



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

Využití senzitivních robotů pro účely rehabilitace

Diplomová práce

Studijní program: N2612 – Elektrotechnika a informatika
Studijní obor: 3902T005 – Automatické řízení a inženýrská informatika
Autor práce: **Bc. Vojtěch Marhoul**
Vedoucí práce: Ing. Tomáš Martinec, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies ■

Use of sensitive robots for rehabilitation purposes

Master thesis

Study programme: N2612 – Electrical Engineering and Informatics
Study branch: 3902T005 – Automatic Control and Applied Computer Science

Author: **Bc. Vojtěch Marhoul**
Supervisor: Ing. Tomáš Martinec, Ph.D.



ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Vojtěch Marhoul**
Osobní číslo: **M15000198**
Studijní program: **N2612 Elektrotechnika a informatika**
Studijní obor: **Automatické řízení a inženýrská informatika**
Název tématu: **Využití senzitivních robotů pro účely rehabilitace**
Zadávající katedra: **Ústav mechatroniky a technické informatiky**

Z á s a d y p r o v y p r a c o v á n í :

1. Seznamte se s vlastnostmi senzitivních (kooperativních) robotů a s možnostmi jejich použití pro účely rehabilitace.
2. Navrhněte a realizujte aplikaci pro robota KUKA iiwa, která bude umožňovat využití tohoto robota pro účely rehabilitace. Aplikace by měla zahrnovat možnost definovat cviky rehabilitačním pracovníkem a sledovat jejich provádění pacientem.
3. Vytvořte aplikaci pro PC, která bude umožňovat nastavení, sledování a vizualizaci rehabilitačního procesu.



Rozsah grafických prací: **dle potřeby dokumentace**

Rozsah pracovní zprávy: **40–50 stran**

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

- [1] **KUKA Sunrise Cabinet, Assembly Instruction, KUKA Roboter GmbH, Augsburg, 2015.**
- [2] **KUKA Sunrise.OS 1.7, KUKA Roboter GmbH, Augsburg, 2015.**

Vedoucí diplomové práce: **Ing. Tomáš Martinec, Ph.D.**

Ústav mechatroniky a technické informatiky

Konzultant diplomové práce: **doc. Ing. Josef Černožorský, Ph.D.**

Ústav mechatroniky a technické informatiky

Datum zadání diplomové práce: **10. října 2017**

Termín odevzdání diplomové práce: **14. května 2018**

prof. Ing. Zdeněk Plíva, Ph.D.
děkan



Kolář
doc. Ing. Milan Kolář, CSc.
vedoucí ústavu

V Liberci dne 10. října 2017

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 14.5.2018

Podpis: Marková

Poděkování

Na tomto místě bych chtěl poděkovat všem, kteří mi při zpracování této diplomové práce pomáhali, nebo mě jakkoliv podporovali. V první řadě děkuji vedoucímu práce Ing. Tomáši Martincovi, Ph.D. za konzultace, cenné rady, podněty a připomínky. Velké poděkování patří také rodině za všestrannou podporu při vypracování práce i během celého studia.

Abstrakt

Tato práce se zabývá problematikou návrhu aplikace pro účely rehabilitace horních končetin s využitím průmyslového senzitivního robota KUKA LBR iiwa. Rešeršní část práce obsahuje přehled výzkumů zabývajících se robotickou rehabilitací s využitím průmyslových robotů i jednoúčelových zařízení. Dále se tato část skládá z popisu vlastností průmyslových senzitivních robotů od známých výrobců, včetně jejich vzájemného porovnání z hlediska využitelnosti pro účely rehabilitace. Závěr rešeršní části obsahuje přehled současných technik robotické rehabilitace spolu s popisem hlavních účelů a funkcí rehabilitační aplikace, jejímž návrhem se zabývají další kapitoly.

Hlavní komponenty systému tvoří kooperativní robot LBR iiwa komunikující s počítačem. Robot slouží jako nástroj pro provádění fyzioterapie, včetně záznamu a vyhodnocení sil, kterými pacient na koncový efektor robota působí. Aplikace v počítači slouží jako uživatelské rozhraní umožňující kromě vizualizace a sledování průběhu fyzioterapie také například správu složek pacientů. Pro kontrolu pokroku pacientů jsou data z rehabilitace přehledně prezentována a ukládána. Uživatelské rozhraní je vytvořeno prostřednictvím programovacího jazyka C#.

Klíčová slova

Fyzioterapie, KUKA LBR iiwa, robotická rehabilitace, senzitivní robot, trajektorie

Abstract

This thesis deals with application design for the purpose of upper limb rehabilitation using KUKA LBR iiwa Industrial Sensitive Robot. The research part contains an overview of research on robotic rehabilitation using industrial robots and dedicated devices. Further, this part consists of a description of the properties of industrial sensitive robots from known manufacturers, including their mutual comparison in terms of usability for rehabilitation purposes. The conclusion of the research section is an overview of the current techniques of robotic rehabilitation together with a description of the main purposes and functions of the rehabilitation application, whose proposal is dealt with in other chapters.

The main components of the system are the LBR cooperative robot iiwa communicating with the computer. The robot serves as a tool for physiotherapy, including recording and evaluating the forces that the patient exerts on the end effector of the robot. The application on the computer serves as a user interface allowing, in addition to visualizing and tracking the course of physiotherapy, for example, patient folder management. To monitor patient progress, rehabilitation data is clearly presented and stored. The user interface is created using C# programming language.

Key words

Physiotherapy, KUKA LBR iiwa, robotic rehabilitation, sensitive robot, trajectory

Obsah

Úvod	12
1 Robotická rehabilitace	14
1.1 Rehabilitace s využitím průmyslových robotů.....	14
1.1.1 Rehabilitace dolních končetin – robot KUKA KR270	14
1.1.2 Rehabilitace horních končetin – robot KUKA LBR iiwa	15
1.1.3 Rehabilitace horních končetin – robot KUKA KR5	17
1.2 Rehabilitace s využitím specifických zařízení	18
1.2.1 Rehabilitace dětí – humanoidní robot SAR.....	18
1.2.2 Rehabilitace prstů a zápěstí.....	18
1.2.3 Rehabilitace kotníku – robot ARBOT.....	19
1.3 Vlastnosti senzitivních robotů od známých výrobců	20
1.4 Techniky a možnosti robotické rehabilitace.....	25
1.5 Funkční popis rehabilitační aplikace	28
1.5.1 Hlavní funkce aplikace	29
2 Rehabilitační aplikace s robotem LBR iiwa.....	35
2.1 Úvod do programování.....	35
2.1.1 Architektura programu a TCP komunikace.....	36
2.2 Správa cviků a pacientů	39
2.3 Mechanismus pro definici trajektorie	41
2.3.1 Zpracování požadavku od klienta.....	42
2.3.2 Mechanismus pro uzamykatelnost os robota	42
2.3.3 Ruční navádění a záznam hodnot	44
2.3.4 Vyhodnocení a ukončení procesu	46
2.3.5 Definice trajektorie v off-line režimu	46
2.4 Proces provádění fyzioterapie	47
2.4.1 Parametrizace trajektorie.....	47

2.4.2	Sestavení trajektorie s konstantní rychlostí.....	48
2.4.3	Sestavení trajektorie s prahovou hodnotou	50
2.4.4	Sestavení trajektorie s proměnnou rychlostí	52
2.5	Zpracování hodnot.....	53
3	Závěr	57
	Seznam použité literatury	59
A	Obsah přiloženého CD	61

Seznam obrázků

Obrázek 1: Výzkumná platforma pro trénink a rehabilitaci dolních končetin [3]	14
Obrázek 2: Snímače pohybu připevněny na paži [2].....	16
Obrázek 3: Zařízení pro rehabilitaci kotníku (ARBOT) [7]	19
Obrázek 4: Přehled senzitivních robotů.....	22
Obrázek 5: Systém robota	23
Obrázek 6: Části a osy robota	23
Obrázek 7: Systém pro rehabilitaci horních končetin	29
Obrázek 8: Dialogové okno po dokončení definice trajektorie.....	30
Obrázek 9: Architektura softwaru (třídy).....	36
Obrázek 10: Hlavní okno uživatelského rozhraní.....	40
Obrázek 11: Okno pro zahájení definice trajektorie	41
Obrázek 12: Uživatelské prostředí pro uzamykatelnost os.....	43
Obrázek 13: Pozice os uložené v souboru	45
Obrázek 14: Rozhraní pro parametrizaci, zahájení a sledování fyzioterapie	52
Obrázek 15: Ukázka záznamového listu.....	55

Seznam zdrojových kódů

Zdrojový kód 1: Metoda pro přijímání dat od klienta	38
Zdrojový kód 2: Metoda pro zasílání dat serveru	39
Zdrojový kód 3: Metoda pro nastavení pohyblivosti os.....	43
Zdrojový kód 4: Metoda pro definici tuhosti os.....	47
Zdrojový kód 5: Ukázka zdrojového kódu pro definici posloupnosti pohybů	49
Zdrojový kód 6: Testování síly působící na efektor robota	51
Zdrojový kód 7: Tvorba posloupností pohybů s definovaným časem	53
Zdrojový kód 8: Zobrazení hodnot v grafu.....	54

Seznam použitých zkratk a symbolů

LBR	Leichtbauroboter, robot lehké konstrukce
IWA	Intelligent Industrial Work Assistant
IMU	Inertial Measurement Units, senzory měřící svalovou aktivitu
SEFRE	Shoulder-Elbow-Forearm Robotics Economic, rehabilitační systém
SAR	Socially Assistive Robot, rehabilitační systém
TCP	Transmission Control Protocol, protokol pro síťovou komunikaci
XML	eXtensibleMarkup Language, rozšiřitelný značkovací jazyk
DOM	Document Object Model, knihovna pro práci s XML soubory
PTP	Point To Point

Úvod

Tato práce pojednává zejména o softwarovém nástroji pro účely rehabilitace, ale také o senzitivním průmyslovém robotu, pomocí kterého se bude rehabilitační cvičení provádět a pro kterého je tento software navržen. Senzitivní robotika je ve světě stále nové a atraktivní téma. Pod pojmem tímto pojmem si lze nejlépe představit přímou spolupráci robota s člověkem.

Senzitivní roboty spojuje několik charakteristických vlastností. Tou hlavní jsou momentové senzory zabudované v každé ose robota. Tato vlastnost otevírá nové možnosti využití robotů. Pomocí senzorů v osách dokáže robot přepočítat sílu působící z jakéhokoli směru na jakýkoli definovaný souřadný systém, kterým může být například koncový bod nástroje připojený k přírubě robota. Vzhledem k této funkci se nabízí nové využití v průmyslu, kde například robot dokáže přesně napasovat díl do určeného otvoru bez předem přesně daných souřadnic (testování působících sil na nástroj ramene ve směru Z).

Momentové senzory v osách mohou být využity také v oblasti bezpečnosti. Pro tento případ je možné nastavit hraniční moment v každé ose robota a při jeho překročení zastavit pohyb. Díky této funkci může robot reagovat na kolizi předmětu nebo osoby s každou jeho osou nebo například s nástrojem (pomocí přepočtu sil). Z tohoto důvodu není nutné ochranné oplocení, jako je tomu u klasických průmyslových robotů. Vlastnost senzitivity robota má široké využití nejen v průmyslu, ale také velký potenciál v oblasti zdravotnictví a rehabilitace. Schopnost vyhodnotit vnější působící síly na definovaný souřadný systém je klíčová i pro rehabilitační aplikaci, které se věnuje tato práce.

Dalším charakteristickým prvkem pro kooperativní roboty je způsob programování. U klasického typu robotů je zvykem většinou změnit polohu robota pomocí ovládacího panelu, zapamatovat si bod a takto poměrně složitě naučit robota celý pohyb. Senzitivní robot lze programovat jednodušeji, a to pomocí ručního navádění. Rameno se vede přímo rukou a ukládá si souřadnice koncového nástroje, nebo pozice (úhly) jednotlivých os do souboru.

Senzitivní roboty spojují i další vlastnosti, jedna z nich je například lehká konstrukce. Roboty lehké konstrukce jsou konstruovány do nosnosti jednotek kilogramů (obvykle ne více než 20 kg), a proto i samotné rameno není příliš robustní a neváží většinou více než 40 kg.

Rychle se rozvíjející oblastí pro využití senzitivních robotů je i medicína a speciálně rehabilitace. Z průběžných demografických výzkumů vyplývá, že světová populace stárne. Obyvatelstvo má v současné době zastoupení přibližně 20 % populace starší 65 let. Dle dlouhodobé předpovědi by se toto číslo mělo do roku 2030 zvýšit o 9 %. Tento trend tak zvyšuje poptávku mimo jiné i po rehabilitačních centrech a zkušených rehabilitačních lékařích, kterých je nedostatek. Trh s rehabilitačními zařízeními proto zaznamenává stále se zrychlující růst. V současné době se pro rehabilitační cvičení a diagnostiku používají z velké části jednoúčelové roboty, které jsou navrženy a uzpůsobeny pouze pro jednu konkrétní úlohu (procvičování pouze jedné části těla jedním způsobem). Hlavní cíl této práce, která navazuje na semestrální projekt věnující se rozboru a porovnání senzitivních robotů od nejznámějších výrobců a tvorbě ukázkové aplikace pro robota LBR iiwa, je vytvoření univerzální rehabilitační aplikace s využitím senzitivního robota LBR iiwa od firmy KUKA. Právě univerzálnost je hlavní výhodou použití průmyslového senzitivního robota v této problematice. V posledních dvou letech vzniká mnoho výzkumů zabývajících se využitím senzitivních robotů pro účely rehabilitace, v praxi však takovéto řešení prozatím nasazeno není. Z výše popsaných důvodů je toto téma velice aktuální i atraktivní, a proto se jím tato práce zabývá.

Vlastní text práce se skládá ze dvou hlavních částí, těmi jsou rešerše a návrh aplikace. V rešeršní části se práce věnuje popisu výzkumů zabývajících se robotickou rehabilitací s využitím průmyslových robotů i jednoúčelových zařízení. Další kapitoly rešerše sestávají z popisu senzitivních robotů od nejznámějších výrobců, včetně popisu robota LBR iiwa. Podkapitola je ukončena souhrnným přehledem a porovnáním jednotlivých robotů mezi sebou s ohledem na jejich využití pro účely rehabilitace. V závěru rešerše je představena rehabilitační aplikace, která je hlavním tématem této práce.

V hlavní kapitole práce, týkající se řešení, je popsán postup při vytváření vlastního softwaru, který zahrnuje volbu typu aplikace a použité prostředky pro realizaci. Dále tato kapitola obsahuje návrh architektury programu a samotný postup při programování aplikace, včetně okomentování a vysvětlení dosažených výsledků.

1 Robotická rehabilitace

Tato kapitola se zabývá popisem výzkumů věnujících se rehabilitaci s využitím průmyslových robotů i jednoúčelových zařízení. Další část kapitoly je věnována vlastnostem senzitivních robotů od nejznámějších výrobců, včetně souhrnného přehledu a porovnání jednotlivých robotů mezi sebou s ohledem na jejich využití pro účely rehabilitace. Závěr kapitoly je věnován současným technikám robotické rehabilitace a funkčnímu popisu rehabilitační aplikace, kterou se tato práce zabývá.

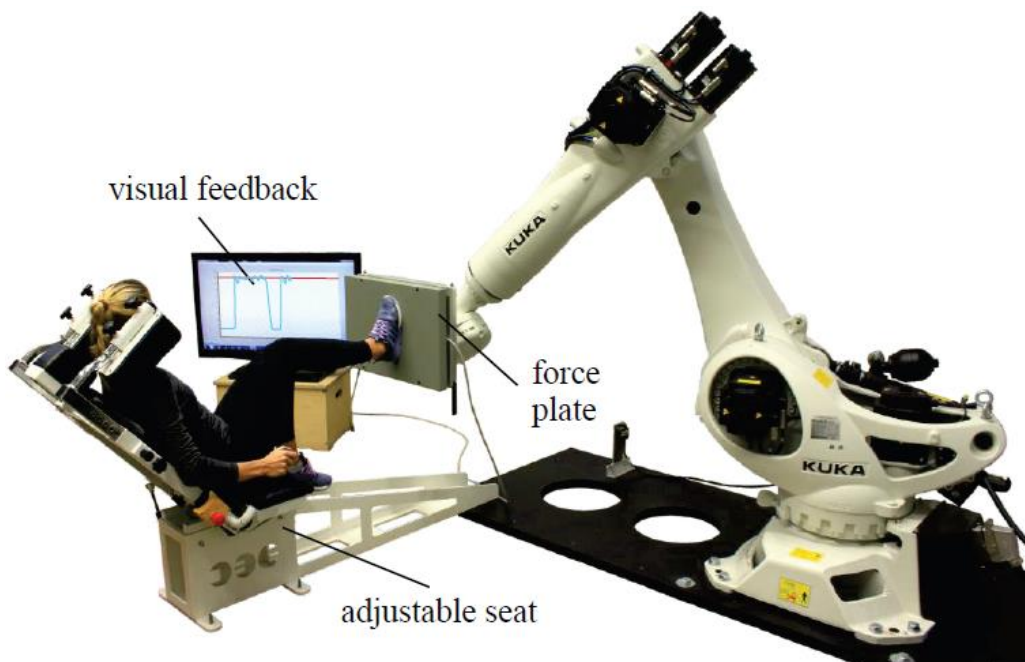
1.1 Rehabilitace s využitím průmyslových robotů

1.1.1 Rehabilitace dolních končetin – robot KUKA KR270

Tento výzkum [3] se zabývá izokinetickým cvičením a rehabilitací dolních končetin za pomoci průmyslového robota KR270 od firmy KUKA. Na výzkumu spolupracovala RWTH Univerzita v Cáchách a Sportovní Univerzita v Kolíně nad Rýnem.

Využití aplikace

- Jako součást výcvikového programu pro sportovce.
- Ke snížení rizika svalového onemocnění zapříčiněného pokročilým věkem.
- Terapeutická intervence poškozeného svalstva vlivem zranění.
- Rehabilitace a prevence artrózy kyčelního nebo kolenního kloubu.



Obrázek 1: Výzkumná platforma pro trénink a rehabilitaci dolních končetin [3]

Použitý hardware

Výzkumná platforma pro rehabilitaci dolních končetin (viz Obrázek 1) se skládá ze čtyř hlavních částí. Jako aktuátor slouží šestiosý průmyslový robot KR270 od firmy KUKA s maximální nosností 270 kg. Silová deska jako koncový nástroj robota slouží pro měření momentu a síly, která je v reálném čase zobrazována na připojeném počítači. Poslední částí platformy je sedadlo s nastavitelným úhlem opěradla, které je pevně uchyceno k podstavci.

Popis rehabilitačního procesu

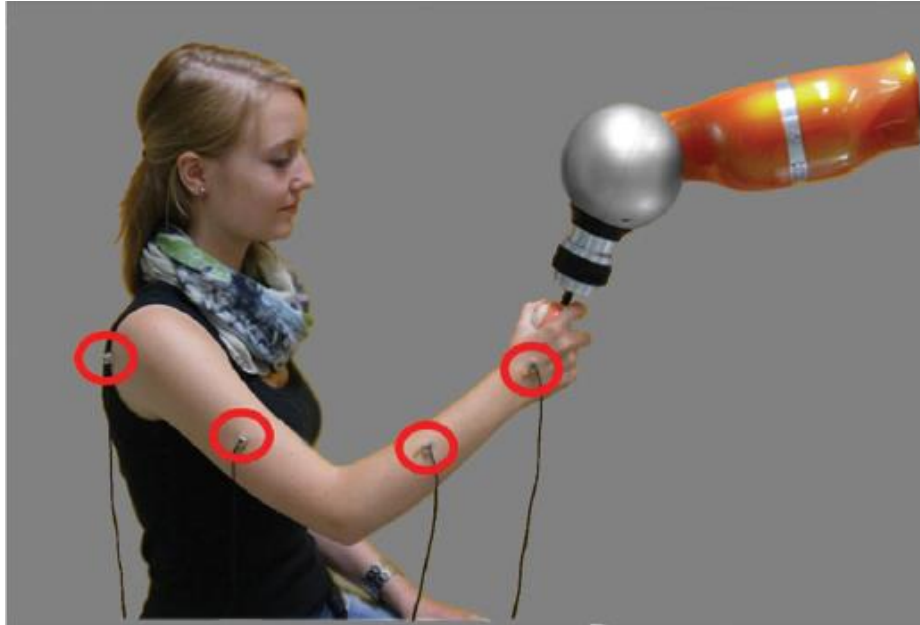
Izokinetická trajektorie pohybu při procvičování dolních končetin se skládá z extenze a flexe nohy v kolenním kloubu po lineární ose s konstantní rychlostí. Pohyb začíná a končí nulovou rychlostí s tím, že lineárně akceleruje na požadovanou rychlost a před konečným bodem trajektorie opět rychleji (nelineárně) zpomaluje na nulovou rychlost, tím je dosaženo hladkého a přirozeného pohybu. Cvičení je zahájeno vyvinutím síly na silovou desku, která je vyšší než minimální definovaná síla. Dokud je působící síla nad touto prahovou hodnotou, robot stále vykonává nedefinovanou trajektorii tam a zpět.

1.1.2 Rehabilitace horních končetin – robot KUKA LBR iiwa

Tento výzkum [2] se zabývá rehabilitací horních končetin za pomoci průmyslového senzitivního robota LBR iiwa od firmy KUKA. Výsledná aplikace je určena především pro pacienty, kteří trpí neurologickou poruchou, kvůli které mají problém s pohybem. Tato porucha bývá zapříčiněna z velké části mozkovou příhodou. Studii vypracoval výzkumný tým z Univerzity RWTH v Cáchách.

Použitý hardware

Výzkumná platforma pro rehabilitaci horních končetin se skládá ze dvou hlavních částí. Tou první je detekční systém pohybu, který slouží pro zachycení kvantitativních údajů o pohybu. Pacienti s neurologickými poruchami jsou velmi citliví na vnější podněty, proto musí být senzory malé a lehké, aby neovlivnily pohyb paže. Tuto funkci plní malé (2 x 2 x 0.5 mm) a lehké (5 g) snímače pohybu zvané Inertial Measurement Units (dále jen IMU). Každý IMU snímač disponuje 3D akcelerometrem, 3D magnetometrem a 3D gyroskopem. Pacient má po procvičované paži rozmístěny celkem čtyři senzory – lopatka, paže, předloktí a ruka (viz Obrázek 2). Druhou částí systému je průmyslový senzitivní robot LBR iiwa od firmy KUKA, který v reálném čase zpracovává informace ze senzorů.



Obrázek 2: Snímače pohybu připevněny na paži [2]

Popis rehabilitačního procesu

Proces rehabilitace je rozdělen na dvě základní fáze: fáze výuky a fáze cvičení. Během výuky fyzioterapeut vede pacientovu paži tak, aby například simuloval libovolnou každodenní činnost, která pacientovi činí problémy. Robot pasivně následuje pacientovu paži a zaznamenává si polohu koncového efektoru tak, aby mohl celý pohyb zopakovat. Současně s tímto je zaznamenáván pohyb pacientovi paže ze snímačů. Data o pohybu paže z IMU senzorů jsou poté využity jako referenční údaje pro fázi cvičení. V této fázi robot vykonává naučenou trajektorii a pacient pohyb robota pomocí koncového efektoru pasivně následuje. Rychlost pohybu je rovna rychlosti, jakou byl pohyb naučen. Počet opakování je dán fyzioterapeutem. Při provádění cviku systém zaznamenává pohyb paže (dohromady čtyři proměnné ze čtyř částí paže) a analyzuje rozdíly mezi skutečností a referencí, která byla vytvořena ve fázi výuky. V rámci výuky byla fyzioterapeutem stanovena tolerance rozsahu pohybu jednotlivých částí paže, která slouží k následnému vyhodnocení cviku a k porovnání reference s naměřenými hodnotami.

Vyhodnocení cviků

Vyhodnocení cviku se rozděluje na dva základní typy: lingvistické a kvantitativní. Pro lingvistické hodnocení jsou použity tři základní proměnné: „perfektní“, „dobré“ a „špatné“. Tyto parametry hodnocení cviku jsou vytvořeny fuzzifikací z naměřených hodnot pomocí přenosových funkcí, na kterých se podílí také fyzioterapeut. Přínos fyzioterapeuta spočívá v definici pravidel, které jsou vytvořeny pomocí jednoduchých logických operátorů jako například „IF-THEN“, nebo „OR/AND“. Výsledkem těchto funkcí

je zobrazení naměřených hodnot v lingvistickém prostoru. V dalším kroku je provedena defuzzikace, tedy převod z jazykových proměnných do reálných proměnných. Výsledkem tohoto procesu je kvantitativní vyhodnocení pohybu jako jeden segment s použitím hodnot v rozsahu 0 až 10, kde nula znamená „velmi špatný“ a deset znamená „perfektní“ výsledek. Tyto hodnoty reprezentují pohybovou výkonnost všech čtyř částí paže, které jsou sledovány.

Závěr

Tento systém umožňuje sledování pohybu paže pacienta a následné vyhodnocení pomocí porovnání dat z výuky a cvičení. Na tomto systému již byly provedeny experimentální testy, kterých se zúčastnily dva subjekty. Jeden zdravý, bez pohybových potíží, a jeden pacient, který prodělal mozkovou příhodu a trpí svalovým ochabnutím. Z těchto testů bylo zjištěno, že je systém schopen detekovat kompenzační pohyby paže a kinematické účinky způsobené vyčerpáním. V budoucnu je možné systém rozšířit o automatické přizpůsobení robota v reálném čase (rychlost, tuhost atd.) na aktuální stav pacienta.

1.1.3 Rehabilitace horních končetin – robot KUKA KR5

Na studii rehabilitace horních končetin [4] s využitím průmyslového robota KUKA KR5 pracoval tým z Národního Technologického Centra v Thajsku. K rehabilitaci slouží systém pojmenovaný Shoulder-Elbow-Forearm Robotics Economic (dále jen SEFRE). Systém se skládá ze tří hlavních komponentů. Pro pohybování paže v ramenním kloubu pacienta slouží malý průmyslový robot KR5 od společnosti KUKA. Druhým komponentem systému je exoskelet, který je připojen k robotu jako koncový efektor. Exoskelet disponuje silovými senzory a vykonává pohyb lokte a předloktí. Posledním komponentem platformy je počítač, který slouží jako uživatelské rozhraní pro přepínání rehabilitačních programů a pro sledování výsledků fyzioterapie. Systém poskytuje pět terapeutických módů: pasivní, aktivní zahájení, aktivní asistence, aktivní rezistence a pasivní extenze. Tyto módy určují, jakým způsobem bude iniciován pohyb a volí se na základě pohybové schopnosti pacienta. Systém SEFRE disponuje také herním módem. V tomto módu pacient procvičuje své schopnosti pomocí jednoduchých her, ve kterých má například za úkol pohybovat nádobou a chytat padající ovoce ze stromu. Kontrola bezpečnosti je zajištěna programovatelnou logickou jednotkou Arduino. Použití tohoto hardwaru je pro účely bezpečnosti diskutabilní a může se tak jevit jako slabší stránkou systému SEFRE.

1.2 Rehabilitace s využitím specifických zařízení

1.2.1 Rehabilitace dětí – humanoidní robot SAR

Tento výzkum [5] vypracoval Australský tým, jehož členové pochází z Technologické Univerzity ve Swinsburne a z dětské nemocnice Royal v Melbourne. Vyvíjený systém je určen primárně pro děti, které prodělaly mozkovou obrnu. Při dětské mozkové obrně dochází k poškození motorických oblastí mozku a to má za následek nedostatečnou kontrolu hybnosti a vadné držení trupu a končetin. Výsledkem tohoto výzkumu je humanoidní robot s názvem Socially Assistive Robot (dále jen SAR). Při fyzioterapii je robot s pacientem sám a fyzioterapeut pouze sleduje průběh v oddělené místnosti. Fyzioterapie je zahájena úvodní řečí robota, která kromě rehabilitačního scénáře zahrnuje také vtipy a předprogramovaný dialog pro podpoření dobrého vztahu dítěte k robotu. Při zahájení cvičení robot nejprve daný cvik demonstruje a vysvětlí klíčové informace pro správné provedení cviku. Poté je pacient vyzván k tomu, aby se připojil a daný cvik dle robota vykonal. Během cvičení robot povzbuzuje pacienta k lepším výkonům a připomíná klíčové aspekty daného cviku. Robot SAR je naprogramován na třináct různých rehabilitačních cviků. Dále systém disponuje rehabilitací prostřednictvím her s hračkami. Cílem této funkce je motivovat pacienta k chůzi, přičemž dostává od robota instrukce k přemísťování hraček. Rehabilitační proces je ukončen rozloučením a odměnou v podobě tance robota.

1.2.2 Rehabilitace prstů a zápěstí

Čínský výzkum [6], který se zabývá rehabilitací zápěstí, ruky a prstů, se skládá z řady vědců a lékařů z Čínských nemocnic a univerzit. Tento systém je navržen převážně pro pacienty, kteří prodělali mozkovou příhodu a potřebují opět obnovit funkci zápěstí, ruky nebo prstů. Rehabilitace těchto částí těla je obtížnější než u ostatních částí, a proto je třeba této problematice věnovat vyšší pozornost. Zařízení, kterému se tato kapitola věnuje, se vymezuje vůči těm stávajícím několika prvky. Na rozdíl od většiny současných systémů tento rehabilitační robot umožňuje při fyzioterapii prstů také zároveň otáčení zápěstím, což zvyšuje účinnost rehabilitace. Další výhodou spočívá v možnosti procvičování funkčního pohybu všech prstů najednou, oproti stávajícím zařízením, která se převážně zaměřují na flexibilní trénink každého prstu zvlášť. Systém se skládá ze dvou hlavních komponentů. Těmi jsou rehabilitační jednotka ruky a rehabilitační jednotka zápěstí. Tyto

komponenty jsou propojeny a fungují jako celek. Rehabilitační jednotka zápěstí umožňuje rotační pohyb ruky dvěma směry (extorze a intorze). Ruční jednotka se skládá z madla a posuvníku, který je určen k pohybu prstů. Pohyb jednotlivých částí systému zajišťují motory s momentovými senzory. Údaje z těchto senzorů jsou zpracovány v připojeném počítači. Pohyby jsou iniciovány robotem, proto je systém vhodný zejména pro pacienty s nulovou nebo velmi nízkou pohybovou schopností.

1.2.3 Rehabilitace kotníku – robot ARBOT

Výzkumem rehabilitace kotníku [7] se zabývá tříčlenný tým složený z King's College v Londýně a Italského Technologického institutu.



Obrázek 3: Zařízení pro rehabilitaci kotníku (ARBOT) [7]

Mechanická struktura se skládá ze základny, centrální vzpěry, pohybující se platformy a tří ramen, která zajišťují pohyb platformy (viz Obrázek 3). Systém má dva stupně volnosti a umožňuje dva druhy fyzioterapie. První způsob je určen ke zvyšování rozsahu pohyblivosti kotníku a může být proveden ve dvou různých módech. Jedná se o pasivní a aktivní rehabilitaci. V pasivním módu jsou veškeré pohyby z nadefinované trajektorie iniciovány robotem. Při spuštění programu v aktivním módu je třeba, aby pacient vyvinul sílu požadovaným směrem. Poté robot přebere iniciativu a pohyb dokončí. Toto cvičení je určeno pro pacienty, kteří jsou již schopni kotníkem alespoň minimálně pohybovat. Další způsob rehabilitace spočívá v izometrickém silovém cvičení. V tomto případě se pacient snaží vyvinout určeným směrem sílu na platformu, kterou robot drží ve stabilní, fixní poloze. Síla je měřena senzorem a vyhodnocena. Jedná se tak o testování pokroku pacienta. Poslední fáze fyzioterapie je izotonické cvičení. Jedná se o rezistentní rehabilitaci, kde při pohybu robot vyvíjí určitý odpor proti vyvíjené síle pacienta.

1.3 Vlastnosti senzitivních robotů od známých výrobců

Tato kapitola se zabývá krátkým popisem průmyslových senzitivních robotů od nejznámějších výrobců. Závěr kapitoly je věnován souhrnu vlastností těchto robotů a důvodům volby robota LBR iiwa pro rehabilitační aplikaci, kterou se tato práce zabývá.

Universal Robots – UR5

Firma Universal Robots nabízí roboty UR3, UR5, UR10, kde číselné označení robota znamená jeho maximální nosnost. Roboty UR umožňují automatizovat opakované a nebezpečné úkoly s max. zatížením 10 kg a pracovním rozsahem až 1 350 mm (pro UR10). Roboty UR jsou navrženy především pro kooperativní procesy s malou hmotností, např. odebírání, ukládání nebo testování. Programování je relativně jednoduché, protože obsahuje SW s 3D vizualizací, která umožňuje nastavit a zprovoznit robota i uživatelům bez předchozích zkušeností s programováním. Jednoduše je možné přemístit rameno robota na požadované místo nebo stisknout kurzorová tlačítka na intuitivním dotykovém tabletu.

ABB – YuMi

Společnost ABB vyvinula průmyslového robota se dvěma pažemi pro montáž drobných součástek, který je schopen spolupracovat s lidmi na stejných úkolech při zajištění jejich bezpečnosti. Robot YuMi, je tvořen flexibilními pažemi, systémem podávání součástek a robotickým řízením na vysoké úrovni. Je velmi přesný (0,02 mm) a rychlý (max. 1500 mm/s). Maximální nosnost jedné paže je 500 g. Název znamená „You and Me“, v překladu „Ty a Já“, což vystihuje spolupráci s člověkem. Velkou výhodou těchto robotů je, že mají zabudovaný bezpečnostní systém, takže nemusí pracovat v oddělených prostorech. Duální robotické paže jsou vybaveny přesným vizuálním systémem, citlivou zpětnou vazbou regulace síly a zabudovanými bezpečnostními prvky, které umožňují programování orientované na bázi „učení“ než na psaní zdrojového kódu. Programování robota je velice intuitivní – paže a ruce robota lze ovládat v reálném čase pomocí vlastních paží a rukou prostřednictvím řady pohybů, kde se jejich souřadnice automaticky zaznamenají na připojeném tabletu v aplikaci YuMi. V případě složitějších aplikací, než „učení“, je možno programovat robota pomocí programovacího jazyka RAPID. Výrobce uvádí, že robot dokáže manipulovat s různými předměty, od hodinek po tablety, a to s přesností, jakou je potřeba pro navléknutí nitě do ucha jehly. Robot byl navržen do běžného výrobního prostředí, kde umožňuje firmám optimální kombinaci spolupráce s lidmi.

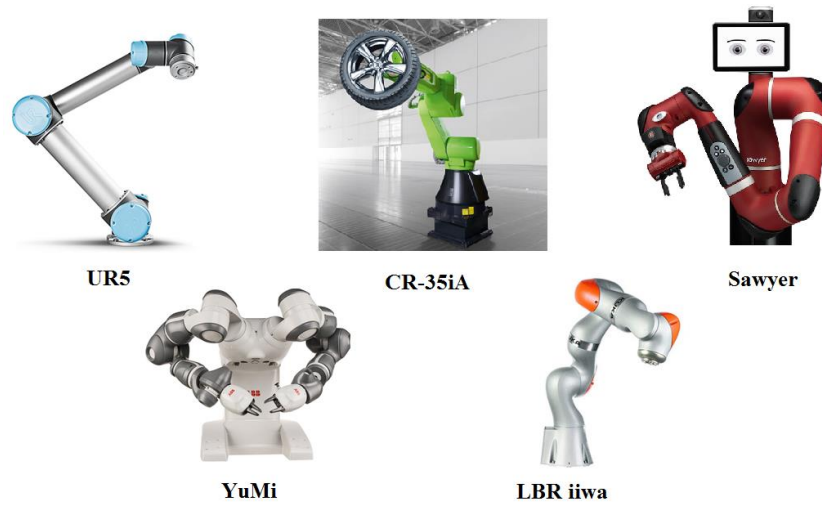
Rethink Robotics – Baxter & Sawyer

Společnost Rethink robotics má sídlo v Bostonu. Vyvinula vysoce výkonného spolupracujícího robota s názvem Sawyer, pro obsluhu strojů, testování desek s plošnými spoji a dalšími přesnými úkoly, které jsou nepraktické pro běžné roboty. V současné době je dostupný v Severní Americe, Evropě a Asijsko-pacifické oblasti. Robot váží 19 kg a má 7 stupňů volnosti. Umožňuje adaptivní opakovatelnost, která je v tomto odvětví robotiky unikátní a dokáže pracovat s přesností +/-0,1 mm v běžném provozu spolu s lidmi. Robot je složen z vestavěného systému strojového vidění s hlavní kamerou pro širokoúhlé zobrazení aplikací a s kamerou Cognex na svém zápěstí, která průběžně zobrazuje, v jaké pozici se robot nachází. Baxter a Sawyer jsou kompletní systémy, které mohou pracovat nezávisle na jednoduchých opakujících se úlohách anebo jako součást nějaké větší automatizace. Programování je možné na platformě Intera, která je nejvíce intuitivní platformou v průmyslu robotiky. Disponuje jednoduchým rozhraním, které umožňuje zaměstnancům mimo jiné intuitivně „učit“ robota pouhým naváděním.

Fanuc – CR-35iA

Společnost FANUC nabízí robota CR-35iA, který dokáže zvednout až 35 kg a je to nejsilnější spolupracující robot s člověkem na světě. Byl navržen tak, aby pracoval na těžkých opakujících se pracích např. v automobilovém průmyslu, balení, distribuci, zpracování kovů, ale hlavně po boku lidí ve výrobních linkách nebo na montážních operacích. V závislosti na tom, s čím chce zákazník pracovat má robot buď FANUC Vision sensor, nebo 3D sensor prostoru, který umožňuje vybrat náhodné části ze zásobníku a předat je člověku. Z hlediska bezpečnosti nepotřebuje žádné oplocení, protože je schopen detekovat předměty a v případě doteku ihned zastaví. Skládá se z měkkého pryžového obložení, které nikoho při doteku nezraní. Pokud by se dostal příliš blízko člověku, lze ho odtlačit pryč. FANUC disponuje více než 250 softwarovými funkcemi pro zvýšení inteligence, pohybu, bezpečnosti a produktivity. Nabízí jednoduché ovládací rozhraní a funkcí pro zjednodušení programování. Software FANUC LVC je ideálním řešením pro operace, které vyžadují přesnost ve vysokých rychlostech. Umožňuje robotovi „učit“ jeho kmitání pomocí akcelerometru, optimalizovat jeho cestu a dosáhnout tak vyššího zrychlení při zachování minimální vibrace (to i při maximální rychlosti). Nadstandardní výbavou je integrovaný snímač síly, který poskytuje vysoce citlivou kontrolu po dobu montáže. Soft-

ware FANUC Roboguide slouží k off-line programování, simulaci a tréninkovému programu. CR-35iA je certifikován a měl by splňovat požadavky mezinárodní normy ISO 10218-1.



Obrázek 4: Přehled senzitivních robotů

KUKA – LBR iiwa 7 R800

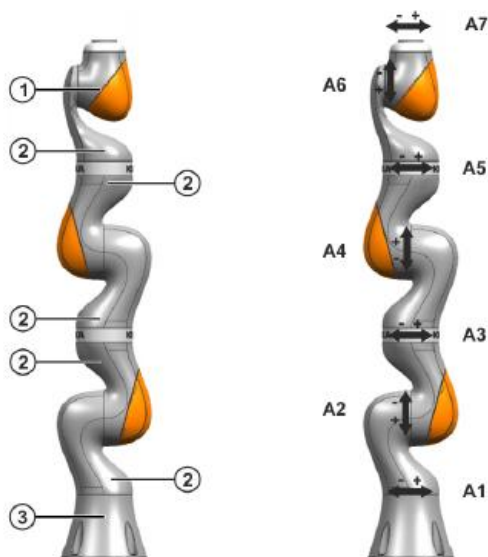
Pro robota LBR iiwa je navržena rehabilitační aplikace, jejíž realizací se zabývá druhá polovina této práce. Zkratka LBR znamená robot lehké konstrukce (z němčiny „Leichtbauroboter“), iiwa je zkratka pro „intelligent industrial work assistant“. Robot lehké konstrukce LBR iiwa se dodává ve dvou provedeních: s nosností 7 a 14 kg. Jeho lehká konstrukce je klíčem pro senzitivitu a citlivost robotu LBR iiwa. Typ robota využitý v této práci má hmotnost 23 kg a nosnost 7 kg. Robot disponuje sedmiosou bionickou konstrukcí a v každé ose má umístěn momentový senzor. Robot tak získal schopnosti hmatu, dokáže rozeznat tlak a díky tomu může vyhledávat objekty, se kterými bude pracovat. Na rozdíl od člověka dokáže regulovat svou sílu v jednotkách procent.

Robot je vybaven standardním dotykovým ovladačem SmartPAD, pomocí něhož lze robota ovládat dotykovými gesty či programovat navedením rukou. Celkový systém robota (viz Obrázek 5) se skládá z propojovacího kabelu do SmartPADu (1), KUKA SmartPAD řídicího panelu (2), manipulátoru – neboli mechanického systému a elektroinstalace (3), propojovacího kabelu do řídicího kabinetu (4) a KUKA Sunrise řídicího kabinetu robota (5).



Obrázek 5: Systém robota

Všechny motorové jednotky a elektrické vodivé kabely jsou chráněny pod krytem desky. Každá osa je chráněna pomocí senzorů rozsahu osy a může být upravována prostřednictvím interních senzorů. Každý kloub (spoj) je vybaven snímačem polohy na vstupní straně. Na výstupní straně jsou rozloženy senzory točivého momentu. Teplotní senzory zabráňují tepelnému přetížení robota. Robot se skládá z těchto částí, (viz Obrázek 6). Levá část obrázku ukazuje zápěstí robota (1), kde v osách A6 a A7 jsou umístěny motory. Spojové moduly (2) se skládají z hliníkové konstrukce, uvnitř těchto modulů jsou umístěny pohonné jednotky. Základní rámec (3) je základnou robota. V pravé části jsou zobrazeny osy A1 až A7.



Obrázek 6: Části a osy robota

Souhrn vlastností senzitivních robotů

Z výše uvedených vlastností vyplývají dvě základní, kterými musí disponovat každý robot tohoto druhu, aby mohl být označen jako senzitivní. Tou první je schopnost vyhodnotit sílu, která působí na efektor robota. Čtyři z pěti popsaných robotů tuto vlastnost mají zabudovanou již z výroby a dokáží pomocí senzorů v osách přepočítat výslednou sílu působící na jakýkoli definovaný souřadný systém (viz Úvod). Pátým robotem, který touto vlastností nedisponuje je CR-35iA od společnosti Fanuc, a proto ho v základu nelze označit jako senzitivní. Tento robot sice senzory v osách má, nicméně nedokáže síly na ně působící přepočítat na jiný souřadný systém. Senzory tak slouží pouze jako bezpečnostní prvek. Firma Fanuc nabízí rozšíření, které je do robotů možné za příplatek integrovat. Jedná se o senzor, který dokáže rozpoznat a vyhodnotit sílu působící na efektor ramene. Každý robot CR-35iA, který je vybaven tímto rozšířením, lze poté mezi senzitivní roboty zahrnout.

Druhou vlastností je bezpečnost. Ta je zajištěna právě momentovými senzory zabudovanými v osách. Robot tak dokáže reagovat na kolizi s jakoukoli jeho osou, nebo pomocí přepočtu sil (integrovaného senzoru v případě CR-35iA) na kolizi, například s efektozem robota. Spolu s ručním naváděním jsou tyto dvě vlastnosti klíčové také pro rehabilitační aplikaci uvedenou v kapitole 0. Vyhodnocení působících sil na koncovém efektoru robota je důležité především pro sledování pohybové schopnosti pacienta. Bezpečnost je požadována z důvodu přímé interakce člověka s robotem, kdy je žádoucí, aby robot reagoval na překážku, například v případě přesunu do startovní pozice. Vlastnost ručního navádění není pro rehabilitační aplikaci nezbytně nutná, nicméně tato možnost výrazně usnadňuje definici trajektorie.

K vytvoření rehabilitační aplikace, kterou se tato práce zabývá, je použit robot KUKA LBR iiwa. Hlavním důvodem volby je skutečnost, že je to jediný senzitivní robot, který je pro tyto účely na Technické Univerzitě v Liberci k dispozici. Bez ohledu na tento fakt je LBR iiwa pro rehabilitační aplikaci v podstatě ideální a z robotů uvedených v předchozí kapitole nejvhodnější. Jediným citelným omezením je maximální nosnost, která činí „pouze“ 7 kg. Z tohoto důvodu není možné robota využít pro rehabilitaci dolních končetin. V tomto ohledu by byl vhodnější jen UR10, u kterého by ale nosnost 10 kg pro rehabilitaci dolních končetin také nestačila, nebo CR-35iA. Druhý jmenovaný robot má sice dostačující nosnost (35 kg), nevýhodou je ovšem počet stupňů volnosti. Těch má CR-35iA šest oproti sedmi, jimiž disponuje LBR iiwa. Stupně volnosti hrají roli

především v počtu variant trajektorií, kterými je robot schopen dosáhnout efektozem požadovaných souřadnic. Přidáním sedmého stupně volnosti se tento počet mnohonásobně zvyšuje a díky tomu je možné naučit robota pohyb, jenž by robot se šesti stupni volnosti nebyl schopen vykonat. V případě rehabilitace tak LBR iiwa nabízí vyšší počet možných trajektorií cviku, než u robota CR-35iA.

V nabídce zůstávají zástupci Rethink Robotics a ABB. Robot YuMi od firmy ABB má dvě omezení, kvůli nimž je pro rehabilitaci horních končetin nepoužitelný. Jsou jimi maximální nosnost (0,5 kg) a maximální dosah (500 mm). Pro dvě paže, jimiž YuMi disponuje by se v oblasti rehabilitace taktéž využití nenašlo. Robot Sawyer od společnosti Rethink Robotics má nosnost 4 kg, která je oproti YuMi několikanásobně vyšší, nicméně pro využití v oblasti rehabilitace opět nedostatečná.

LBR iiwa tak jako jediný z výše uvedených robotů splňuje všechny potřebné parametry pro účely rehabilitace horních končetin. Disponuje sedmi-osou konstrukcí, umožňující vysokou variabilitu při volbě trajektorie. Dále potom maximální nosností 7 kg a dosahem 800 mm. Přesnost robota udávaná výrobcem je přibližně 0,1 mm. Jedná se o přesnost, se kterou je robot schopen opakovaně dosáhnout efektozem požadovaných souřadnic, nacházejících se v maximálním možném dosahu robota (v maximálním dosahu robota je přesnost nejnižší). Tato vlastnost nicméně pro rehabilitační aplikaci není důležitá. V následující tabulce jsou uvedeny hlavní technické parametry výše popsaných senzitivních robotů.

Název robota	Max. nosnost (kg)	Max. dosah (mm)	Přesnost (mm)	Stupně volnosti
UR10	10	1300	~ 0,1	6
YuMi	0,5	500	~ 0,02	7
Sawyer	4	1260	~ 0,1	7
CR-35iA	35	1813	~ 0,04	6
LBR iiwa R 800	7	800	~ 0.1	7

Tabulka 1: Parametry senzitivních robotů

1.4 Techniky a možnosti robotické rehabilitace

Rehabilitace je úsilí o znovuzavedení (obnovení, oživení) původního stavu pacienta. Fyzioterapie se aplikuje převážně na pacienty s pohybovou indispozicí, která může mít více příčin. Častou příčinou svalového onemocnění je fyzické poškození svalové tkáně, například při sportu, nebo jiných každodenních činnostech. Jiná příčina fyziologického původu může být zlomenina, například fraktura pánve. Léčebný proces této zlomeniny je velmi zdoluhavý a pacient potřebuje za pomoci fyzioterapie opět svalstvo kolem pánve posílit

a obnovit tak svou pohybovou schopnost. Druhá častá příčina svalového onemocnění má neurologický původ. Jedná se například o mozkovou příhodu nebo dětskou obrnu. Tyto nemoci často poškodí nervové ovládání svalové kontrakce a dochází tak k ochabnutí svalstva.

Rehabilitaci lze rozdělit do dvou základních skupin, těmi jsou tradiční fyzikální terapie a terapie za asistence rehabilitačních robotů. Stejně jako celá práce se i tato kapitola zabývá robotickou rehabilitací. Rehabilitační roboty mají vysoký potenciál pro zefektivnění fyzioterapie a zpřesnění diagnostiky. Velkou výhodou automatizovaných rehabilitačních systémů je schopnost provádět vysoký počet opakování cviku, což je prokazatelně vysoce přínosné pro léčbu neuromuskulárních chorob. Tyto systémy současně slouží jako extrémně přesné diagnostické nástroje. V současné době se pro rehabilitaci používají převážně jednoúčelová robotická zařízení. Pod jednoúčelovým systémem si lze představit robota, který je určen pro fyzioterapii jedné konkrétní části lidského těla. V posledních letech se i z důvodu nedostatku fyzioterapeutických lékařů robotizace v této oblasti rychle rozšiřuje. Existuje tak mnoho druhů specifických rehabilitačních robotů, sloužících k fyzioterapii prstů, zápěstí, předloktí, nadloktí, ramene, kyčle, kotníku atd. Existují také zařízení (exoskelety), která slouží k procvičování komplexních pohybů, jako je například chůze. Přístupy k fyzioterapii závisí především na pohybové schopnosti rehabilitované části pacientova těla (viz Tabulka 2).

Druh fyzioterapie	Charakteristika
Pasivní mód	Veškeré pohyby z nadefinované trajektorie jsou iniciovány robotem.
Aktivní mód	Pohyb po trajektorii je iniciován pacientem. Během provádění pohybu robot reaguje na pacientem vyvinutou sílu a v případě poklesu síly asistuje při pohybu robot.
Aktivní asistence	Pohyb po trajektorii provádí robot, ale pouze v případě, že pacient vyvíjí požadovaným směrem sílu vyšší, než je síla definovaná prahovou hodnotou. V opačném případě je pohyb pozastaven.
Aktivní rezistence	Pohyb po trajektorii je iniciován pacientem, přičemž robot klade proti směru prováděného pohybu odpor.

Tabulka 2: Přehled tréninkových módů pro rehabilitační roboty

Aktivní rezistence je poslední fází rehabilitačního procesu, která je určena pro pacienty s vyšší pohybovou schopností. Oproti tomu pasivní mód je určen pacientům s žádnou

nebo velmi nízkou pohybovou schopností. Tréninkové módy popsané v tabulce (viz Tabulka 2) jsou aplikovatelné pouze pro určité druhy rehabilitačních robotických systémů. Některé z uvedených módů jsou použity i pro zařízení představená v této práci, jako například rehabilitace horních končetin s využitím průmyslového robota KUKA KR5 (viz Kapitola 1.1.3). Tento systém je určen pro rehabilitaci pacientů s pohybovou schopností v celém rozmezí. Platforma umožňuje fyzioterapii ramene, nadloktí a předloktí. Tyto partie horní končetiny lze procvičovat ve všech módech uvedených v tabulce, od pasivního až po aktivní mód s rezistencí. Další rehabilitační systémy uvedené v přehledu umožňují jen některé druhy fyzioterapie a nemusí tak pokrýt celou škálu možných pohybových schopností pacientů.

Existují i další robotická zařízení, která jsou založena na odlišném principu a řídí se jinými pravidly a přístupy než těmi v uvedené tabulce. Jedním z nich je humanoidní robot SAR (viz Kapitola 1.2.1). SAR je určen především k rehabilitaci dětí a umožňuje provádět fyzioterapii většiny částí lidského těla, včetně komplexních pohybů. V tomto případě robot neslouží jako fyzická podpora, vedoucí systém či diagnostická jednotka, ale pouze jako „učitel“. Robot disponuje 13 předprogramovanými cviky, které předvádí pacientovi a ten je dle robota poté vykoná. Robot není s pacientem v žádném fyzickém kontaktu a jedná se tak o fyzioterapii prováděnou pouze s vlastní pomocí.

Experimentální nástroj pro zvýšení motorické dovednosti paže

Výzkum, na kterém pracuje tým z univerzity v Melbourne slouží převážně pro experimentální účely. Jedná se o systém pro rehabilitaci paže [8] umožňující pacientovi pohyb ve 2D prostoru. Systém disponuje nástavcem, který slouží k manipulaci. Prostřednictvím tohoto nástavce je pacientovi umožněno zařízením pohybovat ve směru x a y. Uživatel dostane za úkol například nakreslit pomocí nástavce obrazec, nebo dosáhnout určitého bodu v rovině. Zařízení disponuje motory, které na nástavec během pohybu vyvíjejí sílu různými směry tak, aby bylo pacientovi dosažení určeného cíle znesnadněno. Pomocí senzorů umístěných na předloktí jsou měřeny svalové aktivity uživatele. Z těchto hodnot spolu s dráhou pohybu je poté vytvořen referenční model sloužící k porovnání s hodnotami obdržnými simulací.

Kolektivní Tele-rehabilitace

Výzkumem tele-rehabilitace [9] se zabýval mezinárodní tým, jehož členové pocházejí z univerzit zabývajících se biomedicínou ve Spojených státech amerických (Milwaukee)

a Velké Británii (Reading). Vyvinutá platforma je určena pro pacienty po mozkové příhodě a zaměřuje se na rehabilitaci horních končetin. Základní myšlenkou systému je vzájemná spolupráce dvou uživatelů, kteří od sebe mohou být vzdáleni tisíce kilometrů. Systém se skládá z rehabilitačního robota Haptic Master, ve kterém je uchycena pacientova paže. Robot disponuje šesti stupni volnosti a asistuje uživateli při pohybu. Platforma se dále skládá z monitoru, dvou kamer a herní desky pro hru Piškvorky. Kolečka a křížky jsou reprezentovány reálnými objekty, které pacient pokládá za pomoci robota do příslušných políček na herní desce. Dva uživatelé disponující identickou platformou hrají zmíněnou hru proti sobě. S pomocí kamer a mikrofonu spolu pacienti mohou komunikovat a zároveň sledovat herní pole protivníka. Hráč A zahájí hru položením křížku do vybraného pole na herní desce. Hráč B nejprve umístí na svém poli křížek na místo, na které ho položil hráč A, aby byla herní pole obou uživatelů identická, a teprve poté vykoná svůj tah objektem reprezentující kolečko. Tímto způsobem se piškvorky hrají, dokud hra nevyústí v libovolný výsledek. Během procesu je robotem zaznamenávána rychlost, síla a pozice paže pacienta. Uživatelé si zároveň zdokonalují komunikační schopnosti a celá hra je především pro starší a osamělé osoby díky interakci zábavná a motivující.

Přístupů k robotické rehabilitaci je mnoho a odlišný způsob od těch zmíněných využívá také rehabilitační aplikace, jejímž návrhem se zabývá tato práce. Fyzioterapie je prováděna za pomoci průmyslového robota a cviky jsou definovány fyzioterapeutem na míru pro každého pacienta zvlášť. Systém je určen primárně pro fyzioterapii horních končetin a umožňuje tak rehabilitovat ramenní a lopatkové svaly, nadloktí, předloktí, ale i zápěstí. Zmíněné partie lze posilovat izolovaně, zároveň je ale možné vytvořit komplexní cvik simulující například libovolnou každodenní činnost, při které se procvičují všechny problémové partie pacienta zároveň. Univerzálnost a vysoká přizpůsobitelnost je hlavní výhodou navržené platformy oproti zmíněným systémům.

1.5 Funkční popis rehabilitační aplikace

Hlavním prvkem systému pro rehabilitaci je výše popsaný průmyslový senzitivní robot LBR iiwa, který disponuje efektorom v podobě madla (viz Obrázek 7). Efektor slouží ke kopírování definované trajektorie robota pacientem. Madlo je vyrobeno z PVC trubic pomocí lepených spojů a T-kusu. Pro vyšší pevnost je celá vnitřní část vyplněna montážní pěnou. Nástavec připevněný k přírubě robota a zátky (oranžové části efektoru) jsou vytištěny pomocí 3D tiskárny. Druhým důležitým prvkem platformy je počítač připojený síťovým kabelem k řídicí jednotce robota. Aplikace v počítači slouží jako nástroj pro

ovládání celého systému. Posledním komponentem je smartPAD, pomocí kterého je možné definovat a parametrizovat trajektorii robota.



Obrázek 7: Systém pro rehabilitaci horních končetin

1.5.1 Hlavní funkce aplikace

1. Správa pacientů a cviků

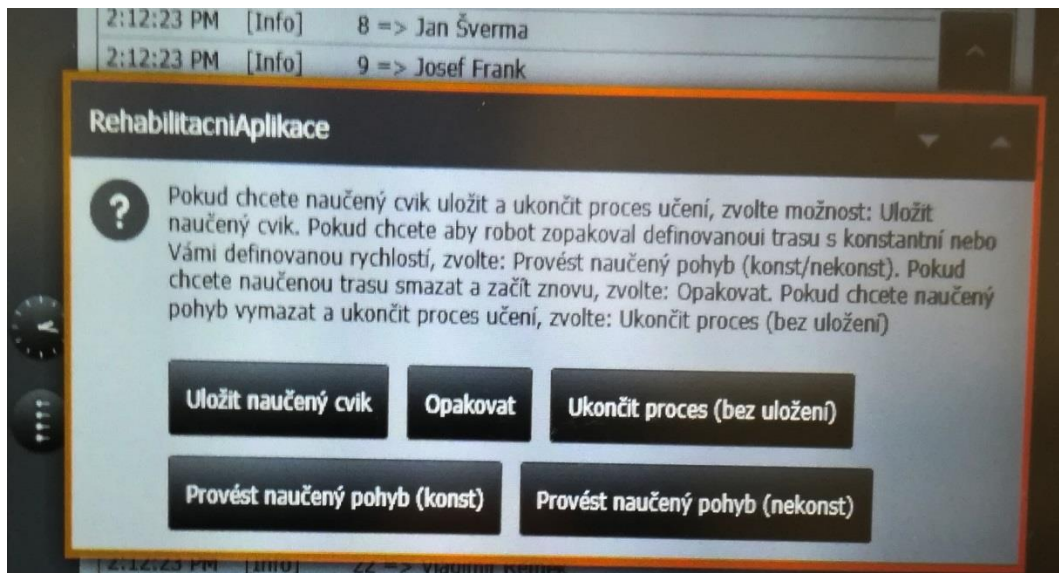
Tato funkce je podstatná pro přehlednou evidenci. Pomocí aplikace v počítači je možné založit složku novému pacientovi, k danému pacientovi přiřadit definovaný cvik, smazat složku pacienta nebo pouze odebrat cvik evidovaný u daného pacienta. Složky pacientů se v případě, že je počítač připojený k robotu a robot je zapnutý (dále jen on-line/off-line režim), vytvářejí i v řídicí jednotce robota, která tak slouží jako záloha. V opačném případě se operace provedené v aplikaci ukládají a v okamžiku přechodu do online režimu proběhne synchronizace složek pacientů a cviků. Složky jsou tak v on-line režimu duplicitní.

2. Definice trajektorie

V případě, že je robot v on-line režimu, je možné pomocí aplikace vyslat příkaz robotu k definici trajektorie pro nový fyzioterapeutický cvik. Tento krok je možné provést v okamžiku, kdy je označen pacient, ke kterému má být cvik přiřazen. Při této akci je po uživateli požadován název, pod kterým bude cvik evidován a také interval. Intervalem je myšlena vzdálenost, po jaké se budou zaznamenávat body při učení trajektorie. Tedy přesnost, s jakou bude trasa naučena. Minimální interval, a tedy nejvyšší přesnost je

1 cm, maximální možný interval je 20 cm. Vzdálenost je možné volit po jednom centimetru. Při této volbě se během definice ukládá pouze trasa, ale nikoli čas. Fyzioterapeutický cvik definovaný tímto způsobem je proto možné provádět pouze s konstantní rychlostí zvolenou uživatelem. Pro záznam času během definice trajektorie slouží volba „Definovat rychlost pohybu“. Při zaškrtnutí této položky je automaticky nastaven interval vzdáleností na 0,5 cm.

Po navolení parametrů a potvrzení dialogu nastává druhá část, která je již v kompetenci robota. Během této fáze je aplikace v počítači v procesním režimu, při kterém není možné provádět žádné operace, do chvíle, než robot odešle počítači informace o výsledku procesu. SmartPAD poté pomocí informativních dialogů vede fyzioterapeuta celým procesem definice trajektorie, při kterém je možné napolohovat robota do startovní pozice a pokud je to požadováno, určit osy robota, které nebudou během ručního navádění pohyblivé. Fáze samotného učení probíhá tak, že fyzioterapeut vede rukou rameno robota po požadované trajektorii, během které si robot zaznamenává úhly všech os a souřadnice koncového efektoru s požadovaným intervalem (a popřípadě časem) do souboru. Po ukončení ručního navádění jsou nabídnuty uživateli prostřednictvím dialogového okna možnosti (viz Obrázek 8). Možnosti „Uložit naučený cvik“ a „Ukončit proces (bez uložení)“ ukončí proces definice s úspěšným, respektive neúspěšným výsledkem. Dalšími možnostmi jsou „Provést naučený pohyb (konst)“ nebo „Opakovat“.



Obrázek 8: Dialogové okno po dokončení definice trajektorie

V prvním případě robot provede pohyb po právě naučené trajektorii s defaultní stanovenou rychlostí a následně se na smartPADu zobrazí opět zmiňované dialogové okno. Volba „Opakovat“ umožňuje zopakovat celý proces definice trajektorie. V tomto případě je

možné navolit znovu všechny parametry, včetně nastavení intervalu vzdáleností, uzamykatelnosti os a napolohování robota do nové zahajovací pozice. Vše je nastavitelné pomocí smartPADu, jelikož aplikace v počítači je stále v procesním režimu a čeká na výsledek procesu. Po opakovaném procesu učení se původní definovaná trajektorie vymaže a uloží se nově naučená trasa. Následně je zobrazeno opět dialogové okno. V případě, že trajektorie byla definována včetně záznamu času, nabídne menu uživateli pátou možnost, kterou je „Provést naučený pohyb (nekonst)“. Zvolením této možnosti robot zopakuje pohyb po právě definované trajektorii s ohledem na rychlost, s jakou bylo rameno během definice vedeno.

Po ukončení celého procesu je aplikace v připojeném počítači přepnuta opět do uživatelského režimu. V tuto chvíli je možné opět s aplikací pracovat a v případě úspěšně dokončeného procesu je cvik zobrazen pod zadaným názvem ve složce daného pacienta.

3. Provedení fyzioterapeutického cviku

Zahájení fyzioterapie je provedeno prostřednictvím aplikace v počítači. Po označení pacienta a příslušného cviku, je možné spustit okno s nastavitelnými parametry. Nastavit je možné počet opakování, rychlost pohybu, tuhost os a prahovou hodnotu. Počet opakování znamená, kolikrát robot zopakuje fyzioterapeutem definovanou trajektorii. Míru tuhosti, lze nastavit pro každou osu robota zvlášť nebo také jedním parametrem definovat poddajnost pro všechny osy stejnou. Čím vyšší tuto hodnotu fyzioterapeut nastaví, tím benevolentněji se robot při provádění trajektorie chová a nemusí tak efektně přesně dosáhnout definovaných souřadnic. Při vyšší poddajnosti lze silou působící na robota ovlivnit do jisté míry jeho pohyb. Tato funkce má využití v případě, kdy například pacient nemá potřebné pohybové schopnosti k tomu, aby zvládl provést trasu přesně tak, jak ji fyzioterapeut definoval. Ve chvíli, kdy pacient bude zaznamenávat pokroky, tuhost os lze zvyšovat, až nakonec pacient bude nucen kopírovat pohyb robota přesně tak, jak byl definován fyzioterapeutem. Dalším parametrem je prahová hodnota. Tento parametr se nastaví v případě provádění izolovaného cviku. Izolovaným cvikem je myšlen cvik zaměřený na jeden konkrétní sval. V případě ochablého ramenního svalu je cvik zahájen v určité pozici, ve které má pacient ještě nějaký pohybový potenciál. Po zahájení fyzioterapie robot začne vykonávat pohyb směrem, ve kterém se bude pacientova schopnost pohybu kvůli ramennímu svalu snižovat. To bude mít za následek zvyšující se sílu, kterou pacient bude působit na madlo robota. Po překročení prahové hodnoty nastavené fyziote-

rapeutem se pohyb robota plynule zastaví a vrátí zpět do zahajovací pozice. Prahová hodnota tedy definuje hranici síly, při které se ukončí dané opakování cviku. Opakování může být i v tomto případě nastaveno více. Prahová hodnota a pohybová schopnost pacienta určí, kolik procent cviku v každém opakování bude provedeno. Posledním parametrem je „Definovaná rychlost“. Tato možnost je nastavitelná pouze v případě, že cvik, který bude prováděn byl definován i se záznamem času. Při zaškrtnutí tohoto parametru je trajektorie provedena s ohledem na rychlost, jakou bylo rameno během definice cviku vedeno.

Po nastavení těchto parametrů a následného potvrzení dialogu robot dostane příkaz k provedení zvoleného cviku a zaujme počáteční pozici. Při tomto pohybu je ošetřena bezpečnost nastavením detekce kolize. V případě, že robot zaznamená kolizi, přeruší pohyb. Opětovným vyvinutím síly na jakoukoli část robota pohyb po dvou vteřinách opět pokračuje, dokud nedosáhne startovní pozice cviku. V počáteční pozici robot opět čeká na signál v podobě síly vyvinuté na efektor robota. Jakmile bude pacient připraven, vyvine sílu jakýmkoli směrem na madlo, které drží rukou a robot začne provádět pohyb po definované trajektorii daného cviku. Pokud cvik nebude schopen zahájit pacient, sílu na robota vyvine fyzioterapeut.

4. Vyhodnocení průběhu fyzioterapie

Při provádění fyzioterapeutického cviku jsou měřeny síly, kterými pacient působí na efektor robota. Jedná se o síly ze směrů x, y, z a o součet těchto sil. Síly jsou měřeny v jednotkách [N]. Síly, kterými pacient působí na madlo robota, je možné sledovat v reálném čase během provádění cviku pomocí aplikace na připojeném počítači. Po ukončení fyzioterapie je automaticky vytvořen záznam. Tento záznam je uložen do souboru s pacientovou složkou. Aplikace pracuje celkem se dvěma druhy záznamových souborů. První záznamový soubor obsahuje složky pacientů s příslušnými definovanými trajektoriemi cviků. Druhý soubor obsahuje složky pacientů obsahující záznamové listy s výsledky provedených cviků. Pokud u daného pacienta prozatím není veden žádný výsledkový záznam, po provedení prvního cvičení se tomuto pacientovi automaticky vytvoří složka s jeho jménem, obsahující první záznamový list. Tento list má podobu tabulkového editoru Microsoft Excel. Pokud pacient provádí jeden a ten samý cvik jen například s jinými parametry vícekrát, všechny záznamy z tohoto cviku se u daného pacienta ukládají do stejného Excel souboru. Po každém cvičení je v tomto souboru vytvořen nový sešit s hodnotami, nazvaný aktuálním datem, ve kterém byl záznam vytvořen. Excelový soubor nese

název příslušného cviku. V případě, že daný pacient začne procvičovat jinou část těla, vytvoří se nový excelový soubor nesoucí název daného cvičení

Každý záznam obsahuje vzorky, pomocí kterých jsou číslovány jednotlivé údaje. Každý údaj (řádek tabulky) obsahuje číslo vzorku, součet sil (x, y, z) a jednotlivé složky sil. Každé opakování cviku je rozpoznatelné pomocí sloupce v záznamu, který obsahuje procenta daného opakování. Z této tabulky je možné jednoduše vytvořit přehledný graf, ze kterého lze vyčíst všechny podstatné informace pro vyhodnocení fyzioterapie a sledování progresu. V případě fyzioterapie s definovanou prahovou hodnotou je v každém záznamu navíc vytvořena tabulka, která obsahuje informace o každém opakování daného izolovaného cviku. Počet řádků této tabulky se rovná počtu opakování provedeného cviku. Každý řádek obsahuje číslo opakování cviku a číslo představující počet procent dosažených v daném opakování. Procenta reprezentují fázi, do které byl pacient schopen kopírovat pohyb robota, aniž by síla vyvíjená pacientem na madlo překročila prahovou hodnotu. V případě, že je pacient schopen provést kompletní cvik definovaný terapeutem, bude v této kolonce 100 %, v opačné situaci méně. Hodnoty se posléze zprůměrují a výsledkem bude hodnota reprezentující procentuální výsledek všech opakování daného cviku.

Dále každý záznam provedeného cvičení obsahuje tabulku s informacemi o parametrech, s jakými byla fyzioterapie prováděna. Jedná se o počet opakování, rychlost pohybu, tuhost každé osy a v případě izolovaného cviku navíc o výši prahové hodnoty, která je v jednotkách [N]. Z těchto údajů lze tak přehledně záznamy vyhodnocovat a sledovat pokroky pacientů.

5. Funkce dostupné v off-line režimu

Off-line režim znamená, že spolu v dané chvíli robot a počítač nekomunikují. To může být zapříčiněno odpojením síťového kabelu, vypnutím robota, ukončením aplikace v počítači nebo neočekávaným technickým problémem. Aplikace v počítači umožňuje v off-line režimu zakládat složky s novými pacienty, vymazat složky s pacienty, odstranit definované cviky u příslušných pacientů a odstranit složky pacientů i se všemi příslušnými cviky. Je možné tedy provádět všechny úkony, které zahrnuje správa cviků a pacientů. Poté, co se naváže spojení s robotem, proběhne synchronizace a všechny tyto úkony se provedou ihned v druhém záznamovém souboru uloženém v řídicí jednotce robota.

V případě robota je v off-line režimu umožněno fyzioterapeutovi definovat novou trajektorii cviku. Po spuštění rehabilitační aplikace v robotu se na smartPADu zobrazí

číslovaný seznam složek všech pacientů, kteří jsou v záznamech vedeni. Dále je možno uživateli jednoduše pomocí čísla daného pacienta zobrazit jemu příslušející cviky. Tato funkce slouží pouze pro přehled, pokud by si fyzioterapeut nebyl jistý, které cviky jsou již vytvořené a které ne. V momentě, kdy se uživatel rozhodne definovat novou trajektorii, zadá pomocí smartPADu číslo pacienta uvedené v zobrazeném seznamu, zvolí název cviku a také interval vzdáleností. Pokud je zvolený interval roven nule, bude se během ručního navádění zaznamenávat také čas a interval bude nastaven automaticky na 0,5 cm. Po potvrzení se spustí proces definice (popsaný v části 2. Naučení nového cviku), bez použití připojeného počítače. Po ukončení procesu se ve složce daného pacienta uložené v řídicí jednotce robota vytvoří textový soubor s právě definovanou trajektorií a v okamžiku přechodu do online režimu se provede ihned automaticky synchronizace a veškeré změny se projeví i v záznamovém souboru uloženém v počítači.

2 Rehabilitační aplikace s robotem LBR iiwa

Tato kapitola se zabývá návrhem rehabilitační aplikace s využitím senzitivního robota KUKA LBR iiwa. Popis návrhu softwaru je hierarchicky rozdělen tak, aby vystihl posloupnost, v jaké byla aplikace vytvořena.

2.1 Úvod do programování

K vývoji aplikací pro robota LBR iiwa slouží vyšší programovací jazyk Java s přidávanými knihovnami. Tyto knihovny poskytují programátorovi mnoho nových funkcí, které jsou určeny výhradně pro programování robota. Základní příkaz pro pohyb robota do určeného bodu vypadá následovně:

```
robot.move( ptp( getApplicationData().getFrame("P1")) );
```

robot – objekt, pro který se při pohybu robota vypočítává dráha. Defaultně je za tento objekt považována příruba robota. Její střed je poté označován jako „tool center point“, což je počátek souřadného systému, ve kterém se bude robot pohybovat. Obvykle robot disponuje nástrojem, nebo také objektem, který je uchopen nástrojem robota. Pro tyto objekty lze poté nadefinovat libovolný „tool center point“, který bude označen jako nový počátek souřadného systému. Pro tyto nadefinované objekty poté platí příkaz `vlastni-Objekt.move(...)`.

move – klíčové slovo pro vykonání pohybu.

point to point (PTP) – způsob, kterým robot zvoleného bodu dosáhne. PTP nebere v úvahu dráhu robota, ale pouze pohyb ramen tak, aby robot vynaložil co nejmenší „úsilí“ pro daný pohyb. Pokud záleží na trajektorii koncového efektoru robota, je na výběr buď kruhová (`circ`) nebo lineární (`lin`) dráha. Pro definici posloupnosti pohybů slouží blok s názvem `spline`. Pomocí tohoto bloku lze kombinovat zmíněné typy trajektorií a vytvořit z nich jeden komplexní pohyb.

getFrame – jako parametr přijímá souřadnice, kterých má „tool center point“ dosáhnout.

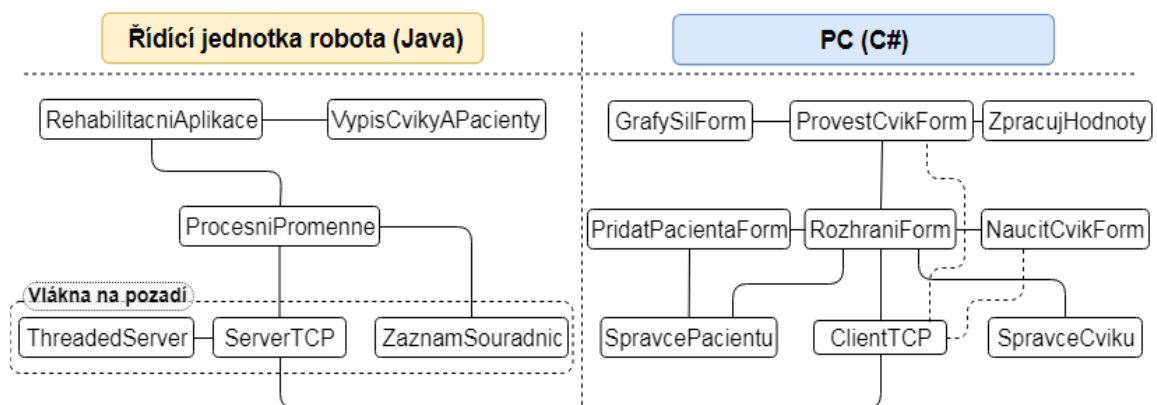
Pro zápis algoritmů slouží vývojové prostředí KUKA Sunrise Workbench. Celá komunikace probíhá tak, že počítač je připojen síťovým kabelem k řídicímu kabinetu robota, do kterého se z počítače vytvořený program nahrává. Nahraná aplikace je posléze viditelná na SmartPADu a lze jí spustit. Komunikace probíhá i obráceně. Pomocí SmartPADu lze změnit pro daný program souřadnice bodů (trajektorii) a takto upravený program rovnou spustit, nebo posléze nahrát zpět do počítače. Program v počítači je přepsán a nahrazen tím z řídicího kabinetu.

Program, který slouží jako uživatelské rozhraní rehabilitační aplikace v připojeném počítači je vytvořen v programovacím jazyce C#. Jedná se o vyšší programovací jazyk, který disponuje mnoha funkcemi a možnostmi. Jednou z možností je vývoj grafických aplikací za pomoci oken a formulářů, které poskytuje Microsoft Windows. Pro zápis algoritmů a přístup ke zmíněným grafickým prvkům slouží vývojové prostředí Microsoft Visual Studio. Jelikož aplikace v počítači slouží jako uživatelské rozhraní, možnost vývoje grafické aplikace byla hlavním kritériem při volbě programovacího jazyka a vývojového prostředí. Jedná se tedy celkem o dva vzájemně komunikující programy – program v robotu (Java) a program v počítači (C#). Ve výsledku tvoří jednu kompaktní aplikaci, která dle potřeb reaguje na požadavky uživatele a volí dle dané situace prostředek (PC – robot) pro svůj vstup a výstup.

Pro účely testování v okamžiku, kdy není robot dostupný, slouží virtuální počítač. Ten je vytvořen pomocí software VirtualBox s nainstalovaným vývojovým prostředím NetBeans. Prostřednictvím tohoto systému lze do jisté míry simulovat chování robota a připravovat si tak podprogramy pro následné testování s připojeným robotem.

2.1.1 Architektura programu a TCP komunikace

Architektura programu v robotu je složena z několika vrstev (viz Obrázek 9). Vrchní vrstvu reprezentuje třída `RehabilitacniAplikace`, která obsahuje veškeré metody pro vykonání pohybu ramene a interakci s uživatelem. Spodní vrstvu reprezentují vlákna, která jsou spuštěna na pozadí. Jedná se o třídy pracující cyklicky, nepřetržitě a nezávisle na vrchní aplikační vrstvě. Tato vrstva obsahuje vlákna s procesy, u kterých je žádoucí paralelní činnost s vrchní vrstvou programu. Při definici nové trajektorie robota je třeba obsloužit ruční navádění ramene a zároveň zaznamenávat souřadnice právě definované trajektorie do souboru.



Obrázek 9: Architektura softwaru (třídy)

Proces ručního navádění je obslužen ve třídě `RehabilitacniAplikace` a záznam souřadnic ve zmíněné paralelní úloze na pozadí. Druhý proces, u kterého je žádoucí paralelní činnost, obsluhuje síťovou Transmission Control Protocol (dále jen TCP) komunikaci. Třída `ProcesniPromenne` obstarává předávání hodnot proměnných sdílených mezi vrchní a spodní vrstvou programu.

Grafická aplikace v počítači obsahuje jeden hlavní formulář s názvem `RozhraniForm`. Tento formulář slouží jako rozcestník aplikace a při jejím spuštění se zobrazí jako první. Třída `RozhraniForm` je propojena se všemi ostatními prvky programu. Formulářové třídy (`PridatPacientaForm`, `NaucitCvikForm`, `ProvestCvikForm`, `GrafySilForm`) slouží především pro interakci s uživatelem, zobrazení zpracovaných dat a pro zadávání nových dat ke zpracování. Ostatní třídy slouží pro výpočty, zpracování dat, správu souborů a také pro komunikaci s robotem.

Komunikaci mezi programem v řídicí jednotce robota a programem v počítači zajišťují protokoly TCP/IP. Tato komunikace je založena na výměně datagramů mezi serverem a klientem. Aplikace v počítači vykonává funkci klienta, zatímco řídicí jednotka robota má funkci serveru. Server i klient musí mít přidělenou IP adresu ze stejného rozsahu a také určený síťový port, na kterém budou obě zařízení komunikovat. IP adresy obou zařízení jsou voleny z rozsahu 172.31.1.xxx a rozsah portů použitelných pro TCP/IP komunikaci je omezen robotem na 30 000 až 30 010.

Server

Server v robotu obstarává úloha spustitelná na pozadí s názvem `ServerTCP`. Tato úloha je vytvořena třídou `ThreadedServer`, která je rovněž spuštěna jako paralelní úloha na pozadí. Po spuštění serverového vlákna je vlákno `ThreadedServer` usnáváno. Třída `ServerTCP` nejprve otevře socket pomocí metody `openServerSocket()` a poté na tomto socketu „naslouchá“ a čeká na připojení klienta. V dalším kroku jsou inicializované proměnné `inFromClient/outToClient`, které slouží pro čtení posloupnosti bytů od klienta a zápis posloupnosti bytů klientovi. Tyto proměnné jsou využity v metodách pro komunikaci s klientem (viz Zdrojový kód 1). Klient zašle serveru dva datagramy. První samostatný datagram obsahuje číslo, reprezentující počet znaků v druhém datagramu, který obsahuje samotná data. Nejprve je alokováno pole o velikosti čtyři byty, a následně je přijat první datagram obsahující počet znaků v druhém datagramu. V dalších krocích je přijat druhý datagram obsahující požadovaná data, která jsou nakopírována do předem alokovaného pole s již přesnou velikostí, zjištěnou z prvního datagramu. V posledním

kroku je provedena konverze dat spolu s definovaným kódováním, které zajistí například správnou reprezentaci alfabetských znaků s diakritikou.

```
public String pozadavekOdKlienta() throws IOException {
    byte[] lenBytes = new byte[4]; // alokace pole bytů
    inFromClient.read(lenBytes, 0, 4); // zjištění počtu znaků v přijímané zprávě
    int len = (((lenBytes[3] & 0xff) << 24) | ((lenBytes[2] & 0xff) << 16)
        | ((lenBytes[1] & 0xff) << 8) | (lenBytes[0] & 0xff));
    byte[] receivedBytes = new byte[len]; // alokace pole pro přijímané znaky
    inFromClient.read(receivedBytes, 0, len); // načtení příchozích znaků do pole
    String s = new String(receivedBytes, 0, len); // konverze byte/string
    String received = new String(receivedBytes, "UTF-8"); // překódování řetězce na UTF-8
    return received;
}
```

Zdrojový kód 1: Metoda pro přijímání dat od klienta

Server je snadno rozšiřitelný k obsluze mnoha klientů. Třída `ThreadedServer`, která nyní spouští jedno vlákno představující TCP server, může takovýchto vláken vytvořit a spustit více. V dané chvíli je tedy teoreticky možné, aby robot komunikoval souběžně s vícero klienty najednou.

Klient

V počítačové aplikaci obstarává síťovou komunikaci metoda `sitovePripojeni()`, kterou obsahuje třída s názvem `ClientTCP`. V okamžiku, kdy uživatel aplikaci spustí, zavolá se metoda `sitovePripojeni()`, která se pokusí připojit k serveru. V případě, že se spojení podaří navázat, proces spojování se ukončí. V opačném případě se příkazem `clientSocket.Close()` socket uzavře a aplikace je přepnuta do režimu off-line. Metoda `sitovePripojeni()` je rozdělena na dvě hlavní části podmínkou, která testuje globální proměnnou s názvem `Pripojeno`. Pokud má tato proměnná hodnotu „true“, znamená to, že při posledním testování bylo spojení se serverem aktivní. Pro tuto situaci je vykonán sled příkazů, jež testují, zda je spojení stále aktivní. Pokud ano, ponechají proměnné hodnotu „true“, v opačném případě se proměnné přiřadí hodnota „false“. Pokud má testovaná proměnná `Pripojeno` hodnotu „false“, znamená to, že při posledním testování nebylo spojení se serverem aktivní a aplikace byla v režimu off-line. V tomto případě se vykoná sled příkazů pro navázání spojení popsanych výše a dle výsledku přiřadí této proměnné opět příslušnou hodnotu. Metoda `sitovePripojeni()` je volána okamžitě při spuštění aplikace a poté automaticky pomocí časovače každé tři vteřiny. V tomto intervalu je prostřednictvím globální proměnné testováno síťové připojení v rámci celé aplikace. Uživatel má v reálném čase informace o stavu spojení díky pře-

hledné vizualizaci. Testování spojení je využito také k účelům zamítnutí/povolení přístupů do sekcí programu, u kterých záleží na tom, v jakém režimu se aplikace aktuálně nachází.

Metoda `pozadavekNaServer()` (viz Zdrojový kód 2) slouží pro odesílání dat serveru. Tato metoda nejprve přijme data v podobě textového řetězce a tento řetězec následně převede na počet znaků. Tento údaj je odeslán prvním datagramem serveru. Druhý odeslaný datagram obsahuje již přímo zakódovaný řetězec v podobě jednorozměrného pole o zjištěné délce ve formátu byte.

```
public void pozadavekNaServer(string jmenoPacienta, string nazevCviku, int interval)
{
    string pozadavek = "naucitCvik"+"=>"+jmenoPacienta+"=>"+nazevCviku+"=>"+interval;
    int delkaRetezce = System.Text.Encoding.UTF8.GetByteCount(pozadavek);
    byte[] toSendBytes = System.Text.Encoding.UTF8.GetBytes(pozadavek);
    byte[] toSendLenBytes = System.BitConverter.GetBytes(delkaRetezce);
    pripojeni.clientSocket.Send(toSendLenBytes);
    pripojeni.clientSocket.Send(toSendBytes);
}
```

Zdrojový kód 2: Metoda pro zaslání dat serveru

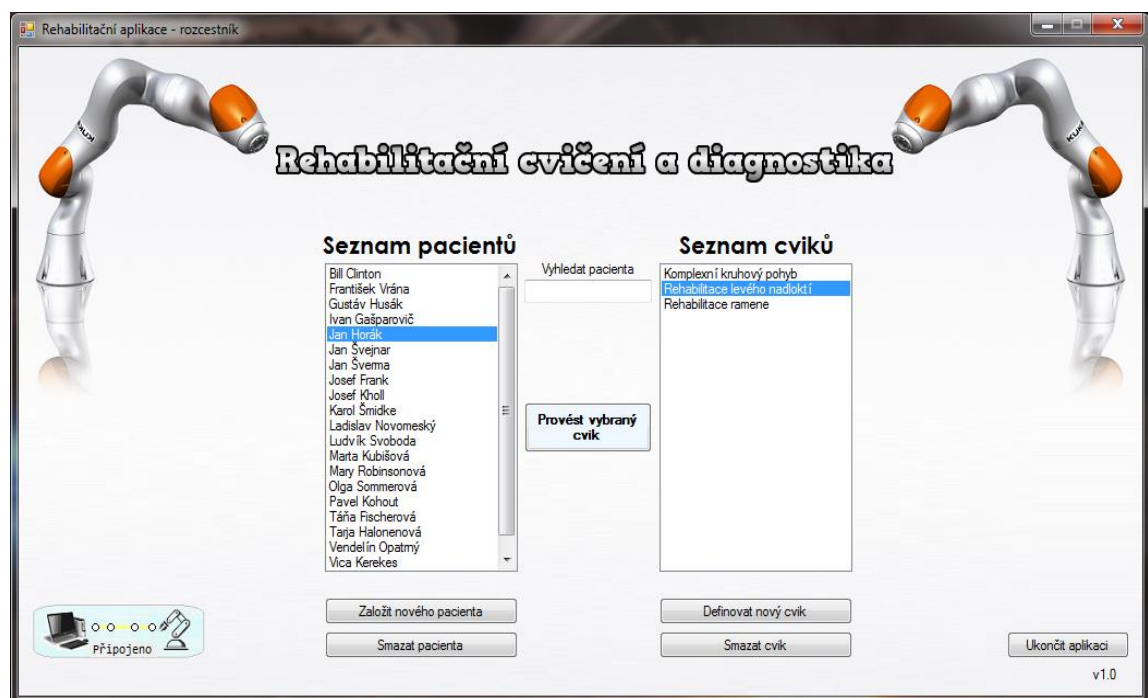
Tyto datagramy jsou přijaty zmíněnou metodou `pozadavekOdKlienta()` (viz Zdrojový kód 1). Komunikace opačným směrem je identická.

2.2 Správa cviků a pacientů

Pro správu pacientů slouží adresář se složkami, pojmenovaných dle pacientů. Složky pacientů obsahují textové soubory s úhly os robota a souřadnicemi, reprezentujícími trajektorii cviku. Každý textový soubor nese název fyzioterapeutického cviku, jehož trajektorie je v daném souboru uložena. Adresář s pacienty a cviky je vytvořen v připojeném počítači i v robotu. Aby byly tyto dva adresáře duplicitní, je třeba vytvořené trajektorie cviků a založené složky pacientů ukládat do obou adresářů zároveň. Pro založení složky nového pacienta slouží formulář vyvolaný možností „Založit nového pacienta“ (viz Obrázek 10). Formulářová třída pro založení nového pacienta komunikuje se třídou `SpravcePacientu`. Tato třída obsahuje metody pro správu adresářů nazvané `pridejPacienta()` a `odeberPacienta()`. Je-li aplikace v on-line módu, metody vytvoří, respektive odeberou vybranou složku (pacienta) také v adresáři umístěném v řídicí jednotce robota. Jestliže je aplikace v off-line módu, je možné tyto operace (založit a odebrat pacienta) provádět též, ale pouze v záložním adresáři umístěném v počítači. Zároveň si aplikace všechny vykonané operace ukládá do mezipaměti v podobě textového souboru a v oka-

mžiku přechodu aplikace do on-line módu se tyto operace automaticky pomocí synchronizace provedou i v adresáři umístěném v robotu. Metoda pro synchronizaci adresářů nese název `synchronizuj()` a je obsažena ve třídě `ClientTCP()`. Tato metoda je zavolána pokaždé, když je navázáno spojení se serverem a aplikace přejde z off-line módu do on-line módu. Funkce se skládá z příkazů, které nejprve za pomoci mezipaměti zjistí, zda byly provedeny nějaké operace, když byla aplikace v off-line módu. Pokud ano, provede tyto operace i v adresáři v robotu a mezipaměť se poté vymaže. Dále metoda zjistí tyto informace také o adresáři v robotu a je-li to třeba (v off-line módu byl robotem definován nový cvik), provede synchronizaci znovu opačným směrem. Je-li aplikace v on-line módu, jsou adresáře vždy duplicitní, což je důležité pro definici nové trajektorie, ale i pro provádění již definovaných cviků.

Adresář se složkami pacientů je načten ihned po spuštění aplikace. Evidování pacienti jsou zobrazeni v přehledném seznamu (viz Obrázek 10). Pro případ velkého množství evidovaných pacientů slouží vyhledávač. Vyhledávání není třeba potvrzovat, jelikož vyhledávač uživateli automaticky po každé změně ve vyhledávacím poli nabízí prozatímní shody se zadanými znaky. Vyhledávání funguje fulltextově, proto stačí, aby byl zadaný sled znaků obsažen v jakékoli části pacientova jména nebo příjmení. Při vyhledávání v seznamu zůstávají vždy pouze relevantní výsledky, přičemž ty nevyhovující jsou skryty. V okamžiku odstranění obsahu vyhledávacího pole se seznam pacientů opět vrátí do původního stavu.

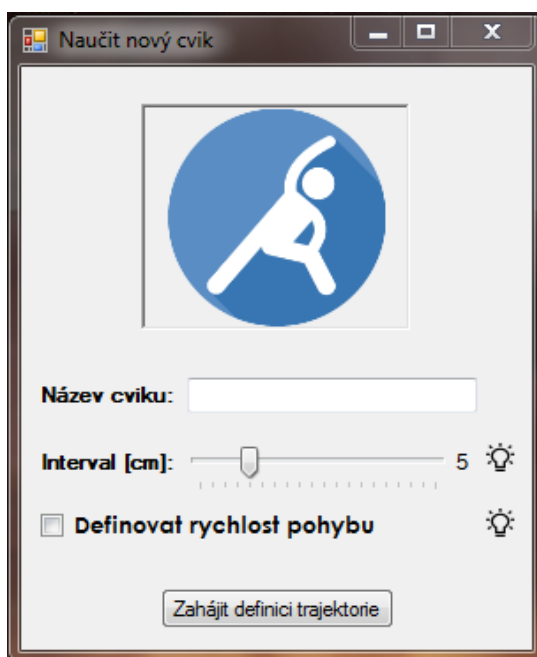


Obrázek 10: Hlavní okno uživatelského rozhraní

Je-li aplikace v off-line módu, seznam pacientů je načten ze záložního adresáře v počítači. Při přechodu aplikace do on-line režimu se provede synchronizace a seznam pacientů je načten z adresáře umístěného v řídicí jednotce robota. Metody pro načítání adresářů do seznamu jsou uloženy v hlavní třídě nazvané `RozhraniForm`. Po označení libovolného pacienta je zavolána metoda pro výpis cviků definovaných u vybrané položky. Názvy cviků jsou poté ihned automaticky vypsány v druhém seznamu nazvaném „Seznam cviků“. Označováním položek v seznamu pacientů lze tak jednoduše prohlížet definované cviky daných pacientů a provádět s nimi povolené operace.

2.3 Mechanismus pro definici trajektorie

Podstatou částí rehabilitační aplikace je možnost definice nového fyzioterapeutického cviku. Nový cvik je možné definovat pomocí uživatelského rozhraní v připojeném počítači, ale také pouze za pomoci robota bez použití počítače. V prvním případě je nutné, aby aplikace byla v módu on-line. Dalším krokem je označení pacienta, u kterého bude fyzioterapeut definovat nový cvik, a nakonec je třeba zvolit položku „Definovat nový cvik“. Ve vyvolaném formuláři (viz Obrázek 11) uživatel zvolí název vytvářeného cviku a také způsob zaznamenávání souřadnic.



Obrázek 11: Okno pro zahájení definice trajektorie

Po potvrzení formuláře se data odešlou robotu a aplikace v počítači je přepnuta do režimu „čekání“, při kterém uživateli není umožněno v aplikaci provádět žádné operace.

2.3.1 Zpracování požadavku od klienta

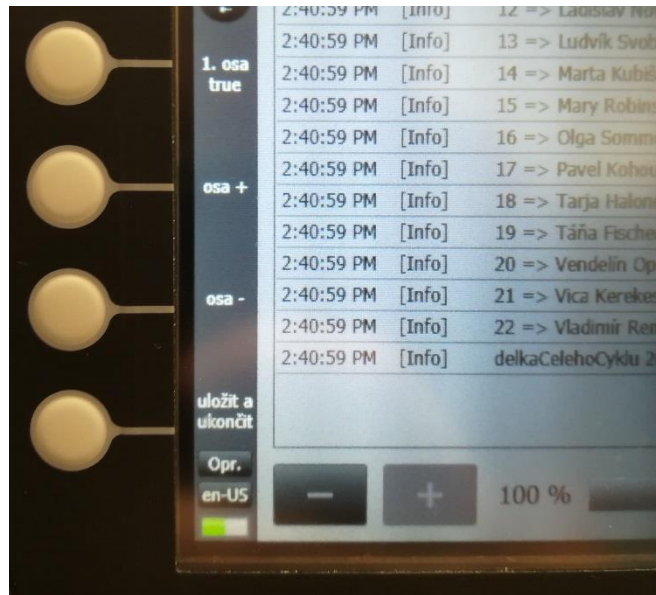
Veškerá data od klienta jsou přijímána úlohou na pozadí nazvanou `ServerTCP`, která je součástí aplikace v robotu. Tato třída obsahuje metodu `zpracujPozadavky()`, která slouží pro přijímání a zpracování požadavků od klienta. Požadavek od klienta sestává z klíčového slova a z parametrů, které jsou k danému požadavku potřebné. Požadavek pro definici nového cviku vypadá následovně:

```
string pozadavek = "naucitCvik"+"=>" + jmenoPacienta + "=>" + na-  
zevCviku + "=>" + interval;
```

Dle klíčových slov zmíněná metoda požadavky od klienta rozřazuje a přiřazuje jejich parametry příslušným proměnným, které jsou uloženy ve třídě `ProcesniPromenne`, reprezentující mezivrstvu programu. Klíčové slovo „naucitCvik“ je přiřazeno jako nová hodnota procesní proměnné s názvem `typPozadavku`. Hodnotu této proměnné cyklicky sleduje metoda obsažená ve třídě s názvem `ZaznamSouradnic`. V případě, že je hodnota proměnné rovna klíčovému slovu pro definici nové trajektorie, zavolá se metoda rovněž obsažená v již zmíněné úloze na pozadí s názvem `zaznamSouradnic()`. Tato metoda slouží k záznamu pozic os a souřadnic robota, během definice nové trajektorie. Záznam souřadnic je spuštěn v okamžiku, kdy uživatel zahájí ruční navádění robota. Ruční navádění robota je obslouženo v metodě `definujTrajektorii()`, která náleží třídě `RehabilitacniAplikace`. Tato metoda provede uživatele celým procesem definice nové trasy. Nejprve je uživatel dialogovým oknem vyzván, aby za pomoci ručního navádění umístil rameno robota do startovní pozice fyzioterapeutického cviku a následně určil osy, které při definici trajektorie nebudou pohyblivé. Uživatel má na výběr ze dvou možností a pokud zvolí, že pohyblivost os definovat nechce, bude aktivováno defaultní nastavení, které ponechá všem osám maximální možný rozsah pohybu.

2.3.2 Mechanismus pro uzamykatelnost os robota

Pro možnost definice pohyblivosti robota je vyvoláno uživatelské prostředí pro uzamykatelnost os, které je vytvořeno pomocí uživatelských tlačítek na smartPADu (viz Obrázek 12). Prostřednictvím druhého a třetího tlačítka uživatel přepíná jednotlivé osy. Prvním tlačítkem vybranou osu označí jako „true“ (pohyblivá) nebo „false“ (nepohyblivá). Pro uložení a ukončení celého procesu slouží spodní (čtvrté) tlačítko.



Obrázek 12: Uživatelské prostředí pro uzamykatelnost os

Metoda, která zajišťuje uzamknutí/odemknutí vybrané osy (viz Zdrojový kód 3) reaguje na událost, kterou je stisknutí prvního uživatelského tlačítka. Pomocí pole si metoda pamatuje předchozí stav dané osy. Pokud uživatel vyvolal tuto událost a daná osa byla ve stavu „true“, přepne se nyní do stavu „false“ (nepohyblivá). Jestliže je osa ve stavu „false“, zjistí se za pomoci příkazu `getCurrentJointPosition()` aktuální pozice osy (v radiánech) a uloží se do pole. Proměnná `motion` je typu `HandGuidingMotion` a slouží jako parametr pro ruční navádění robota. Této proměnné se přiřadí parametry, které nastavují každé ose povolený rozsah pohybu.

```

IUserKeyListener myUserKeyListener3 = new IUserKeyListener() {
    @Override
    public void onKeyEvent(IUserKey key, UserKeyEvent event) {
        if (event.equals(UserKeyEvent.KeyDown)) {
            stavOs[i] = !stavOs[i];
            if(stavOs[i] == false){
                JPosition = lbr.getCurrentJointPosition();
                limitJ[i] = JPosition.get(i-1);
            }
            popisOs = i+". osa "+stavOs[i];
            myUserKey3.setText(UserKeyAlignment.Middle, popisOs);
            popisOs = "";
            motion.setAxisLimitsMax(limitJ[1]+konst, limitJ[2]+konst, limitJ[3]+konst, limitJ[4]+konst,
                limitJ[5]+konst, limitJ[6]+konst, limitJ[7]+konst)
                .setAxisLimitsMin(limitJ[1]-konst, limitJ[2]-konst, limitJ[3]-konst, limitJ[4]-konst,
                limitJ[5]-konst, limitJ[6]-konst, limitJ[7]-konst)
                .setAxisLimitsEnabled(!stavOs[1], !stavOs[2], !stavOs[3], !stavOs[4],
                !stavOs[5], !stavOs[6], !stavOs[7])
                .setAxisLimitViolationFreezesAll(false)
                .setPermanentPullOnViolationAtStart(true);
        }
    }
};

```

Zdrojový kód 3: Metoda pro nastavení pohyblivosti os

Metoda `setAxisLimitMax(a1, ..., a7)` přijímá maximální hodnoty úhlů pro každou osu robota v jednotkách [rad]. Spodní hranice rozsahu je nastavena pomocí metody `setAxisLimitMin(a1, ..., a7)`.

Aktuální zjištěná pozice zvolené osy je uložena do obou zmíněných parametrů. V parametru pro vrchní hranici rozsahu je k této hodnotě úhlu přičtena hodnota proměnné `konst`, která má hodnotu 0.001 rad a od spodní hranice rozsahu je tato hodnota naopak odečtena. Vůle dané osy je tak 0.002 rad, což je zanedbatelná hodnota a vybraná osa se tak jeví jako nepohyblivá. Třetím parametrem proměnné `motion` je metoda `setAxisLimitEnable(a1, ..., a7)`, která přijímá hodnoty „true“ nebo „false“. Pokud je například pro osu „a1“ zvolena hodnota „true“, znamená to, že má být definovaný rozsah pohybu pro danou osu uplatněn. V opačném případě se na zvolený rozsah nebere ohled a robot ponechá dané ose maximální rozsah povolený hardwarem.

2.3.3 Ruční navádění a záznam hodnot

Poté, co uživatel uloží a ukončí proces nastavení pohyblivosti os, může zahájit samotnou definici trajektorie pomocí ručního navádění. Na tuto možnost uživatele upozorní také modře svítící LED pásek na přírubě robota. Ruční navádění je zahájeno stisknutím a držením bílého tlačítka umístěného taktéž na přírubě. Tuto akci obsluhuje příkaz `lbr.move(motion)`. Metody `definujTrajektorii()` a `zaznamSouradnic()` spolu komunikují pomocí procesních proměnných. V okamžiku aktivace ručního navádění se v metodě `zaznamSouradnic()` spustí mechanismus pro záznam údajů. Mechanismus cyklicky zjišťuje pozice os i souřadnice příruby robota. Za pomoci zjištěných souřadnic metoda vypočítává v každém cyklu vzdálenost mezi dvěma body v prostoru. Pokud je vzdálenost rovna nebo větší než interval požadovaný uživatelem, aktuální souřadnice se uloží do proměnných a slouží jako nový počáteční bod, od kterého se bude v dalších cyklech znovu vypočítávat vzdálenost. V tento okamžik se zároveň uloží aktuálně zjištěné pozice os a kartézské souřadnice příruby robota do pole, přičemž je také změněn stav LED pásku na přírubě robota. LED pásek proto bliká frekvencí identickou s frekvencí záznamu bodů trajektorie pro lepší představu uživatele. V případě definice trajektorie se záznamem času je interval automaticky nastaven na 0,5 cm a při zahájení definice je zjištěn aktuální čas v milisekundách. Po dosažení intervalu se znovu pomocí příkazu `System.currentTimeMillis()` zjistí aktuální čas, od kterého je následně odečtena hodnota času v době zahájení definice trajektorie. Výsledná hodnota je uložena jako parametr „time“ (viz Obrázek 13) a ihned poté je opět zjištěna aktuální hodnota času,

aby mohla být po dalším dosažení vzdálenostního intervalu znovu odečtena. Pokud uživatel při definici trajektorie požaduje zastavení a setrvání v určité pozici, je nutné zastavit pohyb robota. V momentě zastavení se může příruba robota vychýlit maximálně o 49 mm, aby nedošlo k uložení nechtěných souřadnic a tím pádem také k restartu inkrementace času. Po zastavení pohybu je nutné stisknout zelené tlačítko na přírubě robota (user-Button). Hodnota metody sledující stav tlačítka je během definice trajektorie cyklicky testována a v případě hodnoty „true“ (stisknuto) jsou okamžitě zaznamenány aktuální pozice os a kartézské souřadnice robota. Po opětovném pohybu robota a překročení intervalu je vypočten čas, po který nebyl robot v pohybu. Spolu s pozicemi robota ve statickém stavu je hodnota času následně uložena jako parametr „stand“. Pro definici trajektorie se záznamem času slouží metoda `zaznamSouradnicSCasem()`. Metody pro záznam zároveň v každém cyklu vyhodnocují procesní proměnnou typu „boolean“, která sleduje, zda proces ručního navádění stále pokračuje. Pokud byl proces navádění ukončen, zavolá se metoda `ulozPoziceDoSouboru()`. Tato metoda vytvoří textový soubor ve složce pacienta, pro kterého uživatel cvik definoval a zkopíruje do něj hodnoty (pozice os, orientaci nástroje, kartézské souřadnice a popř. čas), které jsou do té doby uloženy v poli.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<RoboticsAPIData version="3">
  <jointPositions name="P1" time="200" stand="0">
    <jointtransformation a1="-0.8259" a2="-0.9988" a3="-0.5532" a4="1.7297"
      a5="0.4892" a6="0.9468" a7="-1.1264"/>
    <tcptransformation x="-160.6372" y="634.5225" z="240.8153"
      a="3.1306" b="-0.2509" c="1.5630"/>
  </jointPositions>
  <jointPositions name="P2" time="157" stand="2657">
    <jointtransformation a1="-0.7688" a2="-0.5713" a3="-0.5532" a4="1.7242"
      a5="0.4892" a6="0.5888" a7="-1.1265"/>
    <tcptransformation x="-195.9586" y="625.1771" z="437.5526"
      a="-3.1295" b="-0.5511" c="1.6152"/>
  </jointPositions>
</RoboticsAPIData>
```

Obrázek 13: Pozice os uložené v souboru

Soubor (viz Obrázek 13) s pozicemi je vytvořen tak, aby odpovídal dokumentu ve tvaru značkovacího jazyka eXtensible Markup Language (dále jen XML). Důvodem je jednodušší práce s prvky v souboru za pomoci knihoven jazyka Java, které slouží k práci s XML dokumenty. Soubor obsahuje pro ukázkou pouze dvě uložené pozice, při krátkém intervalu vzdáleností může být těchto pozic v jednom souboru i několik set. Pozice os jsou uloženy v jednotkách [rad], kartézské souřadnice v jednotkách [mm].

2.3.4 Vyhodnocení a ukončení procesu

Název souboru nese název cviku, zadaný uživatelem v aplikaci. Oba tyto parametry spolu s intervalem záznamu jsou předávány metodě `zaznamSouradnic()`. Metoda následně ukončí cyklus, ve kterém probíhal záznam souřadnic a opět cyklicky testuje procesní proměnnou sledující aktivaci ručního navádění, která má nyní hodnotu „false“. Metoda `definujTrajektorii()` vyvolá dialog, který dává uživateli na výběr ze čtyř, respektive pěti možností (viz Kapitola 1.5.1). Dvě (popřípadě tři) volby vedou k pokračování procesu

a dvě k jeho ukončení. V případě volby „Uložit naučený cvik“ a „Ukončit proces (bez uložení)“ je metoda pro záznam souřadnic pomocí procesní proměnné ukončena. Během celého procesu definice trajektorie čeká metoda zpracovávající požadavky od klienta ve třídě `ServerTCP` na výsledek procesu. Ten je definován procesní proměnnou `vysledekProcesu`, jejíž hodnotu tato metoda po celou dobu procesu cyklicky testuje. Při změně hodnoty této proměnné metoda odešle klientovi zprávu o úspěchu nebo neúspěchu celé akce, cyklus se ukončí a metoda následně opět čeká na další požadavek od klienta. Výsledek procesu cyklicky testuje také klient (aplikace v připojeném PC). V okamžiku příchodu zprávy od serveru o konci průběhu definice trajektorie se cyklus ukončí a uživatel je dialogem informován o výsledku. Pokud proces dopadl úspěšně a cvik byl uložen, zavolá se metoda pro synchronizaci, aby byl soubor s trajektorií uložen také v záložním adresáři. Aplikace je poté přepnuta do režimu, ve kterém je uživateli opět umožněno s programem pracovat.

2.3.5 Definice trajektorie v off-line režimu

Cvik je možné definovat také v off-line režimu, bez připojeného počítače. Při spuštění rehabilitační aplikace v robotu je uživateli na display `smartPADu` vypsán seznam všech evidovaných pacientů. Pro výpis slouží třída `VypisCvikyAPacienty` obsahující metody `nactiCviky()` a `nactiPacienty()`. Metoda pro výpis pacientů načte složky pacientů z adresáře do pole a přiřadí ke každému pacientovi číslo. Vybrané číslo může uživatel zadat pomocí procesních dat ve `smartPADu`. Poté je ve třídě `Rehabilitacni-Aplikace` zavolána metoda `nactiCviky()`, která přijímá jako parametr číslo zadané uživatelem. Metoda následně vypíše na display fyzioterapeutické cviky, které jsou u pacienta se zadaným číslem již evidovány. Takto může uživatel vypisovat údaje o libovolných pacientech, dokud se nerozhodne vykonat další operaci, nebo aplikaci ukončit. V případě, že se rozhodne fyzioterapeut definovat u vybraného pacienta nový cvik, zadá

za pomoci procesních dat spolu s číslem pacienta také název cviku a interval. Po potvrzení těchto hodnot je zahájen proces definice trajektorie popsany v této kapitole.

2.4 Proces provádění fyzioterapie

Pro parametrizaci a zahájení fyzioterapie slouží formulářové okno `ProvestCvikForm`. Toto okno je možné vyvolat označením pacienta a konkrétního definovaného cviku ze seznamu. Následným zvolením položky „Provést vybraný cvik“ se otevře formulářové okno (viz Obrázek 14). Po navolení požadovaných parametrů je možné zahájit cvičení pomocí tlačítka „Zahájit fyzioterapii“. Pokud uživatel zaškrtl možnost „Zobrazit síly x,y,z“, otevře se nové formulářové okno, které slouží pro zobrazení jednotlivých složek sil, kterými pacient působí na nástavec robota. Po zvolení tlačítka pro zahájení cviku je odeslán serveru požadavek, který se skládá z klíčového slova „`provestCvik`“ a parametrů, podobně jako je tomu u požadavku pro definici trajektorie. Při odeslání požadavku se zároveň vytvoří nové vlákno na pozadí a spustí metodu `zpracujHodnotySil()` sloužící pro záznam a zpracování hodnot, přijímaných od serveru během provádění cviku. Metoda `zpracujPozadavky()` uložena ve třídě `ServerTCP` přijme požadavek v podobě textového řetězce od klienta, pomocí rozdělojících znaků řetězec rozdělí a hodnoty zjištěné z požadavku přiřadí příslušným procesním proměnným.

2.4.1 Parametrizace trajektorie

Proces sestavení a provedení pohybu po trajektorii s konstantní rychlostí obsluhuje metoda `pohybKonstRychlosti()` náležící hlavní třídě `RehabilitacniAplikace`. Tato metoda je spuštěna na základě cyklického testování hodnot procesních proměnných, které jsou definované z požadavku klienta. V prvním kroku metoda nastaví tuhost os, dle hodnot zadaných uživatelem. K definici tuhosti os slouží pomocná funkce (viz Zdrojový kód 4).

```
public double[] nastavTuhost(){
    double konst[] = new double[11];
    konst[0]=2000; konst[1]=1700;konst[2]=1500;konst[3]=1300;konst[4]=1050;
    konst[5]=900;konst[6]=750;konst[7]=600;konst[8]=450; konst[9]=300;konst[10]=150;
    String tuhostOs = procesniPromenne.getTuhostOs();
    String[] zpracujPozadavek = tuhostOs.split(";");
    double tuhost[] = new double[zpracujPozadavek.length];
    for(int i=0;i<zpracujPozadavek.length;i++){
        tuhost[i] = konst[Integer.valueOf(zpracujPozadavek[i])];
    }
    return tuhost;
}
```

Zdrojový kód 4: Metoda pro definici tuhosti os

V této funkci je nejprve inicializováno pole `konst`, kde hodnota každého prvku definuje míru tuhosti. V dalším kroku jsou zjištěny požadované hodnoty tuhosti zadané uživatelem při zahájení cviku. Pole `zpracujPozadavek` má velikost sedmi prvků, přičemž každý prvek obsahuje hodnotu požadované tuhosti a index pole reprezentuje číslo osy robota. V posledním kroku se přiřadí poli `tuhost` hodnoty tuhostí uložených v poli `konst`, odpovídající indexům, které uživatel zadal. Výsledné tuhosti os jsou přiřazeny funkci `setStiffness(a1-a7)` typu `JointImpedanceControlMode`:

```
impedance.setStiffness(tuhostOs[0], tuhostOs[1], tuhostOs[2],  
tuhostOs[3], tuhostOs[4], tuhostOs[5], tuhostOs[6] - 148.0);
```

Definice prahové hodnoty je založena na podobném principu jako definice tuhosti os robota. Pokud uživatel zvolil prahovou hodnotu vyšší než nula, pomocná funkce definuje dle indexu zadaného uživatelem velikost prahové hodnoty. Ta je poté zpracována následující funkcí:

```
ForceCondition prostorovaSilaTCP = ForceCondition.createSpatial-  
ForceCondition(lbr.getFlange(), prahovaHodnota);
```

Tato funkce přijímá dva parametry: první parametr určuje, ke kterému souřadnému systému má být působící síla sledována a druhý parametr definuje velikost prahové hodnoty [N]. Pokud je velikost této hodnoty alespoň z jednoho směru (x,y nebo z) při pohybu ramene překročena, umožní funkce na tuto skutečnost programátorovi reagovat. V tomto případě je jako sledovaný souřadný systém nastavena příruba robota (flange).

2.4.2 Sestavení trajektorie s konstantní rychlostí

Dalším dílčím cílem metody `pohybKonstRychlosti()` je vytvoření bloku pohybů, které má robot vykonávat. Nejprve je třeba zpracovat soubor s pozicemi os robota, k tomu slouží metoda `vytvorPolePozic()`. Pro přístup k prvkům XML souboru s pozicemi je použito importované rozhraní DOM (Document Object Model). Pomocí funkcí poskytovaných tímto rozhraním jsou ze souboru pro každý bod vybrány hodnoty os `a1, ..., a7` a ty jsou následně uloženy do dvourozměrného pole s názvem `poziceOs`. Počet řádků pole je roven celkovému počtu záznamů (bodů trajektorie) v souboru a počet sloupců je roven počtu os robota. Název zpracovávaného souboru metoda `vytvorPolePozic()` přebírá jako parametr, který fyzioterapeut zadal při zahájení cviku. V dalším kroku procesu je vytvořena startovní pozice cviku, kterou reprezentují prvky z prvního řádku vytvořeného pole. Při pohybu ramene do zahajovací pozice je ošetřena případná kolize robota s dalším

objektem. Pomocí metody `vytvorPodminkyKolize()` jsou pro každou osu definovány hraniční momentové síly (10 Nm) a zároveň tato metoda síly působící na jednotlivé osy sleduje a vyhodnocuje. Během pohybu ramene do zahajovací pozice je funkce cyklicky testována parametrem `hasFired(podminka)`. V případě kolize jakékoli osy robota s cizím objektem podmínka zajistí okamžité zastavení pohybu ramene. Při zastavení pohybu robota v případě kolize je spuštěna metoda `pokracujVPohybu()`. Tato metoda uvede robota pomocí parametru `positionHold()` do stavu čekání. Tento stav je přerušen, pokud senzor první osy robota detekuje momentovou sílu větší nebo rovnu hodnotě 10 Nm. V praxi to znamená například slabé zatlačení rukou na nástavec robota. Robot po detekci opětovné kolize vyčká 1 s a poté pokračuje v pohybu do startovní pozice.

V dalším kroku je třeba definovat pohyb, který robot bude vykonávat. Nejprve je deklarováno pole typu `SplineMotionJP` (viz Zdrojový kód 5).

```
SplineMotionJP<>[] spMotions = new SplineMotionJP<>[poziceOs.length];
delkaCelehoCyklu = 0;
for (int i = 0; i < poziceOs.length; i++) {
    novePozice = new JointPosition(poziceOs[i][0], poziceOs[i][1], poziceOs[i][2],
        poziceOs[i][3], poziceOs[i][4], poziceOs[i][5], poziceOs[i][6]);
    spMotions[i] = ptp(novePozice);
    delkaCelehoCyklu = delkaCelehoCyklu + spoctiVzdalenost(novePozice, i);
}
```

Zdrojový kód 5: Ukázka zdrojového kódu pro definici posloupnosti pohybů

Dále je vytvořen cyklus, který postupně projde celé pole `poziceOs` typu `double`, vytvořené metodou `vytvorPolePozic()`. Pomocí těchto hodnot jsou vytvořeny pozice typu `JointPosition`, které jsou již relevantní pro parametrizaci pohybu robota. Poté je nové pozici přidělen druh pohybu, pomocí kterého robot dané pozice dosáhne. Tento parametr není pro rehabilitační aplikaci příliš podstatný, jelikož je trasa definována uživatelem pomocí ručního navádění, body jsou zaznamenávány po relativně krátkých vzdálenostech a pozice jsou zaznamenávány pomocí úhlů os robota, nikoli prostřednictvím kartézských souřadnic. Z těchto důvodů je typ pohybu, kterým se robot dostane z jednoho bodu do druhého definován převážně uživatelem. Přidáním klíčového slova `ptp` je vytvořen parametr pohybu robota a ten je následně uložen do pole. Pole `spMotions` tak tvoří posloupnost pohybů, které uživatel vytvořil ve fázi definice trajektorie. Následující příkaz vytvoří z posloupnosti pohybů typu PTP jeden komplexní pohyb:

```
SplineJP sp = new SplineJP(spMotions).setMode(impe-
dance).setJointVelocityRel(procesniPromenne.getRychlostPo-
hybu()).setJointAccelerationRel(0.00001);
```

Dané posloupnosti pohybů jsou přiděleny parametry, které zadal uživatel při zahájení cviku v aplikaci. Parametr `impedance` představuje tuhost os a k pohybu je přiřazena funkcí `setMode()`. Rychlost pohybu je konstanta typu `double`, která je přejata z požadavku klienta a následně přepočtena ve třídě `ProcesniPromenne` tak, aby tvořila hodnotu mezi 0 a 1, což je rozmezí přijímané metodou `setJointVelocityRel()`. Parametr `setJointAccelerationRel()` definuje rychlost akcelerace a decelerace ramene. Nízká hodnota zajistí pomalou a plynulou akceleraci při zahájení cyklu cviku a stejně tak plynulou deceleraci při dokončení cyklu. Samotný pohyb s přiřazenými parametry je zahájen příkazem `lbr.move(sp)`. Tento příkaz je vykonáván v cyklu. Počet cyklů je definován procesní proměnnou představující počet opakování definovanou z požadavku klienta.

Po dosažení počáteční pozice je robot uveden do stavu čekání za použití parametru `positionHold()`. Tento stav je přerušen pokud součet sil působících na efektor robota ze směrů x,y a z, překročí hranici 20 N. V okamžiku zahájení pohybu je prostřednictvím procesní proměnné spuštěna metoda `vyhodnotPohyb()` náležící třídě `ServerTCP`. Jelikož je tato třída spuštěna jako úloha na pozadí, lze vykonávat zmíněnou metodu paralelně s metodou `pohybKonstRychlosti()`, která v tento okamžik vykonává pohyb robota. Metoda pro vyhodnocení pohybu vykonává cyklicky každých 10 ms sled příkazů pro vyhodnocení vnějších působících sil a pomocí procesních proměnných komunikuje v reálném čase s druhou paralelně spuštěnou metodou vykonávající pohyb ramene. V každém cyklu jsou zjištěny funkcí `getExternalForceTorque()` složky dílčích sil (x,y,z) působící na efektor robota. Hodnoty jsou poté převedeny na řetězec a odeslány klientovy. Dále je v této metodě testováno prostřednictvím procesních proměnných fáze cyklu pohybu. Pokud byl dokončen cyklus pohybu (jedno opakování cviku), odešle se v řetězci s hodnotami také informace o dokončení opakování. V případě dokončení všech cyklů cviku metoda odešle klientovi spolu s hodnotami sil také informaci o ukončení záznamu a následně je metoda `vyhodnotPohyb()` ukončena.

2.4.3 Sestavení trajektorie s prahovou hodnotou

Metoda `provedTrasu()` je rozvětvena podmínkou určující, která část této metody bude vykonána. Podmínka vyhodnocuje informaci o volbě prahové hodnoty v aplikaci při zahájení cviku. V případě, že uživatel zvolil typ fyzioterapie bez ohledu na prahovou hodnotu odporu, vykoná se pohyb tak, jak je popsán způsobem popsáním v předchozí kapitole. V opačném případě je přidán posloupnosti pohybů parametr `breakWhen()`,

který přebírá proměnnou `prostorovaSilaTCP` s velikostí prahové hodnoty zvolené uživatelem. Během provádění pohybu je působící síla zároveň testována (viz Zdrojový kód 6) a v případě překročení prahové hodnoty je pohyb ramene plynule zastaven. Plynulost decelerace zajišťuje i v tomto případě parametr pohybu `setJointAccelerationRel()`.

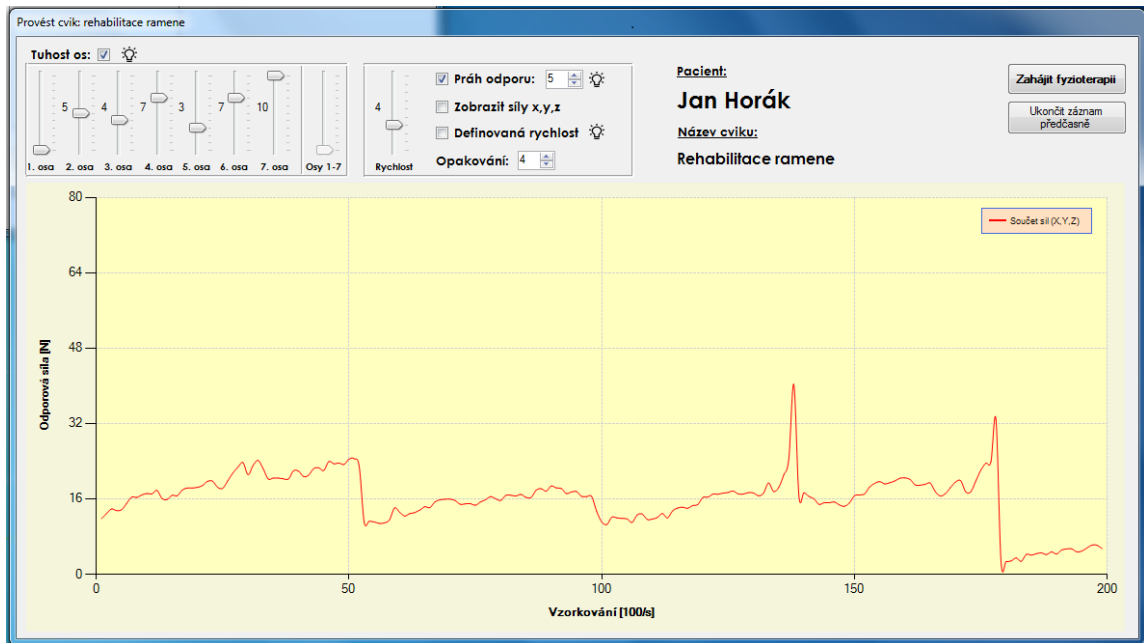
```
if (trajektorieCviku.hasFired(prostorovaSilaTCP)){
    procesniPromenne.setProcesMereni(false);
    hasFired = true;
    String pristiPozice = trajektorieCviku.getLastExecutedMotion().toString();
    Frame actualPos = lbr.getCurrentCartesianPosition(lbr.getFlange());
    procenta = spoctiProcenta(actualPos, pozice0s, pristiPozice, delkaCelehoCyklu);
    procesniPromenne.setPocetProcent(String.valueOf(procenta));
}
```

Zdrojový kód 6: Testování síly působící na efektor robota

Po zastavení pohybu je prostřednictvím procesní proměnné přerušena záznam hodnot a příkazem `getLastExecuteMotion()` je zjištěna příští pozice z posloupnosti pohybů, které by robot dosáhl v případě, že by nebyl jeho pohyb přerušen. V dalším kroku je zjištěna aktuální pozice ramene v kartézských souřadnicích a následně jsou tyto hodnoty spolu s délkou celé trajektorie předány funkci pro výpočet procent. Délka trajektorie je počítána funkcí `spoctiVzdalenost()` při vytváření sekvence pohybů (viz Zdrojový kód 5). Tato funkce provádí výpočet vzdáleností každých dvou sousedících bodů z vytvářené sekvence pohybů a ty poté sčítá. Výsledkem je tak celková délka [mm] jednoho cyklu cviku. Metoda `spoctiProcenta()` nejprve zjistí, na jakém indexu v poli sekvence pozic se nachází souřadnice příští pozice zjištěné funkcí `getLastExecuteMotion()`. Prostřednictvím zjištěných pozic je spočtena vzdálenost od počátku trajektorie až k bodu, ve kterém se rameno po přerušení pohybu nachází. Za pomoci trojčlenky jsou vypočtena procenta, která udávají poměr dosažené vzdálenosti robota ku celkové délce trajektorie. Po výpočtu procent pokračuje proces v metodě `pohybKonstRychlosti()`, kde je pořadí sekvence vykonaných pohybů převráceno a opět uloženo do pole. Výsledkem je pohyb vykonaný do okamžiku přerušení v opačném směru, tak aby rameno opět dosáhlo startovní pozice. Celý tento proces se cyklicky opakuje. Počet cyklů závisí na počtu opakování zvoleném fyzioterapeutem prostřednictvím aplikace v počítači při zahájení cviku.

Paralelně s tímto procesem je na pozadí vykonávána metoda `vyhodnotPohybSprahem()`, která je založena na stejném principu jako metoda `vyhodnotPohyb()`

popsaná výše. Hlavní rozdíl je ve způsobu odesílání dat. Metoda `vyhodnotPohybSprahem()` při zastavení pohybu ramene přeruší zasilání dat klientovi, vyčká než robot dosáhne startovní pozice a při opětovném zahájení měřeného pohybu začne znovu odesílat hodnoty klientovi, oproti druhé zmíněné metodě, která hodnoty odesílá nepřetržitě.



Obrázek 14: Rozhraní pro parametrizaci, zahájení a sledování fyzioterapie

Graf na obrázku (viz Obrázek 14) zobrazuje průběh fyzioterapie se stanovenou prahovou hodnotou. Odporová síla na ose y vykresluje součet sil ze směrů x, y a z, kterými pacient působí na madlo. Osa x znázorňuje jednotlivé vzorky záznamu. Z grafu lze vyčíst, že byla provedena čtyři opakování izolovaného cviku. První dvě opakování (vzorek 0 až ~50 a ~50 až ~100) byla provedena kompletně, aniž by robot přerušil pohyb z důvodu překročení prahové hodnoty. U dalších dvou opakování lze pozorovat špičku, která indikuje bolestivý stav procvičovaného svalu pacienta. Síla vyvíjená na madlo se náhle zvýšila a překročila prahovou hodnotu, přičemž byl pohyb robota přerušen. Negativní výsledek může být způsoben vyčerpáním pacienta z předešlých opakování tohoto fyzioterapeutického cviku.

2.4.4 Sestavení trajektorie s proměnnou rychlostí

Sestavení trajektorie a provedení pohybu s proměnnou rychlostí je možné v případě, že byl při definici trasy zaznamenáván také čas. Pro tento účel slouží metoda `pohybNekonstRychlosti()`. Tato metoda ze souboru vybere kromě pozic úhlů také hodnoty parametrů „time“ a „stand“. Na základě časů u jednotlivých záznamů a vzdáleností (0,5 cm)

jsou spočteny rychlosti pohybu mezi všemi sousedními body trajektorie. Hodnota parametru „stand“ určuje, zda v daném bodu trajektorie bude pohyb robota zastaven. Číslo větší než „0“ značí, že v daném bodu robot setrvá. Délka času setrvání je dána právě tímto číslem, které reprezentuje čas v jednotkách [ms]. Pole pozic je následně rozděleno na segmenty posloupností pozic robota, rozdělenými body trajektorie, u kterých je parametr „stand“ větší než nula. Z těchto segmentů je poté vytvořeno dvourozměrné pole typu `SplineMotionJP` (viz Zdrojový kód 7). Kromě posledního řádku obsahuje poslední prvek každého řádku tohoto pole příkaz pro PTP pohyb s parametrem „stand“ jehož hodnota je větší než nula. Všechny ostatní prvky pole disponují příkazy pro PTP pohyb obsahující parametr „stand“ s nulovou hodnotou (bez zastavení). Každý prvek pole je také parametrizován dříve vypočtenou rychlostí pohybu. Následně jsou jednotlivé posloupnosti cyklicky předávány příkazu pro pohyb robota. Po dokončení posledního pohybu z každého segmentu je pohyb robota plynule zastaven na dobu definovanou parametrem „stand“. Čas je měřen identickým způsobem jako při jeho záznamu během definice trajektorie. Ostatní mechanismy metody jsou stejné jako v případě sestavení trajektorie s konstantní rychlostí. Po vykonání pohybových příkazů ze všech segmentů dané trajektorie je celý proces opakován znovu a nebo ukončen.

```

for (int i = 0; i < spMotions.length; i++) {
    for (int j = 0; j < spMotions[i].length; j++) {
        k++;
        novePozice = new JointPosition(poziceOs[k][0], poziceOs[k][1], poziceOs[k][2],
            poziceOs[k][3], poziceOs[k][4], poziceOs[k][5], poziceOs[k][6]);
        if(j==0){
            spMotions[i][j] = ptp(novePozice).setJointVelocityRel(0.2);
        }else{
            spMotions[i][j] = ptp(novePozice).setJointVelocityRel(poziceOs[k][7]);
        }
        if(poziceOs[k][8] > 0 || k == poziceOs.length-1){ /*pokud je par. "stand" > 0,
            zapisuj na další řádek*/
            spMotions[i][j] = ptp(novePozice).setJointVelocityRel(poziceOs[k-1][7]*1.5);
            break;
        }
    }
}

```

Zdrojový kód 7: Tvorba posloupností pohybů s definovaným časem

2.5 Zpracování hodnot

Ke zpracování hodnot na straně klienta slouží funkce `zpracujHodnotySil()` zmíněná na začátku předchozí kapitoly. Odesílaná data metodami pro vyhodnocení pohybu tato funkce cyklicky přijímá a zpracovává. V prvním kroku je přijatý řetězec dle rozdělovacích znaků rozdělen a rozřazen do příslušných proměnných. Absolutní hodnoty jednotlivých složek sil jsou sečteny a výsledná síla je zobrazena v grafu. Výsledná hodnota součtu sil je ukládána vždy na poslední index pole `vyslednaSila`. Hodnoty tohoto pole jsou

s každým novým prvkem přesouvány na nižší indexy. Funkce `obnovGrafXYZ()` (viz Zdrojový kód 8) je volána při každém posuvu pole, tedy jednou za 10 ms. Metoda sekvenčně projde celé pole a jeho hodnoty vykreslí na grafu v ose y.

```
private void obnovGrafXYZ(){
    grafXYZ.Series["silaXYZ"].Points.Clear();
    for (int i = 0; i < vyslednaSila.Length - 1; i++)
    {
        grafXYZ.Series["silaXYZ"].Points.AddY(vyslednaSila[i]);
    }
}
```

Zdrojový kód 8: Zobrazení hodnot v grafu

Rozsah grafu je tak závislý na velikosti pole s hodnotami. V tomto případě má pole velikost 200 prvků. Vyšší počet prvků může mít za následek negativní vliv na plynulý průběh procesu. V případě, že uživatel zvolí možnost zobrazení jednotlivých složek sil, vyvolá se nový formulář obsahující tři grafy. V těchto grafech jsou zobrazeny složky sil x, y, z odděleně. Výpočet a zobrazení je provedeno identickým způsobem jako popisované zobrazení součtu těchto složek.

Metoda `zpracujHodnotySil()` přijaté hodnoty ukládá také do pole s názvem `hodnotySil`. Toto pole slouží pro vytvoření a uložení záznamu o provedeném cviku. V okamžiku přijetí informace o ukončení procesu jsou hodnoty z tohoto pole zpracovány. V případě, že byl cvik prováděn bez prahové hodnoty, položka procenta v tomto poli obsahuje hodnotu „100“ u vzorku, kterým daný cyklus vždy končil. Hodnoty jsou poté v každém cyklu rozpočítány po 25 procentech tak, aby byla vystižena pomocí procent aktuální fáze. Jestliže byl cvik proveden s prahovou hodnotou, výpočet není třeba provádět, jelikož je při tomto druhu fyzioterapie podstatná pouze procentuální hodnota fáze cyklu, při které byl pohyb přerušen.

Poslední fází zpracování hodnot je vytvoření záznamového listu. Metoda je rozvětvena podmínkami testujícími, zda záznam z provedeného cviku u daného uživatele již existuje či nikoliv. Pokud ano, je spuštěna metoda `pridejZaznam()`, v opačném případě se spustí metoda `vytvoritNovyZaznam()`. Hlavní rozdíl těchto metod spočívá v práci se souborem typu Excel. Druhá jmenovaná metoda vytvoří nový soubor, zatímco metoda `pridejZaznam()` pouze vytvoří nový sešit se záznamem v již existujícím souboru. Pro práci s Excelovými soubory je použita importovaná knihovna určena přímo pro tento účel. Prostřednictvím funkcí nabízených touto knihovnou je vytvořena struktura tabulek, včetně nadpisů, ohraničení, podbarvení, umístění atd. Data z pole `hodnotySil` jsou poté do příslušných buněk zkopírovány. Uložení do souboru je provedeno příkazem

workBook.SaveAs(). Jeden z parametrů této funkce je cesta ke složce, do které bude soubor uložen. Záznamy jsou ukládány do adresáře s názvem “Záznamy pacientů”. V tomto adresáři jsou postupně vytvářeny složky se jmény pacientů, dle toho, zda a kdy byl u daného pacienta vytvořen první záznamový list. Během celého procesu vytváření záznamu je aplikace v režimu čekání, ve kterém není možné provádět žádné operace. Uživatel je pouze informován o probíhajícím procesu a po dokončení je prostřednictvím dialogu informován také o kompletní cestě k vytvořenému záznamu. Poté je aplikace přepnuta opět do standardního režimu.

rehabilitace ramene						Parametry cviku		
Číslo vzorku	Cyklus [%]	Součet sil [N]	Síla x [N]	Síla y [N]	Síla z [N]	Počet op.	Rychlost	Tuhost os (1 - 7)
0		41,06150053	15,22584461	25,10951425	0,72614167	4	4	0;5;4;7;3;7;10
1		33,73616091	12,67779088	20,72293144	0,335438587			
2		11,0867905	3,163954571	7,654355708	0,268480218			
3		2,740635407	0,028988581	2,186359244	0,525287582			
4		5,285174992	0,437763272	4,09555807	0,75185365			
5		3,249060249	0,061984243	3,148460817	0,038615189			
6		4,28777707	0,354766589	3,763209635	0,169800846			
7		12,06572804	3,892746924	7,914014454	0,258966661			
8		12,64906116	3,386798078	8,972090308	0,290172772			
9		11,76339025	1,481945799	9,859161107	0,42228334			
10		12,84875132	0,507436622	12,09637081	0,244943884			
11		13,8380609	1,355384007	11,60564204	0,877034845			
12		13,50153871	0,588979585	12,57692927	0,335629859			
13		13,68821389	0,333845378	12,49256477	0,861803736			
14		15,04778254	1,366407567	11,1418198	2,53955517			
15		15,24881401	1,50063806	10,05331453	2,805476490			
						Dosaženo procent		
						Opakování	Počet %	
						1	100	
						2	100	
						3	71,94	
						4	75,77	
						Průměr:	86,92	

Obrázek 15: Ukázka záznamového listu

Záznamový list pro izolovaný cvik s prahovou hodnotou je zobrazen v tabulce (viz Obrázek 15). Každý sešit na spodní liště obsahuje záznam ze stejného cviku vytvořený v datu a čase, kterým je daný sešit pojmenován. Záznamový list pro cvičení bez prahové hodnoty se od tohoto liší tím, že neobsahuje tabulku s dosaženými procenty, jelikož je každé opakování vždy dokončeno celé. Druhá změna je ve sloupci „Cyklus [%]”, kde je každý cyklus definovaný procenty (0 až 100) tak aby bylo možné vytvořit přehledný graf, ze kterého lze vyčíst jaká fáze cviku činí pacientovi problémy. Záznam zobrazený v této tabulce byl automaticky vytvořen po ukončení rehabilitačního cvičení ramene uvedeného v kapitole 2.4.3. Ze zobrazené tabulky lze vyčíst čtrnáct úvodních vzorků, přičemž vzorky s číslem „0“ a „1“ indikují zahájení fyzioterapie vyvinutím vyšší síly na madlo robota. Ze záznamu je možné dále zjistit počet procent, kterých pacient při daném opakování cviku dosáhl, než byl pohyb robota. V případě dokončení celého cyklu bez přerušování je v dané kolonce 100 %. Procenta ze všech opakování jsou následně zprůměrována.

Poslední informací v tabulce jsou uživatelem definované parametry, se kterými byl fyzioterapeutický cvik prováděn.

3 Závěr

Hlavním cílem práce bylo vytvořit rehabilitační aplikaci s využitím senzitivního robota KUKA LBR iiwa. Výsledkem této práce je aplikace určena pro rehabilitaci horních končetin za pomoci kooperativního robota KUKA LBR iiwa. Aplikace umožňuje definici a provádění komplexních pohybů z každodenního života pacienta, ale také rehabilitaci konkrétní svalové partie horní končetiny prostřednictvím izolovaného cviku. Definici trajektorie i provádění cviků lze parametrizovat tak, aby byla fyzioterapie maximálně přizpůsobitelná danému pacientovi. Uživatelské rozhraní dále umožňuje vizualizaci a sledování průběhu rehabilitace v reálném čase. Výsledky rehabilitačního procesu jsou přehledně prezentovány a ukládány za účelem sledování pokroku pacientů.

V rámci diplomové práce byl proveden také teoretický rozbor týkající se vlastností senzitivních robotů a robotické rehabilitace. Kromě současných technik robotické fyzioterapie byly popsány také výzkumné projekty, zabývající se touto problematikou. Teoretický přehled dále zahrnuje popis skupiny průmyslových senzitivních robotů a jejich vzájemné porovnání s ohledem na využitelnost pro účely rehabilitace.

Navržená aplikace slouží jako ukázka potenciálu využití senzitivního robota pro účely rehabilitace. Tomuto záměru odpovídá také madlo vyrobené z PVC materiálu. Pevnost lepených spojů a celková tuhost konstrukce se v průběhu realizace ukázala být nedostatečná. Pro plné využití potenciálu robota a přesnější vyhodnocení sil, kterými pacient na robota působí, je nezbytná vyšší pevnost efektoru. Pro výrobu madla by tak byla vhodnější lehká slitina (např. dural). Aby byla aplikace využitelná v praxi a odpovídala klinickým požadavkům, je třeba platformu rozšířit o sofistikovanější systémy úchopu (exoskelety) a také o příslušenství pro fixaci určitých částí paže.

Jedním z možných rozšíření aplikace je implementace aktivní fyzioterapie, kde iniciátorem pohybu bude pacient. Systém bude pouze svým pohybem doplňovat fáze cviku, ve kterých je síla vyvinutá pacientem nedostatečná. Za pomoci dvou robotů LBR iiwa a síťové komunikace je možné vytvořit také systém pro tele-rehabilitaci. Aplikaci je možné naprogramovat tak, aby jeden robot sloužil fyzioterapeutovi pro vytváření pohybů pomocí ručního navádění a druhý robot vedoucí pacientovu paži pohyb v reálném čase přesně opakoval. Nezávisle na vzdálenosti by tak bylo v případě potřeby možné pacientovi poskytnout odborné služby fyzioterapeuta. Robustnější verzi robota LBR iiwa s nosností 14 kg je možné využít také pro rehabilitaci dolních končetin dětí.

V rámci diplomové práce se podařilo splnit všechny body zadání, vytvořit rehabilitační aplikaci pro robota LBR iiwa a také navrhnout uživatelské rozhraní umožňující nastavení, sledování a vizualizaci rehabilitačního procesu.

Seznam použité literatury

- [1]. KUKA Robotics [online]. Augsburg: KUKA Roboter CEE GmbH, 2016 [cit. 2016-02-26]. Dostupné z: http://www.kuka-robotics.com/czech_republic/cs/company/group/
- [2]. HENNES, Michael, Kai BOLLUE, Henry ARENBECK a Catherine DISSE-LHORST-KLUG. A proposal for patient-tailored supervision of movement performance during end-effector-based robot-assisted rehabilitation of the upper extremities. *Biomedical Engineering / Biomedizinische Technik* [online]. 2015, **60**(3), - [cit. 2018-03-31]. DOI: 10.1515/bmt-2014-0021. ISSN 1862-278X. Dostupné z: <http://www.degruyter.com/view/j/bmte.2015.60.issue-3/bmt-2014-0021/bmt-2014-0021.xml>
- [3]. KOLDITZ, Melanie, Thivaharan ALBIN, Kirsten ALBRACHT, Gert-Peter BRUGGEMANN a Dirk ABEL. Isokinematic leg extension training with an industrial robot. In: *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)* [online]. IEEE, 2016, 2016, s. 950-955 [cit. 2018-03-31]. DOI: 10.1109/BIOROB.2016.7523750. ISBN 978-1-5090-3287-7. Dostupné z: <http://ieeexplore.ieee.org/document/7523750/>
- [4]. CHONNAPARAMUTT, Winai a Witsarut SUPSI. SEFRE: Semiexoskeleton Rehabilitation System. *Applied Bionics and Biomechanics* [online]. 2016, **2016**, 1-12 [cit. 2018-04-18]. DOI: 10.1155/2016/8306765. ISSN 1176-2322. Dostupné z: <http://www.hindawi.com/journals/abb/2016/8306765/>
- [5]. Mart'i, Felip, Jo Butchart, Sarah Knight, Adam M. Scheinberg, Lisa Wise, Leon Sterling, Chris McCarthy Swinburne University of Technology, D. Hocking CSIRO, Royal Children's Hospital and Murdoch Childrens Research Institute. "Adapting a General Purpose Social Robot for Paediatric Rehabilitation through In-situ Design." (2017). Dostupné z: <https://arxiv.org/abs/1705.05142>
- [6]. WANG, Chunbao, Zhijiang LU, Yulong WANG, et al. Development of a rehabilitation robot for hand and wrist rehabilitation training. In: *2015 IEEE International Conference on Information and Automation*[online]. IEEE, 2015, 2015, s. 106-111 [cit. 2018-04-18]. DOI: 10.1109/ICInfA.2015.7279267. ISBN 978-1-4673-9104-7. Dostupné z: <http://ieeexplore.ieee.org/document/7279267>

- [7]. SAGLIA, Jody A., Nikos G. TSAGARAKIS, Jian S. DAI a Darwin G. CALDWELL. Control Strategies for Patient-Assisted Training Using the Ankle Rehabilitation Robot (ARBOT). *IEEE/ASME Transactions on Mechatronics* [online]. 2013, **18**(6), 1799-1808 [cit. 2018-04-18]. DOI: 10.1109/TMECH.2012.2214228. ISSN 1083-4435. Dostupné z: <http://ieeexplore.ieee.org/document/6296716/>
- [8]. ZHOU, Shou-Han, Justin FONG, Vincent CROCHER, Ying TAN, Denny OETOMO a Iven MAREELS. Learning control in robot-assisted rehabilitation of motor skills – a review. *Journal of Control and Decision* [online]. 2015, **3**(1), 19-43 [cit. 2018-04-24]. DOI: 10.1080/23307706.2015.1129295. ISSN 2330-7706. Dostupné z: <http://www.tandfonline.com/doi/full/10.1080/23307706.2015.1129295>
- [9]. LOUREIRO, R.C.V., M.J. JOHNSON a W.S. HARWIN. Collaborative Tele-rehabilitation: A Strategy for Increasing Engagement. In: *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006* [online]. IEEE, 2006, s. 859-864 [cit. 2018-04-24]. DOI: 10.1109/BIO ROB.2006.1639198. ISBN 1-4244-0040-6. Dostupné z: <http://ieeexplore.ieee.org/document/1639198/>
- [10]. *Operating Instructions: LBR iiwa 7 R800, LBR iiwa 14 R820*. BA LBR iiwa V5. D-86165 Augsburg, Germany, 2015.
- [11]. KEIBEL, Andreas. *Division healthcare* [online]. 12. 9. 2016, 15-17 [cit. 2018-04-18]. Dostupné z: https://www.brainportindustries.com/_asset/_public/Nieuwsberichten/BrainportIndustryKeynote.pdf
- [12]. MARHOUL, Vojtěch a Jindřich TITLBACH. *Senzitivní robotika* [online]. 23.5.2016 [cit. 2018-04-18].

A Obsah přiloženého CD

- text diplomové práce
 - diplomova_prace_2018_Vojtech_Marhoul.pdf
 - kopie_zadani_diplomova_prace_2018_Vojtech_Marhoul.pdf
- zdrojový kód programu
 - pro robota KUKA LBR iiwa (v programovacím jazyce Java)
 - aplikace pro PC (v programovacím jazyce C#)
- fotografie – fotografie platformy, snímky uživatelského rozhraní a grafů z průběhu fyzioterapie
- záznamové listy