

Bakalářská práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra kybernetiky

## Ovládání pohybu průmyslového robotu nástroji pro virtuální realitu

**Dmitrij Sojma**

Vedoucí: Ing. Vladimír Smutný, Ph.D.  
Obor: Kybernetika a Robotika  
Květen 2020



## Poděkování

Rád bych poděkoval panu Ing. Vladimíru Smutnému, Ph.D za věnovaný čas a veškerou pomoc, která umožnila vznik této práce. Dále bych chtěl poděkovat své rodině a dalším blízkým za podporu studia na této škole.

## Prohlášení

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

V Praze, 22. května 2020

## Abstrakt

Tato práce se zabývá využitím systému pro virtuální realitu HTC Vive k ovládní průmyslového robotu. Je popsán princip funkce systému HTC Vive, data z tohoto systému jsou použita v prostředí ROS, kde jsou dále zpracovávána nově vytvořenými aplikacemi, které umožňují ovládat robot s využitím Vive ovladače a jeho tlačítek, klady a zápory použití tohoto systému jsou zvýrazněny. Práce se také věnuje studiu závislosti singulárních čísel Jakobiho matice robotu KUKA iiwa LBR 7 na úhlu, který odpovídá nadbytečnému stupni volnosti tohoto robotu, a návrhem optimalizace jeho hodnoty.

**Klíčová slova:** vzdálené ovládní, robot, HTC VIVE, světelný maják

**Vedoucí:** Ing. Vladimír Smutný, Ph.D.  
Český institut informatiky, robotiky a kybernetiky, České vysoké učení technické v Praze

## Abstract

This thesis explores the possible use of virtual reality system HTC Vive for control of an industrial robot. The principle of operation of the HTC Vive system is described, system's raw data are published to the ROS environment, where they are further processed by newly created applications to allow for robot manipulation via Vive controller and its buttons, pros and cons of this system are highlighted. This work also studies how singular values of Jacobi matrix of the KUKA iiwa LBR 7 robot change with respect to an angle, which corresponds to the robot's redundant degree of freedom, and an optimization method is proposed.

**Keywords:** remote control, robot, HTC VIVE, lighthouse

**Title translation:** Industrial Robot Control Using Virtual Reality Tools

## Obsah

<b>Seznam použitých zkratk</b>	<b>1</b>	<b>4 Zpracování dat ze systému HTC Vive</b>	<b>15</b>
<b>1 Úvod</b>	<b>3</b>	4.1 Informace poskytované rozhraním Libsurvive a jejich sdílení pomocí ROS .....	15
<b>2 Současné technologie vzdáleného ovládání a jejich využití</b>	<b>5</b>	4.1.1 Data o pohybu .....	15
2.0.1 Bridge transported Dual arm ServoManipulator (BDSM) .....	5	4.1.2 Sběr informací o tlačítkách ..	16
2.1 Roboticky asistované operace .....	6	4.2 ViveData .....	18
2.2 Ovládání robotu pomocí chytrých zařízení .....	7	4.2.1 Zpracování dat o poloze Trackeru .....	19
<b>3 Informace o použitých systémech</b>	<b>9</b>	4.2.2 Zpracování dat o poloze ovladače .....	20
3.1 ROS .....	9	<b>5 Ovládací aplikace</b>	<b>23</b>
3.2 KUKA LBR iiwa 7 .....	9	5.1 Studium vlastností Jacobiho matice v závislosti na úhlu $\delta$ .....	23
3.3 HTC Vive .....	10	5.2 Optimalizace úhlu $\delta$ .....	27
3.3.1 Popis komponent .....	10	5.3 Funkce ovládací aplikace .....	29
3.3.2 Princip funkce .....	12	<b>6 Testování a možná vylepšení</b>	<b>31</b>
3.3.3 Konfigurace užívaná při této práci .....	13	6.1 Testování funkčnosti .....	31
3.4 Libsurvive .....	13	6.1.1 Optimalizace úhlu $\delta$ .....	31
		6.1.2 Užití systému HTC Vive pro ovládání robotu .....	33

6.2	Náměty k možným vylepšením .	34
<b>7</b>	<b>Závěr</b>	<b>35</b>
	<b>Literatura</b>	<b>37</b>
<b>A</b>	<b>Seznam příloh</b>	<b>41</b>
<b>B</b>	<b>Zadání práce</b>	<b>43</b>

## Obrázky

2.1 Schema funkce BDSM[15] . . . . .	6	5.3 Závislost na $\delta$ ke konci pracovní obálky. . . . .	26
2.2 Obraz z pohledu chirurga a ovládací mechanismus kleští[14]. . . . .	7	5.4 Příklad závislosti pro neproblémovou polohu chapadla. . . . .	26
3.1 Sada HTC Vive obsahující 2 ovladače, 2 základnové stanice a náhlavní soupravu. . . . .	10	5.5 Příklad rozdílu ve výskytu lokálních maxim. . . . .	27
3.2 Popis ovladače HTC Vive[10]. . . . .	11		
3.3 Zařízení Vive Tracker[13]. . . . .	11		
3.4 Princip funkce majáku pro určování polohy[21]. . . . .	12		
4.1 Osy, jejich hodnoty a zvolené rozložení tlačítka Trackpad. . . . .	17		
4.2 Propojení a komunikace jednotlivých podsystémů. . . . .	18		
4.3 Orientace souřadnicového systému Trackeru. . . . .	19		
4.4 Schéma vztahů mezi souřadnicovými systémy. . . . .	20		
5.1 Poloha robotu v závislosti na úhlu $\delta$ [18]. . . . .	24		
5.2 Vnitřní singularita pro $\delta = 0$ rad. . . . .	25		

## Tabulky

3.1 Význam značek k Obrázku 3.2[10]. 11

4.1 Přehled hodnot umožňujících  
identifikaci tlačítka a příslušné  
události pro ovladač HTC Vive. . . 16

4.2 Význam tlačítka pro Tracker. . . . 17





## Seznam použitých zkratk

ROS	Robot Operating System
IMU	Inertial Measurement Unit
FRI	Fast Robot Interface
LBR	Leichtbauroboter
iiwa	intelligent industrial work assistant
SVD	Singular Value Decomposition





# Kapitola 1

## Úvod

Roboty jsou využívány k provádění repetitivních prací v mnoha oblastech průmyslu. Při uvádění robotu do provozu ale bývá potřeba navést robot do konkrétní polohy například pro vytvoření záchytných bodů výsledné trajektorie, která umožňuje plnit úkol, pro který byl robot zvolen. U malých robotů, pokud to okolnosti dovolují, je pro polohování robotu možné využít silové poddajnosti. Nicméně tento princip nelze využít pro velké stroje či roboty, které pracují s velkými silami například při manipulaci s těžkými díly. V takových případech se často využívá ovládání jednotlivých kloubů robotu pomocí ovládacího panelu, případně lze ovládacím panelem omezeně měnit kartézskou polohu referenčního bodu chapadla robotu. Ani jeden z těchto přístupů není ideální, jelikož ke změně polohy robotu v každém čase může docházet pouze v jednom z předem daných směrů a zvolení pohybu k dosažení cílené polohy bývá velmi neintuitivní.

Hlavním cílem této práce je využít data a hardware z populárního systému pro virtuální realitu HTC Vive k ovládání polohy robotu způsobem, který je jednoduchý, intuitivní a vhodný k budoucímu využití v průmyslu. Práce se nejprve zabývá principem systému HTC Vive, dále jsou zkoumána poskytnutá data a ta jsou využita k implementaci struktur pro jejich následné použití v ovládací aplikaci robotu. Vedlejším cílem je návrh algoritmu k optimalizaci parametru, který odpovídá nadbytečnému stupni volnosti robotu KUKA iiwa LBR 7, kde motivací je využití tohoto parametru pro zlepšení manipulovatelnosti zmíněného robotu.

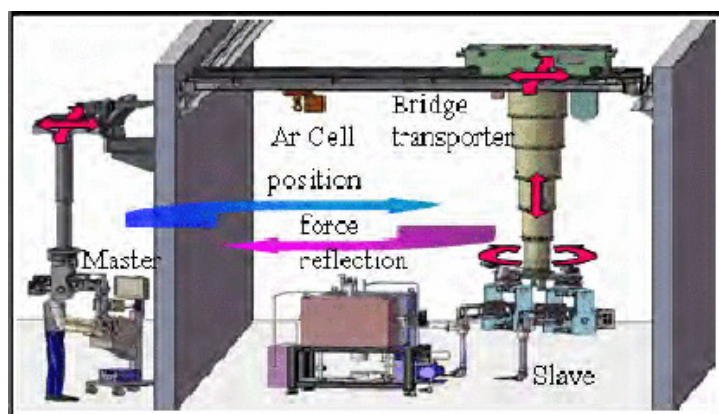


## Kapitola 2

### Současné technologie vzdáleného ovládání a jejich využití

#### ■ 2.0.1 Bridge transported Dual arm ServoManipulator (BDSM)

Tento systém byl vyvinut za účelem údržby a operací s náčiním pro pyroprocesy či jiné práce uvnitř buněk s toxickým plynem, skládá se z mostového transporteru se 4 stupni volnosti, který tvoří vozík umístěný na kolejnicích umožňujících pohyb po celé šířce i délce místnosti a pětičlenná teleskopická trubice zajišťující pohyb ve vertikálním směru. Systém dále tvoří dvě dvojice robotických ramen, ramena uvnitř místnosti jsou umístěna na spodní části transporteru pomocí něhož je lze umístit jakkoliv s výjimkou maximální výšky, která je dána maximálním zatáhnutím teleskopické trubice transporteru, tyto roboty jsou ovládány sériovými manipulátory vně místnosti, s nimiž pohybuje operátor[15], [16] dle Obrázku 2.1. Operátor se řídí jednak vizuálními podněty, které jsou zajišťovány 8 kamerami či přímým pohledem skrze ochranné okno, ale také oboustrannou silovou vazbou, k ovládní transporteru slouží ovládací konzole[16]. Poměr přenášených sil lze měnit od 1:1 do 5:1, dále lze měnit také poměr vykonaných pohybů[20].



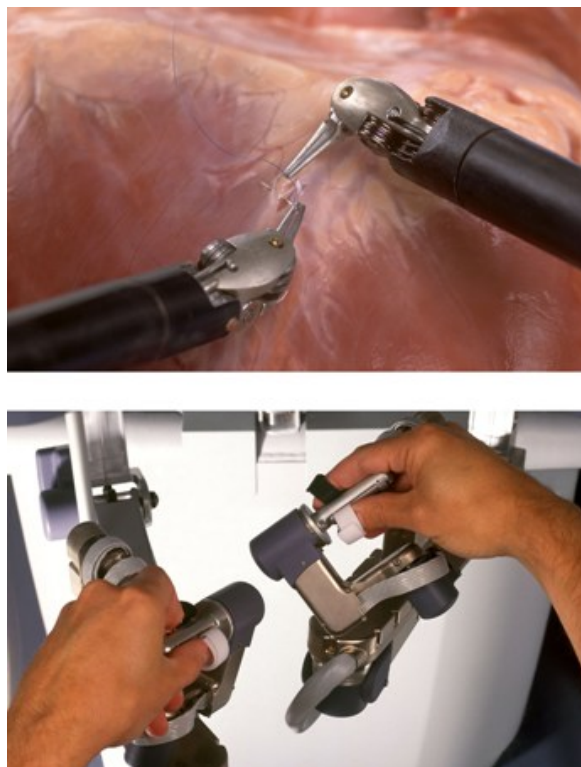
Obrázek 2.1: Schema funkce BDSM[15]

Vzdálené ovládání robotů je také používáno při práci se zbytky jaderného paliva či dalšího radioaktivního odpadu, operátor v takových případech pozoruje průběh práce uvnitř buňky skrze ochranné okno a k ovládání jsou používány joysticky nebo speciálně vyvinuté ovladače, které často pracují s oboustranným přenosem sil[15]. V neposlední řadě byla pro podobné účely vyvinuta francouzská softwarová platforma TAO2000 pro telemanipulaci pomocí silové zpětné vazby, která funguje na obecném přístupu master-slave. Obousměrné propojení funguje na principu simulované šestidimenzionální pružiny a tlumiče [7]. Tato platforma umožňuje použití určitých skupin robotů a to jak zakoupených industriálních robotů tak i speciálně navržených, umožňuje práci s více roboty najednou, změnu pracovních módů a mnoho dalších možností[24].

## 2.1 Roboticky asistované operace

Teleoperace je využívána také v medicíně, nejznámějším zástupcem v kategorii robotické chirurgie je systém Da Vinci od společnosti Intuitive Surgical, který se skládá ze dvou mobilních stanic, kde jedna z nich poskytuje operujícímu lékaři obraz a umožňuje mu ovládat druhou stanicí, která slouží k vykonání úkonů na pacientovi. Chirurg má k dispozici 5 pedálů, které umožňují přepnout na ovládání polohy kamery, ostření obrazu, práci s některými nástroji nebo pohyb některé ze stanic[26]. V [6] je uveden technický popis tohoto systému, k dispozici je 2D i 3D obraz, kleště jsou ovládány konstrukcí, která umožňuje mapovat pohyb zápěstí na pohyb kleští a ukazováček a palec odpovídají jejich čelistím. Pojízdna stanice pacienta se skládá z 3 případně 4 robotických ramen, kde vždy jedno z nich slouží k manipulaci s endoskopickou kamerou a zbylá ramena obsahují kolejnice pomocí nichž lze zasouvat či

vysouvat nástroje z pacientova těla. Každé rameno obsahuje set pasivních kloubů sloužících k nastavení úhlu průniku ramen do pacienta, narozdíl od klasických laparoskopických operací se nachází celý nástroj uvnitř pacienta a nedochází tak k odebrání stupně volnosti. Systém filtruje třes rukou a umožňuje škálování prováděných pohybů, největší nevýhodou je chybějící haptická odezva, čímž je znemožněno rozpoznávání rozdílů v tkáni, ale také šití a vázání uzlů z některých materiálů. Studie ukazují, že se jedná o robustní řešení s nízkým výskytem kritických selhání [6].



**Obrázek 2.2:** Obrázek z pohledu chirurga a ovládací mechanismus kleští[14].

## 2.2 Ovládání robotu pomocí chytrých zařízení

Ovládání robotu i jeho učení pomocí programování záchytných bodů jeho trajektorie nebývá jednoduchou záležitostí ani pro zkušenější operátory, při řízení robotu v kartézském souřadnicovém systému je potřeba si pamatovat orientaci jeho os a zároveň je třeba si uvědomovat, jak jsou pohyby joysticku mapovány na pohyby v tomto souřadnicovém systému. Jednou z možností pro usnadnění ovládání robotu se jeví použití chytrých zařízení, jako jsou telefony a tablety, v [25] je ukázáno využití měření náklonu telefonu k polohování robotu do specifický poloh k učení trajektorie, dále je tato metoda porovnána





## Kapitola 3

### Informace o použitých systémech

#### 3.1 ROS

ROS je open-source nástroj připomínající operační systém, vytvořený za účelem snazší tvorby aplikací pro práci s roboty, poskytuje hojné množství univerzálních nástrojů a knihoven, které umožňují tvorbu a spouštění aplikací napříč vícero počítačů[23]. Jednotlivé programy mezi sebou mohou, mimo jiné, komunikovat pomocí publikování a odebírání zpráv dle zvoleného tématu, tzv. rostopic. Jedním z nástrojů systému ROS je balíček pro 3D vizualizaci s názvem Rviz, který byl při této práci hojně využíván pro simulaci práce s robotem.

#### 3.2 KUKA LBR iiwa 7

Robot LBR iiwa 7 je sedmiosý kolaborativní robot lehké konstrukce od společnosti KUKA s nosností 7 kg, dosahem 800 mm a opakovatelností přibližně 1 mm[19]. Řízení robotu z externího programu je umožněno rozhraním FRI (Fast Robot Interface), které lze také připojit k systému ROS.

### 3.3 HTC Vive

HTC Vive je produkt od společnosti Valve zpřístupňující virtuální realitu, na Obrázku 3.1 lze vidět hlavní komponenty, ke kterým patří náhlavní souprava poskytující obraz a zvuk, 2 ovladače a 2 základnové stanice umožňující sledování pohybu. Dle [22] se jedná o levný systém s vysokou přesností a nízkou latencí, nicméně podle [2] díky algoritmům, které tento systém používá, není jeho užití vhodné pro situace, kdy velmi vysoká přesnost a opakovatelnost pohybů je zásadní, dále je ukázáno, že tento problém lze řešit použitím jiných algoritmů, což je demonstrováno také v [8].



**Obrázek 3.1:** Sada HTC Vive obsahující 2 ovladače, 2 základnové stanice a náhlavní soupravu.

#### 3.3.1 Popis komponent

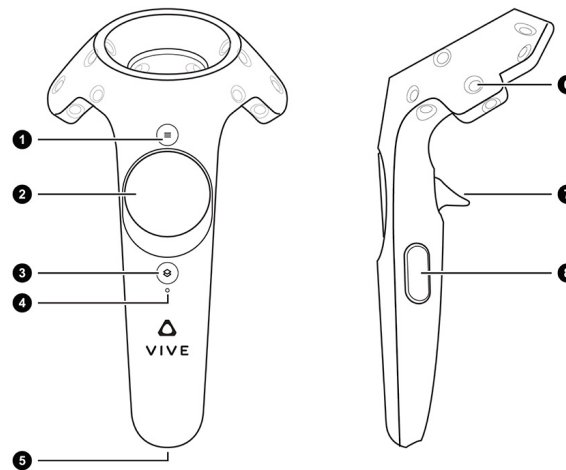
Náhlavní souprava poskytuje obraz pomocí dvou AMOLED displejů, každý s rozlišením 1080 x 1200 pixelů, obnovovací frekvencí 90 Hz a celkovým zorným polem o velikosti 110 stupňů, obsahuje senzory citlivé na infračervené světlo, akcelerometr, gyroskop a senzor přiblížení[12] sloužící k ochraně před naražením do případných okolních překážek. Obraz je do počítače přiváděn rozhraním HDMI a ostatní data pomocí USB.

Každý ovladač obsahuje 24 světelných senzorů, akcelerometr a gyroskop, dále je ovladač vybaven pěti tlačítky, z nichž dvě jsou multifunkční. Ovladač

také disponuje haptickou odezvou pro reálnější požitek z virtuální reality. Data z ovladače jsou při bezdrátovém provozu posílána přes Bluetooth do náhlavní soupravy, alternativně lze ovladač připojit přes rozhraní USB nebo dokoupit přijímač pro komunikaci přes Bluetooth. Popis ovladače je uveden v Obrázku 3.2.

1	Menu button
2	Trackpad
3	System button
4	Status light
5	Micro-USB port
6	Tracking sensor
7	Trigger
8	Grip button

**Tabulka 3.1:** Význam značek k Obrázku 3.2[10].



**Obrázek 3.2:** Popis ovladače HTC Vive[10].

Základnové stanice slouží k určování polohy zařízení pomocí světla principem popsaným v části 3.3.2. K základní sadě je možné přikoupit další zařízení, jedním z nich je Vive Tracker zobrazený na Obrázku 3.3. Jedná se o zařízení připomínající vrchní část ovladače, obsahuje gyroskop, akcelerometr, 23 senzorů a jedno tlačítko. Pro šíření dat využívá Bluetooth připojení ke speciálnímu přijímači a vysílači, který lze k počítači připojit přes rozhraní USB.

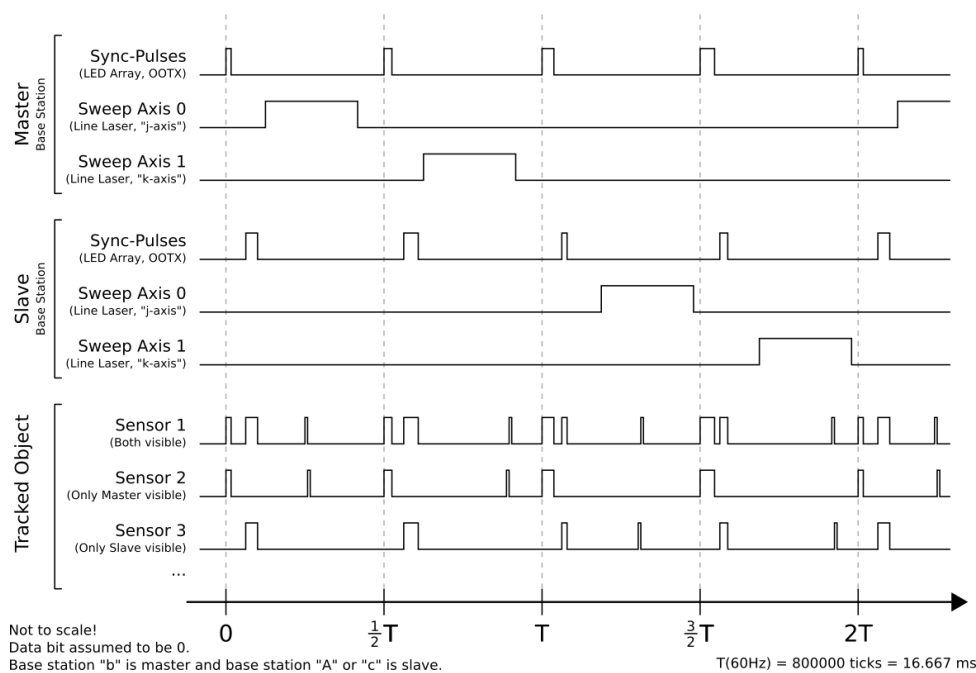


**Obrázek 3.3:** Zařízení Vive Tracker[13].

### 3.3.2 Princip funkce

Základnové stanice fungují jako světelné majáky využívající odrazu infračerveného světla od rotujících zrcadel uvnitř základnové stanice k určení polohy pomocí světelných senzorů umístěných na dílcích zařízeních[27]. Tyto stanice mají 3 provozní kanály, mezi kterými lze cyklit pomocí tlačítka na zadní straně základnové stanice. Provozní kanál je třeba volit dle způsobu synchronizace sekundární stanice, kdy pro optickou synchronizaci je použit kanál C, pro synchronizaci připojením kabelu je potřeba zvolit kanál A a kanál B označuje primární základnovou stanici[21].

Každý cyklus určení polohy začíná vysláním dvou všesměrových synchronizačních světelných pulzů za účelem časové synchronizace a pro přenesení informací o aktuálním stavu vysílajících majáků[21]. Sledovaný prostor je dále zasáhnut světelnou rovinou pohybující se v horizontálním či vertikálním směru, pokračuje se opětovným vysláním synchronizačního pulzu a následným průchodem světelné roviny ve druhém směru, nakonec je tento cyklus opakován pro druhý maják [8], princip funkce je také nastíněn Obrázkem 3.4



Obrázek 3.4: Princip funkce majáku pro určování polohy[21].

Výsledkem funkce majáků je, že senzory sledovaných objektů jsou zasahovány s frekvencí 120 Hz, nicméně každý světelný zásah slouží k určení pouze jedné ze dvou souřadnic senzoru, proto jsou úplná data o poloze k

dispozici pouze s frekvencí 60 Hz[17]. Polohu sledovaného objektu lze díky konstantní frekvenci sestavit z časového rozdílu zásahů jednotlivých světelných senzorů[27], tento způsob umožňuje měřit absolutní polohu s relativně nízkou frekvencí, toho je využito společně s vysokofrekvenčními daty o pohybu hlášených od IMU k určování relativně přesné polohy i s vyšší frekvencí[1], dle [17] je výsledná poloha ovladačů k dispozici s frekvencí 250 Hz a poloha pro náhlavní soupravu s frekvencí 225 Hz.

### ■ 3.3.3 Konfigurace užívaná při této práci

Pro účely této práce byl systém HTC Vive používán v konfiguraci se 2 majáky s optickou synchronizací vždy ve vzdálenosti menší než maximální doporučená hodnota 5 m[11]. Byl používán jeden ovladač a jeden Tracker, oba byly připojeny převážně bezdrátově pomocí Bluetooth přijímače, výjimečně přímo pomocí USB. Náhlavní souprava využívána nebyla, jelikož je kvůli bezpečnosti nutná vizuální vazba s fyzickým robotem a tato souprava je neprůhledná.

## ■ 3.4 Libsurvive

Libsurvive[3] je open-source nástroj určený pro práci s objekty, jejichž poloha je sledována pomocí světelných majáků, jako je tomu u systému HTC Vive. Tento nástroj je výsledek zpětného inženýrství právě systému HTC Vive, nicméně se stále jedná o nedokončený projekt. Výhodou tohoto nástroje je možnost práce s hrubými daty, použití komunitních algoritmů pro práci s nimi, možnost sledování polohy vlastního zařízení a v neposlední řadě jeho nenáročnost a malá velikost umožňují jeho spuštění i na mikročipech. Tento nástroj je používán k určování polohy i v literatuře, příkladem jsou [1] a [2]. Tato práce pracuje s GIT verzí 2457902b.



## Kapitola 4

### Zpracování dat ze systému HTC Vive

#### 4.1 Informace poskytované rozhraním Libsurvive a jejich sdílení pomocí ROS

Tato kapitola popisuje informace poskytované rozhraním Libsurvive, dále jsou tyto informace použity pro přenos vybraných dat do rozhraní ROS pro účely ovládání robotu programem s názvem Survive Publisher.

##### 4.1.1 Data o pohybu

Libsurvive poskytuje data o poloze i rychlosti ve stejnojmenném souřadnicovém systému, nicméně pro další práci bude na tento souřadnicový systém odkazováno pomocí jména Vive, značený  $O_V$ . Poloha je reprezentována strukturou, jejíž součástí je polohový vektor  $\mathbf{r} = [r_x, r_y, r_z]^T$  a kvaternion  $\mathbf{q} = [q_w, q_x, q_y, q_z]^T$  určující orientaci. Data o rychlosti jsou reprezentována vektorem translační rychlosti  $\mathbf{t} = [t_x, t_y, t_z]^T$  a pro rotační rychlost je použita reprezentace osa-úhel  $\mathbf{v} = [v_x, v_y, v_z, v_{angle}]^T$ .

Frekvence poskytovaných dat je závislá na zvoleném algoritmu zpracovávajícím hrubá data. K dispozici je několikero takových algoritmů, v této práci je používán výchozí algoritmus s názvem `poser_mpf`, který využívá Levenberg–Marquardtova algoritmu k minimalizaci chyby projekce polohy

do roviny[3] od obou majáků. Tento algoritmus umožňuje získávat data s frekvencí přibližně 367 Hz.

#### 4.1.2 Sběr informací o tlačítkách

Libsurvive poskytuje funkci, umožňující detekci událostí, jako je například stisk tlačítka. Obdržená datová struktura obsahuje následující položky.

- Objekt - struktura s informacemi o zdrojovém objektu
- Typ události
- ID tlačítka
- ID 1. osy a ID 2. osy
- Hodnota 1. osy a hodnota 2. osy

Význam hodnot pro tyto položky nebyl zdokumentován, proto uvádím jejich význam pro ovladač HTC Vive v Tabulce 4.1, kde jména tlačítek[10] odpovídají Obrázku 3.2 a Tabulce 3.1.

Jméno tlačítka	Událost	ID tlačítka*	Typ/y události/í	ID 1. osy	Hodnota 1. osy	ID 2. osy	Hodnota 2. osy
Menu	Stisk / Uvolnění	5 (12)	1 / 2	0	0	0	0
Trackpad	Dotyk	0	3	2	-32768 až 32767	3	-32768 až 32767
	Začátek / Konec dotyku	1**	1 / 2	0	0	0	0
	Úplný stisk / Uvolnění	2***	1 / 2	0	0	0	0
System	Stisk / Uvolnění	3 (13)	1 / 2	0	0	0	0
Trigger	Míra stisknutí	0	3	1	0 až 32640	0	0
	Úplný stisk / Uvolnění	24	1 / 2	0	0	0	0
Grip	Stisk / Uvolnění	4 (2)	1 / 2	0	0	0	0

\* ID tlačítka se v některých případech liší v závislosti na způsobu připojení. V případech, kdy je ID pro drátové a bezdrátové připojení rozdílné, je hodnota pro drátové připojení uvedena v závorce.

\*\* ID tlačítka pro drátové připojení neexistuje, tudíž ani příslušné události. Událost typu „Konec dotyku“ je pro drátové připojení realizována událostí v předchozím řádku tabulky, kdy obě osy nabývají hodnoty 0, událost typu „Začátek dotyku“ pro drátové připojení není realizována žádným způsobem.

\*\*\* ID tlačítka pro drátové připojení neexistuje, tudíž ani příslušné události. Tyto události nejsou pro drátové připojení realizovány jiným způsobem.

**Tabulka 4.1:** Přehled hodnot umožňujících identifikaci tlačítka a příslušné události pro ovladač HTC Vive.



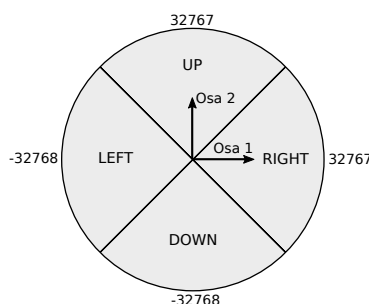
Pro úplnost je význam jediného tlačítka Trackeru uveden v Tabulce 4.2.

Jméno tlačítka	Událost	ID tlačítka*	Typ/y události/í	ID 1. osy	Hodnota 1. osy	ID 2. osy	Hodnota 2. osy
Power	Stisk / Uvolnění	3 (13)	1 / 2	0	0	0	0

\* Hodnota v závorce platí v případě drátového připojení.

**Tabulka 4.2:** Význam tlačítka pro Tracker.

U tlačítka Trackpad odpovídá při orientaci jako v Obrázku 3.2 první ose osa horizontální a druhé ose osa vertikální, hodnoty se řídí dle Obrázku 4.1.



**Obrázek 4.1:** Osy, jejich hodnoty a zvolené rozložení tlačítka Trackpad.

Pro další účely jsem navrhl vlastní soubor pojmenovaných událostí, který se skládá z 15 událostí dle Tabulky 4.1, 2 událostí dle Tabulky 4.2 a 6 nových událostí uvedených níže. Nové události spojené se stiskem tlačítka Trackpad se řídí dle rozdělení na Obrázku 4.1, událost HAPTIC slouží k přijetí požadavku na zavibrování ovladače.

Nové události:

- TRACKPAD\_UP - stisk tlačítka Trackpad v horní části
- TRACKPAD\_DOWN - stisk tlačítka Trackpad ve spodní části
- TRACKPAD\_LEFT - stisk tlačítka Trackpad v levé části
- TRACKPAD\_RIGHT - stisk tlačítka Trackpad v pravé části
- HAPTIC - požadavek na haptickou odezvu ovladače
- ERROR - chybová hlášení

Dále jsem nadefinoval nový typ zprávy s názvem ViveEvents pro odesílání či přijímání těchto událostí v prostředí ROS.

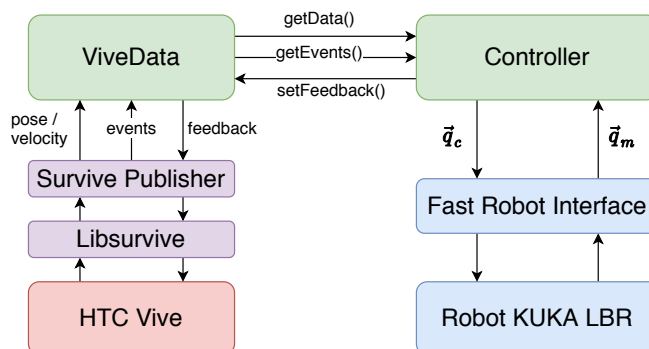
Struktura zprávy ViveEvents:

- Header - hlavička obsahující časovou značku, sekvenční číslo a původ
- Event\_type - typ události např. TRACKPAD\_UP
- Data - pole variabilní délky obsahující případná další data

Součástí nástrojů poskytovaných Libsurvive je program s názvem Survive Publisher, který zvládá posílat data o poloze sledovaných objektů do systému ROS, praktickým výsledkem sekce 4.1 je použití tohoto nástroje a jeho rozšíření, které umožňuje rozpoznávat a odesílat události spojené s používáním tlačítek ovladače, ty jsou odesílány na téma „/survive/events“ a případné příchozí události jsou odebírány na tématu s názvem „/survive/feedback“. Odesílání pohybových dat jsem upravil tak, aby byla také odesílána s předponou „survive“, ta jsou tedy odesílána na téma s názvem „/survive/jmeno\_pose“, kde *jmeno* odpovídá příslušnému sledovanému objektu.

## 4.2 ViveData

V této části se zabývám implementací struktury, která umožňuje pracovat s daty z předchozí sekce ve stylu takovém, který umožňuje tvorbu aplikací ovládajících roboty, jejichž nutnou podmínkou je práce v reálném čase. Výsledkem je implementace třídy ViveData v jazyce C++ zpracovávající data ze systému HTC Vive a poskytující tři metody umožňující využití systému HTC Vive pro účely ovládání robotu. Blokové schéma na Obrázku 4.2 ukazuje propojení jednotlivých subsystémů, kde modře jsou vyznačeny části od firmy KUKA, červeně je značen hardware HTC Vive, fialově jsou označeny části od Libsurvive a zeleně jsou označeny části vzniklé touto prací. Blok Survive Publisher byl rozšířen o další funkcionalitu popsanou v předchozí podkapitole.



Obrázek 4.2: Propojení a komunikace jednotlivých podsystémů.

### 4.2.1 Zpracování dat o poloze Trackeru

Pro využití dat ze systému HTC Vive k řízení robotu je potřeba znát vzájemnou polohu souřadnicového systému, ve kterém jsou poskytována pohybová data, a souřadnicového systému, ve kterém chceme robota řídit. Pro tento účel jsem využil Vive Tracker, který svazuje svůj souřadnicový systém  $O_T$  se souřadnicovým systémem robotu  $O_R$  pomocí volitelného fyzického umístění a otočení Trackeru vůči robotu, orientace souřadnicového systému Trackeru je zobrazena na Obrázku 4.3.



**Obrázek 4.3:** Orientace souřadnicového systému Trackeru.

Poloha souřadnicového systému  $O_T$  vůči souřadnicovému systému  $O_V$  je poskytována dle Kapitoly 4.1 ve zprávě, jejíž obsah tvoří vektor  $\mathbf{r}_T^V$  a jednotkový kvaternion  $\mathbf{q}_T^V$  ve tvaru  $[q_w, q_x, q_y, q_z]^T$ . Pro vytvoření homogenní transformační matice je potřeba převést kvaternion na rotační matici, toho lze docílit pro kvaternion o velikosti jedna pomocí vzorce 4.1[4], homogenní transformační matice ze souřadnicového systému  $O_T$  do souřadnicového systému  $O_V$  je pak dána vztahem 4.2.

$$\mathbf{R}_T^V = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2q_xq_y + 2q_zq_w & 2q_xq_z - 2q_yq_w \\ 2q_xq_y - 2q_zq_w & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2q_yq_z + 2q_xq_w \\ 2q_xq_z + 2q_yq_w & 2q_yq_z - 2q_xq_w & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} \quad (4.1)$$

$$\mathbf{T}_T^V = \left[ \begin{array}{ccc|c} \mathbf{R}_T^V & \mathbf{r}_T^V \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \quad (4.2)$$

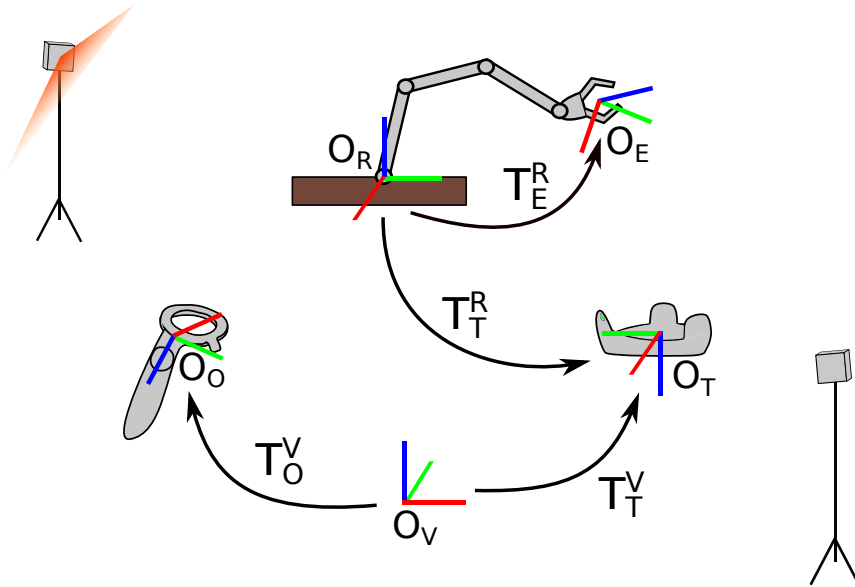
Jelikož je Tracker používán k určení vzájemné polohy souřadnicových systémů, rozhodl jsem se vyjadřovat všechna data ze systému Vive v souřadnicovém systému  $O_T$ , k tomu využiji inverzní transformační matice dle rovnice 4.3.

$$\mathbf{T}_V^T = (\mathbf{T}_T^V)^{-1} = \left[ \begin{array}{ccc|c} (\mathbf{R}_T^V)^T & & & -(\mathbf{R}_T^V)^T \mathbf{r}_T^V \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \quad (4.3)$$

Poloha Trackeru je zpracovávána při vytvoření instance třídy ViveData, kdy je určena průměrná poloha z volitelného počtu vzorků  $N$  s výchozí hodnotou  $N = 100$ , kde průměr vektorů posunutí i kvaternionů určujících rotaci je počítán po složkách. Dle [5] je průměrování kvaternionů tímto způsobem možné, jelikož se jedná o kvaterniony, které by ideálně měly určovat stejnou orientaci, tudíž si jsou blízké. Algoritmus také řeší situaci, že kvaterniony  $\mathbf{q}$  a  $-\mathbf{q}$  označují stejnou orientaci, ta je vyřešena s využitím znaménka skalárního součinu kvaternionů, kdy záporné znaménko označuje, že se jedná o opačný kvaternion, a ten se tudíž musí odečíst. Z průměrné polohy je poté vypočítána transformační matice  $\mathbf{T}_V^T$ .

#### 4.2.2 Zpracování dat o poloze ovladače

Pro potřeby ovládání robotu pomocí ovladače bylo nutné zvolit způsob, který určí, kdy chceme polohu ovladače sledovat. Vzhledem k rozložení tlačítek na ovladači a jejich ovladatelnosti jsem zvolil, že poloha ovladače bude validní, pokud je tlačítko Trigger zmáčknuté. V případě, kdy by pro danou aplikaci bylo vhodné používat míru zmáčknutí tlačítka Trigger, pak bych doporučoval použít k validaci dat tlačítko Grip, jehož jedinou nevýhodou je, že se hůře drží zmáčknuté.



Obrázek 4.4: Schéma vztahů mezi souřadnicovými systémy.

Data o poloze ovladače jsou přijímána v samostatném vlákne, jejich validita je určována porovnáním jejich časové značky vůči času stisknutí či puštění tlačítka Trigger. Dále může být případně použit klouzavý průměr s nastavitelnou délkou na jejich jednoduchou filtraci, průměrování kvaternionů je realizováno stejným způsobem jako v sekci 4.2.1.

Dále jsem se rozhodl, že pro ovládání robotu budu používat pouze změnu polohy ovladače v čase  $t$  tvořenou vektorem  $\mathbf{dr}_t^T$  a kvaternionem  $\mathbf{dq}_t^T$ , tudíž lze použít pouze rotační část matice 4.3. Výsledný vektor posunutí je dán rovnicí 4.4, vztahy mezi souřadnicovými systémy popisuje Obrázek 4.4.

$$\mathbf{dr}_t^T = \mathbf{dr}_{O,t}^T = (\mathbf{R}_O^V)^T (\mathbf{r}_{O,t}^V - \mathbf{r}_{O,t-1}^V) \quad (4.4)$$

Pro určení rotace nejdříve zjistím, jak jsou reprezentovány odpovídající kvaterniony v souřadnicovém systému Trackeru, k tomu využiji rovnic 4.5 až 4.8.

$$\mathbf{q}_{O,t-1}^V = \begin{bmatrix} q_{w,t-1} \\ q_{x,t-1} \\ q_{y,t-1} \\ q_{z,t-1} \end{bmatrix} = \begin{bmatrix} q_{w,t-1} \\ \mathbf{s}_{t-1} \end{bmatrix} \quad (4.5)$$

$$\mathbf{q}_{O,t}^V = \begin{bmatrix} q_{w,t} \\ q_{x,t} \\ q_{y,t} \\ q_{z,t} \end{bmatrix} = \begin{bmatrix} q_{w,t} \\ \mathbf{s}_t \end{bmatrix} \quad (4.6)$$

$$\mathbf{q}_1 = \begin{bmatrix} q_{w,t-1} \\ (\mathbf{R}_T^V)^T \mathbf{s}_{t-1} \end{bmatrix} \quad (4.7)$$

$$\mathbf{q}_2 = \begin{bmatrix} q_{w,t} \\ (\mathbf{R}_T^V)^T \mathbf{s}_t \end{bmatrix} \quad (4.8)$$

Výsledný vztah pro hledaný kvaternion pak udává rovnice 4.9.

$$\mathbf{dq}_t^T = \mathbf{q}_2 \mathbf{q}_1^{-1} \quad (4.9)$$



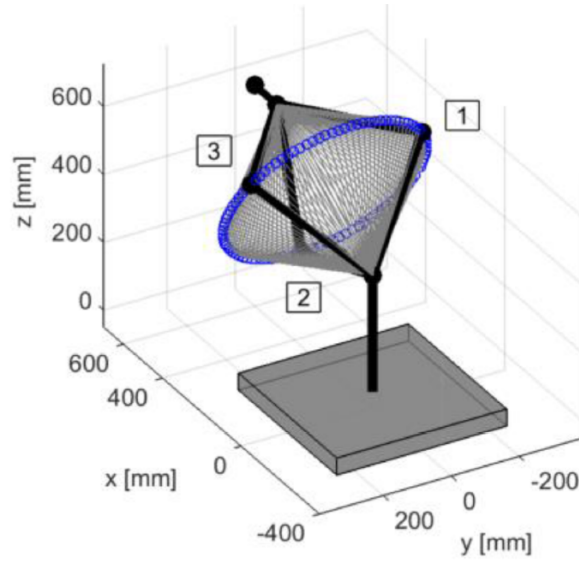
# Kapitola 5

## Ovládací aplikace

Tato kapitola se zabývá studiem podmíněnosti Jacobiho matice robotu v závislosti na úhlu  $\delta$  používaném při řešení inverzní kinematické úlohy, dále je navržen způsob optimalizace tohoto parametru a nakonec je uveden princip a možnosti vytvořené ovládací aplikace.

### 5.1 Studium vlastností Jacobiho matice v závislosti na úhlu $\delta$

Robot KUKA iiwa LBR 7 má sedm stupňů volnosti, proto v obecném případě existuje nekonečně mnoho řešení inverzní kinematické úlohy. Dostupná implementace této úlohy řeší tento problém postupem uvedeným v [18], který zavádí úhel  $\delta$ , jeho volbou se úloha stává jednoznačnou, viz Obrázek 5.1.



**Obrázek 5.1:** Poloha robotu v závislosti na úhlu  $\delta$ [18].

Redundantní stupeň volnosti reprezentovaný úhlem  $\delta$  umožňuje měnit kloubové souřadnice tak, aby se robot nedostal do některých singulárních poloh a také docházelo k maximalizaci prostoru pro změnu polohy chapadla. Toho lze dosáhnout zkoumáním vlastností Jacobiho matice robotu  $\mathbf{J}(\theta) \in \mathbb{R}^{6 \times 7}$ . Jelikož mají jednotlivé prvky Jakobiánu rozdílné fyzikální rozměry je nutné je nejdříve vhodně upravit. K tomu využijí normalizaci rychlostí všech kloubů podle [29], toho docílím pomocí rovnice 5.1.

$$\hat{\mathbf{J}}(\theta) = \mathbf{T}_r \mathbf{J}(\theta) \mathbf{T}_\theta^{-1} \quad (5.1)$$

Diagonální matici  $\mathbf{T}_\theta^{-1}$  tvoří maximální rychlosti jednotlivých kloubů[19] a diagonální matici  $\mathbf{T}_r$  tvoří převrácené hodnoty maximálních rychlostí chapadla, které jsem zvolil odhadem založeným na předpokladu ovládní robotu pomocí Vive ovladače a vlastnostech robotu.

$$\mathbf{T}_\theta^{-1} = \begin{bmatrix} \frac{49}{90}\pi & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{49}{90}\pi & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{5}{9}\pi & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{13}{18}\pi & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{7}{9}\pi & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1\pi & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \pi \end{bmatrix} \quad (5.2)$$

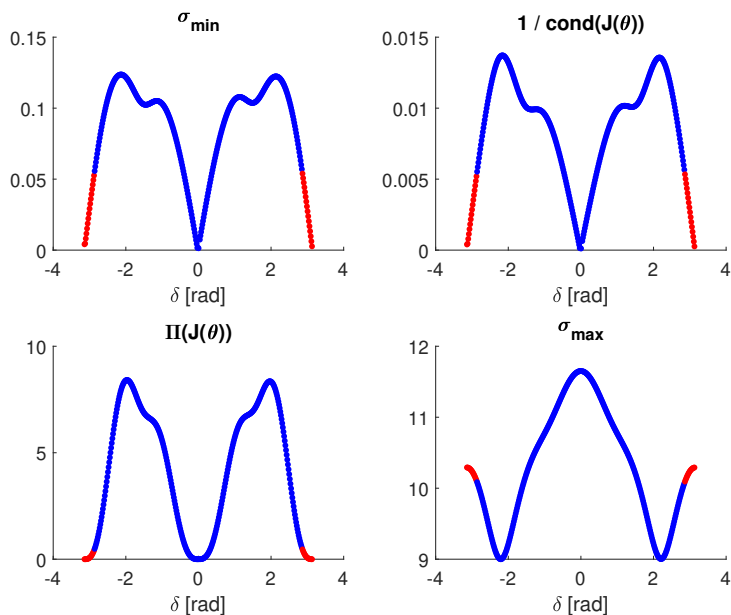
$$\mathbf{T}_r = \begin{bmatrix} 2.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\pi} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\pi} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\pi} \end{bmatrix} \quad (5.3)$$



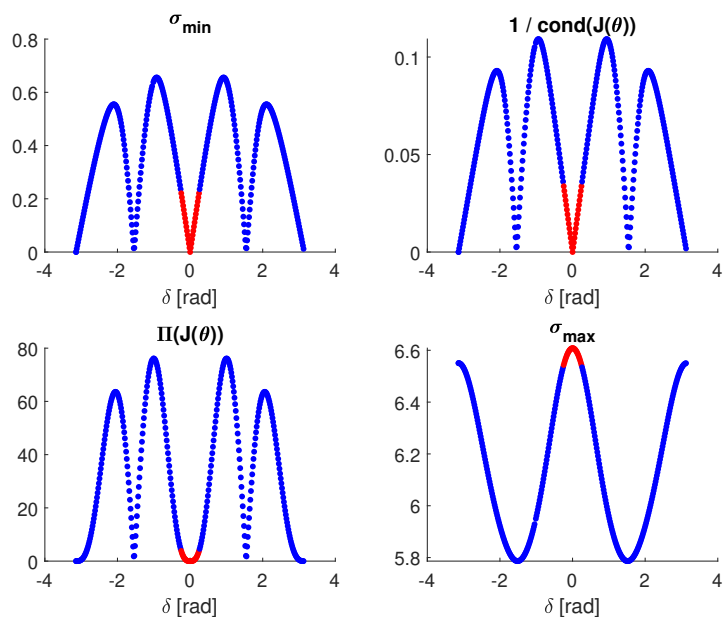
Dále jsem studoval následující způsoby ohodnocení manipulovatelnosti robotu v závislosti na úhlu  $\delta$ .

1. nejmenší singulární číslo matice  $\hat{\mathbf{J}}(\theta)$ :  $\sigma_{\min}$
2. podmíněnost:  $\text{cond}(\hat{\mathbf{J}}(\theta)) = \frac{\sigma_{\max}}{\sigma_{\min}}$
3. součin všech singulárních čísel:  $\prod(\hat{\mathbf{J}}(\theta)) = \prod_{i=1}^6 \sigma_i$ .

Na následujících obrázcích jsou červenou barvou označeny polohy, které jsou nedostupné, modrá barva pak označuje přípustné polohy robotu. Vzdálenost mezi jednotlivými body je jeden stupeň. Typická závislost vybraných parametrů pro polohu v blízkém okolí vnitřní singularity je zobrazena na Obrázku 5.2, dále je na Obrázku 5.3 zobrazena závislost pro polohu, kdy se chapadlo robotu nachází v blízkosti hranice jeho pracovní obálky.

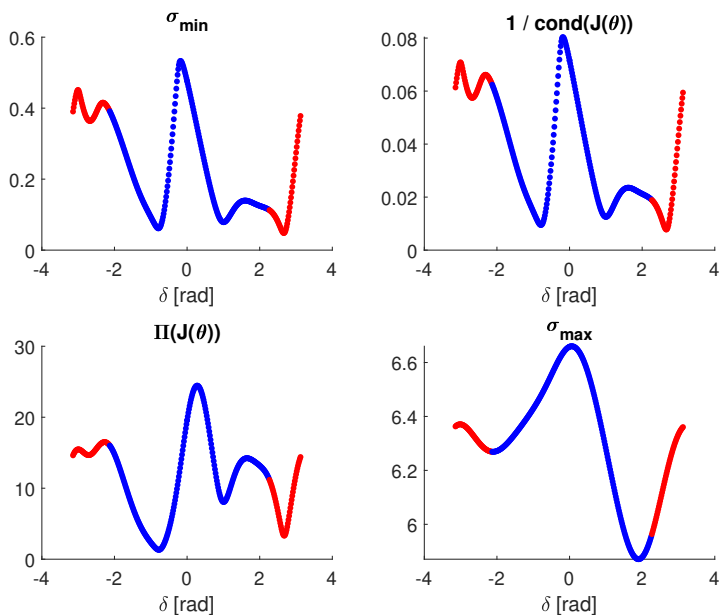


Obrázek 5.2: Vnitřní singularita pro  $\delta = 0$  rad.

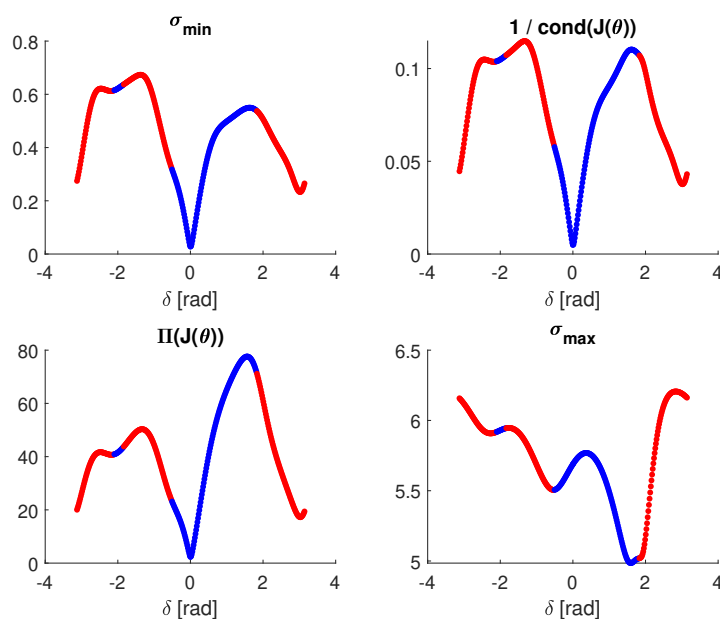


Obrázek 5.3: Závislost na  $\delta$  ke konci pracovní obálky.

V grafech na Obrázku 5.4 je uveden příklad závislosti na úhlu  $\delta$  pro neproblémovou polohu chapadla a grafy Obrázku 5.5 poukazují na možné rozdíly ve výskytu lokálních maxim mezi jednotlivými zkoumanými parametry.



Obrázek 5.4: Příklad závislosti pro neproblémovou polohu chapadla.



**Obrázek 5.5:** Příklad rozdílu ve výskytu lokálních maxim.

Z přiložených grafů lze pozorovat, že jak nejmenší singulární číslo Jakobiánu, tak jeho podmíněnost či součin všech singulárních čísel mají o vlastnostech Jakobiho matice podobný výpovědní charakter, z těchto a dalších obrázků, které jsem analyzoval, se zdá možnost optimalizace úhlu  $\delta$  metodou gradientního stoupání jako přípustná. Optimalizace úhlu  $\delta$  je tématem následující podkapitoly, kde jsem zvolil jako parametr s výpovědní hodnotou součin singulárních čísel Jakobiánu z upravené rovnice 5.1, kde je možné z vlastností ukázaných v [29] pro součin singulárních čísel vypustit z této rovnice matici  $\mathbf{T}_r$ .

## 5.2 Optimalizace úhlu $\delta$

Pro optimalizaci úhlu  $\delta$  jsem zvolil přístup podobný gradientnímu stoupání, gradient je v tomto případě roven derivaci součinu singulárních čísel podle  $\delta$ . Tuto derivaci určím numericky podle rovnice 5.4, kde vektory  $\mathbf{q}$  značí kloubové souřadnice, případně pak souřadnice dané kladnou či zápornou

změnou úhlu  $\delta$  o hodnotu  $\varepsilon > 0$ .

$$\begin{aligned} \frac{d\Pi(\hat{\mathbf{J}}(\theta(\delta)))}{d\delta} &= \frac{d\Pi(\hat{\mathbf{J}}(\mathbf{q}(\delta)))}{d\delta} \\ &= \frac{\Pi(\hat{\mathbf{J}}(\mathbf{q}^+)) - \Pi(\hat{\mathbf{J}}(\mathbf{q}^-))}{(\delta + \varepsilon) - (\delta - \varepsilon)} = \frac{\Pi(\hat{\mathbf{J}}(\mathbf{q}^+)) - \Pi(\hat{\mathbf{J}}(\mathbf{q}^-))}{2\varepsilon} \end{aligned} \quad (5.4)$$

Znaménko derivace v rovnici 5.4 určuje směr pro změnu úhlu  $\delta$ , to lze zapsat rovnicí 5.5, kde parametr  $k > 0$  se určí z podmínek na rychlosti.

$$\hat{\delta} = \delta + \text{sign}\left(\Pi(\hat{\mathbf{J}}(\mathbf{q}^+)) - \Pi(\hat{\mathbf{J}}(\mathbf{q}^-))\right) k\varepsilon \quad (5.5)$$

Parametr  $k$  umožňuje volit změnu úhlu  $\delta$  tak, aby nebyly přesáhnuty maximální rychlosti  $\dot{q}_{i,\max}$  jednotlivých kloubů, parametr  $k$  je tedy pro kladnou hodnotu derivace 5.4 řešením soustavy 5.6, kde  $q_{i,t-1}$  jsou aktuální kloubové souřadnice robotu a  $q_{i,t}$  jsou vypočtené kloubové souřadnice odpovídající nové poloze chapadla. Při záporné derivaci by místo  $q^+$  bylo použito  $q^-$ .

$$\left| \frac{q_{i,t} - q_{i,t-1}}{\Delta t} + \text{sign}(q_i^+ - q_{i,t}) \frac{k_i \varepsilon}{\Delta t} \right| \leq \dot{q}_{i,\max}, \quad i \in \{1, 2, \dots, 7\} \quad (5.6)$$

Tuto soustavu lze snadnou úpravou přepsat do následujícího tvaru

$$\left| k_i - \text{sign}(q_i^+ - q_{i,t}) \frac{q_{i,t} - q_{i,t-1}}{\varepsilon} \right| \leq \frac{\Delta t}{\varepsilon} \dot{q}_{i,\max}, \quad i \in \{1, 2, \dots, 7\} \quad (5.7)$$

Řešení pro jednotlivé nerovnice jsou intervaly dané výrazem 5.8.

$$\begin{aligned} k_i &= [k_{i,\min}, k_{i,\max}] \\ &= \left[ \frac{\Delta t}{\varepsilon} \dot{q}_{i,\max} - \text{sign}(q_i^+ - q_{i,t}) \frac{q_{i,t} - q_{i,t-1}}{\varepsilon}, \frac{\Delta t}{\varepsilon} \dot{q}_{i,\max} + \text{sign}(q_i^+ - q_{i,t}) \frac{q_{i,t} - q_{i,t-1}}{\varepsilon} \right] \end{aligned} \quad (5.8)$$

Hledaný parametr  $k$  je dán maximální hodnotou průniku všech řešení, tj. rovnice 5.9.

$$k = \min_{i=1}^7 k_{i,\max} \quad (5.9)$$

Ke stanovení výsledné hodnoty úhlu  $\delta$  ještě upravím rovnici 5.5, jelikož jsem k výrazu pro hodnotu  $k$  došel linearizací rychlosti, musím tuto hodnotu korigovat korekčním členem  $\eta$  s hodnotou například  $\eta = 0.1$ . Tento člen zajišťuje, že v případě, kdy nedochází ke změně polohy chapadla, nebude díky nelineárnímu vztahu překročena maximální rychlost některého z kloubů.

$$\hat{\delta} = \delta + \text{sign} \left( \prod \left( \hat{\mathbf{J}}(\mathbf{q}^+) \right) - \prod \left( \hat{\mathbf{J}}(\mathbf{q}^-) \right) \right) \eta k \epsilon \quad (5.10)$$

## 5.3 Funkce ovládací aplikace

Ovládací aplikace umožňuje použít data ze systému Vive k ovládní polohy robotu, každých 10 ms je ovládací aplikaci předána aktuální poloha robotu a požadována poloha nová. Při spuštění aplikace je z aktuální polohy robotu určena poloha souřadnicového systému chapadla  $O_E$  a tedy matice  $\mathbf{T}_{E,0}^R$  přímou kinematickou úlohou (DKT).

$$\mathbf{T}_{E,0}^R = \left[ \begin{array}{ccc|c} \mathbf{R}_{E,0}^R & \mathbf{1}_{E,0}^R \\ 0 & 0 & 0 & 1 \end{array} \right] \leftarrow DKT \quad (5.11)$$

Pro určení posunutí chapadla v čase  $t$  pomocí vektoru  $\mathbf{dr}_t^T$  z rovnice 4.4 využiji známé orientace Trackeru vůči robotu, posunutí v souřadnicovém systému robotu je pak dáno rovnicí 5.12.

$$\mathbf{dr}_t^R = \mathbf{R}_T^R \mathbf{dr}_t^T \quad (5.12)$$

K získání rotace v souřadnicovém systému robotu nejprve otočím osu otáčení danou kvaternionem  $\mathbf{dq}_t^T$  a výsledný kvaternion opět vzorcem 4.1 převedu na rotační matici  $\mathbf{R}_t^R$ .

$$\mathbf{dq}_t^T = \begin{bmatrix} dq_w \\ \mathbf{s} \end{bmatrix}_t^T = \begin{bmatrix} dq_w \\ dq_x \\ dq_y \\ dq_z \end{bmatrix}_t^T \quad (5.13)$$

$$\mathbf{R}_t^R \leftarrow \mathbf{dq}_t^R = \begin{bmatrix} dq_w \\ \mathbf{R}_T^R \mathbf{s} \end{bmatrix}_t^R \quad (5.14)$$

Celkové posunutí  $\mathbf{l}_{E,t}^R$  až do času  $t$  je dáno rovnicí 5.15 a celkové otočení  $\mathbf{R}_{E,t}^R$  rovnicí 5.16. Výsledná transformační matice udávající polohu chapadla v čase  $t$  je dána rovnicí 5.17, z té se získají kloubové souřadnice řešením inverzní kinematické úlohy.

$$\mathbf{l}_{E,t}^R = \mathbf{l}_{E,t-1}^R + \mathbf{d}\mathbf{r}_t^R \quad (5.15)$$

$$\mathbf{R}_{E,t}^R = \mathbf{R}_t^R \mathbf{R}_{E,t-1}^R \quad (5.16)$$

$$\mathbf{T}_{E,t}^R = \left[ \begin{array}{ccc|c} \mathbf{R}_{E,t}^R & & & \mathbf{l}_{E,t}^R \\ 0 & 0 & 0 & 1 \end{array} \right] \quad (5.17)$$

Ovládací aplikace také reaguje na stisk tlačítek, držením tlačítka Trackpad vpravo a vlevo lze postupně měnit úhel ramene  $\delta$ , stisk tohoto tlačítka v dolní části přepne do režimu jemného doladění, kdy k otáčení i posuvu dochází o hodnotu 4 krát menší, stisk v horní části pak přepne zpět na poměr pohybů 1:1.

Stisk tlačítka Menu umožňuje změnit funkci tlačítka Trackpad na volbu poměru pohybu ovladače a robotu, kde stisk vpravo vybírá větší poměr translačního pohybu, stisk vlevo tento poměr zase zmenšuje, k ovládání rotační složky slouží stisk nahoře pro zvyšování a stisk dole pro zmenšování tohoto poměru. K dispozici jsou poměry 1:10, 1:4, 1:2, 3:4, 1:1, 3:2 a 2:1.

Tlačítko Grip slouží jako menší bezpečnostní pomůcka, která zastaví ovládání robotu, nicméně se nejedná o bezpečnostní tlačítko. Ovládací aplikace samozřejmě kontroluje, aby nedošlo k překročení maximálních rychlostí kloubů a aby jejich hodnoty nepřekračovaly limity robotu včetně pětistupňové rezervy. K ovládací aplikaci byl ještě vytvořen nástroj pro zobrazování informací ohledně aktuální funkce tlačítka Trackpad, právě zvoleného poměru mapování pohybů ovladače a robotu a dalších vybraných parametrů.

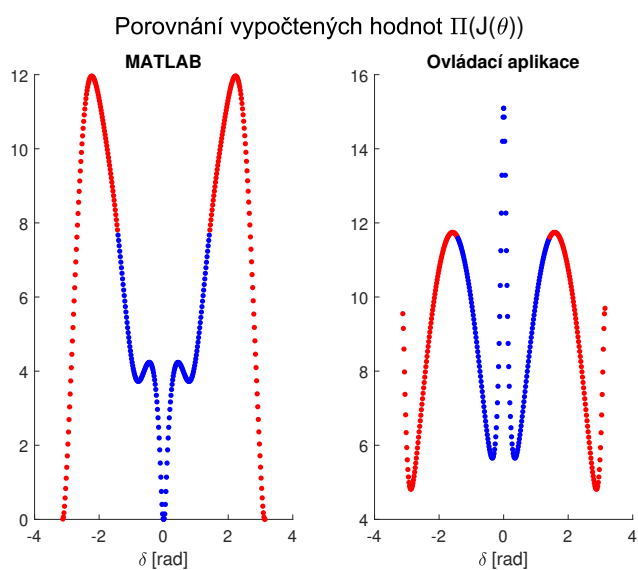
# Kapitola 6

## Testování a možná vylepšení

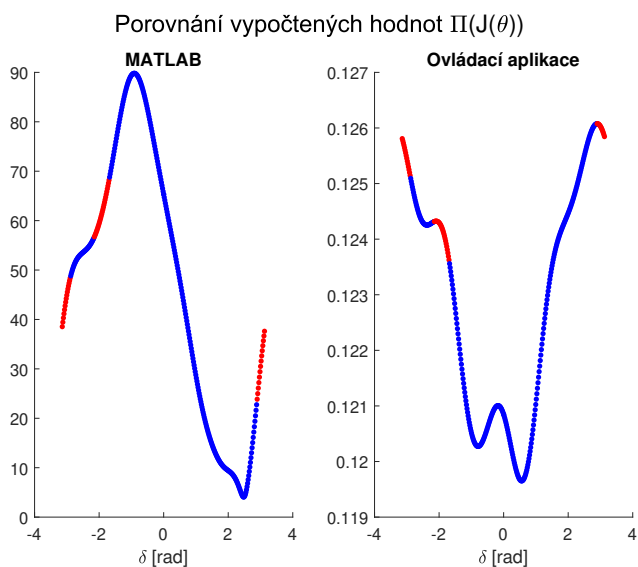
### 6.1 Testování funkčnosti

#### 6.1.1 Optimalizace úhlu $\delta$

Testování implementace optimalizace úhlu probíhalo pomocí simulátoru s vizualizací nástrojem Rviz v prostředí ROS. Pomocí tlačítek ovladače bylo možné cíleně měnit úhel  $\delta$ , ale i zapínat a vypínat jeho automatickou změnu dle postupu uvedeného v Podkapitole 5.2. Prvotní testy ukázaly, že při přibližování se k vnitřní singulární poloze, kdy je robot ve tvaru písmene L, byl úhel měněn ve směru, který zhoršoval polohu robotu, viz Obrázek 6.1a. Dalším testováním jsem zjistil, že hodnoty vypočtené v ovládací aplikaci vůbec nesouhlasí s hodnotami vypočtenými v MATLABu i přesto, že jsou počítány ekvivalentním způsobem ze stejných Jakobiánů. Výpočty provedené v MATLABu jsou správné, jelikož matice vypočtené ze singulárních rozkladů Jakobiánů odpovídají původním Jakobiánům. Totéž nelze říci o výpočtech v ovládací aplikaci, kdy se tyto matice liší o nezanedbatelné hodnoty a to i pro nesingulární polohy, viz Obrázek 6.1b.



(a) : Porovnání hodnot pro polohu robotu blízkou tvaru L.



(b) : Porovnání hodnot pro neproblematickou polohu.

Dále jsem vyzkoušel ostatní dostupné možnosti výpočtu singulárního rozkladu s využitím stejné knihovny[9]. Tyto výsledky opět nebyly správné, proto jsem využil možnosti výjádřit součin všech singulárních čísel pomocí vzorce 6.1[29], nicméně i tento postup dával stejné ale špatné výsledky.

$$\Pi(\hat{\mathbf{J}}(\theta)) = \sqrt{\det(\hat{\mathbf{J}}(\theta)\hat{\mathbf{J}}^T(\theta))} \quad (6.1)$$



Pro výpočty byla použita populární knihovna pro lineární algebru s názvem Eigen[9], konkrétní zdroj chyb prozatím nebyl nalezen. Vzhledem k časové tísní jsem tento problém nevyřešil a optimalizace úhlu není v ovládací aplikaci k dispozici, nicméně její teoreticky funkční podoba je v aplikaci implementována.

## ■ 6.1.2 Užití systému HTC Vive pro ovládání robotu

Využití systému HTC Vive k ovládání robotu bylo otestováno, jak v simulátoru, tak pro reálného robotu. Navržené způsoby ovládání fungují tak, jak byly zamýšlené. Funkčnost tohoto systému stojí na programu Survive Publisher, který sbírá data ze systému HTC Vive a ta dále šíří. Bohužel tento program není úplně spolehlivý, jelikož knihovna Libsurvive není plně vyvinuta a obsahuje tak chyby, díky kterým nastávají případy, kdy program Survive Publisher spadne. Jejich četnost je zhruba 3 až 4 výskyty za hodinu. Vyzkoušeno bylo i několik novějších verzí knihovny Libsurvive, kde u těch nejnovějších se zdá, že tyto chyby jsou vyřešeny, nicméně při současném připojení ovladače i Trackeru, nebylo možné program úspěšně spustit i přesto, že při samostatném zapojení tato zařízení fungovala bezchybně.

Dále jsem testoval vliv zakrytí ovladače či majáku na polohová data, jak odstínění ovladače od jednoho z majáků tak zakrytí jednoho majáku způsobily konstantní offset. Úplné zakrytí ovladače mělo za následek nepřesná data, jejichž offset vždy konstantně rostl v jednom náhodném směru. Při opětovném odkrytí ovladače byla opět přenášena správná data, pokud byl vzniklý offset menší než několik metrů. Bylo ověřeno, že přítomnost odrazivých předmětů v blízkém okolí způsobuje vysoký nárůst šumu v poloze.

Ovládací aplikace funguje tak, aby případný pád programu Survive Publisher nezpůsobil nechtěné pohyby robotu. Dále je uživatel informován pokud po delší dobu nepřichází data, což naznačuje, že program Survive Publisher přestal správně fungovat, ten pak stačí znovu spustit a může se pokračovat v činnosti.

Samotné ovládání robotu je jednoduché a intuitivní, jelikož je možné zároveň ovládat jak pozici, tak orientaci a to v libovolných směrech, což u produktů jako je Smart Pendant od společnosti Yaskawa zatím není možné. Směr pohybu ovladače zároveň odpovídá směru pohybu chapadla robotu a využití tlačítek ke změně poměru pohybů umožňuje i přesné polohování.

Celkově se dá konstatovat, že je možné vytvořený systém využít pro účely, kdy případný výpadek nezpůsobí žádný problém, příkladem by mohlo být

učení trajektorie, kde by jedno z tlačítek mohlo být využito pro uložení jednotlivých záchytných bodů cílové trajektorie.

## 6.2 Náměty k možným vylepšením

Hlavní prioritou pro vylepšení tohoto systému je samozřejmě zajištění stabilního chodu programu Survive Publisher a to buď opravením chyb v této verzi nebo nalezením jiné funkční a stabilní verze pro tyto účely. Hardware využívaný v této práci je také součástí první série pro virtuální realitu od společnosti HTC Vive. Existují již další dvě verze, kde jejich součástí je i druhá generace světelných majáků, které by taktéž měly být podporovány novějšími verzemi knihovny Libsurvive a je možné, že s ní budou fungovat lépe.

Jako jedna z možností rozšíření tohoto systému o další prvky připadá v úvahu využití poloprůhledných brýlí pro smíšenou realitu. Příkladem je zařízení HoloLens, tato technologie by mohla být využita například k zobrazování vybraných informací ohledně robotu a zároveň umožňovat přehled o okolí, čímž by se nenarušovala bezpečnost při práci.

V případě správně fungujícího výpočtu singulárních čísel Jacobiho matice, by mohla být singulární čísla využita k signalizaci špatné podmíněnosti pomocí haptické odezvy, kde by její síla či frekvence odpovídala míře přiblížení singulární poloze. Haptická odezva by také mohla být v některých situacích využita pro primitivní pocítění velikosti sil působících na chapadlo robotu.

# Kapitola 7

## Závěr

V této práci došlo k seznámení s principem určování polohy systémem HTC Vive, data o poloze poskytovaná tímto systémem byla s využitím knihovny Libsurvive programem Survive Publisher publikována do prostředí ROS k jejich dalšímu využití. Došlo k prozkoumání událostí spojených s používáním tlačítek ovladače Vive a zařízení Vive Tracker. Tyto události byly zdokumentovány a dále využity k rozšíření programu Survive Publisher o tyto události.

Byla vytvořena třída ViveData v jazyce C++, která sbírá data o poloze a události tlačítek z prostředí ROS od systému HTC Vive. Data o poloze jsou transformována do vhodného souřadnicového systému. Třída poskytuje tři metody umožňující předávání informací s ovládací aplikací běžící v reálném čase. Jedna metoda předává transformovaná data o poloze, druhá se stará o události tlačítek a třetí slouží k předání dat z ovládací aplikace do systému Vive.

Vytvořená ovládací aplikace robotu používá data ze systému HTC Vive prostřednictvím instance třídy ViveData k ovládní polohy chapadla robotu KUKA iiwa LBR 7. Pro určení polohy je používán Vive ovladač, kdy robot sleduje jeho polohu při držení jednoho zvoleného tlačítka. Ovládací aplikace dále umožňuje například měnit poměr mapování pohybu ovladače na pohyb robotu či měnit úhel ramene robotu, který je k dispozici díky nadbytečnému stupni volnosti, využitím tlačítek ovladače. Ovládní chapadla robotu je intuitivní, jelikož jsou všechny pohyby ovladače brány v souřadnicovém systému robotu, neboli pohyb ovladačem v jakémkoliv směru z pohledu osoby ovládající robot zapříčiní posunutí chapadla robotu ve stejném směru

a ovladač navíc umožňuje měnit najednou jak pozici, tak orientaci a to v libovolných směrech.

Nakonec byla prozkoumána závislost singulárních čísel Jakobiho matice robotu na úhlu jeho ramene. Byla navržena jednoduchá metoda optimalizace tohoto parametru na principu gradientního stoupání. Při jejím testování bylo zjištěno, že knihovna použitá k výpočtům SVD rozkladu a singulárních čísel nedává správné výsledky. Byly vyzkoušeny ostatní dostupné metody této knihovny k výpočtu SVD rozkladu i alternativní vyjádření a výpočet optimalizačního kritéria, nicméně všechny metody selhaly stejným způsobem. Správnost postupu byla ověřena výpočtem v MATLABu. Vzhledem k časové tísní nebyl problém správného spočítání optimalizačního kritéria v ovládací aplikaci vyřešen.

Výsledek této práce byl otestován i pro ovládání reálného robotu KUKA iiwa LBR 7. K aktivnímu využití systému HTC Vive pro řízení robotů, je ale potřeba vyřešit problém s programem Survive Publisher, který díky chybám v knihovně Libsurvive občas spadne. Tudiže nelze robot ovládat, dokud nedojde k jejímu opětovnému spuštění. Systém tak lze prozatím použít pouze pro případy, kdy tato skutečnost není problémem. I přesto věřím, že výsledky této práce naleznou využití alespoň na naší škole.

K možným způsobům rozšíření této práce by mohlo patřit využití haptické odezvy ovladače k signalizaci špatné manipulovatelnosti robotu či velikosti sil, které působí na chapadlo. Další možností je například rozšíření systému o poloprůhledné brýle HoloLens pro smíšenou realitu, které by mohly poskytovat důležité informace pro operátora bez nutnosti opuštění pracovního prostoru a bez nebezpečí způsobeného neprůhledností klasických brýlí pro virtuální realitu.



## Literatura

- [1] Morten Andre Astad, Mathias Hauan Arbo, Esten Ingar Grøtli, and Jan Tommy Gravdahl. Vive for robotics: Rapid robot cell calibration. In *2019 7th International Conference on Control, Mechatronics and Automation (ICCMA)*, pages 151–156. IEEE, 2019.
- [2] Miguel Borges, Andrew Symington, Brian Coltin, Trey Smith, and Rodrigo Ventura. Htc vive: analysis and accuracy improvement. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2610–2615. IEEE, 2018.
- [3] cntools. Libsurvive - Open Source Lighthouse Tracking System [Online]. 2020 [cit. 13.5.2020]. Dostupné z: <https://github.com/cntools/libsurvive>.
- [4] James Diebel. Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix*, 58(15-16):1–35, 2006.
- [5] Elecman. Averaging Quaternions and Vectors. [Online], 2013 [cit. 30.4.2020]. Dostupné z: [http://wiki.unity3d.com/index.php/Averaging\\_Quaternions\\_and\\_Vectors](http://wiki.unity3d.com/index.php/Averaging_Quaternions_and_Vectors).
- [6] Cinzia Freschi, Vincenzo Ferrari, Franca Melfi, Mauro Ferrari, Franco Mosca, and Alfred Cuschieri. Technical review of the da vinci surgical telemanipulator. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 9(4):396–406, 2013.
- [7] Franck Geffard, Philippe Garrec, Gérard Piolain, Marie-Anne Brudieu, Jean-François Thro, Alain Coudray, and Eric Lelann. Tao2000 v2 computer-assisted force feedback telemanipulators used as maintenance and production tools at the areva nc–la hague fuel recycling plant. *Journal of Field Robotics*, 29(1):161–174, 2012.

- [8] Marcus Greiff, Anders Robertsson, and Karl Berntorp. Performance bounds in positioning with the vive lighthouse system. In *2019 22th International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2019.
- [9] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. <http://eigen.tuxfamily.org>, 2010.
- [10] HTC Corporation. About the VIVE controllers. [Online], 2011-2020 [cit. 13.5.2020]. Dostupné z: [https://www.vive.com/us/support/vive/category\\_howto/about-the-controllers.html/](https://www.vive.com/us/support/vive/category_howto/about-the-controllers.html/).
- [11] HTC Corporation. Tips for setting up the base stations. [Online], 2011-2020 [cit. 13.5.2020]. Dostupné z: [https://www.vive.com/eu/support/vive/category\\_howto/tips-for-setting-up-the-base-stations.html](https://www.vive.com/eu/support/vive/category_howto/tips-for-setting-up-the-base-stations.html).
- [12] HTC Corporation. VIVE. [Online], 2011-2020 [cit. 13.5.2020]. Dostupné z: <https://www.vive.com/eu/product/vive/>.
- [13] HTC Corporation. VIVE Tracker (2018) [obraz, online]. 2011-2020 [cit. 29.4.2020]. Dostupné z: <https://www.vive.com/eu/accessory/>.
- [14] Intuitive Surgical, Inc. the da Vinci robot surgical system [obraz, online]. 2020 [cit. 13.5.2020]. Dostupné z: <http://www.allonrobots.com/da-vinci-robot.html>.
- [15] Kiho Kim and Hodong Kim. Remote robotic systems for nuclear environment application. In *ICCAS 2010*, pages 1843–1846. IEEE, 2010.
- [16] Kiho Kim, Jongkwang Lee, Byung Suk Park, Sunghyun Kim, Seungnam Yu, Dongseok Ryu, Jonghee Han, and Ilje Cho. Remote manipulator systems for pyroprocessing facility application. In *IEEE ISR 2013*, pages 1–5. IEEE, 2013.
- [17] Oliver Kreylos. Lighthouse tracking examined [Online]. 2016 [cit. 13.5.2020]. Dostupné z: <http://doc-ok.org>.
- [18] I Kuhlemann, A Schweikard, P Jauer, and F Ernst. Robust inverse kinematics by configuration control for redundant manipulators with seven dof. In *2016 2nd International Conference on Control, Automation and Robotics (ICCAR)*, pages 49–55. IEEE, 2016.
- [19] KUKA. Sensitive robotics\_LBR iiwa. [Online], 2017 [cit. 13.5.2020]. Dostupné z: [https://www.kuka.com/-/media/kuka-downloads/imported/9cb8e311bfd744b4b0eab25ca883f6d3/kuka\\_lbr\\_iiwa\\_brochure\\_en.pdf?rev=7141546264dd45ca9f613adbe6393405](https://www.kuka.com/-/media/kuka-downloads/imported/9cb8e311bfd744b4b0eab25ca883f6d3/kuka_lbr_iiwa_brochure_en.pdf?rev=7141546264dd45ca9f613adbe6393405).

- [20] Jong Kwang Lee, Hyo Jik Lee, Byung Suk Park, and Kiho Kim. Bridge-transported bilateral master–slave servo manipulator system for remote manipulation in spent nuclear fuel processing plant. *Journal of Field Robotics*, 29(1):138–160, 2012.
- [21] Uživatel nairol. Unofficial Documentation for the Lighthouse Tracking System [Online]. 2018 [cit. 13.5.2020]. Dostupné z: <https://github.com/nairol/LighthouseRedox>.
- [22] Diederick C Niehorster, Li Li, and Markus Lappe. The accuracy and precision of position and orientation tracking in the htc vive virtual reality system for scientific research. *i-Perception*, 8(3):2041669517708205, 2017.
- [23] Open Source Robotics Foundation. ROS Introduction. [Online], 2018 [cit. 13.5.2020]. Dostupné z: <http://wiki.ros.org/ROS/Introduction>.
- [24] G Piolain, Franck Geffard, A Coudray, Philippe Garrec, JF Thro, and Yann Perrot. Dedicated and standard industrial robots used as force-feedback telemaintenance remote devices at the areva recycling plant. In *2010 1st International Conference on Applied Robotics for the Power Industry*, pages 1–6. IEEE, 2010.
- [25] Sanghun Pyo, Syed Hassan, Yasir Jan, and Jungwon Yoon. Design of 6-dof manipulator intuitive teaching system using smart phone orientation: User friendly and intuitive teaching operation for 6-dof manipulator. In *2013 4th International Conference on Intelligent Systems, Modelling and Simulation*, pages 364–369. IEEE, 2013.
- [26] JP Ruurda, Th JMV Van Vroonhoven, and IAMJ Broeders. Robot-assisted surgical systems: a new era in laparoscopic surgery. *Annals of the Royal College of Surgeons of England*, 84(4):223, 2002.
- [27] Elliot Williams. Alan Yates: Why valve’s lighthouse can’t work [Online]. 2016 [cit. 13.5.2020]. Dostupné z: <https://hackaday.com/2016/12/21/alan-yates-why-valves-lighthouse-cant-work/>.
- [28] Yaskawa America, Inc. Robot Programming using Smart Frames. *YouTube* [Online], 19.7.2019 [cit. 13.5.2020]. Dostupné z: <https://www.youtube.com/watch?v=hegj571WcZ0>.
- [29] Tsuneo Yoshikawa. Manipulability and redundancy control of robotic mechanisms. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 1004–1009. IEEE, 1985.







## Příloha A

### Seznam příloh

1. vive\_survive\_publisher.h: hlavičkový soubor programu Survive Publisher
2. main.cc: implementace programu Survive Publisher
3. ViveEvents.h: definice událostí tlačítek
4. ViveData.h: hlavičkový soubor třídy ViveData
5. ViveData.cpp: implementace třídy ViveData
6. ViveController.h: hlavičkový soubor ovládací aplikace robotu
7. ViveController.cpp: implementace ovládací aplikace robotu



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Sojma** Jméno: **Dmitrij** Osobní číslo: **474381**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra kybernetiky**  
Studijní program: **Kybernetika a robotika**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Ovládání pohybu průmyslového robotu nástroji pro virtuální realitu**

Název bakalářské práce anglicky:

**Industrial Robot Control Using Virtual Reality Tools**

Pokyny pro vypracování:

1. Seznamte se s robotem KUKA LBR, prostředím ROS a systémem HTC VIVE.
2. Navrhněte algoritmy, které zpracují informace o pohybu snímaném systémem HTC VIVE a použijí je pro řízení pohybu průmyslového robotu.
3. Systém implementujte a otestujte pro reálné scénáře použití.
4. Výsledky vyhodnoťte a navrhněte možná zlepšení.

Seznam doporučené literatury:

- [1] M. E. Walker, H. Hedayati and D. Szafir, "Robot Teleoperation with Augmented Reality Virtual Surrogates," 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Daegu
- [2] Melchiorri C. (2013) Robot Teleoperation. In: Baillieul J., Samad T. (eds) Encyclopedia of Systems and Control. Springer, London
- [3] Andreas Dunser, Martin Lochner, Ulrich Engelke, David Rozado Fernandez, Visual and Manual Control for Human-Robot Teleoperation, May-June 2015, pp. 22-32, vol. 35, IEEE Computer Graphics and Applications

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. Vladimír Smutný, Ph.D., robotické vnímání CIIRC**

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

Datum zadání bakalářské práce: **08.01.2020**

Termín odevzdání bakalářské práce: \_\_\_\_\_

Platnost zadání bakalářské práce: **30.09.2021**

\_\_\_\_\_  
Ing. Vladimír Smutný, Ph.D.  
podpis vedoucí(ho) práce

\_\_\_\_\_  
doc. Ing. Tomáš Svoboda, Ph.D.  
podpis vedoucí(ho) ústavu/katedry

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

## III. PŘEVZETÍ ZADÁNÍ

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

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta