2021. 2. 25 Residual (frame) signal encoding

<Encoder>

$x[t]$ → (A) short-time framing → $x[t,k]$ → ····· → $C_{LSF}[k]$ → $LSF_{coded}[1...M, k] \triangleq Lc[k] \in \mathbb{R}^M$

$x[t,k]$: python과의 호환성을 위해 time, frame index로 순서변경

$C_{LSF}[k]$: 1×1 int, k번째 LSF code

$a_0^{(c)}, a_1^{(c)}, \ldots, a_M^{(c)}$: coded LSF로부터 얻은 LPC 계수 $\triangleq Ac[k] \in \mathbb{R}^{M+1}$

↓ pre-em. noise added

$x_{pn}[t,k]$에 FIR filter $Ac[k]$를 convolution하면 residual $e[n]$을 얻을 수 있음

$\Rightarrow x[t,k]$ < $C_{LSF}[k]$: envelope
$e[t] \triangleq x_{pn} * Ac : \mathbb{R}^{Ns}$

< Residual encoding >

$e[t]$ / $x[t,k]$

(R1) gain estimation: $e[t]$의 크기가 1이 되도록 normalize

$$g_k = Std(e[t,k]) = \sqrt{\frac{1}{Ns}\sum_{t=1}^{Ns} e^2[t,k]}$$

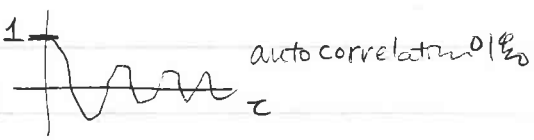
$$e_s[t,k] = e[t,k] / g_k$$

(R2) V/UV decision → UV → (R3) $\hat{e}_s[t,k] = N(0,1)$ 평균 0, variance 1인 random noise

voiced/unvoiced

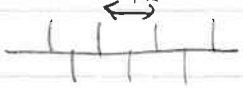
↓ Voiced

(R4) pitch estimation → P_k



auto correlation이용

(R5) pulse train generation

$\hat{e}_s[t,k]$ = 

(R6) $\hat{e}[t,k] = g_k \cdot \hat{e}_s[t,k]$

* Residual signal에서는 median smoothing filter가 필요

(R1) gain estimation

gain은 의미함. 마이크를 통해 녹음되는 frame

음성은 보통 다음과 같이 모델링된다

$$X(w) = \frac{G \cdot E(w)}{A(w)}$$

$G \in R^1$: scalar 실수. gain

$E(w)$: excitation / residual

$A(w)$: all-pole envelope

short-time framing을 했기 때문에 frame energy의 square-root를 구하면 된다 (즉, standard deviation)

$$g_k = std(e[t,k]) \approx \sqrt{\frac{1}{N_s} \sum e^2[t,k]}$$

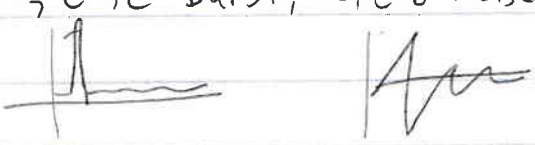
(A) (B)

(A) unbiased estimation이 ($N_s - 1$로 나누기)
통계적으로 의미있는 frame gain을 구할 수 있지만 N_s는 고정되어 있고, gain은 reconstruction에서 다시 곱해주기 때문에 굳이 $N_s - 1$로 나눌 필요는 없다.
(정확한 frame gain이 필요할 때 사용
codec에서는 영향 없음)

(B) 정확한 standard deviation 계산을 위해서는
$(e[t,k] - u_k)^2$, $u_k = \frac{1}{N} \sum e[t,k]$, 가 맞지만
dc 성분이라고 함

preemphasis에서 0-frequency (dc) 성분은 제거가 되고 그 이후의 과정에서 dc값이 바뀌는 연산이 없기 때문에 굳이 평균 차감은 필요없다.

* 극단적인 burst, 비선형 noise가 유입된 경우



문제가 될 수도 있기 때문에 평균 차감을 하는 것이 안전하기는 하다.

R2 Voiced / unvoiced decision (무성음)

- residual (excitation) 신호의 모델링 방법은 voiced (유성음) / unvoiced 에 따라 크게 달라지기 때문에 매우 중요하다.
- 정확히 추정하기는 매우 어렵다 (보조 강의자료에 model-based 방법 있음)
- 다음의 방법을 우선 적용해 보고, 잘 안되면 다른 방법들 시도한다.

① Voiced sound의 특징

power가 -6dB/octave (주파수가 2배가 되면 크기 감소)

= energy가 -12dB/octave

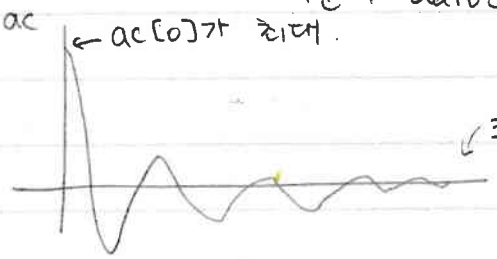
$ac[\tau] = x(t) \cdot x[t-\tau]$: 간격이 τ인 autocorrelation coef ($\tau = 0, \ldots, N_s$)

$\approx \sum_{t=\tau}^{N_s} x[t] \cdot x[t-\tau]$: short-time frame 범위를 벗어나면 $x[t]=0$ 이므로 $t=\tau$부터 시작

둘다 OK

$\approx \frac{1}{N_s} \sum_{t=\tau}^{N_s} x[t] \cdot x[t-\tau]$

↑ normalized denominator (분모)는 N_s 그대로, 즉 τ가 커질수록 작아짐 원래 infinite series에 대해서는 normalize 불가능하기 때문에 autocorrelation 자체에서는 sample 수로 나누는 것 없음



② unvoiced sound : 주기성이 없으므로 이론적으로 $ac[0] = ac[1] = \cdots$

③ decision algorithm

framing 한 신호

- Compute $ac[0], ac[1], \ldots ac[N_s]$ from $x[t,k]$
- if $ac[0] < \varepsilon$ → return unvoiced
  elif $ac[1]/ac[0] > \theta_{VUV}$ → return UV
  else return voiced

* silence로 따로 분류하는 알고리즘도 있지만 어차피 unvoiced와 encoding 방법 같다

* θ_{VUV}는 0.5 근방

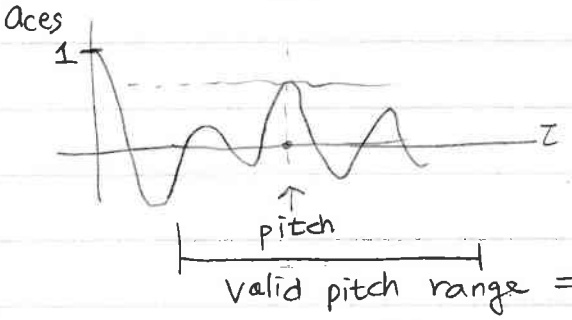
unvoiced excitation encoding

R3 Input : g_k, $e_s[t,k]$ $(= e[t,k]/g_k)$, VUV = UV (boolean)
generate Gaussian random noise with $\mu=0$, $\sigma^2=1$ of size N_s
$\hat{e}_s[t,k] = N(0,1)[1 \dots N_s]$

R4 pitch estimation : 최대 주기 찾기.
* clipping, median filtering 등의 technique은 강의자료를 읽어보기 바람 여기서는 autocorrelation을 쓰는 기본적인 방법
* 주의: VUV decision에서는 raw frame $x[t,k]$를 썼지만 pitch estimation에서는 $x_{pn}[t,k]$의 LPC residual (normalized) $e_s[t,k]$를 쓴다.

① main concept: acc0]을 제외한 나머지에서 max 위치가 pitch

$$acc_{es}[\tau,k] = \sum_{t=\tau}^{N_s} e_s[t,k] \cdot e_s[t-\tau,k]$$



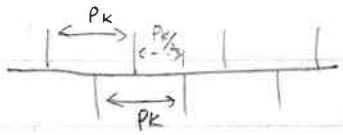
$$p[k] = \arg\max_{p_{min} \le \tau \le p_{max}} acc_{es}[\tau,k]$$

⇒ 찾아보기 바람

② median filter 등을 통해 smoothing

R5 pulse train generater w/ p[k]

① generate



* 실제 glottal pulse의 모양은 이므로 가 더 적합하지만 +/- 신호의 간격을 정확히 추정 못하기 때문에 p_k / p_k들로 +/- 생성.

② time alignment : $e[t,k]$와 correlation이 가장 높은 alignment 찾는다

$$\tau_a = \arg\max_{0 \le \tau \le p_{k-1}} e[t,k] \cdot r_1[t+\tau]$$

$$r_2[t] = r_1[t+\tau]$$

(주의) pulse의 개수가 같아야 한다

③ normalize $\hat{e}_s[t,k] = r_2[t] \Big/ \sqrt{\frac{1}{N_s}\sum_{t=1}^{N_s} r_2^2[t]}$

Encoder output: $C_{LSF}[k] \in \mathbb{Z}$ (1 integer)

$flag_{VUV} \in \{T, F\}$

$g[k] \in \mathbb{R}$

$p[k] \in \mathbb{Z}$ (int) ← voiced only

Decoding:

① $C_{LSF} \rightarrow LSF[1..M] \rightarrow (a_0, a_1, \ldots, a_M) \in \mathbb{R}^{M+1}$

② if $flag_{VUV}$ is F (unvoiced or silence)

$\hat{e}_s[t,k] = N(0,1)[1..N_s] \in \mathbb{R}^{N_s}$ ··· R3

③ else (voiced)

$\hat{e}_s[t,k]$ = periodic pulse train ··· R5

④ apply gain: $\hat{e}[t,k] = g[k] \cdot \hat{e}[t,k]$ ··· R6

⑤ $\hat{x}_{pn}[t,k] = \dfrac{\hat{e}[t,k]}{a[0..M]}$ (IIR filtering)

Programming assignment 3

R1 ~ R6 구현하고 encoder / decoder 구현

- V/UV 경계를 spectrogram 위에 그린다.
- pitch 도 그린다.