleich dem Integral der Funktion) ist. Dies ist eine allgemeine Eigenschaft der FT!

Die Nulldurchgänge von $H(f)$ befinden sich bei den Werten

$$\pi T f_n = n\pi, \quad |n| = 1, 2, 3, \dots$$

woraus folgt

$$f_n = \frac{n}{T}.$$

Mit $n = 1$ sieht man, dass je schmaler (zeitlich kürzer) das Rechteck ist, desto breiter (frequenzmässig unschärfer) ist das Spektrum. Dies ist ein Beispiel der allgemein gültigen Unschärferelation der Fourier Transformation.

Statt die Höhe des Rechteckes gleich 1 zu setzen soll sie im Folgenden gleich $1/2\tau$ sein:

$$h(t) = \begin{cases} \frac{1}{2\tau} & |t| < \tau \\ 0 & |t| > \tau \end{cases}$$

Dann ist die Fläche:

$$\int_{-\infty}^{\infty} h(t) dt = 2\tau \frac{1}{2\tau} = 1.$$

Die Fourier Transformierte ist dann:

$$H(\omega) = \frac{\sin \omega\tau}{\omega\tau}.$$

Lassen wir τ gegen 0 gehen, so dass der Impuls immer schmaler und höher wird, dann erhalten wir für alle ω :

$$\lim_{\tau \rightarrow 0} H(\omega) = \lim_{\tau \rightarrow 0} \frac{\sin \omega\tau}{\omega\tau} = 1.$$

Der Grenzwert $\tau \rightarrow 0$ in obiger Definition von $h(t)$ führt genau auf die Definition des *Dirac Delta Impulses* (Delta-Funktion, Dirac-Stoss):

$$\delta(t) = \begin{cases} \infty & t = 0 \\ 0 & t \neq 0 \end{cases}$$

mit der Eigenschaft,

$$\int_{-\infty}^{\infty} \delta(t) dt = 1.$$

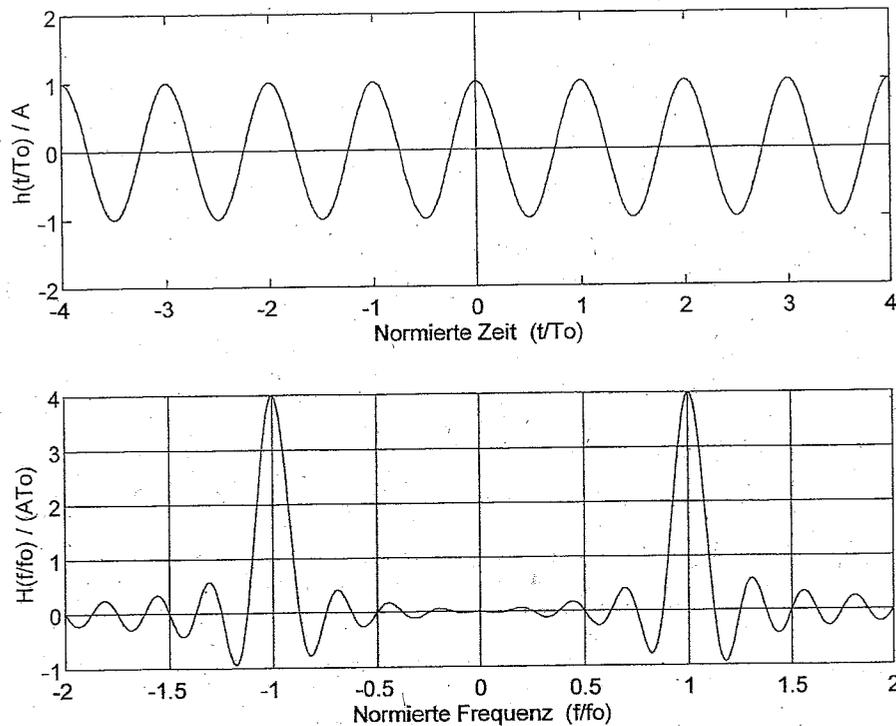
Somit erhält man für die Fourier Transformation des Delta-Impulses:

$$\int_{-\infty}^{\infty} \delta(t) e^{-j\omega t} dt = 1.$$

Beachte, mit $g(\tau) = e^{-j\omega\tau}$ hat diese Beziehung für die Fourier Transformation des Delta-Impulses die Form:

$$\int_{-\infty}^{\infty} \delta(0 - \tau) g(\tau) d\tau = g(0).$$

Dies ist eine allgemeine Eigenschaft der δ -Funktion, welche auch als Definition verwendet wird.



Figur 3.4: Eine zeitlich beschränkte Cosinus-Funktion und der Realteil ihrer Fourier-Transformierten. Da die so definierte harmonische Schwingung eine reelle und gerade Funktion ist, verschwindet der Imaginärteil der FT für alle Frequenzen und somit stellt der Realteil die vollständige Transformierte dar.

FT einer harmonischen Schwingung

Das Fourier Integral einer unendlich langen harmonischen Schwingung existiert nicht, da die Energie eines solchen Signals nicht endlich ist. Da sie aber periodisch ist, kann man sie natürlich in eine Fourier Reihe zerlegen und erhält zu der gegebenen Frequenz den entsprechenden Spektralwert. In der Praxis wendet man auch für solche Signale mit nicht endlicher Energie die Fourier Transformation an, indem man sich ein mehr oder weniger willkürliches Segment davon herausschneidet. Im Folgenden wollen wir die Fourier Transformation eines in dieser Art künstlich beschränkten Signals untersuchen.

Sei $h(t)$ ein Cosinus-Signal mit der Amplitude A und der Periode T_0 , bzw. mit der Eigenfrequenz $f_0 = 1/T_0$. Das betrachtete Zeitintervall T erstrecke sich über M Perioden, so dass $T = MT_0$. Um die Symmetrie-Eigenschaft der FT im Falle einer geraden Funktion ausnützen zu können, definieren wir das betrachtete Zeitintervall, so dass $-MT_0/2 < t < MT_0/2$. Mit $\omega_0 = 2\pi f_0 = 2\pi/T_0$ gilt:

$$h(t) = \begin{cases} A \cos(\omega_0 t) & |t| < \frac{MT_0}{2} = \frac{M\pi}{\omega_0} \\ 0 & |t| > \frac{MT_0}{2} = \frac{M\pi}{\omega_0} \end{cases}$$

Die Fourier Transformation ist dann:

$$H(\omega) = \int_{-\infty}^{\infty} h(t)e^{-j\omega t} dt = 2 \int_0^{\frac{M\pi}{\omega_0}} A \cos(\omega_0 t) e^{-j\omega t} dt,$$

und da h reell und gerade ist folgt,

$$H(\omega) = 2 \int_0^{\frac{M\pi}{\omega_0}} A \cos(\omega_0 t) \cos(\omega t) dt.$$

Die Lösung des Integrals ist:

$$H(\omega) = 2 \left\{ \frac{A \sin[(\omega_0 - \omega)t]}{2(\omega_0 - \omega)} + \frac{A \sin[(\omega_0 + \omega)t]}{2(\omega_0 + \omega)} \right\} \Bigg|_0^{\frac{M\pi}{\omega_0}}$$

$$H(\omega) = \frac{A \sin[(\omega_0 - \omega) \frac{M\pi}{\omega_0}]}{(\omega_0 - \omega)} + \frac{A \sin[(\omega_0 + \omega) \frac{M\pi}{\omega_0}]}{(\omega_0 + \omega)}.$$

Wiederum mit $T = MT_0$ und mit $\omega_0 = 2\pi f_0 = 2\pi/T_0$ gilt

$$H(f) = \frac{AMT_0}{2} \left\{ \frac{\sin[\pi MT_0(f_0 - f)]}{\pi MT_0(f_0 - f)} + \frac{\sin[\pi MT_0(f_0 + f)]}{\pi MT_0(f_0 + f)} \right\}.$$

Das Spektrum besteht also aus einer Summe von zwei *sinc*-Funktionen, die je an den Werten $\pm f_0$ zentriert sind.

Untersuchen wir nun den Betrag des Spektrums an der Stelle f_0 :

$$|H(f_0)| = \lim_{f \rightarrow f_0} \frac{AMT_0}{2} \left\{ \frac{\sin[\pi MT_0(f_0 - f)]}{\pi MT_0(f_0 - f)} + \frac{\sin[2\pi M]}{2\pi M} \right\}.$$

Der erste Summand ist, wie wir im vorigen Beispiel gesehen haben, gleich 1 und der zweite Summand verschwindet für alle ganzzahligen M . Der Betrag des Spektrums einer einzelnen harmonischen Schwingung der Amplitude A ist also proportional zur Länge T des betrachteten Zeitintervalls:

$$|H_{f_0}| = \frac{AMT_0}{2} = \frac{A}{2}T.$$

Statt der intuitiv erwarteten diskreten Spektralwerte bei den Frequenzen $\pm f_0$ (bzw. $\pm \omega_0$) erhalten wir ein Spektrum, welches der Faltung dieser diskreten Spektralwerte mit der *sinc*-Funktion entspricht. Diese Faltung im Spektralbereich folgt auch aus der Tatsache, dass wir die zeitliche Begrenzung des Signals durch eine Multiplikation der Cosinus-Funktion mit einer Rechteck-Funktion im Zeitbereich erhalten haben. Die Breite des Hauptmaximums, die sich aus dem ersten Nulldurchgang ergibt, ist eine Funktion der Länge des betrachteten Zeitfensters. Wenn $M \gg 1$ dann kann man in erster Näherung den Beitrag von $H(-f)$ auf das Verhalten von $H(+f)$ in

der Nähe von $+f_0$ vernachlässigen. Die Nullstellen von $H(f)$ ergeben sich dann aus,

$$\pi T(f_0 - f_n) = n\pi, \quad |n| = 1, 2, 3, \dots$$

oder

$$f_0 - f_n = n/T$$

und für die Halbbreite des Hauptmaximums

$$|f_0 - f_1| = 1/T.$$

Die allgemein gültige Folgerung daraus ist, je länger das Zeitfenster, also je mehr Daten zur Verfügung stehen, desto genauer kann der Spektralgehalt eines Signals aufgelöst werden.

Das Abtasttheorem

Nun sind wir in der Lage eine formale Antwort auf die grundsätzliche Frage zu geben, *was ist die minimale Abtastrate, um eine gegebene kontinuierliche Funktion vollständig und unverfälscht zu diskretisieren?*

Wir definieren zuerst die verallgemeinerte Funktion $\delta(t)$ (Deltafunktion, Dirac-Stoss):

$$\delta(t) = \begin{cases} 0, & \text{für } t \neq 0 \\ \infty, & \text{für } t = 0 \end{cases}$$

so dass

$$\int_{-\infty}^{+\infty} \delta(t) dt = 1.$$

Diese "Funktion" besitzt folgende Eigenschaft:

$$\int_{-\infty}^{+\infty} x(\tau) \delta(t - \tau) d\tau = x(t).$$

Das heisst, dass die Faltung einer Funktion $x(\tau)$ mit dem δ -Impuls eine Selektion des Wertes von x an der Stelle $\tau = t$ bewirkt.

Formal führen wir nun die Abtastung durch, indem wir $x(t)$ mit einer Folge von Deltafunktionen $\delta(t)$ multiplizieren:

$$x(t) \Rightarrow x_k(t) = \sum_{k=-\infty}^{+\infty} x(t) \delta(t - k\Delta t),$$

wobei Δt hier die Abtastperiode darstellt.

Wie müssen wir nun beim Diskretisierungsprozess vorgehen, um sicher zu sein, dass wesentliche Merkmale dieser Funktion nicht verloren gehen?

In den Jahren 1948–1949 veröffentlichte C.E. Shannon Arbeiten, mit denen er die Informationstheorie begründete. In diesen Arbeiten formulierte er das sogenannte **Abtasttheorem** (in der hier verwendete Notation) wie folgt:

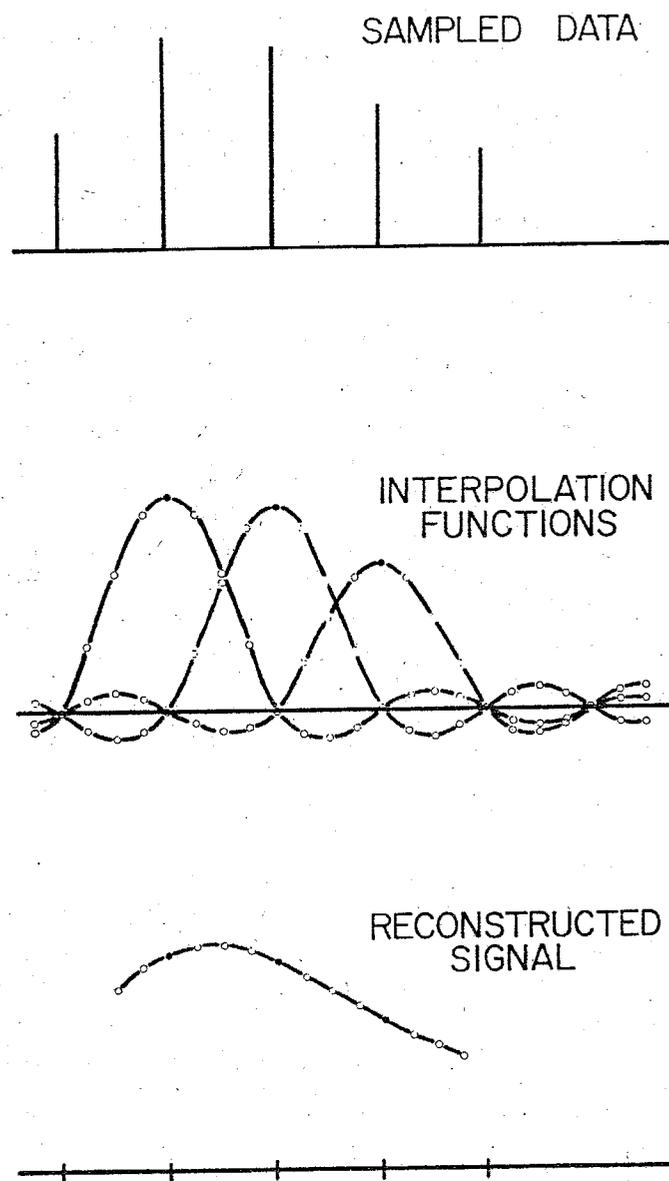
.... Let $x(t)$ contain no frequencies over f_{Nyq} , then:

$$x(t) = \sum_{k=-\infty}^{+\infty} x_k \frac{\sin \pi(2f_{Nyq}t - k)}{\pi(2f_{Nyq}t - k)},$$

where

$$x_k = x\left(\frac{k}{2f_{Nyq}}\right) = x(k\Delta t).$$

Obwohl einige Mathematiker dieses Theorem schon vor Shannon exakt formulierten und auf Probleme der Nachrichtentechnik anwendeten, wird es heute gewöhnlich als "Shannon's Abtasttheorem" bezeichnet. Dieses Theorem sagt, dass man eine beliebige Funktion $x(t)$, die sich aus einer diskreten



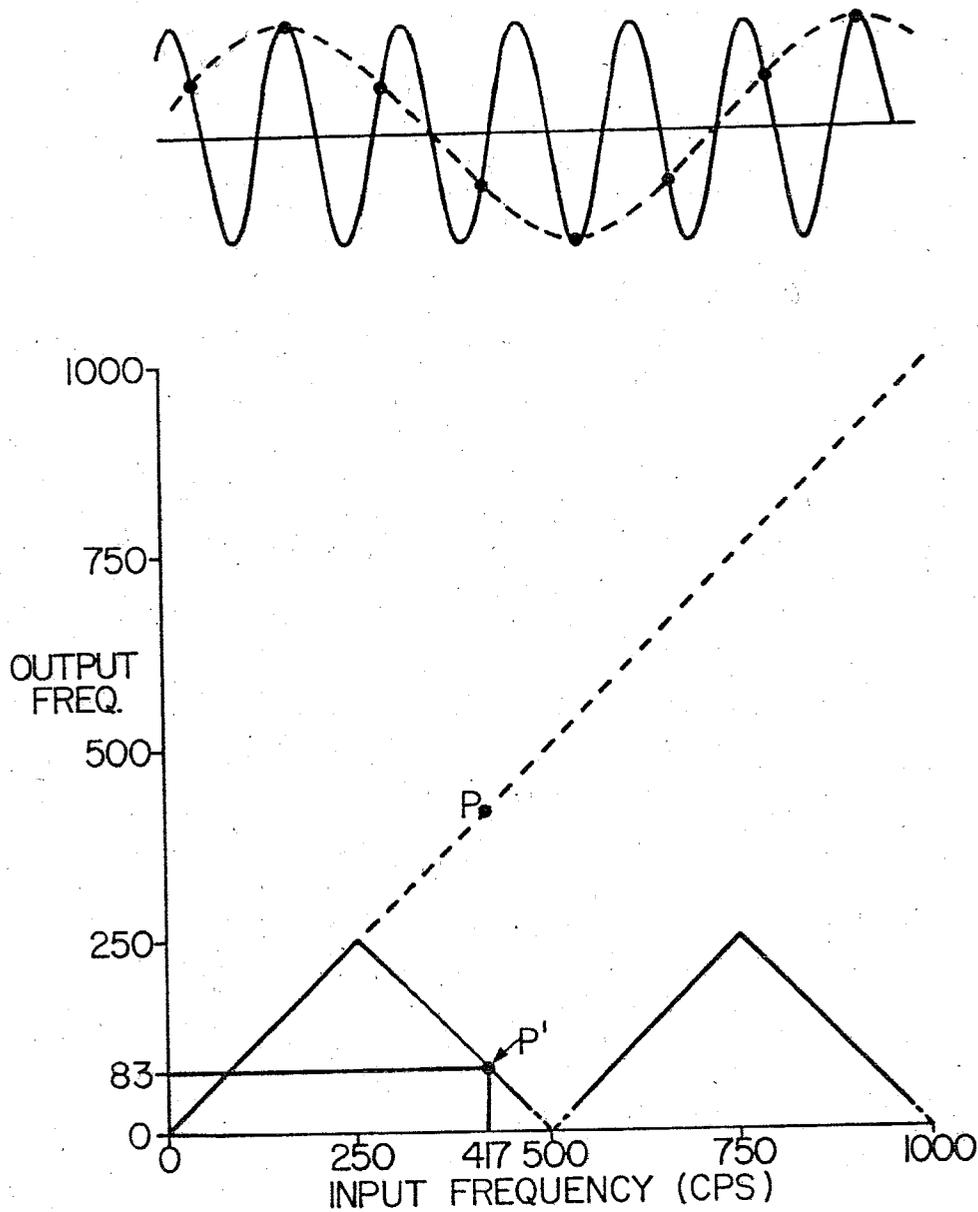
Figur 3.5: Graphische Darstellung des auf Shannons Abtasttheorem basierenden Interpolationsformalismus. (Aus: A Pictorial Digital Atlas, Bendix United Geophysical Corporation, 1966)

Anzahl Frequenzen von 0 bis f_{Nyq} zusammensetzt, durch die obige Reihe darstellen kann. Umgekehrt wiederum besteht eine beliebige Funktion $f(t)$, die durch die obige Reihe dargestellt wird, nur aus Frequenzen von 0 bis f_{Nyq} . In der Formulierung von Shannon liefert das Abtasttheorem auch einen Formalismus, der es erlaubt, eine diskrete Funktion beliebig fein zu interpolieren (siehe Abbildung 3.5).

Wir können das Abtasttheorem auch folgendermassen formulieren: Die Abtastung mit der Periode Δt ist nur dann sinnvoll, wenn die zu diskretisierende Funktion keine Energie bei den Frequenzen $f > 1/(2\Delta t)$ enthält. Die halbe Abtastfrequenz wird Nyquist-Frequenz genannt:

$$f_{Nyquist} = f_{Nyq} = \frac{f_s}{2} = \frac{1}{2\Delta t}.$$

Auf Englisch wird letztere auch *folding frequency* genannt, weil Frequenzen grösser als f_{Nyq} auf kleinere Frequenzen zurückgefaltet werden: $f = f_{Nyq} + y$ wird zu $f_a = f_{Nyq} - y$ (dies hat aber nichts mit dem oben definierten Faltungintegral zu tun!) (siehe Abbildung 3.6). Dieser Effekt wird mit dem englischen Wort *Aliasing* bezeichnet und f_a ist die *Alias Frequenz* von f (*Alias* heisst Doppelname). Mit anderen Worten, zu hohe Frequenzen werden bei ungenügender Abtastrate durch die Analog/Digital Wandlung als falsche tieferfrequente Signalanteile wiedergegeben. Daten, welche durch Aliasing verfälscht wurden, sind unbrauchbar! Um Aliasing zu vermeiden, muss entweder mit einer genügend hohen Frequenz abgetastet werden, oder mit einem Tiefpass-Filter (*Anti-Alias Filter*) vor der A/D Wandlung dafür gesorgt werden, dass keine Frequenzen höher als f_{Nyq} im Signal vorhanden sind. Aliasing ist ein allgemeines Phänomen diskreter Daten und kann sowohl bei Zeitserien, wie Seismogramme oder Temperaturmessungen, als auch bei räumlichen Daten, wie über ein Gebiet verteilte Schweremessungen oder Seismometer-Arrays, auftreten. Letzteres wird im Englischen als *spatial aliasing* bezeichnet.



Figur 3.6: Schematische Darstellung des Aliasing Phänomens: Gegeben sei eine harmonische Schwingung mit einer Frequenz von $f = 417$ Hz, welche mit 500 Hz abgetastet wurde ($f_{Nyquist} = 250$ Hz). Das Ergebnis ist, dass die $f = 250 + 167 = 417$ Hz auf $f_a = 250 - 167 = 83$ Hz zurückgefaltet werden. (Aus: A Pictorial Digital Atlas, Bendix United Geophysical Corporation, 1966)

Kapitel 4

Das Leistungsspektrum

Energie und Leistungsspektrum

Eine wichtige Beziehung zwischen einer Funktion und ihrer Fourier Transformierten ist der Energieerhaltungssatz (*Theorem von Parseval*):

$$\int_{-\infty}^{\infty} |h(t)|^2 dt = \int_{-\infty}^{\infty} |H(f)|^2 df.$$

Betrachtet man $|h(t)|^2$ als ein Mass der momentanen Leistung, dann ergibt das Integral über die Zeit die Energie. Folglich wird die Grösse $|H(f)|^2 = \Re(f)^2 + \Im(f)^2$ als *spektrale Energiedichte* bezeichnet, dessen Integral über alle Frequenzen wiederum gleich der Energie ist. Wenn z.Bsp. die Funktion $h(t)$ eine Verschiebung mit Einheit m ist, dann hat $|H(f)|^2$ die Einheit $\text{m}^2/\text{Hz}^2 = \text{m}^2\text{s}/\text{Hz}$, nämlich Leistung mal Zeit pro Frequenz.

Wenn $H(f)$ die Fourier Transformierte von $h(t)$ ist, dann ist also das Quadrat des Amplitudenspektrums $|H(f)|^2$ die spektrale Energiedichte. Für ein zeitlich begrenztes Signal der Dauer T ergibt sich dann das Leistungsspektrum oder genauer ausgedrückt die spektrale Leistungsdichte $P(f)$ (*power spectral density*) aus

$$P(f) = \frac{1}{T} |H(f)|^2.$$

Dieses direkt aus der Fourier Transformation ermittelte Leistungsspektrum ist als *Periodogramm* bekannt.

Das diskrete Leistungsspektrum

In Analogie zum kontinuierlichen Fall bezeichnet man das Betragsquadrat der DFT $|X_k|^2$ als spektrale Energiedichte (dies folgt daraus, dass x_n^2 als die Leistung des Signals betrachtet wird, und wenn z.Bsp. $[x_n] = \text{m}$ ist, dann ist $[x_n^2] = \text{m}^2$ und $[|X_k|^2] = \text{m}^2/\text{Hz}^2 = \text{m}^2\text{s}/\text{Hz}$, was einer Energie pro Frequenz

entspricht):

$$|X_k|^2 = \left| \Delta t \sum_{n=0}^{N-1} x_n e^{-j \frac{2\pi kn}{N}} \right|^2.$$

Dann ergibt sich der Energieerhaltungssatz (*Theorem von Parseval*) im Diskreten Fall wie folgt:

$$\Delta t \sum_{n=0}^{N-1} |x_n|^2 = \Delta f \sum_{k=0}^{N-1} |X_k|^2 = \frac{1}{N\Delta t} \sum_{k=0}^{N-1} |X_k|^2.$$

Da die Leistung als Energie pro Zeiteinheit definiert ist, erhalten wir die spektrale Leistungsdichte, $P(f_k)$, (engl. power spectral density, PSD) aus der spektralen Energiedichte mittels Division durch die Zeit:

$$P(f_k) = \frac{1}{N\Delta t} |X_k|^2 = \frac{\Delta t}{N} \left| \sum_{n=0}^{N-1} x_n e^{-j \frac{2\pi kn}{N}} \right|^2.$$

So normiert hat das Leistungsspektrum $P(f_k)$ die korrekten Einheiten von Leistung pro Frequenz (z.Bsp. m^2/Hz).

Eine andere oft verwendete Normierung des Leistungsspektrums lautet:

$$P'(f_k) = \frac{1}{N^2} \left| \sum_{n=0}^{N-1} x_n e^{-j \frac{2\pi kn}{N}} \right|^2$$

oder anders geschrieben:

$$P'(f_k) = \frac{1}{N\Delta t} \frac{\Delta t}{N} \left| \sum_{n=0}^{N-1} x_n e^{-j \frac{2\pi kn}{N}} \right|^2 = \frac{1}{N\Delta t} P(f_k).$$

Daraus sieht man, dass $P'(f_k)$ einer spektralen Leistungsdichte pro Zeiteinheit entspricht, mit einer Dimension der Leistung. Bei einem komplexen Signal hat diese Normierung den Vorteil, dass die so gewonnenen Koeffizienten genau der Leistung der jeweiligen harmonischen Schwingung entsprechen. Im Gegensatz zum Fall der Amplituden- und Leistungsdichte, ist ausserdem der Betrag von $P'(f_k)$ unabhängig von der Dauer des analysierten Zeitfensters, was in der Praxis von Vorteil ist, wenn Leistungsspektren von verschiedenen langen Signalen miteinander verglichen werden sollen.

Wenn man beide Seiten der Parseval'schen Beziehung durch die Signallänge $N\Delta t$ dividiert, erhält man eine Beziehung für die mittlere Leistung,

$$\frac{1}{N} \sum_{n=0}^{N-1} |x_n|^2 = \frac{1}{(N\Delta t)^2} \sum_{k=0}^{N-1} |X_k|^2 = \frac{1}{N\Delta t} \sum_{k=0}^{N-1} P(f_k) = \sum_{k=0}^{N-1} P'(f_k).$$

Für eine Zeitreihe deren Mittelwert 0 ist, stellt der linke Term dieser Gleichung die Varianz dar. In diesem Fall ist somit die Summe des Leistungsspektrums über alle Frequenzen gleich der Varianz von x_n :

$$\sum_{k=0}^{N-1} P'(f_k) = \text{Var}(x_n).$$

Das Leistungsspektrum in MATLAB

Das gebräuchlichste Verfahren das Leistungsspektrum einer Zeitreihe zu berechnen, beruht auf der diskreten Fourier Transformation und wird als *Periodogramm* bezeichnet. Da in der Literatur und Praxis sehr oft die verschiedenen Normierungsarten des Leistungsspektrums unterschiedlich definiert sind, ist es wichtig, sich bei der Anwendung von gegebenen Computer-Routinen zu vergewissern, dass das Resultat den Erwartungen entsprechend normiert ist.

In MATLAB kann das Leistungsspektrum am einfachsten mit der Funktion `psd` berechnet werden. Beispiel für eine Zeitreihe im Vektor `x` mit Abtastrate `fs` und Anzahl Werte `N`:

```
w = boxcar(N); % Rechteck-Fenster
[ps,ff] = psd(x,N,fs,w,'none');
```

Der Betrag des Leistungsspektrums befindet sich im Vektor `ps` und die Frequenzwerte als Abszisse für den Plot im Vektor `ff`. Beide haben die Länge $N/2 + 1$. Der Parameter 'none' bedeutet, dass der Mittelwert und Trend von der Zeitreihe nicht abgezogen wird.

Das Leistungsspektrum kann natürlich auch direkt aus der FFT berechnet werden:

```
X = abs(fft(x,N));
nf = N/2 + 1;
ps = X(1:nf) .* X(1:nf);
```

Es zeigt sich, dass das Verhältnis `ps / pd` gleich `N` ist. Da für `ps` gilt

$$ps = \left| \sum_{n=0}^{N-1} x_n e^{-j \frac{2\pi kn}{N}} \right|^2$$

ist offensichtlich das Resultat der Funktion `psd` wie folgt normiert:

$$pd = \frac{1}{N} \left| \sum_{n=0}^{N-1} x_n e^{-j \frac{2\pi kn}{N}} \right|^2$$

Um das Leistungsspektrum gemäss der Definition der *spektralen Leistungsdichte*, $P(f_k)$, zu erhalten, muss also das Resultat der Funktion `psd` mit dem Abtastintervall Δt multipliziert oder durch die Abtastrate f_s dividiert werden. Um einen Wert des Leistungsspektrums zu erhalten, der unabhängig von der Länge des analysierten Zeitfensters ist, $P'(f_k)$, muss hingegen das Resultat von `psd` mit $1/N$ multipliziert werden.

Unter Anwendung des Theorems von Parseval gilt somit für eine Zeitreihe, deren Mittelwert 0 ist, $\text{mean}(ps) = \text{var}(x)$.

Wird eine andere Fensterfunktion `w` als das Rechteckfenster verwendet, spricht man von einem *modifiziertem Periodogramm*, denn dies hat eine

Auswirkung sowohl auf die Auflösung des Leistungsspektrums als auch auf den Betrag. Um die Auswirkung auf den Betrag rückgängig zu machen und die Unabhängigkeit von der Dauer des Zeitfensters zu bewahren, muss das Resultat der Funktion psd, statt mit $1/N$, mit folgendem Normierungsfaktor multipliziert werden:

$$(\text{norm}(\mathbf{w}))^2 / (\text{sum}(\mathbf{w}))^2 = \sum_{n=1}^N |w_n|^2 / \left(\sum_{n=1}^N w_n \right)^2$$

Dass dieser Faktor auch für das Rechteckfenster gilt, lässt sich leicht zeigen:

Rechteckfenster: $w_n = 1, (n = 1, \dots, N)$

$$\sum_{n=1}^N |w_n|^2 / \left(\sum_{n=1}^N w_n \right)^2 = N / N^2 = 1/N.$$

Kapitel 5

Korrelation

Kreuzkovarianz und Korrelation

In der Praxis stellt sich oft das Problem, die Aehnlichkeit zwischen zwei Signalen oder die Verschiebung zwischen zwei Signalen, bei der die Aehnlichkeit maximal ist, zu bestimmen. Oder es stellt sich die Frage nach der möglichen Abhängigkeit zwischen zwei Zeitreihen. In diesem Zusammenhang sind die Begriffe der Kovarianz und Korrelation von grosser Wichtigkeit.

Kovarianz

Die Kreuzkovarianzfunktion zweier kontinuierlicher reeller Funktionen $x(t)$ und $y(t)$ ist wie folgt definiert:

$$R_{xy}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T [x(t + \tau) - \overline{x(t)}][y(t) - \overline{y(t)}] dt.$$

Hier sind \overline{x} und \overline{y} die entsprechenden Mittelwerte

$$\overline{x(t)} = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T x(t) dt,$$

und analog für \overline{y} . Wir wollen im folgenden davon ausgehen, dass die Mittelwerte Null sind oder von unseren Signalen abgezogen worden sind.

Im Fall diskreter Signale lautet dann die Definition für die diskrete Kreuzkovarianzfunktion

$$R_{xy}(k) = \lim_{N \rightarrow \infty} \frac{1}{2N + 1} \sum_{n=-N}^{n=+N} x_{n+k} y_n.$$

Die Grösse τ bzw. der Index k werden als Verschiebung oder *lag* zwischen den zwei Signalen x und y bezeichnet. Auch von im Prinzip unendlich langen stochastischen Signalen stehen uns in der Praxis natürlich nur Signalabschnitte endlicher Länge zur Verfügung ($x_n, y_n; n = 0, \dots, N - 1$), wobei

angenommen wird, dass die Signale für $n < 0$ und $n > N - 1$ Null sind. Wir sprechen dann von einer Schätzung für die Kreuzkovarianz:

$$R_{xy}(-k) = \frac{1}{N} \sum_{n=0}^{N-1-k} x_n y_{n+k}, \quad (k = N-1, \dots, 1)$$

$$R_{xy}(k) = \frac{1}{N} \sum_{n=0}^{N-1-k} x_{n+k} y_n, \quad (k = 0, \dots, N-1)$$

Oft wird der Normierungsfaktor $1/N$ weggelassen. Die Rechenprozedur sieht dann wie folgt aus:

$$\begin{aligned} \hat{R}_{xy}(-N+1) &= x_0 y_{N-1} \\ &\vdots \\ \hat{R}_{xy}(-1) &= x_0 y_1 + x_1 y_2 + \dots + x_{N-2} y_{N-1} \\ \hat{R}_{xy}(0) &= x_0 y_0 + x_1 y_1 + \dots + x_{N-1} y_{N-1} \\ \hat{R}_{xy}(1) &= x_1 y_0 + x_2 y_1 + \dots + x_{N-1} y_{N-2} \\ \hat{R}_{xy}(2) &= x_2 y_0 + x_3 y_1 + \dots + x_{N-1} y_{N-3} \\ &\vdots \\ \hat{R}_{xy}(N-1) &= x_{N-1} y_0 \end{aligned}$$

Stellt man sich die Zeitreihen x und y untereinander geschrieben vor, dann wird das Signal y jeweils um k Stützwerte nach links ($k < 0$) oder nach rechts ($k > 0$) verschoben und die entsprechenden Werte von x und y werden miteinander multipliziert und aufaddiert.

In der folgenden Abbildung wird das Resultat gezeigt für die zwei Folgen

$$x_n = 0, 0, 0, 1, 4, 1, 0, -2, 0, 0$$

und

$$y_n = 0, 0, 4, 2, 1, 0, 0, 0, 0, 0$$

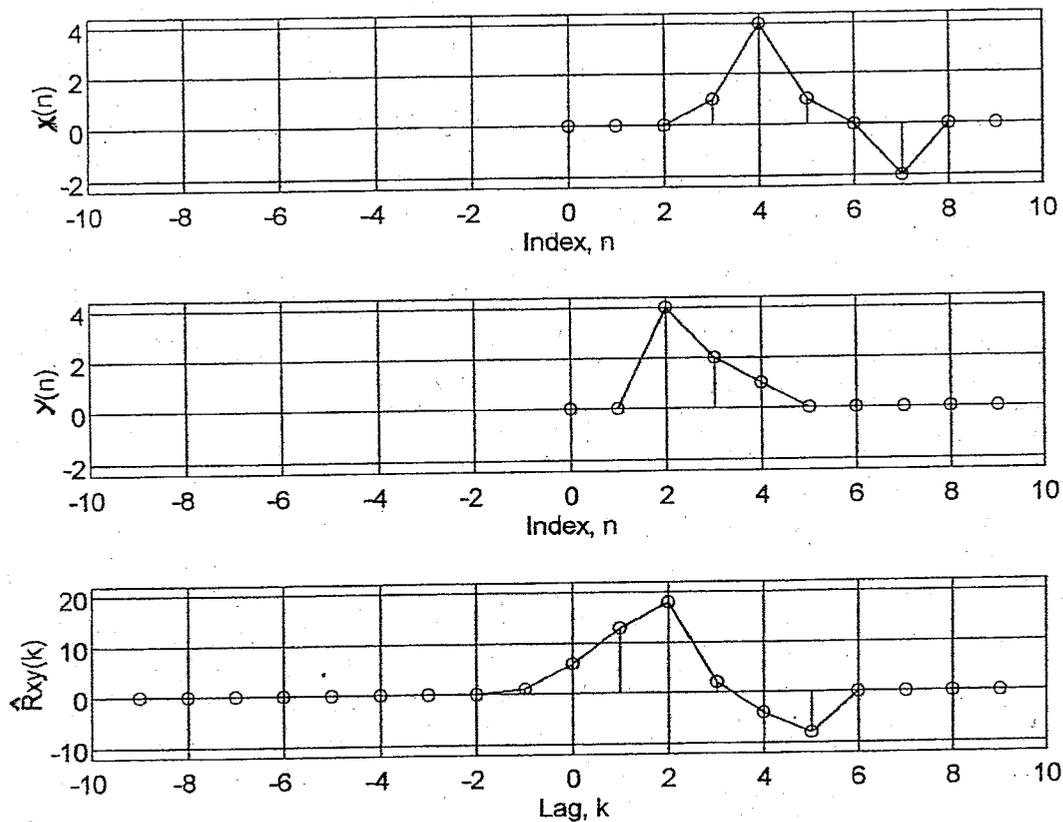
mit der Kovarianz

$$\hat{R}_{xy} = 0, 0, 0, 0, 0, 0, 0, 0, 1, 6, 13, 18, 2, -4, -8, 0, 0, 0, 0.$$

Das Maximum der Kreuzkovarianzfunktion wird bei einer Verschiebung *Lag* von $k = 2$ Stützwerten erreicht. Das entspricht dem Versatz zwischen der Maxima von x und y und gibt somit an, wo die grösstmögliche Übereinstimmung zwischen den zwei Signalen herrscht. (Man beachte aber, dass in diesem Beispiel die Mittelwerte der zwei Zeitreihen, im Gegensatz zur oben gegebenen Definition der Kovarianz, nicht Null sind.)

Das genau gleiche Resultat erhält man, wenn man die Reihenfolge der zwei Signale vertauscht und dann das untere Signal relativ zum oberen nach links verschiebt. Daraus ergibt sich das allgemeine Resultat

$$R_{xy}(k) = R_{yx}(-k).$$



Figur 5.1: Diskrete Kovarianzfunktion $\hat{R}_{xy}(k)$ der Signale $x(n)$ und $y(n)$.

Korrelation

Die Kreuzkovarianzfunktion oder Kovarianzfunktion wird auch oft als Kreuzkorrelation oder einfach Korrelation bezeichnet. Streng genommen sollten die letzten zwei Bezeichnungen der durch die Varianzen normierten Kreuzkovarianzfunktion vorbehalten sein. Für eine kontinuierliche Funktion ist die Varianz definiert als

$$\text{Var}\{x(t)\} = \sigma_x^2 = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T [x(t) - \bar{x}]^2 dt.$$

Wiederum unter der Annahme, dass der Mittelwert Null ist, erhalten wir für die Varianz eines diskreten Signals

$$\text{Var}(x_n) = \sigma_x^2 = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^{n=+N} x_n^2,$$

oder für eine Schätzung der Varianz aus Signalausschnitten endlicher Länge

$$\text{Var}(x_n) = \sigma_x^2 = \frac{1}{N} \sum_{n=0}^{N-1} x_n^2.$$

Die normierte Kreuzkorrelation, auch Korrelationskoeffizient genannt, ist dann

$$C_{xy}(k) = \frac{R_{xy}(k)}{\sqrt{\text{Var}(x)\text{Var}(y)}}.$$

Der Maximalwert der Kreuzkorrelation ist ein Mass für die Aehnlichkeit zweier Signale. Für zwei Signale, die vollkommen unkorreliert sind und somit nichts Gemeinsames haben, ist $C_{xy} = 0$. Die Grösstmögliche Aehnlichkeit haben natürlich zwei identische Signale bei der Verschiebung $k = 0$. Wie sich aus den Definitionen der Kovarianz und der Varianz leicht verifizieren lässt, ist dann $c_{xy}(0) = 1$. Ist hingegen $y_n = -x_n$ dann ist $C_{xy}(0) = -1$.

Autokovarianz und Autokorrelation

Autokovarianz

Analog zur Kovarianz von zwei verschiedenen Signalen koennen wir auch die Kovarianz eines Signal mit sich selber bilden. Diese wird Autokovarianz genannt. Für eine diskrete Zeitreihe x_n mit Mittelwert gleich Null ist sie definiert als

$$R_{xx}(k) = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^{n=+N} x_{n+k} x_n.$$

Aus der für die Kreuzkovarianz gültigen Beziehung $R_{xy}(k) = R_{yx}(-k)$ folgt $R_{xx}(k) = R_{xx}(-k)$. Somit ist die Autokovarianz eine gerade Funktion und es genügt, sie für positive Werte von k zu berechnen. In der Praxis können wir die Autokovarianz eines unendlich langen Signals nicht berechnen, sondern nur aus Signalabschnitten endlicher Länge abschätzen:

$$R_{xx}(k) = \frac{1}{N} \sum_{n=0}^{N-1-k} x_{n+k} x_n, \quad (k = 0, \dots, M),$$

wobei $M \leq N - 1$.

Es zeigt sich jedoch, dass obige Schätzung mit einem systematischen Fehler behaftet ist (*biased estimate*). Eine Fehlerfreie Schätzung (*unbiased estimate*) erhält man wenn man den Faktor $1/N$ vor dem Summenzeichen mit $1/(N - |k|)$ ersetzt. Letztere Normierung hat aber den Nachteil, dass die Varianz der Schätzung für grosse Werte von M sehr gross wird. Da ausserdem für $N \rightarrow \infty$ sowohl der Fehler als auch die Varianz bei beiden Normierungsarten gegen Null gehen, wird in der Praxis meistens die erste Form verwendet.

Wie man sofort sieht, ist der Wert der Autokovarianz bei der Verschiebung $k = 0$ nichts anderes als die Varianz des Signals:

$$R_{xx}(0) = \frac{1}{N} \sum_{n=0}^{N-1} x_n^2 = \sigma_x^2 = \text{Var}(x).$$

Für den Fall eines Leistungssignals entspricht diese Definition der mittleren Leistung des Signals. Ausserdem gilt $R_{xx}(k) \leq R_{xx}(0)$.

Autokorrelation

Die Autokorrelation eines Signals ist einfach die mit der Varianz des Signals normierte Autokovarianz:

$$C_{xx}(k) = \frac{R_{xx}(k)}{R_{xx}(0)}.$$

Der Maximalwert der Autokorrelation ist somit $C_{xx}(0) = 1$.

Für determinierte Signale endlicher Länge (*transient signals* oder *wavelets*), deren Energie endlich ist, geht die mittlere Leistung mit $N \rightarrow \infty$ gegen Null. Für solche Signale wird die Autokovarianz ohne Division durch N definiert:

$$R_{xx}(k) = \sum_{n=0}^{N-1-k} x_{n+k} x_n.$$

Die Autokovarianz für $k = 0$ entspricht dann der Energie des Signals:

$$R_{xx}(0) = \sum_{n=0}^{N-1} x_n^2.$$

Die Definition der Autokorrelation bleibt aber für diesen Fall unverändert.

Aus den Eigenschaften der Autokovarianz lässt sich einiges über die interne Gesetzmässigkeit oder umgekehrt über den Grad der Regellosigkeit eines Signals aussagen. Die Autokovarianz einer harmonischen Schwingung ist wiederum eine harmonische Schwingung der gleichen Frequenz (aber ohne Phaseninformation, da ja die Autokovarianz eine gerade Funktion ist). Je regelloser ein stochastisches Signal ist, umso schmaler wird das Maximum der Autokovarianzfunktion. Ein vollkommen regelloses Signal (sogenanntes weisses Rauschen) ist von einem Zeitpunkt zu einem anderen maximal unkorreliert. Somit gilt im Idealfall für die Autokovarianz eines regellosen Prozesses $R_{xx}(k) = 0$ ($k \neq 0$) und $R_{xx}(0) = \sigma_x^2$.

Eine ausführliche Diskussion über die Aussagekraft der Autokovarianz sowie einige konkrete Beispiele sind in Kapitel 8 des Buches von Buttkus¹ zu finden.

¹B. Buttkus, Spektralanalyse und Filtertheorie in der angewandten Geophysik, Springer Verlag, Berlin (1991)

Autokovarianz und Leistungsspektrum

Eine direkte Anwendung der Autokovarianz besteht im Zusammenhang mit der Schätzung der spektralen Leistungsdichte von stochastischen Signalen.

Wenn $X(f)$ die Fourier Transformierte eines zeitlich begrenzten Signals $x(t)$ der Dauer T ist, dann ist bekanntlich das Leistungsspektrum oder genauer ausgedrückt die spektrale Leistungsdichte $P(f)$ (*power spectral density*)

$$P(f) = \frac{1}{T} |X(f)|^2.$$

Dieses direkt aus der Fourier Transformation ermittelte Leistungsspektrum ist als *Periodogramm* bekannt.

Ein alternatives Vorgehen geht von der Autokovarianz des Signals aus. Nach dem Theorem von Wiener und Khintchine bilden nämlich die Autokovarianzfunktion und die quadratische Spektraldichtefunktion ein Fourier-Transformationspaar:

$$R_{xx}(\tau) = \int_{-\infty}^{\infty} P(f) e^{j2\pi f\tau} df$$

und

$$P(f) = \int_{-\infty}^{\infty} R_{xx}(\tau) e^{-j2\pi f\tau} d\tau.$$

Ein Beweis dieses Satzes ist z.Bsp. im Buch von Buttkus (Kap. 8) zu finden.

Dieser Sachverhalt kann genutzt werden, um die spektrale Leistungsdichte zu schätzen. Die Methode ist als *Korrelogramm* Verfahren oder als Blackman-Tukey Schätzung bekannt. Das praktische Vorgehen sowie eine Diskussion über Auflösung und Varianz sowohl des Korrelogramm- als auch des Periodogramm-Verfahrens sind wiederum bei Buttkus zu finden (Kap 9).

Die Autokovarianz sowie die spektrale Leistungsdichte, zusammen mit dem Mittelwert und der Varianz, werden verwendet, um statistische Eigenschaften von mehr oder weniger regellosen Prozessen von im Prinzip unbegrenzter Dauer zu erfassen. Eine Eigenschaft regelloser Prozesse besteht darin, dass die Phasenlage nicht determiniert ist, sondern zufälligen Schwankungen unterliegt. Somit ist es auch sinnvoll, dass in diesen Größen keine Information über die Phase der analysierten Signale enthalten ist.

Kovarianz und Korrelation in MATLAB

Für die Berechnung von Kovarianz und Korrelation in MATLAB stehen die zwei Funktionen `xcorr` und `xcov` zur Verfügung:

$$\text{xcorr}(x, y) = \hat{R}_{xy}(k) = \begin{cases} \sum_{n=0}^{N-1-k} x_{n+k} y_n, & (k = 0, \dots, N-1) \\ \hat{R}_{yx}(-k), & (k = -(N-1), \dots, -1) \end{cases}$$

Dies wird auch als *rohe Korrelation* bezeichnet.

$$\text{xcorr}(x, y, 'biased') = R_{xy}(k) = \frac{1}{N} \hat{R}_{xy}(k)$$

$$\text{xcorr}(x, y, 'unbiased') = \frac{1}{N - |k|} \hat{R}_{xy}(k)$$

$$\text{xcorr}(x, y, 'coeff') = C_{xy}(k) = \frac{R_{xy}(k)}{\sqrt{\text{Var}(x)\text{Var}(y)}}$$

Das heißt, die ersten drei Befehle liefern die Kreuzkovarianz und der vierte (`xcorr(x, y, 'coeff')`) die Kreuzkorrelation, wobei letztere mit $1/N$ normiert und somit *biased* ist. Um eine Korrelation zu erhalten, die *unbiased* ist, muss man

$$\text{xcorr}(x, y, 'unbiased') / (\text{std}(x) * \text{std}(y))$$

eingeben.

Die Funktion `xcov` unterscheidet sich von `xcorr` dadurch, dass vor der Korrelation die Mittelwerte von x und y abgezogen werden. Bezüglich Normierung verhalten sich die beiden Funktionen gleich.

Die Autokovarianz erhält man mit `xcorr(x, x)` oder `xcorr(x)` bzw. `xcov(x, x)` oder `xcov(x)` und den oben angegebenen Normierungen.

Bei einer Länge N der Zeitreihen (Vektoren) x und y , ist die Länge des Resultates gleich $2N - 1$ und der N -te Wert entspricht $\text{lag} = 0$. Ausserdem kann der maximale lag mit einem zusätzlichen Parameter `maxlag` festgelegt werden, z. Bsp. `xcov(x, y, maxlag, 'unbiased')`.

Kapitel 6

Stochastische Zeitreihen

Definitionen

Im Folgenden sollen einige wichtige Begriffe im Zusammenhang mit der Analyse von stochastischen Prozessen erläutert werden. Weiterführende und ausführliche Darstellungen dieses sehr weitläufigen Gebietes der Zeitreihenanalyse sind zum Beispiel in den Lehrbüchern von Schlittgen und Streitberg sowie von Shumway und Stoffer zu finden ¹

Stochastische Zeitreihen (x_n) sind als Realisierungen von stochastischen Prozessen zu verstehen, d.h. von dynamischen Vorgängen mit Zufallscharakter. Ein stochastischer Prozess ist eine Folge (X_t) von Zufallsvariablen X_t , wobei der Index t Element einer höchstens abzählbaren Menge T ist. In der Analyse von stochastischen Zeitreihen geht es meistens darum, gewisse statistische Gesetzmässigkeiten zu finden, welche den zugrundeliegenden stochastischen Prozess beschreiben. Dabei ist es wichtig, zu unterscheiden zwischen den beobachteten oder berechenbaren Eigenschaften einer oder mehrerer Realisierungen des Prozesses und diesen zugrundeliegenden Eigenschaften, die wir nicht direkt beobachten können sondern aus den Beobachtungen schätzen möchten. Somit unterscheidet die Notation von Schlittgen und Streitberg durch Verwendung von Gross- und Kleinbuchstaben zwischen dem stochastischen Prozess einerseits und der gegebenen Realisierung als Zeitreihe andererseits. Wir werden wie Shumway und Stoffer im Folgenden aber in beiden Fällen Kleinbuchstaben verwenden, in der Meinung, dass der Sinn aus dem Zusammenhang klar wird.

In der Notation muss man jedoch unterscheiden zwischen den entsprechenden statistischen Grössen. So sind die in Kapitel 2 eingeführten Mittelwerte, Varianzen und Autokovarianzen empirisch bestimmte Grössen

¹R. Schlittgen & B. Streitberg: Zeitreihenanalyse. Oldenburg Verlag, München, 8. Aufl., 1999.

R. H. Shumway & D. S. Stoffer: Time Series Analysis and its Applications. Springer Verlag, New York, 2000.

einer gegebenen Zeitreihe x_n :

$$\bar{x} = \frac{1}{N} \sum_{n=0}^{N-1} x_n$$

$$\text{Var}(x_n) = \sigma_x^2 = \frac{1}{N} \sum_{n=0}^{N-1} (x_n - \bar{x})^2$$

$$R_{xx}(k) = \frac{1}{N} \sum_{n=0}^{N-1-k} (x_{n+k} - \bar{x})(x_n - \bar{x}), \text{ mit } k = 0, \dots, M \text{ und } M \leq N - 1.$$

Der Mittelwert des entsprechenden Zufallsprozesses ist hingegen gemäss der klassischen Statistik als Erwartungswert für einen bestimmten Index t über alle möglichen Realisierungen des Prozesses definiert:

$$\mu_t = \mu_{tx} = E[x_t] = \int_{-\infty}^{\infty} x p_t(x) dx,$$

wobei $p_t(x)$ die entsprechende Wahrscheinlichkeitsdichte ist. In gleicher Art ist auch die theoretische Autokovarianz als Erwartungswert über alle möglichen Realisierungen der Werte x_s und x_t definiert:

$$\gamma(s, t) = \gamma_x(s, t) = \text{Cov}[x_s, x_t] = E[(x_s - \mu_s)(x_t - \mu_t)].$$

Die theoretische Autokorrelation ist die durch die Varianzen normierte Autokovarianz:

$$\rho(s, t) = \frac{\gamma(s, t)}{\sqrt{\gamma(s, s)\gamma(t, t)}}.$$

Zufallsprozesse, für die es möglich ist, den Ensemble-Mittelwert μ und die Ensemble-Autokovarianzfunktion γ durch Bildung des zeitlichen Mittelwertes über eine einzige Zeitreihe bzw. durch Berechnung der zeitlichen Autokovarianz zu schätzen, nennt man **ergodisch**. Eine exakte Darlegung der Bedingungen für Ergodizität ist z.Bsp. im Kapitel 5 von Schlittgen und Streitberg zu finden. Es zeigt sich, dass die meisten in der Praxis vorkommenden Zufallsprozesse diesen Bedingungen zumindest annähernd genügen.

Eine weitere Bedingung, die für die Modellierung eines stochastischen Prozesses vorausgesetzt wird, ist dass er **stationär** ist. Konkret, bezeichnet man einen stochastischen Prozess als:

mittelwertstationär wenn $\mu_t = \mu$ konstant ist,

varianzstationär wenn $\sigma_t^2 = \sigma^2$ konstant ist und

kovarianzstationär wenn die $\gamma(s, t) = \gamma(\tau)$ ist, d.h. wenn die Kovarianz nur vom Abstand (oder *lag*) $\tau = s - t$ abhängt. In diesem Fall ist auch die Autokorrelation $\rho(\tau) = \gamma(\tau)/\gamma(0)$.

Ein stochastischer Prozess, der sowohl mittelwertstationär als auch kovarianzstationär ist, wird als **schwach stationär** bezeichnet. (Frage: ist ein schwach stationärer Prozess auch Varianzstationär?) In der Praxis wird meistens der Begriff stationär an Stelle von schwach stationär verwendet.

Beispiele

White-Noise Prozesse

Ein **White-Noise Prozess** ist ein reiner Zufallsprozess der aus einer Folge (w_t) von identisch verteilten und unabhängigen Zufallsvariablen w_t besteht. Oft wird für ein solches **weisses Rauschen** eine Normalverteilung mit Mittelwert 0 und gegebener Varianz σ_w^2 vorausgesetzt. Aus der Unabhängigkeit der einzelnen Werte w_t folgt für die Autokovarianz:

$$\gamma_w(s, t) = \begin{cases} \sigma_w^2, & s = t \\ 0, & s \neq t \end{cases}$$

Daraus ist ersichtlich, dass White-Noise Prozesse stationär sind. White-Noise Prozesse sind zwar an sich nicht besonders interessant, sie dienen aber, wie im Folgenden gezeigt wird, als Input für eine Vielzahl von Modellen von stochastischen Prozessen.

MA Prozesse

Ein MA oder Moving-Average Prozess erhält man aus einer Linearkombination von White-Noise Variablen:

$$x_t = a_0 w_t + a_1 w_{t-1} + a_2 w_{t-2} + \dots + a_q w_{t-q}.$$

Dies entspricht der Konvolution von weißem Rauschen w_t mit einem Moving-Average Operator (a_n) . Man spricht in diesem Fall von einem Moving-Average Prozess der Ordnung q oder kurz von einem $MA(q)$ Prozess.

AR Prozesse

Ein AR oder autoregressiver Prozess x_t wird durch eine Linearkombination von früheren Werten von x und jeweils einer White-Noise Variable erzeugt:

$$x_t = a_0 w_t - b_1 x_{t-1} - b_2 x_{t-2} - \dots - b_p x_{t-p}.$$

Man spricht in diesem Fall von einem autoregressiven Prozess der Ordnung p oder kurz von einem $AR(p)$ Prozess. (Beachte: In den erwähnten Lehrbüchern sind die Koeffizienten b_n mit anderem Vorzeichen definiert.)

ARMA Prozesse

Die Kombination eines MA und eines AR Prozesses führt zwangslos zu einem sogenannten $ARMA(p, q)$ Prozess:

$$x_t = a_0 w_t + a_1 w_{t-1} + \dots + a_q w_{t-q} - b_1 x_{t-1} - \dots - b_p x_{t-p}.$$

ARIMA Prozesse

In der Praxis trifft man oft Zeitreihen an, die man als stationären ARMA Prozess, x_t , modellieren kann, dem aber noch ein linearer Trend überlagert ist. Eine solche Zeitreihe y_t kann man wie folgt beschreiben:

$$y_t = \beta_0 + \beta_1 t + x_t$$

Da der Mittelwert einer solchen Zeitreihe nicht für alle t gleich ist, ist die Zeitreihe y_t nichtstationär. Mit dem Differenzen-Operator Δ erhalten wir:

$$\Delta y_t = y_t - y_{t-1} = \beta_1 + x_t - x_{t-1} = \beta_1 + \Delta x_t.$$

Die neue Zeitreihe Δy_t ist somit wieder stationär und kann als stationärer ARMA Prozess modelliert werden. Dieses Vorgehen kann natürlich beliebig oft wiederholt werden und erlaubt somit auch Trends höherer Ordnung zu entfernen. Die ursprüngliche Zeitreihe y_t kann daher als eine Realisierung eines integrierten ARMA Prozesses angesehen werden. Verallgemeinert spricht man in diesen Fällen von $ARIMA(p, d, q)$ Prozessen, wobei d die Ordnung der Integration, bzw. des Trends bezeichnet.

ARMA Filter in MATLAB

Die Realisierung x_t eines $ARMA(p, q)$ Prozesses als Linearkombination von White Noise und früheren Werten von x entspricht der allgemeinsten Form einer linearen Filteroperation. Die a_n sind die Koeffizienten des Moving-Average Filters und die b_n sind die Koeffizienten des autoregressiven Filters. Letzterer wird auch als rekursives Filter bezeichnet. Dabei wird die Zeitreihe w_t (Input) durch die Gesamtwirkung aller Koeffizienten in eine gefilterte Zeitreihe x_t (Output) verwandelt.

In MATLAB steht dafür die Funktion `filter` zur Verfügung:

$$x = \text{filter}(ma, ar, w).$$

Der Vektor `ma` entspricht den MA Koeffizienten a_n und der Vektor `ar` den AR Koeffizienten b_n . Dabei gilt es aber zu berücksichtigen, dass MATLAB keinen Index 0 kennt. Somit sind

$$ma(1) = a_0 \text{ und } ma(q+1) = a_q$$

$$ar(1) = b_0 \text{ und } ar(p+1) = b_p$$

Unserer Notation entsprechend muss für alle Filter $ar(1) = b_0 = 1$ sein. Ausserdem gilt für ein reines AR Filter `ma = [a_0]` und für ein reines MA Filter `ar = [1]`.

Stichworte zur Vorlesung Zeitreihenanalyse im WS
=====

Klassifizierung diskreter Signale und Systeme:

- Abtastrate, Abtastintervall
- Kausal vs. nichtkausal
- Deterministisch vs. stochastisch
- Was ist ein lineares zeitinvariantes System
- Diskreter Delta-Impuls und diskrete Sprungfunktion:
- Was ist die Beziehung zwischen den beiden?
- Warum ist es zweckmaessig die Amplitude des diskreten Einheitsimpulses gleich der Abtastrate zu setzen?

Konvolution (Faltung)

Fourier Transformation:

Was ist das zu Grunde liegende Prinzip?

Sie ist eine komplexe Groesse:

Amplitudenspektrum, Phasenspektrum, Leistungsspektrum;
Wichtige Eigenschaften:

- FT von geraden und ungeraden Funktionen, Linearitaet, Nullpunktverschiebung, Ableitung, Integral, Faltung.
- Diskrete FT: Aufloesung (= kleinste Frequenz >0); maximale Frequenz (= Nyquist Frequenz);
- Beziehung zwischen Aufloesung und Anzahl Punkte in der Zeitreihe (Folgen des "zero-padding");
- Konsequenzen der Periodizitaet der DFT;
- Skalierung der DFT mit der Abtastrate (Einheiten!).
- Der Betrag (Amplitudenspektrum) eines Ausschnittes einer Zeitreihe, ist proportional zur Laenge des Ausschnittes.

Abtasttheorem:

Beziehung zwischen Abtastrate und Nyquistfrequenz.
Aliasing und seine Folgen!!!

DFT eines Zeitfensters aus einem laengeren Signal: Wirkung der Fensterlaenge und der Fensterform auf die Guete und Aufloesung der DFT; Achtung auf verschiedene Normierung des Leistungsspektrums (zeitunabhaengig und zeitunabhaengig). Energieerhaltungssatz (Theorem von Parseval).

Beziehung (Unterschied) zwischen FT von einem Delta-Impuls und von weissem Rauschen (warum heisst es "weiss"?)

Auto- und Kreuzkorrelation: Verschiedene Normierungen (was bedeutet in diesem Zusammenhang "biased und unbiased"?)

Beziehung zwischen Autokovarianz und Leistungsspektrum (Theorem von Wiener-Khinchine)

Stochastische Zeitreihen: was bedeutet AR, MA, ARMA, ARIMA?

2002/02/04, n.d.

Zusaetzliche Literatur zur

Zeitreihenanalyse und Signalverarbeitung
=====

Matlab:

D. Hanselman and B. Littlefield:
Mastering MATLAB 5.
Prentice Hall, London, 1998.

Teil I und II:

B. Buttkus:

Spektralanalyse und Filtertheorie in der angewandten Geophysik.
Springer, 1991.

R. Schlittgen und B. Streitberg:

Zeitreihenanalyse.
Oldenburg Verlag, Muenchen, 1999.

R. H. Shumway and D. S. Stoffler:
Time Series Analysis and its Applications.
Springer Verlag, New York, 2000.

Teil II:

A. V. Oppenheim and R. W. Schaffer:
Digital Signal Processing.
Prentice-Hall, Englewood Cliffs, New Jersey, 1975.

E. A. Robinson, T. S. Durani, L. G. Peardon:
Geophysical Signal Processing.
Prentice Hall International, 1986.

F. Scherbaum:
Of Poles and Zeros, Fundamentals of Digital Seismology.
Kluwer Academic Publishers, 1996.

2002/02/04, n.d.

Einige nuetzliche UNIX Befehle fuer den Anfang:

Bemerkung: die im Folgenden zwischen "" gesetzten Worte stehen stellvertretend fuer einen konkreten Namen und sind bei der Befehlsangabe nicht zu verwenden!

man "Befehl-Namen" liefert zu jedem Befehl die Gebrauchsanweisung; zu jedem der im Folgenden angefuhrten Befehle gibt es eine Menge Optionen, die das Verhalten der Befehle im Einzelnen steuern.

```
mkdir "Directory-Namen" erstellt ein neues Directory (Ordner)
rmdir "Directory-Namen" loescht ein Directory (es muss leer sein)
cd "Directory" wechselt Directory
Bsp. cd uebung = wechselt von gegenwaertigen Directory ins
Directory uebung
cd /afs/ethz.ch/users/j/jemand/uebungen = wechselt von
irgend einem Directory ins angegebenen Directory.
cd = wechselt immer ins Home-Directory
(e.g. /afs/ethz.ch/users/d/deichman)
cd .. = wechselt ins naechst-uebergeordnete Directory.
```

pwd zeigt den Namen des gegenwaertigen Directories

```
cp "file1" "file2" kopiert file1 auf file2
```

```
cp "directory/file1" . kopiert file1 im angegebenen Directory auf
file1 im gegenwaertigen Directory
(letzteres wird mit einem . angegeben!)
```

```
mv "file1" "directory/file2" verschiebt file1 zu directory/file2
```

```
mv "file1" "file2" umbenennet file1 zu file2
```

```
rm "file" loescht file
```

ls erstellt Liste der Files im gegenwaertigen Directory

```
ls -l wie ls aber mit ausfuehrlichen Angaben zu jedem File
```

```
ls -a wie ls aber inklusive "versteckte" files deren Name
mit einem Punkt anfangen
```

```
find "file" sucht den Ort von file
```

```
wc "file" gibt Groesse eines files an: Anzahl Zeilen, Worte, Zeichen
```

```
grep "string" "file" sucht die zusammenhaengende Zeichenfolge string
in file; ethaelt string ein Leerschlag oder
Sonderzeichen muss es zwischen ' oder " gesetzt
werden
```

```
sort "optionen" "file" sortiert den Inhalt eines Files
```

```
cut "optionen" "file" schneidet gewuenschte Kolonnen aus einem Textfile
```

(Der Befehl man sort oder man cut liefert eine Beschreibung der entsprechenden Optionen, die das Verhalten von sort und cut steuern.)

```
cat "file" druckt Inhalt von file auf den Bildschirm
```

```
cat "file1" "file2" >"file3" schreibt file1 und file2 auf file3
```

```
cat "file1" >>"file2" haengt file1 ans Ende von file2
```

```
head "file" schreibt die ersten 10 Zeilen von file auf den Bildschirm
```

```
tail "file" schreibt die letzten 10 Zeilen von file auf den Bildschirm
```

```
more "file" wie cat aber eine Seite aufs mal (mit Return oder
Leertaste weiter oder mit q abbrechen)
```

Wie im Beispiel vom Befehl cat, kann mit dem Zeichen > in vielen Faellen der Output eines Befehles statt auf den Bildschirm in ein beliebiges File geschrieben werden (= redirect).

Mit dem Zeichen | (= pipe) koennen mehrere Befehle aneinander gereiht werden (e.g. grep xxx gaga | more damit werden alle Zeilen von file gaga, die die Zeichen xxx enthalten auf den Bildschirm geschrieben, aber immer nur so viel aufs mal wie auf dem Bildschirm Platz hat.

Joker-Zeichen:

? steht fuer ein beliebiges einzelnes Zeichen

* steht fuer eine beliebige Anzahl einzelner Zeichen

Bsp. rm *.ps loescht alle Files, die mit .ps enden,

rm ab*.ps loescht alle files, die mit ab anfangen, mit .ps enden und ein beliebiges Zeichen dazwischen haben.

Fruehere Befehle koennen mit der Taste 'pfeil nach oben' geholt und einzelne Zeichen mit der Taste 'Backspace' geloescht werden und nach allfaelligen Aenderungen mit Return als neuer Befehl eingegeben werden.

Esc Esc: zweimaliges hintereinander Druecken der Escape Taste nach Eingabe der ersten Zeichen eines im Directory vorhandenen Filenamens fuehrt dazu, dass der Filename, soweit er eindeutig ist, automatisch vervollstaendigt wird (erspart Tipparbeit).

CTRL c (gleichzeitiges druecken der CTRL Taste und des Buchstabens c) damit kann oft ein laufender Prozess unterbrochen werden.

Wenn die Workstation aus irgendwelchen Gruenden von einem laufenden Prozess vollkommen blockiert ist, dann kann man mit folgendem Vorgehen versuchen, sie wieder zum laufen zu bringen:

1. sich auf einer anderen Maschine einloggen,
2. mit dem Befehl telnet "Maschine" auf die blockierte Maschine einloggen (e.g. telnet slab7),
3. mit dem Befehl ps -e | more die PID Nummer des blockierenden Prozesses suchen,
4. mit kill "PID-Nummer" den Prozess unterbrechen,
5. mit exit u.s.w. aus der zweiten Maschine wieder aussteigen.

(MATLAB)

Befehlsindex

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

A

abs, Addition +, angle, ans, Arithmetische Operatoren (+, -, *, /, ^), axes, axis

B

box

C

caxis, cd, ceil, cla, clc, clf, clear, colorbar, cond, conj, contour, conv, corcoef, cos, cot, cov

D

deconv, delete, demo, det, diag, diff, dir, disp (Kapitel "Vektoren und Matrizen") oder disp (Kapitel "Datenanalyse und Statistik"), Division (linke Division \, rechte Division /)

E

edit, eig, eps, eval, exit, exp, Exponentialfunktion ^, eye, ezplot

F

fclose, feval, fft, figure, fill, fill3, fix, floor, fmin, fmins, fopen, for, format, fplot, function, fzero

G

gca,(gcf, Gleichheitszeichen =, global, gradient, grid, griddata, grösser >, grösser gleich >=, gtext

H

help, hist, hold

I

i, if, imag, inf, input, interp1, interp2, inv, ishold

K

Klammern (eckige Klammer [], runde Klammer ()), kleiner <, kleiner gleich <=, Komma ,

L

legend, length, light, line, linspace, log, loglog, Logische Operatoren (&, |, ~ ==, <=, >=, <, >, logisches und &, logisches oder |, logisches nicht ~), load, lookfor

M

max, mean, mesh, meshgrid, Multiplikation * (elementare *)

N

nargin, nargin, norm

O

ode23, ode45, ones, orient, orth

P

patch, pause, pi, pie, plot, plot3, ploty, polar, poly, polyder, polyfit, polyval, print, printopt, prod, Prozentzeichen %, Punkte (ein Punkt ., zwei Punkte ., drei Punkte ...), pwd

Q

quad, quad8, quit

R

rand, rank, real, realmax, refresh, residue, roots, rotate3d, round, run

S

save, semilogx, sign, sin, size, sort, Spezielle Tasten (ctrl-P, ctrl-N, ctrl-B, ctrl-F, ctrl-A, ctrl-E, ctrl-U, ctrl-C), Spezielle Zeichen ((), [], ., . . ., . . ., . . ., . . ., % , =), spline, sqrt, stairs, std, stem, Strichpunkt ;, subplot, Subtraktion -, sum, surf, surface, svd, switch

T

tan, text, title, trace, Transponierte

V

view, viewmtx

W

which, while, who, whos

X

xlabel

Z

zeros, xlabel, zoom

Einfuehrung in MATLAB

Diese Uebung soll in erster Linie dazu dienen, einen Einstieg in Matlab zu finden. Information zur genauen Funktionsweise aller MATLAB Befehle kann mit dem Befehl help gefolgt vom gewünschten Befehlsnamen geholt werden. Zusätzliche Information ist auch mit den Befehlen helpwin und helpdesk erhaltlich. Ausserdem eroeffnet einm der Besuch bei //www.mathworks.com die gesamte MATLAB-Welt mit zusätzlicher Dokumentation und Referenzlisten.

Eine ausgezeichnete Einfuehrung in MATLAB sind die Kursunterlagen MATLAB and der ETH von Thomas Schubiger, Institut fuer Mess- und Regeltechnik. Diese sind unter //www.imrt.mavt.ethz.ch/~matlab/ zu finden und koennen auch von dort heruntergeladen sowie ausgedruckt werden.

Ein sehr gutes (aber leider nicht ganz billiges) Lehrbuch und Nachschlagwerk fuer MATLAB ist Mastering MATLAB 5 von Duane Hanselman und Bruce Littlefield, Prentice Hall (1998), ISBN 0-13-858366-8.

Die folgende Uebung soll an Hand eines einfachen Beispiels die Funktionsweise von MATLAB erlautern.
 Alle Zeilen mit % sind Kommentare.
 Die Zeilen ohne % sind MATLAB Befehle und sind auszufuehren.

```
% Frage: Wie lang sind Nil, Mississippi und Rhein?
% Gegeben:
% Nil + Mississippi = 10450 Km
% Mississippi + Rhein = 5099 Km
% Nil + Rhein = 7991 Km
% Wir haben drei lineare Gleichungen fuer drei Unbekannte,
% somit ist dies ein klassisches Problem der linearen Algebra,
% welches in Matrix-Schreibweise Ax = y geschrieben werden kann.
% In unserem Fall sind x und y zwei Spalten-(Kolonnen-)Vektoren
% mit x(1) = Nil, x(2) = Mississippi, x(3) = Rhein
% und y(1) = 10450, y(2) = 5099, y(3) = 7991
% und A ist eine 3x3 Matrix.
% Schreiben Sie die Elemente A(1,1),...,A(3,3) dieser Matrix
% fuer unseren Fall auf.
```

% Loesung mit MATLAB:

```
% In MATLAB werden die Elemente von Vektoren und Matrizen zwischen
% eckige Klammern gesetzt. Elemente einer Zeile (row in Englisch)
% werden durch Komma oder Leerzeichen getrennt. Elemente zweier
% verschiedener Zeilen (d.h. auch Elemente eines Spalten-Vektors)
% (Englisch column vector) werden durch Strichpunkt getrennt.
%
% In unserem Fall sind:
% y = [y(1);y(2);y(3)] und
% A = [A(1,1) A(1,2) A(1,3);A(2,1) A(2,2) A(2,3);A(3,1) A(3,2) A(3,3)]
%
% Der erste Index bezeichnet immer die Zeile und der zweite die Spalte.
% Setzen Sie in den folgenden zwei Zeilen die entsprechenden Zahlen ein:
A = [ ]
y = [ ]
% Linke Division ergibt die gesuchte Loesung:
x = A \ y
% y wird von links durch A dividiert
% tatsaechlich: Ax = y
% Die linke Division entspricht einer Pre-Multiplikation beider Seiten
% der Gleichung Ax = y mit der Inversen von A:
x = inv(A)*y
inv(A)*A*x
% Frage: was bedeutet die rechte Division?
% sei B = A-transponiert und v = y-transponiert (in MATLAB wird
% die Transponierte einer Matrix oder eines Vektors mit ' erzeugt):
B = A'
v = y'
% transponierte 3x3 Matrix
% Zeilen-Vektor
% in Matrix-Schreibweise gilt dann: uB = v
% und u ist:
u = v/B
u*B
x'
% Rechte Division
% tatsaechlich, uB = v:
% somit ist u = x-transponiert
% Die rechte Division entspricht einer Post-Multiplikation beider
% Seiten der Gleichung uB = v mit der Inversen von B:
u = v*inv(B)
u*B*inv(B)
```

```

% Also wenn Ax = y dann gilt (Ax)' = y' und x'A' = y'
% und es folgt: x = A\y und x' = y'/A':
y'/A'
x'

% Beachte: In MATLAB sind die Operationen * / \ zwischen
% Matrizen und Vektoren Matrix-Operationen; die entsprechenden
% Operationen und Vektoren einzelnen Elementen von Vektoren oder
% Matrizen (Array-Operationen) sind in MATLAB .* ./ ./.\
% Also ergeben folgende Befehle ganz unterschiedliche Resultate:
A*inv(A)
A.*inv(A)
A./inv(A)
A.\inv(A)
A.\inv(A)
A.\inv(A)

```

% Bei der elementweisen Division messen Sie also darauf achten,
% dass Sie nicht durch 0 dividieren.

```

% Die Elemente einer Matrix sind einfach zu adressieren:
A(1,2) % ist das zweite Element der ersten Zeile
A(:,2) % ist die gesamte zweite Spalte
A(2,1:2) % sind die ersten zwei Elemente der zweiten Zeile
% Einige nützliche Befehle sind:
% zeros(n,m), ones(n,m), eye(n,m)
% wobei n und m beliebige ganze Zahlen sein koennen:
% Beispiele:
zeros(3,5)
ones(5,3)
eye(3,5)

% Wenn n=m ist (quadratische Matrix), dann genuegt eine Zahl:
eye(3) % zum Beispiel die 3x3 Identitaetsmatrix
% Wie sie sehen sind
eye(3) / A
% und
inv(A)
% das Gleiche.

```

```

% Wir wollen nun unser Gleichungssystem erweitern:
Y(4) = x(1) - x(2)
Y(5) = x(2) - x(3)
% Ergaenzen Sie nun die Matrix A entsprechend dieses
% erweiterten Gleichungssystems (fuegen sie in untenstehendem
% Befehl die fehlenden Zahlen ein, um die Matrix A zu ergaenzen):
A( , ) = [
]
% Dies ist ein ueberbestimmtes Gleichungssystem: da die
% letzten zwei Gleichungen Linearkombinationen der ersten
% drei sind, sind sie streng genommen ueberfluessig.
% Wenn Sie jetzt die Inverse von A berechnen
inv(A)
% sehen Sie, dass das nicht geht.
% Trotzdem wird Ihnen MATLAB eine Loesung des Gleichungssystems
% Ax = y berechnen:
A\y
% ist das Gleiche wie das frueher berechnete
x
% Tatsaechlich fuehrt MATLAB eine Ausgleichsrechnung nach der
% Methode der kleinsten Fehlerquadrate durch (aehnlich wie die
% Berechnung der Koeffizienten einer Geraden durch eine
% Punkteschar).
% In unserem Beispiel ist das Resultat identisch mit dem frueher
% berechneten Wert, weil die zusaetzlichen Werte Y(4) und Y(5)
% aus den schon bekannten Werten von x berechnet wurden.
% In der Praxis stellen die Werte von Y jedoch meistens irgendwelche
% Messwerte dar, die fehlerbehaftet sind. Das hat zur Folge,
% dass die ueberaehnlichen Gleichungen nicht als Linearkombination
% der anderen Gleichungen hervorzuheben. Somit gibt es auch
% keine exakte Loesung x des Gleichungssystems Ax = Y.
% sondern nur eine optimale Loesung, fuer welche zum Beispiel die
% Summe der Fehlerquadrate minimal ist.
% Sei A' wieder die Transponierte von A, dann fuehrt die Bedingung
% der kleinsten Fehlerquadrate auf das sogenannte
% Normalgleichungssystem (A'A)x = A'Y
% (s. Vorlesung ueber Inversionstheorie).
% Tatsaechlich ist auch in unserem Fall
(A'*A)*x
% gleich
A'*y

```

5

```

% und da A'A eine quadratische Matrix ist,
A'*A
% existiert
inv(A'*A)
% und somit auch
inv(A'*A)*(A'*y)
% was in MATLAB das Gleiche ist wie
A\y

```

6

In dieser Uebung wollen wir der Frage nachgehen, was war die Anzahl Erdbeben mit M>1 pro Jahr in der Schweiz seit 1975? Gleichzeitig wollen wir verschiedene MATLAB Hilfsmittel kennenlernen, mit denen Daten graphisch dargestellt werden koennen.

Als Daten stehen uns ein File von Erdbeben-Lokalisierungen mit Datum, Uhrzeit, Epizentrum, Herdtiefe, Magnitude, usw. fuer jedes Beben ab 1/01/1975 zur Verfuegung. Da in diesem ASCII-Files die Daten schon als reine Zahlen-Arrays vorhanden sind, koennen sie sehr einfach mit dem MATLAB Befehl load in Matrix-Form eingelesen werden.

Die Erdbeendaten stellen je eine Einzelbeobachtung dar. Als solche sind sie eine Folge von Ereignissen, die einer Zeitreihe mit unregelmassigen Zeitabstaenden entspricht. Diese Daten koennen auf vielfaeltiger Weise dargestellt werden -- als Folge von Magnituden in Funktion der Zeit, als kumulative Anzahl Ereignisse oder als Ereignisse pro Zeiteinheit.

Zur Berechnung der Anzahl Ereignisse/Jahr stehen Ihnen die Routinen decayear und cumml sowie njahr, njahr1 und njahr2 zur Verfuegung. Bevor Sie anfangen, sollten sie sich die entsprechenden M-Files (decayear.m, cumml.m, njahr.m, njahr1.m und njahr2.m) mit folgenden UNIX Befehlen in Ihr Arbeitsdirectory kopieren:

```
cp /afs/ethz.ch/users/d/deichman/mfiles/decayear.m
cp /afs/ethz.ch/users/d/deichman/mfiles/cumml.m
cp /afs/ethz.ch/users/d/deichman/mfiles/njahr*.m
```

Wichtig: vergessen Sie nicht das Leerzeichen und den Punkt am Schluss, denn damit wird das gegenwaertige Directory gekennzeichnet. Das Sternzeichen ist eine Joker-Zeichen und steht hier fuer die Zeichen r, r1 und r2, so dass Sie nur ein Befehl benoetigen, um alle drei Files zu kopieren.

Mit der Taste Pfeil-nach-oben koennen sie sich jeweils den vorhergehenden Befehl wieder auf den Bildschirm holen; dann koennen sie den alten Befehl editieren -- einzelne Buchstaben lassen sich mit der Backspace Taste loeschen -- und mit der Return Taste den neuen Befehl ausfuehren.

Schauen Sie sich die entsprechenden M-Files genau an und versuchen Sie, nachzuvollziehen, wie sie funktionieren.

Format des Files eb.isx mit den Erdbeendaten:

Datum	Zeit (GMT)	Breite	Laenge	X	Y	Z	M	rms	gap	dm	no
1975 01 04 16 03 54.5	47.414	6.443	524800	251800	5	2.1	0.89	272	147	14	
1975 01 05 10 33 51.1	46.040	6.919	559700	98700	8	1.2	0.31	285	79	11	

rms = Wurzel des mittleren Quadrates der Laufzeitresiduen
 gap = Groesster Winkel zwischen Epizentrum und zwei benachbarten Stationen
 (gap > 180 ==> Epizentrum liegt ausserhalb des Stationsnetzes)
 dm = Entfernung zur naechsten Station (km)
 no = Anzahl Beobachtungen, die fuer die Lokalisierung verwendet wurden

```

% M-File decayear.m
function x=decayear(y)
% converts date and time from year, month, day, hour, minute, second
% into units of decimal years.
% input is an array y with date and time in the first 6 columns,
% output is vector x with number of elements equal to the
% number of rows in y.
ndays = [0 31 59 90 120 151 181 212 243 273 304 334];
n = length(y);
x = zeros(1,n); % initialize array x
for i=1:n
    year = y(i,1);
    month = y(i,2);
    day = y(i,3);
    hour = y(i,4);
    min = y(i,5);
    sec = y(i,6);
    leap = 0;
    if fix(year./4).*4 == year
        leap = 1;
    end
    julday = day + ndays(month);
    if month > 2
        julday = julday + leap;
    end
    x(i) = year + (julday-1+(hour+(min+sec./60)./60)./24)./(365+leap);
end

```

```

% M-File cumm1.m
% M-file to convert a vector of increasing times (zeit) into a
% vector of times that stay constant over two subsequent
% samples (zeit2) and a vector of corresponding cumulative
% numbers (ncumm2).
n = length(zeit);
n2 = n*2;
% initialize arrays for new time series (makes routine much faster):
zeit2 = zeros(1,n2);
ncumm2 = zeros(1,n2);
j = 0;
for i = 1:n;
    j = j + 2;
    zeit2(j-1) = zeit(i);
    ncumm2(j-1) = i-1;
    ncumm2(j) = i;
end

```

```

% M-file njahr1.m
function nprojahr = njahr1(eb)
% Function M-file njahr1.m liest Matrix eb, welche Erdbeben-Daten
% der Schweiz enthaelt und berechnet die Anzahl Erdbeben pro Jahr.
% Version mit zwei for-Schlaufen
%
m = length(eb); % Maximale Dimension (=Erdbeben) von Array eb
jahr1 = eb(1,1);
jahr0 = jahr1 - 1;
jahr2 = eb(m,1);
nj = jahr2-jahr0;
nprojahr = zeros(nj,2); % Aufsetzen von Array nprojahr
for jahr=jahr1:jahr2 % Schlaufe ueber alle Jahre
n = 0; % Schlaufe ueber alle Erdbeben
for i=1:m % Schlaufe ueber alle Erdbeben
if eb(i,1) == jahr
n = n+1;
end
end
nprojahr(jahr-jahr0,1) = jahr;
nprojahr(jahr-jahr0,2) = n;
disp(nprojahr(jahr-jahr0,:)) % zeigt das laufende ergebnis
end

% M-file njahr2.m
function nprojahr = njahr2(eb)
% Function M-file njahr2.m liest Matrix eb, welche Erdbeben-Daten
% der Schweiz enthaelt und berechnet die Anzahl Erdbeben pro Jahr.
% Index wird aus jahr-jahr0 berechnet
anzahl_erdbeben = length(eb); % Maximale Dimension von Array eb
jahr1 = eb(1,1);
jahr0 = jahr1 - 1;
jahr2 = eb(anzahl_erdbeben,1);
for jahr=jahr1:jahr2 % Matrix nprojahr initialisieren
k = jahr-jahr0;
nprojahr(k,1) = jahr;
nprojahr(k,2) = 0;
end
for i=1:anzahl_erdbeben % Schlaufe ueber alle Erdbeben
jahr = eb(i,1);
k = jahr-jahr0;
nprojahr(k,2) = nprojahr(k,2) + 1;
end
nprojahr % Zeige das Ergebnis

```

5

```

% M-File njahr.m
function nprojahr = njahr(zeit,option)
% Input: Vektor zeit mit Datum und Zeit als Dezimaljahr
% Output: Matrix mit Jahr und Anzahl Beben pro Jahr
% Option = 1: das Dezimaljahr wird nach unten und nach
% oben gerundet (round).
if option == 1
jahr1 = round(zeit(1));
k1 = jahr1 - 1;
anzahl_erdbeben = length(zeit);
jahr2 = round(zeit(anzahl_erdbeben));
for jahr=jahr1:jahr2 % Matrix nprojahr initialisieren
k = jahr-k1;
nprojahr(k,1) = jahr;
nprojahr(k,2) = 0;
end
for i=1:anzahl_erdbeben % Schlaufe ueber alle Erdbeben
jahr = round(zeit(i));
k = jahr-k1;
nprojahr(k,2) = nprojahr(k,2) + 1;
end
end
else
Option sonst: das Dezimaljahr wird nach unten gerundet (fix).
jahr1 = fix(zeit(1));
k1 = jahr1 - 1;
anzahl_erdbeben = length(zeit);
jahr2 = fix(zeit(anzahl_erdbeben));
for jahr=jahr1:jahr2 % Matrix nprojahr initialisieren
k = jahr-k1;
nprojahr(k,1) = jahr;
nprojahr(k,2) = 0;
end
for i=1:anzahl_erdbeben % Schlaufe ueber alle Erdbeben
jahr = fix(zeit(i));
k = jahr-k1;
nprojahr(k,2) = nprojahr(k,2) + 1;
end
end

```

6

```

% Frage: was war die Anzahl Erdbeben mit M>1 pro Jahr in der Schweiz?
% Moegliches Vorgehen:
% -----
clear % alle Variablen loeschen
% Daten einlesen
load /afs/ethz.ch/users/d/deichman/ebdata/eb.isx
% Konvertiere Datum und Zeit in Dezimaljahr
% (diese Routine ist langsam ==> Geduld!):
zeit = decyear(eb);
% Das M-file decyear ist ein sogenanntes M-function file,
% dessen interne Variablen nicht fuer den Aufrufenden
% verfügbar sind.
who % Liste der vorhandenen Variablen
% Groesse der Matrix und Laenge des Vektors:
size(eb)
size(zeit)
anzahl_erdbeben = length(zeit) % length = max(size)

% Zeichne die Magnitude jedes Ereignisses als Funktion der Zeit.
% Die Form des Befehls ist plot(x,y,'option')
magnitude = eb(:,12);
figure(1) % Aktiviere die erste Figur
plot(zeit,magnitude,'r+') % Zeichne die Daten als rote Kreuze
axis([1974,2001,0,6]) % Achsen nach Wunsch definieren
xlabel('Jahr')
ylabel('Magnitude')

```

```

% Wir wollen jetzt die kumulative Anzahl Beben als Funktion
% der Zeit zeichnen.
% Generiere dazu einen Vektor mit der kumulativen Anzahl Ereignisse:
ncum = 1:1:anzahl_erdbeben;
% Die Form des vorhergehenden Befehls ist
% Vektor = erster Wert:Inkrement:letzter Wert
% Als Alternative waere auch moeglich:
% ncum = linspace(1,length(zeit),length(zeit))
% Dieser Befehl hat die Form
% Vektor = linspace(erster wert, letzter wert, anzahl werte)
figure(2)
plot(zeit,ncum,'r') % zeichne die Daten als rote Kurve
xlabel('Jahr')
ylabel('Kumulative Anzahl Erdbeben')
axis([1974,2001,0,6000])
% Betrachte einen kleinen Ausschnitt aus der unteren linken
% Ecke dieser Figur:
axis([1975.0,1975.1,0,25])
% Die Kurve macht den Anschein, als ob es zwischen zwei Beben
% Bruchteile von Beben gibt, was zumindest unschoen ist.
% Will man nicht einzelne Punkte zeichnen, muss man, wie in der
% routine im M-file cumul, die anzahl punkte der Kurve verdoppeln
cumul % routine zur berechnung von zeit2 und ncum2
% Das M-file cumul.m ist eine reine Sequenz von MATLAB Befehlen
% und die internen Variablen dieser Routine sind nach dem Aufruf
% Teil der laufenden MATLAB Sitzung:
who % zeigt alle gegenwaertigen Variablen
hold on % Letzter Plot halten
plot(zeit2,ncum2,'b') % Blaue Kurve
hold off

```

```

% Wir wollen jetzt die Anzahl Beben pro Jahr berechnen.
% Dazu stehen uns die zwei Routinen njahr1.m und njahr2.m
% als M-Files zur Verfügung: Die Resultate der zwei Routinen
% sind identisch, die Ausführungszeit aber sehr verschieden.
% Schauen Sie sich die zwei M-Files an und überlegen Sie sich,
% warum die erste so viel langsamer ist als die zweite.
format short g % Setze Ausgabe Format (siehe help format)
nprojahr = njahr1(eb);
nprojahr = njahr2(eb);
% Das Resultat der Berechnungen steht im Array nprojahr.
% Zeichnen Sie die Histogramme der Anzahl Beben pro Jahr.
figure(3)
bar(nprojahr(:,1),nprojahr(:,2)),axis([1974,2001,0,400])
% Als Beispiel, wie das Resultat einer Berechnung mit MATLAB
% exportiert werden kann, schreiben wir den Inhalt
% von nprojahr in File njahr.out
save njahr.out nprojahr -ascii
% Dieses File kann in einem anderen Fenster
% z.Bsp. mit dem UNIX Befehl
% more njahr.out
% betrachtet werden.
% Die Routine njahr.m (ein Function M-File) basiert auf
% dem gleichen Algorithmus wie die Routine njahr2.m, hat
% aber zwei Optionen, um die Jahreszahl zu runden.
% Mit dem Befehl subplot koennen beliebig viele Bilder
% in einem einzigen Fenster gezeichnet werden (siehe
% help subplot fuer die genaue Syntax).
nprojahr = njahr(zeit,0) % Mit Option 0
subplot(2,1,1)
bar(nprojahr(:,1),nprojahr(:,2)),axis([1974,2001,0,400])
ylabel('Anzahl Erdbeben pro Jahr')
nprojahr = njahr(zeit,1) % Mit Option 1
subplot(2,1,2)
bar(nprojahr(:,1),nprojahr(:,2)),axis([1974,2001,0,400])
ylabel('Anzahl Erdbeben pro Jahr')
% Frage: Warum sehen die Histogramme so unterschiedlich aus
% und was fuer eine allgemeine Schlussfolgerung ziehen Sie daraus?

```

```

% Zeitreihenanalyse -- Übung WS03
%
%
% Es soll mittels Polynomapproximation der laengstfristige
% Trend in den mittleren Monatstemperaturen in Zuerich zwischen
% 1975 und 1998 abgeschaezt werden.
%
% Gegeben sei also eine Zeitreihe mit Temperaturwerten Y(i),
% die zu den Zeitpunkten x(i) gemessen wurden.
% Es sollen die Koeffizienten p(1),...,p(n) eines Polynoms der
% folgenden Form gefunden werden
%
%  $Y = p(1)*x^n + p(2)*x^{(n-1)} + \dots + p(n)*x + p(n+1)$ .
%
% Dabei sollen die Polynomwerte Y den Messwerten der Zeitreihe Y(i)
% zu den Zeitpunkten x(i) optimal im Sinne der kleinsten
% Fehlerquadrate entsprechen.
% Im Falle von n = 1 entspricht dies einer Geraden.
%
% Man beachte: Y ist offensichtlich nichtlinear bezueglich x,
% aber Y ist linear bezueglich der Koeffizienten p.
%
% Die Polynom Anpassung ist ein Beispiel der allgemeinen linearen
% Ausgleichsrechnung nach der Methode der kleinsten Fehlerquadrate:
% Dabei werden diejenigen Koeffizienten (Modellparameter) gesucht,
% fuer welche die Summe der Quadrate der Differenzen zwischen
% den berechneten und beobachteten Werten minimal ist.
% In unserem Fall sind die Polynomkoeffizienten die gesuchten
% Modellparameter.
% Eine Bedingung dafuer, dass die Fehlerquadrate minimal sind, ist,
% dass ihre partiellen Ableitungen nach den gesuchten Parametern
% gleich 0 sind.
% Diese Bedingung fuehrt auf die sogenannte Normalgleichungsmatrix,
% deren Inversion die gesuchten Parameter liefert. Da dies oft mit
% numerischen Problemen verbunden ist, wird in der Praxis mit Vorteil
% die Methode der Singular Value Decomposition (SVD) verwendet.
% Eine ausfuehrlichere Behandlung erfolgt in der Vorlesung ueber
% Inversionstheorie oder kann im Buch Numerical Recipes von
% Press et al. (Kap. 15 und 2) nachgelesen werden.
%
% In MATLAB werden die Polynomkoeffizienten mit der Funktion
% polyfit und die Polynomwerte mit der Funktion polyval berechnet
% (siehe help polyfit und help polyval).

```

```

% Moegliches Vorgehen:
% -----
clear % allfaellige existierende Variablen loeschen

load /afs/ethz.ch/users/d/deichman/Klima/temp.dat

% Die Daten bilden eine Matrix, bestehend aus 288 Zeilen, je
% mit einem Zeitwert (Dezimaljahr) und einem Temperaturwert.

x = temp(:,1); % Zeit-Werte -- zur Vereinfachung der Schreibweise
xi = x - mean(x); % zur Vermeidung grosser Zahlen
y1 = temp(:,2); % Temperatur -- zur Vereinfachung der Schreibweise

figure(1)
clf % allfaellig vorhandenes Bild loeschen
plot(x,y1) % zeichne die Zeitreihe
ylabel('Temperatur')

p1 = polyfit(xi,y1,1) % Regression einer Geraden:  $y = p(1)*x + p(2)$ 
y1 = polyval(p1,xi); % berechne die Werte der Geraden zu den xi

p2 = polyfit(xi,y1,2) % Regression fuer ein quadratisches Polynom
y2 = polyval(p2,xi); % berechne die Polynomwerte zu den xi

p3 = polyfit(xi,y1,3) % Regression fuer ein kubisches Polynom
y3 = polyval(p3,xi); % berechne die Polynomwerte zu den xi

p7 = polyfit(xi,y1,7) % Regression fuer ein Polynom 7. Grades
y7 = polyval(p7,xi); % berechne die Polynomwerte zu den xi

hold on % Abbildung nicht loeschen
plot(x,y1,'c')
plot(x,y2,'g')
plot(x,y3,'m')
plot(x,y7,'r')
hold off

% Zur Veranschaulichung der Resultate koennen
% die Polynome ohne Daten gezeichnet werden.
% Mit dem Befehl subplot kann man mehrere Bilder
% in einer Figur zeichnen (siehe help subplot).

figure(2)
clf % allfaellig vorhandenes Bild loeschen
subplot(2,1,1)
plot(x,y1,'c')
hold on
plot(x,y2,'g')
plot(x,y3,'m')
plot(x,y7,'r')
hold off
ylabel('Temperatur')
legend('y1','y2','y3','y7')

% Versuch das Polynom 7. Grades auf die Daten ohne
% Nullpunktverschiebung anzuwenden.
% Ergaenzen Sie im Folgenden selber die entsprechenden Befehle.

pp = polyfit( ) % Regression
yy = polyval( ); % berechne die entsprechenden Polynomwerte

subplot(2,1,2) % Zeichne Polynom ohne Null-Korrektur
hold on % Zeichne Polynom mit Null-Korrektur als rote Kurve
plot( ) % Zeichne Polynom mit Null-Korrektur als rote Kurve
hold off
title('Vergleich Polynom 7. Grades mit (y7) und ohne (yy)')
Null-Korrektur')
ylabel('Temperatur')
legend('yy','y7') % Die Legende kann mit der Maus verschoben werden

% Bemerkung: Je hoeher der Grad der verwendeten Polynome, umso
% wichtiger ist es, mit moeglichst kleinen Zahlen zu operieren,
% um numerische Probleme zu vermeiden.

```

```
% Zeitreihenanalyse --- Übung WS04
% =====
```

```
% Zur Illustration der Faltung (engl. convolution) sollen
% die mittleren Monatstemperaturen in Zuerich zwischen
% 1975 und 1998 so geglaettet werden, dass die saisonalen
% Schwankungen verschwinden.
% Die geglaetete Zeitreihe entspricht einem gleitenden
% Mittelwert. Die Berechnung des gleitenden
% Mittelwertes erfolgt durch Faltung der ungeglatteten
% Zeitreihe mit einer Zeitreihe, die den Koeffizienten des
% Gleitungsoperators entspricht.
```

```
% Moegliches Vorgehen:
% -----
```

```
clear
load temp.dat % Dezimaljahr und Monatstemperatur
load /afs/ethz.ch/users/d/deichman/Klima/temp.dat
size(temp)

temp1 = temp(:,2) ; % Bin Temperaturwert pro Monat (Zellenvektor)
length(temp1) % Anzahl Temperaturwerte
figure(1)
plot(temp1) % zeichne die Zeitreihe
xlabel('Monate ab Januar 1975')
ylabel('Temperatur')

f1 = ones(1,12)/12 ; % Koeffizienten des Gleitungsoperators
temp2 = conv(temp1,f1); % Faltung von temp1 mit f1
length(temp2) % Anzahl Werte nach der Faltung
% (Skript S. 15)

hold on
plot(temp2,'r')
hold off
```

```
% Deutlich sieht man hier die Anfangs- und Endeffekte, die auf der
% implizierten Annahme beruhen, dass die Zeitreihe temp1 ausserhalb
% beobachteten Zeitfensters Null ist.
% Etwas mildern laesst sich dieser Effekt, wenn man vorher dem
% Mittelwert abzieht:
```

```
temp1 = temp1 - mean(temp1) ;
temp2 = conv(temp1,f1) ;
figure(2)
plot(temp1)
hold on
plot(temp2,'r')
hold off
```

```
% Das hier angewendete Verfahren ist ein Beispiel eines
% allgemeinen Filter-Verfahrens, naemlich des sogenannten
% Moving Average Filters, oder kurz MA-Filter.
% Dabei wird der Vektor f1 als Filter bezeichnet und enthaelt
% die entsprechenden Filterkoeffizienten.
```

```
% Wie im Skript auf S. 16 dargestellt, kann man die Faltung auch als
% lineares Problem formuliert werden.
% Mit temp1 die ungeglattete Zeitreihe und temp3 die gesuchte
% geglaetete Zeitreihe gilt die Beziehung temp3 = H * temp1.
% Die Matrix H ist eine nicht-symmetrische Toeplitz Matrix deren
% Elemente aus den Koeffizienten von f1 bestehen.
% In MATLAB kann man die Matrix H mit dem Befehl toeplitz(u,v)
% generieren.
% Um die Form der Toeplitz Matrix und das Vorgehen zu veranschaulichen
% wollen wir erst ein einfaches Beispiel ausprobieren:
```

```
u = [1 2 3 0 0];
v = [1 0 0 0 0];
H = toeplitz(u,v)
```

```
% Sie sehen, u bildet die erste Spalte und v die erste Zeile der
% Matrix H; die restlichen Elemente werden so angeordnet, dass die
% Elemente der Diagonalen von links oben nach rechts unten jeweils
% gleich sind. Sie sehen auch, dass diese Matrix genau die gleiche
% Form hat wie diejenige auf S. 16 im Skript.
```

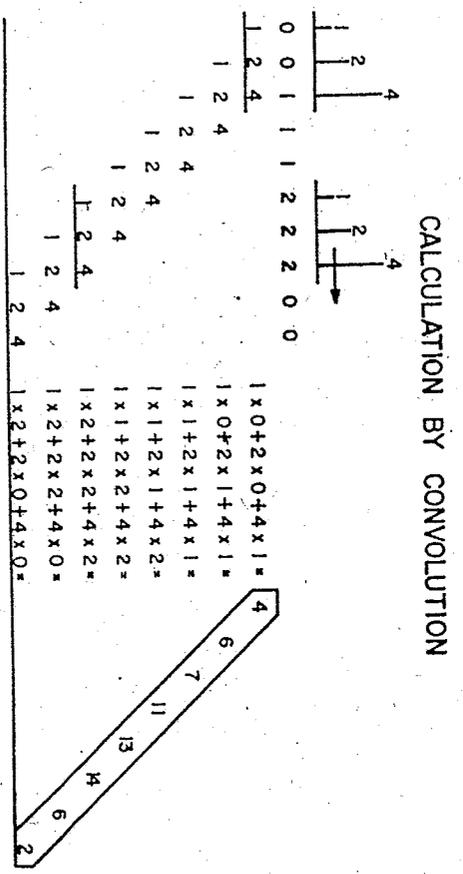
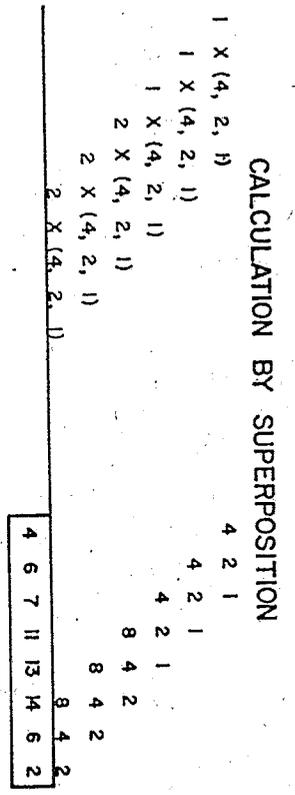
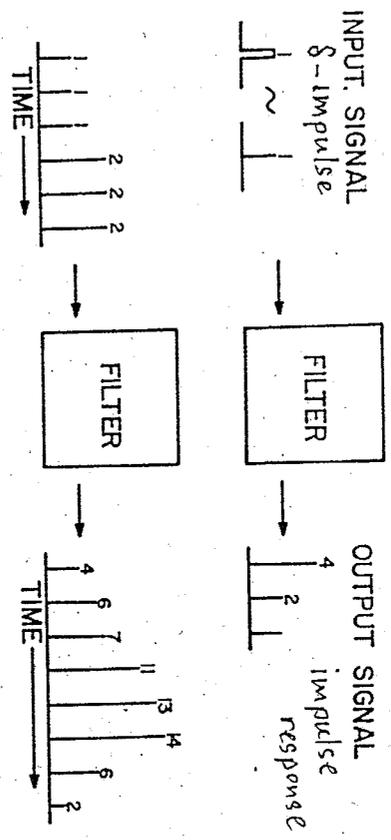
```
H = toeplitz(u,u) % generiert eine symmetrische Toeplitz Matrix.
% In unserem Fall ist H eine 288*288 Matrix und besteht aus den
% Koeffizienten von f1:
```

```
u = zeros(1,288);
v = zeros(1,288);
u(1:12) = f1;
v(1) = f1(1);
H = toeplitz(u,v);
```

```
% Die Faltung ist dann einfach die Multiplikation der Matrix H mit
% dem Vektor temp1 und liefert ein Resultat, welches praktisch
% identisch ist mit den ersten 288 Werten von temp2:
```

```
temp3 = H*temp1; % Achtung: temp1 muss transponiert werden!
hold on
plot(temp3,'g')
hold off
```

```
% Versuchen Sie obige Gleitungsoperation mit einer anderen
% Anzahl Filterkoeffizienten durchzufuehren (z. Bsp 24) und
% zeichnen Sie das Resultat ebenfalls in Figur 2 (mit einer
% anderen Farbe). Beachten Sie die Unterschiede.
```



FILTERING BY SUPERPOSITION AND BY CONVOLUTION

$$g(t) = f(t) * h(t) = \int_{-\infty}^{\infty} f(\tau) h(t-\tau) d\tau$$

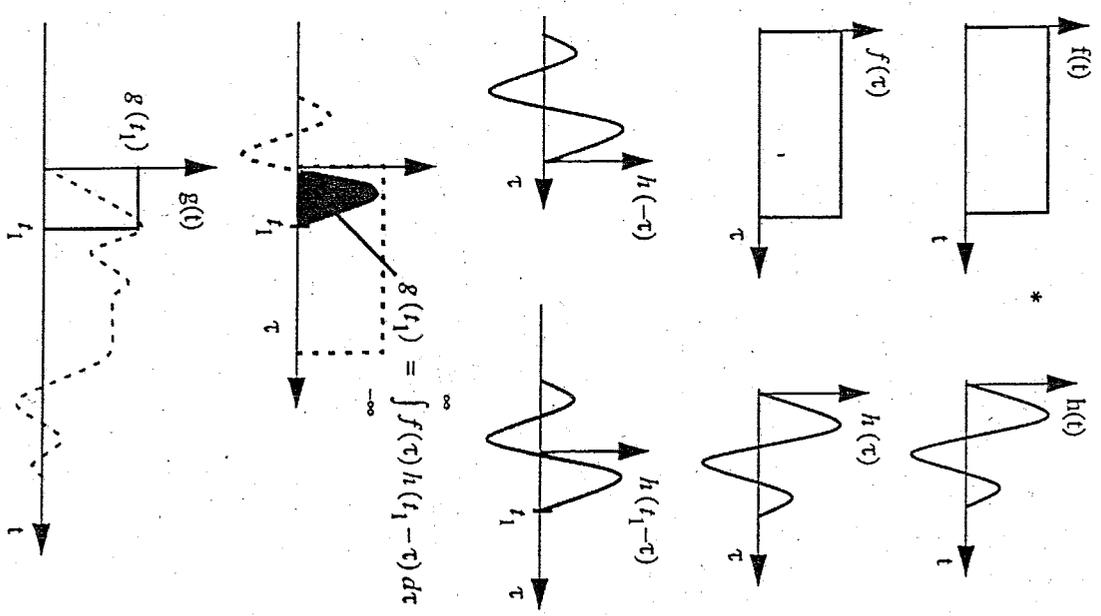


Fig. 2.4 Graphical interpretation of the convolution operation.

```

% Zeitreihenanalyse -- Übung WS05
% =====
% Synthese einer Zeitreihe aus Cosinus- und Sinusfunktionen
% -----
% Mit dieser Übung koennen Sie das Beispiel auf S. 19 im Skript
% Verifizieren. Beachten Sie dabei, dass der Amplitudenwert der
% groessten harmonischen Schwingung (3 Hz) nicht mit dem Maximum
% des Amplitudenspektrums uebereinstimmt. Wie sieht dieser
% Vergleich aus, wenn Sie doppelt so viele Werte nehmen (N=64)?
% Ersetzen Sie die Zeitreihe x mit einer beliebigen anderen
% Zeitreihe und verifizieren Sie die Fourier Transformation.
% Um sich viel Tipp-Arbeit zu ersparen, kopieren Sie sich,
% bevor Sie MATLAB starten oder aus einem anderen UNIX Fenster,
% die M-Files fourier1.m und fourier2.m in Ihr Arbeitsverzeichnis:
% cp /afs/ethz.ch/users/d/deichman/mfiles/fourier*.m
clear
N = 32 ; % Bestimme Anzahl Werte
Delta = 1/32 ; % Bestimme Abtastintervall
T = N*Delta; % Dauer des Zeitfensters
t = 0:Delta:(N-1)*Delta ; % Vektor mit Zeitschritten
% t = linspace(0, (N-1)*Delta, N) % (alternativ)
% Vektor mit Zeitreihe des Beispiels auf Seite 23 im Skript:
% -----
x = zeros(1, N) ;
x(13:16) = [64, 64, 64, 64] ;
x(17:20) = [-64, -64, -64, -64] ;
figure(1)
plot(t, x)
X = Delta * fft(x) ; % Fourier Transformation ==> Komplexer Vektor X
% Wenn Sie das ganze Amplitudenspektrum sehen wollen dann:
figure(2)
stem(abs(X)) % Betrag der Fourier Transformation
% Aber eigentlich genuegen die ersten N/2 + 1 Werte (s. Seiten
% 26 und 30 im Skript)
NF = N/2 + 1 ;
fnyq = 1/(2*Delta) ; % Vektor mit Frequenzschritten
f = linspace(0, fnyq, NF) ; % Darstellung des Amplitudenspektrums
stem(f, abs(X(1:Nf))) %
xlabel('Frequenz, Hz'), ylabel('Amplitude')

```

```

% Berechnung und Darstellung der einzelnen harmonischen Schwingungen
% und Aufsummierung der einzelnen harmonischen Schwingungen:
figure(3)
fourier1
% Hier nochmal das gleiche aber mit hoeherer Aufloesung, damit man
% sieht, dass es tatsaechlich harmonische Schwingungen sind:
figure(4)
fourier2

```

```

% Der Faktor Delta vor dem Aufruf der FFT dient zur korrekten
% Amplitudennormierung (s. Seite 30).
% Die M-Files fourier1 und fourier2 fuehren im Wesentlichen
% Schritt fuer Schritt die inverse Fourier Transformation aus.
% Dabei dient der Faktor 1/(N*Delta) wiederum der korrekten
% Amplitudennormierung.
% Beachten Sie, dass hier, im Gegensatz zur Definition der
% inversen Fourier Transformation im Skript, die Amplituden
% zusaetzlich mit 2 multipliziert werden.
% Frage: Warum die 2? -- Tipp: ueber wieviele Frequenzwerte
% summieren wir in der Routine fourier1?

```



```

% Zeitreihenanalyse -- Übung WS06
% =====
% Die vorliegende Übung koennen Sie sich als m-file in Ihr
% Arbeitsverzeichnis kopieren:
% cp /afs/ethz.ch/users/d/deichman/zr/ws06/ws06.m
% (Abstand und Punkt am Schluss nicht vergessen!)
% Nun messen Sie die unten fett gedruckten unvollstaendigen
% Befehle mit einem Text-Editor ergaenzen, und dann koennen
% Sie die Übung mit dem Befehl ws06 als m-file ausfuehren.

% Fourier Transformation einer Rechteckfunktion
% -----
% Vergleichen Sie das numerische Resultat mit dem
% analytischen Resultat im Skript (Seiten 31 - 32):
% Amplitude? Nulldurchgaenge? Grund fuer Unterschiede?

% Zeitreihe x mit Rechteck generieren:
clear
N = 256 ; % Anzahl Werte der Zeitreihe
Delta = 1/128 ; % Abtastintervall
x = zeros(1,N) ;
x(65:192) = ones(1,128) ; % Vektor mit Zeitachse
t = linspace(0,1,128) ; % Vektor mit Zeitachse
figure(1)
plot(t,x), axis([0,2,0,1.2]), xlabel('Zeit, (Sekunden)')

% Diskrete Fourier Transformation (Beachte Normierung mit Delta):
X = Delta * fft(x/N) ; % FFT - in diesem Fall ist fft(x,N)=fft(x)
F = fft(X) ; % Abtastrate
Fnyq = F(1:N/2+1) ; % Nyquist Frequenz
NF = length(Fnyq) ; % Anzahl Frequenzwerte
f = linspace(0,NF) ; % Vektor mit Frequenzachse
figure(2)
subplot(2,1,1)
plot(f,abs(X(1:NF))), xlabel('Frequenz, (Hz)')
axis([0,10,0,1])

```

```

% Die gleiche Fourier Transformation aber mit groesserer Aufloesung:
N = 2048 ;
X = fft(x,N) ; % FFT
NF = length(X) ; % Anzahl Frequenzwerte
f = linspace(0,NF) ; % Vektor mit Frequenzachse
subplot(2,1,2)
plot(f,abs(X(1:NF))), xlabel('Frequenz, (Hz)')
axis([0,10,0,1])

% Darstellung als Log-Log Plot:
figure(3)
loglog(f,abs(X(1:NF))), xlabel('Frequenz, (Hz)')
axis([0.01,100,0.001,10])

```

```

% Zeitreihenanalyse -- Ubung WS07
% =====

% Die vorliegende Ubung koennen Sie sich als m-file in Ihr
% Arbeitsverzeichnis kopieren:
% cp /afs/ethz.ch/users/d/delchman/zr/ws07/ws07.m
% (Abstand und Punkt am Schluss nicht vergessen!)
% Nun muessen Sie die unten fett gedruckten unvollstaendigen
% Befehle mit einem Text-Editor ergaenzen, und dann koennen
% Sie die Ubung mit dem Befehl ws07 als m-file ausfuehren.

% Amplituden und FT von harmonischen Schwingungen
% -----

% Generieren und zeichnen Sie die zwei Zeitreihen: Es sind
% zwei harmonische Signale mit gleicher Amplitude und Phase
% aber unterschiedlicher Frequenz.

clear
N = 256 ; % Anzahl Werte der Zeitreihe
Fabtast = 64 ; % Abtastrate
Delta = ; % Abtastintervall
t = linspace( ; % Vektor mit Zeitwerten (4 Sek.)

omegal = 2*pi*8 ; % Frequenz von Zeitreihe 1
omega2 = 2*pi*16 ; % Frequenz von Zeitreihe 2
phase = pi/4 ; % Phase
A = 3 ; % Amplitude
x1 = A * sin(omegal*t + phase) ; % Signal mit Frequenz omegal
x2 = A * sin(omega2*t + phase) ; % Signal mit Frequenz omega2
figure(1)
subplot(2,1,1)
plot(t,x1), axis([0,4,-4,4]), xlabel('Zeit', (s)')
subplot(2,1,2)
plot(t,x2,'r'), axis([0,4,-4,4]), xlabel('Zeit', (s)')

% FRAGE: Sehen die Signale so wie erwartet aus? Warum?

% Diskrete Fourier-Transformation (Beachte die Normierung mit Delta!)
X1 = ; % FFT
X2 = ; % FFT

% Amplitudenspektren:
Fnyq = ; % Nyquist-Frequenz
NF = ; % Anzahl Frequenzwerte bis Fnyq
F = linspace( ; % Frequenzwerte
figure(2)
subplot(2,1,1)
plot (F,abs(X1(1:NF))), xlabel ('Frequenz, (Hz)')
hold on
plot (F,abs(X2(1:NF)),'r'), xlabel ('Frequenz, (Hz)')
hold off

% FRAGE: Vergleichen Sie die erhaltenen Spektren mit der Theorie:
% Stimmen die Frequenzen und Amplituden?
% Wo sind die Nebenmaxima geblieben?

% Berechnen Sie die FT vom ersten signal aber mit hoeherer
% Aufloesung (mit "zero-padding" bis 1024 Werte)
N = 1024 ;
X1 = ; % FFT
NF = ; % Anzahl Frequenzwerte bis Fnyq
F = ; % Frequenzwerte
subplot(2,1,2)
plot (F,abs(X1(1:NF))), xlabel ('Frequenz, (Hz)')

% FRAGE: Wie verhalten sich jetzt die Nulldurchgaenge des Spektrums?
% Wo wird das Spektrum tatsaechlich abgetastet?

% Wiederholen Sie diese Ubung mit anderen Verhaeltnissen
% von Signalfrequenz, Abtastrate und Signallaenge:
% z.Bsp. gleiche Signalfrequenz, aber Fabtast = 59 Hz, N = 236.
% FRAGE: Wie erklaren Sie sich die Unterschiede?

```

```

% Zeitreihenanalyse -- Uebung WS08
=====

% Aufgabe:
% Wir generieren eine kunstliche Lufttemperatur-Zeitreihe, bestehend
% aus einer saisonalen Komponente und einer Tageskomponente.
% Zuerst "messen" wir unsere Temperaturen viermal am Tag, dann sagen
% wir uns, dass wir nur an der saisonalen Komponente interessiert sind,
% und "messen" nur noch zweimal pro Monat --> FRAGE: darf man das?

% Vorgehen:
% Kopieren Sie sich das m-file dieser Uebung in Ihr Arbeitsverzeichnis:
% cp /afs/ethz.ch/users/deichman/zr/ws08/ws08.m
% Ergaenzen Sie die fehlenden Angaben und fuehren Sie anschliessend die
% Uebung mit dem Befehl ws08 aus.

% Im Folgenden sei die Zeiteinheit, mit der wir rechnen, Jahre
% und somit die Frequenzeinheit 1/Jahr (=Perioden/Jahr).

% Definiere Amplitude und Frequenz der zwei Signalkomponenten:
-----
Afahr = 8 ; % Amplitude des Jahresgangs +/- 8 Grad
Atag = 5 ; % Amplitude des Tagesgangs +/- 5 Grad
omegat = 2 * pi * 1 ; % Kreisfrequenz des Jahresgangs
omegat2 = 2 * pi * 365 ; % Kreisfrequenz des Tagesgangs

% Wir generieren erst die viermal am Tag gemessene Zeitreihe:
% Setzen Sie selber die fehlenden Werte ein (Skript S. 26-30)
-----
T = 3 ; % Dauer der Messreihe = 3 Jahre
Fahbst1 = ; % Abtastrate (Messungen/Jahr)?
Fnyq1 = ; % Nyquist Frequenz?
Deltat1 = ; % Abtastintervall?
N1 = T * Fahbst1 + 1 ; % Totale Anzahl Messungen

% Zeit-Vektor (N1 Werte, zwischen 0 und T:
t1 = linspace(0,T,N1) ;

% Temperatur-Vektor (Summe von Jahres- und Tagesgang):
Temp1 = Afahr*sin(omegat*t1) + Atag*sin(omegat2*t1) ;

figure(1)
subplot(3,1,1)
plot (t1,Temp1)
xlabel('Zeit (Jahre)')
pause % Mit enter Taste weiter
axis([0,0.2,-15,15]) % Mit enter Taste weiter
pause % Mit enter Taste weiter
axis([0,max(t1),-15,15]) % Mit enter Taste weiter
pause

% Fourier Transformtion:
Nfft = 16384 ; % Anzahl Werte fuer FFT -- mit zero padding!

% FFT (mit Delta multipliziert, damit Skalierung stimmt):
X1 = Delta * fft(Temp1,Nfft) ;
Nf1 = ; % Anzahl unabhangiger Fourier Werte?
f1 = linspace(0,Fnyq1,Nf1) ; % Frequenz-Vektor

subplot(3,1,2)
plot (f1,abs(X1(1:Nf1)))
xlabel('Frequenz (Perioden/Jahr)')
pause % Mit enter Taste weiter
% Wenn Sie die genauen Frequenzen sehen wollen:
subplot(3,2,3)
plot (f1,abs(X1(1:Nf1))), axis([0,5,0,12])
subplot(3,2,4)
plot (f1,abs(X1(1:Nf1))), axis([360,370,0,12])
pause
% Zur besseren Darstellung, mit logarithmischer Frequenzachse:
subplot(3,1,3)
semilogx (f1,abs(X1(1:Nf1))), axis([0.1,1000,0,12])
xlabel('Frequenz (Perioden/Jahr)')
pause

% Jetzt wiederholen wir das Ganze, aber "messen" nur noch
% zweimal pro Monat ueber die gleiche Messperiode von 3 Jahren:
-----
Fahbst2 = ; % was ist jetzt die Abtastrate?
Fnyq2 = ; % Nyquist Frequenz?
Deltat2 = ; % Abtastintervall?
N2 = ; % Anzahl Messwerte?
t2 = ; % Zeit-Vektor?
Temp2 = ; % Temperatur-Vektor?

figure(2)
subplot(3,1,1)
plot (t2,Temp2), axis([0,max(t2),-15,15])
xlabel('Zeit (Jahre)')

Nfft = 1024 ; % Anzahl Werte fuer FFT -- mit zero padding!
X2 = ; % FFT?
Nf2 = ; % Anzahl Frequenzwerte?
f2 = ; % Frequenz-Vektor?

subplot(3,1,2)
plot (f2,abs(X2(1:Nf2)))
subplot(3,1,3)
semilogx (f2,abs(X2(1:Nf2))), axis([0.1,1000,0,12])
xlabel('Frequenz (Perioden/Jahr)')

% Vergleichen Sie die zwei Resultate: Was schliessen Sie daraus?
% (Skript S. 36-39)

```

```

% Zeitreihenanalyse -- Ubung WS09
% =====

% Vorgehen:
% Kopieren Sie sich die m-files dieser Ubung in Ihr Arbeitsverzeichnis:
% cp /afs/ethz.ch/users/d/leichman/zr/ws09/aufgabe1.m
% cp /afs/ethz.ch/users/d/leichman/zr/ws09/aufgabe2.m
% Ergaenzen Sie die fehlenden Angaben und fuhren Sie anschliessend
% die Ubung mit den Befehlen aufgabe1 und aufgabe2 aus.

% Aufgabel: Amplitudenspektren einer regellosen Zeitreihe.
% Es sollen verschiedene Fenster und
% Fensterlaengen probiert und die Ergebnisse
% verglichen werden.

clear

% Holen Sie die Zeitreihe s:
% (sdata ist ein binaires MAT-file
% in dem die Variable s definiert ist)
load /afs/ethz.ch/users/d/leichman/sdata % lesen der Zeitreihe s

fs = 128; % Abtastrate (Hz)
fn = fs/2; % Nyquist Frequenz (Hz)
delta = 1/fs; % Abtastintervall (s)

% zeichne 1 Sek. dieser Zeitreihe

N = 128;
t = linspace(0,(N-1)*delta,N) ; % zeitvektor
figure(1)
clf
subplot(2,1,1)
plot(t,s(1:N)), xlabel('zeit (s)')

% Frage: Was ist der Frequenzinhalt dieser Zeitreihe ?
% Tipp: Die Zeitreihe besteht aus mehreren Sinus-Schwingungen
% plus weissem Rauschen -- koennen Sie die Amplituden
% der einzelnen Sinus-Schwingungen bestimmen (es sind
% ganze Zahlen)?

Nfft = N;
S = ; % FFT ueber 1 Sekunde von s
nf = Nfft/2 + 1; % Anzahl Frequenzen
f = linspace(0,fn,nf); % Frequenz Vektor
subplot(2,1,2)
plot (f,abs(S(1:nf))), xlabel('Frequenz, Hz')

% Versuch mit zero-padding:

Nfft = 1024;
% FFT ueber 1 Sekunde von s mit zero padding bis Nfft:
S = ; % FFT
nf = Nfft/2 + 1; % Anzahl Frequenzen
f = linspace(0,fn,nf); % Frequenz Vektor
figure(2)
clf
subplot(2,1,1)
plot (f,abs(S(1:nf))), xlabel('Frequenz, Hz')

% Versuch mit laengerem Datenfenster (2 Sekunden):
N = 256; % 2 Sekunden
Nfft = N;
S = ; % FFT ueber 2 Sekunden von s
nf = Nfft/2 + 1; % Anzahl Frequenzen
f = linspace(0,fn,nf); % Frequenz Vektor
plot (f,abs(S(1:nf))), xlabel('Frequenz, Hz')

% Versuchen Sie es noch mit anderen Fensterlaengen!
% (zum Beispiel mit N=1024 = 8 Sek. oder mit der gesamten Zeitreihe)
% Vergleich mit einer anderen Fensterfunktion:
N = 256;
w = hanning(N); % Fensterfunktion (Spaltenvektor)
w = w'; % Transponieren
t = linspace(0,(N-1)*delta,N) ; % zeitvektor
% zeichne die gefenstersten daten
figure(3)
clf
subplot(2,1,1)
plot (t,s(1:N).*w), xlabel('zeit (s)')
hold on
plot (t,w*max(s(1:N)),'r') % zeichne Fensterfunktion
hold off

Nfft = N;
nf = Nfft/2 + 1;
f = linspace(0,fn,nf); % Frequenz Vektor
figure(4)
clf
subplot(2,1,1)
S = ; % FFT ueber s ohne Fensterfunktion
plot (f,abs(S(1:nf))), xlabel('Frequenz, Hz')
subplot(2,1,2)
S = ; % FFT ueber s mit Fensterfunktion
plot (f,abs(S(1:nf))), xlabel('Frequenz, Hz')

```

```

% durch N dividiert werden.
% Der Betrag des so normierte Leistungsspektrums ist unabhangig
% von der Laenge des analysierten Zeitfensters.
% In dieser Form gilt dies jedoch nur fuer ein Rechteck-Fenster.
% Damit das Resultat fuer irgendein Fenster w mit beliebig
% Laenge N nicht von der Fensterlaenge abhaengt, muss es wie
% folgt skaliert werden:
scale = norm(w)^2 / sum(w)^2;
plot (ff,pd*scale), xlabel('Frequenz (Hz)');
axis ([0,25,0,5])

% Dabei ist sum(w) die Summe ueber alle Werte von w und
% norm(w) die Wurzel aus der Summe der Quadrate von w
% (s. help norm oder help sum).
% Vergewissern Sie sich, dass in diesem Falle eines Rechteck-
% Fensters diese Skalierung tatsaechlich einer Division durch N
% gleich kommt.

% Im Falle anderer Fenster, kompensiert der so definierte
% Skalierungsfaktor den Einfluss des Fensters auf die
% Amplituden der urspruenglichen Zeitreihe.
% Verifizieren Sie dies, indem Sie ein anderes Fenster
% (z.Bsp. hanning statt boxcar) verwenden.

% Da uns keine unendliche Datenmenge zur Verfuegung steht,
% koennen wir mit den uns zur Verfuegung stehenden Methoden
% bestenfalls einen Schaehtwert des tatsaechlichen Leistungs-
% spektrums berechnen. Es laesst sich zeigen, dass die Varianz
% des wie oben berechneten Leistungsspektrums unabhangig von
% jeweiligen Erwartungswertes des Leistungsspektrums selber ist.
% Ein laengeres Zeitfenster verbessert nur die Aufloesung des
% Spektrums aber nicht die Varianz. Wie bei anderen Messverfahren,
% laesst sich auch bei der Abschaehtung des Leistungsspektrums die
% Varianz durch Mittelwertbildung ueber mehrere Beobachtungen
% reduzieren. Hierzu wird die zu analysierende Zeitreihe in
% einzelne kuerezeere Zeitfenster zerlegt und die Spektren der
% einzelnen Segmente gemittelt. Dies geht aber auf Kosten der
% spektralen Aufloesung. Mit der Matlab-Routine PSD kann
% dies auf einfache Weise gemacht und gleichzeitig koennen auch
% die Konfidenzintervalle berechnet werden:
ueberlap = 0;
p = 0.95;
[pc,pd,ff] = psd(s(1:2048),N,fs,w,ueberlap,p,'none');

% Die Fensterlaenge ist gleich wie vorher aber nun werden
% 2048 Werte der Zeitreihe analysiert und ueber 2048/256=8
% nicht-ueberlappende (ueberlap=0) Segmente gemittelt.
% Das Konfidenzintervall fuer 95% (p=0.95) ist im Array
% pc abgespeichert.

```

4

```

% Aufgabe 2: Abschaehtung des Leistungsspektrums sowie der
% Varianz und Konfidenzintervalle
% -----
clear

load /afs/ethz.ch/users/d/deichman/sdata % Lesen der Zeitreihe s

N = 256;
fs = 128; % Abtastrate (Hz)
delta = 1/fs; % Abtastintervall (s)
t = linspace(0,(N-1)*delta,N); % Zeitvektor
figure(1)
clf
subplot(2,1,1)
plot(t,s(1:N)), xlabel('Zeit (s)')

% Berechne das Leistungsspektrum mittels PSD:
w = boxcar(N); % Rechteck-Fenster
ueberlap = 0;
[pd,ff] = psd(s(1:N),N,fs,w,ueberlap,'none');

% psd kann auf verschiedene Arten aufgerufen werden:
% In diesem Fall wird das Leistungsspektrum ueber die ersten
% N=256 Werte von s berechnet; da auch die Fensterlaenge fuer
% die FFT gleich N ist, besteht das Resultat nur aus der
% Berechnung einer FFT ueber ein einziges Segment, ohne
% Mittelwertbildung;
% fs ist die Abtastrate und wird nur benoetigt, um den
% Frequenzvektor ff fuer die graphische Darstellung zu erzeugen;
% w ist der Vektor mit der Fenster-Funktion, in unserem Fall
% ein Rechteck (boxcar);
% ueberlap gibt an um wieviel Werte die einzelnen Segmente, in
% fuer welche die FFT berechnet wird, ueberlappen sollen, in
% unserem Fall garnicht;
% 'none' besagt, dass vor der Berechnung der FFT weder der
% Mittelwert noch der Trend von den einzelnen Segmenten der
% Zeitreihe abgezogen werden soll (die anderen Optionen sind
% 'mean' oder 'linear' und keine Angabe entspricht 'none')
% Das Resultat steht im Vektor pd.

subplot(2,1,2)
plot (ff,pd), xlabel('Frequenz (Hz)')

pause

% In dieser Form entspricht pd weder der Leistungsdichte
% (power spectral density) noch der Leistung (power),
% so wie sie im Skript auf S. 41 definiert sind.
% Um die Leistungsdichte zu erhalten, muss pd mit der
% Abtastperiode delta multipliziert werden, und um die
% Leistung bzw. das Leistungsspektrum zu erhalten, muss pd

```

3

```

figure (2)
clf
subplot(2,1,1)
plot (ff,pd*scale), xlabel('Frequenz (Hz)')
hold on

plot (ff,pc(:,1)*scale,'r')
plot (ff,pc(:,2)*scale,'r')
axis ([0,25,0,5])
hold off

% Je mehr Segmente fuer die Mittelwert-Bildung zur Verfuegung stehen,
% umso kleiner wird die Varianz: fuehren Sie die Routine psd mit 32
% nicht-ueberlappenden 2 Sek langen Segmenten aus:

[pc,pc,ff] = psd(
);

subplot(2,1,2)
plot (ff,pd*scale), xlabel('Frequenz (Hz)')
hold on

plot (ff,pc(:,1)*scale,'r')
plot (ff,pc(:,2)*scale,'r')
axis ([0,25,0,5])
hold off

% Vergleichen Sie diese Ergebnisse mit den Spektren, die
% nur ueber ein Segment berechnet wurden.

% Wird z.Bsp. ueberlap = 128 gesetzt, dann bewirkt dies bei N=256,
% dass die einzelnen Segmente sich um eine halbe Fensterlaenge
% ueberlappen -- dadurch wird die Anzahl Segmente erhoeht, aber
% da diese nicht mehr statistisch unabhengig sind, fuehrt das
% mit einem Rechteck-Fenster zu keiner weiteren Reduktion der
% Varianz. In diesem Fall empfiehlt es sich eine andere Fenster-
% funktion (z.Bsp. Hanning) zu verwenden, denn dadurch werden die
% Amplituden der Zeitreihe in den sich ueberlappenden Segment-
% abschnitten unterdrueckt, was die statistische Abhaengigkeit
% verringert. Dies geht aber auf Kosten der Aufloesung. Ausserdem
% stimmen die von PSD berechneten Konfidenzintervalle dann nicht.
% Aussteuerliche Behandlungen der Berechnung der Konfidenzintervalle
% sind in den Buechern von Butkus und von Press et al. (Numerical
% Recipes) zu finden.

```

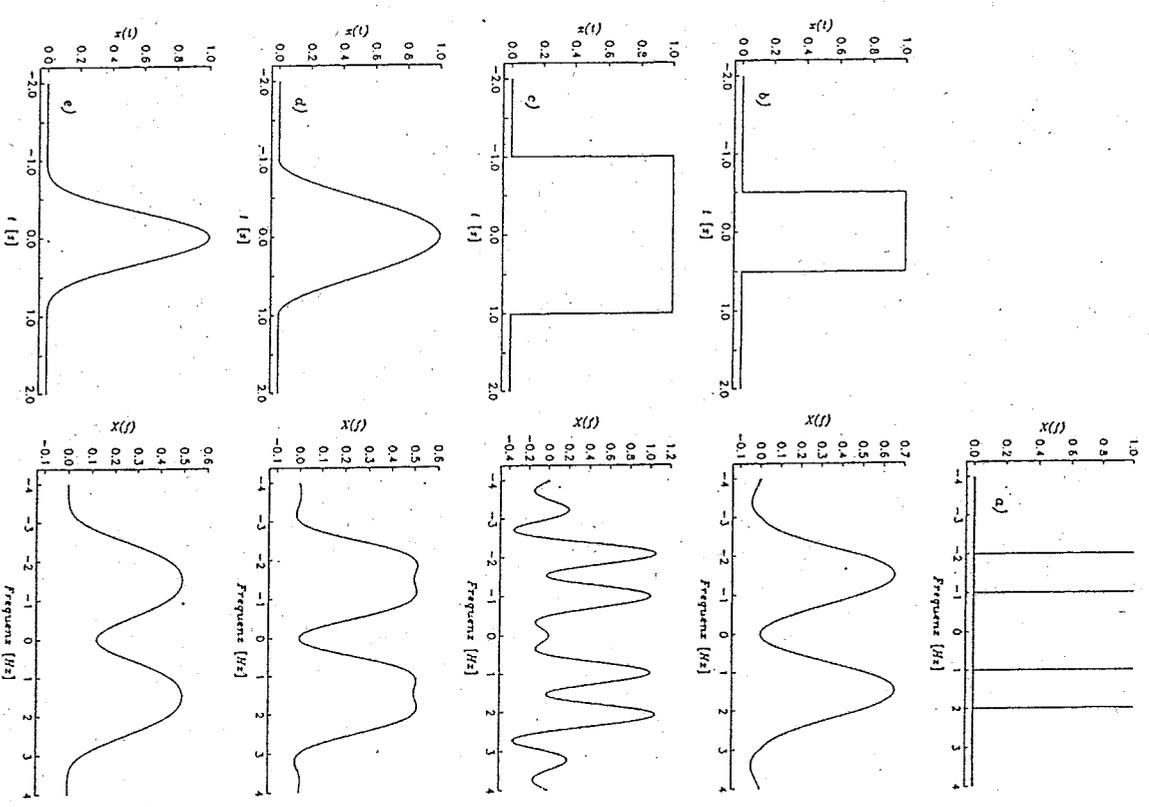


Abbildung 4.4: Der Einfluß von Länge und Gestalt der Gewichtsfunktion (links dargestellt) auf das berechnete Signalspektrum.
 a) vorgegebenes Spektrum einer Zeitfunktion, bestehend aus zwei periodischen Anteilen der Frequenz 1 Hz und 2 Hz, d.h. $\Delta f = 1$ Hz.
 b) Spektrum bei Anwendung des Rechteckfensters der Länge $T = \frac{1}{\Delta f}$;
 c) wie b), jedoch ist hier das Fenster doppelt so lang wie bei b);
 d) und e) Spektren bei Anwendung des Tukey- und Parzenfensters; Fensterlänge wie bei c).

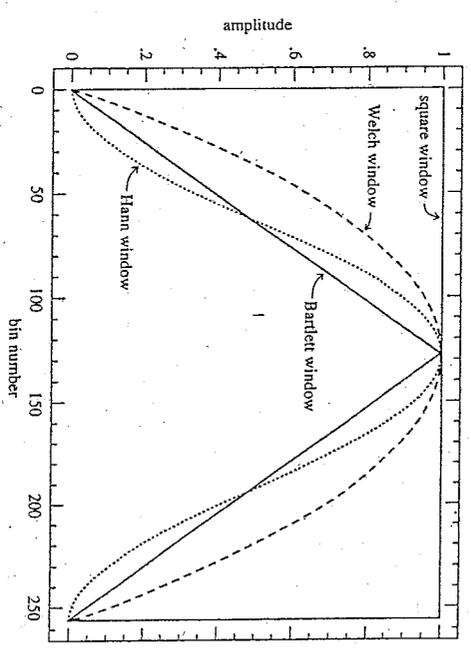


Figure 13.4.1. Window functions commonly used in FFT power spectral estimation. The data segment, here of length 256, is multiplied (bin by bin) by the window function before the FFT is computed. The square window, which is equivalent to no windowing, is least recommended. The Welch and Bartlett windows are good choices.

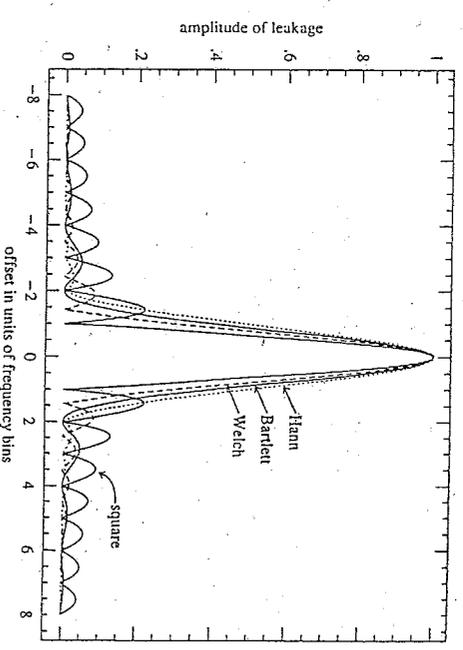


Figure 13.4.2. Leakage functions for the window functions of Figure 13.4.1. A signal whose frequency is exactly located at zero offset, "leaks" into neighboring bins with the amplitude shown. The purpose of windowing is to reduce the leakage at large offsets, where square (no) windowing has large sidelobes. Offset can have a fractional value, since the actual signal frequency can be located between two frequency bins of the FFT.

```

% Zeitreihenanalyse -- Ubung WS10
% =====
% Aufgabe: Frequenzanalyse von Abflussdaten der Landeshydrologie
% -----

% Aus dem Leistungsspektrum der Abflussdaten an zwei Messpunkten
% der Landeshydrologie an der Rhone und am Rhein wollen wir die
% periodischen Anteile identifizieren und deuten.

% Die Daten befinden sich auf
% /afs/ethz.ch/users/d/deichman/lhg/

% Die Datenfiles sind
% pdsx0598 fuer die Station Porte du Scex (Rhone) und
% diep1998 fuer die Station Diepoldsau (Rhein).
% Sie beinhalten die Tagesmittelwerte in Kubikmeter/Sekunde der
% Rhone und des Rheins fuer die Jahre 1905-1998 bzw. 1919-1998.
% Die Datenfiles bestehen aus einem Wert pro Tag pro Zeile und
% koennen mit dem Befehl load von Matlab gelesen werden -- zum
% Beispiel fuer die Daten der Rhone:

s = load('/afs/ethz.ch/users/d/deichman/lhg/pdsx0598');

% Tipp:
% Verwenden Sie fuer die Abtastrate fs = 365.25; dann koennen
% Sie die Daten wie folgt als Funktion der Zeit in Jahre
% darstellen:

fs = 365.25;
ns = length(s)
t0 = 1905;
t = linspace(t0,t0+(ns-1)/fs,ns);

figure(1)
subplot(2,1,1)
plot(t,s)
xlabel('Jahr')
ylabel('Abfluss (m^3/s)')

```

```

% Versuchen Sie mit der Matlab Routine psd wie fuer die
% synthetischen Zeitreihen (letzter Teil der zweiten Aufgabe
% der Ubung WS09) das Leistungsspektrum abzuschätzen.
% Berechnen und zeichnen Sie auch die Konfidenzintervalle
% (p = 0.95).
% Entfernen Sie aber den Mittelwert (psd mit 'mean' statt 'none'
% aufrufen) und waehlen Sie eine Segmentlaenge N=1024 sowie
% ein Rechteckfenster

w = boxcar(N);

% Probieren Sie verschiedene Darstellungsmoeglichkeiten aus
% (plot, semilogx, semilogy, loglog).

% Sie werden sehen, dass zusaetzlich zum erwarteten Jahresgang,
% der durch die jahreszeitlich bedingte Schneeschmelze verursacht
% wird, noch weitere periodische Anteile vorhanden sind.
% Was ist deren Ursache?

% Tipp:
% Bei genauem hinsehen sieht man, dass das Erscheinungsbild
% der Zeitreihe sich mit der Zeit veraendert -- versuchen Sie mal
% das Leistungsspektrum fuer die erste und die zweite Haelfte
% der Zeitreihe getrennt zu berechnen, mit

nm = fix(ns/2); % damit nm ein Integer ist falls ns ungerade ist
% und dann psd fuer s(1:nm) bzw. fuer s(nm:ns) aufrufen.

% Was koennte die Ursache fuer die zeitliche Veraenderung sein?
% Verhalten sich die Messungen an beiden Stationen genau gleich?
% Haben alle Maxima im Spektrum eine physikalische Ursache?

```

```

% Zeitreihenanalyse -- Ubung WS11
% =====
% Die M-Files dieser drei Aufgaben befinden sich in
% /afs/ethz.ch/users/d/deichman/zr/ws11
% unter uebung1.m, uebung2.m und uebung3.m
%
% Aufgabel: Vergleich der FT eines Delta-Impulses mit
% derjenigen von "weissem Rauschen".
% -----
% Kopieren Sie sich die zwei M-Files "dirac.m" und "weisss.m"
% vom Directory /afs/ethz.ch/users/d/deichman/mfiles/
% in Ihr eigenes Arbeitsdirectory.
% Schauen Sie sich die zwei M-Files genauer an: Es sind
% Beispiele von M-Funktionen. Im Gegensatz zu M-Script-Files,
% sind hier Input- und Output-Variablen definiert und die
% internen Variablen erscheinen nicht im Arbeitsplatz.
% Mit help dirac oder help weiss erhalten Sie eine
% Beschreibung dieser zwei M-Files.
clear
N = 256;
% Generieren Sie einen Delta-Impuls
% (Amplitude = 1, an der Stelle i=11)
x = dirac(N,11,1);
% und eine gleichlange Zeitreihe mit "weissem Rauschen"
y = weiss(N);
% Y = randn(1,N)/sqrt(N);
% zeichnen Sie die zwei Signale
n = (0:1:N-1);
subplot(3,2,1)
plot(n,x), axis([0,20,0,1]) % nur die ersten 20 Werte
title('Dirac'), xlabel('Zeitwerte')
subplot(3,2,2)
plot(n,y), axis([0,N-1,-0.5,0.5]) % die ganze Zeitreihe
title('Weisses Rauschen'), xlabel('Zeitwerte')
% Fuehren Sie nun die FFT aus und zeichnen Sie
% die Amplitudenspektren
nf = N/2 + 1;
f = linspace(0,0.5,nf); % Annahme: Delta = 1
X = fft(x);

```

1

```

X = X(1,1:nf);
subplot(3,2,3)
plot(f,abs(X)), axis([0,0.5,0,21])
xlabel('Amplitude'), xlabel('Frequenz')
Y = fft(y);
y = Y(1,1:nf);
subplot(3,2,4)
plot(f,abs(Y)), axis([0,0.5,0,21])
xlabel('Amplitude'), xlabel('Frequenz')
% Zeichnen Sie das Phasenspektrum
% (der MatLab-Befehl angle berechnet das Argument einer komplexen
% Zahl; mittels Division durch pi erhaelt man Einheiten in
% Vielfache von pi)
subplot(3,2,5)
px = (angle(X))/pi;
%px = unwrap(angle(X))/pi;
plot(f,px), axis([0,0.5,round(min(px)),round(max(px))])
xlabel('Phase (*Pi)'), xlabel('Frequenz')
subplot(3,2,6)
py = (angle(Y))/pi;
%py = unwrap(angle(Y))/pi;
plot(f,py), axis([0,0.5,min(py),max(py)])
xlabel('Phase (*Pi)'), xlabel('Frequenz')
% FRAGE: koennen Sie sich die Aehnlichkeiten und Unterschiede
% zwischen den zwei Zeitreihen und ihren Fourier Transformationen
% erklaeeren?
% Tipp: Versuchen Sie zu verstehen, wie das "weisse Rauschen"
% generiert wurde.
% Als Alternative fuehren Sie diese Ubung nochmals aus, indem
% Sie das weisse Rauschen mit dem Befehl
% Y = randn(1,N)/sqrt(N);
% generieren -- randn generiert normalverteilte Zufallszahlen mit
% Mittelwert = 0 und Varianz = 1.
% Wie veraendert sich das Phasenspektrum des Delta-Impulses wenn
% Sie ihn an die Stelle i=1 oder an eine andere Stelle verschieben?
% Weisen Sie nach, dass das Verhalten des Phasenspektrums korrekt
% ist, indem Sie die Beziehung fuer die Fourier Transformation einer
% zeitlich verschobenen Funktion (Skript S. 24) fuer den diskreten
% Fall beweisen.
% Um die Phasensprunge von 2*pi der arctan Funktion zu vermeiden
% koennen Sie px und py auch mit der Funktion unwrap berechnen
% (s. oben oder help unwrap).

```

2

```

% Aufgabe 2: Vergleich der Zeitreihen weiss und randn
% -----
% Mit der Matlab Funktion r = randn(1,N) wird ein Vektor
% von normalverteilten Zufallszahlen mit Varianz = 1 generiert.
% Frage: gehorcht das weisse Rauschen, das mit der Funktion
% weiss generiert wird, der gleichen Wahrscheinlichkeits-
% verteilung wie diejenige von randn?
% Vorgehen:
clear
N = 8192;
w = weiss(N)*sqrt(N); % Faktor sqrt(N) damit Varianz = 1 ist
var(w)
r = randn(1,N);
var(r)

% Vergleichen Sie die kumulative Haeufigkeitsverteilung der
% zwei Zeitreihen oder die Haeufigkeitsverteilung der Werte.
sw = sort(w);
sr = sort(r);
figure(2)
subplot(2,1,1)
plot(sw,'green')
hold on
plot(sr,'red')
hold off
xlabel('Kumulative Haeufigkeit')
ylabel('Wert der Zeitreihe')

b = linspace(-5,5,21);
subplot(2,2,3)
hist(w,b)
subplot(2,2,4)
hist(r,b)

```

```

% Aufgabe 3: Leistungsspektren von weissem Rauschen
% -----
% Vergleichen Sie die Leistungsspektren der zwei Zeitreihen
% weiss und randn,
% in dem Sie mit der Routine psd verschiedenen lange Zeitsegmente
% analysieren.
% Vorgehen:
clear

```

```

N = 8192;
w = weiss(N)*sqrt(N); % Faktor sqrt(N) damit Varianz = 1 ist
r = randn(1,N);

% n = 64;
% n = 256;
% n = 1024;
% n = 8192;

% Annahme: Abtastrate = 1
[psw,ff] = psd(w,n,1,boxcar(n),0,'none');
[psr,ff] = psd(r,n,1,boxcar(n),0,'none');

figure(3)
plot(ff,psr,'red')
hold on
plot(ff,psw,'green')
hold off

% Vergleichen Sie die Varianz von weiss bzw. randn mit dem Mittelwert
% der jeweiligen Resultate der Routine psd (mit den Matlab Befehlen
% var und mean).
% Beweisen Sie an Hand der Beziehung von Parseval und der Definition
% des Leistungsspektrums (Skript S. 41-42), dass diese Resultate
% richtig sind.

```

```

function x = dirac(N,i,a)
%DIRAC generiert einen Dirac-Impuls
% Aufruf: x = dirac(N,i,a)
% x = Dirac-Impuls
% N = Anzahl werbe
% i = Position des Impulses in x
% a = Amplitude des Impulses
% Beispiel:
%
% x = dirac(64,32,1)
x(1,1:N) = zeros(1,N) ;
x(i) = a ;

```

```

function x = weiss(N)
%WEISS generiert weisses Rauschen mit amplitudenspektrum = 1.
% Aufruf: x = weiss(N)
% x = Vektor mit weissem Rauschen
% N = Anzahl Werte (Zweiterpotenz von Vorteil)
% Beispiel:
%
% x = weiss(256)
N2 = N/2 ;
N2m1 = N2 - 1 ;
z1 = 2*rand(1,N2m1)-1 + sign(rand(1,N2m1)-0.5) * j ;
z1 = real(z1) + imag(z1) * sqrt(1-real(z1) * real(z1)) * j ;
z2 = conj(Flipplr(z1)) ;
zz(1) = 1 + 0j ;
zz(2:N2) = z1(1:N2m1) ;
zz(N2+1) = -1 + 0j ;
zz(N2+2:N) = zz(1:N2m1) ;
x = real(ifft(zz)) ;

```

```

% Zeitreihenanalyse -- WS12
=====

% Diese Übung besteht aus zwei Teilen: Ein erster Teil
% soll die Unterschiede zwischen Kovarianz und Korrelation,
% sowie zwischen den verschiedenen Normierungen an Hand
% von synthetischen Zeitreihen aufzeigen; der zweite Teil
% ist eine Anwendung auf Niederschlags- und Grundwasser-Daten.

% Sie koennen das M-file des ersten Teiles dieser Übung in
% Ihr Arbeitsverzeichnis kopieren und ohne Aenderungen
% ausfuehren:

% cp /afs/ethz.ch/users/d/deichman/zr/ws12/aufgab1.m .

% Versuchen Sie zu verstehen, was darin ablaeuft und modifizieren
% Sie dann die entsprechenden Abschnitte fuer die Aufgabe 2.
% (s. Skript S. 44-50.)

% Aufgabe 1: Test verschiedener Normierungen der
% Auto- und Kreuzkorrelationen mit
% synthetischen Signalen

clear

% Sinus-Signal
ns = 288;          % Anzahl Werte
fs = 12;          % Abtastrate
t1 = 0;
t = linspace(t1,t1+(ns-1)/fs,ns);
x = sin(2*pi*t);
figure(1)
subplot(3,1,1)
plot(t,x)
title('X: Sinus')

% Weisses Rauschen
y = randn(1,288)*2;
subplot(3,1,2)
plot(t,y)
title('Y: Weisses Rauschen')

% Summe von Sinus und Rauschen
z = x+y;
subplot(3,1,3)
plot(t,z)
title('X+Y=Z: Summe von Sinus und Weisses Rauschen')
xlabel('zeit (Jahre)')

% Autokorrelationen (biased)
figure(2)
xx = xcov(x,x,'coeff');
yy = xcov(y,y,'coeff');
zz = xcov(z,z,'coeff');
nlag = length(xx);
lag = [-fix(nlag/2):fix(nlag/2)]; % Zeitachse fuer die Korrelation
subplot(3,1,1)
plot(lag,xx)
title('XX: Autokorrelation vom Sinus')
subplot(3,1,2)
plot(lag,yy)
title('YY: Autokorrelation vom weissen Rauschen')
subplot(3,1,3)
plot(lag,zz)
title('ZZ: Autokorrelation vom Sinus+Rauschen')
xlabel('Lag (Monate)')

% Autokorrelationen unterschiedlich normiert:
figure(3)
subplot(3,1,1)
plot(lag,xcov(x,x,'unbiased')) / (std(x)*std(x))
title('XX: Autokorrelation vom Sinus (unbiased)')
subplot(3,1,2)
plot(lag,xcov(y,y,'unbiased')) / (std(y)*std(y))
title('YY: Autokorrelation vom Sinus+Rauschen (unbiased)')
subplot(3,1,3)
plot(lag,xcov(z,z,'biased'))
title('ZZ: Autokovarianz vom Sinus+Rauschen (biased)')
xlabel('Lag (Monate)')

% Kreuzkorrelationen
figure(4)
subplot(3,1,1)
plot(lag,xcov(x,y,'coeff'))
title('XY: Korrelation vom Sinus und Rauschen (biased)')
subplot(3,1,2)
plot(lag,xcov(x,y,'unbiased'))
title('XY: Kovarianz vom Sinus und Rauschen (unbiased)')
subplot(3,1,3)
plot(lag,xcov(x,y,'unbiased')) / (std(x)*std(y))
title('XY: Korrelation vom Sinus und Rauschen (unbiased)')
xlabel('Lag (Monate)')

```

```

% Aufgabe 2: Auto- und Kreuzkorrelation von
Grundwasser- und Niederschlagsdaten

% Das M-File zu dieser Aufgabe koennen Sie sich wiederum
% in Ihr Arbeitsverzeichnis kopieren und die fehlenden Teile
% gemuess Ihren Erkenntnissen aus der Aufgabe 1 ergaenzen:

% cp /afs/ethz.ch/users/d/deichman/zr/ws12/aufgabe2.m
ns = 288;
fs = 12;
t1 = 1975;
t = linspace(t1,t1+(ns-1)/fs,ns);
figure(5)

% Einlesen der Grundwasserdaten
y = load ('/afs/ethz.ch/users/d/deichman/lhg/wasser.dat')
subplot(2,1,1)
plot (t,y)
title('Maienfeld: Grundwasserspiegel Monatsmittel')
xlabel('m.u.m.')

% Einlesen der Niederschlagsdaten
z = load ('/afs/ethz.ch/users/d/deichman/klima/regen.dat')
subplot(2,1,2)
plot (t,z)
title('Chur: Niederschlag Monatssummen')
xlabel('mm')
ylabel('Jahr')

% Autokorrelationen (biased)
figure(6)
yy = ;
zz = ;
nlag = length(xx);
lag = [-fix(nlag/2):1:fix(nlag/2)]; % Zeitachse fuer die Korrelation
subplot(3,1,1)
plot (lag,yy)
title('Grundwasser Autokorrelation')
subplot(3,1,2)
plot (lag,zz)
title('Niederschlag Autokorrelation')

% Kreuzkorrelationen (biased)
yz = ;
subplot(3,1,3)
plot (lag,yz)
title('Korrelation Grundwasser und Niederschlag')
xlabel('Lag (Monate)')

```

```

% Vergleichen Sie die Lage der Maxima der Korrelationen:
find (yy==max (yy))
find (zz==max (zz))
find (yz==max (yz))

```

```

% Mit einer Vergrößerung des zentralen Teils der letzten
% Abbildung koennen Sie (grob) die Verzögerung zwischen
% Grundwasserspiegel und Niederschlag sichtbar machen.

```

```
axis ([-5,5,-0.5,0.5])
```

```

% Fuer eine bessere Auflöesung, muesste man z. Bsp. Tages-
% werte statt Monatswerte verwenden.

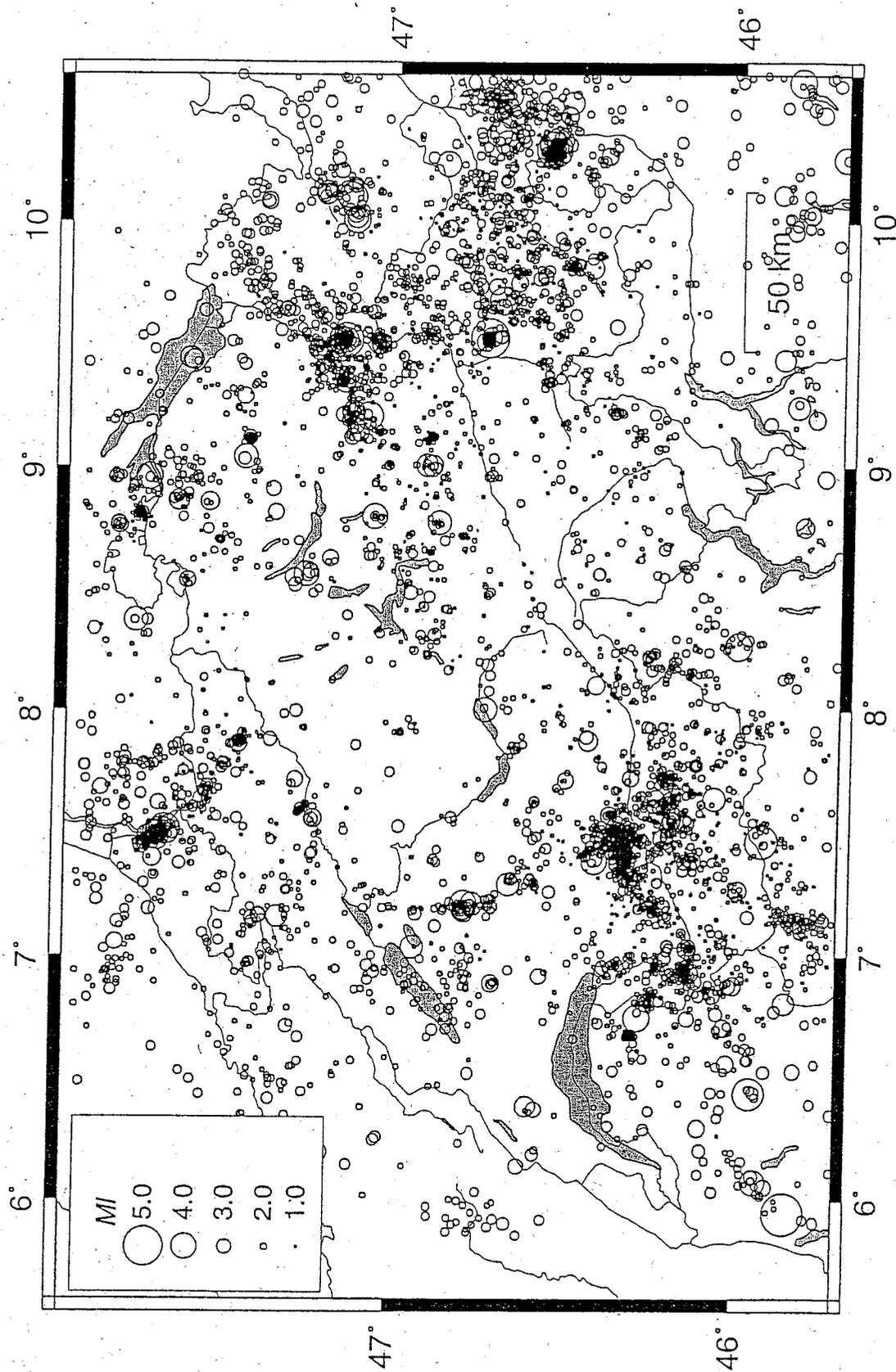
```

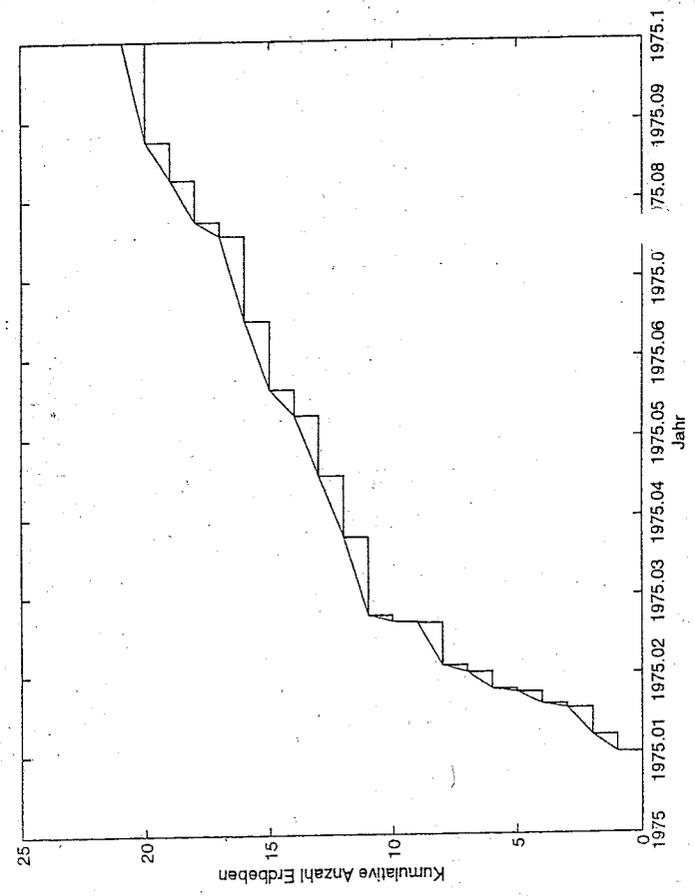
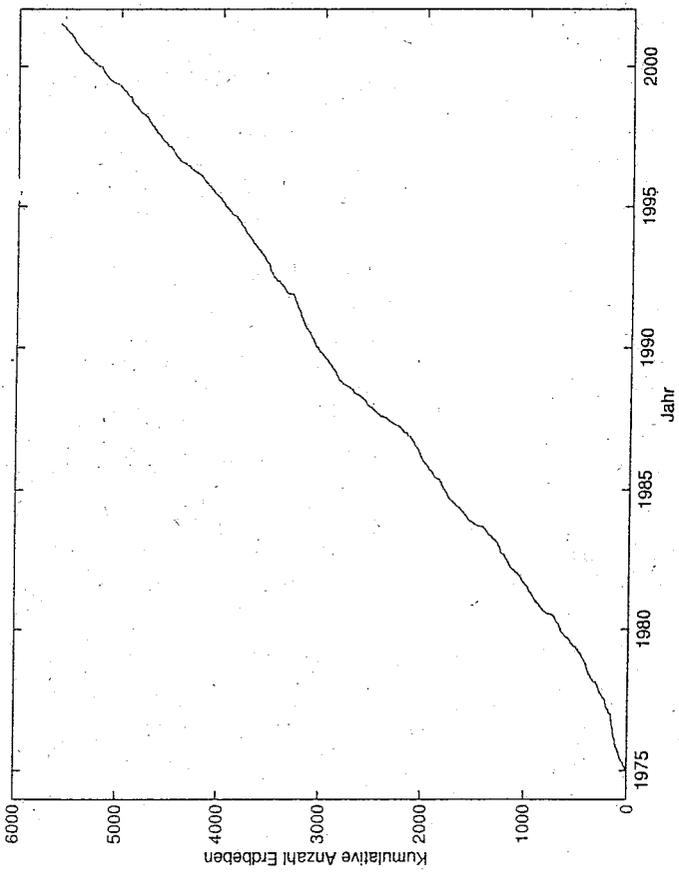
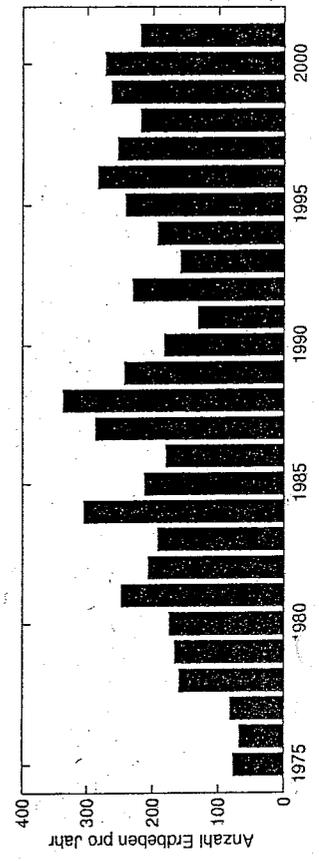
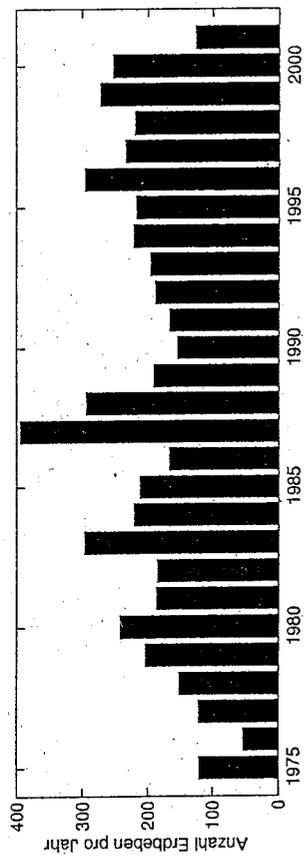
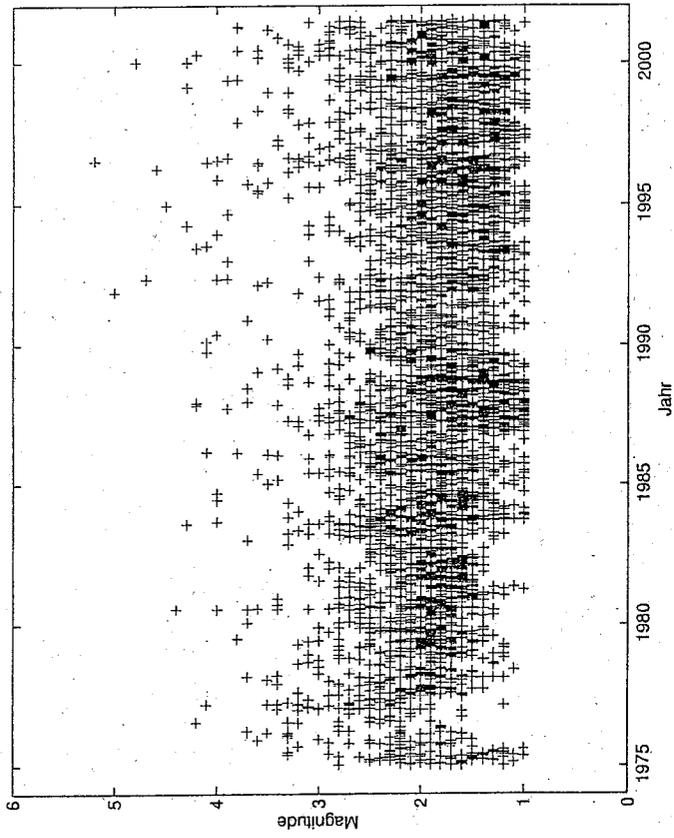
```

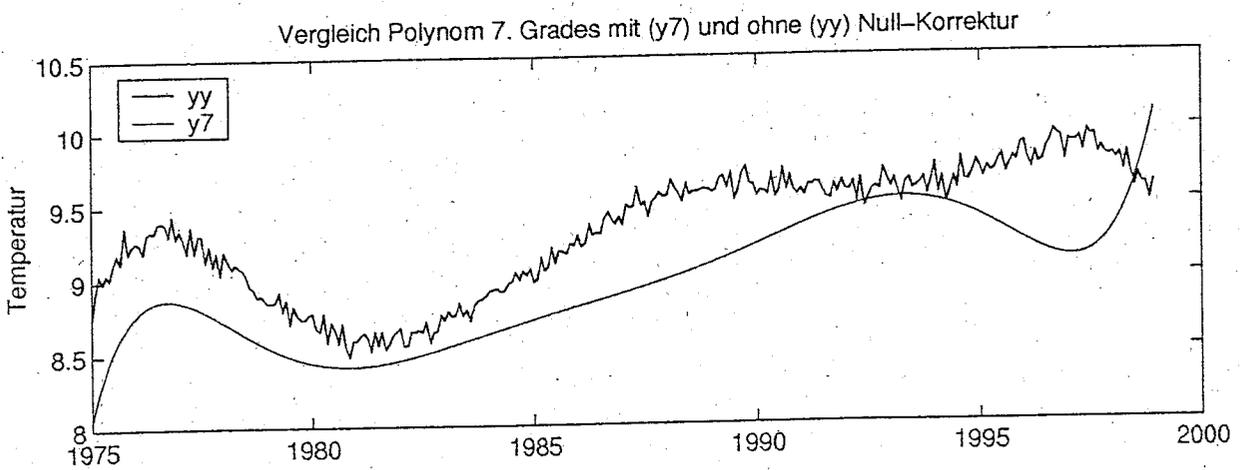
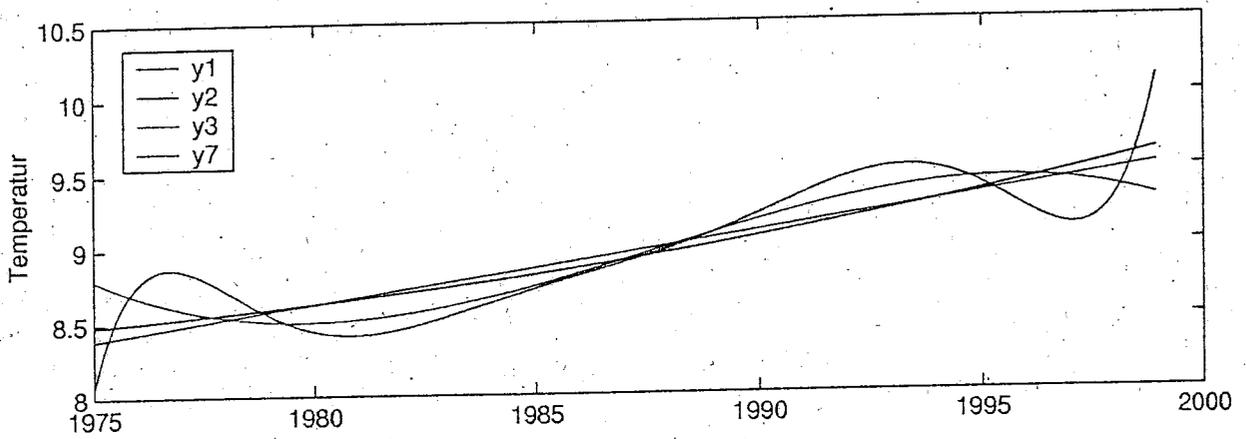
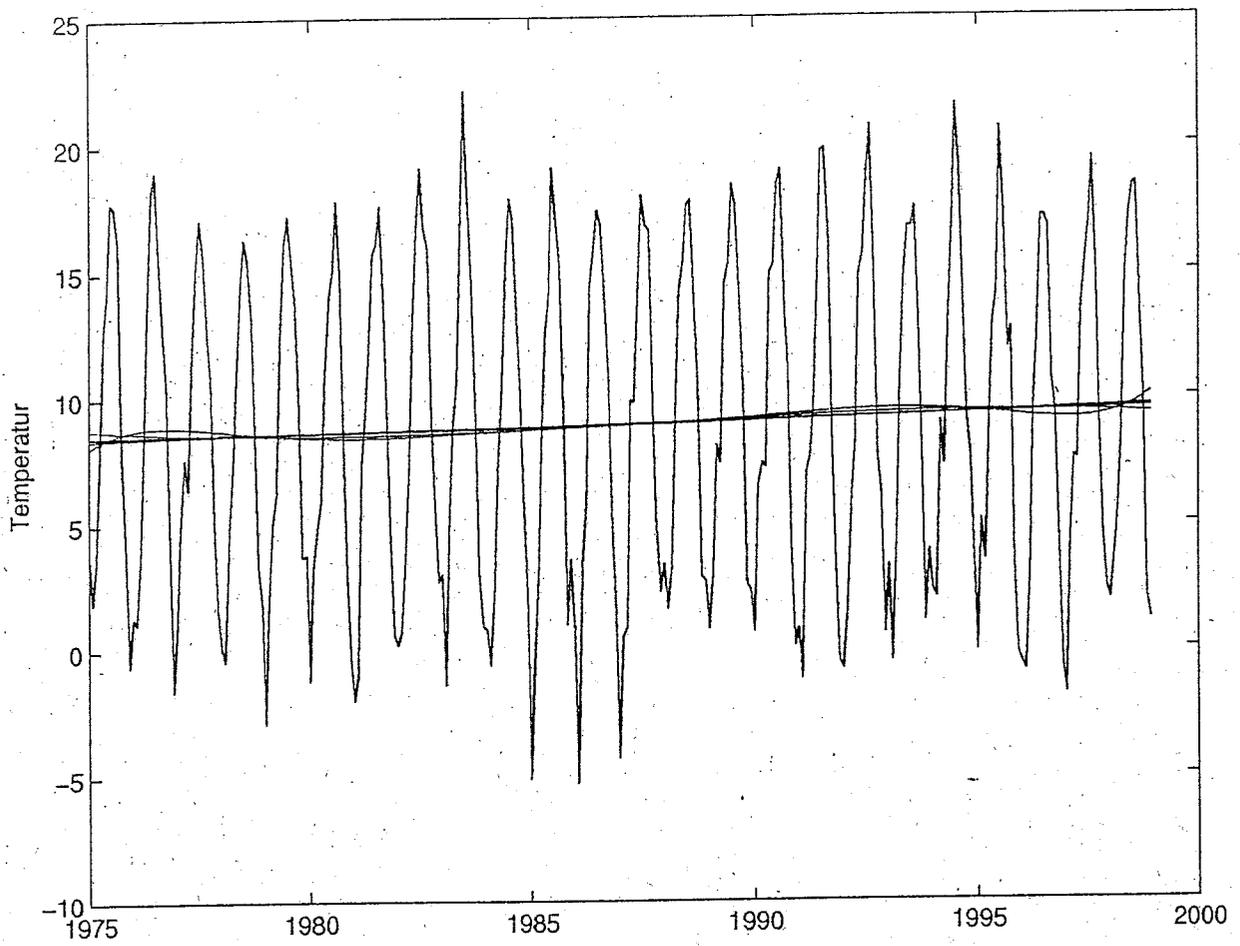
% Vergleichen Sie diese Resultate mit einer "unbiased" Normierung.

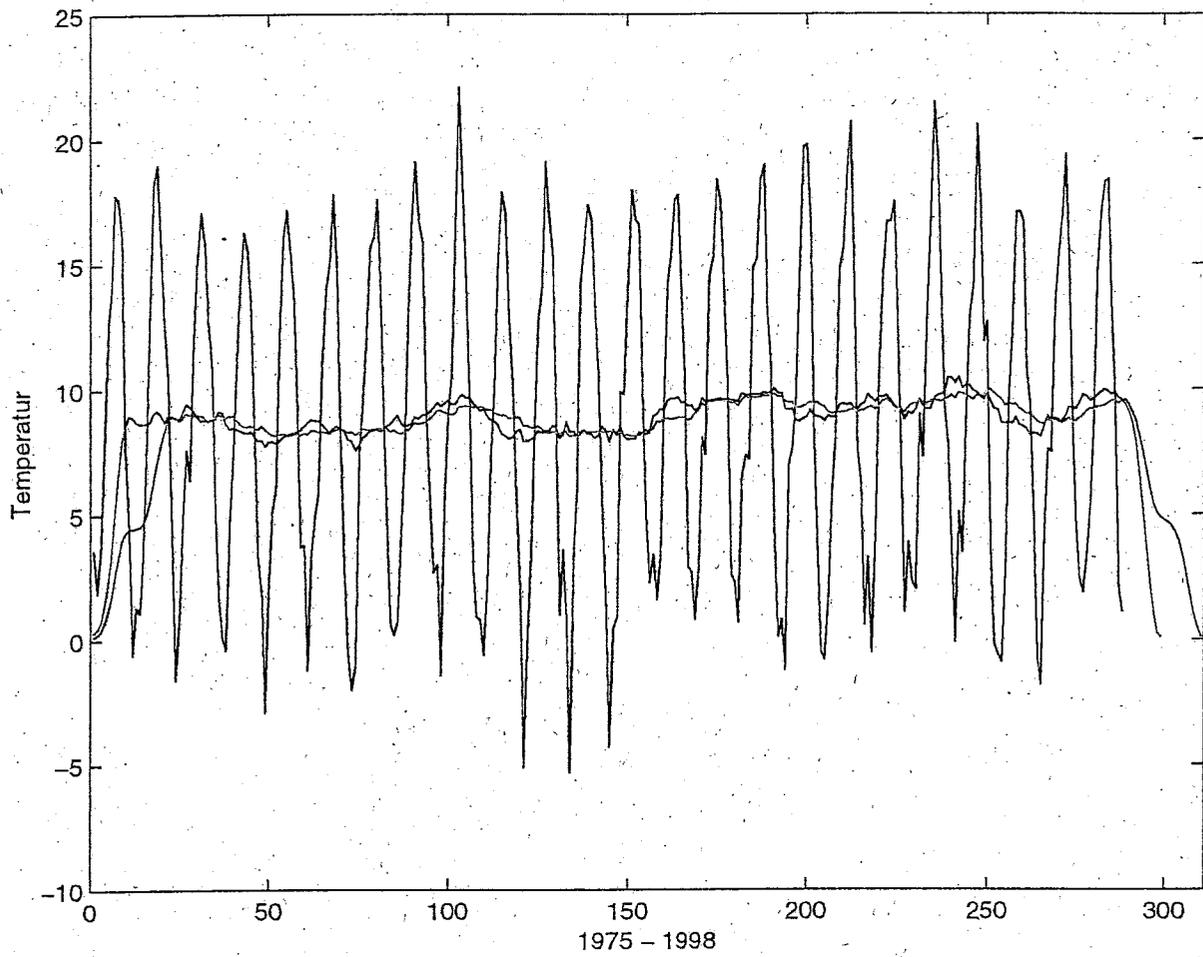
```

1975 - June 2001 M>1









Ergänzung zur DFT

Die Diskrete Fourier Transformation (DFT) und ihre Rücktransformation ist wie folgt definiert:

$$X_k = \Delta t \sum_{n=0}^{N-1} x_n e^{-j \frac{2\pi kn}{N}} \quad k = 0, \dots, N-1$$

$$x_n = \frac{1}{N\Delta t} \sum_{k=0}^{N-1} X_k e^{+j \frac{2\pi kn}{N}} \quad n = 0, \dots, N-1.$$

Dabei ist N die Anzahl Punkte in der Zeitreihe x_n , welche als Folge $\{x_n\}$ zu verstehen ist, und dementsprechend ist auch X_k als eine komplexe Folge $\{X_k\}$ zu betrachten.

Die Hintransformation stellt eine Analyse dar, welche eine gegebene Zeitreihe x_n in ihre Frequenzanteile X_k zerlegt. X_k wird auch als *diskretes Fourier Frequenzspektrum* bezeichnet. Die Rücktransformation entspricht hingegen der Synthese einer Zeitreihe aus einzelnen harmonischen Schwingungen.

Man beachte zweierlei:

- Die komplexe Folge $\{X_k\}$ ist periodisch mit der Periode N .
- Für reelle Signale $\{x_n\}$ gilt $X_{-k} = X_k^*$ für $(0 \leq k \leq N-1)$.

Zum Beweis der ersten Aussage betrachten wir die DFT für einen einzigen beliebigen Wert von k :

$$X_k = \Delta t \sum_{n=0}^{N-1} x_n e^{-j \frac{2\pi kn}{N}} = \Delta t \sum_{n=0}^{N-1} x_n \left\{ \cos\left[\frac{2\pi kn}{N}\right] - j \sin\left[\frac{2\pi kn}{N}\right] \right\}$$

Entsprechend gilt für $k+N$:

$$X_{k+N} = \Delta t \sum_{n=0}^{N-1} x_n \left\{ \cos\left[\frac{2\pi(k+N)n}{N}\right] - j \sin\left[\frac{2\pi(k+N)n}{N}\right] \right\}$$

Durch anwenden der Summationsformeln für Cosinus und Sinus und unter Berücksichtigung, dass für ganzzahlige n , $\cos(2\pi n) = 1$ und $\sin(2\pi n) = 0$, folgt:

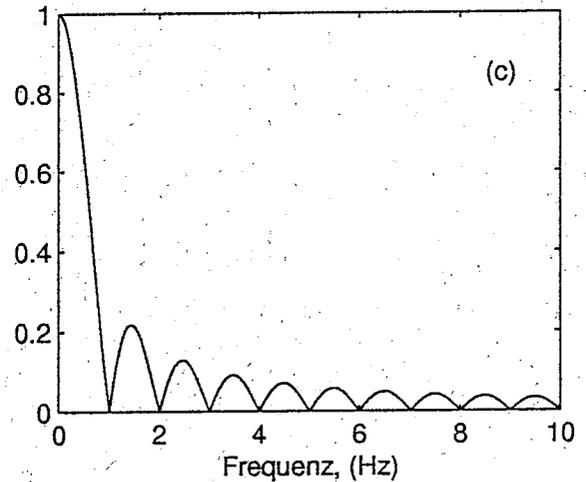
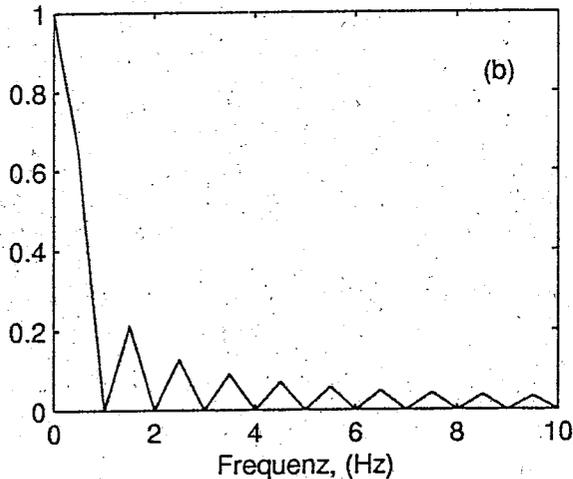
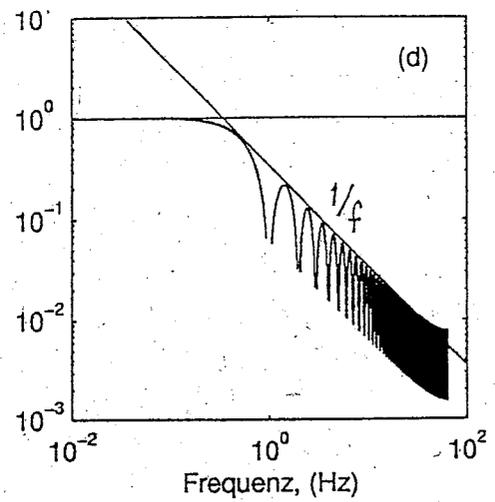
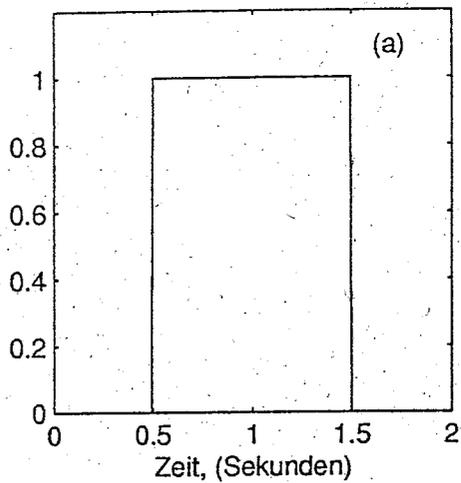
$$\cos\left[\frac{2\pi(k+N)n}{N}\right] = \cos\left[\frac{2\pi kn}{N} + 2\pi n\right] = \cos\left[\frac{2\pi kn}{N}\right]$$

und

$$\sin\left[\frac{2\pi(k+N)n}{N}\right] = \sin\left[\frac{2\pi kn}{N} + 2\pi n\right] = \sin\left[\frac{2\pi kn}{N}\right]$$

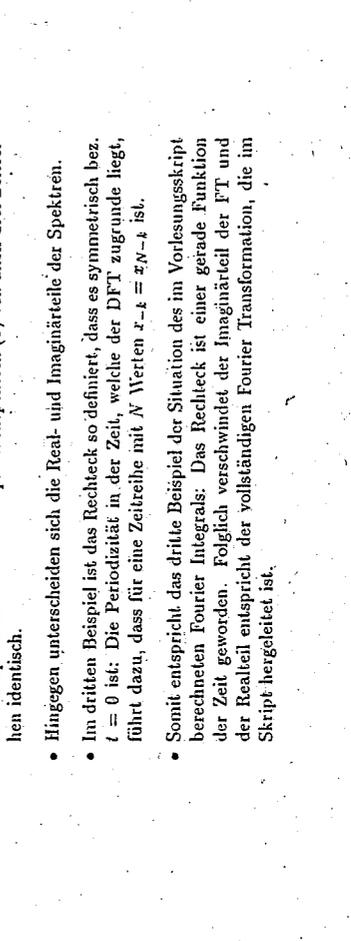
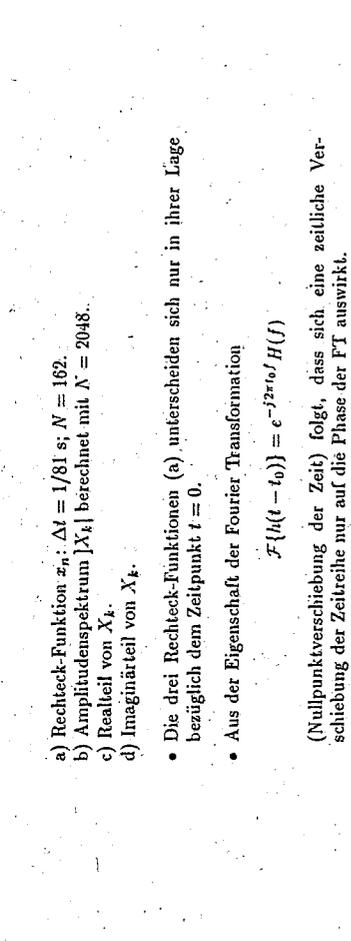
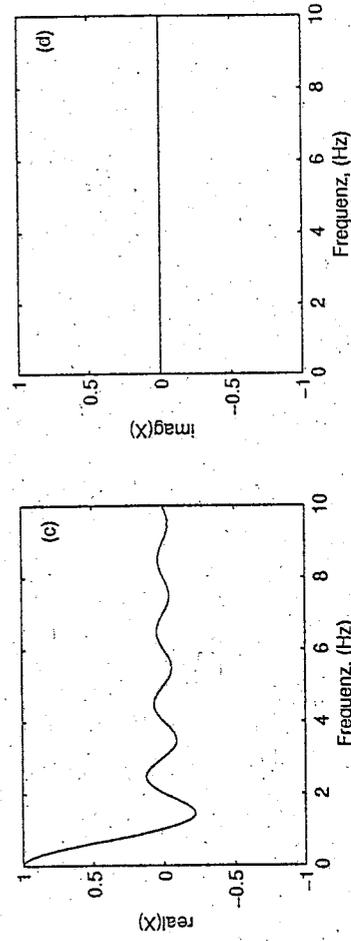
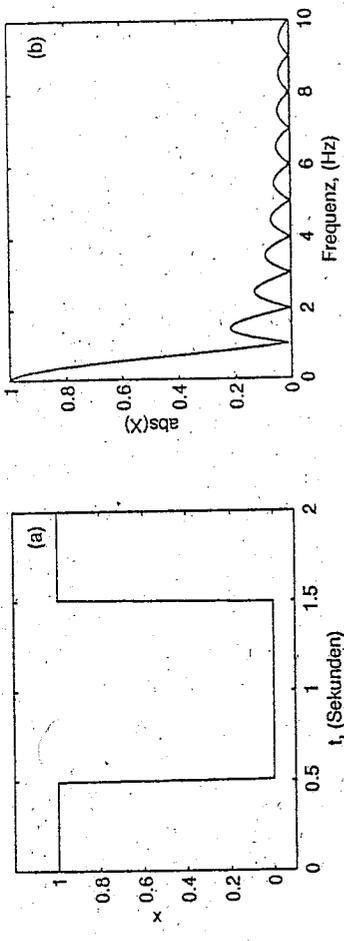
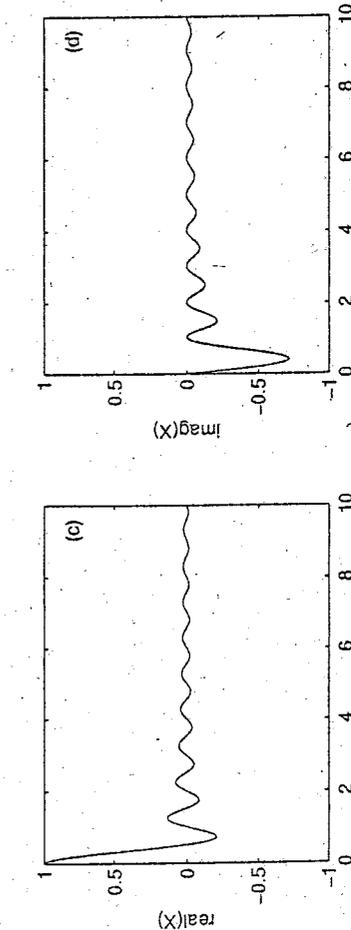
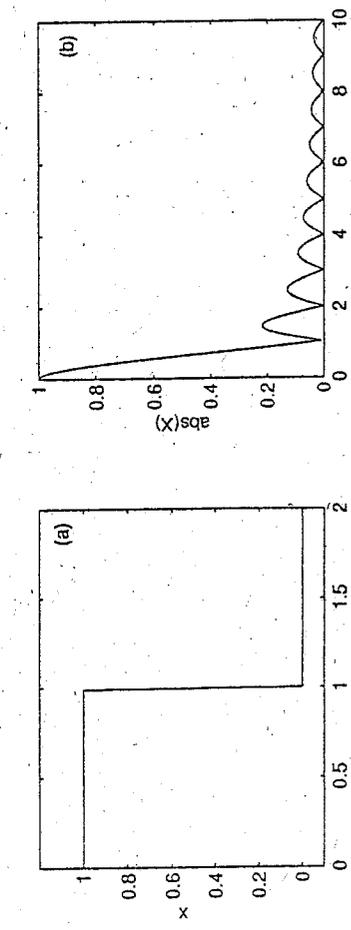
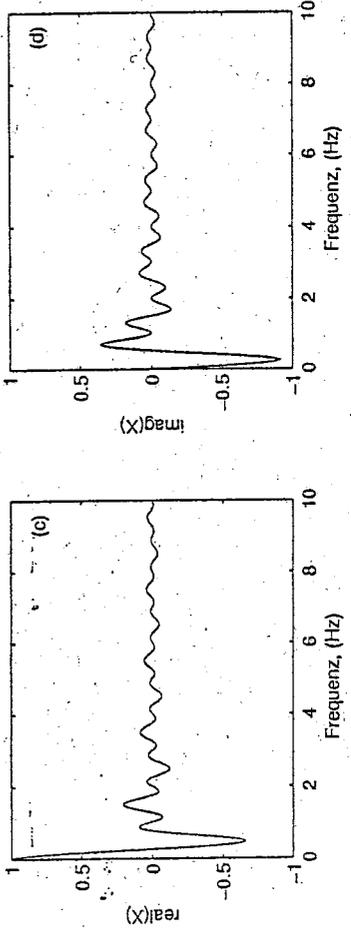
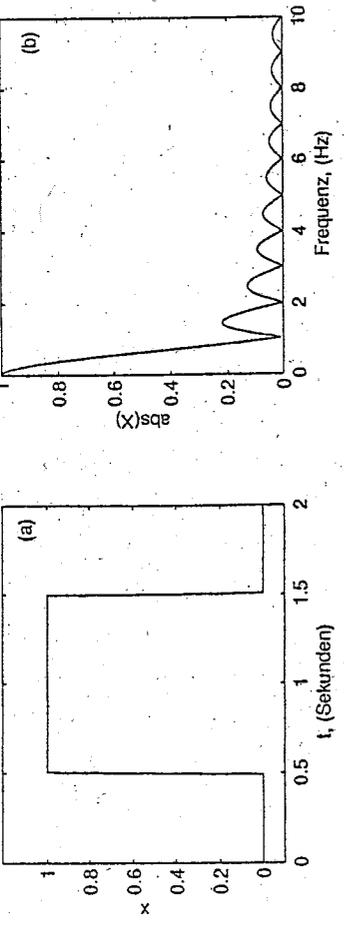
und somit ist

$$X_{k+N} = X_k.$$



- a) Rechteck-Funktion: $\Delta t = 1/128$ s; $N = 256$.
 b) Amplitudenspektrum berechnet mit $N = 256$.
 c) Amplitudenspektrum berechnet mit $N = 2048$.
 d) Wie c) in Log-Log Darstellung.

- Gemäss der Definition des Amplitudenspektrums (Betrag der Fouriertransformation) sind die negativen Spektralwerte positiv dargestellt. Somit erscheinen die Nulldurchgänge im Spektrum (vergleiche mit der Abbildung im Skript) als abwärts gerichtete Spitzen.
- Da die Abtastrate $f_s = 1/\Delta t = 128$ Hz, ist $f_{Nyq} = 64$ Hz. In b) und c) sind jedoch nur die ersten 10 Hz dargestellt.
- Das Spektrum in b) erscheint so zackig, weil die Spektralwerte genau bei den Frequenzen $k\Delta f$, ($k = 0, \dots, 128$) abgetastet werden und $\Delta f = 1/(N\Delta t) = f_s/N = 0.5$ Hz.
- Das Spektrum in c) wurde hingegen mit $N = 2048$ berechnet, somit ist $\Delta f = 0.0625$ Hz und das Spektrum erscheint wesentlich glatter.
- Die Amplitude des Spektrums bei der Frequenz 0 entspricht der Fläche der Funktion im Zeitbereich.
- Für Frequenzen oberhalb des Bereiches des Hauptmaximums ist der Amplitudenverlauf für das Rechteck proportional zu $1/f$.



a) Rechteck-Funktion x_n ; $\Delta t = 1/81$ s; $N = 162$.
 b) Amplitudenspektrum $|X_k|$ berechnet mit $N = 2048$.
 c) Realteil von X_k .
 d) Imaginärteil von X_k .

- Die drei Rechteck-Funktionen (a) unterscheiden sich nur in ihrer Lage bezüglich dem Zeitpunkt $t = 0$.
- Aus der Eigenschaft der Fourier Transformation

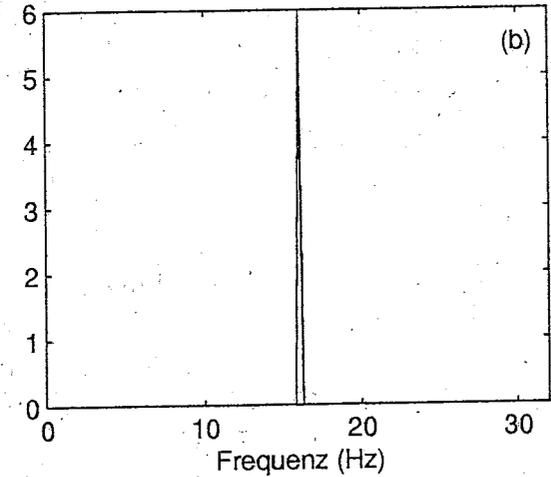
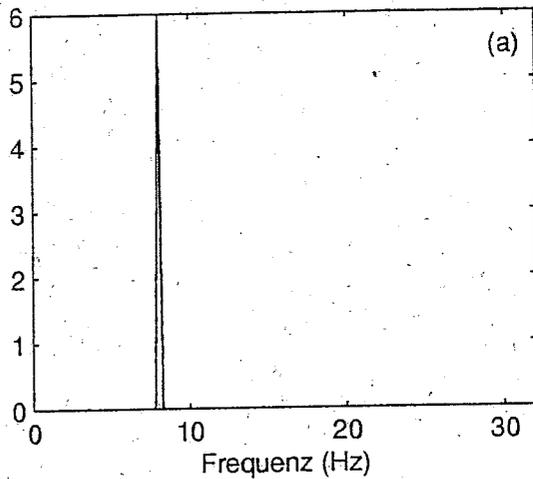
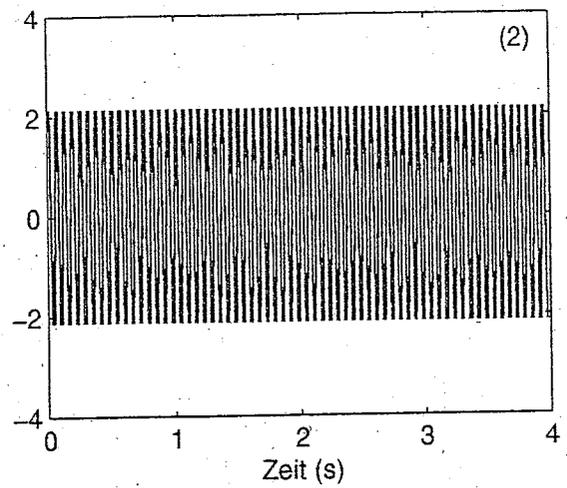
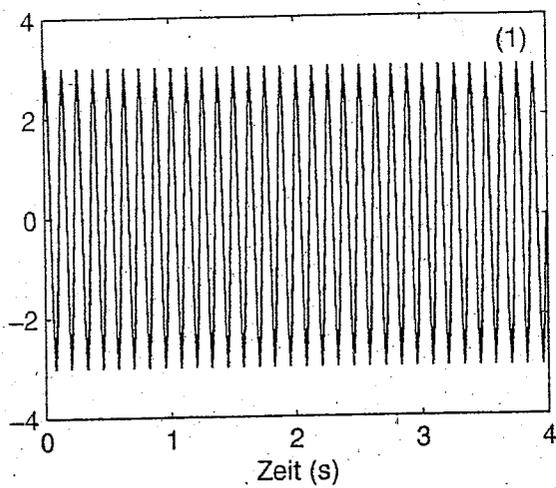
$$\mathcal{F}\{t(t - t_0)\} = e^{-j2\pi f t_0} H(f)$$

(Nullpunktverschiebung der Zeit) folgt, dass sich eine zeitliche Verschiebung der Zeitreihe nur auf die Phase der FT auswirkt.

- Dementsprechend sind die Amplitudenspektren (b) von allen drei Zeitreihen identisch.
- Hingegen unterscheiden sich die Real- und Imaginärteile der Spektren.

• Im dritten Beispiel ist das Rechteck so definiert, dass es symmetrisch bez. $t = 0$ ist: Die Periodizität in der Zeit, welche der DFT zugrunde liegt, führt dazu, dass für eine Zeitreihe mit N Werten $x_{-k} = x_{N-k}$ ist.

• Somit entspricht das dritte Beispiel der Situation des im Vorlesungsskript berechneten Fourier Integrals: Das Rechteck ist einer gerade Funktion der Zeit geworden. Folglich verschwindet der Imaginärteil der FT und der Realteil entspricht der vollständigen Fourier Transformation, die im Skript hergeleitet ist.



Die beiden harmonischen Signale haben die Form

$$x(t) = A \sin(2\pi ft + \phi)$$

mit $A = 3$, $\phi = 45^\circ$ und $f = 8$ (Signal 1) bzw. $f = 16$ (Signal 2). Beide Signale sind 4 Sekunden lang (256 Punkte) und wurden mit 64 Hz abgetastet.

Bei Signal 1 entspricht dies 8 Punkten pro Signalperiode und die Phase ist so gewählt, dass Maxima und Minima der Schwingung immer mit einem Abtastwert zusammenfallen. Somit genügt die lineare Interpolation um das vorgegebene kontinuierliche Signal amplitudengetreu wiederzugeben.

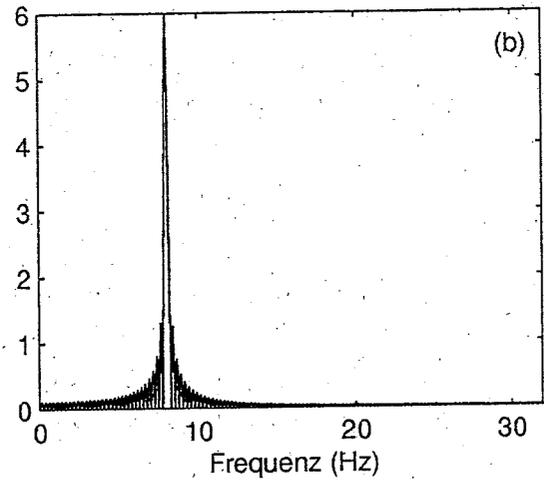
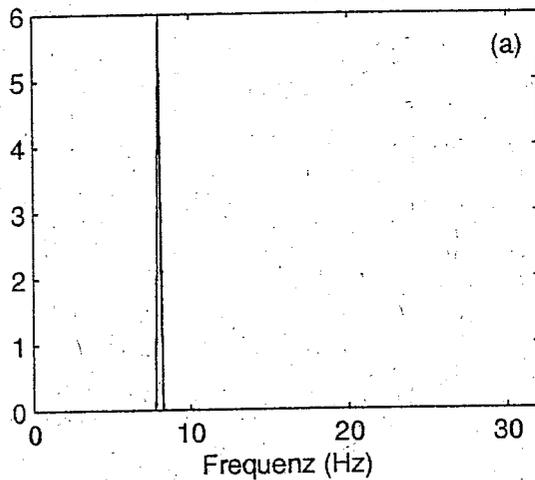
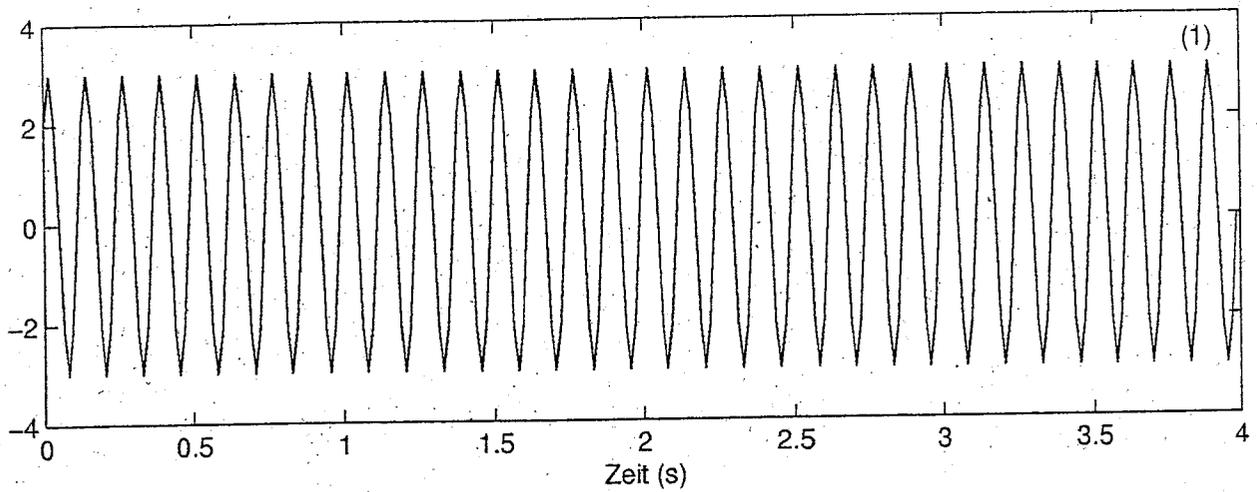
Bei Signal 2 hingegen ergibt die gewählte Abtastrate nur 4 Punkte pro Signalperiode ($1/\Delta t = 64$ Hz) und die fallen immer zusammen mit den Werten

$$2\pi fk\Delta t = \pi/4, 3\pi/4, 5\pi/4, 7\pi/4, \dots$$

Daher sind die Maxima und Minima bei $\pm 2.1 = \pm 3\sqrt{2}/2$ abgeschnitten. Obwohl die Signalfrequenz von 16 Hz noch ein Faktor 2 unterhalb der Nyquistfrequenz liegt ($f_{Nyq} = 32$ Hz) kann die lineare Interpolation das kontinuierliche Signal nicht amplitudengetreu wiedergeben.

Wie die zwei Spektren aber zeigen, ist bei der Abtastrate von 64 Hz trotzdem die volle Information im Signal enthalten: Bei einer Signallänge von $T = 4$ s und einer Signalamplitude $A = 3$ beträgt die spektrale Amplitudendichte $3T/2 = 6$, wie von der Theorie gefordert.

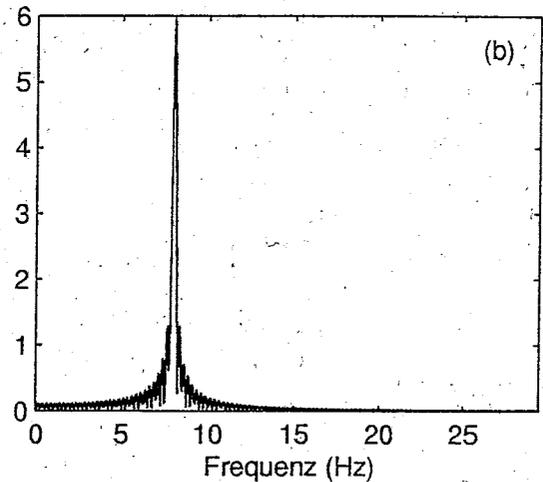
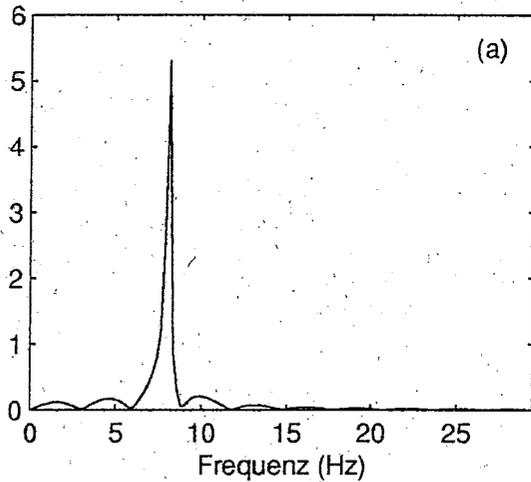
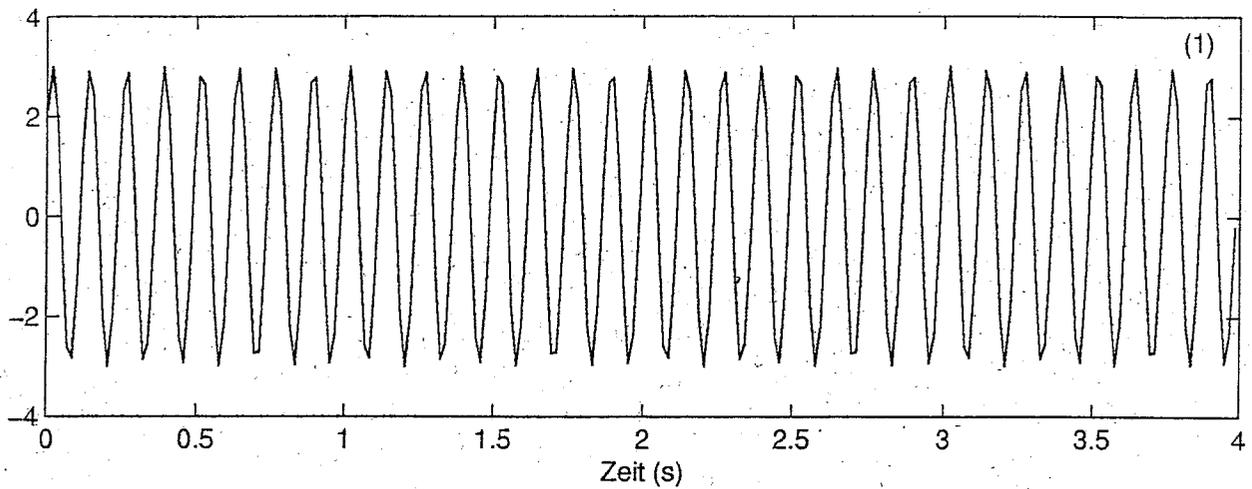
Aber wo sind die von der Theorie geforderten seitlichen Ueberschwinger aus der Faltung mit der *Sinc*-Funktion geblieben?



Das harmonische Signal 1 hat die Form $x(t) = A \sin(2\pi ft + \phi)$, mit $A = 3$, $\phi = 45^\circ$, $f = 8$, Abtastrate $f_s = 64$ Hz und Signallänge $T = 4\text{ s} = 256$ Punkten.

Das Spektrum (a) wurde mittels FFT mit $256 = 2^8$ Punkten berechnet. Bei einer Nyquist Frequenz $f_{Nyq} = f_s/2 = 32$ Hz ergibt dies $256/2 + 1 = 129$ Frequenzwerte und 128 Frequenzintervalle zwischen 0 und 32 Hz. Dies entspricht einem Frequenzintervall $\Delta f = 32/128 = 1/4$ Hz. Das Spektrum wird also bei den Frequenzwerten $k/4$ Hz ($k = 0, 1, 2, \dots, 128$) und somit auch genau bei der Frequenz $f = 8$ Hz abgetastet. Darum wird der Amplitudenwert als $T/2 = 2$ richtig wiedergegeben. Die Nebenmaxima (*side lobes*), wie sie von der Theorie zu erwarten wären, sind aber nicht sichtbar. Der Grund ist, dass die diskreten Frequenzwerte im Spektrum an den Stellen $\pm n/4$ ($n = 1, 2, \dots$) beidseits des Maximums bei 8 Hz angeordnet sind und diese Stellen genau mit den Nulldurchgängen $f_0 - f_n = \pm n/T = \pm n/4$ ($n = 1, 2, \dots$) übereinstimmen.

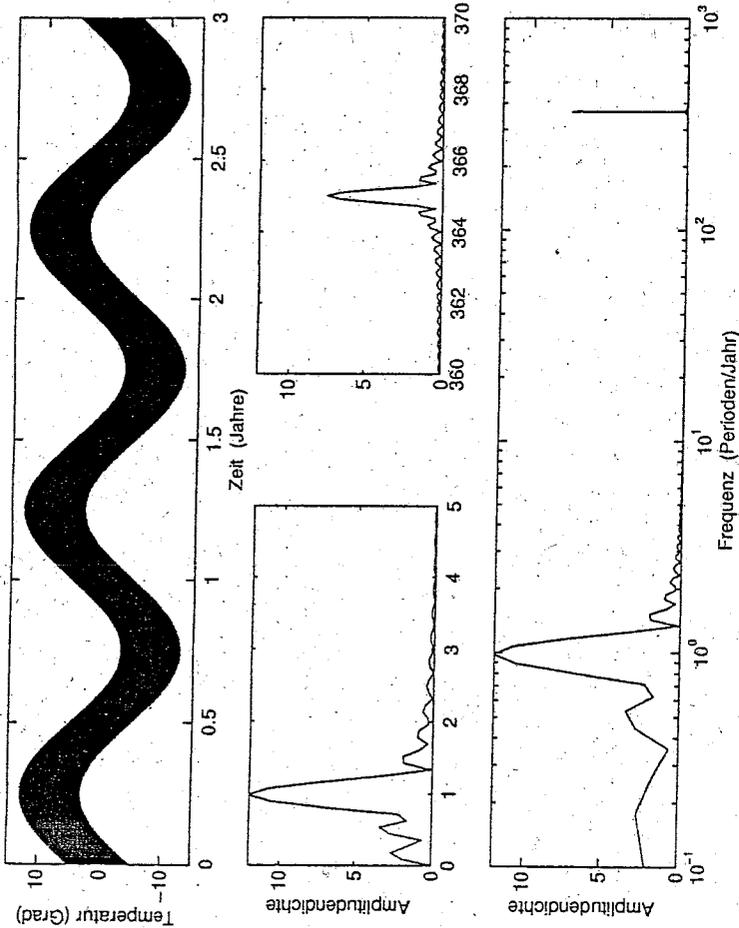
Das Spektrum (b) wurde mit $1024 = 2^{10}$ Punkten berechnet (*zero padding* von 257 bis 1024). Das ergibt 512 diskrete Frequenzintervalle zwischen 0 und 32 Hz sowie $\Delta f = 32/512 = 1/16$ Hz. Das *zero padding* verlängert das Signal natürlich nicht: T ist immer noch 4 Sekunden lang und somit sind die Nulldurchgänge im Spektrum weiterhin bei $n/4$ Hz. Aber jetzt werden auch Spektralwerte zwischen den Nulldurchgängen abgetastet und die erwarteten Nebenmaxima werden sichtbar.



Das harmonische Signal 1 hat die Form $x(t) = A \sin(2\pi f t + \phi)$, mit $A = 3$, $\phi = 45^\circ$, $f = 8$, Abtastrate $f_s = 59$ Hz und Signallänge $T = 4s = 236$ Punkten.

Das Spektrum (a) wurde mittels FFT mit $256 = 2^8$ Punkten berechnet (*zero padding* mit 20 Punkten). Bei einer Nyquist Frequenz $f_{Nyq} = f_s/2 = 29.5$ Hz entspricht dies einem Frequenzintervall $\Delta f = 29.5/128 \approx 0.23$ Hz. Das Spektrum wird also bei den Frequenzwerten $0.23k$ Hz ($k = 0, 1, 2, \dots, 128$) und somit nicht genau bei der Frequenz $f = 8$ Hz abgetastet. Die Lage des Maximums ist leicht nach rechts verschoben und der Amplitudenwert ist $5.3 \neq 3T/2$. Sowohl das Hauptmaximum als auch die Nebenmaxima sind viel breiter als erwartet: Dies ist eine Form von *spectral aliasing*.

Das Spektrum (b) wurde mit $1024 = 2^{10}$ Punkten berechnet. Das ergibt 512 diskrete Frequenzintervalle zwischen 0 und 29.5 Hz sowie $\Delta f = 29.5/512 \approx 0.0575$ Hz. Das Spektrum wird feiner abgetastet: Sowohl die Lage und Form der Maxima als auch der Maximalwert nähern sich dem theoretischen Ergebnis.



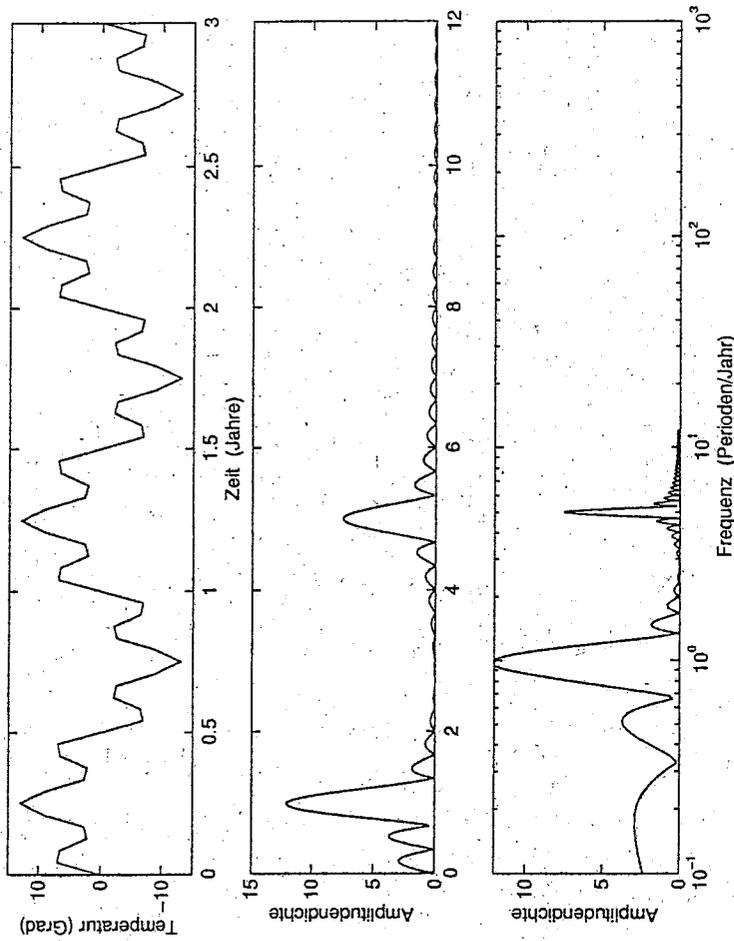
Beispiel Aliasing

In diesem Beispiel wird eine Messreihe der Lufttemperatur simuliert. Sie besteht aus einem Jahresgang von $\pm 8^\circ$ überlagert mit einem Tagesgang von $\pm 5^\circ$.

(Oben) Simulierter Temperaturverlauf: Der hochfrequente Tagesgang erscheint hier als schwarzer Balken.
Die Abtastrate ist 1460 Punkte/Jahr = 4 Messwerte/Tag.

(Mitte) Amplitudenspektrum der Temperatur-Zeitreihe oben: Wie erwartet, besteht die Messreihe aus der Überlagerung von nur zwei Frequenzen bei 1 und 365 Schwingungen/Jahr (zur besseren Darstellung ist das Spektrum in zwei gestrichelten Abschnitten gezeigt).

(Unten) Gesamtes Amplitudenspektrum aber mit logarithmischer Frequenzachse.



(Oben) Gleicher Temperaturverlauf wie vorher aber abgetastet mit nur 24 Punkten/Jahr = 2 Messwerte/Monat.

(Mitte) Amplitudenspektrum der Temperatur-Zeitreihe oben: Das Maximum bei 5 Schwingungen/Jahr ist ein Artefakt, produziert durch eine ungenügende Anzahl Messwerte – ALIASING!

(Unten) Amplitudenspektrum der Temperatur-Zeitreihe oben, aber mit logarithmischer Frequenzachse.

Amplituden Spektren einer synthetischen Zeitreihe

Die Zeitreihe besteht aus sechs harmonischen Schwingungen plus normal verteiltem weissen Rauschen mit Varianz = 6.25:

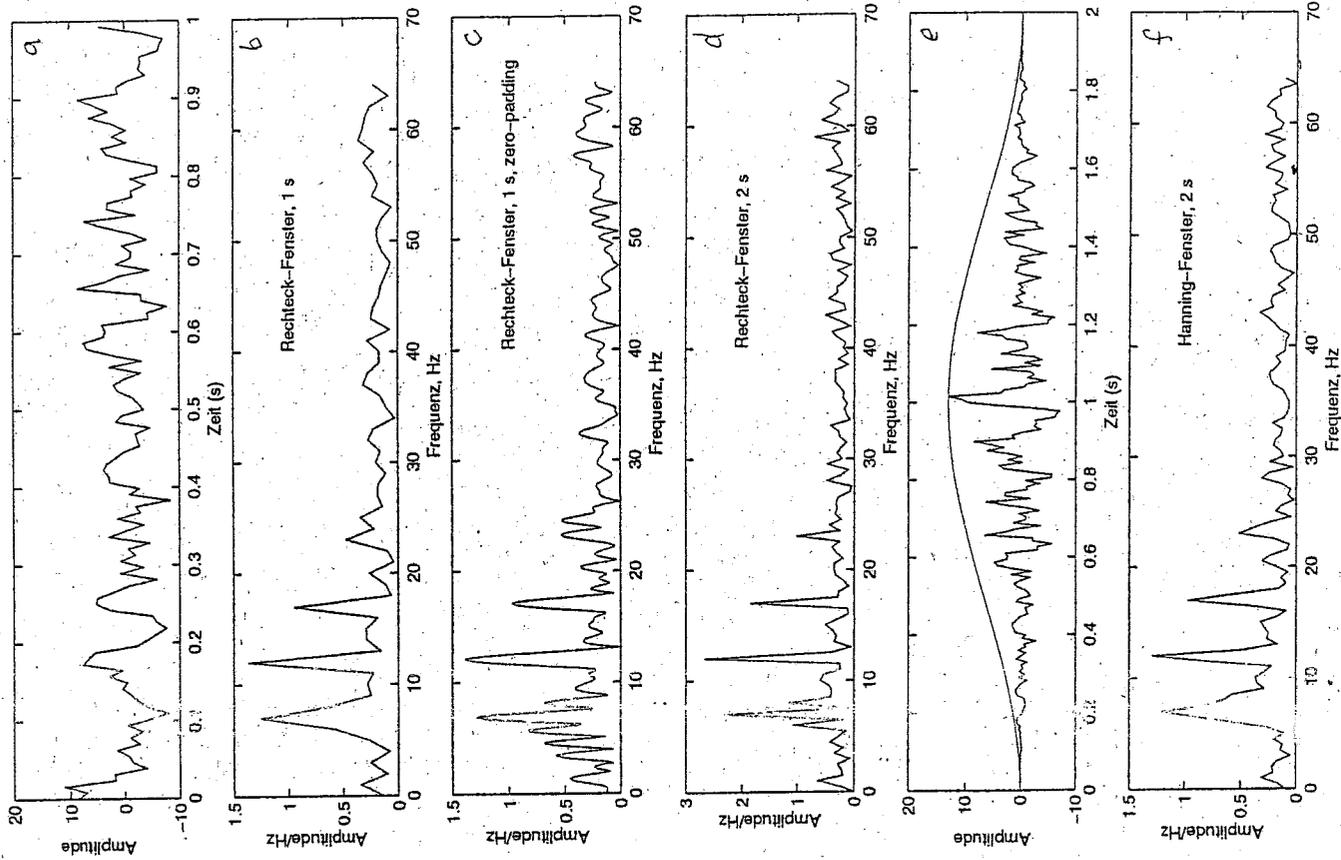
$$s = \sum_{k=1}^6 A_k \sin(2\pi f_k t + \phi_k) + w$$

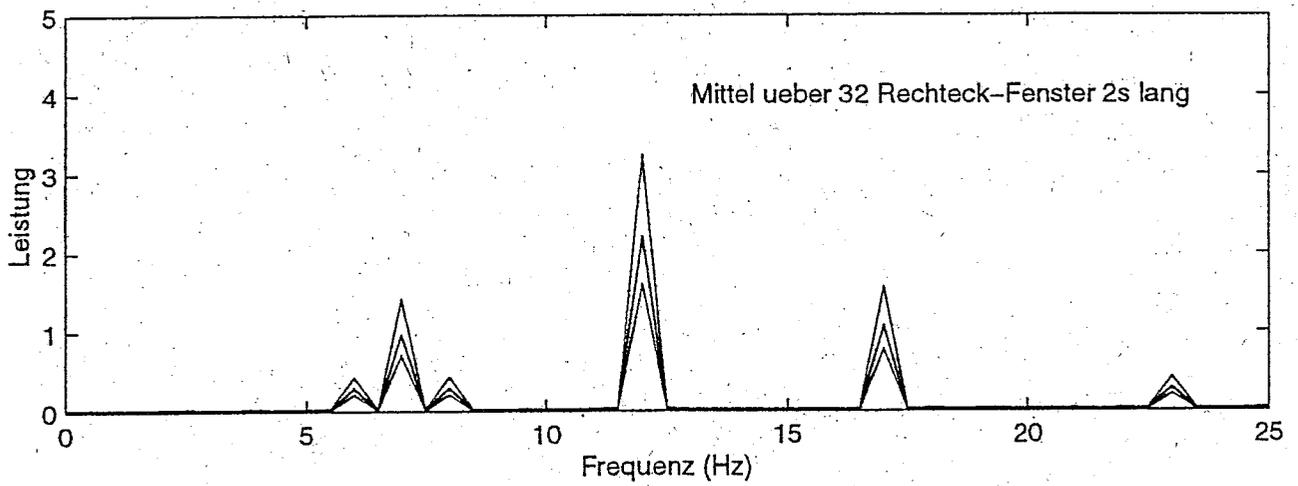
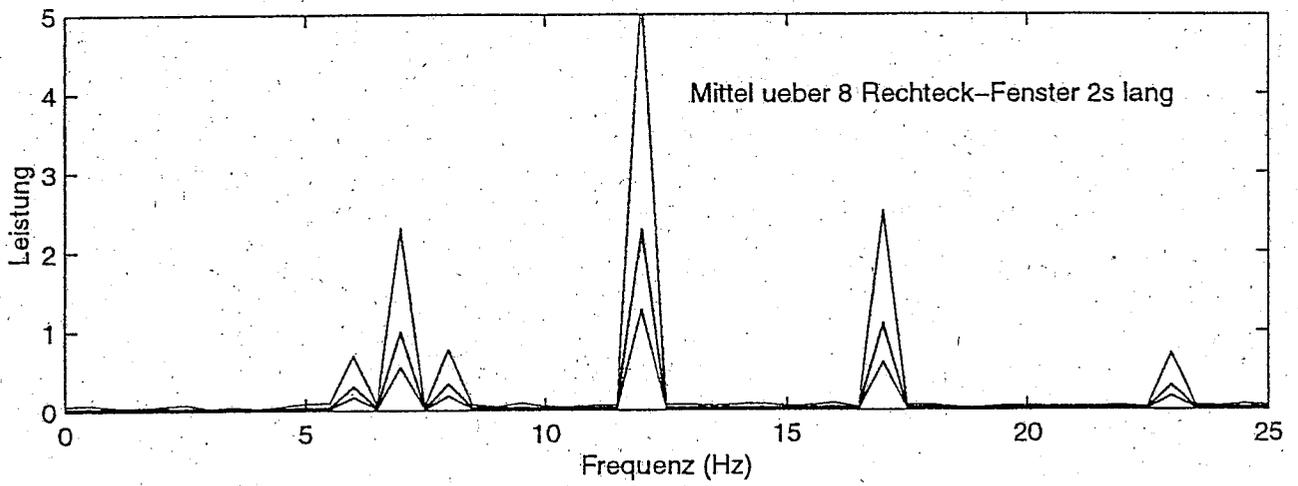
mit

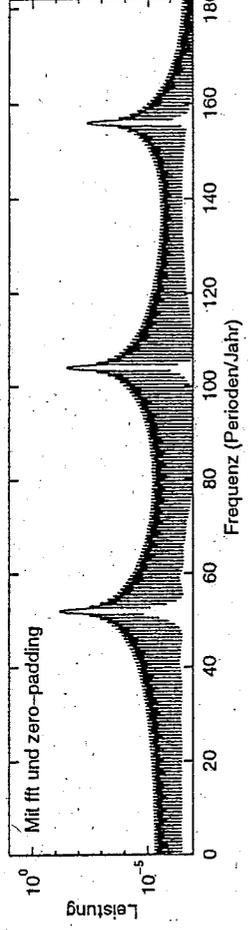
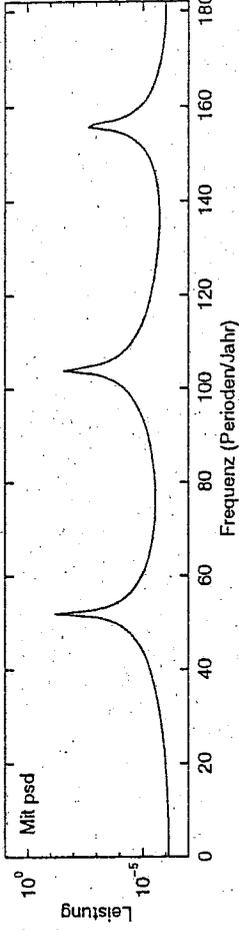
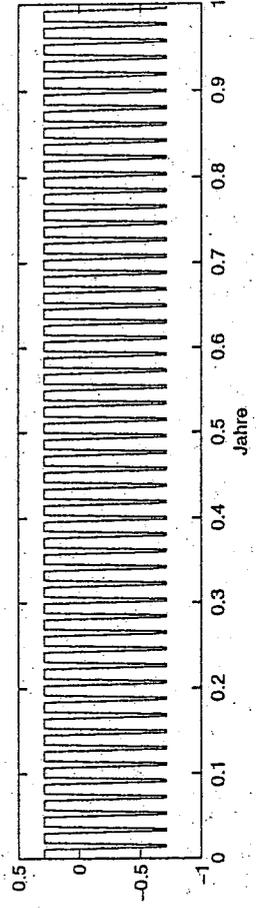
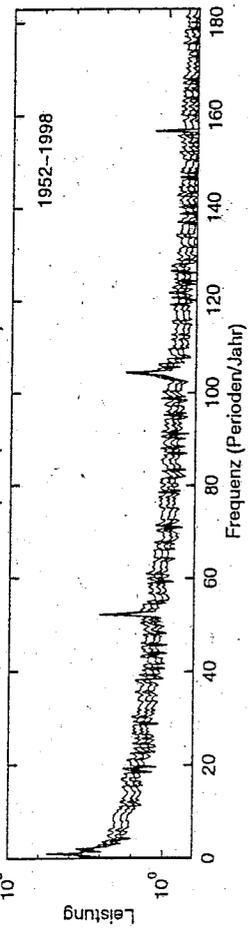
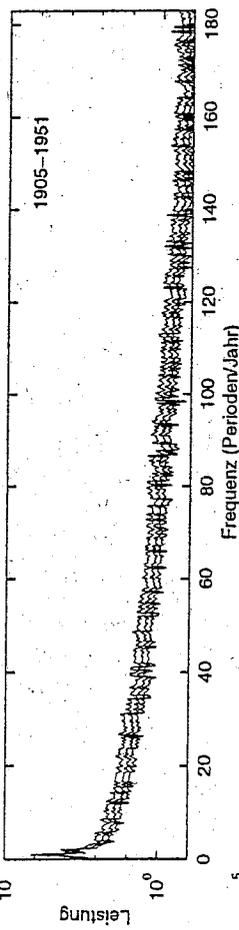
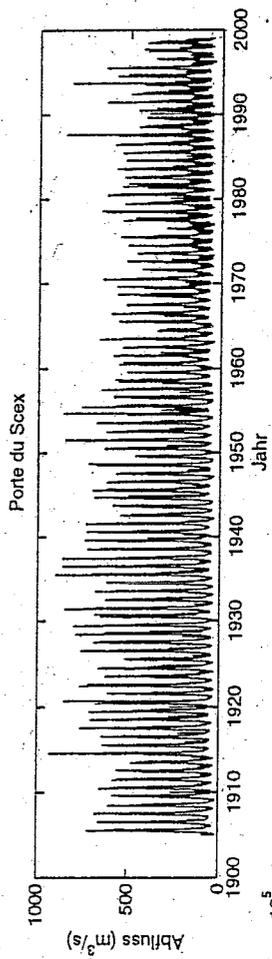
k	A _k	f _k	φ _k · 180°
1	1	6	20
2	2	7	53
3	1	8	35
4	3	12	65
5	2	17	40
6	1	23	70

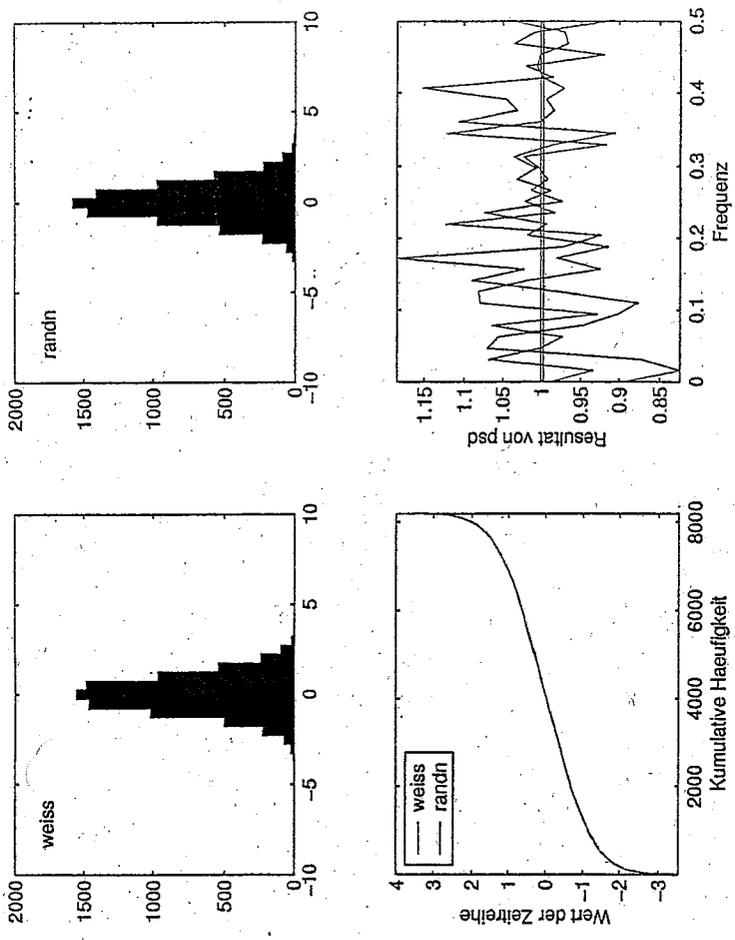
Die total 8192 Werte der Zeitreihe wurden mit einer Abtastfrequenz von 128 Hz berechnet, was einer Gesamtdauer von 64 s entspricht.

- (a) Ein Ausschnitt der Zeitreihe: Rechteck-Fenster, Dauer = 1 s.
- (b) Amplitudenspektrum von (a): FFT mit 128 Punkten berechnet. Die Dauer von (a) ist nicht genügend, damit die Diskrete Fourier Transformation alle wesentlichen Frequenzanteile auflösen kann.
- (c) Amplitudenspektrum von (a): FFT mit 1024 Punkten berechnet. Das Spektrum wird dank der grösseren Anzahl Punkte feiner abgetastet, die erhöhte Auflösung ist aber nur optisch: Nebenmaxima des Spektrums des Rechteck-Fensters unterscheiden.
- (d) Amplitudenspektrum eines 2 s langen Ausschnittes mit Rechteck-Fenster: FFT mit 256 Punkten berechnet. Da hier die verwendete Informationsmenge grösser ist, ist die Auflösung tatsächlich besser: Die Frequenz bei 23 Hz tritt jetzt deutlich aus dem Hintergrund-Rauschen hervor; um entscheiden zu können, ob die Spitzen bei 6 und 8 Hz keine Nebenmaxima des Rechteck-Fensters sind, genügt die Fensterlänge von 2 s jedoch noch nicht.
- (e) Der 2 s lange Ausschnitt der Zeitreihe, aber mit einem Cosinus-Fenster multipliziert (auch Hann- oder Hanning-Fenster genannt).
- (f) Amplitudenspektrum von (e): FFT mit 256 Punkten berechnet. Da die Fourier Transformation des Cosinus-Fensters wesentlich kleinere und schneller abnehmende Nebenmaxima als diejenige des Rechteck-Fensters aufweist, wird das sogenannte *leakage* Problem entschärft. Weil gleichzeitig aber auch das Hauptmaximum des Cosinus-Fensters doppelt so breit ist wie dasjenige des Rechteckes, geht dies auf Kosten der Auflösung: die Frequenzanteile bei 6 und 8 Hz sind nicht mehr sichtbar. Beachte, dass auch der Betrag des Amplitudenspektrums entsprechend der geringeren Fläche des Cosinus-Fensters reduziert ist.







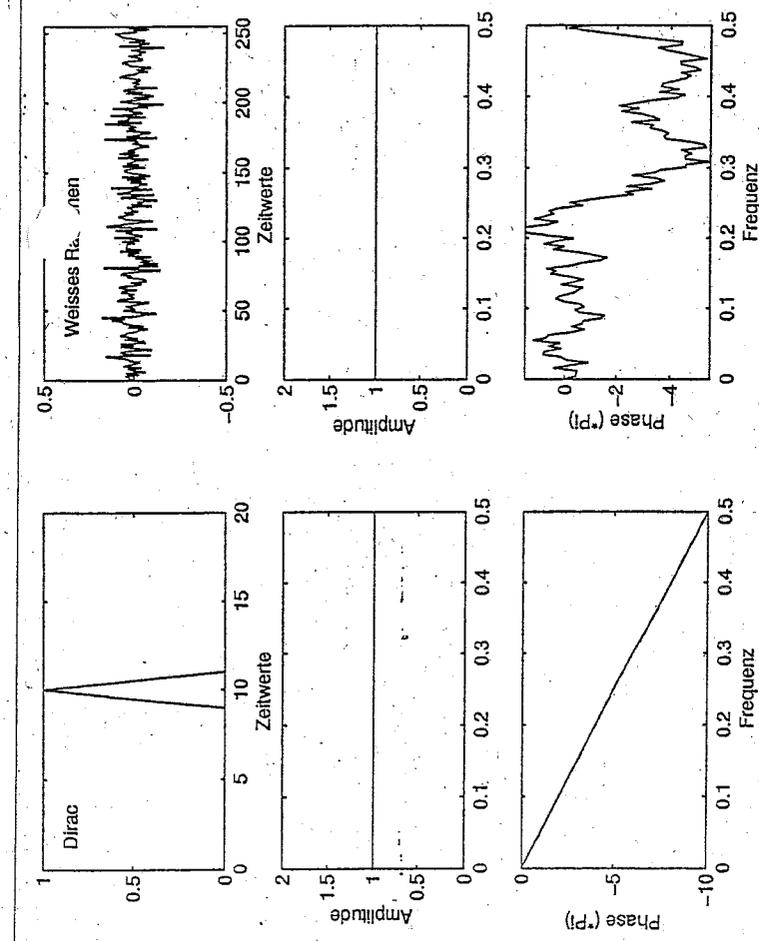


"White noise" als Zufallszahlen

Weisses Rauschen kann auch einfach als eine zufällige Zahlenfolge betrachtet werden. In Matlab kann man eine normalverteilte Zahlenfolge mit Mittelwert = 0 und Varianz = 1 mit der Routine randn generieren. Der Vergleich der entsprechenden Häufigkeitsverteilungen sowie der kumulativen Häufigkeiten zeigt, dass auch das mittels der Routine weiss im Frequenzbereich generierte weisse Rauschen einer normalverteilten Zahlenfolge mit der Varianz = 1 entspricht.

Letzteres ist unten rechts gezeigt: der Mittelwert des Resultates der Routine psd ueber alle Frequenzen ist fuer beide Zeitreihen nahezu gleich 1. Tatsaechlich folgt aus der Skalierung der Routine psd und dem Theorem von Parseval, dass der Mittelwert des Resultates von psd gleich der Varianz der entsprechenden Zeitreihe ist: $\text{var}(x) = \text{mean}(\text{psd}(x, \dots))$.

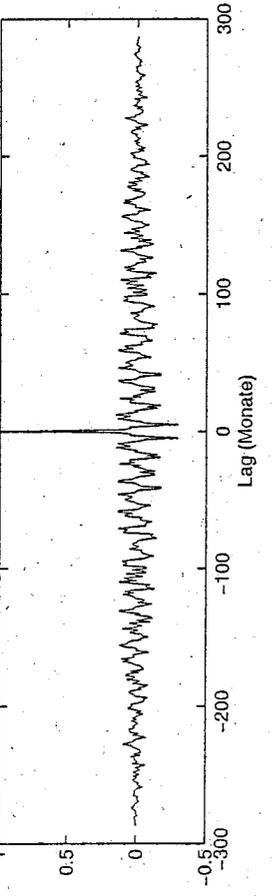
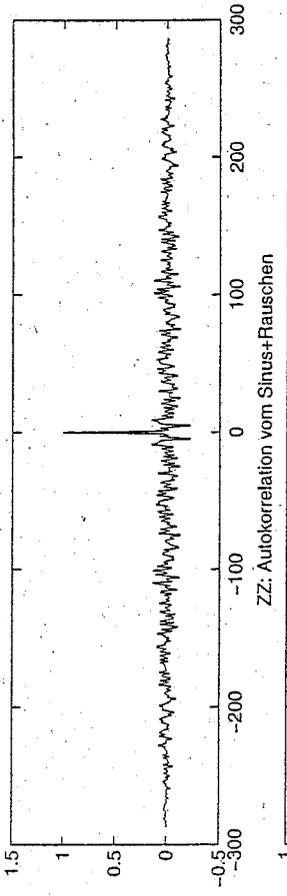
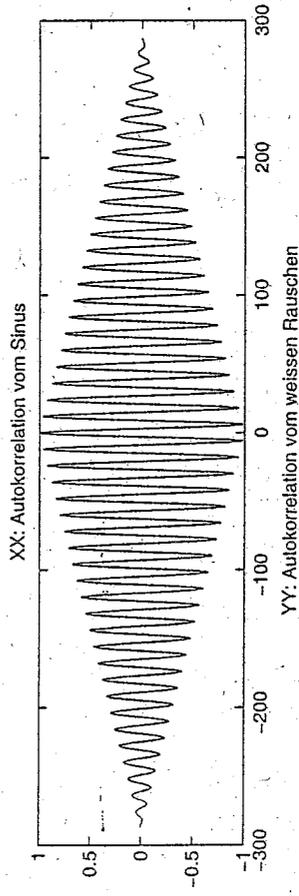
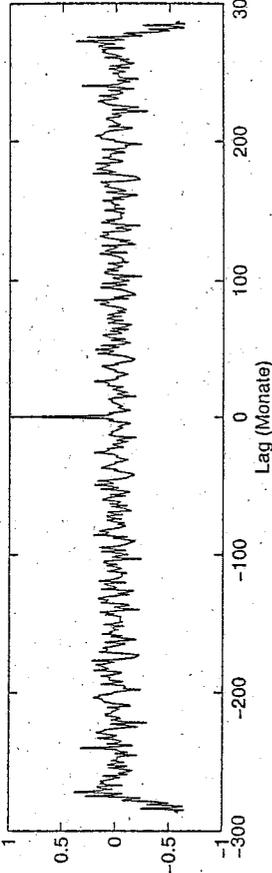
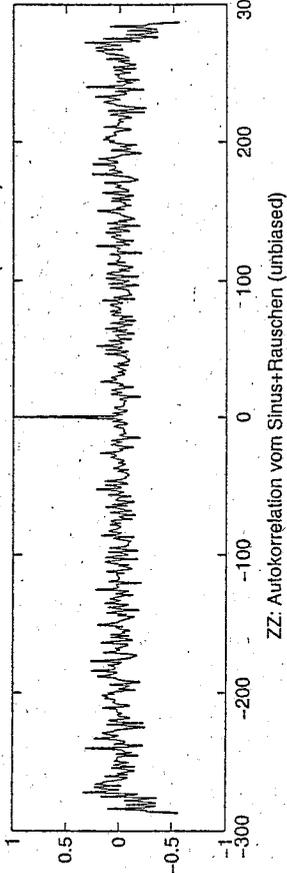
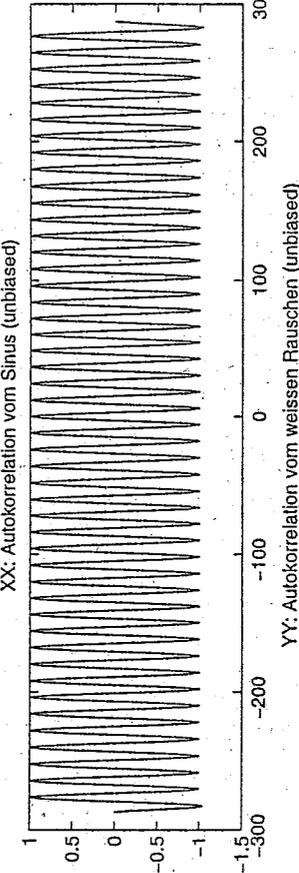
In diesem Beispiel wurde das weisse Rauschen mit 8192 Werten generiert; Zur Berechnung von psd wurde jedoch ueber 128 Segmenten mit einer Laenge von nur 64 Werten gemittelt. Aus diesem Grunde ergibt die Routine psd auch fuer das mit der Routine weiss im Frequenzbereich generierte Rauschen nicht den erwarteten konstanten Wert von 1 fuer alle Frequenzen. Die ist ein Beispiel dafuer, dass man von einer beschraenkten Datenmaenge nur eine Schaeetzung des wahren Leistungsspektrums erhaelt.

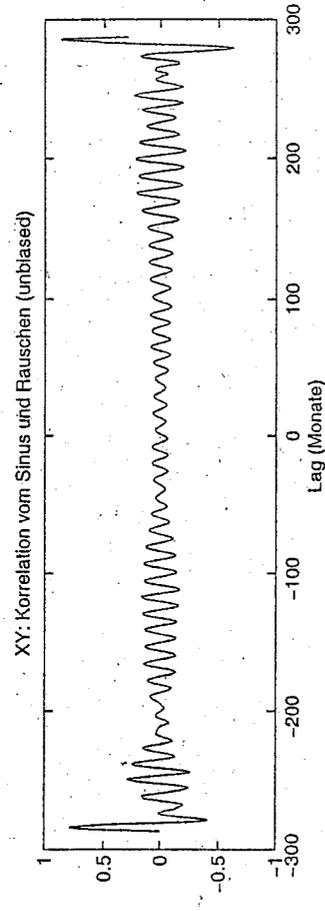
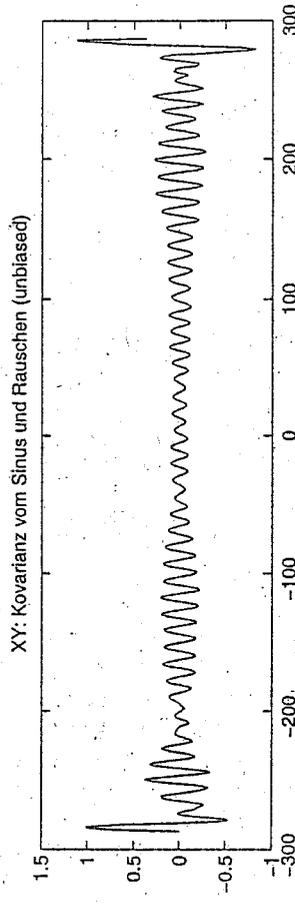
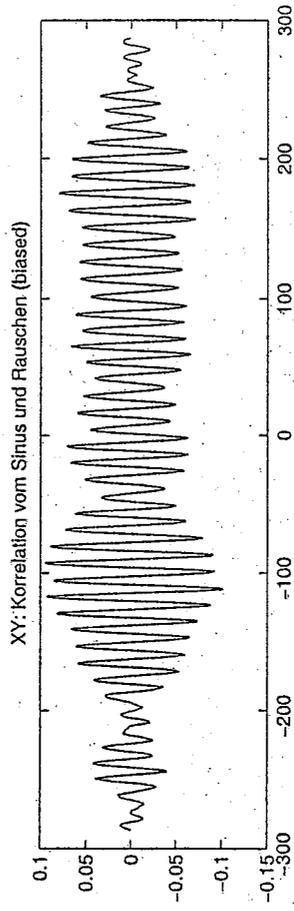
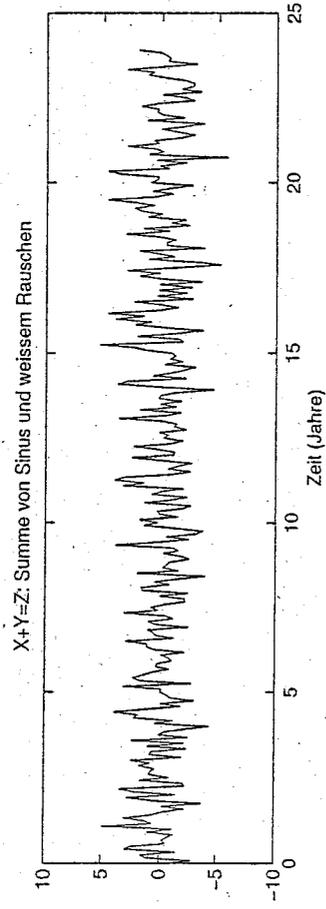
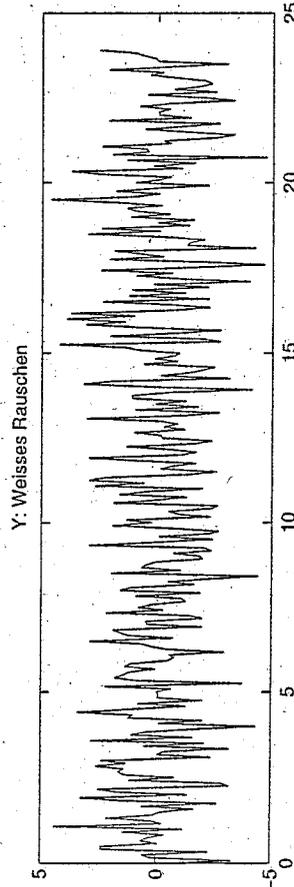
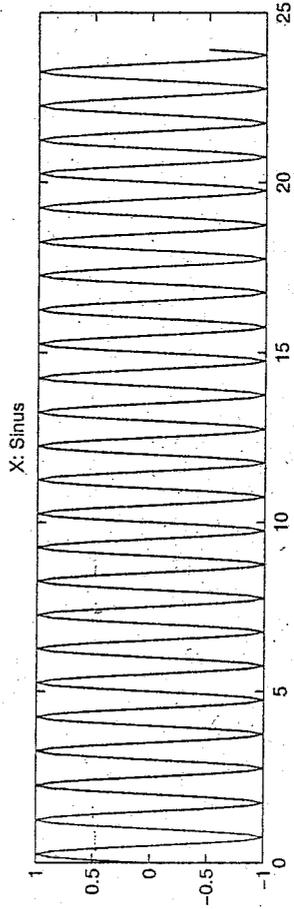


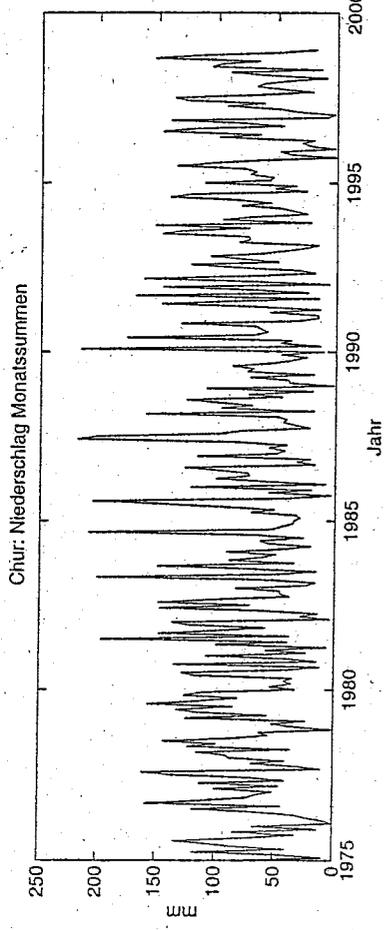
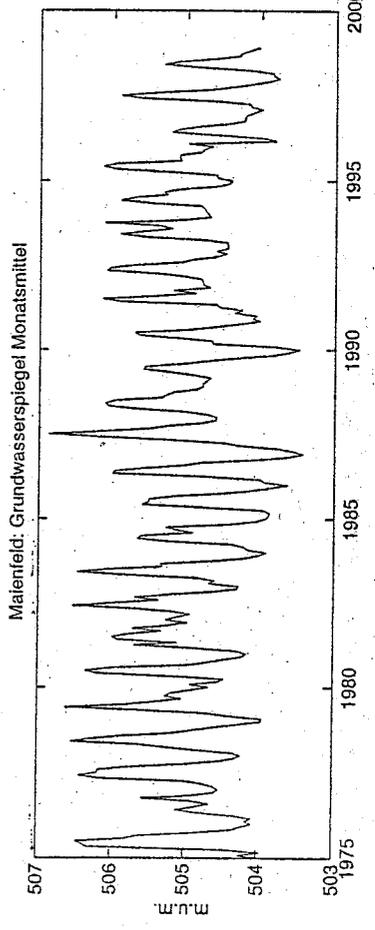
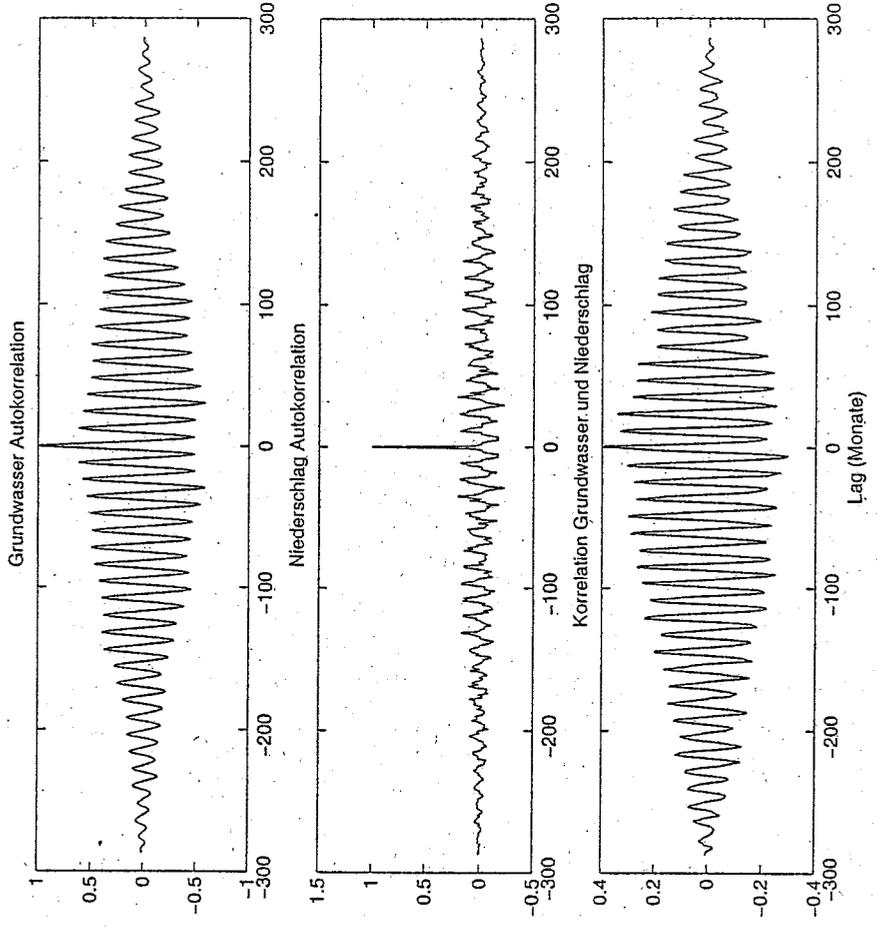
Fourier Trans. von einem delta-Impuls und von "white noise"

Die zwei Zeitreihen wurden mit 256 Werten und einer Abtastrate von 1 generiert. Vom delta-Impuls sind nur die ersten 20 Werte gezeichnet. Wegen der Abtastrate von 1, hat die Nyquist Frequenz den Wert 0.5. Da die Flaeche unter dem delta-Impuls gleich 1 ist, ist auch das Amplitudenspektrum gleich 1 fuer alle Frequenzen (s. Abschnitt ueber die F.T. einer Rechteckfunktion im Skript). Das weisse Rauschen (white noise) wurde mit der Routine weiss im Frequenzbereich generiert und dann mittels inverser Fourier Transformation in eine Zeitreihe umgewandelt: Das Amplitudenspektrum wurde fuer alle Frequenzen auf 1 fixiert (daher "weiss" -- alle Frequenzen sind gleich vertreten, wie im weissen Licht) und die Phase durch einen Zufallszahlen-Generator berechnet (daher Rauschen -- die Phasenlage ist so regellos wie moeglich). Folglich sehen verschiedene Realisierungen einer Zeitreihe von weissem Rauschen sowie die entsprechenden Phasenspektren immer verschieden aus.

Stuende der delta-Impuls an der Stelle $t=0$ statt bei $t=10$, waere die Phase fuer alle Frequenzen gleich 0 (das folgt daraus, dass es sich dann um eine gerade Funktion handeln wuerde, fuer welche der Imaginaerteil verschwindet). In unserem Fall ist wegen der zeitlichen Verschiebung um 10 Zeitwerte die Phase des delta-Impulses gleich $-2\pi \cdot 10 \cdot f$ (s. im Skript den Satz ueber die F.T. einer zeitlich verschobenen Funktion).







Autokorrelation von MA- und AR-Prozessen

Wir nehmen im Folgenden an, dass w_t ein stationärer White-Noise-Prozess mit Mittelwert 0 und Varianz σ_w^2 ist.

Aufgabe 1

Der MA-Operator hat die Koeffizienten $a_0 = 0.5$ und $a_1 = 0.5$. Also handelt es sich um ein MA(1) Prozess mit

$$x_t = a_0 w_t + a_1 w_{t-1}$$

Der AR-Operator hat die Koeffizienten $a_0 = 0.6$, $b_0 = 1$ und $b_1 = -0.4$. Also handelt es sich um ein AR(1) Prozess mit

$$b_0 x_t = a_0 w_t - b_1 x_{t-1}$$

In beiden Fällen haben die Operatoren eine glättende Wirkung auf das weisse Rauschen. Das sieht man auch in den Amplitudenspektren: die hohen Frequenzen sind gegenüber demjenigen im White-Noise-Spektrum gedämpft (*Tiefpass-Filter*).

Die Autokorrelation des White-Noise ist wie erwartet nur bei $\text{lag} = 0$ ungleich 0. Die Autokorrelation des MA(1) Prozesses ist auch bei $\text{lag} = 1$ ungleich 0 und wird erst ab $\text{lag} = 2$ wieder 0. Hingegen scheint die Autokorrelation des AR(1) Prozesses nur asymptotisch gegen 0 zu streben.

Aufgabe 2

Autokovarianz eines MA(1) Prozesses

Sei $a_0 = 1$ und $a_1 = a$, dann ist $x_t = w_t + a w_{t-1}$ und

$$\gamma(0) = \text{Cov}(x_t, x_t) = E[(w_t + a w_{t-1})(w_t + a w_{t-1})]$$

$$\gamma(0) = E(w_t w_t) + 2a E(w_t w_{t-1}) + a^2 E(w_{t-1} w_{t-1}) = \sigma_w^2 + 0 + a^2 \sigma_w^2 = (1 + a^2) \sigma_w^2$$

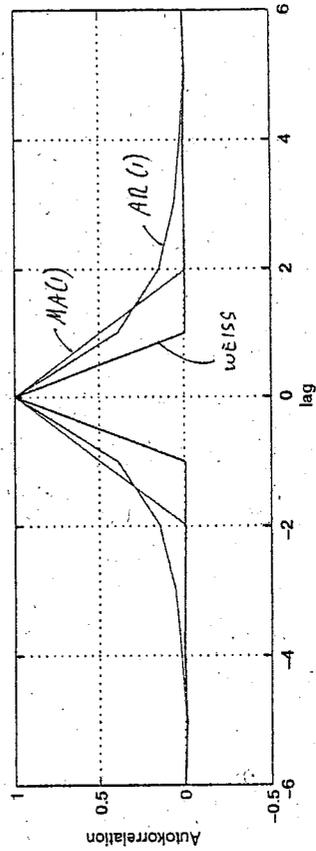
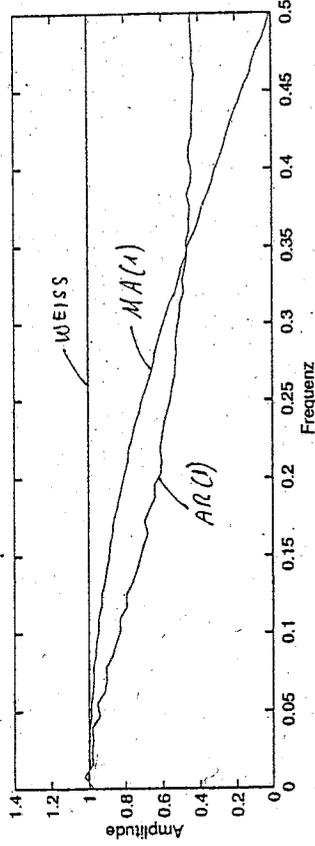
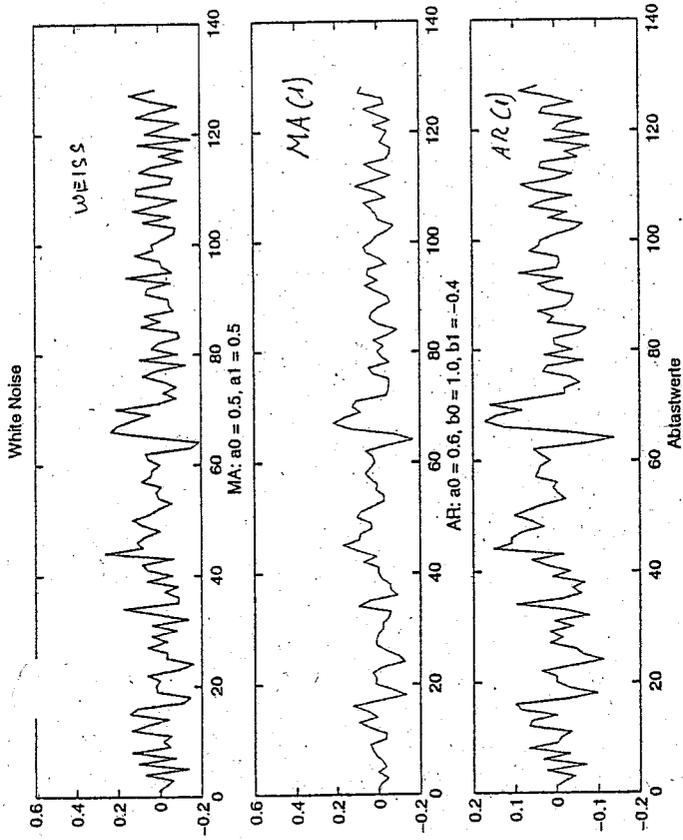
Und analog folgt

$$\gamma(1) = \text{Cov}(x_t, x_{t-1}) = E[(w_t + a w_{t-1})(w_{t-1} + a w_{t-2})] = a \sigma_w^2$$

und

$$\gamma(2) = \text{Cov}(x_t, x_{t-2}) = E[(w_t + a w_{t-1})(w_{t-2} + a w_{t-3})] = 0$$

Der MA(1) Operator hat eine Abhängigkeit zwischen zwei aufeinanderfolgenden Werten der Zeitreihe x_t hergestellt. Genau diese Abhängigkeit zeigt sich in der Tatsache, dass,



im Gegensatz zum vollkommen unkorrelierten weissen Rauschen, die Kovarianzfunktion eine endliche Breite aufweist ($\gamma(1) \neq 0$). Aber, da sich die Abhängigkeit von x_t nur auf den nächsten Wert beschränkt, ist schon $\gamma(2) = 0$. Dieses Resultat lässt sich für beliebige $MA(q)$ Prozesse verallgemeinern:

$$\begin{aligned} \gamma(h) &\geq 0, & |h| \leq q \\ \gamma(h) &= 0, & |h| > q \end{aligned}$$

Als Beispiel sei ein $MA(2)$ Prozess mit den Koeffizienten $a_0 = 0.33$, $a_1 = 0.33$ und $a_2 = 0.33$ gegeben. Die entsprechende Autokovarianzfunktion wird erst ab $\text{lag} = 3$ wieder 0 und der Glättungseffekt ist auch stärker.

Autokovarianz eines $AR(1)$ Prozesses

Sei $a_0 = 1$, $b_0 = 1$ und $b_1 = -b$, mit $|\beta| < 1$, dann ist

$$x_t = bx_{t-1} + w_t = b(bx_{t-2} + w_{t-1}) + w_t = b^2x_{t-2} + bw_{t-1} + w_t$$

und

$$x_t = b^k x_{t-k} + \sum_{j=0}^{k-1} b^j w_{t-j} = \sum_{j=0}^{\infty} b^j w_{t-j}$$

Wir sehen also, dass der $AR(1)$ Prozess eine im Prinzip unendliche Abhängigkeit zwischen den Werten von x_t herstellt, was dazu führt, dass die Autokovarianzfunktion eines AR Prozesses nur asymptotisch gegen 0 geht. Insbesondere gilt für den $AR(1)$ Prozess:

$$\gamma(2) = E(x_t x_{t-2}) = E[(bx_{t-1} + w_t)(x_{t-2})] = E[(b^2x_{t-2} + bw_{t-1} + w_t)(x_{t-2})]$$

Das ergibt

$$\gamma(2) = b^2 E(x_{t-2} x_{t-2}) = b^2 \gamma(0)$$

Für $|\beta| < 1$, lässt sich aus diesem Resultat die Autokorrelationsfunktion eines beliebigen $AR(1)$ Prozesses angeben:

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)} = b^h, \quad h \geq 0.$$

