

## MATLAB - Signalverarbeitung

---

### Themen der Aufgabenstellung:

- Matlab-Skripte und eigene Funktionen
- Signalein- und -ausgabe
- spezielle Signale (chirp)
- einfache Filteroperationen (Mittelwert, nichtlineare Filter)
- Rechnen mit Vektoren
- Element-für-Element-Rechenoperationen
- Funktionsplots und subplots

### Hinweise:

#### a) Signalnormierung

Um Unter- und Übersteuerung von Signalen bei der Audioausgabe mit *sound()* zu vermeiden, normiert man die Signale zuvor auf den Wertebereich  $[-1, +1]$ .

$$y_{\text{normiert}} = y / \max(\text{abs}(y))$$

#### b) Umformung von Stereosignalen (zweikanalig) zu Monosignalen (einkanalig)

Beim Lesen von wav-Dateien mit *audioread()* kann es sein, dass das gelesene Signale nicht das gewünschte Format hat. Mit dem Befehl *size()* kann das Matrixformat festgestellt werden. Ggf. müssen die Daten noch in das gewünschte Format umgewandelt werden.

Beispiel:

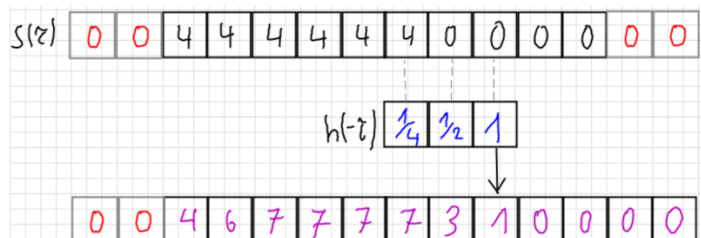
Gewünscht wird das Datenformat  $[n, 1]$ , also ein Spaltenvektor mit  $n$  Elementen.

Gelesen wird aber das Datenformat  $[n, 2]$ , also eine Matrix mit 2 Spalten.

Mit `y = yRead(:, 1)` erhält man das gewünschte Format.

#### c) Realisierung der Faltungsoperation

Zur Realisierung der Faltungsoperation erweitert man den Signalvektor  $s$  am Anfang und Ende um  $k-1$  zusätzliche 0-Werte (= konkat. mit Nullvektor).  
 $k$  = Größe des Faltungskerns



Den (gespiegelten) Faltungskern  $h(-\tau)$  realisiert man am einfachsten mit rückwärts laufendem Index.

**MATLAB - Signalverarbeitung****Aufgabenstellung:****1. Gleitender Mittelwert über k Werte (Input: wav-Datei)**

Es ist ein Matlab-Skript „U1.m“ zu schreiben, welches folgende Schritte ausführt:

- Laden der Audiodatei „JetztGehtsLos.wav“ in den Vektor y,
- Größenbestimmung der Datei,
- Speicher für das gefilterte Signal anlegen,
- Berechnung des gleitenden Mittelwertes über k Werte  $y(n)$ ,  $y(n-1)$ , ...,  $y(n-(k-1))$  (Anm.: Ergebnisse müssen im Wertebereich  $\pm 1$  liegen.),
- Plotten des Originalsignals und des gefilterten Signals in einem subplot,
- Audioausgabe der ungefilterten Datei,
- Weiter mit „pause“ (mit Cursor vor Return in Command-Window gehen)
- Audioausgabe der Mittelwert-gefilterten Datei.

**Weitere Anforderungen:**

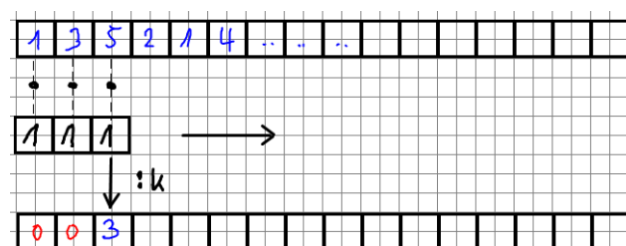
- k (Anzahl der zu mittelnden Werte) soll frei vorgebar sein.
- Geben Sie die Größe des Vektors y aus.
- Geben Sie die Samplerate aus.
- Kommentare (mit %) und keine „Magic Numbers“

**Nützliche Befehle:** pause, size(), zeros(), subplot(), plot()

**Dokumentation:**

- Beschreibung des Versuchszwecks
- Plots für  $k = 10, 50$  und  $250$
- kurze Erklärung und Bewertung

Die Codeabnahme und Vorführung erfolgt am Rechner.



**MATLAB - Signalverarbeitung**

---

**2. Gleitender Mittelwert über k Werte (Input: chirp-Signal)**

Speichern Sie „U1.m“ unter dem Namen „U2.m“ ab.

Ändern Sie „U2.m“ so ab, dass statt der Wave-Datei jetzt ein künstliches Signal (chirp-Signal) verwendet wird.

**Anm.:** Ein chirp-Signal beginnt bei einer Frequenz  $f_{Start}$  und erhöht die Frequenz bis zum Zeitpunkt  $t_{Stop}$  kontinuierlich auf  $f_{Stop}$ .

**Nützliche Befehle:**

```
% Chirp-Parameter
t_Start = 0;      fStart = 10;
t_Stop  = 5;      fStop  = 5000;

FSample    = 10000;  TSample = 1/FSample;
% -----
t = t_Start:TSample:t_Stop;
y = chirp(t, fStart, t_Stop, fStop, 'logarithmic');
```

**Dokumentation:**

- Beschreibung des Versuchszwecks
- Plots für  $k = 10, 50$  und  $250$
- kurze Erklärung und Bewertung

Die Codeabnahme und Vorführung erfolgt am Rechner.

## MATLAB - Signalverarbeitung

### 3. Distortion und Echo

Es ist ein Matlab-Skript „U3.m“ zu schreiben, welches folgende Schritte ausführt:

- a) Laden der Audiodatei „GitRiff02.wav“ in den Vektor  $y$ ,
- b) Größenbestimmung der Datei,
- c) Speicher für das bearbeitete Signal anlegen,
- d) Verzerrung des Signals mit verschiedenen nichtlin. Kennlinien (s.u.).
- e) Realisierung eines Echo-Effektes (s.u.),
- f) Normalisierung des Signals auf Werte  $[-1 \dots +1]$  (= Division durch maximalen Betrag)
- g) Audioausgabe des berechneten Signals (Distortion und Echo).
- h) Plotten der Distortion-Funktion  $y=f(x)$  im Wertebereich  $x = [-1, +1]$

#### Distortion Effekt (Verzerrer):

Der Distortion-Effekt soll in einer eigenen Funktion realisiert werden

$$y = \text{GuitarDistortion}(x, \text{Typ}, D)$$

mit den Übergabe-Parametern:

Eingangssignal	$x$
Distortion-Typ	$\text{Typ}$
Distortionstärke	$D$

Typ 1) Tangens Hyperbolicus

$$y = \tanh(D \cdot x)$$

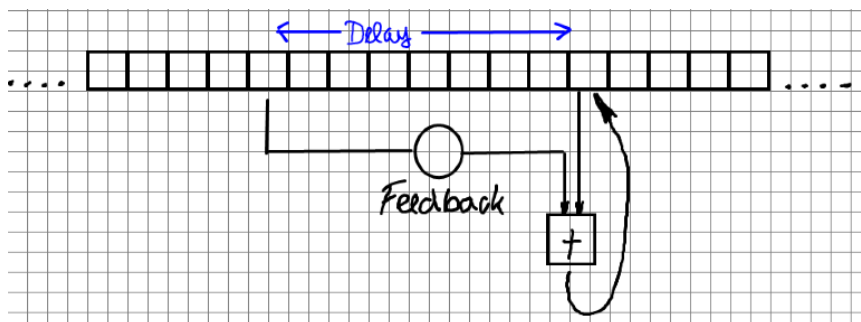
Typ 2) nach Bendiksen

$$z = D \cdot x$$

$$y = -\text{sign}(-z) \cdot (1 - e^{\text{sign}(-z) \cdot z})$$

#### Delay Effekt (Echo):

Hinzufügen des z.B. um  $\text{Delay}=6000$  Samples verzögerten und um  $\text{Feedback}$  gedämpften Signals zum Signal.



## MATLAB - Signalverarbeitung

---

**Nützliche Befehle:** tanh(), sign(), exp(), max(), Element-für-Element-Operationen .\* ./

**Dokumentation:**

- Zweck des Versuchs
- Plot der Distortion-Funktionskennlinien  $Out=f(In)$
- kurze Bewertung (z.B. Einfluß des Parameter *Dist*)

Die Codeabnahme und Vorführung erfolgt am Rechner.