Technische Universität Dresden
Fakultät Elektrotechnik und Informationstechnik
Institut für Akustik und Sprachkommunikation
Professur für Sprachtechnologie und Kognitive Systeme
Prof. Dr.-Ing. P. Birkholz
Mai 2024

Praktikum zur Lehrveranstaltung

Mikrorechentechnik II

Versuch

Akustischer Schalter

Inhaltsverzeichnis:

1. Zielstellung:

Im Praktikumsversuch sollen prinzipielle Fertigkeiten zur Software-Umsetzung von *Algorithmen zur Signalverarbeitung* erworben bzw. gefestigt werden. Im Mittelpunkt steht dabei die Problematik der Unterteilung von Signalen entsprechend der von ihnen getragenen Information. Die Behandlung der Teilaspekte *Signalaufbereitung, Signaltransformation sowie Analyse der Signale* bezüglich der von ihnen getragenen wesentlichen Information bilden den Schwerpunkt des Versuchs.
Ziel des Versuchs ist die Erstellung von drei Softwaremodulen in der Programmiersprache ANSI-C zur Signalverarbeitung für ein einfaches akustisches Objekterkennungssystem.
Ausgehend von dieser allgemeinen Zielstellung enthält Abschnitt 2. einige grundsätzliche Bemerkungen zu *Aufbau und Wirkungsweise von Objekterkennungssystemen*. Diese Ausführungen sind zum tieferen Verständnis des fachlichen Hintergrundes der im Abschnitt 3. formulierten *Praktikumsaufgabe* gedacht. Zur programmtechnischen Bewältigung der Praktikumsaufgabe ist eine detaillierte Beherrschung dieser Grundlagen jedoch nicht unbedingt erforderlich. Abschnitt 4. dient der näheren Erläuterung der für den Versuch zur Verfügung stehenden *technischen Voraussetzungen* und vermittelt wichtige *praktische Hinweise zur Versuchsdurchführung*. Schließlich werden im Abschnitt 5. Schwerpunkte des zum Versuch gehörenden *Kolloquiums* genannt.

2. Theoretische Grundlagen:

2.1. Signalverarbeitung als Problem der Objekterkennung:

Viele durch signalverarbeitende Systeme zu lösende Aufgaben können als Problem der *Objekterkennung* angesehen werden. Insbesondere im Bereich Messwerterfassung und -auswertung sowie Prozesssteuerung, aber auch im Bereich der Kommunikationstechnik sind häufig nicht alle von den jeweiligen Signalen getragenen Informationen sondern nur bestimmte Teilmengen davon relevant. Um z.B. in einer automatischen Flaschensortieranlage grüne und braune Flaschen trennen zu können, spielen Form, Durchmesser, Größe usw. der Flasche keine Rolle und brauchen somit auch nicht erfasst zu werden. Entscheidend ist ausschließlich die *Farbe* der Flaschen. Allgemein kann man die zu verarbeitenden bzw. zu erfassenden Dinge (hier die Flaschen) als *Objekte* bezeichnen, die durch Signale repräsentiert bzw. beschrieben werden. Die eigentliche Aufgabe besteht dann oft darin, die Objekte vorgegebenen *Kategorien (Klassen)* zuzuordnen (im Beispiel: grünes - braunes Glas). Dieser Vorgang wird als *Klassifikation*, die Ableitung der dafür notwendigen Informationsteilmenge der Signale als *Merkmalextraktion oder Analyse* bezeichnet. Signalverarbeitende Systeme, die auf der Basis von Merkmalen Objekte bestimmten Klassen zuordnen können, werden als *Objekt- oder Mustererkennungssysteme* bezeichnet.

2.2. Merkmalproblem:

Eines der bei Objekterkennungsaufgaben zu lösenden Probleme stellt das *Merkmalproblem* dar. Es lassen sich zunächst keine allgemeingültigen Strategien zur Definition und Bestimmung

optimaler Merkmale angeben, da diese sehr stark von der jeweiligen Aufgabe abhängen. Prinzipiell sollte die Merkmalanzahl aus Aufwandsgründen minimal sein, andererseits jedoch eine möglichst fehlerfreie Klassenzuordnung der damit beschriebenen Objekte gewährleisten. Die Merkmale müssen nicht unbedingt gleichartig sein (z.B. gleiche physikalische Bedeutung).
Häufig erweist sich vor der eigentlichen Merkmalsextraktion eine *Transformation der Objektsignale* in einen geeigneten Bildbereich als günstig.
Alle extrahierten Merkmale werden als Komponenten eines das jeweilige Objekt beschreibenden *Merkmalvektors* $\vec{x}$ aufgefasst. Damit können alle zu klassifizierenden Objekte als Vektoren in einem durch die Merkmale (Vektorkomponenten) aufgespannten Raum dargestellt werden.
Diesen Raum nennt man *Merkmalraum*. Seine Dimension entspricht der Merkmalanzahl.
Im oben angeführten Beispiel der automatischen Flaschensortierung könnte als Merkmal die Wellenlänge des mit einem optischen Sensor von dem Objekt "Flasche" reflektierten bzw. nach Durchleuchtung aufgenommenen Lichts benutzt werden. In diesem Fall wäre der Merkmalraum eindimensional. Wenn man die Farbe z.B. an zwei verschiedenen Stellen der Flasche erfasst (z.B. um mehrfarbige Flaschen auszusondern), erhält man einen zweidimensionalen Merkmalvektor der demzufolge in einem zweidimensionalen Merkmalraum dargestellt werden muss (Bild 1).

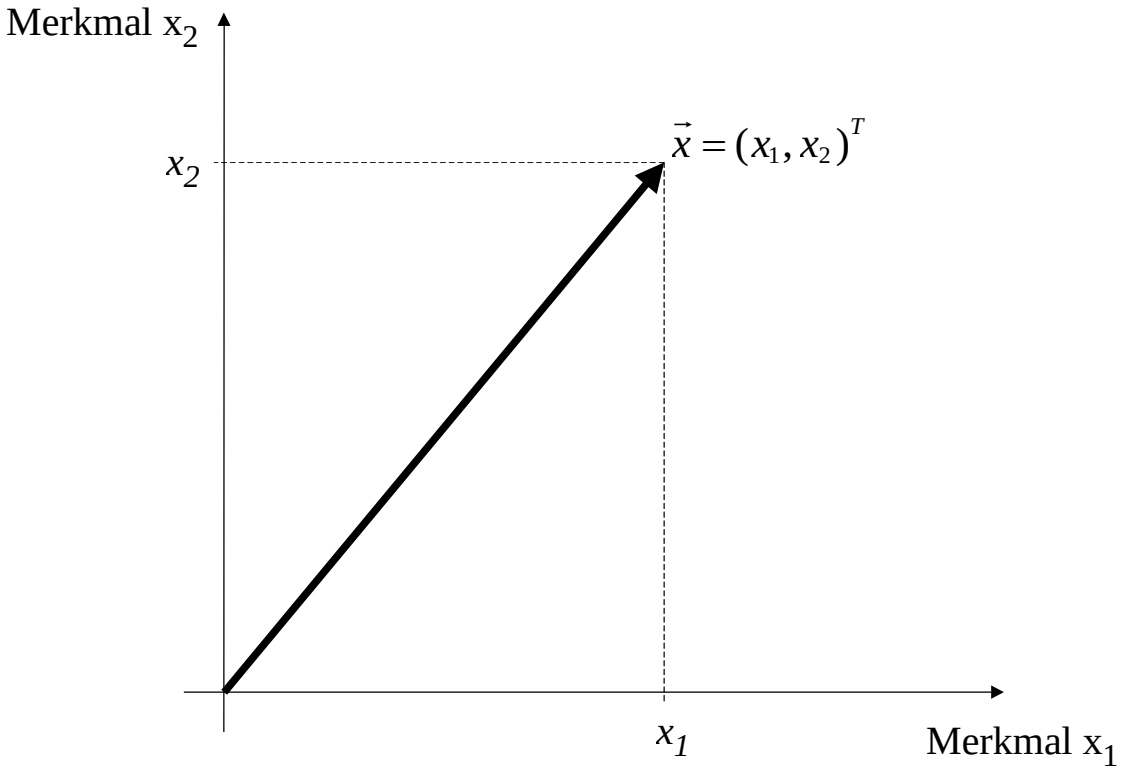


Bild 1
Darstellung eines mit einem zweidimensionalen Merkmalvektor $\vec{x}$ beschriebenen Objekts im Merkmalraum

2.3. Klassifikationsproblem:

Schließlich ist auf der Basis der objektbeschreibenden Merkmale (bzw. des objektbeschreibenden Merkmalvektors) die Zuordnung zu den definierten Kategorien (im obigen Beispiel Flasche grün - Flasche braun) vorzunehmen. Dazu sind Referenzobjekte notwendig, die als "typisches" Objekt für die jeweilige Klasse anzusehen sind (z.B. Flasche in "Normgrün" bzw. "Normbraun"). Ein

Referenzobjekt muss jedoch nicht physisch vorhanden sein, es reicht eine geeignete Beschreibung, ebenfalls durch Merkmale r (bzw. Merkmalvektor $\vec{r}$, r steht für *Referenz*), wobei dafür verschiedene Möglichkeiten existieren. Sollte für eine Kategorie kein "typisches" Objekt direkt angebbar sein, so kann dieses z.B. durch Mittelwertbildung der Merkmalvektoren einer bestimmten Anzahl von Objekten mit sicherer Zugehörigkeit zu dieser Kategorie bestimmt werden ("(An)-Lernen" der Referenzobjekte). Die Referenzobjekte können ebenfalls im Merkmalraum dargestellt werden.
Zur Zuordnung der Objekte zu den Kategorien werden die Abstände des Objektmerkmalvektors $\vec{x}$ zu allen Referenzobjektvektoren $\vec{r}_i$ berechnet. Die Zuordnung erfolgt schließlich zu der Kategorie j, zu deren Referenzobjektvektor $\vec{r}_j$ der minimale Abstand gemessen wurde. Anschaulich können die Abstandsberechnungen als "Ähnlichkeitsmessungen" interpretiert werden. Minimaler Abstand bedeutet dann "maximale Ähnlichkeit" (Bild 2).

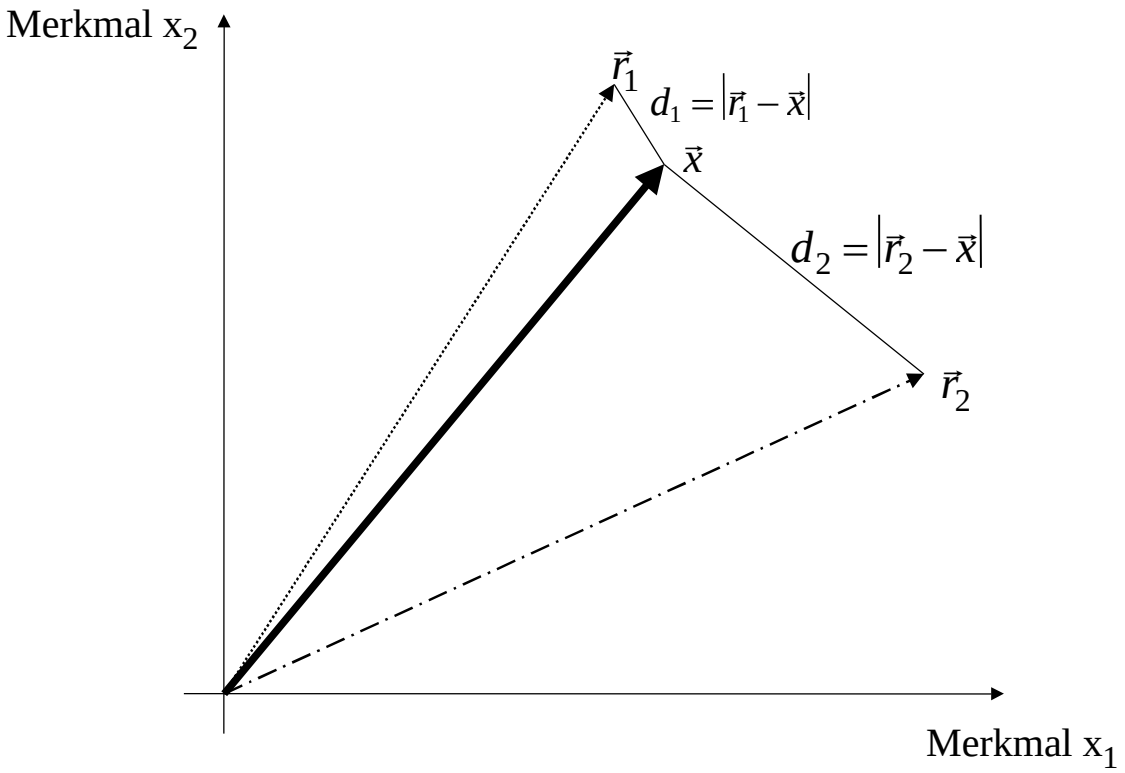


Bild 2
Zuordnung eines Objekts zu einer Kategorie auf der Basis minimalen Abstands zwischen Objektmerkmalvektor $\vec{x}$ und Referenzobjektvektor $\vec{r}_i$ (Zuordnung zu Kategorie i=1, da Abstand d_1 minimal)

Da die Merkmale von realen Objekten trotz optimaler Merkmalwahl relativ stark streuen können, sind Falschzuordnungen nicht auszuschließen. Als Gütemaß für ein Objekterkennungssystem kann eine "Erkennungsrate" angegeben werden (Verhältnis zwischen Anzahl falsch zugeordneter Objekte zur Gesamtzahl zugeordneter Objekte).

3. Praktikumsaufgabe:

3.1. Allgemeine Aufgabenstellung:

In einem sprachakustischen Objekterkennungssystem sollen die Lautkategorien “Laut a” und “Laut i” unterschieden werden (“Lauterkenner”). Die unbekannten Objekte (Laute) liegen in Form von Schalldruck (p) - Zeit (t) -Signalen vor.
Zunächst ist aus dem Eingangssignal der für den Erkennungsvorgang relevante Signalabschnitt zeitlich zu lokalisieren. Dafür wird der zeitliche Energieverlauf des Eingangssignals herangezogen. Als relevanter Signalabschnitt wird der Bereich gewertet, in dem der zugehörige Energieverlauf einen bestimmten Schwellwert überschreitet (Bild 3). Als Maß für die Energie wird das Quadrat des Effektivwerts verwendet. Der Energieverlauf ergibt sich, indem das Effektivwertquadrat sukzessive über einen Teilabschnitt der Länge i des Eingangssignals gebildet wird.

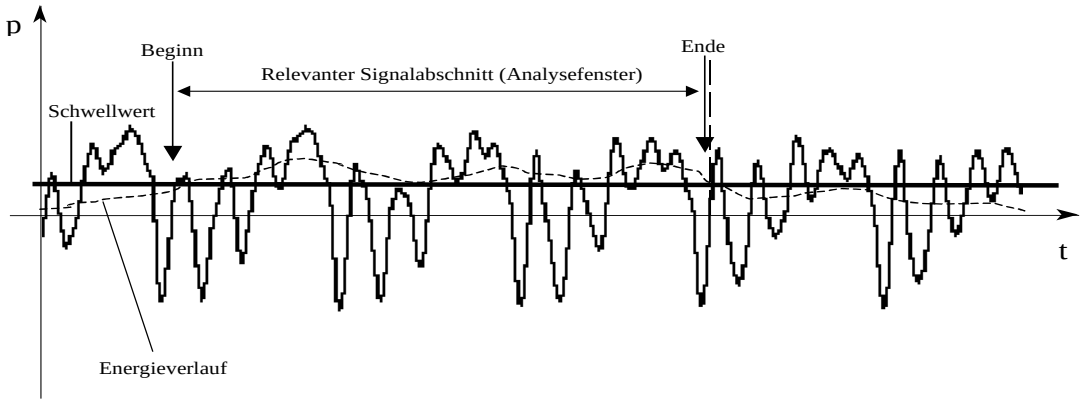


Bild 3
Ermittlung des relevanten Signalabschnitts (Schwellwertüberschreitung des Energieverlaufs)

Nach dieser Signaldetektion sind die Merkmale zu bestimmen. Als einfaches Merkmal für eine Zuordnung kann die Zahl der Nulldurchgänge (NDG) des Signals innerhalb eines bestimmten Analysefensters verwendet werden (Bild 4).

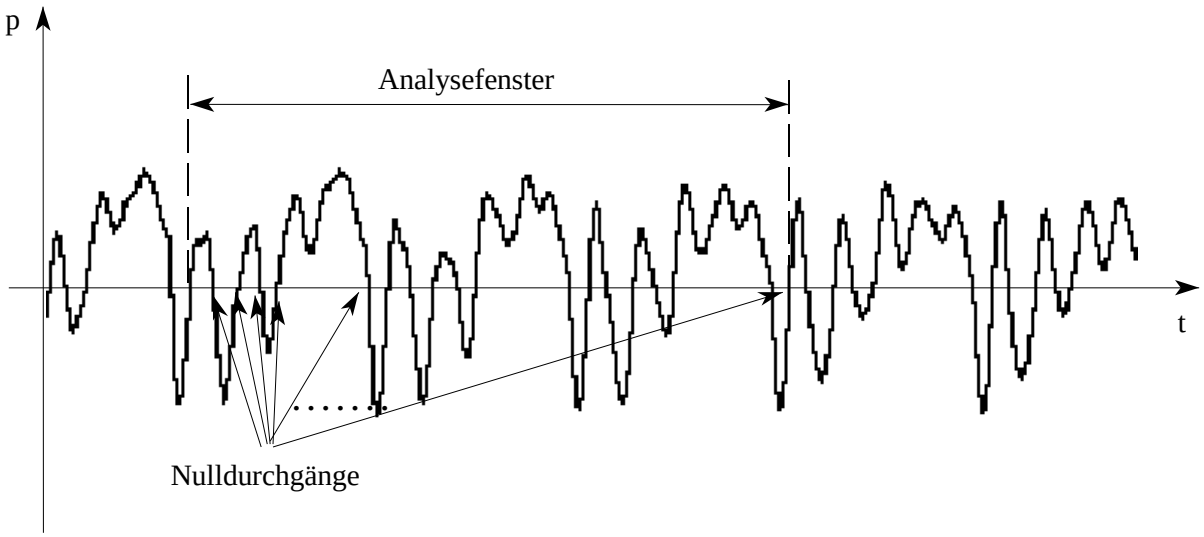
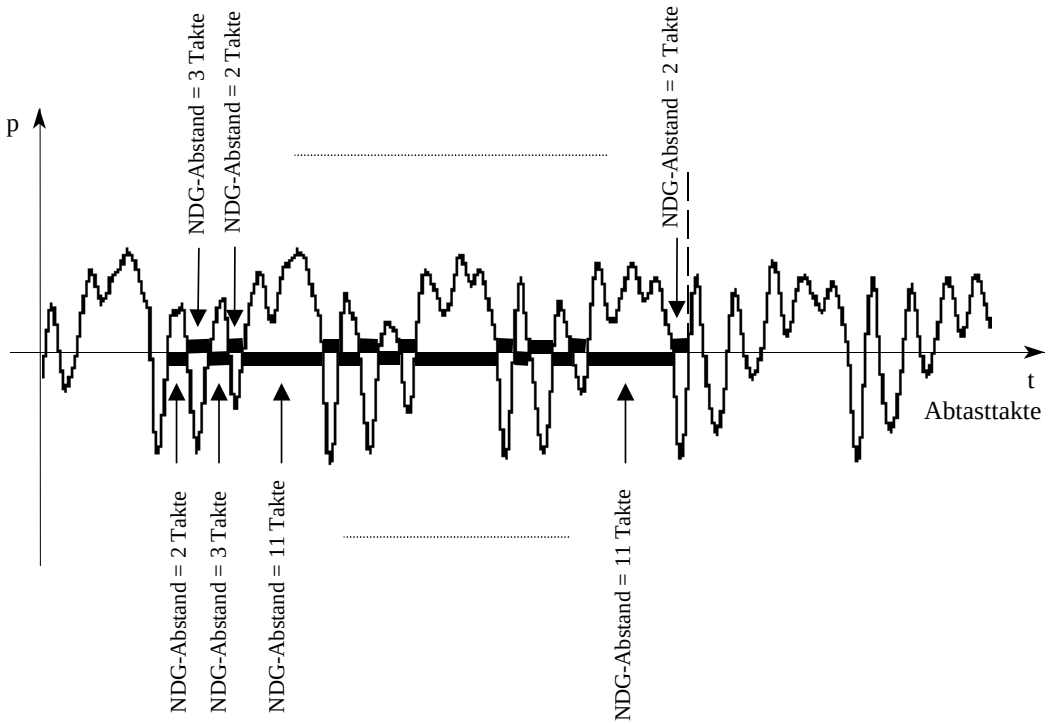
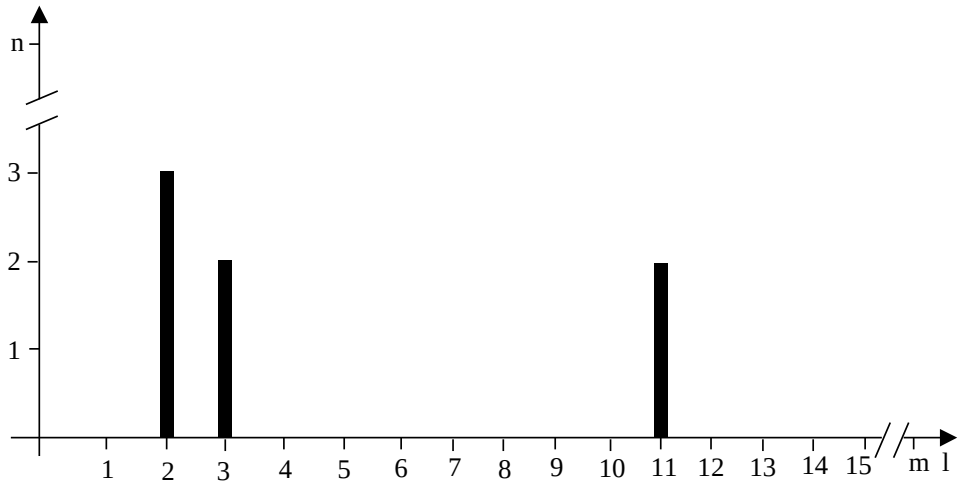


Bild 4
Beschreibung eines Laut-Objekts (Ausschnitt aus der Schalldruck-Zeitfunktion für Vokal a) durch die Anzahl der Nulldurchgänge des Schalldrucks in einem bestimmten Analysefenster

Um die Güte des Objekterkennungssystems zu erhöhen, sollen durch die Bestimmung der Verteilung der Nulldurchgangsabstände im Analysefenster (NDG-(Abstands-) Histogramm) weitere Merkmale gewonnen werden. Dazu sind die Längen aller Nulldurchgangsabstände (Anzahl Abtastwerte zwischen jeweils zwei aufeinanderfolgenden Nulldurchgängen) zu bestimmen und die Auftretenshäufigkeit jedes Nulldurchgangsabstands im Analysefenster zu berechnen (Bild 5).



Häufigkeit NDG-Abstand l = x im Analysefenster



(nur die quantitativ angegebenen NDG-Abstände berücksichtigt)

Bild 5
Beschreibung eines Laut-Objekts (Ausschnitt aus der Schalldruck-Zeitfunktion für Vokal a, oben) durch die Verteilung der Nulldurchgangsabstände des Schalldrucks in einem bestimmten Analysefenster (unten)

Als Merkmale werden hierbei die jeweiligen Häufigkeiten x der NDG-Abstände l verwendet. Um die Anzahl der Merkmale aus Aufwandsgründen möglichst gering zu halten, werden die

Nulldurchgangsabstände l üblicherweise zu Gruppen zusammengefasst. Dabei haben sich für Sprachsignale 8 bis 10 gleich große Gruppen als ausreichend erwiesen.

Eine weitere Verbesserung der Erkennungsleistung ist erreichbar, wenn die oben erläuterten NDG-Merkmale (Dichte, Häufigkeiten) innerhalb des Analysefensters zusätzlich vom *differenzierten* Signal ermittelt werden.

Um den Einfluss von Störungen im Sprachsignal zu reduzieren, muss die Zeitfunktion nach Durchlaufen des Nullpunkts vor dem nächsten Durchlaufen des Nullpunkts betragsmäßig einen bestimmten Mindestwert *s* ("**S**chwelle") überschreiten, um als "Nulldurchgang" gezählt zu werden (Bild 6).

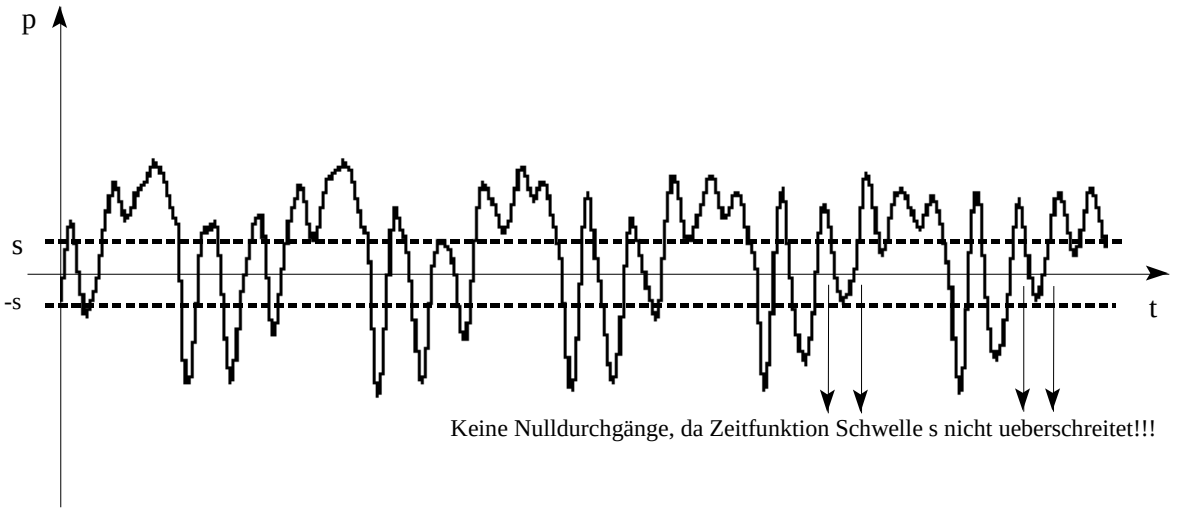


Bild 6
Beispiel für nicht zu zählende Nulldurchgänge

3.2. Detaillierte Aufgabenstellung:

Im Praktikum sind die Module *Signaldetektion*, *Nulldurchgangsdichte* und *Nulldurchgangshistogramm* als jeweils eine Funktion in ANSI-C zu erstellen, mit denen zunächst der relevante Signalabschnitt (Analysefenster) aus dem Gesamtsignal selektiert und vom Gleichanteil (arithmetischer Mittelwert) dieses Signalabschnitts befreit werden kann. Weiterhin sollen die Nulldurchgangsdichte und das NDG-Histogramm der Gleichanteil-befreiten originalen und der differenzierten, in abgetasteter Form vorliegenden, Laut-Zeitfunktion innerhalb dieses Analysefensters berechnet werden können.

***Funktionsaufruf Signaldetektion*:**

int signal_detekt (short **sample_anfang, unsigned int *sample_anzahl)

Parameter:

sample_anfang: Zeiger auf die Adresse des Feldes von Abtastwerten (Signal und Pausen). Die Abtastwerte sind vom Datentyp short. Die Funktion überschreibt die Adresse mit der Anfangsadresse des Signalabschnittes im Gesamtpuffer

sample_anzahl: Zeiger auf die Anzahl zu verarbeitender Abtastwerte. Die Abtastwerteanzahl hat den Datentyp unsigned int. Bei Aufruf der Funktion enthält sample_anzahl die Gesamtanzahl der erfassten Abtastwerte. Die Funktion überschreibt diesen Wert mit der Anzahl von Signalwerten ab sample_anfang

Returnwert:

Die Funktion muss einen Wert vom Datentyp int an das aufrufende Programm zurückgeben:

: **0** kein Fehler aufgetreten
: **-1** es ist ein schwerwiegender Fehler aufgetreten. Das Programm muss sofort beendet werden
: **1** das Programm konnte die vorhandenen Daten nicht verarbeiten (z.B. kein auswertbarer Signalabschnitt im Puffer). Es müssen neue Abtastwerte erfasst werden.

Funktionsbeschreibung:

Die Funktion lokalisiert den Abschnitt im Signalpuffer, der das Nutzsignal (z.B. Abtastwerte eines Lautes „a“) enthält und somit Pausen von der weiteren Verarbeitung ausschließt.
Zuerst ist der Energieverlauf im gesamten Abtastwertepuffer zu berechnen. Der Abschnitt, in dem die Energie einen vorzugebenden Schwellwert überschreitet und der eine vorzugebende Mindestdauer aufweist, ist als Nutzsignalabschnitt anzusehen. Zur Unterdrückung von Lautstärkeunterschieden beim Sprechen soll der Energieschwellwert relativ zum Energie-maximum im Puffer festgelegt werden.
Die Anfangsadresse und die Abtastwertezahl beschreiben den Bereich mit Nutzsignal.
Innerhalb dieses Nutzsignalbereichs ist das Signal vom Gleichanteil zu befreien.

Die Funktion gibt einen Integerwert zurück, der Auskunft über das Ergebnis der Berechnungen gibt.

***Funktionsaufruf Nulldurchgangsdichte*:**

int ndg_dichte(short *feld_ptr,unsigned int anzahl_atw, float *dichte_ori, float *dichte_diff)

Parameter:

feld_ptr: Zeiger auf ein Feld von Abtastwerten des Analysefensters einer Laut-Zeitfunktion. Die Abtastwerte sind vom Datentyp short.

anzahl_atw: Anzahl zu verarbeitender Abtastwerte. Die Abtastwerteanzahl hat den Datentyp unsigned int.

dichte_ori: Zeiger auf eine Speicherzelle, in die die Nulldurchgangsdichte der originalen Zeitfunktion geschrieben werden muss.

dichte_diff: Zeiger auf eine Speicherzelle, in die die Nulldurchgangsdichte der differenzierten Zeitfunktion geschrieben werden muss.

Returnwert:

Die Funktion muss einen Wert vom Datentyp int an das aufrufende Programm zurückgeben:

: **0** kein Fehler aufgetreten
: **-1** es ist ein schwerwiegender Fehler aufgetreten. Das Programm muss sofort beendet werden
: **1** das Programm konnte die vorhandenen Daten nicht verarbeiten. Es müssen neue Abtastwerte erfasst werden.

Funktionsbeschreibung:

Die Funktion berechnet die Nulldurchgangsdichte **dichte_ori** einer vorgegebenen originalen Zeitfunktion durch Bestimmung der Anzahl der Nulldurchgänge im vorgegebenen Abtastwertefeld im Verhältnis zur Anzahl der Abtastwerte im gesamten Analysefenster. Anschließend wird die Zeitfunktion durch Subtraktion aufeinanderfolgender Abtastwerte differenziert. Analog zur Größe **dichte_ori** wird danach die Nulldurchgangsdichte des differenzierten Signals **dichte_diff** bestimmt (siehe Abschnitt 3.1). Das Feld der Abtastwerte darf nicht überschrieben werden.
Die Funktion gibt einen Integerwert zurück, der Auskunft über das Ergebnis der Berechnungen gibt.

In den im Rahmenprogramm (vgl. Abschnitt 4.1.) nachfolgenden Modulen wird der Inhalt des Feldes mittels Soundausgabe wiedergegeben und optional im wav-Format auf Datenträger abgelegt.

Beispiel für den Zugriff auf die Funktionsparameter:

Eine Möglichkeit des Zugriffs auf die Funktionsparameter soll anhand der nachfolgenden Programmzeilen kurz demonstriert werden:

```
int ndg_dichte(short *sample_anfang,unsigned int sample_anzahl, float *dichte_ori, float *dichte_diff)

        {
        unsigned int i;
        short min,max;
        float a;

 /* --Maximum und Minimum berechnen---------- */
        for(i=0,min=32767,max=-32768;i<sample_anzahl;i++)
                { if( *(sample_anfang+i) > max) max = *(sample_anfang+i);
                if( *(sample_anfang+i) < min) min = *(sample_anfang+i);
                }
.
.
```

```
*dichte_ori = a;
*dichte_diff = a/3.;
return (0);
}
```

***Funktionsaufruf Nulldurchgangshistogramm*:**

int ndg_histogramm(short *feld_ptr, unsigned int anzahl_atw, float *hist_ori, float *hist_diff)

Parameter:

feld_ptr: Zeiger auf ein Feld von Abtastwerten des Analysefensters einer Laut-Zeitfunktion. Die Abtastwerte sind vom Datentyp short.

anzahl_atw: Anzahl zu verarbeitender Abtastwerte. Die Abtastwerteanzahl hat den Datentyp unsigned int.

hist_ori: Zeiger auf ein Feld von float-Werten, in das die Nulldurchgangshistogrammwerte der originalen Zeitfunktion geschrieben werden müssen. Die Anzahl der Histogrammwerte ist in der Headerdatei „erkenner.h“ in der Konstante ANZ_HIST_KAN mit 4 definiert.

hist_diff: Zeiger auf ein Feld von float-Werten, in das die Nulldurchgangshistogrammwerte der differenzierten Zeitfunktion geschrieben werden müssen. Die Anzahl der Histogrammwerte ist in der Headerdatei „erkenner.h“ in der Konstante ANZ_HIST_KAN mit 4 definiert.

Returnwert:

Die Funktion muss einen Wert vom Datentyp int an das aufrufende Programm zurückgeben:

- : **0** kein Fehler aufgetreten
- : **-1** es ist ein schwerwiegender Fehler aufgetreten. Das Programm muss sofort beendet werden
- : **1** das Programm konnte die vorhandenen Daten nicht verarbeiten. Es müssen neue Abtastwerte erfasst werden.

Funktionsbeschreibung:

Der Algorithmus zur Berechnung des Nulldurchgangshistogramms ist eine Erweiterung des Algorithmus zur Berechnung der Nulldurchgangsdichte (vgl. Abschnitt 3.1 und Funktion *Nulldurchgangsdichte*). Im Unterschied zur Dichte werden die Nulldurchgänge jedoch nicht einfach gezählt, sondern entsprechend ihrer Länge Histogrammkanälen zugeordnet und aufsummiert. Die Anzahl der Kanäle ist je Histogramm mit 4 (s.o.) vorgegeben. Die Grenzen zwischen den Kanälen sollen mittels Konstanten festgelegt werden. Da kurze Nulldurchgangsabstände bei vorgegebener Signallänge häufiger auftreten können als längere, sind kanalabhänige Inkrementwerte ebenfalls über Konstanten zu definieren. (Die Optimierung der Parameter ist nicht Ziel des Praktikums!).

Das Histogramm ist vom Originalsignal im Feld „hist_ori“ und vom differenzierten Signal im Feld „hist_diff“ abzulegen.

4. Versuchsaufbau:

4.1. Rahmenprogramm:

Das Programm “erkenner” bildet unter Einbeziehung der im Praktikum zu erstellenden Module *Signaldetektion, Nulldurchgangsdichte und Nulldurchgangshistogramm* ein komplettes Objekterkennungssystem für einfache akustische Objekte.

Im Programm werden die Funktionen

- Signalerfassung,
- ***Signaldetektion***,
- ***Signalanalyse mit Berechnung der Nulldurchgangsdichte*** und ***Berechnung des Nulldurchgangshistogrammes,***
- Klassifikation (Zuordnung zu Referenzobjekten),
- Signalspeicherung und
- Datenverwaltung für Referenzobjekte

implementiert. Alle Module außer *Signaldetektion, Nulldurchgangsdichte und Nulldurchgangshistogramm* sind bereits erstellt und liegen in compilierter Form auf dem Praktikumsrechner vor.

Im Folgenden wird die Funktionalität und das Zusammenwirken der einzelnen Module kurz vorgestellt:

Im Modul *Signalerfassung* wird eine Zeitfunktion vorgegebener Länge (32000 Abtastwerte) über eine Soundschnittstelle des Betriebssystems in den Hauptspeicher des Rechners eingelesen. Die Zeitfunktion am Eingang des Analog-Digital-Umsetzers wird mit 16000Hz einkanalig abgetastet und als Folge von 16-Bit-Werten im Zweierkomplement (Datentyp “short”) in einem Feld abgelegt.
Der Modul ***Signaldetektion*** sucht in diesem Feld einen Bereich von Abtastwerten (*Analysefenster*), der eine bestimmte Energie über einen vorgegebenen Zeitabschnitt aufweist. Dadurch soll sichergestellt werden, dass in den nachfolgenden Schritten ein “Nutz”-Signal im Analysefenster verarbeitet wird und somit Pausen abgeschnitten sind. Der selektierte Zeitfunktionsabschnitt “signal” wird durch seine Anfangsadresse und seine Länge beschrieben.
Nachfolgend berechnen die Module ***Nulldurchgangsdichte*** *und* ***Nulldurchgangshistogramm*** aus den Abtastwerten die zur Klassifikation erforderlichen *Merkmale* NDG-Dichte und NDG-Histogramm vom Originalsignal und vom differenzierten Signal (siehe Abschnitt 3.2.).
Optional kann danach die im Feld “signal” vorhandene Zeitfunktion als wav-Datei abgespeichert werden.
In der Betriebsart “*Lernen”* werden die Referenzobjektmerkmale durch Verknüpfung mit den neu berechneten Merkmalen aktualisiert. In der Betriebsart “*Erkennen”* werden die berechneten Merkmale im Modul *Klassifikation* mit den Referenzobjektmerkmalen verglichen. Die Kategorie (*Klasse “0” für Laut a oder Klasse “1” für Laut i*), zu deren Merkmalvektor $\vec{r}$ der Abstand des

Merkmalvektors $\vec{x}$ am geringsten ist, wird als *Erkennungsergebnis* ausgegeben.
Optional können die Referenzdaten auf dem Bildschirm protokolliert, in eine Datei gesichert und aus einer Datei eingelesen werden.
Das Programm ist als *Konsolenapplikation* ausgelegt und wird somit in einer Windows-Konsole („Eingabeaufforderung“) gestartet. Alle Ein- und Ausgaben werden über entsprechende E/A-Funktionen (printf, gets, scanf, ...) gestaltet.

4.2. Programmierumgebung:

- Die zu erstellenden Unterprogramme sollen als *Funktionen* für das im Abschnitt 4.1. erläuterte Rahmenprogramm *(32 bit-Applikation)* erstellt werden.

- Das Programm *erkenner* ist als *Konsolenapplikation* (ohne Grafikoberfläche o.ä.) ausgelegt.

- Für die Programmentwicklung kann die Integrierte Entwicklungsumgebung *eclipse* genutzt werden. Als Compilerplattform wird MinGW verwendet.

5. Kolloquiumsschwerpunkte:

- Aufbau eines Objekterkennungssystems
- Arten der Zahlendarstellung
- Arbeiten mit Datenfeldern
- Anwendung von Arithmetikoperationen
- Parameterübergabe zwischen Funktionen und Hauptprogramm

6. Literaturhinweise:

Hoffmann, R.: Signalanalyse und -erkennung. Abschnitte 2.3.3. und 7.1.
Springer-Verlag Berlin Heidelberg New York. 1. Auflage, 1998.