Počítačové zpracování řeči

Přednáška 2

Úvod do rozpoznávání řeči
Základní kroky při rozpoznávání izolovaných slov

Rozpoznávání izolovaných slov (IWR)

IWR – Isolated Word Recognition

CSR –Continuous Speech Recognition

IWR předpokládá:

- Slova jsou vyslovována vždy s krátkými pauzami mezi nimi
- Pauzy by měly být alespoň 0,5 - 1 s dlouhé, aby se dal spolehlivě určit začátek a konec slova
  Pozn. Je třeba mít na paměti, že krátké pauzy (až do 0,3 s) se vyskytují i uvnitř slov (před explozivami jako jsou p, t, k, ...)
- V některých aplikacích lze za „izolovaná slova“ považovat i víceslovní výrazy či povely, pokud jsou takto jako položky obsaženy ve slovníku a jsou vyslovovány najednou,
  Např. “New York”, “Otevři soubor”, “Do Prahy”, “Z Prahy”, „Přepoj na operátorku“

Proč je IWR jednodušší než CSR?

1. Slova pronášená izolovaně jsou obvykle **vyslovována pečlivěji** (nehrozí mezislovní koartikulace, řečník má dost „sil“ na vyslovení celého slova),
   srovnej izol. a spojenou výslovnost “pět šest”, “mít smůlu”, …
2. Při izolované výslovnosti je mnohem snazší **detekovat hranice slova** (začátek a konec slova)
3. V úloze IWR můžeme předpokládat, že v daném časovém úseku (vymezeném detektorem řeči) je **právě jedno slovo (jedna položka slovníku)**. Klasifikátor pak řeší úlohu: Která z N položek ze slovníku to je?
   Úloha má **lineární složitost**

   Naproti tomu CSR má **exponenciální složitost**: Hledáme mezi všemi možnými kombinacemi N slov ze slovníku.

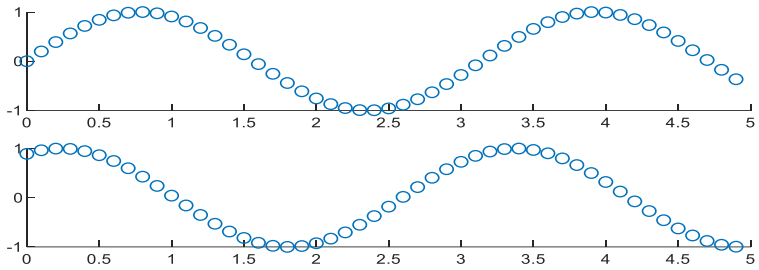
Jaké kroky je třeba udělat při IWR

1. Nahrát slovo, získat navzorkovaný signál
2. Najít okamžiky, kdy slovo v nahrávce začíná a končí
3. Určit příznaky použitelné pro rozpoznávání

   samotné vzorky nejsou pro rozpoznávání vhodné, proč?

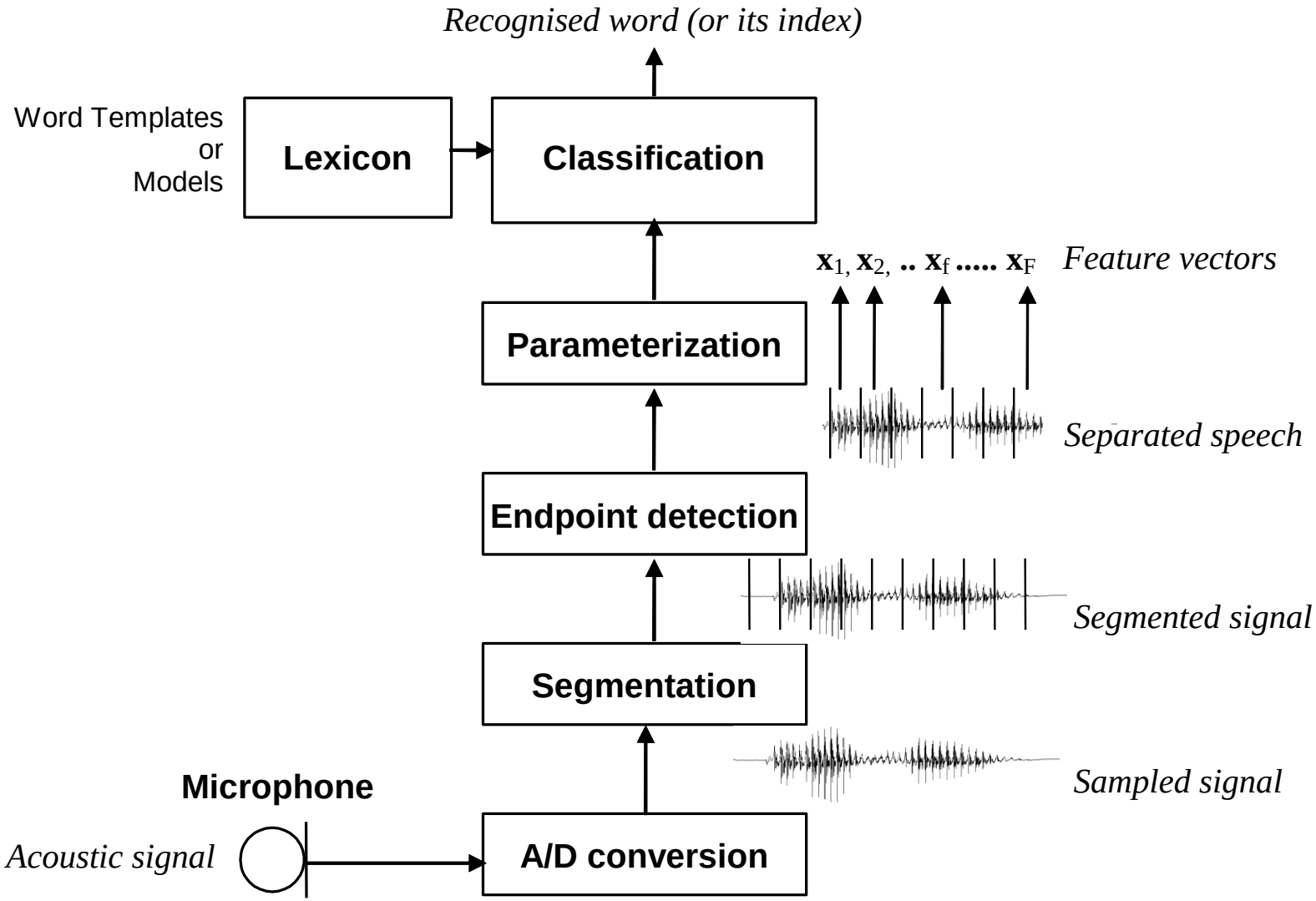
   a) je jich příliš mnoho (např. při vzorkovací frekvenci 16 kHz, 16 000 hodnot za 1 s)

   b) případný fázový posun by mohl značně komplikovat měření podobnosti

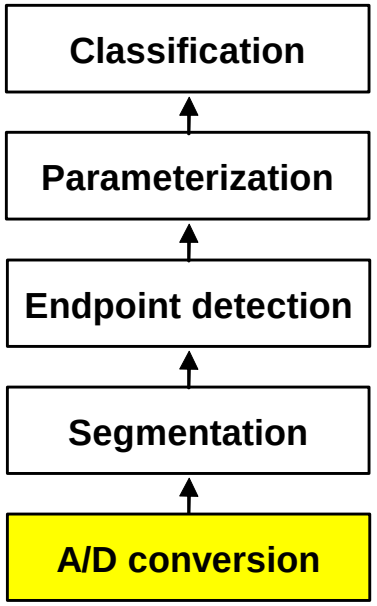


4. Pro každé slovo připravit vzory (reference, modely …)
5. Změřit míru podobnosti mezi slovem a jednotlivými vzory
6. Určit nejpodobnější vzor (model).

Základní kroky IWR



Základní kroky (1)



A/D převod

- **automaticky** prováděn zvukovou kartou
- nejčastěji užívané **vzorkovací frekvence** – 16 kHz a 8 kHz – **mono** (!)
- je nutné používat **stejnou** frekvenci při trénování i provozování klasifikátoru
- ***Pozn. 1:*** některé zvuk. karty generují **nuly** na zač. a konci vzorkovacího procesu – to může způsobit problémy při výpočtu některých parametrů, které používají operace dělení či logaritmování

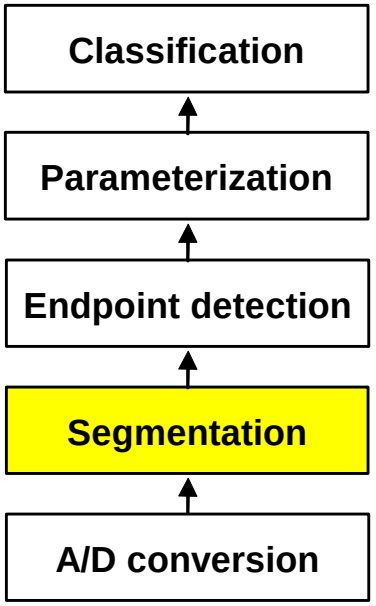
  řešení: přičíst malý šum ke každému vzorku signálu

  $y(n) = x(n) + Rnd(-1, 0, 1)$
- ***Pozn 2:*** některé zvukové karty mají nenulovou (někdy i časově proměnnou) stejnosměrnou složku (DC offset)

  řešení: ta se dá snadno odstranit tzv. preemfázovým filtrem (jednoduchý derivační filtr typu HP)

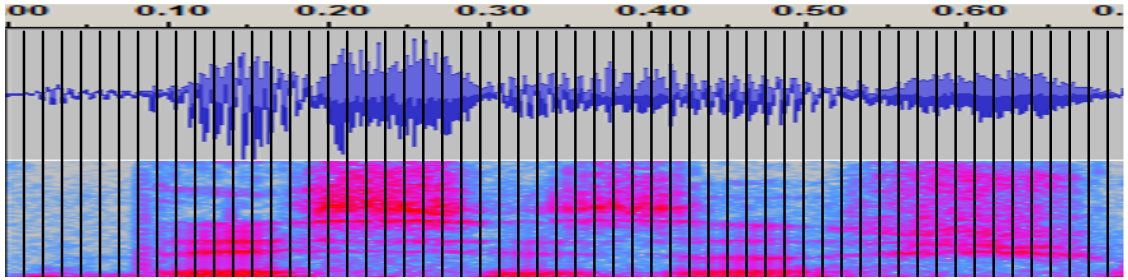
  $y(n) = x(n) - \alpha x(n-1)$ kde $\alpha \approx 1$, obvykle $\alpha = 0.97$

Základní kroky (2)

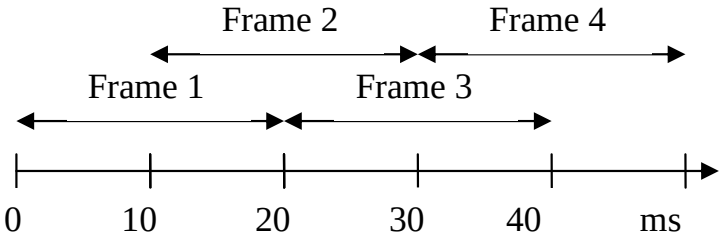


Segmentace signálu

- Charakter signálu se mění v čase, proto je třeba signál nejprve rozdělit do krátkých úseků, **framů**.

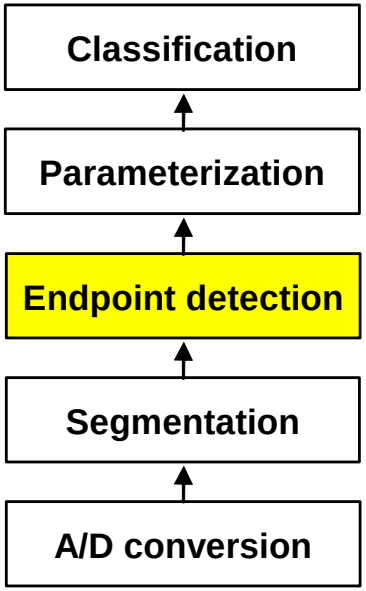


- Segmentace se dělá **pravidelně** a **délka úseků** by měla být kratší než trvání nejkratších fonémů (obvykle 10 – 25 ms).
- Framy se většinou definují tak, že se částečně **překrývají**, abychom docílili hladší průběh parametrů počítaných v jednotlivých framech.



- **Typické hodnoty** pro 16 kHz signál: délka framu 25 ms, framy začínají každých 10 ms (překryv 15 ms)

Základní kroky (3)



Detekce začátku a konce řeči

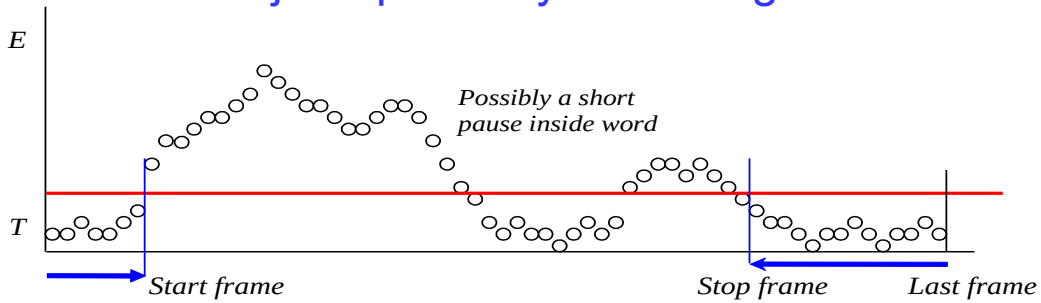
- Předpokládáme, že **energie** *signálu pozadí* ("ticha") je menší než energie *v řečových framech* (při velkém šumu a hluku to nemusí platit).
- Energii signálu ve framu počítáme obvykle v log. měřítku

  L je počet vzorků ve framu

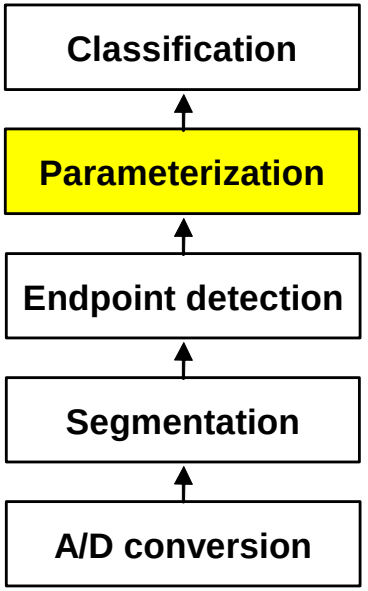
$$E = \log\left(\sum_{n=0}^{L-1} x^2(n)\right)$$

- **<u>Jednoduché praktické řešení (vhodné pro off-line rozpoznávání):</u>**
  Určíme parametr E ve všech framech signálu a hledáme
  a) začátek řeči jako – první frame, kde E > T (T ... práh)
  b) konec řeči jako – první frame **hledaný od konce**, kde E > T
  Vhodná hodnota T se dá najít např. analýzou histogramu

Ilustrace



Základní kroky (4)



Určení příznaků (parametrů) signálu

- **Příznaky** – parametry užívané při rozpoznávání
  Příznakový vektor – soubor příznaků, který se určuje v každém framu
- Příklady jednoduše získatelných příznaků:

Log energy $E = \log(\sum_{n=0}^{L-1} x^2(n))$

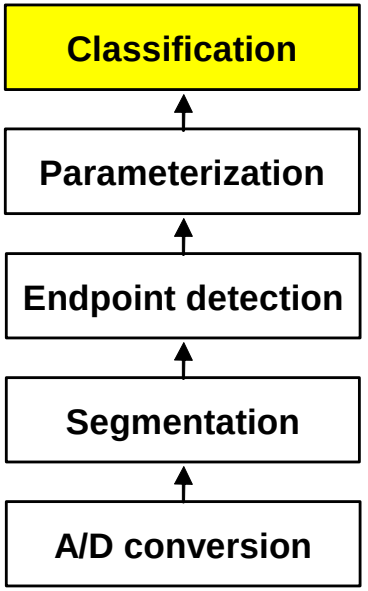
Zero crossing rate (ZCR) $ZCR = \frac{1}{2}(\sum_{n=1}^{L-1} |\operatorname{sgn} x(n) - \operatorname{sgn}(n-1)|)$

Pozn.: Energie je jednoduchá míra související s intenzitou signálu ve framu.
Její logaritmus převádí příliš velký rozsah na výrazně menší rozsah
ZCR je jednoduchá míra související s dominantní frekvencí signálu ve framu
Jak E tak i ZCR vyžadují, že ze signálu musí být odstraněna stejnosm. složka.

- **Pokročilejší příznaky**:

- spektrální příznaky, kepstrální příznaky, dynamické příznaky

Základní kroky (5)



Rozpoznání neznámého slova

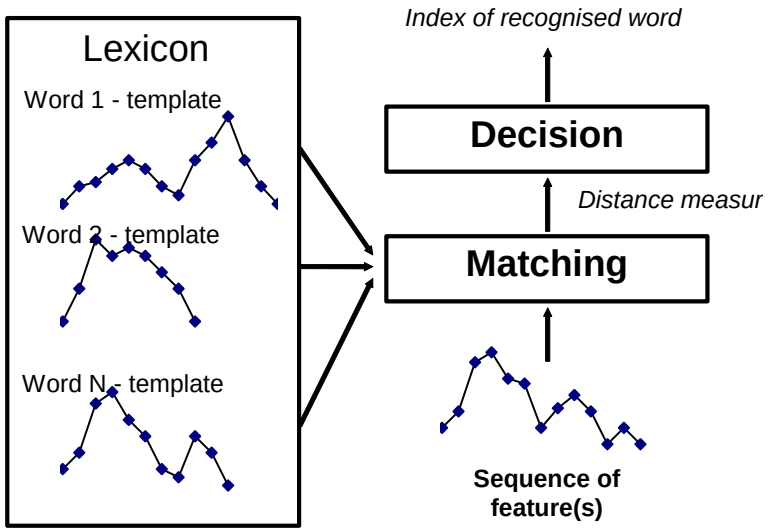
- Neznámé slovo, které má být rozpoznáno je **reprezentováno** *sekvencí příznakových vektorů*.

Metoda porovnávání s referencemi:

- Během "trénovací fáze" jsou všechny položky slovníku namluveny a reprezentovány též *sekvencemi příznakových vektorů.*
- **Reprezentace** neznámého slova je **porovnávána** se všemi referencemi. Reference, která je nejpodobnější, určí identitu slova.

Pokročilejší metody:

Metoda HMM - Porovnávání s pravděpodobnostními modely, (Hidden Markov Models).



Úloha pro cvičení

Vytvořit sadu programů, které

A) připraví vhodná data pro experimentování s rozpoznáváním izolovaných slov (trénovací a testovací sady)

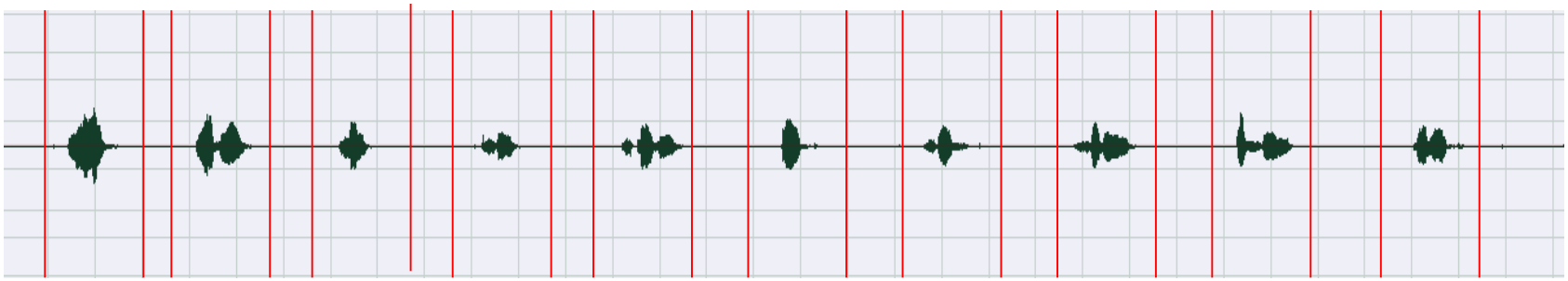
B) najdou začátek a konec slova v každé nahrávce

C) vypočítají a zobrazí sekvence dvou jednoduchých příznaků pro každé slovo

A – Příprava nahrávek číslic

Cíl: Ve zvukové nahrávce automaticky detekovat slova, vyříznout je (i s okolím) a uložit je do souborů

Účel: Připravit si vhodná data pro úlohu rozpoznávání slov



Parametry nahrávek: Fs = 16 kHz, 16 bitů, mono, formát wav

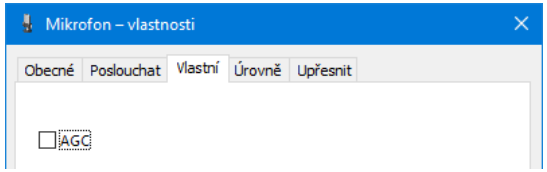
Slova v nahrávce (10): nula, jedna, dva, tři, ….. devět

Pauzy mezi slovy: dostatečně dlouhé (nejméně 2-3 sekundy) kvůli vyříznutí
(výřezy v žádném případě nelze doplňovat nulovým signálem)

Počet nahrávek: 5 od každé osoby

Vyříznutí jednotlivých číslic a uložení

1. Nahrajte si (v tichém prostředí a vhodným mikrofonem) 5 nahrávek (vaším hlasem). Nahrávejte pomocí close-talk mikrofonu,
   na počítači v nastavení mixeru vypněte úpravu signálu
   (zejména Autom. Gain Control, nebo Noise Supression)



2. Napište program, který tyto nahrávky postupně načte, najde v nich slova, tato slova vyřízne a každé uloží do vlastního souboru.

3. Pro nalezení zač. a konce slov použijte analýzu průběhu energie (po 25 ms úsecích). Zvolte vhodný práh, který odliší energii řeči od energie „ticha“ v pozadí.

4. Vyříznuté nahrávky musí být opět 16 kHz, 16bitů, mono. Každá musí být 2s dlouhá (32 000 vzorků) a slovo by mělo ležet přibližně uprostřed.

5. Cílové nahrávky budou ve formátu WAV (s délkou 32000 x 2 + 44 bajtů).

6. Nahrávky budou pojmenovány takto:
   ci_pjjjj_snn.wav, např. c0_p0123_s01.wav, c8_p0878_s04.wav
   kde i je číslo vyslovené číslice, jjjj je 4-místné číslo osoby, nn je 2-místné číslo sady (sady 01, 02, 03, 04, 05). Číslo osoby určuje poslední slajd. Každá osoba bude mít svůj adresář pojmenovaný „pjjjj“, a v něm bude 50 dvousekundových nahrávek s výše uvedenými jmény.

B a C – Nalezení hranic slova a příznaky

V předchozí části jste vytvořili adresáře vždy s 50 nahrávkami.

1. Nyní vytvoříte Program_BC, který postupně načte všechny tyto nahrávky, provede s nimi operace a) přidání malého šumu, b) filtrace, c) segmentace na framy a d) výpočet 2 jednoduchých příznaků.

2. Pro načtení signálu používejte funkce AUDIOREAD, např.
   [x, Fs] = audioread („ c0_p0123_s01.wav“, ‘native’).
   Signál zobrazte a přehrajte.

3. V dalším kroku k signálu přičtěte malý šum
   např. x = x + (randi(3, 32000, 1, 'int16') - 2);
   a signál následně zfiltrujte filtrem FIR xf = FILTER ([1 -0,97], 1, x)
   Signál zobrazte a přehrajte.

4. Nyní signál rozdělte na framy dlouhé 25 ms (400 vzorků), které se částečně překrývají. Framy začínají každých 10 ms (160 vzorků), takže např. první frame začíná vzorkem 1 a končí vzorkem 400, druhý frame (161, 560), třetí (321, 720), ..

B a C – Nalezení hranic slova a příznaky (2)

1. V dalším kroku vypočítáte pro každý frame hodnoty E a ZCR a jejich průběh zobrazíte v diagramu.
2. Na základě průběhu E naleznete framy kde slovo začíná a končí a v digramu je vyznačíte, např. svislou čárou. Program následně vyřízne část signálu odpovídající slovu a toto přehraje. Tímto způsobem projděte všechny nahrávky a ověřte si, že nalezené hranice slova jsou správné. Pokud tomu tak není, změňte nastavení hodnoty prahu.
3. Tipy pro implementace této části najdete na posledním slajdu.

Tipy pro implementaci

Detekce začátku a konce řeči

Každá *sestava počítače, zvukové karty, mikrofonu a nastavení mixeru* ovlivňuje, jak silný bude signál i šum. Je proto nutné, aby se detektor neopíral o konstanty zjištěné na jednom počítači, ale aby byl schopen přizpůsobit se dané sestavě či dokonce nahrávce. Nabízí se 2 jednoduché možnosti adaptace:

1. Na **konkrétním počítači** vždy **předem** zjistit průměrnou úroveň (energie) šumu a řeči, a podle toho pak nastavit práh na určité procento rozdílu mezi těmito úrovněmi.

2. Na **konkrétní nahrávce** najít N (např. 10) framů s nejnižšími hodnotami energie a stejně tak i N framů s nejvyššími hodnotami. (První nejspíše odpovídají šumu, druhé naopak řeči.) Pro obě skupiny určit průměrnou hodnotu energie a podobně jako v 1) vypočítat vhodnou hodnotu prahu.

Čísla osob u nahrávek

Breuer Aleš 2301
Fejgl Miloš 2302
Kasal Luboš 2303
Linhart Tomáš 2304
Svárovský Jan 2305
Šafařík David 2306
Motejlek Martin 2307
Frydrychová Kristýna 2308

Zašlete mi prosím – nejdéle do pondělí 12.00

a) Data vytvořená programem A (v adresářích nazvaných p2301, …atd)
b) Program_BC, který půjde spustit s výše uvedenými daty (vyzkoušejte si, že fungují cesty)