

FYZIKÁLNÍ PRAKTIKUM

Fyzikální praktikum 1

Zpracoval: Milan Suk

Naměřeno: 16. dubna 2018

Obor: F

Skupina: PO 8:00

Testováno:

Úloha č. 9: Měření elektrického napětí a proudu

1. Úvod

V analogové části této plochy je cílem proměřit vnitřní odpor použitého ampérmetru a určit hodnoty pro bočník, resp. předřadník, aby s ním bylo možné měřit hodnoty pro nové zadané rozsahy. V digitální části se má zjistit rozsah užitých D/A převodníků, reálný rozsah napětí, kvantizační krok a rozlišovací schopnost. Nakonec se má otestovat vliv vzorkovací frekvence na kvalitu záznamu analogového signálu.

2. Postup měření

(A) Nejdříve se měl vnitřní odpor ampérmetru určit z ohmova zákona. Ampérmetr se připojil ke zdroji proudu a na svorkách ampérmetru se zároveň měřilo něptí digitálním volmetrem. (B) Při druhé metodě se volmetr nahradil odporovou dekadou. Nejdřív se změřil proud I na ampérmetru bez paralelně připojeného odporu. Poté se hledala taková hodnota odporu, aby ampérmetr měřil polovinu původní hodnoty proudu $\frac{I}{2}$. Tím pádem nastavený odpor na dekádě odpovídá odporu v druhé větvi, tedy vnitřnímu odporu ampérmetru.

Velikost bočníku se hledala pro počáteční rozsah $I_A = 100\mu A$ a vnitřní odpor R_A . Velikost bočníku se pro nový rozsah určí jako $R_B = \frac{R_A}{n-1}$. Odpor předřadníku se spočítá jako $R_P = (\frac{U_N}{U_V} - 1)R_V = (\frac{U_N}{I_A R_A} - 1)R_V$.

Napěťový rozsah převodníků určím jako rozdíl změřených hodnota napětí pro největší a nejmenší hodnota číselného rozsahu. Jelikož je závislost napětí na číselných hodnotách pokud možno lineární, lze jej určit jako napěťový rozsah dělený číselným rozsahem. Kvantizační krok u dvanáctibitového A/D převodníku se má určit měřením napětí při zkratovaných svorkách. V naměřených datech stačí najít nejmenší změnu napětí, tato hodnota by měla odpovídat kvantizačnímu kroku.

3. Výsledky

3.1. Měření vnitřního odporu ampérmetru

Při první metodě pro hodnotu $I = 100\mu A$ naměřil voltmetr hodnotu $U = 180.04mV$. Vnitřní odpor zde vychází

$$R^{(1)} = 1800,4\Omega \quad (1)$$

Pro metodu s bočníkem hodnota odporu vychází

$$R^{(2)} = 1790\Omega \quad (2)$$

3.2. Měření odporu bočníku, resp. předřadníku

I_N	n	R_B
0,5 mA	5	450 Ω
1 mA	10	200 Ω
2 mA	20	95 Ω

Tabulka 1: Hodnoty odporů bočníků

U_N	$\frac{U_N}{U_V} - 1$	R_B
5 V	27	48 600 Ω
10 V	55	99 000 Ω

Tabulka 2: Hodnoty odporů předřadníků

3.3. Digitální část

typ převodníku	číselný rozsah	U_0	U_{max}	napěťový rozsah	kvantizační krok
8-bit D/A	$2^8 = 256$	0.001262 V	9.879249 V	9.877987 V	0.038586 V
16-bit D/A	$2^{16} = 65536$	-10.675357 V	10.698079 V	21.373436 V	0.0003261 V

Tabulka 3: Parametry převodníků

Pro nalezení číselné hodnoty odpovídající hledanému napětí stačí použít jednoduchý vztah.

$$k = \frac{c}{u}U \quad (3)$$

kde jako c jsem si označil číselný rozsah převodníku, u napěťový rozsah a U hledané napětí. Pro hledané napětí v zadání úlohy 3.2V vychází hledaná číselná hodnota pro 8-mibitový převodník mezi 82 a 83 (po změření 3.172783V a 3.209893V), pro 16-tibitový mezi 42547 a 42548 (po změření 3.199839V a 3.200263V).

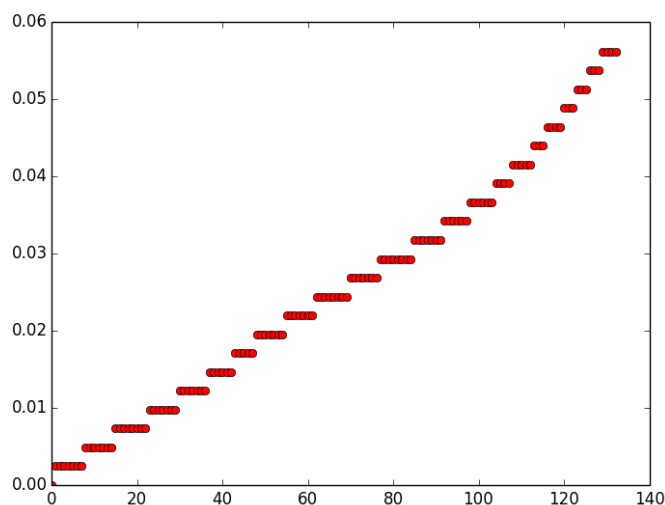
Kvantizační krok 12-tibitového převodníku určím následujícím Python scriptem.

```
1 import numpy
2 import matplotlib.pyplot as plt
3
4 data = numpy.loadtxt("data.txt")
5 data_diffs = []
6
7 for i in range(len(data)):
8     if i == 0:
9         continue
10
11     data_diffs.append(abs(data[i] - data[i - 1]))
12
13 data_sorted = numpy.unique(numpy.sort(data_diffs))
14
15 print(data_sorted[:20])
```

```

16
17 plt.plot(data_sorted, 'ro')
18 plt.show()

```



Obrázek 1: Hodnoty rozdílů napětí

Tedy kvantizační krok je $0.00244199V$.

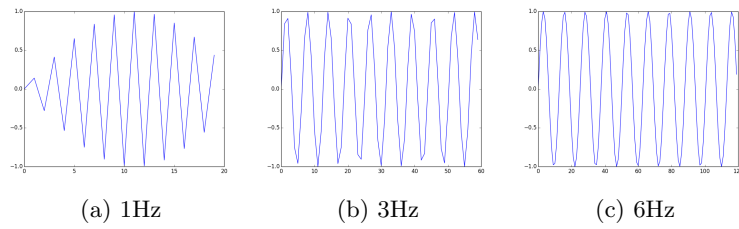
Vliv vzorkovací frekvence na kvalitu záznamu V této úloze se testovalo, jaký vliv má vzorkovací frekvence převodníku na kvalitu získaného záznamu analogového signálu. Největší problémem je, že při vykreslování záznamu předpokládáme, že získané body v grafu jsou si blízké, takže je vykreslovací program generuje tím způsobem, že dva sousedící body spojí vhodnou funkcí - nejčastěji přímkou. Pro vyzkoušení jsem napsal jednoduchý program, který přesně tento proces simuluje.

```

1 import matplotlib.pyplot as plt
2 import numpy
3
4 freq = 3
5 f = 3
6
7 step = 1.0 / f
8 data = [numpy.sin(freq * x) for x in numpy.arange(0, 20, step)]
9 plt.plot(data)
10 plt.show()

```

Proměnná `freq` je frekvence generovaného signálu a `f` je snímkovací frekvence, ze které se generují body do definičního oboru $[0, 20]$.



Obrázek 2: $\sin(3x)$ snímáný různými frekvencemi

Například při snímání frekvencí 1Hz se silně projevuje zmíněný efekt, že při špatně zvolených bodech snímání signálu a následným propojením bodů přímkou působí výsledný graf jako zcela jiná křivka a není vůbec zachycen charakter funkce $\sin(3x)$. U frekvence 3Hz je již alespoň zachycen tvar funkce a při dvojnásobné frekvenci už dostávám poměrně dobrý záznam signálu.