2001-01-14
dpwe@ee.columbia.edu
SAPR – DSP ①

- wavesurfer link
- matlab specgram
- cumix website link

Digital Signal Processing (review)

Sound : 1 D signal – pressure as a function of time $p(t)$

- represent in computer as discrete-time samples $p[n]$ $n = 0, 1, 2 \ldots$

DT sigs

$t = n\,T$ – sampling interval

→ samp. freq. $f_s = \frac{1}{T}$

(8kHz .. 48kHz)

- computer → finite precision ⇒ $pd[n]$ is quantized

e.g. 16 bit linear $pd[n] = k[-3278 .. 32767]$

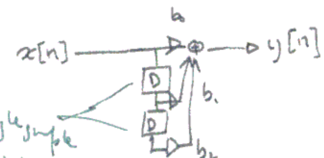
non-integer part thrown away ⇒ Quantization noise.

Signal processing : signals + systems

Sys

Most useful class of systems = LTI / LCCDE

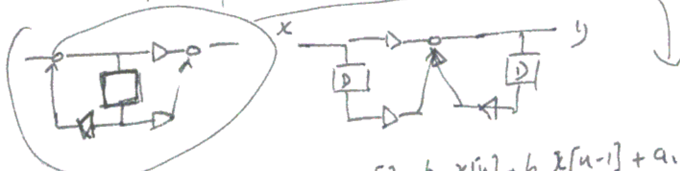
e.g. those as block diagrams:



Single sample delays

→ input/output relationship: $y[n] = b_0 \cdot x[n] + b_1 x[n-1] + b_2\, x[n-2]$

+ feedback



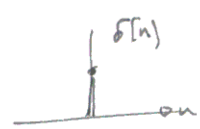
$y[n] = b_0\, x[n] + b_1 x[n-1] + a_1 y[n-1]$

hence, LTI

discrete dont change

$x = \alpha x_1 + \beta x_2 \Rightarrow y = \alpha y_1 + \beta y_2$

⇒ impulse response $\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & \text{ow} \end{cases}$



$x[n] = \delta[n] \to y[n] = h[n]$ "impulse response"

but any $x[n] = \sum_m x_m \delta[n-m]$

Conv

$\Rightarrow y[n] = \sum_m x_m h[n-m]$ → CONVOLUTION $y[n] = x[n] * h[n]$

1. ~~wtd~~ h's
2. inner prods of x

⇒ $h[n]$ equivalent descr. of LTI system.

~~examples~~ – moving average – effect on trend + periodic → MATLAB

– simple feedback – INFINITE IR

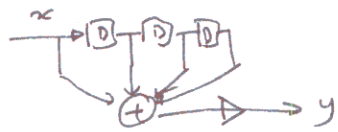
→ Convolution w/ so helpful

– 2nd order feedback → SINUSOID $1 - 2r\cos\theta z^{-1} + r^2 z^{-2}$

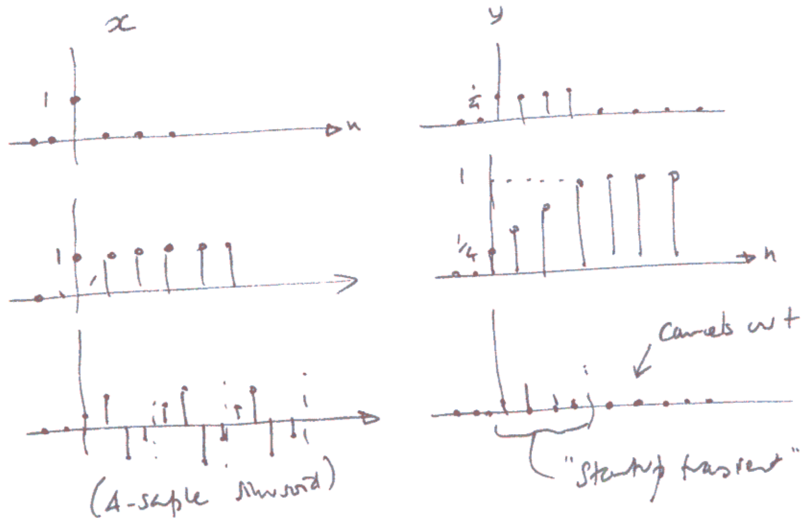
2001-01-16
dpwe@ee.columbia.edu
SAPR - DSP ②

basic systems - examples

"Moving average" filter



$$y[n] = \tfrac{1}{4}\left(x[n] + x[n-1] + x[n-2] + x[n-3]\right)$$
$$= \left[\tfrac{1}{4}\ \tfrac{1}{4}\ \tfrac{1}{4}\ \tfrac{1}{4}\right] * x[n]$$



$\Rightarrow$ trend + period 4 sinusoid

→ LINEAR response

$x = x_1 + x_2$

$\Rightarrow y = y_1 + y_2$

$\Rightarrow$ effect of sinusoid REMOVES

→ matlab

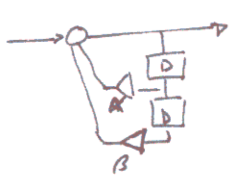
Feedback filter

$$y[n] = x[n] + \alpha\, y[n-1]$$

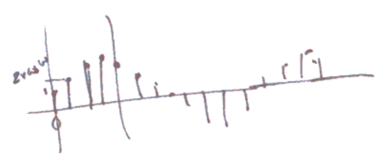
$\alpha > 1$ → unstable

infinite IR
(can't represent h[n] explicitly)

2^{nd} order feedback



→ get oscillation
(damped)
(any freq)



stable / unstable.

$\alpha = 2r\cos\omega_0;\ \beta = -r^2$

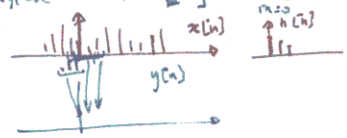
2001-01-21
dpwe @ ee.columbia.edu
SAPR - DSP ③

Convolution or Correlation:

- passing a signal through a system results in convolution of input and IR

$$y[n] = \underbrace{\sum_{m=0}^{N-1} h[m]\, x[n-m]}_{\text{indices } \Sigma \text{ to } n}$$ — weighted sum of delayed x's
— inner product between shifted x & $h[-n]$ $\quad \sum_{m=-(N-1)}^{0} h[-m]\, x[n+m]$

$= \text{“} h[n] * x[n] \text{”}$
$= x[n] * h[n]$ → prove! HW



Correlation is explicitly inner product:

$$r_{xy}[l] = \sum x[n]\, y[n+l]$$
lag

Corr

- peak in r_{xy} at l_0 ⇒ $x[n]$ “resembles” $y[n+l]$ (i.e. y is x delayed by l)
  (⇒ matched filter)

- order matters: $r_{xy}[l] = r_{yx}[-l]$
- relates to convo: $r_{xy}[l] = x[l] * y[-l]$

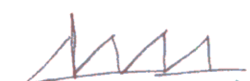
autocorrelation: $r_{xx}[l]$ finds shifts at which x resembles self i.e. PERIODICITY.

ML example: - picking out a pulse in noise
- periodicity of voice (ST window)

SOLA

FOURIER (frequency) domain:

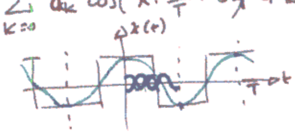
Fourier series : periodic $x(t) = x(t+T)$

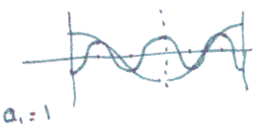


larger → closer approx; truncation is best.

$$\Rightarrow x(t) \approx \sum_{k=0}^{n} a_k \cos\left(k \cdot \frac{2\pi}{T} \cdot t + \phi_k\right)$$ ← harmonics

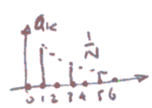
FS





$a_1 = 1$
$a_3 = -\frac{1}{3}$
$a_5 = \frac{1}{5}$
$a_7 = -\frac{1}{7}$ ...

→ fourier domain



complex sinusoids → $x(t) \triangleq \sum_{k=-\infty}^{\infty} c_k e^{j\frac{2\pi k t}{T}}$ complex ; $c_k = c_{-k}^*$ for real x
$c_k = a_k e^{j\phi_k}$ (?)

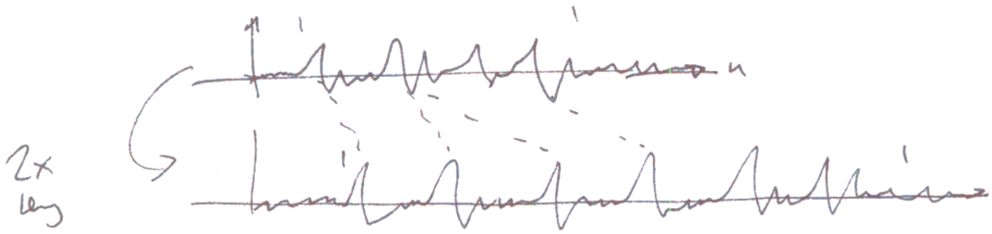
⇒ fourier analysis $\quad \underbrace{c_k = \frac{1}{2\pi}\int_{-\frac{T}{2}}^{\frac{T}{2}} x(t)\, e^{-j\frac{2\pi k t}{T}}\, dt}_{\text{convolution / matched filter / inner product}}$

— works because sinusoids are ORTHOGONAL SET

$$x(t) = \sum c_k e^{jk\omega_0 t}$$
$$\Rightarrow \int x(t)\, e^{-jk_0\omega_0 t}\, dt = \int \sum_k c_k e^{j(k-k_0)\omega_0 t}\, dt = 0 \text{ if } k \neq k_0$$

SOLA - Synchronous overlap Add

- timescale mod by duplicating fragments of signal



duplicate cycles?

but hard to choose cycles.
arbitrary → phase cancellation
→ choose best window (within limits) by
max cross-correlation.

$$y'[rL+m] = \underbrace{\beta[m]\cdot y[rL+m]}_{\text{recursive ..}} + (1-\beta[m])\cdot x[\lfloor q\cdot rL\rfloor + m + k_m]$$

$$k_m : \max_{0\le k\le k_{max}} \frac{\sum_{m=0}^{N_{ov}} x[q\cdot rL+m+k]\cdot y[rL+m]}{\sqrt{\sum x^2[q\cdot rL+m+k]\cdot \sum y^2[rL+m]}}$$

overlap region

best alignment within window.

2001-01-21
dpwe@ee.columbia.edu
SAPR – DSP ④

FT

Fourier transform – any $x(t)$ (not just periodic)
– continuous frq Ω

FT
$$\boxed{\begin{aligned} X(j\Omega) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} x(t)\, e^{-j\Omega t}\, dt \\ x(t) &= \int_{-\infty}^{\infty} X(j\Omega)\, e^{j\Omega t}\, dt \end{aligned}}$$

→ spectrum 

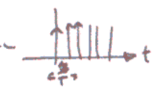
(reduces to $\sum c_k \delta(\Omega - k\Omega_0)$ if $x(t)$ is periodic)

DTFT

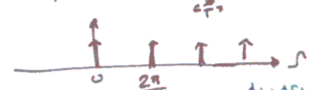
Discrete time → treat $x[n]$ as $x(t) = \sum_n x[n]\,\delta(t - nT)$ IMPULSE TRAIN

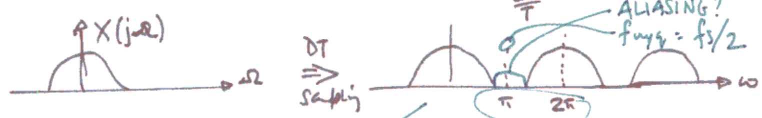
$$\rightarrow \boxed{\text{DTFT}\left\{\begin{aligned} X(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n} \\ x[n] &= \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})\, e^{j\omega n}\, d\omega \end{aligned}\right.}$$

Spectrum is periodic "$e^{j\omega}$" ⇒ $X(e^{j\omega}) = X(e^{j(\omega + 2r\pi)})$

- impulse train: multiplication in time domain by sample train 

– modulation ?

≡ convolution in frq domain by 

i.e. 

filter to recover?

└ NORMALIZED FRQ (rad/samp)
(need to know T
to map $\omega = 2\pi \rightarrow \Omega = \frac{2\pi}{T}$)

Examples – a little hard ∵ computer always ~~DTFT~~ discrete – dense sampling → approx continuous t... ?

- Fourier series – build up square wave
- pulse (2nd order system) → large DTFT
  - sample → duplicated spectrum (periodic) → aliasing.
    ↳ if frq too high / ~~spacing~~ sampling too coarse

FT properties
- Linear
- time shift $g[n - n_0] \leftrightarrow e^{-j\omega n_0} G(e^{j\omega})$ (from $X(e^{j\omega}) = \sum_{\forall n} x[n]\, e^{-j\omega n}$)
- frq shift $e^{j\omega_0 n} g[n] \leftrightarrow G(e^{j(\omega - \omega_0)})$ eg. modulated onto a carrier (AM)
- parseval! $\sum_{\forall n} g[n]\, h^*[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} G(e^{j\omega})\, H^*(e^{j\omega})\, d\omega$

  e.g. $gg^* = |g|^2$

2001-01-21
dpwe@ee.columbia.edu
SAPR - DSP (5)

Z - transform - mathematical convenience for DT analysis

Define $X(z) = \sum_{\forall n} x[n]\, z^{-n}$ → represent $x[n]$ as polynomial in dummy complex z

ZT

Consider $G(z) \cdot H(z)$: polynomial multiplication: groups pairs where exponents add to a constant

$g[n] = \{g_0\ g_1\ g_2\ \ldots\} \ldots$

e.g. $(g_0 + g_1 z^{-1} + g_2 z^{-2}) \cdot (h_0 + h_1 z^{-1} + h_2 z^{-2})$

? $= g_0 h_0 + z^{-1}(g_1 h_0 + g_0 h_1) + z^{-2}(g_0 h_2 + g_1 h_1 + g_2 h_0) + z^{-3}(g_1 h_2 + g_2 h_1) + z^{-4} g_2 h_2$

conv

= CONVOLUTION! - subscripts add to n c/w $y[n] = \sum_{\forall m} g[m]\, h[n-m]$

$n = m + n-m$

ie. $g[n] \rightarrow G(z)$

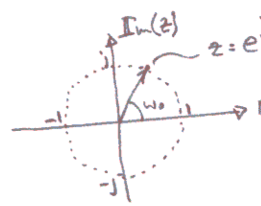
? $g[n] * h[n] \rightarrow G(z) \cdot H(z)$

c/w DTFT if $z = e^{j\omega}$ ⇒ $G(z) = G(e^{j\omega}) \triangleq \sum_{\forall n} g[n]\, e^{-j\omega n}$ = DTFT

⇒ EVALUATING z transform on unit circle $(z = e^{j\omega}, \omega = -\pi..\pi)$ = freq. resp DTFT

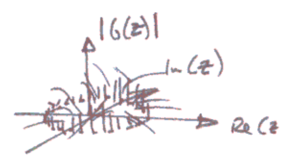
DTFT

→ z-plane (complex)



$z = e^{j\omega_0} = \cos\omega_0 + j \sin\omega_0$

→ can consider $G(z)$ as a surface (but complex)



+ DTFT as cylindrical slice on unit circle

ZT convolution property ⇒ $y[n] = x[n] * h[n]$ ⇔ $Y(e^{j\omega}) = X(e^{j\omega}) \cdot H(e^{j\omega})$ → FILTERING ($H(e^{j\omega})$ selects part of spectrum)

c/w system function if have LCCDE system

from LCCDE

$y[n] = (-a_1)\, y[n-1] + (-a_2)\, y[n-2]$
$+ b_0 x[n] + b_1 x[n-1] + b_2 x[n-2]$

hence

Z transform:

$Z\{x[n-1]\} = z^{-1} x[n]$

⇒ $\sum_{j=0}^{M} a_j\, y[n-j] = \sum_{k=0}^{N} b_j\, x[n-j]$ $(a_0 \triangleq 1)$

ZT ⇒ $\sum_j a_j Y(z) z^{-j} = \sum_k b_k X(z) z^{-k}$

⇒ $\frac{Y(z)}{X(z)} = H(z) = \frac{\sum b_k z^{-k}}{\sum a_j z^{-j}} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$

Rational polynomial z transform.

ZT of IR ∵ $H(z) \cdot X(z) = Y(z)$
$h[n] * x[n] = y[n]$

2001-01-21
dpwe@ee.columbia.edu
SAPR - DSP ⑥

$H(z)$ system function: rational polynomial in z, easily derived from actual system

$\rightarrow H(e^{j\omega})$, frequency response - evaluate polynomial on u. circle

but the points on z plane of interest

$H(z) = 0$: zeros (roots of $\sum b_k z^{-k}$)

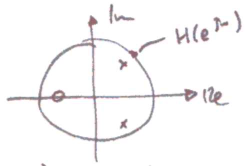
$H(z) = \frac{1}{0} = \infty$: poles (roots of $\sum a_j z^{-j}$)

$H(z) = \frac{b_0 \prod_k (1 - \xi_k z^{-k})}{\prod_j (1 - \lambda_j z^{-1})}$

↑ roots

Poles / zeros

$\rightarrow$ z plane :



$\rightarrow |H(z)|$:

Matlab.

$\Rightarrow$ as $H(e^{j\omega})$ gets close to a pole, it gets large

zero, it gets small.

ie. $H(e^{j\omega}) = \frac{b_0(1 - \xi_1 z^{-1})(1 - \xi_2 z^{-1})}{(1 - \lambda_1 z^{-1})(1 - \lambda_2 z^{-1})}$ term dominates as $\lambda_2 z^{-1} \rightarrow 1$

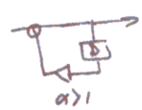
(distance $(z - \lambda_2)$ (mult top & bottom by z^n))

eg.. $y[n] = 2r\cos\omega_0\, y[n-1] + r^2 y[n-2] + x[n]$

$\Rightarrow H(z) \Rightarrow \lambda_i \Rightarrow$ z-plane $\Rightarrow$ freq. resp.

(add a zero)

STABILITY · can make unstable filters :



Stability

VALIDITY of z transform $\sum x[n] z^{-n}$

assumes convergence

ie. for "causal" ($n > 0$)

$\Rightarrow |z| > r_{\text{some}}$ will ensure convergence

- critical radius defined by poles of $H(z)$

- stable filter $\Rightarrow$ converges for $H(e^{j\omega})$

$\Rightarrow$ u. circle in ROC

$\boxed{\Rightarrow \text{all poles inside u. circle}}$

(anticausal would be the reverse

$\rightarrow$ run unstable filters "backwards" in time?)

$\rightarrow$ DFT
(FFT)
~~STFT~~
circconv
OLA
STFT
proc.

2001-01-21
dpwe@ee.columbia.edu
SAPR - DSP (7)

DFT

periodic $x(t)$ → Fourier Series $\sum_k c_k e^{j\frac{2\pi k}{T}t}$

↳ any $x(t)$ → Fourier transform $x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(j\Omega)e^{j\Omega t}\,d\Omega$

↳ discrete $x[n]$ → DTFT $x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})e^{j\omega n}\,d\omega$

↳ periodic $x[n]$ → $\boxed{\text{DFT}}$ . . .

DFT

(i.e. unique values only for $n = 0..N-1$ (N deg. of freedom)

→ can represent with just N transform pts

$$\boxed{\Rightarrow X[k] = \sum_{n=0}^{N-1} x[n]\, \underbrace{e^{-j\frac{2\pi}{N}\cdot k\cdot n}}_{W_N^{nk}}}$$

$W_N = e^{-j\frac{2\pi}{N}}$

$$[X[k]] = \left[W_N^{nk} \right] \begin{bmatrix} x[0] \\ x[n] \\ x[N-1] \end{bmatrix}$$

matrix mul. — same shape I/O

"sample" exact values of $X(e^{j\omega})$ for • $x[n]$ zero outside $0..N-1$ • $\omega = \frac{2\pi k}{N}$

or weights of $\delta(\omega - \frac{2\pi k}{N})$ for DTFT of (unbounded) periodic $x[n]$

IDFT $x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, W_N^{-nk}$

→ can actually calculate on a computer!
(and fast - FFT!)

DFT & Circular Convolution

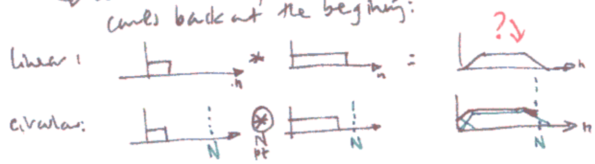
∵ $x[n]$, $X[k]$ imply periodic signal, multiplying DFTs corresponds to "circular convolution"

c/w linear convolution: $G(e^{j\omega})\cdot H(e^{j\omega}) \longleftrightarrow \underbrace{\sum_m g[m]\,h[n-m]}_{L+M-1 \text{ pts}}$

Circ Conv.

$G[k]\cdot H[k] \longleftrightarrow \underbrace{\sum_{m=0}^{N-1} g[m]\,h[\langle n-m\rangle_N]}_{\text{just } N \text{ pts, like inputs}}$

↳ whatever "spills off end" comes back at the beginning:



⇒ must zero pad, ensure $N \geq L+M-1$, when using DFT for convolution.

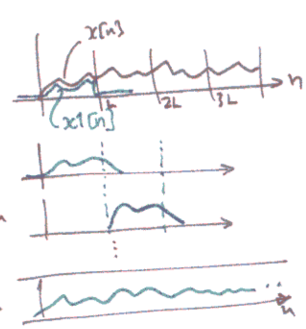
Overlap-Add

OLA

by linearity of convolution, if $x[n] = x_1[n] + x_2[n] + x_3[n] + \cdots$

then $x[n]*h[n] = x_1[n]*h[n] + x_2[n]*h[n] + \cdots$

e.g. break into short-time chunks



⇒ can calculate one long convolution by set of shorter ones ("block convolution")

⇒ if $h[n]$ is short (M pt), can do each block by DFT if zero pad : $N \geq L+M-1$

⇒ use FFT → fast → "BLOCK CONVOLUTION" (practical)

PrFT

Short-time Fourier Transform

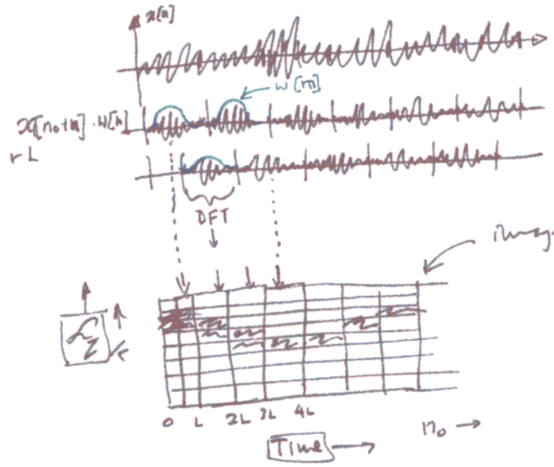
- full-up DTFT gives spectrum over all time
- time varying spectral content?
- break into pieces local to some time
  "short-time" segments
  FT → spectral content "at that time" = STFT

STFT (DFT-based)
$$\underbrace{X[k, n_0]}_{=rL} = \sum_{m=0}^{N-1} x[n_0+m] \cdot w[m] \cdot e^{-j\frac{2\pi k}{N} n}$$
(or $\forall n$)

fn of fq and time

$w[m]$ time-selection window
- zero for $n<0$, $n \geq N$
- tapered? to avoid spectral ghosts?

e.g.



overlap compensates for tapered window.

image of |DFT| or log $|X[k, n_0]|$ (dB)

why useful? what about ϕ - fq selective - relative ϕ insensitive
? - EAR

→ ISTFT...

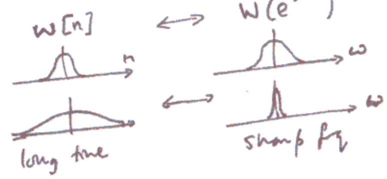
Windows

Effect of window :

$X[\omega, n_0] = \text{DFT}\{x[n_0+m] \cdot w[m]\} = [e^{+j\omega n_0} X(e^{j\omega})] \circledast W(e^{j\omega})$

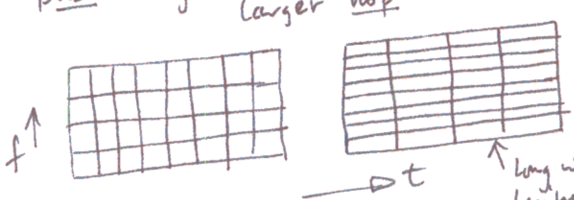
i.e. product
is smoothed/blurred by window DTFT

→ T-F trade-off :



plus long time → larger hop

→ alternate tilings



→ diff choices.

2001-01-22
dpwe@ee.columbia.edu
SAPR - DSP (9)

Phase Vocoder

- "stretch" a sound by "stretching" the spectrogram
  - time scale change w/o frq/spec change
  - (frq change w/o time scale)
  - (resampling = dilate t-f in inverse)
- easy to do by image - but doesn't specify phase, just |STFT|
  → need phase to reconstruct ...
  - duplicate ϕ? → just duplicating little segments - ϕ cancellation
  - avoid ϕ cancellation: maintain ϕ alignment
    → preserve $\dot\phi$ - relatively constant → can interpolate.
- procedure : . analyze to $|X|$, $\dot\theta(t,f)$
  - t scale : just interpolate
    f scale : scale $\dot\theta$
    & then IDFT & overlap-add
    (overlaps ; rewindow to avoid edges)

Flan + Golden:
reduce comms b/w i.e. speedup then slowdown
or use ½ b/w

or transmit $|X[k,n]|$, $\dot\theta[k,n]$ subsampled

what is intrinsic b/w? - analytically hard
- limited by articulators? (somewhat)

signal → STFT → $\left|X[k,rL]\right|$, $\dot\theta_X[k,rL]$

→ interpolate eg by time factor q

$$\Rightarrow \left|Y[k,rL]\right| \approx \left|X[k,q\cdot rL]\right|$$
$$= (1-\alpha)\left|X[k,\rho L]\right| + \alpha\left|X[k,(\rho+1)L]\right|$$
$$\left.\begin{aligned}\rho &= \lfloor q\cdot r\rfloor\\ \alpha &= q\cdot r-\rho\end{aligned}\right\}\text{ linear interp}$$

$$\dot\theta_Y[k,rL] = \ldots$$

$$\rightarrow Y[k,rL] = \left|Y[k,rL]\right| \exp j\left[\theta_Y[k,(r-1)L] + L\cdot\dot\theta_Y[k,rL]\right]$$

$$\rightarrow y[rL+m] = \sum_{k=0}^{N-1} Y[k,rL]\cdot e^{j\frac{2\pi km}{N}}$$

OLA $\left(+ \sum Y[k,(r-1)L]\, e^{j\frac{2\pi k(m+L)}{N}}\right)$