
Übung 1 - Signaltheorie

Table of Contents

Aufgabe 1 - Mittelwert über n Werte	1
Aufgabe 2 - Chirp	2
Aufgabe 3 - Verzerrung	4
Aufgabe 3.1 - Echo	6

Derya Uyargil, Moritz Höwer 26.10.2017

Aufgabe 1 - Mittelwert über n Werte

Datei einlesen und in in Monosignal umwandeln

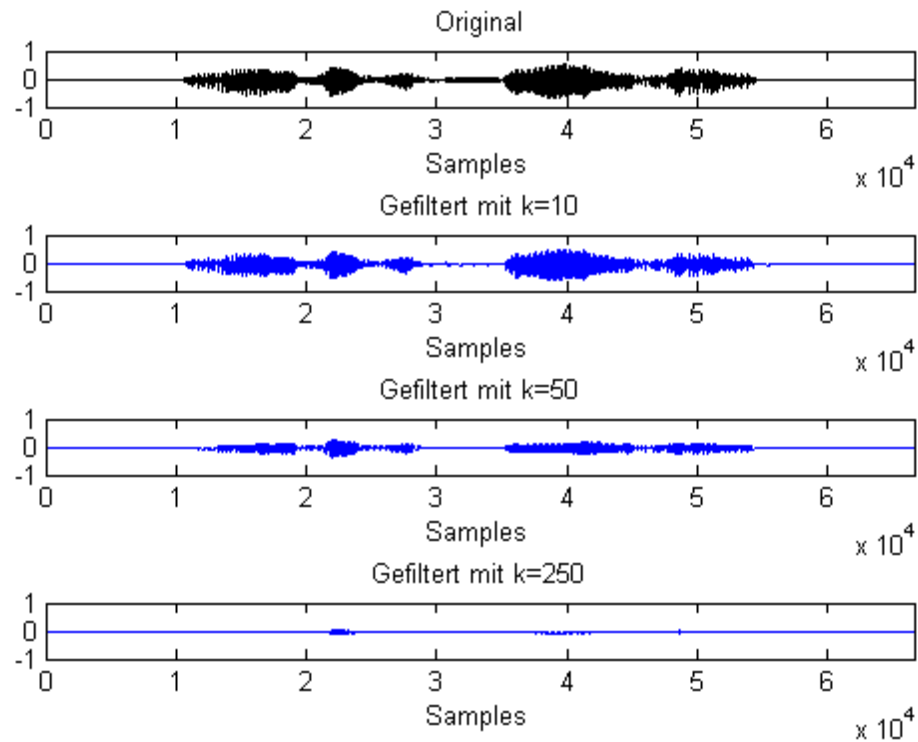
```
[y_in, fS] = audioread('sprech.wav');  
y_in = (y_in(:,1) + y_in(:,2)) / 2;
```

Filter anwenden

```
y_out10 = movmean(y_in, 10);  
y_out50 = movmean(y_in, 50);  
y_out250 = movmean(y_in, 250);
```

Signale plotten

```
subplot(4, 1, 1);  
plot(y_in, 'k');  
title ('Original');  
xlabel 'Samples';  
axis([0 inf -1 1]);  
  
subplot(4, 1, 2);  
plot(y_out10);  
title ('Gefiltert mit k=10');  
xlabel 'Samples';  
axis([0 inf -1 1]);  
  
subplot(4, 1, 3);  
plot(y_out50);  
title ('Gefiltert mit k=50');  
xlabel 'Samples';  
axis([0 inf -1 1]);  
  
subplot(4, 1, 4);  
plot(y_out250);  
title ('Gefiltert mit k=250');  
xlabel 'Samples';  
axis([0 inf -1 1]);  
shg;
```



Zusatzaufgaben

```
[samples, channels] = size(y_in)
samplerate = fS
```

```
samples =  
        66586  
  
channels =  
        1  
  
samplerate =  
        44100
```

Aufgabe 2 - Chirp

Chirp Signal erzeugen Chirp-Parameter

```
tStart = 0; fStart = 10;
```

```
tStop = 5; fStop = 5000;
FSample = 1e4; TSample = 1 / FSample;
% -----
t = tStart:TSample:tStop;
y_in = chirp(t, fStart, tStop, fStop, 'logarithmic');
```

Filter anwenden

```
y_out10 = movmean(y_in, 10);
y_out50 = movmean(y_in, 50);
y_out250 = movmean(y_in, 250);
```

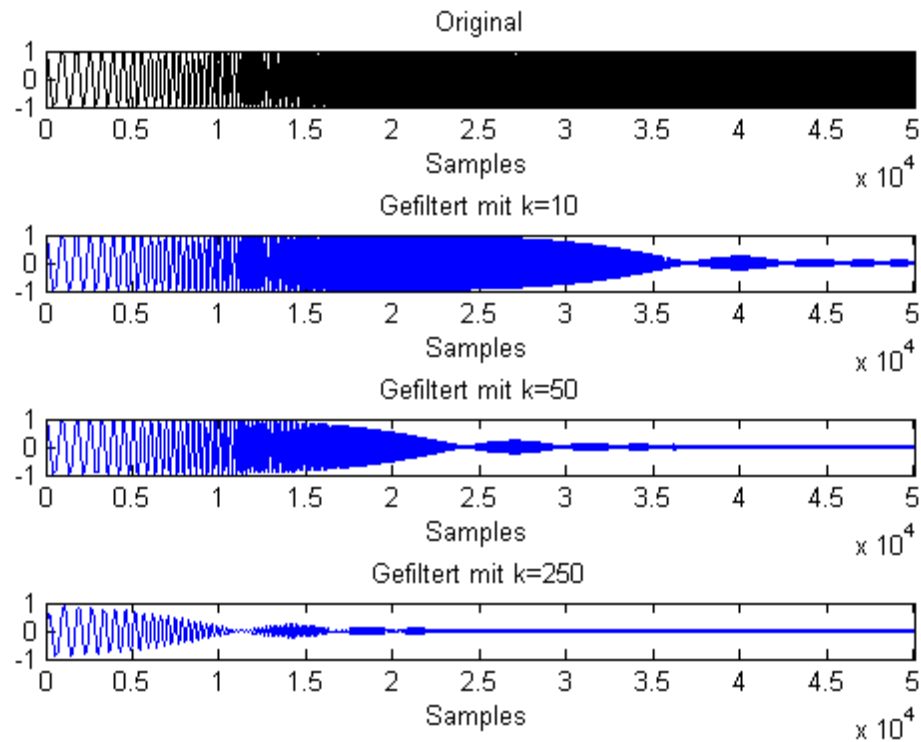
Signale plotten

```
subplot(4, 1, 1);
plot(y_in, 'k');
title ('Original');
xlabel 'Samples';
axis([0 inf -1 1]);

subplot(4, 1, 2);
plot(y_out10);
title ('Gefiltert mit k=10');
xlabel 'Samples';
axis([0 inf -1 1]);

subplot(4, 1, 3);
plot(y_out50);
title ('Gefiltert mit k=50');
xlabel 'Samples';
axis([0 inf -1 1]);

subplot(4, 1, 4);
plot(y_out250);
title ('Gefiltert mit k=250');
xlabel 'Samples';
axis([0 inf -1 1]);
```



Aufgabe 3 - Verzerrung

Datei einlesen und in in Monosignal umwandeln

```
[y_in, fS] = audioread('GitRiff02.wav');
y_in = (y_in(:,1) + y_in(:,2)) / 2;
```

Verzerrungsfunktionen auf [-1;1] anwenden

```
x = linspace(-1, 1, 100);
typ1_1 = GuitarDistortion(x, 1, 1);
typ1_5 = GuitarDistortion(x, 1, 5);
typ1_10 = GuitarDistortion(x, 1, 10);
typ2_1 = GuitarDistortion(x, 2, 1);
typ2_5 = GuitarDistortion(x, 2, 5);
typ2_10 = GuitarDistortion(x, 2, 10);
```

Verzerrungsfunktionen auf Signal anwenden

```
y_out1 = GuitarDistortion(y_in, 1, 5);
y_out2 = GuitarDistortion(y_in, 2, 5);
```

Verzerrungsfunktionen plotten

```
subplot(3, 2, 1);
plot(typ1_1, x);
title('tanh - Verzerrer mit D=1');
grid on;
```

```

subplot(3, 2, 3);
plot(typ1_5, x);
title ('tanh - Verzerrer mit D=5');
grid on;

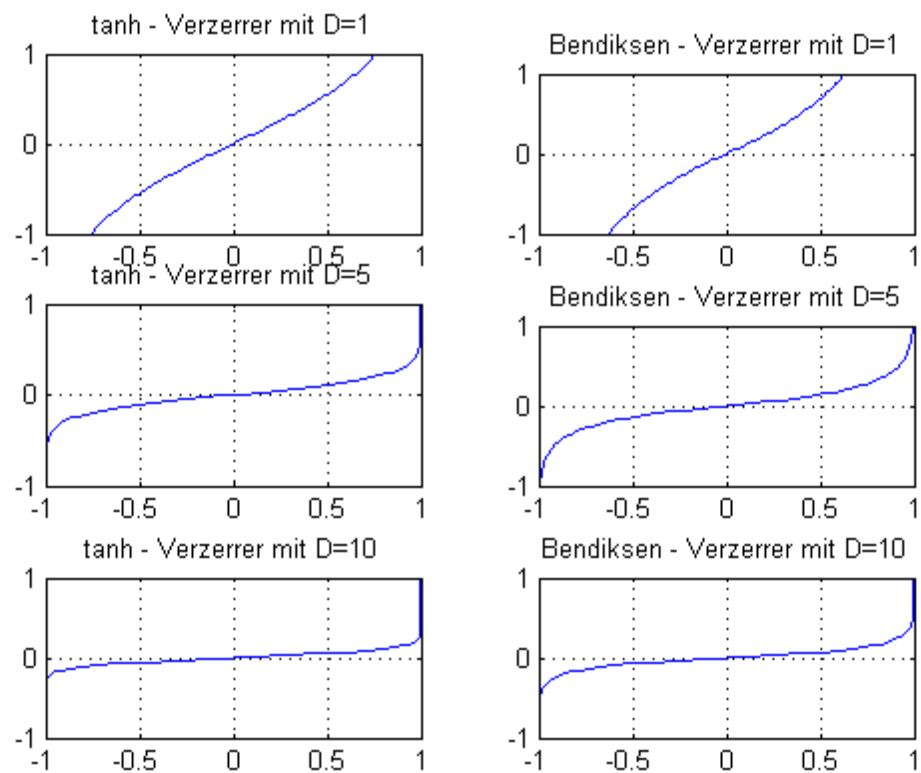
subplot(3, 2, 5);
plot(typ1_10, x);
title ('tanh - Verzerrer mit D=10');
grid on;

subplot(3, 2, 2);
plot(typ2_1, x);
title ('Bendiksen - Verzerrer mit D=1');
grid on;

subplot(3, 2, 4);
plot(typ2_5, x);
title ('Bendiksen - Verzerrer mit D=5');
grid on;

subplot(3, 2, 6);
plot(typ2_10, x);
title ('Bendiksen - Verzerrer mit D=10');
grid on;

```

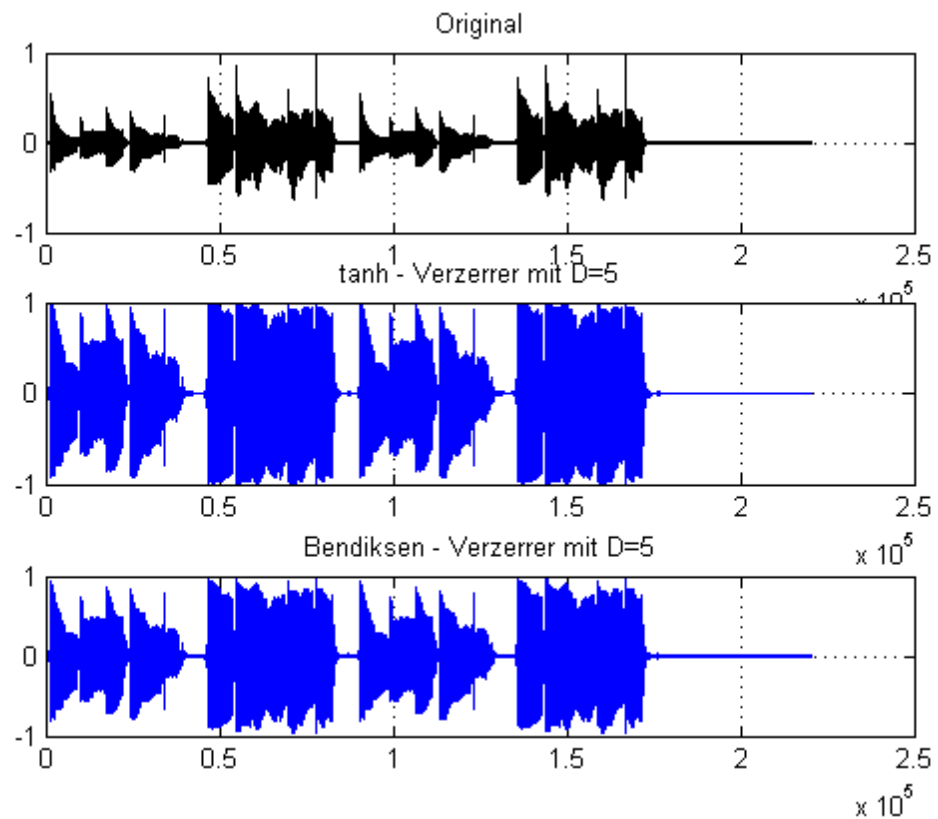


Verzerrtes Signal plotten

```
subplot(3, 1, 1);
plot(y_in, 'k');
title ('Original');
grid on;

subplot(3, 1, 2);
plot(y_out1);
title ('tanh - Verzerrer mit D=5');
grid on;

subplot(3, 1, 3);
plot(y_out2);
title ('Bendiksen - Verzerrer mit D=5');
grid on;
```



Aufgabe 3.1 - Echo

Datei einlesen und in in Monosignal umwandeln

```
[y_in, fS] = audioread('GitRiff01.wav');
y_in = (y_in(:,1) + y_in(:,2)) / 2;
```

Signal verzögern

```
y_in_delay = vertcat(zeros(Delay, 1), y_in(1:(end-Delay)));
y_out = y_in + (Feedback * y_in_delay);
```

Published with MATLAB® R2013b