

ISS Projekt 2018 / 19

Honza Černocký, ÚPGM FIT VUT
November 20, 2018

1 Úvod

Základem projektu je textový soubor obsahující 2000 symbolů 0 a 1. Na základě tohoto souboru bylo vygenerováno audio. Symboly '1' jsou v něm reprezentovány jako 16 vzorků s hodnotou 0.5, symboly '0' jako 16 vzorků s hodnotou -0.5, následně byl přidán šum. Toto audio bude vstupem Vašeho projektu. Oba soubory jsou k nalezení v privátní sekci webu ISS jako:

https://www.fit.vutbr.cz/study/courses/ISS/private/proj2018-19/sig_students/xlogin00.txt

https://www.fit.vutbr.cz/study/courses/ISS/private/proj2018-19/sig_students/xlogin00.wav

kde "xlogin00" je Váš login.

Projekt je možno řešit v Matlabu, Octave, Pythonu, jazyce C nebo v libovolném jiném programovacím nebo skriptovacím jazyce. Je možné použít libovolné knihovny. Projekt se nezaměřuje na "krásu programování", není tedy nutné mít vše úhledně zabalené do okomentovaných funkcí, atd. Důležitý je výsledek. Kód ale musí být možné spustit na školním Linuxu nebo Windows a musí prokazatelně produkovat výsledky obsažené ve Vašem protokolu.

2 Odevzdání projektu

bude probíhat do informačního systému WIS ve dvou souborech:

1. `xlogin00.pdf` (kde "xlogin00" je Váš login) je protokol s řešením.

- V záhlaví prosím uveďte své jméno, příjmení a login.
- Pak budou následovat odpovědi na jednotlivé otázky — obrázky, numerické hodnoty, komentáře.
- U každé otázky uveďte stručný postup - může se jednat o kousek okomentovaného kódu, komentovanou rovnici nebo text. Není nutné kopírovat do protokolu celý zdrojový kód. Není nutné opisovat zadání či teorii, soustřeďte se přímo na řešení.
- Pokud využijete zdroje mimo standardních materiálů (přednášky, cvičení a studijní etapa projektu ISS), prosím uveďte, odkud jste čerpali.
- Protokol je možné psát v libovolném systému (MS-Word, Libre Office, Latex), můžete jej psát i čitelně rukou, dolepit do něj obrázky a pak oskenovat. Protokol může být česky, slovensky nebo anglicky.
- Doporučená délka protokolu je 2 strany, případně 3, pokud se rozhodnete řešit bonusový úkol.

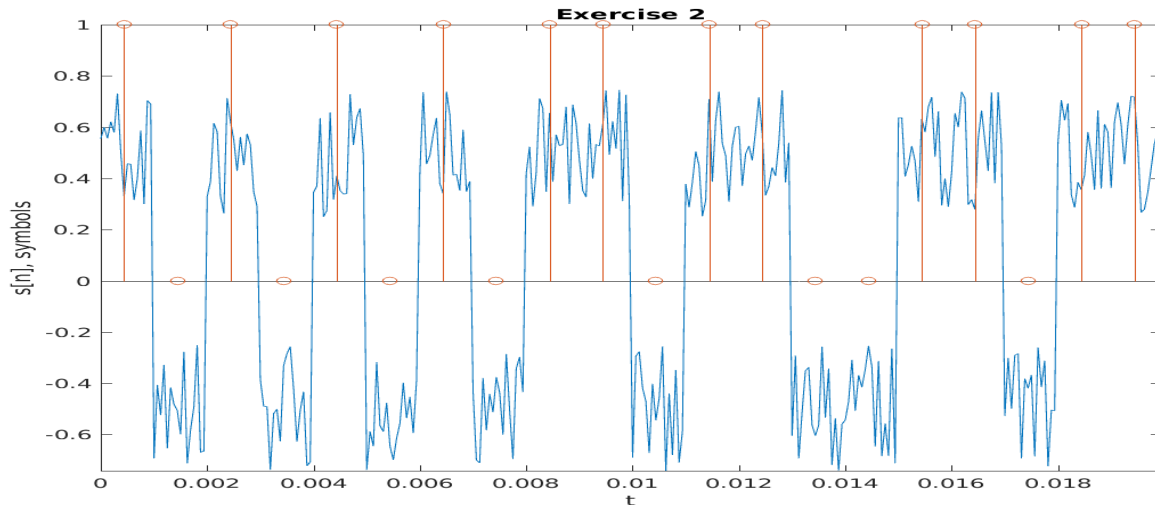
2. soubor s Vašimi zdrojovými kódy.

- Může se jednat o jeden soubor (např. `moje_reseni.m`) nebo o ZIP archiv, pokud budete mít souborů více. Není třeba přikládat vygenerované obrázky, ty uvidíme v protokolu.
- Projekt je **samostatná práce**, proto budou Vaše zdrojové kódy křížově korelovány a v případě silné podobnosti budou vyvozeny příslušné závěry.
- Silná korelace s kódy ze studijní etapy projektu je v pořádku, nemusíte tedy měnit názvy proměnných, přepisovat zbytečné komentáře, atd.

3 Zadání

1. [0.5 bodu] Načtete Váš osobní signál ze souboru `xlogin00.wav`, kde "xlogin00" je Váš login. Napište vzorkovací frekvenci signálu, jeho délku ve vzorcích a v sekundách a počet reprezentovaných binárních symbolů (1 symbol odpovídá 16ti vzorkům). Tento signál budu označovat jako $s[n]$.
2. [1 bod] Proveďte dekódování $s[n]$ do binárních symbolů. Nejjednodušeji tak, že vezmete vždy osmý vzorek ze segmentu 16-ti vzorků a pokud je > 0 , je výstupem 1, pokud < 0 , je výstupem 0. Můžete samozřejmě vyzkoušet i sofistikovanější techniky. Srovnajte se souborem `xlogin00.txt`, zda je vše v pořádku — nejprve

vizuálně, pro srovnání všech symbolů můžete použít funkci XOR. Do protokolu dejte obrázek prvních 20 ms audio signálu s vyznačenými dekódovanými symboly – může vypadat jako na následujícím příkladu, ale symboly mohou být také značeny jako text 0 / 1, nebo jakkoliv jinak chcete.



3. [0.5 bodu] Je zadán filtr s přenosovou funkcí

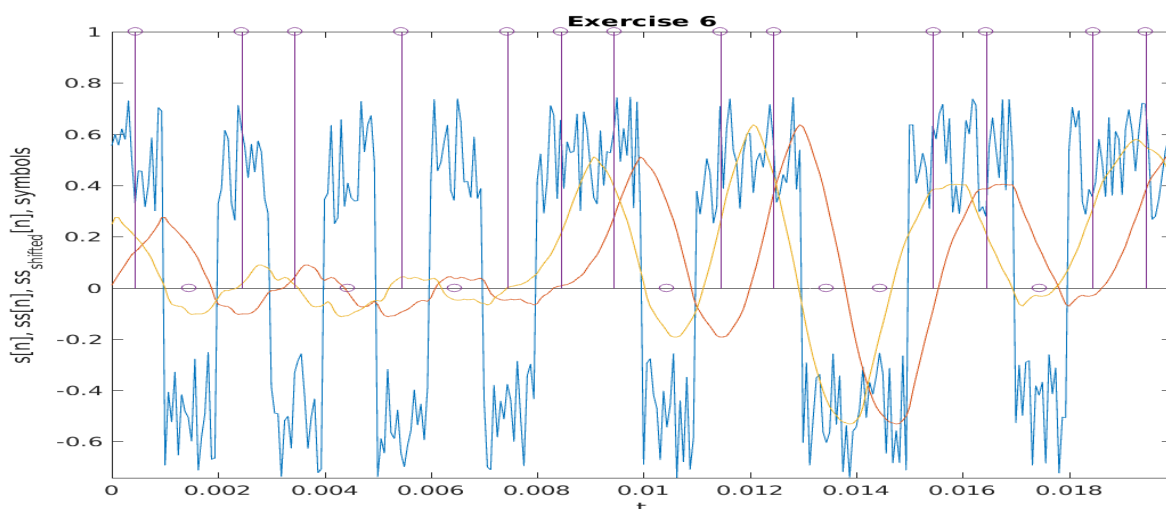
$$H(z) = \frac{0.0192 - 0.0185z^{-1} - 0.0185z^{-2} + 0.0192z^{-3}}{1 - 2.8870z^{-1} + 2.7997z^{-2} - 0.9113z^{-3}}$$

Pokud budete používat Matlab nebo Octave, můžete pro zadání jeho koeficientů jednoduše použít tyto řádky

```
B = [0.0192 -0.0185 -0.0185 0.0192];
A = [1.0000 -2.8870 2.7997 -0.9113];
```

Do protokolu vložte obrázek s nulovými body a póly přenosové funkce tohoto filtru a uveďte, zda je filtr stabilní.

4. [0.5 bodu] Do protokolu vložte obrázek s modulem kmitočtové charakteristiky tohoto filtru (frekvenční osa musí být v Hz a pouze od 0 do poloviny vzorkovací frekvence), uveďte, jakého je filtr typu (dolní propuště / horní propuště / pásmová propuště / pásmová zádrž), a kde leží mezní frekvence v Hz.
5. [1 bod] Filtrujte načtený signál $s[n]$ tímto filtrem. Výsledek budu označovat $ss[n]$. Signály $s[n]$ a $ss[n]$ si zobrazte do jednoho obrázku. Zjistíte, že $ss[n]$ je oproti $s[n]$ nejen zkreslený, ale navíc i posunutý. Najděte, jaký je potřeba posun $ss[n]$, aby se oba signály co nejvíce podobaly. Toto jde udělat vizuálně “od ruky”, ale můžete zapojit i sofistikovanější metody, jako například křížovou korelaci $s[n]$ a $ss[n]$ nebo dokonce minimalizaci výsledné bitové chybovosti (viz níže). Do protokolu napište, o kolik vzorků budete posouvat, kam (zpoždění / předběhnutí) a jak jste k této hodnotě došli.
6. [1 bod] Signál $ss[n]$ skutečně posuňte. Výsledek budu označovat jako $ss_{shifted}[n]$. Proveďte dekódování $ss_{shifted}[n]$ do binárních symbolů. Do protokolu dejte obrázek prvních 20 ms signálů $s[n]$, $ss[n]$, $ss_{shifted}[n]$ a binární symboly dekódované z $ss_{shifted}[n]$, může vypadat např. takto, ale opět se tvořivosti meze nekladou:



7. [1 bod] Vyhodnořte, jakou mají symboly dekódované z $ss_{shifted}[n]$ oproti symbolům dekódovaným z $s[n]$ chybovost (počet špatných bitů lomeno počtem všech bitů, to celé v procentech). Pomoci Vám může funkce XOR. Je možné, že pokud byl $ss_{shifted}[n]$ získán posunem o více než 8 vzorků, bude nutné sekvenci symbolů získanou z $s[n]$ zkrátit, aby Vám jejich počty “seděly”. Do protokolu uveřte počet chyb a chybovost v procentech.
8. [1 bod] Ze signálu $s[n]$ a $ss[n]$ vypočítejte spektra pomocí diskřétní Fourierovy transformace. Do protokolu vložte obrázek modulu obou spekter v závislosti na frekvenci. Frekvenční osa musí být v Hz a pouze od 0 do poloviny vzorkovací frekvence. Komentujte, jak obě spektra vypadají.
9. [1 bod] **V tomto a dalším cvičeních budete pracovat s původním signálem $s[n]$** Odhadněte funkci hustoty rozdělení pravděpodobnosti $p(x)$ signálu $s[n]$. Do protokolu vložte její obrázek, pro jistotu také ověřte, že

$$\int_x p(x)dx = 1$$

10. [0.5 bodu] Spočítejte a do protokolu vložte obrázek korelačních koeficientů $R[k]$ pro $k \in -50 \dots 50$. Použijte vychýlený odhad koeficientů podle vztahu

$$R[k] = \frac{1}{N} \sum_n x[n]x[n+k].$$

(při použití funkce `xcorr` v Matlabu a Octave ‘biased’);

11. [0.5 bodu] Napiřte hodnotu koeficientů $R[0]$, $R[1]$, a $R[16]$. Pokud použijete `xcorr` v Matlabu nebo Octave, prosím zamyslete se, kde je najdete, většinou to není `R(0)` (to ani nejde...), `R(1)` a `R(16)` !
12. [1.5 bodu] Proveřte časový odhad sdružené funkce hustoty rozdělení pravděpodobnosti $p(x_1, x_2, 1)$ mezi vzorky n a $n+1$. Do protokolu vložte 3-D obrázek těchto hodnot. Můžete použít barevnou mapu, odstíny šedi, projekci 3D do 2D, jak chcete. Chcete-li, můžete pro toto a následující dvě cvičení využít či vykuchat dodanou funkci `hist2opt.m`. Funkce ovšem řeší souborový odhad, pro zadaný časový odhad ji musíte modifikovat nebo šikovně zavolat.
13. [1 bod] Ověřte, že se jedná o správnou sdruženou funkci hustoty rozdělení pravděpodobnosti, tedy že

$$\int_{x_1} \int_{x_2} p(x_1, x_2, 1) dx_1 dx_2 = 1$$

14. [1 bod] Vypočteřte z této odhadnuté funkce hustoty rozdělení pravděpodobnosti korelační koeficient $R[1]$:

$$R[1] = \int_{x_1} \int_{x_2} x_1 x_2 p(x_1, x_2, 1) dx_1 dx_2$$

Srovnejte s hodnotou vypočítanou v příkladu 11 a komentujte výsledek.

4 Bonusový úkol

je nepovinný, není hodnocen body, ale nejzajímavější řešenř vyhraje láhev dobrého francouzského červeného vína.

Zadání obsahuje informaci o tom, kolik vzorků je k dispozici na jeden binární symbol: 16. Navrhněte, jak postupovat, kdybyste tuto informaci neměli – někdo Vám dá audio signál $s[n]$ (vzorkovací frekvenci znáte), řekne, že jsou tam asi binární symboly a Vy máte zjistit, kde jsou a jak je dekódovat.

Vyvarujte se prosím triviálních řešenř typu “podívám se na signál, tady to vypadá jako jeden symbol, spočítám, kolik má vzorků, aha, je to 16...”.

Řešenř krátce popiřte a opěř v libovolném jazyce naprogramujte a popiřte výsledky na $s[n]$ a na $ss[n]$. Pokud to na $ss[n]$ nebude fungovat, nezhazujte Vaře řešenř — $ss[n]$ je oproti původnímu opravdu náročnější — ale vysvětlete, proč nefunguje.