Protokol – Projekt ISS 2016/2017
Jiří Matějka (xmatej52)

Datum dokončení: 30. 12. 2016
Dodatečné zdroje: https://www.mathworks.com/help/matlab/

[1] : Načtěte Váš osobní signál ze souboru xlogin00.wav, kde “xlogin00” je Váš login. Napište vzorkovací frekvenci signálu a jeho délku ve vzorcích a v sekundách.

[y,Fs] = audioread('xmatej52.wav');
V y mam nyní uložený signál, ve Fs vzorkovací frekvenci.

N = numel(y);
V N mám nyní uloženou délku signálu ve vzorcích.

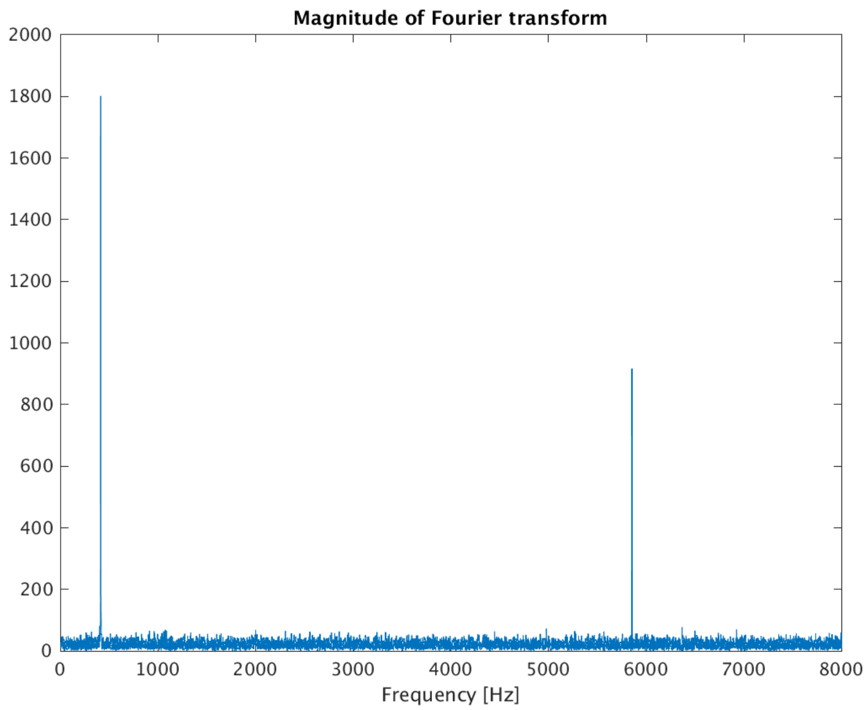
t = N ./ Fs;
V t mám nyní uložen čas.

Vzorkovací frekvence : 16 000 Hz
Délka ve vzorcích : 16 000 vzorků
Délka v sekundách : 1 sekunda

[2]: Vypočítejte spektrum signálu pomocí diskrétní Fourierovy transformace. Do protokolu vložte obrázek modulu spektra v závislosti na frekvenci. Frekvenční osa musí být v Hz a pouze od 0 do poloviny vzorkovací frekvence.
ydft = fft(y);
V ydtf mám nyní uloženu Fourirerovu transformaci.

m = abs(ydft);
V m mám nyní uloženou absolutní hodnotu Fourirerovy transformace, zbývá pouze vytisknout.

freq = linspace(0, Fs/2-1, Fs/2);
plot(freq,m(1:Fs/2));
title('Magnitude of Fourier transform');
xlabel 'Frequency [Hz]';



[3] Určete a napište, na které frekvenci v Hz je maximum modulu spektra.

idx = find(m == max(m));
Nalezne index v poli m, na kterém leží maximum.

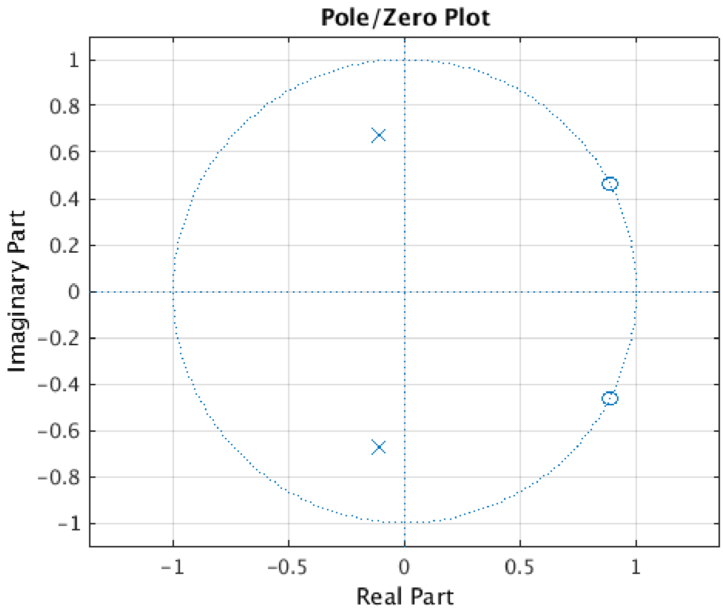
Maximum (1.8007e+03) leží na frekvenci 414 Hz.

[4] Pro další zpracování je dán IIR filtr s následujícími koeficienty:

b0 = 0.2324, b1 = −0.4112, b2 = 0.2324, a1 = 0.2289, a2 = 0.4662

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

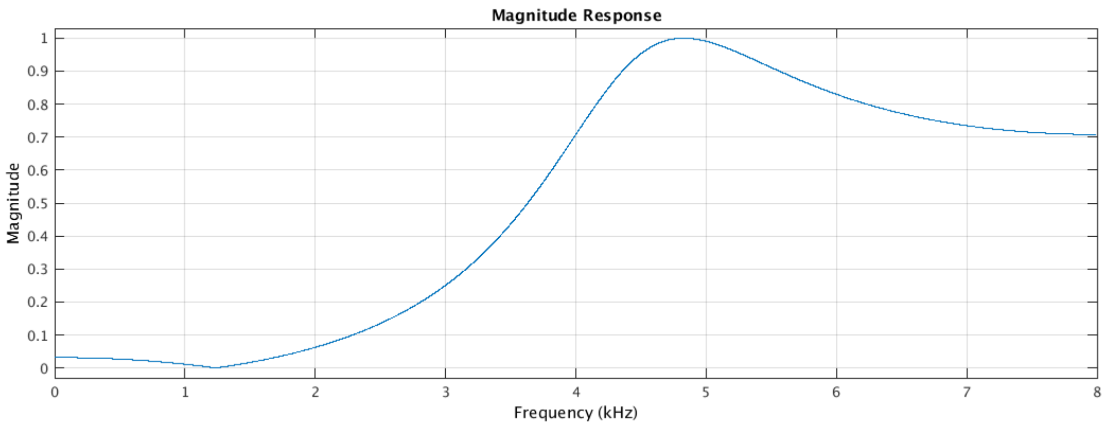
Do pole a a b uložíme koeficienty a provedeme následující příkaz:
fvtool(b,a,'polezero');



Filtr je stabilní – všechny póly leží uvnitř jednotkové kružnice. Stabilitu lze ověřit i použitím funkce isstable.

[5] Do protokolu vložte obrázek s modulem kmitočtové charakteristiky tohoto filtru (frekvenční osa musí být v Hz a pouze od 0 do poloviny vzorkovací frekvence) a uveďte, jakého je filtr typu (dolní propusť / horní propusť / pásmová propusť / pásmová zádrž).

Provedeme následující příkaz:
fvtool(b,a,'polezero');

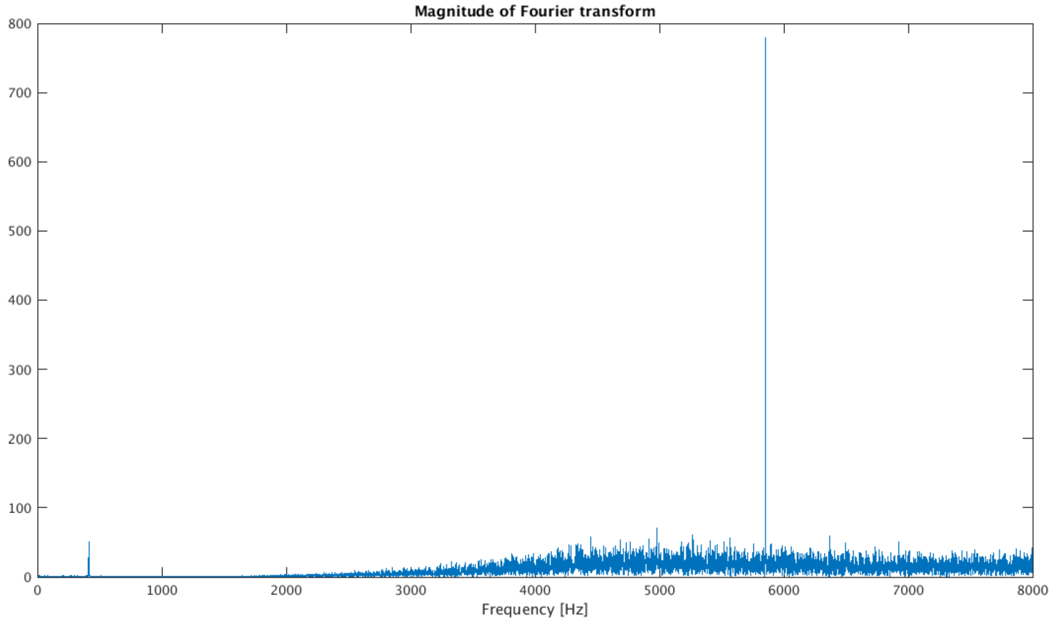


Filtr je typu horní propusť (nízké frekvence výrazně tlumí, lze to poznat i z tohoto grafu)

[6] Filtrujte načtený signál tímto filtrem. Z výsledného signálu vypočítejte spektrum signálu pomocí diskrétní Fourierovy transformace. Do protokolu vložte obrázek modulu spektra v závislosti na frekvenci. Frekvenční osa musí být v Hz a pouze od 0 do poloviny vzorkovací frekvence.

Do x uložíme signál po projití filtrem
x = filter(b, a, y);

Do xdft uložíme Fourierovu transformaci a následně do xm uložíme její absolutní hodnotu a vykreslíme (stejně jako v příkladu č. 1)
xdft = fft(x);
xm = abs(xdft);
plot(freq, xm(1:Fs/2));
title('Magnitude of Fourier transform');
xlabel 'Frequency [Hz]';



[7] Určete a napište, na které frekvenci v Hz je maximum modulu spektra filtrovaného signálu.

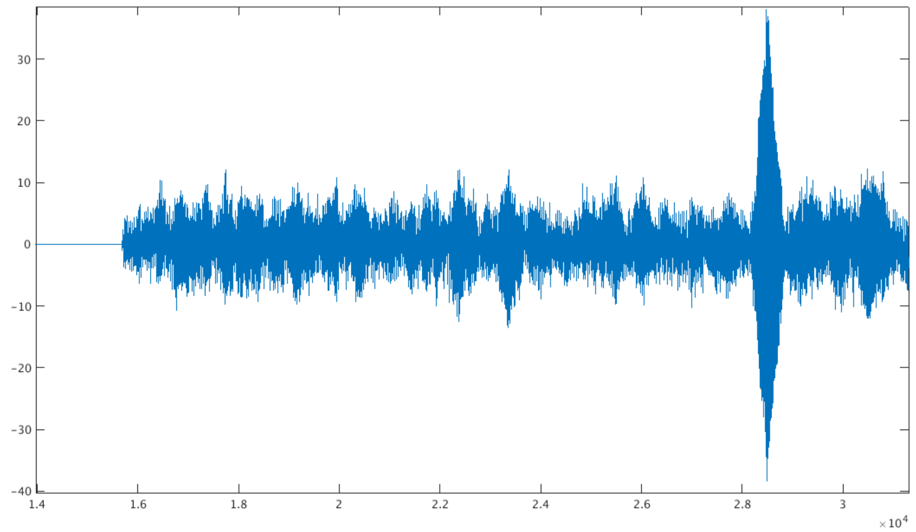
xidx = find(xm == max(xm));
najdeme index v poli xm, na kterém se nachází maximum.

Maximum (779.6024) je na frekvenci 5854.

[8] Do signálu bylo přimícháno 20 ms obdélníkových impulsů se střední hodnotou nula a s třídou 50% na frekvenci 2 kHz. Tedy 40 sekvencí [h h h h −h −h −h −h] (kde h je kladné číslo) za sebou. Najděte, kde jsou, a napište čas ve vzorcích a v sekundách. Pomůcka: pokud netušíte, jak na to, uvažte např. přizpůsobený filtr, výrobu spektra této sekvence a jeho hledání ve spektru signálu rozděleného po 20 ms, poslech, atd. Cílem není matematická čistota, ale vyřešení úkolu.

Vygenerujeme obdélníkový signál a pomocí korelace zjistíme podobnost. V místech, kde jsou korelační koeficienty nejvyšší, jsou si signály nejvíce podobné a tam se nachází hledané impulsy.
*new_signal = square(0:pi/4:80*pi);*
new_signal = new_signal(1:(length(new_signal) - 1));
plot(xcorr(y, new_signal));

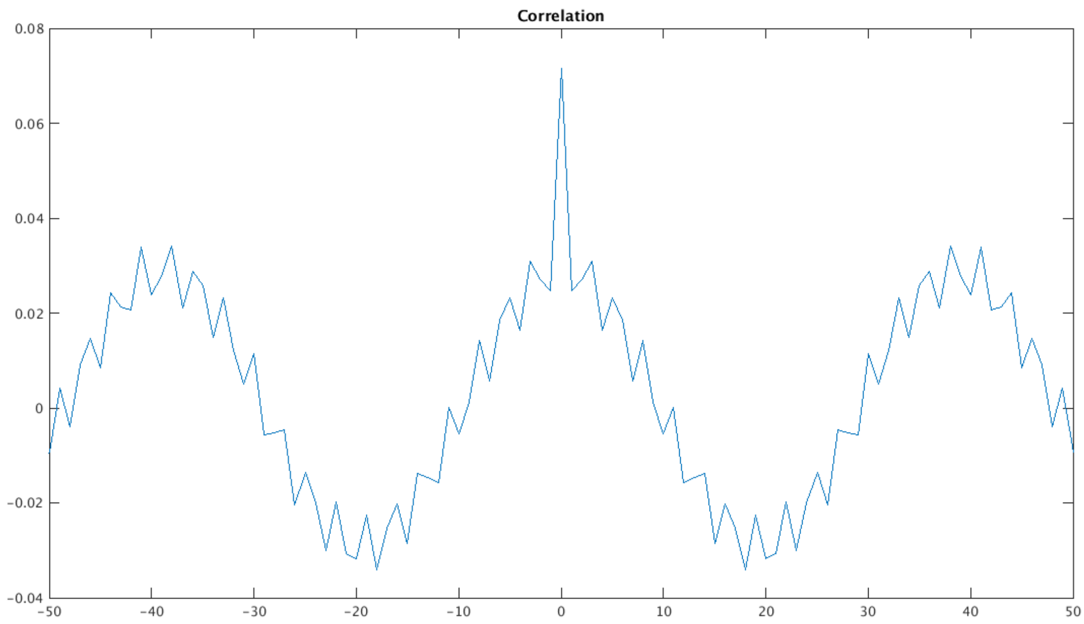
Impulsy začínají přibližně na vzorku 14100, který odpovídá času 0.88125 sekundy. (Zjištěno po přiblížení následujícího obrázku)



[9] Spočítejte a do protokolu vložte obrázek autokorelačních koeficientů R[k] pro k ∈ −50 . . . 50. Použijte vychýlený odhad koeficientů podle vztahu

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

Do proměnné R uložíme autokorelační koeficienty a následně graf vykreslíme
[R, lag] = xcorr(y, 'biased');
plot(lag, R);
title('Correlation');
xlim([-50 50]);

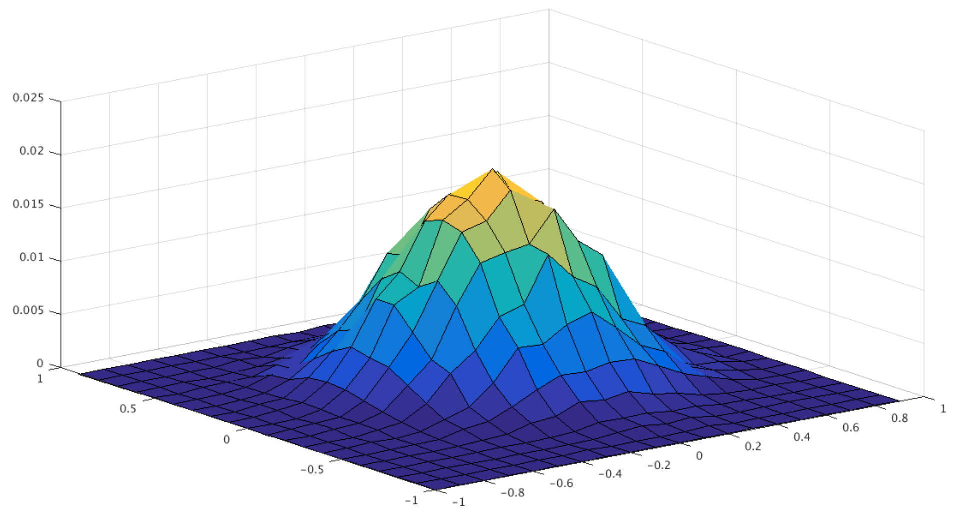


[10] Napište hodnotu koeficientu R[10].

Nalezneme bod 0 v poli R a následně přičteme 10 a tak získáme index v poli, kde je uložena hodnota.
ridx = find(R == max(R));
R10 = R(ridx + 10);

R[10] = -0.0054

[11] Proveďte časový odhad sdružené funkce hustoty rozdělení pravděpodobnosti p(x1, x2, 10) mezi vzorky n a n + 10. Do protokolu vložte 3-D obrázek těchto hodnot. Můžete použít barevnou mapu, odstíny šedi, projekci 3D do 2D, jak chcete. Chcete-li, můžete využít dodanou funkci hist2opt.m



Vytvoříme si několik matic a pomocí for cyklu vytvoříme graf funkce hustoty rozdělení pravděpodobnosti. Hodnoty

```
graph = zeros(20);
vector = zeros(10, 1);
y_x = cat(1, vector, y) * 10;
y_y = cat(1, y, vector) * 10;
vec_x = 0;
vec_y = 0;
for i = 1:(N+10)
  vec_x = round(y_x(i)) + 11;
  vec_y = round(y_y(i)) + 11;
  graph(vec_y, vec_x) = graph(vec_y, vec_x) + 1;
end
graph = graph/(N+10);
surf(-1:0.1:0.9, -1:0.1:0.9, graph);
```

[12] Ověřte, že se jedná o správnou sdruženou funkci hustoty rozdělení pravděpodobnosti, tedy že:

$$\int_{x_1}\int_{x_2} p(x_1, x_2, 10)dx_1dx_2 = 1$$

Pracujeme se vzorky, tak místo integrálu použijeme sumu.

```
check = sum(sum(graph, 2));
```

Ano, rovnice platí.

[13] Vypočtěte z této odhadnuté funkce hustoty rozdělení pravděpodobnosti autokorelační koeficient R[10]:

$$R[10] = \int_{x_1}\int_{x_2} x_1x_2p(x_1, x_2, 10)dx_1dx_2$$

Srovnejte s hodnotou vypočítanou v příkladu 10 a komentujte výsledek.

Pomocí funkce *meshgrid* vygenerujeme 2 matice, a pronásobíme matici graph (výsledek příkladu 11) se součinem vygenerovaných matic.

```
[x_matrix, y_matrix] = meshgrid(-1:0.1:0.9, -1:0.1:0.9);
check = sum(sum(graph .* (x_matrix .* y_matrix), 2));
```

Výsledek je -0.0056, což téměř odpovídá výsledku v příkladu 10 (-0.0054). Výchylku mohlo způsobit zaokrouhlování, které jsem použil v příkladu 11.