

České vysoké učení technické v Praze

Fakulta elektrotechnická

Katedra kybernetiky



Bakalářská práce

Objektivizace rozsahu základního pohybu dílčích částí
těla při rehabilitaci

Autor: Marek Syrový

Vedoucí práce: Ing. Petr Novák, Ph.D.

leden 2022

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Syrový** Jméno: **Marek** Osobní číslo: **483782**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra kybernetiky**
Studijní program: **Kybernetika a robotika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Objektivizace rozsahu základního pohybu dílčích částí těla při rehabilitaci

Název bakalářské práce anglicky:

Objectification of Basic Movement of Partial Parts of the Body during Rehabilitation

Pokyny pro vypracování:

V současné době velmi stoupá počet pohybových úrazů. Stále však schází jednoduchá a objektivní (tedy opakovatelná) diagnostika pro hodnocení zlepšení rozsahu pohybu v průběhu rehabilitace těchto úrazů u pacientů. Cíl práce spočívá ve vytvoření jednoduchého (levného), ale přesto účelného (dostačujícího) zařízení umožňujícího snímat (zejména pomalé) základní dílčí pohyby postavy člověka (pohyb hlavou, trupem, rukou, nohou). Zadání se skládá z následujících částí:

- 1) Zhodnoťte již existující (pouze cenově dostupná) zařízení / systémy pro objektivizaci rozsahu pohybového aparátu člověka. Zaměřte se na jejich snadné využití jak pro medicínské, tak i experimentální / vědecké účely (otevřenost, přístup k datům, konfigurace, ...).
- 2) Vytvořte aplikaci, jež bude pomocí 3D modelu postavy člověka vhodně prezentovat měřené parametry rozsahu pohybu dílčích částí pohybového aparátu člověka (vybrané části těla, rozsahy pohybu, dosažené úhly, ...).
- 3) Pro měření pohybu vytvořte dva senzory. Jeden méně přesný, ale snadno dostupný (např. formou aplikace na běžném mobilním telefonu) a druhý precizní (např. použitím ARM + IMU + USB/Bluetooth se vzorkováním i přes 100Hz).
- 4) Využijte některé (jednoduché) již existující algoritmy pro dostatečně přesnou detekci pohybu (komplementární filtr, tzv. Mahony filtr, nebo tzv. Madgwick filtr). Uveďte a zdůvodněte jejich případné nevýhody.
- 5) Výsledek práce demonstруйте na vzorových postupech / příkladech pro využití navrženého zařízení k objektivizaci zmíněných základních dílčích pohybů člověka.

Seznam doporučené literatury:

- [1] WWW stránky nalezených / obdobných projektů, manuály a další relevantní informace
 - [2] Price Mark, C# 8.0 and .NET Core 3.0 - Modern Cross-Platform Development, Packt, 2019
 - [3] MacDonald Matthew, Pro WPF 4.5 in C#, APress, 2012
 - [4] Noviello Carmine, Mastering STM32, LeanPub, 2017
- Další potřebné materiály poskytne vedoucí práce.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Petr Novák, Ph.D., Analýza a interpretace biomedicínských dat FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **06.01.2021**

Termín odevzdání bakalářské práce: **04.01.2022**

Platnost zadání bakalářské práce: **30.09.2022**

Ing. Petr Novák, Ph.D.
podpis vedoucí(ho) práce

prof. Ing. Tomáš Svoboda, Ph.D.
podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Prohlášení autora práce

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 4. 1. 2022

Marek Syrový

Poděkování

Rád bych poděkoval vedoucímu mé bakalářské práce Ing. Petru Novákovi, Ph.D. za odborné vedení, za připomínky a konstruktivní kritiku, které mě vedly vpřed.

Abstrakt

Tato práce se zabývá vývojem jednoduchého a levného systému pro měření základních dílčích pohybů postavy člověka a sloužícího například k objektivizaci postupu rehabilitačního cvičení. Součástí práce je shrnutí již existujících podobných systémů / zařízení, jejich následné zhodnocení a zdůvodnění potřeby vývoje vlastního zařízení. Cílem práce tedy bylo navrhnout a vytvořit celkem jednoduché a cenově dostupné zařízení s dostatečnou přesností, avšak ne zcela kompletní pro profesionální použití, ale spíše vhodné pro experimenty a další vývoj.

Navržené zařízení se skládá ze dvou hlavních částí. První částí je pohybový IMU/MARG senzor (obsahující akcelerometr, gyroskop a magnetometr), a to ve dvou variantách. Buď formou vlastního vytvořeného zařízení obsahující senzor MPU9250 a doplněného ARM procesorem, nebo formou běžného chytrého mobilního telefonu. Druhou částí je aplikace na PC vytvořená na platformě .NET Core, jež nejprve zpracovává data ze senzoru vhodnými AHRS algoritmy pracujícími na principu komplementárního filtru jejichž výstupem jsou buď Eulerovy úhly, nebo kvaterniony. Aplikace dále vhodně reprezentuje měřené pohyby člověka formou 3D modelu postavy. Poslední částí práce je popis ovládání celkového zařízení a příklad podrobného postupu vykonání jednoho vzorového měření.

Klíčová slova

Pohybový senzor, IMU, MARG, MPU9250, akcelerometr, gyroskop, magnetometr, ARM, Bluetooth, USB, AHRS algoritmus, komplementární filtr, .NET Core, WPF, 3D model postavy, rehabilitace

Abstract

This thesis deals with the development of a simple and inexpensive system for measuring the basic partial movements of the human figure and serving, for example, to objectify rehabilitation exercises. Part of the thesis is a summary of already existing similar systems / devices, their subsequent evaluation and justification of the need to develop our own equipment. The aim of the work was to design and create a fairly simple and affordable device with appropriate accuracy, but not completely complete for professional use, but rather suitable for experiments and further development.

The proposed device consists of two main parts. The first part is a motion IMU / MARG sensor (containing an accelerometer, gyroscope and magnetometer) in two variants. Either in the form of a own created device containing the MPU9250 sensor and supplemented with an ARM processor, or in the form of a regular smartphone. The second part is a PC application written on the .NET Core platform, which first processes data from the sensor using suitable AHRS algorithms working on the principle of a complementary filter which output is either Euler angles or quaternions. The application also appropriately represents the measured movements of a person in the form of a 3D model of the figure. The last part of the thesis is a description of the control of the overall device and a record of the detailed procedure of performing one exemplary measurement.

Keywords

Motion sensor, IMU, MARG, MPU9250, accelerometer, gyroscope, magnetometer, ARM, Bluetooth, USB, AHRS algorithm, complementary filter, .NET Core, WPF, 3D character model, rehabilitation

Obsah

Úvod	1
1 Současný stav	3
1.1 Exoskelet.....	3
1.2 Kamera	3
1.3 IMU senzory.....	4
1.3.1 Armeo Senso.....	5
1.3.2 Pablo.....	6
1.3.3 Bimeo Pro	6
1.3.4 Moover	7
1.3.5 Lynx.....	7
1.3.6 Riablo a Kari.....	8
1.3.7 BTS Telerehab.....	8
1.3.8 Sword health	9
1.3.9 ReFlex	9
1.3.10 BPMpathway	10
1.3.11 InterACTION.....	10
1.4 Souhrn	11
2 Zadání práce	13
2.1 Současný stav	13
2.2 Aplikace	13
2.3 Dva senzory	13
2.4 Algoritmy	13
2.5 Demonstrace výsledku	13
3 Měření orientace v prostoru	14
3.1 Eulerovy úhly	14
3.2 Kvaterniony	14
4 Návrh řešení	16
4.1 Senzor polohy / orientace	16
4.1.1 Senzorový modul s MPU9250/9255.....	17
4.1.2 Senzor tvořen mobilním telefonem	17
4.2 Přenos dat do počítače.....	18
4.3 Formát přenosu dat.....	19
4.4 Komunikace – formát dat.....	19
4.4.1 Textový protokol.....	20

4.4.2	Binární protokol.....	20
4.5	Algoritmy	21
4.5.1	Princip Komplementárního filtru.....	21
4.5.2	Kalmánův filtr	22
4.5.3	Výpočet úhlů orientace ze surových dat senzorů	22
4.5.4	„Dirty“ komplementární filtr	23
4.5.5	Komplementární filtr s horní / dolní propustí (lineární).....	23
4.5.6	Non-linear komplementární filtr	25
4.5.7	Mahony filtr	26
4.5.8	Madgwick filtr.....	26
4.6	Podpora nastavení / experimentů / kalibrace.....	27
4.6.1	Zobrazení dat ze senzorů.....	27
4.6.2	Kalibrace senzoru	27
4.6.3	Kalibrace akcelerometru	28
4.6.4	Kalibrace gyroskopu	29
4.6.5	Kalibrace magnetometru.....	30
4.7	Implementace a struktura aplikace.....	30
4.7.1	RehabMain	31
4.7.2	RehabCommun	31
4.7.3	RehabData	33
4.7.4	RehabCalibration.....	33
4.7.5	RehabTimers.....	34
4.7.6	RehabAlgorithms	34
4.7.7	RehabMovement.....	34
4.7.8	RehabAngles	35
4.7.9	RehabMeasure	35
4.7.10	RehabResults	36
5	Experimentální aplikace	38
5.1	Nastavení.....	38
5.2	Spuštění aplikace.....	41
5.2.1	Profil aplikace	41
5.2.2	Spuštění aplikace (poprvé)	41
5.3	Oblast s postavou (3D scéna)	42
5.4	Manuální pohyb	43
5.5	Nastavení komunikace (VCP, Bluetooth).....	43
5.5.1	Připojení senzoru přes USB	43

5.5.2	Připojení senzoru (mobilu) přes Bluetooth	44
5.6	Data ze senzorů	45
5.6.1	Graph 2D Lines	45
5.6.2	Graph 2D XY.....	47
5.6.3	Graph 3D XYZ.....	47
5.7	Kalibrace	48
5.8	Měření / cvičení.....	49
5.9	Výsledky.....	51
6	Příklad použití aplikace	54
	Závěr	58
	Reference	59
	Seznam zkratk.....	63
	Seznam obrázků	64

Úvod

Rozvojem lidských činností, nejen těch skutečně potřebných, ale v současné době zejména spíše těch zábavných, stále více přibývá různých typů úrazů. Mezi nejčastější lze zřejmě řadit pohybové úrazy (z jízdy autem / kolem, lyží, běhu, ...), tedy nejrozumnější zejména kostní zlomeniny, potřhané svaly a další typy způsobující časově různě dlouhá omezení i základního pohybu.

Při nápravě těchto typů úrazů jsou často důležité tři skutečnosti. V první řadě je to správná diagnostika, v našem případě „následků úrazu“, tedy jak je pohyblivost člověka úrazem omezena. Druhá, a samozřejmě nejpodstatnější, je správná rehabilitace neboli cvičení. Třetí, na první pohled poněkud nepodstatná činnost, spočívá v možnosti hodnotit průběh cvičení / rehabilitace a tím posuzovat správnost, a tedy i úspěšnost navrženého cvičení.

Návrh cvičení a případně průběh cvičení vychází zcela z rozhodnutí lékaře a není tedy nikterak součástí této práce. Avšak diagnostiku pohybu a kontrolu dosažených výsledků lze velmi často vhodně parametrizovat, a tím tedy v podstatě vytvořit dostatečně objektivní měření. Pro takováto hodnocení se v současné době stále často využívá subjektivní pohled lékaře / ošetřujícího, kdy jsou pomocí úhloměru nebo pravítka přiloženého přímo na tělo měřeného člověka odečítány požadované hodnoty. Samozřejmě velmi záleží na aktuálním pohledu lékaře / ošetřujícího, na přesnosti přiložení měřidla, spolupráci pacienta a mnoha dalších vlivech. Opakovatelnost měření tedy neposkytuje dostatečně stejné výsledky a celý proces je rovněž časově velmi náročný a zdlouhavý. Navíc pokud by mělo být takovéto měření vykonáno po každém cvičení, tak jde o velké časové zatížení ošetřujícího personálu.

Naskytá se tedy otázka, zda lze měření rozsahu pohybu při diagnostice a po cvičení uskutečnit mnohem rychleji a správněji, tedy nějak dostatečně objektivně a opakovatelně. Ještě lepším řešením by byla možnost měření rozsahu pohybu přímo v průběhu samotného cvičení, což může být u některých typů cviků celkem snadné.

Pokud omezíme rozsah pohybů / cviků na ty základní jako jsou pohyb hlavou (otáčení vpravo / vlevo, úklon vlevo / vpravo, předklon vpřed / vzad), tělem-hrudí (otáčení, předklon, úklon), rukou (předpažit, upažit, otáčení, ...) a další podobného stylu, lze velmi snadno tyto dílčí pohyby převést na příslušné úhly podle souřadnic ve 3D prostoru. V případě hlavy jde při úklonu vlevo / vpravo o otáčení kolem osy X, při předklonu vpřed / vzad o otáčení kolem osy Y a při jejím otáčení vpravo / vlevo se jedná o otáčení kolem osy Z. Zcela stejný princip je samozřejmě aplikovatelný na hrud', ruce a nohy. Tedy pomocí vhodně softwarově vytvořených / umístěných úhloměrů (ve zmíněném 3D zobrazení) lze snadno měřit rozsahy těchto typů pohybů.

V současné době již není nutno tyto mechanické úhlooměry přikládat přímo na měřenou osobu a ihned na nich odečítat měřené hodnoty. Dnes existují různé dostatečně malé pohybové senzory, ty lze na měřenou osobu velmi snadno upevnit (například pomocí pásků), z nich přenášet data do nedalekého počítače, tam je zpracovávat a získávat z nich tedy požadované úhly informující o rozsahu příslušného typu pohybu. Ne vždy stejné umístění senzorů (ani není reálně možné) se řeší automatickou kalibrací před vlastním měřením / cvičením. Takto lze velmi snadno vykonat nejen prvotní diagnostiku rozsahu pohybu postižené osoby, ale rovněž i přímo v průběhu samotného cvičení získávat dosažené parametry a tím hodnotit správnost / účinnost navrženého cvičení.

Pokud zde hovoříme o správnosti měření (rozsahu pohybu), tak samozřejmě uvažujeme pouze nějakou omezenou přesnost přijatelnou pro daný typ měření. Například otáčení hlavou vpravo / vlevo nemá význam měřit na desetiny úhlového stupně, v mnoha případech ani na stupně, neboť samotný třes tohoto pohybu (snažící se) postižené osoby může být mnohem větší než jeden stupeň při měření.

Z tohoto důvodu je vhodné měření vykonávat raději opakovaně. Při hodnocení cvičení nejde ani o to, jak je skutečně číselně velký dosažený úhel ve sledovaném rozsahu pohybu, ale o to zda se tento rozsah pohybu postupně zvětšuje, což signalizuje správnost navrženého rehabilitačního postupu.

Pro lékaře / ošetřujícího však mnohdy pouze číselné vyjádření výsledků měření není to nejlepší podání. Čísla jsou sice přesná (objektivní), ale jejich porovnání / hodnocení vyžaduje určité soustředění a často je nelze tedy posoudit pouhým (letmým) pohledem. Z tohoto důvodu jsou vhodnější různé formy i jednoduchých grafů.

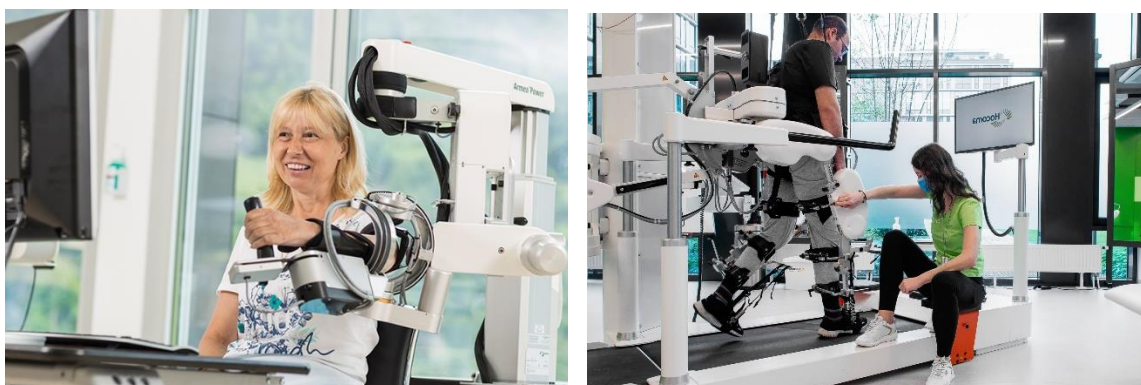
Práce se tedy zabývá návrhem jednoduché aplikace, využívající pohybový senzor pro měření rozsahu dílčích pohybů člověka, vhodné jak pro prvotní diagnostiku, tak i pro sledování zlepšení v průběhu cvičení / rehabilitace. Výsledky práce jsou určeny spíše pro experimenty než přesná lékařská vyšetření.

1 Současný stav

V současné době můžeme rozlišit tři základní technologie pro snímání pohybů lidského těla používané nejen při rehabilitaci, jimiž jsou: exoskelet, kamera a IMU (Inertial Measurement Unit) senzory [1]. V této části práce se nejprve zaměříme na základní princip těchto tří technologií, rovněž na jejich případné výhody / nevýhody a příklady některých komerčních produktů. Dále se podrobněji podíváme na konkrétní cenově dostupnější již existující zařízení založené pouze na technologii s IMU senzory, jíž se věnuje celý zbytek této práce.

1.1 Exoskelet

První technologií a asi i nejstarší je tzv. exoskelet tvořený mechanickou vnější kostrou pro příslušnou část těla. Ke snímání aktivity mohou být na kostře umístěny různé senzory na snímání pohybu, síly či rotace realizované například potenciometry nebo optickými enkodéry. Navíc robotický exoskelet obvykle disponuje i aktivačními prvky jako jsou motory / pohony. Oproti dále zmiňovaným technologiím je toto hlavní výhodou exoskeletu. Tím je umožněno nejen snímání pohybů, ale i aktivní pohybová podpora a vedení těla, což je výhodné zejména při rehabilitaci pacientů s těžkým poškozením hybnosti. Nevýhodou jsou však vysoké pořizovací náklady a potřeba specializovaného pracoviště pro takovéto zařízení. Toto tedy není vůbec vhodné např. pro domácí rehabilitaci. Jako příklad komerčně prodáváného exoskeletu by se dal uvést produkt Armeo Power (Hocoma, Švýcarsko) [2].

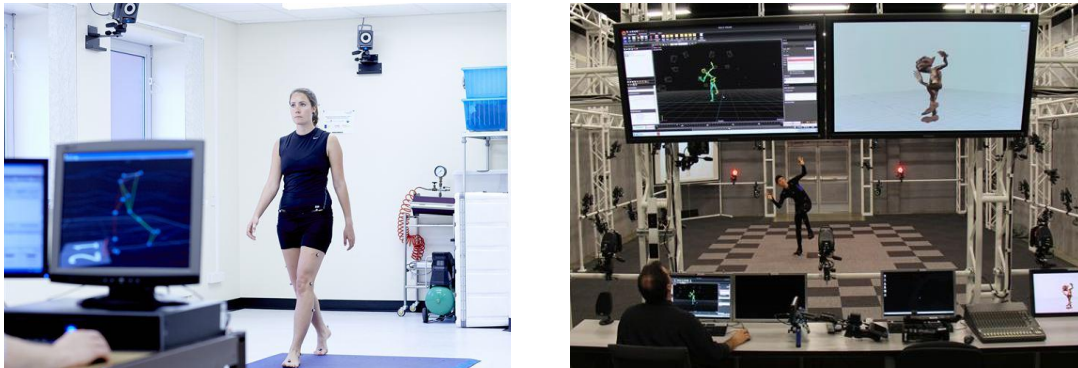


Obrázek 1: Exoskeletické přístroje firmy Hocoma: Armeo Power (vlevo) [2] a Lokomat (vpravo) [3].

1.2 Kamera

Druhá celkem rozšířená technologie je založena na principu snímání pohybů člověka pomocí kamer a využití algoritmů pro počítačové vidění. Nejpresnější technologie v této oblasti pracuje na systému stereofotogrammetrie založené na snímání značek a vyžaduje soustavu obsahující množstvím kamer disponující infračervenými iluminátory. Při zachycování pohybů člověka musí být na těle umístěny reflexní značky k identifikaci a sledování příslušných anatomických bodů zpracovávaných triangulačním algoritmem k získání 3D polohy těchto bodů pohybujících se v zorném poli kamer. Udávaná výsledná přesnost je menší než 1 mm pro polohu jednotlivých značek, což se promítá do chyb v rozmezí 1–4 stupňů pro odhad úhlu kloubu podle konkrétní konfigurace. Výhodou této metody je vysoká přesnost a komplexní zachycení pohybu, využívající se převážně ve výzkumu pohybu člověka, například pro klinickou analýzu chůze, či při tvorbě počítačových animací například ve filmech. Hlavní nevýhodou tohoto systému jsou opět vysoké pořizovací náklady a složitost nastavení i používání. Současně je potřeba odporného personálu a vhodných prostorů, což vylučuje používání při běžné rehabilitaci.

Existuje řada takovýchto komerčních systémů, jako jsou například produkty Vicon Nexus (Vicon Motion Systems, Anglie), Smart DX (BTS Bioengineering, Itálie) a Optitrack Motive (NaturalPoint, USA). [4]



Obrázek 2: Kamerový systém Vicon při použití v medicíně (vlevo) [5] a při tvorbě animace (vpravo) [6].

Technologie založené na principu snímání pomocí kamery mají také své nízko nákladová řešení vhodná pro širší a každodenní použití. Tento systém je obvykle založen na spojení jedné klasické RGB kamery spolu s hloubkovým infračerveným senzorem a následným zpracováním vhodnými algoritmy, bez potřeby jakýchkoliv značek na snímané osobě. Právě zařízení s názvem Kinect (Microsoft) pracuje přesně na tomto principu. Původně bylo sice vyvinuto jako součást herní konzole Xbox, ale díky své uživatelské přívětivosti a relativně nízké ceně našlo své uplatnění nejen v pohybové rehabilitaci, ale i v dalších oblastech. Hlavní výhodou jsou jeho nízké pořizovací náklady a jednoduché použití například právě i při domácí rehabilitaci. Za nevýhodu by se dala považovat menší přesnost či omezenost dána nutností přímým postavením cvičící osoby před zařízení. Mezi produkty pracující na tomto principu můžeme zařadit DoctorKinetic (DoctorKinetic, Nizozemsko) a SilverFit (SilverFit, Nizozemsko). [4]

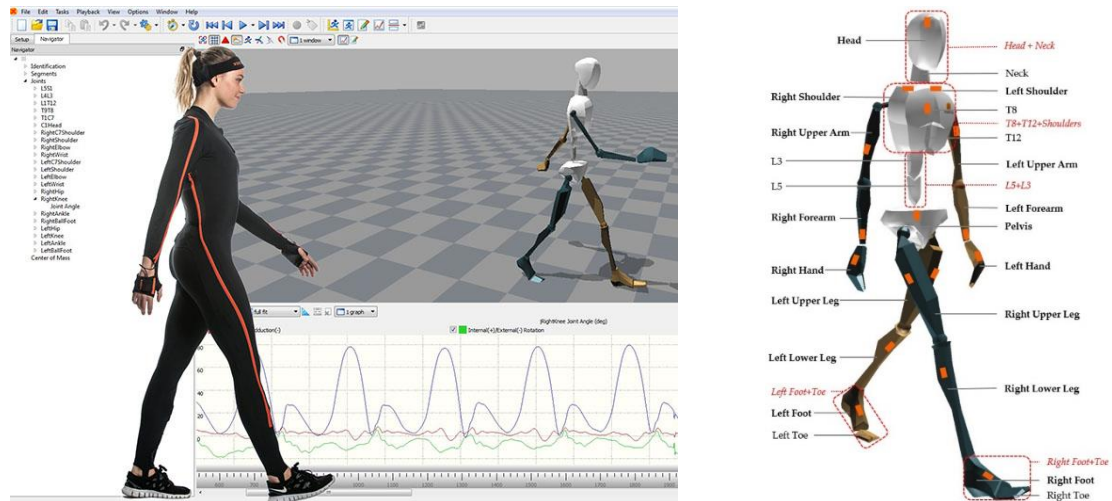


Obrázek 3: Příklad používání rehabilitačních produktů využívajících zařízení Kinect: DoctorKinetic (vlevo) [7] a SilverFit (vpravo) [8].

1.3 IMU senzory

Třetí technologie snímá pohyb částí těla pomocí vhodně umístěných tzv. IMU (MARG – Magnetic, Angular Rate, and Gravity) senzorů obsahující akcelerometr, gyroskop a případně i magnetometr tvořené technologií MEMS (Microelectromechanical systems). Data z většího množství těchto senzorů je nutné efektivně sbírat a dále pečlivě zpracovat fúzními algoritmy, aby bylo dosaženo potřebných výsledků a hlavně přesností. V biomechanice je nejpoužívanější odhad 3D orientace z jednotlivých senzorů a jeho následné použití pro odhad úhlů kloubů. Hlavní výhodou těchto IMU systémů je

kompletní a velmi přesný (srovnatelný s technologií na principu stereofotogrammetrie) záznam pohybů celého člověka. Za nevýhodu se opět dají považovat vcelku vysoké pořizovací náklady, tedy nevhodnost pro běžné použití k obecné rehabilitaci. Existuje několik komerčních řešení poskytující špičkové výsledky snímání pohybů celého těla pro klinickou analýzu či rovněž pro tvorbu počítačových animací, jako jsou MVN Biomech (Xsens Technologies, Nizozemsko), Opal (APDM Technologies, USA), Blue Trident (Vicon Motion Systems, Anglie) a další. [4]



Obrázek 4: IMU systém firmy Xsens tvořený oblekem a softwarem (vlevo) [9]. Rozmístění senzorů na jednotlivých částech těla (vpravo) [10].

Dále se zaměříme pouze na IMU systémy určené přímo k rehabilitaci, ať již větší komerční produkty či menší projekty vhodné dokonce i pro domácí rehabilitaci obsahující nejen menší množství, ale rovněž i méně přesných senzorů.

1.3.1 Armeo Senso

Armeo Senso je profesionální přístroj pro rehabilitaci horní končetiny od švýcarské firmy Hocoma se zastoupením po celém světě. Toto zařízení je tvořeno čtyřmi IMU senzory umístěnými kolem paže, zápěstí, hrudníku a poslední senzor drží cvičící v ruce. Snímač pohybu hrudníku umožňuje detekovat jeho nežádoucí pohyby, a ty kompenzovat například pomocí zpětné vazby o pohybu vyžadovaného pouze horní končetinou. Software k tomuto zařízení nabízí řadu interaktivních cvičení formou rehabilitačních her zobrazovaných na počítači. Kromě těchto motivujících her pro pacienty obsahuje software i nástroje pro tvorbu odborných reportů určených rehabilitačnímu pracovníku. [11]



Obrázek 5: Příklad použití rehabilitačního produktu Armeo Senso. [11]

1.3.2 Pablo

Pablo je kompletní profesionální sestava pro rehabilitaci nejen horních končetin od firmy Tyromotion se sídlem v Rakousku a působností po celém světě. Kromě dvou bezdrátově nabíjecích IMU senzorů pro měření rozsahu pohybů různých částí těla, je součástí této sestavy i speciální úchopový sensor a pohybové platformy (pomocné nástroje) Multiball a MultiBoard. K zařízení je samozřejmě k dispozici potřebný software opět nabízející rehabilitaci pro pacienty formou interaktivních her. Součástí softwaru je také objektivní hodnocení a záznamy podrobných reportů jednotlivých pacientů s příslušnými rozsahy pohybů horních končetin, měření různých sil úchopů či analýza chůze apod. [12]



Obrázek 6: Jednotlivé části rehabilitačního produktu Pablo (vlevo) a příklad jeho použití (vpravo). [12]

1.3.3 Bimeo Pro

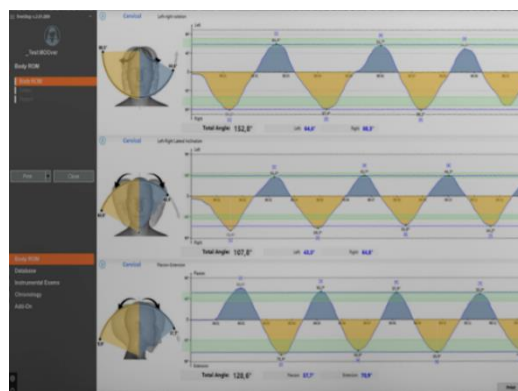
Podobný koncept zařízení pro rehabilitaci horní končetiny jako jsou dva dříve zmíněné nabízí i firma Kinestica se svým sídlem ve Slovinsku. Produkt Bimeo Pro má svoje počátky na univerzitě v Lublani a zatím je jediným produktem této firmy snažící se rozšířit i do mnoha dalších států. Součástí sestavy jsou dva IMU senzory, jeden se upíná kolem zápěstí a druhý kolem paže, jako třetí část je jakási multifunkční koule se senzory určená pro různé úchopy pacienta. Jako svoji přednost výrobce udává možnost cvičení s podporou zdravé končetiny. K dispozici je opět software pro PC s řadou rehabilitačních her pro pacienty spolu s potřebnými daty pro terapeuta sloužící například k monitorování postupu rehabilitace. [13]



Obrázek 7: Dva příklady použití rehabilitačního produktu Bimeo Pro. [13]

1.3.4 Moover

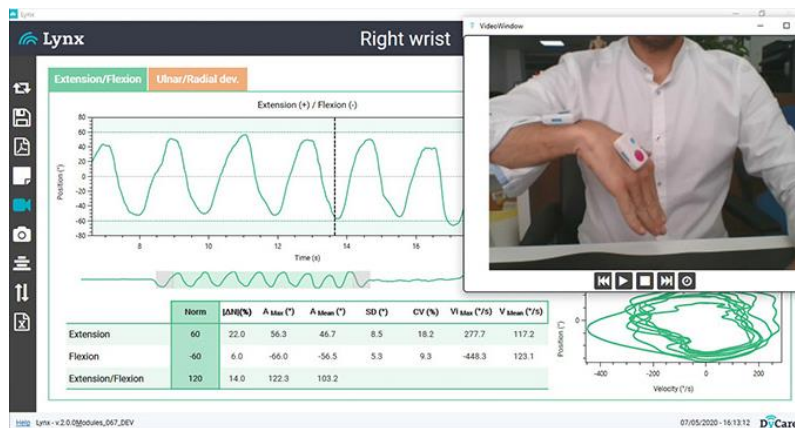
Moover je zařízení speciálně určené na měření rozsahu pohybů od Italské firmy Sensor Medica s centrálou i v USA a působností po celém světě, kde poskytuje mnoho dalších produktů pro analýzu biomedicínských dat. Zařízení pro měření rozsahu pohybu bývají často označována jako ROM (Range Of Motion). Moover je tvořen pouze jedním bezdrátovým IMU senzorem s potřebnými upínacími páskami pro libovolnou část těla. Průběhy měřených aktivních i pasivních rozsahů pohybu jsou v reálném čase zobrazovány ve stupních prostřednictvím softwaru na PC. Navíc je možnost zobrazit i zrychlení v daných úsecích pohybu, maximální dosažené úhly či porovnání s normálními hodnotami a následně uložení reportu pro příslušného pacienta. [14]



Obrázek 8: Princip použití produktu Moover (vlevo) [15] s ukázkou záznamu z obslužného programu (vpravo) [14].

1.3.5 Lynx

Lynx respektive Lynx ROM je produktem na měření rozsahu pohybů od Španělské firmy DyCare nabízející také verzi na měření síly Lynx DYNAMO a kompletní digitální rehabilitační platformu s názvem ReHub umožňující domácí rehabilitace se vzdálenou komunikací s terapeutem. Sestava Lynx ROM je tvořena dvěma až čtyřmi bezdrátovými IMU senzory umístěnými na jakékoli části těla nejen pomocí upínacích pásek, ale i pouhým přilepením senzoru na pacienta. Prostřednictvím softwaru na PC či na tablet (Windows) lze opět sledovat průběhy rozsahu pohybu a rychlosti v reálném čase s maximálními a průměrnými hodnotami. Software obsahuje i management pacientů, kde je možné sledovat postup rehabilitace a automaticky generovat příslušné reporty. [16]



Obrázek 9: Ukázka použití produktu Lynx ROM pro měření pohybu zápěstí s obslužným programem. [16]

1.3.6 Riablo a Kari

Riablo je produkt pro rehabilitaci téměř celého těla vyvíjený od roku 2013 firmou CoRehab v Itálii. Sestava je tvořena pěti bezdrátovými IMU senzory pro umístění na různé části těla pomocí upínacích pásek a balanční platformou se snímači tlaku pro specifická cvičení. Součástí je také software poskytující zpětnou vazbu cvičícímu přes obrazovku, buď formou jednoduchých her anebo základní vizualizací rozsahu pohybu. Terapeut má rovněž k dispozici knihovnu obsahující více než 350 cvičení pro různé části těla s možností individuálního nastavení pro každého cvičícího. Na závěr cvičení je možné automaticky vygenerovat detailní report. Existuje i varianta Riabla pro domácí použití. [17]



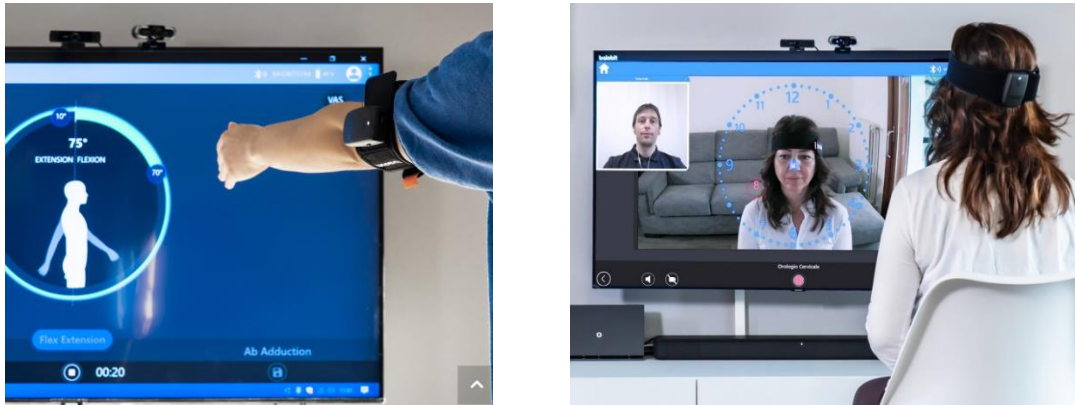
Obrázek 10 Příklad cvičení s produktem Riablo (vlevo) a ukázka obslužného programu (vpravo). [17]

Další produkt firmy CoRehab s názvem Kari je jakousi odlehčenou variantou Riabla určený však primárně pro domácí rehabilitaci. Kari je tvořen jedním IMU senzorem s několika upínacími páskami pro různé části těla a softwarem pro jakýkoliv mobil či tablet s Bluetooth a internetem. Na obrazovce cvičící vidí předvídající postavu spolu s vhodnou vizualizací svého pohybu v reálném čase. Terapeut může na dálku sledovat výsledky a kdykoliv se spojit s pacientem. [18]

1.3.7 BTS Telerehab

Produkt Telerehab určený pro domácí rehabilitaci, obdobně jako již zmíněná řešení ReHub a Kari, je vyvíjen firmou BTS Bioengineering zabývající se technologiemi pro analýzu pohybu člověka již od roku 1986, sídlící v Itálii, USA a s produkty po celém světě. Celkové zařízení se skládá z řídicí jednotky připojitelné k televizi, webkamery, air mouse ovladače a jednoho IMU senzoru s upínacími pásky.

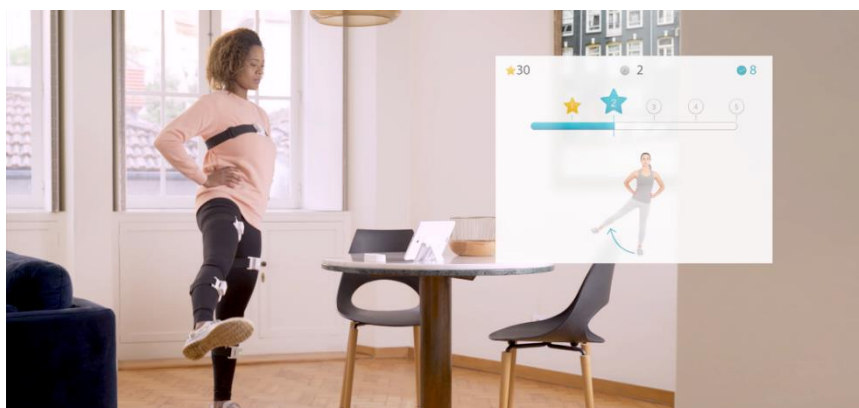
Rehabilitace může probíhat jakousi formou video hovoru s terapeutem, kde cvičící na televizi vidí rozsahy svých pohybů v reálném čase. Díky web kamerě umožňuje Telerehab i tzv. zrcadlovou terapii. Při té se pacient vidí na obrazovce spolu s překrývajícími se grafickými prvky vedoucího při cvičení. Samozřejmostí je také automatické generování reportů po cvičení s následnou možností online konzultace s lékařem. [19]



Obrázek 11: Příklady použití produktu Telerehab určeného pro vzdálenou rehabilitaci. [19]

1.3.8 Sword health

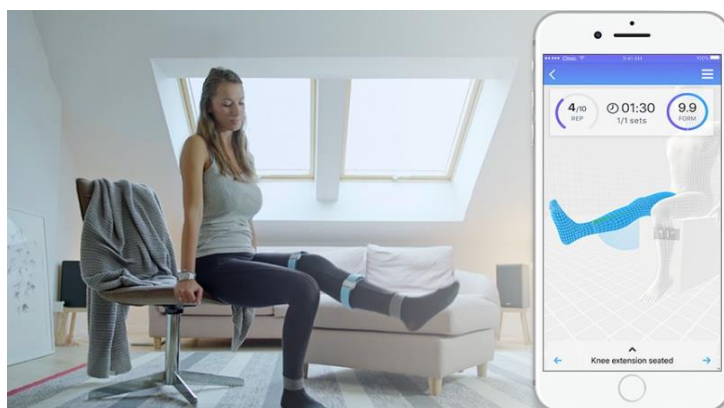
Sword health je další obdobnou inovativní platformou pro domácí rehabilitaci od stejnojmenné firmy z Portugalska s rozšiřující se působností i do dalších zemí. Součástí sestavy je až pět IMU senzorů pro libovolné části těla a tablet s potřebným softwarem. Výrobce vyzdvihuje použití umělé inteligence v aplikaci formou předcvičujících virtuálních postav a interaktivní komunikaci s pacientem. Rovněž jsou veškerá zaznamenaná data dálkově k dispozici terapeutovi. [20] [21]



Obrázek 12: Grafické znázornění využití produktu Sword health při domácím cvičení. [20]

1.3.9 ReFlex

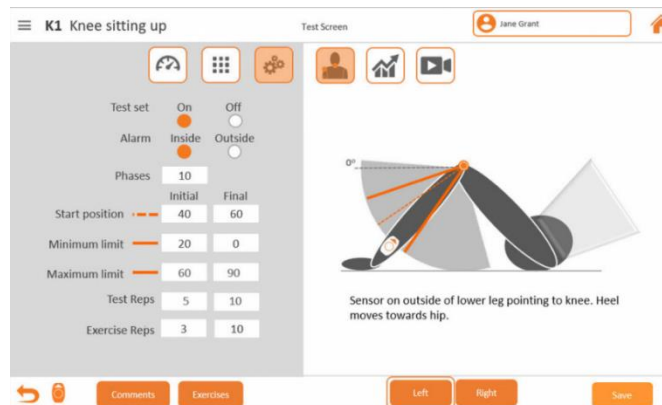
ReFlex je také relativně novým a cenově dostupným produktem pro nejen domácí rehabilitaci horní a dolní končetiny vyvíjený od roku 2015 v Rumunsku. K zaznamenání pohybu dané končetiny slouží dva IMU senzory. Obslužná aplikace je určena pro mobilní telefon jak se systémem Android, tak i iOS a obsahuje celou řadu předdefinovaných cvičení, režim pro pacienta i terapeuta. Jako zpětná vazba slouží cvičícímu 3D vizualizace aktuálního pohybu s měřenými daty v reálném čase. Opět nechybí možnost vzdáleného sledování výsledků jednotlivých cvičení terapeutem. [21] [22]



Obrázek 13: Grafické znázornění využití produktu ReFlex při domácím cvičení kolena. [21]

1.3.10 BPMpathway

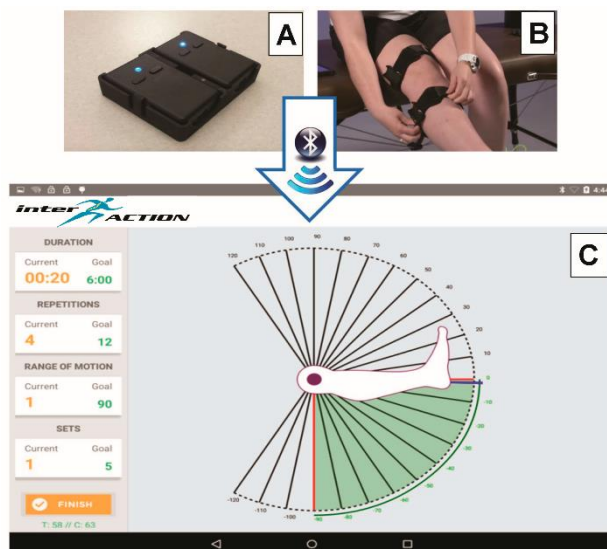
BPMpathway je jednoduchým a levným produktem z Anglie navrženým pro domácí rehabilitaci. K měření rozsahu pohybu končetiny slouží pouze jediný bezdrátový IMU senzor s názvem BPMpro. Měřená data jsou přenášena do aplikace vhodné pro libovolný tablet či telefon se systémy Android nebo iOS. Pacient vidí v reálném čase rozsahy svých pohybů formou animovaného avataru na obrazovce. Aplikace disponuje intuitivním rozhraním s možností označit bolestivé pasáže cvičení a sledovat progres rehabilitace a tím motivovat pacienta. Terapeut může rovněž na dálku nastavovat požadované limity rozsahů pohybu, sledovat výsledky a komunikovat s pacientem. [23] [21]



Obrázek 14: Ukázka z programu sloužícího pro cvičení se senzorem produktu BPMpathway. [23]

1.3.11 InterACTION

InterACTION je název aplikace, jako součást projektu a studie určené k rehabilitaci kolena na univerzitě v Pittsburghu (USA). K měření úhlu kolenního kloubu jsou použity dva bezdrátové IMU senzory připevněné pomocí elastických pásek na stehno a holeň. Aplikace je určena pro mobilní telefon a v reálném čase zobrazuje pacientovi rozsah pohybu pomocí dvourozměrné animace a dosaženého úhlu. Aplikace obsahuje několik předdefinovaných cvičení s podrobným popisem a videoukážkou. Součástí projektu je také webový portál synchronizovaný s aplikací pro interakci s terapeutem. [24] [21]



Obrázek 15: Grafické znázornění použití zařízení InterAction k měření úhlu ohybu kolenního kloubu. [24]

1.4 Souhrn

Na první pohled existuje celkem mnoho nástrojů nebo přímo celých systémů pro nějaký typ měření pohybu. Mnoho jich je využíváno i při různých pohybových rehabilitacích. Tyto existující nástroje / systémy se dělí v podstatě pouze do dvou skupin:

- A. Jednoduché, avšak uzavřené, určené zejména pro komerční využití v domácím, případně i lékařském prostředí. Jsou sice cenově dostupné, ale neposkytují tzv. surová data, a tedy neumožňují jejich využití pro experimenty, při výuce nebo dokonce při vývoji vlastních algoritmů, případně komplexnějších rehabilitačních postupů. Tyto nástroje jsou v naprosté většině případů určeny pro měření / rehabilitaci pouze předem určené části těla, což je dáno jejich obslužnou aplikací.
- B. Velmi sofistikované měřicí systémy, složené dokonce z mnoha částí a poskytující velmi přesná data. Jejich pořizovací cena je však mimo možnosti obecných lékařských pracovišť. Současně jejich výstupy (data) jsou příliš komplexní pro použití při základních experimentech nebo výuce. Tyto systémy lze použít k měření pohybu jakékoli části těla, ale jejich složitost a cena je činí v podstatě v mnoha případech nedostupnými.

Z uvedeného vyplývá absence jednoduchého, avšak univerzálního zařízení (případně systému) využitelného pro měření základních částí pohybového aparátu těla / člověka s možností využití zejména surových výstupních dat (ze senzorů) za účelem vědeckých, medicínských, a i matematických experimentů. Při vývoji zejména medicínských postupů nebo celých zařízení je potřeba mít i několik takových to zařízení (systémů) k dispozici, a to jak ve školním prostředí, tak současně i v cílovém medicínském oboru (pro pacienty). Rovněž je potřeba mít přístup ke všem zejména surovým měřeným datům. V průběhu vývoje totiž není od samého začátku zřejmé, jaká měřená data budou skutečně nakonec potřeba a jaké postupy (například měřicí) jsou nejvhodnější. Z tohoto důvodu je vhodné toto experimentálně ověřovat na několika zařízeních / pracovištích (tedy i pacientech) současně. Z tohoto důvodu jsou tedy i velmi přesná zařízení, ale cenově nákladná dosti nevhodná. Nelze je pořídit v několika kusech. Rovněž i levná zařízení, kde však nelze získat dostatečný přístup ke všem měřeným datům, jsou pro tyto účely taktéž většinou nepoužitelná.

Celková tvorba takových to zařízení / systémů však není samozřejmě nikterak jednoduchá. Ty jsou totiž složeny z mnoha dílčích částí jako například: pohybový senzor, přenos dat, algoritmy na detekci pohybu / polohy, 3D zobrazení, vizualizace a samozřejmě ukládání pořízených dat.

Ze zde popsaných důvodů bylo přistoupeno k tvorbě čistě experimentálního zařízení (tedy nikoli celého systému) vhodného zejména pro výukové, vývojové a experimentální použití jak ve školním, tak i medicínském prostředí. Jednotlivé části jsou dostatečně nezávislé a otevřené. Rovněž je poskytnut přístup ke všem měřeným (surovým) datům. Lze použít téměř libovolné pohybové senzory a lze je rovněž i kalibrovat. Budou využity volně dostupné algoritmy pro vyhodnocení pohybu / polohy s možností jejich rozšíření / doplnění podle budoucí vlastní potřeby. Jednou z velmi složitých částí je dobrá vizualizace poskytující vhodné zobrazení například formou 3D postavy člověka. Dále prezentované zařízení si samozřejmě v žádném případě neklade za cíl konkurovat již existujícím komerčním projektům. Jeho cílem je hlavně poskytnout základní nástroj / možnosti pro vlastní experimenty v oblasti základního měření pohybu.

2 Zadání práce

2.1 Současný stav

První bod zadání byl již zmíněn v úvodní části práce s názvem Současný stav.

2.2 Aplikace

Vyvíjená aplikace pro PC (MS Windows) bude určena zejména pro experimenty při pohybové diagnostice a rehabilitaci. Nebude se však jednat o plně profesionální, a tedy zcela dokončenou aplikaci navrženou přímo a pouze pro rehabilitaci, ale spíše o experimentální / vývojové prostředí vhodné jako základ pro další projekty v tomto oboru. Součástí aplikace bude rovněž mnoho poskytnutých ladících / vývojových nástrojů a vizualizačních prvků vhodných nejen pro účely rehabilitace, ale i částečné diagnostiky. Hlavním měřitelným parametrem při rehabilitaci budou dosažené úhly pohybu dílčích částí těla spolu s vhodnou vizualizací formou 3D postavy.

2.3 Dva senzory

Pro měření pohybu, respektive orientace v prostoru budou využity dva senzory. První variantu bude tvořit přesnější senzor, respektive senzorový modul tvořený dostupným IMU/MARG senzorem doplněný ARM (Advanced RISC Machines) procesorem a komunikací (v současné době) přes USB, případně Bluetooth. Tento přesnější senzorový modul by měl dosahovat preciznějších výsledků než druhý zmíněný použitý senzor. Druhý, avšak podle očekávání méně přesný, bude vytvořen pomocí běžného chytrého mobilního telefonu s potřebnou aplikací a přenášející data přes Bluetooth do počítače. Tato varianta by měla být v dnešní době jednoduše dostupná téměř pro každého.

2.4 Algoritmy

Pro dostatečně přesné detekce pohybu, respektive stanovení orientace bude využito již existujících algoritmů pro zpracování dat z pohybových senzorů. Přesněji řečeno, budou použity algoritmy na principu komplementárního filtru jako je například tzv. Mahony filtr, tzv. Madgwick filtr a některé další implementace tohoto typu. Není tedy nutné vymýšlet zcela nové algoritmy a ani to není náplní této práce. Tyto algoritmy jsou veřejně dostupné, již dostatečně ověřené a pro zde zamýšlené účely zcela postačující. Algoritmů implementujeme několik a případně uvedeme jejich výhody a nevýhody.

2.5 Demonstrace výsledku

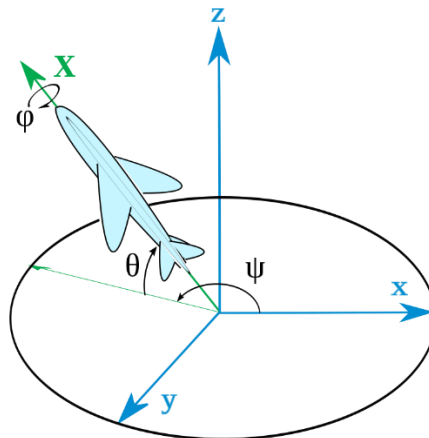
V poslední části práce se zaměříme na možné způsoby a postupy využití zde navrhovaného řešení / zařízení k objektivizaci základních dílčích pohybů člověka. Pokusíme se inspirovat některými známými lékařskými postupy pro jednoduchou rehabilitaci.

3 Měření orientace v prostoru

Abychom mohli objektivně vyhodnotit a popsat, jak je daný objekt orientován (natočen) ve 3D prostoru, musíme použít nějakou matematickou reprezentaci této orientace. K určení orientace musíme znát počáteční stav nebo mít stanoven nějaký referenční bod (souřadnicový systém), vůči kterému orientaci uvádíme. Existuje několik možností popisující orientaci ve 3D, například Eulerovy úhly, kvaternion, rotační matice či popis pomocí osy rotace a úhlu otočení. Všechny zmíněné metody mají své výhody i nevýhody zahrnující rozdílné počty parametrů, možnosti skládání rotací a počítání s nimi, či jejich vhodnost pro snadnou čitelnost. V této práci používáme pro popis orientace Eulerovy úhly a kvaterniony, a proto se v dalších odstavcích stručně zaměříme na jejich definice a vlastnosti.

3.1 Eulerovy úhly

Reprezentace rotace Eulerovými úhly vyžaduje znát pouze tři parametry (úhly) ψ , θ a φ . Existuje řada možností, jak tyto tři rotace definovat. Jednotlivé rotace mohou být buď kolem původního nehybného souřadnicového systému (tzv. vnější rotace), anebo kolem os rotujícího systému měnícího svoji orientaci po každém otočení v dané ose (tzv. vnitřní rotace). Další variabilitou může být rozdílné pořadí vykonání těchto tří rotací. V našem případě používáme tzv. aerospace sekvenci [25] (vnitřní rotace), kde úhel ψ (Yaw) reprezentuje otočení okolo původní osy z , úhel θ (Pitch) okolo nové osy y' a úhel φ (Roll) okolo nové osy x'' . [26]



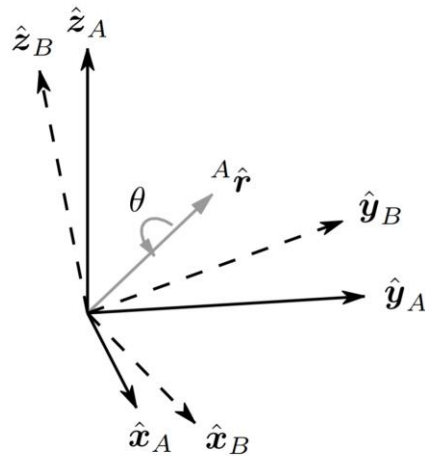
Obrázek 16: Grafické znázornění Eulerových úhlů v tzv. aerospace sekvenci. [26]

Základním problémem Eulerových úhlů je nejednoznačná reprezentace výsledné orientace. Jinak řečeno k jedné výsledné orientaci můžeme dojít několika kombinacemi těchto tří dílčích úhlů. Nastávají také tzv. singulární situace, toto činí problém zejména při reprezentaci spojitého otočení. Problém tzv. „gimbal lock“ nastává tehdy, když splynou dvě rotační osy dohromady a tím pádem se není možné přímým otočením dostat to požadované pozice [27]. Tento jev nastává tehdy, pokud se úhel θ dostane na hodnotu $\pm 90^\circ$. [26]

3.2 Kvaterniony

Kvaternion je čtyř-dimenzionální komplexní číslo používané mimo jiné k popisu orientace ve 3D prostoru. Umožňuje popsat rotaci souřadnicového systému B vzhledem k souřadnicovému systému A znázorněných na Obrázek 17 a to pomocí úhlu otočení θ a osy otáčení ${}^A_B \mathbf{r}$ vyjádřené v souřadnicích systému A . Kvaternion popisující tuto rotaci značíme ${}^A_B \mathbf{q}$ a je definován v rovnici (1), kde r_x , r_y a r_z jsou složky jednotkového vektoru v souřadnicích A . U používaného značení levý spodní index určuje, co rotuje, a horní levý index značí referenci, tedy vůči čemu rotujeme. Je obvyklé, aby kvaterniony

popisující rotaci byly jednotkové velikosti. Výpočty s rotačními kvaterniony totiž často vyžadují jejich prvotní normalizaci. [28]



Obrázek 17: Grafické znázornění kvaternionu popisující rotaci souřadnicového systému B vzhledem k A . [28]

$${}^A_B \mathbf{q} = [q_1 \quad q_2 \quad q_3 \quad q_4] = \left[\cos \frac{\theta}{2} \quad -r_x \sin \frac{\theta}{2} \quad -r_y \sin \frac{\theta}{2} \quad -r_z \sin \frac{\theta}{2} \right] \quad (1)$$

Hlavní výhodou popisu rotace pomocí kvaternionů je jejich spojitá reprezentace otočení ve všech směrech. Nenastávají zde žádné singulární případy, jako je tomu u Eulerových úhlů [28]. Další výhodou je jednoduché skládání rotací pomocí násobení kvaternionů. Za nevýhodu by se dala považovat jejich ne zrovna dobrá čitelnost pro člověka. Pokud tedy potřebujeme znát (lépe čitelné) úhly natočení je možné kvaternion převést na vyjádření v Eulerových úhlech. Tento přepočít je zapsán v rovnicích (2), (3) a (4) [28].

$$\psi = -\text{atan2}(2q_2q_3 - 2q_1q_4, 2q_1^2 + 2q_2^2 - 1) \quad (2)$$

$$\theta = \sin^{-1}(2q_2q_4 + 2q_1q_3) \quad (3)$$

$$\varphi = -\text{atan2}(2q_3q_4 - 2q_1q_2, 2q_1^2 + 2q_4^2 - 1) \quad (4)$$

4 Návrh řešení

V této kapitole podrobněji popíšeme celkové řešení našeho zařízení / systému sloužícího primárně pro objektivní měření a současně zobrazení rozsahů pohybů dílčích částí těla člověka. Systém rovněž obsahuje mnoho vývojových nastavení, možnosti zobrazení přímo surových dat ze senzorů, kalibraci senzorů a další nástroje vhodné pro vlastní experimenty a ladění. Celkové zařízení je tvořeno dvěma typy pohybových senzorů připevňující se na požadovanou část lidského těla a dál ovládací / zobrazovací aplikací určenou pro PC (MS Windows). První zde zmíněný typ senzoru (senzorový modul) je tvořen vlastním hardwarem obsahující IMU/MARG senzor a řídicí mikrokontroler (ARM procesor). Způsob přenosu dat do PC z tohoto senzoru je řešen pomocí USB. Druhý typ použitého senzoru představuje běžný chytrý mobilní telefon komunikující s PC přes Bluetooth. Naměřená surová data ze senzoru (v obou zmíněných případech) jsou v reálném čase vhodně zpracovávána na PC pomocí existujících / převzatých algoritmů k získání absolutní orientace v prostoru. Následně je tato znalost orientace senzoru prezentována v aplikaci formou dosaženého úhlu natočení příslušné části 3D postavy, na které je právě fyzicky senzor umístěn.

Používané pojmy:

- **Senzorový modul** – Celé kompletní zařízení / přístroj tvořené např. dílčími senzory, řídicí jednotkou (ARM procesorem), komunikačním rozhraním apod.
- **Senzor** – Obecné zařízení / součástka měřící i několik fyzikálních veličin a poskytující vhodný výstup (např. vlastní HW senzor).
- **Dílčí senzor** – Jeden elementární senzor v jednom pouzdře / obvodu měřící jednu konkrétní fyzikální veličinu: např. akcelerometr, gyroskop a magnetometr.
- **Aplikace na PC** – Aplikace / program, například pro MS Windows vytvořený jako součást této práce.
- **Surová data** – Data čtená přímo z dílčích senzorů (bez jakékoli úpravy) obsahující šum, odchylky, chyby apod. nebo data ještě nekalibrovaná.
- **Kalibrovaná data** – Měřená data případně upravena např. přičtením či vynásobením jistou (kalibrační) konstantou, tak aby přesněji odpovídala reálnému stavu.
- **Algoritmus / filtr** – Postup na zpracování kalibrovaných dat ze senzorů, zahrnující jejich přepočítání na jinou fyz. veličinu (např. úhel natočení) a případnou filtraci.

4.1 Senzor polohy / orientace

Abychom získali úhel natočení / pozici požadované části lidského těla potřebujeme metodu, jak požadované fyzikální parametry nejlépe elektronicky změřit a převést je do digitální / číslicové formy. K tomuto nám právě poslouží nějaký senzor měřící vhodné fyzikální veličiny pro stanovení polohy / orientace. Hlavním požadavkem na volbu senzoru je zejména jeho cena a dostupnost při požadované přesnosti. Snímání pohybu člověka lze tedy řešit mnoha způsoby (jak je uvedeno v úvodní části této práce), již zmíněnými inerciálními senzory, rovněž i mechanickým způsobem, nebo například přímo zpracováním obrazu snímaného kamerou. Použití senzoru na mechanickém principu je složité zejména pro jeho vlastní mechanickou konstrukci, zbytečnou robustnost a často i velkou hmotnost. Navíc je i cenově nákladný, tedy nevhodný pro naše použití. Pro přesnou detekci kamerou by byla nejspíše potřeba soustava kamer, což by bylo rovněž velmi nákladné, složité na zpracování a celkově nevhodné pro naši aplikaci. Proto jsme zvolili inerciální senzory (IMU) tvořené obvykle akcelerometrem,

gyroskopem a případně i magnetometrem. Tyto dílčí senzory jsou tvořeny technologií MEMS a jsou v dnešní době velmi rozšířené, a to zejména díky jejich malým rozměrům (jeden čip) a zejména nízké ceně.

4.1.1 Senzorový modul s MPU9250/9255

Na trhu existuje opravdu řada dostupných inerciálních senzorů od různých výrobců lišících se v množství obsažených dílčích senzorů (typy senzorů a počet DOF – Degree of Freedom), dále v ceně a samozřejmě v jejich parametrech, tedy zejména přesnosti. Naším hlavním požadavkem je, aby obvod v jednom pouzdře obsahoval (pro jednoduchost) všechny tři požadované typy senzorů: akcelerometr, gyroskop a magnetometr. Tyto tři dílčí senzory jsou potřeba pro kompletní určení orientaci ve 3D prostoru. Jako inerciální sensor jsme zvolili MPU9250 / MPU9255 od firmy TDK InvenSense. Tento sensor obsahuje všechny tři dílčí senzory přímo v jednom pouzdře (obvodu) a je rovněž velmi rozšířen. Na E-Bay jej lze zakoupit na vhodné vývojové destičce za 100–150 Kč a v českých e-shopech tuto destičku pořídíme od 160 Kč. Jako alternativa by se dala použít destička s čipy LSM303DLHC a L3GD20 od firmy STMicroelectronics mající zvlášť v pouzdře akcelerometr s magnetometrem a zvlášť gyroskop. Za jako poněkud dražší (avšak přesnější) alternativu by se dal také považovat čip BNO055 od firmy Bosh.

Stručné parametry MPU9250/55 jsou následující:

Gyroskop parametry:

- Tři osy (X, Y, Z) MEMS technologie, 16 bit AD převodník
- Čtyři nastavitelné rozsahy měření ± 2 g, ± 4 g, ± 8 g a ± 16 g (pro 16bit výstup)
- Frekvence měření až 8 kHz

Akcelerometr parametry:

- Tři osy (X, Y, Z) MEMS technologie, 16 bit AD převodník
- Čtyři nastavitelné rozsahy měření ± 250 , ± 500 , ± 1000 , a ± 2000 °/sec (pro 16bit výstup)
- Frekvence měření až 4 kHz, programovatelný filtr dolní propust

Magnetometr parametry:

- Tři osy (X, Y, Z) MEMS technologie, 14 bit AD převodník
- Rozsah měření ± 4800 μ T (pro 16bit výstup)
- Frekvence měření 8 Hz až 100 Hz

Pro komunikaci s čipem MPU9250 a následný přenos dat jsme zvolili ARM mikrokontroler typu STM32 od firmy ST-Microelectronics. Tyto procesory jsou relativně levné, dobře dostupné a velmi rozšířené ve školách / experimentech. Jelikož je náš sensor experimentální / vývojový, zvolil se mikrokontroler umístěný na vývojové desce NUCLEO64/32 obsahující současně programátor / debugger a další potřebné vývody. Výhodou je i zdarma dostupné vývojové prostředí pro PC s názvem STM32CubeIDE.

4.1.2 Senzor tvořen mobilním telefonem

Jelikož v dnešní době již téměř každý vlastní nějaký chytrý mobilní telefon a ty v sobě obsahují inerciální sensor (například pro detekci otočení displeje), rozhodli jsme se jako druhou variantu pohybového senzoru využít právě mobilní telefon. Důležité je však poznamenat, že ne každý telefon disponuje všemi třemi typy dílčích senzorů: akcelerometr, gyroskop a magnetometr. Dalším problémem může být absence třetí osy jednoho ze senzorů (méně DOF). V této práci jsme se rovněž zaměřili pouze na mobilní telefony se systémem Google / Android pro jejich snadný vývoj aplikací.

Systém Google / Android poskytuje následující výstupní hodnoty ze senzorů:

- Akcelerometr – m/s^2 (zrychlení)
- Gyroskop – rad/s (úhlová rychlost)
- Magnetometr – μT (magnetická intenzita)

4.2 Přenos dat do počítače

Naměřená data ze senzorů je samozřejmě potřeba přenést do PC na jejich další zpracování a reprezentaci.

Při vývoji první přesnější verze měřicího zařízení probíhá komunikace s PC pomocí USB kabelu (data + napájení). Rozhraní USB je vytvořeno přímo v procesoru ARM. USB rozhraní může být rovněž využito pro skutečně velmi rychlé vzorkování / přenos dat (pokud by bylo potřeba).

Nabízela by se také možnost použití bezdrátové komunikace prostřednictvím Bluetooth či Wifi. Pro bezdrátovou komunikaci s přesným senzorem lze tedy rovněž využít rozhraní Bluetooth pro jeho malou energetickou náročnost a zcela dostatečný dosah (5-10 m). Naopak použití Wifi je vzhledem k velkému dosahu (i 100 m a více) a celkem velké energetické náročnosti poněkud nevhodné.

Pro přenos dat z mobilního telefonu je primárně využita technologie Bluetooth. Pokud souhrnně srovnáme Bluetooth a Wifi z našeho pohledu, dostaneme následující pro a proti:

Bluetooth:

- + Obsažený v každém mobilním telefonu
- + Energeticky velmi málo náročný (automatické nastavení výkonu podle nutného dosahu)
- + Možnost přímého spojení mobilního telefonu a počítače
- + Snadná ovladatelnost z vlastní aplikace
- + Celé spojení lze vytvořit / zajistit v podstatě pouze programovou cestou
- Omezený dosah na několik metrů (do 10 m, není však překážkou)
- Omezená rychlost přenosu dat na maximálně zhruba 1Mbit pro BT Classic (BT LowEnergy méně)
- Nutný Bluetooth-USB adaptér do PC (v notebooku skoro vždy obsaženo)

Wifi:

- + Obsažený v každém mobilním telefonu
- + Velmi rychlý přenos dat (xxMBit)
- Proti Bluetooth mnohem větší energetická náročnost
- Ne zcela snadný přenos dat z mobilního telefonu do počítače (jedna z těchto možností)
 - A) Nutnost dosahu Wifi-Point/Router pro přenos dat
 - B) Vytvoření Wifi-HotSpot na počítači (počítače s přístupem na internet, jinak velmi složité)
 - C) Vytvoření Wifi-Direct (ne všechny Wifi adaptéry toto podporují, složité programování)
- Nutný Wifi-USB adaptér do PC (v notebooku skoro vždy obsaženo)
- Při spojení je velmi často potřeba nějaké nastavení (adresy, hesla, ...)

Vzhledem k jednoduchosti a celkově snadného přímého spojení pomocí Bluetooth mezi nejen mobilním telefonem a cílovým počítačem pro nás na dostatečnou vzdálenost a s vyhovující přenosovou rychlostí, byla zvolena právě tato technologie.

4.3 Formát přenosu dat

Dílní senzory (akcelerometr, gyroskop a magnetometr) obsažené v jednom kompletním IMU poskytují surová data, každý z nich ve třech osách (X, Y, Z). Jednou možností je tyto data prostřednictvím ARM procesoru či mobilu pouze vyčítat a zasílat přímo do PC nijak nezpracovaná (tedy surové). V tomto případě je nutné posílat celkem velké množství dat: tři senzory, každý má tři osy a v každé ose jsou 16bitové hodnoty, tedy celkem 9krát 16 bitů (celkem 18 B na jeden záznam). Výhodou této možnosti je netřeba v podstatě žádného výpočetního výkonu v ARM procesoru či mobilu. Přijatá data jsou pak dále zpracovávána až v PC, což je výhodné pro vývoj a testování vlastních vytvářených algoritmů. Je nutno zmínit rovněž celkový objem přenášených dat. Například pro minimální rychlost zhruba 100 vzorků za vteřinu jde o 18 B x 100 tedy zhruba 2 kB za vteřinu, což je na dnešní dobu v podstatě stále velmi malý objem dat.

Druhou možností je zpracování surových dat přímo v ARM procesoru či mobilu a zasílat do PC již (před)zpracovaná, případně výsledná data. Těmito výslednými daty rozumíme ideálně stanovenou výslednou absolutní orientaci senzoru v prostoru ve formě kvaternionů či Eulerových úhlů. Výhodou tohoto způsobu je menší datový tok do PC, tedy minimální nároky na přenos dat nebo snadné zasílání informací paralelně z více senzorů současně. ARM procesor musí na rozdíl od předchozího způsobu disponovat dostatečným výpočetním výkonem, aby na něm mohli běžet algoritmy počítající absolutní orientaci v reálném čase.

První způsob přenosu dat je vhodný pro vývoj a ladění a druhý způsob je vhodný pro již finální odladěnou aplikaci / zařízení.

Aby bylo možné výše zmíněná data přenášet z pohybových senzorů (vlastní sensorový modul či mobil) do počítače (využitím USB / Bluetooth) a správně je číst, je potřeba navrhnout nějaký vhodný komunikační protokol / formát. Máme dvě základní možnosti implementace komunikace. Buď můžeme použít textovou komunikaci. Ta je snadno čitelná, ale je poněkud náročnější na množství přenášených dat. Je tedy zejména vhodná při vývoji systému. Druhou variantou je binární komunikace mající přesně opačné vlastnosti. Tedy je ideální na přenos velkého množství dat, ale je poněkud nečitelná.

4.4 Komunikace – formát dat

Rozlišujeme dvě fáze přenosů (měřených) dat. První fází je přenos dat z MPU9250/9255 do STM32 po lokální sběrnici SPI či I²C vytvořený podle dokumentace výrobce. Tyto sběrnice jsou však určeny pro komunikaci na vcelku krátké vzdálenosti (i jednotky centimetrů, podle použité rychlosti). Získaná data jsou pak ve druhé fázi přenosu zasílána z STM32 do počítače pomocí například USB / Bluetooth i na vzdálenost několika metrů. Obdobně je tomu také při využití mobilního telefonu jako senzoru. Zde však rozlišujeme pouze druhou zmíněnou fázi, tedy přenos rovnou z mobilního telefonu do počítače pomocí Bluetooth (interní přenos dat ze senzoru v mobilním telefonu do jeho procesoru neuvažujeme).

Pro druhou fázi přenosu dat (do PC) byly navrženy dva obecnější komunikační protokoly: textový a binární.

4.4.1 Textový protokol

Textový protokol je využíván pro vysílání / přijímání paketů z pohybového senzoru tvořeného mobilním telefonem do počítačem (Bluetooth). Navržený tvar jednotlivých paketů je následující:

Vysílaný paket (úvodní a koncový znak je dodán při vyslání):

- [CMD C=1 D=AGM S=100]
- CMD – znamená povel
- C – typ povelu (0 = Stop / 1 = Start)
- D – jaká data se požadují, seznam písmen (A = Accel / G = Gyro / M = Mag)
- S – požadovaná měřicí / snímací rychlost data v Hz (20 až ...)

Přijímaný paket (úvodní a koncový znak je dodán při vyslání):

- [DATA A=X.X;Y.Y;Z.Z G=X.X;Y.Y;Z.Z M=X.X;Y.Y;Z.Z]
- DATA – znamená paket obsahující data
- A – data z Accel (tři složky X,Y,Z), jednotky „g“
- G – data z Gyro (tři složky X,Y,Z), jednotky „stupeň/s“
- M – data z Mag (tři složky X,Y,Z), jednotky „ μT “

(jako desetinný oddělovač je použita „tečka“)

4.4.2 Binární protokol

Binární protokol je použit při komunikaci pomocí USART/USB mezi vlastním sensorovým modulem a počítačem.

Pro optimální přenos dat je samozřejmě nejvhodnější plně binární protokol. U tohoto typu přenosu je však problém, jak správně detekovat začátek paketu a jeho konec. Aby toto bylo možné, je nutné použít například unikátní hodnoty (případně posloupnost hodnot) nikde jinde v paketu / datech se nevyskytující.

Nejjednodušším způsobem, jak tohoto docílit, je všechny hodnoty přenášet například jako 7bitové a nejvyšší bit7 v každém byte nechat pro vhodnou signalizaci. V případě, že se jedná o řídicí byte je bit7 = 1, naopak v případě dat je bit7 = 0.

Datové hodnoty z akcelerometru, gyroskopu či magnetometru jsou většinou 12-16 bit čísla. Tato čísla budeme přenášet jako dva za sebou jdoucí byty. První bude mít vždy bit7 = 0 (tedy maximální hodnota 0111111b) a druhý byte bude již libovolné číslo (tedy maximálně 1111111b). Přenášené hodnoty budou tedy v podstatě 15bitové (upravené na 15bitů). Abychom rozlišili řídicí a datové byty, musíme tedy zajistit, aby řídicí byty byly vždy dva po sobě a oba měly bit7 = 1 (takováto posloupnost nebude nikdy v datových hodnotách). Navíc musíme ještě jako řídicí byty detekovat pouze poslední dva byty mající bit7 = 1. Poněvadž před prvním řídicím bytem může být druhá část (nižší) datové hodnoty taktéž s bit7 = 1.

Jelikož nejsme schopni snadno detekovat konec paketu, přidáme do prvního řídicího bytu 7 bitovou hodnotu určující délku paketu. Abychom zajistili správnost přenášených dat, využijeme druhý řídicí byte k uložení kontrolního součtu tohoto paketu. Výsledný tvar protokolu např. pro data pouze z akcelerometru vypadá následovně (první se přenáší vyšší byte a poté nižší byte):

- Řídicí 16bits hodnota: 100001101ZZZZZZb (0000110b = délka paketu)
- Datová 16bits hodnota AccelX (15bitů skutečných dat): 0XXXXXXXXXXXXXXXXXb
- Datová 16bits hodnota AccelY (15bitů skutečných dat): 0YYYYYYYYYYYYYYYYb

- Datová 16bits hodnota AccelZ (15bitů skutečných dat): 0ZZZZZZZZZZZZZZZb
- 10000110b vyšší část řídicí 16bits hodnoty obsahující délku 6 bytes
- 1ZZZZZZb nižší část řídicí 16bits hodnoty obsahující kontrolní součet ZZZZZZ
- 0XXXXXXXb vyšší část datové 16bits hodnoty AccelX (avšak vždy s bit15=0)
- XXXXXXXb nižší část datové 16bits hodnoty AccelX
- 0YYYYYYYb vyšší část datové 16bits hodnoty AccelY (avšak vždy s bit15=0)
- YYYYYYYb nižší část datové 16bits hodnoty AccelY
- 0ZZZZZZb vyšší část datové 16bits hodnoty AccelZ (avšak vždy s bit15=0)
- ZZZZZZZb nižší část datové 16bits hodnoty AccelZ

4.5 Algoritmy

Pod pojmem „algoritmy“ v této práci rozumíme postupy pro zpracování surových dat (zrychlení, úhlová rychlost a magnetická intenzita) z dílčích senzorů a následné získání absolutní orientace v prostoru ve formě kvaternionu či Eulerových úhlů. Tyto algoritmy bývají často označovány jako AHRS (Attitude and Heading Reference Systems) algoritmy či orientační filtry. Úkolem těchto algoritmů, respektive filtrů není pouze přímý výpočet orientace, ale právě i potřebná filtrace surových dat zatížených různými chybami a zejména šumem.

Zpracování dat pouze z IMU (akcelerometr + gyroskop) senzoru nám neumožní získat absolutní orientaci ve 3D prostoru, ale jen orientaci vzhledem k zemi díky měřené gravitační referenci [28]. Tato informace je často postačující pro většinu běžných použití, kde je potřeba znát nějakou základní orientaci (spíše pouze náklon a sklon), ale není zcela dostačující pro plnou činnost zde uvažované aplikace. Algoritmus pro zjištění této základní orientace je relativně snadný a vyžaduje tedy data pouze z akcelerometru. Pro lepší výsledky je však nutné přidat prostřednictvím vhodného filtru i data z gyroskopu. Některé tyto algoritmy byly pro jejich jednoduchost implementovány i v této práci pod označením: „Dirty“ komplementární filtr a komplementární filtr s horní / dolní propustí.

V případě použití MARG (akcelerometr + gyroskop + magnetometr) senzoru jsme již schopni vypočítat absolutní orientaci ve 3D prostoru díky gravitační a magnetické referenci [28]. Algoritmy pro zjištění této absolutní pozice využívající data ze všech tří senzorů a umožňují kompenzaci magnetického zkreslení a odchylky gyroskopu [28]. Mohou však být již poněkud složité. Proto jsme některé volně dostupné implementace těchto filtrů převzali již jako hotové. Jelikož náš používaný senzor MPU9250/55 obsahuje všechny tři dílčí senzory a určení absolutní orientace ve 3D je nutné pro plnou činnost naší aplikace, budeme tedy zejména využívat algoritmy pro MARG.

Existuje mnoho algoritmů volně dostupných na internetu ať už méně či více odborných / přesných / jednoduchých. Není tedy primárně nutné vymýšlet nové. Algoritmy jsou obvykle tvořeny nějakým způsobem výpočtu absolutní orientace ve spojení s vhodným filtrem vstupních dat. Nejrozšířenější metodou pro zpracování dat z více senzorů je tzv. komplementární filtr vyznačující se zejména svým jednoduchým principem. Je možné najít mnoho různorodých implementací lišících se zejména složitostí. V této práci jsme se zaměřili na některé z nich od těch nejjednodušších: tzv. „Dirty“ komplementární filtr [29] [30], filtr s horní / dolní propustí a nelineární komplementární filtr [31] až po ty poněkud složitější: Mahony filtr [32] či Madgwick filtr [28].

4.5.1 Princip Komplementárního filtru

Komplementární filtr, jak už z názvu vyplývá, kombinuje dohromady data z několika (nezávislých) senzorů pro dosažení co nejlepšího výsledku. Vysvětlení tohoto principu ukážeme na příkladu dat z IMU senzoru poskytující pouze zrychlení a úhlovou rychlost. Komplementární filtr jednoduše řečeno

využívá data buď z akcelerometru, nebo gyroskopu podle toho co je pro danou situaci vhodnější. Pro krátkodobé časové úseky je použit gyroskop. Jeho data jsou velmi přesná / citlivá a nejsou nijak náchylná na vnější vlivy. Nevýhodou je však jeho tzv. „drift“, kdy hodnoty postupem času tzv. „ujíždí“. Proto pro dlouhodobé hledisko používáme data z akcelerometru. Akcelerometr reaguje na pohyb pomaleji, ale časem data neujíždí (nemá „drift“) jako v případě gyroskopu. Data z obou typů senzorů se tedy vhodně doplňují podle aktuální rychlosti pohybu. [30]

4.5.2 Kálmánův filtr

Dalším velmi používaným algoritmem v této oblasti je Kálmánův filtr [33] nebo jeho rozšířená verze pro linearizaci problému. Kálmánův filtr je silný nástroj dosahující velmi přesných výsledků využívaný zejména v komerčních produktech, nicméně má i mnoho nevýhod. Jeho implementace není tak snadná jako je tomu v případě komplementárního filtru a je potřeba určit správně parametry příslušného senzoru. Další nevýhodou je velké množství vnitřních stavů a náročné numerické výpočty v reálném čase. Je tedy potřeba celkem výkonný hardware. Kvůli výše zmíněným důvodům se Kálmánovým filtrem v této práci dále nebudeme zabírat. [28]

4.5.3 Výpočet úhlů orientace ze surových dat senzorů

Tato část se věnuje základnímu popisu výpočtu dvou úhlů natočení z dat akcelerometru a gyroskopu. Aby bylo možné použít vhodný komplementární filtr, musí se ve většině základních / jednoduchých případů nejprve zpracovat surová data ze senzorů a získat dostupné parametry orientace (úhly). Přímou ze změřených dat akcelerometru stanovíme dva úhly natočení φ_{accel} a θ_{accel} (a to ty vůči zemi) podle vztahů (5) a (6), kde a_x, a_y, a_z jsou naměřené hodnoty gravitačního zrychlení.

$$\varphi_{accel} = \operatorname{atan2}\left(\frac{a_y}{\sqrt{a_x^2 + a_z^2}}\right) \quad (5)$$

$$\theta_{accel} = \operatorname{atan2}\left(\frac{a_x}{\sqrt{a_y^2 + a_z^2}}\right) \quad (6)$$

Tyto úhly můžeme rovněž stanovit i z dat gyroskopu. Jelikož gyroskop měří přímo úhlové zrychlení ω , dostaneme pomocí jednoduché integrace (7) přímo úhel natočení φ v příslušné ose.

$$\varphi = \int \omega dt \quad (7)$$

Náš gyroskop však měří úhlové zrychlení v diskretních časech, proto provádíme integraci numericky způsobem popsaným rovnicemi (8) a (9) pro stanovení dvou úhlů natočení φ_{gyro} a θ_{gyro} , kde T_s je vzorkovací perioda gyroskopu a k je přirozené číslo.

$$\varphi_{gyro}[kT_s] = \varphi_{gyro}[(k-1)T_s] + \omega_x[kT_s] T_s \quad (8)$$

$$\theta_{gyro}[kT_s] = \theta_{gyro}[(k-1)T_s] + \omega_y[kT_s] T_s \quad (9)$$

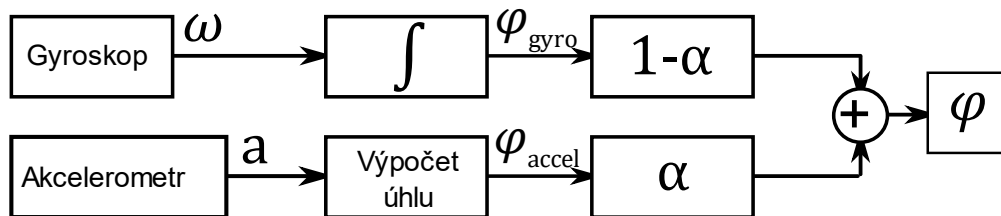
Vzorkovací periodu jednoduše vypočteme ze vzorkovací frekvence f_s .

$$T_s = \frac{1}{f_s} \quad (10)$$

Nyní známe dva úhly natočení v ose X a Y stanovené dvěma způsoby, čehož se využívá při použití IMU senzoru. Můžeme tedy data dále zpracovávat např. pomocí jednoduchého komplementárního filtru, čemuž se věnují dvě následující kapitoly 4.5.4 a 4.5.5.

4.5.4 „Dirty“ komplementární filtr

Jak již bylo zmíněno dříve, název „Dirty“ komplementární filtr označuje nejjednodušší implementaci komplementárního filtru, jehož principiální schéma je na Obrázek 18. Pro tento filtr byly pro jednoduchost použity pouze data z akcelerometru a gyroskopu.



Obrázek 18: Blokové schéma tzv. „Dirty“ komplementárního filtru.

Na začátku schématu je znázorněn výpočet úhlů natočení φ_{accel} a φ_{gyro} (viz. kapitola 4.5.3) a ty jsou následně zpracovány komplementárním filtrem popsaným rovnicemi (11) a (12) [29] [30]. Úhel v ose Z neuvažujeme, je tedy stanoven pevně na nulu.

$$\varphi = \alpha \varphi_{accel} + (1 - \alpha)\varphi_{gyro} \quad (11)$$

$$\theta = \alpha \theta_{accel} + (1 - \alpha)\theta_{gyro} \quad (12)$$

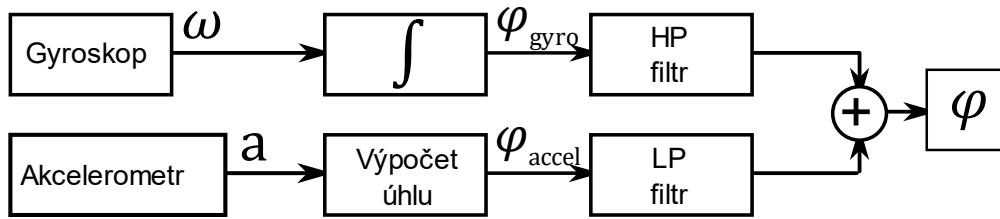
$$\psi = 0 \quad (13)$$

Koeficient $\alpha \in \langle 0; 1 \rangle$ udává, jak moc využíváme data z akcelerometru či gyroskopu. Experimentálně byl stanoven na hodnotu 0,1. Preferujeme tedy více gyroskop a výsledná měřená orientace je plynulejší bez výrazného kmitání a reaguje téměř okamžitě. Problém však nastane tehdy, začneme-li senzorem rychle otáčet. Výrazně se začne projevovat drift gyroskopu a nedostaneme se zpět do výchozí pozice natočení. V případě vysoké α by tomu bylo naopak, snadněji bychom udrželi výchozí pozici, avšak zaznamenali bychom kmitání (šum).

Tento typ filtru je sice velmi jednoduchý, ale neuvažuje žádnou dynamiku (jak rychle se momentálně sensor pohybuje). Jinak řečeno koeficient α je konstantní, tedy data z akcelerometru a gyroskopu se stále používají ve stejném poměru. Někdy by však bylo vhodné preferovat data více z jednoho senzoru a jindy zase z druhého. Tudíž tento filtr není příliš vhodný, a proto se v dalších kapitolách zaměříme na „výkonnější“ filtry / algoritmy.

4.5.5 Komplementární filtr s horní / dolní propustí (lineární)

Tento typ filtru bývá také označován jako „lineární“ díky své lineární struktuře. Základem tohoto komplementárního filtru je filtr horní propust (High Pass) pro data z gyroskopu a filtr dolní propust (Low Pass) pro data z akcelerometru. Výsledný úhel orientace φ je pak tvořen součtem z výstupů těchto dvou filtrů dle principiálního schématu na Obrázek 19. Jako u předchozího filtru („Dirty“ komplementární filtr) je nejprve nutné zpracovat surová data ze sensorů do tvaru úhlů orientace φ_{accel} a φ_{gyro} (4.5.3). [31] [29]



Obrázek 19: Blokové schéma komplementárního filtru s HP a LP filtrem.

Jednoduše řečeno tento filtr bere pomalu měnící se signály z akcelerometru a rychle měnící se signály z gyroskopu a ty následně kombinuje dohromady. Tedy z akcelerometru jsou propouštěny dolní frekvence signálů, a naopak z gyroskopu pouze horní frekvence, než je námi stanovená mezní frekvence.

Následuje popis implementace vnitřních filtrů HP (High Pass) a LP (Low Pass) pracujících diskřetně. Uvažujeme-li diskřetní proměnnou x jako vstup do filtru a proměnnou y jako výstup z filtru v časových okamžicích $k \in \mathbb{N}$, tak rovnice (14) odpovídá popisu HP filtru a rovnice (15) LP filtru. [34] [35]

$$y[k] = \alpha_{HP} y[k - 1] + \alpha_{HP}(x[k] - x[k - 1]) \quad (14)$$

$$y[k] = \alpha_{LP} x[k] + (1 - \alpha_{LP}) y[k - 1] \quad (15)$$

Pro oba filtry máme parametry $\alpha_{HP}, \alpha_{LP} \in \langle 0; 1 \rangle$ definované rovnicemi (16) a (17), kde τ je časová konstanta, T_s je vzorkovací perioda (10) a f_c je mezní frekvence [34] [35]. Velikost koeficientů α_{HP} a α_{LP} nám udává, jak moc výstup filtru závisí na okamžité vstupní hodnotě či hodnotě předcházející.

$$\alpha_{HP} = \frac{\tau}{\tau + T_s} = \frac{1}{2\pi T_s f_c + 1} \quad (16)$$

$$\alpha_{LP} = \frac{T_s}{\tau + T_s} = \frac{2\pi T_s f_c}{2\pi T_s f_c + 1} \quad (17)$$

$$f_c = \frac{1}{2\pi\tau} \quad (18)$$

Tedy celý komplementární filtr můžeme nastavovat jedním parametrem, a to buď pomocí f_c nebo τ , které se dají mezi sebou vzájemně přepočítat dle (18). V našem testovacím případě jsme stanovili hodnotu mezní frekvence experimentálně na $f_c = 1 \text{ Hz}$. Při nastavení desetkrát vyšší frekvence již pozorujeme výrazné kmitání (šum) ve statických pozicích. Naopak při použití např. desetkrát nižší frekvence je pohyb sice velmi plynulý, ale při rychlejším otáčení má dlouhou dobu ustálení do cílové pozice.

Tato implementace filtru je již trochu komplikovanější než předchozí, ale bereme už v potaz, jak rychle se senzor otáčí. Díky parametru f_c můžeme nastavit mez, kdy se využívají data z akcelerometru (ideální ve statických polohách) a kdy se naopak začnou vyžívat data z gyroskopu (ideální v dynamických polohách). Tedy hlavním cílem je správně odladit parametr f_c , tak, aby výsledný pohyb nekmital a zároveň pokud možno pracoval dostatečně rychle v reálném čase (měl krátkou dobu ustálení). Nastavení pro každý typ senzoru je samozřejmě individuální. Nevýhoda lineární struktury filtru spočívá v neposkytnutí spolehlivého odhadu orientace, jelikož celkový systém je nelineární a senzor (např. gyroskop) má proměnnou odchylku (bias) [36] [37]. Proto se v dalších kapitolách zaměříme na přesnější filtry, a to s nelineární strukturou.

4.5.6 Non-linear komplementární filtr

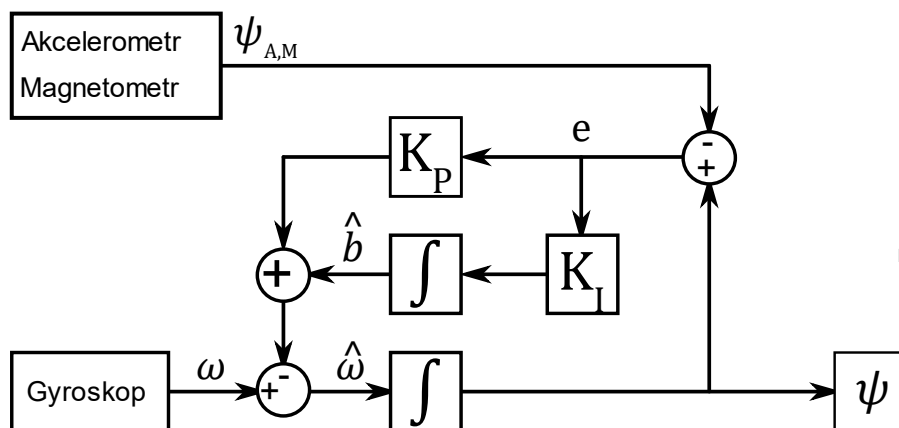
Označení „Non-linear“ tohoto v pořadí třetího komplementárního filtru značí, že se jedná o filtr s nelineární strukturou. U tohoto filtru jsme již použili i magnetometr a lze tedy získat kompletní orientaci senzoru ve 3D prostoru. Principiální schéma (Obrázek 20) tohoto filtru bylo převzato [31].

Vstupními veličinami do filtru jsou nyní přímo úhlová zrychlení gyroskopu a úhly stanovené z dat akcelerometru / magnetometru. Jelikož tentokrát používáme i magnetometr (ne jako v 4.5.3), vypočteme úhly orientace ($\varphi_{A,M}$, $\theta_{A,M}$ a $\psi_{A,M}$) ve všech třech osách dle rovnic (19), (20) a (21), kde a_x , a_y , a_z jsou surová data z akcelerometru a m_x , m_y , m_z normovaná data z magnetometru [31].

$$\varphi_{A,M} = \text{atan2}\left(\frac{a_y}{a_z}\right) \quad (19)$$

$$\theta_{A,M} = \text{atan2}\left(\frac{-a_x}{a_y \sin \varphi + a_z \cos \varphi}\right) \quad (20)$$

$$\psi_{A,M} = \text{atan2}\left(\frac{m_x \sin \varphi - m_y \cos \varphi}{m_x \cos \theta + m_y \sin \theta \sin \varphi + m_z \sin \theta \cos \varphi}\right) \quad (21)$$



Obrázek 20: Blokové schéma nelineárního komplementárního filtru s označením „Non-linear“.

Vysvětlení symbolů ze schématu:

- $\psi_{A,M}$ vypočtený úhel z akcelerometru a magnetometru
- ω měřené úhlové zrychlení z gyroskopu
- ψ výsledný odhadovaný úhel natočení
- e chyba (error) mezi odhadovaným a měřeným (vypočteným) úhlem
- \hat{b} odhadovaná odchylka gyroskopu
- $\hat{\omega}$ odhadované úhlové zrychlení
- K_I integrační zesílení
- K_P proporcionální zesílení

Z blokového schématu jsme odvodili matematické vztahy mezi vstupy a výstupem filtru. Ty jsme následně převedli do diskrétní formy, abychom je mohli implementovat v programu. Nejprve jsme si vyjádřili vztah pro \hat{b} daný integrálem (22), pomocí kterého jsme stanovili $\hat{\omega}$ (23). Výsledný odhadovaný úhel tedy dostaneme pouhou integrací (24) odhadovaného úhlového zrychlení.

$$\hat{b}(t) = \int K_I (\psi(t) - \psi_{A,M}(t)) dt \quad (22)$$

$$\hat{\omega}(t) = \omega(t) - (\hat{b}(t) + K_P (\psi(t) - \psi_{A,M}(t))) \quad (23)$$

$$\psi(t) = \int \hat{\omega}(t) dt \quad (24)$$

Převedení na diskretní popis je vyjádřeno rovnicemi (25) a (26), kde T_s je vzorkovací perioda systému a $k \in \mathbb{N}$ značí diskretní časové okamžiky.

$$\hat{b}[kT_s] = \hat{b}[(k-1)T_s] + K_I T_s (\psi[(k-1)T_s] - \psi_{A,M}[kT_s]) \quad (25)$$

$$\psi[kT_s] = \psi[(k-1)T_s] + T_s (\omega[kT_s] - \hat{b}[kT_s] - K_P (\psi[(k-1)T_s] - \psi_{A,M}[kT_s])) \quad (26)$$

Pro ideální činnost tohoto nelineárního filtru je však nutné správně nastavit dva parametry K_P a K_I sloužících k ladění filtru. V našem případě jsme je dle doporučení a experimentů autora schématu nastavili na hodnoty $K_P = 5$ a $K_I = 0,1$. [31]

Hlavní výhodou nelineárního filtru oproti předchozím lineárním je jeho schopnost brát v potaz chyby mezi orientacemi dílčích senzorů, a tím vylepšovat výstupní odhadované úhly [36]. Tedy tento filtr poskytuje spolehlivější odhad orientace pro nelineární systém se senzorem mající proměnnou odchylku (bias) [37]. Za nevýhodu tohoto základního nelineárního schématu se považuje nutnost ručně nastavit dva konkrétní parametry K_P a K_I závisující na dynamice systému [31]. Pokud však není vyžadována příliš vysoká přesnost, je tento základní filtr velmi často využíván i pro komerční produkty [31]. Za další nevýhodu implementace tohoto filtru by se dal považovat jeho výstup pouze ve formě Eulerových úhlů viz. 3.1. V dalších kapitolách se podíváme na základy složitějších nelineárních filtrů od konkrétních autorů implementovaných s využitím kvaternionů.

4.5.7 Mahony filtr

Jedním ze složitějších ale již přesnějších filtrů / algoritmů pro stanovení orientace je tzv. Mahony filtr [32] představující komplementární filtr navržený Robertem Mahony a jeho kolektivem. Jedná se o nelineární komplementární filtr na speciální ortogonální skupině. I tento filtr byl navržen autorem s motivací využívat ho primárně na levných IMU senzorech tvořených MEMS technologií a na méně výkonném hardwaru. Matematický princip tohoto filtru je založen na matici Lieových grup reprezentované SO(3). [32]

Jako implementaci Mahony filtru (AHRS algoritmu) jsme použili již hotový a volně dostupný kód na webu [38] vytvořený od S. Madgwick. Pro optimální činnost filtru je potřeba správně nastavit parametr s označením K_P . V našem případě byla dle doporučení autora nastavena hodnota na $K_P = 5$ [28]. Oproti předchozím implementacím jednodušších filtrů je tento algoritmus již implementován ve formě kvaternionů, což poskytuje značné výhody viz. 3.2.

4.5.8 Madgwick filtr

Jako druhý složitější a výkonnější filtr / algoritmus byla využita práce od Sebastian O.H. Madgwick [28]. Tento filtr je rovněž navržen pro IMU/MARG senzory a mělo by jej být možné provozovat rovněž na méně výkonném hardwaru než třeba Kálmánův filtr. Madgwick filtr. Pracuje na principu iterativní optimalizační metody sestupný-gradient či Newtonovy metody využívajících analytickou formulaci gradientu odvozenou z kvaternionové reprezentace. Pro výrazné urychlení výpočtu je zde využita analytická derivace Jacobiho matice. Tento filtr zahrnuje kompenzaci magnetického zkreslení a odchylky gyroskopu. [28]

Hlavními výhodami Madgwickova nelineárního filtru (AHRS algoritmu) dle autora je jeho celkem nízká výpočetní náročnost a efektivní výkon i při nízké vzorkovací frekvenci. Podle autora tento filtr dosahuje

stejně přesnosti jako filtr na principu Kálmána. Filtr rovněž pracuje s orientací ve formě kvaternionů a lze ho ladit pouze jedním parametrem s označením β . Jako implementaci tohoto filtru jsme převzali verzi kódu přímo od autora Madgwick volně dostupnou z webu [38]. V našem případě byla dle experimentů autora filtru hodnota parametru nastavena na $\beta = 0,041$. [28]

4.6 Podpora nastavení / experimentů / kalibrace...

Jelikož se jedná o experimentální / vývojové zařízení, bylo v aplikaci pro PC navrženo mnoho možností interpretace, zpracování a kalibrace dat z použitých senzorů. V nastavení aplikace byla vytvořena možnost volby: rychlosti získávání dat ze senzorů, typ algoritmu pro zpracování dat, zobrazení pouze požadovaných záložek aplikace či volba auto-kalibrace pro nastavení výchozí orientace při pohybu určitou částí těla. Pro ulehčení práce s různými typy senzorů (základní dva: s MPU9250 a mobilní telefon) bylo vytvořeno konfigurační menu před spuštěním vlastní aplikace, kde je možné přidávat vlastní profily pro různé senzory a lze tedy jejich kalibrační hodnoty a výše zmíněná nastavení ukládat zcela odděleně.

4.6.1 Zobrazení dat ze senzorů

Základní možností zobrazení přicházejících dat (gravitační zrychlení, úhlové rychlosti a magnetické intenzity) ze senzoru je jejich zobrazení v prosté textové formě, tedy číselnou formou. Tento způsob interpretace dat může být při rychlých změnách číselných hodnot špatně čitelný, proto je možné data zobrazovat v reálném čase do 2D grafu jako časové průběhy ("2DLines"). Jednotlivé průběhy lze v nastavení libovolně skrývat, zvětšovat či zastavit.

Pro přicházející data z magnetometru byl speciálně vytvořen ještě jeden 2D graf ("2DXY") vhodný zejména při jeho kalibraci. Tento 2D graf slouží jako náhrada za 3D graf, ve kterém by se na každou osu vynášely data z odpovídající osy magnetometru. Do jednoho 2D grafu je tedy možné vynášet pouze barevně odlišené projekce do rovin XY, ZY a ZX odpovídající plnému 3D grafu. Jde v podstatě o barevně odlišené tři 2D grafy přes sebe. Tento způsob zobrazení byl využit nejen z důvodu jednodušší implementace, ale i kvůli lepší čitelnosti 2D grafu. Přijaté body se v tomto grafu postupně shromažďují až do jeho vyčištění.

Jako poslední možnost zobrazení přicházejících dat je 3D graf ("3DXYZ") s vektory měřeného gravitačního zrychlení a magnetické intenzity aktualizující se v reálném čase. Tento způsob prezentace je vhodný pro lepší představivost principu činnosti při volném otáčení senzorem. Podrobnější informace ohledně zobrazení dat ze senzorů jsou v kapitole 5.6.

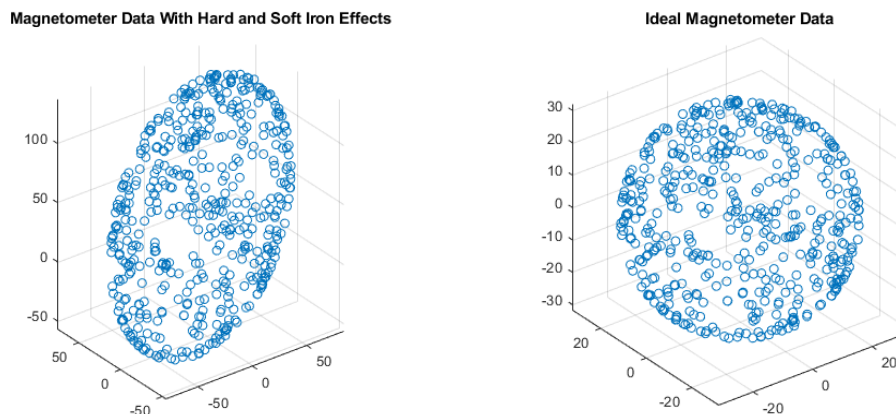
4.6.2 Kalibrace senzoru

Každý vyrobený senzor má samozřejmě poněkud odlišné parametry, proto je vždy potřeba vykonat nějakou minimálně základní kalibraci. V jistých případech může být však kalibrace nutná i před každým novým použitím senzoru, tedy před měřením. Na senzor mohou vždy působit jiné okolní vlivy nebo se také jeho fyzické parametry mění v důsledku opotřebení a samozřejmě nejčastěji s časem nebo teplotou. Po kalibraci (např.: přidání offsetu k hodnotám) již můžeme data dále zpracovávat cílovými algoritmy. Pro použitý samotný senzor MPU9250/55 (napojený na procesor) je samozřejmě nutné vykonání jeho kalibrace, kdy surová data jsou velmi nepřesná pro další zpracování (zejména gyroskop a magnetometr). V případě použití mobilu jako senzoru není kalibrace přicházejících dat pro naši aplikaci většinou vůbec nutná, poněvadž data jsou již kalibrována v mobilu od výrobce. Náš proces kalibrace byl implementován v aplikaci na PC, kde jsou rovněž ukládány kalibrační hodnoty do souboru pro jejich opakované použití (při měření).

Pro většinu základních použití, kde není vyžadována vysoká přesnost (např.: rotace obrazovky na šířku / výšku), není potřeba akcelerometr uživatelem nikterak kalibrovat [39]. Kalibrace tohoto

senzoru z výroby je plně dostačující [39] [40]. Toto platí i pro naši aplikaci akcelerometru. Nicméně i přesto byl navržen velmi jednoduchý způsob kalibrace akcelerometru popsán v kapitole 4.6.3. Naopak vykonání kalibrace gyroskopu je velmi důležité. Je totiž žádoucí, aby v klidovém stavu byly hodnoty ve všech osách gyroskopu nulové. U nekalibrovaného gyroskopu tomu však tak být nemusí. Pro docílení nulových hodnot je nejjednodušší pro každou osu odečíst příslušný offset. Tento postup je použit i v této práci a podrobněji popsán v kapitole 4.6.4.

Nejkomplikovanější a zároveň velmi důležitou činností je kalibrace magnetometru. Magnetometr by měl ideálně měřit pouze magnetické pole země. Pokud by tomu tak skutečně bylo a otáčeli bychom tří-osým magnetometrem, tak z naměřených hodnot (X, Y, Z) vynášených do 3D grafu dostaneme kulovou sféru se středem v počátku [41] [42], jak je uvedeno na Obrázek 21 vpravo. V případě již výše zmíněného 2D grafu (kapitola 4.6.1) by se jednalo o tři totožné vycentrované plné kružnice. Reálně jsou hodnoty z magnetometru zkreslené všemi nejen přímo magnetickými, ale i běžnými kovovými předměty zejména v jeho blízkém okolí. Rozlišujeme dva typy rušení. První je tzv. „Hard-Iron“, toto rušení je způsobeno trvalými magnetickými poli v okolí přičítající se k mag. poli země. Na vykreslené sféře se to projeví jejím posunutím. Druhým typem je rušení tzv. „Soft-Iron“ a je výsledkem materiálů narušujících / deformujících magnetické pole země, např.: kov (železo). Záleží také na skutečné orientaci materiálu vzhledem k magnetometru. Toto zkreslení se projevuje deformací kulové sféry. Oba typy zkreslení současně jsou zřejmé z Obrázek 21 vlevo. V této práci byla implementována jednoduchá a však zcela dostačující metoda kompenzace jak „Hard-Iron“ tak i „Soft-Iron“ zkreslení popsána podrobněji v části 4.6.5. [41] [43]



Obrázek 21: Data z magnetometru se zkreslením typu „Hard-Iron“ i „Soft-Iron“ (vlevo) a ideální / kalibrovaná data (vpravo). [41]

4.6.3 Kalibrace akcelerometru

Uživatelská kalibrace akcelerometru není u většiny základních aplikací vůbec potřeba, a to platí i pro náš vyvíjený systém, jak již bylo zmíněno výše. I přesto zde popíšeme jeden základní a jednoduchý postup kalibrace akcelerometru, který není sice implementován v této aplikaci, ale pro svoji jednoduchost stojí za zmínění.

Postup vykonání této základní kalibrace spočívá v následujícím. Jako referenční hodnoty pro kalibraci se využívá gravitačního zrychlení země $9,8 \text{ m/s}^2$ neboli 1 g, jelikož akcelerometr měří samozřejmě i toto zrychlení. Budeme tedy vycházet ze základního předpokladu, kdy je jistá osa akcelerometru ve svislém směru dolů a je na ní měřena hodnota +1 g (v opačném otočení -1 g). Na zbývajících dvou osách ve vodorovném směru je nulové gravitační zrychlení, tedy 0 g. Máme celkem 6 kalibračních poloh (pro každou osu kladný / záporný vertikální směr). Pro každou osu vypočteme dvě kalibrační hodnoty, a to

zesílení a posun. Zesílení stanovíme jako rozdíl průměrné hodnoty při měření +1 g a průměru při poloze -1 g. Poté rozdíl vydělíme dvěma. Velikost posunu vypočteme jako průměrnou hodnotu z pozic, kdy by se na dané ose nemělo projevit gravitační zrychlení (tedy 0 g). K tomuto dojde pro každou osu celkem čtyřikrát. [44]

Abychom nemuseli při kalibraci senzor natáčet přesně do šesti kalibračních poloh (jak již bylo zmíněno, kalibrace akcelerometru není nikterak potřeba), dovolili jsme si implementovat pouze velmi jednoduchý postup výpočtu kalibračních hodnot. Senzor musí být při tomto procesu kalibrace položen, tak aby vertikální osa (Z) byla směrem nahoru. Postup kalibrace je popsán následujícím pseudokódem, kde stanovíme pro každou osu kalibrační hodnotu (jednotky jsou „g“):

```
calibValue[x] = value[x]
calibValue[y] = value[y]
calibValue[z] = value[z] - 1
```

Pokud kalibrační hodnoty aplikujeme při měření, výpočet korekce probíhá následujícím způsobem:

```
FOREACH axis
    accelValue = accelValue - calibValue
```

4.6.4 Kalibrace gyroskopu

Pro efektivní činnost algoritmů při výpočtu orientace (zejména těch jednoduchých) je vykonání kalibrace gyroskopu nutné, zvláště v naší aplikaci se senzorem MPU9250/55. Bez této kalibrace by docházelo ke zbytečně rychlé integraci chyby (offsetu). Základním cílem zde popisované kalibrace je v klidovém stavu zajistit, aby byly měřené hodnoty ve všech osách gyroskopu nulové. Jednoduše řečeno je tedy potřeba pro každou osu zjistit hodnotu, později odečítanou od dat přicházejících přímo ze senzoru.

V našem případě byl použit velmi jednoduchý postup. Když je senzor v klidovém stavu (bez pohybu na pevné podložce), tak z přicházejících dat pro jednotlivé osy hledáme maximální a minimální hodnoty po dobu např.: 500 vzorků (samples). Následně pro každou osu vypočteme střední hodnotu jako střed mezi zaznamenanou maximální a minimální hodnotou. Tyto vypočtené hodnoty poté slouží jako kalibrační hodnoty pro později přicházející data, a tedy pro vytvoření nulového výstupu v klidovém stavu gyroskopu.

Tento postup je popsán následujícím pseudokódem, kde stanovíme pro každou osu jednu kalibrační hodnotu:

```
FOREACH sample
    FOREACH axis
        IF max[axis] < sample[axis]
            max[axis] = sample[axis]
        IF min[axis] > sample[axis]
            min[axis] = sample[axis]
    FOREACH axis
        calibValue[axis] = (max[axis] + min[axis]) / 2
```

Pokud kalibrační hodnoty aplikujeme při měření, výpočet probíhá následujícím způsobem:

```
FOREACH axis
    gyroValue = gyroValue - calibValue
```

4.6.5 Kalibrace magnetometru

Kalibrace magnetometru je již přímo nutná, zejména abychom měli pevně / dobře stanovenou referenci pro otáčení ve vertikální ose (Z). Důležitost vykonání kalibrace, závisí rovněž na prostředí, v němž je senzor umístěn. Například, pokud se senzor nachází v budově a kolem je ještě navíc mnoho magnetických polí a kovových předmětů zkreslujících mag. pole země, je rozhodně žádoucí kalibrace uskutečnit. To platí i pro naši aplikaci / systém, a tudíž byl implementován následující proces kalibrace.

Prozatím byl využit pouze základní princip kalibrace. Při výpočtu korekce pro rušení typu „Hard-Iron“ jsou hledány minimální a maximální hodnoty v přijímaných datech pro každou ze tří os samostatně. Výsledná korekce je střed mezi nejmenší a největší hodnotou v dané ose, a tedy tvořící posun v této ose. Pro typ rušení „Soft-Iron“ nejprve vypočteme rozměr / rozsah (scale) tvaru elipsy / kružnice pro příslušnou osu (vzdálenost min a max hodnoty). Dále se stanoví průměrný rozměr elipsy / kružnice ze všech tří os (average). Nakonec se dopočtou konstanty pro přepočet / transformaci každé osy do skutečně kruhového rozměru. [42] [43]

Tento postup je popsán následujícím pseudokódem, kde stanovíme pro každou osu dvě kalibrační hodnoty (Hard / Soft – Iron):

```
FOREACH sample
    FOREACH axis
        IF max[axis] < sample[axis]
            max[axis] = sample[axis]
        IF min[axis] > sample[axis]
            min[axis] = sample[axis]
    FOREACH axis
        calibHardIron[axis] = (max[axis] + min[axis]) / 2
        scale[axis] = (max[axis] - min[axis]) / 2
    average = (scale[x] + scale[y] + scale[z]) / 3
    FOREACH axis
        calibSoftIron[axis] = average / scale[axis]
```

Pokud kalibrační hodnoty aplikujeme při měření, výpočet probíhá následujícím způsobem:

```
FOREACH axis
    magValue = (magValue - calibHardIron) * calibSoftIron
```

4.7 Implementace a struktura aplikace

Aplikace pro PC je určena pro operační systém MS Windows a byla vytvořena na bezplatné platformě Microsoft .NET Core. Jedná se o nejčastěji používaný framework pro Microsoft / Windows obsahující vysoko úroňový programovací jazyk C# s částečnou možností použití i pro další operační systémy jako je Linux a macOS. Framework .NET Core je multiplatformní nástupce známého .NET Framework (původně určený pouze pro MS Windows). Pro tvorbu grafického rozhraní aplikace byla použita knihovna tříd WPF (Windows Presentation Foundation) jako součást .NET frameworků. Jelikož v naší aplikaci pracujeme se 3D objekty, byla navíc pro WPF použita rozšiřující knihovna s názvem HelixToolkit. Hlavní výhodou této knihovny je její bezplatnost a možnost jednoduššího načtení, zobrazení, manipulace a tvorby i složitějších 3D objektů.

Programový kód je rozdělen do několika samostatných souborů, avšak (zatím) tvořících jednu (parciální) třídu k základní třídě „MainWindow“. Toto je z důvodu zejména snadného vývoje aplikace, aby byly ve všech souborech přístupné všechny GUI (Graphical User Interface) prvky a hlavní (GUI) vlákno aplikace.

Zde je uveden výčet souborů a jejich stručný obsah:

- RehabMain – Hlavní / úvodní část pro načtení modelu a vytvoření zobrazení.
- RehabCommun – Komunikace pro přenos pohybových dat z mobilního zařízení s Google / Android přes Bluetooth (Classic) nebo z vlastního HW pomocí VCP (Virtual Communication Port) / USB.
- RehabData – Zpracování (externích) přijatých dat a aktualizace modelu podle nich.
- RehabCalibration – Soubor obsahující funkce pro kalibraci.
- RehabTimers – Definice a inicializace časovače auto-kalibraci (výchozí pozice) natočení.
- RehabAlgorithms – Implementace algoritmů / filtrů pro výpočet orientace v prostoru.
- RehabMovement – Vykonání pohybu celého modelu nebo pouze jeho částí.
- RehabAngles – Vytvoření a aktualizace grafických úhloměřů.
- RehabMeasure – Měření / cvičení s nastavitelnými parametry a ovládací prvky.
- RehabResults – Vytvoření a zobrazení výsledků měření (graf, tabulka) jednotlivých cvičeních.
- RehabConverters – Převod číselného rozsahu úhlu na grafické zobrazení (šířka pruhu).
- RehabDef – Nastavení a vytvoření částí modelu postav a jednotlivých měřitelných cvičeních.
- RehabSetupDlg a RehabSetupClasses – Nastavení aplikace.
- RehabConsts – Definice několika výčtových typů (enum) používaných v kódu.
- RehabOthers – Některé další konstanty a třídy použité v aplikaci.
- RehabHelp – Pomocný objekt používaný pro určení pozice v modelu.

V dalších kapitolách se podrobněji zaměříme na implementaci hlavních a nejdůležitějších souborů aplikace. Zejména se podíváme na dílčí funkce / metody jednotlivých částí parciální třídy a jejich detailnější popis.

4.7.1 RehabMain

Hlavní / základní soubor obsahuje inicializace, načtení a vytvoření 3D modelu vybrané postavy skládající se z jednotlivých částí (.obj souborů). Dále je zde obsaženo nastavení zobrazované plochy (HelixToolkit viewportu), vytvoření a nastavení kamery, světla, možnosti ovládání myši a mnoho dalšího. Rovněž je zde proces přiřazení zobrazených částí modelu pro jednotlivá definovaná měřitelná cvičení (MeasureTask).

4.7.2 RehabCommun

Tento soubor obsahuje veškeré činnosti zajišťující komunikaci mezi pohybovým senzorem (vysílačem) a aplikací na PC (příjímačem). V případě ARM procesoru je využit VCP/USB a v případě mobilního telefonu je to Bluetooth. Součástí tohoto souboru je také třída Paket poskytující metody pro práci s daty na úrovni jednoho přenášeného paketu (bloku dat).

CommunInit()

- Inicializace základní komunikace nezávisle na později použitým konkrétním typu skutečné komunikace.

CommunClose()

- Ukončení základní komunikace bez ohledu na jejím dříve použitým typu.

ShowDataReset()

- Nastavení GUI do výchozího stavu vymazáním přijatých dat.

AxisConvert()

- Prohození přijatých dat v osách (X, Y, Z) dle nastaveného požadavku, tak aby uspořádání hodnot / os odpovídalo dalšímu zpracování dat a zejména zobrazení pohybu.

class Paket{...}

- Třída obsahující potřebné metody pro práci s paketem, tedy jeho složení a rozložení na jednotlivé položky.

Metody zajišťující komunikaci přes Bluetooth:

CommunBTOnSearchSelect(...)

- Spuštění hledání a následný výběr požadovaného zařízení ze seznamu všech nalezených zařízení.

CommunBTConnectOnClick(...)

- Připojení a odpojení Bluetooth zařízení vybraného z dříve dostupného seznamu. Připojení na zařízení je nutné pro přenos pohybových dat.

CommunBTStartStopOnClick(...)

- Spuštění a zastavení přenosu dat přes Bluetooth do PC. Spuštění / zastavení přenosu dat je uskutečněno vysláním patřičného povelu do zařízení obsahující pohybový senzor.

CommunBTOnRecvData(...)

- Metoda pro zpracování jednoho celého datového paketu / bloku z pohybového senzoru. Jeden paket / blok obsahuje data ze všech měřených os (X, Y, Z) a senzorů.

CommunBTSetState(...)

- Metoda pro zpracování stavu komunikace pro jeho zobrazení v GUI. Tedy při úspěšném připojení na Bluetooth zařízení a při jeho odpojení.

OnRecvDataText(...)

- Událost vyvolaná při příjmu jednoho celého datového paketu / bloku z pohybového senzoru přes Bluetooth. Jeden paket / blok obsahuje data ze všech měřených os (X, Y, Z) a senzorů.

GetValuesXYZ(...)

- Vyzvednutí jednotlivých hodnot (dat pro osy X, Y, Z) z textového paketu a jejich následný převod na interní čísla typu „Double“.

Metody zajišťující komunikaci přes VCP:

ComboBoxVCPDropDownOpened(...)

- Událost při otevření „ComboBoxu“ pro možnost zobrazení seznamu dostupných COM/VCP portů v použitém počítači.

CommunVCPConnect(...)

- Připojení (otevření) a odpojení (uzavření) komunikačního COM/VCP portu zařízení. Rovněž uložení úspěšně použitého „COM/VCP“ pro příští spuštění aplikace.

CommunVCPStartStopOnClick(...)

- Skutečné spuštění / zastavení přenosu pohybových dat do aplikace na PC. Spuštění / zastavení přenosu dat je uskutečněno vysláním patřičného povelu do zařízení obsahující pohybový senzor.

SerialPortOnDataReceived(...)

- Událost / metoda vyvolaná při příjmu jednoho celého datového paketu / bloku z pohybového senzoru přes Bluetooth. Jeden paket / blok obsahuje data ze všech měřených os (X, Y, Z) a všech senzorů.

CommunVCPSetState(...)

- Metoda pro zpracování stavu komunikace pro jeho zobrazení v GUI. Tedy při úspěšném připojení na Bluetooth zařízení a při jeho odpojení.

OnRecvDataBin(...)

- Událost vyvolaná při příjmu na úrovni jednotlivých byte. Ty jsou skládány do paketu a teprve při jeho úspěšném příjmu je volána metoda „SerialPortOnDataReceived“.

4.7.3 RehabData

V tomto souboru jsou zejména definice proměnných obsahující data přicházející z pohybového senzoru, dále vypočtené úhly pro pohyb 3D postavy a rovněž parametry zobrazované jako výsledky do grafů. Hlavní náplň tohoto programového kódu spočívá ve zpracování a následném zobrazení dat pomocí zde uvedených metod:

DataInit()

- Základní inicializace všech datových položek potřebných později při činnosti aplikace pro jejich ukládání při příjmu, výpočty úhlů a samozřejmě v poslední míře zobrazení pomocí tabulek a grafů.

DataShowTextOnChecked(...)

- Povolení / blokování a vytvoření textového zobrazení přicházejících dat. Data mohou být zobrazena buď jako tzv. surová, nebo již kalibrovaná.

DataShowGraphOnClick(...)

- Zobrazení požadovaného typu grafu, tedy "2DLines", "2DXY" a "3DXYZ". Jde tedy o inicializaci vybraného grafu a směrování příchozích hodnot / dat do tohoto grafu.

CalibrationOnClick(...)

- Žádost o spuštění / ukončení kalibrace, tedy sběr hodnot potřebných pro vykonání kalibraci. Vykonání příslušné kalibrace je obsaženo v souboru „RehabCalibration“.

OnDataProcess()

- Zpracování / zobrazení dat: aplikace kalibračních hodnot, zobrazení textových dat, výpočet algoritmů, aktualizace modelu – vykonání pohybu 3D postavou (pokud je aktivní nějaké měření), zobrazení dat v aktivních grafech.

4.7.4 RehabCalibration

Soubor obsahuje tři metody vykonávající přímo skutečný výpočet kalibračních hodnot (ukládány v PC) pro tři dílčí senzory: akcelerometr, gyroskop a magnetometr.

CalibAccel(...)

- Kalibrace akcelerometru obsahující tyto části: nulování původních kalibračních hodnot, čekání na sběr hodnot pro kalibraci a vlastní výpočet kalibrace.

CalibGyro(...)

- Kalibrace gyroskopu obsahující tyto části: nulování původních kalibračních hodnot, inicializace nové kalibrace, čekání na sběr hodnot pro kalibraci, zpracování dat a vlastní výpočet.

CalibMag(...)

- Kalibrace magnetometru obsahující tyto části: nulování původních kalibračních hodnot, inicializace nové kalibrace, čekání na sběr dostatečného množství hodnot pro kalibraci, zpracování dat a vlastní výpočet.

4.7.5 RehabTimers

Soubor obsahuje definici / inicializaci časovače (třídy Timer) a potřebné metody pro výpočet tzv. auto-kalibrace [45] neboli nulování úhlů natočení do výchozí pozice před začátkem (v podstatě samo) měření / cvičení. Toto se ale vykonává pouze v případě, pokud je k popisu orientace (natočení) využita interpretace pomocí kvaternionů.

TimerInit()

- Inicializace všech parametrů časovače.

TimerStart()

- Spuštění časovače s nastaveným časem spolu s vizualizací dobře viditelného číselného odpočtu na ploše aplikace.

OnTimedEvent(...)

- Událost na jeden tik časovače (1 s) a po uplynutí času vykonání potřebné kalibrace [45] před měřením / cvičením.

4.7.6 RehabAlgorithms

Soubor obsahuje implementace několika algoritmů / filtrů na výpočet orientace v prostoru a jednu metodu pro spuštění konkrétního algoritmu podle aktuálního nastavení v aplikaci.

AlgorithmStep()

- Vykonání výpočtu jednoho kroku nastaveného algoritmu, tedy výpočet skutečné orientace v prostoru. Tato metoda v podstatě pouze volá algoritmus aktuálně nastavený / vybraný v aplikaci, v podstatě mu předává příchozí (již kalibrovaná) data.

AlgorithmNone()

- Testovací / výchozí algoritmus násobící pouze hodnoty z akcelerometru pro stanovení úhlů natočení.

AlgorithmDirtyComplementaryFilter()

- Implementace algoritmu dle 4.5.4 nazvaného „Dirty“ komplementární filtr.

AlgorithmComplementaryFilter()

- Implementace algoritmu dle 4.5.5 Komplementární filtr s horní / dolní propustí (lineární).

AlgorithmNonLinComplementaryFilter()

- Implementace algoritmu dle 4.5.6 Non-linear komplementární filtr.

AlgorithmMahonyFilter()

- Implementace algoritmu dle 4.5.7 Mahony filtr.

AlgorithmMadgwickFilter()

- Implementace algoritmu dle 4.5.8 Madgwick filtr.

4.7.7 RehabMovement

Soubor obsahuje metody zajišťující pohyb / aktualizaci 3D postavy (modelu) podle příchozích pohybových dat a k tomu samozřejmě potřebné výpočty jednotlivých rotačních transformací.

MovementInit()

- Základní inicializace všech dat spojených s pohybem celé postavy a rovněž jejich částí. Tedy zejména zpracování bodů otáčení a návaznosti jednotlivých částí postavy.

MoveResetOnClick(...)

- Nastavení tzv. polohy každé části modelu postavy do výchozí, tedy nulové pozice. Nulová / výchozí pozice je formou běžného stoje.

MoveManualOnUpdate(...)

- Aktualizace natočení / pohybu všech částí 3D modelu při (manuální) změně posuvníku v GUI. Sloužící například pro testovací účely při tvorbě vlastní 3D postavy.

MoveWholeBody()

- Vykonání pohybu / natočení celým modelem / postavou jako jedním objektem. Jde tedy o základní / celkový pohyb postavou.

MoveUpdateModelPositions()

- Postupné procházení všech částí modelu a aktualizace jejich natočení / pohybu podle aktuálního vypočteného úhlu natočení. Při pohybu jednotlivých částí postavy je samozřejmě nutná jejich vzájemná návaznost.

MovePartBase(...)

- Výpočet rotační transformace pro (hlavní) část postavy nemající na další části žádnou návaznost (tělo / hrudník). Podle nastaveného algoritmu je výpočet transformace proveden buď z formátu úhlů, nebo z kvaternionů.

MovePartNext(...)

- Výpočet rotační transformace pro část postavy mající návaznost na další část (všechny ostatní části kromě hrudníku). Podle nastaveného algoritmu je výpočet transformace uskutečněn buď z formátu úhlů, nebo z kvaternionů.

4.7.8 RehabAngles

Soubor zajišťující vytvoření / zobrazení 3D úhloměřů umístěných na požadované části těla postavy, tedy tam kde právě probíhá měření pohybu. Kromě číselné hodnoty dosažených úhlů pohybu máme k dispozici i jejich grafickou reprezentaci. Kompletní úhloměř je tvořen třemi kružnicovými výsečemi (pro osy X, Y a Z) s čárkovanou stupnicí a zobrazuje nejen aktuálně měřené úhly, ale i maximálně dosažený rozsah pohybu (min / max úhel) pro právě měřenou úlohu.

DisplayProtractor(...)

- Vytvoření a přidání do scény nového úhloměru (případně i odebrání již existujícího nepotřebného) s požadovanými parametry dle aktuálního předdefinovaného měřeného cvičení (umístění, rozsah, ...).

OnUpdateX(...), OnUpdateY(...) a OnUpdateZ(...)

- Události o změnách v parametrech pro zobrazený úhloměř v osách X, Y a Z. Jedná se o požadavky na změnu hodnoty / rozsahu, viditelnosti a dosažených min / max hodnot úhlu.

class Protractor{...}

- Třída definující a vytvářející 3D úhloměř s požadovanými parametry (pozice, velikost, rozsah apod.) pomocí knihovny HelixToolkit.

class ProtractorParams{...}

- Třída definující základní parametry úhloměru pro jeho správné zobrazení / vykreslení.

4.7.9 RehabMeasure

Soubor obsahuje třídu definující možnosti jednotlivých měřicích úloh / cvičení spolu s dalšími pomocnými třídami. Jsou zde také implementovány metody mající za úkol zajistit přímo vlastní měření rozsahu pohybů a obsluhu (aktivaci, nulování, zobrazení apod.) předdefinovaných cvičení.

MeasureInit()

- Základní inicializace datových položek pro vykonání skutečného měření. Tvorba / inicializace tříd pro ukládání naměřených dat (rozsah pohybu), jejich možné uložení do databáze a samozřejmě pozdější zobrazení jako výsledky.

MeasureRefresh()

- Aktualizace všech položek zobrazených v GUI. Tedy všech informací ohledně aktuální hodnoty úhlu a samozřejmě maximálních dosažených mezí.

SelectedOnChecked(...)

- Obsluha aktivace / deaktivace požadované měřicí úlohy / cvičení. Dojde ke zjištění nově aktivní úlohy, a tedy nastavení jejích parametrů. Následné procházení a nastavení všech částí postavy do výchozí pozice (nulování). Nakonec zobrazení úhloměru pro požadovanou úlohu a spuštění auto-kalibrace (pokud je zaškrtnuto v nastavení aplikace).

ButtonStartOnClick(...)

- Spuštění vybrané měřicí úlohy. Aktivní úloha je ta vybraná neboli zatržená (označená zelenou barvou).

ButtonStopOnClick(...)

- Zastavení právě aktivní měřicí úlohy. Po zastavení měření jsou samozřejmě uchovány všechny dosažené hodnoty / úhly, zůstanou tedy zobrazeny.

ButtonResetOnClick(...)

- Nulování celého měření pro všechny tři osy X, Y a Z. Uvedení měření do výchozího stavu.

ButtonResXOnClick(...), ButtonResYOnClick(...) a ButtonResZOnClick(...)

- Nulování pouze jedné z požadovaných os.

class MeasureTask{...}

- Třída definující jednu měřicí úlohu s parametry jako jsou aktivní / neaktivní části postavy, nastavení úhloměru, potřebné prohození os, potřebné aktualizace v GUI apod.

class MeasurePartAxe{...}

- Třída obsahující definice a případně aktualizace dosaženého úhlu, rozsahu pohybu, max / min limitu rozsahu a reset těchto hodnot pro jednu osu měření.

class ExcerciseActivePrms{...}

- Třída obsahující definice max / min dosažených úhlů ve všech osách aktivní části postavy. Toho se využívá pro definici jednotlivých měřicích úloh / cvičení.

class ExcerciseOtherPrms{...}

- Třída obsahující definice / parametry / stav / postoj dalších částí postavy neaktivních při tomto cvičení. Toho se využívá pro definici jednotlivých měřicích úloh / cvičení.

4.7.10 RehabResults

Soubor je tvořen metodami mající za úkol grafickou interpretaci (formou tabulky či grafu) naměřených hodnot / výsledků pro jednotlivé měřicí úlohy / cvičení v závislosti na čase. Pro tvorbu grafu byla použita knihovna LiveCharts. Součástí tohoto souboru jsou rovněž třídy definující formát zaznamenávaných / prezentovaných dat (dosažených úhlů).

ResultsInit()

- Základní inicializace datových položek potřebných pro zobrazení výsledků měření formou tabulky nebo grafu.

ResultsDataOnSelectionChanged(...)

- Aktualizace zobrazení požadovaných výsledků měření. Vstupní data jsou vyzvednuta z patřičných tříd a zobrazena podle nastaveného typu.

ResultTableOrGraphOnClick(...)

- Nastavení typu zobrazení výsledků měření. Přepínání mezi zobrazením formu tabulky a grafu.

ResultsShow(...)

- Vlastní zobrazení výsledků měření buď formou grafu, nebo tabulkou.

CreateHorLine(...)

- Vytvoření vodorovné čáry v grafu určující nějakou mezní hodnotu, například ideální mez / hodnotu pro dosažený úhel.

class ResultModel{...}

- Třída obsahující definici sady záznamu s výslednými dosaženými úhly / rozsahy pohybu pro požadovanou měřicí úlohu (jednu část těla).

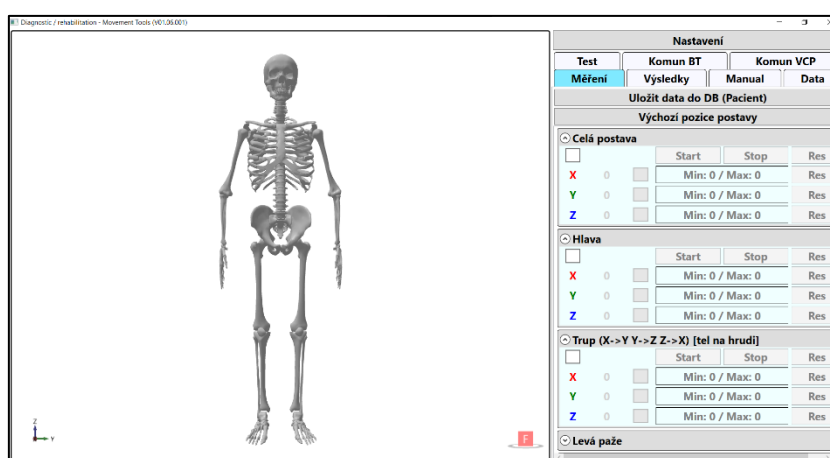
class ResultPart{...}

- Třída obsahující definici jednoho záznamu tvořeného indexem, časem, naměřenými hodnotami (úhly) a potřebnými prvky pro aktualizaci v GUI.

5 Experimentální aplikace

V této kapitole podrobněji popíšeme naši experimentální aplikaci / zařízení z pohledu uživatele. Jinak řečeno jedná se o jakýsi návod na použití aplikace, tedy jak se přesněji postupuje při práci s ní a jaké jsou možnosti nastavení / konfigurace apod. Zároveň se tedy v této části práce přehledně (z pohledu uživatele) dozvíme všechny dostupné funkce / možnosti, jimiž tato aplikace disponuje.

Hlavní okno aplikace se skládá v levé části z 3D scény / zobrazení (v kódu označováno jako viewport) s postavou a v pravé části se nachází momentálně otevřená záložka (jedna z mnoha záložek) aplikace s odpovídajícími funkcemi a tlačítko „Nastavení“ jak je znázorněno na Obrázek 22. V dalších částech textu postupně popíšeme: možnosti nastavení, proces spuštění, 3D zobrazení postavy a obsah jednotlivých záložek aplikace (Manual, Komun VCP, Komun BT, Data, Měření a Výsledky).



Obrázek 22: Hlavní plocha vytvořené aplikace pro PC po jejím spuštění. Vlevo 3D postava a vpravo záložky pro ovládání.

5.1 Nastavení

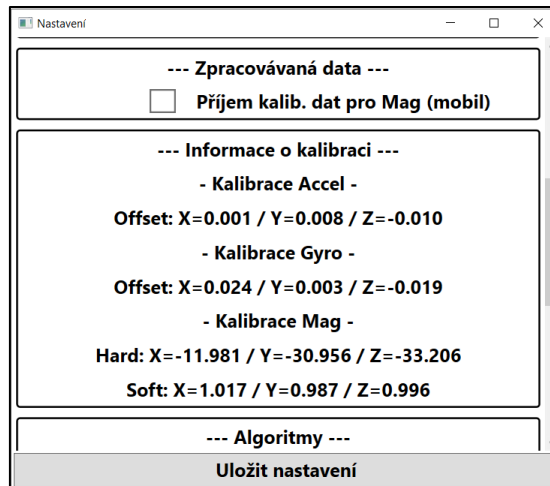
V samostatném dialogu aplikace s názvem „Nastavení“ jsou k dispozici veškerá dostupná nastavení pro aplikaci, se kterými se dále podrobněji seznámíme. Při změně jakéhokoliv nastavení je vždy nutné pro jeho potvrzení, uložení a aktivaci kliknout dole na tlačítko „Uložit nastavení“ viditelné na Obrázek 23. V první sekci „Zobrazena postava (typ)“ je možnost výběru ze čtyř postav (Kostra, Muž, Žena a Ironman) sloužících k interpretaci měřeného pohybu. Další dvě položky „Ambient“ a „Brightness“ slouží pro nastavení světla 3D scény (postavy) a je možné je nastavit na hodnoty od 0,1 do 1 (představují v podstatě jas zobrazené postavy).



Obrázek 23: Část dialogu aplikace pro „Nastavení“ obsahující: nastavení postavy a vzorkování / zobrazení.

Ve druhé sekci „Vzorkování / zobrazení“, rovněž znázorněné na Obrázek 23, je možné nastavit rychlost měřených / přijímaných dat ze senzorů (akcelerometr, gyroskop a magnetometr) položkou „Rychlost dat“ na hodnoty od 1 Hz do 500 Hz po určitých krocích. Vzorkovací frekvence pro magnetometr se dá dále do-nastavit pomocí tzv. děličky – položka „Dělička pro Mag“. Tedy výsledná rychlost dat z magnetometru bude hodnota „Rychlost dat“ dělena hodnotou „Dělička pro Mag“. Jinak řečeno, pokud „Dělička pro Mag“ bude rovna jedné, magnetometr bude mít stejnou vzorkovací frekvenci jako akcelerometr a gyroskop. Jako poslední položka v této sekci je „Dělička pro GUI“ sloužící k nastavení rychlosti / frekvenci aktualizace grafického zobrazení. Tuto děličku lze nastavit na celé hodnoty od 1 do 10. Jako v předchozím případě při nastavení na hodnotu jedna se dělička neuplatní.

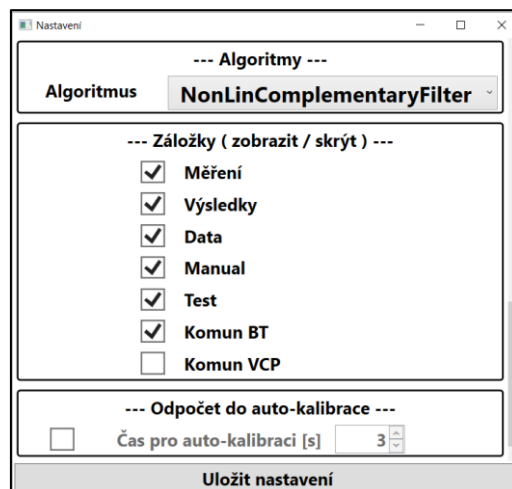
Třetí sekce s označením „Zpracovávaná data“ viditelná na Obrázek 24 obsahuje jediné zaškrtnuté políčko „Příjem kalib. dat pro Mag (mobil)“. Při zaškrtnutí této volby přijímáme již rovnou kalibrovaná data magnetometru ze senzoru v mobilu anebo v opačném případě pouze surová data. Tato možnost nastavení je dostupná pouze pokud používáme jako pohybový senzor mobilní telefon, poněvadž mobilní telefony již přímo v sobě umožňují vykonávat automaticky a průběžně (nevíme však přesně kdy) kalibraci magnetometru od výrobce. Toto je zejména vhodné pro vývojáře mobilních aplikací, ale v našem případě to může být i nežádoucí, protože naše aplikace na PC rovněž obsahuje proces kalibrace magnetometru, tak by mohlo dojít k vzájemnému rušení těchto kalibrací. Pokud tedy přijímáme již kalibrovaná data přímo z mobilu, nedoporučujeme už používat kalibraci v aplikaci na PC a naopak!



Obrázek 24: Část dialogu pro „Nastavení“ obsahující: zpracovávaná data a informace o kalibraci.

Čtvrtá sekce s názvem „Informace o kalibraci“ neobsahuje žádnou možnost nastavení, ale jak již název napovídá a je zřejmé z Obrázek 24, jsou zde k nahlédnutí pouze všechny aktuálně uložené kalibrační hodnoty pro tři dílčí senzory: akcelerometr, gyroskop a magnetometr. V případě akcelerometru a gyroskopu se jedná pouze o hodnoty „Offset“ (posun), ale pro magnetometr jsou zde uloženy dva typy kalibračních hodnot a to „Hard“ a „Soft“ (blíže definované v kapitole 4.6.2 Kalibrace senzoru).

V páté sekci „Algoritmy“, znázorněné na Obrázek 25, je možné zvolit algoritmus / filtr použitý na zpracování dat z dílčích senzorů. Je zde na výběr celkem šest možností, kde možnost „None“ znamená bez použití algoritmu. Algoritmy s názvy „DirtyComplementaryFilter“ a „ComplementaryFilter“ poskytují měření pouze v ose X a Y (nepoužívají magnetometr), tedy není možné změřit natočení kolem osy Z (svislá osa)! Zbývající tři algoritmy „NonLinComplementaryFilter“, „MahonyFilter“ a „MadgwickFilter“ používají již i magnetometr, a tedy poskytují kompletní stanovení orientace ve 3D prostoru. Za poznámku zde stojí ještě informace, že poslední dva zmíněné algoritmy („Mahony, Madgwick“) díky použití kvaternionové interpretace poskytují i plynulou vizualizaci měřeného pohybu v celém úhlovém rozsahu na rozdíl od Eulerových úhlů viz. kapitola 3. Podrobnější informace o jednotlivých algoritmech byly popsány v kapitole 4.5 Algoritmy.



Obrázek 25: Část dialogu pro „Nastavení“ obsahující: nastavení algoritmu, záložek a auto-kalibrace.

Šestá sekce v nastavení nazvaná „Záložky (zobrazit / skrýt)“ obsahuje seznam všech dostupných záložek v aplikaci spolu se zaškrťovacími políčky určujícími, zda je daná záložka (v seznamu záložek) v hlavním okně aplikace viditelná či nikoliv, jako je tomu na Obrázek 25. Podrobný popis funkcí jednotlivých záložek je popsán v následujících podkapitolách.

Poslední a sedmá sekce „Odpčet do auto-kalibrace“ obsahuje zaškrťovací políčko zapnutí / vypnutí auto-kalibrace a možnost nastavení časového intervalu do vykonání vlastního procesu auto-kalibrace, jak je vidět na Obrázek 25. Zapnutí této položky má smysl pouze v případě používání výše zmíněných algoritmů „MahonyFilter“ a „MadgwickFilter“ (protože používají kvaternionovou reprezentaci orientace), u zbývajících algoritmů to nemá žádný význam a nedojde k vykonání žádného výpočtu (auto-kalibrace). Jednoduše řečeno auto-kalibrace uskuteční nulování / otočení měřené části těla do výchozí pozice zobrazení (nezáleží tedy na tom, jak je senzor fyzicky umístěn vůči magnetické referenci – obvykle severní mag. pól Země). Samotný proces auto-kalibrace je vykonáván před každým měřením / cvičením definovaným v záložce aplikace s názvem „Měření“, více informací přímo v této podkapitole 5.8 Měření / cvičení.

5.2 Spuštění aplikace

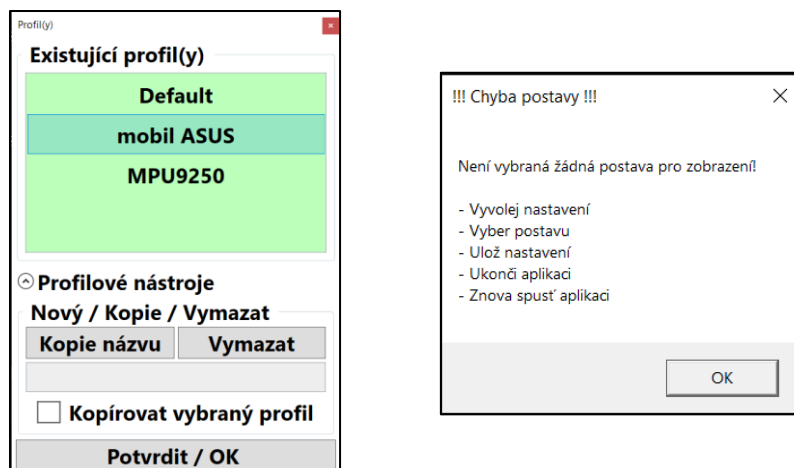
Zde si nejprve definujeme pojem „Profil aplikace“ a dále popíšeme, jak postupovat při nejen prvním spuštění aplikace na PC.

5.2.1 Profil aplikace

Profilem v aplikaci rozumíme jedno specifické nastavení pro aplikaci s veškerými prvky (typ postavy, rychlost dat, kalib. hodnoty apod.) dostupnými v dialogu „Nastavení“ (podkapitola 5.1 Nastavení). Hlavním využitím těchto profilů je možnost mít uložená jednotlivá nastavení například pro několik různých typů pohybových senzorů. V základním případě se může jednat o jeden profil pro senzor tvořený mobilním telefonem a druhý pro senzor využívající vlastní HW s MPU9250 (např. mající jiné kalib. hodnoty a jiný způsob připojení) jako je tomu na Obrázek 26 vlevo.

5.2.2 Spuštění aplikace (poprvé)

Při (prvním) spuštění aplikace se nám zobrazí konfigurační dialog (Obrázek 26 - vlevo) obsahující pouze jeden výchozí profil nazvaný „Default“. Ten je zde trvale a nelze jej tedy odstranit. Po rozkliknutí nabídky „Profilové nástroje“ můžeme přidat nový profil buď s vlastním názvem vepsaným do textového políčka, nebo lze tlačítkem „Kopie názvu“ překopírovat název již existujícího zvýrazněného profilu z nabídky výše. Tedy v případě, kdy chceme nový profil vytvořit jako kopii již existujícího zvýrazněného profilu, zaškrtneme položku „Kopírovat vybraný profil“ a nový profil bude obsahovat zcela stejnou konfiguraci / nastavení (bude mít stejné nastavení, kalib. hodnoty apod.). Zvýrazněný profil lze rovněž jednoduše smazat tlačítkem „Vymazat“. Pro spuštění aplikace s vybraným / označeným profilem stačí stisknout „Potvrdit / OK“.



Obrázek 26: Dialog pro výběr profilu aplikace (vlevo) a upozornění pro „Chyba postavy“ (vpravo).

Když spouštíme aplikaci úplně poprvé nebo jsme vytvořili nový čistý profil, zobrazí se varovná hláška „Chyba postavy“ spolu s postupem, jak tuto postavu vybrat viz. Obrázek 26 vpravo. Po stisku „OK“ se aplikace spustí bez postavy, tu poté jednoduše vybereme v „Nastavení“, nové nastavení uložíme, aplikaci zavřeme, znovu spustíme a pak již bude vše v pořádku.

5.3 Oblast s postavou (3D scéna)

Jednou z nejdůležitějších částí hlavního okna aplikace je 3D scéna / zobrazení neboli jednoduše řečeno oblast s postavou / modelem. Jedná se o pohled kamerou do kompletní 3D scény s jednou z předem nastavených postav (kostra, žena, muž nebo Ironman). Všechny tyto dostupné postavy jsou zobrazené na Obrázek 27. Hlavním úkolem této postavy je přesně reprezentovat v reálném čase měřené pohyby fyzickým senzorem (IMU/MARG) umístěným na vhodné části těla skutečného člověka. Pro dobrou názornost budou všechny další zde uvedené obrázky z vytvořené aplikace obsahovat pouze jednu postavu a to kostru.



Obrázek 27: Dostupné postavy do 3D scény aplikace (z leva): kostra, žena, muž a Ironman.

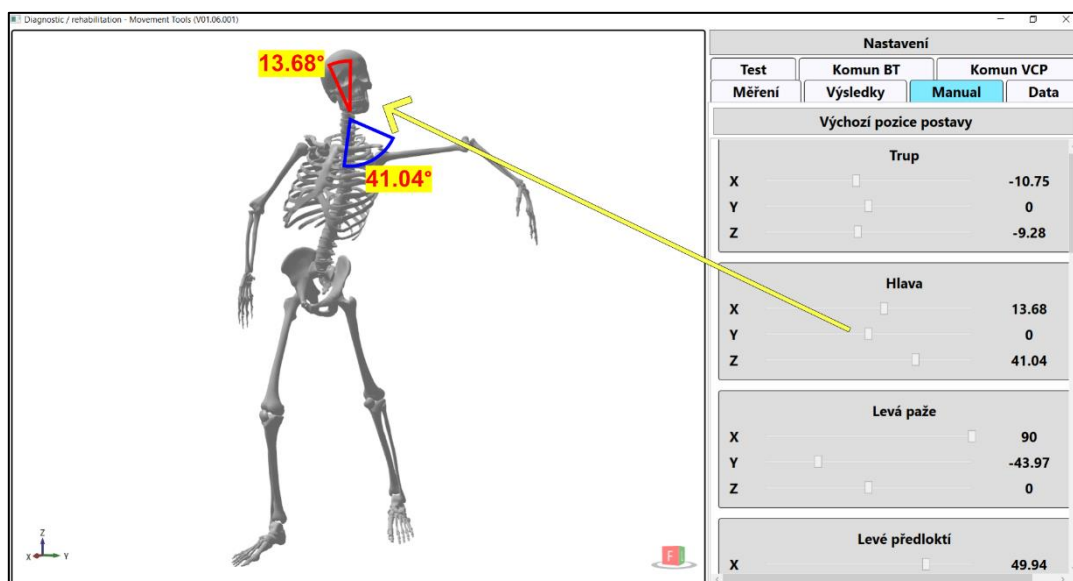
Pohled kamerou na postavu / model lze libovolně podle potřeby ovládat myší. Otáčení (rotace) kolem postavy se uskuteční pohybem myši při držení jejího levého tlačítka. Pohyb v rovině (posun) ovládáme tentokrát přidržením kolečka myši s požadovaným posunem. K přiblížení / oddálení (zoom) slouží klasicky kolečko myši. K rychlému a přesnému nastavení pozice kamery (např. pohled zepředu, z vrchu, zleva atd.) lze využít barevnou krychli umístěnou v pravém dolním rohu, kde stačí kliknout na její požadovanou stěnu (červená – F: pohled zepředu, zelená – L: levá strana postavy atd.) a kamera se automaticky ihned nastaví. Případně krychli otáčet pomocí jejího uchopení myší.

Jako poslední prvek nacházející se v tomto okně je barevný souřadnicový systém umístěný v levém dolním rohu viditelný na každém z dílčích obrázků na Obrázek 27. Tento prvek je zejména vhodný

při vývoji / ladění aplikace a zobrazuje pevný souřadnicový systém používaný v celém tomto projektu, kde osa X – červená směřuje dopředu, osa Y – zelená směřuje doprava a osa Z – modrá nahoru.

5.4 Manuální pohyb

První zde podrobněji popsána záložka aplikace nese název „Manual“ a je zobrazená na Obrázek 28. Ke kompletní činnosti této záložky není potřeba mít připojen k PC žádný pohybový senzor, proto začínáme s popisem jednotlivých záložek právě zde. Záložka „Manual“ slouží k jakési simulaci pohybu libovolných částí postavy. Jednotlivé pohyby (natočení) částí postavy lze tedy manuálně ovládat zde zobrazenými posuvníky (neprovádí se zde žádná reálná měření / výpočty). Postava je celkem strukturovaná na 14 částí, s nimiž lze pohybovat: hlava, trup, pravá / levá – paže, předloktí, ruka, stehno, lýtko a chodidlo.



Obrázek 28: Záložka aplikace „Manual“ se znázorněním nastaveného natočení hlavy kostry pomocí posuvníků.

K natočení / pohybu každé z částí těla slouží tři posuvníky – každým lze nastavit úhel natočení od -90° do 90° pro každou osu otáčení (s označením X, Y a Z). Pokud chceme opět vynulovat všechny nastavené úhly (výchozí pozice), klikneme v horní části na tlačítko s označením „Výchozí pozice postavy“.

5.5 Nastavení komunikace (VCP, Bluetooth)

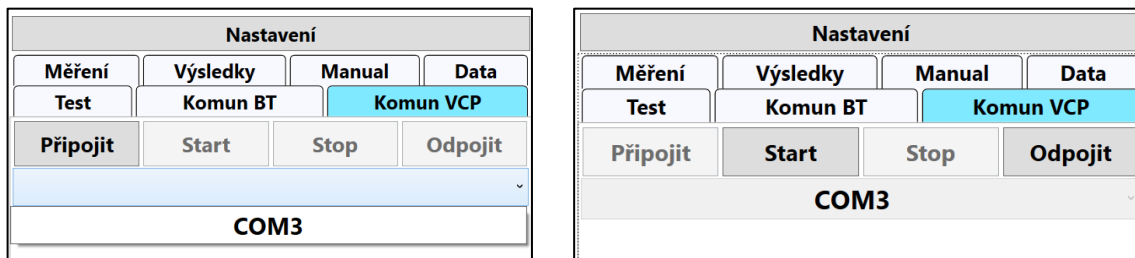
V této podkapitole podrobně popíšeme dvě záložky aplikace s názvy „Komun VCP“ a „Komun BT“ sloužící pro připojení a komunikaci s pohybovým senzorem. Jak již bylo v této práci několikrát zmíněno, rozlišujeme dva vytvořené typy pohybového senzoru:

- Vlastní HW senzor (např. s MPU9250) připojený přes USB kabel (COM/VCP).
- Senzor tvořen mobilním telefonem připojený přes Bluetooth.

5.5.1 Připojení senzoru přes USB

K navázání komunikace se senzorem připojeným pomocí USB slouží záložka s názvem „Komun VCP“. V prvním kroku připojení vybereme ze seznamu název COM/VCP portu, ke kterému je použitý senzor připojen jako je tomu na Obrázek 29 vlevo (v našem případě se jedná o port COM3.). Pro skutečné připojení senzoru stiskneme tlačítko „Připojit“. Pokud připojení proběhne úspěšně, zaktivuje se tlačítko „Start“ viditelné na Obrázek 29 vpravo, na které poté stačí rovněž kliknout. Při úspěšném startu

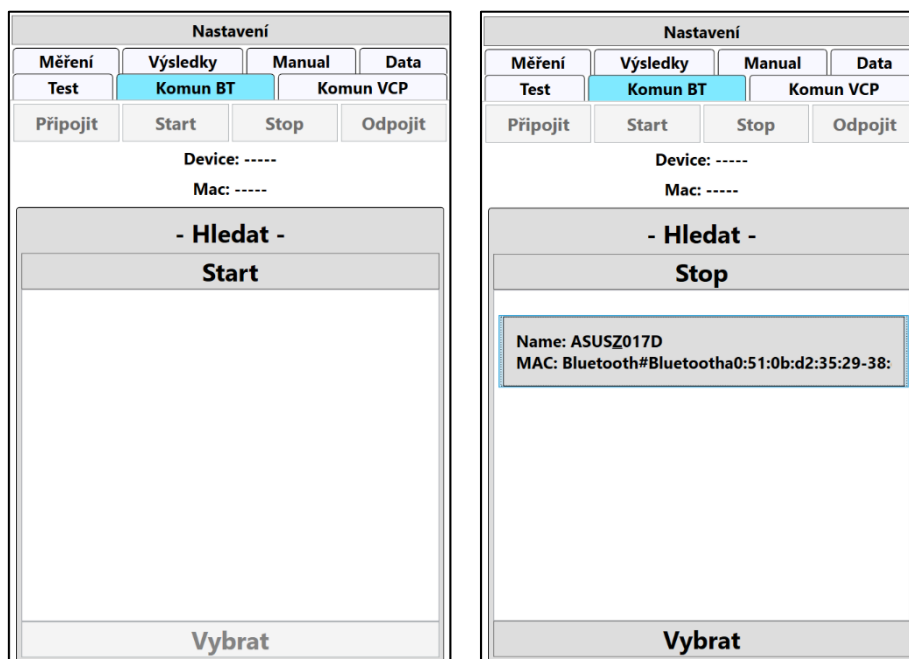
komunikace se zaktivuje tlačítko „Stop“, kterým je možné komunikaci později zastavit a případně následně senzor odpojit kliknutím na tlačítko „Odpojit“.



Obrázek 29: Záložka aplikace „Komun VCP“ pro připojení senzoru přes USB, první krok výběr portu (vlevo) a pak spuštění komunikace (vpravo).

5.5.2 Připojení senzoru (mobilu) přes Bluetooth

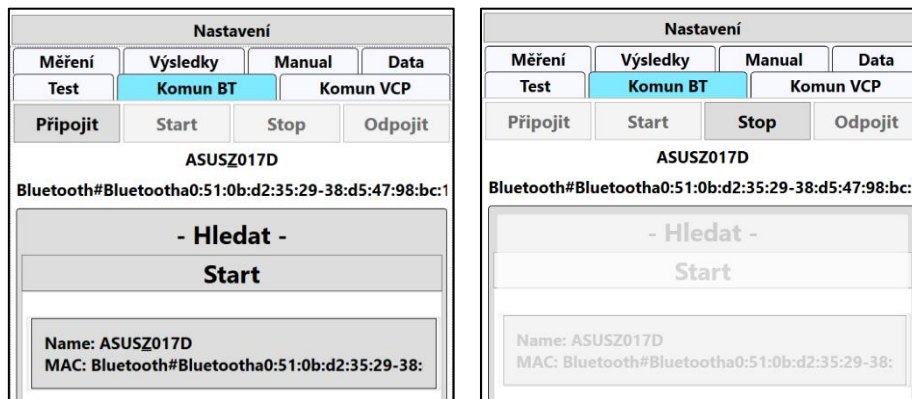
K navázání komunikace s mobilním telefonem (představující pohyb. senzor), na kterém je spuštěna obslužná aplikace (s názvem „MovementUni“) pro získání surových dat z dílčích senzorů a zajišťující Bluetooth přenos, slouží záložka s názvem „Komun BT“. V prvním kroku připojení klikneme v oblasti „- Hledat -“ na tlačítko „Start“ znázorněné na Obrázek 30 vlevo a mělo by dojít k vyhledání dostupných Bluetooth zařízení. Zde vybereme název našeho telefonu a klikneme na tlačítko „Vybrat“ vyobrazené na Obrázek 30 vpravo.



Obrázek 30: Záložka aplikace „Komun BT“ znázorňující start hledání BT zařízení (vlevo) a jeho výběr (vpravo).

Po úspěšném vybrání zařízení se v horní části automaticky vyplní položky „Device:“, příslušná „Mac:“ a zaktivuje se tlačítko „Připojit“ viditelné na Obrázek 31 vlevo, na které následně klikneme. Při úspěšném připojení dojde k zešednutí oblasti „- Hledat -“ a aktivaci tlačítka „Start“. Jako poslední krok pro zahájení skutečného přenosu dat stačí kliknout na tlačítko „Start“. Následně se dostaneme do stavu podle Obrázek 31 vpravo, kde je aktivní tlačítko „Stop“, pro pozdější zastavení přenosu dat a současně

případně i odpojení od Bluetooth zařízení kliknutím na tlačítko „Odpojit“. Samozřejmě vždy nesmíme úplně na začátku zapomenout zapnout Bluetooth, jak na PC, tak i na mobilu.

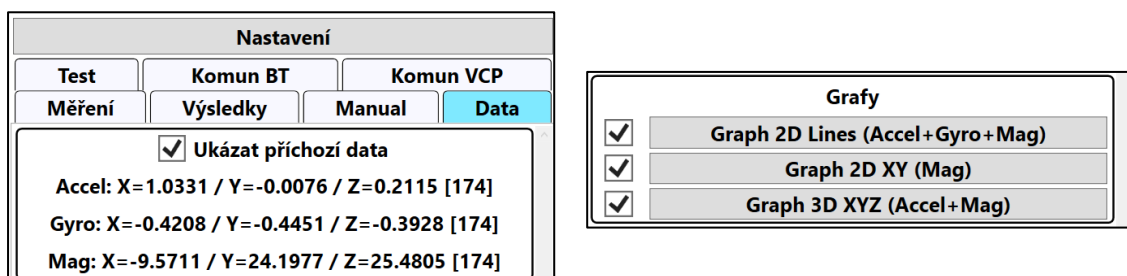


Obrázek 31: Záložka aplikace „Komun BT“ znázorňující stav před připojením k BT zařízení (vlevo) a po připojení, tedy již při příjmu dat (vpravo).

5.6 Data ze senzorů

Tato podkapitola se věnuje popisu záložky s názvem „Data“, avšak pouze části týkající se dat ze senzorů a části obsahující grafy. Pokud úspěšně navážeme komunikaci s pohybovým senzorem, buď už přes USB kabel (vlastní HW senzor) či pomocí Bluetooth (mobilní telefon), dle postupu popsaném v předešlé podkapitole 5.5, můžeme si správnost přijímaných dat jednoduše ověřit právě v záložce „Data“.

Pro základní a nejjednodušší zobrazení dat v textové formě stačí zatrhnout položku „Ukázat příchozí data“ jako je tomu na Obrázek 32 vlevo. Příchozí data ze senzoru je rovněž možné zobrazit i v přehlednější grafické formě, a to pomocí tří typů grafů v sekci „Grafy“ viditelné na Obrázek 32 vpravo. Pro otevření podokna aplikace s požadovaným grafem stačí kliknout na jeho název. Zaškrťovací políčka před názvy grafů slouží k jejich aktivaci / deaktivaci, tedy povolení / blokování vykreslování příchozích dat v konkrétním grafu. Podrobný popis jednotlivých grafů je v následujících podkapitolách.

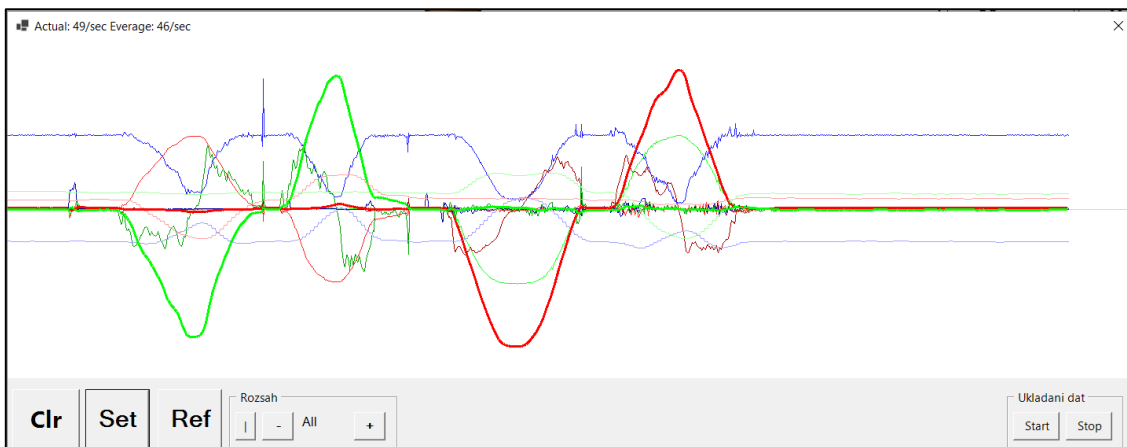


Obrázek 32: Část záložky aplikace „Data“ obsahující příchozí data (vlevo) a část s tlačítky pro zobrazení grafů (vpravo).

5.6.1 Graph 2D Lines

Způsob interpretace rychle měnících se dat (čísel) v textové formě je velmi špatně čitelný a neposkytuje nám informace o časovém vývoji měřených dat (jaké byly hodnoty v minulosti). Proto je možné data rovněž zobrazovat v reálném čase do 2D grafu jako běžné časové průběhy (měřené hodnoty v závislosti na čase). Příklad tohoto grafu je zobrazen na Obrázek 33, kde bylo naměřeno otočení se senzorem nejprve v ose otáčení Y o přibližně -90° , následně vrácení do výchozí pozice (0°) a znovu otočení

tentokrát o 90° s opětovným navrácením do 0°. To samé bylo hned poté vykonáno i v ose otáčení X. V grafu jsou vynášeny měřené hodnoty ve všech třech osách měření (X-červeně, Y-zeleně a Z-modře) a to ze tří dílčích senzorů: akcelerometr, gyroskop a magnetometr. Navíc jsou zde tlustou křivkou zobrazeny i průběhy jednotlivých výsledných úhlů natočení stanovených výpočetními algoritmy / filtry. Podrobný popis významu jednotlivých barev a rozsahů svislé osy je popsán v následujících odstavcích nebo přímo na Obrázek 34.



Obrázek 33: Graf typu „2D Lines“ zobrazující data ze senzorů a výsledné úhly (popis křivek v textu).

Pomocí tlačítek „-“ a „+“ v oblasti „Rozsah“ lze zmenšovat (oddalovat) či poté zpět zvětšovat (přiblížovat) časovou osu. Tlačítkem „Clr“ celý průběh vymažeme (a začne se vykreslovat znovu zleva – od nuly) a při stisku tlačítka „Set“ se nám otevře podokno s podrobným nastavením pro zobrazení grafu s legendou, viditelné na Obrázek 34, vysvětlující význam jednotlivých barevných křivek. Rovněž zde vyčteme i odpovídající rozsahy svislé osy grafu pro jednotlivé senzory / data.

All Default		All Enable		All Disable		
Item	Name	Range / Scale	Zoom		Shift	Note
Accel (Count: 3)		Default	Enable	Disable		
<input checked="" type="checkbox"/>	AccelX	-2 / 2	- 1x +	- 0 +		Accel X
<input checked="" type="checkbox"/>	AccelY	-2 / 2	- 1x +	- 0 +		Accel Y
<input checked="" type="checkbox"/>	AccelZ	-2 / 2	- 1x +	- 0 +		Accel Z
Gyro (Count: 3)		Default	Enable	Disable		
<input checked="" type="checkbox"/>	GyroX	-250 / 250	- 1x +	- 0 +		Gyro X
<input checked="" type="checkbox"/>	GyroY	-250 / 250	- 1x +	- 0 +		Gyro Y
<input checked="" type="checkbox"/>	GyroZ	-250 / 250	- 1x +	- 0 +		Gyro Z
Mag (Count: 3)		Default	Enable	Disable		
<input checked="" type="checkbox"/>	MagX	-200 / 200	- 1x +	- 0 +		Mag X
<input checked="" type="checkbox"/>	MagY	-200 / 200	- 1x +	- 0 +		Mag Y
<input checked="" type="checkbox"/>	MagZ	-200 / 200	- 1x +	- 0 +		Mag Z
Out (Count: 3)		Default	Enable	Disable		
<input checked="" type="checkbox"/>	OutX	-90 / 90	- 1x +	- 0 +		Out Test
<input checked="" type="checkbox"/>	OutY	-90 / 90	- 1x +	- 0 +		Out Test
<input checked="" type="checkbox"/>	OutZ	-90 / 90	- 1x +	- 0 +		Out Test

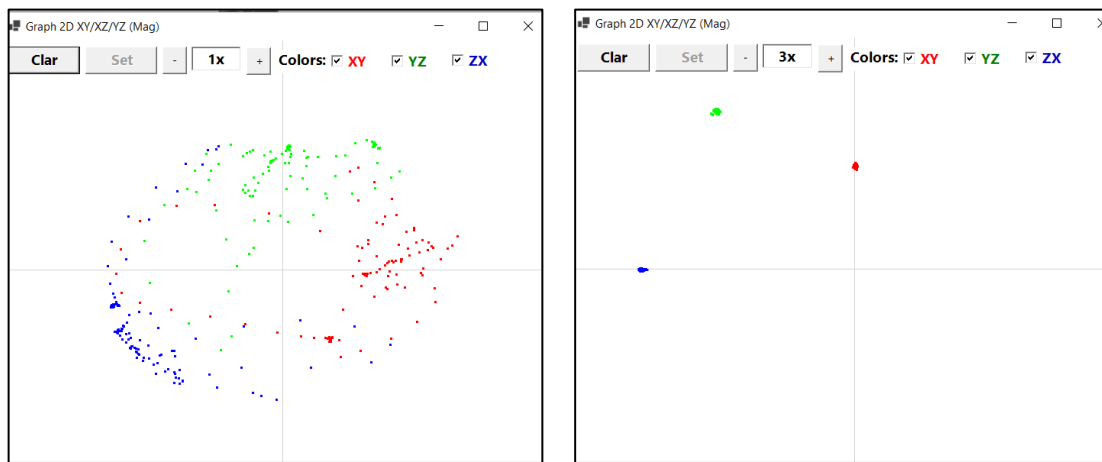
Obrázek 34: Okno s nastavením a legendou pro graf aplikace typu „2D Lines“.

V podrobném nastavení měřených průběhů můžeme libovolně jednotlivé průběhy zakázat / povolit, přiblížit / oddálit a posunout ve vertikální ose či dát jejich nastavení zpět do výchozího stavu. Rovněž

jsou v horní části tlačítka, jimiž lze hromadně vše resetovat do výchozího stavu nebo vše povolit či zakázat.

5.6.2 Graph 2D XY

Tento graf slouží pouze pro zobrazení dat z magnetometru a je vhodný zejména k vizualizaci procesu jeho kalibrace viz. Kalibrace 5.7. Do grafu jsou vynášeny barevně odlišené projekce do rovin XY – červeně, YZ – zeleně a ZX – modře. Jedná se o jakési zjednodušení, aby nemusely být jednotlivé měřené tři dimenzionální body (X, Y a Z) vykreslovány do 3D grafu. Jedna trojice barevných bodů (červený, zelený a modrý) představuje jedno měření (vzorek). Jinak řečeno, čím více bodů v grafu, tím déle probíhá měření (záznam hodnot do tohoto grafu). Na Obrázek 35 vlevo je znázorněn graf, při jehož záznamu bylo po určitou dobu náhodně otáčeno senzorem v různých směrech. Naopak na Obrázek 35 vpravo nebylo po určitou dobu se senzorem vůbec hýbáno, tedy všechny naměřená data / vzorky jsou vykresleny na stejném místě.

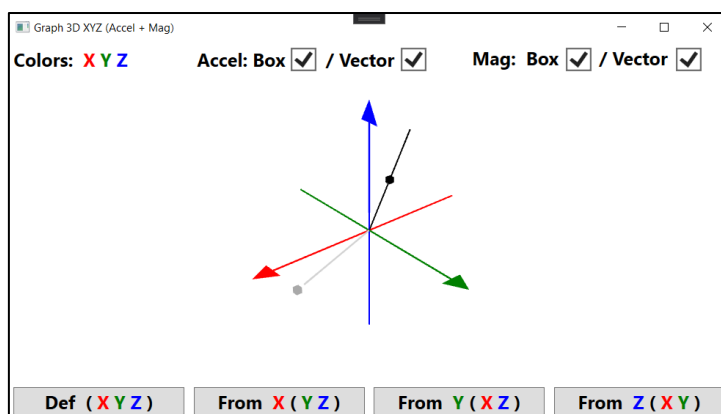


Obrázek 35: Záznam dat z magnetometru při jeho náhodném otáčení (vlevo) a při statické pozici (vpravo).

Záznam všech bodů lze kdykoliv vymazat kliknutím na „Clear“ a body se budou dále vykreslovat znovu na čisté ploše. Pomocí tlačítek „-“ a „+“ lze graf oddálit či přiblížit. Jednotlivé projekce do rovin (barvy) lze skrýt či zviditelnit prostřednictvím zaškrtnutí příslušného políčka v horní části dialogu.

5.6.3 Graph 3D XYZ

Jak již bylo zmíněno dříve, každý dílčí senzor měří fyz. veličiny ve 3D prostoru (hodnoty ve třech osách X, Y a Z), tedy výsledkem jednoho měření / vzorku je vždy tříložkový vektor. V tomto 3D grafu jsou v reálném čase vykreslovány měřená data z akcelerometru a magnetometru právě formou 3D vektoru (vektor zrychlení a vektor mag. intenzity). Pro jednoduchost implementace jsou vektory tvořeny úsečkou a příslušnou krychličkou určující velikost vektoru, jak je vidět na Obrázek 36. Vektor zrychlení je znázorněn černou sytou barvou a magnetický vektor barvou šedou.



Obrázek 36: Graf typu „3D XYZ“ zobrazující vektor zrychlení (černě) a vektor mag. intenzity (šedě).

Tento „graf“ je vynášen do 3D scény, kterou lze libovolně otáčet a přibližovat / oddalovat. V dolní části okna jsou přednastavená tlačítka umožňující nastavit pohled kamery proti směru osy X, Y a Z (projekce do příslušné roviny YZ, XZ či XY). V horní části okna jsou umístěny zaškrťovací boxy pro skrytí krychličky a úsečky (vektoru) jak pro měřený vektor zrychlení „Accel“, tak pro vektor mag. intenzity „Mag“.

5.7 Kalibrace

V této podkapitole pokračujeme s popisem záložky aplikace s názvem „Data“, ale tentokrát se věnujeme zejména sekci označené jako „Kalibrace“ a vše co s ní souvisí. Jak již název napovídá v této části, viditelné na Obrázek 37, je umístěna obsluha pro kalibraci dílčích senzorů a jsou zde zobrazeny i aktuální kalibrační hodnoty.

Kalibrace		
<input type="checkbox"/>	Kalibrace 'Accel'	X
Offset: X=0.001 / Y=0.008 / Z=-0.010		
<input checked="" type="checkbox"/>	Gyro - Start	Vypocet X
Offset: X=0.024 / Y=0.003 / Z=-0.019		
<input checked="" type="checkbox"/>	Mag - Start	Vypocet X
Hard: X=-11.981 / Y=-30.956 / Z=-33.206		
Soft: X=1.017 / Y=0.987 / Z=0.996		

Obrázek 37: Část záložky aplikace „Data“ obsahující tlačítka pro vykonání kalibrace dílčích senzorů.

Zaškrťovací boxy vlevo pro příslušný senzor značí, zda jsou kalibrační hodnoty aplikovány na příchozí měřená data. Nutné je zdůraznit, že zdali je tento box zaškrtnut pro příslušný dílčí senzor, tak jsou ve všech grafech i v textových datech pro odpovídající fyz. veličinu zobrazeny již kalibrované hodnoty. V opačném případě jsou tedy zobrazována pouze příchozí surová data ze senzorů. Tlačítka se symbolem „X“ umístěné vpravo slouží k vynulování odpovídajících kalibračních hodnot (nastavení jejich hodnot na nulu).

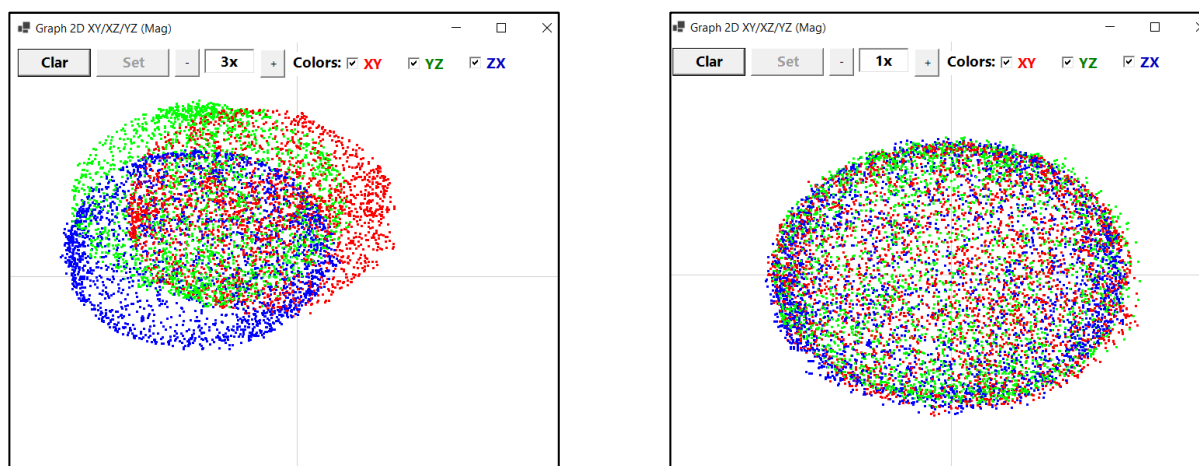
Kompletní kalibraci celého senzoru je samozřejmě nutné vykonat postupně a zvláště pro každý dílčí senzor. Postup provedení **kalibrace akcelerometru** je následující. Senzor je nejprve potřeba umístit na pevnou podložku a to tak, aby osa Z senzoru směřovala směrem vzhůru. Poté stačí pouze kliknout na tlačítko „Kalibrace Accel“ a kalibrační hodnoty jsou okamžitě vypočteny dle postupu popsáno v kapitole 4.6.3. Jak již bylo v této práci zmíněno výše, kalibraci akcelerometru ve většině případů není vůbec potřeba vykonávat.

Obdobný postup je i při **kalibraci gyroskopu**, kde stačí senzor opět nehybně umístit, třeba na pevnou podložku, ale tentokrát lze zvolit libovolnou pozici (natočení) senzoru. Poté stačí kliknout na tlačítko „Gyro – Start“ a dojde k zobrazení čítače vzorků později použitých na výpočet kalib. hodnot. Počkáme například na 1000 naměřených vzorků jako je tomu na Obrázek 38 a pro následný výpočet kalib. hodnot klikneme na tlačítko „Vypocet“. Interní postup tohoto výpočtu byl podrobně popsán v kapitole 4.6.4.



Obrázek 38: Část záložky aplikace „Data“ informující o průběhu kalibrace gyroskopu.

Nejsložitějším procesem kalibrace celého senzoru je **kalibrace magnetometru**, k jejímu průběhu je vhodné si otevřít graf s názvem „Graph 2D XY“. Pro spuštění kalibrace opět stačí kliknout na tlačítko „Mag – Start“ a začnou se načítat / sbírat měřené vzorky. Následně je nutné po určitou dobu otáčet se senzorem ve všech možných směrech, tak abychom dostali dostatečné množství různorodých naměřených dat jako je tomu např. na Obrázek 39 vlevo. (minimálně obrys, ideálně zaplněná plocha!) Zároveň je vhodné / nutné, aby při průběhu kalibrace nebyly na magnetometr aplikovány jiné kalib. hodnoty (box před „Mag – Start“ nebyl zaškrtnut) nebo aby byly momentálně uložené kalib. hodnoty nulové (nulování pomocí tlačítka „X“). Když máme nasbíráno / naměřeno dostatečné množství vzorků (rovnoměrné rozmístění barevných bodů po ploše), tak pro stanovení kalib. hodnot stačí kliknout opět na tlačítko „Vypocet“ a kalibrační hodnoty budou automaticky vypočteny dle postupu popsáno v kapitole 4.6.5.



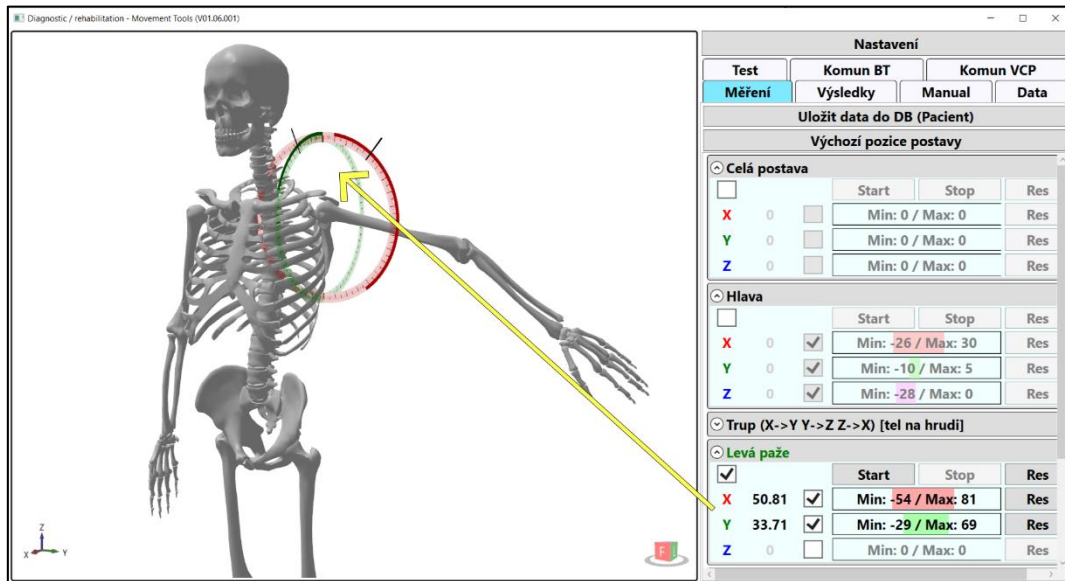
Obrázek 39: Záznam dat z magnetometru při jeho otáčení ve všech směrech a to před kalibrací (vlevo) a po kalibraci (vpravo).

Správnost kalibrace si můžeme ověřit opět pomocí grafu „Graph 2D XY“. V první řadě musí být zaškrtnut box pro aplikaci kalib. hodnot magnetometru a následně začneme otáčet senzorem opět ve všech možných směrech jako dříve při kalibraci. Jestliže byla předchozí kalibrace vykonána správně, měli bychom nyní dostat podobný graf jako na Obrázek 39 vpravo.

5.8 Měření / cvičení

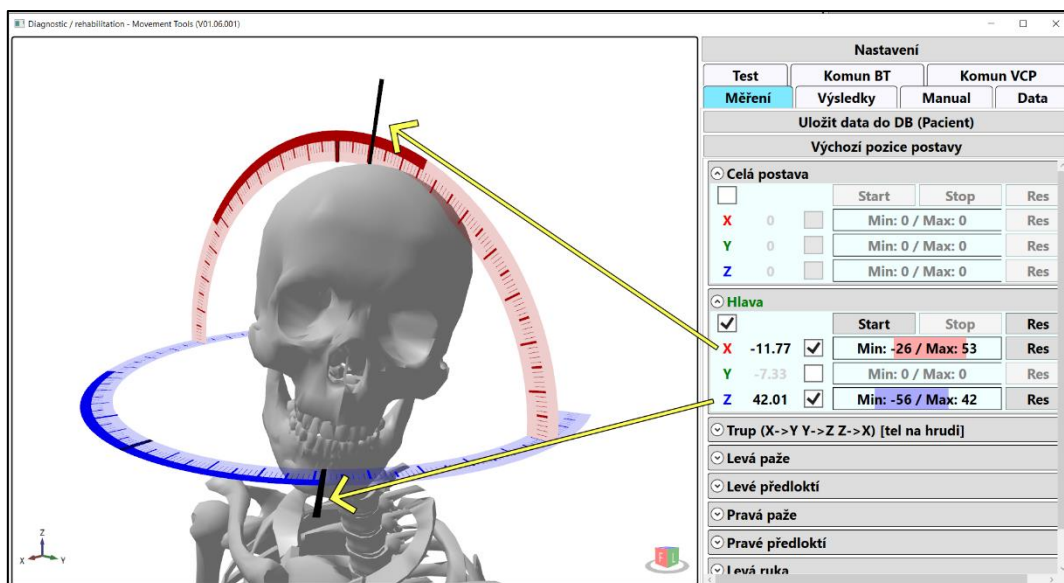
Hlavní částí / účelem této aplikace je možnost objektivně měřit a zobrazit v reálném čase rozsahy pohybů (úhly natočení) dílčích částí těla člověka, k čemuž právě slouží záložka s názvem „Měření“. Zde

jsou předdefinovaná jednotlivá měření / cvičení pro příslušnou část těla člověka, na které lze vhodně fyzicky umístit pohybový senzor (vlastní HW s MPU9250 nebo mobilní telefon). Pro jednotlivá cvičení je vždy zaznamenáván / zobrazován aktuální úhel natočení a minimální / maximální dosažený úhel pro každou osu otáčení (X, Y a Z). Vše je dobře viditelné v textově-grafické formě v oblasti této záložky pod každým názvem odpovídajícího cvičení. Navíc je ve 3D scéně (v levé části okna) příslušný pohyb vykonáván příslušnou částí 3D postavy spolu s plně grafickou interpretací měřených dat formou 3D měřicího úhlooměru jak je patrné z Obrázek 40. V dalším odstavci je podrobně popsán postup, jak takové měření / cvičení vykonat.



Obrázek 40: Záložka aplikace „Měření“ s výběrem aktivního typu cvičení a k tomu odpovídajícím stavem 3D postavy.

V případě již vhodně umístěného senzoru na těle cvičícího a jeho úspěšného připojení k PC (aplikaci) dle postupu podle podkapitoly 5.5 (případně je ještě provedena kalibrace dle 5.7), můžeme spustit požadované cvičení / měření. Nejprve zvolíme dané cvičení jeho zaškrtnutím a tím dojde k zezelenání jeho názvu. Pokud je v nastavení (5.1) zapnuta auto-kalibrace, dochází k odpočtu času, při kterém je dobré se senzorem nehýbat. Pomocí zaškrťovacích boxů pro jednotlivé osy (X, Y a Z) povolíme měření v dané ose a zároveň i zobrazení odpovídajících částí 3D měřicího úhlooměru na dané části postavy. Pro samotné spuštění procesu měření poté stačí kliknout na tlačítko „Start“. Pokud již máme požadované rozsahy pohybu naměřeny, jako tomu může být např. na Obrázek 41, pro ukončení záznamu klikneme na tlačítko „Stop“. V případě, že již nechceme měřit další cvičení (např. jinou část těla), pro uložení naměřených dat do databáze klikneme v horní části záložky na tlačítko „Uložit data do DB (Pacient)“.



Obrázek 41: Záložka aplikace „Měření“ s vyznačením vztahu mezi naměřenými úhly (vpravo v záložce) a hodnotami zobrazenými na úhломěrech (vlevo na postavě).

Pozn. Implementace databáze pro pacienty (cvičící) není součástí této práce! Databázi pro zde popisovanou aplikaci dodal vedoucí práce. Zde uvedené / zobrazené druhy cvičení jsou pouze ilustrativní pro vysvětlení používání aplikace a nemusí tedy být ve skutečnosti účelné, případně ani správné.

Při samotném průběhu měření znázorněném na Obrázek 41 vidíme spuštěné cvičení s názvem „Hlava“. V jeho sekci vedle každého názvu osy (X, Y a Z) je zobrazen aktuální měřený úhel udávaný ve stupních a dále vpravo je vypsán (i graficky znázorněn) doposud dosažený minimální a maximální úhel pro danou osu. Tyto měřené hodnoty lze rovněž pozorovat i v levé části okna na modelu postavy prostřednictvím měřicího 3D úhlooměru. Krátká černá čárka na stupnici úhlooměru značí nulu (počátek), nejmenší dílky stupnice odpovídají kroku hodnotě 1°, větší 5° a další 10°. Aktuální hodnota měřeného úhlu je reprezentována dlouhou černou čárkou (jakoby hodinovou ručičkou). Rozsahu min. a max. dosaženého úhlu odpovídá délka plně barevné (neprůhledné) výšeče kružnice na úhlooměru.

Naměřené hodnoty pro aktuální cvičení lze průběžně libovolně nulovat (resetovat) a to buď pomocí tlačítek „Res“ vpravo pro každou osu zvlášť nebo tlačítkem „Res“ nad nimi sloužící pro nulování všech tří os současně. Pokud však chceme resetovat celý postoj postavy, je potřeba kliknout v horní části záložky na tlačítko „Výchozí pozice postavy“.

Pozn.: Měření (záložka „Měření“) lze ve skutečnosti používat i bez připojeného senzoru. A to prostřednictvím simulace pohybu postavy pomocí záložky s názvem „Manual“ viz. 5.4.

5.9 Výsledky

Aby bylo možné dosažené výsledky jednotlivých měření / cvičení (vykonaná dle kapitoly 5.8) v průběhu času (např. jednou týdně) analyzovat, a tím sledovat progres cvičícího / pacienta, je součástí aplikace i záložka s názvem „Výsledky“. Průběžné výsledky měření jsou nejen zobrazeny v textové formě, ale i vhodně graficky – formou tabulky či grafu, tak aby bylo možno snadno posoudit zlepšení případně zhoršení cvičícího. Toto je vhodné ať už pro odborný personál (např. rehabilitační pracovník), či také přímo pro cvičícího / pacienta sloužící například pro jeho motivaci do dalšího cvičení.

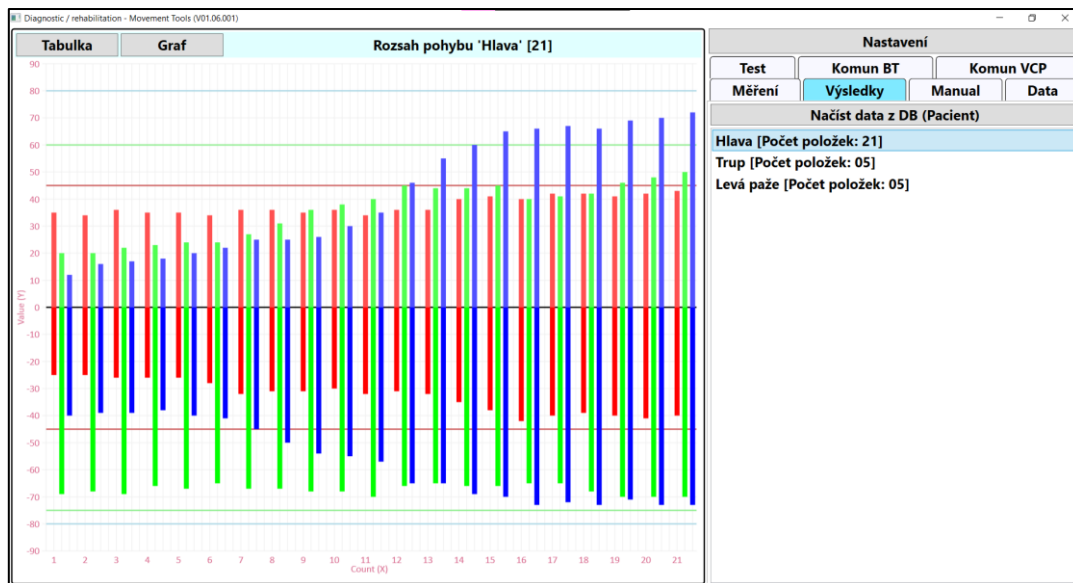
Při otevření záložky „Měření“ zobrazené na Obrázek 42 se v pravé části nachází tlačítko „Načíst data z DB (Pacient)“ sloužící pro načtení dat od požadovaného pacienta z databáze (implementace databáze

není součástí této práce). Dále se v pravé části v seznamu nachází záznamy jednotlivých cvičení (pro určité části těla vybraného pacienta) spolu s informací kolik obsahují položek / dílčích záznamů. Po výběru požadovaného záznamu cvičení se nám v levé části okna zobrazí data formou tabulky obsahující pořadové číslo, datum záznamu a odpovídající minimální a maximální dosažené úhly natočení ve všech třech osách (X, Y a Z) spolu s grafickou reprezentací rozsahu pohybu formou délky barevného pruhu. Jinak řečeno maximální úhel natočení na jednu (záporné hodnoty) a na druhou stranu (kladné hodnoty) od výchozí / nulové pozice.

Tabulka		Rozsah pohybu 'Hlava' [21]		
Index	Date	Angle X	Angle Y	Angle Z
1	2021-10-01	Min: -25 / Max: 35	Min: -69 / Max: 20	Min: -40 / Max: 12
2	2021-10-04	Min: -25 / Max: 34	Min: -68 / Max: 20	Min: -39 / Max: 16
3	2021-10-05	Min: -26 / Max: 36	Min: -69 / Max: 22	Min: -39 / Max: 17
4	2021-10-06	Min: -26 / Max: 35	Min: -66 / Max: 23	Min: -38 / Max: 18
5	2021-10-07	Min: -26 / Max: 35	Min: -67 / Max: 24	Min: -40 / Max: 20
6	2021-10-08	Min: -28 / Max: 34	Min: -65 / Max: 24	Min: -41 / Max: 22
7	2021-10-11	Min: -32 / Max: 36	Min: -67 / Max: 27	Min: -45 / Max: 25
8	2021-10-12	Min: -31 / Max: 36	Min: -67 / Max: 31	Min: -50 / Max: 25
9	2021-10-13	Min: -31 / Max: 35	Min: -68 / Max: 36	Min: -54 / Max: 26
10	2021-10-14	Min: -30 / Max: 36	Min: -68 / Max: 38	Min: -55 / Max: 30
11	2021-10-15	Min: -32 / Max: 34	Min: -70 / Max: 40	Min: -57 / Max: 35
12	2021-10-18	Min: -31 / Max: 36	Min: -66 / Max: 45	Min: -65 / Max: 46
13	2021-10-19	Min: -32 / Max: 36	Min: -65 / Max: 44	Min: -65 / Max: 55
14	2021-10-20	Min: -35 / Max: 40	Min: -66 / Max: 44	Min: -69 / Max: 60
15	2021-10-21	Min: -38 / Max: 41	Min: -66 / Max: 45	Min: -70 / Max: 65
16	2021-10-22	Min: -42 / Max: 40	Min: -65 / Max: 40	Min: -73 / Max: 66
17	2021-10-25	Min: -40 / Max: 42	Min: -65 / Max: 41	Min: -72 / Max: 67
18	2021-10-26	Min: -39 / Max: 42	Min: -68 / Max: 42	Min: -73 / Max: 66

Obrázek 42: Záložka aplikace „Výsledky“ poskytující zobrazení výsledků průběžných měření formou grafické tabulky.

Druhou možností zobrazení výsledků měření / cvičení je forma grafu viditelná na Obrázek 43, do které se přepneme kliknutím na tlačítko „Graf“ v levém horním rohu. Na vodorovné ose grafu jsou vynášena pořadová čísla jednotlivých záznamů měření a svislá osa představuje hodnoty dosažených úhlů ve stupních. Pro každý uložený záznam měření jsou tedy do grafu vyneseny tři svislé čáry (červeně – osa otáčení X, zeleně – Y a modře – Z) reprezentující naměřený rozsah pohybu – od min. úhlu natočení po max. úhel natočení (otočení na jednu a na druhou stranu). Barevně odlišené vodorovné čáry (světlejší odstín červené, zelené a modré) značí optimální rozsah pohybu části těla pro odpovídající úhel / směr jejího otáčení. Pomocí kolečka myši lze graf přibližovat a libovolným tlačítkem myši graf posouvat, a to vše pouze ve vodorovném směru. Mezi formou zobrazení záznamu / výsledků je možné libovolně přepínat odpovídajícími tlačítky „Tabulka“ / „Graf“.



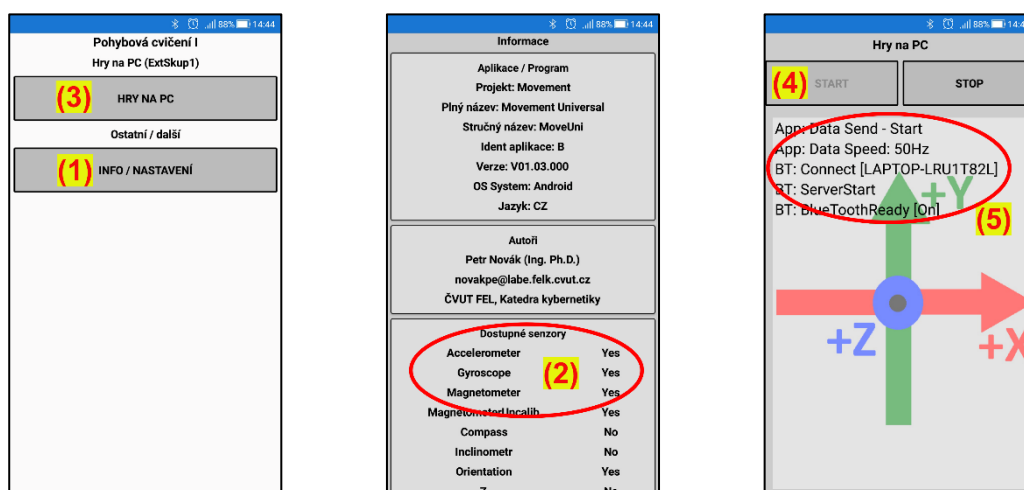
Obrázek 43: Zálůžka aplikace „Výsledky“ poskytující zobrazení výsledků průběžných měření formou grafu.

6 Příklad použití aplikace

V této kapitole popíšeme postup, jak prakticky uskutečnit cvičení / měření určité části lidského těla. Pro jednoduchost a přehlednost se zaměříme pouze na jednu část těla a to hlavu. Při měření ostatních částí těla by se postupovalo v podstatě stejně. Jako pohybový senzor použijeme mobilní telefon s příslušnou aplikací a připojený k počítači přes Bluetooth. Nejprve popíšeme spuštění aplikace na mobilu a navázání spojení s počítačem / aplikací, následně zmíníme různé možnosti nastavení aplikace, poté popíšeme správné umístění senzoru na hlavě, a nakonec samotné vykonání měření s interpretací zjištěných rozsahů pohybu.

Postup:

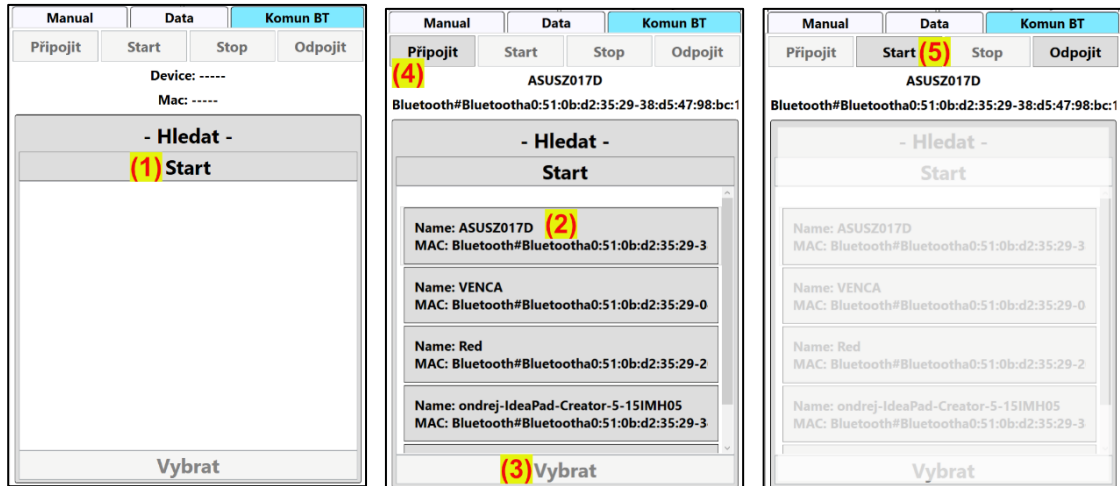
- Nejprve spustíme na mobilním telefonu Bluetooth a mobilní aplikaci s názvem „MovementUni“ viditelnou na Obrázek 44. Pro ověření, zda náš telefon poskytuje veškeré potřebné senzory klikneme na „INFO / NASTAVENÍ“ (1) a zobrazí se nám dostupné senzory v zařízení. Důležité je, aby u prvních třech senzorů (Accel., Gyro., Mag.) bylo uvedeno „Yes“ (2). Následně se vrátíme zpět a klikneme na položku „HRY NA PC“ (3), kde je potřeba pro zahájení vysílání dat podržet tlačítko „START“ (4) po dobu 3 až 5 sekund. Veškeré provedené komunikační pokyny jsou dále vypisovány na obrazovku (5). Kromě toho jsou zde barevně zobrazeny i jednotlivé souřadnicové osy sloužící například ke správnému umístění telefonu na těle cvičícího.



Obrázek 44: Postup použití mobilní aplikace „MovementUni“ využívající (běžný) mobilní telefon jako pohybový senzor.

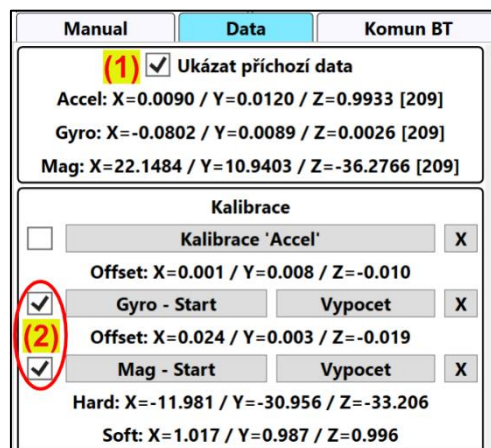
- Dále zapneme Bluetooth na PC a spustíme vytvořenou aplikaci na PC (MS Windows). Vybereme odpovídající profil pro náš senzor (mobilní telefon). V případě, nejen prvního spuštění aplikace, můžeme postupovat dle podrobného popisu uvedeného v podkapitole 5.2.2.
- V nastavení aplikace můžeme zvolit libovolně typ postavy k následné reprezentaci měřeného pohybu a její nasvícení, rychlost dat ze senzorů, frekvenci zobrazení, příjem již kalibrovaných dat z magnetometru, vhodný algoritmus na zpracování dat a případně auto-kalibraci. Podrobnější informace k jednotlivým položkám nastavení byly zmíněny již v podkapitole 5.1. V našem testovacím případě jsme zvolili rychlost dat pro všechny senzory na 50 Hz a jako algoritmus byl zvolen „Madgwick filtr“, tedy je nutné zapnout i auto-kalibraci. Položku „Přijem kalib. dat pro Mag (mobil)“ jsme neaktivovali.

- Pro připojení senzoru (mobilního telefonu) k naší aplikaci využijeme záložku „Komun BT“ viditelnou na Obrázek 45. Zde nejprve klikneme na „Start“ (1) pro hledání zařízení a vybereme naše Bluetooth zařízení (2) a dáme „Vybrat“ (3). Poté klikneme na „Připojit“ (4) a nakonec na „Start“ (5).



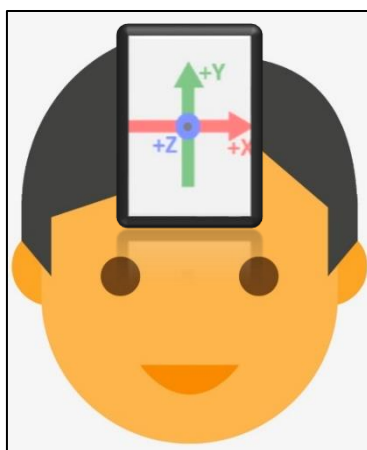
Obrázek 45: Postup navázání Bluetooth komunikace s pohybovým senzorem prostřednictvím aplikace na PC.

- Přicházející data ze senzoru si můžeme ověřit v záložce „Data“, viditelné na Obrázek 46, povolením položky „Ukázat příchozí data“ (1). Dále nezapomeneme zkontrolovat, zda je povolena aplikace kalibračních hodnot (2), a to hlavně pro gyroskop a magnetometr (V „Nastavení“ nesmí být tedy povolena položka „Příjem kalib. dat pro Mag (mobil)“!). Pokud není senzor ještě zkalibrován je vhodné nejprve vykonat kalibraci dle podkapitoly 5.7.



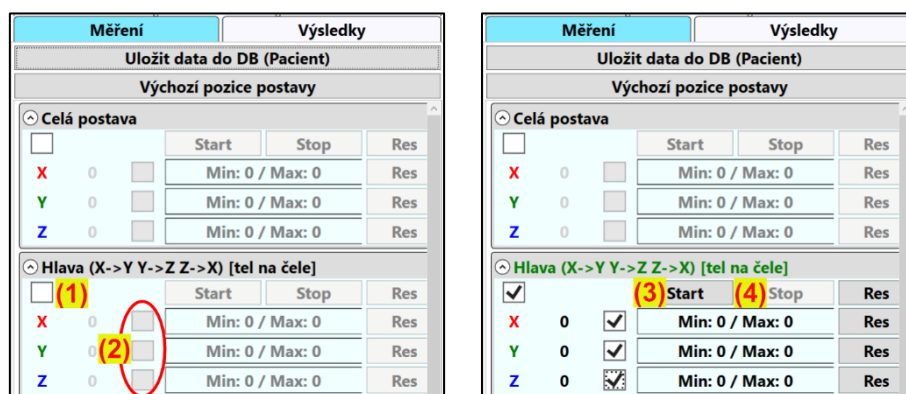
Obrázek 46: Příklad doporučeného nastavení záložky „Data“ aplikace na PC před zahájením měření.

- Senzor (v tomto případě mobilní telefon) připevníme na hlavu cvičícího ve správné orientaci, jak je znázorněno na Obrázek 47. Jedná se zde samozřejmě o názornou ukázkou.



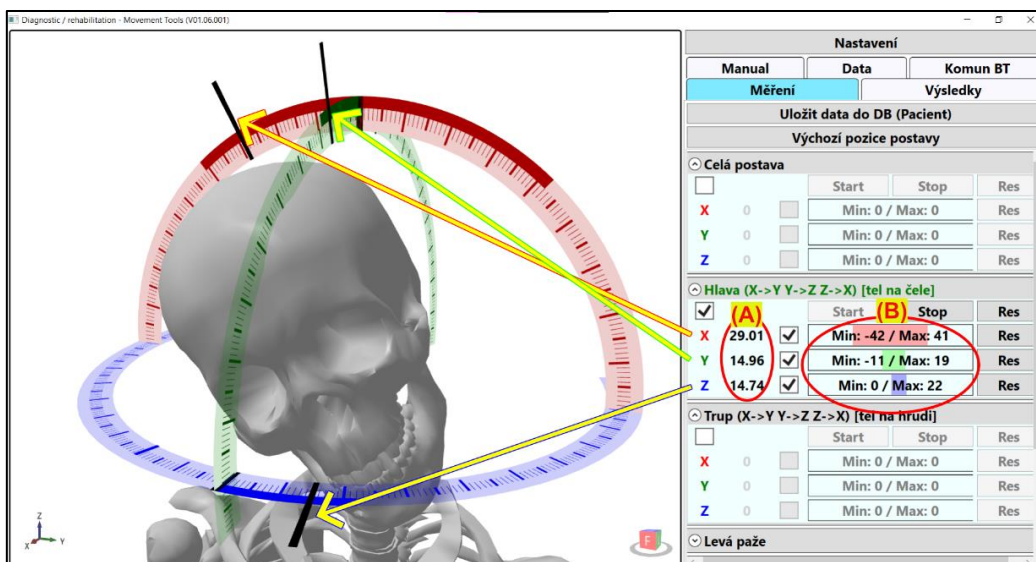
Obrázek 47: Znárodnění správného umístění pohybového senzoru (mobilního telefonu) na hlavě cvičícího.

- Nyní již konečně můžeme přejít do záložky „Měření“, viditelné na Obrázek 48, pro zaznamenání samotného cvičení. Nejprve zaškrtneme cvičení s názvem „Hlava“ (1) a hned po odpočtu času dojde k vykonání auto-kalibrace (nehýbat se senzorem!). Poté povolíme měření, čímž zobrazíme úhlooměry ve všech třech osách měření X, Y a Z (2). Ještě si můžeme libovolně myší nastavit pohled kamerou, abychom co nejlépe viděli na měřenou část 3D postavy a to v tomto případě na hlavu. Nyní je již vše připraveno pro zahájení měření.



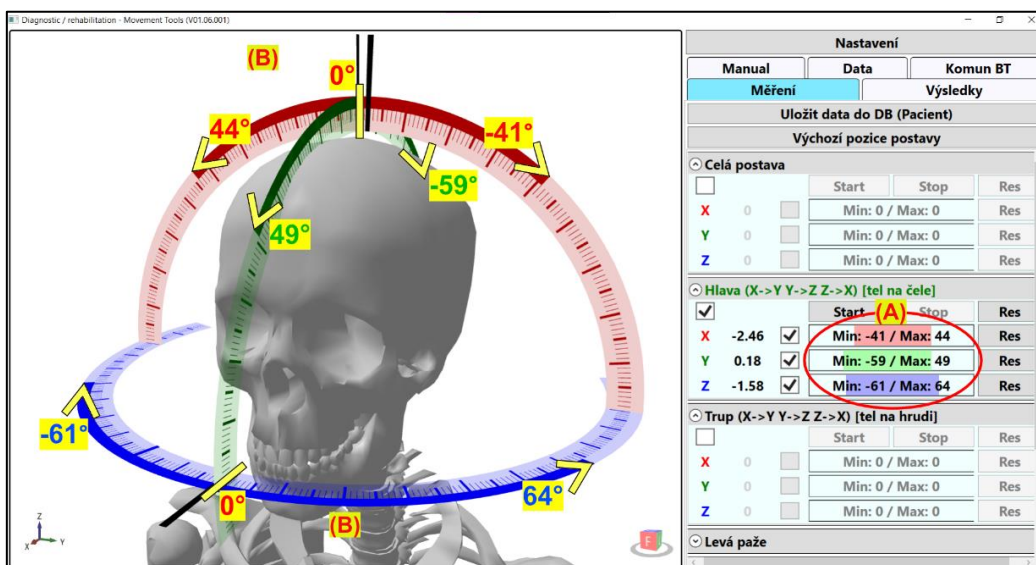
Obrázek 48: Postup spuštění jednoho měření / cvičení s názvem „Hlava“ v aplikaci na PC.

- Samotné měření spustíme tlačítkem „Start“ (3). V našem demonstračním příkladu cvičící vykonal maximální úklon hlavou doprava a doleva, pak maximální předklon a záklon hlavy, a nakonec maximální otočení hlavy doprava a doleva. Zaznamenali jsme tedy rozsahy / úhly natočení ve všech třech osách měření X, Y a Z. Záznam měření ukončíme tlačítkem „Stop“ (4).
- Při samotném průběhu měření, viditelném na Obrázek 49, můžeme sledovat aktuální úhly natočení (A) pro každou povolenou osu (X, Y a Z), a to buď v číselné formě nebo přímo na 3D postavě grafickou formou na zobrazených měřících úhloměrech. Zároveň můžeme sledovat i momentálně dosažený rozsah pohybu (min. a max. úhel) ve všech povolených osách (B).



Obrázek 49: Stav aplikace na PC v průběhu měření s vyznačením vztahů mezi číselnými hodnotami (záložka vlevo) a 3D reprezentací (plocha vpravo).

- Po ukončení měření lze podrobně analyzovat výsledné naměřené rozsahy pohybu / natočení hlavy viditelné na Obrázek 50. Těmito rozsahy rozumíme absolutní maximální úhly natočení hlavy, a to ve všech třech osách, kde pro každou osu je maximální úhel natočení na jednu stranu označen se znaménkem mínus (tedy jakoby minimum) a natočení na opačnou stranu s kladným znaménkem (tedy max.). V našem případě jsme naměřili rozsahy (A) pro osu X: min -41° a max 44° , pro osu Y: min -59° a max 49° , pro osu Z: min -61° a max 64° . Tyto hodnoty jsou zobrazeny nejen číselně, ale i graficky na 3D úhломěrech (B).



Obrázek 50: Stav aplikace na PC po dokončeném měření s vysvětlivky dosažených úhlů.

- Kdybychom toto měření vždy po určitém čase (např. jednou za týden) opakovali, vytvořené záznamy měření bychom ukládali do databáze a poté v záložce aplikace „Výsledky“, lze vše sledovat v čase. Jelikož jsme však vykonali pouze jedno měření, nemá se tedy smysl s touto záložkou aplikace zde zaobírat.

Závěr

Úkolem práce bylo navrhnout a vytvořit zařízení (případně minimální systém) pro experimenty vhodné při diagnostice a rehabilitaci pohybů dílčích částí lidského těla. Takovýto úkol není nikterak jednoduchý, neboť se skládá ze skutečně mnoha komponent, které nejsou na první pohled vůbec zřejmé.

Nejprve byl vybrán vhodný pohybový senzor, a to ve dvou variantách. První jako velmi dostupný i na úkor přesnosti, a to využití běžného chytrého mobilního telefonu, hodícího se zejména pro cvičení neboli rehabilitaci. Druhý jako malé zařízení využívající samostatný pohybový senzor (akcelerometr, gyroskop, magnetometr) spolu s jednočipovým procesorem. Ten je určen pro přesnější měření, tedy zejména pro diagnostiku. Zmíněné senzory byly navrženy pouze pro sběr pohybových dat, která jsou přenášena do blízkého počítače za účelem jejich dalšího zpracování. Jsou tedy určeny pro uchycení na libovolnou část těla pro měření rozsahu jejího pohybu.

Hlavní částí práce je však vytvořená aplikace pro operační systém Microsoft Windows. Nejedná se však o aplikaci určenou pro profesionální použití, ale právě pro experimenty s měřením pohybu. Ta obsahuje několik interních částí, které lze však dále podle vlastní potřeby zcela libovolně rozšiřovat. První část tvoří příjem dat z externích pohybových senzorů přes USB / Bluetooth (další typy komunikací lze doplnit). Druhá část je určena pro zobrazení přijímaných dat za účelem například jejich kontroly. Zde jsou data dostupná v běžné textové podobě nebo formou několika různých a přehledných grafů (obě možnosti jsou velmi často vhodné / potřebné, zejména při vývoji vlastního pohybového senzoru). Další část je určena pro kalibraci použitých senzorů. Toto je vhodné zejména při použití pohybových senzorů zcela vlastní konstrukce, které je nutno kalibrovat před jejich skutečným měřicím použitím. Kalibrace je vykonávána sice manuálně, ale je pro toto vytvořena vhodná podpora (pro magnetometr i grafická). Čtvrtá, a jedna z největších částí, obsahuje algoritmy / filtry pro zpracování dat z pohybového senzoru. Byly využity již existující a odladěné algoritmy využívající princip tzv. komplementárního filtru. Základní / jednoduché algoritmy pracující na tomto principu, jehož základ je obecně znám, byly vlastnoručně upraveny a poté v aplikaci implementovány („Dirty“ komplementární filtr, lineární kompl. filtr, nelineární kompl. filtr), avšak složitější, přesnější a již dostatečně odladěné algoritmy byly plně převzaty (Mahony filtr, Madgwick filtr). Cílem práce nebylo tyto algoritmy vymýšlet / vytvářet (toto by bylo velmi přes rozsah této práce), ale vhodně je převzít a využít. Další lze do aplikace samozřejmě později vložit. Pátá část představuje 3D zobrazení lidského těla formou několika možných postav / modelů. Data z pohybového senzoru, následně zpracovaná vloženým algoritmem, jsou uživateli prezentována nejen číselně / textově (vypočtené hodnoty), ale i formou pohybu částí těla zvolené 3D postavy. Jedná se skutečně o velmi přehlednou vizualizaci jak pro nezavěšeného člověka, tak i pro odborníka. Měřené pohyby jednotlivých částí těla jsou prezentovány pomocí sofistikovaných 3D úhломěrů, a to jak aktuální poloha / úhel, tak i celkový rozsah pohybu pro danou část těla. Předposlední část tvoří seznam dílčích typů cvičení / měření určených pro konkrétní části těla. Zde lze povolit, nejen pro jakou část těla jsou určena data z pohybového senzoru, ale rovněž jaké osy pohybu jsou pro tuto část postavy povoleny. Poslední díl aplikace zajišťuje prezentaci naměřených dat, tedy zejména detekovaného rozsahu pohybu jednotlivých částí těla, a tak možnost následné vlastní analýzy naměřených výsledků uživatelem. Výsledky lze zobrazit buď formou částečně grafické tabulky, nebo pomocí speciálně navrženého grafu.

Z předešlého je zřejmé, jaký velký rozsah možností aplikace poskytuje právě pro experimenty v oblasti detekce a rehabilitace pohybu dílčích částí těla člověka. Výsledkem práce je tedy soubor nástrojů (pohybový senzor, přenos dat a aplikace) určených právě pro experimentální využití nejen ve výukovém a výzkumném, ale i přímo v medicínském prostředí. Možnost jakéhokoli rozšíření aplikace (algoritmy, zobrazení dat, ...), využití libovolného pohybového senzoru a zejména přístup k surovým měřeným datům dělají výstup této práce velmi hodnotným.

Reference

- [1] KIM, Won-Seok, Sungmin CHO, Jeonghun KU, Yuhee KIM, Kiwon LEE, Han-Jeong HWANG a Nam-Jong PAIK. Clinical Application of Virtual Reality for Upper Limb Motor Rehabilitation in Stroke: Review of Technologies and Clinical Evidence. *Journal of Clinical Medicine* [online]. 2020, **9**(10) [cit. 2021-12-20]. ISSN 2077-0383. Dostupné z: doi:10.3390/jcm9103369
- [2] Armeo®Power - Hocoma. *Advanced Technologies for Movement Rehabilitation - Hocoma* [online]. Hocoma, 2021 [cit. 2021-12-20]. Dostupné z: <https://www.hocoma.com/solutions/armeopower/>
- [3] Lokomat Pro 3D - Steps Rehabilitation Center. *STEPS Center for Robotic Rehabilitation in Budapest, Hungary* [online]. Budapest, 2021 [cit. 2021-12-20]. Dostupné z: <https://stepsbudapest.com/lokomat-pro-3d/>
- [4] MILOSEVIC, Bojan, Alberto LEARDINI a Elisabetta FARELLA. Kinect and wearable inertial sensors for motor rehabilitation programs at home: state of the art and an experimental comparison. *BioMedical Engineering OnLine* [online]. 2020, **19**(1) [cit. 2021-12-20]. ISSN 1475-925X. Dostupné z: doi:10.1186/s12938-020-00762-7
- [5] Motion capture set to become more mainstream in medical science predicts Vicon. *Cision News* [online]. 2015 [cit. 2021-12-20]. Dostupné z: <https://news.cision.com/vicon/r/motion-capture-set-to-become-more-mainstream-in-medical-science-predicts-vicon,c9722569>
- [6] Present - The Vicon MX System, in the Performance Capture Lab in the Robert Zemec. *Pinterest* [online]. Mansi, 2012 [cit. 2021-12-20]. Dostupné z: <https://www.pinterest.com/pin/324751823101082429/>
- [7] Functional Training Game | Full body training with Virtual Reality for everyone!. *Functional Training Game | Full body training with Virtual Reality for everyone!* [online]. Dánsko [cit. 2021-12-20]. Dostupné z: <https://doctorkinetic.com>
- [8] Rehabilitation and physiotherapy with games for elderly - SilverFit. *SilverFit: Improving Elderly Care - SilverFit* [online]. c2008-2021 [cit. 2021-12-20]. Dostupné z: <https://silverfit.com/en/products/silverfit-3d-camera>
- [9] Motion Capture. *Home - Xsens 3D motion tracking* [online]. [cit. 2021-12-20]. Dostupné z: <https://www.xsens.com/motion-capture>
- [10] *The definition of the 23 segments in the kinematic model of Xsens MVN.... | Download* [online]. 2017 [cit. 2021-12-20]. Dostupné z: https://www.researchgate.net/figure/The-definition-of-the-23-segments-in-the-kinematic-model-of-Xsens-MVN-An-inertial_fig2_312036960
- [11] Armeo®Senso - Hocoma. *Advanced Technologies for Movement Rehabilitation - Hocoma* [online]. Hocoma, 2021 [cit. 2021-12-20]. Dostupné z: <https://www.hocoma.com/solutions/armeosenso/>
- [12] Interactive Hand-Arm-Rehabilitation | PABLO® | Tyromotion. *Advanced Rehabilitation Technology | Tyromotion Austria* [online]. [cit. 2021-12-20]. Dostupné z: <https://tyromotion.com/en/products/pablo/>

- [13] Bimeo PRO neurological rehabilitation / stroke - Kinestica - motivating neurological. *Kinestica neurological rehabilitation stroke motivation - Kinestica* [online]. Kinestica, 2021 [cit. 2021-12-20]. Dostupné z: <https://www.kinestica.com/bimeo-pro.html>
- [14] Moover™ - Sensor Medica. *Homepage - Sensor Medica* [online]. Sensor Medica, 2021 [cit. 2021-12-20]. Dostupné z: <https://www.sensormedica.com/en/moover/>
- [15] Kardio - Line spol. s r.o. - detailní popis produktu: MOOVER. *Kardioline – prodej a servis zdravotnických, wellness a fitness přístrojů.* [online]. Kardio - Line spol. s r.o. [cit. 2021-12-20]. Dostupné z: https://www.kardioline.cz/produkty/rehabilitace_diagnostika/moover/
- [16] DyCare: Lynx. *DyCare: clinical assessment by wearable sensors solutions* [online]. Barcelona: DyCare, 2021 [cit. 2021-12-20]. Dostupné z: <https://www.dycare.com/lynx/>
- [17] Riablo | CoRehab. *CoRehab - evolve movement together | CoRehab* [online]. Itálie: CoRehab [cit. 2021-12-20]. Dostupné z: <https://www.corehab.it/en/riablo-bf/>
- [18] Kari | CoRehab. *CoRehab - evolve movement together | CoRehab* [online]. Itálie, 2021 [cit. 2021-12-20]. Dostupné z: <https://www.corehab.it/en/kari-app/>
- [19] Telerehabilitation - BTS TELEREHAB. *Home - BTS TELEREHAB* [online]. Itálie: BTS Bioengineering S.p.A., 2021 [cit. 2021-12-20]. Dostupné z: <https://www.btstelerehab.com/en/telerehabilitation/>
- [20] SWORD Health | A better way to treat MSK pain. *SWORD Health | A better way to treat MSK pain* [online]. SWORD Health, 2021 [cit. 2021-12-20]. Dostupné z: <https://swordhealth.com>
- [21] IANCULESCU, Marilena, Bogdan ANDREI a Adriana ALEXANDRU. A Smart Assistance Solution for Remotely Monitoring the Orthopaedic Rehabilitation Process Using Wearable Technology: re.flex System. *Studies in Informatics and Control* [online]. 2019, **28**(3) [cit. 2021-12-20]. ISSN 12201766. Dostupné z: doi:10.24846/v28i3y201908
- [22] Reflex. *Reflex* [online]. REFLEX, 2021 [cit. 2021-12-20]. Dostupné z: <http://reflex.help/>
- [23] How it works - BPMpathway. *The proven dimension in remote patient assessment - BPMpathway* [online]. BPMpathway, c2011-2021 [cit. 2021-12-20]. Dostupné z: <https://www.bmpathway.com/how-it-works/>
- [24] BELL, Kevin, Chukwudi ONYEUKWU, Michael MCCLINCY et al. Verification of a Portable Motion Tracking System for Remote Management of Physical Rehabilitation of the Knee. *Sensors* [online]. 2019, **19**(5) [cit. 2021-12-20]. ISSN 1424-8220. Dostupné z: doi:10.3390/s19051021
- [25] KUIPERS, Jack B. *Quaternions and rotation sequences: a primer with applications to orbits, aerospace, and virtual reality.* Princeton, N.J.: Princeton University Press, 1999. ISBN 0691058725.
- [26] Euler angles - Wikipedia. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2021-12-21]. Dostupné z: https://en.wikipedia.org/wiki/Euler_angles
- [27] Gimbal lock - Wikipedia. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2021-12-21]. Dostupné z: https://en.wikipedia.org/wiki/Gimbal_lock

- [28] O.H. MADGWICK, Sebastian. *An efficient orientation filter for inertial and inertial/magnetic sensor arrays* [online]. [cit. 2021-12-21]. Dostupné z: https://x-io.co.uk/downloads/madgwick_internal_report.pdf
- [29] COLTON, Shane. *The Balance Filter: A Simple Solution for Integrating Accelerometer and Gyroscope Measurements for a Balancing Platform* [online]. In: . 2007 [cit. 2021-12-21]. Dostupné z: <https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxteWltdWVzdGltYXRpb25leHBlcmllbmNlfGd4OjY1Yzk3YzhiZmE1N2M4Y2U>
- [30] Reading a IMU Without Kalman: The Complementary Filter | Pieter-Jan.com. *Pieter-Jan.com | Creativity In Automation & More* [online]. 2013 [cit. 2021-12-21]. Dostupné z: <https://www.pieter-jan.com/node/11>
- [31] KOTTATH, Rahul, Parag NARKHEDE, Vipin KUMAR, Vinod KARAR a Shashi PODDAR. Multiple Model Adaptive Complementary Filter for Attitude Estimation. *Aerospace Science and Technology* [online]. 2017, **69**, 574-581 [cit. 2021-12-21]. ISSN 12709638. Dostupné z: doi:10.1016/j.ast.2017.07.011
- [32] MAHONY, Robert, Tarek HAMEL a Jean-Michel PFLIMLIN. Nonlinear Complementary Filters on the Special Orthogonal Group. *IEEE Transactions on Automatic Control* [online]. 2008, **53**(5), 1203-1218 [cit. 2021-12-21]. ISSN 0018-9286. Dostupné z: doi:10.1109/TAC.2008.923738
- [33] KALMAN, R. E. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering* [online]. 1960, **82**(1), 35-45 [cit. 2021-12-21]. ISSN 0021-9223. Dostupné z: doi:10.1115/1.3662552
- [34] High-pass filter - Wikipedia. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2021-12-21]. Dostupné z: https://en.wikipedia.org/wiki/High-pass_filter
- [35] Low-pass filter - Wikipedia. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2021-12-21]. Dostupné z: https://en.wikipedia.org/wiki/Low-pass_filter
- [36] EUSTON, M., P. COOTE, R. MAHONY, JONGHYUK KIM a T. HAMEL. A complementary filter for attitude estimation of a fixed-wing UAV. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems* [online]. IEEE, 2008, s. 340-345 [cit. 2021-12-21]. ISBN 978-1-4244-2057-5. Dostupné z: doi:10.1109/IROS.2008.4650766
- [37] YOO, Tae Suk, Sung Kyung HONG, Hyok Min YOON a Sungsu PARK. Gain-Scheduled Complementary Filter Design for a MEMS Based Attitude and Heading Reference System. *Sensors* [online]. 2011, **11**(4), 3816-3830 [cit. 2021-12-21]. ISSN 1424-8220. Dostupné z: doi:10.3390/s110403816
- [38] O.H. MADGWICK, Sebastian. Open source AHRS with x-IMU – x-io Technologies. In: *X-io Technologies* [online]. x-io Technologies, 2011 [cit. 2021-12-21]. Dostupné z: <https://x-io.co.uk/open-source-ahrs-with-x-imu/>
- [39] Parameters and calibration of a low-g 3-axis accelerometer. In: *Home - STMicroelectronics* [online]. STMicroelectronics, 2021 [cit. 2021-12-22]. Dostupné z: <https://www.st.com/resource>

/en/application_note/dm00119044-parameters-and-calibration-of-a-lowg-3axis-accelerometer-stmicroelectronics.pdf

- [40] How do you calibrate an accelerometer?. In: *Science Probeware & Experiment Software for Teachers | Vernier* [online]. Beaverton USA, 2021 [cit. 2021-12-22]. Dostupné z: <https://www.vernier.com/til/418>
- [41] Magnetometer Calibration - MATLAB & Simulink. *MathWorks - Makers of MATLAB and Simulink - MATLAB & Simulink* [online]. The MathWorks, c1994-2021 [cit. 2021-12-22]. Dostupné z: <https://www.mathworks.com/help/fusion/ug/magnetometer-calibration.html>
- [42] WINER, Kris. Simple and Effective Magnetometer Calibration. In: *GitHub: Where the world builds software · GitHub* [online]. GitHub, Inc., 2021 [cit. 2021-12-22]. Dostupné z: <https://github.com/kriswiner/MPU6050/wiki/Simple-and-Effective-Magnetometer-Calibration>
- [43] TUUPOLA, Mika. How to Calibrate a Magnetometer?. In: *Mika Tuupola* [online]. 2020 [cit. 2021-12-22]. Dostupné z: <https://appelsiini.net/2018/calibrate-magnetometer/>
- [44] Calibration Instructions. In: *Gulf Coast Data Concepts, LLC* [online]. Gulf Coast Data Concepts, LLC, c2007-2021 [cit. 2021-12-22]. Dostupné z: http://www.gcdataconcepts.com/Calibration_Instructions.pdf
- [45] PRAYUDI, Iman a Doik KIM. Design and implementation of IMU-based human arm motion capture system. In: *2012 IEEE International Conference on Mechatronics and Automation* [online]. IEEE, 2012, s. 670-675 [cit. 2021-12-22]. ISBN 978-1-4673-1278-3. Dostupné z: doi:10.1109/ICMA.2012.6283221

Seznam zkratek

USB – Universal Serial Bus

USART – Universal Synchronous/Asynchronous Receiver/Transmitter

VCP – Virtual Communication Port

ARM – Advanced RISC Machines

MEMS – Microelectromechanical systems

IMU – Inertial Measurement Unit

MARG – Magnetic, Angular Rate, and Gravity

AHRS – Attitude and Heading Reference Systems

DOF – Degree of Freedom

LP – Low Pass

HP – High Pass

GUI – Graphical User Interface

WPF – Windows Presentation Foundation

Seznam obrázků

Obrázek 1: Exoskeletické přístroje firmy Hocoma.	3
Obrázek 2: Kamerový systém Vicon při použití.....	4
Obrázek 3: Příklad používání rehabilitačních produktů: DoctorKinect a SilverFit.....	4
Obrázek 4: IMU systém firmy Xsens tvořený oblekem a softwarem.	5
Obrázek 5: Příklad použití rehabilitačního produktu Armeo Senso.	6
Obrázek 6: Rehabilitačního produkt Pablo a příklad jeho použití.....	6
Obrázek 7: Dva příklady použití rehabilitačního produktu Bimeo Pro.....	7
Obrázek 8: Princip použití produktu Moover a jeho obslužného programu.....	7
Obrázek 9: Ukázka použití produktu Lynx ROM s obslužným programem.	8
Obrázek 10 Příklad cvičení s produktem Riablo a ukázka obslužného programu.....	8
Obrázek 11: Příklady použití produktu Telerehab.....	9
Obrázek 12: Grafické znázornění využití produktu Sword health.....	9
Obrázek 13: Grafické znázornění využití produktu ReFlex.....	10
Obrázek 14: Ukázka z programu pro cvičení se senzorem produktu BPMpathway.	10
Obrázek 15: Grafické znázornění použití zařízení InterAction.	11
Obrázek 16: Grafické znázornění Eulerových úhlů v tzv. aerospace sekvenci.	14
Obrázek 17: Kvaternion popisující rotaci souřadnicového systému B vzhledem k A	15
Obrázek 18: Blokové schéma tzv. „Dirty“ komplementárního filtru.....	23
Obrázek 19: Blokové schéma komplementárního filtru s HP a LP filtrem.	24
Obrázek 20: Blokové schéma nelineárního komplementárního filtru.	25
Obrázek 21: Data z magnetometru se zkreslením a ideální / kalibrovaná data.	28
Obrázek 22: Hlavní plocha vytvořené aplikace pro PC po jejím spuštění.	38
Obrázek 23: „Nastavení“ aplikace obsahující: nastavení postavy a vzorkování / zobrazení.....	39
Obrázek 24: „Nastavení“ aplikace obsahující: zpracovávaná data a informace o kalibraci.....	40
Obrázek 25: „Nastavení“ aplikace obsahující: nastavení algoritmu, záložek a auto-kalibrace.....	40
Obrázek 26: Dialog pro výběr profilu aplikace a upozornění pro „Chyba postavy“.....	42
Obrázek 27: Dostupné postavy do 3D scény aplikace.....	42
Obrázek 28: Záložka aplikace „Manual“ se znázorněním nastaveného natočení hlavy kostry.	43
Obrázek 29: Záložka aplikace „Komun VCP“ pro připojení senzoru přes USB.	44
Obrázek 30: Záložka aplikace „Komun BT“ znázorňující start hledání BT zařízení a jeho výběr.....	44
Obrázek 31: Záložka aplikace „Komun BT“ se stavem před připojením k BT zařízení a po připojení... ..	45
Obrázek 32: Části záložky aplikace „Data“ obsahující příchozí data a tlačítka pro zobrazení grafů.	45
Obrázek 33: Graf typu „2D Lines“ zobrazující data ze senzorů a výsledné úhly.	46
Obrázek 34: Okno s nastavením a legendou pro graf aplikace typu „2D Lines“.....	46

Obrázek 35: Záznam dat z magnetometru při jeho náhodném otáčení při statické pozici.	47
Obrázek 36: Graf typu „3D XYZ“ zobrazující vektor zrychlení a vektor mag. intenzity.	48
Obrázek 37: Část záložky aplikace „Data“ s tlačítky pro vykonání kalibrace senzorů.	48
Obrázek 38: Část záložky aplikace „Data“ informující o průběhu kalibrace gyroskopu.	49
Obrázek 39: Záznam dat z magnetometru při jeho otáčení ve všech směrech, před a po kalibraci. ...	49
Obrázek 40: Záložka aplikace „Měření“ s výběrem aktivního typu cvičení.	50
Obrázek 41: Záložka aplikace „Měření“ s vyznačením vztahu mezi naměřenými úhly a úhломěry.	51
Obrázek 42: Záložka aplikace „Výsledky“ poskytující zobrazení výsledků formou tabulky.	52
Obrázek 43: Záložka aplikace „Výsledky“ poskytující zobrazení výsledků formou grafu.	53
Obrázek 44: Postup použití mobilní aplikace „MovementUni“ jako pohybový senzor.	54
Obrázek 45: Postup navázání BT komunikace se senzorem prostřednictvím aplikace na PC.	55
Obrázek 46: Příklad nastavení záložky „Data“ aplikace na PC před zahájením měření.	55
Obrázek 47: Znárodnění správného umístění pohybového senzoru na hlavě cvičícího.	56
Obrázek 48: Postup spuštění jednoho měření / cvičení s názvem „Hlava“ v aplikaci na PC.	56
Obrázek 49: Stav aplikace na PC v průběhu měření.	57
Obrázek 50: Stav aplikace na PC po dokončeném měření.	57