

<b>Mikrosenzory a mikromechanické systémy</b> <b>Ústav mikroelektroniky</b> <b>FEKT VUT v Brně</b>			Jméno <b>Jakub Charvot</b>	ID <b>240844</b>
			Ročník <b>3.</b>	Obor <b>MET</b>
Spolupracoval <b>Radek Kučera</b>	Měřeno dne <b>02.04. 2024</b>	Odevzdáno dne <b>09.04. 2024</b>	<b>Hodnocení</b>	
Název zadání <b>Měření veličin pomocí polohového MEMS senzoru</b>				Č. úlohy <b>10</b>

## 1 Měření a jeho vyhodnocení

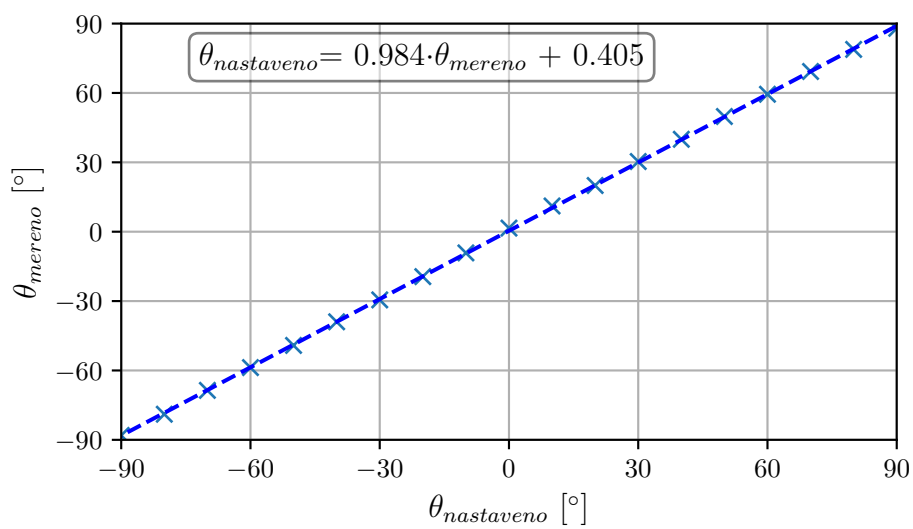
Maximální absolutní chyba použitého senzoru je  $\pm 0,22^\circ$ , což je údaj platný jak pro měření rotace, tak i náklonu – jedná se o stále stejný typ měření, pouze využívá jinou osu senzoru.

Tabulka 1: Nastavené a měřené hodnoty (ADXL345).

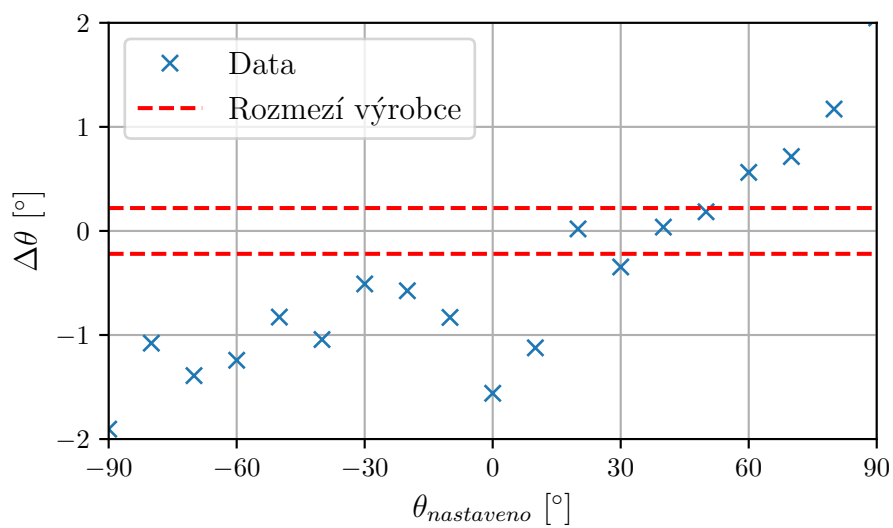
$\theta_{rot-nast}$ [°]	$\theta_{rot-mer}$ [°]	$\Delta\theta_{rot}$ [°]	$\theta_{nak-nast}$ [°]	$\theta_{nak-mer}$ [°]
90,000	87,138	2,047	90,000	87,953
80,000	77,793	1,171	80,000	78,829
70,000	67,770	0,715	70,000	69,285
60,000	57,801	0,563	60,000	59,437
50,000	48,346	0,184	50,000	49,816
40,000	38,345	0,037	40,000	39,963
30,000	28,879	-0,346	30,000	30,346
20,000	18,929	0,019	20,000	19,981
10,000	9,645	-1,123	10,000	11,123
0,000	-0,448	-1,560	0,000	1,560
-10,000	-11,354	-0,831	-10,000	-9,169
-20,000	-21,389	-0,575	-20,000	-19,425
-30,000	-30,961	-0,509	-30,000	-29,491
-40,000	-39,973	-1,043	-40,000	-38,957
-50,000	-50,243	-0,828	-50,000	-49,172
-60,000	-59,785	-1,243	-60,000	-58,757
-70,000	-70,255	-1,390	-70,000	-68,610
-80,000	-79,226	-1,078	-80,000	-78,922
-90,000	-88,276	-1,904	-90,000	-88,096

### 1.0.1 Příklad výpočtu

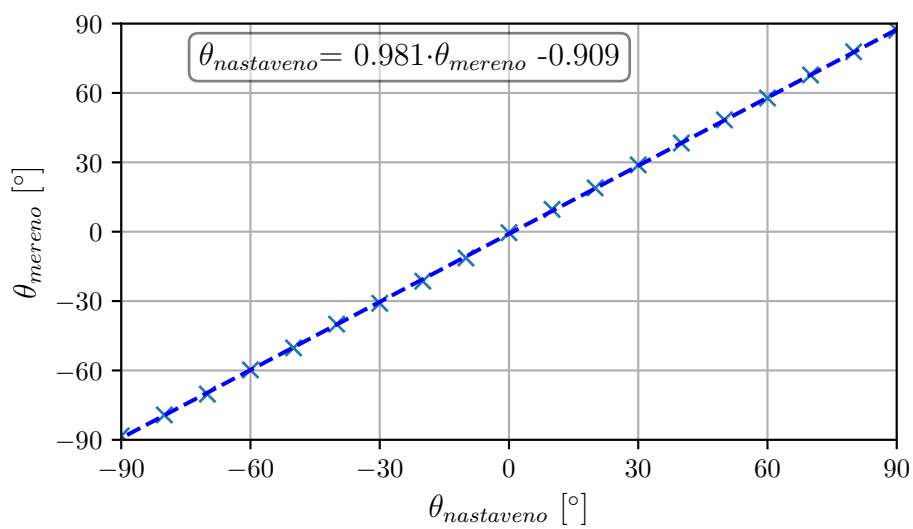
$$\Delta\theta_{rot} = \theta_{rot-nast} - \theta_{rot-mer} \quad \Delta\theta_{nak} = \theta_{nak-nast} - \theta_{nak-mer}$$



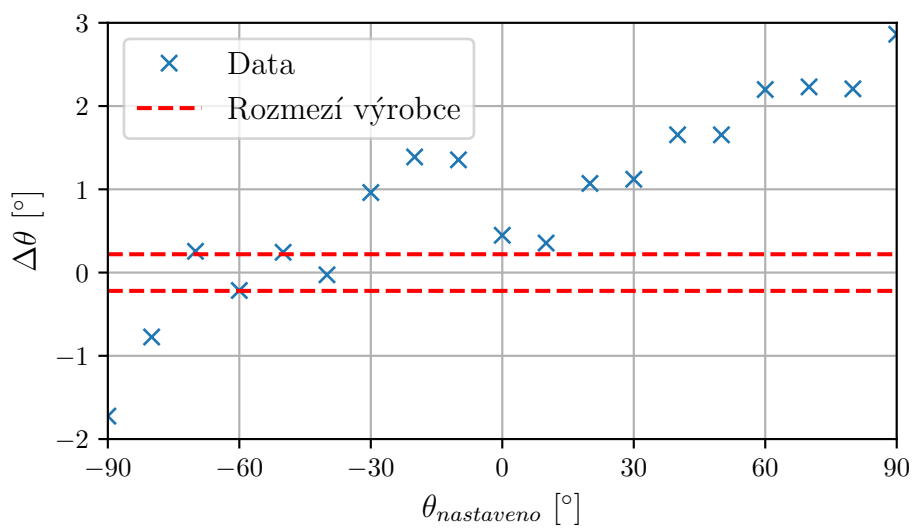
Obr. 1: Kalibrační křivka měření náklonu.



Obr. 2: Korekční křivka měření náklonu.



Obr. 3: Kalibrační křivka měření rotace.



Obr. 4: Korekční křivka měření rotace.

## 1.1 Zdrojový kód

```
1 import smbus
2 import math
3 import kADXL345 as ad
4
5 def main():
6     bus = smbus.SMBus(1)
7     ad.Inicializace345(bus, 0x1E, 0x20)
8
9     print("deg\trot\tnak")
10    deg=0
11    # vložení breakpointu na začátek cyklu umožní jednoduché
12    # krokování při nastavení uhlu
13    while True:
14        x = ad.Precteni(bus, 0x32, 0x33)
15        y = ad.Precteni(bus, 0x34, 0x35)
16        z = ad.Precteni(bus, 0x36, 0x37)
17        if (x == 0) and (z==0) or (y == 0) and (z ==0):
18            pass
19        else:
20            rot = 0
21            nak = 0
22            for deg in range(0, 10):
23                rot += math.atan(y/math.sqrt(math.pow(x, 2) +
24                math.pow(z, 2))) * 180/math.pi
25                nak += math.atan(-1 * x/math.sqrt(math.pow(y, 2)
26                + math.pow(z, 2))) * 180/math.pi
27                rot = format(rot, '.4f')
28                nak = format(nak, '.4f')
29                print(str(deg) + '\t' + str(rot) + '\t' + str(nak) +
30                '\n')
31            deg += 10
32
33 if __name__ == "__main__":
34     main()
```

Listing 1: Použitý kód v jazyce Python

Tabulka 2: Měření a vypočtené hodnoty (ADXL203).

Vibrace v ose X			Vibrace v ose Y		
$f_{\text{Hz}}$ [Hz]	Amplituda [–]	$f_{\text{RPM}}$ [RPM]	$f_{\text{Hz}}$ [Hz]	Amplituda [–]	$f_{\text{RPM}}$ [RPM]
U = 4,5 V					
777	0,034		483	0,013	
678	0,009		387	0,017	
488	0,009		309	0,021	
387	0,0106	23220	195	0,035	11700
U = 6 V					
762	0,06		286	0,015	
503	0,017		251	0,012	
375	0,013		218	0,019	13080
285	0,012	17100	926	0,031	
U = 8 V					
802	0,052		450	0,022	
767	0,022		314	0,016	
602	0,024		285	0,028	
280	0,037	16800	161	0,031	9660
U = 10 V					
777	0,101		549	0,0222	
554	0,025		342	0,016	
336	0,044		192	0,028	
240	0,037	14400	139	0,031	8340
U = 12 V					
625	0,0394		324	0,069	
217	0,062		275	0,058	
754	0,065		166	0,055	
53	0,133	3180	86	0,045	5160

## Závěr

V této úloze jsme za pomoci předpřipraveného Python kódu pracovali se dvěma typy akcelerometrů. V první části jsme využili schopnost senzoru měřit statické zrychlení ve třech osách. Při náklonu senzoru se mění rozložení tíhové síly mezi osami a lze tedy vypočítat úhel náklonu. Z přiložených kalibračních křivek vyplývá, že senzor měří poměrně přesně a lineárně v celém svém rozsahu, absolutní chyba nikdy nepřesahuje  $2^\circ$ , ovšem výrobce garantuje hodnotu ještě o řád menší a tento požadavek rozhodně splněn není. Jelikož program automaticky přepočítává měřenou analogovou hodnotu na údaj náklonu ve stupních a k původní hodnotě nemáme přístup, nemá zde smysl mluvit o citlivosti senzoru, i když to po nás zadání vyžaduje.

Ve druhé části úlohy jsme měřili dynamické zrychlení, v našem případě vibrace. Kvůli naprosto nevhodnému způsobu zpracování dat, který autor úlohy zvolil (i přesto, že měl k dispozici plné možnosti jazyka Python a mohl využít algoritmicky mnohem přesnější metodu) jsou námi získaná data zatížena velkou chybou. Hodnotu základní frekvence ze které se má dále počítat rychlost otáček se nám podařilo zachytit pouze v několika případech a tedy ani vypočtené hodnoty neodpovídají předpokládanému trendu.