

Die DCT Anwendung auf Formanten

1. DCT-Anwendung auf ein Signal

a. Analyse

```
vec = bridge[,2]
```

```
vec.dct = dct(vec)
```

```
# Die Amplituden der Cosinuswellen mit Schwingungen 0, 0.5, 1, 1.5... über
```

```
# denselben Zeitraum wie das ursprüngliche Signal
```

```
# Nur die ersten 3 Koeffiziente, k0, k1, k2
```

```
dct(vec, 2)
```

```
vec.dct
```

b. Synthese

```
# Die ersten 3 Schwingungen:
```

```
cr(vec.dct[1], k=0, N=length(vec))
```

```
cr(vec.dct[2], k=0.5, N=length(vec))
```

```
cr(vec.dct[3], k=1, N=length(vec))
```

```
# Die Summierung davon
```

```
cr(vec.dct[1:3], k = c(0, .5, 1), N=length(vec))
```

```
# Einfacher und zum ursprünglichen Signal skaliert: fit=T Argument verwenden
```

```
vec.dctfit = dct(vec, 2, T)
```

```
par(mfrow=c(1,2))
```

```
plot(vec, type="b"); plot(vec.dctfit, type="b")
```

```
# überlagert
```

```
ylim = range(c(vec, vec.dctfit))
```

```
plot(vec, type="b", ylim=ylim)
```

```
par(new=T)
```

```
plot(vec.dctfit, type="l", ylim=ylim, col=2)
```

```
# mit mehr Koeffizienten
```

```
vec.dctfit = dct(vec, 4, T)
```

```
ylim = range(c(vec, vec.dctfit))
```

```
plot(vec, type="b", ylim=ylim)
```

```
par(new=T)
```

```
plot(vec.dctfit, type="l", ylim=ylim, col=2)
```

2. DCT-Anwendung auf eine Trackdatei

```
# Kann die dct() Funktion auf die Frames von einer Trackdatei angewandt werden?
```

```
# z.B. dct vom 10en Segment der Trackdatei vowelax.fdat
```

```
dct(frames(vowelax.fdat[10,2]), 2)
```

```
# geglättet mit den Koeffizienten k0, k1, k2
```

```
glatt = dct(frames(vowlax.fdat[10,2]), 2, T)
```

```
# DCT Koeffiziente von F2 aller Segmente in der Trackdatei
```

```
# Trackdatei F2 aller Segmente geglättet mit den Koeffizienten  $k_0$ ,  $k_1$ ,  $k_2$ 
```

3. DCT-Anwendung auf Spektre

```
# Spektre zum zeitlichen Mittelpunkt
```

```
# fr, fr.l, fr.dft
```

```
d5 = dcut(fr.dft, .5, prop=T)
```

```
par(mfrow=c(1,2))
```

```
plot(d5, fr.l)
```

```
plot(d5, fr.l, fun=mean, dbnorm=T)
```

```
# Anwendung dct auf die Spektre
```

```
# (Die DCT-Koeffiziente von einem Spektrum = Kepstral Koeffiziente)
```

```
d5.dct = fapply(d5, dct, 2)
```

```
eplot(d5.dct[,2:3], fr.l, centroid=T)
```

```
# Spektrale Glättung - zB mit 4 Koeffizienten
```

```
# Nennt sich auch kepstral geglättete Spektre
```

```
glatt = fapply(d5, dct, 4, T)
```

```
par(mfrow=c(1,2))
```

```
j = 45
```

```
plot(d5[j,])
```

```
plot(glatt[j,])
```

```
#besser
```

```
dat = rbind(d5[j,], glatt[j,])
```

```
dat = as.spectral(dat, trackfreq(d5))
```

```
plot(dat, c("rauh", "glatt"))
```

Übungen: DCT-Analyse

1. Untersuchen Sie, inwiefern [a: o: oy] für einen Sprecher oder Sprecherin sich in der Datenbank `stops` differenzieren lassen. (Sie sollen Ellipsen in einem 2D-Raum erstellen, die Ihrer Meinung nach eine maximale Differenzierung zwischen diesen Kategorien ermöglichen).
2. Untersuchen Sie für die Sprachdatenbank `ausenglisch` inwiefern sich /b, d, g/ in wortinitialer Position sich durch die F2-Transitionen differenzieren lassen. (Sie sollen Ellipsen in einem 2D-Raum erstellen, die Ihrer Meinung nach eine maximale Differenzierung zwischen diesen Kategorien ermöglichen).
3. Wiederholen Sie 2. für die /b, d, g/ Trennung in wortinitialer Position für eine Sprecherin oder für einen Sprecher aus der `stops` Sprachdatenbank.