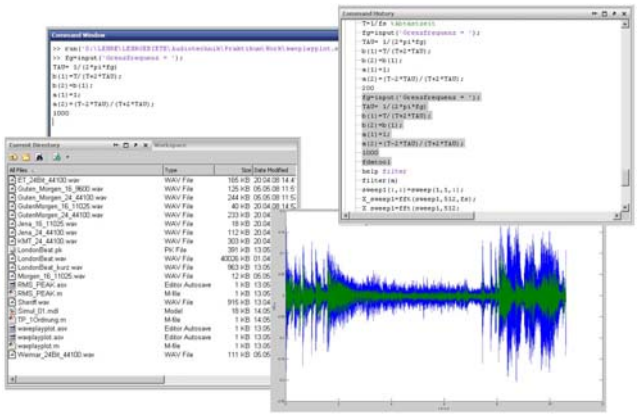
Fachhochschule Jena University of Applied Sciences Jena	**Fachbereich Elektrotechnik und Informationstechnik** Laborpraktikum Digitale Audiotechnik Prof. Dr. H. Kahnt
Übung 3 **Audiosignalverarbeitung mit MATLAB** **Bearbeiten von WAV-Dateien**	
Programme: MATLAB 	

Arbeitsverzeichnis: Laufwerk: E:\ JahrMonat_Name z.B. E:\0805_Meier
(Projekte grundsätzlich nicht! auf Laufwerk C speichern)

Ziel des Versuches

- Einlesen, Analysieren, Darstellen und Wiedergeben von WAV-Dateien
- Verändern von Pegel und Balance
- Ermittlung von Effektivwert und Maximalwert

Aufgabe 1: Vorbereitende Arbeiten

- Erstellen Sie in Ihrem Arbeitsverzeichnis einen Ordner „Work“ als Arbeitsverzeichnis für MATLAB
- Kopieren Sie in diesen Ordner folgende Files aus „Audiobeispiele“

Sheriff.wav
Guten_Morgen_24_44100.wav
Guten_Morgen_16_9600.wav
Jena_24_44100.wav

- Kopieren Sie in diesen Ordner ebenfalls die von Ihnen erstellten Audiodateien aus dem Praktikumsversuch 2 (Protools)

- Öffnen Sie MATLAB und wählen Sie als Work Direktory den Ordner „Work“ aus
- Löschen Sie alle Fensterinhalte

Aufgabe 2: Einlesen einer WAV-Datei und ihrer Basisinformationen

MATLAB-Funktion zum Einlesen:	wavread
>>y= wavread('filename')	Übergibt die Samplewerte an ein Array y
>>[y, fs, nbits,] = wavread('filename')	Liefert die Samplewerte, die Samplingfrequenz und die Auflösung
>>siz=wavread('filename','size')	Liefert Anzahl der Samples und Anzahl der Kanäle (Mono =1, stereo = 2)
>> […] = wavread('filename', n)	Liest die ersten n Werte der WAV-Datei

>> […] = wavread('filename', [n1, n2]) Liest die Werte von n1 bis n2 der WAV-Datei

1. Lesen Sie die Audiodatei 'Jena_24_44100.wav' in ein Datenarray y ein

→ Command Window
>>y = wavread('Jena_24_44100');

Im Workspace von MATLAB erscheint ein Array y mit den Samplewerten der Audiodatei mit Angabe der Anzahl der Samples und der Extremwerte.

2. Visualisieren Sie die Samplewerte

→ Doppelklick re. auf y: Zeigt die Daten im Array Editor an
→ Doppelklick re. auf plot(y) liefert die Darstellung der Samplewerte über dem Laufindex

3. Lesen Sie zusätzliche Informationen aus der WAV-Datei aus

→ Command Window
>> [y, fs, nbits,] = wavread('Jena_24_44100');

>>fs zeigt die Samplingfrequenz
>>nbits zeigt die Auflösung in Bit

Aufgabe 2: Wiedergabe einer Audiodatei

MATLAB-Funktionen zur Wiedergabe: soundsc, wavplay

soundsc skaliert die Audiodaten auf Werte zwischen -1….+1 und gibt sie über die Soundkarte aus
wavplay liefert die Daten unskaliert an die Soundkarte

Optionen

>>soundsc(y)	gibt Datei skaliert mit Samplefrequenz 8192 Hz wieder
>>soundsc(y,fs)	gibt Datei skaliert mit Samplefrequenz fs in Hz wieder
>>soundsc(y,fs,bits)	gibt Datei skaliert mit Samplefrequenz fs in Hz und der Auflösung in Bits wieder
>>soundsc(y,...,slim)	gibt Datei skaliert slim= [slow shigh] für slow =oberer und shigh = unterer Grenzwert wieder, voreingestellt ist slim=[min(y) max(y)]

1. Wiedergabe mit soundsc

→ Command Window
>> soundsc(y,44100)

y wird über die Soundkarte mit maximaler Aussteuerung ausgegeben

2. Wiedergabe einer Audiodatei mit waveplay

→ Command Window
>> wavplay(y,44100)

Die Audiodatei y wird in der Originalaussteuerung wiedergegeben

→ Command Window
>> wavplay(0.5*y,44100)

Die Datei wird mit geringerem Pegel wiedergegeben

3. Verändern sie das Tempo (und damit die Tonhöhe) der Wiedergabe durch Erhöhen oder Erniedrigen der Samplingfrequenz um einen Faktor

→ Command Window
>> wavplay(y,0.8*44100)

y wird langsamer und tiefer (pitch down) wiedergegeben

>> wavplay(y,1.2*44100)

y wird schneller und höher (pitch up) wiedergegeben

4. Lesen Sie eine Stereodatei ein und verändern Sie Gesamtlautstärke und Stereobalance

Aufgabe 3: Darstellung einer Audiodatei

MATLAB-Funktionen zur grafischen Darstellung :
Verwenden sie dazu die plot-Funktionen von MATLAB siehe >>help plot

>>plot(y)	plottet Werte im Vektor y gegen den Laufindex von y
>>plot(x,y)	plottet Werte in Vektor y gegen Werte in Vektor x,

1. Lesen Sie die Wav-Datei 'Sheriff.wav' ein

>>[y,fs]=wavread('Sheriff');	liest Datei 'Sheriff.wav' ein

2. Stellen Sie die Datei grafisch dar

>>plot((1:length(y))/fs, y);	Plotted alle Samples im Array y und normiert die Abszisse auf die Samplefrequenz (Transformation zur Zeitachse)
>>xlabel('Zeit in s');	Beschriftung x-Achse
>>ylabel('y(t)');	Beschriftung y-Achse

Aufgabe 4: Erstellen eines Programms zur Darstellung und Wiedergabe einer Audiodatei „wavplayplot.m“ mit Anzeige von Effektiv- und Peakwert

Erstellen sie im Editor-Fenster mit Hilfe der bisher bekannten Funktionen ein m-File, das folgende Schritte ausführt:

- Einlesen einer WAV-Datei incl. Samplerate, Auflösung und Anzahl der Kanäle
- normierte Wiedergabe
- Plot der WAV-Datei gegen Zeitachse in s
- Berechnung des Effektivwertes über die gesamte Datei und Einzeichnen als Linie in das Diagramm
- Ermittlung des Maximalwertes und Einzeichnen als Linie in das Diagramm

Hinweise:

- Der Effektivwert über alle Samples bestimmt sich über $x_{eff} = \sqrt{\frac{1}{N}\sum_{1}^{N} x_i^2}$

- Verwenden Sie keine for-next-Schleifen für die Berechnungen, da MATLAB dafür sehr viel Zeit braucht!!

Aufgabe 5: Ausgabe von Informationen aus Audodateien

Geben sie aus einer Audiodatei folgende Informationen im Command-Window aus:

1. Samplingfrequenz
2. Auflösung
3. Anzahl der Samples
4. Dauer in s
5. Peakwert
6. Effektivwert über gesamte Dauer

Anhang:

Hilfreiche MATLAB-Funktionen

zeros(1,N)	Erzeugt Zeilenverktor mit N Elementen mit dem Wert 0
zeros(N,1)	Erzeugt Spaltenverktor mit N Elementen mit dem Wert 0
ones (M,N)	Erzeugt Zeilenverktor mit M Zeilen und N Spalten mit Elementen mit dem Wert 1
x'	Wandelt Zeilen- in Spaltenvektor und umgekehrt
a.*x	Multipliziert alle Elemente von x mit a
length(x)	gibt die Anzahl der Elemente eine Zeilenvektors x aus
size(x)	liefert die Anzahl der Zeilen und Spalten in einem Array (Nz Ns)
x.^2	Quadiert Vektor x elementweise
sqrt(x)	zieht elementweise die Wurzel aus Vektor x
yl=y(:,1)	liest erste Zeile von y (linker Kanal) aus
yr=y(:,2)	liest zweite Zeile von y (rechter Kanal) aus