

# MUS4831 - Lydanalyse

Rytme, tempo, og beat tracking

10 oktober 2013

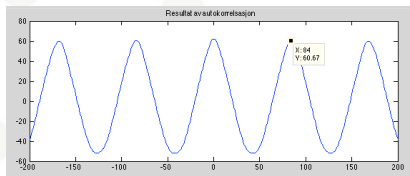
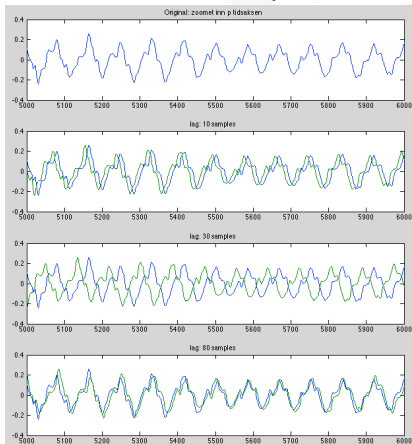
- Forelesning 17. oktober: Tor Halmrast snakker om lyd i rom, filtre, og impulsrespons.
  - Bakgrunnsinfo for dette på foilene fra forelesningen 5. september
  - Se gjennom kapittel 12 i KLANGEN (tilgjengelig på fronter) før forelesningen
- Forelesning 24. oktober: Presentasjon av ideer/utkast til semesteroppgaver
- Oblig 3 ligger ute på kurssiden: Feature Extraction. Frist 25. oktober.

# Denne forelesningen

- Se på noen mekanismer for å analysere tempo og rytme i en lydfil
- Bruke MIR toolbox til å analysere lydfiler med både sterk og svak beat
- Bruke MIR toolbox til å analysere lydfiler med både fast og varierende tempo
- Bruke MIR toolbox til å gjøre analyse på mange lydfiler samtidig
- Tracke tempo i real-time i Max/PD

# Repetisjon fra forrige forelesning: Autocorrelation

Ved å sammenlikne et signal med en tidsforskjøvet versjon av seg selv kan vi finne ut periodisiteter i signalet.



Her har vi en høy korrelasjonskoeffisient ved 84 samples.

Siden vi har en samplingsrate på 44100 finner vi frekvensen ved  $44100/84 = 523$  Hz.

# Autokorrelasjon og musikalsk tempo

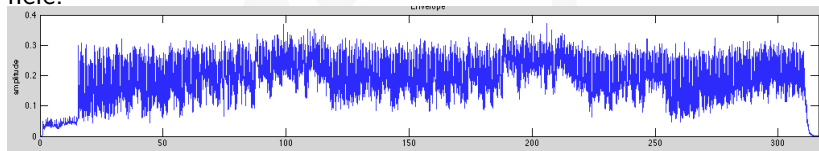
- I forrige forelesning så vi hvordan dette kan brukes til å estimere pitch (raske repetisjoner på mer enn 20 Hz).
- I denne forelesningen skal vi se på tempo (mye tregere repetisjoner, på under 5 Hz, typisk 1-2 Hz)

# Lydfil med fast tempo

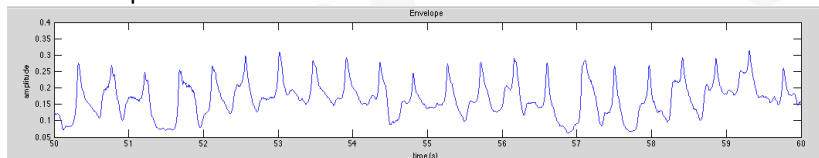
For lydfiler med fast tempo, og sterke beats kan vi gjøre analysen rett på enveloppen til signalet:

- `k = miraudio('Kashmir.wav');`
- `ke = mirenvelope(k)`

hele:



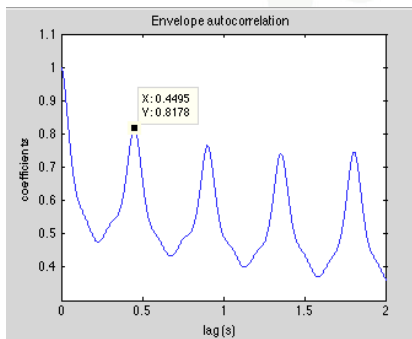
zoom inn på 10 sek:



# To metoder vi har vært innom:

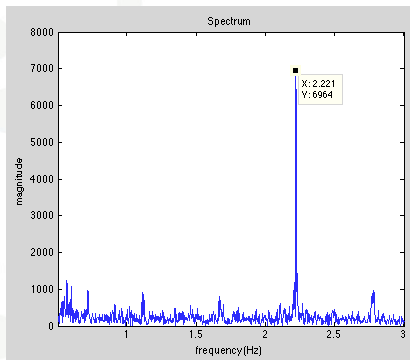
## Autokorrelasjon

- $kac = \text{mirautocorr}(ke)$
- toppunktet = lag i sekunder
- $1/\text{svaret} = \text{tempo i Hz}$
- $\times 60 = \text{tempo i BPM}$



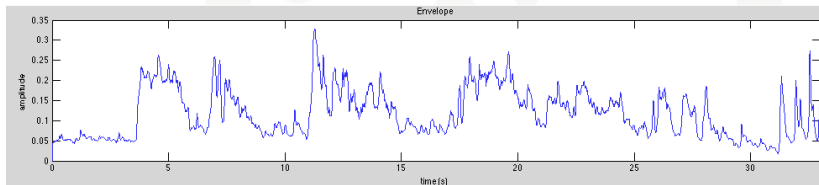
## Spektrum

- $ks = \text{mirspectrum}(ke, 'Max', 3, 'Min', 0.5)$
- toppunktet = svaret i Hz
- $\times 60 = \text{tempo i BPM}$

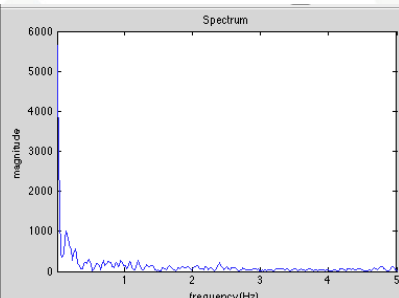
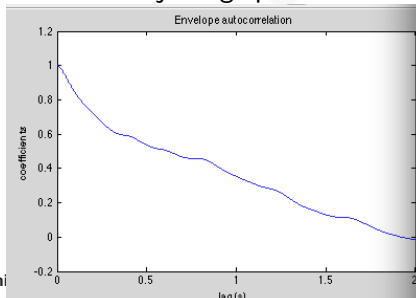


## Fast tempo, men ikke tydelig beat:

- `n = miraudio('NickCave.wav');`
- `ne = mirenvelope(n)`



## Autokorrelasjon og spektrum:





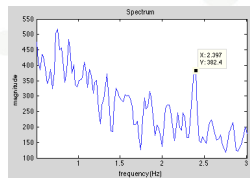
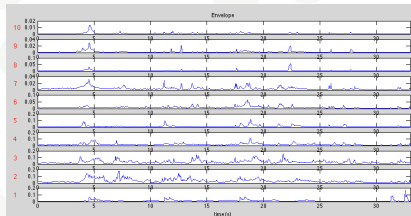
## Fast tempo, men ikke tydelig beat:

En metode som kan hjelpe er å dele opp lyden i ulike frekvensområder.

- $nf = \text{mirfilterbank}(n)$ ;
- $nfe = \text{envelope}(nf)$ ;

Beregner spektrumet (eller autokorrelasjon) på hver av disse, og summerer de sammen:

- $nfs = \text{mirspectrum}(nfe, 'Max', 3, 'Min', 0.5)$
- $nfss = \text{mirsum}(nfs)$
- Vi finner et slags toppunkt ved 2.38 Hz
- Som tilsvarer et tempo på ca 143 BPM (som er dobbelt av det egentlige tempoet)



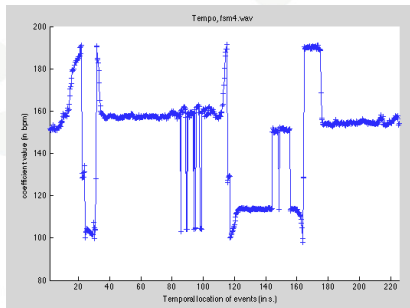
# Beregne tempo i praksis

Vanligvis kan dette gjøres en del enklere:

- `mirtempo(n)` gir 146.7022 BPM (som er dobbelt tempo)
- `mirtempo(k)` gir 133.2853 BPM

Hva med filer med varierende tempo?

- `f = miraudio('fsm4.wav');`
- `mirtempo(f)` gir oss 155.4294 BPM
- Løsningen er å bruke 'frame': `mirtempo(f,'frame')`

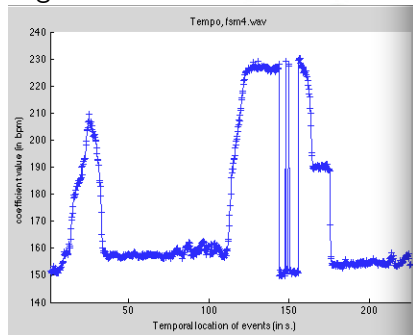


# Beregne tempo i praksis

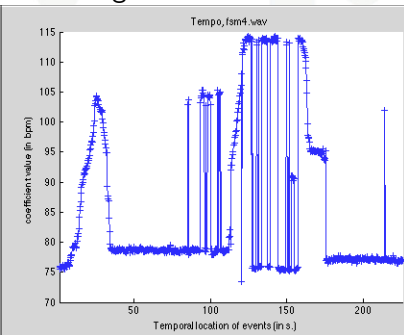
Vi kan sette inn terskelverdier for max og min tempo:

- `mirtempo(f,'Min',70,'Max',120,'frame')`

avgrenset til 140-240 BPM



avgrenset til 70-120 BPM



# Statistisk beregning for en mengde lydfile

```
a = [1 2 3 4 5 6 7 8 9];
```

## Diverse statistiske variabler

gjennomsnitt

median: midterste verdi

typetall: den verdi det er flest av  
antall

minimum

maksimum

standardavvik

varians

spredning

spisshet

mean(a)

median(a)

mode(a)

length(a)

min(a)

max(a)

std(a)

var(a)

skewness(a)

kurtosis(a)

# Egenskapsutrekning for en hel mappe

- `x = mirfeatures('Folder', 'Stat')`

Se på en feature:

- `x.rhythm.tempo.Mean`

Hvis du ønsker å se verdiene i en kolonne

- `x.rhythm.tempo.Mean'`

Kan beregne gjennomsnitt og standardavvik av gjennomsnittstempoet for alle filene

- `mean(x.rhythm.tempo.Mean)`
- `std(x.rhythm.tempo.Mean)`

Hvor stor er forskjellen mellom den langsomste og raskeste filen?

- `max(x.rhythm.tempo.Mean) - min(x.rhythm.tempo.Mean)`

# Plotting

Noen typer plot:

- `boxplot(x.rhythm.tempo.Mean)` % Statistikk for alle verdiene
- `plot(x.rhythm.tempo.Mean)` % Enkelt plot av alle verdiene
- `hist(x.rhythm.tempo.Mean)` % Histogram av alle verdiene

Lager et plot med liggende grafer som kan eksporteres

- `barh(x.rhythm.tempo.Mean)`
- `axis tight;`
- `grid on;`
- `title('Tempo av mine lyder')`
- `xlabel('BPM')`
- `set(gca,'ytick',1:length(x.rhythm.tempo.Mean))`
- `set(gca,'yticklabel', x.FileNamees)`

# Eksportering av data

- Data kan eksporteres til en tekstfil og åpnes i f.eks. Excel eller OpenOffice
- `mirexport('minelyder.txt',x)`

# Tempotracking i sanntid

- i Max kan man bruke **sync~** objektet til å få informasjon om tempo.
- **average~** med argument 'absolute' eller 'rms' kan brukes til å infomrasjon om lydenergi
- **delta~** beregner den deriverte
- **thresh~** detekterer når kurven passerer over og under et nivå
- **biquad~** er et filter som kan brukes til å isolere deler av frekvensspekteret

Noen eksempelpatcher for både Max og PD ligger ute på kurssiden