

# ISS Projekt 2023/24

Honza Pavlus, Honza Brukner a Honza Černocký, ÚPGM FIT VUT

6.11. 2023

## 1. Úvod

V projektu budete pracovat se biomedicínskými signály a to konkrétně se signálem elektrokardiogramu EKG. Vyzkoušíte si filtraci tohoto druhu signálu, abyste dostali krásné EKG křivky, které můžete vidět ve filmech. Dále si zkusíte vybudovat jednoduchý, ale účinný detektor QRS a ti, kteří se vrhnou i na bonusový úkol, si zkusí odhalit srdeční patologii. K dispozici dostanete každý 3 nahrávky jednobáňového EKG signálu, jeden zdravý a dva s různými patologiemi.

Projekt je nejlépe možno řešit v Python-u a to přímo v dodaném Python notebooku, který si můžete zkopírovat do vlastního Google Colabu. Projekt je také možno řešit v Matlab-u, Octave, Julii, jazyce C, Java nebo v libovolném jiném programovacím či skriptovacím jazyce. Je možné použít libovolné knihovny. Projekt se nezaměřuje na "krásu programování", není tedy nutné mít vše úhledně zabalené do okomentovaných funkcí (samozřejmě se ale okomentovaný kód lépe opravuje a to hlavně v případě podivných výsledků), ošetřené všechny chybové stavy, atd. Důležitý je výsledek.

**Vaši práci odevzdáváte vyexportovanou do dvou souborů: (1) do PDF souboru login.pdf, (2) do Python notebooku login.ipynb. PDF musí obsahovat výsledky prokazatelně vytvořené Vaším kódem.** V případě řešení projektu v jiném jazyce nebo prostředí než v dodaném Python notebooku, je prvním souborem protokol v PDF, druhý soubor je archiv s Vaším kódem. Ten musí být spustitelný na standardní fakultní distribuci Windows nebo Linuxu.

## 3. Vstup

Pro řešení projektu má každý student/ka k dispozici osobní soubor se zdravým signálem (sinusovým rytmem): **login.wav**, kde login je váš xlogin popřípadě VUT číslo (pro studenty FSI). Dále jsou k dispozici ještě další dva signály: **FIS.wav** a **KES.wav**. První signál obsahuje fibrilaci a druhý komorovou extrasystolu. Tyhle dva soubory jsou pro všechny společné a využijete je při řešení bonusového úkolu.

```
#Načtení Vašeho signálu - xlogin99 nahraďte Vaším loginem
import soundfile as sf
import matplotlib.pyplot as plt
import numpy as np
from scipy.signal import resample

!wget https://www.fit.vutbr.cz/study/courses/ISS/public/proj2023-
24/xmicha94.wav
!wget https://www.fit.vutbr.cz/study/courses/ISS/public/proj2023-
24/FIB.wav
```

```
!wget https://www.fit.vutbr.cz/study/courses/ISS/public/proj2023-24/KES.wav
```

```
x, fs = sf.read("xmicha94.wav")
```

```
--2023-12-17 21:20:34--
```

```
https://www.fit.vutbr.cz/study/courses/ISS/public/proj2023-24/xmicha94.wav
```

```
Resolving www.fit.vutbr.cz (www.fit.vutbr.cz)... 147.229.9.23, 2001:67c:1220:809::93e5:917
```

```
Connecting to www.fit.vutbr.cz (www.fit.vutbr.cz)|147.229.9.23|:443... connected.
```

```
HTTP request sent, awaiting response... 200 OK
```

```
Length: 10044 (9.8K) [audio/x-wav]
```

```
Saving to: 'xmicha94.wav.1'
```

```
xmicha94.wav.1      100%[=====>]      9.81K  --.-KB/s    in 0s
```

```
2023-12-17 21:20:36 (148 MB/s) - 'xmicha94.wav.1' saved [10044/10044]
```

```
--2023-12-17 21:20:36--
```

```
https://www.fit.vutbr.cz/study/courses/ISS/public/proj2023-24/FIB.wav
```

```
Resolving www.fit.vutbr.cz (www.fit.vutbr.cz)... 147.229.9.23, 2001:67c:1220:809::93e5:917
```

```
Connecting to www.fit.vutbr.cz (www.fit.vutbr.cz)|147.229.9.23|:443... connected.
```

```
HTTP request sent, awaiting response... 200 OK
```

```
Length: 10044 (9.8K) [audio/x-wav]
```

```
Saving to: 'FIB.wav.1'
```

```
FIB.wav.1          100%[=====>]      9.81K  --.-KB/s    in 0s
```

```
2023-12-17 21:20:37 (129 MB/s) - 'FIB.wav.1' saved [10044/10044]
```

```
--2023-12-17 21:20:37--
```

```
https://www.fit.vutbr.cz/study/courses/ISS/public/proj2023-24/KES.wav
```

```
Resolving www.fit.vutbr.cz (www.fit.vutbr.cz)... 147.229.9.23, 2001:67c:1220:809::93e5:917
```

```
Connecting to www.fit.vutbr.cz (www.fit.vutbr.cz)|147.229.9.23|:443... connected.
```

```
HTTP request sent, awaiting response... 200 OK
```

```
Length: 10044 (9.8K) [audio/x-wav]
```

```
Saving to: 'KES.wav.1'
```

```
KES.wav.1          100%[=====>]      9.81K  --.-KB/s    in 0s
```

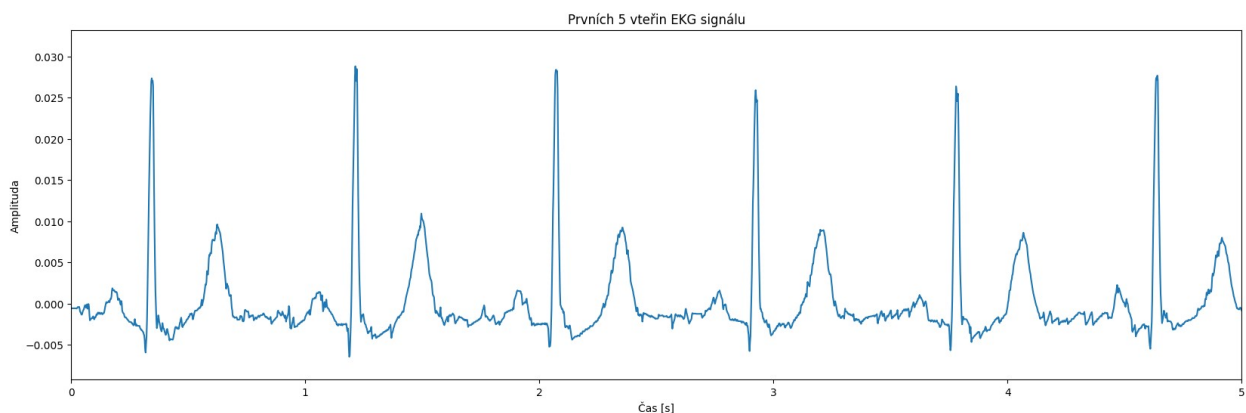
## 4. Úkoly

### 4.1. [2.5b] Nahrání a zobrazení EKG signálu

Nezapomeňte na popisy os u jednotlivých grafů.

a) [1b] Nahrajte EKG signál login.wav, vyberte 5-sekundový úsek a zobrazte ho v časové doméně. Pro nahrání signálu použijte knihovny numpy a soundfile.

```
time = np.arange(0, len(x)) / fs
plt.figure(figsize=(20,6))
plt.plot(time, x)
plt.title('Prvních 5 vteřin EKG signálu')
plt.xlabel('Čas [s]')
plt.ylabel('Amplituda')
plt.xlim(0, 5)
plt.show()
```



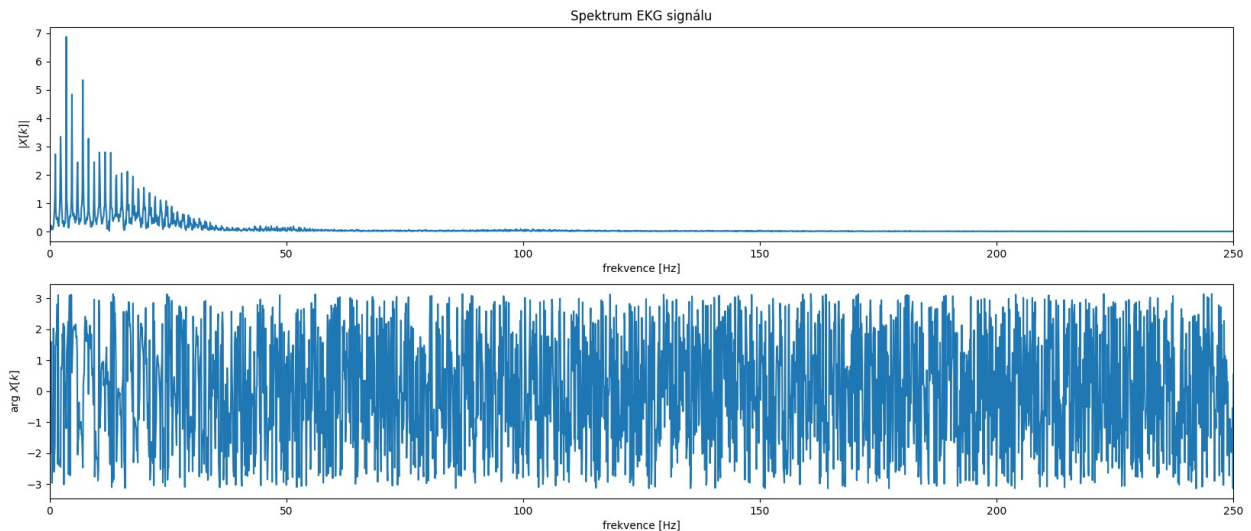
b) [1b] Spočítejte spektrum z 5 sekundového úseku nahraného signálu a zobrazte jej.

```
X = np.fft.fft(x)
kall = np.arange(0, len(x)) / 10 # ?????????????????
Xmag = np.abs(X)
Xphase = np.angle(X)

_, ax = plt.subplots(2,1, figsize=(20,8))
ax[0].set_xlim(0, 250)
ax[0].plot(kall, Xmag)
ax[0].set_title('Spektrum EKG signálu')
ax[0].set_ylabel('$|X[k]|$')
ax[0].set_xlabel('frekvence [Hz]')
ax[1].set_xlim(0, 250)
ax[1].plot(kall, Xphase)
```

```
ax[1].set_ylabel('arg $X[k]$')
ax[1].set_xlabel('frekvence [Hz]')

Text(0.5, 0, 'frekvence [Hz]')
```



c) [0.5b] Ve spektru vidíte rušení na 50Hz nebo 60Hz a jejich harmonických frekvencích. Vysvětlíte, čím je způsobeno.

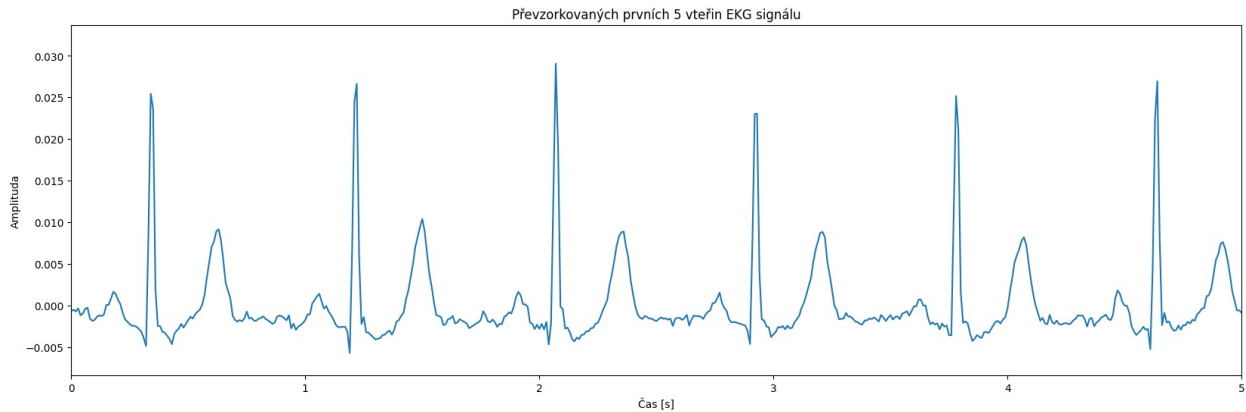
Rušení na 50 a 60 Hz a jejich harmonických frekvencí může být způsobeno elektrickým šumem nebo interferencí ze sítě střídavého proudu.

## 4.2. [3b] Převzorkujte nahraný signál

a) [2b] Převzorkujte signál na vzorkovací frekvenci 100 Hz, nezapomeňte na filtr pro antialiasing.

Můžete například odstranit část spektra od  $\frac{F_s}{2}$  nebo použít filtr dolní propusti.

```
resampling_frequency = 100
Hz100_x = resample(x, int(len(x) * (resampling_frequency / fs)))
new_frequency = np.arange(0, len(Hz100_x)) * (resampling_frequency / fs)
new_time = np.arange(0, len(Hz100_x)) / resampling_frequency
plt.figure(figsize=(20,6))
plt.title('Převzorkovaných prvních 5 vteřin EKG signálu')
plt.xlabel('Čas [s]')
plt.ylabel('Amplituda')
plt.xlim(0,5)
plt.plot(new_time, Hz100_x)
plt.show()
```



b) [1b] Zobrazte 5 sekundový úsek původního a převzorkovaného signálu v časové doméně a zobrazte i jejich spektra.

```
_, ax = plt.subplots(3,2, figsize=(30,10), sharex=False,
gridspec_kw={'hspace': 0.5, 'wspace': 0.1})
ax[0, 0].plot(time, x)
ax[0, 0].set_title('Prvních 5 vteřin EKG signálu')
ax[0, 0].set_xlabel('Čas [s]')
ax[0, 0].set_ylabel('Amplituda')
ax[0, 0].set_xlim(0, 5)

ax[1, 0].plot(kall,Xmag)
ax[1, 0].set_title('Spektrum původního EKG signálu')
ax[1, 0].set_ylabel('$|X[k]|$')
ax[1, 0].set_xlabel('frekvence [Hz]')

ax[2, 0].plot(kall,Xphase)
ax[2, 0].set_ylabel('arg $X[k]$')
ax[2, 0].set_xlabel('frekvence [Hz]')

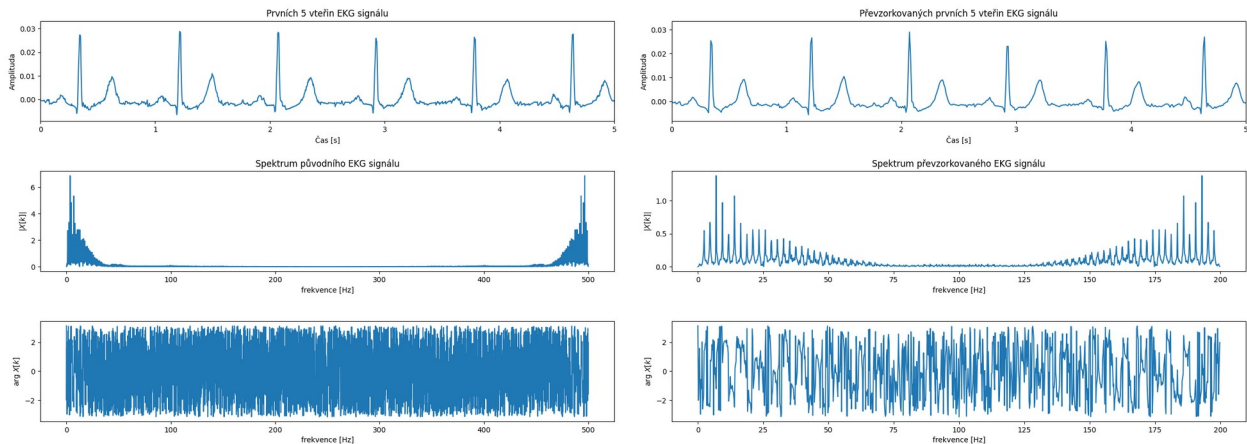
ax[0, 1].set_xlim(0, 5)
ax[0, 1].set_title('Převzorkovaných prvních 5 vteřin EKG signálu')
ax[0, 1].set_xlabel('Čas [s]')
ax[0, 1].set_ylabel('Amplituda')
ax[0, 1].plot(new_time, Hz100_x)

Hz100_X = np.fft.fft(Hz100_x)
Hz100_Xmag = np.abs(Hz100_X)
Hz100_Xphase = np.angle(Hz100_X)

ax[1, 1].set_title('Spektrum převzorkovaného EKG signálu')
ax[1, 1].set_ylabel('$|X[k]|$')
ax[1, 1].set_xlabel('frekvence [Hz]')
ax[1, 1].plot(new_frequency, Hz100_Xmag)

ax[2, 1].plot(new_frequency, Hz100_Xphase)
```

```
ax[2, 1].set_ylabel('arg $X[k]$')
ax[2, 1].set_xlabel('frekvence [Hz]')
Text(0.5, 0, 'frekvence [Hz]')
```



### 4.3. [4b] Vyfiltrujte nahraný signál pásmovou propustí 10Hz-20Hz

a) [2b] Vytvořte filtr pásmové propusti, možnosti jsou dvě: buďto filtrovat pomocí klasického návrhu filtrů, kde získáte koeficienty  $a$  a  $b$  (pomocí např. `scipy.butter`) a zobrazíte charakteristiku filtru + nuly a póly. Nebo se můžete vydat cestou filtrování ve frekvenční doméně, frekvenční charakteristiku vykreslete pomocí spektrální masky.

```
from scipy.signal import butter, lfilter, freqz, zpk2tf, tf2zpk

spodni_hranice = 10
horni_hranice = 20
order = 4
b, a = butter(order, [spodni_hranice, horni_hranice], btype='band',
               fs=fs)

w, h = freqz(b, a, worN=8000)

z, p, k = tf2zpk(b, a)

plt.figure(figsize=(4.5, 8))

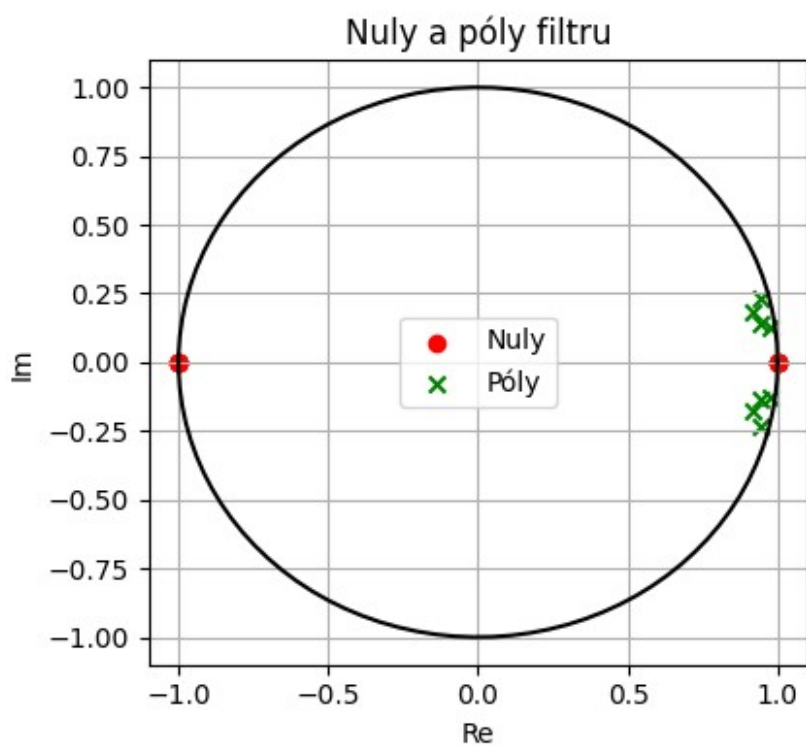
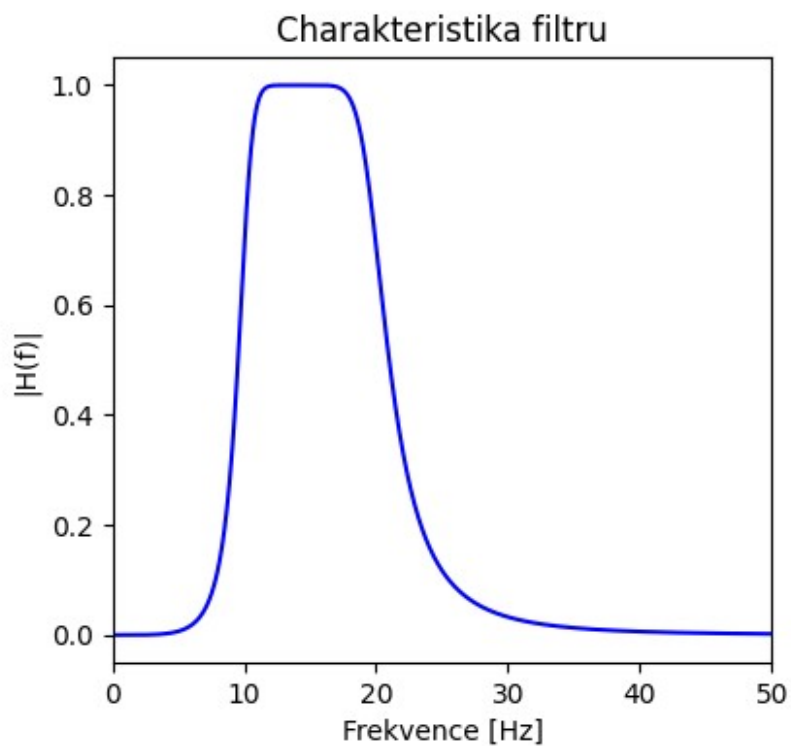
plt.subplot(2, 1, 1)
plt.xlim(0, 50)
plt.plot(0.5 * fs * w/np.pi, np.abs(h), 'b')
plt.title('Charakteristika filtru')
plt.xlabel('Frekvence [Hz]')
plt.ylabel('|H(f)|')

plt.subplot(2, 1, 2)
plt.scatter(np.real(z), np.imag(z), marker='o', color='r',
            label='Nuly')
```

```
plt.scatter(np.real(p), np.imag(p), marker='x', color='g',
label='Póly')
plt.title('Nuly a póly filtru')
plt.xlabel('Re')
plt.ylabel('Im')
plt.legend()

theta = np.linspace(0, 2*np.pi, 100)
plt.plot(np.cos(theta), np.sin(theta), linestyle='solid',
color='black', label='Kružnice')
plt.grid(True)

plt.tight_layout()
plt.show()
```



b) [1b] Použijte navržený filtr na nahraný signál. Pokud máte navržený klasický filtr, proveďte filtrování z obou stran, abyste se vyhnuli fázovému posunu, to za vás zajistí například funkce `scipy.signal.filtfilt`. Vykreslete původní a vyfiltrovaný signál v časové doméně a spočítejte a zobrazte jejich spektra.



```

_, ax = plt.subplots(3,2, figsize=(30,10), sharex=False,
gridspec_kw={'hspace': 0.5, 'wspace': 0.1})
ax[0, 0].plot(time, x)
ax[0, 0].set_title('Prvních 5 vteřin EKG signálu')
ax[0, 0].set_xlabel('Čas [s]')
ax[0, 0].set_ylabel('Amplituda')
ax[0, 0].set_xlim(0, 5)

ax[1, 0].plot(kall,Xmag)
ax[1, 0].set_title('Spektrum původního EKG signálu')
ax[1, 0].set_ylabel('$|X[k]|$')
ax[1, 0].set_xlabel('frekvence [Hz]')

ax[2, 0].plot(kall,Xphase)
ax[2, 0].set_ylabel('arg $X[k]$')
ax[2, 0].set_xlabel('frekvence [Hz]')

filtered_signal = lfilter(b, a, x)

ax[0, 1].set_xlim(0, 5)
ax[0, 1].set_title('Přefiltrovaných prvních 5 vteřin EKG signálu')
ax[0, 1].set_xlabel('Čas [s]')
ax[0, 1].set_ylabel('Amplituda')
ax[0, 1].plot(time, filtered_signal)

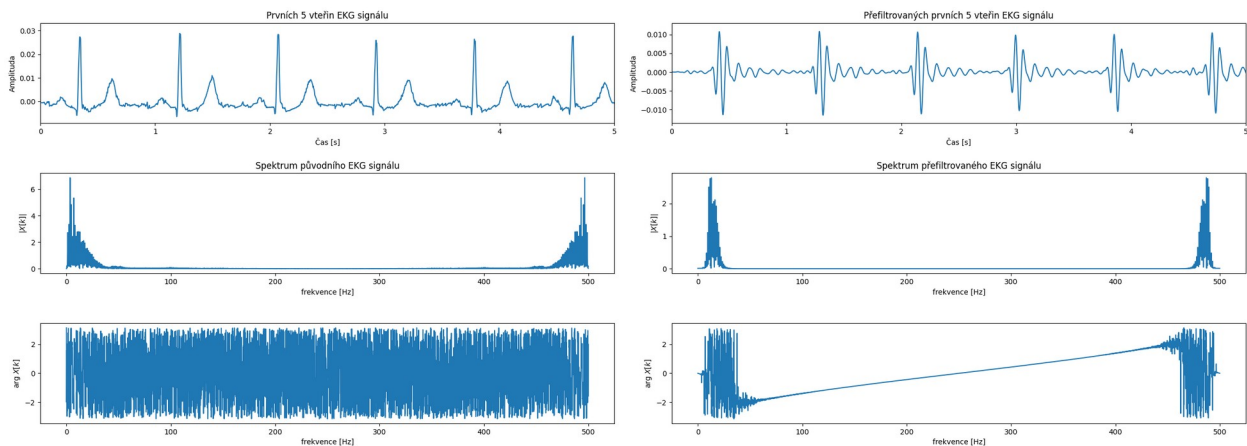
filtered_X = np.fft.fft(filtered_signal)
filtered_Xmag = np.abs(filtered_X)
filtered_Xphase = np.angle(filtered_X)

ax[1, 1].set_title('Spektrum přefiltrovaného EKG signálu')
ax[1, 1].set_ylabel('$|X[k]|$')
ax[1, 1].set_xlabel('frekvence [Hz]')
ax[1, 1].plot(kall,filtered_Xmag)

ax[2, 1].plot(kall,filtered_Xphase)
ax[2, 1].set_ylabel('arg $X[k]$')
ax[2, 1].set_xlabel('frekvence [Hz]')

Text(0.5, 0, 'frekvence [Hz]')

```



c) [1b] Okomentujte rozdíl mezi filtrovaným a nefiltrovaným signálem a jejich spektry. Pokud bychom použili filtrování pouze z jedné strany (obyčejnou konvoluci), jaké je teoreticky největší posunutí ve vzorcích, které se může objevit a proč?

Nefiltrovaný signál obsahuje všechny frekvence, stejně tak šum a různé rušení. Filtrovaný signál obsahuje pouze frekvence v požadovaném pásmu, ostatní frekvence i šum můžou být odstraněny.

a) [1b] Detekujte QRS v převzorkovaném vyfiltrovaném signálu pomocí thresholdu (prahu). Pro tuto detekci musíte nejdříve získat vzorek jednoho QRS ze signálu, spočítat si maximální amplitudu a jako threshold vzít vámi určené procento této hodnoty. **Dávejte pozor na možnost otočeného QRS v signálu.** Do vykresleného signálu s detekcemi vykreslete i čáru udávající použitý threshold.

# Zde napište váš kód

b) [2b] Detekujte QRS v signálu pomocí autokorelace v převzorkovaném nefiltrovaném signálu. Pro tuto detekci musíte nejdříve získat vzorek jednoho QRS ze signálu. Dále budete autokorelovat signál právě s tímto výstřížkem. QRS se budou nacházet na místech, kde vám budou vycházet vysoké hodnoty korelace. Do vykresleného signálu s detekcemi zaznačte i vámi zvolený výstřížek.

# Zde napište váš kód

#### 4.5. [3.5b] Vytvořte detektor QRS v frekvenční doméně a detekované QRS zakreslete jako v předchozí úloze 4.4

a) [2b] Detekujte QRS pomocí použití spektrogramu. Spočítejte a zobrazte spektrogram nahraného převzorkovaného filtrovaného signálu. Použijte parametry, `hop_size=120ms` a `window_len=200ms`, popřípadě si zkuste s těmito parametry pohrát. Spektrogram dále normalizujte v čase. Spočítejte sumy energie spektra pro jednotlivé časové biny. Dále vytvořte práh podle hodnoty energie spektra u prvního vámi zvoleného QRS komplexu. Tento práh použijte pro detekci zbylých QRS komplexů v signálu.

# Zde napište váš kód

b) [1b] Detekujte QRS pomocí použití obálek a Hilbertovy transformace.

Hilbertova transformace je spočítaná podle následujícího vzorce

$$x_a = F^{-1} \left( F(x) 2U \right) = x + i y,$$

kde  $F$  je Fourierova transformace a  $F^{-1}$  je její zpětná varianta.  $U$  je Heavisideova funkce neboli funkce jednotkového skoku, která je definována:  $U(x)$ :

$$\begin{cases} 0.5 & \text{pro } x=0 \\ 1 & \text{pro } 0 < x \leq \frac{N}{2} \\ 0.5 & \text{pro } x=\frac{N}{2} \\ 0 & \text{jinak} \end{cases}$$

kde  $N$  je počet koeficientů Fourierovy transformace - pokud není určeno jinak, je to počet vzorků signálu.

Jinými slovy obálku spočítáte tak, že:

- Spočítáte FFT  $F$  na filtrovaném a převzorkovaném signálu
- Vynulujete pravou symetrickou část spektra
- Levou část spektra vynasobíte 2 kromě prvního a prostředního binu (při sudém počtu frekvenčních binů).
- Provedete zpětnou FFT  $F^{-1}$

Abyste získali obálku signálu, je třeba vzít absolutní hodnotu signálu získaného Hilbertovou transformací.

Obálku a signál vykreslete do jednoho grafu přes sebe, obálka by měla obalovat daný signál.

Detekci QRS poté provedete opět prahováním, tentokrát ale obálky signálu.

Při této úloze **nepoužívejte** funkci `hilbert` z knihovny `scipy`. Cílem je, abyste si ji napsali sami. Můžete ji použít pouze jako baseline pro porovnání vašich výsledků.

# Zde napište váš kód

c) [0.5b] Při kterých metodách detekcí QRS nám vadí otočený (flipnutý) signál, při kterých ne a proč?

*Zde napište vaše řešení.*

## 4.6 [2b] Detekce R-R intervalu

a) Detekujte R-R intervaly pomocí detekovaných QRS z jednotlivých metod, které jste použili dříve. Vykreslete hodnoty R-R intervalu do stejného grafu jako EKG signál a detekované QRS. Vykreslení proveďte nad EKG signál, kde osa  $x$  bude i nadále časová a každý R-R interval bude zakreslen na  $x$  pozici detekovaného QRS. Osa  $y$  pro R-R interval bude určovat hodnotu samotného R-R intervalu.

# Zde napište váš kód

## 4.7 Bonus

a) Načtěte si signál obsahující fibrilaci FIS.wav. Proveďte na něm filtraci a převzorkování. Poté zkuste použít nějaký QRS detektor. Z detekovaných QRS detekujte R-R intervaly. Porovnejte R-R intervaly pro fibrilaci a klasický signál bez patologie (sinusový rytmus). Měli byste vidět prudké změny v R-R intervalech a jejich nepravidelnost. Zároveň se vám může stát, že vám některé metody detekce QRS nepodají tak kvalitní výkon jako při sinusovém rytmu.

*# Zde napište váš kód*

b) Načtěte si signál obsahující komorovou extrasystolu KES.wav. Proveďte na něm filtraci a převzorkování. Spočítejte a zobrazte spektrogram úseku tohoto signálu. Porovnejte spektrogramy vašeho signálu a signálu KES.wav. Měli byste vidět rozšířenou aktivitu na nízkých frekvencích. Dále zobrazte a porovnejte tyto signály v časové doméně. Obsažené komorové extrasystoly by se měly projevit jako zvláštní široké QRS.

*# Zde napište váš kód*