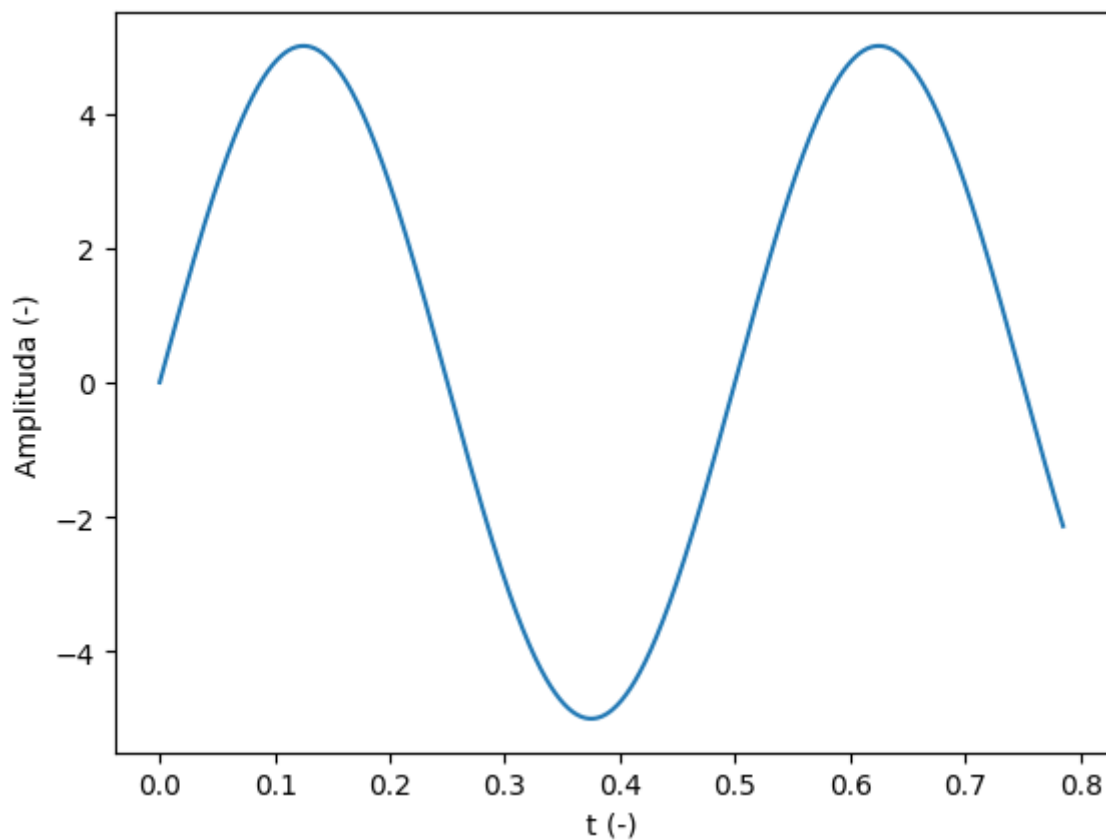


Cvičení 1 - analýza signálu

```
In [ ]: import numpy as np
import sympy as sy
import matplotlib.pyplot as plt
import scipy.io
```

```
In [ ]: # vstupní proměnné
amp = 5
frekv = 2
vzorkFrekv = 1024
delka = np.arange(0, np.pi/4, 1/vzorkFrekv)
funkce = amp*np.sin(2*np.pi*delka*frekv)
```

```
In [ ]: plt.plot(delka, funkce)
plt.xlabel("t (-)")
plt.ylabel("Amplituda (-)")
plt.show()
```



```
In [ ]: # analytický výpočet střední hodnoty a energie
strHodnota = (1/(delka[-1] - delka[0]))*np.trapz(funkce, delka, 1/vzorkFrekv)
print(f"Střední hodnota: {strHodnota:.3f}")

delka2 = np.arange(0, 2*np.pi, 1/vzorkFrekv)
funkce2 = np.sin(2*np.pi*frekv*delka2)
energie = np.trapz(funkce2**2, delka2)
print(f"Energie: {energie:.3f}")
```

Střední hodnota: 0.965

Energie: 3.127

```
In [ ]: # symbolický výpočet
t = sy.symbols("t") # symbolická proměnná
funkceSymb = amp*sy.sin(2*sy.pi*frekv*t)

# symbolický výpočet střední hodnoty
strHodnotaSymb = (1/(sy.pi/4))*sy.integrate(funkceSymb)
print(f"Střední hodnota: {(1/(sy.pi/4))*sy.integrate(funkceSymb, (t, 0, np.pi/4)).evalf():.3f}")
strHodnotaSymb
```

Střední hodnota: 0.964

```
Out[ ]: 
$$-\frac{5 \cos(4\pi t)}{\pi^2}$$

```

```
In [ ]: # symbolický výpočet energie
funkceSymb2 = sy.sin(2*sy.pi*frekv*t)
energieSymb = sy.integrate(funkceSymb2**2)
print(f"Energie: {sy.integrate(funkceSymb2**2, (t, 0, 2*np.pi)).evalf():.3f}")
energieSymb
```

Hodnota energie: 3.127

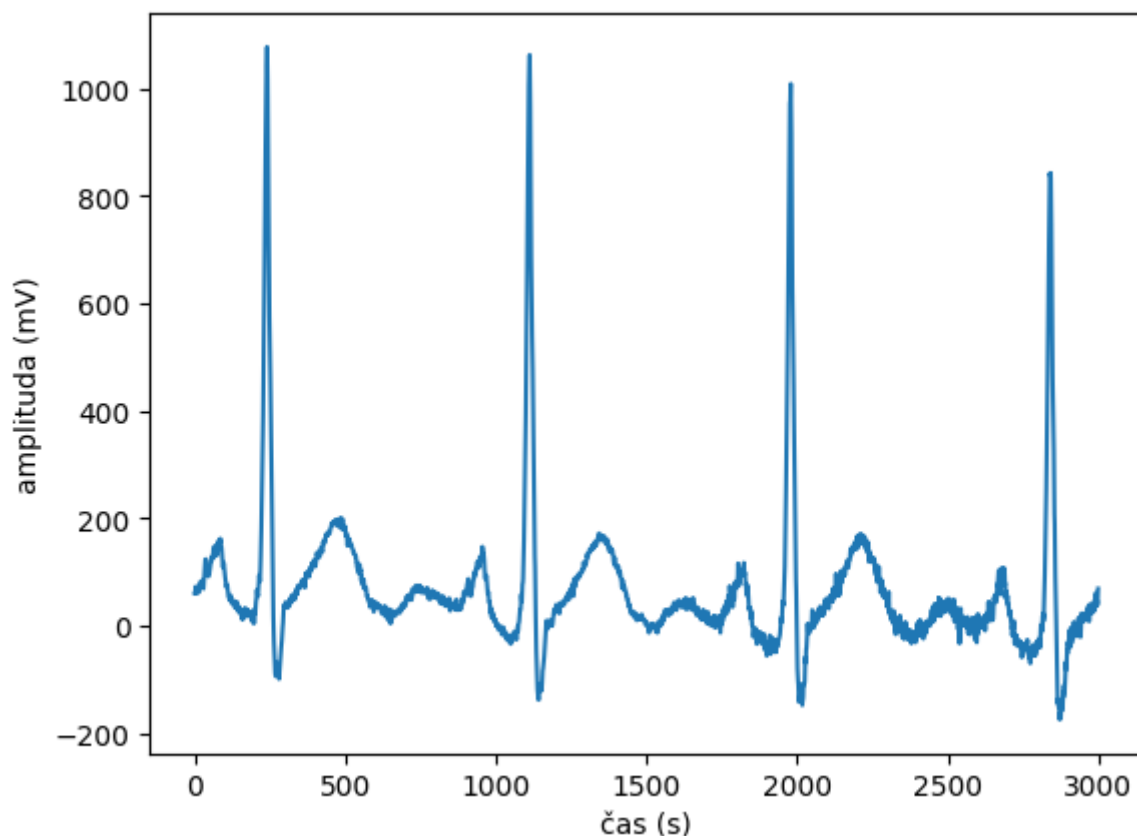
```
Out[ ]: 
$$\frac{2\pi t - \frac{\sin(4\pi t) \cos(4\pi t)}{2}}{4\pi}$$

```

```
In [ ]: # načtení MAT souboru a výběr jednoho signálu
matFile = scipy.io.loadmat("ECG_Test.mat")
ecgSignal = matFile['val'][0, 0:3000]
ecgSignal, ecgSignal.size
```

```
Out[ ]: (array([60, 60, 67, ..., 39, 60, 69], dtype=int16), 3000)
```

```
In [ ]: plt.plot(ecgSignal)
plt.xlabel("čas (s)")
plt.ylabel("amplituda (mV)")
plt.show()
```



```
In [ ]: ampSpec = np.fft.fft(ecgSignal) # amplitudové spektrum
N = len(ecgSignal) # délka signálu
normalize = N/2 # normalizace
samplingRate = 1024.0 # vzorkovací frekvence
freq = np.fft.fftfreq(N, d=1.0/samplingRate) # výpočet frekvenční osy
normAmplitude = np.abs(ampSpec)/normalize # normalizace amplitudy
```

```
In [ ]: plt.plot(freq, normAmplitude)
plt.xlim(0, 100)
plt.ylim(0, 100)
plt.xlabel("frekvence (Hz)")
plt.ylabel("|amplituda| (mV)")
plt.show()
```

