Kapillarrheologische Detektion von mechanischen Instabilitäten bei der Polymerverarbeitung

Diplomarbeit von
Ingo F. C. Naue

Fachbereich: Maschinenbau
Studienbereich: Angewandte Mechanik
Fachgebiet: Strömungsmechanik/ Mechanik der Polymere,
der Technischen Universität Darmstadt
betreut von
Prof. M. Wilhelm

Ingo F.C. Naue

18. März 2007

Versicherung an Eides Statt

Erklärung zur Diplomarbeit gemäß §23 Abs. 7 APB

Diplomarbeit von ________________________________

Hiermit versichere ich, die vorliegende Diplomarbeit ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den ____________________

____
Unterschrift

Inhaltsverzeichnis

Symbolverzeichnis

Symbol	Einheit	Erklärung	Seite
τ_{xy}	Pa	Schubspannung	6
D_k	mm	Testkanal-/ Testkolbendurchmesser	6
L_k	mm	Länge des Testkanals	6
L_D	mm	Länge der Düse	6
α	Grad	Eintrittswinkel in die Düse	6
d_{Kreis}	mm	Durchmesser derer Vollkreisdüse	7
b	mm	Breite der Schlitzdüse	7
h	mm	Höhe der Schlitzdüse	7
$d_{R,aussen}$	mm	Außendurchmesser der Ringschlitzdüse	7
$d_{R,innen}$	mm	Innendurchmesser der Ringschlitzdüse	7
ϑ	^oC	Temperatur	8
p	bar	Druck	8
D_{Duese}	mm	Charakteristische Abmessung der Düse	9
Re	$-$	Reynoldszahl	10
v	$\frac{m}{s}$	Strömungsgeschwindigkeit	10
p'	$\frac{Pa}{s}$	Druckgradient in Strömungsrichtung	10
ρ	$\frac{kg}{m^3}$	Dichte	10
Δt	s	Zeitliche Auflösung	10
ΔV	m^3	Volumenverschiebung	10
$\dot{V}$	$\frac{m^3}{s}$	Volumenänderungsrate, Volumenfluss	10
A_k	m^2	Querschnittsfläche des Testkanals	10
v_{kolben}	$\frac{mm}{s}$	Geschwindigkeit des Testkolbens	10
$\dot{\gamma}_{ap}$	$\frac{1}{s}$	Scheinbare Scherrate	11
η_{ap}	$Pa \cdot s$	Scheinbare Viskosität	11
τ_{ap}	Pa	Scheinbare Schubspannung	11
Δp	bar	Druckdifferenz über die Testdüse	11
p_E	bar	Druck am Düseneingang	11
p_A	bar	Druck am Düsenausgang	11
$\dot{\gamma}$	$\frac{1}{s}$	Scherrate	11
m	$-$	Fließexponent	12

Symbol	Einheit	Erklärung	Seite
ϕ	$-$	Fluidität	12
η_0	$Pa \cdot s$	Plateau- oder Nullviskosität	13
$\dot{\gamma}_{lim}$	$\frac{1}{s}$	Scherrate der Stabilitätsgrenze	14
$\bar{p}$	bar	Mittlerer Druck	14
ν_{max}	Hz	Maximale Abtastrate	16
ν_{mes}	Hz	Größte Frequenz im Messsignal	16
$\nu_{Nyquist}$	Hz	Nyquistfrequenz	16
ν	Hz	Frequenz	16
ν_{over}	Hz	Frequenz des Oversamplings	19
$f(t), g(t)$	$-$	Beliebige Zeitfunktion	21
$F(\omega)$	$-$	Fouriertransformierte von $f(t)$	21
ω	$\frac{Radien}{s}$	Kreisfrequenz	21
a_n, b_n	$-$	Fourierkoeffizienten	21
A_n, B_n, C_n	$-$	Fourierkoeffizienten	22
T	s	Zeitintervall	21
$A(\omega)$	$-$	Frequenzgang	25
$\varphi(\omega)$	$-$	Phasengang	25
τ	s	Korrelationszeit	29
$S_{FF}(\omega)$	$-$	Spektrale Leistungsdichte von $F(\omega)$	29
$R_{ff}(\tau)$	$-$	Autokorrelationsfunktion von $f(t)$	29
$r_{ff}(\tau)$	$-$	Diskrete Autokorrelationsfunktion	30
$c_{ff}(\tau)$	$-$	Diskrete Autokovarianzfolge	32
$s_{ff}(\tau)$	$-$	Diskrete spektrale Leistungsdichte	34
$\Delta\nu$	Hz	Frequenzschrittweite	34
$f_d(t)$	bar	Polynomapproximation für die Drift	43

Kapitel 1

Einführung

Polymere sind eine Gruppe von Materialien, die in allen industriellen Bereichen eingesetzt werden, z.B. Textilindustrie, Verpackungsmittelindustrie, Fahrzeugbau,... . Eine prozentuale Darstellung des Einsatzes von Kunststoff[1] in Deutschland 2005 zeigt Abb. 1.1 von *PlasticsEurope e.V.*. Der weltweite Verbrauch liegt bei ca. 250 Mio t (2006). Die Verbrauchsquote bei Stan-

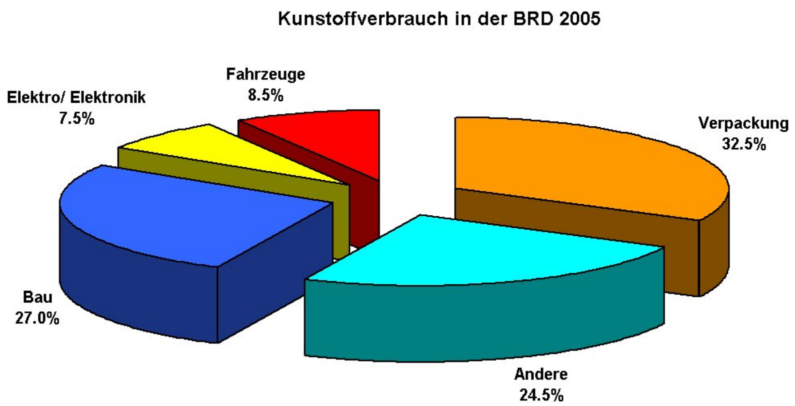


Abbildung 1.1: *Prozentualer Verbrauch an Kunststoff in der BRD 2005. Insgesamt wurden 7,44 Mio. t Standard- und 1,70 Mio. t technische Kunstoffe verbraucht.*

dardkunststoffen ist in 2004/2005 um 2% in der BRD gestiegen. Außerdem ist die Produktion von Kunststoffen seit 1960 stetig gestiegen und besonders die Standardkunststoffe haben einen großen Anteil daran (Abb. 1.2).

In der Industrie werden Polymere mit *Extrudern* (Abb. 1.3) verarbeitet. In diesen Maschinen werden Polymere geschmolzen, gemischt und schließlich durch eine Düse in Form gepresst. Während der Verarbeitung

[1] Kunststoff= Polymer +Additive.

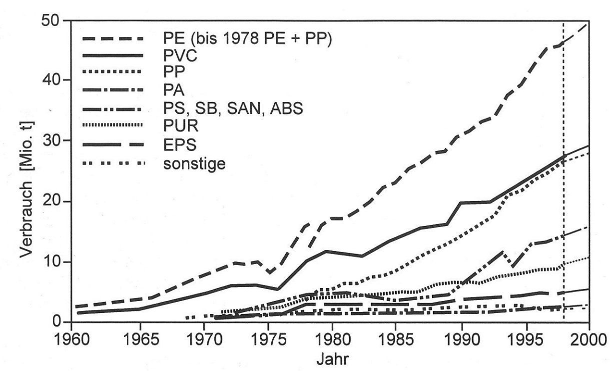


Abbildung 1.2: *Weltweiter Verbrauch von Kunststoffen von 1960 bis 2000 in Mio. t. entnommen aus [12].*

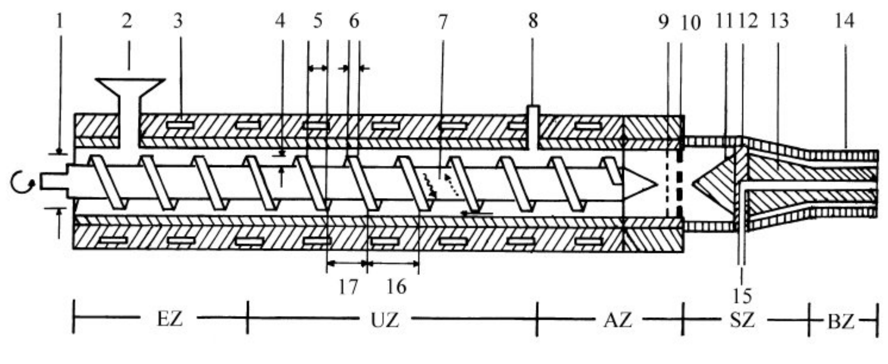


Abbildung 5.79: Schematische Darstellung eines Einschneckenextruders mit Längsspritzkopf
1 Schneckendurchmesser, 2 Einfüllöffnung, 3 Kühl- bzw. Heizelemente, 4 Gangtiefe,
5 Steigungswinkel, 6 Stegbreite, 7 ..... Druck-, -- Schlepp-, — Leckströmung,
8 Entgasungsöffnung, 9 Sieb, 10 Lochscheibe (Brecher), 11 Verdränger, 12 Dornhalter,
13 Dorn, 14 Hülse (Mundstück), 15 Stützluftzufuhr, 16 Steigung, 17 Gangbreite,
EZ Einzugszone; UZ Umwandlungszone (Kompressionszone); AZ Ausstoßzone (Meteringzone);
SZ Schmelzzone; BZ Bügelzone (Profilierungszone)

Abbildung 1.3: *Prinzipskizze eines Extruders entnommen aus [22].*

von Polymeren in Extrudern tritt infolge von zu hohen Verarbeitungsgeschwindigkeiten das Problem der mechanischen Instabilitäten auf. Dies äußert sich zuerst, aber meist unerkannt, in der Düse als Druckfluktuationen. Als sichtbarer Effekt erscheinen die mechanischen Instabilitäten, wenn das Polymer, als *Extrudat*, die Düse verlässt. Seine Oberfläche ist nicht mehr glatt, sondern rauh. In vielen Anwendungen ist dieser Mangel an Oberflächenqualität nicht akzeptabel. Deswegen muss die Bearbeitungsgeschwindigkeit herab gesetzt werden. Damit wird in der gleichen Zeit weniger Produkt hergestellt und das erhöht die Kosten. Für die Verarbeitung von Kunststoffen kann die Industrie auf empirische Werte zurückgreifen, doch diese sind häufig

1. nicht optimiert und

2. für neue Materialien oder Mischungen bekannter Materialien und Additive nur unzureichend bekannt.

Deshalb bietet es sich an die mechanischen Instabilitäteten während der Verarbeitung zu detektieren. Längerfristiges Ziel wäre es einen sich selbst steuernden Extruder ("Smart-Extruder") zu bauen. Dieser würde seine Bearbeitungsgeschwindigkeit an das jeweils zu verarbeitende Polymer anpassen. Ziel dieser Diplomarbeit ist es, erste Vorarbeiten zur Datenerfassung und Analyse an einem Kapillarrheometer durchzuführen.

1.1 Aufbau der Arbeit

Diese Arbeit gliedert sich wie folgt.

In Kap. 1.1 werden die allgemeinen Grundkenntnisse für das Verständnis dieser Arbeit zusammengestellt. Kap. 2.1 stellt die Funktionsweise, technische Umsetzung und nötigen Formeln der Kapilarrheometrie vor. Diesem folgt in Kap. 2.2 eine Beschreibung der Oberflächen Phänomene von extrudierten Polymeren. Kap. 2.3 erklärt die wichtigsten Regeln zur Datenerfassung. Ziel dieser Arbeit ist es die Druckfluktuationen zu charakterisieren. Die Datenanalyse geschah mit Hilfe der *Fourier Transformation* und *Autokorrelationsfunktion*. Den mathematischen Hintergrund präsentiert Kap. 2.4. Der Versuchsaufbau, die Durchführung der Versuche und Analysen wird in Kap. 3 beschrieben. Das Kap. 4 summiert die Mess- und Analyseergebnisse. In Kap. 5 folgt eine kritische Diskussion der Resultate in wieweit ein solcher "Smart-Extruder" prinzipiell möglich ist und welche Einschränkungen sich ergeben.

Kapitel 2

Grundlagen und Theorie

Dieses Kapitel bildet die theoretische Basis zum Verständnis dieser Arbeit. Es wird auf die der Thematik entsprechenden Mechanik, Mathematik und Messtechnik eingegangen. Mit diesen Hilfsmitteln wurden die Ergebnisse aus Kap. 4 zugänglich gemacht.

2.1 Funktionsprinzip von Kapillarrheometern

Kapillarrheometer dienen zur Untersuchung der Eigenschaften von Newtonschen oder nicht Newtonschen Flüssigkeiten. Besonders interessant sind die Fließ-und Viskositätskurven. Diese stellen die scherratenabhängige Viskosität dar. Am häufigsten werden Kapillarrheometer für die Untersuchung von Polymerschmelzen eingesetzt. Die ermittelten Messergebnisse liefern dabei "scheinbare" Werte, welche sich aber über Korrekturgleichungen von *Bagley* [1] und *Weißenberg/ Rabinowitsch* [27] auf die absoluten Werte umrechnen lassen. Der Messbreich des Kapillarrheometers deckt die Lücke des Messwertbereichs zwischen den Rotationsrheometern und den Extrudern ab. Gute und kompakt zusammengefasste Informationen zur Kapillarrheometrie sind in [23] oder in [32] zu finden.

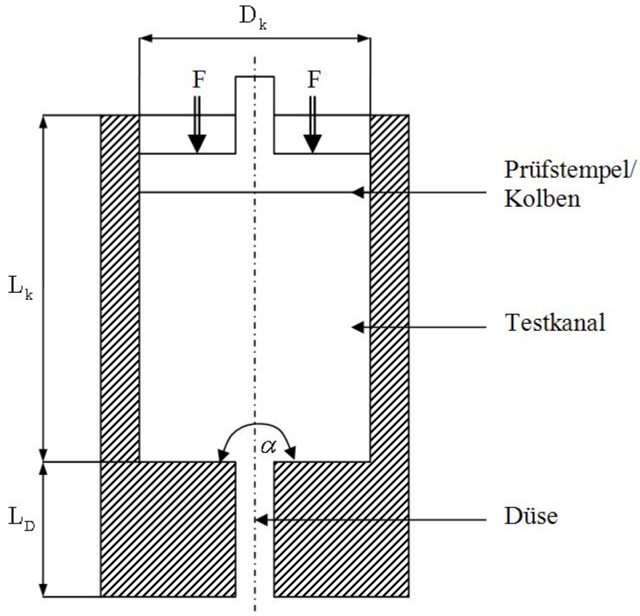


Abbildung 2.1: *Prinzipskizze eines Kapillar-Rheometers mit den charakteristischen Abmessungen.*

2.1.1 Aufbau

Der konkreten Aufbau eines *Kapillarrheometers* besteht aus drei Hauptbaugruppen (Abb. 2.1):

Düse, sie dient zum Erzeugen des zu untersuchenden Strömungsverhaltens bzw. Materialbeanspruchung (Schubspannung τ_{xy} und Scherrate $\dot{\gamma}$). Drei Bauformen werden häufig eingesetzt und sind kommerziell leicht erhältlich

- Kreisdüse (Abb. 2.2.a),
- Schlitzdüse (Abb. 2.2.b) und
- Ringspaltdüse (Abb. 2.2.c).

Testkanal, in ihm wird das Testmaterial bezüglich Druck und Temperatur homogenisiert. D.h. bevor das Material in die Düse gelangt, muss gewährleistet sein, dass an jeder Stelle im Material die selben thermodynamischen Bedingungen (Temperatur ϑ und Druck p) herrschen. Konstruktiv werden Testkanäle als Hohlzylinder ausgeführt. Die Beladung des Zylinders mit dem Probenmaterial erfolgt entweder manuell per Handeinfüllung oder maschinell per Einspritzung mit dem Extruder.

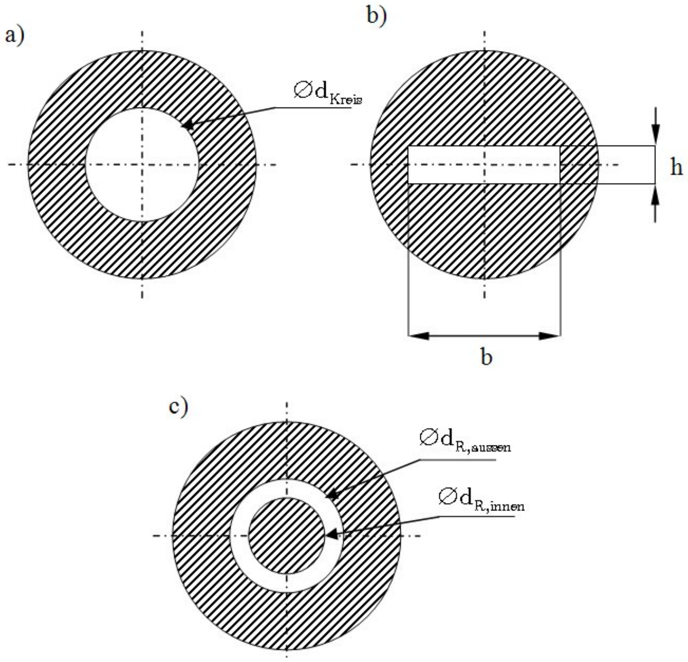


Abbildung 2.2: *Verschiedene Düsenformen: a) Kreisdüse, b) Schlitzdüse und c) Ringspaltdüse.*

Druckerzeugung, damit das Probenmaterial durch die Düse fließt muss Druck im Testkanal erzeugt werden. Dafür gibt es zwei handelsüblich Konstruktionsprinzipien

- Schnecken- oder Spindelantrieb des Prüfkolbens oder
- Gasdruck auf Prüfkolben.

Ein weiterer Bestandteil von Kapillarrheometern ist häufig eine Heizung, um die Düse und den Testkanal auf eine gewünschte Testtemperatur zu bringen.

2.1.2 Versuchsablauf

Der Testkanal wird auf die für die Messung benötigte konstante Temperatur vorgeheizt und das Testmaterial eingefüllt. Nach Erzeugen eines homogenen Zustandes mit einer Temperatur ϑ = konst. und einem Druck p = konst. erreichtist, erfolgt die Messung. Dazu bewegt sich der Kolben nach unten. Zur Erzeugung der Bewegung gibt es zwei prinzipiell unterschiedliche Bauarten:

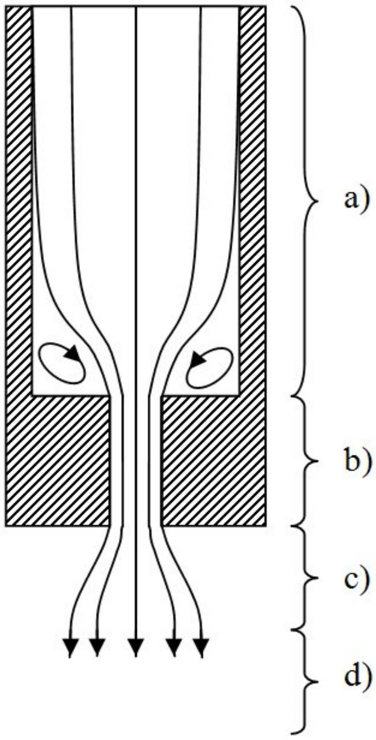


Abbildung 2.3: *Stromlinienverlauf im Kapillarrheometer mit* $\alpha = 180$ ° *(vgl. Abb. 2.1), während der Messung: a) Strömung im Testkanal, b) Strömung in der Düse, c) Strömung am Düsenaustritt und d) Strömung des erstarrten Materials. Auffallend sind die Wirbel am Düseneintritt.*

Konstanter Druck: Der Kolben wird mit einer konstanten Kraft in Richtung Düse gedrückt. Im gesamten Testmaterial entsteht dadurch ein konstanter, voreingestellter Druck.

Konstante Kolbengeschwindigkeit: Der Kolben wird mit einer konstanten definierten Geschwindigkeit in Richtung Düse bewegt. In der Düse wird dadurch eine konstante Scherrate erzeugt.

Dem Versuchsaufbau mit der konstant gewählten Kolbengeschwindigkeit ist ein kontinuierlicher Scherratenbereich zugänglich. Durch die Bewegung des Kolbens wird im Material Druck aufgebaut, wodurch es anfängt zu fließen und durch die Düse gepresst wird. In Abb. 2.3 sind typische Strömungsverhältnisse in dem Kapillarheometer eingezeichnet. Besonders auffallend ist die Wirbelbildung in den Ecken. Diese können in Verbindung mit dem jeweiligen Materialverhalten die Strömung in der Düse stark beeinflussen. In [2] ist beschrieben, wie diese Wirbel in Kombination mit dem viskoelastischen Verhalten von Polyethylen (PE)-Schmelzen[1] die Strömung verändern. Um diesen Effekt zu vermeiden oder zu minimieren muß die Düsenlänge L_D groß genug gewählt sein im Vergleich zur charakteristischen Größe der Düse D_{Duese}. Als Richtlinie gilt $D_{Duese} << L_D$, typischerweise $\frac{L_D}{D_{Duese}} > 10$ Alternativ kann auch der Düsen Eintrittswinkel α verkleinert werden.

[1]Die Rheologie von Polymerschmelzen wird in [21] behandelt.

2.1.3 Auswertung

Die Strömung innerhalb der Geometrie des Kapillarrheometers lässt sich annähernd durch eine *Poiseuille Strömung* [33] beschreiben. Die Poisseuille Strömung gehört zu den laminaren Strömungen. Folgende Annahmen gelten für die Poiseuille Strömung:

1. das Testmaterial ist ein Kontinuum,
2. die *Reynoldszahl*[2] $Re << 1$, d.h. es liegt eine laminare Strömung vor,
3. die Strömung ist vollends ausgebildet, d.h. $\frac{\partial v}{\partial t} = 0$ und $p' = \frac{\partial p}{\partial z} = konst.$,
4. die Strömung ist inkompressibel ($\rho = konst.$, $\frac{\partial \rho}{\partial p} = 0$),
5. die Strömung ist *isotherm* ($\vartheta = konst.$) und
6. es liegt Wandhaftung vor (Abweichungen werden mittels *Weißenberg/ Rabinowitsch* korrigiert).

Abb. 2.4 zeigt typische Geschwindigkeitsprofile in dem Kapillarrheometer. Im Bereich a) und b), also innerhalb des Rheometers, haben die Geschwindigkeitsprofile die typische Parabelform einer Poiseuille Strömung. Nach dem Verlassen des Kanals Abb. 2.4 Bereich c) ist die Schmelze noch flüssig, aber die Wandhaftung fehlt. Damit ist ihre Geschwindigkeit überall von Null verschieden. Das Geschwindigkeitsprofil geht dann mit der Erstarrung der Schmelze in eine Festkörperbewegung über (Abb. 2.4.d). Die sich einstellenden Strömungsprofil wurden in [25] und [31] mittels Laser-Doppler-Velocimetrie untersucht. Die Strömung wird durch die Kolbenbewegung verursacht. Der Kolben verdrängt je inkrementellen Zeitschritt Δt seiner Bewegung ein Volumenelement der Größe ΔV. Die *Volumenänderungsrate* oder der *Volumenfluss* ist $\dot{V} = \frac{dV}{dt}$. Mit Hilfe der *Kontinuitätsgleichung*[3] gilt

$$\dot{V} = A_K \cdot v_{Kolben}. \tag{2.1}$$

Die Querschnittsfläche des zylindrischen Testkanals ist

$$A_k = \frac{D_k^2}{4}\pi. \tag{2.2}$$

Die Kolbengeschwindigkeit v_{Kolben} ist eine vom Experimentator vorgegebene Größe. Damit lässt sich die *scheinbare Scherrate*[4] $\dot{\gamma}_{ap}$ für eine Schlitzdüse

[2] $Re = \frac{v D_{duese} \rho}{\eta}$

[3] Kontinuitätsgleichung: $\frac{d\rho}{dt} = konst.$.

[4] apparent engl. scheinbar

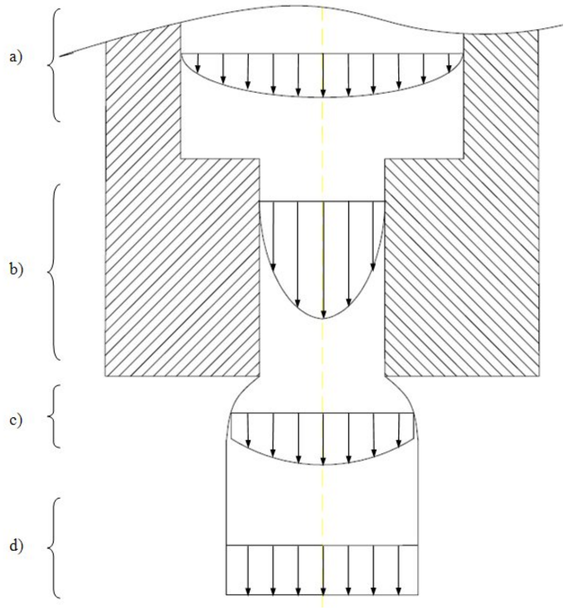


Abbildung 2.4: *Geschwindigkeitsprofile an den verschiedenen Stellen in dem Kapillarrheometer: a) Strömung im Testkanal, b) Strömung in der Düse, c) Strömung am Düsenaustritt mit Strangaufweitung und d) Strömung des erstarrten Materials.*

schreiben als

$$\dot{\gamma}_{ap} = \frac{6\dot{V}}{bh^2}. \tag{2.3}$$

Der Quotient aus Schubspannung und Scherrate ist die *Viskosität* η. Die *scheinbare Viskosität* ist damit gegeben zu

$$\eta_{ap} = \frac{\tau_{ap}}{\dot{\gamma}_{ap}}. \tag{2.4}$$

In Gl. 2.4 fehlt die *scheinbare Schubspannung*. Der Druckverlust $\Delta p = p_E - p_A$, der sich über der Düse einstellt, ist zu τ_{ap} proportional. Daraus folgt, dass

$$\tau_{ap} = \Delta p \frac{h}{2L_D}. \tag{2.5}$$

p_E und p_A sind die Druckwerte jeweils am Eintritt der Düse und der Umgebungsdruck. Es ist möglich, mittels einer einfachen Korrektur, aus den scheinbaren Größen die reale Scherrate zu bestimmen. Das dazu verwendete Korrektionsverfahren geht auf *Weißenberg/Rabinowitsch* [27] zurück. Für die Schlitzdüse ergibt sich dies wie folgt

$$\dot{\gamma} = \frac{2}{3}\dot{\gamma}_{ap} + \frac{1}{3}\tau_{ap}\frac{d\dot{\gamma}_{ap}}{d\tau_{ap}}. \tag{2.6}$$

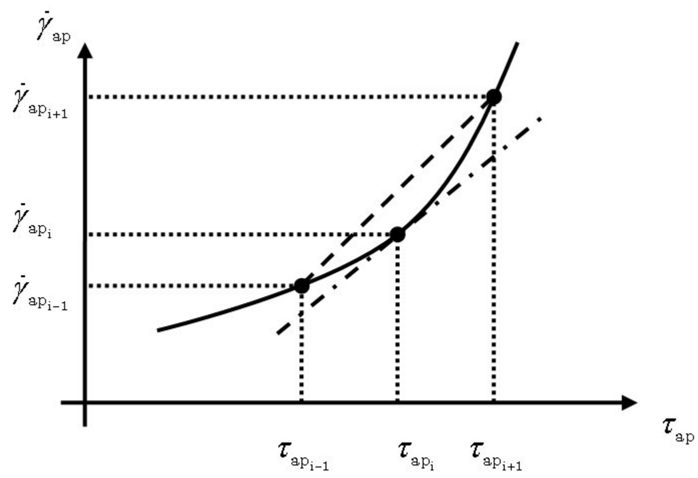


Abbildung 2.5: *Schematische Darstellung der Weißenberg/Rabinowitsch Korrektur.*

Für die Korrektur der Messwerte wird die Gl. (2.6) umgeschrieben zu

$$\dot{\gamma}_i = \frac{2}{3}\dot{\gamma}_{ap_i} + \frac{1}{3}\tau_{ap_i}\frac{\dot{\gamma}_{ap_{i+1}} - \dot{\gamma}_{ap_{i-1}}}{\tau_{ap_{i+1}} - \tau_{ap_{i-1}}} \tag{2.7}$$

mit $i = [1, N]$. Dies ist in Abb. 2.5 zu sehen. Für $i = 1$ und $i = N$ gelten vereinfachte Gleichungen,

$$\dot{\gamma}_1 = \frac{2}{3}\dot{\gamma}_{ap_1} + \frac{1}{3}\tau_{ap_1}\frac{\dot{\gamma}_{ap_2}}{\tau_{ap_2}}, \tag{2.8}$$

$$\dot{\gamma}_N = \dot{\gamma}_{ap_{N-1}}\frac{\dot{V}_N}{\dot{V}_{N-1}}. \tag{2.9}$$

Repräsentation der Viskosität durch analytische Fließkurven

Im Folgenden wird auf drei Ansätze, die scherratenabhängige Viskosität (*Fließkurve*) durch analytische Funktionen beschreiben, beschrieben. Damit ist es möglich, die scherratenabhängige Viskosität durch wenige Parameter zu beschreiben und Materialien einfach zu verstehen. Diese Ansatzfunktionen wurden [3] entnommen.

Ostwald/de Wael'sche Potenzansatz [26]

$$\dot{\gamma} = \phi \cdot \tau_{xy}^m \tag{2.10}$$

Es ist $\phi = \hat{\phi}\,(\vartheta)$ die *Fluidität*. Sie ist eine stoffspezifische Größe. Hierbei ist m der *Fließexponent*. Mit der Gleichung

$$\eta = \frac{\tau_{xy}}{\dot{\gamma}} \tag{2.11}$$

lässt sich die Viskosität schreiben als

$$\eta = \phi^{-\frac{1}{m}} \dot{\gamma}^{\frac{1}{m}-1}. \tag{2.12}$$

Der Nachteil dieses Ansatzes ist, dass er meist nur für einen begrenzten Bereich von $\dot{\gamma}$ gültig ist.

Carreau Ansatz [8]

$$\tau_{xy} = \frac{\eta_0 \cdot \dot{\gamma}}{(1 + b\,|\dot{\gamma}|)^c}. \tag{2.13}$$

Mit Gl. 2.11 ergibt sich für die Viskosität

$$\eta = \frac{a}{(1 + b\,|\dot{\gamma}|)^c}. \tag{2.14}$$

Der Vorteil dieses Modells liegt in seinen physikalisch sinnvollen Konstanten der Plateauviskosität $\eta_0\,[Pas]$, einer Relaxationszeit $b\,[s]$ und einem Skalenparameter $c\,[-]$. Für Messungen bei geringer Scherrate (mit $\dot{\gamma} \to 0$) entspricht $a \approx \eta_0$ der *Nullviskosität*[5].

Münsted'scher Polynomansatz Der Münstedt'sche Ansatz aus [24] beruht auf der Darstellung der Viskosität durch ein Polynom vom Grad vier:

$$\lg \eta = \sum_{k=0}^{4} A_k \left(\lg |\dot{\gamma}|\right)^k. \tag{2.15}$$

In dieser Gleichung haben die Koeffizienten A_k keine direkte physikalische Interpretation. Das Verhalten bei den Grenzübergängen ist

$$\dot{\gamma} \to 0 \quad \Rightarrow \quad \eta \to \infty, \tag{2.16}$$
$$\dot{\gamma} \to \infty \quad \Rightarrow \quad \eta \to 0. \tag{2.17}$$

Es kann ein breiter Bereich an Scherrgeschwindigkeiten abgedeckt werden..

[5]Die *Nullviskosität* von Schmelzen hängt stark von dem Molekulargewicht ab, $\eta_0 = K \cdot \bar{M}_w^{a'}$, $a' = 3,4$.

2.2 Mechanische Instabilitäten

Während der Extrusion von PE-Schmelzen kann in Abhängigkeit der Scherrate $\dot{\gamma}$ der Druck in der Düse fluktuieren (siehe Abb. 2.6).

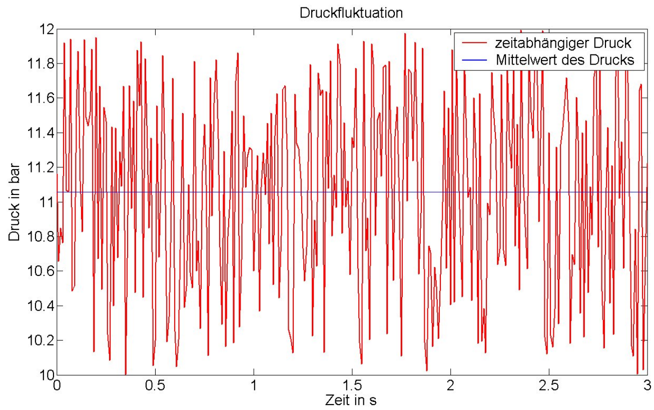


Abbildung 2.6: *Computergenerierte Fluktuationen des Drucks in einer beispielhaften Darstellung. Die rote Kurve beschreibt den Druck und die blaue Kurve beschreibt den mittleren Druck.*

Das sichtbare Resultat dieser *Druckfluktuationen* ist die Oberflächenbeschaffenheit des Extrudats. Bei niedrigen Scherraten bleibt die Oberfläche glatt. Dies ist der produktionstechnisch gewünschte Zustand. Nach dem Erreichen einer Grenzscherrate $\dot{\gamma}_{lim}$ treten Fließanomalien auf. Es sind auf der Oberfläche zunächst periodische Oberflächenzerwürfnisse zu sehen. Ihre Ursache wird in [39] und [10] diskutiert. Bei niedrigen Fluktuationen des Druckes um den mittleren Druck $\bar{p}$ heißt dieses Phänomen *Sharkskin*[6]. Für größere Scherraten geht dieser Effekt in einen Bereich über, in dem glatte und rauhe Oberflächen sich periodisch abwechseln. Dieser Effekt heißt *Stick-Slip*. Die Druckfluktuationen sind dabei wesentlich größer, als noch beim Sharkskin Effekt. Als letzte differenzierbare Erscheinungsform der Polymerschmelze taucht *Gross Melt-Fracture*[7] auf. Dies ist erkennbar an den unregelmäßig geformten Oberflächen des Extrudats. Die Abhängigkeit der Scherrate von der Schubspannung beschreibt [10] durch die Abb. 2.7. Die Abb. 2.8 stellt die verschiedenen Stufen der Fließanomalien dar. Diese Abbildung ist [30] entnommen. Darin ist zu erkennen, dass bei Variation der Schubspannung, nicht alle Bereiche der Fließkurve erreicht werden. Die verschiedenen Fließanomalien sind in Abb. 2.7 verschiedenen Bereichen von Scherraten zu geordnet.

[6] Haifischhaut
[7] Im weiteren als Melt-Fracture bezeichnet.

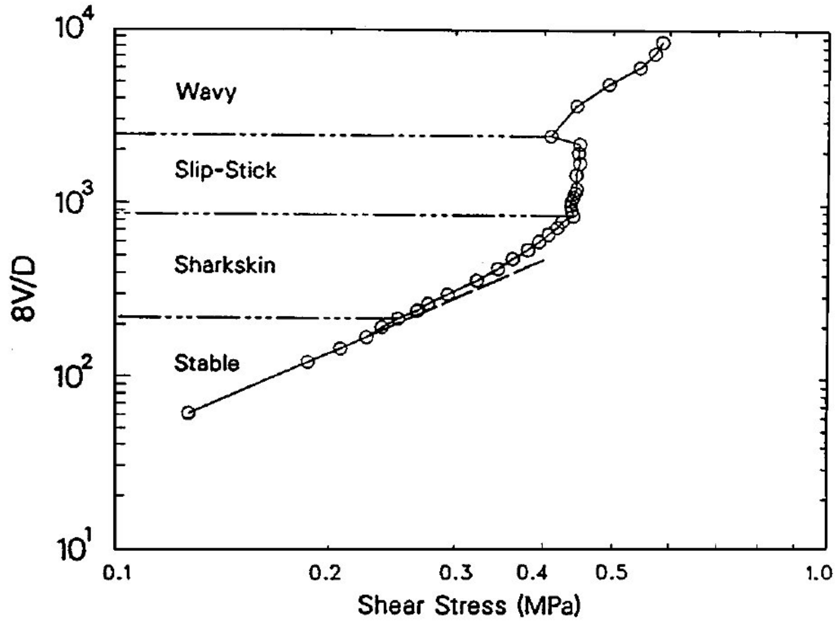


Abbildung 2.7: *Beispiel für Fließkurven aus [10].*

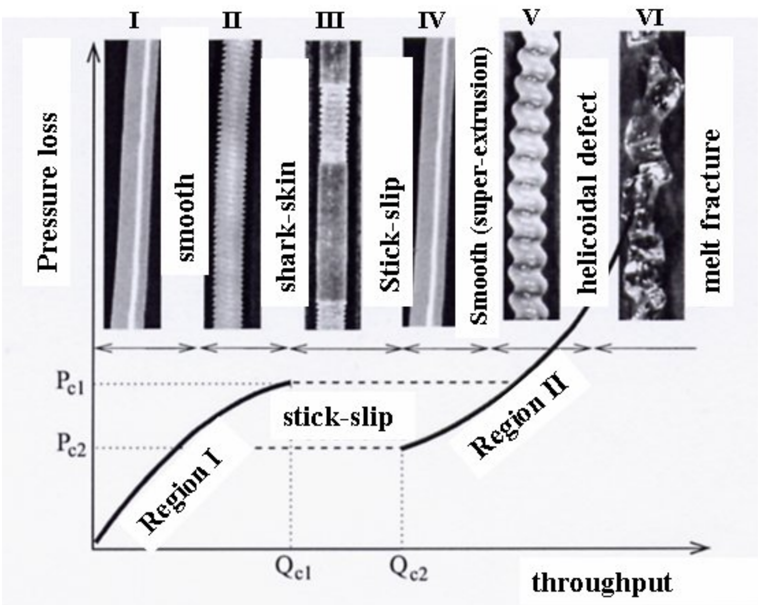


Abbildung 2.8: *Darstellung des zeitabhängigen Drucks und die zugehörigen Fließanomalien aus [30].*

2.3 Datenerfassung

Die gewonnenen Messdaten müssen analysiert werden. Das Aufzeichnen und Analysieren der Messdaten wird heute z.B. mit einem Rechner durchgeführt. Zunächst ist ein geeignetes Experiment nötig, dass denn gewünschten Effekt ohne störende Abhängigkeiten zeigt. Dann werden geeignete *Messsensoren* angeschlossen, die das physikalische Signal detektieren können und in ein analoges elektrisches Signal umwandeln. Da dieses Signal meistens relativ schwach ist wird es zunächst einem *Messverstärker* zugeführt, um es von diesem verstärken zu lassen. Das resultierende Signal gelangt danach zu einem *Analog/Digital Wandler (A/D-Wandler)*, indem das analoge Signal in ein digitales Signal umgewandelt wird. Nur dieses kann der PC speichern oder analysieren. Dieser Ablauf von dem Experiment bis hin zum PC, nennt man die *Messstrecke*. Ein Blockschaltbild der Messstrecke ist in Abb. 2.9.a zu sehen. Außerdem zeigt Abb. 2.9.b eine beispielhafte Belegung der einzelnen Blöcke

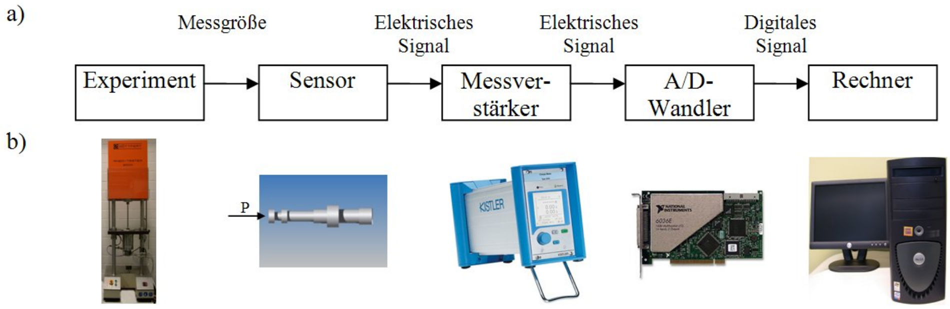


Abbildung 2.9: *Beispiel einer Messstrecke, wie sie bei dem hier durchgeführten Experimenten üblich ist. a) Symbolische Darstellung und b) exemplarische Bilder zu dem jeweiligem Symbol.*

Die *zeitliche Auflösung* Δt des digitalen Messsignals wird durch die *maximale Abtastrate* ν_{max} unserer Messstrecke gegeben. Es ist in erster Linie eine für die Hardware (Sensor und A/D-Wandler) spezifische Größe. Die Größe von ν_{max} muss in einem besonderem Verhältnis zu der größten im Signal enthaltenen Frequenz ν_{mes} stehen. Diese Verhältnis ist gegeben durch das *Shannon'sche Abtasttheorem* ([19] und [20])

$$\nu_{max} > 2\nu_{mes}. \tag{2.18}$$

Die Frequenz, die maximal möglich ist zu detektieren wird als *Nyquistfrequenz* $\nu_{Nyquist}$ bezeichnet. In Abb. 2.11 ist die Wichtigkeit des Shannon'schen Abtasttheorems dargstellt. Es werden bei ungenügender Abtastung fälschlicherweise hohe Frequenzen ($\nu > \nu_{Nyquist}$) in den Bereich der niedrigeren Frequenzen abgebildet. Dieser Effekt heißt fachlich *Aliasing*.

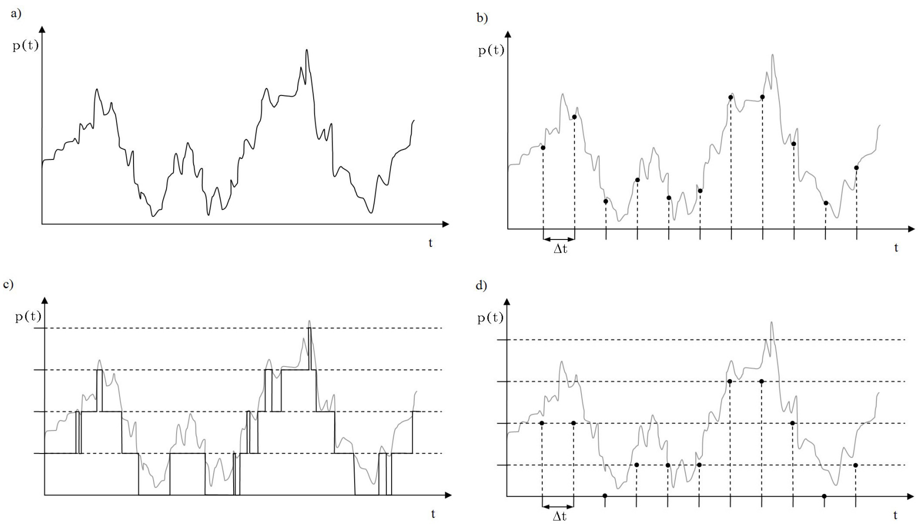


Abbildung 2.10: *Darstellung der Quantisierung und der diskreten Abtastung eines beispielhaften analogen Signales. Die Quantisierung erfolgt über eine Schwellwertdetektion (die gestrichelten Linien). Die Abbildung zeigt in a) das kontinuierliche analoge Signal, b) die diskrete äquidistante Abtastung , c) die Quantisierung (schwarz) und d) die Quantisierung und die äquidistante Abtastung des Signals.Die Punkte (●) repräsentieren Messpunkte und die graue gestrichelte (- -) Linie repräsentiert das analoge Signal.*

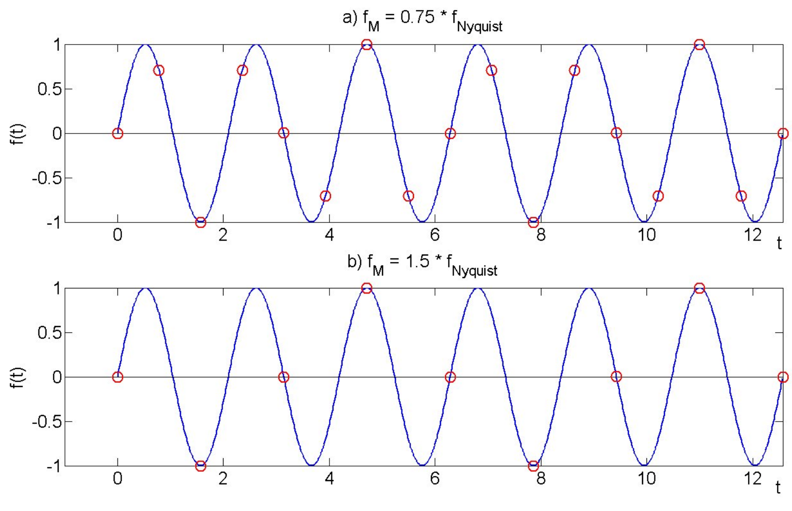


Abbildung 2.11: *Aliasing. Einfluss der Abtastfrequenz auf das erfasste Signal. Abtastfrequenz a) erfüllt und b) erfüllt nicht das Shannon'sche Abtatstheorem. Die durchgezogene Linie (-) beschreibt das kontinuierliches Signal und der Kreis (○) beschreibt die abgetasteten Messpunkte.*

Der AD-Wandler löst das Signal nicht nur zeitlich, sondern auch in der Amplitude diskret auf. Dies muss geschehen, da der Rechner nur digitale Signale weiterverarbeiten kann. Die Amplitude des analogen Signals wird in Stufen zerlegt. Dieser Vorgang nennt sich *Quantisierung*. Die Größenordnung der Quantisierung muss den Bedürfnissen entsprechend gewählt werden. Die Probleme, die sich dabei ergeben können, zeigt die Abb. 2.12. Die Amplitude des Messsignals enthält die gesuchte Information, aber auch

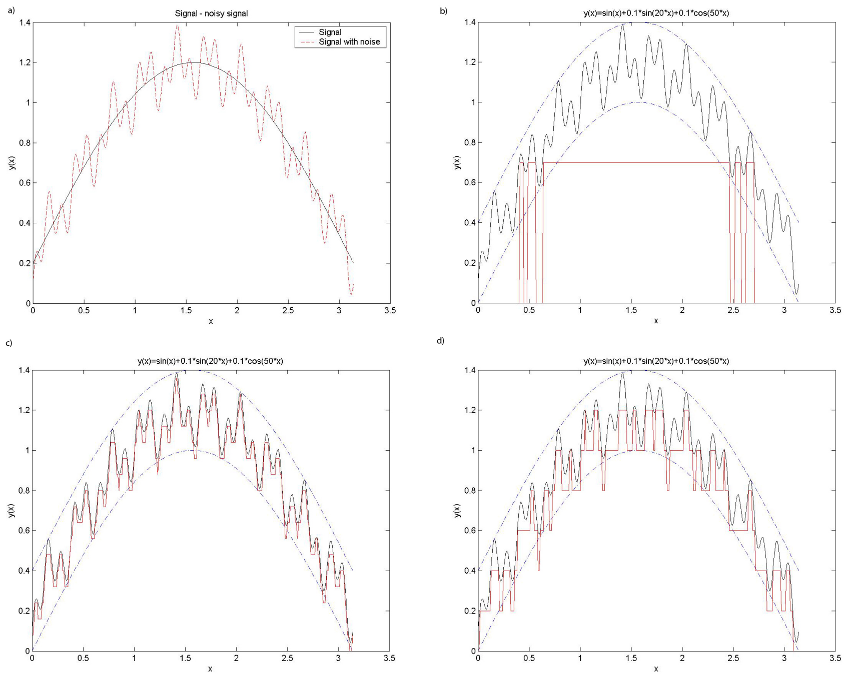


Abbildung 2.12: *Einfluss der Quantisierung auf das digitale Signal. a) Darstellung vom physikalischen Signal und gestörten Signal, b) zu grobe Quantisierung (Informationsverlust), c) zu feine Quantisierung (Informationsüberschuß) und d) ausreichend gute Quantisierung. In a) ist die Rote gestrichelte Linie das Signal mit Rauschen und die schwarze Linie das physikalische Signal. In b-d) ist die schwarze Kurve das Siganl mit Rauschen und die rote Linie ist das digitalisierte Signal.*

hochfrequentes Messrauschen, wie in Abb. 2.12.a zu sehen ist. Durch eine zu grobe Quantisierung (z.B. Abb. 2.12.b) wird das Rauschen unterdrückt. Gleichermaßen entsteht ein Informationsverlust. Bei einer zu feinen Quantisierung wird sogar noch das hochfrequentes Messrauschen gut detektiert. Es liegt ein Überschuss an Informationen vor, der zwar keine Fehler erzeugt, aber eine nicht vorhandene Genauigkeit vortäuscht. Ein guter Mittelweg ist es die Quantisierung in den Größenbereich des Rauschen zu legen, wie es beispielhaft in Abb. 2.12.d geschehen ist. Dies gilt in Fällen in denen das hochfrequente Messrauschen und das erwartete Signal in verschiedenen Größenordnungen liegen. Ist das Messrauschen aber in der Größenordung des Signals sollte das Rauschen auch noch ausreichend quantisiert sein.

Es gibt noch eine weiter Methode das hochfrequentes Rauschen im Messsignal zu unterdrücken. Diese Methode heißt *Oversampling*. Oversampling hat die Eigenschaft eines adaptiven Tiefpassfilters. In einer Messung wird jetzt nicht jeder Messwert gespeichert, sondern es wird über eine Anzahl von N Werten jeweils gemittelt und nur der gemittelte Wert[8] wir festgehalten. Für stochastisch verteiltes Rauschen ein partiell destructives Verhalten. Dieses Verfahren ist nur für Messungen sinnvoll, in denen die Anzahl an Messpunkten ausreichend groß ist. Außerdem muss die "Oversampling" Frequenz ν_{Over} auch dem Schannonschen Theorem genügen. Der klare Vorteil dieser Vorgehensweise ist die Verbesserung des *Signal-to-Noise Ratio (SNR)*. Zudem ist der Speicherbedarf um den Faktor n geringer.

Beispiel Abb. 2.13.a zeigt ein sinusförmiges Signal, dem zunächst Rauschen überlagert ist. Durch das Anwenden der Oversampling Methode wird der Anteil vom Rauschen am Signal in Abb. 2.13.b-d mit zunehmender Anzahl von gemittelten Punkten immer geringer.

[8]Ensemble Mittelung

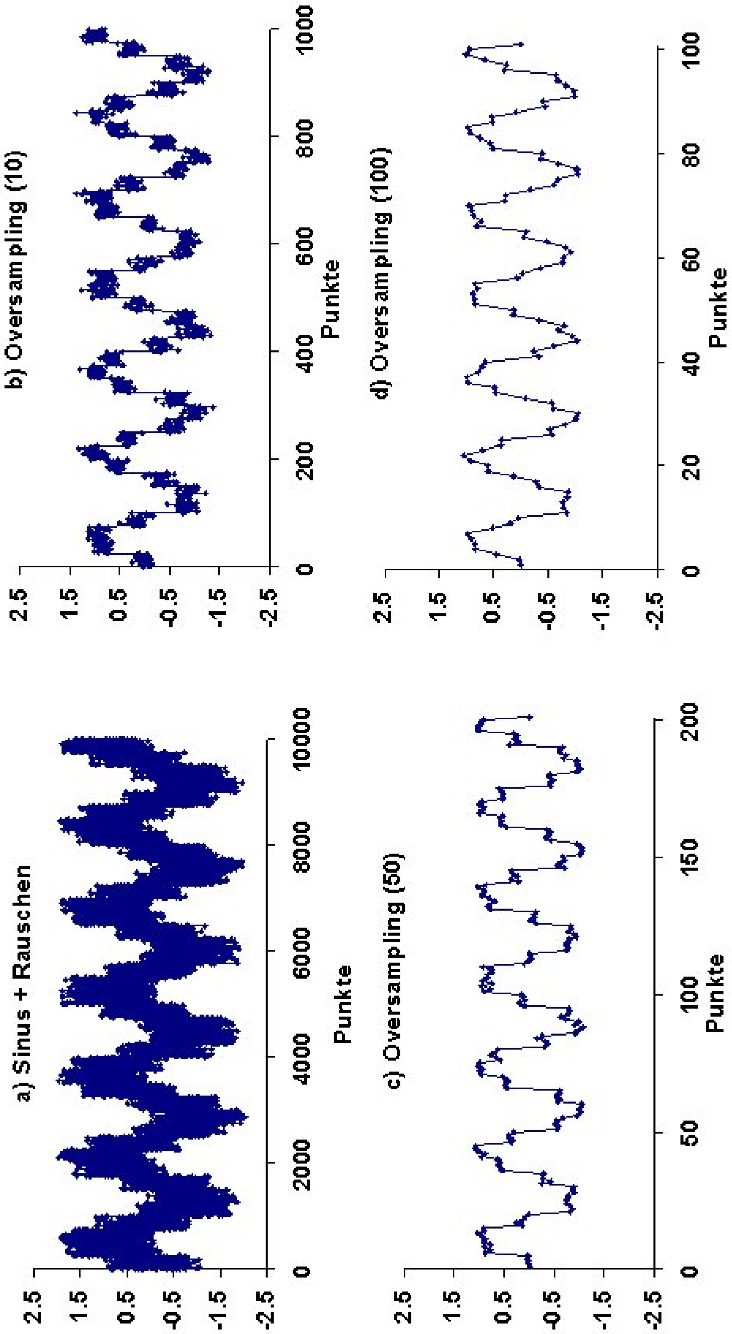


Abbildung 2.13: *Der Einfluss von "Oversampling" auf die zeitabhängigen Daten. a) Signal mit Rauschen aber ohne Oversampling b)-d) Oversampling b) Mittelung über jeweils x Punkte mit dem Abstand von x Punkten,* $x = 10$*, c)* $x = 50$ *und d)* $x = 100$ *Punkte. Die Punkte (•) entsprechen den Messpunkten.*

2.4 Datenanalyse

Kap. 2.3 führte den Begriff der Messstrecke ein. Dieses Kapitel beschreibt die letzte Station der Messstrecke, den Rechner. Nach der Messung liegen die Messdaten in Form von Dateien auf dem Rechner. Nun ist es die Aufgabe von speziellen Programmen diese "Rohdaten" auszuwerten. Die mathematischen Verfahren stellt dieses Kapitel vor. Als Literatur für die Fourierreihe und die Fourier Transformierte sind [37], [7], [5], [28] und [19] zu nennen. Mit den diskreten Darstellungen der Fourierreihe, Fouriertransformation und der Autokorrelationsfunktion befasst sich ausgiebig [20] und [18].

2.4.1 Fourierreihe (FR)

Die *Fourierreihe* dient der Anwendung z.B. zeitabhängige Signale durch eine Superposition periodischer Funktionen darzustellen. Das unendliche Signal muss eine feste Periode T haben ([37], [6]). Ist das Signal endlich wird es periodisch fortgesetzt. D.h. die Periodendauer T ist dann die gesamte Signallänge.

Jedes Signal ist mittels der Superposition von periodischen Standardfunktionen darstellbar. Als solche Standardfunktionen dienen uns die trigonometrischen Funktionen. Damit ergibt sich die nach Jean-Baptist Joseph Fourier bezeichnete Fourierreihe einer Funktion zu

$$f(t) = \sum_{n=-\infty}^{+\infty} \left(a_n \cos(\omega_0 nt) + b_n \sin(\omega_0 nt)\right). \tag{2.19}$$

Dabei ist

$$\begin{aligned} n &\in \mathbb{N}, \\ \omega_0 &= 2\pi\nu_0 = \frac{2\pi}{T}. \end{aligned}$$

Es werden a_n und b_n *Fourierkoeffizienten* genannt. Sie werden mit den folgenden Gleichungen

$$a_0 = \frac{1}{\pi}\int_0^{2\pi} f(t)\,dt, \tag{2.20}$$

$$a_n = \frac{1}{\pi}\int_0^{2\pi} f(t)\cos(nt)\,dt, \tag{2.21}$$

$$b_n = \frac{1}{\pi}\int_0^{2\pi} f(t)\sin(nt)\,dt, \tag{2.22}$$

bestimmt. Aus den Symmetrieeigenschaften der trigonometrischen Funktionen [9] vereinfacht sich die Darstellung der Fourierreihe zu

$$f(t) = \frac{A_0}{2} + \sum_{n=1}^{+\infty} \left(A_n \cos(\omega_0 nt) + B_n \sin(\omega_0 nt)\right). \tag{2.24}$$

Darin sind nun die Fourierkoeffizienten gegeben durch

$$\begin{aligned} A_n &= a_n + a_{-n}, \\ B_n &= b_n - b_{-n}. \end{aligned}$$

Die Fourierreihe lässt sich auch in komplexer Darstellung formulieren zu

$$f(t) = \sum_{n=1}^{+\infty} C_n e^{-i\omega_0 t} \tag{2.25}$$

mit dem Fourierkoeffizienten

$$C_n = \frac{1}{\pi} \int_0^{2\pi} f(t)\, e^{-i\omega_0 t} dt. \tag{2.26}$$

Beispiel: Für die Rechteckfunktion mit der Periodendauer $T = 4\pi$ sind in 2.14.a-d die Fourierreihe mit verschiedenen Ordnungen n dargestellt. Zu sehen ist, dass mit steigender Ordnung n die Fourierreihe immer besser die Ursprungsfunktion annähert. Auch mit steigender Ordnung n bleibt das Überschwingen der Fourierreihe erhalten. Dieses Phänomen heißt *Gibb'sches Phänomen*.

[9] Symmetrieeigenschaften der trigonometrischen Funktionen

$$\begin{aligned} \text{ungerade:} \quad \sin(t) &= -\sin(-t), \\ \text{gerade:} \quad \cos(t) &= \cos(-t), \end{aligned} \tag{2.23}$$

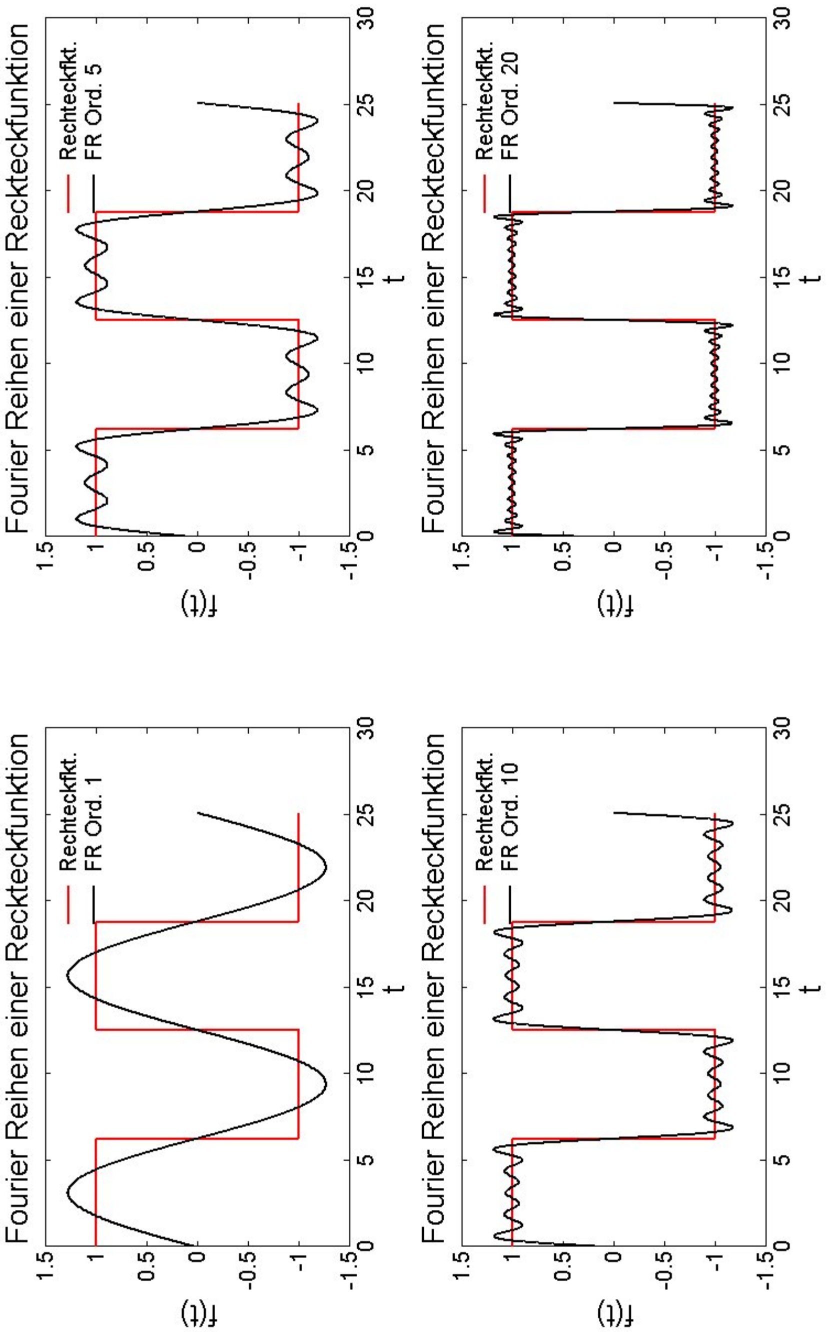


Abbildung 2.14: *Fourier Reihe einer Rechteckfunktion mit verschiedener Ordnung a) $n = 1$, b) $n = 5$, c) $n = 10$ und d) $n = 20$. Die rote Linie kennzeichnet die Rechteckfunktion und die schwarze Linie kennzeichnet die zugehörige Fourierreihe.*

2.4.2 Fourier Transformation (FT)

Für eine periodische Funktion mit einer Periodendauer $T < \infty$ war die Darstellung als Fourierreihe mit ihren diskreten Frequenzen ausreichend. Geht $T \rightarrow \infty$ (aperiodischer Fall), so wird aus der Summation eine Integration. Der diskrete Satz von Frequenzen wird durch einen kontinuierlichen Satz ersetzt. Diese neue Darstellung heißt *Fourier Transformation* oder *Fourier Integral* einer Funktion und wird definiert als

$$F(\omega) = \int\limits_{-\infty}^{+\infty} f(t)\, e^{-i\omega t} dt. \tag{2.27}$$

Die Symmetrieeigenschaften von $f(t)$ gehen auf ihre Fourier Transformierte $F(\omega)$ über. Die Rücktransformation in den Zeitbereich ist analog zur Hintransformierten zu bilden

$$f(t) = \frac{1}{2\pi} \int\limits_{-\infty}^{+\infty} F(\omega)\, e^{+i\omega t} d\omega. \tag{2.28}$$

Es folgen einige wichtige Eigenschaften der FT. Dabei sei

$$\begin{array}{rcl} f(t) & \leftrightarrow & F(\omega), \\ g(t) & \leftrightarrow & G(\omega), \\ a, b & \in & \mathbb{R}. \end{array}$$

Linearitätstheorem:

$$a \cdot f(t) + b \cdot g(t) \leftrightarrow a \cdot F(\omega) + b \cdot G(\omega). \tag{2.29}$$

Erster Verschiebungssatz: Eine Verschiebung in der Zeit erzeugt eine Modulation in der Frequenz,

$$f(a - t) \leftrightarrow F(\omega)\, e^{-i\omega a}. \tag{2.30}$$

Zweiter Verschiebungssatz: Eine Modulation in der Zeit erzeugt eine Verschiebung in der Frequenz,

$$f(t)\, e^{-i\omega_0 t} \leftrightarrow F(\omega - \omega_0). \tag{2.31}$$

Skalierung: Bei einer Stauchung bzw. Streckung der Zeitachse folgt,

$$f(at) \leftrightarrow \frac{1}{|a|} F\left(\frac{\omega}{a}\right). \tag{2.32}$$

Parseval'sche Theorem:

$$\int_{-\infty}^{+\infty} f_1(t) \cdot f_2^*(t)\, dt = \int_{-\infty}^{+\infty} F_1(\omega) \cdot F_2^*(\omega)\, d\omega. \tag{2.33}$$

Das Symbol * bedeutet die konjugiert komplexe Variable. Ein relevanter Spezialfall des Parseval'schen Theorems ist

$$\int_{-\infty}^{+\infty} |f(t)|^2\, dt = \int_{-\infty}^{+\infty} |F(\omega)|^2\, d\omega. \tag{2.34}$$

FT einer Ableitung: Für die zeitliche Ableitung der Funktion $f(t)$ folgt,

$$FT\left(f'(t)\right) \leftrightarrow i\omega F(\omega). \tag{2.35}$$

Zur graphischen Darstellung eignen sich die Amplitude und die Phasenverschiebung. Die Amplitude

$$A(\omega) = |F(\omega)| = \sqrt{Re\{F(\omega)\}^2 + Im\{F(\omega)\}^2} \tag{2.36}$$

wird auch *Frequenzgang* genannt. Die Phasenverschiebung

$$\varphi(\omega) = \arctan\left(\frac{Im\{F(\omega)\}}{Re\{F(\omega)\}}\right) \tag{2.37}$$

trägt auch den Namen *Phasengang*.

2.4.3 Fenster Funktionen

Die Forderung der FT an die zu transformierende Funktion ist, dass sie stetig fortsetzbar ist. In der Anwendung ergeben sich häufig Funktionen oder Folgen von Datenpunkten, die sich nicht stetig fortsetzen lassen. Eine direkte Transformation dieser Funktionen oder Folgen ist ungünstig. Dies zeigt die Abb. 2.15. Der daraus resultierende Effekt wird *Leck-Effekt*

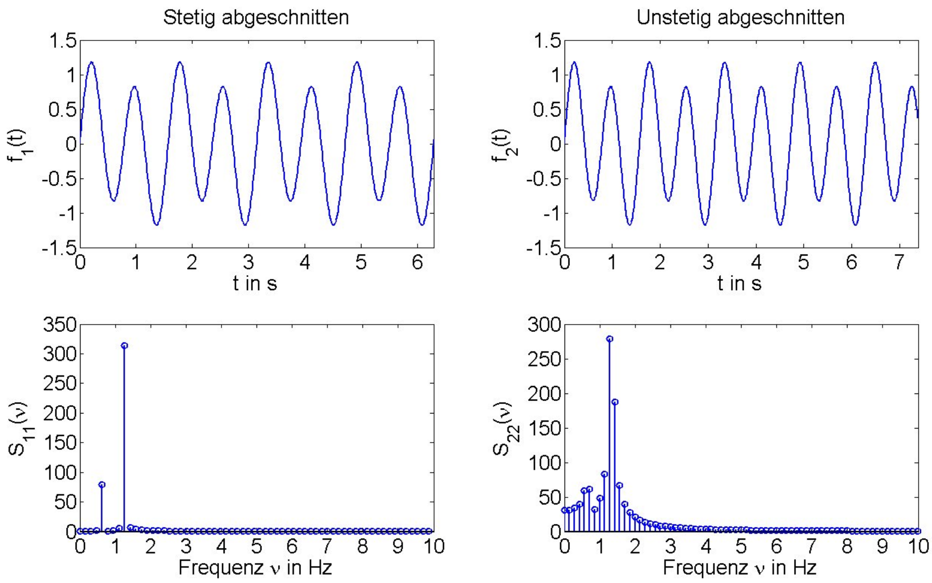


Abbildung 2.15: *Beispiel für den Leck-Effekt. Oben Zeitsignal, unten FT des Signals. Links stetig fortsetzbare Funktion und rechts unstetig fortsetzbare Funktion.*

genannt.

Um dies zu umgehen wird die zu transformierende Funktion mit einer *Wichtungs-* oder *Fensterfunktion* multipliziert. Fensterfunktionen sind immer gerade Funktionen, d.h. ihre FT hat keinen Imaginärteil. Für Qualität einer Fensterfunktion steht

- die Intensität der zentralen Spitze und
- die *3-dB Bandbreite*.

Dabei sollte die zentrale Spitze möglich viel Intensität enthalten, damit die *Seitenbänder* (Sideloops) möglichst klein bleiben. Die 3-dB Bandbreite gibt an, in welchem Bereich die zentrale Spitze auf die Hälfte ihrer Intensität abgefallen ist. In Tab. 2.1 und 2.2 sind die Gleichungen für einige einfache Fensterfunktionen zu finden.

Tabelle 2.1: *Fensterfunktionen im Zeitbereich.*

Fenster	Zeitbereich
Rechteck	$f(t) = \begin{cases} 1, & -\frac{T}{2} \leq t \leq \frac{T}{2} \\ 0 & \text{sonst.,} \end{cases}$
von Hann	$f(t) = \begin{cases} \cos^2 \frac{\pi t}{T}, & -\frac{T}{2} \leq t \leq \frac{T}{2} \\ 0 & \text{sonst.,} \end{cases}$
Hamming	$f(t) = \begin{cases} a + (1-a)\cos^2 \frac{\pi t}{T}, & -\frac{T}{2} \leq t \leq \frac{T}{2} \\ 0 & \text{sonst.,} \end{cases}$
Gauß	$f(t) = \begin{cases} \frac{1}{\sigma\sqrt{2\pi}} \exp -\frac{1}{2}\frac{t^2}{\sigma^2}, & -\frac{T}{2} \leq t \leq \frac{T}{2} \\ 0 & \text{sonst.,} \end{cases}$

Tabelle 2.2: *Fensterfunktionen aus Tab. 2.1 im Frequenzbereich.*

Fenster	Frequenzbereich
Rechteck	$\lvert F(\omega)\rvert^2 = \left(\frac{\sin\left(\omega\frac{T}{2}\right)}{\omega\frac{T}{2}}\right)^2$
von Hann	$F(\omega) = \frac{T}{4}\sin\left(\frac{\omega T}{2}\left(\frac{1}{\pi-\omega\frac{T}{2}} + \frac{2}{\omega\frac{T}{2}} - \frac{1}{\pi+\omega\frac{T}{2}}\right)\right)$
Hamming	$F(\omega) = \frac{T}{4}\sin\left(\frac{\omega T}{2}\left(\frac{1-a}{\pi-\omega\frac{T}{2}} + \frac{2(1+a)}{\omega\frac{T}{2}} - \frac{1-a}{\pi+\omega\frac{T}{2}}\right)\right)$
Gauß	$F(\omega) = \frac{1}{2}exp\left(\sigma^2\omega^2/4\right)\left(erfc\left(-\frac{i\sigma^2\omega^2}{\sqrt{2}} + \frac{T^2}{8\sigma^2}\right) + erfc\left(\frac{i\sigma^2\omega^2}{\sqrt{2}} + \frac{T^2}{8\sigma^2}\right)\right)$

2.4.4 Diskrete Fourier Transformation (DFT)

Für die messtechnische Anwendung ist die integrale Formulierung der FT zunächst wenig hilfreich. Während einer Messung werden diskrete Datenpunkte f_k aufgezeichnet und keine kontinuierliche Funktionen. Deswegen führen wir jetzt die *diskrete Fourier Transformation (DFT)* ein,

$$F_j = \frac{1}{N} \sum_{k=0}^{N-1} f_k e^{-\frac{i2\pi}{N} kj}. \tag{2.38}$$

Die Rücktransformation (FT^{-1}) wird gebildet mittels,

$$f_k = \sum_{j=0}^{N-1} F_j e^{\frac{i2\pi}{N} kj}. \tag{2.39}$$

Es gilt zu beachten, dass die Rücktransformation keinen Faktor $\frac{1}{N}$ mehr enthält!

Beispiel: In Abb.2.16 ist die diskrete Fourier Transformierte der Funktion $f(t) = cos(20\pi t) + sin(40\pi t)$ dargestellt. Sie ist mit der MATLAB[10] Funktion *fft()* berechnet. Die Nyquistfrequenz ergibt sich aus den Parametern des Programms zu $\nu_{Nyquist} = 50$ Hz.

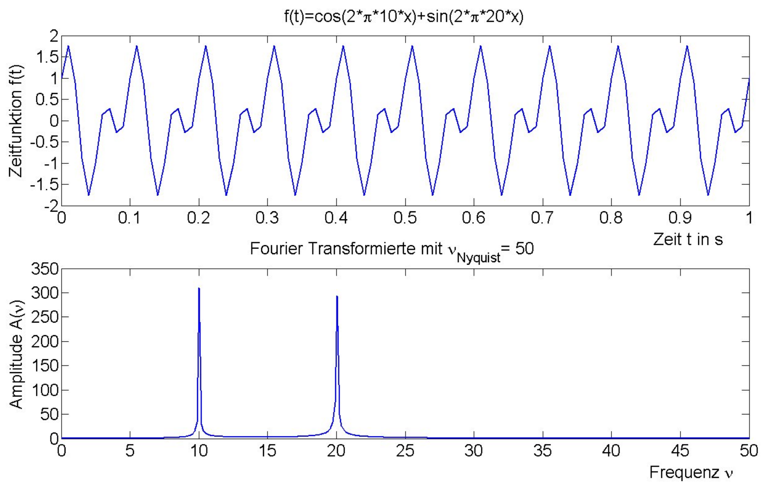


Abbildung 2.16: *DFT einer Testfunktion: a) Testfunktion im Zeitbereich, b) Spektrum der Testfunktion im Frequenzraum.*

[10] Eine interaktives System für numerische Berechnungen ([15]).

2.4.5 Spektrale Leistungsdichte (SLD)

Die in einem FT Spektrum enthaltene Leistung lässt sich durch die Funktion

$$S_{FF} = F^{*}(\omega) \cdot F(\omega) = |F(\omega)|^{2} \tag{2.40}$$

darstellen. $S_{FF}(\omega)$ nennt man die *spektrale Leistungsdichte* oder auch das *Leistungsdichtespektrum*.

2.4.6 Autokorrelationsfunktion (AKF)

Die *Autokorrelationsfunktion* eines kontinuierlichen Signals $f(t)$ ist definiert zu

$$R_{ff}(\tau) = \frac{1}{2T} \int_{-T}^{+T} f(t) \cdot f(t-\tau)\, dt. \tag{2.41}$$

Sie stellt einen Vergleich der Funktionswerte miteinander dar. Sie liefert eine Aussage über das Gedächtnis einer Funktion. Die Variable τ heißt *Korrelationszeit*. Sie ist der zeitliche Abstand zwischen zwei zu vergleichenden Punkten der Funktion.

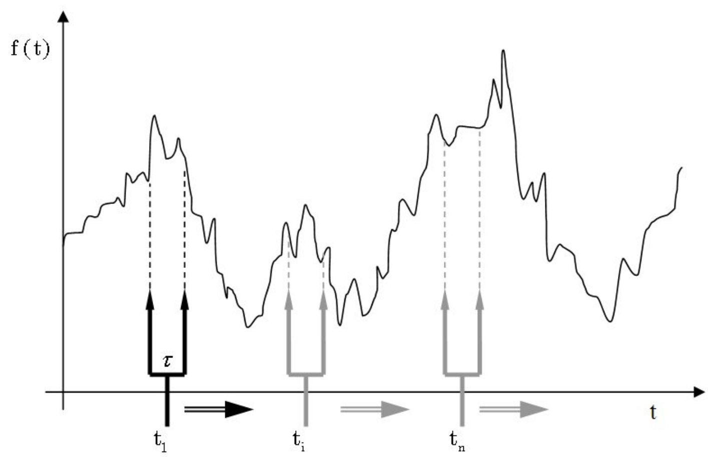


Abbildung 2.17: *Die AKF wird berechnet, indem man die $f(t)$ mit $f(t+\tau)$ für alle Datenpunkte korreliert.*

Beispiel: Abb. 2.17 zeigt anschaulich das Prinzip der AKF. Gegeben sei ein fester Abstand der genau τ beträgt. Dann wird jeder Funktionswert $f(t)$ mit dem um τ verschobenen Funktionswert $f(t+\tau)$ korreliert. Diese Werte summieren sich zu einem $R_{ff}(\tau)$. Dieser Vorgang wird für alle τ durchgeführt. Daraus entsteht die kontinuierliche Autokorrelationsfunktion.

2.4.7 Autokorrelationsfolge (DAKF)

Die Autokorrelationsfunktion ist für kontinuierliche Funktionen definiert. Ist ein diskreter Satz an Messwerten gegeben, ergibt sich die Notwendigkeit eine *Autokorrelationsfolge/ diskrete Autokorrelationsfunktion* zu definieren. Ist ein Prozess stationär, so kann er meistens als ergodisch[11] angenommen werden. Für einen ergodischen Prozess gilt, dass die Zeitmittelung[12] für jedes Teilchen durch eine Ensemble Mittelung[13] über alle Teilche ersetzt werden darf. Damit schreibt sich die Autokorrelationsfolge zu

$$r_{ff}(\tau) = \frac{1}{N} \sum_{t=0}^{N-1-\tau} f(t) f(t+\tau). \tag{2.42}$$

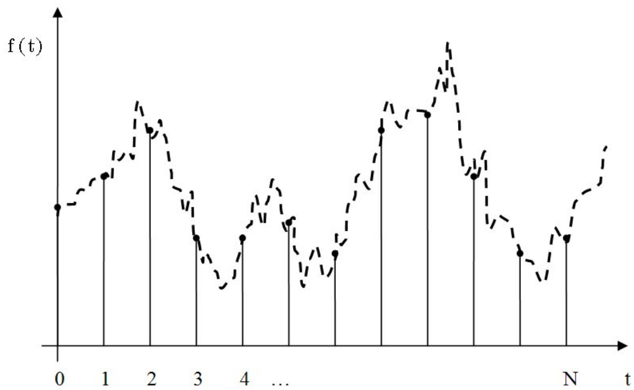


Abbildung 2.18: *Beispielfunktion. Die gestrichelte Linie (- -) beschreibt das kontinuierliche physikalische Signal und die Punkte (•) die diskreten Messpunkte.*

Beispiel: Gegeben sei das Signal aus Abb. 2.18, das äquidistant abgetastet ist. Dann zeigt Abb. 2.19 die ersten beiden Rechenschritte der DAKF. Im ersten Schritt multipliziert sich jeder Punkt mit sich selbst. Dann werden alle Werte aufaddiert. Im zweiten Schritt multipliziert sich jeder Punkt mit seinem Nachbarn. Danach erfolgt wieder die Summation über alle entstandenen Werte.

[11] *Ergoden Hyphothese* ([4], [17], [20]) angenommen werden:

$$\begin{aligned} \bar{f} &\rightarrow f^e, \text{wenn} \\ T &\rightarrow \infty \end{aligned}$$

[12] Zeitmittelung: $\bar{f} = \lim_{T\rightarrow\infty} \frac{1}{T} \int_{t_0}^{t_0+T} f(\vec{x}, t)\, dt$

[13] Ensemble Mittelung: $f^e = \lim_{N\rightarrow\infty} \frac{1}{N} \sum_{n=1}^{N} f(\vec{x}, t)\, dt$

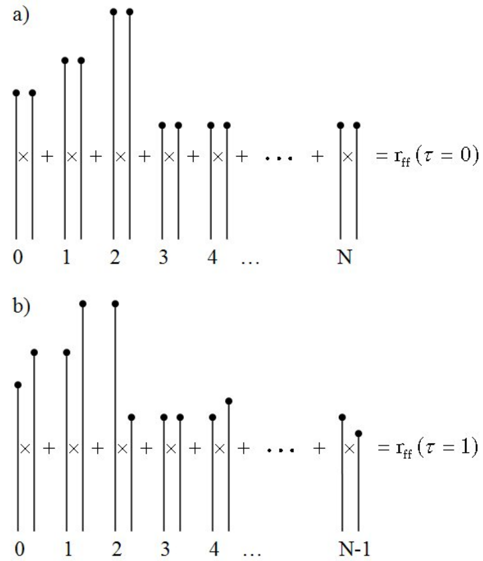


Abbildung 2.19: *Die ersten beiden Rechenschritte der AKF. Für die Beispielfunktion aus Abb. 2.18. Das Kreuz (×) symbolisiert die Multiplikation der jeweiligen Funktionswerte.*

Die Summation erfolgt für $\tau \rightarrow N$ mit immer weniger Summanden. Damit tritt nun das Problem auf, dass der Vorfaktor $\frac{1}{N}$ nicht mehr der Wahrheit an gebrauchten Funktionswerte entspricht. Er wird durch $\frac{1}{N-\tau}$ ersetzt. Dies liefert einen *erwartungstreuen* Wert. Fachlich heißt $\frac{1}{N-\tau}$ eine *Fensterfunktion* (Gewichtung). Die erwartungstreue Autokorrelationsfolge ergibt sich zu

$$r_{ff}(\tau) = \frac{1}{N-\tau} \sum_{t=0}^{N-1-\tau} f(t) f(t+\tau). \tag{2.43}$$

Die Werte der Zeitfunktion können beliebig groß sein und damit nimmt auch die DAKF beliebig große Werte an. Um die DAKF zu normieren wird sie durch ihren Wert an der Stelle $\tau = 0$ dividiert. Damit wird eine allgemeine Vergleichbarkeit dieser Folgen erreicht.

$$r_{ff}(\tau) = \frac{1}{N-\tau} \sum_{t=0}^{N-1-\tau} \frac{f(t) f(t+\tau)}{f(t) f(t+0)}. \tag{2.44}$$

Durch diese Normierung liegen die Werte von $r_{ff}(\tau)$ in Interval $I = [-1; +1]$ und ihren Werten kommt die Bedeutung

$$r_{ff}(\tau) = \begin{cases} +1, & \text{totale Korrelation} \\ 0, & \text{keine Korrelation} \\ -1, & \text{totale Antikorrelation} \end{cases} \tag{2.45}$$

Abschließend wird bei einer DAKF der Erwartungswert $\bar{f}$ von den Funktionswerten subtrahiert. Die resultierende Folge heißt die *Autokovarianzfolge*

$$c_{ff}(\tau) = \frac{1}{N-\tau} \sum_{t=0}^{N-1-\tau} \frac{\left(f(t)-\bar{f}\right)\left(f(t-\tau)-\bar{f}\right)}{\left(f(t)-\bar{f}\right)\left(f(t)-\bar{f}\right)}. \tag{2.46}$$

Sie hat eine ähnlich Aussage, wie die AKF und im Falle einer mittelwertfreien Funktion sind beide identisch.

2.4.8 Wiener-Khinchine Theorem

Das *Wiener-Khinchine-Theorem* besagt:

Die spektrale Leistungsdichte ist die Fourier-Transformierte der Autokorrelationsfunktion [19].

Die Herleitung dieser Aussage erfolgt mit Hilfe des Parseval'schen Theorems und des Verschiebungssatzes:

$$\int_{-T}^{+T} f(t) \cdot f(t+\tau)\, dt = \int_{-\infty}^{+\infty} F^*(\omega) \cdot F(\omega)\, e^{2\pi\omega\tau} d\omega. \tag{2.47}$$

Multipliziert mit $\frac{1}{2T}$ und umgeformt ergibt sich

$$\frac{1}{2T} \int_{-T}^{+T} f(t) \cdot f(t+\tau)\, dt = \int_{-\infty}^{+\infty} \frac{|F(\omega)|^2}{2T} e^{2\pi\omega\tau} d\omega. \tag{2.48}$$

In Kap. 2.4.5 wurde die spektrale Leistungsdichte definiert zu

$$S_{FF}(\omega) = \frac{|F(\omega)|^2}{2T}. \tag{2.49}$$

Damit folgt aus Gl. 2.48 die Bestätigung des Wiener-Khinchine-Theorems

$$r_{ff}(\tau) = \int_{-\infty}^{+\infty} \frac{|F(\omega)|^2}{2T} e^{2\pi\omega\tau} d\omega. \tag{2.50}$$

Diesen Zusammenhang zwischen Zeit- und Frequenzbereich vergegenwärtigt die Abb. 2.20.

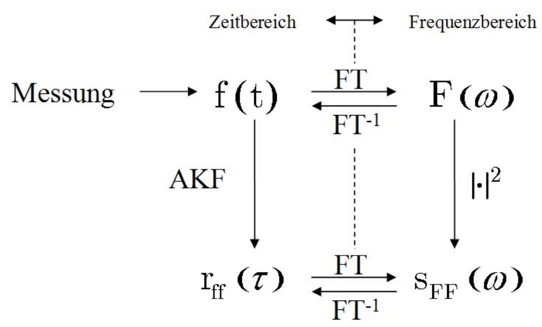


Abbildung 2.20: *Darstellung der Rechenoperationen von FT, AKF, SLD im Zeit- und Frequenzbereich.*

Diskrete spektrale Leistungsdichte

Durch das Bilden der Fourier Transformierte der Autokorrelationsfolge, so entsteht die spektrale Leistungsdichte (DSLD). Da die DAKF symmetrisch zur y-Achse ist, ist der imaginäre Teil der FT gleich Null. Damit lässt sich die diskrete spektrale Leistungsdichte berechnen aus

$$s_{ff}(\omega) = r_{ff}(0) + 2\sum_{\tau=1}^{\infty} r_{ff}(\tau)\cos(\omega\tau). \tag{2.51}$$

Darin treten die folgenden Grössen auf

Abtastfrequenz: $\nu_{abt} = \frac{1}{T}$,

Abtastzeit: Δt, Zeitabstand zwischen zwei Messpunkten,

Kreisfrequenz: $\omega = 2\pi\nu$.

Die Kreisfrequenz ist die am häufigsten genutzte Variable. Um einen einfacheren Zugang zu den resultierenden Größen zu haben wird im Weiteren die Frequenz ν benutzt werden. Die DAKF wurde zuvor bestimmt und wird in die DSLD eingesetzt zu werden. Die DSLD wird nicht an beliebig vielen Punkten ausgewertet, sondern an der gleichen Anzahl N von Punkten die uns durch die Messung gegeben ist. Damit ergibt sich die *Frequenzauflösung*/-schrittweite zu

$$\Delta\nu = \frac{\nu_{abt}}{N-1}. \tag{2.52}$$

Als Mengen treten auf

$$\begin{aligned} \tau &\in [0, T], \\ \omega &\in [0, 2\pi\nu_{abt}], \\ \nu &\in [0, \nu_{abt}]. \end{aligned} \tag{2.53}$$

Kapitel 3

Versuchsaufbau und -durchführung

Der Versuchsaufbau ist Abb. 3.1 abgebildet. Er besteht aus einem Kapillarrheometer, drei Messverstärkern und zwei Rechnern. Das Kapillarheometer wird in Kap. 3.1 genauer dargestellt.

Der Versuch beginnt mit dem Einfüllen des Testgranulates. Für dieses Experimente wurde ausschließlich Polyethylen (PE)[1]-Granulat benutzt. Eine Beschreibung der benutzten Testmaterialien folgt in Kap. 3.2. Das Granulat wird im Testkanal bei 150-200 °C aufgeschmolzen und dann mit konstanter Kolbengeschwindigkeit extrudiert. Während der Extrusion wird der Druck in Testkanal und der Düse mit insgesamt vier Drucksensoren (siehe Abb. 3.2) gemessen. Das elektrische Signal wird dann mit einem Ladungsverstärker (Kap. 3.3) proportional verstärkt und dann digitalisiert.

3.1 Das Kapillarrheometer

Das verwendete Kapillarrheometer ist ein *Rheotester 2000*, der Firma *Goettfert Werkstoff-Prüfmaschinen GmbH*. Es arbeitet entsprechend dem Funktionsprinzip, dass in Kap. 2.1 erklärt wurde. Der Rheotester 2000 kann in zwei Betriebsarten eingesetzt werden

1. konstante Kolbengeschwindigkeit $0,001\ mm/s \leq v_{kolben} \leq 20\ mm/s$ mit einer Schrittweite von $8 \times 10^{-5}\ mm/s$,

2. konstante Kraft, wobei $F \leq 20$ kN.

Für alle Experimente, die dieser Arbeit zugrunde liegen, wurde die Variante 1 gewählt, da hier durch ein breiteres Spektrum an Oberflächendefekten (vgl. Kap. 2.2) zu beobachten ist.

[1]chemisch: $[-CH_2 - CH_2-]_n$

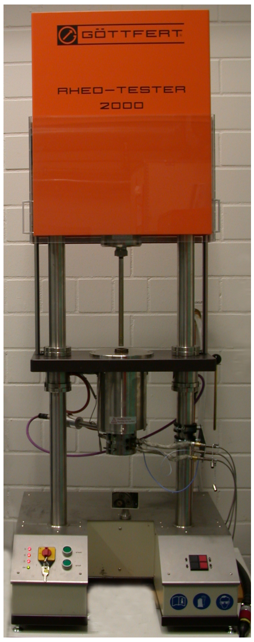


Abbildung 3.1: *Kapilarrheometer: Goettfert Rheotester 2000.*

Der Testkanal hat eine Durchmesser von $D_K = 15$ mm und einen Eintrittswinkel zur Düse von $\alpha = 180°$. Der Testkanal und die Düse werden mit drei *Widerstandsheizungen* auf eine frei wählbare Temperatur im Bereich von 60 °C$\leq \vartheta \leq$ 400 °C gehalten. Die Stabilität der Temperatur liegt bei ± 1 °C. Die Temperatur wird mit drei Thermoelementen PT 100 1/3DIN überwacht. Jedes Thermoelement ist einer Heizung zugeordnet (Abb. 3.2). Der Rheotester 2000 ist standardmäßig mit einem *DMS*[2]-*Drucksensor* im unteren Testkanal (Abb. 3.2) ausgestattet. Der Einsatzbereich des DMS-Drucksensors (App. A.1) liegt bei

- einem Druck von 0 bar$\leq p \leq$ 2000 bar und
- einer Temperatur $\vartheta \leq 400^o$ C.

Durch diesen Drucksensor wird das Experiment kontrolliert, da er der einzige Drucksensor ist, der seine Messergebnisse an die kommerzielle Steuerungssoftware liefert. Zum Beispiel wird bei einem Überschreiten des als maximal definierten Druckes das Experiment automatisch abgebrochen. Außerdem werden seine Messwerte benutzt, um die Viskosität und scheinbare Scherrate zu bestimmen. Seine Abtastungsrate ist nicht konstant. Er tastet somit nicht äquidistant ab.

Die verwendete Düse ist eine Schlitzdüse mit den Abmessungen $(30 \times 3 \times 0,3)$ mm. Sie ist eine Sonderanfertigung, die am Max-Planck-

[2]Dehnungs-Mess-Streifen

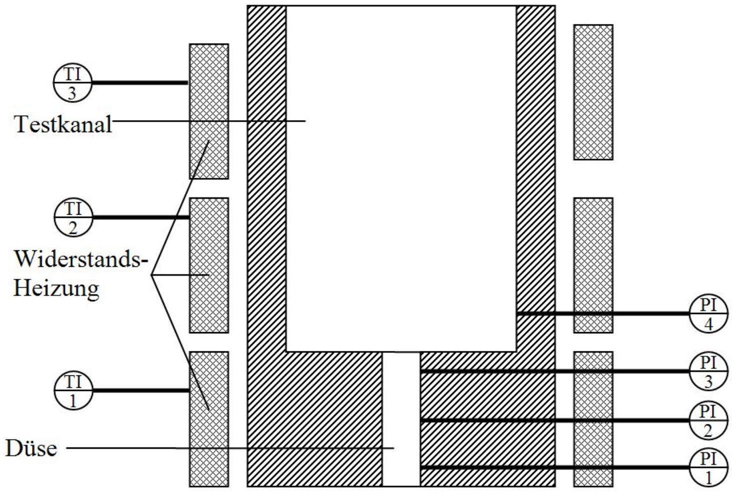


Abbildung 3.2: *Lokalisierung der Sensoren im Kapillarrheometer. TI ⇒ Temperatur Sensor, PI ⇒ Drucksensor. PI1-3: Piezo-Drucksensoren, PI4: DMS-Drucksensor.*

Institut für Polymere (MPI-P) angefertigt wurde. In ihr wurden drei zusätzliche *Piezo-Drucksensoren* (App. A.2) eingebaut, um den Druckverlauf entlang der Düse zu beobachten. Der Messbereich der Kistler Piezo-Drucksensoren ist 0 bar $\leq p \leq$ 2000 bar, mit einer Sensitivität von 2.5 pC/bar. Die maximale Betriebstemperatur ist auf 400 °C limitiert. Der Testkanal hat einen Durchmesser von 15 mm.

3.2 Polyethylen als Testmaterial

Polyethylen ist ein Standard Polymer, das industriell in großen Mengen (60 Mio. t je Jahr) produziert wird. Bei seiner Verarbeitung treten die in Kap. 2.2 charakterisierten Fließanomalien auf. Aufgrund verschiedener Konstitution[3], die das PE annehmen kann, treten die Fließanomalien verschieden stark auf. Daher ist es als Testmaterial für diese Versuche geeignet. Im Temperaturbereich zwischen $140 - 190$ °C ist es flüssig und degradiert noch nicht innerhalb der Versuchszeit. Eine Summierung der Charakteristika der verwendeten Proben ist in Tab. 3.1 zu sehen.

3.3 Datenaquisition

Die Verbindung zwischen Experiment und Rechner (siehe Abb. 3.3) läuft zunächst über einen Ladungsverstärker. Jeder der drei eingesetzten Piezo-Sensoren hat einen eigenen Ladungsverstärker vom Typ Kistler 5015A. Die

[3] Art und Menge an Verzweigungen ([22]).

Name	$\rho_{20°C}$ in kg/m^3	T_m in °C
PE-1	0,897 $\pm 0,026$	128,49
PE-2	0,874 $\pm 0,003$	116,06
PE-3	0,887 $\pm 0,026$	124,49
PE-4	0,845 $\pm 0,025$	118,87

Tabelle 3.1: *Characteristika der LLDPE-Proben.*

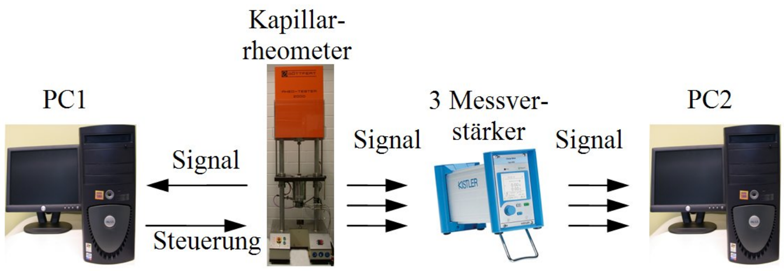


Abbildung 3.3: *Signalflussdiagram für den verwendeten Kapillarrheometraufbau.*

durch ihn verstärkte Ladung wird mit einer AD-Karten von National Instruments PCI-6036E am PC eingelesen. Die Software zur Datenaquisition ist ein in LabVIEW[4] geschriebenes Programm aus [41] und [40]. Dieses Programm führt bei der Messwerterfassung das "Oversampling" aus (Literatur: [36], [16]).

3.4 Versuchsablauf

Die durchgeführten Experimente laufen immer nach dem gleichen Schema ab:

- beide Computer (PC1 und PC2), das Kapillarrheometer und die Ladungsverstärker anschalten,
- Versuchsparameter (ϑ und v_{piston}) am Rechner PC1 einsetzen,
- das Rheometer braucht ca. 30 min, um auf einer konstanten Temperatur zu sein,
- den DMS-Drucksensor auf 0 bar setzen (Software intern; PC1),
- das Granulat für die Messung auswählen und in den Testkanal füllen,

[4]System zur virtuellen Instrumentierung (Kombination von Hardware und Software zur Erzeugung einer Funktion eines klassischen (Mess-) Instruments.).

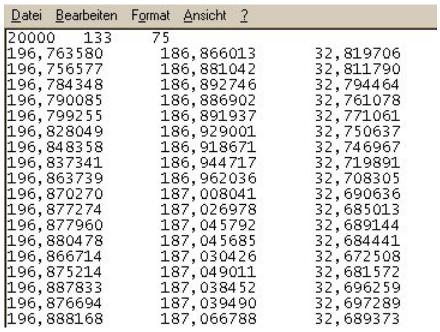
- ca. 10 min das Granulat schmelzen lassen,
- die Schmelze unter Vordruck setzen, damit die Düse gefüllt wird und Hohlräume geschlossen werden,
- Schmelze relaxieren lassen bis zu einem konstanten Druck [5],
- Aufzeichnungsparameter am PC2 wählen (Oversampling, Abtastfrequenz),
- Messung beginnen,
- Messwerte der Piezo-Sensoren aufzeichnen,
- Messung beenden (PC1 und PC2), Testkanal leeren,
- Testkanal gründlich mit einer Drahtbürste reinigen.

Für eine Messung mit PE wird bei dem 15 mm Testkanal ca. 35 g Polymer benötigt. Die Messdauer nimmt mit zunehmender Scherrate ab und liegt üblicherweise im Bereich von ca. 3-30 min.

[5] Die totale Relaxation auf 0 bar wäre wünschenswert, ist aber nicht sinnvoll, da sie zu lange dauert. Es besteht die Gefahr, dass das Material degradiert.

3.5 Aufbereitung und Analyse der zeitabhängigen Druckdaten

Die Daten der drei Drucksensoren werden zusammen in eine Datei gespeichert. Ihr Aufbau ist in Abb. 3.4 zu sehen.


Datei Bearbeiten Format Ansicht ?

20000	133	75
196,763580	186,866013	32,819706
196,756577	186,881042	32,811790
196,784348	186,892746	32,794464
196,790085	186,886902	32,761078
196,799255	186,891937	32,771061
196,828049	186,929001	32,750637
196,848358	186,918671	32,746967
196,837341	186,944717	32,719891
196,863739	186,962036	32,708305
196,870270	187,008041	32,690636
196,877274	187,026978	32,685013
196,877960	187,045792	32,689144
196,880478	187,045685	32,684441
196,866714	187,030426	32,672508
196,875214	187,049011	32,681572
196,887833	187,038452	32,696259
196,876694	187,039490	32,697289
196,888168	187,066788	32,689373

Abbildung 3.4: *Auszug aus einer Messdatei.*

Die Werte der ersten Zeile der Druckdatei von links nach rechts entsprechen:

- *Abtastfrequenz* ν_{abt},
- *Oversampling* n_{over} und
- *Nyquistfrequenz* $\nu_{Nyquist}$.

Für die Analyse der Daten wurde ein neues Programm geschrieben. Die Gründe und Vorteile sind

- das Programm ist optimal an die Form der Messdatei angepasst,
- der Algorithmus für die Berechnungen ist bekannt, weil selbst implementiert und
- es entstehen keine Urheberprobleme.

Als Programmiersprache wurde als erstes C benutzt. Die Vorteile sind:

- C ist die Sprach, in der Windows programmiert ist und sollte somit ein möglichst schnelles Programm liefern.
- Das fertige und compilierte Programm kann eigenständig auf jedem Rechner gestartet werden. Ein Matlab Programm im Gegensatz dazu kann immer nur unter einer MATLAB Umgebung laufen.
- Häufig haben Anwendungsprogramme, wie MATLAB, Mathematica, eine strengere Limitierung der Datenmengen.

Die Nachteile sind

- zeitintensive Programmierung,
- aufwendige Graphikprogrammierung,
- alle mathematischen Funktionen müssen selber programmiert und überprüft werden.

Der Programmcode ist in App. B.2 dieser Arbeit angehängt. Das Programm kann

- die Messdaten der einzelnen Sensoren der Messdatei auslesen und darstellen,
- ungünstige Anfangs- und Endbereiche können manuell entfernt werden,
- mittels der *Ausgleichsrechnung* ([34]) kann eine Ausgleichsgerade oder -parabel bestimmt werden und somit kann die Drift kompensiert werden,
- die Autokorrelation, Autokovarianz und die spektrale Leistungsdichte können berechnet und angezeigt werden,
- die Autokorrelationsfunktion kann mit einer frei wählbaren Filterfunktion multipliziert werden, um die Leistungsdichte zu verbessern,
- es können Testdaten[6] berechnet werden, um die Funktionalität des Programmes zu testen. Außerdem kann die *Housholder QR-Zerlegung* ([34]) auch zum Lösen von Matrizen angewandt werden.

Ein Nachteil der direkten Fourier Transformation ist, dass mit steigernder Anzahl von Punkten Rechenzeit quadratisch zunimmt. Deswegen wurde in einem zweiten Schritt entschlossen die Berechnung der spektralen Leistungsdichte aus der AKF und die FT der Druckdaten in Matlab zu programmieren. Die Vorteile liegen in der

- einfachen Anwendung von Matlab,
- leicht und automatisierbar darstell-, speicher- und ausdruckbaren Graphikdarstellung und

[6] Zur Verfügung stehen

- Sinus,
- Exponential,
- Polynom bis Grad 2 und
- Random Data.

- standartmäßig implementierte *Schnelle Fourier Transformierte (FFT)*[7] .

Die Programmcodes der MATLAB Programme sind in App. B.1.2 angehängt.

3.5.1 Aufbereitung der zeitabhängigen Druckdaten

Die graphische Darstellung der Druckdaten sieht prinzipiell aus, wie in Abb. 3.5. Im Bereich I steigt der Druck an, bis er im Bereich II ein konstantes Plateau erreicht hat, um das er oszilliert. Der Bereich IV entsteht, wenn die Messung beendet ist und das Restmaterial langsam aus der Düse fließt.

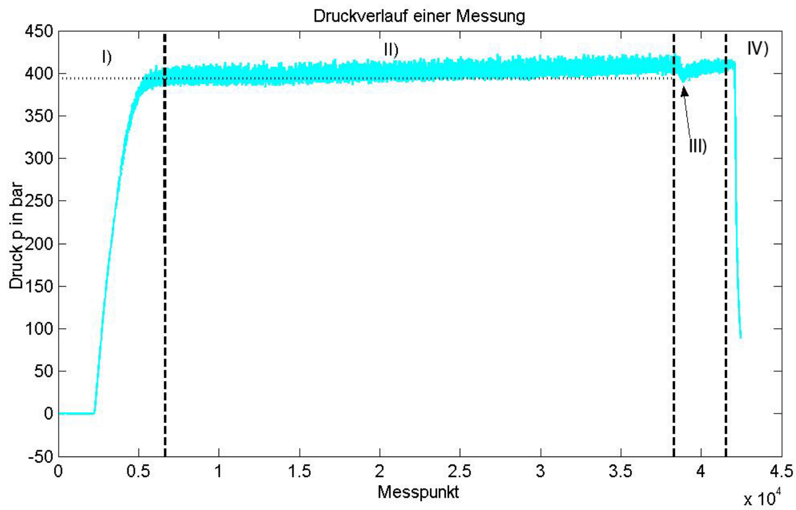


Abbildung 3.5: *Gemessener Druckverlauf mit den vier typischen Bereichen: I) Start der Messung, II) "stationärer" Bereich, IV) nach Beenden der Messung und III) Strömungsänderung.*

Für die Datenanalyse ist nur der Bereich II von Interesse. Daher können die Bereiche I und IV verworfen werden. Diese Möglichkeit ist in dem C-Programm aus App. B.2 gegeben. Prinzipiell besteht auch die Möglichkeit die Messung erst zu beginnen, wenn der Druck in der Düse den Bereich II erreicht hat. Dies ist aber gerade bei niedrigen Scherraten nicht einfach, da die Steigung in Bereich I relativ flach sein kann. Es ist dann sehr wahrscheinlich, das die Daten doch nachbearbeitet werden müssen. Bereich IV lässt sich einfacher umgehen, da an der Eindringtiefe des Kolbens in den Testkanal klar ersichtlich ist, wie lange das Experiment noch dauert.

Bereich III zeigte sich bei allen Messungen. Dieser typische Druckverlust tritt bei immer ab einer bestimmten Eindringtiefe des Testkolbens auf.

[7] Bei der DFT von N Punkten in MATLAB kommt nicht nur der *Butterfly* Algorithmus von [9] zur Anwendung. Die FFT in MATLAB greift auf weitere Faktorisierungen der Datenmenge N zurück.

Dies ist auf Änderungen der Strömungsverhältnisse innerhalb des Testkanals zurückzuführen.

Die eingesetzten Piezosensoren sind für eine statische Messung nicht optimiert. Es entsteht eine von Umgebungsbedingungen abhängige nicht reproduzierbare Drift der Messdaten. Diese Drift ist nicht konstant, damit hat ihre Funktionsdarstellung neben den konstanten Offset noch einen in der Zeit linearen Term. Da aber die Drift nicht unendlich zunehmen kann muss davon ausgegangen werden, dass noch einer weiterer nichtlinearer Term mit negativen Vorzeichen existiert. Damit lässt sich die Funktion der Drift $f_d(t)$ schreiben zu

$$f_d(t) = f_{d,0} + f_{d,1}t - f_{d,2}t^2 \tag{3.1}$$

Aber es ist möglich die Drift mit einer Anpassung an eine Gerade oder Parabel auszugleichen. Dazu ist folgendes Vorgehen ratsam:
Es wird nur der stationäre Bereich der Messung genommen. Mit Hilfe *Ausgleichsrechnung* ergibt sich eine Ausgleichsgerade oder -parabel. Diese wird von den Messdaten subtrahiert. Daraus ergibt sich eine um die Nullachse oszillierende Datenmenge, die sich gut mittels der DAKF analysieren lässt.

3.5.2 Analyse der zeitabhängigen Druckdaten

An die Aufbereitung der Druckdaten schließt sich deren Analyse an. Als mathematische Hilfsmittel diente die DAKF und die DFT. Das Prinzip von beiden ist in Kap. 2.4 erklärt. Die Datenanalyse wurde mit zwei Methoden durchgeführt. Beide endeten jeweils in der SLD. Die beiden Methoden entsprechen den beiden Wegen in Abb. 2.20.

A. Ausgehend von den Druckdaten wird mit der DFT in den Frequenzbereich gewechselt. Danach die SLD durch Quadrieren bestimmt.

B. Die Druckdaten werden zunächst im Zeitbereich mit der AKF bearbeitet und danach mit der DFT die SLD bestimmt.

Aus diesen beiden Methoden stehen vier Möglichkeiten zur Charakterisierung von Fließanomalien zur Verfügung:

1. die Druckfunktion $p(t)$,
2. die Autokorrelationsfunktion $r_{pp}(\tau)$,
3. die spektrale Leistungsdichte aus der AKF $s_{pp}(\nu)$ und
4. die spektrale Leistungsdichte aus der Druckfunktion $s_{PP}(\nu)$.

Die Messungen wurden in verschiedenen Messreihen durchgeführt.

Messreihe I, befasste sich mit dem Auffinden der charakteristischen experimentellen Fließanomalien für ein Polymer.

Messreihe II, diente die gefundenen Charakterisierungen beim gleichen Polymere auf eine andere Temperatur und höhere Scherraten zu erweitern.

Messreihe III, diente die gefundenen Charakterisierungen auf andere Polymere Systeme zu erweitern.

Messreihe IV untersucht den Ursprung für Auffälligkeiten der Messreihe I.

3.5.3 Methode A, AKF und SLD

Ein in MATLAB geschriebenes Programm (siehe App. B.1.2) ruft die gemäß Kap. 3.5.1 bearbeiteten Messdaten auf. Von den Druckdaten wird zu erst die Standardabweichung bestimmt. Im nächsten Schritt werden die Druckdaten mit einem Gauß-Fenster (Kap. 2.4.3) multipliziert. Dies dient der Verbesserung des Resultats der sich anschließenden DFT. Die DFT wird mit dem in MATLAB definierten FFT-Algorithmus bestimmt. Die SLD bestimmt sich gemäß der Grundlagen aus Kap. 2.4.5 in Gl. 2.40.

3.5.4 Methode B, FT und SLD

Im direkten Anschluss an die Aufarbeitung der Druckdaten wird die DAKF berechnet, wie in Kap. 2.4.7 , durchgeführt. Dies geschieht in dem C-Programm aus App. B.2. Anschließend wird die SLD mit Programm (App. B.1.2) berechnet. Dabei wird auf den MATLAB FFT-Algorithmus zurückgegriffen.

3.5.5 Experimentelle Fließanomalien

Das untersuchte Polymer war PE-1 (siehe Tab. 3.1). Testemperatur war $\theta =$ 180 °C. Gemessen wurde bei $v_{kolben} = 0,02; 0,08; 0,5; 0,9\ mm/s$, dies entspricht scheinbaren Scherraten von $\dot{\gamma}_{ap} = 78; 314; 1963; 3534\ s^{-1}$. Die Messungen wurden mindestens dreimal durchgeführt. Dabei wurde mit Ovesamplingfrequenzen von 150 und 200 Hz aufgezeichnet. Die Versuchsdaten sind in Tab. 3.2 zu sehen. Es wurden Proben der Extrudate gesammelt und mit einem *HP scanjet 4890* Scanner mit 2400 dpi, dies entspricht einem Punktabstand von ca. 10 μm, eingescannt. Die beiden Rechner speicherten die Druckdaten aller vier Sensoren.

3.5.6 Erweiterung des Systems PE-1

Das untersuchte Polymer war wieder PE-1. Es wurde bei einer Temperatur von 155 °C gemessen. Außerdem wurde die scheinbare Scherrate bei 180

Tabelle 3.2: *Versuchsdaten der Messreihe I: Experimentelle Fließanomalien.*

Material	ϑ in °C	$\dot{\gamma}_{ap}$ in s^{-1}	v_{kolben} in mm/s	D_K in mm
PE-1	180	78	0,02	15
PE-1	180	314	0,08	15
PE-1	180	1963	0,5	15
PE-1	180	3534	0,9	15

Tabelle 3.3: *Versuchsdaten der Messreihe II: Erweiterung des System PE-1.*

Material	ϑ in °C	$\dot{\gamma}_{ap}$ in s^{-1}	v_{kolben} in mm/s	D_K in mm
PE-1	155	392	0,1	15
PE-1	155	589	0,15	15
PE-1	155	1610	0,41	15
PE-1	180	4712	1,2	15
PE-1	180	5890	1,5	15

°C erhöht. Die Versuchsdaten sind in Tab. 3.3 zu sehen. Es wurden Proben der Extrudate gesammelt und mit einem *HP scanjet 4890* Scanner mit 2400 dpi eingescannt. Die beiden Rechner speicherten die Druckdaten aller vier Sensoren.

3.5.7 Verallgemeinerung auf anderer Polymer Systeme

Die Messungen wurden auf die Polymere PE-2, PE-3 und PE-4 erweitert. Die Messungen wurden bei verschiedenem $v_{kolben} = [0,001; 1,5]$ mm/s durchgeführt. Ebenso wurde die Versuchstemperatur im Bereich $T = [155; 180]$ °C variiert. Die Oversamplingfrequenz bewegte sich in Bereichen von $\nu_{over} = [50\,; 200]$ Hz. Die Versuchsdaten sind in Tab. 3.4 zu sehen. Es wurden Proben der Extrudate gesammelt und mit einem *HP scanjet 4890* Scanner mit 2400 dpi eingescannt. Die beiden Rechner speicherten die Druckdaten aller vier Sensoren.

3.5.8 Stick-Slip Anomalie

Die Experimente dieses Abschnittes werden als Messreihe IV referenziert werden. Das Polymer PE-1 wurde bei $\vartheta = 180$ °C und $v_{kolben} = 0,08$ mm/s untersucht. Die eine Hälfte der Messungen wurden direkt nach einer Messung am
PE-2 durchgeführt. Bei der anderen Hälfte der Versuche wurde der Test-

Tabelle 3.4: *Versuchsdaten der Messreihe III: Verallgemeinerung auf anderer Polymer Systeme.*

Material	ϑ in °C	$\dot{\gamma}_{ap}$ in s^{-1}	v_{kolben} in mm/s	D_K in mm
PE-2	180	1963	0,5	15
PE-3	160	196	0,05	15
PE-3	160	1217	0,31	15
PE-4	180	1963	0,5	15
PE-4	180	3534	0,9	15

kanal gut gereinigt und mehrfach vorher mit PE-1 gemessen. Es wurden Proben der Extrudate gesammelt und mit einem *HP scanjet 4890* Scanner mit 2400 dpi eingescannt. Die beiden Rechner speicherten die Druckdaten aller vier Sensoren.

3.5.9 Optische Analyse der Instabilitäten

Aus den gespeicherten Bildern der Extrudate wurde ein 2000×500 Pixel großes Bildelement ausgeschnitten. Dies entspricht einer Bildgröße von 21× 5 mm. Diese Bildelemente wurden als Graustufenbilder gespeichert. Damit konnten mit der Programmcode aus App. B.1.1 in MATLAB bearbeitet werden. Dabei wurde die Bilder jeweils

- mit einem Laplace-Filter bearbeitet und
- über die gesamte Länge (2000 Pixel) wurde die FFT bestimmt.

Die Resultate der FFT lassen sich sowohl 2 dimensional, als auch 3 dimensional darstellen, vgl. Kap. 4.5.

Kapitel 4

Ergebnisse und Diskussion

Dieses Kapitel stellt die Ergebnisse der Druckmessungen und deren folgende Analyse vor. Die Methoden A und B werden auf jede der vier Messreihen angewandt und in jeweils einem gemeinsamen Kapitel behandelt. Dies dient dazu, dass die Resultate besser verglichen werden können, und auch um den Umfang dieser Arbeit zu reduzieren. Die Messreihe I liefert die erste Charakterisierung der Fließanomalien. Die Messreihen II und III helfen diese zu bestätigen oder zu erweitern. Die Messreihe IV wurde dem Ganzen angehängt, um den Ursprung der gefundenen Stick-Slip Anomalie aus der Messreihe I zu erforschen. Alle Messungen haben gemeinsam, dass die Intensität der Spitzen in der SLD selbst unter gleichen Versuchsbedingungen um mehrere Größenordnungen variieren kann. Deswegen ist sie als quantitative Größe nicht geeignet und wird im Weiteren nicht benutzt.

4.1 Fließanomalie

Die Messreihe I wurde am PE-1 durchgeführt. PE-1 zeigte das breiteste Spektrum an Fließanomalien. Die Fließanomalien in Abb. 4.1.a-d zeigten sich am Extrudat. Bei den Extrudaten sind zwei Auffälligkeiten sichtbar:

a. Die Fließanomalien in Abb. 4.1.c und d lassen sich nicht eindeutig einem Typus aus Kap. 2.2 zuordnen. Ihr Aussehen entspricht dem von Sharkskin. Die bestimmte Scherrate ist zu hoch im Vergleich zur gefunden Stick-Slip Fließanomalie. Gemäß Kap. 2.2 tritt Stick-Slip bei einer größeren Sherraten auf, als die Scherrate von Sharkskin.

b. Die Fließanomalie bei $v_{kolben} = 0,08\ mm/s$ trat nicht immer auf (siehe Kap. 4.4).

Die Auffälligkeit b wurde mit der Messreihe IV näher untersucht. Durch deren Ergebnis konnte Auffälligkeit a erklärt werden. Dies wird in Kap. 5 diskutiert. Für die Extrudate aus Abb. 4.1.c und d lies sich eine weitere

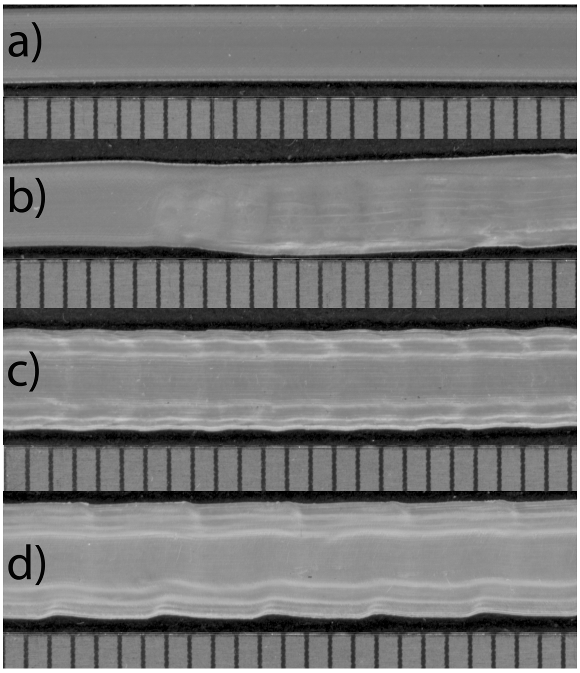


Abbildung 4.1: *Proben des Extrudates aus der Messserie I digitalisiert mit einem* HP Scanjet 4890 *mit 2400 dpi. a) glatt, b) Stick-Slip, c) Sharkskin 1 und d) Sharkskin 2. Oben im jeweiligen Bild (a-d) ist das Extrudat zu sehen darunter ist jeweils ein Lineal mit 1 mm Strichabstand.*

Beobachtung machen. Auf der einen Seite des Extrudats bildete sich eine schienenartige Deformation der Oberfläche aus. Dies ist in Abb. 4.2 zu sehen. Diese zusätzliche Fließanomalie ist ununterbrochen am gesamten Extrudat zu finden.

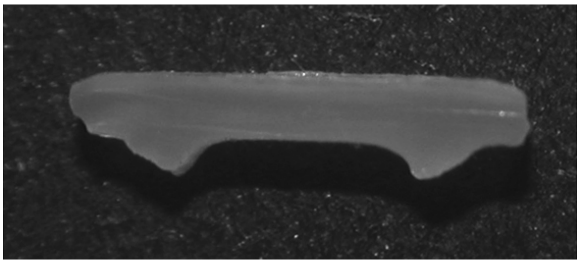


Abbildung 4.2: *Extrudat Querschnitt für PE-1 bei 180°C und 3534* s^{-1}.

Die Abb. 4.3- 4.6 zeigt die resultierenden Druck- und Autokorrelationsfunktion, sowie die beiden spektralen Leistungsdichten bei verschiedenen Kolbengeschwindigkeiten (siehe Tab. 4.1) für die Sensorposition 1, 2 und 3 von einer Messung.

Aus Abb. 4.3.a- 4.6.a, dem zeitabhängigen Druck, ist zu erkennen, dass das glatte Extrudat eine sehr geringe Standardabweichung des Drucks $\bar{\sigma}_i < 0,3$ bar bei 190 bar in der Düse hat. Die Fluktuation um den zeitabhängigen Druck ist um eine weitere Größenordnung kleiner. Die Standardabweichung für Sharkskin (Abb. 4.5.a) ist um einen Faktor 3 größer. Dieser Wert liegt nahe dem von Sharkskin 2 aus Abb. 4.6.a. Zwischen diesen Werten und dem von Stick-Slip liegt ein Faktor von ca. 7. Zudem ist Stick-Slip durch seinen deutlich sichtbaren oszillierenden Druck gekennzeichnet (siehe Abb. 4.4.a).

Die SLD des zeitabhängigen Drucks ist in Abb. 4.3.b- 4.6.b aufgetragen. Das glatte Extrudat zeigt in der SLD (Abb. 4.6.b) keine markanten Frequenzanteile. Die einzige erkennbare Spitze liegt bei 50 Hz, was der elektrischen Netzfrequenz entspricht. Diese Spitze taucht in allen Messungen auf. In der SLD der Stick-Slip Fließanomalie aus Abb. 4.4.b zeigen sich keine weiteren Frequenzanteile. Qualitativ ist ihre Kurve mit der des glatten Extrudats gleich. Die Sharkskin Fließanomalien sind sich qualitativ ähnlich. In Abb. 4.5.b und 4.6.b ist zu sehen, dass bei beiden jeweils zwei erkennbare Spitzen vorhanden sind. Die Spitze bei der niedrigeren Frequenz ist um den Faktor 10 kleiner als bei der höheren Frequenz. Es besteht ein quantitativer Unterschied zwischen den Spitzen von Sharkskin 1 und Sharkskin 2. Die beiden Spitzen bei Sharkskin 1 aus Abb. 4.5.b sind zu höheren Frequenzen verschoben. Die Spitzen bei Sharkskin 1 liegen bei ca. 10 und 20 Hz.

Die Abb. 4.3.c- 4.6.c zeigt die AKF. Der erste auffällige qualitative Unterschied ist die Steigung der AKF, wenn sie ihren Anfangspunkt (0,1) verlässt.

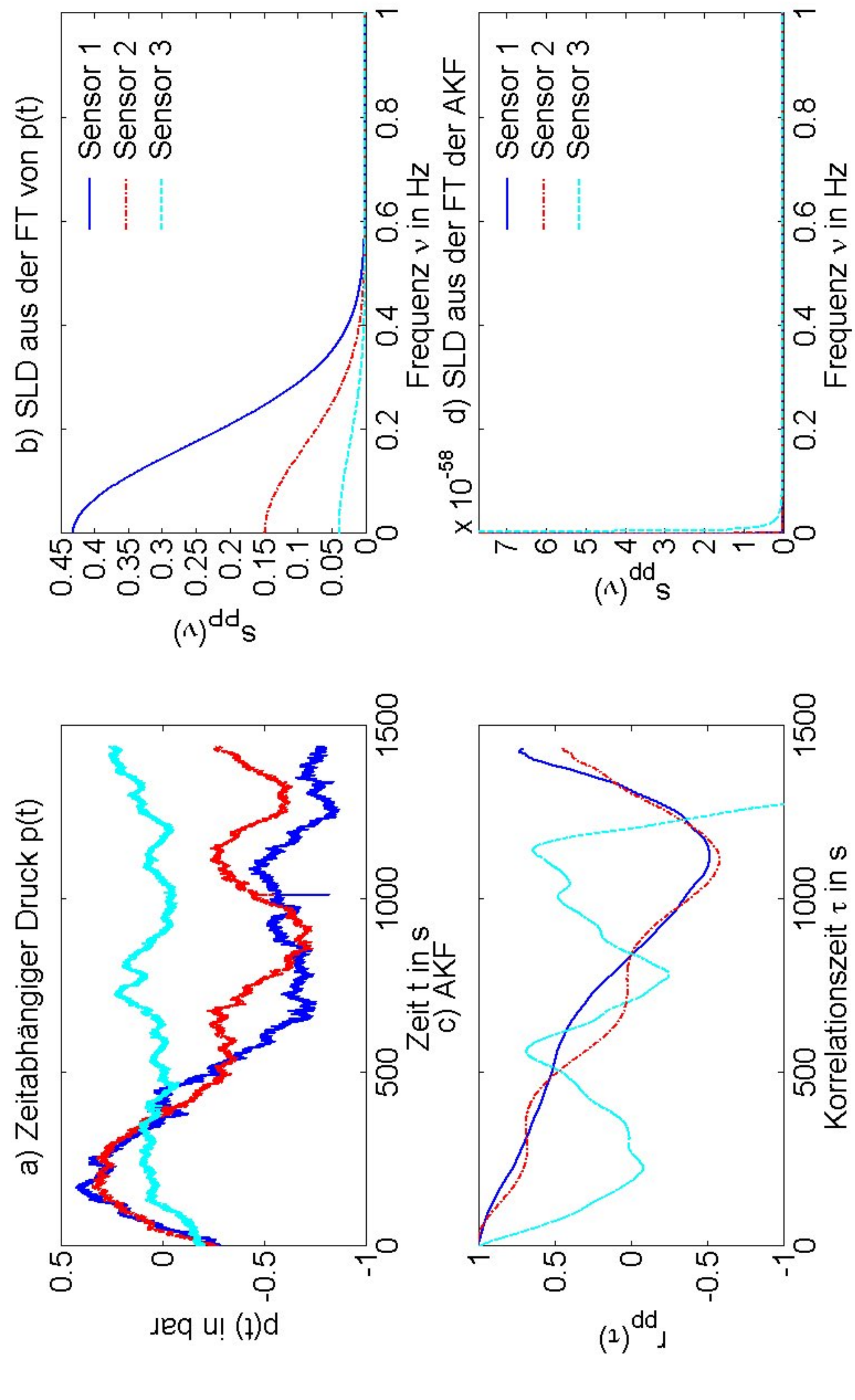


Abbildung 4.3: *Resultate für PE-1 bei 180 °C und 78,5 s^{-1} für alle drei Sensorpositionen. Für das glatte Extrudat ist dargestellt a) der zeitabhängige Druck $p(t)$, b) die SLD $s_{PP}(\nu)$ aus $p(t)$, c) die AKF $r_{pp}(\tau)$ und d) die SLD $s_{pp}(\nu)$ aus der AKF. Die Farben der Kurven bedeuten: (blau) Sensor 1, (rot) Sensor 2 und (türkis) Sensor 3.*

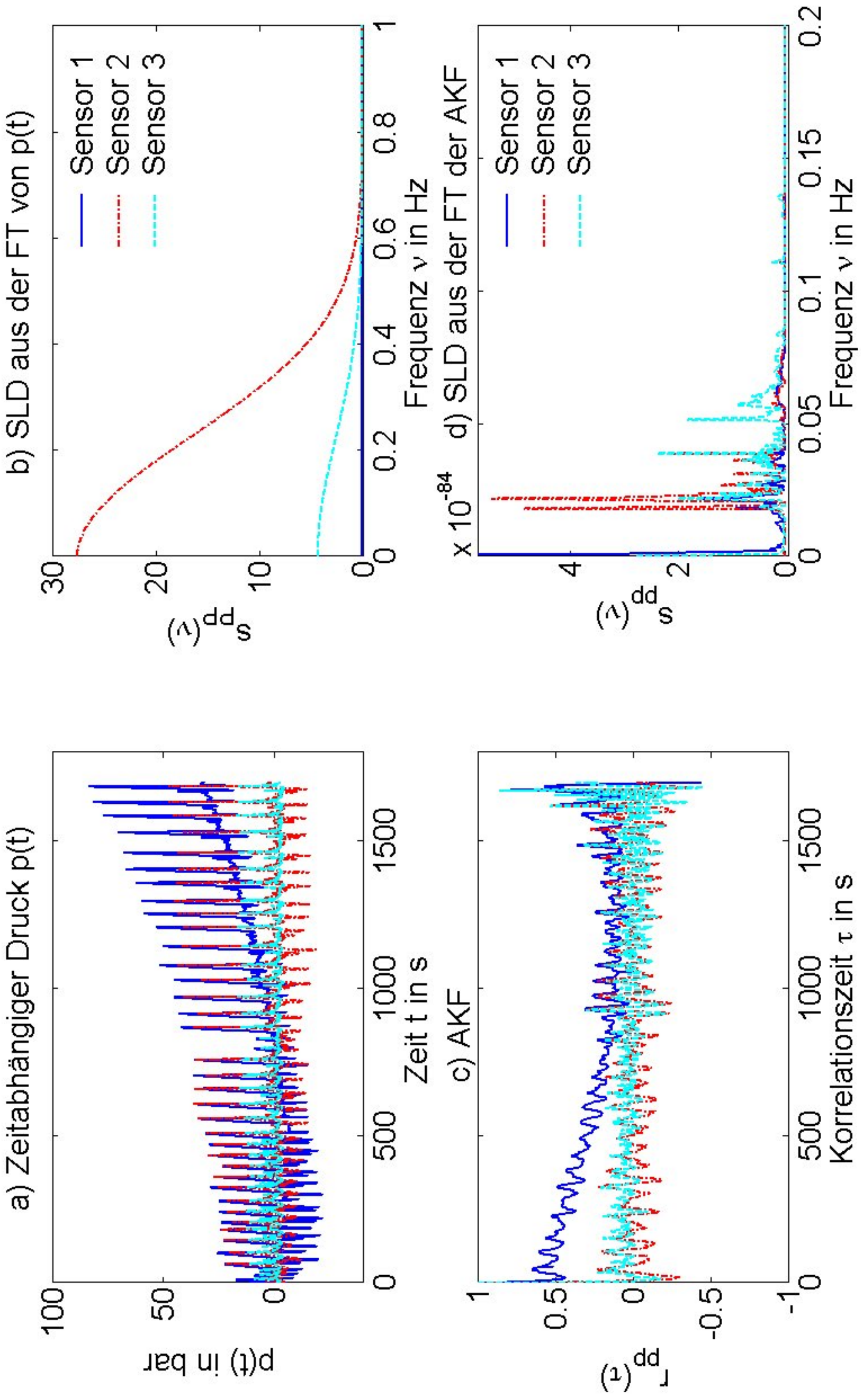


Abbildung 4.4: *Resultate für PE-1 bei 180 °C und 314,2 s^{-1} für alle drei Sensorpositionen. Für die Stick-Slip Fließanomalie ist dargestellt a) der zeitabhängige Druck $p(t)$, b) die SLD $s_{PP}(\nu)$ aus $p(t)$, c) die AKF $r_{pp}(\tau)$ und d) die SLD $s_{pp}(\nu)$ aus der AKF. Die Farben der Kurven bedeuten: (blau) Sensor 1, (rot) Sensor 2 und (türkis) Sensor 3.*

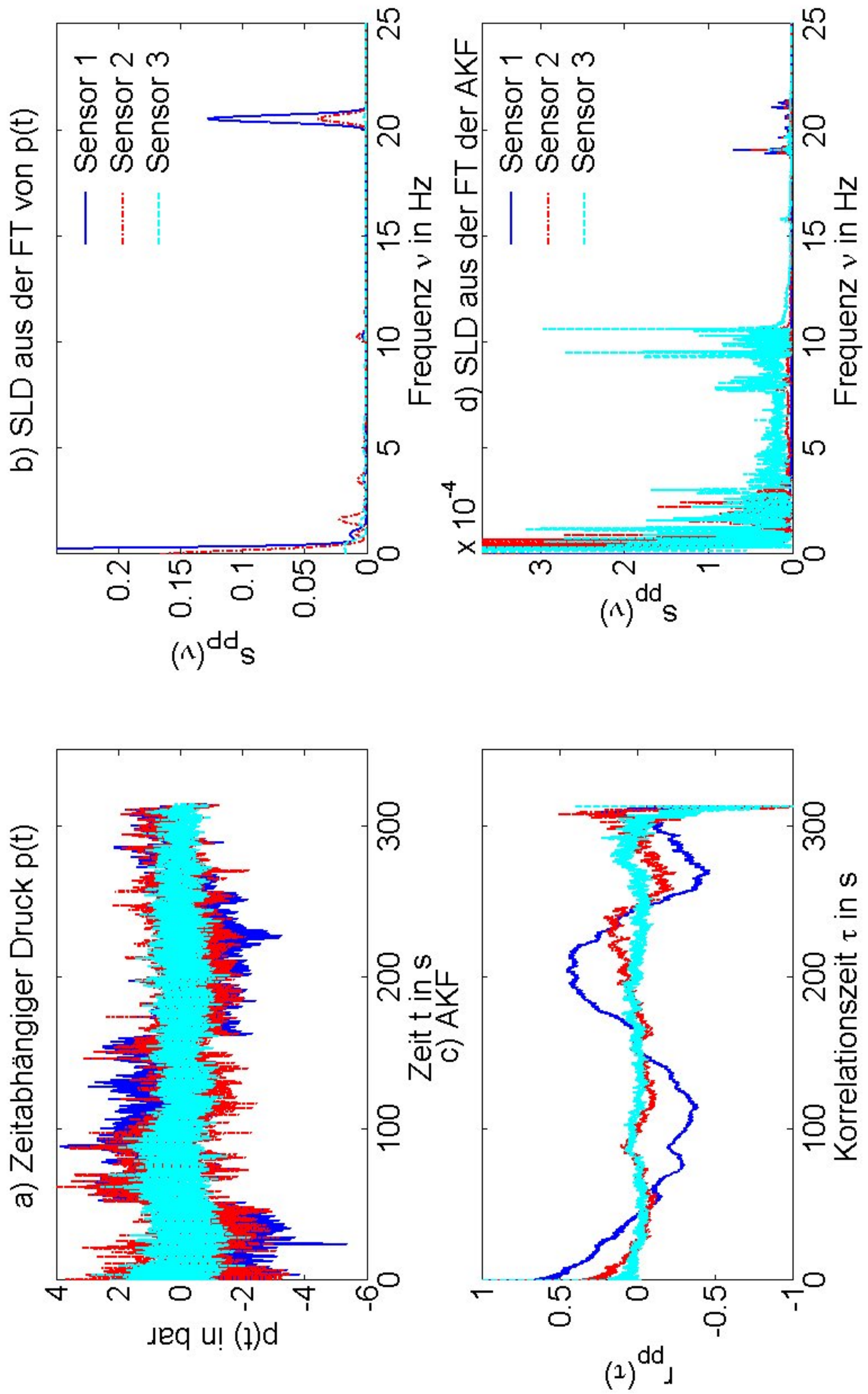


Abbildung 4.5: *Resultate für PE-1 bei 180 °C und 1963 s^{-1} für alle drei Sensorpositionen. Für die Sharkskin 1 Fließanomalie ist dargestellt a) der zeitabhängige Druck $p(t)$, b) die SLD $s_{PP}(\nu)$ aus $p(t)$, c) die AKF $r_{pp}(\tau)$ und d) die SLD $s_{pp}(\nu)$ aus der AKF. Die Farben der Kurven bedeuten: (blau) Sensor 1, (rot) Sensor 2 und (türkis) Sensor 3.*

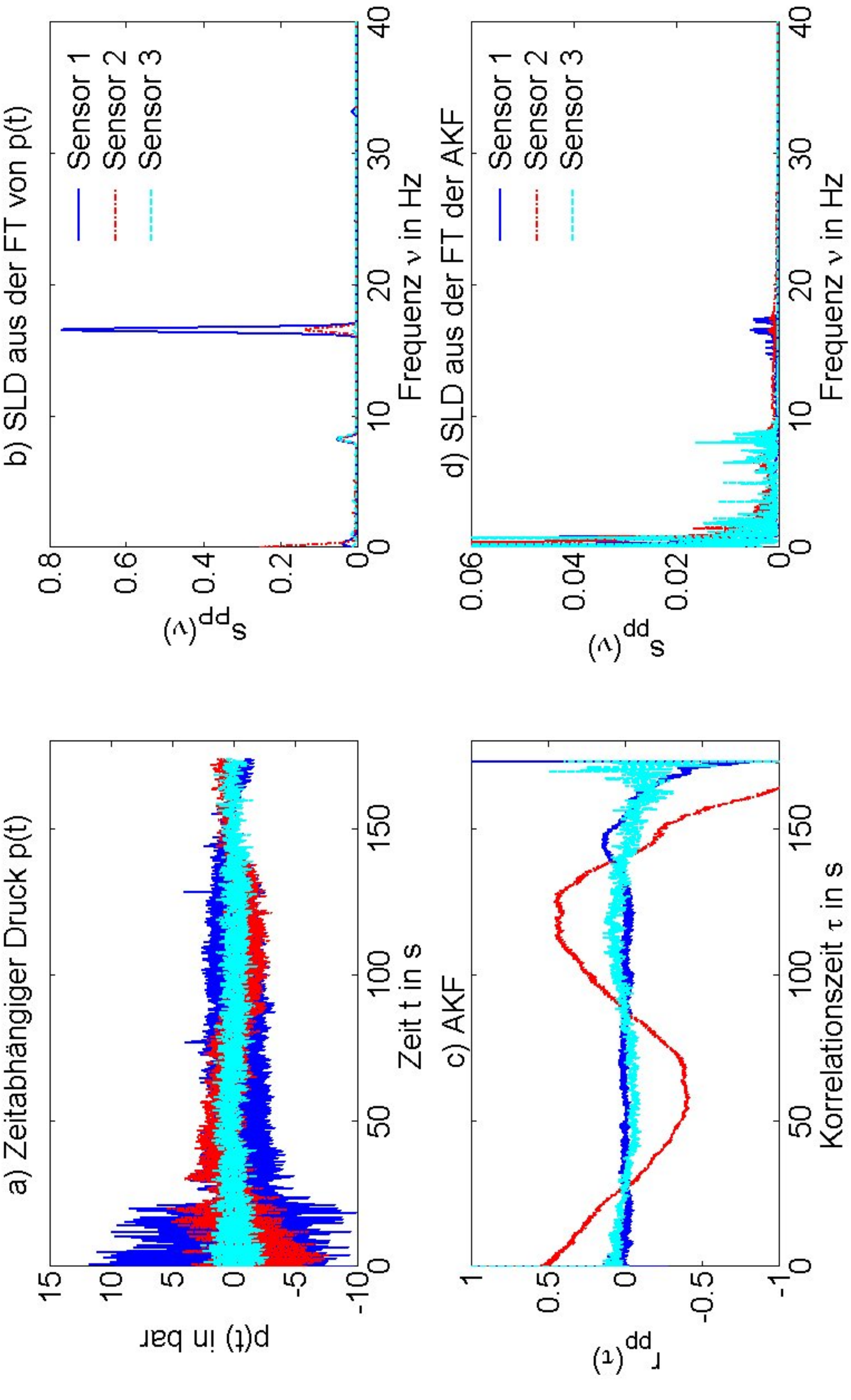


Abbildung 4.6: *Resultate für PE-1 bei 180 °C und 3534 s^{-1} für alle drei Sensorpositionen. Für die Sharkskin 2 Fließanomalie ist dargestellt a) der zeitabhängige Druck $p(t)$, b) die SLD $s_{PP}(\nu)$ aus $p(t)$, c) die AKF $r_{pp}(\tau)$ und d) die SLD $s_{pp}(\nu)$ aus der AKF. Die Farben der Kurven bedeuten: (blau) Sensor 1, (rot) Sensor 2 und (türkis) Sensor 3.*

Dabei hebt sich die AKF des glatten Extrudats (Abb. 4.3.c) deutlich von denen der anderen drei ab. Ihre Anfangssteigung ist wesentlich flacher. Sie braucht ca. 10 s bis sie den Wert 0,8 erreicht[1]. Die anderen drei erreichen ihn innerhalb von weniger als 1,5 s. Außerdem ist allen AKF gemein, dass sie nach dem Überschreiten der halben maximalen Korrellationszeit zunehmend verrauschter werden und teils auch Werte annehmen, die außerhalb des Wertebereichs der AKF liegen. Dies ist bedingt durch die abnehmende Anzahl an Punkten, die von der DAKF verwendet wird (vgl. Gl. 2.51). Dadurch bekommt das Messrauschen größeren Einfluss auf die AKF. Die Abb. 4.4.c zeigt die AKF von Stick-Slip. In ihr ist eine periodische Oszillation zusehen. Die Sharkskin 2 Fließanomalie zeichnet sich in ihrer AKF in Abb. 4.6.c durch einen schnellen Abfall ($\tau_0 = 10$ s) auf die τ-Achse ab. Danach oszilliert sie mit einer max. Amplitude von 0,2 um die τ-Achse. Am Ende weitet sie sich durch den Einfluss des Rauschens auf. Die AKF von Sharkskin 1 in Abb. 4.6.c sieht qualitativ gleich aus. Ihre Abfallszeit auf Null ist mit $\tau_0 = 25$ s größer als bei Sharkskin 2.

Die SLD, die über die AKF bestimmt wurde, ist in den Abb.4.3.d- 4.6.d dargestellt. Im Vergleich zu der SLD aus dem Druck ist sie verrauschter. Abb. 4.3.d ist die SLD des glatten Extrudats. Sie zeigt, wie die SLD in Abb. 4.3.b, keine Frequenzanteile in Form von weiteren Spitzen. Die Stick-Slip Fließanomalie zeigt in ihrer SLD in Abb. 4.4.d eine Spitze bei sehr niedrigen Frequenzen (0,025 Hz). In Abb. 4.5.d ist bei 9 Hz eine 4 Hz breite Erhöhung der Frequenz zu sehen. Dies kann eine breite Spitze sein, dies bedarf weiterer Diskussion in Kap. 5. Sharkskin 2 weist zwei klare Spitzen bei 9 und 17 auf.

Die Tab. 4.1 fasst die wichtigsten Kenngrößen und Merkmale aus dem hier Besprochenen zusammen.

[1] Dieser Wert ist gewählt, um die Anfangssteigung zu charakterisieren.

Tabelle 4.1: *Charakteristische Mess- und Analysedaten zu den Messungen aus Abb. 4.3- 4.6. Die Swell ist die Aufweitung des Extrudatstrangs nach dem Düsenaustritt.*

Eigenschaft	glatt	Stick-Slip	Sharkskin 1	Sharkskin 2
T in °C	180	180	180	180
v_{kolben} in mm/s	$0,02$	$0,08$	$0,50$	$0,90$
$\dot{\gamma}_{ap}$ in s^{-1}	$78,5$	$314,2$	$1963,5$	$3534,3$
$\nu_{Nyquist}$ in Hz	75	75	75	75
Die Swell	3,5	3,5	4,3	4,8
zeitabhängigen Druck				
$\bar{\sigma}_1$ in bar	0,3	14	0,86	1,17
$\bar{\sigma}_2$ in bar	0,27	7,7	0,71	0,72
$\bar{\sigma}_3$ in bar	0,07	3,5	0,52	0,42
AKF				
τ_0 in s	230	9,7	–	–
$\bar{\tau}_{0,8}$ in s	9,5	1,1	0,0175	0,004
SLD aus Druck				
Frequenz ν_i in Hz	50	–	$1,88$ und $19,6$	$8,25$ und $16,25$
SLD aus AKF				
Frequenz ν_i in Hz	–	$0,023$	9 und $19,73$	$17,44$

Die Ergebnisse für verschiedene Messungen unter den gleichen Bedingungen ist in Abb. 4.7 und 4.8 zu sehen. Es ist erkennbar, das die Druckkurven qualitativ gut zueinander passen.

Die Abb. 4.7 vergleicht die Messergebnisse bei 0,50 mm/s und 180 °C für die Sensorposition 1. Abb. 4.7.a zeigt, dass der zeitabhängige Druck der Messungen 2-4 gut zueinander passt. Die Messung 1 hat einen stärker oszillierenden Druck, als die anderen drei Messungen. Dies kann daran liegen, dass während dieser Messung die Drift stärkere nichtlineare Terme zeigte. Diese konnte durch die Aufbereitung der Daten nicht ausgeglichen werden. Der Unterschied zwischen den Messungen 2-4 und der Messung 1 zeigt sich auch in der SLD in Abb. 4.7.b. Während die Messungen 2-4 eine Spitze bei ca.19 Hz haben, ist die Spitze bei der Messung 1 um 2 Hz nach rechts verschoben. Alle anderen Spitzen der Messung 1 sind in gleicher Weise verschoben.

Die Abb. 4.7.c und d ist die AKF und die SLD aus der AKF für die selbe Messung und Sensorpostion. Im Vergleich der AKF aus Abb.4.7.c ist eine qualitative Ähnlichkeit aller Messungen innerhalb der ersten 50 s zu sehen. Dies spiegelt sich auch in einer guten Übereinstimmung in der SLD aus Abb. 4.7.d wider. Die Spektren der Messung 1 scheinen hier nicht verschoben zu sein. Dies liegt daran, dass die Spektren der SLD, die über die AKF berechnet wurden, wesentlich verrauschter und breiter sind als die, der direkt aus den Druckdaten bestimmten SLD. Deshalb fällt keine Verschiebung der Frequenz um ca. 2 Hz auf.

Für die gleichen Versuchsbedingungen ergeben sich ähnliche Ergebnisse für die anderen Sensorposition.

Die Abb. 4.8 zeigt für die Messung bei 0,9 mm/s und 180° C an der Sensorpostion 3 den Vergleich zwischen zwei Messungen. Die Messung 1 wurde mit 100 Hz abgetastet und die Messung 2 mit 75 Hz. Für größere Scherraten ist anzunehmen, dass höhere Frequenzen auftreten. Liegen diese Frequenzen über der Nyquist Frequenz treten sie bei niedrigeren Frequenzen im Spektrum auf. Dies wird als Aliasing bezeichnet und wurde in Kap. 2.3 erläutert. Die Spitze der Messung 1 ist um 1,5 Hz nach rechts verschoben. Die Richtung würde für Aliasing entsprechen, aber die Größe der Verschiebung ist zu gering, um vom Aliasing her zu rühren. Für einen Unterschied in der Abtastfrequenz von 25 Hz müsste eine Frequenzverschiebung von ca. 24 Hz auftreten. Es scheint sich um Abweichungen zu handeln, die durch die Messung bestimmt sind. In der AKF und deren SLD, die in Abb. 4.8.c und d zu sehen sind, ist diese Verschiebung nicht mehr zu sehen. Dies liegt wieder an der Aufweitung der Spitzen. Die SLD in Abb. 4.8.b sind qualitativ ähnlich. Durch das Messen mit höheren Abtastfrequenzen gibt es keine signifikante Verschiebung. Damit ist gesichert, dass die gemessenen Frequenzen in diesem Bereich liegen.

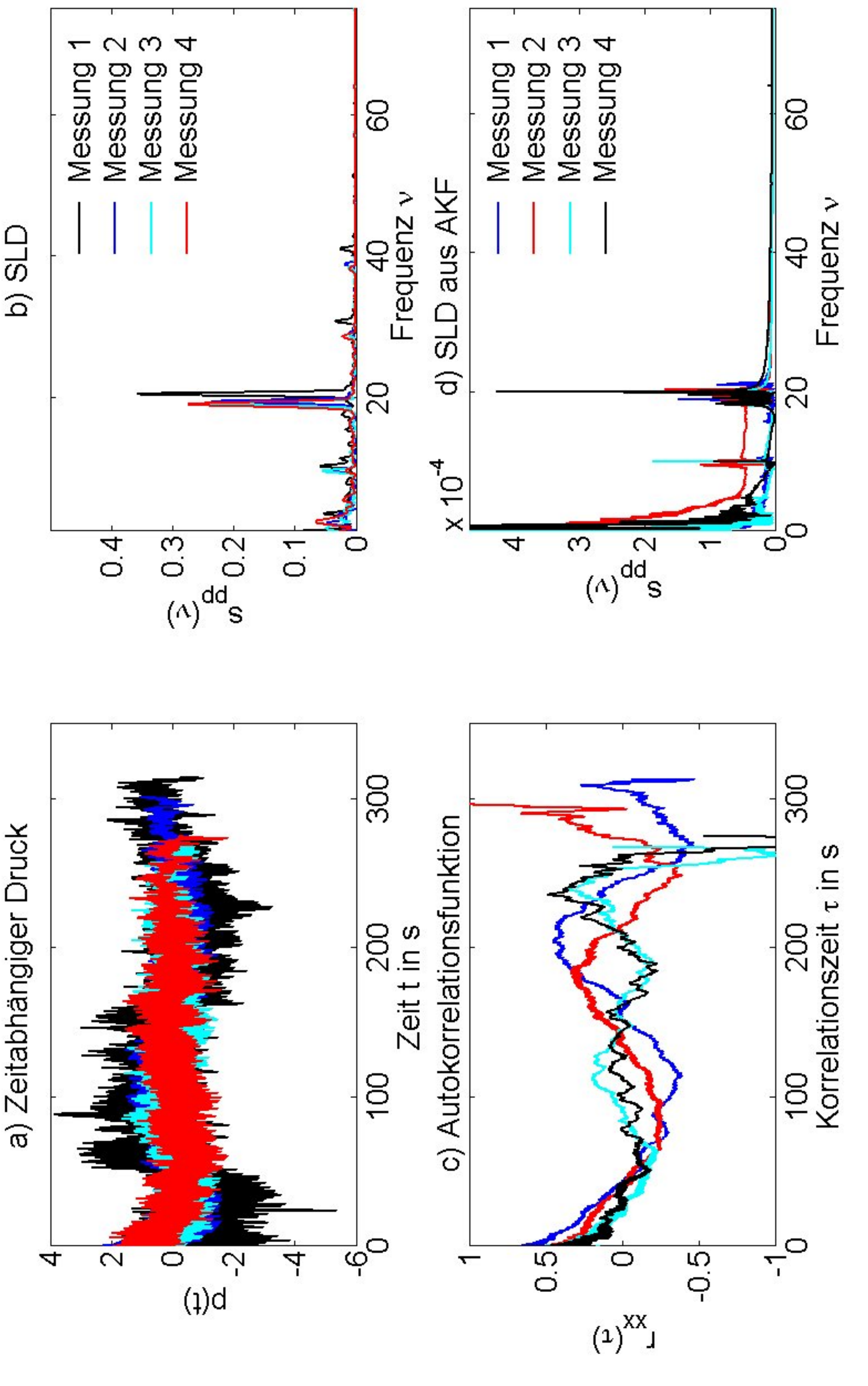


Abbildung 4.7: *Reproduzierbarkeit der Messung für vier Messungen mit PE-1 bei* $\vartheta = 180°C$ *und* $\dot{\gamma}_{app} = 1963\ s^{-1}$. *Die Sensorposition ist 1. a) Druckkurven, b) SLD direkt aus dem Druck, c) AKF und d) SLD aus der AKF.*

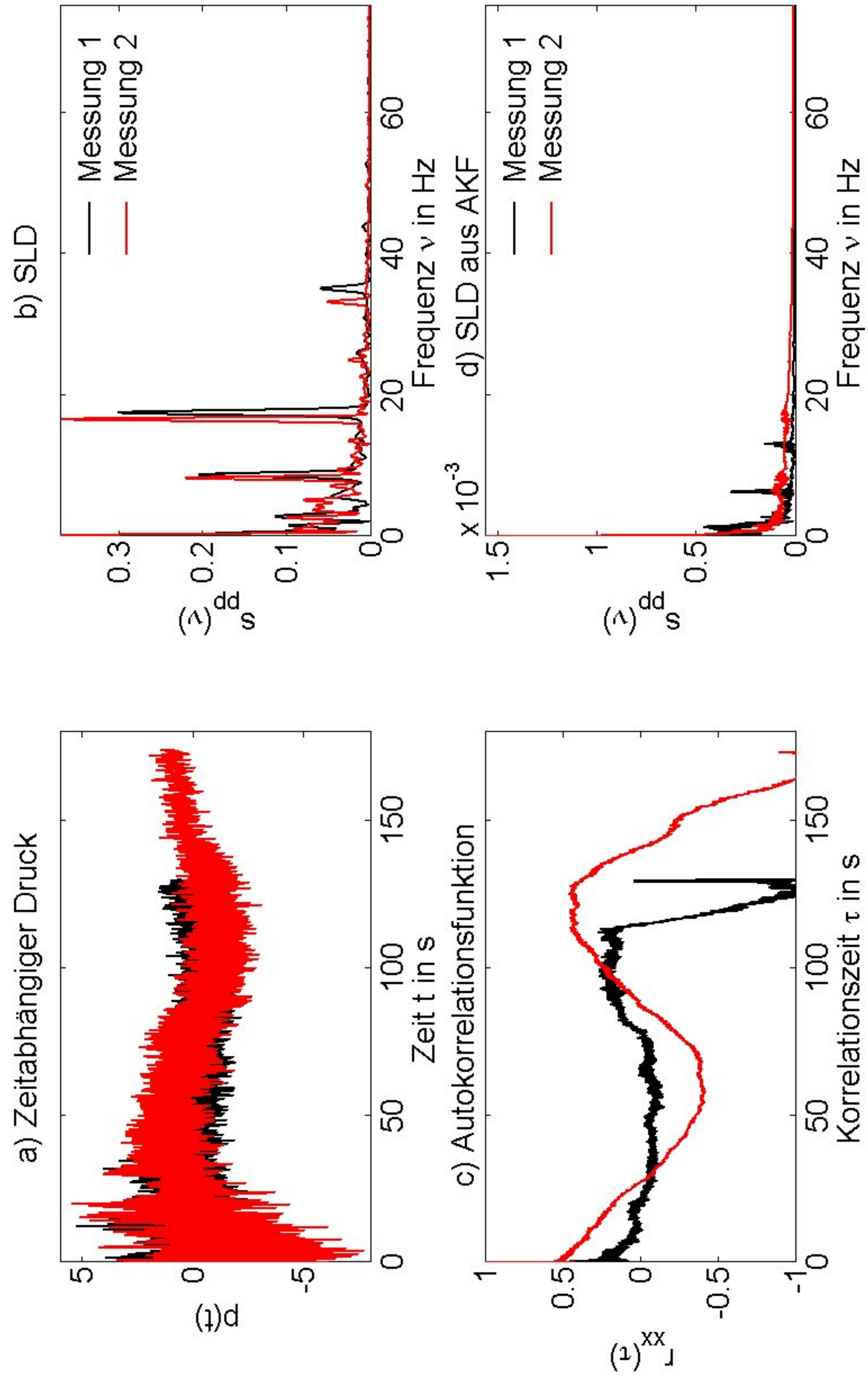


Abbildung 4.8: *Vergleich von zwei Messung bei denen verschiedenen Abtatsfrequenzen benutzt wurden. PE-1 bei* $\vartheta = 180°C$ *und* $\dot{\gamma}_{app} = 3524\ s^{-1}$*. Die Sensorposition ist 3. a) Druckkurven, b) SLD direkt aus dem Druck, c) AKF und d) SLD aus der AKF. Die Schwarze Kurve wurde mit 75 HZ und die blaue mit 100 Hz abgetastet.*

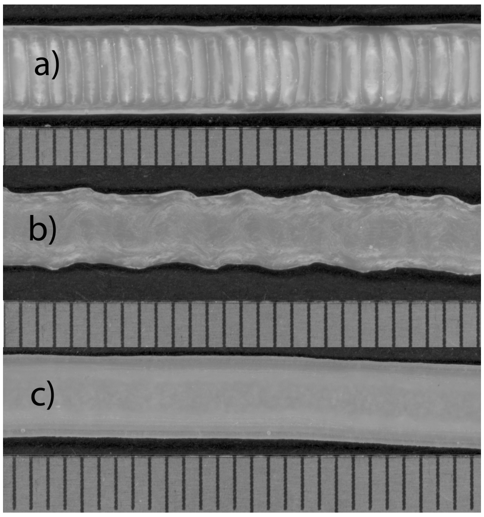


Abbildung 4.9: *Form des Extrudats von a) PE-1 bei* $\vartheta = 155$ °C *und* $\dot{\gamma}_{ap} = 393$ s^{-1}*, b) PE-1 bei* $\vartheta = 155$°C *und* $\dot{\gamma}_{ap} = 1610\ s^{-1}$ *und c) PE-1 bei* $\vartheta = 180$°C *und* $\dot{\gamma}_{ap} = 5890\ s^{-1}$*. Oben im Bild ist jeweils das Extrudat zu sehen darunter jeweils ein Lineal mit 1 mm Strichabstand.*

4.2 Erweiterung auf das System PE-1

Die Messreihe II erweitert die Ergebnisse der Messreihe I für das PE-1 auf

- die Temperatur ϑ =155 °C und
- höhere scheinbare Scherraten $\dot{\gamma}_{ap} = 5890\ s^{-1}$.

Bei der Messung für die niedrigere Temperatur von 155 °C ist eine neue Erscheinungsform des Extrudats zu beobachten. Sie ist in Abb. 4.9.a zu sehen. Das Extrudat zeigt eine deutlich sichtbare Periodizität. Diese Periodizität ist zu beobachten bei scheinbaren Scherraten im Bereich von 393- 589 s^{-1}. D.h. die Zuordnung zu einer Fließanomalie ist nicht eindeutig. Am besten passt sie in die Sharkskin Kategorie. In den Funktionsgraphen für Druck in Abb. 4.10 zeigt sie keine Periodizitäten. Die SLD des dritten Sensors zeigt eine charakteristische Spitze für die Größenordnung der Frequenz von 1 Hz. Dies ist in den Abb. 4.10.b und 4.10.d zu sehen. Ähnliches lässt sich für 589 s^{-1} beobachten. Die AKF aus der Abb. 4.10.c zeigt für den Sensor 3 einen schnellen Verlust der Korrelation mit $\tau_{0,8} \approx 0,1$ s. Dies ist auch bei 589 s^{-1} zu beobachten. Der Sensor 1 verliert deutlich langsamer an Korrelation $\tau_{0,8} \approx 25$ s. Die AKF der Sensoren 1 und 2 zeigen eine sinusförmige langwellige Schwingung ($T \approx 500$ s) mit der mittleren Amplitude von 0,2.

Als nächstes wurde bei gleicher Temperatur (155 °C) bei einer scheinbaren Scherrate von 1610 s^{-1} gemessen. Das Extrudat zeigt Schmelzbruch

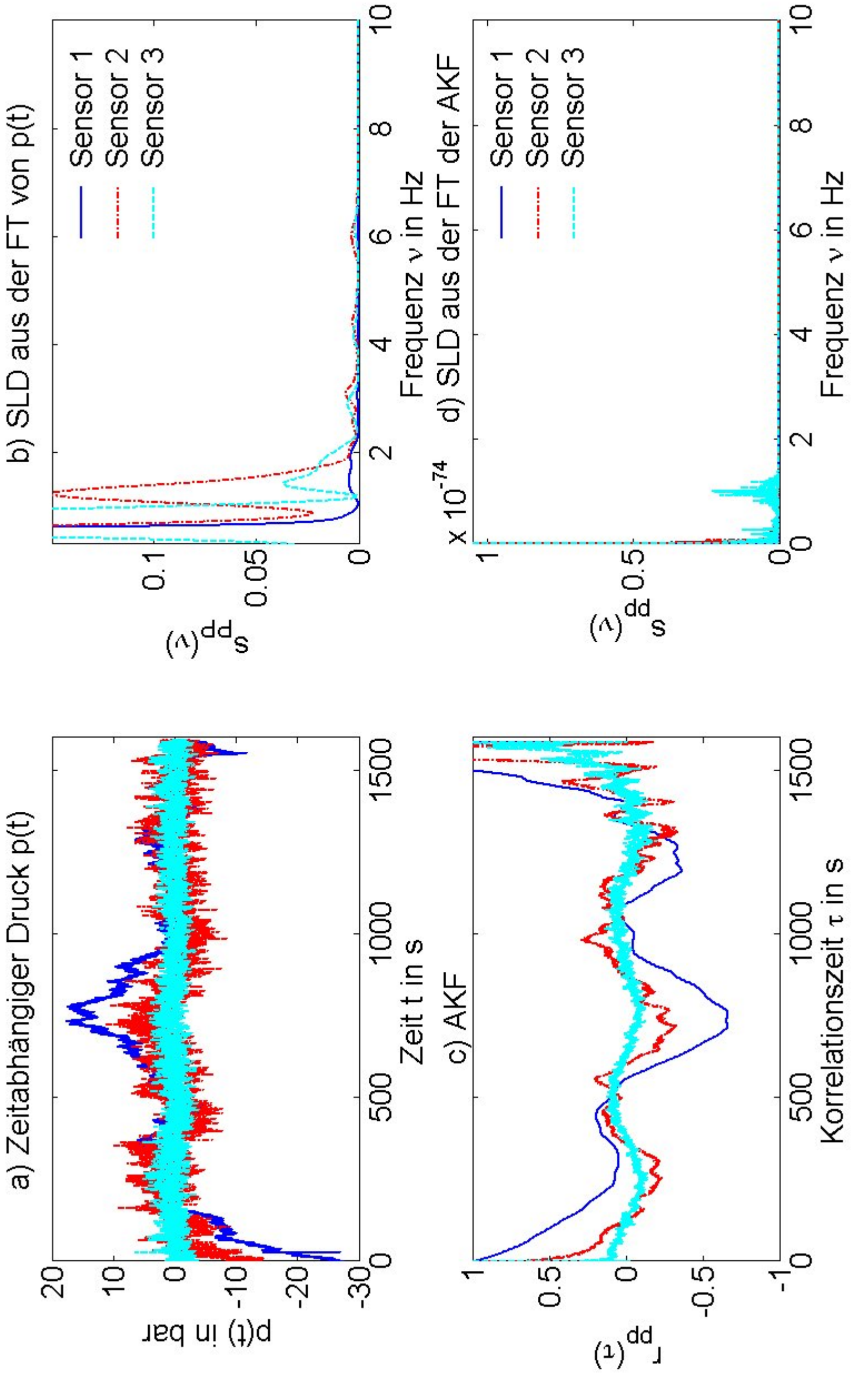


Abbildung 4.10: *PE-1 bei $\vartheta = 155$ °C und $\dot{\gamma}_{ap} = 393\ s^{-1}$: a) Druckkurven, b) SLD direkt aus dem Druck, c) AKF und d) SLD aus der AKF.*

(Melt-Fracture). Dies ist in Abb. 4.9.b abgebildet. Die SLD Abb. 4.11.b und 4.11.d haben jeweils eine breite Spitze in dem Frequenzbereich von 3-8 Hz. In der SLD aus Abb. 4.11.b ist die Spitze in mehrere Spitzen aufgespalten. Dies wird durch das Gauß Fenster verursacht. Die Spitze in der SLD aus Abb. 4.11.c ist nur von geringer Intensität. Sie ist als Merkmal zur klaren Charakterisierung schlecht geeignet. Da sie auch als breite 0 Hz Spitze gedeutet werden kann. In der AKF in Abb. 4.11.c sinkt die Korrelation aller Sensoren mit $\tau_{0,8} = 0,016$ s schnell ab. Danach bleibt die AKF in einem engen Band um die τ-Achse. Diese Fließanomalie ist klar als Schmelzbruch (Melt-Fracture) zu zuordnen. Die Resultate der Messungen bei $\dot{\gamma}_{ap} = 4712$ s^{-1} in Abb. 4.12 zeigen ein qualitativ gleiches Verhalten, wie die Resultate der Messung bei $\dot{\gamma}_{ap} = 5890$ s^{-1} aus Abb. 4.13. Das Extrudat ist in Abb. 4.9.c zu sehen.

Der Druck in den Abb. 4.12.a zeigt anfangs große Fluktuationen, die über die Dauer des Versuchs nachlassen. Dies ist auch bei der höheren Scherrate zu beobachten.

In der SLD aus dem Druck in der Abb. 4.12.b sind zwei Spitzen zu erkennen. In Abb. 4.12.b liegen sie bei ca. 7 Hz und 12 Hz. Für $\dot{\gamma}_{ap} = 5890$ s^{-1} in Abb. 4.13.b liegen sie bei ca. 4 Hz und 12 Hz. Dabei ist auffallend, dass der Versuch mit der höheren Frequenz seine Frequenzanteile bei niedrigeren Frequenzen hat. Dies ist contra intuitiv.

Ähnliches zeigt sich bei der SLD, die über die AKF bestimmt wurde. In Abb. 4.12.d sind zwei Spitzen bei ca. 4 Hz und 12 Hz zu erkennen. In Abb. 4.13.d für $\dot{\gamma}_{ap} = 5890$ s^{-1} ist eine breite Spitze bei 4 Hz zu erkennen. Die AKF in der Abb. 4.12.c zeigt ein qualitativ gleiches Verhalten, wie $\dot{\gamma}_{ap} =$ 5890 s^{-1}. Sie verlieren beide schnell an Korrelation ($\tau_{0,8} \approx 0,015$ s). Nach Erreichen der τ-Achse fluktuiert die AKF in einem Band von $\pm 0,15$ um diese.

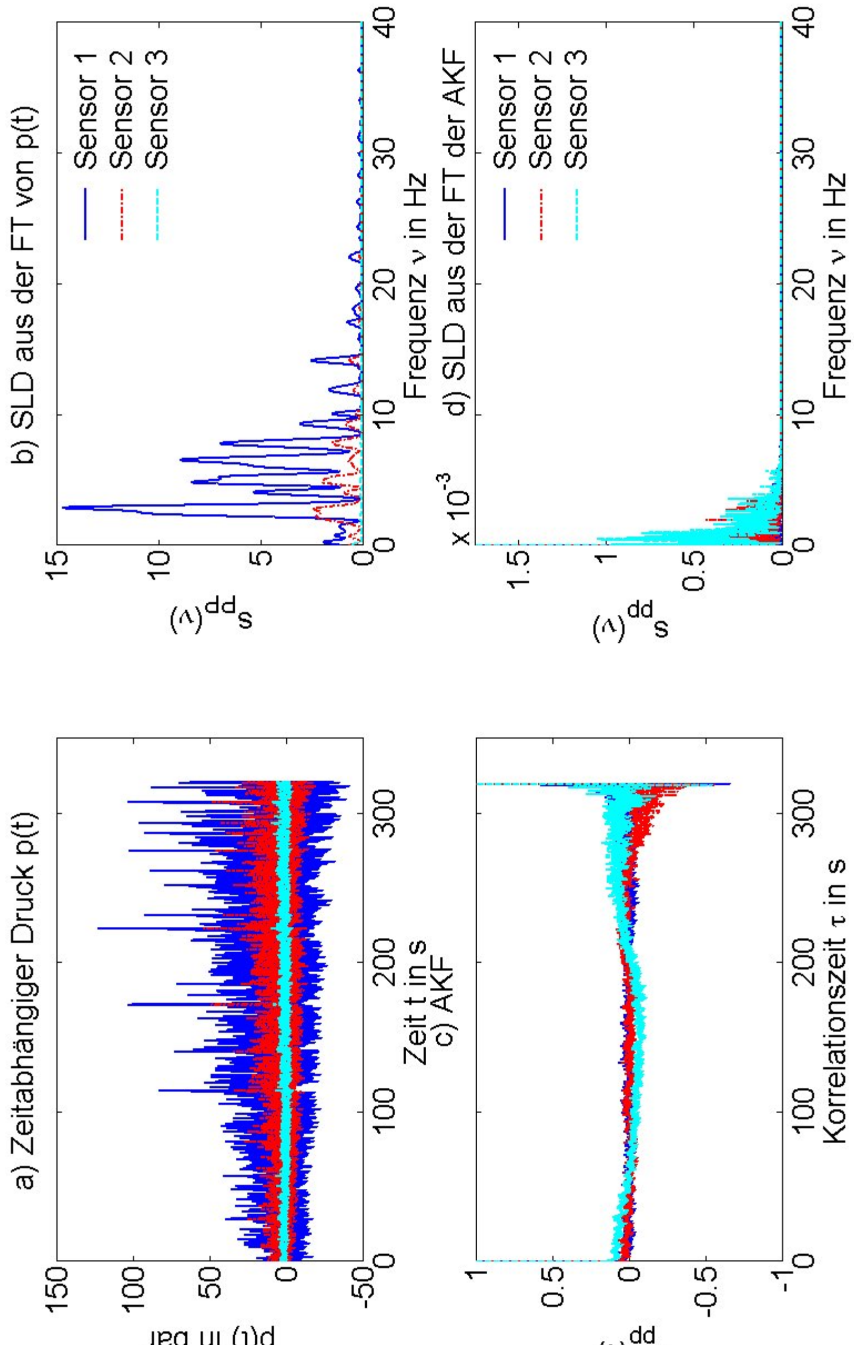


Abbildung 4.11: *PE-1 bei $\vartheta = 155$ °C und $\dot{\gamma}_{ap} = 1610\ s^{-1}$: a) Druckkurven, b) SLD direkt aus dem Druck, c) AKF und d) SLD aus der AKF.*

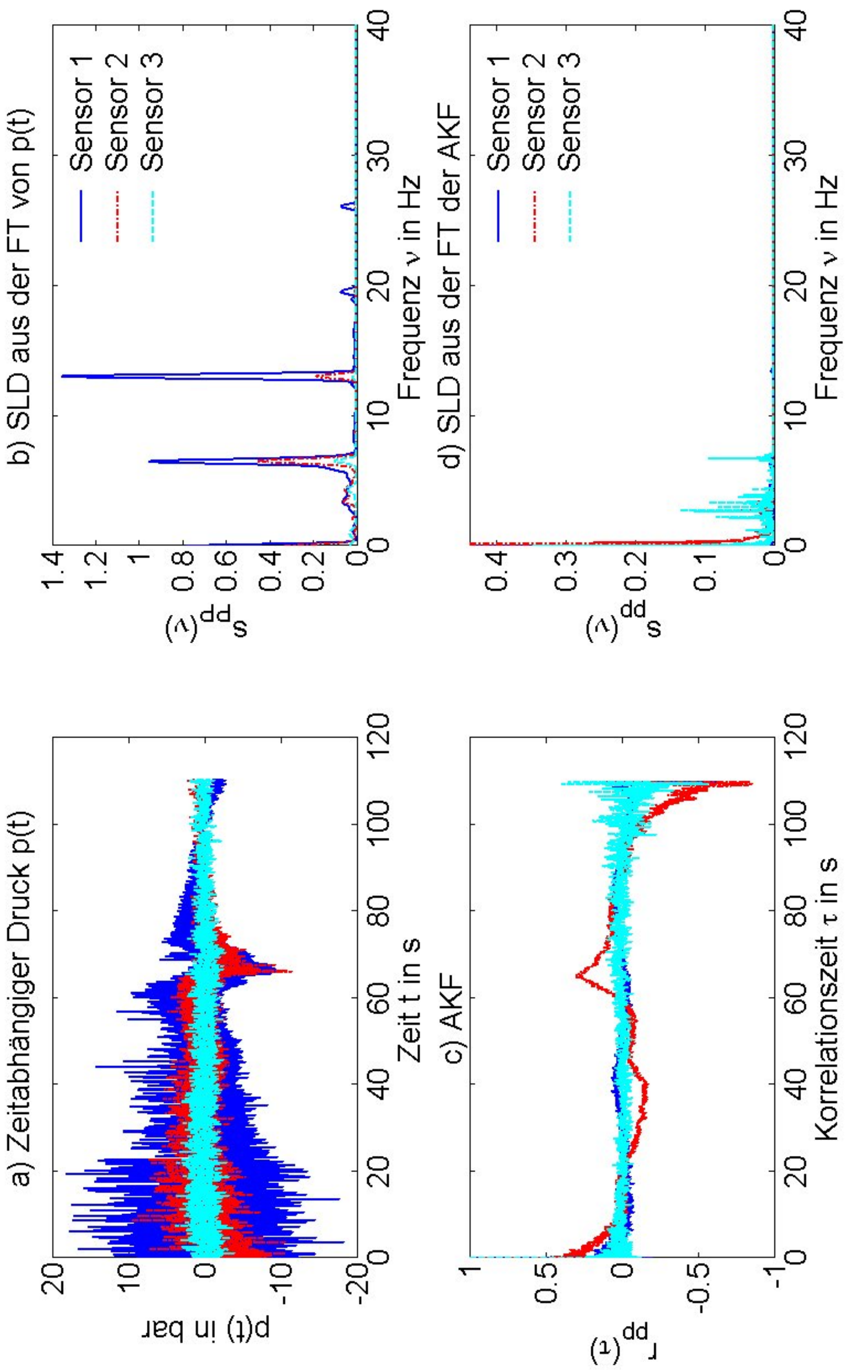


Abbildung 4.12: *PE-1 bei $\vartheta = 180°C$ und $\dot{\gamma}_{ap} = 4712\ s^{-1}$: a) Druckkurven, b) SLD direkt aus dem Druck, c) AKF und d) SLD aus der AKF.*

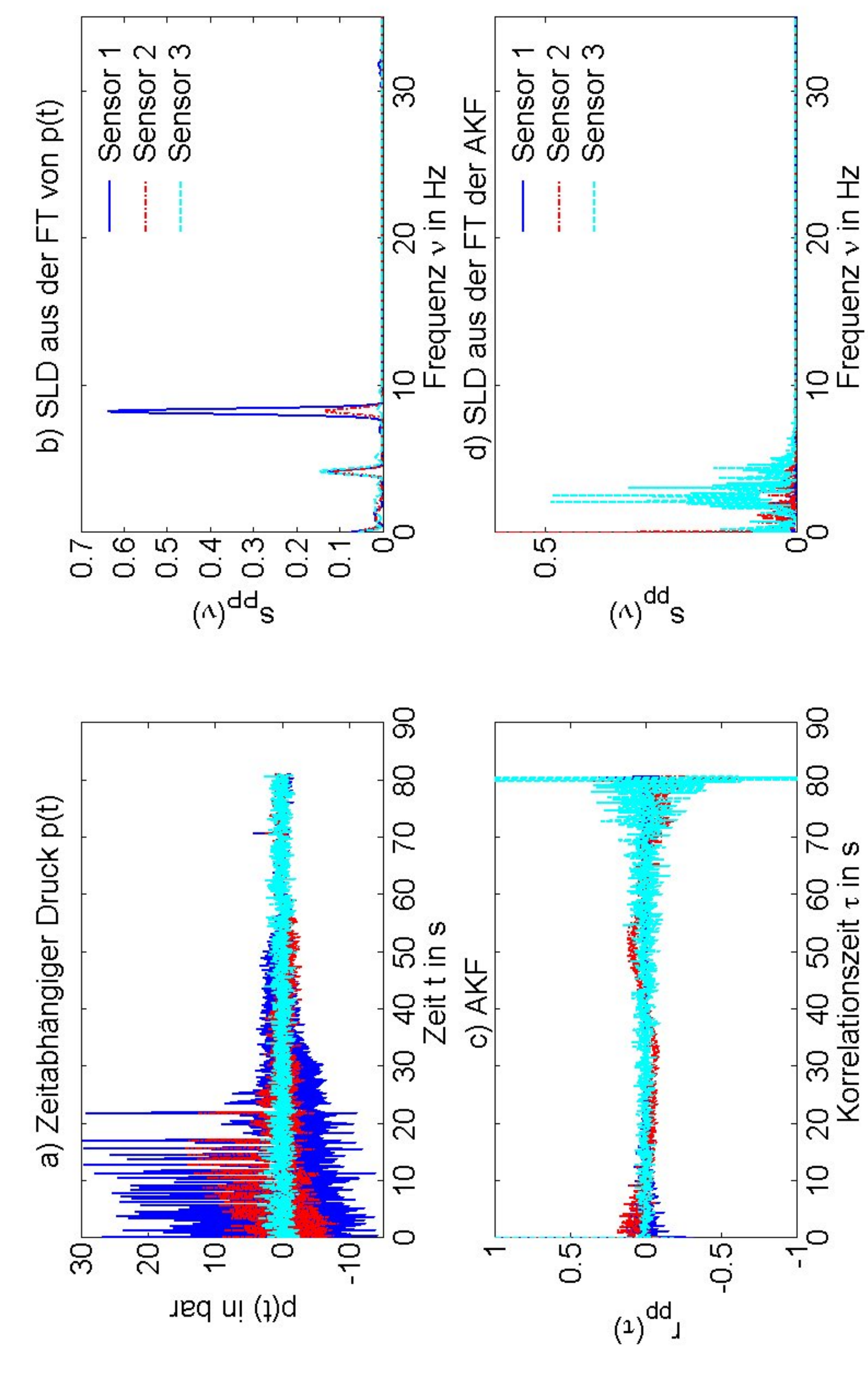


Abbildung 4.13: *PE-1 bei $\vartheta = 180°C$ und $\dot{\gamma}_{ap} = 5890\ s^{-1}$: a) Druckkurven, b) SLD direkt aus dem Druck, c) AKF und d) SLD aus der AKF.*

4.3 Verallgemeinerung auf andere Polymer Systeme

Die Messreihe III dient zur Untersuchung von Fließanomalien bei weiteren Polymeren. Im Rahmen der durchgeführten Experimente zeigte sich nur Sharkskin und Schmelzbruch (Melt-Fracture). Für das Polymer PE-3 wurden

a. Sharkskin bei 196 s^{-1} und

b. Sharkskin für 1217 s^{-1}

gefunden.

Die Versuche von a. charakterisieren sich durch eine starke periodische Oszillation des Druckes in Abb. 4.15.a und der AKF in Abb. 4.15.c. Diese beiden Argumente sprechen für Stick-Slip. In der SLD aus dem Druck in der Abb. 4.15.b zeigen sich Spitzen bei Frequenzen von ca. 1,5 Hz. Die SLD aus der AKF in Abb. 4.15.d zeigt keine Frequenzanteile. Optisch ist die Messung Sharkskin zu zuordnen (siehe Abb. 4.14.a).

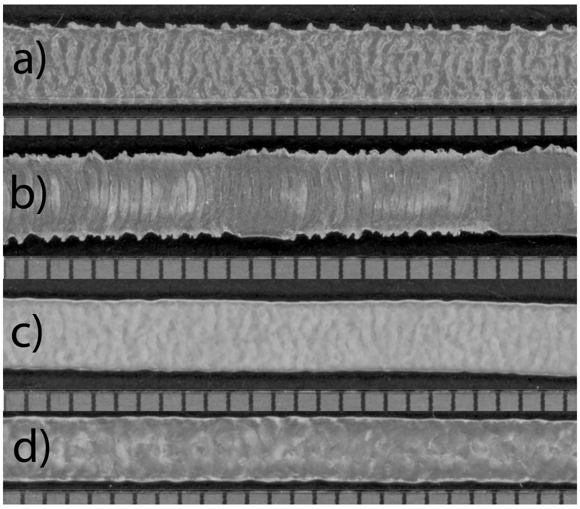


Abbildung 4.14: *Extrudatprobe von a) PE-3 bei $\vartheta = 160$ °C und $\dot{\gamma}_{ap} = 196\ s^{-1}$, b) PE-3 bei $\vartheta = 160$ °C und $\dot{\gamma}_{ap} = 1217\ s^{-1}$, c) PE-4 bei $\vartheta = 180$ °C und $\dot{\gamma}_{ap} = 1963\ s^{-1}$ und d) PE-2 bei $\vartheta = 180$ °C und $\dot{\gamma}_{ap} = 3534\ s^{-1}$. Jeweils oben im Bild ist das Extrudat zu sehen, darunter jeweils ein Lineal mit 1 mm Strichabstand.*

Die Versuche in b. zeigen ein klares Sharkskin Verhalten. Dies ist an dem Extrudat in Abb.4.14.b mit bloßem Auge zu erkennen. In der SLD in Abb. 4.16.b und 4.16.d ist jeweils eine Spitze mit einer Frequenz von 1,5 Hz erkennbar. Diese Spitze zeigt sich in den beiden SLD und auch bei wiederholter Messung. In Abb. 4.16.c ist AKF aufgetragen. Sie verliert schnell ($\tau_{0,08} < 0,015$ s) an Korrelation. Danach bleibt sie in einem Band von 0,1 um die τ-Achse.

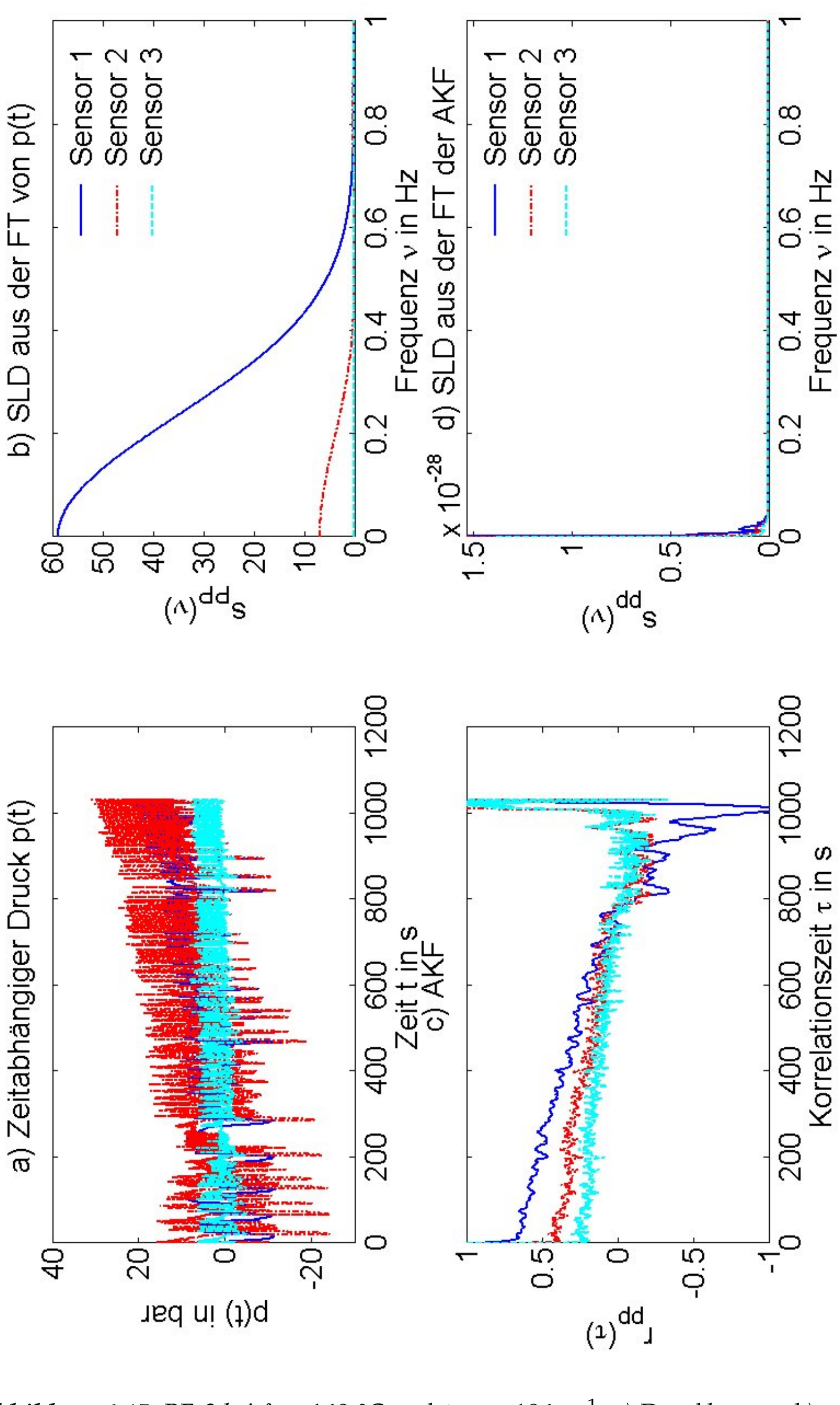


Abbildung 4.15: *PE-3 bei $\vartheta = 160$ °C und $\dot{\gamma}_{ap} = 196\ s^{-1}$: a) Druckkurven, b) SLD direkt aus dem Druck, c) AKF und d) SLD aus der AKF.*

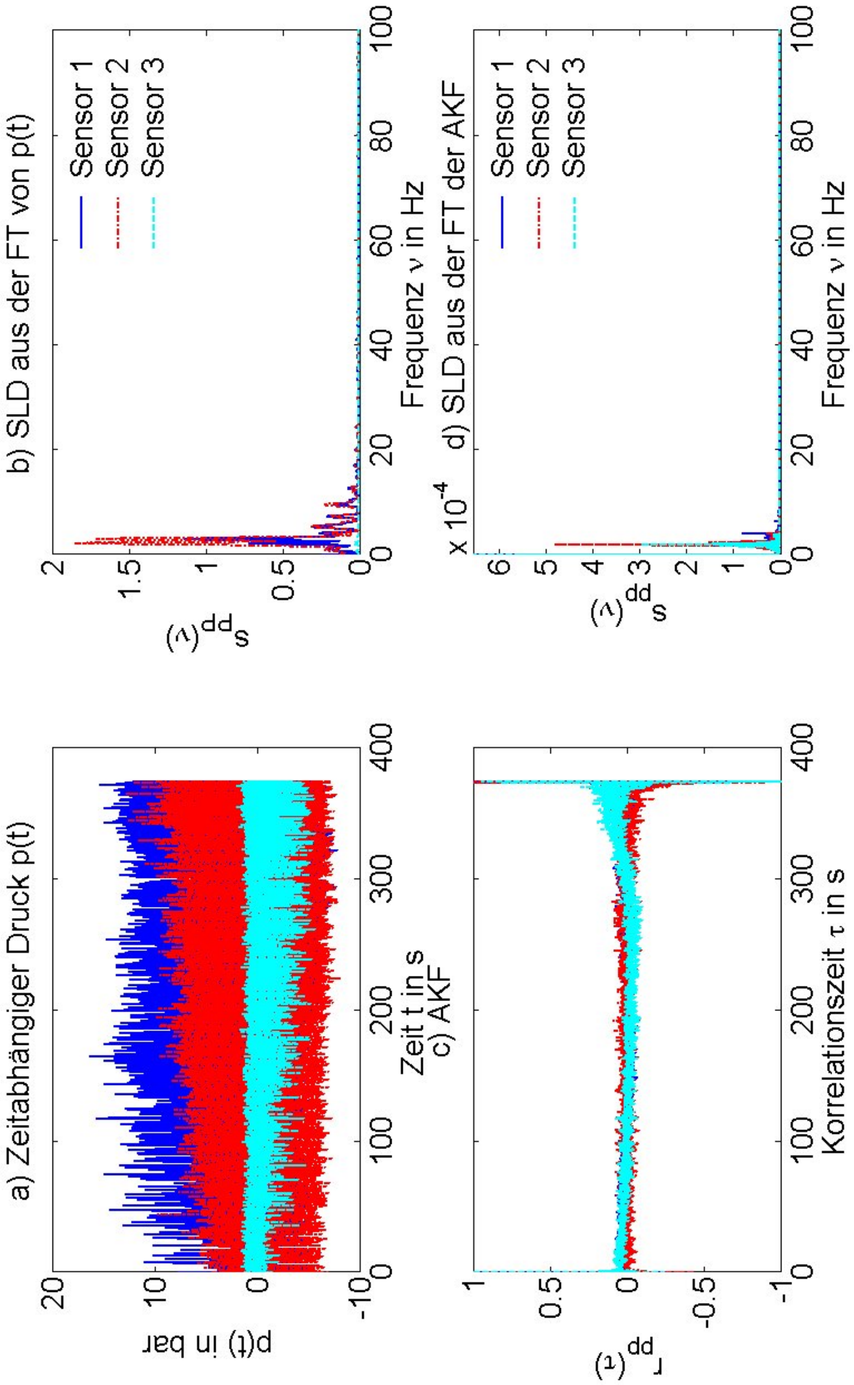


Abbildung 4.16: *PE-3 bei $\vartheta = 160$ °C und $\dot{\gamma}_{ap} = 1217\ s^{-1}$: a) Druckkurven, b) SLD direkt aus dem Druck, c) AKF und d) SLD aus der AKF.*

Die folgenden Messungen wurden an PE-4 durchgeführt bei 180 °C mit den scheinbaren Scherraten von

c. 1963 s^{-1} und

d. 3534 s^{-1}.

Das Extrudat der beiden Versuche c und d zeigt Melt Fracture. Dies ist in Abb. 4.14.c zu sehen. Beide Messungen zeigen qualitativ gleiches Verhalten. Die beiden SLD aus den Abb. 4.17.b und 4.17.d zeigen eine Spitze bei der Frequenz von 2 Hz. Die Spitze in den Abb. 4.17.b ist auf mehrere benachbarte Spitzen verteilt. Diese Verhalten ist durch das Gauß Fenster bedingt. Die AKF aus der Abb. 4.17.c nimmt schnell an Korrelation ab ($\tau_{0,8} < 0,04$ s). Im weiteren Verlauf fluktuiert sie in einem Band von $\pm$0,15 um die τ-Achse.

Abschließend wurde noch PE-2 auf seine Schmelzbruch (Melt-Fracture) Fließanomalie (Abb.4.14.d) untersucht. Diese stellte sich bei 180°C und 1963 s^{-1} ein. Die Resultate in den Abb. 4.18 sind qualitativ denen von zuvor besprochenen PE-4 ähnlich. In Abb. 4.18.b und 4.18.d ist eine 1 Hz breite Spitze in der SLD zu sehen.Ein schneller Verlust der Korrrelation mit $\tau_{0,8} \approx 0,07$ s ist in Abb. 4.18.c zu erkennen. Die AKF verbleibt in einem Band der Breite 0,15 um die τ-Achse. Innerhalb dieses Bandes ist eine langwellige ($T \approx 500$ s) kosinusförmige Oszillation für den Sensor 3 zu beobachten.

Diese Resultate für den Schmelzbruch bei PE-2 sind qualitativ gleich denen von Schmelzbruch bei PE-4.

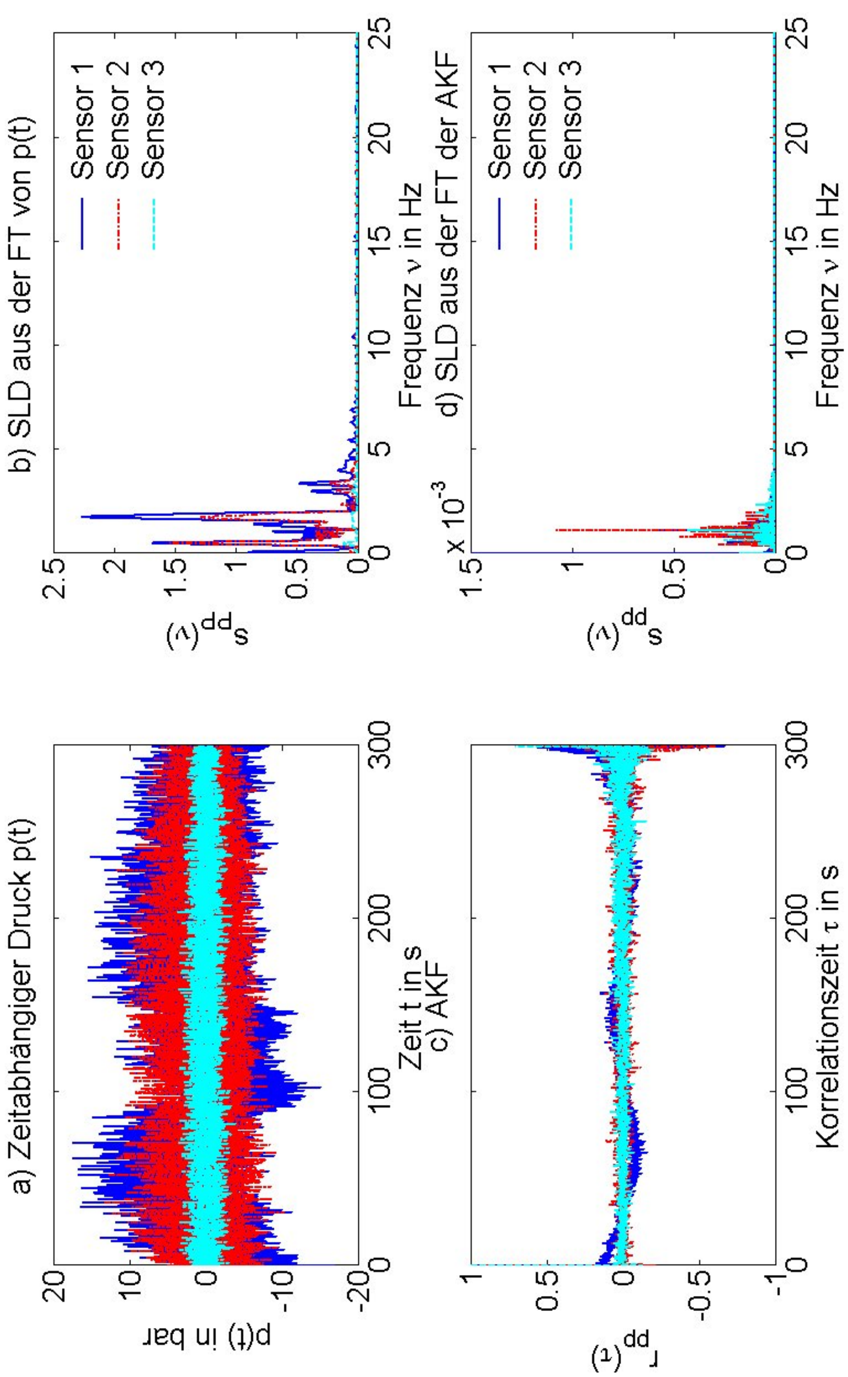


Abbildung 4.17: *PE-4 bei* $\vartheta = 180$ °C *und* $\dot{\gamma}_{ap} = 1963\ s^{-1}$*: a) Druckkurven, b) SLD direkt aus dem Druck, c) AKF und d) SLD aus der AKF.*

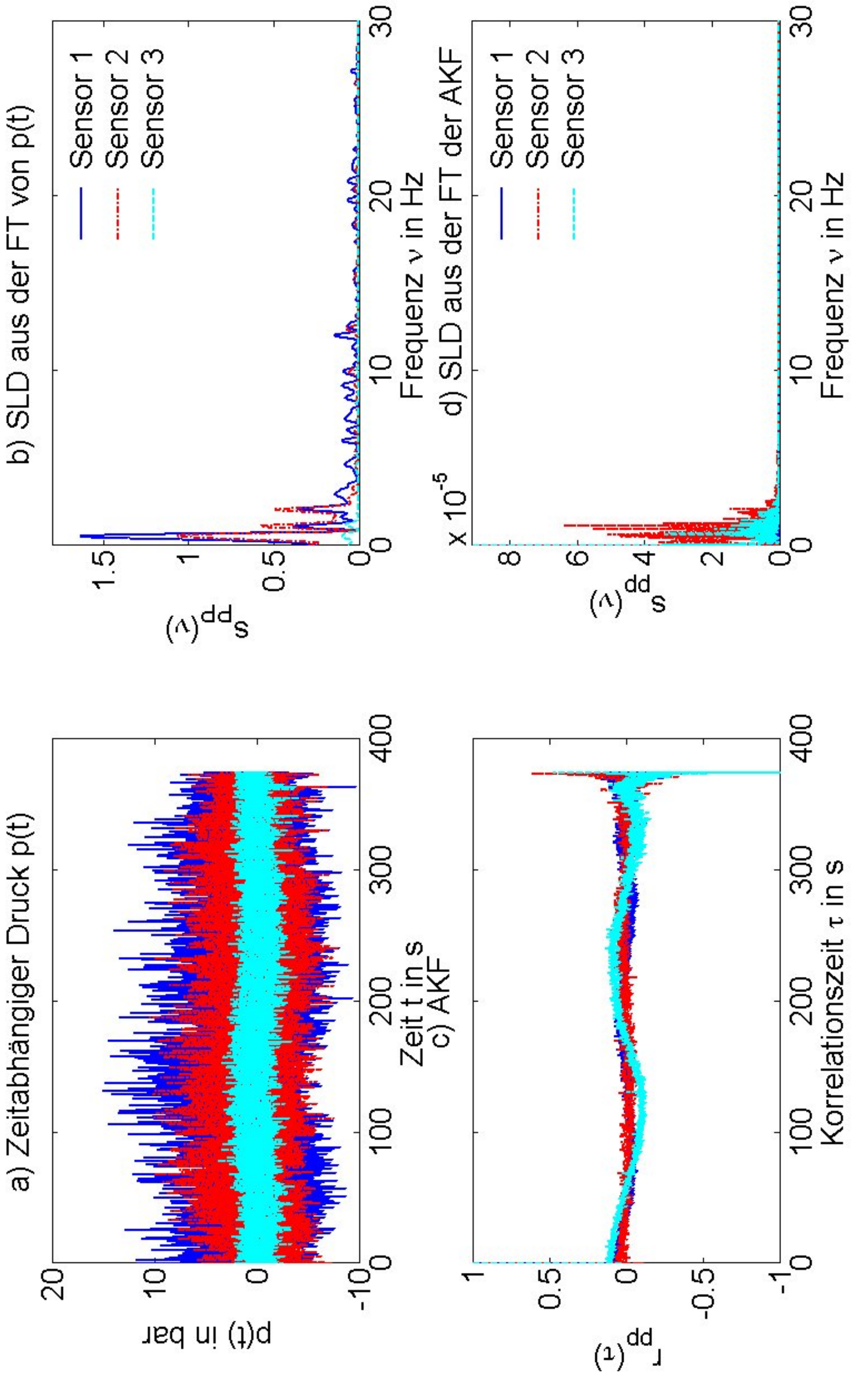


Abbildung 4.18: *PE-2 bei $\vartheta = 180$ °C und $\dot{\gamma}_{ap} = 1963\ s^{-1}$: a) Druckkurven, b) SLD direkt aus dem Druck, c) AKF und d) SLD aus der AKF.*

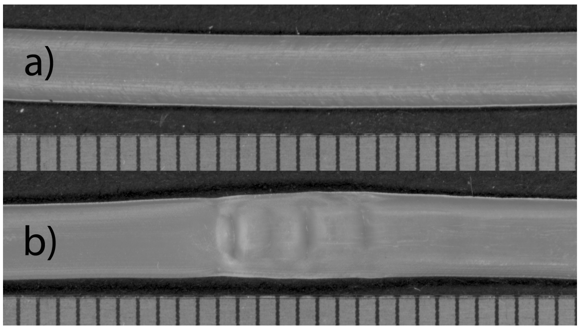


Abbildung 4.19: *Bilder der Extrudatprobe von PE-1 bei* $\vartheta = 180$ °C *und* $\dot{\gamma}_{ap} = 314\ s^{-1}$ *mit a) Sharkskin und b) Sick-Slip. Oben im Bild ist jeweils das Extrudat zu sehen darunter jeweils ein Lineal mit 1 mm Strichabstand.*

4.4 Stick-Slip Anomalie

Für die Messungen des Polymers PE-1 bei $\vartheta = 180$°C und $\dot{\gamma}_{app} = 314$ s^{-1} lässt sich beobachten, dass die Stick-Slip Fließanomalie sich nicht bei allen Versuchen zeigt. Da zwischen den einzelnen Messungen mit anderen Polymeren stattfanden, stellt sich die Frage, ob Verunreinigungen durch andere Polyemere Einfluss auf das Fließverhalten haben. Es wurden zwei Arten von Versuchen gefahren:

1. Messungen bei denen zuvor mindestens 3 Messungen mit gleicher Sorte von Polymer stattfanden und

2. Messungen bei denen direkt vor der Messung mit PE-2 gemessen wurde.

Repräsentativ für die Ergebnisse der Messart 1 sind die Abb. 4.19.a und die Resultate aus Kap. 4.1. Das Extrudat zeigt klares Stick-Slip Verhalten, wie in Abb. 4.19.a zu sehen ist und es Kap. 4.1 beschreibt.

Die Resultate der Messart 2 sind in den Abb. 4.19.a und 4.20 zu sehen. An der Extrudatprobe in Abb. 4.19.a zeigt sich, dass der Typus der Fließanomalie dem Sharkskin zugeordnet ist. In der SLD (Abb. 4.20.b und 4.20.d) treten Spitzen bei 0,1 Hz auf und sind am stärksten an der Sensorposition 3. Die AKF in Abb. 4.20.c verliert langsamer die Korrelation, als die für Stick-Slip. Außerdem sind keine periodischen Oszillationen sichtbar.

Daraus folgt, dass eine kleine Verunreinigung durch fremde Polymere die Erscheinungsform des Extrudats und des Polymers beeinflusst.

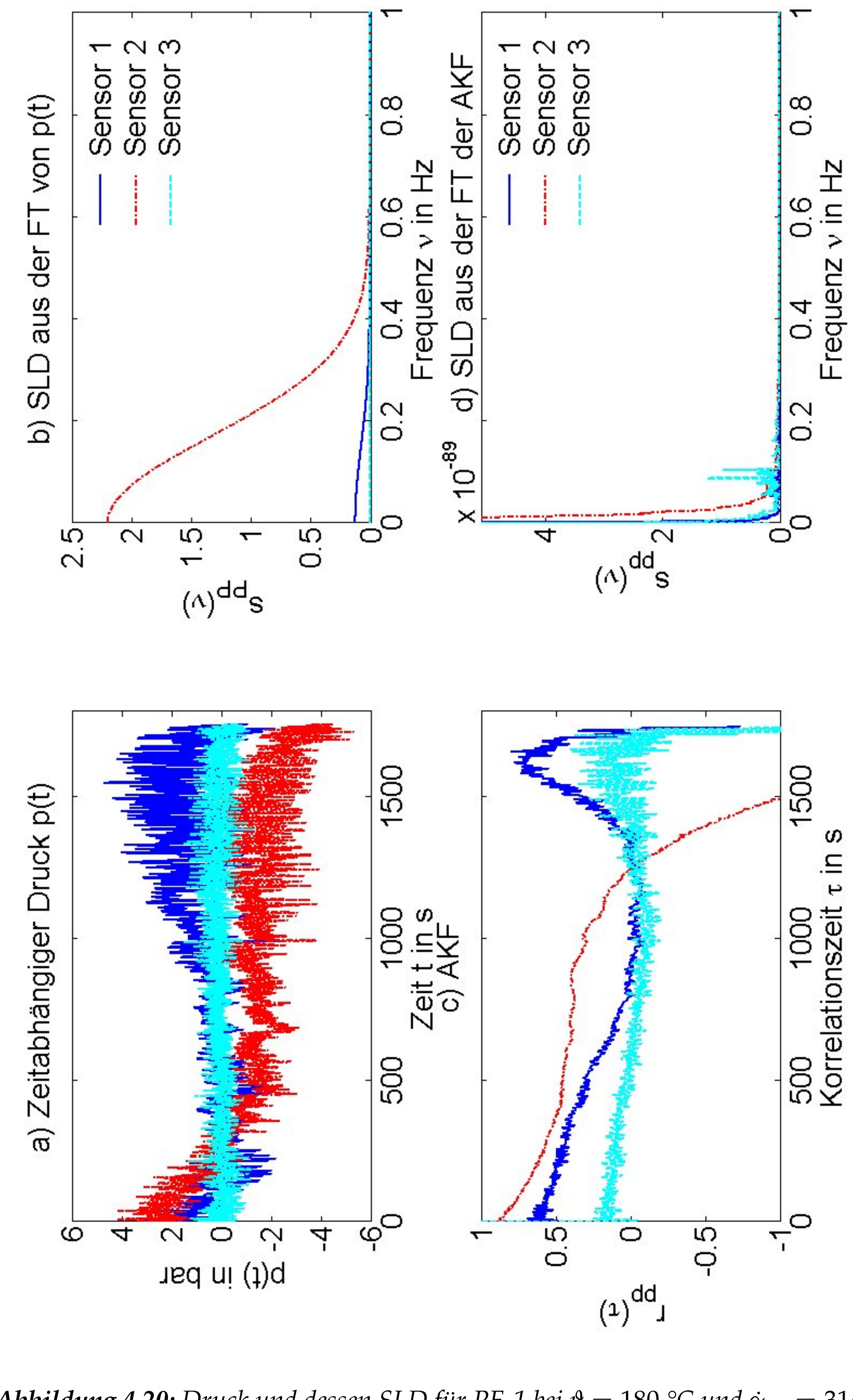


Abbildung 4.20: *Druck und dessen SLD für PE-1 bei* $\vartheta = 180$ °C *und* $\dot{\gamma}_{ap} = 314$ s^{-1}.

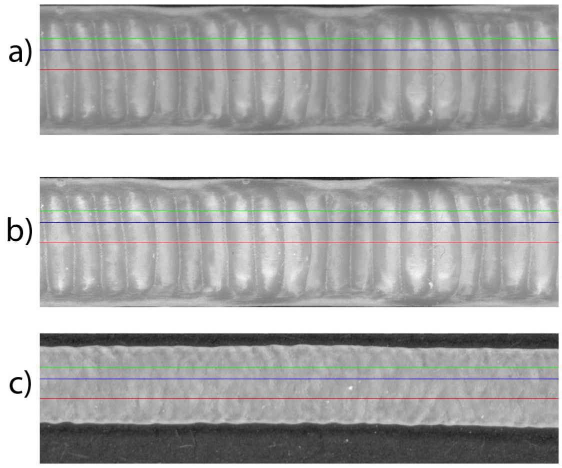


Abbildung 4.21: *Bild der Ausschnitte der Extrudate von a) PE-1 bei* $\vartheta = 155$ °C *und* $\dot{\gamma}_{ap} = 393\ s^{-1}$ *ungefiltertes Bild, b) PE-1 bei* $\vartheta = 155$ °C *und* $\dot{\gamma}_{ap} = 393\ s^{-1}$ *gefiltertes Bild und c) PE-4 bei* $\vartheta = 180$ °C *und* $\dot{\gamma}_{ap} = 1963\ s^{-1}$ *gefiltert. Die farbigen Linien ensprechen den Testzeilen, deren FT in Abb. 4.23 dargestellt ist.*

4.5 Optische Analyse der Instabilitäten

Resultierend aus der Auflösung der Bilder ergab die Bildanalyse nur nennenswerte Resultate für Proben mit deutlich sichtbaren Defekten an der Oberfläche. Dabei waren Sharkskin und Melt-Fracture die einzigen, die der Analyse zur Verfügung standen, da bei dem gefundenen Stick-Slip die Bereiche der Anomalien ca. 10 cm auseinander liegen. Dies ist für die verwendete Methode zur Bilderstellung zu groß.

Für Sharkskin für PE-1 bei $\vartheta = 155$ °C und $\dot{\gamma}_{app} = 393\ s^{-1}$ ergab sich eine FFT wie in Abb. 4.22 zu sehen. Darin sind deutlich Spitzen über die gesamte Breite des Extrudats bei 50 und 100 Pixel zu sehen. Das ursprüngliche Bild des Extrudats ist in Abb. 4.21.a oben zu sehen und das gefilterte in Abb. 4.21.b unten. Die FFT Spektren und die Intesitätskurven für drei Testzeilen ist in Abb. 4.23 zu sehen.

Im Vergleich dazu ist eine Probe mit Melt-Fracture zu sehen. Die Abb. 4.21.c zeigt das gefilterte Bild von PE-4 bei $\vartheta = 180$°C und $\dot{\gamma}_{app} = 1610\ s^{-1}$. In Abb. 4.24 ist die FFT aller Zeilen zu sehen. Hier sind keine charakteristischen Spitzen zu sehen.

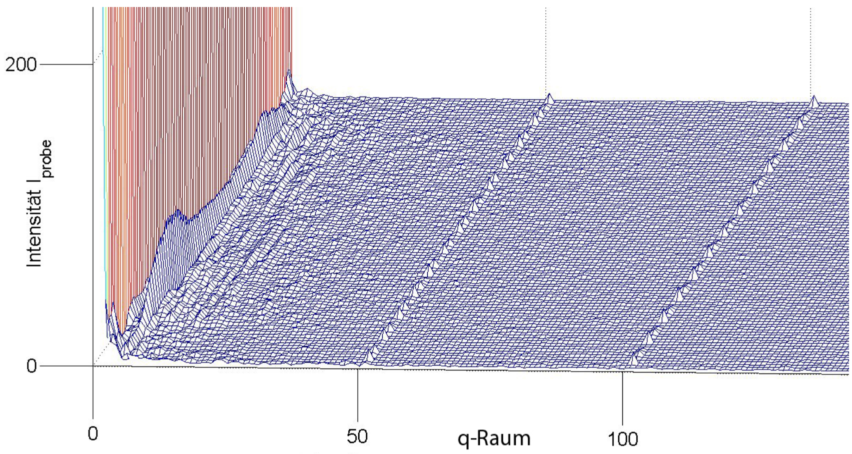


Abbildung 4.22: *3D-Plot der FFT für PE-1 bei* $\vartheta = 155°C$ *und* $\dot{\gamma}_{app} = 393\ s^{-1}$.

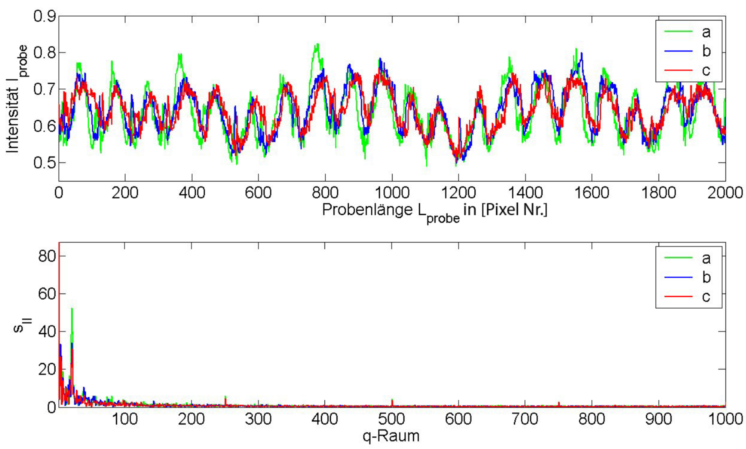


Abbildung 4.23: *Plot der Intensität und der SLD für die drei Testzeilen bei PE-1 bei* $\vartheta = 155°C$ *und* $\dot{\gamma}_{app} = 393\ s^{-1}$. *Die Farben sind analog zu denen aus Abb. 4.21*

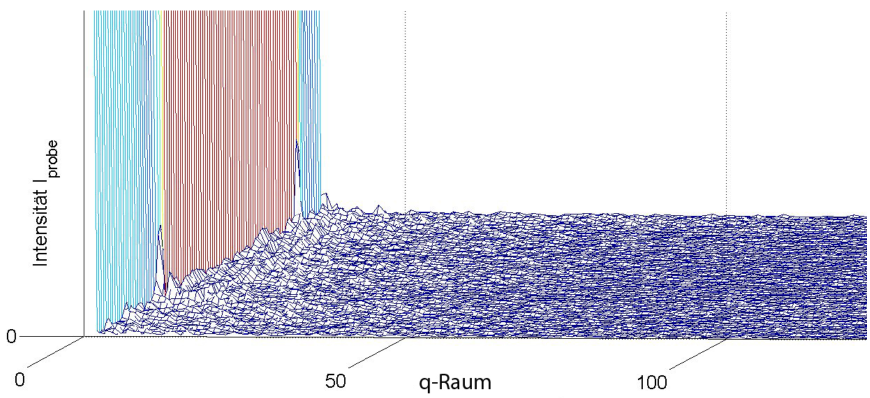


Abbildung 4.24: *3D-Plot der FFT für PE-4 bei* $\vartheta = 180°C$ *und* $\dot{\gamma}_{app} = 1963\ s^{-1}$.

4.6 Zusammenfassung von Kapitel 4

Aus den Resultate der in diesem Kapitel besprochenen Messungen hat sich ein Gesamtbild ergeben, dass sich wie folgt darstellt. Die Druckkurve des gemessenen Drucks liefert mit der Standardabweichung des Drucks eine einfache Größe zur Klassifizierung der Fließanomalie. Dabei sprechen hohe Standardabweichungen für Stick-Slip und sehr kleine Standardabweichungen für glattes Extrudat. Dazwischen liegen Sharkskin und Melt-Fracture. Diese beiden liegen eng bei einander. Die Höhe der Standardabweichung des Drucks wird von der Temperatur und dem verwendeten Polymer beeinflusst. Es lassen sich keine absoluten Grenzen geben.

Die AKF liefert eine klare Trennlinie zwischen glattem Extrudat und allen anderen Fließanomalien. Dafür ist die Anfangssteigung der AKF die entscheidenden Größe. Glattes Extrudat verliert wesentlich langsamer an Korrelation, als die Fließanomalien. Im Weiteren, kann an der AKF Kurve Stick-Slip durch seine deutlichen Oszillationen erkannt werden. Die Sharkskin und Melt-Fracture Fließanomalie zeichnen sich durch ähnliches Verhalten aus. Schneller Verlust der Korrelation und verbleib der Kurve in einem engen Band um die τ-Achse. Die SLD gibt eine Möglichkeit Sharkskin von Melt-Fracture zu unterscheiden. Bei Sharkskin stellt sich in der SLD durch enge Spitzen dar. Während für Melt-Fracture breitere Spitzen typisch sind. Dabei spielt die Art und Wiese, wie die SLD berechnet wurde keine Rolle. Sowohl die direkt aus dem Druck bestimmte, als auch die aus der AKF bestimmte, SLD weisen das gleiche qualitative Verhalten auf. Der einzige Unterschied liegt in dem verrauschteren Spektrum und den aufgeweiteten Spitzen in der SLD aus der AKF.

Diese Charakteristiken ergaben sich, soweit vorhanden, für alle behandelten Polymere und auch bei verschiedenen Temperaturen.

Außerdem wurde noch beobachtet, dass Verunreinigungen durch andere Polymere, die Messung beeinflussen können. Es können somit Fließanomalien erzeugt werden, die nicht charakteristisch für das eigentliche Polymer sind.

Kapitel 5

Diskussion

In diesem Kapitel werden die Resultate aus dem Kap. 4 zusammengetragen und es wird versucht aus ihnen ein Schema zur Charakterisierung von Fließanomalien zu erstellen. Außerdem wird diskutiert, in wie weit die Beobachtungen der mathematischen Theorie entsprechen. Am Ende wird noch ein Ausblick auf potentielle zukünftige Arbeiten gegeben.

5.1 Charakteristika der Fließanomalien

In dem beobachteten Spektrum an Versuchen ergab sich nur für PE-1 bei 180 °C bis zu $\dot{\gamma}_{ap} = 196\ s^{-1}$ ein deutlich sichtbar glattes Extrudat. Da sich keine Veränderungen an der Oberfläche des Extrudats zeigen, ist eine geringe Standardabweichung des Drucks vom mittleren Druck die logische Konsequenz. Dies zeigten die Versuche der Messreihe I. Für den stationären Betrieb, bzw. stationäre Extrusion, sollte ein konstanter Druck herrschen. Dies konnte bis auf die Sensor spezifische Drift auch festgestellt werden. Damit ergibt sich für die stationäre Druckfunktion eine Konstante, wie dies in Abb. 5.1.a beispielhaft dargestellt ist. Die Autokorrelationsfunktion einer solchen Funktion müsste idealer Weise eins sein. In der realen Messung tritt einen Änderung des Drucks über lange Zeit auf, damit ist die Autokorrelationsfunktion nicht permanent eins. Es zeigt sich, dass sie langsam abfällt. Da nun die Theorie sowohl die Autokorrelationsfunktion als auch die Druckfunktion als konstant beschreibt, ergibt sich eine spektrale Leistungsdichte von ihnen, die außer dem 0 Hz Maximum keine andere Anteile aufweist. Dies zeigt sich sehr gut in den bestimmten spektrale Leistungsdichte. Damit lässt sich die Charakteristik von glattem Extrudat darstellen wie in Abb. 5.1.a.

Auch bei der Beobachtung der Stick-Slip Fließanomalie ergaben sich keine ausreichenden Vergleichsmöglichkeiten. Stick-Slip konnte nur für PE-1 gefunden werden. Stick-Slip ergab sich unter den Bedingungen 180 °C und $\dot{\gamma}_{ap} = 314\ s^{-1}$. Erschwerend kommt hinzu, dass es zudem nur unter

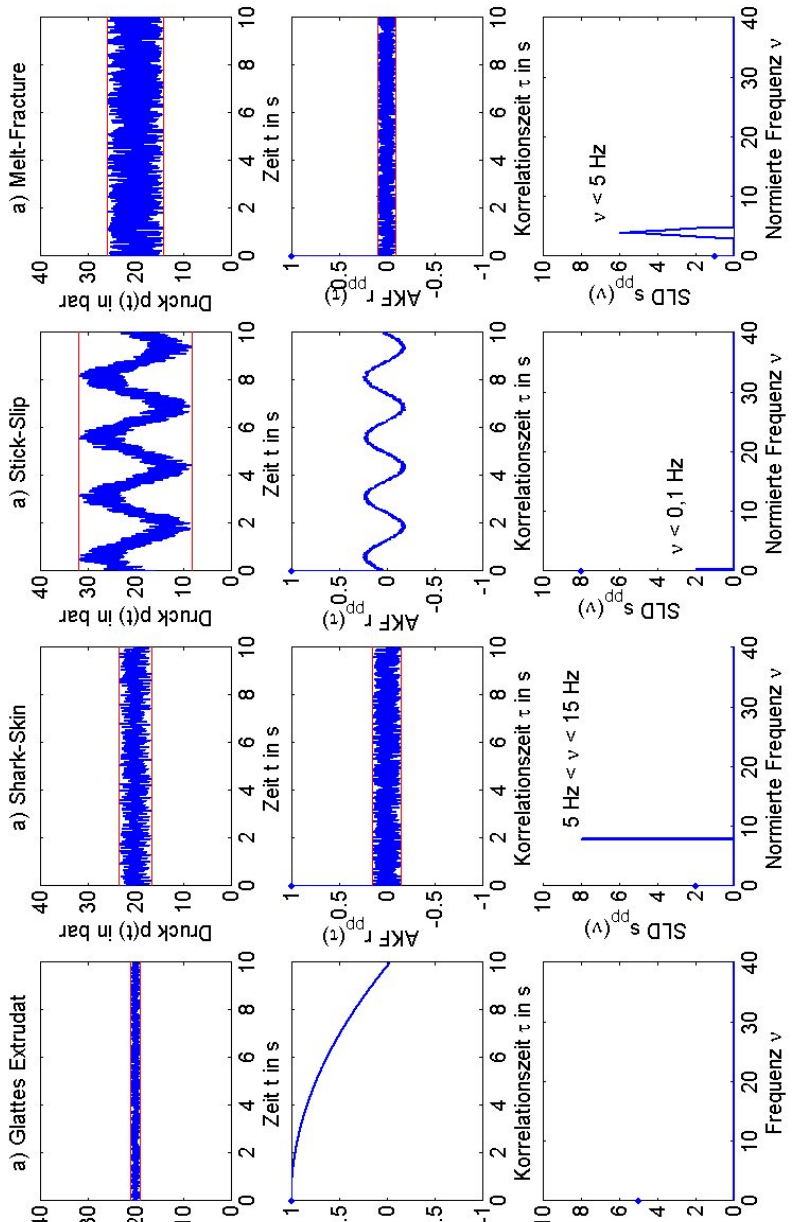


Abbildung 5.1: *Schema für den Druck, Autokorrelationsfunktion und spektrale Leistungsdichte bei den verschiedenen Fließphänomenen. a) glattes Extrudat, b) Sharkskin, c) Stick-Slip und d) Melt-Fracture. (Blau) Funktionskurve, (rot) einhüllende Hilfskurve. In der Autokorrelationsfunktion und der spektralen Leistungsdichte sind die Anfangspunkte mit 'o' markiert.*

der Voraussetzung von Verunreinigungen entsteht. Dies ist auch die Erklärung für das vorzeitige Auftreten dieser Anomalie. Die Literatur [38], [35] und [11] beschreibt die Abfolge der Fließanomalien, bei Erhöhung der Scherrate:

1. Glatt,
2. Sharkskin,
3. Stick-Slip,
4. Schmelzbruch (Melt-Fracture).

Aus den beschriebenen Messungen ergab sich eine Vertauschung zwischen 2 und 3. Diese Vertauschung hat wahrscheinlich ihren Ursprung in der Beeinflussung des Versuchablaufs durch die Verunreinigungen. Die Charakteristik der Stick-Slip Fließanomalie zeigte sich wie folgt. Ihr Druckverlauf weißt große als periodisch zu erkennenden Oszillationen auf. Dies ist bedingt durch das sich abwechselnde Haften und Gleiten an der Wand (siehe [39]). In der Autokorrelation sollte sich diese auch zeigen. Ein einfaches Beispiel dafür ist die Sinusschwingung, welche sich als Kosinusfunktion in der Autokorrelationsfunktion darstellt. In der Autokorrelationsfunktion zeigte sich wieder eine Schwingung. Damit sollte die spektrale Leistungsdichte Frequenzanteile in Form von Spitzen aufweisen. Diese sollten bei niedrigen Frequenzen liegen. Dies zeigte sich bei Frequenzen kleiner als 0,1 Hz. Aber nur in der spektrale Leistungsdichte, die über die Autokorrelationsfunktion bestimmt wurde. Eine Serie an Musterkurven ist in Abb. 5.1.b gegeben.

Die Sharkskin Fließanomalie konnte für mehrere Polymere und Temperaturen gefunden werden. Ihr Extrudat zeigt schnelle periodische Defekte. Damit lässt sich folgern, dass der Druckverlauf Oszillationen unterliegt und damit in der spektralen Leistungsdichte diese sich in Form von Spitzen bei hohen Frequenzen zeigen.Von der Autokorrelationsfunktion lässt sich fordern, dass sie nur über kurze Zeiträume Korrelation aufweist. Die gemessenen Druckverläufe zeigen eine höhere Standardabweichung, als die von glattem Extrudat und eine deutlich niedrigere als die von Stick-Slip. Die spektrale Leistungsdichte kennzeichnet sich durch enge Spitzen, die sich mit höher werdender Extrusionsgeschwindigkeit nach links verschiebt. Die Autokorrelationsfunktion ist der von Schmelzbruch (Melt-Fracture) ähnlich. Sie weißt mit $\tau_{0,8} << 1$ s eine kurze Korrelationszeit auf. Nach einem schnellen Abfall der Korrelation fluktuiert die Autokorrelationsfunktion in einen Band von $\pm 0,15$ um die τ-Achse. Beispielhaft ist der zu erwartende Verlauf für die Sharkskin Fließanomalie in Abb. 5.1.c dargestellt.

Schmelzbruch (Melt-Fracture) lies sich in verschiedenem Maß für verschiedene Polymere und Temperaturen finden. Die in der Literatur beschriebene Obeflächen-Charakteristik ist geprägt von starken Unregelmäßigkeiten. Das setzt voraus, das sich starke Druckfluktuationen zeigen, deren Korrelation gering ist. Die Druckfunktion erfüllt diese Erwartung auch. Die Standardabweichung ist die zweit höchste, nach Stick-Slip. Sie liegt aber noch nahe bei der von Sharkskin. Die Autokorrelationsfunktion zeigt einen schnellen Abfall der Korrelation, dem eine Fluktuation um die τ-Achse folgt. Darin sind sich Sharkskin und Schmelzbruch (Melt-Fracture) gleich. In der spektrale Leistungsdichte treten breite Spitzen auf. Die Leistung ist nicht konzentriert auf eine bestimmte Frequenz, wie bei Sharkskin. Dies ist auch die größte Differenz zwischen den beiden. Eine Serie an Musterkurven ist in Abb. 5.1.d gegeben.

5.2 Ausblick

Die in Abb. 5.1 zusammen gestellten Resultate zeigen, dass eine Unterscheidung der Fließanomalien anhand der Kombination aller drei Funktionen möglich ist. Im nächsten Schritt stellt sich damit die Herausforderung, zu überprüfen, ob eine technische Anwendung machbar ist. Ziel dabei ist es durch in die Düse eines Extruders integrierte Drucksensoren den Extrusionsprozess zu kontrollieren. Dabei ist nur eine Unterscheidung zwischen glattem Extrudat und verändertem Extrudat nötig. Dies macht den Entscheidungsprozess einfacher und die Umsetzung damit auch.

Ein weiteres Thema für folgende Arbeiten ist die genaue Erforschung der Beeinflussung durch Verunreinigungen. Fragestellung dabei könnten sein, welche Kombination an Material-Verunreinigung den Extrusionsprozess erleichtern können ohne die Eigenschaften des Endproduktes signifikant zu ändern. Dabei ist auch interessant, ob die Lokalisierung der Verunreinigungen eine Rolle spielt. Es wird zum Beispiel in [38] beschrieben, das nur der Endbereich der Düse einen Einfluss auf das Sharkskin Verhalten hat.

Während der Messreihe I lies sich bei der Sharkskin Fließanomalie neben der gesuchten Oberflächen Schuppung noch eine weitere Anomalie finden. Diese äußerte sich in einem schienenartigen Aussehen. Ursprung und Charakteristiken dieser Fließanomalien könnten ebenso Thema einer weiteren Arbeit sein.

Als letztes ist noch zu sagen, dass das Spektrum der in dieser Arbeit behandelten Polymere doch beschränkt war. Weitere Versuche mit anderen Polymeren könnten die Charakterisierung der Fließanomalien ergänzen. Ebenso wäre eine Erweiterung auf andere Testkanal- und Düsengeometrien wünschenswert, um diese mit dem hier gefundenen zu vergleichen und deren Einfluss zu bestimmen. Dabei kann eine bessere Bilderstellung der Extrudatproben die Klassifizierung der Fließanomalie helfen.

Anhang A

Entwickelte Messausrüstung

In diesem Abschnitt finden sich die wichtigsten technischen Daten der verwendeten Ausrüstung.

A.1 DMS Drucksensor

Der im Kapillarrheometer eingebaute Dynesco DMS-Ducksensor ist in Abb. A.1 zu sehen und seine technische Daten sind in der Tab. A.1 aufgelistet.

DMS-Drucksensor Typ:		HDA 2174-18-2000-15-1
Artikelnr.:		8.81.168
max. Arbeitstemp.:	[^{o}C]	400
Messbereich:	[bar]	$0 - 2000$
Kalibrationsdruck:	[bar]	1595

Tabelle A.1: *Technische Daten des DMS-Drucksensors.*

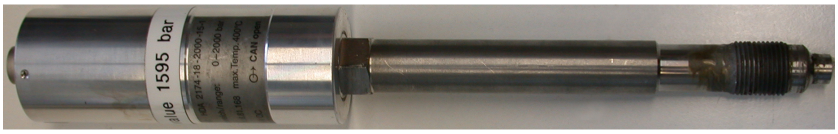


Abbildung A.1: *DMS-Drucksensor.*

A.2 Piezo Drucksensor

Kistler Drucksensor 6182AE (Abb.A.2) (mit O-Ring Dichtung)
Technische Daten sind in Tab. A.2 aufgelistet.

Messprinzip:		Piezoelektrisch
Druckbereich:	[bar]	0...2000
Sensitivität:	[pC/bar]	≈ -2.5
Resonanz Frequenz	[kHz]	> 80
max. Temperatur:	[^oC]	< 450

Tabelle A.2: *Technische Daten des Kistler Druckaufnehmers 6182AE.*

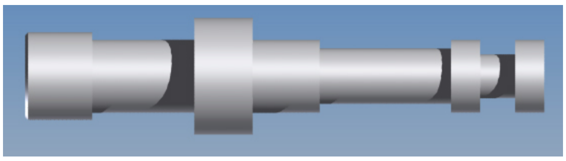


Abbildung A.2: *Kistler Drucksensor 6182AE.*

Anhang B

Datenanalyse Software

B.1 Matlab Programme zur Datenanalyse

Alle in dieser Arbeit präsentierten Messkurven und Funktionsgraphen sind mit den folgenden beiden Programmen erzeugt worden. Für eine Einführung in MATLAB ist [15] hilfreich. Informationen zur Bildverarbeitung finden sich in [14].

B.1.1 Bildanalyse mit Matlab

```
%Bildanalyse%
clear all
Datei='D:\transfer060505\Skripte\Diplomarbeit\Diplomarbeit 2006
\Bilder\Test\3020D180C090mms.jpg';
Bild=imread(Datei);
N=0;
%Bild öffnen und anzeigen %
figure(1);
imshow(Bild);
S=size(Bild)
h=S(1,1);
b=S(1,2);
y=1:1:b;
x=1:1:2000;

%Filterfunktion%
w1=fspecial('laplacian', 0);
%w2=[1 1 1; 1 -10 1; 1 1 1 ];
f2=im2double(Bild);
g1=imfilter(f2,w1,'replicate');
g=f2-g1;

%FFT für jede Bildzeile%
for k=1:1:h
    F=fft(g(k,:));
    S=sqrt((real(F)).^2+(imag(F)).^2);
    P(k,:)=S;
end
hold off
```

```
%2D-FFT und deren 3D-Plot%
figure(5);
G=fft2(Bild);
GX=fftshift(G);
SLD=sqrt((real(GX)).^2+(imag(GX)).^2);
mesh(SLD(1:5:500, 1:5:2000));

%Zeilen FFT als 3D Plot%
figure(2);
mesh(P(1:5:500, 1:5:2000));
xlabel('Probenlänge L_{probe}');
ylabel('Probenbreite B_{probe}');
zlabel('Intensität I_{probe}');
hold off

%Anzeige des gefilterten Bildes%
figure(3)
imshow(g,[]);
hold on
plot(x,250,'r');
hold on
plot(x,175,'b');
hold on
plot(x,130,'g');
hold off

%Einfügen der Testzeilen in das Ursprungsbild%
figure(1);
hold on;
plot(x,250,'r');
hold on
plot(x,175,'b');
hold on
plot(x,130,'g');
hold off

%FFT der Test Zeilen und 2-D Plot%
figure(4)
MF=fft(g(130,:));
 MS=sqrt((real(MF)).^2+(imag(MF)).^2);
 subplot(2,1,1)
 plot(x,g(130,:),'g');
 hold on
 subplot(2,1,2)
 plot(x,MS,'g');
 xlim([1 1000]);
hold on
MF=fft(g(175,:));
 MS=sqrt((real(MF)).^2+(imag(MF)).^2);
 subplot(2,1,1)
 plot(x,g(175,:),'b');
 hold on
 subplot(2,1,2)
 plot(x,MS,'b');
 xlim([1 1000]);
 hold on
 MF=fft(g(250,:));
 MS=sqrt((real(MF)).^2+(imag(MF)).^2);
 subplot(2,1,1)
 plot(x,g(250,:),'r');
 xlabel('Probenlänge L_{probe}');
 ylabel('Intensität I_{probe}');
```

```
  legend('a','b','c');
 hold on
 subplot(2,1,2)
 plot(x,MS,'r');
 xlim([1 1000]);
hold on
xlabel('Frequenz \nu_{probe}');
ylabel('s_{II}(\nu_{probe})');
 legend('a','b','c');
 hold off
```

Listing B.1: *MATLAB Bildanalyse Programm*

B.1.2 SLD mit Matlab

```
%Funktionsgraph des Druck und FFT der gemessenen Druckdaten%
clear all

%Darstellung und Analyse aller Dateien, die in
%DruckdateienListeMainTopic.txt aufgelistet sind%
fop=fopen('DruckdateienListeMainTopic.txt','r');

%NumberofFiles gibt an wieviele Dateien analysiert werden sollen.%
NumberofFiles=19;

%Starten der Analyse für jede Datei.%
for AnalysisSet=1:1:NumberofFiles
figure(AnalysisSet);
    DateiName=fgets(fop);
    Dsize=size(DateiName);
    DateiName(1,Dsize(1,2)-1)=0;

%Darstellung des Druckverlaufs aller drei Sensoren%
%Sensor 1%
name1=sprintf('%sP1.txt',DateiName);
fid=fopen(name1);
X1=fscanf(fid,'%g',[1 inf]);
NumberX=size(X1);
NX=NumberX(1,2);
t=0:1:NX-1;
HighestPressure=max(X1);
LowestPressure=min(X1);
fclose(fid);

%Bestimmung der Standardabweichung für 1%
pm1=1/NX*sum(X1(1,:))
VX1=(X1-pm1).^2;
sigma_f1=sqrt(1/(NX-1)*sum(VX1(1,:)));

%Sensor 2%
name1=sprintf('%sP2.txt',DateiName);
fid=fopen(name1);
X2=fscanf(fid,'%g',[1 inf]);
fclose(fid);

%Bestimmung der Standardabweichung für 2%
pm2=1/NX*sum(X2(1,:))
VX1=(X2-pm2).^2;
sigma_f2=sqrt(1/(NX-1)*sum(VX1(1,:)))

%Sensor 3%
name1=sprintf('%sP3.txt',DateiName);
```

```
fid=fopen(name1);
X3=fscanf(fid,'%g',[1 inf]);
fclose(fid);

%Bestimmung der Standardabweichung für 3%
pm3=1/NX*sum(X3(1,:))
VX1=(X3-pm3).^2;
sigma_f3=sqrt(1/(NX-1)*sum(VX1(1,:)));


%Darstellung in Graph 1%
subplot(4,1,1);
plot(t,X1,'-b','LineWidth',1.25);
hold on;
plot(t,X2,'-.r','LineWidth',1.25);
hold on;
plot(t,X3,'--c','LineWidth',1.25);

%Eigenschaften des Graphes%
xlabel('Messpunkte');
ylabel('Druck p(t) in bar');
TitelName=sprintf('Druckfunktion von %s \n',name1 );
title([TitelName,'Standartabweichung: s_{f_1} = ',num2str(sigma_f1),'
, s_{f_2} = ',num2str(sigma_f2),', s_{f_3} = ',num2str(sigma_f3)]);
legend('Sensor 1','Sensor 2','Sensor 3');legend('boxoff');
hold off;

%Rechteck Fenster-----------------------------------------%
%Fenster=1
%Gauss-Fenster--------------------------------------------%
sigma=100;
Fenster=1./sqrt(2.*pi)./sigma.*exp(-(t-NX/2).^2./sigma.^2/2);
%Hamming-Fenster------------------------------------------%
%Podest=0.1;
%Fenster=Podest+(1-Podest).*(cos(pi.*(t-NX)/NX)).^2;
%Blackmann Fenster----------------------------------------%
%Fenster=0.42-0.5*cos(2*pi*(t-NX/2)/max(t))+0.08*cos(4*pi/max(t)*(t-NX/2));

%FFT-----------------------------------------------------%
%Berechnung der SLD mit Fensterfunktion für alle drei Sensoren/Dateien%
%FFT für Sensor 1%
Z=fft(Fenster.*X1);
Z1=sqrt(real(Z).^2+imag(Z).^2);
Z1=Z1.^2;

%FFT für Sensor 2
Z=fft(Fenster.*X2);
Z2=sqrt(real(Z).^2+imag(Z).^2);
Z2=Z2.^2;

%FFT für Sensor 3
Z=fft(Fenster.*X3);
Z3=sqrt(real(Z).^2+imag(Z).^2);
Z3=Z3.^2;

%Darstellung alle drei SLD in Graph 2
subplot(4,1,2);
plot(2*t/(NX+1),Z1,'-b','LineWidth',1.25);
hold on
plot(2*t/(NX+1),Z2,'-.r','LineWidth',1.25);
hold on
plot(2*t/(NX+1),Z3,'--c','LineWidth',1.25)
```

```

%Eigenschafeten von Graph 2%
xlim([0 1])
%ylim([0 HighestNumber])
xlabel('Normierte Frequenz \Omega');
ylabel('s_{PP}(\Omega)');
legend('Sensor 1','Sensor 2','Sensor 3');legend('boxoff');
hold off

%ACF-plots and SLD-calculation and plotting%
%ACF is determined via ACF.exe%
%The name of all data files is listed in DruckdateienListeMainTopic.txt%
%Start reading the files from DruckdateienListeMainTopic.txt%
%NumberofFiles needs the amount of files to analyse and plot%
name=sprintf('%s',DateiName);

%Reading the ACF data from file%
name1=sprintf('%sACFFOLD1.txt',name);
fid=fopen(name1,'r');
X1=fscanf(fid,'%g %g',[2 inf]);
fclose(fid);
name1=sprintf('%sACFFOLD2.txt',name);
fid=fopen(name1,'r');
X2=fscanf(fid,'%g %g',[2 inf]);
fclose(fid);

name1=sprintf('%sACFFOLD3.txt',name);
fid=fopen(name1,'r');
X3=fscanf(fid,'%g %g',[2 inf]);
fclose(fid);

%Plotting the ACF of all 3 Sensors/Files in graph 3.%
subplot(4,1,3)
plot(X1(1,:),X1(2,:),'-b','LineWidth',1.25)
hold on;
plot(X2(1,:),X2(2,:),'-.r','LineWidth',1.25)
hold on;
plot(X3(1,:),X3(2,:),'--c','LineWidth',1.25)

xlabel('Korrelationszeit \tau in s')
ylabel('r_{pp}(\tau)')
ylim([-1 1]);
legend('Sensor 1','Sensor 2','Sensor 3');legend('boxoff')
hold off;

%SLD scaling.%
MaxX=max(X1(1,:));
X4=2*X1(1,:)/MaxX;
t=max(X4);
NX=max(X1(1,:));

%Gauss-Window---------------------------------------------%
sigma=60;
Fenster=1./sqrt(2.*pi)./sigma.*exp(-1/2*(X4(1,:)-NX/2).^2./sigma.^2);

%Hamming-Window-------------------------------------------%
%Podest=0.15;%
%Fenster=Podest+(1-Podest).*(cos(pi.*(X4(1,:)-NX/2)/NX)).^2;

%Calculation of the SLD via FFT. With Window function.%
Z=fft(X1(2,:).*Fenster);
Z1=sqrt(real(Z).^2+imag(Z).^2);
```

```
Z1=Z1.^2;
HighestNumber=max(Z1)/3;
Z=fft(X2(2,:).*Fenster);
Z2=sqrt(real(Z).^2+imag(Z).^2);
Z2=Z2.^2;
Z=fft(X3(2,:).*Fenster);
Z3=sqrt(real(Z).^2+imag(Z).^2);
Z3=Z3.^2;

%Plotting the SLD of all three files/Sensors in graph 4%
subplot(4,1,4);
plot(X4,Z1,'-b','LineWidth',1.25);
hold on;
plot(X4,Z2,'-.r','LineWidth',1.25)
hold on;
plot(X4,Z3,'--c','LineWidth',1.25)
xlim([0 t/2]);
ylim([0 HighestNumber]);
xlabel('Normierte Frequenz \Omega');
ylabel('s_{pp}(\Omega)');
legend('Sensor 1','Sensor 2','Sensor 3');legend('boxoff');
hold off;

%Save all created figures for coming analysis.%
name1=sprintf('G%s',DateiName);
saveas(gcf,name1,'fig');
end

fclose(fop);

%Printing each figure with windows standard printer%
%for i=1:1:60
%figure(i);
 %   print -dwinc;
 %   close(i);
 %end
```

Listing B.2: *MATLAB SLD Programm*

B.2 Programmierets C-Programm zur Datenanalyse

Bei der Erstellung der graphischen Elemente in C ist [29] hilfreich. Für eine Einführung in den allgemeinen Sprachumfang von C bietet sich [13] an.

```
/*********************************************************
*  ACF für Windows
*  12-12-2006 I.F.C. Naue
*********************************************************/
#include <ACF.h>       /* eigenen Definitionen */
#include <windows.h>
#include <string.h>
#include <math.h>
#include <time.h>
#include <ctype.h>
#include <stdio.h>
#include <stdlib.h>
/*********************************************************
* Deklaration von Window- Functions
*********************************************************/
LRESULT CALLBACK W5_WinProc(HWND, UINT, WPARAM, LPARAM);
BOOL    CALLBACK W2_DiaProc(HWND, UINT, WPARAM, LPARAM);
```

```
BOOL     CALLBACK Beschneiden_DiaProc(HWND, UINT, WPARAM, LPARAM);
BOOL     CALLBACK TESTDATA_DiaProc(HWND, UINT, WPARAM, LPARAM);
BOOL     CALLBACK TESTNAME_DiaProc(HWND, UINT, WPARAM, LPARAM);
BOOL     CALLBACK TESTSPECIAL_DiaProc(HWND, UINT, WPARAM, LPARAM);
BOOL     CALLBACK QR_MATRIX_DiaProc(HWND, UINT, WPARAM, LPARAM);
/*******************************************************
*  externe Variable
*******************************************************/
HINSTANCE  hProgram;
HWND       hWnd;
HDC        hDC;

double            x,y;
double            ixMin, ixMax, iyMin, iyMid, iyMax;
int               iy;


int          ixClient= 1300, iyClient= 800;
char         druckfile[MAX_FILE]="sinus",druckfile1[MAX_FILE],headerstore[MAX_FILE];
int          NameLength;
FILE         *streamdruck, *streaminfo, *streamausgabed1, *streamausgabed2, *streamausgabed3;
FILE         *streamacf1, *streamacf2, *streamacf3;
FILE         *streamacov1, *streamacov2, *streamacov3;
FILE         *streamlds1, *streamlds2, *streamlds3;
FILE         *streamft1, *streamft2, *streamft3;
char         dausgabe1[MAX_FILE], dausgabe2[MAX_FILE], dausgabe3[MAX_FILE], infoausgabe[MAX_FILE];
char         ausgabe1acf[MAX_FILE],ausgabe2acf[MAX_FILE],ausgabe3acf[MAX_FILE];
char         ausgabe1acov[MAX_FILE],ausgabe2acov[MAX_FILE],ausgabe3acov[MAX_FILE];
char         ausgabe1lds[MAX_FILE],ausgabe2lds[MAX_FILE],ausgabe3lds[MAX_FILE];
char         ausgabe1ft[MAX_FILE],ausgabe2ft[MAX_FILE],ausgabe3ft[MAX_FILE];
char         ch1[MAX_EINGABE];
int          j,k,l,n=1,z,i;
int          ka, kappa;
int          number, difft, column;
unsigned long int tstart, tend;
double       scanrate, oversampling, nyquistfrequenz, t_max, delta_t;
double       summe, tau_null, mean;
double       x1[1500000];
double       actualfreq, delta_f, fixed, lds, argum;
double       highest, lowest, DeltaX, DeltaY;
char         text[MAX_EINGABE]="XXX", yAchse[MAX_EINGABE]="Druck [Pa]";
double       xBeschriftung[12];
char         Beschriftung[20],AxisTyp;
char         pAnfang[MAX_EINGABE], pEnde[MAX_EINGABE];
int          pAn, pEn;
int          FunctionTyp;
char         a_value[MAX_EINGABE], b_value[MAX_EINGABE],c_value[MAX_EINGABE];
char         FunctionName[MAX_EINGABE];
double       FirstPar, SecondPar, ThirdPar;
char         Sscanrate[MAX_EINGABE], Sscanfrequency[MAX_EINGABE], Snumber[MAX_EINGABE];
double       scanfrequency, TruePoints, TotalTime;
FILE         *streamtestdata;
char         TestData[MAX_FILE];
FILE         *streamlcf1,*streamlcf2,*streamlcf3;
char         dausgabelcf1[MAX_FILE],dausgabelcf2[MAX_FILE],dausgabelcf3[MAX_FILE];
double       meanX, meanY, ConstA, ConstB;
double       QRx1, QRx2, QRx3;
double       QR1[1500000][3];
double       U1[1500000], bQR[1500000];
int          signQR;
double       BetragQR, AbsValQR, betaQR, SkalarQR[5], NennerQR;
char         textQR[MAX_EINGABE],matrixname[MAX_EINGABE],vektorname[MAX_EINGABE];
```

```
char        ausgabe1acff[MAX_FILE],ausgabe2acff[MAX_FILE],ausgabe3acff[MAX_FILE];
FILE        *streamacff1, *streamacff2, *streamacff3;
char        InfoNamen[2*MAX_EINGABE][50], InfoWerte[2*MAX_EINGABE][50];


void MbAlert(char text1[MAX_EINGABE])
{
    MessageBox(NULL,"File  not open!",text1,MB_OK | MB_ICONHAND);
    }

void MbInfo(char text1[MAX_EINGABE], char text2[MAX_EINGABE])
{
    MessageBox(NULL,text1,text2, MB_OK | MB_ICONINFORMATION);
    }

/*********************************************
*Diese Funktion erzeugt alle Dateinamen, anhand
*des Namens der Ursprungsdatei
*********************************************/
void Eingabe(void)
{
        strcpy(druckfile1,druckfile);
        strcat(druckfile1,".txt");

        strcpy(dausgabe1,druckfile);
        strcat(dausgabe1,"P1.txt");

        strcpy(dausgabe2,druckfile);
        strcat(dausgabe2,"P2.txt");

        strcpy(dausgabe3,druckfile);
        strcat(dausgabe3,"P3.txt");

        strcpy(dausgabelcf1,druckfile);
        strcat(dausgabelcf1,"LCF1.txt");

        strcpy(dausgabelcf2,druckfile);
        strcat(dausgabelcf2,"LCF2.txt");

        strcpy(dausgabelcf3,druckfile);
        strcat(dausgabelcf3,"LCF3.txt");

        strcpy(infoausgabe,druckfile);
        strcat(infoausgabe,"INFO.txt");

        strcpy(ausgabe1acf,druckfile);
        strcat(ausgabe1acf,"ACF1.txt");

        strcpy(ausgabe2acf,druckfile);
        strcat(ausgabe2acf,"ACF2.txt");

        strcpy(ausgabe3acf,druckfile);
        strcat(ausgabe3acf,"ACF3.txt");

        strcpy(ausgabe1acff,druckfile);
        strcat(ausgabe1acff,"ACFFOLD1.txt");

        strcpy(ausgabe2acff,druckfile);
        strcat(ausgabe2acff,"ACFFOLD2.txt");

        strcpy(ausgabe3acff,druckfile);
        strcat(ausgabe3acff,"ACFFOLD3.txt");
```

```

        strcpy(ausgabe1acov,druckfile);
        strcat(ausgabe1acov,"ACOV1.txt");

        strcpy(ausgabe2acov,druckfile);
        strcat(ausgabe2acov,"ACOV2.txt");

        strcpy(ausgabe3acov,druckfile);
        strcat(ausgabe3acov,"ACOV3.txt");

        strcpy(ausgabe1lds,druckfile);
        strcat(ausgabe1lds,"1SPD.txt");

        strcpy(ausgabe2lds,druckfile);
        strcat(ausgabe2lds,"2SPD.txt");

        strcpy(ausgabe3lds,druckfile);
        strcat(ausgabe3lds,"3SPD.txt");

        strcpy(ausgabe1ft,druckfile);
        strcat(ausgabe1ft,"1FT.txt");

        strcpy(ausgabe2ft,druckfile);
        strcat(ausgabe2ft,"2FT.txt");

        strcpy(ausgabe3ft,druckfile);
        strcat(ausgabe3ft,"3FT.txt");
    }

/*************************************************
*Öffnet die Messdatei und überträgt die einzelnen
*Spalten in jeweils eine separate Datei. Die Header-
*informatinen werden in eine weiter separate Datei
*geschrieben.
*************************************************/
void Separate(FILE *streamein, FILE *streamaus1,
FILE *streamaus2,FILE *streamaus3)
{
        if((streamein=fopen(druckfile1,"r"))==NULL){
        MbAlert("Eingabe");
        exit(1);
        }
        if((streamaus1=fopen(dausgabe1,"w"))==NULL){
        MbAlert("Eingabe");
        exit(1);
        }
            if((streamaus2=fopen(dausgabe2,"w"))==NULL){
        MbAlert("Eingabe");
        exit(1);
        }
            if((streamaus3=fopen(dausgabe3,"w"))==NULL){
        MbAlert("Eingabe");
        exit(1);
        }

/*************************************************/
/*Reading the file Header (first row)*/
    fgets(ch1,MAX_EINGABE,streamein);

/*FirstColumn scan rate in [pts/s]*/

            for(j=0; ch1[j] !=' ' ; j++){
```

```
                  if(ch1[j]==','){
                      ch1[j]='.';
                      }
              headerstore[j]=ch1[j];
                  }
              scanrate=atof(headerstore);

              while(ch1[j]==' '){
                  j++;
                  }
/*Second column Abtastfrequenz */
              k=0;
              while(ch1[j]!=' '){

                          if(ch1[j]==','){
                              ch1[j]='.';
                              }
                      headerstore[k]=ch1[j];
                      k++;
                      j++;
                          }
              headerstore[k]='\0';
      oversampling=atof(headerstore);

/*Third column Nyquistfrequency*/
              while(ch1[j]==' '){
                              j++;
                              }
                  k=0;
                          while(ch1[j] !='\n'){

                              if(ch1[j]==','){
                                  ch1[j]='.';
                                  }

                          headerstore[k]=ch1[j];
                  k++;
                  j++;
                      }
                  headerstore[k]='\0';
                  nyquistfrequenz=atof(headerstore);


/*Reading the pressure values*/

      fgets(ch1,MAX_EINGABE,streamein);

      number=0;
      while(!feof(streamein)) {

              for(j=0; ch1[j] != ('    ' ); j++){

                  if(ch1[j]==','){
                      ch1[j]='.';
                      }

                  fprintf(streamaus1,"%c",ch1[j]);
                  }


              for(k=j+1;ch1[k]!=('     ' ) ; k++){
```

```

                if(ch1[k]==','){
                    ch1[k]='.';
                    }

                fprintf(streamaus2,"%c",ch1[k]);
                }

            for(l=k+1;ch1[l] !='\n';  l++){

                if(ch1[l]==','){
                    ch1[l]='.';
                    }

                fprintf(streamaus3,"%c",ch1[l] );
                }

        fprintf(streamaus1,"\n");
        fprintf(streamaus2,"\n");
        fprintf(streamaus3,"\n");
        fgets(ch1,MAX_EINGABE,streamein);
        number++;
        }

    number=number;
    fclose(streamein);
    fclose(streamaus1);
    fclose(streamaus2);
    fclose(streamaus3);
 }

/************************************************
*Öffnet die Messdatei, damit diese im Programm weiter
*bearbeitet werden kann, z.B. auf dem Bildschirm
* angezeigt. Außerdem werden die Header Informationen
*und die Zeilenzahl gelesen.
***********************************************/
void LookFile(FILE *streamein)
{
        if((streamein=fopen(druckfile1,"r"))==NULL){
        MbAlert("Eingabe");
        exit(1);
        }

/*Reading the file Header (first row)*/
    fgets(ch1,MAX_EINGABE,streamein);

/*FirstColumn scan rate in [pts/s]*/

            for(j=0; ch1[j] !=' ' ; j++){

                if(ch1[j]==','){
                    ch1[j]='.';
                    }
            headerstore[j]=ch1[j];
                }
            scanrate=atof(headerstore);

            while(ch1[j]==' '){
                j++;
                }
/*Second column Abtastfrequenz */
```

```
            k=0;
            while(ch1[j]!=' '){

                        if(ch1[j]==','){
                            ch1[j]='.';
                            }
                    headerstore[k]=ch1[j];
                    k++;
                    j++;
                        }
            headerstore[k]='\0';
    oversampling=atof(headerstore);

/*Third column Nyquistfrequency*/
            while(ch1[j]==' '){
                            j++;
                            }
                k=0;
                        while(ch1[j] !='\n'){

                            if(ch1[j]==','){
                                ch1[j]='.';
                                }

                        headerstore[k]=ch1[j];
                k++;
                j++;
                    }
                headerstore[k]='\0';
                nyquistfrequenz=atof(headerstore);

/*Reading the pressure values*/

    fgets(ch1,MAX_EINGABE,streamein);

    number=0;
    while(!feof(streamein))
        {

        fgets(ch1,MAX_EINGABE,streamein);
        number++;
        }

    number=number;
    fclose(streamein);
 }

/*****************************************************
*Funktion, die die Dateien für die ACF öffnet und die
*drei Druckdateien öffnet.
*****************************************************/
void OpenDruckACF(void)
{
    if((streamacf1=fopen(ausgabe1acf,"w"))==NULL){
        MbAlert("Eingabe");
        exit(1);
        }

    if((streamacf2=fopen(ausgabe2acf,"w"))==NULL){
        MbAlert("Eingabe");
        exit(1);
        }
```

```

    if((streamacf3=fopen(ausgabe3acf,"w"))==NULL){
        MbAlert("Eingabe");
        exit(1);
        }

    if((streamausgabed1=fopen(dausgabe1,"r"))==NULL){
        MbAlert("Eingabe");
        exit(1);
        }
    if((streamausgabed2=fopen(dausgabe2,"r"))==NULL){
        MbAlert("Eingabe");
        exit(1);
        }
    if((streamausgabed3=fopen(dausgabe3,"r"))==NULL){
        MbAlert("Eingabe");
        exit(1);
        }

}
void OpenDruckACOV(void)
{
    if((streamacov1=fopen(ausgabe1acov,"w"))==NULL){
        MbAlert("Eingabe");
        exit(1);
        }

    if((streamacov2=fopen(ausgabe2acov,"w"))==NULL){
        MbAlert("Eingabe");
        exit(1);
        }

    if((streamacov3=fopen(ausgabe3acov,"w"))==NULL){
        MbAlert("Eingabe");
        exit(1);
        }

    if((streamausgabed1=fopen(dausgabe1,"r"))==NULL){
        MbAlert("Eingabe");
        exit(1);
        }
    if((streamausgabed2=fopen(dausgabe2,"r"))==NULL){
        MbAlert("Eingabe");
        exit(1);
        }
    if((streamausgabed3=fopen(dausgabe3,"r"))==NULL){
        MbAlert("Eingabe");
        exit(1);
        }

}

/***************************************************
*Diese Funktion bestimmt die Gesamte Messdauer anhand
*der Anzahl der Datenpunkte und die Zeitdifferenz
*zwischen zwei Messpunkten.
***************************************************/
void Messzeit(void)
{
    delta_f=nyquistfrequenz/(number-1);
    fixed=2*PI/nyquistfrequenz/2;
    t_max=number/2/nyquistfrequenz;
```

```
    delta_t=1/(2*nyquistfrequenz);
}


/******************************************
*Diese Funktion schreibt die Werte eines vorher
*geöffneten Datei in die Zellen des Arrays
*x1[] und schließt die Datei danach wieder.
******************************************/
void Transducer(FILE *stream)
{
    i=0;
    fgets(ch1,MAX_EINGABE,stream);
    mean=0;
    while(!feof(stream))
        {
            x1[i]=atof(ch1);
            fgets(ch1,MAX_EINGABE,stream);
            i++;
        }
    fclose(stream);
}

/*********************************************
*Funktion die den Algorithmus enthält, um die
*Autokorrelation einer Datei zu berechnen.
*Das Arbeitsarray x1[] liefert dabei die Daten.
*Die Werte der ACF und die zugehörige Korrela-
*tionszeit werden in die übergebene Datei ge-
*schrieben
*********************************************/
void ACF(FILE *stream)
{
    time(&tstart);
    for(kappa=0; kappa < number-1 ; kappa++)
        {
            summe=0;

            for(ka=0 ; ka < (number-1-kappa) ; ka++ )
                {
                    summe = summe + x1[ka] * x1[ka +kappa];
                }
            summe=summe/(number-1-kappa);
            if(kappa==0)
                {
                    tau_null=summe;
                }
            summe=summe/(tau_null);
            fprintf(stream,"%.6f %.6f\n",kappa*delta_t,summe);
        }
    time(&tend);
    difft=(tend-tstart);
    fclose(stream);
     }


/*********************************************
*Funktion die den Algorithmus enthält, um die
*Autkovarianz einer Datei zu berechnen.
*Das Arbeitsarray x1[] liefert dabei die Daten.
*Die Werte der ACOV und die zugehörige Korrela-
*tionszeit werden in die übergebene Datei ge-
```

```
*schrieben
*******************************************/
void ACOV(FILE *stream)
{
    time(&tstart);
    for(kappa=0; kappa < number-1 ; kappa++)
        {
            summe=0;

            for(ka=0 ; ka < (number-1-kappa) ; ka++ )
                {
                    summe = summe + x1[ka] * x1[ka +kappa]-mean*mean;
                }
            summe=summe/(number-1-kappa);
            if(kappa==0)
                {
                    tau_null=summe;
                }
            summe=summe/(tau_null);
            fprintf(stream,"%.6f %.6f\n",kappa*delta_t,summe);
        }
    time(&tend);
    difft=(tend-tstart);
    fclose(stream);
     }




/*****************************************************
*Erzeugung der Datein mit den Daten für die FT Berechnung
*****************************************************/
void OpenLDS(void)
{
            if((streamlds1=fopen(ausgabe1lds,"w"))==NULL){
                MbAlert("Eingabe");
                exit(1);
                }

            if((streamlds2=fopen(ausgabe2lds,"w"))==NULL){
                MbAlert("Eingabe");
                exit(1);
                }

            if((streamlds3=fopen(ausgabe3lds,"w"))==NULL){
                MbAlert("Eingabe");
                exit(1);
                }

       }

/************************************************************
*This function opens the before calculated ACF Files takes
*its values and puts them into a new file.
*Then it takes the values into an array x1[] for further calculations.
************************************************************/
void Extract(FILE *stream1, FILE *stream2,
 char ausgabe1[MAX_FILE],char ausgabe2[MAX_FILE] )
{
        /*FT der ACF von Transducer 1*/

            if((stream1=fopen(ausgabe1,"r"))==NULL){
```

```
                MbAlert("Eingabe");
                exit(1);
                }

            fgets(ch1,MAX_EINGABE,stream1);

            while(!feof(stream1)) {
                    for(j=0; ch1[j] !=' ' ; j++)
                {
                    /*Die erste Spalte, mit der Zeilennr., ueberspringen.*/
                }

                    for(k=j+1;ch1[k]!='\n'; k++){

                        fprintf(stream2,"%c",ch1[k]);
                        }

                fprintf(stream2,"\n");
                fgets(ch1,MAX_EINGABE,stream1);
                }

                fclose(stream1);
                fclose(stream2);

        /*Schreiben der Werte der Sources Files in ein Array*/

            if((stream2=fopen(ausgabe2,"r"))==NULL){
                MbAlert("Eingabe");
                exit(1);
                }
            z=0;              /*Zeilenzaehler*/
            fgets(ch1,MAX_EINGABE,stream2);
            while(!feof(stream2))
                {
                    x1[z]=atof(ch1);
                    z++;
                    fgets(ch1,MAX_EINGABE,stream2);
                }
        fclose(stream2);
    }


/***********************************************************
*This function calculates the real part FT of an array x1[].
***********************************************************/
void RFT(FILE *stream, char ausgabe[MAX_FILE])
{
/*FT der AKOVF  der ersten Datenspalte in der Ursprungsdatei*/
        if((stream=fopen(ausgabe,"w"))==NULL){
                MbAlert("Eingabe");
                exit(1);
                }
        actualfreq=0;
        time(&tstart);

        /*Frequenzwechsel*/
        for(j=0;  j<number; j++)
            {
            lds=0;
            actualfreq=delta_f*j;
            argum=fixed*(actualfreq);

```

```
            /*SUMME*/
            for(k=1; k<number; k++)
            {
                lds=lds+x1[k]*cos(k*argum);
                }
            lds= x1[0]+2*lds;
            if(lds<0){
                lds=-lds;
                }
            fprintf(stream,"%.6lf %.6lf\n",actualfreq,lds);

            }

            time(&tend);
            difft=(tend-tstart);
            fclose(stream);
     }



/*******************************************************
*Öffenen der separierten Druckdaten und schreiben der Werte
*in das Array x1[]
*******************************************************/
void OpenP(FILE *streamp, char DruckName[MAX_FILE])
{
            if((streamp=fopen(DruckName,"r"))==NULL){
                MbAlert("Eingabe");
                exit(1);
                }

            z=0;
            fgets(ch1,MAX_EINGABE,streamp);
            while(!feof(streamp))
                {
                    x1[z]=atof(ch1);
                    z++;
                    fgets(ch1,MAX_EINGABE,streamp);
                }

    number=z;
    fclose(streamp);
    }

/*******************************************************
*Öffnen der separierten Druckdaten zum Schreiben der neu
* definierten Werte.
*******************************************************/
void WriteP(FILE *streamp, char DruckName[MAX_FILE])
{
    if((streamp=fopen(DruckName,"w"))==NULL){
                MbAlert("Eingabe");
                exit(1);
                }
    for(k=pAn;k<pEn;k++)
        {
            fprintf(streamp,"%.4lf\n",x1[k]);
            }
    fclose(streamp);
    }
```

```
/********************************************************
*Erzeugt eine zweispaltige Datei, in der die erste Spalte
*die Zeit ist und die zweit die Funktionswerte aus dem x1[]
*Vektor enthält.
********************************************************/
void WriteACF(FILE *streamp, char DruckName[MAX_FILE])
{
    if((streamp=fopen(DruckName,"w"))==NULL){
                MbAlert("Eingabe");
                exit(1);
                }
    for(k=pAn;k<pEn;k++)
        {
            fprintf(streamp,"%.4lf %.4lf\n",k*delta_t, x1[k]);
            }
    fclose(streamp);
    }

/********************************************************
*Öffnen der separierten ACF-Werte zum weiterem Bearbeiten.
********************************************************/
void OpenACF(FILE *stream, char ACFName[MAX_FILE])
{
    FILE *streamProxy;
    char NameProxy[MAX_FILE]="proxy.txt";

                if((streamProxy=fopen(NameProxy,"w"))==NULL){
                MbAlert("Eingabe");
                exit(1);
                }
                if((stream=fopen(ACFName,"r"))==NULL){
                MbAlert("Eingabe");
                exit(1);
                }


    fgets(ch1,MAX_EINGABE,stream);

    while(!feof(stream))
        {
            for(j=0; ch1[j] !=' ' ; j++)
                {
                    /*Die erste Spalte, mit der Zeilennr., ueberspringen.*/
                    }

            for(k=j+1;ch1[k]!='\n'; k++)
                {
                    fprintf(streamProxy,"%c",ch1[k]);
                    }
            fprintf(streamProxy,"\n");
            fgets(ch1,MAX_EINGABE,stream);
                }
    fclose(streamProxy);
    fclose(stream);
    OpenP(streamProxy, NameProxy);
    remove("\\proxy.txt");

    }

/********************************************************
*Funktionen um den minimalen/maximalen Wertebereich der Daten
*zu erfassen
```

```
*****************************************************/
void MinSearch(void)
{
    lowest=x1[0];
    for(z=1; z<number;z++)
        {

            if(x1[z]<lowest)
                {
                    lowest=x1[z];
                    }
            }
    }

void MaxSearch(void)
{
    highest=x1[0];
    for(z=1; z<number;z++)
        {
            if(x1[z]>highest)
                {
                    highest=x1[z];
                    }
            }

    }
/**************************************
*Erzeugt die Achsenbeschriftung.
**************************************/
void AchsenBeschriftung(void)
{
        switch(AxisTyp)
        {
        case 't':
            {
            for(k=0;k<=10;k++)
                {
                xBeschriftung[k]=number/10*k*delta_t;
                }
            break;
            }

        case 'f':
            {
            for(k=0;k<=10;k++)
                {
                xBeschriftung[k]=nyquistfrequenz/10*k;
                }
            break;
            }
        case 'n':
            {
            for(k=0;k<=10;k++)
                {
                xBeschriftung[k]=number/10*k;
                }
            break;
            }
        }
    }
```

```
/*******************************************
*Funktion um den physikalischen Messbereich
*auf die Bildschirmgröße zu transformieren
*******************************************/
void LimitsX(void)
{
        DeltaX=(ixMax-ixMin)/number;
    }

void LimitsY(void)
{
DeltaY=(iyMax-iyMin)/(highest-lowest);

    for(z=0;z<number;z++)
        {
            x1[z]=(x1[z]-lowest)*DeltaY;
            }
    }
/*********************************************************
* Diese Funktion ruft alle Funktionen auf die für die
* Autoskalierung des Messdatenbereichs auf den Bildschirm-
*bereich nötig sind und lässt dann die Daten zeichnen.
*********************************************************/
void CallToPaint(void)
{
        Messzeit();
        MinSearch();
        MaxSearch();
        LimitsX();
        LimitsY();
        AchsenBeschriftung();
        InvalidateRect(hWnd,NULL,TRUE);
    }

void CreateName(void)
{
    oversampling=scanrate/scanfrequency;
    TruePoints=oversampling*number;
    TotalTime=TruePoints/scanrate;
    delta_t=TotalTime/number;
    strcpy(TestData,druckfile);
    strcat(TestData,".txt");
    }

/*************************************************
*The implemented testfunctions.
*************************************************/
void SineCalc(void)
{
    double zeit;
    for(k=0; k<number; k++)
    {
        zeit=delta_t*k;
        x1[k]=FirstPar*sin(2*PI*SecondPar*zeit);
        }
    }

void DiracCalc(void)
{
    double zeit;
    x1[0]=FirstPar;
    for(k=1; k<number; k++)
```

```
    {
        zeit=delta_t*k;
        x1[k]=0;
        }
    }

void ParabolaCalc(void)
{
    double zeit;
    for(k=0; k<number; k++)
    {
        zeit=delta_t*k;
        x1[k]=FirstPar+SecondPar*zeit+ThirdPar*zeit*zeit;
        }
    }

void ExpCalc(void)
{
    double zeit;
    for(k=0; k<number; k++)
    {
        zeit=delta_t*k;
        x1[k]=FirstPar*exp(-zeit/SecondPar);
        }
    }

void RandCalc(void)
{
    srand(1);
    int RandNumb;
    RandNumb=rand();
    for(k=1; k<number; k=k+3)
    {
        x1[k]=FirstPar*(rand()-RandNumb)/10000;
        }
srand(2);
    for(k=0; k<number; k=k+3)
    {
        x1[k]=FirstPar*(rand()-RandNumb)/10000;
        }
srand(3);
    for(k=2; k<number; k=k+3)
    {
        x1[k]=FirstPar*(rand()-RandNumb)/10000;
        }
    }

void ExpFold(void)
{
    double  tau_exp;
    tau_exp=number/4.5;
    for(k=0;k<number;k++)
        {
            x1[k]=x1[k]*exp(-k/tau_exp);
            }
    }

void WriteDataFile(void)
{
    if((streamtestdata=fopen(TestData,"w"))==NULL){
                MbAlert("Eingabe");
                exit(1);
```

```
                }
        fprintf(streamtestdata,
        "%.2lf  %.2lf  %.2lf\n",scanrate,scanfrequency, scanfrequency/2);
    for(k=0;k<number;k++)
        {
        fprintf(streamtestdata,
        "%.8lf  %.8lf   %.8lf\n",10*x1[k],5*x1[k],1*x1[k]);
        }
    fclose(streamtestdata);
    }


/*************************************************
*Funktionen, um eine Regressionskurve zu bestimmen.
*************************************************/

/****************************************
*Funktionen, um die nötigen Mittelwerte
*zu Berechnen.
****************************************/
void CalcMean(void)
{
    pAn=0;
    pEn=number;
    for(k=0;k<number;k++)
    {
        meanX=meanX+k;
    }
    for(k=0;k<number;k++)
        {
            meanY=meanY+x1[k];
            }
    meanX=meanX/number;
    meanY=meanY/number;
    }

/****************************************
*Funktionen, um die Konstanten a,b für eine
*Regressionsgerade zu bestimmen.
****************************************/
void CalcParameterLin(void)
{
    ConstA=0;
    ConstB=0;
    double Nenner=0, Zaehler=0;
    for(k=0;k<number;k++)
        {
            Zaehler=Zaehler+(k-meanX)*(x1[k]-meanY);
            Nenner=Nenner+(k-meanX)*(k-meanX);
            }
    ConstB=Zaehler/Nenner;
    ConstA=meanY-ConstB*meanX;
    }

void QRMatrix(char matrixA[MAX_EINGABE], char vectorb[MAX_EINGABE] )
{
    FILE *streamA, *streamb;
    if((streamA=fopen(matrixA,"r"))==NULL)
        {
            MbAlert("Eingabe");
            exit(1);
                }
```

```
    if((streamb=fopen(vectorb,"r"))==NULL){
            MbAlert("Eingabe");
            exit(1);
                }
    for(k=0;k<3;k++)
        {
        fscanf(streamA,"%s",text);
        QR1[0][k]=atof(text);
            }
    j=1;
    while(!feof(streamA))
    {
        for(k=0;k<3;k++)
        {
        fscanf(streamA,"%s",text);
        QR1[j][k]=atof(text);
        }
        j++;
        }
    fscanf(streamb,"%s",text);
    bQR[0]=atof(text);
    k=1;
    while(!feof(streamb))
    {
        fscanf(streamb,"%s",text);
        bQR[k]=atof(text);
        k++;
        }
    number=k;
    fclose(streamA);
    fclose(streamb);
    }

void QRSolve(char QRname[MAX_EINGABE])
{
    FILE *stream;
    stream=fopen(QRname,"a");
    fprintf(stream,"\n----------------------------\n
     y= %.4lf+%.4lf*x+%.4lf*x^2 \n",QRx1,QRx2,QRx3);
    fclose(stream);
    }

void QuadProblem(void)
{
    for(k=0; k<number;k++)
        {
            QR1[k][0]=1;
            QR1[k][1]=k;
            QR1[k][2]=k*k;
            bQR[k]=x1[k];
            }
    }


void QuadFit(void)
{
    FILE *streamtest;
    for(j=0;j<3;j++)
        {
            signQR=1;
            AbsValQR=QR1[j][j];
            if(QR1[j][j]<0)
```

```
                {
                    AbsValQR=-QR1[j][j];
                    signQR=-1;
                    }
            BetragQR=0;
            for(k=j;k<number;k++)
                {
                    BetragQR=BetragQR+QR1[k][j]*QR1[k][j];
                    }
            BetragQR=sqrt(BetragQR);
            U1[j]=signQR*(AbsValQR+BetragQR);
            for(k=j+1;k<number;k++)
                {
                    U1[k]=QR1[k][j];
                    }
            NennerQR=0;
            for(k=j;k<number;k++)
                {
                    NennerQR=NennerQR+U1[k]*U1[k];
                    }
            betaQR=2/NennerQR;
            for(l=j;l<3;l++)
                {
                    SkalarQR[l]=0;
                    for(k=j;k<number;k++)
                        {
                            SkalarQR[l]=SkalarQR[l]+U1[k]*QR1[k][l];
                            }
                    SkalarQR[l]=SkalarQR[l]*betaQR;
                    for(k=j;k<number;k++)
                        {
                            QR1[k][l]=QR1[k][l]-SkalarQR[l]*U1[k];
                            }
                    }
            SkalarQR[3]=0;
            for(k=j;k<number;k++)
                {
                    SkalarQR[3]=SkalarQR[3]+U1[k]*bQR[k];
                    }
            SkalarQR[3]=SkalarQR[3]*betaQR;
            for(k=j;k<number;k++)
                {
                    bQR[k]=bQR[k]-SkalarQR[3]*U1[k];
                    }
            }
    QRx3=bQR[2]/QR1[2][2];
    QRx2=(bQR[1]-QR1[1][2]*QRx3)/QR1[1][1];
    QRx1=(bQR[0]-QR1[0][2]*QRx3-QR1[0][1]*QRx2)/QR1[0][0];
    streamtest=fopen(textQR,"w");
    for(k=0;k<number;k++)
            {
    fprintf(streamtest,"%.2lf   %.2lf   %.2lf   | %.2lf\n",QR1[k][0]
    ,QR1[k][1],QR1[k][2],bQR[k]);
                    }
    fclose(streamtest);
    }

/****************************************
*Diese Funktion überschreibt die ursprünglichen
*Druckdaten mit den neuen gefitteten Werten.
****************************************/
void SubtractCurve(void)
```

```
{
    for(k=0;k<number;k++)
        {
            x1[k]=x1[k]-ConstA-ConstB*k;
            }
    }

void SubtractQuad(void)
{
    for(k=0;k<number;k++)
        {
            x1[k]=x1[k]-QRx1-QRx2*k-QRx3*k*k;
            }
    }

/*********************************************************
*Diese Funktion öffnet die Datei zum speichern von zusätz-
*lichen Informationen. Wie Berechnungsdauer, Mittelwerte,
*etc..
*********************************************************/
void OpenInfo(void)
{
    if((streaminfo=fopen(infoausgabe,"w"))==NULL){
        MbAlert("Eingabe");
        exit(1);
        }

    for(k=0;k<30;k++)
        {
            fprintf(streaminfo,"%s%s\n",InfoNamen[k], InfoWerte[k]);
            }
    fclose(streaminfo);
    }

void InfoHeader(void)
{
        sprintf(InfoWerte[0],"%s",druckfile1);
        sprintf(InfoWerte[1],"%.4lf",scanrate);
        sprintf(InfoWerte[2],"%.4lf",oversampling);
        sprintf(InfoWerte[3],"%.4lf",nyquistfrequenz);
        sprintf(InfoWerte[4],"%i",number);
        sprintf(InfoWerte[24],"%.4lf",delta_f);
        sprintf(InfoWerte[25],"%.4lf",t_max);
        sprintf(InfoWerte[26],"%.4lf",delta_t);
    }

void NamenInfo(void)
{
sprintf(InfoNamen[0],"%s","Dateiname:   ");
sprintf(InfoNamen[1],"%s","\nScanrate[pts/s]:   ");
sprintf(InfoNamen[2],"%s","Oversampling[pts]:   ");
sprintf(InfoNamen[3],"%s","Nyquistfrequency[1/s]:   ");
sprintf(InfoNamen[4],"%s","Initialy Measured Data points[pts]:  ");
sprintf(InfoNamen[5],"%s","\nFor analysis used starting point:      ");
sprintf(InfoNamen[6],"%s","For analysis used ending point:          ");
sprintf(InfoNamen[7],"%s","For analysis used amount of points[pts]: ");
sprintf(InfoNamen[8],"%s","\nOrder of polynom fitted:   ");
sprintf(InfoNamen[9],"%s","Equation of polynom(1):  ");
sprintf(InfoNamen[10],"%s","Equation of polynom(2): ");
sprintf(InfoNamen[11],"%s","Equation of polynom(3): ");
sprintf(InfoNamen[12],"%s","\nAutocorrelationtime(1)[s]:    ");
sprintf(InfoNamen[13],"%s","Autocorrelationtime(2)[s]:  ");
```

```
sprintf(InfoNamen[14],"%s","Autocorrelationtime(3)[s]:  ");
sprintf(InfoNamen[15],"%s","\nAutocovariationtime(1)[s]:    ");
sprintf(InfoNamen[16],"%s","Autocovariationtime(2)[s]:  ");
sprintf(InfoNamen[17],"%s","Autocovariationtime(3)[s]:  ");
sprintf(InfoNamen[18],"%s","\nSpectral Power Density time(1)[s]:    ");
sprintf(InfoNamen[19],"%s","Spectral Power Density time(2)[s]:  ");
sprintf(InfoNamen[20],"%s","Spectral Power Density time(3)[s]:  ");
sprintf(InfoNamen[21],"%s","\nFT time(1)[s]:    ");
sprintf(InfoNamen[22],"%s","FT time(2)[s]:  ");
sprintf(InfoNamen[23],"%s","FT time(3)[s]:  ");
sprintf(InfoNamen[24],"%s","\ndelta f[1/s]:          ");
sprintf(InfoNamen[25],"%s","delta t [s]:             ");
sprintf(InfoNamen[26],"%s","Maximum Measuring time[s]:  ");
sprintf(InfoNamen[27],"%s","\nStandartdeviation(1): ");
sprintf(InfoNamen[28],"%s","Standartdeviation(2):   ");
sprintf(InfoNamen[29],"%s","Standartdeviation(3):   ");
    }

/*******************************************************
* Hauptprogramm mit Message- Loop
*******************************************************/
int WINAPI WinMain(HINSTANCE hInstance,
                   HINSTANCE hPrevInstance,
                   PSTR szCmdLine, int iCmdShow)
{
extern HINSTANCE hProgram;
extern HWND      hWnd;
extern int       ixClient, iyClient;
MSG              msg;
WNDCLASSEX       wndclass;
char             capname[4];
/*-----------------------------------------------------
* Window registrieren
*-----------------------------------------------------*/
hProgram= hInstance;

strcpy(capname, "ACF");

wndclass.cbSize        = sizeof(wndclass);
wndclass.style         = CS_HREDRAW | CS_VREDRAW;
wndclass.lpfnWndProc   = W5_WinProc;
wndclass.cbClsExtra    = 0;
wndclass.cbWndExtra    = 0;
wndclass.hInstance     = hProgram;
wndclass.hIcon         = LoadIcon(hProgram,
                           MAKEINTRESOURCE(ICO_ACF));
wndclass.hCursor       = LoadCursor(NULL, IDC_ARROW);
wndclass.hbrBackground = CreateSolidBrush(
                           RGB(255,255,255));
wndclass.lpszMenuName  = capname;
wndclass.lpszClassName = capname;
wndclass.hIconSm       = NULL;
RegisterClassEx(&wndclass);
/*-------------------------------------------
* Window erzeugen
*-------------------------------------------*/
hWnd = CreateWindow(capname,
                    "Pressure Data Analyser",
                    WS_OVERLAPPEDWINDOW,
                    50,10,
                    ixClient,iyClient,
                    NULL, NULL, hProgram, NULL);
```

```

/*-----------------------------------------------------------
* Window darstellen und updaten
*-----------------------------------------------------------*/
ShowWindow(hWnd,iCmdShow);
UpdateWindow(hWnd);

/*-----------------------------------------------------------
* Message- Loop: Ereigneisse abfangen
*-----------------------------------------------------------*/
while(GetMessage(&msg, NULL, 0, 0))
  {
  TranslateMessage(&msg);
  DispatchMessage(&msg);
  }
return msg.wParam;
}

/************************************************************
* Main Window Procedure: Das eigentliche Steuerprogramm
************************************************************/
LRESULT CALLBACK W5_WinProc(HWND hWnd, UINT Message,
                            WPARAM wParam, LPARAM lParam)
{
extern HDC         hDC;
extern int         ixClient, iyClient;
PAINTSTRUCT        ps;

HPEN               hpen_linie,hpen_function;

/*-----------------------------------------------------------
* Los gehts - je nach Message verzweigen
*-----------------------------------------------------------*/

switch(Message)
  {
/*===========================================================
* WM_COMMAND: Wenn Befehle aus dem Menue kommen, dann
*===========================================================*/
  case WM_COMMAND:
    switch (LOWORD(wParam))
      {
      case IDM_LOAD:
        DialogBox(hProgram,"Eingabe",hWnd,W2_DiaProc);
        MbInfo(druckfile,"Load");
        Eingabe();
        LookFile(streamdruck);
        Messzeit();
        InfoHeader();
        return 0;

    case IDM_CREATE:
        DialogBox(hProgram,"Eingabe",hWnd,W2_DiaProc);
        MbInfo(druckfile,"Create");
        Eingabe();
        Separate(streamdruck, streamausgabed1, streamausgabed2, streamausgabed3);
        Messzeit();
        InfoHeader();
      return 0;

      case IDM_CHANGE:
        AxisTyp='n';
```

```
            sprintf(Beschriftung,"number [pts]");
            MbInfo("Plotting Pressure Data 1","Change Range");
            OpenP(streamausgabed1 , dausgabe1);
            CallToPaint();
            sprintf(pAnfang,"0");
            sprintf(pEnde,"%i",number);
            DialogBox(hProgram,"Beschneide",hWnd,Beschneiden_DiaProc);
            OpenP(streamausgabed1 , dausgabe1);
            WriteP(streamausgabed1 , dausgabe1);
            OpenP(streamausgabed2 , dausgabe2);
            WriteP(streamausgabed2 , dausgabe2);
            OpenP(streamausgabed3 , dausgabe3);
            WriteP(streamausgabed3 , dausgabe3);
            OpenP(streamausgabed1 , dausgabe1);
            CallToPaint();
            sprintf(InfoWerte[5],"%i",pAn);
            sprintf(InfoWerte[6],"%i",pEn);
            sprintf(InfoWerte[7],"%i",pEn-pAn);
          return 0;

          case IDM_CURVE:
             MbInfo("Fitting the data with a linear function","Curve Fit");
            OpenP(streamausgabed1 , dausgabe1);
            CalcMean();
            CalcParameterLin();
            SubtractCurve();
            WriteP(streamausgabed1,dausgabe1);
            WriteP(streamlcf1,dausgabelcf1);
            sprintf(text,"y(t)=%.4lf + (%.4lf)*t",ConstA,ConstB);
            sprintf(InfoWerte[9],"%s",text);

            OpenP(streamausgabed2 , dausgabe2);
            CalcMean();
            CalcParameterLin();
            SubtractCurve();
            WriteP(streamausgabed2,dausgabe2);
            WriteP(streamlcf2,dausgabelcf2);
            sprintf(text,"y(t)=%.4lf + (%.4lf)*t",ConstA,ConstB);
            sprintf(InfoWerte[10],"%s",text);
            OpenP(streamausgabed3 , dausgabe3);
            CalcMean();
            CalcParameterLin();
            SubtractCurve();
            WriteP(streamausgabed3,dausgabe3);
            WriteP(streamlcf3,dausgabelcf3);
            sprintf(text,"y(t)=%.4lf + (%.4lf)*t",ConstA,ConstB);
            sprintf(InfoWerte[11],"%s",text);
            MbInfo("Operation finished","Curve Fit");
          return 0;

           case IDM_QUADF:
            pAn=0;
            pEn=number;
            MbInfo("Fitting the data with a parabola","Quadratic Fit");
            OpenP(streamausgabed1 , dausgabe1);
            sprintf(textQR,"%s",druckfile);
            strcat(textQR,"QR1.txt");
            QuadProblem();
            QuadFit();
            QRSolve(textQR);
            SubtractQuad();
            WriteP(streamausgabed1,dausgabe1);
```

```
         WriteP(streamlcf1,dausgabelcf1);
         sprintf(text,"y(t)=%.4lf + (%.4lf)*t+ (%.4lf)*t^2",QRx1,QRx2,QRx3);
         sprintf(InfoWerte[9],"%s",text);
         OpenP(streamausgabed2, dausgabe2);
         sprintf(textQR,"%s",druckfile);
         strcat(textQR,"QR2.txt");
         QuadProblem();
         QuadFit();
         QRSolve(textQR);
         SubtractQuad();
         WriteP(streamausgabed2,dausgabe2);
         WriteP(streamlcf2,dausgabelcf2);
         sprintf(text,"y(t)=%.4lf + (%.4lf)*t+ (%.4lf)*t^2",QRx1,QRx2,QRx3);
         sprintf(InfoWerte[10],"%s",text);
         OpenP(streamausgabed3, dausgabe3);
         sprintf(textQR,"%s",druckfile);
         strcat(textQR,"QR3.txt");
         QuadProblem();
         QuadFit();
         QRSolve(textQR);
         SubtractQuad();
         WriteP(streamausgabed3,dausgabe3);
         WriteP(streamlcf3,dausgabelcf3);
         sprintf(text,"y(t)=%.4lf + (%.4lf)*t+ (%.4lf)*t^2",QRx1,QRx2,QRx3);
         sprintf(InfoWerte[11],"%s",text);
          MbInfo("Operation finished","Quadratic Fit");
       return 0;

     case IDM_FOLD:
         pAn=0;
         pEn=number;
          MbInfo("Multiplying the ACF values with exp(-t/(t_max/4))",
          "Windowing");
         OpenACF(streamacf1, ausgabe1acf);
         WriteACF(streamacff1,ausgabe1acff);
         ExpFold();
         WriteACF(streamacf1,ausgabe1acf);
         OpenACF(streamacf2, ausgabe2acf);
         WriteACF(streamacff2,ausgabe2acff);
         ExpFold();
         WriteACF(streamacf2,ausgabe2acf);
         OpenACF(streamacf3, ausgabe3acf);
         WriteACF(streamacff3,ausgabe3acff);
         ExpFold();
         WriteACF(streamacf3,ausgabe3acf);
         return 0;

       case IDM_ACF:
          MbInfo("Autocorrelation is started","ACF");
         OpenDruckACF();
         Messzeit();
         sprintf(text,"The Data Aquisitiontime has been %.3lfs", t_max);
          MbInfo(text,"ACF");
         column=1;
         Transducer(streamausgabed1);
         ACF(streamacf1);
         sprintf(InfoWerte[12],"%i",difft);
         sprintf(InfoWerte[27],"%.4lf",tau_null);
         column=2;
         Transducer(streamausgabed2);
         ACF(streamacf2);
         sprintf(InfoWerte[13],"%i",difft);
```

```
        sprintf(InfoWerte[28],"%.4lf",tau_null);
        column=3;
        Transducer(streamausgabed3);
        ACF(streamacf3);
        sprintf(InfoWerte[14],"%i",difft);
        sprintf(InfoWerte[29],"%.4lf",tau_null);
        MbInfo(" Autocorrelation finished","ACF");
        fclose(streamausgabed1);
        fclose(streamausgabed2);
        fclose(streamausgabed3);
      return 0;

      case IDM_ACOV:
           MbInfo("Autocovariation is started","ACOV");
        OpenDruckACOV();
        Messzeit();
        sprintf(text,"The Data Aquisitiontime has been %.3lfs", t_max);
         MbInfo(text,"ACOV");
        column=1;
        Transducer(streamausgabed1);
        ACOV(streamacov1);
        sprintf(InfoWerte[15],"%i",difft);
        fclose(streamacov1);
        column=2;
        Transducer(streamausgabed2);
        ACOV(streamacov2);
        sprintf(InfoWerte[16],"%i",difft);
        fclose(streamacov2);
        column=3;
        Transducer(streamausgabed3);
        ACOV(streamacov3);
        sprintf(InfoWerte[17],"%i",difft);
        fclose(streamacov3);
        MbInfo("Autocovariation finished","ACOV");
        fclose(streamausgabed1);
        fclose(streamausgabed2);
        fclose(streamausgabed3);
        return 0;

      case IDM_SPD:
         MbInfo("Spectral Power Density is started (SPD)","SPD");
        OpenLDS();
        column=1;
        Extract(streamacf1,streamlds1,ausgabe1acf,ausgabe1lds);
        RFT(streamlds1,ausgabe1lds);
        sprintf(InfoWerte[18],"%i",difft);

        column=2;
        Extract(streamacf2,streamlds2,ausgabe2acf,ausgabe2lds);
        RFT(streamlds2,ausgabe2lds);
        sprintf(InfoWerte[19],"%i",difft);

        column=3;
        Extract(streamacf3,streamlds3,ausgabe3acf,ausgabe3lds);
        RFT(streamlds3,ausgabe3lds);
        sprintf(InfoWerte[20],"%i",difft);

        MbInfo(" Spectral Power Density finished","SPD");
      return 0;

        case IDM_PLOTDRUCK:
        AxisTyp='t';
```

```
        sprintf(yAchse,"Druck [Pa]");
        sprintf(Beschriftung,"time [s]");
          MbInfo("Column 1","Plot Pressure");
         OpenP(streamausgabed1, dausgabe1);
         CallToPaint();
           MbInfo("Column 2","Plot Pressure");
         OpenP(streamausgabed2, dausgabe2);
         CallToPaint();
           MbInfo("Column 3","Plot Pressure");
        OpenP(streamausgabed3, dausgabe3);
        CallToPaint();
          return 0;


        case IDM_PLOTACF:
            AxisTyp='t';
        sprintf(yAchse,"ACF magnitude");
        sprintf(Beschriftung,"correlation time [s]");
           MbInfo("Column 1 ","Plot ACF");
         OpenACF(streamacf1, ausgabe1acf);
         CallToPaint();
           MbInfo("Column 2 ","Plot ACF");
         OpenACF(streamacf2, ausgabe2acf);
         CallToPaint();
           MbInfo("Column 3 ","Plot ACF");
         OpenACF(streamacf3, ausgabe3acf);
        CallToPaint();
          return 0;

        case IDM_PLOTACOV:
            AxisTyp='t';
        sprintf(yAchse,"ACOV magnitude");
        sprintf(Beschriftung,"correlation time [s]");
           MbInfo("Column 1 ","Plot ACOV");
         OpenACF(streamacov1, ausgabe1acov);
         CallToPaint();
           MbInfo("Column 2","Plot ACOV");
         OpenACF(streamacov2, ausgabe2acov);
         CallToPaint();
           MbInfo("Column 3","Plot ACOV");
         OpenACF(streamacov3, ausgabe3acov);
        CallToPaint();
        return 0;

    case IDM_PLOTSPD:
        AxisTyp='f';
        sprintf(yAchse,"SPD magnitude");
        sprintf(Beschriftung,"frequency [1/s]");
           MbInfo("Column 1","Plot SPD");
         OpenACF(streamlds1, ausgabe1lds);
         CallToPaint();
           MbInfo("Column 2","Plot SPD");
         OpenACF(streamlds2, ausgabe2lds);
         CallToPaint();
       MbInfo("Column 3","Plot SPD");
         OpenACF(streamlds3, ausgabe3lds);
        CallToPaint();
        return 0;

    case IDM_GSINUS:
             MbInfo(" Creating data from a Sine function y=a*sin(2Pi*b*x)",
             "Sine");
```

```
        FunctionTyp=2;
        sprintf(FunctionName,"Sine");
        sprintf(druckfile,"sinus");
        DialogBox(hProgram,"DataAquisition",hWnd,TESTDATA_DiaProc);
        DialogBox(hProgram,"Eingabe",hWnd,W2_DiaProc);
        sprintf(a_value,"Magnitude");
        sprintf(b_value,"Frequency [1/s]");
        DialogBox(hProgram,"Quadratic",hWnd,TESTSPECIAL_DiaProc);
        CreateName();
        SineCalc();
        WriteDataFile();
        return 0;

    case IDM_GPARABOLA  :
         MbInfo(" Creating data from a Parabola y=a+b*x+c*x^2","Parabola");
        FunctionTyp=3;
        sprintf(FunctionName,"Parabola");
        DialogBox(hProgram,"DataAquisition",hWnd,TESTDATA_DiaProc);
        sprintf(druckfile,"parabola");
        DialogBox(hProgram,"Eingabe",hWnd,W2_DiaProc);
        sprintf(a_value,"Off-Set");
        sprintf(b_value,"Slope");
        sprintf(c_value,"Curvature");
        DialogBox(hProgram,"Kubic",hWnd,TESTSPECIAL_DiaProc);
        CreateName();
        ParabolaCalc();
        WriteDataFile();
        return 0;

    case IDM_GDIRAC :
         MbInfo(" Creating data from a Dirac function","Dirac");
        FunctionTyp=1;
        sprintf(FunctionName,"Dirac-Impuls");
        DialogBox(hProgram,"DataAquisition",hWnd,TESTDATA_DiaProc);
        sprintf(druckfile,"dirac");
        DialogBox(hProgram,"Eingabe",hWnd,W2_DiaProc);
        sprintf(a_value,"Magnitude");
        DialogBox(hProgram,"Linear",hWnd,TESTSPECIAL_DiaProc);
        CreateName();
        DiracCalc();
        WriteDataFile();
        return 0;

    case IDM_GEXPONENTIAL   :
         MbInfo(" Creating data from a Exponential function y=a*exp(-b/x)",
         "Exponential");
        FunctionTyp=2;
        sprintf(FunctionName,"exp");
        DialogBox(hProgram,"DataAquisition",hWnd,TESTDATA_DiaProc);
        sprintf(druckfile,"exponential");
        DialogBox(hProgram,"Eingabe",hWnd,W2_DiaProc);
        sprintf(a_value,"Magnitude");
        sprintf(b_value,"Decay");
        DialogBox(hProgram,"Quadratic",hWnd,TESTSPECIAL_DiaProc);
        CreateName();
        ExpCalc();
        WriteDataFile();
        return 0;

    case IDM_GRANDOM:
         MbInfo(" Creating data from a Random","Random");
        FunctionTyp=1;
```

```
        sprintf(FunctionName,"rand");
        DialogBox(hProgram,"DataAquisition",hWnd,TESTDATA_DiaProc);
        sprintf(druckfile,"random");
        DialogBox(hProgram,"Eingabe",hWnd,W2_DiaProc);
        sprintf(a_value,"Magnitude");
        DialogBox(hProgram,"linear",hWnd,TESTSPECIAL_DiaProc);
        CreateName();
        RandCalc();
        WriteDataFile();
        return 0;

    case IDM_QR_TRANS:
        sprintf(matrixname,"MATRIX-A");
        sprintf(vektorname,"VECTOR-b");
        DialogBox(hProgram,"Matrixeingabe",hWnd,QR_MATRIX_DiaProc);
        sprintf(textQR,"%s",matrixname);
        strcat(textQR,"QR-TRANS.txt");
        strcat(matrixname,".txt");
        strcat(vektorname,".txt");
        QRMatrix(matrixname,vektorname);
        QuadFit();
        QRSolve(textQR);
        return 0;

    case IDM_INFO:
        NamenInfo();
        OpenInfo();
        return 0;

     case IDM_CLOSE:
       PostQuitMessage(0);
     return 0;

     }

/*==========================================================
* WM_PAINT: Wenn das Window neu gezeichnet werden soll,dann
*==========================================================*/
  case WM_PAINT:
    hDC= BeginPaint(hWnd, &ps);
    hpen_linie= CreatePen(PS_SOLID,1,RGB(0,55,55));
    hpen_function= CreatePen(PS_SOLID,1,RGB(255,0,0));
    ixMin= 50;
    ixMax= ixClient-20;
    iyMin= 20;
    iyMax= iyClient-50;
    SelectObject(hDC,hpen_linie);

//------------------------------------------------------------
//Y-Achse
//------------------------------------------------------------
    MoveToEx(hDC,ixMin,iyMin,NULL);
    LineTo  (hDC,ixMin,iyMax);

//------------------------------------------------------------
//X-Achse
//------------------------------------------------------------
    MoveToEx(hDC,ixMin,iyMax,NULL);
    LineTo  (hDC,ixMax,iyMax);

//------------------------------------------------------------
//Achsenunterteilung
```

```
//-----------------------------------------------------------
    int iyDelta=(iyMax-iyMin)/10;
    int ixDelta=(ixMax-ixMin)/10;
//-----------------------------------------------------------
//Beschriftung der y-Achse
//-----------------------------------------------------------
    for(k=0; k<=10;k++)
        {
            MoveToEx(hDC,ixMin,iyMax-k*iyDelta,NULL);
            LineTo  (hDC,ixMin-20,iyMax-k*iyDelta);
            sprintf(text,"%.1lf",lowest+(highest-lowest)*k/10);
            TextOut(hDC,ixMin-45,iyMax-k*iyDelta+5,text,strlen(text));
            }

//-----------------------------------------------------------
//Beschriftung der x-Achse
//-----------------------------------------------------------
    for(k=0; k<=10;k++)
        {
            MoveToEx(hDC,ixMin+k*ixDelta,iyMax,NULL);
            LineTo  (hDC,ixMin+k*ixDelta,iyMax+20);
            sprintf(text,"%.2lf",xBeschriftung[k]);
            TextOut(hDC,ixMin+k*ixDelta-15-k,iyMax+20,text,strlen(text));
            }
    TextOut(hDC,ixMax-90,iyMax-30,Beschriftung,strlen(Beschriftung));
    TextOut(hDC,ixMin-20,iyMin-20,yAchse,strlen(yAchse));
//-----------------------------------------------------------
//Graph zeichnen
//-----------------------------------------------------------
    SelectObject(hDC,hpen_function);
    x = ixMin+DeltaX*0;
    y =iyMax-x1[0];
    MoveToEx(hDC,x,y,NULL);
    for(i=1; i<number; i++)
      {
        x = ixMin+DeltaX*i;
        y = iyMax -x1[i];
        LineTo(hDC,x,y);
        MoveToEx(hDC,x,y,NULL);
      }
    DeleteObject(hpen_linie);
    DeleteObject(hpen_function);

    EndPaint(hWnd, &ps);
  return 0;

/*==========================================================
* WM_SIZE: Wenn das Window veraendert wird,dann
*==========================================================*/
  case WM_SIZE:
    iyClient= HIWORD(lParam);
    ixClient= LOWORD(lParam);
  return 0;

/*==========================================================
* WM_DESTROY: Wenn das Window zerstoert werden soll, dann
*==========================================================*/
  case WM_DESTROY:
    PostQuitMessage(0);
  return 0;

/*==========================================================
```

```
* Wenn keine der obigen Cases zutrifft, dann
*=======================================================*/
  default:
    return DefWindowProc(hWnd,Message,wParam,lParam);
  }
}

/*******************************************************
* Function W2_DiaProc zeigt Dialog- Box an
*******************************************************/
BOOL CALLBACK W2_DiaProc(HWND hDlg, UINT message,
                         WPARAM wParam, LPARAM lParam)
{
/*-------------------------------------------------------
* Dialog aufbauen
*-------------------------------------------------------*/
switch (message)
  {
  case WM_INITDIALOG:
    SetDlgItemText(hDlg,IDD_EINGABE,druckfile);
  return TRUE;

  case WM_COMMAND:
    switch (LOWORD(wParam))
      {
      case ID_INORDNUNG:
        GetDlgItemText(hDlg,IDD_EINGABE,druckfile,MAX_FILE);
        EndDialog(hDlg,0);
      return TRUE;
      case ID_UEBERGEHEN:
        EndDialog(hDlg,0);
      return TRUE;
      }
  }
return FALSE;
}

/*******************************************************
* Function Beschneiden_DiaProc zeigt Dialog- Box an
*******************************************************/
BOOL CALLBACK Beschneiden_DiaProc(HWND hDlg, UINT message,
                         WPARAM wParam, LPARAM lParam)
{
/*-------------------------------------------------------
* Dialog aufbauen
*-------------------------------------------------------*/
switch (message)
  {
  case WM_INITDIALOG:
    SetDlgItemText(hDlg,IDD_ANFANG,pAnfang);
    SetDlgItemText(hDlg,IDD_ENDE,pEnde);
  return TRUE ;
  case WM_COMMAND:
    switch (LOWORD(wParam))
      {
      case ID_INORDNUNG:
        GetDlgItemText(hDlg,IDD_ANFANG,pAnfang,30);
        GetDlgItemText(hDlg,IDD_ENDE,pEnde,30);
        EndDialog(hDlg,0);
        pAn=atoi(pAnfang);
        pEn=atoi(pEnde);
        if(pEnde[0]=='0')
```

```
            {
                pEn=number;
            }
      return TRUE;
      case ID_UEBERGEHEN:
        EndDialog(hDlg,0);
      return TRUE;
      }
  }

return FALSE;
}

/*********************************************************
* Function TESTDATA_DiaProc zeigt Dialog- Box an
*********************************************************/
BOOL CALLBACK TESTDATA_DiaProc(HWND hDlg, UINT message,
                         WPARAM wParam, LPARAM lParam)
{
/*--------------------------------------------------------
* Dialog aufbauen
*-------------------------------------------------------*/
switch (message)
  {
  case WM_INITDIALOG:
    SetDlgItemText(hDlg,IDD_SCANRATE,Sscanrate);
    SetDlgItemText(hDlg,IDD_SCANFREQUENCY,Sscanfrequency);
    SetDlgItemText(hDlg,IDD_POINTS,Snumber);
  return  TRUE;
  case WM_COMMAND:
    switch (LOWORD(wParam))
      {
      case ID_INORDNUNG:
        GetDlgItemText(hDlg,IDD_SCANRATE,Sscanrate,30);
        GetDlgItemText(hDlg,IDD_SCANFREQUENCY,Sscanfrequency,30);
        GetDlgItemText(hDlg,IDD_POINTS,Snumber,30);
        EndDialog(hDlg,0);
        scanrate=atof(Sscanrate);
        scanfrequency=atof(Sscanfrequency);
        number=atoi(Snumber);
      return TRUE;
      case ID_UEBERGEHEN:
        EndDialog(hDlg,0);
      return TRUE;
      }
  }
return FALSE;
}

/*********************************************************
* Function TESTSPECIAL_DiaProc zeigt Dialog- Box an
*********************************************************/
BOOL CALLBACK TESTSPECIAL_DiaProc(HWND hDlg, UINT message,
                         WPARAM wParam, LPARAM lParam)
{
/*--------------------------------------------------------
* Dialog aufbauen
*-------------------------------------------------------*/
switch (message)
  {
  case WM_INITDIALOG:
    SetDlgItemText(hDlg,IDD_A,a_value);
```

```
    if(FunctionTyp>1)
        {
        SetDlgItemText(hDlg,IDD_B,b_value);
        if(FunctionTyp>2)
                {
                SetDlgItemText(hDlg,IDD_C,c_value);
                }
        }
  return TRUE;
  case WM_COMMAND:
    switch (LOWORD(wParam))
      {
      case ID_INORDNUNG:
        GetDlgItemText(hDlg,IDD_A,a_value,MAX_FILE);
        FirstPar=atof(a_value);
        if(FunctionTyp>1)
        {
            GetDlgItemText(hDlg,IDD_B,b_value,MAX_FILE);
            SecondPar=atof(b_value);
            if(FunctionTyp>2)
                {
                GetDlgItemText(hDlg,IDD_C,c_value,MAX_FILE);
                ThirdPar=atof(c_value);
                }
        }
        EndDialog(hDlg,0);
      return TRUE;
      case ID_UEBERGEHEN:
        EndDialog(hDlg,0);
      return TRUE;
      }
  }
return FALSE;
}

/*********************************************************
* Function QR_MATRIX_DiaProc zeigt Eingabe-Box für die
* Eingabe einerin einer Datei gespeicherten Matrix und
* eines Vektors.
*********************************************************/
BOOL CALLBACK QR_MATRIX_DiaProc(HWND hDlg, UINT message,
                         WPARAM wParam, LPARAM lParam)
{
/*---------------------------------------------------------
* Dialog aufbauen
*---------------------------------------------------------*/
switch (message)
  {
  case WM_INITDIALOG:
    SetDlgItemText(hDlg,IDD_MATRIX,matrixname);
    SetDlgItemText(hDlg,IDD_VEKTOR,vektorname);
  return TRUE ;
  case WM_COMMAND:
    switch (LOWORD(wParam))
      {
      case ID_INORDNUNG:
        GetDlgItemText(hDlg,IDD_MATRIX,matrixname,MAX_EINGABE);
        GetDlgItemText(hDlg,IDD_VEKTOR,vektorname,MAX_EINGABE);
        EndDialog(hDlg,0);
      return TRUE;
      case ID_UEBERGEHEN:
        EndDialog(hDlg,0);
```

```
      return TRUE;
      }
  }
return FALSE;
}
```

Listing B.3: *ACF.c*

```
/*********************************************************
* ACF.h
* 03.03.2007 I.F.C. Naue
*********************************************************/
#define MAX_EINGABE 100
#define MAX_FILE 67
#define PI 3.14159265358979323846264338327950

#define ICO_ACF                     10

#define IDM_CREATE              100
#define IDM_ENDE                    110
#define IDM_WERISTS             120
#define IDM_LOAD                    130
#define IDM_CHANGE              140
#define IDM_CLOSE                   150
#define IDM_CURVE                   160

#define IDM_ACF                     170
#define IDM_ACOV                    180
#define IDM_SPD                     200
#define IDM_PLOTDRUCK           210
#define IDM_PLOTACF             220
#define IDM_PLOTSPD             230
#define IDM_PLOTACOV            240
#define IDM_QUADF                   260
#define IDM_QR_TRANS            270
#define IDM_FOLD                    280
#define IDM_INFO                    290

#define IDD_ANFANG              400
#define IDD_ENDE                    410

#define IDD_EINGABE             420
#define ID_UEBERGEHEN           430
#define ID_INORDNUNG            440

#define IDM_GSINUS              600
#define IDM_GDIRAC              610
#define IDM_GEXPONENTIAL    620
#define IDM_GPARABOLA           630
#define IDM_GRANDOM             640

#define IDD_A                           700
#define IDD_B                           710
#define IDD_C                           720
#define IDD_SCANRATE            730
#define IDD_SCANFREQUENCY   740
#define IDD_POINTS              750
#define IDD_MATRIX              760
#define IDD_VEKTOR              770
```

Listing B.4: *ACF.h*

```
/*******************************************************
* ACF.RC
* 12-12-2006 Naue
*******************************************************/
#include <ACF.h>
#include <windows.h>

ICO_ACF ICON ACFLDS01.ICO

/********************************
*Popup Menu definition.
*********************************/

ACF MENU
  {
  POPUP  "&Raw Data"
    {
    MENUITEM "&Load",                   IDM_LOAD
    MENUITEM "&Create",                 IDM_CREATE

    MENUITEM "&Close",                  IDM_CLOSE
    }
    POPUP   "&Makeup"
    {
    MENUITEM "&Change Range",                   IDM_CHANGE
    MENUITEM "&Curve Fit",                      IDM_CURVE
    MENUITEM "&Quadratic Fit",                  IDM_QUADF
    MENUITEM "&ACF with e-Fct.",                IDM_FOLD
    }

  POPUP "&Analyse"
  {

    MENUITEM "&ACF",            IDM_ACF
    MENUITEM "&ACOV",           IDM_ACOV
    MENUITEM "&SPD",            IDM_SPD
    }
    POPUP "&View"
    {
    MENUITEM"&Pressure View"    IDM_PLOTDRUCK
    MENUITEM"&ACF View"         IDM_PLOTACF
    MENUITEM"&ACOV View"        IDM_PLOTACOV
    MENUITEM"&SPD View"         IDM_PLOTSPD
    }
    POPUP "&Testdata"
    {
    MENUITEM"&Sine"         IDM_GSINUS
    MENUITEM"&Dirac"            IDM_GDIRAC
    MENUITEM"&Exponential"      IDM_GEXPONENTIAL
    MENUITEM"&Parabola"         IDM_GPARABOLA
    MENUITEM"&Random"           IDM_GRANDOM
    }
    POPUP"&Matrix OP's"
    {
    MENUITEM"&QR-Transformation"    IDM_QR_TRANS
    }
    POPUP"&Info File"
    {
    MENUITEM"&Create Info"  IDM_INFO
    }
  }
//-----------------------------------------
```

```
//
//Dialogbox for entering the sourcefiles name.
//
//-----------------------------------------
Eingabe DIALOG DISCARDABLE 50, 50, 160, 110

STYLE DS_MODALFRAME | WS_MINIMIZEBOX | WS_MAXIMIZEBOX |
      WS_CAPTION    | WS_SYSMENU

CAPTION "Source File"

FONT 10, "Arial"

BEGIN
    LTEXT       "Please enter the name of the source file.", -1, 10, 10, 140,10
    LTEXT         "file name:",  -1, 11, 23, 30, 10

    EDITTEXT      IDD_EINGABE, 43, 21, 70, 12, ES_AUTOHSCROLL

    DEFPUSHBUTTON "OK", ID_INORDNUNG,  33,80, 44,14
    PUSHBUTTON    "Cancel", ID_UEBERGEHEN, 88,80, 44,14
END

//-----------------------------------------
//
//Dialogbox for entering the range of interest.
//
//-----------------------------------------
Beschneide DIALOG DISCARDABLE 50, 50, 160, 140

STYLE DS_MODALFRAME | WS_MINIMIZEBOX | WS_MAXIMIZEBOX |
      WS_CAPTION    | WS_SYSMENU

CAPTION "Changing Range"

FONT 10, "Arial"

BEGIN
LTEXT "Cuting the begin and the end of the source file data.\n
 If <0> is entered for the end the true end will be taken.", -1, 10,10,150,40
    LTEXT         "New begin (>0):",  -1, 10, 60, 50, 10
    LTEXT         "New end:",  -1, 10, 80, 50, 10
    EDITTEXT      IDD_ANFANG, 80, 60, 70, 10, ES_AUTOHSCROLL
    EDITTEXT      IDD_ENDE, 80, 80, 70, 10, ES_AUTOHSCROLL
    DEFPUSHBUTTON "OK", ID_INORDNUNG,  10,100, 50,15
    PUSHBUTTON    "Cancel", ID_UEBERGEHEN, 80,100, 50,15
END

//-----------------------------------------
//
//Dialogbox for the functiongenerators header
//
//-----------------------------------------
DataAquisition DIALOG DISCARDABLE 50, 50, 160, 140

STYLE DS_MODALFRAME | WS_MINIMIZEBOX | WS_MAXIMIZEBOX |
      WS_CAPTION    | WS_SYSMENU

CAPTION "Data aquisition Parameters"

FONT 10, "Arial"

```

```
BEGIN
    LTEXT          "Scanrate[pts/s]:",       -1, 10, 20, 70, 10
    LTEXT          "Scanfrequency[1/s]:",    -1, 10, 40, 70, 10
    LTEXT          "Amount of Points",       -1, 10, 60, 70, 10
    EDITTEXT       IDD_SCANRATE,                 90, 20, 60, 10, ES_AUTOHSCROLL
    EDITTEXT       IDD_SCANFREQUENCY,            90, 40, 60, 10, ES_AUTOHSCROLL
    EDITTEXT       IDD_POINTS,                   90, 60, 60, 10, ES_AUTOHSCROLL
    DEFPUSHBUTTON "OK", ID_INORDNUNG,  10,100, 50,15
    PUSHBUTTON     "Cancel", ID_UEBERGEHEN, 80,100, 50,15
END

//
//Dialogbox for special function Properties 3
//
//-------------------------------------------
Kubic DIALOG DISCARDABLE 50, 50, 160, 140

STYLE DS_MODALFRAME | WS_MINIMIZEBOX | WS_MAXIMIZEBOX |
      WS_CAPTION    | WS_SYSMENU

CAPTION "Function Parameter List"

FONT 10, "Arial"

BEGIN
    LTEXT          "1.st Parameter",     -1, 10, 20, 60, 10
    LTEXT          "2.nd Parameter",     -1, 10, 40, 60, 10
    LTEXT          "3.rd Parameter",     -1, 10, 60, 60, 10
    EDITTEXT       IDD_A,                80, 20, 70, 10, ES_AUTOHSCROLL
    EDITTEXT       IDD_B,            80, 40, 70, 10, ES_AUTOHSCROLL
    EDITTEXT       IDD_C,                    80, 60, 70, 10, ES_AUTOHSCROLL
    DEFPUSHBUTTON "OK", ID_INORDNUNG,  10,100, 50,15
    PUSHBUTTON     "Cancel", ID_UEBERGEHEN, 80,100, 50,15
END

/*********************************************
*Dialog for two parameter functions
*********************************************/
Quadratic DIALOG DISCARDABLE 50, 50, 160, 140

STYLE DS_MODALFRAME | WS_MINIMIZEBOX | WS_MAXIMIZEBOX |
      WS_CAPTION    | WS_SYSMENU

CAPTION "Function Parameter List"

FONT 10, "Arial"

BEGIN
    LTEXT          "1.st Parameter",     -1, 10, 20, 60, 10
    LTEXT          "2.nd Parameter",     -1, 10, 40, 60, 10
    EDITTEXT       IDD_A,                80, 20, 70, 10, ES_AUTOHSCROLL
    EDITTEXT       IDD_B,            80, 40, 70, 10, ES_AUTOHSCROLL
    DEFPUSHBUTTON "OK", ID_INORDNUNG,  10,100, 50,15
    PUSHBUTTON     "Cancel", ID_UEBERGEHEN, 80,100, 50,15
END


/*********************************************
*Dialog for one parameter functions
*********************************************/
Linear DIALOG DISCARDABLE 50, 50, 160, 140

```

```
STYLE DS_MODALFRAME | WS_MINIMIZEBOX | WS_MAXIMIZEBOX |
      WS_CAPTION    | WS_SYSMENU

CAPTION "Function Parameter List"

FONT 10, "Arial"

BEGIN
    LTEXT          "1.st Parameter",     -1, 10, 20, 60, 10
    EDITTEXT       IDD_A,                80, 20, 70, 10, ES_AUTOHSCROLL
    DEFPUSHBUTTON "OK", ID_INORDNUNG,  10,100, 50,15
    PUSHBUTTON     "Cancel", ID_UEBERGEHEN, 80,100, 50,15
END

//----------------------------------------
//
//Dialogbox for entering the matrix and the vectors.
//
//----------------------------------------
Matrixeingabe DIALOG DISCARDABLE 50, 50, 160, 140

STYLE DS_MODALFRAME | WS_MINIMIZEBOX | WS_MAXIMIZEBOX |
      WS_CAPTION    | WS_SYSMENU

CAPTION "Matrix and Vector"

FONT 10, "Arial"

BEGIN
LTEXT "This will calculate the solution vector x of the system Ax=b.\n
 Please enter the Matrix- and vectorname. ", -1, 10,10,150,40
    LTEXT          "Matrix name:",  -1, 10, 60, 50, 10
    LTEXT          "Vector name:",  -1, 10, 80, 50, 10
    EDITTEXT       IDD_MATRIX, 80, 60, 70, 10, ES_AUTOHSCROLL
    EDITTEXT       IDD_VEKTOR, 80, 80, 70, 10, ES_AUTOHSCROLL
    DEFPUSHBUTTON "OK", ID_INORDNUNG,  10,100, 50,15
    PUSHBUTTON     "Cancel", ID_UEBERGEHEN, 80,100, 50,15
END
```

Listing B.5: *ACF.rc*

Literaturverzeichnis

[Bag57] E.B. Bagley. End corrections in the capillary flow of polyethylene. *J. Appl. Phys.*, 28(5):624–627, 1957.

[BB60] E.B. Bagley and A.M. Birks. Flow of polyethylene into a capillary. *J. Appl. Phys.*, 31(3):556–561, 1960.

[Böh98] J.F. Böhme. *Stochastische Signale*. Teubner, Stuttgart, 1998.

[Böh00] G. Böhme. *Strömungsmechanik nichtnewtonscher Fluide*. B.G.Teubner, Stuttgart,Leipzig,Wiesbaden, 2000.

[Bra65] R.N. Bracewell. *The Fourier Transform and its Applications*. McGraw-Hill, New York, 1965.

[BS01] I.N. Bronstein and K.A. Semendjajew. *Taschenbuch der Mathematik*. Verlag Harri Deutsch, 2001.

[But98] T. Butz. *Fouriertransormierte für Fußgänger*. B.G. Teubner, Stuttgart,Leipzig, 1998.

[CKC97] P.J. Carreau, D. De Kee, and R.P. Chabra. *Rheology of Polymeric Systems: Principle and Application*. Hanser Verlag, München, 1997.

[CT65] J.W. Cooley and J.W. Tuckey. An algorithm for machine calculation of complex fourier series. *Math. Comp.*, 19:297–301, 1965.

[Den01] M.M. Denn. Extrusion instabilities and wall slip. *Annu. Rev. Mech.*, 33:265–287, 2001.

[DK87] M.M. Denn D.S. Kalika. Wallslip and extrudate distorsion in linear-low-density pe. *J. Rheol.*, 31(8):815–, 1987.

[Ehr99] G.W. Ehrenstein. *Polymer-Werkstoff*. Hanser, München, Wien, 1999.

[Erl05] H. Erlenkötter. *C Programmieren von Anfang an*. Rowolth, 2005.

[GWE04] R.C. Gonzales, R.E. Woods, and S.L. Eddins. *Digital Image Processing using MATLAB*. Prentice Hall, 2004.

[HH05] D.J. Higham and N.J. Higham. *MATLAB Guide*. SIAM, 2005.

[Hän97] E. Hänsler. *Statistische Signale*. Springer, Berlin, 1997.

[Hoh94] J. Hohnerkamp. *Stochastic Dynamical Systems: Concepts, Numerical Methods, Data Analysis*. VCH Wiley, New York, 1994.

[HvDM+04] L. Hilliou, D. van Dusschoten, M.Wilhelm, H. Burkin, and E.R. Rodger. Increasing the force torque transducer sensitivity of a rpa 2000 by a factor of 5:10 via advanced data acquisition. *Rubber chem. and techn.*, 77:192–200, 2004.

[Jam02] J.F. James. *A Student's Guide to Fourier Transform*. Cambridge University Press, 2002.

[KK06] K.K. Kammeyer and K. Kroschek. *Digitale Signalverarbeitung*. B.G. Teubner, Wiesbaden, 2006.

[Lar99] R.G. Larson. *The Structure and Rheology of Complex Fluids*. Oxford, New York, 1999.

[LGN03] M. D. Lechner, K. Gehrke, and E. H. Nordmeier. *Makromolekulare Chemie*. 2003.

[Mac94] C. Macosko. *Rheology, Principles, Measurements and Applications*. VCH Wiley, New York, 1994.

[Mün79] H. Münstedt. Viskositätsdaten von kunststoffschmelzen. *Kunststoffe*, 68(2):92–98, 1979.

[MSW00] H. Münstedt, M. Schmidt, and E. Wassner. Stick and slip phenomena during extrusion of polyethylene melts as investigated by lase-doppler velocimetry. *J. Rheol.*, 44:413–427, 2000.

[Ost25] W. Ostwald. Ueber die geschwindigkeitsfunktion der viskosität disperser systeme. i. *Kolloid Zeitschrift*, 36:99–117, 1925.

[Rab27] B. Rabinowitsch. Über die viskosität und elastizität von solen. *Z. physikal. Chemie*, 145(1):1–26, 1927.

[Ram85] R.W. Ramirez. *The FFT Fundamentals and Concepts*. Prentice-Hall, Engelwood Cliffs, 1985.

[Rie05] F. Rieg. *Grafikprogrammierung für Windows*. Fachbuchverlag Leipzig, 2005.

[Rob01] L. Robert. *Instabilite oscillante de polyethylenes lineaires: observations et interpretations*. Nizza, 2001. Doktorarbeit.

[Sch00] G. Schramm. *Einführung in die Rheologie und Rheometrie*. Gebrüder Haake GmbH, Karlsruhe, 2000. 2.Auflage.

[Spu04] J.H. Spurk. *Strömungslehre*. Springer, Berlin, Heidelberg, 2004.

[Sto05] J. Stoer. *Numerische Mathematik 1*. Springer, Berlin, Heidelberg, New York, 2005.

[SWM99] M. Schmidt, E. Wassner, and H. Münstedt. Setup and test of a laser doppler velocimeter for investigation of flow behaviour of polymer melts. *Mech. Time-Dependent Mat.*, 3:371–393, 1999.

[Uhl79] E. Uhland. Das anomale fließverhalten von polyäthylen hoher dichte. *Rheol. Acta*, 18(1):1–23, 1979.

[vDW01] D. van Dusschoten and M. Wilhelm. Increased torque transducer sensitivity via oversampling. *Rheol. Acta*, 40(4):395–399, 2001.

[vFLSW00] K. von Finkenstein, J. Lehn, H. Schellhaas, and H. Wegmann. *Arbeitsbuch der Mathematik für Ingenieure I*. B.G. Teubner, Stuttgart, Leipzig, Wiesbaden, 2000.

[Wan99] Shi-Qi Wang. Molecular transition and dynamics at polymer/wall interfaces: Origins of flow instabilities and wall slip. *Adv. Polymer Scien.*, 138:228–275, 1999.

[WDI96] S.Q. Wang, P.A. Drda, and Y.W. Inn. Exploring the molecular origin of sharkskin, partial slip and slope change in flow curves of linear low density pe i. *J. Rheol.*, 40(8):875–898, 1996.

[Wil02] M. Wilhelm. Fourier transform rheology. *Macromol. Mater. Eng.*, 287:83–105, 2002.

[WRO99] M. Wilhelm, P. Reinheimer, and M. Ortseifer. High sensitivity fourier-transform rheology. *Rheol. Acta*, 38:349–356, 1999.

Index

Danksagung

Ich möchte mich bei den folgenden Personen bedanken:

- Prof. M. Wilhelm für die Aufnahme in seine Gruppe als Diplomstudent und seine Betreuung innerhalb der letzten Monate,
- Prof. H.W. Spiess für die Aufnahme in seine Gruppe und einen Platz zum arbeiten,
- Prof. Wegner für die offizielle Stelle als Diplomstudent am MPIP,
- der Gruppe für Polymeranalytik für die Hilfe bei der Charakterisierung meiner Proben,
- G. Glasser für seine Hilfe bei der Untersuchung der Extrudatproben am Mikroskop,
- meinen Kollegen aus dem Büro für ihre freundliche Aufnahme und Hilfe,
- den Mitarbeitern der Mechanik-Werkstatt für die gut gefertigte Schlitzdüse,
- der Firma Goettfert für die technische Unterstützung und die schnelle Lieferung des benötigten Zubehörs zum Kapillarrheometers,
- der Rheologie Gruppe um Prof. Wilhelm (Dr. V. Barroso, A. Calin, Dr. K. Hyun, K. Riazi) für ihre Unterstützung, Hilfe und Freundschaft,
- den Ehemaligen aus der Rheologie Gruppe Dr. I. Vittorias und Dr. S. Filipe für ihre Einweisung an den Rheometern,
- A. Becker für seine Hilfe bei allen Problemen die sich am Kapillarrheometer ergaben,
- meinen Erst-Korrektoren A. Wachsmuth und Dr. V. Barroso,
- meiner gesamten Familie die mich all die Jahre unterstützt hat und mir geholfen hat diese Arbeit in Ruhe zu vollenden,

- meinen Freunden, die mir die Vernachlässigung meinerseits während der letzten Monate verzeihen,
- und zu guter letzt meiner Freundin, die liebevoll zu mir gestanden ist und mir die Kraft gegeben und die Zeit gelassen hat diese Arbeit zu beenden. Auf das wir für immer zusammen bleiben.