



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

PSK1-3

Název školy:	Vyšší odborná škola a Střední průmyslová škola, Božetěchova 3
Autor:	Ing. Marek Nožka
Anotace:	Význam vysokých a nízkých harmonických složek
Vzdělávací oblast:	Principy přenosu informací
Předmět:	Počítačové sítě a komunikační technika (PSK)
Tematická oblast:	Harmonická analýza
Výsledky vzdělávání:	Žák používá frekvenční oblast k popisu signálu
Klíčová slova:	harmonická analýza, časová oblast, frekvenční oblast, harmonické složky
Druh učebního materiálu:	Návod na domácí samostatné cvičení; Program v jazyce Python
Typ vzdělávání:	Střední vzdělávání, 3. ročník, technické lyceum
Ověřeno:	VOŠ a SPŠE Olomouc; Třída: 3L
Zdroj:	Vlastní poznámky

Harmonická analýza -- cvičení

Vyžijeme programovací jazyk Python, který se učíte používat v předmětu Programování a interaktivní interpret IPython. Předpokládám, že máte nainstalovaný Python(x,y), kde jsou k dispozici všechny potřebné knihovny.

Příprava

Spusťte si program "IPython Qt console" a prostudujte si krátké tutoriály, které jsem pro vás vytvořil:

- [Matplotlib--zakladni myslenky postupy](#)
- [Matplotlib--zaklady pro tvorbu grafu](#)

Nevím jak přesně spouštíte IPython, proto několik málo poznámek. Je možné, že po spuštění programu nemáte importované přístupné matematické a vykreslovací knihovny:

```
[1]>>> plot
```

```
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-1-88d627f14d84> in <module>()  
----> 1 plot  
  
NameError: name 'plot' is not defined
```

Knihovny se zavedou pomocí příkazu:

```
%pylab qt
```

nebo

```
%pylab inline
```

Parametry `qt` a `inline` mění vykreslovací backend a grafy se budou kreslit jinak. Při použití `inline` se budou grafy kreslit přímo do IPython okna. Pokud nastavíte `qt` budou se grafy kreslit do samostatného okna.

Možná v určité chvíli zjistíte, že vám interaktivní práce nevyhovuje a raději byste všechny příkazy zapsali do samostatného souboru. I to je možné ale musíte si dát pozor na jmenné prostory. Vše myslím dobře demonstruje následující příklad:

```
#!/usr/bin/python  
# -*- coding: utf8 -*-  
#####  
import pylab  
from pylab import pi  
  
# sinus 50 Hz  
t = pylab.linspace(0, 30e-3, 512)  
u = 230 * (2 ** 0.5) * pylab.sin(2 * pi * 50 * t)  
  
pylab.plot(t, u)  
pylab.title('Sinus f = 50 Hz')  
pylab.grid(True)  
pylab.xlabel('t [s]')  
pylab.ylabel('u [V]')  
  
pylab.show()
```

`-> [stáhnout](#)

Samostatná práce

Pilovitý časový průběh se dá vyjádřit vztahem:

$$u(t) = \frac{2U_{max}}{\pi} \left(\sin \omega t + \frac{1}{2} \sin \omega 2t + \frac{1}{3} \sin \omega 3t + \dots \right)$$

To znamená, že amplitudy jednotlivých harmonických složek lze zapsat následující způsobilou:

$$U_n = \frac{2U_{max}}{n\pi} \sin n\omega t$$

Vyzkoušejte si, kolik harmonických složek je třeba sečíst, aby tvar výsledného napětí připomínal pilu. Celkový výsledek a alespoň jeden mezivýsledek si vlepte do sešitu. Do sešitu si vlepte také postup, který k výsledku vedl.

Nezapomeňte, že všechny grafy musí mít nadpis a musí mít řádně popsány a ocejchovány osy.