

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

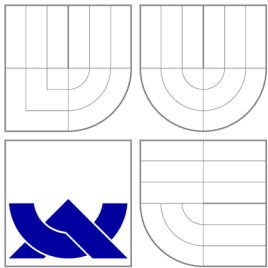
## UŽIVATELSKÉ ROZHRANÍ PRO ŘÍZENÍ ROBOTA PŘI UCHOPOVÁNÍ OBJEKTŮ

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

ROMAN MANĎÁK

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# UŽIVATELSKÉ ROZHŘANÍ PRO ŘÍZENÍ ROBOTA PŘI UCHOPOVÁNÍ OBJEKTŮ

USER INTERFACE FOR ROBOT CONTROL IN OBJECT GRASPING TASK

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ROMAN MANDÁK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VÍTĚZSLAV BERAN, Ph.D.

BRNO 2015

## Abstrakt

Bakalářská práce se zabývá návrhem a implementací uživatelského rozhraní pro řízení robota při uchopení objektů. Poskytuje dva módy ovládání ramene pro jednoduché a složité úkoly uchopení. V práci je využito a porovnáno pro snímání gest uživatele různých typů běžných i méně běžných zařízení. Součástí práce je detailní rozbor použitých nástrojů využívajících se v robotice, popis existujících řešení sloužící jako podklad pro návrh aplikace a pravidla pro tvorbu uživatelského rozhraní. Dále jsou popsány návrh, implementace, provedené experimenty s aplikací a na závěr výsledek porovnání ovládacích zařízení.

## Abstract

This bachelor work is focused on design and implementation of the user interface (UI) for the control of a robot during grasping objects. Common devices as well as the less ordinary ones were used and compared for capturing user gestures of various types. The work includes an in-depth analysis of the devices which are used in robotics, a description of existing solutions which serves as a ground for application design, and rules for creating the UI. Further are described design, implementation, the experiments with the application which were conducted, and in the end the result of the comparison of control devices.

## Klíčová slova

Ovládání robotického manipulátoru, ROS, PR2, Uživatelské rozhraní, Gamepad, 2D myš, 3D myš

## Keywords

Robotic manipulator teleoperation, ROS, PR2, User interface, Gamepad, 2D mouse , 3D mouse

## Citace

Roman Mandák: Uživatelské rozhraní pro řízení robota při uchopování objektů, bakalářská práce, Brno, FIT VUT v Brně, 2015

# Uživatelské rozhraní pro řízení robota při uchopování objektů

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Vítězslava Berana, Ph.D.

.....  
Roman Mandák  
18. května 2015

## Poděkování

Chtěl bych poděkovat vedoucímu mé práce Ing. Vítězslavu Beranovi, Ph.D. za odborné vedení, za pomoc a rady při zpracování této práce.

© Roman Mandák, 2015.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*



# Obsah

|  |           |
|--|-----------|
| <b>1 Úvod</b>  | <b>2</b>  |
| <b>2 Robotika a uživatelské rozhraní</b>                 | <b>3</b>  |
| 2.1 PR2 . . . . .  | 3         |
| 2.2 Robotický operační systém . . . . .                  | 4         |
| 2.3 Nástroje . . . . .                                   | 5         |
| 2.4 Ovládací zařízení . . . . .                          | 7         |
| 2.5 Pravidla pro tvorbu uživatelského rozhraní . . . . . | 8         |
| 2.6 Existující řešení . . . . .                          | 9         |
| <b>3 Ovládání pozice ruky robota</b>                     | <b>13</b> |
| 3.1 Specifikace zadání . . . . .                         | 13        |
| 3.2 Způsob řešení problému . . . . .                     | 14        |
| 3.3 Ovládání . . . . .                                   | 15        |
| 3.4 Mapování dat z ovládacích zařízení . . . . .         | 16        |
| 3.5 Vizualizace zpětné vazby . . . . .                   | 16        |
| <b>4 Vizualizační a řídicí komponenty</b>                | <b>19</b> |
| 4.1 Základ implementace . . . . .                        | 19        |
| 4.2 Ovládací zařízení . . . . .                          | 20        |
| 4.3 Vizualizace scény a bodu uchopení . . . . .          | 21        |
| 4.4 Testování a experimenty . . . . .                    | 24        |
| 4.5 Vyhodnocení experimentů . . . . .                    | 26        |
| <b>5 Závěr</b>   | <b>30</b> |
| <b>A Obsah CD</b>  | <b>32</b> |

# Kapitola 1

## Úvod

Robot je stroj pracující s určitou mírou autonomního chování provádějící předem definované úkoly. Svými senzory vnímá stav okolního světa i jeho vliv na něj, tím získává určitou zpětnou vazbu, která reflektuje jeho zásahy do okolí. S rostoucím uplatněním robotů, a to nejen v oblasti průmyslu, roste jejich dostupnost a množství typů. Díky své samostatnosti mohou například pomoci starým lidem k usnadnění běžných denních záležitostí nebo při zkoumání oblastí nedostupných nebo nebezpečných pro člověka. Přes dnešní velkou míru autonomního chování robotů jsou situace, kdy robot není schopen problém vyřešit nebo situace vyžaduje specifický přístup. Zde je pořad vhodnější využít člověka jako operátora a robotovi poradit. V takovýchto případech operátor skrze uživatelské rozhraní ovládá robota a získává zpětnou vazbu.

Tato práce se týká studia, návrhu, vývoje uživatelského rozhraní pro uchopování objektů. Program s tímto uživatelským rozhraním bude sloužit pro ovládání robotického manipulátoru ve dvou módech pro jednodušší situace uchopení objektu jako přemístění krabice od čaje a složitější vyskládání nákupu z košíku. Jako ovládací zařízení slouží gamepad, 3D myš a 2D myš. Tato zařízení jsou běžná i méně běžná pro normálního uživatele a jsou primárně určena k jiným účelům. V práci se zaměřím na vhodný návrh ovládání jednotlivými zařízeními, jejich srovnání a výběr nejvhodnějšího ovládání v různých situacích. Dále se budu zabývat vhodnou vizualizací dat z robotových senzorů, aby uživateli maximálně usnadnily provedení úkolů. Cílem práce je program s dobře použitelným uživatelským rozhraním a efektivním ovládáním aplikace.

Práce je rozdělena do tří navazujících kapitol. V první kapitole jsou teoreticky popsány nástroje využívající se v robotice, základní informace o robotické platformě PR2, ovládací zařízení využítá ke snímání uživatelových gest a základní pravidla pro tvorbu uživatelského rozhraní. Třetí kapitola se zabývá návrhem aplikace, jejího uživatelského rozhraní a dalších částí k ovládání aplikace. Poslední kapitola popisuje realizaci aplikace, návrh a provedení experimentů.

## Kapitola 2

# Robotika a uživatelské rozhraní

V této kapitole lze nalézt základní informace o robotické platformě PR2 a nástrojích využívaných se v robotice. Čtenář je seznámen také se základními pravidly tvorby uživatelského rozhraní, zařízeními využitými k ovládání aplikace a již existujícími podobnými řešeními.

### 2.1 PR2

PR2 je robotická platforma vyvinutá společností Willow Garage. Platforma je speciálně navržena pro testování a integraci různých aplikací a technologií, dovoluje vývojářům použít různé typy chapadel, předloktí nebo celých paží apod. Případně mohou být přidány nebo odebrány různé senzory.

PR2 je složen ze základny, trupu, robotických ramen a hlavy, jak můžeme vidět na obrázku 2.1. Základna obsahuje výpočetní mozek robota v podobě dvou čtyřjádrových procesorů i7 Xeon s pamětí DD3 RAM 24GB a vnitřní pamětí 500GB[9]. Nachází se zde také laserový senzor UTM-30LX<sup>1</sup> pro detekci objektů ve větší vzdálenosti. K základně jsou připevněna čtyři kola typu Caster umožňující pohyb robota všemi směry maximální rychlostí 1m/s. [9]

Na základně je umístěn trup robota, který je schopen měnit svou výšku a tím i velikost robota. K trupu robota jsou připevněny dvě robotická ramena a hlava. Ve vrchní části trupu pod hlavou je umístěn stejný laserový skener jako na základně využívající se však pro detekci objektů v krátké vzdálenosti od robota.

Robotická ramena se skládají z paže se čtyřmi stupni volnosti, zápěstí se třemi stupni volnosti a chapadel s jedním stupněm volnosti. Na zápěstí robota je připevněna barevná kamera pro lepší pohled na práci chapadel. Dále chapadlo obsahuje 3-osý akcelerometr a na jeho dva prsty je možné umístit tlakové senzory. Ramena PR2 jsou vyvinutá pro maximální bezpečnost, narazí-li rameno do objektu, vyvolá reakci na opačný směr pohybu a okamžité zastavení. Robotické rameno si i při odpojení napájení motorů udrží svoji polohu, díky systému protivah[8]. Složitý systém protivah nám umožňuje využít také relativně malé motory s ohledem na jejich užitečné zatížení.

Hlava robota se může pohybovat 350° horizontálně a 115° vertikálně.[9] Je nezbytnou součástí robota a obsahuje senzory Kinect, 5-megapixelovou barevnou kameru, dva páry stereokamer s úzkým a širokým záběrem a projektor textury.[9]

---

<sup>1</sup>[https://www.hokuyo-aut.jp/02sensor/07scanner/utm\\_30lx.html](https://www.hokuyo-aut.jp/02sensor/07scanner/utm_30lx.html)



Obrázek 2.1: Robot PR2<sup>a</sup>

<sup>a</sup>Převzato z <http://robotics.usc.edu/res1/robots/4/>

## 2.2 Robotický operační systém

Robotický operační systém (dále jen ROS) je flexibilní framework pro psaní robotického softwaru. Jedná se o soubor nástrojů, knihoven, které mají za cíl vytvořit komplexní a robustní prostředí použitelné na široké škále robotických platforem. Pro dosažení těchto cílů musí ROS poskytovat hardwarovou abstrakci, ovladače zařízení, knihovny, správu balíčků atd. [6]

ROS je navržen, aby byl co nejvíce modulární a distribuovaný. Uživatel si může vybrat do jaké míry chce ROS využít. Programování aplikace v ROSu je rozděleno do více uzlů, kde každý uzel implementuje určitou část problému například zpracování dat z kamery, řízení motorů apod. Celou aplikaci lze chápat tedy jako soubor procesů (uzlů). Tyto procesy lze jednoduše seskupovat do balíčku a lehce distribuovat. ROS poskytuje vývojářům možnost implementovat aplikace v různých programovacích jazycích například C++, python nebo LISP. Tyto vlastnosti a veřejná BSD licence udělaly ROS velice rozšířený a využívaný výzkumnými skupinami po celém světě.[7] Základní koncept ROSu je založen na peer-to-peer síti vzájemně komunikujících procesů - uzlů.

## Uzly

Uzly představují jednoduché procesy, přičemž každý proces implementuje jednotlivé části úkolu. Aplikace ROSu je tvořena z velkého počtu uzlů. Programátoři se snaží uzly dobře navrhnout pro jejich znovupoužitelnost, proto je možné nalézt na internetu mnoho naprogramovaných uzlů pro opakující se úlohy a jednoduše je použít. Rozdělení aplikace do uzlů nese další výhody. Při pádu aplikace, kdy chyba postihne určitý uzel případně uzly, je lze jednoduše znovu spustit. Při ladění detailů jako hodnoty určité proměnné lze mít spuštěnou celou aplikaci a určitý uzel překládat a znovu spouštět.

## Zprávy

Jednotlivé uzly mezi sebou komunikují pomocí zpráv. Zpráva je datová struktura obsahující jednoduché datové typy (integer, float, boolean) a pole složené z těchto základních typů. Je také možné jednotlivé zprávy skládat z jichž existujících zpráv a kombinovat je. Zprávy jsou posílány na předem určený, pojmenovaný zdroj. Má-li uzel o tyto zprávy zájem, přihlásí se k jejich odběru. Zprávy publikované z jednoho uzlu tímto způsobem může přijímat více uzlů. Komunikace pomocí zpráv umožňuje jednocestný přenos dat.

## Služby

Služby jsou další možností komunikace mezi uzly, která je založena na principu žádosti a odpovědi. Služby jsou definovány dvojicí zpráv, jednou pro dotaz a druhou pro odpověď. Uzel nabízí službu pod jménem. Klient pošle zprávu se žádostí určité služby a čeká na odpověď. Můžeme si to představit jako volání vzdálené procedury. Výsledkem může být vrácení hodnoty nebo provedení nějaké akce.

## 2.3 Nástroje

### Gazebo

Gazebo je aplikace pro simulování různých typů robotů ve venkovním i vnitřním prostředí. Pomocí tohoto nástroje je možné simulovat různé pohyby a činnosti robota bez rizika jeho poškození. Toto prostředí si můžeme sami vytvořit podle našich požadavků. Gazebo umožňuje interakci s prostředím a poskytuje data sensorům v reálném čase z virtuálního prostředí. Jako vstup pro popis prostředí a prvků robota je dnes ještě i používán starší formát URDF, standardizovaný v ROSu. Nicméně tento formát postrádá mnoho vlastností a není aktualizován s ohledem na dnešní měnící se potřeby robotiky. URDF může specifikovat pouze kinematické a dynamické vlastnosti robota. Tento formát není univerzální, neboť není možné určit paralelní vazby, tření a další vlastnosti. Tyto nedostatky řeší nový formát popisu simulace SDF<sup>2</sup>, který byl vytvořen pro použití v Gazebu. SDF je pro kompletní popis všeho od světa až po úroveň robota.[10]

---

<sup>2</sup>Simulator popis format



Obrázek 2.2: Aplikace Gazebo

## Rviz

Rviz je výkonný 3D vizualizační nástroj. Poskytuje pohodlné uživatelské rozhraní pro vizualizaci modelu robota a dat ze senzorů jako obraz z kamery, mračna bodů z kinectu apod. Rviz je poskytován se standardní distribucí ROSu.

## TF

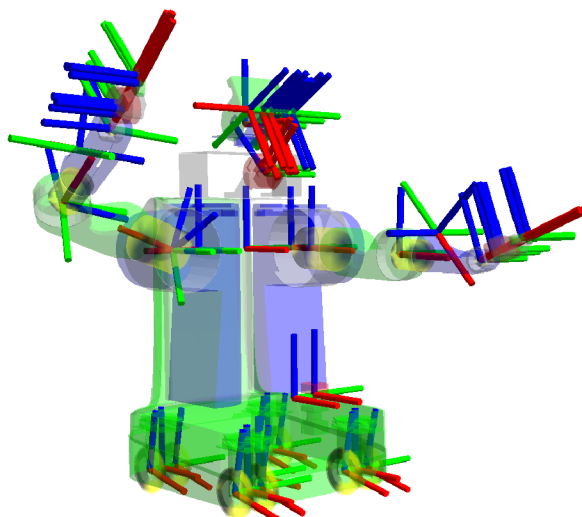
Robotický systém má typicky mnoho 3D koordinačních rámců, které se mění v průběhu času. Příkladem takovýchto rámců může být rámec hlavy, chapadla základny nebo prostředí. TF uchovává stavy všech těchto rámců a uživateli umožňuje získat informace:

- Kde byl rámec hlavy relativně k rámci prostředí před dvěma sekundami?
- Jaká je pozice objektu v chapadle relativně k základně?
- Jaká je aktuální pozice základny v rámci prostředí?

TF je schopno pracovat v distribuovaném systému, tzn. že neexistuje žádný centrální server transformačních informací. Všechny informace souřadnicových rámců robota jsou dostupné pro všechny ROS komponenty na libovolném počítači v systému. [5] Při běhu aplikace na více počítačích je důležitá synchronizace mezi nimi, aby nedocházelo ke zpoždění přijímání transformací. Pro práci s transformacemi existují dva způsoby, vysílání transformací a poslech transformací.

Při vytvoření vysílání transformace je potřeba nastavit relativní posun pozice a orientaci k rodičovskému rámci, pojmenovat vysílanou transformaci a ve vhodně zvolených časových intervalech ji publikovat.

Systém přijímá a ukládá všechny souřadnicové rámce vysílané v systému. Uživatel se dotazuje na transformaci mezi specifickými rámci a tuto informaci dostane v podobě zprávy.



Obrázek 2.3: Přehled transformačních rámců PR2 <sup>a</sup>

<sup>a</sup>Převzato z <http://library.isr.ist.utl.pt/docs/roswiki/tf.html>

Ta obsahuje kromě hlavičky a jména dětského rámce, také zprávu o transformaci složenou z vektoru a quaternionu.

## 2.4 Ovládací zařízení

V dnešní době existuje velké množství zařízení pro ovládání různých strojů, hraní her atd. V následujícím textu budou popsána vybraná zařízení, jejich funkčnost a poskytovaná data.

### Gamepad

Gamepad umožňuje ovládání pomocí dvou joysticků a několika tlačítek. Informace jsou posílány v podobě pole integer hodnot pro tlačítka a pro každý joystick zvlášť dvěma souřadnicemi posunutí vůči základní pozici. U zadních tlačítek dokonce umožňuje sledovat citlivost zmáčknutí. Gamepad je dodáván jako standardní součást robota PR2 a představuje tedy základní ovladač pro ovládání robota.

### 3D connexion space navigator

3D connexion (dále jen 3D myš) je jako polohovací zařízení hojně využíváno ve 3D modelovacích aplikacích pro své jednoduché a intuitivní ovládání. Uživateli dovoluje jednou rukou ovládat 6 stupňů volnosti a navozuje pocit manipulace s objektem přímo v ruce. Zařízení se skládá z 6DOF<sup>3</sup> joysticku. Ten nám umožňuje ovládat pozici v osách x, y a z a rotaci kolem těchto os. Joystick je doplněn dvěma tlačítky ze stran, jak můžeme vidět na obrázku 2.4 vlevo. Pro 3D connexion je v ROSu již vytvořen uzel, který poskytuje data ze šesti směrů posunutí či natočení joysticku a integer hodnotu zmáčknutých tlačítek.

<sup>3</sup>Degrees of freedom - stupňů volnosti





Obrázek 2.4: Ovládací zařízení 3D connexion space navigator<sup>a</sup> a gamepad<sup>b</sup>

<sup>a</sup>Převzato z <http://www.amazon.com/3Dconnexion-3DX-700028-SpaceNavigator-3D-Mouse/dp/B000LB7G00>

<sup>b</sup>Převzato z [http://gamethingdaily.com/?attachment\\_id=4565](http://gamethingdaily.com/?attachment_id=4565)

## 2D myš

Asi nejznámějším vybraným zařízením je 2D myš, se kterým většina uživatelů již pracovala. Toto zařízení využívá pro interakci mezi uživatelem a systémem v základu dvě tlačítka, rolovací kolečko a snímač posunutí ve dvou osách, které jsou v základním protokolu PS2 zabaleny do tříbytového pole. První byte zahrnuje informace o tlačítkách a zmáčknutí rolovacího kolečka, druhý a třetí informace o posunutí v osách x a y. Rozšířený protokol Intellimouse umožňuje v přidaném čtvrtém bytu číst informace z rolovacího kolečka jako posunutí v ose z. [1]

## 2.5 Pravidla pro tvorbu uživatelského rozhraní

Tvorba uživatelského rozhraní je stěžejní věcí při tvorbě aplikace. Proto byla zkoumáním a odvozením ze zkušeností vytvořena určitá pravidla, která je vhodné dodržovat. Například Benem Shneidermanem bylo navrženo 8 zlatých pravidel pro tvorbu uživatelského rozhraní viz. [4].

- **Usilovat o konzistenci**

Používání konzistentní terminologie, dodržování pravidla pro tvorbu rozhraní daného prostředí, směr k tvorbě stereotypů - stejné věci se dělají stejně, podobné věci se dělají podobně.

- **Široká skupina uživatelů**

Je důležité si před tvorbou aplikace ujasnit, jaká skupina uživatelů bude aplikaci využívat a podle toho rozhraní tvořit. Styl práce s aplikací různých uživatelů se může značně lišit a proto je důležité, aby existovaly alternativy k ovládání aplikace.

- **Informativní zpětná vazba**

Uživatel potřebuje být vhodně informován o výsledcích provedených akcí, zda-li akce proběhla úspěšně, či nikoli a proč. Je potřeba ovšem zvážit způsob zpětné vazby, aby aplikace zbytečně neobtěžovala. Rozlišujeme silnou a slabou zpětnou vazbu. Silnou zpětnou vazbu musí uživatel dát najevo, že vzal informaci o výsledku najevo a musí na ni reagovat výběrem některé z možností. U slabé zpětné vazby uživatel nemusí potvrzovat, že se s informací seznámil.



- **Navigace**  
Řada uživatelských úloh je složena z posloupnosti více akcí. Cílem je rozdělení rozhraní úloh na jednoduché logické kroky, které respektují pracovní postup. Příkladem může být objednávání zboží, kdy uživatel postupně vyplňuje informace o objednavce namísto jednoho okna s velkým formulářem.
- **Předcházení chybám**  
Uživatelské rozhraní by mělo minimalizovat možné chyby uživatele. V situaci, kdy však dojde k chybě, je potřeba uživatele informovat a nabídnout jednoduchý způsob vypořádání se s vzniklou chybou.
- **Jednoduchý krok zpět**  
Tato vlastnost uživateli umožňuje jednoduché vrácení akce a také podporuje zkoumání nových možností. Toto je rozdíl oproti znakově orientovaným uživatelským rozhráním, které tuto možnost nenabízí.
- **Předvídatelnost**  
Uživatel musí být řídicím prvkem rozhraní, nikoli tím, kdo plní rozkazy aplikace. Nepředvídatelnost aplikace způsobí, že uživatel ztratí nadvládu nad aplikací a stane se její obětí.
- **Snížení zatížení krátkodobé paměti**  
Zapamatování rozhraní není cílem. Naopak uživatel by při opakovaném použití měl mít přehled o aplikaci bez nutnosti si cokoli pamatovat.

Uvedená základní pravidla nelze brát doslova. Je potřeba s nimi nakládat podle dané situace, kde může mít smysl některé pravidlo zpřísnit, poupravit nebo rozšířit. Kromě těchto pravidel je vhodné dodržovat i další pravidla při vytváření uživatelského rozhraní například rozmístění ovládacích prvků. Uživatel prohlíží obsah obrazovky od levého horního rohu a dále po směru hodinových ručiček. Dále uživatel různé věkové skupiny ocení jiný vzhled aplikace. Je zřejmé, že děti ve školce budou mít jiné nároky než vysokoškolský student.

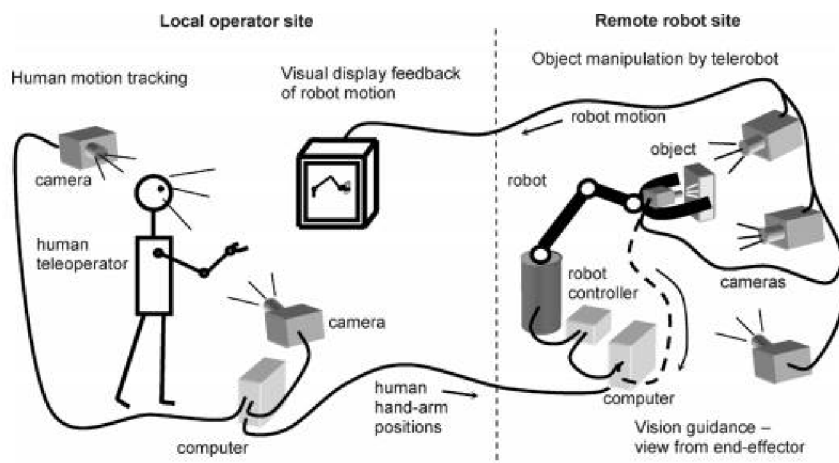
## 2.6 Existující řešení

Popularitou robotů a jejich možnostmi roste počet aplikací vytvořených pro jejich řízení, vizualizaci zkoumaných prostorů apod. Z existujících řešení jsem vybral diplomovou práci Ing. Kapinuse viz. [2], jehož aplikace se zabývá uživatelským rozhráním pro řízení servisního robota ovládacími zařízeními a projekt řízení ovládní robotického manipulátoru založený na obrazovém HRI<sup>4</sup> [3].

Řízení robotického manipulátoru založené na obrazovém rozhraní robot-člověk dovoluje manipulovat s objekty v prostředí, kde je účast člověka na místě nežádoucí. Tato práce ukazuje bezkontaktní obrazově založené rozhraní dovolující lidskému operátorovi provádět přirozené ovládní robotického manipulátoru v 6 stupních volnosti současně. Schéma reprezentace takovéto rozhraní robot-člověk můžeme vidět na obrázku 2.5. Uživatelův pohyb je snímán několika kamerami a převáděn na pohyb manipulátoru. Zpětnou vazbu dostává uživatel v podobě obrazu z kamer snímajících manipulátor a prostor kolem něj.

V aplikaci Ing. Kapinuse získává uživatel zpětnou vazbu také v podobě obrazů z kamer nebo vizualizace, k ovládní aplikace jsou pak použity zařízení gamepad, leap motion a P5

<sup>4</sup>Human-Robot-Interface - rozhraní robot-člověk



Obrázek 2.5: Schéma obrazově založeného rozhraní robot-člověk[3]

glove. Aplikace umožňuje ovládání robotického ramene v 6 stupních volnosti tzn. polohování v osách  $x$ ,  $y$ ,  $z$ . Objekt lze tedy uchopit zepředu s pevným nastavením orientace chapadla. Pro polohování ramene je využito virtuálního kurzoru ve tvaru kostičky. Virtuální kurzor představuje pozici, umístěnou uvnitř chapadla. Pohybujeme tedy s kostičkou a dosáhneme-li správné pozice, potvrdíme polohu gestem. Je-li rameno schopno dosáhnout této pozice, přesune se. Pro zavření a otevření chapadla jsou zde využita další gesta.

Celá aplikace je rozdělena do několika uzlů viz 2.6, které implementují jednotlivé části celé aplikace podle návrhové vzoru MVC<sup>5</sup>. Uzly jsou vyznačeny modrou barvou a zařízení, ze kterých přijímají data, zelenou.

## Hand transformation

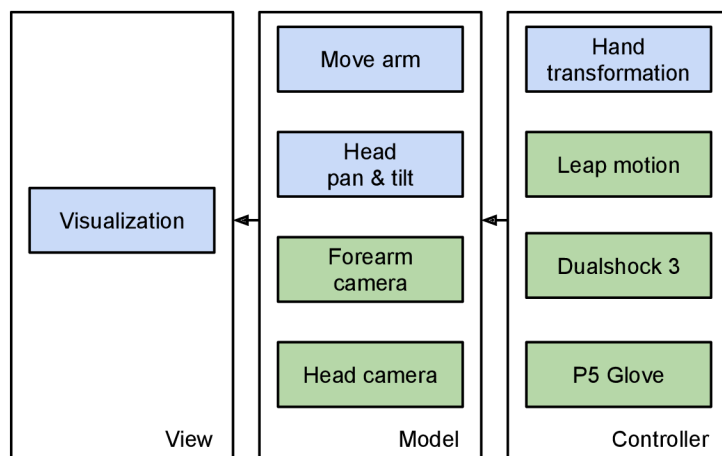
Problematiku sjednocení ovládacích zařízení do předem stanovené podoby informací předávaných pomocí zpráv zařizuje uzel `Hand_transformation`. Tento uzel je spuštěn s předem vybraným ovládacím zařízením a ostatním uzlům předává informace o aktuální pozici vykreslení virtuálního ukazatele, které se publikují ve zprávě `hand_model` s každou přijatou změnou polohovacího zařízení. Samozřejmě nezbytné pro běh uzlu je spuštění uzlu ovládacího zařízení, který publikuje data ze zařízení.

Informaci o požadované akci otevření a zavření chapadla představováno např. „kliknutím“ u `leap motion` publikuje ve zprávě `/but_pr2_hand_transformation/handAction` pomocí `ActionId`. Tato hodnota jednoznačně určí jaká akce se má vykonat. Pro přesunutí ramene do určité pozice publikuje zprávu `/but_pr2_hand_transformation/handPose` s hodnotami této pozice.

## Head pan & tilt

Tento uzel implementuje otáčení robotovi hlavy na pevně stanovený bod. Pevně nastavený bod na robotově „zápěstí“ je kontrolován s frekvencí 10Hz.

<sup>5</sup><http://en.wikipedia.org/wiki/Model-view-controller>



Obrázek 2.6: Schéma uzlů[2]

Přidanou funkcí je také filtrování dat point cloudu ze stereokamer umístěných na hlavě robota pro lepší přenos po síti. Point cloud je ořezán v ose z do 1 metru a je zmenšena velikost bodů.

## Move arm

Přesunutí ramene do požadované pozice má za úkol podle názvu uzel `Move arm`. Pro pohyb ramene využívá uzel knihovnu `MoveIt!`, kde na začátku použití se musí specifikovat skupina se kterou se bude pracovat, zde `left_arm` tzn. levé rameno. Tato skupina definuje počet a způsob propojení jednotlivých částí robota. Poté lze pro koncový efektor, v tomto případě chapadlo levého ramene, nastavit pozici a orientaci. Knihovna určí pomocí inverzní kinematiky natočení jednotlivých kloubů na rameni a přesunutí do stanovených souřadnic provede. Uzel získává souřadnice odběrem zprávy `/but_pr2_hand_transformation/handPose` publikované uzlem `Hand transformation`. Jak bylo psáno výše v této aplikaci je orientace chapadla pevně nastavena, proto se nevyužívá.

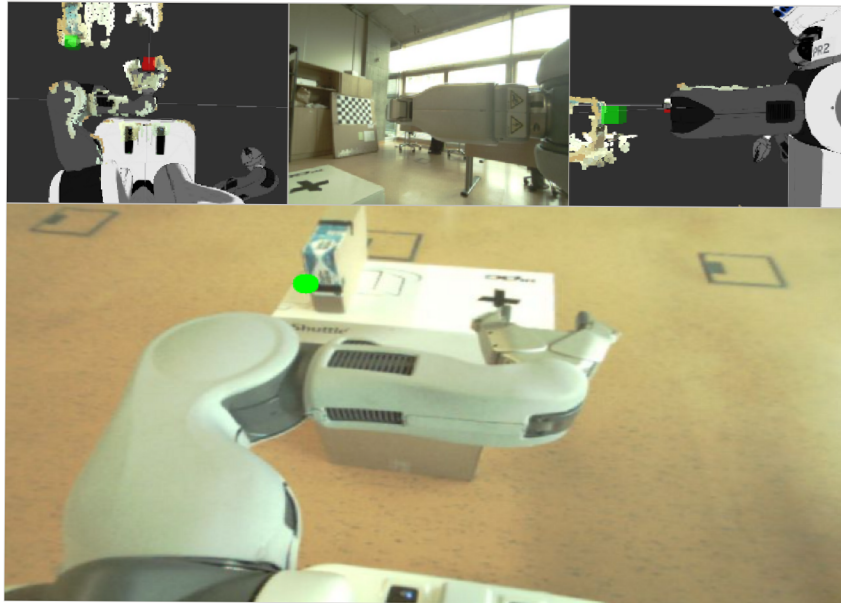
Druhým úkolem uzlu je otevírání a zavírání robotického chapadla. Zde je využita služba `l_gripper_controller/gripper_action`, kde lze nastavit míru rozevření chapadla a sílu stisku. Efektivním nastavením síly stisku lze dosáhnout pevného uchopení předmětu bez toho aniž by byl rozmáčknut. Informaci o otevření a zavření chapadla je zjištěna podle `ActionID` ze zprávy `/but_pr2_hand_transformation/handAction`.

## Vizualization

Uživatelské rozhraní implementuje uzel `Vizualization` pomocí frameworku Qt. Tento uzel má dvě verze. V první verzi je okno rozděleno do čtyř částí a uživatel si může nastavit v každé části jaký pohled na prostředí a robota chce zobrazit.

Druhou verzi rozhraní můžeme vidět na obrázku 2.7. Horní část je rozdělena do tří částí. Krajní části zobrazují vizualizaci generovanou knihovnou Rviz, kde je zobrazen point cloud upravený uzlem `Head pan & tilt` generovaný ze stereokamer umístěných na hlavě robota

a virtuální ukazatel. Uprostřed těchto pohledů je záběr z kamery na zápěstí robota. Největší část rozhraní zabírá pohled z kamery umístěné na hlavě robota. V tomto pohledu je vykreslen virtuální ukazatel ve tvaru kolečka, které mění svoji velikost v závislosti vzdálenosti virtuálního ukazatele od robota.



Obrázek 2.7: GUI verze 2 [2]

## Kapitola 3

# Ovládání pozice ruky robota

Hlavním cílem této kapitoly je čtenáři představit cíl, který tato práce řeší, návrh řešení celé aplikace a popis návrhu jednotlivých důležitých částí.

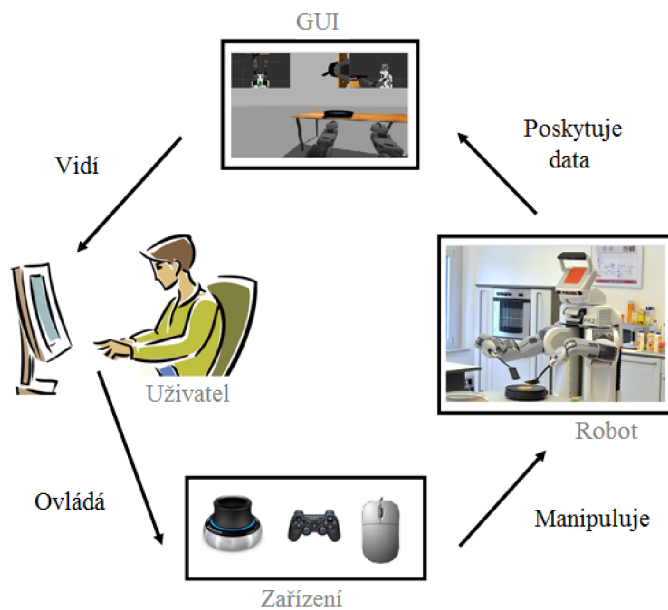
### 3.1 Specifikace zadání

Cílem této práce je vytvořit uživatelské rozhraní pro řízení robota při uchopování objektů a srovnání zařízení použitých k ovládání. Robot se pohybuje v lidském prostředí, proto může dojít k situacím, ve kterých není schopen daný úkol splnit nebo úloha vyžaduje specifický způsob chování. V takovýchto případech i přes velkou míru autonomního chování dnešních robotů potřebuje pomoc člověka se splnění daného úkolu.

Z analýzy problému je zřejmé, že spolu budou interagovat člověk s robotem. Člověk bude pomocí ovládacích zařízení řídit činnost robota. Robot bude vykonávat člověkem zadané akce a prostřednictvím senzorů poskytovat člověku zpětnou vazbu ve formě obrazu. Schéma HRI je znázorněno na obrázku 3.2. Výsledná aplikace by tedy měla uživateli poskytnout možnost řídit robotické rameno k uchopení objektu. Spuštění aplikace bude možné dvěma způsoby pro jednoduché (3DOF) a složitější (6DOF) úkoly řízení ramene. Mód 3DOF představuje možnost ovládání posunu pozice podél os  $x$ ,  $y$  a  $z$ . Při spuštění 6DOF módem přidáme ovládání rotací kolem těchto os.

K ovládání aplikace jsem vybral zařízení gamepad, 2D myš a 3D myš. Tato zařízení budou v experimentech srovnána pro zjištění optimálního ovládání. Objekt jako takový je velice široký pojem, proto se zaměřím na objekty méně rozměrné, běžně dostupné v domácím prostředí. Tyto objekty budou umístěny na stůl a nebude s nimi dále manipulováno mimo robota.

Pro potřeby ovládání robotického ramene ve všech stupních volnosti je velmi důležité vhodně navrhnout uživatelské rozhraní a další prvky usnadňující uživateli ovládat robota při uchopování objektu. Tyto problémy a jejich návrh řešení budou popsány v následujících částech kapitoly.



Obrázek 3.1: Koncept interakce člověk-robot

## 3.2 Způsob řešení problému

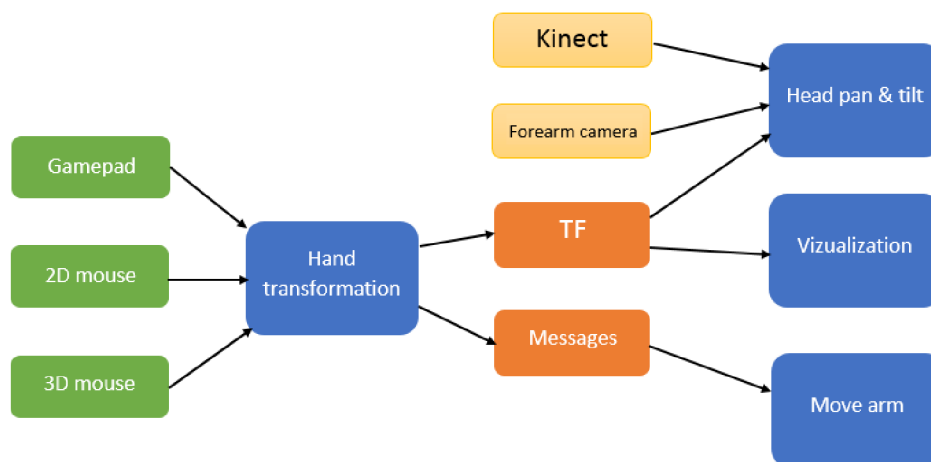
Celá aplikace bude rozdělena do několika uzlů, které budou implementovat jednotlivé úkoly otáčení hlavy, vykonání akcí požadovaných uživatelem, sjednocení informací přijatých z ovládacích zařízení do jednotné podoby a vizualizaci zpětné vazby uživateli.

Uzly jsou znázorněny na obrázku 3.2 (modře). Ve schématu můžeme vidět znázorněné vazby například uzel `Hand transformation` přijímá data od ovládacích zařízení (zeleně). Tyto informace zpracuje do specifikovaného formátu zpráv a dále vytvoří transformační rámce pro snadnější získání informace pozice koncového efektoru a vizualizaci dat (oranžově). Publikované zprávy přijme uzel `Move arm`, který přesune rameno na danou pozici nebo vykoná další určené akce z přijatých zpráv. S vytvořenými transformačními rámci bude pracovat uzel `Head pan & tilt` ke sledování rámce. Uzel `Vizualization` vhodně vizualizuje přijatá data z kinectu a kamery na předloktí také pomocí vytvořených koordinačních rámců.

Aplikace bude vyvíjena v několika fázích a po každé části otestována. Získaná užitečná data budou využita pro další fázi vývoje aplikace. V první části řešení bude implementováno ovládání robotického ramene s využitím zařízení gamepad a 3D myši a testovací verze uživatelského rozhraní. Zde budou získány informace důležité pro další vývoj uživatelského rozhraní a jeho části, citlivost a schopnost ovládání a jejich vhodná úprava.

V další fázi bude upraveno uživatelské rozhraní, virtuální kurzor a další zjištěné nedostatky. Po této části dojde k pilotnímu testování, kde mohou být zjištěny další chyby. Ty budou opraveny a následně provedeny experimenty s aplikací.





Obrázek 3.2: Koncept systému

### 3.3 Ovládání

Ovládání aplikace bude mít dva různé módy. Výběr módu bude určen způsobem spuštění uzlu. První mód ovládání je určen pro situace, kdy uživateli postačuje ovládání ramene robota bez možnosti ovládání pohybu „zápěstí“. Takovouto situaci si představíme jako přesunutí krabičky od čaje z prázdné pracovní plochy do prostoru bez překážek znemožňujících nám bez kolize předmět umístit.

Druhým způsobem ovládání je mód s ovládáním ve všech stupních volnosti robotického ramene, tzn. ovládání dovolující uživateli jakkoli nastavit rameno, je-li to v jeho možnostech. Případů k využití toho způsobu rapidně přibývá, ať už jenom schopnost nekonečné rotace „zápěstí“ nebo jeho natočení. Obvyčné otočení předmětu obráceně je s tímto módem velmi jednoduchou akcí. Vykližení nákupu z košíku nebo roztřídění hromady věcí je již také uskutečnitelné bez větších problémů.

Jako ovládací zařízení jsem vybral pro porovnání tři zařízení s různými principy ovládání a uživatelskou zkušeností s nimi. Vybraná zařízení nedovolují přímé ovládání ramene, protože není snímána pozice ruky v 3D prostoru. Bude tedy využito nepřímého ovládání. Zařízením bude uživatel ovládat pohyb tzv. virtuálního kurzoru. U nepřímého ovládání se využívá inverzní kinematiky, která po získání informace o pozici koncového efektoru naplánuje nastavení ostatních kloubů ramene tak, aby bylo dosaženo požadované pozice. Avšak při posunu ramene s každou změnou virtuálního kurzoru by docházelo k trhavému pohybu ruky, neboť by robot neustále přeplánovával umístění koncového efektoru. Z tohoto důvodu uživatel potvrdí vybranou pozici gestem a rameno se až poté přesune. Pro otevření a zavření chapadla budou zvolena další gesta.

Princip ovládání každého zařízení je jiný tím, pro jaké účely jsou primárně vytvořena a jak jsou ovládána. Například zařízení 3D myš je navrženo přímo pro polohování a orientaci ve 3D prostoru, zatímco 2D myš jen pro navigaci ve 2D. Dále počet tlačítek omezuje kombinace a složitost ovládání.

## 3.4 Mapování dat z ovládacích zařízení

V následujících částech popíšu návrh ovládání jednotlivými zařízeními. V rámci ovládání obou módů jsem se snažil zvolit ovládání tak, aby bylo jednotné pro obě varianty. Spuštění prvního módu zůstane ovládání stejné, ale možnost orientace se nebude projevovat. U každé změny pozice nebo orientace bude brán ohled na citlivost změny polohy myši nebo joysticku, tzn. bude-li uživatel chtít přesunout virtuální kurzor rychleji, vychýlí joystick více v daném směru, v případě pomalé změny méně.

### Gamepad

Gamepad se dvěma joysticky a řadou tlačítek není problémový pro návrh ovládání v obou módech. Pro polohování bude sloužit pravý joystick v ose x a y a levý joystick pro osu z. Po zmáčknutí pravého horního zadního tlačítka se změní režim joysticků z polohování na rotaci kolem jednotlivých os. Ovládací tlačítka v levé části gamepadu jsou zvolena pro potvrzení pozice (levé), otevření chapadla (horní) a zavření chapadla (spodní).

### 2D myš

Ovládání pomocí 2D myši v prvním módu není velký problém, ale s použitím druhého módu dojde ke komplikaci. Základní polohování myši v ose x a y zůstane stejné také pro polohování virtuálního kurzoru, rolovací kolečko poté v ose z. Stisknutím pravého tlačítka dojde k přepnutí režimu polohování na orientaci. Levým tlačítkem potvrdíme pozici. K otevření a zavření chapadla poslouží kombinace pravého tlačítka a levého.

### 3D myš

Pro ovládání polohování i orientace virtuálního kurzoru je velmi vhodný 6DOF joystick. Postranními dvěma tlačítky je ovládáno zavření a otevření chapadla (pravé) a potvrzení polohy (levé).

Po první implementaci ovládání zařízení 3D myš a následném testování byly zjištěny velmi špatné výsledky chybovosti ovládání a přesnosti. Uživatel jenž potřeboval mírně posunout pozici dopředu zároveň nechtěně upravil i orientaci. Zařízení je velmi citlivé na ovládání a vyžaduje delší dobu manipulace s ním, aby si uživatel alespoň částečně zvykl.

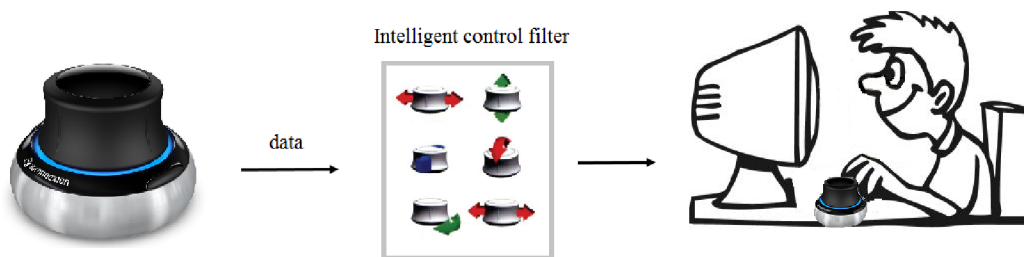
Po této zkušenosti jsem se rozhodl rozdělit ovládání do tří sekcí. První sekce je tzv. klidová, kde je joystick v nulové pozici ve všech osách, druhá sekce umožňuje jen ovládání pozice a třetí jen orientace. Při pohybu joysticku ke změně orientace nebo polohy přejde ovládání z klidové sekce do druhé nebo třetí sekce ovládání. Přejedání mezi možnostmi polohování a orientace jsem zvolil přes klidovou polohu joysticku, tzn. má-li uživatel nastavenou pozici a chce změnit polohu, vrátí joystick do nulové pozice a začne měnit orientaci. Tímto způsobem je odstraněna možnost orientování při polohování a naopak.

## 3.5 Vizualizace zpětné vazby

### Virtuální kurzor

Pro určení pozice a orientace uchopení objektu bude sloužit virtuální kurzor, jak bylo psáno výše. Návrh tohoto prvku je velmi významná věc, neboť uživateli představuje přesně místo a směr uchopení. Vymodelování přesného modelu chapadla není cílem aplikace. Pro možnost



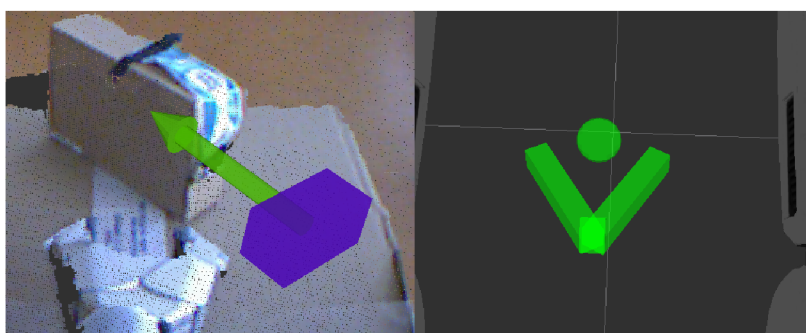


Obrázek 3.3: Schéma zpracování dat z 3D myši

obecného použití bez specifikace robota je nutno navrhnout obecný tvar virtuálního kurzoru, který představuje obecně místo a směr uchopení pro různé typy chapadel.

Pro účely prvního testování byl navržen jednoduchý virtuální kurzor složený z obdélníku v pozici koncového efektoru a ze šipky směřující do středu chapadla tj. bodu uchopení. Tento návrh můžeme vidět na obrázku 3.4 vlevo. Pro jeho nejednoznačnost a chybovost při práci s ním byl navržen nový virtuální kurzor, představující místo uchopení ve tvaru koule. Směr uchopení je složen ze základních geometrických objektů do tvaru „V“. Závěrečný návrh můžeme vidět na obrázku 3.4 vpravo.

Dalším úkolem při návrhu virtuálního ukazatele je ovládání jeho rotace. Jsou možné dva způsoby. První si lze představit jako pohyb ruky člověka jen v zápěstí (v obrázku 3.4 vpravo kolem vrcholu obrazce ve tvaru „V“), druhý celým ramenem kolem bodu uchopení (obrázek 3.4 vpravo tvar koule). V aplikaci bude využit druhý způsob, který umožňuje uživateli upravit polohu kurzoru na dotek s předmětem a dále už jen ovládáním rotací vybrat směr uchopení objektu.



Obrázek 3.4: Návrh virtuálního ukazatele

## Rozhraní

Při první fázi testování uživatelského rozhraní bylo využito existujících, popsané v existujících řešeních viz 2.7. Ze získaných informací od testujících uživatelů bylo zjištěno, že hlavní pohled z kamery umístěné na hlavě, zabírající větší část uživatelského rozhraní, většina uživatelů vůbec nesleduje.

Preferovanými pohledy byly virtuální scény s vizualizací robota a zobrazovaným markerem v levé a pravé horní části rozhraní. Zobrazují vhodněji místo uchopení a lépe poskytují uživateli orientaci v prostoru. Po diskuzi byl zhodnocen hlavní pohled jako, „ne úplně špatný“, ale se změnou zobrazením kurzoru stejného jako v pohledech do virtuální scény generovaných knihovnou Rviz po stranách horní části.

Jako možná odvážný nápad jsem navrhl uživatelské rozhraní s jedním pohledem s vizualizací scény point cloudem a zobrazením virtuálního kurzoru. Po vlastním testování tento návrh odhalil skutečnost, že člověk není schopen uchopit předmět z jednoho pohledu, přestože je scéna zobrazena ve 3D. Dochází k velké chybovosti odhadu vzdálenosti, pozice apod.

Po zhodnocení všech testování a získaných zkušeností navrhuji uživatelské rozhraní s hlavním pohledem z hlavy robota. V tomto pohledu bude však vykreslen virtuální kurzor ve 3D zobrazení. V horní části zobrazí uživateli možnosti pohledu do scény a na virtuální kurzor ze strany a shora.

Uživateli bude poskytnuta kompletní možnost vnímání scény. K přirozenému pohledu člověka do scény slouží obraz z hlavy robota. K odhadu vzdálenosti pozice od robota uplatní zobrazení scény shora, jako pomoc při nastavení výšky slouží pohled ze strany.

## Kapitola 4

# Vizualizační a řídicí komponenty

Poslední kapitola práce popisuje první část realizace pro základní testování, dále už konečnou implementaci všech částí se snahou vylepšení k efektivnějšímu použití aplikace. Na závěr je představen způsob testování aplikace, vyhodnocení výsledků a zhodnocení získaných informací.

### 4.1 Základ implementace

V rámci výsledné aplikace jsem se snažil maximálně použít již existující řešení. Ze všech dostupných distribucí ROSu jsem si vybral hydro. Ačkoli k tomuto tématu je většina aplikací programována pro starší verzi groovy, verze hydro je používána při práci s reálným robotem PR2 dostupným na naší fakultě. Dále také při využití simulátoru Gazebo je doporučená distribuce hydro.

Jako základ realizace jsem nainstaloval distribuci ROSu a zprovoznil simulační nástroj Gazebo. Simulační nástroj jsem použil vždy pro první fázi testování implementovaných částí aplikace a až po funkčním ověření byly vyzkoušeny na reálném robotovi. Také z hlediska bezpečnosti robotické platformy je testování funkčnosti v simulaci výhodnější.

Pro implementaci jsem se rozhodl využít strukturu a části existujícího řešení Ing. Kapinuse viz. 2.6. Po spuštění existující aplikace v simulaci jsem podrobněji nastudoval jeho části, princip každého uzlu, jeho úlohu a poskytované informace jiným uzlům.

Jak bylo psáno v návrhu řešení aplikace, jako první krok jsem si stanovil možnost ovládání robotického ramene ve všech dostupných osmi stupních volnosti pomocí ovladače gamepad a 3D myš. Tento cíl zahrnoval přidání možnosti orientace při ovládání a změnu tvaru virtuálního kurzoru k jasné viditelnosti změny orientace.

V uzlu `hand_transformation` jsem jednoduchou podmínkou přidal ovládání orientace pomocí gamepadu. Změna polohy a orientace virtuálního kurzoru je zvolena jako přírůstková k základní pozici, neboť ovládací zařízení poskytuje hodnoty od -1 do 1. Výsledná pozice je publikována ve zprávě typu `Pose` viz. 4.1. Ta zahrnuje zprávy pro pozici ve tvaru tří souřadnic bodu a orientaci v podobě quaternionu.

| Typ                                   | Název                    |
|---------------------------------------|--------------------------|
| <code>geometry_msgs/Point</code>      | <code>position</code>    |
| <code>geometry_msgs/Quaternion</code> | <code>orientation</code> |

Tabulka 4.1: Zpráva `geometry_msgs/Pose`

Tento uzel zahrnuje také implementaci virtuálního kurzoru. Nyní vzhledem k účelu pouhé viditelnosti orientace byl realizován základní navržený kurzor složený z markeru typu šipky a kostky. Jejich pozice je shodná s pozicí koncového efektoru ramene, lze tedy publikovat oba typy se stejnou pozicí. Ve zprávě je odlišný tudíž jen typ objektu zobrazeného markerem, tomu je nastavena velikost v jednotlivých osách k požadovanému tvaru.

Jelikož je virtuální kurzor zobrazován v aktuální poloze i poslední potvrzené, došlo k zvětšení publikovaných informací v čase tak, že publisher nestihl všechny publikovat. Bylo proto nutné deklarovat více stejných publisherů, aby byla zátěž rozložena.

Dále bylo potřeba přidat přesunutí ramene s požadovanou orientací. Tato část zjednodušila obsluhu přesunutí ramene v uzlu `hand_move` na pouhé přijmutí zprávy o dané poloze a přesunutí, bez sebemenšího předzpracování polohy.

K dokončení první části chybělo přidat transformaci dat ze zařízení 3D myši do definovaných formátů zpráv. Zde jsem využil stejně jako u gamepadu hodnot z joysticku jako přírůstkových, namapoval je na jednotlivé osy a tlačítka přiřadil k jednotlivým akcím. U tlačítka otevření a zavření chapadla bylo nutné vložit zpoždění, aby nedocházelo ke stálému otevírání a zavírání.

Po implementaci první části došlo k testování, vyhodnocení nedostatků a dalších vhodných úprav. Jejich změny jsou navrženy v kapitole 3. V dalších částech bude popsána konečná realizace důležitých prvků aplikace s problémy, které bylo nutné vyřešit.

## 4.2 Ovládací zařízení

Po první části implementace mapování a transformaci získaných dat do jednotné formy bylo nutné upravit ovládání 3D myši a nově vytvořit ovládání pro 2D myš popsané dále.

### 3D myš

Zjištěné problematické ovládání u 3D myši v režimu 6DOF, bylo dle návrhu rozděleno do tří sekcí. V nulové sekci je zařízení v klidové poloze, poskytuje nulové hodnoty. Při změně pozice nebo orientace po překročení hraniční hodnoty se přesune ze sekce nula do příslušné sekce podle toho, zda-li uživatel chce měnit polohu nebo orientaci. V těchto sekcích není možné zároveň upravovat polohu a orientaci virtuálního kurzoru. Klesne-li však hodnota všech tří os pod mezní hodnotu, vrátí se zpět do sekce nula. V případě, kdy uživatel chce při polohování začít měnit orientaci, vrátí se do klidové polohy a začne měnit orientaci virtuálního kurzoru.

### 2D myš

Pro zařízení 2D myš k použití v této práci neexistuje ovladač v ROSu. Tento ovladač bylo nutné vytvořit. Systémová data poskytující informace z 2D myši ve formě 3 bytů jsem vhodným způsobem vyseletoval a vložil do zprávy `sensor_msgs::Joy`, která publikuje v poli integer hodnot informaci o tlačítkách a v poli float hodnot informace o posunu v osách. Formát zprávy můžeme vidět v tabulce 4.2. Bylo využito protokolu PS2, protože rozšířený protokol Intellimouse potřebuje speciální nastavení a nechci uživatele při instalaci zatěžovat zbytečně složitým nastavováním. Použitím základního protokolu PS2 došlo k drobné úpravě ovládání polohování a orientaci v ose z. Ta lze měnit zmáčknutím prostředního tlačítka myši a posunutím myši v ose y, orientaci stejně s přidáním zmáčknutí pravého tlačítka.

| Typ       | Název   |
|-----------|---------|
| Header    | header  |
| float32[] | axes    |
| int32[]   | buttons |

Tabulka 4.2: Zpráva `sensor_msgs::Joy`

Systémová data z 2D myši jsou však publikována proti ostatním zařízením jen při změně zmáčknutí tlačítka nebo poloze jedné z os. Tento problém způsobil konstantní posunování kurzoru, i když nebylo s 2D myší manipulováno. Proto bylo nezbytné po zpracování dat je vždy vynulovat.

### Citlivost změny

K uživatelské nespokojenosti s časově náročnější změnou virtuálního ukazatele o větší vzdálenost nebo větší rotaci jsem se rozhodl změnit jen přímé přenesení hodnoty dělené konstantou na změnu polohy. Jako vhodné řešení jsem po testování zvolil tři konstanty, které reflektují míru uživatelské snahy změny. Získaná hodnota je porovnána se zvolenými rozsahy a poté dělena jednou z konstant. V případě gamepadu při velkém posunu virtuálním kurzorem uživatel zatlačí joystickem do krajních mezí směrem pohybu a virtuální kurzor se pohybuje rychleji. Naopak k doladění detailu pozice jen lehce postrčí.

## 4.3 Vizualizace scény a bodu uchopení

Pro potřeby vizualizace směru uchopení a lepší zobrazení scény byl vytvořen nový virtuální kurzor a upraveno rozhraní aplikace.

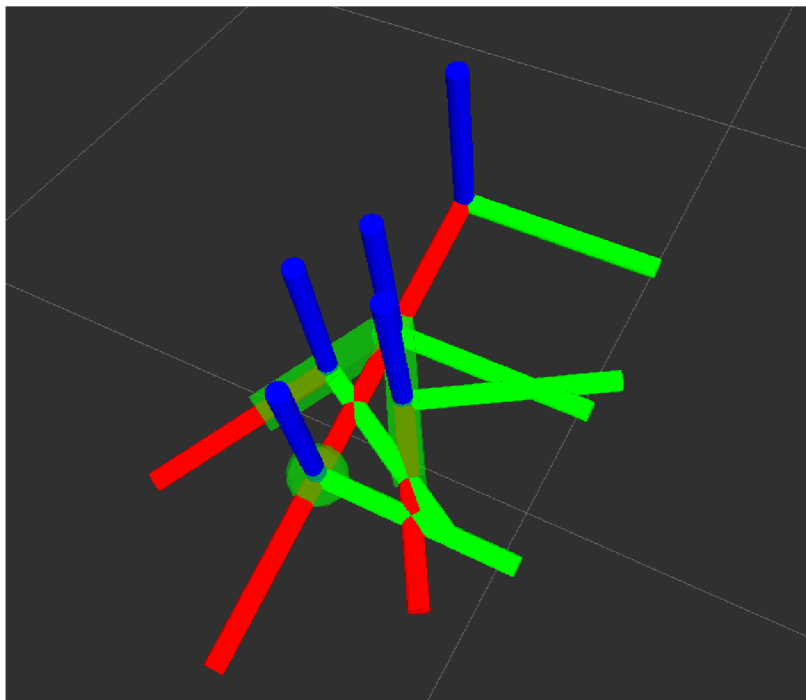
### Vizualizace virtuálního kurzoru

Navržený virtuální kurzor pro obecný robotický manipulátor bylo potřeba rozdělit na jednotlivé části. Tyto části jsou složeny z jednoho nebo více nově vytvořených transformačních rámců viz obrázek 4.1. První část složená z tří rámců `grasp_place` představuje bod uchopení. Bod reprezentovaný tvarem koule je rodičovským rámcem pro další část. Ta zahrnuje tři tří rámců nazvané `hand_model`, `left_finger` a `right_finger`. První jmenovaný je nastaven ve fixní vzdálenosti k bodu uchopení. Další dva s  $60^\circ$  svírajícím úhlem reprezentující směr uchopení do tvaru písmene „V“. Se specifickými hodnotami posunutí vůči bodu uchopení pro každý typ manipulátoru je vytvořen tří rámcem `end_effector`.

Pozice vykreslení markerů pro docílení navrhovaného vzhledu je přímo na koordinačních rámcích. Poslední přesunutou pozici značenou jinou barvou markeru je nutné vždy po potvrzení polohy získat z transformací vůči některému z fixních rámců robota například `torso_lift_link` pro každý rámcem zvlášť. Při zobrazení je pozice opět kreslena v porovnání s tímto rámcem.

Publikovaná pozice určená pro přesunutí ramene je získána z transformace rámcem `end_effector` oproti opět zvolenému fixnímu rámcu robota. Pro získání transformace mezi dvěma rámcemi ve formě zprávy `Pose` je vytvořena speciální funkce. Po získání výsledku z této funkce stačí přidat ke zprávě hlavičku a zprávu publikovat.





Obrázek 4.1: Realizace kurzoru tf

## Rozhraní aplikace

Využil jsem existujícího uzlu `visualization.2.0` a provedl v něm vlastní úpravy. V hlavním pohledu uživatelského rozhraní jsem použil přímo vizualizaci virtuální scény jako hlavní pohled. V této vizualizaci je zobrazen point cloud z kinectů umístěného na hlavě robota a virtuální kurzor. Pohled do scény je zafixován ke koordinačnímu rámci hlavy, ta sleduje bod na zápěstí robota.

Uprostřed horní části jsem nechal pohled z kamery umístěné na paži robota s virtuálním kurzorem transformovaným do 2D obrazu v podobě kruhu. Po stranách virtuální zobrazení scény s modelem robota, virtuálním kurzorem a point cloudem z kinectů byly upraveny směry pohledu. Tyto pohledy jsou zafixovány na stejný bod jako hlava robota s jiným směrem pohledu.

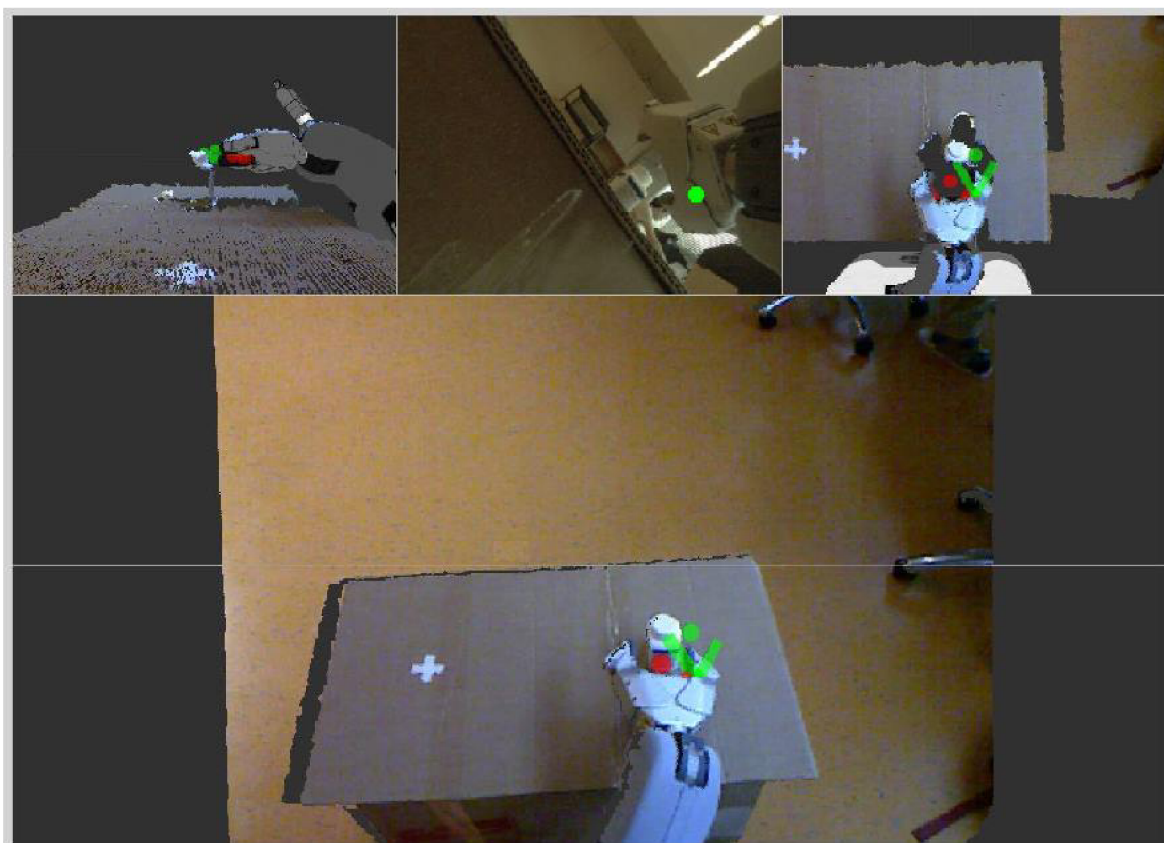
Možností ovládání aplikaci 2D myši vznikl problém reakce uživatelského rozhraní na události myši jako kliknutí, stisknuté tlačítko a posouvání. Tyto akce ve vizualizačním nástroji Rviz umožňují nastavení pohledu do scény, možnost přiblížení atd. V mém případě jsou však nežádané.

Z tohoto důvodu jsem `QWidgetům` nastavil ignorování událostí z klávesnice a myši. To způsobilo aplikaci nereagující na tyto události, avšak při určité kombinaci zmáčknutí tlačítek vyskakovaly do popředí okna z pozadí nebo vlastnosti hlavní plochy. Navíc pohled uprostřed horní části byl černobílý a vizualizované pohledy z Rvizu pořád reagovaly na myš.

Hledání způsobu ignorování událostí jsem přesunul do knihovny Rviz. Zde jsem našel dvě funkce pro přijímání událostí z myši a rolovacího kolečka. Ty jsem překryl prázdnou funkcí, která události dále nepropagovala ani nezpůsobovala volání dalších reakcí. Po této úpravě již Rviz vizualizace přestaly reagovat na události myši. V poslední fázi jsem zobrazení okna roztáhl přes celou obrazovku již při spuštění a skryl ukazatel myši.

Při pilotním testování jsem zjistil nedostatek zafixování pohledů vizualizace na bod robotova zápěstí. I přes vymezení oblasti pohybu virtuálního kurzoru jej nebylo někdy vůbec vidět. Navíc nedovoloval přesunutí ramene o větší vzdálenost neboť nešlo dál vidět, protože hlava sledovala pořád zápěstí. Rozhodl jsem se zafixování těchto pohledů na virtuální kurzor a hlavu robota také na sledování virtuálního kurzoru. To uživateli dovolí větší změny přesunu ramene a zároveň pohled bez nutnosti přesunu ruky do širšího okolí.

Bylo třeba vytvořit speciální transformační rámec reprezentující pozici virtuálního kurzoru bez jeho orientace nazvaný `virtual_cursor`. V případě fixování i s orientací kurzoru by se pohledy v horní části otáčely vzhůru nohama, takto jen sledovaly pozici. Výsledné uživatelské rozhraní je zobrazeno na obrázku 4.2.



Obrázek 4.2: Návrh GUI

## 4.4 Testování a experimenty

Testování aplikace je rozděleno do několika částí, které sledují jednotlivé části problematiky aplikace vedoucí ke kvalitní finální verzi.

V první části testování byla provedena po implementaci ovládání robotického ramene v 6 stupních volnosti, k tomu upravenou verzí virtuálního kurzoru a implementaci ovládání pro 3D myš. V této fázi testování byly sledovány nedostatky uživatelského rozhraní, ovladatelnost a citlivost ovládání s novým zařízením, vhodnost tvaru virtuálního kurzoru a jeho způsobu zobrazení místa uchopení.

Před závěrečnými experimenty proběhlo pilotní testování. S kolegou jsme vyzkoušeli celý průběh experimentů jednoho člověka. Získal jsem tak informaci o přibližné délce celého testu pro rozvržení testování. Tento test byl zároveň zvolen kvůli odhalení chyb a jejich eliminaci při následných experimentech.

### Popis experimentů

Při tvoření experimentů jsem se zaměřil na účel zjistit schopnost učení uživatelů při uchopování objektů, porovnání jednotlivých módů ovládání při stejných situacích, srovnání ovládacích zařízení a efektivitu vytvořeného uživatelské rozhraní.

K získání těchto informací jsem navrhl čtyři úkoly a dva trénovací, jeden pro mód 3DOF (A) a druhý 6DOF (B). Přehled těchto úkolů můžeme vidět na obrázku 4.3. Je zde zobrazená scéna před začátkem experimentu. V navržených experimentech má uživatel za úkol bez kolize s ostatními předměty:

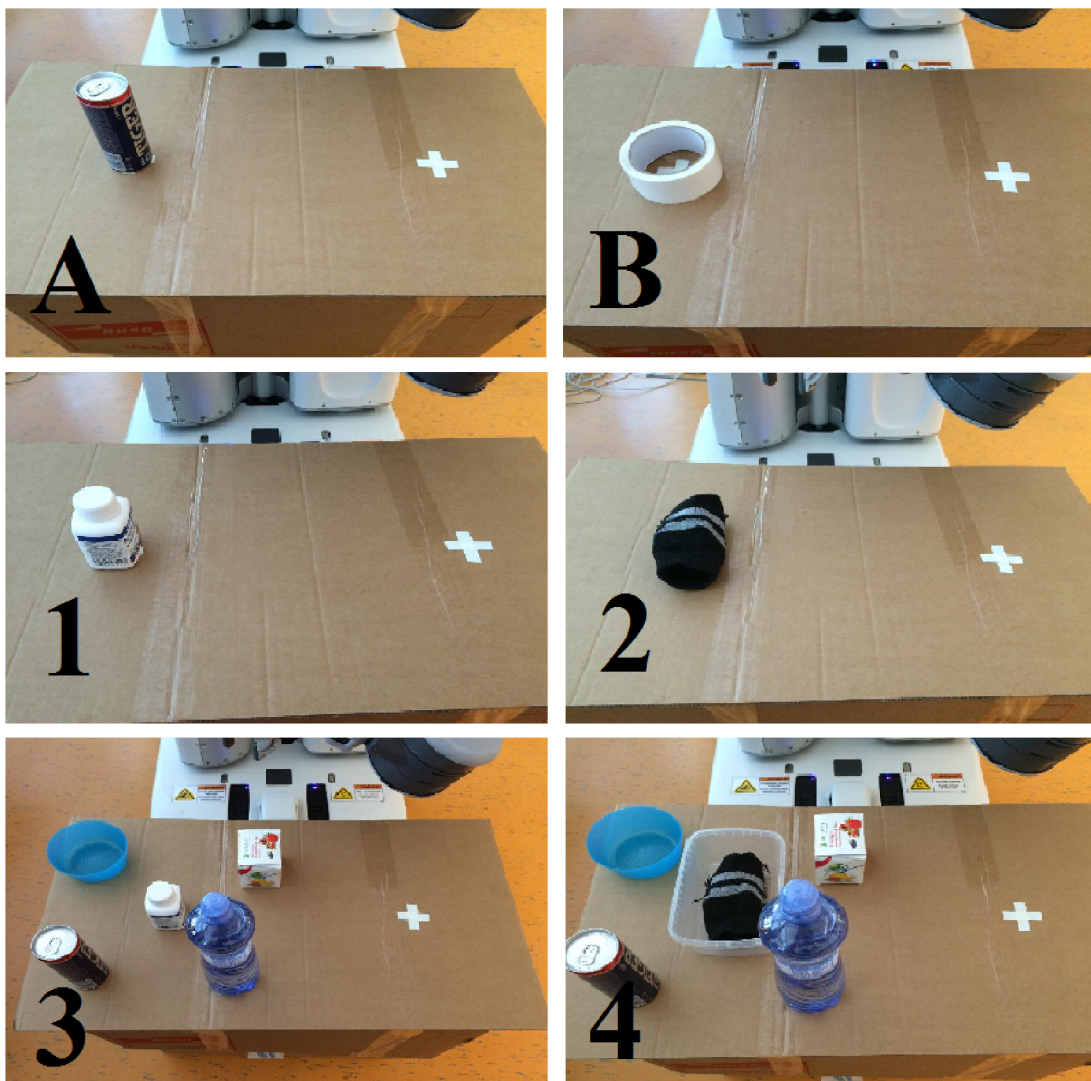
- 1 Uchopit zadaný předmět
- 2 Přemístit předmět na bílý křížek v druhé polovině pracovního stolu
- 3 Pustit předmět

Pro zjištění schopnosti učení uživatelů při práci při uchopování objektů robotem nezávisle na zařízení, módu ovládání a úloze byly pro všechny uživatele náhodně zvoleny módy ovládání, dále posloupnost zařízení a pořadí provedení úkolů pro každé zařízení. Testovaný člověk tedy provedl v jednom módu postupně s náhodně seřazenými zařízeními a v náhodném pořadí úkoly a poté v druhém módu opět s náhodným pořadím zařízení náhodně seřazené úkoly. Před každou změnou zařízení v daném módu bylo předvedeno ovládání aplikace daným zařízením a testující uživatel provedl trénovací úkol, pro mód 3DOF úkol A, 6DOF úkol B.

Pro každého testovaného člověka byla vygenerována jiná posloupnost módů, zařízení a úkolů. Následující posloupnosti představuje pořadí akcí pro prvního testovaného člověka:

1. Představení robotické platformy
2. Uvedení k pracovnímu místu
3. Představení všech ovládacích zařízení
4. Názorné předvedení aplikace gamepadem v 3DOF
5. Provedení trénovacího úkolu A gamepadem v 3DOF
6. Provedení úkolů 1-3 gamepadem v 3DOF





Obrázek 4.3: Přehled experimentů

7. Bod 4 až 6 zařízením 2D myš
8. Bod 4 až 6 zařízením 3D myš
9. Názorné předvedení aplikace 2D myši v 6DOF
10. Provedení trénovacího úkolu B 2D myši v 6DOF
11. Provedení úkolů 1-4 2D myši v 6DOF
12. Bod 9 až 12 zařízením 3D myš
13. Bod 9 až 12 zařízením gamepadem
14. Vyplnění dotazníku

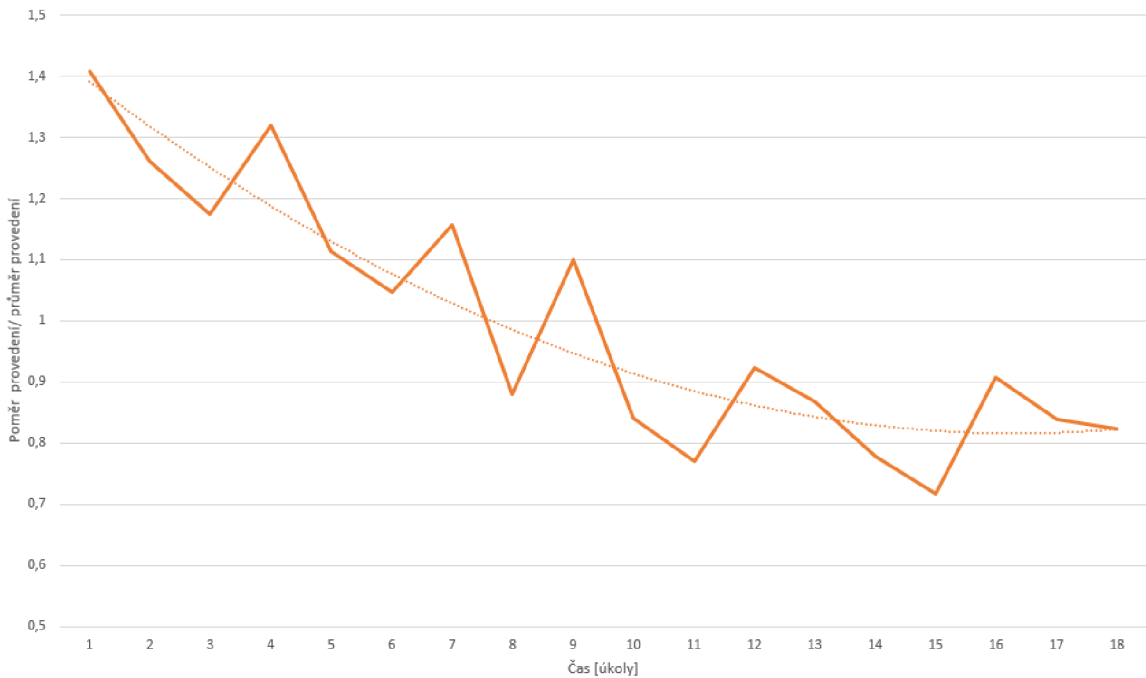
## 4.5 Vyhodnocení experimentů

V následujících částech budou vyhodnoceny zjištěné výsledky jednotlivých oblastí, o které jsem se zajímal. Získaná data nebyla dosažena jen z měření času provedení úkolu, ale také z dotazníku a vlastního sledování lidí při provádění experimentů.

### Křivka učení uchopování objektů

Vzhledem k netypickým úlohám pro běžného uživatele jsem navrhl experimenty tak, abych zjistil schopnost běžného uživatele učit se uchopování objektů robotem. Dokáže-li se i takový uživatel naučit ovládat robota při úlohách uchopení objektů nebo je-li to pro něj obtížné a vyžaduje to speciální přístup a delší dobu učení.

Z výsledků provedených experimentů bylo potřeba normalizovat naměřená data pro zobrazení křivky učení. Nejprve jsem normalizoval počet úloh pro oba módy ovládání, tedy testovací a čtvrtý úkol se v této statistice neprojevuje. Dále jsem každý úkol normalizoval průměrem úlohy prováděný určitým zařízením a dovedností uživatele. Ta byla vypočítána jako průměr všech jím provedených normalizovaných hodnot úkolů. Nakonec jsem seřadil všechny úkoly dle jejich provedení každým uživatelem a zprůměroval každý krok posloupnosti úkolů všech uživatelů dohromady. Vypočítané hodnoty jsem zanesl do grafu a s nimi zobrazil spojnicí trendu. Výsledný graf můžeme vidět na obrázku 4.4.



Obrázek 4.4: Graf učení

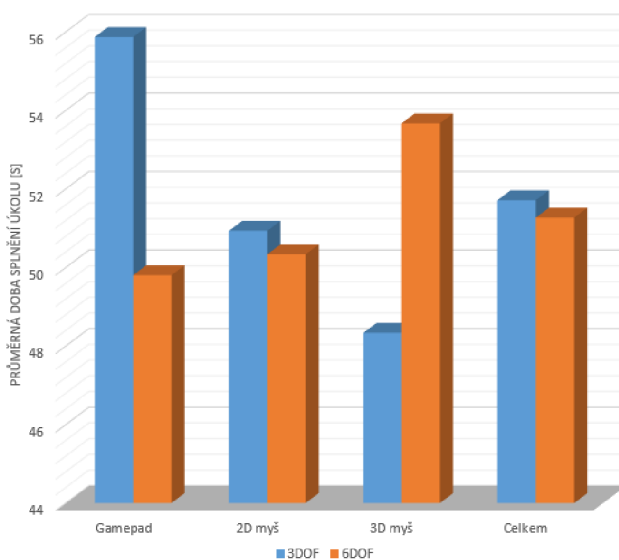
V grafu můžeme vidět na ose x 18 vynesných hodnot, tedy 3 úkoly každým zařízením ve dvou módech, představující časovou osu s jednotkou vykonaného úkolu. Na ose y je zobrazený poměr trvání úlohy a její průměrné doby normalizované dovedností uživatele. Je zřejmý trend všech uživatelů učit se pracovat s robotem při úlohách uchopení objektů. Dále je možné sledovat zhoršení po každé třetí úloze, to je způsobeno změnou zařízení.

Celkový trend hodnot však stále klesá až k hranici dvanáctého úkolu, kde uživatel dosáhl maxima schopnosti učit se plnit zadané úkoly. Uprostřed grafu jsou dva úkoly, které vybočují z trendu učení. To příkládám uživatelovu zájmu vyzkoušet možnosti manipulace s robotem a tedy delšímu provedení úlohy a neodpovídající trendu. Také z mého pozorování jsem dospěl k závěru, že při druhé polovině experimentů již uživatel znal zdroj informací o poloze virtuálního kurzoru, umístění předmětů apod. a jeho výsledky dosahovaly zrychlení.

## Porovnání módů ovládání a zařízení

Pro ověření smyslu rozdělení aplikace na dva módy pro jednoduché a složité úkoly jsem porovnal navržené úkoly 1-3, které lze provést oběma módy. Chtěl jsem tedy zjistit, jestli má mód 3DOF využití nebo jestli lze i jednoduché úkoly provést bez zhoršení doby vykonání úkolu módem 6DOF, případně pro jaké zařízení.

Ze způsobu posloupnosti provedení experimentů nebylo nutné pro potřeby porovnání módu ovládání a ovládacích zařízení naměřené hodnoty nijak normalizovat. Časy provedení všech vybraných úkolů zařízení v daném módu byly jen zprůměrovány a vyneseny do grafu 4.5.



Obrázek 4.5: Graf porovnání módů ovládání a ovládacích zařízení

Z vnesených dat je zřejmé, že porovnané celkové časy obou módů ovládání jsou téměř totožné. Tento výsledek porovnání jsem čekal, protože ovládání bylo navrženo tak, aby ovládání v módu 6DOF neovlivňovalo uživatele, nepotřebuje-li uživatel měnit orientaci.

Srovnání jednotlivých zařízení mě překvapuje v módu 3DOF, nejhorším ovládním je gamepad a nejlepším 3D myš. V druhém módu je tomu naopak. I přesto, že žádný uživatel se před testováním nesetkal s 3D myší, pro jednoduché účely polohování se zdá být nejnintuitivnějším, naopak s přidáním možností ovládat orientaci obtížným na ovladatelnost. To je také ale způsobeno žádnou zkušeností testovaných lidí se zařízením a novým způsobem ovládání. Avšak z pozorování testování, „šikovnější“ člověk si ji po chvíli velmi oblíbil

i pro možnost orientování. Třetí testované zařízení 2D myš v obou módech ovládání dopadl velmi podobně dokonce pod celkový průměr. Zde je vidět častá práce testujících s tímto zařízením.

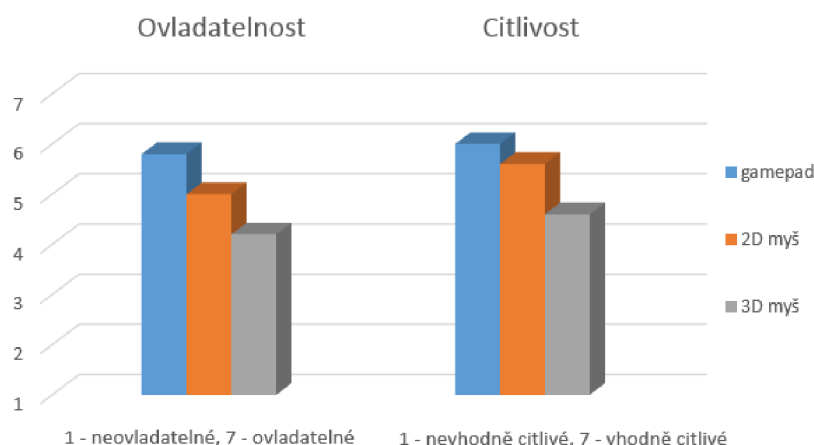
Možnost spuštění aplikace dvěma způsoby má smysl u 3D myši, u které doba trvání jednoduchých úkolů při ovládání 6DOF vzroste, naopak u gamepadu je výhodnější i pro tyto úkoly mód 6DOF. U tohoto srovnání si musíme uvědomit, že provedené úkoly byly navrženy docela jednoduché pro zvládnutí oběma módy ovládání. Složitější úkoly, navržené pro porovnání v situacích, kdy je potřeba měnit orientaci chapadla, jsou popsány a vyhodnoceny dále.

## Ovladatelnost a citlivost zařízení

Ke srovnání zařízení jsem se rozhodl využít dotazník pro získání hodnocení a srovnání zařízení od uživatele. V dotazníku ke zjištění celkové ovladatelnosti a citlivosti jednotlivých zařízení bylo využito metody získání informací zvané Likertova škála. U této metody je na danou otázku vybrána odpověď v rozmezí 1-7, kde je uveden význam krajních hodnot.

V následujícím grafu 4.6 jsou znázorněny průměrné hodnoty odpovědí ze škály 1-7 a jejich význam krajních hodnot. Ze získaných odpovědí vyplývá jako nejpřívětivější ovládání pro uživatele gamepad, naopak nejhorší 3D myš. V otázce citlivosti je výsledek porovnání velmi podobný.

Překvapivě i tady by se uživatelé při zvolení ovládacího zařízení až v poslední možnosti rozhodli pro 3D myš, kterou jsem předpokládal jako nejlepší pro ovládání 6DOF.



Obrázek 4.6: Graf citlivosti zařízení

## Chybovost

Chybovost uživatelů jsem definoval jako kontakt chapadla robota s předmětem, který nebyl vybrán pro manipulaci. Chybovostí jsem sledoval opět uživatelskou schopnost poradit si s netypickými úkoly ovládání robota.

Při pozorování uživatelů docházelo k chybovosti, tzn. naražení do jiného předmětu nebo dokonce shození předmětu, jen výjimečně, a to zejména na začátku experimentů. V této fázi

ještě uživatel nebyl zvyklý na způsob plánování trajektorie přesunu ramene. Příklad situace si můžeme uvést, kdy testující člověk měl vzít předmět umístěný mezi jinými předměty. Zadaný objekt správně uchopil avšak při přemístění naplánoval přesun ramene přímo na cílovou pozici. Plánovač naplánoval pohyb nejkratší možnou cestou a při přesunu chapadlo narazilo nebo shodilo předměty stojící vedle uchopeného předmětu.

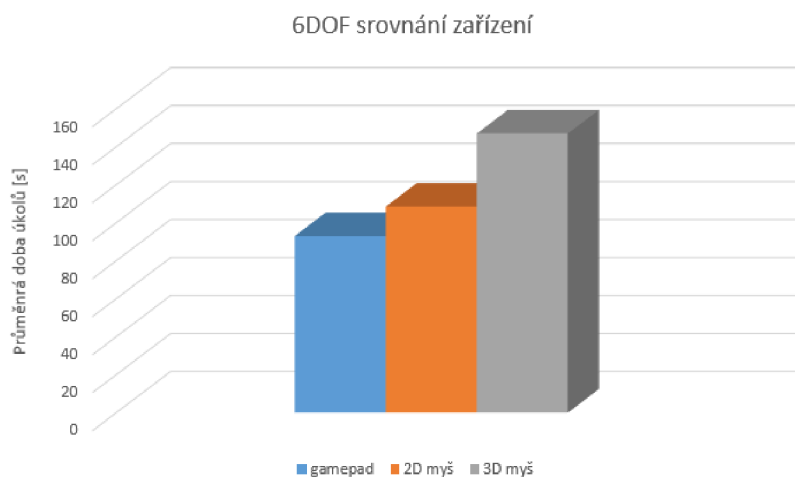
## Vizualizace kurzoru

Otázka vizualizace je myšlena na jasný přehled orientace kurzoru v různých natočeních a celková viditelnost ve scéně. K této zkoumané oblasti části aplikace bylo opět využito Likertovy škály. V našem případě význam krajních hodnot je nevhodnost/vhodnost (1/7) vizualizace. Z pěti testovaných lidí byla průměrná hodnota zjištěna 5,2.

Ze získané hodnoty lze usoudit, že většina uživatelů byla spokojena s viditelností směru uchopení při různých situacích. Vzhledem k fixování pohledů na virtuální kurzor je výsledek celkové viditelnosti očekávaný a nemohlo se tedy stát, že by nešel kurzor vidět vůbec.

## Srovnání zařízení 6DOF ovládání

V předchozím srovnání módů a zařízení jsem porovnal jednoduché úkoly. Ke srovnání zařízení při nutnosti využití i změny orientace byly využity trénovací úkol B a úkol 4. Toto zhodnocení ukáže srovnání ovládacích zařízení a jejich možnost využití v běžných situacích, kdy není vždy dostatek prostoru kolem předmětu nebo předmět vyžaduje různý směr uchopení. Výsledné časy byly zprůměrovány pro každé zařízení a vyneseny do grafu 4.7.



Obrázek 4.7: Graf ovladatelnosti zařízení

Lze vidět srovnatelné výsledky s jednoduchými úlohami i hodnocením z dotazníku. Nejlépe dopadl gamepad, dále 2D myš a nejhůře 3D myš.



# Kapitola 5

## Závěr

Cílem této práce bylo zpracovat teoretické informace z oblasti robotiky, návrhu uživatelského rozhraní pro řízení robota při uchopování objektů, vybrat vhodná zařízení pro ovládání a následně toto rozhraní navrhnout, implementovat a provedené experimenty s aplikací a jejím ovládáním vyhodnotit.

Práci jsem započal studováním nástrojů využívajících se v robotice a základy tvorby uživatelských rozhraní. V rámci studia jsem vyhledal mé práci podobná existující řešení, zhodnotil jejich přínos a možnost případného využití. Po získání dostatečných znalostí jsem navrhl uživatelské rozhraní, způsob ovládání a vybral ovládací zařízení. Moje řešení rozšiřuje práci pana Kapinuse [2] o možnost ovládání rotace robotova chapadla kolem os  $x$ ,  $y$  a  $z$ , s tím spojené úpravy vizualizace zpětné vazby uživateli, ovládání a implementaci vlastních vybraných ovládacích zařízení. Po několika fázích implementace a testování jsem provedl závěrečné experimenty navržené ke zjištění vhodnosti úprav pro rozšířené ovládání, schopnosti učení uživatele při úlohách uchopení objektů robotem a srovnání jednotlivých zařízení.

Z experimentů vyplynulo, že přes velmi netypické úlohy pro běžného člověka je schopen se učit pracovat při úlohách uchopení objektu robotem a již po několika úkolech dosahuje dobrých výsledků. V části ovládání aplikace v celkovém srovnání obou způsobů ovládání se hodnoty nijak výrazně nelišily. U zařízení gamepad a 2D myš je možné aplikaci spouštět v režimu 6DOF bez vlivu na řešení jednoduchých úkolů, naopak u 3D myši to nedoporučuji, protože vykazovala horší výsledky. V porovnání ovládacích zařízení se nejlépe osvědčil gamepad v režimu 6DOF a 3D myš v 3DOF. Překvapivě nejhůře si vedla 3D myš v 6DOF, přestože je primárně vytvořena pro ovládání 3D modelů. 2D myš neočekávaně obstála v ovládání manipulátoru, dokonce lépe než 3D myš. Informace z dotazníku potvrdily výsledky získané měřením z hlediska porovnání jednotlivých zařízení. Většina uživatelů byla také spokojena s vizualizací místa uchopení představované virtuálním ukazatelem.

Vytvořenou aplikaci je možné rozšířit o jiné způsoby snímání uživatele například založené na obrazovém snímání. Dále schopnost ovládání obou rukou robota nebo jeho základny pro větší možnost oblasti působení a složitější úkoly. Přidáním autonomních prvků, jako kolize s objekty, citlivost při uchopení, rozpoznání objektů a nabídky možností uchopení, jsou pak složitější rozšíření vhodné pro další rozvoj.

# Literatura

- [1] Chapweske, A.: The PS/2 Mouse Interface [online]. <http://www.computer-engineering.org/ps2mouse/>, [cit. 2015-01-05].
- [2] Kapinus, M.: *Uživatelské rozhraní pro řízení servisního robota*. Diplomová práce, Vysoké učení technické v Brně, 2014.
- [3] Kofman, J.; Wu, X.; Luu, T.; aj.: Teleoperation of a robot manipulator using a vision- based human-robot interface. *Industrial Electronics, IEEE Transactions on*, ročník 52, č. 5, Oct 2005: s. 1206–1219, ISSN 0278-0046, doi:10.1109/TIE.2005.855696.
- [4] Shneiderman, B.: *Designing the User Interface: Strategies for Effective Human-computer Interaction*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1986, ISBN 0-201-16505-8.
- [5] WWW stránky: tf [online]. <http://wiki.ros.org/tf>, [cit. 2014-01-20].
- [6] WWW stránky: About ROS [online]. <http://www.ros.org/about-ros/>, [cit. 2014-12-17].
- [7] WWW stránky: Is ROS for me [online]. <http://www.ros.org/is-ros-for-me/>, [cit. 2014-12-17].
- [8] WWW stránky: Hardware and Software Platform for Mobile Manipulation R&D [online]. <https://www.willowgarage.com/pages/pr2/design>, [cit. 2015-01-05].
- [9] WWW stránky: Hardware Specs [online]. <https://www.willowgarage.com/pages/pr2/specs>, [cit. 2015-01-05].
- [10] WWW stránky: URDF v Gazebo [online]. <http://gazebo.org/tutorials>, [cit. 2015-1-20].

# Příloha A

## Obsah CD

- Manuál k instalaci všech závislostí a překlad aplikace pojmenovaný **INSTALL.txt**
- Tato práce ve formátu pdf
- Zdrojový kód programu v adresáři **/src**
- Zdrojové kódy tohoto dokumentu v adresáři **/tz\_src**
- Video k prezentování aplikace
- Plakát