

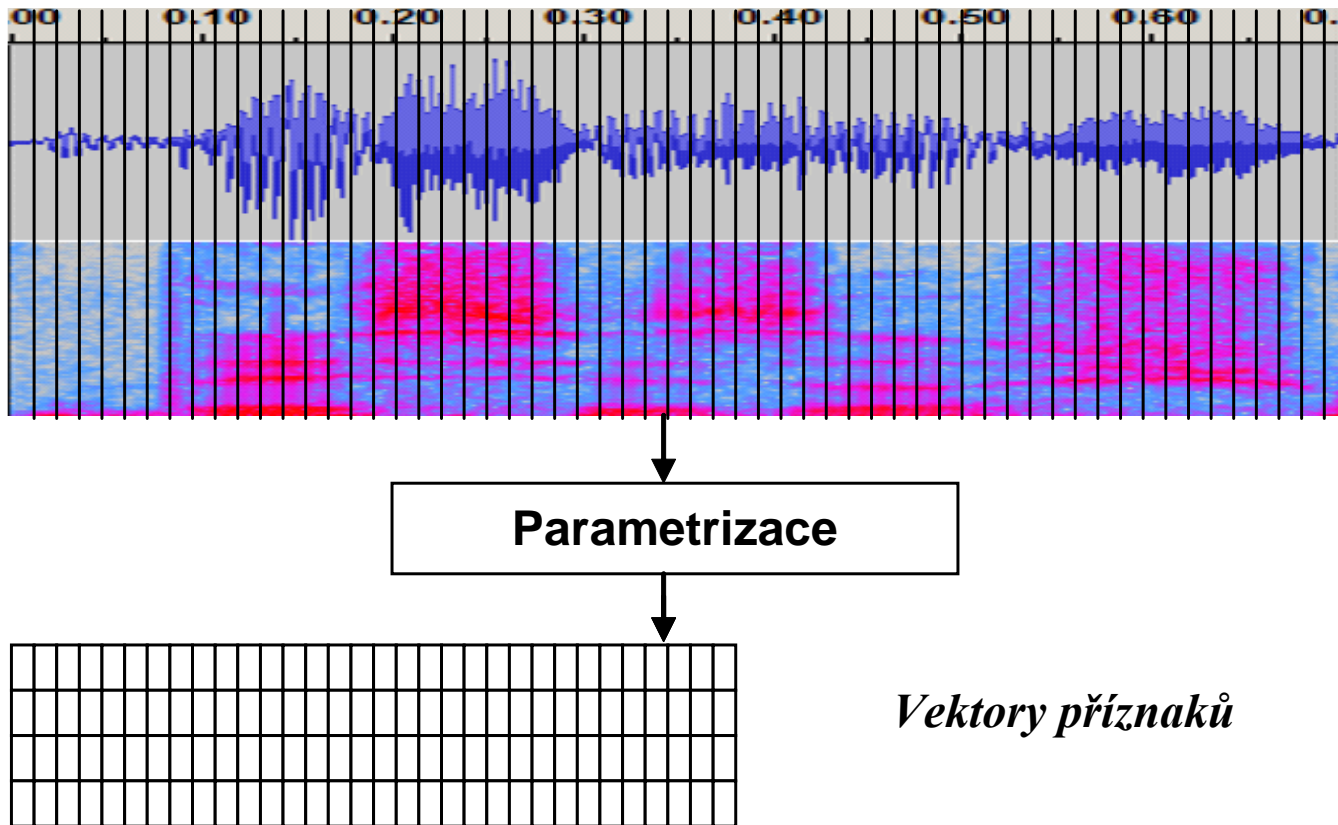
# **Pokročilé metody rozpoznávání řeči**

## **Přednáška 2**

**Kepstrum a kepsrální příznaky**

# Parametrizace signálu

**Cíl:** reprezentovat signál redukováním počtem dat vhodných pro rozpoznávání



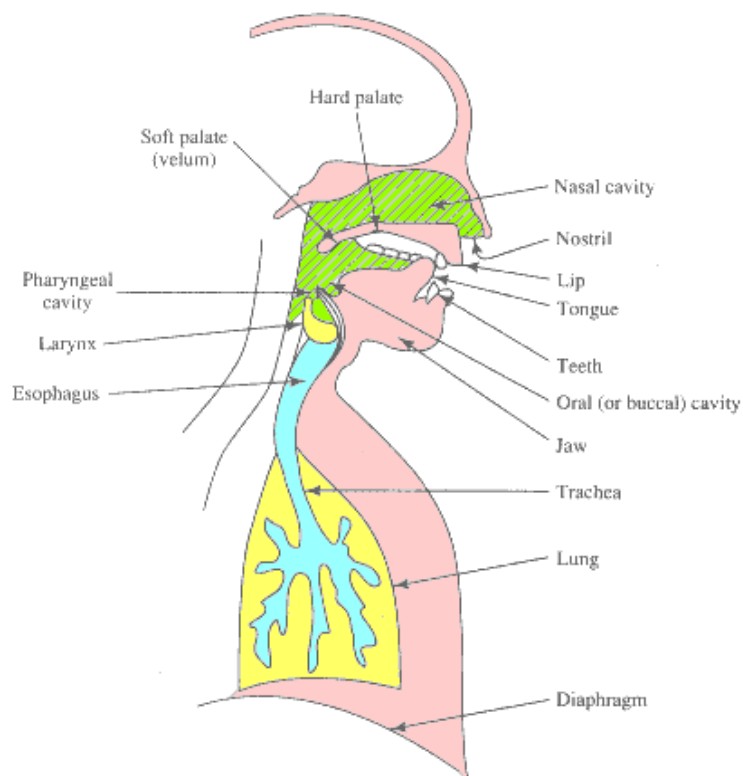
# Historický vývoj příznaků

- ~ 1960 – energie, počet průchodů nulou (minimální výpoč. nároky)
- ~ 1970 – spektrum a spektrální příznaky (možné díky FFT)
- ~ 1980 – lineárně prediktivní koeficienty (LPC)
- ~ 1990 – kepstrum a kepstrální příznaky (MFCC), delta příznaky
- ~ 2000 – různé modifikace kepstrálních příznaků (PLP, RASTA), transformace příznakových vektorů
- ~ 2010 – návrat ke spektr. příznakům (v souvislosti s DNN)
- ~ 2015 – rozvoj DNN umožnil experimentovat přímo se vzorky (bez nutnosti příznaků)

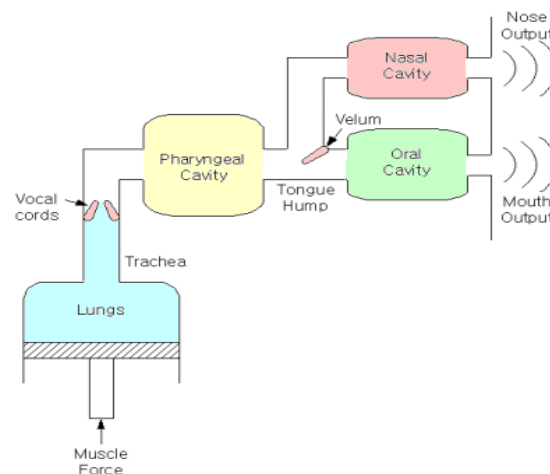
**Kepstrální příznaky patří dodnes k nejpoužívanějším v praxi.**

# Modely tvorby řeči

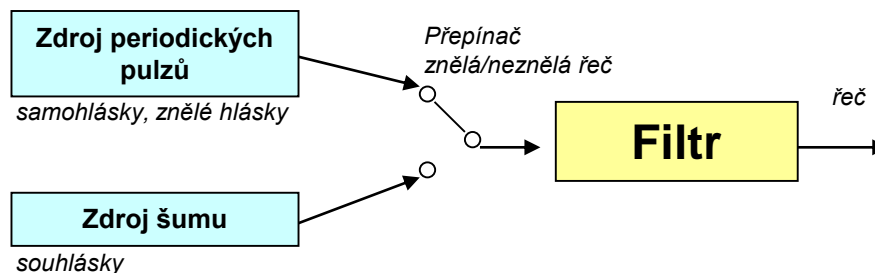
## Řečové orgány



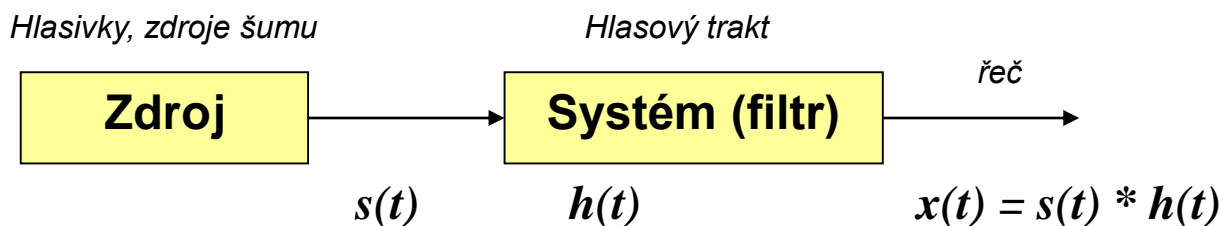
## Mechanický model



## Systémový model: zdroj - filtr



# Systémový model



V modelu je řeč konvolucí zdrojového signálu a imp. odezvy systému

Co je důležitější pro rozpoznávání OBSAHU řeči?

- charakter zdroje závisí na výšce hlasu, na intonaci, ...

- nastavení systému (filtru) se mění v závislosti na hláskách

Jak odělit informaci o systému od informace o zdroji?

Je třeba provést dekonvoluci.

# Kepstrum a jeho princip

Konvoluce v časové oblasti

$$x(t) = s(t) * h(t)$$

se ve spektru změnil na součin

$$X(f) = S(f) \cdot H(f)$$

po zlogaritmování na součet

$$\log X(f) = \log S(f) + \log H(f)$$

po inverzní FFT

$$x'(t) = s'(t) + h'(t)$$

Prostor, ve kterém jsou definovány signály  $x'(t)$ ,  $s'(t)$  a  $h'(t)$  se nazývá

**KEPSTRUM** angl. Cepstrum

Názvosloví vzniklo přesmyčkami (1963)

Spectrum -> Cepstrum

Frequency -> Quefrequency

Filter -> Lifter

# Praktický význam kepstra

Konvoluce v časové oblasti

$$x(t) = s(t) * h(t)$$

se převede na součet v kepstru

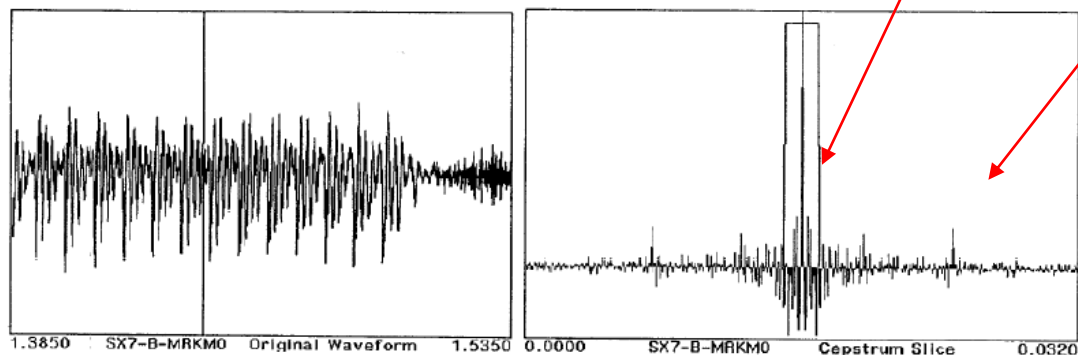
$$x'(t) = s'(t) + h'(t)$$

a pokud se obě složky  $s'(t)$  a  $h'(t)$  nacházejí v různých oblastech na *kefrenční ose*, dají se od sebe separovat.

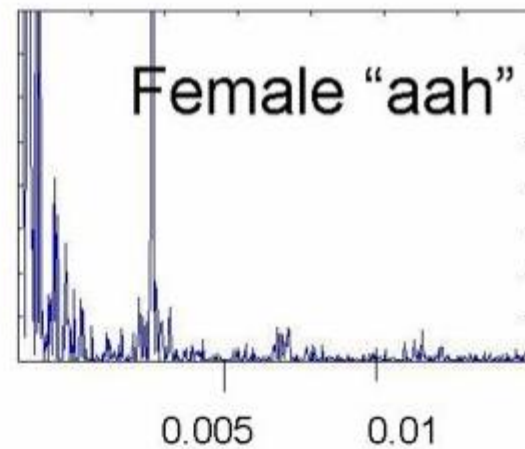
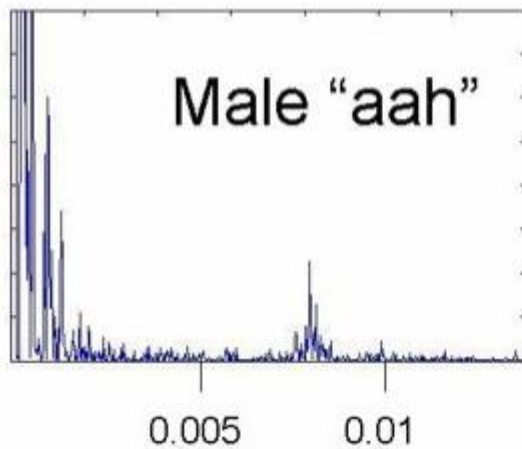
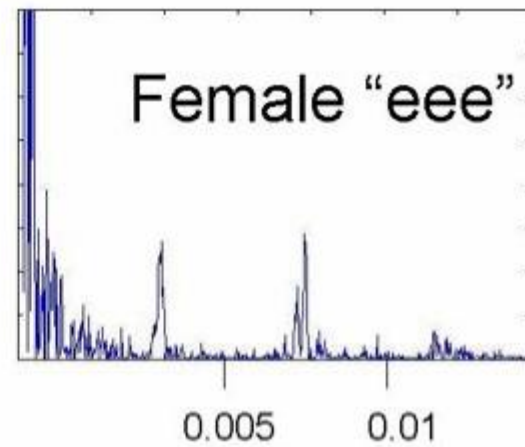
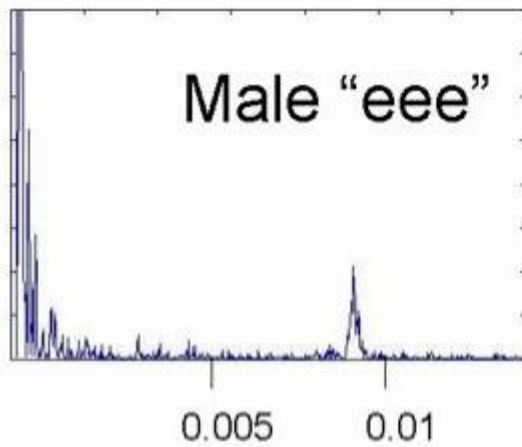
U řeči toto platí, protože

periodické (hlasivkové) buzení **se transformuje do oblasti vyšších kefrencí**, zatímco informace o filtru **se soustředí na nízkých kefrencích**

a lze je tedy oddělit vhodným výřezem (oknem)



# Illustrate kepstra (1)



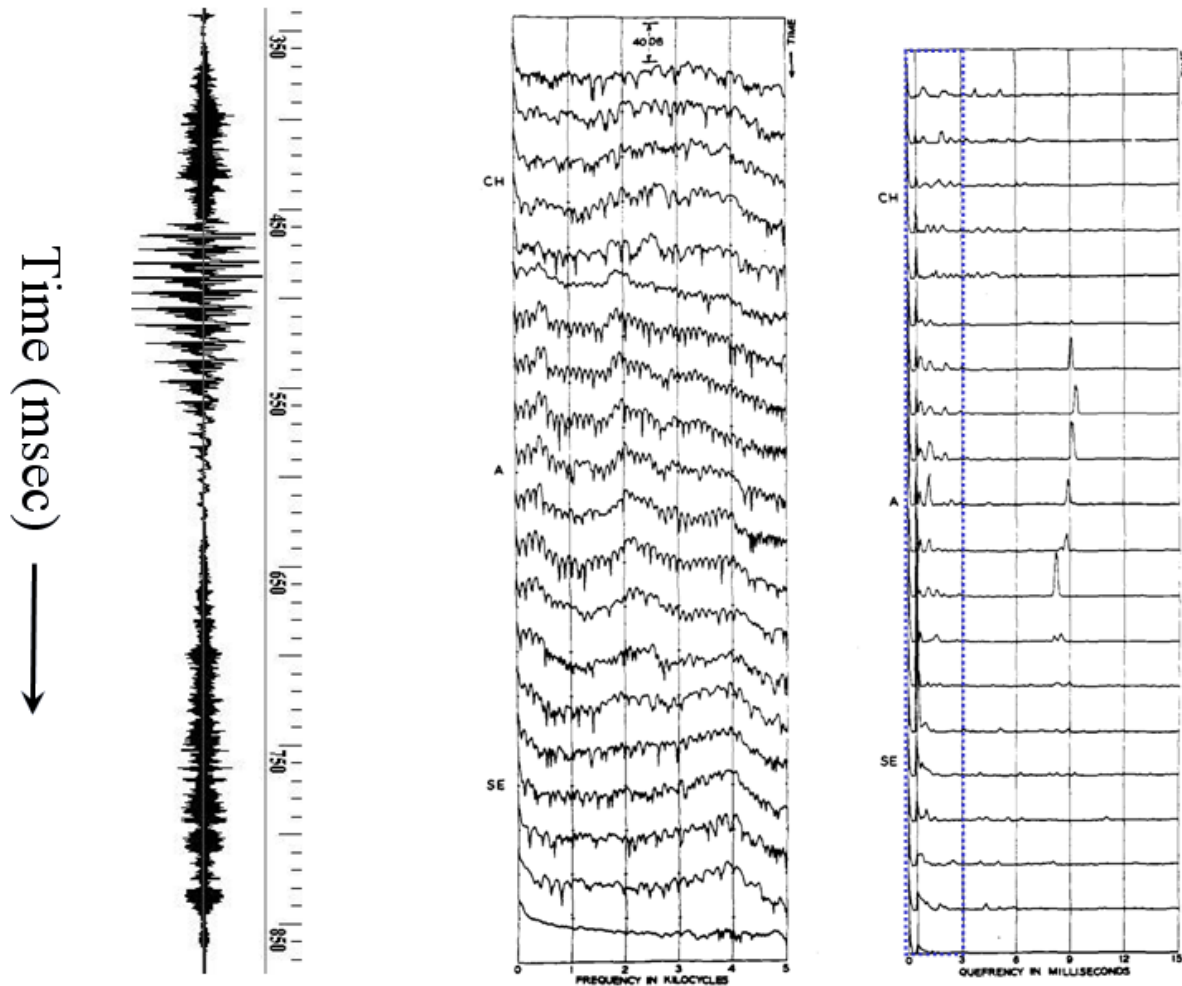


# Illustrate kepstra (2)

time domain

spectral domain

cepstral domain



(spectrum and cepstrum image from A.M. Noll, 1967)

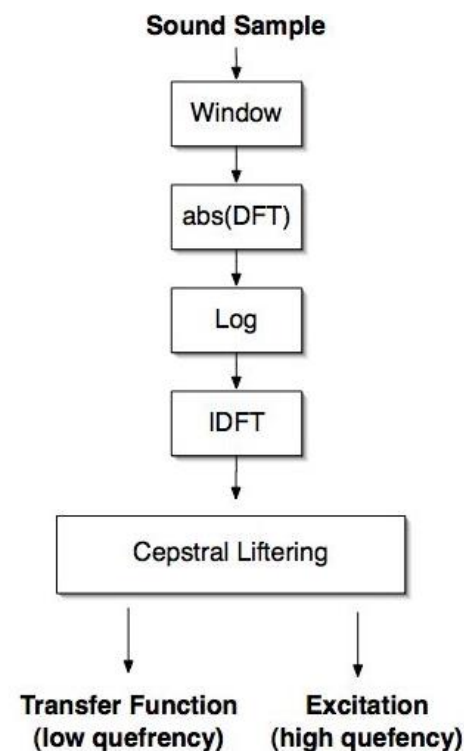
# Reálné kepstrum

Při zjednodušeném výkladu jsme se nezabývali tím, že spektrum je definováno v komplexní rovině, logaritmus by tudíž také musel být komplexní, a komplexní by tudíž bylo i kepstrum.

Pro praxi je dobře použitelné **reálné výkonové kepstrum**.

## Kroky při jeho výpočtu:

1. Signál v daném framu
2. Vynásobení hammingovým oknem
3. FFT
4. Modul FFT a kvadrát (výkon)
5. Logaritmus
6. IFFT
7. Vyříznutí nízkých kefrencí - vynásobení vhodným oknem



# Dvě metody výpočtu kepstra

## 1. LPC kepstrum

**Přes výpočet lineárně predikčních koeficientů (LPC)**

- je rychlejší na výpočet,
- snáze implementovatelné,
- využíváno zejména v 90. letech

## 2. MF kepstrum (MFCC, Mel-Frequency Cepstral Coefficients)

**Výpočet podle definice přes FFT, log a IFFT.**

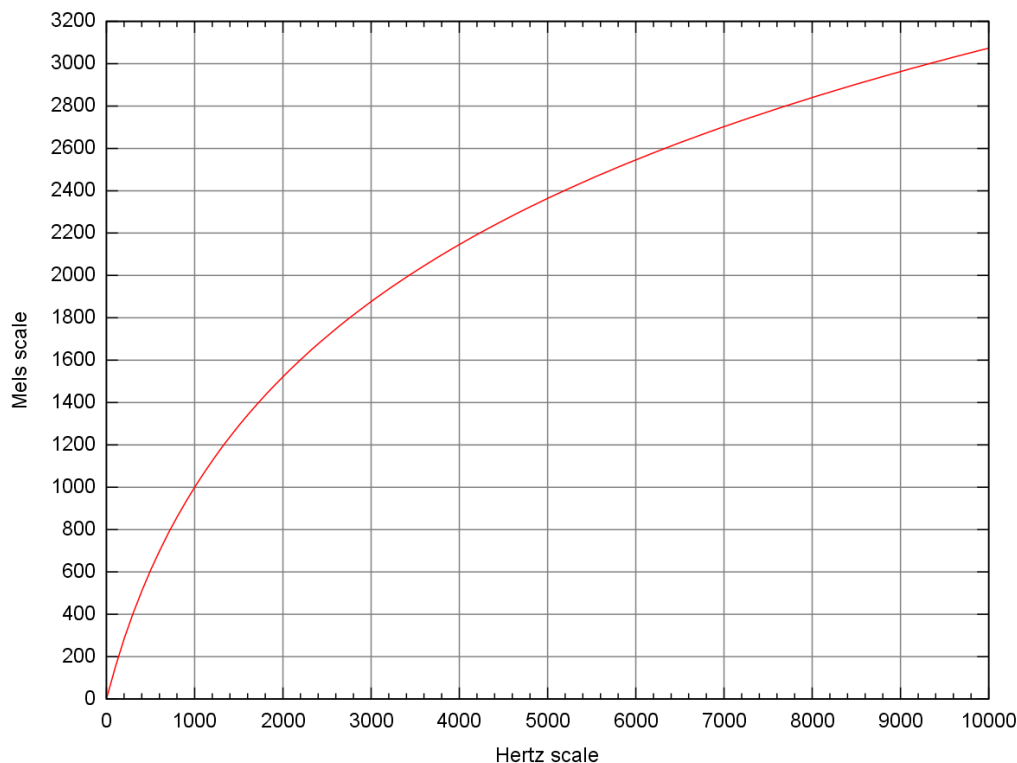
- používá křivku lidského vnímání frekvencí (mel stupnici)
- dává poněkud lepší výsledky při rozpoznávání,
- implementace na dnešních procesorech již není problém

# Melová stupnice frekvencí

Frekvence se standardně měří v jednotkách Hz.

Lidské ucho však vnímá zvukové frekvence poněkud odlišně

– u vyšších frekvencí již není schopno tolik rozlišovat rozdíl.



Experimentálně stanovena křivka  
a převodní vztah:

$$m = 2595 \log_{10}\left(\frac{f}{700} + 1\right) = 1127 \log_e\left(\frac{f}{700} + 1\right)$$

Vznikla myšlenka, že i pro  
rozpoznávací systémy by bylo  
vhodné skutečné frekvence  
transformovat podle této křivky.

Výsledky ukázaly, že to **funguje**.

# Podrobný popis výpočtu MFCC (1)

Uveden popis, který se standardně používá v mnoha ASR systémech (včetně našich na TUL), a který je standardně k dispozici v HTK.

Níže budou uvedeny parametry a nastavení pro řeč vzorkovanou na **16 kHz**.

## 1. Krok – Vyříznutí jednoho framu signálu

Délka framu                      25 ms – 400 vzorků

Posun framu                      10 ms – 160 vzorků

## 2. Krok – aplikace preemfázového filtru

Signál ve framu projde HP filtrem  $y(n) = x(n) - 0,97 x(n-1)$

Přínosy:

- a) posíleny vyšší frekvence (jsou zeslabeny cestou k mikrofonu)
- b) dynamicky potlačena ss složka vznikající na zvukových kartách

# Podrobný popis výpočtu MFCC (2)

## 3. Krok – Aplikace Hammingova okna

Na frame se 400 vzorky je aplikováno H. okno o stejné délce

## 4. Krok – Výpočet FFT

400 vzorků se doplní nulami na 512 a je proveden klasický výpočet 512-bodové FFT

## 5. Krok – Výpočet spektrálního výkonu

Pro prvních 256 hodnot diskrétního spektra se určí vždy nejprve modul (absolutní hodnota) a pak její kvadrát (výkon).

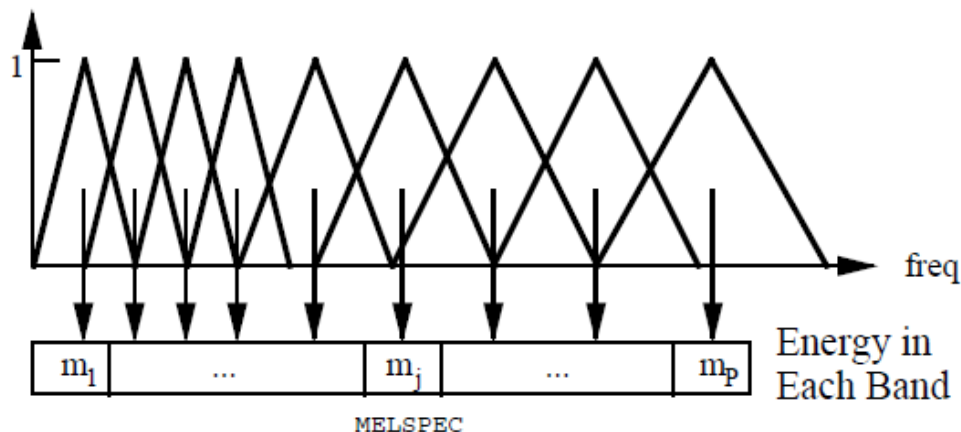
# Podrobný popis výpočtu MFCC (3)

## 6. Krok – Rozdělení spektrálního výkonu do pásem

Zde se využije melová stupnice a na ní se pomocí trojúhelníkových oken definují (částečně se překrývající) pásma.

Výkony jednotlivých složek FFT se vždy vynásobí příslušným koeficientem okna a uvnitř okna se sečtou. Tak dostaneme výkony v jednotlivých pásmech.

Standardní počet pásem: 24



# Podrobný popis výpočtu MFCC (4)

## 7. Krok – Logaritmus

V každém pásmu se spočítá logaritmus výkonu v daném pásmu.

## 8. Krok – IFFT

Zpětná Fourierova transformace se v praxi provede pomocí takzvané DCT (Diskrétní kosinová transformace). Jejím výsledkem jsou už keprální koeficienty – nejčastěji se používá **prvních 13 koeficientů**.

## 9. Krok – Liftrace

Výsledné koeficienty se vynásobí okénkovou funkcí uvedenou níže. Vyrovnají se rozdíly v hodnotách rozptylech mezi koeficienty.

$$c'_n = \left(1 + \frac{L}{2} \sin \frac{\pi n}{L}\right) c_n$$



# Podrobný popis výpočtu MFCC (5)

## 10. Výpočet Delta a Delta-delta koeficientů

Ke statickým MFCC koeficientům se dopočtou dynamické (1. a 2. derivace). Používaný vzorec pracuje většinou s okolím 2 framy na obě strany.

$$d_t = \frac{\sum_{\theta=1}^{\Theta} \theta (c_{t+\theta} - c_{t-\theta})}{2 \sum_{\theta=1}^{\Theta} \theta^2}$$

## 11. Volitelný krok – Normalizace MFCC

Pokud se pracuje s nahrávkami z různého prostředí a získanými různými nahrávacími kanály, je vhodné provést operaci zvanou CMS nebo CMN (Cepstral Mean Subtraction/Normalization). Spočívá ve výpočtu středních hodnot všech koeficientů přes celou nahrávku a odečtení této hodnoty od koeficientů ve všech framech.

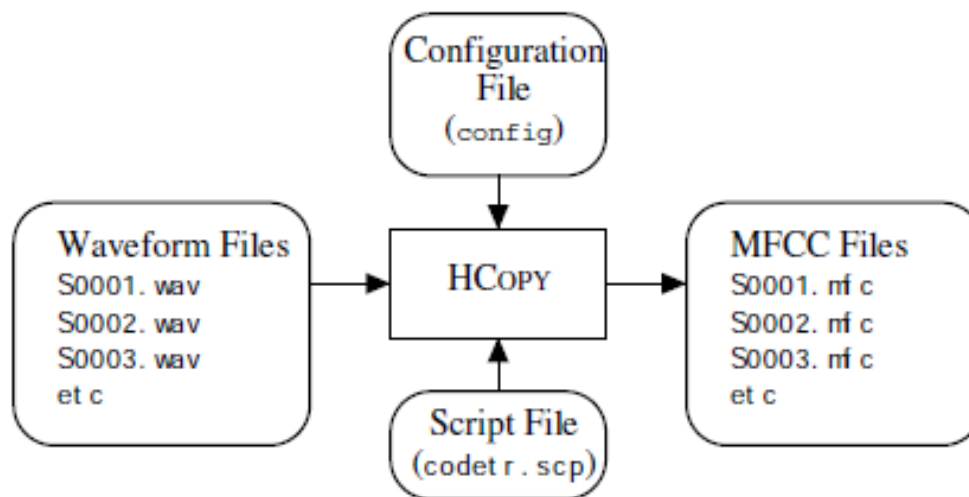
*Tuto operaci nelze provádět on-line*

*(resp. pouze se zpožděním a tzv. plovoucím oknem zahrnujícím cca 1s předchozího signálu)*

# Parametrizace v HTK (1)

V HTK jsou uvedené kroky prováděny programy **HCopy**, **HWave**, **Hparm** s nastavením, které je buď implicitní nebo nastaveno v konfiguračním souboru.

Použití programu HCopy



HCopy -C Param.cfg src.wav tgt.mfc

# Parametrizace v HTK (2)

**Konfig. soubor** pro parametrizaci – vytvoří příznaky typu MFCC\_0\_D\_A

Param.cfg

```
TARGETFORMAT = HTK
TARGETKIND = MFCC_0_D_A
SOURCEFORMAT = WAVE
SOURCEKIND = WAVEFORM
ENORMALISE = F
WINDOWSIZE = 250000
TARGETRATE = 100000
PREEMCOEF = 0.97
USEHAMMING = T
NUMCEPS = 12
NUMCHANS = 24
CEPLIFTER = 22
DELTAWINDOW = 2
ACCWINDOW = 2
EXTENDFILENAME = T
SAVEWITHCRC = F
USEPOWER = F
ADDDITHER = -0.0000306
NATURALREADORDER = T
NATURALWRITEORDER = T
NONUMESCAPES = T
```

# Úkoly do příště


1. Nahrát si trénovací data (100 vět)
2. Vytvořit jejich fonetické přepisy  
(pomocí G2P z minula a ručního doladění).
3. Zparametrizovat všechny nahrávky

# Příprava trénovacích dat pro fonémový akustický model

## Požadavky:

- záznamy řeči v prostředí podobném cílové aplikaci
- nahrávky musí obsahovat všechny fonémy (nejlépe s odpovídající frekvencí)
- nahrávky musí pocházet od co největšího počtu osob
- nahrávky musí být textově a akusticky různorodé
- nahrávky by měly být foneticky jednoznačné (např. bez přeřeků)
- každá nahrávka musí být foneticky správně a přesně přepsána

Ke každé nahrávce musí existovat 4 soubory (jména bez diakritiky!)

1. zaznam001.wav (nahrávka) 
2. zaznam001.txt (textový přepis) K obědu si dám pizzu a džús.
3. zaznam001.phn (fonetický přepis) - k objedu si dám picu a Čús –
4. zaznam001.lab (fonetický přepis ve formátu pro HTK)

# Jak zvolit a nahrávat trénovací věty

1. Vytvořit seznam 100 vět.
2. Věty by měly být snadno vyslovitelné, nejlépe najednou (bez pauzy).
3. Ideální věty obsahují 10-15 slov, číslovky jsou rozepsány.
4. Věty lze brát z tisku či z jiných zdrojů (vyvarovat se cizích slov).
5. Ve větách by se měly objevit všechny fonémy, ty nejméně časté alespoň 3 x.
6. Texty převést do formátu CP1250!!! (ne UTF8, kvůli kompatibilitě)
7. K nahrávání použít vhodný software (Audacity), dobrý mikrofon.
8. Nastavit si 16 kHz a 16 bit, mono!!!
9. Vypnout případnou funkci typu Speech denoising (enhancement)
10. Větu si přečíst a pak v klidu nahrát.  
Zajistit, aby před řečí bylo cca 0,5 sekundy ticha, totéž za větou.
11. Uložit pod správným jménem \*.wav a \*.txt.
12. Vytvořit ke každé větě fonetický přepis pomocí vašeho G2P nebo ručně (pozor na „y“, „ě“, „ď“, „X“)
13. Pozor též na podobu  
„muž je“ -> „muš je“                      ale                      „muž byl“ -> „muž bil“

# Přepis trénovacích dat (1)

## Postup:

1. Máme nahrávku v souboru **\*.wav** a k ní textový přepis v souboru **\*.txt**
2. Pomocí přepisovacích pravidel (nejlépe s využitím programu G2P) vytvoříme nový soubor **\*.phn** obsahující fonetický přepis nahrávky (včetně případného ticha na začátku, konci, případně uprostřed).  
*Společnost Diamo byla založena devatenáctého listopadu*  
*-společnost\_d'iamo\_bila\_založena\_devatenáctého\_listopadu-*  
*(symbol „\_“ je použit pro usnadnění čtení)*
3. Poslechem zkontrolujeme automaticky vytvořený přepis a opravíme případné chyby (u cizích slov, vliv spodoby na švu slov, apod.) či doplníme ticho a šumy  
*-společno**zd**\_dijamo\_bila\_založena\_devatenáctého\_listopadu-*

# Přepis trénovacích dat (2)

4. Ze souboru **\*.phn** automaticky vytvoříme soubor **\*.lab** (jedna hláska na jednom řádku)  
(trénovací program nedovoluje diakritiku, nutno použít angl. symboly – viz soubor **alphabet48-CZ.abc**)

0 0 si  
0 0 s  
0 0 p  
0 0 o  
0 0 l  
0 0 e  
0 0 ch  
0 0 n  
0 0 o  
0 0 z  
0 0 d



# Převodní soubor alphabet48-CZ.abc

Obsahuje celou fonetickou abecedu (48 symbolů fonémů, ticha a hluků) v různém kódování.

Na každém řádku je: index hlásky, symbol v kódu CP1250, anglický symbol

0 a a  
1 á aa  
2 b b  
3 c ts  
4 C dz  
5 č ch  
6 Č dg  
7 d d  
8 d' dj  
....  
40 - si  
41 E swa  
42 1 n1  
43 2 n2  
44 3 n3  
45 4 n4  
46 5 n5  
47 0 n0

# Symboly pro neřečové zvuky

Pokud pracujeme s nahrávkami, kde se vyskytuje nejen řeč a ticho, ale i další neřečové zvuky a ruchy, používáme k jejich anotaci tyto symboly:

Symbol	Typ hluku	Příklad, poznámka	HTK symbol
-	ticho		si
0	ráz	EU [0é0ú]	n0
1	klik	krátký zvuk	n1
2	ruch	delší slabší zvuk	n2
3	nádech		n3
4	hluk	delší silný zvuk, hudba, ...	n4
5	ehm	váhací zvuk	n5