

# Bezdotykové řízení polohy koncového efektoru manipulátoru

Diplomová práce

**Vedoucí práce:**  
Ing. Vít Ondroušek, Ph.D.

**Autor práce:**  
Bc. Josef Kolaja

Brno 2015



Rád bych poděkoval vedoucímu mé diplomové práce, panu Ing. Vítu Ondrouškovi, Ph.D., za jeho odborné rady, trpělivost a čas, který mi při zpracování této práce věnoval. Dále bych chtěl poděkovat studentům z předmětu LVR za pomoc při návrhu GUI.

Prohlašuji, že jsem práci: Bezdotykové řízení polohy koncového efektoru manipulátoru vypracoval samostatně a veškeré použité prameny a informace uvádím v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a v souladu s platnou Směrnicí o zveřejňování vysokoškolských závěrečných prací.

Jsem si vědom/a, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 6. června 2015

---

Josef Kolaja

## **Abstract**

KOLAJA, J. *Touchless position control of the end effector of robotic arm*. Diploma thesis. Brno: Mendel University in Brno, 2015.

The main aim of this diploma thesis is to design and implement the position control of the end effector of robotic arm with five degrees of freedom. The end effector reproduces the position of the operator's hand in the real time. The Kinect sensor is used for tracking position of the operator. The selected problems from the field of human body tracking, robotic arms and inverse kinematics are under discussion in the first part of the diploma thesis. The second part contains detailed description of the designed solution, which consists of the tracking of operator's palm, calculating inverse kinematic problem and development of the control software.

## **Keywords**

Katana HD300s, Kinect, touchless control, effector, C#.

## **Abstrakt**

KOLAJA, J. *Bezdotykové řízení polohy koncového efektoru manipulátoru*. Diplomová práce. Brno: Mendelova univerzita v Brně, 2015.

Cílem diplomové práce je navrhnout a implementovat řízení polohy koncového efektoru manipulátoru s pěti stupni volnosti. Koncový efektor kopíruje polohu ruky operátora v reálném čase. Pro detekci polohy operátora je použit senzor Kinect. První část diplomové práce se zabývá vysvětlením pojmů z oblasti snímání lidského těla, manipulátorů a přehledem metod řešení úlohy inverzní kinematiky. V druhé části je uveden detailní popis navrženého řešení, což zahrnuje snímání ruky operátora, výpočet úlohy inverzní kinematiky a vývoj řídicího software.

## **Klíčová slova**

Katana HD300s, Kinect, bezdotykové řízení, efektor, C#.

# Obsah

<b>1</b>	<b>Úvod a cíl práce</b>	<b>8</b>
1.1	Úvod .....	8
1.2	Cíl práce .....	8
<b>2</b>	<b>Literární řešerše</b>	<b>9</b>
2.1	Snímání lidského těla .....	9
2.1.1	Kinect v1 .....	9
2.1.2	SDK .....	10
2.1.3	Kinect v2 .....	12
2.1.4	Leap Motion .....	13
2.2	Manipulátor Katana HD300s .....	14
2.3	Metody řešení inverzní kinematiky .....	17
2.3.1	Jakobián .....	18
2.3.2	Numerické řešení transcendentních rovnic .....	19
2.3.3	Newtonova aproximační metoda inverzní transformace .....	19
2.3.4	Heuristická metoda – Cyclic Coordinate Descent (CCD) .....	20
2.4	Aktuální stav řešení problematiky .....	21
2.4.1	Bezdotykové řízení stacionárního manipulátoru .....	21
2.4.2	Využití senzoru Kinect pro řízení manipulátoru se servopohony ..	22
2.4.3	Ovládání robotické ruky s využitím senzoru Kinect .....	23
2.4.4	Použití senzoru Kinect pro řízení servo motorů manipulátoru .....	24
2.4.5	Řízení pohybu robotu pomocí senzoru Kinect .....	24
2.4.6	Komplexní interaktivní systém mezi operátorem a robotem .....	25
2.4.7	Systém pro bezdotykové řízení dvojramenného robotu .....	25
2.4.8	Zhodnocení nalezených řešení .....	27
<b>3</b>	<b>Metodika</b>	<b>28</b>
<b>4</b>	<b>Realizace</b>	<b>29</b>
4.1	Návrh celkového konceptu polohování manipulátoru .....	29

---

4.2	Stanovení výchozí pozice koncového efektoru manipulátoru.....	30
4.3	Určení požadované polohy efektoru.....	32
4.3.1	Transformace souřadných systémů.....	32
4.3.2	Výpočet polohy efektoru .....	33
4.4	Výpočet konfigurace manipulátoru .....	33
4.4.1	Algoritmus inverzní úlohy kinematiky .....	33
4.4.2	Detekce polohy ruky a analýza trajektorie .....	36
4.5	Simulační ověření proveditelnosti.....	38
4.6	Návrh a implementace aplikace .....	41
4.6.1	Implementace výpočetní části aplikace .....	42
4.6.2	Návrh a implementace grafického uživatelského rozhraní.....	43
<b>5</b>	<b>Praktické testy</b>	<b>46</b>
5.1	Testování bez připojeného manipulátoru .....	46
5.2	Testování vytvořené aplikace a návrh modifikací .....	48
5.3	Testování koncovými uživateli .....	52
<b>6</b>	<b>Shrnutí</b>	<b>53</b>
<b>7</b>	<b>Závěr a diskuze</b>	<b>54</b>
7.1	Diskuze.....	54
7.2	Závěr .....	55
<b>8</b>	<b>Seznam literatury a zdrojů</b>	<b>56</b>
<b>9</b>	<b>Seznam obrázků</b>	<b>59</b>
<b>10</b>	<b>Seznam tabulek</b>	<b>60</b>
<b>A</b>	<b>Testování bezdotykového řízení</b>	<b>62</b>
<b>B</b>	<b>Obsah přiloženého CD</b>	<b>64</b>

# 1 Úvod a cíl práce

## 1.1 Úvod

S vývojem automatizace a robotiky začala růst náročnost na ovládání pro uživatele a to zapříčinilo vývoj uživatelského rozhraní. První nástroj pro komunikaci s počítačem byla klávesnice. Ta byla dostačující až do příchodu grafického prostředí, ve kterém neumožňovala kvalitní ovládání a tak vznikla myš. Ovládání myši přineslo další možnosti, ale stále se hledala přirozenější cesta ke komunikaci s počítačem. Dalším krokem byly dotykové obrazovky, které poskytly uživateli přirozené ovládání, ovládání hlasem a hlavně ovládání gesty či pohybem. Přirozená komunikace mezi lidmi probíhá právě prostřednictvím hlasu a gest a proto je nejpřirozenější takto komunikovat i s počítačem či roboty.

V dnešní době se rozrostl počet uživatelských rozhraní, která využívají rozpoznávání gest a ovládání hlasem. Kinect, Wii či PS3 Move jsou dnes běžným příslušenstvím pro hraní her nebo ovládání televizoru. Netrvalo ale dlouho a bylo zjištěno, že by tato zařízení mohla být využita nejen na hraní her, ale například i pro řízení robotů.

Stacionární roboty jsou vázány na jedno místo, jak vyplývá z názvu a většinou mají 6 a více stupňů volnosti, které jim umožňují neomezený pohyb v pracovním prostoru. Tyto roboty mají v dnešní době široké využití snad ve všech průmyslech. V automobilovém průmyslu zajišťují bodové svařování, nátěry nebo manipulaci s bočními stěnami karosérií. Manipulace s těžkými předměty je využita jak ve skladech, tak v nápojovém průmyslu a průmyslu stavebních hmot, kde jsou využity také pro paletování. Všechny tyto činnosti jsou pro člověka příliš náročné, nebezpečné nebo zkrátka příliš monotónní a manipulátory je zvládnou vykonávat bez problému a nepřetržitě. Řízení manipulátorů probíhá přes počítač většinou, což klade velký nárok na uživatele, který musí být dostatečně kvalifikován. Tato práce se zaměřuje právě na bezdotykové řízení stacionárního manipulátoru Katana HD300s, které je intuitivnější a neklade tak velké požadavky na uživatele.

## 1.2 Cíl práce

Cílem této diplomové práce je návrh a implementace bezdotykového řízení polohy koncového efektoru stacionárního manipulátoru Katana HD300s. Pro detekci pohybu je použito zařízení Kinect, které snímá polohu ruky operátora a koncový efektor manipulátoru tuto polohu kopíruje v reálném čase. Pro přirozenější řízení efektoru manipulátoru je pohyb ruky zrcadlově kopírován.



## 2 Literární rešerše

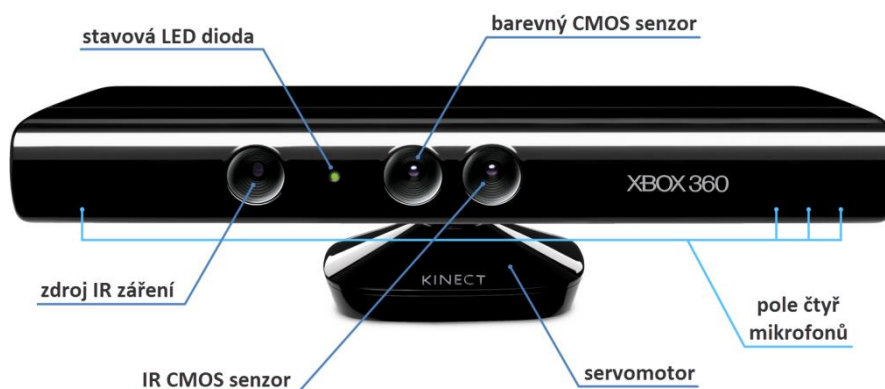
Tato část diplomové práce bude zaměřena na objasnění základních problémů spojených s řešením polohového řízení stroje. Dále především na možnost využití mapování pohybu horních končetin pro plánování trajektorie koncového efektoru. Rešeršní část také obsahuje přehled metod řešení inverzní úlohy kinematiky, na základě které bude zvolen vhodný přístup.

Nakonec kapitoly je uveden přehled existujících řešení problematiky bezdotykového řízení manipulátoru.

### 2.1 Snímání lidského těla

#### 2.1.1 Kinect v1

Zařízení Kinect bylo vyvinuto na konci roku 2010 společností Microsoft. Nejdříve sloužilo jako jen vstupní zařízení pro hrací konzole Xbox 360, ale netrvalo dlouho a společnost Computer Vision objevila, že se dá technologie Kinectu použít nejenom na hraní her. Využití Kinectu pro nekomerční účely bylo umožněno v roce 2011, kdy Microsoft vydal Software development kit zkráceně SDK (Čapek, 2013). V této práci je použito zařízení Kinect v1 určené pro vývoj.



Obr 1. Popis zařízení Kinect

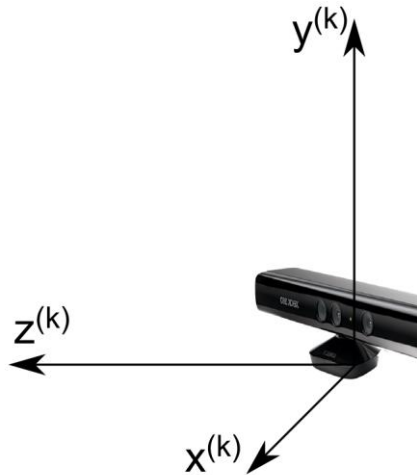
Zdroj: Vinkler, 2012

Kinect využívá RGB kameru a pole čtyř mikrofonů pro záznam obrazu a zvuku. Dále využívá zdroj IR záření, pomocí kterého určuje vzdálenost jednotlivých objektů ve scéně. Díky těmto sensorům je Kinect schopen identifikovat postavy ve scéně, detekovat pohyb jednotlivých částí těla a rozpoznávat definovaná gesta.

RGB kamera snímá tři základní barevné složky a dodává dvourozměrný obraz. Tyto data jsou použita k rozpoznávání obličeje a zobrazení reálného

prostředí v aplikaci. Kamera snímá video o frekvenci 30 Hz s maximálním rozlišením  $640 \times 480$  s hloubkou 32 bitů.

3D senzor se skládá ze zdroje IR záření a IR CMOS senzoru. IR záření je emitováno laserem jako síť bodů, kterou zachytí IR kamera. Hloubková mapa je vytvořena zářením, které se odráží od objektů a je detekováno monochromatickým CMOS senzorem. Tato mapa určuje vzdálenost objektů ve scéně. Přesnost senzoru je ve směru osy  $x^{(k)}$  a  $y^{(k)}$  v řádu milimetrů a v ose  $z^{(k)}$  cca 1 cm (Vinkler, 2012).



Obr 2. Souřadný systém zařízení Kinect

S touto přesností je Kinect je zařízení, které se dá v robotice využít pro indoor navigaci, rozpoznávání a snímání objektů, detekci vzdálenosti a hlasové ovládání.

### 2.1.2 SDK

Prvním zástupcem je SDK pro Windows od Microsoftu, které vyšlo v roce 2011. Microsoft neměl v plánu vydat Kinect pro Windows pro vývoj aplikací a proto byl velice překvapen velkým zájmem lidí o toto zařízení za účelem vývoje. V současné době Microsoft poskytuje velkou podporu v oblasti ovladačů a SDK. Microsoft SDK je volně dostupné a je možné ho spustit v prostředí Windows. Vývoj aplikací probíhá v jazycích C++ a C#. Exportovaný formát souboru závisí na vývojovém prostředí. SDK poskytuje informace o RGB, hloubce a skeletonu a dokáže detekovat až 25 kloubů na jednoho člověka. Hlavní nevýhoda Microsoft SDK spočívá v nemožnosti uchování informací o hloubce a skeletonu na disku pro offline použití. Další nevýhodou se může být vývoj pouze na Windows 7, Windows 8 (x64), Windows 8.1 (x64) nebo Windows Embedded Standard 8 (x64). Naopak silnou stránkou je velké množství možností jako například:

- odstranění pozadí
- realistické zachycení 3D obrazu pomocí Kinect fusion

- vylepšená a robustní detekce skeletonů
- detekce a vizualizace obličeje
- ukázkový kód pro interakci s HTML
- adaptivní uživatelské rozhraní
- využití gest pro ovládání aplikace

S nejnovější verzí SDK 2.0 přišlo také mnoho zlepšení. SDK je nyní schopno detekovat až šest lidí, přičemž jsou zobrazené skeletony stabilnější. Novinkou je detekce palce, konečků prstů a středu mezi rameny. Pomocí Gesture Builder mohou vývojáři vytvořit jejich vlastní gesta, které systém rozpozná a pomocí strojového učení napíše jejich kód.

Další možností je CL NUI SDK. Jedná se o částečně bezplatný balíček, který obsahuje SDK spolu s ovladači. Zkušební verzi je možné využívat 30 dní, poté jsou k dispozici pouze omezené funkce. Vývoj je určen pro platformu Windows a vývojová prostředí v jazycích C, C++ a C#. Soubory exportované z SDK jsou závislé na vývojovém prostředí. Informace o skeletonu lze získat z hloubkové mapy pomocí použití různých algoritmů. Tento proces snižuje celkovou výkonnost systému. Toto SDK nepodporuje více zařízení Kinect současně.

Open Kinect je open source a poskytuje ovladače a SDK pro práci se zařízením Kinect. Výhodou je podpora pro platformy Windows, Linux i Mac a možnost vyvíjet v jazycích Python, C, C++, C# a mnoho dalších. Exportovaný soubor opět závisí na vývojovém prostředí. SDK poskytuje RGB informaci a hloubkovou mapu bez přímé informace o skeletonu. Nicméně lze získat skeleto n z hloubkové mapy pomocí speciálních algoritmů, které ale zpomalují výkon systému a celkovou efektivnost.

Open NI je nejstarším zástupcem SDK a ovladačů pro senzor Kinect. SDK je schopno pracovat současně jen s jedním senzorem a poskytuje pouze informaci o hloubce a RGB. Vývoj je možný na platformách Windows a UNIX v jazyce C++. Skeleton lze získat pomocí algoritmů pro zpracování hloubkové mapy získané ze senzoru. Získaný skeleto n má pouze 17 kloubů lidského těla.

NUI Capture Analyze je aplikace pro záznam, analýzu a uložení informací o hloubce, RGB, skeletonu a zvuku ze senzoru Kinect, přičemž zvládá detekovat 20 kloubů lidského těla. Aplikace umožňuje export poskytnuté informace do souboru ve formátu z prostředí Matlab. Soubor .mat má dostatek informací o každém snímku, který byl nahrán. Každý snímek má vlastní hloubku, barvu, informaci o skeletonu a obličej i mnoho dalších informací. Tato aplikace nemá vlastní vývojové prostředí a slouží pouze pro zachycení dat, ale soubory .mat mohou být jednoduše použity v jakémkoli vývojovém prostředí a na libovolné platformě (Shoaib Farooq, Hassan, Naseer, 2014).

V této práci je použito první zmíněné SDK pro Windows od Microsoftu a je použita ukázková aplikace na snímání skeletonu, přičemž se využívá hlavně snímání horní části těla.

### 2.1.3 Kinect v2

Kinect pro Windows verze 2 poskytuje vývojářům o pět více snímaných kloubů, vyšší citlivost díky barevné kameře s rozlišením 1920 x 1080. Tyto možnosti mohou být využity pro vývoj interaktivních aplikací ovládaných hlasem nebo gesty pro Windows a Windows Store. Kinect v2 je určený k použití s SDK 2.0. pro Windows. Toto SDK umožňuje vývojářům vytvářet aplikace v jazycích C++, C#, VB.Net, Cx nebo JavaScript (Microsoft, 2015b).



Obr 3. Kinect v2

Zdroj: Microsoft, 2015b

Kinect v2 disponuje barevnou kamerou, která poskytuje video v rozlišení 1080p. Sensor nyní zvládá sledovat až šest lidí a 25 kloubů těla na jednoho člověka. SDK poskytuje přesnější informace o pozicích kloubů a těla, které jsou díky tomu anatomicky lépe zobrazeny a sensor má větší zorný úhel. Srovnání parametrů lze vidět v Tab. 1.

Tab. 1 Srovnání Kinect v1 a Kinect v2

	Kinect v1	Kinect v2
Rozlišení barevné kamery	640 × 480 @30 fps	1920 × 1080 @30 fps
Rozlišení hloubkové mapy	320 × 240	512 × 424
Maximální vzdálenost od senzoru	~4,5 m	~4,5 m
Minimální vzdálenost od senzoru	40 cm v režimu na blízko	50 cm
Horizontální zorný úhel	57°	70°
Vertikální zorný úhel	43°	60°
Motor pro náklon	Ano	Ne
Počet kloubů na osobu	20	26
Počet snímaných osob	2	6

Zdroj: Zugara, 2015

Kinect v2 dokáže na hloubkových mapách zobrazit díky vyššímu rozlišení menší objekty, celkově jsou objekty zřetelnější a jsou zobrazeny ve třech dimen-

zích. Na Obr 4 lze vidět zlepšení nového senzoru oproti původní verzi (Microsoft, 2015a).



Obr 4. Srovnání hloubkových map Kinectu verze 1 (nalevo) a verze 2 (napravo)

Zdroj: Microsoft, 2015a

#### 2.1.4 Leap Motion

Leap Motion je malé USB zařízení, které je navrženo k umístění na fyzické ploše směrem vzhůru. Pomocí dvou monochromatických IR kamer a tří infračervených LED diod přístroj sleduje oblast ve tvaru polokoule do vzdálenosti jednoho metru. LED diody generují 3D vzor teček IR světla, který je zachycen kamerami s frekvencí 300 snímků za sekundu. Data jsou následně odeslána prostřednictvím USB do počítače, kde jsou analyzována pomocí Software Development kitu od Leap Motion.

Leap Motion je na rozdíl od senzoru Kinect určen pouze pro snímání rukou. Senzor snímá do vzdálenosti přibližně jeden metr, má zorný úhel  $150^\circ$  a poskytuje přesnost v řádech setin milimetrů. Leap Motion se dá použít pro navigaci na webových stránkách a mapách s využitím gest, kreslení a práci s 3D vizualizací (Leap Motion, 2015).



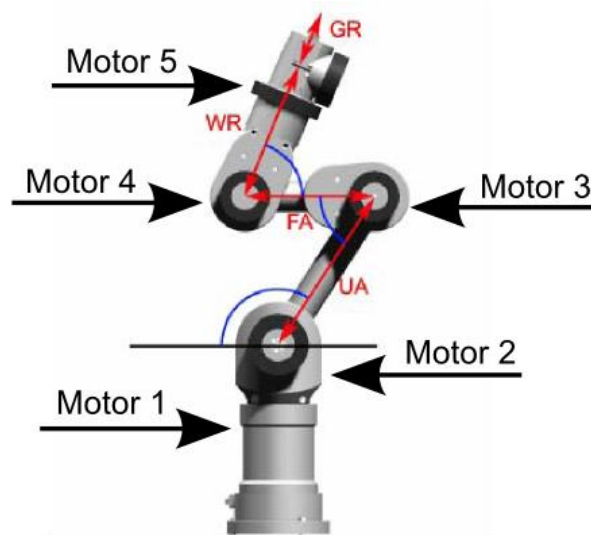
Obr 5. 3D model rukou

Zdroj: Leap Motion, 2015

## 2.2 Manipulátor Katana HD300s

Katana jsou roboty od švýcarské firmy Neuronics AG. Roboty mají speciální konstrukci a bezpečnostní opatření, díky kterým je možná přímá spolupráce vedle lidské obsluhy bez jakéhokoli omezení pracoviště. Roboty jsou malé, mobilní a snadno přenositelné. Z těchto důvodů jsou roboty Katana vhodné pro školní projekty. V této práci je použit model Katana 6M180 (Neuronics, 2011).

Manipulátor je možné naprogramovat pomocí souboru knihoven Katana Native Interface (KNI), které umožňují ovládání robotu pomocí programovacích jazyků C++ nebo Python. Standardně dodávaný software Katana4D od společnosti Neuronics AG umožňuje správu, ovládání robotu, čtení hodnot ze senzorů a vytváření programů pro řízení. Dále je možné využít prostředí Control Web, které pracuje s technologií ActiveX, pomocí které je zajištěno rozhraní mezi manipulátorem a prostředím Control Web. (Vytečka, 2013).



Obr 6. Katana 6M180

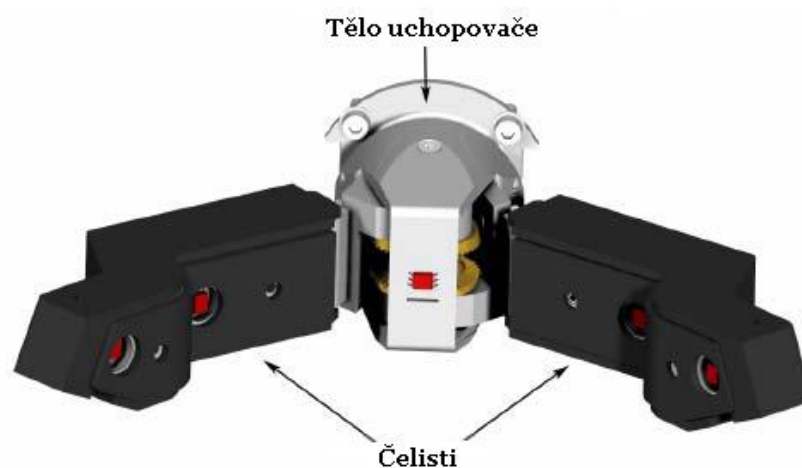
Zdroj: Neuronics, 2006

Tab. 2 Technické parametry manipulátoru

DOF	5
Max. výška [mm]	854
Úhly vyosení [ ° ]	M1: 0
	M2: 124,25
	M3: 52,7
	M4: 63,5
	M5: 8,5
Operační rozsah [ ° ]	M1: 345,7
	M2: 140
	M3: 241,5
	M4: 232
	M5: 332,2
	M6: 140 (čelisti)
Délka ramene [mm]	UA: 190
	FA: 139
	WR: 185
	GR: 130 (s uchopovačem)

Zdroj: Neuronics, 2006

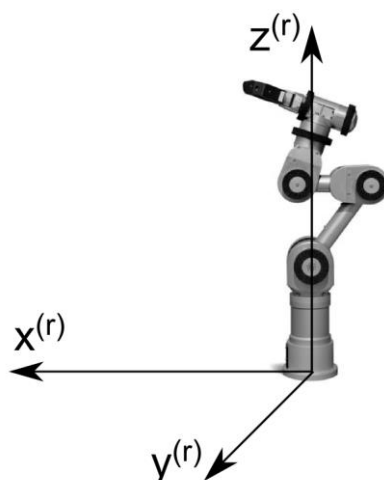
Uchopovače manipulátoru Katana HD300s jsou vybaveny infračervenými snímači a snímači síly.



Obr 7. Univerzální uchopovač manipulátorů Katana

Zdroj: Neuronics, 2006

Manipulátor Katana HD300s pracuje v souřadném systému, který je na Obr 8.



Obr 8. Souřadný systém manipulátoru

Pro snímání rotační polohy se u mobilních a stacionárních robotů využívá hlavně optických enkodérů. Enkodéry jsou miniaturní uzavřené válečky s optickými závorami uvnitř. Při rotaci je světelný paprsek v závoře přerušován diskem s miniaturními otvory. Pro výrobu disku se používá pochromované sklo, kov nebo foto-plast.

Existují dva základní typy enkodérů. Inkrementální neboli relativní, které poskytují pouze relativní informaci o pozici a měří pouze rotační rychlost. A absolutní, které měří absolutní natočení v každém okamžiku.

Relativní enkodéry jsou oproti absolutním levnější, dostupnější a tudíž používanější. Důvod je jednodušší zpracování a vyhodnocení signálu z enkodéru. V praxi se tedy pro měření absolutní rotační polohy využívá relativních enkodérů. Problém relativní pozice je řešen vytvořením vztažného bodu mechanickým kontaktem nebo podmínkou inicializace s pomocí indexového kanálu. Je náročné vyrobít dostatečně přesné mechanické řešení, a proto je pro definování vztažného bodu více využíváno místo prvního index-pulsu po sepnutí spínače. (Šolc a Žalud, 2002).

Motory stacionárního manipulátoru Katana HD300s jsou řízeny relativními enkodéry. Manipulátor má pět motorů, které určují jeho polohu a šestý motor pro otevírání a zavírání uchopovače. V Tab. 3 jsou rozsahy jednotlivých motorů v hodnotách enkodérů a stupních a hodnota ECP (Enkodér/puls), která je potřebná k přepočtu z enkodéru (odměřování) na úhel motoru a zpět. Manipulátor Katana HD300s pracuje s úhly v radiánech, ale pro lepší srozumitelnost je maximální rozsah uveden ve stupních.



Tab. 3 Rozsah jednotlivých motorů

Číslo motoru	Maximální rozsah	Minimální rozsah	EPC	Maximální rozsah [°]
1	31000	-20200	51200	345
2	-31000	63976	94976	140
3	-31000	50408	81408	241
4	31000	-20200	51200	232
5	31000	-20200	51200	332
6	26000	6000	20000	140

### 2.3 Metody řešení inverzní kinematiky

Pro otevřený kinematický řetězec je zpětná neboli inverzní úloha kinematiky pro polohy definována vektorem kartézských souřadnic  $x \in \mathbb{R}^m$  (poloha a orientaci souřadnicového systému  $x_n y_n z_n$ ) koncového efektoru manipulátoru a rozměry mechanismu. Výstupem jsou právě kloubové souřadnice  $q = [q_1, \dots, q_n]^T \in \mathbb{R}^n$  (Grepl, 2007).

Pokud je přímá úloha definována následovně

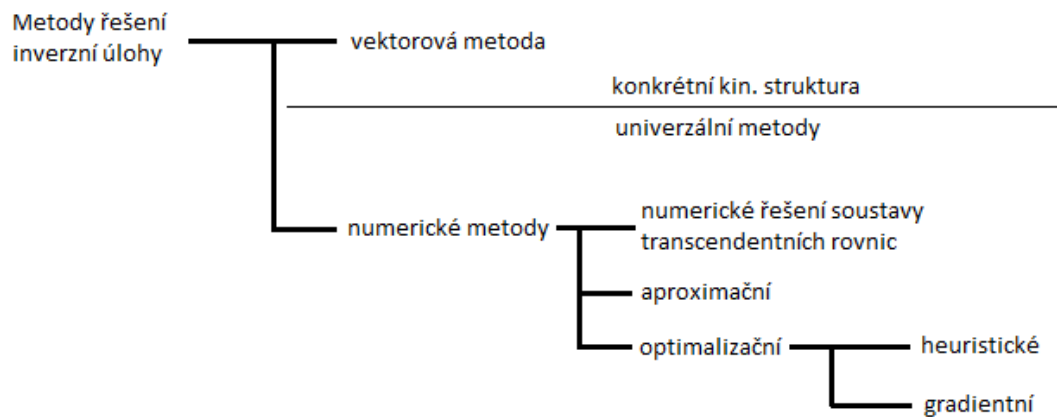
$$x = f(q) \quad (1)$$

potom má inverzní úloha tvar

$$q = f^{-1}(x) \quad (2)$$

Pokud nelze řešení inverzní úlohy vyjádřit analyticky, je zapotřebí použít iteračních numerických metod s využitím Jakobiánu. Pokud je požadována poloha a orientace mimo pracovní prostor manipulátoru, který je omezený, řešení neexistuje. Dalším problémem je víceznačnost řešení, která nastává ve chvíli, kdy je možné dosáhnout jednoho bodu dvěma či více způsoby. Redundantní manipulátory mají ve 3D sedm a více stupňů volnosti a teoreticky můžou dosáhnout definované polohy a orientace nekonečně mnoha způsoby (Grepl, 2007).

Základní rozdělení metod řešení inverzní kinematiky je na obecné numerické metody, které lze bezproblémově použít u manipulátorů a analytické řešení, například vektorové metody, používané pro jednodušší konkrétní problémy. Rozdělení metod řešení inverzní kinematiky lze vidět na Obr 9, přičemž budou popsány pouze numerické metody.



Obr 9. Základní rozdělení metod řešení inverzní úlohy kinematiky

Zdroj: Mostýn, Krys, 2012

### 2.3.1 Jakobián

Obecný vztah pro polohu je možné formálně přepsat následovně (Grepl, 2007):

$$\mathbf{x} = f(\mathbf{q}) = \begin{bmatrix} f_1(\mathbf{q}) \\ f_2(\mathbf{q}) \\ \dots \\ f_m(\mathbf{q}) \end{bmatrix} \quad (3)$$

Rychlosti získáme derivací podle času

$$\frac{d}{dt} \mathbf{x} = \frac{\partial f}{\partial q_1} \frac{dq_1}{dt} + \frac{\partial f}{\partial q_2} \frac{dq_2}{dt} + \dots + \frac{\partial f}{\partial q_n} \frac{dq_n}{dt} \quad (4)$$

a uvedenou derivaci lze přepsat

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}} \quad (5)$$

Tímto vztahem lze na základě znalosti rychlostí kloubových souřadnic určit rychlost kartézských souřadnic. Matice  $\mathbf{J} \in \mathbb{R}^{m \times n}$

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} & \dots & \frac{\partial f_1}{\partial q_n} \\ \frac{\partial f_2}{\partial q_1} & \frac{\partial f_2}{\partial q_2} & \dots & \frac{\partial f_2}{\partial q_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial q_1} & \frac{\partial f_m}{\partial q_2} & \dots & \frac{\partial f_m}{\partial q_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{x}}{\partial q_1} & \frac{\partial \mathbf{x}}{\partial q_2} & \dots & \frac{\partial \mathbf{x}}{\partial q_n} \end{bmatrix} \quad (6)$$

se nazývá *Jakobián* nebo *Jakobiho matice* a má zásadní význam při řešení inverzní úlohy a dalších problémů (Grepl, 2007).

V případě složitějších soustav není možné nalézt analytické řešení, a proto se často využívá inverze Jakobiánu. Totální diferenciál v rovnici (4) se převede na diference a zpětná úloha kinematiky se dále realizuje iteračním procesem.

Transpozice Jakobiánu je jistou obměnou předešlého algoritmu, která je založena na výpočtu transpozice Jakobiánu namísto obtížné inverze.

Nejjednodušší metoda numerického výpočtu Jakobiánu vychází z jednoduché myšlenky nahrazení parciální derivace diferencí (Grepl, 2007).

$$\frac{\partial f_i}{\partial q_j} \doteq \frac{f_i(q_j + \Delta q_j) - f_i(q_j)}{\Delta q_j} \quad (7)$$

### 2.3.2 Numerické řešení transcendentních rovnic

Soustavu rovnic získáme přepočtem souřadnic koncového efektoru manipulátoru na zobecněné souřadnice jednotlivých pohybových jednotek. Tyto rovnice jsou z důvodu použití rotačních vazeb transcendentní. Výpočet těchto rovnic může být proveden například v programovém prostředí Mathcad nebo Matlab. (Mostýn, Krys, 2012).

### 2.3.3 Newtonova aproximační metoda inverzní transformace

Newtonova metoda inverzní transformace aproximuje průběh funkce polohy a orientace koncového bodu kinematické struktury. To probíhá na základě výpočtu diferenciálu funkce polohy při známém výchozím bodě. V tomto případě je použito Jakobiho matice soustavy na místě diferenciálu, a odvození matice pro danou kinematickou strukturu je provedeno intuitivně přímo z definice Jakobiho matice pro soustavu funkcí více proměnných (Mostýn, Krys, 2012).

Je dán předpoklad, že v  $n$ -rozměrném prostoru je definováno  $m$  funkcí soustavou rovnic:

$$\begin{aligned} y_1 &= f_1(x_1, x_2, \dots, x_n) \\ y_2 &= f_2(x_1, x_2, \dots, x_n) \\ &\vdots \\ y_m &= f_m(x_1, x_2, \dots, x_n) \end{aligned} \quad (8)$$

Každý vektor  $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]^T$  z  $n$ -rozměrného prostoru má přiřazen vektor  $\mathbf{y} = [y_1 \ y_2 \ \dots \ y_m]^T$   $m$ -rozměrného prostoru podle zobrazení, které vyplývá ze soustavy rovnic. Za předpokladu, že jsou funkce  $f_1, f_2, \dots, f_m$  diferencovatelné a mají parciální derivaci 1. řádu, lze napsat totální diferenciály

$$\begin{aligned}
 dy_1 &= \frac{\partial f_1}{\partial x_1} dx_1 + \frac{\partial f_1}{\partial x_2} dx_2 + \dots + \frac{\partial f_1}{\partial x_n} dx_n \\
 dy_2 &= \frac{\partial f_2}{\partial x_1} dx_1 + \frac{\partial f_2}{\partial x_2} dx_2 + \dots + \frac{\partial f_2}{\partial x_n} dx_n \\
 &\vdots \\
 dy_m &= \frac{\partial f_m}{\partial x_1} dx_1 + \frac{\partial f_m}{\partial x_2} dx_2 + \dots + \frac{\partial f_m}{\partial x_n} dx_n
 \end{aligned} \tag{9}$$

Tuto soustavu je možno zapsat pomocí maticového zápisu

$$\begin{bmatrix} dy_1 \\ dy_2 \\ \vdots \\ dy_m \end{bmatrix} = \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ u_{21} & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \dots & \vdots \\ u_{m1} & u_{m2} & \dots & u_{mn} \end{bmatrix} \cdot \begin{bmatrix} dx_1 \\ dx_2 \\ \vdots \\ dx_n \end{bmatrix} \text{ kde } u_{ij} = \frac{\partial f_i}{\partial x_j} \tag{10}$$

nebo symbolicky

$$d\mathbf{y} = \mathbf{J} \cdot d\mathbf{x} \tag{11}$$

kde matice  $\mathbf{J}$  je Jakobiho matice soustavy (Mostýn, Krys, 2012).

#### 2.3.4 Heuristická metoda – Cyclic Coordinate Descent (CCD)

Heuristické metody fungují na principu optimalizační techniky přímého heuristického vyhledávání řešení. Vyhledávání je založeno na minimalizaci funkce chyby polohování postupným fiktivním pohybem jednotlivých článků. Výhodou těchto metod je, že poměrně rychle naleznou bod blízký konečnému řešení a dají se použít i pro šest a více stupňů volnosti (Mostýn, Krys, 2012).

Každý cyklus hledání řešení metody CCD má  $n$  kroků, kde  $n$  je počet stupňů volnosti. V prvním kroku se fiktivně pohybuje s posledním článkem, dokud není dosaženo minima chyby polohování. V průběhu jsou ostatní články zablokovány. Poté je zablokován poslední článek a uvolněn předchozí a tímto způsobem se pokračuje až k prvnímu článku. Cyklus iterací se opakuje od posledního článku, dokud není chyba polohování menší než požadovaná hodnota (Mostýn, Krys, 2012).

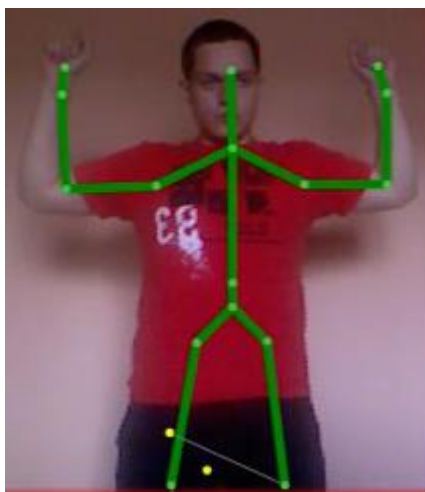
Podmínky pro použití metody CCD:

- Otevřená kinematická struktura
- Tuhé články
- Pohybové jednotky s jedním stupněm volnosti
- Osy  $z$  ve směru otáčení nebo posuvu článků

## 2.4 Aktuální stav řešení problematiky

### 2.4.1 Bezdotykové řízení stacionárního manipulátoru

Cílem práce bylo navrhnout a implementovat bezdotykové řízení manipulátoru. Pro snímání lidského těla bylo zvoleno zařízení Kinect a k řízení byl vybrán stacionární manipulátor Katana HD300s. V práci byly diskutovány dva způsoby řízení, popsané jako absolutní a relativní řízení. Princip obou řízení byl v ovládní jednotlivých motorů pomocí namapovaných gest. Absolutní řízení bylo zavřeno z důvodu nemožnosti namapovat jednotlivé pohony stacionárního manipulátoru na gesta tak, aby se neovlivňovaly navzájem. Proto bylo zvoleno relativní řízení, které vyžaduje kalibraci uživatele, ale umožňuje přerušování snímání, uvedení ruky operátora do výchozí polohy a opětovné pokračování v řízení. Startovací gesto lze vidět na Obr 10. Toto gesto bylo také využito pro kalibraci uživatele (Slaný, 2014).

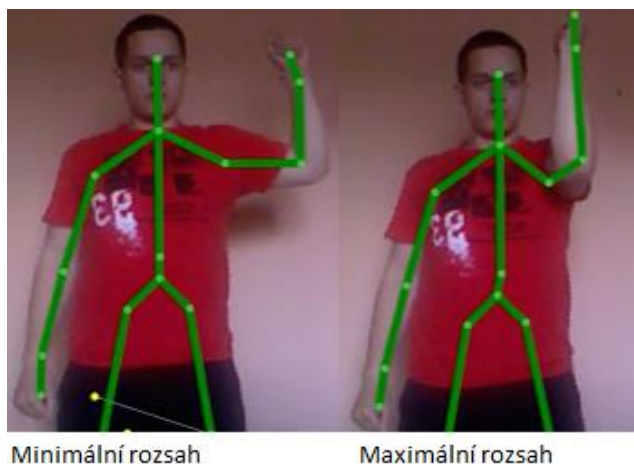


Obr 10. Startovací gesto

Zdroj: Slaný, 2014

Pro převod dat ze zařízení Kinect na vstupní data manipulátoru byly porovnávány dva zachycené snímky kostry. Jeden snímek byl uložen do paměti a následně použit pro porovnání s dalším snímkem. Na základě tohoto porovnání bylo zjištěno, který kloub má největší změnu ve své poloze a tedy kterým motorem manipulátoru se má pohnout.

Pro ovládní jednotlivých motorů byla zvolena různá gesta, kterým byl přiřazen minimální a maximální rozsah. Ukázka gesta pro ovládní motoru číslo jedna je na Obr 11 (Slaný, 2014).



Obr 11. Ovládání motoru číslo 1

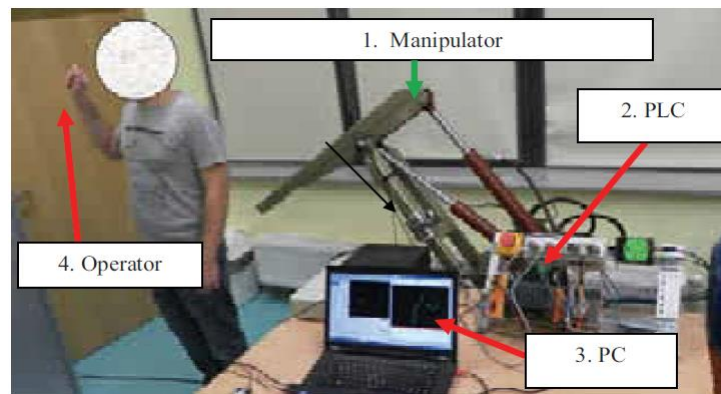
Zdroj: Slaný, 2014

#### 2.4.2 Využití senzoru Kinect pro řízení manipulátoru se servopohony

Práce popisuje ovládání dvouosého elektrohydraulického manipulátoru pomocí pohybu lidské ruky. K detekci operátora byl použit senzor Kinect a k určení polohy manipulátoru byla použita inverzní kinematika. Řízení záviselo na pohybu ramene, loktu a ruky operátora.

Hydraulické písty manipulátoru mohou dosáhnout síly  $\sim 10$  kN na koncovém efektoru. Pro zjednodušení výpočtu byl souřadný systém senzoru převeden do souřadného systému v OPEN CV. Pro výpočet byla mimo jiné důležitá délka robotického ramene, souřadnice ruky operátora, které kopíroval koncový efektor manipulátoru, souřadnice ramene operátora, které odpovídaly základně manipulátoru a souřadnice loktu operátora.

V práci bylo použito PLC od firmy B&R připojené přes TCP/IP s PVI rozhraním (Process Visualization Interface). Pro snímání rotační polohy motorů na manipulátoru byly použity inkrementální enkodéry. Protože měl manipulátor pouze dva stupně volnosti, byla inverzní kinematiky polohy počítána geometricky a nebyla zapotřebí žádná numerická či jiná běžně používaná metoda (Gošliński, 2013).

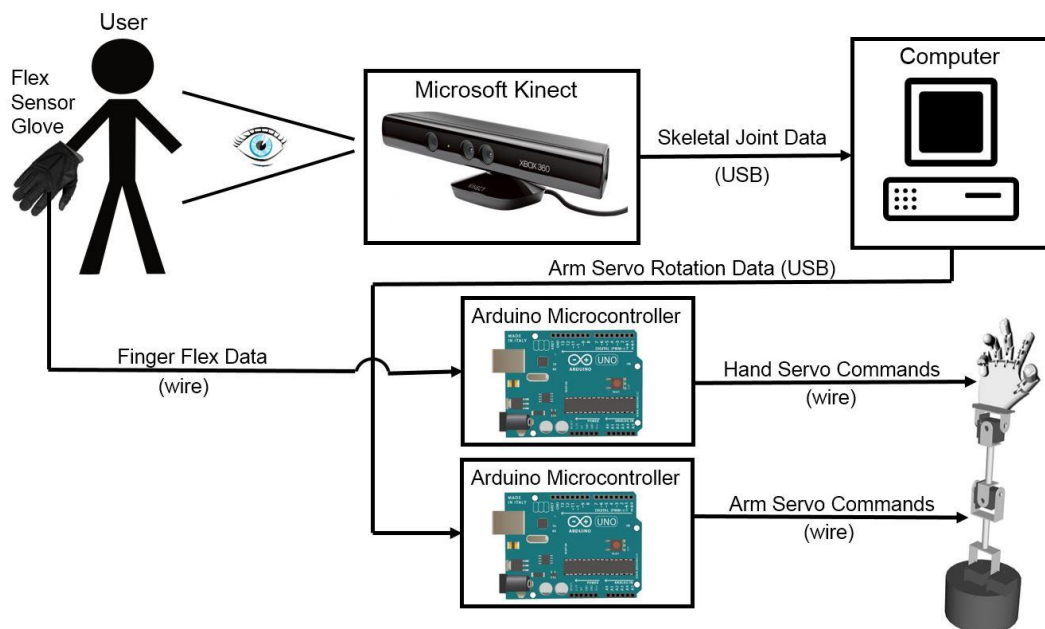


Obr 12. Testování aplikace

Zdroj: Gošliński, 2013

### 2.4.3 Ovládání robotické ruky s využitím senzoru Kinect

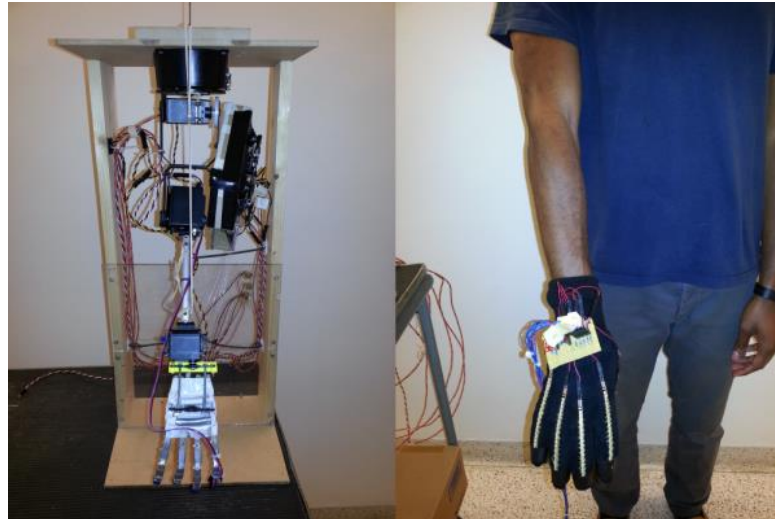
V tomto projektu byla sestrojena robotická ruka, kterou lze ovládat pomocí pohybu ruky operátora. Ovládat lze nejen pohyb ruky, ale i prstů, což zajišťuje speciální rukavice se senzory detekující ohnutí. Hardware rukavice je natolik jednoduchý, že je zpracován jedním mikro kontrolérem Arduino, který se stará o pohyb prstů. Informace ze senzoru Kinect jsou zpracovány v programu napsaném v C++ a poté jsou odeslány do druhého mikro kontroléru Arduino, které zajišťuje pohyb ruky (Reid, 2013).



Obr 13. Schéma řízení

Zdroj: Reid, 2013

Pro sestavení robotické ruky byl použit manipulátor AL5D of firmy Lynx-motion. Tento manipulátor se čtyřmi stupni volnosti a servomotory byl použit jako paže. Na rukavici pro ovládání pohybu prstů byl umístěn akcelerometr, pomocí kterého byl určen náklon zápěstí (Reid, 2013).



Obr 14. Robotická ruka a ruka operátora

Zdroj: Reid, 2013

#### **2.4.4 Použití senzoru Kinect pro řízení servo motorů manipulátoru**

Článek se zaměřuje na návrh řídicího systému pro manipulátor Arexx Robot Arm Pro se šesti stupni volnosti. Řízení je založeno na gestech ruky a pro snímání operátora je použit senzor Kinect. Práce se zaměřuje hlavně na zhodnocení přesnosti a použitelnosti zmíněného senzoru pro polohování jednoho servo motoru. V práci jsou zmiňovány dva řízené pohyby, kruhový a horizontální, přičemž kruhový pohyb splňuje požadovanou přesnost řízení. Průměrná dosažená odchylka naměřených hodnot při horizontálním pohybu je 22 mm, což autoři uvádí jako nedostatečné pro řízení daného servo motoru. V práci není diskutováno chování při řízení celého kinematického řetězce (Parzych 2014).

#### **2.4.5 Řízení pohybu robotu pomocí senzoru Kinect**

Tato práce má podobný přístup jako předchozí článek, kdy je pohyb lidské paže snímán, zpracován a následně vykonán robotickým ramenem. Ve skutečnosti jsou zpracovány pouze dva úhly z nasnímané polohy paže operátora. Z toho vyplývá, že robotické rameno se třemi stupni volnosti je řízeno pomocí těchto dvou vypočítaných hodnot. (Hussein, 2014).



#### 2.4.6 Komplexní interaktivní systém mezi operátorem a robotem

Zcela odlišný přístup je řešen v práci, která se zabývá systémem pro řízení robotu pomocí inerciálního senzorkého obleku Xsens MVN Awinda. Oblek se skládá ze 17 inerciálních gyroskopických senzorů připevněných na tělo operátora. Tyto senzory poskytují informace o poloze, rychlosti i zrychlení v reálném čase. Pro řízení polohy manipulátoru s pěti stupni volnosti jsou využity informace ze senzorů umístěných na pravé ruce a k pohybu robotu jsou použity informace ze senzorů umístěných na těle operátora. Pohyb levé paže určuje příkazy k ovládání systému jako například zapnutí a vypnutí manipulátoru nebo celého robotu. V práci byl použit mobilní robot KUKA youBot s pěti stupni volnosti (Khassanov, 2014).

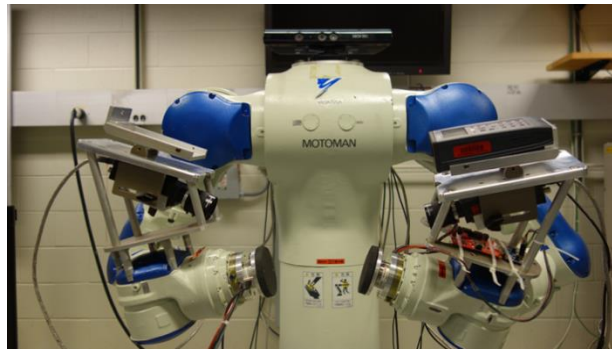


Obr 15. Komplexní interaktivní systém mezi operátorem a robotem

Zdroj: Khassanov, 2014

#### 2.4.7 Systém pro bezdotykové řízení dvojramenného robotu

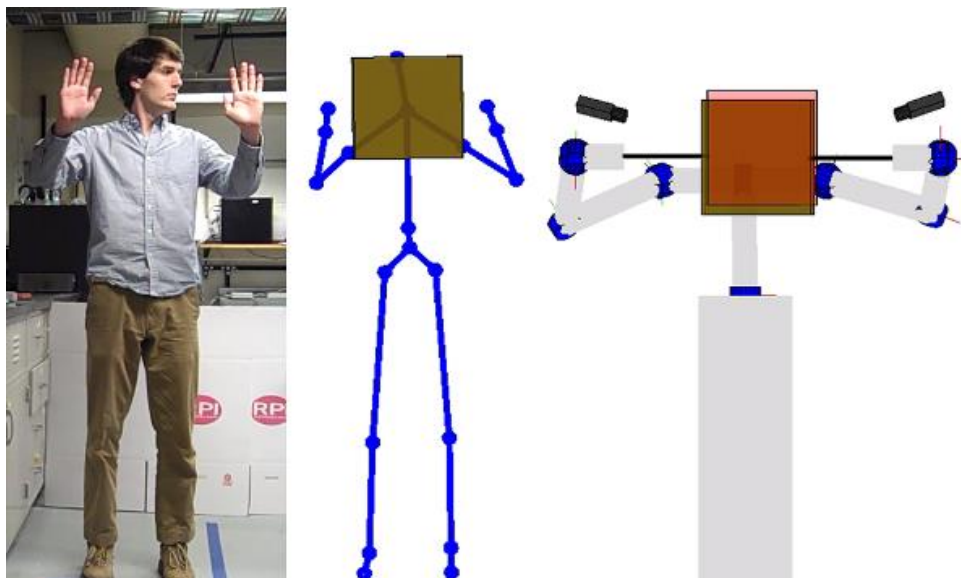
Tento projekt představuje systém řízení dvojramenného průmyslového robotu Yaskawa Motoman SDA10, který se skládá z rotačního trupu a dvou robotických ramen se sedmi stupni volnosti. Ramena jsou vybavena kamerami a jsou schopna nezávislého pohybu a uchopení předmětů s kontrolovanou silou úchopu. Operátor ovládá robotická ramena přes sensor Kinect pomocí pohybu rukou. V případě uchopení předmětu udržuje autonomní regulátor stabilní úchop. Sensor Kinect byl využit hlavně kvůli přirozenému ovládání, které není limitováno mechanickým vstupem.



Obr 16. Yaskawa Motoman SDA10

Zdroj: Kruse, 2013

Systém pro bezdotykové řízení je naimplementován v prostředí MATLAB z důvodu snadného vývoje, prototypování a analýzy dat. Systém zahrnuje sedm procesů běžících na třech počítačích (dva s OS Windows 7 a jeden s OS Windows XP), které spolu komunikují pomocí TCP/IP protokolu. Komunikace mezi komponentami je založena na distribuovaném softwaru pro řízení a komunikaci Robot Raconteur. Průmyslový robot Motoman SDA10 je řízen pomocí vestavěného rozhraní HSC (High Speed Controller), které umožňuje čtení a inkrementální nastavení úhlů na kloubech robotu. Použité kamery, které jsou upevněné na ramenech, jsou Sony XCD-X710CR s rozlišením  $1034 \times 779$  a 30 fps. Na konci každého ramene je víceosý silový a momentový senzor ATI Mini 45, který současně měří síly  $F_x$ ,  $F_y$ ,  $F_z$  a momenty  $T_x$ ,  $T_y$  a  $T_z$  (Kruse, 2013).



Obr 17. Operátor, grafická reprezentace snímaného skeletonu a simulace robotu SDA10

Zdroj: Kruse, 2013

### 2.4.8 Zhodnocení nalezených řešení

První řešení (Slaný, 2014) bylo testováno na třech operátorech a z měření bylo zjištěno, že nezkušený operátor dosahuje horších výsledků kvůli složitosti ovládání. Další nevýhodou se může jevit ovládání motorů samostatně, což může zpomalit dosažení požadované polohy koncového efektoru.

Další řešení jsou více specifická. Navržené řízení dvouosého manipulátoru (Gośliński, 2013) se podobá řešení v této práci. Řízení probíhá na základě snímání ruky a polohové řízení manipulátoru je řešeno inverzní kinematikou. Avšak manipulátor má pouze dva stupně volnosti a proto je použito analytické řešení inverzní kinematiky, které je výrazně jednodušší.

Ovládání robotické ruky (Reid, 2013) využívá kromě senzoru Kinect i senzory na rukavici, díky kterým je umožněno ovládání prstů. Robotická ruka byla sestavena na míru, a proto může poskytovat větší pohyblivost, než manipulátor, který je použit v této práci.

Další dvě řešení (Parzych, 2014; Hussein, 2014) využívají také senzor Kinect, ale nezaměřují se na řízení celého kinematického řetězce manipulátoru. Z toho vyplývá, že nemohou být použity při řízení manipulátoru s pěti a více stupni volnosti a implementace vhodné metody inverzní kinematiky je nezbytná.

V předposledním řešení (Khassanov, 2014) je použita inverzní úloha kinematiky pro výpočet polohy manipulátoru s pěti stupni volnosti na základě nasnímaných hodnot z obleku Xsens MVN. V práci bohužel není uvedena použitá metoda.

Bezdotykové řízení dvojramenného robotu (Kruse, 2013) je velice komplexní. Řízení dvou robotických ramen se sedmi stupni volnosti a automaticky regulovanou silou úchopu by se mohlo považovat za „state of the art“.

### 3 Metodika

V rešeršní části diplomové práce je objasněn základní problém spojený s řešením polohového řízení stroje. Dále je rešeršní část zaměřena na možnost využití mapování pohybu horních končetin pro plánování trajektorie koncového efektoru. Práce se nezaměřuje na kopírování pohybu jednotlivých kloubů ruky, ale spíše na sledování polohy dlaně. Polohové řízení pohonů manipulátoru je určeno na základě řešení inverzní úlohy kinematiky pomocí vhodné metody, přičemž na vstupu této úlohy je právě poloha dlaně operátora. Z tohoto důvodu se rešeršní část zabývá problematikou snímání polohy lidského těla, zvláště horních končetin, pomocí zařízení Kinect a také řízením stacionárního manipulátoru Katana HD300s. Rešeršní část také obsahuje přehled metod řešení inverzní úlohy kinematiky.

V praktické části budou provedeny následující kroky:

- 1) Navržení celkového konceptu polohování manipulátoru, což bude zahrnovat stanovení počátku souřadného systému operátora, navržení řídicího řetězce, stanovení a transformace souřadných systémů a výběr vhodné počáteční pozice manipulátoru Katana HD300s.
- 2) Nasnímaní testovacích dat polohy ruky operátora pomocí zařízení Kinect. Bude potřeba ověřit validitu takto nasnímaných dat v prostředí Matlab a také provést přepočty u těchto dat ze souřadného systému senzoru do souřadného systému manipulátoru pro potřeby simulace.
- 3) Vytvoření simulace manipulátoru v prostředí Matlab. Simulace bude provedena na testovacích datech a bude tak ověřen navržený koncept polohování manipulátoru.
- 4) Návrh řídicí aplikace v jazyce C#. Zde bude potřeba naimplementovat snímání dat ze senzorů, přímou a zpětnou úlohu kinematiky a navrhnout a implementovat grafické uživatelské rozhraní.
- 5) Testování uživatelského rozhraní bez připojeného manipulátoru.
- 6) Testování vytvořené aplikace pro ovládání manipulátoru. Testování bude probíhat na nejméně dvou operátorech, přičemž bude zkoumána schopnost koncového efektoru kopírovat polohu ruky operátora. Pokud budou výsledky praktických testů uspokojivé, budou provedeny testy na komplexní manipulační úloze. Úkolem operátora by v takovémto případě bylo uchopení a přesunutí předem definovaného předmětu na jiné místo.

## 4 Realizace

V této části práce je popsán postup při implementaci bezdotykového řízení koncového efektoru manipulátoru. To zahrnuje návrh konceptu polohování manipulátoru, přepočít souřadných systémů, snímání polohy dlaně, offline simulace a návrh řídicí aplikace. Konec kapitoly je věnován online simulaci a testování.

### 4.1 Návrh celkového konceptu polohování manipulátoru

Cílem je vytvořit takové ovládání, které bude přirozené a také nebude klást vysoké požadavky na operátora. Bezdotykovým ovládáním je řízen stacionární manipulátor Katana HD300s s pěti stupni volnosti. Je tedy nutné vymyslet jak ovládat těchto pět motorů zároveň, aby nevznikaly prodlevy a ovládání bylo plynulé.

Navržený koncept polohování manipulátoru spočívá v kopírování pohybu ruky operátora koncovým efektem manipulátoru v reálném čase. Z důvodu přirozenějšího ovládání je pohyb kopírován zrcadlově. Polohové řízení jednotlivých motorů manipulátoru bude počítáno pomocí vhodné metody řešení inverzní kinematiky. V práci jsou diskutovány dva možné přístupy, které se liší hlavně v počátečním bodu souřadného systému operátora. Je tedy nutné posoudit tyto přístupy a zvolit, který návrh bude přívětivější pro operátora.

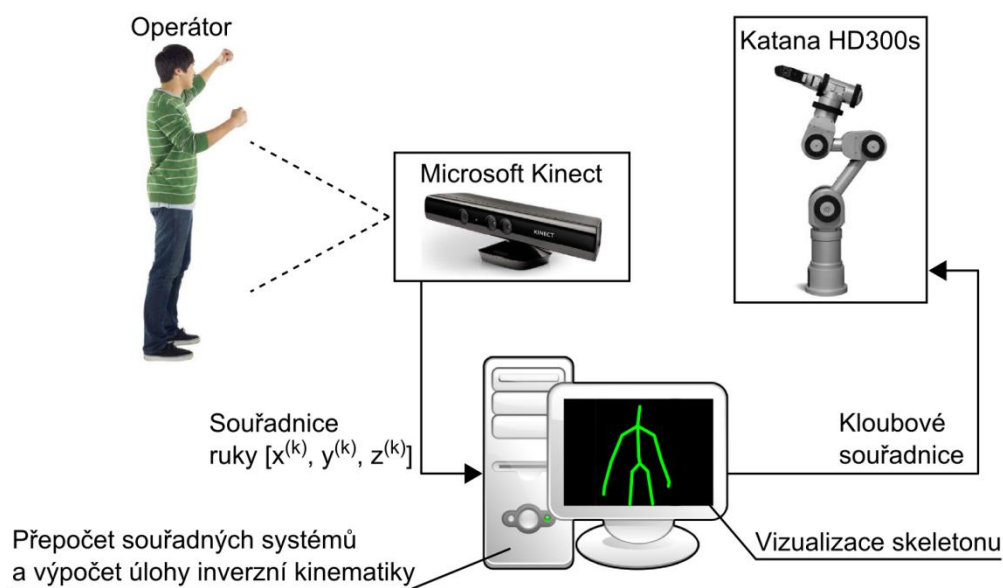
První návrh má počátek souřadného systému v poloze ramene operátora. To znamená, že polohu koncového efektoru ovlivňuje vzdálenost ruky operátora od jeho ramene. Pokud se pozice ruky oproti rameni nemění, může se operátor libovolně pohybovat, ale koncový efektor manipulátoru svou polohu nemění. Nevýhoda spočívá v rozsahu ruky, protože každý člověk má jinou délku ruky a proto je nutná kalibrace, při které se musí přepočítat délka ruky operátora na maximální délku ramene manipulátoru. Před spuštěním ovládání je operátor vyzván ke kalibraci pomocí vzpažení, předpažení a upažení. Tímto je zajištěno, že se rozsah ruky operátora rovná pracovnímu prostoru manipulátoru. Další nevýhoda, která s tímto vzniká, je nutnost provedení kalibrace při každé změně operátora. Tento problém se dá vyřešit pomocí uložení hodnot nasnímaných při kalibraci a vytvoření profilu operátora. Při dalším spuštění si operátor vybere svůj profil bez nutnosti kalibrace. Dalším možným problémem může být náročné ovládání za úrovní ramene neboli v záporných hodnotách osy  $z^{(0)}$  viz souřadné systémy na Obr 20.

Druhý navržený koncept polohování manipulátoru má počátek souřadného systému operátora v poloze ruky při spuštění aplikace. To znamená, že před spuštěním je uživatel vyzván k umístění ruky do pohodlné pozice pro ovládání manipulátoru a teprve potom je aplikace spuštěna gestem levé ruky. Koncový efektor manipulátoru je závislý pouze na poloze operátorovi ruky a je nezávislý na zbytku těla. V tomto návrhu není potřeba zpracovávat informace o poloze

ramene, ale pouze o poloze ruky. Také není potřeba kalibrace, délka operátorovi ruky neovlivňuje ovládání. Operátor může dosáhnout kamkoli v pracovním prostoru manipulátoru, i v záporných hodnotách osy  $z^{(0)}$ .

Po analýze navržených konceptů polohování manipulátoru je zjištěno několik nevýhod prvního návrhu oproti druhému. Z tohoto důvodu je vybrán druhý koncept polohování manipulátoru, u kterého není potřeba kalibrace, délka ruky operátora neovlivňuje velikost pracovního prostoru a pro ovládání manipulátoru je potřeba zpracovat pouze informace o poloze ruky operátora, což zjednodušuje přepočítání souřadnic.

Navržený koncept je tedy založen na zrcadlovém kopírování pohybu ruky operátora koncovým efektem manipulátoru v reálném čase. Operátora je snímán zařízením Kinect, z kterého jsou získány souřadnice ruky v souřadném systému senzoru. Pomocí transformačních matic jsou tyto souřadnice převedeny do souřadného systému manipulátoru Katana. Poté jsou úlohou inverzní kinematiky spočítány úhly natočení jednotlivých kloubů manipulátoru, které jsou odeslány na vstup stávajícího řízení, pomocí kterého je změněna poloha manipulátoru. V průběhu ovládání je v aplikaci vykreslen skeleton operátora a menu pro ovládání aplikace.



Obr 18. Schéma řídicího řetězce

## 4.2 Stanovení výchozí polohy koncového efektoru manipulátoru

Po spuštění aplikace je propojena vzájemná poloha souřadných systémů operátora a manipulátoru na základě polohy ruky operátora a polohy koncového efektoru manipulátoru. Z tohoto důvodu je důležité zvolit takovou výchozí pozici

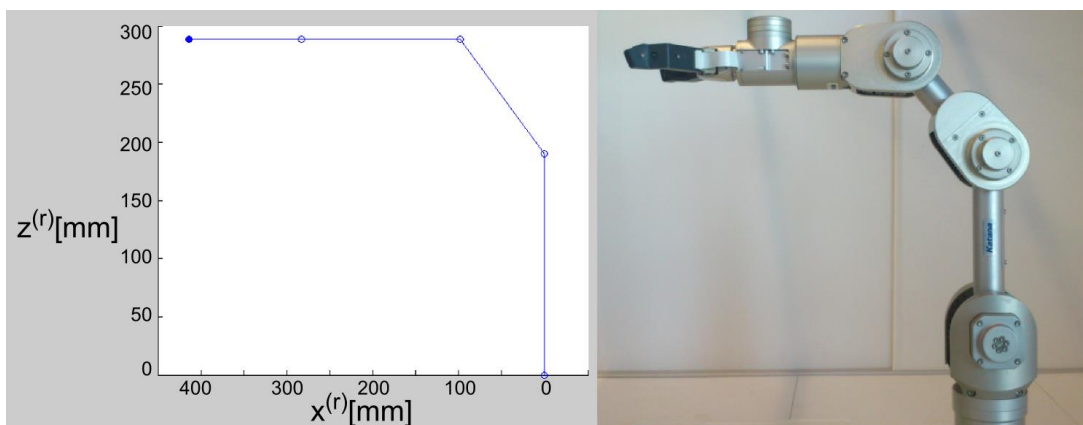
koncového efektoru, která bude zajišťovat okamžitou možnost manipulace s objekty v daném pracovním prostoru manipulátoru. Po kalibraci se manipulátor nachází v poloze, která je na Obr 6. Tato pozice je nevyhovující, jelikož je manipulátor otočen o  $180^\circ$  kolem osy  $z^{(r)}$  od pracovního prostoru, což by mohlo způsobovat problémy s rozsahem motorů. Proto je potřeba pomocí přímé úlohy kinematiky zjistit vhodnější pozici.

Katana HD300s bude v budoucnu pracovat s objekty na průmyslovém dopravním pásu, který bude umístěn v pracovním prostoru před manipulátorem. Z tohoto důvodu je stanovena výchozí pozice koncového efektoru do bodu nad plánovaným umístěním dopravního pásu. Pozice odpovídá těmto úhlům:

Tab. 4 Výchozí poloha manipulátoru

Číslo motoru	Úhel [°]
1	180
2	90
3	135
4	225
5	180

Přičemž poloha motoru 5 neovlivňuje výchozí polohu koncového efektoru manipulátoru, určuje pouze natočení okolo osy  $x^{(r)}$  v  $SS^{(r)}$  a motorem 6 se řídí sevření uchopovače. Tyto úhly jsou použity jako vstup pro přímou úlohu kinematiky, která byla naimplementována v prostředí Matlab a je zjištěna poloha koncového efektoru manipulátoru odpovídající souřadnicím  $[413, 0, 288]$  v souřadném systému manipulátoru v milimetrech. Výchozí pozici manipulátoru lze vidět na Obr 19, kde lze vidět vypočítaná pozice pomocí přímé úlohy kinematiky ve srovnání s reálnou pozicí manipulátoru Katana HD300s.



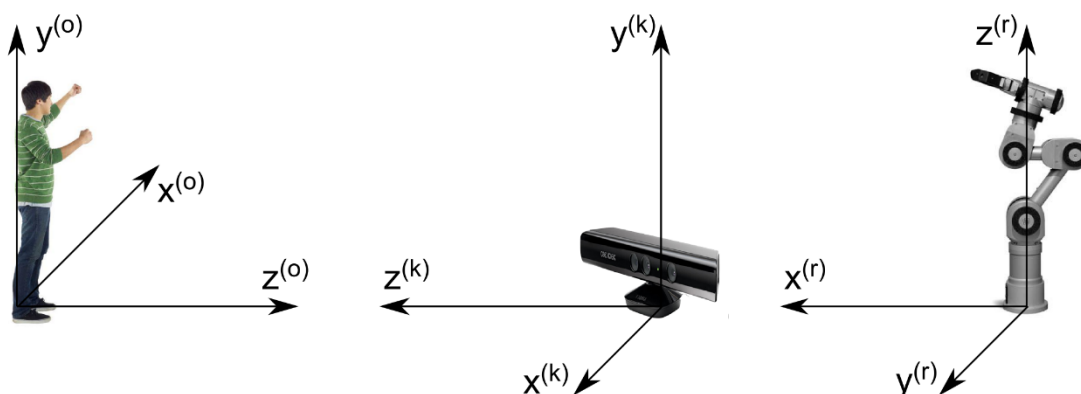
Obr 19. Výchozí poloha manipulátoru v prostředí Matlab a ve skutečnosti

### 4.3 Určení požadované polohy efektoru

Tato kapitola popisuje výpočet požadované polohy efektoru, která je závislá na poloze ruky operátora. Z tohoto důvodu je řešena problematika transformace polohy ruky ze souřadného systému sensoru do souřadného systému manipulátoru.

#### 4.3.1 Transformace souřadných systémů

V této práci jsou řešeny souřadné systémy operátora, senzoru Kinect a manipulátoru Katana HD300s. Souřadné systémy jsou vzájemně potočeny dle Obr 20.  $SS^{(o)}$ , kde  $SS$  označuje souřadný systém, je oproti  $SS^{(k)}$  pootočen o  $180^\circ$  kolem osy  $y^{(o)}$ .  $SS^{(r)}$  je pootočen o  $90^\circ$  kolem osy  $x^{(k)}$  a o  $90^\circ$  kolem osy  $y^{(k)}$  vůči  $SS^{(k)}$ .



Obr 20. Srovnání souřadných systémů (zleva: operátor, Kinect a manipulátor)

Přepočet souřadných systémů je proveden pomocí transformačních matic, viz rovnice (12). Pro rotaci kolem osy  $x$  je použita transformační matice  $M^{RotX}$ , pro rotaci kolem osy  $y$  je použita matice  $M^{RotY}$  a pro rotaci kolem osy  $z$  je použita matice  $M^{RotZ}$ .

$$M^{RotX} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{bmatrix}, M^{RotY} = \begin{bmatrix} \cos \varphi & 0 & \sin \varphi \\ 0 & 1 & 0 \\ -\sin \varphi & 0 & \cos \varphi \end{bmatrix}, M^{RotZ} = \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (12)$$

Z praktických důvodů se zavádějí Denavit–Hartenberg parametry a poté je nutné transformační matici rozšířit takto:

$$M^{RotX} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi & 0 \\ 0 & \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$



Pokud je potřeba nejen rotace souřadného systému, ale i posuv, používá se translační matice, kde je navíc druhý systém posunut oproti prvnímu v jednotlivých osách o [posuvX, posuvY, posuvZ].

$$M^{Trans} = \begin{bmatrix} 1 & 0 & 0 & posuvX \\ 0 & \cos \varphi & -\sin \varphi & posuvY \\ 0 & \sin \varphi & \cos \varphi & posuvZ \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

### 4.3.2 Výpočet polohy efektoru

Poloha ruky  $H^{(k)}$  v souřadném systému senzoru je nasnímána pomocí standardního SDK. Poloha ruky je snímána 30 krát za vteřinu a je použita jako vstup výpočtu požadované polohy koncového efektoru. Na základě těchto skutečností je označena počáteční poloha ruky operátora v souřadném systému senzoru jako  $H_0^{(k)}$  a výchozí poloha efektoru v souřadném systému manipulátoru jako  $G_0^{(r)}$ . Propojení polohy ruky operátora a polohy koncového efektoru je provedeno pouze jednou po spouštěcím gestu levé ruky. Všechny další polohy efektoru,  $G_i^{(r)}$ , jsou počítány na základě rovnice:

$$G_i^{(r)} = M^{RotX}(\varphi_1) \cdot M^{RotY}(\varphi_2) \cdot M^{Trans}(\Delta x, \Delta y, \Delta z) \cdot H_i^{(k)} \quad (15)$$

Kde  $H_i^{(k)}$  označuje novou polohu ruky operátora a úhly  $\varphi_1$  a  $\varphi_2$  odpovídají hodnotě  $90^\circ$ , což vyplývá z pootočení souřadných systémů, viz Obr 20. Parametry  $M^{Trans}$  jsou spočteny podle rovnice (16), kde je vypočítán rozdíl výchozí polohy efektoru v souřadném systému senzoru  $G_0^{(k)}$  a počáteční polohy ruky operátora v souřadném systému senzoru  $H_0^{(k)}$ .

$$\Delta = G_0^{(k)} - H_0^{(k)} \quad (16)$$

## 4.4 Výpočet konfigurace manipulátoru

V této kapitole je uveden postup při řešení výpočtu konfigurace manipulátoru, což zahrnuje algoritmus inverzní úlohy kinematiky. V předchozí kapitole je uveden výpočet požadované polohy koncového efektoru manipulátoru. Na základě této informace je dále prováděn výpočet kloubových souřadnic.

### 4.4.1 Algoritmus inverzní úlohy kinematiky

Pro řešení inverzní úlohy kinematiky byla použita metoda z Matlab Robotics Toolbox. Tato metoda je iterativní a je založená na výpočtu transpozice Jacobiánu. Kloubové souřadnice,  $q$ , lze vypočítat pomocí transpozice Jakobiho matice  $J$ , viz rovnice (17) (Siciliano, 2009).

$$\Delta q = \mathbf{J}^T \Delta x \quad (17)$$

Kde  $x$  je známá poloha koncového efektoru manipulátoru. Na základě této rovnice lze iteračně vypočítat polohu nových kloubových souřadnic  $q_{i+1}$ .

$$q_{i+1} = q_i + \mathbf{J}^T e \quad (18)$$

Známa hodnota kloubových souřadnic  $q_i$  je spočtená pomocí přímé úlohy kinematiky a chybu aktuální polohy  $e$  lze vyjádřit takto:

$$e = x_{cil} - x_i \quad (19)$$

Kde  $x_{cil}$  je požadovaná poloha koncového efektoru a  $x_i$  je poslední známá poloha koncového efektoru (Siciliano, 2009).

Dále je uveden obecný zápis pro výpočet inverzní úlohy kinematiky pro posun efektoru z výchozí polohy stanovené v kapitole 4.2. Pro tuto stanovenou polohu jsou známy:

- Kloubové souřadnice  $q_i$ ,

$$q_i = [\alpha_i \quad \beta_i \quad \gamma_i \quad \delta_i] \quad (20)$$

- poloha efektoru v souřadném systému manipulátoru  $G_0^{(r)} = [g_x^{(r)} \quad g_y^{(r)} \quad g_z^{(r)}]$   
Nejprve je vypočítána chyba aktuální polohy  $e$  pomocí rovnice (19).

$$e = \begin{bmatrix} x_{cil}^{(r)} \\ y_{cil}^{(r)} \\ z_{cil}^{(r)} \\ 1 \end{bmatrix} - \begin{bmatrix} g_{xi}^{(r)} \\ g_{yi}^{(r)} \\ g_{zi}^{(r)} \\ 1 \end{bmatrix} = \begin{bmatrix} \Delta x^{(r)} \\ \Delta y^{(r)} \\ \Delta z^{(r)} \\ 0 \end{bmatrix} \quad (21)$$

Následně je spočtena Jakobiho matice:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} & \dots & \frac{\partial f_1}{\partial q_4} \\ \frac{\partial f_2}{\partial q_1} & \dots & \dots & \frac{\partial f_2}{\partial q_4} \\ \vdots & & & \vdots \\ \frac{\partial f_6}{\partial q_1} & \dots & \dots & \frac{\partial f_6}{\partial q_4} \end{bmatrix}, \mathbf{J}^T = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \frac{\partial f_2}{\partial q_1} & \dots & \frac{\partial f_6}{\partial q_1} \\ \frac{\partial f_1}{\partial q_2} & \dots & \dots & \frac{\partial f_6}{\partial q_2} \\ \vdots & & & \vdots \\ \frac{\partial f_1}{\partial q_4} & \dots & \dots & \frac{\partial f_6}{\partial q_4} \end{bmatrix} \quad (22)$$

Derivace nahrazujeme velikostí přírůstku:

$$\frac{\partial f_i}{\partial q_j} \doteq \frac{f_i(q_j + \Delta q_j) - f_i(q_j)}{\Delta q_j} \quad (23)$$

A poté jsou již spočteny nové kloubové souřadnice pro požadovanou polohu koncového efektoru podle rovnice (18).

$$q_{i+1} = [\alpha_i \quad \beta_i \quad \gamma_i \quad \delta_i] + [\Delta\alpha \quad \Delta\beta \quad \Delta\gamma \quad \Delta\delta] \quad (24)$$

Výsledkem je matice kloubových souřadnic v radiánech.

$$q_{i+1} = [\alpha_{i+1} \quad \beta_{i+1} \quad \gamma_{i+1} \quad \delta_{i+1}] \quad (25)$$

Nakonec je spočtena Euklidova norma matice  $e$ . Pokud je norma menší, než požadovaná tolerance, výpočet končí. Pokud je norma větší, je výpočet opakován od rovnice (21) s nově spočtenými kloubovými souřadnicemi. Tento proces se opakuje, dokud není splněna stanovená tolerance nebo dokud není překročen maximální stanovený iterační limit.

Zde je uveden konkrétní příklad pro výpočet inverzní úlohy kinematiky pro posun efektoru z výchozí polohy stanovené v kapitole 4.2. Ve výpočtech jsou uvažovány úhly v radiánech a vzdálenost v metrech. Pro výchozí stanovenou polohu jsou známy:

- Kloubové souřadnice  $q_i$ , viz Tab. 4, zde uvedené v radiánech,

$$q_i = [3,142 \quad 1,571 \quad 2,356 \quad 3,927] \quad (26)$$

- poloha efektoru v souřadném systému manipulátoru  $G_0^{(r)} = [0,413 \quad 0 \quad 0,288]$   
Výpočet chyby aktuální polohy  $e$ :

$$e = \begin{bmatrix} 0,397 \\ -0,054 \\ 0,298 \\ 1 \end{bmatrix} - \begin{bmatrix} 0,413 \\ 0 \\ 0,288 \\ 1 \end{bmatrix} = \begin{bmatrix} -0,016 \\ -0,054 \\ 0,009 \\ 0 \end{bmatrix} \quad (27)$$

Výpočet Jakobiho matice:

$$J = \begin{bmatrix} 0 & -2,288 & -0,098 & 0 \\ 0,413 & 0 & 0 & 0 \\ 0 & 0,413 & 0,413 & 0,315 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & -1 \\ 1 & 0 & 0 & 0 \end{bmatrix}, J^T = \begin{bmatrix} 0 & 0,413 & 0 & 0 & 0 & 1 \\ -0,288 & 0 & 0,413 & 0 & -1 & 0 \\ -0,098 & 0 & 0,413 & 0 & -1 & 0 \\ 0 & 0 & 0,315 & 0 & -1 & 0 \end{bmatrix} \quad (28)$$

Výpočet kloubových souřadnic pro požadovanou polohu koncového efektoru:

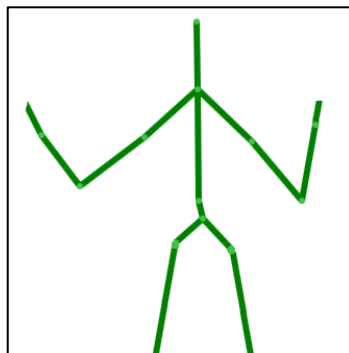
$$q_{i+1} = [3,142 \quad 1,571 \quad 2,356 \quad 3,927] + [-0,022 \quad 0,009 \quad 0,005 \quad 0,003] \quad (29)$$

Výsledná matice kloubových souřadnic v radiánech:

$$q_{i+1} = [3,119 \quad 1,579 \quad 2,362 \quad 3,924] \quad (30)$$

#### 4.4.2 Detekce polohy ruky a analýza trajektorie

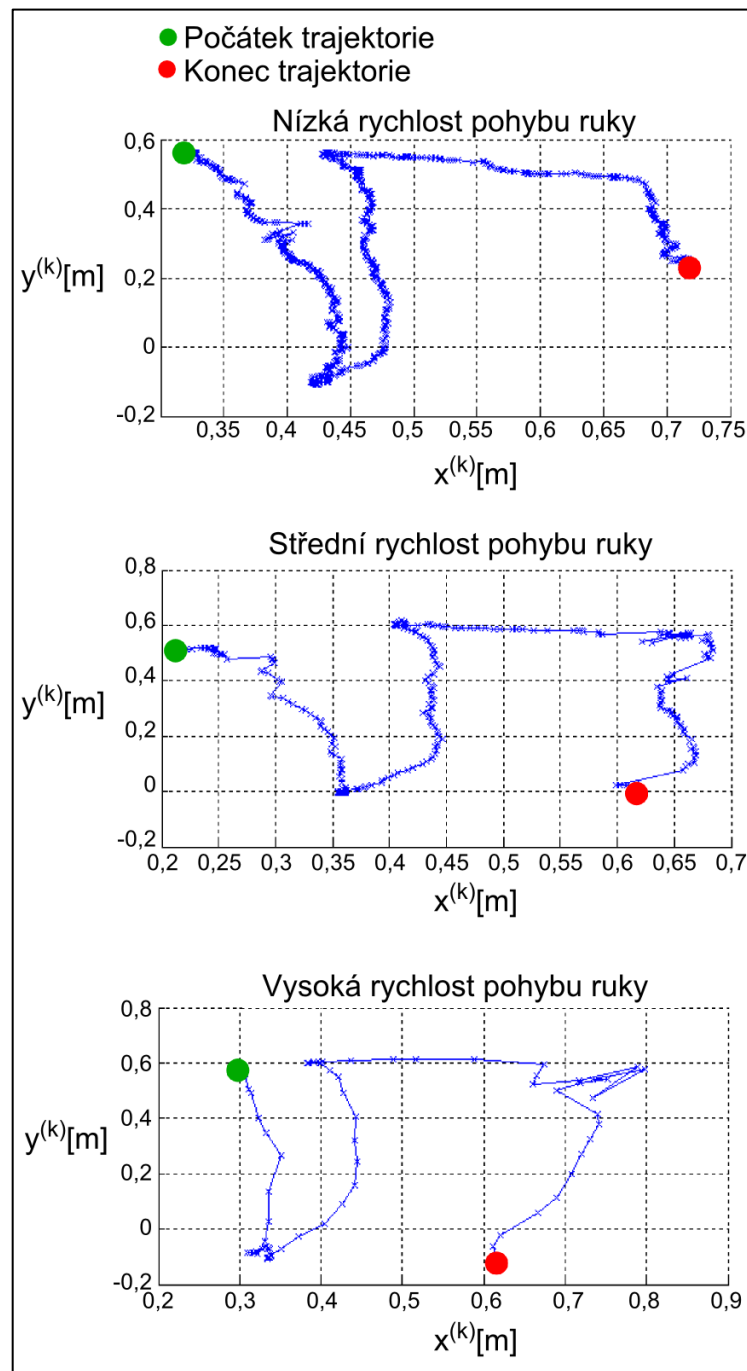
Pro snímání lidského těla je použito zařízení Kinect verze 1 s SDK verze 1.8, které poskytuje ukázkovou aplikaci Skeleton Basics-WPF v jazyce C# pro snímání lidského těla a vykreslení skeletonu. Kromě režimu pro sezení, kdy je snímána pouze horní polovina těla nemá tato aplikace žádné další funkce.



Obr 21. Skeleton Basics-WPF v jazyce C#

Z této aplikace lze jednoduše získat informace o poloze dvaceti kloubů operátora včetně polohy ruky, která je v této práci důležitá. Z těchto důvodů byla tato ukázková aplikace použita pro nasnímání testovacích dat. Bylo nasnímáno deset trajektorií ruky s různou rychlostí pohybu, které simulovali sebrání předmětu a jeho přesunutí zprava doleva, zleva doprava, zepředu dozadu a zezadu dopředu. Informace o poloze ruky byly uloženy do textového souboru. Na každém řádku souboru byly informace o souřadnicích ruky  $x$ ,  $y$ ,  $z$  v souřadném systému senzoru Kinect oddělené mezerami.

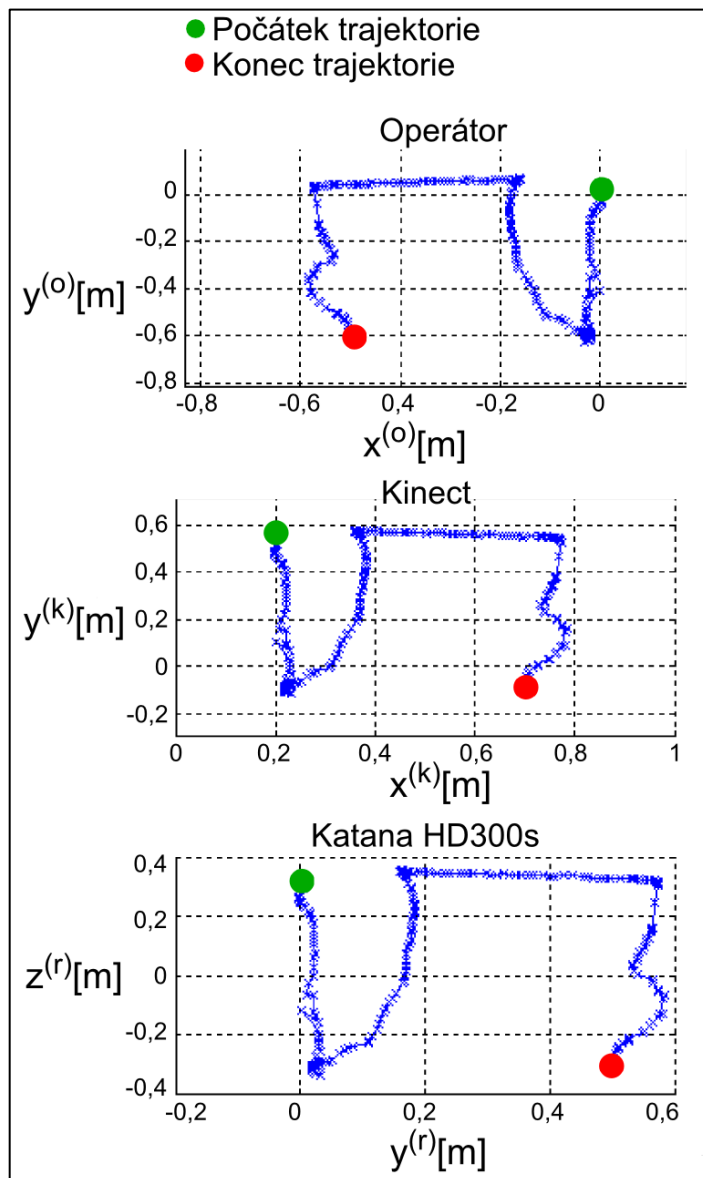
Tyto data byla načtena a vykreslena v prostředí Matlab. Na Obr 22 je nasnímaná trajektorie pohybu ruky při zvednutí a přesunutí předmětu. Frekvence snímání senzoru Kinect je 30 snímků za vteřinu. Na obrázku lze vidět rozdíl v rychlostech pohybu ruky. Na základě proběhlých testování bylo zvoleno subjektivní označení rychlostí pohybu ruky nízká, střední, vysoká. Přičemž nízká rychlost odpovídá nejpomalejšímu pohybu ruky operátora, vysoká odpovídá nejvyšší rychlosti pohybu a střední je průměrná rychlost pohybu. Při střední rychlosti pohybu ruky bylo naměřeno 342 bodů. Při nízké rychlosti pohybu byl počet nasnímaných bodů 745, což je více než 200 % oproti střední rychlosti. Naopak při vysoké rychlosti pohybu byl počet nasnímaných bodů 144, což je méně než 50 % oproti střední rychlosti. Také lze vidět šum v trajektorii, který by mohl ovlivnit schopnost manipulátoru kopírovat pohyb ruky.



Obr 22. Vykreslená trajektorie ruky

Pomocí transformačních matic pro rotaci a translaci byly přepočteny data ze souřadného systému senzoru Kinect do souřadného systému operátora a manipulátoru. Na Obr 23 lze vidět, že počátek trajektorie operátora je v počátku souřadného systému a je pootočen o  $180^\circ$  okolo osy  $y^{(o)}$  oproti souřadnému systému senzoru Kinect. Počátek trajektorie manipulátoru je ve

výchozí poloze koncového efektoru stanovené v kapitole 4.2 a souřadný systém je pootočen o  $90^\circ$  kolem osy  $x^{(k)}$  a o  $90^\circ$  kolem osy  $y^{(k)}$  vůči  $SS^{(k)}$ .



Obr 23. Trajektorie ruky v souřadném systému operátora, senzoru a manipulátoru

## 4.5 Simulační ověření proveditelnosti

Pro simulační testy bylo zvoleno prostředí Matlab s rozšířením Robotics Toolbox, které poskytuje třídy a funkce pro simulační výpočty přímé a inverzní úlohy kinematiky, dynamiky a plánování trajektorie robotických ramen. Toolbox obsahuje předdefinovaná robotická ramena Puma 560 a Stanford. Deváté vydání Robotics Toolbox bylo rozšířeno také o podporu mobilních robotů. Toolbox po-

skytuje algoritmy pro plánování cesty pozemních robotů, lokalizaci, tvorbu map a simultánní lokalizaci a mapování. Toolbox také obsahuje podrobný model kvadrokoptéry (Corke, 2011).

Vlastní manipulátor lze vytvořit pomocí třídy `SerialLink`, která obsahuje části manipulátoru tvořené třídou `Link`, jenž jsou spojeny vazbami rotačními nebo posuvnými. Každá vazba znamená jeden stupeň volnosti. Manipulátor Katana HD300s má pět stupňů volnosti, ale pro zjednodušení výpočtu byly v modelu poslední dvě vazby sloučeny, jelikož motor 5 (viz Obr 6) neovlivňuje polohu koncového efektoru, pouze jeho natočení. Všechny vazby byly nadefinovány jako rotační.

Vazby se definují takto:

```
L(1) = Revolute('d', 0, 'a', 0, 'alpha', pi/2);
L(2) = Revolute('d', 0, 'a', 0.190, 'alpha', 0);
L(3) = Revolute('d', 0, 'a', 0.139, 'alpha', 0);
L(4) = Revolute('d', 0, 'a', 0.315, 'alpha', 0);
```

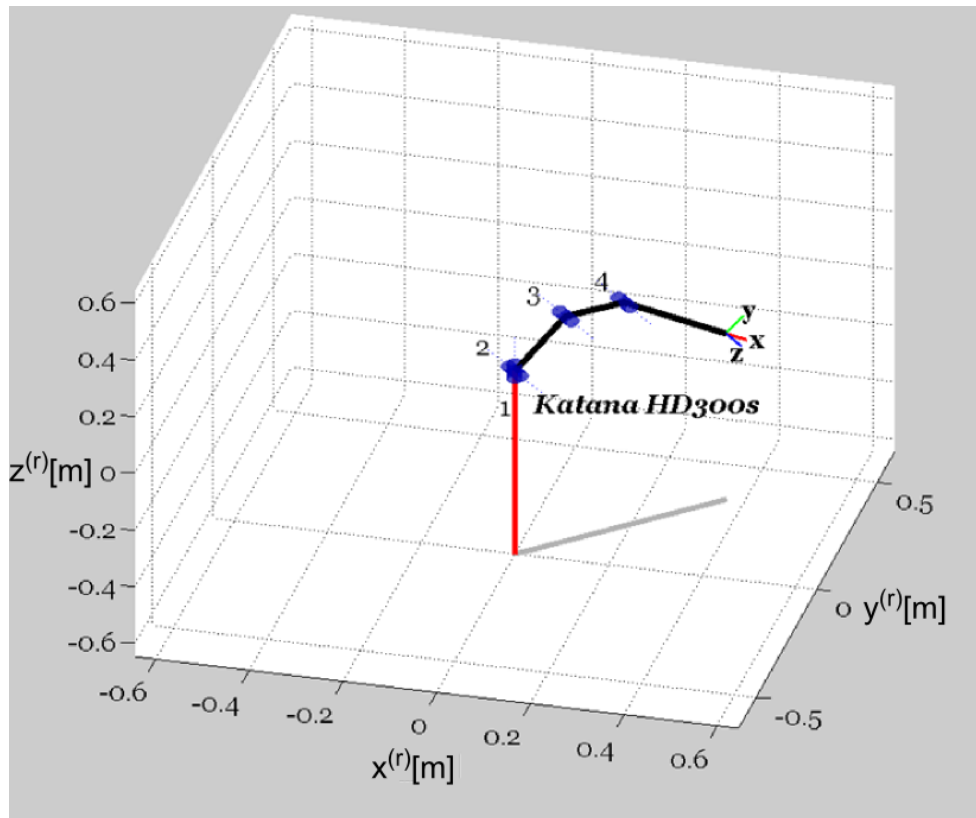
kde

- `Revolute()` znamená definici rotační vazby,
- parametr `d` určuje posun vazby, který je v případě manipulátoru Katana HD300s nulový,
- parametr `a` určuje délku vazby v metrech
- a parametr `alpha` určuje pootočení vazby, které je nenulové pouze v případě první vazby.

Po nadefinování všech vazeb je vytvořen model manipulátoru tímto způsobem:

```
KatanaHD300s = SerialLink(L, 'name', 'Katana HD300s');
```

Je použit konstruktor pouze s volitelným parametrem *name*, který označuje název modelu. Funkcí `plot(q)`, kde *q* je matice s úhly natočení jednotlivých motorů, je možné vykreslit výsledný model, který lze vidět na Obr 24.



Obr. 24. Model manipulátoru Katana HD300s

Pro výpočet úlohy inverzní kinematiky byla použita metoda z Robotics Toolbox `ikine()`. Metoda je iterativní, je založená na výpočtu Jacobiánu a volá se následujícím způsobem:

```
ikine(T, startJoints, M, 'tol', 0.2, 'ilimit', 100);
```

kde

- parametr `T` označuje požadovanou pozici koncového efektoru,
- parametr `startJoints` určuje počáteční odhad polohy manipulátoru, který je roven předcházejícímu řešení metody `ikine()`
- `M` je matice masky definující počet stupňů volnosti pro manipulátory s méně, než šesti stupni volnosti,
- `tol` je volitelný parametr určující akceptovatelnou odchylku požadované a spočtené polohy koncového efektoru
- `ilimit` je volitelný parametr určující maximální počet iterací.

Pro `KatanaHD300s` má matice `M` tvar `[1 1 1 1 0 0]`, což odpovídá čtyřem stupňům volnosti, protože poslední dvě vazby namodelovaného manipulátoru jsou sloučené. Hodnoty pro parametry `tol` a `ilimit` byly zvoleny na základě výsledků simulačního ověření.

Po simulačním ověření metody `ikine()` bylo přistoupeno k simulaci na testovacích datech nasnímaných senzorem Kinect, aby byl ověřen navržený kon-



cept polohování manipulátoru. Nejprve bylo nutné nastavit počáteční parametry, jako je například výchozí pozice manipulátoru a počet stupňů volnosti. Poté byla načtena testovací data z textového souboru a z každého řádku, který obsahoval souřadnice ruky v souřadném systému manipulátoru, byly vypočítány úhly natočení jednotlivých motorů manipulátoru. Takto vypočítaná poloha manipulátoru byla v každém kroku vykreslena funkcí `plot()`. Výsledkem byla sekvence vykreslených modelů manipulátoru jako na Obr 24, kde měl koncový efektor každého modelu polohu načtenou z testovacích dat.

Proběhl deset simulací s deseti různými nasnímanými trajektoriemi ruky, na kterých byla ověřena schopnost koncového efektoru modelu kopírovat zadané souřadnice ruky v souřadném systému manipulátoru. U některých testovacích dat bylo možné snížit parametr *tol* o jedno desetinné místo. Data o polohách manipulátoru byla uložena opět do textového souboru pro otestování stávajícího řízení manipulátoru a ověření validity dat.

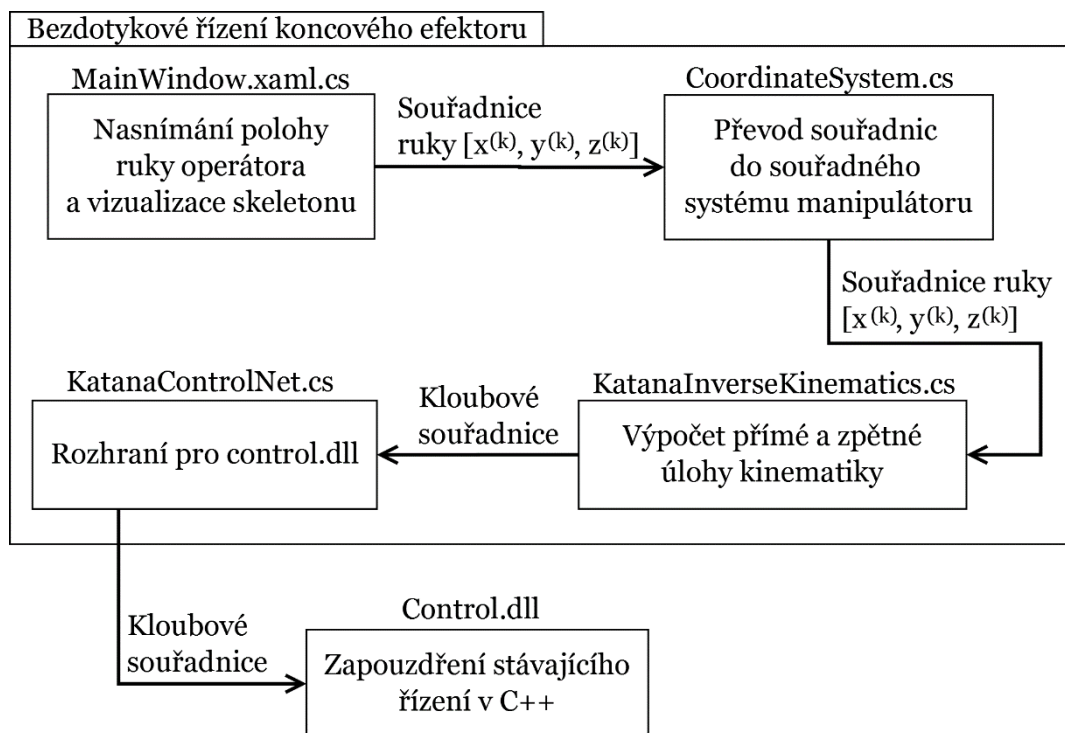
Testováním byla ověřena schopnost koncového efektoru manipulátoru kopírovat offline nasnímaný pohyb ruky. Při testování byly postupně posílány na vstup stávajícího řízení data z textového souboru s polohami vypočítanými pomocí úlohy inverzní kinematiky z Robotics Toolbox. Manipulátor byl schopen dosáhnout všech poloh, kromě výjimek, kdy byl koncový efektor příliš nízko, a hrozila kolize s podstavou ochranné klece.

## 4.6 Návrh a implementace aplikace

Před vlastní implementací bylo nutno rozhodnout, v kterém programovacím jazyku bude aplikace realizována. SDK senzoru Kinect podporuje jazyky C++, C# a Visual Basic .NET a z těchto jazyků byl vybrán C# kvůli uživatelské přívětivosti. Řídicí aplikace je tedy implementována v jazyce C# s využitím vývojového prostředí Visual Studio 2012 a pro grafické uživatelské rozhraní byla použita technologie Windows Presentation Foundation (WPF), kterou poskytuje .NET Framework.

Stávající řízení manipulátoru Katana HD300s je napsané v jazyce C++ a proto bylo nutné využít knihovny `control.dll`, která umožňuje použití stávajícího řízení v jazyce C#. Schéma funkcionality tříd lze vidět na Obr 25.

Pro implementaci následujících tříd byla využita knihovna Math.NET Numerics, která je částí Math.NET iniciativy a je výsledkem sloučení `dnAnalytics` s `Math.NET Iridium`. Knihovna je volně dostupná a je platformně nezávislá (Ruegg, Van Gael, Cuda, 2015). Mimo jiné knihovna poskytuje metody pro práci s maticemi. Pomocí těchto metod lze například vytvořit jednotkovou matici, spočítat determinant matice nebo transponovat matici. Metody také umožňují základní aritmetické operace s maticemi, umožňují sloučení nebo rozdělení matic atd.



Obr 25. Schéma funkcionality tříd

#### 4.6.1 Implementace výpočetní části aplikace

V této kapitole je popsán postup implementace části aplikace zajišťující výpočet polohy koncového efektoru manipulátoru. To zahrnuje snímání dat ze senzoru Kinect, transformace takto získaných dat do souřadného systému manipulátoru a přímou a zpětnou úlohu kinematiky.

Snímání dat ze senzoru Kinect bylo vytvořeno na základě ukázkové aplikace z SDK. Kód byl pročištěn o všechny nepotřebné metody. Na základě výše zvoleného celkového konceptu polohování manipulátoru byla zpracována pouze data o poloze ruky operátora. Ostatní nasnímaná data byla použita pouze pro vykreslení skeletonu v grafickém uživatelském rozhraní. Dále bylo nutné si uložit první nasnímanou polohu ruky operátora, která je zároveň počátkem souřadného systému operátora. Počáteční poloha ruky je potřebná pro spočtení aktuální vzdálenosti ruky operátora od počátku souřadného systému.

Pro transformace souřadných systémů byla vytvořena statická třída `CoordinateSystem`. V této třídě byly naimplementovány metody pro rotace kolem os  $x$ ,  $y$ ,  $z$  a translaci, které byly dále použity v metodách `KinectToKatana()` a `KinectToOperator()` zajišťující transformace souřadnic ze souřadného systému senzoru do souřadných systémů operátora a manipulátoru.

Výpočet přímé a zpětné úlohy kinematiky je řešen třídou `KatanaInverseKinematics`, která je založená na metodě řešení inverzní kinematiky z `Robotics Toolbox`.

Tab. 5 Vybrané atributy a metody třídy `KatanaInverseKinematics`

Název atributu / metody	Datový / návratový typ	Popis
<code>minHeight</code>	<code>double</code>	minimální výška koncového efektoru
<code>anglesStartPosition</code>	<code>Matrix&lt;double&gt;</code>	počáteční poloha manipulátoru
<code>kinectStartCoordinate</code>	<code>Matrix&lt;double&gt;</code>	počáteční poloha ruky v souřadném systému senzoru Kinect
<code>Fkine()</code>	<code>Matrix&lt;double&gt;</code>	výpočet přímé úlohy kinematiky
<code>jacob0()</code>	<code>Matrix&lt;double&gt;</code>	výpočet Jakobiho matice v souřadném systému manipulátoru
<code>jacobn()</code>	<code>Matrix&lt;double&gt;</code>	výpočet Jakobiho matice v souřadném systému koncového efektoru
<code>Ikine()</code>	<code>Matrix&lt;double&gt;</code>	výpočet zpětné úlohy kinematiky

Atribut `minHeight` určuje minimální výšku koncového efektoru v souřadném systému manipulátoru. Pomocí tohoto atributu je zabráněno kolizi manipulátoru s podstavou ochranné klece, ve které je manipulátor umístěn. Atribut `anglesStartPosition` je matice s úhly výchozí polohy manipulátoru, která byla stanovena v kapitole 4.2. První nasnímaná poloha ruky operátora, která je zároveň počátkem souřadného systému operátora je uložena do atributu `kinectStartCoordinate`. Třída má pouze dvě veřejné metody `Fkine()` a `Ikine()` pro výpočet přímé a zpětné úlohy kinematiky. Metoda `getMatrixA()` slouží k vytvoření ramene modelu manipulátoru a přijímá parametry posun ramene, délku ramene a jeho pootočení. Metody `jacob0()` a `jacobn()` slouží k výpočtu Jakobiho matice.

#### 4.6.2 Návrh a implementace grafického uživatelského rozhraní

Pro navržený koncept polohování manipulátoru je nutné vytvořit takové grafické uživatelské rozhraní, které by operátorovi poskytovalo zpětnou vazbu o poloze jeho těla. Důležitá je hlavně poloha ruky, kterou bude kopírovat koncový efektor manipulátoru v reálném čase. Také je potřeba navrhnout a implementovat ovládání aplikace, aby byl operátor schopen aplikaci spustit, vypnout nebo pozastavit pomocí gesta. Další gesta budou použita pro ovládání otevírání a zavírání koncového efektoru za účelem uchopení předmětu. Pro ovládání je tedy zapotřebí vymyslet pět různých gest, která se nebudou navzájem ovlivňovat. Zároveň je nutné poskytnout operátorovi zpětnou vazbu o stavu

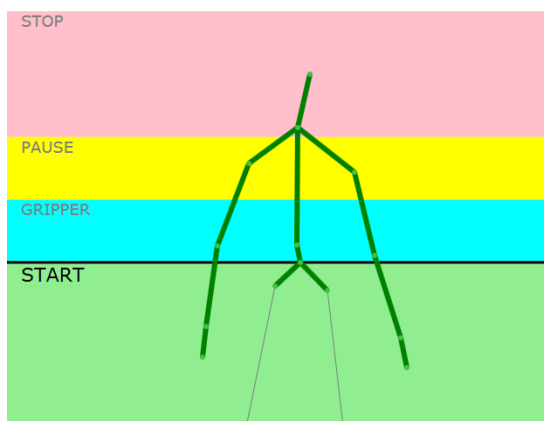
aplikace, aby byl informován, zda je aplikace spuštěná, zastavená či jen pozastavená.

Byla navržena dvě různá grafická uživatelská rozhraní, která splňují stanovené podmínky pro ovládání. Je tedy nutné posoudit tyto návrhy a zvolit, který bude přívětivější pro operátora.

V prvním návrhu je testována výška ruky operátora a podle toho je spuštěn daný příkaz. Rozložení příkazů do úrovní lze vidět na obrázku Obr 26. Úrovně jsou generovány dynamicky podle aktuálního operátora. Výhodou tohoto návrhu je nezávislost na výšce operátora. Úrovně jsou vykreslovány následovně

- od pasu dolů je úroveň START,
- od pasu do poloviny trupu je úroveň GRIPPER,
- od poloviny trupu po ramena je úroveň PAUSE,
- a od ramen nahoru je úroveň STOP.

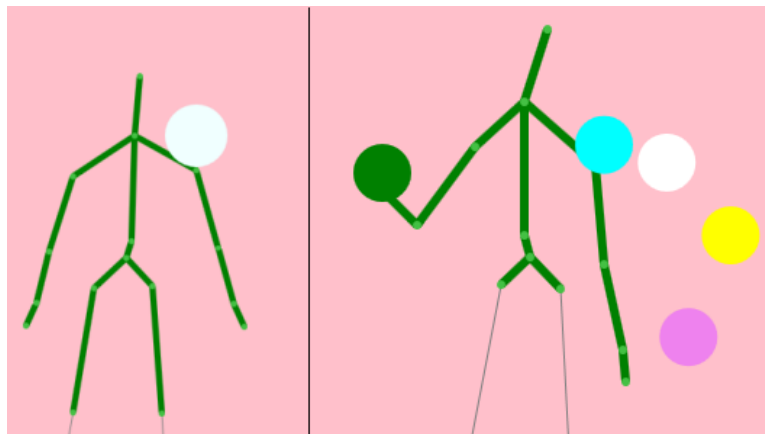
Úroveň START je zobrazena od pasu dolů, aby měl operátor ruku pro spuštění příkazů v přirozené poloze při ovládání manipulátoru. Problém, který tímto vzniká je nechtěné spuštění aplikace po příchodu operátora do zorného pole senzoru. Tento problém by se dal řešit přeuspořádáním úrovní nebo přidáním „neutrální“ úrovně, která by nespouštěla žádný příkaz. Úroveň GRIPPER funguje jako přepínač. To znamená, že pokud je efektor zavřený, otevře ho a naopak.



Obr 26. Uživatelského rozhraní založené na úrovních

Druhý návrh, který lze vidět na Obr 27, je založený na vykreslení menu příkazů okolo loktu operátora. Příkazy jsou vykreslené pomocí různobarevných kruhů, u kterých je testována kolize s rukou operátora. Před spuštěním aplikace je vykreslen kruh pro spuštění aplikace. Menu je zobrazeno dynamicky na základě vykreslené délky ruky operátora ve 2D v grafickém uživatelském rozhraní. Tento návrh je tedy také nezávislý na výšce operátora či délce jeho ruky a operátor se může libovolně pohybovat v zorném poli senzoru. Dále je okolo ruky, kterou kopíruje koncový efektor manipulátoru, vykreslen kruh označující aktuální pozici koncového efektoru. Při pozastavení aplikace zůstane tento kruh v místě, kde se nacházel efektor, při aktivaci příkazu pozastavení. Uspořádání příkazů v návrhu bylo následující:

- Horní modrý kruh pro příkaz PAUSE,
- horní bílý kruh pro OPEN,
- prostřední žlutý kruh pro příkaz CLOSE,
- a spodní fialový kruh pro příkaz STOP.



Obr 27. Startovní obrazovka aplikace (nalevo) a spuštěná aplikace (napravo)

Po analýze a testování funkčnosti navržených grafických uživatelských rozhraní je zvolen druhý návrh s vykreslováním menu příkazů okolo loktu. Tento návrh je uživatelsky přívětivější, může být rozšířen v případě potřeby o další příkazy a nemá problémy s nechtěným spuštěním. Výsledný návrh byl implementován třídou `MainWindow`. Její popis je uveden níže v Tab. 6. Vybraný návrh grafického uživatelského rozhraní byl na základě praktických testů s více operátory značně modifikován, viz kapitola Testování bez připojeného manipulátoru.

Tab. 6 Vybrané metody třídy `MainWindow`

Název metody	Popis
<code>SensorSkeletonFrameReady()</code>	reakce na další připravený snímek ze skeleton streamu
<code>countArmLength()</code>	výpočet délky ruky ve 2D
<code>initMenu2()</code>	práce s menu příkazů
<code>katanaPaused()</code>	pozastavení ovládání manipulátoru
<code>katanaMove()</code>	pohyb manipulátoru
<code>isCollisionCircle()</code>	zjištění kolize s tlačítkem ve 2D
<code>menuDraw()</code>	vykreslení menu příkazů
<code>DrawBonesAndJoints()</code>	vykreslení konkrétního skeletonu (vazby i spojnice)

Všechny vybrané metody mají návratový typ `void`, pouze `isCollisionCircle()` vrací `true` nebo `false` podle toho jestli detekuje kolizi nebo ne.

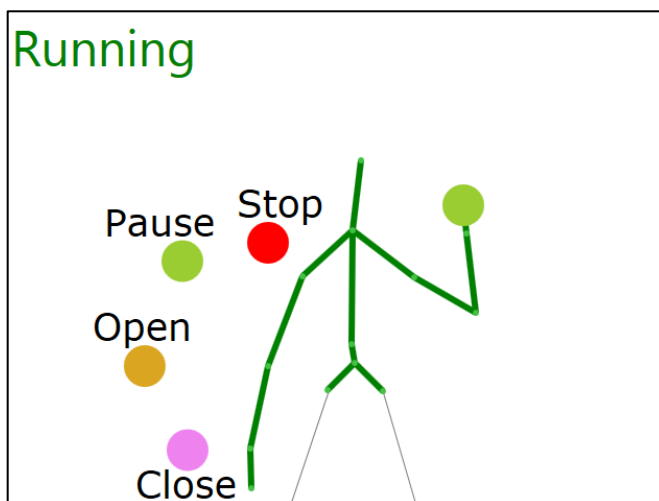
## 5 Praktické testy

V této kapitole je popsáno testování grafického uživatelského rozhraní bez připojeného manipulátoru. Na základě informací zjištěných z testů jsou provedeny patřičné změny. Dále jsou zde uvedeny výsledky z testování vytvořené aplikace pro ovládání manipulátoru. Testování proběhlo na třech operátorech, přičemž je zkoumána schopnost koncového efektoru kopírovat polohu ruky operátora.

### 5.1 Testování bez připojeného manipulátoru

Testování se zúčastnili tři technicky zdatní operátoři, kteří byli seznámeni s navrženým grafickým uživatelským rozhraním. Cílem testování byla simulace pohybu přenesení předmětu a na základě získaných informací odladit návrh grafického uživatelského rozhraní.

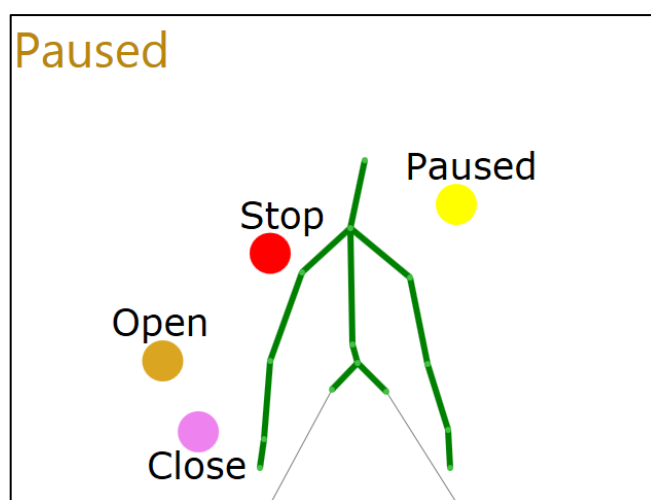
V původním návrhu bylo ovládání mapováno na levou ruku. Jelikož byli všichni testovací operátoři praváci, bylo ovládání pozměněno. Koncový efektor kopíruje pohyb pravé ruky a okolo loktu levé ruky je vykreslováno menu příkazů. Zde by byla možnost rozšíření, které by umožňovalo nastavit ovládání na levou či pravou ruku. Při testování měli operátoři také problém s rozpoznáním menu příkazů. Z tohoto důvodu byly přidány textové popisy, které jednoznačně určují význam jednotlivých příkazů. Zároveň bylo pozměněno rozložení menu, kvůli četnosti použití příkazů. Dalším důvodem pro změnu rozložení bylo nechtěné zastavení aplikace při práci, způsobené nepozorností operátora, který mírně zvedl levou ruku. Z těch to důvodů byl příkaz Stop přesunut na horní pozici a nejpoužívanější příkazy Open a Close byly umístěny do spodní poloviny menu, kde jsou lépe dosažitelné. Výsledné menu lze vidět na Obr 28.



Obr 28. Spuštěná aplikace

Při pozastavení aplikace zůstával v místě poslední známé pozice koncového efektoru vykreslený žlutý kruh jako na Obr 29. Pro znovuspuštění aplikace musel uživatel umístit ruku do tohoto kruhu. Pokud ale operátor aplikaci pozastavil a nechal ruku ve stejné pozici, byla okamžitě detekována kolize a to způsobovalo opětovné spuštění aplikace. Proto bylo pozastavení aplikace pozměněno a kolize s pozicí vykresleného koncového efektoru není detekována ihned po pozastavení aplikace.

V počátečním návrhu bylo při pozastavené aplikaci stále vykreslováno tlačítko Pause. V průběhu testování bylo zjištěno, že tento způsob nedává uživateli dostatečnou zpětnou vazbu o pozastavení aplikace. Proto bylo přistoupeno k řešení, kdy při pozastavení aplikace tlačítko Pause zmizí, jak lze vidět na Obr 29 a operátorovi je takto jasně dáno najevo, že je aplikace pozastavená.



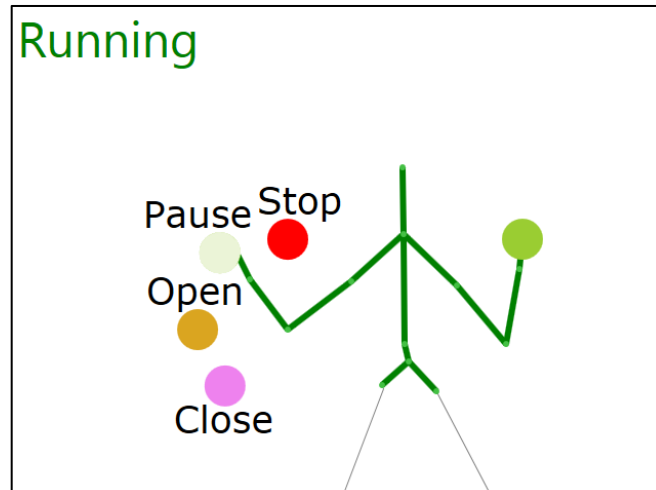
Obr 29. Pozastavená aplikace

Při testování bylo zjištěno, že navržené vykreslování menu příkazů na základě délky vykreslené paže ve 2D nebylo zcela funkční. Délka vykreslené paže byla v určitých pozicích téměř nulová, například při předpažení, a menu příkazů se poté překrývalo. Stejný problém nastal občas po spuštění aplikace, kdy se uživateli menu příkazů vykreslilo přímo v pozici ruky, a aplikace byla zastavena. Z tohoto důvodu bylo přistoupeno k testování předem stanovené vzdálenosti menu příkazů. Tento způsob byl ovšem také nepoužitelný, protože uživatel drobnější postavy nedosáhl na menu příkazů. Řešení spočívalo v kombinaci těchto dvou přístupů. Menu je vykreslováno na základě délky vykreslené paže v grafickém uživatelském rozhraní, přičemž je vykreslování omezeno do předem stanovené minimální úrovně, aby se příkazy nepřekrývaly. Minimální stanovenou úroveň lze vidět na Obr 30.

Na zmíněném obrázku lze také vidět aktivaci příkazu Pause. Změna barvy příkazů je vyvolána aktivací daného tlačítka z důvodu poskytnutí zpětné vazby operátorovi.

Z výsledků testování bylo zjištěno, že menu příkazů v některých případech mírně „problíkává“, což je způsobeno chybnou detekcí délky ruky operátora za-

řízením Kinect. Chyba byla způsobena rozlišovací schopností senzoru a tento problém tedy nebyl odstraněn. Pro vyřešení tohoto problému by bylo nutné použít kvalitnějšího senzoru.



Obr 30. Aktivace tlačítka Pause

## 5.2 Testování vytvořené aplikace a návrh modifikací

Testování se opět zúčastnili tři technicky zdatní operátoři, kterým byla také detailně vysvětlena řešená problematika, a byli obeznámeni s možnostmi navrženého řešení bezdotykového řízení koncového efektoru. Testování probíhalo v laboratoři Q11, viz Obr 31. Během testování byla zkoumána schopnost koncového efektoru kopírovat polohu ruky operátora, zpoždění robotického ramene oproti ruce operátora a četnost výskytu singulárních stavů manipulátoru. Dále bylo vylepšeno grafické uživatelské rozhraní na základě zjištěných informací z testování.



Obr 31. Testování bezdotykového řízení koncového efektoru manipulátoru

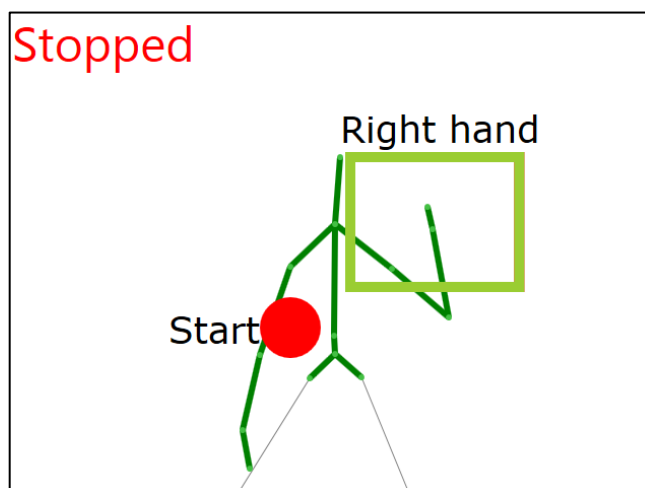


Z testování bylo zjištěno, že je koncový efektor manipulátoru schopen kopírovat polohu ruky s dostatečnou přesností nutnou pro zvednutí a přesunutí předmětu. Avšak koncový efektor stále měnil svou polohu v řádech centimetrů i v případě neměnné polohy ruky operátora. Tento problém byl způsoben přesností senzoru Kinect. Na základě těchto zjištění, bylo testováno zaokrouhlení kloubových souřadnic, které jsou spočtené pomocí úlohy inverzní kinematiky. Kloubové souřadnice ovlivňují přesnost polohování koncového efektoru. Zaokrouhlení na dvě desetinná místa bylo testovacími operátory vyhodnoceno jako neoptimálnější. Tímto zaokrouhlením byla zachována dostatečná přesnost pro manipulaci s předměty a koncový efektor manipulátoru již neměnil polohu při neměnné poloze ruky operátora.

Pro bezdotykové řízení efektoru manipulátoru je ideální vzdálenost cca dva metry od zařízení Kinect. Při této vzdálenosti vidí operátor zřetelně na monitor s grafickým uživatelským rozhraním a zároveň má dostatečný prostor pro manipulaci.

Testování bylo zaměřeno hlavně na pracovní prostor v kladných hodnotách osy  $x^{(r)}$  v  $SS^{(r)}$  (viz Obr 20). V této části pracovního prostoru nedocházelo k singulárním stavům. Naopak v záporných hodnotách osy  $x^{(r)}$ , neboli v pracovním prostoru za manipulátorem, docházelo k singulárním stavům často. Bylo to způsobeno tím, že implementovaná inverzní úloha kinematiky nepočítá s reálným omezením úhlů na motorech. To v praxi znamenalo, že na vstup stávajícího řízení byly poslány hodnoty, které vyvolaly výjimku, a aplikace se ukončila. Při návrhu bylo počítáno pouze s manipulací v prvních dvou kvadrantech  $SS^{(r)}$ , a proto byl problém vyřešen odchyčením výjimky a ponecháním manipulátoru poslední známé poloze.

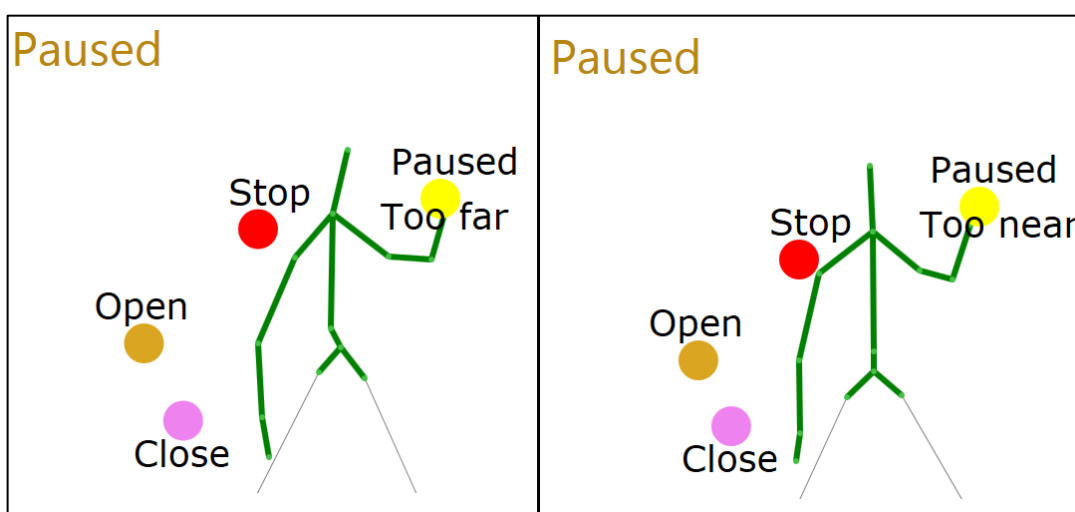
Během testování často nastávala situace, kdy operátor spustil bezdotykové ovládání manipulátoru s pravou rukou u pasu, což způsobilo propojení počátku  $SS^{(o)}$  a  $SS^{(r)}$  podle rovnice (15) v úrovni jeho pasu a operátor nebyl schopen ovládat manipulátor v celém pracovním prostoru. Z tohoto důvodu bylo zvoleno další omezení před spuštěním. Operátor je nucen před zahájením ovládání manipulátoru umístit pravou ruku do vyznačeného obdélníku, který je vykreslován dynamicky podle výšky operátora, viz Obr 32. Pozice a velikost obdélníku byla zvolena na výsledcích z testování a zahrnuje všechny pozice rukou, které operátoři během testování zvolili jako pohodlné a vhodné pro spuštění aplikace.



Obr 32. Zastavená aplikace čekající na spuštění

Aktivace příkazů je založená na kolizi levé ruky operátora s menu příkazů, které je vykreslené ve 2D grafickém uživatelském rozhraní. Při snaze znovu spustit aplikaci z pozastaveného stavu bylo nutné detekovat kolizi i v ose  $z^{(k)}$   $SS^{(k)}$  z důvodu zachování vzájemného posunutí  $SS^{(o)}$  a  $SS^{(r)}$ . Na základě těchto skutečností je zobrazován text, který lze vidět na Obr 33. Text poskytuje operátorovi zpětnou vazbu, jestli stojí příliš daleko nebo příliš blízko vůči předchozí poloze.

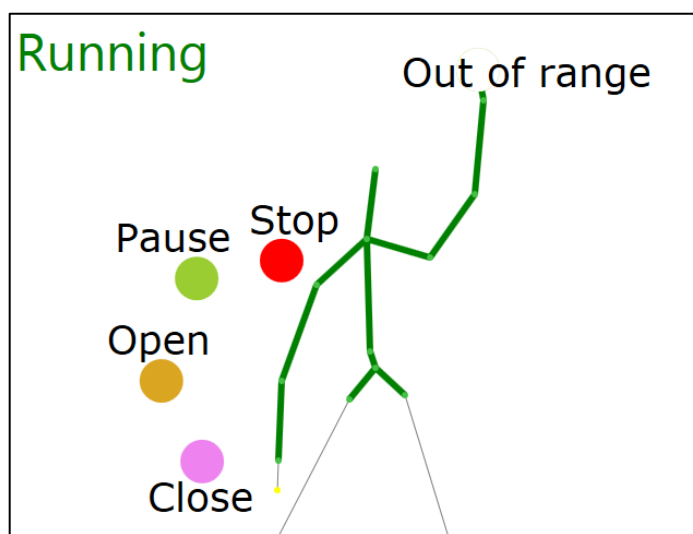
Při otevírání a zavírání efektoru se operátor musel soustředit na polohu obou rukou, což činilo některým operátorům problémy. Z tohoto důvodu bylo potřebné při otevírání a zavírání efektoru aplikaci pozastavit. Nejprve bylo použito již implementované pozastavení aplikace, které ovšem operátora obtěžovalo, jelikož musel aplikaci opět spustit. Z tohoto důvodu bylo implementováno pozastavení, které se aktivuje pouze při otevírání nebo zavírání efektoru a následovně je aplikace opět spuštěna.



Obr 33. Operátor příliš daleko nebo blízko v ose Z souřadného systému operátora

Testována byla také schopnost dosažení celého pracovního prostoru manipulátoru. Během těchto testů samozřejmě nastávala i situace, kdy byla překročena pracovní obálka manipulátoru. Takový bod vede na singulární stav manipulátoru, pro který použitá metoda výpočtu problému inverzní kinematiky není schopna nalézt řešení. Tento neúspěch způsobil vyvolání neošetřené výjimky a aplikace byla ukončena.

Problém byl vyřešen zpracováním výjimky, přičemž manipulátor zůstal v poslední platné poloze a v grafickém uživatelském rozhraní bylo vypsáno, že má operátor ruku mimo dosah manipulátoru. Tento případ lze vidět na Obr 34. V okamžiku, kdy operátor vrátí ruku do pracovního prostoru, pokračuje koncový efektor v kopírování polohy ruky.



Obr 34. Operátor s rukou za hranicí pracovního prostoru manipulátoru

Někteří operátoři měli problém s bezdotykovým ovládním v ose  $x^{(r)}$ . Při pohybu ruky vpřed se totiž koncový efektor manipulátoru od operátora vzdaluje a naopak, což vyplývá ze vzájemné polohy souřadných systémů, viz Obr 20. Tento problém by se dal vyřešit použitím transformačních rovnic (12).

Dále bylo při testování doimplementováno snímání pouze jednoho operátora. Realizována je kombinace snímání nejbližšího a aktivního operátora. Při zastavené nebo pozastavené aplikaci je detekován pouze nejbližší operátor, aby bylo možné odejít z dosahu senzoru a zase zpět nebo aby se mohli operátoři vyměnit. Po spuštění aplikace je detekován pouze aktivní operátor, který ovládá bezdotykové řízení. Pokud aktivní operátor odejde ze zorného pole senzoru nebo je v zákrytu například za jiným člověkem, aplikace se resetuje do výchozího stavu a čeká na spuštění.

### 5.3 Testování koncovými uživateli

Praktické testy byly nejprve provedeny s technicky zdatnými operátory, kteří byli seznámeni s řešenou problematikou, jak bylo uvedeno v předchozí kapitole 5.2. Na základě zjištěných informací z těchto testů byly provedeny poslední modifikace aplikace. Následně bylo přistoupeno k testování koncovými uživateli. Tito uživatelé nebyli seznámeni s omezeními, které byly do aplikace implementovány, a někteří z uživatelů nebyli v dané problematice odborně vzděláni. Přes tyto skutečnosti neměli tito uživatelé problém s ovládním koncového efektoru manipulátoru nebo ovládacím menu aplikace. Uživatelé si také vyzkoušeli komplexní manipulační úlohu, kdy měli za úkol přesunout předem definovaný předmět. Při této úloze dosahovali pomalejších časů, než zkušení operátoři, ale tento fakt byl předpokládán. Důležitějším zjištěním bylo, že byli schopni tuto manipulační úlohu dokončit. Omezení výšky koncového efektoru se ukázalo jako velice důležité. Uživatelé po skončené práci aplikaci nevypínali a odcházeli od spuštěné aplikace s rukou u pasu, aniž si uvědomili, že koncový efektor stále kopíruje polohu této ruky. Díky tomuto omezení tedy nedošlo ke kolizi s podstavou ochranné klece a manipulátor nebyl poškozen.

## 6 Shrnutí

V práci je řešena problematika bezdotykového řízení koncového efektoru manipulátoru v reálném čase, přičemž je pro detekci polohy ruky použito zařízení Kinect.

V práci je použito snímací zařízení Kinect verze 1 s SDK verze 1.8 umožňující snímání skeletonu. Pro ovládání byl zvolen stacionární manipulátor Katana HD300s s pěti stupni volnosti. Použitá metoda řešení inverzní kinematiky je iterativní a je založená na numerickém výpočtu transpozice Jakobiánu. Grafické uživatelské rozhraní je implementováno v jazyce C# s využitím Windows Presentation Foundation.

Kapitola Literární rešerše se zabývá problematikou snímání lidského těla, zvláště pomocí zařízení Kinect. Dále je práce zaměřena na metody řešení inverzní kinematiky a popis stacionárního manipulátoru Katana HD300s.

V kapitole 4.1 jsou popsány dva navržené koncepty polohování manipulátoru. U obou konceptů jsou zmíněny výhody a nevýhody a na základě toho je vybrán koncept, u kterého je počátek  $SS^{(o)}$  v poloze ruky při spuštění aplikace. V kapitole 4.3 jsou nejprve zavedeny souřadné systémy manipulátoru Katana HD300s, snímacího zařízení Kinect a operátora a je popsáno jejich vzájemné posunutí a pootočení. Na základě těchto informací je popsán výpočet požadované polohy koncového efektoru manipulátoru pomocí transformačních matic v kapitole 4.3. Poté je přistoupeno k řešení inverzní úlohy kinematiky, viz kapitola 4.4. Dále je v této kapitole 4.4 řešeno kontinuální snímání polohy dlaně operátora a definování vhodného formátu, ve kterém jsou data ze snímacího zařízení ukládána pro potřeby simulace.

V kapitole 4.5 je řešeno simulační ověření proveditelnosti v prostředí Matlab s využitím vybrané metody řešení úlohy inverzní kinematiky. Následně je provedena offline simulace ovládání manipulátoru pomocí dříve nasnímaných dat. Ze simulace jsou zjištěny a vykresleny informace, na základě kterých je ověřena přesnost výpočtů. Následně je přistoupeno k návrhu řídicí aplikace v jazyce C#, což popisuje kapitola 4.6.

Kapitola 5 se věnuje testování implementovaného řešení bezdotykového řízení koncového efektoru. Nejprve je testováno grafického uživatelského rozhraní bez připojeného manipulátoru. Na základě zjištěných informací jsou provedeny patřičné modifikace. Následně je testována aplikace s připojeným manipulátorem. Testování je provedeno na třech technicky zdatných operátorech a je zkoumána schopnost koncového efektoru kopírovat pozici ruky operátora, zpoždění robotického ramene oproti ruce operátora a četnost výskytu singulárních stavů manipulátoru. Na základě dosažených výsledků ze skutečného provozu s manipulátorem za laboratorních podmínek je provedena další optimalizace za účelem zlepšení dosavadních výsledků. Na závěr je optimalizované řešení testováno koncovými uživateli.

## 7 Závěr a diskuze

### 7.1 Diskuze

Navržené bezdotykové řízení efektoru je určeno jak pro zkušené operátory, tak pro běžné uživatele. Tento fakt byl ověřen v kapitole Praktické testy na komplexní manipulační úloze. Grafické uživatelské rozhraní bylo od počátečního návrhu značně modifikováno, aby poskytovalo uživateli dostatečnou zpětnou vazbu a aby bylo co nejvíce uživatelsky přívětivé. V případě nezkušeného operátora je implementované omezení výšky koncového efektoru nad povrchem dostatečným zabezpečením proti kolizi s podstavou ochranné klece. Tato práce slouží jako „Prove of concept“, že je možné ovládat manipulátor tímto netradičním způsobem. Bezdotykové řízení koncového efektoru manipulátoru je využitelné například v oblasti manipulace s nebezpečnými materiály, se kterými nesmí člověk přijít do styku. Pro průmyslové využití by bylo samozřejmě zapotřebí využít sensorické soustavy s vyšší přesností. Současné řešení je možné využít i na propagačních akcích Provozně ekonomické fakulty Mendelovy univerzity v Brně.

V průběhu návrhu a implementace řešení se také vyskytly určité problémy. Experimenty poukázaly na silné a slabé stránky diskutované v kapitole 5. Na základě testování byla provedena celá řada úprav, které vedly k lepším výsledkům navrženého řešení.

Ukázalo se, že senzor Kinect je největší slabinou řídicího řetězce. Tento senzor je sice cenově snadno dostupný, ale neposkytuje dostatečně přesné informace o poloze operátora. V případě použití senzoru Kinect verze 2 nebo spíše přesné inerciální jednotky, například Xsens MTi-100, by mělo být dosaženo vyšší přesnosti ovládání koncového efektoru. Přesnost polohování koncového efektoru by mohla být také zvýšena optimalizací implementovaného výpočtu úlohy inverzní kinematiky. Zejména současným zvýšením požadované přesnosti numerického výpočtu a snížením časové náročnosti vlastní implementace. Jako perspektivní se jeví výběr uzlových bodů, ve kterých bude počítána inverzní úloha kinematiky a následné plánování trajektorie mezi těmito body.

Dalšího vylepšení by mohlo být dosaženo prostřednictvím zpětné vazby od koncového efektoru, tedy zavedením haptického rozhraní. K tomuto účelu by mohlo být využito infračervených a silových snímačů umístěných v koncovém efektoru manipulátoru, pomocí kterých by mohl být zajištěn optimální úchop předmětu. Využití zmíněného senzoru Leap Motion by mohlo zaručit jemnou manipulaci zajišťující právě úchop předmětu.

Grafické uživatelské rozhraní by mohlo být rozšířeno o možnost výběru ovládání libovolnou rukou, aby bylo umožněno přirozené ovládání pravákům i levákům. Výhoda navrženého GUI spočívá v jeho rozšiřitelnosti v případě potřeby přidání dalšího vysokoúrovňového příkazu.

Výhoda vytvořeného bezdotykové řízení koncového efektoru spočívá v možné přenositelnosti. To znamená, že je možné jej využít i pro jiný model manipulátoru s pěti stupni volnosti. Pokud by měl manipulátor více či méně stupňů volnosti, bylo by nutné pouze upravit parametry metody řešení inverzní kinematiky. V případně použití jiné metody by bylo potřeba nahradit pouze třídu `KatanaInverseKinematics`. Další části aplikace by nebylo nutné modifikovat.

## 7.2 Závěr

Cílem této diplomové práce bylo navrhnout a implementovat bezdotykové řízení koncového efektoru stacionárního manipulátoru Katana HD300s s využitím zařízení Kinect pro detekci polohy operátora.

Cíl práce byl splněn, koncový efektor je schopen kopírovat polohu ruky operátora v reálném čase a ten je tak schopen manipulovat s předměty, což prokázala řada provedených experimentů. Stanovené experimenty posloužili jako prokázání správnosti a použitelnosti navrženého konceptu řešení.

## 8 Seznam literatury a zdrojů

- CORKE, PETER I. *Robotics, vision and control: fundamental algorithms in MATLAB*. Berlin: Springer, 2011, xxiv, 570 p. Springer tracts in advanced robotics, v. 73. ISBN 978-3-642-20143-1.
- ČAPEK, M. *Rozpoznání pohybu uživatele prostřednictvím zpracování obrazu*. Brno, 2013. Dostupné z: <https://is.mendelu.cz/auth/lide/clovek.pl?id=33089;zalozka=7;studium=50328;zp=40658>. Bakalářská práce. Mendelova univerzita v Brně. Vedoucí práce Ing. David Procházka, Ph.D.
- GREPL, R. *Kinematika a dynamika mechatronických systémů*. Vyd. 1. Brno: Akademické nakladatelství CERM, 2007, 158 s. ISBN 978-80-214-3530-8.
- GOŚLIŃSKI, J. *The use of Kinect sensor to control manipulator with electrohydraulic servodrives* [online]. 2013 [cit. 2015-04-01]. Dostupné z: [http://yadda.icm.edu.pl/yadda/element/bwmeta1.element.baztech-article-BSW1-0109-0070/c/Goslinski\\_the.pdf](http://yadda.icm.edu.pl/yadda/element/bwmeta1.element.baztech-article-BSW1-0109-0070/c/Goslinski_the.pdf). Poznań University of Technology.
- HUSSEIN, MOHAMMED A. 2014. *Motion Control of Robot by using Kinect Sensor* [online]. [cit. 2015-05-05]. ISSN 2040-7459. Dostupné z: <http://maxwellsci.com/print/rjaset/v8-1384-1388.pdf>
- KHASSANOV, YERBOLAT. *Inertial motion capture based reference trajectory generation for a mobile manipulator. Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction - HRI '14* [online]. 2014 New York, New York, USA: ACM Press, : 202-203 [cit. 2015-05-05]. DOI: 10.1145/2559636.2559812. ISBN 9781450326582. Dostupné z: <http://dl.acm.org/citation.cfm?doid=2559636.2559812>
- KRUSE, D. *Sensor-Based Dual-Arm Tele-Robotic System*. [online]. 2013 [cit. 2015-04-01]. Dostupné z: <http://www.ecse.rpi.edu/~rjradke/papers/kruse-tase14.pdf>
- LEAP MOTION. *3D Motion and Gesture Control for PC & Mac*. [online]. 2015 [cit. 2015-04-01]. Dostupné z: <https://www.leapmotion.com/product>
- MICROSOFT. *Kinect for Windows features*. [online]. ©2015a [cit. 2015-03-29]. Dostupné z: <http://www.microsoft.com/en-us/kinectforwindows/meetkinect/features.aspx>



- MICROSOFT. *Microsoft Store* [online]. ©2015b [cit. 2015-03-29]. Dostupné z: [http://www.microsoftstore.com/store/msaus/en\\_AU/pdp/Kinect-for-Windows-v2-Sensor/productID.299146900](http://www.microsoftstore.com/store/msaus/en_AU/pdp/Kinect-for-Windows-v2-Sensor/productID.299146900)
- MOSTÝN, V., KRYŠ, V. *Mechatronika průmyslových robotů*. Vyd. 1. Ostrava: Vysoká škola báňská - Technická univerzita, 2012, 1 DVD-ROM. ISBN 978-80-248-2610-3.
- NEURONICS AG. NÁVOD K POUŽITÍ ROBOTŮ KATANA. 2006, 62 s.
- NEURONICS KATANA. *Katana exactech* [online]. 2011 [cit. 2015-03-21]. Dostupné z: <http://katana.exactec.com/>
- PARZYCH, M. *Aspects of Microsoft Kinect sensor application to servomotor control. Bulletin of the Polish Academy of Sciences Technical Sciences* [online]. 2014 62(3): - [cit. 2015-05-05]. DOI: 10.2478/bpasts-2014-0065. ISSN 2300-1917. Dostupné z: <http://www.degruyter.com/view/j/bpasts.2014.62.issue-3/bpasts-2014-0065/bpasts-2014-0065.xml>
- REID, C. *Robotic Arm with Kinect Interface*. [online]. 2013 [cit. 2015-04-01]. Dostupné z: <http://benpyeh.com/2013/12/07/senior-design-robotic-arm-with-kinect-interface-detailed-report-long-post/>
- RUEGG, C, VAN GAEL, J., CUDA, M. *Math.NET Numerics*. [online]. 2015 [cit. 2015-04-28]. Dostupné z: <http://numerics.mathdotnet.com/Matrix.html>
- SHOAI BFAROOQ, M, HASSAN, B., NASEER, M. *Studio Applications and Software Development Kits for Microsoft Kinect: A Survey*. [online]. 2014 [cit. 2015-04-01]. Dostupné z: [http://textroad.com/Old%20version/pdf/JAEBS/J.%20Appl.%20Environ.%20Biol.%20Sci.,%204\(7S\)398-402,%202014.pdf](http://textroad.com/Old%20version/pdf/JAEBS/J.%20Appl.%20Environ.%20Biol.%20Sci.,%204(7S)398-402,%202014.pdf)
- SICILIANO, B., SCIAVICCO, L., VILLANI, L. *Robotics: Modeling, Planning and Control*. 2009 [cit. 2015-04-01]. Springer, ISBN 978-1-84628-641-4.
- SLANÝ, V. *Bezdotykové řízení stacionárního manipulátoru* [online]. Brno, 2014 [cit. 2015-04-01]. Dostupné z: <http://is.mendelu.cz/lide/clovek.pl?id=34680;zalozka=7;studium=63776;zp=46636;lang=cz>. Diplomová práce. Mendelova univerzita v Brně. Vedoucí práce Ing. Vít Ondroušek, Ph.D.
- ŠOLC, F., ŽALUD, L. *Základy robotiky* [online]. 2012 [cit. 2015-03-27]. Dostupné na: [http://www.vutbr.cz/vyzkum-avyvoj/vysledky?vav\\_id=58911#vysledek-58911/hh438998.aspx](http://www.vutbr.cz/vyzkum-avyvoj/vysledky?vav_id=58911#vysledek-58911/hh438998.aspx).

- VINKLER, M. *Využití pohybového snímače Kinectve virtuální realitě*. [online]. 2012 [cit. 2015-03-20]. Bakalářská práce. Masarykova univerzita, Fakulta informatiky. Vedoucí práce Mgr. Jiří Chmelík, Ph.D Dostupné na: [http://is.muni.cz/th/208036/fi\\_m/DP\\_text.pdf](http://is.muni.cz/th/208036/fi_m/DP_text.pdf).
- VYTEČKA, MARCEL. *Šachový automat s využitím manipulátoru Katana*. [online]. 2013 [cit. 2015-03-21]. Diplomová práce. Mendelova univerzita v Brně, Provozně ekonomická fakulta. Vedoucí práce Prof. RNDr. Ing. Jiří Šťastný, Csc. Dostupné z: <https://is.mendelu.cz/auth/lide/clovek.pl?id=30739;zalozka=7;studium=55497>.
- ZUGARA. [online]. 2015 [cit. 2015-04-08]. Dostupné z: <http://zugara.com/how-does-the-kinect-2-compare-to-the-kinect-1>

## 9 Seznam obrázků

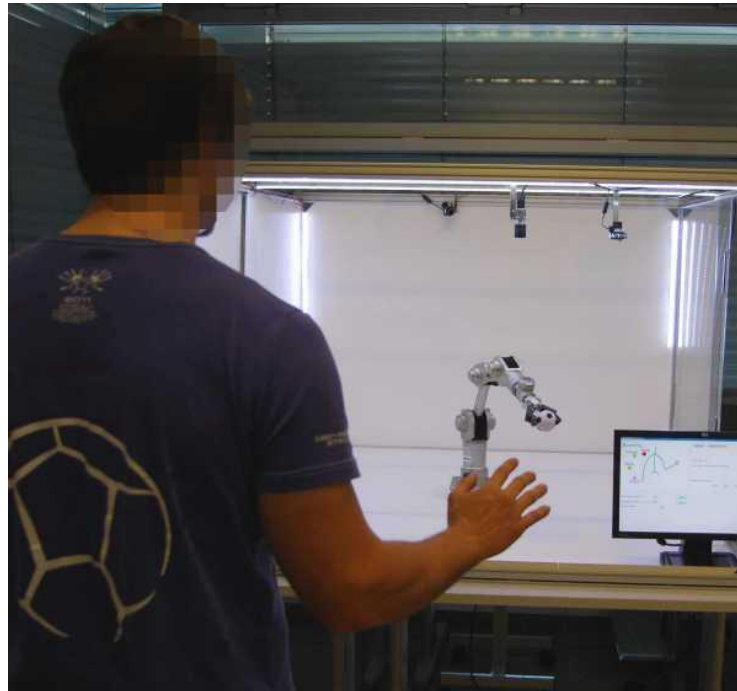
Obr 1.	Popis zařízení Kinect .....	9
Obr 2.	Souřadný systém zařízení Kinect.....	10
Obr 3.	Kinect v2.....	12
Obr 4.	Srovnání hloubkových map Kinectu verze 1 (nalevo) a verze 2 (napravo)....	13
Obr 5.	3D model rukou .....	13
Obr 6.	Katana 6M180.....	14
Obr 7.	Univerzální uchopovač manipulátorů Katana.....	15
Obr 8.	Souřadný systém manipulátoru.....	16
Obr 9.	Základní rozdělení metod řešení inverzní úlohy kinematiky.....	18
Obr 10.	Startovací gesto .....	21
Obr 11.	Ovládání motoru číslo 1 .....	22
Obr 12.	Testování aplikace .....	23
Obr 13.	Schéma řízení.....	23
Obr 14.	Robotická ruka a ruka operátora .....	24
Obr 15.	Komplexní interaktivní systém mezi operátorem a robotem.....	25
Obr 16.	Yaskawa Motoman SDA10 .....	26
Obr 17.	Operátor, grafická reprezentace snímaného skeletonu a simulace robotu SDA10	26
Obr 18.	Schéma řídicího řetězce .....	30
Obr 19.	Výchozí poloha manipulátoru v prostředí Matlab a ve skutečnosti.....	31
Obr 20.	Srovnání souřadných systémů (zleva: operátor, Kinect a manipulátor) .....	32
Obr 21.	Skeleton Basics-WPF v jazyce C#.....	36
Obr 22.	Vykreslená trajektorie ruky .....	37
Obr 23.	Trajektorie ruky v souřadném systému operátora, senzoru a manipulátoru ...	38
Obr 24.	Model manipulátoru Katana HD300s .....	40
Obr 25.	Schéma funkcionality tříd .....	42
Obr 26.	Uživatelského rozhraní založené na úrovních .....	44
Obr 27.	Startovní obrazovka aplikace (nalevo) a spuštěná aplikace (napravo) .....	45
Obr 28.	Spuštěná aplikace.....	46
Obr 29.	Pozastavená aplikace .....	47
Obr 30.	Aktivace tlačítka Pause .....	48
Obr 31.	Testování bezdotykového řízení koncového efektoru manipulátoru .....	48
Obr 32.	Zastavená aplikace čekající na spuštění.....	50
Obr 33.	Operátor příliš daleko nebo blízko v ose Z souřadného systému operátora ...	50
Obr 34.	Operátor s rukou za hranicí pracovního prostoru manipulátoru .....	51
Obr 35.	Testování pohybu ve směru osy $z^{(r)}$ v $SS^{(r)}$ .....	62
Obr 36.	Testování pohybu ve směru osy $z^{(r)}$ v $SS^{(r)}$ .....	62
Obr 37.	Testování pohybu ve směru osy $x^{(r)}$ v $SS^{(r)}$ .....	63
Obr 38.	Testování pohybu ve směru osy $x^{(r)}$ v $SS^{(r)}$ .....	63

## 10 Seznam tabulek

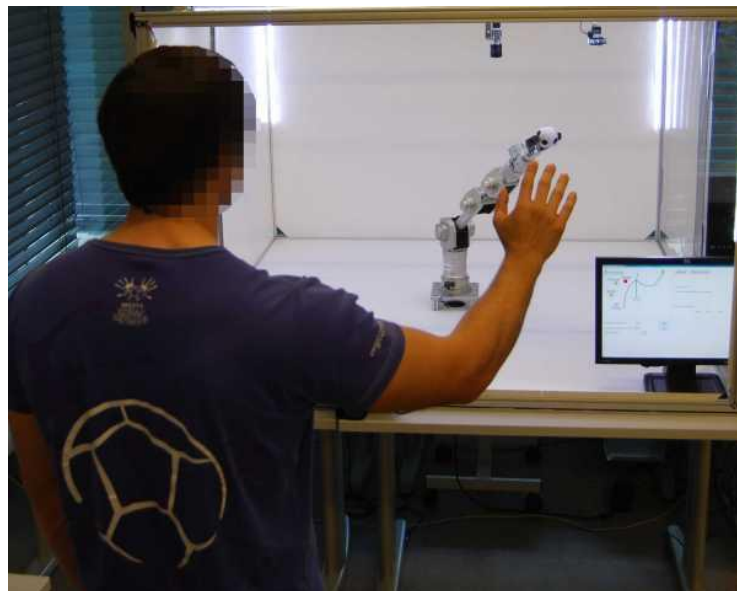
Tab. 1	Srovnání Kinect v1 a Kinect v2.....	12
Tab. 2	Technické parametry manipulátoru .....	15
Tab. 3	Rozsah jednotlivých motorů .....	17
Tab. 4	Výchozí poloha manipulátoru.....	31
Tab. 5	Vybrané atributy a metody třídy KatanaInverseKinematics.....	43
Tab. 6	Vybrané metody třídy MainWindow .....	45

# **Přílohy**

## A Testování bezdotykového řízení



Obr 35. Testování pohybu ve směru osy  $z^{(r)}$  v  $SS^{(r)}$



Obr 36. Testování pohybu ve směru osy  $z^{(r)}$  v  $SS^{(r)}$



Obr 37. Testování pohybu ve směru osy  $x^{(r)}$  v  $SS^{(r)}$



Obr 38. Testování pohybu ve směru osy  $x^{(r)}$  v  $SS^{(r)}$

## **B Obsah přiloženého CD**

Přiložené CD obsahuje následující soubory:

- Zdrojový kód aplikace pro bezdotykové řízení efektoru v C#
- Knihovny potřebné ke spuštění aplikace
- Zdrojový kód simulací v prostředí Matlab