

# ÜBUNGEN ZU MMS IM WISE 16/17

## BLATT 2

**Abgabefrist:** 07.11.16 - 12:00 Uhr

**Themen:** Fourier-Reihenentwicklung und Fouriersynthese

**Abgabe:** Textaufgaben als PDF-Dateien, Praktische Programmieraufgaben bitte als py-Dateien abgeben. Diese Dateien einzeln oder gepackt bitte im Anhang per E-Mail an [seppke@informatik.uni-hamburg.de](mailto:seppke@informatik.uni-hamburg.de) senden!

### 1 THEORETISCHE BETRACHTUNGEN

10 P.

Die Fourier-Reihenentwicklung (synonym: Fourieranalyse) erlaubt – in einem technischen Sinn – die Zerlegung einer periodischen Funktion in parametrisierte trigonometrische Basisfunktionen und damit eine Analyse des Spektral- oder Frequenzgehaltes einer periodischen Funktion bzw. eines periodischen Zeitsignals. Vice versa kann die Funktion bzw. das Zeitsignal durch Superposition von parametrisierten Sinus- und Cosinus-Funktionen mit minimalem quadratischem Fehler approximiert werden (sog. Fouriersynthese). Wie bekannt, hängt die Größe des Fehlers von der Anzahl der Fourier-Koeffizienten bzw. von der Anzahl der Stützstellen in der Periode  $T$  ab.

Lösen Sie folgende Teilaufgaben schriftlich:

1) Gegeben seien periodische Funktionen über die Zeit. (6 P.)

Berechnen Sie die jeweiligen Fourierkoeffizienten für  $t \in \{-\pi, \pi\}$ .

a)  $f(t) = \sin(t)$

b)  $f(t) = \cos(2t)$

c)  $f(t) = 1$

2) Gegeben seien die Fourierkoeffizienten verschiedener Funktionen über die Zeit. (4 P.)

Wie lauten die Funktionen?

a)  $a_0 = 1, \forall k > 0: a_k = 1, b_k = 0$

b)  $a_0 = 1, \forall k > 0: a_{2k} = 1, a_{2k+1} = 0, b_k = 0$

- 1) Mit folgendem Python-Code können Sie eine Fouriersynthese berechnen. (4 P.)  
Speichern Sie den Code als Skript ab und variieren Sie den Parameter  $n$ .

```
import numpy as np
from matplotlib import pyplot as plt

p = 1000
x = np.linspace(-1,1,p)
f = np.sign(x)

plt.plot(x,f, color='blue', linewidth=2)

sum = np.zeros_like(x)
n = 10 #<--- vary this parameter!
for k in range(1,n,2):
    sum = sum + 4/np.pi*np.sin(k*np.pi*x)/k

plt.plot(x, sum, color='red')
plt.show()
```

Geben Sie Ihre gewählten Werte für  $n$  an und erklären Sie Ihre Resultate.  
Nehmen Sie weiterhin einen sehr großen Wert für  $n$  an.  
Welche Beobachtung können Sie machen?

- 2) Experimentieren Sie mit verschiedenen Signalen: (2 P.)  
Erzeugen Sie dafür eine (diskretisierte) Cosinusschwingung mit einer Frequenz von 400 Hz. Der Ton dauert 3 sec., die Abtastrate ist mit 6000 Hz gegeben, also:

```
hertz=400
sampling_rate=np.linspace(0,3,6000)
cos400=np.cos(2*np.pi*sampling_rate*hertz)
```

Um den Ton hörbar zu machen, nutzen Sie bitte das Modul **scipy.io.wavfile.write** und speichern sie das Array als .wav-Datei. Diese Datei können sie sich anschließend mit einem Programm ihrer Wahl anhören.

Variieren Sie die sound-Ausgabe, indem Sie den Wert 12000 als Abtastrate einsetzen.  
Wie verändert sich der Ton und wodurch lässt sich dieser Effekt erklären?

- 3) Erzeugen Sie mindestens zwei weitere Cosinusschwingungen mit jeweils einer (4 P.)  
Frequenz von 800 Hz und 630 Hz und hören Sie sich deren additive Überlagerung mit Ihrem vorher erzeugten cos400 Signal an.  
Beschreiben und erklären Sie nun vergleichend die resultierenden Klänge.