

Protokol ISS – 1. Projekt

1. Náš signál načteme příkazem *audioread* následovně:

```
[y, Fs] = audioread('xtesar36.wav');
```

V *y* máme uložený signál a *Fs* je vzorkovací frekvence. Na délku ve vzorcích můžeme použít např. funkci *numel(y)* nebo *length(y)* a čas vypočítáme podílem délky ve vzorcích a *Fs*.

Vzorkovací frekvence: 16 000 Hz

Délka ve vzorcích: 16 000 vzorků

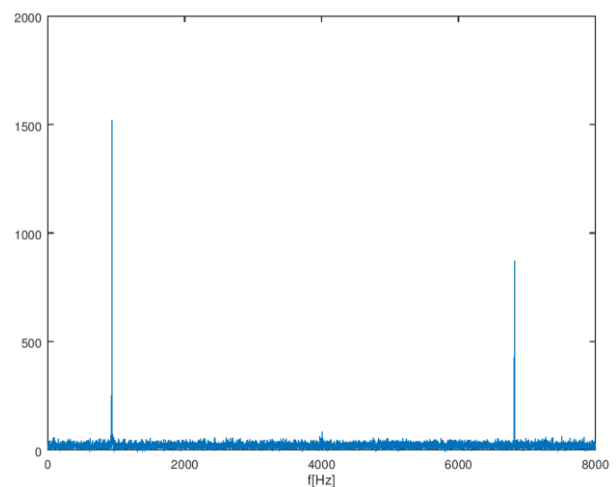
Délka v sekundách: 1 sekunda

2. Nejprve dostaneme Fourierovu transformaci funkcí *fft(y)*, která nám výsledné pole uloží do proměnné *fou*. Chceme pouze absolutní hodnoty, proto funkce *abs*.

Zobrazujeme od 0 do $F_s/2-1$ a chceme $F_s/2$ vzorků (*linspace*).

```
fou = fft(y);
absfou = abs(fou);
f = linspace(0, Fs/2-1, Fs/2);
plot(f, absfou(1:Fs/2));
xlabel 'f[Hz]';
```

Modul spektra FT



3. Maximální modul spektra je na frekvenci 930 Hz odhadem z grafu. Přesný výsledek vypočítáme následovně:

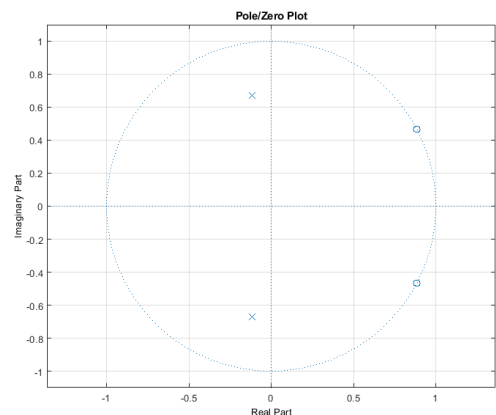
```
[maxf, max_index] = max(absfou);
```

max_index = 936 Hz

4. Koeficienty si uložíme do dvou polí *a*, *b*. Funkcí *fvtool* zobrazíme odezvu filtru.

```
a = [1, 0.2289, 0.4662];
b = [0.2324, -0.4112, 0.2324];
fvtool(b,a,'polezero');
```

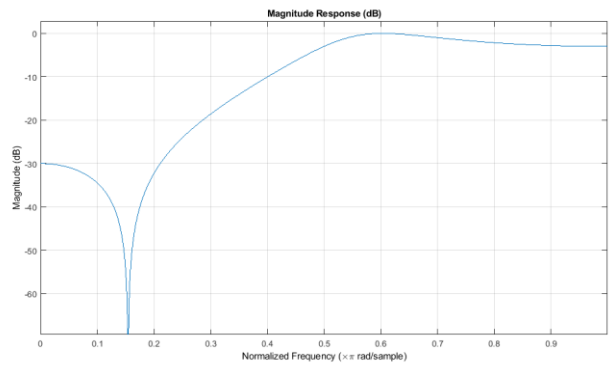
Z obrázku vidíme, že filtr je **stabilní**. Póly a nuly leží v jednotkové kružnici.



5. Kmitočtovou charakteristiku dostaneme stejnou funkcí *fvtool* s možností 'magnitude'.

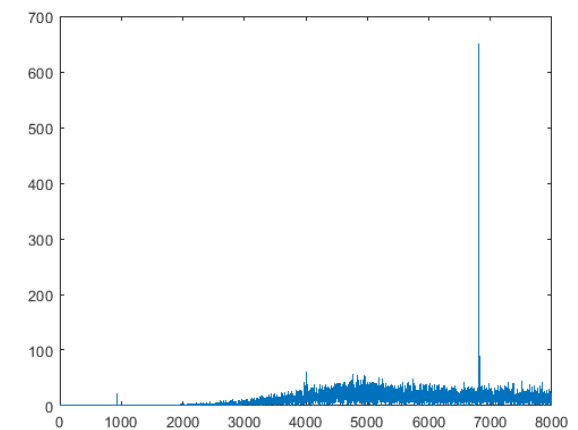
```
fvtool(b,a,'magnitude');
```

Jedná se o typickou **horní** propust. Filtř nepropouští (**tumí**) signál o nízkých frekvencích.



6. Na filtrování slouží funkce *filter*, která nám vyfiltruje vstupní signál *y*. Postupujeme jak v příkladu 2.

```
filtr = filter(b,a,y);  
fourier = fft(filtr);  
absfourier = abs(fourier);  
plot( f, absfourier(1:Fs/2) );
```



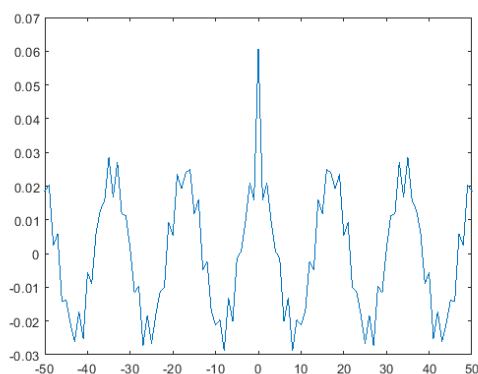
7. Hledáme maximum spektra filtrovaného signálu:

```
[module, index_max] = max(absfourier);  
index_max = 6819 Hz
```

Maximální frekvence je **6819** Hz filtrovaného signálu.

8. -
9. Uložíme si autokorelační koeficienty do proměnné *kor* a graf vykreslíme. Limity na ose *x* nastavíme funkcí *xlim*.

```
[kor, lag] = xcorr(y, 'biased');  
plot(lag, kor);  
xlim( [ -50 50 ] );
```



10. Napište hodnotu koeficientu $R[10]$.

Vypočítáme, kde se nachází index nula, ze znalosti, že se jedná o maximální hodnotu v poli. Následně k vypočítanému indexu přičteme 10.

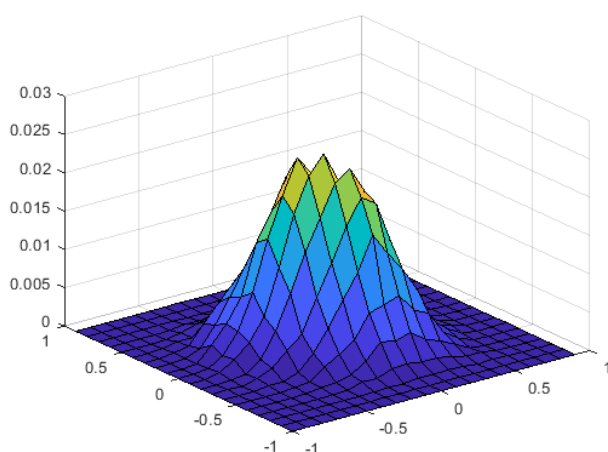
```
index_nula = find(kor == max(kor))
R10 = kor(index_nula+10);
```

$R[10] = -0.0212$

11. Nejprve si nachystáme matici inicializovanou na nuly o rozměru 20 x 20 (jelikož chceme mít jeden krok délky 0.1), do které budeme ukládat výsledky. Budeme mít dva pomocné vektory, které budou kopíř původního signálu s tím, že jeden bude posunutý o 10 vzorků doprava. Následně projedeme všechny vzorky + 10 a počítáme výskyty na indexech (vždy +1). Výslednou funkci hustoty rozdělení pravděpodobnosti dostaneme vydělením všech prvků v matici počtem realizací.

```
matice = zeros(20); % vytvorime matici nul, do kt. budeme ukladat pocy
signal_nuly_zac = cat(1, zeros(10, 1), y); % konkatenace nul a puvodniho
signalu nakonec
signal_nuly_konec = cat(1, y, zeros(10, 1)); % konkatenace nul na zacatek
signal_nuly_konec = signal_nuly_konec * 10;
signal_nuly_zac = signal_nuly_zac * 10;
%indexy
puvodni_vzorek = 0;
posunuty_vzorek = 0;
for i = 1:(vzorky+10) % projizdim od 1 do 16010
    puvodni_vzorek = round(signal_nuly_konec(i)) + 10; % puvodni
    nepusunuty vzorek
    posunuty_vzorek = round(signal_nuly_zac(i)) + 10; % posunuty vzorek
    matice(posunuty_vzorek, puvodni_vzorek) = matice(posunuty_vzorek,
    puvodni_vzorek) + 1; % inkrementujeme
end

matice = matice/(vzorky+10); % delime poctem realizaci
surf(-1:0.1:0.9, -1:0.1:0.9, matice); % krok = 0.1
```



12. Sečteme všechny hodnoty v matici, použijeme 2x funkci sum (jednou sčítáme řádky a pak sloupce).

```
S = sum(sum(matice, 2));
```