

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

Aufgabenstellung:**1. Mittelwert über n 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 Mittelwertes über die k letzten Werte $y(n)$, $y(n-1)$, ..., $y(n-(k-1))$
(Anm.: Ergebnisse müssen im Wertebereich ± 1 liegen.),
- Plotten des Originalsignals und des gefilterten Signals in einem subplot,
- Audioausgabe der unfilterten 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
- Codeausdruck
- Plots für $k = 10, 50$ und 250
- kurze Bewertung

MATLAB - Signalverarbeitung

2. Mittelwert über n 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
- Codeausdruck
- Plots für $k = 10, 50$ und 250
- kurze Bewertung

MATLAB - Signalverarbeitung

3. Distortion und Echo

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

- Laden der Audiodatei „GitRiff02.wav“ in den Vektor y ,
- Größenbestimmung der Datei,
- Speicher für bearbeitete Signal anlegen,
- Verzerrung des Signals mit verschiedenen nichtlin. Kennlinien (s.u.).
- Skalierung des Ergebnisses, das weder Über- noch Untersteuerung auftritt.
- Realisierung eines Echo-Effektes (s.u.),
- Normalisierung des Signals auf Werte $[-1 \dots +1]$ (= Division durch Maximum)
- Audioausgabe des berechneten Signals (Distortion und Echo).

Distortion Effekt (Verzerrer):

Der Distortion-Effekt soll in einer eigenen Funktion „GuitarDistortion“ realisiert werden.

Übergabe-Parameter: Eingangssignal y
 Distortion-Typ Typ
 Distortionstärke D

Typ 1) Tangens Hyperbolicus

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

Typ 2) nach Bendiksen

$$y = D \cdot y$$

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

Delay Effekt (Echo):

Hinzumischen des z.B. um $Delay=6000$ Samples verzögerten Signals zum unverzögerten Signal mit einem Anteil von $Feedback = 0.35$.

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

Dokumentation:

- Zweck des Versuchs
- Codeausdruck
- Plot der Distortion-Funktionen (eigenes Skript)
- kurze Bewertung (z.B. Einfluß des Parameter $Dist$)