



**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

---

**FAKULTA BIOMEDICÍNSKÉHO INŽENÝRSTVÍ**

**Katedra biomedicínské informatiky**

**Návrh aplikace pro kalibraci a korekci polohy soustavy  
nositelných inerciálních MoCap senzorů**

**Design of application for the calibration and position correction of  
a wearable inertial sensors MoCap**

Diplomová práce

Studijní program: Biomedicínská a klinická informatika  
Specializace: Softwarové technologie  
Vedoucí práce: Ing. Michaela Hourová

**Bc. Petr Šrotýř**

---

**Kladno 2022**



# ZADÁNÍ DIPLOMOVÉ PRÁCE

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Šrotýř** Jméno: **Petr** Osobní číslo: **426078**  
Fakulta: **Fakulta biomedicínského inženýrství**  
Garantující katedra: **Katedra biomedicínské informatiky**  
Studijní program: **Biomedicínská a klinická informatika**

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Návrh aplikace pro kalibraci a korekci polohy soustavy nositelných inerciálních MoCap senzorů**

Název diplomové práce anglicky:

**Design of application for the calibration and position correction of a wearable inertial sensors MoCap system**

Pokyny pro vypracování:

Cílem práce je návrh softwarového nástroje pro konfiguraci a kalibraci soustavy inerciálních MoCap senzorů umístovaných na objekty, v závislosti na požadavcích navržených modelů. Navrhněte a implementujte metody umožňující určení relativních úhlů senzorů. Dále navrhněte a implementujte metody, které na základě naměřených hodnot a modelu rozmístění senzorů budou poskytovat uživateli zpětnou vazbu, zdali aktuální poloha senzorů odpovídá požadavku a s jakou chybou jsou senzory umístěny. Vytvořte rozhraní, které bude dávat uživateli real-time zpětnou vazbu aktuální polohy senzorů spolu s vhodnými parametry pro zpětnou vazbu uživateli o jeho manipulaci se senzorem. Navržené metody implementujte v jazyce Python či Matlab. Provedte testování vytvořeného nástroje a výsledky statisticky vyhodnoťte. Vytvořte návod pro použití daného nástroje.

Seznam doporučené literatury:

- [1] Butun, Ismail, Decision Support Systems and Industrial IoT in Smart Grid, Factories, and Cities, ed. 1, IGI Global, 2327-0411
- [2] LEA, Perry, IoT and Edge Computing for Architects: Implementing edge and IoT systems from sensors to clouds with communication systems, analytics, and security, ed. 2, Packt Publishing, 2020, ISBN 978-1839214806
- [3] HILLAR, G., C., Internet of Things with Python, ed. 1, Packt Publishing, 2016, ISBN 978-1785881381

Jméno a příjmení vedoucí(ho) diplomové práce:

**Ing. Michaela Hourová**

Jméno a příjmení konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **14.02.2022**

Platnost zadání diplomové práce: **18.09.2023**

doc. Ing. Zoltán Szabó Ph.D.  
vedoucí katedry

prof. MUDr. Jozef Rosina, Ph.D., MBA  
děkan

## **PROHLÁŠENÍ**

Prohlašuji, že jsem diplomovou práci s názvem Návrh aplikace pro kalibraci a korekci polohy soustavy nositelných inerciálních MoCap senzorů vypracoval samostatně a použil k tomu úplný výčet citací použitých pramenů, které uvádím v seznamu přiloženém k diplomové práci.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů.

V Kladně dne:

.....

Bc. Petr Šrotýř

## **PODĚKOVÁNÍ**

Rád bych poděkoval paní Ing. Michaele Hourové za mimořádně cenné připomínky, odborné rady, trpělivost a ochotu při vypracování diplomové práce, a také svojí rodině za veškerou podporu při studiu.

# ABSTRAKT

## Návrh aplikace pro kalibraci a korekci polohy soustavy nositelných inerciálních MoCap senzorů:

Tato práce se zabývá návrhem aplikace pro kalibraci a korekci polohy soustavy nositelných MoCap senzorů. Podle provedeného průzkumu byly zjištěny nejčastější chyby MoCap senzorů, mezi které patří bias chyba, chyba vychýlených měřících os a nelinearita. Pro minimalizaci těchto chyb byly hledány metody kterými je lze minimalizovat nebo odstranit. Bias chybu lze minimalizovat naměřením většího množství dat v klidové poloze senzoru a vypočtením průměru hodnot. Chybu vychýlení měřících os lze minimalizovat vytvořením kalibrační matice kterou jsou maticově vynásobeny naměřená data. Chyba nelinearity je zahrnuta v kalibrační matici a minimalizována při maticovém součinu.

Cílem této práce je návrh softwarového nástroje schopného zkalibrovat soustavu MoCap senzorů a poskytovat uživateli zpětnou vazbu o aktuální poloze senzoru pro co nejsnažší použití kalibrační metody. Samotný software umožňuje vytvoření kalibrační matice a její aplikaci na naměřená data. Dále software umožňuje uložení vytvořené kalibrační matice ve formátu \*.csv a načtení kalibrační matice zpět do programu. Dalším důležitým prvkem programu je možnost sledování polohy senzoru v prostoru prostřednictvím vytvořených grafů a zobrazených vypočtených parametrů.

Pro určení relativní polohy senzoru je využito gravitačního zrychlení působící na senzor v klidovém stavu na vytvořeném kalibračním stole. Relativní poloha senzoru je vypočítána podle naměřených hodnot senzorem a působícím gravitačním zrychlením na senzor. Navržená kalibrační metoda minimalizuje bias chybu, vychýlení měřících os a nelinearitu.

Bias chyba je vypočítána z měření v klidové poloze senzoru a byla snížena pro osu  $x$  o 0,02359 g, pro osu  $y$  o 0,00544 g a pro osu  $z$  o 0,04119 g. Chyba vychýlených měřících os je minimalizována pomocí kalibrační matice vytvořené pomocí Gram-Schmidtova ortogonalizačního procesu. Chyba vychýlených měřících os je znázorněna jako narovnání os do kolmého tvaru na měřenou osu. V případě os  $x$  a  $y$  došlo k narovnání os o  $0,50^\circ$ , pro osy  $y$  a  $z$  došlo k narovnání os o  $1,28^\circ$  a pro osy  $z$  a  $x$  došlo k narovnání os o  $1,95^\circ$ .

## Klíčová slova

Inerciální senzor, kalibrace, korekce polohy, python

# ABSTRACT

## **Design of application for the calibration and position correction of a wearable inertial sensors MoCap:**

This thesis deals with the design of a program for calibration and position correction of a system of wearable MoCap sensors. According to the survey is the most common errors of MoCap sensors were identified, which include bias error, error of deflected measuring axes and nonlinearity. To minimize these errors was sought methods that can minimize or eliminate them. The bias error can be minimized by measuring a large amount of data in rest position of the sensor and calculate the average of the all values. The error of measuring axes can be minimized by creating calibration matrix by which the measured data are multiplied by the matrix. The nonlinear error is included in the calibration matrix and minimized at the matrix application.

Objective of this thesis is design of a software tool capable of calibration the MoCap sensor system and provide the user feedback about current position of the sensor for the easiest possible using of calibration program. The software itself allows create of calibration matrix and its application on the measured data. Furthermore, software allows save the created calibration matrix in \*.csv format and load the calibration matrix back into the program. Another important element of the software is the ability to monitor position of the sensor in space through the created graphs and displayed calculated parameters.

To determine the relative position of the sensor is use a gravitational acceleration causing on the sensor in the rest, on the created calibration table. The relative position of the sensor is calculate according to the measured values by sensor and the gravitational acceleration acting on the sensor.

The bias error is calculated from the measurement data with rest position of the sensor and was reduced for the  $x$ -axis by 0.02359  $g$ , for the  $y$ -axis by 0.00544  $g$  and for the  $z$ -axis by 0.04119  $g$ . The error of the deflected measuring axes is minimized by calibration matrix created by the Gram-Schmidt orthogonalization process. The error of the deflected measuring axes is shown as the alignment of the axes perpendicular to the measured axis. In the case of the  $x$ -axis and  $y$ -axis were aligned by 0.50 °, in the  $y$ -axis and  $z$ -axis were aligned by 1.28 ° and in the  $z$ -axis and  $x$ -axis were aligned by 1.95 °.

## **Keywords**

Inertial sensor, calibration, position correction, python

# Obsah

<b>Seznam symbolů a zkratk</b> .....	<b>9</b>
<b>Seznam obrázků</b> .....	<b>10</b>
<b>1 Úvod</b> .....	<b>13</b>
<b>2 Přehled současného stavu</b> .....	<b>15</b>
2.1 Motion Capture (MoCap).....	15
2.2 Inerciální MoCap systémy .....	17
2.2.1 Inerciální systémy.....	17
2.2.2 Akcelerometr .....	18
2.2.3 Gyroskop .....	19
2.2.4 Magnetometr.....	19
2.2.5 Využití inerciálního MoCapu .....	20
2.3 Chyby inerciálních MoCap systémů .....	20
2.3.1 Bias .....	21
2.3.2 Chyba měřítka (scale factor error).....	22
2.3.3 Chyba vychýlených měřících os (chyba ortogonality).....	23
2.3.4 Nelinearita .....	23
2.3.5 Náhodný šum (random noise) .....	23
2.3.6 Shrnutí chyb inerciálních MoCap systémů.....	23
2.4 Kalibrace inerciálních MoCap senzorů .....	25
2.5 Metody kalibrace akcelerometru .....	26
2.5.1 Kalibrace metodou nejmenších čtverců.....	27
2.5.2 Kalibrace metodou Levenberg-Marquardova algoritmu .....	27
2.5.3 Kalibrace metodou nelineárního estimátoru parametrů .....	28
2.5.4 Kalibrace metodou nelineárním optimalizačním problémem .....	29
2.5.5 Kalibrace pomocí kalibrační matice vytvořené Gram-Schmidtovým ortogonalizačním procesem .....	29
2.6 Metody kalibrace gyroskopu.....	30
2.6.1 Metoda využívající referenční zemskou rotaci.....	30
2.6.2 Metoda využívající referenční rotaci kalibrovaného akcelerometru ..	30
2.6.3 Metoda využívající kalibrovaného magnetometru .....	30

2.7	Metody pro výpočet relativních úhlů senzorů.....	31
2.7.1	Metoda určení relativního úhlu akcelerometru.....	31
2.7.2	Metoda určení relativního úhlu gyroskopu.....	32
<b>3</b>	<b>Cíle práce.....</b>	<b>33</b>
<b>4</b>	<b>Metody .....</b>	<b>34</b>
4.1	Návrh pouzdra pro pevné umístění kalibrovaného senzoru .....	34
4.2	Návrh určení relativního úhlu akcelerometru.....	35
4.3	Návrh kalibrační metody .....	36
4.3.1	Návrh kalibrační matice pracující s naměřenými hodnotami.....	38
4.4	Návrh implementace navržených metod .....	40
4.5	Návrh implementace uživatelského rozhraní .....	43
<b>5</b>	<b>Výsledky.....</b>	<b>47</b>
5.1	Realizace metod .....	47
5.1.1	Realizované umístění senzoru .....	48
5.1.2	Realizovaný stojan pro kalibraci .....	49
5.1.3	Realizace uživatelského rozhraní .....	49
5.1.4	Realizace implementace komunikace se senzorem .....	51
5.1.5	Realizace korekce polohy a určení relativních úhlů senzoru .....	55
5.1.6	Realizace kalibrační metody.....	56
5.2	Testování a zhodnocení navrženého nástroje.....	63
<b>6</b>	<b>Diskuse.....</b>	<b>68</b>
<b>7</b>	<b>Závěr .....</b>	<b>70</b>
	<b>Seznam použité literatury .....</b>	<b>71</b>
	<b>Příloha A: Uživatelský návod pro použití navrženého nástroje.....</b>	<b>75</b>



# Seznam symbolů a zkratek

## Seznam zkratek

Zkratka	Plný název	Význam
MEMS	Micro Electro Mechanical Systems	Mikroelektromechanické zařízení s integrovanými ovladači
MoCap	Motion Capture	Proces záznamu pohybu skutečného objektu pomocí kamer a markerů nebo pomocí rozpoznávání objektů
LM	Levenberg-Marqardt	Levenberg-Marqardtův algoritmus k řešení nelineárních problémů pomocí metody nejmenších čtverců
NIS	Normalized Innovation Squared value	Filtr používaný pro odstranění šumu z naměřených hodnot
RAR	Relative Angle of Rotation	Relativní úhel natočení senzoru

# Seznam obrázků

Obr. 2.2: Příklad využití MoCap systému ve filmovém průmyslu u filmu Pán prstenů a dvě věže (vlevo) (převzato z [11] ) a Zrození planety opic (vpravo) (převzato z [12] ) .....	16
Obr. 2.3: Příklad využití MoCap systému v herním průmyslu u hry Detroit become human (vlevo) (převzato z [15]) a Uncharted (vpravo) (převzato z [16]).....	16
Obr. 2.4: příklad využití MoCap systému ve zdravotnictví ke zlepšení rehabilitace (převzato z [18]) .....	17
Obr. 2.5: Příklad inerciálního systému .....	18
Obr. 2.6: Příklad MEMS akcelerometru (převzato z [24] ) .....	18
Obr. 2.7: Příklad klasického (vlevo) (převzato z [25] ) a MEMS (vpravo) (převzato z [26] ) gyroskopu .....	19
Obr. 2.8: Příklad MEMS magnetometru (převzato z [29] ) .....	19
Obr. 2.9: Příklad inerciálního celotělového MoCap systému na celé tělo (vlevo) (převzato z [30]) a pouze na paži (vpravo) (převzato z [31] ) .....	20
Obr. 2.10: Charakteristické chyby senzorů (převzato a upraveno z [35] ) .....	21
Obr. 2.11: Grafické znázornění bias chyby (převzato a upraveno z [38] ) .....	22
Obr. 2.12: Příklad 12 definovaných poloh pro kalibraci (převzato a upraveno z [43] ) .....	26
Obr. 2.13: Příklad výsledků kalibrace akcelerometru dronu (převzato a upraveno z [46]) .....	28
Obr. 2.14: Příklad kolmému natočení osy $a_z$ akcelerometru (vlevo) a chybného natočení osy $a_z$ akcelerometru (vpravo).....	31
Obr. 4.1: Návrhový výkres spodního dílu pouzdra, ve kterém je umístěn senzor (vytvořeno pomocí programu SolidWorks student edition .....	34
Obr. 4.2: Navržené pouzdro pro umístění senzoru. Spodní díl, ve kterém jsou umístěny senzory (vlevo) a víčko pouzdra (vpravo).....	35
Obr. 4.3: Model navrženého kalibračního stolu .....	36
Obr. 4.4: Příklad naměření hodnot akcelerometrem v prvních 7 vteřinách osou $x$ kolmo k zemi a následně od 8 vteřiny osou $y$ kolmo k zemi .....	37
Obr. 4.5: Definované polohy pro kalibraci na kalibračním stole.....	38
Obr. 4.6: Stavový diagram pro vytvoření kalibrační matice.....	42
Obr. 4.7: Obrázek návrhu uživatelského rozhraní .....	43
Obr. 5.1: Realizované rozhraní pro měření a kalibraci senzorů .....	47
Obr. 5.2: Použitý senzor UMP-6050 .....	48

Obr. 5.3: Realizované umístění senzoru v pouzdru .....	48
Obr. 5.4: Realizovaný kalibrační stolek .....	49
Obr. 5.5: Realizované uživatelské rozhraní .....	50
Obr. 5.6: Příklad vložení objektů do mřížky .....	50
Obr. 5.7: specifický řádek ukončující vytváření uživatelského rozhraní.....	50
Obr. 5.8: senzor UMP-6050 v pouzdru (vlevo) a usb Bluetooth adaptér (vpravo).....	51
Obr. 5.9: Kód pro vypsání dostupných zařízení .....	51
Obr. 5.10: Vnitřní funkce pro nalezení dostupných zařízení .....	52
Obr. 5.11: Uživatelské rozhraní po nalezení dostupných Bluetooth zařízení .....	52
Obr. 5.12: Funkce pro připojení zařízení .....	53
Obr. 5.13: Kód pro ukládání naměřených dat pomocí funkce <code>test_obj_input()</code> .....	53
Obr. 5.14: Kód pro start měření.....	54
Obr. 5.15: Funkce pro převod unixového času na vteřiny od počátku měření .....	54
Obr. 5.16: Rozhraní po úspěšném připojení zařízení a vykreslení prvního bloku dat .....	55
Obr. 5.17: Funkce pro určení korekce polohy .....	55
Obr. 5.18: Příklad kódu pro výpočet relativních úhlu mezi osou $x$ , $y$ a $x$ , $z$ .....	56
Obr. 5.19: Korektní vodorovná poloha podle vodováh (vlevo), korektní vodorovná poloha podle digitálního úhlooměru (uprostřed) a korektní svislá poloha podle digitálního úhlooměru .....	56
Obr. 5.20: Korektní umístění senzoru pro osu $x$ v kolmém směru na zem.....	57
Obr. 5.21: Kontrola správného umístění senzoru a vytvoření kalibrační matice.....	57
Obr. 5.22: Příklad kódu pro výpočet kalibrační matice pomocí Gram-Schmidtovy metody.....	58
Obr. 5.23: Korektní umístění senzoru pro osu $y$ v kolmém směru na zem.....	58
Obr. 5.24: Kontrola správného umístění senzoru a vytvoření kalibrační matice.....	59
Obr. 5.25: Korektní umístění senzoru pro osu $z$ v kolmém směru k zemi.....	60
Obr. 5.26: Kontrola správného umístění senzoru a vytvoření kalibrační matice.....	60
Obr. 5.27: Okno po vytvoření kalibrační matice s výrazněnou možností uložení matice .....	61
Obr. 5.28: Příklad výsledné kalibrační matice.....	61
Obr. 5.29: Rozhraní po aplikaci vytvořené kalibrační matice .....	62
Obr. 5.30: Výstřížek rozhraní po načtení kalibrační matice .....	62
Obr. 5.31: Naměřené, kalibrované a vypočtené parametry pro osu $x$ .....	63

Obr. 5.32: Naměřené, kalibrované a vypočtené parametry pro osu $y$ .....	64
Obr. 5.33: Naměřené, kalibrované a vypočtené parametry pro osu $z$ .....	64
Obr. 8.1: Korektní vodorovná poloha podle vodováh (vlevo), korektní vodorovná poloha podle digitálního úhlooměru (uprostřed) a korektní svislá poloha podle digitálního úhlooměru .....	75
Obr. 8.2: Rozhraní před skenováním Bluetooth zařízení s výrazněným tlačítkem pro hledání dostupných zařízení.....	76
Obr. 8.3: Rozhraní s výčtem dostupných Bluetooth zařízení s vyznačenou konkrétní MAC adresou měřeného zařízení.....	76
Obr. 8.4: Rozhraní s vloženou MAC adresou do kolonky.....	77
Obr. 8.5: Rozhraní po úspěšném spárování z Bluetooth zařízení a vykreslením prvního bloku naměřených hodnot.....	77
Obr. 8.6: Korektní umístění senzoru pro osu $x$ v kolmém směru k zemi .....	78
Obr. 8.7: Kontrola správného umístění senzoru .....	78
Obr. 8.8: Naměření alespoň pěti vteřin měření, po kterém následuje stisknutí tlačítka vytvoření kalibrační matice.....	79
Obr. 8.9: Korektní umístění senzoru pro osu $y$ v kolmém směru k zemi .....	80
Obr. 8.10: Kontrola správného umístění senzoru .....	80
Obr. 8.11: Naměření alespoň pěti vteřin měření, po kterém následuje stisknutí tlačítka vytvoření kalibrační matice.....	81
Obr. 8.12: Korektní umístění senzoru pro osu $z$ v kolmém směru k zemi.....	81
Obr. 8.13: Kontrola správného umístění senzoru .....	82
Obr. 8.14: Naměření alespoň pěti vteřin měření, po kterém následuje stisknutí tlačítka vytvoření kalibrační matice.....	82
Obr. 8.15: Rozhraní po aplikaci vytvořené kalibrační matice .....	83

# 1 Úvod

Cílem práce je návrh softwarového nástroje pro konfiguraci a kalibraci soustavy inerciálních Motion Capture (MoCap) senzorů, které jsou umístěny na objekty dle požadavků navržených modelů. Součástí této práce je dále návrh metod a algoritmů umožňující určení relativních úhlů senzorů a z naměřených hodnot poskytnutí přímé zpětné vazby uživateli o aktuální poloze zařízení a chybovosti senzorů.

MoCap senzory mají v současné době velmi široké využití, ať už ve filmu [12][13], herním průmyslu [14][15] nebo medicíně [18][19]. Největšími úskalími MoCap systémů jsou vysoké pořizovací náklady, nutnost kalibrace a v některých případech i potřeba školeného personálu. [32] [34]

Mezi inerciální senzory jsou řazeny akcelerometry, gyroskopy a magnetometry. Nejběžnějším druhem inerciálních senzorů jsou MEMS senzory, používané v měření rehabilitačními pomůckami [18], exoskeletech [30] apod.

Největším úskalím MEMS senzorů jsou tzv. charakteristické chyby. Mezi charakteristické chyby MEMS senzorů patří bias chyba, chyba vychýlení měřících os, nelinearita, náhodný šum a chyba měřítka. Bias chybu lze minimalizovat naměřením vícera hodnot ve statické poloze senzoru a následným zjištěním, zda byla na senzoru zjištěna jiná hodnota než nulová. U minimalizace chyby vychýlených měřících os je nutné vypočítat kalibrační matici podle předem definovaného postupu a aplikovat na naměřená data.

Pro kalibraci inerciálních senzorů je velmi často využíváno speciálních vybaveních (např. tříosý stůl) nebo sofistikovaných algoritmů [45] [47] [52].

V případě kalibrace akcelerometru je nejčastěji vycházeno ze skutečnosti, že na akcelerometr umístěný v klidové poloze působí pouze gravitační zrychlení. Tuto skutečnost lze snadno ověřit položením akcelerometru kolmo vůči zemi jednou ze tří os.

Tato práce se zabývá návrhem kalibračního programu pro senzor z řady inerciálních senzorů, akcelerometr, vhodný pro sledování relativních úhlů a korekci polohy. Pro účely této práce byla navržena kalibrační metoda využívající působení gravitační síly na akcelerometr v klidové poloze. Pro správnou aplikaci této metody bylo navrženo a realizováno specifické pouzdro pro akcelerometr a unikátní kalibrační stůl, který je součástí řešení kalibrace senzoru.

Pro samotné ovládání kalibrační procedury bylo navrženo a implementováno uživatelsky přívětivé rozhraní, umožňující provedení kalibrace akcelerometru.

Výsledkem práce je systém kalibrace polohy senzoru, reprezentace relativní polohy a rozdílu polohy od požadavku skládajícího se z kalibračního stolu, pouzdra pro akcelerometr a grafického uživatelského rozhraní.

## 2 Přehled současného stavu

### 2.1 Motion Capture (MoCap)

Motion Capture (MoCap) neboli snímání pohybu, je technologie zachycení pohybu těles pomocí různých typů senzoru, jako optické senzory, inerciální senzory a mnoho dalších. Díky neustálému vývoji digitálního světa se technologie MoCap čím dál více uplatňují pro klinické účely [1] , sportovní odvětví, hry, animace, interakce člověka s počítačem [2] a rehabilitace [3] [4] [5] .

Moderní MoCap systémy jsou nejčastěji založené na optických technologiích, které mohou být buďto markerové nebo bezmarkerové. Markerové systémy jsou založeny na sledování optických značek (nejčastěji z retroreflexního materiálu) upevněných buďto na těle nebo obleku. Mezi optické markerové MoCap systémy patří např. Vicon [6] , Xsens [7] nebo Rokoko [8] . Bezmarkerové systémy využívají techniky počítačového vidění (např. potlačení pozadí, extrakce siluety nebo rekonstrukce kostry z RGB snímku/hloubkové mapy). Mezi optické bezmarkerové MoCap systémy patří např. Microsoft Kinect [9] . Největší nevýhodou optických MoCap systémů jsou velmi často chybějící data, které se mohou vyskytnout např. zakrytím markeru nebo schovanou částí těla za překážkou. [10] .

Optické MoCap systémy využívají zaznamenaná data ze dvou nebo více kamer. Kamery v systému jsou vůči sobě kalibrovány tak, aby poskytovaly překrývající záběry z požadovaných úhlů (např. dvě kamery kolmé na sebe)

Během záznamu pohybu MoCap systémem není zaznamenáván celý objekt, ale na objektu jsou nejčastěji umístěny markery, které jsou pro optický systém snadno detekovatelné.

Hlavními přednostmi MoCap systému jsou možnost téměř reálné získávání dat, lehké zpracování složitých pohybů a možnost získání velkého množství pohybových dat za relativně krátký čas.

Hlavními nevýhodami MoCap systému jsou potřeba specifického hardwaru a softwaru, náklady na pořízení potřebného vybavení a prostorová náročnost systému.

MoCap systémy jsou využívány především v následujících odvětvích:

- **Filmový průmysl**

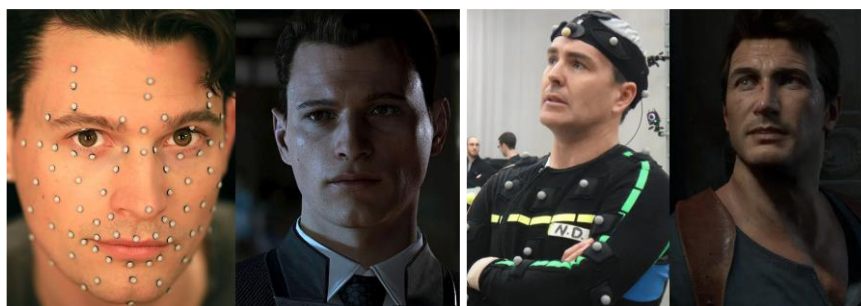
Ve filmovém průmyslu byl pán prstenů a dvě věže jedním z prvních filmů kde byla technologie MoCap použita pro animaci Gluma. Při nahrávání záznamu měl herec na sobě oblečen speciální oblek se senzory snímající jeho pohyb a obličej (viz obrázek 2.9) Dalším filmem kde byl využit MoCap systém pro zaznamenání pohybu postavy byl zrození planety opic. (viz obrázek 2.2)



Obr. 2.2: Příklad využití MoCap systému ve filmovém průmyslu u filmu Pán prstenů a dvě věže (vlevo) (převzato z [11] ) a Zrození planety opic (vpravo) (převzato z [12] )

- **Herní průmysl**

V herním průmyslu je MoCap využíván především pro záznam pohybu postav a obličejové mimiky. Využití je podobné jako u filmu. Zaznamenané pohyby jsou následně převedeny do grafického prostředí. Díky tomuto přístupu jsou pohyby a mimika postav ve hrách přirozenější. Příkladem využití MoCap v počítačových a konzolových hrách jsou Detroit become human, until dawn, série uncharted (viz obrázek 2.3) nebo death stranding. [13] [14]

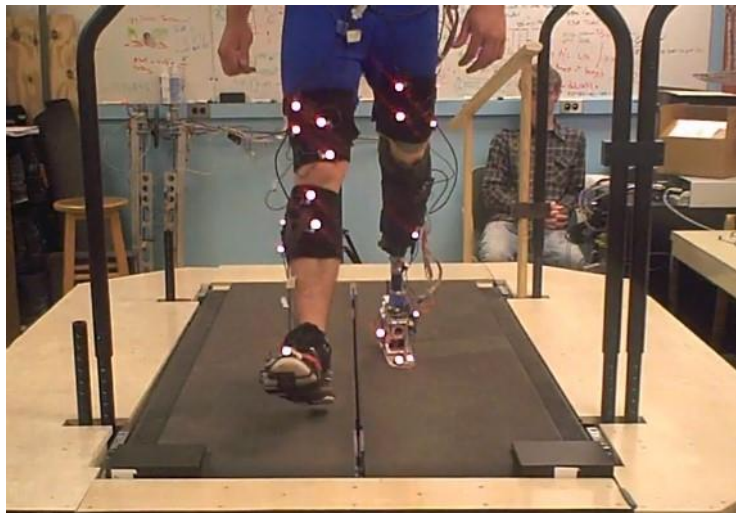


Obr. 2.3: Příklad využití MoCap systému v herním průmyslu u hry Detroit become human (vlevo) (převzato z [15] ) a Uncharted (vpravo) (převzato z [16] )

- **Zdravotnictví**

Ve zdravotnictví je MoCap technologie využívána především k diagnostice pacientů. MoCap ve zdravotnictví umožňuje lékařským profesionálům analyzovat záznam a na jeho základě posoudit možnosti léčby pro pacienty. Hlavní výhodou tohoto záznamu je neinvazivnost zákroku. Využívá se například v ortopedii kde lze pomocí MoCapu analyzovat a vizualizovat různé muskuloskeletální dysfunkce a data následně porovnávat se standartními biomedicínskými daty. Dalším častým využitím MoCapů je v rehabilitaci kde ze záznamu cvičení lze následně analyzovat průběh cvičení a upravit toto cvičení na míru (viz obrázek 2.4). [17]





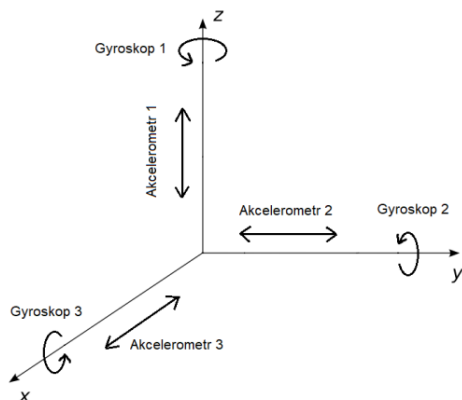
Obr. 2.4: příklad využití MoCap systému ve zdravotnictví ke zlepšení rehabilitace (převzato z [18] )

## 2.2 Inerciální MoCap systémy

Sledování pohybu pomocí inerciálních senzorů se používá již delší dobu, a to především v chytrých telefonech. Soustava inerciálních systémů je složena z inerciálních senzorů jako jsou akcelerometry, gyroskopy, magnetometry a jejich různé variace. Akcelerometr měří velikost vektoru zrychlení v ose X, Y nebo Z, gyroskop měří rotaci v oblasti 360°, v případě rychlostního gyroskopu měří úhlové zrychlení a magnetometr určuje směr vektoru magnetického pole, což má podobnou funkci jako kompas. [19]

### 2.2.1 Inerciální systémy

Inerciální systémy (příklad na obrázku 2.5) jsou založeny na inerciálních senzorech a biomechanických modelech. Údaje naměřené inerciálními senzory jsou převáděny do počítače (bezdrátově nebo pomocí např. SD karty). [20] Hlavními přednostmi tohoto systému je schopnost po inicializaci počáteční polohy pracovat nezávisle na vnějších podmínkách, což má široké uplatnění např. u neoptických MoCap systémů, v navigacích, v mobilech a v chytrých hodinkách. Využití inerciálních senzorů se poměrně rozšířilo do mnoha odvětví, a to především díky menším rozměrům, které umožnila technologie MEMS. [21]



Obr. 2.5: Příklad inerciálního systému

## 2.2.2 Akcelerometr

Akcelerometry (viz obrázek 2.6) mají hlavní využití při měření statického gravitačního zrychlení (senzor ve statické poloze a je měřena síla gravitačního zrychlení na daném místě) nebo pro měření dynamického zrychlení (akcelerometr je umístěn například na ruce a je měřeno zrychlení ruky při hodu předmětem). Akcelerometry umožňují měření v jedné, dvou nebo třech rovinách. Nejpoužívanějšími akcelerometry jsou tříosé akcelerometry, které jsou složeny ze soustavy tří akcelerometrů kde každý z akcelerometrů měří zrychlení v jedné ze tří os (x,y,z).

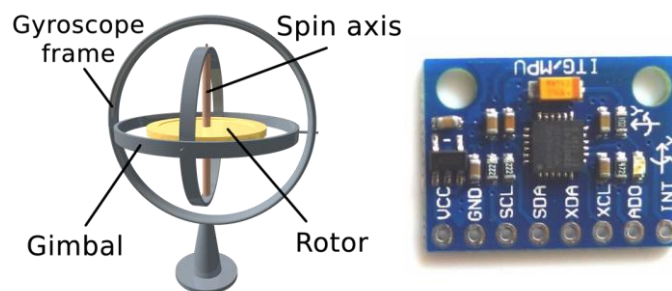
Základními nejběžnějšími druhy akcelerometrů jsou kapacitní MEMS, piezoelektrické a piezorezistivní akcelerometry. [22] [23]



Obr. 2.6: Příklad MEMS akcelerometru (převzato z [24] )

### 2.2.3 Gyroskop

Gyroskop (viz příklady na obrázku 2.7) je zařízení, které pomocí setrvačnicku zachovává středovou polohu rotace v inerciálním prostoru. U původní konstrukce gyroskopu bylo možné měření pouze úhlové výchylky ve vybrané ose. U moderních mechanických, optických nebo MEMS gyroskopů je možné měřit i úhlovou rychlost díky kombinaci gyroskopu a akcelerometru. Nejlepší stabilitu nulové polohy mají, přes poměrně bouřlivý vývoj technologií, gyroskopy mechanické, po nich jsou optické a na konci gyroskopy MEMS. Avšak právě MEMS gyroskopy jsou oproti zbylým dvou podstatně levnější a tím pádem i nejčastěji používané.



Obr. 2.7: Příklad klasického (vlevo) (převzato z [25] ) a MEMS (vpravo) (převzato z [26] ) gyroskopu

### 2.2.4 Magnetometr

Magnetometr (viz příklad na obrázku 2.8) je přístroj určený k měření síly a směru magnetického pole v okolí přístroje. Magnetismus se na planetě Zemi liší místo od místa, což je způsobeno rozdíly magnetického pole Země. Rozdíly magnetického pole Země jsou zapříčiněny odlišnou povahou hornin a vzájemným působením mezi nabitými částicemi ze slunce a magnetosférou planety Země. Magnetometry jsou velmi často používané pro geofyzikální průzkumy s cílem nalezení ložisek železné rudy nebo pro detekci lodních vraků a zasypaných nebo ponořených objektů. [27] Oproti akcelerometrům a gyroskopům, magnetometr nepatří mezi MEMS senzory, což znamená, že magnetometr neobsahuje žádné pohyblivé mikrostruktury. [28]



Obr. 2.8: Příklad MEMS magnetometru (převzato z [29] )

## 2.2.5 Využití inerciálního MoCapu

Inerciální MoCap systém je složen z inerciální měřící jednotky, která je složena z kombinace akcelerometru, gyroskopu a magnetometru. Inerciální měřící jednotky jsou umístěny na oblek (případně exoskelet) ve větším počtu za účelem získu přirozenějších dat. Obleky používané pro inerciální MoCapy mohou být celotělové nebo přímo pro vybraný segment těla (viz příklad na obrázku 2.9). Hlavní výhodou inerciálního MoCap systému je možnost používání bez externích kamer pro záznam (můžou být použity pro přesnější zjištění polohy inerciální měřící jednotky v prostoru), možnost provést měření i ve stísněných prostorech. Hlavní nevýhodou inerciálních MoCap systémů je nižší polohová přesnost, chyby způsobené nedokonalostmi inerciálních senzorů.



Obr. 2.9: Příklad inerciálního celotělového MoCap systému na celé tělo (vlevo) (převzato z [30]) a pouze na paži (vpravo) (převzato z [31])

Inerciální MoCap systémy jsou nejčastěji používány v kombinaci s mechanickými systémy. Inerciální MoCap systémy můžeme rozšířit o mechanické prvky např. exoskelet na kterém jsou sledovány úhlové rozsahy kloubů. Záznam pohybu je v reálném čase přenášen do sledovacího počítače. Úhly mezi klouby jsou vypočítávány z měřených hodnot senzorů umístěných na mechanickém systému. Oproti optickému MoCap systému zde nejsou použity žádné markery, což umožňuje volnější pohyb v exoskeletu bez nutnosti stálého dohledu kamerového systému. [32]

## 2.3 Chyby inerciálních MoCap systémů

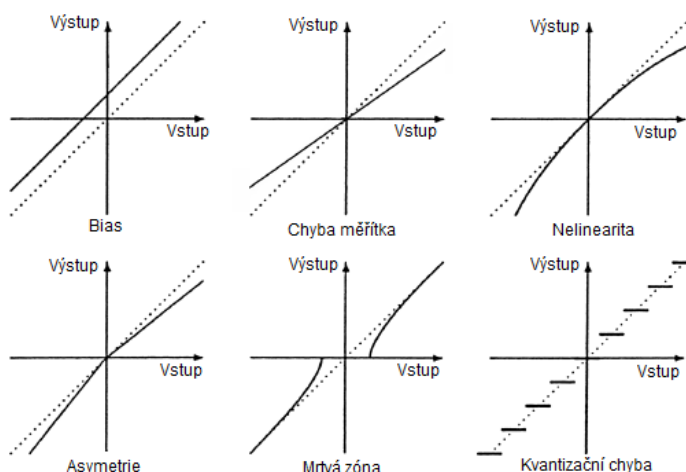
U různých typů senzorů byly zjištěny charakteristické chyby, velikost těchto chyb se liší v závislosti na typu senzoru. Mezi tyto chyby patří bias, chyba měřítka, vychýlení měřících os, nelinearita a náhodný šum (viz obrázek 2.10).

Zdroje těchto chyb jsou složeny ze čtyř komponent: fixního příspěvku, teplotní odchylky, run-to-run odchylky a in-run odchylky.

Fixní příspěvek se na senzoru vyskytuje trvale, ale lze jej optimalizovat pomocí již naměřených dat.

Run-to-run odchylka má různou hodnotu chyby v průběhu činnosti senzoru, avšak na při počátečním spuštění senzoru má tuto hodnotu vždy konstantní.

In-run odchylka má pomalu se měnící tendenci v průběhu činnosti senzoru. Teoreticky lze in-run odchylku minimalizovat zvětšením počtu senzorů, ale v praktickém použití je to obtížné. [33]



Obr. 2.10: Charakteristické chyby senzorů (převzato a upraveno z [34] )

### 2.3.1 Bias

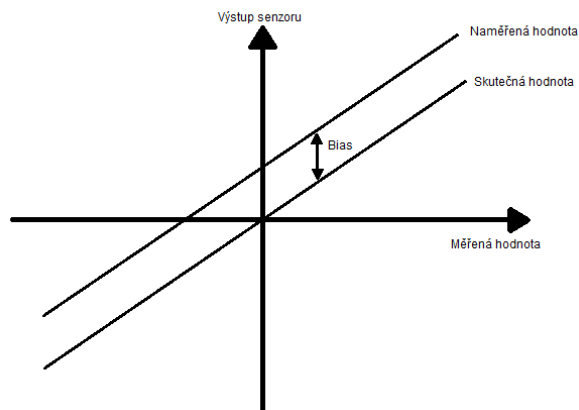
Bias u MEMS senzorů je chyba způsobená posunem měřených hodnot od očekávané hodnoty, a to i když na senzor nepůsobí žádné účinky, např. v případě akcelerometru zrychlení nebo v případě gyroskopu rotace. Jedná se tedy o vychýlení senzoru od skutečné hodnoty, která by měla na výstupu být (viz obrázek 2.11).

V datasheetu od výrobce by mělo být uvedeno, jak citlivé jsou osy senzoru, jaké mají maximální/minimální dovolené odchylky a také jaké jsou možnosti kalibrace daného senzoru. Hodnoty udávané v datasheetu se mohou lišit v závislosti na senzoru [35] .

Odchylku způsobenou biasem lze rozdělit na statickou část (bias offset), náhodnou část (bias drift) a teplotně závislou část.

Díky deterministickému charakteru bias offsetu a teplotně závislé části, lze tyto odchylky poměrně snadno řešit kalibrací.

U bias driftu se jedná o stochastický charakter, a tedy i složitější odstraňování této složky. [36]



Obr. 2.11: Grafické znázornění bias chyby (převzato a upraveno z [37] )

Statickou část (bias offset) je možné zjistit pomocí průměru naměřených hodnot, kdy během měření nebyl senzor ovlivňován rotací. Následná chyba způsobená statickou částí při integraci způsobí odchylku od skutečné hodnoty rostoucí lineárně s časem.

Pro vyhodnocení vlivu dynamické části (bias drift) je nutné mít podrobně nastudované vlastnosti a charakteristiky vybraného senzoru, avšak tuto chybu lze minimalizovat měřením v krátkých časových intervalech.

Vliv bias driftu na naměřená data akcelerometru je velmi nízké oproti bias offsetu, který má na akcelerometr značný vliv. U gyroskopu je to přesně naopak než u akcelerometru, tedy vliv bias offsetu je minimální a vliv bias driftu je značný, což je způsobeno vlivem termo-mechanického bílého šumu [38]

Bias chyba je vyjádřena jako průměrná hodnota bias chyby v jednotkách měření.

### 2.3.2 Chyba měřítka (scale factor error)

Chyba měřítka, jinak známá jako scale factor error, je odchýlení sklonu vstupní a výstupní hodnoty (např. chyba způsobená špatnou reprezentací dat, způsobenou chybně interpretovanými naměřenými daty a časovou osou). Chyba měřítka je, dle výrobců inerciálních senzorů, označována jako citlivost senzoru vzhledem k měřeným datům. U akcelerometru je chyba závislá na velikosti akcelerační síly v měřené ose. U gyroskopu je chyba závislá na velikosti uhlové rychlosti.

Chyba měřítka je udávána v jednotkách měření.

Tuto chybu lze také rozdělit na tři části a to statickou, náhodnou a teplotně závislou. U akcelerometrů a gyroskopů byl zjištěn vliv chyby měřítka nevýznamný. [37]

### 2.3.3 Chyba vychýlených měřících os (chyba ortogonality)

Tato chyba se nachází u všech typů inerciálních senzorů. Jedná se o chybu, kdy jednotlivé osy vůči sobě nejsou kolmé, ale jsou vzájemně posunuté o určitý úhel.

Velikost této chyby by mělo být možné dohledat v datasheetu daného senzoru. K tomuto druhu chyby dochází z důvodů technologických omezení výroby inerciálních senzorů (např. nedokonalé osazení senzoru na desce).

Chyba vychýlených měřících os je udávána ve stupních o kolik je daná osa vyosená vůči jiné ose.

### 2.3.4 Nelinearita

Jedná se o přesnost ideální statické přenosové charakteristiky (přímka) s reálnou kalibrační křivkou. Vyjadřuje se v procentech měřícího rozsahu a vypovídá o maximální odchylce kalibračního bodu od bodu na ideální charakteristice.

### 2.3.5 Náhodný šum (random noise)

Náhodný šum, jinak známý jako random noise, může být způsobený převodem analogového signálu do digitálního signálu. To způsobuje rozdíly mezi amplitudami vzorkovaných hodnot.

Náhodný šum může též vzniknout elektrickým šumem, který má vliv především na MEMS senzory, u kterých jsou poměrně slabé signály.

Chyba náhodného šumu je vyjádřena v jednotkách měření.

Náhodný šum lze minimalizovat naměřením více hodnot, z čehož lze získat průměrnou hodnotu dat. Podle průměrné hodnoty lze následně na data aplikovat filtr pro vyčištění od náhodného šumu. [33]

### 2.3.6 Shrnutí chyb inerciálních MoCap systémů

Z kapitol výše lze usoudit, že největší vliv na měření má bias chyba, chyba vychýlených os a nelinearita (viz shrnutí v tabulce 2.1).

V případě chyby měřítka byl zjištěn nevýznamný vliv při měření akcelerometrem nebo gyroskopem. [37]

Náhodný šum je možné minimalizovat opakovaným měřením, a tedy ho lze také považovat za nevýznamný.

Pro návrh kalibrační aplikace tedy bude důležité navrhnout takovou metodu zahrnující především bias chybu, chybu vychýlených os a nelinearitu.

Tab. 2.1: Shrnutí chyb inerciálních senzorů

název chyby	vliv na akcelerometr	Možnosti minimalizace chyby
Bias	Významný	Odečtem průměrné naměřené hodnoty senzoru v klidu od naměřených hodnot
Chyba měřítka	Nevýznamný	-
Chyba vychýlených os	Významný	Nalezením této hodnoty v datasheetu senzoru a implementace do programu, nebo vypočítání kalibrační matice a vynásobení maticovým součinem kalibrační matice s naměřenými daty
Nelinearita	Významný	Chyba je minimalizována kalibrační maticí při vynásobení maticovým součinem kalibrační matice s naměřenými daty
Náhodný šum (random noise)	Nevýznamný	Chyba je minimalizována měřením více hodnot a z nich získané průměrné hodnoty pro aplikaci filtru



## 2.4 Kalibrace inerciálních MoCap senzorů

U měření senzory je obecně vyžadována vysoká přesnost měření a tím pádem je kladen důraz na správnou kalibraci senzorů. Na senzory lze využít mnoho metod kalibrace, optimální je využívat metody kalibrací popsaných v českých normách ČSN ISO 16063-1. [39]

Na normu ČSN ISO 16063-1 následně navazují níže popsané normy zabývající se již přímo konkrétními typy senzorů a typem kalibrací. Podle normy ČSN ISO 16063-1 lze kalibrační metody rozdělit na primární kalibraci, sekundární kalibraci a kalibraci vlivu prostředí.

Primární kalibrace se zabývá měřením amplitudy nebo rychlosti, kterým je senzor vystaven. Sekundární kalibrace se zabývá porovnáváním kalibrovaného senzoru s kalibrovaným senzorem pomocí primární kalibrace. Kalibrace vlivu prostředí je zaměřena na měření vnějších vlivů (sil) působících na senzor.

Pro správné použití, a i kalibraci senzorů je zapotřebí akcelerometry správně připevnit. Správné připevnění senzoru by mělo být stanoveno výrobcem, avšak pokud tak není uvedeno lze připevnění provést podle normy ČSN ISO 5348, která uvádí doporučení, jakým způsobem lze akcelerometry připevnit.

Použití normy ČSN ISO 5348 je omezeno pouze na akcelerometry, které budou připevněny na pohybující se konstrukci a není použitelná na jiné typy senzorů, jako např. senzory relativního pohybu. [40]

Pro kalibraci senzorů pomocí matematického modelu lze přistoupit k normě ČSN ISO 16063-43, která stanovuje termíny a metody používané pro odhad parametrů v matematických modelech. Popsané metody v normě ČSN ISO 16063-43 úzce souvisí se standardy kalibračních postupů z norem ISO 16063-11, ISO 16063-13, ISO 16063-21 a ISO 16063-22.

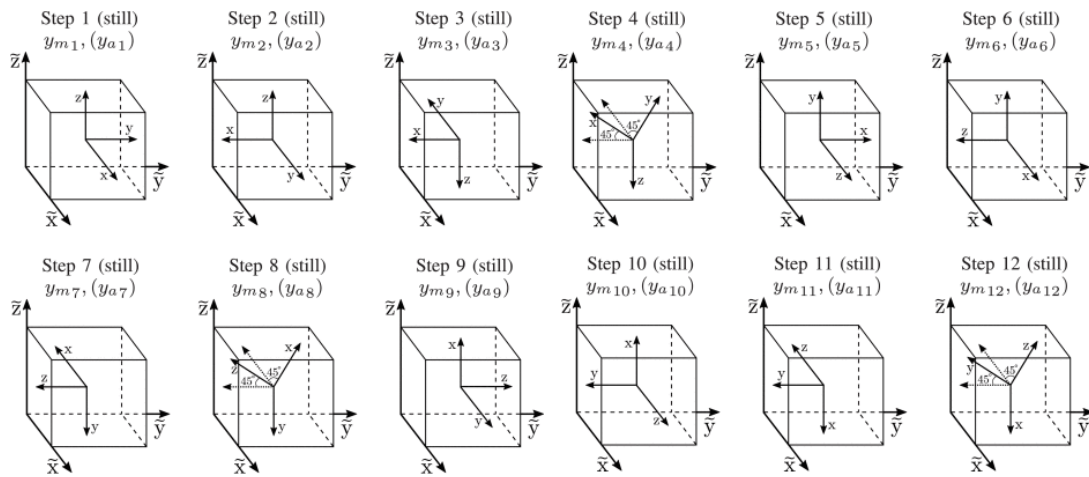
Poskytnutá specifikace v normě ČSN ISO 16063-43 slouží jako rozšíření stávajících postupů a definic uvedených v již zmíněných mezinárodních normách. Hlavním cílem této novější normy je vylepšení kvality kalibrace a měřících aplikací s širokopásmovým vstupem [41]

Kalibračními metodami jsou v tomto případě posloupnosti úkonů za účelem získání dat pro vypočítání kalibračních hodnot. Kalibrační metoda souvisí s několika faktory, přičemž hlavními faktory jsou sensorové složení kalibrované inerciální soustavy, vlastnosti senzorů a vyžadovaná jakost kalibrace. Např. pokud je inerciální soustava složena pouze z akcelerometrů, není nutné měřit rotaci sensorové jednotky, v případě statické kalibrace akcelerometru není potřeba používat rotační tříosý stůl apod.

## 2.5 Metody kalibrace akcelerometru

Pro provedení kalibrace akcelerometru je velmi často využíváno skutečnosti, že pokud je akcelerometr ve statické poloze, tak měří pouze gravitační zrychlení. Gravitační zrychlení je konstantní, nezávisle na orientaci senzoru. Z tohoto pohledu je poté naměřená hodnota akcelerace v ose  $z$  (osa kolmá k zemi) rovna gravitačnímu zrychlení  $9,81 \text{ m/s}^2$  a na zbylých dvou osách je naměřeno nulové zrychlení.

Nejčastěji doporučovaným kalibračním postupem je natáčení akcelerometru do předem definovaných poloh. Na obrázku 2.12 je zobrazen příklad předem definovaných poloh, osy  $X, Y$  a  $Z$  označují souřadnicový systém senzoru a souřadnice  $\bar{X}, \bar{Y}$  a  $\bar{Z}$  označují pevný kalibrační referenční systém. [42]



Obr. 2.12: Příklad 12 definovaných poloh pro kalibraci (převzato a upraveno z [42] )

### 2.5.1 Kalibrace metodou nejmenších čtverců

Tato metoda je založena na vztahu mezi dvěma měřenými veličinami, magnetickým polem a zrychlením. Předpokládá se zde konstantní a přesný vektor intenzity magnetického pole. Magnetické pole země je sice závislé na poloze, ale v případě malého prostoru jej lze považovat za konstantní.

Přesnou hodnotu magnetického pole pro konkrétní místo a čas lze určit pomocí světového magnetického modelu dostupným na stránkách světového magnetického modelu. U magnetického pole je důležité taky dávat pozor na feromagnetické materiály a elektrické proudy ovlivňující magnetické pole. [43]

Při měření akcelerometrem lze předpokládat konstantní zrychlení, čehož lze docílit statickou polohou během měření. U této metody jsou shromažďovány informace o chybách snímačů měřením na plošině, která se otáčí vzhledem k inerciálnímu rámu. Plošina se otáčí po krocích kolem  $x$  a  $y$  osy. Naměřené hodnoty jsou transformovány do matice rotace představující polohu akcelerometru vzhledem k pevnému referenčnímu systému.

U této metody jsou vytvořeny dva modely měření (model magnetického pole a model akcelerometru) ze kterých jsou získány důležité neznámé proměnné jejichž odhad je cílem kalibračního procesu a nezajímavé neznámé proměnné a konstanty, které je potřeba eliminovat. Eliminace nezajímavých neznámých proměnných lze docílit aproximací matice rotace magnetického pole maticí rotace akcelerometru.

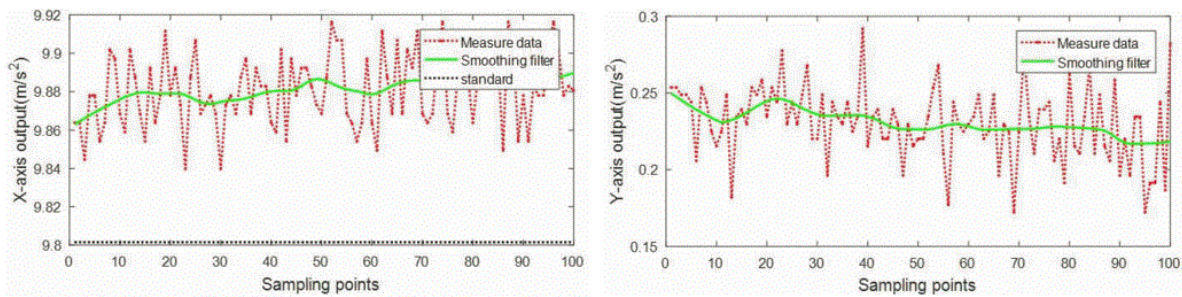
Pomocí aproximace matic a matematickou úpravou je získána lineární soustava rovnic kde jsou proměnné reprezentovány dvanácti rozměrným vektorem. Ze vzniklé lineární soustavy rovnic lze získat jedinečné řešení, podle kterého lze provést potřebnou kalibraci. [44]

### 2.5.2 Kalibrace metodou Levenberg-Marquardova algoritmu

Tato metoda využívá Levenberg-Marquardův (LM) optimalizační algoritmus, který má využití především při řešení nelineárních problémů nejmenších čtverců. LM algoritmus se aplikuje především na prokládání křivek nejmenších čtverců, tedy aby danými páry dat z měření procházela křivka tak, aby součet čtverců odchylek vzdálenosti dat z měření byl co nejmenší.

LM algoritmus je iterativní procedura u které je nutno zadat počáteční hodnotu koeficientu tlumení, se mění při každé iteraci. Pokud dochází k příliš rychlé redukci rozptylu je nutné použít menší koeficient tlumení, v tomto případě se algoritmus blíží Gauss-Newtonově algoritmu. Pokud chyba daná iterací příliš velká, lze koeficient tlumení zvětšit a algoritmus se poté blíží metodě gradientního klesání. Z těchto hledisek lze považovat každou iteraci LM algoritmu za nalezení vhodného koeficientu tlumení.

Díky využití této metody se lze vyhnout singulární matici Gauss-Newtonovy metody a dosáhnout vyšší rychlosti než u metody gradientního klesání. [45] Příklad kalibrace akcelerometru dronu pomocí LM algoritmu je na obrázku 2.13.



Obr. 2.13: Příklad výsledků kalibrace akcelerometru dronu (převzato a upraveno z [45])

### 2.5.3 Kalibrace metodou nelineárního estimátoru parametrů

Tato metoda využívá nelineární a rekurzivní odhad parametrů pomocí statistické linearizace pro rekalibraci akcelerometrů. Tuto metodiku zavádí především z důvodu odstranění nevýhod analytické linearizace a vysokých paměťových nároků na sběr dat.

Hlavní výhodou této metody je schopnost zvládnout velké nejistoty parametrů v nelineárních systémech. Kalibrace je provedena pomocí nescentované transformace (unscented transform). Nescentovaná transformace je matematická funkce sloužící k odhadu výsledku nelineární transformace na pravděpodobnostní rozdělení.

U této metody je zaveden parametr obsahující odhadovaný offset kalibračních parametrů a citlivost modelu akcelerometru. Dalším důležitým prvkem této metody je matice šumu pomocí které lze zavést tzv. faktor zapomínání pro odhad parametrů. Faktorem zapomínání je myšlena možnost odstranění nepodstatných (zašuměných) dat. Přesnost nelineárního odhadu je kontrolována pomocí testu konzistence. Pokud je chyba odhadu nezkreslená a aktuální střední kvadratická chyba odpovídá kovarianci získané z filtru tak se jedná o konzistentní odhad.

Statistický test konzistence filtru (Normalized Estimation Error Squared), lze provést pokud je k dispozici skutečná hodnota stavové proměnné. Při většině měření není k dispozici skutečná hodnota stavové proměnné a vtom případě je nutno provést kontrolu konzistence filtru pomocí normalized Innovation squared value (NIS). NIS má důležitou úlohu při rozpoznávání a odstranění šumu z měření.

Pokud je změřeno rušení (např. intenzivní pohyb akcelerometru), které přesahuje hodnotu NIS předdefinovaného prahu, je tato hodnota zamítnuta a stav filtru se neaktualizuje. Změření abnormální hodnoty vede k přerušení kalibračního algoritmu. [46]

#### **2.5.4 Kalibrace metodou nelineárním optimalizačním problémem**

Tato metoda vychází z předpokladu gravitačního zrychlení, tedy že na statický senzor působí pouze gravitační zrychlení. Modul zrychlení měřený senzorem je poté roven  $G = 9,81 \text{ m/s}^2$  nezávisle na poloze senzoru. Je vyžadováno pouze měření výstupu z MEMS senzoru, který je polohován nejméně do devíti pevných náhodných poloh (viz příklad na obrázku 2.12 výše).

U této metody je předpokládán normalizovaný poměrový výstup akcelerometru a výstup vektoru zrychlení. Normalizovaného poměrového výstupu lze docílit vydělením výstupního napětí napájecím napětím. Vektor zrychlení je definován v lokálním referenčním systému senzoru s osami X,Y ležící na povrchu snímače a osou Z kolmou k povrchu.

Senzor je umístěn do statické náhodné pozice pro kterou je vyhodnocen výstup senzoru. Chyba senzoru je vypočítána jako druhá mocnina rozdílu mezi modulem výstupu a gravitačním zrychlením.

Po provedení všech plánovaných poloh se chyby sečtou do kumulativní chyby, kterou lze vyjádřit jako nelineární funkci parametrů snímače. [47]

#### **2.5.5 Kalibrace pomocí kalibrační matice vytvořené Gram-Schmidtovým ortogonalizačním procesem**

Metoda Gram-Schmidtova ortogonalizačního procesu umožňuje ve vektorovém prostoru nalézt ortonormální bázi podprostoru. Ortonormální báze podprostoru je báze daného vektorového prostoru jejíž vektory jsou normované, to pro jednotlivé vektory báze znamená, že jsou jednotkové a jsou na sebe vzájemně kolmé.

Aplikací matice vzniklé Gram-Schmidtovým ortogonalizačním procesem je možné opravit vychýlení vstupních vektorů. Aplikaci Gram-Schmidtova ortogonalizačního procesu si lze zjednodušeně představit jako pozvolné rovnání vektoru, vstupujícího do procesu, do kolmé roviny.

Proces je prováděn tak aby se měnil pouze úhel mezi vektory bez vlivu na velikost daných vektorů. [48] [49] Gram-Schmidtův ortogonalizační proces byl využit při hledání chyb orientace akcelerometru, používaný při rozpoznávání lidské aktivity [50]

## 2.6 Metody kalibrace gyroskopu

V případě gyroskopu je kalibraci složitější, a to především proto, že není k dispozici žádná vhodná referenční rotace. Valná většina metod kalibrace gyroskopu mají jednu věc společnou, a to využití pomocné referenční rotace ke kalibraci.

### 2.6.1 Metoda využívající referenční zemskou rotaci

Tato metoda odstraňuje hlavní části deterministických chyb senzoru pomocí vícepolohové kalibrační metody. U této metody není vyžadována speciální montáž a kompenzuje chybu měřítka a nesouosost os.

Jako referenční rovina je zde použit otočný stůl, který využívá jako referenci zemskou rotaci. Zemská rotace je ale příliš slabá aby se na senzoru projevila oproti šumu. Se stolem se tedy musí otáčet tak aby vznikl dostatečně silný signál.

Tato metoda byla porovnáována s tradičními kalibračními metodami rychlostního testu a šestipolohové metody, přičemž tato metoda vyšla jako rychlejší. [51]

### 2.6.2 Metoda využívající referenční rotaci kalibrovaného akcelerometru

Tato metoda navrhuje využití již zkalibrovaného akcelerometru na kalibraci gyroskopu. To je možné především díky čipové technologii kdy jak akcelerometr tak gyroskop jsou na podobné křemíkové desce.

Skutečné testy nesouososti akcelerometru a gyroskopu zjistili maximální nesouosost akcelerometru 0,0186 a gyroskopu 0,0003. Nejjednodušší způsob určení zkreslení senzoru je udržet gyroskop ve statické poloze a vzhledem k nízkému vlivu zemské rotace by měl být výstup senzoru roven zkreslení.

Po statickém měření je provedeno dynamické měření, kde je gyroskop otáčen pod předem definovanými úhly. Z měření pro každý úhel lze odvodit lineární rovnice o dvou neznámých. Na vzniklé lineární rovnice je následně aplikována metoda nejmenších čtverců pomocí které jsou nalezeny odchylky. Tyto vypočítané chyby lze následně porovnat automatickým algoritmem s kalibrovaným akcelerometrem a postupně zkalibrovat gyroskop. [52]

### 2.6.3 Metoda využívající kalibrovaného magnetometru

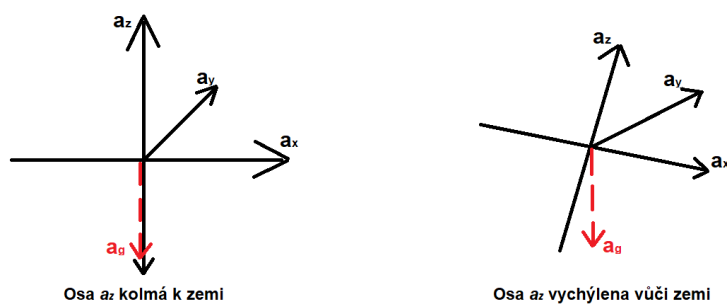
Tato metoda využívá ke kalibraci zkalibrovaný magnetometr a kalibrovaný gyroskop je umístěn v homogenním magnetickém poli. Pokud je dostatečná rotační excitace může být vektor homogenního magnetického pole vhodná reference pro kalibraci gyroskopu.

Touto metodou lze odstranit chyby měřítka, neortogonalitu, vychýlení os a nesouosost. [53]

## 2.7 Metody pro výpočet relativních úhlů senzorů

### 2.7.1 Metoda určení relativního úhlu akcelerometru

Relativní úhel natočení (anglicky relative angle of rotation zkráceně RAR) akcelerometru lze vypočítat pomocí goniometrických funkcí [54]. Za předpokladu, že bude akcelerometr v klidu, tak na něj působí pouze gravitační zrychlení, čehož lze využít pro určení RAR. Pokud bude akcelerometr natočen jednou osou (např. osou  $y$ ) kolmo k zemi, tak by tato osa měla ukazovat hodnotu gravitačního zrychlení a zbylé osy nulové zrychlení. Pokud bude akcelerometr vychýlený, tak bude na ose kolmé k zemi slabší gravitační zrychlení a na zbylých dvou osách nebo pouze jedné ose se projeví gravitační zrychlení (viz příklad na obrázku 2.14).



Obr. 2.14: Příklad kolmého natočení osy  $a_z$  akcelerometru (vlevo) a chybného natočení osy  $a_z$  akcelerometru (vpravo)

Pro výpočet RAR akcelerometru se využívají pouze inverzní trigonometrické funkce, a to především proto, že inverzní funkce mají omezený definiční obor, díky čemuž je dosaženo prosté funkce.

## 2.7.2 Metoda určení relativního úhlu gyroskopu

Měření gyroskopem vrací hodnoty o úhlové rychlosti neboli údaj o rychlosti otáčení. Pro výpočet relativního úhlu natočení gyroskopu je nutné integrovat průběh rychlosti. Výsledkem integrace není absolutní úhel, ale změna úhlu oproti jeho předchozí hodnotě. Za předpokladu nulového počátečního úhlu je poté výsledkem skutečný úhel natočení. [55]

$$RAR_G = \int_0^t a_t \cdot t dt \quad (1)$$

Kde,

$a_t$  – zrychlení v čase  $t$  [ $m \cdot s^2$ ],

$t$  – čas [s],



## 3 Cíle práce

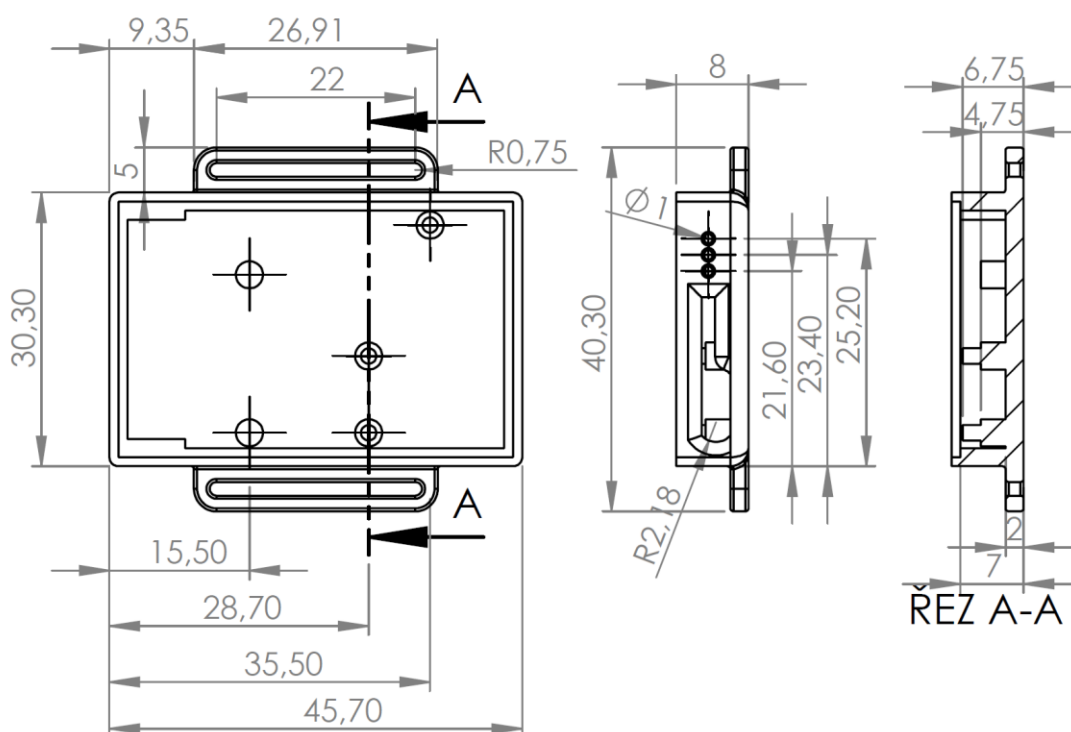
### **Ze zadání práce lze definovat následující cíle:**

- Návrh softwaru pro konfiguraci a kalibraci inerciálních senzorů s následujícími požadavky
  - Navrhnout uživatelsky přívětivé rozhraní pro práci s kalibračním programem
  - Snadné připojení senzoru k programu
  - Grafické zobrazení naměřených hodnot v reálném čase
  - Možnost sledování polohy senzoru v reálném čase
  - Na základě neměřených hodnot je poskytnuta zpětná vazba o poloze senzoru, a zda poloha odpovídá požadavku
  - Možnost uložit vytvořenou matici a nahrát uloženou kalibrační matici
  - Možnost sledování důležitých parametrů
  - Možnost porovnat důležité parametry před a po kalibraci
- Otestování a statistické zhodnocení navržených metod a aplikace pro kalibraci

## 4 Metody

### 4.1 Návrh pouzdra pro pevné umístění kalibrovaného senzoru

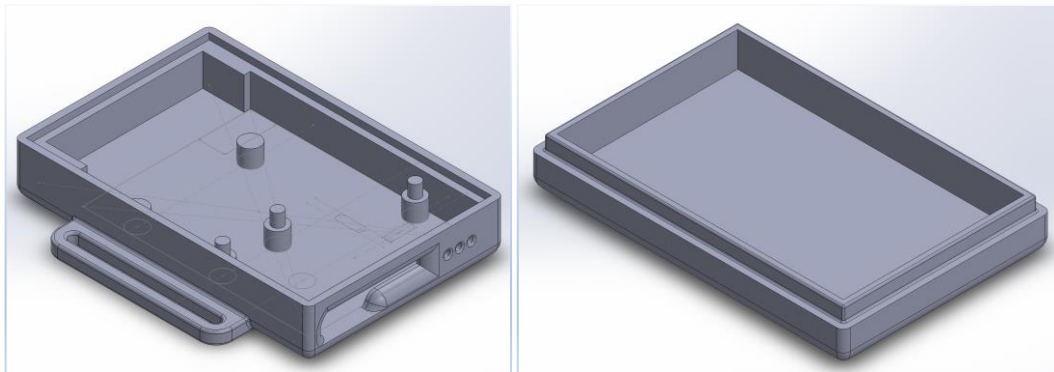
Pro správné umístění akcelerometru je navrženo unikátní pouzdro pro umístění senzoru v souladu s normou ČSN ISO 5348 [40]. V programu SolidWorks student edition je vytvořen návrh pouzdra (viz obrázek 4.1), které bude vytisknuto pomocí 3D tisku. Specifické rozměry pouzdra jsou navrženy přesně podle rozměrů akcelerometru, čímž je dosaženo pevného umístění senzoru.



Obr. 4.1: Návrhový výkres spodního dílu pouzdra, ve kterém je umístěn senzor (vytvořeno pomocí programu SolidWorks student edition)

Samotné pouzdro se skládá ze dvou kusů, tzn. spodního dílu a víčka (viz obrázek 4.2). Do spodního dílu bude pevně vložen senzor. Ve spodním díle jsou předpřipravené sloupky pro přesné umístění senzoru.

Víčko je dimenzováno pro vložení baterie napájející senzor. Mezi víčkem a spodním dílem je vytvořen dostatečný prostor tak, aby bylo možné oba díly pevně sestavit do sebe. Při sestavení je mezi oběma díly milimetrová drážka určená pro aplikaci lepidla zajišťující nerozpadnutí pouzdra např. vlivem teploty.



Obr. 4.2: Navržené pouzdro pro umístění senzoru. Spodní díl, ve kterém jsou umístěny senzory (vlevo) a víčko pouzdra (vpravo)

## 4.2 Návrh určení relativního úhlu akcelerometru

Určení relativního úhlu akcelerometru lze vypočítat pomocí trigonometrické funkce arkus kosinus (viz rovnice (1)), do které lze dosadit naměřené hodnoty zrychlení v rámci jednotlivých os. Pokud by osy mezi sebou svíraly nulový úhel, tak to znamená, že se jedná o rovnoběžky a nejspíše se jedná o implementační chybu, jelikož dvě osy akcelerometru nemohou svírat nulový úhel. Dané osy tedy, musí mezi sebou svírat nenulový úhel.

$$RAR_{ag} = \arccos^{-1}\left(\frac{a}{g}\right) \quad (1)$$

Kde,

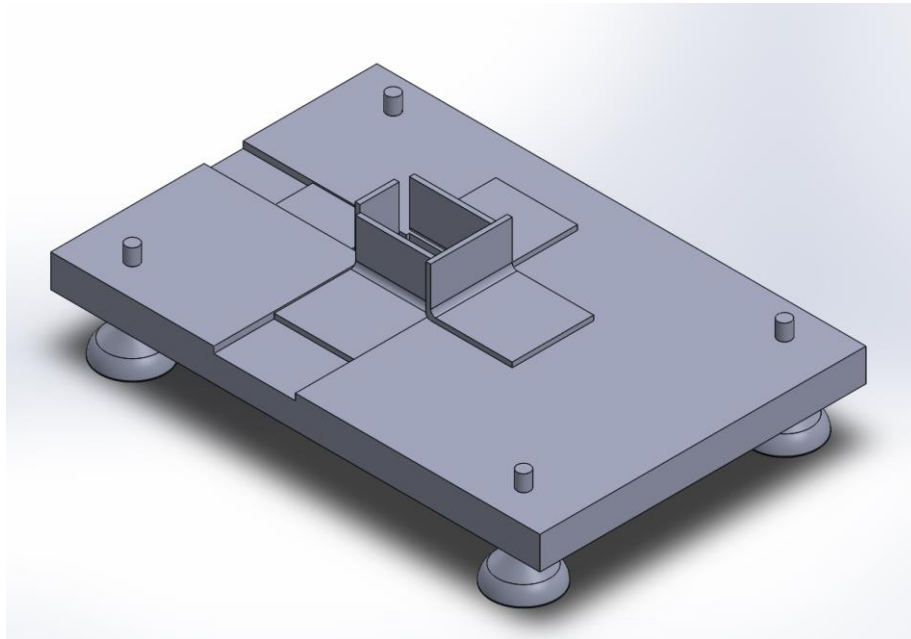
$a$  – zrychlení v ose kolmé na gravitační zrychlení [ $m \cdot s^2$ ],

$g$  – zrychlení v ose gravitačního zrychlení [ $m \cdot s^2$ ].

$RAR_{ag}$  – vypočítaný relativní úhle mezi osou  $a$  a  $g$  [°]

### 4.3 Návrh kalibrační metody

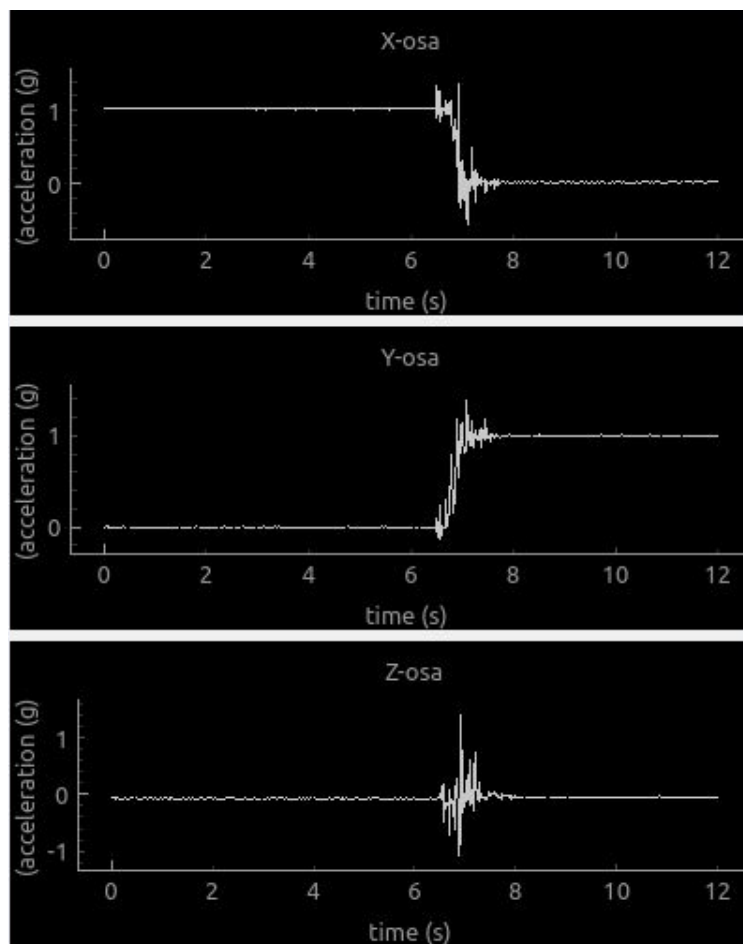
Pro provedení kalibrace je navržen specifický kalibrační stůl (viz obrázek 4.3), který má v sobě přesně definovaný výřez určený pro umístění navrženého pouzdra z kapitoly 4.1 se senzorem uvnitř. Pro správné seřízení kalibračního stolu do vodorovné polohy bude stůl opatřen nastavitelnými nožičkami a vodováhami, pro přesnější nastavení kalibračního stolu lze použít digitální úhломěr.



Obr. 4.3: Model navrženého kalibračního stolu

Navržená kalibrační metoda vychází ze skutečnosti, že pokud je senzor v klidové poloze působí na něj pouze gravitační zrychlení. Tedy pokud je jedna z měřících os senzoru kolmo k zemi působí gravitační síla pouze na tuto osu a na zbylé osy působí minimálně v ideálním případě vůbec. Tuto skutečnost lze ověřit pomocí měření zrychlení senzorem v klidové poloze s jednou ze tří os kolmo na zem (viz obrázek 4.4).

Přesnost navržené kalibrační metody je závislá na rozměrově správně navrženém pouzdru pro kalibrovaný senzor a na správně vyváženém kalibračním stole. Správně vyvážený kalibrační stůl je jehož horní plocha je vodorovná tzn. bublinka vodováhy, umístěné na kalibračním stole, se pohybuje co nejbližší středu vodováhy.

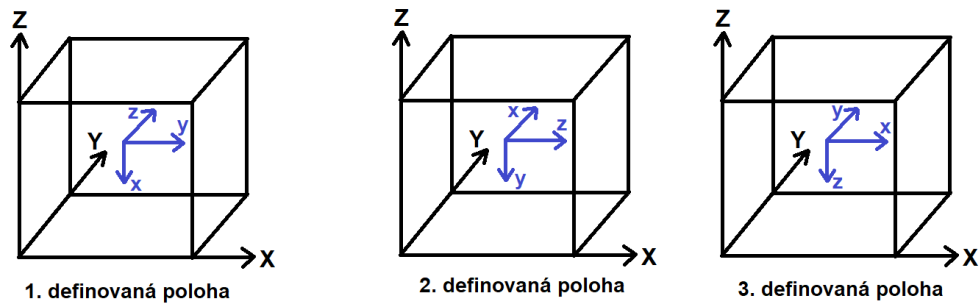


Obr. 4.4: Příklad naměření hodnot akcelerometrem v prvních 7 vteřinách osou  $x$  kolmo k zemi a následně od 8 vteřiny osou  $y$  kolmo k zemi

Pro úspěšné provedení kalibrační metody je důležité mít správně definované polohy umístění senzoru. Kalibrační metoda navržená v této práci vyžaduje pouze tři definované polohy. Jedná se o polohy (viz obrázek 4.5):

1. osou  $X$  k zemi,
2. osou  $Y$  k zemi a
3. osou  $Z$  k zemi.

Pokud je senzor umístěn obráceně, je naměřena hodnota se záporným znamínkem. Pro tuto kalibrační metodu je nezbytné vždy umístit senzor v korektní poloze, v opačném případě nelze zaručit požadované výsledky.



Obr. 4.5: Definované polohy pro kalibraci na kalibračním stole

### 4.3.1 Návrh kalibrační matice pracující s naměřenými hodnotami

Pro správné použití metod je nutné vytvořit odpovídající model akcelerometru. Výstup akcelerometru lze definovat jako:

$$a = [a_x, a_y, a_z] \quad (2)$$

Kde  $a_x$  je naměřená hodnota v ose  $x$ ,  $a_y$  je naměřená hodnota v ose  $y$  a  $a_z$  je naměřená hodnota v ose  $z$

Další důležitou částí je model obsahující chyby senzoru, který lze definovat jako [47] :

$$S = \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{bmatrix} \quad (3)$$

Kde hodnoty na diagonále ( $S_{xx}$ ,  $S_{yy}$ ,  $S_{zz}$ ) vyjadřují chybu měřítka napříč všemi osami, hodnoty mimo diagonálu vyjadřují chybu vychýlení os. U ideálního senzoru jsou předpokládány hodnoty mimo diagonálu rovny nule.

V předchozím modelu není zahrnuta bias chyba, tu lze vyjádřit jako [47] :

$$B = \begin{bmatrix} a_{nx} \\ a_{ny} \\ a_{nz} \end{bmatrix} - \begin{bmatrix} b_{0x} \\ b_{0y} \\ b_{0z} \end{bmatrix} \quad (4)$$

Kde hodnoty  $a_{nx}$ ,  $a_{ny}$  a  $a_{nz}$  jsou naměřené hodnoty, hodnoty  $b_{0x}$ ,  $b_{0y}$  a  $b_{0z}$  jsou hodnoty vyjadřující posun osy v dané ose.

Kalibrační model potom lze vytvořit z navržených modelů (3)-(5) následovně:

$$\begin{bmatrix} a_{kx} \\ a_{ky} \\ a_{kz} \end{bmatrix} = \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{bmatrix} * \left( \begin{bmatrix} a_{nx} \\ a_{ny} \\ a_{nz} \end{bmatrix} - \begin{bmatrix} b_{0x} \\ b_{0y} \\ b_{0z} \end{bmatrix} \right) \quad (5)$$

Kde hodnoty  $a_{kx}$ ,  $a_{ky}$  a  $a_{kz}$  jsou kalibrované hodnoty.

Ve vzniklém modelu se nachází 12 neznámých, které je nutné vypočítat a dosadit do modelu. Navržený model (6) lze upravit do následujícího tvaru [47] :

$$\begin{cases} a_{kx} = S_{xx}(a_{nx} - b_{0x}) + S_{xy}(a_{ny} - b_{0y}) + S_{xz}(a_{nz} - b_{0z}) \\ a_{ky} = S_{yx}(a_{nx} - b_{0x}) + S_{yy}(a_{ny} - b_{0y}) + S_{yz}(a_{nz} - b_{0z}) \\ a_{kz} = S_{zx}(a_{nx} - b_{0x}) + S_{zy}(a_{ny} - b_{0y}) + S_{zz}(a_{nz} - b_{0z}) \end{cases} \quad (6)$$

Hodnoty chybového modelu S (rovnice (4)), lze získat pomocí měření, během kterého je akcelerometr natočen do tří poloh, vždy jednou z os kolmo k zemi. Pomocí relativních úhlů z kapitoly 4.1, lze zajistit přesné umístění senzoru tak aby byla chybovost způsobená vyosením co nejmenší. V ideálním případě by měl chybový model S vypadat ve tvaru:

$$S = \begin{bmatrix} g & 0 & 0 \\ 0 & g & 0 \\ 0 & 0 & g \end{bmatrix} \quad (7)$$

Hodnoty chyby biasu lze zjistit buďto přímo z datasheetu daného akcelerometru nebo pomocí opakovaného měření v klidové poloze akcelerometru. Pokud akcelerometr v klidové poloze vrací nenulové hodnoty jedná se o bias chybu. Tato naměřená chyba se následně zprůměruje a odečte od naměřených hodnot. Ideální bias chyba je nulová tedy:

$$B = \begin{bmatrix} a_{nx} \\ a_{ny} \\ a_{nz} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (8)$$

## 4.4 Návrh implementace navržených metod

Pro implementaci kalibračního programu byl zvolen jazyk Python. Tento jazyk byl zvolen pro jeho univerzálnost, neboť umožňuje vytvářet aplikace pro většinu platform. Další velkou výhodou jazyka Python je mnoho volně dostupných modulů, nástrojů a knihoven usnadňující složité programování nebo složité výpočty.

V navrhovaném programu budou vytvořeny funkce zpracovávající naměřená data a využity volně dostupné moduly, nástroje a knihovny Pythonu.

Samotné rozhraní bude navrženo pomocí knihovny PyQt5, kterou lze nainstalovat v kterémkoliv programovacím prostředí podporující python. Komunikace s měřicím senzorem bude zprostředkována buďto pomocí datového kabelu nebo pomocí bezdrátové komunikace např. pomocí standartu Bluetooth. Připojení k senzoru bude provedeno pomocí navržené funkce `connectDevice()` a data ze senzoru jsou posílány a zpracovávány v reálném čase po měřených úsecích. V programu budou následně pomocí navržené funkce `createCalibrationMatrix()` vytvořena kalibrační matice (tato matice má tvar jako v kapitole 5 rovnice (6)). Při připojení senzoru k programu bude mít uživatel možnost importu již vytvořené kalibrační matice, pokud ještě nebyla kalibrační matice vytvořena je nutno postupovat přesně podle návodu pro vytvoření kalibrační matice. Při vytváření kalibrační matice je důležité mít správně vyvážený kalibrační stůl dle návodu (viz příloha A). Na kalibrační stůl je pokládán senzor ve třech definovaných polohách. Po dokončení kalibrační metody je vytvořena kalibrační matice, která se následně aplikuje na naměřená data. Kalibrovaná data jsou následně vykreslena do grafů v rozhraní. Během měření jsou vypočítávány důležité parametry zobrazované v okně sledované parametry. Z výsledku provedené kalibrace lze následně zjistit o kolik byly chyby (chyba ortogonality, chyba měřítka a bias chyba) minimalizovány.

Základní funkce pro navrhovaný kalibrační program jsou:

### **createGUI():**

`createGUI()` je hlavní funkcí celého programu. Zajišťuje chod hlavního rozhraní a umožňuje snadnou obsluhu celého programu. Tato funkce by měla být založena na knihovně PyQt5, která umožňuje vytváření interaktivních oken s možností vložení ovládacích prvků. Rozvržení prvků v rozhraní by mělo být pomocí dynamické mřížky, umožňující každému prvku přidělit takový prostor aby byl korektně umístěn.

### **connectDevice():**

Funkce `connectDevice()` bude umožňovat v první iteraci skenování dostupných Bluetooth zařízení v okolí měření. Poté co je skenování dokončeno slouží funkce `connectDevice()` k párování s konkrétním zařízením. Pro spárování konkrétního zařízení je nutné znát MAC adresu zařízení, kterou lze získat z předchozí iterace skenování. Pokud



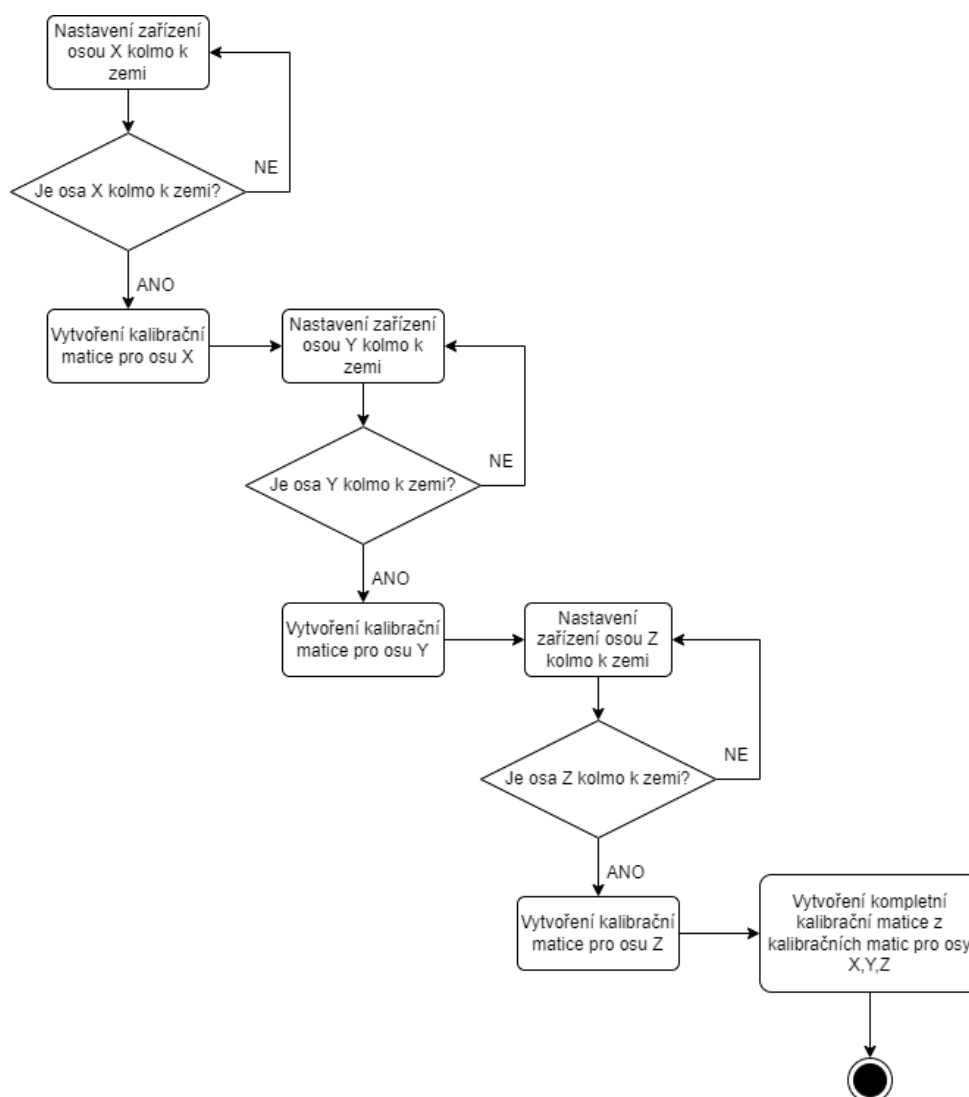
bude MAC adresa zadána správně je zařízení spárováno, pokud je MAC adresa zadána špatně je uživatel požádán o opravu MAC adresy.

#### **measureData():**

Funkce `measureData()` načítá data naměřené akcelerometrem a ukládá je do proměnných v programu. Tato funkce poběží v cyklu a bude ukládat data dokud nebude cyklus ukončen. Po ukončení cyklu budou naměřená data vykreslena do grafů. Naměřená data budou dále použity pro vytváření kalibrační matice a pro výpočet důležitých parametrů.

#### **createMatrix():**

Funkce `createMatrix()` načítá naměřená data a z naměřených dat vypočítává kalibrační matici. Kalibrační matice bude vytvořena pomocí Gram-Schmidtova ortogonalizačního procesu. Vytvořená matice touto metodou bude mít tvar jako v kapitole 4.4.1 vzorec (4). Pro korektní vytvoření kalibrační matice je nutné přesné dodržení kalibrační procedury. Postup pro vytvoření kalibrační je tvořen třemi kroky při kterých je zařízení polohováno do tří definovaných poloh. Práci algoritmu vytvářející kalibrační matici lze popsat jako sled postupných kroků (viz stavový diagram na obrázku 4.6).



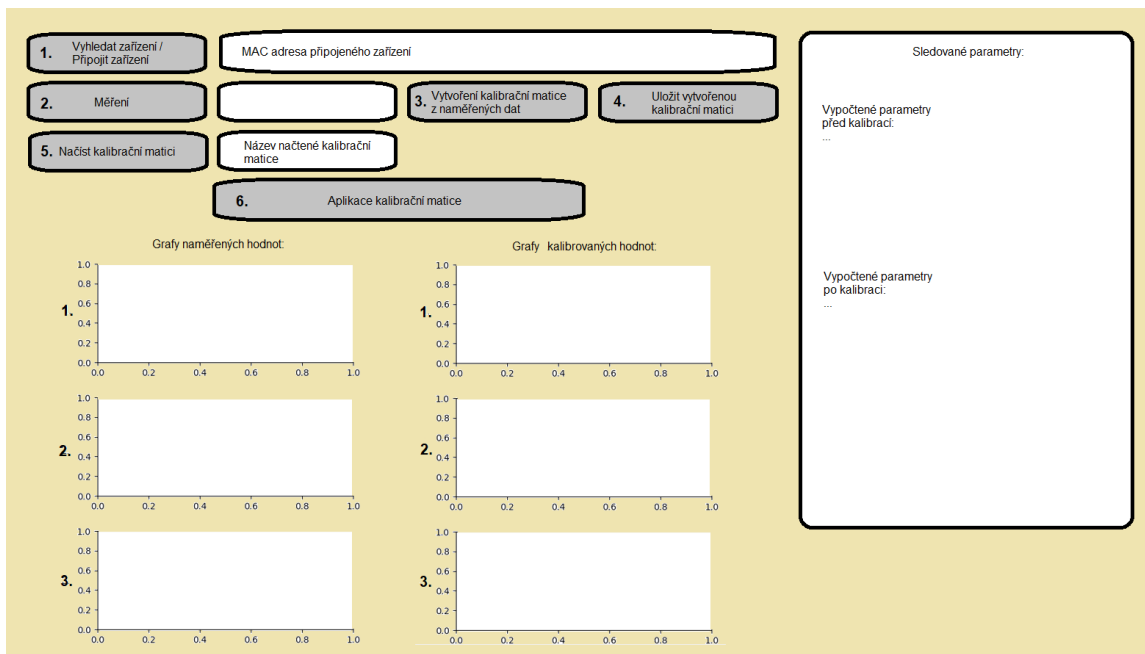
Obr. 4.6: Stavový diagram pro vytvoření kalibrační matice

### **applyMatrix():**

Funkce `applyMatrix()` provede aplikaci vytvořené kalibrační matice na naměřená data. Kalibrovaná data budou následně vykreslena do grafu, kde lze porovnat, zda je mezi daty očividný rozdíl či nikoliv. Z kalibrovaných dat budou také vypočítány důležité sledované parametry, ze kterých lze porovnat, zda došlo ke zlepšení měření.

## 4.5 Návrh implementace uživatelského rozhraní

Uživatelské rozhraní pro kalibraci MoCap senzorů bude navrženo pomocí knihovny PyQt5 v programovacím jazyce Python (viz obrázek 4.7). Uživatelské rozhraní bude vytvořené pomocí dynamické mřížky umožňující uživateli libovolně rozšiřovat okno rozhraní bez vlivu na vnitřní rozpoložení prvků. Samotné uživatelské rozhraní bude obsahovat různá tlačítka, grafy a textová pole, zajišťující uživatelskou přívětivost a snadné ovládání kalibračního programu.



Obr. 4.7: Obrázek návrhu uživatelského rozhraní

Uživatelské rozhraní obsahuje tlačítka:

### Vyhledat zařízení / Připojit zařízení (tlačítko č.1)

Tlačítko vyhledat zařízení / připojit zařízení bude umožňovat provedení skenování dostupných Bluetooth zařízení v okolí a vypsání dostupných zařízení do parametrického okna sledovaných parametrů. Nalezená dostupná zařízení jsou vypsána s MAC adresou a názvem zařízení.

Další funkcí tlačítka vyhledat zařízení / připojit zařízení je možnost připojit vybrané zařízení. Pro připojení vybraného zařízení je nutné zkopírovat MAC adresu ze sledovaných parametrů do textového řádku MAC adresa připojeného zařízení. Pokud je MAC adresa správně zadána je připojení zařízení úspěšně provedeno, tlačítko pro vyhledání zařízení / připojit zařízení zešedivěno a MAC adresa připojeného zařízení je vypsána v textovém řádku Mac adresa připojeného zařízení. Jestliže zařízení není úspěšně připojeno je zapotřebí zkontrolovat, zda je MAC adresa správně zadána a opakovat připojení.

### **Měření (tlačítko č.2)**

Po stisknutí tlačítka měření bude spuštěn cyklus, který přijímá z připojeného zařízení měřená data a ukládá je do proměnné. Měření probíhá až dokud uživatel neukončí probíhané měření pomocí definované klávesy v programu (např. klávesa „f“). Po ukončení měření jsou naměřená data uložena do globální proměnné, která uchovává veškerá naměřená data po celou dobu spuštěného programu.

Další funkcí, která je provedena po ukončení měření je výpočet sledovaných parametrů (výčet sledovaných parametrů je popsán níže v podkapitole sledované parametry) před kalibrací a zobrazení těchto parametrů v parametrovém okně.

### **Vytvoření kalibrační matice z naměřených dat (tlačítko č.3)**

Tlačítko vytvoření kalibrační matice z naměřených dat slouží k vytvoření kalibrační matice z naměřeného bloku dat. Blok dat je definován jako naměřená data od doby stisknutí tlačítka měření až po ukončení měření. Kalibrační matice je vytvořena ve třech krocích pokaždé v jiné definované poloze. Pro správné vytvoření kalibrační matice je důležité správně postupovat podle návodu, který je součástí příloh této práce.

### **Uložit vytvořenou kalibrační matici (tlačítko č.4)**

Vytvořenou kalibrační matici je možné uložit pomocí tlačítka uložit vytvořenou kalibrační matici. Kalibrační matice je uložena ve formátu .csv do adresáře kde se nachází samotný program. V csv. souboru je uložena kalibrační matice a bias chyba, tak aby ji bylo možné lehce a efektivně nahrát zpátky do programu.

### **Načíst kalibrační matici (tlačítko č.5)**

Pomocí tlačítka načíst kalibrační matici lze načíst uloženou kalibrační matici z předchozích chodů kalibračního programu. Pokud uložená matice nebyla modifikována pomocí jiného programu nebo editoru tak bude její načtení bezproblémové a lze ji rovnou aplikovat na měřená data.

### **Aplikace kalibrační matice (tlačítko č.6)**

Vytvořenou nebo načtenou kalibrační matici lze na naměřená data aplikovat pomocí tlačítka aplikace kalibrační matice. Aplikace kalibrační matice provede odečtení bias chyby od naměřených dat a takto upravená data následně pomocí maticového součinu znásobí s kalibrační maticí podle vzorce (6) z kapitoly 4.4.1.

Další funkcí, která je provedena po aplikaci kalibrační matice je vypočítání sledovaných parametrů (výčet sledovaných parametrů je popsán níže v podkapitole sledované parametry) po kalibraci a zobrazení těchto parametrů v parametrovém okně.

V uživatelském rozhraní jsou umístěny grafy:

### **Graf naměřených hodnot č.1**

U grafu naměřených hodnot s číslem 1. jsou zobrazeny naměřené hodnoty pro  $x$  osu senzoru. Data jsou zobrazeny jakožto závislost zrychlení na času kde na ose  $x$  je reprezentován čas v sekundách a na ose  $y$  jsou reprezentovány naměřené hodnoty senzorem na ose  $y$  v normovaných jednotkách  $g$ .

### **Graf naměřených hodnot č.2**

U grafu naměřených hodnot s číslem 2. jsou zobrazeny naměřené hodnoty pro  $y$  osu senzoru. Data jsou zobrazeny jakožto závislost zrychlení na času kde na ose  $x$  je reprezentován čas v sekundách a na ose  $y$  jsou reprezentovány naměřené hodnoty senzorem na ose  $y$  v normovaných jednotkách  $g$ .

### **Graf naměřených hodnot č.3**

U grafu naměřených hodnot s číslem 3. jsou zobrazeny naměřené hodnoty pro  $z$  osu senzoru. Data jsou zobrazeny jakožto závislost zrychlení na času kde na ose  $x$  je reprezentován čas v sekundách a na ose  $y$  jsou reprezentovány naměřené hodnoty senzorem na ose  $z$  v normovaných jednotkách  $g$ .

### **Graf kalibrovaných hodnot č.1**

U grafu kalibrovaných hodnot s číslem 1. jsou zobrazeny kalibrované hodnoty pro  $x$  osu senzoru. Data jsou zobrazeny jakožto závislost zrychlení na času kde na ose  $x$  je reprezentován čas v sekundách a na ose  $y$  jsou reprezentovány naměřené hodnoty senzorem na ose  $x$  v normovaných jednotkách  $g$ .

### **Graf kalibrovaných hodnot č.2**

U grafu kalibrovaných hodnot s číslem 2. jsou zobrazeny kalibrované hodnoty pro  $y$  osu senzoru. Data jsou zobrazeny jakožto závislost zrychlení na času kde na ose  $x$  je reprezentován čas v sekundách a na ose  $y$  jsou reprezentovány naměřené hodnoty senzorem na ose  $y$  v normovaných jednotkách  $g$ .

### **Graf kalibrovaných hodnot č.3**

U grafu kalibrovaných hodnot s číslem 3. jsou zobrazeny kalibrované hodnoty pro  $z$  osu senzoru. Data jsou zobrazeny jakožto závislost zrychlení na času kde na ose  $x$  je reprezentován čas v sekundách a na ose  $y$  jsou reprezentovány naměřené hodnoty senzorem na ose  $z$  v normovaných jednotkách  $g$ .

Rozsah měření není omezen, ale správně naměřené hodnoty jsou normované a měli by být v rozsahu  $-1 g$  až  $+1 g$ , pokud jsou naměřené hodnoty výrazně odlišné od tohoto rozsahu je velice pravděpodobné že došlo k chybě při aplikaci kalibrační procedury a je doporučeno celé měření začít od začátku.

### **Sledované parametry**

Další důležitou částí zobrazovanou v uživatelském rozhraní je výčet sledovaných parametrů mezi sledované parametry jsou zařazeny:

**Korekce polohy** – Tento údaj udává, zda je senzor umístěn ve správné poloze. Podle tohoto údaje bude uživatel schopen umístit zařízení do správné polohy.

**Bias** – Tento údaj bude zobrazen pro každou osu (tedy  $x$ ,  $y$  a  $z$ ) a vyjadřuje o kolik je výstup senzoru vychýlen od skutečné hodnoty, která by měla být na výstupu. Tato hodnota vyjadřuje průměrnou bias chybu senzoru.

**Maximální naměřená hodnota** – Jedná se o maximální naměřenou hodnotu na jednotlivých osách  $x$ ,  $y$ ,  $z$ .

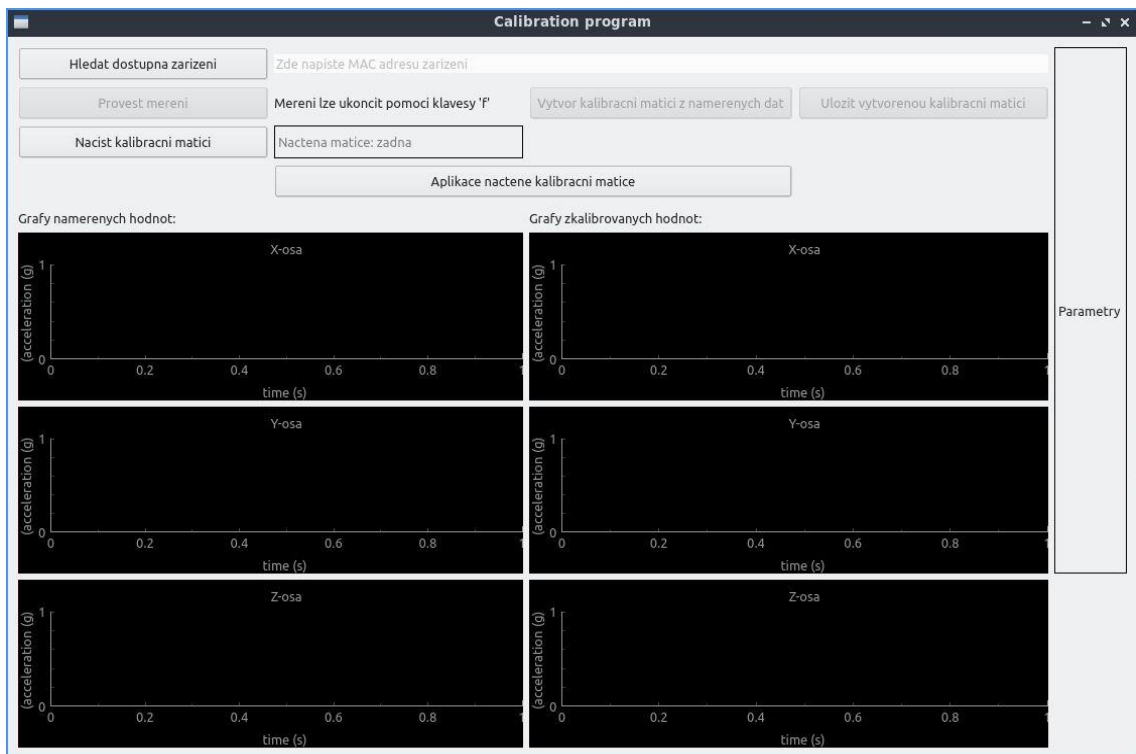
**Minimální naměřená hodnota** – Jedná se o minimální naměřenou hodnotu na jednotlivých osách  $x$ ,  $y$ ,  $z$ .

**Úhel mezi osami** – Tento údaj vyjadřuje jaký úhel mezi sebou svírají osy senzoru. Tato hodnota je počítána vždy vůči ose kolmé k zemi (např. osa  $x$  je kolmo k zemi počítané úhly jsou mezi osami  $x$ ,  $y$  a mezi osami  $x$ ,  $z$ ).

# 5 Výsledky

## 5.1 Realizace metod

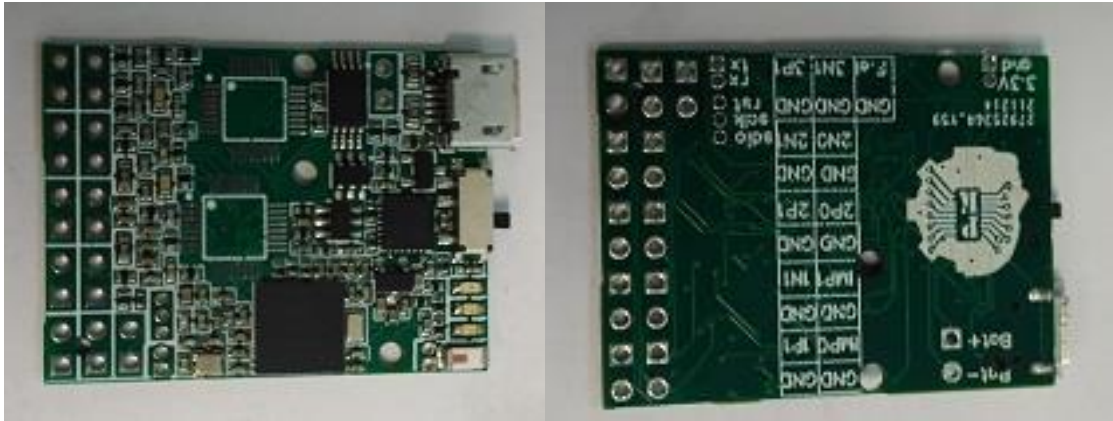
Vytvořený program pro měření a kalibraci senzorů je vytvořen v programovacím prostředí Visual Studio Code v jazyce Python. Pomocí knihovny PyQt5 je navrženo celé grafické rozhraní pro snadné ovládání, u které je pomocí mřížkového rozvržení jednotlivých oken zajištěn dynamický tvar rozhraní. Díky dynamickému tvaru rozhraní je zajištěno optimální rozvržení celého rozhraní na různých rozlišeních počítačových obrazovek (viz obrázek 5.1).



Obr. 5.1: Realizované rozhraní pro měření a kalibraci senzorů

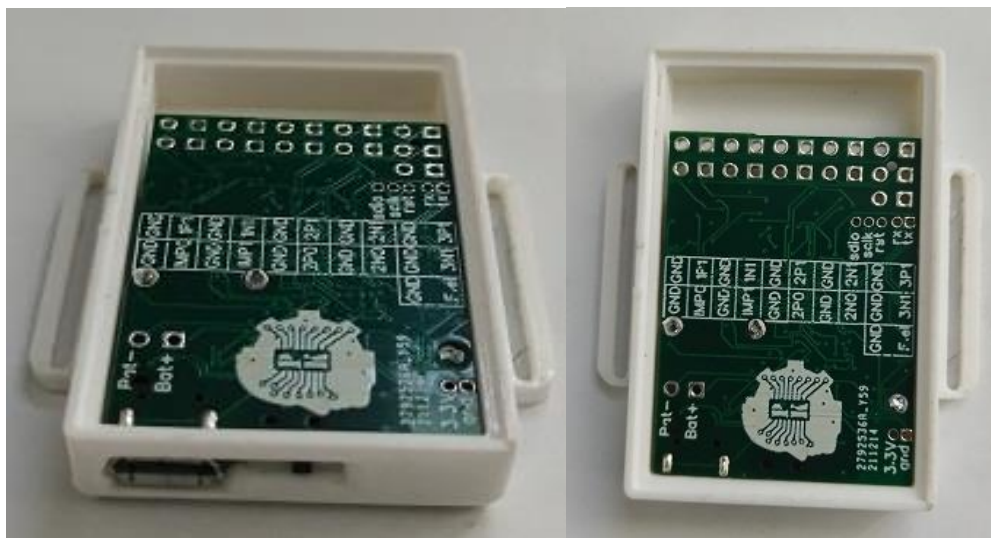
### 5.1.1 Realizované umístění senzoru

V této práci byl program testován na senzoru UMP-6050 (viz obrázek 5.2), který obsahuje jak akcelerometr, tak gyroskop.



Obr. 5.2: Použitý senzor UMP-6050

Pro tento typ akcelerometru bylo navrženo unikátní pouzdro (viz kapitola 4.1), ve které je tento typ senzoru přesně a pevně umístěn (viz obrázek 5.3). Vložení senzoru do pouzdra je díky navrženému designu velmi jednoduché. Po umístění senzoru a baterky je senzor v pouzdru uzavřen horním dílem. Pouzdro lze „zapečetit“ aplikací lepidla do připravených drážek po obvodu pouzdra.

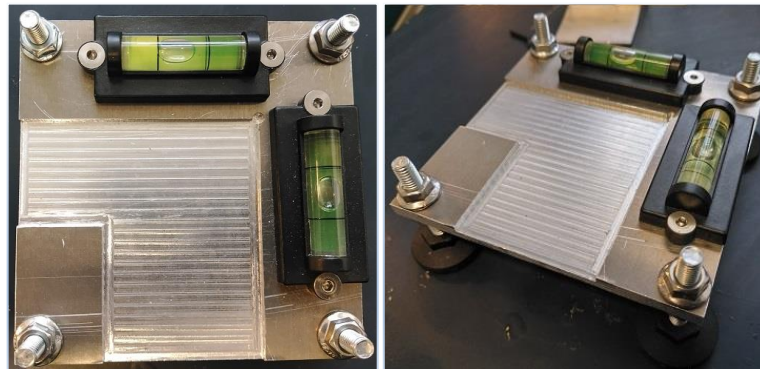


Obr. 5.3: Realizované umístění senzoru v pouzdru



### 5.1.2 Realizovaný stojan pro kalibraci

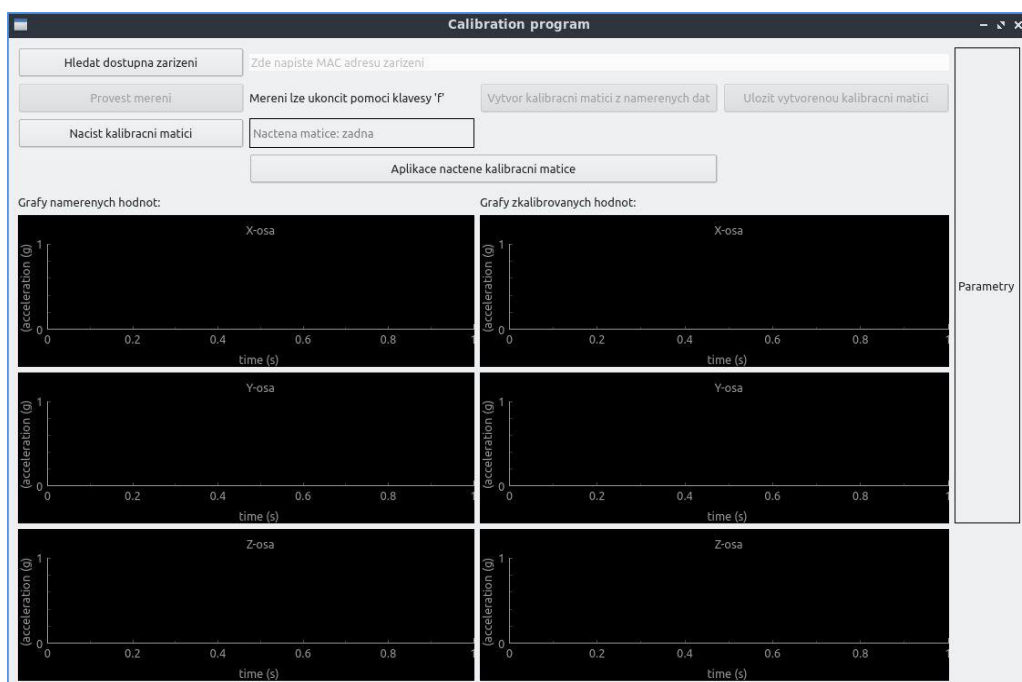
Pro provedení navržené kalibrační metody je další důležitou částí kalibrační stůl, na který je umístěno pouzdro se senzorem. Kalibrační stůl byl realizován podle návrhu z kapitoly 4.4 (viz obrázek 5.4). Kalibrační stůl je navržen tak aby jej bylo možné seřídit do vodorovné polohy pomocí nastavitelných nožiček. Pro sledování a seřizování korektní vodorovné polohy jsou na stolku připevněny vodováhy. Pro zvýšení přesnosti seřízení lze použít digitální úhломěr.



Obr. 5.4: Realizovaný kalibrační stolek

### 5.1.3 Realizace uživatelského rozhraní

Pro vytvoření uživatelského rozhraní je vytvořena funkce createGUI(), jejíž hlavními prvky pro fungování jsou QApplication, QWidget a QGridLayout pocházející z knihovny PyQt5. Realizované uživatelské rozhraní kopíruje navržené rozhraní z kapitoly 4.5 včetně ovládacích prvků. (viz obrázek 5.5)



Obr. 5.5: Realizované uživatelské rozhraní

QApplication je třída spravující ovládání a hlavní nastavení uživatelského rozhraní.

QWidget je základající třídou pro všechny objekty vkládané do uživatelského rozhraní.

QGridLayout je třída rozprostírající vytvářené objekty do přesně určené mřížky (příklad na obrázku 5.6). Objekt je do mřížky vložen pomocí funkce QGridLayoutu a to addWidget. Funkce addWidget umožňuje vložení jakéhokoliv podporovaného objektu na souřadnice x, y.

```
self.layout.addWidget(self.btnConnectDevice, 0,0)
self.layout.addWidget(self.txbConnectDevice, 0, 1, 1, 3 )
self.layout.addWidget(self.btnMeasureData, 1, 0)
self.layout.addWidget(self.lblNotifyMeas, 1, 1)
self.layout.addWidget(self.btnCreateCalMatrix, 1, 2)
self.layout.addWidget(self.btnSaveCalMatrix, 1, 3)
self.layout.addWidget(self.btnLoadCalMatrix, 2, 0)
self.layout.addWidget(self.lblLoadCalMatrix, 2, 1)
self.layout.addWidget(self.btnApplyCalMatrix, 3, 1, 1, 2)
self.layout.addWidget(self.lblCountParam, 0, 4, 8, 2)
```

Obr. 5.6: Příklad vložení objektů do mřížky

V realizovaném rozhraní jsou použity tlačítka (QPushButton), editovatelné textové pole (QLineEdit), needitovatelné textové pole (QLabel) a grafické okno pro vykreslení grafů (PlotWidget). Všechny jmenované prvky pochází z knihovny PyQt5 vyjma PlotWidgetu, který pochází z knihovny pyqtgraph

Funkce vytvářející uživatelské rozhraní musí být ukončena specifickými řádky (viz obrázek 5.7), poté co kompilátor dojde k této řádce je vykresleno celé okno (viz obrázek 5.1).

```
self.window.setLayout(self.layout)
self.window.show()

sys.exit(self.app.exec_())
```

Obr. 5.7: specifický řádek ukončující vytváření uživatelského rozhraní

### 5.1.4 Realizace implementace komunikace se senzorem

Pro komunikaci se senzorem UMP-6050 byla využita knihovna bluepy. K počítači musí být připojen usb Bluetooth adaptér (viz obrázek 5.8) (pokud daný počítač neobsahuje Bluetooth modul). Knihovna bluepy umožňuje připojení Bluetooth zařízení podle MAC adresy. Bohužel tato konkrétní knihovna pracuje pouze pod operačním systémem Linux a tedy tuto aplikaci nelze spustit na systémech Windows.



Obr. 5.8: senzor UMP-6050 v pouzdru (vlevo) a usb Bluetooth adaptér (vpravo)

Po stisknutí tlačítka vyhledat dostupná zařízení v hlavním okně je spuštěna funkce na vypsání dostupných zařízení (viz obrázek 5.9). K vyhledání dostupných zařízení je použita funkce listDevice() (viz obrázek 5.10)

```
from Facade import Facade
f = Facade()
if (self.firstScan == True):
    #listing of available bluetooth device with name and MAC address
    devices = f.list_device()
    availDevice = "MAC adresa           Nazev zarizeni \n"
    for dev in devices:
        print(dev)
        availDevice += "{} - {}\n".format(dev['addr'],dev['name'])
    self.lblCountParam.setTextInteractionFlags(Qt.TextSelectableByMouse)
    self.lblCountParam.setText(availDevice)
    self.firstScan = False
    return
```

Obr. 5.9: Kód pro vypsání dostupných zařízení

```

@staticmethod
def list_device():
    scanner = Scanner()
    devices = scanner.scan(10.0)
    devices_m = []

    for dev in devices:
        name = ""
        power = ""
        for (adtype, desc, value) in dev.getScanData():
            if (desc == "Complete Local Name"):
                name = str(value)
            elif (desc == "Tx Power"):
                power = str(value)

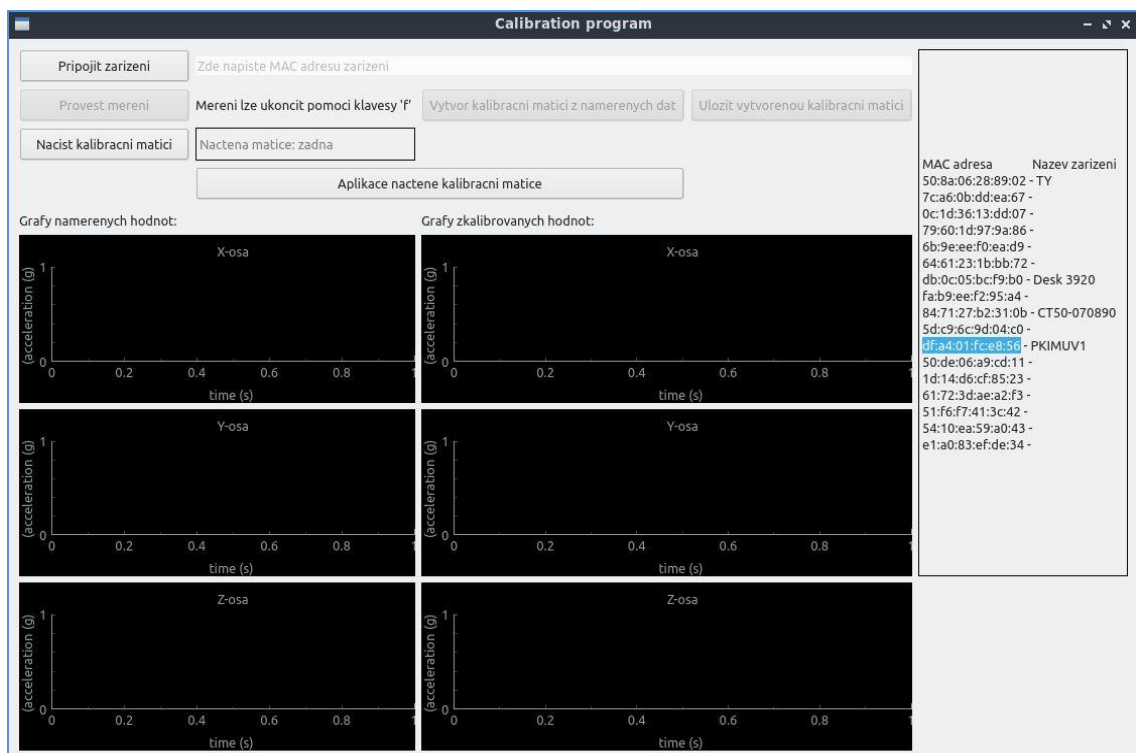
        # add device addr, addType and rssi to devices_m
        devices_m.append({'addr': dev.addr, 'addType': dev.addrType, 'rssi': dev.rssi, 'name': name, 'power': power})

    # json_devices = json.dumps(devices_m)
    return devices_m

```

Obr. 5.10: Vnitřní funkce pro nalezení dostupných zařízení

Nalezení dostupných zařízení je docíleno pomocí funkce `list_device()`, která využívá funkce `Scanner()` z knihovny `bluepy`. Funkce `Scanner()` je napojena na Bluetooth adaptér připojený k počítači a provede skenování dostupných zařízení. Nalezená zařízení jsou následně pomocí `for` cyklu vybrány a z každého nalezeného zařízení převzaty podstatné informace (MAC adresa a název zařízení). Zjištěné MAC adresy a názvy dostupných zařízení jsou posléze vypsány v parametřovém okně (viz. obrázek 5.11)



Obr. 5.11: Uživatelské rozhraní po nalezení dostupných Bluetooth zařízení

Z parametrového okna je možné požadovanou MAC adresu zkopírovat a následně vložit do textového pole pro MAC adresu. Vložení MAC adresy do textového pole a kliknutím na připojit zařízení je spuštěna funkce na připojení zařízení do programu (viz obrázek 5.12). Pokud je zařízení úspěšně připojeno je do textového pole vyplněn text „Připojene zarizeni: “ a MAC adresa zařízení, zároveň je dané textové pole zešedivěno a deaktivováno. V případě neúspěšného připojení je nutné zkontrolovat a opravit zadanou MAC adresu a pokus o připojení opakovat.

```
if (self.firstScan == False):
    #connect to specific bluetooth device via MAC address
    macAddress = self.txbConnectDevice.text()
    dev = f.find_device(macAddress)
    #dev = f.find_device('df:a4:01:fc:e8:56')
    self.txbConnectDevice.setText('Připojene zarizeni: ' + macAddress)
    self.txbConnectDevice.setEnabled(False)
    self.btnConnectDevice.setEnabled(False)
    self.btnMeasureData.setEnabled(True)

t = TestObj()
f.connect_from_to('channel0',dev,t)
```

Obr. 5.12: Funkce pro připojení zařízení

Při úspěšném připojení zařízení je zařízení uloženo jako proměnná dev. Následně jsou pomocí třídy TestObj ukládány měřená data zařízením do proměnné data (viz obrázek 5.13).

```
class TestObj:

    def __init__(self) -> None:
        pass

    def set_input(self, pub, channel):
        pub.subscribe(self.test_obj_input, channel)

    def test_obj_input(self, arg):
        guilogics.data = arg
```

Obr. 5.13: Kód pro ukládání naměřených dat pomocí funkce test\_obj\_input()

Poté co jsou data ukládány do proměnné data, lze začít měření pomocí funkce `dev.start()` a `dev.wait_for_notification()` (viz obrázek 5.14). Od chvíle co je zavolána tato funkce jsou do proměnné data posílány naměřená data zařízením.

```
#start of measuring from device
dev.start()
dev.wait_for_notification(1)
#save of first measured data
self.startTime = self.data['time_ms'][0]
timeAct = timeAct + self.data['time_ms']
accX = accX + self.data['acceleration_X_g']
accY = accY + self.data['acceleration_Y_g']
accZ = accZ + self.data['acceleration_Z_g']
timeAct = self.unixTimeToSecond(self.startTime, timeAct)
#plot of measured data
myPlotX.setData(timeAct, accX)
myPlotY.setData(timeAct, accY)
myPlotZ.setData(timeAct, accZ)
```

Obr. 5.14: Kód pro start měření

Po spuštění začátku měření pomocí funkce `dev.start` a `dev.wait_for_notification()` jsou uloženy naměřená data pro jednotlivé osy do zvláštních proměnných. Na daném senzoru je měřen čas v unixovém čase, tedy čas v mikrosekundách, který uplynul od okamžiku koordinovaného světového času (půlnoc 1. ledna 1970). Pro reprezentaci dat je tento čas nevhodný a tak je pro převod z unixového času na vteřiny od počátku měření navržena funkce `unixTimeToSecond()` (viz obrázek 5.15).

```
def unixTimeToSecond(self, startTime, unixTime):
    for t in unixTime:
        self.time.append(((t - startTime) + 1)/1000)
        # print(self.time)
    return self.time
```

Obr. 5.15: Funkce pro převod unixového času na vteřiny od počátku měření

Na konci funkce pro připojení zařízení je provedeno vykreslení bloku dat do grafů v rozhraní pomocí funkce `setData()` (viz obrázek 5.16)



Obr. 5.16: Rozhraní po úspěšném připojení zařízení a vykreslení prvního bloku dat

### 5.1.5 Realizace korekce polohy a určení relativních úhlů senzoru

Korekce polohy připojeného zařízení je vypočtena z naměřených dat. Na senzor, který je v klidové poloze působí pouze gravitační zrychlení, a tedy při jedné z definovaných poloh působí pouze na jednu ze tří os. Pokud je na ose kolmo k zemi naměřené gravitační zrychlení a na zbylých osách téměř nulové, je potom senzor umístěn v korektní poloze. Této skutečnosti využívá funkce `calculateCorrectionPosition()` (viz obrázek 5.17).

```
import numpy as np

def calculateCorrectionPosition(dataX, dataY, dataZ):
    if(np.mean(dataX)<=1):
        axisXVert = 'Osa X není vhodná pro kalibraci\n' + 'Úhel osy x vůči zemi je: ' + str(f'{np.rad2deg(np.arccos(np.mean(dataX))):.3f}')
    if(np.mean(dataY)<=1):
        axisYVert = 'Osa Y není vhodná pro kalibraci\n' + 'Úhel osy x vůči zemi je: ' + str(f'{np.rad2deg(np.arccos(np.mean(dataY))):.3f}')
    if(np.mean(dataZ)<=1):
        axisZVert = 'Osa Z není vhodná pro kalibraci\n' + 'Úhel osy x vůči zemi je: ' + str(f'{np.rad2deg(np.arccos(np.mean(dataZ))):.3f}')

    axisVert = []
    if(np.mean(dataX) > 0.9 and np.mean(dataX) < 1.1 and np.mean(dataY) < 0.1 and np.mean(dataZ) < 0.1):
        axisXVert = 'Poloha senzoru je vhodná pro kalibraci osy X'
    if(np.mean(dataX) < 0.1 and np.mean(dataY) > 0.9 and np.mean(dataY) < 1.1 and np.mean(dataZ) < 0.1):
        axisYVert = 'Poloha senzoru je vhodná pro kalibraci osy Y'
    if(np.mean(dataX) < 0.1 and np.mean(dataY) < 0.1 and np.mean(dataZ) > 0.9 and np.mean(dataZ) < 1.1):
        axisZVert = 'Poloha senzoru je vhodná pro kalibraci osy Z'
    axisVert = [axisXVert, axisYVert, axisZVert]
    return axisVert
```

Obr. 5.17: Funkce pro určení korekce polohy

Relativní úhly jednotlivých os senzoru jsou počítány z naměřených dat. Naměřená data jsou ukládána do vektoru hodnot pro jednotlivé osy. Pro výpočet úhlu mezi vektory, kteří mezi sebou svírají nenulový úhle, je využita funkce `arccos()` z knihovny `numpy`. Před výpočtem relativního úhlu mezi vektory je nutné provést matematickou operaci bodového součinu dvou vektorů pomocí funkce `dot()` z knihovny `numpy`. Po provedení bodového součinu je vypočten `arccos`, který vrací výsledek v radiánech. Radiány lze lehce převést na stupně pomocí funkce `rad2deg()` (viz obrázek 5.18).

```
if(np.abs(np.mean(dataX)) > 0.9 and np.abs(np.mean(dataX))<1.1):  
    angleXY = np.rad2deg(np.arccos(np.dot(np.mean(dataX), np.mean(dataY))))  
    angleXZ = np.rad2deg(np.arccos(np.dot(np.mean(dataX), np.mean(dataZ))))  
    calParams.append('uhel mezi osou X a Y: ')  
    calParams.append(angleXY)  
    calParams.append('uhel mezi osou X a Z: ')  
    calParams.append(angleXZ)
```

Obr. 5.18: Příklad kódu pro výpočet relativních úhlu mezi osou x, y a x, z

### 5.1.6 Realizace kalibrační metody

Realizovaná kalibrační metoda pracuje se skutečností, že na senzor v klidovém stavu působí pouze gravitační zrychlení. Před zahájením kalibrační procedury je důležité správně nastavit kalibrační stolek pomocí nastavitelných nožiček. Při nastavování pomocí připevněných vodovah na kalibračním stolku je důležité, aby byla bublinka na vodováze co nejbližší ke středu (viz obrázek 5.19 vlevo). Při nastavování stolku pomocí digitálního úhloměru je maximální povolená chyba ve vodorovné ose  $\pm 0,02^\circ$  (viz obrázek 5.19 uprostřed) a ve svislé ose minimálně  $89,8^\circ$  a maximálně  $90,2^\circ$  (viz obrázek 5.19 vpravo).



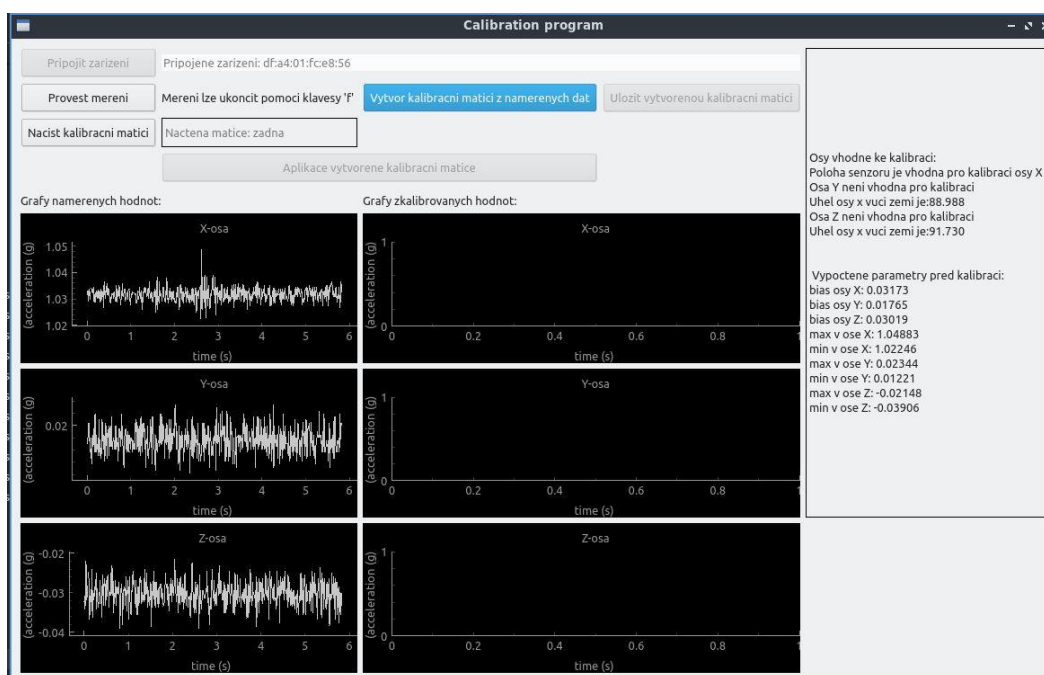
Obr. 5.19: Korektní vodorovná poloha podle vodováh (vlevo), korektní vodorovná poloha podle digitálního úhloměru (uprostřed) a korektní svislá poloha podle digitálního úhloměru



Na korektně nastavený kalibrační stůl, je umístěno pouzdro se senzorem v první definované poloze, kdy je osa  $x$  kolmo k zemi. Pokud je pouzdro opatřeno křídélky pro upevnění pásu, lze ji umístit na okraj stolu (viz obrázek 5.20) V aplikaci lze na konci každého měřicího bloku zkontrolovat, jestli je senzor správně umístěn. Kontrola polohy spočívá v kontrole naměřeného bloku, zda se v něm nevyskytují abnormální hodnoty a v kontrole parametrového okna, jestli je v dané ose senzor správně umístěn. Po kontrole polohy je provedeno alespoň 5 vteřin měření (viz obrázek 5.21) a následné vytvoření kalibrační matice pro osu  $x$  pomocí tlačítka pro vytvoření kalibrační matice.



Obr. 5.20: Korektní umístění senzoru pro osu  $x$  v kolmém směru na zem



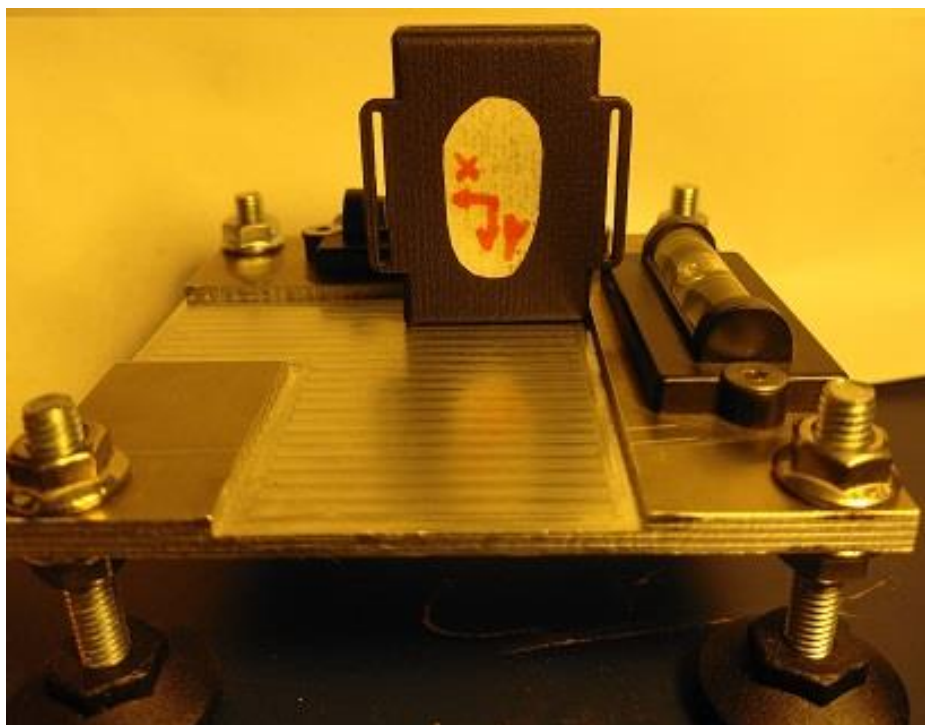
Obr. 5.21: Kontrola správného umístění senzoru a vytvoření kalibrační matice

Kalibrační matice pro první definovanou polohu je vytvořena pomocí funkce `gramSchmidt()`, která využívá knihovnu `numpy` k výpočtu kalibrační matice. Celá tato logika je volaná funkcí `createMatrix()`, zpracovávající blok aktuálně naměřených dat (viz obrázek 5.22).

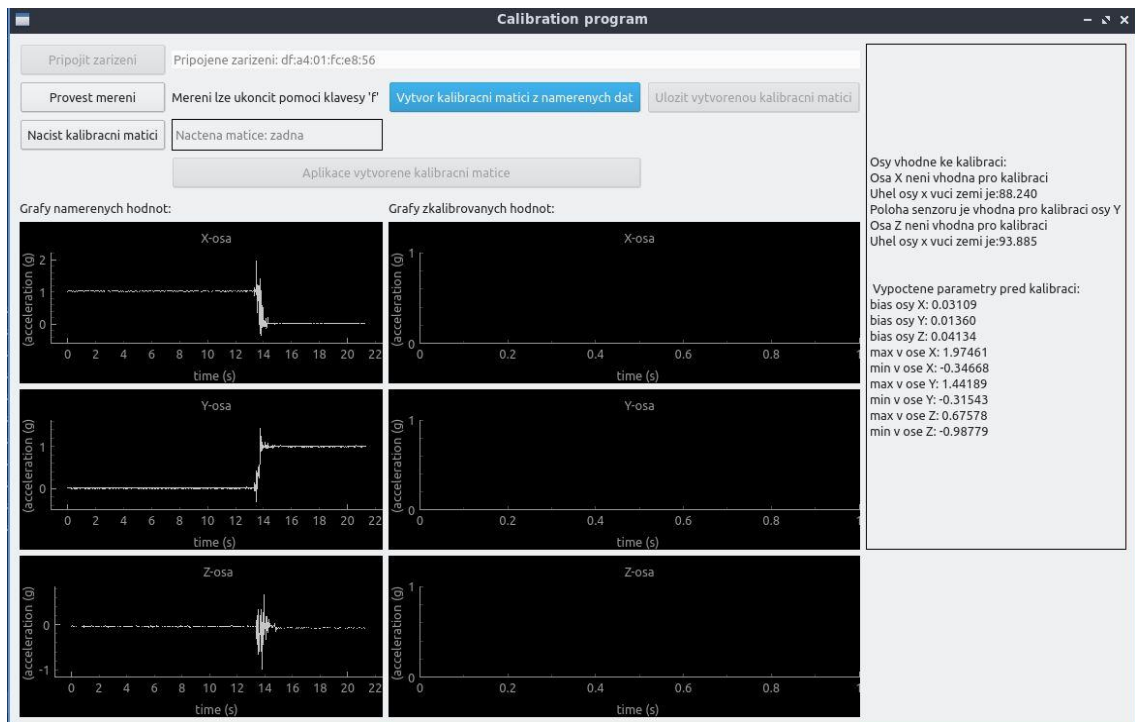
```
#creating of calib matrix for axis X via gram-smith ortghogonal proces
dataForCalib = [self.accX,self.accY,self.accZ]
self.dataCalibMatrix.append(gramSchmidt(dataForCalib))
self.calibMatrix.append(np.abs(self.dataCalibMatrix[0][0]))
```

Obr. 5.22: Příklad kódu pro výpočet kalibrační matice pomocí Gram-Schmidtovy metody

Po provedení měření v první definované poloze je pouzdro se senzorem umístěno v druhé definované poloze, kdy je osa  $y$  kolmo k zemi (viz obrázek 5.23). V aplikaci lze na konci každého měřicího bloku zkontrolovat, jestli je senzor správně umístěn. Kontrola polohy spočívá v kontrole naměřeného bloku, zda se v něm nevyskytují abnormální hodnoty a v kontrole parametrového okna, jestli je v dané ose senzor správně umístěn. Po kontrole polohy je provedeno alespoň 5 vteřin měření (viz obrázek 5.24) a následné vytvoření kalibrační matice pro osu  $y$  pomocí tlačítka pro vytvoření kalibrační matice.



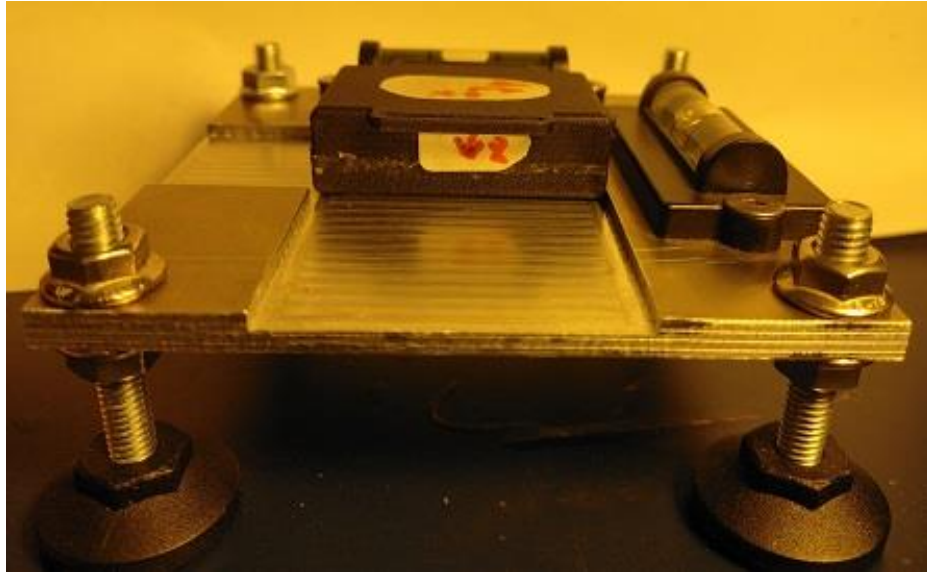
Obr. 5.23: Korektní umístění senzoru pro osu  $y$  v kolmém směru na zem



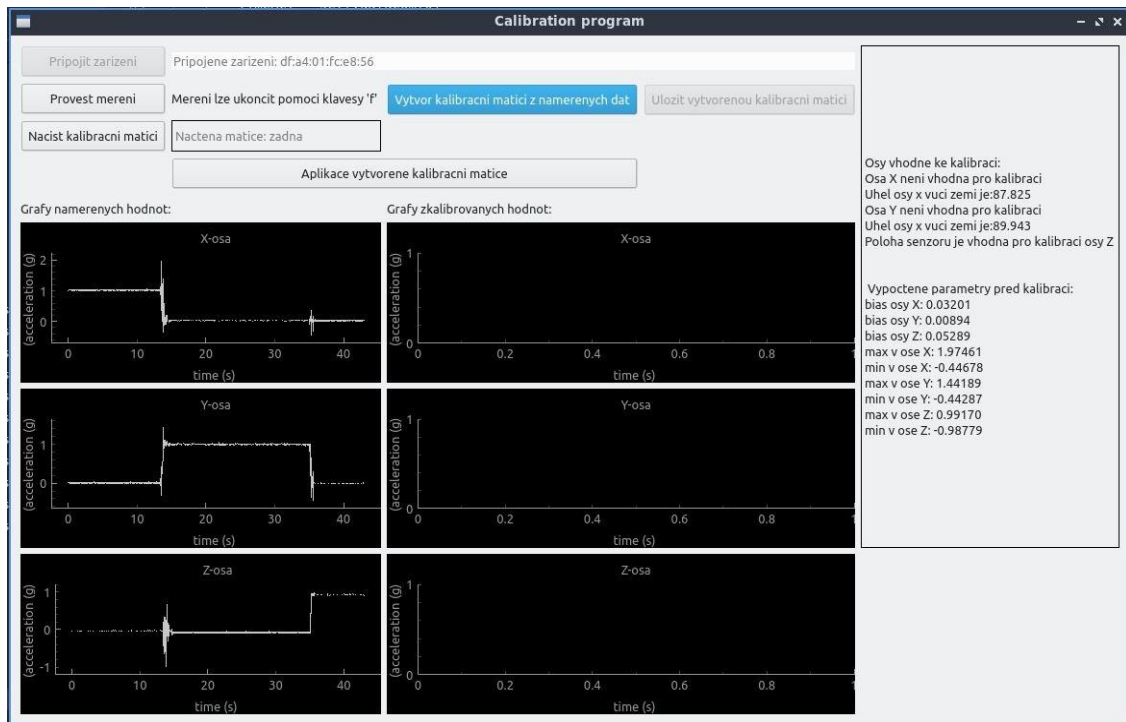
Obr. 5.24: Kontrola správného umístění senzoru a vytvoření kalibrační matice

Kalibrační matice pro druhou definovanou polohu je vytvořena stejně jako pro první definovanou polohu pomocí funkce `gramSchmidt()`, která využívá knihovnu `numpy` k výpočtu kalibrační matice. Celá tato logika je volaná funkcí `createMatrix()`, zpracovávající blok aktuálně naměřených dat.

Po provedení měření v druhé definované poloze je pouzdro se senzorem umístěno ve třetí definované poloze, kdy je osa z kolmo k zemi (viz obrázek 5.25). V aplikaci lze na konci každého měřicího bloku zkontrolovat, jestli je senzor správně umístěn. Kontrola polohy spočívá v kontrole naměřeného bloku, zda se v něm nevyskytují abnormální hodnoty a v kontrole parametřového okna, jestli je v dané ose senzor správně umístěn. Po kontrole polohy je provedeno alespoň 5 vteřin měření (viz obrázek 5.26) a následně vytvoření kalibrační matice pro osu z pomocí tlačítka pro vytvoření kalibrační matice.

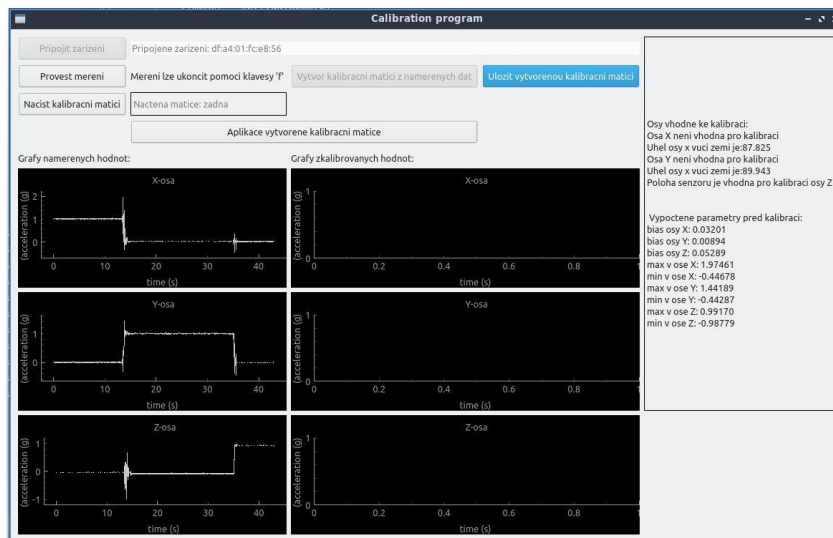


Obr. 5.25: Korektní umístění senzoru pro osu  $z$  v kolmém směru k zemi



Obr. 5.26: Kontrola správného umístění senzoru a vytvoření kalibrační matice

Po dokončení měření v třetí definované poloze a vytvoření kalibrační matice pro osu  $z$  je možné danou kalibrační matici uložit do formátu .csv (viz obrázek 5.27)



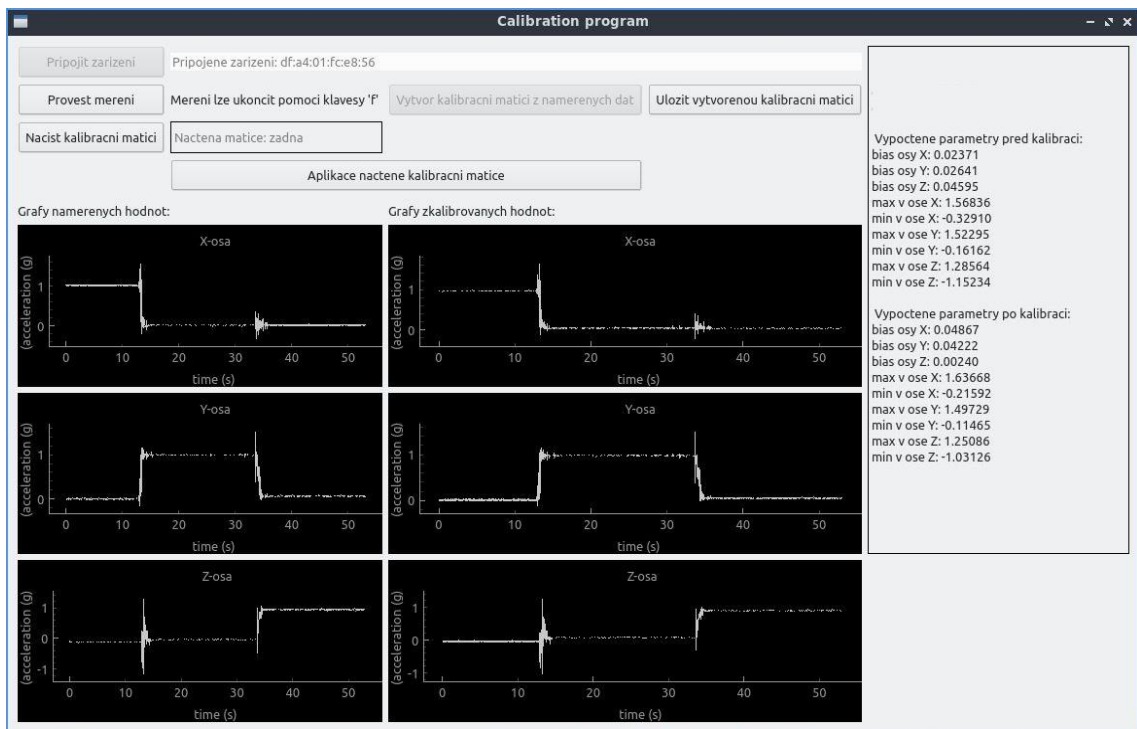
Obr. 5.27: Okno po vytvoření kalibrační matice s výrazněnou možností uložení matice

Kalibrační matice by měla vypadat podobně, jako v kapitole 4.4.1 vzorec (8) (viz obrázek 5.28)

```
CalibMatrix:
[0.99713827 0.01238555 0.07457788]
[0.02925306 0.99918719 0.02773467]
[0.02058412 0.02250579 0.99953478]
```

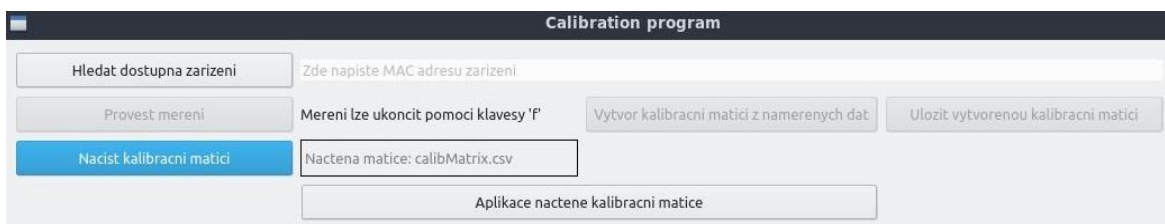
Obr. 5.28: Příklad výsledné kalibrační matice

Další možností po dokončení měření je možnost aplikace kalibrační matice. Po stisknutí tlačítka aplikace kalibrační matice jsou veškerá doposud naměřená data zkalibrována kalibrační maticí a zobrazeny do pravých grafů. Kromě vykreslení jednotlivých grafů je proveden také výpočet všech sledovaných parametrů a zobrazen v parametřovém okně jakožto parametry po kalibraci (viz obrázek 5.29)



Obr. 5.29: Rozhraní po aplikaci vytvořené kalibrační matice

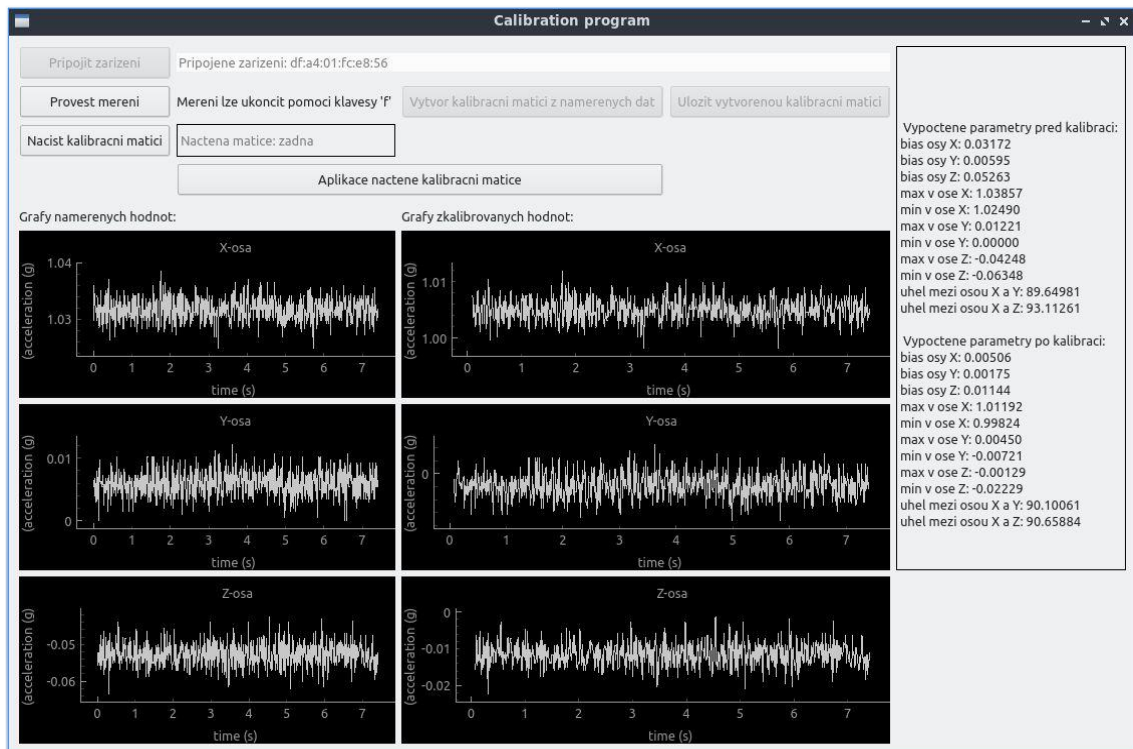
Pokud byla kalibrační matice po vytvoření uložena do počítače, tak při dalším novém měření se již nemusí provádět kalibrační procedura. Stačí pouze pomocí tlačítka načíst kalibrační matici, zda se matice správně načetla lze ověřit pomocí textového pole vedle tlačítka pro načítání ve kterém je zobrazeno jaká matice se načetla (viz obrázek 5.30). Po načtení matice, lze naměřit data a aplikovat kalibrační matici.



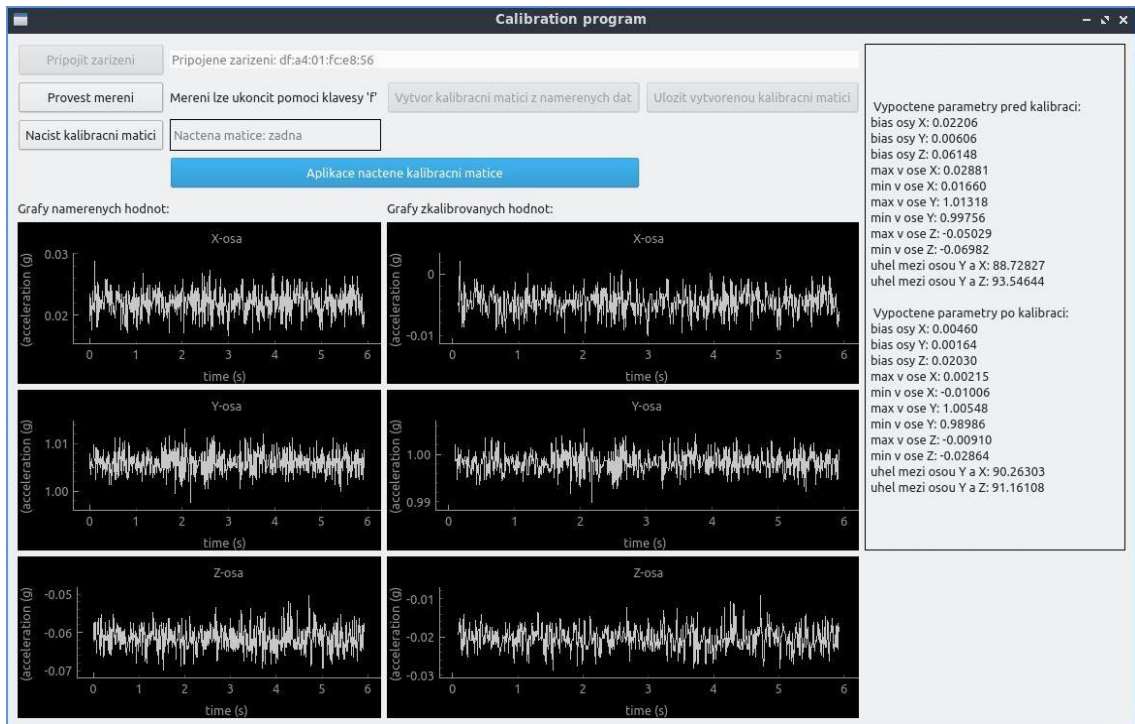
Obr. 5.30: Výstřižek rozhraní po načtení kalibrační matice

## 5.2 Testování a zhodnocení navrženého nástroje

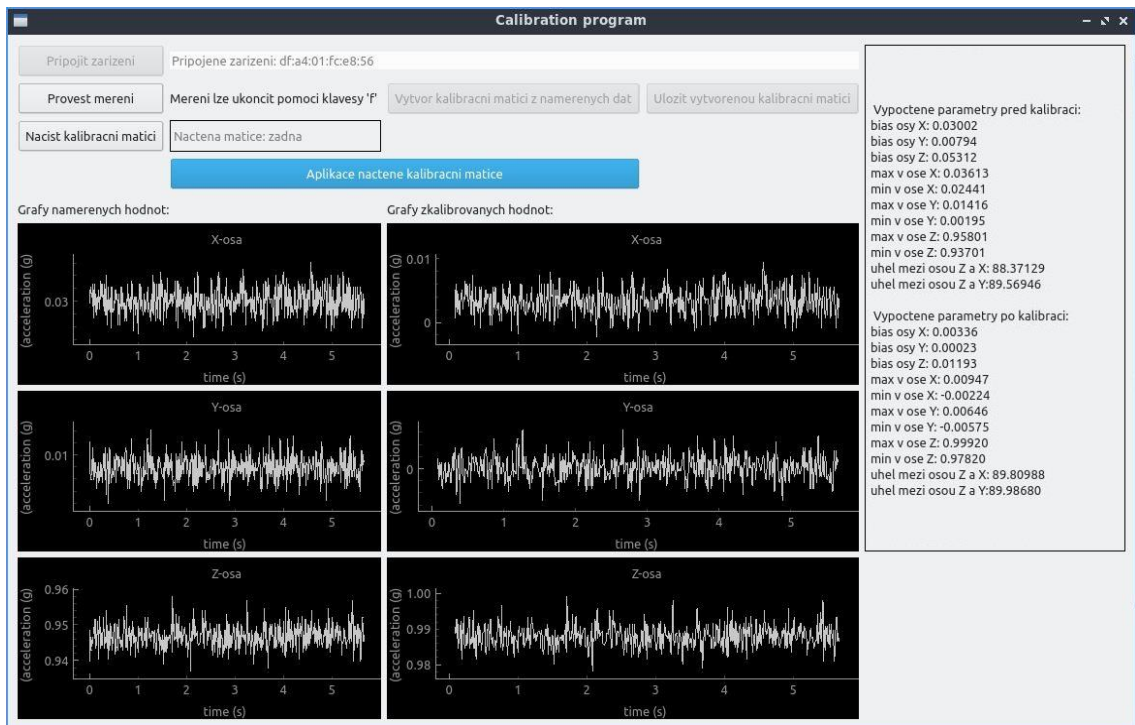
Po vytvoření kalibrační matice byly znovu naměřeny data v ose  $x$  kolmo k zemi, v ose  $y$  kolmo k zemi a v ose  $z$  kolmo k zemi a na tyto data byla aplikována vytvořená kalibrační matice. Výsledky z testovacího měření jsou na obrázku 5.31 pro osu  $x$ , na obrázku 5.32 pro osu  $y$ , na obrázku 5.33 pro osu  $z$  a také jsou výsledky shrnuty v tabulce 5.1.



Obr. 5.31: Naměřené, kalibrované a vypočtené parametry pro osu  $x$



Obr. 5.32: Naměřené, kalibrované a vypočtené parametry pro osu y



Obr. 5.33: Naměřené, kalibrované a vypočtené parametry pro osu z



Tab. 5.1: Shrnutí vypočtených hodnot testovacího měření

vypočtené hodnoty před aplikací kalibrační matice					
osa $x$ kolmo na zem		osa $y$ kolmo na zem		osa $z$ kolmo na zem	
bias osy $x$	0,03172	bias osy $x$	0,02206	bias osy $x$	0,03002
bias osy $y$	0,00595	bias osy $y$	0,00606	bias osy $y$	0,00794
bias osy $z$	0,05263	bias osy $z$	0,06148	bias osy $z$	0,05312
max v ose $x$	1,03857	max v ose $x$	0,02881	max v ose $x$	0,03613
průměr v ose $x$	1,02945	průměr v ose $x$	0,02115	průměr v ose $x$	0,02842
min v ose $x$	1,02490	min v ose $x$	0,01660	min v ose $x$	0,02441
max v ose $y$	0,01221	max v ose $y$	1,01318	max v ose $y$	0,01416
průměr v ose $y$	0,00530	průměr v ose $y$	1,00452	průměr v ose $y$	0,00856
min v ose $y$	0,0000	min v ose $y$	0,99756	min v ose $y$	0,00195
max v ose $z$	-0,04248	max v ose $z$	-0,05029	max v ose $z$	0,95801
průměr v ose $z$	-0,05596	průměr v ose $z$	-0,06102	průměr v ose $z$	0,95201
min v ose $z$	-0,06348	min v ose $z$	-0,06982	min v ose $z$	0,93701
uhel mezi osou $x$ a $y$	89,64981	uhel mezi osou $y$ a $x$	88,72827	uhel mezi osou $z$ a $x$	88,37129
uhel mezi osou $x$ a $z$	93,11261	uhel mezi osou $y$ a $z$	93,54644	uhel mezi osou $z$ a $y$	89,56946

Vypočtené hodnoty po aplikaci kalibrační matice

osa $x$ kolmo na zem		osa $y$ kolmo na zem		osa $z$ kolmo na zem	
bias osy $x$	0,00506	bias osy $x$	0,00460	bias osy $x$	0,00336
bias osy $y$	0,00175	bias osy $y$	0,00164	bias osy $y$	0,00023
bias osy $z$	0,01144	bias osy $z$	0,02030	bias osy $z$	0,01193
max v ose $x$	1,01192	max v ose $x$	0,00215	max v ose $x$	0,00947
průměr v ose $x$	1,00546	průměr v ose $x$	0,00125	průměr v ose $x$	0,00654
min v ose $x$	0,99824	min v ose $x$	-0,01006	min v ose $x$	-0,00224
max v ose $y$	0,00450	max v ose $y$	1,00548	max v ose $y$	0,00646
průměr v ose $y$	-0,00531	průměr v ose $y$	0,99856	průměr v ose $y$	0,00254
min v ose $y$	-0,00721	min v ose $y$	0,98986	min v ose $y$	-0,00575
max v ose $z$	-0,00129	max v ose $z$	-0,00910	max v ose $z$	0,99920
průměr v ose $z$	-0,01358	průměr v ose $z$	-0,01598	průměr v ose $z$	0,98502
min v ose $z$	-0,02229	min v ose $z$	-0,02864	min v ose $z$	0,97820
uhel mezi osou $x$ a $y$	90,10061	uhel mezi osou $y$ a $x$	90,26303	uhel mezi osou $z$ a $x$	89,80988
uhel mezi osou $x$ a $z$	90,65884	uhel mezi osou $y$ a $z$	91,16108	uhel mezi osou $z$ a $y$	89,98680

Z naměřených výsledků je možné vypočítat o kolik se měření akcelerometrem po kalibraci zlepšilo. V tomto případě je zapotřebí zjistit o kolik se změnila bias chyba a o kolik se změnila chyba ortogonalit (tedy úhel mezi osami).

Vypočtené rozdíly naměřených a kalibrovaných hodnot jsou v tabulce 5.2

Tab. 5.2: Vypočtené rozdíly chyby bias a chyby ortogonality naměřených a kalibrovaných hodnot

natočení senzoru	osa	vypočtený bias před kalibrací	vypočtený bias po kalibraci	rozdíl chyby před a po kalibraci
osa $x$ kolmo k zemi	$x$	0,03172	0,00506	0,02666
	$y$	0,00595	0,00175	0,00420
	$z$	0,05263	0,01144	0,04119
osa $y$ kolmo k zemi	$x$	0,02206	0,00460	0,01746
	$y$	0,00606	0,00164	0,00442
	$z$	0,06148	0,02030	0,04118
osa $z$ kolmo k zemi	$x$	0,03002	0,00336	0,02666
	$y$	0,00794	0,00023	0,00771
	$z$	0,05312	0,01193	0,04119
		Průměrný rozdíl chyby bias před a po kalibraci	$x$	0,02359
			$y$	0,00544
			$z$	0,04119
natočení senzoru	osy	vypočtený úhel mezi osami před kalibrací	vypočtený úhel mezi osami po kalibraci	rozdíl chyby před a po kalibraci
osa $x$ kolmo k zemi	$xy$	89,65°	90,10°	0,45°
	$xz$	93,11°	90,65°	2,46°
osa $y$ kolmo k zemi	$yx$	88,73°	90,26°	1,92°
	$yz$	93,55°	91,16°	2,39°
osa $z$ kolmo k zemi	$zx$	88,37°	89,81°	1,44°
	$zy$	89,81°	89,98°	0,17°
		Průměrný rozdíl chyby ortogonality před a po kalibraci	$xy$	0,50°
			$yz$	1,28°
			$zx$	1,95°

Z tabulky vypočtených rozdílů lze vidět, že po aplikaci kalibrační matice byla vypočtená chyba biasu v osách  $x$  a  $z$  výrazně snížena. Dále je v tabulce viditelný posun úhlů mezi osami nejvíce o 2,46°, také je zde vidět že se úhel mezi osami blíží k 90° čímž je snížena chyba ortogonality.

Pro správné statistické zpracování je důležité provedení opakované měření za stejných podmínek na zařízení. Takto naměřená data se mohou od sebe odlišovat např. rozlišovací schopností, náhodnými vlivy apod. V této práci je statistické zhodnocení provedeno na základě naměřených dat

Statistické zpracování lze zpracovat pomocí nejistoty typu A. Nejistotu typu A lze stanovit výpočtem (12) z opakovaně naměřených hodnot.

$$u_A = \sqrt{\frac{1}{n(n-1)} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (10)$$

Kde:

$n$  – celkový počet měření

$x_i$  – naměřená hodnota

$\bar{x}$  – aritmetický průměr

Zhodnocení naměřených dat je provedeno pomocí výpočtu nejistoty typu A a rozšířené nejistoty typu A. Zjištěné výsledky jsou shrnuty v tabulce 9.1.

Tab. 9.1: Vyhodnocené nejistoty typu A naměřených hodnot

Natočení senzoru	Počet naměřených hodnot	Měřená osa	Průměr naměřených hodnot	Průměr kalibrovaných hodnot	Nejistota typu A	Rozšířená nejistota	
			$\bar{x}_n$	$\bar{x}_c$		$2u_A$	$3u_A$
Osa x kolmo k zemi	421	x	1,02945	1,00546	0,0570	0,114	0,1710
		y	0,00530	-0,00131	0,0103	0,0206	0,0309
		z	-0,05596	-0,01358	0,0206	0,0412	0,0618
Osa y kolmo k zemi	386	x	0,02115	0,00125	0,0941	0,1882	0,2823
		y	1,00452	0,99856	0,0290	0,0580	0,0870
		z	-0,06102	-0,01598	0,0219	0,0438	0,0657
Osa z kolmo k zemi	369	x	0,02842	0,00654	0,0114	0,0228	0,0342
		y	0,00856	0,00254	0,0313	0,0626	0,0939
		z	0,95201	0,98502	0,0172	0,0344	0,0516
Průměrná nejistota typu A					0,033	0,0651	0,0976

## 6 Diskuse

Cílem této práce bylo navržení vhodné kalibrační metody, která bude minimalizovat nejčastější chyby inerciálních senzorů. Tato práce je zaměřena na provedení kalibrace inerciálního senzoru akcelerometru. V této práci jsou shrnuty metody kalibrace akcelerometru a navržena metoda pro kalibraci vycházející ze skutečnosti působení gravitačního zrychlení na senzor. Pro minimalizaci chyby vychýlených měřících os a nelinearity byla použita kalibrační matice vytvořená pomocí Gram-Schmidtova ortogonalizačního procesu, protože byla nejvhodnější z nalezených metod.

Jako programovací jazyk byl zvolen Python, který oproti Matlabu má lepší programovací prostředky pro vytvoření uživatelského rozhraní, snadnější implementaci algoritmu pro připojení zařízení a snadnější implementaci kalibračního procesu.

Pro implementaci kalibračního programu byly použity užitečné knihovny jazyka Python. Jednou z důležitých použitých knihoven je PyQt5, která slouží k vytvoření uživatelského rozhraní. Navržené uživatelské rozhraní je navrženo tak aby byl program uživatelsky přívětivý a snadný na obsluhu.

Pro provedení kalibrace byla zvolena metoda pracující s gravitačním zrychlením, a to z důvodu, že tato metoda patří mezi nejvhodnější metody pro akcelerometry. Pro výpočet kalibrační matice byla zvolena metoda Gram-Schmidtova ortogonalizačního procesu, a to z důvodu snadnější implementace na navrženou kalibrační metodu. Jiné způsoby výpočtu kalibrační matice byly zamítnuty především z důvodu nekomptability s navrženou metodou kalibrace.

Pro kalibrační metodu bylo navrženo specifické pouzdro pro umístění senzoru a unikátní nastavitelný kalibrační stolek. Aby byla kalibrační metoda správně provedena a bylo dosaženo co možná nejlepších výsledků je sepsán uživatelský návod. V uživatelském návodě je krok po kroku napsáno, jaké úkony se musejí provést pro správné provedení kalibrační metody.

Při provádění měření je uživatel informován o stavu měření a o poloze zařízení pomocí parametrevého okna v reálném čase, což umožňuje uživateli flexibilně kalibrovat polohu senzoru v čase.

Naměřená a kalibrovaná data jsou zpracovávána pomocí navrženého programu. Z naměřených dat jsou vypočítány podstatné parametry před a po kalibraci, které lze porovnat v parametrevém okně programu. V případě bias chyby dochází ke snížení bias chyby pro osu  $x$  o  $0,02359\text{ g}$ , pro osu  $y$  dochází ke snížení bias chyby o  $0,00544\text{ g}$  a pro osu  $z$  dochází ke snížení bias chyby o  $0,04119$ . Dále je pomocí kalibrační matice minimalizována chyba způsobená chybou vychýlení měřících os a zde dochází ke snížení

chyby ortogonality mezi osou  $x$  a  $y$  o  $0,50^\circ$ , mezi osou  $y$  a  $z$  o  $1,28^\circ$  a mezi osou  $z$  a  $x$  o  $1,95^\circ$ .

Dle vypočtených parametrů lze konstatovat, že došlo k minimalizaci chyby vychýlení měřících os. K největšímu narovnání kolmosti os došlo při měření osou  $x$  kolmo k zemi, a to mezi osou  $x$  a  $z$  a to z  $93,11^\circ$  na  $90,65^\circ$  tedy provedená korekce je o  $2,46^\circ$ . U této navržené metody lze říci, že čím blíže se přiblížíme k  $90^\circ$  tím optimálnější výsledek kalibrace bude. U těchto bodů je možné navržený program optimalizovat, použitím jiné metody pro výpočet kalibrační matice, nebo použitím jiného způsobu výpočtu bias chyby.

Komunikace programu s Bluetooth zařízením je postavena na knihovně bluepy, která je dostupná pouze na operační systém Linux. Celkové řešení programu je do budoucna rozšířitelné o zpřístupnění programu na další platformy (např. operační systém Windows). Dalším možným rozšířením programu je přidání kalibračních algoritmů pro gyroskop.

## 7 Závěr

Hlavním cílem práce bylo navržení kalibrační metody pro inerciální senzory, v tomto případě akcelerometr. Kalibrační metoda je navržena a implementována v jazyce Python.

Pro používaný inerciální senzor akcelerometr UMP-6050 bylo navrženo specifické pouzdro zajišťující pevnou a rovnoměrnou pozici senzoru v pouzdru. Navržené pouzdro se senzorem je následně pokládáno na navržený unikátní kalibrační stůl.

Navržená kalibrační metoda vychází ze skutečnosti, že na akcelerometr v klidové poloze působí pouze gravitační zrychlení. Akcelerometr je pokládán v předem definovaných polohách (jednotlivými osami kolmo vůči zemi) na kalibrační stůl a postupnými kroky je vytvořena kalibrační matice (viz postup v příloze A).

Pro správné provedení kalibrační metody je důležité pouzdro se senzorem umístit na kalibrační stůl v předem definovaných polohách.

Pro správné používání navržené metody je sepsán návod, který vede uživatele krok po kroku ke kalibraci senzoru. V aplikaci je možné sledovat v reálném čase průběh měření, sledované vypočítané parametry a zda je senzor během měření umístěn ve správné poloze pro kalibraci.

## Seznam použité literatury

- [1] MOESLUND, Thomas B. a Erik GRANUM. *A Survey of Computer Vision-Based Human Motion Capture*. *Computer Vision and Image Understanding* [online]. 2001, 81(3), 231-268. ISSN 10773142. Dostupné z: doi:10.1006/cviu.2000.0897
- [2] JAIMES, Alejandro & SEBE, Nicu. *Multimodal human-computer interaction: A survey*. *Computer Vision and Image Understanding* [online]. 2007, 108(1-2), 116-134. ISSN 10773142. Dostupné z: doi:10.1016/j.cviu.2006.10.019
- [3] ZHOU, Huiyu & HU, Huosheng. *Human motion tracking for rehabilitation—A survey*. *Biomedical Signal Processing and Control* [online]. 2008, 3(1), 1-18. ISSN 17468094. Dostupné z: doi:10.1016/j.bspc.2007.09.001
- [4] MULTON Franck, FRANCE Laure, CANI, Marie-Paule, DEBUNNE Gilles. *Computer animation of human walking: a survey*. *Comput Animat Virtual Worlds*. 1999 ; 10 ( 1 ) : 39 – 54
- [5] REGO P.A., MOREIRA Pedro Miguel, REIS Luis Paulo. *Serious games for rehabilitation: A survey and a classification towards a taxonomy*. *Information Systems and Technologies (CISTI), 2010 5th Iberian Conference on. IEEE; 2010*. p. 1–6
- [6] Vicon Motion Systems Ltd. *Vicon | Award Winning Motion Capture Systems* [online] 2021 [cit. 20.11.2021] Dostupné z: <https://www.vicon.com/>
- [7] XSens motion capture. *Home - Xsens 3D motion tracking* [online] 2021 [cit. 20.11.2021] Dostupné z: <https://www.xsens.com/motion-capture>
- [8] Rokoko Electronics. *Rokoko - Animation and MoCap Tools for all Creators*. [online] 2020 [cit. 20.11.2021] Dostupné z: <https://www.rokoko.com/>
- [9] ZHANG, Zhengyou. *Microsoft Kinect Sensor and Its Effect*. *IEEE Multimedia* [online]. 2012, 19(2), 4-10 ISSN 1070-986X. Dostupné z: doi:10.1109/MMUL.2012.24
- [10] TITS, Mickaël, TILMANNE Joëlle, DUTOIT Thierry a FEDEROLF Peter Andreas. *Robust and automatic motion-capture data recovery using soft skeleton constraints and model averaging*. *PLOS ONE* [online]. 2018, 13(7). ISSN 1932-6203. Dostupné z: doi:10.1371/journal.pone.0199744
- [11] MOU, Tsai-Yun. *Keyframe or Motion Capture? Reflections on Education of Character Animation*. *EURASIA Journal of Mathematics, Science and Technology Education* [online]. 2018, 14(12) [cit. 2022-02-09]. ISSN 13058223. Dostupné z: doi:10.29333/ejmste/99174
- [12] Vox Media, LLC. *Andy Serkis on His Breakthrough Motion-Capture Performances*. *Vulture - Entertainment News - TV, Movies, Music, Books, Theater, Art* [online]. 2022 [cit. 08. 02. 2022]. Dostupné z: <https://www.vulture.com/2018/12/andy-serkis-on-his-breakthrough-motion-capture-performances.html>
- [13] VELASCO Del Pozo, María Belén. *Outer Side: Documento de Proyecto de Videojuego (GDD)*. 2019.
- [14] SANTANA, Cláudia a Filipa da Rosa. *A emoção e o storytelling nos videojogos*, 2019. Dizertační práce.
- [15] USgamer Gamer Network Limited, a ReedPop company. *Detroit: Become Human Voice Cast* [online]. 2022 [cit. 20.04.2022]. Dostupné z: <https://www.usgamer.net/articles/01-06-2018-detroit-become-human-guide-tips-tricks-walkthroughs/detroit-voice-actors-guide>

- [16] Digital Spy - TV, Movies and Entertainment News. *Meet gaming's motion capture master*. [online]. 2022 [cit. 20.04.2022]. Dostupné z: <https://www.digitalspy.com/videogames/uncharted/a660971/meet-nolan-north-uncharted-s-nathan-drake-and-gamings-master-of-motion-capture/>
- [17] Hinge. *Integrated Creative Production Studio Motion Capture in Medicine* [online]. 2022 [cit. 08.02.2022]. Dostupné z: <https://hellohinge.com/motion-capture-in-medicine/>
- [18] PhaseSpace Motion Capture . *Sports & Medical – Infinite Possibilities* [online]. 2017 [cit. 08.02.2022]. Dostupné z: <https://www.phasespace.com/applications/sports-medical/>
- [19] Locogreenbiz.com technologie. *Současnost a budoucnost technologie sledování pohybu ve virtuální realitě*. [online] 2021 [cit. 20.11.2021] Dostupné z: <https://cs.locogreenbiz.org/virtual-reality/452-the-present-and-future-of-motion-tracking-technology-in-virtual-reality.html>
- [20] XSens motion capture. *A history of Motion Capture* [online] 2021 [cit. 20.11.2021] Dostupné z: <https://www.xsens.com/a-history-of-motion-capture>
- [21] MAJER, Martin. *Inerciální senzory a jejich využití*. Zlín 2019, Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, Ústav automatizace a řídicí techniky. 2019, 63 s. Dostupné také z: <http://hdl.handle.net/10563/44575>. Diplomová práce
- [22] Hamada, Vladimír. *Automatická kalibrace inerciálních senzorů*. Brno 2014, Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Diplomová práce
- [23] ŠÍPKA, Tomáš. *Zařízení pro snímání zrychlení a jeho přenos do PC*. Praha 2019. České vysoké učení technické v Praze, Fakulta elektrotechnická. Bakalářská práce
- [24] TME Czech Republic s.r.o. - Elektronické součástky. *Three-axis accelerometer board mikroelektronika - Rozšiřující deska | prototypová deska MIKROE-254* [online]. Dostupné z: [https://www.tme.eu/cz/details/mikroe-254/prislusenstvi-pro-vyvojove-kity/mikroelektronika/three-axis-accelerometer-board/?brutto=1&cy=CZK&gclid=Cj0KCQiA\\_JWOBhDRARIsANymNOajRCXkDyWBTJWmDgRSr53-jnswlsZnhESiBmakfwBJjQ9hS0d6VfIaAl3MEALw\\_wcB](https://www.tme.eu/cz/details/mikroe-254/prislusenstvi-pro-vyvojove-kity/mikroelektronika/three-axis-accelerometer-board/?brutto=1&cy=CZK&gclid=Cj0KCQiA_JWOBhDRARIsANymNOajRCXkDyWBTJWmDgRSr53-jnswlsZnhESiBmakfwBJjQ9hS0d6VfIaAl3MEALw_wcB)
- [25] Letem světem Applem - Apple magazin. *Jak funguje digitální gyroskop? Princip toho, proč iPhone dokáže určit, v jaké je poloze*. [online]. [cit. 24.12.2021]. Dostupné z: <https://www.letemsvetemapplem.eu/2014/01/23/jak-funguje-digitalni-gyroskop-princip-toho-proc-iphone-dokaze-urcit-v-jake-je-poloze/>
- [26] ECLIPSE s.r.o. - dratek.cz. *Arduino gyroskop + akcelerometr* [cit. 24.12.2021]. Dostupné z: [https://dratek.cz/arduino/830-iic-i2c-gyroskop-akcelerometr-modul-mpu-6050.html?gclid=Cj0KCQiA\\_JWOBhDRARIsANymNObuYXJXMx\\_nutZt4AFjX7S8GFI0DCvRr2Acch56627L63vQmwoTy1UaAi\\_oEALw\\_wcB](https://dratek.cz/arduino/830-iic-i2c-gyroskop-akcelerometr-modul-mpu-6050.html?gclid=Cj0KCQiA_JWOBhDRARIsANymNObuYXJXMx_nutZt4AFjX7S8GFI0DCvRr2Acch56627L63vQmwoTy1UaAi_oEALw_wcB)
- [27] *Space Microsystems and Micro/nano Satellites*. Elsevier, 2018 [cit. 2021-11-30]. ISBN 9780128126721. Dostupné z: doi:10.1016/C2014-0-04758-2
- [28] RŮŽIČKA, Jiří. *9DOF senzor pro měření orientace zbraně v prostoru*. Brno 2017. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Diplomová práce
- [29] Botland - obchod s robotikou. *Digitální magnetometr GY-273 3osý I2C 3.3V / 5V - HMC5883L / QMC5883* [online]. [cit. 24.12.2021]. Dostupné



- z: <https://botland.cz/magnetometry/9412-digitalni-magnetometr-gy-273-3osy-i2c-33v-5v-hmc5883l-qmc5883-5903351249614.html>
- [30] Motion Capture Sensor Systems. *Sensor Technology / Sensors Development / News* [online]. Copyright © 2000 [cit. 16.01.2022]. Dostupné z: <https://www.azosensors.com/article.aspx?ArticleID=43>
- [31] Motion Capture System | Technaid - Inertial Motion Capture. *Home / Technaid - Leading Motion - Human Rehabilitation Technologies* [online]. Copyright © 2021 [cit. 16.01.2022]. Dostupné z: <https://www.technaid.com/products/motion-capture-system-tech-mcs-hub-imu>
- [32] AZO Sensors - Sensor technology, Sensor Development. *Motion Capture Sensor Systems*. [online] 2012 [cit. 20.11.2021] Dostupné z: <https://www.azosensors.com/article.aspx?ArticleID=43>
- [33] ZAPLATÍLEK, Luděk. *Návrh stabilizačního systému RC modelu letadla*. Pardubice 2011. Univerzita Pardubice, fakulta elektrotechniky a informatiky. Diplomová práce
- [34] MOHINDER, Grewal & LAWRENCE, Weill & ANGUS, Andrews. *Global Positioning Systems, Inertial Navigation, and Integration*. Hoboken 2007 druhé vydání. DOI: 10.1002/9780470099728.ch3.
- [35] SKOG, Isaac & HÄNDEL, Peter. *Calibration of A MEMS inertial measurement unit*. 2006 XVII IMEKO World Congress Metrology for a Sustainable Development.
- [36] NARANJO, Claudia. *Analysis and Modeling of MEMS based Inertial Sensors*. Stockholm (2008)
- [37] VEŠKRNA, Martin. *Positioning system for small devices using principles of inertial navigation systém*. Brno 2013, Masarykova univerzita, Fakulta informatiky. Diplomová práce
- [38] SUŠÁNEK, Filip. *Určení polohy objektu pomocí kombinace sensorů mobilního telefonu*. Praha 2014, České vysoké učení technické v Praze, fakulta elektrotechnická. Bakalářská práce
- [39] ČSN ISO 16063-1:2004 Metody kalibrace snímačů vibrací a rázů – Část 1: Základní pojetí
- [40] ČSN ISO 5348:1999 vibrace a rázy – Mechanické připevnění akcelerometrů
- [41] ČSN ISO 16063-43 Metody kalibrace snímačů vibrací a rázů - Část 43: Kalibrace akcelerometrů pomocí identifikace parametrů založených na modelu
- [42] PAPAFOTIS, Konstantinos a Paul P. SOTIRIADIS. MAG.I.C.AL.–A Unified Methodology for Magnetic and Inertial Sensors Calibration and Alignment. *IEEE Sensors Journal* [online]. 2019, 19(18), 8241-8251 [cit. 2021-11-30]. ISSN 1530-437X. Dostupné z: doi:10.1109/JSEN.2019.2919179
- [43] NCEI. NOAA National Centers for Environmental Information (NCEI), *World Magnetic Model* [online]. Dostupné z: <https://www.ngdc.noaa.gov/geomag/WMM/>
- [44] AMMANN, Nikoulaus & DERKSEN, Alexander & HECK, Constantin, A novel magnetometer-accelerometer calibration based on a least squares approach, 2015 International Conference on Unmanned Aircraft Systems (ICUAS), 2015, pp. 577-585, doi: 10.1109/ICUAS.2015.7152338
- [45] ZHONG, Yuan & XU, Yong & HE, Naixin & YU, Xiang. *A New Drone Accelerometer Calibration Method*. 2018 37th Chinese Control Conference (CCC), 2018, pp. 9928-9933, doi: 10.23919/ChiCC.2018.8482568

- [46] GLUECK, Manuel & BUHMANN, Alexander & MANOLI, Yiannos. *Autocalibration of MEMS accelerometers*. 2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings, 2012, pp. 1788-1793, doi: 10.1109/I2MTC.2012.6229157.
- [47] FROSIO, Iuri & PEDERSINI, Federico & BORGHESE, Nunzio Alberto, *Autocalibration of MEMS Accelerometers*, in IEEE Transactions on Instrumentation and Measurement, vol. 58, no. 6, pp. 2034-2041, June 2009, doi: 10.1109/TIM.2008.2006137."
- [48] STANOVSKÝ, David a Libor BARTO. *Počítačová algebra*. Praha: Matfyzpress, 2011. ISBN 978-80-7378-167-5.
- [49] MAŠKOVÁ, Eva. *Gram-Schmidtův ortogonalizační proces a LLL algoritmus*. Plzeň 2012. Západočeská univerzita v Plzni, fakulta pedagogická. Diplomová práce
- [50] JIANG, Ming, Hong SHANG, Zhelong WANG, Hongyi LI a Yuechao WANG. *A method to deal with installation errors of wearable accelerometers for human activity recognition*. *Physiological Measurement* [online]. 2011, 32(3), 347-358 [cit. 2022-05-07]. ISSN 0967-3334. Dostupné z: doi:10.1088/0967-3334/32/3/006
- [51] SYED, Zainab & AGGARWAL, Prizanka & GOODALL, Christopher & NIU, Xiaoji & EL-SHEIMY, Naser. *A new multi-position calibration method for MEMS inertial navigation systems*. *Measurement science and technology*, 18(7), 1897.
- [52] QURESHI, Umar & GOLNARAGHI, Farid. *An Algorithm for the In-Field Calibration of a MEMS IMU*, in IEEE Sensors Journal, vol. 17, no. 22, pp. 7479-7486, 15 Nov.15, 2017, doi: 10.1109/JSEN.2017.2751572.
- [53] WU, Yuanxin & PEI, Ling. *Gyroscope Calibration via Magnetometer*, in IEEE Sensors Journal, vol. 17, no. 16, pp. 5269-5275, 15 Aug.15, 2017, doi: 10.1109/JSEN.2017.2720756.
- [54] ZHU, Jiaxin & WANG, Zhaoli & HUANG, Shiping & DING, Wei. *An improved Calibration Technique for MEMS Accelerometer-Based Inclinometers*. *Sensors*. Doi: 10.3390/s20020452
- [55] Abyarjoo, Fatemeh & Barreto, Armando & Cofino, Jonathan & Ortega, Francisco. (2012). *Implementing a Sensor Fusion Algorithm for 3D Orientation Detection with Inertial/Magnetic Sensors*. 10.13140/2.1.1295.0406.

## Příloha A: Uživatelský návod pro použití navrženého nástroje

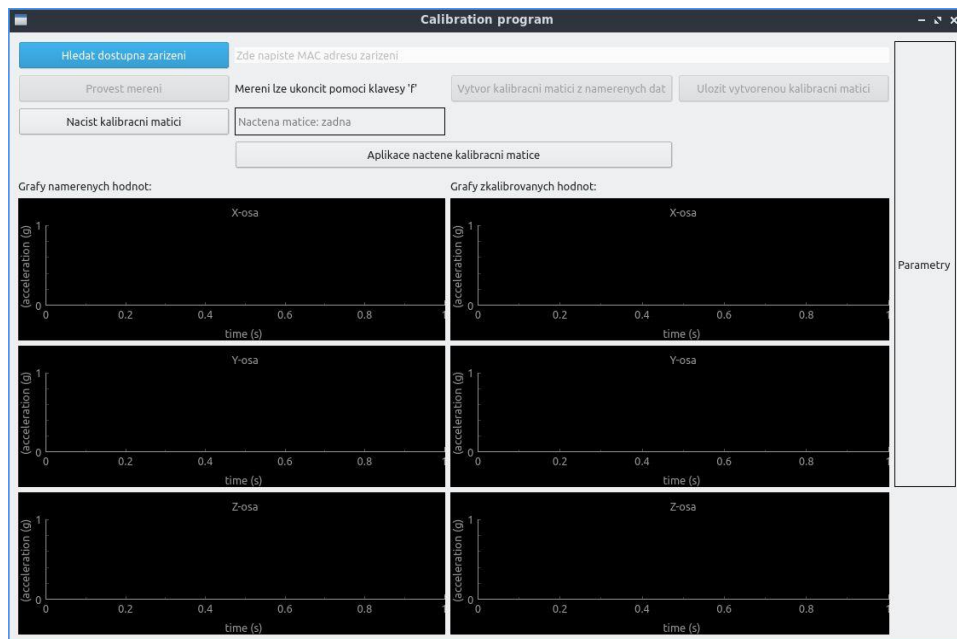
Pro korektní výsledky navržené metody je nutné postupovat podle následujícího sledu činností.

1. Senzor musí být pevně umístěn do předpřipraveného pouzdra. Pouzdro je navrženo tak aby byl senzor lehce umístitelný v přesné pozici. Do pouzdra je navíc umístěna baterie napájející senzor.
2. U navrženého kalibračního stolu je důležité zkontrolovat, zda je ve vodorovné poloze např. pomocí vodováhy (viz obrázek 8.1 vlevo) při tomto seřizování je důležité, aby bublinka vodováhy byla co nejvíce ve středu. Pro přesnější nastavení vodorovnosti stolu lze použít digitální úhloměr (viz obrázek 8.1 uprostřed a vpravo). Při nastavování stolu pomocí digitálního úhlooměru je pro vodorovnou osu dovolená maximální hodnota  $\pm 0,2$  a pro svislou osu minimálně 89,8 a maximálně 90,2. Pokud stůl není ve vodorovné poloze je zapotřebí stůl do vodorovné polohy nastavit pomocí nastavitelných nožiček.

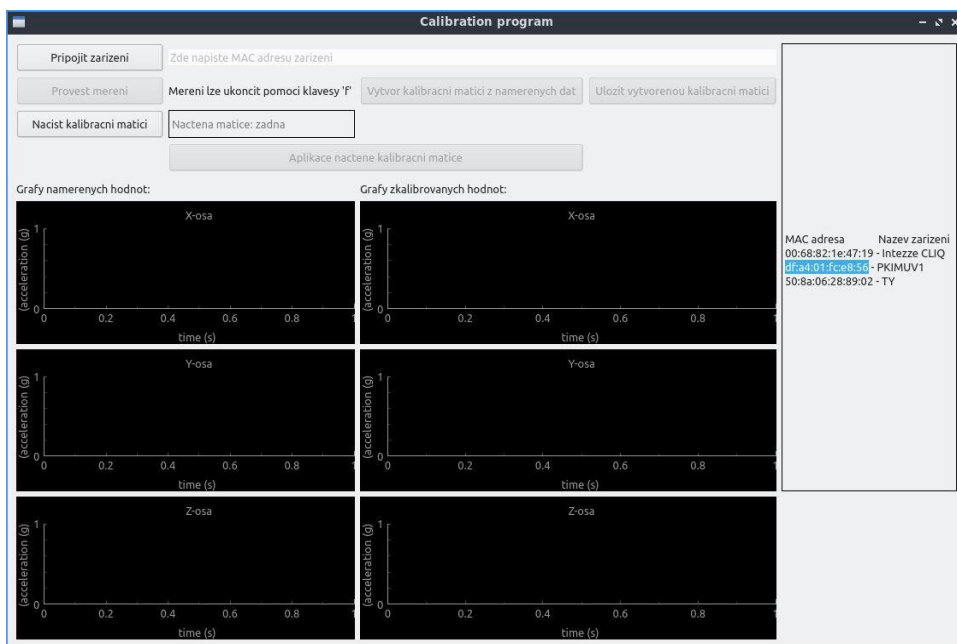


Obr. 8.1: Korektní vodorovná poloha podle vodováhy (vlevo), korektní vodorovná poloha podle digitálního úhlooměru (uprostřed) a korektní svislá poloha podle digitálního úhlooměru

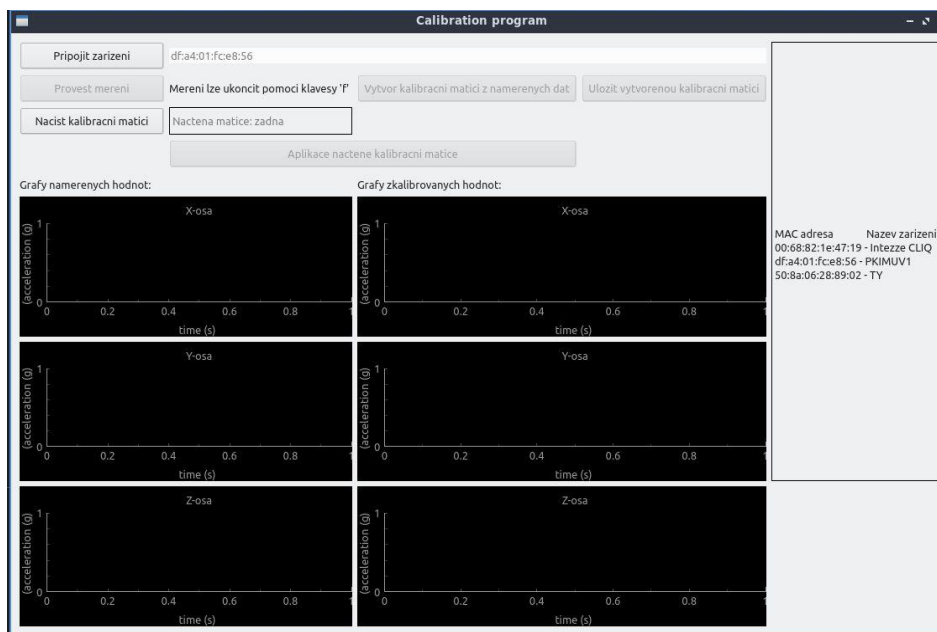
3. Před připojením senzoru je vhodné umístit senzor na kalibrační stůl v 1. definované poloze (osou  $x$  kolmo k zemi). Senzor je nutné připojit pomocí Bluetooth, po stisknutí tlačítka připojit zařízení (viz obrázek 8.2) je provedeno skenování dostupných Bluetooth zařízení v okolí. Po skončení skenování jsou dostupná zařízení vypsána názvem a MAC adresou do parametrového okna (viz obrázek 8.3), ve kterém lze MAC adresu překopírovat do řádku pro MAC adresu (viz obrázek 8.4). Po zadání MAC adresy do okna se znova stiskne tlačítko pro připojení zařízení. V případě správné MAC adresy je zařízení připojeno a tlačítko pro připojení zešedivěno (viz obrázek 8.5). V případě špatně zadané MAC adresy není připojení provedeno a je nutné zkontrolovat MAC adresu a provést připojení znovu dokud není úspěšné.



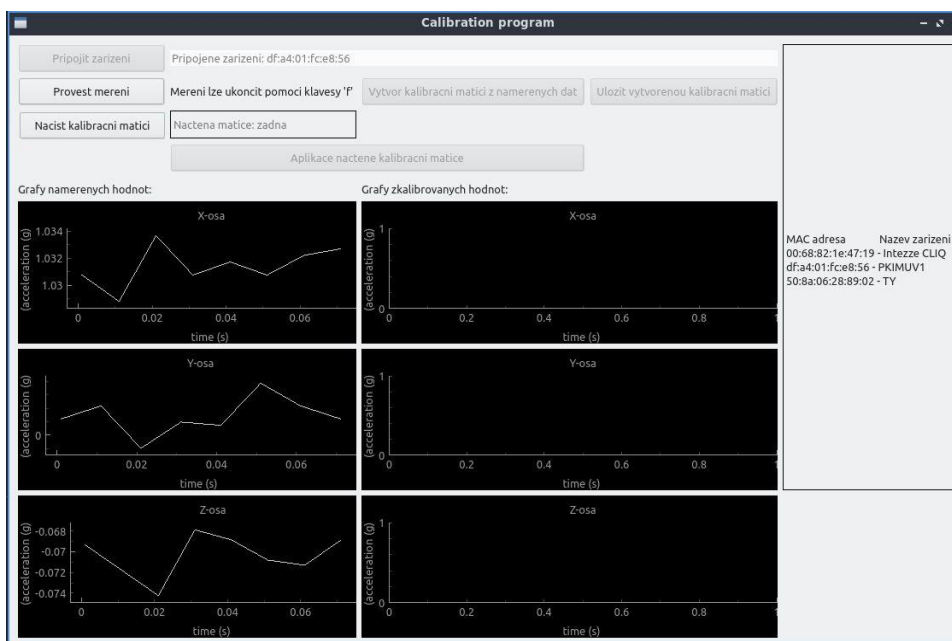
Obr. 8.2: Rozhraní před skenováním Bluetooth zařízení s výrazněným tlačítkem pro hledání dostupných zařízení



Obr. 8.3: Rozhraní s výčtem dostupných Bluetooth zařízení s vyznačenou konkrétní MAC adresou měřeného zařízení



Obr. 8.4: Rozhraní s vloženou MAC adresou do kolonky



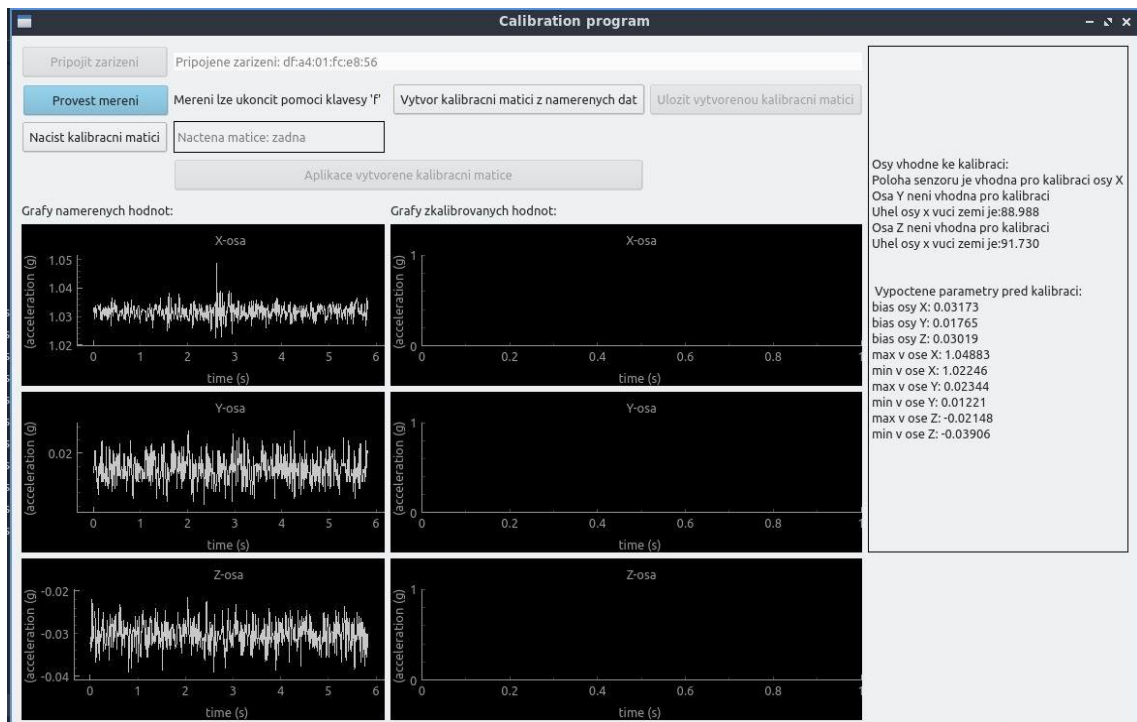
Obr. 8.5: Rozhraní po úspěšném spárování z Bluetooth zařízením a vykreslením prvního bloku naměřených hodnot

- Na kalibrační stůl, který je ve vodorovné poloze, se umístí pouzdro se senzorem v definované poloze (pokud již nebyla umístěna), kdy je osa  $x$  kolmo k zemi. Pokud je pouzdro opatřeno křídélky pro upevnění pásu, lze ji umístit na okraj stolu (viz obrázek 8.6) V aplikaci lze na konci každého měřicího bloku zkontrolovat, zda je senzor správně umístěn (viz obrázek 8.7). Kontrola polohy spočívá v kontrole naměřeného bloku zda se v něm nevyskytují abnormální hodnoty a v kontrole parametrů zda odpovídají

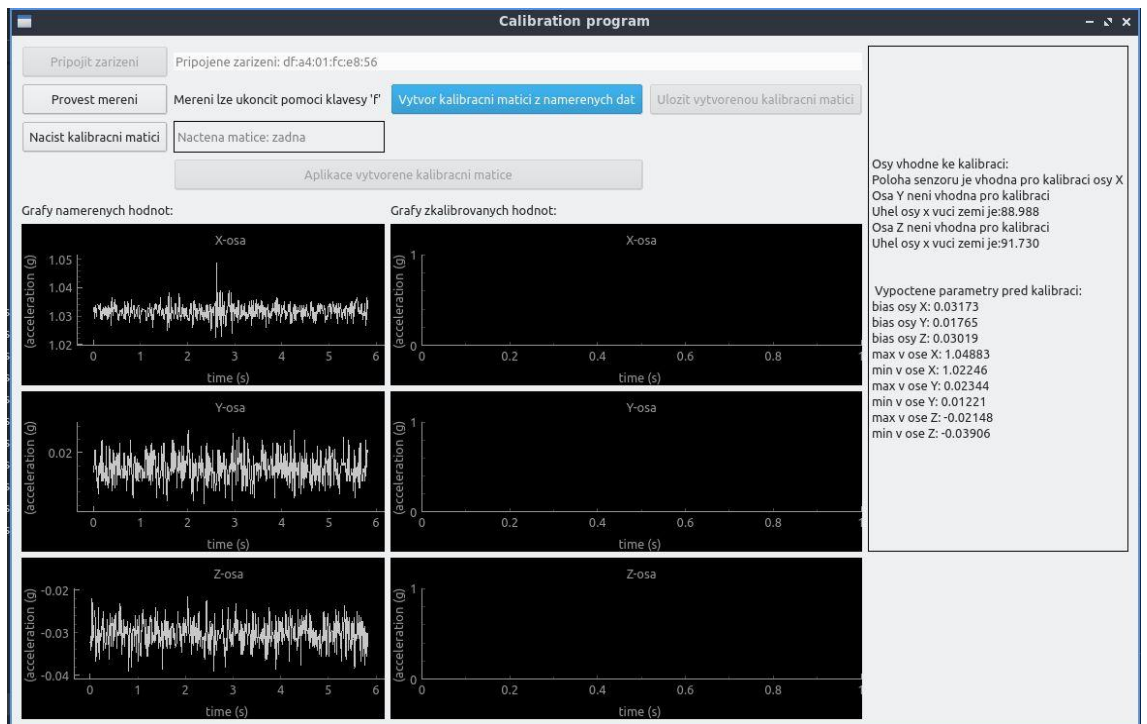
požadavku. Po kontrole polohy se provede alespoň 5 vteřin měření (viz obrázek 8.8) a následné vytvoření kalibrační matice pro osu  $x$ .



Obr. 8.6: Korektní umístění senzoru pro osu  $x$  v kolmém směru k zemi

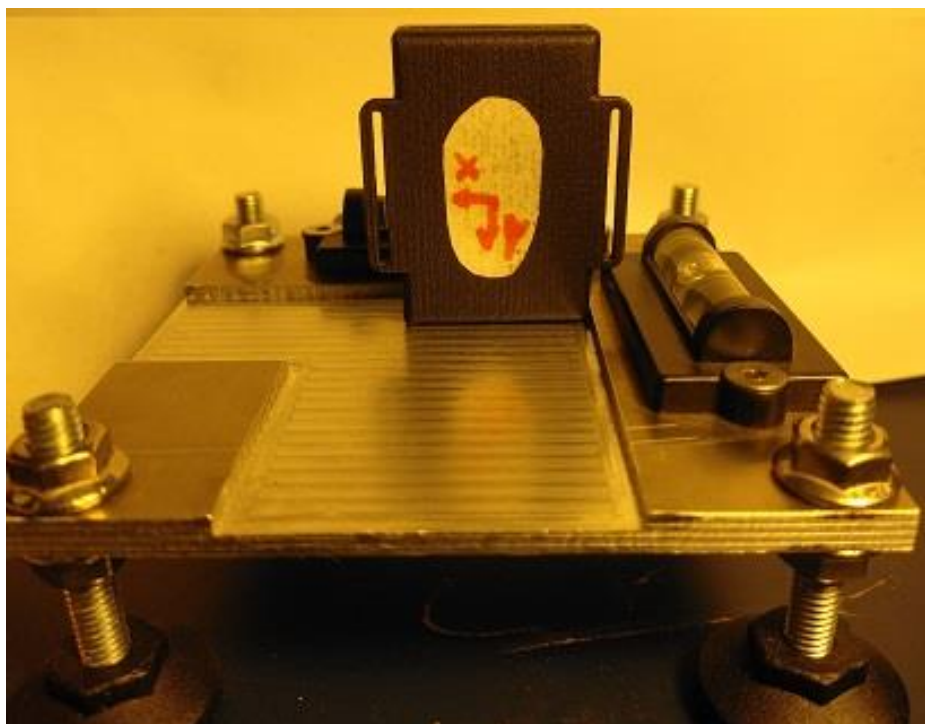


Obr. 8.7: Kontrola správného umístění senzoru

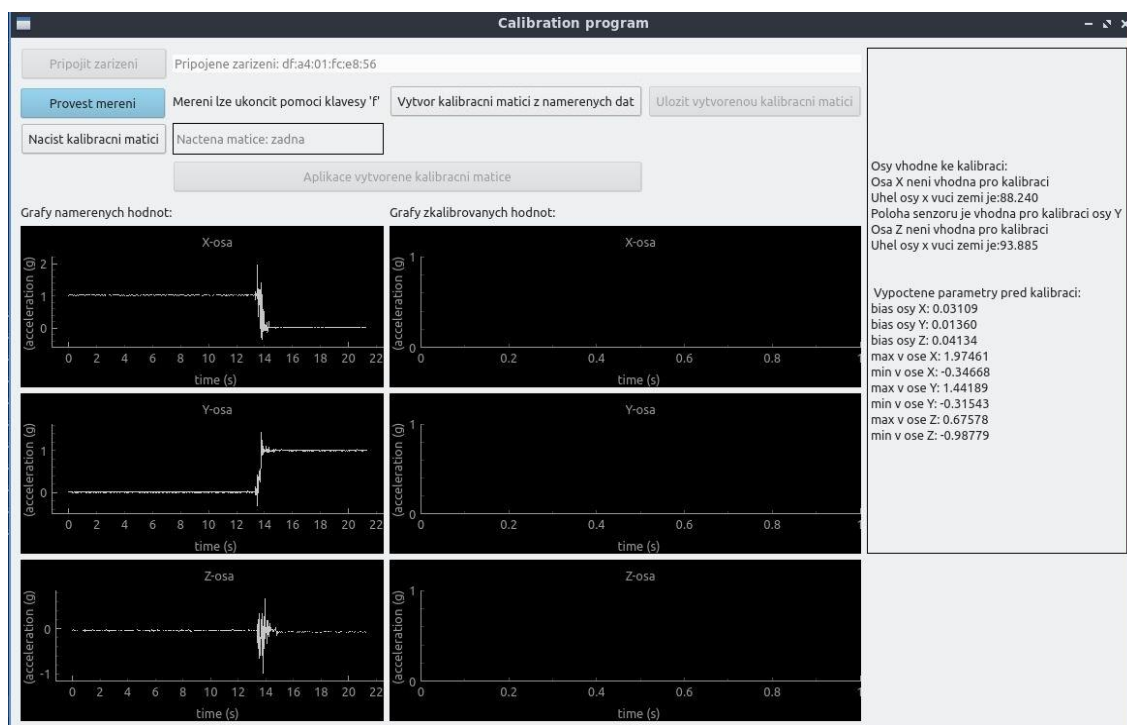


Obr. 8.8: Naměření alespoň pěti vteřin měření, po kterém následuje stisknutí tlačítka vytvoření kalibrační matice

- Na kalibrační stůl, který je ve vodorovné poloze, se umístí pouzdro se senzorem v definované poloze, kdy je osa y kolmo k zemi (viz obrázek 8.9). V aplikaci lze na konci každého měřicího bloku zkontrolovat zda je senzor správně umístěn (viz obrázek 8.10). Kontrola polohy spočívá v kontrole naměřeného bloku zda se v něm nevyskytují abnormální hodnoty a v kontrole parametrů zda odpovídají požadavku. Po kontrole polohy se provede alespoň 5 vteřin měření (viz obrázek 8.11) a následně vytvoření kalibrační matice pro osu y.

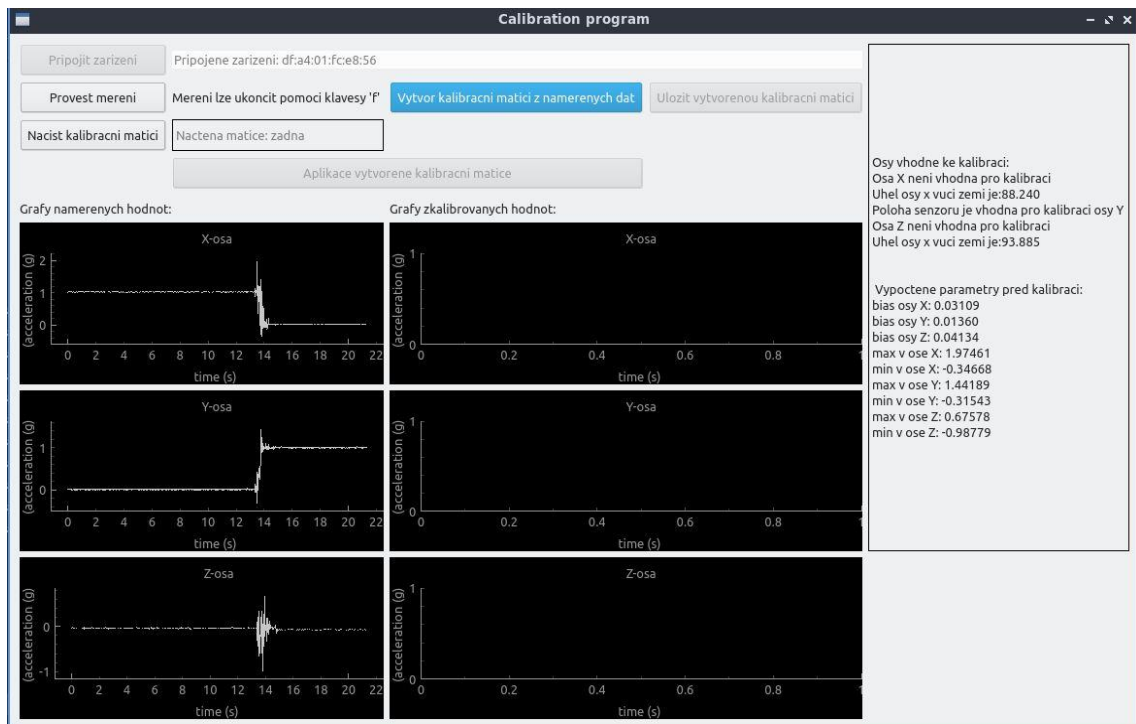


Obr. 8.9: Korektní umístění senzoru pro osu y v kolmém směru k zemi



Obr. 8.10: Kontrola správného umístění senzoru



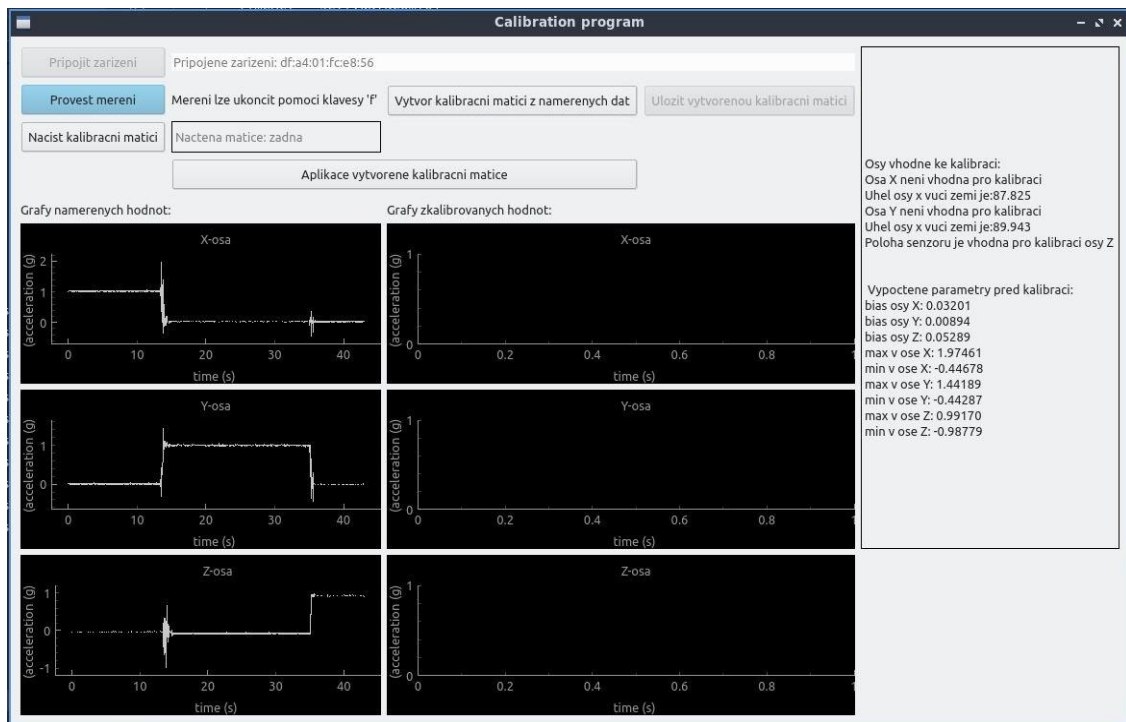


Obr. 8.11: Naměření alespoň pěti vteřin měření, po kterém následuje stisknutí tlačítka vytvoření kalibrační matice

- Na kalibrační stůl, který je ve vodorovné poloze, se umístí pouzdro se senzorem v definované poloze, kdy je osa  $z$  kolmo k zemi (viz obrázek 8.12). V aplikaci lze na konci každého měřicího bloku zkontrolovat zda je sensor správně umístěn (viz obrázek 8.13). Kontrola polohy spočívá v kontrole naměřeného bloku zda se v něm nevyskytují abnormální hodnoty a v kontrole parametrů zda odpovídají požadavku. Po kontrole polohy se provede alespoň 5 vteřin měření (viz obrázek 8.14) a následně vytvoření kalibrační matice pro osu  $z$ .



Obr. 8.12: Korektní umístění senzoru pro osu  $z$  v kolmém směru k zemi

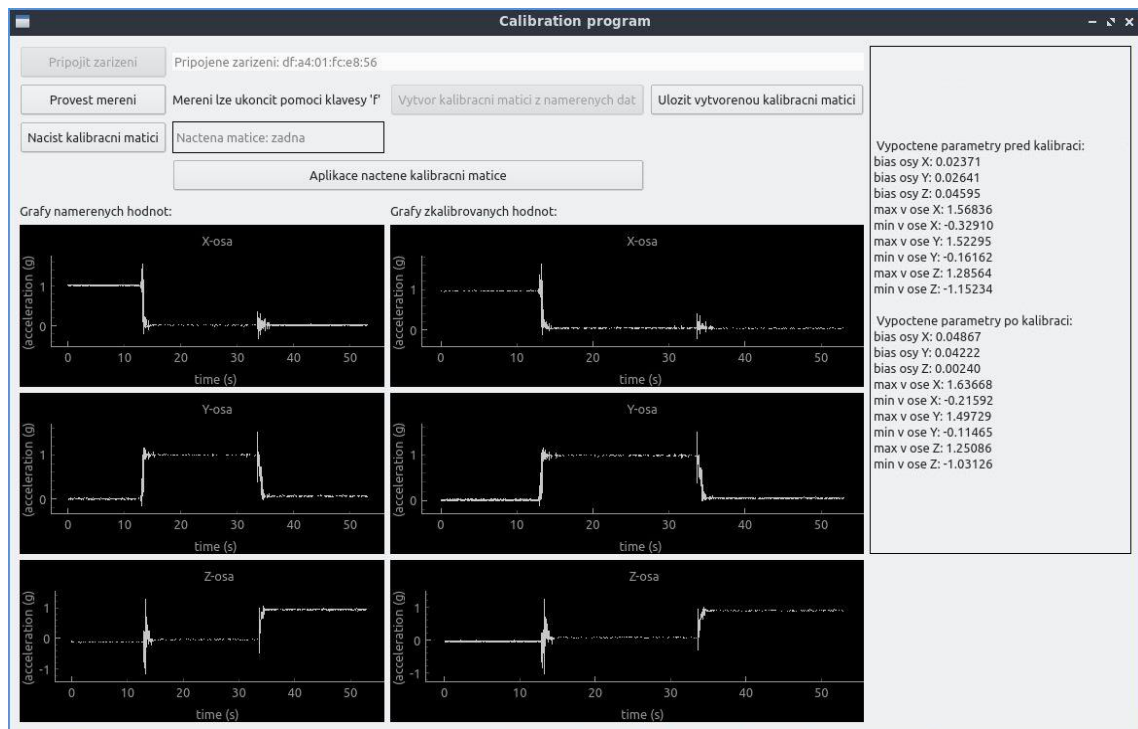


Obr. 8.13: Kontrola správného umístění senzoru



Obr. 8.14: Naměření alespoň pěti vteřin měření, po kterém následuje stisknutí tlačítka vytvoření kalibrační matice

- Po vytvoření kalibrační matice v bodě 6. je možné exportovat kalibrační matici a aplikovat kalibrační matici na naměřená data, která jsou vykresleny do grafu v rozhraní (viz obrázek 8.15).



Obr. 8.15: Rozhraní po aplikaci vytvořené kalibrační matice

8. Z naměřených a zkalibrovaných dat jsou vypočítány důležité sledované parametry a zobrazeny v okně s parametry.
9. Při dalším měření lze na senzor aplikovat kalibrační matici pomocí tlačítka aplikace kalibrační matice a výsledky kalibrace zobrazit v aplikaci společně z aktualizovanými parametry.
10. Po stisknutí tlačítka uložit vytvořenou kalibrační matici je vytvořená matice uložena ve formátu .csv do adresáře s programem. Při započítí nového měření lze uloženou matici načíst do programu pomocí tlačítka načíst kalibrační matici.

