Stochastische Signale Praktikum

1. Allgemeine Befehle

1.1. Standardbefehle

Befehl	Funktion
save(*filename, variable*)	speichert *variable* in matfile
load(*filename*)	lädt Variable aus matfiel
clear *variable*	löscht *variable*
clear all	löscht alle Variablen im Workspace
clc	löscht Inhalt des Kommandofensters
doc *expression*	Hilfedatei zu *expression*
help *expression*	Kurzhilfe zu *expression*

1.2. Allgemeine Rechenoperationen

Befehl	Funktion
mod(x,y)	x modulo y (immer positiv)
rem(x,y)	x modulo y (vorzeichenabhängig)
sqrt(x)	$\sqrt{x}$
exp(x)	e^x
log(x)	Natürlicher Logarithmus $\ln(x)$
floor(x)	Abrunden auf Integer
ceil(x)	Aufrunden auf Integer
sum(x)	Summe über Werte des Vektors x
prod(x)	Produkt über Werte des Vektors x
min(x)	kleinster Wert des Vektors x
max(x)	größter Wert des Vektors x
all(x)	1 für keine 0 in Vektor x
any(x)	1 für eine Nicht-0 in Vektor x
mean(x)	Mittelwert des Vektors x
sign(x)	Vorzeichen von x (0 wenn x=0)
abs(x)	Betrag von x

1.3. Operatoren

Operator	Funktion
a>b	Gibt logisch 1 zurück wenn $a > b$, sonst 0
a>=b	Gibt logisch 1 zurück wenn $a \geq b$, sonst 0
a==b	Gibt logisch 1 zurück wenn $a = b$, sonst 0
a~=b	Gibt logisch 1 zurück wenn $a \neq b$, sonst 0
a*b	Matrix-Matrix-Multiplikation von a und b
a.*b	Elementenweise Multiplikation von a und b
a(a>=0)	Gibt Teilvektor von a zurück, wo der Wert ≥ 0

1.4. Komplexe Zahlen

Befehl	Funktion
complex(a,b)	$a + jb$
real(z)	Realteil von z
imag(z)	Imaginärteil von z
abs(z)	Betrag/Komplexe Amplitude von z
angle(z)	Phase von z
conj(z)	konjugiert komplex von z

1.5. Trigonometrische Funktionen

Befehl	Funktion
sin(x) , cos(x), tan(x)	x in Bogenmaß
sind(x), cosd(x), tand(x)	x in Grad
asin(x), acos(x), atan(x)	Arcusfunktionen (Rad)
asind(x), acosd(x), antans(x)	Arcusfunktionen (Grad)

2. Matrizenrechnung

2.1. Rechenoperationen

Befehl	Funktion
[a b c]	Zeilenvektor (bzw. vertikales aneinanderreihen)
[a; b; c]	Spaltenvektor (bzw. horizontales aneinanderreihen)
[a b c; d e f; g h i]	3x3-Matrix
[a b] = size($\boldsymbol{A}$)	Dimensionen der Matrix (a Zeilen, b Spalten)
inv($\boldsymbol{A}$)	inverse Matrix von $\boldsymbol{A}$
$\boldsymbol{A}$'	$\boldsymbol{A}^\top$ (transponiert konjugiert komplex)
$\boldsymbol{A}$.'	$\boldsymbol{A}^\top$ (transponiert)
$\boldsymbol{A} \setminus \vec{b}$	löst $\boldsymbol{A}\vec{x} = \vec{b}$
$\boldsymbol{A}$(m,n)	Element $\boldsymbol{A}_{m,n}$
$\boldsymbol{A}$(m,:)	m-te Zeile
$\boldsymbol{A}$(:,n)	n-te Spalte
$\boldsymbol{A}$(Bedingung)	Alle Element in A, auf die die Bedingung zutrifft
find($\boldsymbol{A}$)	lokalisiert Nicht-Null-Elemente (Indizes)
det($\boldsymbol{A}$)	Determinante von A
rank($\boldsymbol{A}$)	Rang (Anzahl unabhängiger von A
[V,D]= eig($\boldsymbol{A}$)	Eigenwerte (D) und Eigenvektoren (V) von A
(a:b:c)	Vektor von a bis c mit Schrittweite b
linspace(a,b,n)	n Werte im gleichen Abstand von a bis b
norm(x)	eukl. Norm des Vektors x
sum($\boldsymbol{A}$)	Summer der Werte über Spaltenwerte
sum($\boldsymbol{A}$,2)	Summer der Werte über Zeilenwerte
mean($\boldsymbol{A}$)	Mittelwert der Spaltenwerte
mean($\boldsymbol{A}$,2)	Mittelwert der Zeilenwerte
numel($\boldsymbol{A}$)	Anzahl der Elemente in $\boldsymbol{A}$

Komponentenweises Rechnen durch einen Punkt vor einem Operator
Bsp: $\boldsymbol{A}$.^2 quadriert jedes Element der Matrix $\boldsymbol{A}$
Inlinefunktion: @(x)(f(x))

2.2. Spezielle Matrizen

Befehl	Funktion
eye(m,n)	$m \times n$ Einheitsmatrix
zeros(m,n)	$m \times n$ 0-Matrix
ones(m,n)	$m \times n$ 1-Matrix
diag($\vec{x}$)	Diagonalmatrix mit den Werten von $\vec{x}$
rand(m,n)	$m \times n$ Zufallsmatrix (Werte: 0-1)
randi(imax,m,n)	integer Zufallsmatrix mit max. imax
magic(n)	$n \times n$ magisches Quadrat

3. Schleiflab

while:	for:
while expression	*for* i=0:1:20
statements	statements
end	*end*

Schleife vorzeitig verlassen mit break

4. Plotten

Befehl	Funktion
plot(x,y,'prop')	Plottet x und y mit Farbe/Symbol 'prop'
stem(y)	Plottet diskrete y Werte (nicht verbunden)
bar(x,y)	Plottet x und y in einem Balkendiagramm
xlim([a b])	Begrenzt Bereich der x-Achse auf [a,b]
ylim([c d])	Begrenzt Bereich der y-Achse auf [c,d]
axis([a b c d])	Kombination aus xlim + ylim
xlabel('Name')	Benennt x-Achse
ylabel('Name')	Benennt y-Achse
title('Titel')	Tituliert den Plot
subplot(H,B,Position)	Selektiert Plot in Figure mit H x B Plots

Beispiel:

```
figure(1);              % new figure
clf;                    % clear old figures
plot(x, y, 'k');        % plot y(x) in black 'k'
hold on;                % more plots in same figure
plot(x, z, 'ro')        % plot z(x) in red circles
legend('y', 'z')        % names of plots
hold off;
```

5. Stochastische Zufallsvariablen

5.1. Realisierung von Standardmodellen

Befehl	$m \times n$-Realisierung einer ...
rand(m,n)	gleichverteile ZV (Werte: 0-1)
randn(m,n)	Standardnormalverteilung $\mathcal{N}(0,1)$
gamrnd(k,t,m,n)	gammaverteile ZV, shape k, scale t
binornd(n,p,m,n)	Binomialverteilung mit Parameter n, p
binornd(1,p,m,n)	Bernoulliverteilung mit Wahrscheinlichkeit p
geornd(p,m,n)	Geometrische Verteilung mit Wahrsch. p
exprnd(1/lambda,m,n)	Exponentialverteilung mit Parameter λ

Beispiele:

Befehl	Ergebnis
2*rand+1	gleichverteile ZV im Bereich [1,3]
plot(y,unifpdf((y-1)/2)/2)	plottet PDF der oberen ZV
plot(x,unifpdf(2*x)*2)	PDF einer gv. ZV im Bereich [0,0.5]
sigma*randn+mu	Realisierung der Normalv. $\mathcal{N}(\mu,\sigma^2)$

5.2. Wahrscheinlichkeitsdichtefunktion (PDF) $f_{\mathsf{X}}(x)$

Befehl	PDF an den Stellen x einer ...
unifpdf(x,a,b)	Gleichverteilung im Intervall [a,b]
poisspdf(x,lambda)	Poissonverteilung mit Parameter λ
normpdf(x,mu,sigma)	Normalverteilung mit Parameter mu und sigma
gampdf(x,k,t)	Gammaverteilung mit shape k und scale t

5.3. Kommulative Verteilungsfunktion (CDF) $F_{\mathsf{X}}(x)$

Befehl	CDF an den Stellen x einer ...
unifcdf(x,a,b)	Gleichverteilung im Intervall [a,b]
poisscdf(X,lambda)	Poissonverteilung mit Parameter λ
normcdf(X,mu,sigma)	Normalverteilung mit Parameter μ und σ
cdfplot(X)	Schätzt und plottet CDF von X

6. Analyse von Zufallsvariablen

6.1. Schätzung von Paramtern einer Zufallsvariable X

x ist ein Vektor von Realisierungen von X

Befehl	Funktion
mean(x)	Schätzt Erwartungswert μ der ZV X
var(x)	Schätzt Varianz Var(X) der ZV X
std(x)	Schätzt die Standardabweichung σ der ZV X
length(x)	Anzahl der Realisierungen der ZV X
mean(x.^3)	Schätzung des 3. Moments der ZV X

6.2. Histogramm

Befehl	Funktion
hist(A)	Teilt A in 10 gleiche Bereiche (Bins) und zählt die Elemente im jeweiligen Bin
hist(A,b)	Teilt A in b gleiche Bereiche (Bins) und zählt die Elemente im jeweiligen Bin
hist(A,centers)	Zählt die Elemente in den Bins um den Einträgen in centers
[n,centers]=hist(...)	Gibt in Anzahl der Elemente in n zurück und die Position der Bins in centers
bar(hist(...))	Erstellt ein Balkendiagramm des Histogramms

7. Signalverarbeitung

7.1. Audiosignalverarbeitung

Befehl	Funktion
[a,b]=audioread('....')	Liest Audiodatei in Vektor a ein (Abtastrate b)
sound(x,b)	Spielt das Signal in Vektor x (Abtastrate b) ab
buffer(x,n)	Teilt Signalvektor x in nichtüberlappende Frames der Länge n
buffer(x,n,p)	Teilt Signalvektor x in Frames der Länge n die sich um p überlappen

Beispiel:

```
buffer(1:12,5,3)
%liefert ...
ans =
     0     0     2     4     6     8
     0     1     3     5     7     9
     0     2     4     6     8    10
     1     3     5     7     9    11
     2     4     6     8    10    12
```

7.1.1 Lineare Systeme

Befehl	Funktion
conv(x,y)	Faltung zwischen Vektor x und y

conv(x,y) gibt einen Vektor der Länge length(x) + length(y) - 1 zurück

8. Begriffe

i.i.d.
Independent and identically distributed (unabhängig und gleichverteilt)

Ergodisch
Eine Zufallsfolge heißt ergodisch, wenn Mittelwerte über die Zeit (für eine einzelne gegebene Realisierung der Folge) zum gleichen Ergebnis führen wie Erwartungswertbildung (für einen einzelnen Zeitpunkt).

Klingonisch
Die klingonische Sprache ist die von Klingonen gesprochen Sprache. Sie ist im gesamten Klingonischen Imperium verbreitet.

9. Kapitel 1

9.1. Histogramm einer gleichverteilten ZV plotten

```
function x=hist_rand(N)
% Vektor x mit N Realisierungen von X (Gleichverteilung)
x=rand(N,1);
centers=0+1/40:1/20:1-1/40;
counts=hist(x,centers);
% Normierung von counts fuer eine PDF
counts = counts/N*20;
bar(centers,counts);
xlabel('x')
ylabel('h(x)')
title(sprintf('N = %01d',N));
```

10. Kapitel 2: Quantisierung und Transformation von ZV

10.1. Beispielaufgabe

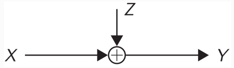
Gegeben sei eine Zufallsvariable X, die im Intervall $[0, 1]$ gleichverteilt ist und eine Zufallsvariable $Y = g(X)$, deren WDF mit der folgenden Funktion ausgewertet werden kann:

```
function f = mypdf(y)
f=unifpdf(exp(y)).*exp(y);
```

Folgerungen:
Die Funktion $g = g(x) = ln(x)$
1000 Realisierungen von Y erzeugen: y=log(rand(1000,1));

11. Kapitel 3: Bedingte Verteilung

11.1. Histogramm eines AWGN-Kanals



X Eingangssignal
$Z = \mathcal{N}(0, \sigma^2)$ weißes Rauschen
Y Ausgangssignal

```
function hist_out(N,p,sigma)
binwidth=0.025;
centers=-2+binwidth/2:binwidth:3-binwidth/2;
figure
subplot(311)

% X 0 oder 1, Y = X + normalverteiltes Rauschen
x=binornd(1,p,N,1);
y=awgn_channel(x,sigma);

counts=hist(y,centers);
bar(centers,counts/(binwidth*sum(counts)),1);
ylim([0 2])
xlabel('y')
ylabel('h(y)')
title(sprintf('N=%01d',N))

subplot(312)
% Fall X ist immer 1
x=ones(N,1);
y=awgn_channel(x,sigma);

counts=hist(y,centers);
bar(centers,counts/(binwidth*sum(counts)),1);
ylim([0 2])
xlabel('y')
ylabel('h(y|x=1)')

subplot(313)
% Fall X ist immer 0
x=zeros(N,1);
y=awgn_channel(x,sigma);

counts=hist(y,centers);
bar(centers,counts/(binwidth*sum(counts)),1);
ylim([0 2])
xlabel('y')
ylabel('h(y|x=0)')
```

11.2. Maximum-Likelihood-Detektion

$$\max_{\hat{x}\in\{0,1\}} f_{Y\,|\,X}(y|\hat{x})$$

In unserem Fall: $\hat{x} = \begin{cases} 0 & y \leq \frac{1}{2} \\ 1 & y > \frac{1}{2} \end{cases}$ Nachteil: Ignoriert das Wissen über die Eingangsverteilung.

```
function xhat = ml_detector(y)
xhat=(y>0.5);
```

Nachteil: Ignoriert Wissen über die Eingangsverteilung

11.3. Maximum-A-Posteriori-Detektion

$$\max_{\hat{x}\in\{0,1\}} p_{X\,|\,Y}(\hat{x}|y)$$

mit

$$p_{X\,|\,Y}(x|y) = \left.\begin{cases} \frac{p f_Z(y-1)}{f_Y(y)} & x = 1 \\ \frac{(1-p) f_Z(y)}{f_Y(y)} & x = 0 \\ 0 & \text{sonst} \end{cases}\right\} = \frac{f_{Y\,|\,X}(y|x) p_X(x)}{f_Y(y)}$$

```
function xhat = map_detector(y,p,sigma)
xhat=(p*normpdf(y-1,0,sigma)>(1-p)*normpdf(y,0,sigma));
```

Nachteil: Die Verteilung der Zufallsvariable am Eingang muss bekannt sein

Spezialfall der diskreten Gleichverteilung $p_X(x) = \frac{1}{|\Omega_X|} \forall x \in \Omega_X$:
p_X für alle x gleich $\rightarrow$ kann aus der Entscheidungsregel gestrichen werden. ML äquivalent zu MAP

11.4. Bitfehlerwahrscheinlichkeit

Wahrscheinlichkeit, dass der Detektor falsch entschieden hat.

```
p = mean(x ~= xhat);
```

12. Kapitel 4: Standardmodelle, Erwartungswert und Varianz

12.1. Modellierung für Mobilfunknetz

Überlagerung von Nutzern im Hotspot-Bereich (Normalverteilung) X_h, Y_h und anderen Nutzern (Gleichverteilung) X_h, Y_h mit der Bernoulli-verteilten ZV B.

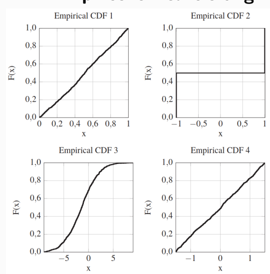
$$X_m = \begin{cases} X_h & \text{wenn } B = 1 \\ X_u & \text{wenn } B = 0 \end{cases} \qquad Y_m = \begin{cases} Y_h & \text{wenn } B = 1 \\ Y_u & \text{wenn } B = 0 \end{cases}$$

```
function M=mixed_positions(N,L,muX,muY,sigma,q)
M=zeros(N,2); % Matrix initialisieren

H=hotspot_positions(N,L,muX,muY,sigma);
U=uniform_positions(N,L);
b=binornd(1,q,N,1);

M(b==1,:)=H(b==1,:);
M(b==0,:)=U(b==0,:);
```

12.2. Empirische Realisierung von CDFs



(1) cdfplot(rand(1000,1));
(2) cdfplot(sign(rand(1000,1)-0.5));
(3) cdfplot(randn(1000,1)*3-1.5);
(4) cdfplot(rand(1000,1)*3-1.5);

13. Kapitel 5: Zufallsfolgen

13.1. Realisierung einer Zufallsfolge

$X_{n+1} = X_n + V_n \qquad Y_{n+1} = Y_n + W_n$
V_n und W_n sind i.i.d. Gleichverteilung mit Intervall $[-\delta; \delta]$
Folgender Code aktualisiert die Positionen (x_i, y_i) im Vektor pos

```
function pos=update_positions(pos,delta)
pos=pos+2*delta*(rand(size(pos))-0.5);
```

14. Kapitel 6: Zufallsfolgen und lineare System

14.1. Signalgenerierung

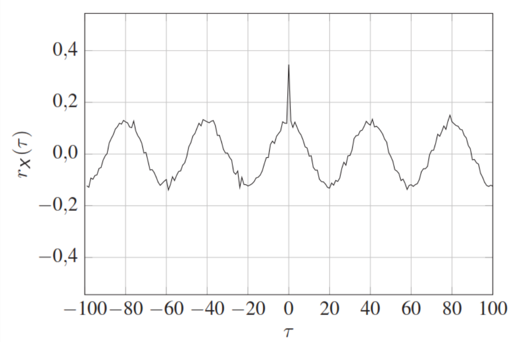
```
%Erstellt Sinussignal der Frequenz f1 mit der Amplitude A1,
%ueberlagert mit weissen Rauschen mit Parameter sigma
function x=create_signal(fS,T,f1,A1,sigma)
t=1:T*fS;
x=A1*sin(2*pi*f1/fS*t) + sigma*randn(1,fS*T);
```

14.2. Geschätzte Autokorrelationsfolge

Gegeben:

$$X_n = A_1 \sin(2\pi \frac{f}{f_s} n + \varphi_0) + Z_n$$

wobei φ_0 im Intervall $[0, 2\pi]$ stetig gleichverteilt ist.



Geschätzte Autokorrelationsfolge von X_n

Folgerung

- Schätzung der Frequenz: $f = \frac{f_S}{40}$
- Der Peak bei $\tau = 0$ ist auf Z_n zurückzuführen
- $\mathrm{E}[Z_k Z_l] = 0$ für $k \neq l$