Signal modifications using the STFT

summer 2006 lecture on analysis,
modeling and transformation of audio signals

Axel Röbel
Institute of communication science TU-Berlin
IRCAM Analysis/Synthesis Team

25th August 2006

Contents

1 STFT domain transformations

The transformations that will be discussed are:

- time invariant filtering
- time variant filtering
- time stretching
- sample rate conversion
- cross synthesis

2 Filtering

2.1 Time invariant filtering

Applying a time invariant FIR filter transfer function to $X(lI,k)$ performs approximate filtering

- inverse transformation according to [Röb06, section 2.2]

$$y(n) = \frac{1}{N}\frac{\sum_{l=-\infty}^{\infty} r_N(n-lI)\sum_{k=0}^{N} X(lI,k)H(k)e^{j\Omega_N k(n-lI)}}{\sum_{l=-\infty}^{\infty} w(n-lI)} \tag{1}$$

- for which we may replace frequency domain multiplication by means of time domain convolution **IF** for the window length M, the length of the impulse response R and the length of the DFT N the relation $N > M+R-1$ holds. We obtain **(FIR filter only!!)**

$$y(n) = \frac{\sum_{l=-\infty}^{\infty} x(lI,n) * h(n)}{\sum_{l=-\infty}^{\infty} w(n-lI)} \tag{2}$$

$$y(n) = \frac{\sum_{l=-\infty}^{\infty}(x(n)w(n-lI)) * h(n)}{C(n)} \tag{3}$$

- Assume normalization factor is constant

$$C(n) = \sum_{l=-\infty}^{\infty} w(n - lI) = K \tag{4}$$

Due to the fact the convolution is a linear operation, we may exchange normalization and convolution, into

$$\begin{aligned} y(n) &= \frac{\sum_{l=-\infty}^{\infty} x(n)w(n-lI)}{K} * h(n) & (5) \\ y(n) &= x(n) * h(n) & (6) \end{aligned}$$

Which shows that in this case the frequency domain filtering is equivalent to time domain filtering.

The general case of $C(n) = K + \epsilon(n)$ is treated in section **6.1**

2.2 Time variant filtering

Time variant filtering in the STFT domain means that the transfer function that is applied to the DFT of the current frame changes with the frame index l.

- again inverse transformation according to [Röb06, section 2.2]

$$y(n) = \frac{1}{N}\frac{\sum_{l=-\infty}^{\infty} r_N(n-lI)\sum_{k=0}^{N} X(lI,k)H(lI,k)e^{j\Omega_N k(n-lI)}}{\sum_{l=-\infty}^{\infty} w(n-lI)} \tag{7}$$

- for sufficiently large DFT length N we may again replace by means of time domain convolution

$$\begin{aligned} y(n) &= \frac{\sum_{l=-\infty}^{\infty} x(lI,n) * h_{lI}(n)}{C(n)} & (8)\\ &= \frac{\sum_{l=-\infty}^{\infty}\sum_{m=-\infty}^{\infty}(x(n-m)w(n-m-lI))h_{lI}(m)}{C(n)} & (9)\end{aligned}$$

- reorder summation

$$y(n) = \frac{\sum_{m=-\infty}^{\infty} x(n-m) \sum_{l=-\infty}^{\infty} (w(n-m-lI)) h_{lI}(m)}{C(n)} \quad (10)$$

- Assuming $C(n) = K$

$$y(n) = \sum_{m=-\infty}^{\infty} x(n-m) \frac{\sum_{l=-\infty}^{\infty} (w(n-m-lI)) h_{lI}(m)}{K} \quad (11)$$

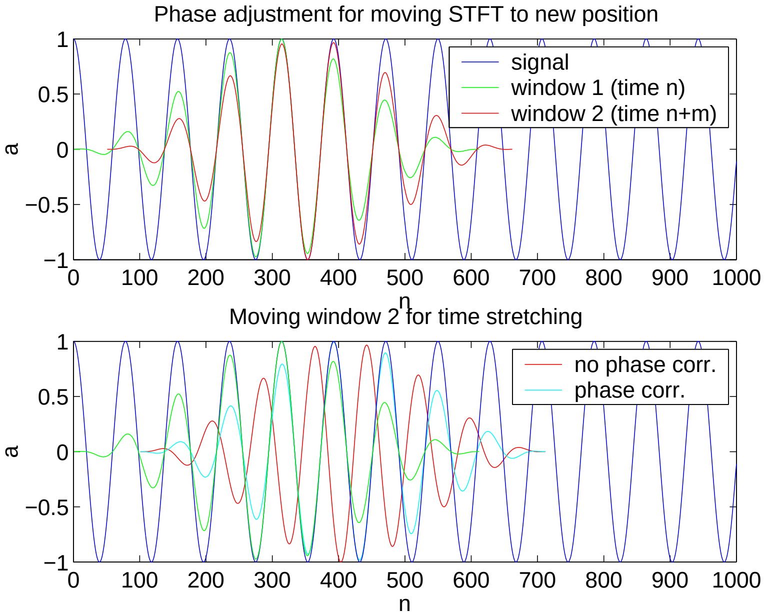
- time varying impulse response is constructed by means of weighted averaging of individual impulse responses $h_{lI}(m)$

3 Time stretching using the phase vocoder

Time stretching → Slow down amplitude and frequency evolution!

The phase vocoder is an STFT representation by means of **amplitude** and **frequency**.

Basic Idea using the phase vocoder: Adjust synthesis rate of the STFT frames and **correct** the STFT data such that **successive frames overlap coherently**.


Phase adjustment for moving STFT to new position
signal
window 1 (time n)
window 2 (time n+m)
a
1
0.5
0
−0.5
−1
0
100
200
300
400
500
600
700
800
900
1000
n
Moving window 2 for time stretching
no phase corr.
phase corr.
a
1
0.5
0
−0.5
−1
0
100
200
300
400
500
600
700
800
900
1000
n

3.1 Parameter adaptation

How do we have to **change the frame data** to achieve coherent overlap after **moving the frame in time?**

- Deriving the **parameter evolution** for a STFT frequency bin in general is **not trivial**.
- The proper modification **depends** on the signal that is represented.
- One of the most important cases is handling of stationary **sinusoids**, therefore **sinusoidal model is assumed**.

Assume:

$$x(n) = e^{j\Omega n+\phi} \tag{12}$$

$$w(n) = \text{analysis window with DFT} \quad W(k) \tag{13}$$

Then we have:

Only the phase is changing with the window position such that

$$\begin{aligned} X(lI,k) &= (e^{j((lI+\frac{N-1}{2})\Omega+\phi)})\cdot(e^{-j\frac{N-1}{2}w})\cdot W(k-\Omega) \qquad (14)\\ &= Ke^{jlI\Omega} \qquad (15)\end{aligned}$$

Phases of all bins change synchronously if they represent the same sinusoid!!

3.2 Modifying phase and the phase vocoder

- For **stationary sinusoids** the frequency Ω can be derived by means of measuring phase difference between successive frames.
- Using the estimated frequency the phase values of the frames that are moved in time can be updated to coherently overlap.
- **Problem:** for frequency Ω greater then Ω_{lim} and step size I **phase difference will wrap around** 2π!

$$\Omega_{lim} = \pm\frac{\pi}{I}$$

- **Solution:** Amplitude of STFT will be significant only within the windows bandwidth around signal frequency Ω, so we need the frequency estimate only in a close neighborhood to the peak maximum.
- therefore, we estimate the **frequency offset** Θ_k of the sinusoid to center frequency of bin k (see Appendix section **6.2**). From

$$X(lI, k) = Ke^{jlI(\Theta_k + \frac{2\pi}{N}k)} \tag{16}$$

and making use of the notation

$$[\phi]_{2\pi} = (\frac{\phi}{2\pi} - \mathrm{round}(\frac{\phi}{2\pi}))2\pi \tag{17}$$

to denote the calculation of the principle value of the argument ϕ we get

$$\hat{\Theta}_k = \frac{[\arg(X((l+1)I, k)) - \arg(X(lI, k)) - I\frac{2\pi k}{N}]_{2\pi}}{I} \tag{18}$$

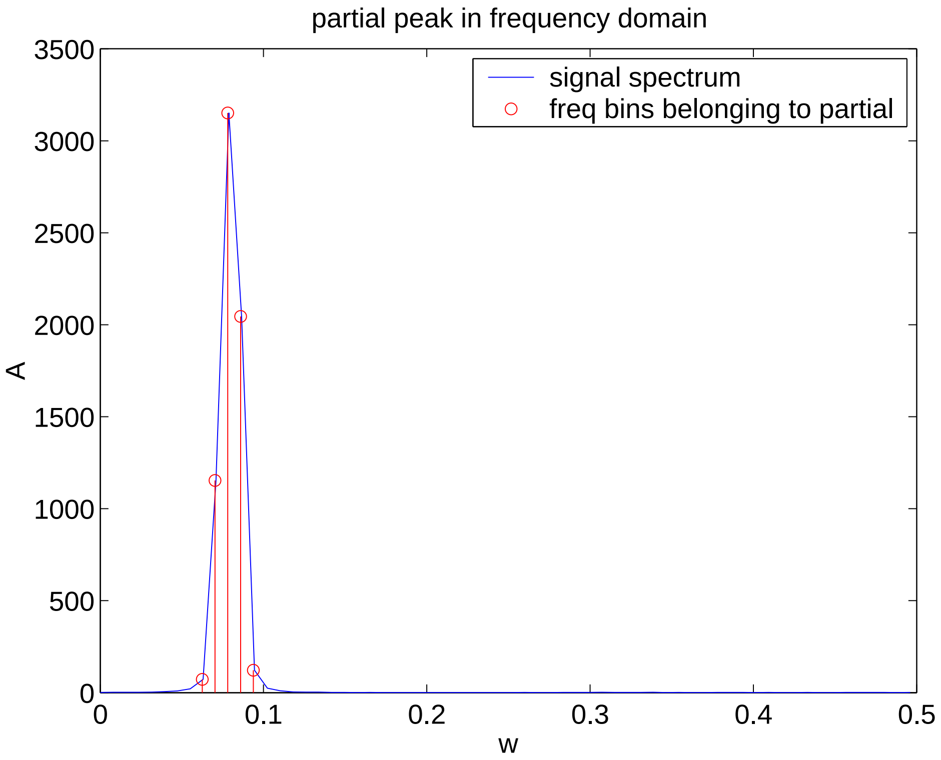
- For synthesis: the phase at bin k of frame l is obtained from the phase of frame $l-1$ by summation of the previous phase and the phase offset between the frames for the new frame offset S

$$\Phi_s(l, k) = S(\Theta_k + \frac{2\pi}{N}k) + \Phi_s(l-1, k)$$

- Transformation of the STFT representation into amplitude/frequency values yields the **phase vocoder** representation of the signal.
- **Standard phase vocoder** approach handles each frequency bin **independently**.

3.3 Local phase synchronization problem

- as shown in eq. (15) the **phase increment** is constant for all frequency bins that **represent the same sinusoid**.
- Due to **instability of phase integration** the **frequency estimation errors** will produce **frequency inconsistencies** for frequency bins that are related to the **same sinusoid**.
- Problem: the STFT bins will **loose vertical synchronization** and the synthesized partial suffers from **amplitude modulation**.


partial peak in frequency domain
signal spectrum
freq bins belonging to partial
A
w
3500
3000
2500
2000
1500
1000
500
0
0
0.1
0.2
0.3
0.4
0.5

Dolson/Laroche Phase synchronization

Vertical phase synchronization: (proposed by [DL99])

- Calculate standard phase update only for **center of spectral peak**.
- Enforce vertical synchronization between center peak and the neighboring bins by simply **copying the phase differences** from the analysis frame.

Question: What bins are to be synchronized - What bins belong to the same partial?

Dolson/Laroche:

- use **all bins** between peak and **next amplitude minimum**.
- experimental evaluation proofs **selection is sub optimal**.
- synchronization of wrong bins introduce **artifacts**.

New Approach:

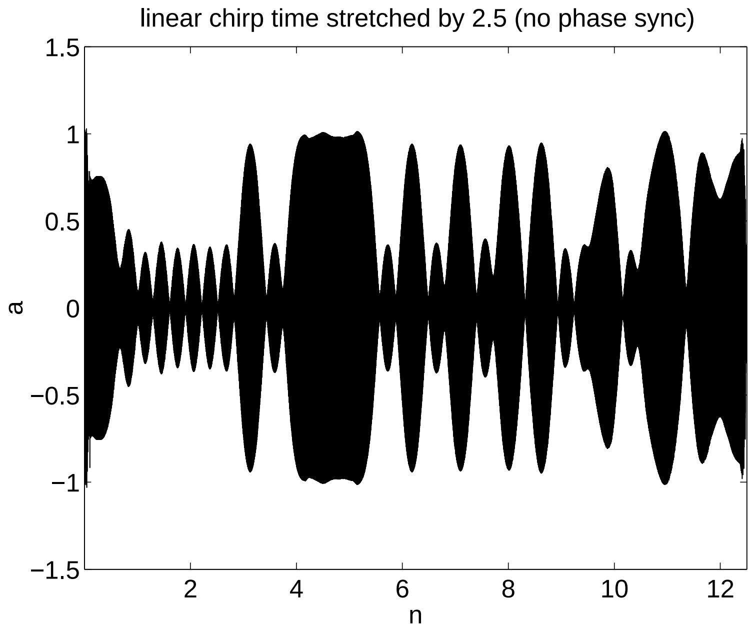
- Group bins according to **frequency estimate**.
- Only bins with frequency estimate close to spectral peak are considered to **belong to the same peak**.

Result:

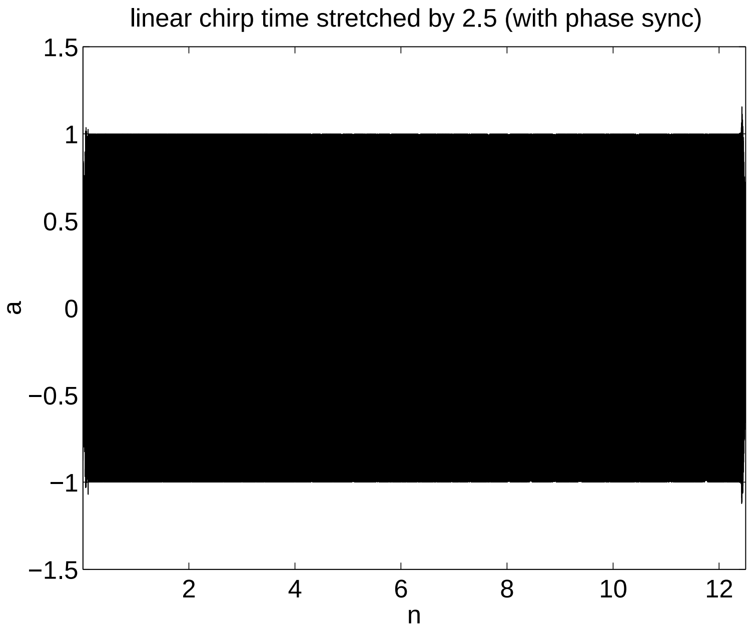
- Phase synchronization significantly **reduces amplitude modulation** because it avoids random cancellation of neighboring bins.

Sound examples

Comparing results for time stretching sinusoid with factor 2.5



Standard phase vocoder

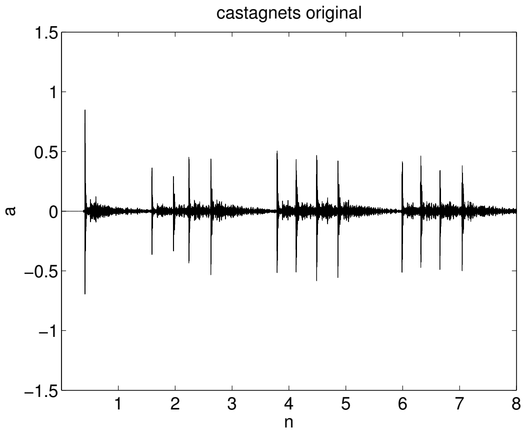


With vertical phase synchronization

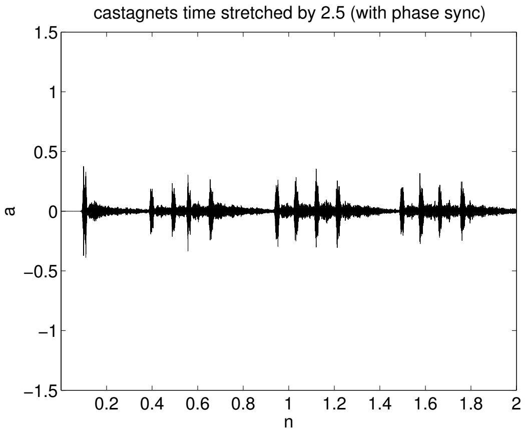
3.4 Transient detection and preservation

- The phase vocoder signal processing is based on the assumption of **stationary sinusoids**.
- For sinusoids with **abrupt changes in amplitude** the phase update equations produce **significant artifacts**.

Example: Time stretching castagnets by factor of 2.5 with **phase synchronization**.



Original Signal



Time stretched factor 2.5

Understanding the problem

Time stretching with phase vocoder

- calculate **STFT**
- **reposition frames**
- **update frame spectrum** according to new position,

Repositioning frames:

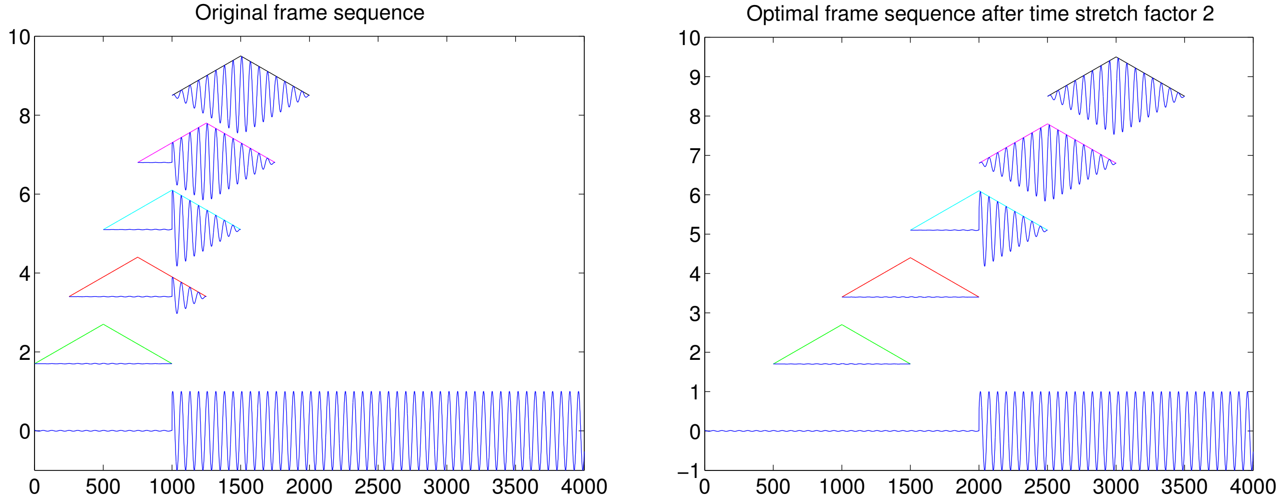
- **stationary sinusoid** signal
  → requires **change of phase spectrum**, only.
- **transient sinusoid** signal
  → requires change of **transient position**
  involve **phase and amplitude spectrum**.

Analysis of the sources of error

Spectral evolution when window moves over transient

- **phase:** changes are nearly linearly
- **amplitude:** complicated nonlinear changes depending on transient position and window form.

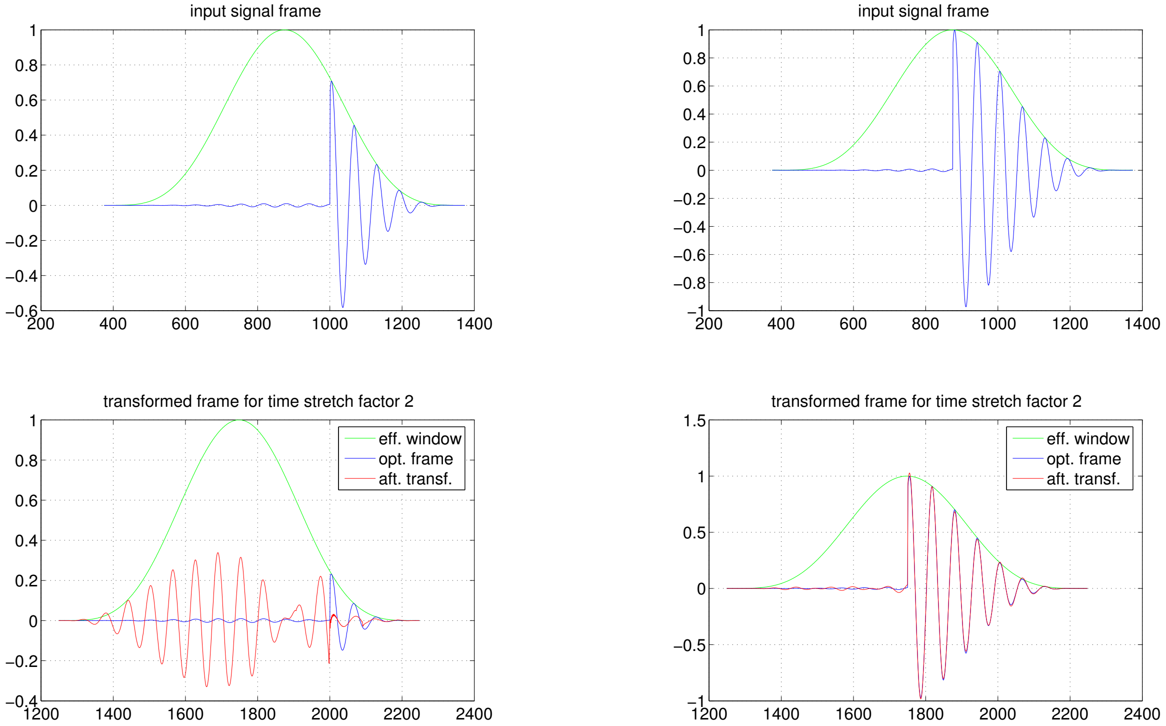
Required changes for **shifting transient frame** to new position:



more on the sources of error

transient after phase vocoder processing

window center before transient window center after transient



Assumption: previous output frame has **proper phase**.

Proper phase handling

Location of window center:

- **before the transient** → reuse old amplitude and frequency for phase vocoder processing to extend signal behavior from previous frames
- **close to the transient** → reinitialize phase and amplitude to exactly reproduce transient
- **after the transient** → do standard phase vocoder processing

Remarks

- phase vocoder processing of previous frames in the transient bins before the transient starts is optimal for frames that would have to move the transient out off the frame.
- for frames that have to move the transient to the right it is suboptimal and has the effect to weaken the transient
- initializing the transient in the center of the window is optimal with respect to position
- the frame with phase initialization is the only one without any error and should have maximum impact on the signal which is the case when the transient is in the center.

3.5 Transient detection

- Most transient detection algorithms are based on a detection of **rapid energy changes** in signal bands.
- The energy changes are calculated from the **amplitude differences** between **successive STFT analysis frames**.
- Using **large energy bands** is sufficient to **detect** transients, however, for **transient preservation** in the phase vocoder a **local decision in frequency** should be achieved.
- In the **IRCAM phase vocoder** transient detection is based on a recent and efficient possibility to estimate the **center of gravity** of the signal under the analysis window based on the **group delay**.

Estimating average signal position and group delay

- The **center of gravity (COG)** of a (squared) signal is defined to be

$$\overline{n} = \frac{\sum_n n|s(n)|^2}{\sum_n |s(n)|^2} \tag{19}$$

- According to [Coh95] and as shown in section **6.3** the COG can be calculated in the **spectral domain** by means of

$$\overline{n} = -\frac{\int_w \frac{\partial \arg(S(w))}{\partial w}|S(w)|^2 dw}{\int_w |S(w)|^2 dw}. \tag{20}$$

- the derivative of the phase of the signal spectrum $\phi(w) = \arg(S(w))$ with respect to frequency is called **group delay**.

$$n_g(w) = -\frac{\partial \phi(w)}{\partial w} \tag{21}$$

- the **group delay** describes the contribution of the spectral energy distribution to the **center of gravity of the squared signal**.

Qualitative description of the phase function

- **stationary sinusoids**:

$$n_g(w) = 0 \tag{22}$$

- **onsets:** the COG will be approximately the same for all frequencies, and the phase spectrum will be **linearly decreasing**.

$$n_g(w) \approx -kw + C \tag{23}$$

- **chirp** due to symmetry the phase slope has a sign change at current instantaneous frequency.

$$n_g(w) \approx kw^2 \tag{24}$$

Efficient calculation of the signal mean time

- the theory of **reassignment** has shown [AF95] that the group delay may be derived efficiently by means of calculating the signal spectrum using the analysis window $h(n)$ and the same window multiplied with time $h(n)n$.

$$X_h(w, n_0) = \sum_n x(n)h(n-n_0)e^{-jwn} \quad \text{and} \tag{25}$$

$$X_{hT}(w, n_0) = \sum_n x(n)h(n-n_0)ne^{-jwn} \tag{26}$$

$$n_g(w, n_0) = \frac{\mathrm{real}(\overline{X_h(w, n_0)}X_{hT}(w, n_0))}{|X_h(w, n_0)|^2} \tag{27}$$

- using eq. (20) and eq. (27) the mean time can be calculated for **each individual spectral peak**.

Phase evolution

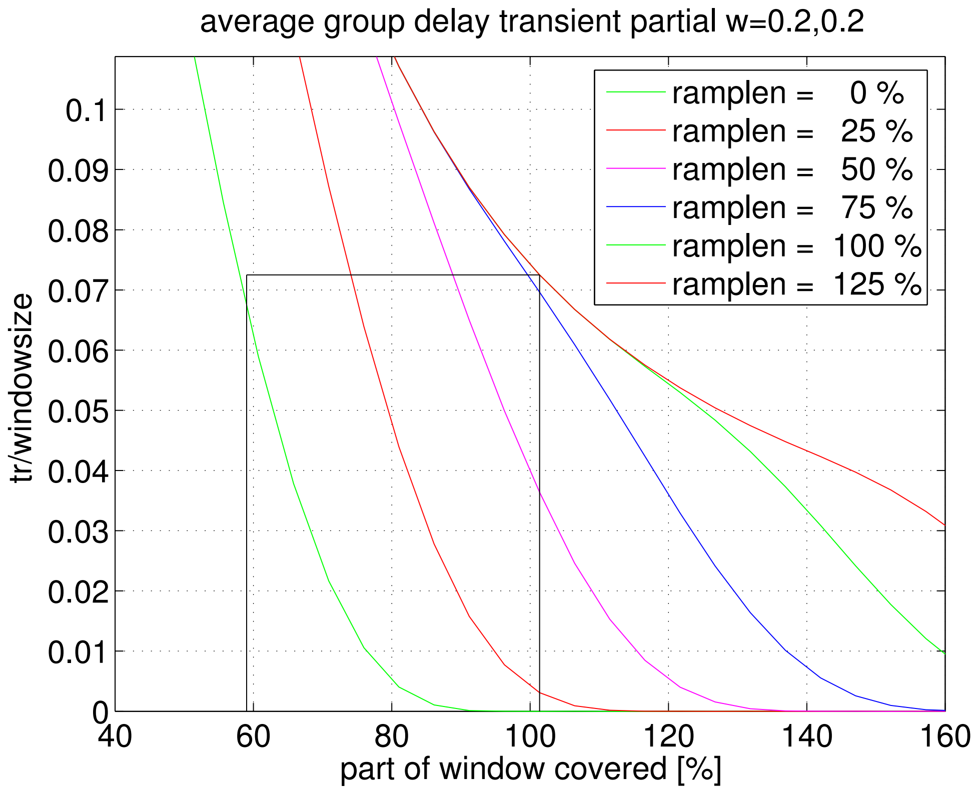
- calculate COG for **each independent component of spectrum** (each spectral peak).
- If the analysis window is moving over a partial with fast attack the COG will first be located at the **far right end of the window**.
- moving on the COG will **decay to zero** and during the release part of the partial it will then **move to the far left** end of the window.
- The exact time evolution depends on the **form of the transient**.
- phase spectrum has two trends:

  1. phase slope decreases due to the fact that the window covers more and more of the signal
  2. the phase value increases according to the frequency of the sinusoid

Proper threshold for transient detection

- To derive a suitable threshold for the **COG** we have compared the **movement of the COG for different forms of transients**.

- For a **threshold** of $t_{glim} = 0.07M$ (M=window size) the detection of all types of transients will be finished after the **transient passed half of the window** and before the transient is **fully covered by the window**.
- **Problem:** The transient detector will detect all situations with the COG beyond the threshold including situations of partial modulation as encountered in **noisy regions**.
- Detected transient energy has to be checked for time **synchronous behavior across frequency** to improve robustness of transient detector.


average group delay transient partial w=0.2,0.2
tr/windowsize
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
40
60
80
100
120
140
160
part of window covered [%]
ramplen = 0 %
ramplen = 25 %
ramplen = 50 %
ramplen = 75 %
ramplen = 100 %
ramplen = 125 %

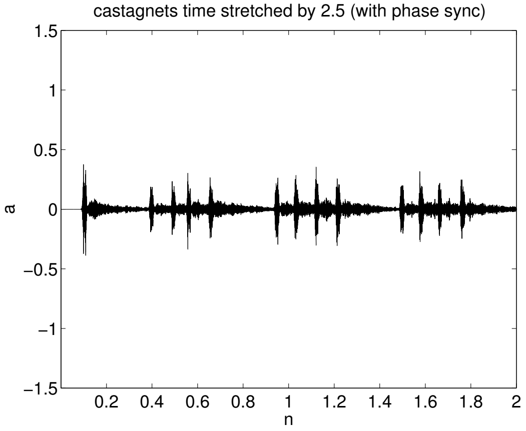
3.6 Transient processing during time stretching

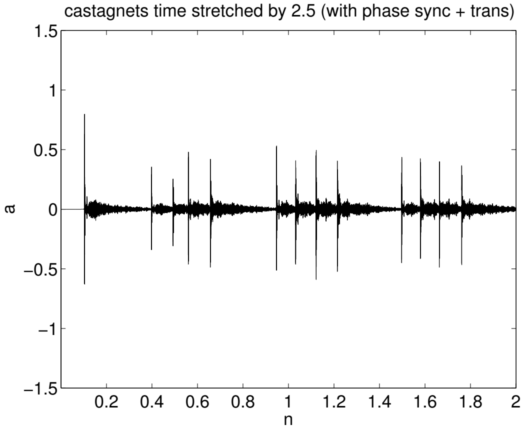
To properly resynthesize a transient we have to approximately **reproduce the phase** of the spectrum.

Strategy

- If attack transient has been detected we use the **amplitude and frequency of previous frame for synthesis** such that the beginning attack will not **spread across frames**.
- If attack transient detector releases the **attack is close to the center of current frame**. **Original phase and amplitude values** are used during synthesis to exactly reproduce the transient.
- **Missing overlap** from the previous frames is compensated by **multiplying amplitudes of transient bins** by a constant factor of 1.5-2.
- For the following frames the attack has already **passed through the window center** and the phase integration will be sufficiently correct such that the transient remains in tact.

Example: Time stretching castagnets by factor of 2.5 with/without **transient preservation**.





Time stretched factor 2.5 Time stretched factor 2.5 with transient pres.

Original Signal

4 Resampling and transposition

Resampling

- Resampling is an operation that changes the sample rate of the signal
- the sample rate is the frequency reference point of the discrete signal.
- a standard approach to achieve signal transposition is to resample too another sample rate, and play the signal with the original rate.
- there are two approaches to sample rate conversion: in frequency domain and in time domain.

4.1 frequency domain resampling

In the first part of this lecture *(Fundamentals of time-frequency analysis)* we have seen that a change of the DFT length in the time domain interpolates the spectrum to another grid in the frequency domain.

- if zeros are appended the grid becomes finer sampling the same underlying continuous spectrum which is the FT of the signal segment
- if the signal is cut the grid becomes coarser, and the spectrum changes if the samples that have been cut were not equal to zero.

Due to the duality of the frequency and time domain the same operation can be applied in the frequency domain (see section **6.4**).

Summary:

- Adding zeros in the DFT spectrum increases sample rate,
- removing bins of the spectrum decreases the sample rate,
- care has to taken that the symmetries of the DFT of a real signal are not destroyed →

only pairs of bins can be deleted or added,

- Frequency bin at $X(N/2)$ has to be set to zero,
- the bins have to be added deleted at the highest frequency of the DFT which for DFT size N is located at $N/2-1$.

Discussion:

- because the number of bins in the DFT can only be changed by an integer multiple of 2 resampling in the frequency grid cannot be used to create a continuous change of the sample rate.
- because the transposition will not change the sample rate the window length will change which slightly increases the complexity of the overlap add algorithm.

4.2 time domain resampling

- discrete time signal is representation of a band limited continuous time signal.
- time domain resampling is best understood in the frequency domain
- suppose a discrete time signal $x(n)$ with

$$X(w) = \sum_{n=-\infty}^{\infty} x(n)e^{-jwn} \tag{28}$$

- assume an expansion procedure that substitutes

$$\begin{aligned} y(n) &= \begin{cases} x(\frac{n}{L}) & \text{for} \quad n = 0, \pm L, \pm 2L, \ldots \\ 0 & \text{else} \end{cases} && (29) \\ &= \sum_{k=-\infty} \infty x(k)\delta(n-kL) && (30) \end{aligned}$$

and changes the sample rate from Ω into $L\Omega$.

The resulting Fourier spectrum would be

$$\begin{aligned} Y(w) &= \sum_{n=-\infty}^{\infty} y(n)e^{-jwn} && (31) \\ &= \sum_{n=-\infty}^{\infty} \sum_{k=-\infty} \infty x(k)\delta(n-kL)e^{-jwn} && (32) \end{aligned}$$

and with $Ln' = n$

$$\begin{aligned} Y(w) &= \sum_{n'=-\infty}^{\infty} x(n')e^{-jwn'L} && (33) \\ &= \sum_{n'=-\infty}^{\infty} x(n)e^{-jwLn'} && (34) \\ &= X(wL) && (35) \end{aligned}$$

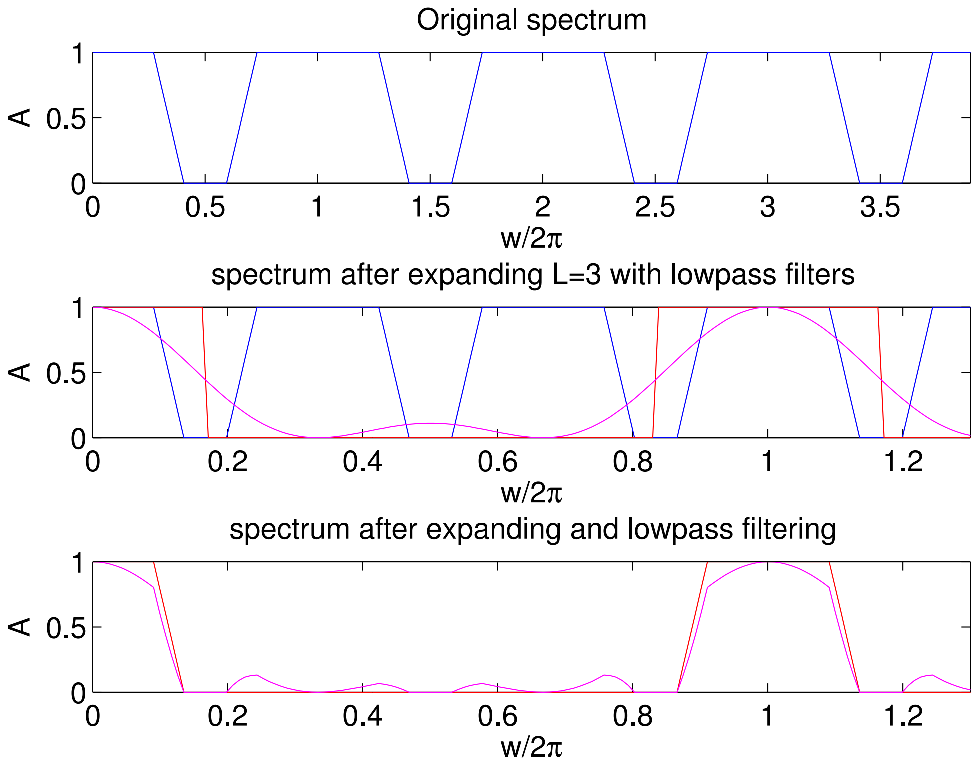


Figure 1: time domain expansion and low pass filtering, top: original periodic spectrum, middle: after expanding with $L = 3$ (in red ideal interpolation lowpas, in magenta linear interpolation lowpass), bottom: after interpolation with ideal and linear lowpass

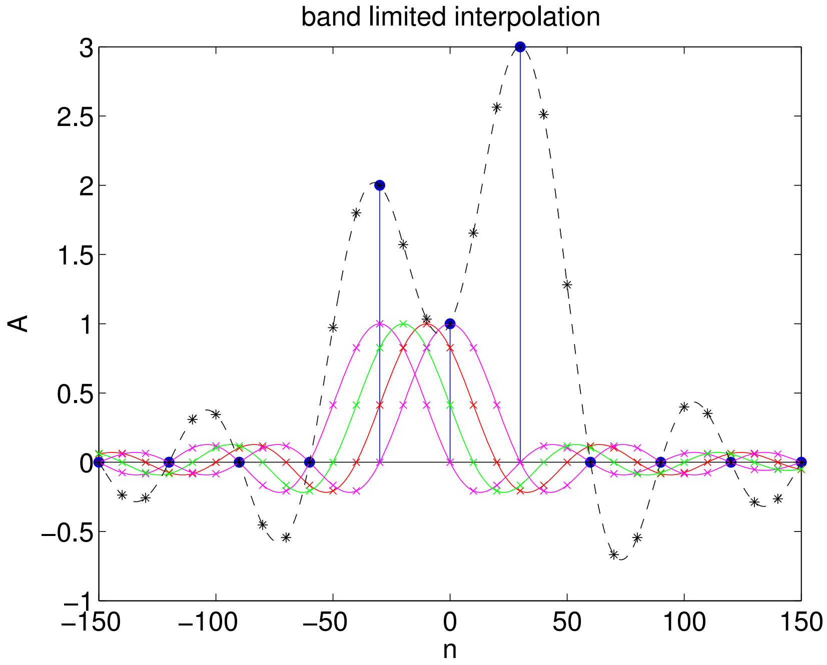


Figure 2: ideal lowpass applies sinc interpolation.

- convolution with length M-point filter and N-point signal, costs: MN
- polyphase implementation, using different $\frac{M}{L}$-point filter (red, green, magenta) for each sample position of the interpolating grid, costs: $L\frac{M}{L}\frac{N}{L} = \frac{NM}{L}$

Summary:

- time domain upsampling can be achieved by means of expansion and filtering with interpolation filter.
- Expansion creates rescaling of spectrum to new sample rate.
- filtering removes spectral duplicates
- linear interpolation creates a maximum attenuation of -6db in passband and at the worst position a -6dB attenuation in the stopband.
- if the original signal has lower bandwidth the linear interpolation becomes much better.
- time domain is less efficient, but, allows arbitrary time varying sample rate conversion.
- efficient time domain interpolation to fixed grid using a polyphase filterbank and linear interpolation to obtain the final samples at arbitrary positions.

4.3 Transposition

- resampling does not change signal (besides eventually changing band limits)

Playing with original sample rate:

- changes pitch according to the ratio between the sample rates.
- changes duration according to the inverse ratio

Pro and Contra:

+ time varying transposition with very high time precision,
- time stretching needed to compensate change of duration,
- calculation time depends on transposition parameters.

4.4 Frequency domain transposition

Alternatively to time domain resampling the transposition can be obtained by means of shifting all spectral peaks to the new spectral position according to the transposition factor [LD99].

General idea explained using a single sinusoid signal

- signal with analysis window $h(n)$

$$x(n) = e^{j\Omega n+\phi}h(n) \tag{36}$$

- spectrum at position mI with hop size I (after removal of linear phase trend)

$$X(mI, w) = e^{j(mI+\frac{M-1}{2})\Omega+\phi}H(\Omega - w) \tag{37}$$

- moving the spectrum up in frequency by Δ_Ω

$$Y(mI, w) = X(mI, w - \Delta_\Omega) = e^{j(mI+\frac{M-1}{2})\Omega+\phi}H(\Omega + \Delta_\Omega - w) \tag{38}$$

- and performing inverse Fourier transform yields

$$y_m(n) = e^{jn(\Omega+\Delta_\Omega)+\phi-(mI+\frac{M-1}{2})\Delta_\Omega} h(n-mI) \tag{39}$$

- moving the spectrum creates a sinusoid that is shifted in frequency by the same amount, preserving the phase in the window center.
- to achieve coherent overlap add with fixed frequency shift Δ_Ω Δ_Ω an additional phase term has to be added to the spectrum as follows

$$Y(mI, w) = X(mI, w-\Delta_\Omega)e^{\Delta_\Omega(mI+\frac{M-1}{2})} \tag{40}$$

- if Δ_Ω varies over time the phase correction summand results from integrating over all precedent values

$$Y(mI, w) = X(mI, w-\Delta_\Omega)e^{\sum_m \Delta_\Omega(m)I+\Delta_\Omega(0)\frac{M-1}{2})} \tag{41}$$

Some properties:

+ because window duration is unchanged there is no need for duration compensation,
+ arbitrary peak displacements possible,
- transposition requires peak individual frequency shift according to original frequency,
- peak detection and individual treatment of each peak is required!
- additional frequency correction is required if time stretching is desired.

4.5 DFT and frequency domain transposition

- transposition in DFT spectra is more complicated because shifting by frequency offset that is not equal to an integer bin offset requires spectral interpolation.
- due to duality between frequency and time domain the interpolation technique described for time domain resampling in section **4.2** can be equivalently applied in the spectral domain
- suggested procedure for factor L spectral interpolation
- original spectrum and signal

$$x(n) = \frac{1}{N}\sum_{k=0}^{N} X(k)e^{j\frac{2\pi}{N}kn} \tag{42}$$

- interpolate periodic spectrum with zeros to achieve oversampling (expansion)

$$Y(k) = \sum_{k=-\infty}^{\infty} X(k)\delta(n-Lk) \tag{43}$$

- the DFT size and the time segment increases by factor L
- inverse signal

$$\begin{aligned} y(n) &= \frac{1}{LN}\sum_{k=0}^{LN} Y(k)e^{j\frac{2\pi}{LN}kn} && (44)\\ &= \frac{1}{LN}\sum_{k=0}^{LN}\sum_{u=-\infty}^{\infty} X(u)\delta(k-Lu)e^{j\frac{2\pi}{LN}kn} && (45)\\ &= \frac{1}{LN}\sum_{k=0}^{N} X(k)e^{j\frac{2\pi}{LN}Lkn} && (46)\\ &= \frac{1}{LN}\sum_{k=0}^{N} X(k)e^{j\frac{2\pi}{N}kn} && (47)\\ &= \frac{x(n)}{L} && (48)\end{aligned}$$

- signal changed by factor L and by the fact that one DFT now creates L periods of the periodic signal

- sinc and linear interpolation applied in the spectral domain will apply modulation in the time domain.
- sinc interpolation is ideal lowtime-pass in time domain corresponding to a rectangular window that cuts exactly the first period of the periodic signal
- the time modulation related to linear interpolation in th spectral domain is non constant in passtime region and has rather weak suppression for the following repetitions,
- because there will be an synthesis window applied during overlap add the only problem is the modulation of the passtime,
- modulation is lowered due to overlap add procedure,
- [LD99] found modulation side bands of -21dB for 50% overlap and -51dB for 75% overlap.
- if still audible a combination of fixed sinc interpolation and linear interpolation should be used.

5 Computational costs

Rough estimation of costs per fixed time interval

- take into account only the number of DFT per sample
- calculate cost factor F_c in relation to simple analysis/resynthesis with at least equivalent analysis and synthesis hop size I_a, I_s .

5.1 time stretching

time stretching, with stretch factor $\beta > 1$

- if synthesis I_s remains unchanged and I_a is reduced then for each output sample the same average amount of DFT operations is required

$$F_c = 1 \tag{49}$$

time compression, with stretch factor $\frac{1}{\beta}, \beta > 1$

- synthesis hop size I_s is reduced by β, I_a is unchanged to reduced to prevent frequency estimation error during frequency estimation.

$$F_c = \beta \tag{50}$$

5.2 transposition

transposing up, pitch factor $\beta > 1$ using time domain resampling

- requires time stretching by β
- transposition compresses by β such that for each output sample the cost is

$$F_c = \beta \tag{51}$$

transposing down, pitch factor $\frac{1}{\beta}$ with $\beta > 1$ using time domain resampling

- requires time compression with $\frac{1}{\beta}$
- transposition expands by β such that over all compression costs will be compensated and

$$F_c = 1 \tag{52}$$

for transposition in frequency domain both $F_c = 1$.

6 Appendix

6.1 Frequency domain filtering with time invariant filter

To investigate the result obtained for multiplying STFT $X(lI,k)$ with a stationary FIR filter transfer function $H(k)$ if the normalization function $C(n)$ is not constant we represent $C(n) = K + \epsilon(n)$ and assume $K \gg \epsilon(n)$.

- Starting with eq. (3) we explicitly perform the convolution

$$\begin{aligned}
y(n) &= \frac{\sum_{l=-\infty}^{\infty}(x(n)w(n-lI)) * h(n)}{C(n)} && (53)\\
&= \frac{\sum_{l=-\infty}^{\infty}\sum_{m=-\infty}^{\infty}(x(m)w(m-lI))h(n-m)}{C(n)} && (54)\\
&= \frac{\sum_{m=-\infty}^{\infty}(\sum_{l=-\infty}^{\infty} x(m)w(m-lI))h(n-m)}{C(n)} && (55)\\
&= \frac{\sum_{m=-\infty}^{\infty} x(n)(K+\epsilon(m))h(n-m)}{K+\epsilon(n)} && (56)
\end{aligned}$$

- with the first order approximation $\frac{1}{K+\epsilon(n)} \approx \frac{1-\frac{\epsilon(n)}{K}}{K}$ we obtain

$$\begin{aligned}
y(n) &\approx \frac{\sum_{m=-\infty}^{\infty} X(m)(K+\epsilon(m))h(n-m)}{K}(1-\frac{\epsilon(n)}{K}) && (57)\\
&= \frac{\sum_{m=-\infty}^{\infty} X(m)(K+\epsilon(m))h(n-m)}{K}(1-\frac{\epsilon(n)}{K}) && (58)\\
&= x(n) * h(n) && (59)\\
&\quad + \frac{\sum_{m=-\infty}^{\infty} x(m)\epsilon(m)h(n-m)}{K} && (60)\\
&\quad - (x(n) * h(n))\frac{\epsilon(n)}{K} && (61)\\
&\quad - \frac{\sum_{m=-\infty}^{\infty} x(m)\epsilon(m)h(n-m)}{K^2}\epsilon(n) && (62)
\end{aligned}$$

- which shows that to first order approximation the error due to non constant normalization function can be expressed in terms of modulations applied to the original signal and the output of the convolution.

6.2 Estimating the frequency in the phase vocoder

We want to derive eq. (18) to be able calculate the frequency of a sinusoid from the observed phase difference between 2 analysis frames in an STFT.

We first remember that for a stationary sinusoid with frequency Ω the STFT can be represented by means of a complex constant K and a frame dependend phase ϕ_l as follows

$$X(lI,k) = Ke^{j\phi_l} = Ke^{jlI\Omega} \tag{63}$$

For FFT size N and for each bin k we can represent the frequency of the sinusoid using 2 summands, the center frequency of bin k which is $\omega_k = \frac{2\pi}{N}k$ and a bin dependend frequency offset Θ_k

$$\Omega = \Theta_k + \omega_k = \Theta_k + \frac{2\pi}{N}k \tag{64}$$

For the phase difference between to consecutive frames we get

$$\Delta_\phi = \phi_{l+1} - \phi_l = I(\Theta_k + \omega_k) + 2\pi C, \tag{65}$$

where C is an integer constant that is due to the fact that the phase values are obtained as the principal values of the inverse tangent. Rearranging yields

$$I\Theta_k = \phi_{l+1} - \phi_l - I\omega_k - 2\pi C \tag{66}$$

The problem is the unknown constant C. It can be removed by means of taking the principle value on both sides of the equation. If we assume that $|I\Theta_k| < \pi$ and if $[]_{2\pi}$ denotes the reduction of the phase argument to its principle value (eq. (17)) we can proceed with

$$\begin{aligned}
[I\Theta_k]_{2\pi} &= [\phi_{l+1} - \phi_l - I\omega_k - 2\pi C]_{2\pi} && (67)\\
I\Theta_k &= [\phi_{l+1} - \phi_l - I\omega_k]_{2\pi} && (68)\\
\Theta_k &= \frac{[\phi_{l+1} - \phi_l - I\omega_k]_{2\pi}}{I} && (69)
\end{aligned}$$

Note, that eq. (69) remains valid as long as $|I\Theta_k| < \pi$, which means that the range of bins in the neighborhood of the sinusoidal frequency that can be used to estimate the frequency offset depends on I and decreases with increasing I. For an DFT of size N we get the offset in bins around the sinusoidal frequency for which a frequency estimate can be calculated to

$$|r| < \frac{\frac{\pi}{I}}{\frac{2\pi}{N}} = \frac{N}{2I} \tag{70}$$

For phase vocoder applications the frame offset I should sufficiently small to ensure that the frequency estimation for all bins of the mainlobe of the related spectral peak will be correct.

For the rectangular window of length M the spectral peak covers $r \approx \frac{N}{M}$ bins and therefore we conclude $I < \frac{M}{2}$. Similar for the Hanning and Hamming window there is $r \approx \frac{2N}{M}$ such that $I < \frac{M}{4}$

6.3 Calculating the signal mean time in the spectral domain

For the detection of transients we are looking for an efficient way to calculate the signal mean time of the center of gravity of the the signal $s(n)$ using its Fourier transform $S(\omega) = A(\omega)e^{j\phi(\omega)}$.

- According to [Coh95] we interpret

$$P_s(n) = \frac{|s(n)|^2}{\sum_n |s(n)|^2} \tag{71}$$

as time distribution and

$$P_S(\omega) = \frac{|S(\omega)|^2}{\int_{\pi}^{\pi} |S(\omega)|^2 d\omega} \tag{72}$$

as frequency distribution.

- Then we can define the *mean time* of the signal in agreement with the probabilistic average as

$$n_m = \sum_n n P_s(n) = \sum_n n \frac{|s(n)|^2}{\sum_n |s(n)|^2} \tag{73}$$

- from Parseval's theorem we have

$$\sum_n |s(n)|^2 = \frac{1}{2\pi} \int_{\pi}^{\pi} |S(\omega)|^2 d\omega = \frac{1}{2\pi} \int_{\pi}^{\pi} A(\omega)^2 d\omega \tag{74}$$

- moreover using the notation X^* to denote the complex conjugate of X and the fact that

$$s^*(n) \rightarrow S^*(-\omega), \tag{75}$$

- and the modulation theorem we have

$$s(n)s^*(n) = \frac{1}{2\pi} \int_{\pi}^{\pi} S(\Omega) S^*(\Omega - \omega) d\Omega. \tag{76}$$

- By means of the frequency differentiation theorem we have

$$\begin{aligned} ns(n)s^*(n) &= \frac{1}{2\pi}j\frac{\partial}{\partial\omega}\int_{\pi}^{\pi} S(\Omega)S^*(\Omega-\omega)d\Omega && (77)\\ &= \frac{1}{2\pi}j\int_{\pi}^{\pi} S(\Omega)\frac{\partial}{\partial\omega}S^*(\Omega-\omega)d\Omega && (78)\\ & && (79)\end{aligned}$$

- from the FT definition and using the shortcut $X'(a) = \frac{\partial}{\partial a}X(a)$ we conclude

$$\begin{aligned} \sum_n ns(n)^2 &= \frac{1}{2\pi}j\int_{\pi}^{\pi} S(\Omega)\frac{\partial}{\partial\omega}S^*(\Omega-\omega)d\Omega\Big|_{\omega=0} && (80)\\ &= -\frac{1}{2\pi}j\int_{\pi}^{\pi} S(\Omega)S'^*(\Omega)d\Omega && (81)\\ &= -\frac{1}{2\pi}j\int_{\pi}^{\pi} A(\Omega)e^{j\phi(\Omega)}\frac{\partial}{\partial\Omega}(A(\Omega)e^{-j\phi(\Omega)})d\Omega && (82)\end{aligned}$$

$$= \quad -\frac{1}{2\pi}j\int_{\pi}^{\pi} A(\Omega)e^{j\phi(\Omega)}(A'(\Omega) - j\phi'(\Omega)A(\Omega))e^{-j\phi(\Omega)}d\Omega \tag{83}$$

$$= \quad \frac{1}{2\pi}\int_{\pi}^{\pi} -jA(\Omega)A'(\Omega) - \phi'(\Omega)A(\Omega)^2 d\Omega \tag{84}$$

- Because the result is by construction real we conclude

$$\int_{\pi}^{\pi} A(\Omega)A'(\Omega)d\Omega = 0 \tag{85}$$

- such that

$$\sum_n n|s(n)|^2 = \frac{1}{2\pi}\int_{\pi}^{\pi} -\phi'(\Omega)A(\Omega)^2 d\Omega \tag{86}$$

- and finally

$$n_m = \frac{\int_{\pi}^{\pi} -\phi'(\omega)A(\omega)^2 d\omega}{\int_{\pi}^{\pi} A(\omega)^2 d\omega} \tag{87}$$

- the quantity -$\phi'(\omega)$ is called the *group delay*
- taking the part of the signal that is confined to an infinitesimal small band at frequency ω of size Δ_ω and calculating its **mean-time** yields

$$
\begin{aligned}
n_m &= \frac{\int_{\omega}^{\omega+\Delta\omega} -\phi'(\omega) A(\omega)^2 d\omega}{\int_{\omega}^{\omega+\Delta\omega} A(\omega)^2 d\omega} && (88)\\
&\approx -\phi'(\omega) \frac{\int_{\omega}^{\omega+\Delta\omega} A(\omega)^2 d\omega}{\int_{\omega}^{\omega+\Delta\omega} A(\omega)^2 d\omega} && (89)\\
&= -\phi'(\omega) && (90)
\end{aligned}
$$

- the *group delay* describes the contribution of frequency ω to the signal **mean-time**.

Efficient calculation of the group delay

Following recent results of [AF95] the phase derivative with respect to frequency can be efficiently calculated by means of a DFT using a a modified analysis window.

- We are looking for en expression to calculate the group delay

$$t_g(\omega) = \frac{\partial\phi(\omega)}{\partial\omega} \tag{91}$$

- We start with the observation that the phase spectrum is the imaginary part of the logarithm of the spectrum

$$\phi(\omega) = \Im(\log(X(\omega))) = \Im(\log(A(\omega)\exp(i\phi(\omega)))) = \Im(\log(A(\omega)) + i\phi(\omega)) \tag{92}$$

- Taking the imaginary part is a linear operation such that we may apply the derivative to the log expression

$$\begin{aligned} \frac{\partial\phi(\omega)}{\partial\omega} &= \frac{\partial}{\partial\omega}\Im(\log(X(\omega))) \quad (93)\\ &= \Im(\frac{\partial}{\partial\omega}\log(X(\omega))) \quad (94) \end{aligned}$$

$$= \Im(\frac{\frac{\partial}{\partial\omega}X(\omega)}{X(\omega)}) \tag{95}$$

- from the DFT frequency differentiation theorem and assuming the use of an analysis window $h(n)$ that is centered around the origin we get

$$\frac{\partial}{\partial\omega} = \Im(\frac{\sum_{n=-\infty}^{\infty} -inh(n)x(n)\exp(-iwn)}{X(\omega)}) \tag{96}$$

$$= -\Re(\frac{X_{ht}(w)}{X_h(\omega)}) = -\Re(\frac{X_{ht}(\omega)\overline{X_h(\omega)}}{|X_h(\omega)|^2}) \tag{97}$$

where the overline denotes complex conjugation and $X_{ht}(\omega)$ is the Fourier transform using the window $nh(n)$.

- we conclude that the phase derivative with respect to frequency can be calculated by means of a DFT using as analysis window

Relation to energy change based methods

Group delay and **normalized derivative of energy** are closely related:

$$\frac{\frac{\partial |X_h(\omega,n_0)|^2}{\partial n_0}}{|X()|^2} = \frac{1}{|X()|^2}\frac{\partial}{\partial n_0}(\overline{\sum_n x(n)h(n-n_0)e^{-j\omega n}}\sum_n x(n)h(n-n_0)e^{-j\omega n}) \tag{98}$$

$$= (\frac{X(\omega,n_0)}{|X()|^2}\overline{\sum_n x(n)\frac{\partial h(n-n_0)}{\partial n_0}e^{-j\omega n}} \tag{99}$$

$$+\frac{\overline{X(\omega,n_0)}}{|X()|^2}\sum_n x(n)\frac{\partial h(n-n_0)}{\partial n_0}e^{-j\omega n} \tag{100}$$

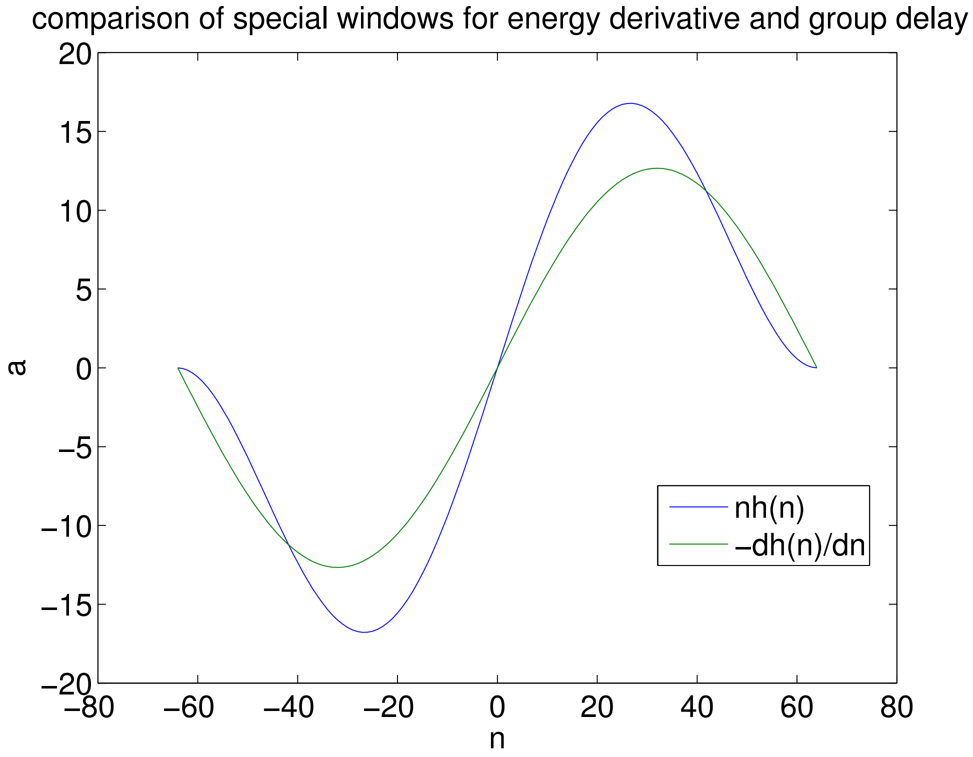
$$= (\overline{\frac{\overline{X(\omega,n_0)}}{|X()|^2}\sum_n x(n)\frac{\partial h(n-n_0)}{\partial n_0}e^{-j\omega n}} \tag{101}$$

$$+\frac{\overline{X(\omega,n_0)}}{|X()|^2}\sum_n x(n)\frac{\partial h(n-n_0)}{\partial n_0}e^{-j\omega n} \tag{102}$$

$$= \quad -2\Re(\frac{\overline{X(\omega,n_0)}}{|X()|^2}\sum_n x(n)\frac{\partial h(n-n_0)}{\partial n}e^{-j\omega n})$$

$$t_g(\omega,n_0) \quad = \quad -\Re(\frac{\overline{X(\omega,n_0)}}{|X()|^2}\sum_n x(n)h(n-n_0)ne^{-j\omega n}) \tag{103}$$

- The difference is replacement of the **derivative of the window with respect to time** by a **multiplication between time and window**
- Besides different scaling the qualitative behavior of both measures is similar.



6.4 Resampling in the frequency domain

- Suppose we have a given time continuous signal $x(t)$ with continuous Fourier transform $X_c(\omega)$ and band limits such that $X_c(\omega) == 0$ for $\omega > \omega_l$. Then we can generate the continuous signal from

$$x(t) = \int_{-\omega_l}^{\omega_l} X_c(\omega) e^{j\omega t} d\omega \tag{104}$$

- after sampling of the signal with sample rate $\Omega > 2\omega_l$ we obtain the discrete time signal $x_d(n)$ with discrete Fourier transform $X(\omega)$ which we may use to generate the discrete time signal

$$x_d(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega) e^{j\omega n} d\omega \tag{105}$$

- letting n take all real values shows that the continuous version of $x(n)$ is just a time scaled version with

$$x_d(n) = x(\frac{n}{\Omega}) \tag{106}$$

- all possible sample rates can be obtained from the spectral representation by simply rescaling the spectrum (for downsampling a low pass filter is needed to prevent aliasing).

- The same relation holds true for the DFT. Assuming $X(\pi) == 0$ we can generate the discrete time signal as well by

$$x_d(n) = \frac{1}{N} \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}-1} X(\frac{2\pi}{N}k)e^{j\frac{2\pi}{N}kn} \tag{107}$$

- zero padding the spectrum to use a new DFT size N' yields

$$Y(k) = \begin{cases} X(k) & \text{for } |k| < \frac{N}{2} \\ 0 & \text{else} \end{cases} \tag{108}$$

and the related signal is

$$y_d(n) \quad = \quad \frac{1}{N'} \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}-1} Y(\frac{2\pi}{N}k)e^{j\frac{2\pi}{N'}kn} \tag{109}$$

$$= \frac{1}{N'} \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}-1} X(\frac{2\pi}{N}k) e^{j\frac{2\pi}{N}k(n\frac{N}{N'})} \tag{110}$$

$$= x(\frac{nN}{N'\Omega}) \tag{111}$$

References

[AF95] F. Auger and P. Flandrin. Improving the readability of time-frequency and time-scale representations by the reassignment method. *IEEE Trans. on Signal Processing*, 43(5):1068–1089, 1995. 29, 64

[Coh95] L. Cohen. *Time-frequency analysis*. Signal Processing Series. Prentice Hall, 1995. 27, 60

[DL99] M. Dolson and J. Laroche. Improved phase vocoder time-scale modification of audio. *IEEE Transactions on Speech and Audio Processing*, 7(3):323–332, 1999. 18

[LD99] J. Laroche and M. Dolson. New phase-vocoder techniques for real-time pitch shifting, chorusing, harmonizing and other exotic audio modifications. *Journal of the AES*, 47(11):928–936, 1999. 44, 49

[Röb06] A. Röbel. Analysis, modelling and transformation of audio signals - Part II: Analysis/resynthesis with the short time fourier transform. lecture slides, 2006. AMT : Part II. 6, 8